Jenkins Master Slave Configuration

Here, we will learn how to setup Jenkins Master Slave configuration in AWS EC2 server.

Module – 1:

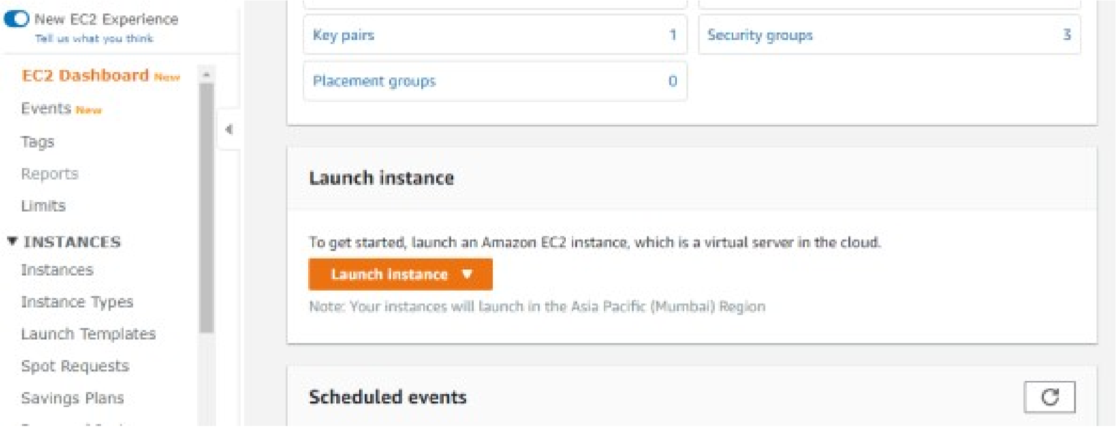
Create and run a new two EC2 Linux instance in AWS.

Prerequisite:

We'll be using all the options which are eligible under the AWS Free tier account. To follow along this tutorial please create a Free tier eligible account in AWS.

Steps to create a new two EC2 instance:

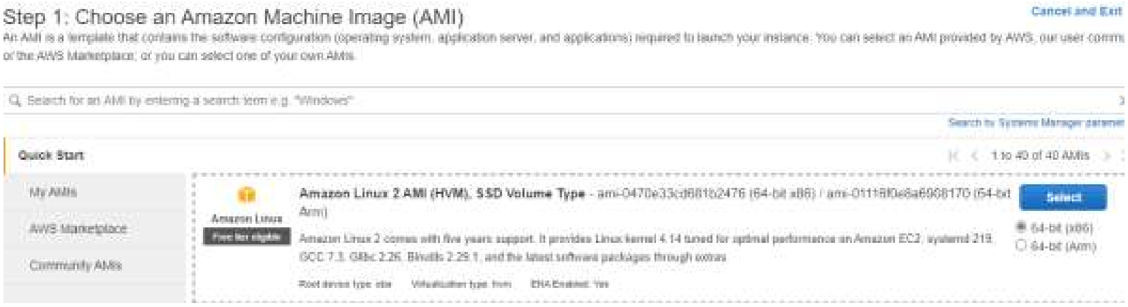
1. In the AWS console click on **Services** -> **EC2.**
2. Click on **Launch instance**.



To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

3. Choose any Amazon Machine Image for Linux (Ex – Amazon Linux 2 AMI (HVM)) and click on **Select.**



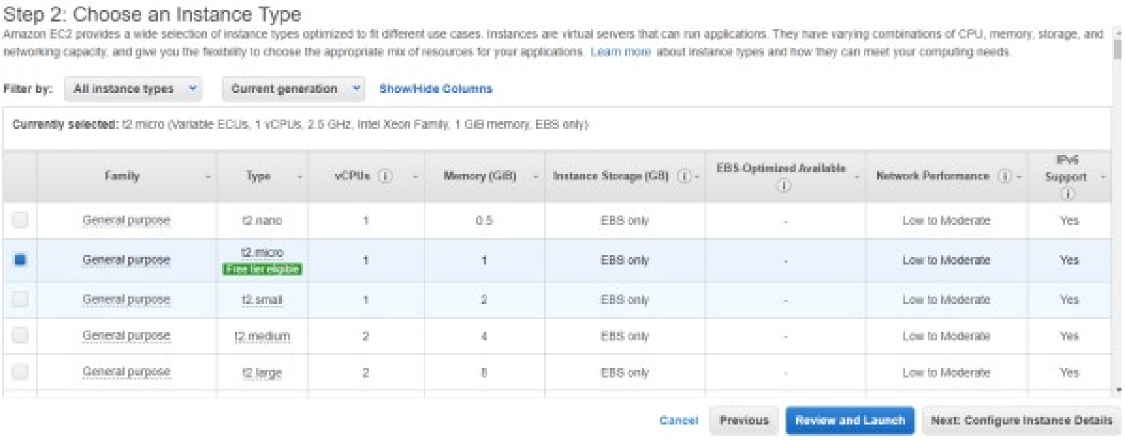
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user comm of the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0470e33cd681b2476 (64-bit x86) / ami-0111f0e8a6908170 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

