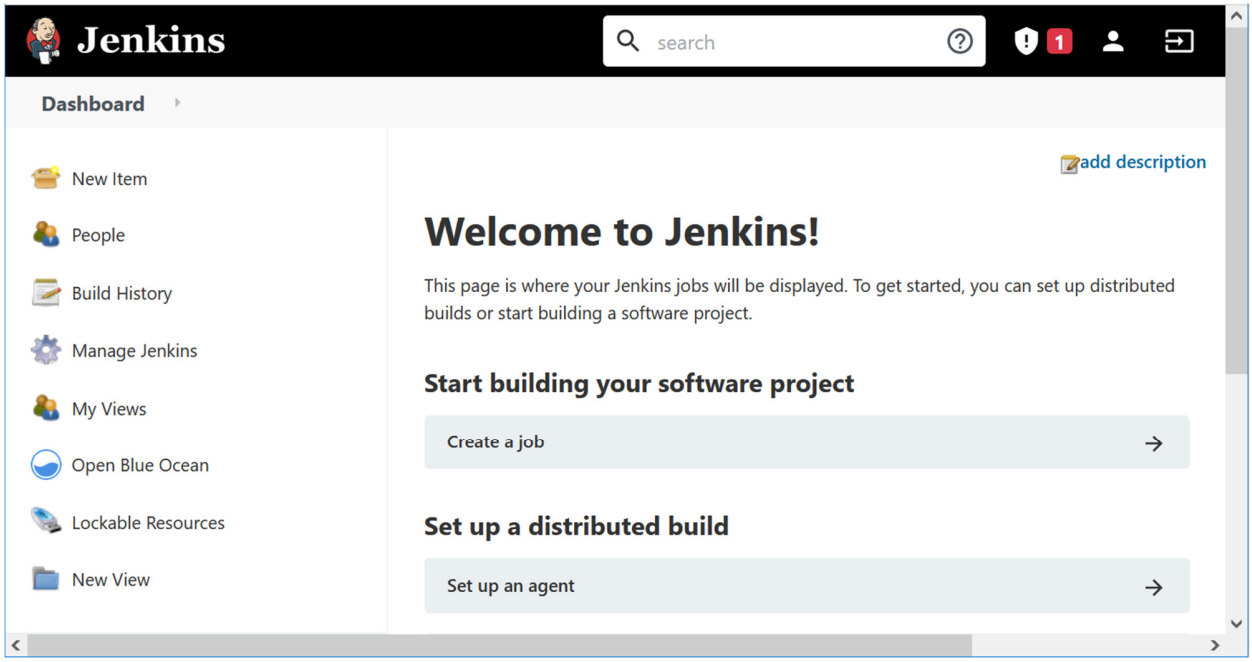
Getting to Know Jenkins with Blue Ocean UI on Docker

Overview



In this lab, you are going to learn how to use Jenkins, the leading open source automation server supported by a large and growing community of developers, testers, designers and other people interested in continuous integration, continuous delivery (CI/CD) and modern software delivery practices.

Built on the Java Virtual Machine (JVM), it provides more than 1,500 plugins that extend Jenkins to automate with practically any technology software delivery teams use. Since 2019, Jenkins has surpassed 200,000 known installations making it the most widely deployed automation server.

Outcomes

Upon completion of this session, you should be able to:

- Install and run Jenkins on Docker locally at your laptop;
- Create Jenkins pipeline to build, test and deliver a sample web application;
- Obtain a valid free CA cert from Let's Encrypt and run Jenkins with NGINX on Docker at your team Digital Ocean VM over HTTPS;
- Hook your team Github repository to Jenkins at your team Digital Ocean VM to commence CI/CD implementation phase for your team project.

1: Installation

Jenkins can be installed on Windows, MacOS, Linux or Docker. To get you familiar and comfortable with Docker which is a valuable skill in the industry, we will continue your use of Docker in Lab-X02 SecurityRAT to install and run Jenkins on Docker in this Lab exercise.

Pre-requisite: Docker

Docker is a platform for running applications in an isolated environment called a "container" (or Docker container).

To install Docker on Windows, follow the instructions at:
https://docs.docker.com/desktop/windows/install/

To install Docker on Ubuntu, follow the instructions at:
https://docs.docker.com/engine/install/ubuntu/

To test that your Docker is installed successfully, open a terminal and run the hello-world Docker image which you should get the following:

```
$ docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:ca0eeb6fb05351dfc8759c20733c91def84cb8007aa89a5bf606bc8b315b9fc7
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
...
```

A list of useful Docker CLI commands are available at:
https://dockerlabs.collabnix.com/docker/cheatsheet/

Co-requisite: Git

With your familiarity of Git for ICT2101/ICT2201 Introduction to Software Engineering, we will continue to use Git as the Source Code Management (SCM) tool, which will be hooked to Jenkins to automatically check out your codes, build and test them whenever there are changes.

In case you need to re-install or update Git, go to https://git-scm.com/downloads to download a copy for your OS platform.

