

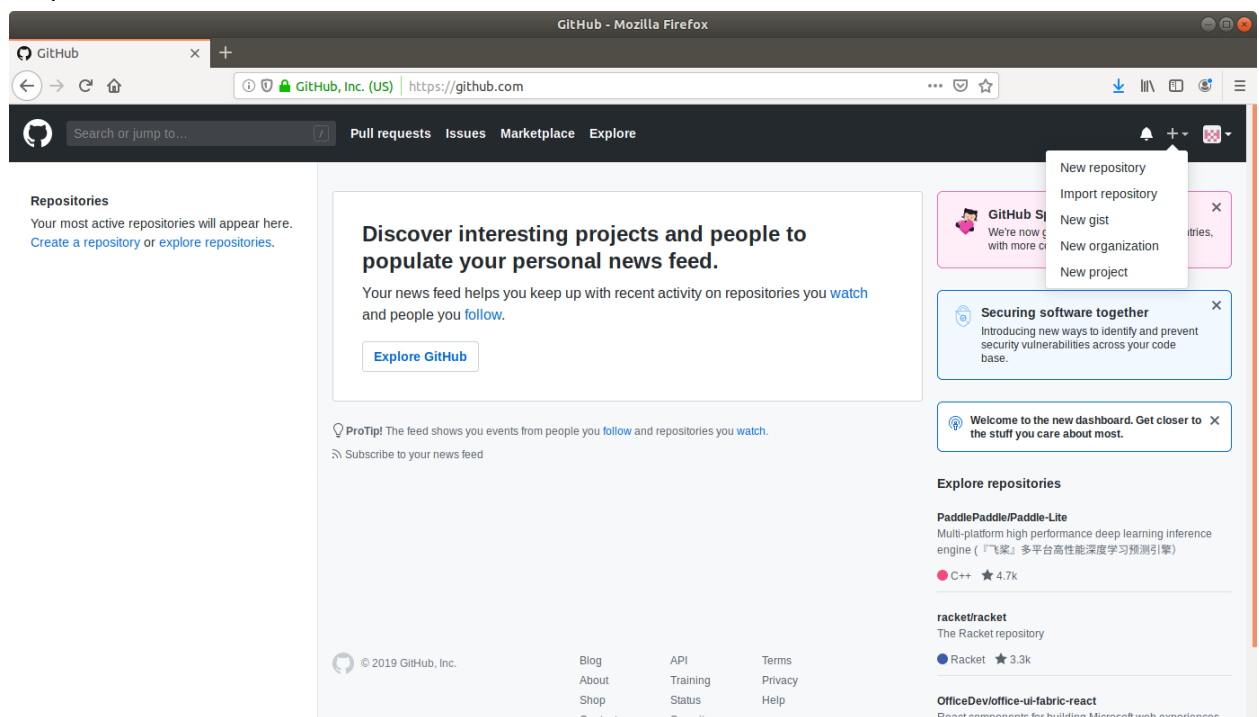
Distributed Builds

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.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: judy-simplilearn / Repository name: piperchat

Great repository names are short and memorable. Need inspiration? How about [reimagined-fiesta](#)?

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None

- 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/

Spring Initializr
Bootstrap your application

Project: Maven Project Gradle Project

Language: Java Kotlin Groovy

Spring Boot: 2.2.3 (SNAPSHOT) 2.2.2 2.1.12 (SNAPSHOT) 2.1.11

Project Metadata

Group: com.simplilearn

Artifact: piperchat

> Options

Dependencies

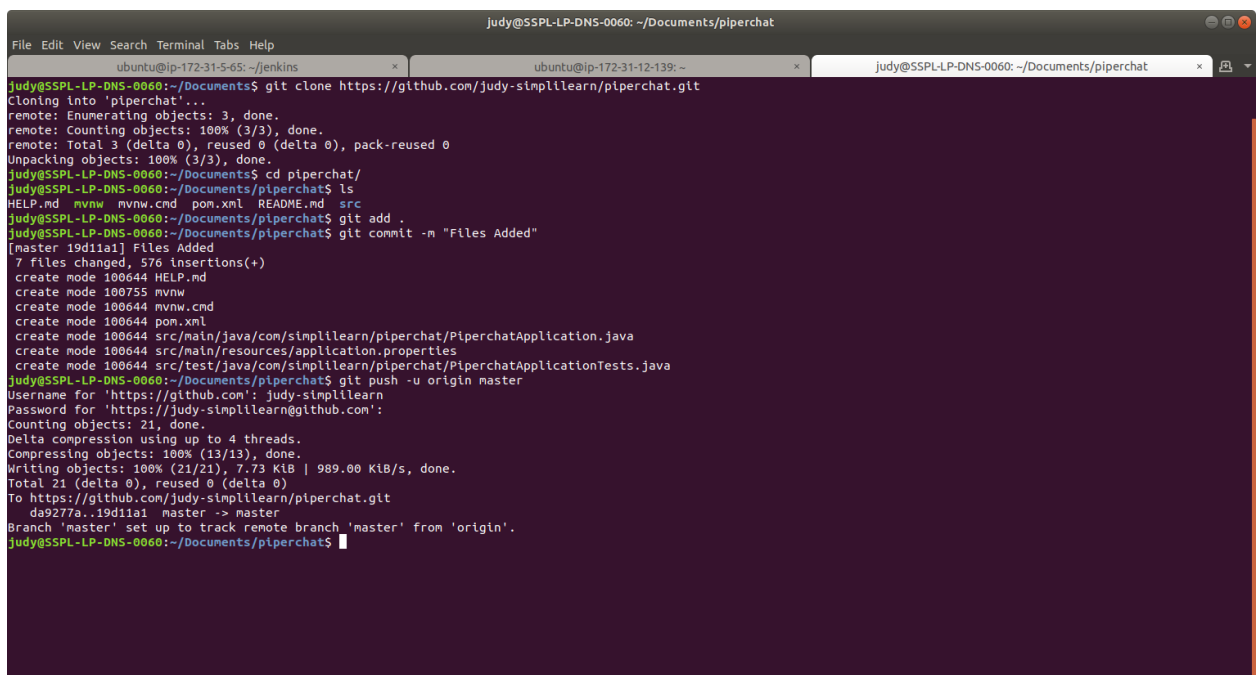
Search dependencies to add Selected dependencies

