ip addr add 127.255.255.255/16 dev lo

ip addr delete 127.255.255.255/16 dev lo

ip link set eth1 down

ip link set eth1 up

ip route add default via 192.0.2.2 dev eth0

Install ***Anchore Container Image Scanner Plugin***

<https://docs.anchore.com/current/docs/deployment/docker_compose/>

docker pull sonarqube  
docker run -d --name sonarqube -p 9000:9000 sonarqube

admin

admin

“My Account” -> “Security” -> “Generate Tokens”.

sqa\_62980597c4751a6d387a44f4da6563aed6052fc4

New token "XP\_2600" has been created. Make sure you copy it now, you won't be able to see it again!

sqa\_62980597c4751a6d387a44f4da6563aed6052fc4

docker-workflow

**Steps to Create a Jenkinsfile**

1. **Create a New File**: In the root of your Git repository, create a new file named Jenkinsfile.
2. **Add Pipeline Code**: Copy the entire pipeline code (the one you’ve written) into the Jenkinsfile.
3. **Commit the Jenkinsfile**: Add and commit the Jenkinsfile to your repository:

bash

1. git add Jenkinsfile
2. git commit -m "Add Jenkinsfile for CI/CD pipeline"
3. git push origin master # or the relevant branch
4. **Configure Jenkins Job**:
   * In Jenkins, create a new job or configure an existing one.
   * Select **Pipeline** as the job type.
   * In the job configuration, under **Pipeline**, choose **Pipeline script from SCM**.
   * Select your SCM (e.g., Git) and provide the repository URL where your Jenkinsfile is stored.
5. **Build the Job**: Once configured, you can build the job in Jenkins, and it will read the pipeline definition from your Jenkinsfile.

**Example of a Simple Jenkinsfile**

Here’s how your Jenkinsfile should look:

groovy

pipeline {

agent any

environment {

SONARQUBE\_URL = 'http://192.168.1.85:9000'

SONARQUBE\_TOKEN = credentials('XP\_2600') // Use Jenkins credentials for security

GIT\_REPO = 'https://github.com/XP2600-hub/nhorizon-java-container.git'

DOCKER\_IMAGE = 'xp2600/japp:latest'