

JENKINS MASTER SLAVE CONFIGURATION

Prepared by,
T.AJAY
T.SUSMITH

Jenkins master-slave configuration

METHOD-1

Creating master-slave configuration with one master and five slave nodes.

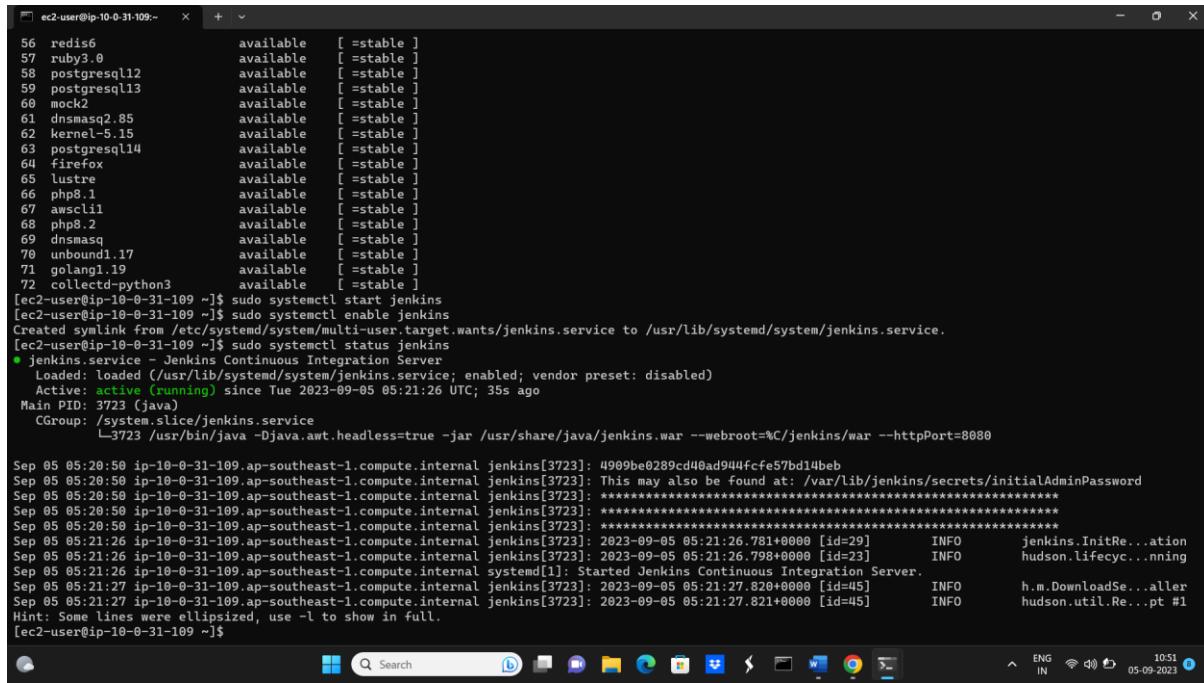
Creating master node:-

1. Create and launch ec2 -instance and connect it through shell or git bash.

2. Allow the all-traffic in the security groups.

3. Now install Jenkins by using below commands

- sudo wget -O /etc/yum.repos.d/jenkins.repo \
<https://pkg.jenkins.io/redhat-stable/jenkins.repo>
- sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
- sudo yum upgrade
- sudo yum install jenkins
- sudo systemctl daemon-reload
- sudo amazon-linux-extras install java-openjdk11
- sudo systemctl start Jenkins
- sudo systemctl enable Jenkins
- sudo systemctl status Jenkins

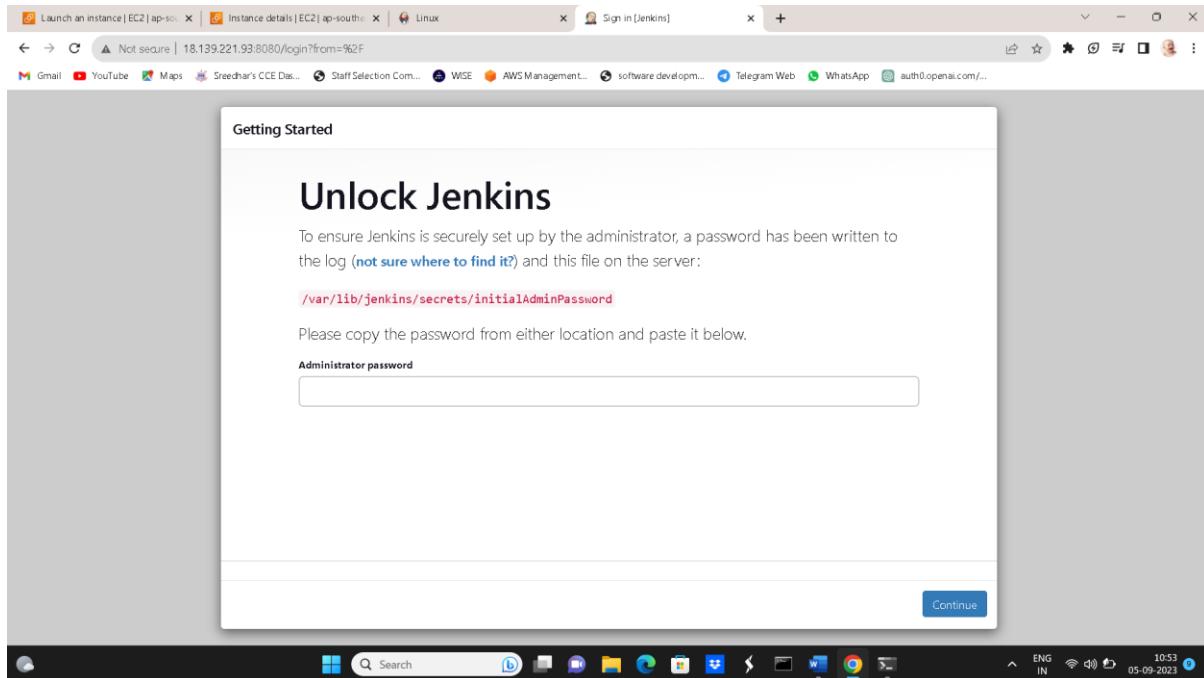


The screenshot shows a terminal window titled 'ec2-user@ip-10-0-31-109: ~'. The window displays the output of several command-line operations used to install Jenkins on an Amazon Linux instance. The commands include wget for the Jenkins repository, rpm for importing the public key, yum for upgrading packages, and systemctl for starting, enabling, and checking the status of the Jenkins service. The Jenkins service is identified as 'Jenkins Continuous Integration Server' with a main PID of 3723. The log also shows the Java command being run to start Jenkins with specific arguments. The terminal window has a standard Linux desktop interface at the bottom, including a taskbar with icons for various applications like a browser, file manager, and terminal.

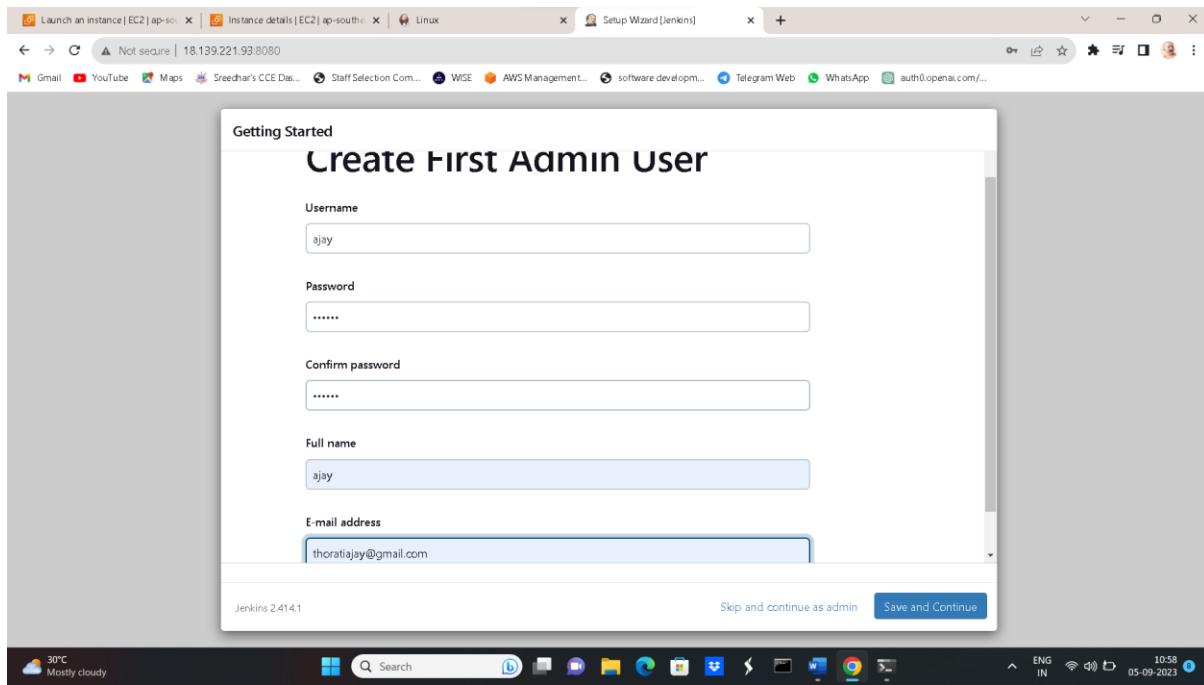
```
56 redis6 available [ =stable ]
57 ruby3.0 available [ =stable ]
58 postgresql12 available [ =stable ]
59 postgresql13 available [ =stable ]
60 mock2 available [ =stable ]
61 dnsmasq available [ =stable ]
62 kernel-5.15 available [ =stable ]
63 postgresql14 available [ =stable ]
64 firefox available [ =stable ]
65 lustre available [ =stable ]
66 php8.1 available [ =stable ]
67 awscli1 available [ =stable ]
68 php8.2 available [ =stable ]
69 dnsmasq available [ =stable ]
70 unbound1.17 available [ =stable ]
71 golang1.19 available [ =stable ]
72 collectd-python3 available [ =stable ]
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl start jenkins
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl status jenkins
● Jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2023-09-05 05:21:26 UTC; 35s ago
    Main PID: 3723 (java)
   CGroup: /system.slice/jenkins.service
           └─3723 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 4909be0289cd40ad944fcfe57bd14beb
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:26.781+0000 [id:29] INFO jenkins.InitRe...ation
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:26.798+0000 [id:23] INFO hudson.lifecyc...ning
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:27.820+0000 [id:45] INFO h.m.DownloadSe...aller
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:27.821+0000 [id:45] INFO hudson.util.Re...pt #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-0-31-109 ~]$
```

4. copy the public ip and browse it along with the port number 8080.



5. Now copy the path and go into that path you will get one key copy it and paste it .in the next step you can give name and password then your account gets created.



6. Now generate one ssh key for all the slaves by using below commands

- cd .ssh/
- ssh -keygen -t rsa
- now ssh key is generated with the name id_rsa.pub

```

[ec2-user@ip-10-0-30-3:~]$ ssh -i "mykeypair.pem" ec2-user@ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com
The authenticity of host 'ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com (13.213.30.174)' can't be established.
ED25519 key fingerprint is SHA256:SyuUpF2nwRU084FZRp72dHJM9CwYMR1hCMJLmg2QU.
This host key is known by the following other names/addresses:
  C:\Users\thora\.ssh\known_hosts:159: ec2-54-179-128-5.ap-southeast-1.compute.amazonaws.com
  C:\Users\thora\.ssh\known_hosts:173: ec2-13-229-206-153.ap-southeast-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Mon Sep  4 17:04:26 2023 from 110.225.231.236
[ec2-user@ip-10-0-30-3:~]$ ssh-keygen -t rsa
[ec2-user@ip-10-0-30-3:~]$ cd .ssh
[ec2-user@ip-10-0-30-3:~]$ cat authorized_keys
authorized_keys  id_rsa  id_rsa.pub
[ec2-user@ip-10-0-30-3:~]$ 

```

- copy this public copy for the future use of slaves connection.

Creating slave nodes:-

1. create and launch five ec2 instances and connect it through shell or gitbash.
2. After launching the instance add below commands to connect with master.
 - cd .ssh/
 - vi authorized_keys

```

[ec2-user@ip-10-0-31-109:~]$ cd .ssh
[ec2-user@ip-10-0-31-109:~]$ vi authorized_keys
[ec2-user@ip-10-0-31-109:~]$

```

- goto end (shift+A) and paste the public key (generated in master id_rsa.pub) and save it.

```

ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCS5IVt7VHG0hfKz4FLY3qAfJQu3U3Gy1qdSWh3r127yh5apZWaj08mwEgjsYV89CAu7DKpxQZrvXi7c4sFPbPUrpoy4Y530rJm8Y2nc8SsANfqYVznNQeYQ
RjXUNi3y2jYK18MBjnIVeyQTkwhX1xsUfLhAu40CpHuviA4hH0tisyGn/9n6xfgLKOMFs3W15VLOVt0x1vLyNUsmvx02l0PAjc01a6ob1RQCazSopN3IxPKOK0Xzu29/ZhNXM28zpevY4gjgpRCVyy
WuP1mEDbhx01ql9b7bpQ8t0e3vhW2Ru6tH0SpnhNuNYOfscn5dh0lzu1Tz48Ltj23jn mykeypair
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCr5dpEJqsvHceixaYBHQx4HeigtIAfi9fFVfV1ZKksDTuCIi60cnvI01KGjd4lxGAQDLLAnrl4CYeemM17P0HSvhDYNV5zuYXY+jGwfMDHs6Sfo8KJ8+
nk7Tz0KhTLAq0ZHM3W70Mn2h2NwYCs8vt74yMdWYrHu9xmgyxlK28ieCn4Rq+oFJpcGBVP1RGraQKM/hbpNEQgwGpZyYe8sFyTzs2n4/ln9UZEbrB3Ty5N9f8HLiHDEER9EA0Ypb2Qkp4H9d3ZRhpGI
1gtZxbro0pq/YpvaI9wPdIChwltRazhHByrQSOW2Efcrqh3FVh+gQWos+QVSi25rt1 ec2-user@master

```

- follow the above steps for all the slaves.
3. Now configure new nodes in the slaves
- Goto manage Jenkins → nodes → give name and select permanent node → click on create node

The screenshot shows the Jenkins 'New node' configuration page. The 'Node name' field contains 'slave5'. The 'Type' section has a radio button selected for 'Permanent Agent'. Below it, a tooltip explains: '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.' A blue 'Create' button is at the bottom.

- No of executors=10
- Remote root directory:/home/ec2-user
- Labels: slave1(provide label name)
- Usage: use this node as much as possible
- Launch method :launch agent via ssh
- Host: provide ipv4 private ip of your instance

The screenshot shows the Jenkins 'Nodes' configuration page after the 'Create' button was clicked. The 'Number of executors' is set to 10. The 'Remote root directory' is set to '/home/ec2-user'. The 'Labels' are set to 'slave5'. The 'Usage' dropdown shows 'Use this node as much as possible'. The 'Launch method' dropdown shows 'Launch agents via SSH'. The 'Host' dropdown shows '10.0.31.109'. A blue 'Save' button is at the bottom.

- In credentials you need to create your credentials click on add select Jenkins kind → ssh username and private key, username → ec2-user, key → copy your pem file and paste here. click on create and select that created credentials.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

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

ID ?

Description ?

- Host key verification strategy : manually trusted key verification strategy
- Click on save

Launch an instance | EC2 | ap-southern... | Instance details | EC2 | ap-southern... | Linux | Jenkins

Not Secure | 18.139.221.93:8080/manage/computer/createnode

Gmail YouTube Maps Sreedhar's CCE Das... Staff Selection Com... WISE AWS Management... software development Telegram Web WhatsApp auth0.openai.com/...

