

# Карта курса DevOps для эксплуатации и разработки

## Глава 1

15 часов  
2 недели

### Знакомство с компанией

Вы попадаете на работу в стартап, знакомитесь с командой и удивляетесь, как вообще возможно 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.

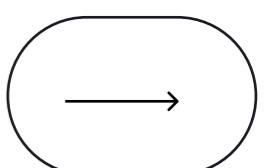
#### Содержание уроков главы:

1. Как устроен жизненный цикл ПО
2. Системы контроля версий.  
Почему все выбирают Git?
  - \* Организация командной работы с помощью Feature Branch Workflow.
  - \* Организация хранения Git-репозитория в Gitea.
3. Бережливое производство: основы и принципы
4. Автоматизация сборки
  - \* Инструменты автоматизации повторяющихся задач.
  - \* Фреймворки для автоматизации сборки проекта.

#### Практика:

Установка Git-сервера Gitea, организация работы с ветками в Git.

Установка и настройка Jenkins, сборка проекта с помощью Maven.



## Достаточно хороший код

Ваш стартап покупает IT-гигант – Крупная Компания (КК). 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.

### Содержание уроков главы:

#### 1. Проблематика DevOps

- \* Функциональные колодцы.
- \* Нисходящая спираль.

#### 2. 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.

#### 3. Continuous Integration

- \* Организация хранения кода.
- \* Организация сборки.
- \* Организация тестирования.
- \* Быстрое получение обратной связи.

#### 4. Сервера Continuous Integration

- \* Обзор интерфейса и функционала GitLab CI.

#### 5. Измерение качества и статическое тестирование безопасности кода

- \* Инструменты статического анализа качества кода:  
SonarQube, GitLab Analyzer.
- \* Инструменты статического анализа безопасности кода:  
SonarQube SAST, GitLab SAST.

### 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.



## Глава 3

15 часов  
2 недели

# Срочные дела

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.

### Содержание уроков главы:

1. Основы работы на серверах Linux, работа с командной строкой
2. Дисковая подсистема Linux
3. Права пользователей Linux.
4. Основы сетей: проблемы с подключением к удалённому серверу.
5. Виртуализация
  - \* Популярные гипервизоры.
  - \* Виртуальные машины.
  - \* Виртуальные сети.

### Практика:

Работа с пользователями, сессиями и процессами.

Назначение и управление правами пользователей.

Изучение трафика и просмотр его содержимого.

## Глава 4

20 часов  
2 недели

# Доступно новое обновление

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.

### Содержание уроков главы:

1. Как устроен процесс поставки IT-продукта
  - \* Что такое процесс поставки.
  - \* Value Stream management.
2. Системы хранения артефактов: Nexus
3. Continuous Delivery
  - \* Визуализация этапов поставки.
  - \* Организация процесса поставки. Delivery Pipeline. Continuous Deployment.
4. Методология Twelve-Factor App

### 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.



## Глава 5

20 часов  
2 недели

# Проблемы в инфраструктуре

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.

### Содержание уроков главы:

#### 1. Бэкапирование

- \* Организация резервного копирования.
- \* Виды бэкапов.
- \* Ротации бэкапов.

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

#### 3. Системы управления конфигурацией

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

### Практика:

Развёртывание инфраструктуры через Terraform.

## Глава 6

15 часов  
2 недели

# Незамеченная оплошность

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.

### Содержание уроков главы:

#### 1. Основы теории баз данных

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

#### 2. Организация высокой доступности БД

- \* Принципы DBOps.
- \* Миграция БД с помощью Flyway.

#### 3. Нереляционные базы данных: mongoDB, redis и Cassandra

### Практика:

Поднятие PostgreSQL и перенос данных в новую БД.

Переподключение приложения на вновь поднятую mongoDB.

Настройка автоматической миграции.



## Изоляция окружения

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 хорошо и наконец-то запускаем приложение в проде.

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

### Содержание уроков главы:

#### 1. Контейнеризация

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

#### 2. Хранение данных: работа с S3, MinIO vs Ceph

#### 3. Хранение и передача чувствительных данных: большой обзор Vault

### 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.



## Наплыв пользователей

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 фич и стратегий поставки.

Научитесь организовывать высоконагруженную систему, используя инструменты балансировки и кэширования (Nginx, Traefik, HAProxy) и очередей (RabbitMQ, Celery).

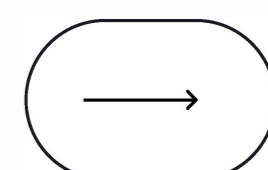
### Содержание уроков главы:

1. Балансировка и кэширование: Nginx, Traefik, HAProxy, Varnish, Redis
2. Работа с очередями сообщений: Kafka, RabbitMQ
3. Микросервисы
  - \* Подходы к организации архитектуры приложения.
  - \* Инструменты для управления микросервисными приложениями: Docker-compose.
4. Стратегии поставки и как выбрать подходящую
  - \* Какие пользователи бывают.
  - \* Сине-зелёный деплой.
  - \* Канареечный деплой.

### Практика:

Поднять Nginx и несколько инстансов сервиса.

Развёртывание микросервисного приложения с помощью Docker-compose.



# Достаточно хорошая инфраструктура

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

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

## Содержание уроков главы:

### 1. Инфраструктура сервисной платформы

- \* Уровни инфраструктуры.
- \* Поставка инфраструктуры, как бизнес ценности.

### 2. Kubernetes

- \* История Kubernetes.
- \* Основные сущности Kubernetes (Nodes, Podes, Deployments, Services, StatefulSet, DeamonSet, Custom Resource Definitions).
- \* Работа с утилитой kubectl.
- \* Инструменты для удобной работы с кластером.
- \* Как развернуть локальный кластер Kubernetes.
- \* Как развернуть боевой кластер Kubernetes.
- \* Managed Kubernetes.

### 3. Организация поставки приложений с Kubernetes

### 4. Шаблонизация: jsonnet, Helm

### 5. GitOps – непрерывное развёртывание облачных приложений: Flux, AgroCD

### 6. Инфраструктура в облаке

- \* Обзор возможностей облачных сервисов.
- \* Сравнение лидеров рынка: AWS, GCP, Azure.

## Практика:

Проектирование инфраструктуры сервисной платформы в Miro.

Развёртывание локального кластера k8s и кластер k8s на виртуальных машинах; развёртывание managed k8s в Облаке.

Организация поставки приложения в k8s с помощью Helm-чарта.

Организация поставки приложения в k8s с помощью GitOps.



# Авария

Ваша сервисная платформа внезапно перестаёт работать.

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

15 часов  
2 недели

## Содержание уроков главы:

### 1. Логирование и мониторинг ошибок

- \* Логи Linux.
- \* ELK stack.
- \* Loki.
- \* Victoria Metrics.
- \* Sentry.

### 2. Мониторинг

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

### 3. APM системы и распределённый трейсинг

### 4. C.A.L.M.S

## Практика:

Создание системы логирования ELK для вашего приложения. Построение графика работоспособности приложения.

Поставка и настройка GAP стека для приложения, настройка алертов.