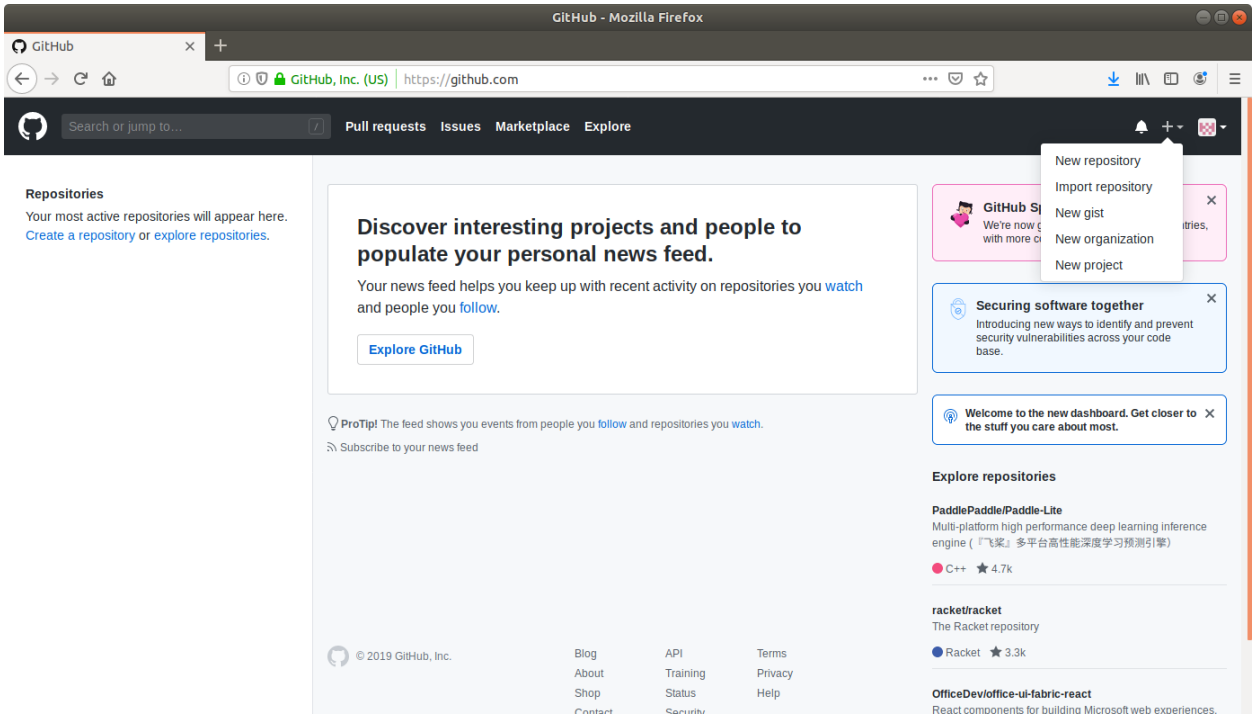
Distributed Builds

This section will guide you to:

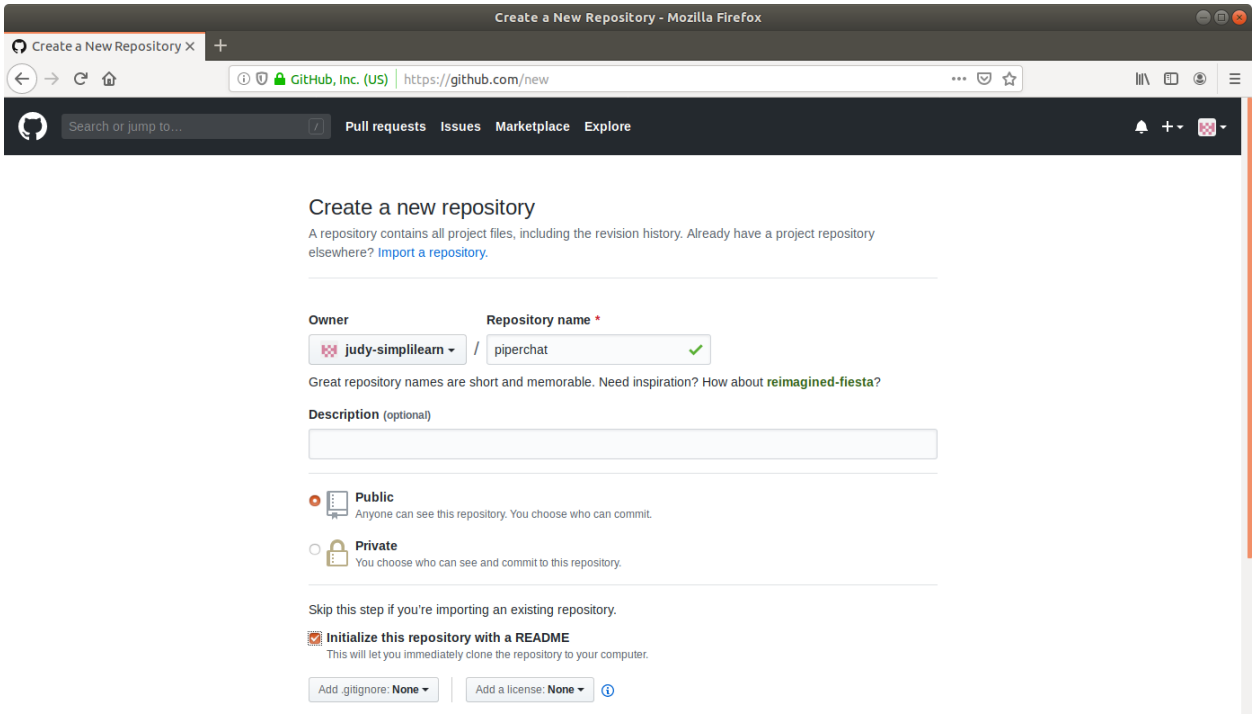
- Create an EC2 instance
- Create and run Jenkins on an EC2 VM
- Connect a Jenkins slave node to a master node
- Create a Maven web app
- Trigger a build job on a slave node from a master node

Step 1: Creating a Git repository for the web app

- Log in to your Github account.
- Click on the plus icon next to the profile picture and select *New repository* from the drop-down menu.



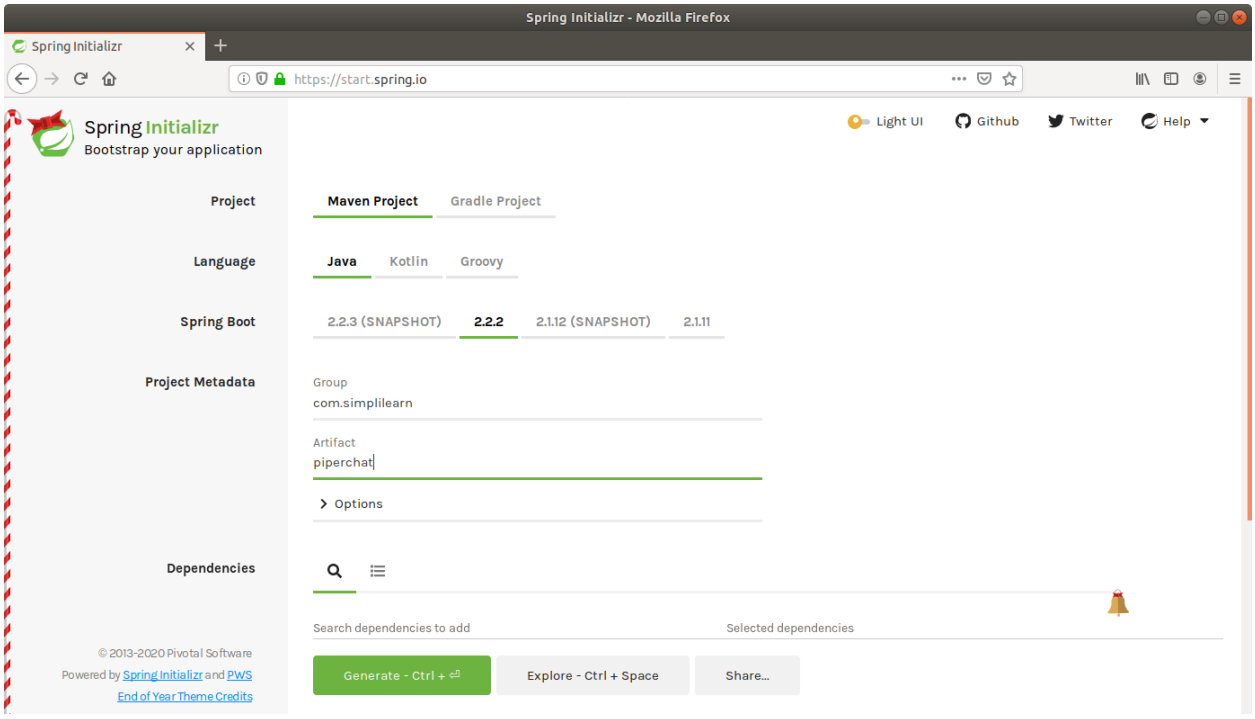
- Fill the required fields in the create repository form.



