Menu

Безопасность

Alauda Container Security

Alauda Container Security

Безопасность и соответствие

Соответствие требованиям

API Refiner

О сервисе соответствия Alauda Container Platform

Модуль Compliance Service для сканирования соответствия STIG и MicroOS с поддержкой планирования и отчетности.

Пользователи и роли

Пользователь

Группа

Роль

IDP

Политика пользователя

Мультиарендность (Project)

Введение

Проект

Пространства имён (Namespaces)

Взаимосвязь между кластерами, проектами и пространствами имён

Руководства

Аудит

Введение

Требования

Процедура

Результаты поиска

Телеметрия

Установка

Требования

Шаги установки

Включение онлайн-операций

Шаги удаления

Сертификаты

Автоматическая ротация сертификатов Kubernetes

Installation

How it works

Operation Considerations

cert-manager

Обзор

Как это работает

Определение сертификатов, управляемых cert-manager

Связанные ресурсы

Сертификаты OLM

Мониторинг сертификатов

Мониторинг статуса сертификатов

Встроенные правила оповещений

Menu

Alauda Container Security

Alauda 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.

Alauda 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.

Note

Поскольку выпуски Alauda Container Security осуществляются в ином режиме, чем у Alauda Container Platform, документация Alauda Container Security теперь доступна в виде отдельного набора по адресу Alauda Container Security ↗。

Menu

Безопасность и соответствие

Соответствие требованиям

Введение

Установка Alauda Container Platform Compliance с Kyverno

Установка через консоль

Установка через YAML

Процедура удаления

Как сделать

Руководство по выполнению определённых задач и процессов.

API Refiner

Введение

Введение в продукт

Ограничения

Установка Alauda Container Platform API Refiner

Установка через консоль

Установка через YAML

Процедура удаления

Конфигурация по умолчанию

О сервисе соответствия Alauda Container Platform

О сервисе соответствия Alauda Container Platform

Модуль Compliance Service для сканирования соответствия STIG и MicroOS с поддержкой планирования и отчетности.

Menu

Соответствие требованиям

Введение

Введение

Установка Alauda Container Platform Compliance с Kyverno

Установка Alauda Container Platform Compliance с Kyverno

Установка через консоль

Установка через YAML

Процедура удаления

Как сделать

Конфигурация доступа к приватному реестру

Зачем Kyverno нужен доступ к реестру?

Быстрый старт

Политика проверки подписи образов

Что такое проверка подписи образа?

Быстрый старт

Распространённые сценарии использования

Политика проверки подписей образов с использованием Secrets

Почему использовать Secrets для публичных ключей?

Быстрый старт

Способы создания Secret

Распространённые сценарии использования

Политика проверки реестра образов

Что такое проверка реестра образов?

Быстрый старт

Распространённые сценарии

Расширенные шаблоны

Лучшие практики

Политика предотвращения выхода из контейнера

Что такое предотвращение выхода из контейнера?

Быстрый старт

Основные политики предотвращения выхода из контейнера

Расширенные сценарии

Тестирование и проверка

Лучшие практики

Политика Принудительного Применения Security Context

Что такое Принудительное Применение Security Context?

Быстрый старт

Основные Политики Security Context

Расширенные Сценарии

Тестирование и Валидация

Политика сетевой безопасности

Что такое сетевая безопасность?

Быстрый старт

Основные политики сетевой безопасности

Расширенные сценарии

Тестирование и проверка

Политика безопасности томов

Что такое безопасность томов?

Быстрый старт

Основные политики безопасности томов

Расширенные сценарии

Тестирование и проверка

Menu

Введение

ACP предоставляет функциональность соответствия на основе компонента с открытым исходным кодом Kyverno, позволяя организациям определять и применять политики в своих кластерах Kubernetes.

Эта функция решает задачу поддержания единых стандартов безопасности, управления и эксплуатации, позволяя пользователям создавать собственные политики с использованием синтаксиса YAML Kyverno и автоматически проверять ресурсы на соответствие этим политикам.

Функциональность соответствия обеспечивает комплексный мониторинг и отчетность о нарушениях, предлагая как представления на уровне ресурсов, так и на уровне политик через интуитивно понятный интерфейс, что помогает командам быстро выявлять несоответствующие ресурсы и принимать соответствующие меры для поддержания желаемого уровня безопасности и соответствия нормативным требованиям.

INFO

Для получения дополнительной информации о Kyverno прочитайте Kyverno Documentation ↗.

Установка Alauda Container Platform Compliance с Kyverno

Alauda Container Platform Compliance с Kyverno — это сервис платформы, который интегрирует Kyverno для управления политиками соответствия на Alauda Container Platform.

Содержание

Установка через консоль

1. Перейдите в раздел **Administrator**
2. В левой навигационной панели выберите **Marketplace** > **Cluster Plugins**
3. Найдите **Alauda Container Platform Compliance with Kyverno** и кликните для просмотра деталей
4. Нажмите **Install** для развертывания плагина

Установка через YAML

1. Проверка доступных версий

Убедитесь, что плагин опубликован, проверив наличие ресурсов ModulePlugin и ModuleConfig в кластере `global`:

```
# kubectl get moduleplugins kyverno
NAME      AGE
kyverno   4d20h

# kubectl get moduleconfigs -l cpaas.io/module-name=kyverno
NAME             AGE
kyverno-v4.0.4   4d21h
```

Это означает, что ModulePlugin `kyverno` существует в кластере, и версия `v4.0.4` опубликована.

2. Создание ModuleInfo

Создайте ресурс ModuleInfo для установки плагина без параметров конфигурации:

```
apiVersion: cluster.alauda.io/v1alpha1
kind: ModuleInfo
metadata:
  annotations:
    cpaas.io/display-name: kyverno
    cpaas.io/module-name: '{"en": "Alauda Container Platform Compliance for Kyverno",
      "zh": "Alauda Container Platform Compliance for Kyverno"}'
  labels:
    cpaas.io/cluster-name: global
    cpaas.io/module-name: kyverno
    cpaas.io/module-type: plugin
    cpaas.io/product: Platform-Center
  name: kyverno-global
spec:
  version: v4.2.0
```

Объяснение полей:

- `name`: Временное имя для плагина кластера. Платформа переименует его после создания на основе содержимого в формате `<cluster-name>-<hash of content>`, например, `global-ee98c9991ea1464aaa8054bdacbab313`.
- `label cpaas.io/cluster-name`: Указывает кластер, в который должен быть установлен плагин.
- `label cpaas.io/module-name`: Имя плагина, должно совпадать с ресурсом ModulePlugin.
- `label cpaas.io/module-type`: Фиксированное поле, должно быть `plugin`; отсутствие этого поля приведет к ошибке установки.
- `.spec.config`: Если соответствующий ModuleConfig пуст, это поле можно оставить пустым.
- `.spec.version`: Указывает версию плагина для установки, должна совпадать с `.spec.version` в ModuleConfig.

Процедура удаления

1. Выполните шаги 1-3 из процесса установки, чтобы найти плагин
2. Нажмите **Uninstall** для удаления плагина

Menu

Как сделать

Конфигурация доступа к приватному реестру

Зачем Kyverno нужен доступ к реестру?

Быстрый старт

Политика проверки подписи образов

Что такое проверка подписи образа?

Быстрый старт

Распространённые сценарии использования

Политика проверки подписей образов с использованием Secrets

Почему использовать Secrets для публичных ключей?

Быстрый старт

Способы создания Secret

Распространённые сценарии использования

Политика проверки реестра образов

Что такое проверка реестра образов?

Быстрый старт

Распространённые сценарии

Расширенные шаблоны

Лучшие практики

Политика предотвращения выхода из контейнера

Что такое предотвращение выхода из контейнера?

Быстрый старт

Основные политики предотвращения выхода из контейнера

Расширенные сценарии

Тестирование и проверка

Лучшие практики

Политика Принудительного Применения Security Context

Что такое Принудительное Применение Security Context?

Быстрый старт

Основные Политики Security Context

Расширенные Сценарии

Тестирование и Валидация

Политика сетевой безопасности

Что такое сетевая безопасность?

Быстрый старт

Основные политики сетевой безопасности

Расширенные сценарии

Тестирование и проверка

Политика безопасности томов

Что такое безопасность томов?

Быстрый старт

Основные политики безопасности томов

Расширенные сценарии

Тестирование и проверка

Конфигурация доступа к приватному реестру

В этом руководстве показано, как настроить Kyverno для доступа к приватным контейнерным реестрам. Когда Kyverno необходимо проверить подписи образов или получить информацию об образах, ему требуются соответствующие учетные данные для доступа к приватным реестрам — как ключ-карта для входа в охраняемое здание.

Содержание

Зачем Kyverno нужен доступ к реестру?

Kyverno нужен доступ к реестрам, когда он:

- **Проверяет подписи образов**: загружает данные подписей, чтобы убедиться, что образы корректно подписаны
- **Проверяет метаданные образов**: читает метки, аннотации и информацию из манифеста образа
- **Сканирует на уязвимости**: загружает образы для проверки безопасности
- **Проверяет содержимое образов**: инспектирует, что именно находится внутри контейнерных образов

Можно представить это как охранника, которому нужно проверить удостоверение личности — Kyverno должен «увидеть» образы, чтобы их проверить.

Быстрый старт

1. Создайте секрет для реестра

```
# Для приватного реестра компании
kubectl create secret docker-registry my-registry-secret \
  --docker-server=registry.company.com \
  --docker-username=<username> \
  --docker-password=<password> \
  --docker-email=<email@company.com> \
  -n kyverno
```

2. Настройте Kyverno для использования секрета (рекомендуется)

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: kyverno
  namespace: kyverno
imagePullSecrets:
- name: my-registry-secret
```

3. Конфигурация деплоя Kyverno

Если требуется более тонкая настройка, можно изменить деплой Kyverno напрямую:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kyverno
  namespace: kyverno
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kyverno
  template:
    metadata:
      labels:
        app: kyverno
    spec:
      serviceAccountName: kyverno
      imagePullSecrets:
      - name: my-registry-secret
      - name: gcr-secret
      - name: dockerhub-secret
      containers:
      - name: kyverno
        image: ghcr.io/kyverno/kyverno:latest
        env:
        - name: REGISTRY_CREDENTIAL_HELPERS
          value: "ecr-login,gcr,acr-env"  # Включить помощники по учетным данным
        # ... другая конфигурация
```

Политика проверки подписи образов

В этом руководстве показано, как настроить Kyverno для проверки того, что образы контейнеров правильно подписаны перед запуском в кластере Kubernetes. Это похоже на проверку удостоверения личности — только образы с действительными «подписями» допускаются к запуску.

Содержание

Что такое проверка подписи образа?

Проверка подписи образа — это как охранник, проверяющий удостоверения на входе. Она гарантирует:

- **Аутентичность образов**: они действительно исходят от заявленного источника
- **Отсутствие изменений**: никто не изменял их после подписания

- **Запуск только доверенных образов**: неподписанные или неправильно подписанные образы блокируются
- **Аудит**: отслеживание, какие образы и когда были проверены

Быстрый старт

1. Генерация ключей

```
# Создайте пару ключей для подписи (как создание системы удостоверений)
cosign generate-key-pair
# Это создаст: cosign.key (приватный, хранить в секрете) и cosign.pub (публичный, можно распространять)
```

2. Подпись образов

```
# Подпишите образы (как поставить официальный штамп)
cosign sign --key cosign.key registry.company.com/app:v1.0.0
```

3. Создание базовой политики проверки

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-signed-images
spec:
  validationFailureAction: Enforce  # Блокировать неподписанные образы
  background: false
  rules:
    - name: check-signatures
      match:
        any:
        - resources:
            kinds:
            - Pod
      verifyImages:
      - imageReferences:
        - "registry.company.com/*"  # Проверять образы из реестра компании
        attestors:
        - count: 1
          entries:
          - keys:
              publicKeys: |-
                -----BEGIN PUBLIC KEY-----
                # Вставьте сюда содержимое cosign.pub
                MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrfM
                5/KAQN0/KjHcorm/J5yctVd7iEcnessRQjU917hmKO6JWVGHpDguIyakZA==
                -----END PUBLIC KEY-----
        mutateDigest: true  # Преобразовывать теги в безопасный формат digest
```

4. Тестирование

```bash
# Примените политику
kubectl apply -f signature-policy.yaml

# Попытка запустить неподписанный образ (должна завершиться ошибкой)
kubectl run test --image=nginx:latest

# Попытка запустить подписанный образ (должно сработать)
kubectl run test --image=registry.company.com/app:v1.0.0
```

Распространённые сценарии использования

Сценарий 1: Несколько команд должны подписывать критические образы

Для критичных приложений и команда разработки, и команда безопасности могут быть обязаны подписывать образы:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-dual-signatures
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: critical-app-signatures
      match:
        any:
        - resources:
            kinds:
            - Pod
      verifyImages:
      - imageReferences:
        - "registry.company.com/critical/*"
        attestors:
        # Обе команды должны подписать
        - count: 1  # Подпись команды безопасности
          entries:
          - keys:
              publicKeys: |-
                -----BEGIN PUBLIC KEY-----
                # Публичный ключ команды безопасности
                MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrfM
                5/KAQN0/KjHcorm/J5yctVd7iEcnessRQjU917hmKO6JWVGHpDguIyakZA==
                -----END PUBLIC KEY-----
        - count: 1  # Подпись команды разработки
          entries:
          - keys:
              publicKeys: |-
                -----BEGIN PUBLIC KEY-----
                # Публичный ключ команды разработки
                MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEyctVd7iEcnessRQjU917hmKO6JWV
                GHpDguIyakZA8nXRh950IZbRj8Ra/N9sbqOPZrfM5/KAQN0/KjHcorm/J5==
                -----END PUBLIC KEY-----
        mutateDigest: true
```

Сценарий 2: Разные правила для разных окружений

В production требуется строгая проверка, в development — более мягкая:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: environment-specific-verification
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    # Строгие правила для production
    - name: production-must-be-signed
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - production
      verifyImages:
      - imageReferences:
        - "*"  # Все образы должны быть подписаны
        failureAction: Enforce  # Блокировать, если не подписаны
        attestors:
        - count: 1
          entries:
          - keys:
              publicKeys: |-
                -----BEGIN PUBLIC KEY-----
                # Ключ подписи для production
                MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrfM
                5/KAQN0/KjHcorm/J5yctVd7iEcnessRQjU917hmKO6JWVGHpDguIyakZA==
                -----END PUBLIC KEY-----
        mutateDigest: true

    # Более мягкие правила для development
    - name: development-warn-unsigned
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - development
            - staging
```

```
    verifyImages:
    - imageReferences:
      - "registry.company.com/*"  # Проверять только образы компании
      failureAction: Audit  # Аудитировать, но разрешать неподписанные образы
      attestors:
      - count: 1
        entries:
        - keys:
            publicKeys: |-
              -----BEGIN PUBLIC KEY-----
              # Ключ подписи для development
              MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEyctVd7iEcnessRQjU917hmKO6JWV
              GHpDguIyakZA8nXRh950IZbRj8Ra/N9sbqOPZrfM5/KAQN0/KjHcorm/J5==
              -----END PUBLIC KEY-----
      mutateDigest: true
```

Сценарий 3: Использование сертификатов вместо ключей

Для корпоративных сред могут использоваться сертификаты X.509:

```
# Подпись с использованием сертификата
cosign sign --cert company-cert.pem --cert-chain ca-chain.pem \
  registry.company.com/myapp:v1.0.0
```

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: certificate-verification
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: verify-with-certificates
      match:
        any:
        - resources:
            kinds:
            - Pod
      verifyImages:
      - imageReferences:
        - "registry.company.com/*"
        attestors:
        - count: 1
          entries:
          - certificates:
              cert: |-
                -----BEGIN CERTIFICATE-----
                # Сертификат подписи компании (замените на реальный сертификат)
                MIIDXTCCAkWgAwIBAgIJAKoK/heBjcOuMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
                BAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCBX
                aWRnaXRzIFB0eSBMdGQwHhcNMTcwODI4MTExNzQwWhcNMTgwODI4MTExNzQwWjBF
                MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50
                ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
                CgKCAQEAuuExVilGcXIZ3ulNuL7wLrA7VkqJoGpB1YPmYnlS7sobTggOGSqMUvqU
                BdLXcAo3ZCOXuKrBHBlltvcNdFHynfxOtkAOCZjirD6uQBrNPiQDlgMYMy14QIDAQAB
                o1AwTjAdBgNVHQ4EFgQUhKs8VQFhVLp5J4W1sFVLOVgnQxwwHwYDVR0jBBgwFoAU
                hKs8VQFhVLp5J4W1sFVLOVgnQxwwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUF
                AAOCAQEAuuExVilGcXIZ3ulNuL7wLrA7VkqJoGpB1YPmYnlS7sobTggOGSqMUvqU
                -----END CERTIFICATE-----
              rekor:
                url: https://rekor.sigstore.dev
        mutateDigest: true
```

Политика проверки подписей образов с использованием Secrets

В этом руководстве показано, как использовать Kubernetes Secrets для хранения публичных ключей для проверки подписей образов Kyverno, обеспечивая лучшую безопасность и управление ключами по сравнению с внедрением ключей непосредственно в политики.

Содержание

Почему использовать Secrets для публичных ключей?

