Assignment 3 :: Setup Docker Containers as Build Agents for Jenkins

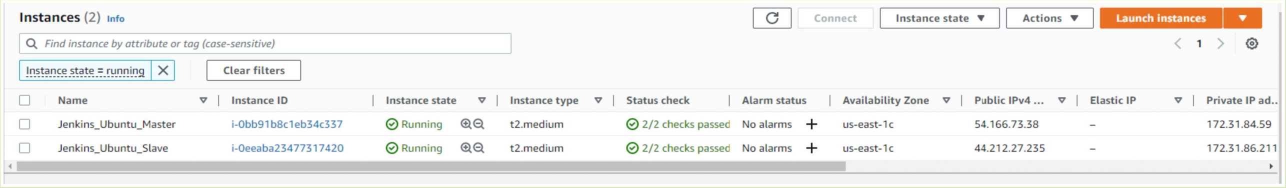
Table of Contents

Step 1 :: Preparing EC2 Instances

Step 1.1 :: Prepare 2 instances

OS :: Ubuntu 20.04

Security_Group :: All Traffic



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 ...	Elastic IP	Private IP ad..
□	Jenkins_Ubuntu_Master	i-0bb91b8c1eb34c337	✓ Running	t2.medium	✓ 2/2 checks passed	No alarms +	us-east-1c	54.166.73.38	–	172.31.84.59
□	Jenkins_Ubuntu_Slave	i-0eeaba23477317420	✓ Running	t2.medium	✓ 2/2 checks passed	No alarms +	us-east-1c	44.212.27.235	–	172.31.86.211

Step 1.2 :: Install Jenkins in Master Node

$ sudo apt -y update

$ sudo apt install openjdk-11-jdk -y

$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \

/etc/apt/sources.list.d/jenkins.list'

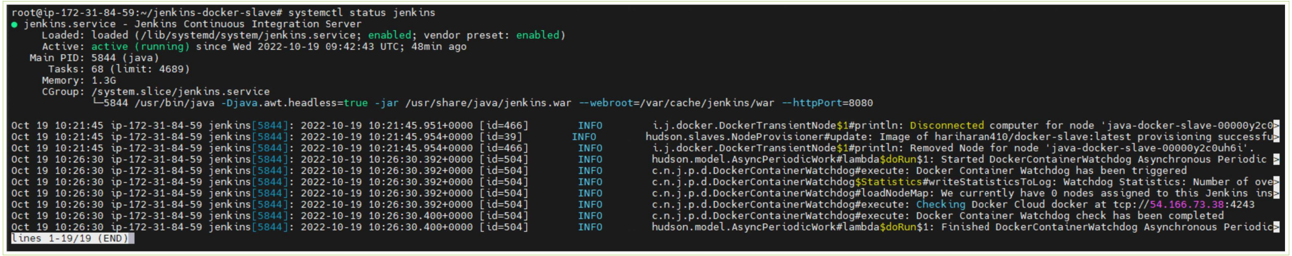
$ sudo apt update -qy

$ sudo apt install jenkins -qy

$ sudo systemctl start jenkins

$ sudo systemctl enable jenkins

$ sudo systemctl status jenkins

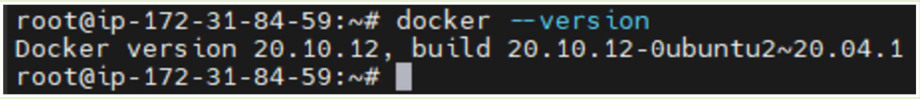


```
root@ip-172-31-84-59:~/jenkins-docker-slave# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-10-19 09:42:43 UTC; 48min ago
   Main PID: 5844 (java)
      Tasks: 68 (limit: 4689)
     Memory: 1.3G
     CGroup: /system.slice/jenkins.service
             └─5844 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Oct 19 10:21:45 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:21:45.951+0000 [id=466]        INFO        i.j.docker.DockerTransientNode$1#println: Disconnected computer for node 'java-docker-slave-00000y2c0>
Oct 19 10:21:45 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:21:45.954+0000 [id=39]        INFO        hudson.slaves.NodeProvisioner#update: Image of hariharan410/docker-slave:latest provisioning successfu>
Oct 19 10:21:45 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:21:45.954+0000 [id=466]        INFO        i.j.docker.DockerTransientNode$1#println: Removed Node for node 'java-docker-slave-00000y2c0uh6i'.
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.392+0000 [id=504]        INFO        hudson.model.AsyncPeriodicWork#lambda$doRun$1: Started DockerContainerWatchdog Asynchronous Periodic >
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.392+0000 [id=504]        INFO        c.n.j.p.d.DockerContainerWatchdog#execute: Docker Container Watchdog has been triggered
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.392+0000 [id=504]        INFO        c.n.j.p.d.DockerContainerWatchdog$Statistics#writeStatisticsToLog: Watchdog Statistics: Number of ove>
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.392+0000 [id=504]        INFO        c.n.j.p.d.DockerContainerWatchdog#loadNodeMap: We currently have 0 nodes assigned to this Jenkins ins>
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.392+0000 [id=504]        INFO        c.n.j.p.d.DockerContainerWatchdog#execute: Checking Docker Cloud docker at tcp://54.166.73.38:4243
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.400+0000 [id=504]        INFO        c.n.j.p.d.DockerContainerWatchdog#execute: Docker Container Watchdog check has been completed
Oct 19 10:26:30 ip-172-31-84-59 jenkins[5844]: 2022-10-19 10:26:30.400+0000 [id=504]        INFO        hudson.model.AsyncPeriodicWork#lambda$doRun$1: Finished DockerContainerWatchdog Asynchronous Periodic>
lines 1-19/19 (END)
```