- Click on the **Create Repository** button.
- Click on the **Clone or download** button and copy the URL.

Step 2: Generating a spring boot project

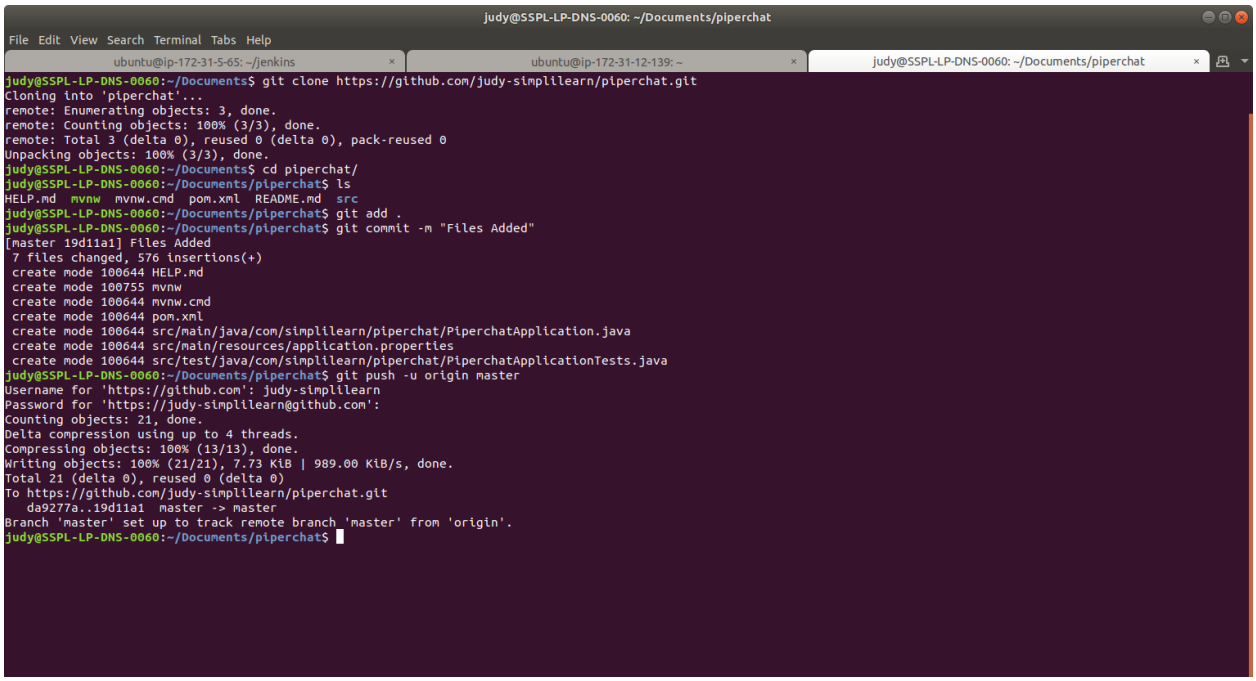
- Go to start.spring.io/



- Select Maven as the project type.
- Fill Group and Artifact with appropriate values. For example, *com.simplilearn* and *freshmart-pos.*
- Click on **Generate Project.**
- The generated skeleton project should be downloaded as a zip file.

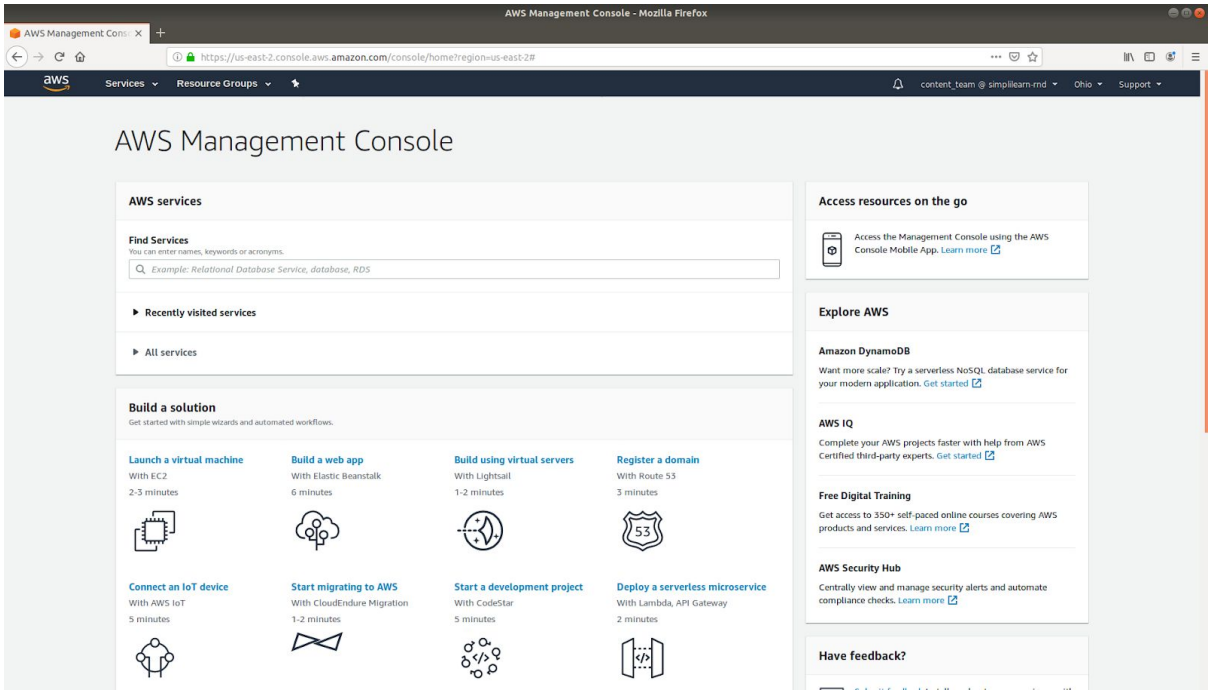
Step 3: Committing the project skeleton to the repository

- Open the terminal and navigate to an appropriate location.
- Run **git clone [URL]** to clone the repository.
- Unzip the downloaded spring boot project to the cloned repository.
- Commit the changes to the remote SCM.
- Run **git add .**
- Run **git commit -m "Add project skeleton"**
- Run **git push -u origin master**

