# Инструменты безопасности в Deckhouse Kubernetes Platform

Онлайн-курс | 6 дней

## Аудитория курса

- DevOps-инженеры
- Системные инженеры Kubernetes
- Инженеры по безопасности (Security Engineer) / специалисты по информационной безопасности

## Цели курса

- Получить знания о принципах работы и применении инструментов безопасности в Deckhouse Kubernetes Platform (DKP)
- Приобрести базовые навыки для администрирования и эксплуатации инструментов безопасности в DKP

## Требования к участникам

- Знать Linux на уровне пользователя
- Знать основные понятия и сущности Kubernetes (pod, deployment, service, ingress)
- Уметь работать с утилитой kubectl

## Формат

- Курс состоит из теоретического материала (вводной видеолекции и онлайн-вебинаров с демонстрацией работы в кластере) и практической части с выполнением лабораторных работ на учебном стенде

# План работы

## Инструменты безопасности в Deckhouse Kubernetes Platform

| Тема | Структура |
| --- | --- |
| **1. Основы информационной безопасности в Kubernetes и Deckhouse Kubernetes Platform** | **Цель:** получить знания об основах безопасности в Kubernetes и DKP, а также о концепции Zero Trust в информационной безопасности.<br><br>**Теория:**<br>● Стандарты безопасности и модель 4C<br>● Процесс обеспечения информационной безопасности (ИБ) информационной системы; модели ИБ<br>● Zero Trust для Kubernetes и DKP<br>● Проекты, направленные на безопасность Kubernetes<br>● Нейтрализация рисков при использовании контейнеризации в DKP<br>● Аспекты безопасности кластера DKP на основе модели Zero Trust<br><br>**Практика:** тестирование по теме. |
| **2. Безопасность пода** | **Цель:** научиться создавать и применять политики безопасности для подов, а также управлять Admission controllers для ограничения ресурсов в объектах кластера.<br><br>**Теория:**<br>● Pod Security Standards<br>● Параметры безопасности пода (Security Context)<br>● Концепция Admission controllers<br>● Возможности модуля admission-policy-engine DKP: расширение стандартных политик, операционные и собственные политики<br>● Admission controllers: ResourceQuota, LimitRange, PodNodeSelector<br><br>**Практика:** работа со стандартными политиками безопасности пода, создание собственного ограничения и операционной политики, работа с Admission controllers. |
| **3. Сетевые политики безопасности. Mutual TLS** | **Цель:** научиться создавать и применять сетевые политики для пространств имён и других объектов кластера, создавать и применять взаимную достоверную аутентификацию сервисов, а также шифрование сетевого трафика кластера.<br><br>**Теория:**<br>● CNI, Cilium<br>● Network Policy, CiliumNetworkPolicy<br>● Примеры сетевых политик<br>● mTLS на основе Istio<br>● Политики авторизации в Istio<br><br>**Практика:** создание сетевых политик для различных объектов кластера, шифрование сетевого трафика кластера, создание политик авторизации Istio. |
| **4. Управление доступом** | **Цель:** научиться создавать и применять аутентификаторы для доступа в DKP, а также распределять на них права доступа. |

**Теория:**

- Аутентификация в кластере DKP: сертификаты X.509, ServiceAccount, OIDC-провайдеры, модуль user-authn DKP
- Текущая ролевая модель
- Экспериментальная ролевая модель

**Практика:** создание пользователя и группы пользователей, защита приложения с помощью DexAuthenticator, применение ролей из различных моделей.

---

### 5. Управление сертификатами. Хранение и доставка секретов из Stronghold

**Цель:** научиться создавать и применять системы для управления цифровыми сертификатами и хранения секретов в DKP, а также доставлять различные секреты в приложения.

**Теория:**

- Модуль cert-manager DKP
- Deckhouse Stronghold: функциональные характеристики, сферы применения и работа с секретами
- Доставка секретов в приложения
- Механизм секретов PKI
- Механизм для работы с SSH-ключами

**Практика:** работа с сертификатами в DKP, доставка секретов в приложения с помощью Deckhouse Stronghold, работа с PKI Deckhouse Stronghold.

---

### 6. Выявление уязвимостей и аудит безопасности в кластере

**Цель:** научиться проверять пользовательские образы в runtime на известные уязвимости, искать угрозы безопасности в кластере, а также реализовывать правила аудита.

**Теория:**

- Недоверенные репозитории и подмена образов
- Сканирование образов: модуль operator-trivy DKP
- Проверка соответствия кластера требованиям безопасности
- Аудит Kubernetes API
- Модуль runtime-audit-engine DKP. Falco

**Практика:** выявление уязвимостей в подах, защита от уязвимых образов, настройка политики аудита API-сервера и правил аудита Falco.