Step 1.3 :: Install Docker in both Master and Slave machines

$ sudo apt install docker.io -qy



```
root@ip-172-31-84-59:~# docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
root@ip-172-31-84-59:~#
```

Step 1.4 :: Configure Docker in Slave Machine

$ sudo vi /lib/systemd/system/docker.service

Search for “ExecStart” and replace that line with the following.

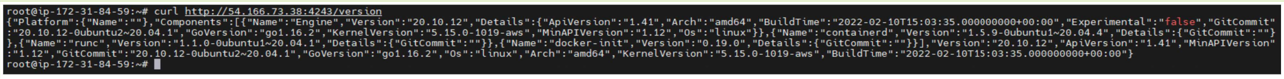
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock

$ sudo systemctl daemon-reload

$ sudo service docker restart

Validate API by executing the following curl command

curl http://<Slave.ip.address>:4243/version

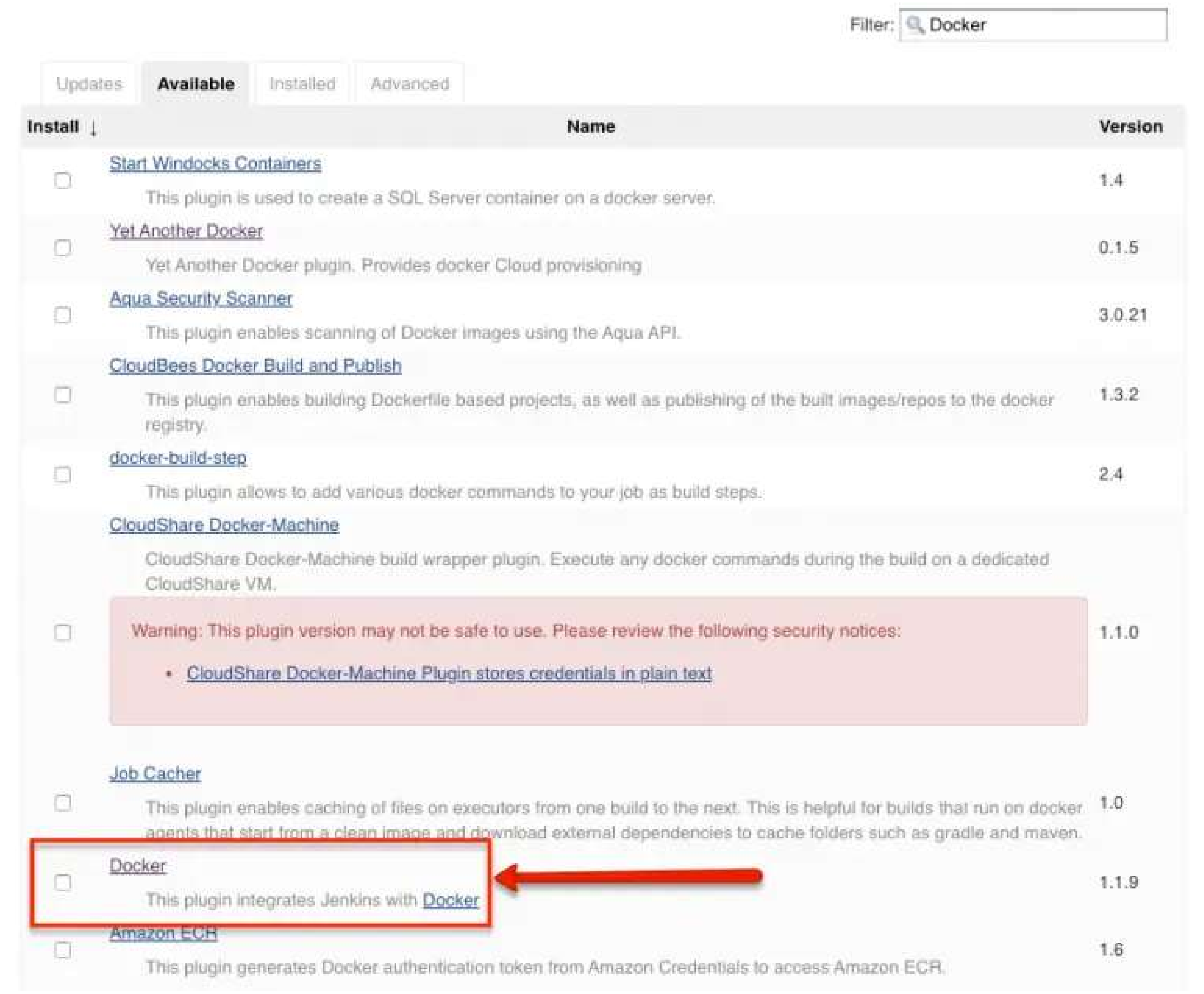


```
root@ip-172-31-84-59:~# curl http://54.166.73.38:4243/version
{"Platform":{"Name":""},"Components":[{"Name":"Engine","Version":"20.10.12","Details":{"ApiVersion":"1.41","Arch":"amd64","BuildTime":"2022-02-10T15:03:35.000000000+00:00","Experimental":"false","GitCommit"
