

Руководство администратора

MWS Container Platform

2025

Содержание

Введение	9
Архитектура.....	10
Архитектурные концепции	10
Принципы архитектурного проектирования	11
Визуализация архитектуры	11
Пользовательский интерфейс	13
Обзор.....	13
Инструменты CLI	15
Работа с кластерами.....	19
Типы кластеров.....	19
Поддержка мультиоблаков и гибридных облаков	20
Особенности реализации и ограничения	20
Рекомендации по управлению кластерами	20
Создание кластера On-Premise	21
Сеть	30
Kube-OVN.....	30
ALB	32
MetalLB	43
Совместимость аннотаций ingress-nginx	45
TCP/HTTP Keepalive	52
ModSecurity.....	53
Сравнение различных методов Ingress	58
HTTP Redirect.....	59
L4/L7 Таймаут.....	60
GatewayAPI.....	61
Создание сервисов.....	61
Ingress	65
Настройка Gateway.....	70
Создание доменного имени	77
Создание сертификатов.....	79
Создание пула внешних IP-адресов.....	79
Создание BGP-пиров	83

Настройка подсетей	84
Создание сетевых политик	94
Настройка сетевых политик кластера.....	96
Подготовка физической сети Kube-OVN Underlay.....	98
Автоматическое взаимное подключение подсетей Underlay и Overlay	101
Настройка балансировщика нагрузки	102
Настройка правил.....	109
Выделение ресурсов CPU и памяти	113
Проброс IPv6-трафика на IPv4-адреса внутри кластера.....	115
Kube-OVN Overlay Network настройка IPsec	116
Хранение	121
Введение	121
Основные концепции.....	121
Persistent Volume	123
Режимы доступа и режимы томов.....	126
Создание Storage Class типа CephFS File Storage.....	127
Создание класса блочного хранилища CephRBD.....	128
Создание локального Storage Class TopoLVM.....	129
Создание класса хранения NFS Shared Storage.....	131
Развёртывание компонента Volume Snapshot	133
Создание PV	133
Создание PVC	136
Использование снимков томов.....	138
Generic ephemeral volumes	141
Использование emptyDir	142
Интеграция стороннего хранилища	143
Распределённое хранилище Ceph в MWS-CP	145
Объектное хранилище с MimIO	186
Локальное хранилище TopoLVM	195
Безопасность.....	202
MWS Container Security.....	202
Архитектура	202
Процесс управления уязвимостями	205

MWS Compliance с Kyverno	207
API Refiner	208
Пользователи	209
Группы пользователей	216
Роли	219
IDP	224
Политики безопасности	232
Проекты	235
Обзор	235
Создание проекта	235
Редактирование проекта	237
Управление кластером проекта	238
Управление участниками проекта	239
Удаление проекта	241
Аудит	243
Обзор	243
Поиск и просмотр событий платформы	243
Результаты поиска	244
Инструменты самообслуживания для разработчиков	245
Обзор	245
Пространства имён	248
Настройка Secrets	256
Создание приложений	260
Создание HPA	270
Создание VPA	272
Описание статусов	274
Управление приложениями	275
Мониторинг	275
Логирование	277
События	278
Вход в контейнер через EXEC	279
Сетевая изоляция узлов	280
GitOps	283

Введение в MWS GitOps	283
Архитектура	283
Возможности	284
Установка MWS GitOps.....	285
Управление приложениями	286
Расширения.....	290
Operator.....	291
Cluster Plugin	292
Service Integration Platform.....	Ошибка! Закладка не определена.
Обзор.....	294
Возможности	294
Наблюдаемость	297
Преимущества	297
Возможности	298
Мониторинг	299
Трассировка	315
Логи	320
Работа с GPU	333
Обзор.....	333
Установка Kubernetes Hardware accelerator Toolkit.....	334
Настройка аппаратного ускорителя на GPU-узлах	334
Мониторинг ресурсов GPU	339
Управление затратами	342
Преимущества	342
Сценарии использования	342
Ограничения использования	343
Архитектура	344
Описание методов биллинга.....	345
Установка	346
Учёт и выставление счетов	348
Статистика потребления	349
Сервисы данных	352
Архитектура	352

Поддерживаемые сервисы данных.....	353
Установка	359
Обновление	364
MySQL-PXC.....	364
MySQL-MGR.....	372
Redis.....	377
Kafka.....	384
RabbitMQ.....	388
PostgreSQL	392
Управление шаблонами параметров	396
Управление резервным копированием	397
Управление инспекциями	400
Мониторинг и оповещение	402
Обновление экземпляра сервисов данных	404
Service Mesh	405
Istio.....	405
Полный жизненный цикл управления Service Mesh	407
Глобальная визуализация сервисов.....	408
Управление выпуском сервисов.....	409
Управление безопасностью сервисов	410
Визуализированная эксплуатация и сопровождение сервисов.....	410
Восстановление сервисов после сбоев	412
Модели развёртывания Service Mesh	412
Компоненты Mesh	414
Создание Service Mesh.....	416
Установка агента узла Istio CNI	420
Установка Istioctl	424
Мониторинг компонентов Mesh	424
Управление пространством имен в Service Mesh.....	425
Добавление сервиса	425
Развёртывание Ingress Gateway	430
Управление конфигурацией Sidecar.....	431
Распределенное трассирование	433

Мультикластерное развёртывание.....	437
Развёртывание Bookinfo	441
DevOps	444
Введение в MWS DevOps	444
Архитектура	444
Обзор возможностей	445
Пайплайны.....	446
Коннекторы.....	464
GitLab	469
Harbor	477
SonarQube	484
Nexus.....	487
AI	490
Введение в MLOps и LLMOps	490
Обзор возможностей.....	490
Установка	495
Управление пространствами имен	497
Развёртывание модели и вывод в эксплуатацию	498
Мониторинг и логирование	501
Виртуализация	503
Запуск виртуальных машин и контейнеров в одном окружении.....	503
Особенности	504
Функциональные возможности	504
Ограничения и условия	505
Установка	505
Добавление образов виртуальных машин	507
Обновление/удаление образов виртуальных машин	509
Обновление/удаление учётных данных образа	509
Создание образов Windows на основе ISO с использованием KubeVirt	509
Создание образов Linux на основе ISO с использованием KubeVirt	517
Экспорт образов виртуальных машин.....	524
Создание виртуальных машин/групп виртуальных машин	526
Вход в виртуальную машину с использованием VNC	530

Управление виртуальными машинами	531
Мониторинг и оповещения.....	533
Настройка проброса USB-хоста.....	535
Клонирование виртуальной машины	537
Контроль сетевых запросов виртуальной машины через Network Policy	540
Хранилище	541
Резервное копирование и восстановление	543
Приложение А.....	546
Приложение Б.....	553

Введение

В настоящем документе приводятся сведения о программном изделии "MWS Container Platform" для администратора программного обеспечения в следующем объеме:

- общее описание модулей ПО "MWS Container Platform", в том числе технологических концепций;
- архитектурные особенности модулей ПО "MWS Container Platform";
- описание возможностей работы с ПО "MWS Container Platform", используя CLI (command-line interface) и web интерфейс.

Настоящий документ предназначен для инженерно-технических специалистов, осуществляющих эксплуатацию, поддержку и сопровождение ПО "MWS Container Platform"; представляет собой руководство для администратора платформы контейнеризации на базе Kubernetes. Для успешного применения данного руководства и безопасной эксплуатации системы администратор должен обладать компетенциями в следующих областях:

- администрирование операционных систем семейства Linux;
- сети передачи данных (TCP/IP, DNS, VPN, балансировка, безопасность);
- Kubernetes (управление кластерами, масштабирование, обновления);
- контейнеризация (Docker и аналоги);
- DevOps (CI/CD, GitOps, IaC);
- Service Mesh (маршрутизация, безопасность, observability);
- сервисы данных (БД, очереди, хранилища);
- AI/ML-сервисы (жизненный цикл моделей, работа с GPU).

Архитектура

Архитектура MWS Container Platform (MWS-CP) разработана для обеспечения надежного, масштабируемого и гибкого облачно-нативного инфраструктурного решения. В этом документе изложены основные архитектурные принципы, типы кластеров и проектные соображения, которые лежат в основе нашей платформы.

Архитектурные концепции

Кластер **global**

Кластер **global** является центральной системой MWS Container Platform. Он служит основным управляющим и контрольным уровнем для всей платформы и отвечает за:

- Управление кластерами на уровне всей платформы
- Управление арендаторами и контроль доступа
- Глобальное планирование и оркестрацию
- Конфигурацию и применение политик на уровне платформы
- Мониторинг и наблюдаемость экосистемы

Ключевые характеристики кластера **global** включают:

- Выделенные физические ресурсы
- Изоляция от обработки рабочих нагрузок
- Централизованные возможности управления
- Высокая доступность и настройка безопасности

Кластеры рабочих нагрузок

Кластеры рабочих нагрузок — это среды выполнения, предназначенные для размещения и запуска реальных приложений и сервисов. Кластеры:

- Предназначены для запуска рабочих нагрузок приложений,
- Обеспечивают горизонтальное масштабирование,
- Гибки для различных типов инфраструктуры,
- Управляются и оркестрируются кластером **global**.

Принципы архитектурного проектирования

Изоляция управляющей плоскости

Архитектура предусматривает строгое разделение между кластером **global** и кластерами рабочих нагрузок:

- Кластер **global** использует выделенные физические ресурсы
- Предотвращает конкуренцию за ресурсы и снижение производительности
- Обеспечивает стабильное и безопасное управление платформой

Операционная гибкость

MWS-CP поддерживает несколько сценариев эксплуатации:

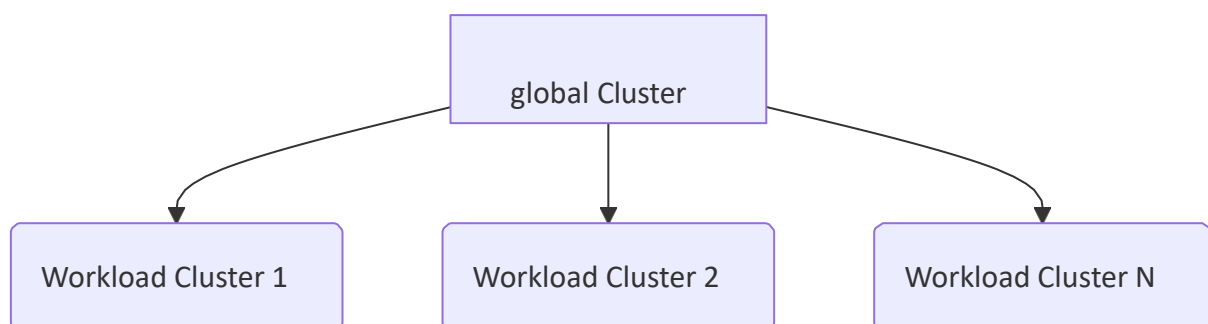
- Полностью разделённые кластеры **global** и рабочих нагрузок (рекомендуется для продакшена)
- Совмещённые конфигурации для разработки и сред с ограниченными ресурсами
- Гибридные и мультиоблачные архитектуры

Доступ и управление кластерами

Существует два основных способа интеграции кластеров рабочих нагрузок:

- Создание новых кластеров рабочих нагрузок
 - Рекомендуемый подход
 - Полностью управляются кластером **global**
 - Единая конфигурация и соответствие требованиям
 - Упрощённое управление жизненным циклом
- Подключение существующих кластеров
 - Поддержка подключения уже существующих Kubernetes кластеров
 - Требуются проверки версии и совместимости
 - Стандартизированные протоколы доступа и управления

Визуализация архитектуры



Ключевые преимущества данной архитектуры:

Повышенная безопасность: изолированные управляющая и исполнительная плоскости

Масштабируемость: простое добавление новых кластеров рабочих нагрузок

Гибкость: поддержка разнообразных инфраструктурных сред

Эффективность: централизованное управление с распределённой обработкой

Соответствие требованиям: единообразное применение политик в кластерах

Пользовательский интерфейс

Обзор

Веб-консоль MWS Container Platform предоставляет графический интерфейс для визуализации данных вашего проекта и выполнения административных, управленческих и диагностических задач. Веб-консоль разработана для улучшения пользовательского опыта за счёт организации функционала платформы в отдельные перспективы, ориентированные на разные роли пользователей и рабочие процессы.

Страница портала

После входа в веб-консоль по умолчанию открывается страница Портала. На странице Портала отображаются все доступные карточки представлений, поддерживаемые веб-консолью. Эти карточки служат точками входа в различные функциональные области платформы.

Переключение представлений

Вы можете переключаться между различными представлениями с помощью переключателя представлений, расположенного в левом верхнем углу веб-консоли. Это обеспечивает плавную навигацию между разными функциональными контекстами внутри платформы.

Основные представления

Веб-консоль MWS Container Platform предлагает несколько ключевых представлений, каждое из которых адаптировано под конкретные роли пользователей и задачи.

Представление MWS Container Platform

Представление MWS Container Platform предназначено для разработчиков. В этом представлении разработчики могут управлять ресурсами в рамках своих проектов, такими как:

- Рабочие нагрузки приложений
- Хранилища томов
- Конфигурации
- Привязки сервисов
- Маршруты и сетевые ресурсы

Представление Administrator

Представление Administrator ориентировано на администраторов. В этом представлении администраторы могут:

- Создавать или импортировать кластеры
- Устанавливать и управлять операторами
- Настраивать параметры платформы на уровне всей системы
- Мониторить состояние системы и метрики

- Управлять безопасностью платформы

Представление Project

Представление Project сосредоточено на администрировании на уровне проектов, позволяя пользователям:

- Создавать и настраивать проекты
- Управлять квотами и лимитами ресурсов проектов
- Администрировать участников проекта и привязки ролей
- Настраивать политики на уровне проектов

Страницы обзора

Каждое представление обычно включает страницу обзора, которая предоставляет сводную информацию по соответствующей области. Например, страница обзора Administrator отображает данные по всей платформе, включая:

- Общее количество проектов
- Статус и состояние кластеров
- Метрики использования ресурсов
- Последние события и оповещения
- Индикаторы состояния платформы

Эти страницы обзора предоставляют быстрый доступ к информации о состоянии и производительности вашей контейнеризированной инфраструктуры, помогая пользователям оперативно оценивать состояние системы и выявлять области, требующие внимания.

Доступ к веб-консоли

Веб-консоль MWS Container Platform предоставляет централизованный графический интерфейс для управления контейнеризованными приложениями и инфраструктурой. В этом документе описаны методы доступа к веб-консоли и навигации между её различными представлениями.

Доступ по URL

Веб-консоль доступна по URL, который обычно имеет следующий формат:

`https://<console-route-hostname>`

Свяжитесь с системным администратором, чтобы получить конкретный URL для вашей установки MWS Container Platform.

Аутентификация

При переходе по URL веб-консоли вы увидите страницу входа. Введите свои учетные данные и выберите метод аутентификации, если настроено несколько вариантов.

Платформа поддерживает различные провайдеры аутентификации, которые могут включать:

- Локальную аутентификацию

- LDAP/Active Directory
- OpenID Connect
- OAuth провайдеры

После успешной аутентификации вы будете перенаправлены на страницу Portal.

Навигация по веб-консоли

Страница Portal

Страница Portal служит точкой входа в веб-консоль и отображает карточки для всех доступных представлений. Каждая карточка представляет собой отдельную функциональную область платформы.

Переключение представлений

Чтобы переключиться между представлениями:

1. Найдите переключатель представлений в верхнем левом углу консоли
2. Нажмите на переключатель, чтобы увидеть доступные представления
3. Выберите нужное представление из выпадающего меню

Элементы пользовательского интерфейса

Интерфейс веб-консоли включает несколько ключевых элементов:

- **Главное навигационное меню:** расположено слева, обеспечивает доступ к ресурсам конкретного представления
- **Переключатель представлений:** расположен в верхнем левом углу, позволяет переключаться между представлениями
- **Меню пользователя:** расположено в верхнем правом углу, предоставляет доступ к настройкам пользователя и выходу из системы
- **Меню помощи:** предоставляет доступ к документации и ресурсам поддержки

Инструменты CLI

Обзор

CLI-инструмент **violet** — это интегрированная с платформой утилита, которая позволяет загружать программные пакеты с вашего кастомного портала на платформу. Инструмент поддерживает три типа программных пакетов: Operators, Cluster Plugins и Helm charts.

Поддерживаемые операционные системы

Linux, macOS и Windows. Перечисленные ОС поддерживают архитектуры x86 ARM.

Требования к разрешениям

Необходимо предоставить действующую учетную запись пользователя платформы (имя пользователя и пароль)

Учетная запись должна иметь свойство роли, установленное в **System** должно быть, а имя роли **platform-admin-system**.

Внимание: Если у вашей учетной записи свойство роли установлено в **Custom**, а не **System**, вы не сможете использовать этот инструмент, даже если роль имеет все разрешения **Platform Management**.

Загрузка инструмента

1. Перейдите в представление **Platform Management**.
2. Нажмите **Marketplace/Upload Packages** в левой навигации.
3. Нажмите **Download Package Upload Tool**.
4. Выберите бинарный файл, соответствующий вашей операционной системе и архитектуре.

Установка инструмента

После загрузки инструмента установите его на сервер упаковки или загрузки либо на персональный ПК.

Для Linux или macOS

Для пользователей без root-прав:

```
# Linux x86
sudo mv -f violet_linux_amd64 /usr/local/bin/violet && sudo chmod +x /usr/loc # Linux
ARM
sudo mv -f violet_linux_arm64 /usr/local/bin/violet && sudo chmod +x /usr/loc
# macOS x86
sudo mv -f violet_darwin_amd64 /usr/local/bin/violet && sudo chmod +x /usr/lo #
macOS ARM
sudo mv -f violet_darwin_arm64 /usr/local/bin/violet && sudo chmod +x /usr/lo
```

Для Windows

Скачайте файл и переименуйте его в **violet.exe** или используйте в PowerShell для переименования:

```
# Windows x86
mv -Force violet_windows_amd64.exe violet.exe
```

Запустите инструмент в PowerShell. или используйте PowerShell.

Внимание: Если вы не добавили путь к инструменту в переменные окружения, при выполнении команд необходимо указывать полный путь к инструменту.

Использование инструмента

Предварительный просмотр пакетов

Администраторы платформы могут использовать команду **violet show** для просмотра и подтверждения информации о пакете перед загрузкой.

```
$ violet show topolvm-operator.v2.3.0.tgz
Name: NativeStor Type: bundle
Arch: [linux/amd64]
Version: 2.3.0
```

```
$ violet show topolvm-operator.v2.3.0.tgz --all
```

```
Name: NativeStor Type: bundle Arch: [] Version: 2.3.0
```

```
Artifact: harbor.demo.io/acp/topolvm-operator-bundle:v3.11.0
```

```
Relatelmages: [harbor.demo.io/acp/topolvm-operator:v3.11.0 harbor.demo.io/acp
```

Загрузка Operators

```
violet push opensearch-operator.v3.14.2.tgz --platform-address https://192.16
```

Параметр	Описание
args[0]	Обязательный. Первый параметр после команды violet push . Укажите путь к файлу пакета для загрузки. Для нескольких пакетов поместите их в одну директорию и укажите путь к директории — инструмент автоматически загрузит все пакеты из этой директории.
--platform-address	Обязательный. Адрес доступа к платформе.
--platform-username	Обязательный. Имя пользователя локального пользователя с ролью Platform Administrator role .
--platform-password	Обязательный. Пароль для указанного имени пользователя.
--skip-push	Необязательный. Добавьте этот параметр, если нужно только создать ресурсы, связанные с загрузкой, без отправки образов.
--force	Необязательный. Добавьте этот параметр для принудительного перезаписывания и обновления связанных ресурсов, если образы из пакета уже существуют на платформе.
--target-chartrepo	Необязательный. Добавьте этот параметр, если необходимо загрузить пакет в конкретный catalogsource .
--clusters	Необязательный. Добавьте этот параметр для указания кластеров, в которые нужно загрузить пакет. Несколько имен кластеров разделяйте запятыми, например, global,cluster1,cluster2 . Если параметр не указан, пакет по умолчанию загружается в кластер global .
--debug	Необязательный. Добавьте этот параметр, если необходимо выводить отладочную информацию.

Загрузка Cluster Plugins

```
violet push plugins-clouddedge-v0.3.16-hybrid.tgz --platform-address https://192.168.0.1  
--platform-username <user> --platform-password <password>
```

Параметр	Описание
args[0]	Обязательный. Первый параметр после команды violet push . Укажите путь к файлу пакета для загрузки. Для нескольких

Параметр	Описание
	пакетов поместите их в одну директорию и укажите путь к директории — инструмент автоматически загрузит все пакеты из этой директории.
--platform-address	Обязательный. Адрес доступа к платформе.
--platform-username	Обязательный. Имя пользователя локального пользователя с ролью Platform Administrator role .
--platform-password	Обязательный. Пароль для указанного имени пользователя.
--skip-push	Необязательный. Добавьте этот параметр, если нужно только создать ресурсы, связанные с загрузкой, без отправки образов.
--force	Необязательный. Добавьте этот параметр для принудительного перезаписывания и обновления связанных ресурсов, если образы из пакета уже существуют на платформе.
--debug	Необязательный. Добавьте этот параметр, если необходимо выводить отладочную информацию.

Загрузка Helm Charts

```
violet push chartmuseum-v3.16.1.tgz --platform-address https://192.168.0.1 --platform-username <user> --platform-password <password>
```

Параметр	Описание
args[0]	Обязательный. Первый параметр после команды violet push . Укажите путь к файлу пакета для загрузки. Для нескольких пакетов поместите их в одну директорию и укажите путь к директории — инструмент автоматически загрузит все пакеты из этой директории.
--platform-address	Обязательный. Адрес доступа к платформе.
--platform-username	Обязательный. Имя пользователя локального пользователя с ролью Platform Administrator role .
--platform-password	Обязательный. Пароль для указанного имени пользователя.
--skip-push	Необязательный. Добавьте этот параметр, если нужно только создать ресурсы, связанные с загрузкой, без отправки образов.
--force	Необязательный. Добавьте этот параметр для принудительного перезаписывания и обновления связанных ресурсов, если образы из пакета уже существуют на платформе.
--debug	Необязательный. Добавьте этот параметр, если необходимо выводить отладочную информацию.

Работа с кластерами

Кластер — это базовый набор ресурсов для запуска контейнеризованных приложений, включающий узлы, балансировщики нагрузки, хранилища и другие критически важные компоненты. Он является необходимым условием для успешного запуска контейнеризованных приложений на платформе. При первоначальной установке платформы создаётся стандартный Kubernetes кластер, известный как **global** кластер. В дальнейшем в **global** кластер можно интегрировать несколько кластеров для их единого управления.

Типы кластеров

On-Premises кластер

On-Premises кластер — это Kubernetes кластеры, создаваемые непосредственно платформой. Пользователи предоставляют виртуальные или физические машины, а платформа устанавливает и настраивает Kubernetes кластеры на этих машинах. Такой подход подходит для предприятий с уже имеющимися аппаратными ресурсами, позволяя полностью использовать инфраструктуру.

Managed кластер

Managed кластер — это Kubernetes кластеры, предоставляемые облачными провайдерами, которые интегрируются в платформу для единого управления.

Поддерживаемые методы интеграции включают:

Метод	Описание	Сценарий использования	Ключевые характеристики
Import	Интеграция существующих Kubernetes кластеров	Существующие кластеры с прямым сетевым доступом	• Информация о кластере передаётся в global кластер • global кластер должен иметь сетевой доступ к кластеру
Register	Интеграция кластеров с жёсткими требованиями безопасности	Кластеры с высокими требованиями к безопасности	• На целевом кластере устанавливаются специальные плагины • Обратный прокси создаёт защищённый туннель • Обеспечивается безопасность кластера при

Метод	Описание	Сценарий использования	Ключевые характеристики
			возможности управления
Proxy Create	Создание кластеров через облачных провайдеров	Использование публичных Kubernetes сервисов облака	• Требуются учётные данные облачного провайдера • Платформа создаёт Kubernetes кластеры с использованием предоставленных учётных данных

Поддержка мультиоблаков и гибридных облаков

Эти подходы к управлению кластерами удовлетворяют потребности предприятий в сценариях мультиоблаков и гибридных облаков, поддерживая контейнерную трансформацию на разных этапах:

- Существующее оборудование: создание кластеров, предоставляемых платформой
- Существующие кластеры: импорт или регистрация в платформе
- Эластичные потребности: быстрое создание кластеров в публичном облаке

Особенности реализации и ограничения

Совместимость версий

- Поддерживаемые версии Kubernetes: 1.28, 1.29, 1.30, 1.31
- Как On-Premises, так и Managed кластеры должны обеспечивать совместимость версий
- Несовпадение версий может привести к ограничениям функционала или проблемам совместимости

Сетевые и требования безопасности

- Обеспечить сетевое соединение между **global** и целевыми кластерами
- Реализовать соответствующие политики безопасности сети и файрвола
- Безопасно управлять учётными данными доступа и механизмами аутентификации

Рекомендации по управлению кластерами

1. Предварительная оценка

- Провести тщательный анализ инфраструктуры и рабочих нагрузок

- Определить конкретные требования для каждого кластера
- Разработать комплексную стратегию миграции и интеграции

2. Безопасность и соответствие требованиям

- Внедрить управление доступом на основе ролей (RBAC)
- Использовать сетевые политики для ограничения коммуникаций между кластерами
- Регулярно проводить аудит и обновление конфигураций безопасности
- Обеспечить соответствие отраслевым стандартам и нормативам

3. Мониторинг и наблюдаемость

- Настроить централизованный сбор логов и мониторинг
- Внедрить проактивные механизмы оповещений
- Использовать инструменты наблюдаемости, предоставляемые платформой
- Отслеживать производительность кластера, использование ресурсов и состояние

4. Резервное копирование и аварийное восстановление

- Установить регулярные процедуры резервного копирования
- Создать и протестировать планы аварийного восстановления
- Реализовать стратегии резервного копирования для нескольких кластеров
- Обеспечить минимальное время простоя и потерю данных

5. Непрерывная оптимизация

- Регулярно пересматривать конфигурации кластеров
- Оптимизировать распределение ресурсов
- Обновлять до последних поддерживаемых версий Kubernetes
- Использовать возможности платформы для автоматических обновлений и масштабирования

Создание кластера On-Premise

Требования

Требования к узлам

1. Если вы скачали пакет установки для одной архитектуры с Download Installation Package, убедитесь, что архитектура ваших узлов совпадает с архитектурой пакета. В противном случае узлы не запустятся из-за отсутствия образов, специфичных для архитектуры.
2. Проверьте, что операционная система и ядро узлов поддерживаются. Подробности смотрите в разделе Supported OS and Kernels.
3. Выполните проверки доступности узлов. Для конкретных пунктов проверки обратитесь к разделу Node Preprocessing > Node Checks.

4. Если IP-адреса узлов недоступны напрямую по SSH, предоставьте для узлов SOCKS5 прокси. Кластер **global** будет обращаться к узлам через этот прокси-сервис.

Балансировка нагрузки

Для производственных сред необходим балансировщик нагрузки для узлов управляющей плоскости кластера, чтобы обеспечить высокую доступность.

Можно использовать собственный аппаратный балансировщик нагрузки или включить **Self-built VIP**, который обеспечивает программную балансировку нагрузки с помощью **haproxy** + **keepalived**.

Рекомендуется использовать аппаратный балансировщик нагрузки, поскольку:

- **Лучшая производительность:** Аппаратная балансировка работает эффективнее программной.
- **Меньшая сложность:** Если вы не знакомы с **keepalived**, неправильная настройка может привести к недоступности кластера, что вызовет длительный поиск и устранение неисправностей и серьезно повлияет на надежность кластера.

При использовании собственного аппаратного балансировщика нагрузки в качестве параметра **IP Address / Domain** можно указать **VIP** балансировщика. Если у вас есть доменное имя, разрешающееся в **VIP** балансировщика, можно использовать это доменное имя.

Важно:

- Балансировщик нагрузки должен корректно перенаправлять трафик на порты **6443**, **11780** и **11781** на всех узлах управляющей плоскости кластера.
- Если в кластере только один узел управляющей плоскости и вы используете IP этого узла как **IP Address / Domain**, то впоследствии масштабировать кластер с одного узла до высокодоступного многозвенного нельзя. Поэтому рекомендуется использовать балансировщик нагрузки даже для однозвенных кластеров.

При включении **Self-built VIP** необходимо подготовить:

1. Доступный **VRID**
2. Хост-сеть, поддерживающую протокол **VRRP**
3. Все узлы управляющей плоскости и **VIP** должны находиться в одной подсети, при этом **VIP** должен отличаться от IP-адресов узлов.

Подключение кластера **global** и рабочих кластеров

Платформа требует взаимного доступа между кластером **global** и рабочими кластерами. Если они находятся в разных сетях, необходимо:

Обеспечить **External Access** для рабочего кластера, чтобы кластер **global** мог получить к нему доступ. Требования к сети: **global** должен иметь доступ к портам **6443**, **11780** и **11781** на всех узлах управляющей плоскости.

Добавить дополнительный адрес в **global**, доступный рабочему кластеру. При создании рабочего кластера добавьте этот адрес в аннотации кластера с ключом **cpaas.io/platform-url** и значением, равным публичному адресу доступа к **global**.

Реестр образов

Образы кластера поддерживают варианты: встроенный в платформу, приватный репозиторий и публичный репозиторий:

- **Platform Built-in:** Использует реестр образов, предоставляемый кластером global. Если кластер не может получить доступ к global, смотрите Add External Address for Built-in Registry.
- **Private Repository:** Использует ваш собственный реестр образов. Для подробностей по загрузке необходимых образов в ваш реестр обратитесь в техническую поддержку.
- **Public Repository:** Использует публичный реестр образов платформы. Перед использованием выполните обновление учётных данных публичного репозитория.

Сетевое взаимодействие контейнеров

Если вы планируете использовать Kube-OVN Underlay для вашего кластера, обратитесь к разделу Подготовка физической сети Kube-OVN Underlay.

Шаги

1. Перейдите в представление **Platform Management** и в левом навигационном меню выберите **Clusters/Clusters**.
2. Нажмите **Create Cluster**.
3. Настройте следующие разделы согласно инструкциям ниже: Basic Info, Container Network, Node Settings:
 - Базовая информация:

Параметр	Описание
Kubernetes Version	Все доступные версии тщательно протестированы на стабильность и совместимость. Рекомендация: Выбирайте последнюю версию для оптимальных возможностей и поддержки.
Container Runtime	По умолчанию используется containerd. Если вы предпочитаете Docker в качестве runtime, обратитесь к разделу Выбор контейнерного рантайма.
Cluster Network Protocol	Поддерживаются три режима: IPv4 single stack, IPv6 single stack, IPv4/IPv6 dual stack. Внимание: При выборе dual stack убедитесь, что на всех узлах корректно настроены IPv6-адреса; после установки изменить сетевой протокол нельзя.
Cluster Endpoint	IP Address / Domain: Введите заранее подготовленное доменное имя или VIP, если домена нет. Self-Built VIP: По умолчанию отключено. Включайте только если не предоставлен LoadBalancer. При включении установщик автоматически развернет

Параметр	Описание
	keepalived для поддержки программной балансировки нагрузки. External Access: Введите внешний адрес доступа, подготовленный для кластера, если он находится в другой сетевой среде, нежели global.

- **Container Network:**

- **Kube-OVN:**

Корпоративная система оркестрации сетей контейнеров Cloud Native Kubernetes, разработанная вендором. Она переносит зрелые сетевые возможности из области OpenStack в Kubernetes, поддерживает управление сетями между облаками, традиционную сетевую архитектуру и инфраструктурное межсоединение, сценарии развертывания на периферийных кластерах, значительно повышая безопасность, эффективность управления и производительность сетей контейнеров Kubernetes.

Параметр	Описание
Subnet	Также известен как Cluster CIDR, представляет подсеть по умолчанию. После создания кластера можно добавлять дополнительные подсети.
Transmit Mode	Overlay: Виртуальная сеть, абстрагированная над инфраструктурой, не потребляющая физические сетевые ресурсы. При создании Overlay подсети по умолчанию все Overlay подсети кластера используют одинаковую конфигурацию cluster NIC и node NIC. Underlay: Этот режим передачи опирается на физические сетевые устройства. Он может напрямую выделять физические сетевые адреса для Pod, обеспечивая лучшую производительность и связь с физической сетью. Узлы в Underlay подсети должны иметь несколько NIC, а NIC, используемый для мостовой сети, должен использоваться исключительно для Underlay и не должен нести другой трафик, например SSH. При создании Underlay подсети по умолчанию cluster NIC фактически является NIC по умолчанию для мостовой сети, а node NIC — это конфигурация NIC узла в мостовой сети. Default Gateway: Адрес шлюза физической сети, который является шлюзом для сегмента Cluster CIDR (должен находиться в диапазоне адресов Cluster CIDR). VLAN ID: Идентификатор виртуальной локальной сети (номер VLAN), например, 0.

Параметр	Описание
	Reserved IPs: Укажите зарезервированные IP-адреса, которые не будут автоматически выделяться, например, IP-адреса в подсети, уже используемые другими устройствами.
Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети по умолчанию.
Join CIDR	В режиме передачи Overlay это диапазон IP-адресов для связи между узлами и pod. Не должен пересекаться с подсетью по умолчанию или Service CIDR.

○ **Calico:**

Calico — решение сетевого уровня 3, обеспечивающее безопасные сетевые соединения для контейнеров.

Параметр	Описание
Default Subnet	Также известен как Cluster CIDR, представляет подсеть по умолчанию. После создания кластера можно добавлять дополнительные подсети.
Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети по умолчанию.

○ **Flannel:**

Flannel обеспечивает плоскую сетевую среду для всех контейнеров в кластере, предоставляя контейнерам, созданным на разных узлах, уникальный виртуальный IP-адрес по всему кластеру. Подсеть pod равномерно делится между узлами кластера согласно маске, и pod на каждом узле получают IP-адреса из сегмента, выделенного этому узлу. Это повышает эффективность связи между контейнерами без необходимости учитывать вопросы трансляции IP.

Параметр	Описание
Cluster CIDR	Диапазон IP-адресов, используемый pod, создаваемыми при запуске кластера. Поддерживается настройка максимального количества IP-адресов, которые могут быть выделены pod на каждом узле в текущей контейнерной сети. Внимание: Платформа автоматически рассчитает максимальное количество узлов, которое может вместить кластер, исходя из этой настройки, и отобразит это в подсказке под полем ввода. После создания кластера сетевую конфигурацию изменить нельзя, поэтому планируйте сеть внимательно.

Параметр	Описание
Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети контейнеров.

○ **Custom:**

Если необходимо установить другие сетевые плагины, выберите режим **Custom**. Вы сможете вручную установить сетевые плагины после успешного создания кластера.

Параметр	Описание
Cluster CIDR	Диапазон IP-адресов, используемый pod, создаваемыми при запуске кластера.
Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети контейнеров.

● **Node Settings:**

Параметр	Описание
Network Interface Card	Имя сетевого интерфейса хоста, используемого сетевым плагином кластера. Внимание: При выборе режима передачи Underlay для подсети по умолчанию Kube-OVN необходимо указать имя сетевого интерфейса, который будет использоваться как NIC по умолчанию для мостовой сети. По умолчанию мониторинг трафика сетевых интерфейсов платформы распознает трафик на интерфейсах с именами, начинающимися на eth. en. wl. ww. Если вы используете интерфейсы с другими именами, после добавления кластера обратитесь к разделу Сбор сетевых данных с сетевых интерфейсов с пользовательскими именами для изменения соответствующих ресурсов и обеспечения корректного мониторинга трафика.
Node Name	Можно выбрать использование IP-адреса узла или имени хоста в качестве имени узла на платформе. Внимание: При выборе имени хоста убедитесь, что имена хостов узлов, добавляемых в кластер, уникальны.
Nodes	Добавьте узлы в кластер или Восстановите из черновика временно сохранённую информацию об узлах. Подробные описания параметров добавления узлов приведены ниже.
Monitoring Type	Поддерживаются Prometheus и VictoriaMetrics. При выборе VictoriaMetrics необходимо указать Deploy Type :

Параметр	Описание
	• Deploy VictoriaMetrics: Разворачивает все компоненты, включая VMStorage, VMAAlert, VMAgent и др • Deploy VictoriaMetrics Agent: Разворачивает только компонент сбора логов VMAgent. При таком способе необходимо связать с уже развернутым экземпляром VictoriaMetrics на другом кластере платформы для предоставления мониторинга.
Monitoring Nodes	Выберите узлы для развертывания компонентов мониторинга кластера. Поддерживается выбор вычислительных узлов и узлов управляющей плоскости, разрешающих развертывание приложений. Чтобы не влиять на производительность кластера, рекомендуется отдавать предпочтение вычислительным узлам. После успешного создания кластера компоненты мониторинга с типом хранения Local Volume будут развернуты на выбранных узлах.

○ **Параметры добавления узлов:**

Параметр	Описание
Type	Control Plane Node: Отвечает за запуск компонентов kube-apiserver, kube-scheduler, kube-controller-manager, etcd, контейнерной сети и некоторых компонентов управления платформой в кластере. При включенной опции Application Deployable узлы управляющей плоскости могут использоваться как вычислительные. Worker Node: Отвечает за размещение бизнес-подов, работающих в кластере.
IPv4 Address	IPv4-адрес узла. Для кластеров, созданных в режиме внутренней сети, укажите приватный IP узла.
IPv6 Address	Действительно при включенном dual stack IPv4/IPv6. IPv6-адрес узла.
Application Deployable	Действительно при типе узла Control Plane Node . Разрешить ли развертывание бизнес-приложений на этом узле управляющей плоскости, то есть планировать бизнес-поды на этот узел.
Display Name	Отображаемое имя узла.
SSH Connection IP	IP-адрес, по которому можно подключиться к узлу через SSH. Если вы можете войти на узел с помощью ssh <username>@<IPv4-адрес узла>, этот параметр не обязателен; в противном случае укажите публичный IP или

Параметр	Описание
	внешний NAT IP узла, чтобы кластер global и прокси могли подключаться к узлу по этому IP.
Network Interface Card	Укажите имя сетевого интерфейса, используемого узлом. Приоритет эффективности конфигурации сетевого интерфейса следующий (слева направо, по убыванию): Kube-OVN Underlay: Node NIC > Cluster NIC Kube-OVN Overlay: Node NIC > Cluster NIC > NIC, соответствующий default route узла Calico: Cluster NIC > NIC, соответствующий default route узла Flannel: Cluster NIC > NIC, соответствующий default route узла
Associated Bridge Network	Внимание: При создании кластера конфигурация мостовой сети не поддерживается; этот параметр доступен только при добавлении узлов в кластер, в котором уже созданы Underlay подсети. Выберите существующую Bridge Network. Если не хотите использовать NIC по умолчанию мостовой сети, можно отдельно настроить node NIC.
SSH Port	Номер порта SSH, например, 22.
SSH Username	Имя пользователя SSH с правами root, например, root
Proxy	Использовать ли прокси для доступа к SSH-порту узла. Если кластер global не может напрямую подключиться к добавляемому узлу по SSH (например, кластер global и рабочий кластер находятся в разных подсетях; IP узла — внутренний, недоступный для global), включите этот переключатель и настройте параметры прокси. После настройки прокси доступ к узлу и развертывание будут осуществляться через прокси. Внимание: В настоящее время поддерживается только SOCKS5 прокси. Access URL: Адрес прокси-сервера, например, 192.168.1.1:1080. Username: Имя пользователя для доступа к прокси-серверу. Password: Пароль для доступа к прокси-серверу.
SSH Authentication	Метод аутентификации и соответствующая информация для входа на добавляемый узел. Варианты: • Password: Требуется имя пользователя с правами root и соответствующий SSH пароль. • Key: Требуется приватный ключ с правами root и пароль к приватному ключу.

Параметр	Описание
Save Draft	Сохраняет текущие настройки в диалоге как черновик и закрывает окно Add Node. Не покидая страницу Create Cluster, можно выбрать Restore from draft для открытия диалога Add Node и восстановления сохранённых данных. Внимание: Восстанавливаются данные самого последнего сохранённого черновика.

Действия после создания

Просмотр прогресса создания

На странице списка кластеров можно просмотреть список созданных кластеров. Для кластеров со статусом **Creating** доступен просмотр прогресса выполнения:

1. Нажмите на значок **View Execution Progress** справа от статуса кластера.
2. В появившемся диалоговом окне можно просмотреть прогресс выполнения кластера (`status.conditions`).

Внимание: Если какой-либо тип находится в процессе или в состоянии ошибки с причиной, наведите курсор на причину (отображается синим текстом), чтобы увидеть подробную информацию о причине (`status.conditions.reason`).

Сеть

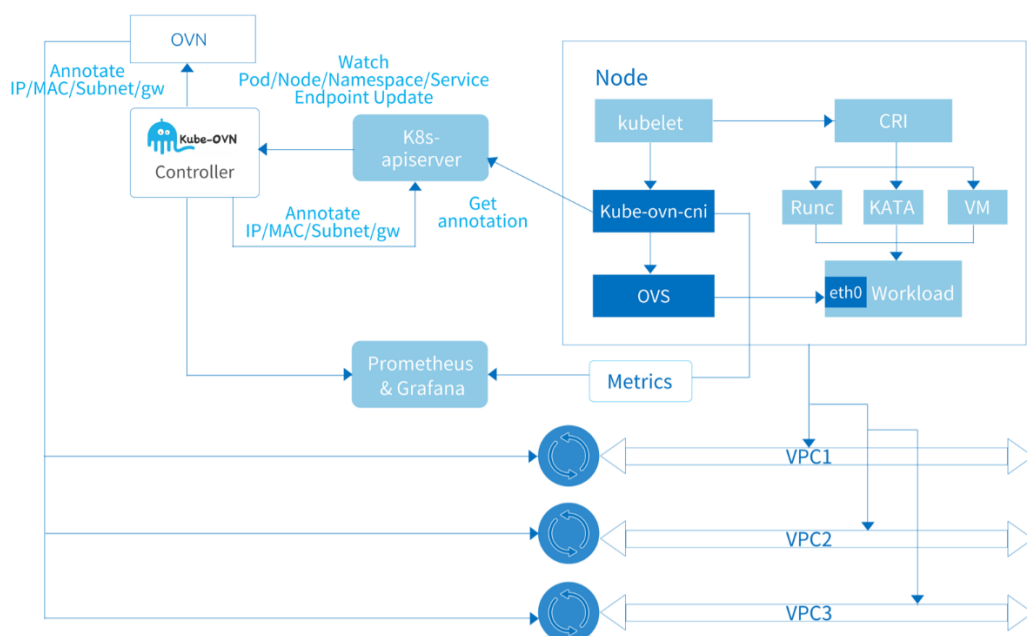
Kube-OVN

Kube-OVN служит мостом между Kubernetes и OVN, объединяя проверенные SDN с Cloud Native. Это означает, что Kube-OVN не только реализует сетевые спецификации в Kubernetes, такие как CNI, Service и NetworkPolicy, но и приносит множество возможностей SDN в облачную нативную среду, таких как логические коммутаторы, логические маршрутизаторы, VPC, шлюзы, QoS, ACL и зеркалирование трафика.

Kube-OVN также поддерживает хорошую открытость для интеграции с множеством технологических решений, таких как Cilium, Submariner, Prometheus, KubeVirt и др.

Компоненты Kube-OVN можно условно разделить на три категории.

- Компоненты upstream OVN/OVS.
- Основной контроллер и агент.
- Инструменты мониторинга, эксплуатации и расширения.



Компоненты upstream OVN/OVS

Данный тип компонентов происходит из сообщества OVN/OVS с конкретными модификациями для сценариев использования Kube-OVN. OVN/OVS — это зрелая SDN-система для управления виртуальными машинами и контейнерами, и мы настоятельно рекомендуем пользователям, заинтересованным в реализации Kube-OVN, сначала ознакомиться с `ovn-architecture`, чтобы понять, что такое OVN и как с ним интегрироваться. Kube-OVN использует northbound-интерфейс OVN для создания и координации виртуальных сетей и отображения сетевых концепций в Kubernetes.

Все компоненты, связанные с OVN/OVS, упакованы в образы и готовы к запуску в Kubernetes.

ovn-central

Deployment **ovn-central** запускает компоненты контрольной плоскости OVN, включая **ovn-nb**, **ovn-sb** и **ovn-northd**.

- **ovn-nb**: Сохраняет конфигурацию виртуальной сети и предоставляет API для управления виртуальной сетью. **kube-ovn-controller** в основном взаимодействует с **ovn-nb** для настройки виртуальной сети.
- **ovn-sb**: Хранит таблицу логических потоков, сгенерированную из логической сети **ovn-nb**, а также фактическое состояние физической сети каждого узла.
- **ovn-northd**: преобразует виртуальную сеть из **ovn-nb** в таблицу логических потоков в **ovn-sb**.

Несколько экземпляров **ovn-central** синхронизируют данные через протокол **Raft** для обеспечения высокой доступности.

ovs-ovn

ovs-ovn запускается как DaemonSet на каждом узле, внутри Pod работают **openvswitch**, **ovsdb** и **ovn-controller**. Эти компоненты выступают агентами для **ovn-central**, преобразуя таблицы логических потоков в реальные сетевые конфигурации.

Основной контроллер и агент

Эта часть является ядром **Kube-OVN**, служит мостом между OVN и Kubernetes, связывая две системы и переводя сетевые концепции между ними. Большинство основных функций реализованы в этих компонентах.

kube-ovn-controller

Этот компонент выполняет трансляцию всех ресурсов внутри Kubernetes в ресурсы OVN и выступает в роли контрольной плоскости всей системы Kube-OVN. **kube-ovn-controller** слушает события по всем ресурсам, связанным с сетевой функциональностью, и обновляет логическую сеть внутри OVN на основе изменений ресурсов. Основные ресурсы, за которыми ведется наблюдение, включают: Pod, Service, Endpoint, Node, NetworkPolicy, VPC, Subnet, Vlan, ProviderNetwork.

На примере события создания Pod: **kube-ovn-controller** слушает событие создания Pod, выделяет адрес с помощью встроенной функции IPAM в памяти, и вызывает **ovn-central** для создания логических портов, статических маршрутов и возможных правил ACL. Далее **kube-ovn-controller** записывает назначенный адрес и информацию о подсети, такую как CIDR, шлюз, маршрут и т.д., в аннотацию Pod. Эту аннотацию затем читает **kube-ovn-cni** и использует для настройки локальной сети.

kube-ovn-cni

Этот компонент запускается на каждом узле как DaemonSet, реализует интерфейс CNI и управляет локальным OVS для настройки локальной сети.

DemonSet копирует бинарный файл **kube-ovn** на каждую машину как инструмент взаимодействия между **kubelet** и **kube-ovn-cni**. Этот бинарный файл отправляет

соответствующий CNI-запрос в **kube-ovn-cni** для дальнейших операций. По умолчанию бинарный файл копируется в каталог **/opt/cni/bin**.

kube-ovn-cni настраивает конкретную сеть для выполнения соответствующих операций с трафиком, основные задачи включают:

1. Настройка **ovn-controller** и **vswitchd**.
2. Обработка запросов CNI Add/Del:
 1. Создание или удаление пары **veth** и привязка или отвязка к портам OVS.
 2. Настройка портов OVS.
 3. Обновление правил **iptables/ipset/route** на хосте.
3. Динамическое обновление QoS сети.
4. Создание и настройка NIC **ovn0** для соединения сети контейнеров и сети хоста.
5. Настройка NIC хоста для реализации Vlan/Underlay/EIP.
6. Динамическая настройка межкластерных шлюзов.

Инструменты мониторинга, эксплуатации и расширения

Эти компоненты предоставляют средства мониторинга, диагностики, инструменты эксплуатации и внешние интерфейсы для расширения основных сетевых возможностей **Kube-OVN** и упрощения повседневных операций и обслуживания.

kube-ovn-speaker

Этот компонент — **DaemonSet**, работающий на узлах с определенной меткой, который публикует маршруты во внешний мир, позволяя внешнему доступу к контейнерам напрямую через IP Pod.

kube-ovn-pinger

Этот компонент — **DaemonSet**, работающий на каждом узле, собирает информацию о состоянии OVS, качестве сети узла, задержках в сети и т.д.

kube-ovn-monitor

Этот компонент собирает информацию о состоянии OVN и метрики мониторинга.

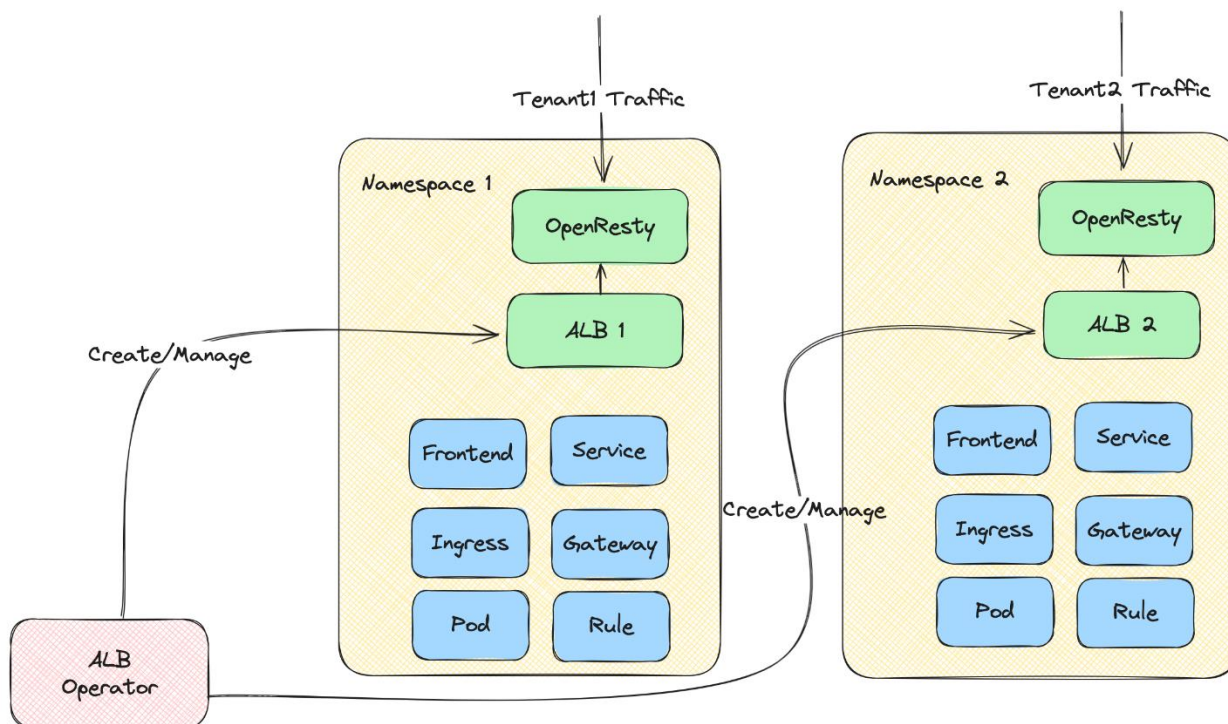
kubectl-ko

Этот компонент — плагин **kubectl**, который позволяет быстро выполнять распространённые операции.

ALB

ALB (Another Load Balancer) — это Kubernetes Gateway на базе OpenResty с многолетним опытом эксплуатации от MWS.

Основные компоненты



- **ALB Operator:** оператор, управляющий жизненным циклом экземпляров ALB. Он отвечает за наблюдение за CR ALB и создание, обновление экземпляров ALB для разных арендаторов.
- **ALB Instance:** экземпляр ALB включает OpenResty, который выступает в роли data plane, и Go контроллер как control plane. Go контроллер отслеживает различные CR (Ingress, Gateway, Rule и др.) и преобразует их в ALB-специфичные DSL правила. OpenResty затем использует эти DSL правила для сопоставления и обработки входящих запросов.

Быстрый старт

Развертывание ALB Operator

1. Создайте кластер.

```
helm repo add alb https://alauda.github.io/alb/;helm repo update;helm search  
repo|grep alb  
helm install alb-operator alb/alauda-alb2
```

Развертывание экземпляра ALB

```
cat <<EOF | kubectl apply -f -  
apiVersion: crd.alauda.io/v2beta1  
kind: ALB2  
metadata:  
  name: alb-demo  
  namespace: kube-system  
spec:
```

```
address: "172.20.0.5" # IP-адрес узла, на котором развернут alb
type: "nginx"
config:
  networkMode: host
  loadbalancerName: alb-demo
  projects:
    - ALL_ALL
  replicas: 1
EOF
```

Запуск демонстрационного приложения

```
cat <<EOF | kubectl apply -f -
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
  labels:
    k8s-app: hello-world
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: hello-world
  template:
    metadata:
      labels:
        k8s-app: hello-world
    spec:
      terminationGracePeriodSeconds: 60
      containers:
        - name: hello-world
          image: docker.io/crccheck/hello-world:latest
          imagePullPolicy: IfNotPresent
---
apiVersion: v1
kind: Service
metadata:
  name: hello-world
  labels:
    k8s-app: hello-world
spec:
```

```
ports:
- name: http
  port: 80
  targetPort: 8000
selector:
  k8s-app: hello-world
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: hello-world
spec:
  rules:
  - http:
    paths:
    - path: /
      pathType: Prefix
      backend:
        service:
          name: hello-world
          port:
            number: 80
EOF
```

Теперь вы можете получить доступ к приложению через **curl http://{ip}**

Общие понятия ALB

Auth

Auth — это механизм, выполняющий аутентификацию до того, как запрос достигнет фактического сервиса. Он позволяет обрабатывать аутентификацию на уровне ALB единообразно, без необходимости реализовывать логику аутентификации в каждом бэкенд-сервисе.

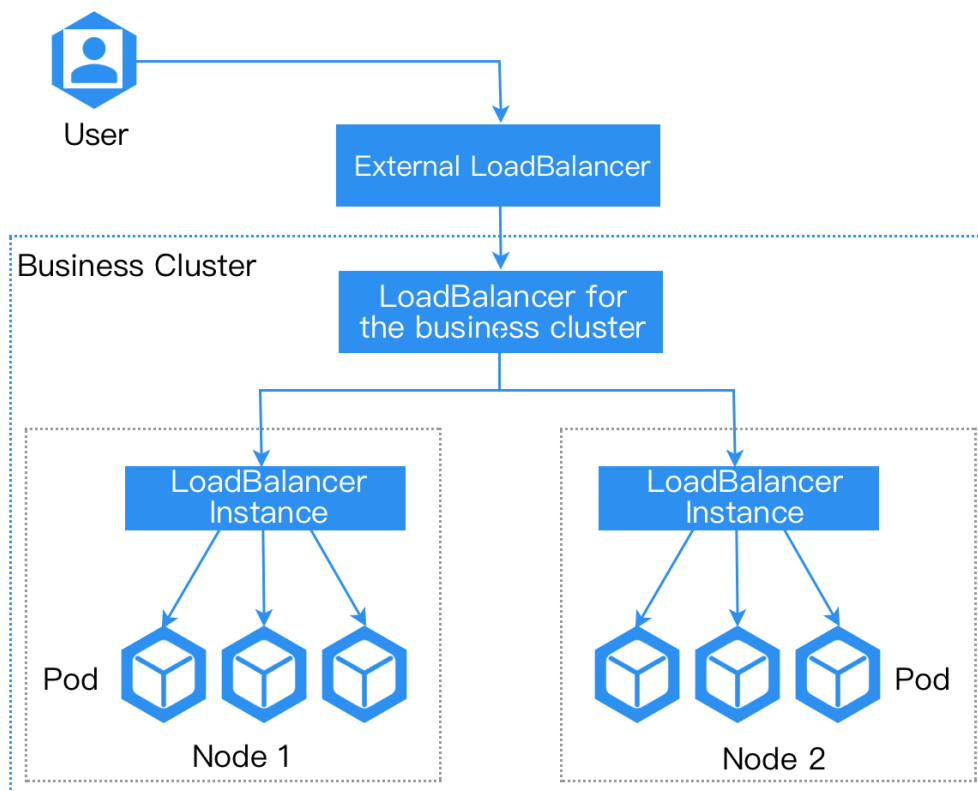
Network Mode

Экземпляр ALB может быть развернут в двух режимах: host network mode и container network mode.

Host Network Mode

Прямое использование сетевого стека узла, совместно используя IP-адрес и порт с узлом. В этом режиме экземпляр балансировщика нагрузки напрямую привязывается к порту узла, без проброса портов или иной контейнерной сетевой инкапсуляции.

Внимание: Чтобы избежать конфликтов портов, на одном узле разрешается развертывать только один экземпляр ALB.



В режиме host-network экземпляр ALB по умолчанию слушает все сетевые интерфейсы узла.

Преимущества:

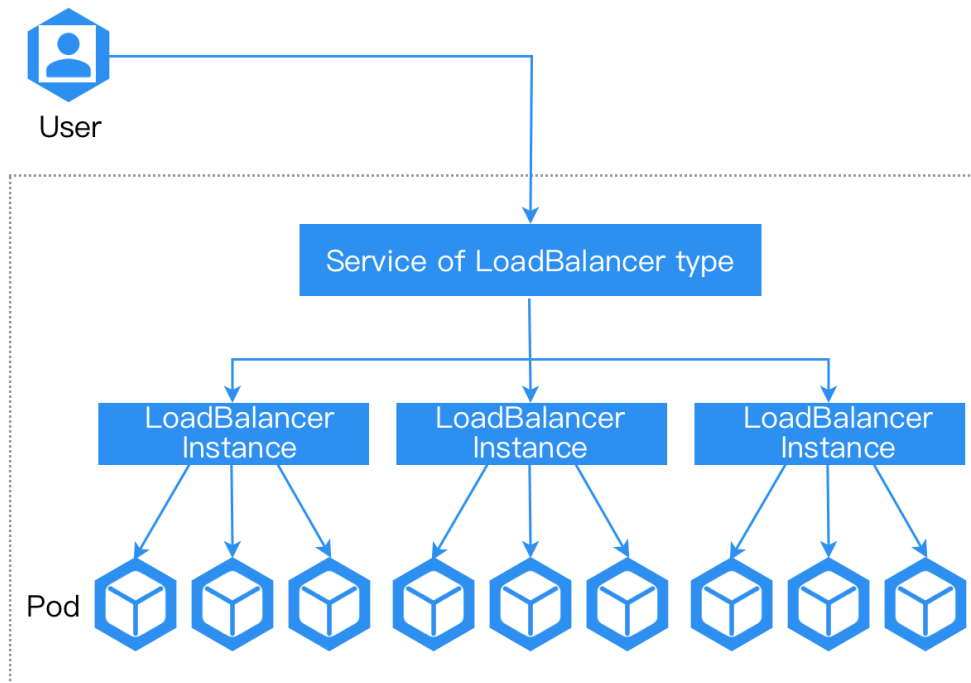
1. Максимальная сетевая производительность.
2. Доступ по IP-адресу узла.

Недостатки:

1. На одном узле разрешён только один экземпляр ALB.
2. Возможны конфликты портов с другими процессами.

Container Network Mode

В отличие от **host network mode**, **container network mode** разворачивает ALB с использованием контейнерной сети.



Преимущества:

1. Поддержка нескольких экземпляров ALB на одном узле.
2. ALB интегрирован с MetalLB, который может предоставлять VIP для ALB.
3. Порты не конфликтуют с другими процессами.

Недостатки:

1. Немного более низкая производительность.
2. Доступ к ALB осуществляется через сервис LoadBalancer.

Frontend

Определён ресурс frontend (сокращённо ft), который используется для объявления всех портов, на которых должен слушать ALB.

Каждый frontend соответствует порту прослушивания на балансировщике нагрузки (LB). Frontend связывается с ALB через метки.

```

apiVersion: crd.alauda.io/v1
kind: Frontend
metadata:
  labels:
    alb2.cpaas.io/name: alb-demo (1)
    name: alb-demo-00080 (2)
    namespace: cpaas-system
spec:
  backendProtocol: "http"
  certificate_name: "" (3)
  port: 80
  protocol: http (4)
  
```

serviceGroup: (5)

services:

- name: hello-world
namespace: default
port: 80
weight: 100 (6)

(1) Обязательно, указывает экземпляр ALB, которому принадлежит этот Frontend.

(2) Формат \$alb_name-\$port.

(3) Формат \$secret_ns/\$secret_name.

(4) Протокол самого Frontend.

- http|https|grpc|grpcs для I7 прокси.
- tcp|udp для I4 прокси.

(5) Для I4 прокси **serviceGroup** обязателен. Для I7 прокси **serviceGroup** является опциональным. При поступлении запроса ALB сначала пытается сопоставить его с правилами, связанными с этим Frontend. Если запрос не соответствует ни одному правилу, ALB перенаправляет его на дефолтный serviceGroup, указанный в конфигурации Frontend.

(6) Конфигурация **weight** применяется для алгоритмов планирования Round Robin и Weighted Round Robin.

Внимание: ALB слушает ingress и автоматически создаёт **Frontend** или **Rule**. Поле **source** определяется следующим образом:

1. **spec.source.type** в настоящее время поддерживает только **ingress**.
2. **spec.source.name** — имя ingress.
3. **spec.source.namespace** — namespace ingress.

Rules

Определён ресурс rule, который описывает, как экземпляр alb должен обрабатывать запросы 7-го уровня.

С помощью Rule можно настроить сложные шаблоны сопоставления и распределения трафика. При поступлении трафика он сопоставляется с внутренними правилами и выполняется соответствующая переадресация, а также предоставляются дополнительные функции, такие как cors, url rewrite и др.

```
apiVersion: crd.alauda.io/v1
kind: Rule
metadata:
  labels:
    alb2.cpaas.io/frontend: alb-demo-00080 (1)
    alb2.cpaas.io/name: alb-demo (2)
name: alb-demo-00080-test
```

```
namespace: kube-system
spec:
  backendProtocol: "" (3)
  certificate_name: "" (4)
  dslx:
    - type: METHOD
      values:
        - - EQ
          - POST
    - type: URL
      values:
        - - STARTS_WITH
          - /app-a
        - - STARTS_WITH
          - /app-b
    - type: PARAM
      key: group
      values:
        - - EQ
          - vip
    - type: HOST
      values:
        - - ENDS_WITH
          - .app.com
    - type: HEADER
      key: LOCATION
      values:
        - - IN
          - east-1
          - east-2
    - type: COOKIE
      key: uid
      values:
        - - EXIST
    - type: SRC_IP
      values:
        - - RANGE
          - "1.1.1.1"
          - "1.1.1.100"
  enableCORS: false
```

```
priority: 4 (5)
serviceGroup: (6)
services:
  - name: hello-world
    namespace: default
    port: 80
    weight: 100
```

- (1) Обязательно, указывает Frontend, которому принадлежит это правило.
- (2) Обязательно, указывает ALB, которому принадлежит это правило.
- (3) Аналогично Frontend.
- (4) Аналогично Frontend.
- (5) Чем меньше число, тем выше приоритет.
- (6) Аналогично Frontend.

dslx

dslx — это предметно-ориентированный язык, используемый для описания критериев сопоставления.

Например, следующее правило сопоставляет запрос, который удовлетворяет всем перечисленным условиям:

- url начинается с /app-a или /app-b
- метод POST
- параметр url с именем group равен vip
- хост соответствует *.app.com
- заголовок LOCATION равен east-1 или east-2
- присутствует cookie с именем uid
- IP-адрес источника находится в диапазоне 1.1.1.1-1.1.1.100

dslx:

```
- type: METHOD
values:
  - EQ
  - POST
- type: URL
values:
  - STARTS_WITH
  - /app-a
  - STARTS_WITH
  - /app-b
- type: PARAM
key: group
```

```

values:
  - - EQ
    - vip
- type: HOST
  values:
    - - ENDS_WITH
      - .app.com
- type: HEADER
  key: LOCATION
  values:
    - - IN
      - east-1
      - east-2
- type: COOKIE
  key: uid
  values:
    - - EXIST
- type: SRC_IP
  values:
    - - RANGE
      - "1.1.1.1"
      - "1.1.1.100"

```

Изоляция проектов

Для rule по умолчанию действует изоляция по проектам: каждый пользователь видит только правила своего проекта.

Режим проекта

ALB может использоваться несколькими проектами, которые могут управлять этим ALB. Все порты ALB видны этим проектам.

Режим порт-проекта

Порт ALB может принадлежать разным проектам. Такой режим называется Port Project Mode. Администратор задаёт диапазон портов для каждого проекта. Пользователи проекта могут создавать порты только в пределах этого диапазона и видеть только эти порты.

ALB Leader

Из нескольких экземпляров ALB выбирается один лидер. Лидер отвечает за:

1. Преобразование Ingress в Rules. Для каждого пути в Ingress создаётся Rule.
2. Создание необходимых FT для Ingress. Например, если у Ingress есть сертификат, создаётся FT 443 (HTTPS), если нет — FT 80 (HTTP).

Project

С точки зрения ALB, Project — это набор namespace.

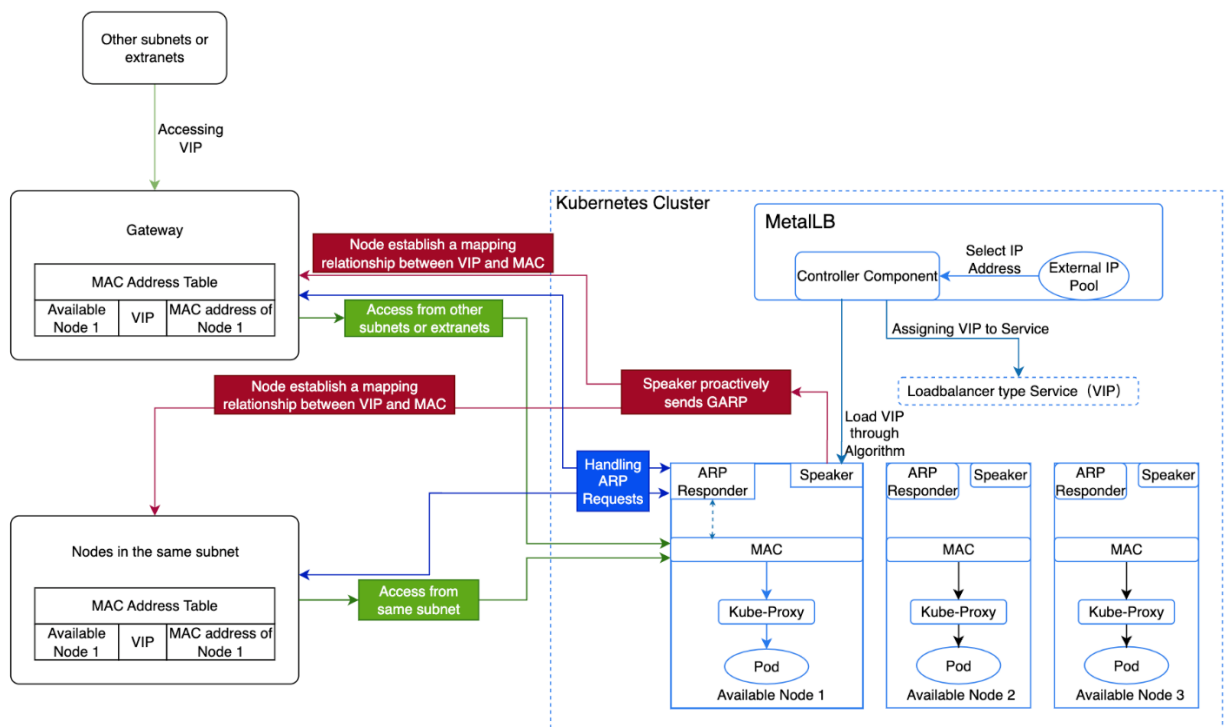
В ALB можно настроить один или несколько Projects. При преобразовании Ingress в Rules ALB Leader игнорирует Ingress в namespace, не принадлежащих проекту.

MetalLB

Терминология

Термин	Описание
VIP	Виртуальный IP-адрес (VIP) — это IP-адрес, назначаемый MetalLB для внутренней маршрутизации типа LoadBalancer, обеспечивающий единый точку доступа для внешнего трафика к сервисам внутри кластера.
ARP	Протокол разрешения адресов (ARP) используется для сопоставления IP-адресов сетевого уровня с MAC-адресами канального уровня.
GARP	Gratuitous ARP (GARP) — это специальный ARP-запрос, используемый для информирования других узлов в сети о привязке IP-адреса к MAC-адресу. В отличие от обычных ARP-запросов, GARP не ожидает ответов, а активно рассылает информацию по сети.
ARP Responder	Компонент MetalLB, отвечающий на ARP-запросы, сопоставляя VIP с MAC-адресом узла. Когда узлу необходимо связаться с VIP, он отправляет ARP-запросы для получения MAC-адреса, соответствующего VIP. Каждый доступный узел имеет ARP Responder, который отвечает на эти запросы, сопоставляя VIP с MAC-адресом узла.
Controller	Компонент MetalLB, который динамически выделяет VIP из пула внешних адресов для внутренней маршрутизации типа LoadBalancer. Controller отслеживает события создания и удаления внутренних маршрутов в кластере для выделения или освобождения VIP по мере необходимости.
Speaker	Компонент MetalLB, который на основе политик или алгоритмов определяет, должны ли узлы размещать VIP и отправлять GARP. Он обеспечивает определённый уровень баланса между узлами, и при недоступности одного узла другие узлы могут взять на себя VIP и отправить GARP, обеспечивая тем самым высокую доступность.

Принципы высокой доступности в MetalLB



По умолчанию платформа использует ARP-режим MetalLB, а конкретный процесс реализации и принципы следующие:

- Компонент Controller MetalLB выбирает IP-адрес из пула внешних адресов и выделяет его для внутренней маршрутизации типа LoadBalancer в качестве VIP.
- MetalLB выбирает доступный узел для размещения VIP на основе алгоритма, который затем перенаправляет трафик.
- Компонент Speaker на этом узле активно отправляет GARP, устанавливая связь между VIP и MAC-адресом на всех узлах.
 - Узлы в одной подсети, узнав о сопоставлении VIP с MAC-адресом доступного узла, будут напрямую взаимодействовать с этим узлом при обращении к VIP.
 - Узлы в разных подсетях сначала направляют трафик на шлюз своей подсети, который затем перенаправляет трафик на узел, размещающий VIP.
- При сбое этого узла MetalLB выбирает другой доступный узел для размещения VIP, обеспечивая тем самым высокую доступность.
- Попав на узел, Kube-Proxy перенаправляет трафик соответствующему Pod.

Алгоритм MetalLB для выбора узлов-хозяев VIP

MetalLB хеширует все доступные узлы, соответствующие пулу внешних адресов, вместе с VIP и сортирует их по определённому алгоритму, выбирая первым доступным узлом для размещения VIP.

Пулы внешних адресов и количество узлов

Создайте пул внешних адресов и добавьте доступные узлы. Все доступные узлы поддерживают **резервные** отношения, то есть только узел, размещающий VIP, может перенаправлять трафик, и он должен обрабатывать весь трафик для VIP из пула внешних адресов.

Формула расчёта

Формула: Количество пулов внешних адресов = $\text{ceil}(n\text{-vip} / n\text{-node})$, где ceil — округление вверх.

Внимание: При использовании виртуальных машин количество виртуальных машин = Количество пулов внешних адресов * n. Здесь n должно быть больше 2, при этом допускается отказ не более одного узла.

- n-vip: количество VIP.
- n-node: количество VIP, которое может обслуживать один узел.

Совместимость аннотаций ingress-nginx

Основные понятия

ingress-nginx — это широко используемый Ingress Controller в Kubernetes, который определяет множество аннотаций для реализации различных функций, выходящих за рамки официального определения ingress.

Поддерживаемые аннотации ingress-nginx

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/app-root	string	x
nginx.ingress.kubernetes.io/affinity	cookie	o ingress не поддерживает. alb rule может настроить cookie hash
nginx.ingress.kubernetes.io/use-regex	bool	
nginx.ingress.kubernetes.io/affinity-mode	"balanced" или "persistent"	o ingress не поддерживает. alb rule может настроить session persistence
nginx.ingress.kubernetes.io/affinity-canary-behavior	"sticky" или "legacy"	o ingress не поддерживает. alb rule может настроить session persistence
nginx.ingress.kubernetes.io/auth-realm	string	v auth
nginx.ingress.kubernetes.io/auth-secret	string	v auth

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/auth-secret-type	string	v auth
nginx.ingress.kubernetes.io/auth-type	"basic" или "digest"	v auth
nginx.ingress.kubernetes.io/auth-tls-secret	string	x
nginx.ingress.kubernetes.io/auth-tls-verify-depth	number	x
nginx.ingress.kubernetes.io/auth-tls-verify-client	string	x
nginx.ingress.kubernetes.io/auth-tls-error-page	string	x
nginx.ingress.kubernetes.io/auth-tls-pass-certificate-to-upstream	"true" или "false"	x
nginx.ingress.kubernetes.io/auth-tls-match-cn	string	x
nginx.ingress.kubernetes.io/auth-url	string	v
nginx.ingress.kubernetes.io/auth-cache-key	string	x
nginx.ingress.kubernetes.io/auth-cache-duration	string	x
nginx.ingress.kubernetes.io/auth-keepalive	number	x
nginx.ingress.kubernetes.io/auth-keepalive-share-vars	"true" или "false"	x
nginx.ingress.kubernetes.io/auth-keepalive-requests	number	x
nginx.ingress.kubernetes.io/auth-keepalive-timeout	number	x
nginx.ingress.kubernetes.io/auth-proxy-set-headers	string	v
nginx.ingress.kubernetes.io/auth-snippet	string	x

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/enable-global-auth	"true" или "false"	o auth
nginx.ingress.kubernetes.io/backend-protocol	string	v
nginx.ingress.kubernetes.io/canary	"true" или "false"	x
nginx.ingress.kubernetes.io/canary-by-header	string	x
nginx.ingress.kubernetes.io/canary-by-header-value	string	x
nginx.ingress.kubernetes.io/canary-by-header-pattern	string	x
nginx.ingress.kubernetes.io/canary-by-cookie	string	x
nginx.ingress.kubernetes.io/canary-weight	number	x
nginx.ingress.kubernetes.io/canary-weight-total	number	x
nginx.ingress.kubernetes.io/client-body-buffer-size	string	x
nginx.ingress.kubernetes.io/configuration-snippet	string	x
nginx.ingress.kubernetes.io/custom-http-errors	int	x
nginx.ingress.kubernetes.io/custom-headers	string	o
nginx.ingress.kubernetes.io/default-backend	string	o можно использовать default-backend ingress
nginx.ingress.kubernetes.io/enable-cors	"true" или "false"	v
nginx.ingress.kubernetes.io/cors-allow-origin	string	v

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/cors-allow-methods	string	v
nginx.ingress.kubernetes.io/cors-allow-headers	string	v
nginx.ingress.kubernetes.io/cors-expose-headers	string	x
nginx.ingress.kubernetes.io/cors-allow-credentials	"true" или "false"	x
nginx.ingress.kubernetes.io/cors-max-age	number	x
nginx.ingress.kubernetes.io/force-ssl-redirect	"true" или "false"	v redirect
nginx.ingress.kubernetes.io/from-to-www-redirect	"true" или "false"	x
nginx.ingress.kubernetes.io/http2-push-preload	"true" или "false"	x
nginx.ingress.kubernetes.io/limit-connections	number	x
nginx.ingress.kubernetes.io/limit-rps	number	x
nginx.ingress.kubernetes.io/global-rate-limit	number	x
nginx.ingress.kubernetes.io/global-rate-limit-window	duration	x
nginx.ingress.kubernetes.io/global-rate-limit-key	string	x
nginx.ingress.kubernetes.io/global-rate-limit-ignored-cidrs	string	x
nginx.ingress.kubernetes.io/permanent-redirect	string	v redirect
nginx.ingress.kubernetes.io/permanent-redirect-code	number	v redirect
nginx.ingress.kubernetes.io/temporal-redirect	string	v redirect

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/preserve-trailing-slash	"true" или "false"	x
nginx.ingress.kubernetes.io/proxy-body-size	string	x
nginx.ingress.kubernetes.io/proxy-cookie-domain	string	x
nginx.ingress.kubernetes.io/proxy-cookie-path	string	x
nginx.ingress.kubernetes.io/proxy-connect-timeout	number	v timeout
nginx.ingress.kubernetes.io/proxy-send-timeout	number	v timeout
nginx.ingress.kubernetes.io/proxy-read-timeout	number	v timeout
nginx.ingress.kubernetes.io/proxy-next-upstream	string	x
nginx.ingress.kubernetes.io/proxy-next-upstream-timeout	number	x
nginx.ingress.kubernetes.io/proxy-next-upstream-tries	number	x
nginx.ingress.kubernetes.io/proxy-request-buffering	string	x
nginx.ingress.kubernetes.io/proxy-redirect-from	string	x
nginx.ingress.kubernetes.io/proxy-redirect-to	string	x
nginx.ingress.kubernetes.io/proxy-http-version	"1.0" или "1.1"	x
nginx.ingress.kubernetes.io/proxy-ssl-secret	string	x
nginx.ingress.kubernetes.io/proxy-ssl-ciphers	string	x
nginx.ingress.kubernetes.io/proxy-ssl-name	string	x

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/proxy-ssl-protocols	string	x
nginx.ingress.kubernetes.io/proxy-ssl-verify	string	x
nginx.ingress.kubernetes.io/proxy-ssl-verify-depth	number	x
nginx.ingress.kubernetes.io/proxy-ssl-server-name	string	x
nginx.ingress.kubernetes.io/enable-rewrite-log	"true" или "false"	x
nginx.ingress.kubernetes.io/rewrite-target	URI	v
nginx.ingress.kubernetes.io/satisfy	string	x
nginx.ingress.kubernetes.io/server-alias	string	x
nginx.ingress.kubernetes.io/server-snippet	string	x
nginx.ingress.kubernetes.io/service-upstream	"true" или "false"	x
nginx.ingress.kubernetes.io/session-cookie-change-on-failure	"true" или "false"	x
nginx.ingress.kubernetes.io/session-cookie-conditional-samesite-none	"true" или "false"	x
nginx.ingress.kubernetes.io/session-cookie-domain	string	x
nginx.ingress.kubernetes.io/session-cookie-expires	string	x
nginx.ingress.kubernetes.io/session-cookie-max-age	string	x
nginx.ingress.kubernetes.io/session-cookie-name	string	x
nginx.ingress.kubernetes.io/session-cookie-path	string	x

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/session-cookie-samesite	string	x
nginx.ingress.kubernetes.io/session-cookie-secure	string	x
nginx.ingress.kubernetes.io/ssl-redirect	"true" или "false"	v
nginx.ingress.kubernetes.io/ssl-passthrough	"true" или "false"	x
nginx.ingress.kubernetes.io/stream-snippet	string	x
nginx.ingress.kubernetes.io/upstream-hash-by	string	x
nginx.ingress.kubernetes.io/x-forwarded-prefix	string	x
nginx.ingress.kubernetes.io/load-balance	string	x
nginx.ingress.kubernetes.io/upstream-vhost	string	v
nginx.ingress.kubernetes.io/denylist-source-range	CIDR	o можно добиться похожего эффекта через modsecurity
nginx.ingress.kubernetes.io/whitelist-source-range	CIDR	o можно добиться похожего эффекта через modsecurity
nginx.ingress.kubernetes.io/proxy-buffering	string	x
nginx.ingress.kubernetes.io/proxy-buffers-number	number	x
nginx.ingress.kubernetes.io/proxy-buffer-size	string	x
nginx.ingress.kubernetes.io/proxy-max-temp-file-size	string	x
nginx.ingress.kubernetes.io/ssl-ciphers	string	x
nginx.ingress.kubernetes.io/ssl-prefer-server-ciphers	"true" или "false"	x

Название	Тип	Поддержка (v — поддерживается, x — не поддерживается, o — частично поддерживается или может быть реализовано через конфигурацию)
nginx.ingress.kubernetes.io/connection-proxy-header	string	x
nginx.ingress.kubernetes.io/enable-access-log	"true" или "false"	o по умолчанию включен access_log, формат фиксирован
nginx.ingress.kubernetes.io/enable-opentelemetry	"true" или "false"	v otel
nginx.ingress.kubernetes.io/opentelemetry-trust-incoming-span	"true" или "false"	v otel
nginx.ingress.kubernetes.io/enable-modsecurity	bool	v modsecurity
nginx.ingress.kubernetes.io/enable-owasp-core-rules	bool	v modsecurity
nginx.ingress.kubernetes.io/modsecurity-transaction-id	string	v modsecurity
nginx.ingress.kubernetes.io/modsecurity-snippet	string	v modsecurity
nginx.ingress.kubernetes.io/mirror-request-body	string	x
nginx.ingress.kubernetes.io/mirror-target	string	x
nginx.ingress.kubernetes.io/mirror-host	string	x

TCP/HTTP Keepalive

Основная концепция

1. ALB поддерживает конфигурацию keepalive на уровне порта. Она может быть настроена на frontend.
2. Keepalive работает между клиентом и ALB, а не между ALB и backend.
3. Реализуется через конфигурацию Nginx, и Nginx **требует и автоматически перезагружается** при изменении конфигурации.
4. TCP keepalive и HTTP keepalive — это два разных понятия:
 1. **TCP keepalive** — это функция TCP-протокола, которая периодически отправляет probe-пакеты для проверки, жива ли связь, когда нет передачи данных. Это помогает обнаруживать и очищать мертвые соединения.

2. **HTTP keepalive** (также известный как persistent connections) позволяет нескольким HTTP-запросам использовать одно и то же TCP-соединение, избегая накладных расходов на установку новых соединений. Это улучшает производительность за счёт снижения задержек и использования ресурсов.

CRD

keepalive:

properties:

http:

description: Downstream L7 keepalive

properties:

header_timeout:

description: Keepalive header timeout. Default is not set.

type: string

requests:

description: Keepalive requests. Default is 1000.

type: integer

timeout:

description: Keepalive timeout. Default is 75s.

type: string

type: object

tcp:

description: TCPKeepAlive defines TCP keepalive parameters (SO_KEEPALIVE)

properties:

count:

description: The TCP_KEEPCNT socket option.

type: integer

idle:

description: The TCP_KEEPIDLE socket option.

type: string

interval:

description: The TCP_KEEPINTVL socket option.

type: string

type: object

type: object

Её можно настроить только на Frontend в **.spec.config.keepalive**.

ModSecurity

ModSecurity — это открытый Web Application Firewall (WAF), предназначенный для защиты веб-приложений от вредоносных атак. Он поддерживается сообществом с открытым исходным кодом и поддерживает различные языки программирования и веб-серверы.

Платформа Load Balancer (ALB) поддерживает настройку ModSecurity, позволяя создавать индивидуальные конфигурации на уровне Ingress.

Терминология

Термин	Объяснение
owasp-core-rules	OWASP Core Rule Set — это набор правил с открытым исходным кодом, используемый для обнаружения и предотвращения распространённых атак на веб-приложения.

Процедура эксплуатации

Настройте ModSecurity, добавив аннотации в YAML-файл соответствующего ресурса или настроив CR.

Метод 1: Добавление аннотаций

Добавьте следующие аннотации в поле `metadata.annotations` соответствующего YAML-файла для настройки ModSecurity.

- **Аннотации, совместимые с Ingress-Nginx**

Аннотация	Тип	Применимый объект	Описание
<code>nginx.ingress.kubernetes.io/enabled-modsecurity</code>	bool	Ingress	Включить ModSecurity.
<code>nginx.ingress.kubernetes.io/enabled-owasp-core-rules</code>	bool	Ingress	Включить OWASP Core Rule Set.
<code>nginx.ingress.kubernetes.io/modsecurity-transaction-id</code>	string	Ingress	Используется для идентификации уникальных ID транзакций для каждого запроса, что помогает при логировании и отладке.
<code>nginx.ingress.kubernetes.io/modsecurity-snippet</code>	string	Ingress, ALB, FT, Rule	Позволяет пользователям вставлять собственные конфигурации ModSecurity для удовлетворения специфических требований безопасности.

- **Специальные аннотации ALB**

Аннотация	Тип	Применимый объект	Объяснение
<code>alb.modsecurity.cpaas.io/use-recommend</code>	bool	Ingress	Включить или отключить рекомендуемые правила

Аннотация	Тип	Применимый объект	Объяснение
			ModSecurity; установите true для применения предопределённого набора правил безопасности.
alb.modsecurity.cpaas.io/cmref	string	Ingress	Ссылка на конкретные конфигурации, например, можно загрузить пользовательские настройки безопасности, указав путь ссылки на ConfigMap (\$ns/\$name#\$section).

Метод 2: Настройка CR

1. Откройте файл конфигурации ALB, FT или Rule, который необходимо настроить.
2. Добавьте следующие поля в spec.config по необходимости.

```
{ "modsecurity": {
  "enable": true, # Включить или отключить ModSecurity
  "transactionId": "$xx", # Использовать ID из Nginx
  "useCoreRules": true, # Добавить modsecurity_rules_file /etc/nginx/owasp-modsecurity-crs/nginx-modsecurity.conf
  "useRecommend": true, # Добавить modsecurity_rules_file /etc/nginx/modsecurity/modsecurity.conf
  "cmRef": "$ns/$name#$section", # Добавить конфигурацию из ConfigMap
}}
```

3. Сохраните и примените файл конфигурации.

Переопределение

Если ModSecurity не настроен в Rule, будет предпринята попытка найти конфигурацию в FT; если в FT конфигурация отсутствует, будет использована конфигурация из ALB.

Пример конфигурации

В следующем примере развёртывается ALB с именем **waf-alb** и демонстрационное backend-приложение с именем **hello**. Кроме того, развёртывается Ingress с именем **ing-waf-enable**, который определяет маршрут **/waf-enable** и настраивает правила ModSecurity. Любой запрос, содержащий параметр запроса **test**, значение которого включает строку **test**, будет заблокирован.

```
cat <<EOF | kubectl apply -f -
apiVersion: crd.alauda.io/v2
kind: ALB2
```

```

metadata:
  name: waf-alb
  namespace: cpaas-system
spec:
  config:
    loadbalancerName: waf-alb
    projects:
      - ALL_ALL
    replicas: 1
    type: nginx
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/enable-modsecurity: "true"
    nginx.ingress.kubernetes.io/modsecurity-transaction-id: "$request_id"
    nginx.ingress.kubernetes.io/modsecurity-snippet: |
      SecRuleEngine On
      SecRule ARGS:test "@contains test" "id:1234,deny,log"
  name: ing-waf-enable
spec:
  ingressClassName: waf-alb
  rules:
    - http:
        paths:
          - backend:
              service:
                name: hello
                port:
                  number: 80
              path: /waf-enable
              pathType: ImplementationSpecific
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ing-waf-normal
spec:
  ingressClassName: waf-alb

```

```

rules:
  - http:
      paths:
        - backend:
            service:
              name: hello
            port:
              number: 80
            path: /waf-not-enable
            pathType: ImplementationSpecific
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello
spec:
  replicas: 1
  selector:
    matchLabels:
      service.cpaas.io/name: hello
      service_name: hello
  template:
    metadata:
      labels:
        service.cpaas.io/name: hello
        service_name: hello
    spec:
      containers:
        - name: hello-world
          image: docker.io/hashicorp/http-echo
          imagePullPolicy: IfNotPresent
---
apiVersion: v1
kind: Service
metadata:
  name: hello
spec:
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4

```

```
ipFamilyPolicy: SingleStack
ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 5678
selector:
  service_name: hello
sessionAffinity: None
type: ClusterIP
EOF
```

Сравнение различных методов Ingress

Платформа MWS Container Platform поддерживает несколько спецификаций ingress-трафика в экосистеме Kubernetes. В этом документе проводится их сравнение (Service, Ingress, Gateway API и ALB Rule), чтобы помочь пользователям сделать правильный выбор.

Для L4 (TCP/UDP) трафика

Сервисы типа LoadBalancer, Gateway API и ALB Rules могут все обеспечивать внешний доступ к L4 трафику. Здесь мы рекомендуем использовать подход с сервисом типа LoadBalancer.

И Gateway API, и ALB Rules реализованы через ALB, который является прокси в пространстве пользователя, и их производительность значительно снижается при обработке L4 трафика по сравнению с сервисами типа LoadBalancer.

Для L7 (HTTP/HTTPS) трафика

Хотя Ingress, GatewayAPI и ALB Rules все могут обеспечивать внешний доступ к L7 трафику, они отличаются по возможностям и моделям изоляции.

Ingress

Ingress — это стандартная спецификация, принятая сообществом Kubernetes и рекомендуемая для использования по умолчанию.

Ingress обрабатывается экземплярами ALB, которые управляются администратором платформы.

GatewayAPI

GatewayAPI предоставляет более гибкий режим изоляции, однако он менее зрелый, чем Ingress.

Используя GatewayAPI, разработчик может создавать собственные изолированные экземпляры ALB для обработки правил GatewayAPI.

Поэтому, если вам нужно делегировать создание и управление экземплярами ALB разработчикам, следует выбрать GatewayAPI.

ALB Rule

ALB Rule (Load Balancer в UI) предоставляет наиболее гибкие правила сопоставления трафика и максимальные возможности. Фактически, и Ingress, и GatewayAPI реализуются путем преобразования в ALB Rules.

Однако ALB Rule сложнее, чем Ingress и GatewayAPI, и не является стандартным API сообщества. Поэтому мы рекомендуем использовать его только в тех случаях, когда Ingress и GatewayAPI не удовлетворяют вашим требованиям.

HTTP Redirect

Обзор

HTTP redirect — это функция, предоставляемая ALB. Она напрямую возвращает HTTP-код 30х для запроса, который соответствует правилу. Заголовок Location используется для указания клиенту перенаправиться на новый URL.

ALB поддерживает настройку редиректа на уровне порта и правила.

CRD

```
redirect:
  properties:
    code:
      type: integer
    host:
      type: string
    port:
      type: integer
    prefix_match:
      type: string
    replace_prefix:
      type: string
    scheme:
      type: string
    url:
      type: string
  type: object
```

Редирект может быть настроен на:

- Frontend: .spec.config.redirect
- Rule: .spec.config.redirect

Аннотация Ingress

Аннотация	Описание
nginx.ingress.kubernetes.io/permanent-redirect	Соответствует URL в CR, по умолчанию устанавливает код 301

Аннотация	Описание
nginx.ingress.kubernetes.io/permanent-redirect-code	Соответствует code в CR
nginx.ingress.kubernetes.io/temporal-redirect	Соответствует URL в CR, по умолчанию устанавливает код 302
nginx.ingress.kubernetes.io/temporal-redirect-code	Соответствует code в CR
nginx.ingress.kubernetes.io/ssl-redirect	Соответствует scheme в CR, по умолчанию устанавливает scheme HTTPS
nginx.ingress.kubernetes.io/force-ssl-redirect	Соответствует scheme в CR, по умолчанию устанавливает scheme HTTPS

SSL-Redirect

1. SSL-redirect и force-ssl-redirect отличаются тем, что SSL-redirect действует только если у ingress есть сертификат для соответствующего домена, тогда как force-ssl-redirect действует независимо от наличия сертификата.
2. Для HTTPS-портов, если настроен только SSL-redirect, редирект не будет установлен.

Редирект на уровне порта

Когда редирект настроен на уровне порта, все запросы к этому порту будут перенаправлены согласно конфигурации редиректа.

Редирект на уровне правила

Когда редирект настроен на уровне правила, запросы, соответствующие этому правилу, будут перенаправлены согласно конфигурации редиректа.

L4/L7 Таймаут

Обзор

L4/L7 таймаут — это функция, предоставляемая ALB. Она используется для настройки времени таймаута для L4/L7 прокси.

Таймаут реализован с помощью Lua-скрипта, и при его изменении **не требуется перезагрузка Nginx**.

CRD

```

timeout:
  properties:
    proxy_connect_timeout_ms:
      type: integer
    proxy_read_timeout_ms:
      type: integer

```



```
proxy_send_timeout_ms:
  type: integer
type: object
```

Конфигурация может быть задана в:

- Frontend: `.spec.config.timeout`
- Rule: `.spec.config.timeout`

Типы таймаутов

Существует три типа таймаутов:

1. **proxy_connect_timeout_ms**: Определяет время таймаута для установления соединения с upstream-сервером. Если соединение не будет установлено в течение этого времени, запрос завершится с ошибкой.
2. **proxy_read_timeout_ms**: Определяет время таймаута для чтения ответа от upstream-сервера. Таймаут устанавливается между двумя последовательными операциями чтения, а не на весь ответ целиком. Если в течение этого времени данные не поступают, соединение закрывается.
3. **proxy_send_timeout_ms**: Определяет время таймаута для отправки запроса upstream-серверу. Аналогично таймауту чтения, он устанавливается между двумя последовательными операциями записи. Если в течение этого времени данные не могут быть отправлены, соединение закрывается.

Аннотация Ingress

Аннотация	Описание
<code>nginx.ingress.kubernetes.io/proxy-connect-timeout</code>	Соответствует <code>proxy_connect_timeout_ms</code> в CRD
<code>nginx.ingress.kubernetes.io/proxy-read-timeout</code>	Соответствует <code>proxy_read_timeout_ms</code> в CRD
<code>nginx.ingress.kubernetes.io/proxy-send-timeout</code>	Соответствует <code>proxy_send_timeout_ms</code> в CRD

Таймаут на уровне порта

Вы можете настроить таймаут непосредственно на порту, который используется как таймаут L4.

GatewayAPI

GatewayAPI — это новый стандарт для Kubernetes ingress.

ALB также поддерживает GatewayAPI. Каждый ресурс Gateway будет преобразован в ресурс ALB.

Listener и Router будут обрабатываться непосредственно в ALB. Они не будут преобразованы в **Frontend** и **Rule**.

Создание сервисов

В Kubernetes Service — это способ открыть сетевое приложение, работающее в виде одного или нескольких Pod в вашем кластере.

Веб-консоль

1. Перейдите в Container Platform.
2. В левой навигационной панели выберите **Network > Services**.
3. Нажмите Create Service.
4. Следуйте инструкциям для настройки соответствующих параметров.

Параметр	Описание
Virtual IP Address	Если включено, для этого Service будет выделен ClusterIP, который можно использовать для обнаружения сервиса внутри кластера. Если отключено, будет создан headless Service, который обычно используется для StatefulSet .
Type	ClusterIP : Открывает сервис на внутреннем IP кластера. При выборе этого значения Service доступен только внутри кластера. NodePort : Открывает сервис на IP каждого Node на статическом порту (NodePort). ExternalName : Отображает сервис на содержимое поля externalName (например, на hostname api.foo.bar.example). LoadBalancer : Открывает сервис внешне с помощью внешнего балансировщика нагрузки. Kubernetes не предоставляет собственный компонент балансировки нагрузки; вы должны обеспечить его самостоятельно или интегрировать кластер с облачным провайдером.
Target Component	Workload : Сервис будет перенаправлять запросы на конкретный workload, который соответствует меткам, например: project.cpaas.io/name: projectname и service.cpaas.io/name: deployment-name. Virtualization : Сервис будет перенаправлять запросы на конкретную виртуальную машину или группу виртуальных машин. Label Selector : Сервис будет перенаправлять запросы на определённый тип workload с указанными метками, например, environment: release .
Port	Используется для настройки сопоставления портов для этого Service. В следующем примере другие Pod внутри кластера могут обращаться к этому Service через виртуальный IP (если включён) и TCP порт 80; запросы будут перенаправлены на внешний TCP порт 6379 или redis Pod целевого компонента. Protocol : Протокол, используемый Service, поддерживаются: TCP, UDP, HTTP, HTTP2, HTTPS, gRPC .

Параметр	Описание
	Service Port: Номер порта, который Service экспонирует внутри кластера, то есть Port, например, <i>80</i>. Container Port: Целевой порт (или имя), на который маппится service port, то есть targetPort, например, <i>6379</i> или <i>redis</i>. Service Port Name: Генерируется автоматически. Формат: <protocol>-<service port>-<container port>, например: <i>tcp-80-6379</i> или <i>tcp-80-redis</i>.
Session Affinity	Сессионная аффиность на основе IP-адреса источника (ClientIP). Если включено, все запросы с одного IP-адреса будут направляться на один и тот же сервер при балансировке нагрузки, что гарантирует обработку запросов от одного клиента одним сервером.

5. Нажмите **Create**.

CLI

```
kubectl apply -f simple-service.yaml
```

Создать сервис на основе существующего ресурса deployment **my-app**.

```
kubectl expose deployment my-app \
  --port=80 \
  --target-port=8080 \
  --name=test-service \
  --type=NodePort \
  -n p1-1
```

Аннотации для сервиса типа LoadBalancer

AWS EKS Cluster

Для подробного описания аннотаций LoadBalancer в EKS, пожалуйста, обратитесь к Annotation Usage Documentation .

Ключ	Значение	Описание
service.beta.kubernetes.io/aws-load-balancer-type	external: Использовать официальный AWS LoadBalancer Controller.	Определяет контроллер для типа LoadBalancer. Внимание: Пожалуйста, заранее свяжитесь с администратором платформы для развертывания AWS LoadBalancer Controller.
service.beta.kubernetes.io/aws-load-balancer-nlb-target-type	instance: Трафик будет направляться на Pod через NodePort.	Определяет, как трафик достигает Pod.

Ключ	Значение	Описание
	ip: Трафик направляется напрямую на Pod (кластер должен использовать Amazon VPC CNI).	
service.beta.kubernetes.io/aws-load-balancer-scheme	internal: Частная сеть. internet-facing: Публичная сеть.	Определяет, использовать ли частную или публичную сеть.
service.beta.kubernetes.io/aws-load-balancer-ip-address-type	IPv4 dualstack	Определяет поддерживаемый стек IP-адресов.

Huawei Cloud CCE Cluster

Для подробного описания аннотаций LoadBalancer в CCE, пожалуйста, обратитесь к Annotation Usage Documentation .

Ключ	Значение	Описание
kubernetes.io/elb.id		Укажите ID облачного балансировщика нагрузки, необходимо использовать существующий облачный балансировщик.
kubernetes.io/elb.autocreate	Пример: <pre>{"type":"public","bandwidth_name":"cce-bandwidth-1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","available_zone":["cn-north-4b"],"l4_flavor_name":"L4_flavor.elb.s1.small"}</pre>	Для создания нового облачного балансировщика нагрузки.
kubernetes.io/elb.subnet-id		ID подсети, в которой находится кластер. При версии Kubernetes 1.11.7-r0 и ниже это поле обязательно для заполнения при создании нового облачного балансировщика нагрузки.
kubernetes.io/elb.class	union:	Определяет тип создаваемого облачного балансировщика нагрузки

Ключ	Значение	Описание
	Общая балансировка нагрузки. performance: Эксклюзивная балансировка нагрузки, поддерживается только в Kubernetes версии 1.17 и выше.	
kubernetes.io/elb.enterpriseID		Определяет проект предприятия, к которому принадлежит создаваемый облачный балансировщик нагрузки.

Azure AKS Cluster

Для подробного описания аннотаций LoadBalancer в AKS обратитесь к документации Azure.

Ключ	Значение	Описание
service.beta.kubernetes.io/azure-load-balancer-internal	true: Частная сеть. false: Публичная сеть.	Определяет, использовать ли частную или публичную сеть.

Google GKE Cluster

Для подробного описания аннотаций LoadBalancer в GKE обратитесь к документации Google Cloud.

Ключ	Значение	Описание
networking.gke.io/load-balancer-type	Internal	Определяет использование частной сети.
load.google.com/l4-rbs	enabled	По умолчанию публичный. Если этот параметр настроен, трафик будет направляться напрямую на Pod.

Ingress

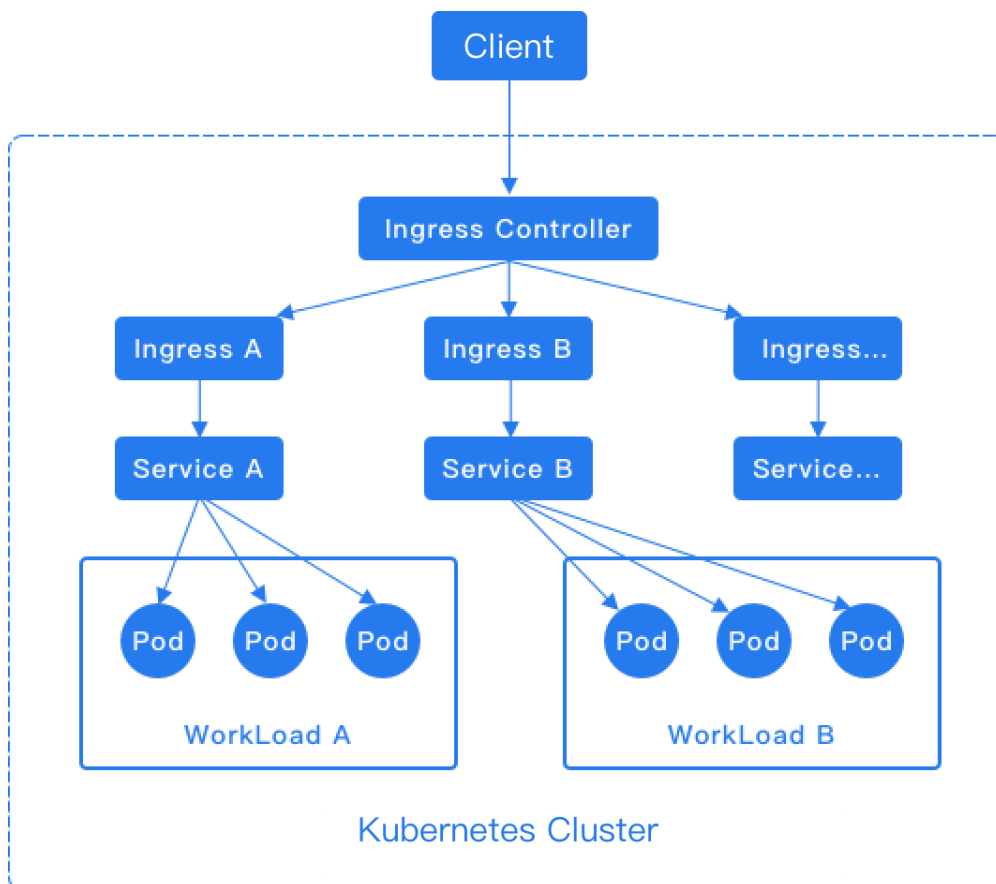
Правила Ingress (Kubernetes Ingress) открывают HTTP/HTTPS маршруты снаружи кластера для внутренней маршрутизации (Kubernetes Service), что позволяет контролировать внешний доступ к вычислительным компонентам.

Создайте Ingress для управления внешним HTTP/HTTPS доступом к Service.

Внимание: При создании нескольких ingress в одном и том же namespace разные ingress **НЕ ДОЛЖНЫ** иметь одинаковые **Domain**, **Protocol** и **Path** (то есть дублирование точек доступа не допускается).

Метод реализации

Правила Ingress зависят от реализации Ingress Controller, который отвечает за отслеживание изменений в Ingress и Service. После создания нового правила Ingress внутри Ingress Controller автоматически создаётся правило переадресации, соответствующее правилу Ingress. Когда Ingress Controller получает запрос, он сопоставляет правило переадресации с правилом Ingress и распределяет трафик по указанным внутренним маршрутам, как показано на схеме ниже.



Внимание: Для протокола HTTP Ingress поддерживает только порт 80 в качестве внешнего порта. Для протокола HTTPS Ingress поддерживает только порт 443 в качестве внешнего порта. Балансировщик нагрузки платформы автоматически добавит порты 80 и 443 для прослушивания.

Быстрый старт

Далее мы используем community-версию Ingress-NGINX, чтобы продемонстрировать, как получить доступ к вашему приложению с помощью контроллера NGINX.

1. Разверните контроллер **Ingress-NGINX**.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.12.2/deploy/static/provider/cloud/deploy.yaml
```

После выполнения этой команды автоматически создаются следующие ресурсы:

Вид	Имя	Описание
Namespace	ingress-nginx	Ресурсы для изоляции контроллеров

Вид	Имя	Описание
ServiceAccount	ingress-nginx	Сервисный аккаунт для контроллера
ClusterRole	ingress-nginx	Разрешения на уровне кластера
ClusterRoleBinding	ingress-nginx	Привязка ClusterRole к SA
ConfigMap	ingress-nginx-controller	Настройка поведения контроллера (например, уровни логирования, таймаут прокси и т.д.)
ValidatingWebhookConfig	ingress-nginx-admission	Вебхук для проверки легитимности конфигурации Ingress (опционально)
Service (TCP/UDP)	ingress-nginx-controller	Тип по умолчанию LoadBalancer , можно изменить на NodePort .
Deployment	ingress-nginx-controller	
Pod	ingress-nginx-controller-xxx	
Role/RoleBinding	связанные с admission	Поддержка вебхука
Job	ingress-nginx-admission-create	Регистрация вебхука

Если вы хотите изменить адрес реестра по умолчанию, можно скачать YAML-файл с помощью **curl**, изменить его и затем применить.

```
curl -O https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.12.2/deploy/static/provider/cloud/deploy.yaml
```

Ожидайте запуска Pod ingress-nginx-controller-xxx.

2. Локальное тестирование

- Создайте простой веб-сервер и связанный с ним сервис:

```
kubectl create deployment demo --image=nginx --port=80
```

```
kubectl expose deployment demo
```

- Создайте ресурс ingress. В этом примере используется хост, который сопоставляется с **localhost**:

```
kubectl create ingress demo-localhost --class=nginx \
--rule="demo.local/*=demo:80"
```

- Пробросьте локальный порт к контроллеру ingress:

```
kubectl port-forward --namespace=ingress-nginx service/ingress-nginx-controller
8080:80
```

- Получите доступ к вашему развертыванию с помощью curl:

```
curl --resolve demo.local:8080:127.0.0.1 http://demo.local:8080
```

Внимание: Этот параметр временно разрешает доменное имя `demo.local` в IP `127.0.0.1` и используется на порту `8080`. При посещении `http://demo.local:8080` вы фактически посещаете `http://127.0.0.1:8080`.

С другой стороны, следует настроить **hosts**:

```
echo "127.0.0.1 demo.local" | sudo tee -a /etc/hosts
```

В итоге вы должны увидеть HTML-ответ с текстом вроде "Welcome to nginx!".

После этого вы можете получить доступ к сайту по адресу **`http://demo.local:8080/`**.

Внимание: Тип **`ingress-nginx-controller`** по умолчанию — **`LoadBalancer`**. Если поле **`EXTERNAL-IP`** показывает **`pending`**, это означает, что вашему Kubernetes кластеру не удалось создать балансировщик нагрузки.

Если вы интегрируетесь с провайдером, который поддерживает указание IP-адресов балансировщика нагрузки для Service через (специфичные для провайдера) аннотации, рекомендуется использовать этот способ.

3. Онлайн-тестирование

Когда у вашего **`ingress-nginx-controller`** (Service типа `LoadBalancer`) появляется **`EXTERNAL-IP`**, вы можете создать ресурс `ingress`. В следующем примере предполагается, что вы настроили DNS-запись для **`www.developer.io`**:

```
kubectl create ingress demo --class=nginx \
  --rule="www.developer.io/*=demo:80"
```

Вы можете получить доступ к **`http://www.developer.io`** и увидеть тот же вывод.

Создание Ingress

Прerequisites

- В текущем **`namespace`** должен быть доступен **`Service`**.
- Убедитесь у администратора, что для проекта, связанного с текущим **`namespace`**, выделено используемое доменное имя.
- Для доступа к домену через HTTPS необходимо предварительно сохранить HTTPS-сертификат в виде TLS-секрета.

Пример Ingress

```
# nginx-ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  namespace: k-1
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: / (1)
spec:
  ingressClassName: nginx (2)
```


rules:

- host: demo.local (3)

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: nginx-service

port:

number: 80

1. Для получения дополнительных настроек смотрите документацию [Kubernetes](#).
2. Используется контроллер **ingress-nginx**.
3. Если вы хотите запускать **ingress** только локально, предварительно настройте **hosts**.

Веб-консоль

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Network > Ingress**.
3. Нажмите Create Ingress.
4. Используйте инструкции ниже для настройки параметров.

Параметр	Описание
Ingress Class	Ingress может реализовываться разными контроллерами с разными именами IngressClass . Если на платформе доступно несколько ingress контроллеров, пользователь может выбрать нужный с помощью этого параметра.
Domain Name	Хосты могут быть точными совпадениями (например, foo.bar.com) или шаблонами с подстановочными знаками (например, *.foo.com). Доступные доменные имена выделяются администратором платформы.
Certificates	TLS-секрет или сертификаты, выделенные администратором платформы.
Match Type и Path	Prefix: Совпадение по префиксу пути, например, /abcd совпадает с /abcd/efg или /abcde. Exact: Совпадение по точному пути, например, /abcd. Implementation specific: Если вы используете кастомный Ingress контроллер для управления правилами Ingress, можно позволить контроллеру решать самостоятельно.
Service	Внешний трафик будет перенаправлен на этот Service.
Service Port	Укажите порт Service, на который будет перенаправлен трафик.

5. Нажмите **Create**.

CLI

```
kubectl apply -f nginx-ingress.yaml
```

Внимание: Если у ingress отсутствует Ingress Class, все ALB-инстансы, выделенные этому проекту, будут обрабатывать этот ingress.

Настройка Gateway

Входящий шлюз (Gateway) — это экземпляр, развернутый из Gateway Class. Он создает слушатели для перехвата внешнего трафика на указанных доменных именах и портах. Вместе с правилами маршрутизации он может направлять указанный внешний трафик на соответствующие backend-экземпляры.

Создайте входящий шлюз для более точного распределения сетевых ресурсов.

Терминология

Название ресурса	Обзор	Инструкция по использованию
Gateway Class	В стандартной документации Gateway API Gateway Class определяется как шаблон для создания шлюзов. Разные шаблоны могут создавать входящие шлюзы для различных бизнес-сценариев, что облегчает быстрое управление трафиком.	Платформа включает выделенные Gateway Classes.
Inbound Gateway	Входящий шлюз соответствует конкретным экземплярам ресурсов, и пользователь может эксклюзивно использовать все слушающие и вычислительные ресурсы этого входящего шлюза. Это конфигурация правил маршрутизации, действующая для слушателя. При обнаружении внешнего трафика шлюз распределяет его на backend-экземпляры согласно правилам маршрутизации.	Его можно рассматривать как экземпляр балансировщика нагрузки.
Route Rule	Правила маршрутизации определяют набор правил для распределения трафика от шлюза к сервисам. В настоящее время стандартно поддерживаемые типы правил маршрутизации в Gateway API включают HTTPRoute, TCPRoute, UDPRoute и др.	Платформа поддерживает прослушивание протоколов HTTP, HTTPS, TCP и UDP.

Предварительные требования

Администратор платформы должен убедиться, что кластер поддерживает внутреннюю маршрутизацию типа LoadBalancer. Для публичных облачных кластеров должен быть установлен LoadBalancer Service Controller. В непубличных облачных кластерах платформа предоставляет функцию пула внешних адресов, которая позволяет внутренней

маршрутизации типа LoadBalancer автоматически получать IP из пула внешних адресов для внешнего доступа после завершения настройки.

Создание Gateway

Веб-консоль

1. Перейдите в **Container Platform**.
2. В левой навигационной панели выберите **Network > Inbound Gateway**.
3. Нажмите Create Inbound Gateway.
4. Следуйте инструкциям для настройки конкретных параметров.

Параметр	Описание
Name	Имя входящего шлюза.
Gateway Class	Gateway Class определяет поведение шлюза, аналогично концепции классов хранения (StorageClasses); это ресурс кластера. Dedicated: входящий шлюз будет соответствовать конкретному экземпляру ресурса, и пользователь сможет использовать все слушатели и вычислительные ресурсы этого шлюза.
Specification	Можно выбрать рекомендованный сценарий использования в зависимости от потребностей или настроить лимиты ресурсов самостоятельно.
Access Address	Адрес входящего шлюза, который по умолчанию получается автоматически.
Internal Routing Annotation	Используется для объявления конфигурации или возможностей внутренней маршрутизации типа LoadBalancer.

5. Нажмите **Create**.

CLI

```
kubectrl apply -f demo-gateway.yaml
```

Просмотр ресурсов, созданных платформой

После создания входящего шлюза платформа автоматически создает множество ресурсов. Не удаляйте перечисленные ниже ресурсы.

Ресурсы, создаваемые по умолчанию	Имя
Ресурс типа ALB2	name-lb-random
Deployment	name-lb-random
Внутренняя маршрутизация	• name-lb-random • name-lb-random-lb-random
Конфигурационный словарь	• name-lb-random-port-info

Ресурсы, создаваемые по умолчанию	Имя
	name-lb-random
Service Account	name-lb-random-serviceaccount

Обновление Gateway

Внимание: Обновление входящего шлюза приведет к прерыванию сервиса на 3-5 минут. Пожалуйста, выбирайте подходящее время для выполнения этой операции.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Network > Inbound Gateway**.
3. Нажмите : > **Update**.
4. Обновите конфигурацию входящего шлюза по необходимости.

Внимание: Задавайте спецификации в соответствии с бизнес-требованиями.

5. Нажмите **Update**.

Добавление слушателя

Отслеживает трафик по указанным доменным именам и перенаправляет его на backend-экземпляры согласно привязанным правилам маршрутизации.

Пререквизиты

- Если необходимо отслеживать протокол HTTP, заранее свяжитесь с администратором для подготовки **доменного имени**.
- Если необходимо отслеживать протокол HTTPS, заранее свяжитесь с администратором для подготовки **доменного имени и сертификата**.

Веб-консоль

1. В левой навигационной панели выберите **Network > Inbound Gateway**.
2. Нажмите на Имя входящего шлюза.
3. Нажмите Add Listener.
4. Следуйте инструкциям для настройки конкретных параметров.

Параметр	Описание
Протокол и порт слушателя	В настоящее время поддерживается мониторинг протоколов HTTP, HTTPS, TCP и UDP, можно ввести порт для мониторинга, например: 80. Внимание: • При совпадении портов типы слушателей HTTP, HTTPS и TCP не могут сосуществовать; можно выбрать только один из протоколов. • При использовании HTTP или HTTPS и совпадении портов доменные имена должны быть разными.

Параметр	Описание
Доменное имя	Выберите доступное доменное имя в текущем namespace для мониторинга сетевого трафика, обращающегося к этому домену. Внимание: протоколы TCP и UDP не поддерживают выбор доменных имен.

5. Нажмите **Create**.

CLI

```
kubectrl patch gateway test \
-n k-1 \
--type=merge \
-p '{
  "metadata": {
    "annotations": {
      "listeners.cpaas.io/creationTimestamp": "[\"2025-05-26T02:05:56.135Z\", \"2025-05-26T03:33:52.431Z\"]",
      "listeners.cpaas.io/display-name": "[\"\", \"\"]"
    }
  },
  "spec": {
    "listeners": [
      {
        "allowedRoutes": {
          "namespaces": {
            "from": "All"
          }
        },
        "name": "gateway-metric",
        "protocol": "TCP",
        "port": 11782
      },
      {
        "allowedRoutes": {
          "namespaces": {
            "from": "All"
          }
        },
        "name": "demo-listener",
        "protocol": "HTTP",
        "port": 8088,
```

```

        "hostname": "developer.test.cn"
    }
]
}
}'

```

Создание правил маршрутизации

Правила маршрутизации задают политики маршрутизации для входящего трафика, аналогично входящим правилам (Kubernetes Ingress). Они обеспечивают доступ к сетевому трафику, отслеживаемому шлюзом, для внутренней маршрутизации кластера (Kubernetes Service), облегчая стратегию маршрутизации и переадресации. Главное отличие в том, что они нацелены на разные объекты сервиса: входящие правила обслуживают Ingress Controller, а правила маршрутизации — Ingress Gateway.

После настройки прослушивания в ingress gateway шлюз будет в реальном времени отслеживать трафик с указанных доменов и портов. Правила маршрутизации могут направлять входящий трафик на backend-экземпляры по желанию.

Пример пользовательского ресурса HTTPRoute (CR)

```

# example-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute (1)
metadata:
  namespace: k-1
  name: example-http-route
  annotations:
    cpaas.io/display-name: ""
spec:
  hostnames:
    - developer.test.cn
  parentRefs:
    - kind: Gateway
      namespace: k-1
      name: test
      sectionName: demo-listener (2)
  rules:
    - matches:
        - path:
            type: Exact
            value: "/demo"
      filters: []
      backendRefs:
        - kind: Service

```

name: test-service
namespace: k-1
port: 80
weight: 100

1. Доступные типы: HTTPRoute, TCPRoute, UDPRoute.
2. Имя слушателя Gateway.

Внимание: Если для объекта **Path** в правиле маршрутизации типа HTTPRoute нет совпадающего правила, автоматически добавляется правило с режимом PathPrefix и значением /.

Создание маршрута

Веб-консоль

1. Зайдите в Container Platform.
2. В левой навигационной панели выберите **Network > Route Rules**.
3. Нажмите Create Route Rule.
4. Следуйте инструкциям для настройки параметров.

Параметр	Описание
Тип маршрута	В настоящее время поддерживаются типы маршрутов: HTTPRoute, TCPRoute, UDPRoute. Внимание: HTTPRoute поддерживает публикацию слушателей (Listener) протоколов HTTP и HTTPS.
Публикация на слушатель	В левой панели выберите созданный Ingress Gateway , в правой — созданный Listener . Платформа опубликует созданные правила маршрутизации на выбранный слушатель, позволяя шлюзу перенаправлять перехваченный трафик на указанные backend-экземпляры. Внимание: Нельзя публиковать правила маршрутизации на слушателя, который работает на порту 11782 или уже содержит TCP или UDP маршруты.
Совпадение (Match)	Можно добавить одно или несколько правил совпадения для перехвата трафика, соответствующего требованиям. Например, перехват трафика с указанным Path, перехват трафика с указанным методом и т.д. Примечание: • Нажмите Add; при добавлении нескольких правил маршрутизации связь между ними — 'AND', все правила должны совпадать для активации. • Нажмите Add Match; при добавлении нескольких групп правил связь между группами — 'OR', достаточно совпадения любой группы для активации.

Параметр	Описание
	• TCPRoute и UDPRoute не поддерживают настройку правил совпадения. • Если объект совпадения — path, а метод совпадения — Exact или PathPrefix, значение value должно начинаться с "/" и не должно содержать символы типа "//", "/./", "/../", "%2f", "%2F", "#", "/..", "/." и т.п.
Действие (Action)	Можно добавить одно или несколько действий для обработки перехваченного трафика. • Header: заголовок HTTP-сообщения содержит много метаданных, предоставляющих дополнительную информацию о запросе или ответе. Изменяя поля заголовка, сервер может влиять на обработку запроса и ответа. • Redirect: совпадающий URL будет обработан указанным образом, затем запрос будет инициирован заново. • Rewrite: совпадающий URL будет обработан указанным образом, затем запрос будет перенаправлен на другой путь ресурса или имя файла. Примечание: • Нажмите Add; при добавлении нескольких правил действий платформа выполнит все действия последовательно в порядке отображения. • TCPRoute и UDPRoute не поддерживают настройку правил действий. • В одном правиле маршрутизации не может быть несколько действий типа Header с одинаковым значением value. • В одном правиле маршрутизации может быть только одно действие типа Redirect или Rewrite, и только один режим из FullPath или PrefixPath. • Если хотите использовать операцию PrefixPath, сначала добавьте правило совпадения с режимом PathPrefix.
Backend Instance	После активации правила трафик будет перенаправлен на backend-экземпляры согласно выбранным внутренним маршрутам и портам в текущем namespace. Можно также настроить веса, при этом экземпляры с большим весом будут выбираться с большей вероятностью. Внимание: Процент рядом с весом показывает вероятность перенаправления на этот экземпляр, рассчитываемую как отношение текущего значения веса к сумме всех весов.

5. Нажмите **Create**.

CLI

```
kubectl apply -f example-httprouete.yaml
```

Создание доменного имени

Добавьте ресурсы доменных имен на платформу и выделите домены для использования всеми проектами в кластере или ресурсами в конкретном проекте. При создании доменного имени поддерживается привязка сертификата.

Внимание: Созданные на платформе доменные имена должны иметь разрешение на адрес балансировщика нагрузки кластера, прежде чем к ним можно будет получить доступ по доменному имени. Поэтому необходимо убедиться, что добавленные на платформу доменные имена успешно зарегистрированы и что доменные имена разрешаются на адрес балансировщика нагрузки кластера.

Успешно созданные и выделенные доменные имена на платформе могут использоваться в следующих функциях **Container Platform**:

- Создание входящих правил: Network Management > Inbound Rules > Create Inbound Rule
- Создание нативных приложений: Application Management > Native Applications > Create Native Application > Add Inbound Rule
- Добавление прослушиваемых портов для балансировки нагрузки: Network Management > Load Balancer Details > Add Listening Port

После привязки доменного имени к сертификату разработчики приложений могут просто выбрать доменное имя при настройке балансировщика нагрузки и входящих правил, что позволит использовать сертификат, связанный с доменным именем, для поддержки https.

Пример ресурса Domain custom resource (CR)

```
# test-domain.yaml
apiVersion: crd.alauda.io/v2
kind: Domain
metadata:
  name: "00000000003075575260129686e67ed4-917a-454a-8553-d55fc4030f81"
  annotations:
    cpaas.io/secret-ref: developer.test.cn-xfd8x (1)
  labels:
    cluster.cpaas.io/name: global
    project.cpaas.io/name: cong
spec:
  name: developer.test.cn
  kind: full
```

1. Если включены сертификаты, необходимо заранее создать Secret типа LTS. **secret-ref** — это имя секрета.

Веб-консоль

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > Domain Names**.
3. Нажмите **Create Domain Name**.
4. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Type	• Domain: Полное доменное имя, например, developer.test.cn. • Wildcard Domain: Подстановочный домен с символом подстановки (*), например, *.test.cn, который включает все поддомены домена test.cn.
Domain	Введите полное доменное имя или суффикс домена в зависимости от выбранного типа доменного имени.
Allocate Cluster	Если выделяется кластер, необходимо также выбрать проект, связанный с выделенным кластером, например, все проекты, связанные с кластером.
Certificate	Включает публичный ключ (tls.crt) и приватный ключ (tls.key) для создания сертификата, привязанного к доменному имени. Проект, которому выделен сертификат, совпадает с проектом привязанного доменного имени. Примечания: • Импорт бинарных файлов не поддерживается. • Привязанный сертификат должен соответствовать условиям корректного формата, быть в пределах срока действия и подписан для доменного имени и т.д. • После создания привязанного сертификата формат имени сертификата: доменное имя - случайные символы. • После создания привязанного сертификата он отображается в списке сертификатов, но обновление и удаление привязанного сертификата поддерживается только на странице деталей домена. • После создания привязанного сертификата поддерживается обновление содержимого сертификата, но замена на другой сертификат не поддерживается.

5. Нажмите **Create**.

CLI

```
kubectl apply -f test-domain.yaml
```

Последующие действия

- **Регистрация домена**: Зарегистрируйте домен, если созданный домен еще не зарегистрирован.
- **Разрешение домена**: Выполните разрешение домена, если домен не указывает на адрес балансировщика нагрузки кластера платформы.

Создание сертификатов

После того как администратор платформы импортирует TLS-сертификат и назначит его определённому проекту, разработчики с соответствующими правами на проект смогут использовать сертификат, импортированный и назначенный администратором платформы, при работе с входящими правилами и функционалом балансировки нагрузки. В дальнейшем, например, при истечении срока действия сертификата, администратор платформы сможет централизованно обновить сертификат.

Внимание: Функциональность сертификатов в настоящее время не поддерживается для использования в кластерах публичного облака. При необходимости вы можете создавать секреты типа TLS в указанном **namespace**.

Чтобы создать сертификат:

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > Certificates**.
3. Нажмите **Create Certificate**.
4. Следуйте инструкциям ниже для настройки соответствующих параметров.

Параметр	Описание
Assign Project	All Projects: Назначить сертификат для использования во всех проектах, связанных с текущим кластером. Specified Project: Назначить сертификат для использования в указанном проекте. No Assignment: Пока не назначать проект. После создания сертификата вы сможете обновить проекты, которые могут использовать сертификат, через операцию Update Project .
Public Key	Это tls.crt. При импорте публичного ключа бинарные файлы не поддерживаются.
Private Key	Это tls.key. При импорте приватного ключа бинарные файлы не поддерживаются.

5. Нажмите **Create**.

Создание пула внешних IP-адресов

Пул внешних IP-адресов — это набор IP-адресов, который MetalLB использует для получения внешних IP-адресов доступа для внутренних маршрутов типа LoadBalancer.

Прerequisites

Если необходимо использовать пул внешних IP-адресов типа BGP, обратитесь к администратору для включения соответствующих функций.

Ограничения и условия

IP-ресурсы для внешнего адреса должны соответствовать следующим условиям:

- Пул внешних адресов должен быть связан на уровне канального уровня (L2) с доступными узлами.
- IP-адреса должны быть пригодны для использования платформой и не могут включать IP-адреса, уже используемые физической сетью, например, IP-адреса шлюзов.
- Не должно быть пересечений с сетями, используемыми кластером, включая Cluster CIDR, Service CIDR, подсети и т.д.
- В среде с двойным стеком необходимо, чтобы в одном пуле внешних адресов одновременно присутствовали как IPv4, так и IPv6 адреса, и их количество было больше 0. В противном случае внутренние маршруты типа LoadBalancer с двойным стеком не смогут получить внешние адреса доступа.
- В среде IPv6 DNS узлов должен поддерживать IPv6, иначе плагин MetalLB не сможет быть успешно развернут.

Развёртывание плагина MetalLB

Использование пула внешних адресов зависит от плагина MetalLB.

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Marketplace > Cluster Plugin**.
3. Найдите MetalLB, нажмите на **MetalLB** справа от : > **Deploy**.
4. Дождитесь, пока статус развёртывания не изменится на **Deployment Successful**, чтобы завершить процесс.

Пример ресурса custom resource (CR) IPAddressPool

```
# ippool-with-L2advertisement.yaml
kind: IPAddressPool
apiVersion: metallb.io/v1beta1
metadata:
  name: test-ippool
  namespace: metallb-system
spec:
  addresses:
    - 13.1.1.1/24
  avoidBuggyIPs: true
---
kind: L2Advertisement
apiVersion: metallb.io/v1beta1
metadata:
  name: test-ippool
  namespace: metallb-system
spec:
  ipAddressPools:
```

- test-ippool (1)

nodeSelectors:

- matchLabels: {}

matchExpressions:

- key: kubernetes.io/hostname

operator: In

values:

- 192.168.66.210

Режим BGP:

```
# ippool-with-bgpadvertisement.yaml
kind: IPAddressPool
apiVersion: metallb.io/v1beta1
metadata:
  name: test-pool-bgp
  namespace: metallb-system
spec:
  addresses:
    - 4.4.4.3/23
  avoidBuggyIPs: true
---
kind: BGPAdvertisement
apiVersion: metallb.io/v1beta1
metadata:
  name: test-pool-bgp
  namespace: metallb-system
spec:
  ipAddressPools:
    - test-pool-bgp
  nodeSelectors:
    - matchLabels:
        alertmanager: "true"
  peers:
    - test-bgp-example
```

1. Ссылка на пул IP-адресов.

Создание пула внешних IP-адресов через веб-консоль

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > External IP Address Pool**.
3. Нажмите Create External IP Address Pool.
4. Следуйте инструкциям для настройки параметров.

Параметр	Описание
Type	- L2: Коммуникация и пересылка на основе MAC-адресов, подходит для небольших или локальных сетей, требующих простой и быстрой коммутации на уровне 2, с преимуществами в простоте настройки и низкой задержке. - BGP (Alpha): Маршрутизация и пересылка на основе IP-адресов, использующая протокол BGP для обмена маршрутной информацией, подходит для крупных сетей с необходимостью сложной маршрутизации между несколькими автономными системами, с преимуществами в высокой масштабируемости и надежности.
IP Resources	Поддерживается ввод в форматах CIDR и диапазона IP. Нажмите Add для добавления нескольких записей, примеры: CIDR: 192.168.1.1/24. Диапазон IP: 192.168.2.1 ~ 192.168.2.255.
Available Nodes	В режиме L2 доступные узлы — это те, которые используются для передачи всего трафика VIP; в режиме BGP доступные узлы — это те, которые несут VIP, устанавливают BGP-соединения с пирами и объявляют маршруты во внешнюю сеть. Имя узла: выбор доступных узлов по имени. Выбор по меткам: выбор доступных узлов по меткам. Показать детали узлов: просмотр итогового списка доступных узлов. Примечание: - При использовании типа BGP доступные узлы — это узлы следующего перехода; убедитесь, что выбранные доступные узлы являются подмножеством BGP Connection Nodes. - Можно настроить либо выбор по меткам, либо по имени узла; если заданы оба варианта, итоговые доступные узлы — пересечение обоих.
BGP Peers	Выберите BGP пиры; подробности настройки смотрите в разделе BGP Peers.

5. Нажмите **Create**.

Создание пула внешних IP-адресов через CLI

```
kubectl apply -f ippool-with-L2advertisement.yaml -f ippool-with-bgpadvertisement.yaml
```

Просмотр политики оповещений

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > External IP Address Pool**.

3. Нажмите **View Alarm Policy** в правом верхнем углу страницы, чтобы просмотреть общую политику оповещений для MetalLB.

Создание BGP-пиров

Узлы, которые устанавливают соединения для обмена маршрутной информацией либо между разными AS, либо внутри одного AS, и которые обмениваются данными по протоколу BGP.

Терминология

Термин	Объяснение
AS Number	AS — это совокупность маршрутизаторов, управляемых одной технической административной организацией и использующих единую политику маршрутизации. Каждому AS в сети BGP присваивается уникальный номер AS для отличия от других AS. Номера AS делятся на 2-байтовые и 4-байтовые. • Диапазон 2-байтовых номеров AS — 1~65535, где 1~64511 — зарегистрированные публичные номера AS в Интернете, аналогичные публичным IP-адресам; 64512~65535 — приватные номера AS, аналогичные приватным IP-адресам. • Диапазон 4-байтовых номеров AS — 1~4294967295. Устройства, поддерживающие 4-байтовые номера AS, совместимы с устройствами, поддерживающими 2-байтовые номера AS.

Прerequisites

Пожалуйста, свяжитесь с администратором для включения соответствующих функций.

Пример пользовательского ресурса BGPPeer (CR)

```
# test-bgb-example.yaml
apiVersion: metallb.io/v1beta2
kind: BGPPeer
metadata:
  name: example
  namespace: metallb-system
spec:
  myASN: 64512
  peerASN: 64512
  peerAddress: 172.30.0.3
  peerPort: 180
  nodeSelectors:
    - matchLabels:
        alertmanager: "true"
```

Веб-консоль

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > BGP Peers**.
3. Нажмите **Create BGP Peer**.
4. Следуйте инструкциям ниже для настройки параметров.

Параметр	Описание
Local AS Number	Номер AS, к которому принадлежит узел, устанавливающий BGP-соединение. Внимание: Если нет особых требований, рекомендуется использовать конфигурацию IBGP, то есть локальный номер AS должен совпадать с номером AS пира.
Peer AS Number	Номер AS, к которому принадлежит BGP-пир.
Peer IP	IP-адрес BGP-пира, который должен быть действительным IP-адресом, способным установить BGP-соединение.
Local IP	IP-адрес узла, устанавливающего BGP-соединение. Если у узла несколько IP-адресов, выберите конкретный локальный IP для установления BGP-соединения с пиром.
Peer Port	Номер порта BGP-пира.
BGP Connected Node	Узел, который устанавливает BGP-соединение. Если этот параметр не задан, BGP-соединения будут устанавливаться со всеми узлами.
eBGP Multi-Hop	Позволяет устанавливать BGP-сессии между BGP-маршрутизаторами, которые не связаны напрямую. При включении этой функции значение TTL по умолчанию для BGP-пакетов равно 5, что позволяет устанавливать пиринговые отношения BGP через несколько промежуточных сетевых устройств, делая дизайн сети более гибким.
RouterID	32-битное числовое значение (обычно представлено в формате с точками, аналогично формату IPv4-адреса), используемое для уникальной идентификации BGP-маршрутизатора в сети BGP. Обычно применяется для установления соседских отношений BGP, обнаружения петель маршрутизации, выбора оптимальных путей и устранения неполадок в сети.

5. Нажмите **Create**.

CLI

```
kubectl apply -f test-bgb-example.yaml
```

Настройка подсетей

Правила выделения IP

Внимание: Если проекту или пространству имён назначено несколько подсетей, IP-адрес будет случайным образом выбран из одной из подсетей.

- Выделение для проекта:
 - Если проект не привязан к подсети, Pods во всех пространствах имён этого проекта могут использовать IP-адреса только из подсети по умолчанию. Если в подсети по умолчанию недостаточно IP-адресов, Pods не смогут запуститься.
 - Если проект привязан к подсети, Pods во всех пространствах имён этого проекта могут использовать IP-адреса только из этой конкретной подсети.
- Выделение для пространства имён:
 - Если пространство имён не привязано к подсети, Pods в этом пространстве имён могут использовать IP-адреса только из подсети по умолчанию. Если в подсети по умолчанию недостаточно IP-адресов, Pods не смогут запуститься.
 - Если пространство имён привязано к подсети, Pods в этом пространстве имён могут использовать IP-адреса только из этой конкретной подсети.

Сеть Calico

Создание подсетей в сети Calico для достижения более тонкой изоляции ресурсов внутри кластера.

Ограничения и особенности

В среде кластера с IPv6 подсети, создаваемые в сети Calico, по умолчанию используют инкапсуляцию VXLAN. Порты, необходимые для инкапсуляции VXLAN, отличаются от портов для инкапсуляции IPIP. Необходимо убедиться, что открыт UDP порт 4789.

Пример custom resource (CR) подсети с сетью Calico

```
# test-calico-subnet.yaml
apiVersion: kubeovn.io/v1
kind: Subnet
metadata:
  name: test-calico
spec:
  cidrBlock: 10.1.1.1/24
  default: false (1)
  ipipMode: Always (2)
  natOutgoing: true (3)
  private: false
  protocol: Dual
  v4blockSize: 30
```

1. При **default** true используется инкапсуляция VXLAN.
2. См. параметры Encapsulation Mode и Encapsulation Protocol.

3. См. параметры Outbound Traffic NAT.

Создание подсети в сети Calico через веб-консоль

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > Subnets**.
3. Нажмите **Create Subnet**.
4. Настройте параметры согласно следующим инструкциям.

Параметр	Описание
CIDR	После назначения подсети проекту или пространству имён, контейнерные группы в этом пространстве будут случайным образом использовать IP-адреса из этого CIDR для связи.
Encapsulation Protocol	Выберите протокол инкапсуляции. IPIP не поддерживается в режиме dual-stack. • IPIP: реализует межсегментную связь с помощью протокола IPIP. • VXLAN (Alpha): реализует межсегментную связь с помощью протокола VXLAN. • No Encapsulation: прямое соединение через маршрутизацию.
Encapsulation Mode	При протоколе инкапсуляции IPIP или VXLAN необходимо указать режим инкапсуляции, по умолчанию Always . • Always: всегда включать туннели IPIP / VXLAN. • Cross Subnet: включать туннели IPIP / VXLAN только если хосты находятся в разных подсетях; при нахождении в одной подсети — прямое соединение через маршрутизацию.
Outbound Traffic NAT	Выберите, включать ли NAT исходящего трафика (Network Address Translation), по умолчанию включено. Используется для установки адреса доступа, видимого из внешней сети, когда контейнерные группы подсети выходят в интернет. При включенном NAT исходящего трафика в качестве адреса доступа используется IP хоста; при отключённом — IP контейнерных групп подсети напрямую видны во внешней сети.

5. Нажмите **Confirm**.
6. На странице с деталями подсети выберите **Actions > Allocate Project / Allocate Namespace**.
7. Завершите настройку и нажмите **Allocate**.

Создание подсети в сети Calico через CLI

```
kubectl apply -f test-calico-subnet.yaml
```

Reference Content

Динамическое соответствие между CIDR и blockSize показано в таблице ниже.

CIDR	blockSize Size	Количество хостов	Размер одного пула IP
prefix<=16	26	1024+	64
16<prefix<=19	27	256~1024	32
prefix=20	28	256	16
prefix=21	29	256	8
prefix=22	30	256	4
prefix=23	30	128	4
prefix=24	30	64	4
prefix=25	30	32	4
prefix=26	31	32	2
prefix=27	31	16	2
prefix=28	31	8	2
prefix=29	31	4	2
prefix=30	31	2	2
prefix=31	31	1	2

Внимание: Подсети с префиксом больше 31 не поддерживаются.

Сеть Kube-OVN

Создание подсети в Overlay-сети Kube-OVN для более тонкой изоляции ресурсов в кластере.

Внимание: Платформа имеет встроенную подсеть **join** для связи между нодами и Pods; избегайте конфликтов сетевых сегментов между **join** и вновь создаваемыми подсетями.

Пример custom resource (CR) подсети с Overlay-сетью Kube-OVN

```
# test-overlay-subnet.yaml
apiVersion: kubeovn.io/v1
kind: Subnet
metadata:
  name: test-overlay-subnet
spec:
  default: false
  protocol: Dual
  cidrBlock: 10.1.0.0/23
  natOutgoing: true (1)
```

excludelips: (2)

- 10.1.1.2

gatewayType: distributed (3)

gatewayNode: "" (4)

private: false

enableEcmp: false(5)

1. См. параметры Outbound Traffic NAT.
2. См. параметры Reserved IP.
3. См. параметры Gateway Type. Доступные значения: **distributed** или **centralized**.
4. См. параметры Gateway Nodes.
5. См. параметры ECMP. Требуется предварительно связаться с администратором для включения feature gate.

Создание подсети в Overlay-сети Kube-OVN через веб-консоль

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > Subnet**.
3. Нажмите **Create Subnet**.
4. Настройте параметры согласно следующим инструкциям.

Параметр	Описание
Network Segment	После назначения подсети проекту или пространству имён IP из этого сегмента будут случайным образом выделяться для использования Pods.
Reserved IP	Указанные зарезервированные IP не будут автоматически выделяться. Например, могут использоваться как фиксированные IP для вычислительных компонентов.
Gateway Type	Выберите тип шлюза для подсети для управления исходящим трафиком. - Distributed : каждый хост в кластере может выступать в роли исходящего узла для Pods на текущем хосте, обеспечивая распределённый выход в сеть. - Centralized : все Pods кластера используют один или несколько конкретных хостов в качестве исходящих узлов, что облегчает внешний аудит и контроль межсетевого экрана. Настройка нескольких централизованных gateway nodes обеспечивает высокую доступность.
ECMP (Alpha)	При выборе Centralized gateway можно использовать функцию ECMP. По умолчанию шлюз работает в режиме мастер-слейв, только мастер-шлюз обрабатывает трафик. При включении ECMP (Equal-Cost Multipath Routing) исходящий трафик маршрутизируется по нескольким равнозначным путям ко всем доступным узлам шлюза, увеличивая суммарную пропускную способность шлюза. Примечание: необходимо предварительно включить соответствующие функции.

Параметр	Описание
Gateway Nodes	При использовании Centralized gateway выберите один или несколько конкретных хостов в качестве узлов шлюза.
Outbound Traffic NAT	Выберите, включать ли NAT исходящего трафика (Network Address Translation). По умолчанию включено. Используется для установки адреса доступа, видимого из внешней сети, когда Pods подсети выходят в интернет. При включенном NAT исходящего трафика в качестве адреса доступа используется IP хоста; при отключённом — IP Pods подсети напрямую видны во внешней сети. В этом случае рекомендуется использовать централизованный шлюз.

5. Нажмите **Confirm**.
6. На странице с деталями подсети выберите **Actions > Allocate Project / Namespace**.
7. Завершите настройку и нажмите **Allocate**.

Создание подсети в Overlay-сети Kube-OVN через CLI

```
kubectrl apply -f test-overlay-subnet.yaml
```

Underlay-сеть

Создание подсетей в Underlay-сети Kube-OVN не только обеспечивает более тонкую изоляцию ресурсов, но и улучшает производительность.

Внимание: Сетевая инфраструктура контейнеров в Kube-OVN Underlay требует поддержки физической сети.

Инструкция по использованию

Общий процесс создания подсетей в Underlay-сети Kube-OVN: Добавить мостовую сеть > Добавить VLAN > Создать подсеть.

1. Имя сетевой карты по умолчанию.
2. Настройка сетевой карты по узлам.

Добавление bridge network через CLI

Bridge network — это мост, после привязки сетевой карты она может передавать трафик контейнерной сети, обеспечивая связь с физической сетью.

