

DevOps для эксплуатации и разработки

-
- 01** Кому подойдёт курс
- Системным администраторам или инженерам по эксплуатации ПО с опытом:
- сможете выполнять более сложные задачи
 - получите возможность перейти на новую роль
 - повысите свой грейд
- Разработчикам или QA-инженерам:
- изучите самые актуальные DevOps-инструменты и освоите новые навыки для работы над инфраструктурными задачами
 - сможете браться за более сложные проекты и расти профессионально
-
- 02** Чему научитесь на курсе
- Какие знания и навыки освоите
- Начнёте применять DevOps в работе:
- разберёте ключевые практики и подходы методологии DevOps, которые стали стандартом в современной разработке и эксплуатации ПО, и научитесь использовать их
- Изучите современные инструменты:
- освоите актуальные подходы и инструменты для тестирования и доставки приложений, описания инфраструктуры и управления ей
- Освоите Docker и Kubernetes:
- будете создавать и масштабировать приложения с помощью контейнерной виртуализации и оркестрации
- Будете сами проектировать пайплайны:
- от исходного кода до отказоустойчивого и нагруженного онлайн-маркета, как инженеры в командах разработки IT-продуктов
-
- 03** Как проходит обучение
- теория и практика на платформе Практикума
 - практические задания на готовой инфраструктуре в Яндекс.Облаке
 - воркшопы
 - индивидуальные консультации с наставником
 - обучение из любой точки мира в удобное время
 - траблшутинг в расширенном тарифе
-

Что вас ждёт на обучении

Начнёте применять в работе принципы и инструменты DevOps

Научитесь решать проблемы на реальной инфраструктуре

Сделаете и добавите в портфолио большой самостоятельный проект

DevOps для эксплуатации и разработки

Сравнение тарифов

	Интенсивный тариф	Стандартный тариф	Расширенный тариф
Длительность	5 месяцев	7 месяцев	9 месяцев
Нагрузка в неделю	~15 часов	~12 часов	~10 часов
Минимальные знания на старте	<ul style="list-style-type: none"> Опыт работы в IT не менее трёх лет Умение писать небольшие скрипты на любом языке Опыт работы с UNIX подобными системами Знание языка SQL на уровне написания простых запросов 	<ul style="list-style-type: none"> Опыт работы в IT не менее трёх лет Умение писать небольшие скрипты на любом языке Опыт работы с UNIX подобными системами Знание языка SQL на уровне написания простых запросов 	<ul style="list-style-type: none"> Опыт работы в IT не менее трёх лет Умение писать небольшие скрипты на любом языке
Помощь с трудоустройством	Карьерный курс	Карьерный курс	Карьерный курс
Дополнительные темы	✗	✗	<ul style="list-style-type: none"> Основы Linux Базовое программирование на Bash Курс по Soft skills
Дополнительная практика	Практические задания в «песочнице» с самопроверкой	Практические задания в «песочнице» с самопроверкой	<ul style="list-style-type: none"> Практические задания в «песочнице» с самопроверкой Регулярный траблшутинг
Сопровождение: наставник, куратор и поддержка учебной инфраструктуры	✓	✓	✓
Готовая инфраструктура в Yandex Cloud, симулирующая реальные рабочие ситуации	✓	✓	✓
Вебинары с экспертами	✓	✓	✓
Дипломный проект	✓	✓	✓

