Практикум

Продолжительность курса – 6 месяцев

Карта курса DevOps для эксплуатации и разработки

Глава 1

15 часов
2 недели

Системы контроля версий и автоматизация сборки приложений

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.

Что изучите:

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-репозиториев в Gitea.
3. Бережливое производство: основы и принципы
4. Автоматизация сборки
   * Инструменты автоматизации повторяющихся задач.
   * Фреймворки для автоматизации сборки проекта.

Практика:

Организация работы с ветками в Git.

Установка и настройка Jenkins, сборка проекта с помощью Maven.

Глава 2

15 часов
2 недели

Гибкие методологии и Continuous Integration

Ваш стартап покупает IT-гигант — Krupnaya 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.

Что изучите:

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

Сети и основы работы на серверах Linux

Вы знакомитесь с системным администратором KK, 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.

Что изучите:

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.

Глава 4

20 часов
2 недели

Continuous Delivery и Continuous Deployment

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.

Что изучите:

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.

Проективание 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

Infrastructure as Code и системы управления конфигурацией

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.

Что изучите:

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

DBOps: реляционные и нереляционные базы данных

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.

Что изучите:

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

Практика:

Поднятие PostgreSQL и перенос данных в новую БД.

Переподключение приложения на вновь поднятую mongoDB.

Настройка автоматической миграции.

→

Глава 7

20 часов
2 недели

Docker-контейнеризация и хранение данных

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,е и наконец-то запускаем приложение в проде.

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

Что изучите:

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.

Глава 8

15 часов
2 недели

Микросервисы, балансировка и кэширование

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.

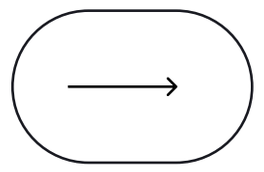
Что изучите:

1. Балансировка и кэширование
2. Микросервисы
   * Подходы к организации архитектуры приложения.
   * Инструменты для управления микросервисными приложениями: Docker-compose.
3. Стратегии поставки и как выбрать подходящую
   * Какие пользователи бывают.
   * Сине-зелёный деплой.
   * Канареечный деплой.

Практика:

Поднять Nginx и несколько инстансов сервиса.

Развёртывание локального кластера k8s и managed k8s кластера в Облаке.



Глава 9

50 часов
5 недель

Kubernetes. Деплой и обеспечение надёжности приложения

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

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

Что изучите:

1. Оркестрация контейнеров:
   * Запуск приложения в современной инфраструктуре.
   * Для чего нужны системы оркестрации, и какие они бывают?
   * Kubernetes: причины и цели разработки.
2. Запуск Kubernetes кластера для тестирования и разработки:
   * Как получить Kubernetes: описание 5 основных способов установки.
   * Разбор простых команд kubectl.
3. Архитектура Kubernetes:
   * Основные компоненты и их взаимодействие: scheduler, kubelet, kube-proxy, controller manager, etcd и api-сервер.
4. Основные сущности Kubernetes:
   * 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.
5. Продвинутые сущности Kubernetes:
   * ReplicaSet, Deployment, DaemonSet и StatefulSet.
   * Liveness, Readiness probe + Init контейнеры.
   * Дополнение про kubectl: edit, apply, delete, create.
   * Сетевое взаимодействие в кластере.
6. Продвинутая работа с утилитой kubectl:
   * 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.

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.

→

7. Деплой приложений в Kubernetes
    * Деплой фронтенда приложения.
    * Деплой и передеплой через ArgoCD.
8. Стратегии деплоя:
    * Связь Kubernetes и Twelve-Factor App.
    * Стратегии деплоя в Kubernetes.
    * Как в Kubernetes реализован механизм rollback.
9. Практики GitOps для работы с инфраструктурой
10. Написание Helm чарта для деплоя приложения
11. Настройка ArgoCD
12. Использование шаблонизаторов

Глава 10

15 часов
2 недели

Логирование и мониторинг ошибок

Ваша сервисная платформа внезапно перестаёт работать.

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

Что изучите:

1. Логирование и мониторинг ошибок
    * Логи Linux.
    * Loki.
    * Sentry.
2. Мониторинг
    * Типы метрик, типовые аномалии.
    * Алерты.
    * 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 Loki для вашего приложения. Построение графика работоспособности приложения.

Поставка и настройка GAP стека для приложения, настройка алертов.

30 часов
3 недели

Финальный проект

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.