# **DevOps Assessment**

# Task1

#### **Jenkins Installation:**

Ensure Jenkins is installed and running.

- Install necessary plugins:
  - o Pipeline
  - o NodeJS
  - Docker
  - SonarQube Scanner
- Configure Jenkins system settings as needed.

# Node.js and NPM:

- Install Node.js on the Jenkins server. You can use a tool like nvm for version management.
- Ensure that the node and npm commands are available in the Jenkins environment.

# **Pipeline Job Creation:**

Create a new multi-branch pipeline job in Jenkins.

# **Pipeline Configuration:**

Configure source code management with the GitHub repository URL. Add the Jenkinsfile path in the repository.

#### **Credentials and Secrets:**

Ensure all necessary credentials and secrets are configured in Jenkins.

### **Run Pipeline:**

Save and run the Jenkins pipeline job.

```
[Pipeline] sn
 + npm install
 > hackathon-starter@8.0.1 postinstall
 > patch-package && npm run scss
 patch-package 8.0.0
 Applying patches...
 passport@0.6.0 🗸
 passport-oauth2@1.7.0 🗸
 > hackathon-starter@8.0.1 scss
 > sass --no-source-map --load-path=./ --update ./public/css:./public/css
 > hackathon-starter@8.0.1 prepare
 > if [ "$NODE_ENV" != "production" ]; then husky install; fi
 husky - Git hooks installed
 up to date, audited 786 packages in 4s
 168 packages are looking for funding
  run `npm fund` for details
 2 moderate severity vulnerabilities
Some issues need review, and may require choosing
```

```
[Pipeline] sn
+ npm install
> hackathon-starter@8.0.1 postinstall
> patch-package && npm run scss
patch-package 8.0.0
 Applying patches...
passport@0.6.0 /
 passport-oauth2@1.7.0 🗸
> hackathon-starter@8.0.1 scss
> sass --no-source-map --load-path=./ --update ./public/css:./public/css
> hackathon-starter@8.0.1 prepare
 > if [ "$NODE_ENV" != "production" ]; then husky install; fi
 husky - Git hooks installed
up to date, audited 786 packages in 4s
168 packages are looking for funding
  run `npm fund` for details
 2 moderate severity vulnerabilities
Some issues need review, and may require choosing
```

# **Install Plugins:**

Ensure Jenkins has the necessary plugins installed:

- Pipeline
- Docker
- Trivy

### **Configure Global Tools:**

Configure Docker and Trivy as global tools in Jenkins.

#### **Create Jenkinsfile:**

Add a Jenkinsfile to the root of your GitHub repository:

# **Configure Jenkins:**

Create a new multi-branch pipeline job in Jenkins.

Configure the GitHub source for the pipeline, pointing to your repository.

Define Jenkins credentials for Docker registry if needed.

### **Run Pipeline:**

Trigger the pipeline manually or set up webhooks for automatic triggering on GitHub events.

#### **View Results:**

Jenkins will execute the pipeline stages.

If Trivy detects HIGH or CRITICAL vulnerabilities, the pipeline will abort, preventing deployment.

# **Install Jenkins Plugins:**

Install the necessary Jenkins plugins for multi-branch pipelines, Git integration, NodeJS, Docker, and SonarQube Scanner.

### **Configure Global Tools:**

Configure global tools in Jenkins for NodeJS and the SonarQube Scanner.

#### **Set Up GitHub Source:**

Create a new multi-branch pipeline job in Jenkins. Configure the GitHub source for the job to point to the hackathon-starter repository.

