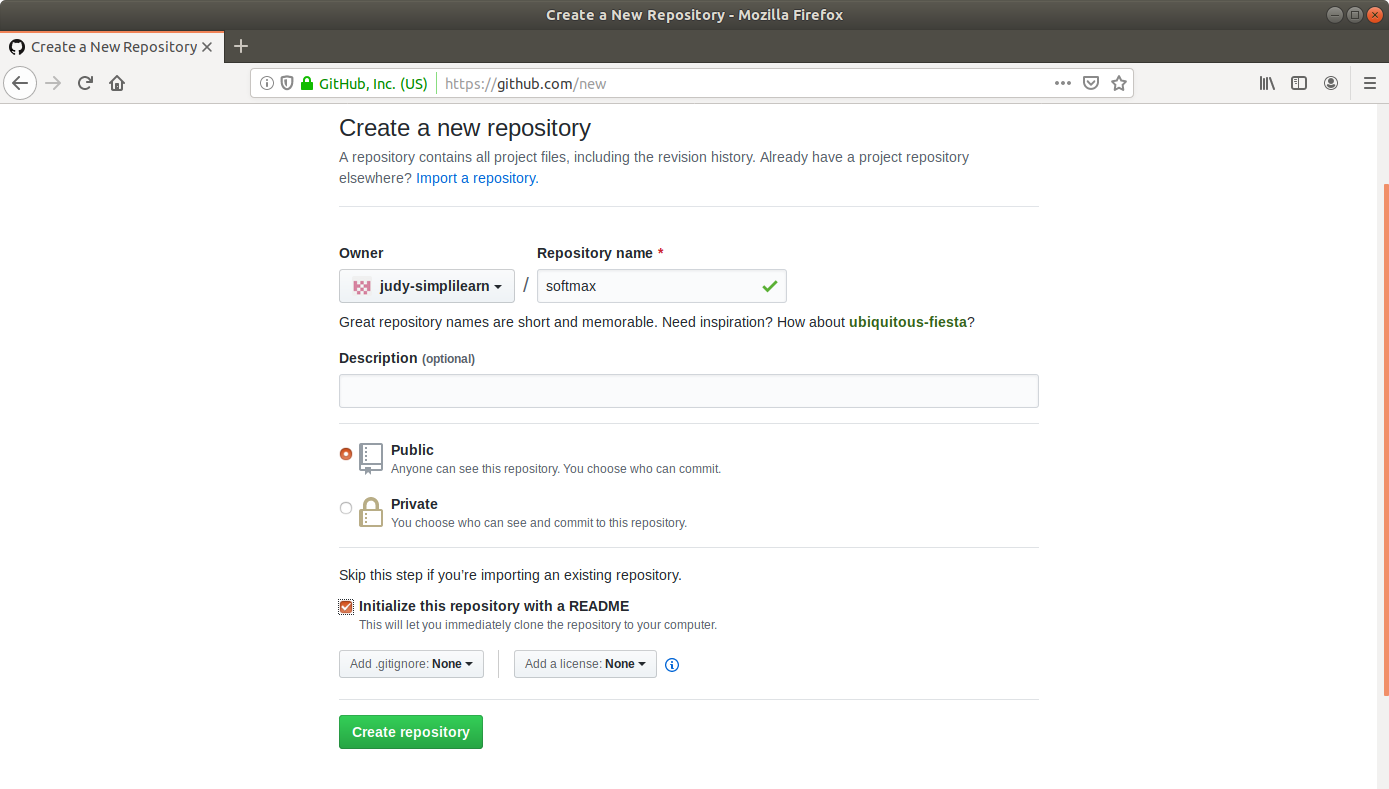
**Architecting Jenkins Pipeline for Scale**

**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.



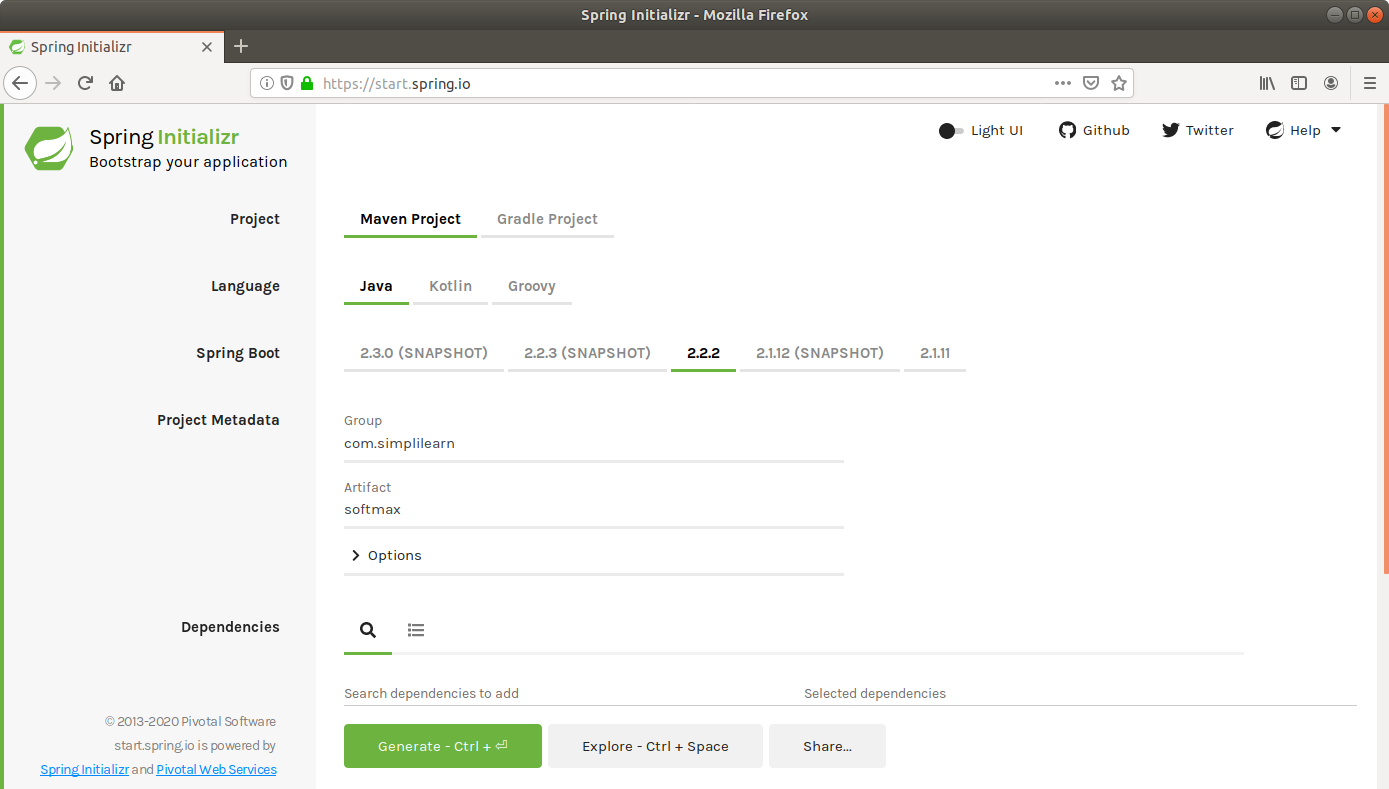
* Fill the required fields in the create repository form.