Generate - Ctrl + G Explore - Ctrl + Space Share...

- 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

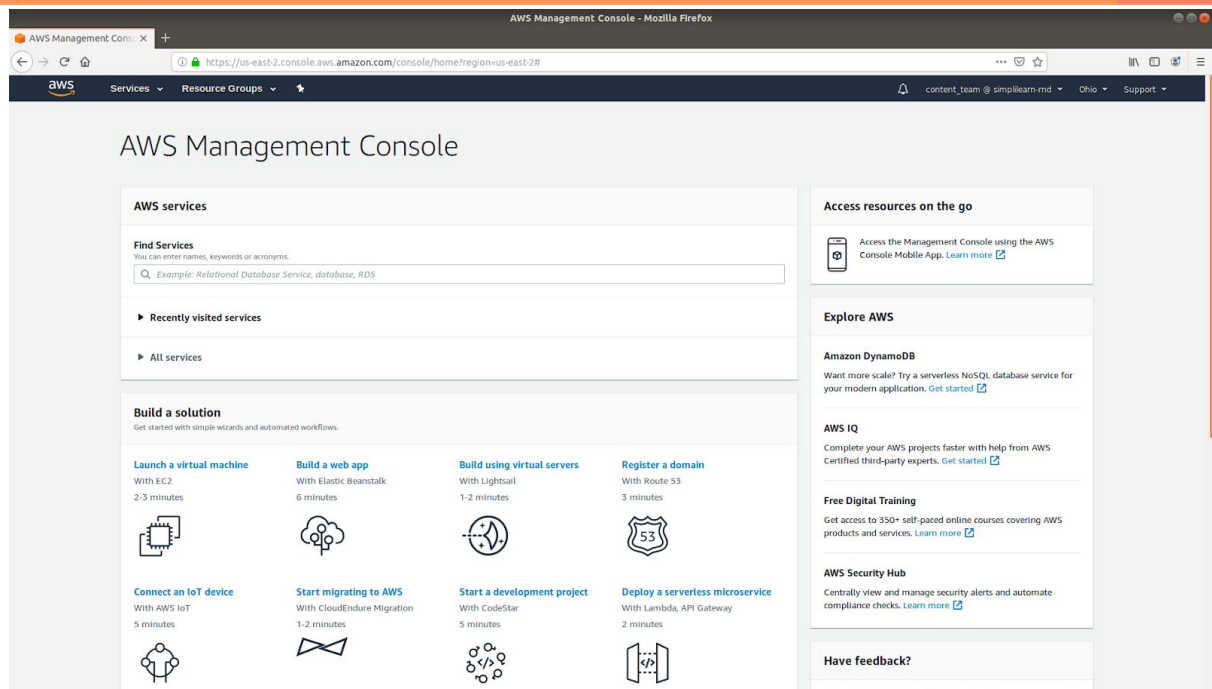
- Open the terminal and navigate to an appropriate location.
- Run **git clone [URL]** to clone the repository.
- Unzip the downloaded spring boot project to the cloned repository.
- Commit the changes to the remote SCM.
- Run **git add**.
- Run **git commit -m "Add project skeleton"**
- Run **git push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