```
kubectrl apply -f test-provider-network.yaml
```

Добавление VLAN через CLI

```
kubectrl apply -f test-vlan.yaml
```

Пример custom resource (CR) подсети с Underlay-сетью Kube-OVN

```
# test-underlay-network.yaml
apiVersion: kubeovn.io/v1
kind: Subnet
metadata:
```

```
name: test-underlay-network
spec:
  default: false
  protocol: Dual
  cidrBlock: 11.1.0.0/23
  gateway: 11.1.0.1
  excludeIps:
    - 11.1.0.3
  private: false
  allowSubnets: []
  vlan: test-vlan (1)
  enableEcmp: false
```

1. Ссылка на VLAN.

Создание подсети в Underlay-сети Kube-OVN через веб-консоль

Внимание: Платформа преднастроила подсеть **join** для связи между нодами и Pods в режиме Overlay. Эта подсеть не используется в режиме Underlay, поэтому важно избегать конфликтов IP-сегментов между **join** и другими подсетями.

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > Subnet**.
3. Нажмите **Create Subnet**.
4. Настройте параметры согласно следующим инструкциям.

Параметр	Описание
VLAN	VLAN, к которой принадлежит подсеть.
Subnet	После назначения подсети проекту или пространству имён IP из физической подсети будут случайным образом выделяться для использования Pods.
Gateway	Физический шлюз в указанной подсети.
Reserved IP	Указанные зарезервированные IP не будут автоматически назначаться. Например, могут использоваться как фиксированные IP вычислительных компонентов.

5. Нажмите **Confirm**.
6. На странице с деталями подсети выберите **Action > Assign Project / Namespace**.
7. Завершите настройку и нажмите **Assign**.

Создание подсети в Underlay-сети Kube-OVN через CLI

```
kubectl apply -f test-underlay-network.yaml
```

Связанные операции

Если в кластере одновременно существуют подсети Underlay и Overlay, при необходимости можно настроить Автоматическую взаимосвязь между подсетями Underlay и Overlay.

Управление подсетями

Обновление шлюза

Веб-консоль

Включает изменение способа исходящего трафика, узлов шлюза и конфигурации NAT.

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Network Management > Subnets**.
3. Нажмите на имя подсети.
4. Выберите **Action > Update Gateway**.
5. Обновите параметры.
6. Нажмите **OK**.

CLI

```
kubectrl patch subnet test-overlay-subnet --type=json -p='[
  {"op": "replace", "path": "/spec/gatewayType", "value": "centralized"},
  {"op": "replace", "path": "/spec/gatewayNode", "value": "192.168.66.210"},
  {"op": "replace", "path": "/spec/natOutgoing", "value": true},
  {"op": "replace", "path": "/spec/enableEcmp", "value": true}
]'
```

Обновление зарезервированных IP

Веб-консоль

IP шлюза нельзя удалить из зарезервированных IP, остальные зарезервированные IP можно редактировать, удалять или добавлять.

1. Перейдите в Platform Management.
2. В левой панели выберите **Network Management > Subnets**.
3. Нажмите на имя подсети.
4. Выберите Action > Update Reserved IP.
5. После внесения изменений нажмите **Update**.

CLI

```
kubectrl patch subnet test-overlay-subnet --type=json -p='[
  {
    "op": "replace",
    "path": "/spec/excludeIps",
    "value": ["10.1.0.1", "10.1.1.2", "10.1.1.4"]
  }
]'
```

Назначение проектов

Веб-консоль

Назначение подсетей конкретным проектам помогает командам лучше управлять и изолировать сетевой трафик для разных проектов, обеспечивая необходимые сетевые ресурсы для каждого проекта.

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Network Management > Subnets**.
3. Нажмите на имя подсети.
4. Выберите Action > Assign Project.
5. После добавления или удаления проектов нажмите **Assign**.

CLI

```
kubectrl patch subnet test-overlay-subnet --type=json -p='[
{
  "op": "replace",
  "path": "/spec/namespaceSelectors",
  "value": [
    {
      "matchLabels": {
        "cpaas.io/project": "cong"
      }
    }
  ]
}'
```

Назначение пространств имён

Веб-консоль

Назначение подсетей конкретным пространствам имён позволяет добиться более тонкой сетевой изоляции.

Внимание: Процесс назначения приведёт к перестройке шлюза, и исходящие пакеты будут отброшены! Пожалуйста, убедитесь, что в данный момент нет бизнес-приложений, обращающихся к внешним кластерам.

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Network Management > Subnets**.
3. Нажмите на имя подсети.
4. Выберите **Action > Assign Namespace**.
5. После добавления или удаления пространств имён нажмите **Assign**.

CLI

```
kubectrl patch subnet test-overlay-subnet --type=json -p='[
{
  "op": "replace",
```



```

    "path": "/spec/namespaces",
    "value": ["cert-manager"]
  }
]'

```

Расширение подсетей

Веб-консоль

Когда диапазон зарезервированных IP подсети достигает предела использования или близок к исчерпанию, его можно расширить на основе исходного диапазона подсети без влияния на нормальную работу существующих сервисов.

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Network Management > Subnets**.
3. Нажмите на имя подсети.
4. Выберите **Action > Expand Subnet**.
5. Завершите настройку и нажмите **Update**.

CLI

```

kubectl patch subnet test-overlay-subnet --type=json -p='[
  {
    "op": "replace",
    "path": "/spec/cidrBlock",
    "value": "10.1.0.0/22"
  }
]'

```

Управление сетями Calico

Поддерживается назначение проектов и пространств имён.

Удаление подсети

Веб-консоль

Обратите внимание!

- При удалении подсети, если есть контейнерные группы, использующие IP из этой подсети, они смогут продолжать работу с сохранением IP, но не смогут обмениваться сетевым трафиком. Контейнерные группы можно пересоздать для использования IP из подсети по умолчанию или назначить новую подсеть для пространства имён, в котором они находятся.
 - Подсеть по умолчанию удалить нельзя.
1. Перейдите в **Platform Management**.
 2. В левой навигационной панели выберите **Network Management > Subnets**.
 3. Нажмите **: > Delete** и подтвердите удаление.

CLI

```
kubectl delete subnet test-overlay-subnet
```

Создание сетевых политик

NetworkPolicy — это ресурс Kubernetes с областью видимости в пределах namespace, реализуемый плагинами CNI.

С помощью сетевых политик можно контролировать сетевой трафик Pod'ов, обеспечивая сетевую изоляцию и снижая риск атак.

По умолчанию все Pod'ы могут свободно обмениваться трафиком, разрешая входящий и исходящий трафик из любых источников.

При применении NetworkPolicy целевые Pod'ы будут принимать только трафик, соответствующий спецификации.

Внимание: Сетевые политики применяются только к трафику контейнеров. Они не влияют на Pod'ы, работающие в режиме **hostNetwork**.

Пример NetworkPolicy

```
# example-network-policy.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: example
  namespace: demo-1
  annotations:
    cpaas.io/display-name: test
spec:
  podSelector:
    matchLabels:
      pod-template-hash: 55c84b59bb
  ingress:
    - ports:
        - protocol: TCP
          port: 8989
      from: (1)
        - podSelector:
            matchLabels:
              kubevirt.io/vm: test
  egress:
    - ports:
        - protocol: TCP
          port: 80
      to:
        - ipBlock:
```

cidr: 192.168.66.221/23

except: []

policyTypes:

- Ingress
- Egress

1. from и to поддерживают namespaceSelector, podSelector, ipBlock

Веб-консоль

1. Войдите в **Container Platform**.
2. В левой навигационной панели выберите **Network > Network Policies**.
3. Нажмите **Create Network Policy**.
4. Следуйте инструкциям ниже для заполнения соответствующих параметров.

Область	Параметр		Описание
Целевой Pod	Выбор Pod		Введите метки целевых Pod в формате ключ-значение; если не указано, политика применяется ко всем Pod в текущем namespace.
	Предварительный просмотр целевых Pod, затронутых текущей политикой		Нажмите Preview , чтобы увидеть целевые Pod, на которые влияет данная сетевая политика.
Ingress	Блокировать весь входящий трафик		Блокирует весь входящий трафик к целевому Pod. Примечание: • Если в поле spec.policyTypes в YAML добавлен Ingress без настройки конкретных правил, при возврате к форме опция Блокировать весь входящий трафик будет автоматически установлена. • Если одновременно удалить поля spec.ingress, spec.egress и spec.policyTypes в YAML, при возврате к форме опция Блокировать весь входящий трафик также будет автоматически установлена.
	Правила Описание: Если в правилах	Pod'ы в текущем namespace	Соответствуют Pod'ам с указанными метками в текущем namespace; только такие Pod могут обращаться к целевому Pod. Можно нажать Preview , чтобы увидеть Pod'ы, затронутые текущим правилом. Если этот параметр не настроен, по умолчанию все

Область	Параметр		Описание
	добавлено несколько источников, между ними действует логическое или .		Pod'ы в текущем namespace имеют доступ к целевому Pod.
		Pod'ы в текущем кластере	Соответствуют namespace или Pod'ам с указанными метками в кластере; только такие Pod могут обращаться к целевому Pod. Можно нажать Preview, чтобы увидеть Pod'ы, затронутые текущим правилом. - Если настроены и селекторы namespace, и Pod, будет взято пересечение — то есть выбраны Pod с указанными метками из указанного namespace. - Если этот параметр не настроен, по умолчанию все Pod'ы из всех namespace в кластере имеют доступ к целевому Pod.
		IP-диапазон	Введите CIDR, который может обращаться к целевому Pod, и можно исключить CIDR-диапазоны, которым доступ запрещён. Если этот параметр не настроен, любой трафик может обращаться к целевому Pod. Описание: Можно добавить исключения в формате <i>example_ip/32</i> для исключения отдельного IP-адреса.
	Порт		Соответствует трафику на указанных протоколах и портах; можно добавить числовые порты или имена портов на Pod'ах. Если этот параметр не настроен, будут соответствовать все порты.
Egress	Блокировать весь исходящий трафик		Блокирует весь исходящий трафик от целевого Pod. Примечание: Если в поле <code>spec.policyTypes</code> в YAML добавлен Egress без настройки конкретных правил, при возврате к форме опция Блокировать весь исходящий трафик будет автоматически установлена.
	Другие параметры		Аналогично параметрам Ingress , здесь не приводится подробное описание.

5. Нажмите **Create**.

CLI

```
kubectl apply -f example-network-policy.yaml
```

Настройка сетевых политик кластера

Сетевые политики кластера отвечают за управление правилами контроля доступа **на уровне проектов**. При включении этой функции разные проекты по умолчанию изолируются друг от друга, и вычислительные компоненты в разных проектах не могут взаимодействовать по сети. Связь может быть организована путем добавления правил **доступа одного проекта** или **доступа по IP-сегменту**.

После настройки сетевые политики кластера будут синхронизированы с пространствами имён в кластере и могут быть просмотрены в модуле функции **Network Policies** платформы контейнеров.

Примечания

- Эффективность сетевых политик кластера зависит от того, поддерживает ли используемый в кластере сетевой плагин сетевые политики.
 - Kube-OVN и Calico поддерживают сетевые политики.
 - Flannel не поддерживает сетевые политики.
 - При доступе к кластеру или использовании кастомного сетевого плагина можно обратиться к соответствующей документации для подтверждения поддержки.
- Функциональность находится на уровне Alpha в режиме сети Kube-OVN.

Чтобы настроить сетевые политики кластера:

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Network Management > Cluster Network Policies**.
3. Нажмите Configure Now.
4. Следуйте инструкциям ниже для завершения соответствующей настройки.

Параметр настройки	Описание
Полная изоляция между проектами	Включение или отключение переключателя полной изоляции между проектами, который включён по умолчанию и может быть выключен нажатием. При включении достигается сетевая изоляция между всеми проектами текущего кластера, и другим ресурсам запрещён доступ к любому проекту внутри кластера (например, внешним IP, балансировщикам нагрузки). Это не влияет на доступ проектов к ресурсам вне кластера.
Доступ одного проекта	Параметр действует только при включённом переключателе Полная изоляция между проектами. Настройте исходный проект и целевой проект для однонаправленного доступа. Нажмите Add для добавления записи конфигурации, поддерживается несколько записей. В выпадающем списке исходный проект выберите проект, который

Параметр настройки	Описание
	будет иметь доступ к целевому проекту, или выберите все проекты; в списке целевой проект выберите проект, к которому будет осуществлён доступ.
Доступ по IP-сегменту	Параметр действует только при включённом переключателе Полная изоляция между проектами. Настройте конкретный IP/сегмент и целевой проект для однонаправленного доступа. Нажмите Add для добавления записи конфигурации, поддерживается несколько записей. В поле ввода исходный IP-сегмент введите IP или CIDR-сегмент, который будет иметь доступ к целевому проекту; в списке целевой проект выберите проект, к которому будет осуществлён доступ.

5. Нажмите **Configure**.

Подготовка физической сети Kube-OVN Underlay

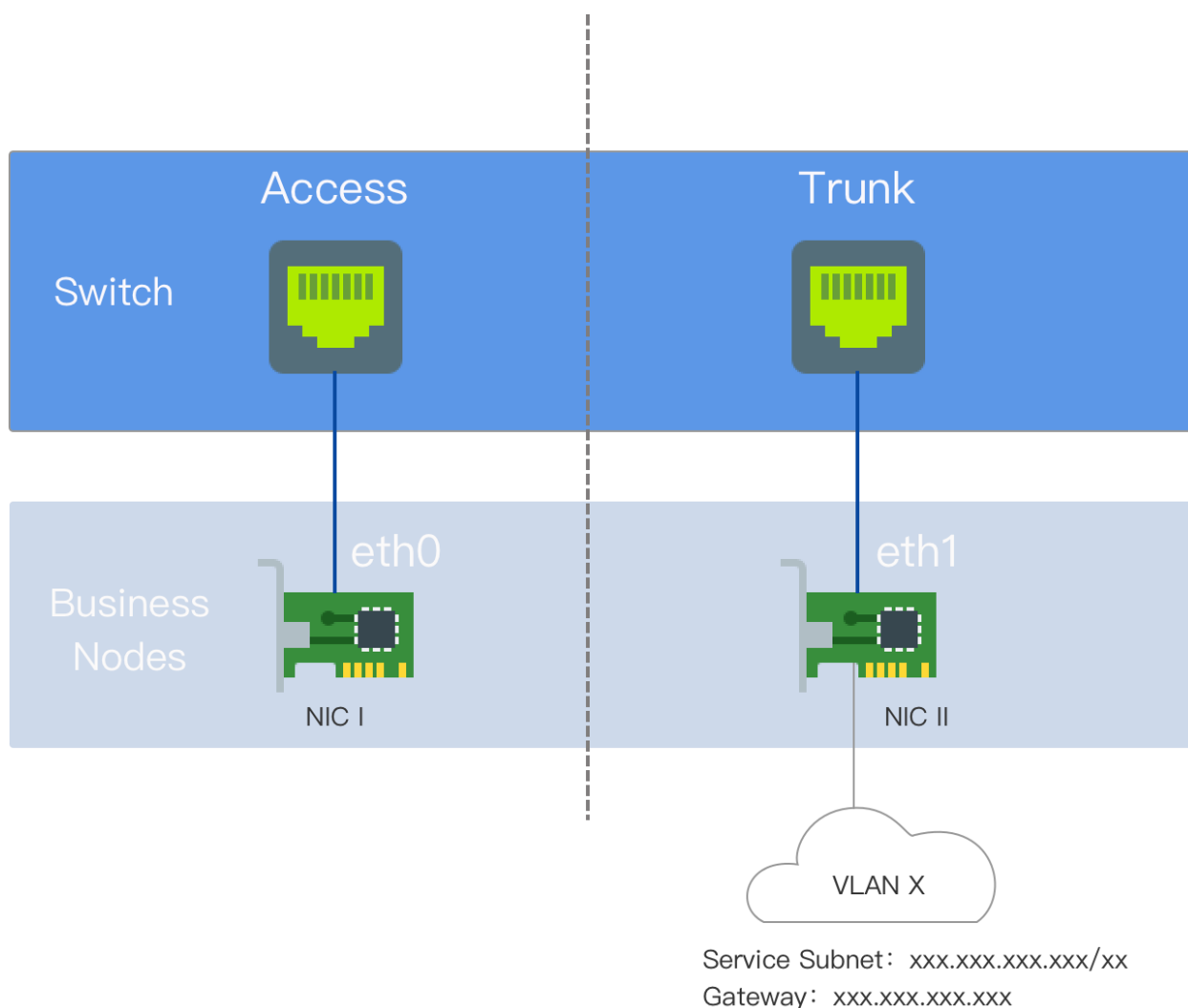
Сетевая инфраструктура контейнеров в режиме передачи Kube-OVN Underlay зависит от поддержки физической сети. Перед развёртыванием сети Kube-OVN Underlay необходимо совместно с сетевым администратором спланировать и выполнить соответствующие настройки физической сети заранее, чтобы обеспечить сетевую связность.

Инструкции по использованию

Kube-OVN Underlay требует развёртывания с несколькими сетевыми интерфейсными картами (NIC), при этом подсеть Underlay должна использовать исключительно один NIC. Другие типы трафика, например SSH, не должны использовать этот NIC и должны работать через другие NIC.

Перед использованием убедитесь, что на сервере узла имеется как минимум **двухсетевой NIC**, рекомендуется, чтобы скорость NIC была **не менее 10 Гбит/с или выше** (например, 10 Гбит/с, 25 Гбит/с, 40 Гбит/с).

- **NIC Один:** NIC с маршрутом по умолчанию, настроенный с IP-адресом, соединённый с внешним интерфейсом коммутатора, который настроен в режиме Access.
- **NIC Два:** NIC без маршрута по умолчанию и без настроенного IP-адреса, соединённый с внешним интерфейсом коммутатора, который настроен в режиме Trunk. Подсеть Underlay использует исключительно NIC Два.



Терминология

VLAN (Virtual Local Area Network) — это технология, которая логически разделяет локальную сеть на несколько сегментов (или меньших LAN) для облегчения обмена данными между виртуальными рабочими группами.

Появление технологии VLAN позволяет администраторам логически сегментировать различных пользователей в одной физической локальной сети на отдельные домены широковещания в соответствии с реальными потребностями приложений. Каждый VLAN состоит из группы рабочих станций с похожими требованиями и обладает теми же свойствами, что и физически сформированная LAN. Поскольку VLAN делятся логически, а не физически, рабочие станции в одном VLAN не ограничены одной физической областью; они могут находиться в разных физических сегментах LAN.

Основные преимущества VLAN включают:

- Сегментация портов. Даже на одном коммутаторе порты в разных VLAN не могут взаимодействовать друг с другом. Физический коммутатор может функционировать как несколько логических коммутаторов. Это часто используется для контроля взаимного доступа между разными отделами и площадками в сети.
- Безопасность сети. Разные VLAN не могут напрямую обмениваться данными, что исключает небезопасность широковещательной информации. Широковещательный

и одноадресный трафик внутри VLAN не будет передаваться в другие VLAN, что помогает контролировать трафик, снижать затраты на оборудование, упрощать управление сетью и повышать безопасность сети.

- Гибкое управление. При изменении сетевой принадлежности пользователя нет необходимости менять порты или кабели; достаточно изменить конфигурацию программно.

Требования к среде

В режиме Underlay Kube-OVN связывает физический NIC с OVS и отправляет пакеты напрямую во внешнюю сеть через этот физический NIC. Возможности L2/L3 пересылки зависят от базовых сетевых устройств. Соответствующие шлюз, VLAN и политики безопасности должны быть предварительно настроены на базовых сетевых устройствах.

Требования к сетевой конфигурации

- Kube-OVN проверяет доступность шлюза с помощью протокола ICMP при запуске контейнеров; базовый шлюз должен отвечать на ICMP-запросы.
- Для трафика доступа к сервисам Pods сначала отправляют пакеты на шлюз, который должен иметь возможность пересылать пакеты обратно в локальную подсеть.
- Если на коммутаторе или мосту включена функция Hairpin, **Hairpin должен быть отключён**. При использовании среды виртуальных машин VMware необходимо установить **Net.ReversePathFwdCheckPromisc** на хосте VMware в значение **1**, тогда Hairpin отключать не нужно.
- NIC, используемый для мостирования, **не может** быть **Linux Bridge**.
- Режимы агрегации NIC поддерживают Mode 0 (balance-rr), Mode 1 (active-backup), Mode 4 (802.3ad), Mode 6 (balance-alb), рекомендуется использовать 0 или 1. Другие режимы агрегации не тестировались, используйте их с осторожностью.

Требования к конфигурации слоя IaaS (виртуализации)

Для среды виртуальных машин OpenStack необходимо отключить **PortSecurity** для соответствующего сетевого порта.

Для сети vSwitch VMware параметры **MAC Address Changes**, **Forged Transmits** и **Promiscuous Mode Operation** должны быть установлены в **Accept**.

Для публичных облаков, таких как AWS, GCE и Alibaba Cloud, сети в режиме Underlay не поддерживаются из-за отсутствия возможности задания MAC-адресов пользователем.

Пример конфигурации

В этом примере узлы — физические машины с двумя NIC. NIC Один — это NIC с маршрутом по умолчанию; NIC Два — NIC без маршрута по умолчанию и без настроенного IP-адреса, используемый исключительно для подсети Underlay. NIC Два соединён с внешним коммутатором.

- На стороне коммутатора интерфейс, подключённый к NIC Два, должен быть настроен в режиме Trunk с разрешением соответствующих VLAN.

- Настройте адрес шлюза подсети кластера на соответствующем интерфейсе vlan-interface. При необходимости двойного стека можно одновременно настроить IPv6-адрес шлюза.
- Если шлюз находится за файрволом, необходимо разрешить доступ от узлов к сети cluster-cidr.
- Конфигурация NIC сервера не требуется.

Конфигурация коммутатора

Настройка VLAN интерфейса:

```
#
interface Vlan-interface74
  ip address 192.168.74.254 255.255.255.0 //IPv4 адрес шлюза
  ipv6 address 2074::192:168:74:254/64 //IPv6 адрес шлюза
#
Настройка интерфейса, подключённого к NIC Два:
#
interface Ten-GigabitEthernet1/0/19
  port link mode bridge
  port link-type trunk // Настройка интерфейса в режим Trunk
  undo port trunk permit vlan 1
  port trunk permit vlan 74 // Разрешение прохождения соответствующего VLAN
#
```

Проверка сетевой связности

Проверьте, может ли NIC Два связаться с адресом шлюза:

```
ip link add ens224.74 link ens224 type vlan id 74 // Имя NIC — ens224, VLAN ID — 74
ip link set ens224.74 up
ip addr add 192.168.74.200/24 dev ens224.74 // Выберите тестовый адрес из
подсети Underlay, здесь 192.168.74.200/24
ping 192.168.74.254 // Если пинг до шлюза успешен, значит физическая среда
соответствует требованиям развертывания
ip addr del 192.168.74.200/24 dev ens224.74 // Удалите тестовый адрес после
проверки
ip link del ens224.74 // Удалите подинтерфейс после проверки
```

Автоматическое взаимное подключение подсетей Underlay и Overlay

Если в кластере присутствуют подсети как Underlay, так и Overlay, по умолчанию Pods в подсети Overlay могут обращаться к IP-адресам Pods в подсети Underlay через шлюз с использованием NAT. Однако Pods в подсети Underlay для доступа к Pods в подсети Overlay должны настроить маршрутизацию на узлах.

Для автоматического взаимного подключения подсетей Underlay и Overlay можно вручную изменить YAML-файл подсети Underlay. После настройки Kube-OVN также будет использоваться дополнительный IP Underlay для подключения подсети Underlay и логического маршрутизатора `ovn-cluster`, устанавливая соответствующие правила маршрутизации для обеспечения взаимосвязи.

Шаги

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Cluster Management > Resource Management**.
3. Введите **Subnet** для фильтрации объектов ресурсов.
4. Нажмите **: > Update** рядом с подсетью Underlay, которую необходимо изменить.
5. Измените YAML-файл, добавив поле **u2oInterconnection: true** в раздел **Spec**.
6. Нажмите **Update**.

Внимание: Существующие вычислительные компоненты в подсети Underlay необходимо пересоздать, чтобы изменения вступили в силу.

Настройка балансировщика нагрузки

Балансировщик нагрузки — это сервис, который распределяет трафик по контейнерным экземплярам. Используя функциональность балансировки нагрузки, он автоматически распределяет входящий трафик для вычислительных компонентов и перенаправляет его на контейнерные экземпляры этих компонентов. Балансировка нагрузки может повысить отказоустойчивость вычислительных компонентов, масштабировать внешнюю сервисную способность этих компонентов и улучшить доступность приложений.

Администраторы платформы могут создавать одноточечные или высокодоступные балансировщики нагрузки для любого кластера на платформе, а также централизованно управлять и распределять ресурсы балансировщиков нагрузки. Например, балансировка нагрузки может быть назначена проектам, обеспечивая использование балансировки только пользователями с соответствующими правами в проекте.

Пожалуйста, обратитесь к таблице ниже для объяснения связанных понятий в этом разделе.

Параметр	Описание
Load Balancer	Программное или аппаратное устройство, которое распределяет сетевые запросы между доступными узлами в кластере. Используемый на платформе балансировщик нагрузки — это программный балансировщик уровня 7.
VIP	Виртуальный IP-адрес (Virtual IP Address) — IP-адрес, который не соответствует конкретному компьютеру или конкретной сетевой карте. При использовании балансировщика нагрузки типа высокодоступности адрес доступа должен быть VIP.

Предварительные требования

Высокая доступность **Load Balancer** требует наличия VIP.

Пример ресурса ALB2 (CR)

```
# test-alb.yaml
apiVersion: crd.alauda.io/v2beta1
kind: ALB2
metadata:
  name: alb-demo
  namespace: cpaas-system
  annotations:
    cpaas.io/display-name: ""
spec:
  address: 192.168.66.215
  config:
    vip: (1)
    enableLbSvc: false
    lbSvcAnnotations: {}
  networkMode: host (2)
  enablePortProject: false (3)
  nodeSelector:
    cpu-model.node.kubevirt.io/Nehalem: "true"
  projects: (4)
    - ALL_ALL
  replicas: 1
  resources: (5)
    limits:
      cpu: 200m
      memory: 256Mi
    requests:
      cpu: 200m
      memory: 256Mi
  type: nginx
```

1. При **enableLbSvc** равном true будет создан внутренний сервис типа LoadBalancer для адреса доступа балансировщика нагрузки. **lbSvcAnnotations** — ссылка на конфигурацию аннотаций сервиса типа **LoadBalancer**.
2. Ознакомьтесь с конфигурацией Network Mode ниже.
3. Ознакомьтесь с методом распределения ресурсов ниже.
4. Ознакомьтесь с назначенным проектом ниже.
5. Ознакомьтесь с техническими характеристиками ниже.

Создание балансировщика нагрузки

Веб-консоль

1. Перейдите в **Platform Management**.
2. В левой боковой панели нажмите **Network Management > Load Balancer**.
3. Нажмите Create Load Balancer.
4. Следуйте инструкциям ниже для завершения настройки сети.

Параметр	Описание
Network Mode	• Host Network Mode: На одном узле разрешается развертывать только один реплика балансировщика нагрузки, при этом несколько сервисов используют один ALB, что обеспечивает лучшую производительность сети. • Container Network Mode: На одном узле можно развернуть несколько реплик балансировщика нагрузки, чтобы удовлетворить требования отдельных ALB для каждого сервиса, с немного меньшей производительностью сети.
Service and Annotations (Alpha)	• Service: При включении создается внутренний сервис типа LoadBalancer для адреса доступа балансировщика нагрузки. Перед использованием убедитесь, что текущий кластер поддерживает сервис типа LoadBalancer. Можно использовать встроенный сервис типа LoadBalancer платформы; при отключении необходимо настроить External Address Pool для балансировщика нагрузки. • Annotations: Используются для объявления конфигурации или возможностей внутреннего маршрутизации типа LoadBalancer
Access Address	Адрес доступа для балансировки нагрузки, то есть адрес сервиса инстанса балансировщика нагрузки. После успешного создания балансировщика к нему можно обращаться по этому адресу. • В режиме host network mode заполните согласно реальным условиям; это может быть доменное имя или IP-адрес (внутренний IP, внешний IP, VIP). • В режиме container network mode адрес будет получен автоматически.

5. Следуйте инструкциям ниже для завершения настройки ресурсов.

Параметр	Описание
Specification	Установите характеристики разумно в соответствии с бизнес-требованиями.

Параметр	Описание
Deployment Type	• Single Point: Контейнерная группа балансировщика нагрузки развёрнута на одном узле, что может привести к недоступности балансировщика при сбое машины. • High Availability: Несколько контейнерных групп балансировщика нагрузки развёрнуты на соответствующем количестве узлов, обычно 3. Это удовлетворяет потребности балансировки при больших объёмах бизнеса и обеспечивает возможности аварийного восстановления.
Replicas	Количество реплик — это количество контейнерных групп балансировщика нагрузки. Совет: Для обеспечения высокой доступности балансировщика рекомендуется количество реплик не менее 3.
Node Labels	Фильтрация узлов по меткам для развертывания балансировщика нагрузки. Совет: • Рекомендуется, чтобы количество узлов, удовлетворяющих требованиям, было больше количества реплик балансировщика. • Метка с одинаковым ключом может выбрать только одну (если выбрано несколько, подходящих хостов не будет).
Resource Allocation Method	• Instance: Любой порт в диапазоне 1-65535, на котором может слушать инстанс балансировщика, может быть предоставлен для использования проектом. • Port (Alpha): Можно выделять только порты в указанном диапазоне для использования проектом. Этот метод позволяет более тонко контролировать ресурсы при ограниченности портов.
Assigned Project	• При выборе Resource Allocation Method = Instance балансировщик может быть выделен всем проектам, связанным с текущим кластером, или указанным проектам. Во всех выделенных проектах все Pod во всех пространствах имён могут получать запросы, распределяемые балансировщиком. • All Projects: Выделяет балансировщик для использования всеми проектами, связанными с текущим кластером. • Specified Projects (Alpha): Нажмите выпадающий список под Specified Projects и отметьте чекбоксы слева от

Параметр	Описание
	названий проектов для выбора одного или нескольких проектов, выделяя балансировщик для использования этими проектами. Совет: Можно фильтровать проекты, вводя их имена в выпадающем списке. • No Allocation (Alpha): Временно не выделяет ни один проект. После создания балансировщика можно использовать операцию Update Project для обновления параметров выделения проекта. • При выборе Resource Allocation Method = Port этот пункт не нужно настраивать. Порты выделяются вручную после создания балансировщика.

6. Нажмите **Create**. Процесс создания займет некоторое время, пожалуйста, подождите.

CLI

```
kubectl apply -f test-alb.yaml -n cpaas-system
```

Обновление балансировщика нагрузки

Внимание: Обновление балансировщика вызовет прерывание сервиса на 3–5 минут. Пожалуйста, выбирайте подходящее время для этой операции!

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Network Management > Load Balancer**.
3. Нажмите : > **Update**.
4. Обновите сетевые и ресурсные настройки по необходимости.
 - Устанавливайте характеристики разумно в соответствии с бизнес-требованиями.
 - **Internal routing** поддерживает обновление только из состояния **Disabled** в состояние **Enabled**.
5. Нажмите **Update**.

Удаление балансировщика нагрузки

Веб-консоль

Внимание: После удаления балансировщика будут удалены связанные порты и правила, восстановить их будет невозможно.

1. Перейдите в Platform Management.
2. В левой навигационной панели нажмите **Network Management > Load Balancer**.
3. Нажмите : > **Delete** и подтвердите.

CLI

```
kubectrl delete alb2 test-alb -n cpaas-system
```

Настройка портов слушателя (Frontend)

Балансировщик нагрузки поддерживает приём клиентских соединений через порты слушателя и соответствующие протоколы, включая HTTPS, HTTP, gRPC, TCP и UDP.

Пререквизиты

Если необходимо добавить HTTPS-порт слушателя, следует также обратиться к администратору для назначения TLS-сертификата текущему проекту для шифрования.

Пример ресурса Frontend (CR)

```
# alb-frontend-demo.yaml
apiVersion: crd.alauda.io/v1
kind: Frontend
metadata:
  labels:
    alb2.cpaas.io/name: alb-demo (1)
  name: alb-demo-00080 (2)
  namespace: cpaas-system
spec:
  backendProtocol: "http"
  certificate_name: "" (3)
  port: 80
  protocol: http (4)
  serviceGroup: (5)
  services:
    - name: hello-world
      namespace: default
      port: 80
      weight: 100 (6)
```

1. Обязательно, указывает ALB-инстанс, к которому принадлежит этот **Frontend**.
2. Формат `$alb_name-$port`.
3. Формат `$secret_ns/$secret_name`.
4. Протокол самого **Frontend**.
 - **http|https|grpc|grpcs** для прокси уровня 7.
 - **tcp|udp** для прокси уровня 4.
5. Для прокси уровня 4 **serviceGroup** обязателен. Для уровня 7 **serviceGroup** опционален. При поступлении запроса ALB сначала пытается сопоставить его с правилами, связанными с этим **Frontend**. Если запрос не соответствует ни одному правилу, ALB перенаправляет его в дефолтный **serviceGroup**, указанный в конфигурации **Frontend**.

6. Конфигурация **weight** применяется для алгоритмов планирования Round Robin и Weighted Round Robin.

Внимание: ALB слушает ingress и автоматически создаёт **Frontend** или Rule. Поле **source** определяется следующим образом:

1. **spec.source.type** в настоящее время поддерживает только ingress.
2. **spec.source.name** — имя ingress.
3. **spec.source.namespace** — пространство имён ingress.

Создание портов слушателей (Frontend)

Веб-консоль

1. Перейдите в Container Platform.
2. В левой навигационной панели нажмите **Network > Load Balancing**.
3. Нажмите на имя балансировщика нагрузки для перехода на страницу деталей.
4. Нажмите Add Listener Port.
5. Ознакомьтесь с инструкциями ниже для настройки параметров.

Параметр	Описание
Protocol	Поддерживаемые протоколы: HTTPS, HTTP, gRPC, TCP и UDP. При выборе HTTPS необходимо добавить сертификат; для gRPC добавление сертификата опционально. Примечание: • При выборе протокола gRPC протокол бэкенда по умолчанию — gRPC, при этом не поддерживается сессия. • Если для gRPC задан сертификат, балансировщик снимет сертификат gRPC и перенаправит незашифрованный трафик gRPC на бэкенд. • При использовании кластера Google GKE балансировщик одного типа контейнерной сети не может одновременно иметь протоколы слушателя TCP и UDP.
Internal Routing Group	- При выборе алгоритма балансировки Round Robin (RR) трафик распределяется по портам внутренней маршрутизации в порядке группы маршрутов. - При выборе алгоритма Weighted Round Robin (WRR) внутренние маршруты с большим значением веса имеют большую вероятность выбора; трафик распределяется по портам внутренней маршрутизации согласно рассчитанной вероятности на основе настроенного weight. Совет: Вероятность рассчитывается как отношение текущего значения веса к сумме всех весов.

Параметр	Описание
Session Persistence	Всегда перенаправляет определённые запросы на бэкенд-сервисы, соответствующие вышеуказанной внутренней группе маршрутов. Определённые запросы включают (выберите один): - Хэш исходного адреса: все запросы с одного IP-адреса. Примечание: В публичных облаках исходный адрес часто меняется, что может привести к тому, что запросы от одного клиента будут иметь разные IP-адреса в разное время, и метод хэширования по адресу не даст ожидаемого результата. - Ключ cookie: запросы, содержащие указанный cookie. - Имя заголовка: запросы, содержащие указанный заголовок.
Backend Protocol	Протокол, используемый для передачи трафика на бэкенд-сервисы. Например, при передаче на Kubernetes или dex-сервисы бэкенда следует выбрать протокол HTTPS.

6. Нажмите **ОК**.

CLI

```
kubectcl apply -f alb-frontend-demo.yaml -n cpaas-system
```

Последующие действия

Для трафика с портов HTTP, gRPC и HTTPS, помимо стандартной внутренней группы маршрутов, можно настроить более разнообразные правила сопоставления бэкенд-сервисов rules. Балансировщик сначала пытается сопоставить соответствующий бэкенд согласно установленным правилам; если совпадений нет, то направляет трафик на бэкенд из указанной внутренней группы маршрутов.

Настройка правил

Добавление правил переадресации для портов слушателя протоколов HTTPS, HTTP и gRPC. Балансировщик будет сопоставлять бэкенд-сервисы на основе этих правил.

Внимание: Правила переадресации нельзя добавлять для протоколов TCP и UDP.

Создание правила

Веб-консоль

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Network > Load Balancing**.
3. Нажмите на имя балансировщика нагрузки.
4. Нажмите на имя порта слушателя.
5. Нажмите **Add Rule**.
6. Ознакомьтесь с описаниями ниже для настройки параметров.

Параметр	Описание
Internal Route Group	- При выборе алгоритма балансировки Round Robin (RR) трафик распределяется по портам внутренней маршрутизации в порядке группы маршрутов. - При выборе алгоритма Weighted Round Robin (WRR) внутренние маршруты с большим значением веса имеют большую вероятность выбора; трафик распределяется по портам внутренней маршрутизации согласно рассчитанной вероятности на основе настроенного weight. Внимание: Вероятность рассчитывается как отношение текущего значения веса к сумме всех весов.
Rule	Критерии сопоставления балансировщика с бэкенд-сервисами, включая индикаторы правил и их значения. Отношение между разными индикаторами — «и». • Доменное имя: поддерживается добавление wildcard-доменов и точных доменных имён. При равенстве приоритетов для одного правила, если существуют и wildcard, и точные доменные правила, приоритет имеет точное доменное правило. • URL: RegEx — регулярные выражения URL, начинающиеся с /; StartsWith — префиксы URL, начинающиеся с /. • IP: Equal — конкретный IP-адрес; Range — диапазон IP-адресов. • Заголовок: кроме ключа заголовка, необходимо задать правило сопоставления. Equal — конкретное значение заголовка; Range — диапазон значений заголовка; RegEx — регулярное выражение заголовка. • Cookie: кроме ключа cookie, необходимо задать правило сопоставления. Equal — конкретное значение cookie. • URL Param: в правилах сопоставления Equal — конкретный параметр URL; Range — диапазон параметров URL. • Service Name: имя сервиса, использующего протокол gRPC. При использовании gRPC можно настроить этот параметр, чтобы трафик перенаправлялся на соответствующий сервис по имени, например: /helloworld.Greeter.
Session Persistence	Всегда перенаправляет определённые запросы на бэкенд-сервисы, соответствующие внутренней группе маршрутов. Определённые запросы включают (выберите один): • Хэш исходного адреса: все запросы с одного IP-адреса. • Ключ cookie: запросы, содержащие указанный cookie.

Параметр	Описание
	Имя заголовка: запросы, содержащие указанный заголовок.
URL Rewrite	Перезаписывает обращаемый адрес на адрес бэкенд-сервиса платформы. Для работы требуется настройка правила URL с индикатором StartsWith, а адрес перезаписи (rewrite-target) должен начинаться с /. Например: при настройке домена <i>bar.example.com</i> и начального пути URL /, включении функции URL Rewrite и установке адреса перезаписи в <i>/test</i> обращение к <i>bar.example.com</i> будет переписано в <i>bar.example.com/test</i>.
Backend Protocol	Протокол, используемый для передачи трафика на бэкенд-сервис. Например, при передаче на Kubernetes или dex-сервисы бэкенда следует выбрать протокол HTTPS.
Redirection	Перенаправляет трафик на новый адрес, а не на бэкенд-сервисы внутренней группы маршрутов. Например: при обновлении страницы по исходному адресу, чтобы избежать ошибок 404 или 503, трафик можно перенаправить на новый адрес. - HTTP Status Code: код состояния, который браузер покажет пользователю перед перенаправлением. - Redirect Address: при вводе относительного адреса (например, <i>/index.html</i>) трафик будет направлен на <i>адрес балансировщика/index.html</i>; при вводе абсолютного адреса (например, <i>https://www.example.com</i>) трафик будет направлен на указанный адрес.
Rule Priority	Приоритет сопоставления правил: 10 уровней от 1 до 10, где 1 — самый высокий приоритет, по умолчанию 5. Если одновременно удовлетворяются несколько правил, выбирается правило с более высоким приоритетом; при равенстве приоритетов применяется правило по умолчанию.
Cross-Origin Resource Sharing (CORS)	CORS (Cross-origin resource sharing) — механизм, использующий дополнительные HTTP-заголовки, чтобы указать браузеру, что веб-приложение с одного источника (домена) может получить доступ к ресурсам с другого источника. При запросе ресурса с сервера другого домена, протокола или порта инициируется кросс-доменный HTTP-запрос.
Allowed Origins	Указывает разрешённые источники доступа.

Параметр	Описание
	• *: разрешает запросы с любого источника. • Доменное имя: разрешает запросы с указанного домена.
Allowed Headers	Указывает HTTP-заголовки, разрешённые в CORS, чтобы избежать лишних предварительных запросов и повысить эффективность. Примеры: Примечание: Другие часто используемые или кастомные заголовки не перечислены; заполняйте по необходимости. • Origin: указывает источник запроса, то есть домен, отправляющий запрос. • Authorization: указывает данные авторизации, например Basic Authentication или Token. • Content-Type: указывает тип содержимого запроса/ответа, например application/json, application/x-www-form-urlencoded и др. • Accept: указывает типы содержимого, которые клиент может принимать, обычно для получения определённого типа ответа.

7. Нажмите **Add**.

CLI

```
kubectl apply -f alb-rule-demo.yaml -n cpaas-system
```

Логи и мониторинг

Сочетая визуализированные логи и данные мониторинга, можно быстро выявлять и устранять проблемы или сбои балансировщика нагрузки.

Просмотр логов

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Network Management > Load Balancer**.
3. Нажмите на Load Balancer Name.
4. Во вкладке **Logs** просмотрите логи работы балансировщика с точки зрения контейнера.

Метрики мониторинга

Внимание: В кластере, где расположен балансировщик, должны быть развернуты сервисы мониторинга.

1. Перейдите в Platform Management.
2. В левой навигационной панели нажмите **Network Management > Load Balancer**.
3. Нажмите на Load Balancer Name.

4. Во вкладке **Monitoring** просмотрите тренды метрик балансировщика с точки зрения узла.

- **Usage Rate:** текущая загрузка CPU и памяти балансировщика на данном узле.
- **Throughput:** общий входящий и исходящий трафик экземпляра балансировщика.

Выделение ресурсов CPU и памяти

Для предложенных платформой спецификаций для **малых, средних, больших** и **пользовательских** производственных сред, а также методов выделения ресурсов для **инстансов** и **портов**, ниже приведены рекомендации для развертывания.

Маленькая производственная среда

Для меньших масштабов бизнеса, например, при наличии не более 5 узлов в кластере и использовании только для запуска стандартных приложений, достаточно **одного** балансировщика нагрузки. Рекомендуется использовать его в режиме **высокой доступности** с минимум 2 репликами для обеспечения стабильности среды.

Вы можете изолировать балансировщик нагрузки с помощью **портовой** изоляции, что позволит нескольким проектам совместно его использовать.

Пиковый QPS, измеренный в лабораторных условиях для этой спецификации, составляет примерно 300 запросов в секунду.

Средняя производственная среда

Когда объем бизнеса достигает определенного масштаба, например, при наличии не более 30 узлов в кластере и необходимости обработки высококонкурентного бизнеса наряду с запуском стандартных приложений, **одного** балансировщика нагрузки по-прежнему будет достаточно. Рекомендуется использовать режим **высокой доступности** с минимум 3 репликами для поддержания стабильности среды.

Вы можете использовать либо метод **изоляции по портам**, либо выделение по **инстансам** для совместного использования балансировщика нагрузки несколькими проектами. Конечно, вы также можете создавать новые балансировщики нагрузки для выделенного использования ключевыми проектами.

Пиковый QPS, измеренный в лабораторных условиях для этой спецификации, составляет около 10 000 запросов в секунду.

Большая производственная среда

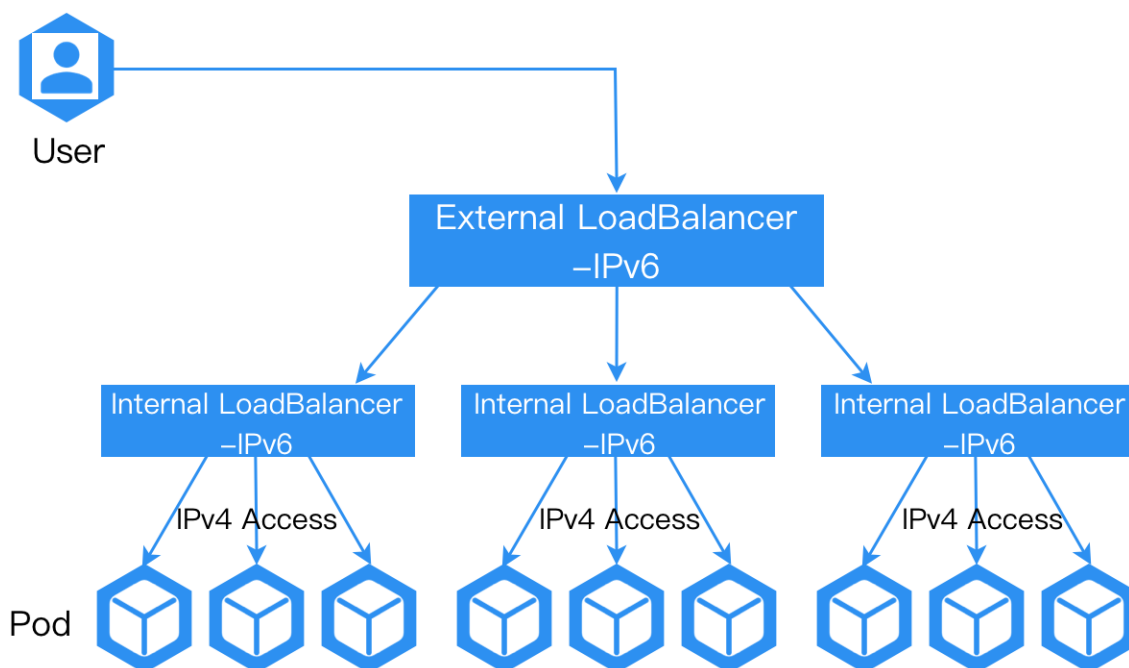
Для больших объемов бизнеса, например, при наличии более 30 узлов в кластере и необходимости обработки высококонкурентного бизнеса, а также долгоживущих соединений с данными, рекомендуется использовать **несколько** балансировщиков нагрузки, каждый из которых работает в типе **высокой доступности** с минимум 3 репликами для обеспечения стабильности среды.

Вы можете изолировать балансировщик нагрузки с помощью либо метода **изоляции по портам**, либо выделения по **инстансам** для совместного использования несколькими проектами. Также можно создавать новые балансировщики нагрузки для эксклюзивного использования ключевыми проектами.

Пиковый QPS, измеренный в лабораторных условиях для этой спецификации, составляет примерно 20 000 запросов в секунду.

Проброс IPv6-трафика на IPv4-адреса внутри кластера

Настроив внешний балансировщик нагрузки для кластера, мы можем перенаправлять IPv6-трафик на внутренние IPv4-адреса внутри кластера. Это позволяет внедрить возможности IPv6 поверх существующей IPv4-сети, обеспечивая большую гибкость и масштабируемость архитектуры системы, а также лучшее удовлетворение разнообразных сетевых требований.



Метод настройки

1. Настройте IPv6-адрес для узла, на котором расположен балансировщик нагрузки (load balancer).
2. Убедитесь, что у внешнего балансировщика нагрузки (load balancer) есть IPv6-адрес, и что трафик, обращающийся к IPv6-адресу балансировщика, может быть перенаправлен на IPv6-адрес узла, где расположен load balancer.

После выполнения вышеуказанной настройки IPv4-сервисы, размещённые на load balancer, смогут предоставлять внешние возможности доступа по IPv6 через балансировщик.

Проверка результата

После настройки доступ к IPv6-адресу внешнего балансировщика нагрузки должен обеспечивать нормальный доступ к приложению.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Kube-OVN Overlay Network настройка IPsec

В этом разделе представлен подробный гид по включению и отключению зашифрованного туннельного трафика IPsec в оверлейной сети Kube-OVN. Поскольку туннельный трафик OVN передаётся через физические маршрутизаторы и коммутаторы, которые могут находиться в ненадёжных публичных сетях или подвергаться атакам, включение шифрования IPsec эффективно предотвращает мониторинг и подмену данных трафика.

Терминология

Термин	Объяснение
IPsec	Протокол и технология, используемые для защиты и проверки данных, передаваемых через интернет. Обеспечивает безопасную связь на уровне IP и в основном используется для создания виртуальных частных сетей (VPN) и защиты передачи IP-пакетов. IPsec обеспечивает безопасность данных преимущественно следующими способами: • Шифрование данных: с помощью технологий шифрования IPsec гарантирует, что данные не будут украдены или изменены во время передачи. Распространённые алгоритмы шифрования включают AES, 3DES и др. • Проверка целостности данных: IPsec использует хэш-функции (например, SHA-1, SHA-256) для проверки целостности данных, гарантируя, что данные не были изменены в процессе передачи. • Аутентификация: IPsec может проверять идентичность обеих сторон связи с помощью различных методов (например, предварительно разделённые ключи, цифровые сертификаты), чтобы предотвратить несанкционированный доступ. • Управление ключами: IPsec использует протокол Internet Key Exchange (IKE) для согласования и управления ключами шифрования.

Внимание:

- Включение IPsec может вызвать кратковременное прерывание сети на несколько секунд.

- Если версия ядра — 3.10.0-1160.el7.x86_64, при включении функции IPsec в Kube-OVN могут возникнуть проблемы совместимости.

Прerequisites

Выполните следующую команду, чтобы проверить, поддерживает ли текущее ядро операционной системы модули, связанные с IPsec. Если вывод показывает, что все модули, связанные с XFRM, имеют значение **y** или **m**, это означает поддержку IPsec.

```
cat /boot/config-$(uname -r) | grep CONFIG_XFRM
```

Вывод:

```
CONFIG_XFRM_ALGO=y
CONFIG_XFRM_USER=y
CONFIG_XFRM_SUB_POLICY=y
CONFIG_XFRM_MIGRATE=y
CONFIG_XFRM_STATISTICS=y
CONFIG_XFRM_IPCOMP=m
```

Включение IPsec

Примечание: если не указано иное, все команды необходимо выполнять в CLI-инструменте на Master-узле кластера.

1. Измените конфигурационный файл **kube-ovn-controller**.
 1. Выполните следующую команду для редактирования YAML-конфигурации **kube-ovn-controller**.

```
kubectrl edit deploy kube-ovn-controller -n kube-system
```

2. Измените указанные поля согласно следующим инструкциям.

spec:

template:

spec:

containers:

- args:

- --enable-ovn-ipsec=true # Добавьте это поле

securityContext:

runAsUser: 0 # Измените значение на 0

Объяснение полей:

- spec.template.spec.containers[0].args: Добавьте - --enable-ovn-ipsec=true под этим полем.
- spec.template.spec.containers[0].securityContext.runAsUser: Измените значение этого поля на 0.

3. Сохраните изменения.

2. Измените конфигурационный файл **kube-ovn-cni**.

1. Выполните следующую команду для редактирования YAML-конфигурации kube-ovn-cni.

```
kubectrl edit ds kube-ovn-cni -n kube-system
```

2. Измените указанные поля согласно следующим инструкциям.

spec:

template:

spec:

containers:

- args:

- --enable-ovn-ipsec=true # Добавьте это поле

volumeMounts: # Добавьте путь монтирования, смонтируйте том с именем ovs-ipsec-keys в контейнер

- mountPath: /etc/ovs_ipsec_keys

name: ovs-ipsec-keys

volumes: # Добавьте том с именем ovs-ipsec-keys типа hostPath

- name: ovs-ipsec-keys

hostPath:

path: /etc/origin/ovs_ipsec_keys

Описание полей:

- spec.template.spec.containers[0].args: Добавьте --enable-ovn-ipsec=true под этим полем.
- spec.template.spec.containers[0].volumeMounts: Добавьте путь монтирования и смонтируйте том с именем ovs-ipsec-keys в контейнер.
- spec.template.spec.volumes: Добавьте том с именем ovs-ipsec-keys типа hostPath под этим полем.

3. Сохраните изменения.

3. Проверьте, успешно ли функция была включена.

1. Выполните следующую команду для входа в Pod kube-ovn-cni.

```
kubectrl exec -it -n kube-system $(kubectrl get pods -n kube-system -l app=kube-ovn-cni -o=jsonpath='{.items[0].metadata.name}') -- /bin/bash
```

2. Выполните следующую команду для проверки количества соединений Security Associations. Если их (число узлов - 1) в статусе up, это означает успешное включение.

```
ipsec status | grep "Security"
```

Вывод:

Security Associations (2 up, 0 connecting): # Поскольку в кластере 3 узла, видно, что количество соединений — 2 в статусе up

Отключение IPsec

Примечание: если не указано иное, все команды необходимо выполнять в CLI-инструменте на Master-узле кластера.

1. Измените конфигурационный файл kube-ovn-controller:
 - 1.1. Выполните следующую команду для редактирования YAML-конфигурации kube-ovn-controller:
kubectcl edit deploy kube-ovn-controller -n kube-system
 - 1.2. Измените указанные поля согласно следующим инструкциям:

```
spec:
  template:
    spec:
      containers:
      - args:
        - --enable-ovn-ipsec=false # Измените на false
      securityContext:
        runAsUser: 65534 # Измените значение на 65534
```

Описание полей:

- spec.template.spec.containers[0].args: Измените значение поля enable-ovn-ipsec на false.
- spec.template.spec.containers[0].securityContext.runAsUser: Измените значение этого поля на 65534.

- 1.3. Сохраните изменения.

2. Измените конфигурационный файл kube-ovn-cni:
 - 2.1. Выполните следующую команду для редактирования YAML-конфигурации kube-ovn-cni:
kubectcl edit ds kube-ovn-cni -n kube-system
 - 2.2. Измените указанные поля согласно следующим инструкциям:

- Конфигурация до изменения:

```
spec:
  template:
    spec:
      containers:
      - args:
        - --enable-ovn-ipsec=true # Измените на false
      volumeMounts: # Удалите путь монтирования с именем ovs-ipsec-keys
      - mountPath: /etc/ovs_ipsec_keys
        name: ovs-ipsec-keys
      volumes: # Удалите том с именем ovs-ipsec-keys типа hostPath
      - name: ovs-ipsec-keys
        hostPath:
          path: /etc/origin/ovs_ipsec_keys
```

Описание полей:

- `spec.template.spec.containers[0].args`: Измените значение поля `enable-ovn-ipsec` на `false`.
- `spec.template.spec.containers[0].volumeMounts`: Удалите путь монтирования с именем `ovs-ipsec-keys`.
- `spec.template.spec.volumes`: Удалите том с именем `ovs-ipsec-keys` типа `hostPath`.

- Конфигурация после изменения:

`spec:`

`template:`

`spec:`

`containers:`

`- args:`

`- --enable-ovn-ipsec=false`

`volumeMounts:`

`volumes:`

2.3. Сохраните изменения.

3. Проверьте, успешно ли функция была отключена:

3.1. Выполните следующую команду для входа в Pod `kube-ovn-cni`:

```
kubectl exec -it -n kube-system $(kubectl get pods -n kube-system -l app=kube-ovn-cni -o=jsonpath='{.items[0].metadata.name}') -- /bin/bash
```

3.2. Выполните следующую команду для проверки статуса соединения:

```
ipsec status
```

Если вывода нет, это означает успешное отключение.

Хранение

Введение

Kubernetes предлагает гибкий и масштабируемый механизм хранения для управления сохранением данных в контейнеризованных средах. Абстрагируя ресурсы хранения, такие как Volumes, PersistentVolumes и PersistentVolumeClaims, Kubernetes отделяет приложения от базовых систем хранения, обеспечивая динамическое выделение, автоматическое монтирование и сохранение данных между узлами.

Ключевые возможности включают поддержку множества систем хранения (например, локальные диски, NFS, облачные сервисы хранения), динамическое выделение, контроль режимов доступа (например, права на чтение/запись) и управление жизненным циклом — что удовлетворяет потребности в хранении для stateful-приложений. Для корпоративных нагрузок, требующих высокой доступности, сохранения данных и изоляции многопользовательской среды, хранение в Kubernetes является важной базовой функцией.

Хранение в Kubernetes разработано для разработчиков, инженеров эксплуатации и платформенных команд, помогая им эффективно и безопасно управлять данными в контейнеризованных нагрузках.

Основные концепции

Хранение данных в Kubernetes строится вокруг трёх ключевых понятий: **PersistentVolume (PV)**, **PersistentVolumeClaim (PVC)** и **StorageClass**. Они определяют, как запрашивается, выделяется и настраивается хранилище внутри кластера. В основе часто лежат драйверы **CSI** (Container Storage Interface), которые отвечают за фактическое предоставление и подключение хранилища. Давайте кратко рассмотрим каждый компонент и выделим роль CSI-драйвера.

Persistent Volume (PV)

PersistentVolume (PV) — это часть хранилища в кластере, которая была выделена (либо статически администратором, либо динамически через StorageClass). Он представляет собой базовое хранилище — например, диск у облачного провайдера или файловую систему с сетевым доступом — и рассматривается как ресурс в кластере, аналогично узлу.

Persistent Volume Claim (PVC)

PersistentVolumeClaim (PVC) — это запрос на хранилище. Пользователи определяют, сколько хранилища им нужно и режим доступа (например, чтение-запись). Если подходящий PV доступен или может быть динамически создан (через StorageClass), PVC связывается с этим PV. После связывания Pod'ы могут ссылаться на PVC для сохранения или совместного использования данных.

Generic Ephemeral Volumes

Generic Ephemeral Volumes для Kubernetes — это функция, введённая в Kubernetes, которая позволяет использовать CSI-управляемые **временные** тома в течение жизненного цикла

Pod, аналогично **emptyDir**, но более мощная и позволяющая монтировать любой тип CSI тома (с поддержкой снимков, масштабирования и т.д.).

emptyDir

1. **emptyDir** — это временный том типа пустой директории.
2. Он создаётся при запуске Pod на узле, и хранилище располагается на локальной файловой системе этого узла (по умолчанию диск узла).
3. При удалении Pod данные в **emptyDir** также удаляются.

hostPath

В Kubernetes том **hostPath** — это специальный тип тома, который отображает файл или директорию с файловой системы хост-узла непосредственно в контейнер Pod'a.

Позволяет Pod'у получить доступ к файлам или директориям на хост-узле.

Полезен для:

- Доступа к ресурсам уровня хоста (например, сокет Docker)
- Отладки
- Использования предварительно существующих данных на узле

ConfigMap

ConfigMap в Kubernetes — это объект API, используемый для хранения неконфиденциальных конфигурационных данных в виде пар ключ-значение. Он позволяет отделить конфигурацию от кода приложения, делая приложения более переносимыми и удобными в управлении.

Secret

В Kubernetes **Secret** — это объект API, который хранит конфиденциальные данные, такие как:

- пароли
- OAuth токены
- SSH ключи
- TLS сертификаты
- учётные данные баз данных

Secrets помогают защитить эти данные, избегая их прямого хранения в спецификациях Pod или образах контейнеров.

StorageClass

StorageClass описывает как тома должны динамически выделяться. Он сопоставляется с конкретным провайдером (часто CSI драйвером) и может включать параметры, такие как уровни хранения, характеристики производительности или другие настройки бэкенда. Создавая несколько **StorageClass**, можно предложить разработчикам различные типы хранилищ.

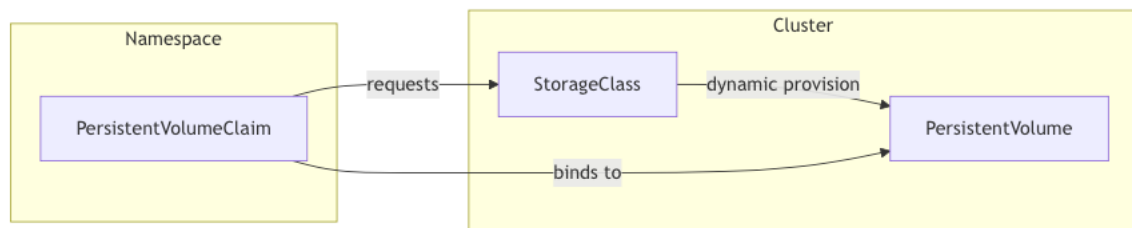


Диаграмма: Взаимосвязь между PVC, PV и StorageClass.

Container Storage Interface (CSI)

Container Storage Interface (CSI) — это стандартный API, который Kubernetes использует для интеграции с драйверами хранилища. Он позволяет сторонним поставщикам создавать плагины вне ядра Kubernetes, то есть вы можете устанавливать или обновлять драйвер хранилища без изменения самого Kubernetes.

CSI драйвер обычно состоит из двух компонентов:

Компонент контроллера: работает в кластере (часто как Deployment) и обрабатывает операции высокого уровня, такие как **создание** или **удаление** томов. Для сетевого хранилища он также может управлять подключением и отключением томов к узлам

Компонент узла: работает на каждом узле (часто как DaemonSet) и отвечает за **монтирование** и **отмонтирование** тома на конкретном узле. Он взаимодействует с kubelet, чтобы обеспечить доступность тома для Pod'ов.

Когда пользователь создаёт PVC, ссылающийся на StorageClass с CSI-драйвером, драйвер отслеживает этот запрос и выделяет хранилище при необходимости динамического выделения. После создания хранилища драйвер уведомляет Kubernetes, который создаёт соответствующий PV и связывает его с PVC. Когда Pod использует этот PVC, компонент

Используя **PV**, **PVC**, **StorageClass** и **CSI**, Kubernetes предоставляет мощный декларативный подход к управлению хранилищем. Администраторы могут определить один или несколько StorageClass, представляющих разные бэкенды или уровни производительности, а разработчики просто запрашивают хранилище через PVC — не беспокоясь о внутренней инфраструктуре.

Persistent Volume

PersistentVolume (PV) представляет собой отображение с томами бэкенд-хранилища в кластере Kubernetes и функционирует как ресурс API Kubernetes. Это ресурс кластера, создаваемый и настраиваемый администраторами единообразно, отвечающий за абстрагирование реальных ресурсов хранения и формирование инфраструктуры хранения кластера.

PersistentVolumes обладают жизненным циклом, независимым от Pod'ов, что позволяет сохранять данные Pod'ов постоянно.

Администраторы могут вручную создавать статические **PersistentVolumes** или генерировать динамические **PersistentVolumes** на основе классов хранения. Если разработчикам необходимо получить ресурсы хранения для приложений, они могут запросить их через

PersistentVolumeClaims (PVC), которые сопоставляются и связываются с подходящими **PersistentVolumes**.

Динамические Persistent Volumes и статические Persistent Volumes

Платформа поддерживает управление двумя типами PersistentVolumes, создаваемыми администраторами: динамическими и статическими Persistent Volumes.

- **Динамические Persistent Volumes:** Реализуются на основе классов хранения. Классы хранения создаются администраторами и представляют собой ресурс Kubernetes, описывающий категорию ресурсов хранения. Как только разработчик создаёт PersistentVolumeClaim, связанный с классом хранения, платформа динамически создаёт подходящий PersistentVolume в соответствии с параметрами, настроенными в PersistentVolumeClaim и классе хранения, и связывает его с PersistentVolumeClaim для динамического выделения ресурсов хранения.
- **Статические Persistent Volumes:** Persistent Volumes, созданные вручную администратором. В настоящее время поддерживается создание статических Persistent Volumes типа HostPath или NFS shared storage. Когда разработчики создают PersistentVolumeClaim без использования класса хранения, платформа сопоставляет и связывает подходящий статический PersistentVolume в соответствии с параметрами, настроенными в PersistentVolumeClaim.
- **HostPath:** Использует файловый каталог на хосте узла (локальное хранилище не поддерживается) в качестве бэкенд-хранилища, например: /etc/kubernetes. Обычно применяется только в тестовых сценариях внутри кластера с одним вычислительным узлом.
- **NFS Shared Storage:** Относится к Network File System — распространённому типу бэкенд-хранилища для Persistent Volumes. Пользователи и программы могут обращаться к файлам на удалённых системах так, как если бы они были локальными.

Жизненный цикл Persistent Volumes

1. **Provisioning (создание):** Администраторы вручную создают статические Persistent Volumes. После создания Persistent Volume переходит в состояние Available; либо платформа динамически создаёт подходящие Persistent Volumes на основе PersistentVolumeClaims, связанных с классами хранения.
2. **Binding (связывание):** Как только статический Persistent Volume сопоставлен и связан с PersistentVolumeClaim, он переходит в состояние Bound; динамические Persistent Volumes создаются динамически на основе запросов, соответствующих PersistentVolumeClaims, и также переходят в состояние Bound после успешного создания.
3. **Using (использование):** Разработчики связывают PersistentVolumeClaims с экземплярами контейнеров вычислительных компонентов, используя ресурсы бэкенд-хранилища, отображённые Persistent Volumes.
4. **Releasing (освобождение):** После удаления PersistentVolumeClaim разработчиками Persistent Volume освобождается.

5. Reclaiming (восстановление): После освобождения Persistent Volume на нём выполняются операции восстановления в соответствии с параметрами политики восстановления Persistent Volume или класса хранения.

Режимы доступа и режимы томов

В Kubernetes PersistentVolumeClaims (PVC) и StorageClasses работают вместе для управления тем, как хранилище предоставляется и используется рабочими нагрузками. Два ключевых понятия в этой области — это **режимы доступа** и **режимы томов**.

Режимы доступа в Kubernetes

Режимы доступа определяют, как том может быть смонтирован и использован подами. Основные режимы доступа:

- **ReadWriteOnce (RWO)**: том может быть смонтирован в режиме чтения-записи одним узлом.
- **ReadOnlyMany (ROX)**: том может быть смонтирован в режиме только для чтения несколькими узлами.
- **ReadWriteMany (RWX)**: том может быть смонтирован в режиме чтения-записи несколькими узлами.

Режимы доступа по StorageClass

Storage Class	Поддержка RWO	Поддержка ROX	Поддержка RWX
CephFS File Storage	Да	Нет	Да
CephRBD Block Storage	Да	Нет	Нет
TopoLVM	Да	Нет	Нет
NFS Shared Storage	Да	Нет	Да

Как показано выше, файловые системы хранения, такие как **CephFS** и **NFS**, поддерживают множественные одновременные операции записи или чтения, что делает их подходящими для сценариев совместного доступа. С другой стороны, блочные системы хранения, такие как **CephRBD** и **TopoLVM**, обеспечивают эксклюзивный доступ только одному узлу за раз.

Режимы томов в Kubernetes

Режимы томов определяют, как данные предоставляются поду:

Filesystem: том монтируется в под как файловая система.

Block: том представлен как необработанное блочное устройство.

Режимы томов по StorageClass

Storage Class	Тип	Поддерживаемые режимы томов
CephFS File Storage	File Storage	Filesystem
CephRBD Block Storage	Block Storage	Filesystem, Block
TopoLVM	Block Storage	Filesystem, Block
NFS Shared Storage	File Storage	Filesystem

Блочные системы хранения, такие как **CephRBD** и **TopoLVM**, предлагают как доступ через файловую систему, так и через необработанный блочный доступ, обеспечивая гибкость для

различных потребностей приложений. Файловые системы хранения, такие как **CephFS** и **NFS**, напротив, поддерживают только режим файловой системы.

Особенности хранения: снимки и расширение

Kubernetes также поддерживает расширенные функции, такие как снимки томов и динамическое расширение PVC, в зависимости от используемого StorageClass.

Storage Class	Снимок тома	Расширение
CephFS File Storage	Поддерживается	Поддерживается
CephRBD Block Storage	Поддерживается	Поддерживается
TopoLVM	Поддерживается	Поддерживается
NFS Shared Storage	Не поддерживается	Не поддерживается

Снимки томов поддерживаются только для PVC, динамически выделенных с использованием StorageClass. Эта функция полезна для резервного копирования и клонирования окружений.

Заключение

При настройке хранилища в Kubernetes понимание **режимов доступа** и **режимов томов** для PVC и соответствующих **StorageClasses** критично для выбора правильного решения для вашей рабочей нагрузки. Файловые решения хранения, такие как CephFS и NFS, идеально подходят для сценариев совместного доступа, тогда как блочные системы хранения, такие как CephRBD и TopoLVM, превосходят в высокопроизводительных развертываниях на одном узле. Кроме того, поддержка таких функций, как снимки и расширение, значительно повышает гибкость хранения и стратегии управления данными.

Создание Storage Class типа CephFS File Storage

CephFS file storage — это встроенная файловая система Ceph, которая предоставляет платформе метод доступа к хранилищу на основе Container Storage Interface (CSI), обеспечивая безопасный, надежный и масштабируемый общий файловый сервис, подходящий для сценариев, таких как совместное использование файлов и резервное копирование данных. Перед началом необходимо сначала создать Storage Class для CephFS file storage.

После привязки Storage Class в Persistent Volume Claim (PVC) платформа динамически создаст persistent volumes на узлах в соответствии с запросом persistent volume для бизнес-приложений.

Развёртывание Volume Plugin

После нажатия Deploy на странице Distributed Storage Создайте Storage Service или Подключитесь к Storage Service.

Создание Storage Class

1. Перейдите в Platform Management.
2. В левой навигационной панели нажмите **Storage Management > Storage Classes**.

3. Нажмите Create Storage Class.

Внимание: Вы также можете создать Storage Class с помощью YAML.

4. Выберите CephFS File Storage и нажмите Next.

5. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Reclaim Policy	Политика возврата для persistent volumes. - Delete: При удалении persistent volume claim связанный persistent volume также будет удалён. - Retain: Связанный persistent volume останется, даже если persistent volume claim удалён.
Access Modes	Все режимы доступа, поддерживаемые текущим хранилищем. При объявлении persistent volumes позже можно выбрать только один из этих режимов. - ReadWriteOnce (RWO): Может быть смонтирован как для чтения, так и для записи одним узлом. - ReadWriteMany (RWX): Может быть смонтирован для чтения и записи несколькими узлами.
Allocate Project	Укажите проекты, которые могут использовать этот тип хранилища. Если в данный момент нет проектов, которым нужно использовать этот тип хранилища, можно не выделять их сейчас и обновить позже.

Внимание: Следующие параметры необходимо задать в distributed storage, они будут применены здесь напрямую.

Storage Cluster: Встроенный Ceph storage cluster в текущем кластере.

Storage Pool: Логический раздел, используемый для хранения данных в storage cluster.

6. Нажмите **Create**.

Создание класса блочного хранилища CephRBD

Блочное хранилище CephRBD — это встроенное блочное хранилище Ceph для платформы, предоставляющее метод доступа к хранилищу на основе Container Storage Interface (CSI), способный обеспечивать высокие IOPS и низкую задержку, что подходит для сценариев, таких как базы данных и виртуализация. Перед использованием необходимо создать класс блочного хранилища CephRBD.

После того как Persistent Volume Claim (PVC) будет привязан к классу хранилища, платформа динамически создаст Persistent Volume на основе Persistent Volume Claim для использования бизнес-приложениями.

Развёртывание плагина тома

После нажатия **Deploy** на странице **Distributed Storage** создайте сервис хранения или подключитесь к сервису хранения.

Создание класса хранилища

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management > Storage Classes**.
3. Нажмите **Create Storage Class**.

Внимание: Вы также можете выбрать YAML для выполнения операции.

4. Выберите CephRBD Block Storage и нажмите **Next**.
5. Настройте соответствующие параметры по необходимости.

Параметр	Описание
File System	По умолчанию EXT4 — журналируемая файловая система для Linux, способная обеспечивать хранение экстендов и обработку больших файлов. Вместимость файловой системы может достигать 1 EiB, поддерживаемый размер файла — до 16 TiB.
Reclaim Policy	Политика возврата для постоянных томов. - Delete: связанный постоянный том будет удалён вместе с Persistent Volume Claim. - Retain: связанный постоянный том сохранится даже при удалении Persistent Volume Claim.
Access Modes	Поддерживается только ReadWriteOnce (RWO): том может быть смонтирован в режиме чтения-записи только одним узлом.

Внимание: Следующие параметры необходимо задать в распределённом хранилище, они будут применены здесь напрямую.

Storage Cluster: Встроенный Ceph storage cluster в текущем кластере.

Storage Pool: Логический раздел, используемый для хранения данных в storage cluster.

6. Нажмите **Create**.

Создание локального Storage Class TopoLVM

TopoLVM — это локальное хранилище на основе LVM, которое обеспечивает простые, удобные в обслуживании и высокопроизводительные локальные сервисы хранения, подходящие для сценариев, таких как базы данных и middleware. Перед использованием необходимо создать Storage Class TopoLVM.

После того как Persistent Volume Claim (PVC) будет привязан к этому Storage Class, платформа динамически создаст persistent volumes на узлах на основе PVC для использования бизнес-приложениями.

Обзор

Преимущества использования

- По сравнению с удалённым хранилищем (например, **NFS shared storage**): хранилище типа TopoLVM расположено локально на узле, что обеспечивает лучшую производительность по IOPS и пропускной способности, а также меньшую задержку.

- По сравнению с hostPath (например, **local-path**): хотя оба варианта являются локальным хранилищем на узле, TopoLVM позволяет гибко планировать группы контейнеров на узлы с достаточными доступными ресурсами, избегая ситуации, когда группы контейнеров не могут запуститься из-за нехватки ресурсов.
- TopoLVM по умолчанию поддерживает автоматическое расширение томов. После изменения требуемой квоты хранения в Persistent Volume Claim расширение происходит автоматически без перезапуска группы контейнеров.

Сценарии использования

- Когда требуется только временное хранилище, например, для разработки и отладки.
- Когда есть высокие требования к I/O хранилища, например, для индексирования в реальном времени.

Ограничения и предостережения

Рекомендуется использовать локальное хранилище только для приложений, где возможно реализовать репликацию и резервное копирование данных на уровне приложения, например, MySQL. Избегайте потери данных из-за отсутствия гарантии сохранности данных в локальном хранилище.

Развёртывание Volume Plugin

После нажатия кнопки **deploy** на новой открывшейся странице настройте локальное хранилище.

Создание Storage Class

1. Перейдите в Platform Management.
2. В левой навигационной панели выберите **Storage Management > Storage Classes**.
3. Нажмите Create Storage Class.
4. Выберите **TopoLVM**, затем нажмите **Next**.
5. Настройте параметры **Storage Class**, как описано ниже.

Внимание: Вы также можете создать **Storage Class** с помощью YAML.

Параметр	Описание
Name	Имя Storage Class, которое должно быть уникальным в пределах текущего кластера.
Display Name	Имя, которое поможет вам идентифицировать или фильтровать Storage Class, например, описание на русском языке.
Device Class	Device Class — это способ классификации устройств хранения в TopoLVM, где каждый класс соответствует группе устройств с похожими характеристиками. Если нет особых требований, используйте Device Class Automatically Assigned .
File System	XFS — высокопроизводительная журналируемая файловая система, хорошо подходящая для параллельных I/O нагрузок, поддерживает работу с большими файлами и обеспечивает плавную передачу данных.

Параметр	Описание
	EXT4 — журналируемая файловая система в Linux, предоставляющая extent-хранение файлов и поддержку больших файлов, с максимальной ёмкостью файловой системы 1 EiB и максимальным размером файла 16 TiB.
Reclamation Policy	Политика освобождения persistent volumes. • Delete: связанный persistent volume будет удалён вместе с PVC. • Retain: связанный persistent volume останется даже после удаления PVC.
Access Mode	ReadWriteOnce (RWO): может быть смонтирован в режиме чтения-записи только одним узлом.
PVC Reconstruction	Поддержка реконструкции PVC между узлами. При включении необходимо настроить Reconstruction Wait Time . Если узел, на котором размещён PVC, созданный с использованием этого Storage Class, выходит из строя, PVC автоматически восстанавливается на других узлах после указанного времени ожидания для обеспечения непрерывности работы. Внимание: • Восстановленный PVC не содержит исходных данных. • Убедитесь, что количество узлов хранения больше количества реплик экземпляров приложения, иначе это повлияет на реконструкцию PVC.
Allocated Projects	PVC данного типа можно создавать только в определённых проектах. Если в данный момент проект не выделен, его можно обновить позже.

6. После проверки правильности настроек нажмите кнопку Create.

Последующие действия

Когда всё будет готово, вы можете уведомить разработчиков о возможности использования функций **TopoLVM**. Например, создать Persistent Volume Claim и привязать его к Storage Class TopoLVM на странице **Storage > Persistent Volume Claims** в контейнерной платформе.

Создание класса хранения NFS Shared Storage

На основе общественного драйвера хранения **NFS CSI** (Container Storage Interface) предоставляется возможность доступа к нескольким системам хранения или аккаунтам NFS.

В отличие от традиционной клиент-серверной модели доступа к NFS, NFS Shared Storage использует общий плагин хранения NFS CSI (Container Storage Interface), который более соответствует принципам проектирования Kubernetes и позволяет клиентам обращаться к нескольким серверам.

Пререквизиты

Должен быть настроен NFS сервер, а также получены методы доступа к нему. В настоящее время платформа поддерживает три версии протокола NFS: **v3**, **v4.0** и **v4.1**. Вы можете выполнить команду **nfsstat -s** на стороне сервера, чтобы проверить информацию о версии.

Развёртывание плагина NFS Shared Storage

1. Перейдите в Platform Management.
2. В левой навигационной панели нажмите **Storage Management > Storage Classes**.
3. Нажмите Create Storage Class.
4. Справа от **NFS Shared Storage** нажмите Deploy, чтобы перейти на страницу **Plugins**.
5. Справа от плагина **NFS** нажмите : > **Deploy**.
6. Дождитесь, пока статус развёртывания не изменится на **Deployment Successful**, после чего завершите развёртывание.

Создание класса хранения NFS Shared Storage

1. Нажмите Create Storage Class.

Внимание: Вы также можете выполнить операцию с помощью YAML.

2. Выберите NFS Shared Storage и нажмите Next.
3. Следуйте приведённым ниже инструкциям для настройки соответствующих параметров.

Параметр	Описание
Name	Имя класса хранения. Должно быть уникальным в пределах текущего кластера.
Service Address	Адрес доступа к NFS серверу. Например: 192.168.2.11 .
Path	Путь монтирования файловой системы NFS на серверном узле. Например: /nfs/data .
NFS Protocol Version	В настоящее время поддерживаются три версии: v3 , v4.0 и v4.1 .
Reclaim Policy	Политика возврата для persistent volume . - Delete : При удалении persistent volume claim будет также удалён связанный persistent volume. - Retain : Даже при удалении persistent volume claim связанный persistent volume будет сохранён.
Access Modes	Все режимы доступа, поддерживаемые текущим хранилищем. При последующем объявлении persistent volume можно выбрать только один из этих режимов для монтирования persistent volume. - ReadWriteOnce (RWO) : Может быть смонтирован как для чтения и записи одним узлом. - ReadWriteMany (RWX) : Может быть смонтирован как для чтения и записи несколькими узлами. - ReadOnlyMany (ROX) : Может быть смонтирован как только для чтения несколькими узлами.

Параметр	Описание
Allocated Projects	Пожалуйста, выделите проекты, которые могут использовать этот тип хранилища. Если в настоящее время нет проектов, нуждающихся в этом типе хранилища, вы можете не выделять проекты сейчас и обновить их позже.

4. После подтверждения правильности конфигурации нажмите **Create**.

Развёртывание компонента Volume Snapshot

Volume snapshot — это снимок **persistent volume**, представляющий собой копию persistent volume на определённый момент времени. Если в кластере используются persistent volumes с поддержкой функции snapshot, можно развернуть компонент volume snapshot для включения этой возможности.

В настоящее время платформа поддерживает создание volume snapshots только для PVC, которые динамически создаются с использованием storage classes. На основе этих снимков можно создавать новые привязки PVC.

Внимание: Режимы доступа, поддерживаемые при создании PVC из снимков, отличаются от тех, что поддерживаются при создании PVC с помощью storage classes, и выделены **жирным** в таблице ниже.

Storage Class, используемый для создания Volume Snapshots	Single Node Read-Write (RWO)	Multi-Node Read-Only (ROX)	Multi-Node Read-Write (RWX)
TopoLVM	Поддерживается	Не поддерживается	Не поддерживается
CephRBD Block Storage	Поддерживается	Не поддерживается	Не поддерживается
CephFS File Storage	Поддерживается	Поддерживается	Поддерживается

Шаги

1. Перейдите в Platform Management.
2. В левой навигационной панели выберите **Storage Management > Volume Snapshots**.
3. Нажмите **Quick Deployment**, после чего вы будете перенаправлены в плагин кластера.
4. Нажмите на **:** > **Deploy** рядом с **Snapshot Controller** и дождитесь успешного завершения развёртывания.

Создание PV

Ручное создание статического persistent volume типа **HostPath** или **NFS Shared Storage**.

HostPath: Монтирует файловый каталог с хоста, на котором расположен контейнер, в указанный путь внутри контейнера (соответствует HostPath в Kubernetes), позволяя

контейнеру использовать файловую систему хоста для постоянного хранения. Если хост становится недоступен, HostPath может стать недоступен.

NFS Shared Storage: NFS Shared Storage использует общественный плагин хранения NFS CSI (Container Storage Interface), который более соответствует принципам проектирования Kubernetes, предоставляя возможности клиентского доступа для нескольких сервисов. Перед использованием убедитесь, что в текущем кластере развернут **NFS storage plugin**.

Пререквизиты

Подтвердите размер создаваемого **persistent volume** и убедитесь, что бэкенд-хранилище в данный момент имеет возможность предоставить соответствующий объем.

Получите адрес доступа к бэкенд-хранилищу, путь к монтируемой директории, учетные данные доступа (если требуются) и другую необходимую информацию.

Пример PersistentVolume

```
# example-pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: example-pv
spec:
  capacity:
    storage: 5Gi (1)
  accessModes:
    - ReadWriteOnce (2)
  persistentVolumeReclaimPolicy: Retain (3)
  storageClassName: manual (3)
  hostPath: (5)
    path: "/mnt/data"
```

(1) Объем хранилища.
(2) Способ монтирования тома.
(3) Что происходит после удаления PVC (Retain, Delete, Recycle).
(4) Название StorageClass (для динамического связывания).
(5) Тип бэкенд-хранилища.

Веб-консоль

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Storage Management > Persistent Volumes (PV)**.
3. Нажмите Create Persistent Volume.
4. Следуйте инструкциям ниже и настройте параметры перед нажатием **Create**.

Информация о хранилище

Тип	Параметр	Описание
HostPath	Path	Путь к каталогу файлов на узле, который поддерживает том хранилища. Например: /etc/kubernetes .
NFS Shared Storage	Server Address	Адрес доступа к NFS серверу.
	Path	Путь монтирования файловой системы NFS на серверном узле, например /nfs/data .
	NFS Protocol Version	Поддерживаемые на платформе версии протокола NFS: v3, v4.0 и v4.1 . Для просмотра версии на сервере выполните nfsstat -s .

CLI

```
kubectl apply -f example-pv.yaml
```

Режимы доступа

Режимы доступа **persistent volume** зависят от соответствующих параметров, заданных бэкенд-хранилищем.

Режим доступа	Значение
ReadWriteOnce (RWO)	Может быть смонтирован для чтения и записи одним узлом.
ReadWriteMany (RWX)	Может быть смонтирован для чтения и записи несколькими узлами.
ReadOnlyMany (ROX)	Может быть смонтирован только для чтения несколькими узлами.

Политики восстановления

Политика восстановления	Значение
Delete	При удалении persistent volume claim одновременно удаляется связанный persistent volume , а также ресурс тома в бэкенд-хранилище. Внимание: политика восстановления для PV типа NFS Shared Storage не поддерживает Delete .
Retain	Даже при удалении persistent volume claim связанный persistent volume и данные хранилища сохраняются. Требуется ручное управление данными и удаление persistent volume .

Создание PVC

Создайте **PersistentVolumeClaim (PVC)** и при необходимости задайте параметры для запрашиваемого **PersistentVolume (PV)**.

Вы можете создать **PersistentVolumeClaim** либо через визуальную форму UI, либо с помощью пользовательского YAML-файла оркестрации.

Прerequisites

Убедитесь, что в **namespace** достаточно оставшейся квоты хранилища для удовлетворения требуемого размера хранилища при выполнении операции создания.

Пример PersistentVolumeClaim

```
# example-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: example-pvc
  namespace: k-1
  annotations: {}
  labels: {}
spec:
  storageClassName: cephfs
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 4Gi
```

Создание Persistent Volume Claim

Веб-консоль

1. Перейдите в **Container Platform**.
2. В левой боковой панели выберите **Storage > PersistentVolumeClaims (PVC)**.
3. Нажмите Create PVC.
4. Настройте параметры по необходимости.

Внимание: Вы также можете переключиться в режим YAML для выполнения операции.

Параметр	Описание
Name	Имя PersistentVolumeClaim, которое должно быть уникальным в пределах текущего namespace.
Creation Method	- Dynamic Creation: Динамически создаёт PersistentVolume на основе storage class и связывает его.

Параметр	Описание
	- Static Binding: Выполняет сопоставление и связывание на основе настроенных параметров и существующих PersistentVolumes.
Storage Class	После выбора метода динамического создания платформа динамически создаст PersistentVolume в соответствии с описанием в указанном storage class.
Access Mode	- ReadWriteOnce (RWO): Может быть смонтирован одним узлом в режиме чтения и записи. - ReadWriteMany (RWX): Может быть смонтирован несколькими узлами в режиме чтения и записи. - ReadOnlyMany (ROX): Может быть смонтирован несколькими узлами в режиме только для чтения. Внимание: Рекомендуется учитывать количество экземпляров workload, которые планируется связать с текущим PersistentVolumeClaim, а также тип контроллера развертывания. Например, при создании мультиэкземплярного развертывания (Deployment), поскольку все экземпляры используют один PersistentVolumeClaim, не рекомендуется выбирать режим доступа RWO, который может быть подключён только к одному узлу.
Capacity	Размер запрашиваемого PersistentVolume.
Volume Mode	- Filesystem: Связывает PersistentVolume как файловую директорию, монтируемую в Pod. Этот режим доступен для любого типа workload. - Block Device: Связывает PersistentVolume как необработанное блочное устройство, монтируемое в Pod. Этот режим доступен только для виртуальных машин.
More	- Labels - Annotations - Selector: После выбора метода статического связывания можно использовать селектор для выбора PersistentVolumes с определёнными метками. Метки PersistentVolume могут использоваться для обозначения специальных атрибутов хранилища, таких как тип диска или географическое расположение.

5. Нажмите **Create**. Дождитесь, пока статус PersistentVolumeClaim не изменится на Bound, что означает успешное сопоставление PersistentVolume.

CLI

```
kubectl apply -f example-pvc.yaml
```

Операции с PVC

- **Связывание PersistentVolumeClaim:** При создании приложений или workload, требующих постоянного хранения данных, свяжите PersistentVolumeClaim для запроса соответствующего PersistentVolume.
- **Создание PersistentVolumeClaim с использованием Volume Snapshots:** Это помогает создавать резервные копии данных приложений и восстанавливать их по необходимости, обеспечивая надёжность данных бизнес-приложений.

- **Удаление PersistentVolumeClaim:** Вы можете нажать кнопку Actions в правом верхнем углу страницы с деталями для удаления PersistentVolumeClaim при необходимости. Перед удалением убедитесь, что PersistentVolumeClaim не связан с какими-либо приложениями или workload и не содержит volume snapshots. После удаления PersistentVolumeClaim платформа обрабатывает PersistentVolume в соответствии с политикой рекламации, что может привести к очистке данных в PersistentVolume и освобождению ресурсов хранилища. Пожалуйста, действуйте осторожно, учитывая безопасность данных.

Расширение ёмкости PersistentVolumeClaim

Веб-консоль

1. В левой навигационной панели выберите Storage > Persistent Volume Claims (PVC).
2. Найдите нужный persistent volume claim и нажмите : > Expand.
3. Укажите новый размер.
4. Нажмите Expand. Процесс расширения может занять некоторое время, пожалуйста, будьте терпеливы.

CLI

```
kubectl patch pvc example-pvc -n k-1 --type='merge' -p '{
  "spec": {
    "resources": {
      "requests": {
        "storage": "6Gi"
      }
    }
  }
}'
```

Внимание: Если расширение PVC в Kubernetes не удалось, администраторы могут вручную восстановить состояние Persistent Volume Claim (PVC) и отменить запрос на расширение.

Использование снимков томов

Снимок тома — это копия persistent volume claim (PVC) на определённый момент времени, которая может использоваться для настройки новых persistent volume claim (предварительное заполнение данными из снимка) или для отката существующих persistent volume claim к предыдущему состоянию, достигая эффекта резервного копирования данных приложения и их восстановления по необходимости, тем самым обеспечивая надёжность данных приложения.

Пререквизиты

Администратор развернул компонент снимков томов **Snapshot Controller** для текущего кластера и включил функции, связанные со снимками, в кластере хранения.

Persistent volume claim должен быть создан динамически, а его статус должен быть **Bound**.

Storage class, связанный с persistent volume claim, должен поддерживать функциональность снимков, например, **CephRBD Built-in Storage**, **CephFS Built-in Storage** или **TopoLVM**.

Пример ресурса VolumeSnapshot (CR)

Этот пример создаёт снимок PVC с именем example-pvc с использованием CSI snapshot class.

```
# example-snapshot.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: example-pvc-20250527-111124
  namespace: k-1
  labels:
    snapshot.cpaas.io/sourcepvc: example-pvc
  annotations:
    cpaas.io/description: demo
spec:
  volumeSnapshotClassName: csi-cephfs-snapshotclass
  source:
    persistentVolumeClaimName: example-pvc
```

Создание снимков томов

Веб-консоль

Создание снимка тома на основе указанного persistent volume claim (PVC).

Способ 1:

1. Войдите в **Container Platform**.
2. В левой навигационной панели выберите Storage > Persistent Volume Claims (PVC).
3. Нажмите **:** рядом с соответствующим persistent volume claim в списке и выберите **Create Volume Snapshot**.
4. Заполните описание снимка.
5. Нажмите **Create**. Время создания снимка зависит от состояния сети и объёма данных; пожалуйста, подождите.

Когда статус снимка изменится на **Available**, это означает успешное создание.

Способ 2:

1. Войдите в **Container Platform**.
2. В левой навигационной панели выберите Storage > Persistent Volume Claims (PVC).
3. Нажмите на имя persistent volume claim в списке.
4. Перейдите на вкладку **Volume Snapshots**.
5. Нажмите **Create Volume Snapshot** и настройте необходимые параметры.

- Нажмите **Create**. Время создания снимка зависит от состояния сети и объёма данных; пожалуйста, подождите.

Когда статус снимка изменится на **Available**, это означает успешное создание.

CLI

```
kubectl apply -f example-snapshot.yaml
```

Создание снимков томов в произвольном порядке

- Войдите в **Container Platform**.
- В левой навигационной панели выберите **Storage > Volume Snapshots**.
- Нажмите **Create Volume Snapshot** и настройте необходимые параметры.
- Нажмите **Create**. Время создания снимка зависит от состояния сети и объёма данных; пожалуйста, подождите.

Когда статус снимка изменится на **Available**, это означает успешное создание.

Создание persistent volume claims из снимков томов

В настоящее время платформа поддерживает создание снимков томов только для PVC, созданных из storage class с **Dynamic Provisioning**. Вы можете создавать новые PVC на основе этих снимков и связывать их.

Внимание: Режимы доступа, поддерживаемые при создании PVC из снимка, отличаются от поддерживаемых при создании PVC из storage class, что выделено **жирным** в таблице.

Storage Class, используемый для создания снимков томов	Single Node Read- Write (RWO)	Multi-Node Read- Only (ROX)	Multi-Node Read- Write (RWX)
TopoLVM	Поддерживается	Не поддерживается	Не поддерживается
CephRBD Block Storage	Поддерживается	Не поддерживается	Не поддерживается
CephFS File Storage	Поддерживается	Поддерживается	Поддерживается

Способ 1:

- Войдите в **Container Platform**.
- В левой навигационной панели выберите **Storage > Persistent Volume Claims (PVC)**.
- Нажмите на имя persistent volume claim в списке.
- Перейдите на вкладку **Volume Snapshots**.
- Нажмите **:** рядом с соответствующим снимком тома в списке и выберите **Create Persistent Volume Claim**, настройте необходимые параметры.
- Нажмите **Create**.

Способ 2:

1. Войдите в **Container Platform**.
2. В левой навигационной панели выберите **Storage > Volume Snapshots**.
3. Нажмите **:** рядом с соответствующим снимком тома в списке и выберите **Create Persistent Volume Claim**, настройте необходимые параметры.
4. Нажмите **Create**.

Generic ephemeral volumes

Generic Ephemeral Volumes в Kubernetes — это функция, которая позволяет создавать эфемерные (временные) тома на уровне пода с использованием существующих StorageClasses и CSI-драйверов, без необходимости предварительного определения PersistentVolumeClaims (PVC).

Они сочетают гибкость динамического выделения с простотой объявления томов на уровне пода.

Это временные тома, которые автоматически:

- создаются при запуске Pod
- удаляются при завершении Pod

Используют те же базовые механизмы, что и PersistentVolumeClaim.

Требуют CSI (Container Storage Interface) драйвер с поддержкой динамического выделения.

Пример ephemeral volumes

Этот пример автоматически создаёт временный PVC для Pod с использованием указанного StorageClass.