Использование Kubernetes Secrets для хранения публичных ключей предлагает несколько преимуществ:

- **Повышенная безопасность**: ключи хранятся надежно в хранилище Kubernetes Secret
- **Простая ротация ключей**: обновление ключей без изменения политик
- **Контроль доступа**: использование RBAC для управления доступом к секретам

Быстрый старт

1. Генерация и сохранение ключей в Secret

```
# Генерация пары ключей cosign
cosign generate-key-pair

# Создание секрета из файла публичного ключа
kubectl create secret generic cosign-public-key \
  --from-file=cosign.pub=./cosign.pub \
  --namespace=kyverno

# Проверка создания секрета
kubectl get secret cosign-public-key -n kyverno
```

2. Конфигурация RBAC для Kyverno

Создайте Service Account для Kyverno

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: kyverno-secret-reader
  namespace: kyverno
```

Создайте Role для доступа к Secret

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: kyverno
  name: secret-reader
rules:
- apiGroups: [""]
  resources: ["secrets"]
  verbs: ["get", "list", "watch"]
  resourceNames: ["cosign-public-key", "team-keys"]  # Только конкретные секреты
```

Свяжите Role с Service Account

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: read-secrets
  namespace: kyverno
subjects:
- kind: ServiceAccount
  name: kyverno-secret-reader
  namespace: kyverno
roleRef:
  kind: Role
  name: secret-reader
  apiGroup: rbac.authorization.k8s.io
```

3. Создание политики с использованием ссылки на Secret

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: verify-with-secret
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: check-signatures
      match:
        any:
        - resources:
            kinds: [Pod]
      verifyImages:
      - imageReferences:
        - "registry.company.com/*"
        attestors:
        - count: 1
          entries:
          - keys:
              secret:
                name: cosign-public-key
                namespace: kyverno
                key: cosign.pub
              rekor:
                url: https://rekor.sigstore.dev
        mutateDigest: true
```

4. Тестирование конфигурации

```
# Подписать образ
cosign sign --key cosign.key registry.company.com/app:v1.0.0

# Применить политику
kubectl apply -f verify-with-secret.yaml

# Тест с подписанным образом (должно работать)
kubectl run test --image=registry.company.com/app:v1.0.0

# Тест с неподписанным образом (должно завершиться ошибкой)
kubectl run test-fail --image=nginx:latest
```

Способы создания Secret

Метод 1: Из файла

```
# Создание секрета из существующего файла публичного ключа cosign
kubectl create secret generic cosign-public-key \
  --from-file=cosign.pub=./cosign.pub \
  --namespace=kyverno
```

Метод 2: Из литеральной строки

```
# Создание секрета с содержимым публичного ключа в строке
kubectl create secret generic cosign-public-key \
  --from-literal=cosign.pub="-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrfM
5/KAQN0/KjHcorm/J5yctVd7iEcnessRQjU917hmKO6JWVGHpDguIyakZA==
-----END PUBLIC KEY-----" \
  --namespace=kyverno
```

Метод 3: Из YAML-манифеста

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: cosign-public-key
  namespace: kyverno
  labels:
    app: kyverno
    component: image-verification
type: Opaque
stringData:
  cosign.pub: |
    -----BEGIN PUBLIC KEY-----
    MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrfM
    5/KAQN0/KjHcorm/J5yctVd7iEcnessRQjU917hmKO6JWVGHpDguIyakZA==
    -----END PUBLIC KEY-----
```

```bash
kubectl apply -f cosign-secret.yaml
```

Распространённые сценарии использования

Сценарий 1: Одна команда с одним секретом

Простая настройка, когда одна команда управляет всеми подписями образов:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: single-team-verification
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: verify-team-signatures
      match:
        any:
        - resources:
            kinds: [Pod, Deployment, StatefulSet, DaemonSet]
      exclude:
        any:
        - resources:
            namespaces: [kube-system, kyverno]

      verifyImages:
      - imageReferences:
        - "registry.company.com/*"
        - "gcr.io/myproject/*"

        failureAction: Enforce

        attestors:
        - count: 1
          entries:
          - keys:
              secret:
                name: team-cosign-key
                namespace: kyverno
                key: cosign.pub
              rekor:
                url: https://rekor.sigstore.dev

        mutateDigest: true
        verifyDigest: true
        required: true
```

Сценарий 2: Несколько команд с разными секретами

Разные команды имеют свои ключи подписи и секреты:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: multi-team-verification
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    # Образы команды frontend
    - name: verify-frontend-images
      match:
        any:
        - resources:
            kinds: [Pod]
            namespaces: [frontend-*]

      verifyImages:
      - imageReferences:
        - "registry.company.com/frontend/*"

        attestors:
        - count: 1
          entries:
          - keys:
              secret:
                name: frontend-team-key
                namespace: kyverno
                key: cosign.pub
              rekor:
                url: https://rekor.sigstore.dev

        mutateDigest: true
        required: true

    # Образы команды backend
    - name: verify-backend-images
      match:
        any:
        - resources:
            kinds: [Pod]
            namespaces: [backend-*]

      verifyImages:
```

```
    - imageReferences:
      - "registry.company.com/backend/*"

      attestors:
      - count: 1
        entries:
        - keys:
            secret:
              name: backend-team-key
              namespace: kyverno
              key: cosign.pub
            rekor:
              url: https://rekor.sigstore.dev

      mutateDigest: true
      required: true
```

Сценарий 3: Критические образы, требующие нескольких подписей

Среды с повышенной безопасностью, где несколько команд должны подписывать критические образы:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: critical-multi-signature
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: verify-critical-images
      match:
        any:
        - resources:
            kinds: [Pod]
            namespaces: [production]

      verifyImages:
      - imageReferences:
        - "registry.company.com/critical/*"

        failureAction: Enforce

        attestors:
        # Подпись команды безопасности (обязательно)
        - count: 1
          entries:
          - keys:
              secret:
                name: security-team-key
                namespace: kyverno
                key: security.pub
              rekor:
                url: https://rekor.sigstore.dev

        # Подпись команды разработки (обязательно)
        - count: 1
          entries:
          - keys:
              secret:
                name: dev-team-key
                namespace: kyverno
                key: development.pub
              rekor:
                url: https://rekor.sigstore.dev
```

```
    # Подпись команды релиза (обязательно)
    - count: 1
      entries:
      - keys:
          secret:
            name: release-team-key
            namespace: kyverno
            key: release.pub
          rekor:
            url: https://rekor.sigstore.dev

    mutateDigest: true
    required: true
```

Сценарий 4: Офлайн-среда с использованием Secrets

Использование секретов в изолированных (air-gapped) средах:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: offline-verification-with-secret
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: verify-offline-images
      match:
        any:
        - resources:
            kinds: [Pod, Deployment, StatefulSet, DaemonSet]

      verifyImages:
      - imageReferences:
        - "registry.internal.com/*"
        - "airgap.company.com/*"

        failureAction: Enforce
        emitWarning: false

        attestors:
        - count: 1
          entries:
          - keys:
              secret:
                name: offline-cosign-key
                namespace: kyverno
                key: cosign.pub

              # Конфигурация офлайн-режима
              rekor:
                url: ""                    # Пустой URL для офлайн-режима
                ignoreTlog: true           # Игнорировать журнал прозрачности
                ignoreSCT: true            # Игнорировать SCT

              ctlog:
                ignoreTlog: true           # Игнорировать журнал прозрачности
сертификатов
                ignoreSCT: true            # Игнорировать SCT

        mutateDigest: true
```

```
        verifyDigest: true
        required: true
```

Политика проверки реестра образов

В этом руководстве показано, как настроить Kyverno для контроля того, какие контейнерные реестры могут использоваться в кластере Kubernetes. Реализуются политики контроля доступа к реестрам, чтобы гарантировать, что развертываются только образы из одобренных и доверенных реестров.

Содержание

Что такое проверка реестра образов?

Проверка реестра обеспечивает централизованный контроль источников образов. Это позволяет:

- **Контролировать источники образов**: разрешать только образы из доверенных реестров
- **Блокировать рискованные реестры**: предотвращать использование неизвестных или скомпрометированных реестров
- **Обеспечивать соответствие требованиям**: соблюдать требования безопасности к источникам образов
- **Разные правила для разных сред**: строгие правила для production, более мягкие для разработки
- **Отслеживать использование**: мониторить, какие реестры используются

Быстрый старт

1. Заблокировать все, кроме корпоративного реестра

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: company-registry-only
spec:
  validationFailureAction: Enforce  # Блокировать неразрешённые образы
  background: false
  rules:
    - name: check-registry
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: "Разрешён только корпоративный реестр: registry.company.com"
        pattern:
          spec:
            containers:
            - image: "registry.company.com/*"
```

2. Проверка

```bash
# Применить политику
kubectl apply -f registry-policy.yaml

# Это должно завершиться ошибкой (nginx из Docker Hub)
kubectl run test --image=nginx:latest

# Это должно сработать (если образы есть в реестре)
kubectl run test --image=registry.company.com/nginx:latest
```

Распространённые сценарии

Сценарий 1: Разрешить несколько доверенных реестров

Организации обычно используют несколько реестров:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: multiple-trusted-registries
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: check-approved-registries
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: "Образы должны поступать из одобренных реестров: корпоративный реестр, GCR или официальные образы Docker"
        anyPattern:
        - spec:
            containers:
            - image: "registry.company.com/*"      # Корпоративный реестр
        - spec:
            containers:
            - image: "gcr.io/project-name/*"       # Google Container Registry
        - spec:
            containers:
            - image: "docker.io/library/*"         # Только официальные образы Docker
        - spec:
            containers:
            - image: "quay.io/organization/*"      # Red Hat Quay
```

Сценарий 2: Разные правила для разных сред

Для production-среды нужны строгие правила, для разработки — более гибкие:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: environment-based-registry-rules
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    # Production: только сертифицированные образы
    - name: production-strict-registries
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - production
            - prod-*
      validate:
        message: "В production разрешены только сертифицированные корпоративные образы"
        pattern:
          spec:
            containers:
            - image: "registry.company.com/certified/*"

    # Development: разрешено больше реестров
    - name: development-flexible-registries
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - development
            - dev-*
            - staging
            - test-*
      validate:
        message: "В разработке разрешены корпоративный реестр, GCR и официальные образы Docker"
        anyPattern:
        - spec:
```

```
      containers:
      - image: "registry.company.com/*"
    - spec:
      containers:
      - image: "gcr.io/dev-project/*"
    - spec:
      containers:
      - image: "docker.io/library/*"
    - spec:
      containers:
      - image: "docker.io/organization/*"
```

Сценарий 3: Блокировать конкретные рискованные реестры

Блокировать определённые реестры, разрешая остальные:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: block-risky-registries
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    # Метод 1: Использовать deny-лист
    - name: block-untrusted-registries
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: "Образы из untrusted-registry.com не разрешены"
        deny:
          conditions:
          - key: "{{ request.object.spec.containers[?contains(image, 'untrusted-registry.com')] | length(@) }}"
            operator: GreaterThan
            value: 0

    # Метод 2: Использовать allow-лист для Docker Hub (только официальные образы)
    - name: allow-only-official-dockerhub
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: "Разрешены только официальные образы Docker Hub (docker.io/library/*)"
        deny:
          conditions:
          - key: "{{ request.object.spec.containers[?starts_with(image, 'docker.io/') && !starts_with(image, 'docker.io/library/')] | length(@) }}"
            operator: GreaterThan
            value: 0
```

Сценарий 4: Доступ к реестрам для конкретных команд

Разные команды могут иметь доступ к разным реестрам:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: team-specific-registries
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    # Команда frontend может использовать образы Node.js
    - name: frontend-team-registries
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - frontend-*
      validate:
        message: "Команда frontend может использовать корпоративный реестр и
официальные образы Node.js"
        anyPattern:
        - spec:
            containers:
            - image: "registry.company.com/*"
        - spec:
            containers:
            - image: "docker.io/library/node:*"
        - spec:
            containers:
            - image: "docker.io/library/nginx:*"

    # Команда data может использовать реестры ML/AI
    - name: data-team-registries
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - data-*
            - ml-*
      validate:
        message: "Команда data может использовать корпоративный реестр и образы
```

```
ML/AI"
        anyPattern:
        - spec:
            containers:
            - image: "registry.company.com/*"
        - spec:
            containers:
            - image: "docker.io/tensorflow/*"
        - spec:
            containers:
            - image: "docker.io/pytorch/*"
        - spec:
            containers:
            - image: "nvcr.io/nvidia/*"
```

Расширенные шаблоны

Эффективное использование подстановочных знаков

```
# Примеры шаблонов:
- image: "registry.company.com/*"          # Любой образ из этого реестра
- image: "registry.company.com/team-a/*"   # Только образы команды team-a
- image: "*/database:*"                    # Любой образ базы данных из любого
реестра
- image: "gcr.io/project-*/app:*"          # Любой app из проектов project-* в GCR
```

Лучшие практики

Начинайте с предупреждений

```
spec:
  validationFailureAction: Audit  # Начинайте с режима аудита, не блокируя
```

Исключайте системные пространства имён

```yaml
rules:
  - name: check-registries
    match:
      any:
      - resources:
          kinds:
          - Pod
    exclude:
      any:
      - resources:
          namespaces:
          - kube-system
          - kyverno
          - kube-public
```

Распространённые ошибки

1. **Неправильный формат образа**:
   - ❌ `registry.company.com:5000/app` (отсутствует протокол)
   - ✅ `registry.company.com/app:latest`
2. **Ошибки с подстановочными знаками**:
   - ❌ `registry.company.com*` (отсутствует слэш)
   - ✅ `registry.company.com/*`
3. **Формат Docker Hub**:
   - ❌ `nginx` (неявный docker.io)
   - ✅ `docker.io/library/nginx`

Политика предотвращения выхода из контейнера

В этом руководстве показано, как настроить Kyverno для предотвращения атак с выходом из контейнера, блокируя конфигурации контейнеров с высоким риском, которые могут позволить контейнерам выйти за пределы их изоляционных границ.

Содержание

1. Начинайте с режима аудита
2. Исключайте системные пространства имён

Что такое предотвращение выхода из контейнера?

Предотвращение выхода из контейнера включает обнаружение и блокировку опасных конфигураций контейнеров, которые могут позволить злоумышленникам выйти из изоляции контейнера и получить доступ к хост-системе. Это включает:

- **Привилегированные контейнеры**: Контейнеры, работающие с повышенными привилегиями
- **Доступ к пространствам имён хоста**: Контейнеры, использующие PID, сетевые или IPC пространства имён хоста
- **Монтирование путей хоста**: Контейнеры, монтирующие пути файловой системы хоста
- **Опасные capabilities**: Контейнеры с избыточными Linux capabilities
- **Доступ к портам хоста**: Контейнеры, привязывающиеся к сетевым портам хоста

Быстрый старт

1. Блокировка привилегированных контейнеров

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-privileged-containers
  annotations:
    policies.kyverno.io/title: Disallow Privileged Containers
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Privileged mode disables most security mechanisms and must not be allowed.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: privileged-containers
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Privileged mode is disallowed. The fields
spec.containers[*].securityContext.privileged,
          spec.initContainers[*].securityContext.privileged, and
spec.ephemeralContainers[*].securityContext.privileged
          must be unset or set to false.
        pattern:
          spec:
            =(ephemeralContainers):
              - =(securityContext):
                  =(privileged): "false"
            =(initContainers):
              - =(securityContext):
                  =(privileged): "false"
            containers:
              - =(securityContext):
                  =(privileged): "false"
```

2. Тестирование политики

```
# Применить политику
kubectl apply -f disallow-privileged-containers.yaml

# Попытка создать привилегированный контейнер (должно не пройти)
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-privileged
spec:
  containers:
  - name: nginx
    image: nginx
    securityContext:
      privileged: true
EOF

# Попытка создать обычный контейнер (должно сработать)
kubectl run test-normal --image=nginx

# Очистка
kubectl delete pod test-privileged test-normal --ignore-not-found
```

Основные политики предотвращения выхода из контейнера

Политика 1: Запрет доступа к пространствам имён хоста

Запретить контейнерам доступ к пространствам имён хоста:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-host-namespaces
  annotations:
    policies.kyverno.io/title: Disallow Host Namespaces
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Host namespaces (Process ID namespace, Inter-Process Communication namespace, and
      network namespace) allow access to shared information and can be used to elevate
      privileges. Pods should not be allowed access to host namespaces.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-namespaces
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Sharing the host namespaces is disallowed. The fields spec.hostNetwork,
          spec.hostIPC, and spec.hostPID must be unset or set to false.
        pattern:
          spec:
            =(hostPID): "false"
            =(hostIPC): "false"
            =(hostNetwork): "false"
```

Политика 2: Запрет монтирования путей хоста

Блокировать контейнеры от монтирования путей файловой системы хоста:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-host-path
  annotations:
    policies.kyverno.io/title: Disallow Host Path
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod,Volume
    policies.kyverno.io/description: >-
      HostPath volumes let Pods use host directories and volumes in containers.
      Using host resources can be used to access shared data or escalate privileges
      and should not be allowed.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-path
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          HostPath volumes are forbidden. The field spec.volumes[*].hostPath must be
unset.
        pattern:
          spec:
            =(volumes):
              - X(hostPath): "null"
```

Политика 3: Запрет портов хоста

Запретить контейнерам привязываться к сетевым портам хоста:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-host-ports
  annotations:
    policies.kyverno.io/title: Disallow Host Ports
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Access to host ports allows potential snooping of network traffic and should not be
      allowed, or at minimum restricted to a known list.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-ports-none
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Use of host ports is disallowed. The fields
spec.containers[*].ports[*].hostPort,
          spec.initContainers[*].ports[*].hostPort, and
spec.ephemeralContainers[*].ports[*].hostPort
          must either be unset or set to 0.
        pattern:
          spec:
            =(ephemeralContainers):
              - =(ports):
                  - =(hostPort): 0
            =(initContainers):
              - =(ports):
                  - =(hostPort): 0
            containers:
              - =(ports):
                  - =(hostPort): 0
```

Политика 4: Запрет опасных capabilities

Блокировать добавление опасных Linux capabilities в контейнерах:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-capabilities-strict
  annotations:
    policies.kyverno.io/title: Disallow Capabilities (Strict)
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Adding capabilities other than `NET_BIND_SERVICE` is disallowed. In addition,
      all containers must explicitly drop `ALL` capabilities.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: require-drop-all
      match:
        any:
        - resources:
            kinds:
            - Pod
      preconditions:
        all:
        - key: "{{ request.operation || 'BACKGROUND' }}"
          operator: NotEquals
          value: DELETE
      validate:
        message: >-
          Containers must drop `ALL` capabilities.
        foreach:
        - list: request.object.spec.[ephemeralContainers, initContainers, containers][]
          deny:
            conditions:
              all:
              - key: ALL
                operator: AnyNotIn
                value: "{{ element.securityContext.capabilities.drop || `[]` }}"
    - name: adding-capabilities
      match:
        any:
        - resources:
            kinds:
```