Dashboard > Manage Jenkins > Nodes >

Credentials ?

ec2-user

Add ▾

Host Key Verification Strategy ?

Manually trusted key Verification Strategy

Require manual verification of initial connection ?

Advanced ▾

Availability ?

Keep this agent online as much as possible

Node Properties

Disable deferred wipeout on this node ?

Environment variables

Tool Locations

Save

30°C Mostly cloudy Search ENG IN 11:37 05-09-2023

- Now your node created successfully and your agent connected and online.

```

[09/05/23 06:07:23] [SSH] Checking java version in /home/ec2-user/jdk/bin/java
Couldn't figure out the Java version of /home/ec2-user/jdk/bin/java
bash: /home/ec2-user/jdk/bin/java: No such file or directory

[09/05/23 06:07:23] [SSH] Starting sftp client.
[09/05/23 06:07:23] [SSH] Copying latest remoting.jar...
[09/05/23 06:07:23] [SSH] Copied 1,371,113 bytes.

Expanded the channel window size to 4MB
[09/05/23 06:07:23] [SSH] Starting agent process: cd "/home/ec2-user" && java -jar remoting.jar -workDir /home/ec2-user -jar-cache /home/ec2-user/remoting/jarCache
Sep 05, 2023 6:07:24 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory
Sep 05, 2023 6:07:24 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/remoting
<==[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 3131.vf2b_b_798b_ce99
Launcher: SSHLauncher
Communication Protocol: Standard in/out
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 stdbuf
Agent successfully connected and online

```

- Follow the above process for all slaves create nodes in every slave.

Installing httpd on slave1

- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

Execute concurrent builds if necessary ?

Restrict where this project can be run ?

Label Expression ?

slave1

Label **slave1** matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Add build step to install httpd and start httpd .

Build Steps

≡ Execute shell ?

Command

See the list of available environment variables

```
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

- Save the job and Click on build now .
- Copy the public ip browse it.



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.

You are free to use the image below on web sites powered by the Apache HTTP Server:



Installing nginx on slave2:

- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

Restrict where this project can be run ?

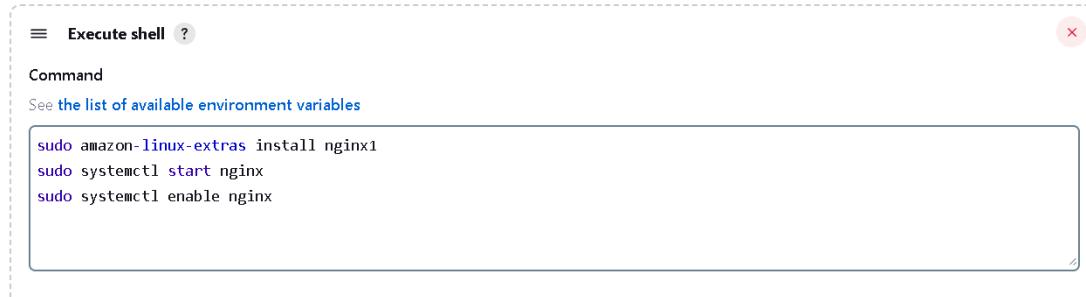
Label Expression ?

Label **slave2** matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install nginx and start nginx.

Build Steps



```

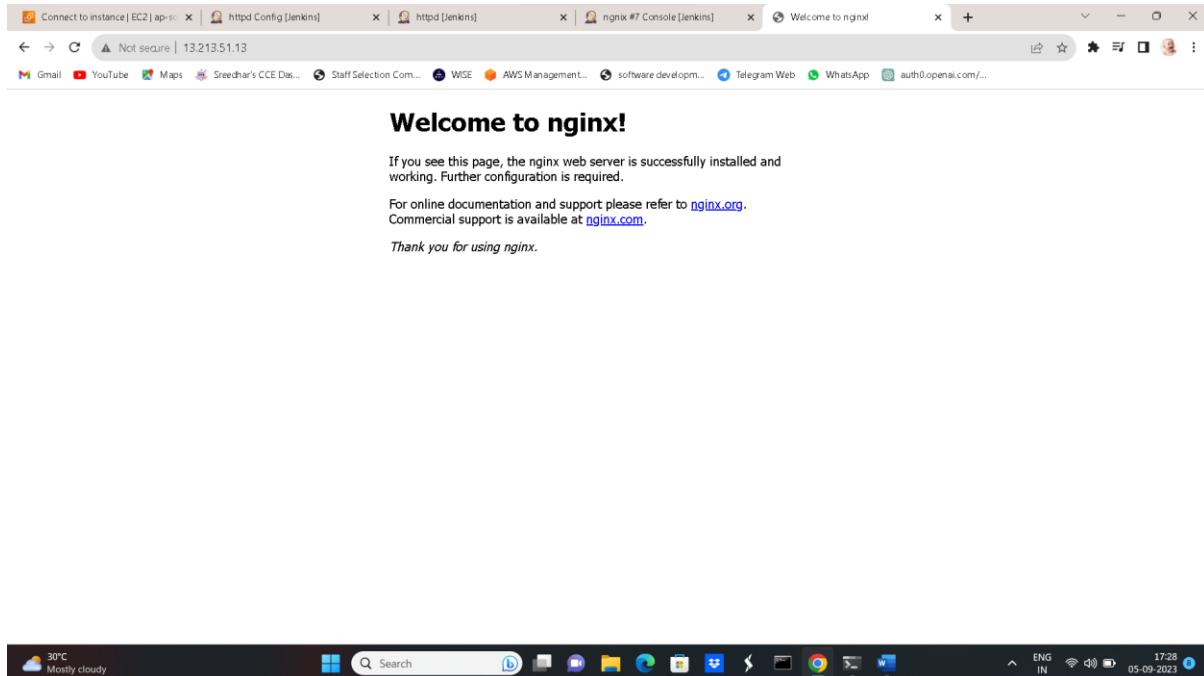
  Execute shell ? x

Command
See the list of available environment variables

sudo amazon-linux-extras install nginx1
sudo systemctl start nginx
sudo systemctl enable nginx

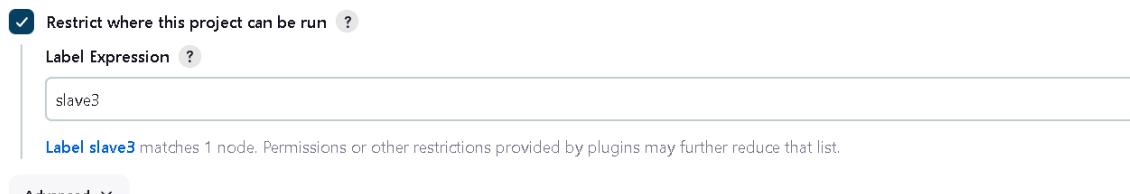
```

- Save the job and click on build now .
- Copy the public IP and browse it.



Installing sonar qube on slave3:

- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression



Restrict where this project can be run ?

Label Expression ?

slave3

Label **slave3** matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install and run sonarqube.

Build Steps

```
cd /opt
sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.6.zip
sudo unzip sonarqube-7.6.zip
sudo useradd sonar
sudo chown -R sonar:sonar /opt/sonarqube-7.6
sudo chmod -R 775 /opt/sonarqube-7.6
cd /opt/sonarqube-7.6/bin/linux-x86-64
sudo su sonar ./sonar.sh start
```

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

- Before going to build the job add sonar user in sudoers file.

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
jenkins ALL=(ALL) NOPASSWD:ALL
sonar  ALL=(ALL) NOPASSWD:ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
"/etc/sudoers.tmp" 121L, 4387B
```

- Now save the job and click on build now .

- Copy the public ip and browse it along with port number 9000.

Continuous Code Quality

0 Bugs
0 Vulnerabilities
0 Code Smells

Projects Analyzed

Multi-Language

20+ programming languages are supported by SonarQube thanks to our in-house code analyzers, including:

Java	C/C++	C#	COBOL	ABAP	HTML	RPG	JavaScript	TypeScript	Objective C	XML
VB.NET	PL/SQL	T-SQL	Flex	Python	Groovy	PHP	Swift	Visual Basic	PL/I	

Quality Model

Bugs track code that is demonstrably wrong or highly likely to yield unexpected behavior.

Vulnerabilities are raised on code that is potentially vulnerable to exploitation by hackers.

Code Smells will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.

Installing apache tomcat on slave4:

- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

Restrict where this project can be run ?

Label Expression ?

slave4

Label slave4 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install and run tomcat server.

The screenshot shows the Jenkins interface for configuring a job named 'tomcat'. The 'Build Steps' section contains a single 'Execute shell' step with the following command:

```

sudo rm -rf *
sudo su -
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz
tar xvzf apache-tomcat-9.0.80.tar.gz
cd apache-tomcat-9.0.80/bin
sh startup.sh

```

The 'Post-build Actions' section is empty, with 'Save' and 'Apply' buttons visible.

- Now save the job and click on build now.
- Copy the public ip and browse it along with port number 8080.

The screenshot shows the Apache Tomcat 9.0.80 homepage. It features a green banner at the top stating, "If you're seeing this, you've successfully installed Tomcat. Congratulations!" Below the banner is a cartoon cat icon. To the right are three buttons: "Server Status", "Manager App", and "Host Manager". The main content area includes sections for "Developer Quick Start", "Documentation", and "Getting Help".

Developer Quick Start

- Tomcat Setup
- First Web Application
- Realms & AAA
- JBDC DataSources
- Examples
- Servlet Specifications
- Tomcat Versions

Documentation

- [Tomcat 9.0 Documentation](#)
- [Tomcat 9.0 Configuration](#)
- [Tomcat Wiki](#)

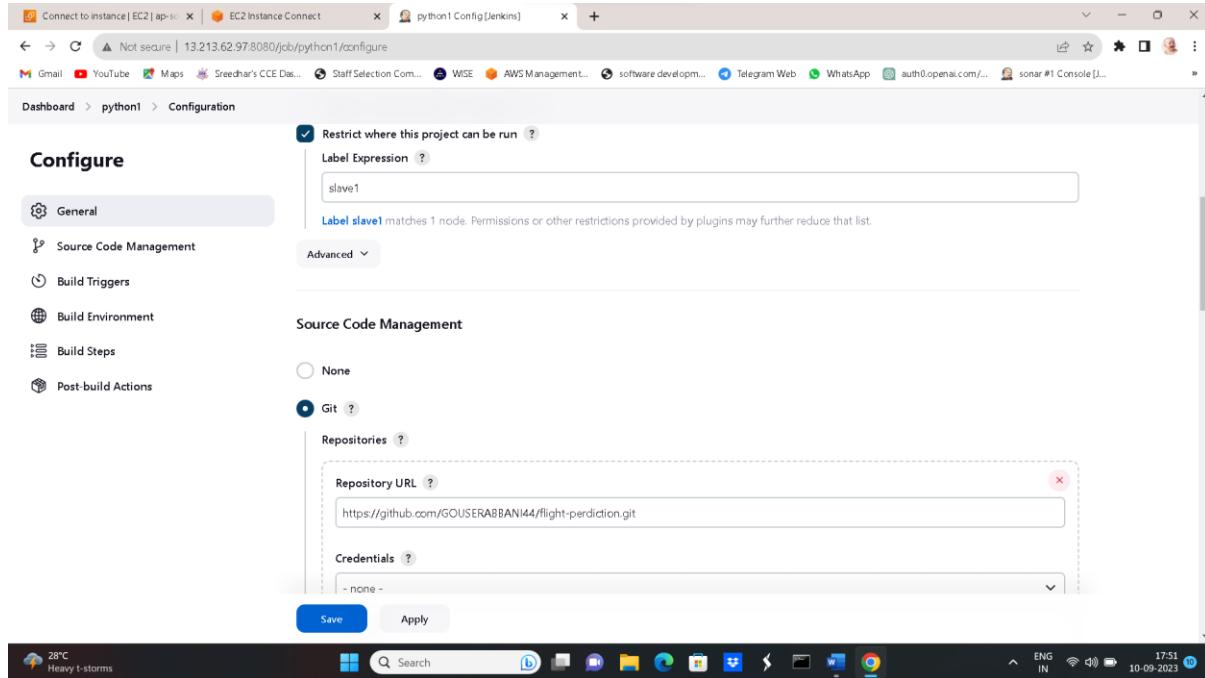
Getting Help

- [FAQ and Mailing Lists](#)
- The following mailing lists are available:
 - [tomcat-announce](#): Important announcements, releases, security vulnerability notifications. (Low volume).
 - [tomcat-users](#): User support and discussion.
 - [taglibs-user](#): User support and discussion for [Apache Taglibs](#).
 - [tomcat-dev](#): Development mailing list, including commit messages.

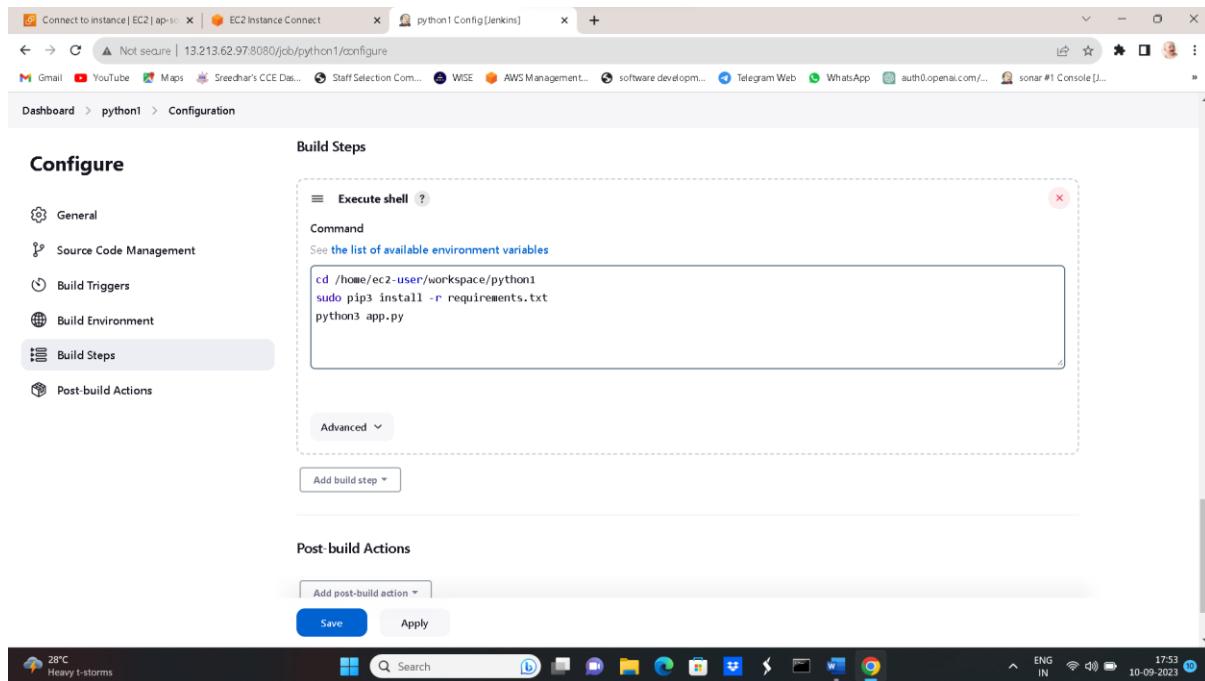
Method-2

Step-1: Clone, build and deploy any python application.

