

# ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК MWS CONTAINER PLATFORM

## 1. Наименование и назначение Системы

Наименование Системы – MWS Container Platform

Назначение Системы:

MWS Container Platform — это платформа для управления контейнерной инфраструктурой на базе Kubernetes.

Она автоматизирует процессы создания, масштабирования, обновления и удаления Kubernetes-кластеров, а также развёртывания и сопровождения приложений. Поддерживается работа в любых типах инфраструктуры — на физических серверах (bare metal), в виртуальных средах, в частных или публичных облаках, а также на edge-устройствах.

Платформа обеспечивает централизованное управление множеством кластеров через единую панель и включает все необходимые инструменты для промышленной эксплуатации: средства наблюдаемости (observability), обеспечения безопасности, построения сетевой инфраструктуры, управления хранилищами (локальными, блочными и объектными), резервного копирования и восстановления. Поддерживается разграничение прав доступа (RBAC), интеграция с корпоративными системами аутентификации и централизованный аудит действий пользователей и системных компонентов.

В состав платформы входят модули для автоматизации CI/CD, реализации GitOps-подхода, Service Mesh на базе Istio, управления виртуальными машинами в Kubernetes-кластерах, а также средства для автоматизированного управления жизненным циклом баз данных и брокеров сообщений. Кроме того, в платформу встроены инструменты для управления моделями машинного обучения и большими языковыми моделями, включая их развёртывание, масштабирование и мониторинг в Kubernetes-кластерах.

Все компоненты интегрированы в единое решение, что упрощает эксплуатацию, снижает операционные затраты и ускоряет цифровую трансформацию.

## 2. Система представляет собой: платформу контейнеризации

## 3. Технические характеристики устройства для установки Системы:

3.1. Функциональные возможности Системы: Контейнерная платформа, Виртуализация, Хранилище, Сеть, DevOps, AI, Application Services, Service Mesh, Data Services, Платформа интеграции сервисов

3.2. Категории пользователей Системы: Разработчики, команда по эксплуатации и техническому обслуживанию (Ops / SRE), администраторы платформы.

3.3. Функциональные возможности Системы для определенных категорий пользователей:

Разработчики имеют возможность:

- пользоваться контейнерной платформой для развёртывания и отладки своих приложений,
- использовать инструменты DevOps для автоматизации сборок и развёртываний,
- работать с прикладными сервисами, такими как MySQL, PostgreSQL, Redis, Kafka, RabbitMQ

- применять Service Mesh для расширенного управления трафиком и тестирования,
- обучать, настраивать и выводить модели искусственного интеллекта,
- при работе с виртуализацией развертывать свои приложения на базе виртуальных машин и управлять ими.

Команда по эксплуатации и техническому обслуживанию (Ops / SRE) имеет возможность:

- использовать функционал контейнерной платформы и Service Mesh для мониторинга развертывания приложений и устранения неполадок,
- применять в своей работе сбор логов, мониторинг для анализа производительности и обнаружения проблем,
- управлять хранилищем и сетевыми ресурсами,
- использовать DevOps для автоматизации.

Администраторам доступен весь функционал платформы, они имеют возможность:

- осуществлять контроль над жизненным циклом контейнерной платформы и квотами ресурсов,
- осуществлять мультикластерное управление, обеспечивая согласованность политик и развертываний во всей инфраструктуре,
- настраивать хранилище и сеть, управлять ими на базовом уровне,
- участвовать в настройке и обслуживании инструментов DevOps и CI / CD,
- управлять определенным пространством имен AI, общедоступными моделями AI и аппаратными ускорителями платформы,
- контролировать инфраструктуру виртуализации и доступность прикладных служб.

### 3.4. Состав программных средств

Система состоит из следующих логических модулей:

- Веб-консоль: веб-консоль контейнерной платформы предоставляет графический пользовательский интерфейс для визуализации данных и выполнения задач администрирования, управления, мониторинга и устранения неполадок.
- Управление кластером: отвечает за создание и управление различными кластерами Kubernetes.
- Многопользовательское управление: обеспечивает надежную изоляцию проектов и пространств имен, обеспечивая безопасную и изолированную среду для различных групп пользователей.
- Управление приложениями: позволяет управлять рабочими нагрузками Kubernetes, собственными приложениями, приложениями OAM (Open Application Model), Helm Charts, GitOps. Обеспечивает гибкое и унифицированное развертывание приложений и управление жизненным циклом.
- Управление хранилищем: предоставляет решения для постоянного хранения данных, необходимые для приложений с возможностью отслеживания состояния. Поддерживаются различные типы хранилища (блочное, файловое, объектное хранилище), предлагая динамическое расширение, распределение (on-demand allocation) по требованию и функции управления данными, обеспечивающие надежность, такие как резервное копирование, восстановление и аварийное восстановление.
- Управление сетью: обеспечивает возможность организации сетевых подключений и политик, включая несколько CNI, контроль сетевых политик, управление Ingress и Egress.