4. Choose any Instance Type (Ex – General Purpose, t2.micro) and click on **Next.**



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

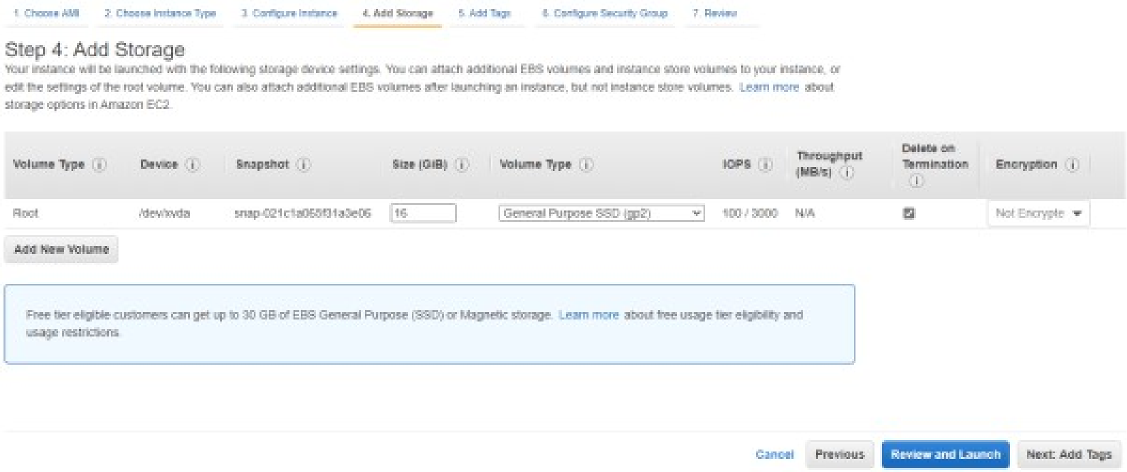
Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

5. Don't change anything on the Configure Instance Details page and click on **Next.**

6. In Add Storage page, enter the volume size (Ex – 16 GB) and click on **Next.**



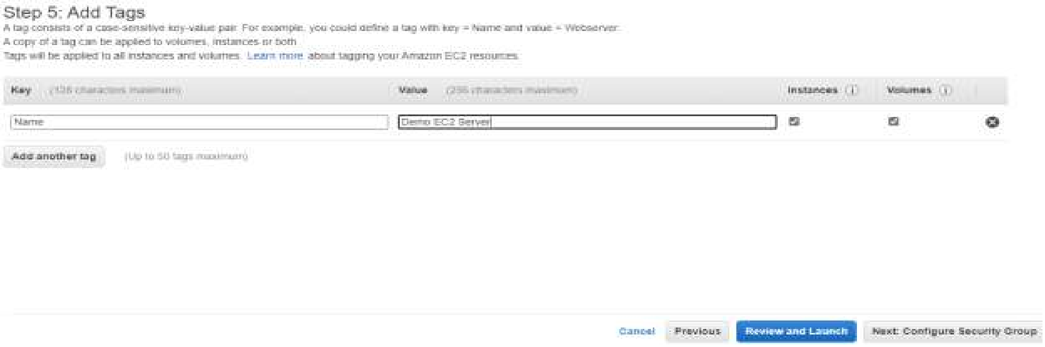
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-021c1a065f31a3e06	16	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypte

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

7. Click on **Add Tag,** enter Key = Name and Value = Demo EC2 Server. Click on **Next.**

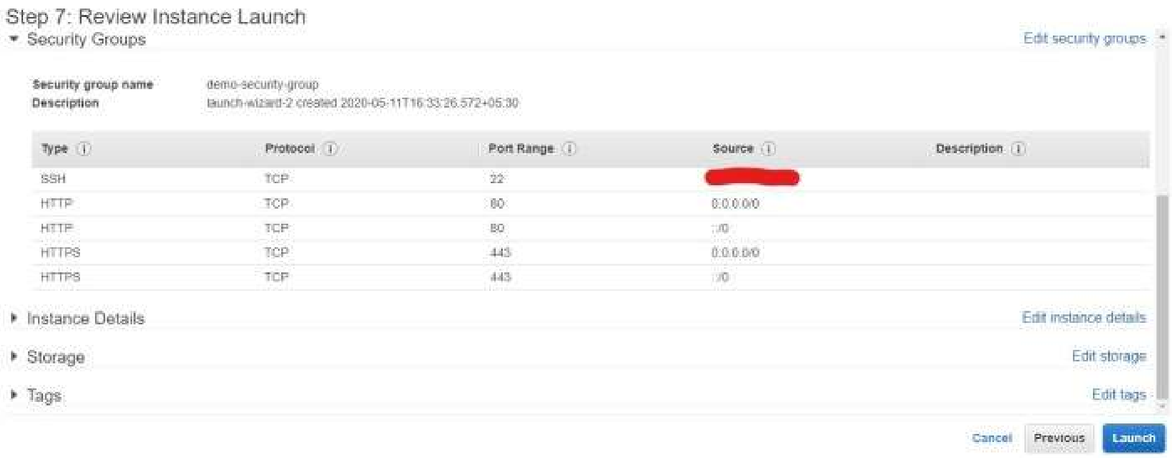


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	
Name	Demo EC2 Server	☑	☑	⊗

