

JENKINS

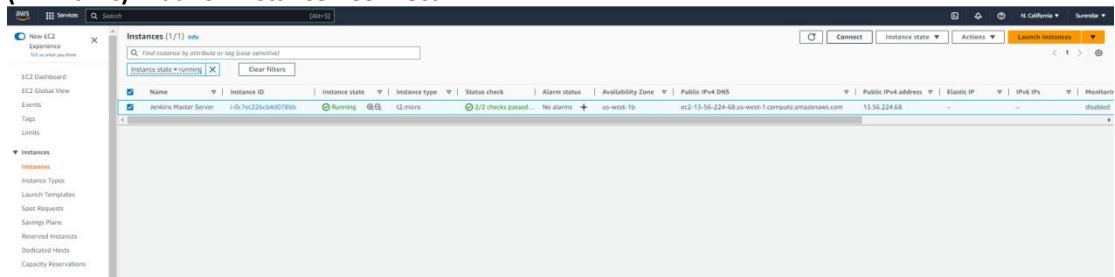
## **Definition:**

Jenkins is an open source **continuous integration/continuous delivery** and **deployment (CI/CD) automation software DevOps tool** written in the **Java** programming language. It is used to implement CI/CD workflows, called **pipelines**.

## **Day-1 - Email Notification.**

## STEPS

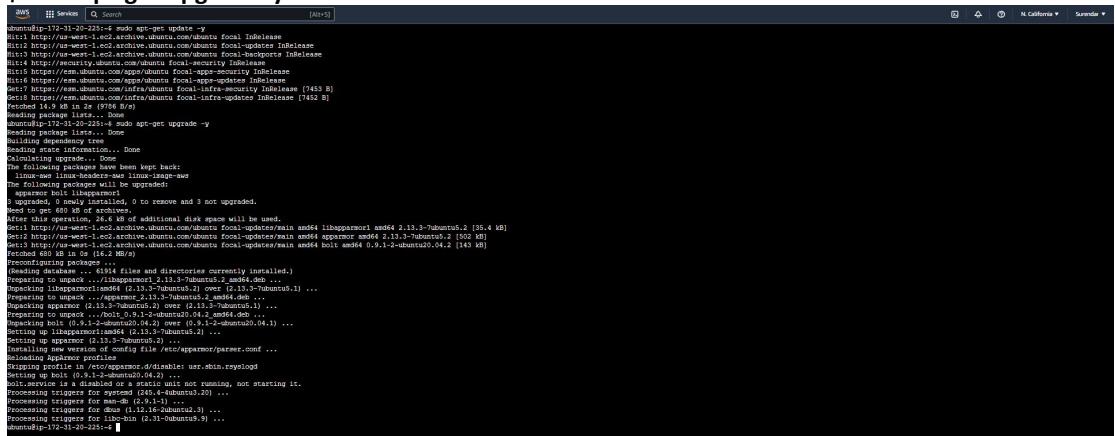
Create an EC2 instances - Ubuntu 20.04 - Create both Key Pair .ppk and connect with pemfile - SG (All Traffic) - Launch Instance - Connect.



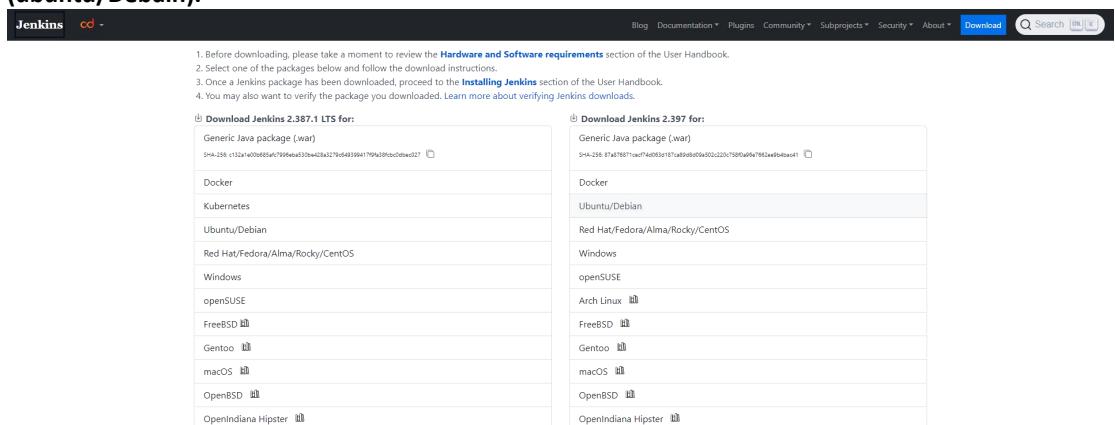
**As it is in normal user - Update Packages and Upgrade.**

```
$sudo apt-get update -y
```

```
$sudo apt-get upgrade -y
```



In Chrome - Search ([jenkins.io](https://jenkins.io)) - Download - Choose stable latest version Jenkins 2.397 (ubuntu/Debian).





```

Update your local package index, then finally install Jenkins:
sudo apt-get update
sudo apt-get install fontconfig openjdk-11-jre
sudo apt-get install jenkins

ubuntu@ip-172-31-20-225:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 97.9 MB of archives.
After this operation, 99.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

**After Successfully installed jenkins**

**To check the jenkins portnumber to connect the jenkins dashboard in webserver**

**\$ps -ef | grep jenkins**

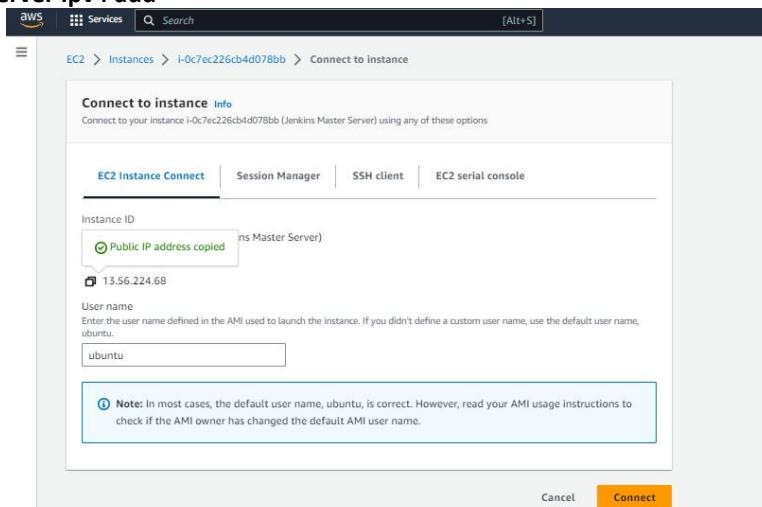
**httpPort=8080**

```

aws Services Search [Alt+S]
ubuntu@ip-172-31-20-225:~$ ps -ef | grep jenkins
jenkins   7993      1  4 06:29 ?    00:00:56 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
ubuntu    8322    3455  0 06:52 pts/0    00:00:00 grep --color=auto jenkins
ubuntu@ip-172-31-20-225:~$ 

```

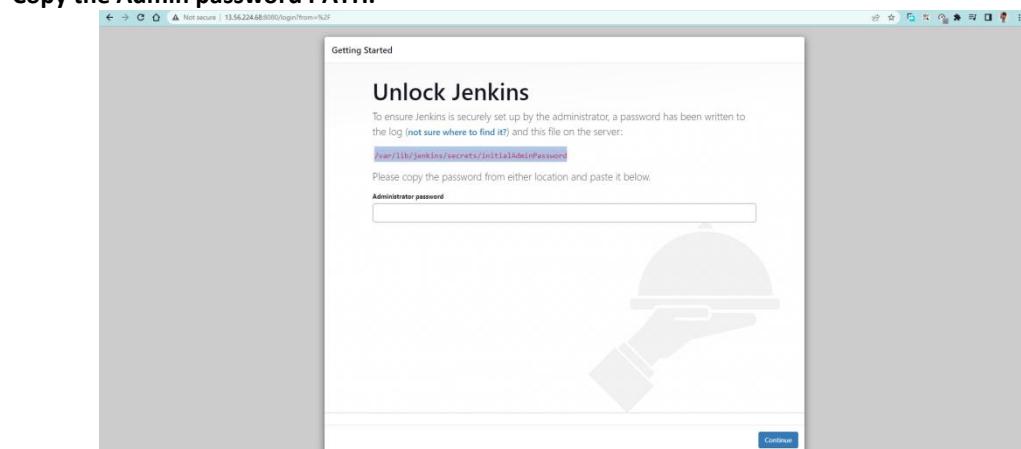
**Copy Master Server Ipv4 add**



**Paste Master Ipv4 In chrome tab mention then jenkins portno as :8080**



**Copy the Admin password PATH.**

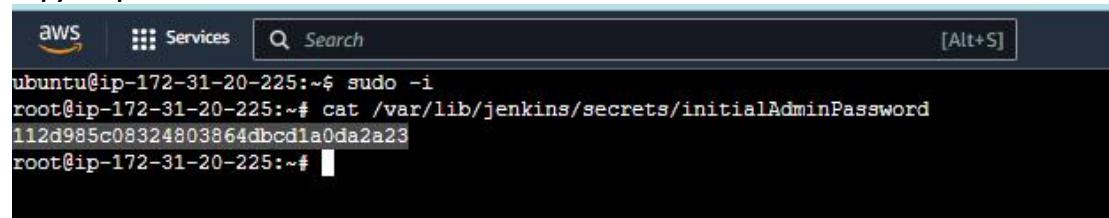


In Master Server

\$sudo -l

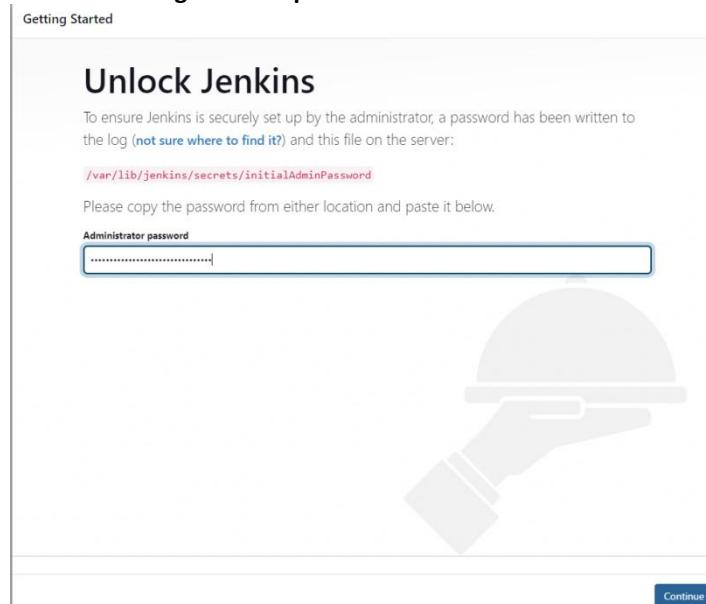
#cd <paste the Default Administrator Password path>

Copy the password.