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

**Step 2:** Generating a spring boot project

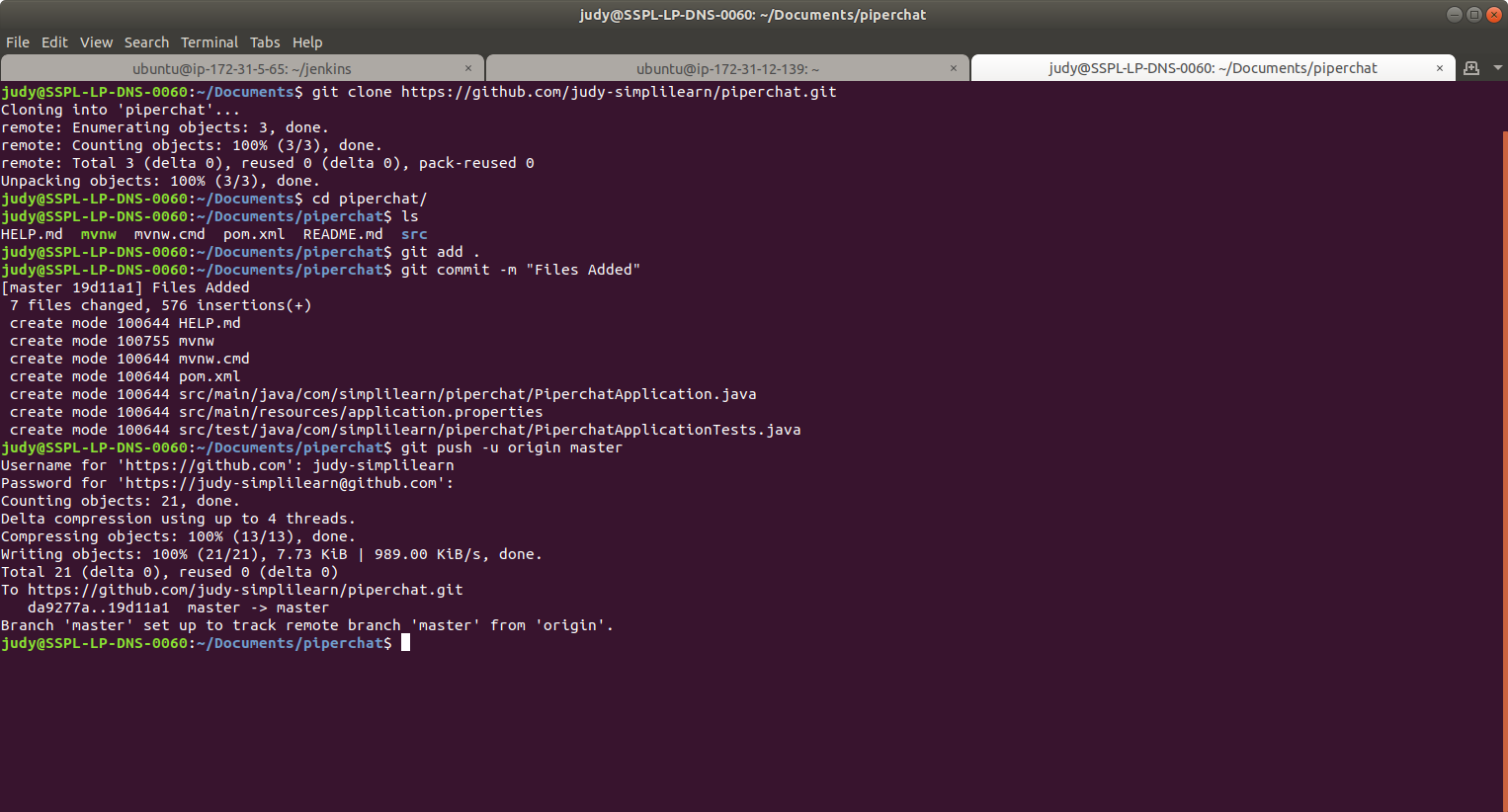
* Go to ​start.​spring.​io/​



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

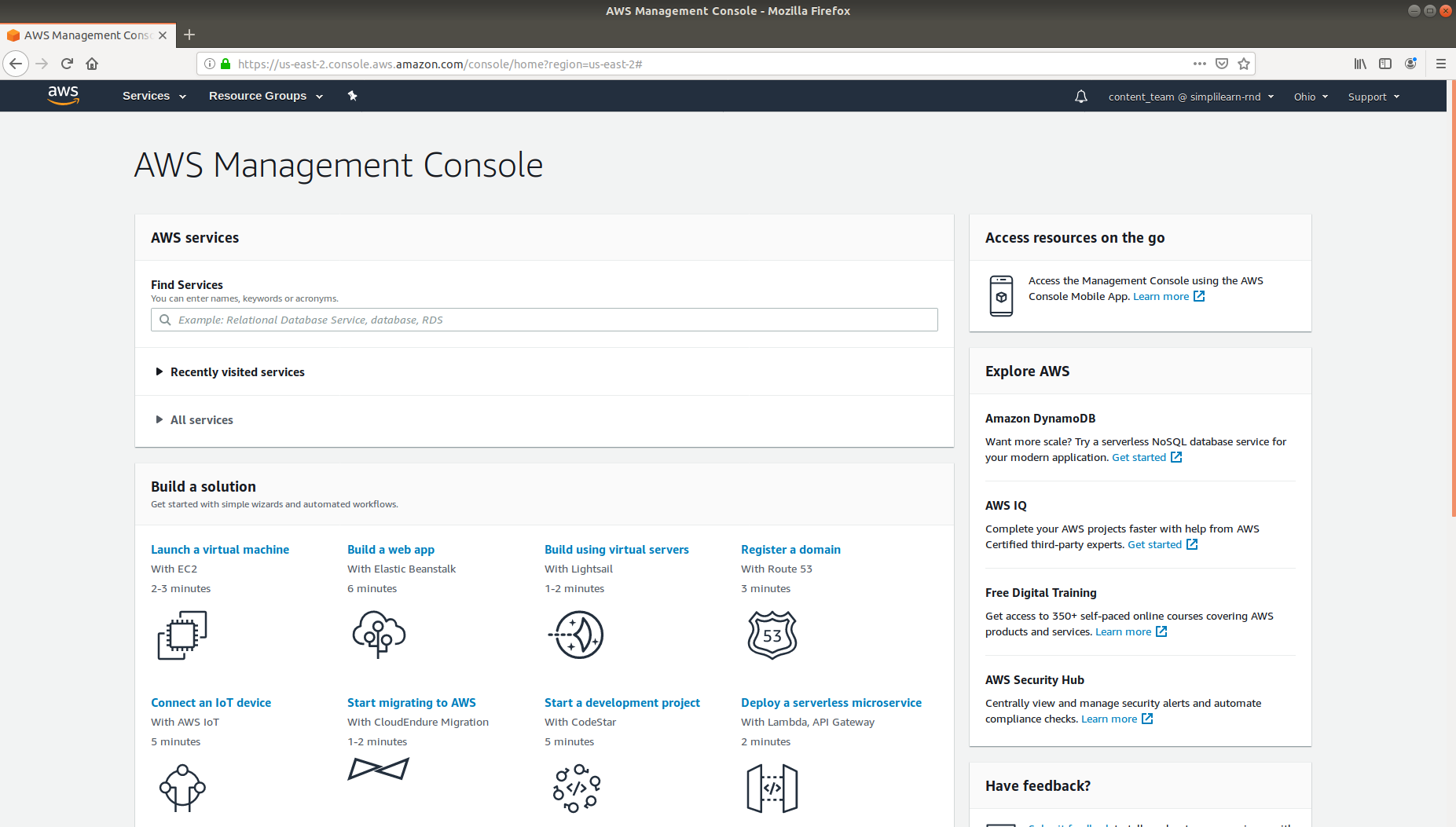
**Step 3:** Committing the project skeleton to the repository

* 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**

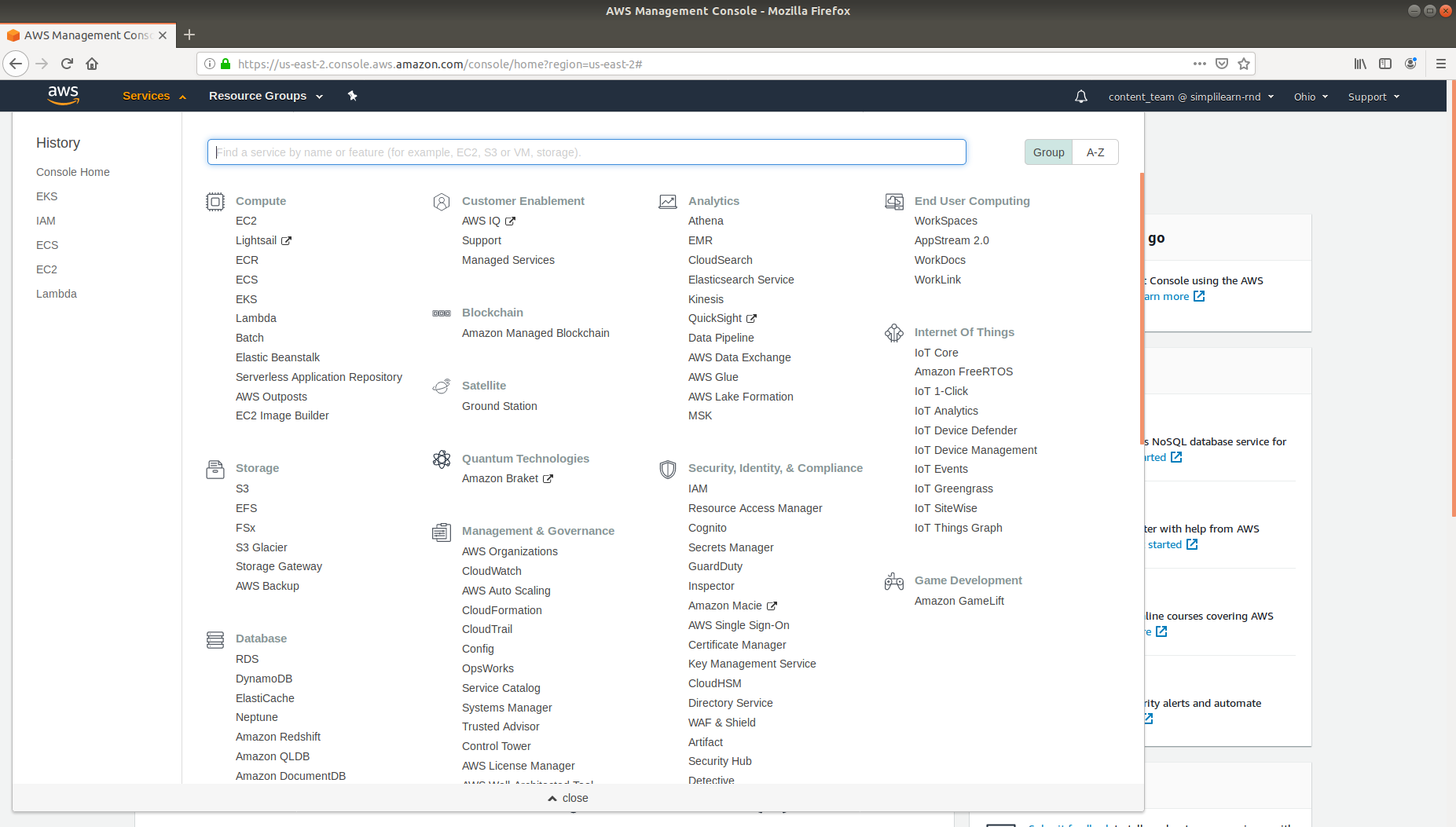


**Step 4:** Creating an EC2 instance

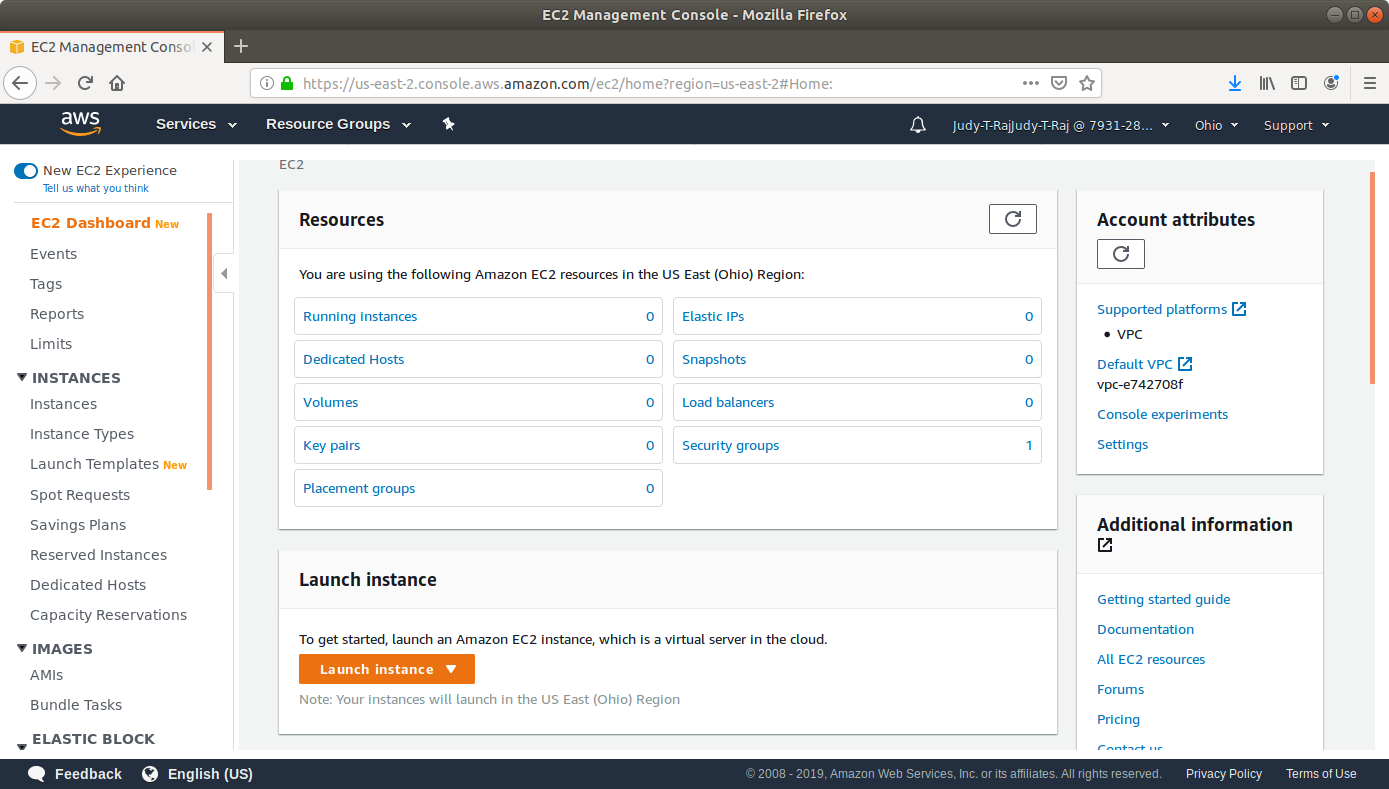
* Log in to the AWS lab account provided. You will be able to see the following screen:



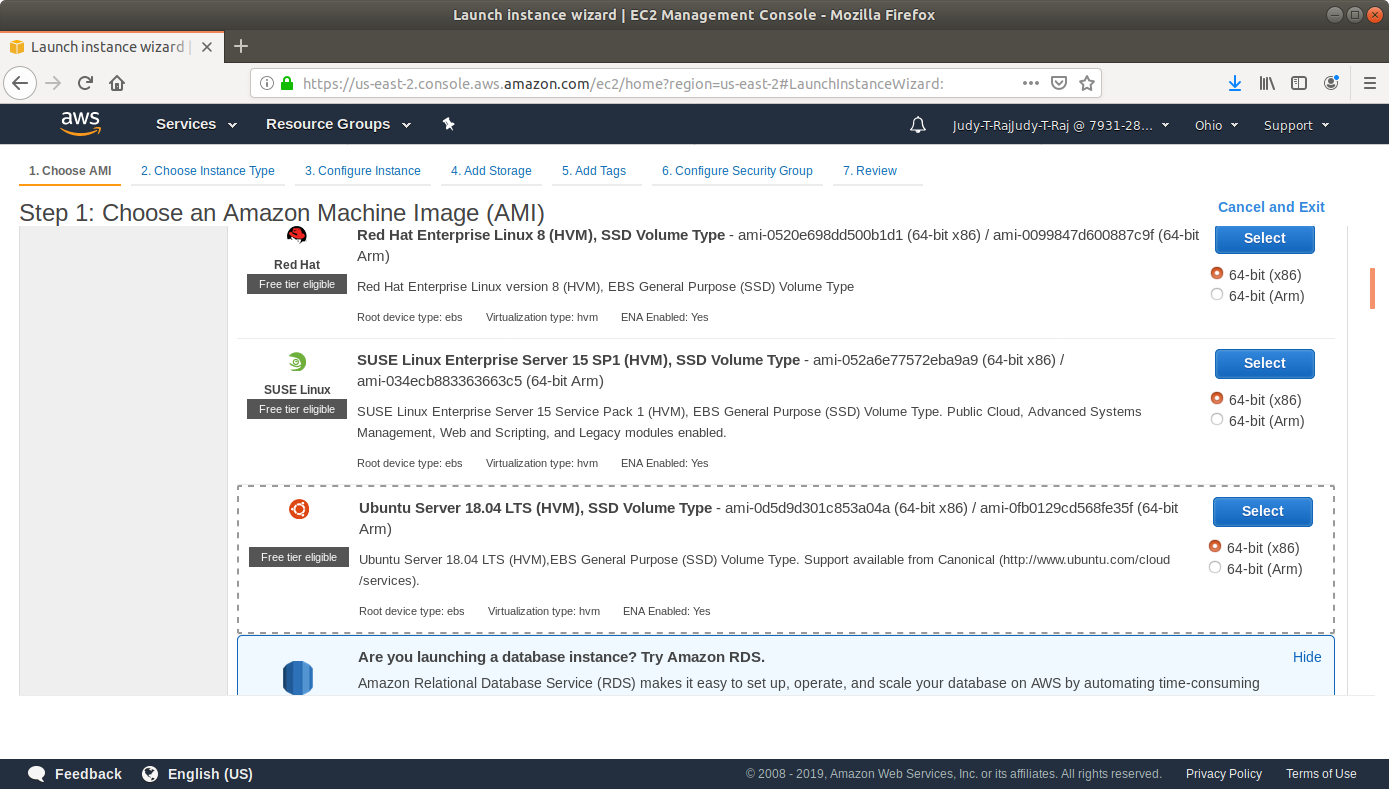
* Click on Services at the top left to view the drop-down list of resources.
* Click on EC2 under the Compute menu from the drop-down list.