8. In Configure Security Page, enter a Security group name: demo-security-group and add the following Rules.



Step 7: Review Instance Launch

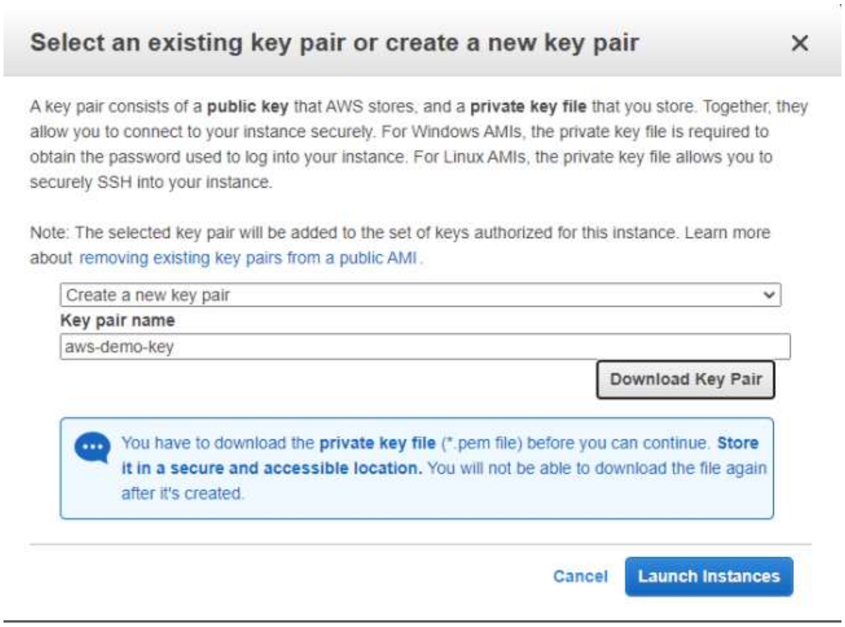
Security group name demo-security-group
Description launch-wizard-2 created 2020-05-11T16:33:26.572+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	[redacted]	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	::/0	

9. Click on **Review and Launch.**

10. In Review Instance Launch page, review all the details entered for ec2 instance and click on Launch.

11. Choose Create a new key pair, enter a new key pair name “aws-demo-key” and click on **Download Key Pair.**



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

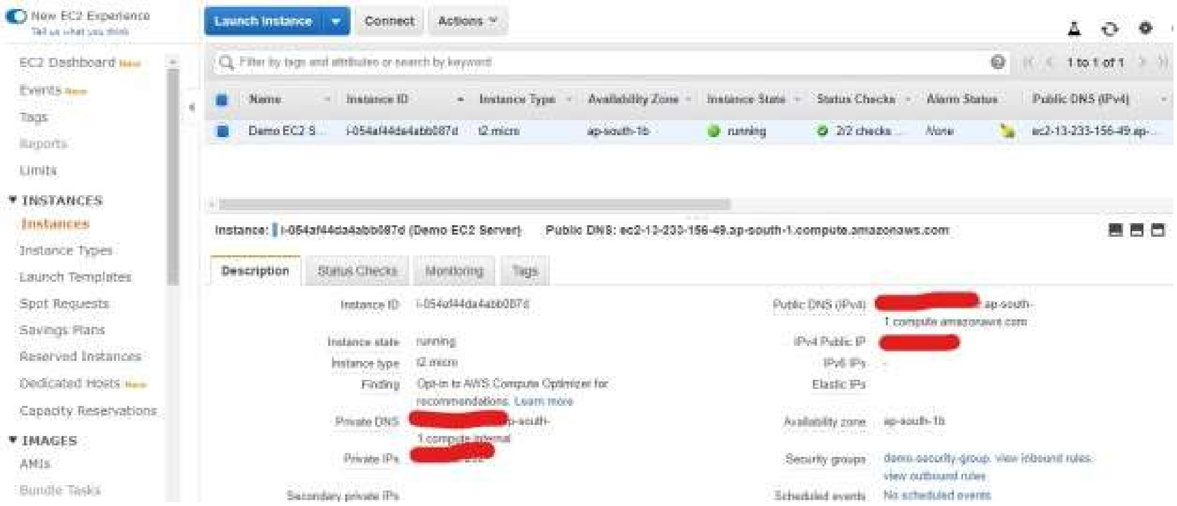
Create a new key pair

Key pair name: aws-demo-key

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

12. Click on **Launch Instances**.

13. Go back to your EC2 dashboard and a new Linux EC2 instance will be in running status.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Demo EC2 S...	i-054af44da4abb087d	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-233-156-49.ap-...

Instance: i-054af44da4abb087d (Demo EC2 Server) Public DNS: ec2-13-233-156-49.ap-south-1.compute.amazonaws.com

Instance ID	i-054af44da4abb087d	Public DNS (IPv4)	[redacted] ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	[redacted]
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	[redacted] p-south-1.compute.internal	Availability zone	ap-south-1b
Private IPs	[redacted]	Security groups	demo-security-group. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events

Jenkins Master Slave Configuration

Module – 2:

How to setup a Jenkins Build Server on AWS EC2.

