

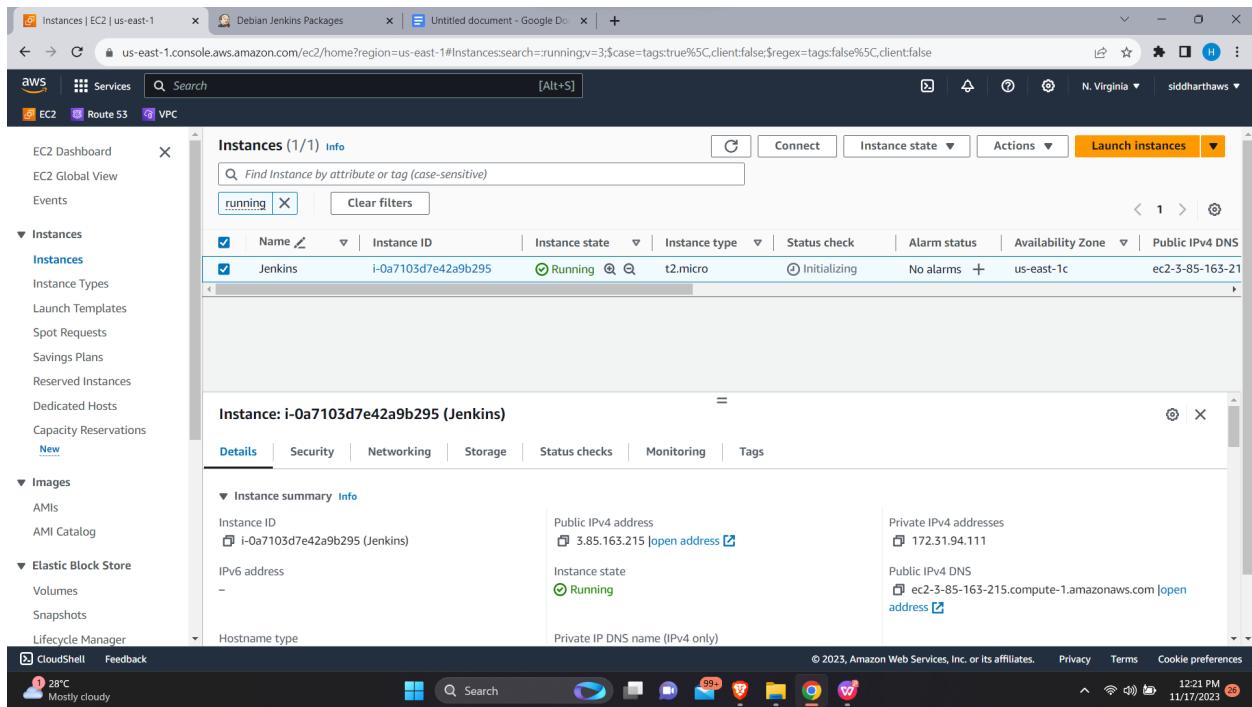
JENKINS

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.

JENKINS INSTALLATION STEPS

Create an ec2 instance - Ubuntu 20.04 - t2.micro - keyfile - Security group(all traffic) - launch instance



Connect to the ec2 instance through putty
Search jenkins.io in browser
Copy & paste the jenkins packages in the server

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system (for the Weekly Release Line):

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable 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 openjdk-17-jre
sudo apt-get install jenkins
```

The apt packages were signed using this key:

```
pub    rsa4096 2023-03-27 [SC] [expires: 2026-03-26]
       63667EE74BBA1F0A08A6987258A31D57EF5975CA
uid           Jenkins Project
sub    rsa4096 2023-03-27 [S] [expires: 2026-03-26]
```

After Successfully installed jenkins.

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

```
#ps -ef | grep jenkins
```

```
ubuntu@ip-172-31-94-111:~$ ps -ef | grep jenkins
jenkins   4967      1  16 06:59 ?        00:00:41 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
ubuntu     5266    1333  0  06:59 pts/0    00:00:00 grep --color=auto jenkins
```

Hit jenkins server pub-ip along with port number in browser

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

.....

Continue

Copy the password from above path using #cat command

