INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Dharwad

ज्ञानेन विकासः

DevOps and its Applications

CS457

Assignment-2

Jenkins

Under the Guidance of – Dr Uma S

Submitted By:

1. Aqtar parveez (18BCS010)
2. G K Bharath Bhushan (18BCS026)
3. K V S Chaitanya (18BCS045)
4. Neha T (18BCS060)
5. Rahul S (18BCS075)
6. Varun Awati (18BCS108)
7. Sushanth B Patil (18BCS102)
8. Trishul K S (18BCS104)

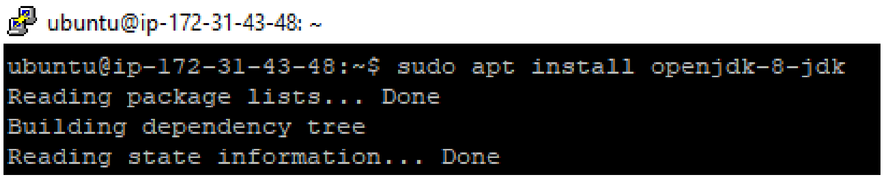
Jenkins Master- Slave pipeline

Requirements:

- Git
- Docker
- Jenkins
- AWZ EC2 Instances (3)

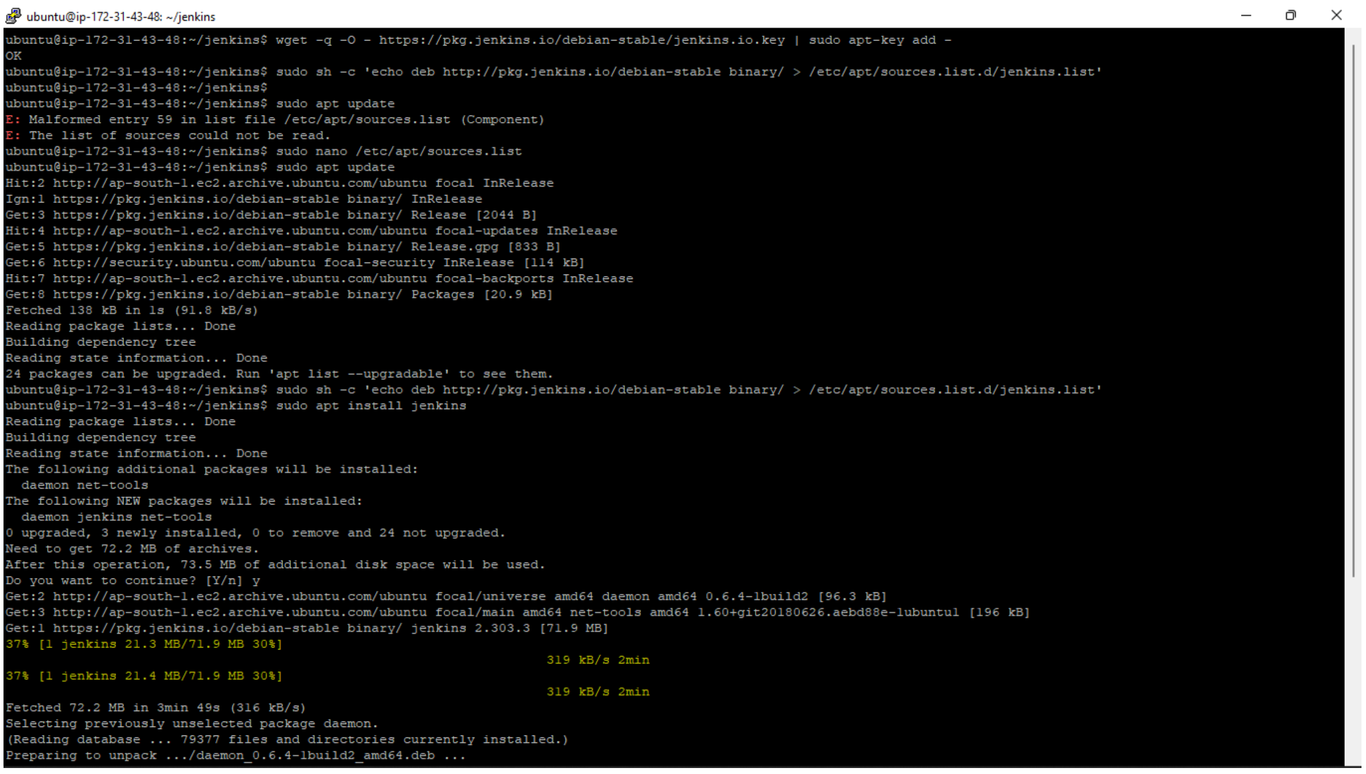
Step-1: Install Jenkins on an EC2 instance with the commands mentioned below in the screenshot.

Install Jdk (pre requisite)



ubuntu@ip-172-31-43-48: ~

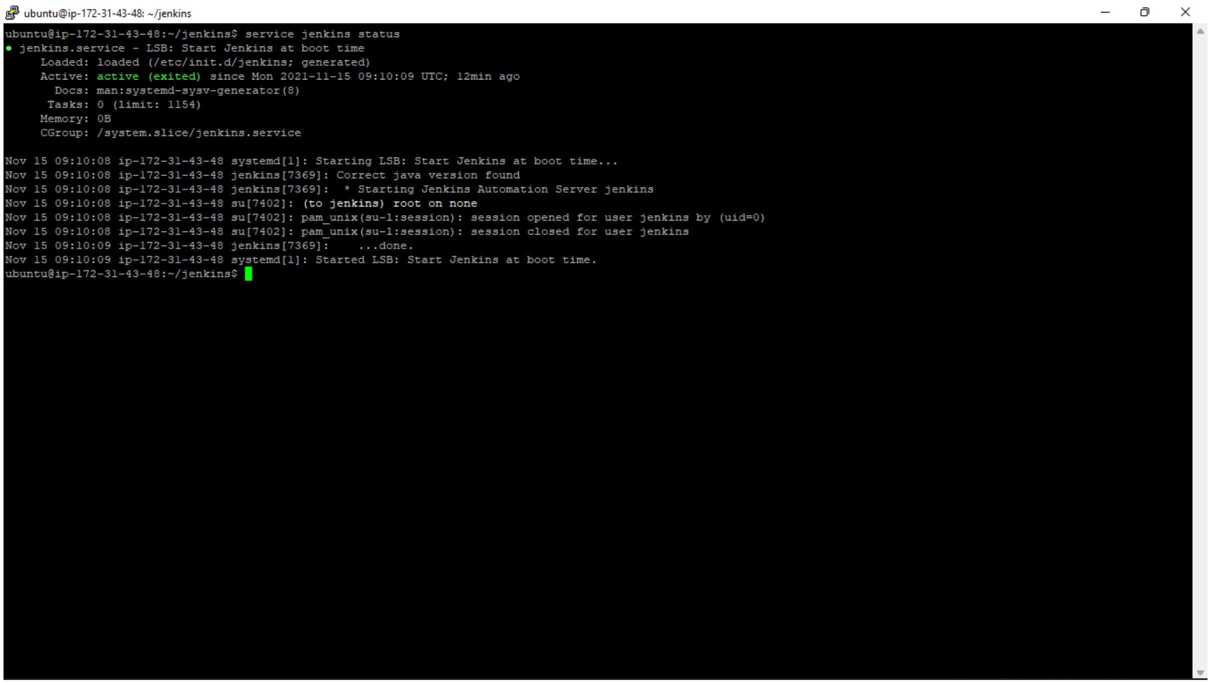
```
ubuntu@ip-172-31-43-48:~$ sudo apt install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
```



ubuntu@ip-172-31-43-48: ~/jenkins

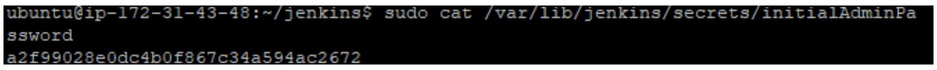
```
ubuntu@ip-172-31-43-48:~/jenkins$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
OK
ubuntu@ip-172-31-43-48:~/jenkins$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ip-172-31-43-48:~/jenkins$
ubuntu@ip-172-31-43-48:~/jenkins$ sudo apt update
E: Malformed entry 59 in list file /etc/apt/sources.list (Component)
E: The list of sources could not be read.
ubuntu@ip-172-31-43-48:~/jenkins$ sudo nano /etc/apt/sources.list
ubuntu@ip-172-31-43-48:~/jenkins$ sudo apt update
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Hit:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [20.9 kB]
Fetched 138 kB in 1s (91.8 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
24 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-43-48:~/jenkins$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ip-172-31-43-48:~/jenkins$ sudo apt install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon net-tools
The following NEW packages will be installed:
  daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 24 not upgraded.
Need to get 72.2 MB of archives.
After this operation, 73.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 daemon amd64 0.6.4-1build2 [96.3 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.303.3 [71.9 MB]
37% [1 jenkins 21.3 MB/71.9 MB 30%]
                                                                    319 kB/s 2min
37% [1 jenkins 21.4 MB/71.9 MB 30%]
                                                                    319 kB/s 2min
Fetched 72.2 MB in 3min 49s (316 kB/s)
Selecting previously unselected package daemon.
(Reading database ... 79377 files and directories currently installed.)
Preparing to unpack .../daemon_0.6.4-1build2_amd64.deb ...
```

Step-2: After installation, we can see the status of the Jenkins service. It should be “active” as mentioned in the screenshot below.