```
apiVersion: v1
kind: Pod
metadata:
  name: ephemeral-demo
spec:
  containers:
    - name: app
      image: busybox
      command: ["sh", "-c", "echo hello > /data/hello.txt && sleep 3600"]
      volumeMounts:
        - mountPath: /data
          name: ephemeral-volume
  volumes:
    - name: ephemeral-volume
      ephemeral: (1)
      volumeClaimTemplate:
        metadata:
          labels:
```

```
type: temporary
spec:
  accessModes: [ "ReadWriteOnce" ]
  resources:
    requests:
      storage: 1Gi
  storageClassName: standard
```

(1) - **Pod** создаст PVC, используя этот шаблон.

Основные характеристики

Характеристика	Описание
Эфемерный	Том удаляется при удалении Pod
Динамическое выделение	Поддерживается любым CSI-драйвером с динамическим выделением
Без отдельного PVC	VolumeClaim встроен непосредственно в спецификацию Pod
Работает через CSI	Совместим с любым CSI-драйвером (EBS, RBD, Longhorn и др.)

Рекомендуем использовать Generic Ephemeral Volumes для случаев:

Когда требуется временное хранилище с такими возможностями, как:

- Изменяемый размер томов
- Снимки (snapshots)
- Шифрование
- Хранилище, не привязанное к локальному узлу (например, облачное блочное хранилище)

Идеально подходит для:

- Кэширования промежуточных данных
- Временных рабочих директорий
- Конвейеров, AI/ML рабочих процессов

Использование emptyDir

В Kubernetes `emptyDir` — это простой временный тип тома, который предоставляет временное хранилище для пода на время его жизни. Он создаётся, когда под назначается на узел, и удаляется, когда под удаляется с этого узла.

Пример emptyDir

Этот Pod создаёт временный том, смонтированный в `/data`, который используется контейнером.

Опциональная настройка Medium

Вы можете выбрать, где будут храниться данные:

`emptyDir:`

medium: "Memory"

Medium	Описание
(default)	Использует диск узла, SSD или сетевое хранилище, в зависимости от среды
Memory	Использует RAM (tmpfs) для более быстрого доступа (но данные нестабильны)

Основные характеристики

Особенность	Описание
Начинается пустым	Нет данных при создании
Совместное использование	Один и тот же том может использоваться несколькими контейнерами в поде
Удаляется с подом	Том уничтожается при удалении пода
Локальный для узла	Том хранится на локальном диске или в памяти узла
Быстрый	Идеально подходит для производительного временного хранилища

Распространённые сценарии использования

- Кэширование промежуточных артефактов сборки
- Буферизация логов
- Временные рабочие директории
- Совместное использование данных между контейнерами в одном поде (например, сайдкары)

Интеграция стороннего хранилища

С ростом использования как публичных, так и частных облачных сред интеграция сторонних хранилищ становится всё более важной. В этом руководстве описывается, как аннотировать возможности стороннего хранилища с помощью ConfigMap, чтобы ваша платформа могла автоматически распознавать и отображать эти возможности.

Шаг 1. Открытие конфигурации Storage Class

1. Перейдите в **Platform Management** в интерфейсе вашей платформы.
2. В левой боковой панели выберите Storage Management > Storage Classes.
3. Нажмите **Create Storage Class**, чтобы начать определение нового класса хранилища.

Шаг 2. Заполнение информации о Storage Class

Укажите следующие данные в форме:

Поле	Описание
Name	Название вашего нового класса хранилища.

Storage Class	Выберите или определите идентификатор класса хранилища.
Provisioner	Введите имя provisioner, используемого вашим плагином хранилища.

Шаг 3. Аннотация возможностей хранилища с помощью ConfigMap

Чтобы включить аннотации возможностей, создайте **ConfigMap** в пространстве имён **kube-public** с соответствующей меткой и форматом данных.

Пример YAML:

```

apiVersion: v1
kind: ConfigMap
metadata:
  name: sd-built-in
  namespace: kube-public
  labels:
    features.alauda.io/type: StorageDescription
data:
  storage.type1.com: |-
    type: Filesystem
    volumeMode:
      - Filesystem
    accessModes:
      - ReadWriteOnce
      - ReadWriteMany
      - ReadWriteOncePod
  storage.type2.com: |-
    type: Filesystem
    snapshot: true
    volumeMode:
      - Filesystem
      - Block
    accessModes:
      - ReadWriteOnce
      - ReadOnlyMany
      - ReadWriteOncePod

```

Основные моменты:

metadata.name: должен начинаться с sd-, например, sd-configmap1.

metadata.namespace: должен быть kube-public.

metadata.labels: должен содержать features.alauda.io/type = StorageDescription.

data:

- Каждый ключ соответствует полю provisioner в классе хранилища.

- Каждое значение — это YAML-строка, описывающая поддерживаемые возможности хранилища.

Шаг 4. Завершение создания Storage Class

После заполнения всех данных:

1. Нажмите **Create**, чтобы сохранить класс хранилища.
2. Платформа автоматически сопоставит **provisioner** с ConfigMap и аннотирует класс хранилища определёнными возможностями.

Распределённое хранилище Ceph в MWS-CP

MWS Container Platform (MWS-CP) Storage с Ceph — это гиперконвергентное решение для хранения данных, предоставляемое платформой внутри кластера. Основанное на open-source решении Rook + Ceph, распределённое хранилище обеспечивает автоматическое управление, автоматическое масштабирование и автоматическое восстановление, удовлетворяя потребности малых и средних приложений в блочном, файловом и объектном хранении.

В данном документе **распределённое хранилище** означает Ceph-хранилище внутри данного кластера, а **внешнее хранилище** — Ceph-хранилище вне этого кластера.

Обзор возможностей

Простое развертывание: Предоставляет графические сервисы для автоматического развертывания и управления кластерами хранения; поддерживает как интегрированный, так и отдельный режим развертывания вычислений и хранения.

Профессиональная эксплуатация: Обеспечивает функции создания снимков постоянных томов, резервного копирования и клонирования новых томов; визуальный мониторинг ёмкости, производительности и состояния компонентов; оснащён встроенными политиками оповещений для удовлетворения большинства сценариев эксплуатации хранилища.

Безопасность и надёжность: Распределённый механизм с несколькими репликами гарантирует безопасность и надёжность данных; простое и надёжное автоматизированное управление поддерживает онлайн-расширение ресурсов хранения.

Отличная производительность: Предоставляет эластичные и высокопроизводительные сервисы хранения; поддерживает развертывание гибридных дисковых устройств для повышения производительности и эффективности системы хранения.

Сравнение решений для хранения

Платформа поддерживает два типа решений для хранения; вы можете выбрать одно из них.

Создание кластера хранения

Требование	Преимущества

Вы можете выбрать создание либо кластера стандартного типа, либо кластера расширенного типа	Нет необходимости в дополнительной подготовке решения для хранения; конфигурация может быть выполнена на бизнес-кластере, что экономит затраты.

Доступ к внешнему хранилищу

Вариант 1: Использовать распределённые ресурсы хранения других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что упрощает управление и обслуживание.

Вариант 2: Интегрировать внешние Ceph-ресурсы хранения как распределённое хранилище.

Требование (выберите один)	Преимущества
Вариант 1: Распределённое хранилище уже развернуто в других бизнес-кластерах.	Позволяет полноценно использовать ресурсы хранения между кластерами и избегать влияния изменений в бизнесе. Обеспечивает безопасность и стабильность данных при снижении сложности эксплуатации. Примечание: Если хранилище, к которому осуществляется доступ, является распределённым хранилищем с других платформ, например, основной/резервной платформой в среде аварийного восстановления, используйте метод интеграции внешнего Ceph.
Вариант 2: Внешнее Ceph-хранилище вне платформы, версия $\geq$ 14.2.3.	По сравнению с прямым созданием класса хранения, этот метод удобнее для использования интерфейса платформы для создания снимков томов, масштабирования и других функций.

Внимание: Если необходимо поддерживать пул хранения, устройства хранения и другие конфигурации внешнего хранилища, операции должны выполняться в интерфейсе управления кластера хранения.

Создание кластера стандартного типа

Кластер стандартного типа — это наиболее типичный способ развертывания хранилища Ceph. Он распределяет реплики данных по жестким дискам на разных хостах, обеспечивая, что в случае отказа одного хоста копии данных на других хостах смогут сохранить доступность сервиса.

Пререквизиты

- В кластере хранения должно быть не менее 3 узлов.
- Каждый узел должен иметь как минимум 1 пустой жесткий диск или 1 неотформатированный раздел жесткого диска.
- Рекомендуемый объем доступного жесткого диска должен быть больше 50 ГБ.
- Если вы используете присоединённый Kubernetes-кластер с Containerd в качестве компонента runtime, убедитесь, что параметр **LimitNOFILE** в файле **/etc/systemd/system/containerd.service** на всех узлах кластера установлен в

значение **1048576**, чтобы обеспечить успешное развертывание распределённого хранилища.

Внимание: При обновлении с версий ранее v3.10.2 до текущей версии, если необходимо развернуть распределённое хранилище Ceph на вашем кастомном Kubernetes-кластере с Containerd в качестве runtime, также требуется установить параметр **LimitNOFILE** в файле **/etc/systemd/system/containerd.service** на всех узлах кластера в значение **1048576**.

Меры предосторожности

Создание сервисов хранения и Доступ к сервисам хранения поддерживают выбор только одного метода.

Шаги

1. Развёртывание Operator
 - 1) Перейдите в раздел **Platform Management**.
 - 2) В левой боковой панели нажмите Storage Management > Distributed Storage.
 - 3) Нажмите Configure Now.
 - 4) На странице мастера **Deploy Operator** нажмите кнопку **Deploy Operator** в правом нижнем углу.
 - Если страница автоматически перейдет к следующему шагу, это означает успешное развертывание Operator.
 - Если развертывание не удалось, следуйте подсказке на интерфейсе **Clean Up Deployed Information and Retry** и повторно разверните Operator; чтобы вернуться на страницу выбора распределённого хранилища, нажмите **Application Store**, сначала удалите ресурсы уже развернутого rook-operator, затем удалите сам rook-operator.
2. Создание кластера
 - 1) На странице мастера **Create Cluster** настройте соответствующие параметры и нажмите кнопку **Create Cluster** в правом нижнем углу.

Параметр	Описание
Cluster Type	Выберите Standard .
Device Class Type	Классы устройств — это группировки жестких дисков; вы можете настраивать device class в соответствии с вашими потребностями хранения, распределяя разный контент по дискам с разной производительностью. Default Device Class: Платформа автоматически классифицирует типы жестких дисков на узлах кластера. Например, создаются device class с именами hdd, ssd, nvme. Custom Device Class: Настройте имя device class для конкретных комбинаций дисков на узле; поддерживается добавление нескольких device class. Один и тот же жесткий диск может принадлежать только одному device class.

Параметр	Описание
Device Class - Name	Имя device class . При выборе Custom Device Class имя device class не может быть hdd, ssd, nvme .
Device Class - Storage Devices	Выберите Blank Hard Disk или Unformatted Hard Disk Partition на узлах. Если переключатель "Open All Blank Devices" включен: все пустые устройства на узле будут добавлены в device class; Если переключатель "Open All Blank Devices" выключен: вручную введите имена пустых устройств на узле, например sda .
Snapshot	При включении поддерживается создание снимков PVC и использование снимков для конфигурации новых PVC для быстрого резервного копирования и восстановления данных. Если при создании хранилища снимки не были включены, их можно включить при необходимости в разделе Operations на странице деталей кластера хранения. Внимание: Перед использованием убедитесь, что для текущего кластера развернуты плагины volume snapshot.
Monitoring Alarm	При включении предоставляются готовые возможности сбора метрик мониторинга и оповещений. Внимание: Если не включить сейчас, потребуется искать альтернативные решения для мониторинга и оповещений хранилища, например, вручную настраивать дашборды мониторинга и стратегии оповещений в центре эксплуатации и обслуживания.

2) Нажмите **Advanced Configuration** для расширенной настройки компонентов.

Параметр	Описание
Network Configuration	Host Network: Кластер хранения будет использовать сеть хоста, необходимо заполнить соответствующие параметры оптимизации сети в колонке параметров оптимизации, например, указать подсети public и cluster . Если оставить пустым, будет использована подсеть хоста по умолчанию. Внимание: Использование сети хоста может представлять угрозу безопасности из-за передачи данных в незашифрованном (открытом) виде через порты хоста. Обратитесь в службу поддержки платформы для получения решения по шифрованию передачи. Container Network: Кластер хранения будет использовать контейнерную сеть; можно создать подсети в управлении сетью и назначить их пространству имён rook-ceph. Если оставить пустым, будет использована подсеть по умолчанию. Внимание: IPv6 не поддерживается. При использовании контейнерной сети доступ к хранилищу возможен только внутри кластера.

Параметр	Описание
	Сбои или перезапуски Pod Ceph CSI могут привести к прерыванию сервиса.
Optimization Parameters	Поддерживается заполнение параметров в формате конфигурационного файла Ceph; система перезапишет параметры по умолчанию на основе предоставленного содержимого. Внимание: После первого заполнения или изменения параметров инициализации необходимо нажать на параметры инициализации; успешная инициализация обязательна для создания кластера.
Component Fixed-point Deployment	Можно развернуть компоненты на указанных узлах; требуется минимум три узла для обеспечения минимальной доступности. Компоненты, доступные для настройки фиксированного развертывания: MON, MGR, MDS, RGW.

- Если страница автоматически перейдет к следующему шагу, это означает успешное развертывание кластера Ceph.
- Если создание не удалось, можно нажать очистку **Created Information or Retry** для автоматической очистки ресурсов и повторного создания кластера, либо вручную очистить ресурсы.

3. Создание пула хранения

- 1) На странице мастера **Create Storage Pool** настройте соответствующие параметры и нажмите кнопку **Create Storage Pool** в правом нижнем углу.

Параметр	Описание
Storage Type	- File Storage: обеспечивает безопасные, надежные и масштабируемые услуги совместного файлового хранения. Подходит для совместного использования файлов, резервного копирования данных и т.д. - Block Storage: предоставляет хранилище с высокой производительностью IOPS и низкой задержкой. Подходит для баз данных, виртуализации и т.д. - Object Storage: предоставляет хранилище с интерфейсом S3, подходит для больших данных, резервного архивирования, облачного хранения и т.д.
Replica Count	Чем больше количество реплик, тем выше избыточность и безопасность данных; однако снижается эффективность использования хранилища. Обычно устанавливается значение 3, что удовлетворяет большинству потребностей.
Device Class	Единообразно классифицируйте типы для одного типа устройств или дисков с одинаковой бизнес-логикой, выбирая из device class, добавленных на предыдущем шаге.

Параметр	Описание
	При выборе device class данные будут храниться в выбранном device class. Если device class не выбран, данные будут случайным образом распределены по всем устройствам пула хранения.

- Если это объектное хранилище, необходимо также настроить следующие параметры:

Параметр	Описание
Region	Укажите регион, в котором расположен пул хранения.
Gateway Type	По умолчанию S3, изменить нельзя.
Internal Port	Укажите порт для внутреннего доступа в кластере.
External Access	Включение/отключение внешнего доступа создаст/удалит сервис типа Nodeport.
Instance Count	Количество экземпляров ресурсов для объектного хранилища.

- Если страница автоматически перейдет к следующему шагу, это означает успешное развертывание пула хранения.
- Если развертывание не удалось, следуйте подсказкам интерфейса для проверки основных компонентов, затем нажмите **Clean Up Created Information and Retry** для повторного создания пула хранения.

2) Нажмите **Create Storage Pool**. На вкладке **Details** можно просмотреть информацию о созданном пуле хранения.

Создание Stretch Type Кластера

Stretch кластер может распространяться на две географически разнесённые локации, обеспечивая возможности аварийного восстановления для инфраструктуры хранения данных. В случае катастрофы, когда одна из зон доступности в двух зонах полностью недоступна, Serp всё равно может сохранять доступность.

Терминология

Термин	Объяснение
Quorum Availability Zone	Обычно располагается в отдельной зоне, которая не несёт основных рабочих нагрузок, сосредоточена на поддержании согласованности кластера и используется преимущественно для принятия арбитражных решений при сбоях в основном дата-центре или сетевых разрывах.
Data Availability Zone	Основная зона в кластере Serp, где фактически хранятся и обрабатываются данные, несущая операционные нагрузки и задачи

	хранения данных, вместе с зоной кворума формирующая полноценную систему хранения с высокой доступностью.

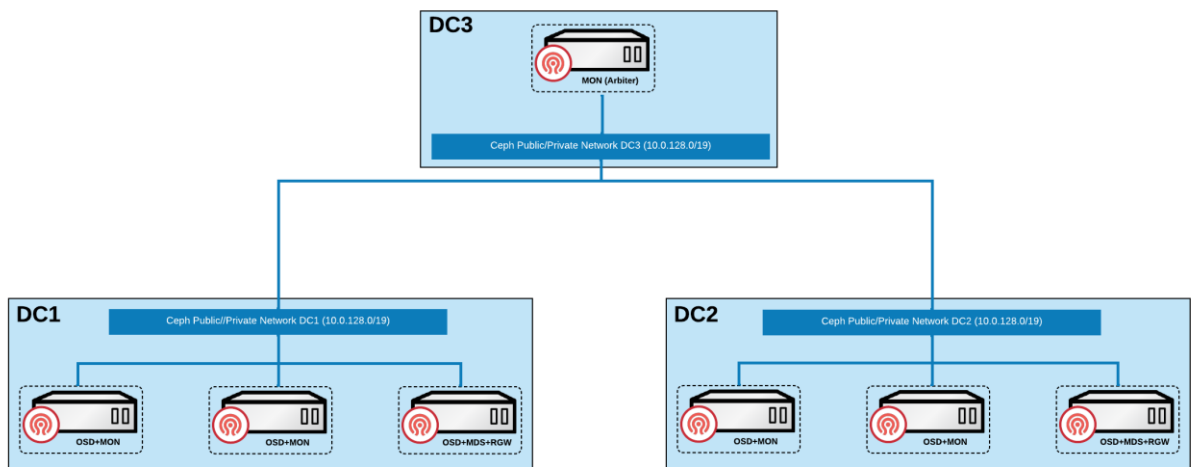
Типовая схема развёртывания

Ниже приведено типовое решение по развёртыванию stretch кластера, а также описание компонентов и принципы аварийного восстановления.

Описание компонентов

Узлы необходимо распределить по трём зонам доступности, включая две зоны доступности данных и одну зону кворума.

- В обеих зонах доступности данных необходимо полностью развернуть все основные компоненты Ceph (MON, OSD, MGR, MDS, RGW), при этом в каждой зоне доступности данных должно быть настроено по два экземпляра MON для обеспечения высокой доступности. Если оба экземпляра MON в одной зоне доступности данных становятся недоступны, система определяет эту зону как находящуюся в состоянии сбоя.
- В зоне кворума требуется развернуть только один экземпляр MON, который служит узлом для принятия арбитражных решений.



Описание аварийного восстановления

- При полном отказе зоны доступности данных Ceph кластер автоматически переходит в деградированное состояние и генерирует уведомление об аварии. Система изменит минимальное количество реплик в пуле хранения (`min_size`) с значения по умолчанию 2 на 1. Поскольку другая зона доступности данных сохраняет двойное дублирование, кластер остаётся доступным. После восстановления отказавшей зоны система автоматически выполнит синхронизацию данных и вернётся в здоровое состояние; если сбоя не удастся устранить, рекомендуется заменить её на новую зону доступности данных.

- При разрыве сетевого соединения между двумя зонами доступности данных, но при сохранении нормального подключения к зоне кворума, зона кворума на основе заранее заданных политик проведёт арбитраж между двумя зонами данных, выбрав ту, которая находится в лучшем состоянии, для продолжения предоставления сервисов в качестве основной зоны данных.

Ограничения и лимиты

- **Ограничения по пулам хранения:** Пулы хранения с erasure coding не поддерживаются, допускается только механизм репликации для защиты данных.
- **Ограничения по классификации устройств:** Функциональность device class не поддерживается, хранение не может быть стратифицировано по характеристикам устройств.
- **Ограничения по региональному развертыванию:** Поддерживаются только две зоны доступности данных; не допускается более двух зон доступности данных.
- **Требования к балансировке данных:** Вес OSD в двух зонах доступности данных должен строго совпадать для обеспечения сбалансированного распределения данных.
- **Требования к носителям хранения:** Разрешена только конфигурация All-Flash OSD, что минимизирует время восстановления после восстановления соединения и максимально снижает риск потери данных.
- **Требования к сетевой задержке:** RTT (время кругового обхода) между двумя зонами доступности данных не должен превышать 10 мс, а зона кворума должна соответствовать требованиям по задержкам спецификации ETCD для обеспечения надёжности арбитражного механизма.

Предварительные условия

Заранее необходимо классифицировать все или часть узлов кластера по трём зонам доступности следующим образом:

- Обеспечить распределение не менее 5 узлов между одной зоной кворума и двумя зонами доступности данных. При этом в зоне кворума должен быть минимум один узел, который может быть виртуальной машиной или облачным хостом.
- Обеспечить наличие как минимум одного Master узла (контрольного узла) в одной из трёх зон доступности.
- Обеспечить равномерное распределение не менее 4 вычислительных узлов по 2 зонам доступности данных, при этом в каждой зоне доступности данных должно быть минимум по 2 вычислительных узла.
- По возможности обеспечить одинаковое количество узлов и конфигурацию дисков в двух зонах доступности данных.

Процедура

1. Тегирование узлов

- 1) Перейдите в раздел **Platform Management**.

- 2) В левой боковой панели выберите **Cluster Management > Cluster**.
- 3) Кликните по названию соответствующего кластера, чтобы перейти на страницу обзора кластера.
- 4) Переключитесь на вкладку Nodes.
- 5) В соответствии с планированием добавьте метку **topology.kubernetes.io/zone=<zone>** этим узлам для классификации их в указанную зону доступности. Здесь вместо <zone> укажите имя зоны доступности.

2. Создание сервисов хранения

В данном разделе описаны только параметры, отличающиеся от стандартных кластеров.

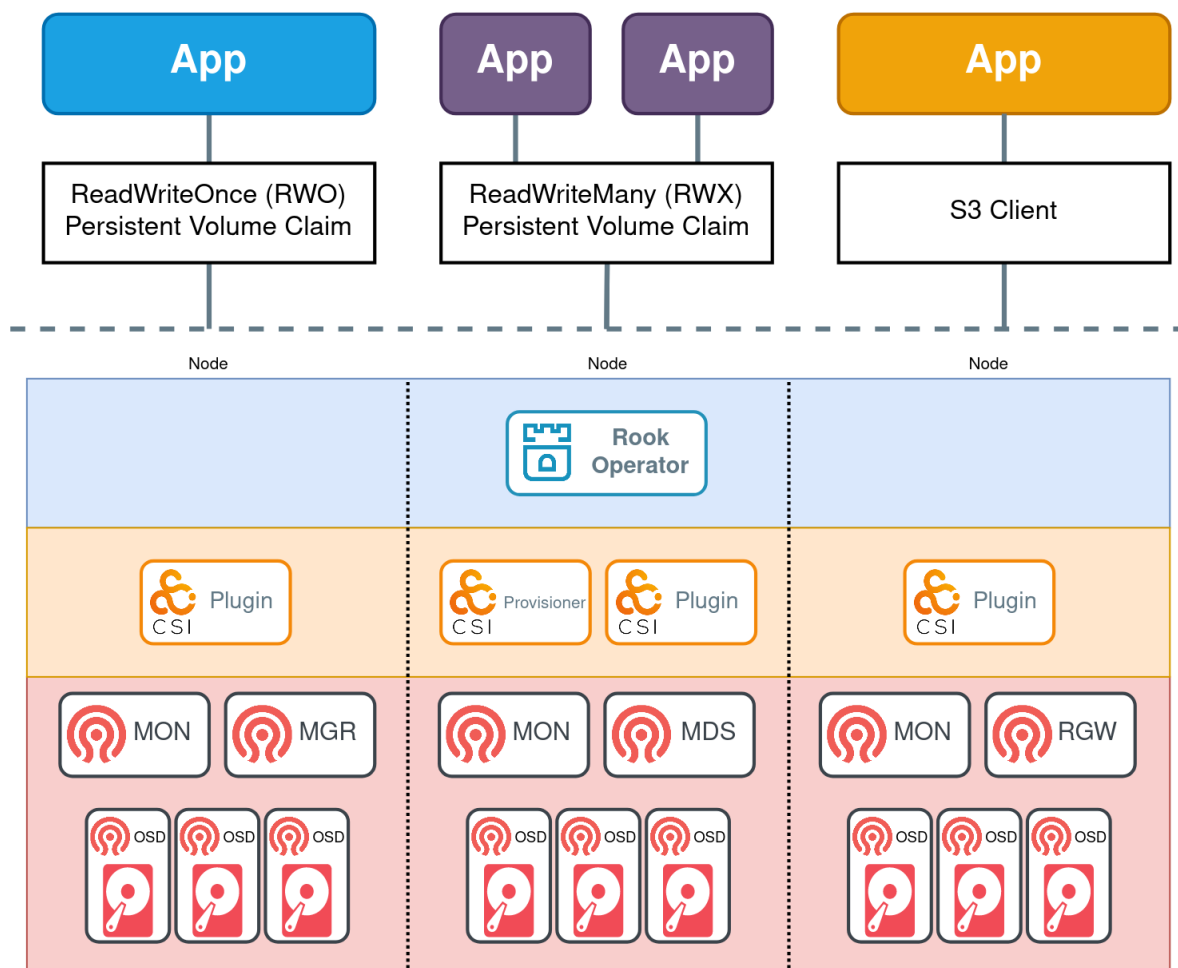
Создание кластера

Параметр	Описание
Cluster Type	Выберите Stretch.
Quorum Availability Zone	Выберите имя зоны кворума.
Data Availability Zone	Выберите имена зон доступности и укажите узлы.

Создание пула хранения

Параметр	Описание
Number of Replicas	Значение по умолчанию — 4.
Number of Instances	При типе хранения Object Storage для обеспечения доступности минимальное количество экземпляров — 2, максимальное — 5.

Rook Architecture



На рисунке выше показаны примерные приложения для трёх поддерживаемых типов хранилищ:

- Блочное хранилище представлено синим приложением, к которому подключён том с режимом ReadWriteOnce (RWO). Приложение может читать и записывать данные в том RWO, при этом Ceph управляет вводом-выводом.
- Общая файловая система представлена двумя фиолетовыми приложениями, которые совместно используют том с режимом ReadWriteMany (RWX). Оба приложения могут одновременно активно читать и записывать данные в том. Ceph обеспечивает безопасную защиту данных для нескольких писателей с помощью демона MDS.
- Объектное хранилище представлено оранжевым приложением, которое может читать и записывать данные в бакет с помощью стандартного S3 клиента.

Ниже пунктирной линии на диаграмме компоненты разделены на три категории:

- Rook operator (синий слой): оператор автоматизирует конфигурацию Ceph
- CSI plugins and provisioners (оранжевый слой): драйвер Ceph-CSI обеспечивает создание и монтирование томов
- Ceph daemons (красный слой): демоны Ceph запускают основную архитектуру хранилища. Подробнее о каждом демоне см. в Глоссарии.

Блочное хранилище

На диаграмме выше процесс создания приложения с томом RWO следующий:

- (Синее) приложение создаёт PVC для запроса хранилища.
- PVC определяет класс хранения Ceph RBD (sc) для создания хранилища.
- K8s вызывает provisioner Ceph-CSI RBD для создания образа Ceph RBD.
- kubelet вызывает CSI плагин RBD для монтирования тома в приложении.
- Том становится доступен для чтения и записи.
- Том ReadWriteOnce может быть смонтирован только на одном узле одновременно.

Общая файловая система

На диаграмме выше процесс создания приложений с томом RWX следующий:

- (Фиолетовое) приложение создаёт PVC для запроса хранилища.
- PVC определяет класс хранения CephFS (sc) для создания хранилища.
- K8s вызывает provisioner Ceph-CSI CephFS для создания подтома CephFS.
- kubelet вызывает CSI плагин CephFS для монтирования тома в приложении.
- Том становится доступен для чтения и записи.
- Том ReadWriteMany может быть смонтирован на нескольких узлах для использования приложением.

Объектное хранилище S3

На диаграмме выше процесс создания приложения с доступом к бакету S3 следующий:

- (Оранжевое) приложение создаёт BucketClaim для запроса бакета.
- Драйвер Ceph COSI создаёт бакет Ceph RGW.
- Драйвер Ceph COSI создаёт секрет с учётными данными для доступа к бакету.
- Приложение получает учётные данные из секрета.
- Приложение теперь может читать и записывать данные в бакет с помощью S3 клиента.

Основные концепции

Rook Operator

Оператор Rook — это простой контейнер, который содержит всё необходимое для инициализации и мониторинга кластера хранения. Оператор запускает и контролирует поды мониторов Ceph, демоны Ceph OSD для предоставления хранилища RADOS, а также запускает и управляет другими демонами Ceph. Оператор управляет CRD для пулов,

объектных хранилищ (S3/Swift) и файловых систем, инициализируя поды и другие ресурсы, необходимые для работы сервисов.

Оператор следит за демонами хранения, чтобы обеспечить здоровье кластера. Мониторы Ceph будут запускаться или переключаться при необходимости, а также вноситься другие корректировки по мере роста или уменьшения кластера. Оператор также отслеживает изменения желаемого состояния, указанные в пользовательских ресурсах Ceph (CR), и применяет эти изменения.

Rook автоматически настраивает драйвер Ceph-CSI для монтирования хранилища к вашим подам. Образ rook/ceph включает все необходимые инструменты для управления кластером.

Ceph CSI

Плагины Ceph CSI реализуют интерфейс между CSI-совместимым оркестратором контейнеров (CO) и кластерами Ceph. Они обеспечивают динамическое выделение томов Ceph и их подключение к рабочим нагрузкам.

Функции модулей Ceph

Модуль	Функция
MON	Монитор (MON) — самый важный компонент в кластере Ceph. Он управляет кластером Ceph и поддерживает статус всего кластера. MON обеспечивает синхронизацию связанных компонентов кластера одновременно. Он выполняет роль лидера кластера и отвечает за сбор, обновление и публикацию информации о кластере.
MGR	Менеджер (MGR) — это система мониторинга, которая обеспечивает сбор, хранение, анализ (включая оповещения) и визуализацию данных. Он делает определённые параметры кластера доступными для внешних систем.
OSD	Демоны объектного хранилища (OSD) хранят фактические пользовательские данные. Каждый OSD обычно привязан к одному физическому диску. OSD обрабатывают запросы на чтение и запись от клиентов.
MDS	Сервер метаданных Ceph (MDS) отслеживает иерархию файлов и хранит метаданные, используемые только для CephFS. RBD и RGW не требуют метаданных. MDS не предоставляет клиентам прямые сервисы данных.
RGW	Шлюз RADOS (RGW) — это объектный шлюз Ceph, который предоставляет RESTful API, совместимые с S3 и Swift. RGW также поддерживает мультиарендность и сервис идентификации OpenStack (Keystone).
RADOS	Надёжное автономное распределённое объектное хранилище (RADOS) — это ядро кластера хранения Ceph. Всё в Ceph хранится в виде объектов через RADOS, независимо от типа данных. Слой RADOS обеспечивает согласованность и надёжность данных через репликацию, обнаружение и восстановление сбоев, а также восстановление данных между узлами кластера.
LIBRADOS	Librados — это метод, упрощающий доступ к RADOS. В настоящее время он поддерживает языки программирования PHP, Ruby, Java, Python, C и C++. Он предоставляет RADOS — локальный интерфейс к кластеру хранения Ceph, и

Модуль	Функция
	является базовым компонентом других сервисов, таких как блочное устройство RADOS (RBD) и шлюз RADOS (RGW). Кроме того, он предоставляет интерфейс Portable Operating System Interface (POSIX) для файловой системы Ceph (CephFS). API Librados можно использовать для прямого доступа к RADOS, что позволяет разработчикам создавать собственные интерфейсы для доступа к хранилищу кластера Ceph.
RBD	Блочное устройство RADOS (RBD) — это блочное устройство Ceph, которое предоставляет блочное хранилище для внешних систем. Его можно отображать, форматировать и монтировать как диск на сервере.
CephFS	CephFS предоставляет распределённую файловую систему, совместимую с POSIX, любого размера. Она зависит от Ceph MDS для отслеживания иерархии файлов, то есть метаданных.

Доступ к сервисам хранения

Доступ к сервисам хранения поддерживает два способа интеграции: во-первых, интеграция распределённых ресурсов хранения из других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что облегчает управление и сопровождение; во-вторых, подключение внешних ресурсов хранения Ceph для использования распределённого хранения.

Пререквизиты

Выберите один из вариантов:

- В других бизнес-кластерах развернуто распределённое хранилище, создан пул хранения. Запишите имя пула хранения для последующего использования при интеграции.
- Вне платформы создано внешнее хранилище Ceph (версия $\geq 14.2.3$) с пулом хранения. Запишите имя пула хранения для последующего использования при интеграции.

Открытие портов

IP назначения	Порты назначения	IP источника	Порт источника
IP узла Ceph	3300, 6789, 6800-7300, 7480	IP всех узлов бизнес-кластера	любой

Получение данных аутентификации (внешний Ceph)

Если подготовленное хранилище — внешнее Ceph, необходимо получить данные аутентификации с помощью следующих команд.

Параметр	Способ получения
FSID	ceph fsid

Параметр	Способ получения
Информация о компоненте MON	ceph mon dump Должна быть в формате {name= IP}, например a=192.168.100.100:6789 .
Административный ключ	ceph auth get-key client.admin
Пул хранения	Файловое хранилище: используйте команду ceph fs ls для получения значения name. Блочное хранилище: ceph osd dump grep "application rbd" awk '{print \$3}'
Пул хранения данных	(требуется только для файлового хранилища) используйте команду ceph fs ls для получения значения data pools.

Шаги

Внимание: В следующих шагах в качестве примера рассматривается доступ к внешнему хранилищу Ceph, операции для доступа к распределённому хранилищу аналогичны.

1. В левой навигационной панели **нажмите Storage Management > Distributed Storage**.
2. Нажмите **Access Storage**.
3. На странице мастера **Access Configuration** выберите **External Ceph**.

Параметр	Описание
Snapshot	При включении поддерживается создание снимков PVC и использование снимков для настройки новых PVC для быстрого резервного копирования и восстановления бизнес-данных. Если снимки не были включены при доступе к хранилищу, их можно включить позже в разделе Operations на странице деталей кластера хранения по мере необходимости. Внимание: Перед использованием убедитесь, что для текущего кластера развернут плагин volume snapshot.
Network Configuration	Host Network: Вычислительные компоненты в этом кластере будут обращаться к кластеру хранения через host network . Container Network: Вычислительные компоненты в этом кластере будут обращаться к кластеру хранения через container network . Можно создать подсеть в управлении сетью и назначить её пространству имён rook-ceph . Если оставить пустым, будет использована подсеть по умолчанию.
Other Parameters	Заполните параметры аутентификации для внешнего Ceph, полученные на этапе предварительных требований.

4. На странице мастера **Create Storage Class** завершите настройку и нажмите **Access**.

Параметр	Описание
Type	В зависимости от типа созданного пула хранения, по умолчанию будет соответствующий класс хранения:

Параметр	Описание
	Файловое хранилище: CephFS File Storage Блочное хранилище: CephRBD Block Storage
Reclaim Policy	Политика возврата для постоянных томов. Delete: при удалении запроса на постоянный том, связанный постоянный том также будет удалён. Retain: даже при удалении запроса на постоянный том, связанный постоянный том останется.
Project Allocation	Проекты, которые могут использовать этот тип хранилища. Если в данный момент нет проектов, требующих этот тип хранилища, можно не выделять проекты сейчас и обновить их позже.

5. Дождитесь успешной интеграции, это займет примерно 1-5 минут.

Управление Storage Pools

Storage pool — это логический раздел, используемый для хранения данных. Один кластер хранения поддерживает одновременное использование различных типов storage pools, таких как файловое хранилище и блочное хранилище, чтобы удовлетворить разные бизнес-требования.

Создание Storage Pool

Помимо storage pools, созданных при настройке распределённого хранилища, вы также можете создать дополнительные типы storage pools.

Внимание: В рамках одного кластера хранения допускается наличие только одного файлового и одного объектного storage pool, при этом можно создать до восьми блочных storage pools.

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите Storage Management > Distributed Storage.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool** и нажмите **Create Storage Pool**.
4. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Storage Type	Выберите тип хранилища, который ещё не развернут. - File Storage: Обеспечивает безопасные, надёжные и масштабируемые услуги совместного файлового хранения. Подходит для совместного использования файлов, резервного копирования данных и т.д. - Block Storage: Обеспечивает хранилище с высокой производительностью IOPS и низкой задержкой. Подходит для баз данных, виртуализации и т.д. - Object Storage: Предоставляет стандартный интерфейс S3, подходит

Параметр	Описание
	для big data, резервного архивирования, облачных сервисов хранения и др.
Replica Count	Для кластера типа Standard: Большое количество реплик повышает отказоустойчивость и безопасность данных, но снижает эффективность использования хранилища. Обычно достаточно значения 3. Для кластера типа Extended: Значение реплик по умолчанию — 4, изменить нельзя.
Device Class	Для кластера типа Standard: Выберите уже добавленный device class в рамках созданного storage pool. • При выборе device class данные будут храниться в выбранном классе устройств. • Если device class не выбран, данные будут случайным образом распределены по всем устройствам в storage pool. Для кластера типа Extended: Добавление device class не поддерживается.

Если выбран тип object storage, можно также настроить следующие параметры:

Параметр	Описание
Region	Укажите регион, в котором расположен storage pool.
Gateway Type	По умолчанию S3, изменить нельзя.
Internal Port	Укажите порт для внутреннего доступа к кластеру.
External Access	Включение/отключение внешнего доступа создаст/удалит Service типа NodePort.
Instance Count	Количество ресурсных экземпляров для object storage.

5. Нажмите **Create**.

Удаление Storage Pool

Если определённый тип хранилища больше не нужен, **storage pool** можно удалить после отвязки его от **storage class**:

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Storage Management > Distributed Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool**, нажмите на  рядом с нужным **storage pool** и выберите **Delete**.
4. Ознакомьтесь с предупреждением и введите имя **storage pool**.
5. Нажмите **Delete**.

Просмотр адресов Object Storage Pool

После создания **object storage pool** вы можете просмотреть адреса внутреннего и внешнего доступа к **storage pool**:

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Storage Management > Distributed Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool**, нажмите на  : рядом с object storage pool и выберите **View Address**.

Развёртывание компонентов на конкретных узлах

После создания распределённого хранилища вы можете просматривать и изменять место развёртывания компонентов, что облегчает расширение и обслуживание хранилища.

Обновление конфигурации развёртывания компонентов

Меры предосторожности

Обновление конфигурации вызовет автоматическую пересборку экземпляров компонентов системой, что может повлиять на доступ к сервисам хранилища. Рекомендуется выполнять обновление в непииковое время.

При типе кластера **Extend** функция фиксированного развёртывания компонентов не поддерживается.

Шаги

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
3. Во вкладке **Storage Components** нажмите **Component Deployment Configuration**.
4. Включите или отключите переключатель **Fixed Deployment** в соответствии с бизнес-требованиями и разверните компоненты на указанных узлах. Количество узлов должно быть не менее трёх для обеспечения минимальной доступности. Компоненты, для которых доступна настройка фиксированного развёртывания, включают MON, MGR, MDS, RGW.
5. Нажмите **Update**, после чего компоненты начнут планироваться на указанные узлы.

Перезапуск компонентов хранилища

При удалении развернутых компонентов хранилища система автоматически переназначит и повторно развернёт компоненты на узлах согласно текущей стратегии развёртывания компонентов.

Шаги

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
3. Во вкладке **Storage Components** нажмите  : рядом с именем компонента > **Delete**.

Добавление устройств/классов устройств

Добавление классов устройств

Объедините классификацию устройств одного типа или жестких дисков с одинаковой бизнес-логикой в узлах кластера, настройте классы устройств в соответствии с

требованиями к хранению и распределите различное содержимое хранения по разным типам дисков.

Примечания

Добавление классов устройств не поддерживается, если тип кластера — **Extend**.

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите Storage Management > Distributed Storage.
3. Нажмите на вкладку **Device Classes**.
4. Нажмите **Add Device Class** и настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Name	Имя класса устройства. Следующие имена нельзя использовать для класса устройства: hdd , ssd , nvme .
Storage Devices	Выберите Blank Disks или Unformatted Disk Partitions в узле. Если переключатель для всех пустых устройств включен: добавляются все пустые устройства узла в этот класс устройств; Если переключатель для всех пустых устройств выключен: вручную введите имена пустых устройств узла, например, sda .

Добавление устройств

Отобразите доступные жесткие диски как устройства хранения для использования и управления.

Внимание: После добавления жестких дисков в качестве устройств хранения обновление или удаление через интерфейс не поддерживается.

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите Storage Management > Distributed Storage.
3. Нажмите на вкладку **Device Classes**.
4. Справа от класса устройства нажмите **Add Device** и настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Node Type	Выберите тип узла, в котором находится жесткий диск, который вы хотите добавить в качестве устройства хранения. Compute Node: Узел, в котором не добавлены устройства хранения. Storage Node: Узел, в котором уже добавлены устройства хранения.
Add Type	Выберите способ добавления жестких дисков в качестве устройств хранения. All Empty Disks: Выбрать для добавления всех неразмеченных монтированных дисков узла в качестве устройств хранения. Specified Disks: Выбрать для добавления некоторых дисков узла в качестве устройств хранения, включая пустые диски или уже размеченные

Параметр	Описание
	монтированные диски. Если тип узла — Storage Node , можно выбрать только Specified Disks .
Specified Disks	Если выбран тип добавления Specified Disks , введите имена всех жестких дисков, которые нужно добавить в качестве устройств хранения, например, sda , sdb . После ввода каждого имени жесткого диска нажмите Enter для подтверждения. Внимание: Рекомендуется использовать весь жесткий диск в качестве устройства хранения, а не разделы на жестком диске.

5. Нажмите **Add**.

Мониторинг и оповещения

Распределённое хранилище предоставляет встроенные возможности по сбору метрик мониторинга и уведомлению об оповещениях. После включения функций мониторинга и оповещений вы можете отслеживать и получать оповещения по таким аспектам, как кластер хранения, производительность хранилища и компоненты хранилища, с поддержкой настройки стратегий уведомлений.

Интуитивно представленные данные мониторинга могут использоваться для поддержки принятия решений при проведении инспекций эксплуатации и обслуживания или оптимизации производительности, а комплексный механизм оповещений поможет обеспечить стабильную работу системы хранения.

Внимание: Если функции мониторинга и оповещений не были включены при создании распределённого хранилища, вам потребуется искать альтернативные решения для мониторинга и оповещений хранилища. Например, вручную настроить панели мониторинга и стратегии оповещений в центре эксплуатации и обслуживания.

Мониторинг

Платформа автоматически собирает распространённые метрики мониторинга для распределённого хранилища, такие как производительность чтения и записи, использование CPU и памяти. В разделе **Storage Management > Distributed Storage** на вкладке **Monitoring** вы можете просматривать данные мониторинга в реальном времени по этим метрикам.

Обзор хранилища

Отслеживайте состояние здоровья хранилища, использование физической ёмкости и количество активных компонентов OSD/MON. В случае аномального состояния хранилища вы можете проверить причину оповещения.

Мониторинг производительности

Отслеживайте пропускную способность чтения и записи, а также IOPS чтения и записи с трёх уровней: кластер, storage pool и OSD. Кроме того, можно мониторить задержки чтения и записи конкретно для OSD.

Мониторинг компонентов

Отслеживайте использование CPU и памяти таких компонентов, как MON и OSD.

Оповещения

В платформе включён набор стандартных стратегий оповещений. Как только ресурс становится аномальным или данные мониторинга достигают состояния предупреждения, оповещения автоматически срабатывают. Предусмотренные стратегии достаточно для типичных операционных задач, таких как оповещения о состоянии компонентов и кластера, оповещения о ёмкости устройств и оповещения по пользовательским данным.

Настройка уведомлений

Для своевременного получения оповещений рекомендуется настроить стратегии уведомлений в центре эксплуатации и обслуживания: отправлять информацию об оповещениях по электронной почте, SMS и другим каналам соответствующим сотрудникам, напоминая им принять необходимые меры для устранения проблем или предотвращения сбоев. Нажмите **Alert Configuration**, чтобы перейти в центр эксплуатации и обслуживания для завершения настройки.

Обработка оповещений

- Если кластер хранения находится в состоянии **Warning**, это означает, что сработало оповещение, и связанная аномалия может привести к сбою. Пожалуйста, оперативно проверьте детали в разделе **Real-time Alerts** и выявите и устраните неисправность на основе причины.
- Если кластер хранения находится в состоянии **Failure**, это указывает на то, что кластер хранения не может нормально функционировать. Необходимо немедленно локализовать проблему и провести устранение неисправности.

В таблице ниже приведены значения уровней оповещений, используемых в предустановленных стратегиях, которые могут служить вам ориентиром при формировании принципов обработки оповещений.

Уровень оповещения	Значение
Disaster	Ресурс, соответствующий правилу оповещения, вышел из строя, что вызвало прерывание работы платформы, потерю данных и значительное воздействие.
Severe	Ресурс, соответствующий правилу оповещения, имеет известные проблемы, которые могут привести к сбоям функций платформы и повлиять на нормальную работу сервисов.
Warning	Ресурс, соответствующий правилу оповещения, подвергается операционным рискам, которые могут повлиять на нормальную работу сервисов при отсутствии своевременных действий.

Анализ инцидентов

В разделе **Alert History** фиксируются все сработавшие оповещения, которые больше не требуют действий. При проведении анализа инцидентов с использованием истории оповещений для эффективного подведения итогов рекомендуется ответить на следующие вопросы.

- Каковы были конкретные аномальные условия в момент инцидента.
- Есть ли закономерность в повторяющемся оповещении, можно ли предотвратить его появление в будущем.
- Показывает ли временная шкала всплеск оповещений в определённый период; был ли он вызван форс-мажором или операционной ошибкой, требуется ли корректировка плана эксплуатации.

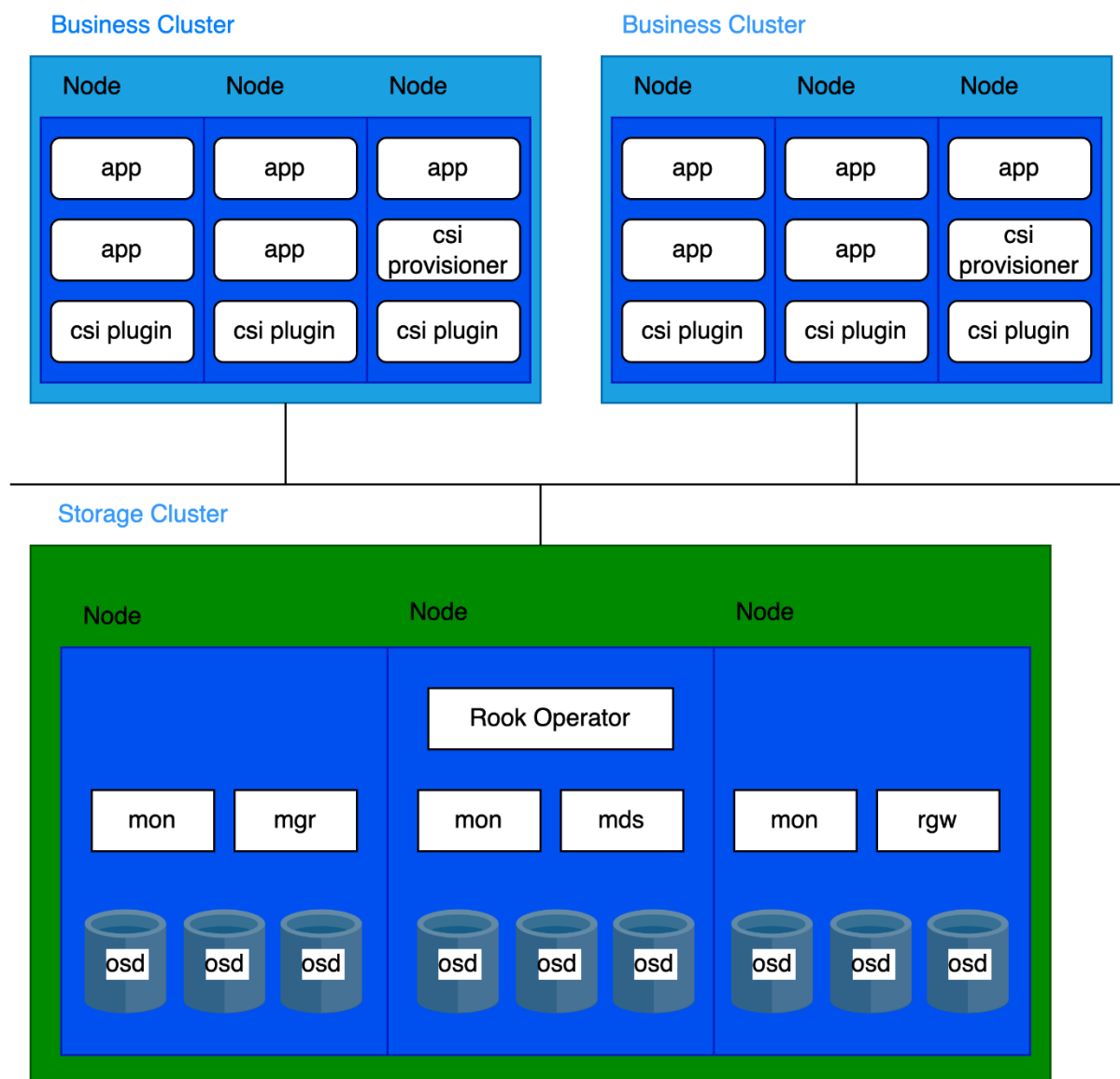
Настройка выделенного кластера для распределённого хранилища

Развёртывание выделенного кластера подразумевает использование отдельного кластера для развёртывания распределённого хранилища платформы, при этом другие бизнес-кластеры внутри платформы получают доступ и используют предоставляемые им сервисы хранения через интеграцию.

Для обеспечения производительности и стабильности распределённого хранилища платформы в выделенном кластере развёртываются только основные компоненты платформы и компоненты распределённого хранилища, избегая совместного размещения других бизнес-нагрузок. Такой отдельный подход к развёртыванию является рекомендуемой лучшей практикой для распределённого хранилища платформы.

Архитектура

Архитектура разделения хранения и вычислений:



Требования к инфраструктуре

Требования к платформе

Поддерживается в версии 3.18 и выше.

Требования к кластеру

Рекомендуется использовать **bare-metal** кластеры в качестве выделенных кластеров хранения.

Требования к ресурсам

Пожалуйста, ознакомьтесь с Основными понятиями для компонентов развёртывания распределённого хранилища.

Каждый компонент имеет свои требования к CPU и памяти. Рекомендуемые конфигурации следующие:

Процесс	CPU	Память

MON	2c	3Gi
MGR	3c	4Gi
MDS	3c	8Gi
RGW	2c	4Gi
OSD	4c	8Gi

В кластере обычно запускаются:

- 3 MON
- 2 MGR
- несколько OSD
- 2 MDS (если используется CephFS)
- 2 RGW (если используется CephObjectStorage)

Исходя из распределения компонентов, применимы следующие рекомендации по ресурсам на узел:

CPU	Память
16c + (4c * OSD на узел)	20Gi + (8Gi * OSD на узел)

Требования к устройствам хранения

Рекомендуется развёртывать не более 12 устройств хранения на узел. Это помогает ограничить время восстановления после сбоя узла.

Требования к типу устройств хранения

Рекомендуется использовать корпоративные SSD с ёмкостью 10TiB или меньше на устройство и обеспечить одинаковый размер и тип всех дисков.

Планирование ёмкости

Перед развёртыванием ёмкость хранения должна быть спланирована в соответствии с конкретными бизнес-требованиями. По умолчанию система распределённого хранения использует стратегию избыточности с 3 репликами. Следовательно, доступная ёмкость рассчитывается как общая сырая ёмкость хранения (со всех устройств) делённая на 3.

Пример для 30(N) узлов (число реплик = 3), сценарий доступной ёмкости:

Размер устройства хранения(D)	Устройств на узел(M)	Общая ёмкость(DMN)	Доступная ёмкость(DMN/3)
0.5 TiB	3	45 TiB	15 TiB
2 TiB	6	360 TiB	120 TiB
4 TiB	9	1080 TiB	360 TiB

Мониторинг ёмкости и расширение

1. Проактивное планирование ёмкости

Всегда следите, чтобы доступная ёмкость хранения превышала потребление. Если хранилище полностью исчерпано, восстановление требует ручного вмешательства и не может быть решено простым удалением или миграцией данных.

2. Оповещения о ёмкости

Кластер генерирует оповещения при двух порогах:

80% использования («почти заполнено»): проактивно освободите место или расширьте кластер.

95% использования («заполнено»): хранилище полностью исчерпано, стандартные команды не могут освободить место. Немедленно обратитесь в службу поддержки платформы.

Всегда оперативно реагируйте на оповещения и регулярно контролируйте использование хранилища, чтобы избежать простоев.

3. Рекомендации по масштабированию

Избегайте: добавления устройств хранения к существующим узлам.

Рекомендуется: масштабирование путём добавления новых узлов хранения.

Требование: новые узлы должны использовать устройства хранения идентичного размера, типа и количества с существующими узлами.

Требования к сети

Распределённое хранилище должно использовать **HostNetwork**.

Сетевая изоляция

Сеть делится на два типа:

Публичная сеть: используется для взаимодействия клиентов с компонентами хранилища (например, I/O запросы).

Кластерная сеть: выделена для репликации данных между репликами и балансировки данных (например, восстановление).

Для обеспечения качества сервиса и стабильности производительности:

1. Для выделенных кластеров хранения:

Зарезервируйте два сетевых интерфейса на каждом хосте:

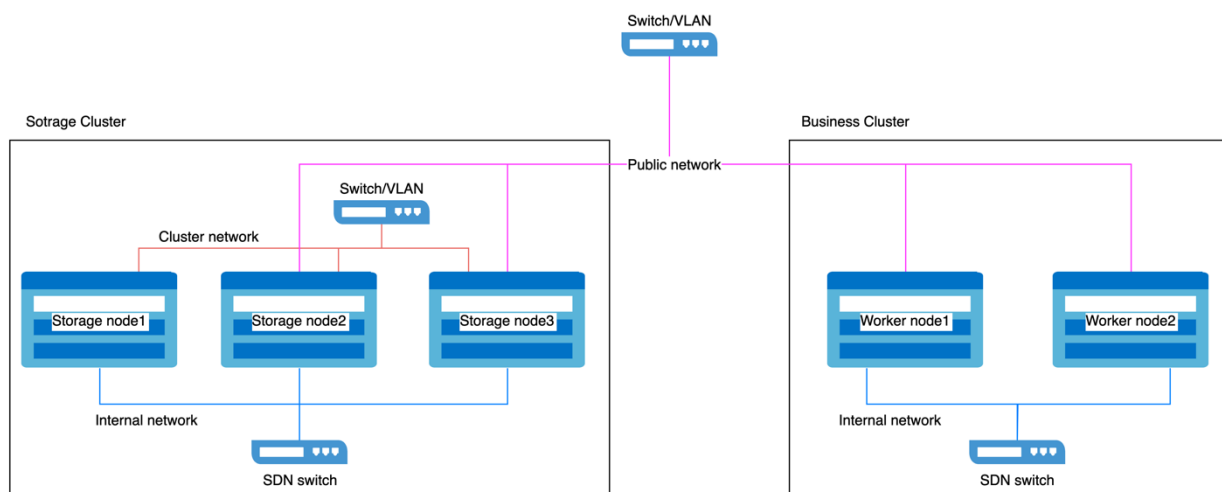
Публичная сеть: для связи клиентов и компонентов.

Кластерная сеть: для внутреннего трафика репликации и балансировки.

2. Для бизнес-кластеров:

Зарезервируйте один сетевой интерфейс на каждом хосте для доступа к публичной сети хранилища.

Пример конфигурации сетевой изоляции



Требования к скорости сетевых интерфейсов

1. Узлы хранения

Для **Публичной сети** и **Кластерной сети** требуются сетевые интерфейсы 10GbE или выше.

2. Узлы бизнес-кластеров

Сетевой интерфейс, используемый для доступа к публичной сети хранилища, должен быть 10GbE или выше.

Процедура

Развёртывание Operator

1. Перейдите в Управление платформой.
2. В левой боковой панели нажмите Управление хранилищем > Распределённое хранилище.
3. Нажмите Создать сейчас.
4. На странице мастера **Развёртывание Operator** нажмите кнопку **Deploy Operator** в правом нижнем углу.

Если страница автоматически перейдёт к следующему шагу, это означает успешное развёртывание **Operator**.

Если развёртывание не удалось, следуйте подсказке на интерфейсе **Очистить информацию о развёртывании и повторить попытку**, и повторно разверните Operator; если хотите вернуться на страницу выбора распределённого хранилища, нажмите **Application Store**, сначала удалите ресурсы уже развернутого rook-operator, затем удалите сам **rook-operator**.

Создание кластера ceph

Выполните команды на **контрольном узле** кластера хранения.

```
cat << EOF | kubectl create -f -
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata:
  name: ceph-cluster
```

```
namespace: rook-ceph
spec:
  cephConfig:
    global:
      mon_memory_target: "3221225472"
      mds_cache_memory_limit: "8589934592"
      osd_memory_target: "8589934592"
      bluefs_buffered_io: "false"
    mon:
      auth_allow_insecure_global_id_reclaim: "true"
      mon_warn_on_insecure_global_id_reclaim: "false"
      mon_warn_on_insecure_global_id_reclaim_allowed: "false"
  cephVersion:
    image: build-harbor.alauda.cn/3rdparty/ceph/ceph:v18.2.4-0
  dashboard:
    enabled: true
  dataDirHostPath: /var/lib/rook
  mgr:
    count: 2
    modules:
      - enabled: true
        name: pg_autoscaler
  mon:
    count: 3
  monitoring:
    enabled: true
  network:
    ipFamily: IPv4
    addressRanges:
      public:
        - <public network cidr>
      cluster:
        - <cluster network cidr>
    provider: host
  placement:
    all:
      tolerations:
        - effect: NoSchedule
          operator: Exists
        - key: "node-role.kubernetes.io/master"
```

- operator: "Exists"
- effect: "NoSchedule"
- key: "node-role.kubernetes.io/control-plane"
 - operator: "Exists"
 - effect: "NoSchedule"
- key: "node-role.kubernetes.io/cpaas-system"
 - operator: "Exists"
 - effect: "NoSchedule"

mgr:

podAffinity:

podAntiAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:
 - matchExpressions:
 - key: app
 - operator: In
 - values:
 - rook-ceph-mgr
 - topologyKey: kubernetes.io/hostname

priorityClassNames:

all: system-node-critical

resources:

crashcollector:

limits:

cpu: 200m

memory: 128Mi

requests:

cpu: 100m

memory: 64Mi

mgr:

requests:

cpu: "3"

memory: 4Gi

mon:

requests:

cpu: "2"

memory: 3Gi

osd:

requests:

cpu: "4"

```
memory: 8Gi
storage:
  <storage devices>
EOF
```

Параметры:

public network cidr: CIDR публичной сети хранилища (например, - **10.0.1.0/24**).

cluster network cidr: CIDR кластерной сети хранилища (например, - **10.0.2.0/24**).

storage devices: Укажите устройства хранения, которые будут использоваться распределённым хранилищем.

Пример форматирования:

```
nodes:
- name: storage-node-01
  devices:
  - name: /dev/disk/by-id/wwn-0x5000cca01dd27d60
  useAllDevices: false
- name: storage-node-02
  devices:
  - name: sdb
  - name: sdc
  useAllDevices: false
- name: storage-node-03
  devices:
  - name: sdb
  - name: sdc
  useAllDevices: false
```

Внимание: Используйте World Wide Name (WWN) диска для стабильного именования, что позволяет избежать зависимости от нестабильных путей устройств, таких как sdb, которые могут изменяться после перезагрузок.

Создание пулов хранения

Доступны три типа пулов хранения. Выберите и создайте подходящие в соответствии с вашими бизнес-требованиями.

Создание файлового пула

Выполните команды на контрольном узле кластера хранения.

```
cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephFilesystem
metadata:
  name: cephfs
  namespace: rook-ceph
```



```

spec:
  metadataPool:
    failureDomain: host
    replicated:
      requireSafeReplicaSize: true
      size: 3
  dataPools:
  - failureDomain: host
    replicated:
      requireSafeReplicaSize: true
      size: 3
  preserveFilesystemOnDelete: false
  metadataServer:
    activeCount: 1
    activeStandby: true
  placement:
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchExpressions:
              - key: app
                operator: In
                values:
                  - rook-ceph-mds
            topologyKey: kubernetes.io/hostname
  tolerations:
  - effect: NoSchedule
    operator: Exists
  resources:
    requests:
      cpu: "3"
      memory: 8Gi
EOF

```

Создание блочного пула

Выполните команды на контрольном узле кластера хранения.

```

cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephBlockPool
metadata:
  name: block

```

```
namespace: rook-ceph
spec:
  failureDomain: host
  replicated:
    size: 3
EOF
```

Создание объектного пула

Выполните команды на **контрольном узле** кластера хранения.

```
cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephObjectStore
metadata:
  name: object
  namespace: rook-ceph
spec:
  metadataPool:
    failureDomain: host
    replicated:
      requireSafeReplicaSize: true
      size: 3
  dataPool:
    failureDomain: host
    replicated:
      requireSafeReplicaSize: true
      size: 3
  preservePoolsOnDelete: false
  gateway:
    instances: 2
  placement:
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchExpressions:
              - key: app
                operator: In
                values:
                  - rook-ceph-rgw
            topologyKey: kubernetes.io/hostname
  tolerations:
    - effect: NoSchedule
```

```
operator: Exists
port: 7480
resources:
  requests:
    cpu: "2"
    memory: 4Gi
```

EOF

Очистка распределённого хранилища

Если необходимо удалить кластер **rook-ceph** и развернуть новый, следует последовательно очистить ресурсы, связанные с сервисом распределённого хранилища, согласно данной инструкции.

Удаление VolumeSnapshotClasses

1. Удалите VolumeSnapshotClasses
`kubectl delete VolumeSnapshotClass csi-cephfs-snapshotclass csi-rbd-snapshotclass`
2. Проверьте, что VolumeSnapshotClasses удалены.
`kubectl get VolumeSnapshotClass | grep csi-cephfs-snapshotclass`
`kubectl get VolumeSnapshotClass | grep csi-rbd-snapshotclass`

Удаление StorageClasses

1. Перейдите в Platform Management.
2. В левой навигационной панели выберите **Storage Management > Storage Classes**.
3. Нажмите **: > Delete** и удалите все **StorageClasses**, использующие решения хранилища **Ceph**.

Удаление Storage Pools

Этот шаг выполняется после завершения предыдущего.

1. Перейдите в Platform Management.
2. В левой навигационной панели выберите Storage Management > Distributed Storage.
3. В разделе **Storage Pool Area** нажмите **: > Delete** и удалите все пулы хранилища по одному. Когда в области пула хранилища отображается сообщение No Storage Pools, это означает успешное удаление.
4. (Опционально) Если режим кластера — **Extended**, выполните следующую команду на Master-узле кластера для удаления созданных встроенных пулов хранилища.
`kubectl -n rook-ceph delete cephblockpool -l cpaas.io/builtin=true`

Ответ:

```
cephblockpool.ceph.rook.io "builtin-mgr" deleted
```

Удаление ceph-cluster

Этот шаг выполняется после завершения предыдущего.

1. Обновите ceph-cluster, включив политику очистки.

```
kubectrl -n rook-ceph patch cephcluster ceph-cluster --type merge -p
'{"spec":{"cleanupPolicy":{"confirmation":"yes-really-destroy-data"}}}'
```

2. Удалите ceph-cluster.

```
kubectrl delete cephcluster ceph-cluster -n rook-ceph
```

3. Удалите задания, выполняющие очистку.

```
kubectrl delete jobs --all -n rook-ceph
```

4. Проверьте, что очистка ceph-cluster завершена.

```
kubectrl get cephcluster -n rook-ceph | grep ceph-cluster
```

Удаление rook-operator

Этот шаг выполняется после завершения предыдущего.

1. Удалите rook-operator.

```
kubectrl -n rook-ceph delete subscriptions.operators.coreos.com rook-operator
```

2. Проверьте, что очистка rook-operator завершена.

```
kubectrl get subscriptions.operators.coreos.com -n rook-ceph | grep rook-operator
```

Отсутствие вывода означает, что очистка завершена.

3. Проверьте, что все ConfigMaps удалены.

```
kubectrl get configmap -n rook-ceph
```

Отсутствие вывода означает, что очистка завершена. Если есть результаты, выполните следующую команду для очистки, заменив **<configmap>** на фактическое имя.

```
kubectrl -n rook-ceph patch configmap <configmap> --type merge -p
'{"metadata":{"finalizers": []}}'
```

4. Проверьте, что все Secrets удалены.

```
kubectrl get secret -n rook-ceph
```

Отсутствие вывода означает, что очистка завершена. Если есть результаты, выполните следующую команду для очистки, заменив **<secret>** на фактическое имя.

```
kubectrl -n rook-ceph patch secrets <secret> --type merge -p '{"metadata":{"finalizers":
[]}}'
```

5. Проверьте, что очистка **rook-ceph** завершена.

```
kubectrl get all -n rook-ceph
```

6. Отсутствие вывода означает, что очистка завершена.

Выполнение скрипта очистки

После выполнения вышеуказанных шагов **Kubernetes** и ресурсы **Ceph** будут очищены. Далее необходимо удалить остатки **rook-ceph** на хосте.

Скрипт очистки

Содержимое скрипта clean-rook.sh:

```
#!/bin/bash
```

```
DISK="$1"
```

```

if [ ! -n "$DISK" ]
then
    echo "you must input block dev"
    exit
else
    echo "are you sure to clean device: $DISK ? yes/no"
    read ANSWER
    case $ANSWER in
        [Yy]*)
            echo " you input is y or Y !"
            ;;
        [Nn]*)
            echo " you input a "$ANSWER
            exit
            ;;
    esac
fi

echo "clean /var/lib/rook"
rm -rf /var/lib/rook

echo "clean block dev"

# Recreate partition table and add partition using fdisk, do not delete the empty lines in
between
echo "g
n

w" | sudo fdisk $DISK

# Zap the disk to a fresh, usable state (zap-all is important, because MBR has to be clean)
# You will have to run this step for all disks.
sgdisk --zap-all $DISK

# Clean HDDs with dd
dd if=/dev/zero of="$DISK" bs=1M count=100 oflag=direct,dsync

```

```
# Clean disks such as SSDs with blkdiscard instead of dd
blkdiscard $DISK
```

```
# These steps only have to be run once on each node
```

```
# If rook sets up OSDs using ceph-volume, teardown leaves some devices mapped that
lock the disks.
```

```
ls /dev/mapper/ceph-* | xargs -l% -- dmsetup remove %
```

```
# ceph-volume setup can leave ceph-<UUID> directories in /dev (unnecessary clutter)
rm -rf /dev/ceph-*
```

Меры предосторожности

Скрипт очистки зависит от команды `sgdisk`, поэтому убедитесь, что она установлена перед запуском скрипта.

- Команда установки для Ubuntu: **`sudo apt install gdisk`**
- Команда установки для RedHat или CentOS: **`sudo yum install gdisk`**

Шаги

1. Запустите скрипт очистки **`clean-rook.sh`** на каждой машине в бизнес-кластере, где развернуто распределённое хранилище.
`sh clean-rook.sh /dev/[device_name]`

Пример:

```
sh clean-rook.sh /dev/vdb
```

При выполнении будет запрошено подтверждение очистки устройства. Для подтверждения введите **yes**.

2. Используйте команду **`lsblk -f`** для проверки информации о разделах. Если в столбце **FSTYPE** вывод отсутствует, очистка завершена.

Восстановление после сбоев блочного хранилища

RBD Mirror — это функция Ceph Block Storage (RBD), которая обеспечивает асинхронную репликацию данных между разными кластерами Ceph, предоставляя межкластерное восстановление после сбоев (Disaster Recovery, DR). Основная функция — синхронизация данных в режиме primary-backup, обеспечивающая быстрое переключение на резервный кластер при сбое основного.

Внимание:

- RBD Mirror выполняет инкрементальную синхронизацию на основе снимков (snapshots), с интервалом создания снимков по умолчанию раз в час (настраивается). Дифференциальные данные между основным и резервным кластерами обычно соответствуют записям за один цикл снимка.

- RBD Mirror обеспечивает только резервное копирование данных на уровне хранилища и не занимается резервным копированием ресурсов Kubernetes. Для резервного копирования PVC и PV используйте функцию **Backup and Restore** платформы.

Терминология

Термин	Объяснение
Primary Cluster	Кластер, который в данный момент предоставляет услуги хранения.
Secondary Cluster	Резервный кластер, используемый для целей бэкапа.

Настройка резервного копирования

Предварительные условия

- Подготовьте два кластера, способных развернуть **MWS-CP Storage с Ceph**: Primary и Secondary, с сетевым соединением между ними.
- Оба кластера должны работать на одной версии платформы (v3.12 или выше).
- Создайте распределённые сервисы хранения в обоих кластерах — Primary и Secondary.
- Создайте блочные пулы хранения с **одинаковыми именами** в обоих кластерах.
- Убедитесь, что следующие три образа загружены в приватный репозиторий образов платформы:

quay.io/csiaddons/k8s-controller:v0.5.0

quay.io/csiaddons/k8s-sidecar:v0.8.0

quay.io/brancz/kube-rbac-proxy:v0.8.0

Шаги

Шаг 1. Включение зеркалирования для блочного пула Primary кластера

Выполните следующую команду на управляющем узле Primary кластера:

Command:

```
kubectl -n rook-ceph patch cephblockpool <block-pool-name> \
--type merge -p '{"spec":{"mirroring":{"enabled":true,"mode":"image"}}}'
```

Output:

```
cephblockpool.ceph.rook.io/<block-pool-name> patched
```

Параметры:

<block-pool-name>: имя блочного пула хранения.

Шаг 2. Получение Peer Token

Этот токен является ключевым учётным данным для установления зеркальных соединений между кластерами.

Выполните следующую команду на управляющем узле Primary кластера:

Command:

```
kubectrl get secret -n rook-ceph \  
$(kubectrl get cephblockpool.ceph.rook.io <block-pool-name> -n rook-ceph -o  
jsonpath='{.status.info.rbdMirrorBootstrapPeerSecretName}') \  
-o jsonpath='{.data.token}' | base64 -d
```

Output:

```
# Вывод сокращён из-за чувствительной информации  
eyJmc2lkIjoiMjc2N2I3ZmEtY2YwYi00N...
```

Параметры:

<block-pool-name>: имя блочного пула хранения.

Шаг 3. Создание секрета Peer Token в Secondary кластере

Выполните следующую команду на управляющем узле Secondary кластера:

Command:

```
kubectrl -n rook-ceph create secret generic rbd-primary-site-secret \  
--from-literal=token=<token> \  
--from-literal=pool=<block-pool-name>
```

Output:

```
secret/rbd-primary-site-secret created
```

Параметры:

<token>: токен, полученный на Шаге 2.

<block-pool-name>: имя блочного пула хранения.

Шаг 4. Включение зеркалирования для блочного пула Secondary кластера

Выполните следующую команду на управляющем узле Secondary кластера:

Command:

```
kubectrl -n rook-ceph patch cephblockpool <block-pool-name> --type merge -p \  
{  
  "spec": {  
    "mirroring": {  
      "enabled": true,  
      "mode": "image",  
      "peers": {  
        "secretNames": [  
          "rbd-primary-site-secret"  
        ]  
      }  
    }  
  }  
}
```

Output:


```
cephblockpool.ceph.rook.io/<block-pool-name> patched
```

Параметры:

<block-pool-name>: имя блочного пула хранения.

Шаг 5. Развёртывание демона зеркалирования в Secondary кластере

Этот демон отвечает за мониторинг и управление процессами синхронизации RBD mirror, включая синхронизацию данных и обработку ошибок.

Выполните следующую команду на управляющем узле Secondary кластера:

Command:

```
cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephRBDMirror
metadata:
  name: rbd-mirror
  namespace: rook-ceph
spec:
  count: 1
EOF
```

Output:

```
cephrbdmirror.ceph.rook.io/rbd-mirror created
```

Шаг 6. Проверка статуса зеркалирования

Выполните следующую команду на управляющем узле Secondary кластера:

Command:

```
kubectl get cephblockpools.ceph.rook.io <block-pool-name> -n rook-ceph -o
jsonpath='{.status.mirroringStatus.summary}'
```

Output:

```
# Все статусы "OK" означают нормальную работу
{"daemon_health":"OK","health":"OK","image_health":"OK","states":{}}
```

Параметры:

<block-pool-name>: имя блочного пула хранения.

Шаг 7. Включение репликационного sidecar

Эта функция обеспечивает эффективную репликацию и синхронизацию данных без прерывания работы основного приложения, повышая надёжность и доступность системы.

1 Развёртывание csiaddons-controller

Выполните следующие команды на управляющих узлах обоих кластеров — Primary и Secondary:

```
kubectl create -f https://raw.githubusercontent.com/csi-addons/kubernetes-csi-
addons/v0.5.0/deploy/controller/crds.yaml
kubectl create -f https://raw.githubusercontent.com/csi-addons/kubernetes-csi-
addons/v0.5.0/deploy/controller/rbac.yaml
```

```

cat << EOF | kubectl apply -f -
apiVersion: v1
data:
  controller_manager_config.yaml: |
    apiVersion: controller-runtime.sigs.k8s.io/v1alpha1
    kind: ControllerManagerConfig
    health:
      healthProbeBindAddress: :8081
    metrics:
      bindAddress: 127.0.0.1:8080
    webhook:
      port: 9443
    leaderElection:
      leaderElect: true
      resourceName: e8cd140a.openshift.io
kind: ConfigMap
metadata:
  name: csi-addons-manager-config
  namespace: csi-addons-system
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/name: csi-addons
  name: csi-addons-controller-manager
  namespace: csi-addons-system
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: csi-addons
  template:
    metadata:
      annotations:
        kubectl.kubernetes.io/default-container: manager
      labels:
        app.kubernetes.io/name: csi-addons
    spec:

```

```

containers:
- args:
  - --secure-listen-address=0.0.0.0:8443
  - --upstream=http://127.0.0.1:8080/
  - --logtostderr=true
  - --v=10
  image: <registry>/brancz/kube-rbac-proxy:v0.8.0
  name: kube-rbac-proxy
  ports:
  - containerPort: 8443
    name: https
    protocol: TCP
  resources:
    limits:
      cpu: 500m
      memory: 128Mi
    requests:
      cpu: 10m
      memory: 64Mi
- args:
  - --health-probe-bind-address=:8081
  - --metrics-bind-address=127.0.0.1:8080
  - --leader-elect
  command:
  - /manager
  image: <registry>/csiaddons/k8s-controller:v0.5.0
  livenessProbe:
    httpGet:
      path: /healthz
      port: 8081
    initialDelaySeconds: 15
    periodSeconds: 20
  name: manager
  readinessProbe:
    httpGet:
      path: /readyz
      port: 8081
    initialDelaySeconds: 5
    periodSeconds: 10
  resources:

```

```

limits:
  cpu: 500m
  memory: 128Mi
requests:
  cpu: 10m
  memory: 64Mi
securityContext:
  allowPrivilegeEscalation: false
securityContext:
  runAsNonRoot: true
serviceAccountName: csi-addons-controller-manager
terminationGracePeriodSeconds: 10

```

EOF

Параметры:

<registry>: адрес реестра платформы.

2 Включение **csi sidecar**

Выполните следующие команды на управляющих узлах обоих кластеров — Primary и Secondary:

```

kubectl patch cm rook-ceph-operator-config -n rook-ceph --type json --patch \
'[
  {
    "op": "add",
    "path": "/data/CSI_ENABLE_OMAP_GENERATOR",
    "value": "true"
  },
  {
    "op": "add",
    "path": "/data/CSI_ENABLE_CSIADDONS",
    "value": "true"
  }
]'

```

Шаг 8. Создание **VolumeReplicationClass**

Выполните следующие команды на управляющих узлах обоих кластеров — Primary и Secondary:

Command:

```

cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplicationClass
metadata:

```

```

name: rbd-volumereplicationclass
spec:
  provisioner: rook-ceph.rbd.csi.ceph.com
  parameters:
    mirroringMode: snapshot
    schedulingInterval: "<scheduling-interval>" (1)
    replication.storage.openshift.io/replication-secret-name: rook-csi-rbd-provisioner
    replication.storage.openshift.io/replication-secret-namespace: rook-ceph
EOF

```

(1) **<scheduling-interval>**: Интервал планирования, (например, schedulingInterval: "1h" означает выполнение каждые 1 час).

Output:

```

volumereplicationclass.replication.storage.openshift.io/rbd-volumereplicationclass
created

```

Шаг 9. Включение зеркалирования для PVC

Выполните следующую команду на управляющем узле Primary кластера:

Command:

```

cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplication
metadata:
  name: <vr-name> (1)
  namespace: <namespace> (2)
spec:
  autoResync: false
  volumeReplicationClass: rbd-volumereplicationclass
  replicationState: primary
  dataSource:
    apiGroup: ""
    kind: PersistentVolumeClaim
    name: <pvc-name> (3)
EOF

```

(1) **<vr-name>**: имя объекта VolumeReplication, рекомендуется совпадать с именем PVC.

(2) **<namespace>**: namespace, к которому принадлежит VolumeReplication, должен совпадать с namespace PVC.

(3) **<pvc-name>**: имя PVC, для которого нужно включить зеркалирование.

Внимание: После включения RBD-образ в Secondary кластере становится доступен только для чтения.

Output:

```
volumereplication.replication.storage.openshift.io/<mirror-pvc-name> created
```

Переключение при сбое (Failover)

При сбое Primary кластера необходимо переключить отношения primary-backup для RBD-образа.

Предварительные условия

Ресурсы Kubernetes Primary кластера были забэкаплены и восстановлены в Secondary кластер, включая PVC, PV, рабочие нагрузки приложений и т.д.

Создание VolumeReplication

Выполните следующую команду на управляющем узле Secondary кластера:

```
cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplication
metadata:
  name: <vr-name>
  namespace: <namespace>
spec:
  autoResync: false
  dataSource:
    apiGroup: ""
    kind: PersistentVolumeClaim
    name: <mirror-pvc-name>
  replicationHandle: ""
  replicationState: primary
  volumeReplicationClass: rbd-volumereplicationclass
EOF
```

(1) **<vr-name>**: имя VolumeReplication.

(2) **<namespace>**: namespace PVC.

(3) **<mirror-pvc-name>**: имя PVC.

Внимание: После создания RBD-образ в Secondary кластере становится primary и доступен для записи.

Объектное хранилище с MinIO

MWS Container Platform (MWS-CP) Object Storage с MinIO — это сервис объектного хранения, лицензированный под Apache License v2.0. Он совместим с интерфейсом облачного хранилища Amazon S3, что делает его особенно подходящим для хранения больших объемов неструктурированных данных, таких как изображения, видео, файлы журналов, резервные копии и образы контейнеров/виртуальных машин. Размер объекта может варьироваться от нескольких КБ до максимума в 5 ТБ.

Основные преимущества

Простота: Минимализм — главный принцип проектирования MinIO, обеспечивающий функциональность «из коробки». Простота снижает вероятность ошибок, увеличивает время безотказной работы и повышает надежность, а также улучшает производительность.

Высокая производительность: MinIO является мировым лидером в области объектного хранения. На стандартном оборудовании скорости чтения/записи могут достигать до 183 ГБ/с и 171 ГБ/с соответственно.

Масштабируемость: Можно создавать несколько небольших и средних кластеров, легко управляемых, с поддержкой объединения нескольких кластеров в сверхбольшой пул ресурсов между дата-центрами, вместо прямого использования крупномасштабного централизованно управляемого распределенного кластера.

Облачно-нативность: Соответствует всем нативным архитектурам и процессам построения облачных вычислений, включает новейшие технологии и концепции облачных вычислений, что делает объектное хранение более удобным для Kubernetes

Установка

Прerequisites

MinIO строится на базовом хранилище, поэтому убедитесь, что в текущем кластере создан класс хранилища. Рекомендуется использовать TopoLVM.

Развёртывание оператора

1. В левой навигационной панели нажмите **Storage > Object Storage**.
2. Нажмите **Configure Now**.
3. На странице мастера **Deploy MinIO Operator** нажмите в правом нижнем углу **Deploy Operator**.
 - Если страница автоматически перейдет к следующему шагу, это означает успешное развёртывание оператора.
 - Если развёртывание не удалось, следуйте подсказкам интерфейса для **Clean Up Deployed Information and Retry** и повторно разверните оператора.

Создание кластера

1. На странице мастера **Create Cluster** настройте базовую информацию.

Параметр	Описание
Access Key	Идентификатор ключа доступа. Уникальный идентификатор, связанный с приватным ключом доступа; используется вместе с Access Key ID для шифрования и подписи запросов.
Secret Key	Приватный ключ доступа, используемый совместно с Access Key ID для шифрования и подписи запросов, идентификации отправителя и предотвращения подделки запросов.

- В разделе **Resource Configuration** настройте характеристики согласно следующим инструкциям.

Параметр	Описание
Small scale	Подходит для обработки до 100 000 объектов, поддерживает не более 50 одновременных подключений в тестовых средах или сценариях резервного копирования. Запрос и лимит ресурсов CPU по умолчанию установлены на 2 ядра, запрос и лимит памяти — 4 Gi.
Medium scale	Предназначен для корпоративных приложений, требующих хранения 1 000 000 объектов и способных обрабатывать до 200 одновременных запросов. Запрос и лимит CPU по умолчанию — 4 ядра, запрос и лимит памяти — 8 Gi.
Large scale	Предназначен для групп пользователей с потребностями хранения 10 000 000 объектов и обработки до 500 одновременных запросов, подходит для сценариев с высокой нагрузкой. Запрос и лимит CPU по умолчанию — 8 ядер, запрос и лимит памяти — 16 Gi.
Custom	Предлагает гибкие параметры настройки для профессиональных пользователей с особыми требованиями, обеспечивая точное соответствие масштаба сервиса и требований к производительности. Примечание: при настройке пользовательских характеристик необходимо убедиться, что:

- Запрос CPU больше 100 м.
 - Запрос памяти не менее 2 Gi.
 - Лимиты CPU и памяти не меньше запросов. |
- В разделе **Storage Pool** настройте соответствующую информацию согласно следующим инструкциям.

Параметр	Описание
Instance Number	Увеличение количества экземпляров в кластере MinIO значительно повышает производительность и надежность системы, обеспечивая высокую доступность данных. Однако слишком большое количество экземпляров может привести к следующим проблемам:

- Повышенное потребление ресурсов.
- Если на одном узле размещено несколько экземпляров, сбой узла может привести к одновременному отключению нескольких экземпляров, что снижает общую надежность кластера.

Создание бакета

Войдите на управляющий узел кластера и используйте команду для создания бакета.

Шаги

1. На странице сведений о кластере перейдите на вкладку **Access Method**, чтобы просмотреть адрес доступа MinIO, или выполните следующую команду для запроса.
`kubectl get svc -n <tenant ns> minio | grep -w minio | awk '{print $3}'`

Примечание: Замените tenant ns на фактическое пространство имён minio-system.

Пример:

- ```
kubectl get svc -n minio-system minio | grep -w minio | awk '{print $3}'
```
2. Получите команду mc.  
`wget https://dl.min.io/client/mc/release/linux-amd64/mc -O /bin/mc && chmod a+x /bin/mc`
  3. Настройте псевдоним кластера MinIO.
    - IPv4:  
`mc --insecure alias set <minio cluster alias> http://<minio endpoint>:<port> <accessKey> <secretKey>`
    - IPv6:  
`mc --insecure alias set <minio cluster alias> http://[<minio endpoint>]:<port> <accessKey> <secretKey>`
    - Доменное имя:  
`mc --insecure alias set <minio cluster alias> http://<domain name>:<port> <accessKey> <secretKey>`  
`mc --insecure alias set <minio cluster alias> https://<domain name>:<port> <accessKey> <secretKey>`

## Примечание:

- Введите IP-адрес, полученный на шаге 1, для **minio endpoint**.
- Введите **Access Key** и **Secret Key**, созданные при создании кластера, для **accessKey** и **secretKey**.

## Примеры настройки:

- IPv4:  
`mc --insecure alias set myminio http://12.4.121.250:80 07Apples@ 07Apples@`
  - IPv6:  
`mc --insecure alias set myminio http://[2004::192:168:143:117]:80 07Apples@ 07Apples@`
  - Доменное имя:  
`mc --insecure alias set myminio http://test.minio.alauda:80 07Apples@ 07Apples@` или  
`mc --insecure alias set myminio https://test.minio.alauda:443 07Apples@ 07Apples@`
4. Создайте бакет.  
`mc --insecure mb <minio cluster alias>/<bucket name>`

## Загрузка/скачивание файлов

После создания бакета вы можете использовать командную строку для загрузки файлов в бакет или скачивания существующих файлов из бакета.

### Шаги

1. Создайте файл для тестирования загрузки. Этот шаг можно пропустить, если загружаете уже существующий файл.  
`touch <file name>`
2. Загрузите файлы в бакет.  
`mc --insecure cp <file name> <minio cluster alias>/<bucket name>`
3. Просмотрите файлы в бакете, чтобы подтвердить успешную загрузку.  
`mc --insecure ls <minio cluster alias>/<bucket name>`
4. Удалите загруженные файлы.  
`mc --insecure rm <minio cluster alias>/<bucket name>/<file name>`

### Связанная информация

Таблица соответствия коэффициента избыточности

**Внимание:** При добавлении дополнительных пулов хранения коэффициент избыточности необходимо рассчитывать на основе количества экземпляров в первом пуле хранения.

| Количество экземпляров | Коэффициент избыточности |
|------------------------|--------------------------|
| 4 - 5                  | 2                        |
| 6 - 7                  | 3                        |
| $\geq 8$               | 4                        |

### Обзор пула хранения

| Параметр обзора пула хранения                                                                                                                                                                                                                                              | Формула расчёта                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Доступная ёмкость                                                                                                                                                                                                                                                          | Если Instance Number $\leq 16$ , Доступная ёмкость = Вместимость одного тома $\times$ (Количество экземпляров - Коэффициент избыточности). |
| Если количество экземпляров $> 16$ , Доступная ёмкость = Вместимость одного тома $\times$ (Количество экземпляров - $4 \times (\text{Количество экземпляров} + 15) / 16$ ). Результат выражения " $4 \times (\text{Количество экземпляров} + 15) / 16$ " округляется вниз. |                                                                                                                                            |
| Общая ёмкость                                                                                                                                                                                                                                                              | Общая ёмкость = Количество экземпляров $\times$ Вместимость одного тома                                                                    |

| Параметр обзора пула хранения                                                                                            | Формула расчёта                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Количество отказоустойчивых сервисов хранения                                                                            | Если Instance Number > 2 × Коэффициент избыточности, Количество отказоустойчивых сервисов = Коэффициент избыточности. |
| Если Instance Number = 2 × Коэффициент избыточности, количество отказоустойчивых сервисов = Коэффициент избыточности - 1 |                                                                                                                       |

## Архитектура

**MWS Container Platform (MWS-CP) Object Storage с MinIO** — это высокопроизводительная распределённая система объектного хранения, разработанная для облачных нативных сред. Она использует кодирование с удалением, распределённые пулы хранения и механизмы высокой доступности для обеспечения надёжности данных и масштабируемости в Kubernetes.

Основные компоненты:

- **MinIO Operator:** Управляет развертыванием и обновлением кластеров MinIO.
- **MinIO Peer:** Настраивает и управляет функциональностью репликации сайта MinIO.
- **MinIO Pool:** Основной компонент MinIO, отвечающий за обработку запросов объектного хранения. Каждый пул соответствует StatefulSet и предоставляет ресурсы хранения.

## Архитектура развёртывания

Для развёртывания MinIO в Kubernetes необходимо определить MinIO tenant, указав количество серверных экземпляров (pod) и количество томов (дисков) на каждый экземпляр. Каждый сервер MinIO управляется через StatefulSet, что обеспечивает стабильные идентификаторы и постоянное хранилище. MinIO объединяет все диски в один или несколько erasure set и применяет кодирование с удалением для обеспечения отказоустойчивости.

## Масштабирование с несколькими пулами

Кластеры MinIO могут масштабироваться за счёт добавления дополнительных серверных пулов, каждый из которых имеет свой собственный erasure set. Хотя это увеличивает ёмкость хранения, это усложняет обслуживание кластера и снижает общую надёжность. Сбой в любом серверном пуле может сделать весь кластер MinIO недоступным, даже если другие пулы продолжают работать.

## Заключение

MinIO — это высокомасштабируемое облачное объектное хранилище, которое балансирует производительность и надёжность. При проектировании кластера MinIO важно тщательно продумывать пулы хранения, настраивать параметры кодирования с удалением и

реализовывать стратегии высокой доступности для обеспечения целостности данных и стабильности работы в Kubernetes

### Основные концепции

- **Erase Coding (EC):** MinIO использует кодирование с удалением (Reed-Solomon erasure coding) для разбиения объектов на фрагменты данных и контрольные фрагменты (parity shards), распределяя их по нескольким дискам для обеспечения отказоустойчивости. Например, в конфигурации с 16 дисками данные могут быть разделены на 12 фрагментов данных и 4 контрольных фрагмента, что позволяет системе восстанавливать данные даже при отказе до 4 дисков.
- **Server Pools & Erasure Sets: Server Pools MinIO** — это логические объединения ресурсов хранения, где каждый пул состоит из нескольких узлов, совместно использующих возможности хранения и вычислений. Внутри пула диски автоматически организуются в один или несколько **Erasure Sets**.
  - **Распределение данных:** При сохранении объекта он разбивается на фрагменты данных и контрольные фрагменты, которые распределяются по разным дискам внутри erasure set.
  - **Модель избыточности:** Erasure sets формируют базовую единицу избыточности данных, обеспечивая устойчивость на основе настроенного соотношения фрагментов данных и контрольных фрагментов.
  - **Масштабируемость:** Один пул хранения MinIO может содержать несколько erasure sets, и новые данные всегда записываются в erasure set с наибольшей доступной емкостью.

### Добавление пула хранения

Пул хранения — это логический раздел, используемый для хранения данных. В одном и том же кластере хранения могут одновременно использоваться различные типы базового хранилища для удовлетворения различных бизнес-требований.

Помимо пулов хранения, созданных при настройке объектного хранилища, вы также можете добавить дополнительные пулы хранения.

**Примечание:** Добавление пула хранения вызовет кратковременное прерывание работы сервиса MinIO, но после этого он автоматически восстановится в нормальное состояние.

### Шаги

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Storage Management > Object Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до раздела **Storage Pool** и нажмите **Add Storage Pool**.
4. Настройте соответствующие параметры согласно приведённым ниже инструкциям.

| Параметр              | Описание                                                                                                                                                                                                                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Underlying Storage    | Базовое хранилище, используемое кластером MinIO. Пожалуйста, выберите существующий класс хранения, созданный в текущем кластере, рекомендуется использовать TopoLVM.                                                                                                                                                          |
| Storage Nodes         | Выберите узлы хранения, необходимые для кластера MinIO. Рекомендуется использовать от 4 до 16 узлов хранения; платформа развернёт по 1 сервису хранения для каждого выбранного узла.<br><b>Внимание:</b> при использовании 3 узлов хранения для обеспечения надёжности будет развернуто по 2 сервиса хранения на каждый узел. |
| Single Storage Volume | Вместимость одного тома хранения PVC. Каждый сервис хранения управляет 1 томом хранения, и после ввода вместимости одного тома платформа автоматически рассчитает ёмкость пула хранения и другую информацию, которую можно просмотреть в разделе <b>Storage Pool Overview</b> .                                               |

5. Нажмите **Confirm**.

## Мониторинг и оповещения

Система объектного хранения оснащена встроенными возможностями мониторинга и оповещений, охватывающими кластеры хранения, состояние сервисов и использование ресурсов. Также поддерживаются настраиваемые политики уведомлений для информирования вашей операционной команды. Информация мониторинга в реальном времени помогает оптимизировать производительность и принимать операционные решения, а автоматические оповещения обеспечивают стабильность и надежность вашей системы хранения.

### Мониторинг

По умолчанию платформа собирает ключевые метрики по кластерам хранения и состоянию сервисов. Вы можете получить доступ к данным мониторинга в реальном времени в разделе **Storage Management > Object Storage > Monitoring**.

### Обзор хранилища

Этот раздел предоставляет общий обзор состояния системы хранения, статуса сервисов и использования сырой емкости. Если состояние хранилища ненормальное, детали оповещения укажут на первопричину, что поможет эффективно диагностировать и устранять проблемы.

### Мониторинг кластера

Отслеживайте использование емкости и производительности ввода-вывода по всему кластеру хранения. Это помогает выявлять узкие места, оптимизировать распределение ресурсов и обеспечивать бесперебойную работу с данными.

## Мониторинг объектов

Контролируйте паттерны доступа, включая общее количество запросов и количество неудачных запросов. Эти данные помогают анализировать нагрузку на хранилище и выявлять аномалии, которые могут свидетельствовать о сбоях сервисов или угрозах безопасности.

## Оповещения

Платформа поставляется с преднастроенными политиками оповещений для обнаружения аномалий и отправки уведомлений при достижении заданных порогов. Встроенные правила охватывают ключевые области, такие как состояние компонентов, использование емкости и целостность пользовательских данных.

## Настройка уведомлений

Для обеспечения своевременного реагирования настройте политики уведомлений в **Operations Center**. Оповещения могут отправляться по электронной почте, SMS или другим каналам, чтобы уведомлять ответственных сотрудников. Тонко настройте параметры в соответствии с рабочим процессом реагирования вашей организации.

## Обработка оповещений

- **Кластер в состоянии "Alert"**: Сработало предупреждение, и стабильность системы может быть под угрозой. Проверьте раздел **Live Alerts** для получения подробностей, определите первопричину и примите корректирующие меры.
- **Кластер в состоянии "Failure"**: Кластер хранения перестал работать. Требуется немедленное вмешательство для восстановления доступности сервиса.

Платформа классифицирует оповещения по уровням серьезности, что помогает командам приоритизировать реагирование на инциденты:

| Уровень серьезности | Описание                                                                                                  |
|---------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Critical</b>     | Сбой системы, влияющий на бизнес-процессы или приводящий к потере данных. Требуется немедленное действие. |
| <b>Major</b>        | Известная проблема, которая может привести к сбоям в функциональности и нарушению бизнес-процессов.       |
| <b>Warning</b>      | Потенциальный риск, который при отсутствии реакции может повлиять на производительность или доступность.  |

## Анализ после инцидента

Журнал **Alert History** содержит все прошлые инциденты, предоставляя ценные данные для анализа и улучшения системы. При рассмотрении прошлых оповещений учитывайте следующее:

1. Каковы были точные симптомы во время инцидента?

2. Повторяются ли определённые оповещения со временем? Можно ли принять превентивные меры для предотвращения повторения?
3. Был ли определённый временной интервал с резким увеличением оповещений? Был ли он вызван операционной проблемой или внешним фактором? Следует ли скорректировать стратегию реагирования?

Постоянно анализируя паттерны оповещений и совершенствуя стратегии мониторинга, команды могут повысить устойчивость системы, минимизировать время простоя и обеспечить бесперебойную работу хранилища.

## Локальное хранилище TopoLVM

### Введение

TopoLVM — это плагин Container Storage Interface (CSI), разработанный специально для Kubernetes, предназначенный для эффективного и удобного управления локальными томами хранения.

### Основные особенности и преимущества:

- **Управление локальными томами:** TopoLVM ориентирован на управление локальными устройствами хранения (такими как диски и SSD) на узлах Kubernetes. По сравнению с традиционным сетевым хранилищем, локальные тома обеспечивают меньшую задержку и более высокую производительность.
- **Осведомлённость о топологии:** TopoLVM способен распознавать топологию кластера Kubernetes (например, узлы, зоны доступности), что позволяет автоматически выделять тома хранения на том же узле, где запланированы Pods, дополнительно оптимизируя производительность.
- **Динамическое выделение томов:** TopoLVM поддерживает динамическое создание, удаление и изменение размера томов хранения без ручного вмешательства, значительно упрощая операции и снижая сложность.
- **Глубокая интеграция с Kubernetes:** В качестве плагина CSI TopoLVM бесшовно интегрируется с API управления хранилищем Kubernetes, позволяя пользователям управлять локальными томами напрямую через стандартные ресурсы Kubernetes, такие как PersistentVolumeClaims.

Таким образом, TopoLVM решает распространённые проблемы, связанные с использованием локального хранилища в Kubernetes, такие как ручное управление, отсутствие учёта топологии и недостаточные возможности динамического выделения. Он предоставляет более эффективное и удобное решение для приложений, требующих высокопроизводительного локального хранилища, таких как базы данных и кэши.

### Установка

**Local storage** — это программно-определяемое локальное хранилище на сервере, обеспечивающее простую, удобную в обслуживании и высокопроизводительную возможность локального хранения данных. Основано на решении TopoLVM, реализует

оркестрацию управления постоянными томами локального хранилища через системный подход LVM.

### Прerequisites

Пакет `lvm2` должен быть установлен на каждом узле кластера хранения. Если он не установлен, выполните команду `yum install -y lvm2` на узле.

### Шаги

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management > Local Storage**.
3. Нажмите **Configure Now**.
4. На странице мастера **Install Operator** нажмите **Start Deployment**.
  - Если страница автоматически переходит к следующему шагу, это означает, что развертывание Operator прошло успешно.
  - Если развертывание не удалось, следуйте подсказкам интерфейса для устранения проблемы. Затем нажмите **Clean Up** и повторно разверните Operator.
5. На странице мастера **Create Cluster** добавьте устройства.

| Параметр       | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Select Node    | Узел с минимум одним неразмеченным диском.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Device Class   | Каждый класс устройств соответствует набору устройств хранения с одинаковыми характеристиками. Рекомендуется указывать имя, исходя из типа диска, например <i>hdd</i> , <i>ssd</i> .                                                                                                                                                                                                                                                                                        |
| Device Type    | Поддерживаются только типы дисков.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Storage Device | Например, <i>/dev/sda</i> . Если дисков несколько, их можно добавлять по одному.                                                                                                                                                                                                                                                                                                                                                                                            |
| Snapshot       | При включении поддерживается создание снимков PVC и использование этих снимков для настройки новых PVC, что обеспечивает быструю резервную копию и восстановление бизнес-данных.<br>Если при создании хранилища снимки не были включены, их можно активировать при необходимости в разделе <b>Operations</b> на странице деталей кластера хранения.<br><b>Внимание:</b> Перед использованием убедитесь, что для текущего кластера развернут <b>Volume Snapshot Plugin</b> . |

- Нажмите **Далее**. Если страница автоматически переходит к следующему шагу, это означает успешное развертывание кластера.



- Если создание не удалось, следуйте подсказкам интерфейса и своевременно очистите ресурсы.
6. На странице мастера Create Storage Class настройте соответствующие параметры.

| Параметр           | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name               | Имя класса хранения. Должно быть уникальным в рамках текущего кластера.                                                                                                                                                                                                                                                                                                                                                                                    |
| Display Name       | Имя, помогающее идентифицировать или фильтровать, например, описание класса хранения на русском языке.                                                                                                                                                                                                                                                                                                                                                     |
| Device Class       | Класс устройств — способ категоризации устройств хранения в TopoLVM. Каждый класс устройств соответствует набору устройств с одинаковыми характеристиками. Если нет особых требований, можно использовать класс устройств <b>Auto-Allocated</b> из кластера.                                                                                                                                                                                               |
| File System        | <p>- <b>XFS</b> — высокопроизводительная журналируемая файловая система, хорошо справляющаяся с параллельными I/O нагрузками, поддерживающая обработку больших файлов и обеспечивающая плавную передачу данных.</p> <p>- <b>EXT4</b> — журналируемая файловая система в Linux, использующая методы хранения с экстендами и поддерживающая обработку больших файлов. Максимальная емкость файловой системы — 1 EiB, максимальный размер файла — 16 TiB.</p> |
| Recycling Policy   | <p>Политика переработки для постоянных томов.</p> <p>- <b>Delete</b>: при удалении persistent volume claim связанный persistent volume также удаляется.</p> <p>- <b>Retain</b>: даже при удалении persistent volume claim связанный persistent volume сохраняется.</p>                                                                                                                                                                                     |
| Access Mode        | ReadWriteOnce (RWO): может быть смонтирован одним узлом в режиме чтения и записи.                                                                                                                                                                                                                                                                                                                                                                          |
| Allocation Project | <p>Этот тип <b>persistent volume claim</b> может быть создан только в определенных проектах.</p> <p>Если проект не назначен временно, его можно <b>Обновить</b> позже.</p>                                                                                                                                                                                                                                                                                 |

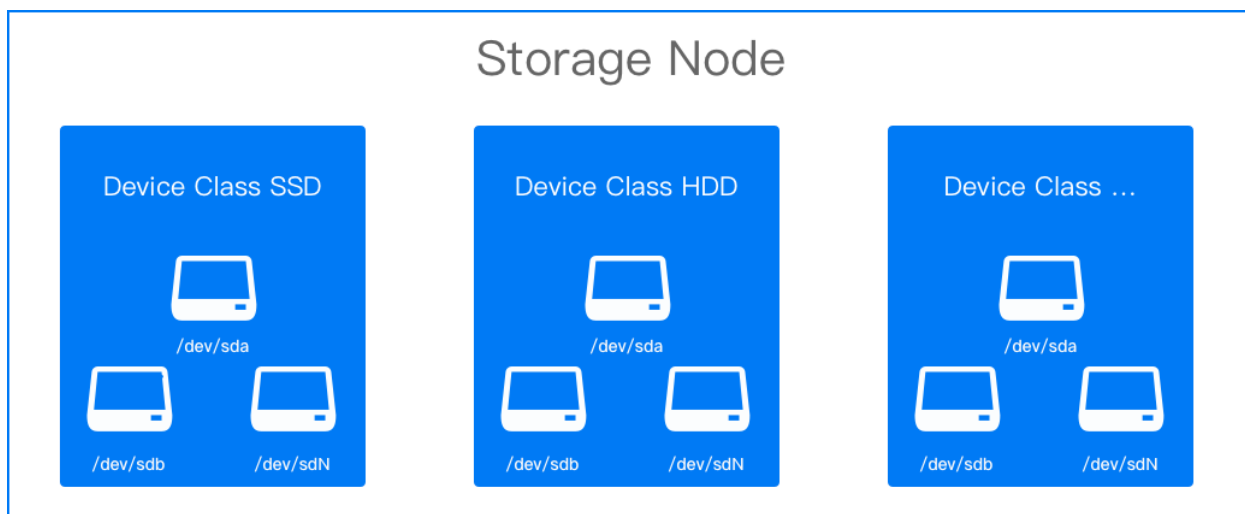
7. Нажмите **Next** и дождитесь завершения создания ресурсов.

### Управление устройствами

Будь то для первоначального развертывания или расширения ресурсов, необходимо сопоставить доступные диски на узле с устройствами хранения для использования и управления.

Устройства хранения с похожими характеристиками обычно используются централизованно, и такие устройства классифицируются в локальном хранилище под

**Device Classes.** Использование **device classes** эквивалентно прямому использованию дисков, обеспечивая нулевые потери и высокую производительность, а также снижая осведомленность приложений и зависимость от конкретных устройств.



### Прerequisites

- При создании кластера локального хранилища должен быть добавлен как минимум 1 Device Class (deviceClasses.classes), включая устройства в этом **device class**.
- На узле должен присутствовать как минимум 1 чистый диск.

### Добавление устройств

1. Перейдите в Platform Management.
2. В левой навигационной панели выберите **Storage Management > Local Storage**.
3. На вкладке Details нажмите Add Storage Node.
4. Настройте соответствующие параметры согласно приведённым ниже инструкциям.

| Параметр       | Описание                                                                                                                                                                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Storage Node   | Узел, на котором имеется как минимум 1 чистый диск.                                                                                                                                                        |
| Device Class   | Каждый device class соответствует группе устройств хранения с одинаковыми характеристиками; рекомендуется называть его в соответствии с типом дисков, например, hdd, ssd.                                  |
| Storage Device | Например, /dev/sda. Если дисков несколько, их можно добавлять по одному.<br><b>Внимание:</b> Устройство хранения должно быть целым жёстким диском, а не разделом на диске, так как это приведёт к ошибкам. |

5. Нажмите **Add**.

**Внимание:** Если статус класса устройств **Unavailable** из-за отсутствия добавленных устройств, можно продолжить выполнение следующих операций.

6. Перейдите на вкладку **Storage Devices** и нажмите **Add Storage Device**.

7. Добавьте устройства согласно подсказкам интерфейса.
8. Нажмите **Add**.

## **Мониторинг и оповещения**

Локальное хранилище предоставляет готовые возможности по сбору метрик мониторинга и оповещений. После включения компонента мониторинга платформы можно настроить мониторинг и оповещения на основе кластеров хранения, производительности и ёмкости хранилища с поддержкой настройки политик уведомлений.

Интуитивно представленные данные мониторинга могут использоваться для поддержки принятия решений при операционных проверках или настройке производительности, а комплексный механизм оповещений поможет обеспечить стабильную работу системы хранения.

### **Мониторинг**

#### **Мониторинг производительности**

По умолчанию платформа собирает часто используемые метрики мониторинга производительности, такие как пропускная способность чтения и записи, IOPS и задержки для локального хранилища. Данные мониторинга в реальном времени по этим метрикам можно просмотреть на вкладке **Monitoring** страницы **Local Storage** в разделе **Storage Management**. Платформа визуально отображает эти метрики с помощью графиков и диаграмм, что позволяет администраторам чётко наблюдать текущую производительность хранилища и быстро выявлять потенциальные проблемы.

#### **Мониторинг ёмкости**

Поскольку локальное хранилище может использовать только локально доступные ресурсы хранения на узлах, пользователи должны убедиться в наличии достаточного объёма свободной ёмкости на узлах перед объявлением локального хранилища, чтобы избежать проблем, вызванных избыточным объявлением.

Для помощи в этом платформа предоставляет подробный мониторинг ёмкости в разделе **Details** локального хранилища, классифицированный по типам устройств. Пользователи могут проверить доступное пространство хранения, отображаемое в числовом и графическом форматах. Если для какого-либо типа устройства доступная ёмкость недостаточна, необходимо освободить место или добавить дополнительные дисковые устройства перед использованием локального хранилища.

### **Оповещения**

Платформа включает набор стандартных политик оповещений. Если ресурсы становятся аномальными или данные мониторинга достигают порога предупреждения, оповещения автоматически срабатывают. Преднастроенные политики оповещений эффективно покрывают типичные операционные потребности, включая оповещения о состоянии здоровья кластера и ёмкости по типам устройств.

#### **Настройка уведомлений**

Для своевременного получения оповещений необходимо настроить политики уведомлений в центре операций. Уведомления могут отправляться по электронной почте, SMS или другими способами соответствующим сотрудникам, что обеспечивает оперативное реагирование для устранения проблем или предотвращения сбоев. Пользователи могут получить доступ к настройкам политик уведомлений непосредственно из интерфейса центра операций.

### Обработка оповещений

- Если состояние здоровья кластера хранения изменяется на **Alert**, администраторы должны немедленно провести расследование. Раздел **Details** предоставляет информацию для диагностики и решения этих проблем. Распространённые причины включают аномалии в службах узлов или проблемы с конкретными типами устройств.

| Пункт проверки            | Соответствующее состояние | Причина                                                                                        |
|---------------------------|---------------------------|------------------------------------------------------------------------------------------------|
| Состояние здоровья        | Alert                     | Вызвано аномалиями в службах узлов или проблемами с типом устройства.                          |
| Состояние службы          | Unknown                   | Узел находится в состоянии <b>notready</b> , возможно из-за сбоев сети или отключения питания. |
| Состояние типа устройства | Unavailable               | Используемый диск может не быть raw-диском или отсутствовать.                                  |

- Оповещения в реальном времени, срабатывающие на вкладке **Alert**, требуют оперативного внимания, даже если текущее состояние кластера хранения отображается как **Healthy**. Быстрая реакция предотвращает развитие более серьёзных проблем. В следующей таблице приведены уровни оповещений и их значение:

| Уровень оповещения | Значение                                                                                                                  |
|--------------------|---------------------------------------------------------------------------------------------------------------------------|
| Critical           | Указывает на серьёзные проблемы, вызывающие перебои в работе платформы или потерю данных с серьёзными последствиями.      |
| Major              | Известные проблемы, которые могут повлиять на функциональность платформы и нормальное ведение бизнеса.                    |
| Warning            | Существует риск операционных проблем; требуется своевременное вмешательство, чтобы избежать влияния на нормальную работу. |

### Анализ после инцидента

В журнале **Alert History** фиксируются все ранее сработавшие оповещения, которые больше не требуют немедленных действий. При анализе после инцидента следует рассмотреть следующие вопросы:

- Какие конкретные аномалии наблюдались в момент инцидента?
- Есть ли повторяющиеся шаблоны конкретных оповещений? Как их можно проактивно предотвратить в будущем?
- Был ли всплеск оповещений в определённые периоды, связанный с внешними факторами или операционными инцидентами? Следует ли скорректировать операционные стратегии соответственно?

## Безопасность

### MWS Container Security

MWS Container Security — это комплексное решение для обеспечения безопасности, разработанное для Kubernetes и контейнеризованных сред. Оно предоставляет централизованное управление, автоматизированное сканирование уязвимостей, применение политик и проверки соответствия, помогая организациям защищать свою контейнерную инфраструктуру в нескольких кластерах.

MWS Container Security использует распределённую архитектуру на основе контейнеров, состоящую из Central Services (для управления, API и UI) и Secured Cluster Services (для мониторинга, применения политик и сбора данных). Оно интегрируется с CI/CD пайплайнами, SIEM, системами логирования и поддерживает встроенный сканер уязвимостей Scanner V4.

Задачи, решаемые MWS Container Security:

- **Управление уязвимостями:** MWS Container Security непрерывно сканирует образы контейнеров и запущенные рабочие нагрузки на наличие известных уязвимостей, помогая предотвратить эксплуатацию, которая может привести к отказу в обслуживании, удалённому выполнению кода или несанкционированному доступу к данным.
- **Применение политик:** Определяя и применяя политики безопасности, MWS Container Security помогает предотвратить рискованные развертывания и обеспечивает своевременное реагирование на инциденты безопасности во время выполнения.
- **Автоматизация соответствия:** MWS Container Security автоматизирует проверки соответствия отраслевым стандартам и нормативным требованиям CIS, обеспечивая прозрачность статуса соответствия и помогая организациям эффективно устранять выявленные пробелы.
- **Централизованный обзор:** Предоставляет единый портал для мониторинга событий безопасности, статуса соответствия и нарушений политик во всех защищённых кластерах.
- **Интеграция:** MWS Container Security интегрируется с CI/CD пайплайнами, реестрами образов и сторонними инструментами для оптимизации безопасности на протяжении всего жизненного цикла контейнеров.

### Архитектура

MWS Container Security использует распределённую архитектуру на основе контейнеров для масштабируемой и малозатратной безопасности на платформе MWS Container Platform или кластерах Kubernetes.

## Основные компоненты

- **Центральные сервисы:** Развёртываются в одном кластере, обеспечивают управление, API и UI (MWS Container Security Portal). Включают Central, Central DB (PostgreSQL 13) и сканер уязвимостей Scanner V4.
- **Сервисы защищённого кластера:** Развёртываются в каждом защищаемом кластере. Включают Sensor (мониторинг кластера и применение политик), Admission Controller (применение политик при приёме), Collector (сбор данных времени выполнения и сети) и опциональные компоненты сканера.

## Обзор сканера

- **Scanner V4:** Сканер по умолчанию, единственный поддерживаемый сканер с версии 4.7. Поддерживает сканирование образов с учётом языка и ОС. Состоит из Indexer, Matcher и DB.

## Источники уязвимостей

- Scanner V4: OSV, NVD и дополнительные источники ОС.

## Особенности развёртывания

- Оператор устанавливает лёгкий Scanner V4 на каждый кластер для интегрированного сканирования реестров.
- Для установки через Helm необходимо указать **scannerV4.disable=false** для включения лёгкого Scanner V4.
- Если Central и сервисы защищённого кластера находятся в одном пространстве имён, компоненты Scanner V4 разворачиваются только в Central.

## Внешние интеграции

- Сторонние системы (CI/CD, SIEM, логирование, email)
- CLI roxctl
- Реестры образов (автоматическая/ручная интеграция)

## Взаимодействие компонентов

### MWS Container Security со Scanner V4

| Компонент | Направление | Компонент          | Описание                               |
|-----------|-------------|--------------------|----------------------------------------|
| Central   | ↔           | Scanner V4 Indexer | Индексация образов и генерация отчётов |
| Central   | ↔           | Scanner V4 Matcher | Сопоставление уязвимостей и отчётность |
| Sensor    | ↔           | Scanner V4 Indexer | Делегированная индексация образов      |

| Компонент            | Направление | Компонент          | Описание                                       |
|----------------------|-------------|--------------------|------------------------------------------------|
| Scanner V4 Indexer   | →           | Image Registries   | Получение метаданных и слоёв образов           |
| Scanner V4 Matcher   | →           | Scanner V4 Indexer | Запрос индексных отчётов                       |
| Scanner V4 Indexer   | →           | Scanner V4 DB      | Хранение результатов индексации                |
| Scanner V4 Matcher   | →           | Scanner V4 DB      | Хранение и обновление данных об уязвимостях    |
| Sensor               | ↔           | Central            | Синхронизация конфигурации и событий           |
| Collector            | ↔           | Sensor             | Отправка данных времени выполнения и сети      |
| Admission controller | ↔           | Sensor             | Применение политик и запросы сканирования      |
| Admission controller | →           | Central            | Прямое взаимодействие при недоступности Sensor |

#### Стандартные порты и протоколы

| Соединение                      | Тип        | Порт | Примечания                         |
|---------------------------------|------------|------|------------------------------------|
| Central ↔ Scanner V4 Indexer    | gRPC       | 8443 |                                    |
| Central ↔ Sensor                | TCP/gRPC   | 443  | Двунаправленное, инициирует Sensor |
| Central ↔ CLI                   | gRPC/HTTPS | 443  | См. roxctl                         |
| Central ↔ Потоки уязвимостей    | HTTPS      | 443  | definitions.stackrox.io            |
| Collector → Sensor              | gRPC       | 443  |                                    |
| Collector (Compliance) → Sensor | gRPC       | 8444 | При включённом сканировании узлов  |
| Scanner V4 Indexer → Central    | HTTPS      | 443  |                                    |
| Scanner V4 Indexer/Matcher → DB | TCP        | 5432 |                                    |
| Sensor ↔ Admission Controller   | gRPC       | 443  | Двунаправленное                    |



## Процесс управления уязвимостями

Управление уязвимостями — это непрерывный процесс выявления и устранения уязвимостей. MWS Container Security помогает вам организовать эффективный процесс управления уязвимостями.

### Основные этапы управления уязвимостями

Успешная программа управления уязвимостями обычно включает следующие ключевые задачи:

- Оценка активов
- Приоритизация уязвимостей
- Оценка воздействия
- Принятие мер
- Непрерывная переоценка

MWS Container Security позволяет организациям постоянно оценивать их MWS Container Platform и Kubernetes кластеры, предоставляя контекстную информацию, необходимую для более эффективной приоритизации и устранения уязвимостей.

### Оценка активов

Для оценки активов вашей организации выполните следующие шаги:

1. Определите активы в вашей среде.
2. Просканируйте эти активы для обнаружения известных уязвимостей.
3. Сообщите об уязвимостях соответствующим заинтересованным сторонам.

При установке MWS Container Security в ваш Kubernetes или кластер MWS Container Platform, система агрегирует активы, работающие внутри кластера, чтобы помочь вам их идентифицировать. MWS Container Security позволяет организациям выполнять постоянные оценки и предоставляет контекст, необходимый для эффективной приоритизации и устранения уязвимостей.

### Ключевые активы для мониторинга

Ключевые активы для мониторинга в процессе управления уязвимостями с использованием MWS Container Security включают:

- **Компоненты:** программные пакеты, используемые в составе образа или работающие на узле. Компоненты — это самый низкий уровень, на котором существуют уязвимости. Организации должны обновлять, изменять или удалять программные компоненты для устранения уязвимостей.
- **Образы:** наборы программных компонентов и кода, создающие среду для запуска исполняемого кода. Образы — это место, где обновляют компоненты для исправления уязвимостей.

- **Узлы:** серверы, используемые для управления и запуска приложений с помощью MWS Container Platform или Kubernetes, включая компоненты, составляющие платформу или сервис.

MWS Container Security организует эти активы в следующие структуры:

- **Deployment:** определение приложения в Kubernetes, которое может запускать поды с контейнерами на основе одного или нескольких образов.
- **Namespace:** группировка ресурсов, таких как Deployments, которые поддерживают и изолируют приложение.
- **Cluster:** группа узлов, используемых для запуска приложений с помощью MWS Container Platform или Kubernetes.

## Сканирование и оценка уязвимостей

MWS Container Security сканирует активы на наличие известных уязвимостей и использует данные Common Vulnerabilities and Exposures (CVE) для оценки их воздействия.

## Приоритизация уязвимостей

Для приоритизации уязвимостей для действий и расследований учитывайте следующие вопросы:

- Насколько важен затронутый актив для вашей организации?
- Насколько серьезной должна быть уязвимость, чтобы требовать расследования?
- Можно ли исправить уязвимость путем патча затронутого программного компонента?
- Нарушает ли уязвимость какие-либо политики безопасности вашей организации?

Ответы на эти вопросы помогают командам безопасности и разработки определить уровень воздействия и необходимую реакцию на уязвимость.

MWS Container Security предоставляет инструменты для упрощения приоритизации уязвимостей в ваших приложениях и компонентах. Вы можете использовать данные, предоставляемые MWS Container Security, чтобы решить, какие уязвимости критично устранить. Например, при рассмотрении результатов сканирования уязвимостей по CVE учитывайте следующие данные, предоставляемые MWS Container Security, для сортировки и приоритизации уязвимостей:

- **CVE severity:** количество образов, затронутых CVE, и рейтинг серьезности (например, низкий, умеренный, важный или критический).
- **Top CVSS:** наивысший балл Common Vulnerability Scoring System (CVSS) от поставщиков для данного CVE по образам.
- **Top NVD CVSS:** наивысший балл CVSS из National Vulnerability Database для данного CVE по образам. Для просмотра этих данных должен быть включен Scanner V4.
- **EPSS probability:** вероятность эксплуатации уязвимости согласно Exploit Prediction Scoring System (EPSS). Это процентная оценка вероятности того, что эксплуатация будет зафиксирована в течение следующих 30 дней. Данные EPSS следует

использовать вместе с другой информацией, например, возрастом CVE, для помощи в приоритизации уязвимостей.

### Оценка воздействия

Для оценки вашего воздействия уязвимости задайте себе вопросы:

- Влияет ли уязвимость на ваше приложение?
- Смягчается ли уязвимость другими факторами?
- Известны ли угрозы, которые могут привести к эксплуатации?
- Используете ли вы уязвимый программный пакет?
- Стоит ли тратить время на устранение именно этой уязвимости и пакета?

### Принятие мер

На основе вашей оценки вы можете предпринять следующие действия:

- Отметить уязвимость как ложное срабатывание, если воздействия нет или она не применима в вашей среде.
- Решить, устранять, смягчать или принимать риск, если вы подвержены воздействию.
- Удалить или изменить программный пакет, чтобы снизить поверхность атаки.

После принятия решения о действиях по уязвимости вы можете:

- Устранить уязвимость,
- Смягчить и принять риск,
- Принять риск,
- Отметить уязвимость как ложное срабатывание.

### Методы устранения

Для устранения уязвимостей вы можете:

- Удалить программный пакет,
- Обновить программный пакет до версии без уязвимостей.

## MWS Compliance с Kyverno

MWS Compliance с Kyverno — это сервис платформы, который интегрирует Kyverno для управления политиками соответствия в платформе MWS Container Platform.

### Установка

1. Перейдите в Platform Management.
2. В левой навигационной панели нажмите **Marketplace > Cluster Plugins**.
3. Найдите MWS Container Platform Compliance with Kyverno и кликните для просмотра деталей.
4. Нажмите **Install**.

### Удаление

1. Перейдите в Platform Management.

2. В левой навигационной панели нажмите **Marketplace > Cluster Plugins**.
3. Найдите MWS Container Platform Compliance with Kyverno и кликните для просмотра деталей.
4. Нажмите **Uninstall**.

## API Refiner

### Обзор

MWS-CP API Refiner — это сервис фильтрации данных, предоставляемый платформой MWS Container Platform, который повышает безопасность мультиарендности и изоляцию данных в Kubernetes-средах. Он фильтрует данные ответов Kubernetes API на основе прав пользователя, проектов, кластеров и пространств имён, а также поддерживает фильтрацию на уровне полей, включение и десенситизацию данных.

### Преимущества:

- Многомерная изоляция данных
  - Поддержка фильтрации ответов API по измерениям проекта, кластера и пространства имён
  - Обеспечение корректных границ данных между разными проектами
  - Предотвращение несанкционированного доступа к ресурсам с областью действия кластера
- Гибкая фильтрация данных
  - Поддержка исключения, включения и десенситизации конкретных полей в ответах API
  - Настраиваемые правила фильтрации через YAML-конфигурацию
  - Динамическая генерация Ingress для разных типов ресурсов
- Повышенная безопасность
  - Реализация аутентификации пользователей на основе JWT-токенов
  - Предоставление тонкой настройки контроля доступа на основе прав пользователя
  - Поддержка десенситизации данных для конфиденциальной информации

### Сценарии использования

Основные сценарии использования:

- Мультиарендная среда:
  - Обеспечение корректной изоляции данных между разными проектами
  - Предотвращение несанкционированного доступа к ресурсам с областью действия кластера
  - Эффективное управление сценариями с общими пространствами имён
- Защита конфиденциальных данных
  - Фильтрация конфиденциальной информации из ответов API

- Поддержка десенситизации данных на уровне полей
- Защита конфиденциальных метаданных и аннотаций
- Соответствие требованиям
  - Помощь в выполнении требований по изоляции данных
  - Поддержка аудита и требований соответствия
  - Поддержание границ доступа к данным

## Ограничения

- Ресурсы должны содержать определённые метки, связанные с проектами, для изоляции данных:
  - `сраас.іо/project`
  - `сраас.іо/cluster`
  - `сраас.іо/namespace`
  - `kubernetes.іо/metadata.name`
  - Необязательно: `сраас.іо/creator`
- Запросы **LabelSelector** не поддерживают логическую операцию OR
- Пользовательские привязки на уровне платформы не фильтруются
- Фильтрация применяется только к операциям API GET и LIST

## Установка API Refiner

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Marketplace > Cluster Plugins**.
3. Выберите кластер **global** в верхней навигационной панели.
4. Найдите **MWS-CP API Refiner** и нажмите для просмотра деталей.
5. Нажмите **Install**.

## Удаление API Refiner

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Marketplace > Cluster Plugins**.
3. Выберите кластер **global** в верхней навигационной панели.
4. Найдите **MWS-CP API Refiner** и нажмите для просмотра деталей.
5. Нажмите **Uninstall**.

## Пользователи

Платформа поддерживает аутентификацию пользователей и проверку входа для всех пользователей.

### Источники пользователей

#### *Локальные пользователи*

- Учётная запись администратора, созданная при развёртывании платформы.

- Учётные записи, созданные через интерфейс платформы.
- Пользователи, добавленные через локальный конфигурационный файл dex.

### **Сторонние пользователи**

#### **LDAP-пользователи**

- Корпоративные пользователи, синхронизированные с LDAP-серверов.
- Учётные записи импортируются через интеграцию с IDP (Identity Provider).
- Источник отображается как имя конфигурации IDP.
- Интеграция настраивается через параметры IDP.

#### **OIDC-пользователи**

- Пользователи сторонних платформ, аутентифицированные через протокол OIDC.
- Источник отображается как имя конфигурации IDP.
- Интеграция настраивается через параметры IDP.

**Внимание:** Для OIDC-пользователей, добавленных в проект до их первого входа:

- Источник отображается как "-" до успешного входа в платформу,
- После успешного входа источник меняется на имя конфигурации IDP.

#### **Другие сторонние пользователи**

- Пользователи, аутентифицированные через поддерживаемые коннекторы dex (например, GitHub, Microsoft).
- Для получения дополнительной информации см. официальную документацию dex.

#### **Правила учётных записей**

- Локальные имена пользователей должны быть уникальны среди всех типов пользователей.
- Пользователи третьих сторон (OIDC/LDAP) с совпадающими именами автоматически связываются.
- Связанные пользователи наследуют права от существующих учётных записей.
- Пользователи могут входить через соответствующие источники.
- В платформе отображается только одна запись пользователя на имя пользователя.
- Источник пользователя определяется по последнему способу входа.

#### **Жизненный цикл пользователя**

| Статус   | Описание                                                                                                                              |
|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| Normal   | Учётная запись пользователя активна и он может войти в платформу                                                                      |
| Disabled | Учётная запись пользователя неактивна и он не может войти. Обратитесь к администратору платформы для активации.<br>Возможные причины: |

| Статус  | Описание                                                                                                                                                                                                                                                                                                                                                   |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <ul style="list-style-type: none"> <li>Отсутствие входа в течение 90+ дней подряд.</li> <li>Истечение срока действия учетной записи.</li> <li>Ручное отключение администратором</li> </ul>                                                                                                                                                                 |
| Locked  | <p>Учётная запись временно заблокирована из-за 5 неудачных попыток входа в течение 24 часов.</p> <p>Особенности:</p> <ul style="list-style-type: none"> <li>Продолжительность блокировки: 20 минут.</li> <li>Может быть разблокирована вручную администратором.</li> <li>После окончания периода блокировки учетная запись становится доступной</li> </ul> |
| Invalid | <p>Учётная запись, синхронизированная с LDAP, которая была удалена с LDAP-сервера.</p> <p><b>Примечание:</b> недействительные учётные записи не могут войти в платформу</p>                                                                                                                                                                                |

### Создание пользователя

Пользователи с ролями администратора платформы могут создавать локальных пользователей и назначать им роли через интерфейс платформы.

Чтобы создать пользователя:

1. Перейдите в **Platform Management**.
2. В левой панели навигации нажмите **Users > Users**.
3. Нажмите Create User.
4. Настройте следующие параметры:

| Параметр      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Password Type | <p>Выберите способ генерации пароля:</p> <ul style="list-style-type: none"> <li><b>Random:</b> Система генерирует безопасный случайный пароль.</li> <li><b>Custom:</b> Пользователь вводит пароль вручную</li> </ul>                                                                                                                                                                                                           |
| Password      | <p>Введите или сгенерируйте пароль в зависимости от выбранного типа.</p> <p>Требования к паролю:</p> <ul style="list-style-type: none"> <li>Длина: 8-32 символа.</li> <li>Должен содержать буквы и цифры.</li> <li>Должен содержать специальные символы (~!@#%&amp;*() - _ = + ?).</li> </ul> <p>Функции поля пароля:</p> <ul style="list-style-type: none"> <li>Нажмите на иконку глаза для показа/скрытия пароля.</li> </ul> |

| Параметр          | Описание                                                                                                                                                                                                                                                  |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <ul style="list-style-type: none"> <li>Нажмите на иконку копирования для копирования пароля</li> </ul>                                                                                                                                                    |
| Mailbox           | Электронная почта пользователя: <ul style="list-style-type: none"> <li>Должна быть уникальной.</li> <li>Может использоваться как имя пользователя для входа.</li> <li>Связана с именем пользователя</li> </ul>                                            |
| Validity Period   | Установите срок действия учетной записи пользователя: <ul style="list-style-type: none"> <li><b>Permanent:</b> Без ограничения по времени.</li> <li><b>Custom:</b> Установите время начала и окончания с помощью выпадающего списка Time Range</li> </ul> |
| Roles             | Назначьте пользователю одну или несколько ролей                                                                                                                                                                                                           |
| Continue Creating | Переключатель для управления поведением после создания: <ul style="list-style-type: none"> <li><b>On:</b> Перенаправляет на страницу создания нового пользователя</li> <li><b>Off:</b> Показывает страницу с деталями созданного пользователя</li> </ul>  |

#### 4. Нажмите **Create**.

После создания пользователя в зависимости от настроек опции **Continue Creating**:

- Если опция включена, вы будете перенаправлены на создание следующего пользователя.
- Если опция отключена, отобразится страница с деталями созданного пользователя.

### Управление пользователями

Платформа предоставляет гибкие возможности управления пользователями, поддерживая как индивидуальное управление, так и пакетные операции для повышения эффективности в определённых сценариях (например, для команд на месте или удалённых команд).

#### Ограничения:

- Системные учётные записи управлению не подлежат (роль администратора платформы, локальный источник).
- Текущие вошедшие в систему пользователи не могут управлять своими собственными учётными записями.
- Для изменения личных данных (отображаемое имя, пароль) используйте страницу личной информации.

#### **Сброс пароля локального пользователя**

Пользователи с правами управления платформой могут сбрасывать пароли других локальных пользователей.

Чтобы сбросить пароль локального пользователя:

1. Перейдите в **Platform Management**.



2. В левой панели выберите **Users > Users**.
3. В меню нужного пользователя нажмите **Reset Password**.
4. В диалоговом окне выберите тип пароля:
  - **Random**: Система сгенерирует надёжный случайный пароль.
  - **Custom**: Введите новый пароль вручную.

Особенности поля пароля:

- Нажмите на иконку глаза для показа/скрытия пароля.
- Нажмите на иконку копирования для копирования пароля.

Требования к паролю:

- Длина: 8-32 символа.
  - Должен содержать буквы и цифры.
  - Должен содержать специальные символы (~!@#%\$^&\*() - \_ = + ?).
5. Нажмите **Reset**.

### ***Обновление даты истечения срока действия пользователя***

Вы можете обновлять даты истечения срока действия пользователей со статусом **normal**, **disabled** или **locked**. Пользователи, у которых срок действия истёк, будут автоматически отключены.

Чтобы обновить дату:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Users**.
3. Нажмите **Update Expiry Date** рядом с нужным пользователем.
4. В диалоговом окне выберите опцию даты истечения:
  - **Permanent**: Без ограничения по времени,
  - **Custom**: Установите время начала и окончания с помощью выпадающего списка Time Range.
5. Нажмите **Update**.

### ***Активация пользователя***

Вы можете активировать пользователей со статусом **disabled** или **locked**.

Поведение при активации:

- Если пользователь находится в пределах срока действия: дата истечения остаётся без изменений,
- Если срок действия пользователя истёк: дата истечения становится **Permanent**.

Чтобы активировать пользователя:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Users**.
3. В меню нужного пользователя нажмите **Activate**.

4. Подтвердите действие. Статус пользователя изменится на **normal**.

### **Отключение пользователя**

Вы можете отключить пользователей со статусом **normal** или **locked** в пределах срока действия. Отключённые пользователи не могут войти в систему, но могут быть повторно активированы.

Чтобы отключить пользователя:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Users**.
3. В меню нужного пользователя нажмите **Disable** и подтвердите действие.

### **Добавление пользователя в локальную группу пользователей**

Вы можете добавить пользователей с **Source** равным **Local** или **LDAP** в одну или несколько локальных групп пользователей.

Поведение ролей групп:

- Пользователи автоматически наследуют роли своих групп.
- Роли групп видны только на странице деталей группы (вкладка **Configure Roles**).
- Список ролей отдельного пользователя показывает только роли, назначенные напрямую.

Чтобы добавить пользователя в группу:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Users**.
3. В меню нужного пользователя нажмите **Add to User Group**.
4. Выберите одну или несколько локальных групп пользователей.
5. Нажмите **Add**.

### **Удаление пользователя**

Администраторы платформы могут удалять любых пользователей, кроме текущей залогиненной учётной записи, включая:

- Пользователей, настроенных через IDP,
- Пользователей с источником “-”,
- Локальных пользователей.

Чтобы удалить пользователя:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Users**.
3. В меню нужного пользователя нажмите **Delete**.
4. Нажмите **Confirm**.

### **Пакетные операции**

Вы можете выполнять пакетные операции для:

- Обновления сроков действия,
- Активации пользователей,
- Отключения пользователей,
- Удаления пользователей.

Чтобы выполнить пакетную операцию:

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > Users**.
3. Выберите одного или нескольких пользователей с помощью чекбоксов
4. Нажмите **Batch Operations** и выберите нужное действие:
  - **Update Validity** – установка постоянного или пользовательского временного диапазона,
  - **Activate** – подтверждение активации в диалоговом окне,
  - **Deactivate** – подтверждение деактивации в диалоговом окне,
  - **Delete** – Ввод пароля текущей учётной записи и подтверждение.

### **Управление ролями пользователей**

Администраторы платформы могут управлять ролями других пользователей (не своей собственной учётной записи), чтобы предоставлять или отзывать разрешения.

#### **Назначение роли пользователю**

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > Users**.
3. Нажмите на имя пользователя, для которого нужно изменить роли.
4. Перейдите на вкладку **Configure Roles**.
5. Нажмите **Add Role**.
6. В диалоговом окне назначения роли задайте:
  - роль из выпадающего списка **Role Name**,
  - область действия разрешений роли (кластер, проект или пространство имён).
7. Нажмите **Add**.

#### **Примечание:**

- Вы можете назначить пользователю несколько ролей.
- Каждая роль может быть добавлена только один раз для каждого пользователя.
- Уже назначенные роли отображаются в выпадающем списке, но выбрать их нельзя.
- Роль **Cluster Administrator** нельзя назначить для глобального кластера.

#### **Отзыв роли у пользователя**

1. Перейдите в Platform Management.
2. В левой панели выберите **Users > Users**.
3. Нажмите на имя пользователя, для которого нужно изменить роли

4. Перейдите на вкладку **Configure Roles**.
5. Нажмите **Remove** рядом с ролью, которую хотите удалить
6. Подтвердите удаление.

**Внимание:** Разрешения на управление ролями:

- Управлять ролями других пользователей могут только администраторы платформы.
- Пользователи не могут изменять роли своей собственной учётной записи.

## Группы пользователей

### Обзор

Платформа поддерживает управление пользователями через пользовательские группы.

Управляя ролями групп, вы можете эффективно:

- Одновременно предоставлять права на работу с платформой нескольким пользователям,
- Одновременно отзывать права у нескольких пользователей,
- Реализовывать пакетное управление доступом на основе ролей.

Например, при кадровых изменениях в организации, когда необходимо предоставить права на работу с проектом или пространством имён нескольким пользователям, вы можете:

1. Создать пользовательскую группу.
2. Импортировать соответствующих пользователей в члены группы.
3. Настроить роли проекта и пространства имён для группы.
4. Применить единые права ко всем членам группы.

### Типы групп

Платформа поддерживает два типа групп:

- **Локальная пользовательская группа:**
  - Создаётся непосредственно в платформе,
  - Источник отображается как **Local**,
  - Может быть обновлена или удалена,
  - Поддерживает:
    - Добавление или удаление пользователей из любого источника,
    - Добавление или удаление ролей,
- **Синхронизированная с IDP пользовательская группа:**
  - Синхронизируется из подключённого IDP (LDAP, Azure AD),
  - Источник отображается как имя подключённого IDP,
  - Не может быть обновлена или удалена,
  - Поддерживает:

- Добавление или удаление ролей,
- Не поддерживает управление членами группы (добавление или удаление).

### Создание локальной группы пользователей

Локальные группы пользователей позволяют реализовать управление доступом на основе ролей для нескольких пользователей из любого источника.

Чтобы создать группу пользователей:

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > User Groups**.
3. Нажмите **Create User Group**.
4. Введите следующую информацию:
  - **Name:** Название группы пользователей,
  - **Description:** Описание группы.
5. Нажмите **Create**.

### Управление группами пользователей

Управляйте группами пользователей с помощью меню конкретной группы на странице списка групп или выбрав **Operations** в правом верхнем углу на странице с деталями группы:

| Операция                | Описание                                                                                                                                                                                                                                                                                               |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Update User Group       | <p>Обновление информации о группе в зависимости от источника группы:</p> <ul style="list-style-type: none"> <li>• Для групп с <b>Source</b> равным Local: можно обновить как название, так и описание,</li> <li>• Для групп с <b>Source</b> равным IDP name: можно обновить только описание</li> </ul> |
| Delete Local User Group | <p>Удаление групп пользователей с <b>Source</b> равным Local</p> <p>При удалении группы:</p> <ul style="list-style-type: none"> <li>• Все участники группы будут удалены,</li> <li>• Все роли, назначенные группе, будут удалены,</li> <li>• Это действие нельзя отменить</li> </ul>                   |

### Управление ролями групп пользователей

Пользователи с правами управления платформой могут управлять ролями как для локальных групп пользователей, так и для групп пользователей, синхронизированных через IDP.

### **Назначение роли группе**

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > User Groups**.
3. Нажмите на имя нужной группы пользователей.
4. Перейдите на вкладку **Configure Roles**.
5. Нажмите **Add Role**.
6. В диалоговом окне назначения роли задайте:
  - роль из выпадающего списка **Role Name**,
  - область действия разрешений роли (кластер, проект или пространство имён).

Правила назначения ролей:

- Вы можете добавить несколько ролей в одну группу.
  - Каждая роль может быть добавлена в одну группу только один раз.
7. Нажмите **Add**.

### **Отзыв роли у группы**

**Внимание:** При удалении роли из группы:

- Все права, предоставленные этой ролью членам группы, будут отозваны.
- Это действие нельзя отменить.

Чтобы отозвать роль у группы:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > User Groups**.
3. Нажмите на имя нужной группы пользователей.
4. На вкладке **Configure Role** нажмите **Remove** рядом с ролью.
5. Нажмите **Confirm**.

### **Управление пользователями в группе пользователей**

Управлять пользователями в локальных группах пользователей могут только пользователи с правами Platform Management.

### **Ограничения**

- Управлять группами и их участниками могут только пользователи с правами Platform Management.
- Системные аккаунты и аккаунты, в данный момент вошедшие в систему, не подлежат управлению (не могут быть импортированы в группы или удалены из них).
- В каждой локальной группе пользователей может быть не более 5000 участников.
- При достижении лимита в 5000 участников дальнейший импорт невозможен.

## **Импорт участников**

Вы можете импортировать пользователей из платформы в локальные группы пользователей для централизованного управления правами доступа.

Пользователи, импортированные в группу, автоматически наследуют все операционные права, назначенные этой группе.

Чтобы импортировать пользователей в группу:

1. Перейдите в Platform Mamagement.
2. В левой панели выберите **Users > User Groups**.
3. Нажмите на имя нужной группы пользователей.
4. На вкладке Group Member Management нажмите Import Member.
5. Выберите одного или нескольких пользователей платформы, отметив флажками их имена пользователей/отображаемые имена.

Вы можете выбрать только тех пользователей, которые в данный момент не являются членами группы.

Используйте кнопку **Import All**, чтобы импортировать всех пользователей из списка сразу.

6. Нажмите **Import**.

## **Удаление участников**

При удалении пользователя из группы все операционные права, предоставленные этому пользователю через группу, автоматически аннулируются.

Чтобы удалить пользователя из группы:

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > User Groups**.
3. Нажмите на имя нужной группы пользователей.
4. На вкладке **Group Member Management** удалите участников одним из способов:
  - Нажмите **Remove** рядом с именем участника и подтвердите действие,
  - Выберите одного или нескольких участников с помощью флажков, затем нажмите **Batch Remove** и подтвердите действие.

## **Роли**

### **Обзор**

Управление ролями пользователей на платформе реализовано с помощью Kubernetes RBAC (Role-Based Access Control). Эта система позволяет гибко настраивать права доступа, связывая роли с пользователями.

Роль представляет собой набор разрешений для работы с ресурсами Kubernetes на платформе. Эти разрешения включают:

- Создание ресурсов,
- Просмотр ресурсов,

- Обновление ресурсов,
- Удаление ресурсов.

Роли классифицируют и объединяют разрешения для различных ресурсов. Назначая роли пользователям и устанавливая области действия разрешений, можно быстро предоставить права на операции с ресурсами.

Разрешения можно так же легко отозвать, удалив роли у пользователей.

Роль может включать:

- Один или несколько типов ресурсов,
- Одно или несколько разрешений на операции,
- Несколько назначенных пользователей.

Например:

- Роль А: Может только просматривать и создавать проекты,
- Роль В: Может создавать, просматривать, обновлять и удалять пользователей, проекты и namespaces.

Платформа поддерживает системные и пользовательские роли.

### **Системные роли**

Для удовлетворения распространённых сценариев настройки прав доступа платформа предоставляет стандартные системные роли. Эти роли обеспечивают гибкий контроль доступа к ресурсам платформы и эффективное управление правами пользователей.

| Название роли                 | Описание                                                                                     | Уровень роли |
|-------------------------------|----------------------------------------------------------------------------------------------|--------------|
| Platform Administrator        | Имеет полный доступ ко всем бизнес-ресурсам и ресурсам платформы                             | Platform     |
| Platform Auditors             | Может просматривать все ресурсы платформы и записи операций, но не имеет других прав         | Platform     |
| Cluster Administrator (Alpha) | Управляет и поддерживает ресурсы кластера с полным доступом ко всем ресурсам уровня кластера | Cluster      |
| Project Administrator         | Управляет администраторами namespace и квотами namespace                                     | Project      |
| namespace-admin-system        | Управляет участниками namespace и назначениями ролей                                         | Namespace    |
| Developers                    | Разрабатывает, развёртывает и поддерживает кастомные приложения внутри namespaces            | Namespace    |



## Пользовательские роли

Платформа поддерживает пользовательские роли для расширения сценариев контроля доступа к ресурсам. Пользовательские роли обладают рядом преимуществ по сравнению с системными ролями:

- Гибкая настройка разрешений,
- Возможность обновления прав роли,
- Опция удаления ролей, когда они больше не нужны.

**Внимание:** Будьте осторожны при обновлении или удалении пользовательских ролей. Удаление пользовательской роли автоматически отзовёт все права, предоставленные этой ролью связанным пользователям.

## Создание роли

Пользователи с разрешениями ролей платформы могут создавать пользовательские роли с разрешениями, которые равны или меньше их собственных разрешений согласно нужным сценариям использования. При создании роли можно настроить:

- Разрешения на операции функциональных модулей платформы,
- Разрешения доступа к пользовательским ресурсам (Kubernetes CRD).

Чтобы создать роль:

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > Roles**.
3. Нажмите **Create Role**.
4. Настройте основную информацию о роли:
  - имя,
  - описание
  - тип роли. Область разрешений роли будет ограничена в зависимости от типа роли:
    - **Platform Role:** На следующем шаге будут отображаться все разрешения платформы.
    - **Project Role:** На следующем шаге будут отображаться разрешения для разделов платформы:
      - Project Management,
      - Container Platform,
      - Service Mesh,
      - DevOps,
      - Middleware.
    - **Namespace Role:** На следующем шаге будут отображаться разрешения для разделов платформы:
      - Project Management,
      - Container Platform,

- Service Mesh,
- DevOps,
- Middleware.

5. Нажмите **Next**.

6. Настройте разрешения для разных ресурсов платформы. Виды разрешений:

- View – просмотр,
- Create – создание,
- Update – обновление,
- Delete – удаление.

**Примечание:** Разрешения вашей учетной записи ограничивают, какие разрешения вы можете настраивать. Например:

- Если у вашей учётной записи нет разрешения на просмотр **Project Management**
- Пункт **Project Management** будет неактивна при создании роли
- Вы можете создавать роли только с разрешениями, равными или ниже ваших собственных

7. (Опционально) Настройте разрешения для пользовательских ресурсов (Kubernetes CRD) :

7.1. Нажмите Add Custom Permission.

7.2. Задайте параметры:

| Параметр   | Описание                                                                                                                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Groups | <ul style="list-style-type: none"> <li>• <b>System</b> – системная группа разрешений,</li> <li>• <b>Custom</b> – пользовательская группа разрешений</li> </ul> |
| Group Name | Название пользовательской группы разрешений. Группы отображаются в модуле разрешений в порядке их добавления.                                                  |
| Resources  | Название ресурсов. Нажмите Enter для добавления нескольких названий пользовательских ресурсов.                                                                 |

8. Нажмите **Create**.

## Управление пользовательскими ролями

### Обновление основной информации

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Roles**.
3. Нажмите на имя пользовательской роли.
4. Нажмите **Actions > Update Basic Info**.
5. Обновите:

- Отображаемое имя роли,
- Описание.

6. Нажмите **Update**.

#### **Обновление разрешений роли**

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > Roles**.
3. Нажмите на имя пользовательской роли.
4. Нажмите **Manage Permissions**.
5. Внесите изменения:
  - Добавьте новые разрешения на операции с ресурсами платформы,
  - Удалите существующие разрешения,
  - Измените разрешения для пользовательских ресурсов.
6. Нажмите **Confirm**.

#### **Копирование существующей роли**

Вы можете создать новую роль, скопировав существующую роль (системную или пользовательскую). Новая роль унаследует всю информацию о разрешениях исходной роли, которую затем можно изменить в соответствии с вашими потребностями.

**Внимание:** Разрешения новой роли не могут превышать разрешения роли, к которой принадлежит создатель.

Чтобы скопировать роль:

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > Roles**.
3. Нажмите на имя роли, которую нужно скопировать.
4. Нажмите **Actions > Duplicate**.
5. Настройте параметры для новой роли:
  - Имя,
  - Отображаемое имя,
  - Описание,
  - Тип,
  - Разрешения.
6. Нажмите **Create**.

#### **Удаление пользовательской роли**

Вы можете удалить пользовательские роли, которые больше не используются.

При удалении пользовательской роли:

- Связи роли с пользователями будут удалены.

- Пользователи, назначенные на эту роль, потеряют все разрешения, предоставленные ролью.
- Роль будет удалена из списков ролей пользователей.

Чтобы удалить пользовательскую роль:

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > Roles**.
3. Нажмите на имя пользовательской роли, которую нужно удалить.
4. Нажмите **Actions > Delete**.
5. Введите имя роли для подтверждения удаления.
6. Нажмите **Delete**.

## **IDP**

### **Обзор**

Платформа интегрируется с сервисом аутентификации Dex, что позволяет использовать предустановленные коннекторы Dex для аутентификации аккаунтов платформы через настройку IDP. Для получения дополнительной информации обратитесь к официальной документации Dex.

### **Поддерживаемые методы интеграции**

#### ***Интеграция LDAP***

Если в вашей организации используется **LDAP** (Lightweight Directory Access Protocol) для управления пользователями, вы можете настроить LDAP на платформе для подключения к LDAP-серверу вашей организации.

Преимущества интеграции LDAP:

- Обеспечивает связь между платформой и LDAP-сервером,
- Позволяет пользователям организации входить с использованием LDAP-учётных данных,
- Автоматически синхронизирует учётные записи пользователей организации с платформой.

#### ***Интеграция OIDC***

Платформа поддерживает интеграцию с IDP-сервисами, использующими протокол OpenID Connect (OIDC) для аутентификации сторонних пользователей.

Преимущества интеграции OIDC:

- Позволяет пользователям входить с помощью сторонних аккаунтов,
- Поддерживает корпоративные IDP-сервисы,
- Обеспечивает безопасную аутентификацию через протокол OIDC.

**Примечание:** Для аутентификации с использованием других коннекторов, не упомянутых выше, пожалуйста, свяжитесь с технической поддержкой.

## **LDAP**

### **Обзор**

LDAP (Lightweight Directory Access Protocol) — это зрелый, гибкий и хорошо поддерживаемый стандартный механизм взаимодействия с директориями. Он организует данные в иерархическую древовидную структуру для хранения информации о пользователях и организациях предприятия, преимущественно используется для реализации единого входа (SSO).

Администраторы платформы могут добавлять, обновлять и удалять LDAP-сервисы в платформе.

Ключевые особенности LDAP:

- Обеспечивает связь между клиентами и LDAP-серверами,
- Поддерживает операции хранения, извлечения и поиска данных,
- Обеспечивает возможности аутентификации клиентов,
- Способствует интеграции с другими системами.

Для получения дополнительной информации обратитесь к официальной документации LDAP.

### **Поддерживаемые типы LDAP**

#### **OpenLDAP**

OpenLDAP — это реализация LDAP с открытым исходным кодом. Если в вашей организации используется open-source LDAP для аутентификации пользователей, вы можете настроить платформу для взаимодействия с LDAP-сервисом, добавив LDAP и настроив соответствующие параметры.

Интеграция OpenLDAP:

- Обеспечивает аутентификацию пользователей LDAP в платформе,
- Поддерживает стандартные протоколы LDAP,
- Обеспечивает гибкое управление пользователями.

Для получения дополнительной информации об OpenLDAP обратитесь к официальной документации OpenLDAP.

#### **Active Directory**

Active Directory — это LDAP-основанное программное обеспечение Microsoft для предоставления служб хранения каталогов в системах Windows. Если в вашей организации используется Microsoft Active Directory для управления пользователями, вы можете настроить платформу для взаимодействия с сервисом Active Directory.

Интеграция Active Directory:

- Обеспечивает аутентификацию пользователей AD в платформе,
- Поддерживает интеграцию с доменами Windows,
- Обеспечивает управление пользователями корпоративного уровня.

## Терминология LDAP

- Общие термины OpenLDAP:

| Термин                     | Описание                   | Пример                              |
|----------------------------|----------------------------|-------------------------------------|
| dc (Domain Component)      | Компонент домена           | dc=example,dc=com                   |
| ou (Organizational Unit)   | Организационная единица    | ou=People,dc=example,dc=com         |
| cn (Common Name)           | Общее имя                  | cn=admin,dc=example,dc=com          |
| uid (User ID)              | Идентификатор пользователя | uid=example                         |
| objectClass (Object Class) | Класс объекта              | objectClass=inetOrgPerson           |
| mail (Mail)                | Электронная почта          | mail=example@126.com                |
| givenName (Given Name)     | Имя                        | givenName=xq                        |
| sn (Surname)               | Фамилия                    | sn=ren                              |
| objectClass: groupOfNames  | Группа пользователей       | objectClass: groupOfNames           |
| member (Member)            | Атрибут участника группы   | member=cn=admin,dc=example,dc=com   |
| memberOf                   | Атрибут членства в группе  | memberOf=cn=users,dc=example,dc=com |

- Общие термины Active Directory:

| Термин                           | Описание                      | Пример                                               |
|----------------------------------|-------------------------------|------------------------------------------------------|
| dc (Domain Component)            | Компонент домена              | dc=example,dc=com                                    |
| ou (Organizational Unit)         | Организационная единица       | ou=People,dc=example,dc=com                          |
| cn (Common Name)                 | Общее имя                     | cn=admin,dc=example,dc=com                           |
| sAMAccountName/userPrincipalName | Идентификатор пользователя    | userPrincipalName=example или sAMAccountName=example |
| objectClass: user                | Класс объекта пользователя AD | objectClass=user                                     |

| Термин                 | Описание                  | Пример                              |
|------------------------|---------------------------|-------------------------------------|
| mail (Mail)            | Электронная почта         | mail=example@126.com                |
| displayName            | Отображаемое имя          | displayName=example                 |
| givenName (Given Name) | Имя                       | givenName=xq                        |
| sn (Surname)           | Фамилия                   | sn=ren                              |
| objectClass: group     | Группа пользователей      | objectClass: group                  |
| member (Member)        | Атрибут участника группы  | member=CN=Admin,DC=example,DC=com   |
| memberOf               | Атрибут членства в группе | memberOf=CN=Users,DC=example,DC=com |

### Добавление LDAP

После успешной интеграции LDAP:

- Пользователи могут входить на платформу с использованием своих корпоративных учётных записей,
- Множественное добавление одного и того же LDAP перезапишет ранее синхронизированных пользователей.

### Прerequisites

Перед добавлением LDAP подготовьте следующую информацию:

- Адрес LDAP-сервера,
- Имя пользователя администратора,
- Пароль администратора.

### Шаги

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Users > IDPs**.
3. Нажмите **Add LDAP**.
4. Настройте основную информацию:

| Параметр              | Описание                                                                                                   |
|-----------------------|------------------------------------------------------------------------------------------------------------|
| Server Address        | Адрес доступа к LDAP-серверу (например, 192.168.156.141:31758)                                             |
| Username              | DN администратора LDAP (например, cn=admin,dc=example,dc=com)                                              |
| Password              | Пароль учетной записи администратора LDAP                                                                  |
| Login Username Prompt | Сообщение-приглашение для ввода имени пользователя (например, "Пожалуйста, введите ваше имя пользователя") |

5. Настройте параметры LDAP-сервера.

6. Настройте поиск:

| Параметр               | Описание                                                                                                                                                                                                |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Type            | ObjectClass для пользователей: <ul style="list-style-type: none"><li>• OpenLDAP: inetOrgPerson,</li><li>• Active Directory: organizationalPerson,</li><li>• Группы: posixGroup</li></ul>                |
| Login Field            | Атрибут, используемый в качестве имени пользователя для входа: <ul style="list-style-type: none"><li>• OpenLDAP: mail (адрес электронной почты)</li><li>• Active Directory: userPrincipalName</li></ul> |
| Filter Conditions      | Условия фильтра LDAP для фильтрации пользователей/групп<br>Пример: (&(cn=John*)(givenName=*xq*))                                                                                                        |
| Search Starting Point  | Базовый DN для поиска пользователей/групп<br>(например, dc=example,dc=org)                                                                                                                              |
| Search Scope           | Область поиска: <ul style="list-style-type: none"><li>• sub: весь поддерево каталога</li><li>• one: один уровень ниже начальной точки</li></ul>                                                         |
| Login Attribute        | Уникальный идентификатор пользователя: <ul style="list-style-type: none"><li>• OpenLDAP: uid</li><li>• Active Directory: distinguishedName</li></ul>                                                    |
| Name Attribute         | Атрибут имени объекта (по умолчанию: cn)                                                                                                                                                                |
| Email Attribute        | Атрибут электронной почты: <ul style="list-style-type: none"><li>• OpenLDAP: mail</li><li>• Active Directory: userPrincipalName</li></ul>                                                               |
| Group Member Attribute | Идентификатор участника группы (по умолчанию: uid)                                                                                                                                                      |
| Group Attribute        | Атрибут связи с группой пользователей (по умолчанию: memberuid)                                                                                                                                         |

**Примечание:** Назначение настроек поиска:

- Соответствие LDAP-записям пользователей по заданным условиям,
- Извлечение ключевых атрибутов пользователей и групп,
- Отображение атрибутов LDAP на атрибуты пользователей платформы.



7. В разделе **IDP Service Configuration Validation** введите действительное имя пользователя и пароль LDAP-аккаунта. Имя пользователя должно совпадать с настройкой **Login Field**.
8. (Опционально) Настройте **LDAP Auto-Sync Policy**:
  - 1.1. Включите переключатель **Auto-Sync Users**.
  - 1.2. Настройте правила синхронизации.
9. Нажмите **Add**. После добавления LDAP:
  - Пользователи могут входить в платформу до синхронизации,
  - Информация о пользователях синхронизируется автоматически при первом входе,
  - Автоматическая синхронизация происходит согласно настроенным правилам.

### ***Синхронизация пользователей LDAP***

После успешной синхронизации пользователей LDAP на платформу вы можете просмотреть синхронизированных пользователей в списке пользователей.

Вы можете настроить политику автоматической синхронизации при добавлении LDAP (которую можно обновить позже) или вручную запустить синхронизацию после успешного добавления LDAP. Ниже описано, как вручную запустить операцию синхронизации.

#### **Примечания:**

- Новые пользователи, добавленные в LDAP, интегрированный с платформой, могут войти на платформу до выполнения операции синхронизации пользователей. После успешного входа их информация автоматически синхронизируется с платформой.
- Пользователи, удалённые из LDAP, после синхронизации получают статус **Invalid**.
- По умолчанию срок действия вновь синхронизированных пользователей — **Постоянный**.
- Синхронизированные пользователи с тем же именем, что и существующие пользователи (локальные или IDP), автоматически связываются. Их права и срок действия будут соответствовать существующим пользователям. Они могут входить на платформу, используя метод входа, соответствующий их источнику.

#### **Шаги**

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > IDPs**.
3. Нажмите на имя LDAP, для которого хотите выполнить ручную синхронизацию.
4. Нажмите **Actions > Sync user**.
5. Нажмите **Sync**.

**Примечание:** Если вы вручную закроете диалоговое окно синхронизации, появится диалог подтверждения закрытия. После закрытия диалога система продолжит синхронизацию пользователей. Если вы остаётесь на странице списка пользователей, вы получите обратную

связь о результате синхронизации. Если покинете страницу списка пользователей, результат синхронизации не будет получен.

### Управление LDAP

Чтобы обновить или удалить LDAP, перейдите на страницу LDAP и нажмите **Actions**:

| Операция      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Обновить LDAP | Обновить конфигурацию добавленного LDAP или LDAP Auto-Sync Policy.<br><b>Примечание:</b> После обновления LDAP пользователи, синхронизированные с платформой через этот LDAP, также будут обновлены. Пользователи, удалённые из LDAP, станут недействительными в списке пользователей платформы. Для очистки данных выполните операцию очистки недействительных пользователей                                                                                                                                                                   |
| Удалить LDAP  | После удаления LDAP все пользователи, синхронизированные с платформой через этот LDAP, получают статус <b>Invalid</b> (связь между пользователями и ролями сохраняется), и они не смогут войти в платформу.<br>После повторной интеграции необходимо повторно выполнить синхронизацию для активации пользователей.<br><b>Совет:</b> После удаления IDP, если необходимо удалить пользователей и группы пользователей, синхронизированные с платформой через LDAP, установите флажок <b>Clean IDP Users and User Groups</b> в окне подтверждения |

## OIDC

### Обзор

Платформа поддерживает протокол OIDC (OpenID Connect), позволяя администраторам платформы входить в систему с использованием сторонних аккаунтов после добавления конфигурации OIDC. Администраторы платформы также могут обновлять и удалять настроенные сервисы OIDC.

OIDC (OpenID Connect) — это стандартный протокол аутентификации пользователя, основанный на протоколе OAuth 2.0. Он использует сервер авторизации OAuth 2.0 для предоставления аутентификации пользователя сторонним клиентам и передачи соответствующей информации об аутентификации пользователя клиенту.

OIDC позволяет всем типам клиентов (включая серверные, мобильные и JavaScript-клиенты) запрашивать и получать аутентифицированные сессии и информацию о конечном пользователе. Этот набор спецификаций является расширяемым, что позволяет участникам использовать дополнительные функции, такие как шифрование данных личности, обнаружение OpenID Provider и управление сессиями. Для получения дополнительной информации обратитесь к официальной документации OIDC.

## Добавление OIDC

Добавив OIDC, вы сможете использовать сторонние аккаунты для входа в платформу.

**Примечание:** После успешного входа пользователей OIDC в платформу платформа будет использовать атрибут email пользователя в качестве уникального идентификатора. Пользователи сторонних платформ с поддержкой OIDC должны иметь атрибут **email**, иначе они не смогут войти в платформу.

### Шаги

1. Перейдите в **Platform Management**.
  2. В левой панели выберите **Users > IDPs**.
  3. Нажмите **Add OIDC**.
  4. Настройте параметры **Basic Information**.
  5. Настройте параметры OIDC Server Configuration:
    - **Identity Provider URL:** адрес доступа к провайдеру идентификации OIDC.
    - **Client ID:** идентификатор клиента для OIDC клиента.
    - **Client Secret:** секретный ключ для OIDC клиента.
    - **Redirect URI:** адрес обратного вызова после входа в стороннюю платформу, который представляет собой URL провайдера dex + /callback.
    - **Logout URL:** адрес, который посещает пользователь после выполнения операции **Logout**. Если пусто, адрес выхода будет начальной страницей входа платформы.
  6. В области **IDP Service Configuration Validation** введите **Username** и **Password** действительной учётной записи OIDC для проверки конфигурации.
- Примечание:** Если имя пользователя и пароль неверны, при добавлении будет выдана ошибка с указанием недействительных учётных данных, и OIDC не сможет быть добавлен.
7. Нажмите **Create**.

### Управление OIDC

Чтобы обновить или удалить OIDC, перейдите на страницу OIDC и нажмите **Actions**:

| Операция      | Описание                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Обновить OIDC | Обновить добавленную конфигурацию OIDC. После обновления конфигурации OIDC исходные пользователи и методы аутентификации будут сброшены и синхронизированы в соответствии с текущей конфигурацией                                                                                                                                                                              |
| Удалить OIDC  | Удалить OIDC, который больше не используется платформой. После удаления OIDC все пользователи, синхронизированные с платформой через этот OIDC, получают статус <b>Invalid</b> (связь между пользователями и ролями сохраняется), и они не смогут войти в платформу.<br><br>После повторной интеграции пользователи могут быть активированы путем успешного входа в платформу. |

| Операция | Описание                                                                                                                                                                                                               |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <b>Совет:</b> После удаления IDP, если необходимо удалить пользователей и группы пользователей, синхронизированные с платформой через OIDC, отметьте флажок <b>Clean IDP Users and User Groups</b> во всплывающем окне |

## Политики безопасности

### Обзор

Платформа предоставляет комплексные политики безопасности для повышения безопасности входа и защиты от вредоносных атак.

Платформа поддерживает следующие политики безопасности:

- Управление безопасностью паролей,
- Отключение учётных записей пользователей,
- Блокировка учётных записей пользователей,
- Уведомления пользователей,
- Контроль доступа.

### Доступные политики безопасности

| Политика                   | Описание                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User Authentication Policy | Включает двойную аутентификацию для входа по паролю: <ul style="list-style-type: none"> <li>• Пользователи получают коды подтверждения через указанные методы уведомлений</li> <li>• Поддерживает различные серверы уведомлений (например, Enterprise Communication Tool Server)</li> </ul>                                                                                                                              |
| Password Security Policy   | Управляет требованиями к паролям: <ul style="list-style-type: none"> <li>• Первый вход: <ul style="list-style-type: none"> <li>○ Обязательная смена пароля при первом входе на платформу</li> </ul> </li> <li>• Регулярное обновление: <ul style="list-style-type: none"> <li>○ Требуется смена пароля после заданного периода (например, 90 дней),</li> <li>○ Вход запрещён до обновления пароля</li> </ul> </li> </ul> |
| User Disablement Policy    | Автоматически отключает неактивные аккаунты: Срабатывает после заданного периода отсутствия входа                                                                                                                                                                                                                                                                                                                        |
| User Locking Policy        | Защищает от атак перебором паролей: <ul style="list-style-type: none"> <li>• Условия блокировки:</li> </ul>                                                                                                                                                                                                                                                                                                              |

| Политика            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>○ Срабатывает после заданного количества неудачных попыток входа в течение 24 часов</li> <li>• Длительность блокировки: <ul style="list-style-type: none"> <li>○ Аккаунт остаётся заблокированным в течение заданного времени в минутах,</li> <li>○ Автоматически разблокируется после истечения периода блокировки</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Notification Policy | Управляет уведомлениями пользователей:<br>Отправляет первоначальный пароль по электронной почте после создания пользователя                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Access Control      | <p>Управляет сессиями пользователей и доступом:</p> <ul style="list-style-type: none"> <li>• Управление сессиями: <ul style="list-style-type: none"> <li>○ Автоматический выход из неактивных сессий после заданного времени,</li> <li>○ Ограничение максимального количества одновременных онлайн-пользователей</li> </ul> </li> <li>• Контроль браузера: <ul style="list-style-type: none"> <li>○ Завершение сессии при закрытии всех вкладок продукта,</li> <li>○ Запрет на множественные входы с одного клиента</li> </ul> </li> </ul> <p><b>Внимание:</b></p> <ul style="list-style-type: none"> <li>• Контроль доступа влияет только на новые входы после обновления политики.</li> <li>• Восстановление вкладок браузера может не привести к завершению сессии.</li> <li>• При запрете повторного входа разрешён только последний вход с клиента.</li> </ul> |

#### Настройка политик безопасности

1. Перейдите в **Platform Mamagement**.
2. В левой панели выберите **Users > Policies**.
3. Нажмите **Update**.
4. Настройте политики безопасности:
  - Отметьте галочкой политику для её включения,
  - Снимите галочку для отключения политики.

Отключённые политики сохраняют свои настройки.

При повторном включении политики восстанавливаются предыдущие настройки.

5. Нажмите **Update** для сохранения изменений.

## Проекты

### Обзор

Проект — это единица изоляции ресурсов, которая обеспечивает сценарии мультиарендного использования в предприятиях. Он разделяет ресурсы одного или нескольких кластеров на изолированные среды, обеспечивая как изоляцию ресурсов, так и изоляцию персонала. Проекты могут представлять различные дочерние компании, отделы или проектные команды внутри предприятия. С помощью управления проектами достигаются:

- Изоляция ресурсов между проектными командами,
- Управление квотами внутри арендаторов,
- Эффективное распределение и контроль ресурсов.

### Пространства имён (Namespaces)

Пространства имён — это меньшие, взаимно изолированные пространства ресурсов внутри проекта. Они служат рабочими областями для пользователей для реализации своих производственных нагрузок. Основные характеристики:

- В рамках проекта можно создавать несколько пространств имён,
- Общая квота ресурсов всех пространств имён не может превышать квоту проекта,
- Квоты ресурсов выделяются гранулярно на уровне пространства имён,
- Размеры контейнеров (CPU, память) ограничиваются на уровне пространства имён,
- Повышенная эффективность использования ресурсов за счёт тонкого контроля.

### Взаимосвязь между кластерами, проектами и пространствами имён

Иерархия ресурсов платформы подчиняется следующим правилам:

- Проект может использовать ресурсы (CPU, память, хранилище) из нескольких кластеров, а кластер может выделять ресурсы нескольким проектам,
- В рамках проекта можно создавать несколько пространств имён, при этом суммарные квоты ресурсов не должны превышать общие ресурсы проекта,
- Квота ресурсов пространства имён должна поступать из одного кластера, и пространство имён может принадлежать только одному проекту.

### Создание проекта

Перед началом работы вашей проектной команды вы можете создать проект на основе существующих ресурсов кластера в платформе. Проект будет изолирован от других проектов как по ресурсам, так и по персоналу. При создании проекта вы можете выделить ресурсы в соответствии с масштабом проекта и фактическими бизнес-потребностями. Проект может использовать ресурсы из нескольких кластеров на платформе.

**Внимание:** При создании проекта платформа автоматически создаст пространство имён с таким же именем, как у проекта, в связанных кластерах для изоляции ресурсов на уровне платформы. **Запрещается** изменять это пространство имён и его ресурсы.

Чтобы создать проект:

1. Перейдите в **Project Management**.
2. Нажмите **Create Project**.
3. На странице **Basic information** настройте следующие параметры:

| Параметр | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name     | <p>Имя проекта, которое не может совпадать с именем существующего проекта или любым именем из черного списка имён проектов. В противном случае проект не может быть создан.</p> <p><b>Примечание:</b> Черный список имён проектов включает специальные имена пространств имён в кластерах платформы:</p> <ul style="list-style-type: none"><li>• cpaas-system,</li><li>• cert-manager,</li><li>• default,</li><li>• global-credentials,</li><li>• kube-ovn,</li><li>• kube-public,</li><li>• kube-system,</li><li>• nsx-system,</li><li>• alauda-system,</li><li>• kube-federation-system,</li><li>• ALL-ALL,</li><li>• true</li></ul> |
| Cluster  | <p>Кластер(ы), связанные с проектом, в которых администратор может выделять квоты ресурсов. Нажмите на выпадающий список, чтобы выбрать один или несколько кластеров.</p> <p><b>Примечание:</b> Кластеры в аномальном состоянии выбрать нельзя</p>                                                                                                                                                                                                                                                                                                                                                                                     |

4. Нажмите **Next** и на шаге настройки квот проекта задайте квоты ресурсов, которые будут выделены текущему проекту для выбранных кластеров:
  - CPU (ядра),
  - Память (Gi),
  - Хранилище (Gi),
  - Количество PVC,
  - Pods (количество),
  - Виртуальный GPU (GPU-Manager/MPS),
  - pGPU (физический GPU, ядра),
  - Память GPU.



Квоты ресурсов GPU можно настроить только при развёрнутых GPU-плагинах в кластере. Для ресурса **GPU-Manager** или **MPS GPU** также можно настроить квоту **vMemory**.

**GPU Units:** 100 единиц виртуальных ядер эквивалентны 1 физическому ядру (1 pGPU = 1 ядро = 100 ядер GPU-Manager = 100 ядер MPS), и единицы pGPU можно выделять только целыми числами. 1 единица памяти GPU-Manager равна 256 Mi, 1 единица памяти MPS GPU равна 1 Gi, 1024 Mi = 1 Gi.

Значения квот проекта должны находиться в пределах диапазона квот, отображаемого на странице. Под каждым полем ввода квоты ресурса отображается информация о выделенной квоте и общем объеме для этого ресурса для справки.

**Внимание:** Если для какого-либо типа ресурса квота не установлена, по умолчанию она считается Неограниченной. Это означает, что проект может использовать доступные ресурсы соответствующего типа в кластере по мере необходимости без максимального ограничения.

4. Нажмите **Create Project**.

## Редактирование проекта

### Обновление основной информации о проекте

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта, который нужно обновить.
3. В левой панели выберите **Details**.
4. Нажмите **Actions > Update Basics**.
5. Отредактируйте параметры:
  - **Display name** – отображаемое имя,
  - **Description** – описание.
6. Нажмите **Update**.

### Обновление квот проекта

#### Ограничения

Если проект связан с кластером в состоянии **Abnormal**, обновление квот, назначенных проекту в этом кластере, не поддерживается.

#### Шаги

Обновите квоты ресурсов для проекта в каждом связанном кластере:

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта, который нужно обновить.
3. В левой панели выберите **Details**.
4. Нажмите **Update Quota**.
5. Обновите квоты, назначенные проекту в кластере, согласно следующим рекомендациям:

- Квоты ресурсов GPU (vGPU/pGPU) можно настраивать только при наличии GPU-ресурсов в кластере. Если GPU-ресурс является **vGPU**, также можно настроить квоты **vMemory**.

**GPU Units:** 100 единиц виртуальных ядер эквивалентны 1 физическому ядру (1 pGPU = 1 ядро = 100 vGPU); 1 единица видеопамати равна 256 Ми; единицы pGPU — целые числа и могут назначаться только целыми числами.

- Если для какого-либо ресурса квота не установлена, по умолчанию ресурс имеет неограниченную квоту,
- Значение квоты должно находиться в пределах, отображаемых на странице.

5. Нажмите **Update**.

## Управление кластером проекта

Вы можете добавлять кластеры в проект для выделения их ресурсов или удалять кластеры, чтобы вернуть выделенные ресурсы. Эта функциональность полезна в следующих случаях:

- Когда ресурсов проекта недостаточно для бизнес-операций,
- Когда необходимо выделить недавно созданный или добавленный кластер существующему проекту,
- Когда нужно вернуть ресурсы кластера из проекта,
- Когда конкретному проекту требуется эксклюзивный доступ к кластеру.

## Добавление кластера

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Details**.
4. Нажмите **Actions > Add Cluster**.
5. Выберите кластер и установите квоту ресурсов, которая будет выделена текущему проекту:
  - CPU (ядра),
  - Память (Gi),
  - Хранилище (Gi),
  - Количество PVC (число),
  - Pods (число),
  - vGPU (виртуальный GPU)/MPS/pGPU (физический GPU, ядра),
  - Квота видеопамати.

Квота ресурсов GPU может быть настроена только при разворачивании GPU-плагинов в кластере. Если GPU-ресурс является **GPU-Manager** или **MPS GPU**, также можно настроить квоту **vMemory**.

**GPU Units:** 100 единиц виртуальных ядер эквивалентны 1 физическому ядру (1 pGPU = 1 ядро = 100 ядер GPU-Manager = 100 ядер MPS), и единицы pGPU могут выделяться только

целыми числами. 1 единица памяти GPU-Manager равна 256 Mi, 1 единица памяти MPS GPU равна 1 Gi, 1024 Mi = 1 Gi.

Значение установленной квоты проекта должно находиться в пределах диапазона квот, отображаемого на странице. Под каждым полем ввода квоты ресурса отображаются выделенная квота и общий объем этого ресурса для справки.

**Внимание:** Если для какого-либо типа ресурса квота не установлена, по умолчанию она считается **Неограниченной**. Это означает, что проект может использовать доступные ресурсы соответствующего типа в кластере без максимального ограничения.

5. Нажмите **Add**.

## Удаление кластера

После удаления кластера проект не сможет использовать бизнес-ресурсы, находящиеся под управлением удалённого кластера.

Если удаляемый кластер находится в аномальном состоянии, ресурсы под этим кластером не могут быть очищены. Рекомендуется исправить состояние кластера перед его удалением.

Чтобы удалить кластер:

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Details**.
4. Нажмите **Actions > Remove Cluster**.
5. В появившемся диалоговом окне введите имя удаляемого кластера и нажмите кнопку **Remove**.

## Управление участниками проекта

### Импорт участников

Вы можете импортировать всех пользователей или выбрать конкретных пользователей из списка участников.

Вы можете предоставить пользователям права на управление проектом и его пространствами имён, импортируя существующих пользователей платформы или добавляя пользователей OIDC. Вы можете назначать роли, такие как администраторы проекта, администраторы пространств имён, разработчики или настраиваемые роли с правами управления проектом и пространствами имён.

### Ограничения

- Если в платформе не настроен OIDC IDP:
  - В качестве участников проекта можно импортировать только существующих пользователей платформы, включая:
    - OIDC пользователей, которые успешно вошли в систему,
    - Пользователей, синхронизированных через LDAP,

- Локальных пользователей,
- Пользователей, добавленных в другие проекты как OIDC пользователи (с источником, отмеченным как “-“),
- Если настроен OIDC IDP:
  - Можно добавить действительные OIDC-аккаунты, соответствующие требованиям ввода,
  - Валидность аккаунта не проверяется при добавлении,
  - Убедитесь, что аккаунт действителен, иначе вход в систему будет невозможен,
- Пользователи по умолчанию с правами системного администратора и текущий вошедший пользователь не могут быть импортированы.

#### ***Импорт всех пользователей***

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Members**.
4. Нажмите **Import Member**.
5. Выберите **Member List**.
6. Нажмите на выпадающее меню **Bind** и выберите роль, которую хотите назначить всем пользователям.

Если роль требует указания пространства имён, выберите его в выпадающем меню **Namespaces**.

7. Нажмите **Import All**.

#### ***Импорт конкретных пользователей***

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Members**.
4. Нажмите **Import Member**.
5. Выберите **Member List**.
6. Отметьте одного или нескольких пользователей с помощью флажков.

**Совет:** Используйте выпадающее меню групп пользователей в правом верхнем углу и поле поиска для фильтрации пользователей по имени.

7. Нажмите на выпадающее меню **Bind** и выберите роль, которую хотите назначить выбранным пользователям.

Если роль требует указания пространства имён, выберите его в выпадающем меню **Namespaces**.

8. Нажмите **Import**.

### **Импорт пользователей OIDC**

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Members**.
4. Нажмите **Import Member**.
5. Выберите **OIDC Users**.
6. Нажмите **Add** для создания записи участника (повторите для нескольких участников).
7. Введите имя пользователя, прошедшего аутентификацию через OIDC, в поле **Name**.

**Внимание:** Убедитесь, что имя пользователя соответствует аккаунту, который может быть аутентифицирован настроенной системой OIDC, иначе вход будет невозможен.

8. Выберите роль из выпадающего меню **Roles**.

Если роль требует указания пространства имён, выберите его в выпадающем меню **Namespaces**.

9. Нажмите **Import**. После успешного импорта:
  - Пользователь появится в списке участников проекта,
  - Пользователь появится в разделе **Platform Management > Users**:
    - Источник отображается как "-" до первого входа/синхронизации,
    - Источник обновляется после успешного входа/синхронизации.

### **Удаление участников**

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Members**.
3. Нажмите **Remove** рядом с участником, которого хотите удалить.

**Совет:** Используйте выпадающий список рядом с полем поиска для фильтрации участников по **Name**, **Display name** или **User Group**.

4. Подтвердите удаление.

### **Удаление проекта**

**Внимание:** После удаления проекта ресурсы, занятые проектом в кластере, будут освобождены.

Удалите проекты, которые больше не используются:

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. В левой панели выберите **Details**.
4. Нажмите **Actions > Delete Project**.
5. Введите имя проекта и нажмите **Delete**.



## Аудит

### Обзор

Функциональность аудита платформы предоставляет упорядоченные по времени записи операций, связанных с пользователями и безопасностью системы. Это помогает анализировать конкретные проблемы и быстро решать вопросы, возникающие в кластерах, пользовательских приложениях и других областях.

С помощью аудита вы можете отслеживать различные изменения в Kubernetes кластере, включая:

- Какие изменения произошли в кластере за определённый период времени
- Кто выполнил эти изменения (системные компоненты или пользователи)
- Детали важных событий изменений (например, обновления параметров POD)
- Результаты событий (успех или неудача)
- Местоположение оператора (внутри или вне кластера)
- Записи операций пользователей (обновления, удаления, управленческие операции) и их результаты

### Поиск и просмотр событий платформы

#### Прerequisites

Ваша учетная запись должна иметь права управления платформой или права аудита платформы.

#### Шаги

1. Перейдите в **Platform Management**
2. В левой панели нажмите **Auditing**.
3. Выберите область аудита:
  - **User Operations**: Просмотр записей операций пользователей, вошедших в платформу,
  - **System Operations**: Просмотр записей системных операций (операторы начинаются с “system:”).
4. Настройте условия запроса для фильтрации событий платформы:

| Условие запроса | Описание                                                                                              |
|-----------------|-------------------------------------------------------------------------------------------------------|
| Operated at     | Диапазон времени для поиска событий аудита. Можно выбрать предустановленный диапазон или задать свой. |
| Operator        | Имя пользователя или системной учётной записи оператора (по умолчанию: All).                          |
| Actions         | Тип операции (create, update, delete, manage, rollback, stop и др., по умолчанию: All).               |

| Условие запроса | Описание                                                   |
|-----------------|------------------------------------------------------------|
| Clusters        | Кластер, содержащий управляемый ресурс (по умолчанию: All) |
| Resource Type   | Тип управляемого ресурса (по умолчанию: All)               |
| Resource Name   | Имя управляемого ресурса (поддерживается нечеткий поиск)   |

5. Нажмите **Search**.

Нажмите на иконку обновления, чтобы обновить результаты поиска.

Нажмите на иконку экспорта, чтобы скачать результаты в формате .csv.

## Результаты поиска

В результатах поиска отображается следующая информация:

| Параметр           | Описание                                                                         |
|--------------------|----------------------------------------------------------------------------------|
| Operator           | Имя пользователя или системной учётной записи оператора                          |
| Action             | Тип операции (create, update, delete, manage, rollback, stop и др.)              |
| Resource Name/Type | Имя и тип управляемого ресурса                                                   |
| Clusters           | Кластер, содержащий управляемый ресурс                                           |
| Namespaces         | Пространство имён, содержащее управляемый ресурс                                 |
| Client IP          | IP-адрес клиента, с которого была выполнена операция                             |
| Operation Result   | Результат операции на основе кода возврата API (2xx = успех, остальные = ошибка) |
| Operated At        | Время операции                                                                   |

Нажмите кнопку **Details**, чтобы просмотреть детали события в формате JSON.



# Инструменты самообслуживания для разработчиков

## Обзор

Модуль Developer view предоставляет разработчикам возможности оркестрации и эксплуатации облачно-нативных приложений. Он обеспечивает единый интерфейс для компоновки приложений из различных источников с интеграцией встроенных инструментов наблюдаемости для эксплуатации в релизных средах.

MWS Container Platform предоставляет единый интерфейс для создания, редактирования, удаления и управления облачными нативными приложениями как через веб-консоль, так и через CLI (Command-Line Interface). Приложения могут быть развёрнуты в нескольких пространствах имён с использованием политик RBAC.

## Описание единиц ресурсов

- CPU: Допустимые единицы: core, m (милликор). Где 1 core = 1000 m.
- Память: Допустимые единицы: Mi (1 MiB = 2<sup>20</sup> байт), Gi (1 GiB = 2<sup>30</sup> байт). Где 1 Gi = 1024 Mi.
- Виртуальный GPU (опционально): Этот параметр действует только при наличии GPU-ресурсов в кластере. Количество виртуальных ядер GPU; 100 виртуальных ядер равны 1 физическому ядру GPU. Поддерживаются положительные целые числа.
- Видеопамять (опционально): Этот параметр действует только при наличии GPU-ресурсов в кластере. Видеопамять виртуального GPU; 1 единица видеопамяти равна 256 Mi. Поддерживаются положительные целые числа.

## Типы приложений

В разделе платформы **Container Platform > Application Management** можно создавать следующие типы приложений:

- **Application:** Полноценное бизнес-приложение, состоящее из одного или нескольких связанных вычислительных компонентов (Workloads), внутренних маршрутов (Services) и других нативных ресурсов Kubernetes. Поддерживает создание через редактирование в UI, YAML-оркестрацию и шаблоны, может работать в средах разработки, тестирования или производства. Различные типы нативных приложений можно создавать следующими способами:
  - **Create from Image:** Быстрое создание приложений с использованием существующих контейнерных образов.
  - **Create from Catalog:** Создание приложений с использованием пакетов Helm Chart.
  - **Create from YAML:** Создание приложений с использованием YAML-конфигурационных файлов.
  - **Create from Code:** Создание приложений с использованием исходного кода.

- **Operator Backed App:** На основе компонентов приложения (Operator backed) можно быстро развернуть компонентное приложение и использовать возможности Operators для автоматизации всего жизненного цикла управления приложением.
- **OAM Application:** Используется для определения модели облачно-нативных приложений. По сравнению с логикой оркестрации контейнеров или Kubernetes, OAM больше фокусируется на самом приложении. На основе OAM общие возможности приложений инкапсулируются в высокоуровневые интерфейсы для использования на протяжении всего процесса развёртывания, разработки и эксплуатации приложений.

### Типы рабочих нагрузок

Помимо создания нативных приложений и компонентных приложений, рабочие нагрузки также можно создавать напрямую в **Container Platform > Computing Components**:

- **Deployment:** Наиболее часто используемый контроллер рабочих нагрузок для развёртывания приложений, не хранящих свое состояние. Он обеспечивает запуск заданного количества реплик Pod в кластере, поддерживает rolling update и откаты, подходит для сценариев приложений, не хранящих свое состояние, таких как веб-сервисы и API-сервисы.
- **DaemonSet:** Обеспечивает запуск реплики Pod на каждом узле кластера (или на определённых узлах). При добавлении узла в кластер Pod автоматически создаётся; при удалении узла из кластера соответствующие Pods также удаляются. Подходит для сценариев, требующих запуска логирования, мониторинга и т.п. на каждом узле.
- **StatefulSet:** Контроллер рабочих нагрузок для управления состоянием приложений. Он сохраняет фиксированную идентичность для каждого Pod и обеспечивает стабильное хранилище и сетевую идентичность, которые не меняются даже при пересоздании Pod. Подходит для приложений, хранящих свое состояние, таких как базы данных и распределённые кэши.
- **Job:** Рабочая нагрузка для выполнения одноразовых задач. Job создаёт один или несколько Pods и гарантирует, что заданное количество Pods успешно завершит задачу до того, как поды будут удалены. Подходит для пакетной обработки, миграции данных и других одноразовых задач.
- **CronJob:** Используется для управления Jobs, запланированными на выполнение по расписанию. Можно задать временное выражение для запуска задачи, и система автоматически создаст и выполнит Job в назначенное время. Подходит для периодических задач, таких как резервное копирование данных, генерация отчётов и периодическая очистка.

Помимо создания вышеуказанных вычислительных компонентов через интерфейс платформы, платформа также поддерживает создание Pods и Containers с помощью CLI-инструментов:

- **Pod:** Наименьшая единица развёртывания в Kubernetes, Pod может содержать один или несколько контейнеров, которые совместно используют хранилище, сеть и

конфигурационные декларации. Pods обычно управляются контроллерами (например, Deployments).

- **Container:** Стандартная программная единица, которая упаковывает код и все зависимости, позволяя приложениям быстро и надёжно запускаться в различных вычислительных средах. Контейнеры работают внутри Pods и используют ресурсы Pod.

### Управление пространствами имён

Пространства имён обеспечивают логическую изоляцию ресурсов Kubernetes. Основные операции включают:

- Создание пространств имён: Определение квот ресурсов и политик допуска безопасности подов.
- Импорт пространств имён: Импорт существующих пространств имён Kubernetes в MWS Container Platform обеспечивает полное соответствие функциональности платформы с нативно созданными пространствами имён.

### Управление жизненным циклом приложений

MWS Container Platform поддерживает сквозное управление жизненным циклом, включая:

- **Шаблоны создания приложений**

В MWS Container Platform приложения можно создавать несколькими способами. Вот некоторые распространённые методы:

- Создание из образов: Создание пользовательских приложений с использованием заранее собранных контейнерных образов. Этот метод поддерживает создание полного приложения, включающего Deployments, Services, ConfigMaps и другие ресурсы Kubernetes.
- Создание из каталога: MWS Container Platform предоставляет каталоги приложений, позволяя пользователям выбирать предопределённые шаблоны приложений (Helm Charts или Operator Backed) для создания.
- Создание из YAML: Импортируя YAML-файл, создайте пользовательское приложение со всеми включёнными ресурсами за один шаг.
- Создание из кода: Сборка образов с помощью Source to Image (S2I).
- **Операции с приложениями**
  - Обновление приложений: Обновление версии образа приложения, переменных окружения и других конфигураций, либо импорт существующих ресурсов Kubernetes для централизованного управления.
  - Экспорт приложений: Экспорт приложений в форматах YAML, Kustomize или Helm Chart с последующим импортом для создания новых экземпляров приложений в других пространствах имён или кластерах.

- Управление версиями: Поддержка автоматического или ручного создания версий приложений, а в случае проблем доступен однокликовый откат к определённой версии для быстрого восстановления.
- Удаление приложений: Удаление приложения одновременно удаляет само приложение и все напрямую входящие в него ресурсы Kubernetes. Кроме того, эта операция разрывает любые связи приложения с другими ресурсами Kubernetes, которые не были напрямую частью его определения.

- **Наблюдаемость приложений**

Для непрерывного управления эксплуатацией платформа предоставляет логи, события, мониторинг и др.

- Логи: Поддержка просмотра логов в реальном времени из текущего запущенного Pod, а также логов с предыдущих перезапусков контейнеров.
- События: Поддержка просмотра информации о событиях для всех ресурсов в пространстве имён.
- Панели мониторинга: Предоставление панелей мониторинга на уровне пространства имён, включая отдельные представления для приложений, Workloads и Pods, а также поддержку настройки панелей мониторинга под конкретные операционные требования.

## **Управление рабочими нагрузками Kubernetes**

Поддержка основных типов рабочих нагрузок:

- Deployments: Управление приложениями, не хранящих состояние, с поддержкой rolling updates.
- StatefulSets: Запуск stateful-приложений со стабильными сетевыми идентификаторами.
- DaemonSets: Развёртывание сервисов на уровне узлов (например, сборщиков логов).
- CronJobs: Планирование пакетных заданий с политиками повторных попыток.

## **Пространства имён**

В Kubernetes пространства имён обеспечивают механизм изоляции групп ресурсов внутри одного кластера. Имена ресурсов должны быть уникальными внутри пространства имён, но не обязательно уникальными между разными пространствами имён. Область действия, основанная на пространстве имён, применима только к объектам с пространством имён (например, Deployments, Services и т.д.), но не к объектам, охватывающим весь кластер (например, StorageClass, Nodes, PersistentVolumes и т.д.).

Подробная информация о пространствах имён приведена в официальной документации Kubernetes.

## **Создание пространств имён с помощью веб-консоли**

Внутри кластера, связанного с проектом, создайте новое пространство имён, соответствующее доступным квотам ресурсов проекта. Новое пространство имён работает

в рамках квот ресурсов, выделенных проекту (например, CPU, память), и все ресурсы в пространстве имён должны находиться в связанном кластере.

Чтобы создать пространство имён:

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта, для которого хотите создать пространство имён.
3. В левой панели выберите **Namespaces > Namespaces**.
4. Нажмите **Create Namespace**.
5. Настройте **Basic Information**:

| Параметр  | Описание                                                                            |
|-----------|-------------------------------------------------------------------------------------|
| Cluster   | Выберите кластер, связанный с проектом, в котором будет размещено пространство имён |
| Namespace | Имя пространства имён должно содержать обязательный префикс — имя проекта           |

6. (Опционально) Настройте квоты ресурсов.

Каждый раз, когда для контейнера в пространстве имён задаётся ограничение ресурсов (limits) по вычислительным ресурсам или ресурсам хранения, или при добавлении нового Pod или PVC, это будет расходовать квоту, установленную здесь.

**Поведение квоты по умолчанию:** если для типа ресурса квота не указана, по умолчанию она не ограничена. Это означает, что пространство имён может использовать все доступные ресурсы данного типа, выделенные проекту, без явных ограничений.

**Внимание:** Квота ресурсов пространства имён наследуется от выделенной проекту квоты в кластере. Максимально допустимая квота для типа ресурса не может превышать оставшуюся доступную квоту проекта. Если доступная квота по какому-либо ресурсу достигает 0, создание пространства имён будет заблокировано. Обратитесь к администратору платформы для корректировки квот.

Требования к настройке квоты GPU:

- Квоты GPU (vGPU или pGPU) можно настроить только при наличии GPU-ресурсов в кластере.
- При использовании vGPU также можно задавать квоты по памяти.

Определение единиц GPU:

- **vGPU Units:** 100 виртуальных GPU (vGPU) = 1 физическое ядро GPU (pGPU).

**Примечание:** pGPU учитываются только целыми числами (например, 1 pGPU = 1 ядро = 100 vGPU).

- Единицы памяти:
  - 1 единица памяти = 256 MiB.
  - 1 GiB = 4 единицы памяти (1024 MiB = 4 × 256 MiB).

Параметры квот:

| Категория               | Тип квоты                                         | Значение и единица | Описание                                                                                                                                                                                                                                                                                   |
|-------------------------|---------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Квота ресурсов хранения | Все                                               | Gi                 | Общая запрашиваемая ёмкость хранения всех Persistent Volume Claims (PVC) в этом пространстве имён не может превышать это значение                                                                                                                                                          |
|                         | Storage Class                                     |                    | Общая запрашиваемая ёмкость хранения всех Persistent Volume Claims (PVC), связанных с выбранным StorageClass в этом пространстве имён, не может превышать это значение.<br><b>Примечание:</b> Заранее выделите StorageClass проекту, к которому принадлежит пространство имён              |
| Расширенные ресурсы     | Получены из конфигурационного словаря (ConfigMap) | -                  | Эта категория не отображается, если отсутствует соответствующий конфигурационный словарь                                                                                                                                                                                                   |
| Другие квоты            | Введите пользовательские квоты                    | -                  | Для предотвращения проблем с дублированием ресурсов следующие значения не допускаются в качестве типов квот: <ul style="list-style-type: none"> <li>limits.cpu</li> <li>limits.memory</li> <li>requests.cpu</li> <li>requests.memory</li> <li>Pods</li> <li>cpu</li> <li>memory</li> </ul> |

7. (Опционально) Настройте **Container Limit Range**.

8. (Опционально) Настройте **Pod Security Admission**.

9. (Опционально) В разделе **More Configuration** добавьте метки и аннотации для текущего пространства имён.

**Совет:** Вы можете определить атрибуты пространства имён с помощью меток или дополнить пространство имён дополнительной информацией через аннотации; оба способа можно использовать для фильтрации и сортировки пространств имён.

10. Нажмите **Create**.

## Создание пространства имён с помощью CLI

### Примеры YAML-файлов

#### example-namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
 name: example
 labels:
 pod-security.kubernetes.io/audit: baseline # Option, to ensure security, it is
recommended to choose the baseline or restricted mode.
 pod-security.kubernetes.io/enforce: baseline
 pod-security.kubernetes.io/warn: baseline
```