Prerequisites:

1. AWS two EC2 instances are running.

2. Connect to your two EC2 instances with SSH.

Steps to Download and Install Jenkins:

1. Run the following command to update all software packages on ec2 instance.

```
sudo yum update
```

2. Add the Jenkins repo using the following command:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo
```

3. Import a key file from Jenkins-CI to enable installation from the package:

```
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key
```

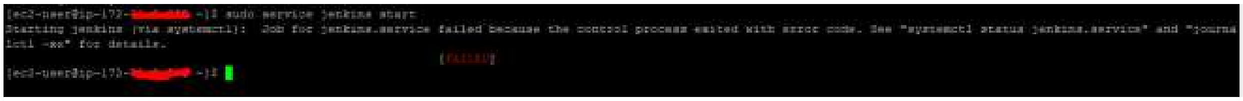
4. Install Jenkins:

```
sudo yum install jenkins -y
```

5. Start Jenkins as a service:

```
sudo service jenkins start
```

If you get the below error while starting the jenkins service, then run the following command to update java on EC2:

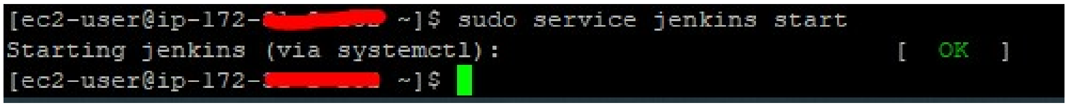


```
[ec2-user@ip-172-[redacted] ~]$ sudo service jenkins start
Starting jenkins (via systemctl):  Job for jenkins.service failed because the control process exited with error code. See "systemctl status jenkins.service" and "journa
lctl -xe" for details.
                                                           [FAILED]
[ec2-user@ip-172-[redacted] ~]$
```