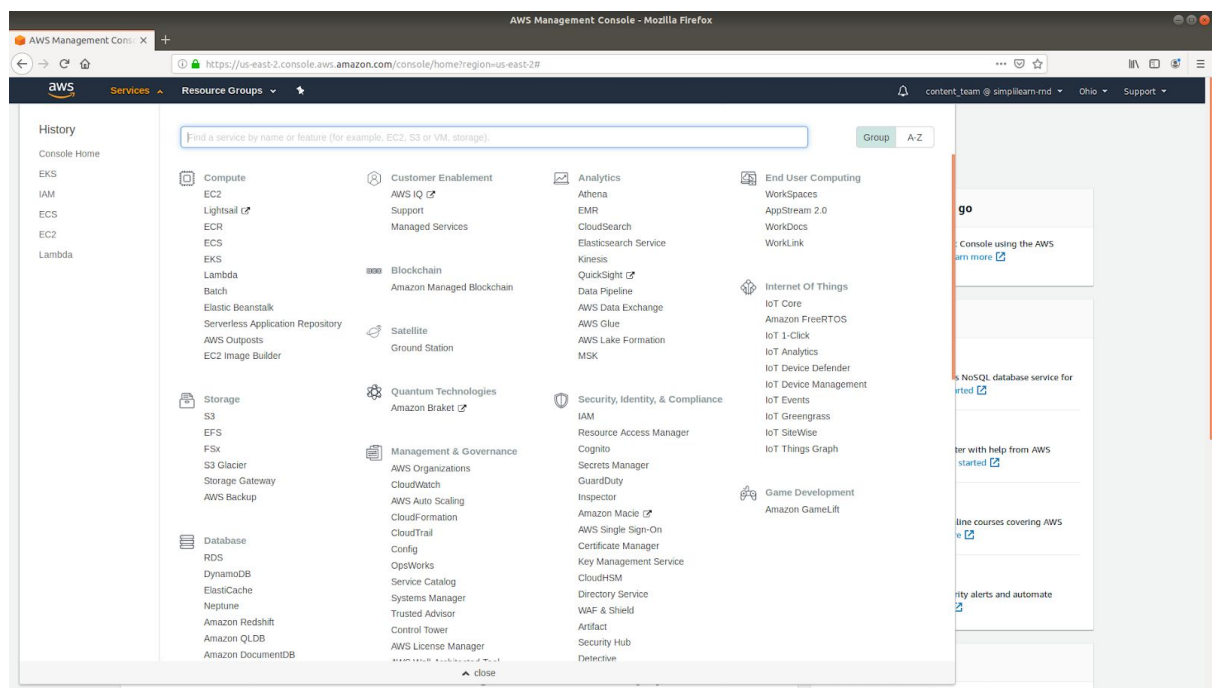
```
judy@SSPL-LP-DNS-0060: ~/Documents/piperchat
File Edit View Search Terminal Tabs Help
ubuntu@ip-172-31-5-65: ~/jenkins
ubuntu@ip-172-31-12-139: ~
judy@SSPL-LP-DNS-0060: ~/Documents/piperchat
judy@SSPL-LP-DNS-0060:~/Documents$ git clone https://github.com/judy-simplilearn/piperchat.git
Cloning into 'piperchat'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
judy@SSPL-LP-DNS-0060:~/Documents$ cd piperchat/
judy@SSPL-LP-DNS-0060:~/Documents/piperchat$ ls
HELP.md  mvnw  mvnw.cmd  pom.xml  README.md  src
judy@SSPL-LP-DNS-0060:~/Documents/piperchat$ git add .
judy@SSPL-LP-DNS-0060:~/Documents/piperchat$ git commit -m "Files Added"
[master 19d11a1] Files Added
 7 files changed, 576 insertions(+)
 create mode 100644 HELP.md
 create mode 100755 mvnw
 create mode 100644 mvnw.cmd
 create mode 100644 pom.xml
 create mode 100644 src/main/java/com/simplilearn/piperchat/PiperchatApplication.java
 create mode 100644 src/main/resources/application.properties
 create mode 100644 src/test/java/com/simplilearn/piperchat/PiperchatApplicationTests.java
judy@SSPL-LP-DNS-0060:~/Documents/piperchat$ git push -u origin master
Username for 'https://github.com': judy-simplilearn
Password for 'https://github.com': judy-simplilearn@github.com:
Counting objects: 21, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (21/21), 7.73 KiB | 989.00 KiB/s, done.
Total 21 (delta 0), reused 0 (delta 0)
To https://github.com/judy-simplilearn/piperchat.git
 da9277a..19d11a1 master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
judy@SSPL-LP-DNS-0060:~/Documents/piperchat$
```

Step 4: Creating an EC2 instance

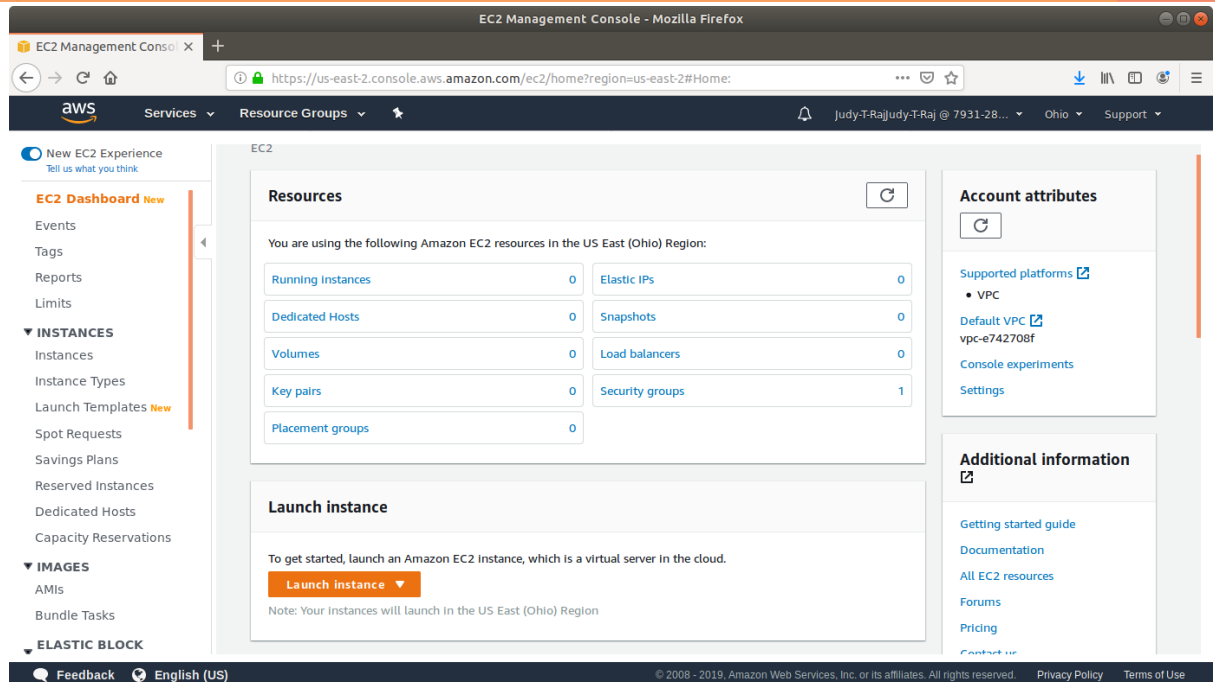
- Log in to the AWS lab account provided. You will be able to see the following screen:



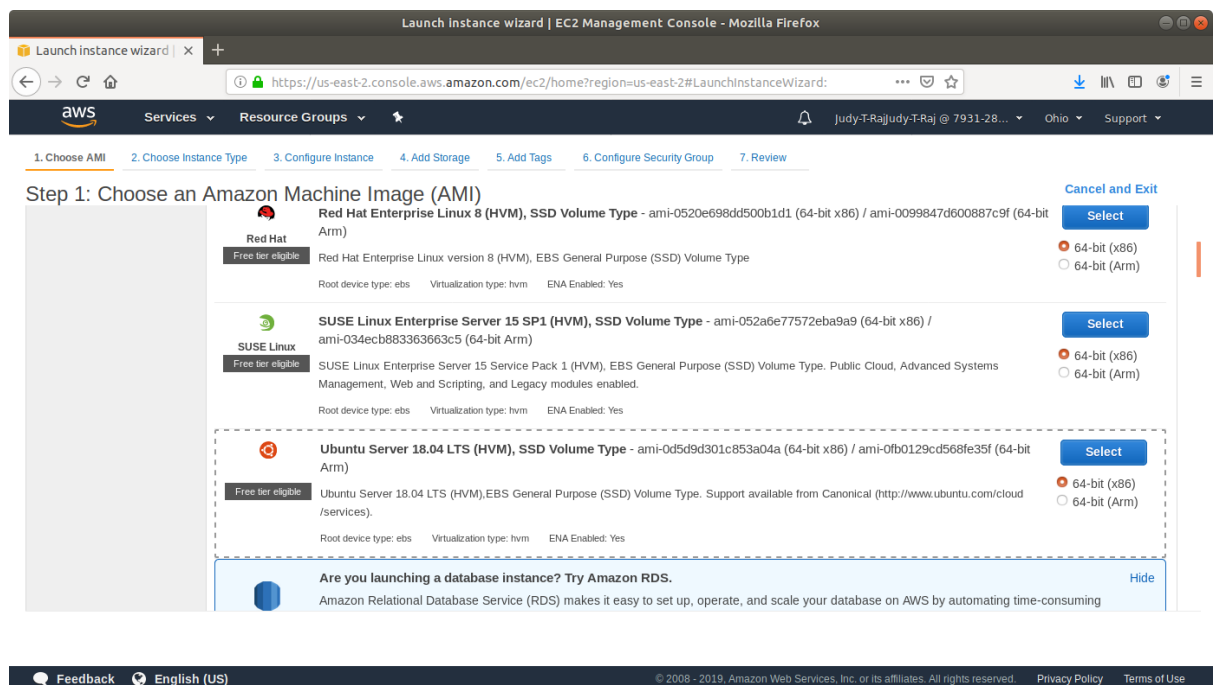
- Click on Services at the top left to view the drop-down list of resources.
- Click on EC2 under the Compute menu from the drop-down list.



- 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*.

Launch instance wizard | EC2 Management Console - Mozilla Firefox

Launch instance wizard | x +

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Judy-T-Raj@7931-28... Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click on *Launch*.

Launch instance wizard | EC2 Management Console - Mozilla Firefox

Launch instance wizard | x +

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Judy-T-Raj@7931-28... Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details [Edit AMI](#)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0d5d9d301c853a04a