```
ubuntu@ip-172-31-20-225:~$ sudo -i
root@ip-172-31-20-225:~# cat /var/lib/jenkins/secrets/initialAdminPassword
112d985c08324803864dbcda10da2a23
root@ip-172-31-20-225:~#
```

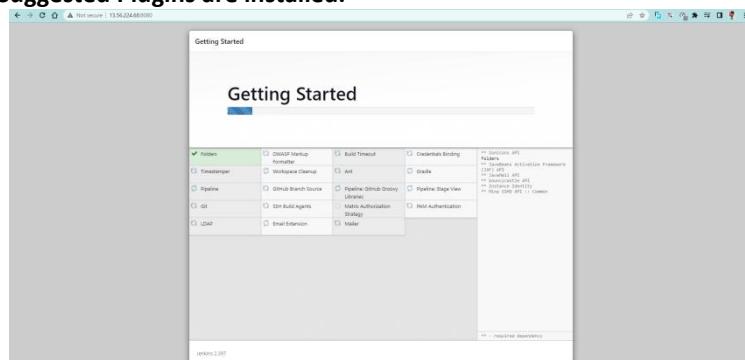
Paste the password in Jenkins Login Admin password field - Continue.



Click Install Suggested Plugins



Wait till all the Suggested Plugins are installed.



## Enter the first Admin User Credentials - Save and Continue

Getting Started

### Create First Admin User

Username: Surey

Password:

Confirm password:

Full name: surendar

E-mail address: surendarsurya72@gmail.com

Jenkins 2.397 Skip and continue as admin Save and Continue

## Save and Finish.

Getting Started

### Instance Configuration

Jenkins URL: http://13.56.224.68:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD\_URL environment variable provided to build steps.

The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.397 Not now Save and Finish

Jenkins

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

New Item

People

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Create a job

Set up a distributed build

Configure a cloud

Learn more about distributed builds

Dashboard > Add description

Search (CTRL+K)

surendar log out

Successfully Configured in Master Server and Login the Jenkins Dashboard page.

## STEPS TO CONFIGURE EMAIL NOTIFICATION FOR JOB FAILURE.

### In Jenkins Dashboard Manage Jenkins - System

The screenshot shows the Jenkins Manage Jenkins interface. In the top right, there are links for 'Search (CTRL+K)', 'surendar', and 'log out'. Below that is a search bar labeled 'Search settings'. On the left, there's a sidebar with links for 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is selected and highlighted in grey), and 'My Views'. The main content area has a yellow banner at the top stating: 'Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)'. Below the banner are four sections: 'System' (Configure global settings and paths.), 'Tools' (Configure tools, their locations and automatic installers.), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins.), and 'Nodes and Clouds' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on.). At the bottom of the main content area, there are buttons for 'Set up agent', 'Set up cloud', and 'Delete'.

### Scroll down to Email Notification

In SMTP server

Enter **smtp.gmail.com**

Default e-mail suffix

Enter **@gmail.com**

The screenshot shows the 'System' configuration page in Jenkins. At the top, it says 'E-mail Notification'. Under 'SMTP server', the value 'smtp.gmail.com' is entered. Under 'Default user e-mail suffix', the value '@gmail.com' is entered. The URL in the browser is '13.56.224.68:8080/manage/configure'.

### In google account

Security - 2 step verification - scroll down to app password - Set the app password (Mail - Windows Computer) - generate.

The screenshot shows the 'App passwords' page in Google Accounts. It says 'You don't have any app passwords.' and 'Select the app and device you want to generate the app password for.' with dropdowns for 'Mail' and 'Windows Computer'. A blue 'GENERATE' button is visible. The URL is 'myaccount.google.com/appspasswords?utm\_source=google-account&utm\_medium=myaccountsecurity&utm\_campaign=tss-settings&rapt=AjHl4OmKPGgrWNvewBk41eX-OwlabBWtvOH29CzWuHoljnODfcuF5RzCSDV8...'. The browser title is 'Google Account'.

### Copy the Password.

Generated app password

Your app password for Windows Computer

**zhyk nfwf rcga falt**

### How to use it

1. Open the "Mail" app.
2. Open the "Settings" menu.
3. Select "Accounts" and then select your Google Account.
4. Replace your password with the 16-character password shown above.

Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

[Learn more](#)

DONE

The screenshot shows a 'Add your Google account' form. It asks for 'Email address' (securerally@gmail.com) and 'Password' (\*\*\*\*\*). There's also a checkbox for 'Include your Google contacts and calendars' which is unchecked. Below the form, it says 'Enter the information below to connect to your Google account.'

### In Email Notification

#### Advanced - Choose Use SMTP Authentication

##### Username

Enter gmail ID

Ex:surendarsurya72@gmail.com

##### Password

Paste App password

##### Choose Use SSL

##### SMTP Port

465

#### Choose Test configuration by sending e-mail

##### Recipient

Enter Same gmail ID

##### Click Test configuration

Ensure Email was successfully sent

#### Verify in Gmail account if the test mail is received.

The screenshot shows the Jenkins 'System' configuration page under 'Manage Jenkins'. The 'Email Configuration' section is selected. The 'Use SMTP Authentication' checkbox is checked. The 'User Name' field contains 'surendarsurya72@gmail.com'. The 'Password' field is masked. The 'Use SSL' checkbox is checked, while 'Use TLS' is unchecked. The 'SMTP Port' field contains '465'. The 'Reply-To Address' field is empty. The 'Charset' field contains 'UTF-8'. The 'Test configuration by sending test e-mail' checkbox is checked, and the 'Test e-mail recipient' field also contains 'surendarsurya72@gmail.com'. Below this, a message says 'Email was successfully sent' with a 'Test configuration' button. At the bottom are 'Save' and 'Apply' buttons. The browser toolbar at the top includes icons for back, forward, search, and refresh.

Test email #1 [Inbox X]

address not configured yet <surendarsurya72@gmail.com>  
to me ▾ 12:41

This is test email #1 sent from Jenkins

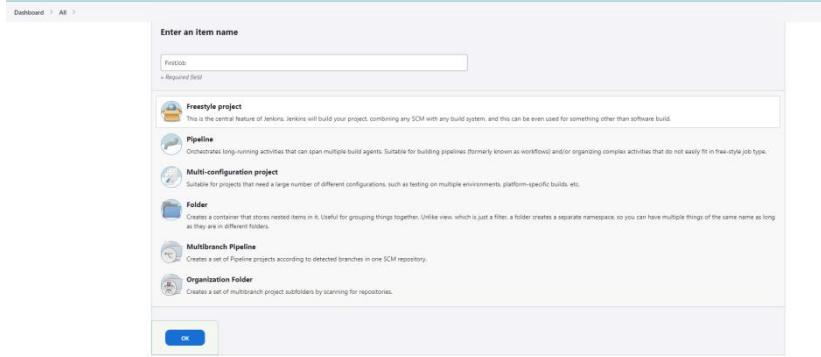
Reply Forward

Successfully Configured Email Notification om Jenkins Dashboard.

Now Set a new 2 jobs and add configured Email-ID to get an E-mail if job is failure.

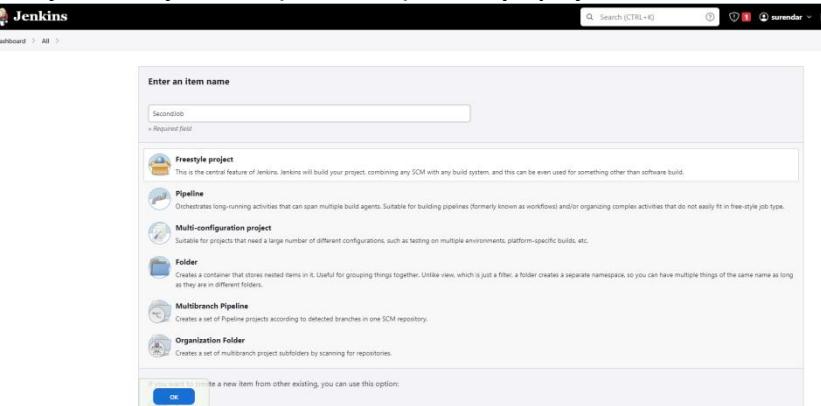
The screenshot shows the Jenkins dashboard. The top navigation bar has the Jenkins logo and the word 'Dashboard'. Below the dashboard, there is a large button with a plus sign and the text 'New Item'.

### Set a job name (firstjob) - select Freestyle project - ok.



In configure - Build Steps - choose execute shell script - enter Wrong cmnd to ensure the email notification working properly when job gets failure - In Post build Action - Enter already configured Email-ID - Save.

### Create another job - Set a job name (SecondJob) - Freestyle project - Ok.



In configure - Build Steps - Choose Execute Shell - Enter working cmnd in ubuntu server Ex: pwd - Save.

In Dashboard - Choose FirstJob configure to set an automation job done when firstjob gets failure.

The screenshot shows the Jenkins dashboard with the 'FirstJob' project selected. A context menu is open over the 'FirstJob' entry in the list, with 'Configure' highlighted. Other options in the menu include 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Rename'. The main dashboard view shows other projects like 'SecondJob' and 'ThirdJob' with their status and build history.

In Configure - Post-build Action - Choose build other projects - Enter the Secondjob name - Choose Trigger even if the build fails - Save.

The screenshot shows the 'Configuration' screen for the 'FirstJob' project under the 'Post-build Actions' section. A sub-menu 'Build other projects' is open, showing the 'Projects to build' field filled with 'SecondJob'. Below it, three trigger options are listed: 'Trigger only if build is stable' (unchecked), 'Trigger even if the build is unstable' (unchecked), and 'Trigger even if the build fails' (checked). At the bottom of the dialog are 'Save' and 'Apply' buttons.

In dashboard - Start the job - Can view the Firstjob gets failure and automatically secondjob get success as we Configure the jobs. Now Ensure Email notification is received for job failure.

The screenshot shows the Jenkins dashboard with the 'FirstJob' project failing (red icon) and the 'SecondJob' project succeeding (green icon). Below the dashboard, an email inbox is displayed with a new message from 'address not configured yet <surendarsurya72@gmail.com>' to 'me <me>'. The message subject is 'Build failed in Jenkins: FirstJob #1'. The message body contains the Jenkins log output:

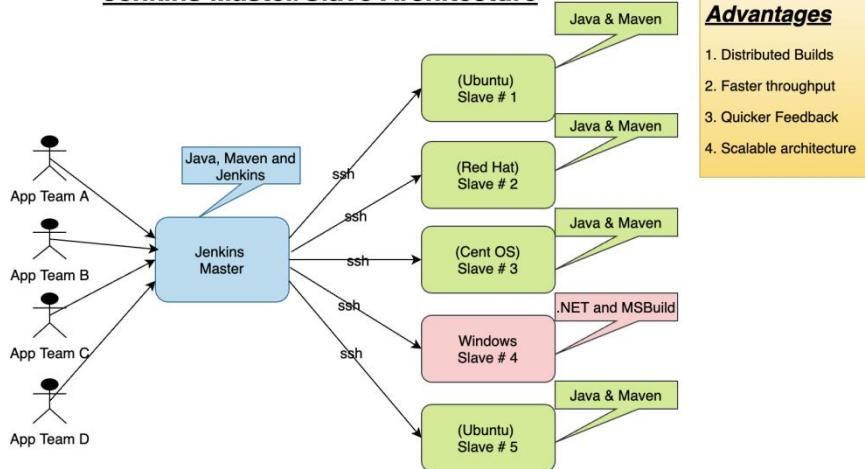
```
Started by user surendar
Running as SYSTEM
Building in workspace <http://13.56.224.68:8080/job/FirstJob/ws/>
[FirstJob] $ /bin/sh -xe /tmp/jenkins15469201796871121121.sh
+ haki
/tmp/jenkins15469201796871121121 sh: 2: haki: not found
Build step 'Execute shell' marked build as failure
```

At the bottom of the email view are 'Reply' and 'Forward' buttons.

Successfully configured the Email notification and Configured Automation Job done using Jenkins. Also received job failure Email notification.

## Master/Slave Server Configuration

### Jenkins Master/Slave Architecture



### **Master and Slave Concept:**

In some cases, files may have different code and require multiple builds and the Jenkins server cannot handle multiple builds simultaneously for this, the Master distributes the workload and allows us to run different builds on different environments each called a Slave.

### **Steps to Configure Master and Slaver Server.**

**Launch 2 Instance - ubunut 20.04 - Key pair create ppk file to connect with putty create pem file  
Choose pem keyfile - Security Group (All Traffic) - Launch an Instances - Rename one server in name of Slave Server - Connect Master server.**

Name	Instance ID	Instance State	Instance Type	Status Check	Alarm Status	Availability Zone	Public IPv4 DNS	Public IPv4 Address	Elastic IP	IPv6 IP	Monitoring
Jenkins Master Server	i-0ccb9d2c2e95a146f	Running	t2.micro	2/2 checks passed...	No alarms	us-west-1b	ec2-13-56-229-253.us-west-1.compute.amazonaws.com	13.56.229.253	-	-	disabled
Slave Server	i-0c797e3fb0b1396	Running	t2.micro	2/2 checks passed...	No alarms	us-west-1b	ec2-54-176-182-99.us-west-1.compute.amazonaws.com	54.176.182.99	-	-	disabled

**Instance: i-0ccb9d2c2e95a146f (Jenkins Master Server)**

**Details** | **Security** | **Networking** | **Status Checks** | **Monitoring** | **Tags**

**Instance summary info**

Instance ID: i-0ccb9d2c2e95a146f (Jenkins Master Server)  
 IP address: 13.56.229.253  
 Hostname type: IP name: ip-172-31-29-204.us-west-1.compute.internal.  
 Answer private resource DNS name: ip-172-31-29-204.us-west-1.compute.internal  
 Instance type: t2.micro  
 Auto-assigned IP address: 13.56.229.253 (Public IP)  
 IAM Role: -  
 Subnet ID: subnet-08089d124d36434f1

**Networking**

Public IPv4 address: 13.56.229.253 | open address  
 Private IPv4 DNS: 172.31.29.204  
 Public IPv4 DNS: ec2-13-56-229-253.us-west-1.compute.amazonaws.com | open address  
 Elastic IP addresses: -  
 AWS Compute Optimizer finding: Opt in to AWS Compute Optimizer for recommendations. | Learn more  
 Auto Scaling Group name: -

## In normal user

### Update and Upgrade the repo packages