```
      - Pod
  preconditions:
    all:
    - key: "{{ request.operation || 'BACKGROUND' }}"
      operator: NotEquals
      value: DELETE
  validate:
    message: >-
      Any capabilities added other than NET_BIND_SERVICE are disallowed.
    foreach:
    - list: request.object.spec.[ephemeralContainers, initContainers, containers][]
      deny:
        conditions:
          any:
          - key: "{{ element.securityContext.capabilities.add || `[]` }}"
            operator: AnyNotIn
            value:
            - NET_BIND_SERVICE
```

Политика 5: Требовать запуск контейнеров не от root

Обеспечить запуск контейнеров от пользователей, не являющихся root:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-run-as-nonroot
  annotations:
    policies.kyverno.io/title: Require Run As Non-Root User
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Containers must run as a non-root user. This policy ensures runAsNonRoot is set to
true.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: run-as-non-root
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Running as root is not allowed. Either the field
spec.securityContext.runAsNonRoot
          must be set to true, or the field
spec.containers[*].securityContext.runAsNonRoot
          must be set to true.
        anyPattern:
        - spec:
            securityContext:
              runAsNonRoot: "true"
        - spec:
            containers:
            - securityContext:
                runAsNonRoot: "true"
```

Расширенные сценарии

Сценарий 1: Политики для разных окружений

Разные уровни безопасности для разных окружений:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: environment-container-security
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Production: Строгая безопасность
    - name: production-strict-security
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - production
            - prod-*
      validate:
        message: "Production environments require strict container security"
        pattern:
          spec:
            =(hostPID): "false"
            =(hostIPC): "false"
            =(hostNetwork): "false"
            securityContext:
              runAsNonRoot: "true"
            containers:
            - securityContext:
                privileged: "false"
                runAsNonRoot: "true"
                capabilities:
                  drop:
                  - ALL

    # Development: Более разрешительная, но всё ещё безопасная
    - name: development-basic-security
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
```

```
            - development
            - dev-*
            - staging
      validate:
        message: "Development environments require basic container security"
        pattern:
          spec:
            =(hostPID): "false"
            =(hostIPC): "false"
            containers:
            - securityContext:
                =(privileged): "false"
```

Сценарий 2: Исключения для конкретных нагрузок

Разрешить определённые нагрузки с контролируемыми исключениями:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: workload-specific-security
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: system-workloads-exception
      match:
        any:
        - resources:
            kinds:
            - Pod
      exclude:
        any:
        - resources:
            namespaces:
            - kube-system
            - kyverno
        - resources:
            kinds:
            - Pod
            names:
            - "monitoring-*"
            - "logging-*"
      validate:
        message: "Container security policies apply to application workloads"
        pattern:
          spec:
            =(hostNetwork): "false"
            containers:
            - securityContext:
                =(privileged): "false"
```

Тестирование и проверка

Тест привилегированного контейнера

```
# Это должно быть заблокировано
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-privileged
spec:
  containers:
  - name: test
    image: nginx
    securityContext:
      privileged: true
EOF
```

Тест доступа к пространствам имён хоста

```
# Это должно быть заблокировано
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-host-network
spec:
  hostNetwork: true
  containers:
  - name: test
    image: nginx
EOF
```

Тест монтирования путей хоста

```
# Это должно быть заблокировано
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-hostpath
spec:
  containers:
  - name: test
    image: nginx
    volumeMounts:
    - name: host-vol
      mountPath: /host
  volumes:
  - name: host-vol
    hostPath:
      path: /
EOF
```

Тест корректного безопасного контейнера

```
# Это должно быть разрешено
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-secure
spec:
  securityContext:
    runAsNonRoot: true
    runAsUser: 1000
  containers:
  - name: test
    image: nginx
    securityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop:
        - ALL
      readOnlyRootFilesystem: true
      runAsNonRoot: true
      runAsUser: 1000
EOF
```

Лучшие практики

1. Начинайте с режима аудита

```
spec:
  validationFailureAction: Audit  # Начинайте с предупреждений, а не блокировки
```

2. Исключайте системные пространства имён

```
exclude:
  any:
  - resources:
      namespaces:
      - kube-system
      - kyverno
      - kube-public
```

Политика Принудительного Применения Security Context

В этом руководстве показано, как настроить Kyverno для обеспечения правильных security context контейнеров, гарантируя, что они запускаются с соответствующими настройками безопасности и ограничениями.

Содержание

Тест Корректного Безопасного Контейнера (Должен Пройти)

Что такое Принудительное Применение Security Context?

Принудительное применение security context подразумевает контроль над запуском контейнеров путём установки параметров, связанных с безопасностью. Правильная конфигурация security context предотвращает:

- **Повышение привилегий root**: запуск контейнеров от имени пользователя root
- **Атаки повышения привилегий**: получение контейнерами повышенных прав
- **Небезопасное выполнение процессов**: запуск контейнеров с опасными возможностями
- **Подделка файловой системы**: контейнеры с доступной для записи корневой файловой системой
- **Обход механизмов безопасности**: контейнеры, обходящие механизмы безопасности

Быстрый старт

1. Политика Требования Запуска Не от Root

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-run-as-nonroot
  annotations:
    policies.kyverno.io/title: Require Run As Non-Root User
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Containers must run as a non-root user. This policy ensures runAsNonRoot is set to
true.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: run-as-non-root
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Running as root is not allowed. Either the field
spec.securityContext.runAsNonRoot
          must be set to true, or the field
spec.containers[*].securityContext.runAsNonRoot
          must be set to true.
        anyPattern:
        - spec:
            securityContext:
              runAsNonRoot: "true"
        - spec:
            containers:
            - securityContext:
                runAsNonRoot: "true"
```

2. Тестирование Политики

```
# Применить политику
kubectl apply -f require-run-as-nonroot.yaml

# Попытка создать контейнер, явно запускающийся от root (должно провалиться)
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-root
spec:
  containers:
  - name: nginx
    image: nginx
    securityContext:
      runAsUser: 0
      runAsNonRoot: false
EOF

# Попытка создать контейнер с пользователем не root (должно сработать)
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-nonroot
spec:
  securityContext:
    runAsNonRoot: true
    runAsUser: 1000
  containers:
  - name: nginx
    image: nginx
EOF

# Очистка
kubectl delete pod test-root test-nonroot --ignore-not-found
```

Основные Политики Security Context

Политика 1: Запрет Повышения Привилегий

Запретить контейнерам повышать привилегии:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-privilege-escalation
  annotations:
    policies.kyverno.io/title: Disallow Privilege Escalation
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Privilege escalation, such as via set-user-ID or set-group-ID file mode, should not
be allowed.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: privilege-escalation
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Privilege escalation is disallowed. The fields
          spec.containers[*].securityContext.allowPrivilegeEscalation,
          spec.initContainers[*].securityContext.allowPrivilegeEscalation,
          and spec.ephemeralContainers[*].securityContext.allowPrivilegeEscalation
          must be set to false.
        pattern:
          spec:
            =(ephemeralContainers):
              - securityContext:
                  allowPrivilegeEscalation: "false"
            =(initContainers):
              - securityContext:
                  allowPrivilegeEscalation: "false"
            containers:
              - securityContext:
                  allowPrivilegeEscalation: "false"
```

Политика 2: Требование Диапазона User ID

Обеспечить запуск контейнеров с определённым диапазоном user ID:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-user-id-range
  annotations:
    policies.kyverno.io/title: Require User ID Range
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Containers must run with a specific user ID range to prevent privilege escalation.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: user-id-range
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Containers must run with user ID between 1000 and 65535.
        deny:
          conditions:
            any:
            # Проверка securityContext на уровне Pod
            - key: "{{ request.object.spec.securityContext.runAsUser || 0 }}"
              operator: LessThan
              value: 1000
            - key: "{{ request.object.spec.securityContext.runAsUser || 0 }}"
              operator: GreaterThan
              value: 65535
            # Проверка securityContext на уровне контейнеров
            - key: "{{ request.object.spec.containers[?securityContext.runAsUser &&
(securityContext.runAsUser < `1000` || securityContext.runAsUser > `65535`)] | length(@)
}}"
              operator: GreaterThan
              value: 0
```

Политика 3: Требование Не-Root Групп

Обеспечить запуск контейнеров с не-root group ID:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-non-root-groups
  annotations:
    policies.kyverno.io/title: Require Non-Root Groups
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Containers should be required to run with a non-root group ID or supplemental
groups.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: non-root-groups
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Containers must run with non-root group ID. Either
spec.securityContext.runAsGroup
          or spec.containers[*].securityContext.runAsGroup must be set and not be 0.
        deny:
          conditions:
            any:
            # Проверка, если runAsGroup на уровне Pod равен 0
            - key: "{{ request.object.spec.securityContext.runAsGroup || 0 }}"
              operator: Equals
              value: 0
            # Проверка, если у любого контейнера runAsGroup равен 0
            - key: "{{ request.object.spec.containers[?securityContext.runAsGroup == `0`]
| length(@) }}"
              operator: GreaterThan
              value: 0
```

Политика 4: Ограничение Seccomp Профилей

Обеспечить использование безопасных seccomp профилей:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-seccomp-strict
  annotations:
    policies.kyverno.io/title: Restrict Seccomp (Strict)
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Seccomp profile must be explicitly set to one of the allowed values.
      Both the Unconfined profile and the absence of a profile are prohibited.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: seccomp-strict
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Use of custom Seccomp profiles is disallowed. The field
          spec.securityContext.seccompProfile.type must be set to RuntimeDefault or
Localhost.
        anyPattern:
        - spec:
            securityContext:
              seccompProfile:
                type: RuntimeDefault
        - spec:
            securityContext:
              seccompProfile:
                type: Localhost
        - spec:
            containers:
            - securityContext:
                seccompProfile:
                  type: RuntimeDefault
        - spec:
            containers:
```

```
            - securityContext:
                seccompProfile:
                  type: Localhost
```

Политика 5: Требование Сброса ВСЕХ Возможностей

Обеспечить, чтобы контейнеры сбрасывали все capabilities:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-drop-all-capabilities
  annotations:
    policies.kyverno.io/title: Require Drop ALL Capabilities
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Containers must drop all capabilities and only add back those that are specifically
needed.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: require-drop-all
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Containers must drop ALL capabilities.
        foreach:
        - list: request.object.spec.[ephemeralContainers, initContainers, containers][]
          deny:
            conditions:
              all:
              - key: ALL
                operator: AnyNotIn
                value: "{{ element.securityContext.capabilities.drop || `[]` }}"
```

Политика 6: Ограничение AppArmor Профилей

Контроль использования AppArmor профилей:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-apparmor-profiles
  annotations:
    policies.kyverno.io/title: Restrict AppArmor Profiles
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      On supported hosts, the runtime/default AppArmor profile is applied by default.
      The baseline policy should prevent overriding or disabling the default AppArmor
profile.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: apparmor-profiles
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          AppArmor profile must be set to runtime/default or a custom profile.
          Unconfined profiles are not allowed.
        pattern:
          metadata:
            =(annotations):
              =(container.apparmor.security.beta.kubernetes.io/*): "!unconfined"
```

Расширенные Сценарии

Сценарий 1: Security Context для Разных Сред

Различные требования безопасности для разных сред:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: environment-security-contexts
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Production: Строгие security context
    - name: production-strict-security
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - production
            - prod-*
      validate:
        message: "Production environments require strict security contexts"
        pattern:
          spec:
            securityContext:
              runAsNonRoot: "true"
              runAsUser: "1000-65535"
              runAsGroup: "1000-65535"
              seccompProfile:
                type: RuntimeDefault
            containers:
            - securityContext:
                allowPrivilegeEscalation: "false"
                readOnlyRootFilesystem: "true"
                runAsNonRoot: "true"
                capabilities:
                  drop:
                  - ALL

    # Development: Базовые требования безопасности
    - name: development-basic-security
      match:
        any:
        - resources:
            kinds:
```

```
            - Pod
            namespaces:
            - development
            - dev-*
            - staging
        validate:
          message: "Development environments require basic security contexts"
          pattern:
            spec:
              containers:
              - securityContext:
                  allowPrivilegeEscalation: "false"
                  runAsNonRoot: "true"
```

Сценарий 2: Security Context для Разных Приложений

Различные security context для разных типов приложений:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: application-security-contexts
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Базы данных: Специфичные user/group ID
    - name: database-security-context
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                app.type: database
      validate:
        message: "Database applications must use specific security contexts"
        pattern:
          spec:
            securityContext:
              runAsUser: "999"
              runAsGroup: "999"
              fsGroup: "999"
            containers:
            - securityContext:
                runAsNonRoot: "true"
                readOnlyRootFilesystem: "true"

    # Веб-приложения: Стандартный security context
    - name: web-app-security-context
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                app.type: web
      validate:
        message: "Web applications must use standard security contexts"
```

```
        pattern:
          spec:
            containers:
            - securityContext:
                runAsNonRoot: "true"
                allowPrivilegeEscalation: "false"
                capabilities:
                  drop:
                  - ALL
                  add:
                  - NET_BIND_SERVICE
```

Сценарий 3: Поэтапное Применение Security Context

Реализация прогрессивных требований безопасности:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: graduated-security-contexts
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Уровень 1: Базовая безопасность (все namespaces)
    - name: basic-security-level
      match:
        any:
        - resources:
            kinds:
            - Pod
      exclude:
        any:
        - resources:
            namespaces:
            - kube-system
            - kyverno
      validate:
        message: "All containers must have basic security contexts"
        pattern:
          spec:
            containers:
            - securityContext:
                allowPrivilegeEscalation: "false"

    # Уровень 2: Усиленная безопасность (чувствительные namespaces)
    - name: enhanced-security-level
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - finance-*
            - hr-*
            - security-*
      validate:
        message: "Sensitive namespaces require enhanced security contexts"
        pattern:
```

```
      spec:
        securityContext:
          runAsNonRoot: "true"
        containers:
        - securityContext:
            readOnlyRootFilesystem: "true"
            capabilities:
              drop:
              - ALL

  # Уровень 3: Максимальная безопасность (критические namespaces)
  - name: maximum-security-level
    match:
      any:
      - resources:
          kinds:
          - Pod
          namespaces:
          - critical-*
          - payment-*
    validate:
      message: "Critical namespaces require maximum security contexts"
      pattern:
        spec:
          securityContext:
            runAsNonRoot: "true"
            runAsUser: "1000-1999"
            runAsGroup: "1000-1999"
            seccompProfile:
              type: RuntimeDefault
          containers:
          - securityContext:
              allowPrivilegeEscalation: "false"
              readOnlyRootFilesystem: "true"
              runAsNonRoot: "true"
              capabilities:
                drop:
                - ALL
```

Тестирование и Валидация

Тест Контейнера с Root (Должен Провалиться)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-root-user
spec:
  containers:
  - name: test
    image: nginx
    securityContext:
      runAsUser: 0
EOF
```

Тест Повышения Привилегий (Должен Провалиться)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-privilege-escalation
spec:
  containers:
  - name: test
    image: nginx
    securityContext:
      allowPrivilegeEscalation: true
EOF
```

Тест Отсутствия Сброса Возможностей (Должен Провалиться)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-missing-drop-all
spec:
  containers:
  - name: test
    image: nginx
    securityContext:
      capabilities:
        add:
        - NET_ADMIN
EOF
```

Тест Корректного Безопасного Контейнера (Должен Пройти)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-secure-context
spec:
  securityContext:
    runAsNonRoot: true
    runAsUser: 1000
    runAsGroup: 1000
    seccompProfile:
      type: RuntimeDefault
  containers:
  - name: test
    image: nginx
    securityContext:
      allowPrivilegeEscalation: false
      readOnlyRootFilesystem: true
      runAsNonRoot: true
      runAsUser: 1000
      capabilities:
        drop:
        - ALL
        add:
        - NET_BIND_SERVICE
EOF
```

Политика сетевой безопасности

В этом руководстве показано, как настроить Kyverno для применения политик сетевой безопасности, которые контролируют доступ контейнеров к сети и предотвращают сетевые атаки.

Содержание

Что такое сетевая безопасность?