Free tier eligible Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

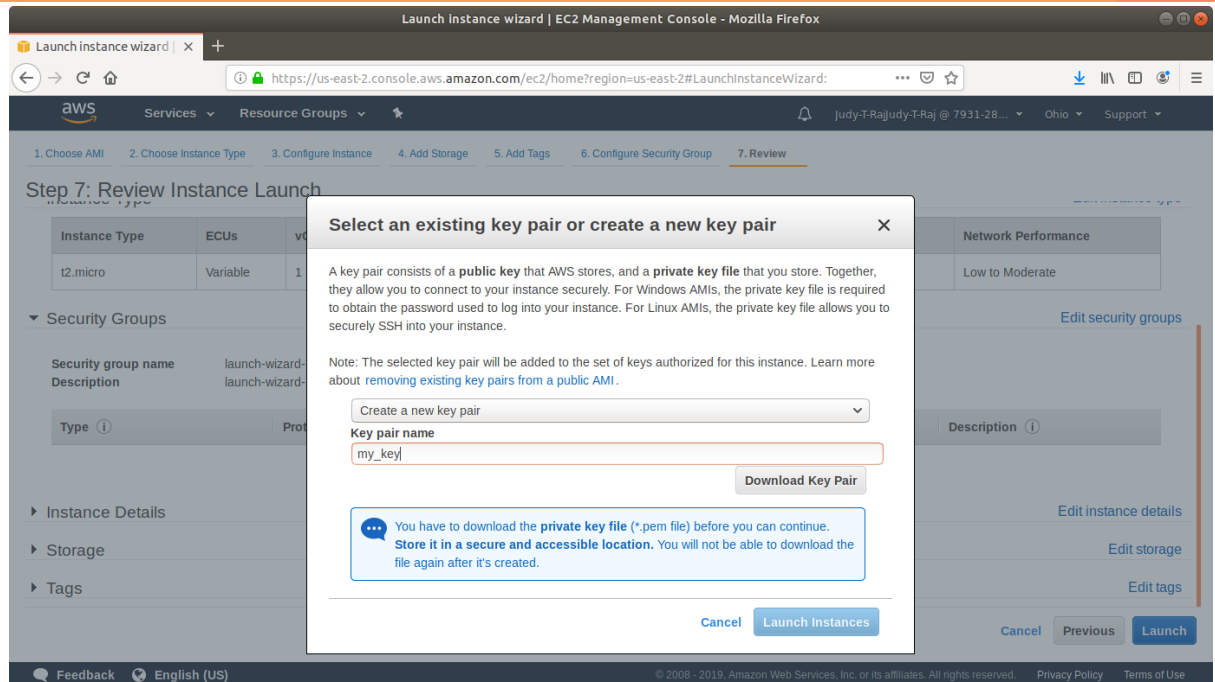
Security group name: launch-wizard-1
Description: launch-wizard-1 created 2019-12-17T15:56:06.249+05:30

Type	Protocol	Port Range	Source	Description

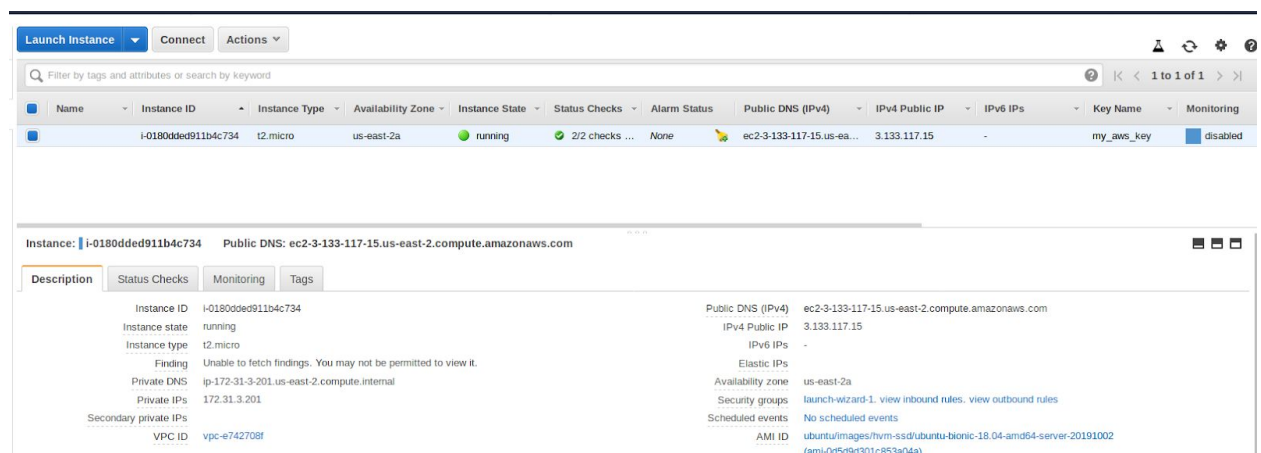
Cancel Previous **Launch**

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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