DevOps для эксплуатации и разработки

5, 7 или 9 месяцев

продолжительность курса

1 или 2 НЕДЕЛИ

01

Системы контроля версий и автоматизация сборки приложений

2 НЕДЕЛИ

02

Гибкие методологии и Continuous Integration

3 НЕДЕЛИ

03

Основы Linux

Только в расширенном тарифе

2 НЕДЕЛИ

04

Создание скриптов на Bash

Только в расширенном тарифе

2 НЕДЕЛИ

05

Сети и основы работы на серверах Linux

1 или 2 НЕДЕЛИ

06

Continuous Delivery и Continuous Deployment

2 или 3 НЕДЕЛИ

07

Infrastructure as Code и системы управления конфигурацией

2 НЕДЕЛИ

08

DBOps: реляционные и нереляционные базы данных

2 или 3 НЕДЕЛИ

09

Docker-контейнеризация и хранение данных

1 или 2 НЕДЕЛИ

10

Микросервисы, балансировка и кеширование

4 НЕДЕЛИ

11

Kubernetes. Деплой и обеспечение надёжности приложения

2 НЕДЕЛИ

12

Логирование и мониторинг ошибок

2 или 3 НЕДЕЛИ

13

Дипломный проект



Факультативный курс. Эффективное взаимодействие с командой и управление ей

Только в стандартном и расширенном тарифе

Системы контроля версий и автоматизация сборки приложений

01

1 неделя (интенсивный тариф)
2 недели (стандартный и расширенный тарифы)
Практическая работа

Вы попадаете на работу в стартап, знакомитесь с командой и удивляетесь, как вообще можно так работать. Желая скорее всё поправить, вы разбираетесь в том, как устроена разработка программного обеспечения в наши дни, и делаете первые шаги на пути ускорения производства.

Узнаете основы концепции бережливого производства и поймёте, как они перекладываются на IT-процессы.

Научитесь автоматизировать повторяющиеся задачи при помощи Jenkins.

Практика

- Организация работы с ветками в Git
- Установка и настройка Jenkins, сборка проекта с помощью Maven

Содержание

01. Как устроен жизненный цикл ПО

02. Системы контроля версий. Почему все выбирают Git?

- Организация командной работы с помощью Feature Branch Workflow
- Организация хранения Git-репозитория в Gitea

03. Бережливое производство: основы и принципы

04. Автоматизация сборки

- Инструменты автоматизации повторяющихся задач
- Фреймворки для автоматизации сборки проекта

2 недели

Практическая работа

Ваш стартап покупает IT-гигант — *Kрупnaya Kompaniya (KK)*. Тимлид вашей команды становится вашим проводником на пути познания DevOps. Но проблемы с приложением никуда не уходят.

Вы узнаете, что приложение работает плохо на продуктивной среде, а в разработку постоянно приходят новые заявки на исправление дефектов, из-за чего копится работа и о выпуске нового функционала и стабильности даже не приходится мечтать.

Сможете организовать процесс CI в работе команды. Сумеете представить процесс сборки и поставки ПО в виде пайплайна, используя Gitlab CI.

Встроите в процесс поставки ПО этап анализа кода на безопасность и качество, используя анализаторы кода.

Практика

- Изучение структуры пайплайна в GitLab CI
- Настройка SonarQube, добавление в пайплайн этапов анализа качества (SonarQube) и безопасности (Gitlab SAST) кода
- Написание Jenkins Declarative Pipeline

Содержание

01. Проблематика DevOps

- Функциональные колодцы
- Исходящая спираль

02. Гибкие методологии и DevOps культура

- Взаимодействие в команде
- Современные подходы. Чем обусловлено появление Agile?
- Три пути DevOps: цикл поставки ценности, петли обратной связи, эксперименты и обучение

03. Continuous Integration

- Организация хранения кода
- Организация сборки
- Организация тестирования
- Быстрое получение обратной связи

04. Сервера Continuous Integration

- Обзор интерфейса и функционала Gitlab CI

05. Измерение качества и статическое тестирование безопасности кода

- Инструменты статического анализа качества кода: SonarQube, Gitlab Analyzer
- Инструменты статического анализа безопасности кода: SonarQube SAST, Gitlab SAST

3 недели

Практическая работа

Техлид КК выбирает вас, чтобы провести инструктаж по Linux для новых сотрудников. Но оказывается, что ваших знаний недостаточно и нужно снова вливаться в ряды стажёров. Шаг за шагом вы научитесь обращаться с Linux, как с родной операционной системой.

