Continuous Integration/Deployment

This lab demonstrates the concepts of continuous integration/continuous deployment (CI/CD) for our development. We will be leveraging Github Actions which comes integrated with Github, for automating the process of CI/CD.

Learning Outcomes

After completing the lab, you will be able to:

1. Describe how to create pipeline using Github Actions
2. Continuously deploy your code to Kubernetes cluster

Set-up Github Actions

1. Configure the jobs within the pipeline workflow.

```
Pipeline workflow:

 Job sequence in the workflow:

 1.build-artifact ---> 2.deploy-to-cluster

 Tasks within each job:
 1.build-artifact:
   a.Build with Gradle
   b.Upload Artifact
   c.Build-Docker-Image

 2.deploy-to-cluster:
   a.Install Kubectl
   b.Configure AWS credentials
   c.Login to production cluster
   d.Create K8s deployments and resources
```

2. Create the following secrets in github

```
DOCKER_USERNAME
DOCKER_PASSWORD
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY
```

3. To add secrets, click on settings and select secrets from the left navigation menu within the github repository.
4. The aws access key id and secret will be provided to you.
5. Create `.github/workflows/pipeline.yaml` in the root project directory with the configuration. Refer to pipeline.yaml
6. Update the tag name in `pages-deployment.yaml` file to `pipeline`
7. Push your code to git repository
8. Navigate to github actions menu in the github dashboard to see the progress of the pipeline.
9. Upon successful completion of the workflow, you should be able to see the deployment and other objects in K8s cluster.

Advanced Usecases

1. Automating security concerns
2. Automating code quality compliance
3. Advanced testing using test clusters