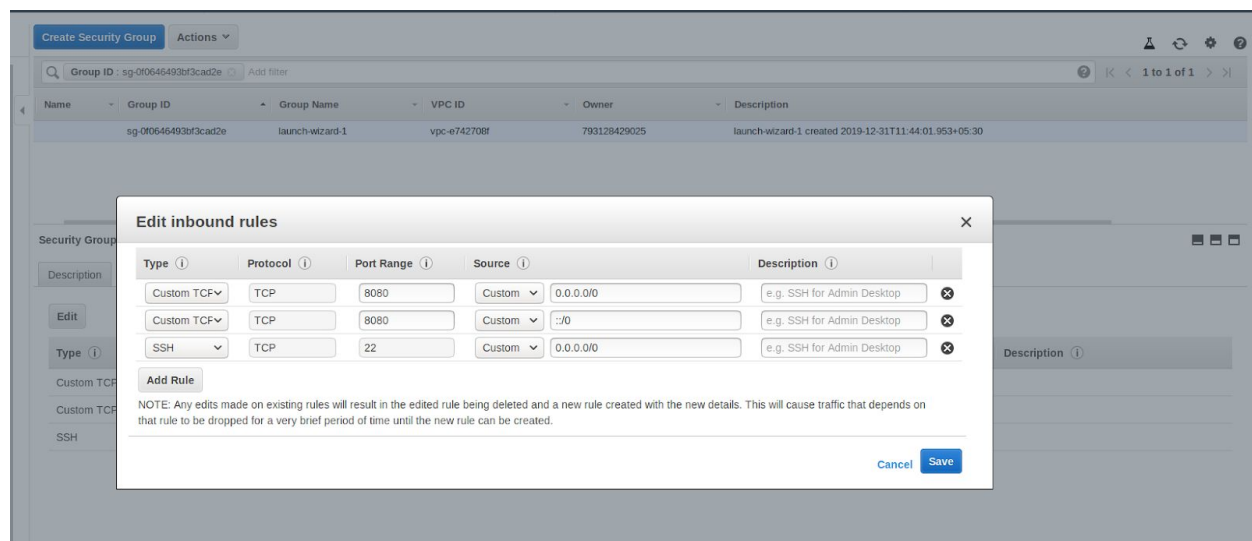
- Add a rule to the security group to which the instance belongs to allow HTTP traffic to port 8080 with the following settings:

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 ubuntu@<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++ysA1NkErThNdtLMBctMZeE.
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 -

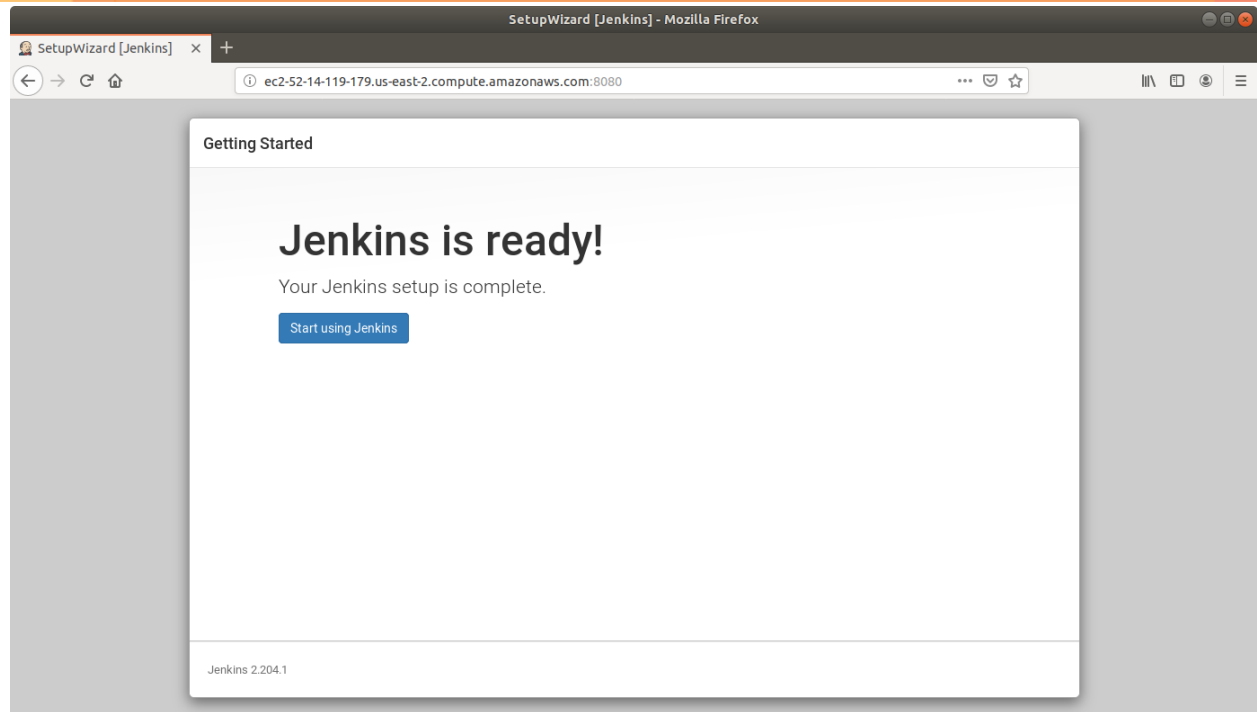
**sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'**

