

## Введение в Docker

**Код:** PTRN-042

**Длительность:** 8 ч.

### Описание:

Слушатели узнают про подходы и инструменты, которые проверены на реальных проектах.

В рамках курса слушатели:

- Познакомятся с экосистемой контейнеров на основе Docker;
- Разберутся в отличиях контейнеров от виртуальных машин;
- Рассмотрят основные компоненты Docker;
- На практике применят концепцию “инфраструктура как код”;
- Научатся запускать приложения в Docker контейнерах и вести разработку с использованием Docker;
- Научатся взаимодействовать с другими компонентами проекта при разработке в Docker;
- Затронут типовые задачи эксплуатации: работа с сетями, управление конфигурацией.

### Цели:

Познакомить с возможностями Docker для построения Continuous Delivery процесса поставки ПО.

### Разбираемые темы:

1. Введение в Docker.
  - Описание проблем, которые решает система Docker, описание области применимости;
  - Описание области применимости;
  - Виртуализация vs контейнеризация;
  - Сущности Docker: Docker Daemon, Container, Image, Dockerfile, Docker registry;
  - Практика:
  - Подготовка рабочего окружения для работы с Docker;
  - Запуск контейнеров;
  - Сборка image;
  - Работа с Docker Hub.

## 2. Работа с сетью и с данными.

- Сеть в Docker, взаимодействие приложений в Docker контейнерах;
- Хранение персистентных данных в Docker, жизненный цикл stateful приложений. Практика:

- Использование и управление основными видами сетей в Docker;
- Интеграция с существующей сетевой инфраструктурой проекта;
- Работа со stateful приложением на примере PostgreSQL.

## 3. Разработка и запуск проекта в Docker.

- Декларативное описание проекта с помощью Dockerfile и docker compose, развертывание локальных окружений;
- Разработка в Docker окружении, подготовка компонентов проекта для работы в Docker. Практика:

- Сборка image с приложением из Dockerfile;
- Описание сервиса (балансировщик, приложение, база данных) с помощью docker-compose;
- Практика разработки компонентов проекта в Docker.

## 4. Рекомендуемые практики работы с Docker.

- Уменьшения размера образов, ускорение сборки;
- Практики разработки ПО в Docker контейнерах, версионирование, организация логирования в Docker, вопросы безопасности, последовательность запуска контейнеров в проекте.

### **Целевая аудитория:**

Системные администраторы, инфраструктурные инженеры, разработчики, продвинутые тестировщики.

### **Предварительная подготовка - общее:**

Базовое знание ОС Linux.