Описание технической архитектуры программного обеспечения «ASTRA DEV PLATFORM»

Москва 2025

Оглавление

Глоссарий

ТЕРМИН	ОПИСАНИЕ
ОС	Операционная система
Astra Linux SE	Astra Linux Special Edition. Сертифицированная ОС со встроенными средствами защиты информации
Kubernetes (K8s)	Kubernetes— система автоматизации развертывания, масштабирования и управления контейнеризированными приложениями
Боцман	Российская платформа управления мультикластерами контейнеров оркестратора kubernetes
GitFlic	Платформа работы с кодом, CI/CD конвейер, версия Enterprise Edition
CI/CD	Непрерывная интеграция и доставка
RBAC	Role-Based Access Control
RTO	Recovery Time Objective — время восстановления
RPO	Recovery Point Objective — точка восстановления
LFS	Large File Storage
MR	Merge Request — запрос на слияние
Auto Merge	Автоматическое слияние после выполнения условий
Merge Train	Цепочка MR, последовательно сливающихся в git ветку
PVC	Persistent Volume Claim — постоянное хранилище в K8s
Namespace (неймспейс)	виртуальное, логическое разделение кластера Kubernetes для изоляции ресурсов
GPG	GNU Privacy Guard — шифрование и подпись
SAST/DAST/SCA	Инструменты анализа безопасности кода

1. Общие сведения

1.1 Назначение документа

Настоящий документ описывает техническую архитектуру и эксплуатационные характеристики Astra Dev Platform (далее Платформа).

1.2 Платформа

Astra Dev Platform – это DevOps-платформа созданная на основе продуктов GitFlic и Боцман, работающий под управлением защищенной операционной системы Astra Linux Special Edition и обеспечивающая полный жизненный цикл разработки, тестирования, доставки и эксплуатации программного обеспечения в enterprise-среде.

Цель платформы — создание DevOps-цикла, где разработанный программный код, собирается, тестируется, развертывается, эксплуатируется и мониторится.

Ключевые принципы:

- Автоматизация: автоматизация DevOps-цикла;
- Изоляция и безопасность: RBAC, GPG-подписи, политики доступа;
- Масштабируемость: поддержка роста команд, проектов и нагрузки.

Платформа объединяет: управление кодом, задачами, CI/CD конвейеров, оркестрацию и эксплуатацию контейнерных приложений.

Программный комплекс предназначен для развертывания в локальной или частной облачной инфраструктуре и ориентировано на высокие требования к безопасности, отказоустойчивости и контролю.

2. Техническая архитектура платформы

2.1 Архитектурный подход

Платформа построена на основе гибридной (монолитно-микросервисной) архитектуры и использует Kubernetes контейнеризацию как основу оркестрации. Все компоненты взаимодействуют через стандартизованные протоколы: HTTP/HTTPS, WebSocket, TCP/IP, UDP.

Архитектура обеспечивает:

Высокую отказоустойчивость, горизонтальное и вертикальное масштабирование, непрерывное обновление без простоя.

2.2 Компонентная схема Платформы

Уровень управления ("Боцман Управляющий кластер")

Решаемые задачи:

Централизованное управление: Управление всеми рабочими кластерами (создание, удаление, конфигурация).

Определение политик: Установка единых правил безопасности, сетевых политик, квот для всех нижестоящих кластеров.

Мониторинг и логирование: Агрегация метрик и логов со всех кластеров для единой точки наблюдения.

Уровень выполнения ("Боцман рабочие кластеры 1...N")

Изолированные Kubernetes-кластеры, на которых непосредственно работают приложения и запускаются конвейеры. Их может быть несколько для разных целей:

- Изоляция по окружениям: кластер-1 (dev/stage), кластер-2 (production);
- Изоляция по командам/проектам: отдельный кластер для каждой команды для максимальной безопасности;
- Изоляция по географии: кластеры в разных дата-центрах или регионах.