```
ubuntu@ip-172-31-94-111:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
d6b6b7190de94a69a9fd6c003a640a01
```

Enter the first Admin User Credentials - Save and Continue
Jenkins dashboard is successfully hosted

STEPS TO CONFIGURE EMAIL NOTIFICATION

In jenkins dashboard, go to Manage Jenkins - System

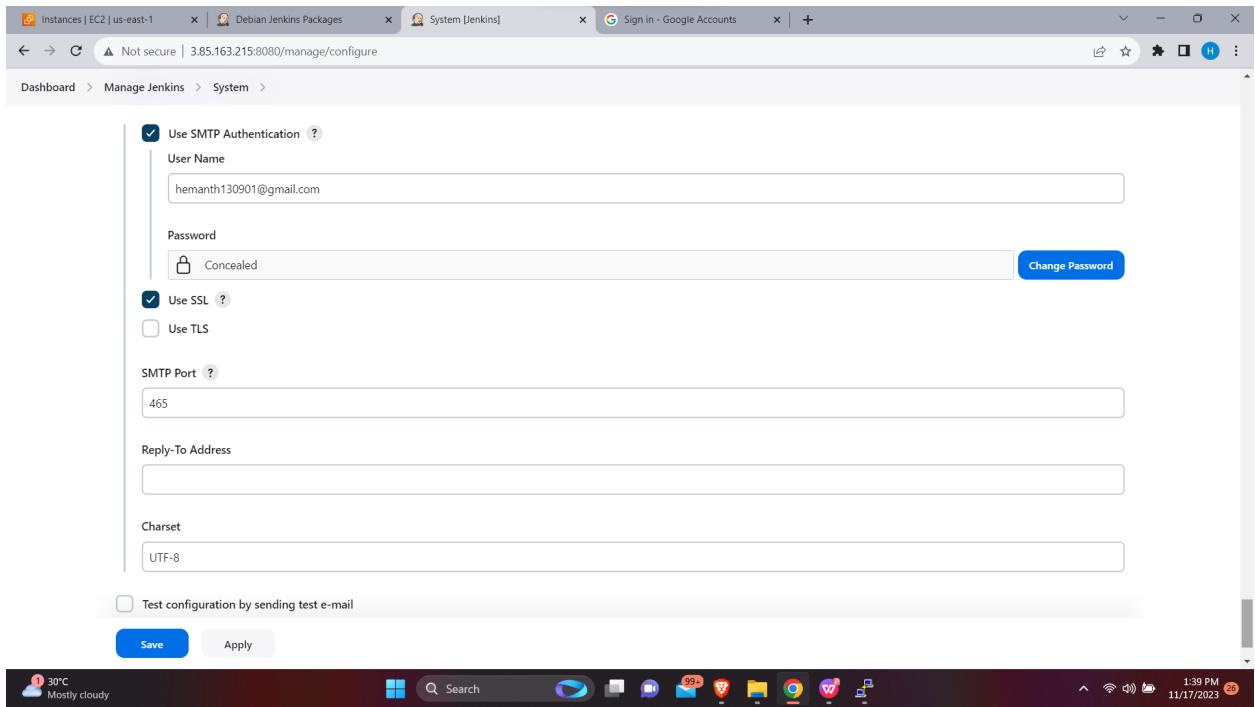
The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links for New Item, People, Build History, Manage Jenkins (which is selected), and My Views. Below this are sections for Build Queue (empty) and Build Executor Status (1 Idle). The main area is titled "Manage Jenkins" and contains a "System Configuration" section with tabs for System, Tools, Nodes, Clouds, and Plugins. A message at the top says: "Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)." At the bottom of the page, there are "Set up agent", "Set up cloud", and "Dismiss" buttons.

Scroll down to email notification

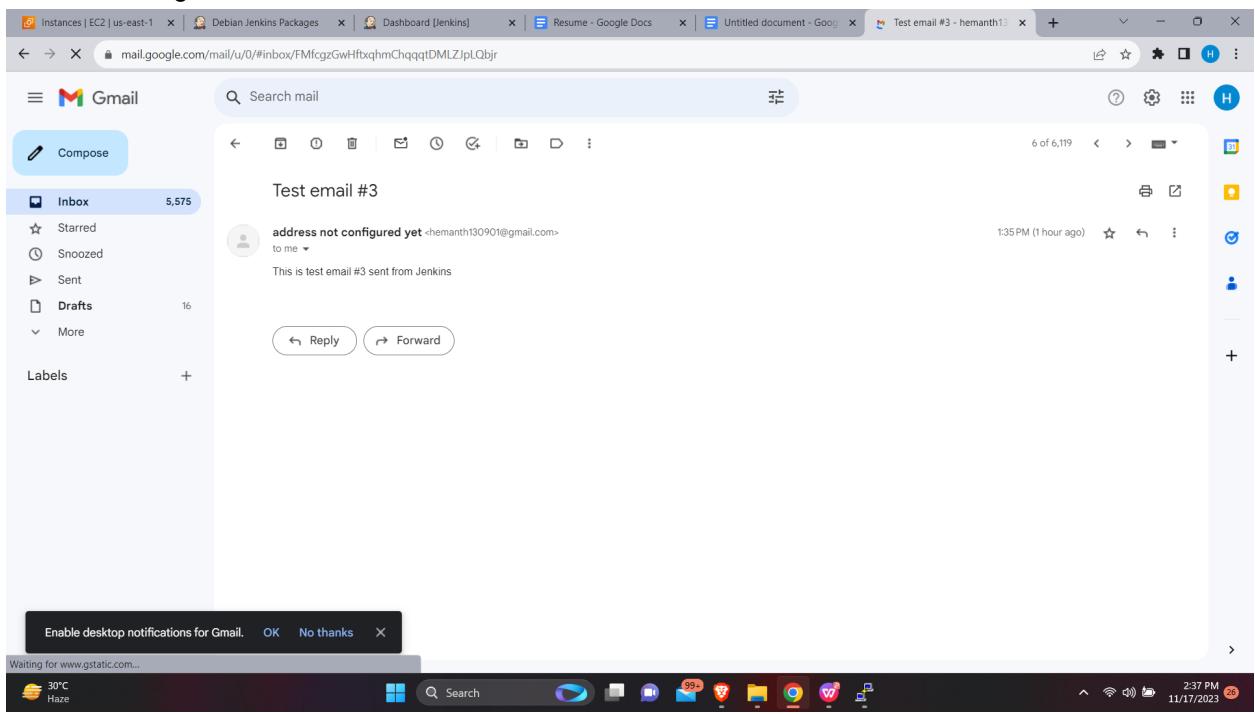
The screenshot shows the Jenkins System configuration page under the "E-mail Notification" tab. It includes fields for "SMTP server" (smtp.gmail.com), "Default user e-mail suffix" (@gmail.com), and an "Advanced" dropdown. There's also a checkbox for "Test configuration by sending test e-mail". At the bottom are "Save" and "Apply" buttons. The status bar at the bottom right shows "12:52 PM 11/17/2023".

In google account, go to manage your account - security - 2 step verification - get password - create password

Then click advanced and configure recipient's email



Click test configuration



Successfully Configured Email Notification in Jenkins Dashboard.

Now create 2 jobs and add configured email to get notification if the job fails.

Create a job(first job)

In configure - Build Steps - choose execute shell - enter Wrong command to ensure the email notification works properly if the job gets failed - In Post build Action - Enter already configured Email-ID - Save.

The screenshot shows the Jenkins job configuration interface for a job named "first job".

Build Steps:

- Execute shell:
 - Command: borut

Post-build Actions:

- E-mail Notification:
 - Recipients: hemanth130901@gmail.com
 - Send e-mail for every unstable build (checked)
 - Send separate e-mails to individuals who broke the build (unchecked)

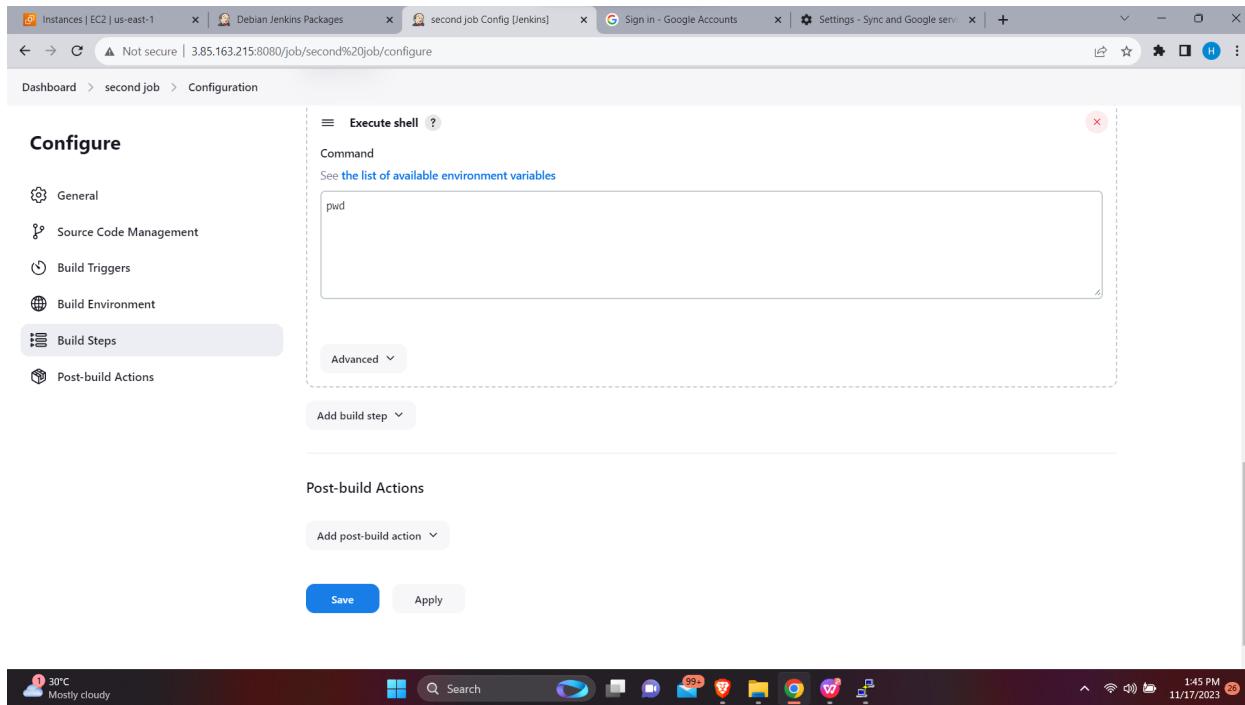
Toolbar:

- Instances | EC2 | us-east-1
- Debian Jenkins Packages
- first job Config [Jenkins]
- Sign in - Google Accounts
- Settings - Sync and Google service

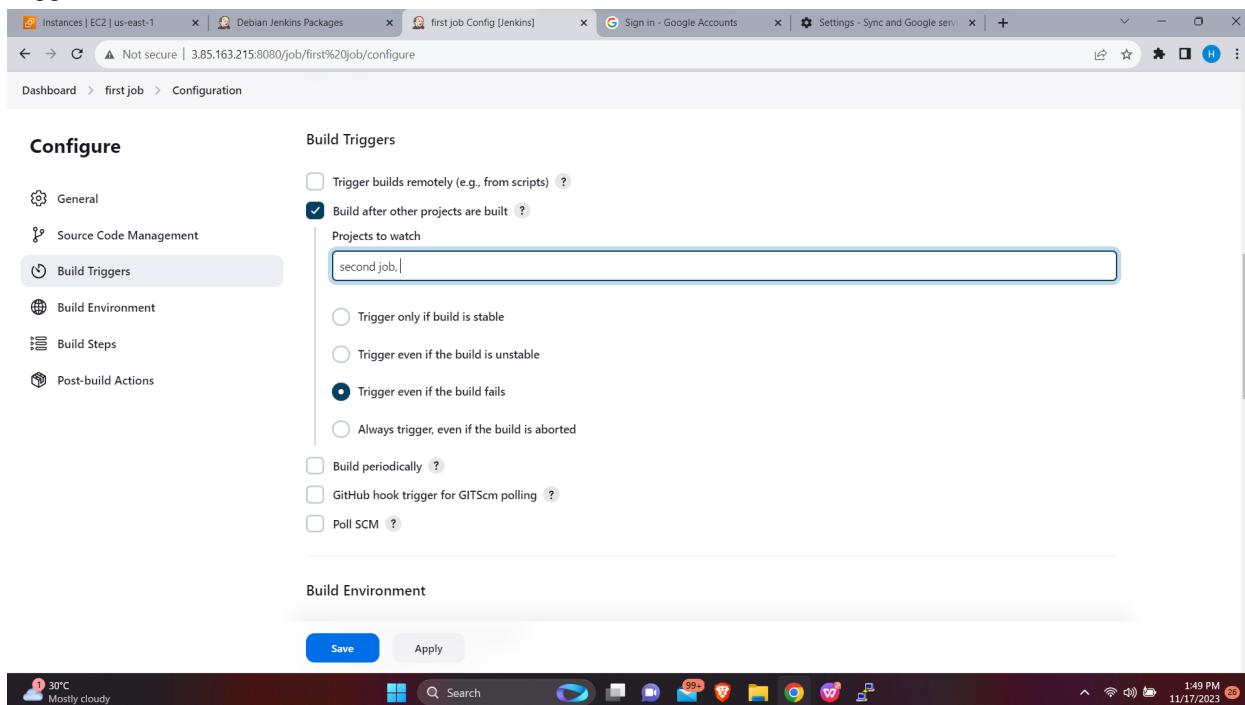
System Status:

- 30°C Mostly cloudy
- Search bar
- Icons for various services (Windows Taskbar style)
- 1:44 PM 11/17/2023
- REST API Jenkins 2.426.1

Create another job - Set a job name (SecondJob) - Freestyle project - Ok.
In configure - Build Steps - Choose Execute Shell - Enter working command in ubuntu server Eg: pwd - Save.



In configure - build triggers - Choose build other projects - Enter the Second job name - Choose Trigger even if the build fails - save.



In dashboard - Start the job - Can view the First job gets failure and automatically second job get success as we Configure the jobs.

Ensure Email notification is received for job failure.

The screenshot shows a web browser with several tabs open. The active tab is 'Build failed in Jenkins: first job #1' at 3.85.163.215:8080. The page content includes:

- Instances | EC2 | us-east-1**
- Debian Jenkins Packages**
- Dashboard [Jenkins]**
- Build failed in Jenkins: first job #1**

The Jenkins dashboard shows the following data:

S	W	Name	Last Success	Last Failure	Last Duration
✗	☁️	first job	N/A	6 min 40 sec #3	0.49 sec
✓	☀️	second job	6 min 48 sec #1	N/A	28 ms

Build Queue: No builds in the queue.

Build Executor Status:

- 1 Idle
- 2 Idle

At the bottom of the page, there is a message about enabling desktop notifications for Gmail, with 'OK' and 'No thanks' buttons.

The browser's address bar shows 'mail.google.com/mail/u/0/#inbox/...'. The system tray at the bottom displays a weather forecast for Manali Road: 30°C, Mostly cloudy.

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

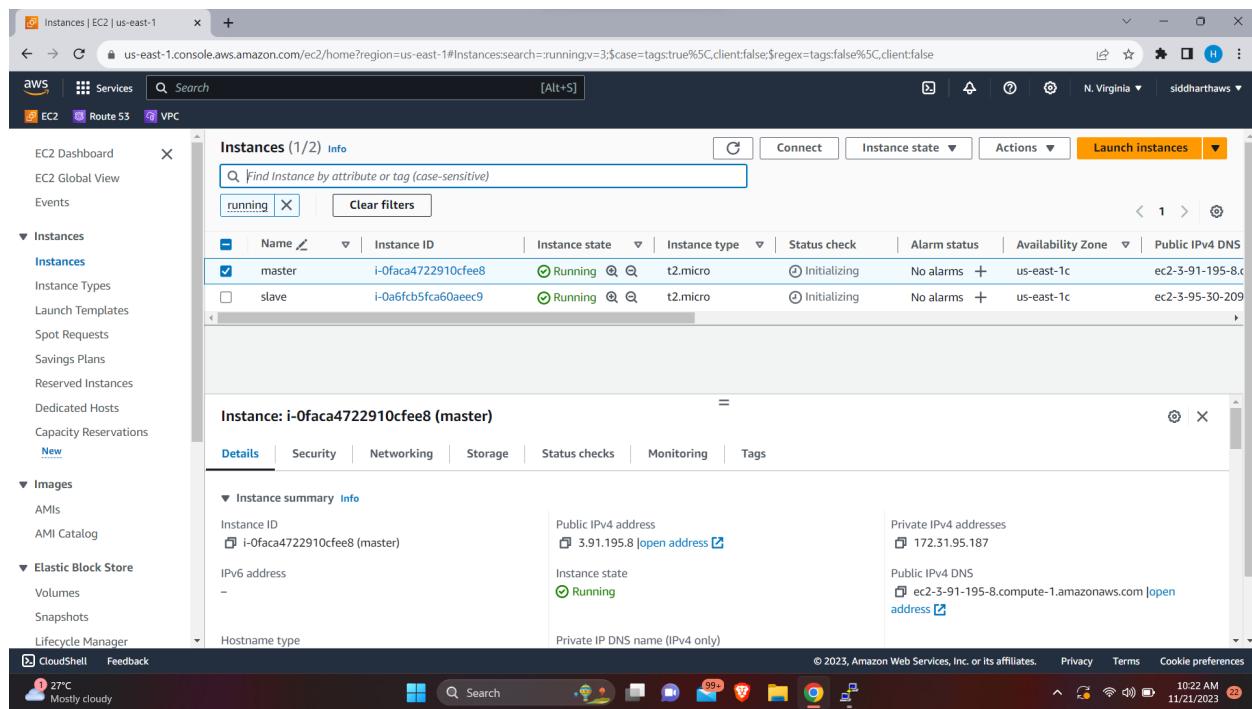
MASTER/SLAVE SERVER CONFIGURATION

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 SLAVE SERVER

Launch 2 instances - ubuntu 20.04 - choose keyfile - security group(all traffic) - name one server as master & another as slave.



Connect to the master server and install jenkins packages in it.

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system (for the Weekly Release Line):

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable 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 openjdk-17-jre
sudo apt-get install jenkins
```

