

Описание функциональных характеристик  
DevKube - для управления жизненным циклом  
контейнерных приложений

ООО " Аплана. Центр Разработки ", г. Москва

2025 г.

## ОГЛАВЛЕНИЕ

Перечень сокращений и терминов .....	3
1. Введение .....	5
2. Описание программы.....	6
3. Цели применения программы .....	7
4. Описание функциональных характеристик.....	8
4.1. Управление средой Kubernetes .....	8
4.2. Управление конвейерами .....	8
4.3. Шаблоны конвейеров .....	8
4.4. Встроенные интеграции с открытым исходным кодом .....	8
4.5. Интеграция инструментов сторонних производителей .....	9
4.6. Панель показателей и мониторинга .....	9
4.7. Управление несколькими проектами .....	10
5. Требования и рекомендации к IT-инфраструктуре .....	11
5.1. Программное обеспечение .....	11
5.2. Аппаратные требования .....	11
5.3. Сетевые и доменные ресурсы .....	12
5.4. Протоколы и интерфейсы .....	12
6. Рекомендуемые этапы для внедрения платформы .....	13

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ТЕРМИНОВ

Сокращение/термин	Полное наименование
ArgoCD	Инструмент для непрерывной доставки в Kubernetes
AWS (Amazon Web Services)	Облачная платформа Amazon
AWS Secrets Manager	Сервис управления секретами AWS
Azure	Облачная платформа Microsoft
CI/CD (Continuous Integration/Continuous Delivery)	Практика автоматизации процессов интеграции кода, тестирования и развертывания
Clair	Инструмент для анализа уязвимостей контейнеров
Datadog	Платформа для мониторинга и аналитики
DevOps (Development and Operations)	Методология, объединяющая разработку и эксплуатацию программного обеспечения
GCP (Google Cloud Platform)	Облачная платформа Google
GitHub Actions	Инструмент для автоматизации workflows в GitHub
Grafana	Инструмент визуализации метрик
HashiCorp Vault	Инструмент для управления секретами
Jaeger	Система трассировки для мониторинга распределенных систем
Kubernetes	Платформа для оркестрации контейнеризированных приложений
LDAP (Lightweight Directory Access Protocol)	Протокол для доступа к службам каталогов
OAuth2 (Open Authorization 2.0)	Протокол авторизации

<b>Сокращение/термин</b>	<b>Полное наименование</b>
OIDC (OpenID Connect)	Слой идентификации поверх OAuth2
Prometheus	Система мониторинга и оповещения
RBAC (Role-Based Access Control)	Управление доступом на основе ролей
Tekton	Платформа для создания CI/CD-конвейеров
Trivy	Сканер уязвимостей для контейнеров

# 1. ВВЕДЕНИЕ

Документ описывает платформу DevKube и содержит рекомендации и требования, необходимые для развертывания и ввода в эксплуатацию платформы.

Документ предназначен для ознакомления с функциональными возможностями программы специалистами потенциальных пользователей.

## 2. ОПИСАНИЕ ПРОГРАММЫ

DevKube — это корпоративная облачно-нативное программное обеспечение для управления контейнерной инфраструктурой, построенное на базе Kubernetes.

DevKube обеспечивает сквозное управление жизненным циклом приложений, включая развертывание, обновление, мониторинг, масштабирование и безопасность в гибридных и мультиоблачных средах.

DevKube предоставляет унифицированную панель управления, инструменты автоматизации DevOps, централизованное управление кластерами, а также встроенные средства обеспечения отказоустойчивости, аудита, хранения и наблюдаемости.

DevKube обеспечивает работу с несколькими проектами в единой инфраструктуре с полным разделением прав, ресурсов и изоляцией данных, адаптируется под различные сценарии эксплуатации — от разработки до эксплуатации в продуктивной среде, совместима с современными требованиями по информационной безопасности.

### 3. ЦЕЛИ ПРИМЕНЕНИЯ ПРОГРАММЫ

Программа DevKube предназначена для обеспечения комплексного управления контейнеризованной ИТ-инфраструктурой предприятия, построенной на Kubernetes. Платформа ориентирована на автоматизацию и упрощение ключевых бизнес-процессов, связанных с разработкой, поставкой и эксплуатацией современных микросервисных и облачных приложений.

