

JENKINS MASTER SLAVE CONFIGURATION

This article shows how to setup slave nodes using Docker and integrate with Jenkins Master.

Pre-requisites:

- Jenkins Master is already setup and running
- port 8080 opened in Jenkins EC2's firewall rule
- Setup Docker host. Creating another EC2 instance is recommended to serve as a Docker Host.
- port 4243 opened in docker host machine
- 32768 - 60999 opened in docker host machine

▼ Inbound rules

Filter rules				
Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0ff7c8e6f46340e84	4243	TCP	0.0.0.0/0	launch-wizard-10
sgr-0afffb45bdc60fe4e	32768 - 60999	TCP	0.0.0.0/0	launch-wizard-10
sgr-0fa271db5531dfe5f	22	TCP	0.0.0.0/0	launch-wizard-10

Step 1 - Configure Docker Host with Remote API

Login to Docker host machine. Open docker service file. Search for ExecStart and replace that line with the following.

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

```
for containers run by docker
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock
ExecReload=/bin/kill -s HUP $MAINPID
TimeoutSec=0
RestartSec=2
Restart=always
```

You can replace with below line:

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

Restart Docker service

```
sudo systemctl daemon-reload
sudo service docker restart
```

Validate API by executing below curl command

curl http://localhost:4243/version

```
ubuntu@ip-172-31-15-25:~$ curl http://localhost:4243/version
{"Platform":{"Name":"","Components":[{"Name":"Engine","Version":"20.10.7","Version":"1.41","Arch":"amd64","BuildTime":"2021-10-22T00:57:37.000000000+00:00","Details":{"GitCommit":"20.10.7-0ubuntu5~18.04.3","GoVersion":"go1.13.8","Os":"linux","Arch":"amd64","BuildTime":"2021-10-22T00:57:37.000000000+00:00"}},{"Name":"containerd","Version":"1.5.5-0ubuntu3~18.04.2","Details":{"GitCommit":"","GoVersion":"go1.13.8","Os":"linux","Arch":"amd64","BuildTime":"2021-10-22T00:57:37.000000000+00:00"}},{"Name":"runc","Version":"1.0.1-0ubuntu2~18.04.1","Details":{"GitCommit":"","GoVersion":"go1.13.8","Os":"linux","Arch":"amd64","BuildTime":"2021-10-22T00:57:37.000000000+00:00"}},{"Name":"docker-init","Version":"0.19.0-0ubuntu1~18.04.1","Details":{"GitCommit":"","GoVersion":"go1.13.8","Os":"linux","Arch":"amd64","BuildTime":"2021-10-22T00:57:37.000000000+00:00"}]}]}},"Version":"20.10.7","ApiVersion":"1.41","MinAPIVersion":"1.41","GitCommit":"20.10.7-0ubuntu5~18.04.3","GoVersion":"go1.13.8","Os":"linux","Arch":"amd64","BuildTime":"2021-10-22T00:57:37.000000000+00:00"}
```

Step 2 - Build Jenkins slave Docker image

Download Dockerfile from below repo.

git clone <https://github.com/sushant-technocirrus/jenkins-slave.git> ;
cd jenkins-docker-slave

Build Docker image

sudo docker build -t my-jenkins-slave .

```
ubuntu@ip-172-31-15-25:~$ sudo docker build -t my-jenkins-slave .
Sending build context to Docker daemon 303.2MB
Step 1/7 : FROM ubuntu:18.04
--> 8d5df41c547b
Step 2/7 : LABEL maintainer="devops.coaching@gmail.com"
--> Using cache
--> 1f1227974fcb
Step 3/7 : RUN apt-get update && apt-get install -qy git && apt-get install -qy openssh-server && sed -i 's|session required pam_loginuid.so|session optional pam_loginuid.so|' /etc/pam.d/ssh && mkdir -p /var/run/ssh && apt-get install -qy default-jdk && apt-get install -qy maven && apt-get -qy autoremove && adduser --quiet jenkins && echo "jenkins:password" | chpasswd && mkdir /home/jenkins/.m2
--> Using cache
--> a1a2834b1ea0
Step 4/7 : COPY .ssh/authorized_keys /home/jenkins/.ssh/authorized_keys
--> Using cache
--> 5c9ff7780454c
Step 5/7 : RUN chown -R jenkins:jenkins /home/jenkins/.m2/ && chown -R jenkins:jenkins /home/jenkins/.ssh/
--> Using cache
--> 666c13a5bcc5
Step 6/7 : EXPOSE 22
--> Using cache
--> c10e4f7653d1
Step 7/7 : CMD ["/usr/sbin/sshd", "-D"]
--> Using cache
--> d2ae67fef47c
Successfully built d2ae67fef47c
Successfully tagged my-jenkins-slave:latest
```