```
$sudo apt-get update -y
```

```
$sudo apt-get upgrade -y
```

```
aws Services Search [Alt+S]
applicable law.

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

ubuntu@ip-172-31-29-204:~$ sudo apt-get update -y
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://esm.ubuntu.com/apps/ubuntu focal-apps-security InRelease
Hit:6 https://esm.ubuntu.com/apps/ubuntu focal-apps-updates InRelease
Get:7 https://esm.ubuntu.com/infra/ubuntu focal-infra-security InRelease [7453 B]
Get:8 https://esm.ubuntu.com/infra/ubuntu focal-infra-updates InRelease [7452 B]
Fetched 14.9 kB in 1s (10.0 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Building state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws
The following packages will be upgraded:
  apparmor bolt libapparmor1
3 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 680 kB of archives.
After this operation, 26.6 kB of additional disk space will be used.
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libapparmor1 amd64 2.13.3-7ubuntu5.2 [35.4 kB]
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 apparmor amd64 2.13.3-7ubuntu5.2 [502 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 bolt amd64 0.9.1-2-ubuntu20.04.2 [145 kB]
Fetched 680 kB in 0s (15.9 kB/s)
Preconfiguring packages...
(Reading database ... 61914 files and directories currently installed.)
Preparing to unpack .../libapparmor1_2.13.3-7ubuntu5.2_amd64.deb ...
Unpacking libapparmor1:amd64 (2.13.3-7ubuntu5.2) over (2.13.3-7ubuntu5.1) ...
Preparing to unpack .../apparmor_2.13.3-7ubuntu5.2_amd64.deb ...
Unpacking apparmor (2.13.3-7ubuntu5.2) over (2.13.3-7ubuntu5.1) ...
Preparing to unpack .../bolt_0.9.1-2-ubuntu20.04.2_amd64.deb ...
Unpacking bolt (0.9.1-2-ubuntu20.04.2) over (0.9.1-2-ubuntu20.04.1) ...
Setting up libapparmor1:amd64 (2.13.3-7ubuntu5.2) ...
Setting up apparmor (2.13.3-7ubuntu5.2) ...
Installing new version of config file /etc/apparmor/parser.conf ...
Replacing old version of config file /etc/apparmor/parser.conf
Skipping profile in /etc/apparmor.d/disable: user.sbin.rsyslogd
Setting up bolt (0.9.1-2-ubuntu20.04.2) ...
bolt.service is a disabled or a static unit not running, not starting it.
Processing triggers for systemd (245.4-4ubuntu3.20) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for libc-bin (2.31-1ubuntu9.9) ...
ubuntu@ip-172-31-29-204:~$
```

Copy the jenkins key cmnd and jenkins key add to repo cmnd from website.Paste in Master server.

```
Jenkins Debian Packages

Important Notice: Beginning with LTS 2.387.2 and weekly 2.397, releases will be signed with a new GPG key.
Administrators must install the new key on their servers before attempting to update Jenkins.
Read more about the key rotation on the blog.

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system.

curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.gpg > /dev/null

Then add a Jenkins apt repository entry.

echo deb [signed-by=jenkins-keyring] https://pkg.jenkins.io/debian binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

Update your local package index, then finally install Jenkins.

sudo apt-get update
sudo apt-get install fontconfig openjpa-1.1-jre
sudo apt-get install jenkins

The apt packages were signed using this key.

pub   rsa4096 2023-03-27 [SC] [expires: 2030-03-26]
      4AECB0E8A3D93E80A83A7E7097C8
      Jenkins Project
uid   RSA4096 2023-03-27 [S] [expires: 2030-03-26]
      Read more about the key rotation on the blog.
```

```
Jenkins Debian Packages

Important Notice: Beginning with LTS 2.387.2 and weekly 2.397, releases will be signed with a new GPG key.
Administrators must install the new key on their servers before attempting to update Jenkins.
Read more about the key rotation on the blog.