#### example-resourcequota.yaml

```
apiVersion: v1
kind: ResourceQuota
metadata:
 name: example-resourcequota
 namespace: example
spec:
 hard:
 limits.cpu: '20'
 limits.memory: 20Gi
 pods: '500'
 requests.cpu: '2'
 requests.memory: 2Gi
```

#### example-limitrage.yaml

```
apiVersion: v1
kind: LimitRange
metadata:
 name: example-limitrage
 namespace: example
spec:
 limits:
```

```
- default:
 cpu: 100m
 memory: 100Mi
defaultRequest:
 cpu: 50m
 memory: 50Mi
max:
 cpu: 1000m
 memory: 1000Mi
type: Container
```

### ***Создание через YAML-файлы***

```
kubectl apply -f example-namespace.yaml
kubectl apply -f example-resourcequota.yaml
kubectl apply -f example-limitrage.yaml
```

### ***Создание напрямую через командную строку***

```
kubectl create namespace example
kubectl create resourcequota example-resourcequota --namespace=example --
hard=limits.cpu=20,limits.memory=20Gi,pods=500
kubectl create limitrage example-limitrage --namespace=example --
default='cpu=100m,memory=100Mi' --default-request='cpu=50m,memory=50Mi' --
max='cpu=1000m,memory=1000Mi'
```

### **Коэффициент Overcommit**

MWS Container Platform позволяет задавать коэффициент overcommit ресурсов (CPU и памяти) для каждого пространства имён. Это управляет соотношением между лимитами контейнеров (максимальное использование) и запросами (гарантированный минимум) внутри данного namespace, оптимизируя использование ресурсов.

Настраивая этот коэффициент, вы обеспечиваете, что заданные пользователем лимиты и запросы контейнеров остаются в разумных пределах, улучшая общую эффективность использования ресурсов кластера.

### ***Ключевые понятия***

- **Лимиты:** Максимальное количество ресурса, которое контейнер может использовать. Превышение лимитов может привести к троттлингу (CPU) или завершению процесса (память).
- **Запросы:** Гарантированный минимум ресурса, необходимый контейнеру. Kubernetes планирует контейнеры, исходя из этих запросов.
- **Коэффициент Overcommit:** Лимиты / Запросы. Эта настройка определяет допустимый диапазон этого коэффициента в namespace, балансируя гарантии ресурсов и предотвращая их чрезмерное потребление.



## Основные возможности

- Повышение плотности использования ресурсов и стабильности приложений в namespace за счёт установки подходящего коэффициента overcommit для управления балансом между лимитами и запросами ресурсов.

### Пример

Предположим, что коэффициент overcommit для namespace установлен в 2. При создании приложения и указании лимита CPU в 4с, соответствующее значение запроса CPU рассчитывается как:

$\text{CPU Request} = \text{CPU Limit} / \text{Overcommit ratio}.$

Таким образом, запрос CPU будет  $4с / 2 = 2с$ .

### Определение CRD

```
example-namespace-overcommit.yaml
apiVersion: resource.alauda.io/v1
kind: NamespaceResourceRatio
metadata:
 namespace: example
 name: example-namespace-overcommit
spec:
 cpu: 3 # Отсутствие этого поля означает наследование коэффициента overcommit
кластера; 0 — отсутствие ограничения.
 memory: 4 # Отсутствие этого поля означает наследование коэффициента
overcommit кластера; 0 — отсутствие ограничения.
status:
 clusterCPU: 2 # Коэффициент overcommit кластера
 clusterMemory: 3
```

### Создание коэффициента overcommit с помощью CLI

```
kubectl apply -f example-namespace-overcommit.yaml
```

### Создание/обновление коэффициента overcommit через веб-консоль

Позволяет настроить коэффициент overcommit для namespace, чтобы управлять соотношением между лимитами и запросами ресурсов. Это гарантирует, что выделение ресурсов контейнерам остаётся в заданных пределах, улучшая использование ресурсов кластера.

### Меры предосторожности

Если в кластере используется виртуализация узлов (например, виртуальные узлы), перед настройкой oversubscription для виртуальных машин необходимо отключить oversubscription на уровне кластера/namespace.

### Шаги

1. Перейдите в **Project Management**.

2. Нажмите на имя проекта.
3. В левой панели перейдите в **Namespaces**.
4. Нажмите на имя пространства имён.
5. Нажмите **Actions > Update Overcommit**.
6. Выберите подходящий **способ настройки** коэффициента overcommit для CPU или памяти в namespace.

| Параметр             | Описание                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Inherit from Cluster | <ul style="list-style-type: none"> <li>• Namespace наследует коэффициент oversubscription кластера.</li> </ul> <p>Пример: если коэффициент CPU/памяти кластера равен 4, namespace по умолчанию будет 4.</p> <ul style="list-style-type: none"> <li>• Запросы контейнера = лимит/коэффициент кластера.</li> <li>• Если лимит не задан, используется квота контейнера по умолчанию для namespace.</li> </ul> |
| Custom               | <ul style="list-style-type: none"> <li>• Установить коэффициент, специфичный для namespace (целое число &gt; 1).</li> </ul> <p>Пример: коэффициент кластера = 4, коэффициент namespace = 2 → запросы = лимит / 2.</p> <ul style="list-style-type: none"> <li>• Оставьте пустым, чтобы отключить oversubscription для namespace</li> </ul>                                                                  |

5. Нажмите **Update**.

**Примечание:** Изменения применяются только к вновь создаваемым Pod. Существующие Pod сохраняют свои исходные запросы до пересоздания.

## Управление участниками пространства имён

### Импорт участников

Платформа поддерживает массовый импорт участников в пространство имён с назначением ролей, таких как Namespace Administrator или Developer, для предоставления соответствующих прав.

### Ограничения и условия

- Участников можно импортировать в пространство имён только из **Project Members** проекта, к которому принадлежит пространство имён.
- Платформа не поддерживает импорт системных администраторов по умолчанию или активного пользователя.

### Прerequisites

Для импорта пользователей в качестве участников пространства имён они должны быть предварительно добавлены в проект пространства имён.

### Шаги

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. Перейдите в **Namespaces**.
4. Нажмите на имя пространства имён, в которое нужно импортировать участников.
5. Во вкладке Namespace Members нажмите Import Members.
6. Импортируйте всех или некоторых пользователей из списка в пространство имён:
  - Импортировать всех пользователей из списка в качестве участников пространства имён и массово назначить им роли:
    1. Нажмите на выпадающий список справа от пункта **Set Role** внизу диалога и выберите роль для назначения.
    2. Нажмите Import All.
  - Импортировать одного или нескольких пользователей из списка в качестве участников пространства имён.
    1. Отметьте чекбокс перед именем пользователя/отображаемым именем для выбора одного или нескольких пользователей.
    2. Нажмите на выпадающий список справа от пункта **Set Role** внизу диалога и выберите роль для выбранных пользователей.
    3. Нажмите **Import**.

**Совет:** Вы можете выбрать группу пользователей с помощью выпадающего списка в правом верхнем углу диалога и выполнить поиск, введя имя пользователя в поле поиска по имени.

### ***Добавление участников***

Если на платформе добавлен IDP типа OIDC, пользователей OIDC можно добавлять в качестве участников пространства имён.

Вы можете добавить действительные OIDC-аккаунты, соответствующие требованиям ввода, в роли пространства имён и назначить пользователю соответствующие роли.

**Примечание:** При добавлении участников система не проверяет действительность аккаунтов. Пожалуйста, убедитесь, что добавляемые аккаунты действительны, иначе эти аккаунты не смогут успешно войти на платформу.

**Действительные OIDC-аккаунты:** действительные аккаунты в службе аутентификации OIDC, настроенной через IDP для платформы, включая как успешно вошедших на платформу, так и не вошедших.

### **Пререквизиты**

На платформе добавлен IDP типа **OIDC**.

### **Шаги**

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. Перейдите в **Namespaces**.

4. Нажмите на имя пространства имён, в которое нужно добавить участника.
5. Во вкладке **Namespace Members** нажмите **Add Member**.
6. В поле ввода **Username** введите имя пользователя сторонней платформы, поддерживаемой текущей платформой.

**Примечание:** Убедитесь, что введённое имя пользователя соответствует существующему аккаунту на сторонней платформе, иначе этот аккаунт не сможет успешно войти на эту платформу.

7. В выпадающем списке **Role** выберите роль, которую нужно назначить этому пользователю.
8. Нажмите **Add**. После успешного добавления участник появится в списке участников пространства имён.

Одновременно в списке пользователей (**Platform Management > Users**) вы сможете увидеть этого пользователя. До того, как пользователь успешно войдёт или будет синхронизирован с платформой, источник будет “-“, и его можно удалить; после успешного входа или синхронизации платформа зафиксирует источник аккаунта и отобразит его в списке пользователей.

### **Удаление участников**

Удалите участников пространства имён и их связанные роли для отзыва прав в пространстве имён:

1. Перейдите в **Project Management**.
2. Нажмите на имя проекта.
3. Перейдите в **Namespaces**.
4. Нажмите на имя пространства имён, из которого нужно удалить участника.
5. Во вкладке **Namespace Members** в меню участника выберите **Remove**.
6. Нажмите **Remove**.

### **Настройка Secrets**

В Kubernetes (k8s) Secret — это базовый объект, предназначенный для хранения и управления конфиденциальной информацией, такой как пароли, OAuth-токены, SSH-ключи, TLS-сертификаты и API-ключи. Его основная задача — предотвратить прямое внедрение чувствительных данных в определения Pod или образы контейнеров, что повышает безопасность и портативность.

Secrets похожи на ConfigMaps, но предназначены специально для конфиденциальных данных. Обычно они хранятся в виде base64-кодированных значений и могут использоваться подами различными способами, включая монтирование в виде томов или предоставление в виде переменных окружения.

### **Создание Secret типа Opaque**

```
kubectcl create secret generic my-secret \
--from-literal=username=admin \
```

```
--from-literal=password=Pa$$w0rd
```

#### YAML

```
apiVersion: v1
kind: Secret
metadata:
 name: my-secret
type: Opaque
data:
 username: YWRtaW4= # base64 от "admin"
 password: UGEkJHcwcmQ= # base64 от "Pa$$w0rd"
```

Вы можете декодировать их так:

```
echo YWRtaW4= | base64 --decode # вывод: admin
```

#### Создание Secret типа Docker registry

```
kubectl create secret docker-registry my-docker-creds \
--docker-username=myuser \
--docker-password=mypass \
--docker-server=https://index.docker.io/v1/ \
--docker-email=my@example.com
```

#### YAML

```
apiVersion: v1
kind: Secret
metadata:
 name: my-docker-creds
type: kubernetes.io/dockerconfigjson
data:
 .dockerconfigjson:
 eyJhdXRob2l6eyJodHRwczovL2luZGV4LmRvY2tldi5pby92MS8iOmsidXNlcm5hbWUiOiJteX
 VzZXliLCJwYXNzd29yZCI6Im15cGFzcyIsImVtYWlsIjoibXlAZXhhbXBsZS5jb20iLCJhdXRoljoiYl
 hsMWMMyVnlpbTE1Y0dGemN3PT0ifX19
```

K8s автоматически преобразует ваше имя пользователя, пароль, email и информацию о сервере в стандартный формат входа Docker:

```
{
 "auths": {
 "https://index.docker.io/v1/": {
 "username": "myuser",
 "password": "mypass",
 "email": "my@example.com",
 "auth": "bXl1c2VyOm15cGFzcmw==" # base64(username:password)
 }
 }
}
```

```
}
```

Этот JSON затем кодируется в base64 и используется в поле data объекта Secret.

Используйте его в Pod:

```
imagePullSecrets:
 - name: my-docker-creds
```

### **Создание Secret типа Basic Auth**

```
apiVersion: v1
kind: Secret
metadata:
 name: basic-auth-secret
type: kubernetes.io/basic-auth
stringData:
 username: myuser
 password: mypass
```

### **Создание Secret типа SSH-Auth**

Сценарий использования: хранение приватных SSH-ключей (например, для доступа к Git).

```
apiVersion: v1
kind: Secret
metadata:
 name: ssh-key-secret
type: kubernetes.io/ssh-auth
stringData:
 ssh-privatekey: |
 -----BEGIN OPENSSH PRIVATE KEY-----
 ...
 -----END OPENSSH PRIVATE KEY-----
```

### **Создание Secret типа TLS**

Сценарий использования: TLS-сертификаты (используются Ingress, webhook и др.)

```
kubectl create secret tls tls-secret \
--cert=path/to/tls.crt \
--key=path/to/tls.key
```

### **YAML**

```
apiVersion: v1
kind: Secret
metadata:
 name: tls-secret
type: kubernetes.io/tls
data:
```

```
tls.crt: <base64>
```

```
tls.key: <base64>
```

### Создание Secret через веб-консоль

1. Перейдите в **Container Platform**.
2. В левой выберите **Configuration > Secrets**.
3. Нажмите **Create Secret**.
4. Настройте параметры.

**Примечание:** В форме ввода конфиденциальные данные, такие как имя пользователя и пароль, автоматически кодируются в формат Base64 перед сохранением в Secret. Преобразованные данные можно просмотреть в YAML-виде.

5. Нажмите **Create**.

### Как использовать Secret в Pod

При создании workloads для нативных приложений в том же namespace можно ссылаться на уже созданные Secrets.

- **В виде переменных окружения**

```
env:
```

```
- name: DB_USERNAME
```

```
valueFrom:
```

```
secretKeyRef:
```

```
name: my-secret
```

```
key: username
```

Из секрета с именем my-secret берется значение по ключу username и присваивается переменной окружения DB\_USERNAME.

- **В виде смонтированных файлов (тома)**

```
volumes:
```

```
- name: secret-volume
```

```
secret:
```

```
secretName: my-secret
```

```
volumeMounts:
```

```
- name: secret-volume
```

```
mountPath: "/etc/secret"
```

### Операции с секретами

В разделе **Configuration > Secrets** нажмите на меню в строке секрета и выберите нужное действие или нажмите на имя секрета и выберите нужное действие в **Actions**:

| Операция | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Обновить | После добавления или обновления Secret, workloads, которые ссылались на этот Secret (или его элементы конфигурации) через переменные окружения, необходимо пересобрать Pods, чтобы новая конфигурация вступила в силу                                                                                                                                                                                                                                                                                                                             |
| Удалить  | <p>После удаления Secret workloads, которые ссылались на этот Secret (или его элементы конфигурации) через переменные окружения, могут столкнуться с проблемами из-за отсутствия источника ссылки при пересборке Pods.</p> <p><b>Внимание:</b> Не удаляйте Secrets, автоматически созданные платформой, так как это может привести к некорректной работе платформы.</p> <p>Например: Secrets типа service-account-token, содержащие информацию аутентификации для ресурсов namespace, и Secrets в системных namespace (таких как kube-system)</p> |

## Создание приложений

### Создание приложений из образа

#### Шаг 1. Настройка рабочей нагрузки

1. Перейдите в **Container Platform**.
2. В левой панели перейдите в **Applications > Applications**.
3. Нажмите **Create**.
4. Выберите способ создания **Create from Image**.
5. **Выберите** или **введите** образ и нажмите **Confirm**.

**Примечание:** При использовании образов из репозитория, интегрированного в веб-консоль, можно фильтровать образы по **Already Integrated**. Название интеграционного проекта, например, образы (docker-registry-projectname), включает имя проекта projectname в этой веб-консоли и имя проекта containers в репозитории образов.

6. В разделе **Workload > Basic Info** настройте декларативные параметры для workloads:

| Параметр               | Описание                                                                                                                                |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Model                  | <p>Выберите workload:</p> <ul style="list-style-type: none"> <li>• Deployment,</li> <li>• DaemonSet ,</li> <li>• StatefulSet</li> </ul> |
| Replicas               | Определяет желаемое количество реплик Pod в Deployment (по умолчанию: 1). Настраивается в зависимости от требований workload            |
| More > Update Strategy | Настраивает стратегию rollingUpdate для обновлений без простоя:                                                                         |



| Параметр | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <ul style="list-style-type: none"> <li>• <b>Max surge (maxSurge):</b> <ul style="list-style-type: none"> <li>○ Максимальное число Pod, превышающее желаемое количество реплик во время обновления.</li> <li>○ Принимает абсолютные значения (например, 2) или проценты (например, 20%).</li> <li>○ Расчёт процентов: <math>\text{ceil}(\text{current\_replicas} \times \text{percentage})</math>.</li> </ul> </li> </ul> <p>Пример: 4.1 → 5 при расчёте от 10 реплик.</p> <ul style="list-style-type: none"> <li>• <b>Max unavailable (maxUnavailable):</b> <ul style="list-style-type: none"> <li>○ Максимальное число Pod, которые могут быть временно недоступны во время обновления.</li> <li>○ Процентные значения не могут превышать 100%.</li> <li>○ Расчёт процентов: <math>\text{floor}(\text{current\_replicas} \times \text{percentage})</math>.</li> </ul> </li> </ul> <p>Пример: 4.9 → 4 при расчёте от 10 реплик.</p> <p>Примечания:</p> <ol style="list-style-type: none"> <li>1. Значения по умолчанию: <code>maxSurge=1</code>, <code>maxUnavailable=1</code>, если явно не заданы.</li> <li>2. Неработающие Pod (например, в состояниях <code>Pending/CrashLoopBackOff</code>) считаются недоступными.</li> <li>3. Одновременные ограничения: <ul style="list-style-type: none"> <li>• <code>maxSurge</code> и <code>maxUnavailable</code> не могут одновременно быть 0 или 0 %.</li> <li>• Если процентные значения для обоих параметров равны 0, Kubernetes принудительно устанавливает <code>maxUnavailable=1</code> для обеспечения прогресса обновления.</li> </ul> </li> </ol> <p>Пример:</p> <p>Для Deployment с 10 репликами:</p> <ul style="list-style-type: none"> <li>• <code>maxSurge=2</code> → Общее число Pod во время обновления: <math>10 + 2 = 12</math>.</li> <li>• <code>maxUnavailable=3</code> → Минимальное число доступных Pod: <math>10 - 3 = 7</math>.</li> </ul> <p>Это обеспечивает доступность при контролируемом развёртывании</p> |

7. В разделе **Pod** настройте параметры контейнеров:

| Параметр                  | Описание                                                                                                                                                                                                                                                                               |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Volumes                   | Монтирование персистентных томов в контейнеры. Поддерживаемые типы томов: PVC, ConfigMap, Secret, emptyDir, hostPath и др.                                                                                                                                                             |
| Image Credential          | Требуется только при подтягивании образов из сторонних регистров (через ручной ввод URL образа).<br><b>Примечание:</b> Образы из интегрированного реестра платформы автоматически наследуют связанные секреты                                                                          |
| More > Close Grace Period | Время (по умолчанию: 30s), отведённое Pod для корректного завершения работы после получения сигнала завершения.<br>В этот период Pod завершает текущие запросы и освобождает ресурсы.<br>Установка 0 приводит к немедленному удалению (SIGKILL), что может вызвать прерывание запросов |

#### 8. Настройте **Node Affinity Rules**:

| Параметр             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| More > Node Selector | Ограничивает планирование Pod на узлы с определёнными метками (например, kubernetes.io/os: linux)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| More > Affinity      | <p>Определяет тонкие правила планирования на основе существующих Pod.</p> <p>Типы Pod Affinity:</p> <ul style="list-style-type: none"> <li>Inter-Pod Affinity: Планирует новые Pod на узлы, где размещены определённые Pod (одинаковый топологический домен).</li> <li>Inter-Pod Anti-affinity: Запрещает совместное размещение новых Pod с определёнными Pod.</li> </ul> <p>Режимы применения:</p> <ul style="list-style-type: none"> <li>RequiredDuringSchedulingIgnoredDuringExecution: Pod планируются только при выполнении правил.</li> <li>PreferredDuringSchedulingIgnoredDuringExecution: Предпочтение узлам, удовлетворяющим правилам, но допускаются исключения.</li> </ul> <p>Поля конфигурации:</p> |

| Параметр | Описание                                                                                                                                                                                                                          |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <ul style="list-style-type: none"> <li>• topologyKey: Метка узла, определяющая топологические домены (по умолчанию: kubernetes.io/hostname).</li> <li>• labelSelector: Фильтр целевых Pod с помощью запросов по меткам</li> </ul> |

9. Задайте настройки сети:

- Kube-OVN:

| Параметр          | Описание                                                                                                                                                                                                                                                                             |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bandwidth Limits  | <p>Обеспечивает QoS для сетевого трафика Pod:</p> <ul style="list-style-type: none"> <li>• Ограничение исходящего трафика: Максимальная скорость исходящего трафика (например, 10Mbps).</li> <li>• Ограничение входящего трафика: Максимальная скорость входящего трафика</li> </ul> |
| Subnet            | Назначает IP из предопределённого пула подсетей. Если не указано, используется подсеть по умолчанию для namespace                                                                                                                                                                    |
| Static IP Address | <p>Привязывает постоянные IP-адреса к Pod:</p> <p>Несколько Pod в разных Deployment могут претендовать на один IP, но одновременно использовать его может только один Pod.</p> <p><b>Внимание:</b> Количество статических IP должно быть <math>\geq</math> числу реплик Pod</p>      |

- Calico:

| Параметр          | Описание                                                                                                                                                                                                                                     |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Static IP Address | <p>Назначает фиксированные IP с жёстким ограничением уникальности:</p> <p>Каждый IP может быть привязан только к одному Pod в кластере.</p> <p><b>Внимание:</b> Количество статических IP должно быть <math>\geq</math> числу реплик Pod</p> |

10. В разделе **Container** настройте параметры контейнеров:

| Параметр                   | Описание                                                                                                                                                                                                                                                    |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resource Requests & Limits | <ul style="list-style-type: none"> <li>• Requests: Минимальные CPU/память, необходимые для работы контейнера.</li> <li>• Limits: Максимальные CPU/память, разрешённые во время выполнения контейнера. Для определения единиц см. Resource Units.</li> </ul> |

| Параметр           | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <p>Коэффициент overcommit namespace:</p> <ul style="list-style-type: none"> <li>Без коэффициента overcommit:</li> </ul> <p>Если существуют квоты ресурсов namespace: Requests/limits контейнера наследуют значения по умолчанию namespace (можно изменить).</p> <p>Без квот namespace: Нет значений по умолчанию; настраивается Request.</p> <ul style="list-style-type: none"> <li>С коэффициентом overcommit:</li> </ul> <p>Requests рассчитываются автоматически как Limits / Overcommit ratio (неизменяемо).</p> <p>Ограничения:</p> <ul style="list-style-type: none"> <li><math>\text{Request} \leq \text{Limit} \leq \text{максимальная квота namespace.}</math></li> <li>Изменение коэффициента overcommit требует пересоздания pod для вступления в силу.</li> <li>Коэффициент overcommit отключает ручную настройку request.</li> <li>При отсутствии квот namespace — нет ограничений ресурсов контейнера</li> </ul> |
| Extended Resources | Настройка расширенных ресурсов, доступных в кластере (например, vGPU, pGPU)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Volume Mount       | <p>Конфигурация персистентного хранилища.</p> <p>Операции:</p> <ul style="list-style-type: none"> <li>Для существующих томов pod: нажмите Add</li> <li>Если томов pod нет: нажмите Add &amp; Mount</li> </ul> <p>Параметры:</p> <ul style="list-style-type: none"> <li>mountPath: Путь в файловой системе контейнера (например, /data)</li> <li>subPath: Относительный путь файла/каталога внутри тома.</li> </ul> <p>Для ConfigMap/Secret: выбор конкретного ключа</p> <ul style="list-style-type: none"> <li>readOnly: Монтировать только для чтения (по умолчанию: чтение-запись)</li> </ul>                                                                                                                                                                                                                                                                                                                                |
| Port               | <p>Открытие портов контейнера.</p> <p><b>Пример:</b> Открыть TCP порт 6379 с именем redis.</p> <p>Поля:</p> <ul style="list-style-type: none"> <li>protocol: TCP/UDP</li> <li>Port: Открываемый порт (например, 6379)</li> <li>name: Идентификатор, совместимый с DNS (например, redis)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Параметр                       | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Startup Commands & Arguments   | <p>Переопределение стандартных ENTRYPOINT/CMD:</p> <p><b>Пример 1:</b> Выполнить top -b</p> <p>- Command: ["top", "-b"]</p> <p>- ИЛИ Command: ["top"], Args: ["-b"]</p> <p><b>Пример 2:</b> Вывод \$MESSAGE:</p> <p>/bin/sh -c "while true; do echo \$(MESSAGE); sleep 10; done"</p>                                                                                                                                                                                             |
| More > Environment Variables   | <ul style="list-style-type: none"> <li>Статические значения: прямые пары ключ-значение</li> <li>Динамические значения: ссылки на ключи ConfigMap/Secret, поля pod (fieldRef), метрики ресурсов (resourceFieldRef)</li> </ul> <p><b>Примечание:</b> Переменные окружения переопределяют настройки образа/конфигурационных файлов</p>                                                                                                                                              |
| More > Referenced ConfigMap    | Внедрение всего ConfigMap/Secret как переменных окружения. Поддерживаемые типы Secret: Оpaque, <a href="https://kubernetes.io/basic-auth">kubernetes.io/basic-auth</a>                                                                                                                                                                                                                                                                                                           |
| More > Health Checks           | <ul style="list-style-type: none"> <li>Liveness Probe: Проверка здоровья контейнера (перезапуск при сбое)</li> <li>Readiness Probe: Проверка доступности сервиса (удаление из endpoints при сбое)</li> </ul>                                                                                                                                                                                                                                                                     |
| More > Log File                | <p>Настройка путей логов:</p> <ul style="list-style-type: none"> <li>По умолчанию: сбор stdout</li> <li>Шаблоны файлов: например, /var/log/*.log</li> </ul> <p>Требования:</p> <ul style="list-style-type: none"> <li>Драйвер хранения overlay2: поддерживается по умолчанию</li> <li>devicemapper: необходимо вручную монтировать EmptyDir в каталог логов</li> <li>Узлы Windows: обеспечить монтирование родительского каталога (например, c:/a для c:/a/b/c/*.log)</li> </ul> |
| More > Exclude Log File        | Исключение определённых логов из сбора (например, /var/log/aaa.log)                                                                                                                                                                                                                                                                                                                                                                                                              |
| More > Execute before Stopping | <p>Выполнение команд перед завершением контейнера.</p> <p><b>Пример:</b> echo "stop"</p> <p><b>Примечание:</b> Время выполнения команды должно быть меньше terminationGracePeriodSeconds pod</p>                                                                                                                                                                                                                                                                                 |

11. Нажмите **Add Container** или **Add Init Container**. Особенности Init Container:

- Запускается перед контейнерами приложения (последовательное выполнение).
- Освобождает ресурсы после завершения.
- Удаление разрешено, если:
  - Pod содержит >1 контейнера приложения И  $\geq 1$  init контейнер.
  - Не разрешено для Pod с одним контейнером приложения.

12. Нажмите **Create**.

## Шаг 2. Настройка сервиса

1. Задайте параметры:

| Параметр | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service  | <p>Kubernetes Service, обеспечивает доступ к приложению, работающему в вашем кластере, через единую внешнюю точку доступа, даже если workload распределён по нескольким backend.</p> <p><b>Примечание:</b> Префикс имени по умолчанию для внутреннего маршрута, создаваемого под приложением, — это имя вычислительного компонента. Если тип вычислительного компонента (режим развёртывания) — StatefulSet, рекомендуется не менять имя внутреннего маршрута (имя workload), иначе могут возникнуть проблемы с доступом к workload</p> |

## Шаг 3. Настройка Ingress

1. Задайте параметры:

| Параметр | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ingress  | <p>Kubernetes Ingress, позволяет сделать ваш HTTP (или HTTPS) сетевой сервис доступным с помощью протоколно-ориентированной конфигурации, понимающей веб-концепции, такие как URI, имена хостов, пути и др. Концепция Ingress позволяет направлять трафик на разные backend в зависимости от правил, определённых через Kubernetes API.</p> <p><b>Примечание:</b> Service, используемый при создании Ingress под приложением, должен быть ресурсом, созданным в рамках текущего приложения. При этом убедитесь, что Service связан с workload под приложением, иначе обнаружение и доступ к workload не будут работать</p> |

13. Нажмите **Create**.

## Создание приложений из Helm Chart

Основываясь на Helm Chart, представляет собой нативный шаблон развёртывания приложений. Helm Chart — это набор файлов, определяющих ресурсы Kubernetes, предназначенный для упаковки приложений и облегчения их распространения с возможностями контроля версий. Это обеспечивает беспрепятственный переход между средами, например, миграцию из среды разработки в производственную среду.

### Шаги

1. Перейдите в **Container Platform**.
2. В левой панели перейдите в **Applications > Applications**.
3. Выберите способ создания **Create from Catalog**.
4. Выберите **Chart** и настройте параметры, такие как `resources.requests`, `resources.limits` и остальные.
5. Нажмите **Create**.

Веб-консоль перенаправит вас на страницу деталей **Application > [Native Applications]**. Процесс создания займет некоторое время. В случае неудачи операции следуйте подсказкам интерфейса для её завершения.

### Создание приложений из YAML

1. Перейдите в **Container Platform**.
2. В левой панели перейдите в **Applications > Applications**.
3. Выберите **Create from YAML**.
4. Заполните конфигурацию и нажмите **Create**.
5. Соответствующий **Deployment** можно просмотреть на странице деталей.

```
webapp-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 name: webapp
 labels:
 app: webapp
 env: prod
spec:
 replicas: 3
 selector:
 matchLabels:
 app: webapp
 template:
 metadata:
 labels:
 app: webapp
 tier: frontend
 spec:
 containers:
 - name: webapp
 image: nginx:1.25-alpine
 ports:
 - containerPort: 80
```

```

resources:
 requests:
 cpu: "100m"
 memory: "128Mi"
 limits:
 cpu: "250m"
 memory: "256Mi"

webapp-service.yaml
apiVersion: v1
kind: Service
metadata:
 name: webapp-service
spec:
 selector:
 app: webapp
 ports:
 - protocol: TCP
 port: 80
 targetPort: 80
 type: ClusterIP

```

### Создание приложений из кода

Создание приложения из кода реализовано с использованием технологии Source to Image (S2I). S2I — это автоматизированная платформа для построения контейнерных образов непосредственно из исходного кода. Такой подход стандартизирует и автоматизирует процесс сборки приложения, позволяя разработчикам сосредоточиться на разработке исходного кода, не беспокоясь о деталях контейнеризации.

#### Шаги

1. Перейдите в **Container Platform**.
2. В левой панели перейдите в **Applications > Applications**.
3. Выберите **Create from Code**.
4. Задайте параметры.
5. Нажмите **Create**. Соответствующий деплоймент можно просмотреть на странице **Detail Information**.

### Создание приложений на основе Операторов

Приложения на основе Операторов — это наборы ресурсов, предоставляемые Operator. На основе этих приложений, поддерживаемых Operator, вы можете быстро развернуть компонентное приложение и использовать возможности Operator для автоматизации всего жизненного цикла управления приложением.



## Шаги

1. Перейдите в **Container Platform**.
2. В левой панели перейдите в **Applications > Applications**.
3. Нажмите **Create**.
4. Выберите **Create from Catalog**.
5. Выберите экземпляр, поддерживаемый Operator, и настройте **Custom Resource Parameters**. Выберите экземпляр приложения, управляемого Operator, и настройте его спецификации Custom Resource (CR) в манифесте CR, включая:
  - `spec.resources.limits` (ограничения ресурсов на уровне контейнера),
  - `spec.resourceQuota` (политики квот, определённые Operator). Другие параметры, специфичные для CR, такие как `spec.replicas`, `spec.storage.className` и т.д.
6. Нажмите **Create**. Веб-консоль перейдёт на страницу **Applications > Operator Backed Apps**.

**Примечание:** Процесс создания ресурсов Kubernetes требует асинхронного согласования (reconciliation). Завершение может занять несколько минут в зависимости от состояния кластера.

## Устранение неполадок

Если создание ресурсов не удалось:

1. Проверьте ошибки согласования контроллера:

```
kubectl get events --field-selector involvedObject.kind=<Your-Custom-Resource> --sort-by=.metadata.creationTimestamp
```
2. Проверьте доступность API-ресурсов:

```
kubectl api-resources | grep <Your-Resource-Type>
```
3. Повторите создание после проверки готовности CRD/Operator:

```
kubectl apply -f your-resource-manifest.yaml
```

## Создание приложений с помощью CLI

`kubectl` — это основной интерфейс командной строки (CLI) для взаимодействия с кластерами Kubernetes. Он функционирует как клиент для Kubernetes API Server — RESTful HTTP API, который служит управляющим программным интерфейсом. Все операции Kubernetes доступны через API endpoints, и `kubectl` фактически преобразует команды CLI в соответствующие API-запросы для управления ресурсами кластера и рабочими нагрузками приложений (Deployments, StatefulSets и др.).

CLI-инструмент облегчает развёртывание приложений, интеллектуально интерпретируя входные артефакты (образы, Chart и т. д.) и создавая соответствующие объекты Kubernetes API. Создаваемые ресурсы зависят от типа входных данных:

- **Image:** напрямую создаёт Deployment,
- **Chart:** инстанцирует все объекты, определённые в Helm Chart.

## Шаги

1. Перейдите в **Container Platform**.
2. Нажмите на иконку терминала в правом нижнем углу.
3. Дождитесь инициализации сессии (1-3 секунды).
4. Выполните команды `kubectl` в интерактивной оболочке:  
`kubectl get pods -n ${CURRENT_NAMESPACE}`
5. Просматривайте вывод команд в реальном времени

## Создание HPA

HPA (Horizontal Pod Autoscaler) автоматически масштабирует количество подов вверх или вниз на основе заданных политик и метрик, позволяя приложениям справляться с резкими всплесками нагрузки, одновременно оптимизируя использование ресурсов в периоды низкой активности.

## CLI

Вы можете создать горизонтальный автоскейлер подов с помощью командной строки, определив YAML-файл и используя команду `kubectl create`. В следующем примере показан автоскейлинг для объекта `Deployment`. Изначально требуется 3 пода. Объект HPA увеличивает минимум до 5. Если использование CPU на подах достигает 75%, количество подов увеличивается до 7:

1. Создайте YAML-файл с именем `hpa.yaml` со следующим содержимым:  

```
apiVersion: autoscaling/v2 # Используйте API autoscaling/v2
kind: HorizontalPodAutoscaler # Имя ресурса HPA
metadata:
 name: hpa-demo # Имя деплоя для масштабирования
 namespace: default
spec:
 maxReplicas: 7 # Максимальное количество реплик для масштабирования вверх
 minReplicas: 3 # Минимальное количество реплик для поддержания
 scaleTargetRef:
 apiVersion: apps/v1 # Укажите версию API объекта для масштабирования
 kind: Deployment # Укажите тип объекта. Объект должен быть Deployment,
 ReplicaSet или StatefulSet
 name: deployment-demo # Целевой ресурс, к которому применяется HPA
 targetCPUUtilizationPercentage: 75 # Целевой процент использования CPU, который
 запускает масштабирование
```
2. Примените YAML-файл для создания HPA:  
`kubectl create -f hpa.yaml`

Пример вывода:

```
horizontalpodautoscaler.autoscaling/hpa-demo created
```

- После создания HPA вы можете посмотреть новое состояние деплоя, выполнив команду:

```
kubectl get deployment deployment-demo
```

Пример вывода:

```
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-demo 5/5 5 5 3m
```

- Также можно проверить статус вашего HPA:

```
kubectl get hpa hpa-demo
```

Пример вывода:

```
NAME REFERENCE TARGETS MINPODS MAXPODS
REPLICAS AGE
hpa-demo Deployment/deployment-demo 0%/75% 3 7 3
2m
```

### Веб-консоль

- Перейдите в **Container Platform**.
- В левой панели выберите **Workloads > Deployments**.
- Кликните на имя деплоймента.
- Прокрутите вниз до раздела **Elastic Scaling** и нажмите **Update**.
- Выберите **Horizontal Scaling** и заполните конфигурацию политики:

| Параметр       | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pod Count      | <p>После успешного создания деплоя необходимо оценить <b>Минимальное количество подов</b>, соответствующее известным и регулярным изменениям бизнес-объема, а также <b>Максимальное количество подов</b>, которое может поддерживаться квотой namespace при высокой нагрузке.</p> <p>Максимальное и минимальное количество подов можно изменять после настройки, рекомендуется сначала получить более точное значение через нагрузочное тестирование и постоянно корректировать в процессе эксплуатации для удовлетворения бизнес-требований.</p> |
| Trigger Policy | <p>Перечислите <b>Метрики</b>, чувствительные к изменениям бизнеса, и их <b>Целевые пороги</b>, которые запускают действия масштабирования вверх или вниз.</p> <p>Например, если вы установите <i>CPU Utilization = 60 %</i>, как только использование CPU отклонится от 60%, платформа начнет автоматически корректировать количество подов на основе недостаточного или избыточного распределения ресурсов текущего деплоя.</p> <p><b>Примечание:</b> Типы метрик включают встроенные и пользовательские.</p>                                   |

| Параметр                   | Описание                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | Пользовательские метрики применимы только к деплоям в нативных приложениях, и их необходимо предварительно добавить custom metrics                                                                                                                                                                                                                                                                        |
| Scale Up/Down Step (Alpha) | Для бизнесов с особыми требованиями к скорости масштабирования можно постепенно адаптироваться к изменениям объема, задавая <b>Шаг масштабирования вверх</b> или <b>Шаг масштабирования вниз</b> .<br>Для шага масштабирования вниз можно настроить <b>Окно стабильности</b> , по умолчанию 300 секунд, что означает необходимость ожидания 300 секунд перед выполнением действий по масштабированию вниз |

6. Нажмите **Update**.

## Создание VPA

Вы можете создать VerticalPodAutoscaler (VPA), который будет рекомендовать или автоматически обновлять запросы и лимиты ресурсов CPU и памяти для ваших подов на основе их моделей использования.

### CLI

Вы можете создать VerticalPodAutoscaler через интерфейс командной строки, определив YAML-файл и используя команду `kubectl create`. В следующем примере показано вертикальное автомасштабирование подов для объекта Deployment:

1. Создайте YAML-файл с именем `vpa.yaml` со следующим содержимым:

```

apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
 name: my-deployment-vpa # Имя VPA
 namespace: default
spec:
 targetRef:
 apiVersion: apps/v1 # Версия API объекта для масштабирования
 kind: Deployment # Укажите целевой объект рабочей нагрузки. VPA использует
 селектор рабочей нагрузки для поиска подов, которым требуется корректировка
 ресурсов. Поддерживаемые типы рабочих нагрузок включают DaemonSet,
 Deployment, ReplicaSet, StatefulSet, ReplicationController, Job и CronJob
 name: my-deployment # Целевой ресурс, к которому применяется VPA
 updatePolicy:
 updateMode: 'Off' # Политика обновления, определяющая, как VPA применяет
 рекомендации
 resourcePolicy:
```

containerPolicies:

- containerName: '\*' # Применить политику ко всем контейнерам в поде
- mode: 'Auto' # Установить режим в Auto или Off. Auto означает, что для этого контейнера будут генерироваться рекомендации, Off — что рекомендации не будут генерироваться

Параметр updateMode может принимать значения:

- Auto: Автоматически устанавливает запросы ресурсов при создании подов и обновляет текущие поды до рекомендуемых значений. В настоящее время эквивалентно "Recreate". Этот режим может вызвать простой приложения. После поддержки обновлений ресурсов подов на месте режим "Auto" будет использовать этот механизм.
- Recreate: Автоматически устанавливает запросы ресурсов при создании подов и эвакуирует текущие поды для обновления до рекомендуемых значений. Не использует обновления на месте.
- Initial: Устанавливает запросы ресурсов только при создании подов, без последующих изменений.
- Off: Не изменяет запросы ресурсов подов автоматически, только предоставляет рекомендации в объекте VPA.

resourcePolicy — политика ресурсов, которая может задавать конкретные стратегии для разных контейнеров. Например, установка режима контейнера в "Auto" означает, что для этого контейнера будут рассчитываться рекомендации, а "Off" — что рекомендации не будут рассчитываться.

2. Примените YAML-файл для создания VPA:

```
kubectl create -f vpa.yaml
```

Пример вывода:

```
verticalpodautoscaler.autoscaling.k8s.io/my-deployment-vpa created
```

3. После создания VPA вы можете посмотреть рекомендации, выполнив команду:

```
kubectl describe vpa my-deployment-vpa
```

Пример вывода (частичный):

Status:

Recommendation:

Container Recommendations:

Container Name: my-container

Lower Bound:

Cpu: 100m

Memory: 262144k

Target:

Cpu: 200m

Memory: 524288k

Upper Bound:

Cpu: 300m  
Memory: 786432k

### Веб-консоль

1. Перейдите в **Container Platform**.
2. В левой панели выберите **Workloads > Deployments**.
3. Кликните по имени деплоймента.
4. Прокрутите вниз до раздела **Elastic Scaling** и нажмите **Update**.
5. Выберите **Vertical Scaling** и настройте правила масштабирования:

| Параметр              | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Режим масштабирования | В настоящее время поддерживается режим <b>Manual Scaling</b> , который предоставляет рекомендуемые конфигурации ресурсов на основе анализа прошлых данных об использовании ресурсов. Вы можете вручную корректировать значения согласно рекомендациям. Корректировки приведут к пересозданию и перезапуску подов, поэтому выбирайте подходящее время, чтобы не повлиять на работающие приложения. Обычно после того, как поды работают более 8 дней, рекомендуемые значения становятся точными. Обратите внимание, что при недостатке ресурсов в кластере масштабирование может привести к состоянию Pending у подов. Убедитесь, что в кластере достаточно ресурсов или установлены разумные квоты, либо настройте оповещения для мониторинга условий масштабирования |
| Целевой контейнер     | По умолчанию выбирается первый контейнер рабочей нагрузки. Вы можете включить рекомендации по лимитам ресурсов для одного или нескольких контейнеров при необходимости                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

6. Нажмите **Update**.

### Описание статусов

#### Статусы приложений

| Статус            | Значение                                                         |
|-------------------|------------------------------------------------------------------|
| Running           | Все вычислительные компоненты работают в нормальном режиме       |
| Partially Running | Некоторые вычислительные компоненты работают, другие остановлены |
| Stopped           | Все вычислительные компоненты остановлены                        |

| Статус                  | Значение                                                                     |
|-------------------------|------------------------------------------------------------------------------|
| Processing              | По крайней мере один вычислительный компонент находится в состоянии ожидания |
| No Computing Components | В приложении отсутствуют вычислительные компоненты                           |
| Failed                  | Развёртывание завершилось с ошибкой                                          |

**Примечание:** Числа после статуса указывают количество вычислительных компонентов. Числа в статусе вычислительных компонентов указывают количество групп контейнеров.

### Статусы развёртывания

- Running: Все Pods работают в нормальном режиме.
- Processing: Есть Pods, которые не находятся в состоянии Running.
- Stopped: Все Pods остановлены.
- Failed: Развертывание завершилось с ошибкой.

## Управление приложениями

### Запуск приложения

1. Перейдите в **Container Platform**.
2. В левой панели нажмите **Application > Applications**.
3. Нажмите на название приложения.
4. Нажмите **Start**.

### Остановка приложения

1. Перейдите в **Container Platform**.
2. В левой панели нажмите **Application > Applications**.
3. Нажмите на название приложения.
4. Нажмите **Stop**.
5. Ознакомьтесь с сообщением-подтверждением и, убедившись, что всё верно, нажмите **Stop**.

### Удаление приложений

При удалении приложения одновременно удаляется само приложение и все Kubernetes-ресурсы, которые непосредственно входят в его состав. Кроме того, это действие разрывает любые связи приложения с другими Kubernetes-ресурсами, которые не были напрямую частью его определения.

## Мониторинг

- Поддерживается просмотр данных мониторинга ресурсов для компонентов рабочих нагрузок на платформе за последние 7 дней (с возможностью настройки периода

хранения данных мониторинга). Включает статистику по приложениям, рабочим нагрузкам, подам, а также тренды/рейтинги использования CPU/памяти.

- Поддерживается мониторинг на уровне Namespace.
- Поддерживаемый мониторинг на уровне рабочих нагрузок: **Applications, Deployments, DaemonSets, StatefulSets** и **Pods**.

## Мониторинг на уровне Namespace

### Просмотр панели мониторинга

1. В **Container Platform** нажмите **Observe > Dashboards**.
2. Просмотрите данные мониторинга в рамках namespace. Предоставляются три панели: **Applications Overview, Workloads Overview** и **Pods Overview**.
3. Переключайтесь между панелями для мониторинга целевого **Overview**.

### Создание панели мониторинга

1. В **Platform Management** создайте выделенную панель мониторинга, следуя инструкции в разделе “Наблюдаемость”.
2. Настройте следующие метки для отображения панели мониторинга на уровне Namespace в **Container Platform**:
  - `cpaas.io/dashboard.folder: container-platform`,
  - `cpaas.io/dashboard.tag.overview: "true"`.

## Мониторинг на уровне рабочих нагрузок

Просмотр панели мониторинга по умолчанию

1. В **Container Platform** нажмите **Workloads > Deployments**.
2. Выберите имя Deployment из списка.
3. Перейдите на вкладку **Monitoring** для просмотра стандартных метрик мониторинга:

| Ресурс мониторинга | Гранулярность метрики    | Техническое определение                                                                                                                                                                            |
|--------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU                | Utilization/Usage        | <b>Utilization</b> = Usage/Limit (лимиты)<br>Оценка конфигурации лимитов контейнера. Высокая загрузка указывает на недостаточные лимиты.<br><b>Usage</b> отражает фактическое потребление ресурсов |
| Memory             | Utilization/Usage        | <b>Utilization</b> = Usage/Limit (лимиты)<br>Метод оценки аналогичен CPU. Высокий показатель может вызвать нестабильность компонента                                                               |
| Network Traffic    | Inflow Rate/Outflow Rate | Сетевой трафик (байт/сек) входящий/исходящий из подов                                                                                                                                              |



| Ресурс мониторинга | Гранулярность метрики        | Техническое определение                                                         |
|--------------------|------------------------------|---------------------------------------------------------------------------------|
| Network Packet     | Receiving Rate/Transmit Rate | Сетевые пакеты (кол-во/сек) принимаемые/отправляемые подами                     |
| Disk Rate          | Read/Write                   | Скорость чтения/записи (байт/сек) смонтированных томов на рабочую нагрузку      |
| Disk IOPS          | Read/Write                   | Операции ввода/вывода в секунду (IOPS) смонтированных томов на рабочую нагрузку |

### Пользовательская панель мониторинга

Нажмите **Toggle Icon** для переключения на пользовательские панели. Для создания выделенной панели мониторинга на уровне рабочих нагрузок обратитесь к разделу “Наблюдаемость”.

**Примечание:** Наведите курсор на кривые графика, чтобы просмотреть метрики по каждому поду и выражения PromQL в конкретные моменты времени. Если количество подов превышает 15, отображаются только топ-15 записей, отсортированных по убыванию.

### Логирование

Когда приложения, рабочие нагрузки или другие ресурсы проявляют аномальное поведение, анализ логов помогает диагностировать коренные причины.

Просматривайте логи контейнеров в режиме реального времени через интерфейс Deployment:

1. В **Container Platform** нажмите **Workloads > Deployments**.
2. Выберите имя Deployment из списка.
3. Перейдите на вкладку **Logs** для просмотра подробных записей. Управляйте логами с помощью интерфейса:

| Операция      | Описание                                                                                              |
|---------------|-------------------------------------------------------------------------------------------------------|
| Pod/Container | Переключайтесь между Pod и Container с помощью выпадающего списка для просмотра соответствующих логов |
| Previous Logs | Просмотр логов завершённых контейнеров (доступно при restartCount контейнера > 0)                     |
| Lines         | Настройка размера буфера отображаемых логов: 1k/10k/100k строк                                        |
| Wrap Line     | Переключение переноса строк для длинных записей лога (включено по умолчанию)                          |
| Find          | Полнотекстовый поиск с подсветкой совпадений и переходом по нажатию Enter                             |

| Операция    | Описание                                                                                                                                                 |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Raw         | Необработанные потоки логов, напрямую захваченные с интерфейсов контейнеров в режиме реального времени (CRI) без форматирования, фильтрации или усечения |
| Export      | Скачивание необработанных логов                                                                                                                          |
| Full Screen | Клик по усечённой строке для просмотра полного содержимого в модальном окне                                                                              |

#### Особенности:

- **Обработка усечения:** Записи лога длиной более 2000 символов будут усечены с добавлением многоточия ...
  - Усечённые части не могут быть найдены функцией поиска на странице.
  - Нажмите на маркер многоточия ... в усечённых строках для просмотра полного содержимого в модальном окне.
- **Надёжность копирования:** Избегайте прямого копирования из визуального просмотрщика логов при наличии маркеров усечения (...) или ANSI цветовых кодов. Всегда используйте функции **Export** или **Raw** для получения полных логов.
- **Политика хранения:** Живые логи подчиняются настройкам ротации логов Kubernetes. Для исторического анализа используйте логи в разделе “Наблюдаемость”.

#### События

Информация о событиях, генерируемая изменениями состояния ресурсов Kubernetes и обновлениями операционного статуса, с интегрированным визуальным интерфейсом запросов. Когда приложения, рабочие нагрузки или другие ресурсы сталкиваются с исключениями, анализ событий в реальном времени помогает выявить первопричины.

Просматривайте события контейнеров в режиме реального времени через интерфейс Deployment:

1. В Container Platform нажмите Workloads > Deployments.
2. Выберите имя Deployment из списка.
3. Перейдите на вкладку **Events**, чтобы просмотреть подробные записи.

Ниже панели с кратким описанием событий перечислены все соответствующие события за указанный период времени.

4. Нажмите на карточку события, чтобы просмотреть полные детали события. Каждая карточка отображает:
  - **Тип ресурса:** Тип ресурса Kubernetes, обозначенный иконками и сокращениями:
    - P = Pod
    - RS = ReplicaSet
    - D = Deployment

- SVC = Service
- **Имя ресурса:** Название целевого ресурса.
- **Причина события:** Причина, зарегистрированная Kubernetes (например, FailedScheduling).
- **Уровень события:** Важность события.
  - Normal : Информационное
  - Warning : Требуется немедленного внимания
- **Время:** Время последнего возникновения, количество повторений.

**Примечание:** Kubernetes позволяет администраторам настраивать период хранения событий через контроллер Event TTL с периодом хранения по умолчанию 1 час. Просроченные события автоматически удаляются системой. Для получения полного исторического архива обращайтесь к событиям в разделе “Наблюдаемость”.

## Вход в контейнер через EXEC

### Вход в контейнер через приложения

Вы можете войти во внутренний экземпляр контейнера с помощью команды `kubectl exec`, что позволяет выполнять операции в командной строке в окне веб-консоли. Кроме того, вы можете легко загружать и скачивать файлы внутри контейнера с помощью функции передачи файлов.

### Требования

- Контейнер должен работать корректно.
- При использовании функции передачи файлов в контейнере должен быть доступен инструмент `tar`, а операционная система контейнера не должна быть Windows.

### Шаги

1. Перейдите в **Container Platform**.
2. В левой панели нажмите **Application > Applications**.
3. Нажмите на приложения.
4. Найдите `workload` и нажмите **EXEC > Pod Name**.
5. Введите команду, которую хотите выполнить.
6. Нажмите **OK**, чтобы войти в окно веб-консоли и выполнить операции в командной строке.
7. Нажмите **File Transfer**. Введите **Upload Path** для загрузки файлов для тестирования в контейнер; или введите **Download Path** для скачивания логов и других файлов из контейнера на локальную машину для анализа.

### Вход в контейнер через Pod

Вы можете войти во внутренний экземпляр контейнера с помощью команды `kubectl exec`, что позволяет выполнять операции в командной строке в окне веб-консоли. Кроме того, вы

можете легко загружать и скачивать файлы внутри контейнера с помощью функции передачи файлов.

### Требования

- Контейнер должен работать корректно.
- При использовании функции передачи файлов в контейнере должен быть доступен инструмент `tar`, а операционная система контейнера не должна быть Windows.

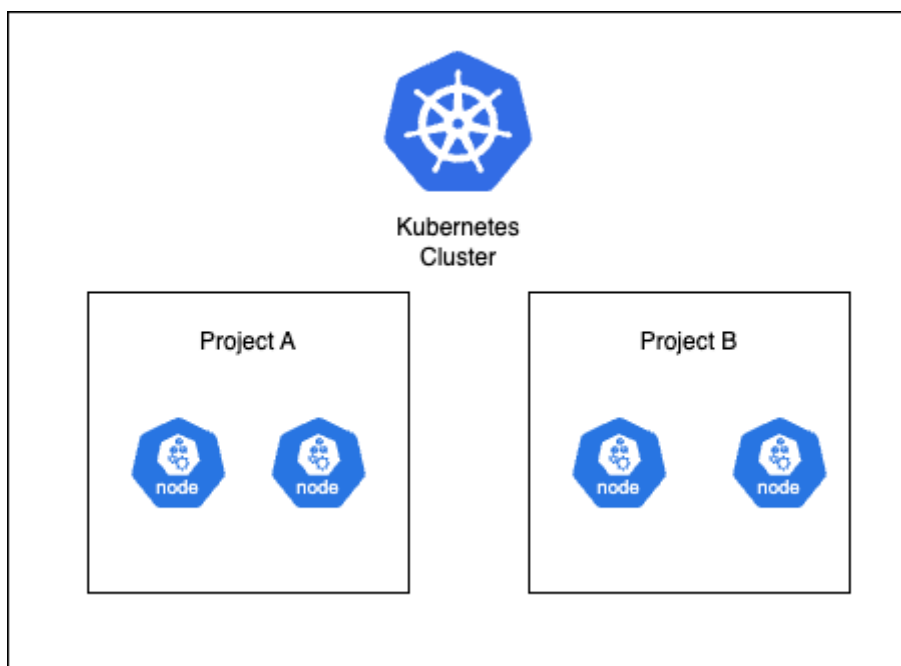
### Шаги

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Workloads > Pods**.
3. Нажмите **:** > **EXEC > Container Name**.
4. Введите команду, которую хотите выполнить.
5. Нажмите **ОК**, чтобы войти в окно веб-консоли и выполнить операции в командной строке.
6. Нажмите **File Transfer**. Введите **Upload Path** для загрузки файлов для тестирования в контейнер; или введите **Download Path** для скачивания логов и других файлов из контейнера на локальную машину для анализа.

### Сетевая изоляция узлов

Node Isolation Strategy предоставляет стратегию изоляции узлов на уровне проекта, которая позволяет проектам использовать узлы кластера эксклюзивно.

### Архитектура



Node Isolation Strategy реализована на основе компонента Container Platform Cluster Core, обеспечивая возможность изоляции узлов между проектами за счёт выделения узлов в

каждом кластере нагрузки. При создании контейнеров в проекте они принудительно планируются на узлы, выделенные именно этому проекту.

## Настройка сетевой изоляции узлов

Создайте политику изоляции узлов для текущего кластера, позволяющую указанным проектам иметь эксклюзивный доступ к узлам сгруппированных ресурсов внутри кластера, тем самым ограничивая узлы, на которых могут запускаться Pods в рамках проекта, достигая физической изоляции ресурсов между проектами.

### Создание стратегии изоляции узлов

1. Перейдите в **Platform Management**.
2. В левой панели нажмите **Security Settings > Node Isolation Strategy**.
3. Нажмите **Create Node Isolation Strategy**.
4. Настройте параметры:

| Параметр            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Project Exclusivity | <p>Включение или отключение переключателя для узлов, содержащихся в политике изоляции проекта, настроенной в стратегии; нажмите для переключения в положение вкл или выкл:</p> <ul style="list-style-type: none"> <li>• Когда переключатель включен, только Pods в указанном проекте из политики могут запускаться на узлах, включённых в политику;</li> <li>• при выключенном переключателе Pods из других проектов текущего кластера также могут запускаться на узлах, включённых в политику, помимо указанного проекта.</li> </ul> <p>По умолчанию переключатель включен.</p> |
| Project             | <p>Проект, для которого настроено использование узлов в политике. Нажмите на выпадающий список <b>Project</b> и отметьте флажок перед названием проекта для выбора нескольких проектов.</p> <p><b>Примечание:</b> Для проекта может быть настроена только одна политика изоляции узлов: если проект уже имеет назначенную политику изоляции узлов, его нельзя выбрать. Поддерживается ввод ключевых слов в выпадающем списке для фильтрации и выбора проектов</p>                                                                                                                |
| Node                | <p>IP-адреса вычислительных узлов, выделенных для использования проектом в политике.</p> <p>Нажмите на выпадающий список <b>Node</b> и отметьте флажком нужные узлы.</p> <p><b>Примечание:</b> Узел может принадлежать только одной политике изоляции: если узел уже принадлежит другой политике изоляции, его нельзя выбрать. Поддерживается ввод ключевых слов в выпадающем списке для фильтрации и выбора узлов</p>                                                                                                                                                           |

5. Нажмите **Create**.

После создания политики существующие Pods в проекте, не соответствующие текущей политике, будут запланированы на узлы, включённые в текущую политику, после их пересоздания.

При включённом параметре **Project Exclusivity Pods**, уже запущенные на узлах, не будут автоматически исключены; при необходимости исключения требуется ручное планирование.

**Удаление стратегии изоляции узлов**

**Примечание:** После удаления политики изоляции узлов проект больше не будет ограничен запуском на определённых узлах, и узлы перестанут использоваться эксклюзивно этим проектом.

1. В левой навигационной панели нажмите **Security > Node Isolation Strategy**.
2. Найдите политику изоляции узлов, нажмите **:** > **Delete**.

## GitOps

### Введение в MWS GitOps

MWS Container Platform GitOps — это современный подход к непрерывной доставке и эксплуатации, который использует Git в качестве центрального «единого источника правды» (SSOT) для определения и управления инфраструктурой, конфигурациями приложений и процессами развертывания. Объединяя код приложений, конфигурационные файлы и определения Infrastructure as Code (IaC) в одном репозитории Git. GitOps обеспечивает полный контроль версий и автоматизированное управление всем жизненным циклом доставки программного обеспечения. В этой парадигме команды разработки и эксплуатации беспрепятственно сотрудничают на всех этапах разработки, тестирования и развертывания программного обеспечения, используя мощные механизмы ветвления, ревью кода и merge request в Git.

Когда изменения в коде или конфигурациях отправляются в репозиторий Git, автоматизированные инструменты обнаруживают эти обновления и запускают цепочку процессов, включая сборку, тестирование и развертывание. Такой рабочий процесс способствует непрерывной доставке и непрерывному развертыванию (CI/CD) программного обеспечения, обеспечивая быстрые и надежные релизы.

Решение нативно интегрирует Argo CD Operator для автоматизации операций жизненного цикла развертывания, включая подготовку, обновления и откаты.

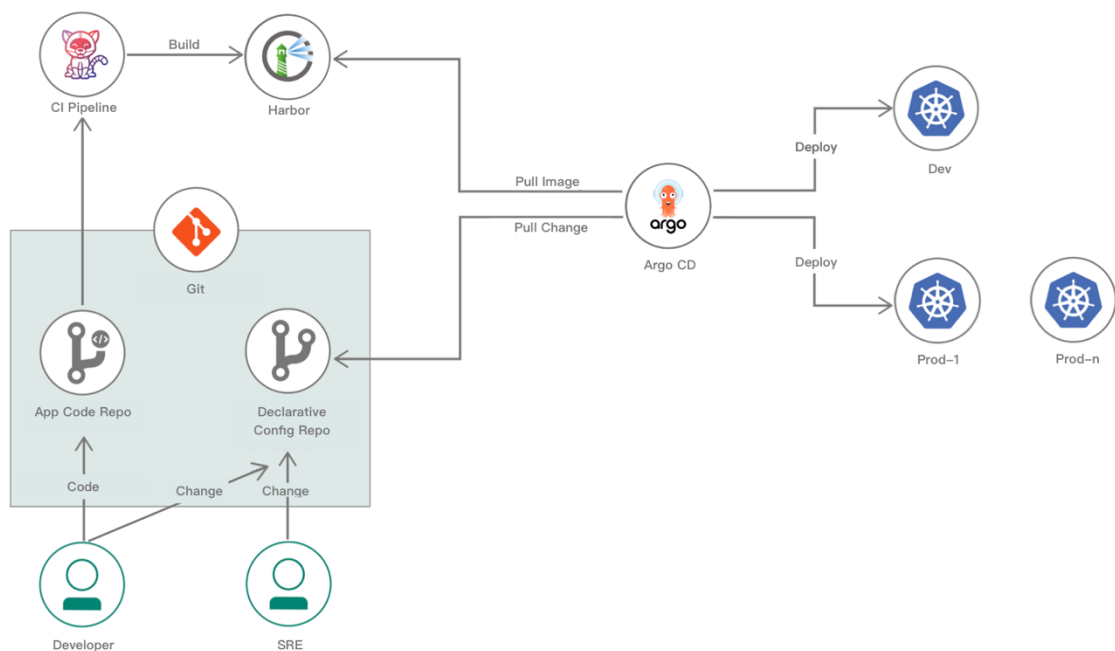
### Архитектура

GitOps — это современная концепция для непрерывной доставки и эксплуатации, а Argo CD — мощный инструмент, реализующий GitOps путем мониторинга файлов конфигурации в Git-репозитории и автоматической синхронизации их с целевой средой. Такой подход повышает скорость, надежность и безопасность поставки программного обеспечения, интегрируя весь процесс доставки в систему контроля версий Git.

MWS Container Platform GitOps, построенный на базе Argo CD, использует Git-репозиторий как единственный надежный источник для хранения приложений, конфигураций инфраструктуры и других файлов, обеспечивая быструю и точную доставку и развертывание в одном или нескольких кластерах Kubernetes.

Основные отличия GitOps от традиционных методов управления приложениями заключаются в следующем:

- Вместо прямого управления средой выполнения, GitOps контролирует её через поддержание репозитория конфигураций приложений в Git.
- Argo CD постоянно опрашивает репозиторий и исправляет расхождения между средой выполнения и репозиторием конфигураций приложений, гарантируя соответствие среды ожиданиям, предотвращая дрейф конфигураций и обеспечивая быструю восстановимость в случае сбоев.



## Возможности

**Непрерывная доставка контейнеризованных приложений:** В экосистеме Kubernetes GitOps отлично справляется с управлением развёртыванием, обновлениями и откатами контейнеризованных приложений. Разработчики коммитят код приложений и конфигурационные файлы Kubernetes в репозиторий Git, а инструменты GitOps автоматически развёртывают эти приложения в кластере Kubernetes, синхронизируя их с изменениями в конфигурационных файлах.

**Управление несколькими средами:** GitOps упрощает управление инфраструктурой и конфигурациями приложений в различных средах, таких как разработка, тестирование, staging и продакшн. С помощью ветвления и тегирования сред обеспечивается согласованность конфигураций при учёте необходимых кастомизаций для каждой среды.

**Эксплуатация микросервисной архитектуры:** В микросервисных архитектурах GitOps помогает командам эффективно организовывать развёртывание и обновления множества микросервисов. Код и конфигурации каждого микросервиса могут храниться отдельно в репозитории Git, что позволяет инструментам GitOps автоматизировать развёртывания и обновления с учётом зависимостей микросервисов и стратегий обновления, обеспечивая стабильность системы.

**Управление Infrastructure as Code (IaC):** GitOps бесшовно интегрируется с инструментами IaC, такими как Terraform и Ansible, для управления облачной инфраструктурой, конфигурациями серверов и сетевыми ресурсами. Хранение файлов конфигураций IaC в репозитории Git обеспечивает версионирование и автоматизацию развёртывания инфраструктуры, повышая управляемость и повторяемость.

**Межкомандное сотрудничество и совместное использование кода:** В крупных организациях нескольким командам часто необходимо совместно использовать код и конфигурации. GitOps предоставляет единую платформу для совместной работы над



разработкой, обмена кодом и управления конфигурациями через репозиторий Git, повышая эффективность сотрудничества и повторное использование кода.

## Установка MWS GitOps

### Установка оператора

1. Войдите в систему и перейдите на страницу **Platform Management**.
2. Нажмите **Marketplace > OperatorHub**.
3. Найдите оператор MWS Build of Argo CD, нажмите **Install** и перейдите на страницу **Install Argo CD**.

#### Параметры конфигурации:

| Параметр                 | Рекомендуемая конфигурация                                                                                                                  |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Channel</b>           | Канал по умолчанию — alpha                                                                                                                  |
| <b>Installation Mode</b> | Cluster: Все пространства имён в кластере используют один экземпляр оператора для создания и управления, что снижает использование ресурсов |
| <b>Namespace</b>         | Выберите Recommended Namespace: создаётся автоматически, если отсутствует                                                                   |
| <b>Upgrade Strategy</b>  | Auto: OperatorHub автоматически обновит оператор до последней версии при её доступности                                                     |

Рекомендуется использовать предложенную конфигурацию по умолчанию; просто нажмите **Install** для завершения установки оператора MWS Build of Argo CD.

### Создание инстанса Argo CD

1. Нажмите **Marketplace > OperatorHub**.
2. Найдите оператор MWS Build of Argo CD, кликните по нему, чтобы перейти на страницу с подробной информацией об Argo CD.
3. Нажмите **All Instances**.
4. Нажмите **Create Instance**, выберите карточку экземпляра Argo CD.
5. Нажмите **Create Instance**.

**Внимание:** на странице параметров конфигурации экземпляра используйте настройки по умолчанию, если нет специальных требований. Примечание: Если глобальный кластер не является высокодоступным (например, содержит только один управляющий узел), переключитесь в режим YAML при создании экземпляра и установите значение поля `ha.enabled` в `false`.

### Создание инстанса AppProject

**Внимание:** если вам не требуется использовать функцию Cluster Configuration Management, управляемую платформой, следующие шаги выполнять не нужно.

1. Найдите оператор MWS Build of Argo CD, кликните по нему, чтобы перейти на страницу с подробной информацией об MWS Argo CD.

2. Нажмите **All Instances**, затем **Create Instance**, выберите карточку экземпляра AppProject.
3. Переключитесь в режим YAML и замените существующее содержимое YAML на интерфейсе следующим кодом:

```
apiVersion: argoproj.io/v1alpha1
kind: AppProject
metadata:
 name: cpaas-system
 namespace: argocd
spec:
 clusterResourceWhitelist:
 - group: '*'
 kind: '*'
 destinations:
 - namespace: '*'
 server: '*'
 sourceRepos:
 - '*'
```

4. Нажмите **Create**.

После выполнения вышеописанных шагов вы успешно установили Argo CD.

## Управление приложениями

### Создание GitOps приложения

Используйте возможности управления приложениями MWS Container Platform GitOps для создания Argo CD ApplicationSe или Application через графический интерфейс платформы, обеспечивающий комплексное управление жизненным циклом контейнеризованных приложений через GitOps Applications.

### Создание Argo CD Application через веб-консоль

1. Перейдите в **Container Platform** и откройте **GitOps Applications**.
2. Нажмите **Create GitOps Application**.
3. Настройте параметры в разделах **Basic Info** и **Code Repository**:

| Параметр                        | Описание                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Type</b>                     | <b>Application</b> : объект Argo CD Application для развертывания в одном namespace.<br><b>ApplicationSet</b> : Argo CD ApplicationSet для кросс-кластерных/кросс-namespace развертываний                                                                                    |
| <b>Source</b>                   | <b>Platform integrated</b> : Предварительно настроенные репозитории GitLab/GitHub/Bitbucket.<br><b>ArgoCD integrated</b> : репозитории GitLab/GitHub/Bitbucket/Gitee/Gitea, интегрированные через Argo CD. Подробнее см. Integrating Code Repositories via Argo CD dashboard |
| <b>Integration Project Name</b> | Проект Toolchain, назначенный администратором                                                                                                                                                                                                                                |

| Параметр                   | Описание                                                                                                                                                                                                                                                 |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Version Identifiers</b> | Основа развертывания: Branch/Tag/Commit<br>Примечание:<br><b>Branch</b> использует последний коммит<br><b>Tag/Commit</b> по умолчанию последний, но настраивается                                                                                        |
| <b>Source File Type</b>    | <b>Kustomize</b> : использует kustomization.yaml для оверлеев; подробности в Kustomize Official Documentation.<br><b>Helm</b> : использует values.yaml для шаблонов; подробности в Helm Official Documentation.<br><b>Directory</b> : исходные манифесты |
| <b>Source Directory</b>    | Путь в репозитории, содержащий базовые манифесты. Поддерживается выбор корневой директории. Все ресурсы в этом пути будут развернуты в целевых кластерах                                                                                                 |
| <b>Custom Values</b>       | Если <b>Source File Type — Helm</b> , можно выбрать пользовательский файл Helm Values                                                                                                                                                                    |

#### 4. **Sync Policy** (интервал согласования 3 минуты):

| Параметр                         | Описание                                                                                                                                                                                                                       |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Manually Sync</b>             | Требуется подтверждение пользователя при обнаружении расхождений.                                                                                                                                                              |
| <b>Automatic Sync</b>            | Автоматическое согласование без вмешательства пользователя.                                                                                                                                                                    |
| <b>Sync Ignore Configuration</b> | Настройка с использованием встроенных/пользовательских шаблонов игнорирования, можно Просмотреть поля конфигурации Sync Ignore в YAML файле.<br><b>Примечание:</b> пользовательские шаблоны требуют настройки администратором. |

#### 5. Нажмите **Create**.

**Примечание:** при ручной синхронизации: Выберите Synchronize Immediately для немедленного развертывания или Synchronize Later для запуска вручную через страницу деталей.

#### **Просмотр полей конфигурации игнорирования синхронизации в YAML файле**

После настройки правил игнорирования синхронизации проверьте следующим образом:

1. Перейдите в **GitOps Application**.
2. Выберите целевое приложение.
3. Нажмите **Action > Update**.
4. Просмотрите YAML файл.

```
ignoreDifferences: # Конфигурация, фактически игнорируемая выбранным
 # шаблоном пользовательской синхронизации игнорирования
- group: apps
 kind: Deployment
```

jsonPointers:  
- /spec/replicas

## Создание Argo CD Application через YAML

1. Перейдите в **Container Platform** и откройте **GitOps Applications**.
2. Нажмите **Create GitOps Application**.
3. Переключитесь на вкладку **YAML**.
4. В разделе **YAML** используйте следующий **YAML**-файл и настройте соответствующую информацию. Замените `namespace` и `project` на свои `namespace` и `project`.

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
 name: guestbook
 namespace: argocd # Замените на ваш namespace
spec:
 project: default # Замените на ваш проект
 source:
 repoURL: https://github.com/argoproj/argocd-example-apps.git
 targetRevision: master
 path: helm-guestbook
 destination:
 server: https://kubernetes.default.svc
 namespace: guestbook
 syncPolicy:
 automated:
 prune: true
 selfHeal: true
 syncOptions:
 - CreateNamespace=true
```

5. Нажмите **Create**.

## Создание Argo CD Application через CLI

### Прerequisites

Плагин `web-cli` установлен, и переключатель `web-cli` включен.