- Perform below command to see the list of docker images:

sudo docker images

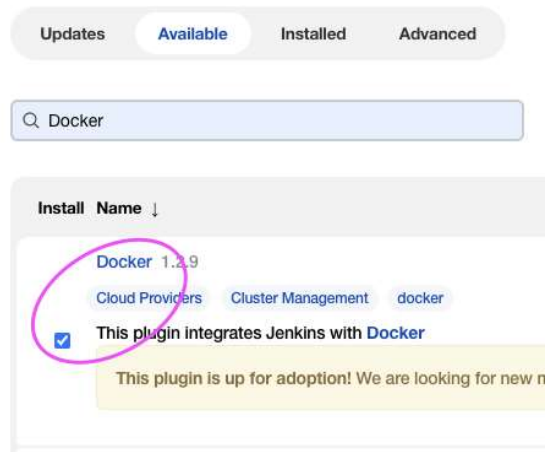
```
ubuntu@ip-172-31-15-25:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-jenkins-slave	latest	d2ae67fef47c	40 minutes ago	857MB
ubuntu	18.04	8d5df41c547b	10 days ago	63.1MB

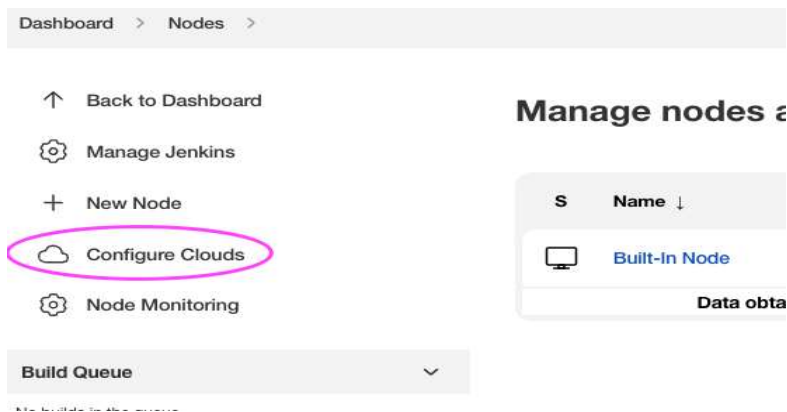
Step 3 - Configure Jenkins Server with Docker plug-in

- Now login to Jenkins Master. Make sure you install Docker plug-in in Jenkins.

Plugin Manager

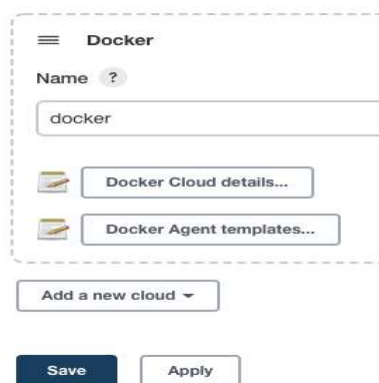


- Now go to Manage Jenkins -> Configure Nodes Cloud



- Click on Docker Cloud Details

Configure Clouds



Docker

Name ?

docker

Docker Cloud details...

Docker Agent templates...

Add a new cloud ▾

Save Apply

Enter docker host dns name or ip address

tcp://:<docker_host_dns>:4243

Make sure **Enabled** is selected

Now click on **Test Connection** to make sure connecting with docker host is working.

Configure Clouds



Docker

Name ?

docker

Docker Host URI ?

tcp://ec2-44-195-48-101.compute-1.amazonaws.com:4243

Server credentials

- none - ▾

+ Add

Advanced...

Version = 20.10.7, API Version = 1.41

Test Connection

☒ Enabled ?

Error Duration ?

Default = 300

Save Apply

Step 4 - Configure Docker Agent Templates

Now click on Docker Agent templates:

Enter label as "docker-slave" and give some name

Click on **Enabled**

Now enter the name of the docker image you have built previously in docker host.
enter **/home/jenkins** as Remote file system root

Docker Agent templates

List of Images to be launched as agents

Docker Agent templates

Labels ?

docker-slave

☒ Enabled ?

Name ?

java-docker-slave

Docker Image ?

my-jenkins-slave

Registry Authentication...

Container settings...

Instance Capacity ?

Remote File System Root ?

/home/jenkins

Usage ?

Use this node as much as possible

Choose Connect with SSH as connection method:

Enter SSH credentials per your Dockerfile - **jenkins/password**

Idle timeout ?

10

Connect method ?

Connect with SSH

→ Prerequisites:

- The docker container's mapped SSH port, typically a port on the docker host, has to be accessible over network from the master.
- Docker image must have **sshd** installed.
- Docker image must have **Java** installed.
- Log in details configured as per **ssh-slaves** plugin.

SSH key ?

Use configured SSH credentials

SSH Credentials

jenkins/***** (updated ssh password)

+ Add

Host Key Verification Strategy

Non verifying Verification Strategy

Advanced...

Stop timeout ?

10

choose **Never Pull** as pull strategy as we have already image stored in DockerHost.

☐ Remove volumes ?

Pull strategy ?

Never pull

Pull timeout ?

300

Node Properties

Add Node Property ▾

Click on Save.

Step 5 - Create build job in Jenkins

Now Create a pipeline job in Jenkins with below pipeline code:

```
pipeline {  
  agent {  
    label "docker-slave"  
  }  
  stages {  
    stage('Hello') {  
      steps {  
        echo 'Hello World'  
      }  
    }  
  }  
}
```

Click Apply and Save.

Now build the job. Now you will see output like below:

The screenshot shows the Jenkins web interface. At the top, there's a header with the Jenkins logo, a search bar, and user information (sushant.karanjekar). Below the header, the breadcrumb trail reads "Dashboard > Jenkins Job 1 > #4". On the left sidebar, the "Console Output" tab is selected. The main area displays the console output for the build, which includes the following text:

```
Started by user sushant.karanjekar  
[Pipeline] Start of Pipeline  
[Pipeline] node  
Running on docker-slave-00001ao7ovvad on docker in /home/jenkins/workspace/Jenkins Job 1  
[Pipeline] {  
[Pipeline] stage  
[Pipeline] { (Hello)  
[Pipeline] echo  
Hello World  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] End of Pipeline  
Finished: SUCCESS
```