The apt packages were signed using this key:

```
pub rsa4096 2023-03-27 [SC] [expires: 2026-03-26]
  63667EE74BBA1F0A08A6987258A31D57EF5975CA
uid   Jenkins Project
sub rsa4096 2023-03-27 [S] [expires: 2026-03-26]
```

Connect to the slave server & install java in it.

```
ubuntu@ip-172-31-89-17:~$ sudo apt-get install fontconfig openjdk-17-jre
```

Login to jenkins dashboard using master server ip along with jenkins port no(8080)

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.426.1

Skip and continue as admin Save and Continue

Successfully logged in to the jenkins dashboard

The screenshot shows the Jenkins dashboard with the following elements:

- Top Bar:** Instances | EC2 | us-east-1, Debian Jenkins Packages, Dashboard [Jenkins], Not secure | 3.91.195.8:8080, Search (CTRL+K), Jenkins logo, Hemanth Kumar B, log out.
- Header:** Jenkins, Search (CTRL+K), Add description.
- Left Sidebar:** New Item, People, Build History, Manage Jenkins, My Views, Build Queue (No builds in the queue), Build Executor Status (1 idle, 2 idle).
- Middle Content:** Welcome to Jenkins!, Start building your software project, Create a job, Set up a distributed build, Set up an agent, Configure a cloud, Learn more about distributed builds.
- Bottom Right:** REST API, Jenkins 2.426.1.
- Taskbar:** Weather (27°C Light rain), Search, various icons (calculator, file manager, etc.), 10:43 AM, 11/21/2023.

In dashboard - select manage jenkins - set up agent

The screenshot shows the Jenkins Manage Jenkins > Nodes page with the following elements:

- Top Bar:** Instances | EC2 | us-east-1, Debian Jenkins Packages, Nodes [Jenkins], Not secure | 3.91.195.8:8080/manage/computer/, Search (CTRL+K), Jenkins logo, Hemanth Kumar B, log out.
- Header:** Nodes, Search (CTRL+K), New Node, Node Monitoring, Refresh icon.
- Left Sidebar:** Nodes, Clouds, Build Queue (No builds in the queue), Build Executor Status (1 idle, 2 idle).
- Middle Content:** Nodes table:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	4.99 GB	! 0 B	4.99 GB	0ms
2	Data obtained	15 min	15 min	15 min	15 min	15 min	15 min
- Bottom Right:** REST API, Jenkins 2.426.1.
- Taskbar:** Weather (27°C Light rain), Search, various icons, 10:43 AM, 11/21/2023.