:"20.10.12-0ubuntu2~20.04.1","GoVersion":"go1.16.2","KernelVersion":"5.15.0-1019-aws","MinAPIVersion":"1.12","Os":"linux"}},{"Name":"containerd","Version":"1.5.9-0ubuntu1~20.04.4","Details":{"GitCommit":""}
},{"Name":"runc","Version":"1.1.0-0ubuntu1~20.04.1","Details":{"GitCommit":""}},{"Name":"docker-init","Version":"0.19.0","Details":{"GitCommit":""}}],"Version":"20.10.12","ApiVersion":"1.41","MinAPIVersion"
:"1.12","GitCommit":"20.10.12-0ubuntu2~20.04.1","GoVersion":"go1.16.2","Os":"linux","Arch":"amd64","KernelVersion":"5.15.0-1019-aws","BuildTime":"2022-02-10T15:03:35.000000000+00:00"}
root@ip-172-31-84-59:~# 
```

Step 2 :: Configure Jenkins Server with Docker Plugin

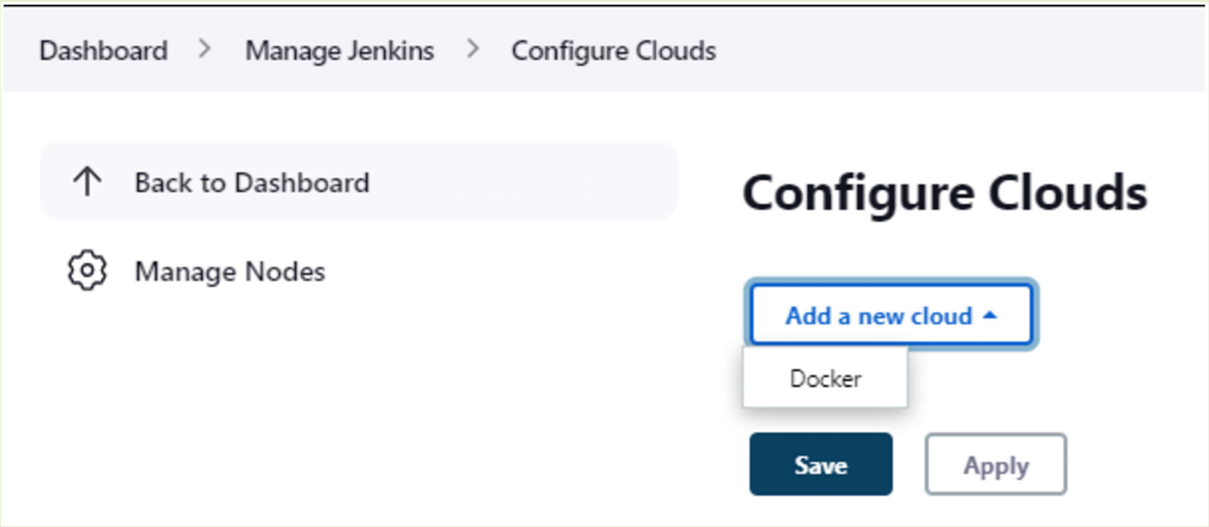
Step 2.1 :: Head over to Jenkins Dashboard –> Manage Jenkins –> Manage Plugins

