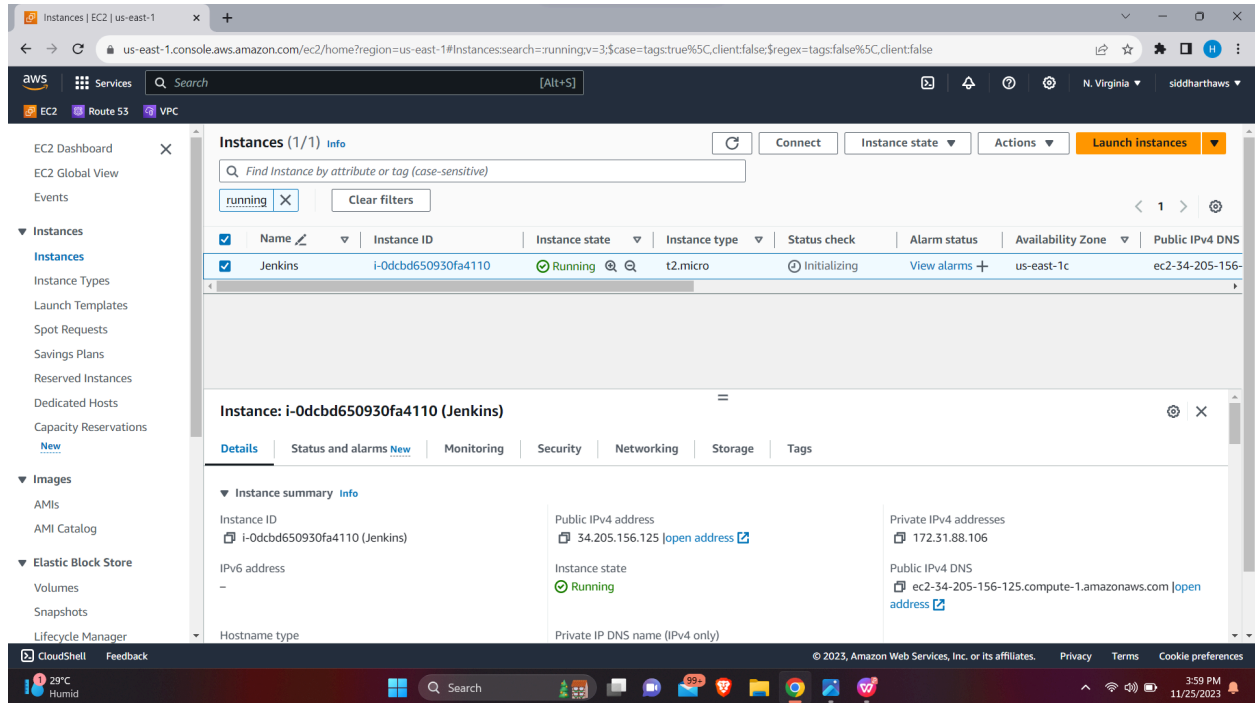


AUTO DEPLOYMENT

STEPS TO DEPLOY A WAR FILE INTO TOMCAT SERVER USING JENKINS

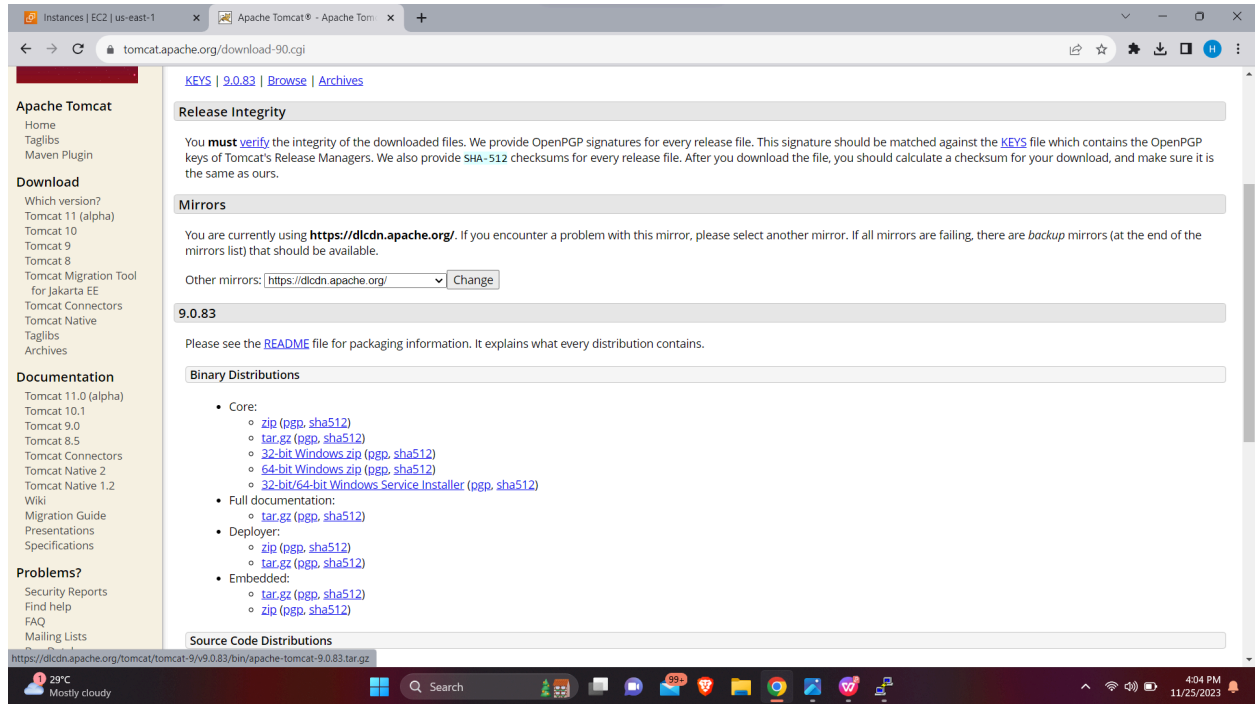
Create an ec2 instance

Install java & jenkins in it



```
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  CONTRIBUING.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)

```
root@ip-172-31-88-106: /opt/apache-tomcat-9.0.83/conf
type="org.apache.catalina.UserDatabase"
description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>