Create a node to sync the Master server and Slave server.
Select a name - select permanent agent - create

Instances | EC2 | us-east-1 | Debian Jenkins Packages | Node1 Configuration [Jenkins] | JENKINS EMAIL NOTIFICATION | 3.91.195.8:8080/computer/Node1/configure

Not secure | 3.91.195.8:8080/computer/Node1/configure

Dashboard > Nodes > Node1 > Configure

Status
Delete Agent
Configure
Build History
Load Statistics
Script Console
Log
System Information
Disconnect

Name ?
Node1

Description ?
Plain text Preview

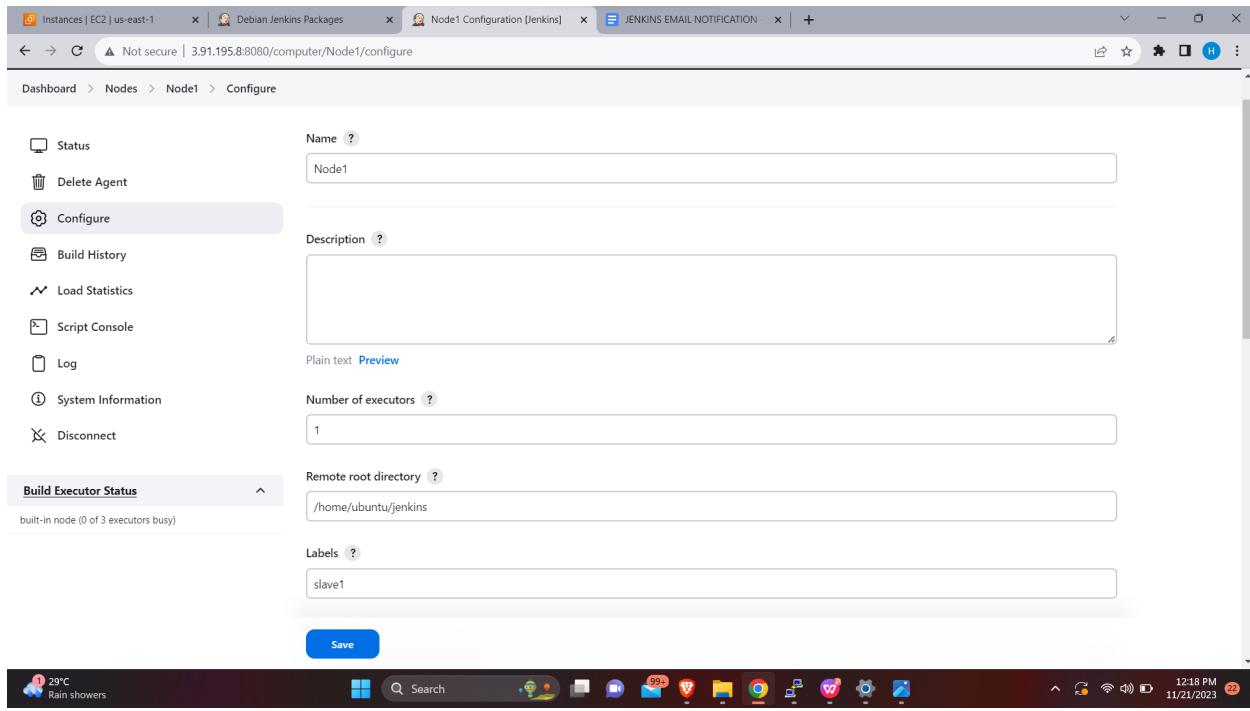
Number of executors ?
1

Remote root directory ?
/home/ubuntu/jenkins

Labels ?
slave1

Save

29°C Rain showers Search 12:18 PM 11/21/2023



Instances | EC2 | us-east-1 | Debian Jenkins Packages | Node1 Configuration [Jenkins] | JENKINS EMAIL NOTIFICATION | 3.91.195.8:8080/computer/Node1/configure

Not secure | 3.91.195.8:8080/computer/Node1/configure

Dashboard > Nodes > Node1 > Configure

Usage ?
Use this node as much as possible

Launch method ?
Launch agent by connecting it to the controller

Disable WorkDir ?

Custom WorkDir path ?
[empty input field]

Internal data directory ?
remoting

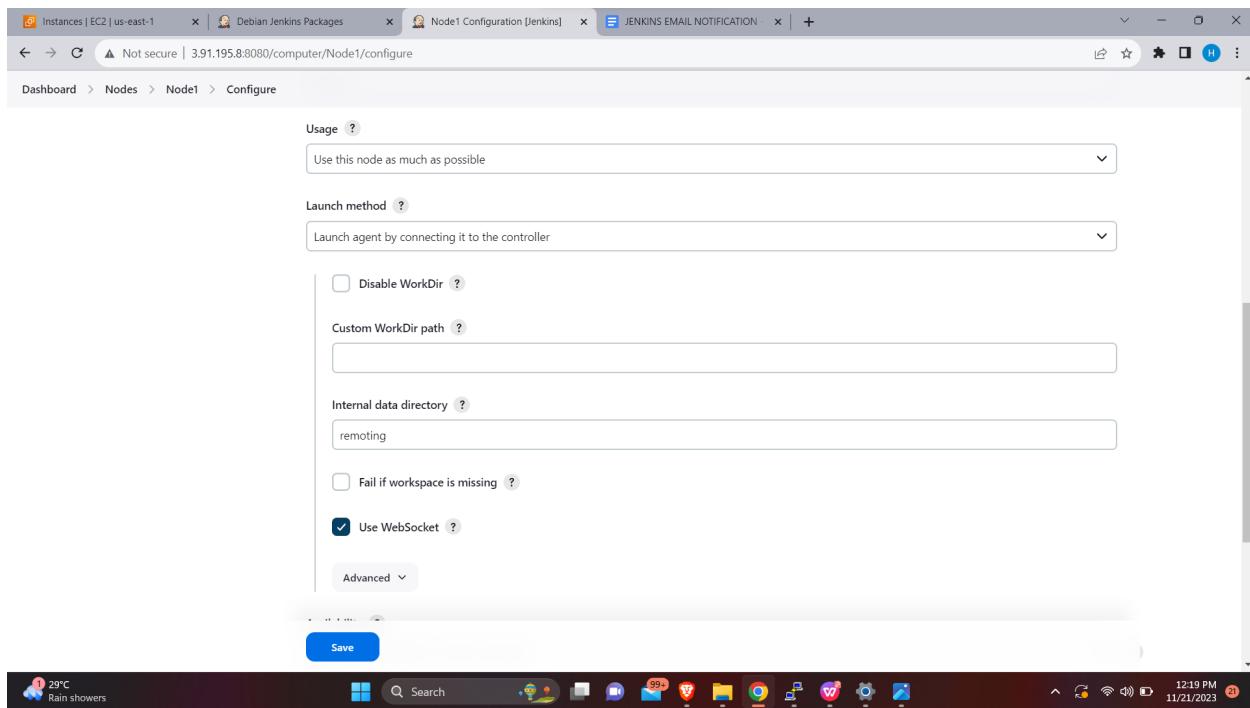
Fail if workspace is missing ?

Use WebSocket ?

Advanced ▾

Save

29°C Rain showers Search 12:19 PM 11/21/2023



Availability ?