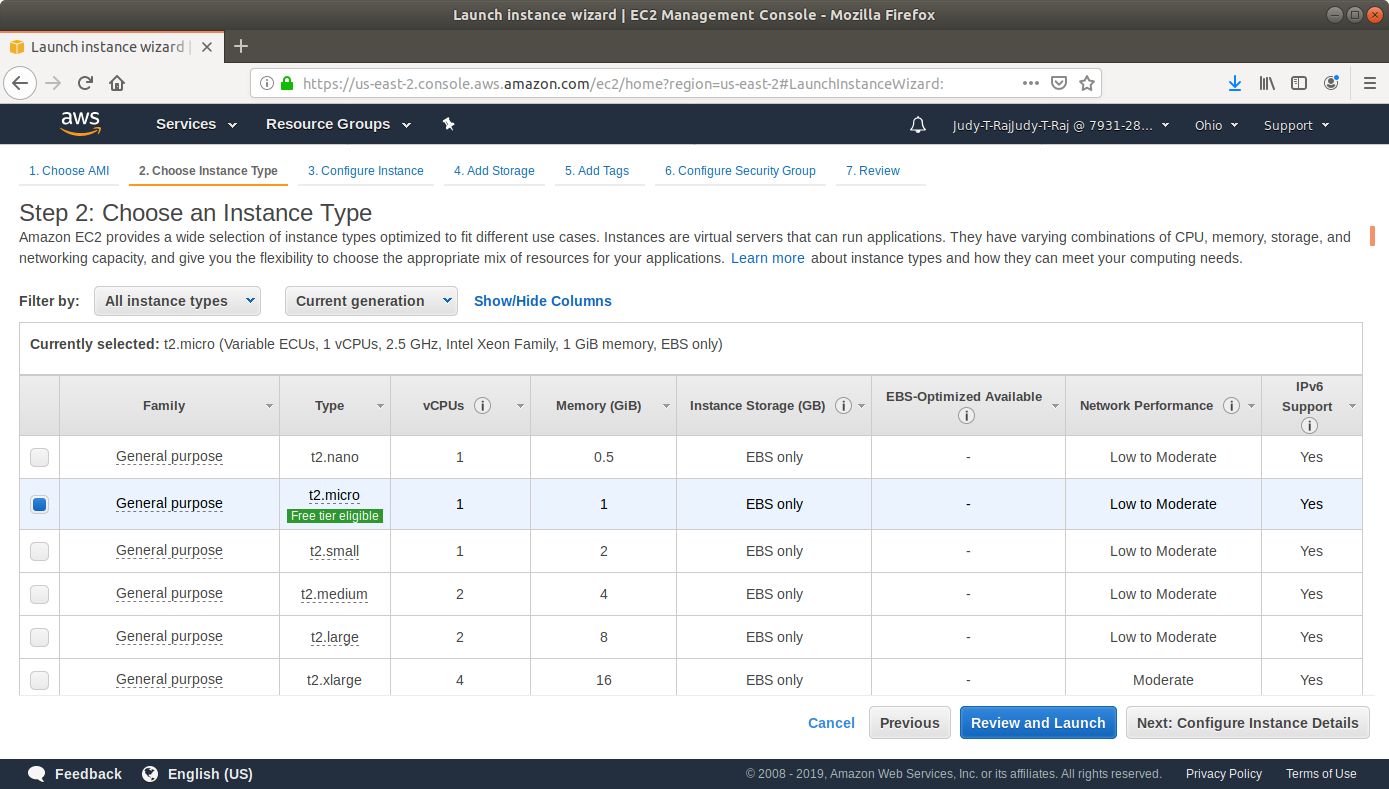
* Click on *Launch Instance* button and select Launch Instance from the menu.



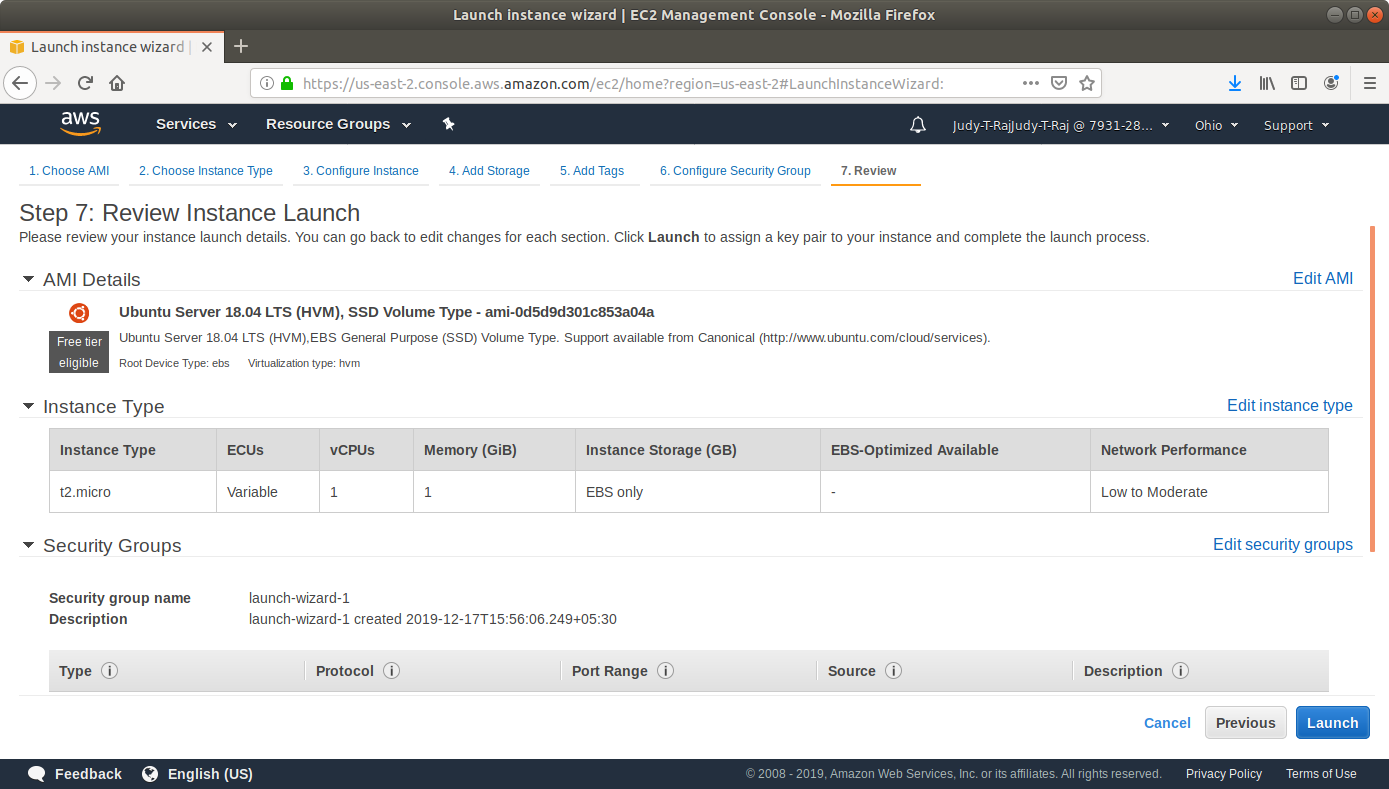
* Choose an Amazon Machine Image (AMI) from the list of AMIs and click on Select.



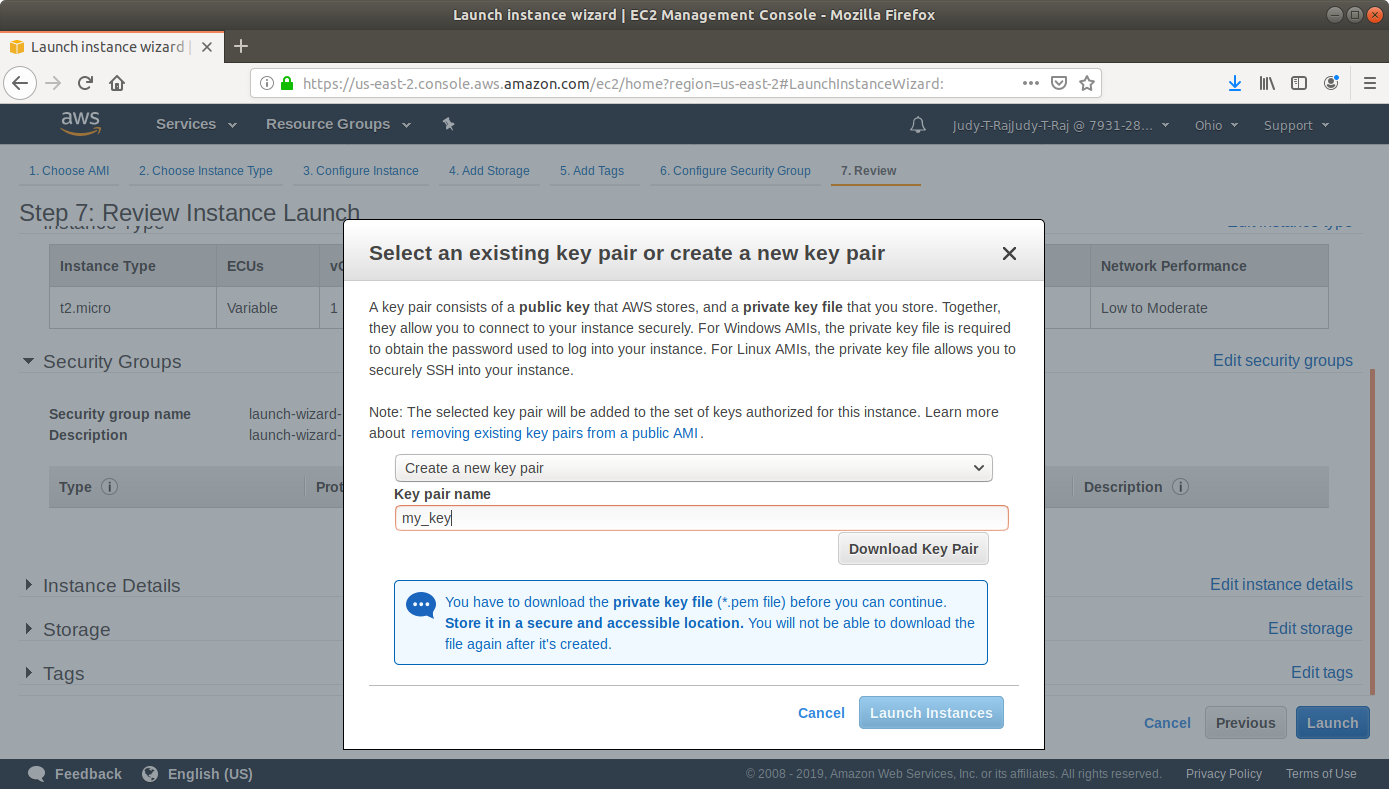
* Choose an Instance Type and click *Review and Launch*.



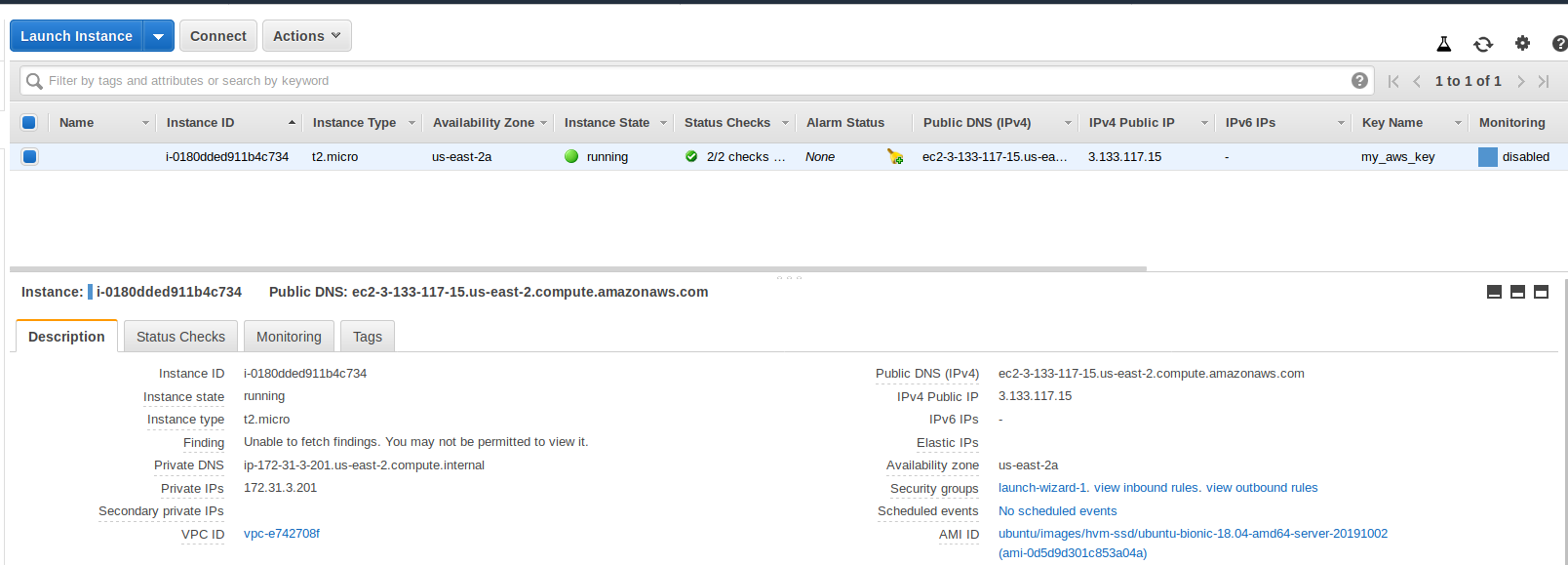
* Click on *Launch*.