<!-- 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="150" 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="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"
  />
  -->

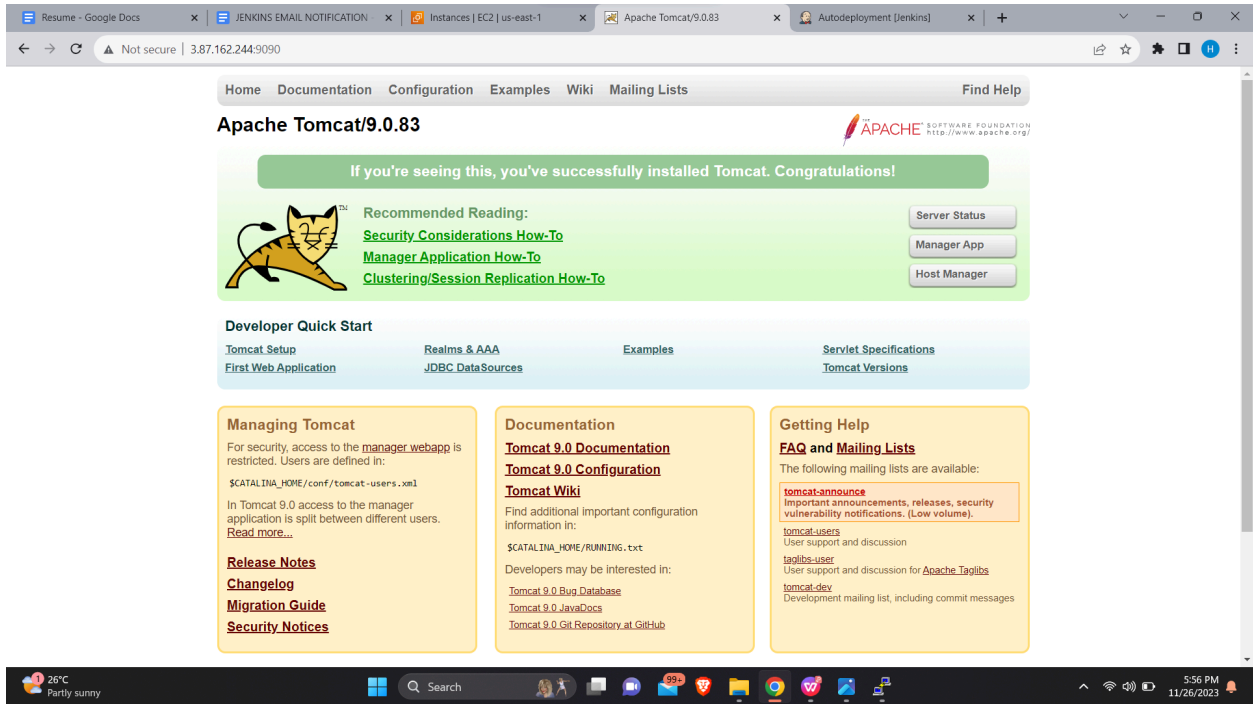
  <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation. The default
SSLImplementation will depend on the presence of the APR/native
library and the useOpenSSL attribute of the AprLifecycleListener.
Either JSSE or OpenSSL style configuration may be used regardless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