Step 2.2 :: Under the Available tab, search for “Docker” and install the docker cloud plugin



Install ↓	Name	Version
□	Start Windocks Containers This plugin is used to create a SQL Server container on a docker server.	1.4
□	Yet Another Docker Yet Another Docker plugin. Provides docker Cloud provisioning	0.1.5
□	Aqua Security Scanner This plugin enables scanning of Docker images using the Aqua API.	3.0.21
□	CloudBees Docker Build and Publish This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.	1.3.2
□	docker-build-step This plugin allows to add various docker commands to your job as build steps.	2.4
□	CloudShare Docker-Machine CloudShare Docker-Machine build wrapper plugin. Execute any docker commands during the build on a dedicated CloudShare VM. Warning: This plugin version may not be safe to use. Please review the following security notices: • CloudShare Docker-Machine Plugin stores credentials in plain text	1.1.0
□	Job Cacher This plugin enables caching of files on executors from one build to the next. This is helpful for builds that run on docker agents that start from a clean image and download external dependencies to cache folders such as gradle and maven.	1.0
□	Docker This plugin integrates Jenkins with Docker	1.1.9
□	Amazon ECR This plugin generates Docker authentication token from Amazon Credentials to access Amazon ECR.	1.6

Step 2.3 :: Once installed head over to Dashboard -> Manage Jenkins -> Manage Nodes and Clouds -> Configure Clouds -> Add a new cloud -> Docker