* 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*.
* Repeat the same steps to create two more instances.
* 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

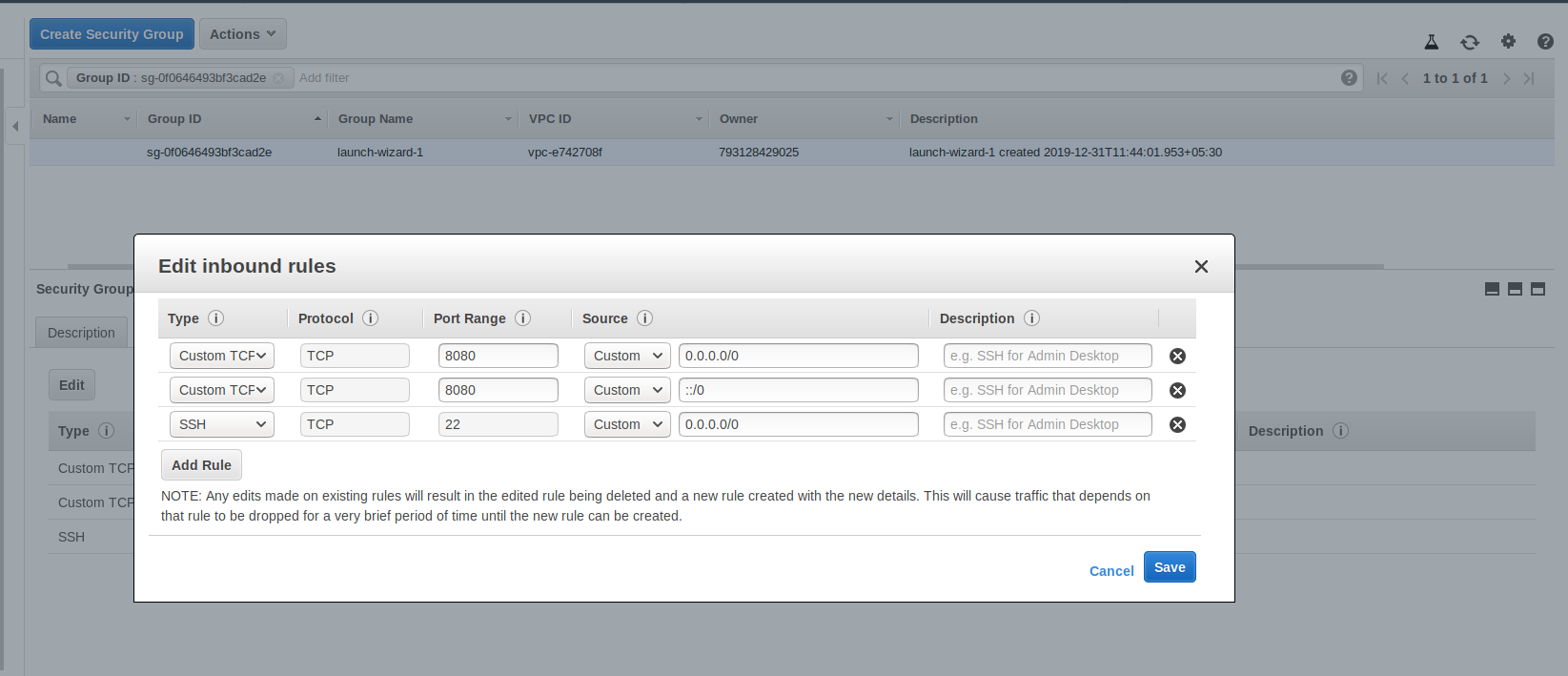
* Add a rule to the security group to which the instance belongs to allow HTTP traffic to port 8081 with the following settings:

**Type:** Custom TCP Rule

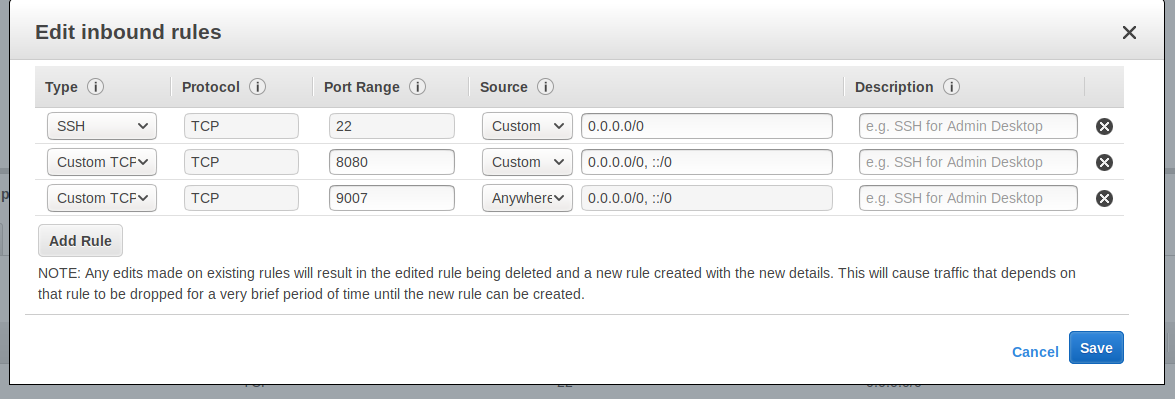
**Protocol:** TCP

**Port Range:** 8081

**Source:** Anywhere 0.0.0.0/0

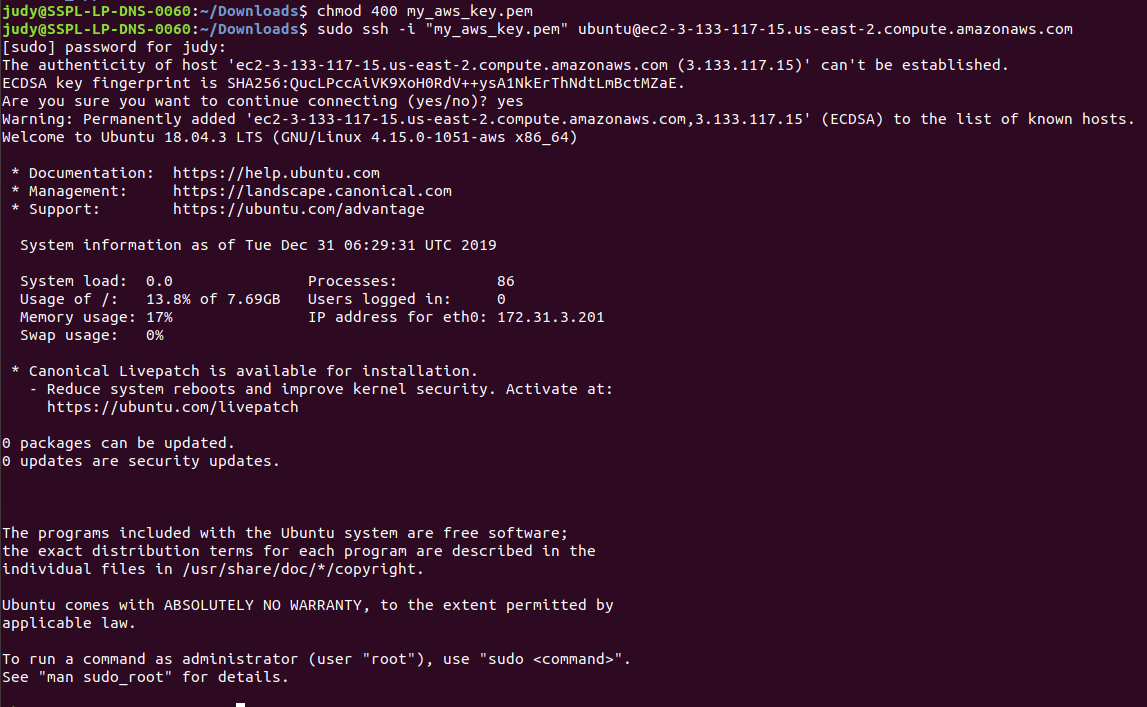


* Add the same rules to the Security Groups to which the other two VMs belong. For the master node, add an inbound rule that allows traffic to port 9007.



**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>**



* 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 -**

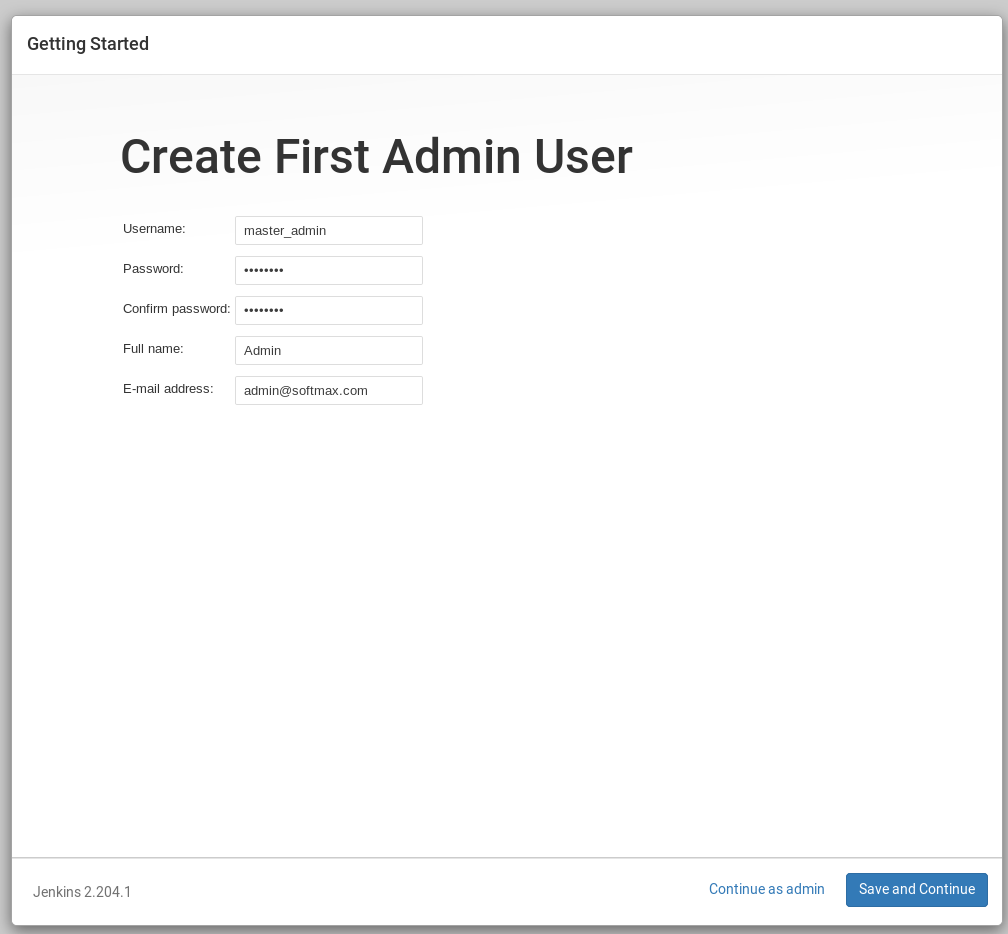
**sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'**