Освойте базу Linux, без которой невозможно представить работу DevOps.

Практика

- Работа с командной строкой
- Взаимодействие с файловой системой и разделами в Linux
- Управление ресурсами ОС и распределение их по задачам
- Траблшутинг: устранение неисправностей в «песочнице»

Содержание

01. Архитектура (устройство Linux)

02. Терминал и основные команды

03. Дисковая подсистема
и ввод-вывод

04. Процессы и память

05. Сетевая подсистема

06. Графическая система

07. Администрирование: демоны,
пользователи, загрузка
операционной системы
и введение в Bash

08. Траблшутинг

2 недели

Практическая работа

Вы переходите на новый уровень изучения UNIX-подобных систем и погружаетесь в один из самых популярных языков для написания скриптов — Bash. И вы должны быть готовы встретиться с ним на практике.

Научитесь уверенно программировать на языке Bash и сможете писать скрипты на уровне, который необходим для прохождения курса.

Практика

- Создание полезных скриптов для закрепления синтаксиса Bash
- Взаимодействие с пользователями
- Выполнение задач на удалённых машинах

Содержание

01. Автоматизация рутинных операций с помощью Bash

02. Работа с потоками ввода-вывода

03. Использование Bash за пределами Linux

04. Синтаксис языка Bash

05. Основные функции языка Bash

2 недели

Практическая работа

Вы знакомитесь с системным администратором КК, который рассказывает вам, как устроена сеть и как у него тут вообще всё работает. В процессе демонстрации вы видите, что админ гораздо быстрее управляется с командами и перемещениями курсора в консоли, и просите рассказать про эти приёмы. Админ предлагает вам разобраться на примере задачи по переносу сервиса со старого сервера на новый и заодно решить горячий вопрос.

Научитесь уверенно работать с сетями и серверами на Linux.

Практика

- Работа с пользователями, сессиями и процессами
- Назначение и управление правами пользователей
- Трешшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01. Основы работы на серверах Linux, работа с командной строкой

02. Дисковая подсистема Linux

03. Права пользователей Linux

04. Основы сетей: проблемы с подключением к удалённому серверу

05. Виртуализация

- Популярные гипервизоры
- Виртуальные машины
- Виртуальные сети

1 неделя (интенсивный тариф)
2 недели (стандартный и расширенный тарифы)
Практическая работа

Получив очередное за неделю обновление приложения на смартфоне, вы заинтересовались, как часто обновления приходят пользователям. Оказалось, совсем не часто. А нужно ли чаще?

Сможете организовать часть деплоя в пайплайне и автоматизировать её, используя инструменты Gitlab CI. Сможете проводить оценку потребности в частых обновлениях ПО с точки зрения бизнеса.

Практика

- Публикация артефактов сборки проекта в хранилище Nexus
- Проектирование процесса поставки
- Добавление в пайплайн части с установкой на тестовую среду
- Откат на предыдущую версию приложения с помощью GitLab

Содержание

01. Как устроен процесс поставки ИТ-продукта

- Что такое процесс поставки
- Value Stream management

02. Системы хранения артефактов: Nexus

03. Continuous Delivery

- Визуализация этапов поставки
- Организация процесса поставки. Delivery Pipeline. Continuous Deployment

04. Методология Twelve-Factor App

05. Бэкапирование

- Организация резервного копирования
- Виды бэкапов
- Ротации бэкапов

2 недели (интенсивный и стандартный тарифы)
3 недели (расширенный тариф)
Практическая работа

Вы впервые сталкиваетесь с проблемой в инфраструктуре. Кто-то изменил конфигурацию серверов разработчиков, на которых они проводили эксперименты, и теперь сервера работают настолько неправильно, что больше невозможно их использовать.