```
sudo amazon-linux-extras install java-openjdk11
```

Run the command again to start Jenkins service.

Jenkins is now installed and running on your EC2 instance.



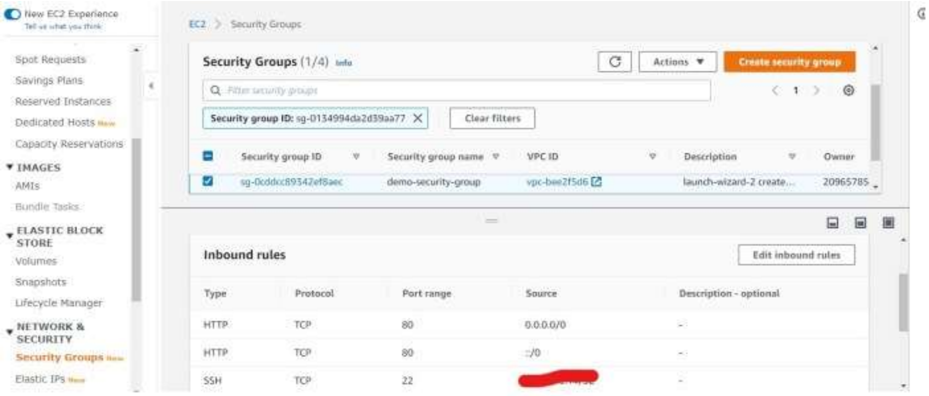
```
[ec2-user@ip-172-[redacted] ~]$ sudo service jenkins start
Starting jenkins (via systemctl):                          [  OK  ]
[ec2-user@ip-172-[redacted] ~]$
```

Modify EC2 Security Group:

1. Go to EC2 Dashboard.

Jenkins Master Slave Configuration

2. Go to Security Groups and select the security group associated with EC2 instance. (Ex: demo-security-group)



Security group ID: sg-0134994da2d39aa77

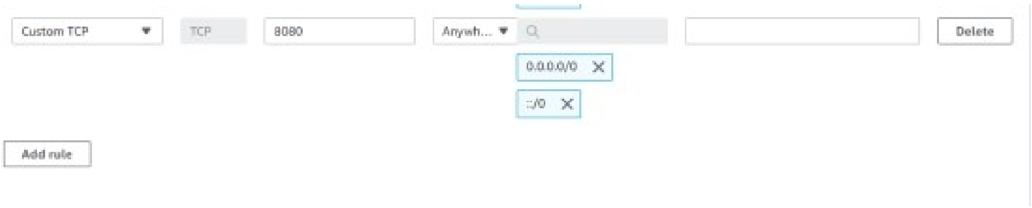
☑	Security group ID	Security group name	VPC ID	Description	Owner
☑	sg-0cddcc89342ef8aec	demo-security-group	vpc-bee2f5d6	launch-wizard-2 create...	20965785

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
SSH	TCP	22	[redacted]	-

3. Click on Edit Inbound Rules.

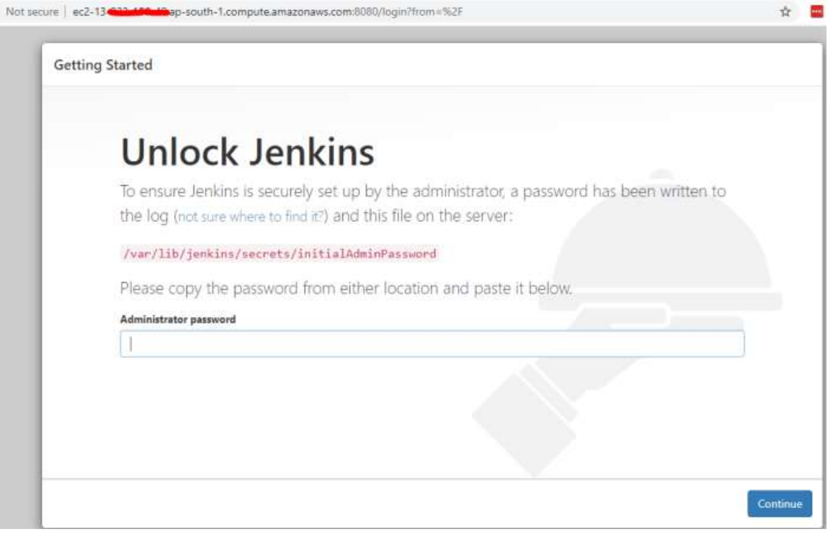
4. Click Add Rule, and then choose Custom TCP Rule from the Type list. Under Port Range enter 8080.



Custom TCP ▼	TCP	8080	Anywh... ▼	0.0.0.0/0 ✕ ::/0 ✕		Delete

Configure Jenkins:

1. Connect to http://<ec2-server-public-dns>:8080 from your browser. You will be able to access Jenkins through its management interface.



Not secure | ec2-13[redacted]ap-south-1.compute.amazonaws.com:8080/login?from=%2F

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

2. Enter the password found in /var/lib/jenkins/secrets/initialAdminPassword. Use the following
command to display this password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

3. The Jenkins installation script directs you to the Customize Jenkins page. Click Install suggested plugins.

4. Create a new Admin User and complete the setup.

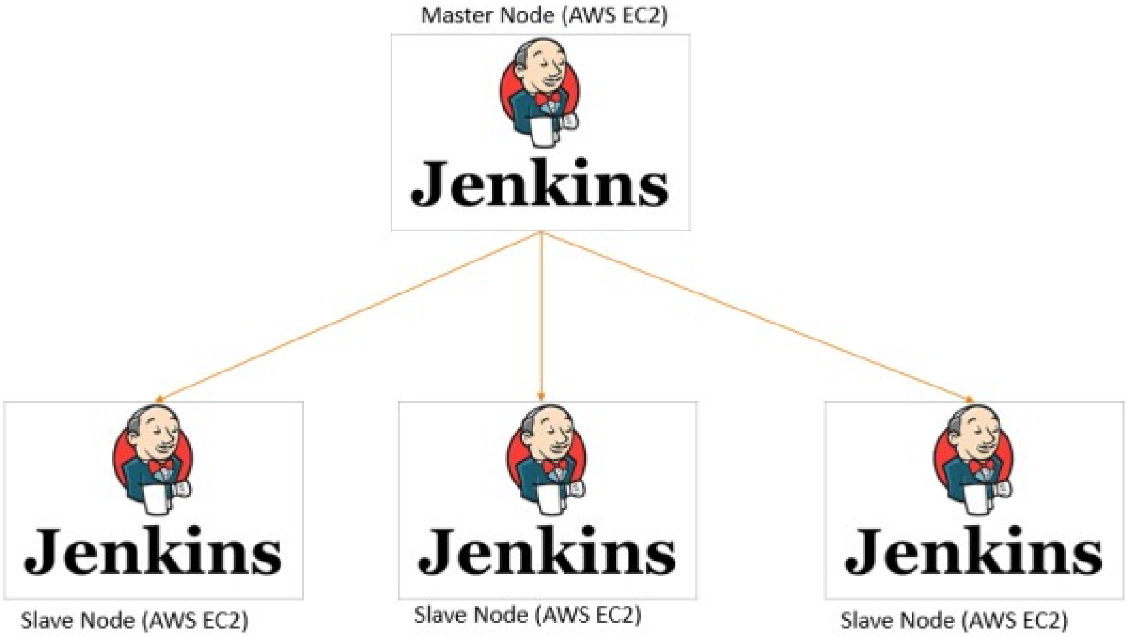
5. Click Start Using Jenkins.

Jenkins Build Server is ready to be used on the AWS EC2 server.

Jenkins Master Slave Configuration

Module – 3:

How to setup Jenkins Master Slave configuration in AWS EC2 server.



Prerequisites:

1. Launch 2 AWS EC2 Linux instances **(Module – 1)**
2. Configure Jenkins on an EC2 Linux instance which will act as Master Node
3. Second EC2 Linux Server will act as a Slave Node for Jenkins.

Configure Jenkins Slave Node:

Create user, ssh keys and copy it to authorized_keys.