Затрагиваемые бизнес-процессы:

- Разработка, тестирование и CI/CD-поставки программных продуктов;
- Управление инфраструктурой: настройка, масштабирование, обновление кластеров;
- Безопасность, аудит и соответствие внутренним политикам и регуляторным требованиям;
- Мониторинг, инцидент-менеджмент и эксплуатация приложений;
- Управление затратами и эффективностью ресурсов;

Бизнес-цели применения DevKube:

- Сокращение времени вывода продукта на рынок (time-to-market);
- Повышение стабильности и управляемости ИТ-среды;
- Автоматизация процессов разработки и эксплуатации (DevOps, GitOps);
- Упрощение соблюдения требований безопасности и регуляторов;
- Обеспечение гибкости и масштабируемости ИТ-инфраструктуры.

## **4. ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК**

### **4.1. Управление средой Kubernetes**

DevKube обеспечивает оптимизированное управление жизненным циклом кластера Kubernetes, обеспечивая беспрепятственное предоставление, масштабирование, модернизацию и всесторонний мониторинг состояния в мультиоблачных и гибридных средах.

### **4.2. Управление конвейерами**

DevKube предоставляет надежную систему оркестровки конвейеров, позволяющую организациям управлять и автоматизировать сложные рабочие процессы CI/CD. Конвейеры настраиваются с помощью кода или графических интерфейсов, что обеспечивает гибкость для различных операционных требований.

### **4.3. Шаблоны конвейеров**

Чтобы ускорить процесс внедрения и стандартизировать развертывание, DevKube предлагает predefined шаблоны конвейеров. Команды могут настраивать эти шаблоны в соответствии со своими требованиями, обеспечивая быстрое развертывание и согласованность в приложениях.

### **4.4. Встроенные интеграции с открытым исходным кодом**

- Jenkins: Надежный, гибкий сервер автоматизации для задач сборки и развертывания.
- GitLab: Интегрированное управление исходным кодом и автоматизированные триггеры конвейера.
- Harbor: Реестр контейнеров корпоративного уровня, обеспечивающий безопасное управление артефактами.
- SonarQube: Сканирование качества кода в режиме реального времени и проверка соответствия требованиям безопасности.

- Docker Registry — частный реестр Docker-образов для хранения, версионирования и безопасной доставки контейнеров внутри организации.
- JFrog Artifactory — универсальное хранилище артефактов, поддерживающее множество форматов пакетов (Docker, Maven, npm и др.), интеграцию с CI/CD и управление зависимостями.
- Nexus Repository Manager — надёжное решение для хранения и распространения артефактов, артефакт-кэширующее хранилище с поддержкой различных форматов (Maven, npm, Docker и др.).
- TestLink — система управления тестированием, позволяющая планировать, документировать и отслеживать выполнение тест-кейсов с возможностью интеграции в автоматизированные процессы.

#### **4.5. Интеграция инструментов сторонних производителей**

DevKube поддерживает широкую интеграцию со сторонними инструментами и сервисами DevOps, включая:

- Инструменты CI/CD: ArgoCD, Tekton, GitHub Actions;
- Сканеры безопасности: Trivy, Clair;
- Управление секретами: HashiCorp Vault, AWS Secrets Manager;
- Мониторинг: Prometheus, Grafana, Jaeger, Datadog.

#### **4.6. Панель показателей и мониторинга**

DevKube предоставляет интуитивно понятные информационные панели, фиксирующие ключевые показатели, включая статус выполнения конвейера, показатели успешности, продолжительность сборки, уязвимости в системе безопасности и частоту развертывания. Команды получают полезные сведения для постоянной оптимизации рабочих процессов.

В платформе DevKube реализован модуль оповещений, который позволяет настраивать отправку уведомлений пользователям по электронной почте. Уведомления формируются на основе шаблонов, где можно задать тему, содержимое и параметры сообщений. Механизм используется для информирования о событиях мониторинга, ошибках и других системных изменениях.