Keep this agent online as much as possible

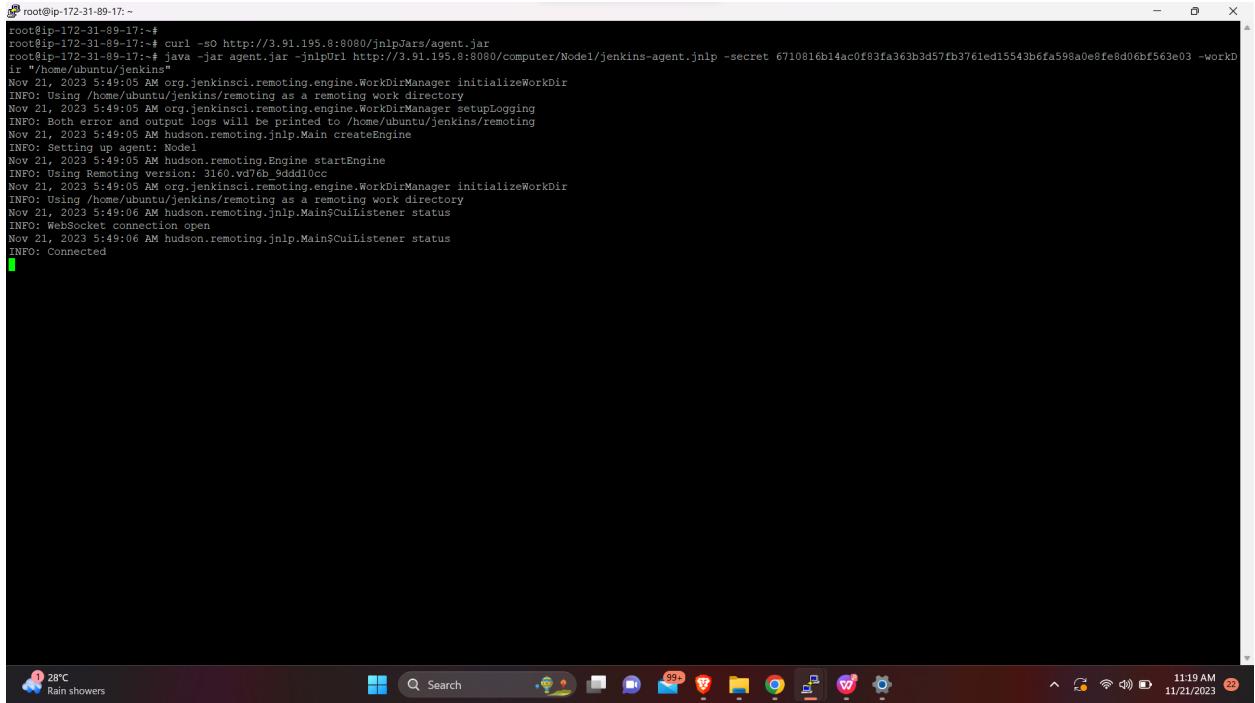
Node Properties

- Disable deferred wipeout on this node ?
- Environment variables
- Tool Locations

Save

The created node will not be synced yet. To sync it, select the created node & copy the command & paste it in the slave server.

The screenshot shows the Jenkins Node configuration page for 'Node1'. The top navigation bar includes tabs for 'Instances | EC2 | us-east-1', 'Debian Jenkins Packages', and 'Node1 [Jenkins]'. The main content area is titled 'Agent Node1' and contains sections for 'Status', 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', and 'Log'. Below these are sections for 'Run from agent command line: (Unix)' and 'Run from agent command line: (Windows)', each containing a code block with Jenkins agent startup commands. At the bottom of the page is a 'Build Executor Status' dropdown. The status bar at the bottom of the browser window shows the weather as '28°C Rain showers' and various system icons.



```

root@ip-172-31-89-17: ~
root@ip-172-31-89-17: ~# curl -sO http://3.91.195.8:8080/jnlpJars/agent.jar
root@ip-172-31-89-17: ~# java -jar agent.jar -jnlpUrl http://3.91.195.8:8080/computer/Node1/jenkins-agent.jnlp -secret 6710816b14ac0f83fa363b3d57fb3761ed15543b6fa598a0e8fe8d06bf563e03 -workDir "/home/ubuntu/jenkins"
Nov 21, 2023 5:49:05 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Nov 21, 2023 5:49:05 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/jenkins/remoting
Nov 21, 2023 5:49:05 AM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: Node1
Nov 21, 2023 5:49:05 AM hudson.remoting.Engine startEngine
INFO: Engine started, version 3160.vd7fb9dd10cc
Nov 21, 2023 5:49:05 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Nov 21, 2023 5:49:06 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: WebSocket connection open
Nov 21, 2023 5:49:06 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected

```

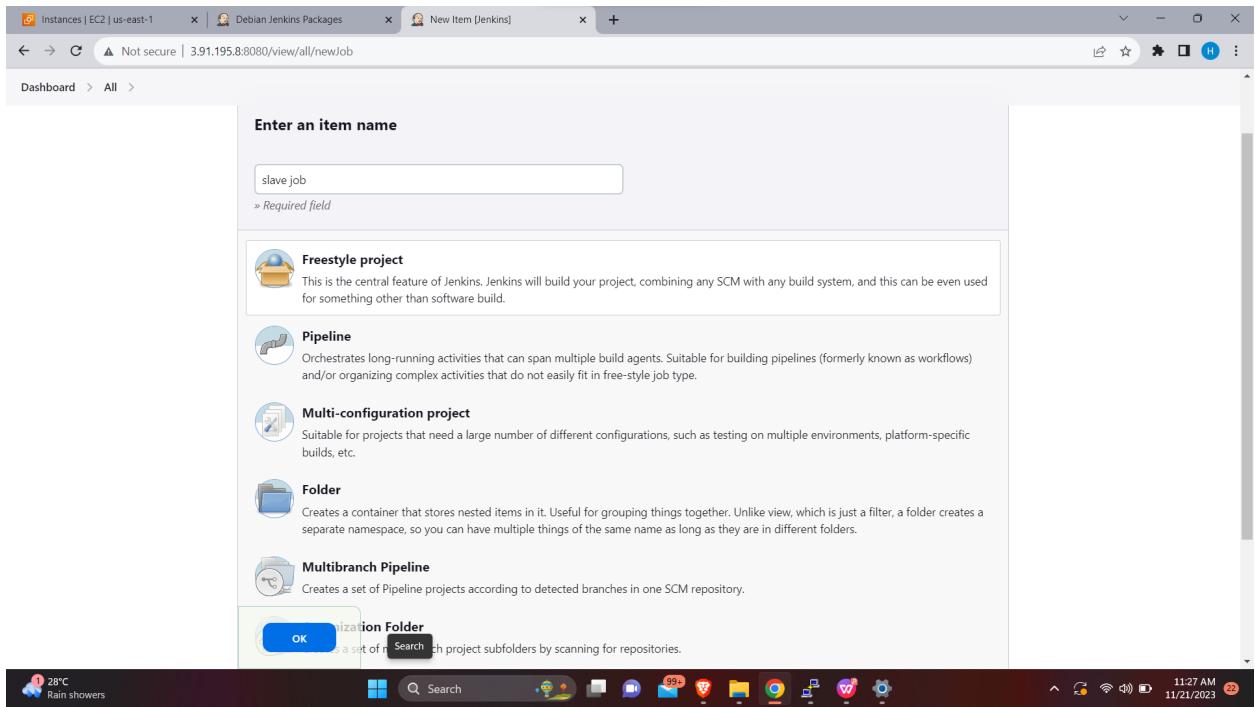
Now the created node will be in sync with the master.