Сетевая безопасность включает контроль того, как контейнеры получают доступ и взаимодействуют с сетевыми ресурсами. Правильная сетевая безопасность предотвращает:

- **Доступ к сети хоста**: Контейнеры, получающие доступ к сетевым интерфейсам хоста
- **Повышение привилегий через сеть**: Использование сетевого доступа для получения повышенных прав
- **Сканирование портов и разведка**: Несанкционированные действия по обнаружению сети
- **Латеральное перемещение**: Контейнеры, получающие доступ к нежелательным сетевым ресурсам
- **Экфильтрация данных**: Несанкционированные сетевые коммуникации

Быстрый старт

1. Запретить доступ к сети хоста

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-host-network
  annotations:
    policies.kyverno.io/title: Disallow Host Network
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Доступ к сети хоста позволяет потенциально прослушивать сетевой трафик и не должен быть разрешён.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-network
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Использование сети хоста запрещено. Поле spec.hostNetwork должно быть не установлено или установлено в false.
        pattern:
          spec:
            =(hostNetwork): "false"
```

2. Проверка политики

```bash
# Применить политику
kubectl apply -f disallow-host-network.yaml

# Попытаться создать pod с сетью хоста (должно не пройти)
kubectl run test-hostnet --image=nginx --overrides='{"spec":{"hostNetwork":true}}'

# Попытаться создать обычный pod (должно пройти)
kubectl run test-normal --image=nginx
```

Основные политики сетевой безопасности

Политика 1: Запретить порты хоста

Запретить контейнерам связываться с портами сети хоста:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-host-ports
  annotations:
    policies.kyverno.io/title: Disallow Host Ports
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Доступ к портам хоста позволяет потенциально прослушивать сетевой трафик и
не должен быть
      разрешён, либо должен быть ограничен известным списком.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-ports-none
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Использование портов хоста запрещено. Поля
spec.containers[*].ports[*].hostPort,
          spec.initContainers[*].ports[*].hostPort и
spec.ephemeralContainers[*].ports[*].hostPort
          должны быть либо не установлены, либо установлены в 0.
        pattern:
          spec:
            =(ephemeralContainers):
              - =(ports):
                  - =(hostPort): 0
            =(initContainers):
              - =(ports):
                  - =(hostPort): 0
            containers:
              - =(ports):
                  - =(hostPort): 0
```

Политика 2: Ограничить диапазон портов хоста

Разрешить использование только определённых диапазонов портов хоста для контролируемого доступа:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-host-port-range
  annotations:
    policies.kyverno.io/title: Restrict Host Port Range
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Порты хоста, если используются, должны находиться в разрешённом диапазоне для предотвращения конфликтов и проблем безопасности.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-port-range
      match:
        any:
        - resources:
            kinds:
            - Pod
      preconditions:
        all:
        - key: "{{ request.object.spec.containers[].ports[?hostPort] | length(@) }}"
          operator: GreaterThan
          value: 0
      validate:
        message: >-
          Порты хоста должны находиться в разрешённом диапазоне 30000-32767.
        foreach:
        - list: request.object.spec.[ephemeralContainers, initContainers, containers][].ports[]
          preconditions:
            any:
            - key: "{{ element.hostPort }}"
              operator: GreaterThan
              value: 0
          deny:
            conditions:
              any:
              - key: "{{ element.hostPort }}"
                operator: LessThan
```

```
      value: 30000
    - key: "{{ element.hostPort }}"
      operator: GreaterThan
      value: 32767
```

Политика 3: Требовать NetworkPolicies

Обеспечить наличие у pod связанных NetworkPolicies для контроля трафика:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-network-policies
  annotations:
    policies.kyverno.io/title: Require Network Policies
    policies.kyverno.io/category: Network Security
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod,NetworkPolicy
    policies.kyverno.io/description: >-
      У pod должны быть связанные NetworkPolicies для контроля сетевого трафика.
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: require-netpol
      match:
        any:
        - resources:
            kinds:
            - Pod
      exclude:
        any:
        - resources:
            namespaces:
            - kube-system
            - kyverno
      context:
      - name: netpols
        apiCall:
          urlPath: "/apis/networking.k8s.io/v1/namespaces/{{ request.namespace
}}/networkpolicies"
          jmesPath: "items[?spec.podSelector.matchLabels.app == '{{
request.object.metadata.labels.app }}'] | length(@)"
      validate:
        message: >-
          У pod должен быть связанный NetworkPolicy. Создайте NetworkPolicy, которая
выбирает этот pod.
        deny:
          conditions:
            all:
            - key: "{{ netpols }}"
              operator: Equals
```

```
            value: 0
```

Политика 4: Ограничить типы Service

Контролировать, какие типы Service могут создаваться:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-service-types
  annotations:
    policies.kyverno.io/title: Restrict Service Types
    policies.kyverno.io/category: Network Security
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Service
    policies.kyverno.io/description: >-
      Ограничить типы Service для предотвращения экспонирования сервисов во
внешние сети.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: restrict-nodeport
      match:
        any:
        - resources:
            kinds:
            - Service
      validate:
        message: >-
          Сервисы типа NodePort не разрешены. Используйте ClusterIP или
LoadBalancer.
        pattern:
          spec:
            type: "!NodePort"
    - name: restrict-loadbalancer
      match:
        any:
        - resources:
            kinds:
            - Service
            namespaces:
            - development
            - dev-*
            - staging
      validate:
        message: >-
          Сервисы типа LoadBalancer не разрешены в средах разработки.
        pattern:
```

```
        spec:
          type: "!LoadBalancer"
```

Политика 5: Контроль конфигураций Ingress

Обеспечить безопасную конфигурацию Ingress:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: secure-ingress-configuration
  annotations:
    policies.kyverno.io/title: Secure Ingress Configuration
    policies.kyverno.io/category: Network Security
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Ingress
    policies.kyverno.io/description: >-
      Ресурсы Ingress должны быть настроены безопасно с использованием TLS и
правильных аннотаций.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: require-tls
      match:
        any:
        - resources:
            kinds:
            - Ingress
      validate:
        message: >-
          Ingress должен использовать TLS. Поле spec.tls должно быть указано.
        pattern:
          spec:
            tls:
            - hosts:
              - "*"
    - name: require-security-annotations
      match:
        any:
        - resources:
            kinds:
            - Ingress
      validate:
        message: >-
          Ingress должен иметь аннотации безопасности для SSL redirect и HSTS.
        pattern:
          metadata:
            annotations:
              nginx.ingress.kubernetes.io/ssl-redirect: "true"
```

```
    nginx.ingress.kubernetes.io/force-ssl-redirect: "true"
```

Политика 6: Ограничить конфигурацию DNS

Контролировать настройки DNS для предотвращения DNS-атак:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-dns-configuration
  annotations:
    policies.kyverno.io/title: Restrict DNS Configuration
    policies.kyverno.io/category: Network Security
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Ограничить конфигурацию DNS для предотвращения перехвата DNS и
экфильтрации данных.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: restrict-dns-policy
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Пользовательская политика DNS не разрешена. Используйте только Default
или ClusterFirst.
        pattern:
          spec:
            =(dnsPolicy): "Default | ClusterFirst"
    - name: restrict-custom-dns
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Пользовательская конфигурация DNS не разрешена в продуктивных средах.
        pattern:
          spec:
            X(dnsConfig): "null"
```

Расширенные сценарии

Сценарий 1: Сетевые политики для разных сред

Различные сетевые ограничения для разных сред:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: environment-network-security
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Продакшн: строгий контроль сети
    - name: production-network-restrictions
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - production
            - prod-*
      validate:
        message: "В продуктивных средах требуется строгая сетевая безопасность"
        pattern:
          spec:
            hostNetwork: "false"
            dnsPolicy: "ClusterFirst"
            containers:
            - ports:
              - =(hostPort): 0

    # Разработка: базовая сетевая безопасность
    - name: development-network-restrictions
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - development
            - dev-*
            - staging
      validate:
        message: "В средах разработки требуется базовая сетевая безопасность"
        pattern:
          spec:
```

```
            hostNetwork: "false"
```

Сценарий 2: Сетевые политики для разных типов приложений

Различные сетевые политики для разных типов приложений:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: application-network-policies
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Базы данных: запрет внешнего сетевого доступа
    - name: database-network-policy
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                app.type: database
      validate:
        message: "Приложения баз данных не могут использовать сеть хоста или порты хоста"
        pattern:
          spec:
            hostNetwork: "false"
            containers:
            - ports:
              - =(hostPort): 0

    # Веб-приложения: контролируемый доступ к портам
    - name: web-app-network-policy
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                app.type: web
      validate:
        message: "Веб-приложения могут использовать только стандартные HTTP/HTTPS порты"
        foreach:
        - list: request.object.spec.containers[].ports[]
```

```
        deny:
          conditions:
            any:
            - key: "{{ element.containerPort }}"
              operator: AnyNotIn
              value:
              - 80
              - 443
              - 8080
              - 8443
```

Сценарий 3: Принудительное разделение сети

Обеспечить сегментацию сети между разными уровнями:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: network-segmentation-enforcement
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: frontend-backend-separation
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                tier: frontend
      validate:
        message: "Frontend pod не могут напрямую обращаться к backend сети"
        deny:
          conditions:
            any:
            - key: "{{ request.object.metadata.labels.tier }}"
              operator: Equals
              value: backend
    - name: require-network-labels
      match:
        any:
        - resources:
            kinds:
            - Pod
      exclude:
        any:
        - resources:
            namespaces:
            - kube-system
            - kyverno
      validate:
        message: "У pod должны быть метки уровня сети для сегментации"
        pattern:
          metadata:
            labels:
              tier: "frontend | backend | database"
```

Тестирование и проверка

Проверка доступа к сети хоста (должно не пройти)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-host-network
spec:
  hostNetwork: true
  containers:
  - name: test
    image: nginx
EOF
```

Проверка связывания порта хоста (должно не пройти)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-host-port
spec:
  containers:
  - name: test
    image: nginx
    ports:
    - containerPort: 80
      hostPort: 8080
EOF
```

Проверка сервиса NodePort (должно не пройти)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Service
metadata:
  name: test-nodeport
spec:
  type: NodePort
  ports:
  - port: 80
    targetPort: 80
    nodePort: 30080
  selector:
    app: test
EOF
```

Проверка корректной сетевой конфигурации (должно пройти)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-secure-network
  labels:
    app: web-app
    tier: frontend
spec:
  dnsPolicy: ClusterFirst
  containers:
  - name: test
    image: nginx
    ports:
    - containerPort: 80
      protocol: TCP
---
apiVersion: v1
kind: Service
metadata:
  name: test-service
spec:
  type: ClusterIP
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: web-app
EOF
```

Политика безопасности томов

В этом руководстве показано, как настроить Kyverno для применения политик безопасности томов, которые ограничивают использование опасных типов томов и конфигураций, способных скомпрометировать безопасность контейнеров.

Содержание

Что такое безопасность томов?

Безопасность томов включает контроль над тем, какие типы томов могут монтировать контейнеры и каким образом они могут к ним обращаться. Правильная безопасность томов предотвращает:

- **Доступ к файловой системе хоста**: Несанкционированный доступ к директориям хоста
- **Повышение привилегий**: Использование томов для получения повышенных прав
- **Экфильтрация данных**: Доступ к конфиденциальным данным хоста через монтирование томов
- **Выход из контейнера**: Нарушение изоляции контейнера через доступ к томам
- **Небезопасные типы томов**: Использование типов томов, обходящих механизмы безопасности

Быстрый старт

1. Ограничение типов томов

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-volume-types
  annotations:
    policies.kyverno.io/title: Restrict Volume Types
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod,Volume
    policies.kyverno.io/description: >-
      Only allow safe volume types. This policy restricts volumes to configMap, csi,
      downwardAPI, emptyDir, ephemeral, persistentVolumeClaim, projected, and secret.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: restrict-volume-types
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Only the following types of volumes may be used: configMap, csi, downwardAPI,
          emptyDir, ephemeral, persistentVolumeClaim, projected, and secret.
        foreach:
        - list: "request.object.spec.volumes || []"
          deny:
            conditions:
              all:
              - key: "{{ element.keys(@) }}"
                operator: AnyNotIn
                value:
                - name
                - configMap
                - csi
                - downwardAPI
                - emptyDir
                - ephemeral
                - persistentVolumeClaim
                - projected
                - secret
```

2. Тестирование политики

```
# Применить политику
kubectl apply -f restrict-volume-types.yaml

# Попытка создать pod с томом hostPath (должно завершиться ошибкой)
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-hostpath
spec:
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - name: host-vol
      mountPath: /host
  volumes:
  - name: host-vol
    hostPath:
      path: /
EOF

# Сначала создать тестовый ConfigMap
kubectl create configmap test-config --from-literal=key=value

# Попытка создать pod с разрешённым томом (должно сработать)
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-configmap
spec:
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - name: config-vol
      mountPath: /config
  volumes:
  - name: config-vol
    configMap:
      name: test-config
EOF
```

```
# Очистка
kubectl delete pod test-hostpath test-configmap --ignore-not-found
kubectl delete configmap test-config --ignore-not-found
```

Основные политики безопасности томов

Политика 1: Запрет томов HostPath

Запретить контейнерам монтировать пути файловой системы хоста:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-host-path
  annotations:
    policies.kyverno.io/title: Disallow Host Path
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod,Volume
    policies.kyverno.io/description: >-
      HostPath volumes let Pods use host directories and volumes in containers.
      Using host resources can be used to access shared data or escalate privileges
      and should not be allowed.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-path
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          HostPath volumes are forbidden. The field spec.volumes[*].hostPath must be
unset.
        pattern:
          spec:
            =(volumes):
              - X(hostPath): "null"
```

Политика 2: Ограничение томов HostPath (только для чтения)

Разрешить определённые тома hostPath с доступом только для чтения:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-host-path-readonly
  annotations:
    policies.kyverno.io/title: Restrict Host Path (Read-Only)
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod,Volume
    policies.kyverno.io/description: >-
      HostPath volumes which are allowed must be read-only and restricted to specific
paths.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: host-path-readonly
      match:
        any:
        - resources:
            kinds:
            - Pod
      preconditions:
        all:
        - key: "{{ request.object.spec.volumes[?hostPath] | length(@) }}"
          operator: GreaterThan
          value: 0
      validate:
        message: >-
          HostPath volumes must be read-only and limited to allowed paths.
        foreach:
        - list: "request.object.spec.volumes[?hostPath]"
          deny:
            conditions:
              any:
              # Запрет, если путь не входит в разрешённый список
              - key: "{{ element.hostPath.path }}"
                operator: AnyNotIn
                value:
                - "/var/log"
                - "/var/lib/docker/containers"
                - "/proc"
                - "/sys"
```

```
      foreach:
      - list: "request.object.spec.containers[].volumeMounts[?name]"
        deny:
          conditions:
            any:
            # Запрет, если монтирование тома не является только для чтения
            - key: "{{ element.readOnly || false }}"
              operator: Equals
              value: false
```

Политика 3: Запрет привилегированных типов томов

Блокировать типы томов, которые могут обходить механизмы безопасности:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-privileged-volumes
  annotations:
    policies.kyverno.io/title: Disallow Privileged Volume Types
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: high
    policies.kyverno.io/subject: Pod,Volume
    policies.kyverno.io/description: >-
      Certain volume types are considered privileged and should not be allowed.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: disallow-privileged-volumes
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Privileged volume types are not allowed: hostPath, gcePersistentDisk,
          awsElasticBlockStore, gitRepo, nfs, iscsi, glusterfs, rbd, flexVolume,
          cinder, cephFS, flocker, fc, azureFile, azureDisk, vsphereVolume, quobyte,
          portworxVolume, scaleIO, storageos.
        foreach:
        - list: "request.object.spec.volumes || []"
          deny:
            conditions:
              any:
              - key: "{{ element.keys(@) }}"
                operator: AnyIn
                value:
                - hostPath
                - gcePersistentDisk
                - awsElasticBlockStore
                - gitRepo
                - nfs
                - iscsi
                - glusterfs
                - rbd
```

```
    - flexVolume
    - cinder
    - cephFS
    - flocker
    - fc
    - azureFile
    - azureDisk
    - vsphereVolume
    - quobyte
    - portworxVolume
    - scaleIO
    - storageos
```

Политика 4: Требовать корневую файловую систему только для чтения

Обеспечить использование контейнерами корневой файловой системы только для чтения:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-readonly-rootfs
  annotations:
    policies.kyverno.io/title: Require Read-Only Root Filesystem
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      A read-only root file system helps to enforce an immutable infrastructure strategy;
      the container only needs to write on the mounted volume that persists the state.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: readonly-rootfs
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Root filesystem must be read-only. Set readOnlyRootFilesystem to true.
        foreach:
        - list: request.object.spec.[ephemeralContainers, initContainers, containers][]
          deny:
            conditions:
              any:
              - key: "{{ element.securityContext.readOnlyRootFilesystem || false }}"
                operator: Equals
                value: false
```

Политика 5: Контроль разрешений монтирования томов

Ограничить разрешения и пути монтирования томов:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: control-volume-mounts
  annotations:
    policies.kyverno.io/title: Control Volume Mount Permissions
    policies.kyverno.io/category: Pod Security Standards (Restricted)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod,Volume
    policies.kyverno.io/description: >-
      Control where volumes can be mounted and with what permissions.
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: restrict-mount-paths
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Volume mounts to sensitive paths are not allowed.
        foreach:
        - list: request.object.spec.[ephemeralContainers, initContainers, containers]
[].volumeMounts[]
          deny:
            conditions:
              any:
              # Блокировать монтирование в чувствительные системные пути
              - key: "{{ element.mountPath }}"
                operator: AnyIn
                value:
                - "/etc"
                - "/root"
                - "/var/run/docker.sock"
                - "/var/lib/kubelet"
                - "/var/lib/docker"
                - "/usr/bin"
                - "/usr/sbin"
                - "/sbin"
                - "/bin"
```

```
    - name: require-readonly-sensitive-mounts
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: >-
          Mounts to /proc and /sys must be read-only.
        foreach:
        - list: request.object.spec.[ephemeralContainers, initContainers, containers]
[].volumeMounts[]
          preconditions:
            any:
            - key: "{{ element.mountPath }}"
              operator: AnyIn
              value:
              - "/proc"
              - "/sys"
          deny:
            conditions:
              any:
              - key: "{{ element.readOnly || false }}"
                operator: Equals
                value: false
```

Расширенные сценарии

Сценарий 1: Политики томов для разных сред

Различные ограничения томов для разных сред:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: environment-volume-security
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Production: Строгий контроль томов
    - name: production-volume-restrictions
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - production
            - prod-*
      validate:
        message: "Production environments allow only secure volume types"
        foreach:
        - list: "request.object.spec.volumes || []"
          deny:
            conditions:
              all:
              - key: "{{ element.keys(@) }}"
                operator: AnyNotIn
                value:
                - name
                - configMap
                - secret
                - persistentVolumeClaim
                - emptyDir

    # Development: Более либеральные, но безопасные
    - name: development-volume-restrictions
      match:
        any:
        - resources:
            kinds:
            - Pod
            namespaces:
            - development
```

```
        - dev-*
        - staging
    validate:
      message: "Development environments allow additional volume types"
      foreach:
      - list: "request.object.spec.volumes || []"
        deny:
          conditions:
            any:
            - key: "{{ element.keys(@) }}"
              operator: AnyIn
              value:
              - hostPath  # Всё ещё блокировать hostPath в dev
              - nfs       # Блокировать сетевые файловые системы
```

Сценарий 2: Политики томов для разных типов приложений

Различные политики томов для разных типов приложений:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: application-volume-policies
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    # Приложения баз данных: Разрешить постоянное хранилище
    - name: database-volume-policy
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                app.type: database
      validate:
        message: "Database applications must use persistent volumes"
        pattern:
          spec:
            volumes:
            - persistentVolumeClaim: {}

    # Веб-приложения: Ограничить безопасными томами
    - name: web-app-volume-policy
      match:
        any:
        - resources:
            kinds:
            - Pod
            selector:
              matchLabels:
                app.type: web
      validate:
        message: "Web applications can only use safe volume types"
        foreach:
        - list: "request.object.spec.volumes || []"
          deny:
            conditions:
              all:
              - key: "{{ element.keys(@) }}"
```

```
      operator: AnyNotIn
      value:
      - name
      - configMap
      - secret
      - emptyDir
      - projected
```

Сценарий 3: Ограничения по размеру томов и ресурсам

Контроль размеров томов и использования ресурсов:

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: volume-resource-limits
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: limit-emptydir-size
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: "EmptyDir volumes must have size limits"
        foreach:
        - list: "request.object.spec.volumes[?emptyDir]"
          deny:
            conditions:
              any:
              - key: "{{ element.emptyDir.sizeLimit || '' }}"
                operator: Equals
                value: ""
    - name: limit-emptydir-memory
      match:
        any:
        - resources:
            kinds:
            - Pod
      validate:
        message: "EmptyDir memory volumes are not allowed"
        foreach:
        - list: "request.object.spec.volumes[?emptyDir]"
          deny:
            conditions:
              any:
              - key: "{{ element.emptyDir.medium || '' }}"
                operator: Equals
                value: "Memory"
```

Тестирование и проверка

Тест тома HostPath (должен завершиться ошибкой)

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: test-hostpath
spec:
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - name: host-vol
      mountPath: /host
  volumes:
  - name: host-vol
    hostPath:
      path: /
EOF
```

Menu

API Refiner

Введение

Введение в продукт

Ограничения

Установка Alauda Container Platform API Refiner

Установка через консоль

Установка через YAML

Процедура удаления

Конфигурация по умолчанию

Введение

Содержание

Введение в продукт

ACP API Refiner — это сервис фильтрации данных, предоставляемый платформой Alauda Container Platform, который повышает безопасность мультиарендности и изоляцию данных в Kubernetes-средах. Он фильтрует данные ответов Kubernetes API на основе разрешений пользователей, проектов, кластеров и пространств имён, а также поддерживает фильтрацию на уровне полей, включение и десенситизацию данных.