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system.

curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.gpg > /dev/null

Then add a Jenkins apt repository entry.

echo deb [signed-by=jenkins-keyring] https://pkg.jenkins.io/debian binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
aws Services Search [Alt+S]
ubuntu@ip-172-31-29-204:~$ curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
> /usr/share/keyrings/jenkins-keyring.gpg > /dev/null
ubuntu@ip-172-31-29-204:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.gpg] \
> https://pkg.jenkins.io/debian binary/ | sudo tee \
> /etc/apt/sources.list.d/jenkins.list > /dev/null
ubuntu@ip-172-31-29-204:~$ sudo apt-get update
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Ign:4 https://pkg.jenkins.io/debian binary/ InRelease
Get:5 https://pkg.jenkins.io/debian binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:8 https://esm.ubuntu.com/apps/ubuntu focal-apps-security InRelease
Hit:9 https://esm.ubuntu.com/apps/ubuntu focal-apps-updates InRelease
Get:10 https://esm.ubuntu.com/infra/ubuntu focal-infra-security InRelease [7453 B]
Get:11 https://esm.ubuntu.com/infra/ubuntu focal-infra-updates InRelease [7452 B]
Get:12 https://pkg.jenkins.io/debian binary/ Packages [52.5 kB]
Fetched 70.3 kB in 2s (42.2 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-29-204:~$
```

**After successfully completed the above steps.,Update the repo packages.**

```
$sudo apt-get update -y
```

## Then install java package

### **Lastly Install the Jenkins package**

```
$sudo apt-get install fontconfig openjdk-11-jre
```

```
$sudo apt-get install jenkins
```

```
aws Services [Search] [Alt+5]
ubuntu@ip-172-31-29-204:~$ sudo apt-get install jenkins
Reading package lists... done
Building dependency tree
Reading state information... done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 97.9 MB of archives.
After this operation, 99.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

**Ensure Both Java and Jenkins packages are installed properly by checking versions and ports**

```
$java --version
```

```
$ps -ef | grep jenkins
```

```
root@ip-172-31-29-2041:~# java -version
openjdk 11.0.18 2023-01-17
OpenJDK Java Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu120.64.1)
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu120.64.1, mixed mode, sharing)
root@ip-172-31-29-2041:~# ps -ef | grep jenkins
jenkins   7928      1  12 2017 ?    00:01:14 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
jenkins   8469     6247  0 07:37 pts/0    00:00:00 grep --color=auto jenkins
root@ip-172-31-29-2041:~#
```

~\$ cd .ssh/

### Create two touch files

\$touch id\_rsa.pub

\$touch id\_rsa

and copy authorized key public key files in id\_rsa.pub file.

```
aws Services Search [Alt+S]
ubuntu@ip-172-31-29-204:~$ ll
total 28
drwxr-xr-x 4 ubuntu ubuntu 4096 Apr  2 07:23 /
drwxr-xr-x 3 root  root  4096 Apr  2 07:19 /
-rw-r--r-- 1 ubuntu ubuntu 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Feb 25 2020 .bashrc
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr  2 07:22 .cassandra/
-rw-r--r-- 1 ubuntu ubuntu 807 Feb 25 2020 .profile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr  2 07:19 .ssh/
-rw-r--r-- 1 ubuntu ubuntu 0 Apr  2 07:23 .sudo_as_admin_successful
ubuntu@ip-172-31-29-204:~$ cd .ssh/
ubuntu@ip-172-31-29-204:~/ssh$ ll
total 12
drwx----- 2 ubuntu ubuntu 4096 Apr  2 07:19 .
drwxr-xr-x 4 ubuntu ubuntu 4096 Apr  2 07:23 ..
-rw----- 1 ubuntu ubuntu 389 Apr  2 07:19 authorized_keys
ubuntu@ip-172-31-29-204:~/ssh$ touch id_rsa
ubuntu@ip-172-31-29-204:~/ssh$ touch id_rsa.pub
ubuntu@ip-172-31-29-204:~/ssh$ cp authorized_keys id_rsa.pub
ubuntu@ip-172-31-29-204:~/ssh$
```

\$vi id\_rsa

Paste pem key file content

:wg!

Schmod 400 \*

```
ubuntu@ip-172-31-28-204:~/ssh> ssh user id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIDAEQCAQw17AA...nR03ENZEpxRwXm3zvQERGjKRXeRu1HJM
RptZny2o[...]/ORR2E3...23ACfFED232NxKtPZ2q5mWXXQppM4cve61u...n1HKsA
E8C17Tzv...p...Sx5-K5...OzRb1Kc5...egNrc...XvCa...Kt...g...Tz...Sx13K
2uAuM...M75Gig...Hq...Qpu...90L...T...n...g...b...Tm...12...3...B...7...9...v...v...f...8...0
TCVXCMOf...RNU...J...u...g...h...M...7...T...2...l...u...m...M...y...b...v...f...1...P...d...1...q...y...B...R...S...N...I
-----END RSA PRIVATE KEY-----
```

\$ ssh localhost

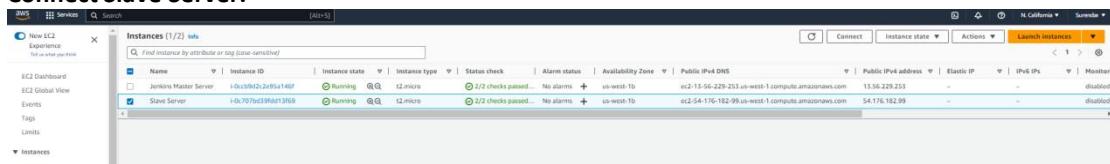
Yes

SII

**Ensure known hosts file is created.**

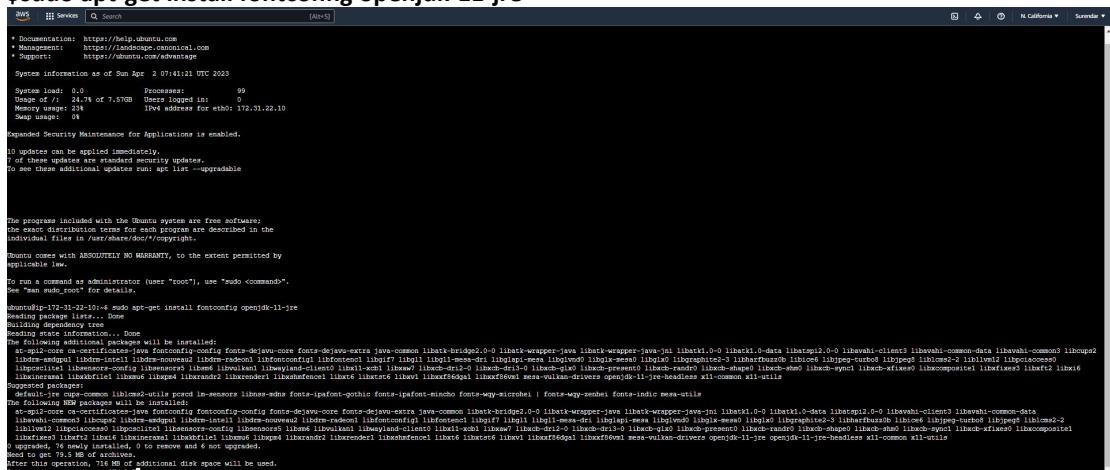
```
ubuntu@ip-172-31-29-204:~/ssh$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:ApkQGQay2vGh2ZnC6U7fCEpYDpxU1yxPcaIxG5dcPA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
@@@@@@@@@@@aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
@           WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Permissions 0664 for '/home/ubuntu/.ssh/id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/ubuntu/.ssh/id_rsa": bad permissions
ubuntu@localhost: Permission denied (publickey).
ubuntu@ip-172-31-29-204:~/ssh$ ll
total 24
drwx-----  2 ubuntu  ubuntu  4096 Apr  2 07:50  .
drwxr-xr-x  4 ubuntu  ubuntu  4096 Apr  2 07:39  ..
-r-----  1 ubuntu  ubuntu   389 Apr  2 07:19 authorized_keys
-r-----  1 ubuntu  ubuntu  1675 Apr  2 07:39 id_rsa
-r-----  1 ubuntu  ubuntu   389 Apr  2 07:38 id_rsa.pub
-rw-r--r--  1 ubuntu  ubuntu   222 Apr  2 07:50 known_hosts
ubuntu@ip-172-31-29-204:~/ssh$
```

### **Connect Slave Server.**



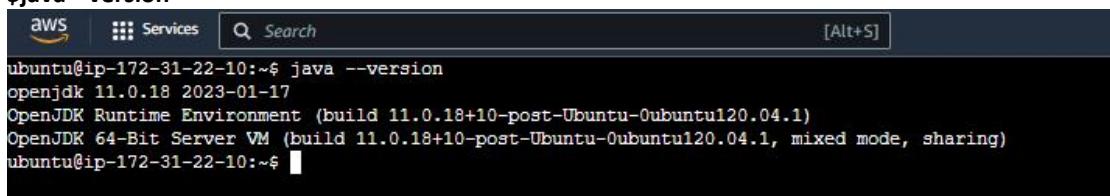
## Install Java Packages in Slave server

```
$sudo apt-get install fontconfig openjdk-11-jre
```