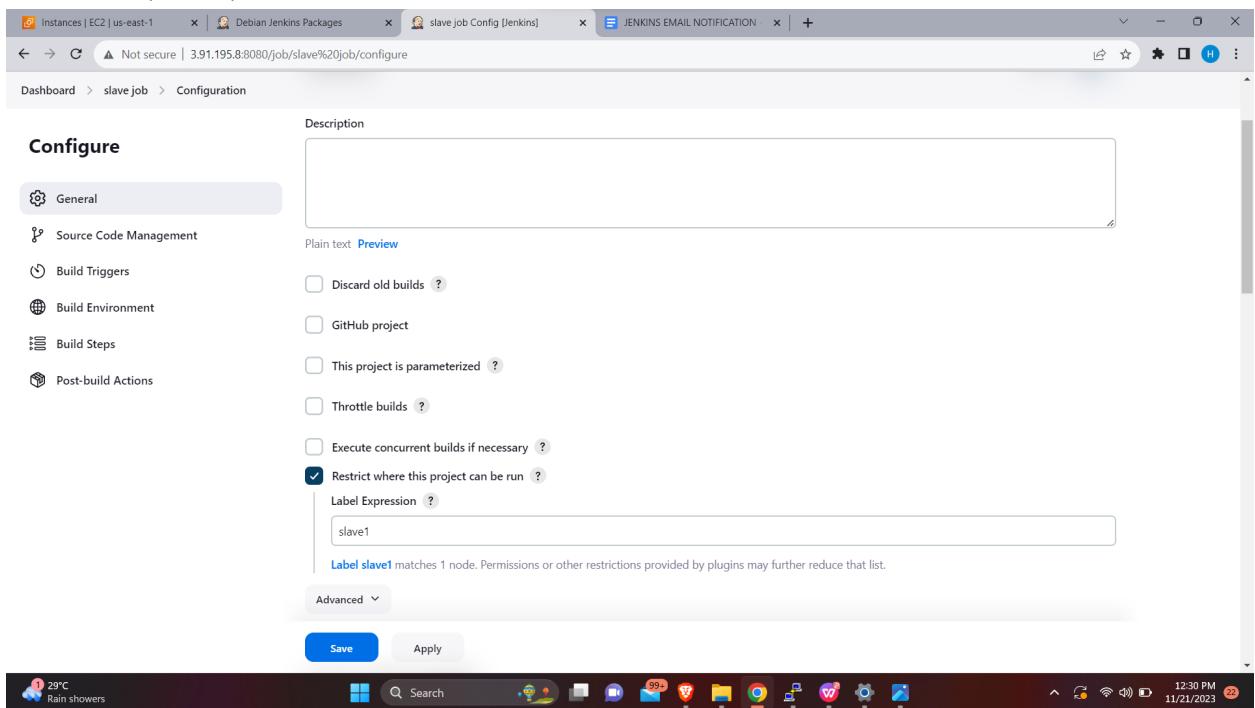

The screenshot shows the Jenkins "Nodes" page with the URL [Not secure | 3.91.195.8:8080/computer/](http://3.91.195.8:8080/computer/). The page title is "Nodes". The main content area displays the "Nodes" table with the following data:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.78 GB	0 B	4.78 GB	0ms
	Node1	Linux (amd64)	In sync	5.25 GB	0 B	5.25 GB	118ms

In Dashboard - New items - Enter Item name (slave job) choose Freestyle project - save



In Configure - general - Choose Restrict where this project can be run - In label Expression - Choose The created(slave1) label.



In build Steps - Create a Execute Shell - Save.

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](#)

```
date
```

Save **Apply**

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

Dashboard - Manage jenkins - Nodes - choose the created node(node1) - select the created label(slave1)

S	W	Name	Last Success	Last Failure	Last Duration
		slave job	21 sec #1	N/A	0.77 sec

Icon: S W M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

Successfully configured Master/Slave server. Managed both master and slave server (In sync) with help of creating Node in jenkins dashboard webserver.

AUTODEPLOYMENT

STEPS TO DEPLOY A WAR FILE INTO TOMCAT SERVER USING JENKINS

Create an ec2 instance

Install java & jenkins in it

The screenshot shows the AWS EC2 Instances page. The main table lists one instance named 'Jenkins' with the ID 'i-0dcdbd650930fa4110'. The instance is shown as 'Running' with an 'Initializing' status check. It is an 't2.micro' instance type, located in the 'us-east-1c' availability zone, with a public IPv4 DNS name 'ec2-34-205-156-125.compute-1.amazonaws.com'. The sidebar on the left shows various EC2 management options like Instances, AMIs, and Elastic Block Store.

```
root@ip-172-31-88-106:~# java --version
openjdk 17.0.8.1 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu120.04, mixed mode, sharing)
root@ip-172-31-88-106:~# ps -ef | grep jenkins
jenkins      5340      1 65 10:33 ?    00:00:38 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
root       5640  1342  0 10:33 pts/0    00:00:00 grep --color=auto jenkins
root@ip-172-31-88-106:~#
```

In browser - search for tomcat - open tomcat official website - choose any version & copy tar.gz URL

In the jenkins server paste the copied tomcat URL to download tomcat

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

```
root@ip-172-31-88-106:~# cd /opt
root@ip-172-31-88-106:/opt# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.83/bin/apache-tomcat-9.0.83.tar.gz
--2023-11-25 10:35:00-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.83/bin/apache-tomcat-9.0.83.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42:644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11801705 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.83.tar.gz'

apache-tomcat-9.0.83.tar.gz      100%[=====] 11.25M --.-KB/s   in 0.07s

2023-11-25 10:35:00 (152 MB/s) - `apache-tomcat-9.0.83.tar.gz' saved [11801705/11801705]
root@ip-172-31-88-106:/opt#
```

Untar the tomcat file

```
#tar -xvzf <tar file name>
```

```
root@ip-172-31-88-106:/opt# tar -xvzf apache-tomcat-9.0.83.tar.gz
```

Directory will be created. Get in to the directory and open server.xml file in vi editor

```
root@ip-172-31-88-106:/opt# cd apache-tomcat-9.0.83/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# cd conf/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf# ls
catalina.policy  catalina.properties  context.xml  jaspic-providers.xml  jaspic-providers.xsd  logging.properties  server.xml  tomcat-users.xml  tomcat-users.xsd  web.xml
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf# vi server.xml
```

In the server.xml file, Change the Tomcat server default Connector port from 8080 to something(say 9090)

```

<!-- A "Service" is a collection of one or more "Connectors" that share
     a single "Container". Note: A "Service" is not itself a "Container",
     so you may not define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/service.html
-->
<service name="Catalina">

    <!-- The connectors can use a shared executor, you can define one or more named thread pools-->
    <!--
        <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
            maxThreads="1500" minSpareThreads="4"/>
    -->

    <!-- 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="8080" 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"
            />
    -->
    <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
        This connector uses the NIO implementation. The default
        SSLImplementation will depend on the presence of the APR/native
        library and the useOpenSSL attribute of the APRLifecycleListener.
        Either JSSE or OpenSSL style configuration may be used regardless
        of the SSLImplementation selected. JSSE style configuration is used below.
    -->
-- INSERT --

```

Then go the bin directory & start the tomcat service

```

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

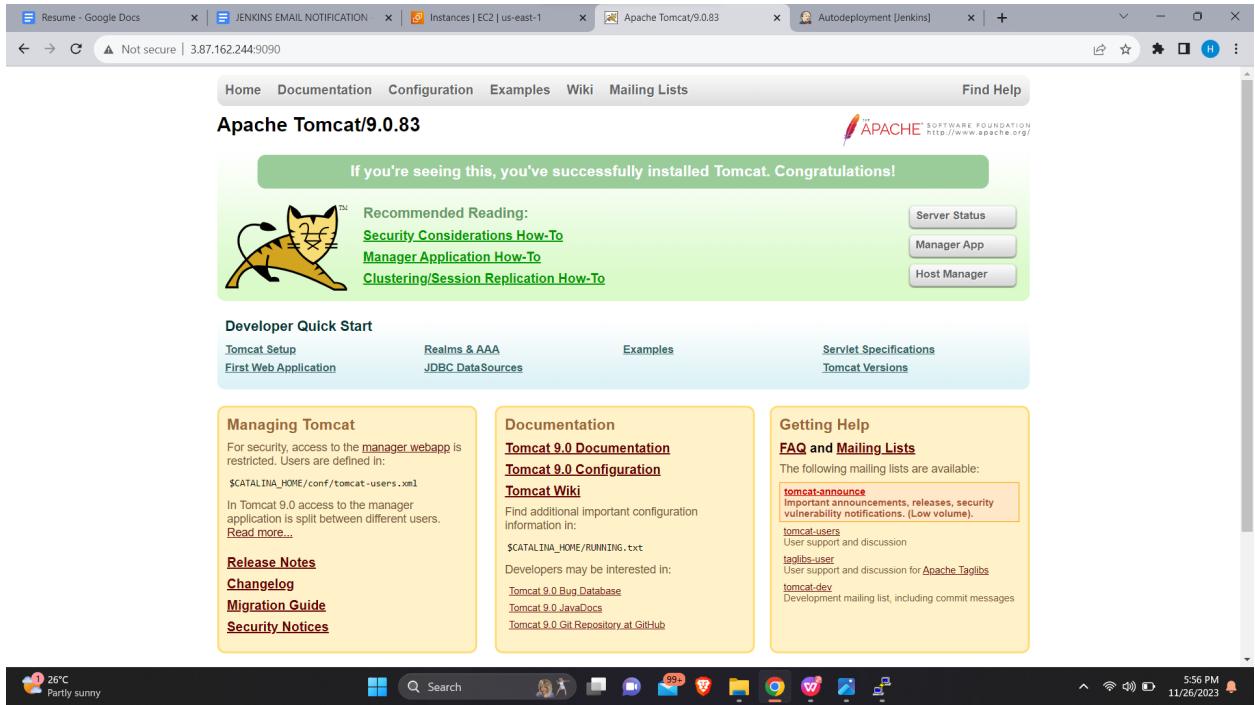
```

```

root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf# cd ..
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# cd bin/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin# ls
bootstrap.jar      catalina.sh  commons-daemon-native.tar.gz  configtest.sh  digest.sh   setclasspath.bat  shutdown.sh  tomcat-juli.jar  tool-wrapper.sh
catalina-tasks.xml  ciphers.bat  commons-daemon.jar       daemon.sh    makebase.bat  setclasspath.sh  startup.bat  tomcat-native.tar.gz version.bat
catalina.bat        ciphers.sh   configtest.bat          digest.bat   makebase.sh   shutdown.bat   startup.sh   tool-wrapper.bat version.sh
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin# ./startup.sh
Using CATALINA_BASE:  /opt/apache-tomcat-9.0.83
Using CATALINA_HOME:   /opt/apache-tomcat-9.0.83
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.83/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-9.0.83/bin/bootstrap.jar:/opt/apache-tomcat-9.0.83/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin#