```
sudo useradd jenkins-slave1
sudo su - jenkins-slave1
ssh-keygen -t rsa -N "" -f /home/jenkins-slave1/.ssh/id_rsa
cd .ssh
cat id_rsa.pub > authorized_keys
chmod 700 authorized_keys
```

Configure Jenkins Master Node:

Copy the slave node’s public key to master node’s known_hosts file.

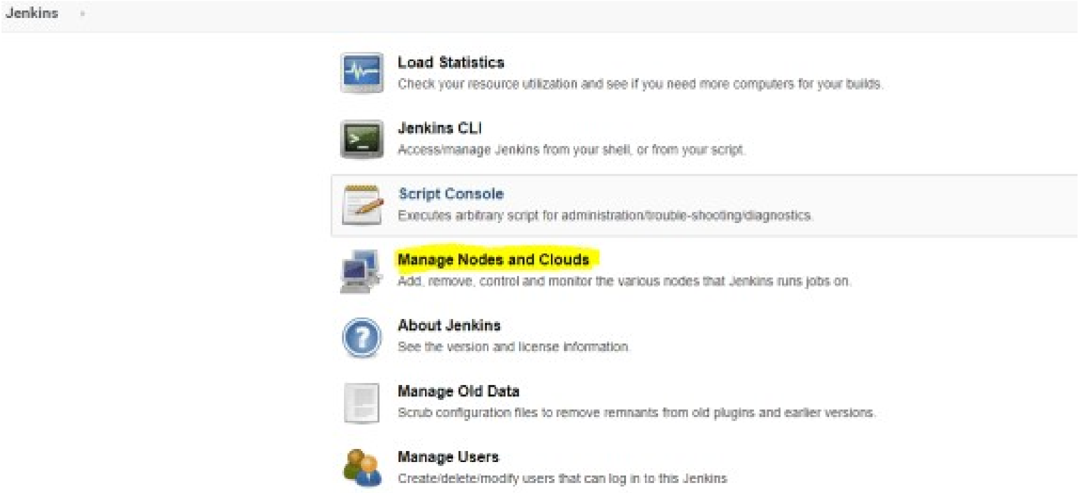
```
sudo mkdir -p /var/lib/jenkins/.ssh
cd /var/lib/jenkins/.ssh
cd ..
sudo chmod 777 .ssh
cd .ssh
sudo ssh-keyscan -H SLAVE_NODE_PRIVATE_IP >>/var/lib/jenkins/.ssh/known_hosts
sudo chown jenkins:jenkins known_hosts
sudo chmod 700 known_hosts
```

Jenkins Master Slave Configuration

Configure New Node in Jenkins:

1. Click on Manage Jenkins.

2. Click on Manage Nodes and Clouds



Jenkins ›

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

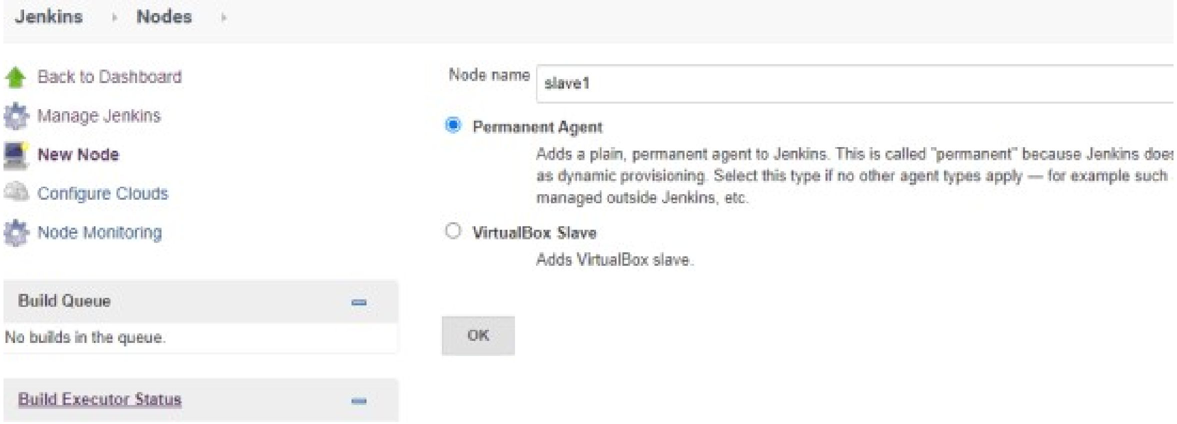
About Jenkins
See the version and license information.

Manage Old Data
Scrub configuration files to remove remnants from old plugins and earlier versions.

Manage Users
Create/delete/modify users that can log in to this Jenkins

3. Click on New Node

4. Enter Node name and select Permanent Agent. Click OK



Jenkins › Nodes ›

Back to Dashboard
Manage Jenkins
New Node
Configure Clouds
Node Monitoring

Build Queue
No builds in the queue.

Build Executor Status

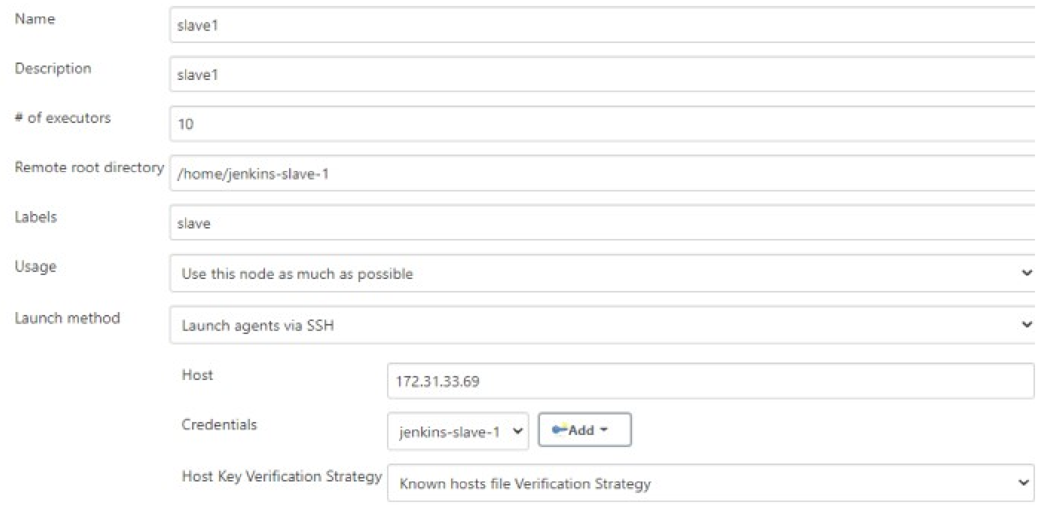
Node name: slave1

● Permanent Agent
