CLOUDBEES
UNIVERSITY

CloudBees Jenkins Platform: Pipeline with Docker

Labs

Table of Contents

Introduction

This workbook is designed to supplement your CloudBees Jenkins training. It consists of a sequence of lab exercises that will introduce Jenkins Pipeline and Docker concepts and best practices.

The Project - Part 2: Exercise

The solution to all tasks is located at the end. Please try to solve them by yourself and look at the solutions only if you get stuck or want to validate your work.

The exercise consists of practicing with other job types. Since this training is language-agnostic, you are not expected to know the language. You will not need any additional tools. All the tasks should be completed with Docker.

Before diving into the exercise, please SSH into the machine you are assigned and enter the */mnt/training-books-ms/exercises* directory. All the code you'll need is inside.

Task: Convert the job into a template

The requirements for this task are as follows.

- Convert the *docker-flow-proxy* Pipeline job into a Job template called *docker-flow-proxy-template*.
- Convert repository name and container name into job properties.
- Create a new job called *docker-flow-proxy-from-template*. The job should be based on the *docker-flow-template*.

Task: Create a Multi-Branch Pipeline

The requirements for this task are as follows.

- Create a new job called *docker-flow-proxy-from-git*.
- The type of the job should be *Multibranch Pipeline*.
- The repository with Jenkins file should be *https://github.com/cloudbees/training-books-ms*.
- The *master* branch should be excluded.

Solution: Convert the job into a template

- Click the *New Item* link located in the left-hand menu in the home screen.

- Type *docker-flow-template* as *Item Name*, select *Job Template*, and click the *OK* button.
- Type *Docker Flow Template* in the *Display Name* field.
- Click the *Add* button in the *Attributes* section, type *repository* as ID and *Repository* as name.
- Click the *Add* button in the *Attributes* section, type *container* as ID and *Container* as name.
- Select *Groovy template for Pipeline* as the *Tranformer* type.
- Write the following script inside the *Pipeline Script* field.

```
node("cd") {
    git branch: 'pipeline', url: "https://github.com/${repository}"

    stage 'test'
    docker.image("golang").inside('-u 0:0') {
        sh 'ln -s $PWD /go/src/docker-flow'
        sh 'cd /go/src/docker-flow && go get -t && go test --cover -v'
        sh 'cd /go/src/docker-flow && go build -v -o docker-flow-proxy'
    }

    stage 'build'
    docker.build("localhost:5000/${container}")
    docker.image("localhost:5000/${container}").push()
    archive '${container}'
}

checkpoint 'deploy'

node('production') {
    stage 'deploy'
    try {
        sh "docker rm -f ${container}"
    } catch(e) { }
    docker.image("localhost:5000/${container}").run("--name ${container} -p
8081:80 -p 8082:8080")
}
```

- Click the *New Item* link located in the left-hand menu in the home screen.
- Type *docker-flow-proxy-from-template* as *Item Name*, select *Docker Flow Template*, and click the *OK* button.
- Set *docker-flow-proxy-from-template* as *Name, _cloudbees/training-books-ms* as *Repository*, and *docker-flow-proxy* as *Container*.
- Click the *Save* button.

Solution: Create a Multi-Branch Pipeline

- Click the *New Item* link located in the left-hand menu in the home screen.

- Type *docker-flow-proxy-from-git* as *Item Name*, select *Multibranch Pipeline*, and click the *OK* button.
- Select *Git* from the *Add Source* drop-down.
- Type *https://github.com/cloudbees/training-books-ms* as *Project Repository*.
- Click the *Advanced* button and type *master* in the *Exclude branches* field.
- Click the *Save* button.

Go back to slides