

# Lab 1: Environment Setup

Author : Gourav Shah

Publisher: School of Devops

Version : v2024.04.30.01

---

Following are the learning objectives of this lab

- You will launch Jenkins along with Docker Server (DIND) as your Continuous Integration Lab Environment
- You will do the initial Jenkins configurations including installing plugins, configuring admin user and get Jenkins ready to start building your CI workflows

## Setup Jenkins with Docker

Here you are going learn, how to setup jenkins using docker.

Ensure that you have Docker Installed on your system. Refer to [Get Docker](#) for instructions on setting up Docker. Start by validating your environment with,

```
docker version
docker-compose
```

The first command should show you both , client and server information. The second command should return the help menu for docker compose, which validate that you have Docker installed, started and docker-compose ready to go.

Now, you could launch a jenkins container on your docker host by using recommended jenkins image similar to whats recommended in the official documentation [Docker](#) using the following sequence of commands.

```
git clone https://github.com/udbc/bootcamp.git
cd bootcamp/jenkins
docker-compose up -d
```

[output]

```
Creating network "jenkins_custom" with driver "bridge"
Creating volume "jenkins_jenkins-docker-certs" with default driver
Creating volume "jenkins_jenkins-data" with default driver
Pulling docker (docker:dind)...
dind: Pulling from library/docker
df20fa9351a1: Pull complete
25ad7478873d: Pull complete
4684f6177b5d: Pull complete
8ba584e970af: Pull complete
3cdc74d2b06d: Pull complete
4cf5a0d07c1f: Pull complete
fca0ccc462d5: Pull complete
8a08b8f19995: Pull complete
e60a2aec8c6b: Pull complete
84edc63b9e2e: Pull complete
a8919df01d06: Pull complete
Digest: sha256:973c39d7eadb05e45923173bc484961e5b6b527d8b0693c0881e07e9d2fa8ee7
Status: Downloaded newer image for docker:dind
Pulling jenkins (jenkinsci/blueocean:)...
latest: Pulling from jenkinsci/blueocean
df20fa9351a1: Already exists
1cb481a13af0: Pull complete
f5efbd400588: Pull complete
f2d0037ca04a: Pull complete
f50c4373c0c3: Pull complete
cfa42149c07e: Pull complete
e6784003bb85: Pull complete
73a1e720e18b: Pull complete
2e41633b95a1: Pull complete
83587819eac3: Pull complete
1389b752912a: Pull complete
feb326d4942c: Pull complete
05fc0701116d: Pull complete
06a6bc155e06: Pull complete
72b0e728c908: Pull complete
53b5c96969ae: Pull complete
Digest: sha256:257958006c0fa8fe499a40b8a6ea32c145a781f2d4cfda6efe67d5a56c022c78
Status: Downloaded newer image for jenkinsci/blueocean:latest
Creating jenkins_docker_1 ... done
Creating jenkins_jenkins_1 ... done
```

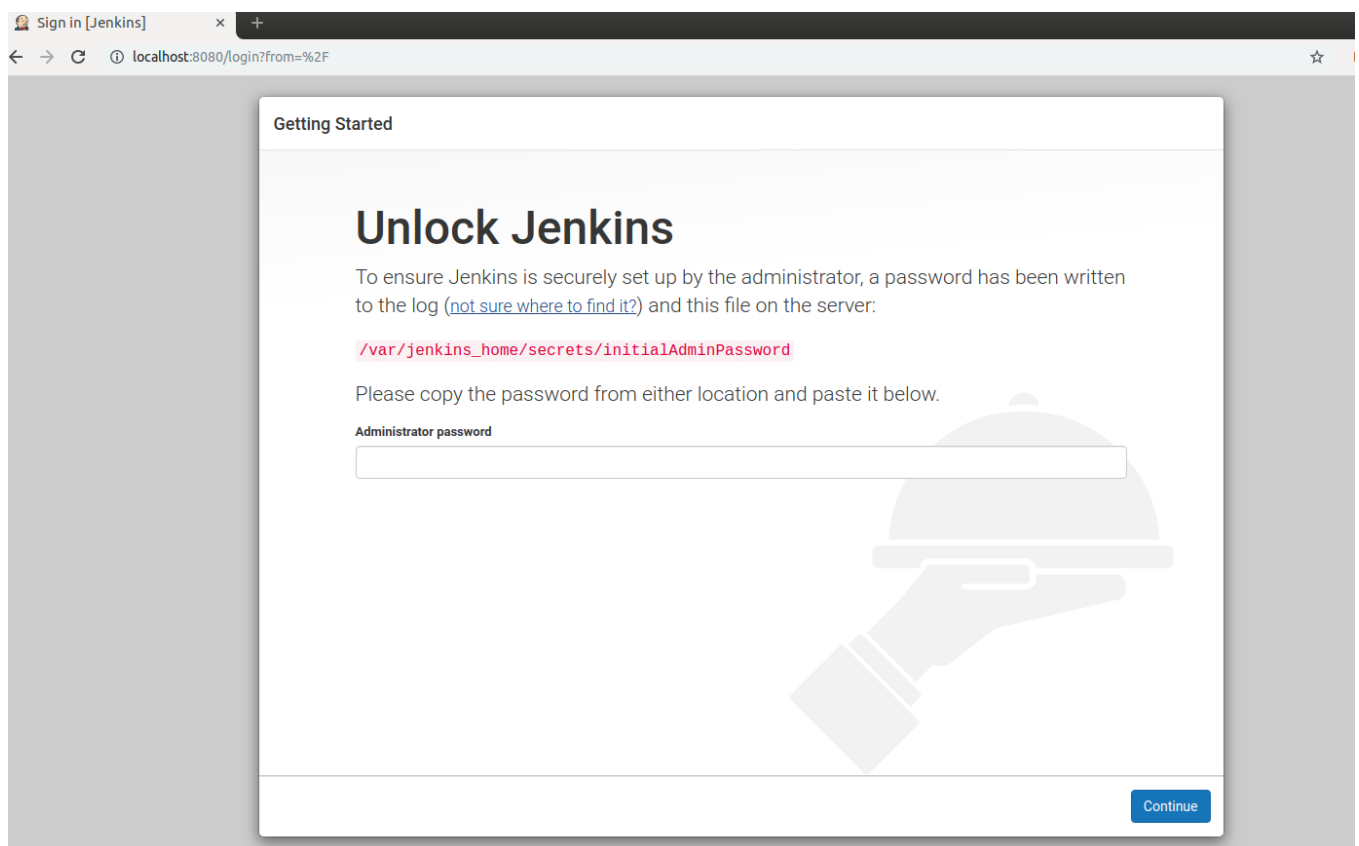
You could validate that Jenkins is setup along with Docker using the following command

```
docker-compose ps
```