#### **Create Jenkinsfile:**

Add a Jenkinsfile at the root of the repository. Define stages for checkout, build, test, and deploy. Include a stage for SonarQube analysis

### **Configure Sonar Qube in Jenkins:**

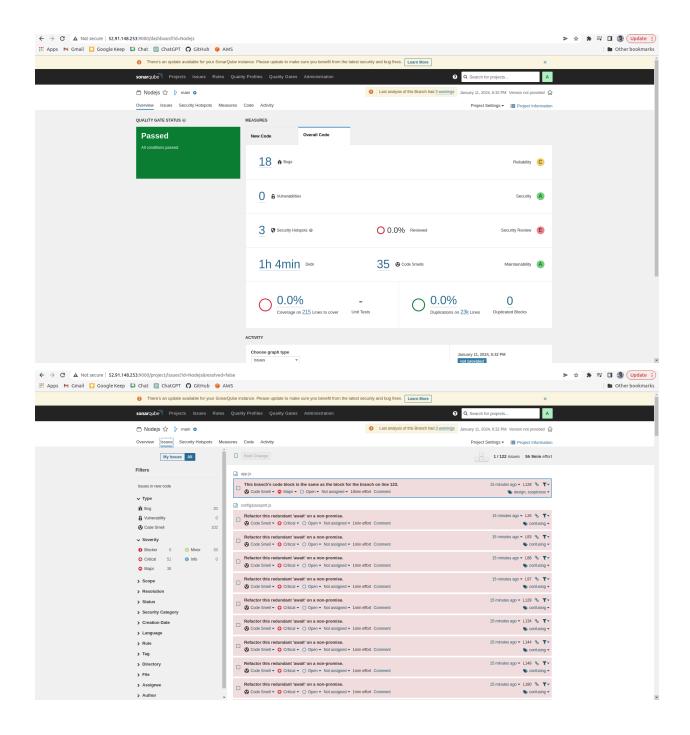
Configure SonarQube server details in Jenkins. Provide SonarQube server URL and authentication token.

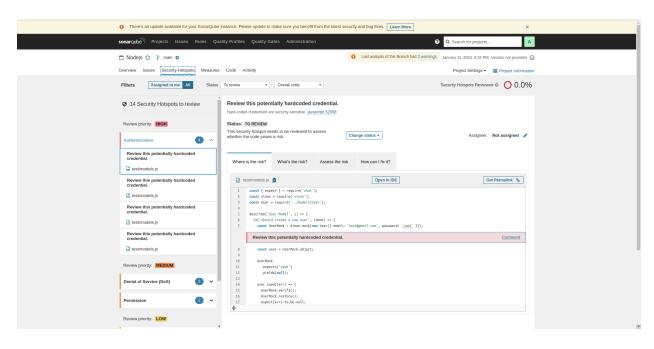
# **Set Up Docker Registry Credentials:**

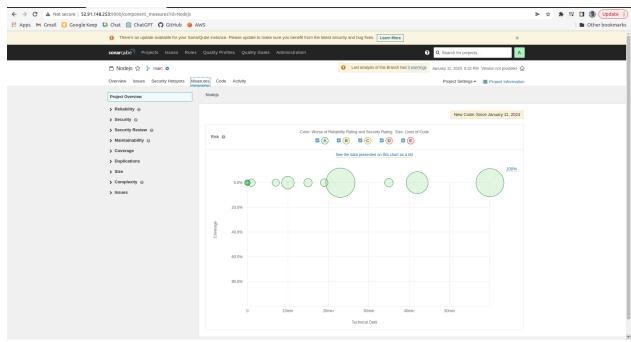
If deploying Docker images, configure Docker registry credentials in Jenkins.

### **Run Pipeline:**

Run the pipeline job in Jenkins. Monitor the progress in the Jenkins console.







# **Install Jenkins Plugins:**

Install Jenkins plugins for Git, Docker, Kubernetes, and GitHub integration.

# **Set Up Credentials:**

Add GitHub credentials and Docker registry credentials in Jenkins.

### **Setup K8 Cluster**

Create a kubernetes using terraform and ansible

# **Global Tool Configuration:**

Configure NodeJS and Docker tools in Jenkins global tool configuration.

#### **Create Jenkinsfile:**

Write a Jenkinsfile in the repository with stages for checkout, build, test, and deploy.