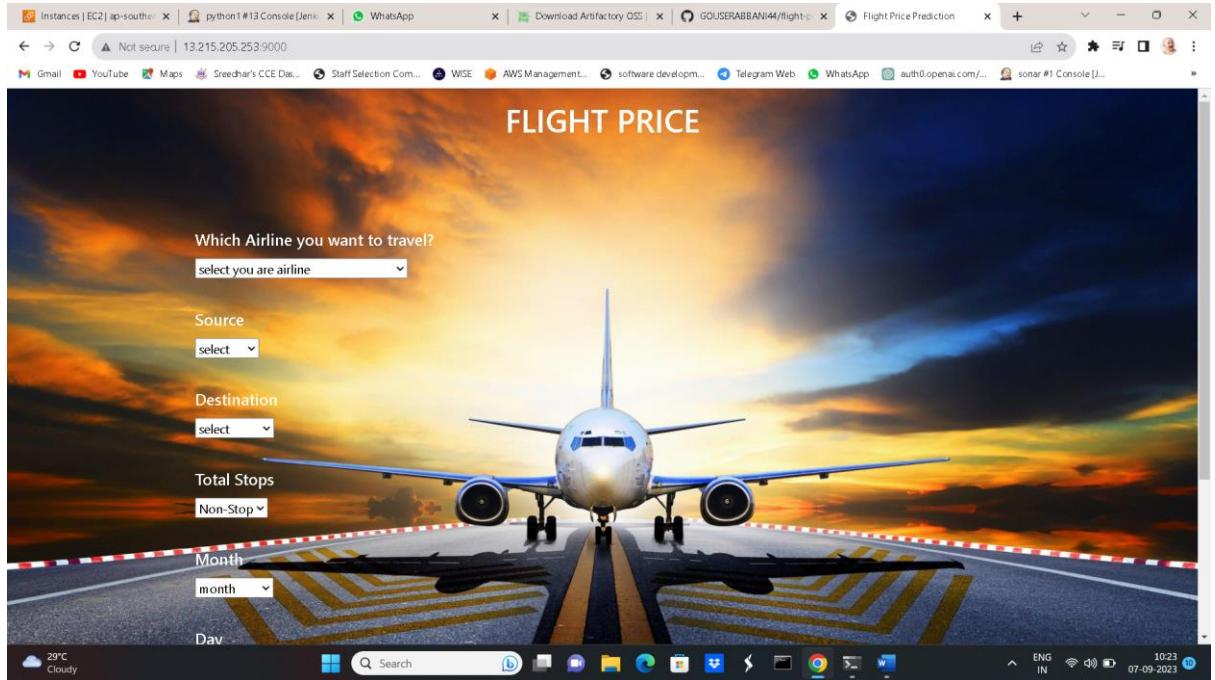
- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.
- Choose the source code management as git and provide repository url and the branch name.



- Create build step to download the requirements and run python application.



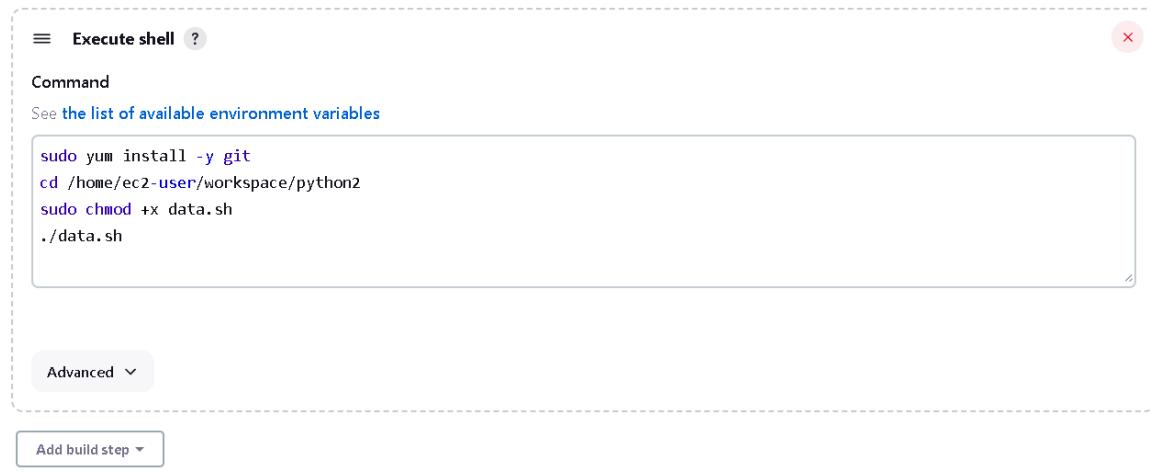
- Save the job and click on build now.
- Copy the public ip address and browse along with port number 9000.



Step-2: Clone, build and deploy any python application using bash script.

- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.
- Choose the source code management as git and provide repository url and the branch name.

- select build step and choose execute shell and add commands to execute the bash script.



- browse the public ip along with the port number 9000.

Library

Navigation

- Home
- Books Store
- Book Management
- Books
- Add Books
- Issue Books
- Return Books
- Customer Management
- Customers
- Add Customer

Dashboard

Lorem, ipsum dolor sit.

Most Popular Books		Highest Paying Customers		Total Customers	
Harry Potter	3	Grejo Joby	2400	1	
The Death Collectors (Carson Ryder #2)	1				

Customers With Highest Debt		Books with Minimum Stock Left		Total Times Issued	
Grejo Joby	100	The Mystical Poems of Rumi 1: First Selection Poems 1-200	10	7	
		Harry Potter	52		
		The Death Collectors (Carson Ryder #2)	99		

Total Book Titles		Total Books in Stock		Total Transactions Amount	
4		161		2400	

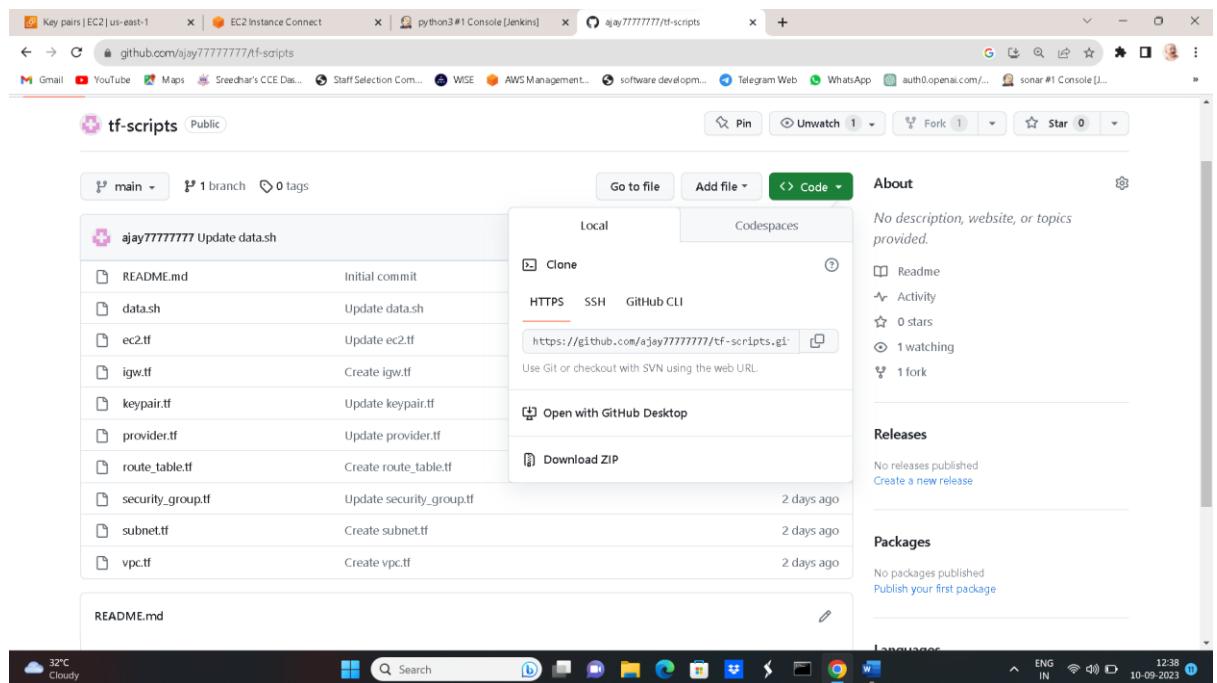
Total Debt Due	
100	

32°C
Mostly cloudy

Search ENG IN 12:17 07-09-2023

Step-3: Clone, build and deploy any python application using terraform scripts.

- Create one repository in github and add all terraform scripts to run python application with .tf extension.



- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.
- Choose the source code management as git and provide repository url and the branch name.

The screenshot shows the Jenkins job configuration for a 'Free Style Project'. Under 'Restrict where this project can be run', the 'Label Expression' is set to 'slave3'. Under 'Source Code Management', the 'Source Code Management' type is set to 'Git' and the 'Repository URL' is set to 'https://github.com/ajay77777777/tf-scripts.git'.

The screenshot shows the Jenkins job configuration for a 'Free Style Project'. Under 'Restrict where this project can be run', the 'Label Expression' is set to 'slave3'. Under 'Source Code Management', the 'Source Code Management' type is set to 'Git' and the 'Repository URL' is set to 'https://github.com/ajay77777777/tf-scripts.git'.

- select build step and choose execute shell and add commands to execute the terraform scripts.

The screenshot shows the Jenkins configuration interface for a job named 'python3'. In the 'Build Steps' section, there is a 'Execute shell' step. The command entered is:

```

sudo yum install -y yum-utils shadow-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
sudo yum -y install terraform
cd /home/ec2-user/workspace/python3
terraform init
export AWS_ACCESS_KEY="AKIAFM32SHIPP6Y4AGY4X"
export AWS_SECRET_ACCESS_KEY="Rwqq5RZMlu6xBUIBx37bqjdcZ800qy10qjuIug"
terraform validate
terraform apply --auto-approve

```

Below the command, there is an 'Advanced' dropdown and a 'Post build Actions' section with 'Save' and 'Apply' buttons.

- save the job and click on build now then your job created successfully.

The screenshot shows the Jenkins console output for a build step. The output shows the execution of Terraform commands to initialize, validate, and apply the configuration. The final message indicates success:

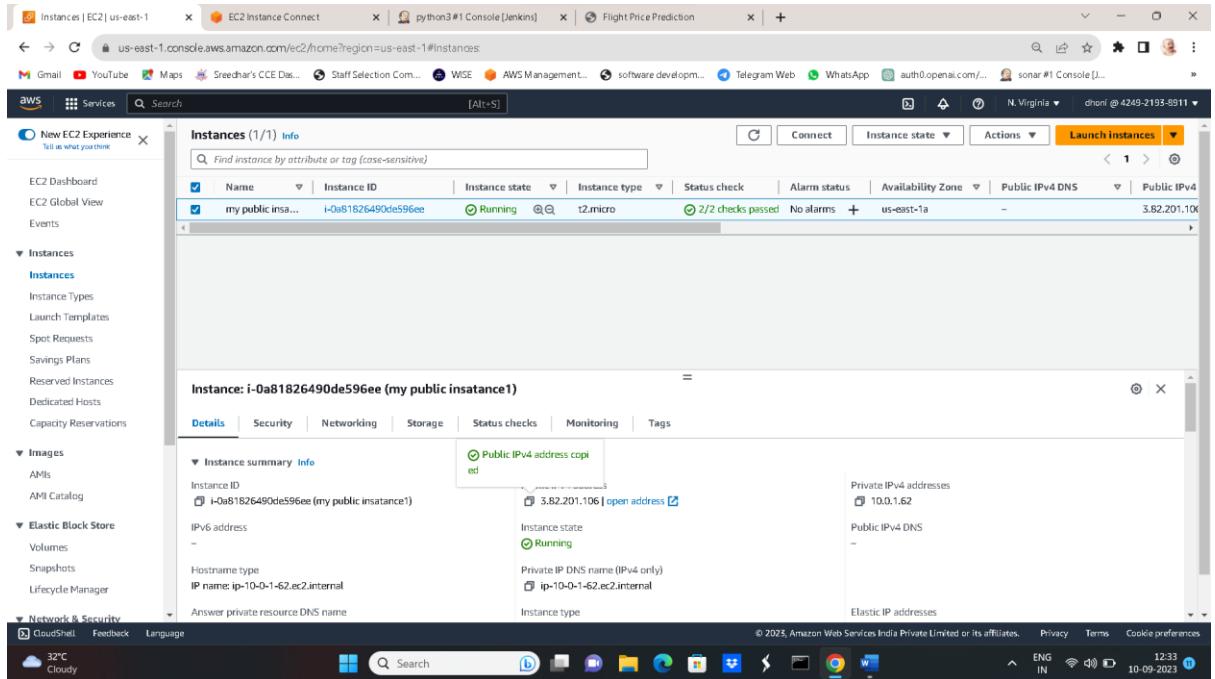
```

[0m@1maws_key_pair_vijay: Creating...[0m[0m
[0m@1maws_key_pair_vijay: Creation complete after [id=vijay][0m
[0m@1maws_vpc_demo: Creation complete after 5s [id=awpc-0ae91cda3d5a83019][0m
[0m@1maws_route_table.route: Creating...[0m[0m
[0m@1maws_internet_gateway.demogateway: Creating...[0m[0m
[0m@1maws_subnet.public-subnet-1: Creating...[0m[0m
[0m@1maws_security_group.demosg: Creating...[0m[0m
[0m@1maws_route_table.route: Creation complete after 2s [id=rtb-06c9ede866d394b5e][0m
[0m@1maws_internet_gateway.demogateway: Creation complete after 2s [id=igw-0d79b4a5c54621a88][0m
[0m@1maws_route.default_route: Creating...[0m[0m
[0m@1maws_route.default_route: Creation complete after 1s [id=~rtb-06c9ede866d394b5e1080289494][0m
[0m@1maws_security_group.demosg: Creation complete after 4s [id=sg-0d6527bd7b172ace3][0m
[0m@1maws_subnet.public-subnet-1: Still creating... [18s elapsed][0m[0m
[0m@1maws_subnet.public-subnet-1: Creation complete after 12s [id=subnet-0658df320d950ea0e][0m
[0m@1maws_route_table_association.rtl: Creating...[0m[0m
[0m@1maws_instance.instance1: Creating...[0m[0m
[0m@1maws_route_table_association.rtl: Creation complete after 1s [id=rthbassoc-0503a03076c274d26][0m
[0m@1maws_instance.instance1: Still creating... [10s elapsed][0m[0m
[0m@1maws_instance.instance1: Still creating... [20s elapsed][0m[0m
[0m@1maws_instance.instance1: Still creating... [30s elapsed][0m[0m
[0m@1maws_instance.instance1: Creation complete after 35s [id=1-0a81826490de596ee][0m
[0m@1maws_instance.instance1: Still creating... [32m
Apply complete! Resources: 9 added, 0 changed, 0 destroyed.
[0mFinished: SUCCESS

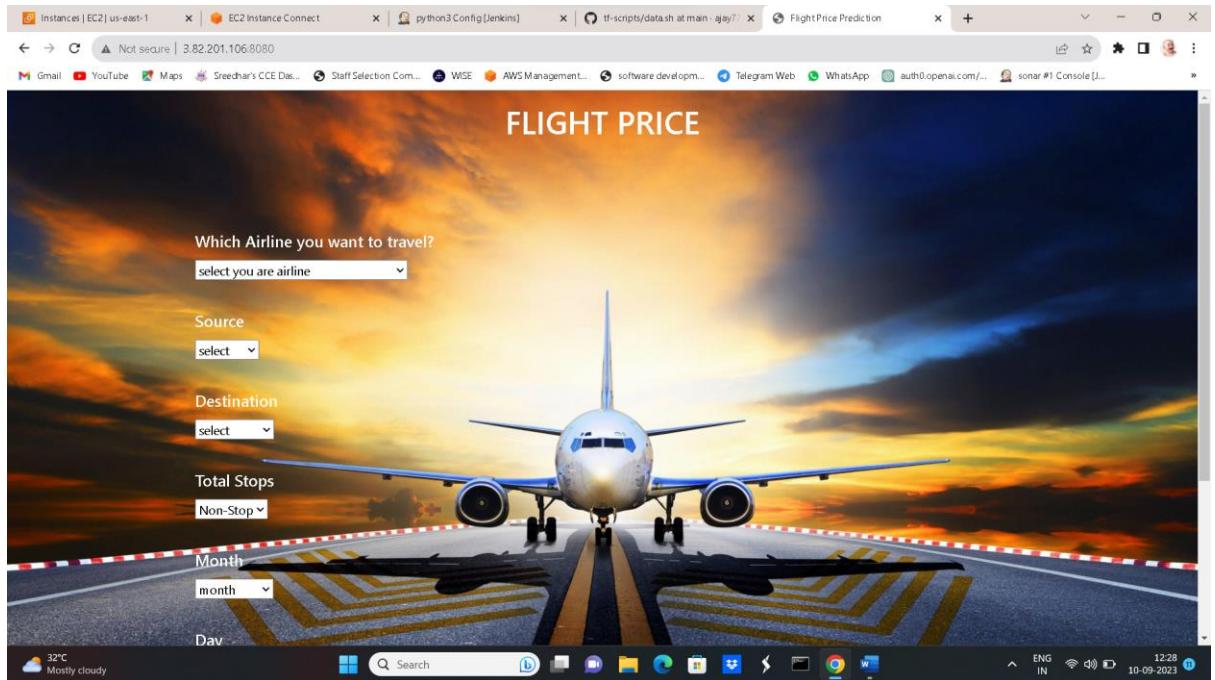
```

At the bottom right, it says 'REST API Jenkins 2.414.1'.