```
kubectl apply -f application.yaml
```

## Управление GitOps Applications

| Действие | Описание                                                                                                                                                                                                                                                           |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Update   | Запуск обновлений через: <ul style="list-style-type: none"><li>• Иконку редактирования (✎) в списке GitOps Application</li><li>• Action &gt; Update в подробном просмотре.</li></ul> <b>Внимание:</b> эта операция перезапишет все созданные экземпляры приложения |

|                                 |                                                                                                                                                                                                                                                                            |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Manually Sync</b>            | При Sync Policy = Manually Sync : <ul style="list-style-type: none"> <li>• Запуск синхронизации через Action &gt; Sync в подробном просмотре при обнаружении расхождения конфигураций</li> <li>• Распространяет последние коммиты на все управляемые экземпляры</li> </ul> |
| <b>Delete</b>                   | Удаление через: <ul style="list-style-type: none"> <li>• Иконку удаления (🗑️) на странице списка</li> <li>• Action &gt; Delete в подробном просмотре</li> </ul> ОПАСНО: удаляет приложение и ВСЕ дочерние ресурсы                                                          |
| <b>Automatic Sync</b>           | Включение авто-согласования для поддержания желаемого состояния. Все экземпляры автоматически синхронизируются с изменениями в репозитории каждые 3 минуты                                                                                                                 |
| <b>Source</b>                   | Для приложений типа ApplicationSet: Нажмите ссылку Source для перехода на страницу деталей родительского Application.                                                                                                                                                      |
| <b>Application Distribution</b> | Расширение: <ul style="list-style-type: none"> <li>• Обновление существующей конфигурации ApplicationSet</li> <li>• В деталях ApplicationSet: Applications &gt; Add Distribution</li> </ul>                                                                                |

## Интеграция репозитория кода через панель управления Argo CD

Используйте нативную панель управления Argo CD для интеграции репозитория кода и их распределения, что позволяет разработчикам управлять GitOps-приложениями на протяжении всего их жизненного цикла через визуальный интерфейс.

### Подключение репозитория кода

1. Войдите в Argo CD с использованием URL доступа.
2. Нажмите на **Settings** в левой навигационной панели.
3. Нажмите на карточку **REPO**.
4. Нажмите **CONNECT REPO** в верхнем левом углу страницы.
5. Выберите метод подключения репозитория и заполните соответствующие параметры по необходимости.
6. Нажмите **CONNECT**.

### Связывание проекта

1. Нажмите на **Settings** в левой навигационной панели.
2. Нажмите на карточку **Projects**.
3. Выберите проект, в котором необходимо создать GitOps-приложение  
**Примечание:** Argo CD автоматически синхронизирует проекты в кластере, поэтому создавать их вручную не нужно
4. Нажмите **EDIT** в разделе **SOURCE REPOSITORIES**.
5. Нажмите **ADD SOURCE**, введите URL репозитория из шага «Подключение репозитория» и свяжите его с проектом.
6. Нажмите **SAVE**.

## Расширения

Платформа предоставляет комплексную систему расширений, которая позволяет пользователям расширять функциональность своих Kubernetes кластеров. Эта система разработана с учётом гибкости и удобства использования, что позволяет легко добавлять новые функции и возможности в кластеры.

Система состоит из двух основных типов расширений:

- **Operators:** Operators построены на базе фреймворка Operator Lifecycle Manager (OLM) v0 и обеспечивают специализированные операционные возможности для платформы. Эти расширения позволяют автоматизировать управление сложными приложениями и сервисами внутри вашего кластера.
- **Cluster Plugins:** Платформа включает собственную систему кластерных плагинов, специально разработанную для плагинов типа Chart. Эта система обеспечивает улучшенный опыт установки и управления по сравнению со стандартными методами, предлагая удобный интерфейс для работы с расширениями на основе Chart.

Поддержка множества операторов (Operator) и кластерных плагинов позволяет пользователям значительно расширять возможности платформы для удовлетворения конкретных эксплуатационных требований и сценариев использования.

## Operator

Операторы автоматизируют полный жизненный цикл Kubernetes-приложений, включая создание, обновления и удаление. Администраторы платформы могут разворачивать Операторы из OperatorHub на уровне каждого кластера.

### Источники

Платформа поддерживает пять источников Операторов:

- **MWS CP**

Предоставляется и поддерживается компанией MWS, с полным управлением жизненным циклом, обновлениями безопасности, технической поддержкой и обязательствами по SLA.

- **Curated**

Отобранные из open-source сообществ, поддерживаются в соответствии с версиями сообщества без изменений кода или перекомпиляции. MWS предоставляет рекомендации и обновления безопасности при необходимости, но не несёт обязательств по SLA и управлению жизненным циклом.

- **Community**

Предоставляются open-source сообществами с периодическими обновлениями для обеспечения возможности разворачивания, но без гарантий функциональности. Нет SLA и поддержки от MWS.

- **Marketplace**

Предоставляются и поддерживаются сертифицированными сторонними поставщиками MWS. MWS поддерживает интеграцию с платформой, при этом основная ответственность остаётся за поставщиком.

- **Custom**

Разрабатываются и загружаются пользователями для удовлетворения индивидуальных бизнес- или управленческих потребностей кластера.

### Процесс установки

1. Войдите в Веб-консоль и переключитесь в режим **Administrator**.
2. Перейдите в Marketplace > OperatorHub.
3. Если статус "Absent":
  - a. Скачайте пакет Оператора из Custom Portal или обратитесь в техническую поддержку.
  - b. Используйте violet для загрузки пакета в целевой кластер. См. Документацию по интерфейсу командной строки.
  - c. Перейдите в **Marketplace > Upload Packages**, переключитесь на вкладку Operator и подтвердите загрузку.
4. Если статус "Ready", нажмите **Install**. Для инструкций по использованию обратитесь к пользовательской документации Оператора.

## Процесс обновления

1. Загрузите новую версию Оператора.
2. Обновление будет выполнено согласно настроенной стратегии:
  - a. **Automatic Upgrade:** автоматическое обновление после загрузки.
  - b. Manual Upgrade:
    - i. Batch Upgrade: через Platform Management > Cluster Management > Clusters > Features.
    - ii. **Individual Upgrade:** подтвердите обновление в OperatorHub.

## Cluster Plugin

Плагины кластера позволяют расширить функциональность вашей платформы путем установки дополнительных компонентов. В этом документе описывается, как просматривать, устанавливать и обновлять плагины кластера на вашей платформе.

### Просмотр доступных плагинов

Чтобы просмотреть все плагины, предоставляемые платформой:

1. Перейдите в представление управления платформой.
2. Нажмите в левом навигационном меню: **Platform Management > Marketplace > Cluster Plugin.**

На этой странице отображаются все доступные плагины и их текущий статус.

### Установка плагина

Если у плагина отображается статус «absent», выполните следующие шаги для его установки:

1. Скачайте пакет плагина:
  - a. Посетите Custom Portal, чтобы скачать соответствующий пакет плагина.
  - b. Если у вас нет доступа к Custom Portal, обратитесь в техническую поддержку.
2. Загрузите пакет в платформу:
  - a. Используйте инструмент violet для публикации пакета в платформе.
  - b. Подробные инструкции по использованию этого инструмента смотрите в разделе CLI.
3. Проверьте загрузку:
  - a. Перейдите в **Platform Management > Marketplace > Upload Packages.**
  - b. Переключитесь на вкладку **Cluster Plugin.**
  - c. Найдите имя загруженного плагина.
  - d. В деталях плагина будет отображена версия(и) пакета, загруженного в платформу.
4. Установите плагин:
  - a. Если у плагина статус «ready», нажмите **Install.**



- b. Некоторые плагины требуют параметров установки; подробности параметров смотрите в документации пользователя конкретного плагина.
- c. Некоторые плагины не требуют параметров установки и начнут установку сразу после нажатия **Install**.

## Обновление плагинов

Чтобы обновить существующий плагин до новой версии:

1. Загрузите новую версию:
  - a. Выполните тот же процесс загрузки новой версии плагина в платформу.
2. Проверьте новую версию:
  - a. Перейдите в **Platform Management > Marketplace > Upload Packages**.
  - b. Переключитесь на вкладку **Cluster Plugin**.
  - c. В деталях пакета соответствующего плагина будет отображена загруженная новая версия.
3. Выполните обновление:
  - a. Перейдите в **Platform Management > Clusters > Clusters**.
  - b. Кластеры с установленными плагинами, которые можно обновить, будут иметь иконку обновления.
  - c. Войдите в детали кластера и переключитесь на вкладку **Features**.
  - d. Кнопка обновления под компонентом features станет активной.
  - e. Нажмите **Upgrade**, чтобы завершить процесс обновления плагина.

# Платформа интеграции сервисов

## Обзор

С ускорением цифровой трансформации предприятия выдвигают всё более высокие требования к унифицированному доступу к разрозненным приложениям и сервисам, централизованному управлению и самостоятельному использованию технологической платформы.

Чтобы решить такие проблемы, как разрозненный доступ к компонентам, несогласованный пользовательский опыт и низкая эффективность эксплуатации и поддержки, была разработана платформа интеграции сервисов.

Она обеспечивает единый каталог сервисов, возможность самостоятельного запроса услуг, управление компонентами и интеграцию функций эксплуатации и обслуживания компонентов, что способствует более эффективному взаимодействию между технологиями и бизнесом.

Основные цели разработки платформы:

- Предоставить единый портал сервисов компонентов для улучшения пользовательского опыта;
- Обеспечить изоляцию ресурсов компонентов между средами, проектами и приложениями;
- Интегрировать разнородные системы эксплуатации и обслуживания компонентов для формирования единой системы управления компонентами;
- Объединить процессы разработки и эксплуатации для повышения эффективности распределения ресурсов и решения проблем;
- Поддерживать экосистему доверенных компонентов и создавать независимые и контролируемые базовые платформенные возможности.

Ценность использования:

- Повышает уровень визуализации компонентов (сервисов) и уровень самообслуживания, снижает порог входа;
- Усиливает возможности управления и эксплуатации компонентов, повышая стабильность платформы;
- Обеспечивает единый вход, унифицированную эксплуатацию и централизованный мониторинг ресурсов компонентов;
- Поддерживает сценарии подключения различных компонентов и поставщиков в будущем, обладает высокой масштабируемостью.

## Возможности

### Функции самообслуживания

Каталог сервисов:

- Отображает все доступные компоненты;

- Предоставляет возможности классификации, поиска и подачи заявок на компоненты.

Инстансы сервисов:

- Поддерживает изолированный просмотр экземпляров компонентов по проектам;
- Отображает информацию об экземплярах: детали, связанные приложения, статус и др.;
- Поддерживает единые операции над экземплярами подключенных компонентов.

Управление заявками:

- Обеспечивает подачу, согласование, просмотр истории и настройку процессов обработки заявок;
- Поддерживает настройку индивидуальных процессов согласования для разных компонентов.

Обратная связь и поддержка:

- Пользователи могут отправлять вопросы и предложения, платформа поддерживает отслеживание обработки запросов и обратную связь;
- Процесс обработки проблем включает отслеживание и фиксацию этапов.

Изоляция ресурсов:

- Обеспечивается на двух уровнях: проект и приложение, с контролем владения ресурсами;
- Приложение — минимальная единица управления ресурсами.

Поддержка информации о приложении:

- Поддерживается хранение основной информации о приложении: сокращённое название, описание, система и др.

### **Функции управления компонентами**

Унифицированная эксплуатация и обслуживание компонентов:

- Поддерживает интеграцию эксплуатации между разными средами и несколькими экземплярами.

Управление средами:

- Поддерживается добавление, изменение и администрирование различных сред внутри организации.

Управление проектами:

- Позволяет вносить и редактировать информацию о проекте: описание, сокращение, ответственный и др.

Аудит процессов:

- Поддерживает настройку этапов согласования, назначение согласующих, конфигурацию процессов и др.

Эксплуатационные возможности платформы:

- Включают мониторинг состояния экземпляров компонентов, аудит пользователей, регулярные проверки, сигналы об отклонениях и др.

Управление доступом:

- Основано на модели RBAC (распределение ролей); поддерживает гибкое назначение прав вплоть до операций с конкретными компонентами.

Системные настройки:

- Поддерживаются глобальные конфигурации: логотип, политика паролей, политика логирования и др.

Мониторинг и оповещения:

- Поддерживает мониторинг компонентов в реальном времени, сбор метрик и настройку уведомления;
- Возможна настройка шаблонов уведомлений.

## Наблюдаемость

Модуль observability — это ориентированная на приложения, готовая к использованию облачно-нативная платформа наблюдаемости, предоставляющая комплексные решения для мониторинга, адаптированные для предприятий. Он обеспечивает мониторинг приложений и их ресурсов в реальном времени, собирая различные метрики, логи и данные событий для анализа состояния здоровья приложений. Этот модуль обладает возможностями оповещений, предлагает всестороннюю визуализацию данных, совместим с основными open-source компонентами и поддерживает быструю локализацию неисправностей и диагностику.

## Преимущества

Основные преимущества модуля observability заключаются в следующем:

- Унифицированный сбор данных

Обеспечивает единый сбор метрик, логов и трассировок, предоставляя комплексный обзор системы и упрощая процессы управления.

- Механизм оповещений

Позволяет настраивать оповещения для метрик и логов, гарантируя своевременное информирование пользователей о потенциальных проблемах.

- Интуитивно понятный интерфейс визуализации

Предоставляет простой и понятный визуальный интерфейс управления, позволяющий пользователям быстро получать ключевую информацию и принимать решения.

- Высокая совместимость

Идеально совместим с основными open-source компонентами, позволяя пользователям легко интегрировать и повышать гибкость системы.

- Готовность к использованию

Предлагает преднастроенные шаблоны и лучшие практики, позволяя пользователям быстро начать работу без сложных настроек.

Основные сценарии использования модуля observability включают:

- Мониторинг микросервисной архитектуры

В облачной микросервисной архитектуре приложения состоят из множества независимых сервисов. Модуль observability может в реальном времени отслеживать производительность и состояние каждого микросервиса, помогая командам разработки быстро выявлять и устранять проблемы с зависимостями между сервисами, обеспечивая стабильность всей системы.

- Мониторинг контейнеров и Kubernetes

Для контейнерных приложений, развернутых в средах Kubernetes, модуль observability может контролировать использование ресурсов, состояние и логи контейнеров, обеспечивая управление жизненным циклом и устранение неполадок контейнеров, гарантируя высокую доступность контейнеризованных приложений.

- Динамическое управление ресурсами

В облачных средах использование ресурсов может изменяться в зависимости от бизнес-требований. Модуль observability способен мониторить использование ресурсов в реальном времени и поддерживать динамическую настройку распределения ресурсов, оптимизируя затраты и производительность.

- Мониторинг мультиоблачных сред

Для предприятий, использующих мультиоблачные стратегии, модуль observability обеспечивает единое управление и мониторинг приложений и ресурсов на разных облачных платформах, обеспечивая видимость и согласованность в мультиоблачных средах.

## **Возможности**

### **Мониторинг**

- Probes

Платформа предоставляет blackbox мониторинг на основе «Blackbox Exporter», позволяя проверять сетевые сервисы через протоколы ICMP, TCP и HTTP. В отличие от whitebox мониторинга, который опирается на внутренние метрики системы, Probe оценивает сервисы извне с точки зрения пользователя, быстро выявляя сбои, влияющие на пользовательский опыт.

- Панель мониторинга

Платформа оснащена функцией управления дашбордами, обеспечивая более удобный интерфейс настройки по сравнению с традиционным Grafana. Предоставляя единый обзор мониторинга, она агрегирует и отображает различные данные метрик, помогая пользователям быстро создавать необходимые дашборды.

### **Оповещения**

- Стратегии оповещений

Платформа предоставляет комплексные возможности оповещений, поддерживая настройку правил оповещений на основе метрик, логов и событий. Благодаря богатому набору встроенных метрик мониторинга и шаблонов оповещений пользователи могут быстро настроить стратегии оповещений, соответствующие требованиям, обеспечивая своевременное обнаружение и решение проблем.

- Шаблоны оповещений

Шаблоны оповещений стандартизируют и инкапсулируют правила оповещений и стратегии уведомлений, поддерживая быстрое повторное использование для множества объектов мониторинга. Конфигурация на основе шаблонов значительно снижает затраты на управление стратегиями оповещений и повышает эффективность эксплуатации.

- История оповещений

Система полностью фиксирует жизненный цикл оповещений, включая время срабатывания, время восстановления, статус оповещения, уровень и содержание

оповещения. Пользователи могут отслеживать и анализировать проблемы через историю оповещений, постоянно оптимизируя настройки оповещений.

- Уведомления

Платформа поддерживает несколько каналов оповещений, включая email и Webhook, обеспечивая своевременную доставку информации ответственным лицам. Пользователи могут гибко настраивать стратегии уведомлений в соответствии с реальными потребностями.

## **Трассировка**

Распределённое трассирование предоставляет возможности полного трассирования цепочек вызовов в микросервисной архитектуре. Сбор метаданных межсервисных вызовов помогает пользователям быстро локализовать проблемы в кросс-сервисных взаимодействиях.

## **Логи**

Платформа автоматически собирает и централизованно управляет стандартным выводом и файловыми логами с кластеров, узлов и контейнеров. Она обеспечивает мощные возможности хранения, поиска и анализа логов, поддерживает многомерные запросы и визуализацию логов, помогая пользователям быстро выявлять проблемы.

## **События**

Платформа в реальном времени собирает критическую информацию о событиях из Kubernetes кластеров, фиксируя полный процесс изменения состояния ресурсов. При возникновении исключений в кластерах, узлах, Pods и т.д. события позволяют проследить причины и значительно повысить эффективность решения проблем.

## **Состояние платформы**

- Инспектирование

Опираясь на обширный опыт эксплуатации, платформа предлагает возможности автоматизированной инспекции компонентов. Посредством многомерных проверок состояния она помогает пользователям в реальном времени контролировать работоспособность ресурсов, выявлять потенциальные риски на ранних стадиях и снижать затраты на ручные проверки.

- Состояние здоровья платформы

Предоставляется интуитивный обзор функционального состояния платформы, поддерживающий просмотр условий развертывания и статусов работы компонентов. Пользователи с правами управления платформой могут углубиться в подробные данные проверок здоровья, быстро находя и устраняя проблемы на уровне платформы.

## **Мониторинг**

Модуль мониторинга предоставляет операционные возможности, такие как метрики, дашборды, оповещения и уведомления для пользователей платформы.

Платформа интегрирует открытые компоненты, такие как Prometheus / VictoriaMetrics и панели мониторинга, обеспечивая мониторинг в реальном времени кластеров, узлов, компонентов, пользовательских приложений, Pod'ов, контейнеров и других объектов, управляемых платформой.

Поддерживается быстрая настройка оповещений по метрикам мониторинга на уровне кластера, узла и вычислительных компонентов, оповещений по логам и оповещений по событиям. Кроме того, возможно создание пользовательских алгоритмов мониторинга метрик на основе реальных требований, что позволяет увеличить количество необходимых метрик и правил оповещений. Стратегии уведомлений можно настроить для своевременной отправки информации об оповещениях операционному персоналу, что помогает избежать сбоев системы или оперативно решать возникающие проблемы, снижая затраты на эксплуатацию и обеспечивая стабильность системы.

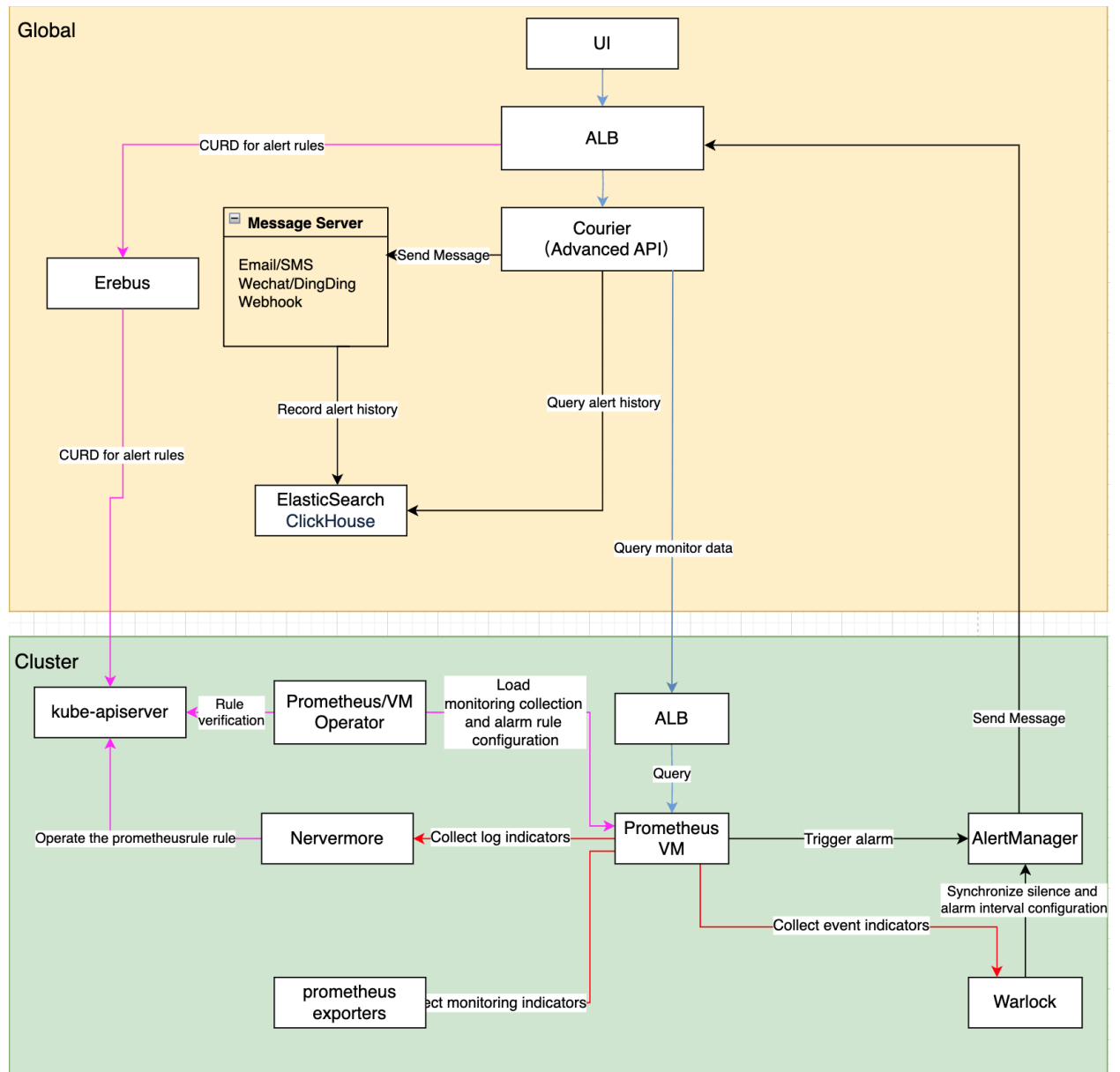
### **Ограничения использования**

При использовании модуля мониторинга обратите внимание на следующие ограничения:

- Срок хранения данных мониторинга зависит от настроек емкости хранилища, по умолчанию период хранения составляет 7 дней
- Prometheus и VictoriaMetrics не могут быть установлены одновременно в одном кластере, необходимо выбрать один из них для установки
- Минимальный поддерживаемый интервал сбора пользовательских метрик мониторинга составляет 60 секунд
- Для каналов уведомлений об оповещениях должны быть предварительно настроены соответствующие сервисы (например, почтовые серверы, SMS-шлюзы, боты и др.)



## Архитектура



Система мониторинга состоит из следующих основных функциональных модулей:

1. Система мониторинга
  - a. Сбор и хранение данных: сбор и сохранение метрик мониторинга из различных источников
  - b. Запрос и визуализация данных: предоставление гибких возможностей для запроса и визуализации данных мониторинга
2. Система оповещений
  - a. Управление правилами оповещений: настройка и управление политиками оповещений
  - b. Генерация оповещений и уведомлений: оценка правил оповещений и отправка уведомлений

- c. Статус оповещений в реальном времени: предоставление актуального статуса оповещений системы

### 3. Система уведомлений

- a. Конфигурация уведомлений: управление шаблонами уведомлений, группами контактов и политиками
- b. Сервер уведомлений: управление конфигурацией различных каналов уведомлений

## **Руководство по выбору системы мониторинга**

При установке мониторинга кластера платформа предоставляет два компонента мониторинга на выбор: VictoriaMetrics и Prometheus. Обратите внимание:

- При установке компонентов мониторинга кластера можно выбрать только один из VictoriaMetrics или Prometheus
- VictoriaMetrics подходит для сценариев с требованиями высокой доступности и мониторинга нескольких кластеров
- Prometheus подходит для сценариев мониторинга одного кластера, особенно небольшого масштаба

### **Сценарии, подходящие для использования VictoriaMetrics**

- Потребности в высокой производительности и масштабируемости: подходит для сценариев мониторинга с высокой пропускной способностью данных и долгосрочным хранением
- Управление несколькими кластерами: требуется единое управление данными мониторинга нескольких кластеров

### **Сценарии, подходящие для использования Prometheus**

- Один кластер небольшого масштаба: небольшой масштаб мониторинга без требований к высокой доступности
- Пользователи с существующим Prometheus: уже имеется полноценная система мониторинга Prometheus
- Простые требования к стабильности: стремление к простому и надёжному решению мониторинга
- Глубокая интеграция с экосистемой: тесная интеграция с экосистемой Prometheus, высокая стоимость миграции

## **Установка**

Компонент мониторинга служит инфраструктурой для функций мониторинга, оповещений, инспекции и проверки состояния в модуле наблюдаемости. В этом документе описывается, как установить MWS CP Monitoring с плагином Prometheus или MWS CP Monitoring с плагином VictoriaMetrics в кластере.

### **Установка MWS CP Monitoring с плагином Prometheus**

1. Перейдите в **App Store Management > Cluster Plugins** и выберите целевой кластер.
2. Найдите плагин MWS Monitoring with Prometheus и нажмите **Install**.
3. Настройте следующие параметры:

| Параметр                | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scale Configuration     | Поддерживает три конфигурации: Small Scale, Medium Scale и Large Scale: <ol style="list-style-type: none"> <li>1. Значения по умолчанию установлены на основе рекомендованных нагрузочных тестов платформы</li> <li>2. Можно выбрать или настроить квоты в зависимости от фактического масштаба кластера</li> <li>3. Значения по умолчанию будут обновляться с версиями платформы; для фиксированных конфигураций рекомендуется использовать пользовательские настройки</li> </ol> |
| Storage Type            | <ul style="list-style-type: none"> <li>• LocalVolume: Локальное хранилище с данными, расположенными на указанных узлах</li> <li>• StorageClass: Автоматически создаёт persistent volume с использованием класса хранения</li> <li>• PV: Использует существующие persistent volume</li> </ul> <b>Примечание:</b> конфигурация хранилища не может быть изменена после установки                                                                                                      |
| Replica Count           | Задаёт количество подов компонента мониторинга<br><b>Примечание:</b> Prometheus поддерживает только установку на одном узле                                                                                                                                                                                                                                                                                                                                                        |
| Parameter Configuration | Параметры данных для компонента мониторинга можно при необходимости настроить                                                                                                                                                                                                                                                                                                                                                                                                      |

4. Нажмите **Install** для завершения установки

## Доступ

После завершения установки компоненты доступны по следующим адресам (**замените <> на актуальные значения**):

| Компонент    | Адрес доступа                                             |
|--------------|-----------------------------------------------------------|
| Thanos       | <platform_access_address>/clusters/<cluster>/prometheus   |
| Prometheus   | <platform_access_address>/clusters/<cluster>/prometheus-0 |
| Alertmanager | <platform_access_address>/clusters/<cluster>/alertmanager |

## Установка MWS CP Monitoring с плагином VictoriaMetrics

1. Перейдите в **App Store Management > Cluster Plugins** и выберите целевой кластер.
2. Найдите плагин MWS Monitoring with VictoriaMetrics и нажмите **Install**.
3. Настройте следующие параметры:

| Параметр                | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scale Configuration     | Поддерживает три конфигурации: Small Scale, Medium Scale и Large Scale: <ul style="list-style-type: none"> <li>• Значения по умолчанию установлены на основе рекомендованных нагрузочных тестов платформы</li> <li>• Можно выбрать или настроить квоты в зависимости от фактического масштаба кластера</li> <li>• Значения по умолчанию будут обновляться с версиями платформы; для фиксированных конфигураций рекомендуется использовать пользовательские настройки</li> </ul> |
| Install Agent Only      | <b>Off:</b> Установить полный набор компонентов VictoriaMetrics<br><b>On:</b> Установить только компонент сбора VMAgent, который зависит от VictoriaMetrics Center                                                                                                                                                                                                                                                                                                              |
| VictoriaMetrics Center  | Выберите кластер, в котором установлен полный набор компонентов VictoriaMetrics                                                                                                                                                                                                                                                                                                                                                                                                 |
| Storage Type            | <ul style="list-style-type: none"> <li>• LocalVolume: Локальное хранилище с данными, расположенными на указанных узлах</li> <li>• StorageClass: Автоматически создаёт persistent volume с использованием класса хранения</li> <li>• PV: Использует существующие persistent volume</li> </ul>                                                                                                                                                                                    |
| Replica Count           | Задаёт количество подов компонента мониторинга:<br>Тип хранилища LocalVolume не поддерживает несколько реплик<br>Для других типов хранилища следуйте подсказкам на экране для настройки                                                                                                                                                                                                                                                                                         |
| Parameter Configuration | Параметры данных для компонента мониторинга можно настроить                                                                                                                                                                                                                                                                                                                                                                                                                     |

4. Нажмите Install для завершения установки

### Управление сбором метрик

Помимо встроенных метрик платформы, вы также можете интегрировать метрики, предоставляемые вашими приложениями или сторонними приложениями через ServiceMonitor или PodMonitor. В этом разделе в качестве примера используется Elasticsearch Exporter, установленный в виде pod в том же кластере.

**Создайте Service/Endpoint для Exporter, чтобы открыть метрики:**

```
apiVersion: v1
kind: Service
metadata:
 labels:
 chart: elasticsearch
 service_name: cpaas-elasticsearch
```

```
name: cpaas-elasticsearch
namespace: cpaas-system
spec:
 clusterIP: 10.105.125.99
 ports:
 - name: cpaas-elasticsearch
 port: 9200
 protocol: TCP
 targetPort: 9200
 selector:
 service_name: cpaas-elasticsearch
 sessionAffinity: None
 type: ClusterIP
```

**Создайте объект ServiceMonitor для описания метрик, предоставляемых вашим приложением:**

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
 labels:
 app: cpaas-monitor
 chart: cpaas-monitor
 heritage: Helm
 prometheus: kube-prometheus (1)
 release: cpaas-monitor
 name: cpaas-elasticsearch-Exporter
 namespace: cpaas-system (2)
spec:
 jobLabel: service_name (3)
 namespaceSelector: (4)
 any: true
 selector: (5)
 matchExpressions:
 - key: service_name
 operator: Exists
 endpoints:
 - port: cpaas-elasticsearch (6)
 path: /_prometheus/metrics (7)
 interval: 60s (8)
 honorLabels: true
 basicAuth: (9)
 password:
 key: ES_PASSWORD
 name: acp-config-secret
 username:
 key: ES_USER
 name: acp-config-secret
```

- (1) К какому Prometheus должен быть синхронизирован ServiceMonitor; оператор будет слушать соответствующий ресурс ServiceMonitor на основе конфигурации serviceMonitorSelector в Prometheus CR. Если метки ServiceMonitor не совпадают с конфигурацией serviceMonitorSelector в Prometheus CR, этот ServiceMonitor не будет мониториться оператором.
- (2) В каких namespace оператор будет слушать ServiceMonitor на основе конфигурации serviceMonitorNamespaceSelector в Prometheus CR; если ServiceMonitor не находится в serviceMonitorNamespaceSelector Prometheus CR, этот ServiceMonitor не будет мониториться оператором.
- (3) Метрики, собираемые Prometheus, будут иметь метку job, значение которой соответствует значению метки сервиса, указанной в jobLabel.
- (4) ServiceMonitor сопоставляется с соответствующим Service на основе конфигурации namespaceSelector.
- (5) ServiceMonitor сопоставляется с Service на основе конфигурации selector.
- (6) ServiceMonitor сопоставляется с портом Service на основе конфигурации port.
- (7) Путь доступа к Exporter, по умолчанию /metrics.
- (8) Интервал, с которым Prometheus опрашивает метрики Exporter.
- (9) Если для доступа к пути Exporter требуется аутентификация, необходимо добавить данные аутентификации; также поддерживается bearer token, tls-аутентификация и другие методы.

#### **Проверьте, мониторится ли ServiceMonitor Prometheus:**

Зайдите в UI компонента мониторинга и проверьте, существует ли задача cpaas-elasticsearch-exporter.

- Адрес UI Prometheus: <https://<your platform access address>/clusters/<cluster name>/prometheus-0/targets>
- Адрес UI VictoriaMetrics: <https://<your platform access address>/clusters/<cluster name>/vmselect/vmui/?#/metrics>

#### **Управление оповещениями**

Функция управления оповещениями платформы предназначена для помощи пользователям в комплексном мониторинге и своевременном обнаружении аномалий системы. Используя предустановленные системные оповещения и гибкие возможности создания пользовательских оповещений, в сочетании со стандартизированными шаблонами оповещений и многоуровневым механизмом управления, она предоставляет полное решение по оповещениям для операционного и технического персонала.

Как администраторы платформы, так и пользователи могут удобно настраивать и управлять политиками оповещений в рамках своих полномочий для эффективного мониторинга ресурсов платформы.

#### **Создание**

1. Перейдите в **Центр эксплуатации и обслуживания > Alerts > Alert Policies**.

2. Нажмите **Create Alert Policy**.
3. Настройте основную информацию.

#### Тип оповещения

- Оповещение по ресурсу
  - Типы оповещений классифицированы по типу ресурса (например, состояние deployment в namespace)
  - Описание выбора ресурса:
    - По умолчанию "Any", если параметр не выбран, поддерживается автоматическая привязка к вновь добавленным ресурсам
    - При выборе "Select All" применяется только к текущему ресурсу
    - При выборе нескольких namespace имена ресурсов поддерживают регулярные выражения (например, cert.\*)
- Оповещение по событию
  - Типы оповещений классифицированы по конкретным событиям (например, аномальное состояние Pod)
  - По умолчанию выбираются все ресурсы под указанным ресурсом и поддерживается автоматическая привязка к вновь добавленным ресурсам

#### Настройка правил оповещений

Нажмите **Add Alert Rule** и настройте параметры в зависимости от типа оповещения:

- Параметры оповещения по ресурсу:

| Параметр          | Описание                                                                                                                                                                                                                                                                                   |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Expression        | Алгоритм метрики мониторинга в формате Prometheus, например, <code>rate(node_network_receive_bytes{instance="\$server",device!~"lo"}[5m])</code>                                                                                                                                           |
| Metric Unit       | Единица измерения пользовательской метрики мониторинга, можно ввести вручную или выбрать из предустановленных на платформе единиц                                                                                                                                                          |
| Legend Parameter  | Управляет названием, соответствующим кривой на графике, формат <code>{{.LabelName}}</code> , например, <code>{{.hostname}}</code>                                                                                                                                                          |
| Time Range        | Временной интервал для запросов логов/событий                                                                                                                                                                                                                                              |
| Log Content       | Поля запроса для содержимого логов (например, Error), несколько полей связаны через OR                                                                                                                                                                                                     |
| Event Reason      | Поля запроса для причин событий (Reason, например, BackOff, Pulling, Failed и т.д.), несколько полей связаны через OR                                                                                                                                                                      |
| Trigger Condition | Условие, состоящее из операторов сравнения, порогов оповещения и длительности (опционально). Определяет, срабатывает ли оповещение на основе сравнения текущих значений/количества логов/событий с порогом оповещения, а также длительности нахождения текущих значений в пределах порога. |

|             |                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alert Level | Делится на четыре уровня: Critical, Serious, Warning и Info. Можно установить разумный уровень оповещения в зависимости от влияния правил оповещения на бизнес для соответствующих ресурсов. |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Параметры оповещения по событию:

| Параметр              | Описание                                                                                     |
|-----------------------|----------------------------------------------------------------------------------------------|
| Time Range            | Временной интервал для запросов событий                                                      |
| Event Monitoring Item | Поддерживает мониторинг уровней событий или причин событий, несколько полей связаны через OR |
| Trigger Condition     | Основано на количестве событий для оценки сравнения                                          |
| Alert Level           | Определение такое же, как для уровней оповещений по ресурсам                                 |

### Другие настройки

1. Выберите одну или несколько созданных политик уведомлений
2. Настройте интервалы отправки оповещений:
  - a. Global: использовать конфигурацию по умолчанию платформы
  - b. Custom: можно задать разные интервалы отправки в зависимости от уровней оповещений
  - c. При выборе "Do Not Repeat" уведомления отправляются только при срабатывании и восстановлении оповещения

### Особенности

В разделе "More" правила оповещения можно задать labels и annotations.

Пожалуйста, обратитесь к Prometheus Alerting Rules Documentation для настройки labels и annotations.

**Внимание:** Не используйте переменную \$value в labels, так как это может вызвать исключения оповещений.

### Управление уведомлениями

С помощью уведомлений вы можете интегрировать функции мониторинга и оповещения платформы для своевременной отправки информации о предварительном предупреждении получателям уведомлений, напоминая соответствующему персоналу принять необходимые меры для решения проблем или предотвращения сбоев.

### Notification Server

Сервер уведомлений предоставляет сервисы для отправки сообщений уведомлений получателям на платформе. В настоящее время платформа поддерживает следующие серверы уведомлений:

- Email Server: отправляет уведомления по электронной почте через email-сервер
- Webhook Type Server: поддерживает отправку WebHook на указанный сервер



## Email Server

1. В левой навигационной панели нажмите **Platform Settings > Notification Server**.
2. Нажмите Configure Now.
3. Следуйте инструкциям ниже для настройки соответствующих параметров:

| Параметр              | Описание                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service Address       | Адрес сервера уведомлений, поддерживающего протокол SMTP, например, smtp.yeah.net.                                                                                                                                                                                                                                                                                                                                        |
| Port                  | Номер порта сервера уведомлений. При включённом Use SSL необходимо указать порт SSL.                                                                                                                                                                                                                                                                                                                                      |
| Server Configuration  | <b>Use SSL:</b> Secure Socket Layer (SSL) — стандартная технология безопасности. Переключатель SSL управляет установкой зашифрованного соединения между сервером и клиентом.<br><b>Skip Insecure Verification:</b> переключатель insecureSkipVerify управляет проверкой сертификата клиента и имени хоста сервера. Если включён, сертификаты и соответствие имени хоста в сертификате имени сервера проверяться не будут. |
| Sender Email          | Email-аккаунт отправителя на сервере уведомлений, используемый для отправки уведомлений по электронной почте.                                                                                                                                                                                                                                                                                                             |
| Enable Authentication | Если требуется аутентификация, настройте имя пользователя и код авторизации для email-сервера.                                                                                                                                                                                                                                                                                                                            |

4. Нажмите OK.

## Webhook Type Server

1. В левой навигационной панели нажмите **Cluster Management > Cluster**.
2. Нажмите кнопку действий рядом с кластером global > **CLI Tool**.
3. Выполните следующую команду на мастер-ноде кластера global:

```
kubectrl patch secret -n cpaas-system platform-webhook-server -p
'{"data":{"enable":"dHJ1ZQo="}}'
```

**Подсказка:** «dHJ1ZQo=» — это значение true, закодированное в base64; чтобы отключить, замените «dHJ1ZQo=» на «ZmFsc2UK»(без кавычек), что является значением false в base64.

## Notification Contact Group

Контактная группа уведомлений — это набор получателей уведомлений с похожими логическими характеристиками. Например, вы можете назначить команду эксплуатации и обслуживания в качестве контактной группы уведомлений для удобного выбора и управления при настройке стратегий уведомлений.

### Настройка

1. В левой навигационной панели нажмите **Operations Center > Notifications**.
2. Перейдите на вкладку **Notification Contact Group**.

3. Нажмите **Create Notification Contact Group** и настройте соответствующие параметры согласно приведённым ниже инструкциям:

| Параметр              | Описание                                                                                                                                                                                                                                |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Email                 | Добавьте email для всей контактной группы уведомлений. Платформа будет отправлять уведомления на этот email и на email всех контактов в группе.                                                                                         |
| Webhook URL           | Укажите соответствующий URL метода уведомления в зависимости от настроенного сервера уведомлений. После настройки контакты в этой группе будут уведомляться этим способом.                                                              |
| Contact Configuration | Нажмите Add Contact, чтобы добавить существующих пользователей платформы в контактную группу. Убедитесь в корректности контактных данных выбранных контактов (телефон, email, callback интерфейс), чтобы избежать пропуска уведомлений. |

4. Нажмите **Add**.

### Notification Template

Шаблон уведомления — это стандартизированная структура, состоящая из настраиваемого содержимого, переменных содержимого и параметров форматирования. Он используется для стандартизации содержания и формата сообщений оповещений для стратегий уведомлений.

### Настройка

1. В левой навигационной панели нажмите **Operations Center > Notifications**.
2. Перейдите на вкладку **Notification Template**.
3. Нажмите **Create Notification Template**.
4. В разделе **Basic Information** настройте следующие параметры:

| Параметр     | Описание                                                                                                                                                                                                                                                                                                                                                            |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Message Type | Выберите тип сообщения в зависимости от цели уведомления. <ul style="list-style-type: none"><li>• <b>Alert Message</b>: отправляет сообщения оповещений, вызванных правилами оповещения, в сочетании с функцией оповещения платформы</li><li>• <b>Component Exception Message</b>: отправляет уведомления, вызванные исключениями в отдельных компонентах</li></ul> |

5. В разделе **Template Configuration** настройте переменные и параметры форматирования содержимого, ориентируясь на различные типы шаблонов.

Содержимое шаблона может состоять только из переменных, отображаемых имён переменных и специального языка разметки форматирования, поддерживаемого платформой. Переменные и другие элементы можно свободно комбинировать при соблюдении синтаксических правил.

В шаблоне можно использовать только переменные, поддерживаемые платформой. Вы можете изменять отображаемые имена переменных и формат содержимого, но не можете изменять саму переменную. См. ниже в разделе «Переменные».

Платформа предоставляет стандартное содержимое шаблонов уведомлений для различных типов уведомлений на основе реальных сценариев эксплуатации, что покрывает большинство потребностей в настройке сообщений уведомлений. При отсутствии особых требований можно использовать содержимое шаблонов по умолчанию.

6. Нажмите **Create**.

## Переменные

Переменные — это ключи меток или аннотаций в сообщениях уведомлений (NotificationMessage), оформленные как {{.labelKey}}. Для удобства быстрого получения ключевой информации пользователям можно назначать переменным настраиваемые отображаемые имена; например:

Alert Level: {{.externalLabels.severity }}.

Платформа по умолчанию предоставляет следующие базовые переменные:

| Отображаемое имя  | Переменная                                                      | Описание                                                                   |
|-------------------|-----------------------------------------------------------------|----------------------------------------------------------------------------|
| Alert Status      | {{ .externalLabels.status }}                                    | Например: Alerting.                                                        |
| Alert Level       | {{ .externalLabels.severity }}                                  | Например: Critical.                                                        |
| Alert Cluster     | {{ .labels.alert_cluster }}                                     | Например: кластер 1, в котором произошло оповещение.                       |
| Alert Object      | {{ .externalLabels.object }}                                    | Тип и имя ресурса, где произошло оповещение, например, node 192.168.16.53. |
| rule Name         | {{ .labels.alert_resource }}                                    | Имя правила оповещения, например, cpaas-node-rules.                        |
| Alert Description | {{ .externalLabels.summary }}                                   | Описание правила оповещения.                                               |
| Trigger Value     | {{ .externalLabels.currentValue }}                              | Значение, вызвавшее оповещение.                                            |
| Alert Time        | {{ dateFormatWithZone "2006-01-02 15:04:05" "Asia/Chongqing" }} | Время начала оповещения.                                                   |
| Recovery Time     | {{ dateFormatWithZone "2006-01-02 15:04:05" "Asia/Chongqing" }} | Время окончания оповещения.                                                |
| Metric Name       | {{ .labels.alert_indicator }}                                   | Название метрики мониторинга.                                              |

## Notification rule

Правило уведомления — это набор правил, определяющих, как отправлять сообщения уведомлений конкретным контактам.

### Настройка

1. В левой навигационной панели нажмите **Operations Center > Notifications**.
2. Нажмите **Create Notification rule** и настройте соответствующие параметры согласно следующим инструкциям:

| Параметр                   | Описание                                                                                                                                                         |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Notification Contact Group | Контактная группа уведомлений — логический набор получателей уведомлений, которых платформа будет уведомлять указанным способом.                                 |
| Notification Recipients    | Выберите одного или нескольких получателей уведомлений, и платформа будет отправлять уведомления согласно контактными данным в Personal Information получателей. |
| Notification Method        | Поддерживает: WebHook URL                                                                                                                                        |
| Notification Template      | Выберите шаблон уведомления для отображения информации уведомления.                                                                                              |

3. Нажмите **Create**.

## Управление отображением данных мониторинга

Платформа предоставляет мощный функционал управления отображением мониторинга, предназначенный для замены традиционных инструментов Grafana, предлагая пользователям более комплексный и гибкий опыт.

### Основные возможности

- Поддержка настройки пользовательского отображения метрик как для пользовательских, так и для платформенных нагрузок
- Возможность просмотра публичных дашбордов, настроенных в платформенных областях. Пользовательские области можно просмотреть с изоляцией данных по namespace, к которому принадлежит нагрузка
- Поддержка управления панелями: добавление, удаление, изменение панелей, масштабирование панелей и перемещение панелей с помощью drag-and-drop
- Возможность настройки пользовательских переменных внутри дашбордов для фильтрации данных запросов
- Поддержка настройки групп внутри дашбордов. Группы могут отображаться повторно на основе пользовательских переменных
- Поддерживаемые типы панелей: trend, step line chart, bar chart, horizontal bar chart, bar gauge chart, gauge chart, table, stat chart, XY chart, pie chart, text
- Функция однократного импорта панелей Grafana

## Особенности

В настоящее время панели мониторинга поддерживают только просмотр данных мониторинга, собранных компонентами мониторинга, установленными в платформе.

## Создание дашборда

1. Нажмите **Create Dashboard**, руководствуясь следующими инструкциями для настройки соответствующих параметров:

| Параметр              | Описание                                                                                                                                                                                             |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Folder                | Папка, в которой находится панель мониторинга; можно ввести или выбрать существующую папку.                                                                                                          |
| Label                 | Метка панели мониторинга; позволяет быстро находить существующие панели по фильтру меток при переключении.                                                                                           |
| Set as Main Dashboard | Если включено, текущая панель будет установлена как основная после успешного создания; при повторном входе в функцию панели мониторинга по умолчанию будет отображаться основная панель.             |
| Variables             | Добавьте переменные при создании панели для использования в качестве параметров метрик в добавленных панелях, которые также могут использоваться как фильтры на главной странице панели мониторинга. |

2. После добавления нажмите **Create** для завершения создания панели. Далее необходимо добавить переменные, добавить панели и добавить группы для завершения общей компоновки.

## Импорт дашборда

Платформа поддерживает прямой импорт JSON из Grafana для преобразования его в дашборд платформы для отображения.

В настоящее время поддерживается только Grafana JSON версии V8 и выше

Если какие-либо панели в импортированной панели не входят в поддерживаемый платформой диапазон, они могут отображаться как `unsupported panel types`, но вы можете изменить настройки панели для нормального отображения

После импорта дашборда вы можете выполнять любые операции управления, как обычно, без отличий от панелей, созданных в платформе

## Другие операции

Вы можете нажать кнопку операций справа на странице панели для выполнения нужных действий с панелью:

| Операция | Описание                                                                                                                                              |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| YAML     | Открывает фактический код ресурса CR панели, хранящийся в Kubernetes кластере. Вы можете изменить всё содержимое панели, редактируя параметры в YAML. |

| Операция          | Описание                                                                                                             |
|-------------------|----------------------------------------------------------------------------------------------------------------------|
| Export Expression | Позволяет экспортировать метрики и соответствующие выражения запросов, используемые в текущей панели, в формате CSV. |
| Copy              | Копирует текущую панель; вы можете редактировать панели по необходимости и сохранить как новую панель.               |
| Settings          | Изменяет основную информацию текущей панели, например, смену меток и добавление дополнительных переменных.           |
| Delete            | Удаляет текущую панель мониторинга.                                                                                  |

## Управление Probe

Функция probe платформы реализована на основе Blackbox Exporter, позволяя пользователям выполнять проверку сети через ICMP, TCP или HTTP для быстрого выявления сбоев, происходящих на платформе.

## Настройка

1. В левой навигационной панели нажмите **Operations Center > Monitoring > Blackbox Monitoring**.

Совет: Blackbox мониторинг является функцией на уровне кластера. Для переключения между кластерами используйте верхнюю панель навигации.

2. Нажмите **Create Blackbox Monitoring Item**.
3. Следуйте инструкциям ниже для настройки соответствующих параметров:

| Параметр          | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Метод проверки    | <p><b>ICMP:</b> проверяет доступность сервера, отправляя ping на доменное имя или IP-адрес, введенный в поле Target Address.</p> <p><b>TCP:</b> проверяет бизнес-порт хоста, слушая &lt;domain:port&gt; или &lt;IP:port&gt;, указанный в Target Address.</p> <p><b>HTTP:</b> проверяет URL, введенный в Target Address, для проверки доступности сайта.</p> <p><b>Совет:</b> Метод HTTP по умолчанию поддерживает только GET-запросы</p> |
| Интервал проверки | Интервал времени между проверками.                                                                                                                                                                                                                                                                                                                                                                                                       |
| Целевой адрес     | <p>Целевой адрес для проверки, максимум 128 символов. Формат ввода зависит от метода проверки:</p> <p><b>ICMP:</b> доменное имя или IP-адрес, например, 192.168.0.1.</p> <p><b>TCP:</b> &lt;domain:port&gt; или &lt;IP:port&gt;, например, 192.168.0.1:8080.</p> <p><b>HTTP:</b> URL, начинающийся с http или https, например, https://mws.ru/.</p>                                                                                      |

4. Нажмите **Create**.

После успешного создания вы можете в реальном времени просматривать последние результаты проверки на странице списка, а также на основе элементов blackbox мониторинга создавать политики оповещений.

После успешного создания элементов blackbox мониторинга системе требуется около 5 минут для синхронизации конфигурации. В течение этого времени проверки не выполняются, и результаты проверки недоступны.

### **Создание элементов Blackbox мониторинга и оповещений через CLI**

1. Создайте новый YAML-файл конфигурации с именем example-probe.yaml.
2. Добавьте ресурс PrometheusRule в YAML-файл и отправьте его.

Пример создаёт новую политику оповещений с именем prometheus-liveness:

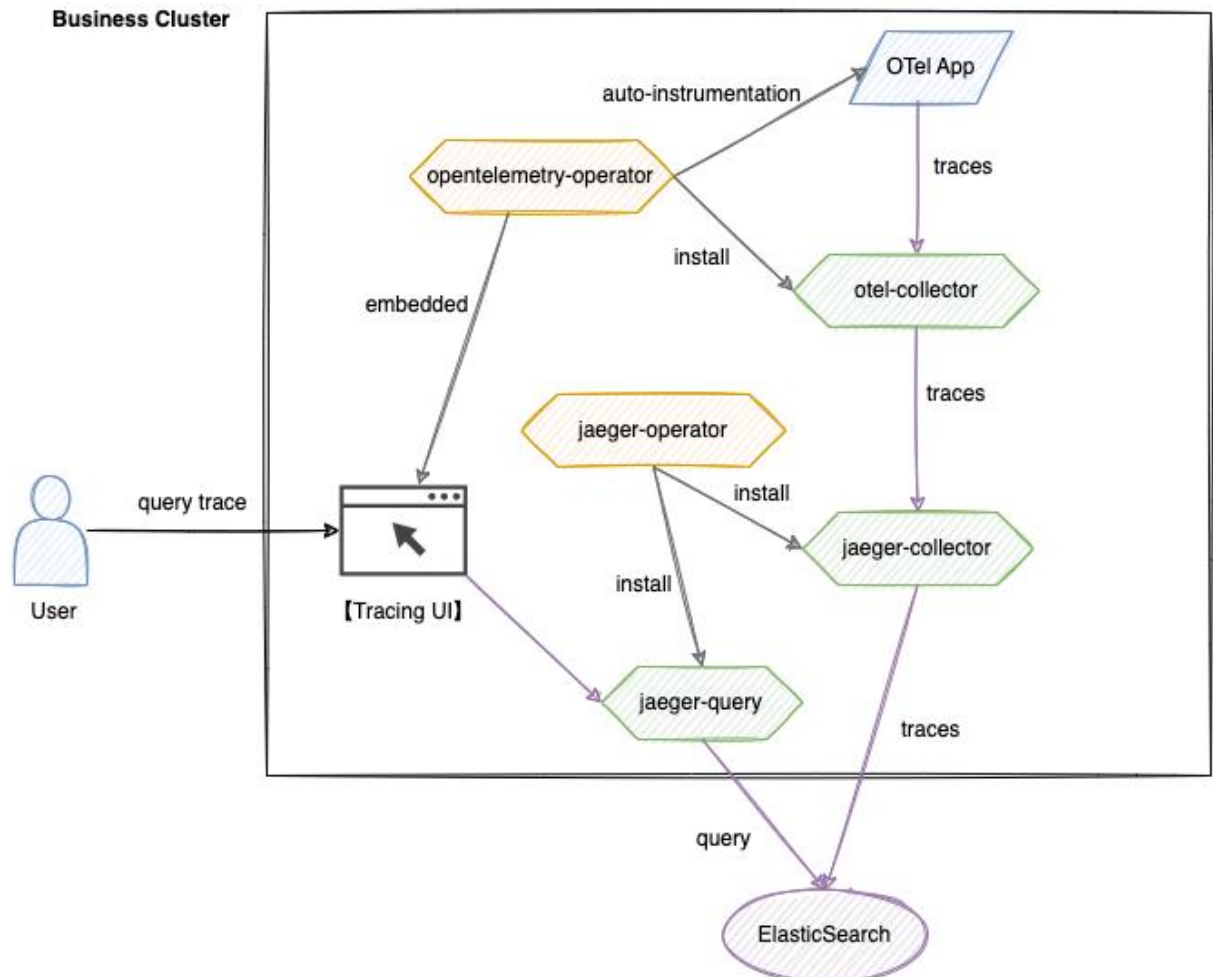
```
apiVersion: monitoring.coreos.com/v1
kind: Probe
metadata:
 labels:
 prometheus: kube-prometheus # Метка, используемая для имени prometheus
 name: prometheus-liveness # Имя элемента probe
 namespace: cpaas-system # Namespace, используемый для namespace prometheus
spec:
 jobName: prometheus-liveness # Имя элемента probe
 prober:
 url: cpaas-monitor-prometheus-blackbox-exporter:9115 # URL для метрик Blackbox,
 полученный из features
 module: http_2xx # Имя модуля элемента probe
 targets:
 staticConfig:
 static:
 - https://mws.ru # Целевой адрес элемента probe
 labels:
 module: http_2xx # Имя модуля элемента probe
 prober: http # Метод проверки элемента probe
 interval: 30s # Интервал проверки элемента probe
 scrapeTimeout: 10s
```

### **Трассировка**

Distributed Tracing — ключевой модуль в системе наблюдаемости контейнерных платформ, используемый для сквозного трассирования и анализа распределённых систем. Этот модуль построен на основе стандарта OpenTelemetry (OTel), предоставляя комплексное решение от сбора данных и их хранения до визуального анализа, помогая разработчикам и операционному персоналу быстро выявлять аномалии вызовов сервисов, анализировать узкие места в производительности и отслеживать поведение запросов на протяжении всего их жизненного цикла.

## Архитектура

Данная архитектура построена на стеке технологий OpenTelemetry и Jaeger, обеспечивая полный жизненный цикл управления распределённым трассированием. Система состоит из пяти основных модулей: сбор данных, передача, хранение, запросы и визуализация.



## Особенности

В условиях высокой нагрузки сбор трассировочных данных может оказывать определённое давление на производительность и хранение в Elasticsearch; рекомендуется разумно настраивать коэффициент сэмплирования в зависимости от бизнес-условий.

Компоненты Tracing и Service Mesh являются взаимоисключающими. Если вы уже развернули компонент Service Mesh, пожалуйста, сначала удалите его.

## Установка

Перед началом установки убедитесь, что компонент Elasticsearch настроен для хранения данных трассировки, включая URL доступа и информацию Basic Auth.

### Установка Jaeger Operator через Веб-консоль

Вы можете установить Jaeger Operator из раздела **Marketplace > OperatorHub** платформы MWS Container Platform, где перечислены доступные операторы.



1. В представлении **Platform Management** веб-консоли выберите кластер, в который хотите развернуть Jaeger Operator, затем перейдите в **Marketplace > OperatorHub**.
2. В строке поиска найдите MWS build of Jaeger в каталоге. Нажмите на заголовок MWS build of Jaeger.
3. Ознакомьтесь с вводной информацией об операторе на странице MWS build of Jaeger. Нажмите **Install**.
4. На странице **Install**:
  1. Выберите **Manual** для параметра **Upgrade Strategy**. При стратегии одобрения **Manual** OLM создаст запросы на обновление. Как администратор кластера, вы должны вручную одобрять запросы OLM для обновления оператора до новой версии.
  2. Выберите канал **stable (Default)**.
  3. Выберите **Recommended** для **Installation Location**. Установите оператор в рекомендуемом пространстве имен `jaeger-operator`, чтобы оператор мог мониторить и быть доступным во всех пространствах имен кластера.
5. Нажмите **Install**.
6. Убедитесь, что в поле **Status** отображается **Succeeded**, что подтверждает успешную установку Jaeger Operator.

7. Проверьте, что все компоненты Jaeger Operator были успешно установлены. Выполните команду:

```
kubectl -n jaeger-operator get csv
```

Пример вывода

| NAME                   | DISPLAY         | VERSION | REPLACES | PHASE     |
|------------------------|-----------------|---------|----------|-----------|
| jaeger-operator.vx.x.0 | Jaeger Operator | x.x.0   |          | Succeeded |

Если в поле PHASE указано Succeeded, значит оператор и его компоненты установлены успешно.

### Установка OpenTelemetry Operator через Веб-консоль

Вы можете установить OpenTelemetry Operator из раздела **Marketplace > OperatorHub** платформы MWS Container Platform, где перечислены доступные операторы.

1. В представлении **Platform Management** веб-консоли выберите кластер, в который хотите развернуть OpenTelemetry Operator, затем перейдите в **Marketplace > OperatorHub**.
2. В строке поиска найдите MWS build of OpenTelemetry в каталоге. Нажмите на заголовок MWS build of OpenTelemetry.
3. Ознакомьтесь с вводной информацией об операторе на странице MWS build of OpenTelemetry. Нажмите **Install**.
4. На странице **Install**:
  - a. Выберите **Manual** для параметра **Upgrade Strategy**. При стратегии одобрения **Manual** OLM создаст запросы на обновление. Как администратор кластера, вы должны вручную одобрять запросы OLM для обновления оператора до новой версии.
  - b. Выберите канал **alpha (Default)**.
  - c. Выберите **Recommended** для **Installation Location**. Установите оператор в рекомендуемом пространстве имен opentelemetry-operator, чтобы оператор мог мониторить и быть доступным во всех пространствах имен кластера.
5. Нажмите **Install**.
6. Убедитесь, что в поле **Status** отображается **Succeeded**, что подтверждает успешную установку OpenTelemetry Operator.

7. Проверьте, что все компоненты OpenTelemetry Operator были успешно установлены.  
Выполните команду:

```
kubectl -n opentelemetry-operator get csv
```

Пример вывода

| NAME                          | DISPLAY       | VERSION  | PHASE     |
|-------------------------------|---------------|----------|-----------|
| openTelemetry-operator.vx.x.0 | OpenTelemetry | Operator | Succeeded |

Если в поле PHASE указано Succeeded, значит оператор и его компоненты установлены успешно.

### Развёртывание экземпляра Jaeger

Экземпляр Jaeger и связанные с ним ресурсы можно установить с помощью скрипта `install-jaeger.sh` (смотри Приложение А), который принимает три параметра:

- `--es-url`: URL доступа к Elasticsearch
- `--es-user-base64`: имя пользователя Basic Auth для Elasticsearch, закодированное в base64
- `--es-pass-base64`: пароль Basic Auth для Elasticsearch, закодированный в base64

Пример запуска:

```
./install-jaeger.sh --es-url='https://xxx' --es-user-base64='xxx' --es-pass-base64='xxx'
```

Пример вывода скрипта:

```
ES_URL: https://xxx
ES_USER_BASE64: xxx
ES_PASS_BASE64: xxx
CLUSTER_NAME: cluster-xxx
PLATFORM_URL: https://xxx
INSTALLED_CSV: jaeger-operator.vx.x.x
OAUTH2_PROXY_IMAGE: build-harbor.alauda.cn/3rdparty/oauth2-proxy/oauth2-proxy:vx.x.x
configmap/jaeger-oauth2-proxy created
secret/jaeger-oauth2-proxy created
secret/jaeger-elasticsearch-basic-auth created
serviceaccount/jaeger-prod-acp created
role.rbac.authorization.k8s.io/jaeger-prod-acp created
rolebinding.rbac.authorization.k8s.io/jaeger-prod-acp created
jaeger.jaegertracing.io/jaeger-prod created
podmonitor.monitoring.coreos.com/jaeger-monitor created
ingress.networking.k8s.io/jaeger-query created
Jaeger installation completed
```

### Развёртывание экземпляров OpenTelemetry

Экземпляры OpenTelemetry и связанные с ними ресурсы можно установить с помощью скрипта `install-otel.sh` (смотри Приложение Б).

Пример запуска:

```
./install-otel.sh
```

Пример вывода скрипта:

```
CLUSTER_NAME: cluster-xxx
serviceaccount/otel-collector created
clusterrolebinding.rbac.authorization.k8s.io/otel-collector:cpaas-system:cluster-admin
created
opentelemetrycollector.opentelemetry.io/otel created
instrumentation.opentelemetry.io/acp-common-java created
servicemonitor.monitoring.coreos.com/otel-collector-monitoring created
servicemonitor.monitoring.coreos.com/otel-collector created
OpenTelemetry installation completed
```

### **Включение функции**

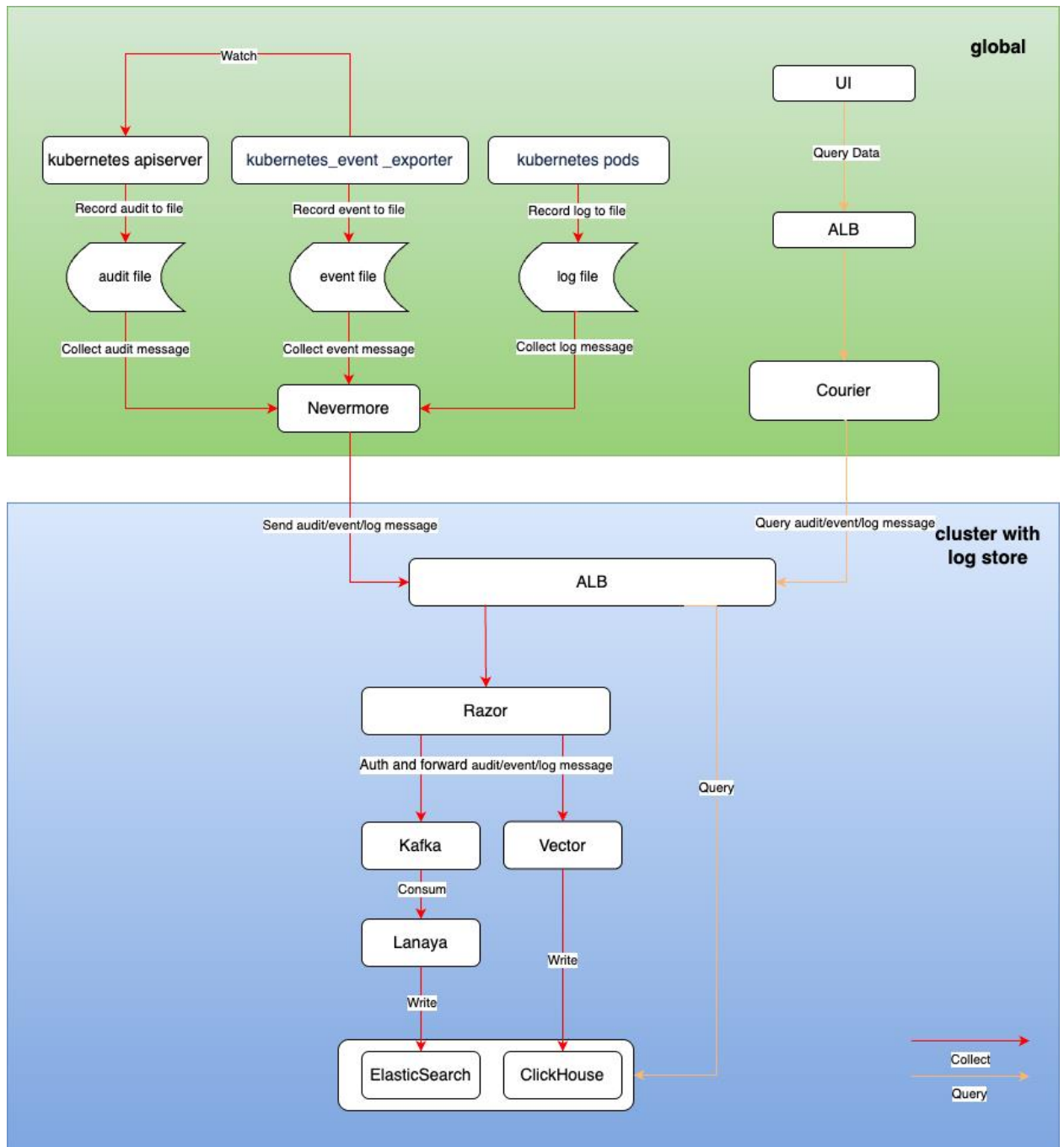
Система трассировки находится в фазе Alpha и требует ручного включения переключателя функции `acp-tracing-ui` в представлении Feature Switch.

Затем перейдите в представление Container Platform и откройте Observability > Tracing, чтобы просмотреть функцию трассировки.

### **Логи**

Модуль логирования — это эффективное и надежное решение для управления логами, предоставляемое MWS CP, разработанное для обеспечения пользователей комплексными возможностями по сбору, хранению, запросам и анализу логов. Основанная на мощных open-source компонентах, система использует Filebeat для сбора логов, а Elasticsearch и Clickhouse в качестве бэкендов для хранения логов, что гарантирует пользователям возможность легко обрабатывать большие объемы данных и получать ключевые бизнес-инсайты в режиме реального времени.

## Архитектура



### Общее описание архитектуры

Система логирования состоит из следующих основных функциональных модулей:

1. Сбор логов
  - a. Основан на open-source компоненте filebeat
  - b. Сбор логов: поддерживается сбор логов стандартного вывода, файловых логов, событий Kubernetes и аудитов
2. Хранение логов
  - a. Предоставляются два различных решения для хранения логов на базе open-source компонентов Clickhouse и ElasticSearch

- b. Хранение логов: поддерживается долговременное хранение лог-файлов
- c. Управление временем хранения логов: поддерживается управление сроком хранения логов на уровне проекта

### 3. Визуализация логов

- a. Обеспечивает удобный и надежный поиск логов, экспорт логов и возможности анализа логов

#### Сбор логов

nevermore устанавливается как daemonset в namespace cpaas-system каждого кластера. Этот компонент состоит из 4 контейнеров:

| Название              | Функция                                                |
|-----------------------|--------------------------------------------------------|
| audit                 | Сбор данных аудита                                     |
| event                 | Сбор данных событий                                    |
| log                   | Сбор логов (включая стандартный вывод и файловые логи) |
| node-problem-detector | Сбор информации об аномалиях на узлах                  |

После того как nevermore собирает информацию аудита/событий/логов, он отправляет данные в кластер хранения логов, проходя аутентификацию через Razor, после чего данные окончательно сохраняются в Elasticsearch или ClickHouse.

#### Визуализация логов

Пользователи могут выполнять запросы аудита/событий/логов через URL-адреса из UI интерфейса продукта для отображения:

- Запрос логов /platform/logging.alauda.io/v1
- Запрос событий /platform/events.alauda.io/v1
- Запрос аудита /platform/audits.alauda.io/v1

Запросы обрабатываются компонентом advanced API Courier, который выполняет запросы к данным логов из кластеров хранения Elasticsearch или Clickhouse и возвращает результаты на страницу.

#### Руководство по выбору компонента логирования

Для установки компонента хранения логов кластера можно выбрать только один из Elasticsearch или Clickhouse.

Для сбора логов и взаимодействия с данными хранилища можно выбрать компонент хранения логов любого кластера.

Текущая версия продукта DevOps не поддерживает архивирование записей выполнения Jenkins pipeline с использованием Clickhouse. Если вам необходимы функции Jenkins pipeline, пожалуйста, не используйте плагин MWS Log Storage с Clickhouse.

Текущая версия ServiceMesh не поддерживает интеграцию с Clickhouse. Если вам нужны функции ServiceMesh, пожалуйста, не используйте плагин MWS Log Storage с Clickhouse.

Текущая версия плагина MWS Log Storage с Clickhouse не поддерживает кластеры с одним стеком IPv6 или с двойным стеком IPv6.

### Рекомендации по выбору

- Если вы привыкли использовать Elasticsearch и сильно зависите от языка Lucene, рекомендуется продолжать использовать плагин MWS Log Storage с ElasticSearch
- Если вы используете функции Jenkins pipeline или ServiceMesh платформы, рекомендуется продолжать использовать плагин MWS Log Storage с ElasticSearch
- Если у вас высокие требования к производительности и потреблению ресурсов компонента логирования, но базовые потребности в запросах логов, рекомендуется выбрать плагин MWS Log Storage с Clickhouse

### Установка

Система логирования платформы состоит из двух плагинов: MWS Log Collector и MWS Log Storage.

#### Внимание:

1. После успешной установки плагина MWS Log Storage кластер, в котором установлен компонент, может использоваться в качестве кластера хранения для различных кластеров платформы (опционально при установке плагина MWS Log Collector) для предоставления услуг хранения логов для всех кластеров платформы.
2. Кластер global может выполнять запросы к данным логов, хранящимся на любом рабочем кластере внутри платформы. Убедитесь, что кластер global имеет доступ к порту 11780 рабочего кластера.
3. Плагины MWS Log Storage с Clickhouse и MWS Log Storage с ElasticSearch не могут быть установлены в одном и том же кластере.

### Установка MWS Log Storage с ElasticSearch

1. Перейдите в **App Store Management > Cluster Plugin** и выберите целевой кластер.
2. На вкладке Plugins нажмите кнопку действий справа от MWS Log Storage with ElasticSearch > **Install**.
3. Ознакомьтесь с инструкциями ниже для настройки соответствующих параметров:

| Параметр                        | Описание                                                                                                                                                                                                                                                                                                                            |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connect External Elasticsearch  | Оставьте закрытым для установки плагина хранения логов внутри платформы.                                                                                                                                                                                                                                                            |
| Component installation Settings | <b>LocalVolume:</b> Локальное хранилище, данные логов будут сохраняться в локальном пути хранения выбранного узла. Преимущество этого способа в том, что компонент логирования напрямую привязан к локальному хранилищу, что исключает необходимость доступа к хранилищу по сети и обеспечивает лучшую производительность хранения. |

| Параметр         | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <p><b>StorageClass:</b> Динамическое создание ресурсов хранения с использованием классов хранения для сохранения данных логов. Преимущество этого способа — большая гибкость; при наличии нескольких классов хранения для всего кластера администраторы могут выбирать соответствующий класс хранения для компонентов логирования в зависимости от сценариев использования, что снижает влияние сбоев хоста на хранение. Однако производительность StorageClass может зависеть от таких факторов, как пропускная способность сети и задержки, а также опирается на механизмы избыточности, предоставляемые бэкендом хранения, для обеспечения высокой доступности хранения.</p> |
| Retention Period | <p>Максимальное время хранения логов, событий и аудиторских данных в кластере. Данные, превышающие период хранения, будут автоматически удалены.</p> <p>Совет: Вы можете создавать резервные копии данных, которые необходимо хранить длительное время. Если нужна помощь, обратитесь к технической поддержке.</p>                                                                                                                                                                                                                                                                                                                                                              |

4. Нажмите **Install**.

#### Установка MWS Log Storage с Clickhouse

1. Перейдите в **App Store Management > Cluster Plugin** и выберите целевой кластер.
2. На вкладке **Plugins** нажмите кнопку действий справа от MWS Log Storage with Clickhouse > **Install**.
3. Ознакомьтесь с инструкциями ниже для настройки соответствующих параметров.

| Параметр                        | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Component installation Settings | <p><b>LocalVolume:</b> Локальное хранилище, данные логов будут сохраняться в локальном пути хранения выбранного узла. Преимущество этого способа в том, что компонент логирования напрямую привязан к локальному хранилищу, что исключает необходимость доступа к хранилищу по сети и обеспечивает лучшую производительность хранения.</p> <p><b>StorageClass:</b> Динамическое создание ресурсов хранения с использованием классов хранения для сохранения данных логов. Преимущество этого способа — большая гибкость; при наличии нескольких классов хранения для всего кластера</p> |



| Параметр         | Описание                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | администраторы могут выбирать соответствующий класс хранения для компонентов логирования в зависимости от сценариев использования, что снижает влияние сбоев хоста на хранение. Однако производительность StorageClass может зависеть от таких факторов, как пропускная способность сети и задержки, а также опирается на механизмы избыточности, предоставляемые бэкендом хранения, для обеспечения высокой доступности хранения. |
| Retention Period | Максимальное время хранения логов, событий и аудиторских данных в кластере. Данные, превышающие период хранения, будут автоматически удалены.<br><br>Совет: Вы можете создавать резервные копии данных, которые необходимо хранить длительное время. Если нужна помощь, обратитесь к технической поддержке.                                                                                                                        |

4. Нажмите **Install**.

#### Установка плагина MWS Log Collector

1. Перейдите в **App Store Management > Cluster Plugin** и выберите целевой кластер.
2. На вкладке **Plugins** нажмите кнопку действий справа от MWS Log Collector > **Install**.
3. Выберите Storage Cluster (где установлен MWS Log Storage) и нажмите Select/Deselect типы логов для настройки области сбора логов в кластере.
4. Нажмите **Install**.

#### События

Платформа интегрируется с событиями Kubernetes, регистрируя значительные изменения статуса и различные изменения операционного состояния ресурсов Kubernetes. Также предоставляются возможности для хранения, запроса и визуализации. При возникновении аномалий с ресурсами, такими как кластеры, узлы или Pods, пользователи могут анализировать события для определения конкретных причин.

На основе выявленных корневых причин из событий пользователи могут создавать политики оповещений для рабочих нагрузок. Когда количество критических событий достигает порога оповещения, оповещения могут автоматически срабатывать для уведомления соответствующего персонала с целью своевременного вмешательства, что снижает операционные риски на платформе.

Данная функция зависит от системы логирования. Пожалуйста, убедитесь, что в платформе предварительно установлены плагины MWS Log Collector и MWS Log Storage.

#### Использование

Нажмите **Operations Center > Events** в левой навигационной панели.

**Совет:** Используйте выпадающий список в верхней навигационной панели для переключения кластера и просмотра событий.

### Обзор событий

На странице событий по умолчанию отображается обзор значимых событий, произошедших за последние 30 минут (вы можете выбрать или настроить временной диапазон), а также записи событий ресурсов.

- **Обзор значимых событий:** Эта карточка показывает причину значимых событий и количество ресурсов, у которых произошли такие события в выбранном временном диапазоне.

Примечание: если один и тот же ресурс испытывал этот тип события несколько раз, количество ресурсов не суммируется.

- **Записи событий ресурсов:** ниже области обзора значимых событий отображаются все записи событий, соответствующие условиям запроса в выбранном временном диапазоне. Вы можете отфильтровать соответствующие типы событий, нажав на карточку значимого события, либо развернуть просмотр и ввести условия запроса для поиска. Условия запроса следующие:
  - **Тип ресурса:** Тип Kubernetes-ресурса, у которого произошло событие, например Pod.
  - **Namespace:** Пространство имён Kubernetes-ресурса, в котором произошло событие.
  - **Причина события:** Причина возникновения события.
  - **Уровень события:** Значимость события, например Warning.
  - **Имя ресурса:** Имя Kubernetes-ресурса, у которого произошло событие. Можно выбрать или ввести несколько имён.

## **Состояние платформы**

Для помощи клиентам в снижении затрат на ручные проверки базовая функциональность мониторинга состояния платформы разработана на основе опыта проведения ручных проверок для клиентов. Она позволяет клиентам в режиме реального времени понимать состояние работы всех ресурсов на платформе, своевременно выявлять аномалии и снижать риски.

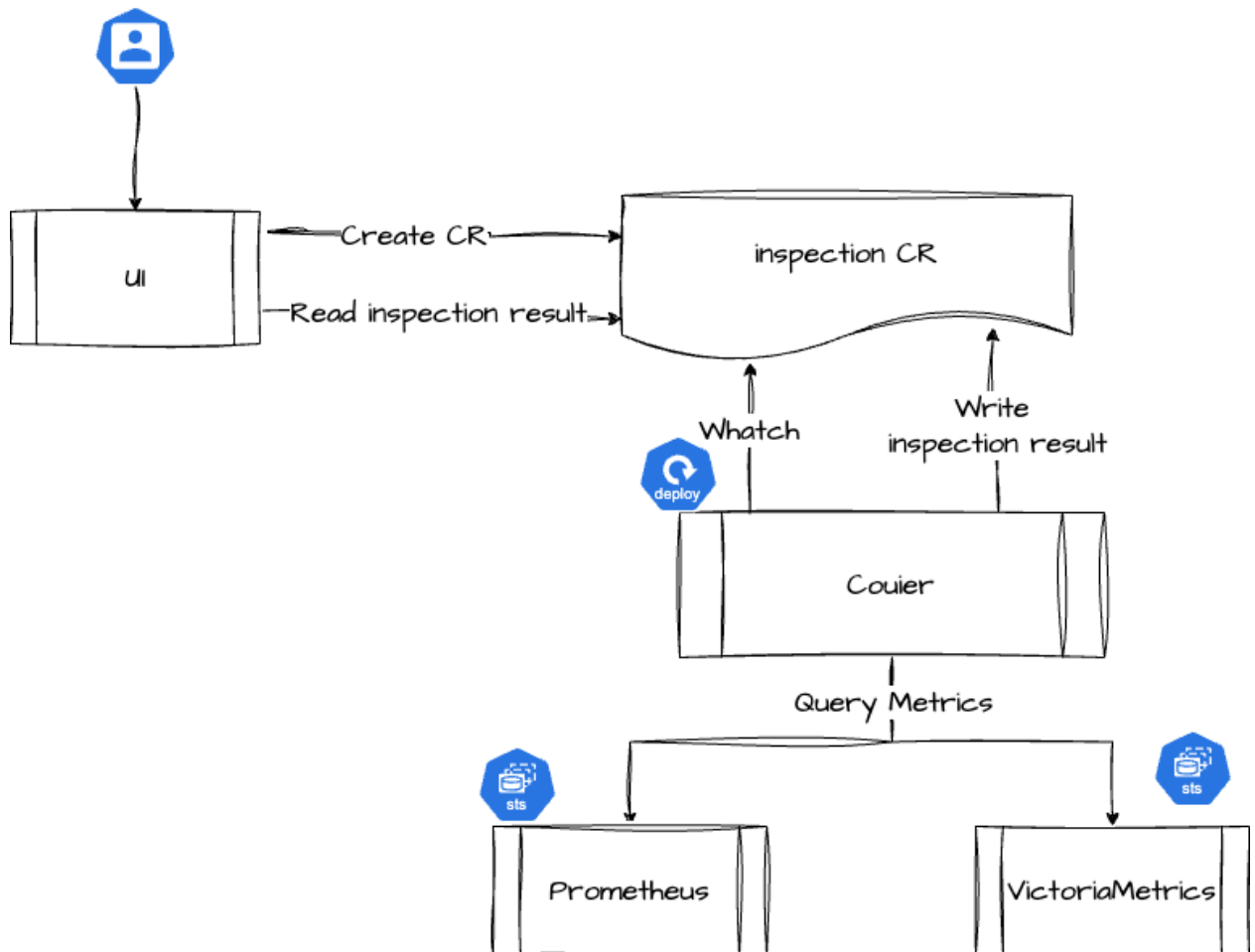
- Поддерживает онлайн-исполнение задач инспекции, включая проверки ресурсов всех кластеров, узлов, подов и сертификатных ресурсов на платформе, а также проверки использования обычных ресурсов с возможностью отслеживания прогресса инспекции в реальном времени
- По завершении инспекции результаты визуально отображаются, включая информацию о состоянии ресурсов и их использовании
- Поддерживает формирование отчетов инспекции в формате PDF или Excel
- Для обеспечения безопасности данных клиентов использовать функциональность инспекции могут только пользователи с соответствующими правами доступа

## **Ограничения использования**

Некоторые пункты проверки состояния на платформе зависят от наличия в кластерах установленных компонентов мониторинга. Пожалуйста, убедитесь, что в каждом кластере заранее установлен либо плагин MWS Monitoring с Prometheus, либо плагин MWS Monitoring с VictoriaMetrics.

Проверка состояния платформы поддерживает отправку результатов по электронной почте. Пожалуйста, убедитесь, что конфигурация сервера уведомлений по электронной почте была выполнена заранее.

## Архитектура



Модуль проверки состояния предоставляет компонент платформы Courier и компонент мониторинга, включая следующие процессы:

- Создание задачи проверки состояния: платформа отправляет CR типа inspection в кластер global,
- Выполнение задачи проверки: компонент Courier отслеживает появление CR типа inspection и запрашивает у компонентов мониторинга каждого кластера различные метрики, связанные с проверкой,
- Запись результатов проверки: после завершения оценки каждого пункта, компонент Courier записывает результаты проверки обратно в соответствующий CR инспекции,
- Просмотр результатов проверки: пользователи могут проверить статус и результаты задач через платформу, где данные будут получены из соответствующего CR инспекции.

## Использование

### Выполнение проверки

1. Нажмите **Operation Center > Inspection > Basic Inspection** в левой навигационной панели.

**Совет:** на странице инспекции отображается информация о данных инспекции из последней проверки. Во время процесса инспекции вы можете в реальном времени просматривать данные ресурсов завершённых инспекций

2. На странице Basic Inspection поддерживаются следующие действия:
  - a. **Execute Inspection:** нажмите кнопку Inspection в правом верхнем углу страницы, чтобы выполнить инспекцию на платформе.
  - b. **Download Inspection Report:** нажмите кнопку Download Report в правом верхнем углу страницы, выберите формат отчёта (PDF или Excel) в появившемся диалоговом окне и нажмите для скачивания, что загрузит отчёт соответствующего формата на ваш локальный компьютер:
    - Отчёт инспекции в формате **PDF** не включает страницу с деталями рисков ресурсов.
    - Отчёт инспекции в формате **Excel** содержит все данные инспекции.

Поддерживается одновременная загрузка отчётов в обоих форматах.

### Описание отчёта

#### Most Recent Inspection

В области информации **Most Recent Inspection** вы можете просмотреть соответствующую информацию о последней инспекции:

- **Inspection Time:** время начала и окончания последней инспекции.
- **Total Number of Inspection Resources:** общее количество ресурсов (кластеров, узлов, pod-ов, сертификатов), проверенных в последней инспекции.
- **Risks:** количество ресурсов с рисками, включая те, которые классифицированы как Fault и Warning.

## Resource Risk Inspection

На странице Resource Risk Inspection вы можете просмотреть обзор информации о рисках для global кластеров, собственных кластеров, подключённых кластеров, а также всех узлов, pod-ов и сертификатов в этих кластерах.

Нажмите кнопку Risk Details на карточке соответствующего типа ресурса (Cluster, Node, pod, Certificate), чтобы перейти на страницу с деталями рисков для этого типа ресурса. На странице деталей вы можете просмотреть информацию о последней инспекции ресурса, а также список ресурсов с ошибками и предупреждениями.

- Нажмите на имя ресурса, чтобы перейти на страницу деталей ресурса
- Нажмите кнопку раскрытия справа от поля Name в списке, чтобы развернуть условия и причины срабатывания ошибок и предупреждений

Для объяснения критериев оценки статуса риска (Fault, Warning) для каждого ресурса смотрите таблицу ниже.

Примечание: для оценки ошибок и предупреждений каждого типа ресурса используется несколько условий; если данные инспекции ресурса соответствуют хотя бы одному из условий, это считается рискованными данными.

| Тип ресурса | Область проверки                                         | Критическая ситуация                                                 | Опасная ситуация                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------|----------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cluster     | global cluster<br>Self-built cluster<br>Accessed cluster | Статус кластера<br>Abnormal<br>Нарушено<br>соединение с<br>apiserver | После увеличения масштаба кластера (число узлов/pod-ов/метрик) ресурсы компонентов мониторинга не обновлены.<br>После увеличения объёма логов и частоты сбора логов ресурсы лог-компонентов не обновлены.<br>Использование CPU кластера превышает 60%<br>Использование памяти кластера превышает 60%<br>Любой pod компонента ETCD кластера находится не в состоянии Running<br>Любой хост кластера находится не в состоянии Ready<br>Разница времени между любыми двумя узлами |

| Тип ресурса | Область проверки                                | Критическая ситуация                                                                                                                                                             | Опасная ситуация                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                                 |                                                                                                                                                                                  | <p>кластера превышает 40 секунд</p> <p>Коэффициент запроса CPU кластера (фактическое значение / общее) превышает 60%</p> <p>Коэффициент запроса памяти кластера (фактическое значение / общее) превышает 80%</p> <p>Компоненты мониторинга не установлены в кластере</p> <p>Компоненты мониторинга кластера работают с ошибками</p> <p>Любой pod компонента kube-controller-manager кластера находится не в состоянии Running</p> <p>Любой pod компонента kube-scheduler кластера находится не в состоянии Running</p> <p>Любой pod компонента kube-apiserver кластера находится не в состоянии Running.</p> |
| Node        | - Все управляющие узлы- Все вычислительные узлы | <p>Статус узла Abnormal</p> <p>Pod компонента node-exporter на узле находится не в состоянии Running</p> <p>Pod компонента kubelet на узле находится не в состоянии Running.</p> | <p>Свободных inode на узле менее 1000-</p> <p>Использование CPU узла превышает 60%</p> <p>Использование памяти узла превышает 60%</p> <p>Использование дискового пространства каталога узла превышает 60%</p> <p>Системная нагрузка узла превышает 200% и длится более 15 минут</p> <p>За последние сутки произошло как минимум одно событие</p>                                                                                                                                                                                                                                                             |

| Тип ресурса | Область проверки                                    | Критическая ситуация                                                   | Опасная ситуация                                                                                                                                                                                                                                                                                                                            |
|-------------|-----------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                                     |                                                                        | <p>NodeDeadlock (зависание узла)</p> <p>За последние сутки произошло как минимум одно событие NodeOOM (недостаток памяти)</p> <p>За последние сутки произошло как минимум одно событие NodeTaskHung (зависание задачи)</p> <p>За последние сутки произошло как минимум одно событие NodeCorruptDockerImage (повреждённый Docker-образ).</p> |
| pod         | Все pod-ы                                           | Статус pod-a Error<br>Pod находится в состоянии запуска более 5 минут. | <p>Использование CPU pod-a превышает 80%</p> <p>Использование памяти pod-a превышает 80%</p> <p>Количество перезапусков pod-a за последние 5 минут больше или равно 1.</p>                                                                                                                                                                  |
| Certificate | - Сертификаты Certmanager<br>Сертификаты Kubernetes | Статус сертификата Expired.                                            | Срок действия сертификата менее 29 дней.                                                                                                                                                                                                                                                                                                    |

### Resource Utilization Inspection

Перейдите на вкладку Resource Utilization Inspection, чтобы открыть страницу Resource Utilization Inspection.

На странице **Resource Utilization Inspection** вы можете просмотреть общий объём, использование и коэффициент использования CPU, памяти и диска для global кластеров, подключённых кластеров и собственных кластеров, а также количество ресурсов, таких как кластеры, узлы, pod-ы и проекты на платформе.

- **Resource Usage Statistics:** вы можете просмотреть общий объём и общий коэффициент использования CPU, памяти и диска для глобальных, подключённых и собственных кластеров.
- **Platform Resource Quantity:** вы можете просмотреть количество ресурсов, работающих на платформе.



# Работа с GPU

## Обзор

Набор аппаратных акселераторов Kubernetes — это корпоративное решение для оптимизации распределения, изоляции и совместного использования ресурсов GPU в облачных нативных средах. Основанный на плагинах устройств Kubernetes и технологиях NVIDIA, он включает три основных модуля:

### 1. vGPU Module

Основан на Opensource GPU-Manager. Модуль обеспечивает тонкую виртуализацию GPU, разделяя физические GPU на виртуальные единицы с квотами по памяти и вычислительным ресурсам. Идеально подходит для мультиарендных сред с динамическим распределением ресурсов.

### 2. pGPU Module

Используя официальный Device Plugin NVIDIA, обеспечивает полную изоляцию физического GPU с планированием, учитывающим NUMA. Предназначен для высокопроизводительных вычислительных задач (HPC), требующих выделенного доступа к GPU.

### 3. MPS Module

Реализует Multi-Process Service NVIDIA для одновременного выполнения нескольких контекстов GPU с ограничениями ресурсов. Оптимизирует приложения с чувствительностью к задержкам за счёт слияния CUDA-ядер.

## vGPU (на основе Opensource GPU-Manager)

- Тонкое разделение ресурсов

Делит физические ядра GPU на квоты от 1 до 100. Поддерживает динамическое распределение для мультиарендных сред, таких как AI inference и виртуальные рабочие столы.

- Планирование с учётом топологии

Автоматически приоритизирует GPU, соединённые через NVLink/C2C, чтобы минимизировать задержки передачи данных между сокетами. Обеспечивает оптимальное сочетание GPU для распределённых тренировок.

## pGPU (NVIDIA Device Plugin)

- Оптимизация под NUMA

Обеспечивает сопоставление 1:1 GPU с Pod с привязкой к NUMA-узлу, снижая конкуренцию на шине PCIe для задач высокопроизводительных вычислений (HPC), таких как обучение LLM.

- Эксклюзивный доступ к оборудованию

Предоставляет полную изоляцию физического GPU через PCIe passthrough, что идеально подходит для критически важных приложений с требованием детерминированной производительности (например, обработка медицинских изображений).

## MPS (NVIDIA Multi-Process Service Plugin)

- Оптимизация задержек выполнения

Позволяет слияние CUDA-ядер между процессами, снижая задержки инференса на 30- 50 % для приложений реального времени, таких как видеоаналитика.

- Совместное использование ресурсов с ограничениями

Позволяет одновременное выполнение контекстов GPU с контролем вычислительных (0 - 100%) и памятных лимитов на процесс через переменные окружения.

## Установка Kubernetes Hardware accelerator Toolkit

### Пререквизиты

Перед установкой убедитесь, что выполнены следующие условия:

- **Kubernetes Cluster:** версия  $\geq 1.25$  с включённым флагом функции **DevicePlugins**.
- **NVIDIA Drivers:** установлены на всех GPU-узлах. Проверьте с помощью `nvidia-smi`.
- **Container Runtime:** настроен с NVIDIA Container Toolkit ( $\geq 1.7.0$ ) для поддержки GPU.

### Веб-консоль

1. Перейдите в **Platform Management**.
2. В левой панели выберите **Catalog > Cluster Plugin**.
3. В поиске введите "gpu" и в строке найденного плагина нажмите **Install**.
4. Настройте расширенные возможности:

| Опция | Функциональность                                                                              | Рекомендуемый сценарий                            |
|-------|-----------------------------------------------------------------------------------------------|---------------------------------------------------|
| PGPU  | Изоляция физического GPU с планированием с учётом NUMA                                        | Обучение ИИ/вычисления высокой производительности |
| vGPU  | Виртуальное разделение GPU через GPU-Manager                                                  | Мультиарендное использование/квоты ресурсов       |
| MPS   | Multi-Process Service для совместного использования вычислительных ресурсов и ресурсов памяти | Вывод с малой задержкой/параллельные задачи       |

5. Нажмите **Install**.

## Настройка аппаратного ускорителя на GPU-узлах

### Конфигурация физического GPU

В интерфейсе управления GPU-кластером выполните развёртывание плагина физического GPU в GPU бизнес-кластере:

1. В левой боковой панели выберите подраздел **Cluster Plugins**, запустите установку плагина **MWS Container Platform Device Plugin** и включите опцию **pGPU**.

2. Во вкладке **Nodes** выберите узлы, на которых необходимо развернуть физический GPU, затем нажмите **Label and Taint Manager** добавьте **device label** и выберите **pGPU**, нажмите **OK**.
3. На вкладке **Pods** проверьте состояние выполнения группы контейнеров, соответствующих `nvidia-device-plugin-ds`, чтобы увидеть, нет ли отклонений, и убедитесь, что он работает на указанных узлах.

### **Конфигурация NVIDIA MPS (драйвер должен поддерживать CUDA версии >= 11.5)**

#### ***Развёртывание плагина NVIDIA MPS в GPU бизнес-кластере***

В интерфейсе управления GPU-кластером выполните следующие действия:

1. В левой боковой панели выберите подраздел **Cluster Plugins**, запустите установку плагина **MWS Container Platform GPU Device Plugin** и включите опцию **MPS**.
2. Во вкладке **Nodes** выберите узлы, на которых необходимо развернуть физический GPU, затем нажмите **Label and Taint Manager** добавьте **device label** и выберите **MPS**, нажмите **OK**.
3. Во вкладке **Pods** проверьте статус работы контейнерной группы, соответствующей `nvidia-mps-device-plugin-daemonset`, чтобы убедиться в отсутствии сбоев и что она запущена на указанных узлах.

#### ***Настройка kube-scheduler (kubernetes >= 1.23)***

1. На **узле управления бизнес-кластером** проверьте, правильно ли scheduler ссылается на политику планирования:

```
cat /etc/kubernetes/manifests/kube-scheduler.yaml
```

Проверьте, есть ли опция `--config` со значением `/etc/kubernetes/scheduler-config.yaml`, например:

```
apiVersion: v1
kind: Pod
metadata:
 creationTimestamp: null
 labels:
 component: kube-scheduler
 tier: control-plane
 name: kube-scheduler
 namespace: kube-system
spec:
 containers:
 - command:
 - kube-scheduler
 - --config=/etc/kubernetes/scheduler-config.yaml
```

**Примечание:** Указанные параметры и значения — это конфигурации по умолчанию платформы. Если вы их изменяли, пожалуйста, верните к значениям по умолчанию. Ваши оригинальные пользовательские настройки можно скопировать в файл политики планирования.

2. Проверьте конфигурацию файла политики планирования.

2.1. Выполните команду:

```
kubectl describe service kubernetes -n default | grep Endpoints
Expected effectEndpoints: 192.168.130.240:6443
```

2.2. Замените содержимое файла `/etc/kubernetes/scheduler-config.yaml` на всех Master-узлах следующим содержимым, где `${kube-apiserver}` нужно заменить на вывод из предыдущего шага:

```
apiVersion: kubescheduler.config.k8s.io/v1beta2
kind: KubeSchedulerConfiguration
clientConnection:
 kubeconfig: /etc/kubernetes/scheduler.conf
extenders:
- enableHTTPS: true
 filterVerb: predicates
 managedResources:
 - ignoredByScheduler: false
 name: nvidia.com/mps-core
 nodeCacheCapable: false
 urlPrefix: https://${kube-apiserver}/api/v1/namespaces/kube-system/services/nvidia-mps-scheduler-plugin/proxy/scheduler
 tlsConfig:
 insecure: false
 certFile: /etc/kubernetes/pki/apiserver-kubelet-client.crt
 keyFile: /etc/kubernetes/pki/apiserver-kubelet-client.key
 caFile: /etc/kubernetes/pki/ca.crt
```

Если в `schedule-config.yaml` уже есть `extenders`, добавьте этот `yaml` в конец:

```
- enableHTTPS: true
 filterVerb: predicates
 managedResources:
 - ignoredByScheduler: false
 name: nvidia.com/mps-core
 nodeCacheCapable: false
 urlPrefix: https://${kube-apiserver}/api/v1/namespaces/kube-system/services/nvidia-mps-scheduler-plugin/proxy/scheduler
 tlsConfig:
 insecure: false
 certFile: /etc/kubernetes/pki/apiserver-kubelet-client.crt
```

```
keyFile: /etc/kubernetes/pki/apiserver-kubelet-client.key
caFile: /etc/kubernetes/pki/ca.crt
```

2.3. Выполните команду для получения ID контейнера:

- Containerd:

Выполните `crictl ps |grep kube-scheduler`, вывод будет примерно таким, первый столбец — это ID контейнера:

```
1d113ccf1c1a9 03c72176d0f15 2 seconds ago Running kube-scheduler
3 ecd054bbdd465 kube-scheduler-192.168.176.47
```

- Docker:

Выполните `docker ps |grep kube-scheduler`, вывод будет примерно таким, первый столбец — это ID контейнера:

```
30528a45a118 d8a9fef7349c "kube-scheduler --au..." 37 minutes ago Up 37
minutes k8s_kube-scheduler_kube-scheduler-192.168.130.240_kube-
system_3e9f7007b38f4deb6ffd1c7587621009_28
```

2.4. Перезапустите контейнер Containerd/Docker, используя полученный ID контейнера:

- Containerd:

```
crictl stop <Container ID>
```

2.5. Перезапустите Kubelet:

```
systemctl restart kubelet
```

## Конфигурация GPU-Manager

### *Настройка kube-scheduler (kubernetes >= 1.23)*

1. На **узле управления бизнес-кластером** проверьте, правильно ли scheduler ссылается на политику планирования.

```
cat /etc/kubernetes/manifests/kube-scheduler.yaml
```

Проверьте, есть ли опция `--config` со значением `/etc/kubernetes/scheduler-config.yaml`, например:

```
apiVersion: v1
kind: Pod
metadata:
 creationTimestamp: null
 labels:
 component: kube-scheduler
 tier: control-plane
 name: kube-scheduler
 namespace: kube-system
spec:
 containers:
 - command:
 - kube-scheduler
```

- --config=/etc/kubernetes/scheduler-config.yaml

**Примечание:** Указанные параметры и значения — это конфигурации по умолчанию платформы. Если вы их изменяли, пожалуйста, верните к значениям по умолчанию. Ваши оригинальные пользовательские настройки можно скопировать в файл политики планирования.

2. Проверьте конфигурацию файла политики планирования.

2.1. Выполните команду:

```
kubectl describe service kubernetes -n default | grep Endpoints.
```

```
Expected effectEndpoints: 192.168.130.240:6443
```

2.2. Замените содержимое файла /etc/kubernetes/scheduler-config.yaml на всех Master-узлах следующим содержимым, где \${kube-apiserver} нужно заменить на вывод из предыдущего шага:

```
apiVersion: kubescheduler.config.k8s.io/v1beta2
```

```
kind: KubeSchedulerConfiguration
```

```
clientConnection:
```

```
 kubeconfig: /etc/kubernetes/scheduler.conf
```

```
extenders:
```

```
- enableHTTPS: true
```

```
 filterVerb: predicates
```

```
 managedResources:
```

```
- ignoredByScheduler: false
```

```
 name: tencent.com/vcuda-core
```

```
nodeCacheCapable: false
```

```
urlPrefix: https://${kube-apiserver}/api/v1/namespaces/kube-system/services/gpu-quota-admission/proxy/scheduler
```

```
tlsConfig:
```

```
 insecure: false
```

```
 certFile: /etc/kubernetes/pki/apiserver-kubelet-client.crt
```

```
 keyFile: /etc/kubernetes/pki/apiserver-kubelet-client.key
```

```
 caFile: /etc/kubernetes/pki/ca.crt
```

2.3. Выполните следующую команду для получения ID контейнера:

- Containerd:

Выполните `crictl ps | grep kube-scheduler`, вывод будет примерно таким, первый столбец — это ID контейнера:

```
1d113ccf1c1a9 03c72176d0f15 2 seconds ago Running kube-scheduler
3 ecd054bbdd465 kube-scheduler-192.168.176.47
```

- Docker:

Выполните `docker ps | grep kube-scheduler`, вывод будет примерно таким, первый столбец — это ID контейнера:

```
30528a45a118 d8a9fef7349c "kube-scheduler --au..." 37 minutes ago Up 37
minutes k8s_kube-scheduler_kube-scheduler-192.168.130.240_kube-
system_3e9f7007b38f4deb6ffd1c7587621009_28
```

2.4. Перезапустите контейнер Containerd/Docker, используя полученный ID контейнера:

- Containerd:  
`crictl stop <Container ID>`

2.5. Перезапустите Kubelet:

```
systemctl restart kubelet
```

### ***Развёртывание плагина GPU Manager в GPU бизнес- кластере***

В интерфейсе управления GPU-кластером выполните следующие действия:

1. В левой боковой панели выберите подраздел **Cluster Plugins**, запустите установку плагина **MWS Container Platform GPU Device Plugin** и включите опцию **GPU-Manager**.
2. Во вкладке **Nodes** выберите узлы, на которых необходимо развернуть физический GPU, затем нажмите **Label and Taint Manager**, добавьте **device label** и выберите **vGPU**, нажмите **OK**.
3. Во вкладке **Pods** проверьте статус работы контейнерной группы, соответствующей `gpu-manager-daemonset`, чтобы убедиться в отсутствии сбоев и что она запущена на указанных узлах.

### **Проверка результатов**

**Метод 1:** Проверьте наличие доступных GPU-ресурсов на GPU-узлах, выполнив следующую команду на управляющем узле бизнес-кластера:

```
kubectl get node ${nodeName} -o=jsonpath='{.status.allocatable}'
```

**Метод 2:** Разверните GPU-приложение на платформе, указав необходимое количество GPU-ресурсов. После развёртывания выполните `exec` в Pod и выполните следующую команду:

```
nvidia-smi
```

Проверьте, корректно ли получена информация о GPU.

## **Мониторинг ресурсов GPU**

### **Обзор**

Мониторинг ресурсов — это ключевой компонент Kubernetes Hardware Accelerator Suite, предназначенный для обеспечения всесторонней видимости использования GPU-ресурсов в ваших контейнеризованных рабочих нагрузках. Этот модуль предоставляет как **использование вычислительных ресурсов**, так и **потребление памяти GPU** на двух основных уровнях:

- **Мониторинг на уровне узла:** отслеживание совокупного использования GPU-ресурсов на всех узлах Kubernetes.
- **Мониторинг на уровне пода:** анализ потребления GPU по отдельным рабочим нагрузкам с детализацией по подам.

Интегрированный с основными модулями платформы для ускорителей (pGPU/vGPU(GPU-Manager)/MPS), этот инструмент мониторинга позволяет пользователям оптимизировать распределение GPU, обеспечивать соблюдение квот ресурсов и устранять узкие места в производительности AI/ML нагрузок, сервисов реального времени и т.д.

Ограничения:

- Зависимости модулей: требуется развертывание как минимум одного модуля ускорителя (pGPU/vGPU/MPS) в кластере.

## Мониторинг узлов

Отслеживайте ресурсы GPU на уровне узла.

### Шаг 1 . Доступ к панелям GPU

1. Перейдите в **Platform Management**.
2. В левой панели выберите Operations Center > Monitoring > Dashboards.
3. Переключитесь в каталог **GPU**.

### Шаг 2. Выбор метрик узла

1. Выберите панель **Node Monitoring**.
2. Выберите целевой узел из выпадающего списка.
3. Укажите временной диапазон:
  - Последние 30 минут
  - Последний 1/6/12/24 часа
  - Последние 2/7 дней
  - Пользовательский диапазон

### Шаг 3. Интерпретация метрик

| Metric           | Description                                                   |
|------------------|---------------------------------------------------------------|
| GPU Utilization  | Процент использования вычислительной мощности GPU (0 - 100 %) |
| GPU Memory Usage | Объем потребляемой памяти по сравнению с доступной (в ГБ)     |

## Мониторинг подов

Анализируйте использование GPU на уровне пода с детальной фильтрацией.

### Шаг 1. Доступ к метрикам пода

1. Перейдите в панели каталога **GPU**.
2. Выберите Pod Monitoring.



### **Шаг 2. Настройка фильтров**

1. Выберите тип GPU:
  - pGPU
  - GPU-Manager(vGPU)
  - MPS
2. Выберите namespace, содержащий GPU-поды.
3. Выберите конкретный под.

### **Шаг 3. Ключевые метрики**

| Метрика             | Описание                                                  |
|---------------------|-----------------------------------------------------------|
| Pod GPU Utilization | Использование вычислительной мощности GPU выбранным подом |
| Pod GPU Memory      | Распределение памяти для выбранного пода                  |

### **Выбор диапазона времени**

Для мониторинга подов и узлов поддерживается гибкий выбор диапазона времени:

Available Presets:

- Last 30 minutes
- Last 1 hour
- Last 6 hours
- Last 12 hours
- Last 24 hours
- Last 2 days
- Last 7 days
- Custom range

## Управление затратами

В эпоху стремительного развития облачных вычислений предприятиям необходима гибкая и точная модель выставления счетов для управления и оптимизации расходов на облачные ресурсы. Платформа предлагает продукт для выставления счетов, специально разработанный для облачно-ориентированных сценариев, который помогает вам достичь детального управления ресурсами и контроля затрат.

Платформа предоставляет обобщенные возможности для измерения и выставления счетов, а также основные методы выставления счетов, поддерживая измерение и выставление счетов за ресурсы CPU, памяти и хранения в кластерах, а также предлагая статистику и возможности подробного просмотра:

- **Статистика затрат:** поддерживает администраторов платформы в обобщении затрат для каждого проекта.
- **Детализация затрат:** позволяет администраторам платформы запрашивать подробности о затратах для каждого проекта с целью сверки.

## Преимущества

- **Тщательное управление ресурсами:** точно измеряет использование ресурсов CPU, памяти и хранения, помогая пользователям глубже понять потребление ресурсов.
- **Контроль затрат:** с помощью детализированных данных измерений и информации о выставлении счетов пользователи могут лучше контролировать и оптимизировать затраты на ресурсы.
- **Гибкие методы выставления счетов:** поддерживает различные основные методы выставления счетов для удовлетворения разных потребностей пользователей.
- **Прозрачность:** предоставляет статистику использования ресурсов и функции детального просмотра, повышая прозрачность использования ресурсов.
- **Интеграция с OpenCost:** управление затратами основано на OpenCost, что обеспечивает точность и надежность данных измерений.

OpenCost — это инструмент для мониторинга и управления затратами в Kubernetes с открытым исходным кодом, предназначенный для отслеживания и анализа использования ресурсов и затрат в Kubernetes кластерах в реальном времени. Он предоставляет точные данные о замерах ресурсов, чтобы помочь пользователям оптимизировать распределение ресурсов и контролировать расходы.

## Сценарии использования

- **Управление ресурсами предприятия:** помогает предприятиям точно измерять и распределять затраты на IT-ресурсы, оптимизируя распределение ресурсов.
- **Облачно-ориентированные приложения:** подходят для архитектур микросервисов в облачно-ориентированных средах, точно измеряя потребление ресурсов каждым сервисом.

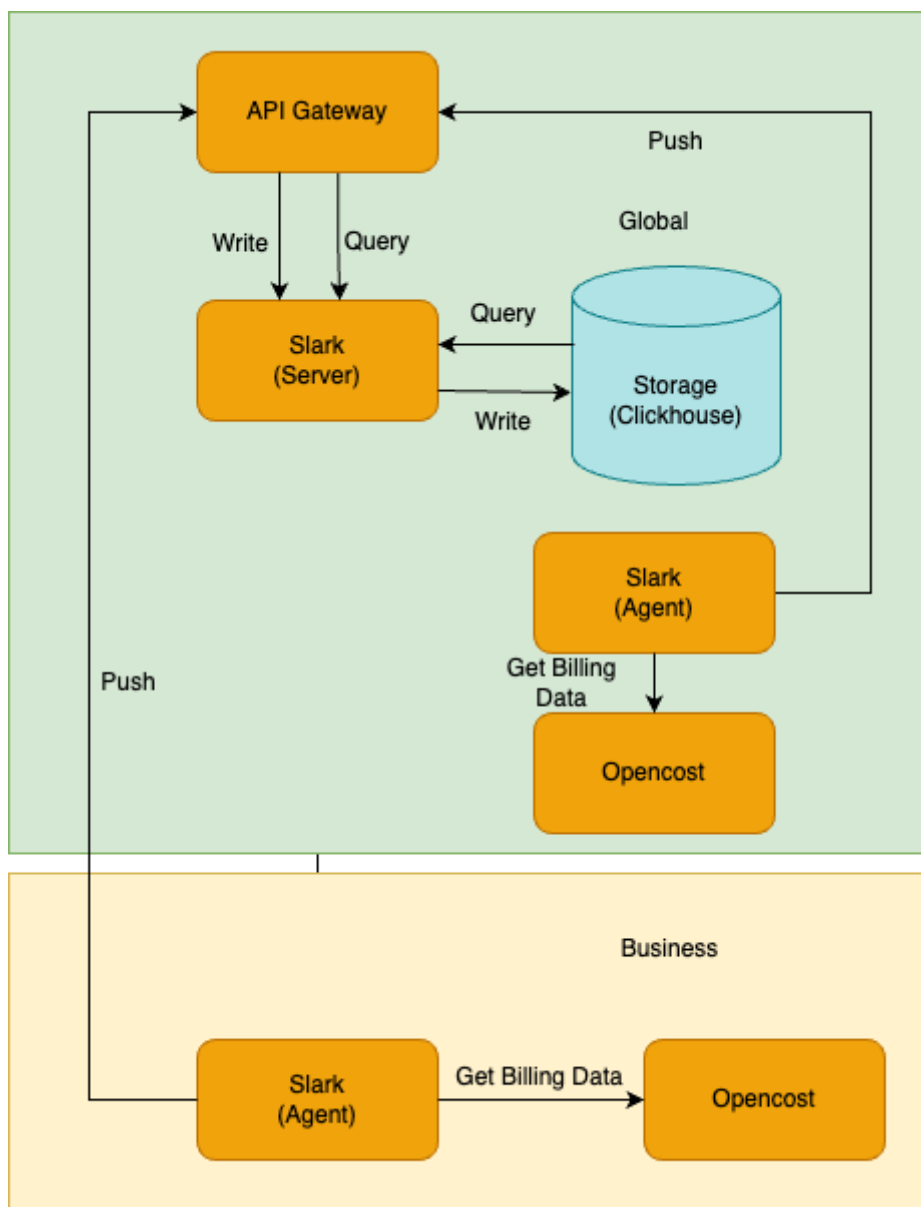
- **Мультиарендные среды:** в мультиарендных средах предоставляет независимое измерение и выставление счетов за ресурсы для каждого проекта, обеспечивая справедливость и прозрачность.
- **Оптимизация затрат:** на основе детализированных данных о потреблении ресурсов определяет и оптимизирует ресурсы с высокими затратами для снижения общих операционных затрат.

## **Ограничения использования**

- **Зависимость от Prometheus:** данные измерений получаются на основе Prometheus; убедитесь, что в кластерах, требующих измерений, установлен MWS-CP Monitor с плагином Prometheus.
- **Ограничения по типам ресурсов:** в настоящее время поддерживаются только измерение и выставление счетов за ресурсы CPU, памяти и хранения; другие типы ресурсов в данный момент не поддерживаются.
- **Совместимость кластеров:** функциональность может быть несовместима со всеми типами кластеров Kubernetes и должна использоваться в поддерживаемых средах кластеров.
- **Задержка данных:** из-за задержки в канале сбора данных возможно некоторое отклонение во времени получения данных измерений.

Функциональность помогает пользователям лучше управлять и оптимизировать использование ресурсов своей контейнерной платформы, достигая более эффективного контроля затрат.

## Архитектура



Платформа обеспечивает учёт и выставление счетов благодаря совместной работе сервера управления затратами MWS Container Platform Cost Management Server и агента управления затратами MWS Container Platform Cost Management Agent:

- **MWS Container Platform Cost Management Server:** необходимо установить этот плагин в кластер global платформы, чтобы использовать функции учета и выставления счетов платформы. Основная функциональность компонента:
  - Взаимодействует с Clickhouse, отвечает за получение данных, сообщаемых клиентами, и их хранение в базе данных Clickhouse без ограничения по времени хранения данных.
  - Предоставляет API для запроса данных об учете и выставлении счетов.
  - Предлагает возможность изменять цены за единицу и настройки валютных единиц.

**Примечание:** Валютная единица, установленная платформой, в настоящее время влияет только на отображение и не поддерживает конвертацию валют между различными единицами.

- **MWS Container Platform Cost Management Agent:** необходимо установить этот плагин в кластер, в котором планируется учет и выставление счетов, чтобы платформа могла учитывать и выставять счета за ресурсы в кластере. Основная функциональность компонента:
  - Взаимодействует с Prometheus, предоставляя экспортёр для хранения данных о метриках учета и выставления счетов в Prometheus.
  - Подключается к MWS Container Platform Cost Management Server, агрегирует и вычисляет данные об учете и выставлении счетов для всех ресурсов в кластере на следующий день в 12:00 каждый день и передает их на сервер. Если отчет не удастся отправить, он попытается повторно отправить его каждые два часа до успешной отправки.

## Описание методов биллинга

| Пункт биллинга | Метод биллинга                    | Правила биллинга                                                                                                                                                                                                                                                       | Описание |
|----------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| CPU            | Эффективное использование (CPU/ч) | Рассчитывается на почасовой основе с использованием большего значения среднего потребления (AVG(Usage)) и запрашиваемого объема (Request) POD за последний час, умноженного на фактическую продолжительность работы POD (исчисляется как 5 минут, если менее 5 минут). |          |
| Память         | Запрашиваемый объем (ГБ/ч)        | Рассчитывается на почасовой основе с использованием большего значения среднего потребления (AVG(Usage)) и запрашиваемого объема (Request) POD за                                                                                                                       |          |

| Пункт биллинга | Метод биллинга             | Правила биллинга                                                                                                                                                                                                                                                                                                       | Описание                                                                                                                                |
|----------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|                |                            | последний час, умноженного на фактическую продолжительность работы POD (исчисляется как 5 минут, если менее 5 минут)                                                                                                                                                                                                   |                                                                                                                                         |
| Хранилище      | Запрашиваемый объем (ГБ/ч) | Рассчитывается на почасовой основе с использованием размера запроса (Request) PV за последний час, умноженного на фактическую продолжительность работы PV (исчисляется как 5 минут, если менее 5 минут). Если PV одновременно смонтирован к нескольким POD, использование PV будет равномерно распределено между ними. | PV, которые не смонтированы ни к одному POD, будут учитываться как не смонтированные записи, при этом имя объекта будет *unmounted-pvcs |

Текущие проблемы расчёта статистики будут решены в будущих версиях:

- Когда POD StatefulSet останавливается и перезапускается, платформа будет использовать самое раннее время работы POD в тот день в качестве стартового времени и самое позднее время работы в качестве конечного времени, игнорируя при этом время, в течение которого POD не работал.

## Установка

### Установка плагина сервера управления затратами

#### Пререквизиты

Плагин MWS Container Platform Cost Management Server использует ClickHouse для хранения данных. Вы можете либо установить хранилище журналов MWS-CP с плагином ClickHouse, предоставляемым платформой, либо самостоятельно установить ClickHouse для интеграции с сервером управления затратами.

### Установить плагин сервера

1. Войдите в кластер, где установлено хранилище журналов MWS-CP с плагином ClickHouse, и получите информацию об интеграции с ClickHouse:

# Получить адрес подключения ClickHouse

```
kubectl get feature log-center -o jsonpath='{.spec.accessInfo.clickhouse.address}'
```

# Получить имя пользователя и пароль ClickHouse:

```
kubectl get secret -n cpaas-system clickhouse-basic-auth -o
jsonpath='{.data.username}' | base64 -d
```

```
kubectl get secret -n cpaas-system clickhouse-basic-auth -o
jsonpath='{.data.password}' | base64 -d
```

2. В левой навигационной панели нажмите **App Store > Plugin Management**.
3. Выберите кластер global.
4. Найдите плагин MWS Container Platform Cost Management Server и нажмите кнопку Install в правой панели.
5. Настройте параметры:

| Параметр                | Описание                                                                                                                                                                                                          |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clickhouse Address      | Настройте адрес подключения ClickHouse                                                                                                                                                                            |
| Clickhouse Cluster Name | Имя кластера для ClickHouse; по умолчанию для ClickHouse, созданного на платформе, используется replicated                                                                                                        |
| Clickhouse Secret Name  | Настройте секрет для имени пользователя и пароля ClickHouse                                                                                                                                                       |
| CPU Unit Price          | Установите цену за единицу учета CPU; платформа будет выставлять счета на основе фактического использования (CPU/ч), что является максимальным значением между Запросом и Использованием в течение одного часа    |
| Memory Unit Price       | Установите цену за единицу учета памяти; платформа будет выставлять счета на основе фактического использования (CPU/ч), что является максимальным значением между Запросом и Использованием в течение одного часа |
| Storage Unit Price      | Установите цену за единицу учета хранилища; платформа будет выставлять счета на основе объема запроса (ГБ/ч), что соответствует размеру запрашиваемого пространства хранилища PV                                  |
| Currency Unit           | Выберите валюту для отображения затрат                                                                                                                                                                            |

6. Нажмите Установить.

## Установка плагина агента управления затратами

### Пререквизиты

Убедитесь, что в кластере установлен плагин MWS Container Platform Monitor with Prometheus.

### Установить плагин агента

1. В левой навигационной панели нажмите **App Store > Plugin Management**.
2. Выберите кластер, где необходимо установить плагин агента.
3. Найдите плагин MWS Container Platform Cost Management Agent и нажмите кнопку Install в правой панели.

## Учёт и выставление счетов

**Внимание:** Учёт и выставление счетов — альфа-версия. Чтобы её использовать, включите `ait-billing`.

### Статистика затрат

В статистике затрат поддерживается анализ затрат на ресурсы платформы по различным параметрам:

- **Детализация времени:** поддерживает статистику затрат, основанную на проекте, кластере или пространстве имен.
- **Период отчетности:** поддерживает статистику затрат на ежемесячной, еженедельной или ежедневной основе.
- **Разделение выставления счетов по элементам:** вы можете указать, следует ли отображать расходы по каждому элементу счёта отдельно, например, по CPU, памяти или хранилищу.

Чтобы посмотреть статистику затрат:

1. В левой панели навигации нажмите **Metering and Billing > Cost Statistics**.
2. Выберите гранулярность времени, период отчетности, необходимость выставления счетов с разделением по элементам, определите диапазон времени и настройте фильтр по имени. После этого нажмите **Search**.

### Примечание:

- Максимальный период для запроса данных — один год.
- Валюта, используемая в метрике затрат, определяется конфигурацией плагина MWS Container Platform Cost Management Server.

### Детализация затрат

Детализация затрат позволяет запрашивать данные о деталях счетов в пользовательском диапазоне времени (максимальный срок — 1 год) с минимальной гранулярностью времени — **день**.

Чтобы просмотреть детализацию затрат:



1. В левой панели навигации нажмите **Metering and Billing > Cost Details**.
2. Выберите проект, кластер и пространство имен и нажмите кнопку **Search**.

Примечание:

- Максимальный период для запроса данных — один год.
- Валюта, используемая в метрике затрат, определяется конфигурацией плагина MWS Container Platform Cost Management Server.
- Не смонтированные PV, которые не подключены к поду, будут учитываться как не смонтированные записи с использованием имени объекта `*unmounted-pvcs`.

## Статистика потребления

**Внимание:** Статистика потребления — это функциональность, предоставляемая платформой, которая в будущем будет заменена новой функциональностью учёта и выставления счетов.

### Обзор потребления

Страница с обзором потребления ресурсов платформы представляет собой визуальный отчет, демонстрирующий общее использование процессора/памяти в установленном диапазоне времени, а также статистику использования для 5 лучших проектов или пространств имен.

Поддерживается просмотр квот проектов за текущий месяц (по умолчанию), за прошлый месяц и за последние 3 месяца (текущий месяц и два предыдущих полных месяца), а также предоставляются статистические данные по подам, квотам подов, квотам пространств имен и общему использованию процессора/памяти.

Чтобы просмотреть общую статистику потребления, в левом навигационном меню нажмите **Operation Statistics > Operation Overview**.

Вы можете переключать диапазон времени, элементы статистики и гранулярность, и просматривать данные операций.

**Примечание:** Использование подов рассчитывается на основе фактического использования подов, в то время как квоты подов основываются на запрашиваемых значениях (request) подов.

### Отчет о статистике потребления

Данные статистики потребления могут храниться на платформе максимум 540 дней, после чего платформа автоматически удалит статистику, превышающую срок хранения.

Поддерживается просмотр визуальных отчетных данных по элементам статистики в установленном пользовательском диапазоне времени (максимальный диапазон времени — 6 месяцев). Фильтрация может производиться на основе статистической гранулярности (проект, пространство имен) и проекта (все, пользовательские проекты).

Элементы статистики включают: использование подов, квоты подов, квоты проектов и квоты пространств имен. Дополнительно, нажав **Export Results**, вы сможете экспортировать отфильтрованные статистические данные в формате отчета CSV.

Чтобы просмотреть отчёт о статистике потребления:

1. В левом навигационном меню нажмите **Operation Statistics > Statistical Report**.
2. Выберите диапазон времени, элемент статистики и гранулярность, затем установите условия фильтрации проектов и нажмите кнопку **Search**.

**Примечания:**

- Платформа хранит данные только за последние 540 дней, а максимальный поддерживаемый диапазон времени для одного запроса составляет 6 месяцев.
- Использование подов: рассчитывается на основе фактического использования подов, суммируя среднее использование, рассчитанное каждые 5 минут.
- Квоты подов: рассчитываются на основе запрашиваемых значений (request) подов.
- Квоты пространств имен: рассчитываются на основе квот процессоров и памяти пространства имен, со статистикой, собираемой раз в день, начиная на следующий день после изменений квот.
- Квоты проектов: рассчитываются на основе квот процессоров и памяти проекта, рассчитываемых раз в день, начиная со следующего дня после изменений квот.
- Подробные данные можно просматривать только когда элемент статистики равен квоте на под или использованию пода.
- Доступна сортировка по имени пространства имен (первая буква), принадлежащему проекту (первая буква), общему использованию процессора (размер) и общему использованию памяти (размер).

**Детализация потребления**

Платформа хранит детализацию потребления около 18 месяцев и поддерживает запрос визуальных данных отчета в установленном пользовательском диапазоне времени (в пределах последних 31 дня) для элементов статистики.

Фильтрация может выполняться на основе элементов статистики (использование подов, квоты подов), названия проекта, связанного кластера проекта и пространств имен под кластером. Кроме того, нажав **Export Results**, вы сможете экспортировать отфильтрованные статистические данные в формате отчета CSV.

Чтобы посмотреть детализацию потребления ресурсов:

1. В левом навигационном меню нажмите **Operation Statistics > Metering Details** или на странице отчёта о статистике потребления в строке нужной записи нажмите **Details**.
2. Выберите диапазон времени, элемент статистики, проект, кластер и пространство имен, затем нажмите кнопку **Search**.

**Примечания:**

- Платформа хранит данные только за последние 18 месяцев, а максимальный поддерживаемый диапазон времени для одного запроса составляет 31 день.
- Использование подов: рассчитывается на основе фактического использования подов, суммируя среднее использование, рассчитанное каждые 5 минут.

- Квоты подов: рассчитываются на основе запрашиваемых значений (request) подов.
- Доступна сортировка по имени пода (первая буква), общему использованию процессора (размер) и общему использованию памяти (размер).

## Сервисы данных

Сервисы данных — это платформа управления сервисами, построенная на технологиях контейнеров и Kubernetes, ориентированная на разработчиков и специалистов по эксплуатации. Она обеспечивает бесшовную интеграцию быстрого развёртывания, упрощённого управления и интуитивно понятных визуальных операций для решения задач развёртывания и эксплуатации в корпоративной среде, предоставляя надёжную поддержку для разработки бизнес-приложений.

## Архитектура



Сервисы данных — это надежное решение для управления сервисами с открытым исходным кодом, построенное на Kubernetes (K8S) и адаптированное для бесшовной работы. Она служит централизованным узлом, эффективно оркестрируя разнообразные популярные сервисные компоненты, включая MySQL, Redis, Kafka, PostgreSQL и RabbitMQ. Архитектура включает в себя:

- **Сервисы данных:** Платформа поддерживает ключевые сервисы, такие как MySQL, PostgreSQL, Redis, Kafka и RabbitMQ, обеспечивая стабильную и надежную работу всех компонентов.
- **Общие сервисы:** Предоставляет шаблоны параметров, инспекцию, управление оповещениями, сервисы резервного копирования и др., предлагая пользователям комплексное решение для управления сервисами.
- **Operator Controller:** Operator Controller — это расширение Kubernetes, разработанное для упрощения развёртывания и управления приложениями. Он расширяет API Kubernetes с помощью Custom Resource Definitions (CRD) для управления жизненным циклом приложений.
- **Контейнерная инфраструктура:** сервисы данных построены на Kubernetes и используют базовую инфраструктуру, которую он предоставляет, чтобы обеспечить фундаментальные возможности, такие как сохранение данных и внешний доступ.

## Поддерживаемые сервисы данных

### MySQL-PXC

- Управление экземплярами

Поддерживает управление жизненным циклом экземпляров (создание, обновление, апгрейд, удаление, перезапуск, остановка).

- Конфигурация параметров

Преднастроенные шаблоны параметров, подходящие для продакшена, поддерживает создание и управление шаблонами параметров; поддерживает инициализацию параметров с помощью шаблонов, а также обновление, импорт и экспорт параметров.

- Методы доступа

Поддерживает различные методы экспонирования сервисов внутри и вне кластера, такие как NodePort, MetalB, ALB и др. Поддерживает копирование информации для подключения в один клик, с предоставлением примеров кода для помощи.

- Топология

Поддерживает отображение топологии экземпляров и состояния работы каждого узла.

- Управление пользователями

Поддерживает управление пользователями экземпляров, включая создание, удаление пользователей и контроль прав доступа.

- Резервное копирование и восстановление

Поддерживает ручное и плановое резервное копирование с возможностью сохранения на тома хранения и внешнее хранилище S3.

- Масштабирование

Поддерживает операции масштабирования, включая горизонтальное масштабирование узлов и вертикальное масштабирование характеристик.

- Терминал

Поддерживает командный терминал, облегчающий быстрый вход на узлы для операций пользователя.

- Анализ запросов

Поддерживает отображение анализа SQL-запросов, показывая SQL-выражения, частоту, время выполнения и другую информацию.

- Логи

Поддерживает запросы логов в реальном времени, включая поиск по времени, контейнеру и др.; поддерживает сохранение логов.

- Мониторинг и оповещение

Отображает данные мониторинга, включая использование различных ресурсов, количество подключений, TPS, операции блокировки таблиц, медленные запросы и др.; поддерживает создание и обновление политик оповещений, с предустановленными метриками, включая

количество запросов чтения/записи InnoDB, количество подключений, количество потоков и десять других метрик; поддерживает настройку оповещений для сотен метрик мониторинга (непредустановленных).

- Инспекция

Поддерживает инспекцию экземпляров с возможностью ручного и автоматического запуска; поддерживает отображение статистической информации результатов инспекции; поддерживает экспорт отчетов инспекции.

- Восстановление после сбоев

Поддерживает горячее резервирование для восстановления после сбоев с опциями асинхронной и синхронизированной по версии репликации для кластеров восстановления.

- Поддержка сетевого протокольного стека

Поддерживает двойной стек IPv4/IPv6, одиночный стек IPv4 и одиночный стек IPv6.

## **MySQL-MGR**

- Управление экземплярами

Поддерживает управление жизненным циклом экземпляров (создание, обновление, апгрейд, удаление, перезапуск, остановка), поддерживает архитектуры с одним мастером и с несколькими мастерами.

