

Final Project: CI/CD pipeline

EPAM DevOps external course
Oleg Romanov

Personal Facts



I am a **computer enthusiast**. My passion for computers started with assembler coding for Sinclair ZX-Spectrum.

There is a whole dozen of binaries and protocols in my personal list of **reverse-engineered** stuff.

Two servers in my **homelab** store my data with high reliability and availability. They also provide services to facilitate media consumption for my family.

EPAM DevOps course helped me to discover **clouds** and now I am actively **exploring** its possibilities.

I smoke, but I want to quit)

Project Description

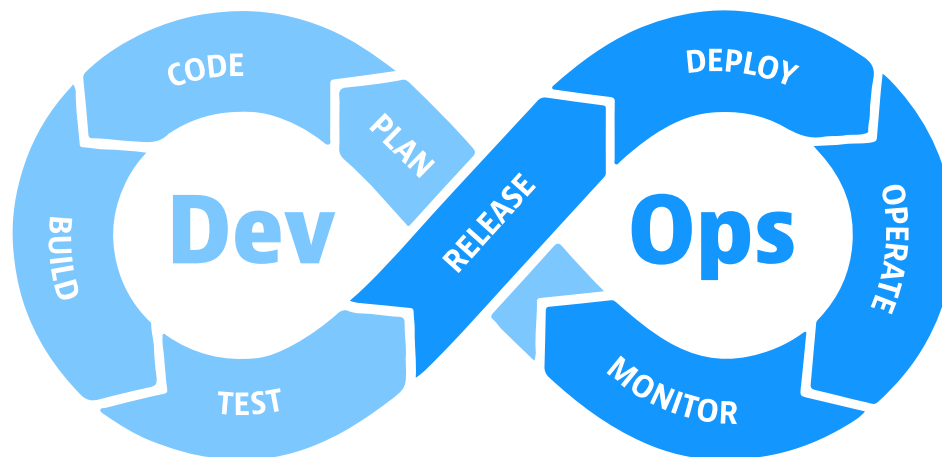
Theme

Zero-Downtime Deployment of a Web Application

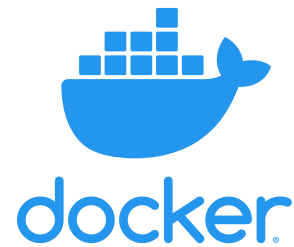
Project Goals

Get familiar with DevOps tools and technologies in order to:

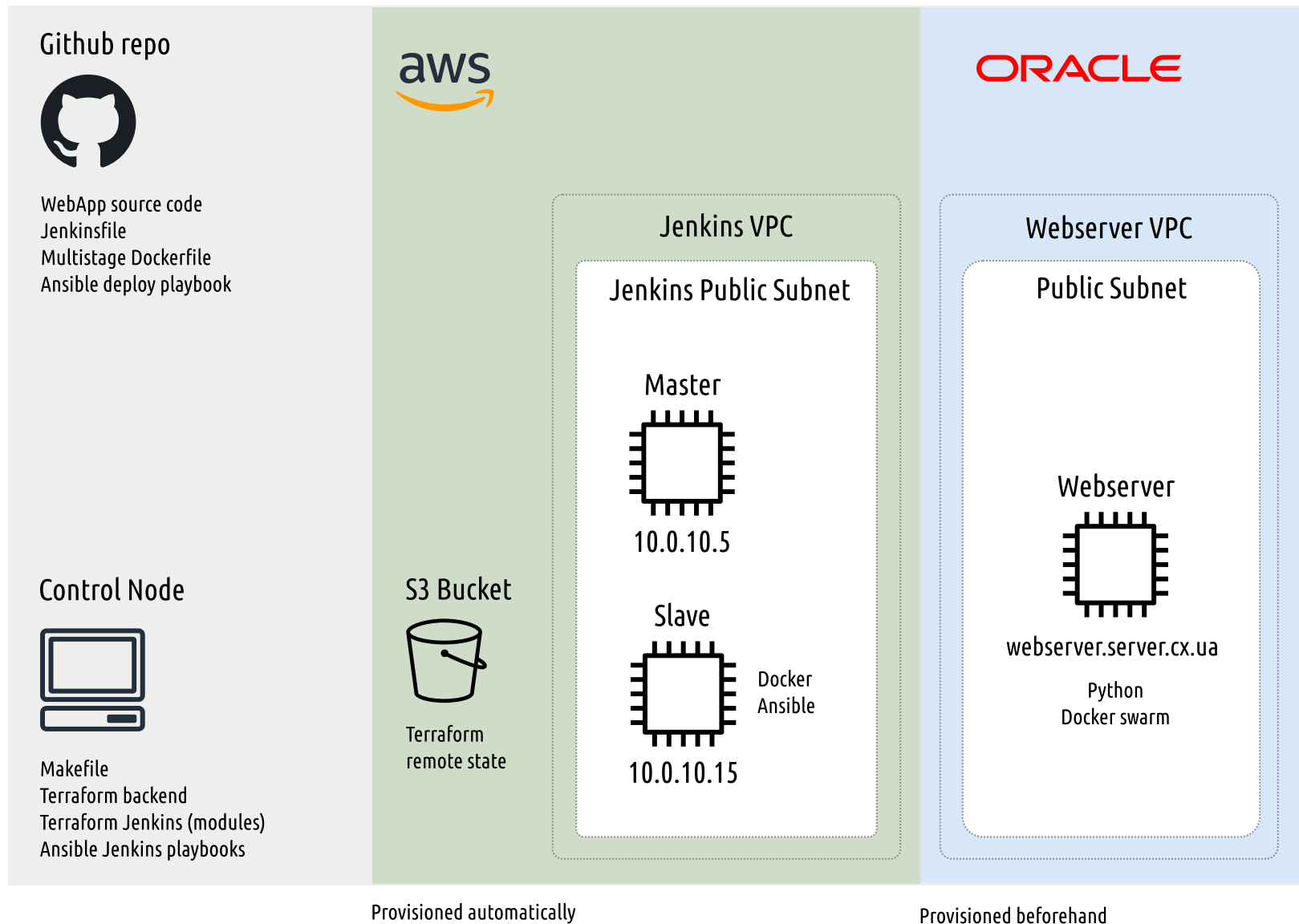
- automate Jenkins infrastructure deployment
- build a CI/CD pipeline with Zero-Downtime deployment