sudo apt update

sudo apt install jenkins

sudo ufw allow 8080

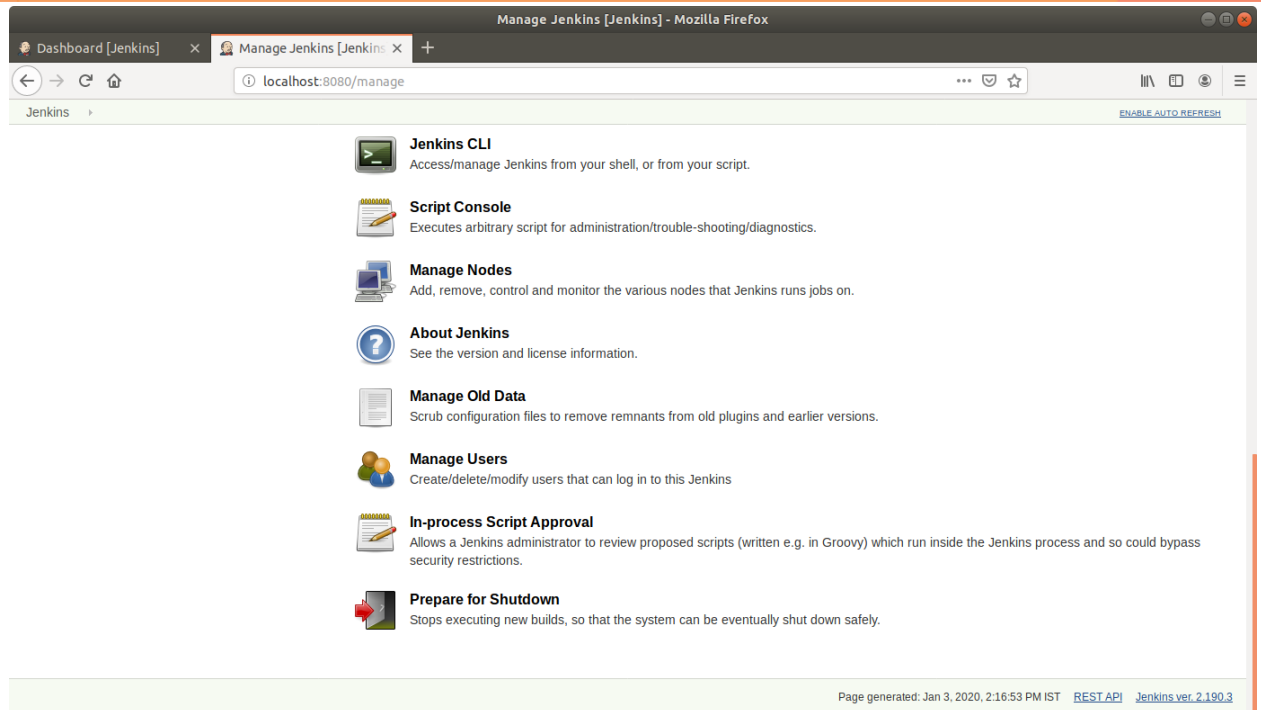
- Navigate to [http://<Public DNS \(IPv4\)>:8080](http://<Public DNS (IPv4)>:8080) to view the Jenkins server.
- Follow the instructions on screen to complete installation.

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

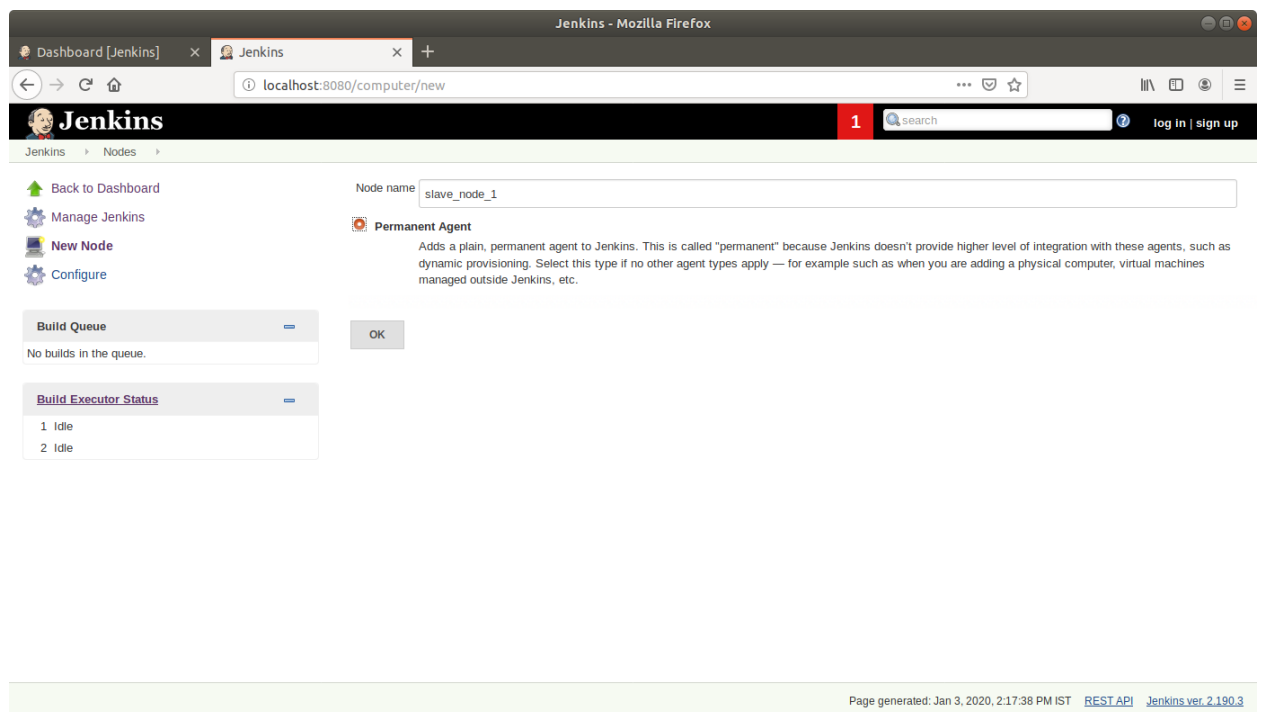
- Go to Jenkins dashboard.
- Click on *Manage Jenkins* and select *Manage Global Security*.

