Testing in DevOps

DOu – Certified Tester in DevOps (CTD)
Exercise Solutions

HO-1.5.2(HO-0)
Exercise - Demonstrate how to apply the main features of continuous delivery and deployment tools

- Add the stages in jenkins pipeline and trigger the jenkins build

HO-1.5.2(HO-0)
Jira Setup

- Create & Launch AWS Ubuntu 18.04 instance(t2.medium, 8GB HD)with all traffic
  (Refer to “Testing in DevOps_Exercise-Solutions_Pre-requisite_V0.1” slide-deck)
- Connect with putty
- Run below commands on terminal
  - sudo apt-get update
  - sudo apt-get install docker.io
  - docker --version
  - sudo docker volume create --name jiraVolume
  - sudo docker volume ls
  - sudo docker run -v jiraVolume:/var/atlassian/application-data/jira --name="jira" -d -p 8080:8080 atlassian/jira-software
  - sudo docker images
  - sudo docker container ls -a
- Launch <http://your aws ip address:8080> on browser
- JIRA UI will be displayed

HO-1.5.2(HO-0)
Jira Setup…

- Click “Set it up for me”
- Click “Continue to MyAtlassian” button
- Do login in JIRA if not done already
- Select “JIRA Software(Server)”
- Enter your organization name
- Click “Generate License”
- Fill the form(Email, Username, Password)
- Click Next(it can take 10-15min to be finished)

HO-1.5.2(HO-0)
Jira Setup…

- Click “Let's Get Started” button
- Select the language English
- Click Continue > Click Next
- Click “Create new project” button (by default Scrum software development option will be selected)
- Click Next
- Click Select
- Enter project name --> e.g. auto-bug-logging
- Enter key --> e.g. AUT
- Click Submit
- You will navigate to JIRA home page
- Click “Dashboards” tab
- Click “View System Dashboard”

HO-1.5.2(HO-0)
Exercise Solution - Demonstrate how to apply the main features of continuous delivery and deployment tools

- Make sure you are already logged in Jenkins and be on Dashboard UI
- Add Install, Launch Tomcat Server, War deployed, System Test stages in Jenkinsfile.txt on local machine:
  - Clean(already added)
  - Pmd(already added)
  - Compile(already added)
  - Code Analysis(already added)
  - Unit Test(already added)
  - Code Coverage(already added)
  - **Install**
  - **Launch Tomcat Server**
  - **War deployed on Tomcat Server**
  - **API Test**
  - **System Test**

HO-1.5.2(HO-0)
Add Stage – Install

1. **Install** defines a stage that appears on the Jenkins UI
2. **sh** runs the Maven command to install the web application without running the unit tests

```
        //Code starts for stage Install
        stage('Install') {              (1)
    steps {
        sh 'mvn install -DskipTests'    (2)
    }
}
        //Code ends for stage Install
```

HO-1.5.2(HO-0)
Add Stage – Launch Tomcat Server

1. **Launch Tomcat Server** defines a stage that appears on the Jenkins UI
2. **sh** runs the command launch Tomcat server

```
        //Code starts for stage Launch Tomcat Server
        stage('Launch Tomcat Server') {                 (1)
    steps {
        sh '/tmp/apache-tomcat-9.0.20/bin/startup.sh'   (2)
    }
}
        //Code ends for stage Launch Tomcat Server
```

HO-1.5.2(HO-0)
Add Stage – Deploy War on Tomcat Server

1. **Deploy War on Tomcat Server** defines a stage that appears on the Jenkins UI
2. **sh** runs the command to move war file from code project location to Tomcat's location

```
//Code starts for Deploy War on Tomcat Server
        stage('Deploy War on Tomcat Server') {          (1)
    steps {
        sh 'cp /var/jenkins_home/workspace/verity-devops-ex/target/ExpenseApp-1.war /tmp/apache-tomcat-9.0.20/webapps'  (2)
    }
}
        //Code ends for stage Deploy War on Tomcat Server
```

HO-1.5.2(HO-0)
Add Stage – API Test

1. **API Test** defines a stage that appears on the Jenkins UI
2. **sh** runs the Maven command to run the system tests

```
        //Code starts for stage API Test
        stage('API Test') {                                                    (1)
    steps {
                    //Change git url below as per your forked github repository URL
                    git url: 'https://github.com/umangsaltuniv/EMAPITests-ex.git'
                    sh 'mvn -Dtest=ExpenseManagerAPITest test'                 (2)
        }
}
        //Code ends for stage API Test
```