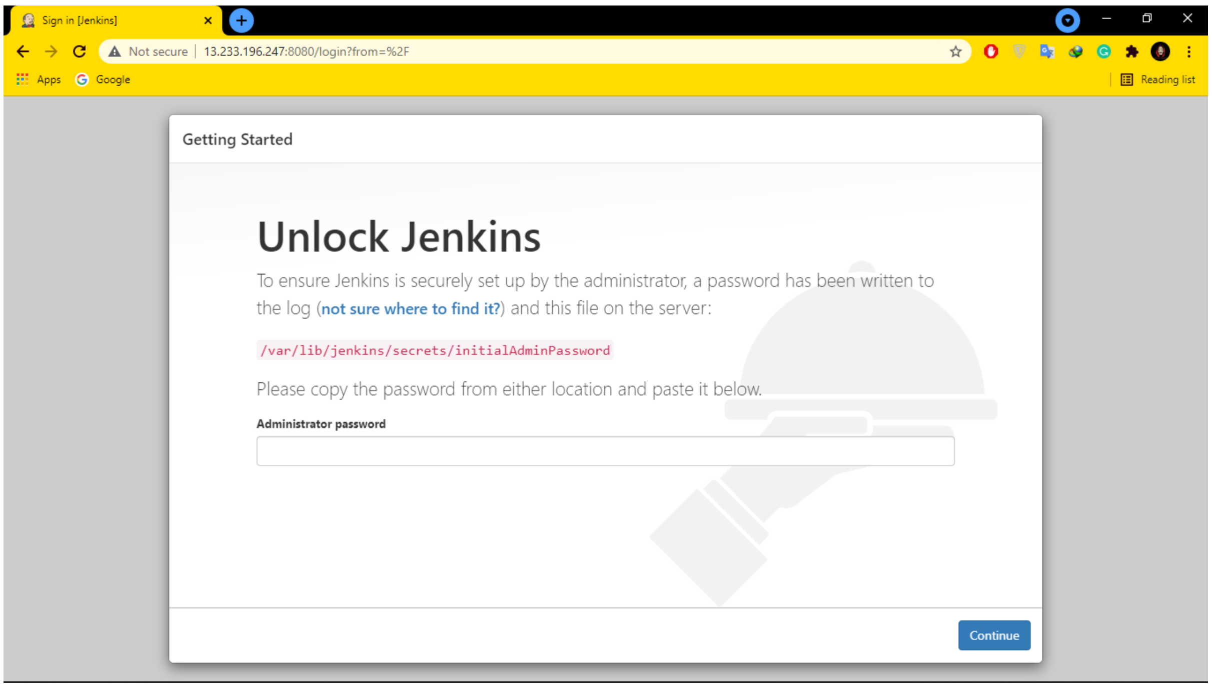

ubuntu@ip-172-31-43-48: ~/jenkins

```
ubuntu@ip-172-31-43-48:~/jenkins$ service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Mon 2021-11-15 09:10:09 UTC; 12min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 0 (limit: 1154)
     Memory: 0B
     CGroup: /system.slice/jenkins.service

Nov 15 09:10:08 ip-172-31-43-48 systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 15 09:10:08 ip-172-31-43-48 jenkins[7369]: Correct java version found
Nov 15 09:10:08 ip-172-31-43-48 jenkins[7369]:  * Starting Jenkins Automation Server jenkins
Nov 15 09:10:08 ip-172-31-43-48 su[7402]: (to jenkins) root on none
Nov 15 09:10:08 ip-172-31-43-48 su[7402]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Nov 15 09:10:08 ip-172-31-43-48 su[7402]: pam_unix(su-l:session): session closed for user jenkins
Nov 15 09:10:09 ip-172-31-43-48 jenkins[7369]:    ...done.
Nov 15 09:10:09 ip-172-31-43-48 systemd[1]: Started LSB: Start Jenkins at boot time.
ubuntu@ip-172-31-43-48:~/jenkins$
```

Step-3:Now enter the public IP address of the EC2 instance followed by port number 8080 (by default, Jenkins runs on port 8080) in a browser to access the Jenkins Dashboard. Here, it will prompt to enter the Admin Password to unlock Jenkins. The Admin Password can be found in the file whose absolute path will be mentioned in the prompt. Read the content of the file as mentioned in the screenshot below, and enter that in the Admin Password field.



```
ubuntu@ip-172-31-43-48:~/jenkins$ sudo cat /var/lib/jenkins/secrets/initialAdminPa
ssword
a2f99028e0dc4b0f867c34a594ac2672
```



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

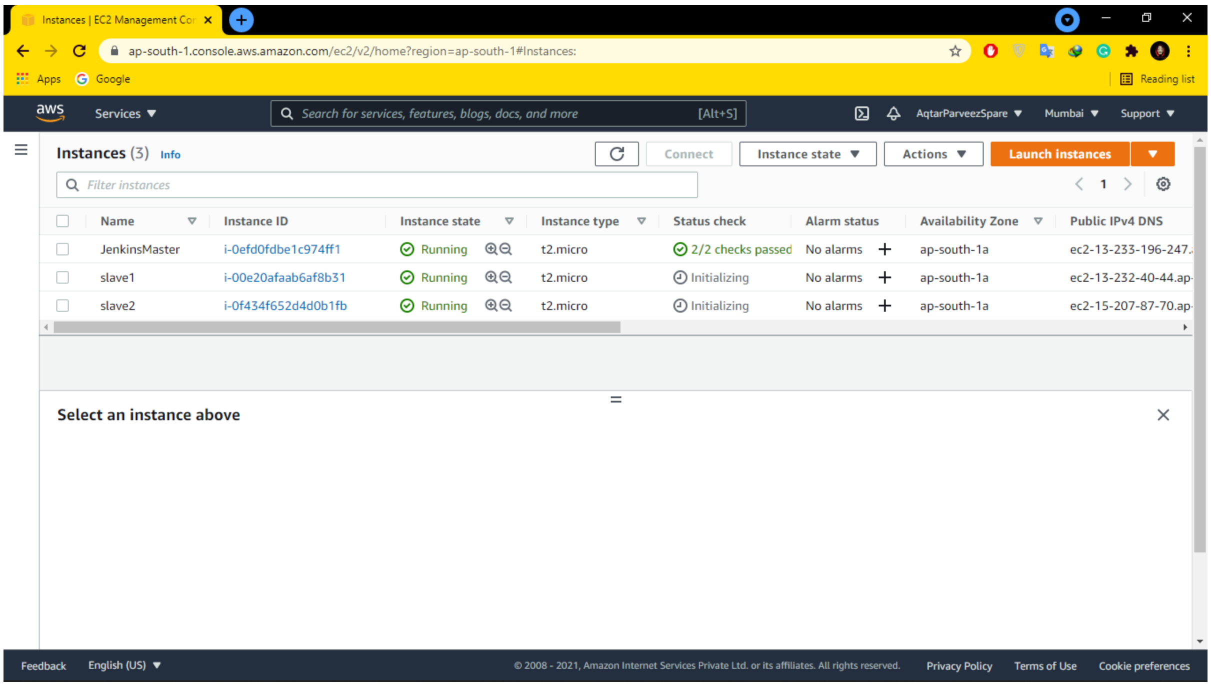
Please copy the password from either location and paste it below.

Administrator password

Step-4: After entering the correct password, Jenkins will prompt to create a new user by entering username, password, and email. Enter the same and login.

Jenkins is now ready to use.

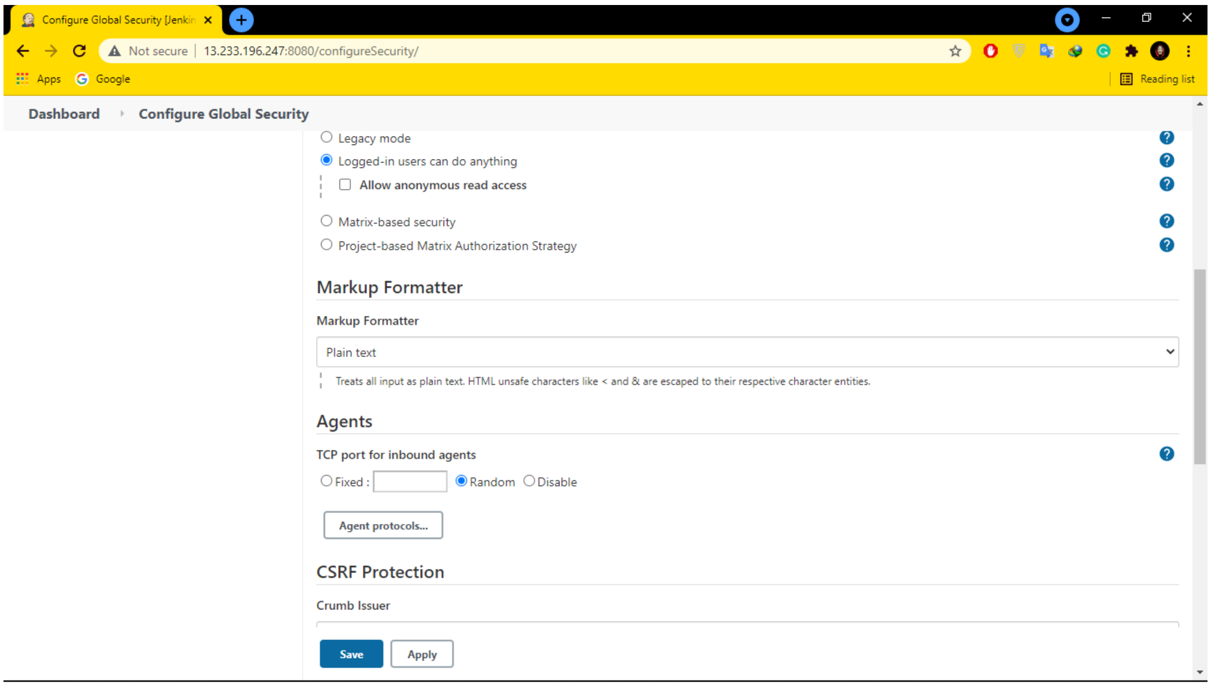
Step-5: Name the other two EC2 instances as ‘slave1’ and ‘slave2’ to be set-up as test and production servers. Then, from the Jenkins Dashboard, go to Manage Jenkins ⇒ Configure global security. In ‘Agents’, change the ‘TCP port for inbound agents’ to ‘Random’, and click ‘Save’


Instances | EC2 Management Co
ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:
Services
Search for services, features, blogs, docs, and more
AqtarParveezSpare
Mumbai
Support
Connect
Instance state
Actions
Launch instances
Filter instances
Feedback
English (US)
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Instances (3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
JenkinsMaster	i-0efd0fdbe1c974ff1	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-233-196-247.
slave1	i-00e20afaab6af8b31	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-13-232-40-44.ap
slave2	i-0f434f652d4d0b1fb	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-15-207-87-70.ap

Select an instance above


Configure Global Security [Jenkin
Not secure | 13.233.196.247:8080/configureSecurity/
Dashboard
Configure Global Security
Agent protocols...
Save
Apply

○ Legacy mode
● Logged-in users can do anything
☐ Allow anonymous read access
○ Matrix-based security
○ Project-based Matrix Authorization Strategy

Markup Formatter

Markup Formatter: Plain text

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

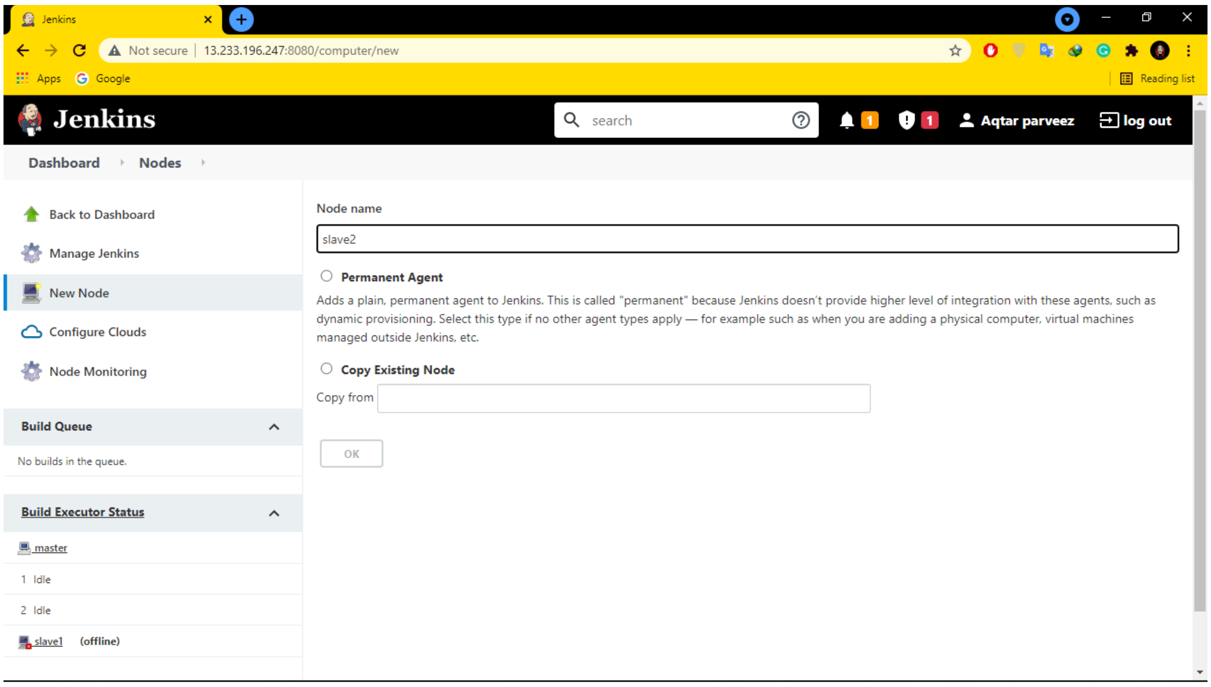
Agents

TCP port for inbound agents: ○ Fixed : ● Random ○ Disable

CSRF Protection

Crumb Issuer

Step-6: Add slave1 and slave2 nodes to Jenkins master. Go to Manage Jenkins ⇒ Manage Nodes and Clouds ⇒ New Node. Now enter the name as slave1/slave2 and enable ‘Permanent Agent’. Then click ‘OK’.