#### **4.7. Управление несколькими проектами**

DevKube разработан с учётом масштабирования в рамках крупных организаций и поддерживает работу множества независимых команд и проектов. Платформа обеспечивает надёжную изоляцию ресурсов, данных и прав доступа между различными рабочими группами. Администраторы могут централизованно управлять квотами, политиками доступа и распределением ресурсов для каждой команды или проекта, гарантируя безопасность и устойчивость инфраструктуры.

## 5. ТРЕБОВАНИЯ И РЕКОМЕНДАЦИИ К ИТ-ИНФРАСТРУКТУРЕ

Для развертывания и эксплуатации DevKube требуется следующая ИТ-инфраструктура:

### 5.1. Программное обеспечение

1. Операционные системы и версии ядра (тестированы разработчиком):

- Red Hat Enterprise Linux (RHEL):
  - RHEL 7.8: 3.10.0-1127.el7.x86\_64
  - RHEL 8.0–8.6: 4.18.0-80.el8.x86\_64 до 4.18.0-372.9.1.el8.x86\_64
- CentOS 7.6–7.9: 3.10.0-1127 до 3.11
- Ubuntu:
  - 20.04 LTS: 5.4.0-124-generic
  - 22.04 LTS: 5.15.0-56-generic
- РЕД ОС

2. Дополнительные компоненты (включены в установочный пакет):

- Kubernetes (настраиваемая версия);
- Kube-OVN или Calico (сетевые плагины);
- Облачное хранилище: Ceph, MinIO, TopoLVM;
- GitOps (Argo CD), Source-to-Image (S2I), DevOps pipeline.

### 5.2. Аппаратные требования

1. Минимальные требования к глобальному кластеру (control plane):

- Количество узлов: не менее 3
- Процессор:  $\geq 8$  ядер,  $\geq 2.5$  ГГц, без энергосбережения
- Оперативная память: от 16 GB (рекомендуется 32 GB)
- Диск: SSD с IOPS  $\geq 2000$ , пропускная способность  $\geq 200$  MB/s

- Архитектура:
  - x86\_64 (обязательно)
  - ARM64 (Kunpeng 920 с удвоенными требованиями:  $\geq 24$  ядер,  $\geq 48$  GB RAM)

## 2. Сетевые характеристики:

- Пропускная способность:
  - Внутрисетевые соединения: от 1 Gbps (рекомендуется 10 Gbps)
  - Межсетевые соединения: от 100 Mbps (рекомендуется 1 Gbps)
- Задержка (latency):
  - $\leq 2$  мс в пределах одного сегмента
  - $\leq 100$  мс между сегментами (рекомендуется  $\leq 30$  мс)

### 5.3. Сетевые и доменные ресурсы

- Virtual IP (VIP) — обязателен для отказоустойчивого доступа к Web UI и API;
- Доменное имя — необходимо при использовании IPv6 и/или при наличии плана аварийного восстановления (DR);
- Сертификат — рекомендуется использовать доверенный TLS-сертификат. В противном случае применяется самоподписанный.

### 5.4. Протоколы и интерфейсы

- Сетевые протоколы: HTTPS (443), Kubernetes API (6443), внутренние сервисы (11443 и др.);
- Интерфейсы доступа: Web UI, CLI-инструменты, REST API;
- Дополнительно: поддержка YAML-конфигураций, загрузка и установка операторов и плагинов через Web-интерфейс.

## **6. РЕКОМЕНДУЕМЫЕ ЭТАПЫ ДЛЯ ВНЕДРЕНИЯ ПЛАТФОРМЫ**

Этап 1. Анализ и исследование задачи:

- Анализ объектов автоматизации;
- Выявление ограничений;
- Проработка и согласование вариантов реализации задачи.

Этап 2. Регулировка механизмов, тестирование:

- Регулировка, настройка DevKube;
- Проверка соответствия разработанного решения требованиям;
- Тестирование доработок;
- Корректировка и выкладка доработок;
- Подготовка методологии и инструкции по пользованию.

Этап 3. Обучение пользователей работе с программой:

- Определение ключевых пользователей;
- Обучение ключевых пользователей.

Этап 4. Развертывание в продуктивной среде:

- Подготовка инфраструктуры для развертывания;
- Развертывание DevKube.