Ограничения

К ACP API Refiner применяются следующие ограничения:

- Ресурсы должны содержать определённые метки, связанные с арендатором, для изоляции данных:
  - `cpaas.io/project`
  - `cpaas.io/cluster`
  - `cpaas.io/namespace`
  - `kubernetes.io/metadata.name`

  - Необязательно: `cpaas.io/creator`

- Запросы LabelSelector не поддерживают логические операции OR
- Привязки пользователей на уровне платформы не фильтруются
- Фильтрация применяется только к операциям API GET и LIST

Установка Alauda Container Platform API Refiner

Alauda Container Platform API Refiner — это сервис платформы, который фильтрует данные ответов Kubernetes API. Он предоставляет возможности фильтрации по проекту, кластеру и namespace, а также поддерживает исключение, включение и десенситизацию полей в ответах API.

Содержание

Установка через консоль

1. Перейдите в **Administrator**
2. В левой навигационной панели нажмите **Marketplace** > **Cluster Plugins**
3. Выберите кластер **global** в верхней панели навигации
4. Найдите **Alauda Container Platform API Refiner** и кликните для просмотра деталей
5. Нажмите **Install** для развертывания плагина

Установка через YAML

1. Проверка доступных версий

Убедитесь, что плагин опубликован, проверив наличие ресурсов ModulePlugin и ModuleConfig в кластере `global`:

```
# kubectl get moduleplugins apirefiner
NAME      AGE
apirefiner   4d20h

# kubectl get moduleconfigs -l cpaas.io/module-name=apirefiner
NAME             AGE
apirefiner-v4.0.4   4d21h
```

Это означает, что ModulePlugin `apirefiner` существует в кластере, а версия `v4.0.4` опубликована.

2. Создание ModuleInfo

Создайте ресурс ModuleInfo для установки плагина без параметров конфигурации:

```yaml
apiVersion: cluster.alauda.io/v1alpha1
kind: ModuleInfo
metadata:
  annotations:
    cpaas.io/display-name: apirefiner
    cpaas.io/module-name: '{"en": "Alauda Container Platform API Refiner", "zh": "Alauda
      Container Platform API Refiner"}'
  labels:
    cpaas.io/cluster-name: global
    cpaas.io/module-name: apirefiner
    cpaas.io/module-type: plugin
    cpaas.io/product: Platform-Center
  name: apirefiner-global
spec:
  version: v4.2.0-default.1.g8f0543e4
```

Объяснение полей:

- `name`: Временное имя для кластерного плагина. Платформа переименует его после создания на основе содержимого в формате `<cluster-name>-<hash содержимого>`, например, `global-ee98c9991ea1464aaa8054bdacbab313`.
- `label cpaas.io/cluster-name`: API Refiner можно установить только в кластере `global`, оставьте это поле равным global.
- `label cpaas.io/module-name`: Имя плагина, должно совпадать с ресурсом ModulePlugin.
- `label cpaas.io/module-type`: Фиксированное поле, должно быть `plugin`; отсутствие этого поля приведёт к ошибке установки.
- `.spec.config`: Если соответствующий ModuleConfig пуст, это поле можно оставить пустым.
- `.spec.version`: Указывает версию плагина для установки, должна совпадать с `.spec.version` в ModuleConfig.

Процедура удаления

1. Выполните шаги 1-4 из процесса установки для поиска плагина

2. Нажмите **Uninstall** для удаления плагина

Конфигурация по умолчанию

Фильтруемые ресурсы

По умолчанию фильтруются следующие ресурсы:

Ресурс	API Version
namespaces	v1
projects	auth.alauda.io/v1
clustermodules	cluster.alauda.io/v1alpha2
clusters	clusterregistry.k8s.io/v1alpha1

Десенситизация полей

По умолчанию десенситизируется следующее поле:

- `metadata.annotations.cpaas.io/creator`

Menu

About Alauda Container Platform Compliance Service

Compliance Service — это модуль платформы, предназначенный для поддержки сканирования соответствия требованиям STIG и операционной системы MicroOS. Он предоставляет готовые возможности сканирования на соответствие с поддержкой планового сканирования и подробной отчетности.

Note

Поскольку выпуски Compliance Service осуществляются в ином режиме, чем у Alauda Container Platform, документация Compliance Service теперь доступна в виде отдельного набора по адресу Compliance Service ↗。

Menu

Пользователи и роли

Пользователь

Введение

Источники пользователей

Правила управления пользователями

Жизненный цикл пользователя

Руководства

Группа

Введение

Введение в группы

Типы групп

Руководства

Роль

Введение

Введение в роли

Системные роли

Пользовательские роли

Руководства

IDP

Введение

Overview

Supported Integration Methods

Руководства

Устранение неполадок

Политика пользователя

Введение

Menu

Пользователь

Введение

Руководства

Управление пользователями

Сброс пароля локального пользователя

Обновление даты истечения срока действия пользователя

Активация пользователя

Отключение пользователя

Добавление пользователя в локальную группу пользователей

Удаление пользователя

Пакетные операции

Введение

Платформа поддерживает аутентификацию пользователей и проверку входа для всех пользователей.

Содержание

Источники пользователей

Локальные пользователи

- Учетная запись администратора, созданная при развертывании платформы
- Учетные записи, созданные через интерфейс платформы
- Пользователи, добавленные через локальный конфигурационный файл dex

Пользователи третьих сторон

LDAP-пользователи

- Корпоративные пользователи, синхронизированные с LDAP-серверов
- Учетные 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

> **WARNING**
>
> 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 пользователи третьих сторон

- Пользователи, аутентифицированные через поддерживаемые коннекторы dex (например, GitHub, Microsoft)
- Для получения дополнительной информации см. официальную документацию dex ↗

Правила управления пользователями

> **WARNING**
>
> Обратите внимание на следующие важные правила:
>
> - 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 учетных 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

В следующей таблице описаны различные статусы пользователей на платформе:

Статус	Описание
Normal	Учетная запись пользователя активна и может войти в платформу
Disabled	Учетная запись пользователя неактивна и не может войти. Обратитесь к администратору платформы для активации. Возможные причины: - Отсутствие входа в течение 90+ дней подряд - Истечение срока действия учетной записи - Ручное отключение администратором
Locked	Учетная запись временно заблокирована из-за 5 неудачных попыток входа в течение 24 часов. Подробности: - Продолжительность блокировки: 20 минут - Может быть разблокирована вручную администратором - После окончания периода блокировки учетная запись становится доступной
Invalid	Учетная запись, синхронизированная с LDAP, которая была удалена с LDAP-сервера.

Статус	Описание
	Примечание: недействительные учетные записи не могут войти в платформу

Menu

Руководства

Управление ролями пользователей

Добавление ролей

Удаление ролей

Создание пользователя

Шаги

Управление пользователями

Сброс пароля локального пользователя

Обновление даты истечения срока действия пользователя

Активация пользователя

Отключение пользователя

Добавление пользователя в локальную группу пользователей

Удаление пользователя

Пакетные операции

Управление ролями пользователей

Администраторы платформы могут управлять ролями других пользователей (не своей собственной учетной записи), чтобы предоставлять или отзывать разрешения.

Содержание

Добавление ролей

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите на имя пользователя, для которого нужно изменить роли
3. Прокрутите до раздела **Role List**
4. Нажмите **Add Role**
5. В диалоговом окне назначения роли:

- Выберите роль из выпадающего списка **Role Name**
- Выберите область действия разрешений роли (кластер, проект или namespace)
- Нажмите **Add**

NOTE

Важные замечания:

- Вы можете добавить 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

Удаление ролей

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите на имя пользователя, для которого нужно изменить роли
3. Прокрутите до раздела **Role List**
4. Нажмите **Remove** рядом с ролью, которую хотите удалить
5. Подтвердите удаление

WARNING

Разрешения на управление ролями:

- Управлять ролями других пользователей могут только администраторы платформы
- Пользователи не могут изменять роли своей собственной учетной записи

Создание пользователя

Пользователи с ролями администратора платформы могут создавать локальных пользователей и назначать им роли через интерфейс платформы.

Содержание

Шаги

1. В левой панели навигации нажмите **Users** > **User Management**
2. Нажмите **Create User**
3. Настройте следующие параметры:

Параметр	Описание
Password Type	Выберите способ генерации пароля: **Random**: Система генерирует безопасный случайный пароль **Custom**: Пользователь вводит пароль вручную
Password	Введите или сгенерируйте пароль в зависимости от выбранного типа. Требования к паролю: - Длина: 8-32 символа - Должен содержать буквы и цифры

Параметр	Описание
	- Должен содержать специальные символы (`~!@#$%^&*() -_=+?`) Функции поля пароля: - Нажмите на иконку глаза для показа/скрытия пароля - Нажмите на иконку копирования для копирования пароля
Mailbox	Электронная почта пользователя: - Должна быть уникальной - Может использоваться как имя пользователя для входа - Связана с именем пользователя
Validity Period	Установите срок действия учетной записи пользователя: Варианты: - **Permanent**: Без ограничения по времени - **Custom**: Установите время начала и окончания с помощью выпадающего списка Time Range
Roles	Назначьте пользователю одну или несколько ролей
Continue Creating	Переключатель для управления поведением после создания: - **On**: Перенаправляет на страницу создания нового пользователя - **Off**: Показывает страницу с деталями созданного пользователя

1. Нажмите **Create**

NOTE

После успешного создания пользователя:

- Если включена опция "Continue Creating", вы будете перенаправлены на создание следующего пользователя
- Если 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).

WARNING

Важные 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

Содержание

Сброс пароля локального пользователя

Пользователи с правами управления платформой могут сбрасывать пароли других локальных пользователей.

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите на иконку рядом с записью нужного пользователя
3. Нажмите **Reset Password**
4. В диалоговом окне выберите тип пароля:

- **Random**: Система сгенерирует надёжный случайный пароль
- **Custom**: Введите новый пароль вручную

NOTE

Требования к паролю:

- Длина: 8-32 символа
- Должен содержать буквы и цифры
- Должен содержать специальные символы (`~!@#$%^&*() -_=+?`)

Особенности поля пароля:

- Нажмите на иконку глаза для показа/скрытия пароля

- Нажмите на иконку копирования для копирования пароля

1. 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.

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите **Update Expiry Date** рядом с нужным пользователем
3. В диалоговом окне выберите опцию даты истечения:

- **Permanent**: Без ограничения по времени
- **Custom**: Установите время начала и окончания с помощью выпадающего списка Time Range

4. Нажмите **Update**

Активация пользователя

Вы можете активировать пользователей со статусом **disabled** или **locked**.

NOTE

Поведение при активации:

- Если пользователь находится в пределах срока действия: дата истечения остаётся без изменений
- Если срок действия пользователя истёк: дата истечения становится **Permanent**

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите **Activate** рядом с нужным пользователем
3. В диалоговом окне подтверждения нажмите **Activate**
4. 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.

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите на иконку рядом с нужным пользователем
3. Нажмите **Disable** и подтвердите

Добавление пользователя в локальную группу пользователей

Вы можете добавить пользователей с **Source** равным **Local** или `LDAP` в одну или несколько локальных групп пользователей.

WARNING

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

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите на иконку рядом с нужным пользователем
3. 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 вошедшей в систему учётной записи, включая:

- Пользователей, настроенных через IDP
- Пользователей с источником `-`
- Локальных пользователей

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Нажмите на иконку рядом с нужным пользователем
3. Нажмите **Delete**
4. Нажмите **Confirm**

Пакетные операции

Вы можете выполнять пакетные операции для:

- Обновления сроков действия
- Активации пользователей
- Отключения пользователей
- Удаления пользователей

Шаги

1. В левой навигационной панели нажмите **Users** > **User Management**
2. Выберите одного или нескольких пользователей с помощью чекбоксов
3. Нажмите **Batch Operations** и выберите действие:

- **Update Validity**
- **Activate**
- **Deactivate**
- **Delete**

NOTE

Подробности пакетных операций:

- **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

Menu

Группа

Введение

Введение

Введение в группы

Типы групп

Руководства

Управление ролями групп пользователей

Добавление роли группе

Удаление роли из группы

Создание локальной группы пользователей

Создание группы пользователей

Управление группами пользователей

Управление членством в локальной группе пользователей

Предварительные требования

Импорт участников

Удаление участников

Введение

Содержание

Введение в группы

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 namespace нескольким пользователям, вы можете:

1. Создать пользовательскую группу
2. Импортировать соответствующих пользователей в члены группы
3. Настроить роли проекта и namespace для группы
4. Применить единые права ко всем членам группы

Типы групп

Платформа поддерживает два типа групп:

Локальная пользовательская группа

- Создаётся непосредственно на 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)

Menu

Руководства

Управление ролями групп пользователей

Добавление роли группе

Удаление роли из группы

Создание локальной группы пользователей

Создание группы пользователей

Управление группами пользователей

Управление членством в локальной группе пользователей

Предварительные требования

Импорт участников

Удаление участников

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.

Содержание

Добавление роли группе

Шаги

1. В левой навигационной панели нажмите **Users** > **User Group Management**
2. Нажмите на название нужной группы пользователей
3. На вкладке **Configure Role** нажмите **Add Role**
4. Нажмите, чтобы добавить роль

NOTE

Правила назначения ролей:

- Вы можете добавить несколько ролей в одну группу

- Каждая роль может быть добавлена в одну группу только один раз

1. Выберите название роли из выпадающего списка
2. Выберите область действия роли (кластер, проект или namespace)
3. Нажмите **Add**

Удаление роли из группы

WARNING

При удалении роли из группы:

- Все права, предоставленные этой ролью членам группы, будут отозваны
- Это действие нельзя отменить

Шаги

1. В левой навигационной панели нажмите **Users** > **User Group Management**
2. Нажмите на название нужной группы пользователей
3. На вкладке **Configure Role** нажмите **Remove** рядом с ролью
4. Нажмите **Confirm** для удаления роли

Создание локальной группы пользователей

Локальные группы пользователей позволяют реализовать управление доступом на основе ролей для нескольких пользователей из любого источника.

Содержание

Создание группы пользователей

Шаги

1. В левой боковой панели нажмите **Users** > **User Group Management**
2. Нажмите **Create User Group**
3. Введите следующую информацию:

- **Name**: Название группы пользователей
- **Description**: Описание назначения группы

4. Нажмите **Create**

Управление группами пользователей

Вы можете управлять группами пользователей, нажав на иконку на странице списка или выбрав **Operations** в правом верхнем углу на странице с деталями.

Операция	Описание
Update User Group	Обновление информации о группе в зависимости от источника группы: - Для групп с **Source** равным `Local` : можно обновить как название, так и описание - Для групп с **Source** равным `IDP name` : можно обновить только описание
Delete Local User Group	Удаление групп пользователей с **Source** равным `Local`

WARNING

При удалении группы:

- Все участники группы будут удалены
- Все роли, назначенные группе, будут удалены
- Это действие нельзя отменить

Управление членством в локальной группе пользователей

Управлять членством в локальных группах пользователей могут только пользователи с правами Platform Management.

Содержание

Предварительные требования

WARNING

Перед управлением членством в группах обратите внимание на следующие 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.

TIP

Пользователи, импортированные в группу, автоматически наследуют все операционные права, назначенные этой группе.

Шаги

1. В левой навигационной панели нажмите **Users** > **User Group Management**
2. Нажмите на название локальной группы пользователей, в которую хотите добавить участников
3. На вкладке **Group Member Management** нажмите **Import Member**
4. Выберите одного или нескольких пользователей платформы, отметив флажками их имена пользователей/отображаемые имена
5. Нажмите **Import**

NOTE

- 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

Удаление участников

При удалении пользователя из группы все операционные права, предоставленные этому пользователю через группу, автоматически аннулируются.