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doe
as dynamic provisioning. Select this type if no other agent types apply — for example such
managed outside Jenkins, etc.

○ VirtualBox Slave
Adds VirtualBox slave.

OK

5. Enter node configuration details as shown below.



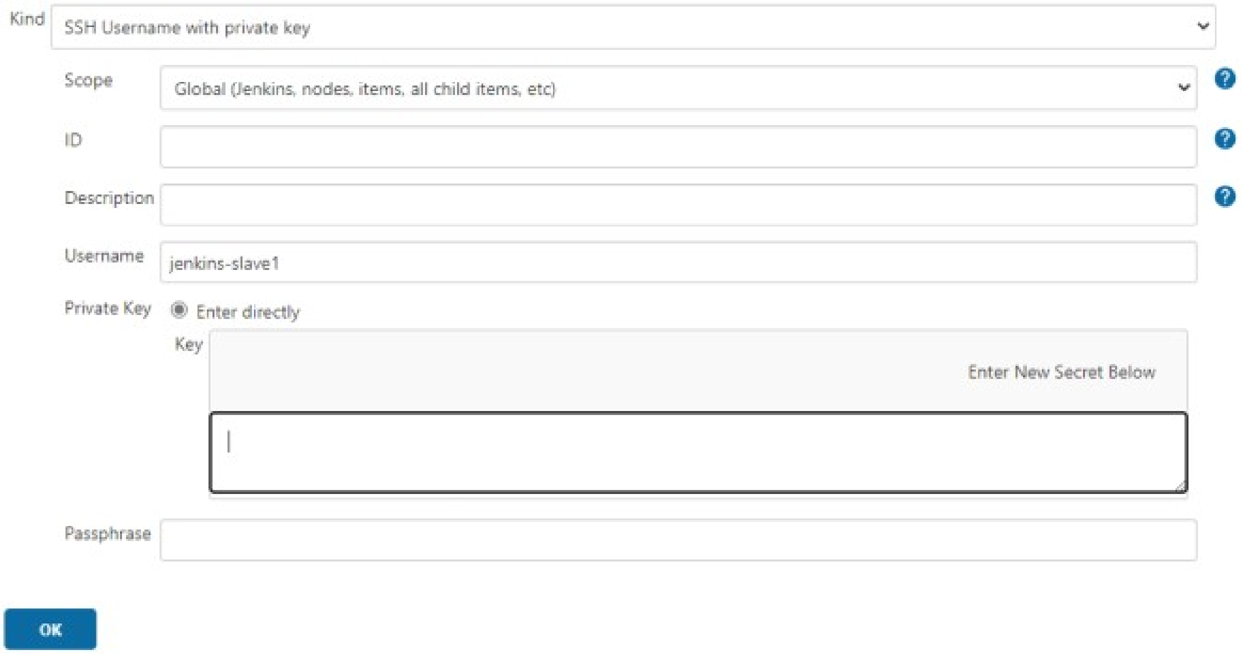
Name	slave1
Description	slave1
# of executors	10
Remote root directory	/home/jenkins-slave-1
Labels	slave
Usage	Use this node as much as possible
Launch method	Launch agents via SSH
Host	172.31.33.69
Credentials	jenkins-slave-1
Host Key Verification Strategy	Known hosts file Verification Strategy

6. In Credentials, Click Add -> Jenkins

7. In Add Credentials, choose kind as “SSH Username with private key”

8. In Username field, enter the same username which we created on Slave Node – jenkins-slave1

9. In Private Key, choose Enter directly



Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID:

Description:

Username: jenkins-slave1

Private Key: ● Enter directly

Key: Enter New Secret Below

Passphrase:

10. Go to Slave EC2 server and copy the private key.

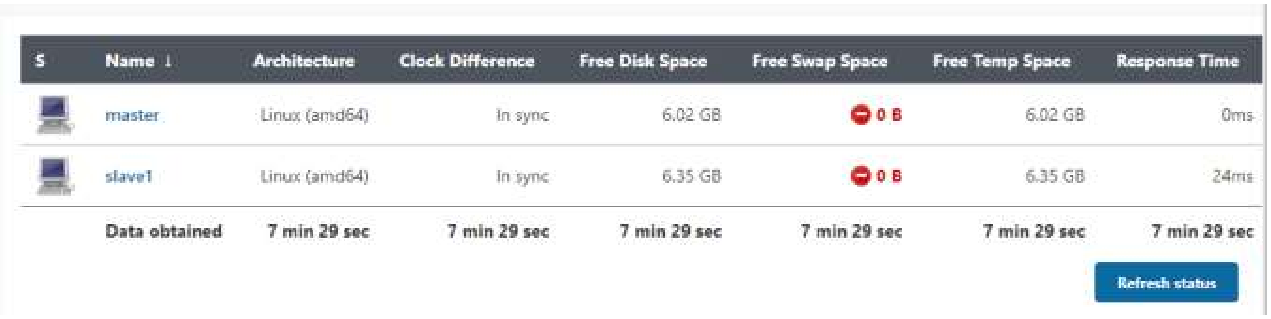
```
sudo su - jenkins-slave1
cd .ssh
more id_rsa
```

11. Paste the key in the Private key field in Jenkins.

12. Click on Add.

13. In the Node Configuration page choose the new credential.

14. Click on Save.

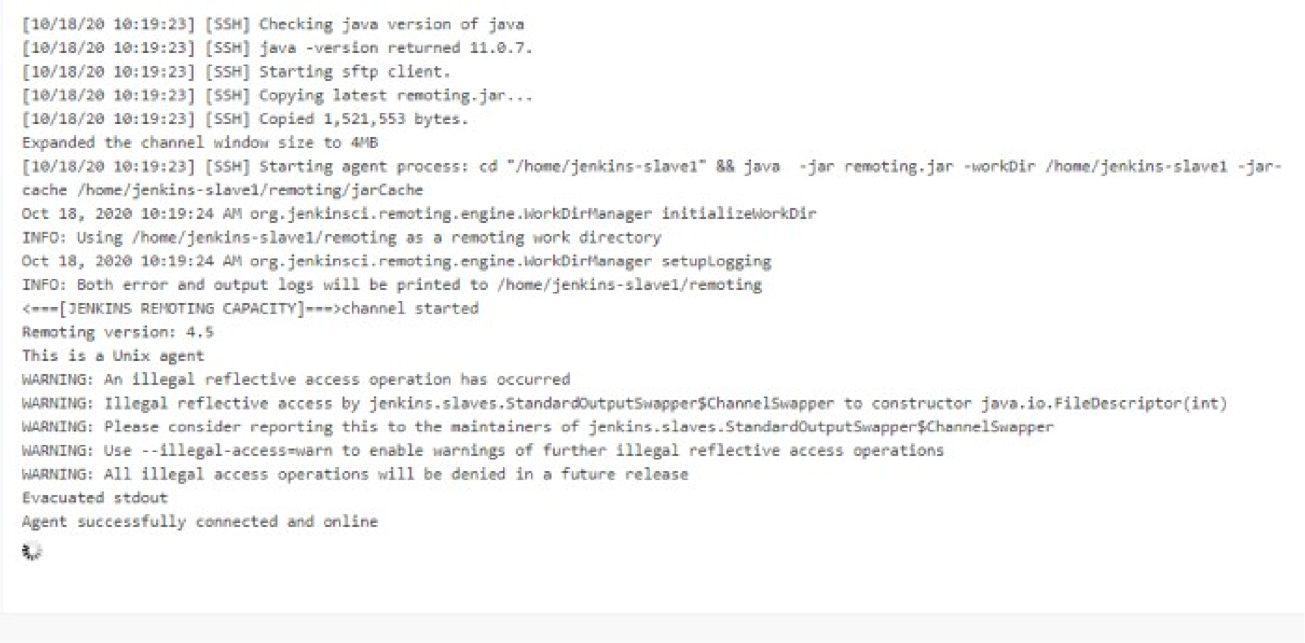