Similar as Docker, you can open a terminal to run Git commands. A list of useful commands are available at https://training.github.com/downloads/github-git-cheat-sheet.pdf

Jenkins with Blue Ocean UI Plugin

Applications like Jenkins can be downloaded as read-only "images" (or Docker images), each of which is run on Docker as a container. A Docker container is in effect a "running instance" of a Docker image. From this perspective, an image is stored permanently more or less (i.e. insofar as image updates are published), whereas containers are stored temporarily.

To install and run Jenkins with Blue Ocean UI on Docker locally at your laptop, follow the Jenkins Installation Guide at https://www.jenkins.io/doc/book/installing/docker/

Notes:

1. Besides running myjenkins-blueocean:1.1 Jenkins container, the installation guide also runs docker:dind (docker-inside-docker) container in order to be able to execute Docker commands inside Jenkins pipeline which is required for the next Section 2: Tutorial.

2. Alternatively, without the need to run docker:dind, you may try docker-outside-of-docker which is running Jenkins to directly access docker:sock using user root as follows:

On Windows:

```
docker run --name jenkins-blueocean --rm --detach ^
  --user root ^
  --volume /var/run/docker.sock:/var/run/docker.sock ^
  --volume jenkins-data:/var/jenkins_home ^
  --volume "%HOMEDRIVE%%HOMEPATH%":/home ^
  --publish 8080:8080 myjenkins-blueocean:1.1
```

On Linux:

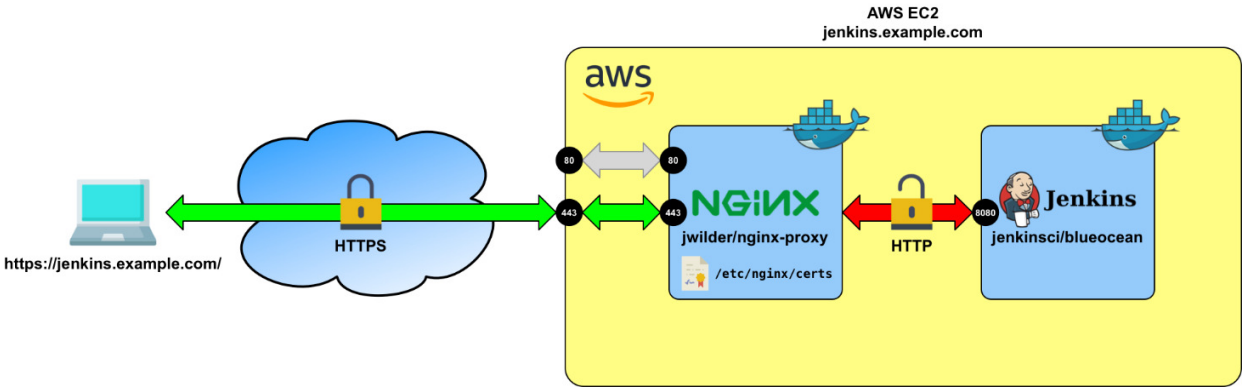
```
docker run --name jenkins-blueocean --rm --detach \
  --user root \
  --volume /var/run/docker.sock:/var/run/docker.sock \
  --volume jenkins-data:/var/jenkins_home \
  --volume "$HOME":/home \
  --publish 8080:8080 myjenkins-blueocean:1.1
```

2: Tutorial

To understand the use of CI/CD for secure web application development, you will now try using Jenkins to automate the process of building, testing and delivering of a sample web application. This process of building, testing and delivering is going to be repetitive during the implementation phase which explains why CI/CD is getting popular in the industry. Please follow the Jenkins Tutorial Guide at https://www.jenkins.io/doc/tutorials/build-a-node-js-and-react-app-with-npm/

3: Jenkins for Team Project

The easier way to run Jenkins over HTTPS is to link with a reverse proxy like NGINX. You may wish to refer to the helpful guide at https://itnext.io/setting-up-https-for-jenkins-with-nginx-everything-in-docker-4a118dc29127?gi=79e1fe44f1fe



Note: Please use domain name jenkins.*yourteamname*.sitict.net and port 8443 for accessing Jenkins over HTTPS at your team Digital Ocean VM.

4: Jenkins and Github

There are a number of examples from the Internet on the configuration of Github webhook to trigger Jenkins to automatically check-out, build and test the codes after each commit by any team members; e.g.:

https://prodevans.com/index.php/2020/06/13/how-to-integrate-your-github-repository-to-your-jenkins-project/

Notes:

1. Due to the need for ICT Github Repositories admin right, we will configure the webhooks for all teams with the payload URL https://jenkins.*yourteamname*.sitict.net:8443/github-webhook

2. Github has removed password authentication and instead replaced it with a personal access token (PAC). If you do not have a PAC yet, please follow the instructions to generate https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

3. To integrate Jenkins with Github, at the Jenkins Source Code Management section, select Git and enter the Repository URL https://*yourPAC*@github.com/SIT-ICT3x03/Team...git

Hope that you will have an enjoyable experience with Jenkins!

END OF DOCUMENT