Шаги

1. В левой навигационной панели нажмите **Users** > **User Group Management**

2. Нажмите на название локальной группы пользователей, из которой хотите удалить участников

3. На вкладке **Group Member Management** можно удалить участников двумя способами:

- Нажать **Remove** рядом с именем участника и подтвердить действие
- Выбрать одного или нескольких участников с помощью флажков, затем нажать **Batch Remove** и подтвердить действие

Menu

Роль

Введение

Введение

Введение в роли

Системные роли

Пользовательские роли

Руководства

Создание роли

Конфигурация основной информации

Конфигурация просмотра

Конфигурация разрешений

Управление пользовательскими ролями

Обновление основной информации

Обновление разрешений роли

Копирование существующей роли

Удаление пользовательской роли

Введение

Содержание

Введение в роли

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 A: Может только просматривать и создавать проекты
- Роль B: Может создавать, просматривать, обновлять и удалять пользователей, проекты и namespaces

---

Системные роли

Для удовлетворения распространённых сценариев настройки прав доступа платформа предоставляет следующие 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

Название роли	Описание	Уровень роли
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

> **WARNING**
>
> Будьте осторожны при обновлении или удалении пользовательских ролей. Удаление пользовательской роли автоматически отзовёт все права, предоставленные этой ролью связанным пользователям.

Menu

Руководства

Создание роли

Конфигурация основной информации

Конфигурация просмотра

Конфигурация разрешений

Управление пользовательскими ролями

Обновление основной информации

Обновление разрешений роли

Копирование существующей роли

Удаление пользовательской роли

Создание роли

Пользователи с разрешениями ролей платформы могут создавать пользовательские роли с разрешениями, которые равны или меньше их собственных разрешений в зависимости от реальных сценариев использования. При создании роли можно настроить:

- Разрешения на операции функциональных модулей платформы
- Разрешения доступа к пользовательским ресурсам (Kubernetes CRD)

Содержание

Конфигурация основной информации

1. В левой навигационной панели нажмите **Users** > **Roles**.
2. Нажмите **Create Role**.
3. Настройте основную информацию роли:

Тип роли

При назначении ролей пользователям область разрешений будет ограничена в зависимости от типа роли:

- **Platform Role**: Отображаются все разрешения платформы
- **Project Role**: Отображаются разрешения в разделах:
  - Project Management
  - Container Platform
  - Service Mesh
  - DevOps
  - Middleware
- **Namespace Role**: Отображаются разрешения в разделах:
  - Project Management
  - Container Platform
  - Service Mesh
  - DevOps
  - Middleware

1. Нажмите **Next**.

Конфигурация просмотра

В разделе конфигурации просмотра вы управляете разрешениями роли на доступ к указанным видам. Виды, которые не выбраны, не будут отображаться в верхней навигации для пользователей с этой ролью.

NOTE

1. Разрешения вашей учетной записи ограничивают, какие карточки видов вы можете настраивать. Например:
   - Если у вашей учетной записи нет разрешения на просмотр **Project Management**
   - Карточка вида **Project Management** будет неактивна при создании роли

- Вы можете создавать роли только с разрешениями, равными или ниже ваших собственных

2. Статус входа в вид:

- Если в функции **Products** у вида отключена опция **Show Entry**
- Разрешения вида в **Permission Configuration** по-прежнему будут действовать
- Вид временно будет недоступен, пока вход не будет включен
- После включения ранее выбранные разрешения будут работать как обычно

Конфигурация разрешений

1. Нажмите **Add Custom Permission** в левом верхнем углу страницы.

2. Настройте разрешения для роли на работу с пользовательскими ресурсами (Kubernetes CRD):

Параметр	Описание
Group Name	Название группы разрешений. Группы отображаются под модулем разрешений в порядке их добавления.
Resource Name	Название ресурса. Нажмите Enter для добавления нескольких названий пользовательских ресурсов.
Operation Permission	Разрешение на выполнение операций с ресурсом.

1. Нажмите **Create**.

Управление пользовательскими ролями

В этом руководстве описывается, как управлять пользовательскими ролями на платформе, включая:

- Обновление основной информации и разрешений
- Копирование существующих ролей для создания новых
- Удаление пользовательских ролей

Содержание

Обновление основной информации

Вы можете обновить отображаемое имя и описание пользовательских ролей на платформе.

Шаги

1. В левой навигационной панели нажмите **Users** > **Roles**
2. Нажмите на имя ***роли, которую нужно обновить***
3. В правом верхнем углу нажмите **Actions** > **Update**
4. Обновите:

- Отображаемое имя роли
- Описание

5. Нажмите **Update**

Обновление разрешений роли

Вы можете обновить информацию о разрешениях пользовательских ролей, включая:

- Добавление новых разрешений на операции с ресурсами платформы
- Удаление существующих разрешений
- Изменение разрешений для пользовательских ресурсов

Шаги

1. В левой навигационной панели нажмите **Users** > **Roles**
2. Нажмите на имя ***роли, которую нужно обновить***
3. В правом верхнем углу области разрешений нажмите **Actions** > **Update Role Permissions**
4. Внесите изменения на странице **Update Role Permissions**
5. 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.

WARNING

Разрешения новой роли не могут превышать разрешения роли, к которой принадлежит создатель.

Шаги

1. В левой навигационной панели нажмите **Users** > **Roles**
2. Нажмите на имя ***роли, которую нужно скопировать***
3. В правом верхнем углу нажмите **Actions** > **Copy as new role**
4. На странице **Copy as new role** настройте:

- Имя
- Отображаемое имя
- Описание
- Тип

5. Нажмите **Create**

Удаление пользовательской роли

Вы можете удалить пользовательские роли, которые больше не используются.

WARNING

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

Шаги

1. В левой навигационной панели нажмите **Users** > **Roles**
2. Нажмите на имя ***роли, которую нужно удалить***
3. В правом верхнем углу нажмите **Actions** > **Delete**
4. Введите имя роли для подтверждения удаления
5. Нажмите **Delete**

Menu

IDP

Введение

Введение

Overview

Supported Integration Methods

Руководства

Управление LDAP

Обзор LDAP

Поддерживаемые типы LDAP

Терминология LDAP

Добавление LDAP

Примеры конфигурации LDAP

Синхронизация пользователей LDAP

Соответствующие операции

Управление OIDC

Обзор OIDC

Добавление OIDC

Добавление OIDC через YAML

Соответствующие операции

Устранение неполадок

Удаление пользователя

Описание проблемы

Решение

Введение

Содержание

Overview

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↗.

Supported Integration Methods

LDAP Integration

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-учетных данных
- Автоматически синхронизирует учетные записи пользователей организации с платформой

OIDC Integration

Платформа поддерживает интеграцию с IDP-сервисами, использующими протокол OpenID Connect (OIDC) для аутентификации пользователей третьих сторон.

Преимущества интеграции OIDC:

- Позволяет пользователям входить с помощью аккаунтов третьих сторон
- Поддерживает корпоративные IDP-сервисы
- Обеспечивает безопасную аутентификацию через протокол OIDC

> **NOTE**
>
> Для аутентификации с использованием других коннекторов, не упомянутых выше, пожалуйста, свяжитесь с технической поддержкой.

Menu

Руководства

Управление LDAP

Обзор LDAP

Поддерживаемые типы LDAP

Терминология LDAP

Добавление LDAP

Примеры конфигурации LDAP

Синхронизация пользователей LDAP

Соответствующие операции

Управление OIDC

Обзор OIDC

Добавление OIDC

Добавление OIDC через YAML

Соответствующие операции

Управление LDAP

Администраторы платформы могут добавлять, обновлять и удалять LDAP-сервисы на платформе.

Содержание

Обзор LDAP

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).

NOTE

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.

NOTE

Интеграция OpenLDAP:

- Обеспечивает аутентификацию пользователей LDAP на 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.

NOTE

Интеграция Active Directory:

- Обеспечивает аутентификацию пользователей AD на 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`

Термин	Описание	Пример
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,d`
cn (Common Name)	Общее имя	`cn=admin,dc=example,dc`
sAMAccountName/userPrincipalName	Идентификатор пользователя	`userPrincipalName=exam` `sAMAccountName=example`

Термин	Описание	Пример
objectClass: user	Класс объекта пользователя AD	`objectClass=user`
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=exa`
memberOf	Атрибут членства в группе	`memberOf=CN=Users,DC=e`

Добавление LDAP

TIP

После успешной интеграции LDAP:

- Пользователи могут входить на платформу с использованием своих корпоративных учетных записей
- Множественное добавление одного и того же LDAP перезапишет ранее синхронизированных пользователей

Предварительные требования

Перед добавлением LDAP подготовьте следующую информацию:

- Адрес LDAP-сервера
- Имя пользователя администратора
- Пароль администратора
- Другие необходимые параметры конфигурации

Шаги

1. В левой навигационной панели нажмите **Users** > **IDPs**
2. Нажмите **Add LDAP**
3. Настройте следующие параметры:

Основная информация

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 Box Username Prompt	Сообщение-приглашение для ввода имени пользователя (например, "Пожалуйста, введите ваше имя пользователя")

Настройки поиска

NOTE

Назначение настроек поиска:

- Соответствие LDAP-записям пользователей по заданным условиям

- Извлечение ключевых атрибутов пользователей и групп
- Отображение атрибутов LDAP на атрибуты пользователей платформы

Параметр	Описание
Object Type	ObjectClass для пользователей: - OpenLDAP: `inetOrgPerson` - Active Directory: `organizationalPerson` - Группы: `posixGroup`
Login Field	Атрибут, используемый в качестве имени пользователя для входа: - OpenLDAP: `mail` (адрес электронной почты) - Active Directory: `userPrincipalName`
Filter Conditions	Условия фильтра LDAP для фильтрации пользователей/групп Пример: `(&(cn=John*)(givenName=*xq*))`
Search Starting Point	Базовый DN для поиска пользователей/групп (например, `dc=example,dc=org`)
Search Scope	Область поиска: - `sub`: весь поддерево каталога - `one`: один уровень ниже начальной точки
Login Attribute	Уникальный идентификатор пользователя: - OpenLDAP: `uid` - Active Directory: `distinguishedName`
Name Attribute	Атрибут имени объекта (по умолчанию: `cn`)
Email Attribute	Атрибут электронной почты: - OpenLDAP: `mail` - Active Directory: `userPrincipalName`
Group Member Attribute	Идентификатор участника группы (по умолчанию: `uid`)

Параметр	Описание
Group Attribute	Атрибут связи с группой пользователей (по умолчанию: `memberuid`)

4. В разделе **IDP Service Configuration Validation**:

   - Введите действительное имя пользователя и пароль LDAP-аккаунта
   - Имя пользователя должно совпадать с настройкой **Login Field**
   - Нажмите для проверки конфигурации

5. (Опционально) Настройте **LDAP Auto-Sync Policy**:

   - Включите переключатель **Auto-Sync Users**
   - Установите правила синхронизации
   - Используйте онлайн-инструмент↗ для проверки выражений CRON

6. Нажмите **Add**

> **NOTE**
>
> После добавления LDAP:
>
> - Пользователи могут входить до синхронизации
> - Информация о пользователях синхронизируется автоматически при первом входе
> - Автоматическая синхронизация происходит согласно настроенным правилам

Примеры конфигурации LDAP

Конфигурация LDAP-коннектора

Ниже приведён пример настройки LDAP-коннектора:

```yaml
apiVersion: dex.coreos.com/v1
kind: Connector
id: ldap        # Connector ID
name: ldap      # Connector display name
type: ldap      # Connector type is LDAP
metadata:
  name: ldap
  namespace: cpaas-system
spec:
  config:
    # LDAP server address and port
    host: ldap.example.com:636
    # DN and password for the service account used by the connector.
    # This DN is used to search for users and groups.
    bindDN: uid=serviceaccount,cn=users,dc=example,dc=com
    # Service account password, required when creating a connector.
    bindPW: password

    # Login account prompt. For example, username
    usernamePrompt: SSO Username

    # User search configuration
    userSearch:
      # Start searching from the base DN
      baseDN: cn=users,dc=example,dc=com
      # LDAP query statement, used to search for users.
      # For example: "(&(objectClass=person)(uid=<username>))"
      filter: (&(objectClass=organizationalPerson))

      # The following fields are direct mappings of user entry attributes.
      # User ID attribute
      idAttr: uid
      # Required. Attribute to map to email
      emailAttr: mail
      # Required. Attribute to map to username
      nameAttr: cn
      # Login username attribute
      # Filter condition will be converted to "(<attr>=<username>)", such as
(uid=example).
      username: uid

      # Extended attributes
      # phoneAttr: phone
```

```
    # Group search configuration
    groupSearch:
      # Start searching from the base DN
      baseDN: cn=groups,dc=freeipa,dc=example,dc=com
      # Group filter condition
      # "(&(objectClass=group)(member=<user uid>))".
      filter: "(objectClass=group)"
      # User group matching field
      # Group attribute
      groupAttr: member
      # User group member attribute
      userAttr: uid
      # 组显示名称
      nameAttr: cn
```

Примеры фильтров пользователей

```
# 1. Базовый фильтр: Найти всех пользователей
(&(objectClass=person))

# 2. Комбинация нескольких условий: Найти пользователей в конкретном отделе
(&(objectClass=person)(departmentNumber=1000))

# 3. Найти активных пользователей (Active Directory)
(&(objectClass=user)(!(userAccountControl:1.2.840.113556.1.4.803:=2)))

# 4. Найти пользователей с определённым доменом электронной почты
(&(objectClass=person)(mail=*@example.com))

# 5. Найти участников конкретной группы
(&(objectClass=person)(memberOf=cn=developers,ou=groups,dc=example,dc=com))

# 6. Найти недавно вошедших пользователей (Active Directory)
(&(objectClass=user)(lastLogon>=20240101000000.0Z))

# 7. Исключить системные аккаунты
(&(objectClass=person)(!(uid=admin))(!(uid=system)))

# 8. Найти пользователей с определённым атрибутом
(&(objectClass=person)(mobile=*))

# 9. Найти пользователей в нескольких отделах
(&(objectClass=person)(|(ou=IT)(ou=HR)(ou=Finance)))

# 10. Пример сложной комбинации условий
(&
  (objectClass=person)
  (|(department=IT)(department=Engineering))
  (!(title=Intern))
  (manager=cn=John Doe,ou=People,dc=example,dc=com)
)
```

Примеры конфигурации поиска групп

```
# 1. Базовый фильтр: Найти все группы
(objectClass=groupOfNames)

# 2. Найти группы с определённым префиксом
(&(objectClass=groupOfNames)(cn=dev-*))

# 3. Найти непустые группы
(&(objectClass=groupOfNames)(member=*))

# 4. Найти группы с определённым участником
(&(objectClass=groupOfNames)(member=uid=john,ou=People,dc=example,dc=com))

# 5. Найти вложенные группы (Active Directory)
(&(objectClass=group)(|(groupType=-2147483646)(groupType=-2147483644)))

# 6. Найти группы с определённым описанием
(&(objectClass=groupOfNames)(description=*admin*))

# 7. Исключить системные группы
(&(objectClass=groupOfNames)(!(cn=system*)))

# 8. Найти группы с определёнными участниками
(&(objectClass=groupOfNames)(|(cn=admins)(cn=developers)(cn=operators)))

# 9. Найти группы в конкретном OU
(&(objectClass=groupOfNames)(ou=IT))

# 10. Пример сложной комбинации условий
(&
  (objectClass=groupOfNames)
  (|(cn=prod-*)(cn=dev-*))
  (!(cn=deprecated-*))
  (owner=cn=admin,dc=example,dc=com)
)
```

Примеры операторов AND(&) и OR(|) в LDAP-фильтрах

```
# Оператор AND (&) - все условия должны быть выполнены
# Синтаксис: (&(condition1)(condition2)(condition3)...)

# Пример AND с несколькими атрибутами
(&
  (objectClass=person)
  (mail=*@example.com)
  (title=Engineer)
  (manager=*)
)

# Оператор OR (|) - должно быть выполнено хотя бы одно условие
# Синтаксис: (|(condition1)(condition2)(condition3)...)

# Пример OR с несколькими атрибутами
(|
  (department=IT)
  (department=HR)
  (department=Finance)
)

# Комбинирование AND и OR
(&
  (objectClass=person)
  (|
    (department=IT)
    (department=R&D)
  )
  (employeeType=FullTime)
)

# Сложная комбинация условий
(&
  (objectClass=person)
  (|
    (&
      (department=IT)
      (title=*Engineer*)
    )
    (&
      (department=R&D)
      (title=*Developer*)
    )
```

```
    )
    (!(status=Inactive))
    (|(manager=*)(isManager=TRUE))
  )
```

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 получат 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.

Порядок действий

1. В левой навигационной панели нажмите **Users** > **IDPs**.
2. Нажмите на ***имя LDAP***, для которого хотите выполнить ручную синхронизацию.
3. В правом верхнем углу нажмите **Actions** > **Sync user**.
4. Нажмите **Sync**.

Примечания: Если вы вручную закроете диалоговое окно синхронизации, появится диалог подтверждения закрытия. После закрытия диалога система продолжит синхронизацию пользователей. Если вы остаетесь на странице списка пользователей, получите обратную связь о результате синхронизации. Если покинете страницу списка пользователей, результат синхронизации не будет получен.

Соответствующие операции

Вы можете нажать на
справа на странице списка или нажать **Actions** в правом верхнем углу на странице деталей, чтобы при необходимости обновить или удалить LDAP.

Операция	Описание
Обновить LDAP	Обновить конфигурацию добавленного LDAP или **LDAP Auto-Sync Policy**. **Примечание**: 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 мусорных данных выполните операцию очистки недействительных пользователей.
Удалить LDAP	После удаления LDAP все пользователи, синхронизированные с платформой через этот LDAP, получат статус **Invalid** (связь между пользователями и ролями сохраняется), и они не смогут войти на платформу. После повторной интеграции необходимо повторно выполнить синхронизацию для активации пользователей. **Совет**: После удаления IDP, если необходимо удалить пользователей и группы пользователей, синхронизированные с платформой через LDAP, установите флажок **Clean IDP Users and User Groups** под окном подтверждения.

