Testing in DevOps

DOu – Certified Tester in DevOps (CTD)
Exercise Solutions

HO-2.4.1(HO-2)
Exercise - Perform an exercise providing a hint of integrating a static analysis tool in the DevOps pipeline and checking the test coverage of a given code and its tests

- Add Code Coverage stage in jenkins pipeline and trigger the jenkins build. Discuss code coverage reports.

HO-2.4.1(HO-2)
Exercise Solution - Perform an exercise providing a hint of integrating a static analysis tool in the DevOps pipeline and checking the test coverage of a given code and its tests

- Make sure you are already logged in Jenkins and be on Dashboard UI
- Add Code Coverage stage 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**

HO-2.4.1(HO-2)
Add Stage – Code Coverage

1. **Code Coverage** defines a stage that appears on the Jenkins UI
2. **sh** runs the Maven command to perform the code coverage

```
        //Code starts for stage Code Coverage
        stage('Code Coverage') {          (1)
    steps {
        sh 'mvn jacoco:report'            (2)
    }
}
        //Code ends for stage Code Coverage
```

HO-2.4.1(HO-2)
Exercise Solution - Perform an exercise providing a hint of integrating a static analysis tool in the DevOps pipeline and checking the test coverage of a given code and its tests

- Saving Jenkinsfile.txt & Commit it on GitHub
- Running The Pipeline
  - Go to Jenkins Dashboard
  - Build will be triggered automatically because of webhook
  - Click Build no. link
  - Click Console Output link
  - Verify build result is successful
  - Verify jacoco reports

HO-2.4.1(HO-2)
JaCoCo Reports

- You can JaCoCo report at below location in container:

  /var/jenkins_home/workspace/<your pipeline project name>
  /target/site/jacoco/index.html

ExpenseApp

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.expense.controller		42%		38%	10	15	54	83	6	11	0	2
com.expense.entity		67%		n/a	11	47	35	90	11	47	1	3
com.expense.dto		52%		n/a	3	22	5	33	3	22	0	1
com.expense.service.impl		70%		25%	7	18	10	35	5	16	0	3
com.expense.mapper		7%		n/a	6	7	4	5	6	7	0	1
com.expense		58%		n/a	1	3	2	4	1	3	0	1
com.expense.securityconfig		100%		n/a	0	5	0	24	0	5	0	1
Total	521 of 1,234	58%	8 of 12	33%	38	117	110	274	32	111	1	12

HO-2.4.1(HO-2)
Get JaCoCo reports using tomcat

- Steps are given in PMD slide deck for below actions:
  - Setup tomcat server
  - Modify tomcat-users.xml
  - Modify Server.xml
  - Verify tomcat web app manager UI

HO-2.4.1(HO-2)
Get JaCoCo reports using tomcat…

- Create a putty duplicate session
- Run below commands:
  - **pwd**
  - **ls**
  - **sudo docker container ls –a**

    Note down the container id from above command where jenkins is up & running and me tion in below command
  - **sudo docker cp <<container id>>:var/jenkins_home/workspace/verity-devops-ex/target/site/jacoco myjacoco**

    Above command will create the "myjacoco" folder and move it to "/home/ubuntu" directory

HO-2.4.1(HO-2)
Get JaCoCo reports using tomcat…

- **ls**
- **sudo mv myjacoco /var/lib/tomcat9/webapps/**

  Above command will move “myjacoco” folder from "/home/ubuntu" directory to "/var/lib/tomcat9/webapps/" directory
- **ls**
- **cd /var/lib/tomcat9/webapps/**
- **ls**

HO-2.4.1(HO-2)
Get JaCoCo reports using tomcat…

- Go to tomcat web application manager page on browser
- Refresh the page
- You will see "/myjacoco" will be displayed under Applications > Path
- Click “/myjacoco” link
- See the jacoco reports on browser