Deploying a Jenkins Pipeline with SonarQube, Docker, Minikube, and ArgoCD

This guide will walk you through the steps to set up a Jenkins pipeline, SonarQube integration, Docker, Minikube, and ArgoCD on an EC2 instance, along with configuring Jenkins for CI/CD.

Step 1: Create an EC2 Instance (t2.large)

- Create a new EC2 instance with type t2.large and select the desired OS.
- Open necessary ports, such as:
    - **Port 22** for SSH access.
    - **Port 8080** for Jenkins.
    - **Port 9000** for SonarQube (if needed).

Step 2: Copy the .pem File to WSL (Optional)

1. **Copy the PEM file to the WSL:**

    cp jenkins.pem ~/.ssh/

    chmod 400 ~/.ssh/jenkins.pem

    ls -l ~/.ssh/jenkins.pem

2. **SSH into the EC2 instance:**

    ssh -i ~/.ssh/jenkins.pem ubuntu@54.221.94.107 # Replace with your EC2 IP

Step 3: Install Java and Jenkins

Install Java

1. Update package repositories:

    sudo apt update

2. Install Java:

    sudo apt install openjdk-17-jre

3. Verify Java installation: java -version

Install Jenkins

1. Add Jenkins key and repository:

   curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

   echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

2. Update repositories and install Jenkins:

   sudo apt-get update

   sudo apt-get install jenkins

Open Port 8080 in AWS EC2 Security Groups

- Navigate to your EC2 instance in AWS.
- Under **Security**, click on **Security groups**.
- Add an inbound rule to allow **TCP port 8080** (or all traffic if necessary).

Step 4: Access Jenkins

1. Open a web browser and navigate to the Jenkins server:

   http://<EC2_PUBLIC_IP>:8080

2. Retrieve the Jenkins initial admin password:

   sudo cat /var/lib/jenkins/secrets/initialAdminPassword

3. Complete the Jenkins setup wizard.

Step 5: Install Docker Pipeline Plugin in Jenkins

1. Log in to Jenkins.

2. Go to **Manage Jenkins** > **Manage Plugins**.
3. In the **Available** tab, search for **Docker Pipeline** and **SonarQube Scanner**.
4. Install the plugins and restart Jenkins.

Step 6: Configure SonarQube on EC2 Instance

1. Install unzip and create a SonarQube user:

   sudo apt install unzip

   sudo adduser sonarqube

2. Download and install SonarQube:

   sudo su - sonarqube

   wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.4.0.54424.zip

   unzip sonarqube-9.4.0.54424.zip

   sudo chmod -R 755 /home/sonarqube/sonarqube-9.4.0.54424

   sudo chown -R sonarqube:sonarqube /home/sonarqube/sonarqube-9.4.0.54424

   cd sonarqube-9.4.0.54424/bin/linux-x86-64/

   ./sonar.sh start

3. Access SonarQube via EC2 public IP on port 9000:

   http://<EC2_PUBLIC_IP>:9000

Step 7: Authenticate SonarQube in Jenkins

1. Generate a **SonarQube token**:
   - Go to **SonarQube** > **My account** > **Security** > **Tokens** and generate a new token.
2. In Jenkins, go to **Manage Jenkins** > **Manage Credentials** > **Global credentials** > **Add Credentials**:

- Add a **Secret text** credential with the generated SonarQube token.

3. Similarly, add DockerHub and GitHub credentials.

Step 8: Install Docker on EC2 Instance

1. Install Docker:

   sudo apt update

   sudo apt install docker.io

2. Grant Jenkins and Ubuntu users permission to use Docker:

   sudo usermod -aG docker jenkins

   sudo usermod -aG docker ubuntu

   sudo systemctl restart docker

3. Restart Jenkins:

   http://<EC2_PUBLIC_IP>:8080/restart

Step 9: Install Kubectl and Minikube on Local Machine

Install Kubectl

1. Install kubectl:

   curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

   chmod +x kubectl

   sudo mv kubectl /usr/local/bin/

   kubectl version --client

Install Minikube

1. Install Minikube:

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

chmod +x minikube-linux-amd64

sudo mv minikube-linux-amd64 /usr/local/bin/minikube

2. Start Minikube:

   minikube start --driver=docker

3. Verify the cluster status:

   kubectl get nodes

Troubleshooting Docker Permissions

If you encounter issues with Docker, add your user to the Docker group:

sudo usermod -aG docker $USER

newgrp docker

docker run hello-world

Step 10: Install ArgoCD on Minikube

1. Install ArgoCD via Operator Hub (or using the ArgoCD installation commands):

   kubectl create namespace argocd

   kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

2. Access the ArgoCD UI:

   - Edit the ArgoCD service to use NodePort:

   kubectl edit svc argocd-server -n argocd

   - Set type: NodePort.

3. Get the ArgoCD admin password:

kubectl get secret argocd-initial-admin-secret -n argocd -o jsonpath="{.data.password}" | base64 -d

4. Access ArgoCD at:

minikube service argocd-server -n argocd --url

Step 11: Create and Run the Jenkins Pipeline

1. In Jenkins, create a **new item**.
2. Select **Pipeline**, then configure the pipeline:
    - Use **Git** for the repository URL.
    - Select the **main** branch.
    - Specify the **Jenkinsfile** script path.
3. Run the pipeline and monitor the results in Jenkins.

Step 12: Deploy with ArgoCD

1. Create a new application in ArgoCD:
    - Go to **Applications** > **New Application**.
    - Set the repository URL and path to the **deployment file**.
2. View your deployed application:

minikube service <app-name> --url