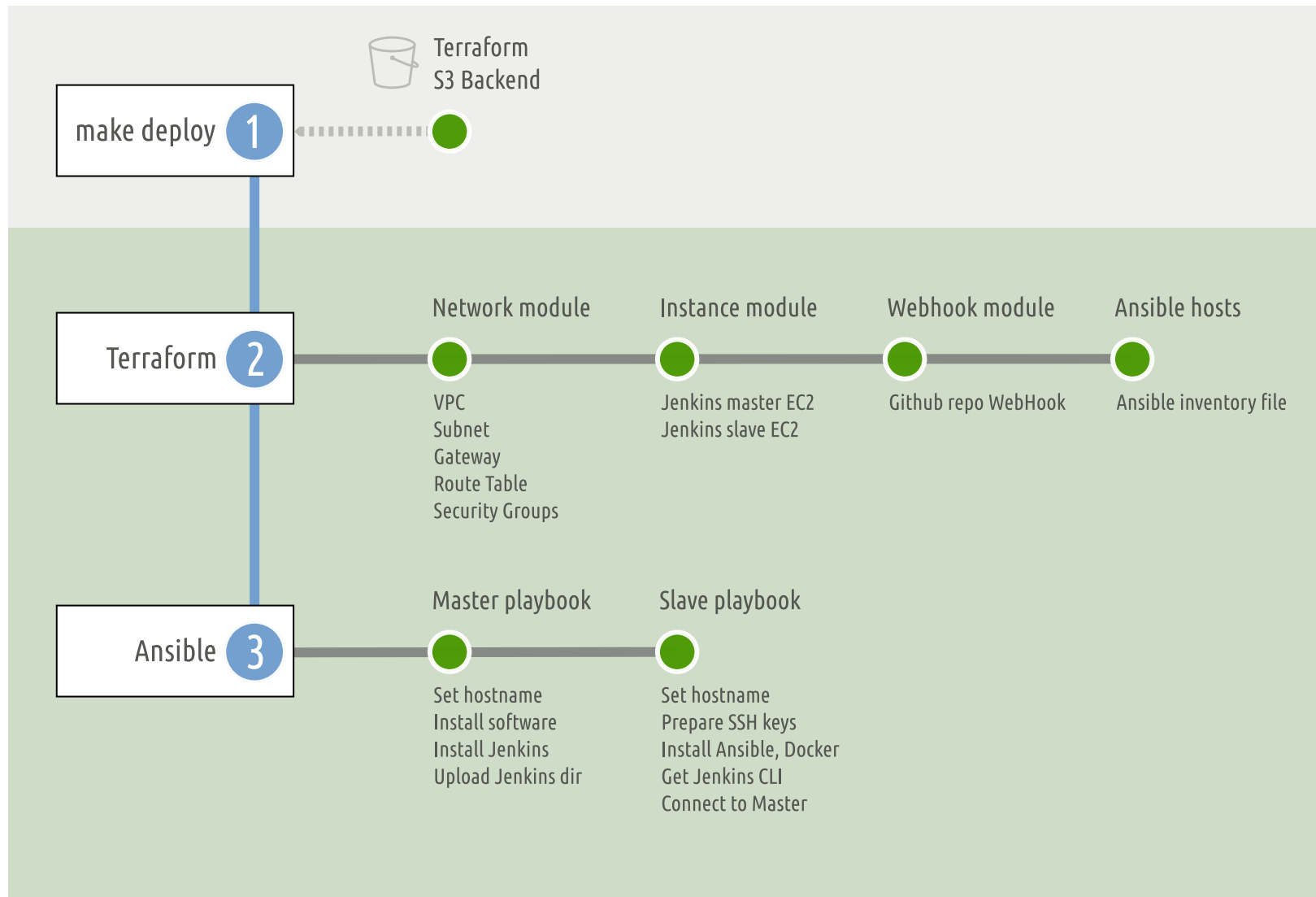
Technology Stack



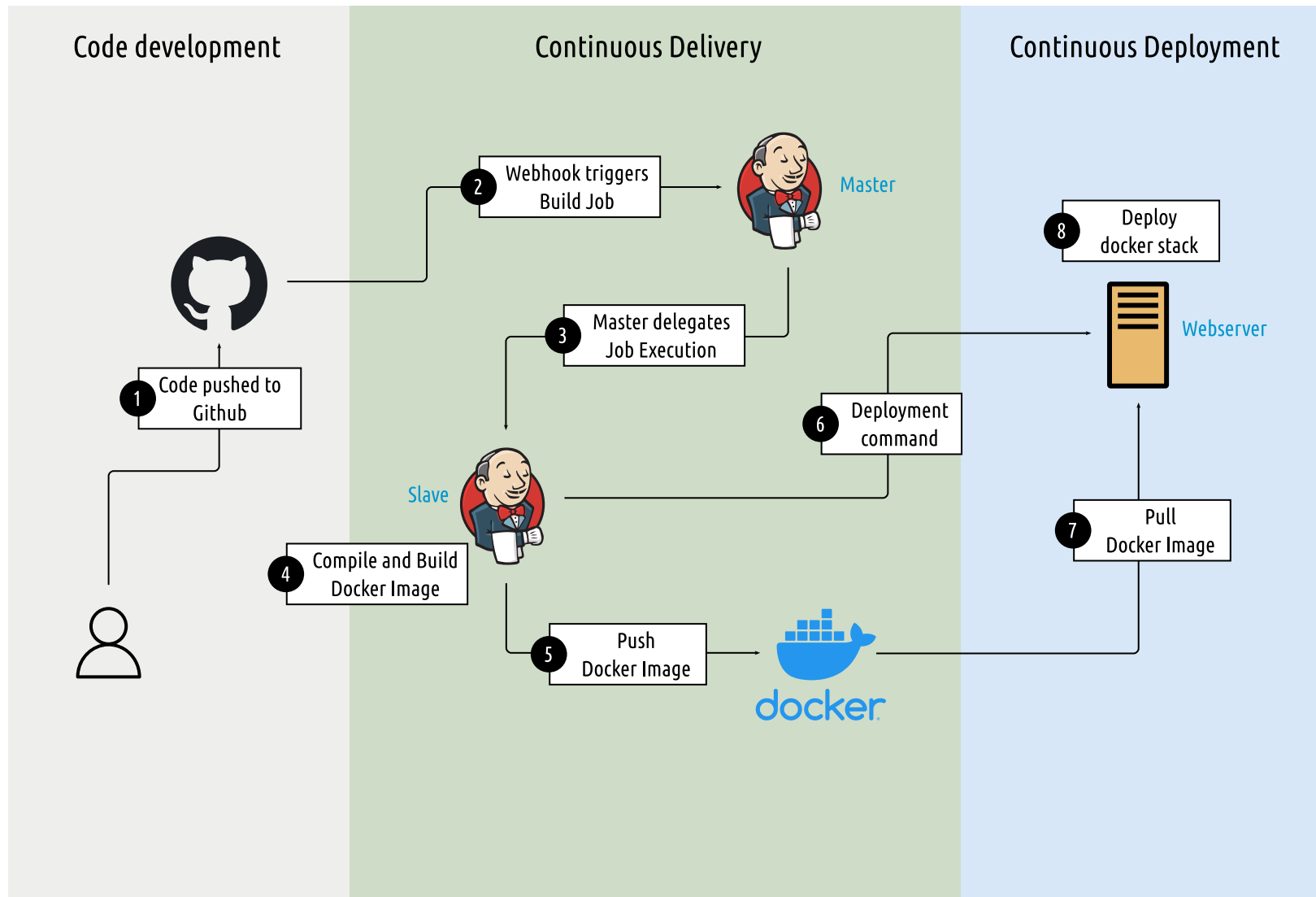
Project Infrastructure



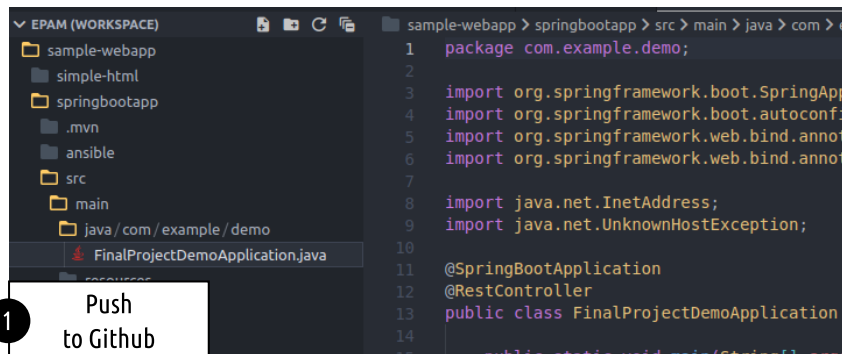
Infrastructure Provisioning



CI/CD Outline



CI/CD Details



```
1 package com.example.demo;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.web.bind.annotation.RestController;
6 import org.springframework.web.bind.annotation.RequestMapping;
7
8 import java.net.InetAddress;
9 import java.net.UnknownHostException;
10
11 @SpringBootApplication
12 @RestController
13 public class FinalProjectDemoApplication {
14
15     public static void main(String[] args) {
```

1 Push to Github

```
stage('Deploy') {
    steps {
        echo "=====
        dir('springbootapp/ansible') {
            withCredentials([usernamePassword(
                credentialsId: 'dockerhub',
                passwordVariable: 'DOCKER_PASSWORD',
                usernameVariable: 'DOCKER_USERNAME')])
            {
                sh 'cat hosts.ini'
                sh 'ansible -i hosts.ini all -m ping'
                sh ''
                ansible-playbook deploy.yaml --extra-vars "DOCK