Научитесь развёртывать и управлять инфраструктурой приложения согласно подходу IaC, используя Terraform.

Практика

- Развёртывание инфраструктуры через Terraform
- Трешшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01. Infrastructure as Code: описываем инфраструктуру кодом, перенимаем опыт разработки

02. Системы управления конфигурацией

- Ansible (roles, tasks, inventory, vault, awx)
- Знакомство с другими системами - Chef, Salt, Puppet

2 недели

Практическая работа

Вы встречаетесь с DBA. К вам приходят грозные комментарии от пользователей, что сайт тормозит. Эмпирическим путём вы обнаруживаете, что последнее обновление сильно замедлило работу базы данных.

Научитесь применять практики DevOps к администрированию баз данных.

Практика

- Поднятие PostgreSQL и перенос данных в новую БД
- Переподключение приложения на вновь поднятую MongoDB
- Настройка автоматической миграции
- Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01. Основы теории баз данных

- Реляционные базы данных: PostgreSQL
- Основные SQL-запросы
- Как ускорить выполнение запросов в БД: оптимизация, кеширование, рост мощностей, индексация

02. Организация высокой доступности БД

- Принципы DBOps
- Миграция БД с помощью Flyway

03. Нереляционные базы данных: MongoDB

2 недели (интенсивный
и стандартный тарифы)

3 недели (расширенный
тариф)

Практическая работа

Вы окажетесь меж двух огней: с одной стороны — разработчики, у которых локально всё работает, а у кого-то из них, возможно, и нет; с другой — отдел эксплуатации, у которого вообще ничего не работает. Разбираемся, как сделать так, чтобы у всех было всё одинаково хорошее и наконец-то запускаем приложение в проде.

Научитесь разворачивать приложение, используя Docker, и поддерживать его работоспособность.

Практика

- Написание Docker-файла, сборка образа, публикация в Nexus, использование в пайплайне и развёртывание приложения на стенде с помощью Docker
- Установка MinIO, загрузка данных в хранилище
- Установка Vault, публикация секретов в Vault
- Трешшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01. Контейнеризация

- Обзор Linux- и Docker-контейнеров
- Устройство Docker: слои, образы, контейнеры, Docker-файл, Registry
- Альтернативы Docker (containerd, podman, LXC, cri-o)

02. Хранение данных: работа с S3, MinIO vs Ceph

03. Хранение и передача чувствительных данных: Содержание уроков главы: большой обзор Vault

1 неделя (интенсивный тариф)
2 недели (стандартный и расширенный тарифы)
Практическая работа

Вы все готовитесь к выходу новой версии приложения. Вы хотите учесть ошибки прошлого, когда ваш сервис не выдержал трафика и лежал три дня. На этот раз к вам пришёл бизнес и потребовал, чтобы подобного не случилось! Вы настолько погрузитесь в процесс, что привнесёте новых крутых фиш и стратегий поставки.

Научитесь организовывать высоконагруженную систему, используя инструменты балансировки и кеширования.

Практика

- Поднять Nginx и несколько экземпляров сервиса
- Развёртывание локального кластера k8s и managed k8s кластера в Облаке
- Трешшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01. Балансировка и кеширование

02. Микросервисы

- Подходы к организации архитектуры приложения
 - Инструменты для управления микросервисными приложениями: Docker-compose
-

03. Стратегии поставки и как выбрать подходящую

- Какие пользователи бывают
 - Сине-зелёный деплой
 - Канареечный деплой
-

4 недели

Практическая работа

Вы создаёте сервисную платформу для обеспечения качественной работы всего цикла поставки.

Научитесь использовать k8s для деплоя и обеспечения надёжности приложения. Познакомитесь с практиками GitOps, научитесь организовывать поставку приложений с помощью инструментов Flux, ArgoCD.