- instance is created in your specified region. copy the public ip of that instance.



- browse the public ip along with the port number .



Step-4: Clone, build and deploy any java application .

- in this step we are going to deploy java application in the tomcat server. Previously we are installed tomcat server in that server we are going to deploy our java application
- click on manager app

Apache Tomcat/8.5.93



- you will get an error it specifies that you are not authorized to view this page.so in order to view the page we need to modify some permissions.

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `[conf/tomcat-users.xml]` in your installation. That file must contain the credentials to let you use this webapp.
For example, to add the `Manager-gui` role to a user named `tomcat` with a password of `secret`, add the following to the config file listed above:

```
<role rolename="manager-gui"/>
<user username="tomcat" password="secret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App HowTo](#).

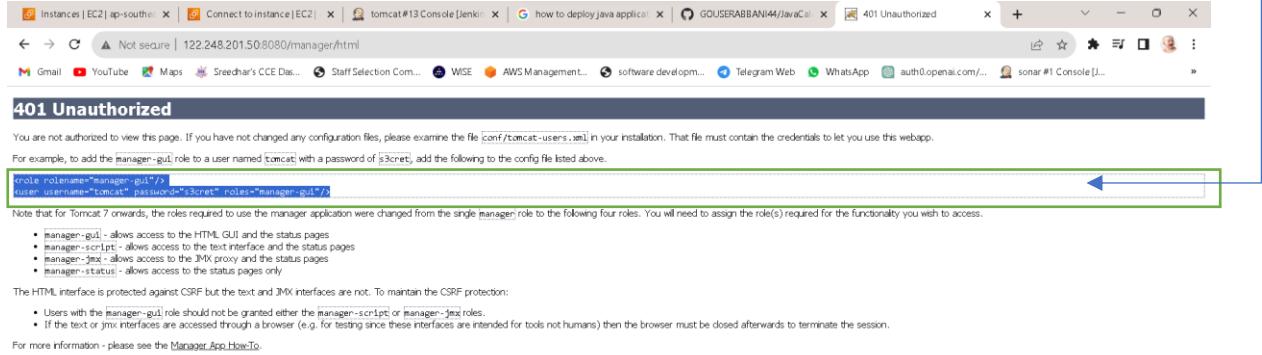


- first you connect to the instance via shell and go in to that apache tomcat server .
- modify the following files . by commenting the valve class name .

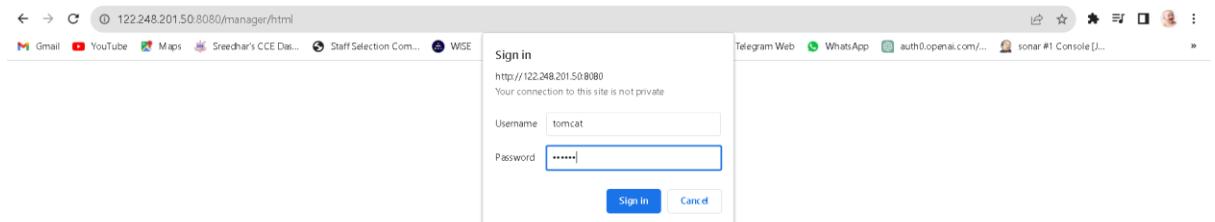
```
[ec2-user@ip-10-0-19-9 ~]$ cd workspace/
[ec2-user@ip-10-0-19-9 workspace]$ cd tomcat/
[ec2-user@ip-10-0-19-9 tomcat]$ cd apache-tomcat-8.5.93/
[ec2-user@ip-10-0-19-9 apache-tomcat-8.5.93]$ sudo vi webapps/host-manager/META-INF/context.xml
[ec2-user@ip-10-0-19-9 apache-tomcat-8.5.93]$ sudo vi webapps/manager/META-INF/context.xml
[ec2-user@ip-10-0-19-9 apache-tomcat-8.5.93]$ sudo vi conf/tomcat-users.xml
[ec2-user@ip-10-0-19-9 apache-tomcat-8.5.93]$
```

```
<Context antiResourceLocking="false" privileged="true" >
    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                      sameSiteCookies="strict" />
    <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" /> -->
    <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|HashMap)">
        <Cache type="Map" />
    </Manager>
</Context>
```

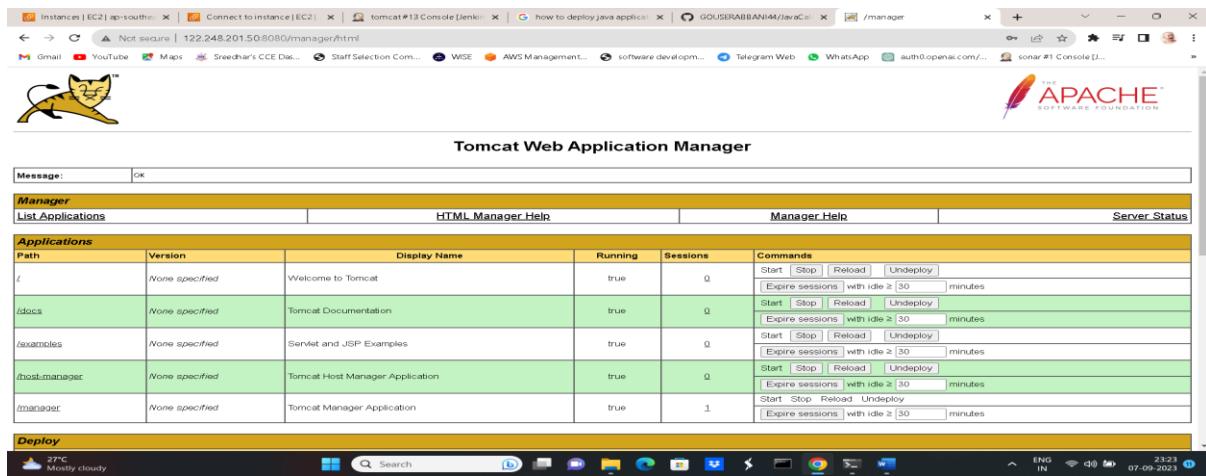
- in the tomcat-users.xml add the user name and password by adding the below lines.



- Now browse the public ip along with port number 8080.
 - Enter your username and password.



- Now you successfully enter into the manager App console.



- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.
- Choose the source code management as git and provide repository url and the branch name.

Restrict where this project can be run ?

Label Expression ?

Label slave4 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

Git ?

Repositories ?

Repository URL ?

Credentials ?

- We need to convert our pom.xml into war or jar file using build tool.
- Now choose the build step as invoke top level maven targets select maven version and choose the goal as clean and package .
- Select one more build step as execute shell and add commands to place the .WAR folder into the webapps directory.

Configure

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

Invoke top-level Maven targets

Maven Version: maven
Goals: clean package

Execute shell

Command:

```
cd /home/ec2-user/workspace/java1/webapp/target
sudo cp -r webapp-2.war /home/ec2-user/workspace/tomcat/apache-tomcat-8.5.93/webapps
```

Save Apply

- Refresh the page now you can see webaapp-2 folder is added into the tomcat server .

Manager

List Applications HTML Manager Help Manager Help Server Status

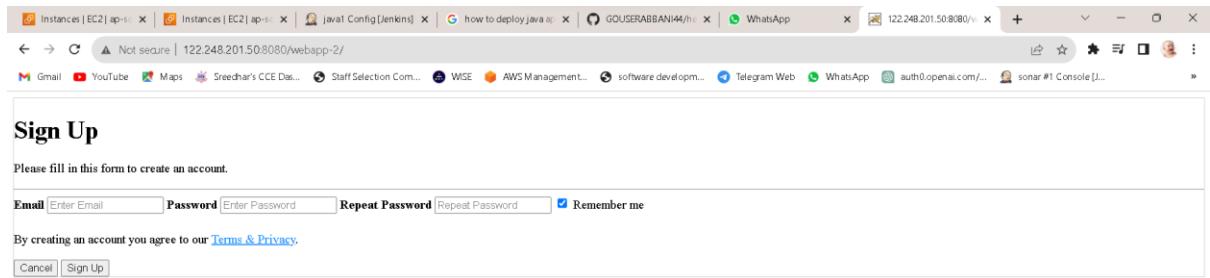
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy [Expire sessions with idle ≥ 30 minutes]
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy [Expire sessions with idle ≥ 30 minutes]
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy [Expire sessions with idle ≥ 30 minutes]
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy [Expire sessions with idle ≥ 30 minutes]
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy [Expire sessions with idle ≥ 30 minutes]
/webapp-2	None specified	Webapp	true	0	Start Stop Reload Undeploy [Expire sessions with idle ≥ 30 minutes]

Deploy

Deploy directory or WAR file located on server

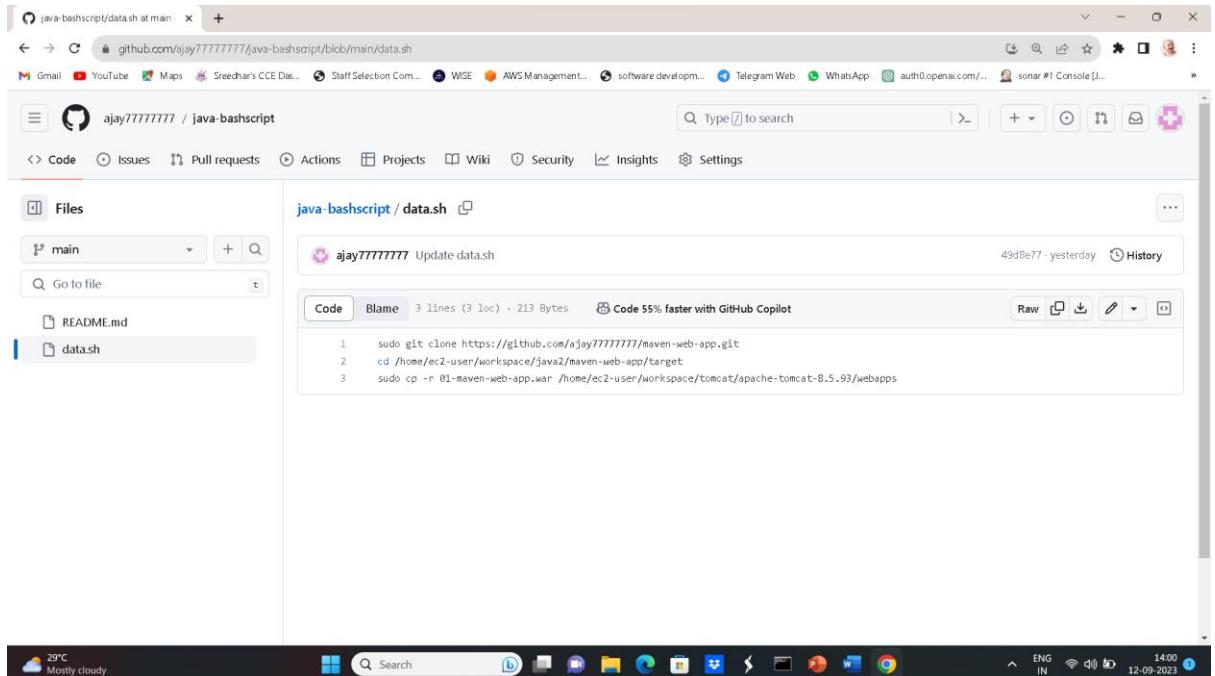
Context Path (required):
 XML Configuration file path:
 WAR or Directory path:

- Click on webapp-2 then you can see your java application.



Step-4: Clone, build and deploy any java application through bash script.

- Create one repository in github and write the bash script to run the java application.



- Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

- Choose the source code management as git and provide repository url and the branch name.

Restrict where this project can be run ?

Label Expression ?

slave3

Label slave3 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

Source Code Management

None

Git ?

Repositories ?

Repository URL ?

https://github.com/ajay7777777/tf-scripts.git

- select build step and choose execute shell and add commands to execute the bashscripts.

With Ant ?

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Steps

Execute shell ?

Command

See the list of available environment variables

```
sudo chmod +x data.sh
./data.sh
```

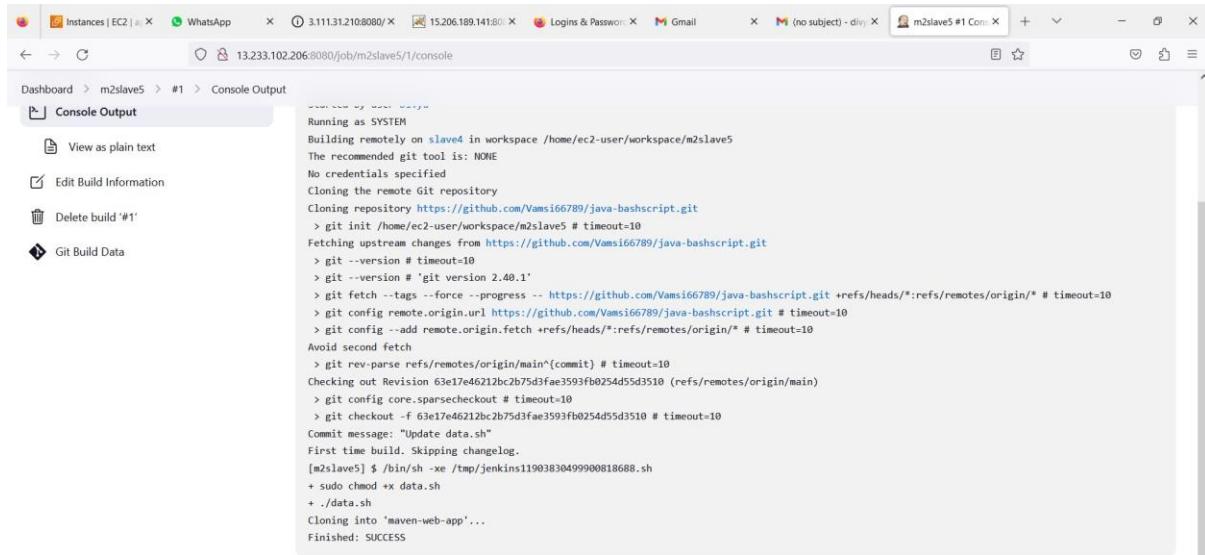
Advanced ▾

Add build step ▾

Post-build Actions

Save Apply

- save the job and click on build now then your job created successfully.