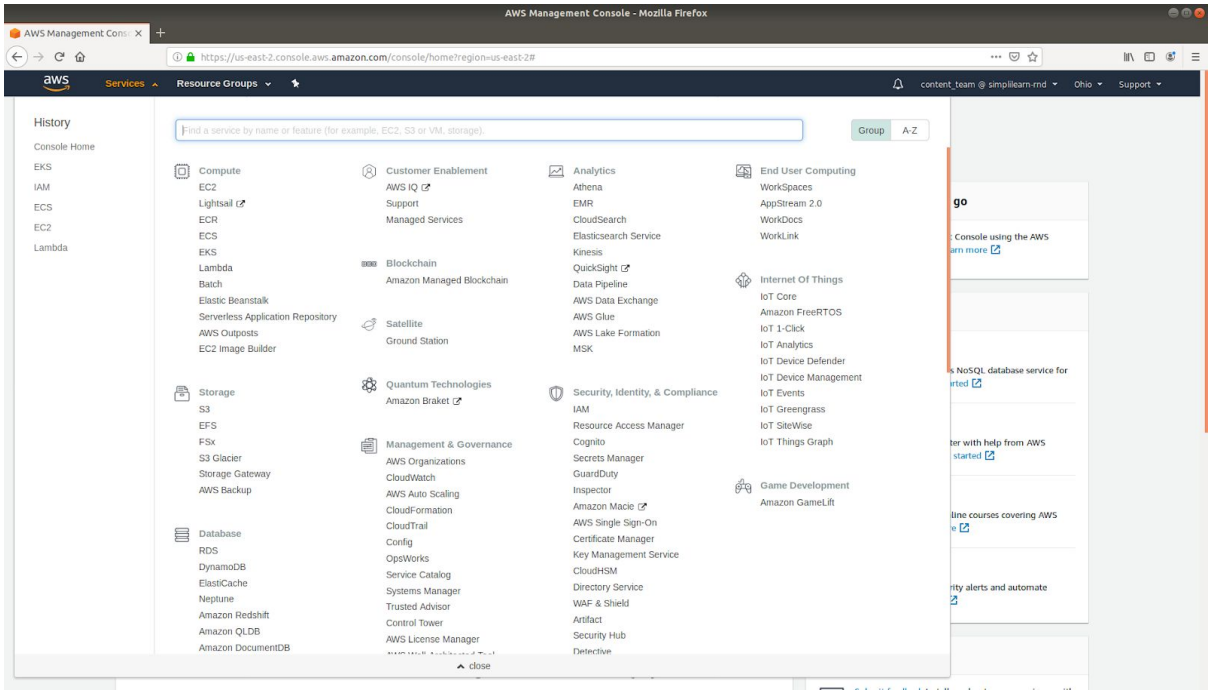


Step 4: Creating an EC2 instance

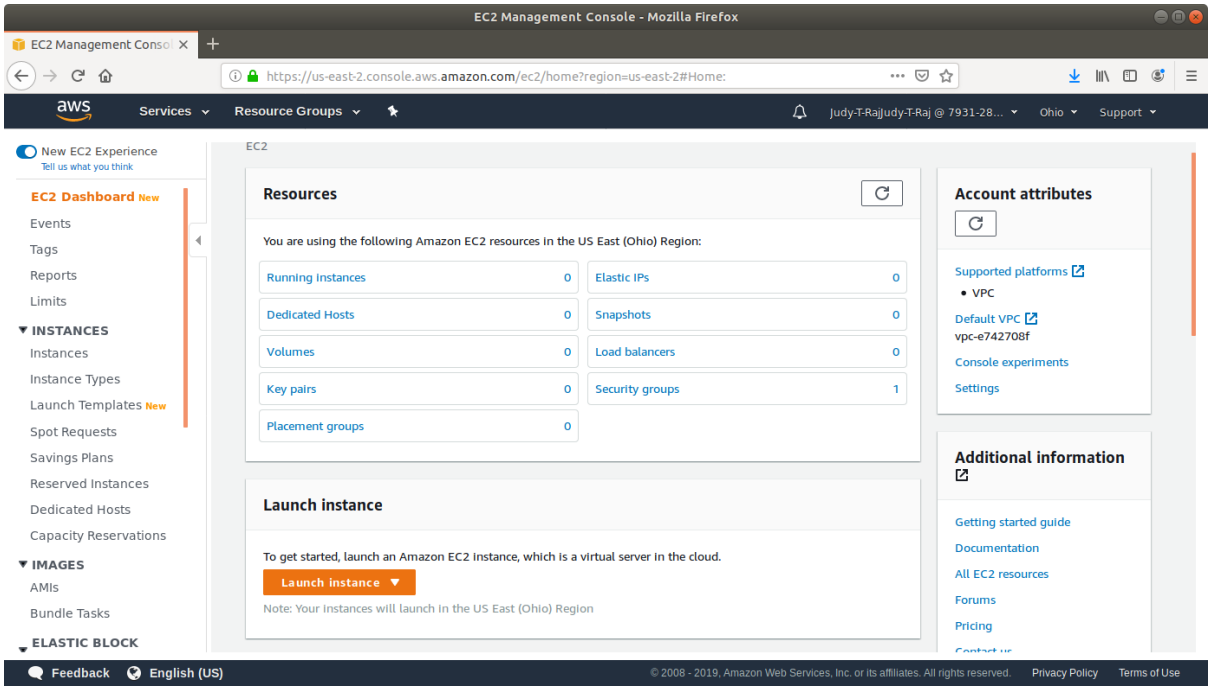
- Log in to the AWS lab account provided. You will be able to see the following screen:



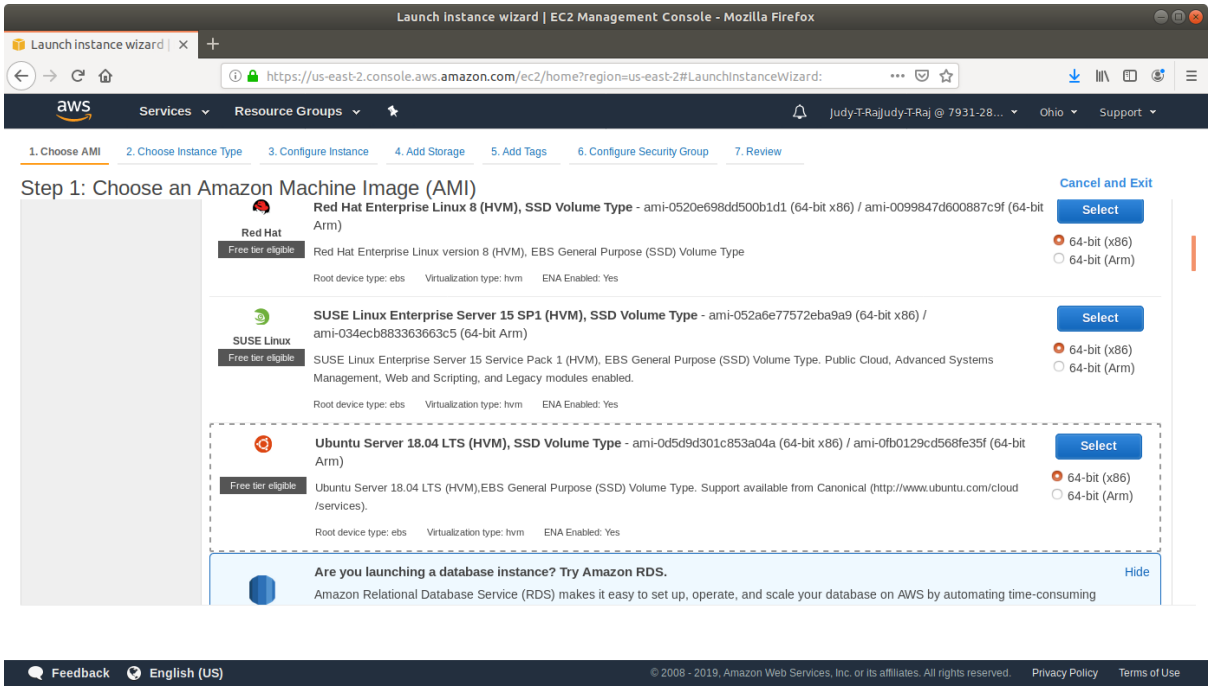
- Click on Services at the top left to view the drop-down list of resources.
- Click on EC2 under the Compute menu from the drop-down list.



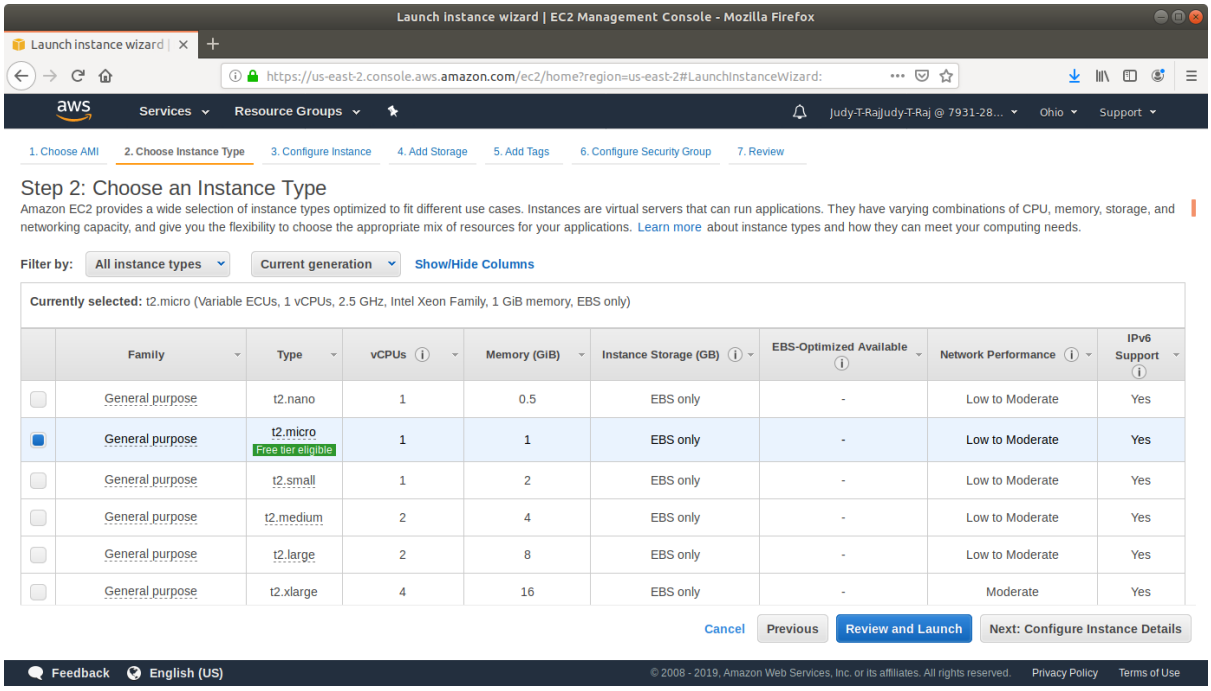
- Click on *Launch Instance* button and select Launch Instance from the menu.



