

CI/CD на AWS, Azure и Gitlab

Курс как для разработчиков, так и для администраторов, отвечающим за настройку процессов непрерывной разработки и доставки (continuous integration / continuous delivery)

Длительность курса: 48 академических часов

1 Разработка в облаке (Code)

1 СУБД, поддерживающие SQL

вы сможете:

1. перечислять основные различия СУБД, поддерживающих SQL (например – MS SQL, СУБД Oracle, PostgreSQL)
2. писать SQL запросы, используя операторы CREATE, DROP, ALTER и др.

Это базовые операторы, которые используются во всех SQL-запросах. Без этих знаний и умений вы не сможете написать SQL-запрос.

- 2 **Введение в архитектуру, знакомство с демо-проектом**
- рассмотрим архитектуру облачных решений Microsoft Azure DevOps, сервисов AWS, используемых в DevOps процессах.
- Демонстрация реализации демо проекта в двух вариантах - Azure DevOps и AWS.
- Обратим внимание на достоинства и недостатки каждого решения.
интеграция с Gitlab и Jenkins.
-
- 3 **Знакомство моделями Workflow (git flow, trunk-based и другие)**
- анализируем различные виды моделей ветвления в git-e:
- Centralized Workflow
 - Git flow
 - Trunc-based
 - Forking Workflow
- Обсудим какую модель в каком случае имеет смысл применить.
-
- 4 **Настройка учетных записей**
- изучаем работу с ролевыми моделями - сравниваем какие возможности и ограничения предоставляют нам облачные решения.
- Регистрируем группы пользователей, самих пользователей, раздаем права на ресурсы.
-
- 5 **Настройка репозитория кода, миграция с других решений**
- сравниваем возможности AWS CodeCommit, Azure DevOps Repos и Gitlab (в части репозития кода).
- Настраиваем репозитории для демо-проекта.
- Проводим миграцию кода демо-проекта с GitHub-a.
-

6 Знакомство с IDE в облаке

попробуем работать с кодом при помощи исключительно браузера.

Настраиваем, вносим правки в код и выполняем коммит в репозиторий через AWS Cloud9, Microsoft Visual Studio Online (если будет доступен), Gitlab Web IDE.

7 Работа с агентами сборки

- Агенты сборки в travis
 - GitlabCI runners
 - Jenkins slaves
-

8 Хранилища артефактов

сравним что и где мы можем хранить, используя сервисы:

- Azure Artifacts
- Azure Container Registry
- Amazon S3
- AWS EC2 Container Registry
- GitLab Package Registry

Создадим необходимые нашему демо-проекту репозитории.

Сохраним артефакты демо-проекта в созданных репозиториях.

2 Автоматизация сборки и тестирования (Continuous Integration)

1 **Первый pipeline (вручную и не только)**

2 **Запуск модульных тестов**

3 **Статическая проверка качества кода**

- Обзор линтеров и статических анализаторов
 - Рассмотрение основных линтеров для языков:
 - yaml
 - python
 - java
 - npm
 - Best practices
-

4 **Проверка кода на кибербезопасность**

изучаем разницу между статической проверкой кода (SAST) на уязвимости и динамической проверкой (DAST).

Анализируем функциональные возможности open-соурсных и коммерческих утилит.

Подключаем к нашему пайплайну обе проверки.

5 **Интеграция Jenkins с облачными провайдерами, k8s, интеграция с gitlab и github**

- Обзор базы по докер(Команды, Dockerfile)
 - MultyStage build
 - Docker в CICD
 - Docker-in-docker
 - Docker without docker
 - Kaniko
 - Buildah
 - Использование skopeo для копирования образов
-
- Обзор CVE
 - Docker bench security
 - Clair
-

6 **Проверка на лицензионную чистоту**

рассмотрим виды лицензирования - их отличия и возможности использования.

Обсудим в чем смысл проверок на соответствие продукта используемым лицензиям и к чему может привести нарушение ""лицензионной чистоты"" продукта.

Добавляем в пайплайн демо-приложения шаг проверки на ""чистоту лицензирования"".

7 **Настройка окружения для Code Review**

обсуждаем различные подходы к процессу ревью изменений до их принятия в основую ветку разработки.

Настраиваем деплой изменений на статическое и динамическое окружение.

Добавляем шаг Review в пайплайн демо-проекта

3 Автоматизация установки (Continuous Delivery)

1 Работа с секретами

как деплоить на прод, сохраняя в секрете логины/пароли, ключи и другие sensitive data.

Исследуем возможности облачных провайдеров и внешние инструменты.

Прописываем для пайплайна демо-приложения данные для доступа на промышленные среды.

2 Стратегии развертывания (Fail fast, Canary deployment, Feature flags)

3 Установка в базу данных

- flyaway
 - liquidbase
 - Best Practices
-

4 Инсталляция web-приложения

устанавливаем web-приложение используя:

- AWS Elastic Beanstalk
- Amazon Lightsail
- Службы приложений Azure
- пайплайны Gitlab и Jenkins

Добавляем шаг деплоя фронтенд-части демо-проекта

5 Установка приложения в докер-образе

устанавливаем докер-образы используя:

- Amazon Elastic Container Service
- Mesosphere DC/OS Open Source
- пайплайны Gitlab и Jenkins

Добавляем шаг деплоя бэкенд-части демо-проекта

6 Установка в Kubernetes

- Обзор инструментов (Без упоминания примитивов k8s)
 - Deployments
 - Spinnaker
 - Helm
 - Примеры CD, с helm и istio
-

7 Мониторинг работы приложения

настраиваем мониторинг приложения средствами облачных провайдеров:

- Amazon CloudWatch
- Azure Monitor
- Мониторинг в Gitlab

Подключаем мониторинг к демо-проекту, анализируем доступность приложения при установке новой версии.

4 ИТОГОВЫЙ МОДУЛЬ

1 **консультация
по проекту**

2 **Итоговое
занятие**

получение обратной связи по курсу.

Обсуждение проектных работ.