```

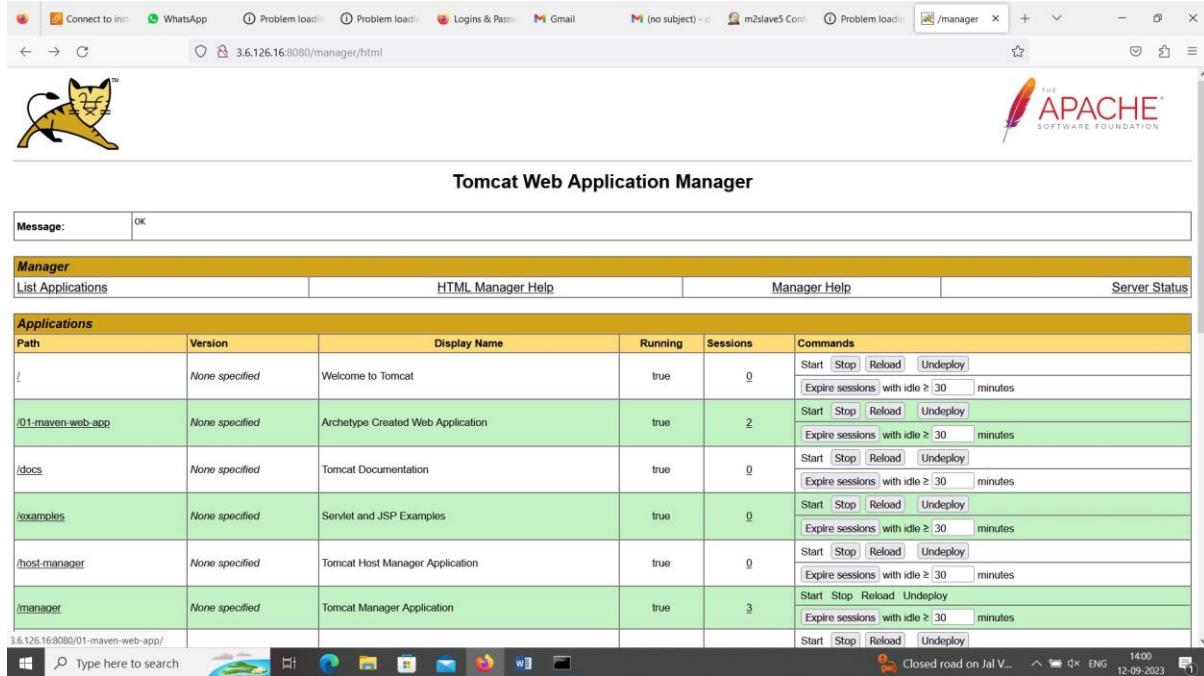
Started by user vamsi
Running as SYSTEM
Building remotely on slave4 in workspace /home/ec2-user/workspace/m2slave5
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Vamsi66789/java-bashscript.git
> git init /home/ec2-user/workspace/m2slave5 # timeout=10
Fetching upstream changes from https://github.com/Vamsi66789/java-bashscript.git
> git -version # timeout=10
> git -version # git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Vamsi66789/java-bashscript.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/Vamsi66789/java-bashscript.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 63e17e46212bc2b75d3fae3593fb0254d55d3510 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 63e17e46212bc2b75d3fae3593fb0254d55d3510 # timeout=10
Commit message: "Update data.sh"
First time build. Skipping changelog.
[m2slave5] $ /bin/sh -xe /tmp/jenkins11903830499900818688.sh
+ sudo chmod +x data.sh
+ ./data.sh
Cloning into 'maven-web-app'...
Finished: SUCCESS

```

Jenkins 2.414.1

Type here to search Closed road on HMT... 12:50 12-09-2023

- browse the public ip along with the port number 8080 .



Tomcat Web Application Manager																																															
Message:		OK																																													
Manager List Applications HTML Manager Help Manager Help Server Status																																															
Applications <table border="1"> <thead> <tr> <th>Path</th> <th>Version</th> <th>Display Name</th> <th>Running</th> <th>Sessions</th> <th>Commands</th> </tr> </thead> <tbody> <tr> <td>/</td> <td>None specified</td> <td>Welcome to Tomcat</td> <td>true</td> <td>0</td> <td> Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes </td> </tr> <tr> <td>/01-maven-web-app</td> <td>None specified</td> <td>Archetype Created Web Application</td> <td>true</td> <td>2</td> <td> Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes </td> </tr> <tr> <td>/docs</td> <td>None specified</td> <td>Tomcat Documentation</td> <td>true</td> <td>0</td> <td> Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes </td> </tr> <tr> <td>/examples</td> <td>None specified</td> <td>Servlet and JSP Examples</td> <td>true</td> <td>0</td> <td> Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes </td> </tr> <tr> <td>/host-manager</td> <td>None specified</td> <td>Tomcat Host Manager Application</td> <td>true</td> <td>0</td> <td> Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes </td> </tr> <tr> <td>/manager</td> <td>None specified</td> <td>Tomcat Manager Application</td> <td>true</td> <td>3</td> <td> Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes </td> </tr> </tbody> </table>						Path	Version	Display Name	Running	Sessions	Commands	/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes	/01-maven-web-app	None specified	Archetype Created Web Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes	/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes	/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes	/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes	/manager	None specified	Tomcat Manager Application	true	3	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
Path	Version	Display Name	Running	Sessions	Commands																																										
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes																																										
/01-maven-web-app	None specified	Archetype Created Web Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes																																										
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes																																										
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes																																										
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes																																										
/manager	None specified	Tomcat Manager Application	true	3	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes																																										

3.6.126.16:8080/01-maven-web-app/ Type here to search Closed road on Jal V... 14:00 12-09-2023

- click on 01-maven-webapp then your page will appear.



Welcome to Ashok IT..!!

Learn Here.. Lead Anywhere..!!

[Click Here To See Training Schedules](#)



METHOD-3

Creating master-slave configuration AJAY creates one master and Susmith creates five slave nodes and Ajay will build the jobs in all slave nodes in same AWS account.

Creating master node:-

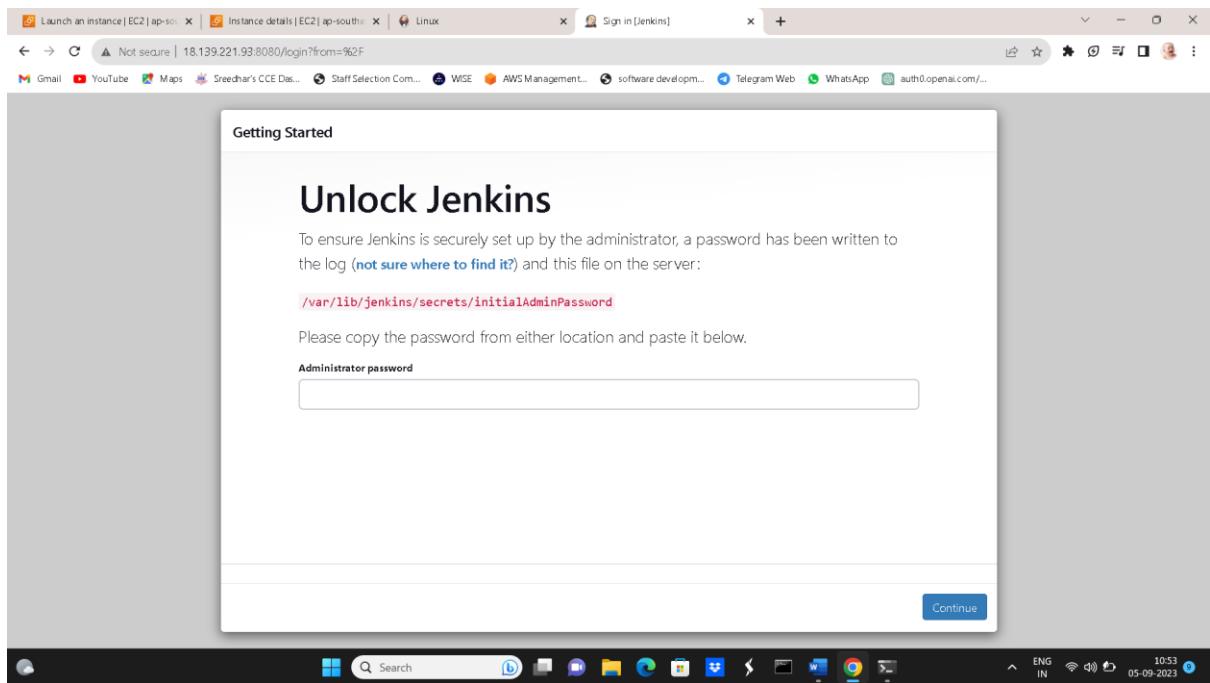
1. Create and launch ec2 -instance and connect it through shell or git bash.
2. Allow the all-traffic in the security groups.
3. Now install Jenkins by using below commands
 - sudo wget -O /etc/yum.repos.d/jenkins.repo \
 https://pkg.jenkins.io/redhat-stable/jenkins.repo
 - sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
 - sudo yum upgrade
 - sudo yum install jenkins
 - sudo systemctl daemon-reload
 - sudo amazon-linux-extras install java-openjdk11
 - sudo systemctl start Jenkins
 - sudo systemctl enable Jenkins
 - sudo systemctl status Jenkins

```

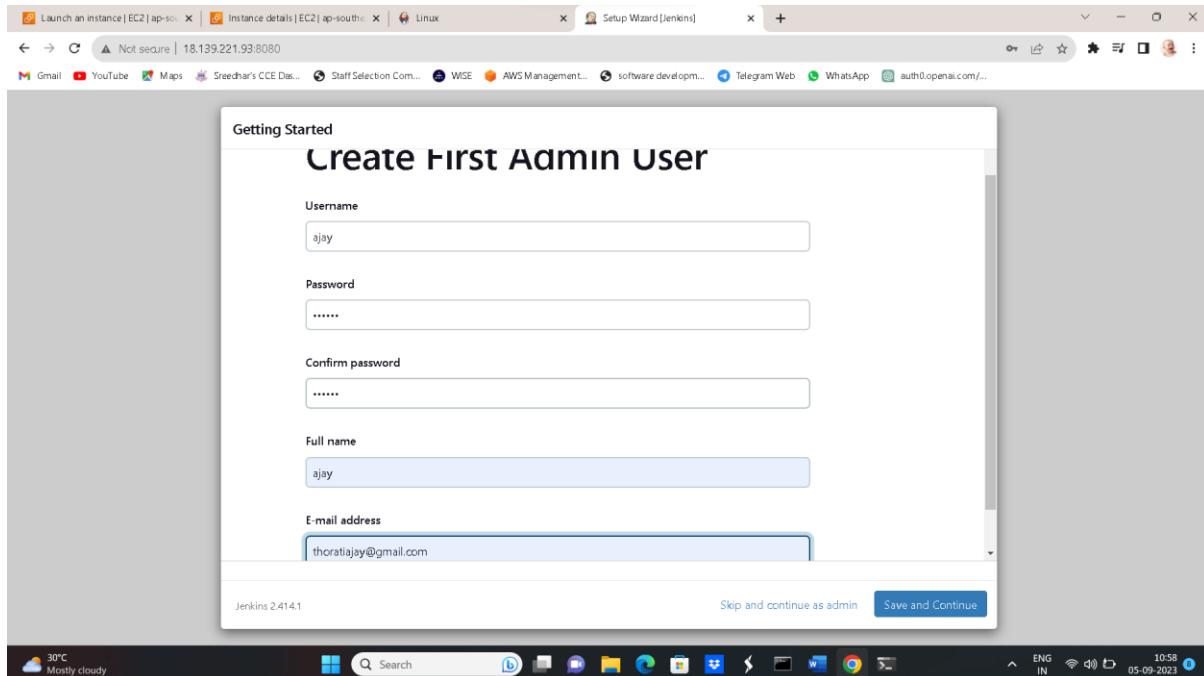
56 redis6 available [ =stable ]
57 ruby3.0 available [ =stable ]
58 postgresql12 available [ =stable ]
59 postgresql13 available [ =stable ]
60 mock2 available [ =stable ]
61 dnsmasq.85 available [ =stable ]
62 kernel-5.15 available [ =stable ]
63 postgresql14 available [ =stable ]
64 firefox available [ =stable ]
65 lustre available [ =stable ]
66 php8.1 available [ =stable ]
67 awscli1 available [ =stable ]
68 php8.2 available [ =stable ]
69 dnsmasq available [ =stable ]
70 unbound1.17 available [ =stable ]
71 golang1.19 available [ =stable ]
72 collectd-python3 available [ =stable ]
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl start jenkins
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2023-09-05 05:21:26 UTC; 35s ago
    Main PID: 3723 (java)
       CGroup: /system.slice/jenkins.service
               └─3723 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 4909be0289cd40ad944fcfe57bd14beb
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:26.781+0000 [id=29]      INFO      jenkins.InitRe...ation
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:26.798+0000 [id=23]      INFO      hudson.lifecyc...ning
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:27.820+0000 [id=45]      INFO      h.m.DownloadSe...aller
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:27.821+0000 [id=45]      INFO      hudson.util.Re...pt #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-0-31-109 ~]$
```

- copy the public ip and browse it along with the port number 8080.



- Now copy the path and go into that path you will get one key copy it and paste it .in the next step you can give name and password then your account gets created.



- Now generate one ssh key in master for all the slaves by using below commands
 - cd .ssh/
 - ssh -keygen -t rsa
 - now ssh key is generated with the name id_rsa.pub

```
ec2-user@ip-10-0-30-3:~/.ssh % + 
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thora>cd Downloads

C:\Users\thora\Downloads>ssh -i "mykeypair.pem" ec2-user@ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com
The authenticity of host 'ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com (13.213.30.174)' can't be established.
ED25519 key fingerprint is SHA256:SyuUpFY2nwRU084FZRp72dHJM9CwvYMR1nCMJLmg2QU.
This host key is known by the following other names/addresses:
  C:\Users\thora\.ssh\known_hosts:159: ec2-54-179-128-5.ap-southeast-1.compute.amazonaws.com
  C:\Users\thora\.ssh\known_hosts:173: ec2-13-229-206-153.ap-southeast-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Mon Sep  4 17:04:26 2023 from 110.225.231.236

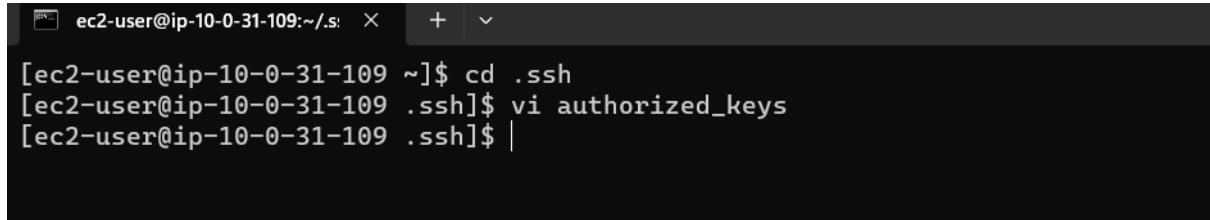
  _|_ --|_
  _| |  /   Amazon Linux 2 AMI
  ___|_|_|_|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-30-3 ~]$ ls
remoting remoting.jar
[ec2-user@ip-10-0-30-3 ~]$ cd .ssh
[ec2-user@ip-10-0-30-3 ~]$ ll
authorized_keys  id_rsa  id_rsa.pub
[ec2-user@ip-10-0-30-3 .ssh]$ |
```

- copy this public key for the future use of slaves connection

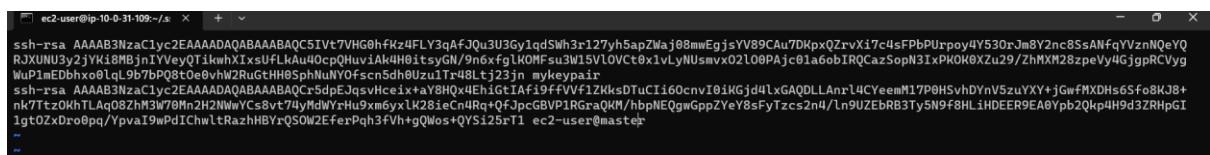
Creating slave nodes:-

1. Susmith create and launch five ec2 instances and connect it through shell .
2. After launching the instance add below commands to connect with master.
 - cd .ssh/
 - vi authorized_keys