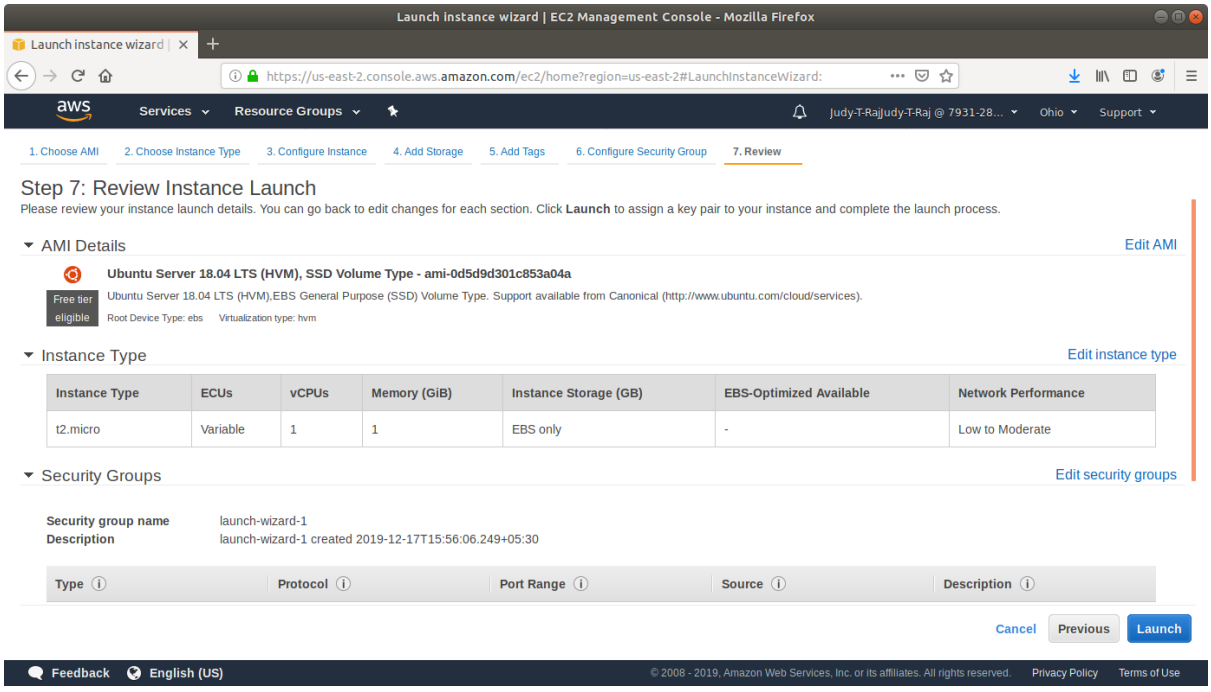
- Choose an Amazon Machine Image (AMI) from the list of AMIs and click on Select.



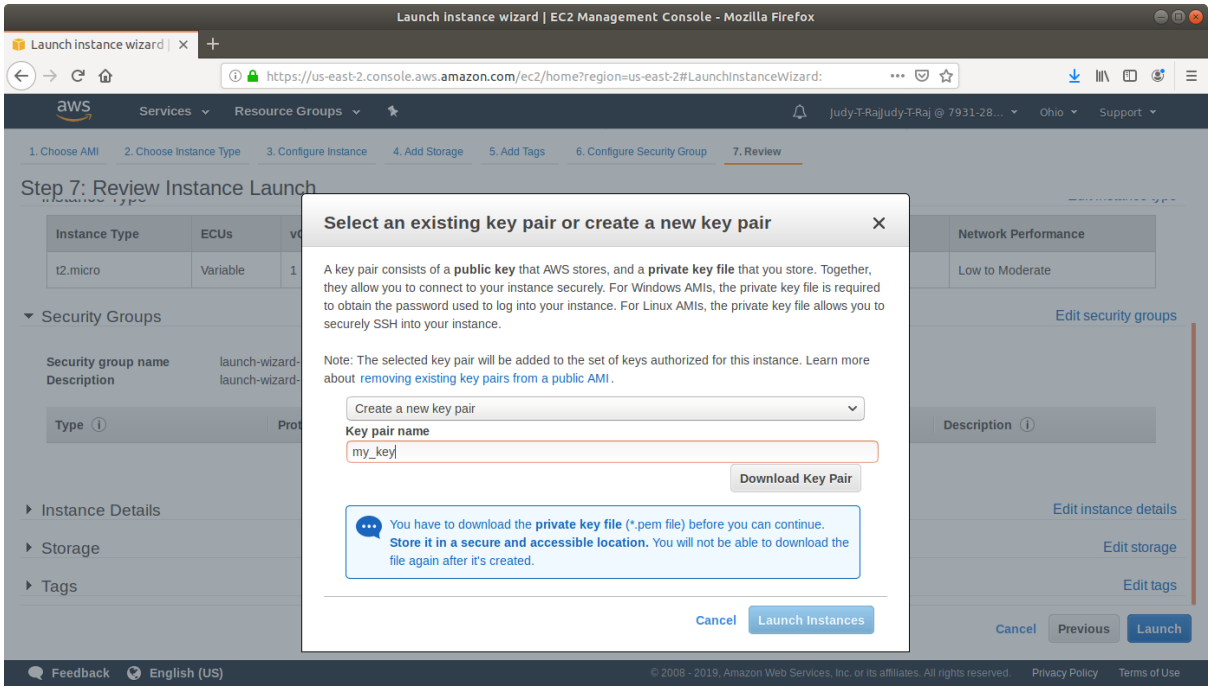
- Choose an Instance Type and click *Review and Launch*.



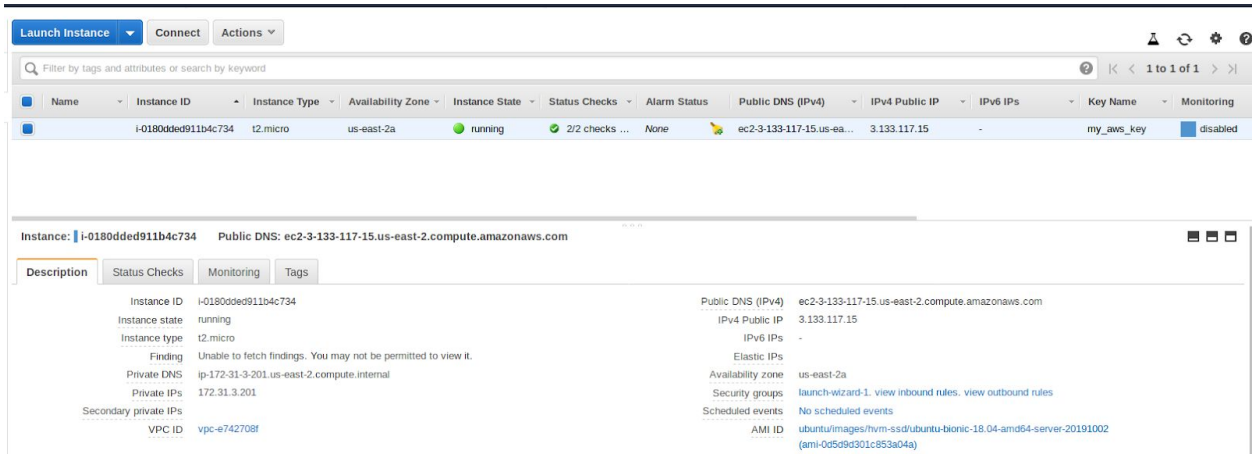
- Click on *Launch*.