S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	6.02 GB	⛔ 0 B	6.02 GB	0ms
	slave1	Linux (amd64)	In sync	6.35 GB	⛔ 0 B	6.35 GB	24ms
	Data obtained	7 min 29 sec	7 min 29 sec	7 min 29 sec	7 min 29 sec	7 min 29 sec	7 min 29 sec

Jenkins Master Slave Configuration

15. Click on the new node (slave1) and select “log”.

16. “Agent successfully connected and online” is displayed in the logs.



```
[10/18/20 10:19:23] [SSH] Checking java version of java
[10/18/20 10:19:23] [SSH] java -version returned 11.0.7.
[10/18/20 10:19:23] [SSH] Starting sftp client.
[10/18/20 10:19:23] [SSH] Copying latest remoting.jar...
[10/18/20 10:19:23] [SSH] Copied 1,521,553 bytes.
Expanded the channel window size to 4MB
[10/18/20 10:19:23] [SSH] Starting agent process: cd "/home/jenkins-slave1" && java  -jar remoting.jar -workDir /home/jenkins-slave1 -jar-
cache /home/jenkins-slave1/remoting/jarCache
Oct 18, 2020 10:19:24 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins-slave1/remoting as a remoting work directory
Oct 18, 2020 10:19:24 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins-slave1/remoting
<===[JENKINS REMOTING CAPACITY]===>channel started
Remoting version: 4.5
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSwapper$ChannelSwapper to constructor java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSwapper$ChannelSwapper
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```