Управление OIDC

Платформа поддерживает протокол OIDC (OpenID Connect), позволяющий администраторам платформы входить в систему с использованием сторонних аккаунтов после добавления конфигурации OIDC. Администраторы платформы также могут обновлять и удалять настроенные сервисы OIDC.

Содержание

Обзор OIDC

OIDC (OpenID Connect) — это стандартный протокол аутентификации личности, 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 личности 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 расширяем, что позволяет участникам использовать дополнительные функции, такие как шифрование данных личности, обнаружение OpenID Provider и управление сессиями, когда это имеет смысл.

Для получения дополнительной информации обратитесь к официальной документации OIDC ↗.

Добавление OIDC

Добавив OIDC, вы можете использовать сторонние аккаунты платформы для входа на платформу.

Примечание: После успешного входа пользователей OIDC на платформу платформа будет использовать атрибут email пользователя в качестве уникального идентификатора. Пользователи сторонних платформ с поддержкой OIDC должны иметь атрибут **email**; в противном случае они не смогут войти на платформу.

Порядок действий

1. В левой навигационной панели нажмите **Users** > **IDPs**.
2. Нажмите **Add OIDC**.
3. Настройте параметры **Basic Information**.
4. Настройте параметры **OIDC Server Configuration**:
   - **Identity Provider URL**: URL издателя, который является адресом 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 на стороннюю платформу, который представляет собой URL издателя dex + `/callback`.
   - **Logout URL**: адрес, на который пользователь будет перенаправлен после выполнения операции **Logout**. Если пусто, адрес выхода будет начальной страницей входа платформы.
5. В области **IDP Service Configuration Validation** введите **Username** и **Password** действительной учетной записи OIDC для проверки конфигурации.

Совет: Если имя пользователя и пароль введены неверно, при добавлении будет выдана ошибка с указанием недействительных учетных данных, и OIDC не сможет быть добавлен.

6. Нажмите **Create**.

Добавление OIDC через YAML

Помимо конфигурации через форму, платформа также поддерживает добавление OIDC через YAML, что позволяет более гибко настраивать параметры аутентификации, сопоставление claims, синхронизацию групп пользователей и другие расширенные функции.

Пример: Конфигурация OIDC коннектора

Следующий пример демонстрирует, как настроить OIDC коннектор для интеграции с сервисами аутентификации личности OIDC. Этот пример конфигурации подходит для следующих сценариев:

1. Необходима интеграция OIDC в качестве сервера аутентификации личности.
2. Требуется поддержка синхронизации информации о группах пользователей.
3. Нужно настроить адрес перенаправления после выхода из системы.
4. Необходимо настроить конкретные области (scopes) OIDC.
5. Требуется настроить сопоставление claims.

```yaml
apiVersion: dex.coreos.com/v1
kind: Connector
# Connector basic information
id: oidc              # Уникальный идентификатор коннектора
name: oidc            # Отображаемое имя коннектора
type: oidc            # Тип коннектора - OIDC
metadata:
  annotations:
    cpaas.io/description: "11"  # Описание коннектора
  name: oidc
  namespace: cpaas-system
spec:
  config:
    # Конфигурация OIDC сервера
    # Настройка информации для подключения к серверу, включая адрес сервера, учетные данные клиента и адрес обратного вызова
    issuer: http://auth.com/auth/realms/master              # Адрес OIDC сервера
    clientID: dex                                           # Client ID
    # Секретный ключ сервисного аккаунта, актуален при первом создании ресурсов Connector
    clientSecret: xxxxxxx
    redirectURI: https://example.com/dex/callback           # Адрес обратного вызова, должен совпадать с адресом, зарегистрированным у клиента OIDC

    # Конфигурация безопасности
    # Настройка проверки SSL и способа получения информации о пользователе
    insecureSkipVerify: true                                # Пропускать ли проверку SSL, рекомендуется установить в false в продакшн-среде
    getUserInfo: false                                      # Получать ли дополнительную информацию о пользователе через endpoint UserInfo

    # Конфигурация выхода из системы
    # Настройка адреса перенаправления после выхода пользователя
    logoutURL: https://test.com                             # Адрес перенаправления при выходе, можно настроить на страницу, куда будет переход после выхода пользователя

    # Конфигурация областей (scopes)
    # Настройка необходимых областей авторизации, убедитесь, что сервер OIDC поддерживает эти области
    scopes:
      - openid                                              # Обязательно, используется для базовой аутентификации OIDC
      - profile                                             # Опционально, используется
```

```
для получения базовой информации о пользователе
      - email                                          # Опционально, используется
для получения email пользователя

    # Конфигурация сопоставления claims
    # Настройка соответствия между возвращаемыми OIDC claims и атрибутами
пользователя платформы
    claimMapping:
      email: email                                     # Сопоставление email,
используется для уникальной идентификации пользователя
      groups: groups                                   # Сопоставление групп
пользователя, используется для структуры организации
      phone: ""                                        # Сопоставление телефона,
опционально
      preferred_username: preferred_username           # Сопоставление имени
пользователя, используется для отображаемого имени

    # Конфигурация дополнительных claimextra
    # Внешние пользовательские поля будут динамически добавлены в поле spec.extra
объекта пользователя
    claimExtra:
      - field: xxx           # Имя пользовательского поля
        type: string         # Тип поля соответствует определению типа языка golang.
Например: string, int, bool, map[string]string, []string, []int

    # Конфигурация групп пользователей
    # Настройка параметров синхронизации групп пользователей, убедитесь, что токен
содержит информацию о группах
    groupsKey: groups                                  # Указание имени ключа
информации о группах
    insecureEnableGroups: false                        # Включена ли функция
синхронизации групп
```

Соответствующие операции

Вы можете нажать на
справа на странице списка или нажать **Actions** в правом верхнем углу на странице деталей, чтобы при необходимости обновить или удалить OIDC.

Операция	Описание
Обновить OIDC	Обновить добавленную конфигурацию OIDC. После обновления информации 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 на платформу через этот OIDC, получат статус **Invalid** (связь между пользователями и ролями сохраняется), и они не смогут войти на платформу. После повторной интеграции пользователи могут быть активированы успешным входом на платформу. **Совет**: После удаления IDP, если необходимо удалить пользователей и группы пользователей, синхронизированные на платформу через OIDC, отметьте флажок **Clean IDP Users and User Groups** под всплывающим окном.

Menu

Устранение неполадок

Удаление пользователя

Содержание

Описание проблемы

Проблема: При создании или синхронизации нового пользователя система сообщает, что пользователь уже существует. Как следует поступить в этом случае?

По соображениям безопасности платформа не позволяет создавать новых пользователей (как локальных, так и IDP-пользователей) с именами, совпадающими с ранее удалёнными пользователями. Это ограничение распространяется на:

- Создание новых локальных пользователей с именами, совпадающими с удалёнными пользователями
- Синхронизацию IDP-пользователей с именами, совпадающими с удалёнными пользователями

После обновления до текущей версии вы можете столкнуться с этой проблемой при:

- Создании новых пользователей с именами, совпадающими с пользователями, удалёнными до обновления
- Синхронизации новых пользователей с именами, совпадающими с пользователями, удалёнными до обновления

Решение

Для устранения этой проблемы необходимо очистить информацию об удалённых пользователях, выполнив определённые скрипты на узлах управления вашего глобального кластера.

Очистка удалённых IDP-пользователей

Выполните следующую команду на любом узле управления вашего глобального кластера:

```
kubectl delete users -l 'auth.cpaas.io/user.connector_id=<IDP
Name>,auth.cpaas.io/user.state=deleted'
```

Пример:

```
kubectl delete users -l
'auth.cpaas.io/user.connector_id=github,auth.cpaas.io/user.state=deleted'
```

Очистка удалённых локальных пользователей

Выполните последовательно два скрипта на любом узле управления вашего глобального кластера:

1. Очистка паролей пользователей:

```
kubectl get users -l
'auth.cpaas.io/user.connector_id=local,auth.cpaas.io/user.state=deleted' | awk '{print
$1}' | xargs kubectl delete password -n cpaas-system
```

2. Очистка пользователей:

```
kubectl delete users -l
'auth.cpaas.io/user.connector_id=local,auth.cpaas.io/user.state=deleted'
```

Menu

Политика пользователя

Введение

Платформа предоставляет комплексные политики безопасности пользователей для повышения безопасности входа и защиты от вредоносных атак.

Содержание

Обзор

Платформа поддерживает следующие политики безопасности:

- Управление безопасностью паролей
- Отключение учетных записей пользователей
- Блокировка учетных записей пользователей
- Уведомления пользователей
- Контроль доступа

Настройка политики безопасности

Шаги

1. В левой навигационной панели нажмите **User Role Management** > **User Security Policy**
2. Нажмите **Update** в правом верхнем углу
3. Настройте политики безопасности по необходимости
4. Нажмите **Update** для сохранения изменений

WARNING

Примечания по настройке политики:

- Отметьте галочкой политику для её включения
- Снимите галочку для отключения политики
- Отключённые политики сохраняют свои данные конфигурации
- При повторном включении политики восстанавливаются предыдущие настройки

Доступные политики

Политика	Описание
User Authentication Policy	Включает двойную аутентификацию для входа по паролю: - Пользователи получают коды подтверждения через указанные методы уведомлений - Поддерживает различные серверы уведомлений (например, Enterprise Communication Tool Server)
Password Security Policy	Управляет требованиями к паролям: Первый вход: - Обязательная смена пароля при первом входе на платформу

Политика	Описание
	Регулярное обновление: - Требуется смена пароля после заданного периода (например, 90 дней) - Вход запрещён до обновления пароля
User Disablement Policy	Автоматически отключает неактивные аккаунты: - Срабатывает после заданного периода отсутствия входа
User Locking Policy	Защищает от атак перебором паролей: Условия блокировки: - Срабатывает после заданного количества неудачных попыток входа в течение 24 часов Длительность блокировки: - Аккаунт остаётся заблокированным в течение заданного времени в минутах - Автоматически разблокируется после истечения периода блокировки
Notification Policy	Управляет уведомлениями пользователей: - Отправляет первоначальный пароль по электронной почте после создания пользователя
Access Control	Управляет сессиями пользователей и доступом: Управление сессиями: - Автоматический выход из неактивных сессий после заданного времени - Ограничение максимального количества одновременных онлайн-пользователей Контроль браузера: - Завершение сессии при закрытии всех вкладок продукта - Запрет на множественные входы с одного клиента :::note

Политика	Описание
	Важные замечания: - Контроль доступа влияет только на новые входы после обновления политики - Восстановление вкладок браузера может не привести к завершению сессии - При запрете повторного входа разрешён только последний вход с клиента :::

Menu

Мультиарендность (Project)

Введение

Введение

Проект

Пространства имён (Namespaces)

Взаимосвязь между кластерами, проектами и пространствами имён

Руководства

Создание проекта

Процедура

Управление квотами проекта

Что такое ProjectQuota?

Как это работает

Когда использовать ProjectQuota

Ключи квот и единицы измерения

Советы по стратегии распределения

Лучшие практики и часто задаваемые вопросы

Управление проектом

Обновление основной информации проекта

Удаление проекта

Управление кластером проекта

Введение

Добавить кластер

Удалить кластер

Управление участниками проекта

Импорт участников

Удаление участников

Введение

Содержание

Проект

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 можно достичь:

- Изоляции ресурсов между проектными командами
- Управления квотами внутри арендаторов
- Эффективного распределения и контроля 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 пространств имён включают:

- В рамках проекта можно создавать несколько пространств имён
- Общая квота ресурсов всех пространств имён не может превышать квоту проекта
- Квоты ресурсов выделяются более детально 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.

Menu

Руководства

Создание проекта

Процедура

Управление квотами проекта

Что такое ProjectQuota?

Как это работает

Когда использовать ProjectQuota

Ключи квот и единицы измерения

Советы по стратегии распределения

Лучшие практики и часто задаваемые вопросы

Управление проектом

Обновление основной информации проекта

Удаление проекта

Управление кластером проекта

Введение

Добавить кластер

Удалить кластер

Управление участниками проекта

Создание проекта

Прежде чем ваша команда начнет работу, вы можете создать проект на основе существующих ресурсов кластера на платформе. Проект будет изолирован от других проектов (тенантов) как по ресурсам, так и по персоналу. При создании проекта вы можете выделить ресурсы в соответствии с масштабом проекта и реальными бизнес-потребностями. Проект может использовать ресурсы из нескольких кластеров на платформе.

WARNING

При создании проекта платформа автоматически создаст namespace с таким же именем, как у проекта, в связанных кластерах для изоляции ресурсов на уровне платформы. Пожалуйста, не изменяйте этот namespace и его ресурсы.

Содержание

Процедура

1. В представлении **Project Management** нажмите **Create Project**.
2. На странице **Basic information** настройте следующие параметры:

Параметр	Описание
Name	Имя проекта, которое не может совпадать с именем существующего проекта или с любым именем из черного списка

Параметр	Описание
	имен проектов. В противном случае проект создать нельзя. **Примечание**: Черный список имен проектов включает специальные имена namespace в кластерах платформы: `cpaas-system` , `cert-manager` , `default` , `global-credentials` , `kube-ovn` , `kube-public` , `kube-system` , `nsx-system` , `alauda-system` , `kube-federation-system` , `ALL-ALL` и `true` .
Cluster	Кластер(ы), связанные с проектом, в которых администратор может выделять квоты ресурсов. Нажмите на выпадающий список, чтобы выбрать один или несколько кластеров. **Примечание**: Кластеры в аномальном состоянии выбрать нельзя.

3. Нажмите **Next** и на шаге настройки квот проекта ознакомьтесь с разделом Manage Resource Quotas, чтобы задать квоты ресурсов, выделяемые текущему проекту для выбранных кластеров.

4. Нажмите **Create Project**.

Управление квотами проекта

В этом руководстве объясняется, как ACP расширяет Kubernetes ResourceQuota с помощью агрегированной квоты на уровне проекта (ProjectQuota). ProjectQuota позволяет ограничить сумму ResourceQuota по всем namespace в проекте, что позволяет планировать и управлять ресурсами на уровне проекта, при этом делегируя лимиты отдельным namespace.

Содержание

Что такое ProjectQuota?

- ResourceQuota (встроенный в Kubernetes) ограничивает ресурсы на уровне namespace (CPU, память, количество объектов и т.д.). Для понятий, ключей и использования см.:
  - Resource Quotas
- ProjectQuota задаёт верхний предел для всего проекта: сумма всех ResourceQuota по namespace в проекте не должна превышать жёсткие лимиты проекта по тем же ключам.

Проще говоря: ResourceQuota ограничивает один namespace; ProjectQuota ограничивает сумму по всем namespace в проекте.

Как это работает

- Порядок работы: сначала определите или скорректируйте ProjectQuota, затем распределяйте ResourceQuota по namespace в рамках бюджета проекта.
- Область действия: ProjectQuota применяется к платформенному проекту и управляет всеми namespace, которые к нему принадлежат.
- Агрегированное применение при приёме запроса:
  - При создании или обновлении ResourceQuota namespace платформа вычисляет сумму по тем же ключам (например, `limits.cpu`, `requests.memory`, `pods`) по всем namespace проекта, включая вносимое изменение.
  - Запрос разрешается только если новая сумма остаётся меньше или равна соответствующим жёстким лимитам ProjectQuota. В противном случае изменение отклоняется с объяснением ошибки.
- Модель исполнения:
  - ProjectQuota ограничивает то, что можно выделить через ResourceQuota namespace (предварительное выделение), а не текущее использование ресурсов. Фактическое потребление регулируется ResourceQuota каждого namespace и планировщиком.

Когда использовать ProjectQuota

- Управление бюджетом/ёмкостью по проекту: выделить фиксированный бюджет CPU/памяти/объектов и затем распределить его по namespace.
- Много командные или много средовые проекты (например, `dev` / `staging` / `prod`), которые используют общий верхний предел.

- Предотвращение «дрейфа» квот: поддерживать один «большой контейнер» на уровне проекта, чтобы квоты namespace не увеличивались незаметно со временем.

Ключи квот и единицы измерения

ProjectQuota поддерживает те же распространённые ключи, что и ResourceQuota (неполный список):

- Вычислительные ресурсы и память: `limits.cpu`, `limits.memory`, `requests.cpu`, `requests.memory`
- Количество рабочих нагрузок/объектов: `pods`, `services`, `configmaps`, `secrets`, `pvc` и другие

Единицы и правила подсчёта:

- CPU измеряется в ядрах (например, `2`, `500m`)
- Память измеряется в байтах (например, `8Gi`)
- Ключи, связанные с объектами, считаются целыми числами

Если сумма соответствующих ключей по всем namespace приближается или превышает жёсткий лимит ProjectQuota, ACP блокирует создание или расширение ResourceQuota для этого ключа.

Советы по стратегии распределения

- Сначала определите «большой контейнер» проекта (ProjectQuota), затем разделите его на ResourceQuota по namespace для команд/сред.
- Оставляйте запас 10% - 30% для пиков и эластичного масштабирования.
- Регулярно проверяйте: возвращайте неиспользуемые квоты и перераспределяйте; увеличивайте квоты для постоянно ограниченных namespace и корректируйте общий лимит проекта.

Лучшие практики и часто задаваемые вопросы

- В: При увеличении `limits.memory` в namespace возникает ошибка о превышении квоты проекта. Почему?
  - О: Жёсткий лимит ProjectQuota по этому ключу будет превышен запрашиваемым изменением. Уменьшите квоты в других namespace или сначала увеличьте лимит проекта, затем повторите изменение namespace.
- В: Я увеличил ProjectQuota, но рабочие нагрузки всё равно не запускаются.
  - О: Убедитесь, что ResourceQuota каждого namespace также увеличена соответствующим образом, и проверьте доступную ёмкость кластера/узлов.
- Рекомендация: Управляйте ProjectQuota в рамках обычного процесса контроля изменений, согласованного с планированием ёмкости (узлы/хранилище) и управлением бюджетом.

Управление проектом

В этом руководстве объясняется, как обновить основную информацию и квоты проекта для указанного проекта или удалить проект.

Содержание

Обновление основной информации проекта

Обновите основную информацию для указанного проекта, такую как отображаемое имя и описание.