**Ensure Java Packages installed correctly by verifying java version**

```
$java --version
```



```

$mkdir jenkins
$cd jenkins/
$pwd
/home/ubuntu/jenkins <-- copy and paste in Remote root directory file while creating node.
ubuntu@ip-172-31-22-10:~$ mkdir jenkins
ubuntu@ip-172-31-22-10:~$ ll
total 32
drwxr-xr-x 5 ubuntu ubuntu 4096 Apr  2 07:44 .
drwxr-xr-x 3 root  root  4096 Apr  2 07:19 ../
-rw-r--r-- 1 ubuntu ubuntu 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Feb 25 2020 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Apr  2 07:41 .cache/
-rw-r--r-- 1 ubuntu ubuntu 807 Feb 25 2020 .profile
drwx----- 2 ubuntu ubuntu 4096 Apr  2 07:19 .ssh/
-rw-r--r-- 1 ubuntu ubuntu    0 Apr  2 07:41 .sudo_as_admin_successful
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr  2 07:44 jenkins/
ubuntu@ip-172-31-22-10:~$ cd jenkins/
ubuntu@ip-172-31-22-10:~/jenkins$ ll
total 8
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr  2 07:44 .
drwxr-xr-x 5 ubuntu ubuntu 4096 Apr  2 07:44 ../
ubuntu@ip-172-31-22-10:~/jenkins$ pwd
/home/ubuntu/jenkins
ubuntu@ip-172-31-22-10:~/jenkins$ 

```

**Copy master server IPv4 add - Paste in chrome tab in the end mention jenkins portno:8080**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Elastic IP	IPv6 IPs	Month
Jenkins Master Server	i-0cc9d2c2e95a146f	Running	t2.micro	2/2 checks passed	No alarms	us-west-1b	ec2-13-56-229-253.us-west-1.compute.amazonaws.com	13.56.229.253	-	-	disabled
Slave Server	i-0707bd319fb139cb	Running	t2.micro	2/2 checks passed	No alarms	us-west-1b	ec2-54-176-182-99.us-west-1.compute.amazonaws.com	54.176.182.99	-	-	disabled

**After successfully logged in the jenkins dashboard webserver.**

**Dashboard - Manage Jenkins - Set up agent.**

**Create a node to sync the Master server and Slave server.**

**Set a name (Node1) - Select Permanent agent - Create.**

**Paste Slave server root directory path**

**Set a label name (Slave 1)**

**In Launch method Choose ( launch agents via SSH)**

**In Host ( Paste Slave Server IPv4 add)**

**In credentials - Add - Choose jenkins**

The screenshot shows the Jenkins 'Nodes' configuration page. A new node is being created with the following details:

- Usage:** Use this node as much as possible
- Launch method:** Launch agents via SSH
- Host:** 54.176.182.99
- Credentials:** None
- Jenkins Credentials Provider:** Jenkins (selected)

**In Kind - Choose SSH username with private key**

**In Username - Enter Ubuntu**

**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

(empty field)

Description

(empty field)

Username

ubuntu

**Private key - choose Enter directly - Paste pemkey file content - Add.**

Treat username as secret

Private Key

Enter directly

Key

Enter New Secret Below  
-----  
-----END RSA PRIVATE KEY-----

Passphrase

(empty field)

**Add** **Cancel**

**Choose Ubuntu in Credentials - In host key verification strategy - Choose Manually trusted key verification strategy - save.**

The screenshot shows the Jenkins 'Credentials' configuration page. A new credential is being created with the following details:

- Credentials:** ubuntu
- Host Key Verification Strategy:** Manually trusted key Verification Strategy
- Require manual verification of initial connection:** Unchecked
- Availability:** Keep this agent online as much as possible

**Wait till the Node1 comes to (in sync) state with master server. Then create a new job in slave server.**

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.72 GB	0 B	4.72 GB	0ms
	Node1	Linux (amd64)	In sync	5.24 GB	0 B	5.24 GB	14ms
		Data obtained	0.28 sec	0.28 sec	0.28 sec	0.28 sec	0.28 sec

**In Dashboard - New items - Enter Item name (project1) choose Freestyle project - ok.**

**In Configure - general - Choose Restrict where this project can be run - In label Expression Choose Slave1 label.**

**In build Steps - Create a Execute Shell - Save.**

**Now Ensure the Created Project is properly executed in Node1 slave server.**

**Dashboard - Manage jenkins - Nodes and Cloud.**

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is selected), and 'My Views'. The main content area is titled 'Manage Jenkins' and has a sub-section 'System Configuration'. It includes tabs for 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), and 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins). A note at the top says: 'Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See [the documentation](#)'. There are 'Manage' and 'Dismiss' buttons at the top right of the 'Plugins' section. A sidebar on the right is titled 'Nodes and Clouds' with the sub-instruction: 'Add, remove, control and monitor the various nodes that Jenkins runs jobs on.'

**Choose Node1 - Status - select Slave1 - There can able to view created project1 job is restricted to executed in Node1.**

The screenshot shows the Jenkins slave1 node status page. The left sidebar has links for 'Overview', 'Configure', and 'Load Statistics'. The main content area is titled 'slave1' and shows a table of build statuses. The table has columns: S (Status), W (Working), Name, Last Success, Last Failure, and Last Duration. One row is visible: 'Project1' with a green circle icon, '1 min 14 sec' under 'Last Success', 'N/A' under 'Last Failure', and '88 ms' under 'Last Duration'. At the bottom, there are icons for 'Icon: S M L' and links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

S	W	Name	Last Success	Last Failure	Last Duration
		Project1	1 min 14 sec	N/A	88 ms

**Successfully configured Master / Slave server. Managed both master and slave server (In sync) with help of creating Node in jenkins dashboard webserver. And also Deployed automated execute shell script job done in Slave server.**

## STEPS TO HOST AN JENKINS DASHBOARD WEB SERVER IN OUR OWN DOMAIN NAME

### In Route53 - Create hosted zone

Domain name: `harhasurey.today`

Description: (optional)

Type:  Public hosted zone

Tags: None

**Create hosted zone**

### Copy name servers for created hosted zones domain name

Name servers:

- ns-1683.awsdns-18.co.uk
- ns-130.awsdns-16.com
- ns-1360.awsdns-42.org
- ns-748.awsdns-29.net

### Paste in third-party domain application name-servers field - save.

Edit nameservers

Enter My Own Nameservers

ns-1683.awsdns-18.co.uk

ns-130.awsdns-16.com

ns-1360.awsdns-42.org

ns-748.awsdns-29.net

Add Nameserver

Cancel Back Save

### In certificate Manager Create certificate to host jenkins dashboard server in securely Request certificate - Enter qualified domain name as (jenkins.harhasurey.today) - request.

Domain names: `jenkins.harhasurey.today`

Validation method:  DNS validation - recommended

Key algorithm: RSA-2048

Tags: None

**Request**

## In certificate name - Create DNS records in Amazon Route53

The screenshot shows the AWS Certificate Manager interface. A search bar at the top right contains the query "Search domain". Below it, a filter bar has "Validation status: Pending validation" and "Validation status: Failed" selected. A dropdown menu "Is domain in Route 53? Yes" is open. A table lists a single record: "jenkins.hansl... today" with type "CNAME" and value "f5e2340a3b...0244...". The status is "Pending validation". A "Create records" button is visible at the bottom right.

## Now Ensure cname is created in hostedzone.

The screenshot shows the AWS Route 53 Hosted Zone Details page for the zone "harshasurey.today". It displays basic information like the zone name, ID, and record count. Under the "Records (4) Info" section, a table lists four records: "harshasurey.today" (NS), "harshasurey.today" (SOA), and two CNAME records pointing to Jenkins instances. A "Create record" button is located at the top right of the records table.

## In loadbalancer

### Create a Classic Load balancer.

The screenshot shows the "Create a Classic Load Balancer" wizard. The first step, "Set a name - Choose Same security group which already created with master server - next", is completed. The second step, "Step 2: Assign Security Groups", is active. It shows a list of security groups available for selection. A checkbox for "Select an existing security group" is checked, and a dropdown menu "Filter [VPC security groups]" is open. Several security groups are listed, including "default", "jenkins", and "Splunk".

## Set a name - Choose Same security group which already created with master server - next

The screenshot shows the "Step 2: Assign Security Groups" screen of the wizard. It lists several security groups with their names and descriptions. A "Filter [VPC security groups]" dropdown is at the top right. At the bottom, there are "Cancel", "Previous", and "Next: Configure Security Settings" buttons.

Security Group ID	Name	Description	Actions
sg-020352e2a2399e89	default	default VPC security group	<a href="#">Copy to new</a>
sg-014a48915071c9408	ec2-rts-1	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-03055e4fbefc405	ec2-rts-10	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-07a7c750129b9392	ec2-rts-11	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-07375151949e6066	ec2-rts-12	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-049049c9c2556268	ec2-rts-13	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-0f0a2550e4131819	ec2-rts-2	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-04047ed7d5776155	ec2-rts-3	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-040520be9301d19	ec2-rts-4	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-030475008eb3e3603	ec2-rts-5	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-0770de37b67f9e9ee	ec2-rts-6	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-09270e9b082c240	ec2-rts-7	Security group attached to instances to securely connect to ProdDB. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-04057b67e5262af	Jenkins	launched-2 created 2023-03-31T05:36:17.03Z	<a href="#">Copy to new</a>
sg-04083e3e40e1515	ec2-rts-8	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-040332000000000004	ec2-rts-9	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-040999e95116a0	ec2-rts-10	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-0408300f30365544	ec2-rts-11	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-041212a3e83276d0	ec2-rts-12	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-0570e179c7c331c	ec2-rts-13	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-040709b9a2693a3	ec2-rts-14	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-055530b917319711d	ec2-rts-15	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-0489c737a20e1125	ec2-rts-16	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-04927ad07fb7bf473	ec2-rts-17	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-02706a8308d61913	ec2-rts-18	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-07709312424123f	ec2-rts-19	Security group attached to ProdDB to allow EC2 instances with specific security groups attached to connect to the database. Modification could lead to connection loss.	<a href="#">Copy to new</a>
sg-04140aa3a97b646079	Splunk	launched-1 created 2023-03-29T03:56:21.03Z	<a href="#">Copy to new</a>

## In health check set default - next

Step 4: Configure Health Check  
Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Configure the health check to meet your specific needs.