```

Now hit the tomcat server public along with port number in the browser. <server pub-ip>:9090



Login to the jenkins dashboard.

Create a new freestyle project job in name of AutoDeployment.

Enter an item name

Autodeployment
» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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

OK

Select the created job & click configure - In build jobs - Choose execute a shell command - write a shell command (say pwd) - Then apply & save.

The screenshot shows the Jenkins configuration interface for an 'Autodeployment' job. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. The main area is titled 'Configure' and contains a 'Command' section with the command 'pwd' entered. Below it is an 'Advanced' dropdown and a 'Post-build Actions' section with a 'Add post-build action' button. At the bottom are 'Save' and 'Apply' buttons. A green 'Saved' message is displayed at the top right.

Build the Job Now - Ensure the Job deployed successfully.

The screenshot shows the Jenkins dashboard. The top navigation bar includes links for 'Instances | EC2 | us-east-1', 'Apache Tomcat 9 - Apache Tom...', 'Apache Tomcat/9.0.83', and 'Autodeployment Config [Jenkins]'. The main content area displays the 'Autodeployment' job details. It shows the job name 'Autodeployment', its status as 'Last Success' (15 sec ago), and its last failure (N/A). Below this is a 'Build Queue' section stating 'No builds in the queue.' and a 'Build Executor Status' section showing '1 Idle' and '2 Idle'. The bottom of the screen shows a taskbar with various icons and the system tray.

In Jenkins Server, go to jenkins workspace and ensure the job directory is created and ensure nothing is in the directory.

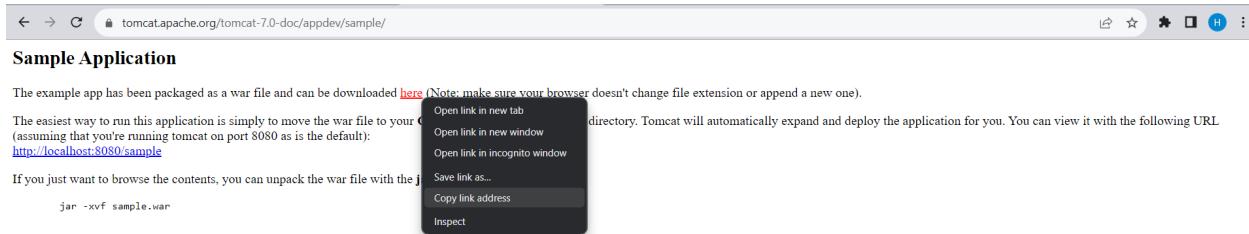
```
#cd /var/lib/jenkins/workspace
#cd autodeployment/
#ls
```

```

root@ip-172-31-88-106:~# cd /var/lib/jenkins/workspace/
root@ip-172-31-88-106:/var/lib/jenkins/workspace# ls
Autodeployment
root@ip-172-31-88-106:/var/lib/jenkins/workspace# cd Autodeployment/
root@ip-172-31-88-106:/var/lib/jenkins/workspace/Autodeployment# ls
root@ip-172-31-88-106:/var/lib/jenkins/workspace/Autodeployment# 

```

In chrome, browse for tomcat sample war file & copy the download url for the tomcat sample war file.



Paste the war file URL in autodeployment directory.

```
#wget <tomcat sample war file url>
```

```
#||
```

```

root@ip-172-31-88-106:/var/lib/jenkins/workspace/Autodeployment# wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
--2023-11-25 10:43:36-- 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

2023-11-25 10:43:36 (40.2 MB/s) - 'sample.war' saved [4606/4606]
root@ip-172-31-88-106:/var/lib/jenkins/workspace/Autodeployment# 

```

```

root@ip-172-31-88-106:/var/lib/jenkins/workspace/Autodeployment# ll
total 16
drwxr-xr-x 2 jenkins jenkins 4096 Nov 25 10:43 .
drwxr-xr-x 3 jenkins jenkins 4096 Nov 25 10:41 ..
-rw-r--r-- 1 root    root    4606 Mar 31 2012 sample.war

```

In Jenkins dashboard, download the deploy to the container plugin to attach the war file in the freestyle project job.

Manage jenkins - plugins - Available plugins - search for deploy to the container plugin - install.

The screenshot shows the Jenkins Plugins page. On the left, there's a sidebar with links: Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main area has a search bar at the top with the placeholder 'deploy to'. Below it is a table of available plugins:

Install	Name	Released
<input checked="" type="checkbox"/>	Deploy to container 1.16	3 yr 0 mo ago
<input type="checkbox"/>	Azure Virtual Machine Scale Set 0.2.4	2 yr 9 mo ago
<input type="checkbox"/>	Package Drone Deployer 0.6.0	5 yr 3 mo ago
<input type="checkbox"/>	Deploy to webMethods Integration Server 1.0	11 mo ago

Below the table, there's a note: "This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information." The status bar at the bottom shows the weather as 29°C Mostly cloudy, the date as 11/25/2023, and the time as 4:14 PM.

After successfully installing the plugin, go to the autodeployment job which is already created. Select the autodeployment job & click configure - In post build actions - choose deploy war/ear to a container option.

The screenshot shows the Jenkins Autodeployment Configuration page. On the left, there's a sidebar with links: General, Source Code Management, Build Triggers, Build Environment, Build Steps (which is selected), and Post-build Actions. The main area has a 'Configure' section with a 'Command' field containing 'See [the list of available environment variables](#)'. A dropdown menu is open over the 'Deploy war/ear to a container' option, which is highlighted with a red box. Other options in the dropdown include: Aggregate downstream test results, Archive the artifacts, Build other projects, Publish JUnit test result report, Record fingerprints of files to track usage, Git Publisher, E-mail Notification, Editable Email Notification, Set GitHub commit status (universal), Set build status on GitHub commit [deprecated], and Delete workspace when build is done. At the bottom, there are 'Save' and 'Apply' buttons, and a status bar at the bottom shows the weather as 29°C Mostly cloudy, the date as 11/25/2023, and the time as 4:15 PM.

In WAR/EAR files:

Enter (Sample.war)

In context path:

Enter /sample

Containers:

Choose Tomcat version
Ex: Tomcat 9.x Remote

Credentials:

Add jenkins

Tomcat URL

Enter tomcat test page url

Leave the rest as default.

The screenshot shows the Jenkins Autodeployment Config job configuration screen. The 'Post-build Actions' section is selected. Under 'Deploy war/ear to a container', a 'WAR/EAR files' field contains 'sample.war'. The 'Context path' field contains '/sample'. Under 'Containers', a 'Tomcat 9.x Remote' section is expanded, showing 'Credentials' set to 'admin/******** (creds)' and 'Tomcat URL' set to 'http://34.205.156.125:9090/'. At the bottom are 'Save' and 'Apply' buttons.