- In the pop-up menu, select Create a new key-value pair.
- Click on *Download Key Pair*. You'll need this key to SSH to the VM later.



- Click on *Launch*.
- Navigate to the security groups console.



- Add a rule to the security group to which the instance belongs to allow SSH with the following settings:

  Type: SSH

  Protocol: TCP

  Port Range: 22

Source: Anywhere 0.0.0.0/0

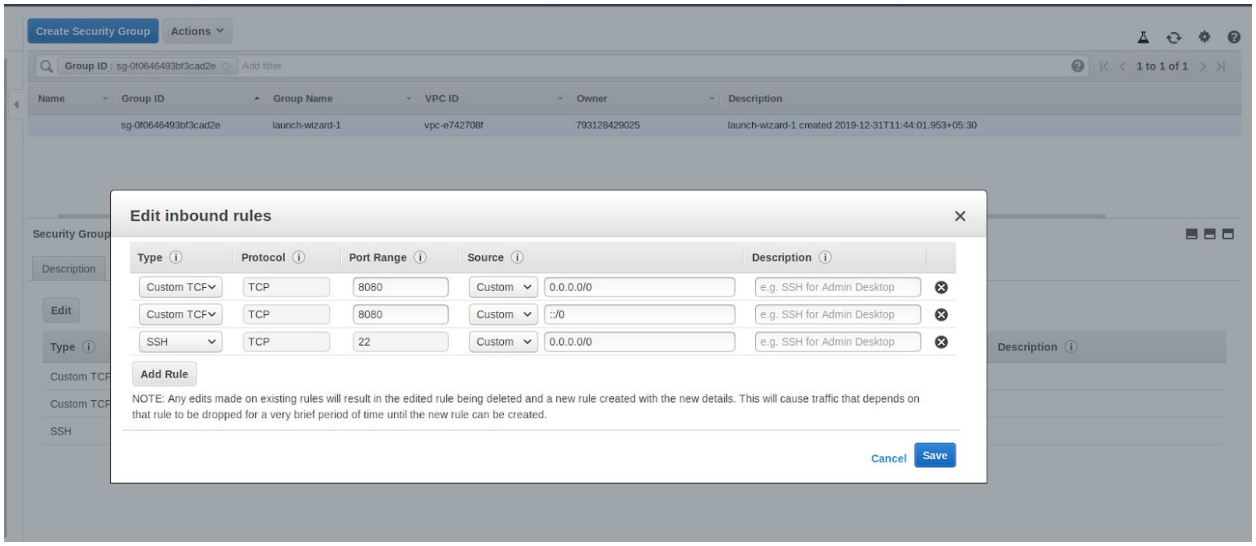
- Add a rule to the security group to which the instance belongs to allow HTTP traffic to port 8080 with the following settings:

Type: Custom TCP Rule

Protocol: TCP

Port Range: 8080

Source: Anywhere 0.0.0.0/0



Step 5: Installing Jenkins on EC2

- Open the terminal.
- Navigate to the location where the AWS key is stored.
- Make the key file executable with the command **chmod 400 <key-name>.pem**
- SSH to the EC2 instance with the command **sudo ssh -i <key-name>.pem ubunutu@<public-dns>**

```
judy@SSPL-LP-DNS-0060:~/Downloads$ chmod 400 my_aws_key.pem
judy@SSPL-LP-DNS-0060:~/Downloads$ sudo ssh -i "my_aws_key.pem" ubuntu@ec2-3-133-117-15.us-east-2.compute.amazonaws.com
[sudo] password for judy:
The authenticity of host 'ec2-3-133-117-15.us-east-2.compute.amazonaws.com (3.133.117.15)' can't be established.
ECDSA key fingerprint is SHA256:QucLPccAiVK9XoH0RdV++ysA1NkErThNdtLmBctMZaE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-133-117-15.us-east-2.compute.amazonaws.com,3.133.117.15' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

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

  System information as of Tue Dec 31 06:29:31 UTC 2019

  System load:  0.0               Processes:           86
  Usage of /:   13.8% of 7.69GB   Users logged in:     0
  Memory usage: 17%               IP address for eth0: 172.31.3.201
  Swap usage:   0%

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

- Run the following commands to install Java and Jenkins

  sudo apt update

  sudo apt install openjdk-8-jdk

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

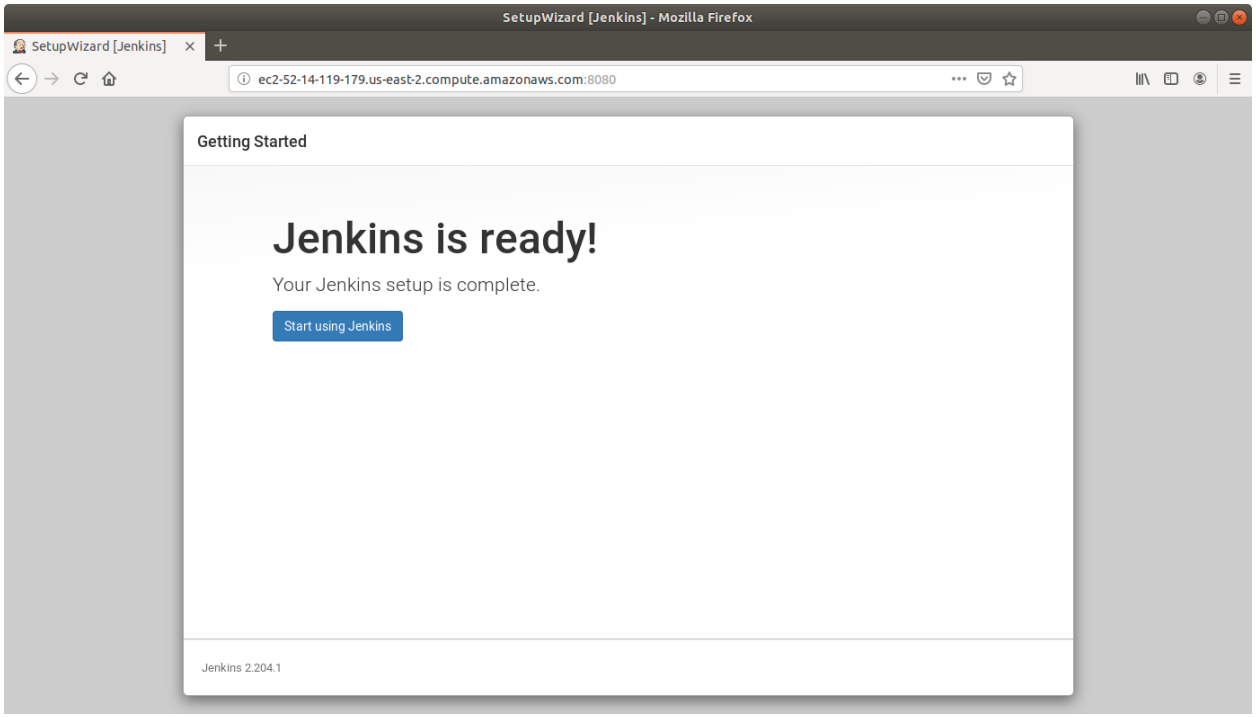
  sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

  sudo apt update

  sudo apt install jenkins

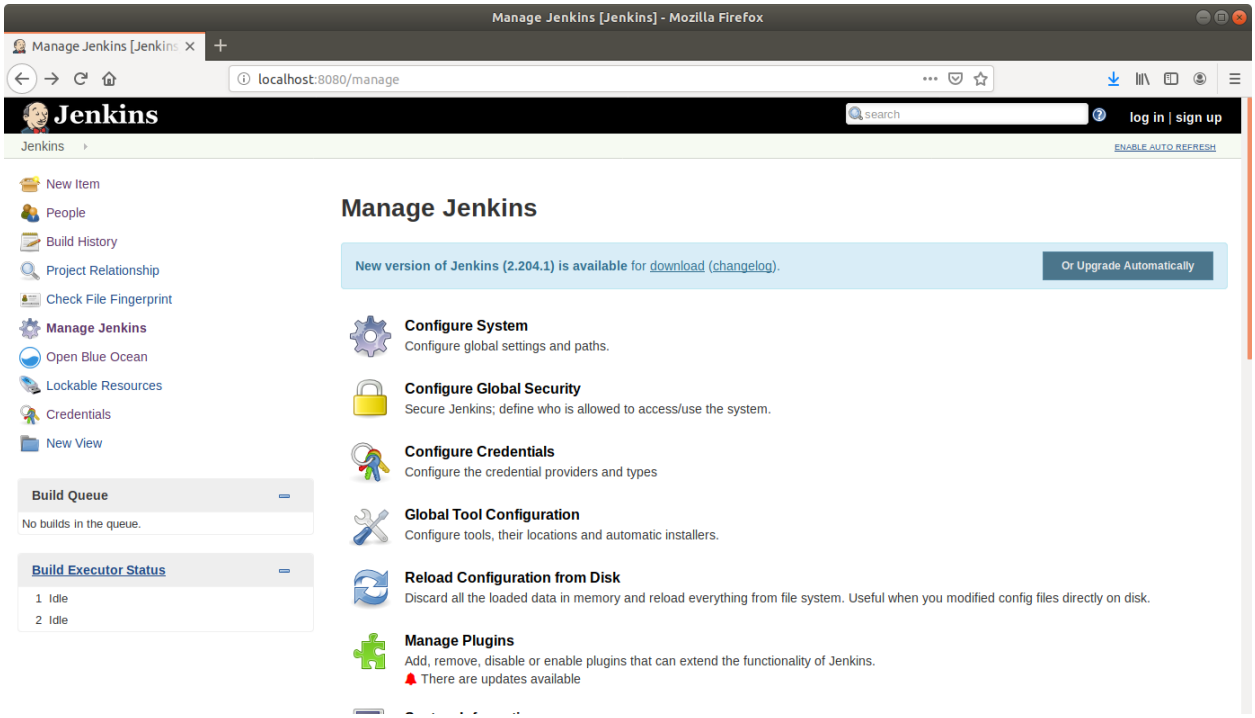
  sudo ufw allow 8080

- Navigate to http://<Public DNS (IPv4)>:8080 to view the Jenkins server.
- Follow the instructions on screen to complete installation.

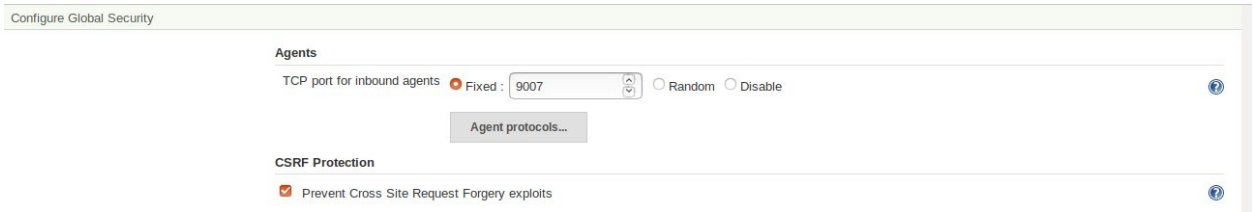


Step 6: Connecting a slave node to Jenkins master

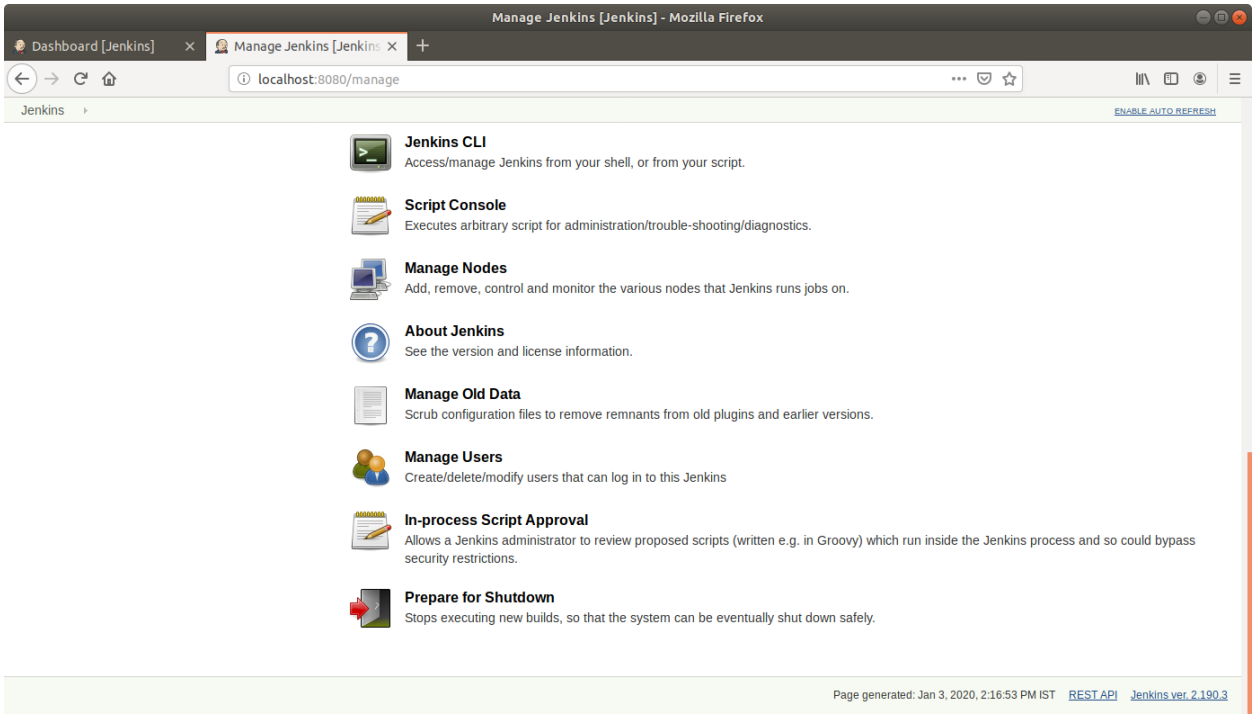
- Go to Jenkins dashboard.
- Click on *Manage Jenkins* and select *Manage Global Security.*



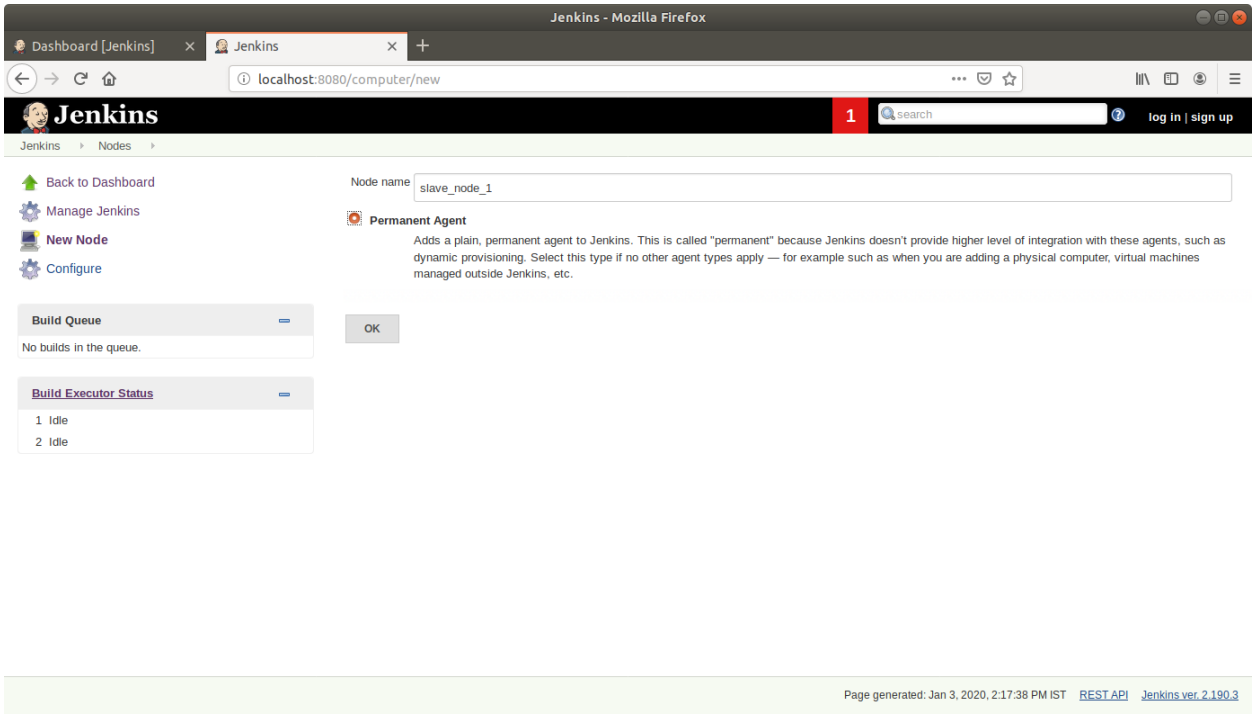
- Scroll down to Agents and enable inbound traffic at port 9007.



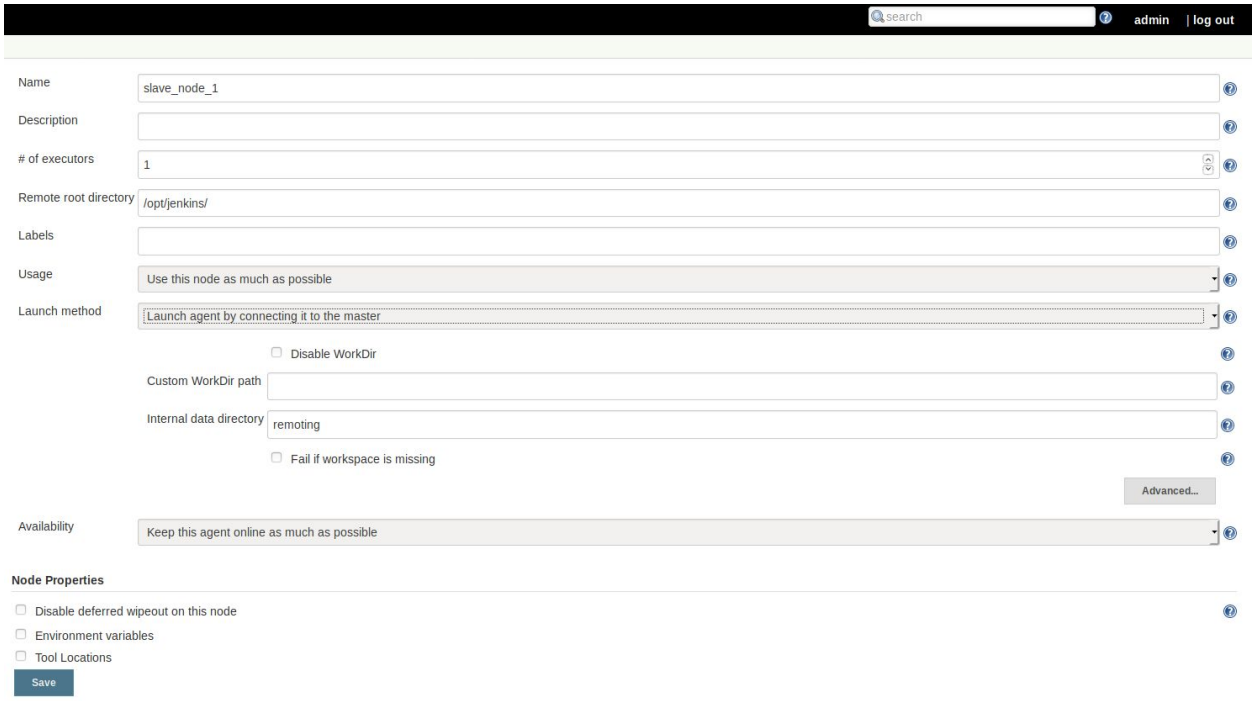
- From the Jenkins dashboard, click on *Manage Jenkins* and select *Manage Nodes*



