# 3103 Guide

## Follow this for Jenkins:

https://www.jenkins.io/doc/tutorials/build-a-node-js-and-react-app-with-npm/

docker network create jenkins

dind

```
docker run --name jenkins-docker --detach ^
  --privileged --network jenkins --network-alias docker ^
  --env DOCKER_TLS_CERTDIR=/certs ^
  --volume jenkins-docker-certs:/certs/client ^
  --volume jenkins-data:/var/jenkins_home ^
  --publish 3000:3000 --publish 5000:5000 --publish 2376:2376 ^
  docker:dind
```

Dockerfile

```
FROM jenkins/jenkins:2.361.4-jdk11
USER root
RUN apt-get update && apt-get install -y lsb-release
RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
  https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
  https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
RUN apt-get update && apt-get install -y docker-ce-cli
USER jenkins
RUN jenkins-plugin-cli --plugins "blueocean:1.25.8 docker-workflow:521.v1a_a_dd2073b_2e"
```

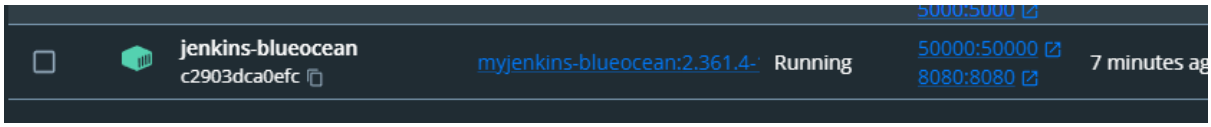
Cd to Jenkinsfile location

docker build -t myjenkins-blueocean:2.361.4-1 .

Build blueocean image

```
docker run --name jenkins-blueocean --detach ^
  --network jenkins --env DOCKER_HOST=tcp://docker:2376 ^
  --env DOCKER_CERT_PATH=/certs/client --env DOCKER_TLS_VERIFY=1 ^
  --volume jenkins-data:/var/jenkins_home ^
  --volume jenkins-docker-certs:/certs/client:ro ^
  --volume "%HOMEDRIVE%%HOMEPATH%":/home ^
  --restart=on-failure ^
  --env JAVA_OPTS="-Dhudson.plugins.git.GitSCM.ALLOW_LOCAL_CHECKOUT=true" ^
  --publish 8080:8080 --publish 50000:50000 myjenkins-blueocean:2.361.4-1
```

Open docker image




Find pwd in the logs

Create pipeline

- Go back to Jenkins, log in again if necessary and click create new jobs under Welcome to Jenkins!
- Note: If you don't see this, click New Item at the top left.
- In the Enter an item name field, specify the name for your new Pipeline project (e.g. simple-node-js-react-npm-app).
- Scroll down and click Pipeline, then click OK at the end of the page.
- ( Optional ) On the next page, specify a brief description for your Pipeline in the Description field (e.g. An entry-level Pipeline demonstrating how to use Jenkins to build a simple Node.js and React application with npm.)
- Click the Pipeline tab at the top of the page to scroll down to the Pipeline section.
- From the Definition field, choose the Pipeline script from SCM option. This option instructs Jenkins to obtain your Pipeline from Source Control Management (SCM), which will be your locally cloned Git repository.
- From the SCM field, choose Git.
- In the Repository URL field, specify the directory path of your locally cloned repository above, which is from your user account/home directory on your host machine, mapped to the /home directory of the Jenkins container - i.e.
- For Windows - /home/Documents/GitHub/*replace with proj name*
- Click Save to save your new Pipeline project. You're now ready to begin creating your Jenkinsfile, which you'll be checking into your locally cloned Git repository.

## Integrating OWASP Dependency Check into Jenkins Pipeline

Git clone: https://github.com/0xprime/JenkinsDependencyCheckTest

Jenkinsfile

```
pipeline {
        agent any
        stages {
                stage('Checkout SCM') {
                        steps {
                                git '/home/*add path here*/JenkinsDependencyCheckTest'
                        }
```

```
                }

                stage('OWASP DependencyCheck') {
                        steps {
                                dependencyCheck additionalArguments: '--format HTML --format
XML', odcInstallation: 'Default'
                        }
                }
        }
        post {
                success {
                        dependencyCheckPublisher pattern: 'dependency-check-report.xml'
                }
        }
}
```

git add .

git commit -m “Initial OWASP”

## php unit testing

Jenkinsfile

```
pipeline {
        agent {
                docker {
                        image 'composer:latest'
                }
        }
        stages {
                stage('Build') {
                        steps {
                                sh 'composer install'
                        }
                }
                stage('Test') {
                        steps {
          sh './vendor/bin/phpunit tests --log-junit logs/unitreport.xml -c tests/phpunit.xml tests'
        }
                }
        }
        post {
                always {
                        junit testResults: 'logs/unitreport.xml'
                }
        }
}
```

Php selenium:

Take note: (if got space in sh file must quote, also add double // in front of any paths)

#!/usr/bin/env sh

set -x

docker run -d -p 80:80 --name my-apache-php-app -v //home/documents/"sit year 3"/"3103"/jenkins-php-selenium-test/jenkins-php-selenium-test/src:/var/www/html php:7.2-apache

sleep 1

set +x

echo 'Now...'

echo 'Visit http://localhost to see your PHP application in action.'

## Lab 8:

SAST

Use docker terminal

```
cd /var/jenkins_home

curl http://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven- 3.6.3-bin.tar.gz --output apache-maven-3.6.3-bin.tar.gz

tar -xvzf apache-maven-3.6.3-bin.tar.gz && cd apache-maven-3.6.3

pwd
```

Website for SAST check (depends what language):

Replace them inside the Jenkinsfile

https://github.com/jenkinsci/warnings-ng-plugin/blob/master/SUPPORTED-FORMATS.md

Check jenkinsfile if got quotes make sure same line (make below same line):

e.g. sh '/var/jenkins_home/apache-maven-3.6.3/bin/mvn --batch-mode -V -U -e

checkstyle:checkstyle pmd:pmd pmd:cpd findbugs:findbugs'

## Lab 9:

Install sonarqube scanner on jenkins

```
docker pull sonarqube
```

Ip for sonarqube scanner in Jenkins put ip of computer

Put secret key in config sys, add -> jenkins -> put key in secret field

## Lab 10:

Check docker settings: Settings -> Docker Engine -> buildkit = false

docker compose

| |
|---|
| docker exec -it compose-mysqldb-1 mysql -u root -p<br><br>for password look at compose file<br><br>workaround for compose cannot find dockerfile<br><br>docker compose untick v2, in settings<br><br>put dockerfile in root folder |