**sudo apt update**

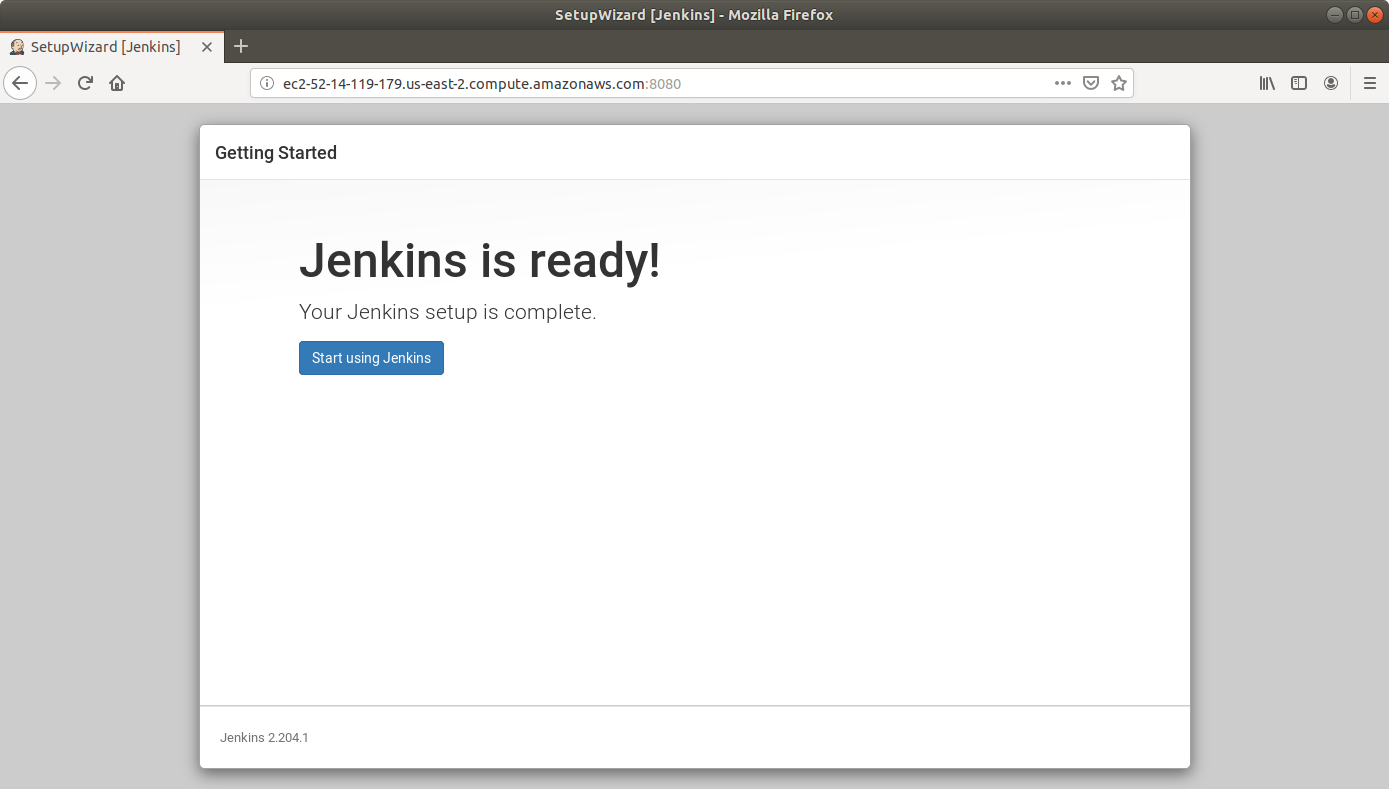
**sudo apt install jenkins**

**sudo ufw allow 8081**

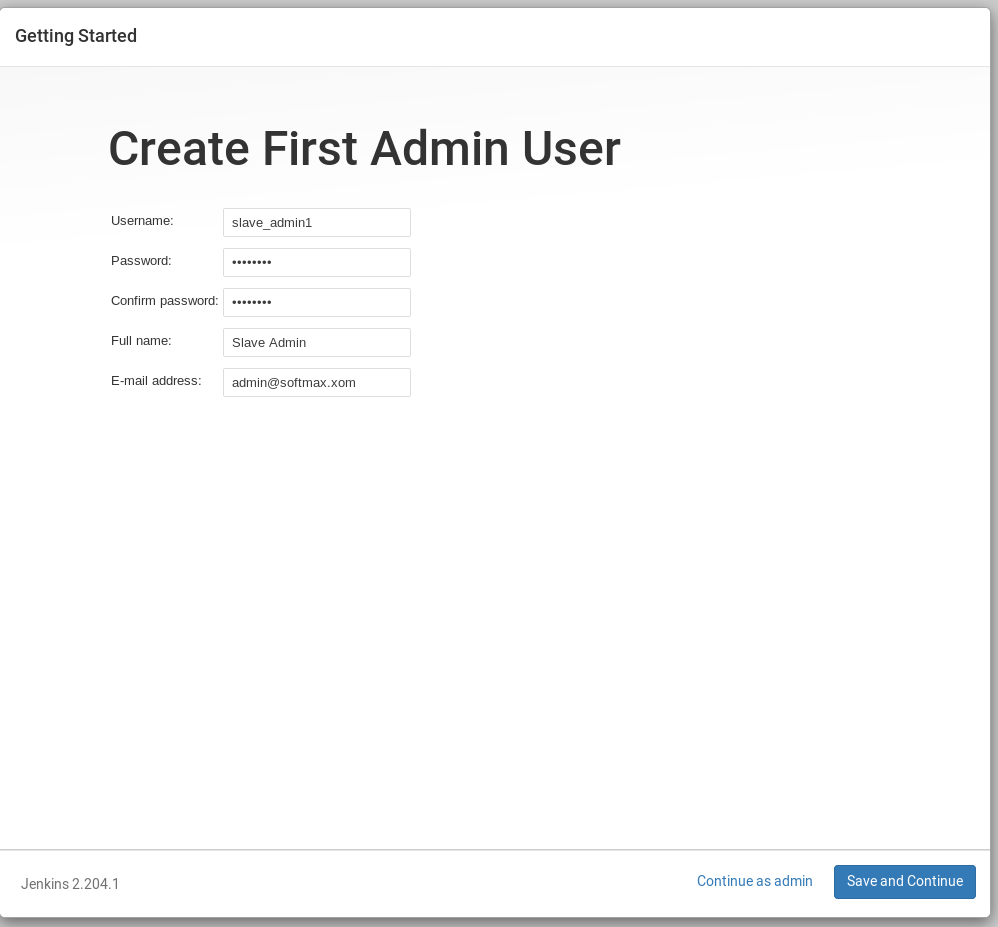
* Navigate to http://<Public DNS (IPv4)>:8081 to view the Jenkins server.
* Set up Admin User and install plugins.

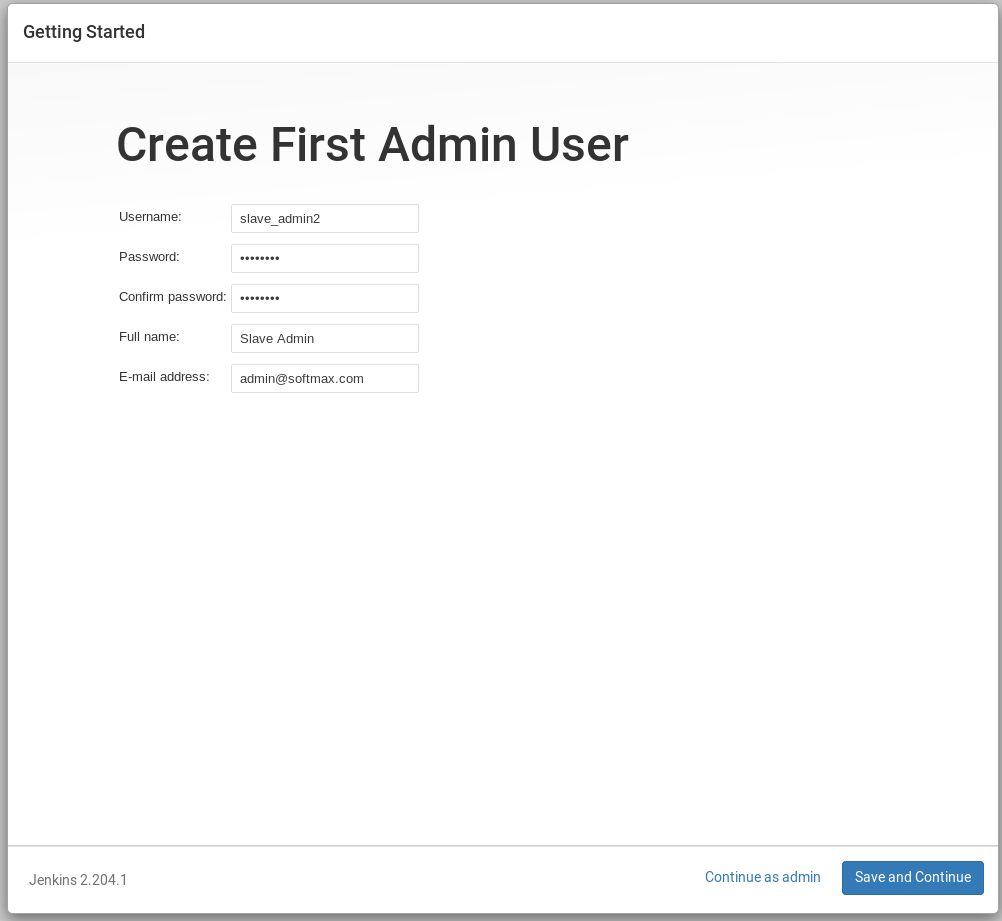


* Follow the instructions on screen to complete installation.

****

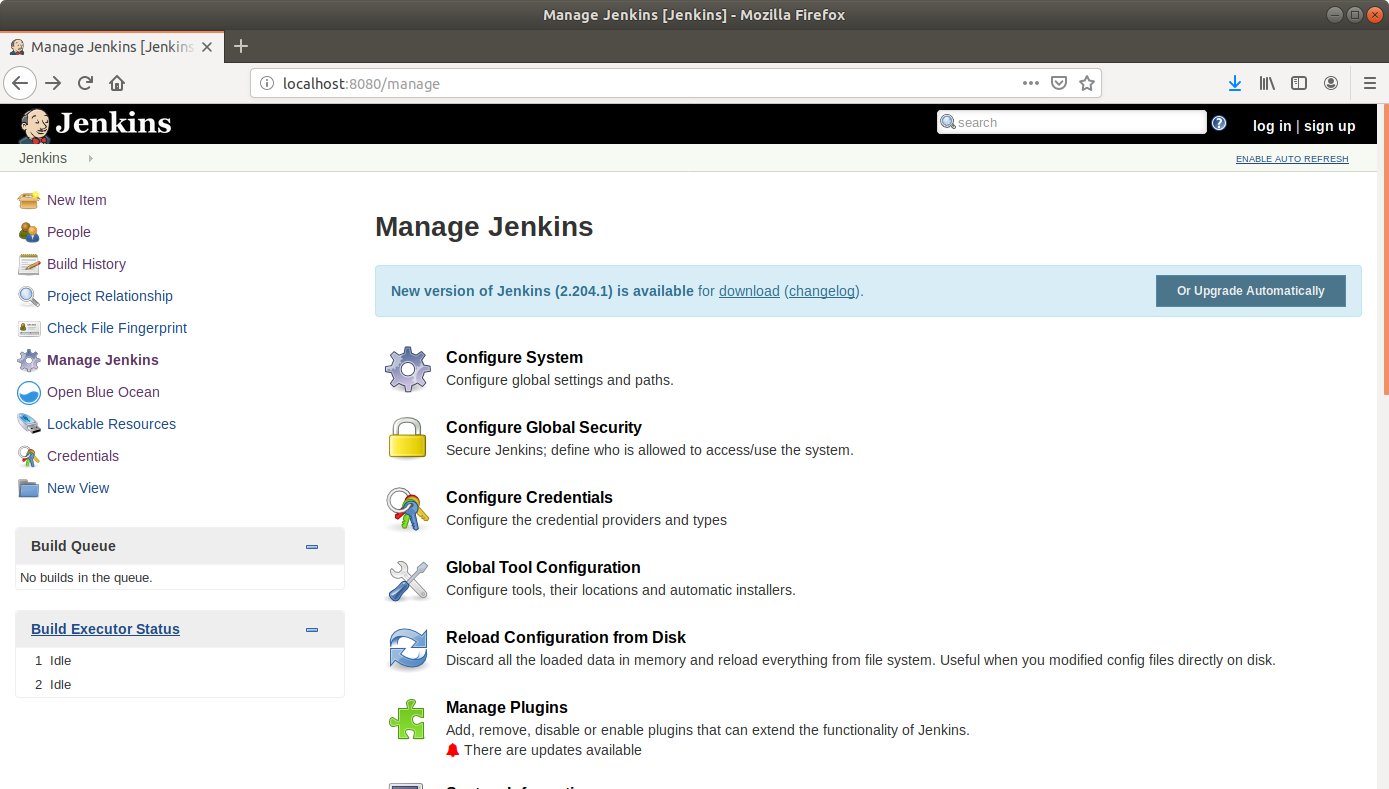
* Repeat the same steps to set up Jenkins on the other two nodes.