```
[ec2-user@ip-10-0-31-109:~/.ssh] $ cd .ssh
[ec2-user@ip-10-0-31-109 .ssh] $ vi authorized_keys
[ec2-user@ip-10-0-31-109 .ssh] $ |
```

- goto end (shift+A) and paste the public key (generated in master id_rsa.pub) and save it.

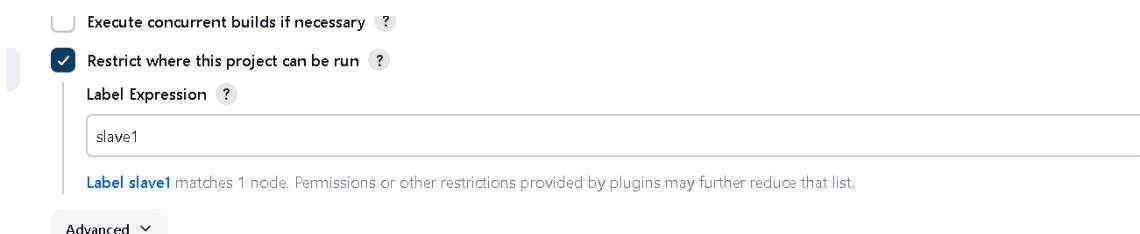


```
ssh-rsa AAAAB3NzaC1yc2EAAADQABAAQCS1Vt7VHGhFkz4FLY3qAfJQn3U3Gy1d5Wh3x127yh5apZWaJ88mwEgjSYV89CAu7DkpX0ZrvXi7c4sFPbPUtpoy4V530rJm8Y2nc85sANfqVznQoYQ
RJXUNU3y2jYK18MBjnIVeyOTikwhXtxsUfLkAu4OcpHuvIAk4H0itsyGN/9n6xfgLKOMFsu3W15VLOVCt0x1vLyNUsmvxO2l0OPAjc01a6ob1RQCazSopN3IxPKOK0Xzu29/ZhMXM28zeV4GjgpRCVyg
WuP1mEDbhxx0lql_9b7bpQ8t0e0vh2RuGtHHSpnNuNOfscn5d0l0zu1Tx48Lj23jn mykeypair
ssh-rsa AAAAB3NzaC1yc2EAAADQABAAQCr5dpEJqsVHceix+aY8HQx4Eh1GtIAfi9ffVVf1ZKksDtuc1i60cnvI0iKGjdulxGAQDLAnx14CYemM17P0hSvhDvnV5zuYXYt+jGwfMXDHs6Sf08KJ8+
nk7Tz0KhTLAg0S2HM3W9Mn2h2NWwCs8vt74yMdWYrHu9xmgyxlK261eCn4Rw+QfJpcGBVP1RGxQKM/hbpNEQgwGppzYeY8sFytzcs2n4/ln9UZEbRB3Ty5N9f8HLiHDEER9EA0Ypb2Qkp4H9d3RHpGI
1gt0Zxbro0pq/Ypvai9wPdIChwltRazhIBYrQSOW2EferrPqh3fVh+gQWos+QYS125rtI ec2-user@master
```

- follow the above steps for all the slaves.

Installing httpd on slave1:

- Ajay Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.



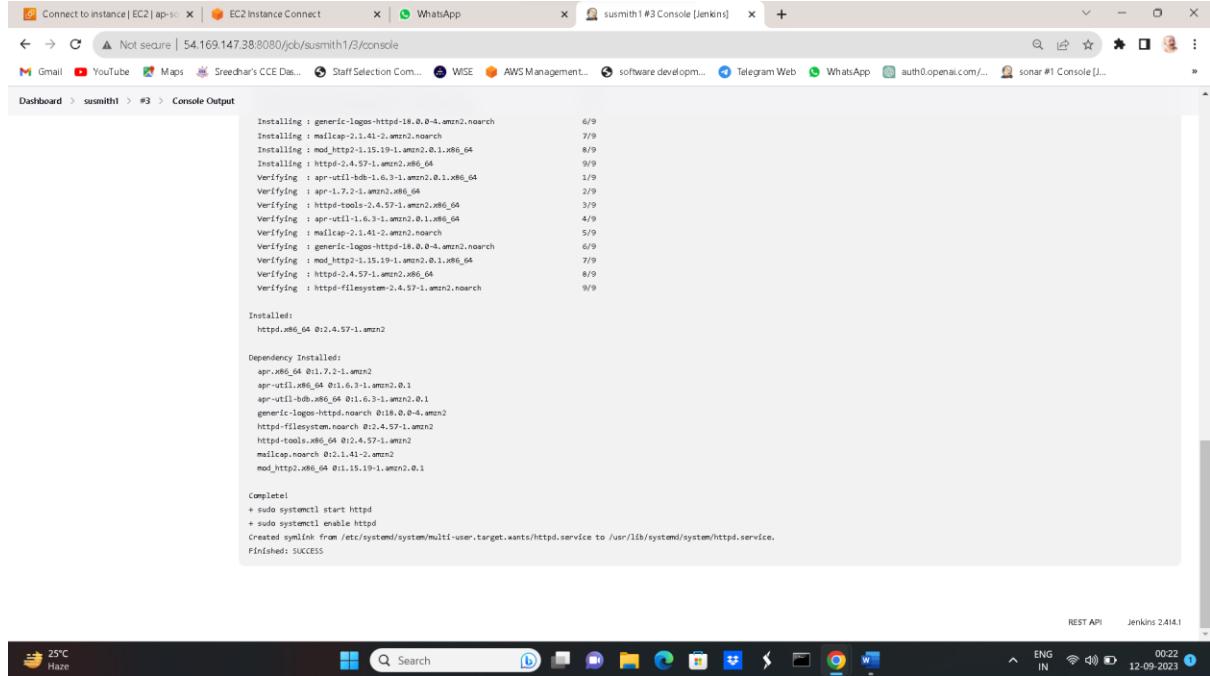
- Add build step to install httpd and start httpd .



The screenshot shows the Jenkins build step configuration for 'Execute shell'. The 'Command' field contains the following script:

```
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

- Save the job and Click on build now .



```

Installing : generic-logs-httdp-18.0.0-4.amzn2.noarch
Installing : mod_mllap-2.1.41-2.amzn2.x86_64
Installing : mod_http2-1.15.19-1.amzn2.x86_64
Installing : httpd-2.4.57-1.amzn2.x86_64
Verifying  : httpd-2.4.57-1.amzn2.x86_64
Verifying  : apr-util-1.7.2-1.amzn2.x86_64
Verifying  : apr-1.7.2-1.amzn2.x86_64
Verifying  : http-tools-2.4.57-1.amzn2.x86_64
Verifying  : apr-util-1.6.3-1.amzn2.x86_64
Verifying  : mllap-2.1.41-2.amzn2.noarch
Verifying  : generic-logs-httdp-18.0.0-4.amzn2.noarch
Verifying  : mod_http2-1.15.19-1.amzn2.x86_64
Verifying  : httpd-2.4.57-1.amzn2.x86_64
Verifying  : httpd-filesystem-2.4.57-1.amzn2.noarch

Installed:
httpd.x86_64 0:2.4.57-1.amzn2

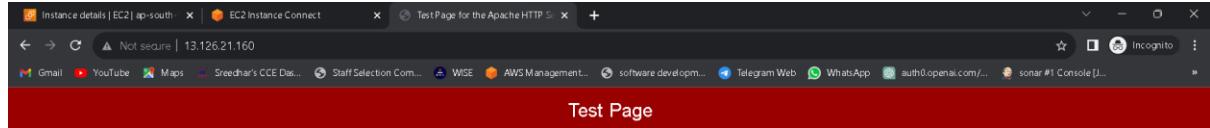
Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
apr-util.x86_64 0:1.6.3-1.amzn2.0.1
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logs-httdp.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.57-1.amzn2
http-tools.x86_64 0:2.4.57-1.amzn2
mllap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.19-1.amzn2.x86_64

Completed!
+ sudo systemctl start httpd
+ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
Finished: SUCCESS

```

REST API Jenkins 2.14.1

- Copy the public IP browse it.



If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



Installing nginx on slave2:

- Ajay Create one job and choose free style project.
 - Select restrict where this project can be run and give label expression.

Restrict where this project can be run ?

Label Expression ?

slave2

Label slave2 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

- Select build step and choose execute shell and add commands to install ngnix and start ngnix.

Build Steps

≡ Execute shell ? X

Command

See the list of available environment variables

```
sudo amazon-linux-extras install nginx1
sudo systemctl start nginx
sudo systemctl enable nginx
```

- Save the job and click on build now .
 - Copy the public ip and browse it.



Installing sonar qube on slave3:

- Ajay Create one job and choose free style project.

- Select restrict where this project can be run and give label expression

Restrict where this project can be run ?

Label Expression ?

slave3

Label slave3 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install and run sonarqube.

≡ Execute shell ?

Command

See the list of available environment variables

```
cd /opt
sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.6.zip
sudo unzip sonarqube-7.6.zip
sudo useradd sonar
sudo chown -R sonar:sonar /opt/sonarqube-7.6
sudo chmod -R 775 /opt/sonarqube-7.6
cd /opt/sonarqube-7.6/bin/linux-x86-64
sudo su sonar -
./sonar.sh start
```

Advanced ▾

Add build step ▾

- Before going to build the job add sonar user in sudoers file.

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
jenkins ALL=(ALL) NOPASSWD:ALL
sonar  ALL=(ALL) NOPASSWD:ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
"/etc/sudoers.tmp" 121L, 4387B
```

- Now save the job and click on build now.
- Copy the public IP and browse it along with port number 9000.

Installing apache tomcat on slave4:

- Ajay Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

- Select build step and choose execute shell and add commands to install and run tomcat server.

The screenshot shows the Jenkins interface for configuring a new job named 'tomcat'. The left sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', **Build Steps** (which is selected), and 'Post-build Actions'. The 'Build Steps' section contains an 'Execute shell' step with the following command:

```

sudo rm -rf *
sudo su -
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz
tar xvzf apache-tomcat-9.0.80.tar.gz
cd apache-tomcat-9.0.80/bin
sh startup.sh

```

Below this is an 'Advanced' dropdown and a 'Save' button.

- Now save the job and click on build now.
- Copy the public ip and browse it along with port number 8080.

The screenshot shows the Apache Tomcat 9.0.80 welcome page. At the top, there's a banner stating 'If you're seeing this, you've successfully installed Tomcat. Congratulations!' with a cartoon cat icon. Below the banner, there's a 'Developer Quick Start' section with links to 'Tomcat Setup', 'First Web Application', 'Realms & AAA', 'JDBC DataSources', 'Examples', 'Servlet Specifications', and 'Tomcat Versions'. To the right, there are three buttons: 'Server Status', 'Manager App', and 'Host Manager'. The main content area is divided into three sections: 'Managing Tomcat', 'Documentation', and 'Getting Help'. The 'Managing Tomcat' section includes links to 'Release Notes', 'Changelog', and 'Migration Guide'. The 'Documentation' section links to 'Tomcat 9.0 Documentation', 'Tomcat 9.0 Configuration', and 'Tomcat Wiki'. The 'Getting Help' section links to 'FAQ and Mailing Lists' and lists several mailing lists: 'tomcat-announce' (announcements), 'tomcat-users' (user support), 'tomcat-dev' (development), 'tomcat-jdbc' (JDBC support), and 'tomcat-jira' (issues).

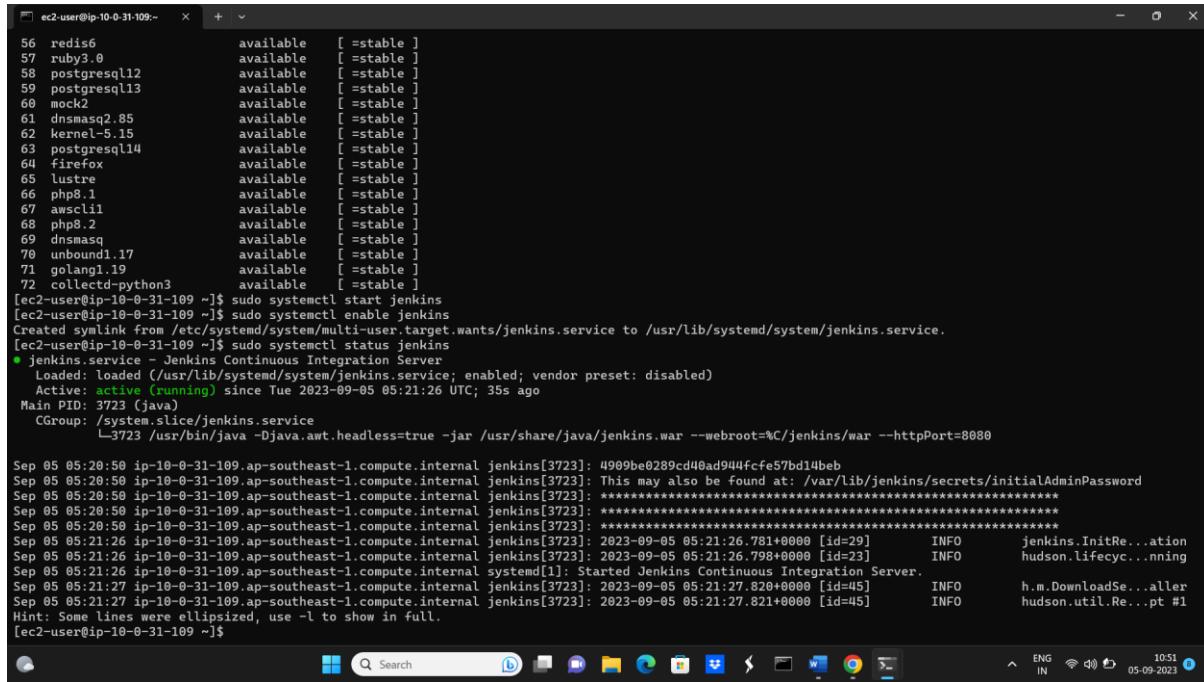
METHOD-4

Creating master-slave configuration AJAY creates one master and Susmith creates five slave nodes and Ajay build the jobs in all slaves using different AWS accounts.

AJAY creates master node:

1. Create and launch ec2 -instance and connect it through shell or git bash.
2. Allow the all-traffic in the security groups.
3. Now install Jenkins by using below commands

- sudo wget -O /etc/yum.repos.d/jenkins.repo \
<https://pkg.jenkins.io/redhat-stable/jenkins.repo>
- sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
- sudo yum upgrade
- sudo yum install jenkins
- sudo systemctl daemon-reload
- sudo amazon-linux-extras install java-openjdk11
- sudo systemctl start Jenkins
- sudo systemctl enable Jenkins
- sudo systemctl status Jenkins

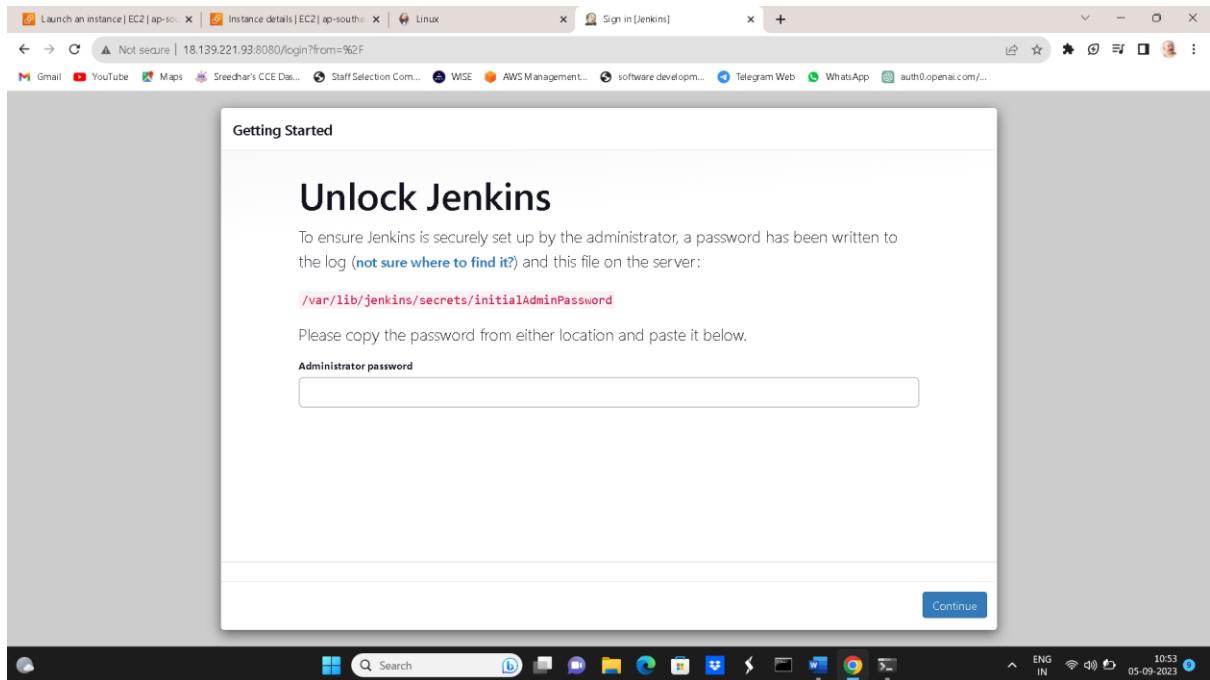


The screenshot shows a terminal window titled "ec2-user@ip-10-0-31-109:~". The window displays a series of log entries from the Jenkins installation process. It starts with a list of available packages (redis6, ruby3.0, postgresql12, postgresql13, mock2, dnsmasq2.85, kernel-5.15, postgresql14, firefox, lustre, php8.1, awscli1, php8.2, dnsmasq, unbound1.17, golang1.19, collectd-python3) followed by the execution of several commands: "sudo systemctl start jenkins", "sudo systemctl enable jenkins", and "sudo systemctl status jenkins". The output indicates that Jenkins is loaded and active (running). The log then continues with several informational messages from the Jenkins service, including logins, startup logs, and system events. At the bottom, there is a note about ellipsized lines and a timestamp of "05-09-2023". The terminal window has a standard Windows-style interface with a taskbar at the bottom.

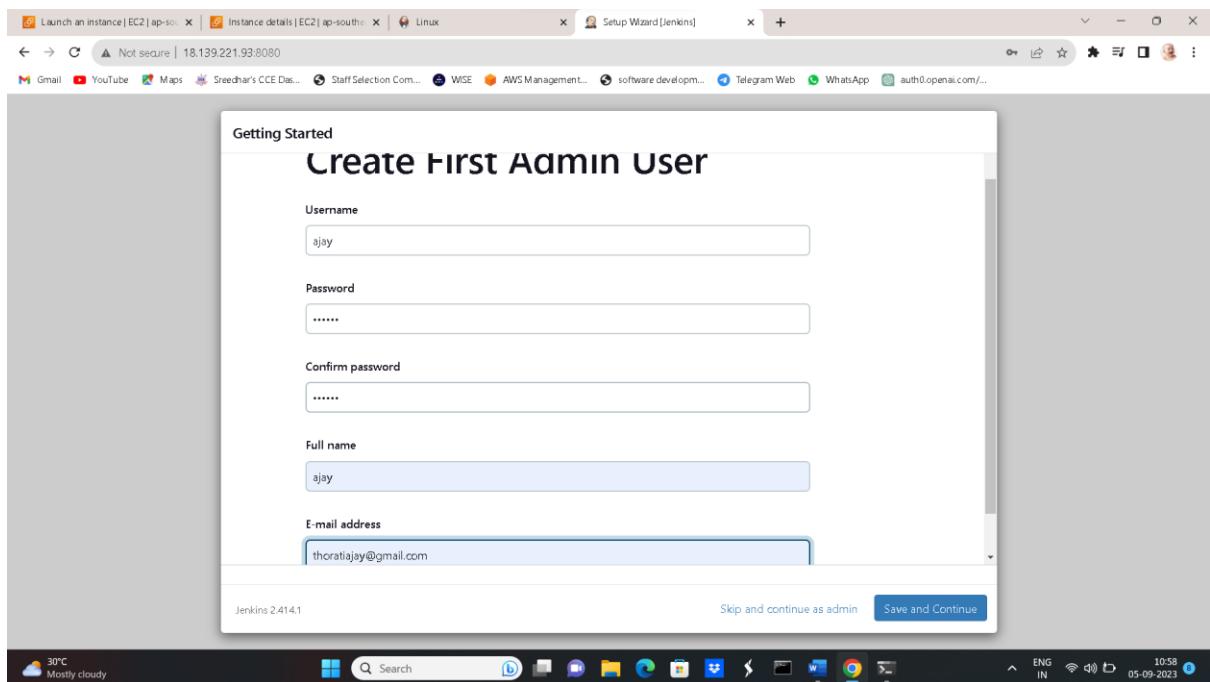
```
56 redis6           available  [ =stable ]
57 ruby3.0          available  [ =stable ]
58 postgresql12      available  [ =stable ]
59 postgresql13      available  [ =stable ]
60 mock2            available  [ =stable ]
61 dnsmasq2.85      available  [ =stable ]
62 kernel-5.15       available  [ =stable ]
63 postgresql14      available  [ =stable ]
64 firefox           available  [ =stable ]
65 lustre            available  [ =stable ]
66 php8.1            available  [ =stable ]
67 awscli1          available  [ =stable ]
68 php8.2            available  [ =stable ]
69 dnsmasq           available  [ =stable ]
70 unbound1.17       available  [ =stable ]
71 golang1.19        available  [ =stable ]
72 collectd-python3  available  [ =stable ]
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl start jenkins
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-10-0-31-109 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2023-09-05 05:21:26 UTC; 35s ago
    Main PID: 3723 (java)
   CGroup: /system.slice/jenkins.service
           └─3723 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 4099be0289cd0ad944fcfe57bd14beb
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 05 05:20:50 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: ****
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:26.781+0000 [id=29]      INFO  jenkins.InitRe..ation
Sep 05 05:21:26 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:26.798+0000 [id=23]      INFO  hudson.lifecyc..nning
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:27.820+0000 [id=45]      INFO  h.m.DownloadSe...aller
Sep 05 05:21:27 ip-10-0-31-109.ap-southeast-1.compute.internal jenkins[3723]: 2023-09-05 05:21:27.821+0000 [id=45]      INFO  hudson.util.Re..pt #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-0-31-109 ~]$
```

- copy the public ip and browse it along with the port number 8080.



- Now copy the path and go into that path you will get one key copy it and paste it .in the next step you can give name and password then your account gets created.



- Now generate one ssh key in master for all the slaves by using below commands
 - cd .ssh/
 - ssh -keygen -t -rsa

- now ssh key is generated with the name id_rsa.pub

```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thora>cd Downloads

C:\Users\thora\Downloads>ssh -i "mykeypair.pem" ec2-user@ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com
The authenticity of host 'ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com (13.213.30.174)' can't be established.
ED25519 key fingerprint is SHA256:SyuUpFy2nwRU084FZRp72dHJM9CwYMR1hCMJLmg2QU.
This host key is known.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-213-30-174.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Mon Sep  4 17:04:26 2023 from 110.225.231.236

|_ _| / Amazon Linux 2 AMI
---|---|_|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-30-3 ~]$ ls
remoting remoting.jar
[ec2-user@ip-10-0-30-3 ~]$ cd .ssh
[ec2-user@ip-10-0-30-3 .ssh]$ ls
authorized_keys id_rsa id_rsa.pub
[ec2-user@ip-10-0-30-3 .ssh]$ |

```

- copy this public key for the future use of slaves connection

Creating slave nodes:-

1. Susmith creates and launch five ec2 instances and connect it through shell .
2. After launching the instances add the public key generated in master node using below commands to connect with master.
 - cd .ssh/
 - vi authorized_keys

```

[ec2-user@ip-10-0-31-109 ~]$ cd .ssh
[ec2-user@ip-10-0-31-109 .ssh]$ vi authorized_keys
[ec2-user@ip-10-0-31-109 .ssh]$ |

```

- goto end (shift+A) and paste the public key (generated in master id_rsa.pub) and save it.

```

Last login: Mon Sep 11 11:04:59 2023 from 10.21.20.211
|_ _| / Amazon Linux 2 AMI
---|---|_|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-36-178 ~]$ ls
[ec2-user@ip-172-31-36-178 ~]$ cd .ssh
[ec2-user@ip-172-31-36-178 .ssh]$ ls
authorized_keys
[ec2-user@ip-172-31-36-178 .ssh]$ vi authorized_keys
[ec2-user@ip-172-31-36-178 .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABQH5RweswwJMDeqahljFLUPp9LynSnqsBmWeMbZETWnyhButMrcsdvuDhl8Ru9fSTQWLL/nKUzk7MJt7ZA0B0fkwcudNNny6nJKWpETLlzR474LuFLHqxfJMWzquCb8TQOARUd
R6Gb1j3nH1z1Bf11FVlghcdht782+NUVpJ1RKvK7b13CiFlcqXJdvrJVB7+xSzjz8CNP9cfKKXnBENJKS5sc+bk4FNv1Ut4CTVjzu8mB2PDxw/T/eqjeWasXal+F1kICNpY6nc1Mlk6yelVtPTWubw77lmnMjseQsvrd
MmJC4G8WKeqYcltOfN0os5c2+WEduAlnJ7/ m5
ssh-rsa AAAAB3NzaC1yc2EAAAQABQjmkKG1g7BUUWcjnywQl2ci+PbJP6gAkpgVvgsHRpZEMD9QKnlz4a33yLCVq8iM6VkkJMD2hcof3AK1kohGIXLE5fr462FFbDCHKZuH0CCS02q5Mp/KaD5cuW90Vbx
vq/fvZu0H1D7lqvKj5kMqTEjextbJ05ih9sF4/EZN+8oV7eotu9cAeOpfIowEgsz2qT6+6onKffBqencZunGtExIfLas7fuA715z3m17N8Ljf3DRGNQov/7psZYi7/2kdyVheOpLHS5mf97lHjtWcd06/8ngQVSTi
dhAKWv+TBp1jUCWftNSThMEU97A2j+1gnceR ec2-user@ip-10-0-18-37.ap-southeast-1.compute.internal
[ec2-user@ip-172-31-36-178 .ssh]$ |

```

2. Now configure new nodes in the slaves

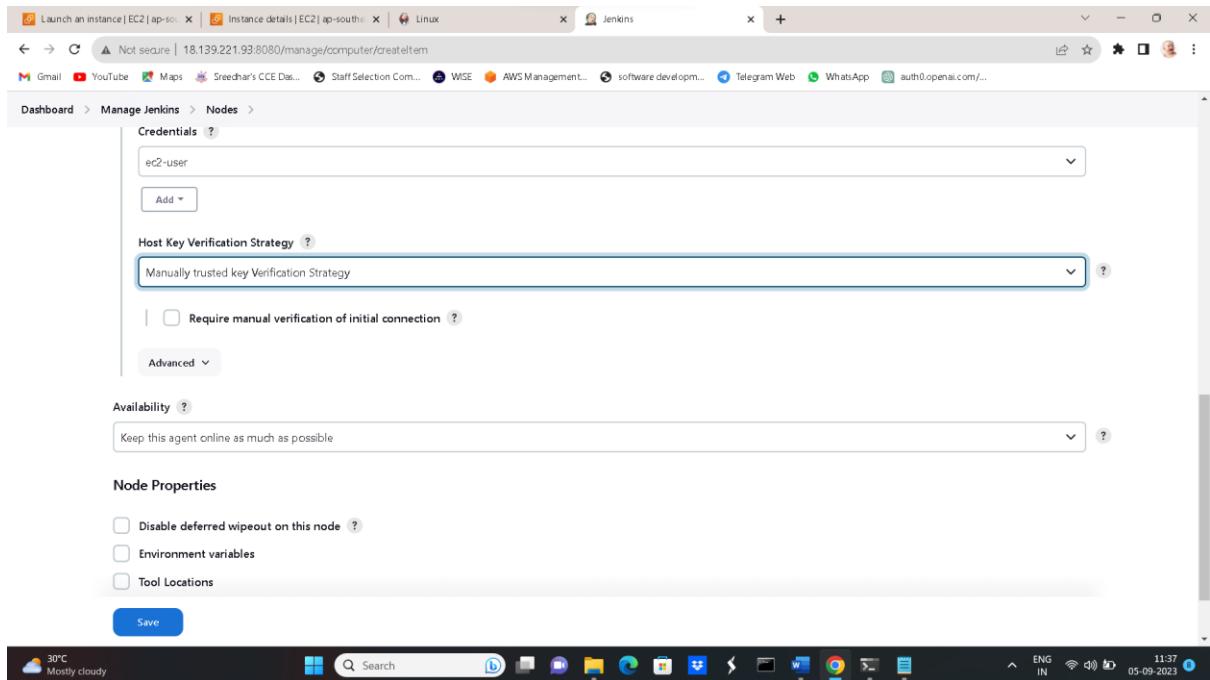
- Goto manage Jenkins → nodes → give name and select permanent node → click on create node
- No of executors = 10
- Remote root directory: /home/ec2-user
- Labels: slave1 (provide label name)
- Usage: use this node as much as possible
- Launch method: launch agent via ssh
- Host: provide ipv4 public IP of your slave instance

The screenshot shows the Jenkins 'Configure' page for a node named 'susmith1'. The page includes fields for Name, Description, Number of executors, Remote root directory, Label, and Usage. A sidebar displays the build executor status with 8 idle slots.

- In credentials you need to create your credentials click on add select Jenkins kind → ssh username and private key, username → ec2-user, key → copy your pem file and paste here. click on create and select that created credentials.

The screenshot shows the Jenkins 'Add Credentials' page. It includes fields for Domain, Kind, Scope, ID, and Description.

- Host key verification strategy: manually trusted key verification strategy
- Click on save



- Now your node created successfully and your agent connected and online.