- Конфигурация параметров

Преднастроенные шаблоны параметров, подходящие для продакшена, поддерживает создание и управление шаблонами параметров; поддерживает инициализацию параметров с помощью шаблонов, а также обновление, импорт и экспорт параметров.

- Методы доступа

Поддерживает различные методы экспонирования сервисов внутри и вне кластера, такие как NodePort, MetaLB, ALB и др. Поддерживает копирование информации для подключения в один клик, с предоставлением примеров кода для помощи.

- Топология

Поддерживает отображение топологии экземпляров и состояния работы каждого узла.

- Управление пользователями

Поддерживает управление пользователями экземпляров, включая создание, удаление пользователей и контроль прав доступа.

- Резервное копирование и восстановление

Поддерживает ручное и плановое резервное копирование с возможностью сохранения на внешнее хранилище S3, поддерживает восстановление до определенного момента времени.

- Масштабирование

Поддерживает операции масштабирования, включая горизонтальное масштабирование узлов и вертикальное масштабирование характеристик.

- Терминал

Поддерживает командный терминал, облегчающий быстрый вход на узлы для операций пользователя.

- Анализ запросов

Поддерживает отображение анализа SQL-запросов, показывая SQL-выражения, частоту, время выполнения и другую информацию.

- Логи

Поддерживает запросы логов в реальном времени, включая поиск по времени, контейнеру и др.; поддерживает сохранение логов.

- Мониторинг и оповещение

Отображает данные мониторинга, включая использование различных ресурсов, количество подключений, TPS, операции блокировки таблиц, медленные запросы и др.; поддерживает создание и обновление политик оповещений, с предустановленными метриками, включая количество запросов чтения/записи InnoDB, количество подключений, количество потоков и десять других метрик; поддерживает настройку оповещений для сотен метрик мониторинга (непредустановленных).

- Инспекция

Поддерживает инспекцию экземпляров с возможностью ручного и автоматического запуска; поддерживает отображение статистической информации результатов инспекции; поддерживает экспорт отчетов инспекции.

- Восстановление после сбоев

Поддерживает горячее резервирование для восстановления после сбоев с опциями асинхронной и синхронизированной по версии репликации для кластеров восстановления.

- Поддержка сетевого протокольного стека

Поддерживает двойной стек IPv4/IPv6, одиночный стек IPv4 и одиночный стек IPv6.

## **Redis**

- Управление экземплярами

Поддерживает управление жизненным циклом экземпляров (создание, обновление, апгрейд, удаление, перезапуск, остановка), поддерживает архитектуры кластера и sentinel.

- Конфигурация параметров

Преднастроенные шаблоны параметров, подходящие для продакшена, поддерживает создание и управление шаблонами параметров; поддерживает инициализацию параметров с помощью шаблонов, а также обновление, импорт и экспорт параметров.

- Методы доступа

Поддерживает различные методы экспонирования сервисов внутри и вне кластера, такие как NodePort, MetalB, ALB и др. Поддерживает копирование информации для подключения в один клик, с предоставлением примеров кода для помощи.

- Топология

Поддерживает отображение топологии экземпляров и состояния работы каждого узла.

- Управление пользователями

Поддерживает управление пользователями экземпляров, включая создание, удаление пользователей и контроль прав доступа.

- Резервное копирование и восстановление

Поддерживает ручное и плановое резервное копирование с возможностью сохранения на тома хранения и внешнее хранилище S3.

- Масштабирование

Поддерживает операции масштабирования, включая горизонтальное масштабирование узлов и вертикальное масштабирование характеристик.

- Терминал

Поддерживает командный терминал, облегчающий быстрый вход на узлы для операций пользователя.

- Логи

Поддерживает запросы логов в реальном времени, включая поиск по времени, контейнеру и др.; поддерживает сохранение логов.

- Мониторинг и оповещение

Отображает данные мониторинга, включая использование различных ресурсов, количество подключений, время работы, типы команд и др.; поддерживает создание и обновление политик оповещений, с предустановленными метриками, включая количество подключений, переключения мастер-слейв, коэффициенты попадания по ключам и десять других метрик; поддерживает настройку оповещений для сотен метрик мониторинга (непредустановленных).

- Инспекция

Поддерживает инспекцию экземпляров с возможностью ручного и автоматического запуска; поддерживает отображение статистической информации результатов инспекции; поддерживает экспорт отчетов инспекции.

- Восстановление после сбоев

Поддерживает горячее резервирование для восстановления после сбоев с опциями асинхронной и синхронизированной по версии репликации для кластеров восстановления.

- Поддержка сетевого протокольного стека

Поддерживает двойной стек IPv4/IPv6, одиночный стек IPv4 и одиночный стек IPv6.

## **Kafka**

- Управление экземплярами

Поддерживает управление жизненным циклом экземпляров (создание, обновление, апгрейд, удаление, перезапуск, остановка).

- Конфигурация параметров



Преднастроенные шаблоны параметров, подходящие для продакшена, поддерживает создание и управление шаблонами параметров; поддерживает инициализацию параметров с помощью шаблонов, а также обновление, импорт и экспорт параметров.

- Методы доступа

Поддерживает различные методы экспонирования сервисов внутри и вне кластера, такие как NodePort, MetaLB, ALB и др. Поддерживает копирование информации для подключения в один клик, с предоставлением примеров кода для помощи.

- Топология

Поддерживает отображение топологии экземпляров и состояния работы каждого узла.

- Управление пользователями

Поддерживает управление пользователями экземпляров, включая создание, удаление пользователей и контроль прав доступа.

- Управление топиками

Поддерживает создание, обновление, удаление и синхронизацию топиков.

- Управление группами потребителей

Поддерживает просмотр информации о группах потребителей для экземпляров.

- Масштабирование

Поддерживает операции масштабирования, включая горизонтальное масштабирование узлов и вертикальное масштабирование характеристик.

- Терминал

Поддерживает командный терминал, облегчающий быстрый вход на узлы для операций пользователя.

- Логи

Поддерживает запросы логов в реальном времени, включая поиск по времени, контейнеру и др.; поддерживает сохранение логов.

- Мониторинг и оповещение

Отображает данные мониторинга, включая использование различных ресурсов, количество подключений, скорость запросов, размеры логов и др.; поддерживает создание и обновление политик оповещений, с предустановленными метриками,

включая количество отставших сообщений, частоту записи и десять других метрик; поддерживает настройку оповещений для сотен метрик мониторинга (непредустановленных).

- Инспекция

Поддерживает инспекцию экземпляров с возможностью ручного и автоматического запуска; поддерживает отображение статистической информации результатов инспекции; поддерживает экспорт отчетов инспекции.

- Восстановление после сбоев

Поддерживает горячее резервирование для восстановления после сбоев с опциями асинхронной и синхронизированной по версии репликации для кластеров восстановления.

- Поддержка сетевого протокольного стека

Поддерживает двойной стек IPv4/IPv6, одиночный стек IPv4 и одиночный стек IPv6.

## **RabbitMQ**

- Управление экземплярами

Поддерживает управление жизненным циклом экземпляров (создание, обновление, апгрейд, удаление, перезапуск, остановка).

- Конфигурация параметров

Преднастроенные шаблоны параметров, подходящие для продакшена, поддерживает создание и управление шаблонами параметров; поддерживает инициализацию параметров с помощью шаблонов, а также обновление, импорт и экспорт параметров.

- Методы доступа

Поддерживает различные методы экспонирования сервисов внутри и вне кластера, такие как NodePort, MetalB, ALB и др. Поддерживает копирование информации для подключения в один клик, с предоставлением примеров кода для помощи.

- Топология

Поддерживает отображение топологии экземпляров и состояния работы каждого узла.

- Масштабирование

Поддерживает операции масштабирования, включая горизонтальное масштабирование узлов и вертикальное масштабирование характеристик.

- Терминал

Поддерживает командный терминал, облегчающий быстрый вход на узлы для операций пользователя.

- Логи

Поддерживает запросы логов в реальном времени, включая поиск по времени, контейнеру и др.; поддерживает сохранение логов.

- Мониторинг и оповещение

Отображает данные мониторинга, включая использование различных ресурсов, количество подключений, количество сообщений в секунду, скорость производства, скорость потребления и др.; поддерживает создание и обновление политик оповещений, с предустановленными метриками, включая количество подключений, частоту записи и десять других метрик; поддерживает настройку оповещений для сотен метрик мониторинга (непредустановленных).

- Инспекция

Поддерживает инспекцию экземпляров с возможностью ручного и автоматического запуска; поддерживает отображение статистической информации результатов инспекции; поддерживает экспорт отчетов инспекции.

- Восстановление после сбоев

Поддерживает горячее резервирование для восстановления после сбоев с опциями асинхронной и синхронизированной по версии репликации для кластеров восстановления.

- Поддержка сетевого протокольного стека

Поддерживает двойной стек IPv4/IPv6, одиночный стек IPv4 и одиночный стек IPv6.

## PostgreSQL

- Управление экземплярами

Поддерживает управление жизненным циклом экземпляров (создание, обновление, апгрейд, удаление).

- Резервное копирование и восстановление

Поддерживает ручное и плановое резервное копирование с возможностью сохранения на внешнее хранилище S3, поддерживает восстановление данных.

- Масштабирование

Поддерживает операции масштабирования, включая горизонтальное масштабирование узлов и вертикальное масштабирование характеристик.

- Логи

Поддерживает запросы логов в реальном времени, включая поиск по времени, контейнеру и др.; поддерживает сохранение логов.

- Мониторинг и оповещение

Отображает данные мониторинга, включая использование различных ресурсов, количество подключений, TPS, операции блокировки таблиц, медленные запросы и др.; поддерживает создание и обновление политик оповещений, с предустановленными метриками, включая количество запросов чтения/записи InnoDB, количество подключений, количество потоков и десять других метрик; поддерживает настройку оповещений для сотен метрик мониторинга (непредустановленных).

- Инспекция

Поддерживает инспекцию экземпляров с возможностью ручного и автоматического запуска; поддерживает отображение статистической информации результатов инспекции; поддерживает экспорт отчетов инспекции.

## Установка

### Установка MWS Container Platform Data Services Essentials

1. В левой навигационной панели перейдите в **Marketplace > Cluster Plugins**.
2. Выберите кластер **global**.
3. Нажмите кнопку **Install** рядом с плагином **MWS Container Platform Data Services Essentials**.

### Установка MWS Container Platform Data Services RDS Framework

1. Войдите в платформу и перейдите на страницу **Administrator**.

2. В левой навигационной панели выберите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите MWS Container Platform Data Services RDS Framework, нажмите **Install** для перехода на страницу установки.

Параметры конфигурации:

| Параметр              | Рекомендуемая конфигурация                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Channel               | По умолчанию Channel — stable                                                                                                                   |
| Version               | Выберите желаемую версию для установки                                                                                                          |
| Installation Mode     | Cluster: Все пространства имён в кластере используют одного Operator для создания и управления экземплярами, что снижает использование ресурсов |
| Installation Location | Выберите Recommended: если его нет, он будет создан автоматически                                                                               |
| Upgrade Strategy      | Manual: при наличии новой версии в OperatorHub требуется ручное подтверждение для обновления Operator до последней версии                       |

4. На странице **Install Operator** выберите **Default Configuration**, затем нажмите **Install** для завершения установки MWS Container Platform Data Services RDS Framework.

## Установка операторов сервисов данных, доступных в платформе

### MySQL

1. Войдите в платформу и перейдите на страницу **Administrator**.
2. В левой навигационной панели выберите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите MWS Container Platform Database Service for MySQL, нажмите **Install** и перейдите на страницу установки.

Параметры конфигурации:

| Параметр | Рекомендуемая конфигурация              |
|----------|-----------------------------------------|
| Channel  | По умолчанию Channel — alpha            |
| Version  | Выберите желаемую версию для установки. |

| Параметр              | Рекомендуемая конфигурация                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Installation Mode     | Cluster: Все пространства имён в кластере используют одного Operator для создания и управления экземплярами, что снижает использование ресурсов |
| Installation Location | Выберите Recommended: если его нет, он будет создан автоматически                                                                               |
| Upgrade Strategy      | Manual: при наличии новой версии в OperatorHub требуется ручное подтверждение для обновления Operator до последней версии                       |

4. На странице **Install Operator** выберите **Default Configuration**, нажмите **Install** и завершите установку.

### **Redis**

1. Войдите в платформу и перейдите на страницу **Administrator**.
2. В левой навигационной панели выберите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите MWS Container Platform Cache Service for Redis OSS, нажмите **Install** и перейдите на страницу установки.

Параметры конфигурации:

| Параметр              | Рекомендуемая конфигурация                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Channel               | По умолчанию Channel — stable                                                                                                                   |
| Version               | Выберите желаемую версию для установки                                                                                                          |
| Installation Mode     | Cluster: Все пространства имён в кластере используют одного Operator для создания и управления экземплярами, что снижает использование ресурсов |
| Installation Location | Выберите Recommended: если его нет, он будет создан автоматически                                                                               |
| Upgrade Strategy      | Manual: при наличии новой версии в OperatorHub требуется ручное подтверждение для обновления Operator до последней версии                       |

4. На странице **Install Operator** выберите **Default Configuration**, нажмите **Install** и завершите установку.

### **Kafka**

1. Войдите в платформу и перейдите на страницу **Administrator**.
2. В левой навигационной панели выберите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите MWS Container Platform Streaming Service for Kafka или MWS Container Platform Streaming Service for Kafka 2 в зависимости от желаемой версии, нажмите **Install** и перейдите на страницу установки.

Параметры конфигурации:

| Параметр              | Рекомендуемая конфигурация                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Channel               | По умолчанию Channel — stable                                                                                                                   |
| Version               | Выберите желаемую версию для установки                                                                                                          |
| Installation Mode     | Cluster: Все пространства имён в кластере используют одного Operator для создания и управления экземплярами, что снижает использование ресурсов |
| Installation Location | Выберите Recommended: если его нет, он будет создан автоматически                                                                               |
| Upgrade Strategy      | Manual: при наличии новой версии в OperatorHub требуется ручное подтверждение для обновления Operator до последней версии                       |

4. На странице **Install Operator** выберите **Default Configuration**, нажмите **Install** и завершите установку.

### **RabbitMQ**

1. Войдите в платформу и перейдите на страницу **Administrator**.
2. В левой навигационной панели выберите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите MWS Container Platform Streaming Service for RabbitMQ, нажмите **Install** и перейдите на страницу установки.

Параметры конфигурации:

| Параметр | Рекомендуемая конфигурация |
|----------|----------------------------|
|----------|----------------------------|

|                       |                                                                                                                                                 |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Channel               | По умолчанию Channel — stable                                                                                                                   |
| Version               | Выберите желаемую версию для установки                                                                                                          |
| Installation Mode     | Cluster: Все пространства имён в кластере используют одного Operator для создания и управления экземплярами, что снижает использование ресурсов |
| Installation Location | Выберите Recommended: если его нет, он будет создан автоматически                                                                               |
| Upgrade Strategy      | Manual: при наличии новой версии в OperatorHub требуется ручное подтверждение для обновления Operator до последней версии                       |

4. На странице **Install Operator** выберите **Default Configuration**, нажмите **Install** и завершите установку.

### **PostgreSQL**

1. Войдите в платформу и перейдите на страницу **Administrator**.
2. В левой навигационной панели выберите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите PostgreSQL, нажмите **Install** и перейдите на страницу установки.

Параметры конфигурации:

| Параметр              | Рекомендуемая конфигурация                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Channel               | По умолчанию Channel — stable                                                                                                                   |
| Version               | Выберите желаемую версию для установки                                                                                                          |
| Installation Mode     | Cluster: Все пространства имён в кластере используют одного Operator для создания и управления экземплярами, что снижает использование ресурсов |
| Installation Location | Выберите Recommended: если его нет, он будет создан автоматически                                                                               |

| Параметр         | Рекомендуемая конфигурация                                                                                                |
|------------------|---------------------------------------------------------------------------------------------------------------------------|
| Upgrade Strategy | Manual: при наличии новой версии в OperatorHub требуется ручное подтверждение для обновления Operator до последней версии |

- На странице **Install Operator** выберите **Default Configuration**, нажмите **Install** и завершите установку.

## Обновление

MWS Container Platform Data Services Essentials разработан для автоматического обновления во время обновлений платформы, не требуя ручного вмешательства.

Система бесшовно выполняет все необходимые миграции и обновления в фоновом режиме.

MWS Container Platform Data Services RDS Framework будет выполнять обновления в соответствии с настроенной стратегией обновления:

- **Automatic:** Автообновления запускаются сразу после обнаружения новых версий компонентов.
- **Manual:** Требуется ручное подтверждение перед началом процесса обновления.

## MySQL-PXC

MWS Database Service для MySQL-PXC — это решение для высокодоступных баз данных с использованием синхронной репликации на базе Percona XtraDB Cluster, работающее в Kubernetes. Оно предоставляет возможность автоматизированного развёртывания и управления объектами внутри Percona XtraDB Cluster, а также может использоваться для запуска новых экземпляров Percona XtraDB Cluster или масштабирования существующих.

### Создание экземпляра

**Примечание:** После создания экземпляра некоторые параметры можно изменять динамически; конфигурацию планирования изменить нельзя.

### CLI

- Создайте пароль:

```
kubectrl -n <namespace> create secret generic <name> \
 --from-literal=clustercheck=<password> \
 --from-literal=monitor=<password> \
 --from-literal=operator=<password> \
 --from-literal=proxyadmin=<password> \
 --from-literal=root=<password> \
 --from-literal=xtrabackup=<password>
```

Где:



- <name> — имя секрета с паролями, используемое для идентификации экземпляра, должно быть уникальным; рекомендуется использовать то же имя, что и у экземпляра, которое задаётся в поле spec.pxc.secretsName экземпляра.
- <namespace> — пространство имён, к которому принадлежит экземпляр.
- <password> — пароль; для перечисленных пользователей можно задать разные пароли.

## 2. Создайте экземпляр:

```
kubectl -n <namespace> apply -f - <<EOF
apiVersion: middleware.alauda.io/v1
kind: Mysql
metadata:
 annotations:
 middleware.upgrade.component.version: ""
 middleware.upgrade.crVersion: ""
 labels:
 mysql/arch: pxc
 name: <name>
 namespace: <namespace>
spec:
 params:
 mysql:
 client:
 default_character_set: "utf8mb4"
 mysql:
 default_character_set: "utf8mb4"
 prompt: mysql(\u@\h:\d) >
 mysqld:
 autocommit: "ON"
 big_tables: "OFF"
 binlog_cache_size: "32768"
 binlog_checksum: "crc32"
 binlog_format: "ROW"
 binlog_row_image: "FULL"
 binlog_rows_query_log_events: "1"
 bulk_insert_buffer_size: "8388608"
 character_set_server: "utf8mb4"
 collation_server: "utf8mb4_general_ci"
 connect_timeout: "28800"
 default_authentication_plugin: "mysql_native_password"
 default_password_lifetime: "0"
```

default\_storage\_engine: "INNODB"  
default\_time\_zone: "+8:00"  
default\_tmp\_storage\_engine: "INNODB"  
event\_scheduler: "ON"  
expire\_logs\_days: "2"  
explicit\_defaults\_for\_timestamp: "1"  
general\_log: "OFF"  
init\_connect: "SET NAMES utf8mb4"  
innodb\_adaptive\_flushing: "ON"  
innodb\_adaptive\_flushing\_lwm: "10"  
innodb\_adaptive\_hash\_index: "OFF"  
innodb\_autoinc\_lock\_mode: "2"  
innodb\_buffer\_pool\_dump\_at\_shutdown: "1"  
innodb\_buffer\_pool\_dump\_pct: "40"  
innodb\_buffer\_pool\_instances: "2"  
innodb\_buffer\_pool\_load\_at\_startup: "1"  
innodb\_buffer\_pool\_size: "1536M"  
innodb\_checksum\_algorithm: "crc32"  
innodb\_checksums: "ON"  
innodb\_cmp\_per\_index\_enabled: "OFF"  
innodb\_compression\_level: "6"  
innodb\_deadlock\_detect: "1"  
innodb\_default\_row\_format: "DYNAMIC"  
innodb\_doublewrite: "1"  
innodb\_fast\_shutdown: "1"  
innodb\_file\_format: "Barracuda"  
innodb\_file\_format\_check: "ON"  
innodb\_file\_format\_max: "Barracuda"  
innodb\_file\_per\_table: "ON"  
innodb\_flush\_log\_at\_timeout: "1"  
innodb\_flush\_log\_at\_trx\_commit: "1"  
innodb\_flush\_method: "O\_DIRECT\_NO\_FSYNC"  
innodb\_flush\_neighbors: "0"  
innodb\_flushing\_avg\_loops: "30"  
innodb\_force\_recovery: "0"  
innodb\_log\_file\_size: "1G"  
innodb\_log\_files\_in\_group: "2"  
innodb\_lru\_scan\_depth: "2048"  
innodb\_max\_dirty\_pages\_pct: "90"  
innodb\_max\_dirty\_pages\_pct\_lwm: "10"

innodb\_max\_purge\_lag: "0"  
innodb\_max\_purge\_lag\_delay: "0"  
innodb\_max\_undo\_log\_size: "2G"  
innodb\_numa\_interleave: "1"  
innodb\_online\_alter\_log\_max\_size: "32M"  
innodb\_page\_cleaners: "2"  
innodb\_page\_size: "16384"  
innodb\_print\_all\_deadlocks: "1"  
innodb\_print\_lock\_wait\_timeout\_info: "ON"  
innodb\_purge\_batch\_size: "300"  
innodb\_purge\_rseg\_truncate\_frequency: "128"  
innodb\_purge\_threads: "2"  
innodb\_random\_read\_ahead: "OFF"  
innodb\_read\_ahead\_threshold: "56"  
innodb\_read\_io\_threads: "4"  
innodb\_rollback\_on\_timeout: "0"  
innodb\_rollback\_segments: "128"  
innodb\_sort\_buffer\_size: "1M"  
innodb\_status\_output: "0"  
innodb\_status\_output\_locks: "1"  
innodb\_strict\_mode: "1"  
innodb\_support\_xa: "1"  
innodb\_undo\_log\_truncate: "1"  
innodb\_undo\_logs: "128"  
innodb\_use\_native\_aio: "ON"  
innodb\_write\_io\_threads: "8"  
interactive\_timeout: "28800"  
internal\_tmp\_disk\_storage\_engine: "INNODB"  
join\_buffer\_size: "262144"  
lock\_wait\_timeout: "50"  
log\_bin: "bin"  
log\_error\_verbosity: "3"  
log\_queries\_not\_using\_indexes: "1"  
log\_slow\_admin\_statements: "1"  
log\_slow\_slave\_statements: "1"  
log\_throttle\_queries\_not\_using\_indexes: "10"  
long\_query\_time: "20"  
max\_allowed\_packet: "64M"  
max\_binlog\_size: "1024M"  
max\_connect\_errors: "10000"

```

max_connections: "300"
max_heap_table_size: "16M"
max_join_size: "18446744073709551615"
max_user_connections: "60"
min_examined_row_limit: "100"
open_files_limit: "65535"
pxc_strict_mode: "ENFORCING"
query_cache_type: "0"
read_buffer_size: "131072"
read_rnd_buffer_size: "262144"
show_compatibility_56: "OFF"
skip_external_locking: "1"
skip_name_resolve: "1"
skip_symbolic_links: "ON"
slow_query_log: "0"
sort_buffer_size: "262144"
sql_mode:
"ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ER
ROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"
sync_binlog: "1"
table_open_cache: "4096"
table_open_cache_instances: "2"
thread_handling: "pool-of-threads"
thread_pool_idle_timeout: "60"
thread_pool_max_threads: "1000"
thread_pool_oversubscribe: "8"
thread_pool_size: "4"
tmp_table_size: "16M"
transaction_isolation: "READ-COMMITTED"
wait_timeout: "28800"
wsrep_OSU_method: "TOI"
wsrep_RSU_commit_timeout: "5000"
wsrep_auto_increment_control: "ON"
wsrep_causal_reads: "OFF"
wsrep_certification_rules: "strict"
wsrep_certify_nonPK: "ON"
wsrep_convert_LOCK_to_trx: "OFF"
wsrep_debug: "OFF"
wsrep_desync: "OFF"
wsrep_dirty_reads: "OFF"
wsrep_drupal_282555_workaround: "OFF"

```

wsrep\_forced\_binlog\_format: "ROW"  
wsrep\_load\_data\_splitting: "ON"  
wsrep\_log\_conflicts: "OFF"  
wsrep\_max\_ws\_rows: "0"  
wsrep\_max\_ws\_size: "2147483647"  
wsrep\_on: "ON"  
wsrep\_preordered: "OFF"  
wsrep\_recover: "OFF"  
wsrep\_reject\_queries: "NONE"  
wsrep\_replicate\_myisam: "OFF"  
wsrep\_restart\_slave: "OFF"  
wsrep\_retry\_autocommit: "1"  
wsrep\_slave\_FK\_checks: "ON"  
wsrep\_slave\_UK\_checks: "OFF"  
wsrep\_slave\_threads: "4"  
wsrep\_sync\_wait: "0"

pxc:

proxysql:

resources:

limits:

cpu: 500m

memory: 512Mi

requests:

cpu: 500m

memory: 512Mi

size: 3

volumeSpec:

persistentVolumeClaim:

resources:

requests:

storage: <size>Gi

storageClassName: <storageClassName>

pxc:

resources:

limits:

cpu: "2"

memory: 4Gi

requests:

cpu: "2"

memory: 4Gi

```

size: 3
volumeSpec:
 persistentVolumeClaim:
 resources:
 requests:
 storage: <size>Gi
 storageClassName: <storageClassName>
 secretsName: <name>
runAsRoot: false
upgradeOption:
 autoUpgrade: true
version: "5.7"
EOF

```

Где:

- <name> — имя экземпляра, используемое для идентификации, должно быть уникальным.
  - <namespace> — пространство имён, к которому принадлежит экземпляр.
  - <spec.pxc.secretsName> — имя секрета с паролями, созданного на шаге 1.
  - <spec.params.mysql> — конфигурация параметров MySQL, опции соответствуют MySQL 5.7.
  - <spec.pxc.pxc.resources> — конфигурация ресурсов для Pod PXC, соответствует конфигурации ресурсов Pod.
  - <spec.pxc.pxc.volumeSpec.persistentVolumeClaim> — конфигурация тома данных для Pod PXC.
  - <spec.pxc.pxc.size> — количество Pod PXC.
  - <spec.pxc.proxysql.resources> — конфигурация ресурсов для Pod ProxySQL, соответствует конфигурации ресурсов Pod.
  - <spec.pxc.proxysql.volumeSpec.persistentVolumeClaim> — конфигурация тома данных для Pod ProxySQL.
  - <spec.pxc.proxysql.size> — количество Pod ProxySQL.
3. Дождитесь создания экземпляра. Выполните команду и дождитесь, пока поле STATE в результате не изменится на ready:

```
kubectl get mysql -n <namespace> <name> -w
```

#### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **MySQL-PXC**.
3. Выберите пространство имён.
4. Нажмите Create MySQL-PXC Instance.

## 5. Настройте параметры:

| Элемент конфигурации         | Описание                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Конфигурация параметров      | Шаблон параметров: можно выбрать системный шаблон или пользовательский                                                                                                                                                                                                                                                                                                                                                |
| Конфигурация ресурсов        | Спецификации: ориентируйтесь на производительность MySQL, предоставляемую платформой, исходя из объема данных, и настройте CPU и память соответственно                                                                                                                                                                                                                                                                |
| Конфигурация учетных записей | Пароль администратора: храните пароль в безопасности. Для снижения риска утечки рекомендуется регулярно менять пароль                                                                                                                                                                                                                                                                                                 |
| Конфигурация планирования    | Метки узлов: выберите подходящие метки для фильтрации доступных узлов в кластере. Поды будут запланированы на доступных узлах.<br>Примечание: эту конфигурацию нельзя изменить после создания экземпляра                                                                                                                                                                                                              |
| Pod Tolerations              | Если на узле установлены метки (taints), только экземпляры с заданными Pod Tolerations могут быть запланированы на этом узле. Правила соответствия: <ul style="list-style-type: none"><li>• Equal: ключ, значение и эффект Pod Tolerations должны полностью совпадать с меткой узла.</li><li>• Exists: ключ и эффект Pod Tolerations должны совпадать с ключом и эффектом метки узла, значение игнорируется</li></ul> |

Для получения дополнительной информации обратитесь к официальной документации Kubernetes.

## 6. Нажмите **Create** и дождитесь изменения статуса на Running.

### Удаление экземпляра

**Внимание:** Если при удалении выбрана опция не сохранять данные экземпляра, они будут удалены безвозвратно и не подлежат восстановлению.

### CLI

#### 1. Определите, сохранять ли данные:

- Сохранить данные:

```
kubectl patch mysql -n <namespace> <name> --type json -p '{"op": "add", "path": "/metadata/finalizers/-", "value": "delete-pvc"}'
```

- Удалить данные:

```
current_finalizers=$(kubectl get mysql -n <namespace> <name> -o
jsonpath='{.metadata.finalizers}' | jq --arg f "delete-pvc" 'map(select(. != $f))')
kubectl patch mysql -n <namespace> <name> --type merge -p
{"\metadata\":"finalizers\":"$current_finalizers}"
```

2. Удалите экземпляр:

```
kubectl delete mysql -n <namespace> <name>
```

### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **MySQL-PXC**.
3. Нажмите на имя пространства имён.
4. Нажмите на имя экземпляра.
5. Нажмите **Actions > Delete** и подтвердите действие.

## MySQL-MGR

MWS Database Service for MySQL-MGR — это решение для высокодоступных баз данных, основанное на технологии MySQL Group Replication и работающее на Kubernetes. Оно предоставляет возможности для автоматизированного развёртывания и управления кластерами MySQL Group Replication.

### Создание экземпляра

#### CLI

1. Создайте пароль:

```
kubectl -n ${namespace} create secret generic mgr-${instance_name}-password \
--from-literal=clusterchecker={password} \
--from-literal=exporter=${password} \
--from-literal=manage=${password} \
--from-literal=root=${password}
```

Где:

- `${instance_name}` — имя экземпляра, используется для идентификации экземпляра и должно быть уникальным.
- `${namespace}` — пространство имён, к которому принадлежит экземпляр.
- `${password}` — пароль, при этом для разных пользователей можно задать разные пароли.

2. Создайте CR экземпляра:

```
kubectl apply -n $namespace -f - <<EOF
apiVersion: middleware.alauda.io/v1
kind: Mysql
metadata:
 labels:
 mysql/arch: mgr
```



```
name: ${instance_name}
spec:
 mgr:
 enableStorage: true
 members: 3
 monitor:
 enable: true
 resources:
 server:
 limits:
 cpu: "2"
 memory: 4Gi
 requests:
 cpu: "2"
 memory: 4Gi
 router:
 replicas: 2
 resources:
 limits:
 cpu: 800m
 memory: 640Mi
 requests:
 cpu: 800m
 memory: 640Mi
 svcRO:
 type: ClusterIP
 svcRW:
 type: ClusterIP
 volumeClaimTemplate:
 spec:
 resources:
 requests:
 storage: 20Gi
 storageClassName: sc-topolvm
 params:
 mysql:
 mysqld:
 binlog_expire_logs_seconds: "604800"
 binlog_format: ROW
 character_set_server: utf8mb4
```

default\_storage\_engine: InnoDB  
default\_time\_zone: +08:00  
disabled\_storage\_engines: MyISAM  
event\_scheduler: ON  
general\_log: OFF  
innodb\_adaptive\_flushing: ON  
innodb\_adaptive\_hash\_index: OFF  
innodb\_autoinc\_lock\_mode: "2"  
innodb\_buffer\_pool\_chunk\_size: "134217728"  
innodb\_buffer\_pool\_dump\_at\_shutdown: ON  
innodb\_buffer\_pool\_dump\_pct: "25"  
innodb\_buffer\_pool\_instances: "2"  
innodb\_buffer\_pool\_load\_abort: OFF  
innodb\_buffer\_pool\_load\_at\_startup: ON  
innodb\_buffer\_pool\_size: 1536M  
innodb\_deadlock\_detect: ON  
innodb\_disable\_sort\_file\_cache: OFF  
innodb\_fast\_shutdown: "1"  
innodb\_file\_per\_table: ON  
innodb\_flush\_log\_at\_trx\_commit: "1"  
innodb\_flush\_method: O\_DIRECT\_NO\_FSYNC  
innodb\_flush\_neighbors: "0"  
innodb\_io\_capacity: "4000"  
innodb\_io\_capacity\_max: "4294967295"  
innodb\_max\_purge\_lag: "0"  
innodb\_max\_undo\_log\_size: "1073741824"  
innodb\_online\_alter\_log\_max\_size: "134217728"  
innodb\_page\_cleaners: "4"  
innodb\_purge\_batch\_size: "600"  
innodb\_purge\_rseg\_truncate\_frequency: "128"  
innodb\_purge\_threads: "4"  
innodb\_read\_io\_threads: "4"  
innodb\_redo\_log\_capacity: 1G  
innodb\_rollback\_segments: "128"  
innodb\_strict\_mode: ON  
innodb\_thread\_concurrency: "5"  
innodb\_undo\_log\_truncate: ON  
innodb\_write\_io\_threads: "4"  
interactive\_timeout: "3600"  
log\_bin: bin

log\_bin\_trust\_function\_creators: OFF  
log\_output: FILE  
loose\_group\_replication\_communication\_max\_message\_size: "10485760"  
loose\_group\_replication\_exit\_state\_action: READ\_ONLY  
loose\_group\_replication\_flow\_control\_applier\_threshold: "1000"  
loose\_group\_replication\_flow\_control\_certifier\_threshold: "1000"  
loose\_group\_replication\_member\_expel\_timeout: "5"  
loose\_group\_replication\_message\_cache\_size: 256M  
loose\_group\_replication\_paxos\_single\_leader: ON  
loose\_group\_replication\_poll\_spin\_loops: "0"  
loose\_group\_replication\_transaction\_size\_limit: "150000000"  
loose\_group\_replication\_unreachable\_majority\_timeout: "0"  
loose\_group\_replication\_xcom\_ssl\_accept\_retries: "10"  
max\_allowed\_packet: "67108864"  
max\_binlog\_size: "1073741824"  
max\_connect\_errors: "1000000"  
max\_connections: "512"  
max\_heap\_table\_size: 16M  
max\_prepared\_stmt\_count: "8192"  
performance\_schema: ON  
replica\_parallel\_type: LOGICAL\_CLOCK  
replica\_parallel\_workers: "4"  
skip\_external\_locking: ON  
skip\_name\_resolve: ON  
sort\_buffer\_size: "262144"  
sql\_generate\_invisible\_primary\_key: ON  
sql\_mode: NO\_ENGINE\_SUBSTITUTION  
sync\_binlog: "1"  
table\_definition\_cache: "2000"  
table\_open\_cache\_instances: "16"  
thread\_handling: pool-of-threads  
thread\_pool\_max\_threads: "128"  
thread\_pool\_oversubscribe: "32"  
thread\_pool\_size: "2"  
tmp\_table\_size: 16M  
wait\_timeout: "3600"  
router:  
  DEFAULT:  
    max\_total\_connections: "300"  
logger:

```
level: info
upgradeOption:
 autoUpgrade: false
version: "8.0"
```

EOF

3. Проверьте статус экземпляра:

```
kubectl get mysql -n {namespace} {instance_name} -w
```

### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **MySQL-MGR**.
3. Выберите пространство имён.
4. Нажмите **Create Instance**.
5. Настройте базовые параметры:
  - Имя экземпляра: уникальный идентификатор
  - Конфигурация архитектуры: Single Master (по умолчанию) / Multi-Master
6. Сконфигурируйте ресурсы:
  - Количество реплик: рекомендуется начинать с 3 узлов
  - Спецификации: значения запросов и лимитов CPU/памяти для участников MySQL
  - Класс хранения: выберите класс хранения типа TopoLVM
  - Размер хранилища: оцените потребности в хранилище на основе бизнес-требований
7. Настройте учётные записи: настройте пароль администратора root.
8. Сконфигурируйте подключение: Метод доступа: внешний доступ / внутренний доступ.
9. Нажмите **Create** и дождитесь изменения статуса на Running.

### Удаление экземпляра

Удаление экземпляра автоматически удалит все безопасно удаляемые ресурсы, созданные этим экземпляром, такие как StatefulSets, Deployments, ConfigMaps, Secrets и Services. Специальные ресурсы, которые могут содержать важные данные, необходимо удалять вручную, например PersistentVolumeClaims и резервные ресурсы, связанные с MySQL-MGR, а именно mysqlbackups.mysql.middleware.alauda.io и связанные с ним фактические данные.

### CLI

1. Удалите экземпляр:

```
kubectl delete mysql -n ${namespace} ${name}
```
2. Очистите постоянные данные:

```
kubectl get pvc -n ${namespace} -l
v1alpha1.mysql.middleware.alauda.io/cluster=${name} -o name | xargs kubectl delete -
n ${namespace}
```

## Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **MySQL-MGR**.
3. Выберите пространство имён.
4. Нажмите на имя экземпляра.
5. Нажмите Actions > Delete.
6. Выберите параметры удаления:
  - По желанию: удалить PersistentVolumeClaim
7. Подтвердите удаление:
  - Проверьте связанные ресурсы
  - Введите имя экземпляра для окончательного подтверждения удаления

## Redis

MWS Cache Service for Redis OSS — это система хранения структур данных в памяти с открытым исходным кодом, которая поддерживает различные структуры данных, такие как строки, хэши, списки, множества и отсортированные множества, а также предоставляет функциональность, такую как сохранение данных, высокая доступность и распределённый кластер. Благодаря высокой производительности и низкой задержке, она широко используется в сценариях, таких как кэширование, очереди сообщений и анализ данных в реальном времени.

### Выбор архитектуры

В Redis поддерживаются типы архитектуры:

- Cluster – кластер,
- Sentinel – Redis Sentinel,
- Standalone – Одиночный Redis. Не имеет реплик и отсутствует функция переключения на резервный узел, поэтому используйте её только в средах разработки и тестирования.

### Создание экземпляра

**Примечание:** Описания параметров, используемых для создания экземпляра через CLI, приведено в документации API Redis.

#### Sentinel CLI

Создайте экземпляр Redis 6.0 в режиме Sentinel:

```
$ cat << EOF | kubectl -n default create -f -
apiVersion: middleware.alauda.io/v1
kind: Redis
metadata:
 name: s6
spec:
```

arch: sentinel  
customConfig:  
  databases: "16"  
  hz: "10"  
  save: 60 10000 300 100 600 1  
  timeout: "0"  
exporter:  
  enabled: true  
  resources:  
    limits:  
      cpu: 100m  
      memory: 384Mi  
    requests:  
      cpu: 50m  
      memory: 128Mi  
replicas:  
  sentinel:  
    master: 1  
    slave: 1  
resources:  
  limits:  
    cpu: "1"  
    memory: 2Gi  
  requests:  
    cpu: "1"  
    memory: 2Gi  
sentinel:  
  monitorConfig:  
    down-after-milliseconds: "30000"  
    failover-timeout: "180000"  
    parallel-syncs: "1"  
  replicas: 3  
  resources:  
    limits:  
      cpu: 100m  
      memory: 128Mi  
    requests:  
      cpu: 100m  
      memory: 128Mi  
version: "6.0"

EOF

## Cluster CLI

Создайте экземпляр Redis 6.0 в режиме Cluster:

```
$ cat << EOF | kubectl -n default create -f -
apiVersion: middleware.alauda.io/v1
kind: Redis
metadata:
 name: c6
spec:
 affinityPolicy: AntiAffinityInSharding
 arch: cluster
 customConfig:
 hz: "10"
 save: 60 10000 300 100 600 1
 timeout: "0"
 exporter:
 enabled: true
 resources:
 limits:
 cpu: 100m
 memory: 384Mi
 requests:
 cpu: 50m
 memory: 128Mi
 expose:
 type: NodePort
 replicas:
 cluster:
 shard: 3
 slave: 1
 resources:
 limits:
 cpu: 200m
 memory: 200Mi
 requests:
 cpu: 200m
 memory: 200Mi
 version: "6.0"
```

EOF

## Standalone CLI

Создайте экземпляр Redis 6.0 в режиме Standalone:

```
$ cat << EOF | kubectl -n default create -f -
apiVersion: middleware.alauda.io/v1
kind: Redis
metadata:
 name: standalone
spec:
 arch: standalone
 customConfig:
 databases: "16"
 hz: "10"
 save: 60 10000 300 100 600 1
 timeout: "0"
 exporter:
 enabled: true
 resources:
 limits:
 cpu: 100m
 memory: 384Mi
 requests:
 cpu: 50m
 memory: 128Mi
 replicas:
 sentinel:
 master: 1
 resources:
 limits:
 cpu: "1"
 memory: 1Gi
 requests:
 cpu: "1"
 memory: 1Gi
 version: "6.0"
EOF
```

#### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **Redis**.
3. Выберите пространство имён.
4. Нажмите **Create Redis Instance**.
5. Настройте параметры:



| Элемент конфигурации      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Конфигурация параметров   | Шаблон параметров: можно выбрать системный шаблон или пользовательский                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Конфигурация ресурсов     | Спецификации: ориентируйтесь на производительность MySQL, предоставляемую платформой, исходя из объёма данных, и настройте CPU и память соответственно                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Конфигурация подключения  | <ul style="list-style-type: none"> <li>• Установить пароль по умолчанию: Рекомендуется установить пароль для пользователя redis default при запуске экземпляра</li> <li>• Метод доступа: Если приложения и экземпляры Redis изменяются в разных кластерах нагрузки, используйте метод внешнего доступа. Метод внешнего доступа поддерживает LoadBalancer и NodePort, где LoadBalancer рекомендуется для производственных сред. Если приложения и экземпляры Redis изменяются в одном и том же кластере нагрузки, используйте метод доступа Внутри кластера.</li> <li>• Порт хоста: При использовании NodePort для внешнего доступа вы можете указать номера портов службы: <ul style="list-style-type: none"> <li>○ Режим Sentinel: При обновлении экземпляра порты не могут быть обменены; если вам нужно поменять порты, сначала обновите до другого незанятого порта, затем пересоздайте их.</li> <li>○ Режим кластер: При обновлении экземпляра, если нужно заменить только один порт, вы можете ввести или изменить соответствующий порт узла в YAML.</li> </ul> </li> </ul> |
| Конфигурация планирования | <ul style="list-style-type: none"> <li>• Политика Anti-Affinity: Для обеспечения высокой надежности и доступности в различных бизнес-сценариях платформа предоставляет политику антипринадлежности для Кластерного режима, где Принудительная антипринадлежность является автоматически применяемой настройкой по умолчанию.<br/>Примечание: После создания экземпляра эту настройку нельзя изменить.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Элемент конфигурации | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | <ul style="list-style-type: none"> <li>Метки узлов: Фильтрация доступных узлов в текущем кластере через метки. Поды будут планироваться на доступные узлы.<br/>Примечание: После создания экземпляра эту настройку нельзя изменить.</li> <li>Pod Tolerations: Если на узле установлены метки (taints), только экземпляры с заданными Pod Tolerations могут быть запланированы на этом узле.<br/>Правила соответствия: <ul style="list-style-type: none"> <li>Equal: ключ, значение и эффект Pod Tolerations должны полностью совпадать с меткой узла.</li> <li>Exists: ключ и эффект Pod Tolerations должны совпадать с ключом и эффектом метки узла, значение игнорируется</li> </ul> </li> </ul> |

Для получения дополнительной информации обратитесь к официальной документации Kubernetes.

6. Нажмите **Create** и дождитесь изменения статуса на Running.

### Проверка

После создания экземпляра вы можете использовать следующую команду для проверки статуса экземпляра:

```
-n default get redis
```

| NAME       | ARCH       | VERSION | ACCESS   | STATUS       | MESSAGE | BUNDLE VERSION | AUTOUPGRADE | AGE   |
|------------|------------|---------|----------|--------------|---------|----------------|-------------|-------|
| c5         | cluster    | 5.0     | NodePort | Ready        | 4.0.1   |                |             | 2m46s |
| c6         | cluster    | 6.0     |          | Initializing | 4.0.1   |                |             | 3m26s |
| c7         | cluster    | 7.2     | NodePort | Initializing | 4.0.1   |                |             | 98s   |
| s5         | sentinel   | 5.0     |          | Ready        | 4.0.1   |                |             | 3m10s |
| s6         | sentinel   | 6.0     |          | Ready        | 4.0.1   |                |             | 25m   |
| s7         | sentinel   | 7.2     | NodePort | Ready        | 4.0.1   |                |             | 2m15s |
| standalone | standalone | 6.0     |          | Initializing | 4.0.1   |                |             | 6s    |

Значения полей итоговой таблицы следующие:

| Поле | Описание                                                                                                                                                                                                                                                                                                              |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME | Имя экземпляра                                                                                                                                                                                                                                                                                                        |
| ARCH | <p>Это значение соответствует архитектуре Redis.</p> <ul style="list-style-type: none"> <li>cluster: Кластерный режим</li> <li>sentinel: Режим Sentinel</li> <li>standalone: Одиночный Redis</li> </ul> <p>Эта строка показывает, управляет ли данный ресурс CR одиночным, Sentinel или кластерным режимом Redis.</p> |

| Поле           | Описание                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VERSION        | В настоящее время поддерживаются 3 версии: 5.0, 6.0, 7.2                                                                                                                                                                                                                                                                                                                                                         |
| ACCESS         | Методы доступа, поддерживаемые экземпляром: <ul style="list-style-type: none"> <li>ClusterIP: Экземпляр предоставляет услуги через ClusterIP</li> <li>NodePort: Экземпляр предоставляет услуги через NodePort</li> <li>LoadBalancer: Экземпляр предоставляет услуги через LoadBalancer</li> </ul>                                                                                                                |
| STATUS         | Текущий статус экземпляра, который может включать в себя следующие состояния: <ul style="list-style-type: none"> <li>Initializing: Экземпляр инициализируется или обновляется</li> <li>Rebalancing: Экземпляр кластера масштабируется вверх или вниз</li> <li>Error: Экземпляр столкнулся с необратимой ошибкой</li> <li>Paused: Экземпляр был временно приостановлен</li> <li>Ready: Экземпляр готов</li> </ul> |
| MESSAGE        | Причина, по которой экземпляр находится в текущем статусе                                                                                                                                                                                                                                                                                                                                                        |
| BUNDLE VERSION | Номер версии оператора, управляющего экземпляром                                                                                                                                                                                                                                                                                                                                                                 |
| AUTOUPGRADE    | Включены ли автоматические обновления для экземпляра                                                                                                                                                                                                                                                                                                                                                             |
| AGE            | Как давно был создан экземпляр                                                                                                                                                                                                                                                                                                                                                                                   |

### Удаление экземпляра

Удаление экземпляра приведет не только к удалению созданных ресурсов CR, но и к удалению StatefulSets, ConfigMaps, Secrets, PersistentVolumeClaims и других ресурсов, созданных оператором на основе ресурсов CR.

### CLI

Если у экземпляра не включена поддержка сохранения данных, или если вам необходимо сохранить постоянные данные экземпляра, выполните следующую команду для удаления экземпляра:

```
удалить экземпляр redis напрямую
$ kubectl -n default delete redis <Имя Экземпляра>
```

Если нужно удалить экземпляр и также удалить его заявку на постоянный том, выполните следующие команды:

```
добавить финализатор delete-pvc к экземпляру
$ kubectl -n default patch redis <Имя Экземпляра> --type='merge' -p '{"metadata": {"finalizers": ["delete-pvc"]}]'
```

```
удалить экземпляр redis с pvc
```

```
$ kubectl -n default delete redis <Имя Экземпляра>
```

### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **Redis**.
3. Выберите пространство имён.
4. Из списка экземпляров выберите экземпляр, который хотите удалить и перейдите на страницу сведений об экземпляре.
5. Нажмите **Actions > Delete**. Для экземпляров с включенной сохранностью данных можно отметить **Delete PVC**, чтобы освободить место, или оставить его неотмеченным, чтобы сохранить данные.
6. Введите имя экземпляра, который нужно удалить.
7. Нажмите **Delete**.

### Kafka

MWS Streaming Service for Kafka — это распределённая платформа потоковой обработки данных на базе Kafka, работающая в Kubernetes. В качестве распределённой очереди сообщений она эффективно передаёт и хранит большие объёмы данных между различными системами. Поддерживает несколько форматов данных и способна обрабатывать потоки данных в реальном времени, что делает её широко используемой в таких областях, как сбор логов, событийно-ориентированные архитектуры, аналитика в реальном времени и интеграция данных.

### Создание экземпляра

#### CLI

```
cat << EOF | kubectl -n default create -f -
apiVersion: middleware.alauda.io/v1
kind: RdsKafka
metadata:
 name: my-cluster
spec:
 entityOperator:
 topicOperator:
 resources:
 limits:
 cpu: 1
 memory: 2Gi
 requests:
 cpu: 1
 memory: 2Gi
 userOperator:
```

```
resources:
 limits:
 cpu: 1
 memory: 2Gi
 requests:
 cpu: 1
 memory: 2Gi
tlsSidecar:
 resources:
 limits:
 cpu: 200m
 memory: 128Mi
 requests:
 cpu: 200m
 memory: 128Mi
version: 3.8
replicas: 3
config:
 auto.create.topics.enable: "false"
 auto.leader.rebalance.enable: "true"
 background.threads: "10"
 compression.type: producer
 default.replication.factor: "3"
 delete.topic.enable: "true"
 log.retention.hours: "168"
 log.roll.hours: "168"
 log.segment.bytes: "1073741824"
 message.max.bytes: "1048588"
 min.insync.replicas: "1"
 num.io.threads: "8"
 num.network.threads: "3"
 num.recovery.threads.per.data.dir: "1"
 num.replica.fetchers: "1"
 unclean.leader.election.enable: "false"
resources:
 limits:
 cpu: 2
 memory: 4Gi
 requests:
 cpu: 2
```

```

 memory: 4Gi
storage:
 size: 1Gi
 # Замените на доступный класс хранилища
 class: local-path
 deleteClaim: false
kafka:
 listeners:
 plain: {}
 external:
 type: nodeport
 tls: false
zookeeper:
 # В настоящее время такой же, как у Kafka брокера
 replicas: 3
 resources:
 limits:
 cpu: 1
 memory: 2Gi
 requests:
 cpu: 1
 memory: 2Gi
 # Хранилище поддерживается такое же, как у Kafka брокера
 storage:
 size: 1Gi
 # Замените на доступный класс хранилища
 class: local-path
 deleteClaim: false
EOF

```

#### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **Kafka**.
3. Выберите пространство имён.
4. Нажмите Create Kafka Instance.
5. Настройте параметры:

| Элемент конфигурации    | Описание                                                               |
|-------------------------|------------------------------------------------------------------------|
| Конфигурация параметров | Шаблон параметров: можно выбрать системный шаблон или пользовательский |

| Элемент конфигурации      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Узлы Kafka                | <ul style="list-style-type: none"> <li>• Количество брокерских узлов: Для обеспечения высокой доступности брокера каждый брокер будет запланирован на разных узлах кластера.</li> <li>• Класс хранилища: Если в выпадающем списке отсутствует класс хранилища, обратитесь к администратору платформы для его добавления</li> </ul>                                                                                                                                                                                                                                                                                                                                                                 |
| Метод доступа             | <ul style="list-style-type: none"> <li>• Метод аутентификации: Для обеспечения безопасной передачи данных рекомендуется включить зашифрованную аутентификацию. Например: метод прослушивания TLS, метод аутентификации SCRAM - SHA – 512.</li> <li>• Порт хоста: При использовании NodePort для доступа к кластеру и открытии указанного порта хоста вы можете указать номер порта сервиса. Примечание: при обновлении экземпляра порты не взаимозаменяемы. Если необходимо поменять порты местами, сначала обновите на другой свободный порт, затем укажите нужный</li> </ul>                                                                                                                     |
| Конфигурация планирования | <ul style="list-style-type: none"> <li>• Метки узлов: Фильтрация доступных узлов в текущем кластере через метки. Поды будут планироваться на доступные узлы. Примечание: После создания экземпляра эту настройку нельзя изменить.</li> <li>• Pod Tolerations: Если на узле установлены метки (taints), только экземпляры с заданными Pod Tolerations могут быть запланированы на этом узле. Правила соответствия: <ul style="list-style-type: none"> <li>○ Equal: ключ, значение и эффект Pod Tolerations должны полностью совпадать с меткой узла.</li> <li>○ Exists: ключ и эффект Pod Tolerations должны совпадать с ключом и эффектом метки узла, значение игнорируется</li> </ul> </li> </ul> |

Для получения дополнительной информации обратитесь к официальной документации Kubernetes.

6. Нажмите **Create** и дождитесь изменения статуса на Running.

### Проверка

После создания экземпляра вы можете проверить его статус с помощью следующей команды:

```
$ kubectl get rdskafka -n <namespace> -o=custom-
columns=NAME:.metadata.name,VERSION:.spec.version,STATUS:.status.phase,MESSAGE
:.status.reason,CreationTimestamp:.metadata.creationTimestamp
NAME VERSION STATUS MESSAGE
CreationTimestamp
my-cluster 3.8 Active <none>
2025-03-06T08:46:57Z
test38 3.8 Failed Pod is unschedulable or is not starting
2025-03-06T08:46:36Z
```

## Удаление экземпляра

Удаление экземпляра не только удаляет созданные ресурсы CR, но также удаляет StatefulSet, ConfigMap, Secret, PersistentVolumeClaim и другие ресурсы, созданные Operator на основе ресурса CR.

### CLI

Если экземпляр не настроен на сохранение данных, или если необходимо сохранить постоянные данные экземпляра, выполните следующую команду для удаления экземпляра:

```
delete kafka instance directly
kubectl delete -n <namespace> rdskafka <name>
```

Если вы хотите удалить экземпляр вместе с его persistent volume claim, выполните следующие команды:

```
delete kafka instance directly
kubectl delete -n <namespace> rdskafka <name>
patch delete pvc
kubectl delete -n <namespace> pvc -l trimzi.io/cluster=<name>,strimzi.io/kind=Kafka
```

### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **Kafka**.
3. Выберите пространство имён.
4. Из списка экземпляров выберите экземпляр, который хотите удалить и перейдите на страницу сведений об экземпляре.
5. Нажмите **Actions > Delete**. Для экземпляров с включенной сохранностью данных можно отметить **Delete Persistent Volume Claim**, чтобы освободить место, или оставить его неотмеченным, чтобы сохранить данные.
6. Введите имя экземпляра, который нужно удалить.
7. Нажмите **Delete**.

## RabbitMQ

MWS Streaming Service for RabbitMQ — это open-source брокер сообщений, реализующий протокол Advanced Message Queuing Protocol (AMQP), работающий на Kubernetes. Он



служит коммуникационным мостом между приложениями, позволяя различным системам асинхронно отправлять и получать сообщения. Обеспечивает надежную основу для построения разьединённых, масштабируемых и отказоустойчивых распределённых систем.

## Создание экземпляра

### CLI

**Примечание:** Описание полей приведено в документации RabbitMQ API.

```
$ cat << EOF | kubectl -n default create -f -
apiVersion: rabbitmq.com/v1beta1
kind: RabbitmqCluster
metadata:
 name: rrrr
spec:
 affinity:
 podAntiAffinity:
 requiredDuringSchedulingIgnoredDuringExecution:
 - labelSelector:
 matchLabels:
 app.kubernetes.io/name: cluster-in
 topologyKey: kubernetes.io/hostname
 persistence:
 storage: 1Gi
 storageClassName: sc-topolvm
 rabbitmq:
 additionalConfig: |-
 channel_max=1000
 cluster_keepalive_interval=10000
 collect_statistics=none
 collect_statistics_interval=5000
 default_vhost=/
 delegate_count=16
 disk_free_limit.absolute=50MB
 handshake_timeout=10000
 heartbeat=60
 max_message_size=134217728
 mirroring_sync_batch_size=4096
 mnesia_table_loading_retry_limit=10
 mnesia_table_loading_retry_timeout=30000
 num_acceptors.tcp=10
 queue_index_embed_msgs_below=4096
```

```

raft.wal_max_size_bytes=64000000
vm_memory_calculation_strategy=allocated
vm_memory_high_watermark.relative=0.4
vm_memory_high_watermark_paging_ratio=0.5
replicas: 3
resources:
 limits:
 cpu: 1
 memory: 2Gi
 requests:
 cpu: 1
 memory: 2Gi
service:
 type: ClusterIP
upgradeOption:
 autoUpgrade: false
version: "3.12"
EOF

```

После создания экземпляра вы можете просмотреть экземпляр с помощью следующей команды:

```

$ kubectl -n default get rabbitmqclusters
NAME AGE
rrrr 74s

```

#### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **RabbitMQ**.
3. Выберите пространство имён.
7. Нажмите Create RabbitMQ Instance.
8. Настройте параметры:

| Элемент конфигурации     | Описание                                                                                                                                               |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Конфигурация параметров  | Шаблон параметров: можно выбрать системный шаблон или пользовательский                                                                                 |
| Конфигурация ресурсов    | Класс хранения: Если в выпадающем списке не найдено доступного класса хранения, пожалуйста, свяжитесь с администратором платформы, чтобы добавить один |
| Конфигурация подключения | Порт хоста: Этот вариант конфигурации появляется при выборе доступа из внешнего кластера.<br>Примечание: при обновлении экземпляра порты не            |

| Элемент конфигурации      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | взаимозаменяемы. Если необходимо поменять порты местами, сначала обновите на другой свободный порт, затем укажите нужный                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Конфигурация планирования | <ul style="list-style-type: none"> <li>Метки узлов: Фильтрация доступных узлов в текущем кластере через метки. Поды будут планироваться на доступные узлы.<br/>Примечание: После создания экземпляра эту настройку нельзя изменить.</li> <li>Pod Tolerations: Если на узле установлены метки (taints), только экземпляры с заданными Pod Tolerations могут быть запланированы на этом узле.<br/>Правила соответствия: <ul style="list-style-type: none"> <li>Equal: ключ, значение и эффект Pod Tolerations должны полностью совпадать с меткой узла.</li> <li>Exists: ключ и эффект Pod Tolerations должны совпадать с ключом и эффектом метки узла, значение игнорируется</li> </ul> </li> </ul> |

Для получения дополнительной информации обратитесь к официальной документации Kubernetes.

9. Нажмите **Create** и дождитесь изменения статуса на Running.

### Удаление экземпляра

Удаление экземпляра не только удаляет созданные ресурсы CR, но также удаляет StatefulSet, ConfigMap, Secret, PersistentVolumeClaim и другие ресурсы, созданные RabbitMQ на основе ресурса CR.

### CLI

Если экземпляр не настроен на сохранение данных, или если необходимо сохранить постоянные данные экземпляра, выполните следующую команду для удаления экземпляра:

```
удалить экземпляр rabbitmq напрямую
```

```
$ kubectl -n default delete rabbitmqclusters <Имя экземпляра>
```

По умолчанию удаление экземпляра не приведет к удалению созданного PVC. Используйте следующую команду для его удаления:

```
$ kubectl -n default delete pvc -l
app.kubernetes.io/component=rabbitmq,app.kubernetes.io/part-
of=rabbitmq,app.kubernetes.io/name=<Имя экземпляра>
```

### Веб-консоль

1. Перейдите в **Data Services**.

2. В левой панели выберите **RabbitMQ**.
3. Выберите пространство имён.
4. Из списка экземпляров выберите экземпляр, который хотите удалить и перейдите на страницу сведений об экземпляре.
5. Нажмите **Actions > Delete**. Для экземпляров с включенной сохранностью данных можно отметить **Delete Persistent Volume Claim**, чтобы освободить место, или оставить его неотмеченным, чтобы сохранить данные.
6. Введите имя экземпляра, который нужно удалить.
7. Нажмите **Delete**.

## PostgreSQL

MWS support for PostgreSQL — это облачное решение для управления PostgreSQL, построенное на базе Zalando Postgres Operator. Оно обеспечивает полное управление жизненным циклом кластеров PostgreSQL через Custom Resource Definitions (CRD), поддерживая основные операционные возможности, такие как автоматический failover, поэтапные обновления и восстановление из резервных копий, что позволяет предприятиям легко запускать производственные базы данных PostgreSQL на Kubernetes.

### Выбор архитектуры

Оператор Postgres поддерживает следующие типы архитектуры:

- Single Node: подходит для сред разработки и тестирования.
- Primary-Replica Replication: рекомендуется для производственных сред с:
  - Автоматическим переключением при сбое,
  - Высокой доступностью,
  - Избыточностью данных.

### Создание экземпляра

Примечание:

- Для производственных сред используйте архитектуру primary-replica.
- Перед развертыванием убедитесь, что ваш класс хранилища поддерживает динамическое выделение.
- Настройте соответствующие лимиты ресурсов.
- Реализуйте регулярное резервное копирование критически важных данных.

Внимание:

- Неправильные лимиты ресурсов могут привести к проблемам с производительностью или удалению подов.
- Отсутствие резервных копий может привести к потере данных при сбоях.

### CLI Single Node

# Пример минимальной конфигурации кластера HA

```

cat << EOF | kubectl create -f -
apiVersion: acid.zalan.do/v1
kind: postgresql # Определение пользовательского ресурса для кластеров PostgreSQL
metadata:
 name: pg-single
 namespace: c1-midautons
spec:
 ipFamilyPrefer: ""
 teamId: ACID
 enableExporter: true
 enablePgpool2: false
 spiloPrivileged: false
 spiloRunAsGroup: 103
 spiloRunAsUser: 101
 spiloAllowPrivilegeEscalation: false
 enableReadinessProbe: true
 postgresql:
 parameters:
 log_directory: /var/log/pg_log
 version: "14"
 numberOfInstances: 1
 resources:
 requests:
 cpu: "1"
 memory: 2Gi
 limits:
 cpu: "1"
 memory: 2Gi
 volume:
 size: 5Gi
 storageClass: c1-topolvmc
EOF

```

После создания вы можете проверить статус экземпляра с помощью следующей команды:

```

kubectl -n $NAMESPACE get postgresql
NAME AGE
pg-ha 21h

```

### **CLI Кластер с высокой доступностью**

```

cat << EOF | kubectl -n $NAMESPACE create -f -
apiVersion: acid.zalan.do/v1

```

```
kind: postgresql
metadata:
 name: pg-ha
 namespace: default
spec:
 ipFamilyPrefer: ""
 teamId: ACID
 enableExporter: true
 enablePgpool2: false
 spiloPrivileged: false
 spiloRunAsGroup: 103
 spiloRunAsUser: 101
 spiloAllowPrivilegeEscalation: false
 enableReadinessProbe: true
 postgresql:
 parameters:
 log_directory: /var/log/pg_log
 version: "14"
 numberOfInstances: 3
 resources:
 requests:
 cpu: "1"
 memory: 2Gi
 limits:
 cpu: "1"
 memory: 2Gi
 volume:
 size: 50Gi
 storageClass: default
 patroni:
 ttl: 30
 loop_wait: 10
 retry_timeout: 10
EOF
```

После создания вы можете проверить статус экземпляра с помощью следующей команды:

```
kubectl -n $NAMESPACE get postgresql
NAME AGE
pg-ha 21h
```

### Веб-консоль

1. Перейдите в **Data Services**.

2. В левой панели выберите **PostgreSQL**.
3. Выберите пространство имён.
4. Нажмите **Create PostgreSQL Instance**.
5. Настройте параметры:

| Элемент конфигурации             | Описание                                  |
|----------------------------------|-------------------------------------------|
| Имя экземпляра                   | Уникальный идентификатор кластера         |
| Версия PostgreSQL                | Выберите поддерживаемую версию PostgreSQL |
| Количество реплик                | Установите количество узлов кластера      |
| Спецификации (лимиты контейнера) | Установите лимиты ресурсов CPU и памяти   |
| Класс хранилища                  | Выберите класс хранилища                  |
| Квота хранилища                  | Установите класс хранилища и размер       |

6. Нажмите **Create** и дождитесь изменения статуса на Running.

### Удаление экземпляра

Удаление экземпляра PostgreSQL приведет к удалению:

- Пользовательского ресурса (CR),
- StatefulSet и Pods,
- ConfigMaps и Secrets,
- PersistentVolumeClaims (PVC),
- Ресурсов мониторинга и журнала.

### CLI

Чтобы удалить экземпляр PostgreSQL (включая все связанные PVC), выполните:

```
kubectl -n $NAMESPACE delete postgresql <instance-name>
```

Пример вывода:

```
postgresql.acid.zalan.do "example-postgres" удален
```

После удаления вы можете подтвердить статус экземпляра с помощью:

```
kubectl -n $NAMESPACE get postgresql
```

### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **PostgreSQL**.
3. Выберите пространство имён.
4. Из списка экземпляров выберите экземпляр, который хотите удалить и перейдите на страницу сведений об экземпляре.
5. Нажмите **Actions > Delete**.
6. Подтвердите удаление экземпляра и всех его данных.

7. Введите имя экземпляра, который нужно удалить.
8. Нажмите **Delete**.

## Управление шаблонами параметров

Функциональность шаблонов параметров предназначена для унифицированного управления и поддержки наборов параметров для различных компонентов. Включает в себя просмотр, какие параметры входят в шаблоны параметров, а также их значения по умолчанию и описание. Следующие компоненты поддерживают шаблоны параметров:

- MySQL (включая PXC и MGR),
- Redis,
- Kafka.

Платформа поставляется с преднастроенными шаблонами параметров для различных сценариев и спецификаций компонентов, которые отражают лучшие отраслевые практики для использования в этих сценариях, что позволяет пользователям применять их напрямую. Кроме того, пользователи могут создавать собственные шаблоны параметров на основе своих конкретных бизнес-требований, что облегчает внедрение внутренних стандартов. При создании экземпляров можно выбрать любой шаблон параметров для использования.

### Просмотр списка шаблонов параметров

1. Перейдите в **Data Services**.
2. В левой панели выберите **Parameter Templates**.

На этой странице доступен просмотр шаблонов параметров для конкретных проектов в зависимости от типа компонента и типа шаблона. В настоящее время поддерживаемые типы компонентов: MySQL, Redis и Kafka.

Доступные типы шаблонов — преднастроенные шаблоны платформы и пользовательские шаблоны. Преднастроенные шаблоны платформы универсально применимы ко всему кластеру, тогда как пользователи могут создавать собственные шаблоны специально для текущего проекта.

**Примечание:** Шаблоны параметров для MySQL могут применяться как к архитектуре PXC, так и к архитектуре MGR.

### Просмотр деталей шаблона параметров

Чтобы перейти к странице конкретного шаблона параметров, нажмите на его имя.

На странице отображается подробная информация о шаблоне, включая название шаблона, тип шаблона, тип компонента, описание шаблона, список параметров и значения параметров по умолчанию.

### Создание шаблона параметров

1. Перейдите в **Data Services**.
2. В левой панели выберите **Parameter Templates**.
3. Нажмите **Create Parameter Template**.



4. Заполните параметры: название шаблона, тип шаблона, тип компонента и описание шаблона.
5. Задайте параметры и их значения по умолчанию вручную или через импорт. Поддерживается пакетный импорт параметров и импорт параметров из документации.

Формат для пакетного импорта параметров:

```
Parameter1 ParameterValue1
Parameter2 ParameterValue2
...
```

**Примечание:** При создании экземпляра можно указать только один шаблон параметров, поэтому шаблон должен содержать максимально полный список параметров.

6. Нажмите **Create**.

### Экспорт шаблона параметров

1. Перейдите в **Data Services**.
2. В левой панели выберите **Parameter Templates**.
3. Экпортируйте шаблон параметров одним из способов:
  - В меню нужного шаблона выберите **Export**;
  - Нажмите на имя нужного шаблона и в меню **Action** выберите **Export**.

Формат экспортируемого файла — текстовый файл, соответствующий формату импорта.

## Управление резервным копированием

### Обзор

Backup Center – центр управления резервным копированием, один из ключевых модулей в платформе сервисов данных, служит центром управления операциями резервного копирования и восстановления данных между инстансами. Он предоставляет пользователям комплексное и удобное решение для управления процессом резервного копирования и восстановления данных.

Пользователи могут использовать Backup Center для фильтрации инстансов по типу компонента, пространству имён или ключевым словам, что позволяет быстро и точно отображать необходимые записи резервных копий. Одновременно пользователи могут легко выполнять операции резервного копирования и восстановления выбранных инстансов, а также удобно просматривать историю резервных копий и управлять файлами резервных копий. Центр управления резервным копированием также поддерживает централизованное управление внешним объектным хранилищем, используемым для резервных копий, включая операции регистрации, обновления и удаления методов доступа и ключей для внешнего хранилища.

## Внешнее хранилище S3

### Обзор

Функциональность предоставляет сервисы регистрации внешнего хранилища S3 для текущего выбранного проекта. С помощью простого и интуитивно понятного интерфейса вы легко выполните шаги для успешной регистрации внешнего хранилища S3 для проекта. Вы можете вручную инициировать сканирование для проверки доступности внешнего хранилища S3. Если возникнут какие-либо проблемы, например, сбой подключения, будут отправлены мгновенные уведомления, позволяющие вам в реальном времени отслеживать состояние хранилища. Когда вы или другие участники проекта захотите выполнить резервное копирование данных, будь то создание автоматической задачи резервного копирования или ручной запуск мгновенного бэкапа, достаточно выбрать зарегистрированное внешнее хранилище S3 в настройках резервного копирования, и система оперативно и безопасно сохранит резервные данные в этом хранилище.

### Добавление внешнего хранилища S3

Перед добавлением внешнего хранилища S3 соберите следующую информацию о хранилище:

- URL конечной точки (Endpoint)
- ключ доступа Access Key,
- секретный ключ доступа Secret Access Key,
- имя бакета.

Внешнее хранилище не обязательно должно быть развернуто в текущем кластере или пространстве имён, главное — наличие сетевого подключения. Пользователи могут заранее проверить доступность внешнего хранилища через веб-запросы.

Чтобы добавить внешнее хранилище S3:

1. Перейдите в **Data Services**.
2. В левой панели выберите **Backups > External S3**.
3. Нажмите Add External S3.
4. Заполните параметры, используя заранее собранную информацию об S3.

### Просмотр списка внешних хранилищ S3

Чтобы просмотреть список внешних хранилищ S3:

1. Перейдите в **Data Services**.
2. В левой панели выберите **Backups>External S3**.

При добавлении нового внешнего хранилища S3 автоматически запускается проверка состояния подключения. Чтобы запустить проверку подключения внешнего хранилища вручную, нажмите кнопку **Check** в строке нужного хранилища. Результаты проверки отображаются в строке хранилища.

## Резервные копии

### *Просмотр списка резервных копий*

1. Перейдите в **Data Services**.
2. В левой панели выберите **Backups > Backups**.

На странице списка резервных копий доступен просмотр статуса резервных копий для указанного проекта и пространства имён, включая включено ли автоматическое создание резервных копий, количество успешных и неудачных резервных копий, а также результат последнего резервного копирования

### *Просмотр истории резервных копий*

1. Перейдите в **Data Services**.
2. В левой панели выберите **Backups > Backups**.
3. В меню нужной резервной копии выберите **History**.

### *Восстановление из резервной копии*

1. Перейдите в **Data Services**.
2. В левой панели выберите **Backups > Backups**.
3. В меню нужной резервной копии выберите **History**.
4. В меню нужной записи выберите **Recovery**.
5. Задайте параметры восстановления. Разные компоненты могут поддерживать различные методы восстановления. По умолчанию все компоненты поддерживают восстановление на новый экземпляр, а MGR дополнительно поддерживает восстановление на существующий экземпляр:
  - Если пользователь выбирает восстановление на новый экземпляр, система перенаправит на страницу создания нового экземпляра. После ввода соответствующей информации для нового экземпляра (имя, характеристики, параметры и т. д.) система автоматически применит данные резервной копии для восстановления после создания экземпляра. Как только статус нового экземпляра изменится на ready, указанные данные резервной копии будут успешно восстановлены.
  - Если пользователь выбирает восстановление на существующий экземпляр, система предложит выбрать экземпляр и выполнить операцию восстановления.

### *Удаление резервных копий*

**Внимание:** Операция удаления резервной копии необратима.

1. Перейдите в **Data Services**.
2. В левой панели выберите **Backups > Backups**.
3. В меню нужной резервной копии выберите **History**.
4. Удалите записи одним из способов:
  - Выберите нужные записи и нажмите кнопку **Batch Delete**.

- В меню нужной записи выберите **Delete**.

## Управление инспекциями

### Обзор

Inspection Center — это один из ключевых модулей в платформе сервиса данных, который в первую очередь сосредоточен на проведении комплексных и тщательных операций инспекции middleware, поддерживаемого платформой. Inspection Center предлагает разнообразные методы запуска, позволяя вручную инициировать инспекции для удовлетворения потребности пользователя в проактивных проверках в определённые моменты времени, а также поддерживает автоматический запуск инспекций на основе предустановленных правил для немедленных или периодических проверок, обеспечивая непрерывный и регулярный мониторинг состояния middleware. Кроме того, Inspection Center обладает мощными возможностями представления и анализа результатов, генерируя отчёты по инспекциям и предоставляя последние обзоры инспекций, чтобы пользователи могли быстро получить общее представление, а также предоставляя профессиональную диагностику и ценные рекомендации по оптимизации при обнаружении аномалий в процессе инспекции. Для облегчения дальнейшего использования и обмена данными Inspection Center предусмотрительно поддерживает экспорт отчётов по инспекциям.

### Создание задачи инспекции

Задача инспекции представляет собой планирование и управление инспекциями для конкретных экземпляров компонентов. При создании задачи инспекции вы можете указать экземпляры компонентов, которые необходимо проверить, а также цикл и стратегию инспекции. Также можно задать максимальное количество записей инспекции для хранения; любые результаты инспекций, превышающие это количество, будут автоматически перезаписаны.

После успешного создания задачи инспекции бэкенд-сервис выполнит соответствующие операции инспекции в соответствии с настройками задачи.

### CLI

```
kubectrl -n <namespace> apply -f - <<EOF
{
 "apiVersion": "middleware.alauda.io/v1",
 "kind": "Inspection",
 "metadata": {
 "name": "insp-task",
 "namespace": "tongrds-1"
 },
 "spec": {
 "component": "mysqlcluster",
 "instances": [
```

```

 {
 "name": "mgr1"
 }
],
 "jobsHistoryLimit": 3,
 "trigger": "Manual"
}
}
EOF

```

Где:

- <name> — имя ресурса задачи инспекции.
- <namespace> — пространство имён, к которому принадлежит задача инспекции, а также целевое пространство имён для инспекции. Например, если вы хотите проверить экземпляр компонента mgr1 в пространстве имён tongrds-1, необходимо указать <namespace> как tongrds-1.
- <spec.component> — используется для указания целевого типа компонента для задачи инспекции, например, mysqlcluster.
- <spec.instances.name> — используется для указания имени целевого экземпляра для задачи инспекции; можно указать несколько экземпляров в одном пространстве имён или использовать подстановочный знак \* для соответствия всем экземплярам.
- <spec.jobsHistoryLimit> — задаёт максимальное количество записей проверок, которые будет хранить текущая задача инспекции. Записи, превышающие это значение, будут автоматически перезаписаны.
- <spec.trigger> — задаёт способ запуска текущей задачи инспекции, поддерживаются значения:
  - Manual — ручной запуск,
  - Schedule — запуск по расписанию. Если установлено значение Schedule, необходимо также задать следующий параметр:
    - <spec.schedule> — задаёт время запуска текущей задачи инспекции, например, "0 0 1 \* \*" означает запуск в 1 час ночи каждый день. Поле использует выражения Crontab, которые позволяют точно задавать время срабатывания по таким параметрам, как минуты, часы, дни, месяцы, недели и годы. Поскольку Kubernetes работает по UTC, необходимо сначала перевести местное время в UTC, а затем преобразовать UTC в выражения Crontab.

#### Веб-консоль

1. Перейдите в **Data Services**.
2. В левой панели выберите **Inspections > Tasks**.
3. В левом верхнем углу выберите проекты, которые нужно проверить.

4. Нажмите **Create Tasks** и заполните необходимые поля в появившейся форме: имя задачи, способ запуска, инстансы для проверки, максимальное количество записей проверок, которые будет хранить текущая задача инспекции.
5. Нажмите **Create**.

## Мониторинг и оповещение

### Обзор

Во время эксплуатации и обслуживания инстанса оповещение может использоваться для уведомления об исключениях или состояниях, требующих особого внимания. Когда инстанс сталкивается с аномалией или достигает состояния предупреждения, оповещения могут автоматически срабатывать, помогая своевременно обнаруживать и локализовать проблемы.

Для повышения эффективности эксплуатации и обслуживания система позволяет установить правила оповещений на основе мониторинговых показателей, используемых при устранении распространённых неисправностей инстансов, классифицируя и объединяя их во встроенные стратегии оповещений.

Платформа поддерживает правила оповещений на основе predetermined мониторинговых показателей и пользовательских показателей. При возникновении аномалий ресурсов или достижении состояния предупреждения оповещения могут автоматически срабатывать. В сочетании с функцией уведомлений платформы информация об оповещениях может активно передаваться операционному персоналу, напоминая им своевременно реагировать на оповещаемые ресурсы, обеспечивая бесперебойную работу бизнеса на платформе.

Для удобства настройки оповещений для большого количества ресурсов на платформе поддерживается возможность настраивать стандартизированные конфигурации оповещений для похожих инстансов с помощью шаблонов оповещений, что позволяет быстро создавать стратегии оповещений для ресурсов на основе существующих шаблонов.

### Типы оповещений

- **Metric Alert:** Общие мониторинговые показатели, извлечённые платформой, которые удовлетворяют большинству потребностей клиентов. Вы можете настроить оповещения, выбрав мониторинговые показатели и установив условия срабатывания. Когда мониторинговые данные соответствуют условиям срабатывания правила оповещения, оповещение будет вызвано.
- **Custom Alert:** Вы можете добавить корпоративные правила метрик на основе собственных сценариев использования, чтобы лучше удовлетворять расширенным потребностям предприятия в оповещениях на основе metric alerts.

### Статусы

Статусы оповещений:

- **Alert:** Данные, полученные при запросе конфигурации правил оповещений, соответствуют условиям срабатывания и вызвали оповещение.

- **Pending:** Мониторинговые данные, полученные при запросе конфигурации правил оповещений, больше или равны пороговому значению оповещения в условиях срабатывания, но длительность ещё не достигла условий срабатывания, что представляет собой критическое состояние перед вызовом оповещения.

Например, условие срабатывания правила оповещения — использование CPU превышает 80 % и длится 3 минуты. Когда система впервые обнаружит, что CPU превышает 80 %, правило будет помечено как Pending. Оценка будет продолжаться:

- если использование CPU останется выше 80 % в течение 3 минут, статус правила изменится на Alert;
  - если при последующей оценке использование CPU упадёт ниже 80%, статус правила вернётся в Normal.
- **Normal:** Данные, полученные при запросе конфигурации правил оповещений, не достигают порогового значения оповещения.

Статусы тишины:

- **Silent Pending:** Статус перед началом периода тишины после его установки. В этом статусе при срабатывании правил стратегии оповещения уведомления отправляются как обычно.
- **Silent:** Статус с начала до окончания периода тишины. В этом статусе при срабатывании правил стратегии оповещения уведомления не отправляются.

### Оповещение в реальном времени

Платформа отображает количество ресурсов, находящихся в состоянии оповещения, и подробную информацию об оповещениях через оповещения в реальном времени, позволяя операционному персоналу и администраторам в режиме реального времени отслеживать общую ситуацию с оповещениями бизнеса на платформе, своевременно выявлять и устранять неисправности для обеспечения нормальной работы платформы.

Чтобы открыть страницу оповещений в реальном времени:

1. Перейдите в **Data Services**.
2. В левой панели выберите **Alerts > Realtime Alerts**.

### История оповещений

Для помощи операционному персоналу и администраторам в анализе последних состояний мониторинговых оповещений и отслеживании истории проблем платформа поддерживает просмотр истории записей об оповещениях за определённый период времени. Доступная для просмотра информация включает время возникновения, правила оповещений, неисправные ресурсы, методы уведомления, статус и получателей уведомлений и др.

Чтобы просмотреть историю оповещений:

1. Перейдите в **Data Services**.
2. В левой панели выберите **Alerts > History**.

**Примечание:** Время хранения истории оповещений совпадает со временем хранения событий и может быть обновлено в разделе **Administrator** путём изменения параметров конфигурации компонента хранения логов.

## Обновление экземпляра сервисов данных

Функциональность управления обновлением экземпляров позволяет пользователям самостоятельно выбирать временное окно для обновления экземпляров. Разные экземпляры могут обновляться в разные временные окна для обеспечения непрерывности бизнес-процессов.

### Просмотр списка операторов

1. Перейдите в **Data Services**.
2. В левой панели выберите **Operators**.

В списке отображается ключевая информация, такая как режим развёртывания, область управления, статус и версия.

После переключения текущего проекта селектор кластера в правом верхнем углу также сбросится, и пользователю потребуется повторно выбрать кластер, чтобы список операторов отображался корректно. Такая конструкция обусловлена моделью мультиарендности, при которой проект может содержать несколько разных кластеров.

### Просмотр списка экземпляров

1. Перейдите в **Data Services**.
2. В левой панели выберите **Operators**.
3. Нажмите на имя конкретного оператора.

На странице списка экземпляров пользователи могут просмотреть имена, статусы и версии всех экземпляров, управляемых выбранным оператором, в текущем проекте и кластере. На странице будут отображаться только экземпляры в рамках текущего проекта и кластера в соответствии с права доступа для просмотра проекта.

### Обновление экземпляра

1. Перейдите в **Data Services**.
2. В левой панели выберите **Operators**.
3. Нажмите на имя конкретного оператора.
4. Выберите нужные экземпляры и нажмите **Batch Upgrade**. Статус выбранных экземпляров изменится на **Progressing**. После завершения обновления экземпляра статус вернется к **Running**.

**Примечание:** Согласно особенностям K8S порядок обновления нескольких экземпляров не фиксирован, но в конечном итоге все экземпляры будут обновлены.



## Service Mesh

Service Mesh — это PaaS-платформа на базе Istio, предоставляющая облачные решения для приложений и микросервисов.

Платформа предлагает возможности развертывания и обновления инфраструктуры в один клик, визуальное управление сервисами, эффективное управление производительностью приложений, а также высокую доступность и производительность при управлении сервисным шлюзом. Она обеспечивает комплексное, стабильное, надежное и открытое решение для микросервисов для предприятий, проходящих цифровую трансформацию, помогая повысить эффективность управления сервисами и снизить затраты на поддержку микросервисного фреймворка, при этом минимизируя зависимость разработчиков от фреймворка, позволяя предприятиям сосредоточиться на развитии бизнеса и постоянно повышать свою конкурентоспособность.

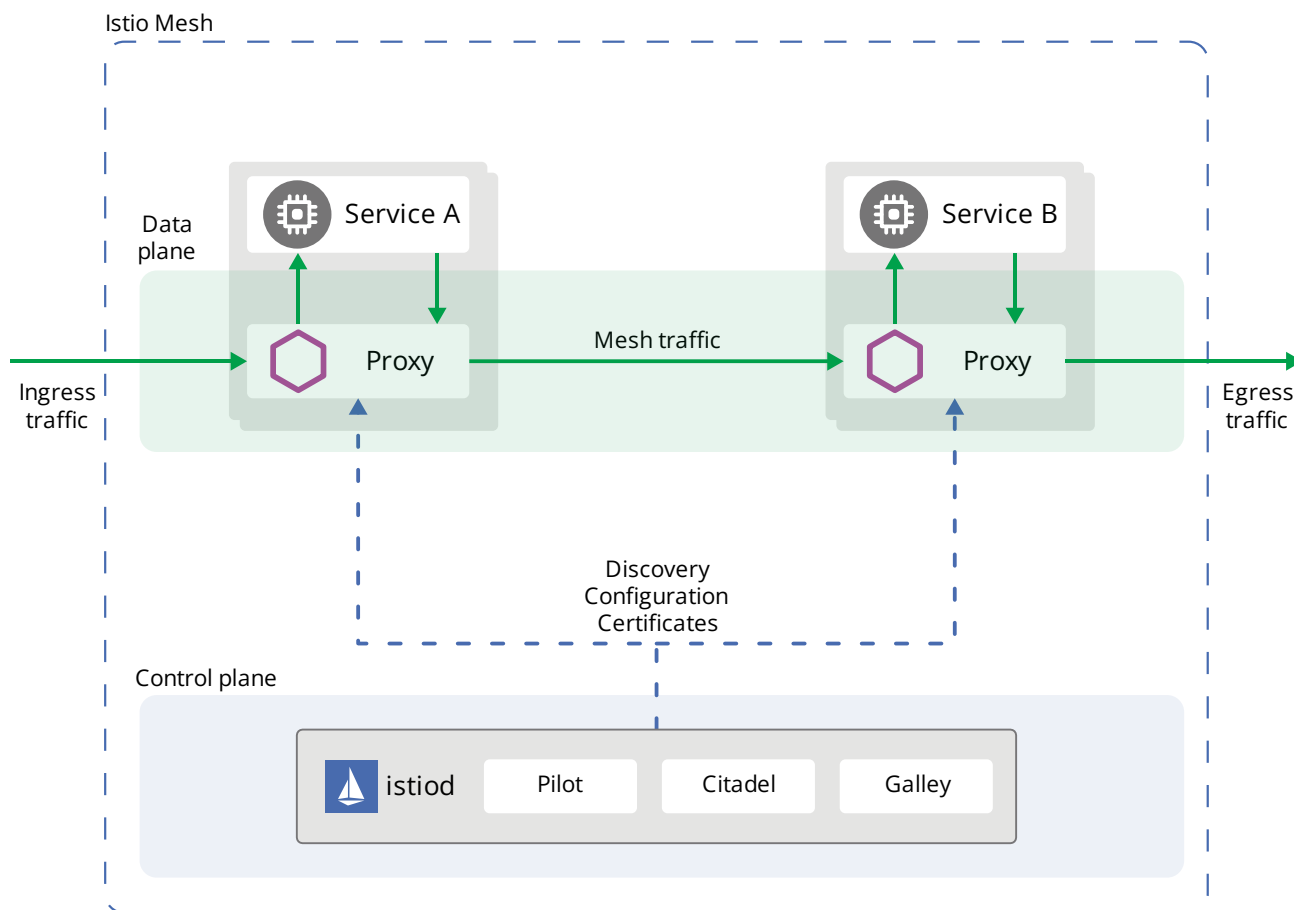
### Istio

Istio — это open-source Service Mesh, который прозрачно интегрируется в существующие распределённые приложения для обеспечения управления трафиком, применения политик, сбора телеметрии и обеспечения безопасности. Istio предлагает комплексные возможности управления сервисами для микросервисной архитектуры, поддерживает несколько языков и в основном использует Kubernetes в качестве реестра сервисов.

Istio предоставляет мощные функции для защиты, подключения и мониторинга сервисов единым и эффективным способом. С помощью Istio можно добиться балансировки нагрузки, межсервисной аутентификации и мониторинга с минимальными или без изменений в коде сервисов. Ключевые возможности включают:

- Безопасное межсервисное взаимодействие внутри кластера, включая TLS-шифрование, аутентификацию и авторизацию.
- Автоматическую балансировку нагрузки для HTTP, gRPC, WebSocket и TCP трафика.
- Тонкий контроль поведения трафика с помощью богатых правил маршрутизации, повторных попыток, переключения при сбоях и инъекции ошибок.
- Подключаемый слой политик и API конфигурации для контроля доступа, ограничения скорости и управления квотами.
- Автоматический сбор метрик, логов и трассировок со всего трафика внутри кластера, включая входящий и исходящий трафик.
- Масштабируемость для удовлетворения различных потребностей развертывания. Управляющий слой на Kubernetes, и вы можете добавлять приложения, развернутые в кластере, расширять mesh на другие кластеры и даже подключать виртуальные машины или другие конечные точки вне Kubernetes.

Архитектура Istio показана ниже.



## Control Plane

Управляющий слой Istio отвечает за управление и конфигурирование прокси. Он взаимодействует с прокси плоскости данных (Envoy), отправляя конфигурации и инструкции для управления трафиком сервисов, обеспечения безопасности и сбора и обработки телеметрии. Это позволяет администраторам и разработчикам централизованно управлять всеми сервисами с гибкой настройкой.

## Data Plane

Плоскость данных Istio состоит из набора интеллектуальных прокси (Envoy), развернутых и работающих как sidecar-контейнеры в pod сервисов. Эти прокси перехватывают, обрабатывают и перенаправляют весь трафик между сервисами.

Плоскость данных выполняет управление трафиком, восстановление после сбоев, применение политик безопасности и другие функции, обеспечивая надежность и безопасность межсервисного взаимодействия. Также она предоставляет богатые возможности мониторинга и трассировки, помогая разработчикам понимать и отлаживать поведение коммуникаций сервисов.

## Envoy

Envoy используется в качестве прокси для плоскости данных Istio и является единственным компонентом Istio, взаимодействующим с трафиком плоскости данных.

Envoy развёртывается и работает как sidecar в pod сервисов, обрабатывая и перенаправляя весь входящий и исходящий трафик сервисов в Service Mesh, предоставляя сетевые функции и возможности управления, такие как контроль трафика, балансировка нагрузки, восстановление после сбоев и безопасность.

### Sidecar

Паттерн sidecar широко используется в микро сервисах и контейнеризованных приложениях для расширения функциональности и производительности приложения путём добавления вспомогательных контейнеров или прокси рядом с основным контейнером приложения.

Envoy, формирующий плоскость данных Istio, развёртывается как sidecar-контейнер в pod сервисов.

### Service Gateway

Сервисные шлюзы включают Egress Gateway и Ingress Gateway, работающие на границе Service Mesh для управления трафиком, входящим и выходящим из mesh. Они позволяют задавать, какому трафику разрешено покидать или входить в mesh.

## Полный жизненный цикл управления Service Mesh

Управление сервисной сеткой становится особенно сложным в мульти кластерных средах. Полный жизненный цикл управления сервисной сеткой включает не только развертывание, обновление и удаление сервисной сетки, но и мониторинг её состояния в режиме реального времени, что позволяет операционному персоналу своевременно обнаруживать аномалии и устранять проблемы, обеспечивая тем самым стабильную и непрерывную поддержку бизнес-операций.

| Функция                                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Сервисная сетка с несколькими primary-кластерами   | Функция управления сервисной сеткой с архитектурой нескольких primary-кластеров визуализирует процесс развертывания компонентов платформы управления микросервисами.                                                                                                                                                                                                                                                                                                                                               |
| Визуальное развертывание и управление Service Mesh | Администраторы могут легко развернуть все компоненты платформы управления микросервисами, включая Istio, для каждого Kubernetes-кластера, требующего возможностей управления сервисами, используя визуальные формы для настройки ключевых параметров. При необходимости могут быть интегрированы внешние компоненты для обеспечения контейнеризованного управления микросервисами.<br><br>Администраторы могут настраивать конфигурацию ресурсов компонентов сетки в зависимости от фактического масштаба бизнеса. |

|                         |                                                                                                                                                                                                                                                                                                        |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Мониторинг Service Mesh | Функция мониторинга сервисной сетки включает несколько встроенных панелей мониторинга, отображающих данные мониторинга управляющей и пользовательской плоскостей с разных точек зрения и с различной детализацией, что облегчает администраторам запросы и просмотр данных в режиме реального времени. |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Глобальная визуализация сервисов

| Функция            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Топология сервисов | <p>Граф топологии сервисов наглядно показывает взаимосвязи между сервисами внутри сервисной сетки и между сервисами внутри и вне сетки (например, middleware, сервисы виртуальных машин с внедрённым OpenTelemetry Java Agent). Граф топологии помогает понять, какие компоненты взаимодействуют, и быстро определить точное место ошибки.</p> <p>Возможность фокусировки с глобального уровня на локальный и просмотра подробной информации о конкретном сервисе, включая статус сервиса, условия запросов, информацию о трафике и др.</p> <p>При настройке сервисов с политиками безопасности, circuit breaker и оповещениями они также отображаются специальными маркерами на графе топологии.</p> <p>Определение состояния здоровья подов сервиса по цвету узла.</p> <p>Определение состояния трафика между сервисами по цвету соединения.</p>        |
| Трассировка        | <p>Функция трассировки платформы поддерживает поиск цепочки вызовов между сервисами в текущем namespace по имени сервиса или TraceID, мониторинг ключевых показателей, таких как статус вызова и длительность сервисов в цепочке вызовов. В сочетании с детальными логами это помогает разработчикам понять реакцию каждого узла или конкретного API в цепочке бизнес-запроса для быстрой локализации и устранения проблем.</p> <p>Обеспечивает глубокий анализ вызовов между сервисами для разработчиков и операционного персонала.</p> <p>Отображает диаграмму, абстрагированную от времени начала и длительности цепочки вызовов, для быстрого выявления узких мест с низкой производительностью.</p> <p>Поддерживает полное трассирование (сервисы в архитектуре Service Mesh, сервисы Java 8+).</p> <p>Быстрый поиск цепочек вызовов по TraceID.</p> |

| Функция                  | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | Запрос информации по ссылкам на основе сервиса или TracelD.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Сервисный шлюз           | <ul style="list-style-type: none"> <li>• Сервисный шлюз, включая egress и ingress шлюзы, работает на границе сервисной сетки, управляя трафиком, входящим и выходящим из сетки, позволяя задавать разрешённый трафик.</li> <li>• Конфигурация нескольких ingress или egress шлюзов для сервисной сетки, при этом каждый шлюз допускает доступ через несколько настраиваемых портов.</li> <li>• Ingress шлюз поддерживает маршрутизацию внешнего трафика к различным backend-ресурсам (API-группы, сервисы) через различные сценарии маршрутизации; egress шлюз позволяет сервисам внутри сетки обращаться к внешним сервисам без влияния на опыт разработки и мониторинг трафик к ним.</li> <li>• Поддержка различных сценариев использования, таких как сетевой изоляции между бизнес-областями, выделенные шлюзы для конкретных бизнесов и изоляция портов.</li> <li>• Легкое выведение из эксплуатации ненужных шлюзов без потерь.</li> <li>• Аутентификация и авторизация клиентских запросов на основе JWT.</li> <li>• Настройка правил маршрутизации шлюза на основе пути запроса, заголовков запроса и переписывания URI.</li> <li>• Поддержка мониторинга шлюзов на уровне подов с несколькими встроенными стратегиями оповещений для своевременного обнаружения и устранения проблем. Данные мониторинга включают CPU, память, QPS, количество соединений, метрики входящего и исходящего трафика</li> </ul> |
| Список нативных ресурсов | Позволяет пользователям, знакомым с Kubernetes, просматривать нативные ресурсы Istio (YAML-файлы), классифицированные по типу ресурса                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Управление выпуском сервисов

Интеграция с Flagger обеспечивает автоматизированный опыт выпуска сервисов, поддерживая несколько стратегий развертывания, таких как Canary, A/B Testing и Blue/Green.

| Функция                                | Описание                                                                                                                                                                                                                                                             |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Конфигурация выпуска</b>            | Определение конфигураций выпуска сервисов для предоставления шаблонных параметров выполнения для будущих изменений, что позволяет автоматизировать развертывание и откат canary-версий Pod.                                                                          |
| <b>Стратегия маршрутизации трафика</b> | Предлагает гибкие стратегии маршрутизации трафика. Сервисы могут автоматически маршрутизировать трафик на основе шаблонов конфигурации выпуска или настраивать правила маршрутизации трафика для конкретных задач выпуска.                                           |
| <b>Управление выпуском</b>             | Поддерживает как ручные, так и автоматические варианты выпуска. Пользователи могут делегировать решения по выпуску автоматизированным системам или выбирать ручной выпуск. Процесс выпуска допускает ручное вмешательство и возможность отката задач в любой момент. |
| <b>Наблюдаемость</b>                   | Обеспечивает сравнение метрик и трассировку между основной и canary-версиями, что позволяет проводить анализ качества и локализацию ошибок при canary- релизах.                                                                                                      |

## Управление безопасностью сервисов

В сценариях управления микросервисами вызовы сервисов переходят с локальных на вызовы через сетевой протокол, что влечёт за собой риски безопасности.

| Функция                       | Описание                                                                                                                                               |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Стратегия безопасности</b> | Обеспечивает шифрование трафика между сервисами. При настройке стратегий безопасности для сервисов трафик может быть зашифрован с использованием mTLS. |

## Визуализированная эксплуатация и сопровождение сервисов

При возникновении сбоя сервиса обычный процесс устранения неполадок включает:  
отслеживание аномального трафика -> определение места сбоя -> сужение области ->  
анализ конкретной причины

Исходя из этого процесса, платформа предоставляет полный путь устранения проблем с сервисом, помогая быстро находить подозрительные места на основе аномального трафика и оперативно решать проблемы для минимизации влияния на бизнес.

| Функция                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Мониторинг трафика и JVM</b>     | <p>Предоставляет многомерные панели мониторинга трафика для клиентских и серверных сервисов с детализацией до уровня API;</p> <p>Мониторинг JVM поддерживает сбор и отображение данных по различным показателям, таким как CPU/память, потоки и классы. Позволяет просматривать данные одного пода или сравнивать данные двух подов, помогая разработчикам быстро локализовать показатели и временные точки возникновения конкретных проблем.</p>                                                                                                                                                                                                                                                                                                                                                                |
| <b>Оповещения</b>                   | <p>Обеспечивает функции оповещений на основе распространённых показателей мониторинга, поддерживая сбор показателей, управление стратегиями, срабатывание оповещений и уведомления. В сочетании с мониторингом трафика и JVM обеспечивает полный интеллектуальный опыт эксплуатации мониторинг-оповещение-уведомление.</p> <ul style="list-style-type: none"> <li>• Предоставляет шаблоны конфигурации стратегий оповещений, поддерживая настройку одной стратегии оповещений или быстрое создание стратегий через шаблоны.</li> <li>• Позволяет просматривать информацию о ресурсах, находящихся под оповещением, через панель реального времени оповещений, фильтровать данные по уровню оповещения и просматривать детали оповещений для понимания масштабов проблемы и быстрого поиска источника.</li> </ul> |
| <b>Логи в реальном времени</b>      | <p>Предоставляет панель логов в реальном времени на уровне сервиса, позволяя быстро находить и устранять проблемы с помощью логов (включая access логи) при возникновении аномалий сервиса</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Быстрое устранение неполадок</b> | <p>При сбое сервиса можно использовать богатые визуальные данные (топология сервисов, трассировка) и данные мониторинга и эксплуатации (мониторинг, оповещения, логи) для отслеживания и локализации аномального трафика и оперативного решения проблем</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

## Восстановление сервисов после сбоев

В сценариях с несколькими регионами и дата-центрами, когда pod сервиса в одном дата-центре выходит из строя и изолируется с помощью circuit breaking, платформа автоматически распределяет трафик клиента на поды сервиса в других дата-центрах, обеспечивая стабильное предоставление сервиса.

| Функция                                                     | Описание                                                                                                                                                                                                                                    |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Маршрутизация по близости</b>                            | Все вызовы сервисов по умолчанию используют маршрутизацию по близости во всех дата-центрах. При длинных цепочках вызовов это обеспечивает приоритетную обработку трафика аварийного восстановления конечной точки сервиса того же кластера. |
| <b>Пользовательский приоритет аварийного восстановления</b> | Позволяет настраивать региональные нагрузки для задания приоритета обработки аварийного трафика между несколькими кластерами на основе расположения регионов кластеров.                                                                     |

## Модели развёртывания Service Mesh

Понимание моделей развёртывания сервисной сетки имеет решающее значение для ее эффективного использования. Основные модели развёртывания включают сервисную сетку с одним кластером и сервисную сетку с несколькими кластерами. В следующих разделах приводится подробное объяснение этих двух моделей развёртывания.

### Сервисная сетка с одним кластером

#### Обзор

Сервисная сетка с одним кластером развёртывается внутри одного кластера Kubernetes, управляет и контролирует все микросервисы внутри этого кластера. Эта модель подходит для сценариев, когда сервисы преимущественно сосредоточены в одном кластере, без необходимости в межкластерной или межрегиональной связи.

#### Сценарии использования

- Микросервисы развернуты только внутри одного кластера Kubernetes.
- Не требуется учитывать межкластерные сетевые подключения и совместное использование ресурсов.
- Упрощенное развёртывание и управление, что снижает сложность и операционные расходы.

#### Преимущества



- Относительно простое развертывание и управление, поскольку необходимо настроить только один кластер.
- Высокая производительность сети, так как все коммуникации происходят внутри одного кластера.
- Централизованное управление и мониторинг, что упрощает решение проблем.

#### ***Недостатки***

- Единственная точка отказа: если кластер столкнется с проблемами, все сервисы будут затронуты.
- Ограниченная масштабируемость: межкластерное управление сервисами и управление трафиком нецелесообразны.

#### **Сервисная сетка с несколькими кластерами**

##### ***Обзор***

Сервисная сетка с несколькими кластерами развертывается на нескольких кластерах Kubernetes, что позволяет осуществлять межкластерное управление микросервисами. Эта модель подходит для сценариев, требующих развертывания сервиса на нескольких кластерах или имеющих потребности в межрегиональном восстановлении после аварий.

##### ***Сценарии использования***

- Микросервисы распределены по нескольким кластерам Kubernetes.
- Необходимы высокие возможности доступности и восстановления после аварий в разных регионах.
- Желание достичь межкластерного управления трафиком и управления сервисами.

##### ***Преимущества***

- Высокая доступность: межкластерное развертывание избегает единой точки отказа и повышает надежность сервиса.
- Высокая масштабируемость: кластеры можно добавлять или удалять в зависимости от потребностей бизнеса, что позволяет гибко управлять ресурсами.
- Межрегиональное восстановление после аварий: кластеры, развернутые в разных регионах, обеспечивают географическое восстановление после аварий и избыточность данных.

##### ***Недостатки***

- Высокая сложность развертывания и управления, требующая учета межкластерных сетевых подключений и конфигураций.
- Производительность сети может пострадать, требуется оптимизация задержки и ограничения пропускной способности в межкластерном взаимодействии.
- Необходимы более сильные возможности мониторинга и операционного управления для обеспечения стабильности межкластерных сервисов.

## Компоненты Mesh

### Участвующие решения с открытым исходным кодом

- **Istio**: Решение с открытым исходным кодом для сервисной сетки, предоставляющее функции управления трафиком, безопасности и наблюдаемости.
- **OpenTelemetry**: Решение с открытым исходным кодом для наблюдаемости, поддерживающее инструментирование на уровне кода с широким применением.
- **Jaeger**: Решение с открытым исходным кодом для распределенной трассировки.
- **Flagger**: Инструмент постепенной доставки с открытым исходным кодом, поддерживающий различные стратегии развертывания, включая Canary, A/B Testing и Blue/Green.

### Описание компонентов

#### *Компоненты контрольной плоскости*

| Название компонента   | Описание                                                                                                                                                                                                                                                                                 |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Istio</b>          |                                                                                                                                                                                                                                                                                          |
| istiod                | Компонент управляющей плоскости Istio, название которого различает различные основные версии Istio.                                                                                                                                                                                      |
| istio-cni             | Узловой агент Istio CNI, используемый для настройки перенаправления трафика для Pod в сети. Этот компонент не развертывается по умолчанию, если не используются OpenShift.                                                                                                               |
| tier2-gateway         | Gateway Tier2 в сочетании с gateway Tier1 предоставляет усовершенствованные функции управления трафиком, такие как серые релизы и маршрутизацию сервисов для входящего трафика в сети.                                                                                                   |
| istio-eastwestgateway | Восточно-западный (east west) gateway в Istio для связи сервисной сетки между несколькими кластерами. Его основная функция - обеспечить возможность взаимодействия сервисов в различных кластерах. Этот компонент развертывается только в сервисных сетях multi-network и multi-cluster. |
| asm-core              | Компонент, предоставляющий глобальное ограничение скорости для Istio. Он разворачивается по умолчанию, но работает только в том случае, если сервисная сетка интегрирована с Redis.                                                                                                      |
| <b>OpenTelemetry</b>  |                                                                                                                                                                                                                                                                                          |

| Название компонента   | Описание                                                                                                                                                                                                                                                                                 |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| asm-otel-collector    | Компонент решения OpenTelemetry для получения, обработки и экспорта данных наблюдаемости.<br>Поддерживает открытые форматы данных, такие как Jaeger и Prometheus. Данные трассировки отправляются в Jaeger, в то время как данные Метрики отправляются в Prometheus или VictoriaMetrics. |
| asm-otel-collector-lb | Обеспечивает балансировку нагрузки для компонента asm-otel-collector. Собранные данные наблюдаемости, проведенные OTel Java Agent, передаются в asm-otel-collector через этот компонент.                                                                                                 |
| <b>Jaeger</b>         |                                                                                                                                                                                                                                                                                          |
| jaeger-prod-collector | Компонент коллектора Jaeger, собирающий данные распределенной трассировки от sidecars и asm-otel-collector, и взаимодействующий с Elasticsearch для хранения данных трассировки.                                                                                                         |
| jaeger-prod-query     | Компонент запроса данных Jaeger, запрашивающий данные трассировки из Elasticsearch.                                                                                                                                                                                                      |
| <b>Flagger</b>        |                                                                                                                                                                                                                                                                                          |
| flagger               | Обеспечивает функциональность серых релизов и интегрируется с Istio для реализации стратегий постепенной доставки для сервисов.                                                                                                                                                          |
| <b>Другие</b>         |                                                                                                                                                                                                                                                                                          |
| asm-controller        | Управляет и координирует различные ресурсы внутри сервисной сетки.                                                                                                                                                                                                                       |

#### **Компоненты плоскости данных**

| Название компонента    | Описание                                                                                                                                       |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Ingress Gateway</b> | Независимый прокси Envoy, работающий на краю сети. Позволяет настраивать публичные порты и протоколы для указания разрешенного трафика в сеть. |

| Название компонента             | Описание                                                                                                                                                                                                                                                                         |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Egress Gateway</b>           | Независимый прокси Envoy, работающий на краю сети. Позволяет мониторинг, ведение журналов и соблюдение политик безопасности для трафика, покидающего сеть.                                                                                                                       |
| <b>Sidecar</b>                  | Обычно разворачивается в том же Pod, что и контейнер приложения. Обрабатывает сетевые запросы от приложения, включая управление трафиком, соблюдение политики безопасности, балансировку нагрузки и обнаружение сервисов.                                                        |
| <b>OpenTelemetry Java Agent</b> | Автоматизированный инструмент для сбора данных распределенной трассировки и метрик внутри Java приложений. Инжекция OpenTelemetry Java Agent позволяет выполнить распределенную трассировку маршрутов запросов и сбор данных о производительности без изменения кода приложения. |

## Создание Service Mesh

1. В левой навигационной панели нажмите **Service Mesh > Mesh**.
2. Нажмите Создать сервисную сетку.
3. Выберите кластер и версию Istio для разворачивания сервисной сетки. В расширенной конфигурации убедитесь, что архитектура сетки является однокластерной, и заполните параметры интерфейса для Elasticsearch и системы мониторинга. Вы можете выбрать существующую систему платформы или внешнюю систему.
  - Если строго требуется высокая доступность, установите обязательное anty-afinity Pod.
  - Ресурсы компонентов могут использовать значения по умолчанию, но по мере роста масштаба сервисов в сетки компоненты потребуют увеличения. Настройте политики оповещения для компонентов сетки вовремя, чтобы быть в курсе необходимости масштабирования.

Для получения дополнительной информации смотрите: [Описание параметров сетки](#)

**Примечание:** Когда кластер является кластером OpenShift, сетка автоматически обнаружит и развернет компонент istio-cni по умолчанию.

**Следующие шаги**

- Включить Istio CNI, чтобы устранить необходимость в привилегированных контейнерах инициализации в каждом Pod.
- Включить глобальное ограничение скорости.
- Использовать инструмент Istioctl.
- Мониторинг компонентов сетки.

## Описание параметров Service Mesh

### Глобальная конфигурация

Глобальная конфигурация сетки будет применена ко всем кластерам, где развернута сетка.

| Параметр                         | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Интерфейс с Elasticsearch        | <p><b>Платформа:</b> Интерфейс с плагином журнала Elasticsearch на любом кластере платформы. Выберите кластер, где находится плагин журнала Elasticsearch.</p> <p><b>Внешний:</b> Интерфейс с внешним плагином журнала Elasticsearch. Пользователям необходимо настроить следующие параметры:</p> <p><b>Адрес доступа:</b> Адрес доступа к Elasticsearch, начинающийся с http:// или https://.</p> <p><b>Метод аутентификации:</b> Метод аутентификации для доступа к Elasticsearch.</p> <p><b>Basic Auth:</b> Введите имя пользователя и пароль для проверки пользователя.</p> <p><b>Без аутентификации:</b> Аутентификация не требуется для доступа.</p>                                                                                                                                                                                                                                         |
| Интерфейс с системой мониторинга | <p><b>Платформа:</b> Интерфейс с плагином Prometheus или плагином VictoriaMetrics в кластере.</p> <p><b>Внешний:</b> Интерфейс с Prometheus или VictoriaMetrics, предоставляемыми внешним плагином. Пользователям необходимо настроить следующие параметры:</p> <p><b>Адрес запроса данных:</b> Адрес запроса данных компонента мониторинга, начинающийся с http:// или https://.</p> <p><b>Метод аутентификации:</b> Метод аутентификации для доступа к системе мониторинга.</p> <p><b>Basic Auth:</b> Введите имя пользователя и пароль для проверки пользователя.</p> <p><b>Без аутентификации:</b> Аутентификация не требуется для доступа.</p> <p><b>Примечание:</b> Из-за невозможности плагина Prometheus агрегировать данные мониторинга из нескольких кластеров сервисная сетка в этом кластере не может добавлять больше кластеров для формирования многокластерной сервисной сетки.</p> |

### Конфигурация по размеру кластера

Конфигурация по размеру кластера применяется только к выбранному кластеру.

| Параметр                         | Описание                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Конфигурация Sidecar             | <b>Квота ресурсов:</b> Значение квоты ресурсов sidecar по умолчанию на уровне кластера. Она может быть изменена в зависимости от фактических условий при инъекции sidecar для конкретных сервисов, но не может превышать максимальный лимит квоты контейнера пространства имен (LimitRange), в котором находится sidecar. |
| Конфигурация Trace               | <b>Частота выборки:</b> Значение частоты выборки трассировки sidecar по умолчанию на уровне кластера.                                                                                                                                                                                                                     |
| Конфигурация Redis               | Должен быть настроен только при использовании функции <b>Глобальное ограничение скорости</b> в плоскости данных. Для конкретных методов конфигурации смотрите <u><a href="#">Включить глобальное ограничение скорости</a></u> .                                                                                           |
| Политика повторного запроса HTTP | <b>Количество повторов:</b> Максимальное количество повторов запроса HTTP по умолчанию на уровне кластера.<br><b>Примечание:</b> Количество повторов в политике маршрутизации сервиса <b>Таймаут и повторы</b> переопределит это значение по умолчанию.                                                                   |

### Конфигурация компонентов

**Примечание:** Компоненты сетки устанавливаются в определенных пространствах имен кластера в виде развертываний. После успешного создания сетки вы можете просмотреть статус работы компонентов на вкладке **Компоненты** или щелкнуть **Имя компонента**, чтобы перейти к пространству имен, где развернут компонент на контейнерной платформе, и просмотреть подробную информацию о работе развертывания компонента.

| Параметр               | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Anty-afinity Pod       | <p>Kubernetes будет планировать Pod компонентов на узлы, которые соответствуют настройкам <b>Anty-afinity Pod</b>.</p> <p><b>Обязательно:</b> Разрешено запускать только один Pod одного компонента на одном узле.</p> <p><b>Предпочтительно:</b> Разрешено запускать несколько Pods одного компонента на одном узле. Kubernetes попытается равномерно распределить Pods компонента по доступным узлам в соответствии с алгоритмом планирования, но не может гарантировать наличие Pods компонента на каждом узле.</p> |
| Количество экземпляров | Ожидаемое количество Pods компонентов, которое должно быть запущено, устанавливаемое в зависимости от фактического объема запросов бизнеса.                                                                                                                                                                                                                                                                                                                                                                            |
| Квота ресурсов         | Значение запроса ресурса (процессор, память) для каждого экземпляра контейнера компонента при его создании, которое также является лимитом доступных ресурсов для контейнера. Установите его разумно в зависимости от фактического объема потребления и количества экземпляров.                                                                                                                                                                                                                                        |
| Узел развертывания     | Узлы, на которых развернут компонент. При создании сетки Pods компонентов могут быть запланированы только на выбранных узлах.                                                                                                                                                                                                                                                                                                                                                                                          |
| ELB                    | Балансировщик нагрузки Huawei Cloud Elastic Load Balancer (ELB), используемый для обеспечения возможности балансировки нагрузки для шлюза Istio в кластере CCE. Когда выбранный кластер является кластером CCE, вам необходимо заполнить <b>ID ELB</b> и <b>тип ELB</b> , чтобы связать ELB, подготовленный заранее для кластера.                                                                                                                                                                                      |

| Параметр      | Описание                                                                                                                                                                                                                                                                            |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Общие ресурсы | Общая квота ресурсов (процессор, память), необходимая для создания всех экземпляров контейнеров компонента в рамках текущей конфигурации. Когда компонент активирован для автоматического масштабирования, он рассчитывается в зависимости от максимального количества экземпляров. |

## Установка агента узла Istio CNI

Агент узла Istio CNI используется для настройки перенаправления трафика для подов в mesh. Он запускается как DaemonSet на каждом узле с повышенными правами. Агент узла Istio CNI может использоваться в двух режимах плоскости данных Istio.

Для режима sidecar плоскости данных агент узла Istio CNI является необязательным. Он устраняет необходимость запуска привилегированных init контейнеров в каждом поде mesh, заменяя эту модель одним привилегированным подом агента узла на каждом узле Kubernetes.

Агент узла Istio CNI **обязателен** в режиме ambient плоскости данных.

Этот документ сосредоточен на использовании агента узла Istio CNI как опциональной части режима sidecar плоскости данных.

**Подсказка:** Если вы создаёте сервисный mesh в кластере **OpenShift**, Istio CNI включён по умолчанию, поэтому вам не нужно выполнять шаги из этого документа

### Пререквизиты

- Среды, где необходимо использовать Istio CNI:

Среды с высокими требованиями к безопасности. (Например, среды, запрещающие использование разрешений NET\_ADMIN и NET\_RAW в бизнес-нагрузках)

- Среды, где можно использовать Istio CNI:

Любой кластер Kubernetes с включённой поддержкой CNI

- Среды, где нельзя использовать Istio CNI:

Кластеры с отключённой поддержкой Kubernetes CNI (только среды разработки, такие как KIND)

### Установка агента узла Istio CNI

Сначала установите сервисный mesh как обычно. После установки временно не добавляйте сервисы (то есть не внедряйте Sidecar). Затем выполните следующие шаги для установки и настройки Istio CNI.



### **Установка с компонентом ServiceMesh Istio CNI**

В большинстве сред базовый кластер Istio с включённым компонентом **istioCni** можно установить с помощью следующих команд:

```
apiVersion: asm.alauda.io/v1alpha1
kind: ServiceMesh
spec:
 componentConfig:
 - name: istioCni
 group: istio
 cni:
 namespace: kube-system
```

- **istioCni** необходим для установки агента узла Istio CNI.
- Обычно устанавливается в namespace kube-system .

### **Дополнительная конфигурация**

```
apiVersion: asm.alauda.io/v1alpha1
kind: ServiceMesh
spec:
 componentConfig:
 - name: istioCni
 group: istio
 cni:
 namespace: kube-system
 # Расширенная конфигурация
 values:
 logLevel: info
 cniBinDir: ""
 cniConfDir: /etc/cni/net.d
 cniConfFileName: ""
 cniNetnsDir:
 excludeNamespaces:
 - istio-system
 - kube-system
 chained: true
 provider: default
```

**Внимание:** При использовании OpenShift или Multus CNI:

- cni.values.chained должно быть false,
  - cni.values.provider должно быть multus.
1. Уровень логирования. (По умолчанию: info)
  2. Каталог бинарных файлов CNI. (По умолчанию: /opt/cni/bin)

3. Каталог конфигурации CNI. (По умолчанию: `/etc/cni/net.d`)
4. Файл конфигурации для вставки конфигурации плагина `istioCni`, по умолчанию это будет первый найденный файл в `cniConfDir`.
5. Этот каталог должен существовать на узле, если нет — обратитесь к документации вашего контейнерного рантайма для определения правильного пути. (По умолчанию: `/var/run/netns`, в `minikube/docker/` других может быть `/var/run/docker/netns`)
6. Развёртывать конфигурационные файлы как цепочку плагинов (значение `true`) или как отдельные файлы в каталоге конфигурации (значение `false`).
7. Пользовательская конфигурация на основе провайдера CNI, возможные значения: `default` и `multus`. (По умолчанию: `default`)

### Проверка

**Подсказка:** Убедитесь, что все поды агента узла CNI находятся в состоянии `Running` перед переходом к следующему шагу.

### Управление внедрением `init` контейнера

# При необходимости измените часть `-nkube-system`, если агент `Istio CNI` установлен в другом `namespace`:

```
kubectl -nkube-system get po -lk8s-app=istio-cni-node
```

По умолчанию `Istio` использует `webhook` для внедрения контейнера `istio-init` для прозрачного перехвата трафика. Чтобы использовать `Istio CNI` вместо контейнера `istio-init`, необходимо изменить ресурс `ServiceMesh` следующим образом:

```
apiVersion: asm.alauda.io/v1alpha1
kind: ServiceMesh
spec:
 istioConfig:
 cni:
 enabled: true
 provider: default
```

1. Включение `Istio CNI` вместо контейнера `istio-init`. (По умолчанию: `false`)
2. Провайдер CNI. Должен быть `multus`, если используется **Multus CNI**. (По умолчанию: `default`)

### Использование и проверка

После установки и настройки используйте обычный процесс внедрения сервиса без дополнительных изменений.

После установки и правильной настройки компонента `Istio CNI` в подах произойдут следующие изменения:

Контейнер `istio-init` больше не будет присутствовать (прозрачный перехват трафика будет осуществляться агентом узла `Istio CNI`). Вместо него `Istio` внедрит контейнер с именем `istio-`

validation, который проверяет успешность установки правил iptables для прозрачного перехвата трафика.

#### Состояние гонки и способы устранения

DaemonSet Istio CNI устанавливает сетевой плагин CNI на каждом узле. Однако существует временной промежуток между моментом, когда под DaemonSet планируется на узел, и моментом, когда плагин CNI установлен и готов к использованию. В этот промежуток может запуститься под приложения, и kubelet не будет знать о плагине Istio CNI. В результате под приложения запускается без перенаправления трафика Istio и обходит sidecar Istio.

Для устранения гонки между подом приложения и DaemonSet Istio CNI в рамках внедрения sidecar добавляется init контейнер istio-validation, который проверяет корректность настройки перенаправления трафика и блокирует запуск пода при ошибках. DaemonSet CNI обнаружит и обработает любой под, застрявший в таком состоянии; способ обработки пода зависит от конфигурации, описанной ниже. Эта мера включена по умолчанию и может быть отключена установкой `cni.values.repair.enabled` в `false`

```
apiVersion: asm.alauda.io/v1alpha1
kind: ServiceMesh
spec:
 componentConfig:
 - name: istioCni
 group: istio
 cni:
 values:
 repair:
 enabled: true
 repairPods: true
 deletePods: false
 labelPods: false
```

1. Включение устранения гонки. (По умолчанию: true)
2. При включении поды динамически перенастраиваются с правильной конфигурацией. При перезапуске контейнера под продолжит нормальную работу. (По умолчанию: true)
3. При включении поды **удаляются**, при повторном планировании они будут иметь правильную конфигурацию. (По умолчанию: false)
4. При включении поды только маркируются меткой `cni.istio.io/uninitialized=true`. Пользователю необходимо вручную решить проблему. (По умолчанию: false)

**Примечание:** Приоритет конфигурации: `repairPods` > `deletePods` > `labelPods`. Таким образом, будет использовано первое поле со значением `true` с более высоким приоритетом.

Например, чтобы активировать deletePods, недостаточно установить true для deletePods, необходимо также установить false для repairPods

## Установка Istioctl

Istioctl — это утилита командной строки, предоставляемая Istio, которая может использоваться для диагностики проблем внутри Service Mesh или для создания, просмотра, изменения и удаления конфигурационных ресурсов в mesh.

**Примечание:** При развертывании Service Mesh на платформе автоматически установится соответствующий Istioctl в зависимости от выбранной версии Istio. Вы можете использовать инструмент Istioctl для управления вашим Service Mesh через UI, предоставляемый платформой. Если вы хотите использовать Istioctl непосредственно на своей машине вне UI платформы, вам необходимо скачать соответствующий дистрибутив Istio из сообщества Istio и распаковать его на своей машине для использования istioctl.

Для часто используемых команд и инструкций, пожалуйста, обратитесь к Istioctl.

### Шаги

1. В левой навигационной панели нажмите **Service Mesh > Mesh**.
2. Нажмите на Имя Service Mesh.
3. В области **Mesh Deployment** нажмите на иконку меню справа от кластера, в котором вы хотите открыть инструмент Istioctl , затем выберите **Istioctl Tool**.

## Мониторинг компонентов Mesh

Платформа предоставляет графики мониторинга для компонентов mesh, что делает удобным понимание общего рабочего состояния сетки.

### Шаги

1. В левом боковом меню нажмите **Service Mesh > Mesh**.
2. Нажмите на **Имя Service Mesh**, для которого вы хотите просмотреть мониторинг.
3. На вкладке **Monitoring** вы можете просмотреть соответствующие данные мониторинга сетки.

### Объяснение показателей мониторинга

| Показатель мониторинга                  | Описание                                                                                                                                                            |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Уровень использования памяти компонента | Уровень использования памяти компонентов Service Mesh, развернутых в кластере.<br>Уровень использования = Использование памяти компонента / Квота памяти компонента |
| Уровень использования CPU компонента    | Уровень использования CPU компонентов Service Mesh, развернутых в кластере.<br>Уровень использования = Использование CPU компонента / Квота CPU компонента          |

| Показатель мониторинга      | Описание                                                                                                                                                                                                                              |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Уровень успешных соединений | Уровень успешности соединений между компонентами Service Mesh и компонентами мониторинга, Redis и Elasticsearch в кластере. <b>Уровень успешных соединений = Количество успешных соединений / Общее количество попыток соединения</b> |

## Управление пространством имен в Service Mesh

Для обеспечения точного контроля над службами в сетки вы можете ограничить область, которую сетка может контролировать, управляя пространствами имен. Службы могут быть добавлены и управляться в пространстве имен только в том случае, если пространство имен управляется сервисной сеткой.

Примечание:

Вы можете исключить пространство имен из управления, удалив его, но только пространства имен без служб могут быть удалены.

### Шаги

1. В левой навигационной панели нажмите **Service Mesh > Meshes**.
2. Нажмите на **Имя сервисной сетки**, для которого вы хотите установить политику.
3. Прокрутите вниз до раздела **Namespaces**, выберите кластер и нажмите **Управлять пространством имен**.
4. Выберите пространство имен в подкластере и нажмите **Подтвердить**.

## Добавление сервиса

Этот документ проведет вас через создание службы ServiceMesh или службы OpenTelemetry.

### Пререквизиты

- Текущее пространство имён было добавлено в сетку обслуживания.
- Тип рабочей нагрузки — это Deployment и с ним связана ассоциация один к одному со службой.