                DOCK
                IMAG
```

6 Deployment command

```
FROM maven:3.8.6-eclipse-temurin-17
ENV MAVEN_OPTS="-XX:+TieredCompilation"
WORKDIR /opt/demo
COPY pom.xml .
RUN mvn dependency:go-offline -B
COPY ./src ./src
RUN mvn clean install -Dmaven.test.skip=true
FROM eclipse-temurin:17.0.5_8-jre-alpine
WORKDIR /opt/demo
ARG JAR_FILE=*.jar
COPY --from=BUILD /opt/demo/target/dependency/*.jar app.jar
EXPOSE 8080
CMD ["java", "-jar", "app.jar"]
```

4 Build Docker Image

```
stage('Compile') {
    steps {
        echo "=====
        dir('springbootapp') {
            sh 'sed -i "s/%DATE%/${date}/" src/main/java/com/
            sh 'sed -i "s/%BUILD%/${BUILD_NUMBER}/" src/main/
        }
    }
}

stage('Build Image') {
    steps {
        echo "=====
        dir('springbootapp') {
            dockerImage = docker.build DOCKER_IMAGE + ":%$
```

4 Compile the source code

```
webapp:
  image: ${IMAGE}
  hostname: virtual-server
  ports:
    - 80:8080
  networks:
    - webapp-network
  healthcheck:
    test: "wget --no-verbose --tries=1 --spider http://localhost:8080/"
  deploy:
    mode: replicated
    replicas: 2
    update_config:
      order: start-first
      failure_action: rollback
      delay: 5s
```

8 Deploy docker stack

```
name: Deploy docker image to warehouse
hosts: server
vars:
  ansible_python_interpreter: /usr/bin/python3.8
  image_name: "{{ IMAGE_NAME }}"
  # container_name: simple-webapp
  stack_path: /home/ubuntu/docker/demo

tasks:
  - name: Stack Deploy | Upload docker-compose.yaml
    ansible.builtin.copy:
      src: docker-compose.yaml
      dest: "{{ stack_path }}/docker-compose.yaml"

  - name: Stack Deploy | Log into DockerHub
    community.docker.docker_login:
      username: "{{ DOCKER_USERNAME }}"
      password: "{{ DOCKER_PASSWORD }}"

  - name: Stack Deploy | Deploy Docker Stack
    community.docker.docker_stack:
      state: present
      name: demo
      with_registry_auth: true
      compose:
        - "{{ stack_path }}/docker-compose.yaml"
      environment:
        IMAGE: "{{ image_name }}"

  - name: Stack Deploy | Log out of DockerHub
    community.docker.docker_login:
      state: absent
```

7 Pull Docker Image

Final Project: Live Demonstration

Thank you!