Jenkins

Not secure | 13.233.196.247:8080/computer/new

Dashboard › Nodes ›

Back to Dashboard
Manage Jenkins
New Node
Configure Clouds
Node Monitoring

Build Queue
No builds in the queue.

Build Executor Status
master
1 Idle
2 Idle
slave1 (offline)

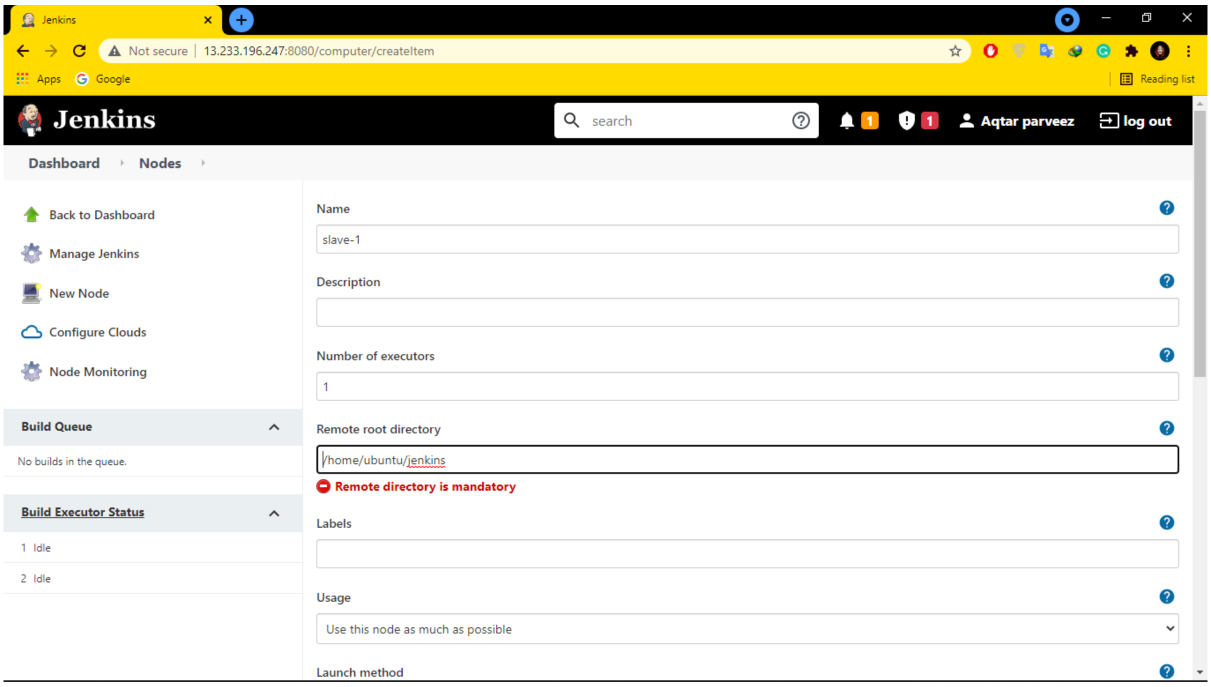
Node name
slave2

Permanent Agent
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Copy Existing Node
Copy from

OK

Step-7: configure the created Nodes and enter ‘/home/ubuntu/jenkins’ in the ‘Remote root directory’ as mentioned in the screenshot below and click Save.


Jenkins

Not secure | 13.233.196.247:8080/computer/createItem

Dashboard › Nodes ›

Back to Dashboard
Manage Jenkins
New Node
Configure Clouds
Node Monitoring

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Name
slave-1

Description

Number of executors
1

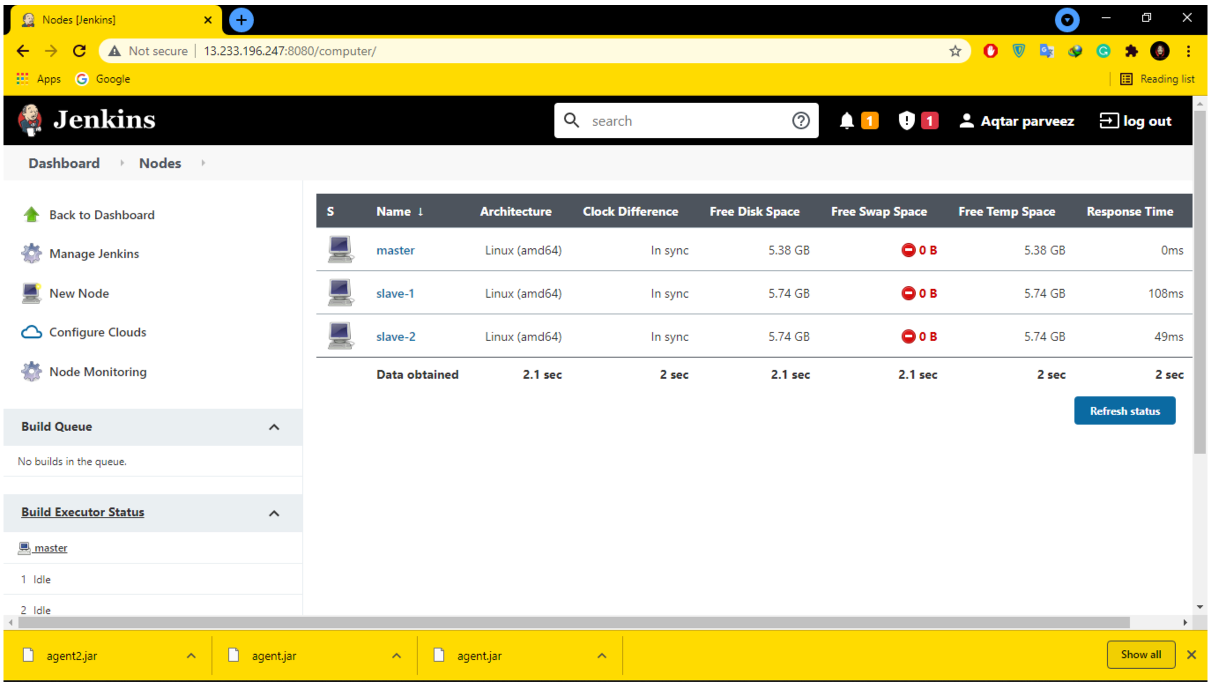
Remote root directory
/home/ubuntu/jenkins
Remote directory is mandatory

Labels

Usage
Use this node as much as possible

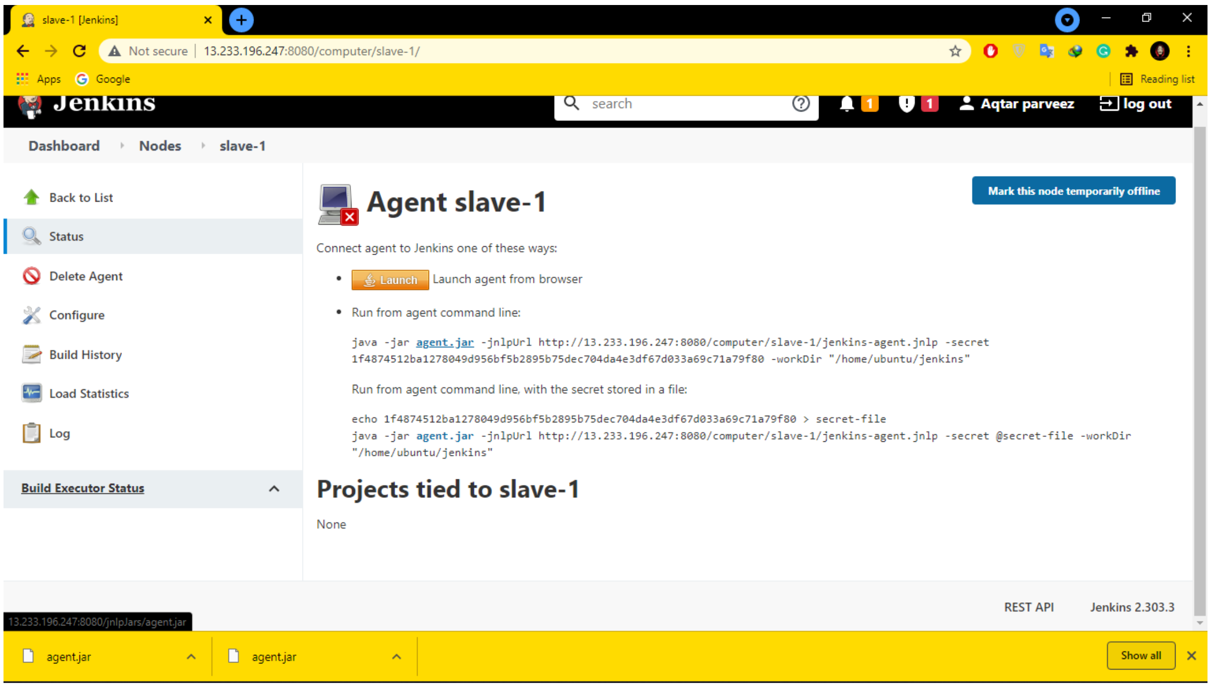
Launch method

After Saving the Node Management looks like this



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.38 GB	0 B	5.38 GB	0ms
	slave-1	Linux (amd64)	In sync	5.74 GB	0 B	5.74 GB	108ms
	slave-2	Linux (amd64)	In sync	5.74 GB	0 B	5.74 GB	49ms
	Data obtained	2.1 sec	2 sec	2.1 sec	2.1 sec	2 sec	2 sec

Step-8: Go to slave1 and download the agent.jar file. Now open FileZilla application and connect it to slave1 EC2 node. Then transfer the agent.jar file to the node using SFTP.