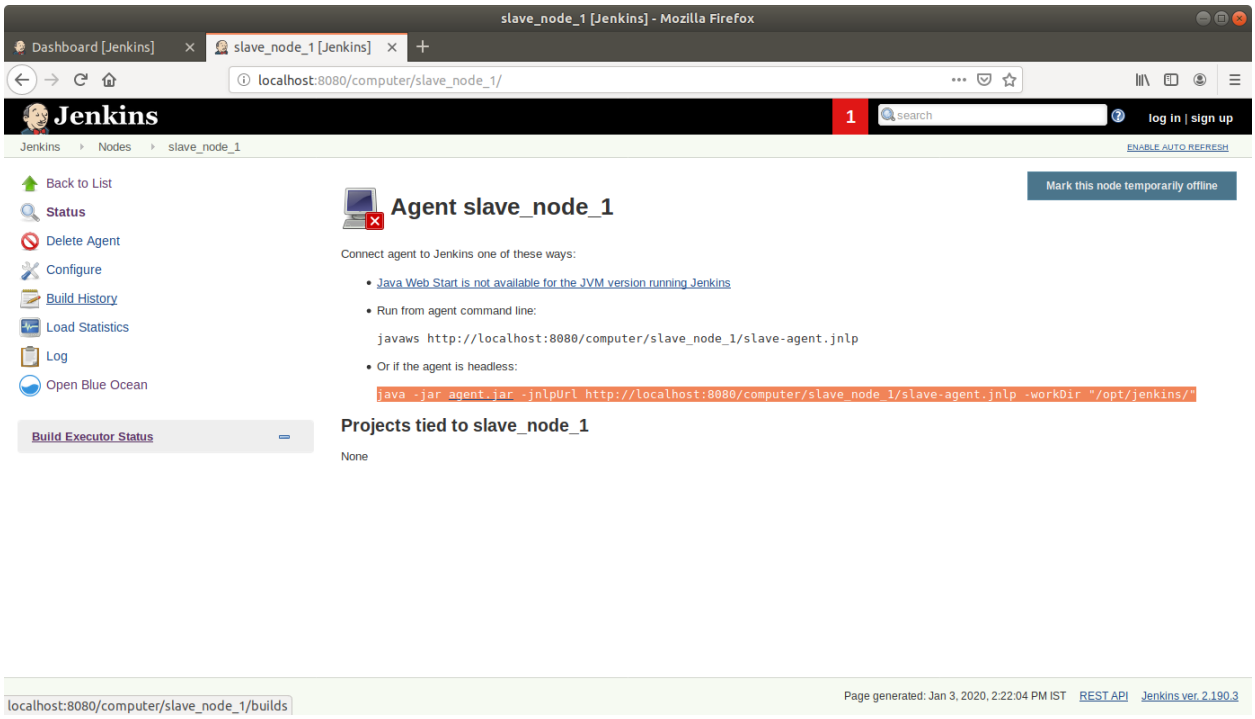
- Give a name to the node and then click OK. Select the *Permanent Agent* checkbox.



- Enter the required information as shown in the screenshot below:



- Click on the *agent.jar* hyperlink in the new page and copy the link on the new page.
- Copy the command given on the page to be run from the slave agent.

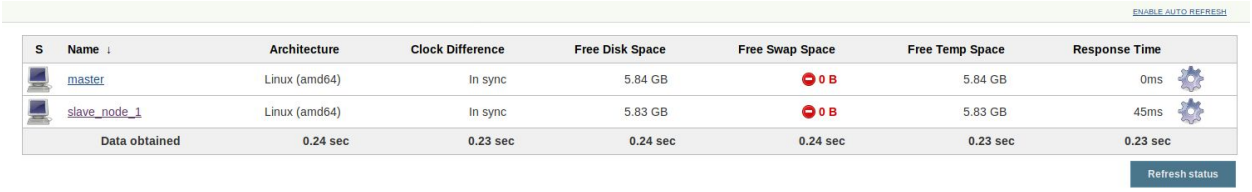


- SSH to the EC2 instance acting as the slave node.
- Create a directory at the path specified as *Remote root directory*.
- Download the agent.jar file to the newly created directory using the wget command.

- Run the command copied from the Jenkins master to connect the instance to master

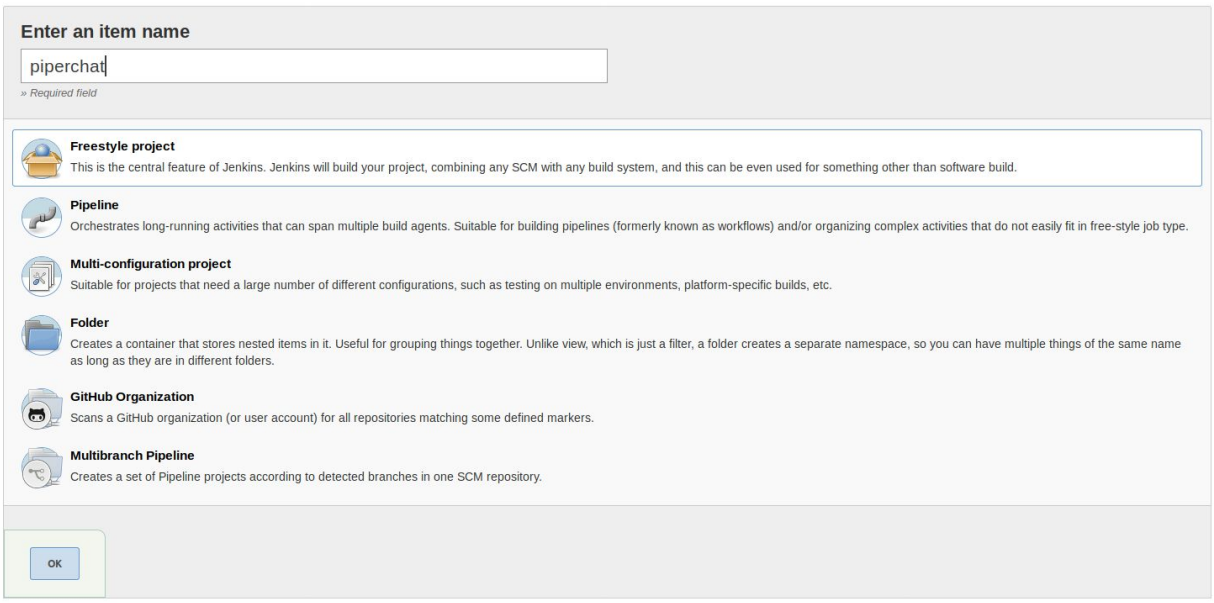
```
ubuntu@ip-172-31-5-65:~/jenkins$ sudo java -jar agent.jar -jnlpUrl http://ec2-3-14-129-49.us-east-2.compute.amazonaws.com:8080/computer/slave_node_1/slave-agent.jnlp -s
ecret 7649de7f88ffa7d9b463f57849e13af8e0952949339d9ed797cd04509097ec4e -workDir "/home/jenkins/"
Jan 03, 2020 10:50:53 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins/remoting as a remoting work directory
Jan 03, 2020 10:50:53 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins/remoting
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: slave_node_1
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Jan 03, 2020 10:50:54 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3.36
Jan 03, 2020 10:50:54 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins/remoting as a remoting work directory
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://ec2-3-14-129-49.us-east-2.compute.amazonaws.com:8080/]
Jan 03, 2020 10:50:54 AM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: ec2-3-14-129-49.us-east-2.compute.amazonaws.com
  Agent port:    9007
  Identity:      49:2d:d6:70:d6:f6:68:c9:74:7e:40:2a:82:48:49:15
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to ec2-3-14-129-49.us-east-2.compute.amazonaws.com:9007
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Jan 03, 2020 10:50:54 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 49:2d:d6:70:d6:f6:68:c9:74:7e:40:2a:82:48:49:15
Jan 03, 2020 10:50:55 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

