

DevOps практики и инструменты

DevOps решает задачу взаимодействия инженеров между собой с целью помочь им быстрее создавать и обновлять сервисы и приложения.

Длительность курса: 172 академических часа

1 DevOps. Необходимость меняться

- | | | |
|---|---|---|
| 1 | Что такое DevOps и история его развития. | Как DevOps меняет жизнь и работу компаний. DevOps как профессия. Обзор курса. |
| 2 | DevOps как система. | Практики и методики. DevOps компетенции. |

3 Система контроля версий как основа разработки и поставки ПО. Знакомство с Git.	Домашние задания 1 Работа с локальным и удаленным репозиториями, исправление комитов, решение конфликтов, работа с ветками и практика Code Review.
4 Локальное окружение инженера. ChatOps и визуализация рабочих процессов. Командная работа с Git. Работа в GitHub.	Домашние задания 1 Настройка локального окружения и практика ChatOps.
5 Знакомство с облачной инфраструктурой и облачными сервисами.	Практики безопасного доступа к ресурсам (SSH, Bastion Host, VPN). Домашние задания 1 Запуск VM в GCP, управление правилами фаервола, настройка SSH подключения, настройка SSH подключения через Bastion Host, настройка VPN сервера и VPN-подключения.

6 Основные сервисы Google Cloud Platform (GCP).

Способы управления ресурсами в GCP.

Домашние задания

- 1 Практика управления ресурсами GCP через gcloud.

Ручной деплой тестового приложения.
Написание bash скриптов для автоматизации задач настройки VM и деплоя приложения.

2 Управление инфраструктурой и конфигурацией

1 Модели управления инфраструктурой.

Работа с образами VM в облаке. Знакомство с Packer и экосистемой компании HashiCorp.

Домашние задания

- 1 Подготовка базового образа VM при помощи Packer.

Деплой приложения с использованием подготовленного базового образа.

2 Практика Infrastructure as a Code (IaC).

Знакомство с Terraform.

Домашние задания

- 1 Декларативное описание в виде кода инфраструктуры GCP, требуемой для запуска тестового приложения, при помощи Terraform.
-

3 Принципы организации инфраструктурного кода и работа над инфраструктурой в команде на примере Terraform.

Домашние задания

- 1 Создание Terraform модулей для управления компонентами инфраструктуры.

Описание и настройка инфраструктуры нескольких окружений. Работа с Terraform remote backend.

4 **Управление конфигурацией.**

Основные DevOps инструменты: Chef, Ansible, Puppet, Salt. Знакомство с Ansible.

Домашние задания

- 1 Написание Ansible плейбуков на основе имеющихся bash скриптов.

Сборка базовых образов при помощи Packer и Ansible.

5 **Продолжение знакомства с Ansible: templates, handlers, dynamic inventory, vault, tags.**

Домашние задания

- 1 Управление настройками хостов и деплой приложения при помощи Ansible.
-

6 **Принципы организации кода для управления конфигурацией.**

Ansible роли, управление настройками нескольких окружений и best practices.

Домашние задания

- 1 Написание Ansible ролей для управления конфигурацией сервисов и настройками хостов.

Организация репозитория для работы с несколькими окружениями. Работа с Ansible Galaxy и комьюнити ролями.

7 **Локальная
разработка Ansible
ролей с Vagrant.
Тестирование
конфигурации.**

Домашние задания

- 1 Доработка имеющихся ролей локально с использованием Vagrant.

Тестирование конфигурации при помощи Molecule и TestInfra.

3 Continuous Integration & Continuous Delivery с использованием Docker

1 Технология контейнеризации. Введение в Docker.

Домашние задания

- 1 Установка Docker, запуск контейнера на локальной машине, выполнение команд внутри контейнера, создание образа контейнера на основе запущенного.
-

2 Docker контейнеры. Docker под капотом

Домашние задания

- 1 Запуск VM с установленным Docker Engine при помощи Docker Machine. Написание Dockerfile и сборка образа с тестовым приложением. Сохранение образа на DockerHub.
-

3 Docker образы. Микросервисы

Домашние задания

- 1 Разбиение приложения на несколько микросервисов. Выбор базового образа. Подключение volume к контейнеру.
-

4 Сетевое взаимодействие Docker контейнеров. Docker Compose. Тестирование образов

Домашние задания

- 1 Практика работы с основными типами Docker сетей. Декларативное описание Docker инфраструктуры при помощи Docker Compose.
-

5 **Технология
непрерывной
поставки ПО**

6 **Устройство Gitlab
CI. Построение
процесса
непрерывной
интеграции**

Домашние задания

- 1 Gitlab CI. Построение процесса непрерывной интеграции

4 Fast Feedback Loop (мониторинг и логирование)

1 Введение в мониторинг. Модели и принципы работы систем мониторинга

Домашние задания

- 1 Создание и запуск системы мониторинга Prometheus.

Мониторинг состояния микросервисов, сбор метрик при помощи prometheus exporters.

2 Мониторинг приложения и инфраструктуры

3 Применение инструментов для обработки лог данных

Что такое логи и зачем они нужны? Системы централизованного хранения лог данных, требования к ним и примеры. Форматы и варианты логирования приложений

Домашние задания

- 1 Логирование приложений.

Структурированные и неструктурированные логи. Организация централизованной системы логирования.

4 **Применение системы логирования в инфраструктуре на основе Docker.**

Рассмотрение систем распределенной отладки данных (Distributed tracing)

Домашние задания

1 Логирование приложений.

Структурированные и неструктурированные логи. Продолжение создания и работы централизованной системой логирования.

5 Контейнерная Оркестрация

-
- 1 **Контейнерная оркестрация.** Примеры известных и актуальных систем оркестрации (DC/OS, Aurora, Nomad, Docker Swarm)
-
- 2 **Введение в Kubernetes.** Основные концепции и архитектура системы.
Домашние задания
- 1 Установка и настройка Kubernetes.
- Работа над автоматизацией процесса развертывания системы.
-
- 3 **Основные модели безопасности и контроллеры в Kubernetes** Домашние задания
- 1 Установка и настройка Google Kubernetes Engine, настройка локального профиля администратора для GKE. Работа с с контроллерами: StatefulSet, Deployment, DaemonSet
-
- 4 **Ingress-контроллеры и сервисы в Kubernetes.** Работа с удаленными хранилищами данных.
Домашние задания
- 1 Настройка балансировщиков нагрузки в Kubernetes и SSLTerminating.
- Подключение удаленных хранилищ GCP данных к POD'ам.
-

**5 Интеграция
Kubernetes в
GitlabCI.**

Использование формата упаковки приложений Charts в Kubernetes Helm.

Домашние задания

- 1 Создание Helm Chart'ов для компонент приложения, управление зависимостями Helm.

Построение CI/CD pipeline в Gitlab в интеграции с Kubernetes.

**6 Kubernetes.
Мониторинг и
логирование**

Развертывание Prometheus в Kubernetes. Настройка Prometheus и Grafana для сбора метрик кластера и приложений. Развертывание и настройка EFK для сбора логов.

6 Проектная работа

- 1 **Финальное занятие и выдача курсового проекта**
- Домашние задания
- 1 Проектная работа
-

- 2 **Промежуточная приемка проектов - 10 октября**
-

- 3 **Финальное занятие. Подведение итогов обучения**