**Ping Port**  **Ping Path**

**Advanced Details**

Response Timeout	<input type="text" value="5"/>	seconds
Interval	<input type="text" value="10"/>	seconds
Unhealthy threshold	<input type="text" value="2"/>	
Healthy threshold	<input type="text" value="10"/>	

## In add EC2 instances - Select jenkins Master EC2 server - create CLB

Step 5: Add EC2 Instances  
The table below lists all your running EC2 instances. Check the boxes in the Selected column to add those instances to this load balancer.

VPC: vpc-09873fb5bd99ed0d (172.31.0.0/16)

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/> ip-10.0.0.12.ec2.internal	Jenkins Master server	Running	Jenkins	us-west-1b	subnet-009873fb5bd99ed0d	172.31.16.0/20

Availability Zone Distribution

1 instance in us-west-1b

- Enable Cross-Zone Load Balancing (i)
- Enable Connection Draining (i)  seconds

## Click Load balancer name - Wait till Instances status changes to (In-Service) -

New EC2 Experience  Tell me what you think

EC2 Dashboard  EC2 Global View  Events  Tags  Limits  Instances  Instance Types  Launch Templates  Spot Requests  Savings Plans

EC2 | Load balancers | JenkinsCL

Load balancer: JenkinsCL

Description Instances Health check Listeners Monitoring Tags Migration

Connection draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
ip-10.0.0.12.ec2.internal	Jenkins Master server	us-west-1b	In-Service (i)	Remove from Load Balancer

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-west-1a	subnet-009873fb5bd99ed0d	172.31.0.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-west-1b	subnet-0098412420454243	172.31.16.0/20	1	Yes	Remove from Load Balancer

## In Listener - edit - Set Loadbalancer Protocol (HTTPS) - load balancer port (8080) - Set Instance Protocol (HTTP) - Instance Port (8080) - Attach SSL Certificate - Save.

Edit listeners

The following listeners are currently configured for this load balancer:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Cipher	SSL Certificate
HTTPS (Secure HTTP) <input type="button" value="▼"/>	8080	HTTP <input type="button" value="▼"/>	8080	Change	5e4ae15f-32cd-4ef8-86aa-358886ceb84e (ACM) <input type="button" value="Change"/> Remove

Add

## In Route53 - Create Record - Set record name (Jenkins) - Enable Alias - Select Alias to application and Classic load Balancer - Select loadbalancer region - choose loadbalancer (jenkinsCL) - Create record.

Create record

Quick create record

**Record 1**

Record name:  .harshasurey.today

Record type:  A – Routes traffic to an IPv4 address and some AWS resources

Keep blank to create a record for the root domain.

**Alias**

Route traffic to:

Alias to Application and Classic Load Balancer

US West (N. California) [us-west-1]

Q dualstack.JenkinsCL-1375157701.us-west-1.elb.amazonaws.com

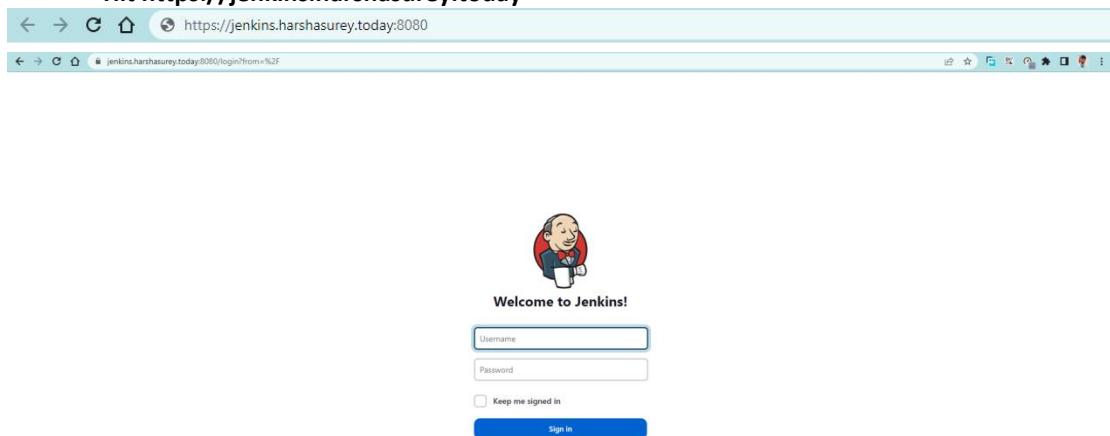
Alias hosted zone ID: Z3608ELL9RE2K40

Routing policy:  Simple routing  Evaluate target health  Yes

**View existing records**  
The following table lists the existing records in harshasurey.today.

In chrome tab

Hit <https://jenkins.harshasurey.today>



Successfully hosted Jenkins Dashboard in a **jenkins.harshasurey.today** domain name.

## AUTODEPLOYMENT

### STEPS TO DEPLOY A WAR FILE TO TOMCAT APPLICATION USING JENKINS

#### In Ubuntu 20.04 EC2 Server

After successfully installed java and Jenkins packages

```
ubuntu@ip-172-31-31-239:~$ java --version
openjdk 11.0.18 2023-01-17
OpenJDK Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1)
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1, mixed mode, sharing)
ubuntu@ip-172-31-31-239:~$ ps -ef | grep jenkins
jenkins 12 43 0 08:56 ? 00:00:41 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
ubuntu 13093 1338 0 08:57 pts/0 00:00:00 grep --color=auto jenkins
ubuntu@ip-172-31-31-239:~$
```

#### In chrome

Browse Tomcat download - Tomcat.apache.org - choose any version and copy tar.gz URL.

You are currently using <https://dlcdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors: <https://dlcdn.apache.org/> Change

**9.0.74**

Please see the [README](#) file for packaging information. It explains what every distribution contains.

**Binary Distributions**

- Core:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
  - [32-bit Windows zip \(pgp, sha512\)](#)
  - [64-bit Windows zip \(pgp, sha512\)](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)

#### In Jenkins EC2 Server

```
#cd /opt
# wget <paste Tomcat .tar.gz URL>
```

```
root@ip-172-31-31-239:/opt#
root@ip-172-31-31-239:~# cd /opt
root@ip-172-31-31-239:/opt# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.74/bin/apache-tomcat-9.0.74.tar.gz
```

After successfully downloaded the tomcat .tar file.Untar the file using

```
#tar -xvzf <tomcat .tar file>
```

```
root@ip-172-31-31-239:/opt# ls
apache-tomcat-9.0.74.tar.gz
root@ip-172-31-31-239:/opt# tar -xvzf apache-tomcat-9.0.74.tar.gz
```

After untar the tomcat file.Directory will be created.Get in to the directory.

```
#cd <dir name>
```

```
root@ip-172-31-31-239:/opt# ls
apache-tomcat-9.0.74 apache-tomcat-9.0.74.tar.gz
root@ip-172-31-31-239:/opt# cd apache-tomcat-9.0.74
```

#### In Apache Tomcat file

Change the Tomcat server Connector default port.

```
#cd conf/
```

```
#vi server.xml
```

```
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# ls
BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# cd conf/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/conf# ls
catalina.policy context.xml jaspic-providers.xsd server.xml tomcat-users.xsd
catalina.properties jaspic-providers.xml logging.properties tomcat-users.xml web.xml
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/conf# vi server.xml
```

## In server.xml file

Change the connector port="8080" --> ="9090"

```
<!-- A "Connector" represents an endpoint by which requests are received  
and responses are returned. Documentation at :  
Java HTTP Connector: /docs/config/http.html  
Java AJP Connector: /docs/config/ajp.html  
APR (HTTP/AJP) Connector: /docs/apr.html  
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080  
-->  
<Connector port="9090" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443"  
maxParameterCount="1000"  
/>  
<!-- A "Connector" using the shared thread pool-->  
<!--  
<Connector executor="tomcatThreadPool"  
port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443"  
maxParameterCount="1000"  
/>>
```