- The slave node will show as connected to the master UI.

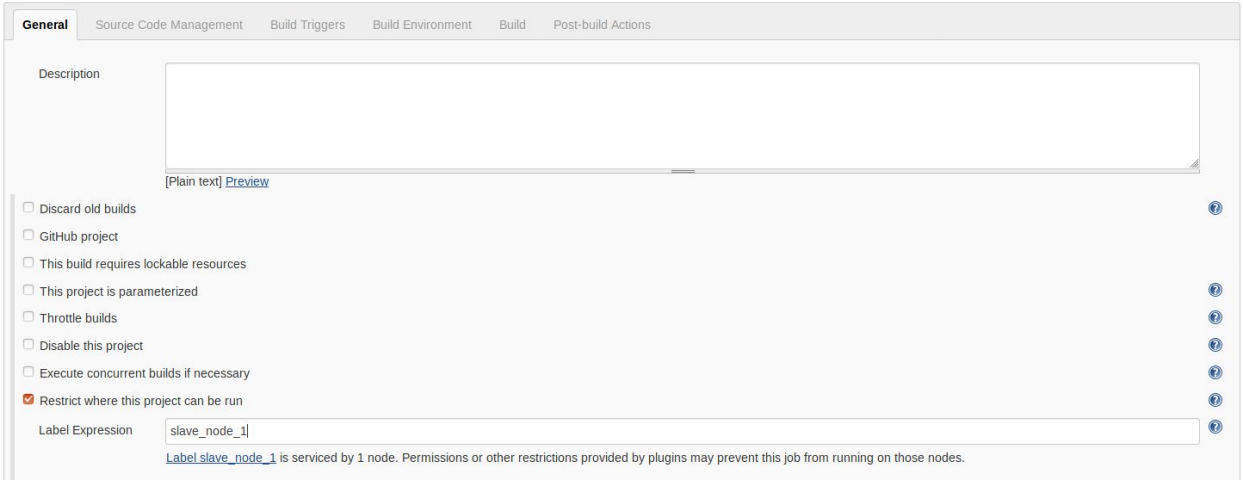


Step 7: Creating a build pipeline in Jenkins

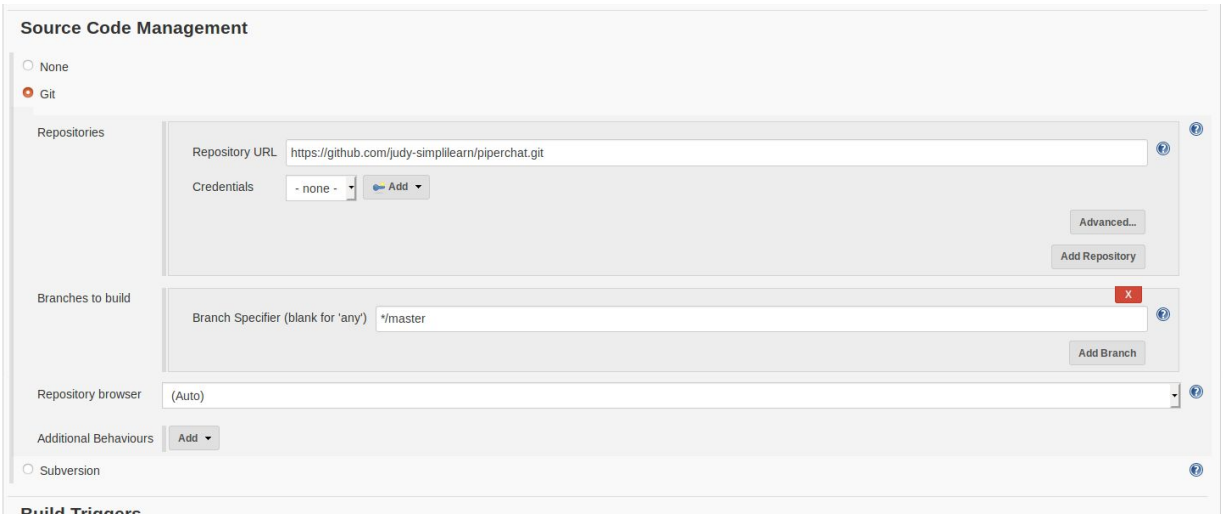
- Click on *New Item*.
- Enter a name for your build job.
- Select *Freestyle Job* as the build job type.



- Click OK.
- Under *General* tab on the configuration page, check the box that reads "*Restrict where this build will be run*" and enter the slave node's name.



- Scroll down to the Source Code Management section.
- Select *Git in SCM.*
- Add the repository URL.



- Drag to the bottom and go to the *Build steps* section.
- Select on *Execute Shell command* from the drop-down.
- Enter the following command in the textbox:
  mvn clean package

- Click *Save*.

Step 7: Running a deployment pipeline in Jenkins

- Click on *Build Now* in the project window.
- Jenkins will now build your pipeline and output the logs.
- The logs will show that the job is being run on the slave node.

Console Output

Progress:

```
Started by user admin
Running as SYSTEM
Building remotely on slave_node_1 in workspace /home/jenkins/workspace/piperchat
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/judy-simplilearn/piperchat.git
 > git init /home/jenkins/workspace/piperchat # timeout=10
Fetching upstream changes from https://github.com/judy-simplilearn/piperchat.git
 > git --version # timeout=10
 > git fetch --tags --progress -- https://github.com/judy-simplilearn/piperchat.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/judy-simplilearn/piperchat.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/judy-simplilearn/piperchat.git # timeout=10
Fetching upstream changes from https://github.com/judy-simplilearn/piperchat.git
 > git fetch --tags --progress -- https://github.com/judy-simplilearn/piperchat.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 19d11a18d59043a7ff7f04f53657163fe818d800 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 19d11a18d59043a7ff7f04f53657163fe818d800 # timeout=10
Commit message: "Files Added"
First time build. Skipping changelog.
[piperchat] $ /bin/sh -xe /tmp/jenkins3986448176009877108.sh
+ mvn clean package
```