Agent slave-1

Connect agent to Jenkins one of these ways:

- Launch agent from browser
- Run from agent command line:

```
java -jar agent.jar -jnlpUrl http://13.233.196.247:8080/computer/slave-1/jenkins-agent.jnlp -secret
1f4874512ba1278049d956bf5b2895b75dec704da4e3df67d033a69c71a79f80 -workDir "/home/ubuntu/jenkins"
```

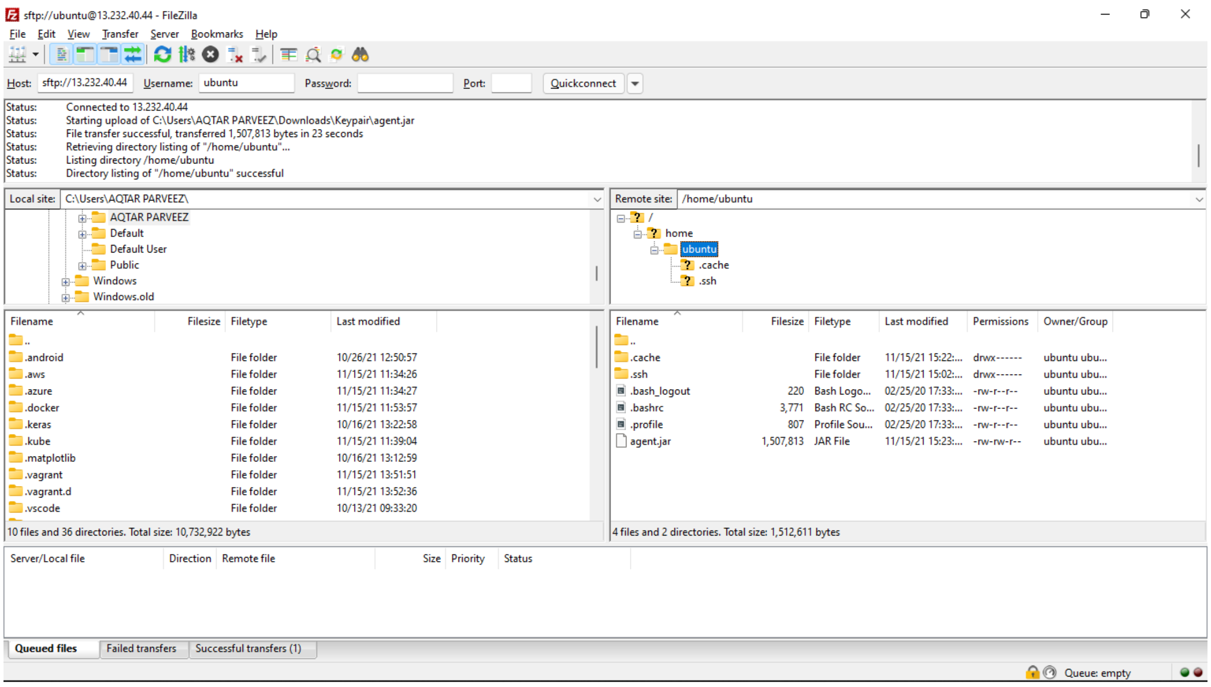
Run from agent command line, with the secret stored in a file:

```
echo 1f4874512ba1278049d956bf5b2895b75dec704da4e3df67d033a69c71a79f80 > secret-file
java -jar agent.jar -jnlpUrl http://13.233.196.247:8080/computer/slave-1/jenkins-agent.jnlp -secret @secret-file -workDir
"/home/ubuntu/jenkins"
```

Projects tied to slave-1

None

Find the downloaded agent.jar file and drag it into the ubuntu folder of the Slave 1 instance.


sftp://ubuntu@13.232.40.44 - FileZilla
File Edit View Transfer Server Bookmarks Help
Host: sftp://13.232.40.44 Username: ubuntu Password: Port: Quickconnect

```
Status:    Connected to 13.232.40.44
Status:    Starting upload of C:\Users\AQTAR PARVEEZ\Downloads\Keypair\agent.jar
Status:    File transfer successful, transferred 1,507,813 bytes in 23 seconds
Status:    Retrieving directory listing of "/home/ubuntu"...
Status:    Listing directory /home/ubuntu
Status:    Directory listing of "/home/ubuntu" successful
```

Local site: C:\Users\AQTAR PARVEEZ\

Filename	Filesize	Filetype	Last modified
..			
.android		File folder	10/26/21 12:50:57
.aws		File folder	11/15/21 11:34:26
.azure		File folder	11/15/21 11:34:27
.docker		File folder	11/15/21 11:53:57
.keras		File folder	10/16/21 13:22:58
.kube		File folder	11/15/21 11:39:04
.matplotlib		File folder	10/16/21 13:12:59
.vagrant		File folder	11/15/21 13:51:51
.vagrant.d		File folder	11/15/21 13:52:36
.vscode		File folder	10/13/21 09:33:20

10 files and 36 directories. Total size: 10,732,922 bytes

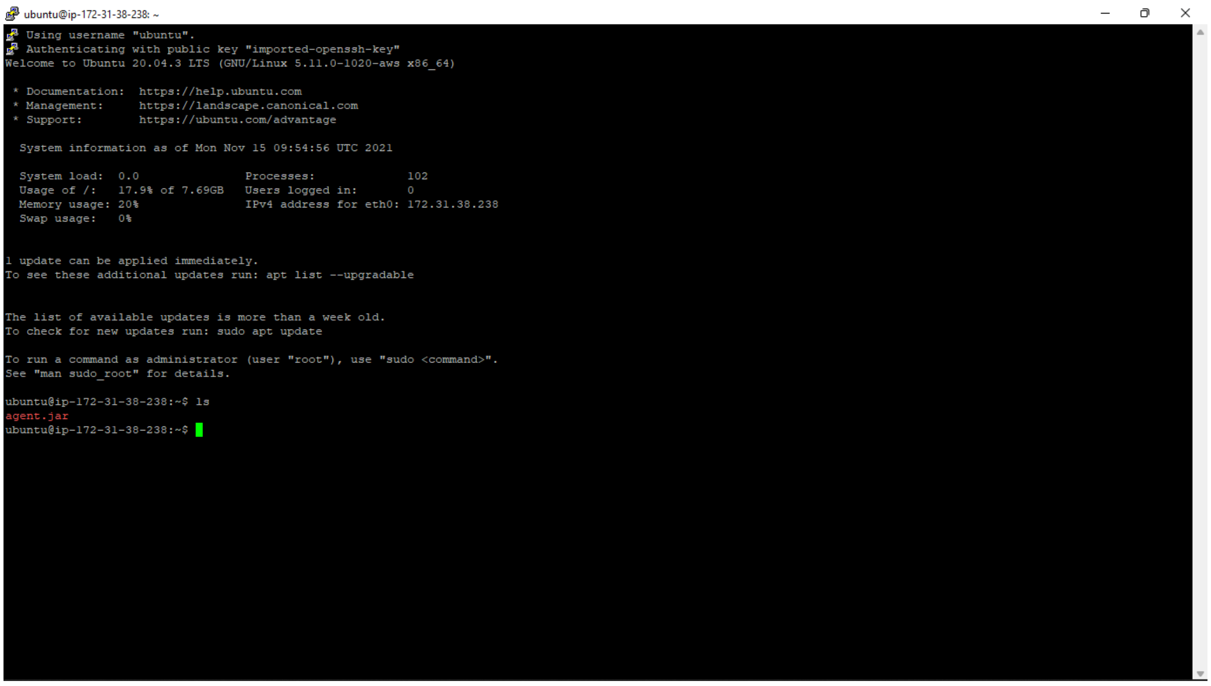
Remote site: /home/ubuntu

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
.cache		File folder	11/15/21 15:22:...	drwx------	ubuntu ubu...
.ssh		File folder	11/15/21 15:02:...	drwx------	ubuntu ubu...
.bash_logout	220	Bash Logo...	02/25/20 17:33:...	-rw-r--r--	ubuntu ubu...
.bashrc	3,771	Bash RC So...	02/25/20 17:33:...	-rw-r--r--	ubuntu ubu...
.profile	807	Profile Sou...	02/25/20 17:33:...	-rw-r--r--	ubuntu ubu...
agent.jar	1,507,813	JAR File	11/15/21 15:23:...	-rw-rw-r--	ubuntu ubu...

4 files and 2 directories. Total size: 1,512,611 bytes

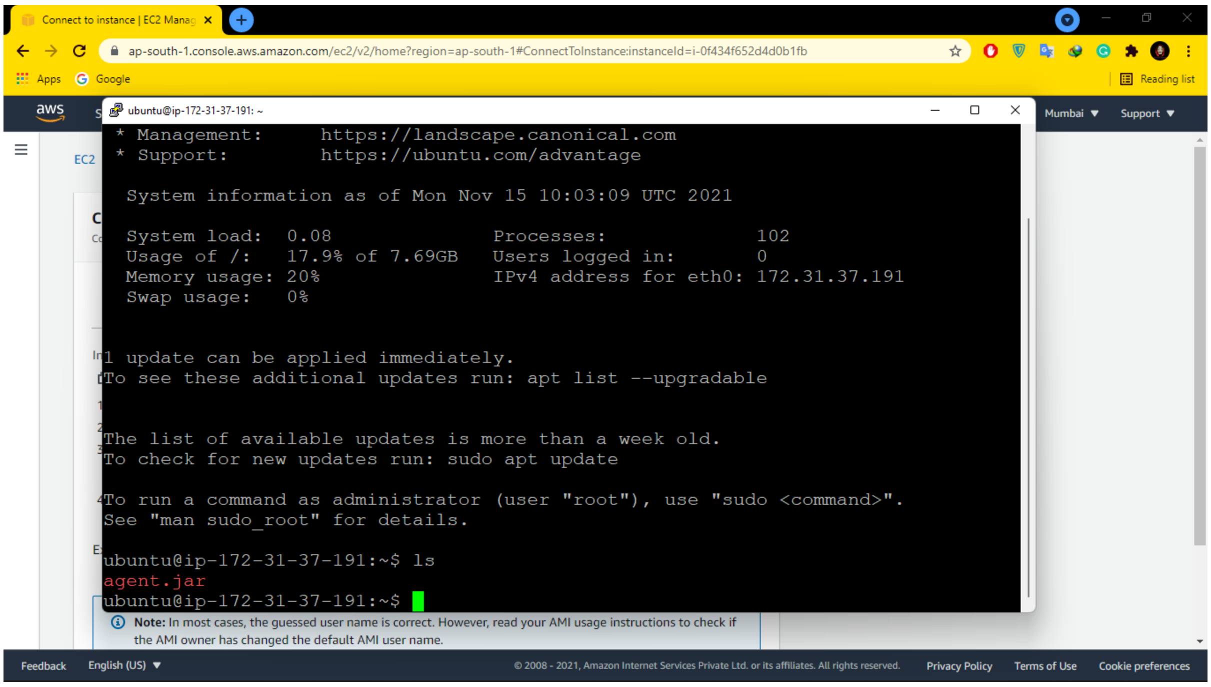
Do this process for both nodes i.e slave 1 and slave 2

Slave 1



```
 Using username "ubuntu".
 Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 15 09:54:56 UTC 2021

  System load:  0.0               Processes:             102
  Usage of /:   17.9% of 7.69GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.38.238
  Swap usage:   0%


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-38-238:~$ ls
agent.jar
ubuntu@ip-172-31-38-238:~$
```