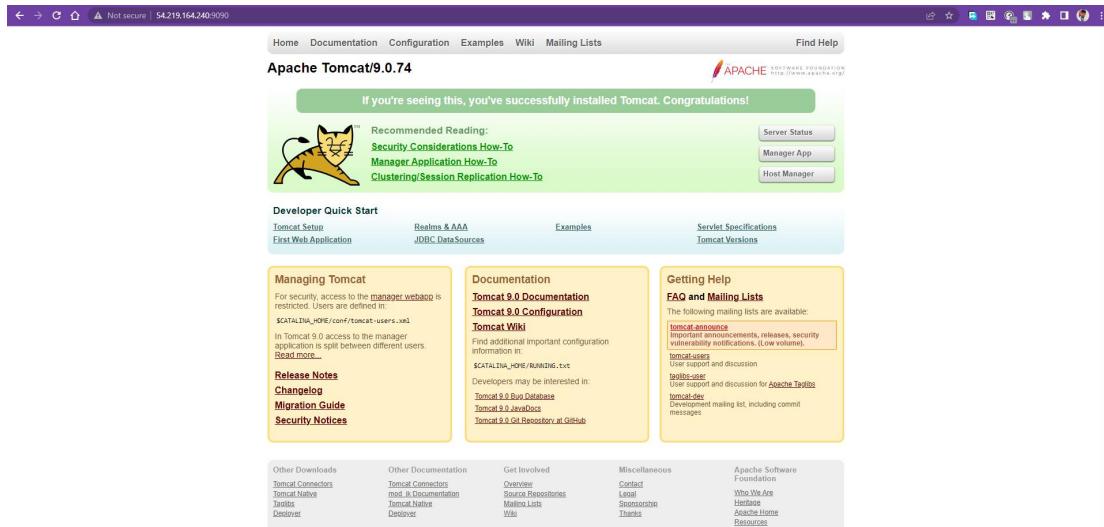
Then start the tomcat service

```
#cd ..  
#cd bin/  
#./startup.sh
```

```
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# ls  
BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work  
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# cd bin  
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# ls  
bootstrap.jar ciphers.bat configtest.bat digest.sh setclasspath.sh startup.sh tool-wrapper.sh  
catalina-tasks.xml configtest.bat makebase.bat shutdown.bat tomcat-juli.jar version.bat  
catalina.bat commands-daemon-native.tar.gz daemon.sh makebase.sh shutdown.sh tomcat-native.tar.gz version.sh  
catalina.sh commands-daemon.jar digest.bat setclasspath.bat startup.bat tool-wrapper.bat  
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# ./startup.sh  
Using CATALINA_BASE: /opt/apache-tomcat-9.0.74  
Using CATALINA_HOME: /opt/apache-tomcat-9.0.74  
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.74/temp  
Using JRE_HOME: /usr  
Using CLASSPATH: /opt/apache-tomcat-9.0.74/bin/bootstrap.jar:/opt/apache-tomcat-9.0.74/bin/tomcat-juli.jar  
Using CATALINA_OPTS:  
Tomcat started.  
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin#
```

Now hit the Tomcat Test Page in browser.

<Jenkins Server IP>:9090



Tomcat Test page is working fine

## In Jenkins Dashboard

Create a new freestyle project job in name of Autodeployment.

The screenshot shows the Jenkins 'Enter an item name' dialog. A text input field contains the text 'autodeployment'. Below the input field is a list of project types with descriptions:

- 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.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**: Creates a set of multibranch project subfolders by scanning for repositories.

A blue 'OK' button is at the bottom right of the dialog.

## In Build Steps

Execute a shell cmnd

Ex: pwd

Apply & Save.

The screenshot shows the Jenkins 'Configuration' screen for the 'autodeployment' job. Under the 'Build Steps' section, there is a 'Command' input field containing 'pwd'. The 'Advanced' dropdown is open. At the bottom of the screen, there are 'Save' and 'Apply' buttons, with 'Saved' indicated next to the 'Save' button.

Build the Job Now - Ensure Job successfully deployed.

The screenshot shows the Jenkins Dashboard. In the center, there is a table listing the 'autodeployment' job. The job has a green status icon, indicating it is successful. The table columns are: S, W, Name, Last Success, Last Failure, and Last Duration. The 'autodeployment' row shows values: S, W, autodeployment, 6.5 sec, N/A, 38 ms. Below the table, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 idle).

## In Jenkins Server

go to jenkins workspace and ensure the job dir is created and ensure nothing in dir.

```
#cd /var/lib/jenkins/workspace
```

```
#cd autodeployment/
```

```
#ls
```

```
root@ip-172-31-31-239:~# cd /var/lib/jenkins/workspace
root@ip-172-31-31-239:/var/lib/jenkins/workspace# ls
autodeployment
root@ip-172-31-31-239:/var/lib/jenkins/workspace# cd autodeployment/
root@ip-172-31-31-239:/var/lib/jenkins/workspace/autodeployment# ls
root@ip-172-31-31-239:/var/lib/jenkins/workspace/autodeployment#
```

## In Chrome

Browse Tomcat war file to download  
in Tomcat official link - Copy the download URL of the sample Tomcat war file.



### Sample Application

The example app has been packaged as a war file and can be downloaded [here](#) (Note: make sure your browser doesn't change file extension or append a new one).

The easiest way to run this application is simply to move the war file to your **CATALINA\_HOME/webapps** directory. Tomcat will automatically expand and deploy the app. You can view it with the following URL (assuming that you're running tomcat on port 8080 as is the default):  
<http://localhost:8080/sample>

If you just want to browse the contents, you can unpack the war file with the **jar** command.

```
jar -xvf sample.war
```

### Paste the war file URL link in autodeployment dir.

```
#wget <Tomcat war file URL>
```

```
#ls
```

```
root@ip-172-31-239:~/var/lib/jenkins/workspace/autodeployment# wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
--2023-04-26 09:15:45-- https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
Resolving tomcat.apache.org (tomcat.apache.org)... 151.101.2.132, 2a04:4e42:644
Connecting to tomcat.apache.org (tomcat.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4606 (4.5K)
Saving to: 'sample.war'

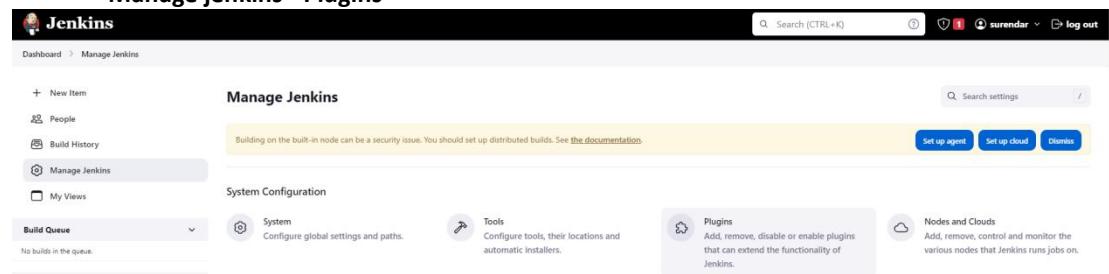
sample.war          100%[=====] 4.50K --.-KB/s   in 0s

2023-04-26 09:15:45 (46.0 MB/s) - 'sample.war' saved [4606/4606]
root@ip-172-31-239:~/var/lib/jenkins/workspace/autodeployment# ls
sample.war
root@ip-172-31-239:~/var/lib/jenkins/workspace/autodeployment#
```

### In Jenkins Dashboard

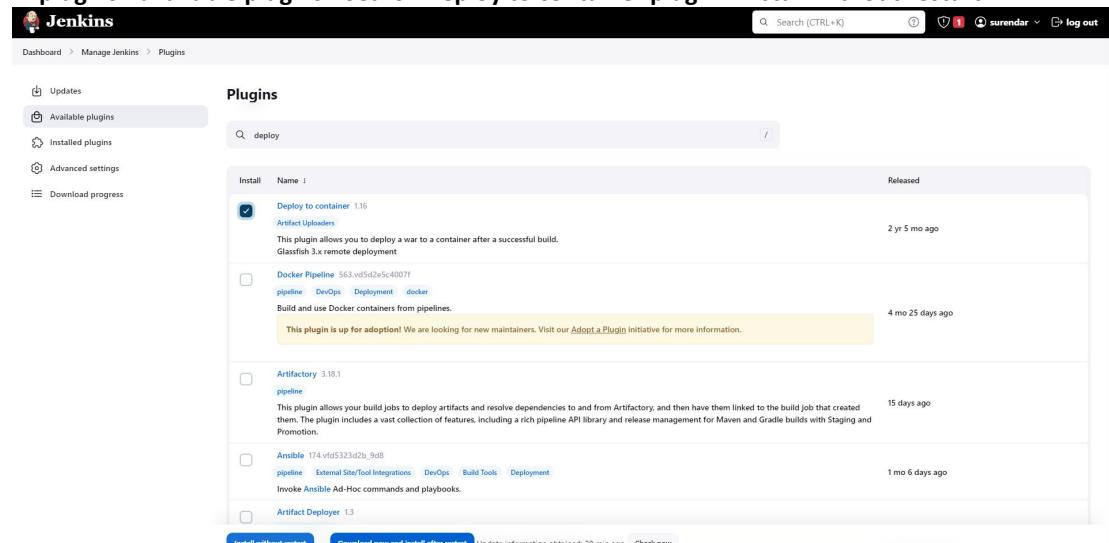
Download the deploy to the container plugins in jenkins dashboard to attach the war file in the freestyle project job.

#### Manage jenkins - Plugins



The screenshot shows the Jenkins Manage Jenkins dashboard under the Plugins section. It includes links for Set up agent, Set up cloud, and Dismiss. Below the links, there are sections for System Configuration (System, Tools), Plugins (Add, remove, disable or enable plugins), and Nodes and Clouds (Add, remove, control and monitor the various nodes that Jenkins runs jobs on).

### In plugins - available plugins - Search Deploy to container plugin - Install without restart.



The screenshot shows the Jenkins Plugins page with the search term "deploy" entered. The Deploy to container plugin is selected for installation. Other listed plugins include Docker Pipeline, Artifactory, Ansible, and Artifact Deployer. At the bottom, there are buttons for "Install without restart" and "Download now and install after restart".

After successfully installed the plugins.

**Go to the autodeployment job which is already created - Configure.**

**In Post -Build-Action**

**Select deploy war/ear to a container option**

The screenshot shows the Jenkins job configuration interface for an 'autodeployment' job. In the 'Post-build Actions' section, a dropdown menu is open under the 'Deploy war/ear to a container' option. The menu includes various Jenkins actions like 'Aggregate downstream test results', 'Archive the artifacts', and 'Deploy war/ear to a container'. The 'Deploy war/ear to a container' option is highlighted. At the bottom of the menu, there is a link 'Add post-build action >'. Below the menu, there are 'Save' and 'Apply' buttons.