Практика

- Работа с Minikube
- Подготовка манифестов Kubernetes для деплоя в кластер Kubernetes и push в репозиторий GitLab
- Модификация файла .gitlab-ci.yml и изменение шагов деплоя приложения
- Деплой и передеплой через ArgoCD
- Доведение YAML-файла до нового релиза по ТЗ
- Написание Helm чарта для деплоя приложения
- Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

- | | |
|--|--|
| 01. Оркестрация контейнеров | <ul style="list-style-type: none">• Запуск приложения в современной инфраструктуре• Для чего нужны системы оркестрации и какие они бывают• Kubernetes: причины и цели разработки |
| 02. Запуск Kubernetes кластера для тестирования и разработки | <ul style="list-style-type: none">• Как получить Kubernetes: описание 5 основных способов установки• Разбор простых команд kubectl |
| 03. Архитектура Kubernetes | <ul style="list-style-type: none">• Основные компоненты и их взаимодействие: scheduler, kubelet, kube-proxy, controller manager, etcd и api-сервер |
| 04. Основные сущности Kubernetes | <ul style="list-style-type: none">• Nodes, Podes, Services, Persistent Volumes, Persistent Volume Claim• Манифесты Kubernetes• Использование kubectl на уровне подключения к кластеру• Просмотр конфигурации Node, Pod'ов, Secrets и т.д. |
| 05. Продвинутое сущности Kubernetes | <ul style="list-style-type: none">• ReplicaSet, Deployment, DaemonSet и StatefulSet• Liveness, Readiness probe + Init контейнеры• Дополнение про kubectl: edit, apply, delete, create• Сетевое взаимодействие в кластере |
| 06. Продвинутое сущности Kubernetes с утилитой kubectl | <ul style="list-style-type: none">• Дополнительные команды kubectl, флаги, представление сущностей Kubernetes в виде YAML или JSON/YAML файлов• Деплой приложения двумя способами: с помощью команды и через YAML-файл |

07. Деплой приложений в Kubernetes

- Деплой фронтенда приложения
- Деплой и передеплой через ArgoCD

08. Стратегии деплоя

- Связь Kubernetes и Twelve-Factor App
- Стратегии деплоя в Kubernetes
- Как в Kubernetes реализован механизм rollback

09. Практики GitOps для работы с инфраструктурой

10. Написание Helm чарта для деплоя приложения

11. Настройка ArgoCD

12. Использование шаблонизаторов

Логирование и мониторинг ошибок

12

2 недели

Практическая работа

Ваша сервисная платформа внезапно перестаёт работать.

Научитесь настраивать систему сбора логов с помощью инструментов Loki, Sentry. Научитесь настраивать систему мониторинга, используя GAP стек.

Практика

- Создание системы логирования Loki для вашего приложения. Построение графика работоспособности приложения
- Поставка и настройка GAP стека для приложения, настройка алертов
- Трешшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01. Логирование и мониторинг ошибок

- Логи Linux
- Loki
- Sentry

02. Мониторинг

- Типы метрик, типовые аномалии
- Алерты
- GAP стек (Grafana, Prometheus, Alertmanager)
- Golden Signals
- Метрики уровня приложения

03. АРМ системы и
распределённый трейсинг

04. C.A.L.M.S

Дипломный проект

13

2 недели (интенсивный
тариф)

3 недели (стандартный
и расширенный тарифы)

Практическая работа

В финальном проекте вам предстоит подготовить инфраструктуру и настроить конвейер деплоя: системы автоматической сборки, автоматического тестирования и проверки кода на ошибки и уязвимости. Вы выстроите логику развёртывания приложения для разных окружений, настроите системы для мониторинга и логирования приложения.

Факультативный курс. Эффективное взаимодействие с командой и управление ей

Только в стандартном и расширенном тарифах

Содержание

01. Приоритеты и делегирование

02. Настройка коммуникации
в команде

03. Настройка процессов
и эффективность команды

04. Управление коммуникацией
в IT-командах

05. Взаимодействие с командой
разработки