- Мониторинг: предназначен для наблюдения за работоспособностью системы и приложений. Модуль включает в себя систему сбора и анализа логов, систему мониторинга показателей, систему управления оповещениями (алертинг и уведомления).
- Служба приложений: предоставляет полный набор инструментов и сервисов для облачной среды, которые позволяют обеспечивать высокую доступность, безопасность данных и оптимизировать работу разработчиков.
- Каталог приложений: представлен расширяемым маркетплейсом, обеспечивающим бесперебойное развертывание, обновление, настройку и управление жизненным циклом различных компонентов, повышающими гибкость и расширяемость платформы.
- Управление виртуализацией: позволяет расширить функционал платформы, предоставляя возможность создавать и управлять виртуальными машинами наряду с контейнерами, используя KubeVirt.
- DevOps: представляет собой модуль, интегрированный с популярными инструментами, такими как Harbor, GitLab, Sonarqube, Nexus и другими, использует Tekton для CI / CD. Обеспечивает гибкие, автоматизированные рабочие процессы доставки программного обеспечения и надежное управление артефактами.
- Искусственный интеллект (AI): предоставляет комплексное управление моделями, обеспечивает инференс моделей на базе бессерверных вычислений с помощью KServe и Knative, а также контейнерную среду web IDE для разработки моделей с использованием Kubeflow Notebooks.
- Service Mesh: обеспечивает комплексное управление микросервисами. Предоставляет возможность управления трафиком, надежную защиту (mTLS) и мониторинг распределенных приложений, обеспечивая бесперебойное и устойчивое взаимодействие между сервисами.
- Data Services: включает в себя готовые сервисы данных (PostgreSQL, MySQL, Redis, Kafka, RabbitMQ), разворачивающиеся в несколько кликов с возможностью выбора версии и типа конфигурации, настройкой резервного копирования.
- Платформа интеграции сервисов состоит из двух частей.
  - Платформа самообслуживания PaaS: включает в себя функцию управления приложениями (создание, просмотр, удаление), каталог служб отображает доступные компоненты и службы, экземпляр службы используется для управления экземплярами служб, поддерживая операции модификации и удаления, обеспечивает мониторинг состояния и контроль доступа к экземплярам. Функция формы запроса используется для отправки запросов на услуги или ресурсы, а пользователи могут отслеживать ход выполнения запроса и обратную связь по ним. Функция создания заявок используется для отправки и отслеживания проблем или запросов на выполнение задач, поддерживает описание деталей проблемы, установку приоритетов, загрузку вложений.
  - Единая платформа управления PaaS: включает в себя управление заявками (тикетами), используется для управления получением заявок и обработки отправленных пользователем вопросов или запросов на выполнение задач, управление средой для эффективной группировки ресурсов, при этом пользователи могут создавать, просматривать, обновлять и удалять среды в соответствии с потребностями. Управление доступом к сторонним компонентам позволяет управлять доступом к компонентам сторонних производителей, отправлять им заявки на загрузку компонентов. Управление платформой автоматически завершит загрузку после проверки и согласования, после согласования компоненты можно использовать в платформе, а пользователи могут подать заявку на удаления поле того, как компонент потеряет актуальность. Управление конфигурацией каталога служб позволяет выполнять

операции создания, просмотра, обновления и удаления категорий каталога служб. Пользователи, в свою очередь, могут эффективно управлять категоризованной информацией

Система поддерживает современные браузеры: Mozilla Firefox 134+, Google Chrome 131+, Yandex

3.5. Для функционирования Системы требуются:

— Аппаратные ресурсы:

- Управляющий слой (Control Plane): не менее 3 физических или виртуальных машин с не менее чем 8 ядрами центрального процессора, 16 ГБ оперативной памяти и 500 ГБ высокоскоростного SSD-накопителя. Эти спецификации предназначены для базового развертывания и будут расширяться в зависимости от количества управляемых кластеров и сложности рабочих нагрузок.
- Рабочие узлы: общее количество и спецификации рабочих узлов напрямую зависят от рабочей нагрузки приложения и требований к ресурсам. Рекомендуем использовать не менее 2 физических или виртуальных машин с минимум 12 ядрами процессора, 24 ГБ оперативной памяти и 200 ГБ высокоскоростного SSD-накопителя на узел.

— Операционная система: MWS Container Platform предназначена для работы в дистрибутивах Linux. Поддерживаемые операционные системы включают Astra Linux 1.7.6, Ubuntu Server 24.04 LTS и другие.

— Сетевое подключение: важно наличие стабильного сетевого соединения с высокой пропускной способностью и низкой задержкой, обеспечивающее бесперебойную связь между всеми компонентами кластера в разных кластерах. Необходимо открыть определенные сетевые порты внутри кластеров и за их пределами для различных служебных коммуникаций.

— IP-адрес и доменное имя: необходим виртуальный IP (VIP). Он служит единой точкой входа в веб-консоль, обеспечивая высокую доступность и упрощенный доступ. Кроме того, доменное имя может потребоваться для удобного доступа и брендинга. Если используется доменное имя, необходимы сертификаты SSL / TLS для обеспечения безопасной связи по протоколу HTTPS, шифрования передаваемых данных и укрепления доверия пользователей.

3.6. Язык программирования

При создании модулей MWS Container Platform использовались следующие языки программирования: C, Java (OpenJDK), Golang, Javascript (Angular), Python.