**In WAR/EAR files**

**Enter (Sample.war)**

**In context path**

**Enter /sample**

**Containers**

**Choose Tomcat version**

**Ex: Tomcat 9.x Remote**

The screenshot shows the Jenkins job configuration interface for an 'autodeployment' job. In the 'Post-build Actions' section, the 'WAR/EAR files' field contains 'sample.war' and the 'Context path' field contains '/sample'. In the 'Containers' section, a dropdown menu is open, showing a list of available containers including GlassFish 4.x, JBoss AS 3.x, JBoss AS 4.x, JBoss AS 5.x, JBoss AS 6.x, JBoss AS 7.x, Tomcat 4.x Remote, Tomcat 5.x Remote, Tomcat 6.x Remote, Tomcat 7.x Remote, Tomcat 8.x Remote, and Tomcat 9.x Remote. The 'Tomcat 9.x Remote' option is highlighted. At the bottom right of the screen, there are links for 'REST API' and 'Jenkins 2.402'.

**Then Add - Jenkins.**

**Containers**

The screenshot shows the Jenkins job configuration interface for an 'autodeployment' job. In the 'Credentials' section, a dropdown menu is open, showing options like '- none -' and 'Add'. A button labeled 'Jenkins' is highlighted with a blue box. At the bottom right of the screen, there are links for 'REST API' and 'Jenkins 2.402'.

### In credentials

Choose kind

Username with password

Ex: Username: admin

Password: admin

ID: admin

Add - Choose the added credentials.

Kind

Username with password

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

Username ?  
admin

Treat username as secret ?

Password ?  
.....

ID ?  
admin

Description ?

**Add** **Cancel**



### In Tomcat URL

Enter the Tomcat testpage URL

Apply and Save.

Containers

Credentials

admin/\*\*\*\*\*\*

**Add**

Tomcat URL ?

http://54.219.164.240:9090

Advanced

Add Container

Deploy on failure

Add post-build action ▾

**Save** **Apply**



### In apache.tomcat file

Enter the user details already created the credentials in tomcat-users.xml file.

#cd /opt/apache-tomcat-9.0.74/conf/

#vi tomcat-users.xml

```
root@ip-172-31-31-239:/opt# ls
apache-tomcat-9.0.74  apache-tomcat-9.0.74.tar.gz
root@ip-172-31-31-239:/opt# cd apache-tomcat-9.0.74/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# cd conf/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/conf# ls
Catalina      catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
catalina.policy  context.xml       jaspic-providers.xsd  server.xml        tomcat-users.xsd
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/conf# vi tomcat-users.xml
```

**Enter the xml user content under the Tomcat-users as shown below.**

```
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
               version="1.0">
    <role rolename="manager-gui"/>
    <role rolename="manager-script"/>
    <role rolename="manager-jmx"/>
    <role rolename="manager-status"/>
    <user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
    <user username="deployer" password="deployer" roles="manager-script"/>
<!--
By default, no user is included in the "manager-gui" role required
to operate the "/manager/html" web application. If you wish to use this app,
you must define such a user - the username and password are arbitrary.
-->
```

**Then go the bin dir and restart the tomcat service.**

```
#cd ..  
#cd bin/  
#./shutdown.sh  
#./startup.sh
```

```
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/conf# cd ..
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# cd bin/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# ./shutdown.sh
Using CATALINA_BASE: /opt/apache-tomcat-9.0.74
Using CATALINA_HOME: /opt/apache-tomcat-9.0.74
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.74/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/apache-tomcat-9.0.74/bin/bootstrap.jar:/opt/apache-tomcat-9.0.74/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# ./startup.sh
Using CATALINA_BASE: /opt/apache-tomcat-9.0.74
Using CATALINA_HOME: /opt/apache-tomcat-9.0.74
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.74/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/apache-tomcat-9.0.74/bin/bootstrap.jar:/opt/apache-tomcat-9.0.74/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin#
```

**Now run the job in Jenkins dashboard.**

The Jenkins dashboard displays a single job entry for 'autodeployment'. The job status shows 'Last Success' at 22 min and 'Last Failure' at 17 sec. The 'Build History' section indicates a duration of 38 ms. Navigation links include 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. A search bar at the top right is populated with 'surendar'. A 'Build Queue' section shows 'No builds in the queue.' and a 'Build Executor Status' section shows '1 idle'.

S	W	Name	Last Success	Last Failure	Last Duration
		autodeployment	22 min <a href="#">#2</a>	17 sec <a href="#">#5</a>	38 ms

Icon: S M L

Icon legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

Build Queue: No builds in the queue.

Build Executor Status: 1 idle

#### **Autodeployment job gets unsuccessful.**

## **Go the job console output - Find the Error**

**Copy the Error line and browse in Chrome to find the solution**

**Copy the Error line and browse in Chrome to find the solution.**

Caused by: org.codehaus.cargo.container.tomcat.internal.TomcatManagerException: The username you provided is not allowed to use the text-based Tomcat Manager (error 403)

at org.codehaus.cargo.container.tomcat.internal.TomcatManager.invoke(TomcatManager.java:710)  
at org.codehaus.cargo.container.tomcat.internal.TomcatManager.list(TomcatManager.java:882)  
at org.codehaus.cargo.container.tomcat.internal.TomcatManager.getStatus(TomcatManager.java:895)  
at org.codehaus.cargo.container.tomcat.internal.TomcatManagerDeployer.deploy(TomcatManagerDeployer.java:161)

## Solution to fix the error.

Edit the file /webapps/manager/META-INF/context.xml:

Previous:

```
<Context antiResourceLocking="false" privileged="true">
    <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127\.\d+\.\d+\>
</Context>
```

Change this file to comment the Value:

```
<Context antiResourceLocking="false" privileged="true">
    <!--
        <Valve className="org.apache.catalina.valves.RemoteAddrValve"
            allow="127\.\d+\.\d+\.:1|0:0:0:0:0:0:1" />
    -->
</Context>
```

## In Jenkins Server

```
#cd /webapps/manager/META-INF/
#vi context.xml
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74# cd webapps/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps# ls
ROOT  docs  examples  host-manager  manager
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps# cd manager/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps/manager# ls
META-INF  WEB-INF  css  images  index.jsp  status.xsd  xform.xsl
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps/manager# cd META-INF/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps/manager/META-INF# ls
context.xml
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps/manager/META-INF# vi context.xml █
```

## In Context.xml file

Change the file to comment the value by mentioning

<Context

<!--

-->

</Context>

## As shown below Screenshot.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more
    contributor license agreements. See the NOTICE file distributed with
    this work for additional information regarding copyright ownership.
    The ASF licenses this file to You under the Apache License, Version 2.0
    (the "License"); you may not use this file except in compliance with
    the License. You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<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+\.:1|0:0:0:0:0:0:1" />
        <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.CarfPrevention
Filter$IRuleValue(?:\$1)?java\.util\.?:LinkedHashMap"/>
    -->
</Context>
-->
```

## Then Restart the Tomcat Service

# cd /opt/apache-tomcat-9.0.74/bin/

#./shutdown.sh

#./startup.sh

```
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/webapps/manager/META-INF# cd /opt/apache-tomcat-9.0.74/bin/
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# ls
bootstrap.jar      configtest.bat  digest.sh      setclasspath.sh  startup.sh      tool-wrapper.sh
catalina-juli.jar   configtest.sh   makebase.bat  shutdown.bat   tomcat-juli.jar  version.bat
catalina.bat       commons-daemon-native.tar.gz  daemon.sh     makebase.sh   shutdown.sh   tomcat-native.tar.gz  version.sh
catalina.sh        commons-daemon.jar   digest.bat   setclasspath.bat  startup.bat   tool-wrapper.bat
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# ./shutdown.sh
Using CATALINA_BASE:  /opt/apache-tomcat-9.0.74
Using CATALINA_HOME:  /opt/apache-tomcat-9.0.74
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.74/temp
Using JRE_HOME:       /usr
Using CLASSPATH:      /opt/apache-tomcat-9.0.74/bin/bootstrap.jar:/opt/apache-tomcat-9.0.74/bin/tomcat-juli.jar
Using CATALINA_OPTS: 
NOTE: Picked up JDWP JAVA_OPTS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi=sun.rmi.transport=ALL-UNNAMED
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# ./startup.sh
Using CATALINA_BASE:  /opt/apache-tomcat-9.0.74
Using CATALINA_HOME:  /opt/apache-tomcat-9.0.74
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.74/temp
Using JRE_HOME:       /usr
Using CLASSPATH:      /opt/apache-tomcat-9.0.74/bin/bootstrap.jar:/opt/apache-tomcat-9.0.74/bin/tomcat-juli.jar
Using CATALINA_OPTS: 
Tomcat started.
root@ip-172-31-31-239:/opt/apache-tomcat-9.0.74/bin# █
```

**Then go back to Jenkins dashboard**

**Again rebuild the autodeployment job**

**Now can able to view that job is successfully deployed.**

The screenshot shows the Jenkins dashboard with the 'autodeployment' job listed. The job has a green icon, indicating it is successful. The last success was 6.8 sec ago, and the last failure was 5 min 3 sec ago. The last duration was 0.34 sec. There are links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

S	W	Name	Last Success	Last Failure	Last Duration
		autodeployment	6.8 sec #11	5 min 3 sec #10	0.34 sec

**In chrome tab**

**Enter the Tomcat Application webpage.**

**<serverIPv4>:9090/sample**

The screenshot shows a browser window displaying the Tomcat application homepage. The URL is <serverIPv4>:9090/sample. The page title is "Sample 'Hello, World'" Application. It features a yellow cat icon and the text: "This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide." Below this, there are links to "To a JSP page" and "To a servlet".

**Successfully Auto-deployed the Sample.warfile to Tomcat Application using Jenkins Dashboard.**