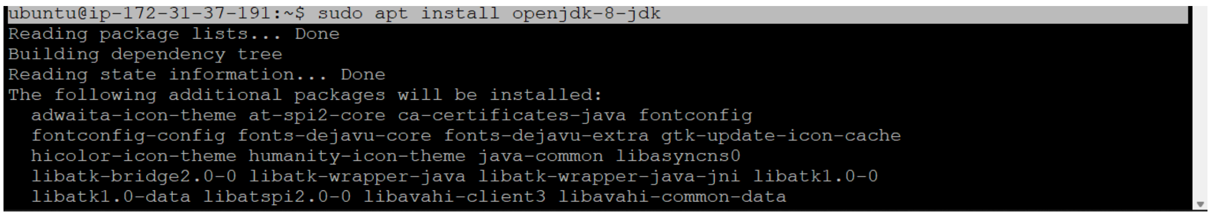
Slave 2



```
ubuntu@ip-172-31-37-191: ~
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 15 10:03:09 UTC 2021

  System load:  0.08              Processes:             102
  Usage of /:   17.9% of 7.69GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.37.191
  Swap usage:   0%


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-37-191:~$ ls
agent.jar
ubuntu@ip-172-31-37-191:~$
```

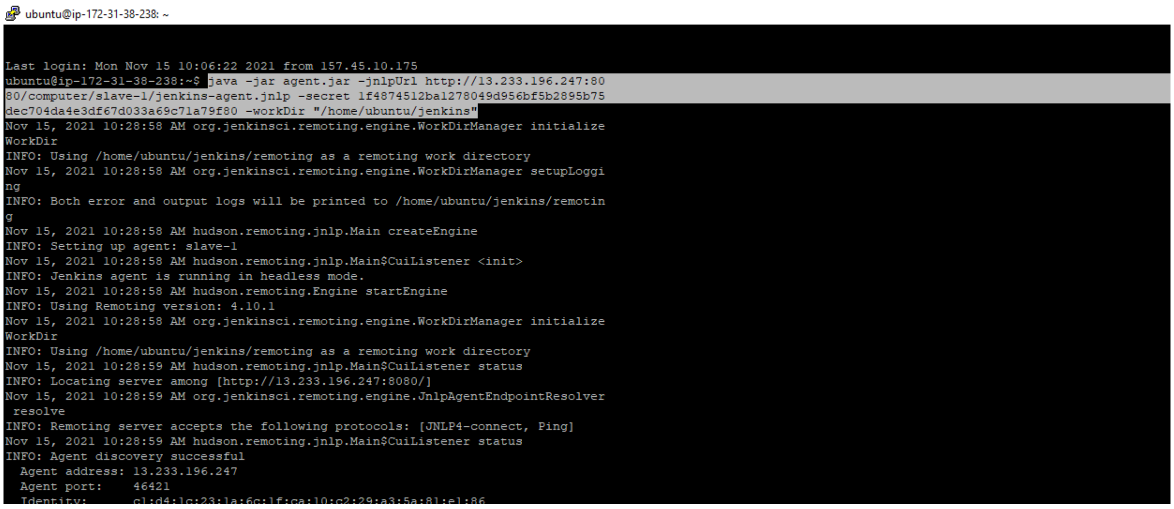
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Step-9: Install JDK on both slave nodes



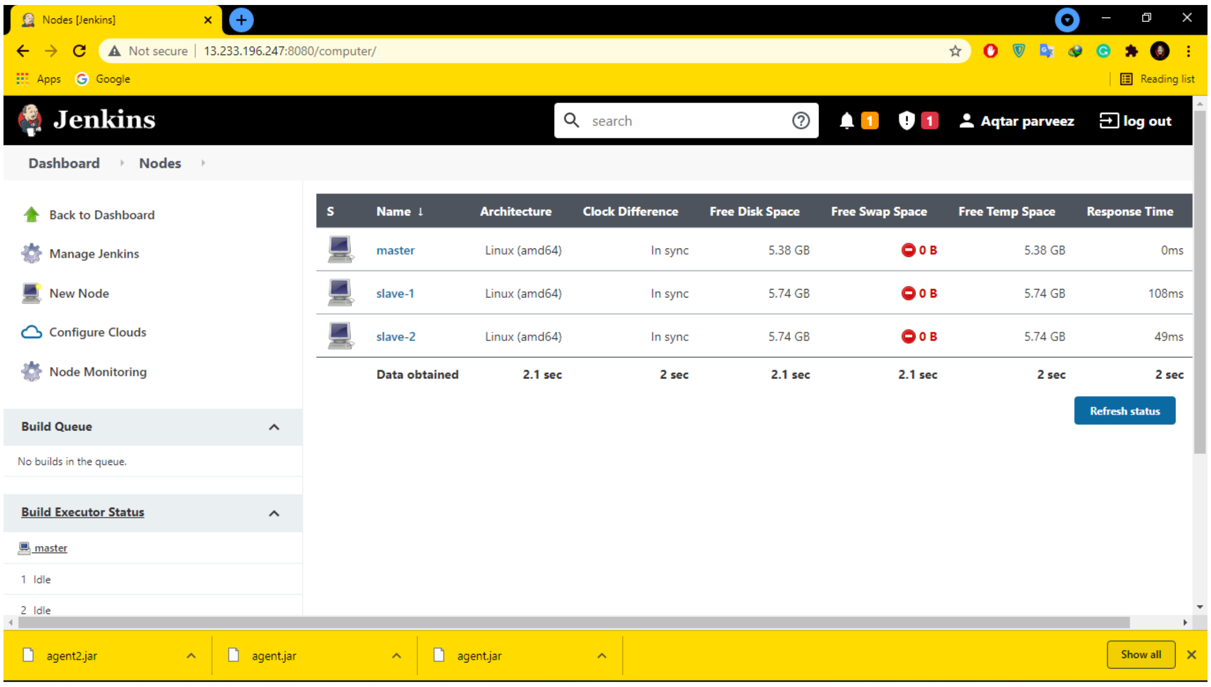
```
ubuntu@ip-172-31-37-191:~$ sudo apt install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme at-spi2-core ca-certificates-java fontconfig
  fontconfig-config fonts-dejavu-core fonts-dejavu-extra gtk-update-icon-cache
  hicolor-icon-theme humanity-icon-theme java-common libasyncns0
  libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0
  libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data
```

Step-10: Now copy the command from the slave nodes on Jenkins, and execute them on the EC2 instances as shown below, This should be done for both the nodes.



```
ubuntu@ip-172-31-38-238: ~

Last login: Mon Nov 15 10:06:22 2021 from 157.45.10.175
ubuntu@ip-172-31-38-238:~$ java -jar agent.jar -jnlpUrl http://13.233.196.247:80
80/computer/slave-1/jenkins-agent.jnlp -secret 1f4874512ba1278049d956bf5b2895b75
dec704da4e3df67d033a69c71a79f80 -workDir "/home/ubuntu/jenkins"
Nov 15, 2021 10:28:58 AM org.jenkinsci.remoting.engine.WorkDirManager initialize
WorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Nov 15, 2021 10:28:58 AM org.jenkinsci.remoting.engine.WorkDirManager setupLoggi
ng
INFO: Both error and output logs will be printed to /home/ubuntu/jenkins/remotin
g
Nov 15, 2021 10:28:58 AM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: slave-1
Nov 15, 2021 10:28:58 AM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Nov 15, 2021 10:28:58 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10.1
Nov 15, 2021 10:28:58 AM org.jenkinsci.remoting.engine.WorkDirManager initialize
WorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Nov 15, 2021 10:28:59 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://13.233.196.247:8080/]
Nov 15, 2021 10:28:59 AM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver
 resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Nov 15, 2021 10:28:59 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: 13.233.196.247
  Agent port:    46421
  Identity:      c1:d4:1c:23:1a:6c:1f:ca:10:c2:29:a3:5a:81:e1:86
```

Both the slave nodes will now be in sync with Jenkins master


Nodes [Jenkins]

Not secure | 13.233.196.247:8080/computer/

Apps   Google   Reading list

Jenkins   search   1   1   Aqtar parveez   log out

Dashboard › Nodes ›

- Back to Dashboard
- Manage Jenkins
- New Node
- Configure Clouds
- Node Monitoring

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.38 GB	0 B	5.38 GB	0ms
	slave-1	Linux (amd64)	In sync	5.74 GB	0 B	5.74 GB	108ms
	slave-2	Linux (amd64)	In sync	5.74 GB	0 B	5.74 GB	49ms
	Data obtained	2.1 sec	2 sec	2.1 sec	2.1 sec	2 sec	2 sec

Refresh status

Build Queue

No builds in the queue.

Build Executor Status

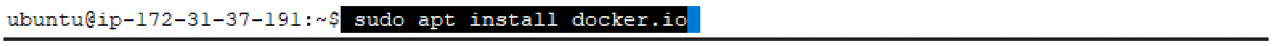
master

1 Idle

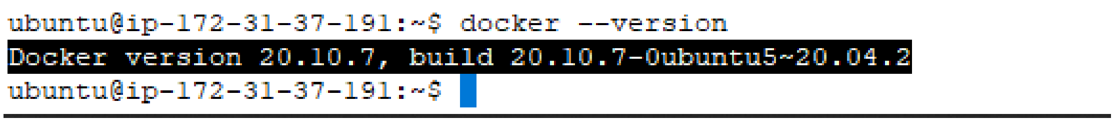
2 Idle

agent2.jar   agent.jar   agent.jar   Show all

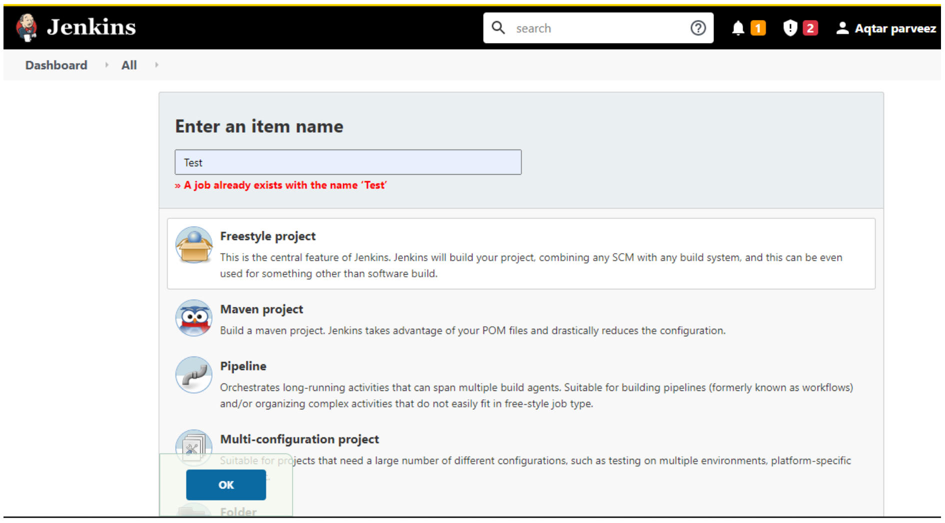
Step-11: Now install docker on both the slave nodes.


```
ubuntu@ip-172-31-37-191:~$ sudo apt install docker.io
```

Check if installed


```
ubuntu@ip-172-31-37-191:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~20.04.2
ubuntu@ip-172-31-37-191:~$
```