In the apache tomcat file, enter the user details already created in the credentials in tomcat-users.xml file.

```
#cd /opt/apache-tomcat-9.0.74/conf/  
#vi tomcat-users.xml
```

```
root@ip-172-31-88-106:~# cd /opt/apache-tomcat-9.0.83/conf/  
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf# ls  
Catalina catalina.policy catalina.properties context.xml jaspic-providers.xml jaspic-providers.xsd logging.properties server.xml tomcat-users.xml tomcat-users.xsd web.xml  
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf# vi tomcat-users.xml
```

Enter the xml user content under the Tomcat-users

```

root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf</pre>
<!-- 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. -->
<!-->
<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="must-be-changed" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
<user username="Deployer" password="Deployer" roles="manager-script"/>
</tomcat-users>
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.
Built-in Tomcat manager roles:
- manager-gui - allows access to the HTML GUI and the status pages
- manager-script - allows access to the HTTP API 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 users below are wrapped in a comment and are therefore ignored. If you wish to configure one or more of these users for use with the manager web application, do not forget to remove the <!--> --> that surrounds them. You will also need to set the passwords to something appropriate.
-->
<!-->
<user username="admin" password="must-be-changed" roles="manager-gui"/>
<user username="robot" password="must-be-changed" roles="manager-script"/>
-->
<!-->
-- INSERT --

```

Then go to the bin directory and restart the tomcat service.

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

```

root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/conf# cd ..
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# cd bin/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin# ./shutdown.sh
Using CATALINA_BASE: /opt/apache-tomcat-9.0.83
Using CATALINA_HOME: /opt/apache-tomcat-9.0.83
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.83/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/apache-tomcat-9.0.83/bin/bootstrap.jar:/opt/apache-tomcat-9.0.83/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.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi=sun.rmi.transport=ALL-UNNAMED
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin# ./startup.sh
Using CATALINA_BASE: /opt/apache-tomcat-9.0.83
Using CATALINA_HOME: /opt/apache-tomcat-9.0.83
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.83/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/apache-tomcat-9.0.83/bin/bootstrap.jar:/opt/apache-tomcat-9.0.83/bin/tomcat-juli.jar
Tomcat started.
```

Now run the job in Jenkins dashboard.

The screenshot shows the Jenkins dashboard at <http://34.205.156.125:8080>. The 'Autodeployment' job is listed in the build history with a status of '#1'. The job has failed twice, with the most recent failure occurring 2.8 seconds ago. Below the dashboard, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (one idle executor and one busy executor for Autodeployment).

The job(autodeployment) gets failed.
Go to the job console output - Find the Error.

The screenshot shows the Jenkins job console output for 'Autodeployment #2' at <http://34.205.156.125:8080/job/Autodeployment/2/console>. The error message indicates that the build step failed with an exception due to a Tomcat Manager exception. The error line is highlighted in blue.

```
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/Autodeployment/sample.war to container Tomcat 9.x Remote with context /sample
ERROR: Build step failed with exception
org.codehaus.cargo.container.ContainerException: Failed to redeploy [/var/lib/jenkins/workspace/Autodeployment/sample.war]
    at org.codehaus.cargo.container.tomcat.internal.AbstractTomcatManagerDeployer.redeploy(AbstractTomcatManagerDeployer.java:176)
    at hudson.plugins.deploy.CargoContainerAdapter.deploy(CargoContainerAdapter.java:81)
    at hudson.plugins.deploy.CargoContainerAdapter$DeployCallable.invoke(CargoContainerAdapter.java:167)
    at hudson.plugins.deploy.CargoContainerAdapter$DeployCallable.invoke(CargoContainerAdapter.java:136)
    at hudson.FilePath.act(FilePath.java:1198)
    at hudson.FilePath.act(FilePath.java:1181)
    at hudson.plugins.deploy.CargoContainerAdapter.redeployFile(CargoContainerAdapter.java:133)
    at hudson.plugins.deploy.PasswordProtectedAdapterCargo.redeployFile(PasswordProtectedAdapterCargo.java:95)
    at hudson.plugins.deploy.DeployPublisher.perform(DeployPublisher.java:113)
    at jenkins.tasks.SimpleBuildStep.perform(SimpleBuildStep.java:123)
    at hudson.tasks.BuildStepCompatibilityLayer.perform(BuildStepCompatibilityLayer.java:80)
    at hudson.tasks.BuildStepMonitor$3.perform(BuildStepMonitor.java:47)
    at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:818)
    at hudson.model.AbstractBuild$AbstractBuildExecution.performAllBuildSteps(AbstractBuild.java:767)
    at hudson.model.BuildableBuildExecution.post2(Build.java:179)
    at hudson.model.AbstractBuild$AbstractBuildExecution.post(AbstractBuild.java:711)
    at hudson.model.Run.execute(Run.java:1918)
    at hudson.model.FreeStyleBuild.run(FreeStyleBuild.java:44)
    at hudson.model.ResourceController.execute(ResourceController.java:101)
    at hudson.model.Executor.run(Executor.java:442)
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.AbstractTomcatManagerDeployer.redeploy(AbstractTomcatManagerDeployer.java:161)
    ... 19 more
Caused by: java.io.IOException: Server returned HTTP response code: 403 for URL: http://34.205.156.125:9090/manager/text/list
    at java.base/java.net.http.HttpURLConnection.getInputStream(HttpURLConnection.java:2002)
    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.AbstractTomcatManagerDeployer.redeploy(AbstractTomcatManagerDeployer.java:161)
```

Copy the Error line and browse in Chrome to find the solution.
Solution to the error is

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+\.\d+|::1|0:0:0:0:0:0:1" />
    -->
</Context>
```

In the jenkins server

```
#cd /webapps/manager/META-INF/
#vi context.xml
```

```
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# ls
BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# cd webapps/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/webapps# ls
ROOT docs examples host-manager manager
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/webapps# cd manager/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/webapps/manager# ls
META-INF WEB-INF css images index.jsp status.xsd xform.xsl
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/webapps/manager# cd META-INF/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/webapps/manager/META-INF# ls
context.xml
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/webapps/manager/META-INF# vi context.xml █
```

In the context.xml file, comment the lines from 19 to 25.

Then Restart the Tomcat Service

```
# cd /opt/apache-tomcat-9.0.74/bin/  
#./shutdown.sh  
# ./startup.sh
```

```
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83# cd bin/
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin/ ls
bootstrap.jar  catalina.sh  commons-daemon-native.tar.gz  configtest.sh  digest.sh  setclasspath.bat  shutdown.sh  tomcat-juli.jar  tool-wrapper.sh
catalina-tasks.xml  ciphers.bat  commons-daemon.jar  daemon.sh  makebase.bat  setclasspath.sh  startup.bat  tomcat-native.tar.gz  version.dat
catalina.bat  ciphers.sh  configtest.bat  digest.bat  makebase.sh  shutdown.bat  startup.sh  tool-wrapper.bat  version.sh
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin# ./shutdown.sh

Using CATALINA_BASE:  /opt/apache-tomcat-9.0.83
Using CATALINA_HOME:  /opt/apache-tomcat-9.0.83
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.83/temp
Using JRE_HOME:  /usr
Using CLASSPATH:  /opt/apache-tomcat-9.0.83/bin/bootstrap.jar:/opt/apache-tomcat-9.0.83/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up UNK 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.base/java.net=ALL-UNNAMED
variables currently set-ALL-UNNAMED -> add-opensjava.rmi/sun.rmi.transport=ALL-UNNAMED
/root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin# ./startup.sh

Using CATALINA_BASE:  /opt/apache-tomcat-9.0.83
Using CATALINA_HOME:  /opt/apache-tomcat-9.0.83
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.83/temp
Using JRE_HOME:  /usr
Using CLASSPATH:  /opt/apache-tomcat-9.0.83/bin/bootstrap.jar:/opt/apache-tomcat-9.0.83/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started!
```

Now go back to Jenkins dashboard & rebuild the autodeployment job.

The screenshot shows the Jenkins dashboard with a successful build for 'Autodeployment'. The build status is green, last success was 14 seconds ago, and it has had 3 successes and 2 failures. The dashboard also includes sections for Build Queue (empty), Build Executor Status (2 idle), and various Jenkins management links.

Now the job is deployed successfully.

In the browser, enter <tomcat server public-ip>:9090/sample

The screenshot shows a browser window displaying the 'Sample "Hello, World" Application'. The page features a cartoon cat icon and text explaining it's a sample application. Below the text, there are links to a JSP page and a servlet. The browser taskbar shows the public IP address 34.205.156.125:9090/sample.

Successfully Auto-deployed the Sample.war file to Tomcat Application using Jenkins.