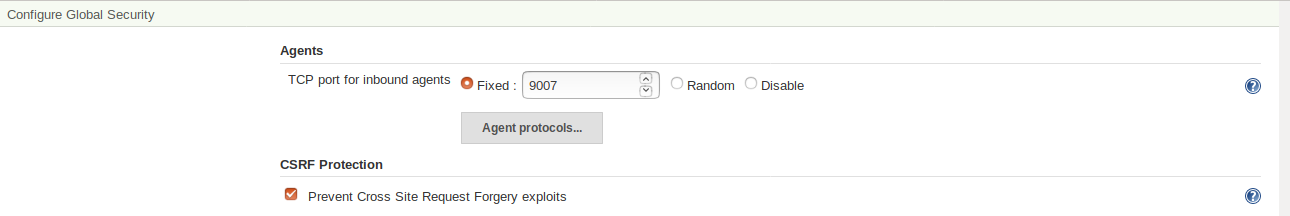


**Step 6:** Connecting a slave node to Jenkins master

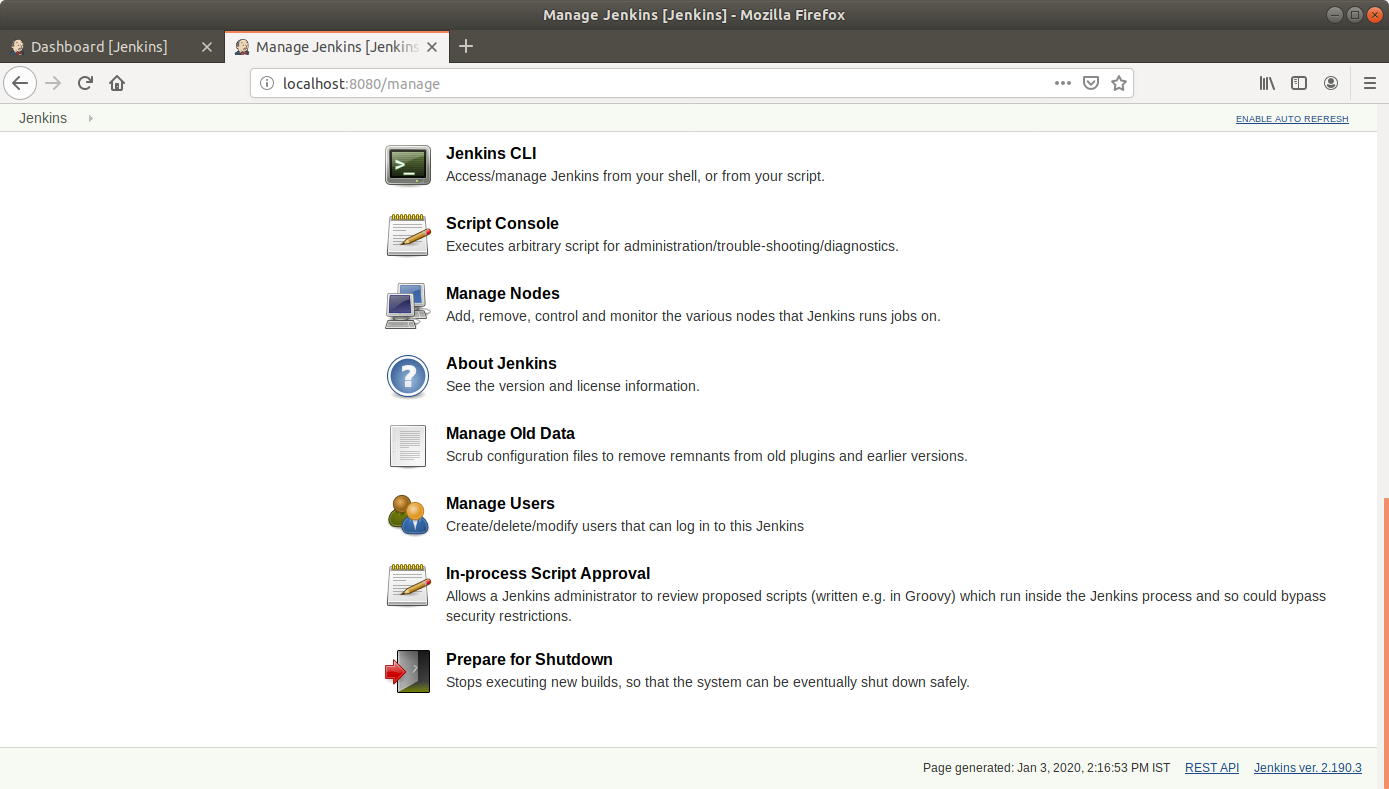
* Go to Jenkins dashboard on the master node.
* Click on *Manage Jenkins* and select *Manage Global Security.*

**

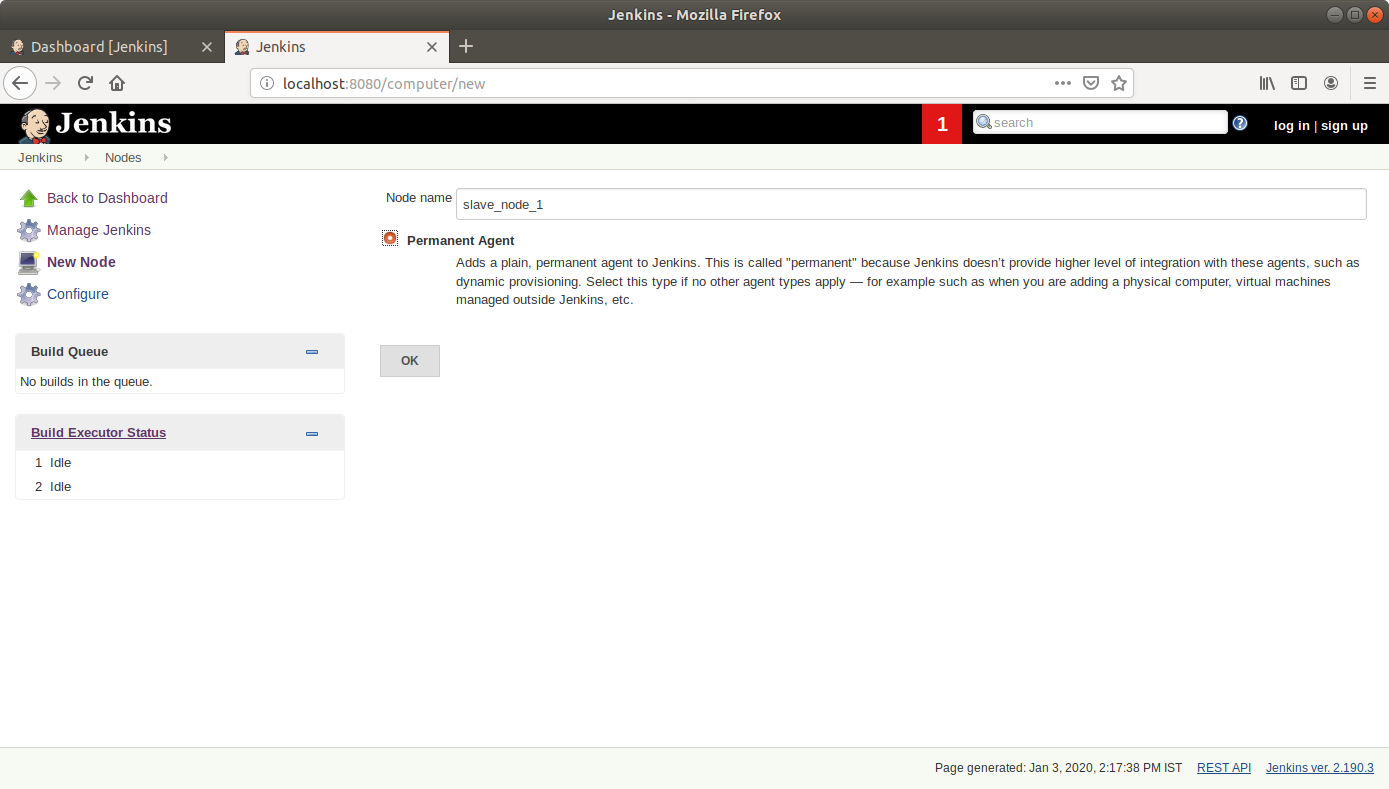
* Scroll down to Agents and enable inbound traffic at port 9007.

**

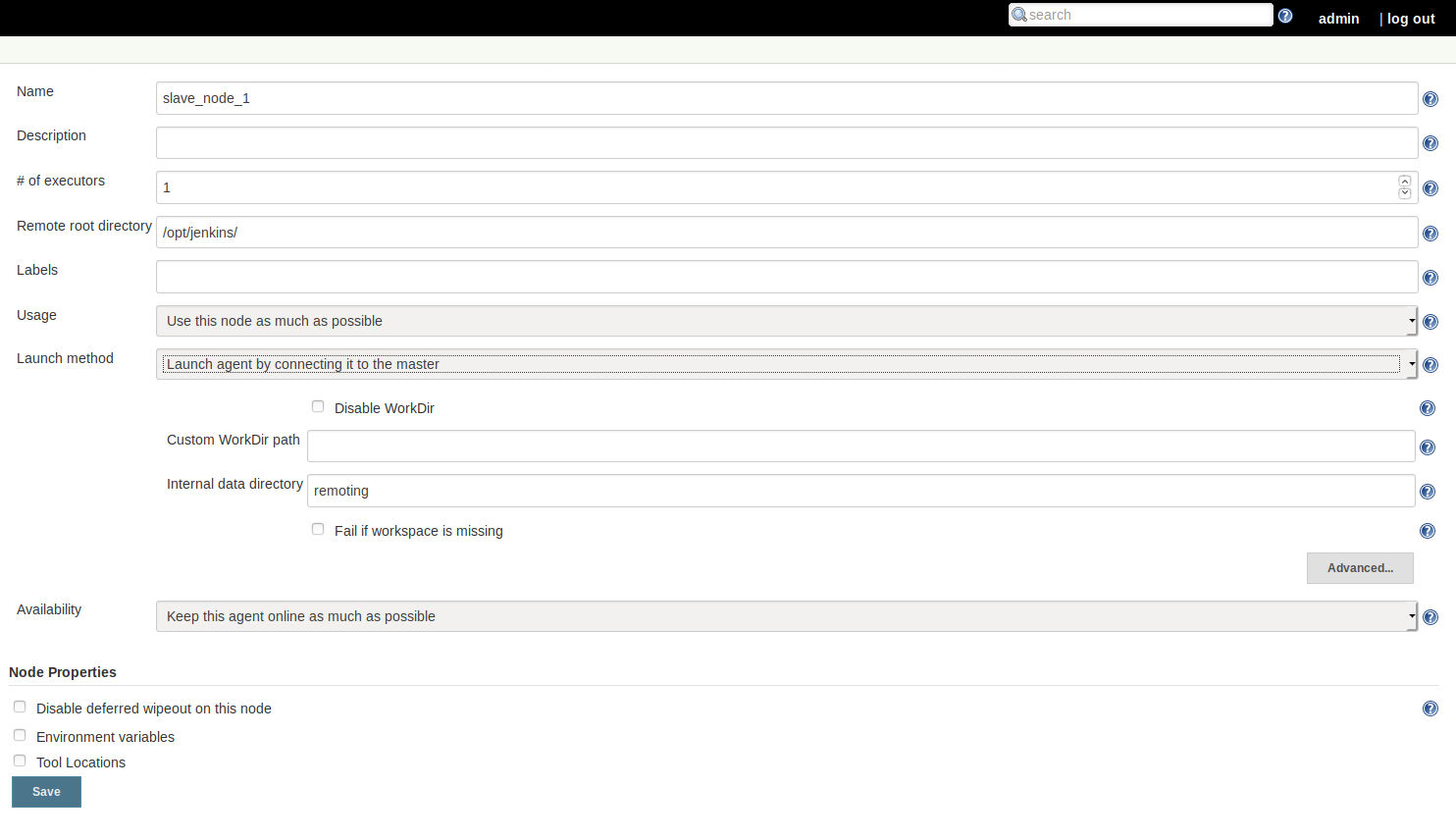
* From the Jenkins dashboard, click on *Manage Jenkins* and select *Manage Nodes.*

**

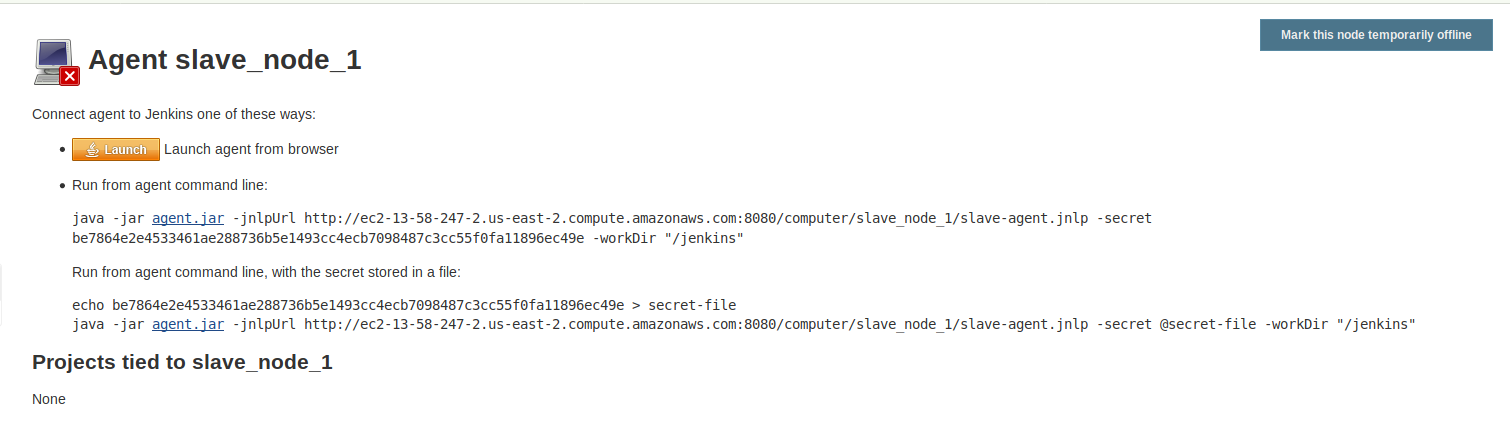
* Give a name to the node and then click OK. Select the *Permanent Agent* checkbox.