[output]

Name	Command	State
Ports		
-----		
jenkins_docker_1	dockerd-entrypoint.sh	Up
0.0.0.0:2376->2376/tcp		
jenkins_jenkins_1	/sbin/tini -- /usr/local/b ...	Up
0.0.0.0:50000->50000/tcp, 0.0.0.0:8080->8080/tcp		

You could now access jenkins UI by browsing to `http://IPADDRESS:8080`



To fetch the initialAdminPassword use the following command

```
docker-compose logs jenkins
```

[sample output]

```
2020-02-09 14:42:40.900+0000 [id=28] INFOjenkins.install.SetupWizard#init:
```

```
*****
*****
*****
```

Jenkins initial setup is required. An admin user has been created and a password generated.

Please use the following password to proceed to installation:

```
4856656f53ea4c76a5cca69dfda5656f
```

This may also be found at: /var/jenkins\_home/secrets/initialAdminPassword

```
*****
*****
*****
```

```
2020-02-09 14:42:51.261+0000 [id=28] INFOjenkins.InitReactorRunner$1#onAttained:
Completed initialization
```

Copy the exact password that you see on your screen and paste it on the Jenkins UI to unlock.

In the next step , choose install suggested plugins to configure the default plugins automatically.

# Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

## Install suggested plugins

Install plugins the Jenkins community finds most useful.

## Select plugins to install

Select and install plugins most suitable for your needs.

You can also observe the progress of plugins installation process as follows,



## Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	Editor bundle ** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) ** Pipeline: SCM Step ** Pipeline: Groovy ** Pipeline: Job ** Display URL API Mailer ** Apache HttpComponents Client 4.x API ** Pipeline: Basic Steps Gradle ** Pipeline: Milestone Step ** Jackson 2 API ** Pipeline: Input Step ** Pipeline: Stage Step
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
✓ Pipeline	🔄 GitHub Branch Source	🔄 Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	
✓ Git	🔄 Subversion	🔄 SSH Slaves	<input type="radio"/> Matrix Authorization Strategy	
<input type="radio"/> PAM Authentication	🔄 LDAP	🔄 Email Extension	✓ Mailer	

If the plugins installation process fails, you could retry it.

Once plugins are installed, you will create the admin user using the form presented.

# Create First Admin User

Username:	<input type="text" value="admin"/>
Password:	<input type="password" value="....."/> 
Confirm password:	<input type="password" value="....."/> 
Full name:	<input type="text" value="Devops Admin"/>
E-mail address:	<input type="text" value="admin@exmple.com"/>

completing this process will get you to jenkins configuration page, and continuing that will take you to the main page where you could get started creating jobs from.

## Stopping, Starting and Resetting Jenkins

While you start building your Continuous Integration workflows, there would be times when you would need to stop and start this environment e.g. you shut down your laptop at the end of the day and then start if the next day, and you would want your learning lab environment to be back up and running.

This can be achieved by gracefully stopping your lab environment with,

```
docker-compose stop
```

and then when you start your system, by executing

```
docker-compose up -d
```

Ensure both the commands are run from the same path where `docker-compose.yaml` for this project exist e.g. `.../bootcamp/jenkins` .

If you ever need to reset this environment completely follow this process,

```
cd bootcamp/jenkins
docker-compose down
docker volume rm jenkins_jenkins-data jenkins_jenkins-docker-certs
docker-compose build
docker-compose up -d
```

Do note we are deleting the volumes which are shared between docker and jenkins containers. This is been designed so that even if you restart or recreate the containers, you do not loose your jobs etc. Only when you want to reset (starting from scratch again) should you use this sequence of instructions.

#cicd/labsv3