Step-12: In Jenkins dashboard, create a new Job by going to Jenkins Dashboard ⇒ Create New Job ⇒ enter Job Name ⇒ select Freestyle Project ⇒ click OK


Jenkins   search   1   2   Aqtar parveez

Dashboard › All ›

Enter an item name

Test

» A job already exists with the name 'Test'

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific

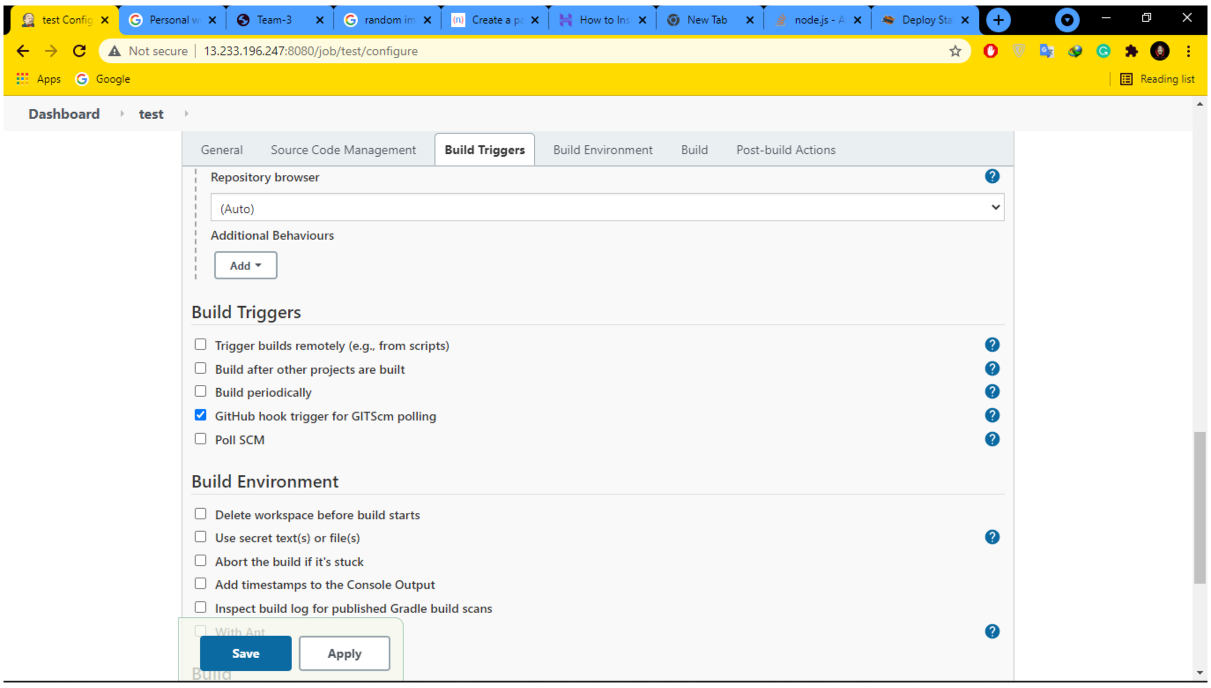
OK

Folder

Step-13: Now configure the new job by entering the details as follows:

- GitHub Project URL - https://github.com/aqtar010/devopsiq.git
- Select 'Restrict where this project can be run' and enter the appropriate node name.
- In 'Source Code Management' select Git and enter the repository URL mentioned above.
- Select 'GitHub hook trigger for GITScm polling' under 'Build Triggers'. (This step is only for the testing server i.e. Slave 1).
- Under 'Build' select 'Execute shell' and enter the commands as shown in the screenshot below.
- Under 'Post-build Actions' select 'Build other projects' and enter the production node name.
- Then select 'Trigger only if build is stable'. (This step should be done only on the testing server).
- Click Save.

This step should be done for both testing and production servers.


13.233.196.247:8080/job/test/configure

Dashboard › test

General | Source Code Management | **Build Triggers** | Build Environment | Build | Post-build Actions

Repository browser: (Auto)

Additional Behaviours: Add

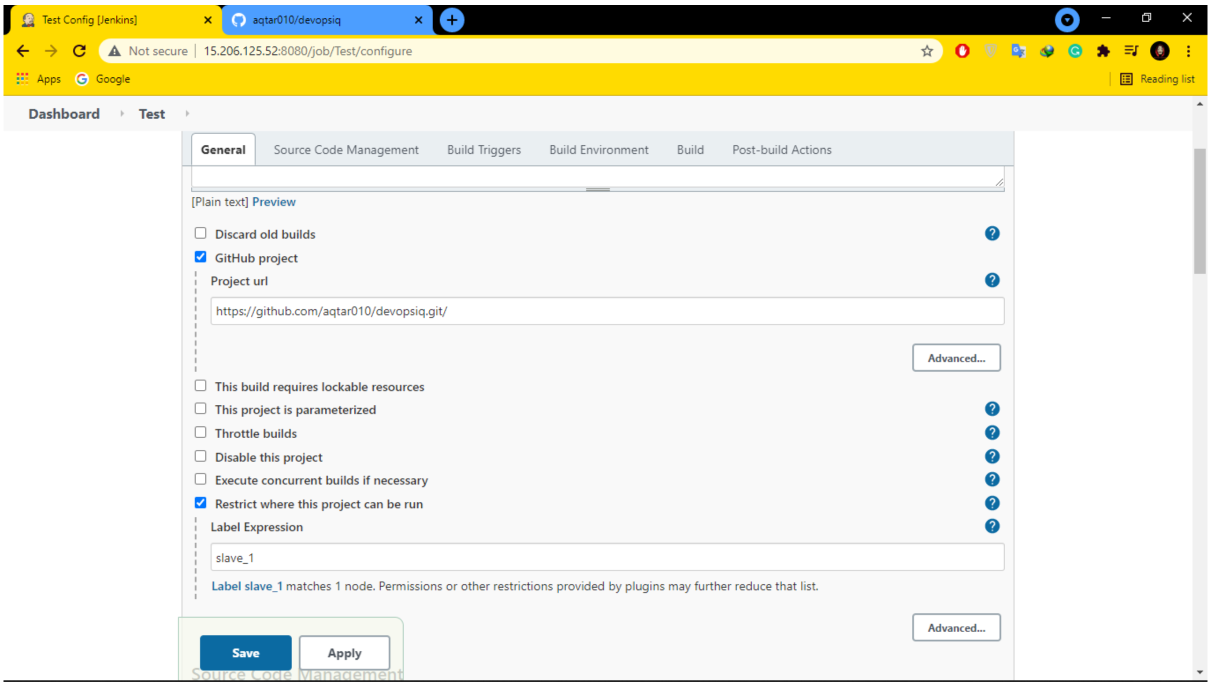
Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [x] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published Gradle build scans
- [ ] With Ant

Save | Apply



[Plain text] Preview

- [ ] Discard old builds
- [x] GitHub project

Project url

https://github.com/aqtar010/devopsiq.git/

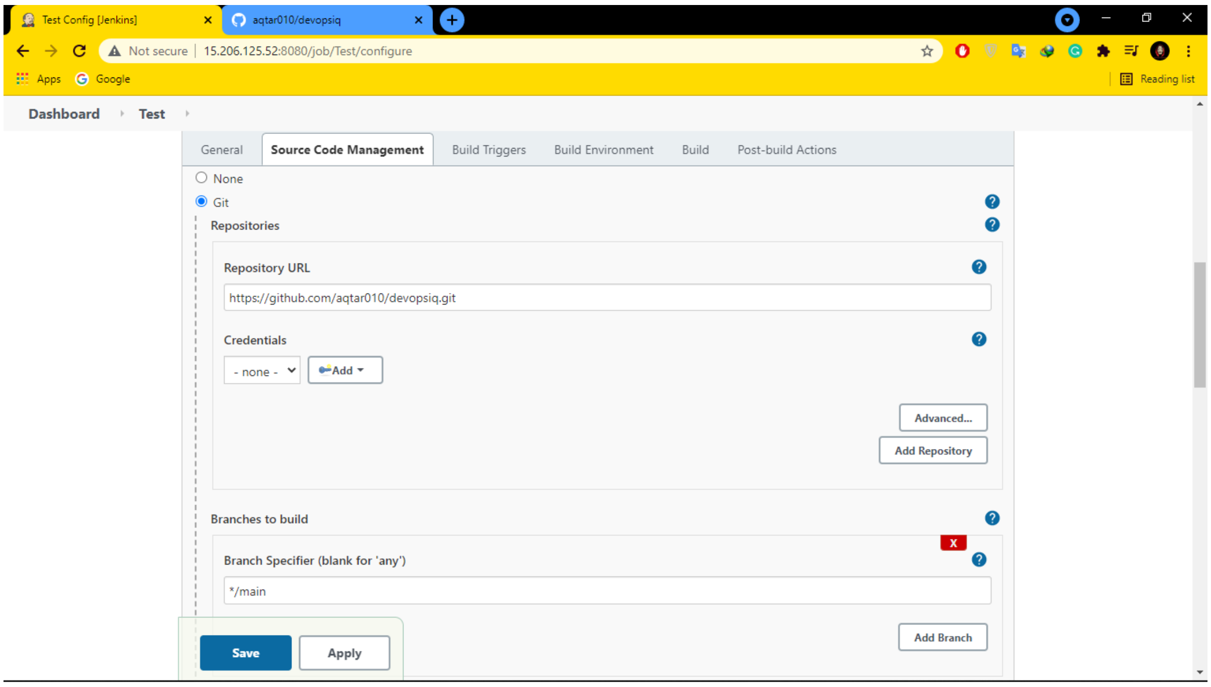
Advanced...

- [ ] This build requires lockable resources
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Disable this project
- [ ] Execute concurrent builds if necessary
- [x] Restrict where this project can be run

Label Expression

slave_1

Label slave_1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.