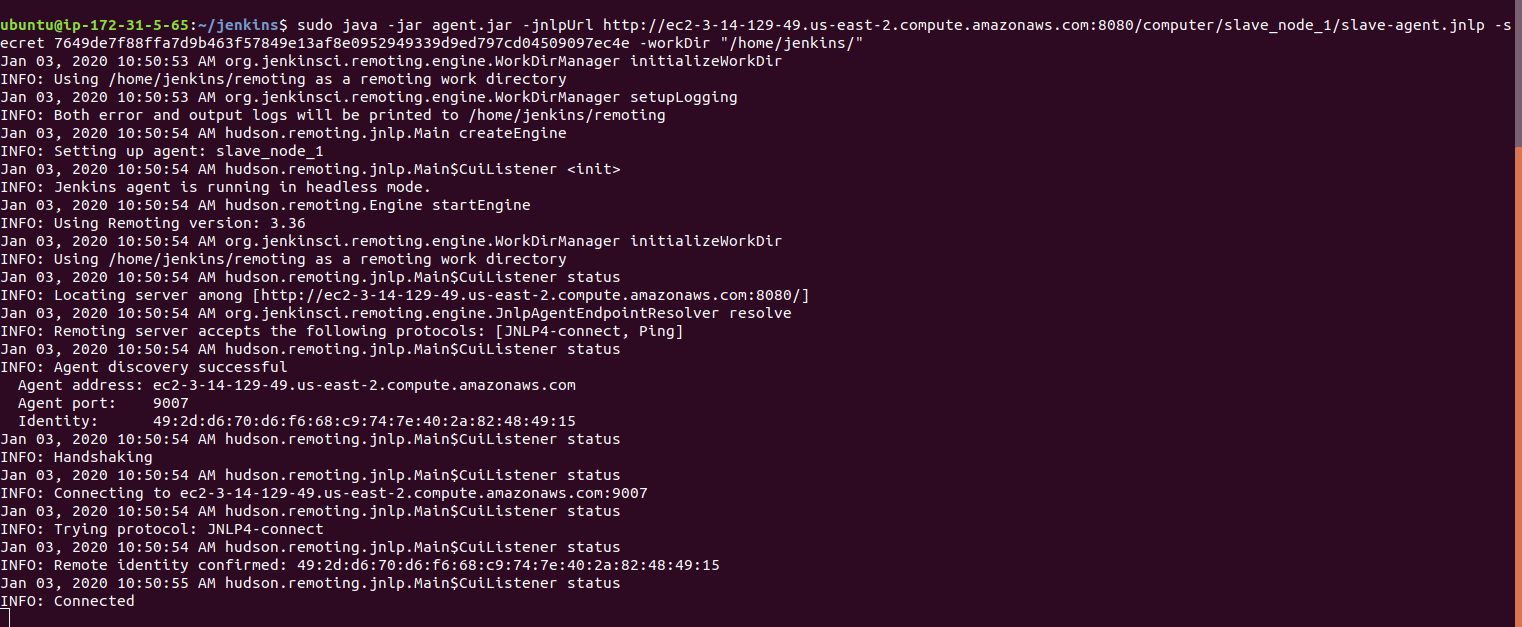
* Enter the required information as shown in the screenshot below:



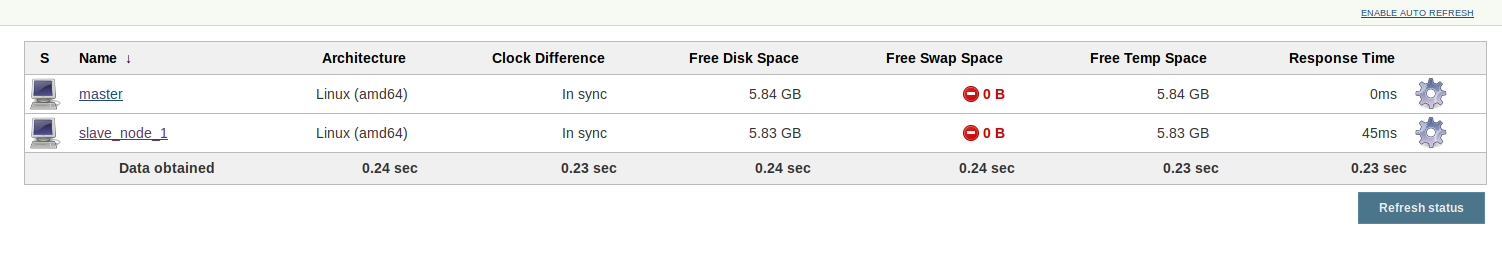
* Click on the *agent.ja*r hyperlink in the new page and copy the link on the new page.
* Copy the command given on the page to be run from the slave agent.



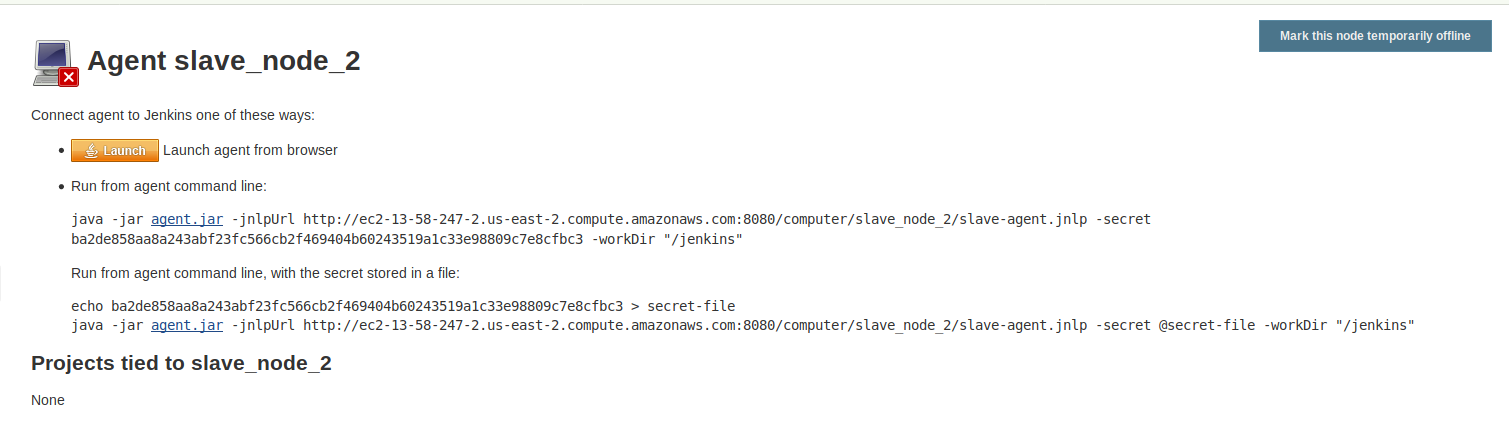
* SSH to the EC2 instance acting as the slave node 1.
* Create a directory at the path specified as *Remote root directory*.
* Download the agent.jar file to the newly created directory using the wget command.
* Run the command copied from the Jenkins master to connect the instance to master

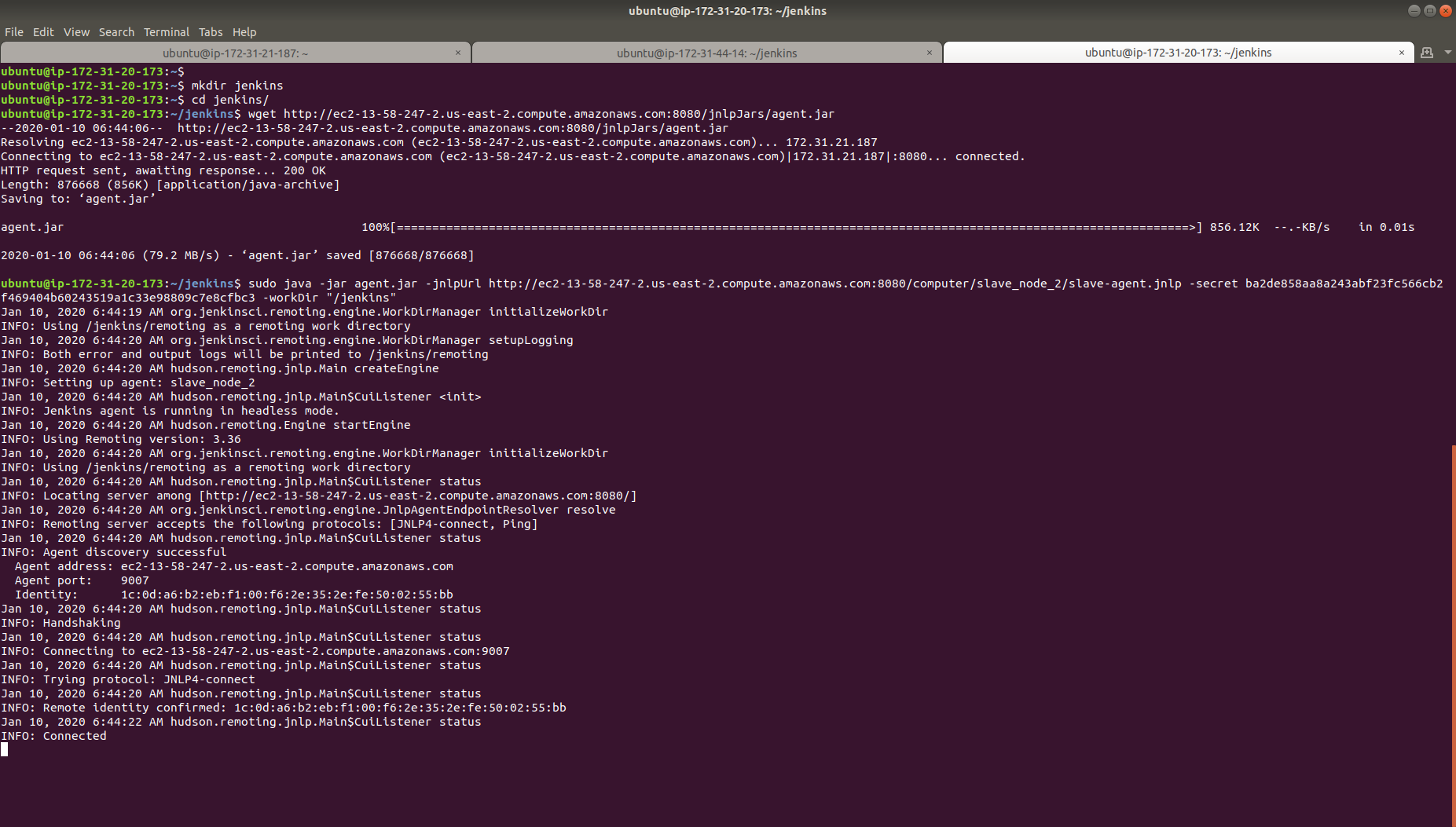


* The slave node will show as connected to the master UI.