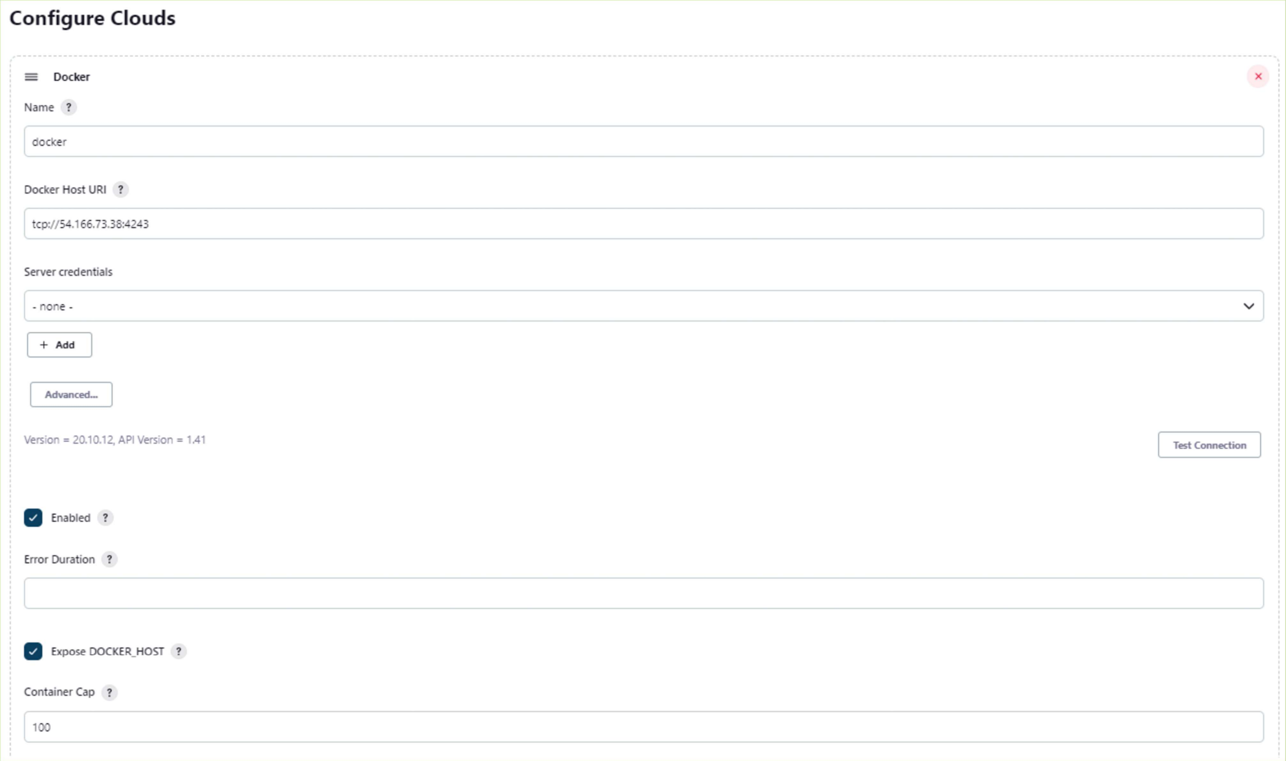


Step 2.4 :: Configure Cloud as below

Name : docker

Docker Host URI : tcp://<client.server.ip>:4243

Ensure to check <u>Enabled</u> and Expose <u>DOCKER_HOST</u> check boxes



Configure Clouds

Docker

Name: docker

Docker Host URI: tcp://54.166.73.38:4243

Server credentials: - none -

Version = 20.10.12, API Version = 1.41

☑ Enabled

Error Duration:

☑ Expose DOCKER_HOST

Container Cap: 100

Step 2.5 :: Configure Docker Agent Template

Label : java-docker-slave

Ensure to tick the <u>Enabled</u> check box

Docker Image : hariharan410/docker-slave:latest

Remote File System : /home/jenkins

Connect method : Connect with SSH

SSH credentials : Create a new one with user name and password as “jenkins”

Host Key Verification Strategy : Non verifying Verification Strategy

Click on Advanced and update JavaPath : /usr/bin/java

Click on Save

Note : Other values can be left as default

Step 3 :: Run a job

Step 3.1 :: Goto Dashboard -> New Item

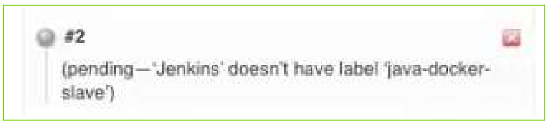
Step 3.2 :: Create a freestyle project

Step 3.3 :: Enable “Restrict where this project can be run” and enter value as “java-docker-slave”

Step 3.4 : Select Execute Shell under Build Steps and the below

echo "Hello World"

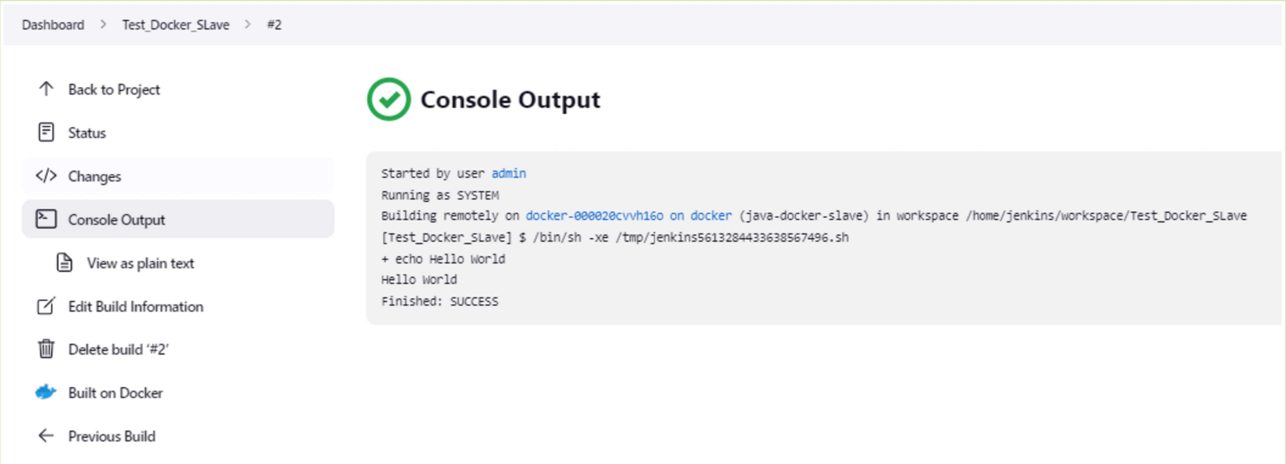
Save and Build the job (Wait for few mins... Initially we will get as below)



#2

(pending—'Jenkins' doesn't have label 'java-docker-slave')

Step 4 :: Result



Console Output

```
Started by user admin
Running as SYSTEM
Building remotely on docker-000020cvvh16o on docker (java-docker-slave) in workspace /home/jenkins/workspace/Test_Docker_SLave
[Test_Docker_SLave] $ /bin/sh -xe /tmp/jenkins5613284433638567496.sh
+ echo Hello World
Hello World
Finished: SUCCESS
```