Порядок действий

1. В представлении **Управление проектами** нажмите на имя проекта, который нужно обновить.
2. В левой навигационной панели нажмите **Детали**.
3. В правом верхнем углу нажмите **Действия** > **Обновить основные данные**.
4. Измените или введите **Отображаемое имя** и **Описание**.
5. Нажмите **Обновить**.

Удаление проекта

Удалите проекты, которые больше не используются.

> **WARNING**
>
> После удаления проекта ресурсы, занятые проектом в кластере, будут освобождены.

Порядок действий

1. В представлении **Управление проектами** нажмите на имя проекта, который нужно удалить.
2. В левой навигационной панели нажмите **Детали**.
3. В правом верхнем углу нажмите **Действия** > **Удалить проект**.
4. Введите имя проекта и нажмите **Удалить**.

Управление кластером проекта

В этом руководстве объясняется, как управлять ассоциациями кластеров для проекта. Вы можете добавлять кластеры, чтобы выделить их ресурсы проекту, или удалять кластеры, чтобы вернуть выделенные ресурсы.

Содержание

Введение

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

Добавить кластер

Добавьте кластер в проект и установите его квоту ресурсов.

Процедура

1. В представлении **Project Management** нажмите на название проекта, в который хотите добавить кластер.
2. В левой навигационной панели нажмите **Details**.
3. В правом верхнем углу нажмите **Actions** > **Add Cluster**.
4. Выберите кластер и установите квоту ресурсов, которая будет выделена текущему проекту. Можно настроить следующие ресурсы:
   - CPU (ядра)
   - Память (Gi)
   - Хранилище (Gi)
   - Количество PVC (число)
   - Pods (число)
   - vGPU (виртуальный GPU)/MPS/pGPU (физический GPU, ядра)
   - Квота видеопамяти

NOTE

- Квота ресурсов GPU может быть настроена только при развертывании GPU-плагинов в кластере.

  Когда ресурсы GPU — это **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.

- 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.
- 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.

1. Нажмите **Add**.

Удалить кластер

Удалите кластер, связанный с проектом.

WARNING

- 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.

Процедура

1. В представлении **Project Management** нажмите на название проекта, из которого хотите удалить кластер.
2. В левой навигационной панели нажмите **Details**.
3. В правом верхнем углу нажмите **Actions** > **Remove Cluster**.

4. В появившемся диалоговом окне **Remove Cluster** введите имя удаляемого кластера, затем нажмите кнопку **Remove** для успешного удаления кластера.

Управление участниками проекта

В этом руководстве объясняется, как управлять участниками проекта, включая импорт участников и назначение ролей, связанных с проектом.

Содержание

Импорт 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 и условия

- Если на 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 с правами системного администратора по умолчанию и текущий вошедший пользователь не могут быть импортированы

Процедура

1. В представлении **Project Management** нажмите на имя проекта, которым хотите управлять.
2. В левой навигационной панели выберите **Members**.
3. Нажмите **Import Member**.
4. Выберите либо **Member List**, либо **OIDC Users**.

Импорт из списка участников

Вы можете импортировать всех пользователей или выбрать конкретных пользователей из списка участников.

TIP

Используйте выпадающее меню групп пользователей в правом верхнем углу и поле поиска для фильтрации пользователей по имени.

Чтобы импортировать всех пользователей:

1. Выберите **Member List**.
2. 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**.
3. Нажмите **Import All**.

Чтобы импортировать конкретных пользователей:

1. Выберите **Member List**.
2. Отметьте одного или нескольких пользователей с помощью флажков.
3. 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**.
4. Нажмите **Import**.

Импорт пользователей OIDC

1. Выберите **OIDC Users**.
2. Нажмите **Add** для создания записи участника (повторите для нескольких участников).
3. Введите имя пользователя, прошедшего аутентификацию через OIDC, в поле **Name**.

WARNING

Убедитесь, что имя пользователя соответствует аккаунту, который может быть аутентифицирован настроенной системой OIDC, иначе вход будет невозможен.

4. Выберите роль из выпадающего меню **Roles**.
   Если роль требует указания пространства имён, выберите его в выпадающем меню **Namespaces**.
5. Нажмите **Import**.

После успешного импорта вы сможете увидеть:

- Участника в списке участников проекта
- Пользователя в разделе **Platform Management > Users**
  - Источник отображается как "-" до первого входа/синхронизации
  - Источник обновляется после успешного входа/синхронизации

Удаление участников

Удалите участника проекта, чтобы отозвать его права.

Процедура

1. В представлении **Project Management** нажмите на имя проекта.
2. В левой навигационной панели выберите **Members**.

> **TIP**
>
> Используйте выпадающий список рядом с полем поиска для фильтрации участников по **Name**, **Display name** или **User Group**.

3. Нажмите **Remove** рядом с участником, которого хотите удалить.
4. Подтвердите удаление в появившемся диалоговом окне.

Menu

Аудит

Введение

Функция аудита платформы предоставляет упорядоченные по времени записи операций, связанных с пользователями и безопасностью системы. Это помогает анализировать конкретные проблемы и быстро решать возникающие в кластерах, пользовательских приложениях и других областях вопросы.

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

Содержание

Требования

Ваша учетная запись должна иметь права управления платформой или права аудита платформы.

Процедура

1. В левой навигационной панели нажмите **Auditing**.
2. Выберите область аудита на вкладках:
   - **User Operations**: Просмотр записей операций пользователей, вошедших в платформу
   - **System Operations**: Просмотр записей системных операций (операторы начинаются с `system:`)
3. Настройте условия запроса для фильтрации событий аудита:

Условие запроса	Описание
Operator	Имя пользователя или системной учетной записи оператора (по умолчанию: `All`)
Actions	Тип операции (create, update, delete, manage, rollback, stop и др., по умолчанию: `All`)
Clusters	Кластер, содержащий управляемый ресурс (по умолчанию: `All`)
Resource Type	Тип управляемого ресурса (по умолчанию: `All`)
Resource Name	Имя управляемого ресурса (поддерживается нечеткий поиск)

4. Нажмите **Search**.

TIP

- Используйте выпадающий список **Time Range** для установки временного диапазона аудита (по умолчанию: `Last 30 Minutes`). Можно выбрать предустановленный диапазон или задать свой.
- 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 учетной записи оператора
Actions	Тип операции (create, update, delete, manage, rollback, stop и др.)
Resource Name/Type	Имя и тип управляемого ресурса
Clusters	Кластер, содержащий управляемый ресурс
Namespaces	Namespace, содержащий 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)
Operation Time	Временная метка операции
Details	Нажмите кнопку **Details**, чтобы просмотреть полный аудиторский запись в формате JSON в диалоговом окне **Audit Details**

Menu

Телеметрия

Установка

ACP Telemetry — это сервис платформы, который собирает телеметрические данные с кластеров для онлайн-операций и обслуживания. Он собирает системные метрики и загружает их в Alauda Cloud для мониторинга и анализа.

Содержание

Требования

Перед установкой убедитесь, что:

- В Alauda Container Platform имеется действующая лицензия
- Глобальный кластер имеет доступ в интернет

Шаги установки

1. Перейдите в **Administrator**
2. В левой навигационной панели нажмите **Marketplace** > **Cluster Plugins**
3. Выберите кластер **global** в верхней навигационной панели

4. Найдите **ACP Telemetry** и нажмите для просмотра деталей
5. Нажмите **Install** для развертывания плагина

Включение онлайн-операций

1. В левой навигационной панели нажмите **System Settings** > **Platform Maintenance**
2. Нажмите кнопку **On** для **Online Operations**

Шаги удаления

1. Выполните шаги 1-4 из процесса установки, чтобы найти плагин
2. Нажмите **Uninstall** для удаления плагина

Menu

Сертификаты

Automated Kubernetes Certificate Rotation

Это руководство поможет вам установить, понять и управлять Kubernetes Certificate Rotator в ACP для автоматизации ротации сертификатов Kubernetes в ваших кластерах.

Содержание

Installation

Смотрите Cluster Plugin для инструкций по установке.

> **Примечание:**
>
> - В настоящее время поддерживаются:
>
> - On-Premises кластеры
> - DCS кластеры

How it works

Этот плагин обрабатывает автоматическую ротацию следующих сертификатов.

Certificate file	Function	Node Type
apiserver.crt	Серверный сертификат для kube-apiserver	Control Plane Node
apiserver-etcd-client.crt	Клиентский сертификат для kube-apiserver для доступа к etcd	Control Plane Node
apiserver-kubelet-client.crt	Клиентский сертификат для kube-apiserver для доступа к kubelet	Control Plane Node
front-proxy-client.crt	Клиентский сертификат для kube-apiserver для доступа к агрегированным API серверам	Control Plane Node
etcd/server.crt	Серверный сертификат для etcd	Control Plane Node
etcd/peer.crt	Сертификат для взаимодействия между членами etcd	Control Plane Node
/root/.kube/config, admin.conf, super-admin.conf	Клиентский сертификат в kubeconfig для администрирования кластера	Control Plane Node
controller-manager.conf	Клиентский сертификат в kubeconfig для kube-controller-manager	Control Plane Node
scheduler.conf	Клиентский сертификат в kubeconfig для kube-scheduler	Control Plane Node
kubelet.crt	Серверный сертификат для kubelet	All Nodes

Certificate file	Function	Node Type
kubelet-client-current.pem	Клиентский сертификат для kubelet (ссылается в kubelet.conf)	All Nodes

Rotation Process

1. **Загрузка информации о сертификатах**

   На первом этапе собирается метаданные для всех целевых сертификатов. Поскольку эти сертификаты хранятся в разных путях на хосте, их содержимое считывается из соответствующих файлов. Для этого на целевом узле создаётся временный Pod с примонтированными каталогами сертификатов, что позволяет Pod'у прочитать информацию. Информация о сертификатах собирается один раз в день. Детали сертификатов (пути, срок действия) хранятся в ConfigMap `cpaas-system/node-local-certs-<node-name>`. Зашифрованный CA сертификат хранится в Secret `cpaas-system/kubernetes-ca`.

2. **Условие запуска ротации**

   Поля `notBefore` и `notAfter` сертификата указывают период его действия. Ротация запускается, если оставшийся срок действия меньше 20% или 30 дней.

3. **Очередь ротации**

   Сертификаты, требующие ротации, помещаются в очередь на обработку. Программа ротации оценивает недавние действия по ротации и срочность ожидающих задач, чтобы решить, обрабатывать их немедленно или нет. Это предотвращает возможные проблемы со здоровьем кластера из-за одновременной ротации нескольких сертификатов.

4. **Генерация новых сертификатов**

   Программа ротации генерирует новые сертификаты на основе внутренне хранящейся информации CA. В процессе ротации создаётся временный Pod на целевом узле с примонтированными необходимыми каталогами сертификатов, что позволяет контролируемо изменять файлы.

5. **Перезапуск компонентов**

Требующие перезапуска:

- `kube-apiserver`: необходимо перезапустить для загрузки новых сертификатов. При перезапуске он регенерирует внутренний loopback сертификат (действительный один год, используется только внутри и не может быть внешне ротирован).
- `kube-controller-manager`: необходимо перезапустить для перезагрузки kubeconfig файла.
- `kube-scheduler`: необходимо перезапустить для перезагрузки kubeconfig файла.
- `kubelet`: необходимо перезапустить для перезагрузки серверного сертификата.

Метод перезапуска: Добавить аннотации в YAML-файлы соответствующих статических Pod'ов, чтобы вызвать пересоздание Pod'ов kubelet'ом. Для перезапуска kubelet смонтируйте файловую систему хоста с `hostPID is true` и выполните в контейнере команду "systemctl restart kubelet".

Автоматическая перезагрузка:

- Etcd может автоматически перезагружать сертификаты.

6. **Сроки ротации**

- Сертификаты `kubelet`: ротация через 61 день (срок действия 91 день)
- Сертификаты контрольной плоскости: ротация через 292 дня (срок действия 365 дней)

Operation Considerations

Если `kubelet` находится в ненормальном состоянии в окне ротации и не может автоматически ротировать сертификаты, требуется ручная ротация:

Операторы должны вручную обновить сертификаты.

Выполните следующие команды для ручного обновления сертификатов:

```
cert-renew --ca-cert <ca-cert-path> --ca-key <ca-key-path> --days <days> <certificate or kubeconfig 1> <certificate or kubeconfig 2> ...
```

Например, чтобы обновить `kubelet.crt`:

```
cert-renew --ca-cert /etc/kubernetes/pki/ca.crt --ca-key /etc/kubernetes/pki/ca.key --
days 91 /etc/kubernetes/pki/kubelet.crt
```

Чтобы скачать и подготовить инструмент `cert-renew`, выполните:

```
curl "$(kubectl get services -n cpaas-system frontend -o
jsonpath='{.spec.clusterIP}'):8080/cluster-cert-rotator/download/cert-renew" -o ./cert-
renew && chmod +x ./cert-renew
```

Опционально, скачайте `renew-all.sh` для обновления всех сертификатов на узле:

```
curl "$(kubectl get services -n cpaas-system frontend -o
jsonpath='{.spec.clusterIP}'):8080/cluster-cert-rotator/download/renew-all.sh" -o
./renew-all.sh
```

cert-manager

В каждом кластере автоматически разворачивается **Certificate для cert-manager**

cert-manager — это нативный контроллер управления сертификатами Kubernetes, который автоматически генерирует и управляет TLS-сертификатами на основе ресурсов `Certificate`. Многие компоненты в Kubernetes кластерах используют cert-manager для управления своими TLS-сертификатами, обеспечивая безопасное взаимодействие.

Содержание

Обзор

Cert-manager управляет жизненным циклом сертификатов через Custom Resource Definitions (CRD) Kubernetes:

- **Certificate**: Определяет сертификаты, которые необходимо управлять
- **Issuer/ClusterIssuer**: Определяет эмитентов сертификатов
- **CertificateRequest**: Внутренний ресурс для обработки запросов на сертификаты

Как это работает

Когда создаётся ресурс `Certificate`, cert-manager автоматически:

1. Генерирует приватные ключи и запросы на подпись сертификата (CSR)
2. Получает подписанные сертификаты от указанного Issuer
3. Сохраняет сертификаты и приватные ключи в Kubernetes Secrets

Кроме того, cert-manager отслеживает срок действия сертификатов и обновляет их до истечения срока, чтобы обеспечить непрерывную доступность сервиса.

Определение сертификатов, управляемых cert-manager

Сертификаты, управляемые cert-manager, имеют соответствующие ресурсы `Secret` с типом `kubernetes.io/tls` и определёнными метками и аннотациями.

Распространённые метки и аннотации

Ресурсы `Secret`, управляемые cert-manager, обычно содержат следующие метки и аннотации:

Метки:

- `controller.cert-manager.io/fao: "true"`: Указывает, что этот Secret управляется cert-manager и включает фильтрацию кеша Secret контроллером.

Аннотации:

- `cert-manager.io/certificate-name`: Имя сертификата
- `cert-manager.io/common-name`: Общее имя сертификата
- `cert-manager.io/alt-names`: Альтернативные имена сертификата
- `cert-manager.io/ip-sans`: IP-адреса сертификата

- `cert-manager.io/issuer-kind`: Тип эмитента сертификата
- `cert-manager.io/issuer-name`: Имя эмитента сертификата
- `cert-manager.io/issuer-group`: API-группа эмитента
- `cert-manager.io/uri-sans`: URI Subject Alternative Names

Связанные ресурсы

- cert-manager Official Documentation ↗

Menu

Сертификаты OLM

Все сертификаты для компонентов **Operator Lifecycle Manager (OLM)** — включая `olm-operator`, `catalog-operator`, `packageserver` и `marketplace-operator` — автоматически управляются системой.

При установке операторов, которые определяют **webhooks** или **API services** в своем объекте **ClusterServiceVersion (CSV)**, OLM автоматически генерирует и обновляет необходимые сертификаты.

Мониторинг сертификатов

Cluster Enhancer предоставляет возможности мониторинга сертификатов, используемых в Kubernetes кластерах. Область мониторинга включает:

1. **Сертификаты компонентов Kubernetes**, включая сертификаты control plane и kubelet сервер/клиент (включая клиентские сертификаты kubeconfig)
2. **Сертификаты компонентов, работающих в кластере**, реализованные путем проверки всех Secrets с типом `kubernetes.io/tls`
3. **Серверные сертификаты, фактически используемые kube-apiserver** (включая внутренние loopback сертификаты для самодоступа) через доступ к Endpoints `kubernetes`

Пользователи могут найти и установить **Cluster Enhancer** во вкладке **Administrator**, перейдя в **Marketplace** > **Cluster Plugins** в левом навигационном меню.

Содержание

Мониторинг статуса сертификатов

Статус истечения срока действия сертификатов можно просмотреть через метрику `certificate_expires_status`. Время истечения срока действия сертификатов можно просмотреть через метрику `certificate_expires_time`.

Текущий статус сертификата и время истечения срока действия можно увидеть во вкладке **Certificate Status**. Чтобы открыть эту вкладку, перейдите во вкладку **Administrator**, затем в **Clusters > Clusters**, выберите конкретный кластер и перейдите на вкладку **Monitoring**.

Встроенные правила оповещений

Cluster Enhancer предоставляет встроенные правила оповещений `cpaas-certificates-rule` с следующими предупреждениями:

Оповещения по сертификатам Kubernetes

Правило	Уровень
Время истечения сертификата kubernetes приближается (меньше 30 дней) <= 30d и последние 1 минута	Средний
Время истечения сертификата kubernetes приближается (меньше 10 дней) <= 10d и последние 1 минута	Высокий
Сертификат kubernetes истек <= 0d и последние 1 минута	Критический

Оповещения по сертификатам компонентов платформы

Правило	Уровень
Время истечения сертификата компонентов платформы приближается (меньше 30 дней) <= 30d и последние 1 минута	Средний
Время истечения сертификата компонентов платформы приближается (меньше 10 дней) <= 10d и последние 1 минута	Высокий
Сертификат компонентов платформы истек <= 0d и последние 1 минута	Критический