Пространства имен (Namespaces) в рабочих кластерах

В каждом рабочем кластере используется разделение на Namespaces, определяя служебную зону работы Gitlic и рабочую зону работы пользовательских приложений

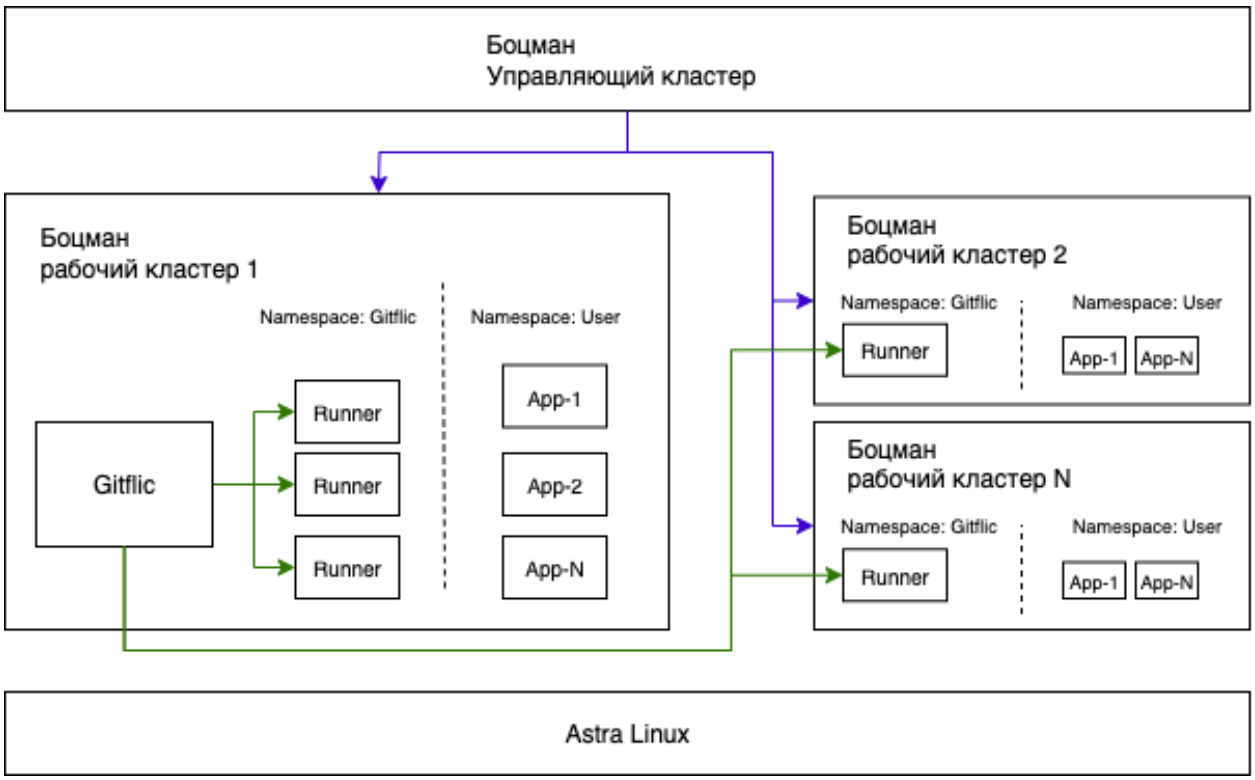
- Namespace: Gitific. Назначение: Служебное пространство для инфраструктуры CI/CD.При этом компоненты с типом Runner имеют права создавать и управлять ресурсами в других Namespaces (например, user), чтобы развертывать приложения;
- Namespace: User. Назначение: Окружение пользовательских приложения (App-1, App-2, ..., App-N).

Компонент Gitific

Экземпляр Giflic, развернутый на одном из рабочих кластеров и взаимодействующий с компонентами типа Runner.

Операционная система: Astra Linux

Обеспечивает работу платформы организациях с строгими требованиями информационной безопасности РФ.



2.3 Компоненты платформы

Платформа состоит из следующих модулей:

МОДУЛЬ	ОТВЕТСТВЕННЫЙ	ОПИСАНИЕ
Управление исходным кодом	GitFlic	Хранение репозиториев, веток, тегов, LFS, зеркал.
Управление проектами	GitFlic	Проекты, компании, команды, права доступа.
CI/CD	GitFlic и Боцман	Конвейеры, окружения развёртывания.
Управление приложениями	Боцман	Жизненный цикл контейнерных приложений.
Управление кластерами	Боцман	Создание, масштабирование, обновление K8s-кластеров.
Безопасность и доступ	GitFlic и Боцман	RBAC, LDAP/SAML/OIDC, GPG-подписи, аудит.
Мониторинг и логирование	Боцман	Метрики, события, оповещения, журналы.
Хранение данных	Боцман	PVC, хранилища, квотирование.
Реестр пакетов	GitFlic	Поддержка Docker, NPM, Maven, PyPI, Go и др.
Администрирование	GitFlic и Боцман	Web-интерфейс, CLI, API.