None [✓ Git]

Repositories

Repository URL

https://github.com/aqtar010/devopsiq.git

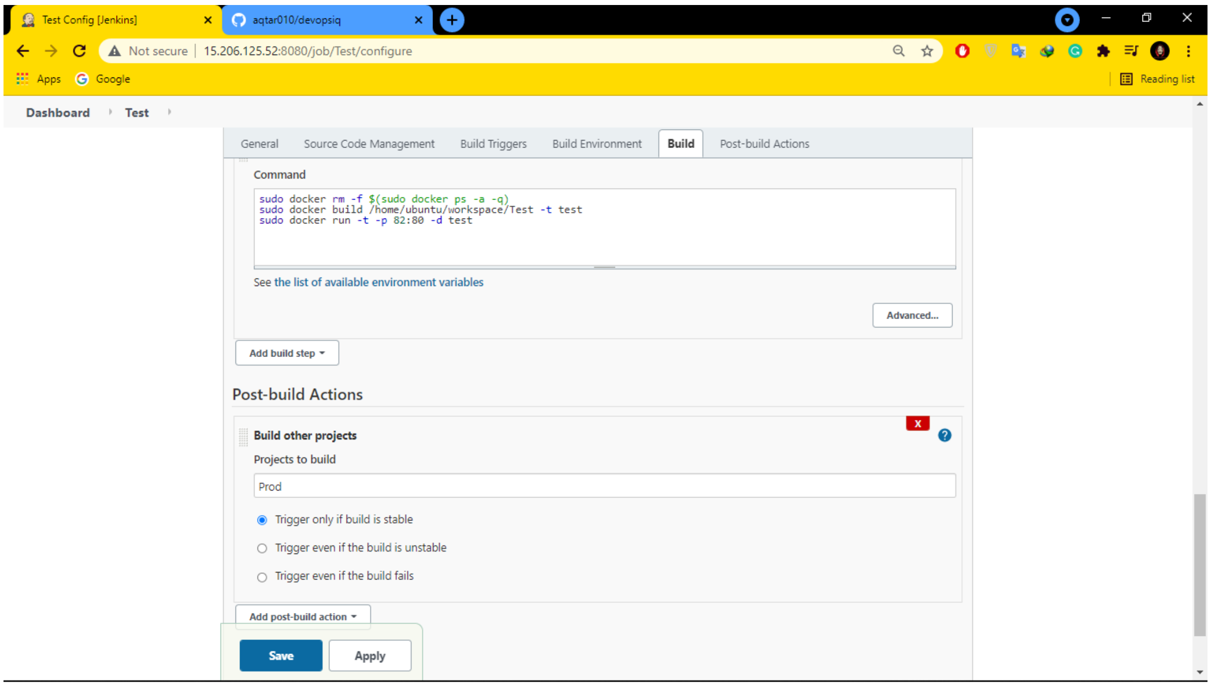
Credentials

- none -

Branches to build

Branch Specifier (blank for 'any')

*/main



Command

```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build /home/ubuntu/workspace/Test -t test
sudo docker run -t -p 82:80 -d test
```

See the list of available environment variables

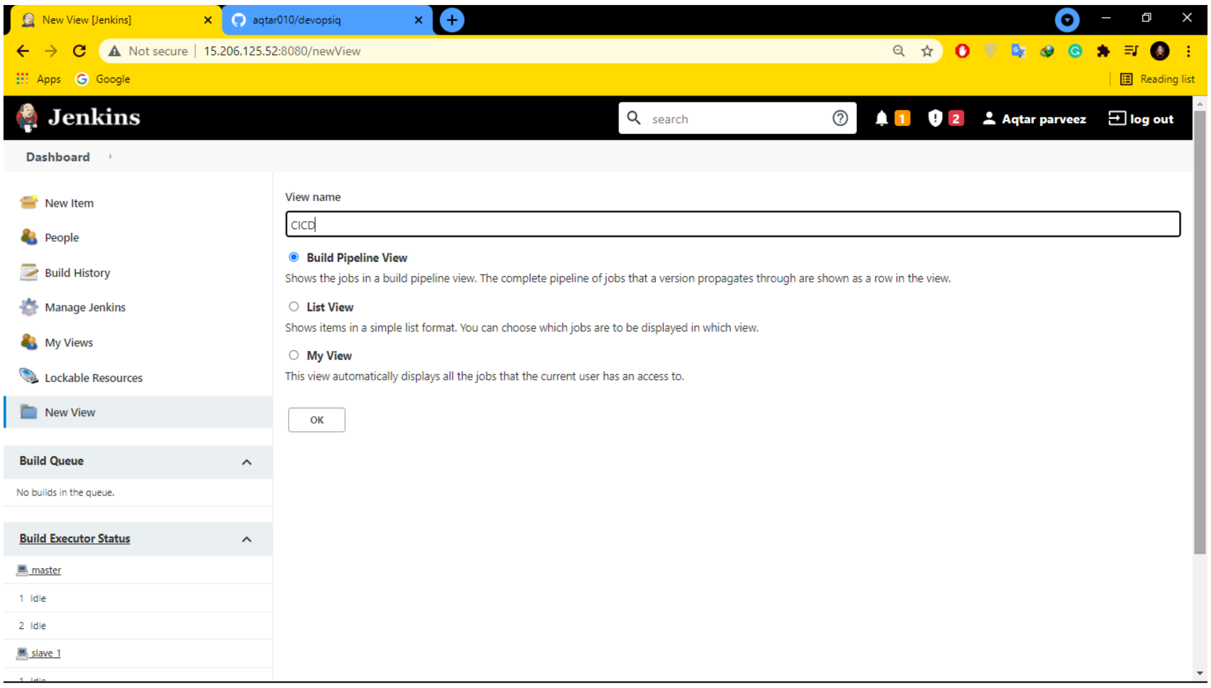
Post-build Actions

Build other projects

Projects to build: Prod

[✓ Trigger only if build is stable]
Trigger even if the build is unstable
Trigger even if the build fails

Step-14: Now setup the Pipeline view. First install the 'build pipeline' plugin in Jenkins. Then go to Home and select the '+' icon beside 'All'. Then select 'Build Pipeline View' and give a name to the view (CICD here). Then click on OK.



No builds in the queue.

1 Idle
2 Idle

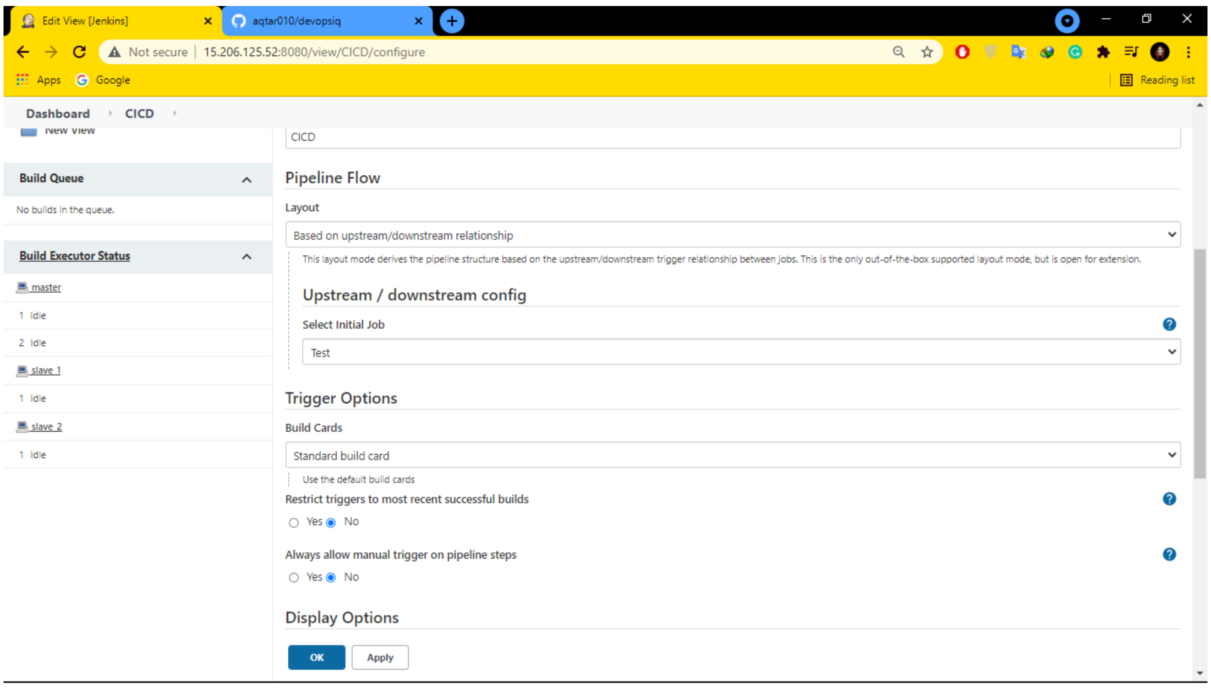
View name: CICD

[✓ Build Pipeline View]
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

My View
This view automatically displays all the jobs that the current user has an access to.

Step-15: Now under ‘Pipeline Flow’, select initial job as ‘Test’. Then click on OK


Pipeline Flow

Layout: Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job: Test

Trigger Options

Build Cards: Standard build card

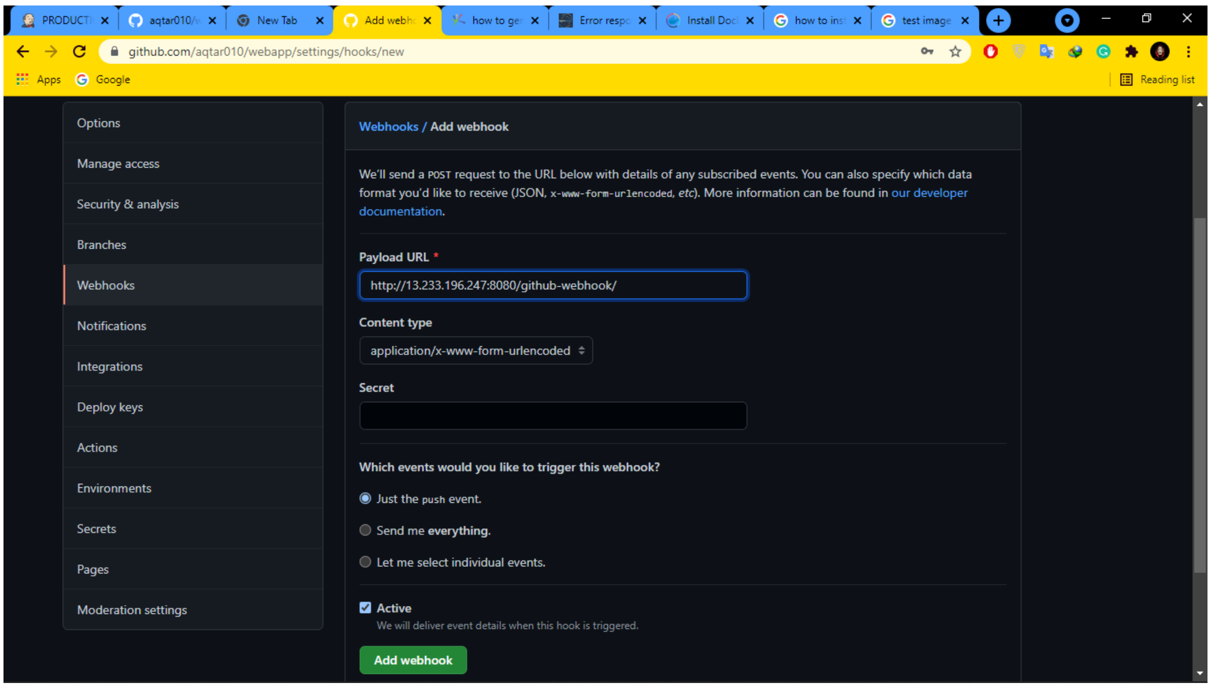
Use the default build cards

Restrict triggers to most recent successful builds: Yes [✓ No]

Always allow manual trigger on pipeline steps: Yes [✓ No]

Display Options

Step-16: Now open the GitHub repository and go to ‘Settings’. Then select ‘Webhooks’. Click on ‘Add webhook’. Then enter the ‘Payload URL’ as ‘http://{ip-address-of-the-Jenkins-master-node}:8080/github-webhook/’ and select ‘Just the push event’ and then click on ‘Add webhook’


Webhooks / Add webhook

We’ll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you’d like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *: http://13.233.196.247:8080/github-webhook/

Content type: application/x-www-form-urlencoded

Secret:

Which events would you like to trigger this webhook?