- 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:

search admin | log out

Name

Description

of executors

Remote root directory

Labels

Usage

Launch method

☐ Disable WorkDir

Custom WorkDir path

Internal data directory

☐ Fail if workspace is missing

Advanced...

Availability

Node Properties

☐ Disable deferred wipeout on this node

☐ Environment variables

☐ Tool Locations

Save

- 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.

slave_node_1 [Jenkins] - Mozilla Firefox

Dashboard [Jenkins] x slave_node_1 [Jenkins] x +

localhost:8080/computer/slave_node_1/

Jenkins 1 search log in | sign up

Jenkins > Nodes > slave_node_1

Back to List

Status

Delete Agent

Configure

Build History

Load Statistics

Log

Open Blue Ocean

Build Executor Status

Agent slave_node_1

Connect agent to Jenkins one of these ways:

- [Java Web Start is not available for the JVM version running Jenkins](#)
- Run from agent command line:
javaws http://localhost:8080/computer/slave_node_1/slave-agent.jnlp
- Or if the agent is headless:
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slave_node_1/slave-agent.jnlp -workDir "/opt/jenkins/"

Projects tied to slave_node_1

None

Mark this node temporarily offline

localhost:8080/computer/slave_node_1/builds

Page generated: Jan 3, 2020, 2:22:04 PM IST REST API Jenkins ver. 2.190.3

- 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 -jnlprUrl http://ec2-3-14-129-49.us-east-2.compute.amazonaws.com:8080/computer/slave_node_1/slave-agent.jnlp -s
secret 7649de7f88ffa7d9b463f57849e13af8e09529d9339d9ed797cd04509097ec4e -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.

ENABLE AUTO REFRESH

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.84 GB		5.84 GB	0ms
	slave_node_1	Linux (amd64)	In sync	5.83 GB		5.83 GB	45ms
Data obtained		0.24 sec	0.23 sec	0.24 sec	0.24 sec	0.23 sec	0.23 sec


Refresh status


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.


Enter an item name


» Required field


 **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.

 **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 builds, etc.

 **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

 **GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

 **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

- 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.

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

Description

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This build requires lockable resources

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Disable this project ?

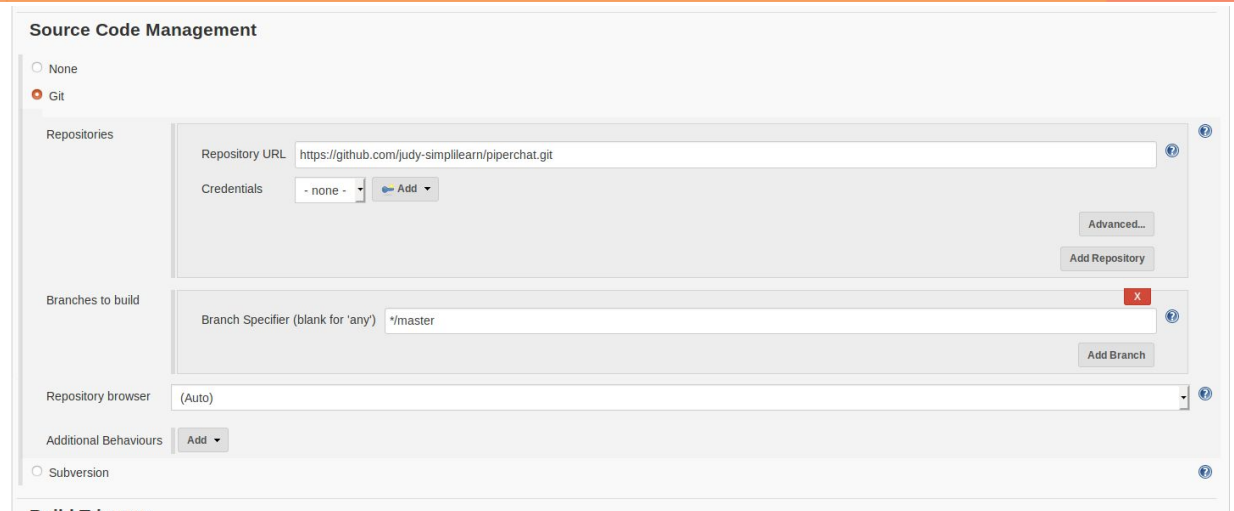
☐ Execute concurrent builds if necessary ?

☒ Restrict where this project can be run ?

Label Expression ?

[Label slave_node_1](#) is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

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



Source Code Management

☐ None

☒ Git

Repositories

Repository URL:

Credentials:

Branches to build

Branch Specifier (blank for 'any'):

Repository browser:

Additional Behaviours:

☐ Subversion

- 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



Build

☒ Execute shell

Command:

[See the list of available environment variables](#)

- 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
```