### **Define Multi-Branch Pipeline:**

Create a new multi-branch pipeline job in Jenkins, link it to the GitHub repository, and set up the repository source.

### **Pipeline Script:**

In the Jenkinsfile, define pipeline steps:

- Checkout code.
- Build and tag Docker image.
- Push Docker image to the registry.
- Deploy the application using Kubernetes manifests.

#### Jenkins Build Trigger:

Configure the Jenkins job to build on push events and pull requests for all branches.

### **Run Pipeline:**

Run the pipeline manually or wait for automatic triggers to test the CI process.

root@Master:/home/ubuntu/dep# kubectl get pods				
NAME	READY	STATUS	RESTARTS	AGE
argocd-application-controller-0	1/1	Running	0	132m
argocd-applicationset-controller-568bbf57d8-7hr27	1/1	Running	0	132m
argocd-dex-server-bf667bf68-g9nz2	1/1	Running	0	132m
argocd-notifications-controller-67bfbc5bf8-lngt8	1/1	Running	0	132m
argocd-redis-db489f5bf-445b9	1/1	Running	0	132m
argocd-repo-server-85d8455d48-nwvff	1/1	Running	0	132m
argocd-server-549c85d755-tflkv	1/1	Running	0	132m
hackerrank-67bdbcf9f-mmwrg	2/2	Running	0	17s

root@Master:/home/ubuntu/dep# kub	ectl get svc		_		
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
argocd-applicationset-controller	ClusterIP	10.96.187.170	<none></none>	7000/TCP	132m
argocd-dex-server	ClusterIP	10.99.2.212	<none></none>	5556/TCP,5557/TCP	132m
argocd-redis	ClusterIP	10.98.48.150	<none></none>	6379/TCP	132m
argocd-repo-server	ClusterIP	10.110.66.248	<none></none>	8081/TCP	132m
argocd-server	ClusterIP	10.109.227.25	<none></none>	80/TCP,443/TCP	132m
nackerrank-service	LoadBalancer	10.101.163.93	<pending></pending>	8080:31830/TCP	26s
kubernetes	ClusterIP	10.96.0.1	<none></none>	443/TCP	156m

# Install ArgoCD:

Set up ArgoCD in your Kubernetes cluster.

### **Create Helm Charts:**

Design Helm charts for your application.

Organize the charts in a way that allows easy configuration via the values.yaml file.

```
drwxrwxr-x 4 rahul rahul 4096 Jan 11 23:32 ./
drwxrwxr-x 3 rahul rahul 4096 Jan 11 23:30 ../
drwxrwxr-x 2 rahul rahul 4096 Jan 11 23:30 charts/
-rw-rw-r-- 1 rahul rahul 119 Jan 11 23:55 Chart.yaml
drwxrwxr-x 2 rahul rahul 4096 Jan 11 23:31 templates/
-rw-rw-r-- 1 rahul rahul 159 Jan 11 23:55 values.yaml
```

```
root@Master:/home/ubuntu/dep# kubectl get pods
                                                        READY
                                                                 STATUS
                                                                           RESTARTS
argocd-application-controller-0
                                                        1/1
                                                                 Running
                                                                                       132m
                                                        1/1 1/1
argocd-applicationset-controller-568bbf57d8-7hr27
                                                                 Running
                                                                           0
                                                                                       132m
argocd-dex-server-bf667bf68-g9nz2
                                                                 Running
                                                                           0
                                                                                       132m
argocd-notifications-controller-67bfbc5bf8-lngt8
                                                        1/1
                                                                 Running
                                                                                       132m
argocd-redis-db489f5bf-445b9
                                                                 Running
                                                                                        132m
argocd-repo-server-85d8455d48-nwvff
                                                                 Running
                                                                                       132m
argocd-server-549c85d755-tflkv
                                                        1/1
                                                                 Running
                                                                                        132m
hackerrank-67bdbcf9f-mmwrg
root@Master:/home/ubuntu/dep# kubectl get svc
                                                        2/2
                                                                                        17s
                                                                 Running
```