**INTRODUCTION**

This is your general course work, which covered of CI/CD pipeline with Git, Ansible, Docker, Jenkins, Cloud tools.

**SUCCESS CRITERIA**

After the course, on the interview, you should demonstrate simple CI/CD pipeline, which handle build and deploy of simple Spring Boot application with the next scenarios:

* Jenkins should build each Pull Requests for verifications
* Jenkins should pull latest changes and build service after each commit to develop/master branch via any trigger
* Jenkins should create artifact of the service and upload to storage system
* Jenkins should triggering deployment of CI environment after each success build with latest built artifact (Continuous Integration / Continuous Deployment)
* Jenkins should deploy QA environment with the any created artifact on demand (Continuous Delivery)
* During interview, student should demonstrate coursework to an interviewer for discussion

**REQUIREMENTS:**

* Any Public Cloud Free Tier account, like AWS/AZURE/GCP
* Any Public SCM account, like GitHub/GitLab/Bitbucket

**Step 1: SCM activities**

Login to SCM under your account and fork Spring Boot example to your new public repository with the any name. Application example <https://github.com/spring-projects/spring-petclinic>

**Step 2: Cloud activities**

* All Public Cloud infrastructure should be created via Terraform
* Possible infrastructure approach
  + All located in Compute service
  + Mix of Compute and Containers services
  + All located in Containers services (mostly preferred)

**Step 3: Prepare DevTools**

DevTools: latest JDK8, Git, Jenkins and Ansible. As an artifact storage solution you could use Docker Trusted Registry, Jfrog Artifactory, Nexus or any other

**Step 4: Configure CI/CD tools**

**DTR/Artifactory/Nexus:**

* Create repository for artifacts. Also, this repository will be used for deployment procedure
* Create user, which will have access to created repository

**Jenkins:**

* Install required plugin, like git, maven, matrix role
* Disable anonymous access and create some user
* Create build pipeline flow with the next steps - CHECKOUT, BUILD, CREATE ARTIFACT, DEPLOY
* Create deployment jobs for CI and QA environment's

Steps descriptions:

* CHECKOUT: should be triggered after each commit to develop/master branch in repository
* BUILD: to build application use **./mvnw package**
* CREATE ARTIFACT: create Docker Image or Jar and push it to artifactory storage. Artifact should have next name convention: <application-name>.<build-version>
* CI DEPLOY: Ansible role should deploy latest version of application from artifactory storage to CI environment

**Ansible:**

* Create deployment role which pull and run application.
  + For JAR – use next to run application: **java –jar path/to/jar/file.jar --server.port=8080** (port should be parametrized)
  + For Containers – pull and run application
* Create provisioning role, which install java, docker, create system user, create folders and other required staff on CI and QA environment's (only for Compute approach)

**Step 5: BUILD and DEPLOY**

So, now we got simple CI/CD process, let's improve it:

* The first, let's containerize app with Docker for the build and deployment flow
  + Create Compute instance and install Docker (only for Compute approach)
  + Create Ansible build role, which
    - Initiate generation of application.dockerfile which will be used to build application container
    - Build application container
    - Push application container to DTR
  + Create Ansible deploy role, which deploy container on Docker node during CI DEPLOY step
  + Create Jenkins jobs for deploy of CI and QA environment's with containers

* Build and Deploy flow on Jenkins should be in declarative pipeline (Jenkins DSL)
  + Build process should use jenkinsfile, which located in build branch
  + Deploy process should use application\_deploy.jenkinsfile and this file should be pulled from SCM

* Add possibility to choice version of artifact / containers during deployment. In simple - it could be common string field. Advanced - drop down menu with artifacts list

**ADDITIONAL RESOURCES**

* <https://jenkins.io/doc/book/pipeline/>