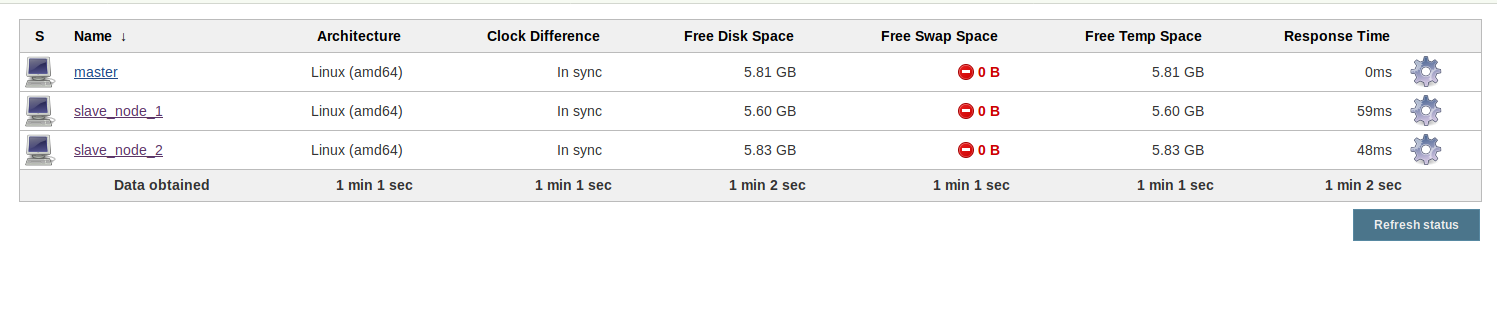


* Repeat the same steps for the second slave node.



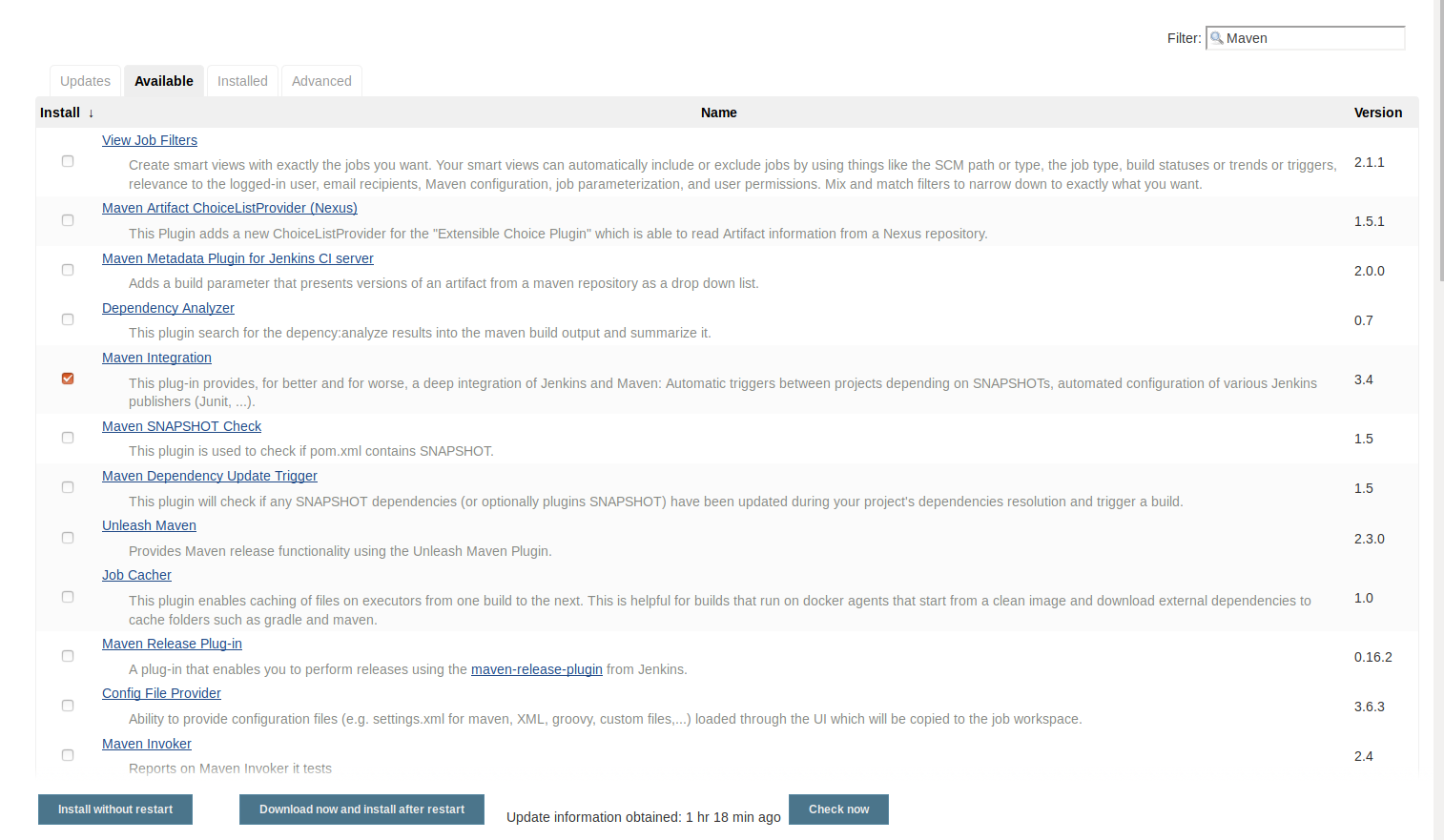


* The master node UI will now reflect two slave nodes.

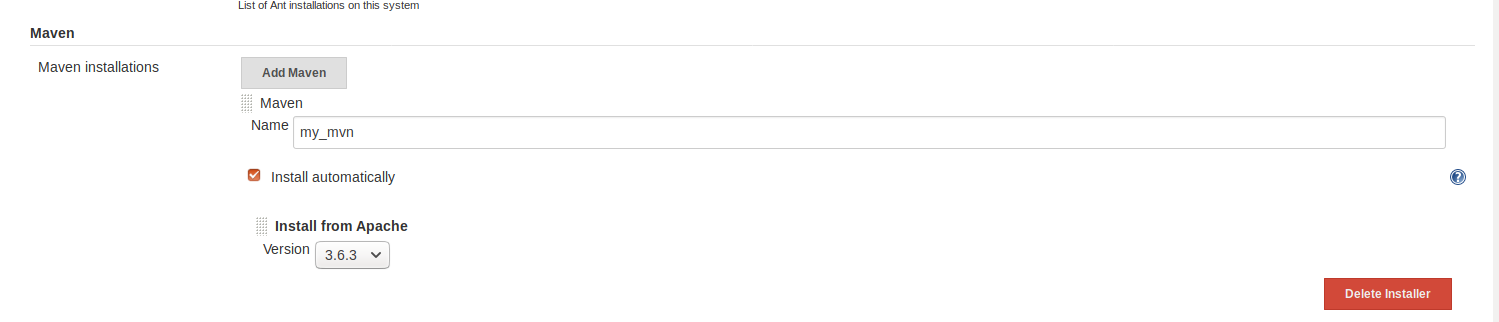


**Step 7:** Creating a build pipeline in Jenkins

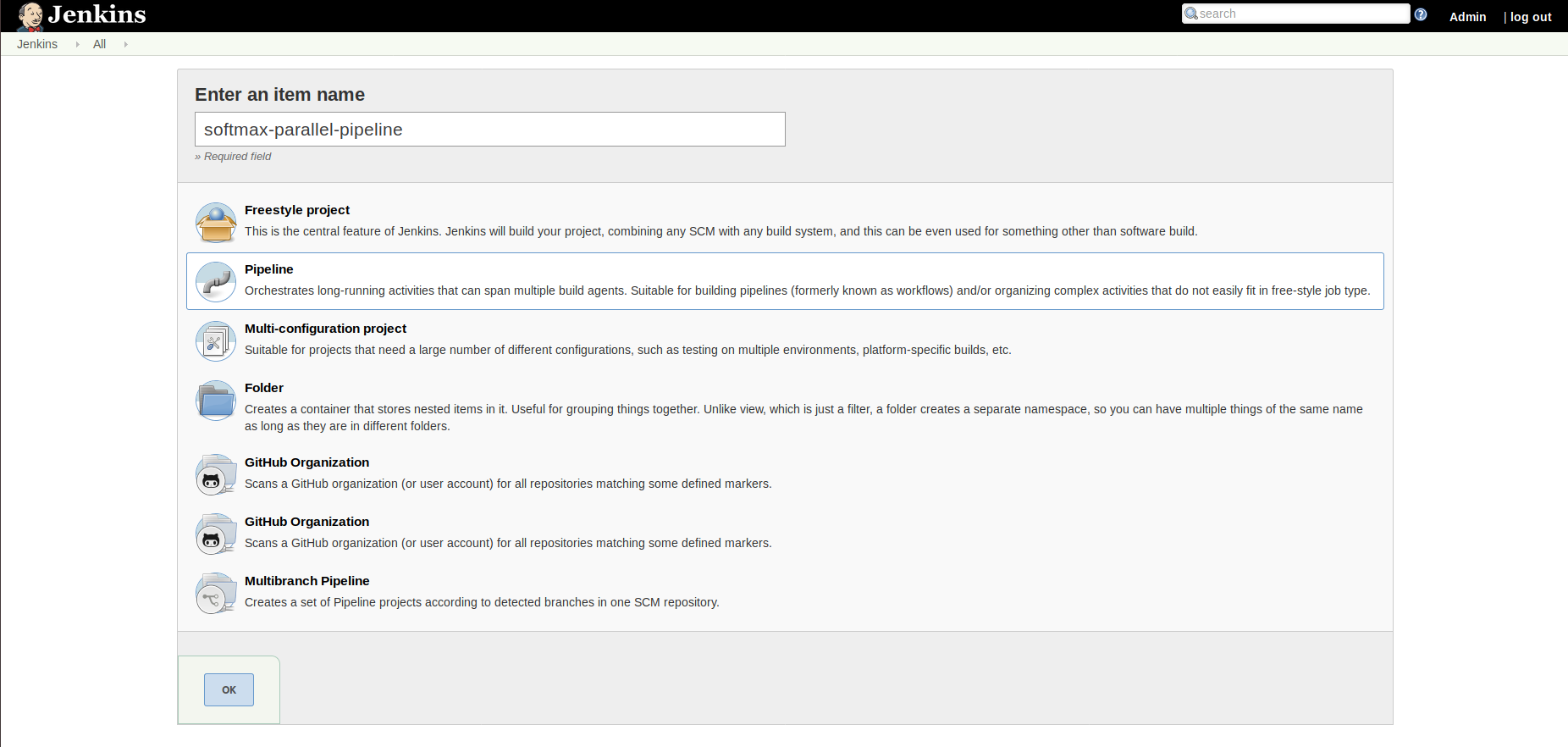
* Navigate to slave\_node\_1
* Install the Maven Plugin and add a Maven Installer in the *Global Tools Configuration* page.



* Repeat the same steps for slave\_node\_2.



* Go to Jenkins dashboard for the master node.
* Click on *New Item*.
* Enter a name for your build job.
* Select *Pipeline* as the build job type.



* Click OK.
* On the configuration page, scroll down to the Pipeline section.
* In the script field, paste the script below:

**pipeline {**

**agent none**

**tools {**

**maven 'my\_mvn'**

**}**

**stages {**

**stage("Checkout") {**

**agent { label 'slave\_node\_1' }**

**steps {**

**git url: 'https://github.com/judy-simplilearn/Softmax.git'**

**}**

**}**

**stage('Build') {**

**agent { label 'slave\_node\_1' }**

**steps {**

**sh "mvn compile"**

**}**

**}**

**stage("Unit test") {**

**agent { label 'slave\_node\_2' }**

**steps {**

**git url: 'https://github.com/judy-simplilearn/Softmax.git'**

**sh "mvn test"**

**}**

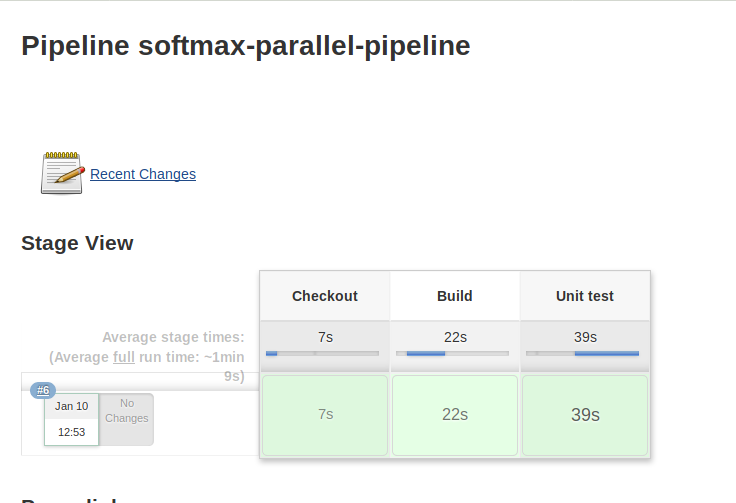
**}**

**}**

**}**

* Click Save.



* 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 nodes.