```

Instances | EC2 | ap-southeast-1 | susmith1 [Jenkins] | WhatsApp | + | 
← → ⌂ Not secure | 54.169.147.38:8080/computer/susmith1/log
Gmail YouTube Maps Sreedhar's CCE Das... Staff Selection Com... WISE AWS Management... software developm... Telegram Web WhatsApp auth0.openai.com/...
Dashboard > Nodes > susmith1 > Log
[09/11/23 15:54:38] [SSH] Checking java version of java
[09/11/23 15:54:38] [SSH] java -version returned 11.0.20.
[09/11/23 15:54:38] [SSH] Starting sftp client.
[09/11/23 15:54:38] [SSH] Copying latest remoting.jar...
[09/11/23 15:54:41] [SSH] Copied 1,371,113 bytes.
Expanded the channel window size to 4MB
[09/11/23 15:54:41] [SSH] Starting agent process: cd "/home/ec2-user" && java -jar remoting.jar -workDir /home/ec2-user -jar-cache /home/ec2-user/remoting/jarCache
Sep 11, 2023 3:54:41 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory
Sep 11, 2023 3:54:41 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/remoting
<==[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 3131.vf2b_b_790b_ce99
Launcher: SSHLauncher
Communication Protocol: Standard in/out
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 stdbuf
Agent successfully connected and online

```

- Follow the above process for all slaves create nodes in every slave.

Installing httpd on slave1:

- Ajay Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

Restrict where this project can be run ?

Label Expression ?

susmith1

Label susmith1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Add build step to install httpd and start httpd .

Build Steps

Execute shell ?

Command

See the list of available environment variables

```
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

Advanced ▾

Add build step ▾

- Save the job and Click on build now .

```
Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch          6/9
Installing : mod_mllap-2.1.41-2.amzn2.noarch                   7/9
Installing : mod_http2-1.15.19-1.amzn2.0.1.x86_64            8/9
Installing : http-2.4.57-1.amzn2.x86_64                     9/9
Verifying : apr-util-bdb-1.6.3-3.amzn2.0.1.x86_64           1/9
Verifying : apr-1.7.2-1.amzn2.x86_64                         2/9
Verifying : httpd-tools-2.4.57-1.amzn2.x86_64              3/9
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64              4/9
Verifying : mllap-2.1.41-2.amzn2.noarch                    5/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch       6/9
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64           7/9
Verifying : http-2.4.57-1.amzn2.x86_64                     8/9
Verifying : httpd-filesystem-2.4.57-1.amzn2.noarch         9/9

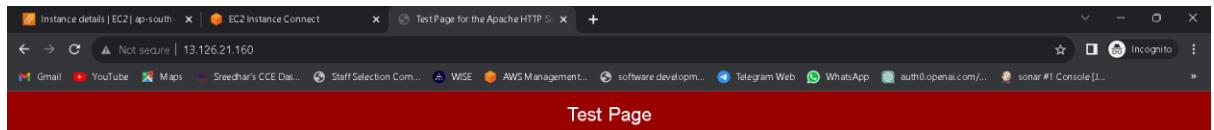
Installed:
httpd.x86_64 0:2.4.57-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
apr-util.x86_64 0:1.6.3-1.amzn2.0.1
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.57-1.amzn2
httpd-tools.x86_64 0:2.4.57-1.amzn2
mllap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Completed!
+ sudo systemctl start httpd
+ sudo systemctl enable httpd
created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
Finished: SUCCESS
```



- Copy the public IP browse it.



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to ["webmaster@example.com"](mailto:webmaster@example.com).

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the [next section](#).

More configuration information can be found in the Apache HTTP Server documentation.



Installing nginx on slave2:

- Ajay Create one job and choose free style project.
 - Select restrict where this project can be run and give label expression.

Restrict where this project can be run ?

Label Expression ?

susmith2

Label susmith2 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install ngnix and start ngnix.

Build Steps

The screenshot shows the Jenkins interface for configuring build steps. A single 'Execute shell' step is defined with the following command:

```
sudo yum update  
sudo amazon-linux-extras install -y nginx1  
sudo service nginx start
```

Below the command, there is an 'Advanced' dropdown and a 'Add build step' button.

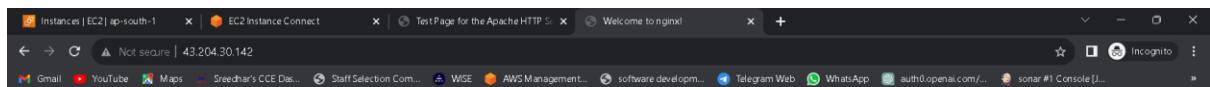
- Save the job and click on build now .

The screenshot shows the Jenkins console output window for a build step. The output displays the results of the commands run in the shell:

```
45 haproxy2      available  [ *stable ]  
46 collectedd   available  [ *stable ]  
47 axs-nitro-enclaves-clif available  [ *stable ]  
48 R4           available  [ *stable ]  
— kernel-5.4     available  [ *stable ]  
50 selinuxng    available  [ *stable ]  
51 tpp8_0        available  [ *stable ]  
52 tomcat9      available  [ *stable ]  
53 unbound1.13   available  [ *stable ]  
54 mariadb10.5   available  [ *stable ]  
55 kernel-5.10-latest enabled   [ *stable ]  
56 redis6       available  [ *stable ]  
57 ruby3.0       available  [ *stable ]  
58 *postgresql12 available  [ *stable ]  
59 *postgresql13 available  [ *stable ]  
60 mock2         available  [ *stable ]  
61 dnsmasq;#5   available  [ *stable ]  
62 kernel-5.15   available  [ *stable ]  
63 postgresql14  available  [ *stable ]  
64 firefox       available  [ *stable ]  
65 lxc          available  [ *stable ]  
66 tpp8_1        available  [ *stable ]  
67 avscif1      available  [ *stable ]  
68 tpp8_2        available  [ *stable ]  
69 dnsmasq      available  [ *stable ]  
70 unbound1.17   available  [ *stable ]  
71 *polargraph19 available  [ *stable ]  
72 collected-python3 available  [ *stable ]  
+ Note on end-of-support. Use 'Info' subcommand.  
+ sudo service nginx start  
Redirecting to /bin/systemctl start nginx.service  
Finished: SUCCESS
```

At the bottom of the window, it says 'REST API Jenkins 2.141.1'. The system tray at the bottom of the screen shows a weather icon (25°C Haze), a search bar, and other system icons.

- Copy the public IP and browse it.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Installing sonar qube on slave3:

- Ajay Create one job and choose free style project.
- Select restrict where this project can be run and give label expression

Restrict where this project can be run ?

Label Expression ?

susmith3

Label susmith3 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install and run sonarqube.

Build Steps

Execute shell ?

Command

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

```
cd /opt
sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.6.zip
sudo unzip sonarqube-7.6.zip
sudo useradd sonar
sudo chown -R sonar:sonar /opt/sonarqube-7.6
sudo chmod -R 775 /opt/sonarqube-7.6
cd /opt/sonarqube-7.6/bin/linux-x86-64
sudo su sonar ./sonar.sh start
```

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

- Before going to build the job add sonar user in sudoers file.

```

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)          ALL
jenkins ALL=(ALL) NOPASSWD:ALL
sonar  ALL=(ALL) NOPASSWD:ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

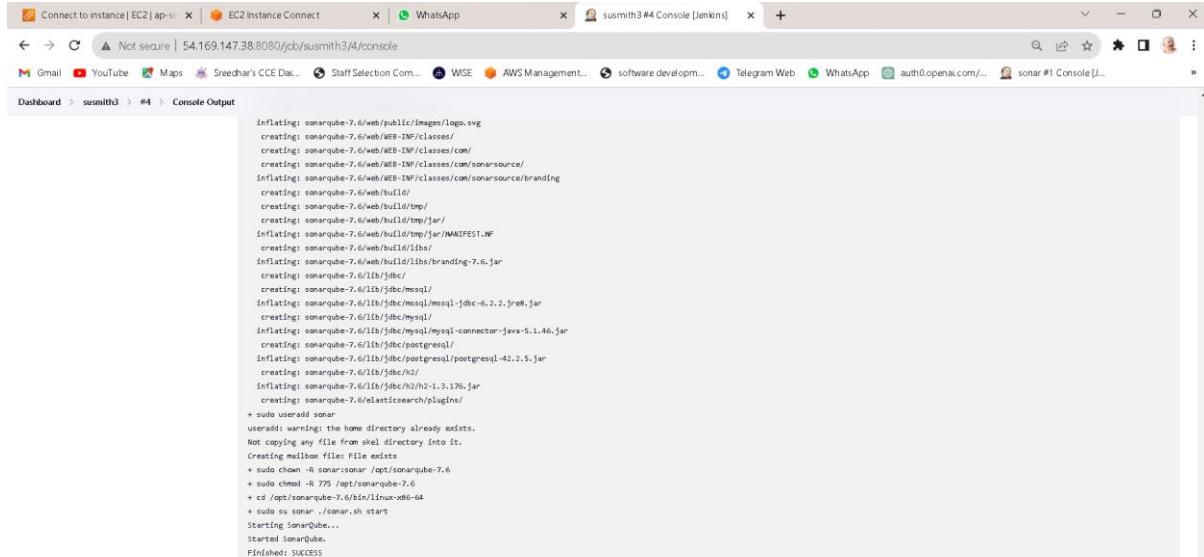
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)          ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
"/etc/sudoers.tmp" 121L, 4387B

```

- Now save the job and click on build now.



```

Dashboard > susmith3 > #4 > Console Output
Inflating: sonarqube-7.6/web/public/images/logo.svg
creating: sonarqube-7.6/web/WEB-INF/classes/
creating: sonarqube-7.6/web/WEB-INF/classes/com/
creating: sonarqube-7.6/web/WEB-INF/classes/com/sonarsource/
�flating: sonarqube-7.6/web/WEB-INF/classes/com/sonarsource/branding
creating: sonarqube-7.6/web/build/
creating: sonarqube-7.6/web/build/tmp/
creating: sonarqube-7.6/web/build/tmp/jar/
�flating: sonarqube-7.6/web/build/tmp/MANIFEST.MF
creating: sonarqube-7.6/web/build/libs/
�flating: sonarqube-7.6/web/build/libs/branding-7.6.jar
creating: sonarqube-7.6/lib/jdbc/
creating: sonarqube-7.6/lib/jdbc/msql/
�flating: sonarqube-7.6/lib/jdbc/msql/mysql-jdbc-6.2.2.jar
creating: sonarqube-7.6/lib/jdbc/mysql/
�flating: sonarqube-7.6/lib/jdbc/mysql/mysql-connector-java-5.1.46.jar
creating: sonarqube-7.6/lib/postgresql/
�flating: sonarqube-7.6/lib/postgresql/postgresql-42.2.5.jar
creating: sonarqube-7.6/lib/h2/
�flating: sonarqube-7.6/lib/h2/h2-1.3.176.jar
creating: sonarqube-7.6/elasticsearch/plugins/
+ sudo useradd sonar
useradd: warning: the home directory already exists.
Not copying any file from /skel directory into it.
Creating mailbox file: /file edits
+ sudo chown -R sonar:sonar /opt/sonarqube-7.6
+ sudo chmod -R 775 /opt/sonarqube-7.6
+ cd /opt/sonarqube-7.6/bin/linux-x86-64
+ sudo su sonar
./sonar.sh start
Starting SonarQube...
started SonarQube.
Finished: SUCCESS

```

REST API Jenkins 2.14.1



- Copy the public ip and browse it along with port number 9000.

Continuous Code Quality

0 Projects Analyzed

0 Bugs
0 Vulnerabilities
0 Code Smells

Multi-Language

20+ programming languages are supported by SonarQube thanks to our in-house code analyzers, including:

Java	C/C++	C#	COBOL	ABAP	HTML	RPG	JavaScript	TypeScript	Objective C	XML
VB.NET	PL/SQL	T-SQL	Flex	Python	Groovy	PHP	Swift	Visual Basic	PL/I	

Quality Model

Bugs track code that is demonstrably wrong or highly likely to yield unexpected behavior.

Vulnerabilities are raised on code that is potentially vulnerable to exploitation by hackers.

Code Smells will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.

Installing apache tomcat on slave4:

- Ajay Create one job and choose free style project.
- Select restrict where this project can be run and give label expression.

Restrict where this project can be run ?

Label Expression ?

susmith4

Label susmith4 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

- Select build step and choose execute shell and add commands to install and run tomcat server.

Build Steps

```
sudo wget https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.93/bin/apache-tomcat-8.5.93.zip
sudo unzip apache-tomcat-8.5.93.zip
cd apache-tomcat-8.5.93/bin/
sudo chmod +x *
sudo ./startup.sh
```

Advanced ▾

Add build step ▾

- Now save the job and click on build now.

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | WhatsApp | susmith4 #1 Console [Jenkins] | +

Dashboard > susmith4 > #1 > Console Output

```

Inflating: apache-tomcat-8.5.99/webapps/host-manager/WEB-INF/context.xml
Inflating: apache-tomcat-8.5.99/webapps/host-manager/WEB-INF/jsp/401.jsp
Inflating: apache-tomcat-8.5.99/webapps/host-manager/WEB-INF/jsp/404.jsp
Inflating: apache-tomcat-8.5.99/webapps/host-manager/WEB-INF/manager.xml
Inflating: apache-tomcat-8.5.99/webapps/host-manager/WEB-INF/web.xml
Inflating: apache-tomcat-8.5.99/webapps/host-manager/css/manager.css
Inflating: apache-tomcat-8.5.99/webapps/host-manager/images/xfi-logo.svg
Inflating: apache-tomcat-8.5.99/webapps/host-manager/images/tomcat.svg
Inflating: apache-tomcat-8.5.99/webapps/host-manager/index.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/context.xml
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/401.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/403.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/404.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/connectorcerts.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/connectorciphers.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/connectortrustedcerts.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/sessiondetails.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/jsp/sessionslist.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/WEB-INF/web.xml
Inflating: apache-tomcat-8.5.99/webapps/manager/css/manager.css
Inflating: apache-tomcat-8.5.99/webapps/manager/images/xfi-logo.svg
Inflating: apache-tomcat-8.5.99/webapps/manager/images/tomcat.svg
Inflating: apache-tomcat-8.5.99/webapps/manager/index.jsp
Inflating: apache-tomcat-8.5.99/webapps/manager/status.xml
Inflating: apache-tomcat-8.5.99/webapps/manager/form.xml
+ cd apache-tomcat-8.5.99/bin/
+ sudo chmod +x bootstrap.sh catalina.bat catalina.sh catalina-tasks.xml ciphers.bat ciphers.sh commons-daemon.jar commons-daemon-native.tar.gz configtest.bat configtest.sh daemon.sh digest.bat digest.sh setclasspath.bat setclasspath.sh shutdown.bat shutdown.sh startup.bat startup.sh tomcat-juli.jar tomcat-native.tar.gz tool-wrapper.bat tool-wrapper.sh version.bat version.sh
+ sudo ./startup.sh
Tomcat started.
Finished: SUCCESS

```

REST API Jenkins 2.14.1



- Copy the public ip and browse it along with port number 8080.

Instances | EC2 | ap-south-1 | EC2 Instance Connect | Test Page for the Apache HTTP Server | Welcome to nginx! | Apache Tomcat/8.5.93 | Incognito | :ENG IN 23:38 11-09-2023

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.5.93

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

Server Status Manager App Host Manager

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the [manager.webapp](#) is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`. In Tomcat 8.5 access to the manager application is split between different users. [Read more...](#)

Release Notes
Changelog
Migration Guide
Compatibility Notes

Documentation
[Tomcat 8.5 Documentation](#)
[Tomcat 8.5 Configuration](#)
[Tomcat Wiki](#)
Find additional important configuration information in:
`$CATALINA_HOME RUNNING.txt`
Developers may be interested in:
[Tomcat 8.5 Bug Database](#)
[Tomcat 8.5 JavaDocs](#)
[Tomcat 8.5 Git Repository at GitHub](#)

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
[tomcat-users](#)
User support and discussion.
[taglibs-user](#)
User support and discussion for [Apache Taglibs](#).
[tomcat-dev](#)
Development mailing list, including commit messages.

25°C Haze