### Добавить службу ServiceMesh

1. В левой навигационной панели нажмите: **Список служб**.
2. Нажмите: **Добавить службу**.
3. Ознакомьтесь с следующими инструкциями для конфигурации соответствующих параметров.

| Параметр                           | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Развертывание                      | Развертывание в текущем пространстве имен на платформе контейнеров, которое является вычислительным компонентом, выполняющим службу.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Внутренняя маршрутизация           | <p>Информация о внутреннем маршруте, который соответствует выбранному развертыванию и поддерживает модификации протокола маршрута и имени контейнера.</p> <p>Внутренняя маршрутизация должна соответствовать следующим условиям:</p> <ul style="list-style-type: none"> <li>• С выбранным развертыванием связана только одна внутренняя маршрутизация, и этот маршрут эксклюзивно связан с этим развертыванием.</li> <li>• Тип внутреннего маршрута либо NodePort, либо ClusterIP.</li> <li>• Служба поддерживает только протоколы HTTP, HTTP2, gRPC, TCP. Не добавляйте неподдерживаемые протоколы внутренней маршрутизации, так как это может привести к ошибкам вызова службы.</li> </ul> |
| Конфигурация side контейнера       | <p>В режимах управления Service Mesh или Composite side контейнер будет автоматически вводиться в добавленную службу. Это означает, что когда Pod службы запускается, контейнер side программы также запускается внутри Pod для управления трафиком, входящим и выходящим из службы, а также управления и контроля службы.</p> <p>Вы можете контролировать распределение ресурсов для side контейнера через конфигурацию ресурсов.</p> <p>Вы можете настроить уровень журналирования side контейнера.</p> <p>Вы можете настроить пропуск трафика по мере необходимости</p>                                                                                                                   |
| Обнаружение служб между кластерами | <p>Этот параметр виден только тогда, когда текущая сетка обслуживания является <b>много-кластерной сеткой обслуживания</b> (по крайней мере 2 кластера).</p> <p>Управляет открытием служб для всех кластеров, управляемых сеткой, что позволяет любой службе или шлюзу в других кластерах под сеткой напрямую обращаться к текущей службе.</p> <p>Переключатель <b>Обнаружение служб между кластерами</b> является глобальным (по всей сетки) параметром.</p> <p>Изменение состояния переключателя для любой службы с тем же именем под сеткой повлияет на все службы с тем же именем (службы с тем же именем, что и внутренний маршрут, связанный</p>                                       |

| Параметр | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <p>с выбранным развертыванием) во всех кластерах.</p> <p><b>Примечание:</b></p> <p><b>Службы с тем же именем</b> обозначают службы в разных кластерах, управляемых сеткой обслуживания, в пространстве имен с тем же именем;</p> <p>Успешно добавленная служба и внутренний маршрут, связанный с выбранным развертыванием, <b>носят одно и то же имя.</b></p> <p><b>Когда переключатель включен</b>, после успешного добавления службы:</p> <ul style="list-style-type: none"> <li>• Если в любом другом управляемом кластере нет службы с именем внутреннего маршрута, соответствующего тому, что связано с выбранным развертыванием, внутренний маршрут с тем же именем будет синхронно создан в соответствующем кластере для обнаружения службы. <b>При создании внутреннего маршрута с тем же именем в других кластерах:</b> <ul style="list-style-type: none"> <li>• Если в кластере нет пространства имен с тем же именем, что и текущее пространство имен, это приведет к сбою создания внутреннего маршрута, что, в свою очередь, не позволит обеспечить возможности кросс-кластерного обнаружения служб.</li> <li>• Если существует существующий внутренний маршрут с тем же именем, что и тот, который связан с выбранным развертыванием в пространстве имен (с тем же именем, что и текущее пространство имен), будет использован существующий внутренний маршрут. <b>Примечание:</b> Если существующий внутренний маршрут имеет отличные параметры конфигурации от текущего внутреннего маршрута, связанного с выбранным развертыванием, обнаружение служб между кластерами может быть невозможно.</li> </ul> </li> <li>• Переключатель <b>Обнаружение служб между кластерами</b> для всех служб с тем же именем в сетки во всех кластерах будет включен.</li> </ul> |

#### 4. Нажмите **Добавить**.

После успешного добавления службы:

- Служба с тем же именем, что и **Служба**, будет создана в списке служб. (В методе управления **Service Mesh**)
- Платформа автоматически обновит параметры шаблона Pod развертывания

службы, добавляя или обновляя метку **app: <имя службы>**.

Рекомендуется не изменять эти метки напрямую в файле YAML.

- Перезапустите развертывание службы.

В процессе перезапуска, пока хотя бы один Pod развертывания находится в состоянии **Работает**, служба находится в **Онлайне**; в противном случае служба находится в **Оффлайне**.

### Добавить службу OpenTelemetry

1. В левой навигационной панели нажмите **Список служб**.
2. Нажмите **Добавить службу**. В всплывающем окне метода управления выберите OpenTelemetry.
3. Ознакомьтесь с следующими инструкциями для конфигурации соответствующих параметров.

| Параметр      | Описание                                                                                                                                                                                                                                                                                   |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Развертывание | Развертывание, установленные в текущем пространстве имен платформы контейнеров, которое является вычислительным компонентом, выполняющим службу.                                                                                                                                           |
| Имя службы    | В методе управления <b>OpenTelemetry</b> необходимо установить имя службы.<br>Примечание:<br>Когда <b>metadata.labels</b> выбранного развертывания содержит метку <b>asm.cpaas.io/msname: xxx</b> , <b>Имя службы</b> по умолчанию принимает значение этой метки и не может быть изменено. |

4. Нажмите **Добавить**.

После успешного добавления службы:

В режимах управления OpenTelemetry или Composite служба (Java 8+) по умолчанию будет внедрена с Java-агентом OpenTelemetry, позволяя платформе ненавязчиво собирать телеметрические данные службы и данные мониторинга JVM:

- Компонент **opentelemetry-operator** будет перехватывать создание Pod служб и добавлять конфигурацию Java-агента.
- Перезапустите развертывание службы.



В процессе перезапуска, пока хотя бы один Pod развертывания находится в состоянии **Работает**, служба находится в **Онлайне**; в противном случае служба находится в **Оффлайне**.

### Настройка шифрования (mTLS)

Взаимное шифрование транспортного уровня (mTLS) — это протокол, который обеспечивает взаимную аутентификацию между двумя сторонами. Это основной режим аутентификации в некоторых протоколах (IKE, SSH) и необязательный в других (TLS).

Вы можете использовать mTLS без изменения кода вашего приложения или сервиса. TLS полностью обрабатывается инфраструктурой сервиса Mesh и проходит между двумя прокси-боками.

По умолчанию mTLS включен в Service Mesh и установлен в режим с разрешением. В этом режиме прокси в Service Mesh принимают как незашифрованный трафик, так и соединения, зашифрованные с помощью mTLS. Если сервисы в mesh настроены на использование строгого mTLS для связи с сервисами за пределами mesh, связь может быть нарушена, поскольку строгий mTLS требует взаимной аутентификации между клиентом и сервером. Используйте режим с разрешением при миграции рабочих нагрузок в Service Mesh. Затем вы можете включить строгий mTLS между mesh, пространствами имен или приложениями.

### Создание политик безопасности

1. В левом навигационном меню нажмите **Список сервисов**.
2. Нажмите на **Имя сервиса**, для которого вы хотите создать правила безопасности.
3. Вкладке **Политики** нажмите **Создать политику** > **Безопасность** на правой стороне кластера.
4. В разделе Правила выберите Режим с разрешением или Строгий режим.

**Режим с разрешением:** позволяет сервису принимать как незашифрованный трафик, так и трафик с взаимным TLS.

**Строгий режим:** позволяет сервису принимать только трафик с взаимным TLS. Сервис будет обрабатывать только трафик, зашифрованный прокси, а незашифрованный трафик будет отклонен.

Примечание:

Включение **Режима с разрешением** позволяет сервису одновременно принимать как незашифрованный, так и трафик с взаимным TLS, что значительно упрощает внедрение взаимного TLS.

Например:

Когда операционный персонал хочет перенести сервис в Istio с включенным взаимным TLS, может возникнуть множество проблем с общением клиентов и серверов, не использующих Istio. Как правило, операционный персонал не может установить прокси Istio на всех клиентах одновременно и не имеет разрешений на это. Даже если прокси Istio установлены на сервере, они не могут включить взаимный TLS, не нарушая существующие соединения.

**Режим с разрешением** гибко решает эту ситуацию. Прокси Istio, установленный в сервисе, сразу принимает трафик с взаимным TLS, не нарушая существующий незашифрованный

трафик. Таким образом, операционный персонал может постепенно установить и настроить прокси Istio клиента для отправки трафика с взаимным TLS. Как только клиенты настроены, сервер можно настроить на **Строгий режим** (только TLS).

5. Нажмите **Создать**. При создании политик безопасности система автоматически создаст клиентские правила и включит TLS по умолчанию.

## Развёртывание Ingress Gateway

### Введение

Развертывание Ingress Gateway обеспечивает безопасный внешний доступ к ресурсам сервисной сетки за счёт:

- Предоставления выделенных точек входа для север-южного (North-South) трафика
- Поддержки гибких методов предоставления способа подключения (LoadBalancer/NodePort)
- Возможности настройки слушателей с учётом протоколов
- Изоляции компонентов gateway от управляющей плоскости

Основная ценность: безопасное и масштабируемое управление входящим трафиком.

### Особенности

- Поддержка слушателей для нескольких протоколов (HTTP/HTTPS/TCP)
- Режимы развертывания: общий/выделенный gateway
- Интеграция с балансировщиками нагрузки облачных провайдеров (AWS ELB, MetalLB и др.)
- Резервный вариант NodePort для bare-metal окружений

### Преимущества

- **Изоляция безопасности:** развертывание в неймспейсах вне управляющей плоскости
- **Готовность к мульти-облаку:** поддержка основных облачных LB-решений
- **Гибкость портов:** настройка сервисных и node портов
- **Контроль доступа:** управление видимостью на уровне неймспейсов

### Развёртывание Gateway через LoadBalancer

#### Шаг 1. Инициация развертывания

- Перейдите в **Service Mesh > Gateway**
- Выберите целевой кластер
- Нажмите Deploy Gateway

| Параметр     | Требование  | Описание              |
|--------------|-------------|-----------------------|
| Gateway Name | Обязательно | Соответствие DNS-1123 |

|                     |               |                                     |
|---------------------|---------------|-------------------------------------|
| <b>Namespace</b>    | Обязательно   | Неймспейс вне управляющей плоскости |
| <b>Service Type</b> | Автоустановка | LoadBalancer                        |

## Шаг 2. Настройка протоколов

ports:

- protocol: HTTPS

port: 443

- protocol: TCP

port: 3306

| Параметр          | Ограничения    |
|-------------------|----------------|
| <b>Protocol</b>   | HTTP/HTTPS/TCP |
| <b>Port Range</b> | 1-65535        |

## Развёртывание Gateway через NodePort

### Шаг 1. Конфигурация NodePort

1. Установите **Service Type** в NodePort
2. Определите node порты:

| Протокол | NodePort | Ограничение           |
|----------|----------|-----------------------|
| HTTP     | 30080    | Уникальный в кластере |
| HTTPS    | 30443    | Без конфликтов        |

### Шаг 2. Настройка внешнего прокси

Рекомендуемая архитектура:

Client → F5/HAProxy → NodeIP:NodePort → Ingress Gateway

#### Проверка

1. Откройте страницу с деталями gateway
2. Убедитесь, что статус: **Running**
3. Проверьте доступность конечной точки:  
curl -I http://<EXTERNAL-IP>:<PORT>/healthz

## Управление конфигурацией Sidecar

### Введение

Конфигурация Sidecar позволяет точно контролировать поведение прокси Envoy в сервисных сетях Istio. Эта функциональность:

- Определяет область применения политики трафика
- Оптимизирует использование ресурсов прокси
- Реализует контроль доступа на уровне namespace
- Поддерживает иерархические стратегии конфигурации

Основная ценность: балансировка производительности и безопасности сервисной сети через детальное управление трафиком

#### Особенности

- Конфигурация по умолчанию для всего кластера
- Пользовательские правила для отдельных namespace
- Поддержка шаблонов с подстановочными знаками
- Механизм приоритетного переопределения

#### Управление конфигурацией по умолчанию

##### Параметры конфигурации

| Параметр     | Формат            | Пример         |
|--------------|-------------------|----------------|
| Egress Hosts | namespace/dnsName | istio-system/* |

#### Процесс обновления

1. Перейдите: **Service Mesh > Meshes**
2. Выберите целевую mesh > вкладка **Sidecar Config**
3. Найдите кластер > нажмите иконку редактирования
4. Настройте egress hosts:
  - Выберите namespace из выпадающего списка
  - Выберите шаблон:
    - \*/\* для всего кластера
    - ./\* для текущего namespace
  - Пользовательский namespace
5. Нажмите **Confirm**

#### Ограничения:

Влияет на все namespace без пользовательской конфигурации.

#### Настройка пользовательской конфигурации

##### Шаблон конфигурации

apiVersion: networking.istio.io/v1beta1

```
kind: Sidecar
metadata:
 name: ns-custom
 namespace: target-ns
spec:
 egress:
 - hosts:
 - "dev/*"
 - "prod/db-service"
```

### **Шаги**

1. Откройте страницу деталей **mesh**.
2. Перейдите на вкладку **Sidecar Config**.
3. Нажмите Create Sidecar Configuration.
4. Выберите целевой namespace.
5. Настройте шаблоны egress hosts:  
Используйте подстановочный знак \* обозначает текущий namespace
6. Проверьте приоритет переопределения для широкого сопоставления

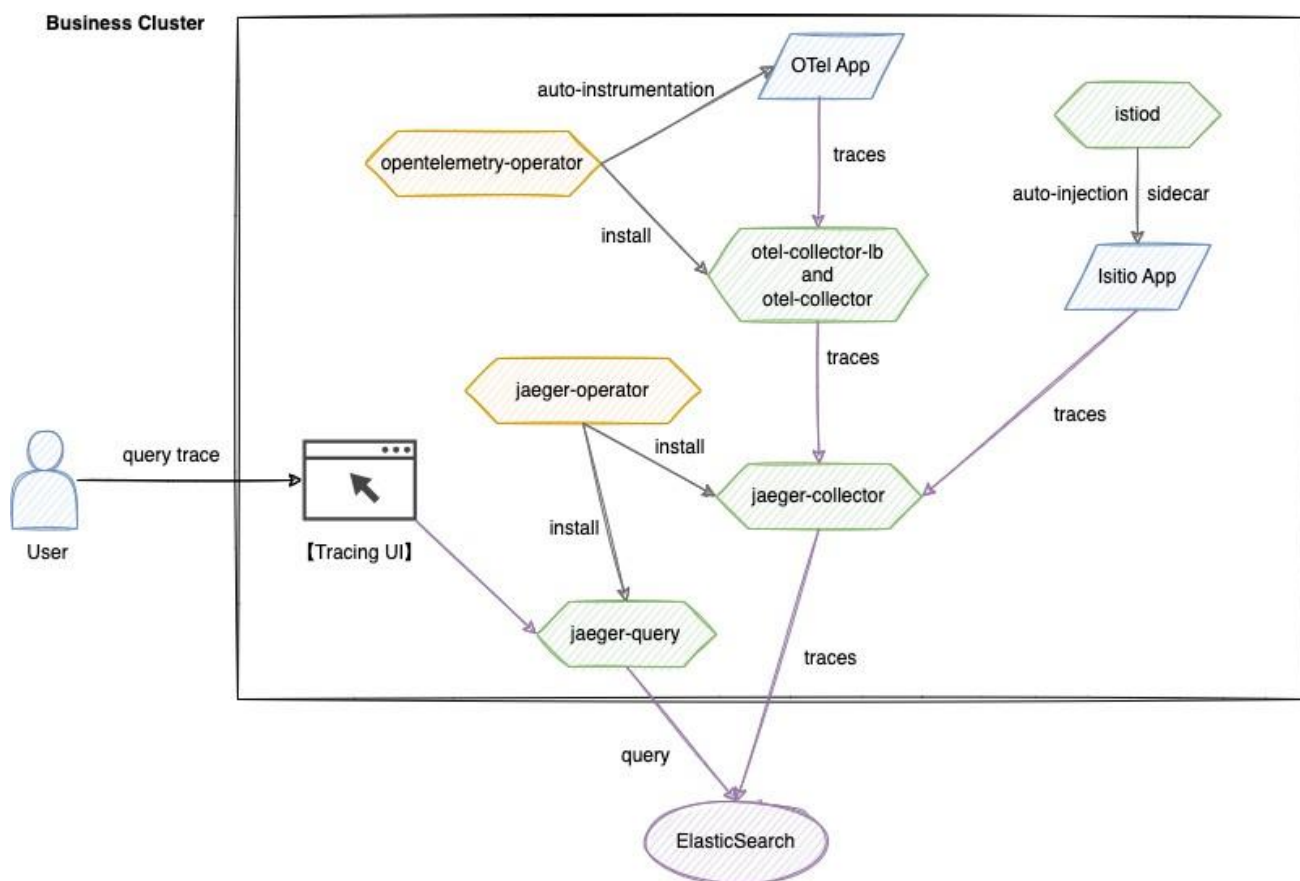
## **Распределенное трассирование**

Распределенное трассирование является ключевым модулем в системе наблюдаемости, используемым для достижения сквозного трассирования и анализа распределенных систем. Этот модуль предоставляет полное решение от сбора данных, хранения до визуального анализа, помогая разработчикам и операционному персоналу быстро находить аномалии вызовов сервисов, анализировать узкие места производительности и отслеживать поведение запросов на протяжении всего жизненного цикла.

Интегрируя с открытыми технологическими стеками и разработанными компонентами, этот модуль поддерживает возможности сквозного распределенного трассирования: приложения генерируют данные трассировки через методы инъекции или интеграции SDK, которые затем единообразно собираются и хранятся в Elasticsearch, что в конечном итоге реализуется через настроенный пользовательский интерфейс для многомерного визуального анализа. Пользователи могут проводить точные поиски, используя разные условия, такие как TraceID, имя сервиса, теги и т. д.

### **Архитектура**

Эта архитектура построена на технологиях Istio, OpenTelemetry и Jaeger, обеспечивая управление полным жизненным циклом распределенного трассирования. Система состоит из пяти основных модулей: сбора данных, передачи, хранения, запросов и визуализации.



### Система Istio

- **istio-proxy**

Прокси-сервис, который автоматически захватывает и генерирует данные трассировки для связи между сервисами без изменения кода.

- **istiod**

Управляет конфигурацией и жизненным циклом istio-proxy, включая конфигурацию трассировки для сервисной сетки.

### Система OpenTelemetry

- **opentelemetry-operator**

Оператор на уровне кластера, отвечающий за развертывание и управление компонентом otel-collector, обеспечивая автоматическую инъекцию OTEL.

- **otel-collector**

Принимает данные трассировки от приложений, фильтрует и группирует их, а затем перенаправляет на jaeger-collector.

- **Интерфейс визуализации трассировки**

Самостоятельно разработанный визуализирующий интерфейс, интегрирующийся с API jaeger-query, поддерживает многомерные условия запросов.

## ***Система Jaeger***

- **jaeger-operator**

Разворачивает и управляет компонентами jaeger-collector и jaeger-query.

- **jaeger-collector**

Принимает данные трассировки, перенаправленные и обработанные otel-collector, выполняет преобразование форматов и записывает их в Elasticsearch.

- **jaeger-query**

Предоставляет API для запросов трассировки, поддерживает много-условный поиск, включая TraceID и метки.

### ***Уровень хранения***

- **Elasticsearch**

Распределенный движок хранения, поддерживающий эффективную запись и извлечение большого объема данных Span.

### **Запрос трассировки**

#### ***Обзор***

Функциональность запроса распределенной трассировки предоставляет возможности полной трассировки для микросервисной архитектуры, собирая метаданные о межсервисных вызовах, помогая пользователям быстро находить проблемы с кросс-сервисными вызовами. Эта функциональность в первую очередь решает следующие задачи:

- **Трассировка ссылок запросов:** Восстановление полного пути запроса в сложных распределенных системах.
- **Анализ узких мест производительности:** Выявление аномальных узлов вызовов с точки зрения временных затрат в рамках ссылки.
- **Локализация коренной причины неисправности:** Быстрое определение места возникновения проблемы с помощью маркеров ошибок.

Сценарии использования включают:

- Быстрое выявление аномальных сервисов во время устранения неполадок в производственной среде.
- Выявление цепочек вызовов с высокой задержкой во время настройки производительности.
- Проверка взаимосвязей межсервисных вызовов после выхода новой версии.

Основные ценности:

- Повышение наблюдаемости распределенных систем.
- Сокращение среднего времени на восстановление (MTTR).
- Оптимизация производительности межсервисных вызовов.

## Основные возможности

- Многомерный запрос: Поддержка 9 комбинаций условий запроса, таких как TracelD, имя сервиса, метки и др.
- Визуальный анализ: Интуитивное отображение иерархии вызовов и распределения времени с помощью графиков водопадов временных рядов.
- Точный поиск: Поддержка фильтрации ошибок, Span и вторичных поисков с помощью меток.

## Создать запрос трассировки

### Шаг 1. Комбинирование условий запроса

**Совет:** Условия запроса можно комбинировать. Вы можете уточнить свой запрос, добавив несколько условий запроса.

| Условие запроса              | Описание                                                                                                                                                                        |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TracelD                      | Уникальный идентификатор для полной ссылки, который может использоваться для запроса указанной трассировки.                                                                     |
| Сервис                       | Сервис или шлюз входа, инициирующий/принимающий вызовы запросов (ввод обязательный). Формат: <имя сервиса/шлюза>. <имя пространства имен> , например: bookinfo-gw.bookinfo-ns . |
| Клиент                       | Клиентский сервис или шлюз входа, инициирующий вызов запроса (ввод обязательный). Формат: <имя сервиса/шлюза>.<имя пространства имен> .                                         |
| Сервер                       | Серверный сервис или шлюз входа, отвечающий на вызов запроса (ввод обязательный). Формат: <имя сервиса/шлюза>.<имя пространства имен> .                                         |
| API                          | API, добавленный для сервиса в разделе Сервисы > API.                                                                                                                           |
| Метки                        | Вы можете отфильтровать результаты запроса, введя метки (Tag). Поддерживаемые метки включают метки, найденные в деталях Span.                                                   |
| Продолжительность Span       | Spans, у которых продолжительность больше или равна входному значению (мс).                                                                                                     |
| Только ошибочные трассировки | Ошибочные Spans относятся к Spans, значение метки которых <b>error</b> равно true .                                                                                             |



| Условие запроса                  | Описание                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Тип Span                         | <p><b>Корневой Span:</b> Ищет корневые Spans, инициированные заданным <b>сервисом</b>. Этот режим поиска используется, когда заданный сервис является инициатором всего вызова запроса.</p> <p><b>Служебный входной Span:</b> Ищет первый Span, созданный, когда заданный <b>сервис</b> вызывается как сервер.</p>                                                                                   |
| Максимальное количество запросов | <p>Максимальное количество Spans, которые можно запросить, по умолчанию 200.</p> <p><b>Совет:</b> По соображениям производительности платформа может отображать максимум 1000 Spans за раз. Если количество Spans, соответствующих условиям запроса, превышает <b>максимальное количество запросов</b>, вы можете уточнить условия запроса или сузить временной диапазон для поэтапных запросов.</p> |

## Шаг 2. Выполнение запроса

- После выбора условий запроса и ввода соответствующих значений нажмите кнопку **Добавить в запрос**, и текущие условия отобразятся в области вывода **Запрос**, чтобы инициировать запрос.
- Вы также можете развернуть **Общие условия поиска**, чтобы быстро добавить недавно использованные условия поиска.

## Мультикластерное развёртывание

### Несколько управляющих плоскостей

Платформа разворачивает управляющую плоскость Istio для всех кластеров в сетке, создавая ресурсы ServiceMesh в кластерах.

Каждая управляющая плоскость может независимо управлять службами, шлюзами и другими ресурсами (ServiceEntry, EnvoyFilter, плагины WASM), развернутыми в кластере, и может обновляться отдельно.

### Независимые домены отказа

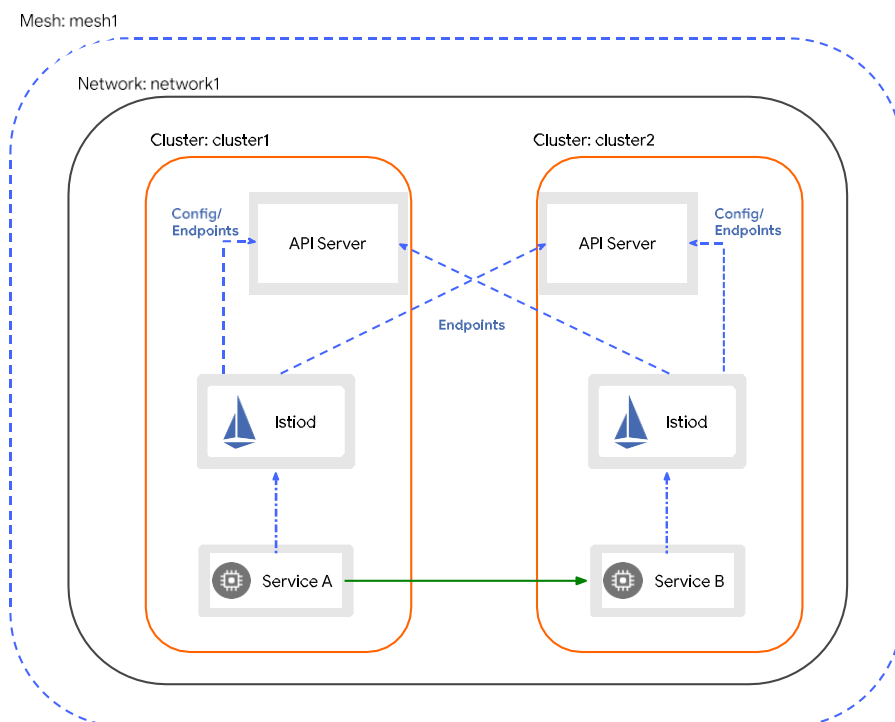
Кластеры в пределах сетки полностью независимы, что позволяет разворачивать их в разных облаках и регионах. Каждый кластер является независимым доменом отказа.

- Отказ в любом кластере в пределах сетки не влияет на другие кластеры.
- Службы могут быть развернуты в любом кластере в пределах сетки.

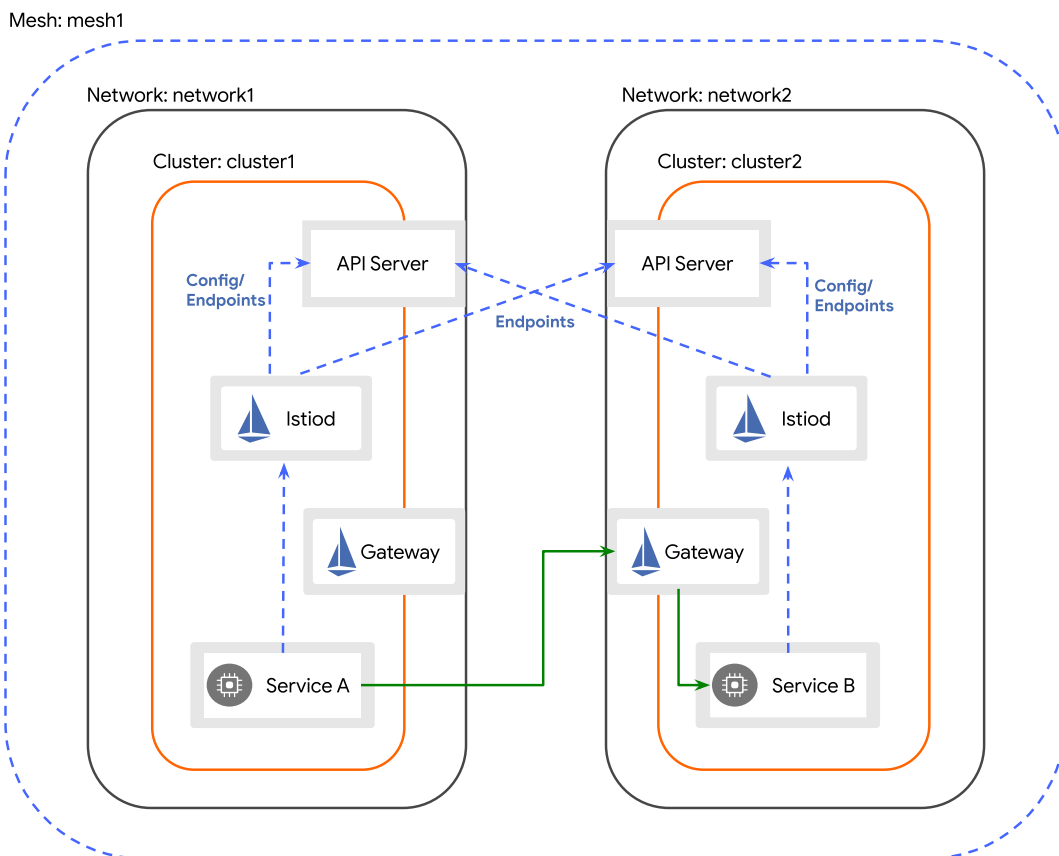
- Службы могут достичь восстановления после аварий в разных регионах и балансировки нагрузки.

#### Режим единой сетки/многофункциональной сетки

- **Единая сетка:** Когда несколько кластеров Kubernetes в сети взаимосвязаны, и поды в разных кластерах могут взаимодействовать друг с другом через IP-адреса подов, сетка работает в режиме единой сети. В этом режиме поды могут напрямую общаться без шлюзов Istio или других компонентов для передачи.



- **Многофункциональная сетка:** Когда кластеры в пределах сетки находятся не в одной и той же сетевой плоскости (VPC), поды в разных кластерах должны использовать шлюзы Istio для общения.



### Настройка и валидация отказоустойчивости между кластерами

В этом документе объясняется, как реализовать отказоустойчивость между кластерами для приложений на основе много кластерной сервисной сетки с использованием приложения Bookinfo в качестве примера.

#### Шаги:

1. В левом навигационном меню нажмите **Service Mesh > Mesh**.
2. Нажмите **имя сервисной сетки**, которую нужно настроить.
3. На вкладке **Mesh Policies** нажмите **Create Policy > Circuit Breaker** с правой стороны кластера c1.

#### Примечание:

- Если нет особых требований, используйте параметры по умолчанию.
  - Если служба не настроена с **Circuit Breaker Policy**, конфигурация **Failover** не будет применяться к ней, и входящий трафик будет случайным образом распределяться между Pods службы в различных кластерах сервисной сетки. Если служба настроена с **Circuit Breaker Policy**, входящий трафик будет направляться внутри кластера службы до тех пор, пока все Pods в этом кластере не будут отключены.
4. На вкладке **Mesh Policies** нажмите **Create Policy > Regional Load Balancing** с правой стороны кластера c1.
  5. Выберите **Failover** в качестве **Policy Type** и настройте связанные параметры, как

указано ниже.

| Параметр                        | Описание                                                                                                    |
|---------------------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Failure Region</b>           | Регион кластера, где находятся отключенные сервисные Pods, т.е. регион кластера, где настроена политика.    |
| <b>Disaster Recovery Region</b> | Регион, который предоставляет балансировку нагрузки для кластера, где находятся отключенные сервисные Pods. |

### Метод валидации

1. В верхней навигационной панели щелкните переключатель просмотра продукта, чтобы переключиться на **Container Platform** и войдите в пространство имен ns1 под кластером c1.
2. В левом навигационном меню нажмите **Compute Components > Deployments**.
3. Нажмите **имя развертывания** продукта productpage.
4. Нажмите **Stop** в правом верхнем углу, чтобы остановить все Pods продукта productpage.
5. Обратитесь к методу Проверка доступа для продолжения доступа к службе productpage на основе конфигурации маршрутизации (External Access Address), созданной шлюзом, развернутым в кластере c1.

**Примечание:** Если к приложению Bookinfo по-прежнему можно получить доступ, это означает, что отказоустойчивость между кластерами была достигнута. Кроме того, вы можете перейти на платформу Service Mesh, чтобы дополнительно проверить пересылку трафика, просматривая топологию услуги (между кластерами) и цепочки вызовов.

### Настройка и проверка распределения веса кластеров между регионами

Этот документ объясняет, как добиться распределения веса между кластерами для приложений на основе многокластерной сервисной сетки, используя приложение Bookinfo в качестве примера.

#### Шаги

1. В левой панели навигации щелкните **Service Mesh > Mesh**.
2. Щелкните по **имени сервисной сетки**, которую необходимо настроить.
3. На вкладке Mesh Policies щелкните Create Policy > Regional Load Balancing на правой стороне кластера c1.
4. Выберите **Weight Distribution** в качестве **Policy Type** и настройте связанные параметры, как описано ниже.

| Параметр | Описание |
|----------|----------|
|----------|----------|

|                |                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Region</b>  | Регион, способный предоставить балансировку нагрузки между регионами в пределах текущей сервисной сетки. <b>Текущий регион</b> - это регион, к которому принадлежит кластер. |
| <b>Cluster</b> | Кластер в данном регионе. Когда в одном регионе имеется несколько кластеров, трафик будет равномерно распределяться между кластерами в этом регионе.                         |
| <b>Weight</b>  | Процент нагрузки трафика.<br><u>Примечание:</u> Общий вес всех регионов должен равняться 100.                                                                                |

**Пример:** Установите вес региона, в котором находится кластер c1, на 80%, а вес региона, в котором находится кластер c2, на 20%.

5. Щелкните **Create**.

#### Метод валидации

Обратитесь к методу Access Verification для сервиса productpage, чтобы постоянно получать доступ к сервису productpage на основе конфигурации маршрутизации ( External Access Address), созданной шлюзом, развернутым в кластере c1.

Вы можете перейти на страницу **Monitoring** платформы Service Mesh и сравнить данные мониторинга сервиса productpage в каждом кластере на панели **Cluster Traffic Comparison**. Если данные мониторинга трафика соответствуют весовым пропорциям, это указывает на то, что распределение веса между кластерами эффективно.

## Развёртывание Bookinfo

### Прerequisites

Перед началом убедитесь, что у вас есть аккаунт в **Customer Portal** с доступом к загрузке из **App Market**.

Если у вас нет аккаунта, обратитесь в техническую поддержку.

### Шаг 1. Создание проекта и namespace для Bookinfo

Если у вас уже есть подходящий проект и namespace для развёртывания Bookinfo, этот шаг можно пропустить, заменив namespace в последующих командах на ваш.

**Важно:** Создание проекта требует прав администратора платформы. Если у вас нет необходимых прав, обратитесь к администратору за помощью.

Чтобы создать проект и пространство имён:

1. В разделе **Project Management** нажмите **Create Project**.

2. Задайте имя demo и выберите кластер, в котором планируете развернуть приложение Bookinfo.
3. После создания проекта перейдите на страницу с деталями проекта.
4. В левой навигационной панели нажмите **Namespaces**, затем **Create Namespace** и задайте имя demo-dev.

## Шаг 2. Загрузка ресурсов Bookinfo

1. Запросите у MWS приложение Bookinfo.

## Шаг 3. Загрузка образов

1. Загрузите скачанный пакет на сервер или ПК с доступом к вашему реестру образов.
2. Распакуйте пакет, чтобы получить следующие файлы:
3. `tar -xzf <package>`
  - bookinfo-images-arm64.tar: образы Bookinfo для архитектуры ARM.
  - bookinfo-images-amd64.tar: образы Bookinfo для архитектуры x86.
  - push-images.sh: скрипт для загрузки образов, автоматически обновляющий ссылки на образы в bookinfo.yaml.
  - bookinfo.yaml: YAML-файл для развертывания Bookinfo, включающий конфигурации Service и Deployment.
4. Назначьте права на выполнение скрипта:

`chmod +x push-images.sh`

5. Запустите скрипт для загрузки образов: в целевой реестр:

```
Пример ./push-images.sh ./bookinfo-images-amd64.tar 192.168.143.0:32767 true
admin 'Aa123!@#' bookinfo-demo
./push-images.sh <source-image> <target-registry> <requires-auth> [username]
[password] [harbor-project]
```

Где:

- source-image: путь к файлу образа;
- target-registry: адрес и порт реестра образов;

**Важно:** префикс `https://` или `http://` указывать не нужно.

- requires-auth: требуется ли аутентификация. Если нет — укажите false; если да — true и дополнительно укажите username и password;
- username: имя пользователя для реестра образов;
- password: пароль для реестра образов. **Рекомендуется** заключать пароль в одинарные кавычки, чтобы избежать ошибок разбора командной строки из-за специальных символов;
- harbor-project: если используется реестр Harbor, укажите имя проекта в Harbor; если Docker Registry — можно оставить пустым.

#### Шаг 4. Развёртывание приложения Bookinfo

1. Откройте контейнерную платформу или страницу управления платформой, затем нажмите на инструмент kubectl в правом нижнем углу.
2. Примените YAML-файл: С помощью vi или другого терминального редактора вставьте содержимое bookinfo.yaml и выполните команду:

```
kubectl apply -f bookinfo.yaml -n <your-namespace>
```

3. Проверьте успешность развёртывания:

- Убедитесь, что все Pods находятся в состоянии Running:

```
kubectl get pods -n <your-namespace>
```

- Проверьте доступность приложения:

```
kubectl exec "$(kubectl get pod -n <your-namespace> -l app=ratings -o
jsonpath='{.items[0].metadata.name}')" \
-n <your-namespace> -c ratings -- curl -s productpage:9080/productpage | \
grep -o "<title>.*</title>"
```

Ожидаемый вывод:

```
<title>Simple Bookstore App</title>
```

## DevOps

### Введение в MWS DevOps

MWS DevOps — это платформа DevOps, созданная на основе Kubernetes для быстрого создания и доставки приложений. От кода до производства MWS DevOps предоставляет полный набор инструментов DevOps, который включает управление исходным кодом, управление артефактами и зависимостями, непрерывную интеграцию, непрерывную доставку и контроль качества исходного кода.

### Архитектура

MWS DevOps состоит из нескольких надстроек, чтобы создать полный цикл DevOps, включая:

- **Пайплайны MWS DevOps:** облачные нативные потоки и управление задачами на основе Tekton
- **Коннекторы MWS DevOps:** безопасный и эффективный способ подключения к внешним услугам и инструментам DevOps
- **Сборка MWS Gitlab:** платформа DevOps с открытым исходным кодом на основе веб-технологий, предоставляющая управление кодом, непрерывную интеграцию и возможности отслеживания проблем для команд разработки программного обеспечения
- **Сборка MWS Harbor:** реестр образов, который безопасно хранит, подписывает и сканирует образы контейнеров с управлением доступом на основе ролей, сканированием уязвимостей и доверия к содержимому
- **Сборка MWS SonarQube:** продвинутая платформа для обеспечения качества и безопасности кода, предлагает повышенную безопасность с комплексным обнаружением уязвимостей и обеспечивает более высокую продуктивность разработчиков за счет сокращения времени анализа и бесшовной интеграции с IDE
- **Сборка MWS Nexus:** универсальный менеджер артефактов, позволяющий хранить, проксировать и управлять различными типами пакетов и контейнеров



## Обзор возможностей

### Пайплайны MWS DevOps

- Tekton Pipelines

Основу Tekton, которая предоставляет пользовательские ресурсы для определения рабочих процессов CI/CD в виде серии шагов, задач и пайплайнов.

- Tekton Triggers

Система, основанная на событиях, которая отслеживает внешние события и автоматически запускает выполнение пайплайнов в ответ на них.

- Tekton Chains

Рамочная структура безопасности поставок, которая автоматически захватывает и подписывает информацию о происхождении для всех выполнений пайплайнов.

- Интеграция Tekton Hub

Доступ к каталогу повторно используемых задач и пайплайнов для ускорения разработки.

- Tekton Results

Долговременное хранение и запрос истории выполнения пайплайнов и журналов.

- Интеграция вебхуков

Автоматический запуск пайплайнов из репозитория Git и реестров контейнеров.

### Коннекторы MWS DevOps

- Коннекторы

Предоставляет единый интерфейс для безопасного подключения к внешним инструментам и платформам. От стандартных протоколов, таких как git и oci, до коннекторов, ориентированных на платформу, и пользовательских коннекторов – весь инструментарий можно легко интегрировать в процесс DevOps.

- Класс коннектора

Создает стандартный интерфейс для настройки и расширения возможностей коннекторов и поддерживаемых инструментов без изменения существующих коннекторов.

- Драйвер коннектора CSI

Способ использования коннекторов в облачных нативных рабочих нагрузках, используя инструменты, которые вы уже знаете, изменяя лишь то, как настраивается рабочая нагрузка.

### Развёртывание и управление инструментами

- Инструменты

Предоставляет шаблоны для легкого развертывания инструментов, включая Gitlab, Harbor, SonarQube и Nexus.

- Пользовательские Helm Chart

Позволяет пользователям настраивать все параметры развертывания инструмента на основе Helm.

## Пайплайны

Tekton — это мощная и гибкая открытая платформа для создания систем CI/CD, позволяющая разработчикам создавать, тестировать и развертывать приложения на облачных провайдерах и локальных системах.

Как часть Continuous Delivery Foundation, проекта Linux Foundation, Tekton предоставляет набор пользовательских ресурсов Kubernetes, которые служат строительными блоками для создания облачных CI/CD пайплайнов.

Оператор MWS DevOps Pipelines предлагает упрощенный способ установки, обновления и управления жизненным циклом компонентов Tekton в кластерах Kubernetes. Он расширяет функциональность Kubernetes, устанавливая и управляя Tekton Pipelines и его связанными компонентами, что упрощает настройку и поддержку надежной среды CI/CD.

## Ключевые компоненты

### *Оператор MWS DevOps Pipelines*

Оператор MWS DevOps Pipelines — это оператор Kubernetes, который управляет жизненным циклом компонентов Tekton. Он предоставляет:

- Упрощенную установку и управление компонентами Tekton
- Автоматические обновления и управление конфигурацией
- Централизованное управление ресурсами Tekton по всему кластеру
- Определения пользовательских ресурсов (CRD) для управления компонентами Tekton

### *Tekton Pipelines*

Tekton Pipelines — это основной компонент, который предоставляет основные строительные блоки для создания рабочих процессов CI/CD. Он включает в себя:

- **Задачи:** Наименьшие, атомарные единицы работы, выполняющие определенные операции
- **Пайплайны:** Коллекция задач, расположенных в определенном порядке выполнения
- **Запуски задач (TaskRuns):** Выполнение и статус задачи
- **Запуски пайплайнов (PipelineRuns):** Выполнение и статус пайплайна
- **Рабочие пространства (Workspaces):** Механизм для обмена данными между задачами

Tekton Pipelines позволяет вам определять сложные рабочие процессы в виде кода, делая их контролируемыми версиями, повторно используемыми и портируемыми между различными окружениями.

## ***Tekton Triggers***

Tekton Triggers расширяет Tekton Pipelines возможностями на основе событий, позволяя вам:

- **Слушать** внешние события (вебхуки из репозитория Git, реестров контейнеров и т. д.)
- **Извлекать** информацию из этих событий
- **Создавать** PipelineRuns или TaskRuns на основе данных событий
- **Передавать** данные событий в созданные PipelineRuns или TaskRuns

Это позволяет полностью автоматизированным CI/CD рабочим процессам реагировать на события, такие как коммиты кода, запросы на слияние кода или публикации образов ПО.

## ***Tekton Chains***

Tekton Chains сосредоточен на безопасности цепочки поставок, автоматически фиксируя и подписывая информацию о происхождении для всех TaskRuns:

- Автоматически скачивая и подписывая информацию о происхождении для всех TaskRuns
- Поддерживая несколько форматов подписей и облачных хранилищ
- Предоставляя аттестации в общепринятых форматах
- Защищая цепочку поставок программного обеспечения от источника до развертывания

Chains наблюдает за выполнением TaskRun, делает снимки, когда они завершаются, преобразует их в стандартные форматы, подписывает и надежно хранит.

## ***Tekton Results***

Tekton Results решает проблему долгосрочного хранения данных выполнения CI/CD:

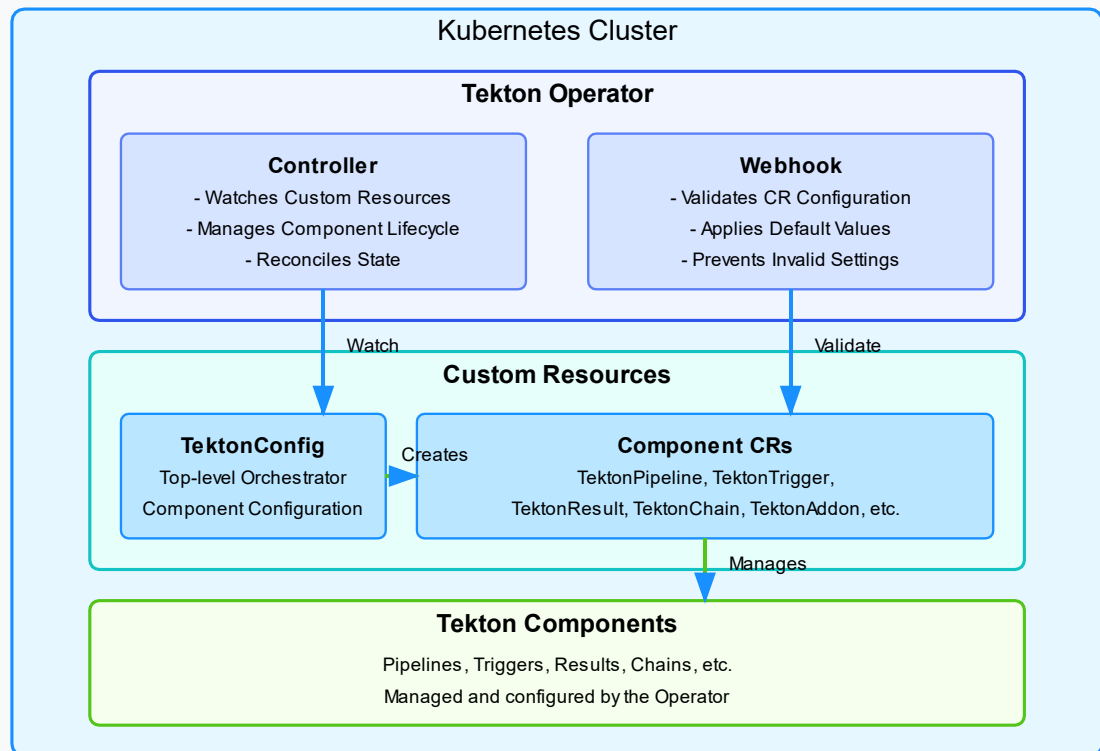
- Отделяя хранение результатов от контроллера пайплайнов
- Предоставляя API для доступа к историческим данным пайплайнов
- Позволяя использовать пользовательские метаданные для рабочих процессов CI/CD
- Группируя связанные рабочие загрузки вместе
- Храня логи от TaskRuns и PipelineRuns
- Освобождая ресурсы кластера, позволяя очищать завершенные запуски

## **Архитектура**

### ***Обзор архитектуры Tekton Operator***

Tekton Operator — это расширение Kubernetes, которое устанавливает, обновляет и управляет компонентами TektonCD на любом кластере Kubernetes. Оно следует паттерну Kubernetes Operator для обеспечения управления жизненным циклом компонентов Tekton.

## Tekton Operator Architecture



Tekton Operator Architecture - Custom Resource Management Flow

### Основные компоненты

#### Определения пользовательских ресурсов (CRDs)

Tekton Operator определяет несколько пользовательских ресурсов для управления различными компонентами Tekton:

- **TektonConfig**: ресурс верхнего уровня, который координирует установку других компонентов
- **TektonPipeline**: управляет компонентом Tekton Pipelines
- **TektonTrigger**: управляет компонентом Tekton Triggers
- **TektonResult**: управляет компонентом Tekton Results
- **TektonChain**: управляет компонентом Tekton Chains
- **OpenShiftPipelinesAsCode**: Управляет компонентом Pipelines as Code

#### Контроллер

Контроллер оператора отслеживает изменения в этих пользовательских ресурсах и сопоставляет фактическое состояние с целевым:

- Обнаруживает изменения в пользовательских ресурсах
- Устанавливает или обновляет соответствующие компоненты Tekton
- Управляет зависимостями между компонентами
- Обрабатывает изменения конфигурации

- Выполняет проверки на работоспособность и восстановление после ошибок

## Вебхуки

Компонент вебхука обеспечивает валидацию и установку значений по умолчанию для пользовательских ресурсов:

- Проверяет конфигурации пользовательских ресурсов
- Устанавливает значения по умолчанию, где это необходимо
- Предотвращает недопустимые конфигурации

## Пошаговая инструкция

| Номер | Оперативные шаги               | Описание                                                          |
|-------|--------------------------------|-------------------------------------------------------------------|
| 1     | Установка Tekton Operator      | Развернуть Tekton Operator в вашем кластере Kubernetes            |
| 2     | Провка установки оператора     | Подтвердить, что Tekton Operator работает правильно               |
| 3     | Установка компонентов Tekton   | Использовать Operator для установки Tekton Pipeline, Triggers     |
| 4     | Проверка установки компонентов | Проверить, что компоненты Tekton правильно установлены и работают |

### Шаг 1. Установка Tekton Operator

Tekton Operator позволяет вам устанавливать, обновлять и управлять компонентами Tekton Pipeline и Triggers в вашем кластере Kubernetes.

1. Откройте Operator Hub на платформе Marketplace управления
2. Найдите "MWS DevOps Pipelines" и нажмите "Установить"
3. Выберите пространство имен для установки оператора, например tekton-operator
4. После установки вы увидите pod оператора в пространстве имен

### Шаг 2. Проверка установки оператора

После установки Tekton Operator вам следует проверить, что он работает правильно, прежде чем продолжать.

1. Проверьте, что поды оператора работают:  
`kubectl get pods -n tekton-operator`
2. Убедитесь, что CRD (определения пользовательских ресурсов) оператора установлены:  
`kubectl get crds | grep tekton`

### Шаг 3. Установка компонентов Tekton

Tekton Operator использует пользовательские ресурсы для управления компонентами Tekton. По умолчанию оператор установит Pipeline, Triggers, Chains и Hub автоматически. Вы можете установить Results, Pipelines-as-Code, создавая эти пользовательские ресурсы.

## Шаг 4. Проверка установки компонентов

После установки компонентов Tekton вам следует проверить, что они работают правильно.

1. Проверьте ресурс TektonConfig:

```
kubectl get tektonconfig config
```

Результаты должны быть похожи на следующие:

```
NAME VERSION READY REASON
config v0.74.1-fb53414 True
```

2. Если ресурс TektonConfig не готов, вы можете продолжать проверять ресурс TektonInstallerSet для получения подробностей.

```
kubectl get tektoninstallerset
```

Результаты должны быть похожи на следующие:

```
NAME READY REASON
chain-config-5scx2 True
chain-secret-xrq7b True
chain-t6wb6 True
pipeline-main-deployment-9d9cc True
pipeline-main-static-c7zlz True
result-j74p5 True
tekton-hub-api-jl9xj True
tekton-hub-db-5g89c True
tekton-hub-db-migration-q7m8l True
tekton-hub-ui-jn6bw True
tektoncd-pruner-fgzfh True
trigger-main-deployment-q8zkd True
trigger-main-static-8x5w6 True
validating-mutating-webhook-ksz2k True
```

### Результаты

После завершения всех шагов у вас должны быть:

- Рабочий Tekton Operator в пространстве имен,
- Один или несколько установленных компонентов Tekton (Pipeline, Triggers, Hub) в пространстве имен,
- Определения пользовательских ресурсов (CRD) для управления ресурсами Tekton.

## Настройка квот ресурсов для компонентов пайплайна

Настройте квоты ресурсов, связанные с компонентом Pipeline.

### Сценарии использования

- Настройка квот ресурсов для компонента Pipeline
- Установка значений по умолчанию для квот ресурсов для init-контейнеров и контейнеров, создаваемых TaskRun

### Предварительные условия

- Компонент Tekton Operator должен быть установлен

- Ресурсы TektonConfig должны быть автоматически созданы в среде
- Следует ознакомиться с документом о Корректировке дополнительных элементов конфигурации для подставляющих

### **Пошаговая инструкция**

#### **Шаг 1**

Отредактируйте ресурс

```
kubectrl edit tektonconfigs.operator.tekton.dev config
```

#### **Шаг 2**

**Внимание:** Модификация конфигурации может вызвать поэтапное обновление компонентов Pods, что может привести к временной недоступности сервиса.

Пример изменения конфигурации:

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
 name: config
spec:
 pipeline:
 options:
 disabled: false

 configMaps:
 config-defaults:
 data:
 # Добавьте квоты ресурсов для контейнеров по умолчанию
 default-container-resource-requirements: |
 place-scripts: # обновляет требования к ресурсам контейнера 'place-scripts'
 requests:
 memory: "64Mi"
 cpu: "50m"
 limits:
 memory: "128Mi"
 cpu: "100m"

 prepare: # обновляет требования к ресурсам контейнера 'prepare'
 requests:
 memory: "64Mi"
 cpu: "50m"
 limits:
 memory: "256Mi"
 cpu: "100m"

 working-dir-initializer: # обновляет требования к ресурсам контейнера
 'working-dir-initializer'
 requests:
```

```
memory: "64Mi"
cpu: "50m"
limits:
 memory: "512Mi"
 cpu: "100m"
```

prefix-scripts: # обновляет требования к ресурсам контейнеров, которые начинаются с 'scripts-'

```
requests:
 memory: "64Mi"
 cpu: "50m"
limits:
 memory: "128Mi"
 cpu: "100m"
```

prefix-sidecar-scripts: # обновляет требования к ресурсам контейнеров, которые начинаются с 'sidecar-scripts-'

```
requests:
 memory: "64Mi"
 cpu: "50m"
limits:
 memory: "128Mi"
 cpu: "100m"
```

sidecar-tekton-log-results: # обновляет требования к ресурсам контейнера 'place-scripts'

```
requests:
 memory: "64Mi"
 cpu: "50m"
limits:
 memory: "128Mi"
 cpu: "100m"
```

deployments:

tekton-pipelines-controller:

```
spec:
 replicas: 1
 template:
 spec:
 containers:
 - name: tekton-pipelines-controller
 resources:
 requests:
 cpu: 500m
 memory: 512Mi
 limits:
 cpu: "1"
 memory: 1Gi
```



tekton-pipelines-remote-resolvers:

```
spec:
 replicas: 1
 template:
 spec:
 containers:
 - name: controller
 resources:
 requests:
 cpu: 200m
 memory: 256Mi
 limits:
 cpu: 500m
 memory: 512Mi
```

tekton-pipelines-webhook:

```
spec:
 replicas: 1
 template:
 spec:
 containers:
 - name: webhook
 resources:
 requests:
 cpu: "500m"
 memory: 256Mi
 limits:
 cpu: "1"
 memory: 500Mi
```

tekton-events-controller:

```
spec:
 replicas: 1
 template:
 spec:
 containers:
 - name: tekton-events-controller
 resources:
 requests:
 cpu: 100m
 memory: 100Mi
 limits:
 cpu: 200m
 memory: 256Mi
```

### Шаг 3

Подайте конфигурацию и дождитесь обновления Pods.

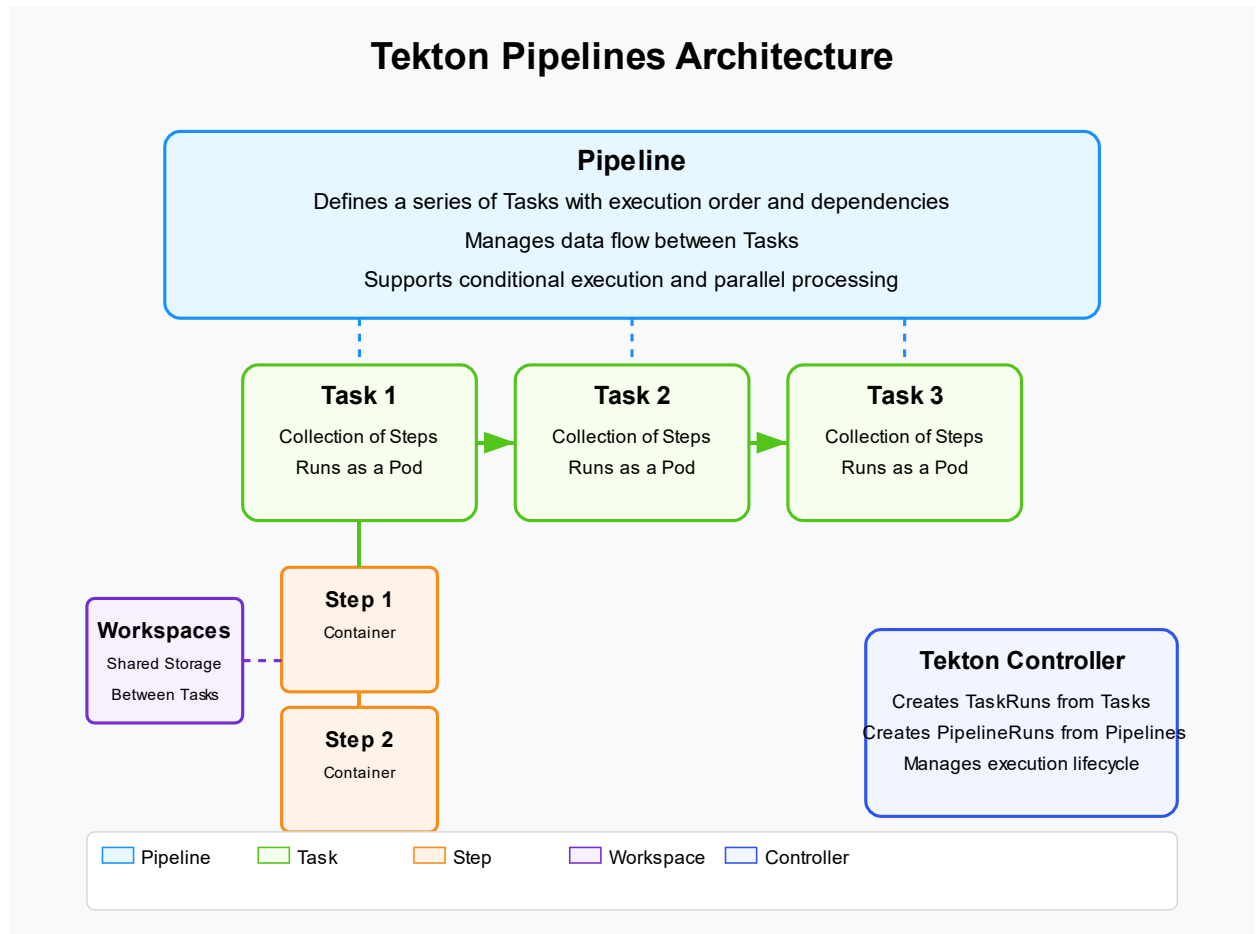
```
$ kubectl get pods -n tekton-pipelines -w
```

NAME READY STATUS

|                                                    |     |         |
|----------------------------------------------------|-----|---------|
| tekton-pipelines-controller-648d87488b-fq9bc       | 1/1 | Running |
| tekton-pipelines-remote-resolvers-79554f5959-cbm6x | 1/1 | Running |
| tekton-pipelines-webhook-5cd9847998-864zf          | 1/1 | Running |
| tekton-events-controller-5c97b7554c-m59m6          | 1/1 | Running |

### Компоненты пайплайна

Tekton Pipelines — это основанная на Kubernetes система CI/CD, которая предоставляет набор стандартных определений пользовательских ресурсов (CRD) для определения и выполнения CI/CD рабочих процессов в виде последовательности шагов. Она предназначена для облачной среды, масштабируемости и расширяемости.



### Основные компоненты

Tekton Pipeline работает с помощью следующих компонентов:

#### Задача (Task)

- Определяет серию шагов, которые должны выполняться последовательно
- Выполняется в изолированных контейнерах внутри пода Kubernetes
- Может определять входные и выходные данные, а также требуемые ресурсы

#### Запуск задачи (TaskRun)

Выполнение задачи — это экземпляр задачи во время выполнения.

- Создает экземпляр и выполняет задачу

- Предоставляет параметры и ресурсы во время выполнения
- Отслеживает статус выполнения и результаты

### **Пайплайн (Pipeline)**

Пайплайн определяет серию задач для выполнения.

- Определяет упорядоченную серию задач
- Указывает порядок выполнения и зависимости
- Управляет потоком данных между задачами

### **Запуск пайплайна (PipelineRun)**

Запуск пайплайна — это экземпляр пайплайна во время выполнения

- Создает экземпляр и выполняет пайплайн
- Предоставляет параметры и ресурсы во время выполнения
- Отслеживает общий статус выполнения пайплайна

### **Рабочие пространства (Workspaces)**

Рабочие пространства предоставляют механизм для совместного использования данных.

- Гибкие параметры монтирования
- Обеспечивает общую память между задачами
- Поддерживает различные бекенды хранения (PVC, ConfigMap и др.)

### **Модель выполнения пайплайна**

Когда пайплайн выполняется через PipelineRun:

1. Tekton анализирует пайплайн, чтобы определить порядок выполнения задач
2. Задачи без зависимостей начинают выполняться немедленно
3. Задачи с зависимостями ожидают завершения своих зависимостей
4. Задачи, которые могут выполняться параллельно, выполняются одновременно
5. Пайплайн завершается, когда все задачи завершены или когда критическая задача терпит неудачу

### **Пошаговая инструкция**

| Шаг | Операция                      | Описание                                                          |
|-----|-------------------------------|-------------------------------------------------------------------|
| 1   | Создание задачи "Hello World" | Создание базовой задачи, которая выводит приветственное сообщение |
| 2   | Создание задачи "Goodbye"     | Создание второй задачи, которая принимает параметры               |
| 3   | Создание пайплайна            | Создание пайплайна, который соединяет обе задачи последовательно  |
| 4   | Запуск пайплайна              | Выполнение пайплайна и просмотр результатов                       |
| 5   | Очистка                       | Удаление созданных ресурсов                                       |

## Шаг 1. Создание задачи "Hello World"

Задача — это основной строительный блок пайплайна Tekton. Она состоит из серии шагов, которые выполняются последовательно в контейнере.

1. Создайте файл с именем `hello-task.yaml` и следующим содержимым:

```
apiVersion: tekton.dev/v1
kind: Task
metadata:
 name: hello
spec:
 steps:
 - name: echo
 image: ubuntu
 command:
 - echo
 args:
 - "Hello World!"
```

2. Примените задачу к вашему кластеру:

```
kubectrl apply -f hello-task.yaml
```

Объект Task Run создает экземпляр и выполняет эту задачу. Создайте другой файл с именем `hello-world-run.yaml` и следующим содержимым:

```
apiVersion: tekton.dev/v1
kind: TaskRun
metadata:
 name: hello-task-run
spec:
 taskRef:
 name: hello
```

3. Примените TaskRun к вашему кластеру:

```
kubectrl apply -f hello-world-run.yaml
```

4. Мониторинг выполнения TaskRun:

```
kubectrl get taskrun hello-task-run
```

Вы должны увидеть вывод, подобный этому:

| NAME           | SUCCEEDED | REASON    | STARTTIME | COMPLETIONTIME |
|----------------|-----------|-----------|-----------|----------------|
| hello-task-run | True      | Succeeded | 1m        | 2m             |

5. Ознакомьтесь с логами:

```
kubectrl logs --selector=tekton.dev/taskRun=hello-task-run
```

Вы должны увидеть вывод, подобный этому:

```
Hello World!
```

## Шаг 2. Создание задачи "Goodbye" с параметрами

Теперь создадим вторую задачу, которая принимает параметры, чтобы сделать её более гибкой.

1. Создайте файл с именем `goodbye-task.yaml` и следующим содержимым:

```

apiVersion: tekton.dev/v1
kind: Task
metadata:
 name: goodbye
spec:
 params:
 - name: username
 type: string
 description: Имя пользователя, которому следует попрощаться
 default: "World"
 steps:
 - name: goodbye
 image: ubuntu
 command:
 - echo
 args:
 - "Goodbye ${params.username}!"

```

2. Примените задачу к вашему кластеру:

```
kubectl apply -f goodbye-task.yaml
```

### Шаг 3. Создание пайплайна

Пайплайн определяет упорядоченную серию задач в рамках вашего рабочего процесса CI/CD.

1. Создайте файл с именем hello-goodbye-pipeline.yaml и следующим содержимым:

```

apiVersion: tekton.dev/v1
kind: Pipeline
metadata:
 name: hello-goodbye
spec:
 params:
 - name: username
 type: string
 description: Имя пользователя, которому следует попрощаться
 default: "World"
 tasks:
 - name: hello
 taskRef:
 name: hello
 - name: goodbye
 runAfter:
 - hello
 taskRef:
 name: goodbye
 params:
 - name: username
 value: "${params.username}"

```

Этот пайплайн:

- Определяет параметр с именем username
- Включает две задачи:
  - Указывает, что hello и goodbye должен выполняться после завершения hello
  - Передает параметр пайплайна в задачу goodbye

2. Примените пайплайн к вашему кластеру:

```
kubectl apply -f hello-goodbye-pipeline.yaml
```

#### Шаг 4. Запуск пайплайна

PipelineRun используется для выполнения пайплайна и предоставления любых необходимых параметров.

1. Создайте файл с именем hello-goodbye-pipeline-run.yaml и следующим содержимым:

```
apiVersion: tekton.dev/v1
kind: PipelineRun
metadata:
 name: hello-goodbye-run
spec:
 pipelineRef:
 name: hello-goodbye
 params:
 - name: username
 value: "Tekton"
```

2. Примените PipelineRun к вашему кластеру, чтобы запустить пайплайн:

```
kubectl apply -f hello-goodbye-pipeline-run.yaml
```

3. Мониторинг выполнения пайплайна:

```
tkn pipelinerun logs hello-goodbye-run -f
```

Вы должны увидеть вывод, подобный этому:

```
[hello : echo] Hello World!
[goodbye : goodbye] Goodbye Tekton!
```

#### Шаг 5. Очистка

Чтобы удалить ресурсы, созданные в этом учебнике

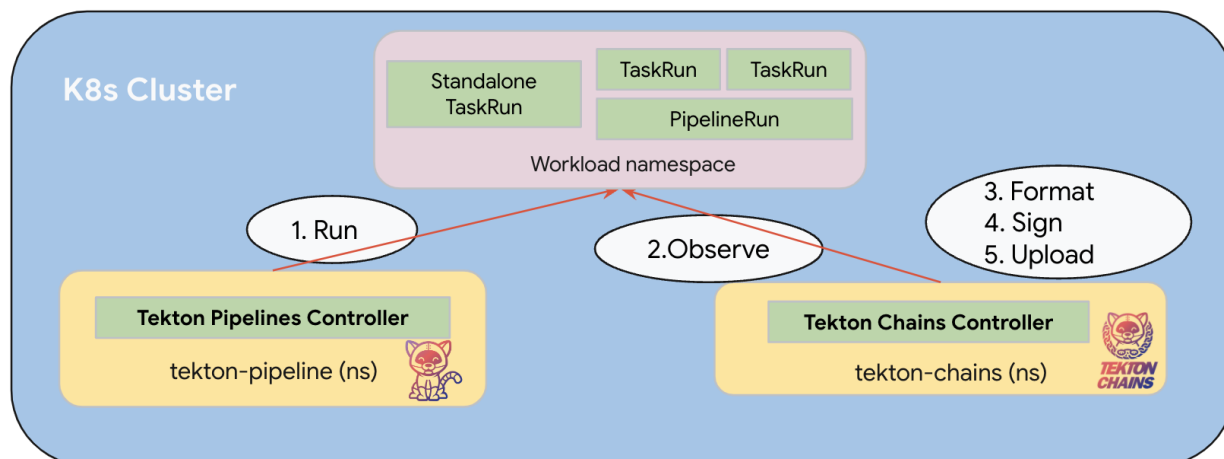
```
kubectl delete -f hello-task.yaml
kubectl delete -f hello-task-run.yaml
kubectl delete -f goodbye-task.yaml
kubectl delete -f hello-goodbye-pipeline.yaml
kubectl delete -f hello-goodbye-pipeline-run.yaml
```

#### Безопасность пайплайнов

Tekton Chains — это контроллер определения пользовательского ресурса Kubernetes (CRD), который предоставляет функции безопасности цепочки поставок для пайплайнов Tekton. Он работает, наблюдая за выполнением TaskRun и PipelineRun, захватывая соответствующую информацию и сохраняя ее в формате с криптографической подписью.

Это обеспечивает безопасный и проверяемый учет всех действий в пайплайне, гарантируя целостность вашей цепочки поставок программного обеспечения.

Основанный на Tekton Pipelines, Tekton Chains расширяет возможности CI/CD с функциями безопасности, которые позволяют организациям реализовывать практики безопасности цепочки поставок программного обеспечения в соответствии с отраслевыми стандартами, такими как SLSA (Уровни цепочки поставок для программных артефактов).



Примечание: Диаграмма выше иллюстрирует высокоуровневую архитектуру Tekton Chains. Фактическая реализация может различаться в зависимости от конфигурации.

## Основные компоненты

### Контроллер Tekton Chains

Контроллер Tekton Chains — это центральный компонент архитектуры. Он работает как контроллер Kubernetes, который:

- Наблюдает за завершениями TaskRun и PipelineRun в кластере
- Делает снимки завершенных TaskRun и PipelineRun
- Обрабатывает эти снимки для генерации криптографических подписей и аттестаций
- Хранит подписи и аттестации в настроенных хранилищах

Контроллер работает как Deployment Kubernetes в пространстве имен и отвечает за всю основную функциональность Tekton Chains.

### Компоненты подписи

Tekton Chains поддерживает несколько механизмов подписи для обеспечения гибкости в том, как защищаются артефакты:

- **Подпись X509:** Использует сертификаты X509 для подписи артефактов
- **Подпись Cosign:** Интегрируется с Cosign для подписи контейнерных образов
- **Подпис KMS:** Поддерживает различные системы управления ключами:
  - GCP KMS
  - AWS KMS
  - Azure KMS

- HashiCorp Vault

Эти компоненты подписи отвечают за криптографическую подпись артефактов и аттестаций, сгенерированных Tekton Chains.

### Компоненты хранения

Tekton Chains может хранить подписи и аттестации в различных хранилищах:

- **Хранилище Tekton:** хранит подписи в виде аннотаций на объектах TaskRun или PipelineRun
- **OCI хранилище:** хранит подписи и аттестации вместе с контейнерными образами в OCI-совместимых реестрах
- **GCS хранилище:** хранит подписи и аттестации в бакетах Google Cloud Storage
- **DocDB хранилище:** хранит подписи и аттестации в документальных базах данных (например, MongoDB, Firestore)
- **Grafeas хранилище:** хранит подписи и аттестации на серверах Grafeas

### Поток данных

Поток данных в Tekton Chains выполняет следующие шаги:

- **Наблюдение:** Контроллер Chains наблюдает за выполнением TaskRun и PipelineRun в кластере Kubernetes.
- **Создание снимка:** Когда TaskRun или PipelineRun завершается, контроллер создает снимок его состояния, включая входные данные, выходные данные и детали выполнения.
- **Преобразование формата:** Снимок преобразуется в один или несколько стандартных форматов полезной нагрузки (например, ин-тото аттестации, SLSA происхождение).
- **Подпись:** Форматированные полезные нагрузки подписываются с использованием настроенного механизма подписи (X509, Cosign, KMS).
- **Хранение:** Подписанные полезные нагрузки хранятся в настроенных хранилищах.
- **Проверка:** Внешние инструменты могут позже извлекать и проверять эти подписи, чтобы гарантировать целостность артефактов.

### Генерация SLSA происхождения

Для SLSA (Уровни цепочки поставок для программных артефактов) происхождения Tekton Chains:

- **Извлекает** информацию об входных артефактах (исходный код, зависимости) из результатов TaskRun
- **Фиксирует** детали окружения и процесса сборки
- **Регистрирует** информацию об исходных артефактах (контейнерные образы, бинарные файлы)
- **Генерирует** стандартизированный документ происхождения SLSA



- **Подписывает** документ происхождения
- **Хранит** подписанное происхождение вместе с артефактами или в указанном месте

### **Архитектура развёртывания**

Tekton Chains разворачивается как контроллер Kubernetes внутри кластера:

- **Пространство имен:** Контроллер работает в пространстве имен
- **Развертывание:** Одно развертывание управляет подами контроллера
- **Служебная учетная запись:** Контроллер использует выделенную служебную учетную запись с соответствующими правами
- **Конфигурация:** Конфигурация управляется через ConfigMap под названием
- **Секреты:** Ключи подписи и учетные данные хранятся как секреты Kubernetes

### **Высокая доступность**

Контроллер Tekton Chains может масштабироваться горизонтально для обеспечения высокой доступности:

- Можно развернуть несколько реплик для дублирования
- Выбор лидера обеспечивает, чтобы только один экземпляр обрабатывал события в данный момент
- Состояние поддерживается через ресурсы Kubernetes, что позволяет осуществлять безпроблемный переход при сбое

### **Точки интеграции**

Tekton Chains интегрируется с несколькими внешними системами:

- **Tekton Pipelines:** наблюдает за TaskRuns и PipelineRuns из Tekton Pipelines
- **OCI Реестры:** хранит подписи и аттестации вместе с контейнерными образами
- **Системы управления ключами:** интегрируется с различными поставщиками KMS для безопасного управления ключами
- **Журналы прозрачности:** может публиковать аттестации в журналы прозрачности, такие как Rekor
- **Системы проверки:** обеспечивает подписи, которые могут быть проверены системами, такими как Kyverno, OPA или Cosign

Эта архитектура позволяет Tekton Chains предоставлять комплексные функции безопасности цепочки поставок при сохранении гибкости в развертывании и конфигурации.

### **Пошаговая инструкция**

| Шаг | Операция                 | Описание                                            |
|-----|--------------------------|-----------------------------------------------------|
| 1   | Генерация ключей подписи | Создание пары ключей для подписи артефактов         |
| 2   | Настройка Tekton Chains  | Настройка Chains для использования хранилища Tekton |

|   |                        |                                        |
|---|------------------------|----------------------------------------|
| 3 | Запуск тестовой задачи | Создание и выполнение простого TaskRun |
| 4 | Проверка подписи       | Извлечение и проверка подписи TaskRun  |

### Шаг 1. Генерация ключей подписи

Tekton Chains использует криптографические ключи для подписи артефактов. По умолчанию он ищет секрет с именем signing-secrets в пространстве имен Chains.

1. Установите cosign, если вы этого еще не сделали
2. Сгенерируйте пару ключей и сохраните ее в виде секрета Kubernetes:

```
cosign generate-key-pair k8s://tekton-pipelines/signing-secrets
```

Вам будет предложено ввести пароль, который будет сохранен в секрете. Эта команда также сгенерирует публичный ключ cosign.pub в текущей папке, который будет использован позже при проверке.

3. Проверьте, что секрет был создан:

```
kubectl get secret signing-secrets -n tekton-pipelines
```

### Шаг 2. Настройка Tekton Chains

По умолчанию Tekton Chains настроен для хранения подписей в OCI реестре. Для этого быстрого старта мы настроим его для хранения подписей в виде аннотаций на самом TaskRun.

1. Настройте Chains для использования хранилища Tekton:

```
kubectl patch configmap chains-config -n tekton-pipelines -
p='{"data":{"artifacts.taskrun.storage": "tekton"}}'
```

Установите формат в in-toto (SLSA v0.2):

```
kubectl patch configmap chains-config -n tekton-pipelines -
p='{"data":{"artifacts.taskrun.format": "in-toto"}}'
```

2. Перезапустите контроллер Chains, чтобы применить изменения:

```
kubectl delete pod -n tekton-pipelines -l app=tekton-chains-controller
```

### Шаг 3. Запуск тестовой задачи

Теперь давайте создадим простой TaskRun, который Chains автоматически подпишет.

1. Создайте простой Task и TaskRun:

```
cat <<EOF | kubectl apply -f -
apiVersion: tekton.dev/v1
kind: Task
metadata:
 name: hello-world
spec:
 steps:
 - name: hello
 image: alpine
 script: |
 #!/bin/sh
```

```

 echo 'Hello, Tekton Chains!'

apiVersion: tekton.dev/v1
kind: TaskRun
metadata:
 name: hello-world-run
spec:
 taskRef:
 name: hello-world
EOF

```

2. Дождитесь завершения TaskRun:

```
kubectl get taskrun hello-world-run -w
```

Подождите, пока статус не станет Succeeded.

#### Шаг 4. Проверка подписи

После завершения TaskRun Tekton Chains автоматически подпишет его. Давайте проверим подпись.

1. Получите UID TaskRun:

```
export TASKRUN_UID=$(kubectl get taskrun hello-world-run -o
jsonpath='{.metadata.uid}')
```

2. Извлеките подпись:

```
kubectl get taskrun hello-world-run -o
jsonpath="{.metadata.annotations.chains\\.tekton\\.dev/signature-taskrun-
$TASKRUN_UID}" | base64 -d > signature
```

3. Проверьте подпись с помощью cosign:

```
cosign verify-blob-attestation \
--key cosign.pub \
--signature signature \
--type slsaprovenance \
--check-claims=false \
--insecure-ignore-tlog \
/dev/null
```

Если все прошло успешно, вы увидите сообщение «Verified OK».

4. Удалите созданные ресурсы:

```
kubectl delete taskrun hello-world-run
kubectl delete task hello-world
```

#### Ожидаемые результаты

После выполнения:

- У вас есть работающая установка Tekton Chains с ключом подписи,
- Ваши TaskRuns автоматически подписываются по завершении,
- Вы можете проверять подписи, чтобы гарантировать целостность ваших сборок.

Это демонстрирует основные функции Tekton Chains. В реальных сценариях вы бы:

1. Настроили Chains для подписания контейнерных образов и хранения подписей в вашем реестре.
2. Настроили этап проверки в вашем процессе развертывания.
3. Возможно, использовали облачный KMS для более безопасного управления ключами.

Для получения более сложных конфигураций обратитесь к документации Tekton Chains.

## **Коннекторы**

### **Введение**

Как правило, предприятиям необходимо интегрировать разнообразные инструменты в средах разработки, тестирования и эксплуатации, при этом поддерживая учетные данные для этих инструментов, чтобы иметь возможность получать доступ и управлять ресурсами внутри них. Примеры включают Git-репозитории, OCI Registry, K8S-кластеры и т.д.

Connectors предназначены для предоставления единой, открытой, удобной и безопасной возможности интеграции инструментов и доступа к ресурсам внутри Kubernetes кластеров.

### **Обзор возможностей**

#### **Единое сопровождение и управление информацией об доступе к инструментам**

Внутри кластера можно единообразно поддерживать различные типы доступных инструментов, а также уже интегрированные инструменты; адреса интегрированных инструментов, учетные данные и конфигурации, необходимые клиенту, также могут управляться централизованно.

Определяя типы инструментов, которые могут быть интегрированы, можно ограничить способ доступа к инструменту, например, адрес доступа к инструменту и метод аутентификации.

При интеграции инструментов можно одновременно настроить необходимые учетные данные для централизованного управления.

#### **Предоставление единого стиля Rest API для доступа к ресурсам внутри инструментов**

Connectors предоставляют набор API-интерфейсов с единым стилем для интегрированных инструментов, которые можно использовать для доступа к ресурсам внутри инструментов. Например, получение ссылок из Git-репозитория или тегов репозитория из OCI Registry.

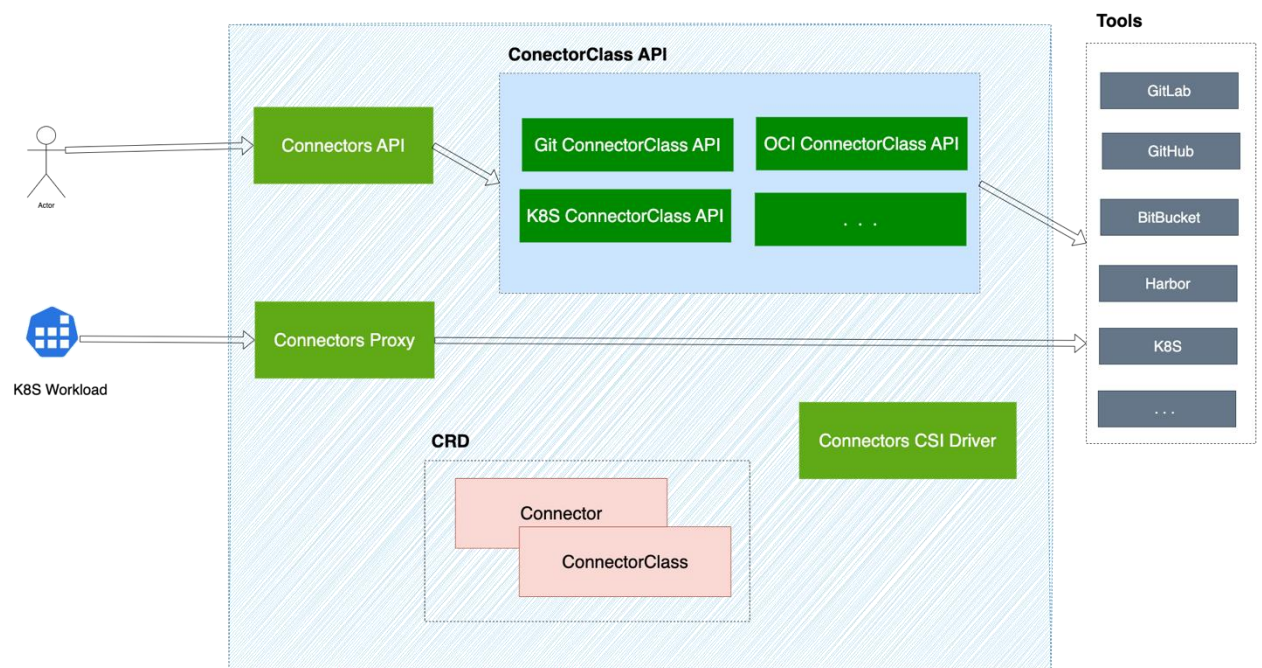
Клиенты могут опираться на эти API для доступа к ресурсам инструментов, что улучшает пользовательский интерфейс и опыт (UI/UX). Благодаря таким единообразным API клиентам не нужно беспокоиться о конкретном типе инструмента, достаточно использовать унифицированные API-интерфейсы.

#### **Обеспечение возможности доступа к ресурсам инструментов внутри кластера без использования секретов**

После интеграции инструментов в кластер доступ к ресурсам может осуществляться без необходимости распространять учетные данные клиентам, что обеспечивает безопасность этих учетных данных. Например:

- В K8S Job/Tekton Pipeline при выполнении Git clone нет необходимости хранить информацию для аутентификации Git Server внутри задания.
- В K8S Job/Tekton Pipeline при отправке образов нет необходимости хранить информацию для аутентификации OCI Registry внутри задания.

## Архитектура



Connector — это ресурс, представляющий интегрированный экземпляр конкретного инструмента. Настроив URL доступа к инструменту и данные для аутентификации, мы можем создать экземпляр для интеграции этого инструмента.

Например, интеграция с <https://github.com> с использованием GitHub Private Access Token осуществляется через Connector.

В Kubernetes Connector является кастомным ресурсом на уровне namespace. Пользователи могут создавать несколько Connector в одном namespace для интеграции разных инструментов.

Например, в namespace default можно создать как Connector для интеграции с <https://github.com>, так и Connector для интеграции с <https://hub.docker.com/>

Администраторы платформы могут управлять интеграциями инструментов по всему кластеру, управляя ресурсами Connector.

## ConnectorClass

ConnectorClass определяет методы доступа и спецификации поведения для конкретных типов инструментов. Он задаёт параметры, необходимые при интеграции с определённым типом инструмента, такие как адрес инструмента и данные для аутентификации.

Например, `Git ConnectorClass` определяет конфигурационные элементы, которые необходимо предоставить при интеграции с Git-инструментами, включая адрес Git-сервиса и данные аутентификации `BasicAuth`.

В `Kubernetes`, `ConnectorClass` является кастомным ресурсом на уровне кластера. Разработчики могут расширять поддерживаемые платформой типы инструментов, определяя новые `ConnectorClass`.

Например, можно определить `Harbor ConnectorClass` для поддержки интерации с репозиторием образов `Harbor`, `MySQL ConnectorClass` для интеграции с базами данных `MySQL` или `Jira ConnectorClass` для интеграции с инструментами управления `MySQL` или проектами `Jira`

## **Connectors API**

`Connectors API` предоставляет возможности доступа к внутренним ресурсам инструментов на основе экземпляров `Connector`. Например, для `Git Connector`, `Connectors API` может получить список веток (`References`) в Git-репозитории.

Разработчики могут удобно получать доступ к ресурсам внутри инструментов через `Connectors API`, не заботясь о конкретных адресах инструментов и деталях аутентификации.

Этот API очень полезен в практических приложениях, таких как:

- Получение списка тегов для контейнерных образов при создании приложений
- Получение списка веток (`References`) для репозитория кода при операции `Git Clone`

Реализация `Connectors API` основана на базовых возможностях, предоставляемых `ConnectorClass API`

## **ConnectorClass API**

`ConnectorsClass API` определяет API, предоставляемые конкретными типами инструментов.

Разные типы инструментов могут предлагать различные возможности API, например:

- **`GIT ConnectorClass API`** может предоставлять возможность получения списка веток для репозитория кода
- **`OCI ConnectorClass API`** может предоставлять возможность получения списка тегов для репозитория артефактов

Разработчики могут определять уникальные возможности API для каждого `ConnectorClass`, и эти возможности в конечном итоге будут доступны клиентам через `Connectors API`.

## **Connectors CSI Driver**

Для упрощения использования возможностей `Connectors-Proxy` рабочими нагрузками `K8S` можно использовать `Connectors CSI Driver`.

`Connectors CSI Driver` может монтировать отрендеренное содержимое шаблонов конфигурационных файлов, поддерживаемых в `ConnectorClass`, в рабочую нагрузку. Конфигурационный файл может содержать информацию для доступа к `Connectors Proxy`,

что позволяет пользователям использовать возможности Connectors Proxy с минимальными изменениями в их исходных скриптах.

Для получения дополнительной информации смотрите [connectors csi driver](#).

### Пошаговая инструкция для интеграции с GitHub

| № | Шаг операции                     | Описание                                                       |
|---|----------------------------------|----------------------------------------------------------------|
| 1 | Проверка необходимых компонентов | Убедиться, что инфраструктура коннектора установлена корректно |
| 2 | Создание секрета аутентификации  | Создать Kubernetes Secret с учетными данными GitHub            |
| 3 | Создание GitHub коннектора       | Настроить коннектор для GitHub                                 |
| 4 | Проверка статуса коннектора      | Проверить, готов ли коннектор к использованию                  |

#### Шаг 1. Проверка необходимых компонентов

Перед созданием GitHub коннектора убедитесь, что инфраструктура коннектора корректно установлена в вашем кластере.

1. Проверьте, что Connectors Operator, ConnectorsCore и ConnectorsGit установлены:

```
kubectrl get connectorscore -A
```

```
kubectrl get connectorsgit -A
```

Вы должны увидеть оба ресурса со статусом "Ready".

2. Убедитесь, что существует Git ConnectorClass:

```
kubectrl get connectorclass git
```

Если Git ConnectorClass не найден, сначала установите ConnectorsGit. Инструкции смотрите в Installation Guide.

#### Шаг 2. Создание секрета аутентификации

Создайте Kubernetes Secret, содержащий учетные данные для GitHub.

1. Выполните следующую команду для создания секрета с вашими учетными данными GitHub:

```
kubectrl create secret generic github-auth \
 --namespace default \
 --type=kubernetes.io/basic-auth \
 --from-literal=username=your-github-username \
 --from-literal=password=your-github-token
```

Замените your-github-username на ваш логин в GitHub, your-github-token на ваш Personal Access Token GitHub

2. Проверьте, что секрет успешно создан:

```
kubectrl get secret github-auth -n default
```

#### Шаг 3. Создание GitHub коннектора

Создайте коннектор к GitHub, используя Git ConnectorClass.

1. Сохраните следующий YAML в файл github-connector.yaml

```
apiVersion: connectors.alauda.io/v1alpha1
kind: Connector
metadata:
 name: github-connector
 namespace: default
spec:
 connectorClassName: git
 address: "https://github.com"
 auth:
 name: basicAuth
 secretRef:
 name: github-auth
 namespace: default
```

Примените коннектор:

```
kubectl apply -f github-connector.yaml
```

#### Шаг 4. Проверка статуса коннектора

После создания коннектора проверьте его статус, чтобы убедиться, что он работает корректно.

1. Используйте следующую команду для проверки статуса коннектора:

```
kubectl get connector github-connector -n default
```

2. Для просмотра подробной информации о статусе используйте:

```
kubectl get connector github-connector -n default -o yaml
```

#### Результаты

После успешного выполнения всех шагов вы увидите следующие результаты:

1. Статус ресурса GitHub коннектора отображается как «Ready»

| NAME             | CLASS | ADDRESS            | READY | AGE |
|------------------|-------|--------------------|-------|-----|
| github-connector | git   | https://github.com | True  | 2m  |

2. Все условия в подробном статусе имеют значение "True":

```
status:
 conditions:
 - type: ConnectorClassReady
 status: "True"
 - type: SecretReady
 status: "True"
 - type: LivenessReady
 status: "True"
 - type: AuthReady
 status: "True"
 - type: ProxyServiceReady
 status: "True"
 - type: Ready
 status: "True"
```



3. Если у коннектора настроена прокси-функциональность, в статусе будет указан адрес прокси:

status:

proxy:

httpAddress:

url: `http://c-github-connector.default.svc.cluster.local`

Теперь вы успешно создали GitHub коннектор и убедились, что он работает.

Ваши приложения могут использовать этот коннектор для безопасного доступа к репозиториям GitHub без необходимости встраивать учетные данные непосредственно в код приложений.

## GitLab

### Введение

GitLab Community Edition (CE) v17.8 — это комплексная платформа DevOps, которая предлагает полное решение для управления процессом разработки программного обеспечения в одном приложении. Как открытое, само размещаемое решение для разработчиков, GitLab CE предоставляет мощные возможности для управления исходным кодом, непрерывной интеграции и развертывания (CI/CD), сканирования безопасности и управления проектами.

GitLab CE v17.8 расширяет возможности платформы, предлагая улучшенные функции безопасности, улучшенное управление контейнерами и расширенные возможности CI/CD, что делает его идеальным решением для команд разработчиков любого размера, стремящихся оптимизировать свои рабочие процессы разработки программного обеспечения.

### Особенности

#### Особенности развёртывания

- Шаблонные развертывания

Поддержка готовых шаблонов для быстрого развертывания и конфигураций, готовых к эксплуатации, а также пользовательских шаблонов.

- Развертывания с использованием YAML

Настраиваемый метод развертывания с помощью YAML, позволяющий полностью контролировать конфигурацию экземпляра GitLab с использованием значений Helm.

#### Расширенные возможности

- Поддержка IPv6

Развертывание и эксплуатация экземпляров GitLab в сетевых средах с поддержкой только IPv6 или с двойным стеком, обеспечивая совместимость с современными сетевыми инфраструктурами

- Совместимость с архитектурой ARM

Полная поддержка инфраструктуры на базе ARM, позволяющая развертывать на энергоэффективных и экономичных ARM-процессорах для улучшенного соотношения производительности и энергопотребления.

- Усиление безопасности

Устранение критических уязвимостей системы и приложений с помощью проактивного патчинга и усиления безопасности. Включает исправления для проблем, идентифицированных по CVE, обновления безопасности зависимостей и устранение потенциальных векторов повышения привилегий

### **Установка оператора**

Перейдите на страницу **Administrator > Marketplace > Operator Hub**, выберите целевой кластер и разверните оператор MWS Build of Gitlab.

### **Развёртывание экземпляра GitLab**

В этом разделе представлено описание подписки на GitLab Operator и функционала развёртывания экземпляра GitLab.

#### ***Пререквизиты***

Данный документ применим к версиям GitLab 17 и выше, предоставляемым платформой. Он основан на технологиях, таких как Operator, и отделён от платформы.

Убедитесь, что GitLab Operator развернут (подписан) в целевом кластере, то есть экземпляр GitLab может быть создан через GitLab Operator.

#### ***Планирование развёртывания***

GitLab поддерживает различные конфигурации ресурсов для удовлетворения различных сценариев использования. В разных сценариях требуемые ресурсы и настройки могут значительно отличаться. Поэтому в этом разделе описаны аспекты, которые следует учитывать при планировании развёртывания экземпляра GitLab, а также влияние ключевых решений, чтобы помочь пользователям принимать обоснованные решения для последующих развёртываний.

### **Основная информация**

1. GitLab Operator, предоставляемый платформой, основан на официальном сообществе GitLab Operator с корпоративными улучшениями, такими как поддержка IPv6, ARM и исправления уязвимостей безопасности. Он полностью совместим с сообществом по функциональности и повышает удобство развёртывания GitLab за счёт опциональных и настраиваемых шаблонов
2. Экземпляр GitLab включает несколько компонентов, таких как Gitaly — компонент PostgreSQL для доступа к Git-репозиториям, — компонент для хранения метаданных приложений и информации о пользователях, и Redis — компонент для кэширования. Платформа предоставляет профессиональные PostgreSQL Operator и Redis Operator, поэтому при развёртывании экземпляра GitLab ресурсы Redis и

PostgreSQL не разворачиваются напрямую, а используются через настройку соответствующих учётных данных доступа к существующим экземплярам

### **Планирование ресурсов перед развертыванием**

Планирование ресурсов перед развертыванием — это определение ресурсов, которые необходимо задать до развертывания и которые вступают в силу во время развертывания. Основные моменты включают:

- **Высокая доступность**

GitLab поддерживает развертывание с высокой доступностью. Основные особенности и ограничения этого режима:

- Каждый компонент разворачивается с несколькими репликами
- Сетевой доступ больше не поддерживает NodePort, требуется доступ через доменные имена, настроенные с помощью Ingress
- Метод хранения не поддерживает node storage, требуется доступ через StorageClass или PVC

- **Ресурсы**

Согласно рекомендациям сообщества и практике, экземпляр GitLab без высокой доступности может работать с минимум 3 ядрами CPU и 6Gi памяти, а в режиме высокой доступности требуется минимум 18 ядер CPU и 22Gi памяти для стабильной работы.

- **Хранение**

Для GitLab можно использовать распространённые методы хранения, предоставляемые платформой, такие как классы хранения, persistent volume claims, node storage и др.

- Для хранения GitLab рекомендуется использовать класс хранения TopoLVM
- Node storage не подходит для режима high availability, так как хранит файлы на заданном пути на узле хоста

Кроме того, GitLab поддерживает объектное хранилище. В этом режиме требуется отдельная адаптация метода доступа к хранилищу. Для подробностей обратитесь к поставщику.

- **Сеть**

Платформа предоставляет два основных метода сетевого доступа: NodePort и Ingress.

- NodePort требует указания HTTP и SSH портов и обеспечения их доступности. не подходит для режима высокой доступности
- Ingress требует указания доменного имени и корректной работы разрешения доменного имени

Платформа поддерживает протокол HTTPS, который необходимо настроить после развертывания экземпляра.

- **Redis**

Версия компонента Redis, необходимая GitLab, — v6. Рекомендуется разворачивать экземпляры Redis с помощью Redis Operator, предоставляемого платформой, а затем завершать интеграцию Redis через настройку учётных данных доступа.

Доступ к Redis осуществляется путём настройки ресурса secret с определённым форматом содержимого. Подробнее см. Настройка учётных данных Redis, PostgreSQL и аккаунтов.

- **PostgreSQL**

Версия компонента PostgreSQL, необходимая GitLab, — v14. Рекомендуется разворачивать экземпляры PostgreSQL с помощью PostgreSQL Operator, предоставляемого платформой, а затем завершать интеграцию PostgreSQL через настройку учётных данных доступа.

Доступ к PostgreSQL осуществляется путём настройки ресурса secret с определённым форматом содержимого. Подробнее см. Настройка учётных данных Redis, PostgreSQL и аккаунтов.

Примечание: в режиме высокой доступности компонент Gitaly работает в кластерном режиме и требует дополнительной учётной записи PostgreSQL, настроенной аналогично вышеописанному способу, для доступа к отдельной базе данных. Это могут быть две разные базы данных в рамках одного экземпляра PostgreSQL.

- **Учётные данные аккаунта**

При создании экземпляра GitLab необходимо настроить root-аккаунт и его пароль. Это secret. Подробнее см. Настройка учётных данных Redis, PostgreSQL и аккаунтов.

### ***Планирование конфигурации после развертывания***

Планирование конфигурации после развертывания — это настройки, которые не нужно определять до развертывания, а можно выполнить по мере необходимости с помощью стандартизированных операций после развертывания. К ним относятся единый вход (SSO), настройка HTTPS, конфигурация внешнего балансировщика нагрузки и др.

### ***Развёртывание экземпляра***

GitLab Operator, предоставляемый платформой, предлагает два способа развертывания: развертывание из шаблонов и развертывание из YAML.

Платформа предоставляет два встроенных шаблона: шаблон GitLab Quick Start и шаблон GitLab High Availability, а также поддерживает пользовательские шаблоны для удовлетворения специфических сценариев.

Информация о встроенных шаблонах и развертывании из YAML приведена ниже:

#### **Развертывание из шаблона GitLab Quick Start**

Этот шаблон используется для быстрого создания лёгкого экземпляра GitLab, подходящего для сценариев разработки и тестирования, не рекомендуется для производственных сред.

- Вычислительные ресурсы: 3 ядра CPU, 6Gi памяти
- Метод хранения: используется локальное node storage, требуется указать IP узла хранения и путь

- Сетевой доступ: используется метод NodePort, IP узла совпадает с IP узла хранения, требуется указание портов
- Зависимые сервисы: требуется настройка учётных данных доступа к существующим Redis и PostgreSQL
- Другие настройки: требуется настройка учётных данных аккаунта, функция SSO по умолчанию отключена

Завершите развертывание, заполнив соответствующую информацию согласно подсказкам шаблона.

### **Развертывание из шаблона GitLab High Availability**

Развёртывание экземпляра GitLab с высокой доступностью требует более высокой конфигурации ресурсов и обеспечивает более высокий уровень доступности.

- Вычислительные ресурсы: 18 ядер CPU, 22Gi памяти
- Метод хранения: используется класс хранения для хранения Gitaly — компонента репозитория кода
- Сетевой доступ: используется метод Ingress, требуется указание доменного имени
- Зависимые сервисы: требуется настройка учётных данных доступа к существующим Redis и PostgreSQL, а также отдельные учётные данные для другой независимой базы данных в экземпляре PG для компонента Gitaly
- Другие настройки: требуется настройка учётных данных аккаунта, функция SSO по умолчанию отключена

Для обеспечения высокой доступности GitLab внешние зависимости должны соответствовать следующим условиям:

1. Класс хранения вложений должен поддерживать многозвёздный доступ на чтение и запись (ReadWriteMany)
2. Экземпляры Redis и PostgreSQL должны быть высокодоступными
3. Сетевой балансировщик нагрузки должен быть высокодоступным; при использовании ALB необходимо настроить VIP
4. В кластере должно быть более 2 узлов

Завершите развертывание, заполнив соответствующую информацию согласно подсказкам шаблона.

### **Настройка учётных данных Redis, PostgreSQL и учётных записей**

#### ***Учётные данные Redis***

#### **Формат учетных данных**

Создайте Secret в namespace, где планируется развертывание экземпляра GitLab, выберите типOpaque и добавьте следующие поля в конфигурацию:

| Поле             | Описание                                                                                    | Архитектура         | Пример значения                                       |
|------------------|---------------------------------------------------------------------------------------------|---------------------|-------------------------------------------------------|
| host             | Адрес подключения к Redis. Убедитесь, что сервис GitLab может подключиться к нему.          | standalone          | 192.168.1.1                                           |
| port             | Порт подключения к Redis. Убедитесь, что сервис GitLab может подключиться к этому порту.    | standalone          | 6379                                                  |
| password         | Пароль учетной записи Redis. Требуется при включенной аутентификации Redis.                 | standalone,sentinel | password111                                           |
| address          | Адрес узла Sentinel.                                                                        | sentinel            | 192.168.1.1:26379,192.168.1.2:26379,192.168.1.3:26379 |
| masterName       | Имя группы экземпляров, контролируемой Sentinel в sentinel.conf.                            | sentinel            | mymaster                                              |
| sentinelPassword | Пароль Sentinel для экземпляров Sentinel. Требуется при включенной аутентификации Sentinel. | sentinel            | password111                                           |

**Внимание:** Если присутствуют обе конфигурации — sentinel и standalone, приоритет имеет конфигурация sentinel.

При развертывании с шаблонами высокой доступности, если настроен standalone Redis, ответственность за обеспечение высокой доступности Redis лежит на пользователе.

### Пример для Standalone

```
apiVersion: v1
data:
 host: <base64 encode host>
 password: <base64 encode password>
 port: <base64 encode port>
kind: Secret
metadata:
 name: gitlab-redis
 namespace: <ns-of-gitlab-instance>
type: Opaque
```

### Пример для Sentinel

```
apiVersion: v1
data:
 password: <base64 encode password>
 address: <base64 encode address>
 masterName: <base64 encode masterName>
 sentinelPassword: <base64 encode sentinelPassword>
kind: Secret
metadata:
 name: gitlab-redis
 namespace: <ns-of-gitlab-instance>
type: Opaque
```

### Учётные данные PostgreSQL

#### Формат учётных данных

Создайте Secret в namespace, где планируется развертывание экземпляра GitLab, выберите тип Opaque и добавьте следующие поля в конфигурацию:

| Поле     | Описание                                                                                                                                          | Пример значения |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| host     | Адрес подключения к базе данных. Убедитесь, что сервис GitLab может подключиться к этому адресу базы данных.                                      | 192.168.1.1     |
| port     | Порт подключения к базе данных. Убедитесь, что сервис GitLab может подключиться к этому порту базы данных.                                        | 5432            |
| username | Имя пользователя базы данных                                                                                                                      | gitlab          |
| password | Пароль пользователя базы данных                                                                                                                   | password111     |
| database | Имя базы данных. Эта база должна уже существовать и быть пустой. Вы можете создать ее командой <code>create database &lt;database name&gt;</code> | gitlab_db       |

| Поле    | Описание                                                                                                                                                                                      | Пример значения |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| sslmode | Включение SSL для подключения к базе данных. Доступные опции: <ul style="list-style-type: none"> <li>• enable: Включить SSL</li> <li>• disable: Отключить SSL, подробнее о sslmode</li> </ul> | enable          |

Пример:

```

apiVersion: v1
data:
 database: <base64 encode database name>
 host: <base64 encode host>
 password: <base64 encode password>
 port: <base64 encode port>
 username: <base64 encode username>
 sslmode: <base64 encode sslmode>
kind: Secret
metadata:
 name: gitlab-pg
 namespace: <ns-of-gitlab-instance>
type: Opaque

```

### **Учётные данные учётной записи GitLab**

Создайте Secret в namespace, где планируется развертывание экземпляра GitLab, выберите тип Opaque и добавьте следующие поля в конфигурацию:

| Поле      | Описание                                                                                                                                                                                                  | Пример значения |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| password  | Установите пароль для учетной записи root по умолчанию. Пароль должен содержать буквы, цифры и специальные символы, иметь длину не менее 8 символов, и нельзя использовать распространённые слабые пароли | password111@    |
| namespace | Установите тот же namespace, что и у экземпляра gitlab                                                                                                                                                    | tools           |

Пример:

```

apiVersion: v1
data:
 password: <base64 encode password>
kind: Secret
metadata:
 name: gitlab-root-password

```



namespace: <ns-of-gitlab-instance>  
type: Opaque

## Harbor

### Введение

Harbor — это реестр с открытым исходным кодом, который обеспечивает безопасность артефактов с помощью политик и контроля доступа на основе ролей, гарантирует сканирование образов и отсутствие уязвимостей, а также подписывает образы как доверенные. Как проект с уровнем Graduated в Cloud Native Computing Foundation (CNCF), Harbor версии 2.12.3 обеспечивает соответствие требованиям, производительность и совместимость, помогая вам последовательно хранить, подписывать и сканировать контейнерные образы в облачных нативных средах.

Изначально разработанный компанией VMware, Harbor превратился в комплексное решение для предприятий, стремящихся безопасно управлять контейнерными образами в облачных нативных средах. Harbor предоставляет централизованную платформу для хранения, управления и распространения Docker, Helm, OCI и других контейнерных артефактов, что делает его важным компонентом современных подходов разработки и развертывания контейнеризованных приложений.

### Особенности

#### Особенности развёртывания

- Шаблонные развертывания

Поддержка готовых шаблонов для быстрого развертывания и конфигураций, готовых к эксплуатации, а также пользовательских шаблонов.

- Развертывания с использованием YAML

Настраиваемый метод развертывания с помощью YAML, позволяющий полностью контролировать конфигурацию экземпляра Harbor с использованием значений Helm.

#### Расширенные возможности

- Поддержка сети IPv6

Развертывание и эксплуатация экземпляров Harbor в сетевых средах только с IPv6 или с двойным стеком, обеспечивая совместимость с современными сетевыми инфраструктурами.

- Совместимость с архитектурой ARM

Полная поддержка инфраструктуры на базе ARM, позволяющая развертывать на энергоэффективных и экономичных процессорах ARM для улучшенного соотношения производительности и энергопотребления.

- Усиление безопасности

Устранение критических уязвимостей системы и приложений с помощью проактивного патчинга и усиления безопасности. Включает исправления для проблем,

идентифицированных по CVE, обновления безопасности зависимостей и устранение потенциальных векторов повышения привилегий.

## **Установка оператора**

Перейдите на страницу **Administrator > Marketplace > Operator Hub**, выберите целевой кластер и разверните оператор MWS Build of Harbor.

## **Развертывание экземпляра Harbor**

В этом разделе описывается подписка на Harbor Operator и функциональность, основанная на развертывании экземпляра Harbor.

### ***Пререквизиты***

Этот документ применим к версиям Harbor 2.12 и выше, предоставляемым платформой. Он отделён от платформы и основан на таких технологиях, как Operator.

Убедитесь, что Harbor Operator был развернут (подписан) в целевом кластере, то есть Harbor Operator готов к созданию экземпляров.

### ***Планирование развёртывания***

Harbor поддерживает различные конфигурации ресурсов для удовлетворения различных сценариев использования. В разных сценариях требуемые ресурсы и конфигурации могут значительно отличаться. Поэтому в этом разделе описывается, какие аспекты необходимо учитывать при планировании развертывания перед созданием экземпляров Harbor, а также как решения влияют на последующую работу, чтобы помочь пользователям выполнить конкретное развертывание экземпляров на основе этой информации.

## **Основная информация**

1. Harbor Operator, предоставляемый платформой, основан на официальном сообществе Harbor Operator с корпоративными улучшениями, такими как поддержка ARM и исправления уязвимостей безопасности. Он полностью совместим с версией сообщества по функциональности и улучшает удобство развертывания Harbor за счёт опциональных и настраиваемых шаблонов.
2. Экземпляр Harbor включает несколько компонентов, таких как Registry — компонент, отвечающий за управление файлами образов, PostgreSQL — компонент, обеспечивающий хранение метаданных приложения и информации о пользователях, и Redis — компонент для кэширования и др. Платформа предоставляет профессиональные PostgreSQL Operator и Redis Operator, поэтому при развертывании экземпляров Harbor ресурсы Redis и PostgreSQL не разворачиваются напрямую, а подключаются через настройку соответствующих учётных данных доступа к существующим экземплярам.

## **Планирование ресурсов перед развёртыванием**

Планирование ресурсов перед развёртыванием — это решения, которые необходимо принять до развертывания и которые вступают в силу во время развертывания. Основные аспекты включают:

- **Высокая доступность**

Harbor поддерживает развертывание с высокой доступностью, с основными последствиями и ограничениями:

- Каждый компонент использует несколько реплик
- Сетевой доступ больше не поддерживает NodePort, требуется доступ через доменные данные, настроенные через Ingress
- Методы хранения больше не поддерживают node storage, требуется доступ через Storage Class или PVC

- **Ресурсы**

Согласно рекомендациям сообщества и практике, экземпляр Harbor без высокой доступности может работать с минимум 2 ядрами CPU и 4Gi памяти, а в режиме высокой доступности требуется минимум 8 ядер CPU и 16Gi памяти для стабильной работы.

- **Хранение**

Для Harbor можно использовать распространённые методы хранения, предоставляемые платформой, такие как классы хранения, persistent volume claims, node storage и др.

Node storage не подходит для режима high availability, так как хранит файлы в указанном пути на узле хоста.

Кроме того, Harbor поддерживает объектное хранилище. Для этого требуется отдельная адаптация метода доступа к хранилищу. Пожалуйста, обратитесь к поставщику за подробностями.

- **Сеть**

Платформа предоставляет два основных метода сетевого доступа: NodePort и Ingress

- NodePort требует указания HTTP и SSH портов и обеспечения их доступности  
NodePort не подходит для режима high availability
- Ingress требует указания доменного имени и обеспечения корректного разрешения домена

Платформа поддерживает протокол HTTPS, который необходимо настроить после развертывания экземпляра.

- **Redis**

Текущая версия компонента Redis, от которой зависит Harbor, — v6. Рекомендуется использовать Redis Operator, предоставляемый платформой, для развертывания экземпляров Redis, а затем завершить интеграцию Redis, настроив учётные данные доступа. Доступ к Redis осуществляется путём настройки ресурса secret с определённым форматом содержимого. Подробнее см. Настройка учётных данных доступа к Redis, PostgreSQL и аккаунтам.

Известная проблема: Ошибка внутреннего сервера при изменении разрешений проекта Harbor

- **PostgreSQL**

Текущая версия компонента PostgreSQL, от которой зависит Harbor, — v14. Рекомендуется использовать PostgreSQL Operator, предоставляемый платформой, для развертывания экземпляров PostgreSQL, а затем завершить интеграцию PostgreSQL, настроив учётные данные доступа.

Доступ к PostgreSQL осуществляется путём настройки ресурса secret с определённым форматом содержимого. Подробнее см. Настройка учётных данных доступа к Redis, PostgreSQL и аккаунтам.

- **Учётные данные аккаунта**

При инициализации экземпляра Harbor необходимо настроить учётную запись администратора и её пароль. Это делается путём настройки ресурса secret.

Подробнее см. Настройка учётных данных доступа к Redis, PostgreSQL и аккаунтам.

### ***Планирование конфигурации после развертывания***

Планирование конфигурации после развертывания — это планирование, которое не требует решений до развертывания, но может изменяться по мере необходимости через стандартизированные операции после развертывания. Это включает в себя единую аутентификацию (SSO), настройку HTTPS, конфигурацию внешнего балансировщика нагрузки и др.

### ***Развёртывание***

Harbor Operator, предоставляемый платформой, предлагает два основных метода развертывания: развертывание из шаблонов и развертывание из YAML.

Платформа предоставляет два встроенных шаблона: шаблон Harbor Quick Start и шаблон Harbor High Availability, а также поддерживает пользовательские шаблоны для удовлетворения конкретных сценариев заказчиков.

Информация о встроенных шаблонах и развертывании из YAML приведена ниже.

#### **Развертывание из шаблона Harbor Quick Start**

Этот шаблон используется для быстрого создания лёгкого экземпляра Harbor, подходящего для сценариев разработки и тестирования, не рекомендуется для производственной среды.

- Вычислительные ресурсы: 2 ядра CPU, 4Gi памяти
- Метод хранения: используется локальное node storage, требуется настройка IP узла хранения и пути
- Сетевой доступ: используется метод NodePort, общий IP узла с хранением, требуется указание портов
- Зависимые сервисы: требуется настройка учётных данных доступа к существующим Redis и PostgreSQL
- Другие настройки: требуется настройка учётных данных аккаунта, функция SSO по умолчанию отключена

Завершите развертывание, заполнив соответствующую информацию согласно подсказкам шаблона

## Развёртывание из шаблона Harbor High Availability

Развёртывание экземпляра Harbor с высокой доступностью требует более высокой конфигурации ресурсов и обеспечивает более высокий стандарт доступности.

- Вычислительные ресурсы: 16 ядер CPU, 16 Gi памяти
- Метод хранения: используется класс хранения для хранения файлов образов, журналов фоновых задач и базы данных уязвимостей сканирования образов
- Сетевой доступ: используется метод Ingress, требуется указание доменного имени
- Зависимые сервисы: требуется настройка учётных данных доступа к существующим Redis и PostgreSQL
- Другие настройки: требуется настройка учётных данных аккаунта, функция SSO по умолчанию отключена

Для обеспечения высокой доступности Harbor внешние зависимости должны соответствовать следующим условиям:

1. Экземпляры Redis и PostgreSQL должны быть высокодоступными
2. Балансировщики нагрузки сети должны быть высокодоступными, при использовании ALB необходимо настроить VIP
3. В кластере должно быть более 2 узлов

Завершите развёртывание, заполнив соответствующую информацию согласно подсказкам шаблона.

### Настройка учётных данных для Redis, PostgreSQL и доступа к аккаунту

#### Учётные данные Redis

Harbor предъявляет следующие требования к развёртыванию Redis:

- Рекомендуется использовать Redis версии 7.x.
- Поддерживаются режимы развёртывания Standalone и Sentinel. Режим Redis Cluster не поддерживается.

#### Формат учётных данных

Создайте Secret в пространстве имен, где планируется развёртывание экземпляра Harbor, выберите тип Opaque и добавьте следующие поля в конфигурацию:

| Поле | Описание                                                                                 | Архитектура | Пример значения |
|------|------------------------------------------------------------------------------------------|-------------|-----------------|
| host | Адрес подключения к Redis. Убедитесь, что сервис Harbor может подключиться к нему.       | standalone  | 192.168.1.1     |
| port | Порт подключения к Redis. Убедитесь, что сервис Harbor может подключиться к этому порту. | standalone  | 6379            |

|            |                                                                             |                        |                                                       |
|------------|-----------------------------------------------------------------------------|------------------------|-------------------------------------------------------|
| password   | Пароль учетной записи Redis. Требуется при включенной аутентификации Redis. | standalone<br>sentinel | password111                                           |
| address    | Адрес узла Sentinel.                                                        | sentinel               | 192.168.1.1:26379,192.168.1.2:26379,192.168.1.3:26379 |
| masterName | Имя группы экземпляров, контролируемой Sentinel в sentinel.conf.            | sentinel               | mymaster                                              |

#### Пример для Standalone:

```
apiVersion: v1
data:
 host: <base64 encode host>
 port: <base64 encode port>
 password: <base64 encode password>
kind: Secret
metadata:
 name: harbor-redis
 namespace: <ns-of-harbor-instance>
type: Opaque
```

#### Пример для Sentinel:

```
apiVersion: v1
data:
 password: <base64 encode password>
 address: <base64 encode address>
 masterName: <base64 encode masterName>
kind: Secret
metadata:
 name: harbor-redis
 namespace: <ns-of-harbor-instance>
type: Opaque
```

### Учётные данные PostgreSQL

#### Формат учётных данных

Создайте Secret в пространстве имен, где планируется развертывание экземпляра Harbor, выберите тип Opaque и добавьте следующие поля в конфигурацию:

| Поле | Описание                                                                                         | Пример значения |
|------|--------------------------------------------------------------------------------------------------|-----------------|
| host | Адрес подключения к базе данных. Убедитесь, что сервис Harbor может подключиться к этому адресу. | 192.168.1.1     |

| Поле     | Описание                                                                                                                                                                                                                                                                                           | Пример значения |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| port     | Порт подключения к базе данных. Убедитесь, что сервис Harbor может подключиться к этому порту.                                                                                                                                                                                                     | 5432            |
| username | Имя пользователя базы данных                                                                                                                                                                                                                                                                       | postgres        |
| password | Пароль учетной записи базы данных                                                                                                                                                                                                                                                                  | password111     |
| database | Имя базы данных. Эта база должна уже существовать и быть пустой. Для создания базы используйте команду create database <database name>                                                                                                                                                             | harbor_db       |
| sslmode  | Включение SSL для подключения к базе данных. Доступные опции: <ul style="list-style-type: none"> <li>• require: Требовать SSL-соединение</li> <li>• disable: Отключить SSL</li> <li>• verify-ca: Проверять сертификат сервера</li> <li>• verify-full: Проверять сертификат и имя хоста.</li> </ul> | require         |

#### Пример:

```

apiVersion: v1
data:
 database: <base64 encode database name>
 host: <base64 encode host>
 password: <base64 encode password>
 port: <base64 encode port>
 username: <base64 encode username>
 sslmode: <base64 encode sslmode>
kind: Secret
metadata:
 name: harbor-database
 namespace: <ns-of-harbor-instance>
type: Opaque

```

#### Учётные данные аккаунта Harbor

Создайте Secret в пространстве имен, где планируется развертывание экземпляра Harbor, выберите тип Opaque и добавьте следующие поля в конфигурацию:

| Поле     | Описание                                                                                                                                      | Пример значения |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| password | Установите пароль для учетной записи администратора по умолчанию. Пароль должен содержать буквы, цифры и специальные символы, быть не менее 8 | password111@    |

|           |                                                               |       |
|-----------|---------------------------------------------------------------|-------|
|           | символов и не может быть использован повторно                 |       |
| namespace | Установите то же пространство имен, что и у экземпляра Harbor | tools |

Обратите внимание, что имя пользователя по умолчанию для Harbor — admin.

#### Пример:

```
apiVersion: v1
data:
 password: <base64 encode password>
kind: Secret
metadata:
 name: harbor-admin-password
 namespace: <ns-of-harbor-instance>
type: Opaque
```

## SonarQube

### Введение

SonarQube — это ведущая платформа статического анализа кода, которая помогает разработчикам писать более чистый и безопасный код. В качестве решения с открытым исходным кодом, SonarQube Community Edition предоставляет организациям основные инструменты для обнаружения ошибок, уязвимостей и кодовых запахов в их кодовой базе, позволяя осуществлять непрерывную проверку качества кода на протяжении всего жизненного цикла разработки программного обеспечения.

SonarQube 2025.1 Community Edition — это последняя версия с долгосрочной поддержкой (LTA), предлагающая улучшенную стабильность и множество усовершенствований по сравнению с предыдущими версиями. Эта версия строится на основе основных функций SonarQube и одновременно вводит новые возможности для улучшения управления качеством кода и повышения продуктивности разработчиков.

### Особенности

#### Особенности развёртывания

- Шаблонные развёртывания

Поддержка готовых шаблонов для быстрого развёртывания и конфигураций, готовых к эксплуатации, а также пользовательских шаблонов.

- Развёртывания YAML

Настраиваемый метод развёртывания YAML для полного контроля над конфигурацией экземпляра SonarQube с использованием значений Helm.

#### Улучшения безопасности

Устранение критических уязвимостей систем и приложений с использованием проактивного патчинга и усиления безопасности. Включает исправления для проблем,



идентифицированных CVE, в основных компонентах, обновления безопасности зависимостей и устранение возможных векторов повышения привилегий

## **Установка Operator**

Перейдите на страницу **Administrator > Marketplace > Operator Hub**, переключитесь на целевой кластер и разверните оператор MWS of SonarQube.

## **Развёртывание экземпляра SonarQube**

Этот раздел описывает подписку на SonarQube Operator и функциональность развертывания экземпляров SonarQube.

### ***Планирование развёртывания***

SonarQube поддерживает различные конфигурации ресурсов для удовлетворения различных сценариев клиентов. В разных сценариях требуемые ресурсы, конфигурации и т. д. будут существенно отличаться. Поэтому в этом разделе описано, какие аспекты необходимо учитывать при планировании развертывания перед развертыванием экземпляра SonarQube, а также каковы последствия ключевых решений, чтобы пользователи могли на основе этой информации выполнять последующие конкретные развертывания экземпляров.

### **Основная информация**

SonarQube Operator, предоставляемый платформой, основан на официальном сообществе SonarQube Chart с расширенными корпоративными возможностями, такими как поддержка IPv6 и исправления уязвимостей безопасности. Он полностью совместим с версией сообщества по функциональности и улучшает удобство развертывания SonarQube за счёт опциональных, настраиваемых шаблонов и других методов.

### **Планирование ресурсов перед развёртыванием**

Планирование ресурсов перед развертыванием включает принятие решений до развертывания, которые будут применены в процессе развертывания.

Для получения дополнительных рекомендаций по ресурсам среды обратитесь к официальной документации SonarQube.

### ***Развёртывание***

#### **Развёртывание из шаблона Quickstart Template**

Этот шаблон используется для быстрого создания лёгкого экземпляра SonarQube, подходящего для сценариев разработки и тестирования, не рекомендуется для производственных сред.

- Вычислительные ресурсы: 800m CPU, 4 Gi памяти
- Хранилище: используется локальное хранилище узла, необходимо настроить IP и путь узла хранения
- Сетевой доступ: используется NodePort для доступа к сервису, IP узла совпадает с узлом хранения, указывается порт

## Развёртывание из шаблона Production Template

Этот шаблон используется для быстрого создания производственного экземпляра SonarQube, подходит для производственных сценариев, рекомендуется для производственных сред.

- Вычислительные ресурсы: 8 ядер CPU, 16 Gi памяти
- Хранилище: используется PVC, настраивается класс хранилища
- Сетевой доступ: используется доменное имя для доступа к сервису

## Настройка учётных данных для PostgreSQL и доступа к аккаунту

### Учётные данные PostgreSQL

Создайте Secret, выберите тип Opaque, добавьте и заполните следующие поля в элементе конфигурации:

| Поле          | Описание                                                                                                                                                                                | Обязательно                                                           | Пример значения |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------|
| host          | Адрес подключения к базе данных. Убедитесь, что служба SonarQube может подключиться к этому адресу базы данных.                                                                         | Нет, требуется только при создании экземпляра SonarQube через шаблон. | 192.168.1.1     |
| port          | Порт подключения к базе данных. Убедитесь, что служба SonarQube может подключиться к этому порту базы данных.                                                                           | Нет, требуется только при создании экземпляра SonarQube через шаблон. | 5432            |
| username      | Имя пользователя учетной записи базы данных.                                                                                                                                            | Нет, требуется только при создании экземпляра SonarQube через шаблон. | postgres        |
| jdbc-password | Пароль базы данных.                                                                                                                                                                     | Да                                                                    |                 |
| database      | Имя базы данных. Эта база данных должна уже существовать и быть пустой. Вы можете использовать команду <code>create database &lt;имя базы данных&gt;</code> , чтобы создать базу данных | Нет, требуется только при создании экземпляра SonarQube через шаблон. | sonar_db        |

### Пример:

```
apiVersion: v1
stringData:
 host: 192.168.1.1
```

```
port: 5432
username: postgres
jdbc-password: pg-password
database: sonar_db
kind: Secret
metadata:
 name: sonarqube-pg
type: Opaque
```

### ***Учётные данные учётной записи SonarQube***

Имя пользователя для входа по умолчанию - admin, а пароль должен соответствовать следующим требованиям:

- Не менее 12 символов
- Содержит 1 заглавную букву
- Содержит 1 строчную букву
- Содержит 1 число
- Содержит 1 специальный символ

Создайте Secret, используя тип Opaque, и добавьте поле password в элементы конфигурации:

```
apiVersion: v1
data:
 password: <base64 кодированный пароль>
kind: Secret
metadata:
 name: sonarqube-root-password
type: Opaque
```

## **Nexus**

### **Введение**

Nexus Repository Manager является решением для управления репозиториями, которое позволяет вашей команде эффективно управлять программными компонентами, зависимостями и артефактами на протяжении всего цикла поставки программного обеспечения. Как центральный репозиторий для бинарных компонентов, Nexus Repository Manager предоставляет единый источник правды для всех ваших программных артефактов, улучшая надежность сборок и ускоряя циклы разработки.

Nexus Repository Manager v3.76.0 является последним выпуском общественной версии, предоставляющим надежные возможности управления репозиториями с улучшенными функциями и производительностью. MWS Build of Nexus интегрирует этот мощный инструмент в среды Kubernetes через специализированного оператора, упрощая развертывание и управление.

## Ограничения

Требования к ресурсам: Nexus Repository Manager требует достаточных ресурсов CPU, памяти и хранения, особенно для производственных сред с высокой нагрузкой.

Масштабирование хранилища: по мере роста репозитория увеличиваются и требования к хранилищу. Планируйте достаточную емкость хранилища и подумайте об внедрении политик очистки репозитория.

Соображения по резервному копированию: регулярное резервное копирование как базы данных Nexus, так и бэкап-хранилища имеет важное значение для защиты данных.

## Особенности

### Особенности развёртывания

- Шаблонные развёртывания

Поддерживает готовые шаблоны для быстрого развёртывания и конфигураций, готовых к производству, а также пользовательские шаблоны.

- YAML-развёртывания

Пользовательский метод развёртывания YAML для полного контроля над конфигурацией экземпляра Nexus Repository Manager с использованием значений Helm.

### Установка Operator

Перейдите на страницу **Administrator > Marketplace > Operator Hub**, переключитесь на целевой кластер и разверните оператор MWS Build of Nexus.

### Развёртывание экземпляра Nexus

В этом разделе описывается подписка на Nexus Operator и функциональность для развёртывания экземпляров Nexus.

### Пререквизиты

Этот документ применим к версиям Nexus 3.76 и выше, предоставляемым платформой. Он отделён от платформы и основан на таких технологиях, как Operator.

Пожалуйста, убедитесь, что Nexus Operator был развернут (подписан) в целевом кластере, то есть Nexus Operator готов к созданию экземпляров.

### Планирование развёртывания

Nexus поддерживает различные конфигурации ресурсов для удовлетворения различных сценариев клиентов. В разных сценариях требуемые ресурсы и конфигурации могут значительно отличаться. Поэтому в этом разделе описывается, какие аспекты необходимо учитывать при планировании развёртывания перед созданием экземпляров Nexus, а также каковы последствия принимаемых решений, чтобы помочь пользователям на основе этой информации выполнить последующее конкретное развёртывание экземпляров.

Nexus Operator, предоставляемый платформой, основан на официальном сообществе Nexus Chart с расширенными корпоративными возможностями, такими как исправления уязвимостей безопасности. Он полностью совместим с версией сообщества по

функциональности и улучшает удобство развертывания Nexus за счёт опциональных, настраиваемых шаблонов и других методов с точки зрения пользовательского опыта.

## ***Развёртывание***

### **Развертывание из шаблона Quick Start Template**

Этот шаблон используется для быстрого создания лёгкого экземпляра Nexus, подходящего для сценариев разработки и тестирования, не рекомендуется для производственных сред.

- Вычислительные ресурсы: 2 ядра CPU, 4 Gi памяти
- Хранилище: используется локальное хранилище узла, настройте IP-адрес и путь хранилища узла
- Сетевой доступ: используется NodePort для доступа к сервису, IP-адрес узла совпадает с хранилищем, укажите порт

Завершите развертывание, заполнив соответствующую информацию согласно подсказкам шаблона.

### **Развёртывание из шаблона Production Template**

Этот шаблон используется для быстрого создания производственного экземпляра Nexus, подходит для производственных сценариев, рекомендуется для производственных сред.

- Вычислительные ресурсы: 4 ядра CPU, 8 Gi памяти
- Хранилище: используется PVC-хранилище, настройте класс хранилища
- Сетевой доступ: используется домен для доступа к сервису.

Завершите развертывание, заполнив соответствующую информацию согласно подсказкам шаблона.

## **Настройка учётных данных доступа к аккаунту**

### ***Учётные данные Nexus***

Создайте Secret, выберите тип Opaque и добавьте поле password в элементы конфигурации:

```
apiVersion: v1
data:
 password: <base64 кодированный пароль>
kind: Secret
metadata:
 name: nexus-root-password
type: Opaque
```

# AI

## Введение в MLOps и LLMOps

MLOps (эксплуатация машинного обучения) описывает ряд принципов и инструментов в процессах разработки, обучения, выпуска и эксплуатации и обслуживания машинного обучения, больших моделей или других приложений ИИ, становясь лучшей практикой для эффективного, точного и прослеживаемого выпуска приложений ИИ.

MWS Container Platform AI охватывает не только поддержку инструментов для целевого процесса в MLOps, но также предоставляет более лаконичный пользовательский интерфейс для практики LLMOps (эксплуатации больших языковых моделей). От выпуска сервисов вывода для больших моделей до создания интеллектуальных агентов и приложений — всё это выполняется без программирования.

## Обзор возможностей

### Репозиторий моделей

#### Создание и удаление репозитория моделей

Поддерживает создание репозитория моделей типа 'private', и пользователи могут удалять эти репозитории. Только администраторы могут создавать репозитории моделей типа 'shared', и только администраторы могут их удалять.

#### Управление версиями моделей

Поддерживает контроль версий для моделей, включая создание тегов и веток. Также позволяет делать коммиты для изменения файлов в существующих ветках.

### Сервисы вывода

- Развертывание пользовательского сервиса вывода

Позволяет развертывать любую модель из репозитория моделей в качестве сервиса вывода, настраивая параметры.

- Развертывание сервиса вывода на основе шаблонов

Пользователи могут задавать шаблоны сервисов вывода и использовать их для создания сервисов вывода.

- Динамическое масштабирование сервисов вывода

Поддерживает автоматическое масштабирование реплик в зависимости от объема трафика. Позволяет настраивать без серверное конфигурирование, чтобы уменьшить количество реплик до 0 в периоды бездействия (освобождая ресурсы GPU) и автоматически увеличивать количество при возобновлении трафика.

- Временные окружения для вывода

Предустановлено с общими временными окружениями вывода: vllm-cpu, vllm-gpu, mlserver-cpu, mlserver-gpu и т.д. Также поддерживает пользовательские сторонние временные окружения.

- Понимание вывода

После развертывания пользователи могут испытать сервисы вывода через пользовательский интерфейс. В настоящее время поддерживает понимание для трех категорий: 'Генерация текста', 'Классификация текста' и 'Генерация изображений'.

- Наблюдаемость сервиса вывода

Обеспечивает мониторинг и просмотр журналов, включая:

- Мониторинг ресурсов
- Мониторинг вычислительной мощности
- Мониторинг бизнес-метрик (например, токены).

Пакетные операции для сервисов вывода

Позволяет выполнять массовые операции по управлению несколькими сервисами вывода, включая 'Пакетный запуск', 'Пакетная остановка' и 'Пакетное удаление'.

Этот документ предназначен для того, чтобы помочь новым пользователям быстро понять, как развернуть сервисы вывода в MWS Container Platform AI. Развернув простой сервис вывода "генерация текста" и испытав его, вы сможете быстро овладеть основными функциями и методами использования платформы.

## Процедура

| Шаг | Операция                     | Описание                                                                                                                            | Примечания                                                                                       |
|-----|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| 1   | Создать пространство имен    | Создайте пространство имен в контейнерной платформе и настройте соответствующие роли для MWS Container Platform AI для пользователя | Пропустите этот шаг, если у вас уже есть пространство имен и назначенные разрешения пользователя |
| 2   | Управлять пространством имен | Включите пространство имен в управление MWS Container Platform AI                                                                   | Пропустите этот шаг, если пространство имен уже управляется                                      |
| 3   | Загрузить модель             | Загрузите файл модели в репозиторий модели                                                                                          | Пропустите этот шаг, если вы уже загрузили модель или используете совместимую модель платформы   |
| 4   | Опубликовать сервис вывода   | Опубликуйте модель в качестве онлайн сервиса вывода                                                                                 |                                                                                                  |
| 5   | Вызвать сервис вывода        | Вызовите сервис вывода через API или функцию "Опыт"                                                                                 |                                                                                                  |

### Шаг 1. Создание пространства имён и назначение разрешений пользователю

**Примечание:** Пропустите этот шаг, если у вас уже есть пространство имен и назначенные разрешения пользователю

Пространства имен являются основой изоляции многопользовательской среды в MWS Container Platform AI, и каждый проект должен использовать независимое пространство имен.

1. Войдите в контейнерную платформу как администратор.
2. Перейдите в **Управление проектами**, выберите или создайте проект.



3. На странице деталей проекта нажмите **Пространство имен**.
4. Нажмите **Создать пространство имен** и введите имя (например, "text-classification-demo").
5. Нажмите **Создать**, чтобы завершить создание пространства имен.
6. Назначьте разрешения пространства имен пользователю:
  - Перейдите в Администратор > Пользователи > Пользователи.
  - Создайте пользователя или выберите существующего пользователя, которому необходимо использовать это пространство имен.
  - Нажмите Настроить роли > Добавить роль.
  - Добавьте Роли MWS Container Platform AI и свяжите их с созданным пространством имен и проектом, к которому принадлежит пространство имен.
- ml-namespace-editor: используется разработчиками пространства имен, имеет разрешения на создание, удаление, изменение и запросы моделей и сервисов вывода.
- ml-namespace-owner: используется менеджером пространства имен.
- ml-namespace-viewer: может только просматривать модели, сервисы вывода и другие ресурсы.

## Шаг 2. Управление пространством имен

**Примечание:** Пропустите этот шаг, если пространство имен уже управляется

Включите созданное пространство имен в управление MWS Container Platform AI:

1. Войдите в MWS Container Platform AI, выберите **Администратор** в верхнем навигационном меню и выберите кластер, в котором находится недавно созданное пространство имен в "Кластеры" справа от **Администраторы**.
2. Нажмите Управление пространством имен в левом навигационном меню и нажмите кнопку Управление пространством имен.
3. Выберите недавно созданное пространство имен "text-classification-demo" в диалоговом окне.
4. Нажмите **Управление**, чтобы завершить операцию управления.

## Шаг 3. Загрузка модели

**Примечание:** Пропустите этот шаг, если вы уже загрузили модель или используете совместимую модель платформы

Загрузите модель классификации текста в репозиторий модели:

1. Войдите в MWS Container Platform AI, выберите **Бизнес-представление** в верхнем навигационном меню и выберите управляющее пространство имен из предыдущего шага.

2. Нажмите **Репозиторий моделей** в левом навигационном меню, нажмите **Создать репозиторий моделей** и введите подготовленное имя модели, например "gpt2".
3. После создания перейдите на вкладку **Управление файлами** на странице деталей модели.
4. Нажмите **Импортировать файл модели**, и перетащите или выберите файлы/подпапки модели для загрузки. Если загружаете большую языковую модель, интерфейс может зависать из-за большого размера файла. Рекомендуется использовать команду git push для загрузки больших файлов модели в репозиторий модели.
5. Нажмите кнопку **Импорт**, и дождитесь завершения загрузки.
6. На вкладке **Управление файлами** нажмите **Обновить метаданные** и выберите правильный "Тип задачи" и "Фреймворк" в соответствии с атрибутами большой модели.
  - Тип задачи: это атрибут самой модели, его можно получить, просматривая метку на странице деталей загрузки модели. Разделяется на "Генерация текста", "Генерация изображений" и т. д.
  - Фреймворк: это также атрибут самой модели, его можно получить, просматривая метку на странице деталей загрузки модели. Разделяется на "Transformers", "MLflow" и т. д. Большинство популярных открытых языковых моделей имеют тип "Transformers".

#### Шаг 4. Публикация сервиса вывода

Опубликуйте модель в качестве онлайн-сервиса вывода:

1. На странице деталей модели нажмите **Опубликовать API вывода** > Пользовательская публикация.
2. Настройте параметры сервиса:
  - Имя: gpt2-service
  - Модель: gpt2
  - Версия: Branch-main
  - Исполнение вывода: Необходимо выбрать на основе установленной версии cuda на узле GPU. Например, если установлен драйвер cuda11, выберите "vllm-cuda11.8-x86". Если установлен cuda12, выберите "vllm-cuda12.1-x86".
  - Запросы ресурсов: 1CPU/4Gi Оперативной памяти
  - Ограничения ресурсов: 2CPU/6Gi Оперативной памяти
  - Ускорение GPU: Менеджер GPU
    - GPU vcore: 30
    - GPU vmemory: 32
  - Хранение: Подключить существующий PVC/создать PVC или Временное хранилище/емкость 10Gi

- Авто масштабирование: Выключено
  - Количество экземпляров: 1
3. Нажмите **Опубликовать** и дождитесь, пока сервис не запустится.
  4. Просмотрите статус сервиса на странице **Сервисы вывода**.

## Шаг 5. Вызов сервиса вывода

Проверьте опубликованный сервис вывода:

1. Нажмите **Сервисы вывода** в левом навигационном меню, нажмите на имя "Опубликованный сервис вывода" и нажмите **Опыт** на странице деталей сервиса вывода.
2. Введите тестовый текст, например "Рекомендуйте несколько хороших книг".
3. Просмотрите сгенерированный текст и параметры генерации, возвращенные моделью.

## Установка

### Процедура

В Управление платформой консоли MWS:

1. В верхней части консоли в выпадающем списке **Кластер** выберите кластер назначения, в который хотите установить оператор MWS Container Platform AI Cluster.
2. Щелкните Marketplace / Операторы Hub.
3. Выберите **MWS Container Platform AI Cluster**, затем щелкните Установить.
4. Откроется окно Установка MWS Container Platform AI Cluster.
5. Затем в окне Установка MWS Container Platform AI Cluster.
6. Оставьте **Канал** без изменений.
7. Проверьте, соответствует ли **Версия** версии **MWS Container Platform AI Cluster**, которую вы хотите установить.
8. Оставьте **Место установки** без изменений, по умолчанию должно быть ml-operator.
9. Выберите Ручной для Стратегии обновления.
10. Щелкните **Установить**.

### Проверка

Убедитесь, что плитка **MWS Container Platform AI Cluster** отображает одно из следующих состояний:

- Устанавливается: установка в процессе; дождитесь изменения состояния на Установлено
- Установлено: установка завершена.

### Создание экземпляра кластера

После установки оператора MWS Container Platform AI Cluster вы можете создать экземпляр кластера MWS Container Platform AI.

## Процедура

В Управление платформой консоли MWS:

1. В верхней части консоли в выпадающем списке **Кластер** выберите кластер назначения, в который хотите установить оператор MWS Container Platform AI Cluster.
2. Щелкните Marketplace / Операторы Hub.
3. Выберите MWS Container Platform AI Cluster, затем щелкните Щелкнуть.
4. На странице MWS Container Platform AI Cluster щелкните Все экземпляры на вкладке.
5. Щелкните Создать экземпляр. Откроется окно Выбор типа экземпляра.
6. Найдите плитку **MwsCluster** в окне **Выбор типа экземпляра**, затем щелкните: Создать экземпляр. Появится форма **Создание MwsCluster**.
7. Оставьте default без изменений для **Имя**.
8. Выберите **Развернуть Flavor** из выпадающего списка:
  - a. single-node для обычных развертываний.
  - b. На-cluster для развертываний кластера HA (**Рекомендуется** для производства).
9. Выберите Управляемый для Режим Knative Serving.
10. Выберите Управляемый для Режим KServe.
11. Введите допустимый домен в поле **Домен**.

**Подсказка:** Этот домен используется шлюзом входа для предоставления услуг обслуживания моделей. Скорее всего, вы захотите использовать имя с подстановочным знаком, например \*.example.com.

Вы можете указать следующие типы сертификатов, обновив поле Тип сертификата домена:

- Предоставленный
- Самоподписанный
- MWSDefaultIngress

По умолчанию конфигурация использует тип сертификата **Самоподписанный для защиты** входящего трафика в ваш кластер. Сертификат хранится в секрете knative-serving- cert , который указан в поле **Секрет сертификата домена**

Чтобы использовать сертификат, предоставленный вами, храните секрет сертификата в пространстве имен istio-system , затем обновите значение поля Секрет сертификата домена и измените значение поля **Секрет сертификата домена** на Предоставленный .

12. В разделе **Gitlab**:
  - a. Введите URL вашего self-hosted Gitlab в поле **Base URL**.
  - b. Введите cpaas-system в поле **Admin Token Secret Namespace**.
  - c. Введите ml-gitlab-admin-token в поле **Admin Token Secret Name**.
13. В разделе **MySQL**:
  - a. Введите адрес MySQL-инстанса в поле Host.

- b. Введите порт MySQL-инстанса в поле Port (по умолчанию: 3306).
  - c. Введите пользователя для подключения к MySQL в поле Username.
  - d. Введите базу данных, которая будет использоваться для MWS Container Platform AI, в поле Database (по умолчанию: ml).
  - e. Выберите пространство имен секрета с паролем для пользователя в поле Password Secret Namespace.
  - f. Введите имя секрета с паролем для пользователя в поле Password Secret Name.
14. Проверьте настройки и нажмите **Create**.

### Проверка

Проверьте поле статуса ресурса MwsCluster с именем default:

```
kubectl get mlcluster default
```

Должно вернуть Ready:

|      |    |        |
|------|----|--------|
| NA   | RE | REASO  |
| ME   | AD | N      |
|      | Y  |        |
| defa | Tr | Succee |
| ult  | ue | ded    |

### Управление пространствами имен

Тенанты, поддерживаемые MWS ML, представляют собой пространства имен K8s, которые необходимо управлять в **Управлении пространствами имен** перед их использованием как тенанты MWS ML.

Процедура создания пространства имен и добавления пользователей в это пространство имен выглядит следующим образом:

1. Войдите в платформу MWS как администратор
2. Создайте пространство имен
  - a. Перейдите в **Управление проектами MWS**
  - b. Создайте новый проект или выберите существующий, затем нажмите, чтобы перейти на страницу сведений о текущем проекте.
  - c. Нажмите **Пространство имен > Пространство имен**
  - d. Нажмите **Создать пространство имен**
  - e. На странице **Создание пространства имен**, в разделе **Политики безопасности подов**, выберите **привилегированный** в качестве стандарта безопасности для всех трех режимов безопасности enforce, audit и warn.

**Внимание:** Стандарт безопасности должен быть либо пустым, либо привилегированным, в противном случае рабочая нагрузка AI Inference может отказаться от выполнения

3. Назначьте права на пространство имен пользователям
  - a. Перейдите в **Управление платформой MWS**

- b. Выберите **Управление ролями пользователей** > **Управление пользователями** из навигации
  - c. Создайте нового пользователя или выберите пользователя, которого нужно добавить в это пространство имен
  - d. Перейдите на вкладку **Настроить роль** и нажмите кнопку **Добавить роль**
  - e. Настройте роли, связанные с MWS ML, для пользователя и выберите ранее созданное пространство имен
  - f. Редактор: Пользователи могут создавать и удалять рабочие нагрузки в пространстве имен и использовать ресурсные квоты, такие как CPU и GPU, для выполнения работы
  - g. Владелец: Пользователи имеют право управлять этим пространством имен MWS ML
  - h. Зритель: Пользователи могут только просматривать ресурсы пространства имен и не могут создавать или удалять рабочие нагрузки
4. Вернитесь в MWS ML, нажмите **Управление пространством имен** в **Управлении платформой** > **Управлении пространствами имен** и выберите ранее созданное пространство имен для завершения управления

## Развёртывание модели и вывод в эксплуатацию

### Управление моделями

Репозиторий моделей — это способ хранения и управления, специально разработанный для моделей LLM на платформе MWS Container Platform AI. Основанный на технологии Git LFS, он предоставляет эффективное и надежное решение для хранения моделей, охватывающее критические этапы, такие как развёртывание службы вывода, настройка обучения модели и оценка модели. С помощью Репозитория моделей пользователи могут легко управлять версиями моделей, делиться активами модели и осуществлять управление жизненным циклом моделей.

### Основные функции

#### Создание и удаление репозитория моделей

- Создайте пустой репозиторий с поддержкой Git с метаданными (название / описание / доступность).
- Удалите модели после проверки зависимостей (например, убедитесь, что нет активных служб вывода).

#### Управление файлами

- Загрузка через веб-интерфейс
  - Перетаскивание файлов / папок (ограничено для небольших и средних размеров).
- CLI/Git LFS
  - Использование git lfs track для больших файлов (например, \*.bin, \*.h5).

- Пример:

```
git clone <model_repo_url>
```

```
git lfs install
```

```
cp ~/local_model/* . && git add . && git commit -m "v1.0" && git push
```

## Контроль версий

- Ветвление
  - Поддержка параллельных версий (например, ветки experimental и main).
- Тегирование
  - Пометка релизов через UI/CLI (например, git tag -a v2.0 -m "Stable release").
- Синхронизация метаданных
  - Автоматическое чтение README.md из ветки по умолчанию для описания модели.

## Совместное использование между арендаторами

- Общие модели
  - Установка видимости "Shared" при создании для доступа между арендаторами.
- Публичный маркетплейс
  - Использование пространства имён public для публикации open-source моделей (например, конверсии HuggingFace).

## Интеграция с MLOps

- Готовность к развертыванию
  - Запуск сервиса инференса в один клик из тегированных версий моделей.
- Интеграция с Notebook
  - Прямое получение моделей в MWS ML Notebooks для тестирования:

```
!git clone https://ml-public/resnet50.git
```

## Сервис вывода

- Прямое развертывание модели для сервисов вывода
  - Позволяет пользователям напрямую выбирать конкретные версии модели из репозитория и указывать образ времени выполнения вывода для быстрого развертывания онлайн-сервиса. Система автоматически загружает, кэширует и загружает модель, а затем запускает сервис вывода. Это упрощает процесс развертывания и снижает операционную сложность.
- Развертывание пользовательского образа для сервисов вывода
  - Поддерживает пользователей в написании Dockerfile для упаковки моделей и их зависимостей в пользовательские образы, а затем развертывание сервисов вывода через стандартные развертывания Kubernetes. Этот подход предоставляет большую гибкость, позволяя пользователям настраивать среду вывода в соответствии с их потребностями.

- Опыт работы с сервисом вывода
  - Поддерживает пакетные операции на нескольких сервисах вывода, такие как пакетный запуск, остановка, обновление и удаление.
  - Способен поддерживать создание, мониторинг и экспорт результатов пакетных задач вывода.
  - Предоставляет управление ресурсами пакетами, которое может выделять и настраивать ресурсы сервисов вывода пакетами.

### Служба вывода

Служба вывода — это ключевая функция платформы MWS Container Platform AI, предназначенная для эффективного развертывания моделей LLM в качестве онлайн-служб вывода, поддерживающая различные методы вызова, такие как HTTP API и gRPC. С помощью Службы вывода пользователи могут быстро создавать приложения LLM и предоставлять внешние стабильные и высокопроизводительные возможности LLM.

### Создание службы вывода

1. Выберите пользовательскую публикацию

**Подсказка:** Публикация пользовательской службы вывода требует ручной настройки параметров. Вы также можете создать «шаблон», комбинируя входные параметры для быстрого развертывания служб вывода.

2. Укажите детали службы вывода для публикации модели:

| Параметры                | Описание                                                                                        |
|--------------------------|-------------------------------------------------------------------------------------------------|
| Имя                      | Обязательно, имя API вывода.                                                                    |
| Описание                 | Подробное описание API вывода, объясняющее его функциональность и назначение.                   |
| Модель                   | Обязательно, имя модели, используемой для вывода.                                               |
| Версия                   | Обязательно, версия модели. варианты включают ветку и тег.                                      |
| Временные размеры вывода | Обязательно, движок, используемый для времени выполнения вывода                                 |
| Запрашиваемый CPU        | Обязательно, количество запрашиваемых ресурсов CPU для службы вывода.                           |
| Запрашиваемая память     | Обязательно, количество запрашиваемых ресурсов памяти для службы вывода.                        |
| Ограничения CPU          | Обязательно, максимальное количество ресурсов CPU, которые может использовать служба вывода.    |
| Ограничения памяти       | Обязательно, максимальное количество ресурсов памяти, которые может использовать служба вывода. |
| Тип ускорения GPU        | Тип ускорения GPU.                                                                              |
| Значение ускорения GPU   | Значение ускорения GPU.                                                                         |
| Временное хранилище      | Временное хранилище, используемое службой вывода.                                               |



| Параметры                   | Описание                                                                                                                           |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Подключите существующий PVC | Подключите существующий запрос на постоянный объем Kubernetes (PVC) в качестве хранилища.                                          |
| Вместимость                 | Обязательно, размер емкости временного хранилища или PVC.                                                                          |
| Авто масштабирование        | Включите или отключите функциональность автоматического масштабирования.                                                           |
| Количество экземпляров      | Обязательно, количество экземпляров, запущенных службой вывода.                                                                    |
| Переменные среды            | Пары ключ-значение, внедренные в среду выполнения контейнера.                                                                      |
| Добавить параметры          | Параметры, передаваемые исполняемому файлу контейнера. Массив строк (например, ["-port=8080", "--batch_size=4"]).                  |
| Команда запуска             | Переопределяет инструкцию ENTRYPOINT по умолчанию в контейнере.<br>Исполняемый файл + аргументы (например, ["python", "serve.py"]) |

3. Нажмите **Опубликовать**, чтобы создать службу вывода.

## Мониторинг и логирование

### Мониторинг

#### Мониторинг ресурсов

- Использование ЦП/ОЗУ
- Отслеживайте метрики использования ЦП и ОЗУ для служб вывода, чтобы оптимизировать распределение ресурсов и предотвратить узкие места.

#### Доступ к мониторингу ресурсов

1. Перейдите в **Сервисы вывода** в левом навигационном меню.
2. Щелкните по имени целевого сервиса вывода, чтобы открыть его страницу деталей.
3. Выберите вкладку **Монитор**.
4. Убедитесь, что раздел **Монитор ресурсов** развернут (по умолчанию).
5. Используйте выбор времени в правом верхнем углу, чтобы выбрать предопределённый или настраиваемый диапазон:

| Предустановленные опции | Настраиваемый диапазон    |
|-------------------------|---------------------------|
| Последние 30 минут      | Дата и время начала/конца |
| Последний час           |                           |
| Последние 6 часов       |                           |
| Последние 24 часа       |                           |
| Последние 2 дня         |                           |

| Предустановленные опции | Настраиваемый диапазон |
|-------------------------|------------------------|
| Последние 7 дней        |                        |

### **Мониторинг вычислений**

Метрики GPU и VRAM

Мониторьте использование вычислений GPU и потребление видеопамати (VRAM) для обеспечения эффективного использования аппаратного обеспечения для ускоренных рабочих нагрузок.

### **Дополнительный мониторинг**

- Пропускная способность токенов

Измеряйте скорость обработки токенов для оценки производительности и масштабируемости модели.

- Аналитика трафика запросов

Анализируйте объем запросов, задержку и отслеживайте успешные/неудачные запросы в секунду (QPS) для поддержания надежности сервиса и выполнения SLAs.

### **Логирование**

Модуль **Логирование** в наборах инструментов мониторинга и операций MWS Container Platform AI является решением для журналирования в реальном времени, предназначенным для сервисов вывода в рабочих процессах MLOps/LLMOps/GenOps. Он обеспечивает мгновенную видимость операционного состояния реплик подов, поддерживающих ваши AI-сервисы, позволяя эффективную отладку и наблюдаемость. При потоковой передаче журналов контейнеров с задержкой на уровне миллисекунд и предоставлении встроенных инструментов анализа, он помогает пользователям поддерживать здоровое состояние сервиса, одновременно ускоряя реагирование на инциденты.

### **Основные возможности**

- **Потоковая передача журналов в реальном времени:** Автоматически отображает журналы по мере их генерации выбранным подом.
- **Выбор пода-реплики:** Переключение между подами для просмотра журналов от конкретных реплик.
- **Поиск по ключевым словам:** Найдите конкретные записи журнала, используя точное или частичное соответствие ключевым словам.
- **Экспорт журнала:** Скачайте журналы в формате .txt для офлайн-анализа или архивации.

### **Преимущества функциональности**

- **Мгновенная видимость:** Не требуется ручное обновление – журналы обновляются динамически.

- **Поддержка нескольких подов:** Легко переключайтесь между подами для сравнения журналов по репликам.
- **Операционная эффективность:** Встроенные инструменты поиска и экспорта ускоряют анализ первопричин.

### ***Просмотр журналов***

1. Перейдите в **Сервисы-вывода** в левой панели навигации.
2. Нажмите на имя целевого сервиса-вывода, чтобы открыть его страницу с деталями.
3. Выберите вкладку **Ведение журналов** в панели вкладок под именем сервиса-вывода.
4. Используйте выпадающий список **Реплика**, чтобы выбрать под. Страница журналов автоматически отобразит и начнет потоковую передачу журналов из выбранного пода.

### ***Поиск по ключевым словам***

1. Нажмите кнопку **Найти** в правом верхнем углу страницы журналов.
2. Введите ключевое слово или фразу в поле поиска.

Совпадающие записи выделены желтым цветом.

Используйте кнопки вверх и вниз чтобы перейти между совпадениями.

### ***Экспорт журналов***

Чтобы экспортировать журналы для офлайн-хранения и анализа, нажмите кнопку **Экспорт** в правом верхнем углу страницы журналов.

Браузер автоматически скачает файл журнала на локальный компьютер.

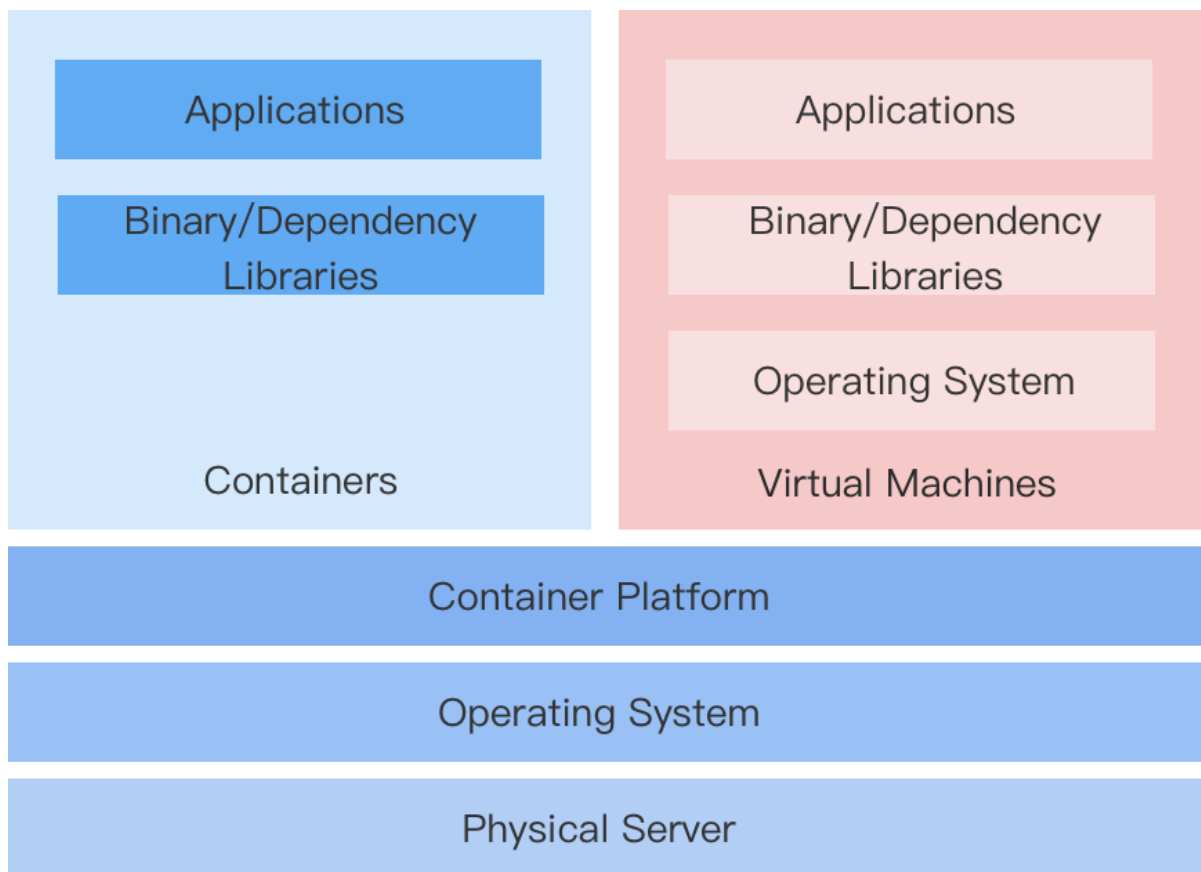
## **Виртуализация**

Для предприятий, использующих архитектуру на основе виртуальных машин, переход к архитектуре на основе Kubernetes и контейнеров неизбежно требует модернизации приложений. Однако из-за таких ограничений, как необходимость непрерывной работы бизнеса или сложность изменения привычек разработки, предприятия часто не могут полностью отказаться от архитектуры виртуализации за короткий срок.

Поэтому решение, которое может единообразно настраивать, управлять и контролировать ресурсы контейнеров и виртуальных машин на одной платформе, становится особенно важным.

### **Запуск виртуальных машин и контейнеров в одном окружении**

Данная платформа реализует решение виртуальной машины (VMI, VirtualMachineInstance) на основе открытого компонента KubeVirt, что позволяет проще и быстрее создавать виртуальные машины с оркестрацией контейнеров и запускать виртуализированные приложения.



## Особенности

### Быстрая трансформация

Нет необходимости переписывать приложения или изменять образы. Достаточно упаковать существующее приложение в образ виртуальной машины формата `qcow2` или `raw` и создать виртуальную машину на платформе с использованием этого образа, что позволяет развернуть приложение на контейнерной платформе.

### Сохранение привычек работы

Контейнеризированные виртуальные машины можно управлять аналогично традиционным виртуальным машинам, не уделяя внимания внутренней реализации контейнеров, включая управление жизненным циклом виртуальной машины, дисками и сетями, а также управление снимками.

### Сосуществование виртуализации и контейнеризации

Унифицированная платформа поддерживает управление виртуализированными сервисами и одновременно обеспечивает планирование и управление контейнерами на базе Kubernetes.

- При сохранении использования рабочих нагрузок виртуальных машин позволяет постепенно модернизировать контейнеризированные приложения.
- Разработка новых контейнеризированных приложений, которые должны взаимодействовать с виртуализированными приложениями, не затруднена.

## Функциональные возможности

- **Виртуальная машина:** Поддержка создания виртуальных машин с образами, выделенными администраторами, и управление ими, включая запуск и остановку виртуальных машин, управление снимками, удалённый вход в виртуальные машины и изменение конфигураций виртуальных машин.
- **Виртуальный диск:** Поддержка просмотра и управления информацией о дисках, созданных в текущем проекте, включая создание дисков, просмотр имён дисков, классов хранилища, ёмкостей и связанных виртуальных машин.
- **Снимки виртуальных машин:** Поддержка просмотра деталей, таких как статус снимков виртуальных машин, связанная виртуальная машина и время последнего отката.
- **Образы виртуальных машин:** Поддержка просмотра информации об образах виртуальных машин в текущем проекте, включая способ предоставления образа и операционную систему.
- **Ключевые пары:** Поддержка просмотра и управления ключевыми парами, созданными в текущем проекте, включая создание ключевых пар и просмотр списка связанных виртуальных машин.

## Ограничения и условия

Реализация должна базироваться на кластере физических машин, при этом компоненты KubeVirt должны быть развернуты внутри кластера с включённой виртуализацией.

Платформа предоставляет возможность развертывания компонентов KubeVirt через Operator и интерфейс для включения виртуализации, при этом все связанные настройки выполняет администратор платформы.

## Установка

Для того чтобы сотрудники проекта могли полноценно использовать функции виртуализации в контейнерной платформе, администратору платформы необходимо выполнить следующие операции для подготовки среды виртуализации.

### Шаги

#### *Включение виртуализации на узлах*

Если узлы собственного кластера — это **физические машины**, вы можете управлять разрешением на планирование Virtual Machine Instances (VMI) Kubernetes на этом узле, включая или отключая переключатель виртуализации узла.

- При включении новые виртуальные машины могут планироваться на узле физической машины; физические узлы Windows не поддерживают включение виртуализации.
- При отключении новые виртуальные машины не будут планироваться на узле физической машины, но это не влияет на уже запущенные виртуальные машины на этом узле.

#### Порядок действий

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Cluster Management > Clusters**.
3. Нажмите на **Self-Built Cluster Name**.

4. На вкладке **Nodes** нажмите : справа от узла, на котором хотите установить переключатель **виртуализации** > **Enable Virtualization**.
5. Нажмите **Confirm**.

#### **Развёртывание оператора**

1. Войдите в систему и перейдите на страницу **Platform Management**.
2. Нажмите **Marketplace** > **OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите **MWS-CP Virtualization with KubeVirt**, нажмите **Install** и перейдите на страницу **Install MWS-P Virtualization with KubeVirt**.

Параметры конфигурации:

| Параметр                  | Рекомендуемая конфигурация                                                                                                                |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Channel</b>            | Канал по умолчанию — alpha.                                                                                                               |
| <b>Installation Mode</b>  | Cluster : Все пространства имён кластера используют один экземпляр оператора для создания и управления, что снижает потребление ресурсов. |
| <b>Installation Place</b> | Выберите Recommended , пространство имён поддерживается только kubevirt.                                                                  |
| <b>Upgrade Strategy</b>   | Manual: При появлении новой версии в Operator Hub требуется ручное подтверждение для обновления оператора до последней версии.            |

#### **Создание экземпляра HyperConverged**

1. Перейдите в **Platform Management**.
2. Нажмите **Marketplace** > **OperatorHub**.
3. Найдите **MWS-Virtualization with KubeVirt**, нажмите на него, чтобы перейти на страницу с подробной информацией.
4. Нажмите **All Instances**.
5. На карточке экземпляра **HyperConverged** нажмите **Create Instance**.

**Примечание:** В каждом кластере необходимо создать только один экземпляр HyperConverged.

6. Переключитесь в режим просмотра YAML и замените только **placeholder** в поле spec.storageImport.insecureRegistries в примере на правильный **адрес репозитория образов виртуальных машин**, например: 192.168.16.214:60080, остальные параметры оставьте по умолчанию.

спец:

storageImport:

insecureRegistries:

- placeholder

Результат замены:

список:

storageImport:

insecureRegistries:

- "192.168.16.214:60080"

7. Нажмите **Create** и дождитесь автоматического создания экземпляров типов CDI и KubeVirt в списке ресурсов, при этом убедитесь, что в YAML поле **status.phase** отображается как `deployed`, что означает успешное создание экземпляра HyperConverged.

### Настройка коэффициента **overcommit** виртуальных машин (опционально)

- Настройка коэффициента **overcommit** для кластера, в котором находятся виртуальные машины, в разделе **Cluster Management > Clusters**.
- Или настройка коэффициента **overcommit** для пространства имён, в котором расположены виртуальные машины, в разделе **Project Management > Namespaces**

### Важные замечания

- Виртуальные машины поддерживают только коэффициент **overcommit** по CPU, рекомендуемое значение — от 2 до 4.
- После включения коэффициента **overcommit** для виртуальных машин при создании виртуальной машины значение запроса контейнера (`requests`) фиксируется как **указанное значение лимита (limits) / коэффициент **overcommit** виртуальной машины**, что делает пользовательские настройки запроса через YAML неэффективными.

Например: если коэффициент **overcommit** CPU для виртуальной машины установлен в 4, а пользователь при создании виртуальной машины указывает лимит CPU равный 4с, то значение запроса CPU будет  $4с/4 = 1с$ .

### Описание квоты ресурсов

Квота ресурсов памяти для виртуальных машин ограничивается квотой памяти пространства имён, в котором они находятся. Поскольку память Pod, в котором размещена виртуальная машина, обычно больше фактически доступной памяти виртуальной машины, рекомендуется резервировать 20% ресурсов. Если оставшиеся доступные ресурсы в пространстве имён опускаются ниже 20%, необходимо своевременно масштабировать ресурсы.

### Добавление образов виртуальных машин

Платформа поддерживает добавление образов виртуальных машин архитектур X86\_64 и ARM64 (Alpha), что позволяет разработчикам быстро создавать виртуальные машины для существующих сервисов и облегчает миграцию бизнес-систем.

### Шаги

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Virtualization Management > Virtual Machine Images**.
3. Нажмите **Add Virtual Machine Image**.

4. Следуйте инструкциям ниже для настройки соответствующих параметров.

| Параметр                   | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Provisioning Method</b> | В настоящее время поддерживается только метод <b>Public Image</b> , то есть добавленный образ может использоваться в назначенных проектах.                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Operating System</b>    | Поддерживаемые операционные системы: <b>CentOS/Ubuntu/RedHat/Debian/TLinux/Other Linux/Windows (Alpha)</b> .<br>Поддерживаемые архитектуры систем: <b>X86_64</b> и <b>ARM64 (Alpha)</b> .                                                                                                                                                                                                                                                                                                                                           |
| <b>Source</b>              | <b>Image Repository</b> : образы виртуальных машин, хранящиеся в репозитории контейнерных образов.<br><b>HTTP</b> : образы виртуальных машин, хранящиеся на файловом сервере с использованием протокола HTTP.<br><b>Object Storage (S3)</b> : образы виртуальных машин, доступные через протокол Object Storage (S3). Если аутентификация не требуется, рекомендуется использовать HTTP в качестве источника.                                                                                                                       |
| <b>CPU Architecture</b>    | Укажите архитектуру CPU. Для источников из репозитория образов поддерживается множественный выбор; для остальных источников — только одиночный выбор.                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Image Address</b>       | Поддерживаются образы виртуальных машин KVM, включая форматы qcow2/raw.<br>Если источник — репозиторий образов, введите repository_address:image_version, например, index.docker.io/library/ubuntu:latest.<br>Если источник — HTTP, введите URL файла образа, который должен начинаться с http:// или https://, например, http://192.168.0.1/vm_image/centos_7.8.qcow2.<br>Если источник — Object Storage (S3), введите адрес образа, доступный через протокол Object Storage (S3), например, https://endpoint/bucket/centos.qcow2. |
| <b>Authentication</b>      | В зависимости от необходимости аутентификации в репозитории образов, переключатель можно включить или выключить. Если включено, можно выбрать из существующих учетных данных образа или нажать <b>Add Credentials</b> , поддерживаются только учетные данные типа <b>Username/Password</b> .<br><b>Примечание</b> : при источнике Object Storage (S3) отключить аутентификацию нельзя.                                                                                                                                              |
| <b>Assigned Project</b>    | Назначьте права использования этого образа проектам.<br>All Projects: назначить права использования образа всем проектам.<br>Specific Project: назначить права использования образа конкретному проекту.                                                                                                                                                                                                                                                                                                                            |



| Параметр | Описание                                                                                                                       |
|----------|--------------------------------------------------------------------------------------------------------------------------------|
|          | No Assignment: пока не назначать ни одному проекту. После создания образа можно назначить через операцию <b>Update Image</b> . |

5. Нажмите **Add**.

## Обновление/удаление образов виртуальных машин

1. Перейдите в **Platform Management**.
2. В левой боковой панели нажмите **Virtualization Management > Virtual Machine Images**.
3. Нажмите : > **Update/Delete**.
4. После подтверждения нажмите **Update/Delete**.

## Обновление/удаление учётных данных образа

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Virtualization Management > Virtual Machine Images**.
3. На вкладке **Image Credentials** нажмите : > **Update/Delete**.
4. После подтверждения нажмите **Update/Delete**.

## Создание образов Windows на основе ISO с использованием KubeVirt

В этом разделе рассматривается решение виртуальной машины на базе открытого компонента KubeVirt, использующее технологию виртуализации KubeVirt для создания образа операционной системы Windows через ISO-образ.

### Ограничения и особенности

- При запуске KubeVirt размер файловой системы пользовательского образа влияет на скорость записи образа на диск в PVC. Если файловая система слишком большая, это может привести к увеличению времени создания.
- Рекомендуется держать корневой раздел Linux или диск с Windows менее 100 ГБ для минимизации начального размера. Последующее расширение можно выполнить через cloud-init (для Windows систем расширение необходимо делать вручную после создания).

### Шаги

Создайте Docker-образ из подготовленных ISO-образов Windows и virtio-win и отправьте его в репозиторий. В данном документе в качестве примера используется Windows Server 2019.

### Создание Docker-образа из ISO Windows

1. Перейдите в каталог, где хранится ISO-образ, и выполните в терминале команду для переименования ISO-образа в win.iso.  
mv <ISO image name> win.iso # Замените <ISO image name> на фактическое имя образа, например, mv en\_windows\_server\_2019\_x64\_dvd\_4cb967d8.iso win.iso
2. Выполните команду для создания Dockerfile.  
touch Dockerfile
3. Отредактируйте Dockerfile, добавьте следующий контент и сохраните файл.

FROM scratch

ADD --chown=107:107 win.iso /disk/

4. Выполните команду для сборки Docker-образа.

docker build -t build-harbor.example.com/3rdparty/vmdisks/winiso:2019 . # Замените репозиторий в соответствии с вашей средой

5. Выполните команду для отправки образа в репозиторий.

docker push build-harbor.example.com/3rdparty/vmdisks/winiso:2019 # Замените репозиторий в соответствии с вашей средой

### Создание Docker-образа из ISO virtio-win

1. Перейдите в каталог, где хранится ISO-образ, и выполните команду для создания Dockerfile.

touch Dockerfile

2. Отредактируйте Dockerfile, добавьте следующий контент и сохраните файл.

FROM scratch

ADD --chown=107:107 virtio-win.iso /disk/

3. Выполните команду для сборки Docker-образа.

docker build -t build-harbor.example.com/3rdparty/vmdisks/win-virtio:latest . #  
Замените репозиторий в соответствии с вашей средой

4. Выполните команду для отправки образа в репозиторий.

docker push build-harbor.example.com/3rdparty/vmdisks/win-virtio:latest # Замените репозиторий в соответствии с вашей средой

### Создание виртуальной машины

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Virtualization > Virtual Machines**.
3. Нажмите Create Virtual Machine.
4. Заполните необходимые параметры, такие как Name, Image и др. Для подробных параметров и конфигурации смотрите раздел о создании виртуальной машины.
5. Переключитесь в режим YAML.
6. Замените конфигурацию в поле spec.template.spec.domain.devices.disks следующим содержимым.

domain:

devices:

disks:

- disk:

bus: virtio

name: cloudinitdisk

- bootOrder: 1

cdrom:

bus: sata

name: containerdisk

- cdrom:

```
 bus: sata
 name: virtio
 - disk:
 bus: sata
 name: rootfs
 bootOrder: 10
```

7. Добавьте следующее содержимое в поле `spec.template.spec.volumes`.

```
- containerDisk:
 image: registry.example.com:60070/3rdparty/vmdisks/winiso:2019 # Замените
образ в соответствии с вашей средой
 name: containerdisk
- containerDisk:
 image: registry.example.com:60070/3rdparty/vmdisks/win-virtio # Замените
образ в соответствии с вашей средой
 name: virtio
```

8. Проверьте YAML-файл. Полный YAML после завершения конфигурации выглядит следующим образом.

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
 annotations:
 cpaas.io/creator: test@example.io
 cpaas.io/display-name: ""
 cpaas.io/updated-at: 2024-09-01T14:57:55Z
 kubevirt.io/latest-observed-api-version: v1
 kubevirt.io/storage-observed-api-version: v1
 generation: 16
 labels:
 virtualization.cpaas.io/image-name: debian-2120-x86
 virtualization.cpaas.io/image-os-arch: amd64
 virtualization.cpaas.io/image-os-type: debian
 virtualization.cpaas.io/image-supply-by: public
 vm.cpaas.io/name: aa-test
 name: aa-test
 namespace: cp-service-self
spec:
 dataVolumeTemplates:
 - metadata:
 creationTimestamp: null
 labels:
 vm.cpaas.io/reclaim-policy: Delete
```

```
 vm.cpaas.io/used-by: aa-test
 name: aa-test-rootfs
spec:
 pvc:
 accessModes:
 - ReadWriteOnce
 resources:
 requests:
 storage: 100Gi
 storageClassName: vm-cephrbd
 volumeMode: Block
 source:
 http:
 url: http://192.168.254.12/kube-debian-12.2.0-x86-out.qcow2
running: true
template:
 metadata:
 annotations:
 cpaas.io/creator: test@example.io
 cpaas.io/display-name: ""
 cpaas.io/updated-at: 2024-09-01T14:55:44Z
 kubevirt.io/latest-observed-api-version: v1
 kubevirt.io/storage-observed-api-version: v1
 creationTimestamp: null
 labels:
 virtualization.cpaas.io/image-name: debian-2120-x86
 virtualization.cpaas.io/image-os-arch: amd64
 virtualization.cpaas.io/image-os-type: debian
 virtualization.cpaas.io/image-supply-by: public
 vm.cpaas.io/name: aa-test
spec:
 affinity:
 nodeAffinity: {}
 architecture: amd64
 domain:
 devices:
 disks:
 - disk:
 bus: virtio
 name: cloudinitdisk
```

- bootOrder: 1
  - cdrom:
    - bus: sata
    - name: containerdisk
- cdrom:
  - bus: sata
  - name: virtio
- disk:
  - bus: sata
  - name: rootfs
  - bootOrder: 10
- interfaces:
  - bridge: {}
    - name: default
- machine:
  - type: q35
- resources:
  - limits:
    - cpu: "4"
    - memory: 8Gi
  - requests:
    - cpu: "4"
    - memory: 8Gi
- networks:
  - name: default
    - pod: {}
- nodeSelector:
  - kubernetes.io/arch: amd64
  - vm.cpaas.io/baremetal: "true"
- volumes:
  - cloudInitConfigDrive:
    - userData: >-
      - #cloud-config
      - disable\_root: false
      - ssh\_pwauth: true
    - users:
      - default
      - name: root
    - lock\_passwd: false

```
 hashed_passwd:
$6$0vhl57e$0rawYwaeu9jL6hBf3XP9Ik6XXaMUS9/W6LPbWRinUoXujo39IP3I98VOcOOB
tr.LDoAv/ylm85FLQmxwNIWFe/
 name: cloudinitdisk
 - containerDisk:
 image: registry.example.com:60070/3rdparty/vmdisks/winiso:2019 # Замените
образ в соответствии с вашей средой
 name: containerdisk
 - containerDisk:
 image: registry.example.com:60070/3rdparty/vmdisks/win-virtio # Замените
образ в соответствии с вашей средой
 name: virtio
 - dataVolume:
 name: aa-test-rootfs
 name: rootfs
```

9. Нажмите **Create**.

10. Нажмите **Actions > VNC Login**.

11. Когда появится подсказка **press any key boot from CD or DVD**, нажмите любую клавишу для входа в программу установки Windows; если подсказка не отображается, нажмите **Send Remote Command** в левом верхнем углу страницы, затем выберите **Ctrl-Alt-Delete** из выпадающего меню для перезагрузки сервера.

12. **Примечание:** Если в верхней части страницы с деталями виртуальной машины появится сообщение **The current virtual machine has configuration changes that require a restart to take effect, please restart**, это сообщение можно игнорировать; перезагрузка не требуется.

### Установка операционной системы Windows

1. Следуйте инструкциям установки после входа на страницу установки.

**Примечание:** На этапе выбора раздела шина должна быть sata, чтобы диск корректно распознавался. Поэтому необходимо последовательно выбрать каждый раздел и нажать **Delete** для удаления всех разделов, чтобы система могла обработать их автоматически.

2. После настройки пароля администратора нажмите **Send Remote Command** в левом верхнем углу страницы, затем выберите **Ctrl-Alt-Delete** из выпадающего меню.

3. При появлении запроса **The Ctrl+Alt+Delete combination will restart the server, confirm to restart** нажмите **OK**.

4. Введите пароль для доступа к рабочему столу Windows; на этом установка операционной системы Windows завершена.

### Установка virtio-win-tools

Этот инструмент содержит необходимые драйверы.

1. Откройте Проводник.

2. Дважды щелкните **CD Drive(E:) virtio-win-<version>**, запустите каталог **virtio-win-guest-tools** для входа на страницу установки и следуйте инструкциям. Часть <version> должна соответствовать фактической версии.

3. После завершения установки выключите систему Windows.

### Экспорт пользовательского образа Windows

Пожалуйста, обратитесь к Export Virtual Machine Image для конкретных действий.

### Использование образа Windows

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Virtualization > Virtual Machines**.
3. Нажмите **Create Virtual Machine**.
4. Заполните необходимые параметры на странице формы. Для образа выберите экспортированный образ Windows. Для подробных параметров и конфигурации смотрите Create Virtual Machine.
5. (Опционально) Если используется более новая операционная система, например Windows 11, включите такие функции, как часы, UEFI, TPM и др. Переключитесь в YAML и замените исходный YAML следующим файлом.

```
apiVersion: kubevirt.io/v1
```

```
kind: VirtualMachineInstance
```

```
metadata:
```

```
 labels:
```

```
 special: vmi-windows
```

```
 name: vmi-windows
```

```
spec:
```

```
 domain:
```

```
 clock:
```

```
 timer:
```

```
 hpet:
```

```
 present: false
```

```
 hyperv: {}
```

```
 pit:
```

```
 tickPolicy: delay
```

```
 rtc:
```

```
 tickPolicy: catchup
```

```
 utc: {}
```

```
 cpu:
```

```
 cores: 2
```

```
 devices:
```

```
 disks:
```

```
 - disk:
```

```
 bus: sata
```

```
 name: pvcdisk
```

```
 interfaces:
```

```
 - masquerade: {}
```

```
 model: e1000
```

```
 name: default
 tpm: {}
 features:
 acpi: {}
 apic: {}
 hyperv:
 relaxed: {}
 spinlocks:
 spinlocks: 8191
 vapic: {}
 smm: {}
 firmware:
 bootloader:
 efi:
 secureBoot: true
 uuid: 5d307ca9-b3ef-428c-8861-06e72d69f223
 resources:
 requests:
 memory: 4Gi
 networks:
 - name: default
 pod: {}
 terminationGracePeriodSeconds: 0
 volumes:
 - name: pvcdisk
 persistentVolumeClaim:
 claimName: disk-windows
 - name: winiso
 persistentVolumeClaim:
 claimName: win11cd-pvc
```

#### 6. Нажмите **Create**

#### **Добавление внутреннего маршрута**

Настройте внутренний маршрут типа NodePort для открытия порта для подключения по удаленному рабочему столу.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Virtualization > Virtual Machines**.
3. В списке нажмите на имя виртуальной машины, созданной с образом Windows, чтобы перейти на страницу деталей.
4. Нажмите на иконку **Add** рядом с **Internal Route** в области **Login Information**.
5. Настройте параметры согласно следующим указаниям.



| Параметр | Описание                                                                                                                                                                                                                 |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Type     | Выберите <b>NodePort</b> .                                                                                                                                                                                               |
| Port     | <ul style="list-style-type: none"> <li>• Протокол: выберите TCP.</li> <li>• Service Port: используйте 3389.</li> <li>• Virtual Machine Port: используйте 3389.</li> <li>• Service Port Name: используйте rdp.</li> </ul> |

6. Нажмите **ОК** для возврата на страницу деталей.
7. Нажмите на ссылку **Internal Route** в области **Login Information**.
8. Сохраните информацию **Virtual IP** из области базовой информации и **Host Port** из области портов.

### Удалённый доступ

В данном разделе в качестве примера рассматривается удаленное подключение к операционной системе Windows. Для других ОС можно использовать программное обеспечение, поддерживающее протокол RDP.

1. Откройте **Remote Desktop Connection**.
2. Введите сохранённые Virtual IP и Host Port из шага **Add Internal Route** в формате **Virtual IP:Host Port**, например: 192.1.1.1:3389 .
3. Нажмите **Connect**.

### Создание образов Linux на основе ISO с использованием KubeVirt

В данном разделе описывается решение виртуальной машины, реализованное на базе открытого компонента KubeVirt. Оно использует технологию виртуализации KubeVirt для создания образа операционной системы Linux из ISO-образа.

#### Ограничения и особенности

- При запуске KubeVirt размер файловой системы пользовательского образа влияет на скорость записи образа на диск PVC. Если файловая система слишком большая, это может привести к длительному времени создания.
- Рекомендуется держать размер корневого раздела Linux менее 100G для минимизации начального размера. После настройки cloud-init при создании виртуальной машины выделяйте больший объем хранилища для корневого раздела, и система автоматически его расширит.

#### Шаги

##### Конвертация ISO-образа Linux в Docker-образ

1. Перейдите в каталог, где хранится ISO-образ, и выполните в терминале команду для переименования ISO-образа в ubuntu.iso.  
mv <ISO image name> ubuntu.iso # Замените <ISO image name> на фактическое имя образа, например, mv ubuntu-24.04-live-server-amd64.iso ubuntu.iso
2. Создайте Dockerfile, выполнив следующую команду.  
touch Dockerfile
3. Отредактируйте Dockerfile, добавьте следующий контент и сохраните файл.  
FROM scratch

ADD --chown=107:107 ubuntu.iso /disk/

4. Постройте Docker-образ, выполнив следующую команду.  
docker build -t build-harbor.example.com/3rdparty/vmdisks/ubuntu-iso:24.04 . #  
Замените репозиторий согласно вашей реальной среде
5. Отправьте образ в репозиторий, выполнив следующую команду.  
docker push build-harbor.example.com/3rdparty/vmdisks/ubuntu-iso:24.04 # Замените  
репозиторий согласно вашей реальной среде

### Создание виртуальной машины

1. Войдите в **Container Platform**.
2. В левом навигационном меню выберите **Virtualization > Virtual Machines**.
3. Нажмите **Create Virtual Machine**.
4. Заполните параметры на странице формы следующим образом. Для конкретных параметров и настроек смотрите раздел создания виртуальной машины.

| Параметр            | Описание                                                             |
|---------------------|----------------------------------------------------------------------|
| <b>Select Image</b> | Выберите шаблонный образ для виртуальной машины.                     |
| <b>IP Address</b>   | Оставьте по умолчанию, IP будет получен через <b>DHCP</b> .          |
| <b>Network Mode</b> | Используйте режим <b>NAT</b> ; не используйте режим <b>bridged</b> . |

5. Переключитесь в режим YAML.
6. Замените конфигурацию в поле spec.template.spec.domain.devices.disks следующим содержимым.

```
domain:
 devices:
 disks:
 - bootOrder: 1
 cdrom:
 bus: sata
 name: containerdisk
 - disk:
 bus: virtio
 name: cloudinitdisk
 - disk:
 bus: virtio
 name: rootfs
 bootOrder: 10
```

7. Добавьте следующее содержимое в поле spec.template.spec.volumes.

```
- containerDisk:
 image: registry.example.com:60070/3rdparty/vmdisks/ubuntu-iso:24.04 #
 name: containerdisk
```

Замените образ согласно вашей реальной среде

8. Проверьте YAML-файл; полный YAML после завершения выглядит следующим образом.

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
 annotations:
 kubevirt.io/latest-observed-api-version: v1
 kubevirt.io/storage-observed-api-version: v1
 labels:
 virtualization.cpaas.io/image-name: debian-2120-x86
 virtualization.cpaas.io/image-os-arch: amd64
 virtualization.cpaas.io/image-os-type: debian
 virtualization.cpaas.io/image-supply-by: public
 vm.cpaas.io/name: aa
 name: aa
spec:
 dataVolumeTemplates:
 - metadata:
 creationTimestamp: null
 labels:
 vm.cpaas.io/reclaim-policy: Delete
 vm.cpaas.io/used-by: aa
 name: aa-rootfs
 spec:
 pvc:
 accessModes:
 - ReadWriteOnce
 resources:
 requests:
 storage: 100Gi
 storageClassName: vm-cephrbd
 volumeMode: Block
 source:
 http:
 url: http://192.168.254.12/kube-debian-12.2.0-x86-out.qcow2
 running: true
 template:
 metadata:
 annotations:
 cpaas.io/creator: test@example.io
 cpaas.io/display-name: ""
```

cpaas.io/updated-at: 2024-09-09T03:49:08Z  
kubevirt.io/latest-observed-api-version: v1  
kubevirt.io/storage-observed-api-version: v1  
creationTimestamp: null  
labels:  
  virtualization.cpaas.io/image-name: debian-2120-x86  
  virtualization.cpaas.io/image-os-arch: amd64  
  virtualization.cpaas.io/image-os-type: debian  
  virtualization.cpaas.io/image-supply-by: public  
  vm.cpaas.io/name: aa  
spec:  
  accessCredentials:  
    - sshPublicKey:  
      propagationMethod:  
        qemuGuestAgent:  
          users:  
            - root  
      source:  
        secret:  
          secretName: test-xeon  
  affinity:  
    nodeAffinity: {}  
  architecture: amd64  
  domain:  
  devices:  
    disks:  
      - bootOrder: 1  
        cdrom:  
          bus: sata  
          name: containerdisk  
      - disk:  
          bus: virtio  
          name: cloudinitdisk  
      - disk:  
          bus: virtio  
          name: rootfs  
          bootOrder: 10  
  interfaces:  
    - bridge: {}  
      name: default

```

machine:
 type: q35
resources:
 limits:
 cpu: "1"
 memory: 2Gi
 requests:
 cpu: "1"
 memory: 2Gi
networks:
 - name: default
 pod: {}
nodeSelector:
 kubernetes.io/arch: amd64
 vm.cpaas.io/baremetal: "true"
volumes:
 - containerDisk:
 image: registry.example.com:60070/3rdparty/vmdisks/ubuntu-iso:24.04 #
Замените образ согласно вашей реальной среде
 name: containerdisk
 - cloudInitConfigDrive:
 userData: |-
 #cloud-config
 disable_root: false
 ssh_pwauth: false
 users:
 - default
 - name: root
 lock_passwd: false
 hashed_passwd: ""
 name: cloudinitdisk
 - dataVolume:
 name: aa-rootfs
 name: rootfs

```

9. Нажмите **Create**.

10. Нажмите **Actions > VNC Login**.

11. При появлении сообщения **press any key boot from CD or DVD** нажмите любую клавишу, чтобы войти в программу установки Windows; если сообщение не отображается, нажмите **Send Remote Command** в левом верхнем углу страницы, затем выберите **Ctrl-Alt-Delete** из выпадающего меню для перезагрузки сервера.

**Примечание:** Если в верхней части страницы с деталями виртуальной машины появляется сообщение **Current virtual machine has configuration changes that require a restart to take effect. Please restart.**, его можно игнорировать; перезагрузка не требуется.

### Установка операционной системы Linux

1. После входа на страницу установки следуйте инструкциям мастера установки. В данном документе приведен пример установки операционной системы Ubuntu; параметры конфигурации в процессе установки разных ОС обычно схожи, поэтому подробности не приводятся. Ниже приведены некоторые пояснения по конфигурации.

| Конфигурация           | Описание                                                                                                                                                                                                      |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Тип установки          | Рекомендуется использовать минимальную установку для минимизации размера образа.                                                                                                                              |
| Конфигурация хранилища | Выберите пользовательское хранилище. Отформатируйте диск в формате ext4 или xfs и смонтируйте в корневой раздел (/).<br><u>Примечание:</u> Не используйте LVM для разметки диска (Create volume group (LVM)). |
| Конфигурация SSH       | Выберите установку OpenSSH для доступа по SSH.                                                                                                                                                                |

2. Дождитесь завершения установки.

### Изменение YAML-файла

1. Войдите в **Container Platform**.
2. В левом навигационном меню выберите **Virtualization > Virtual Machines**.
3. Нажмите на **Имя виртуальной машины** в списке, чтобы перейти на страницу деталей.
4. Нажмите **Stop**.
5. В правом верхнем углу нажмите **Actions > Update**.
6. Переключитесь в режим YAML.
7. Убедитесь, что диск с именем **rootfs** в `spec.template.spec.domain.devices.disks` имеет `bootOrder` равный 1. Если нет, измените на 1.
8. Удалите соответствующий блок для диска с именем **containerdisk** в `spec.template.spec.domain.devices.disks`; конкретно удалите следующий контент.  
- `bootOrder: 1`  
  `cdrom:`  
    `bus: sata`  
    `name: containerdisk`
9. Удалите соответствующий блок для диска с именем **containerdisk** в `spec.template.spec.volumes`; конкретно удалите следующий контент.  
- `containerDisk:`  
  `image: registry.example.com:60070/3rdparty/vmdisks/ubuntu-iso:24.04`  
  `name: containerdisk`
10. Нажмите **Update**.
11. Нажмите **Start**.

## Установка необходимого ПО и изменение конфигурации

**Примечание:** Следующие команды и файлы конфигурации могут незначительно отличаться в разных операционных системах; корректируйте их согласно вашей реальной среде.

1. Войдите в операционную систему, используя имя пользователя и пароль.
2. Переключитесь на пользователя root.
3. Установите необходимые пакеты.
  - Для CentOS выполните команду:  
`yum install cloud-utils cloud-init qemu-guest-agent vim`
  - Для Debian выполните команду:  
`apt install cloud-init cloud-guest-utils qemu-guest-agent vim`
4. Отредактируйте конфигурационный файл SSHD:
  - 4.1. Выполните команду для редактирования `sshd_config`.  
`vim /etc/ssh/sshd_config`
  - 4.2. Добавьте следующие настройки.  
`PermitRootLogin yes # Разрешить вход root-пользователю с паролем`  
`PubkeyAuthentication yes # Разрешить вход по ключу`
  - 4.3. Сохраните изменения.
5. Выполните команду для удаления пароля root-пользователя.  
`passwd -d root`
6. Измените файл источников пакетов.
  - 6.1. Выполните команду для редактирования файла источников и замените адрес на подходящий зеркальный сайт.  
`vim /etc/apt/sources.list.d/ubuntu.sources`
  - 6.2. Сохраните изменения.
7. Измените конфигурацию cloud-init для автоматического расширения корневого раздела.
  - 7.1. Выполните команду для редактирования файла `cloud.cfg`.  
`vim /etc/cloud/cloud.cfg`
  - 7.2. Добавьте следующий контент.  
`runcmd:`  
`[growpart, /dev/vda, 1] # Команда growpart расширяет раздел на диске, расширяя раздел /dev/vda1.`  
`- [xfs_growfs, /dev/vda1] # Команда xfs_growfs расширяет файловую систему XFS, чтобы занять все доступное пространство раздела. /dev/vda1 — раздел с расширяемой файловой системой. После расширения раздела xfs_growfs гарантирует расширение файловой системы до нового размера раздела.`
  - 7.3. Сохраните изменения.
8. После завершения конфигурации выключите операционную систему.

## Экспорт и использование пользовательского образа Linux

Для конкретных операций смотрите [Экспорт образов виртуальных машин](#).

### Экспорт образов виртуальных машин

Эта функциональность используется для экспорта системного образа виртуальной машины и загрузки его в объектное хранилище, что позволяет добавлять файлы из объектного хранилища в качестве источников для образов виртуальных машин платформы.

#### Шаги

**Примечание:** Все операции ниже необходимо выполнять на управляющем узле кластера, где находится виртуальная машина.

#### Остановка виртуальной машины

1. Перейдите в **Управление платформой**.
2. В левой навигационной панели нажмите **Управление виртуализацией > Виртуальные машины**.
3. Нажмите на имя виртуальной машины, системный образ которой необходимо экспортировать, после чего вы перейдёте на страницу с деталями виртуальной машины в Container Platform.
4. Нажмите **Остановить**.

#### Создание ресурса vmexport

1. Откройте CLI инструмент.
2. Выполните следующую команду для установки переменных.  
NAMESPACE=<namespace>  
VM\_NAME=<vm\_name>  
TTL\_DURATION=2h

Описание параметров:

1. **NAMESPACE:** имя namespace, в котором находится виртуальная машина; замените часть <namespace> на это имя.
  2. **VM\_NAME:** имя виртуальной машины, системный образ которой нужно экспортировать; замените часть <vm\_name> на это имя.
  3. **TTL\_DURATION:** время жизни задачи экспорта, по умолчанию 2 часа, но при необходимости можно увеличить.
3. Выполните следующую команду для создания ресурса vmexport.
- ```
cat <<EOF | kubectl create -f -
apiVersion: export.kubevirt.io/v1alpha1
kind: VirtualMachineExport
metadata:
  name: export-$VM_NAME
  namespace: $NAMESPACE
spec:
  ttlDuration: $TTL_DURATION
  source:
    apiGroup: "kubevirt.io"
```



```
kind: VirtualMachine
name: $VM_NAME
```

EOF

Если появится подобное сообщение, значит создание прошло успешно.

```
virtualmachineexport.export.kubevirt.io/export-k1 created
```

4. Выполните следующую команду для проверки статуса ресурса vmexport.

```
kubectl -n $NAMESPACE get vmexport export-$VM_NAME -w
```

Вывод:

```
NAME      SOURCEKIND  SOURCENAME  PHASE
export-k1 VirtualMachine k1      Ready
```

5. Когда в поле PHASE в выводе появится значение Ready, нажмите ctrl (control) + c для остановки операции watch.
6. Выполните следующую команду для получения TOKEN.

```
TOKEN=$(kubectl -n $NAMESPACE get secret export-token-export-$VM_NAME -o
jsonpath={.data.token} | base64 -d)
```

Загрузка файла образа виртуальной машины

1. Выполните следующую команду для получения IP-адреса Pod'a виртуального экспорта машины в указанном namespace и сохраните его в переменную окружения EXPORT_SERVER_IP.

```
EXPORT_SERVER_IP=$(kubectl -n $NAMESPACE get po virt-export-export-$VM_NAME -o
jsonpath='{.status.podIP}')
```

2. Выполните следующую команду для установки переменной окружения URL, указывающей на файл дискового образа виртуальной машины.

```
URL=https://$EXPORT_SERVER_IP:8443/volumes/$VM_NAME-rootfs/disk.img.gz
```

3. Выполните следующую команду для загрузки файла образа, при этом загруженный файл будет называться disk.img.gz.

```
curl -k -O -H "x-kubevirt-export-token: $TOKEN" $URL
```

Загрузка файла образа виртуальной машины в объектное хранилище

Загрузите скачанный файл образа в объектное хранилище. Для загрузки можно использовать любой S3-инструмент, в данном документе в качестве примера используется инструмент mc (minio-client).

1. Выполните следующую команду для настройки инструмента mc и подключения к указанному S3-сервису хранения.

```
mc alias set minio <ENDPOINT> <ACCESSKEY> <SECRETKEY>
```

Описание параметров:

- ENDPOINT: адрес S3-сервиса хранения; замените часть <ENDPOINT> на этот адрес.
 - ACCESSKEY, SECRETKEY: ак и sk пользователя S3-сервиса хранения, используемые для аутентификации; по связанным вопросам смотрите MinIO Object Storage.
2. Выполните следующую команду для создания бакета для хранения файлов образов виртуальных машин.
- ```
mc mb minio/vmdisks
```

3. Выполните следующую команду для загрузки экспортированного файла образа виртуальной машины disk.img.gz в созданный бакет.  
mc put disk.img.gz minio/vmdisks

### Создание образа виртуальной машины

1. Перейдите в **Управление платформой**.
2. В левой навигационной панели нажмите **Управление виртуализацией > Образы виртуальных машин**.
3. Нажмите **Добавить образ виртуальной машины**.
4. В поле адреса образа укажите <ENDPOINT>/vmdisks/disk.img.gz, заменив часть <ENDPOINT> на адрес S3-сервиса хранения. По другим параметрам смотрите Добавление образов виртуальных машин.
5. Нажмите **Добавить**.

### Создание виртуальных машин/групп виртуальных машин

KubeVirt предоставляет CRD (Custom Resource Definitions), такие как **VirtualMachine** и **VirtualMachineInstance**, для абстрагирования ресурсов виртуальных машин (VM). На основе этих CRD пользователи получают полноценные возможности управления виртуальными машинами. В дополнение к этому, **MWS CP Virtualization With KubeVirt** расширяет удобство использования, предлагая **Web Console**, которая позволяет пользователям выполнять различные операции значительно проще.

Создайте виртуальную машину (VirtualMachineInstance) с использованием образа и запланируйте виртуальную машину на физические узлы с установленными компонентами Kubevirt и включённой виртуализацией.

Вы можете создать одну виртуальную машину через Create Virtual Machine или быстро создать несколько виртуальных машин (VirtualMachineInstance) с одинаковой конфигурацией, используя Create Virtual Machine Group (virtualMachinePool).

**Примечание:** При использовании виртуальных машин Windows поддерживается только вход по **имени пользователя/паролю**, заданным в образе виртуальной машины. Пожалуйста, заранее свяжитесь с администратором платформы для получения этой информации.

### Создание виртуальной машины

**Примечание:** Ниже приведён пример создания виртуальной машины с использованием формы, также вы можете переключиться на формат YAML для выполнения операции.

1. Войдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Нажмите **Create Virtual Machine**.
4. В области **Basic Information** заполните имя и отображаемое имя виртуальной машины, а также задайте теги или аннотации.

| Параметр | Описание |
|----------|----------|
|----------|----------|

|                    |                                                                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tags</b>        | Используются для выбора объектов и поиска коллекций объектов, соответствующих определённым критериям. Должны быть парой ключ-значение, например: <i>app.kubernetes.io/name: hello-app</i> . |
| <b>Annotations</b> | Используются для предоставления любой информации командам разработки и эксплуатации. Должны быть парой ключ-значение, например: <i>cpaas.io/maintainer: kim</i> .                           |

5. Установите тип машины и выберите образ виртуальной машины.

| Параметр                    | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Specifications</b>       | Вы можете выбрать рекомендованные сценарии использования или задать собственные ограничения ресурсов в зависимости от ваших потребностей.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Physical GPU (Alpha)</b> | <p>Выберите модель физического GPU; каждому виртуальному компьютеру может быть выделен только один физический GPU.</p> <p><b>Примечание:</b> Passthrough физического GPU для виртуальной машины означает прямое выделение реального графического процессора (GPU) виртуальной машине в виртуализированной среде, что позволяет ей напрямую обращаться к физическому GPU и использовать его для достижения графической производительности, эквивалентной работе на физической машине, избегая узких мест производительности, вызванных виртуальными графическими адаптерами, и повышая общую производительность.</p> |
| <b>Image</b>                | <p>Выберите публичный образ, назначенный проекту платформы администратором платформы.</p> <p><b>Примечание:</b> Поддерживается только выбор образов с той же <b>CPU архитектурой</b>, что и архитектура кластера.</p>                                                                                                                                                                                                                                                                                                                                                                                               |

6. В области **Storage** настройте соответствующую информацию согласно следующим инструкциям.

| Параметр             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Disk Name</b>     | Имя диска хранения; имя системного диска изменить нельзя.                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Type</b>          | <ul style="list-style-type: none"> <li>• <b>Root Disk:</b> Система автоматически создаёт системный диск rootfs типа VirtIO для хранения операционной системы и данных.</li> <li>• <b>Data Disk:</b> Нажмите для добавления нескольких дисков данных для постоянного хранения данных. По умолчанию устройство VirtIO.</li> </ul> <p><b>Примечание:</b> Имена дисков данных не должны дублировать существующие имена дисков.</p> |
| <b>Volume Mode</b>   | <ul style="list-style-type: none"> <li>• <b>File System:</b> Монтировать диск как смонтированный файловый каталог.</li> <li>• <b>Block Device:</b> Монтировать диск как блочное устройство.</li> </ul>                                                                                                                                                                                                                         |
| <b>Storage Class</b> | Платформа поддерживает диски виртуальных машин, создавая и управляя persistent volume claims. Необходимо указать storage class, требуемый для динамического создания persistent volume claims.                                                                                                                                                                                                                                 |

|                       |                                                                                                                                                                 |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | Разные storage class поддерживают разные volume mode; если для выбранного volume mode нет доступного storage class, обратитесь к администратору для добавления. |
| <b>Capacity</b>       | Требуемая ёмкость для хранения виртуальной машины; минимум для системного диска — 20 ГБ.                                                                        |
| <b>Delete with VM</b> | По умолчанию включено и не может быть изменено, что означает удаление данных диска при удалении виртуальной машины.                                             |

7. В области **Network** настройте соответствующую информацию согласно следующим инструкциям.

| Параметр                      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>IP Address</b>             | <ul style="list-style-type: none"> <li>По умолчанию <b>Dynamic (DHCP)</b>; IP-адрес динамически назначается при запуске виртуальной машины и освобождается при её остановке.</li> <li>Если привязан <b>Static IP</b>, виртуальная машина всегда будет использовать этот IP-адрес даже после перезапуска. Если в текущем проекте нет доступных IP, сначала освободите IP.</li> </ul>                                                                                                                                                                         |
| <b>Network Mode</b>           | <ul style="list-style-type: none"> <li><b>Bridged</b>: Виртуальная машина использует тот же IP-адрес, что и контейнерная группа, и взаимодействует с внешним миром через этот IP.</li> <li><b>NAT</b>: Виртуальной машине назначается внутренний IP, который транслируется в IP контейнерной группы для внешнего взаимодействия. Открытые порты указывают порты виртуальной машины, например порт SSH 22; если <b>Open Ports</b> не заполнены, считаются открытыми все порты.</li> </ul>                                                                    |
| <b>Auxiliary Network Card</b> | <p>Добавляйте вспомогательные сетевые карты по необходимости.</p> <p><b>Примечание:</b></p> <ul style="list-style-type: none"> <li>Если требуется функция вспомогательных сетевых карт или нет доступных типов вспомогательных сетей, обратитесь к администратору платформы для настройки.</li> <li>Типы SR-IOV поддерживаются только для Linux на архитектуре x86_64.</li> <li>По умолчанию IP-адреса получают через DHCP.</li> <li>После нескольких перезагрузок SR-IOV виртуальные машины могут иметь два разных VF с одинаковым MAC-адресом.</li> </ul> |

8. В области **Initialization Settings** настройте соответствующую информацию согласно следующим инструкциям.

| Параметр    | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Keys</b> | <p>Всегда используйте SSH-ключи для проверки удалённого входа. Этот метод не требует проверки пароля; рекомендуется входить в виртуальную машину с помощью ключей.</p> <ul style="list-style-type: none"> <li>Вы можете использовать уже имеющиеся ключи на платформе или создать новые; все ключи доступны на странице <b>Virtualization &gt; Key Pairs</b>.</li> <li>Доступ по SSH к виртуальной машине возможен только у лиц, имеющих приватный ключ. Если несколько человек обслуживают виртуальную машину, можно связать несколько ключей и распределить приватные ключи между пользователями. В случае утечки ключа соответствующий ключ можно быстро отозвать для минимизации ущерба.</li> <li>Публичный ключ SSH хранится на платформе в конфиденциальном</li> </ul> |

| Параметр                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | <p>виде; приватный ключ не хранится, поэтому храните его в безопасности самостоятельно.</p> <ul style="list-style-type: none"> <li>Пожалуйста, обратитесь к документации соответствующей операционной системы для пароля пользователя root.</li> </ul>                                                                                                                                                                                                                                               |
| <b>Password</b>          | <p>Используйте пользователя операционной системы и пароль для проверки входа, который впоследствии можно обновить на метод с ключами.</p> <ul style="list-style-type: none"> <li>Пользователь — это только начальная учётная запись; после успешного создания виртуальной машины можно создавать других пользователей ОС для входа.</li> <li>Платформа шифрует и хранит пароль пользователя root, и вы не увидите его в открытом виде, поэтому храните его в безопасности самостоятельно.</li> </ul> |
| <b>Start Immediately</b> | <p>По умолчанию включено. При включении виртуальная машина запускается сразу после создания, иначе создаётся только виртуальная машина без запуска.</p>                                                                                                                                                                                                                                                                                                                                              |

9. (Опционально) В области **Advanced Configuration** настройте соответствующую информацию согласно следующим инструкциям.

| Параметр             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Health Check</b>  | <ul style="list-style-type: none"> <li><b>Liveness Check:</b> Проверяет, находится ли виртуальная машина в здоровом состоянии; при обнаружении аномалий определяется необходимость перезапуска экземпляра согласно настройкам проверки здоровья.</li> <li><b>Availability Check:</b> Проверяет, завершён ли запуск виртуальной машины и находится ли она в нормальном рабочем состоянии; при обнаружении аномалий состояние виртуальной машины обновляется.</li> </ul>                 |
| <b>Node Affinity</b> | <ul style="list-style-type: none"> <li><b>Preferred:</b> Виртуальная машина будет по возможности запланирована на узлы, соответствующие требованиям affinity. Система определяет узлы, способные запустить виртуальную машину, комбинируя веса affinity и другие требования планирования (например, требования к вычислительным ресурсам).</li> <li><b>Required:</b> Виртуальная машина будет запланирована только на узлы, полностью соответствующие требованиям affinity.</li> </ul> |

10. После проверки правильности информации нажмите **Create**.

Дождитесь, пока статус виртуальной машины изменится с **Creating** на **Running**.

## Операции с виртуальной машиной

Вы можете нажать на значок : справа на странице списка или выбрать **Actions** в правом верхнем углу страницы деталей для обновления или удаления виртуальной машины по необходимости. Для других операций, таких как сброс пароля или обновление ключей, пожалуйста, обратитесь к разделу управления виртуальными машинами.

### Примечания:

- Обновление возможно только при статусах виртуальной машины **Abnormal**, **Unknown** или **Stopped**.
- Обновление не поддерживает отображение дисков, которые были отдельно присоединены или созданы после создания виртуальной машины.

- По умолчанию при обновлении опция **Start Immediately** отключена; вы можете включить её при необходимости.

## Создание группы виртуальных машин

**Примечание:** Ниже приведён пример создания группы виртуальных машин с использованием формы, также вы можете переключиться на формат YAML для выполнения операции.

1. Войдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machine Groups**.
3. Нажмите **Create Virtual Machine Group**.
4. В области **Basic Information** настройте информацию для группы виртуальных машин согласно следующим инструкциям.

| Параметр                               | Описание                                                                                                                                                                                                                                           |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Number of Instances</b>             | Количество виртуальных машин, создаваемых группой виртуальных машин.                                                                                                                                                                               |
| <b>Anti-Affinity between Instances</b> | Если включено, при планировании нескольких виртуальных машин на узлы будет предпринята попытка распределить виртуальные машины по разным узлам, что повышает высокую доступность группы виртуальных машин.                                         |
| <b>Tags</b>                            | Для группы виртуальных машин можно добавить теги. Теги используются для выбора объектов и поиска коллекций объектов, соответствующих определённым критериям. Должны быть парой ключ-значение, например: <i>app.kubernetes.io/name: hello-app</i> . |

5. В области **Virtual Machine Template** настройте единые теги, аннотации, спецификации, образы, хранилище и другую информацию для всех виртуальных машин группы согласно Create Virtual Machine.
6. После проверки правильности информации нажмите **Create**.

**Совет:** После успешного создания вы можете перейти на страницу списка **Virtual Machines**, чтобы просмотреть информацию о виртуальных машинах, созданных через группу виртуальных машин.

## Вход в виртуальную машину с использованием VNC

Выполните вход в виртуальную машину через Web Console (VNC) в качестве аварийного способа управления:

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Нажмите **: > VNC Login**.

4. Окно консоли откроется автоматически; для входа необходимо ввести имя пользователя и пароль.

#### Примечания:

- Поддерживается отправка стандартных команд с клавиатуры.
- Поддерживается копирование и вставка команд и параметров.

## Управление виртуальными машинами

### Сброс пароля

Пароль пользователя root служит паролем для входа в виртуальную машину при использовании входа по паролю. Сбросьте пароль:

1. Зайдите в **Container Platform**. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
2. Найдите виртуальную машину и выберите : > **Reset Password**.
3. Установите пароль.
4. Нажмите **Reset**.

**Примечание:** Пожалуйста, храните пароль в безопасности. Для обеспечения безопасности среды платформа шифрует и сохраняет ваш пароль, и вы не сможете увидеть пароль в открытом виде повторно.

### Обновление ключа

Обновите SSH-ключи:

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Найдите виртуальную машину и выберите : > **Update Key**.
4. Выберите один или несколько связанных ключей либо **Create Key**.
5. Выберите, нужно ли перезапускать сразу; для применения обновления ключей требуется перезапуск виртуальной машины.
6. Нажмите **Update**.

### Обновление спецификаций

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Найдите нужную виртуальную машину и выберите : > **Update Specifications**.
4. Измените соответствующие ресурсы в соответствии с рекомендованными платформой сценариями или пользовательскими требованиями.
5. Выберите, нужно ли Restart Immediately; конфигурация вступит в силу после перезапуска.
6. Нажмите **Update**.

### Живая миграция

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Найдите нужную виртуальную машину и выберите : > **Live Migration**.

4. Нажмите **Confirm**.

#### Обновление конфигурации NAT-сети

При использовании режима NAT-сети платформа по умолчанию открывает порт 22 для SSH-сервисов, а также вы можете открыть другие порты по необходимости.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Нажмите на **Virtual Machine Name**.
4. В разделе **Basic Information** нажмите на иконку справа от **Open Port**.
5. Введите номер порта и нажмите клавишу Enter для подтверждения.
6. Выберите, нужно ли **Restart Immediately**; конфигурация вступит в силу после перезапуска.
7. Нажмите **Update**.

#### Обновление тегов и аннотаций

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Нажмите на **Virtual Machine Name**.
4. В разделе **Basic Information** нажмите на иконку справа от **Tags** или **Annotations**.
5. Настройте по необходимости и нажмите **Update**.

#### Добавление сервиса

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Нажмите на **Virtual Machine Name**.
4. В разделе **Login Information** нажмите на иконку справа от Internal Route.
5. Ознакомьтесь со страницей Create Service для быстрого добавления внутренних маршрутов для виртуальной машины.
6. Нажмите **Confirm**.

#### Переустановка операционной системы

Сделайте резервную копию данных перед переустановкой операционной системы, чтобы избежать потери данных.

**Внимание:** Эта операция очистит все данные на **системном диске** виртуальной машины, а также все **снимки**, и является необратимой. Пожалуйста, будьте осторожны!

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Найдите виртуальную машину и выберите **: > Reinstall Operating System**.
4. В окне Reinstall Operating System настройте следующие параметры.
  - a. **Provisioning Method**: в настоящее время поддерживаются публичные образы.
  - b. **Select Image**: по умолчанию для переустановки будет использоваться текущий образ операционной системы. Если вы хотите переустановить новую ОС, сначала выберите операционную систему образа виртуальной машины, затем выберите образ виртуальной машины, принадлежащий этой ОС.



5. Нажмите **Reinstall**.

### Настройка IP

Назначьте виртуальной машине IP с помощью динамического выделения (DHCP) или привяжите фиксированный IP; новый IP вступит в силу после перезапуска виртуальной машины.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machines**.
3. Найдите нужную виртуальную машину и выберите : > **Configure IP**.
4. Настройте **IP Address**.
  - а. Заполните доступный IP: привязка фиксированного IP означает, что даже после перезапуска виртуальная машина будет постоянно использовать этот IP-адрес.
  - б. Оставьте поле пустым: будет использоваться динамическое выделение (DHCP), при котором IP назначается при запуске виртуальной машины и освобождается при её остановке.
5. Выберите, нужно ли **Restart Immediately**; конфигурация вступит в силу после перезапуска.
6. Нажмите **Configure**.

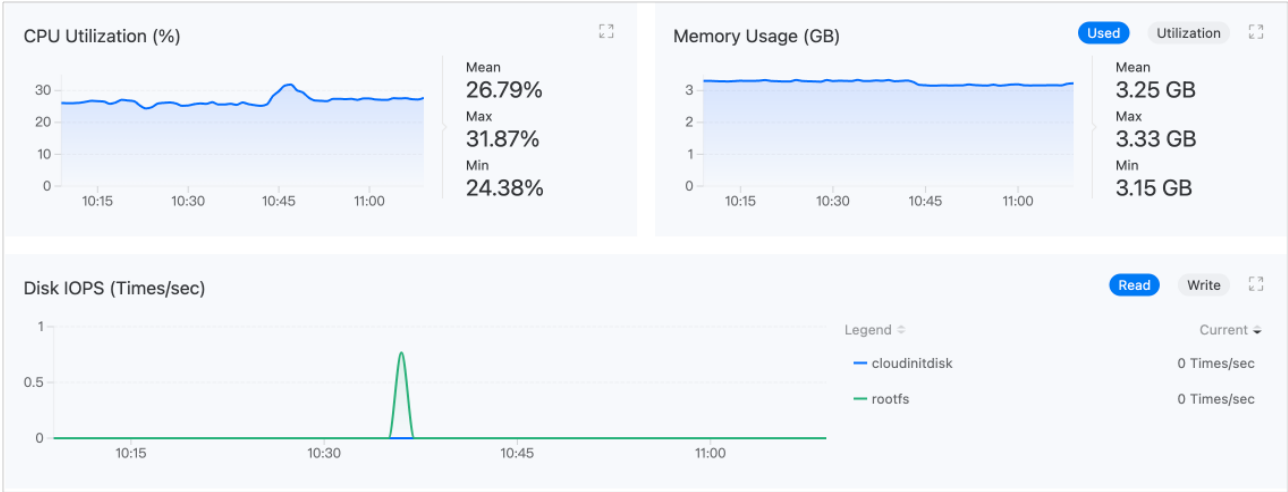
### Мониторинг и оповещения

Мониторинг и оповещение виртуальных машин по параметрам CPU, памяти, хранилища и сети. Для своевременного информирования также можно настроить политики уведомлений.

Интуитивно представленные данные мониторинга могут использоваться для поддержки принятия решений при проверке работы или настройке производительности, а комплексный механизм оповещений и уведомлений поможет обеспечить стабильную работу виртуальных машин.

### Мониторинг

По умолчанию платформа собирает часто используемые метрики мониторинга производительности виртуальных машин, включая CPU, память, хранилище и сеть. Перейдите в **Virtualization > Virtual Machines**, и на вкладке **Monitoring** в деталях виртуальной машины вы можете просмотреть данные мониторинга в реальном времени по этим метрикам.



### Оповещения

#### Настройка политик оповещений

Для включения оповещений необходимо сначала создать политику оповещений. Политика оповещений описывает объекты, которые вы хотите мониторить, условия, при которых необходимо оповещать, и способы уведомления о соответствующих оповещениях. Перейдите в **Container Platform > Virtualization > Virtual Machines**, и в деталях виртуальной машины на вкладке **Alerts** нажмите **Create Alert Policy** для завершения настройки.

| Параметр          | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alert Type        | <p>Metric Alert: объект мониторинга — предопределённая метрика платформы, например <i>Memory Usage Rate</i>.</p> <p>Event Alert: объект мониторинга — причина события, то есть причина, по которой виртуальная машина перешла в текущее состояние, например BackOff, Pulling, Failed.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Trigger Condition | <p>Состоит из операторов сравнения, порогов оповещения и длительности. Путём сравнения результатов мониторинга в реальном времени с установленными порогами определяется необходимость оповещения.</p> <p>Если задана длительность, платформа также сравнивает время, в течение которого объект мониторинга находился в состоянии оповещения.</p>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Alert Level       | <ul style="list-style-type: none"><li>Hint: у объекта мониторинга есть ожидаемые проблемы, которые не влияют немедленно на бизнес-процессы, но представляют потенциальный риск. Например, если использование CPU превышает 70% в течение 3 минут.</li><li>Warning: у объекта мониторинга есть операционные риски, которые при отсутствии своевременного реагирования могут повлиять на нормальную работу бизнеса. Например, если использование CPU превышает 80% в течение 3 минут.</li><li>Serious: у объекта мониторинга выявлены проблемы, которые могут привести к сбоям функциональности платформы и повлиять на нормальную работу бизнеса.</li><li>Disaster: объект мониторинга вышел из строя, что привело к прерыванию сервисов платформы, потере данных и</li></ul> |

|  |                            |
|--|----------------------------|
|  | значительному воздействию. |
|--|----------------------------|

**Совет:** Функциональность оповещений виртуальной машины схожа с общей функцией оповещений платформы. Для более подробных инструкций по настройке обратитесь к общей документации по Alerts.

### **Обработка оповещений**

Перейдите на вкладку **Alerts**, и если указаны стратегии состояния оповещений, пожалуйста, оперативно примите меры.

### **Привязка политик уведомлений**

Помимо оповещений в реальном времени на вкладке **Alerts**, платформа также поддерживает отправку информации об оповещениях по электронной почте, SMS и другим способам соответствующим сотрудникам, уведомляя их о необходимости принять меры для устранения проблем или предотвращения сбоев. Политика уведомлений настраивается через администратора.

## **Настройка проброса USB-хоста**

### **Обзор**

Функциональность проброса USB (Universal Serial Bus) позволяет получить доступ к USB-устройствам и управлять ими из виртуальной машины.

### **Сценарии использования**

Некоторые приложения, работающие в виртуальных машинах (VM), имеют требования к шифрованию и нуждаются во взаимодействии с выделенными USB-устройствами. В таких случаях необходимо пробросить USB-устройства с хост-машины в виртуальную машину.

### **Требования**

- Версия платформы должна быть не ниже v3.18.

### **Шаги**

#### **Шаг 1. Открытие USB-устройств**

Чтобы назначить USB-устройство виртуальной машине, USB-устройство должно быть открыто через **ResourceName**. Это можно настроить, отредактировав раздел `spec.permittedHostDevices.usbHostDevices` в **HyperConverged CR** в пространстве имён `kubevirt`.

Ниже приведён пример конфигурации для USB-устройства с **ResourceName** `kubevirt.io/storage`, где `vendor` — `Obda`, а `product` — `8812`:

```
spec:
 permittedHostDevices:
 usbHostDevices:
 - resourceName: kubevirt.io/storage
 selectors:
 - vendor: 'Obda'
 product: '8812'
```

**Совет:** Идентификаторы `vendor` и `product` USB-устройства можно получить с помощью команды `lsusb`. Например:

```
lsusb
```

```
Bus 001 Device 007: ID 0bda:8812 Realtek Semiconductor Corp. RTL8812AU
802.11a/b/g/n/ac 2T2R DB WLAN Adapter
```

Эта команда выводит список всех подключенных USB-устройств, где ID отображает пару vendor:product .

## Шаг 2. Назначение USB-устройств виртуальной машине

Теперь в конфигурации VM можно добавить `spec.domain.devices.hostDevices.deviceName`, чтобы сослаться на `ResourceName`, указанный на предыдущем шаге, и присвоить ему локальное имя. Например:

спес:

```
domain:
```

```
devices:
```

```
hostDevices:
```

```
- deviceName: kubevirt.io/storage
```

```
name: usb-storage
```

**Совет:** Убедитесь, что виртуальная машина остановлена перед редактированием конфигурации.

## Результат выполнения

После завершения настройки выполните команду `lsusb` внутри виртуальной машины. Если в выводе отображается USB-устройство хост-узла, значит проброс прошёл успешно. Например:

```
lsusb
```

```
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
```

```
Bus 001 Device 002: ID 0bda:8812 Realtek Semiconductor Corp. RTL8812AU
802.11a/b/g/n/ac 2T2R DB WLAN Adapter
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

## Открытие нескольких USB-устройств

Возможно, вам потребуется пробросить несколько USB-устройств в виртуальную машину, например клавиатуру, мышь или устройство смарт-карты. Платформа поддерживает назначение нескольких USB-устройств под одним `resourceName`. Для этого выполните следующие настройки:

спес:

```
permittedHostDevices:
```

```
usbHostDevices:
```

```
- resourceName: kubevirt.io/peripherals
```

```
selectors:
```

```
- vendor: '0bda'
```

```
product: '8812'
```

```
- vendor: '062a'
```

```
product: '4102'
```

```
- vendor: '072f'
```

```
product: 'b100'
```

**Совет:** Внимание: Все USB-устройства должны быть физически подключены и обнаружены на хосте для успешного назначения виртуальной машине.

#### Назначение USB-устройств виртуальной машине

```
спес:
 domain:
 devices:
 hostDevices:
 - deviceName: kubevirt.io/peripherals
 name: local-peripherals
```

#### Клонирование виртуальной машины

Клонирование виртуальной машины означает создание копии виртуальной машины на определённый момент времени. Клонированная виртуальная машина содержит все конфигурации, операционные системы, приложения и данные исходной виртуальной машины.

**Примечание:** Все операции ниже должны выполняться на мастер-узле кластера, где находится виртуальная машина.

1. Откройте CLI-инструмент.
2. Выполните следующую команду для создания и открытия файла vm-clone.sh.

```
vi vm-clone.sh
```

3. Нажмите i и скопируйте следующий содержимое в файл vm-clone.sh.

```
#!/bin/bash
```

```
vm_clone() {
 NAMESPACE=$1
 VM_NAME=$2
 VM_CLONE_NAME=$3

 cat <<EOF | kubectl create -f -
kind: VirtualMachineClone
apiVersion: clone.kubevirt.io/v1alpha1
metadata:
 name: clone-$VM_NAME
 namespace: $NAMESPACE
spec:
 source:
 apiGroup: kubevirt.io
 kind: VirtualMachine
 name: $VM_NAME
 target:
 apiGroup: kubevirt.io
```

```
kind: VirtualMachine
name: $VM_CLONE_NAME
```

```
labelFilters:
```

```
 - "*"
 - "!ovn.kubernetes.io/*"
```

```
annotationFilters:
```

```
 - "*"
 - "!ovn.kubernetes.io/*"
```

```
template:
```

```
 labelFilters:
```

```
 - "*"
 - "!ovn.kubernetes.io/*"
```

```
 annotationFilters:
```

```
 - "*"
 - "!ovn.kubernetes.io/*"
```

```
EOF
```

```
if [$? -eq 0]; then
```

```
 echo "Create vmclone resource Succeeded"
```

```
else
```

```
 echo "Create vmclone resource failed"
```

```
 exit 1
```

```
fi
```

```
echo "Waiting for vm clone completion"
```

```
while true; do
```

```
 phase=$(kubectl -n $NAMESPACE get vmclone clone-$VM_NAME -o
jsonpath='{.status.phase}')
```

```
 if ["$phase" == "Succeeded"]; then
```

```
 break
```

```
 elif ["$phase" == "Failed"]; then
```

```
 echo "VirtualMachineClone resource phase is Failed"
```

```
 exit 1
```

```
 fi
```

```
 sleep 5
```

```
done
```

```
echo "vm clone completion"
```

```
dvList=$(kubectl -n $NAMESPACE get vm $VM_CLONE_NAME -o
jsonpath='{.spec.template.spec.volumes}' | jq . | grep restore- | grep name | awk '{print
$2}')
```

```

for dv in $dvList; do
 kubectl -n $NAMESPACE label --overwrite dv $(echo $dv | sed 's/"//g')
 vm.cpaas.io/used-by=$VM_CLONE_NAME
 if [$? -ne 0]; then
 echo "update DV label failed"
 exit 1
 fi
done

pvcList=$(kubectl -n $NAMESPACE get vm $VM_CLONE_NAME -o
jsonpath='{.spec.template.spec.volumes}' | jq . | grep restore- | grep claimName | awk
'{print $2}')

for pvc in $pvcList; do
 kubectl -n $NAMESPACE label --overwrite pvc $(echo $pvc | sed 's/"//g')
 vm.cpaas.io/used-by=$VM_CLONE_NAME
 if [$? -ne 0]; then
 echo "update PVC label failed"
 exit 1
 fi
done
}

if [$# -ne 3]; then
 echo "error: parameters error"
 echo "Usage: ./vm-clone.sh NAMESPACE VM_NAME VM_CLONE_NAME"
 exit 1
fi

exec vm clone
vm_clone "$1" "$2" "$3"

```

4. Нажмите shift+:wq для сохранения файла.
5. Выполните следующую команду, чтобы добавить права на выполнение для файла vm-clone.sh.  
chmod +x vm-clone.sh
6. Выполните следующую команду для запуска скрипта.{#clone}  
./vm-clone.sh <NAMESPACE> <VM\_NAME> <VM\_CLONE\_NAME>

Описание параметров:

- **NAMESPACE:** Укажите namespace виртуальной машины, которую нужно клонировать, заменив часть <NAMESPACE> на этот namespace.
- **VM\_NAME:** Укажите имя виртуальной машины, которую нужно клонировать,

заменяв часть <VM\_NAME> на это имя.

- **VM\_CLONE\_NAME:** Укажите имя клонированной виртуальной машины, заменив часть <VM\_CLONE\_NAME> на это имя.

7. Когда появится сообщение, похожее на следующее, это означает, что клонирование завершено.

```
virtualmachineclone.clone.kubevirt.io/clone-k1 created
```

```
Create vmclone resource Succeeded
```

```
Waiting for vm clone completion
```

```
vm clone completion
```

```
datavolume.cdi.kubevirt.io/restore-e8ff0e7b-dc7e-4140-aec7-8556cfcf4533-rootfs
labeled
```

```
datavolume.cdi.kubevirt.io/restore-e8ff0e7b-dc7e-4140-aec7-8556cfcf4533-1 labeled
```

## Контроль сетевых запросов виртуальной машины через Network Policy

Решение платформы для виртуальных машин реализовано на базе open-source компонента KubeVirt, который фактически работает внутри Pod'ов. Используя функциональность Network Policies, можно контролировать входящие и исходящие запросы виртуальных машин.

1. Войдите в **Container Platform**.
2. В левой навигационной панели нажмите **Network > Network Policies**.
3. Нажмите **Create Network Policy**.
4. Настройте следующие параметры по необходимости.

| Параметр                  | Описание                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Association Method</b> | <ul style="list-style-type: none"><li>• <b>Compute Component:</b> Выберите целевой вычислительный компонент по необходимости; рекомендуется выбрать <b>All</b> в качестве целевого вычислительного компонента.</li><li>• <b>Label Selector:</b> Соответствие Pod'ов по меткам.</li></ul>                                                                                                   |
| <b>Direction</b>          | <ul style="list-style-type: none"><li>• <b>Ingress:</b> Запросы, направленные извне к Pod.</li><li>• <b>Egress:</b> Запросы, направленные из Pod во внешний мир; выберите этот вариант, если необходимо запретить виртуальной машине делать запросы к определённому внешнему адресу.</li></ul>                                                                                             |
| <b>Protocol</b>           | Выберите TCP или UDP.<br><b>Примечание:</b> При использовании доменных имён в виртуальной машине для запросов к внешним сервисам необходимо добавить белый список для протокола UDP, так как DNS использует UDP.<br>Форма не поддерживает настройку протокола ICMP; при включении правил белого списка протокол ICMP будет отключён, что приведёт к невозможности выполнять операции Ping. |
| <b>Access Ports</b>       | Укажите, трафик по каким портам разрешён для входящих или исходящих запросов. Если поле оставить пустым, по умолчанию разрешён трафик по всем портам.<br><b>Примечание:</b> Необходимо разрешить порты 1053 и 53 для протоколов UDP и TCP, чтобы разрешить исходящий DNS-трафик; иначе разрешение доменных имён не будет работать.                                                         |
| <b>Remote Type</b>        | Укажите разрешённый тип удалённого доступа. Варианты: вычислительный                                                                                                                                                                                                                                                                                                                       |



| Параметр              | Описание                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | компонент, namespace, IP-сегменты.                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Exclude Remote</b> | Если тип удалённого доступа — IP Segment, исключите указанный IP из белого списка (то есть запретите доступ). Один IP можно исключить, указав его в формате IP/32.<br><b>Примечание:</b> В это поле можно вводить только IP; если соответствующий IP доменного имени неизвестен, используйте команду <code>curl -vvv &lt;domain&gt;</code> , чтобы запросить домен и получить IP из возвращаемой информации. |

5. Нажмите **Create**.

## Хранилище

MWS-CP Virtualization с KubeVirt Storage предоставляет возможности постоянного хранения для виртуальных машин (VM) за счёт бесшовной интеграции с нативными для Kubernetes механизмами хранения. Он использует **PersistentVolumeClaim (PVC)** для хранения данных дисков VM и применяет **Container Storage Interface (CSI)** для интеграции с различными системами хранения. Кроме того, для инициализации данных дисков VM используется **Containerized Data Importer (CDI)**. Основываясь на этих компонентах, платформа расширяет функциональность управления дисками VM, обеспечивая комплексный контроль жизненного цикла.

### Преимущества

- Удобство использования  
Большинство операций с дисками VM можно легко выполнять через **Web UI**, что минимизирует необходимость владения CLI.
- Управление жизненным циклом дисков VM  
Настройка автоматического удаления дисков VM при завершении работы соответствующей VM.

### Управление виртуальными дисками

Дисковые накопители данных могут использоваться для удовлетворения требований бизнеса по сохранению данных.

#### Создание виртуального диска

Создайте **дисковый накопитель данных** для виртуальной машины. За один раз можно добавить только **один** виртуальный диск; если требуется несколько дисков, повторите эту операцию.

**Примечание:** Виртуальные диски можно монтировать онлайн, когда виртуальная машина находится в состоянии **запущена**.

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Disk**.
3. Нажмите **Create Virtual Disk**.
4. Настройте параметры согласно следующим инструкциям.

| Параметр           | Описание                                                                                                                     |
|--------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Volume Mode</b> | <ul style="list-style-type: none"> <li>• <b>File System:</b> Монтирование диска с доступом к файловой директории.</li> </ul> |

|                       |                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <ul style="list-style-type: none"> <li>• <b>Block Device:</b> Монтирование диска как блочного устройства.</li> </ul>                                                                                                                                                                                                                                                                             |
| <b>Storage Class</b>  | <p>Платформа поддерживает диски виртуальных машин, автоматически создавая и управляя persistent volume claims. Необходимо указать класс хранения, требуемый для динамического создания persistent volume claims.</p> <p>Разные классы хранения поддерживают разные режимы томов. Если для выбранного режима тома нет доступных классов хранения, обратитесь к администратору для добавления.</p> |
| <b>Delete with VM</b> | Если включено, данные диска будут удалены при удалении виртуальной машины.                                                                                                                                                                                                                                                                                                                       |
| <b>Mount</b>          | <ul style="list-style-type: none"> <li>• <b>Do Not Mount:</b> Только создать виртуальный диск; его можно смонтировать позже при необходимости.</li> <li>• <b>Mount to VM:</b> Выберите целевую виртуальную машину, к которой необходимо примонтировать виртуальный диск.</li> </ul>                                                                                                              |

5. Нажмите **Create**.

#### Монтирование виртуального диска

Примонтируйте **дисковый накопитель данных** к виртуальной машине, присоединив уже созданный виртуальный диск к целевой виртуальной машине.

**Примечание:** Виртуальные диски можно монтировать онлайн, когда виртуальная машина находится в состоянии **запущена**.

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Disk**.
3. Нажмите : > **Mount** рядом с виртуальным диском, который нужно примонтировать.
4. Выберите целевую виртуальную машину и нажмите **Mount**.

#### Расширение виртуального диска

Расширьте **системный диск** и **дисковый накопитель данных**, уже примонтированные к виртуальной машине.

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Machine**.
3. Нажмите на имя виртуальной машины, чтобы перейти на страницу **Details**.
4. В разделе **Virtual Disk** найдите диск, который нужно расширить, и нажмите : > **Expand**.
5. Введите новый размер и нажмите **Expand**.

#### Отсоединение виртуального диска

Отсоедините **дисковый накопитель данных** от виртуальной машины; диски можно отсоединить только у виртуальных машин в состоянии **stopped**.

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Disk**.
3. Нажмите : > **Unmount** рядом с виртуальным диском, который нужно отсоединить, и подтвердите действие.

#### Удаление виртуального диска

Удаление поддерживается только для отсоединенных виртуальных дисков.

**Примечание:** Системные диски удалять нельзя.

1. Перейдите в **Container Platform**.
2. В левой навигационной панели нажмите **Virtualization > Virtual Disk**.
3. Нажмите **: > Delete** рядом с виртуальным диском, который нужно удалить, и подтвердите действие.

## Резервное копирование и восстановление

**MWS-CP Virtualization With Kubevirt** предоставляет возможности создания снимков виртуальных машин, позволяя пользователям выполнять резервное копирование и восстановление VM с помощью снимков.

### Сценарии использования

- Восстановление после сбоев и откат при ошибках  
Когда виртуальная машина теряет данные из-за аппаратных сбоев, ошибок пользователя (например, случайного удаления файлов) или вредоносных атак (например, программ-вымогателей), снимки служат последней линией защиты для восстановления работы.

### Ограничения использования

Для создания снимка необходимо сначала остановить виртуальную машину.

PVC (Persistent Volume Claim), используемый диском виртуальной машины, должен быть настроен с режимом совместного доступа на нескольких узлах.

### Использование снимков

Снимок виртуальной машины сохраняет текущее состояние виртуальной машины и может быть использован для восстановления виртуальной машины в этом состоянии в случае неожиданного сбоя.

**Примечание:** Если в кластере присутствует несколько однотипных типов хранилищ, например, подключено несколько различных источников Ceph RBD, функциональность снимков дисков может работать некорректно при использовании виртуальной машиной такого хранилища.

### Требования

**Volume Snapshot** был развернут администратором в управлении платформой.

Снимки виртуальной машины основаны на volume snapshots. Убедитесь, что хотя бы один диск привязан к storage class, поддерживающему volume snapshots, например, встроенному хранилищу CephFS.

- Поддерживаются только офлайн-снимки виртуальной машины. Пожалуйста, сначала остановите виртуальную машину перед созданием или откатом к снимку.

### Создание снимка

В снимок виртуальной машины включается: настройки виртуальной машины и состояние дисков, поддерживающих volume snapshots.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Virtualization > Virtual Machines**.
3. Найдите виртуальную машину и нажмите : > **Create Snapshot**.
4. Заполните описание снимка. Описание поможет задокументировать текущее состояние виртуальной машины, например Initial Installation, Before Application Upgrade.
5. Нажмите **Create**. Время создания снимка зависит от сетевых условий и нагрузки, пожалуйста, подождите.
6. Проверьте статус снимка.
  - Когда статус снимка изменится на Ready, это означает успешное создание.
  - Если снимок долго остается в статусе Not Ready, нажмите  > Просмотрите причины и устраните неполадки, затем создайте снимок заново.

### **Откат к снимку**

Откатит настройки виртуальной машины и диски, поддерживающие volume snapshots, к состоянию на момент создания снимка. Например, диски, добавленные после создания снимка, будут удалены; изменённые данные дисков будут восстановлены.

**Примечание:** Если диски привязаны к storage class, поддерживающему механизм LVM (например, TaroLVM), пожалуйста, подтвердите у администратора, что политика восстановления для этого storage class установлена в Retain (reclaimPolicy: Retain), чтобы корректно использовать функцию отката снимка.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Virtualization > Virtual Machines**.
3. Нажмите на **Virtual Machine Name**.
4. Во вкладке **Snapshots** найдите нужный снимок и нажмите : > **Rollback**.
5. Ознакомьтесь с информацией в подсказке и нажмите **Rollback** после подтверждения правильности данных.

**Примечание:** Операция отката не может быть прервана или отменена, будьте внимательны.

6. Нажмите на имя снимка, чтобы проверить в разделе «Snapshot Rollback Records», завершён ли откат. Время отката зависит от сетевых условий и нагрузки, пожалуйста, подождите.

### **Описание**

- Если откат не удался, состояние виртуальной машины останется без изменений. Вы можете запустить виртуальную машину в обычном режиме или попытаться выполнить откат снимка снова.

- Если виртуальная машина была запущена во время процесса отката, она вернётся в состояние до остановки, а при следующей остановке продолжит откат к состоянию на момент создания снимка.
- Чтобы избежать конфликтов в операциях, убедитесь, что последний откат завершён, прежде чем выполнять другие действия с этой виртуальной машиной.

### **Удаление снимка**

Удаляйте ненужные снимки виртуальной машины для освобождения ресурсов диска.

**Примечание:** При удалении снимка виртуальной машины, к которому был выполнен откат, если для диска виртуальной машины требуется копирование данных на основе снимка (например, TopoLVM), необходимо дождаться запуска виртуальной машины на основе версии после отката перед удалением, иначе виртуальная машина не сможет запуститься.

1. Зайдите в **Container Platform**.
2. В левой навигационной панели выберите **Virtualization > Virtual Machines**.
3. Нажмите на **Virtual Machine Name**.
4. Во вкладке **Snapshots** найдите нужный снимок и нажмите : > **Delete**.
5. Ознакомьтесь с информацией в подсказке и нажмите **Delete** после подтверждения правильности данных.

## Приложение А

Скрипт `install-jaeger.sh` для развёртывания экземпляра Jaeger:

```
#!/bin/bash

set -euo pipefail

get arg
while ["$#" -gt 0]; do
 case $1 in
 --es-url=*)
 ES_URL="${1#*=}"
 ;;
 --es-user-base64=*)
 ES_USER_BASE64="${1#*=}"
 ;;
 --es-pass-base64=*)
 ES_PASS_BASE64="${1#*=}"
 ;;
 *)
 echo "unknown argument: $1"
 exit 1
 ;;
 esac
 shift
done

print arg
echo "ES_URL: $ES_URL"
echo "ES_USER_BASE64: $ES_USER_BASE64"
echo "ES_PASS_BASE64: $ES_PASS_BASE64"

get global-info from ConfigMap
CLUSTER_NAME=$(kubectl get configmap global-info -n kube-public -o
jsonpath='{.data.clusterName}')
echo "CLUSTER_NAME: ${CLUSTER_NAME}"
ISSUER_URL=$(kubectl get configmap global-info -n kube-public -o
jsonpath='{.data.oidcIssuer}')
CLIENT_ID=$(kubectl get configmap global-info -n kube-public -o
jsonpath='{.data.oidcClientID}')
CLIENT_SECRET=$(kubectl get configmap global-info -n kube-public -o
jsonpath='{.data.oidcClientSecret}')
CLIENT_SECRET_BASE64=$(echo -n "${CLIENT_SECRET}" | base64 -w0)
PLATFORM_URL=$(kubectl get configmap global-info -n kube-public -o
jsonpath='{.data.platformURL}')
echo "PLATFORM_URL: ${PLATFORM_URL}"

TARGET_NAMESPACE="cpaas-system"
JAEGER_BASEPATH="clusters/${CLUSTER_NAME}/acp/jaeger"
```

```

_apply_resource() {
 if [-z "$1"]; then
 echo "Usage: _apply_resource <yaml_content>"
 return 1
 fi

 local yaml_content="$1"
 echo "$yaml_content" | kubectl apply -f -
}

_install_configmap() {
 local yaml_content=$(cat <<EOF
apiVersion: v1
data:
 OAUTH2_PROXY_CLIENT_ID: $CLIENT_ID
 OAUTH2_PROXY_COOKIE_SECURE: "false"
 OAUTH2_PROXY_EMAIL_DOMAINS: "*"
 OAUTH2_PROXY_HTTP_ADDRESS: 0.0.0.0:4180
 OAUTH2_PROXY_INSECURE_OIDC_ALLOW_UNVERIFIED_EMAIL: "true"
 OAUTH2_PROXY_OIDC_ISSUER_URL: $ISSUER_URL
 OAUTH2_PROXY_PROVIDER: oidc
 OAUTH2_PROXY_PROXY_PREFIX: /$JAEGER_BASEPATH/oauth2
 OAUTH2_PROXY_REDIRECT_URL:
$PLATFORM_URL/$JAEGER_BASEPATH/oauth2/callback
 OAUTH2_PROXY_SCOPE: openid profile email groups ext
 OAUTH2_PROXY_SKIP_JWT_BEARER_TOKENS: "true"
 OAUTH2_PROXY_SKIP_PROVIDER_BUTTON: "true"
 OAUTH2_PROXY_SSL_INSECURE_SKIP_VERIFY: "true"
 OAUTH2_PROXY_UPSTREAMS: http://127.0.0.1:16686
kind: ConfigMap
metadata:
 name: jaeger-oauth2-proxy
 namespace: $TARGET_NAMESPACE
EOF
)
 _apply_resource "$yaml_content"
}

_install_secret() {
 local yaml_content=$(cat <<EOF
apiVersion: v1
data:
 OAUTH2_PROXY_CLIENT_SECRET: $CLIENT_SECRET_BASE64
 OAUTH2_PROXY_COOKIE_SECRET: $CLIENT_SECRET_BASE64
kind: Secret
metadata:
 name: jaeger-oauth2-proxy
 namespace: $TARGET_NAMESPACE

```

```

type: Opaque

apiVersion: v1
data:
 ES_PASSWORD: $ES_PASS_BASE64
 ES_USERNAME: $ES_USER_BASE64
kind: Secret
metadata:
 name: jaeger-elasticsearch-basic-auth
 namespace: $TARGET_NAMESPACE
type: Opaque
EOF
)
 _apply_resource "$yaml_content"
}

_install_sa() {
 local yaml_content=$(cat <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
 name: jaeger-prod-acp
 namespace: $TARGET_NAMESPACE

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
 name: jaeger-prod-acp
 namespace: $TARGET_NAMESPACE
rules:
- apiGroups: [""]
 resources: ["secrets"]
 verbs: ["get"]

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name: jaeger-prod-acp
 namespace: $TARGET_NAMESPACE
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: Role
 name: jaeger-prod-acp
subjects:
- kind: ServiceAccount
 name: jaeger-prod-acp
 namespace: $TARGET_NAMESPACE
EOF
)

```



```

 _apply_resource "$yaml_content"
 }

 _install_jaeger() {
 local yaml_content=$(cat <<EOF
apiVersion: jaegertracing.io/v1
kind: Jaeger
metadata:
 name: jaeger-prod
 namespace: $TARGET_NAMESPACE
spec:
 collector:
 affinity:
 podAntiAffinity:
 requiredDuringSchedulingIgnoredDuringExecution:
 - labelSelector:
 matchLabels:
 app.kubernetes.io/name: jaeger-prod-collector
 topologyKey: kubernetes.io/hostname
 replicas: 1
 resources:
 limits:
 cpu: "2"
 memory: 512Mi
 requests:
 cpu: 250m
 memory: 256Mi
 imagePullSecrets:
 - name: global-registry-auth
 ingress:
 enabled: false
 labels:
 service_name: jaeger
 query:
 affinity:
 podAntiAffinity:
 requiredDuringSchedulingIgnoredDuringExecution:
 - labelSelector:
 matchLabels:
 app.kubernetes.io/name: jaeger-prod-query
 topologyKey: kubernetes.io/hostname
 annotations:
 oauth2-proxy.github.io/image: ""
 oauth2-proxy.github.io/inject: "true"
 oauth2-proxy.github.io/oidc-configmap: jaeger-oauth2-proxy
 oauth2-proxy.github.io/oidc-secret: jaeger-oauth2-proxy
 oauth2-proxy.github.io/proxyCPULimit: 100m
 oauth2-proxy.github.io/proxyMemoryLimit: 128Mi
 options:

```

```
 query:
 base-path: /$JAEGER_BASEPATH
 replicas: 1
 resources:
 limits:
 cpu: "1"
 memory: 512Mi
 requests:
 cpu: 250m
 memory: 256Mi
 resources:
 limits:
 cpu: 100m
 memory: 300Mi
 requests:
 cpu: 100m
 memory: 300Mi
 sampling:
 options: {}
 serviceAccount: jaeger-prod-acp
 storage:
 dependencies:
 enabled: false
 resources: {}
 schedule: 55 23 * * *
 elasticsearch:
 name: elasticsearch
 nodeCount: 3
 redundancyPolicy: SingleRedundancy
 esIndexCleaner:
 enabled: true
 numberOfDays: 7
 resources: {}
 schedule: 55 23 * * *
 esRollover:
 resources: {}
 schedule: 0 0 * * *
 options:
 es.asm.cname: jaeger-elasticsearch-basic-auth
 es.asm.cnamespace: $TARGET_NAMESPACE
 es.index-prefix: acp-tracing-$CLUSTER_NAME
 es.max-span-age: 168h0m0s
 es.server-urls: $ES_URL
 es.tls.enabled: true
 es.tls.skip-host-verify: true
 secretName: ""
 type: elasticsearch
 strategy: production
 tolerations:
```

```

- effect: NoSchedule
 key: node-role.kubernetes.io/control-plane
 operator: Exists
- effect: NoSchedule
 key: node-role.kubernetes.io/master
 operator: Exists
ui:
 options:
 dependencies:
 menuEnabled: false
EOF
)
_apply_resource "$yaml_content"
}

_install_pod_monitor() {
 local yaml_content=$(cat <<EOF
apiVersion: monitoring.coreos.com/v1
kind: PodMonitor
metadata:
 labels:
 monitoring: pods
 prometheus: kube-prometheus
 name: jaeger-monitor
 namespace: $TARGET_NAMESPACE
spec:
 jobLabel: app.kubernetes.io/name
 namespaceSelector:
 matchNames:
 - $TARGET_NAMESPACE
 podMetricsEndpoints:
 - interval: 60s
 path: /metrics
 port: admin-http
 selector:
 matchLabels:
 app.kubernetes.io/instance: jaeger-prod
EOF
)
_apply_resource "$yaml_content"
}

_install_ingress() {
 local alb_annotation=""
 if [["$CLUSTER_NAME" == "global"]]; then
 alb_annotation=$(cat <<EOF
alb.ingress.cpaas.io/rewrite-request: |
 {"headers_var":{"Authorization":"cookie_cpaas_id_token"}}
EOF
)

```

```

)
fi
local ingress_class=""
if [["$CLUSTER_NAME" != "global"]]; then
 ingress_class=" ingressClassName: cpaas-system"
fi

local yaml_content=$(cat <<EOF
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
 name: jaeger-query
 namespace: $TARGET_NAMESPACE
 annotations:
 nginx.ingress.kubernetes.io/enable-cors: "true"
$alb_annotation
spec:
 $ingress_class
 rules:
 - http:
 paths:
 - path: /$JAEGER_BASEPATH
 pathType: ImplementationSpecific
 backend:
 service:
 name: jaeger-prod-query
 port:
 number: 4180
EOF
)
_apply_resource "$yaml_content"
}

final install
_install_configmap
_install_secret
_install_sa
_install_jaeger
_install_pod_monitor
_install_ingress

echo "Jaeger installation completed"

```

## Приложение Б

Скрипт install-otel.sh для развёртывания экземпляров OpenTelemetry:

```
#!/bin/bash

set -euo pipefail

TARGET_NAMESPACE="cpaas-system"

get global-info from ConfigMap
CLUSTER_NAME=$(kubectl get configmap global-info -n kube-public -o
jsonpath='{.data.clusterName}')
echo "CLUSTER_NAME: ${CLUSTER_NAME}"

_apply_resource() {
 if [-z "$1"]; then
 echo "Usage: _apply_resource <yaml_content>"
 return 1
 fi

 local yaml_content="$1"
 echo "$yaml_content" | kubectl apply -f -
}

_install_rbac() {
 local yaml_content=$(cat <<EOF
apiVersion: v1
imagePullSecrets:
 - name: global-registry-auth
kind: ServiceAccount
metadata:
 name: otel-collector
 namespace: $TARGET_NAMESPACE

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
 name: otel-collector:cpaas-system:cluster-admin
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: cluster-admin
subjects:
 - kind: ServiceAccount
 name: otel-collector
 namespace: $TARGET_NAMESPACE
EOF
)
 _apply_resource "$yaml_content"
}
```

```

}

_install_otel_collector() {
 local yml_content=$(cat <<EOF
apiVersion: opentelemetry.io/v1beta1
kind: OpenTelemetryCollector
metadata:
 name: otel
 namespace: $TARGET_NAMESPACE
spec:
 affinity:
 podAntiAffinity:
 requiredDuringSchedulingIgnoredDuringExecution:
 - labelSelector:
 matchLabels:
 app: otel-collector
 topologyKey: kubernetes.io/hostname
 config:
 exporters:
 debug: {}
 otlp:
 balancer_name: round_robin
 endpoint: dns:///jaeger-prod-collector-headless.$TARGET_NAMESPACE:4317
 tls:
 insecure: true
 prometheus:
 endpoint: 0.0.0.0:8889
 extensions:
 health_check:
 endpoint: 0.0.0.0:13133
 processors:
 batch: {}
 filter/metric_apis:
 metrics:
 datapoint:
 - attributes["http.route"] == "/actuator/health" or attributes["uri"] ==
"/actuator/health"
 - attributes["http.route"] == "/actuator/prometheus" or attributes["uri"] ==
"/actuator/prometheus"
 transform:
 metric_statements:
 - context: datapoint
 statements:
 - delete_key(attributes, "inner.client.ms.name")
 - delete_key(attributes, "inner.client.ms.namespace")
 - delete_key(attributes, "inner.client.cluster.name")
 - delete_key(attributes, "inner.client.env.type")
 - set(attributes["namespace"], resource.attributes["k8s.namespace.name"])
 - set(attributes["container"], resource.attributes["k8s.container.name"])

```

- set(attributes["service\_name"], resource.attributes["service.name"])
- set(attributes["pod"], resource.attributes["k8s.pod.name"])

memory\_limiter:

- check\_interval: 5s
- limit\_percentage: 85
- spike\_limit\_percentage: 25

receivers:

otlp:

protocols:

- grpc:
  - endpoint: 0.0.0.0:4317
- http:
  - endpoint: 0.0.0.0:4318

service:

extensions:

- health\_check

pipelines:

metrics:

exporters:

- debug
- prometheus

processors:

- memory\_limiter
- filter/metric\_apis
- transform
- batch

receivers:

- otlp

traces:

exporters:

- debug
- otlp

processors:

- memory\_limiter
- batch

receivers:

- otlp

telemetry:

logs:

- level: info

metrics:

- address: 0.0.0.0:8888
- level: detailed

managementState: managed

mode: deployment

replicas: 1

resources:

limits:

- cpu: "2"

```

 memory: 1Gi
 requests:
 cpu: 250m
 memory: 512Mi
 securityContext:
 readOnlyRootFilesystem: true
 runAsNonRoot: true
 serviceAccount: otel-collector
 tolerations:
 - effect: NoSchedule
 key: node-role.kubernetes.io/control-plane
 operator: Exists
 - effect: NoSchedule
 key: node-role.kubernetes.io/master
 operator: Exists
 upgradeStrategy: automatic
 EOF
)
 _apply_resource "$yaml_content"
}

_install_instrumentation() {
 local yaml_content=$(cat <<EOF
apiVersion: opentelemetry.io/v1alpha1
kind: Instrumentation
metadata:
 name: acp-common-java
 namespace: $TARGET_NAMESPACE
spec:
 env:
 - name: SERVICE_CLUSTER
 value: "$CLUSTER_NAME"
 - name: OTEL_TRACES_EXPORTER
 value: otel
 - name: OTEL_METRICS_EXPORTER
 value: otel
 - name: OTEL_EXPORTER_OTLP_ENDPOINT
 value: http://otel-collector.$TARGET_NAMESPACE:4317
 - name: OTEL_SERVICE_NAME
 value: \$(SERVICE_NAME).\$(SERVICE_NAMESPACE)
 - name: OTEL_RESOURCE_ATTRIBUTES
 value:
service.namespace=\$(SERVICE_NAMESPACE),cluster.name=\$(SERVICE_CLUSTER)
 sampler:
 type: parentbased_traceidratio
 argument: "1"
 EOF
)
 _apply_resource "$yaml_content"

```



```

}

_install_service_monitor() {
 local yml_content=$(cat <<EOF
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
 labels:
 monitoring: services
 prometheus: kube-prometheus
 name: otel-collector-monitoring
 namespace: $TARGET_NAMESPACE
spec:
 endpoints:
 - interval: 60s
 path: /metrics
 port: monitoring
 jobLabel: app.kubernetes.io/name
 namespaceSelector:
 matchNames:
 - $TARGET_NAMESPACE
 selector:
 matchLabels:
 app.kubernetes.io/instance: $TARGET_NAMESPACE.otel
 operator.opentelemetry.io/collector-service-type: monitoring

apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
 labels:
 monitoring: services
 prometheus: kube-prometheus
 name: otel-collector
 namespace: $TARGET_NAMESPACE
spec:
 endpoints:
 - honorLabels: true
 interval: 60s
 path: /metrics
 port: prometheus
 jobLabel: app.kubernetes.io/name
 namespaceSelector:
 matchNames:
 - $TARGET_NAMESPACE
 selector:
 matchLabels:
 app.kubernetes.io/instance: $TARGET_NAMESPACE.otel
 operator.opentelemetry.io/collector-service-type: base
EOF

```

```
)
_apply_resource "$yaml_content"
}

_install_rbac
_install_otel_collector
_install_instrumentation
_install_service_monitor

echo "OpenTelemetry installation completed"
```