3. Функциональные модули

3.1 Управление исходным кодом (GitFlic)

- Поддержка Git-репозиториев с ветками, тегами, коммитами;
- Защищённые ветки, force-push, cherry-picking;
- LFS для больших файлов;
- Зеркалирование (PULL/PUSH) с GitHub, GitLab и др;
- GPG-подпись коммитов и тегов.

3.2 Управление кластерами Kubernetes (Боцман)

- Автоматическое развертывание кластера;
- Управление нодами, Pod’ами, Namespaces;
- Горизонтальное и вертикальное масштабирование;
- Сущность «Проект» - изоляция через namespaces и квотирование ресурсов.

3.3 Интегрированный CI/CD

- Поддержка шаблонов конфигураций;
- Auto Merge и Merge Trains для последовательного слияния MR;
- Запуск конвейеров по событиям: push, MR, тег, расписание;
- Интеграция с внешними системами: Jira, Jenkins, Telegram;
- Агенты CI/CD;
- Обработка и визуализация SCA, SASTS, DAST отчетов.

3.4 Реестр пакетов и зависимостей

Поддержка реестров:

Generic, Maven, NPM, PyPI, NuGet, Docker, Composer, Go, DEB, и др.

Функции реестра:

- Загрузка, скачивание, удаление пакетов через UI/API;
- Типы репозиториев: локальный, проксирующий, виртуальный;
- GPG-подпись deb пакетов.

3.4 Безопасность и доступ

- RBAC (на уровне пользователей, команд, проектов, компаний);
- Поддержка LDAP, SAML, OIDC для единого входа (SSO);

- Двухфакторная аутентификация (2FA);
- GPG-подпись коммитов, тегов, deb-пакетов;
- Аудит действий: создание, изменение, удаление объектов;
- Политики безопасности контейнеров;
- Интеграция с провайдерами аутентификации (VK ID, Yandex ID).

3.5 Мониторинг, логирование и аналитика

- Визуализация метрик потребления CPU, RAM, дискового пространства;
- Логирование событий: PUSH/PULL, запуск конвейеров, ошибки;
- Оповещения в реальном времени (e-mail, веб-уведомления, Telegram);
- Поиск и фильтрация по событиям;
- Аналитика вклада разработчиков (commit activity);
- Сохранение истории развёртываний по окружениям.

3.6 Административное управление

- Управление пользователями;
- Создание, блокировка, редактирование;
- Назначение предустановленных ролей и создание собственных;
- Аудит действий;
- Управление проектами и командами.

4. Эксплуатационные характеристики

4.1 Доступность и отказоустойчивость

ПОКАЗАТЕЛЬ	ЗНАЧЕНИЕ
Режим работы	24×7
SLA	≥ 99.9%
RTO (время восстановления)	≤ 2 часа
RPO (точка восстановления)	≤ 6 часов
Обновление без простоя	Поддерживается

4.2 Производительность

ПОКАЗАТЕЛЬ	ЗНАЧЕНИЕ
Кол-во одновременных пользователей	Зависит от мощности кластера
Время отклика UI	≤ 2 секунды
Время развертывания K8s-кластера (6 нод)	≤ 25 минут
Поддержка 1000+ репозиториев	Да
Поддержка 100+ активных CI/CD конвейеров	Да

4.3 Масштабируемость

- Горизонтальное масштабирование: добавление нод в кластер K8s;
- Вертикальное масштабирование: увеличение ресурсов узлов;
- Поддержка мультикластерности;
- Горизонтальное масштабирование реплик системы хранения кода и артефактов, CI/CD-агентов.

4.4 Резервное копирование и восстановление

- Регулярное резервное копирование БД (PostgreSQL), репозиториев, конфигураций;
- Поддержка репликации данных;
- Возможность восстановления на любой момент в пределах RPO;
- Хранение резервных копий на внешних хранилищах (NFS, S3, Ceph).