[✓ Just the push event.]

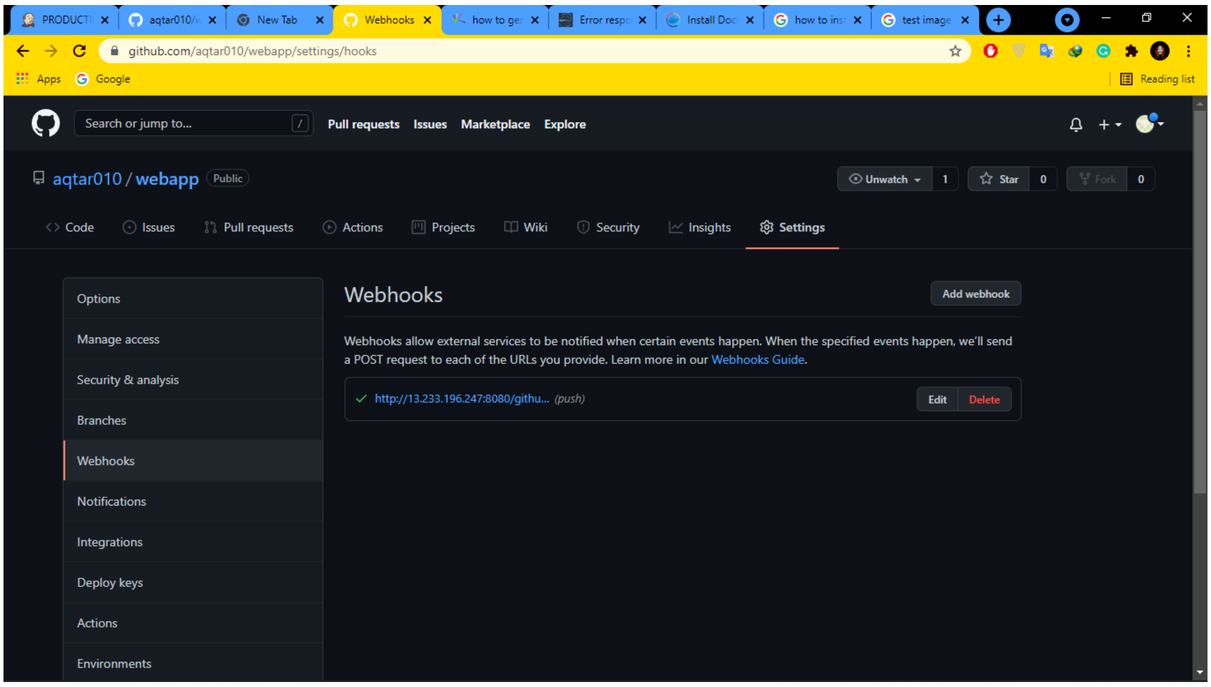
Send me everything.

Let me select individual events.

[✓ Active]

We will deliver event details when this hook is triggered.

After successful ping, the webhook screen will look as in the Screenshot below with a green verified tick mark



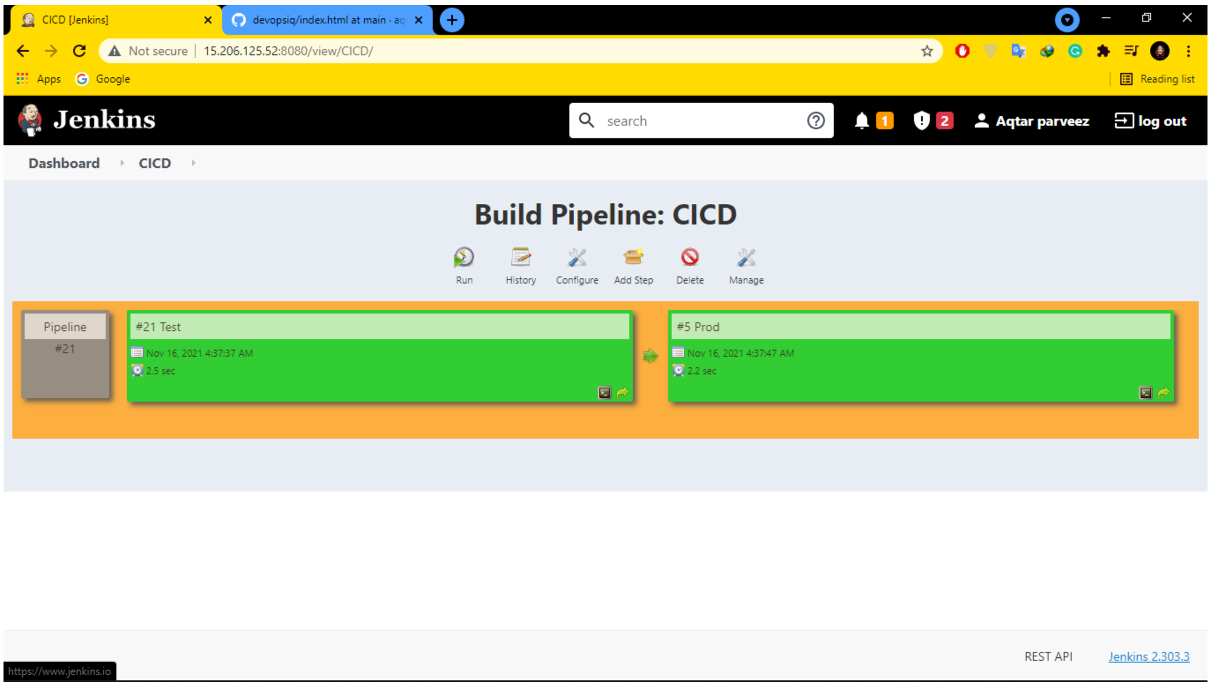
Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://13.233.196.247:8080/githu... (push)

Step-17: Now commit changes and push to the repository to trigger the webhook. This will trigger the Jenkins CICD pipeline and build the docker image on the test server first, and on successful build, it will build the same on the production server as well.

CICD pipeline looks like this



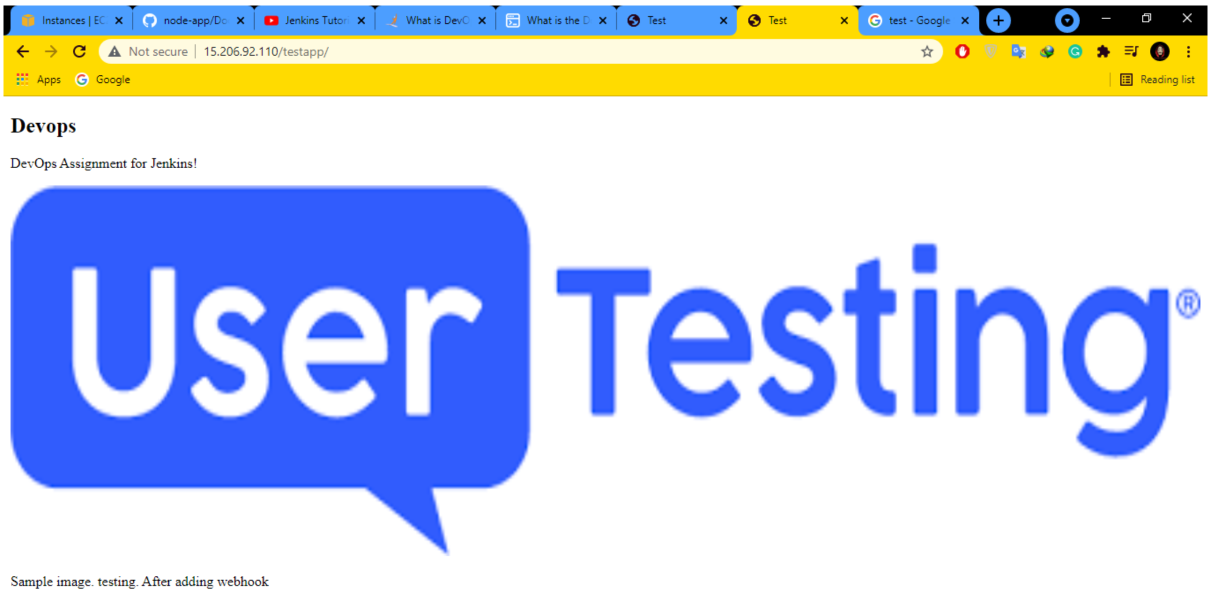
Build Pipeline: CICD

Pipeline #21	#21 Test Nov 16, 2021 4:37:37 AM 2.5 sec	#5 Prod Nov 16, 2021 4:37:47 AM 2.2 sec

On Successful deployment Both the slave nodes will host the updated websites.

On any changes committed in the GitHub Repo the changes are reflected after a very short delay.

Slave 1 node:

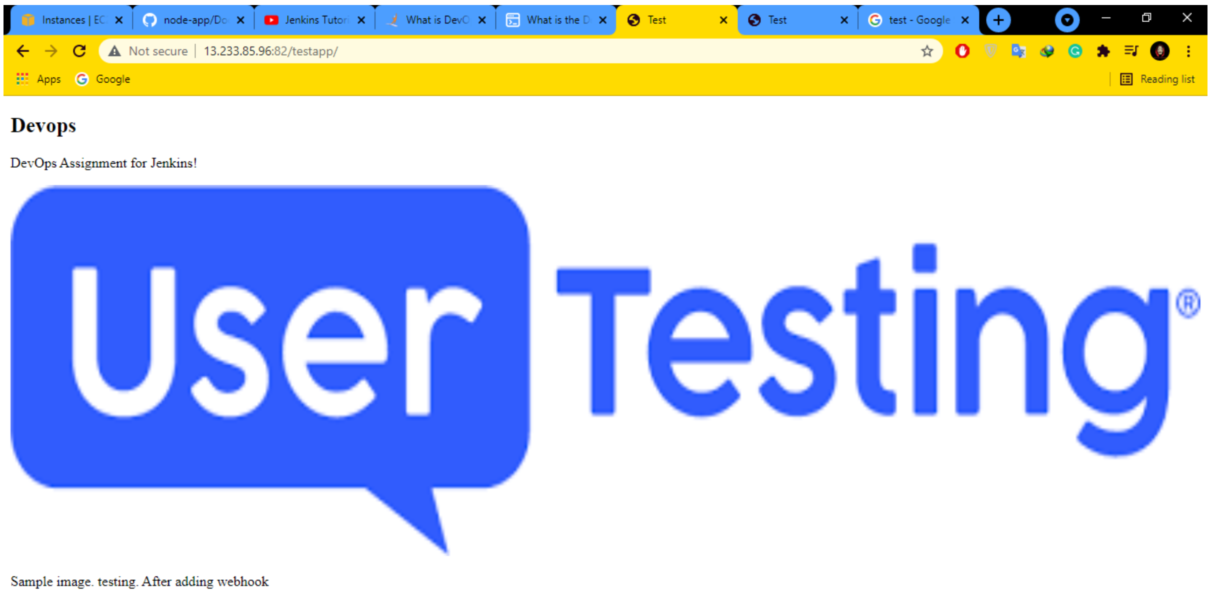


Devops

DevOps Assignment for Jenkins!

Sample image. testing. After adding webhook

Slave 2 Node:



Devops

DevOps Assignment for Jenkins!

Sample image. testing. After adding webhook