Then go the bin directory & start the tomcat service

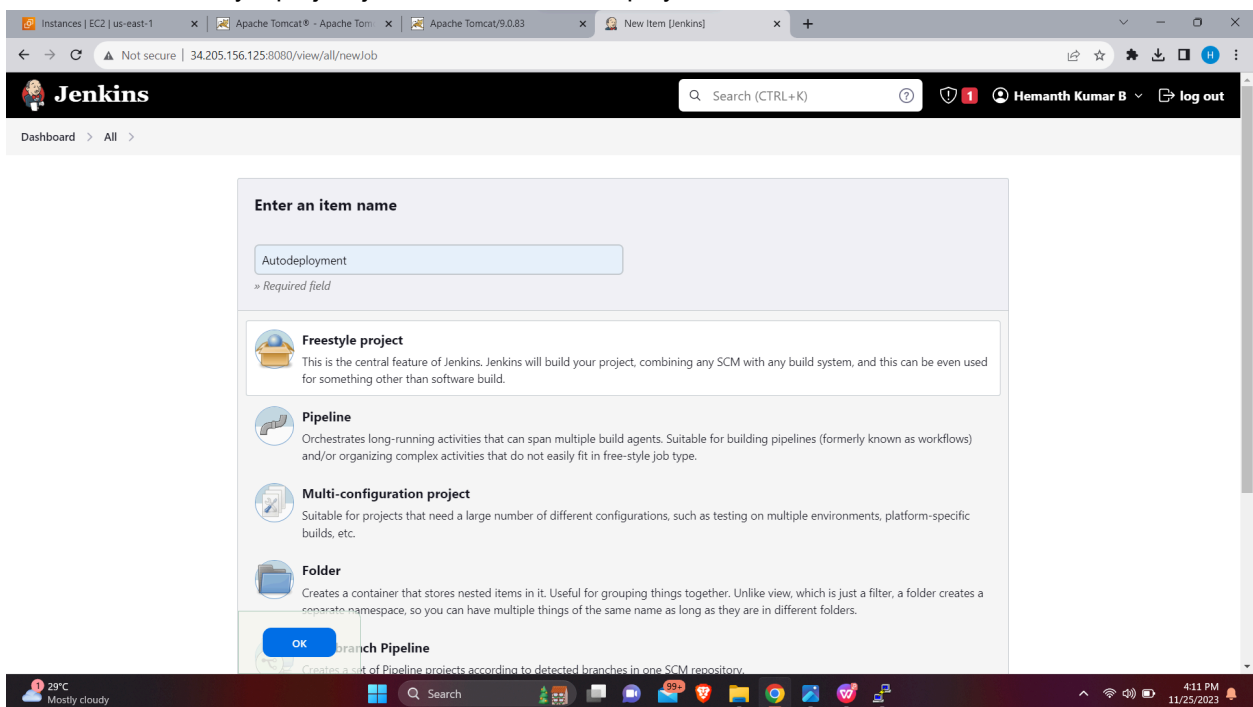
```
#cd ..
#cd bin/
#./startup.sh
```

```
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.sh    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
Tomcat started.
root@ip-172-31-88-106:/opt/apache-tomcat-9.0.83/bin#
```

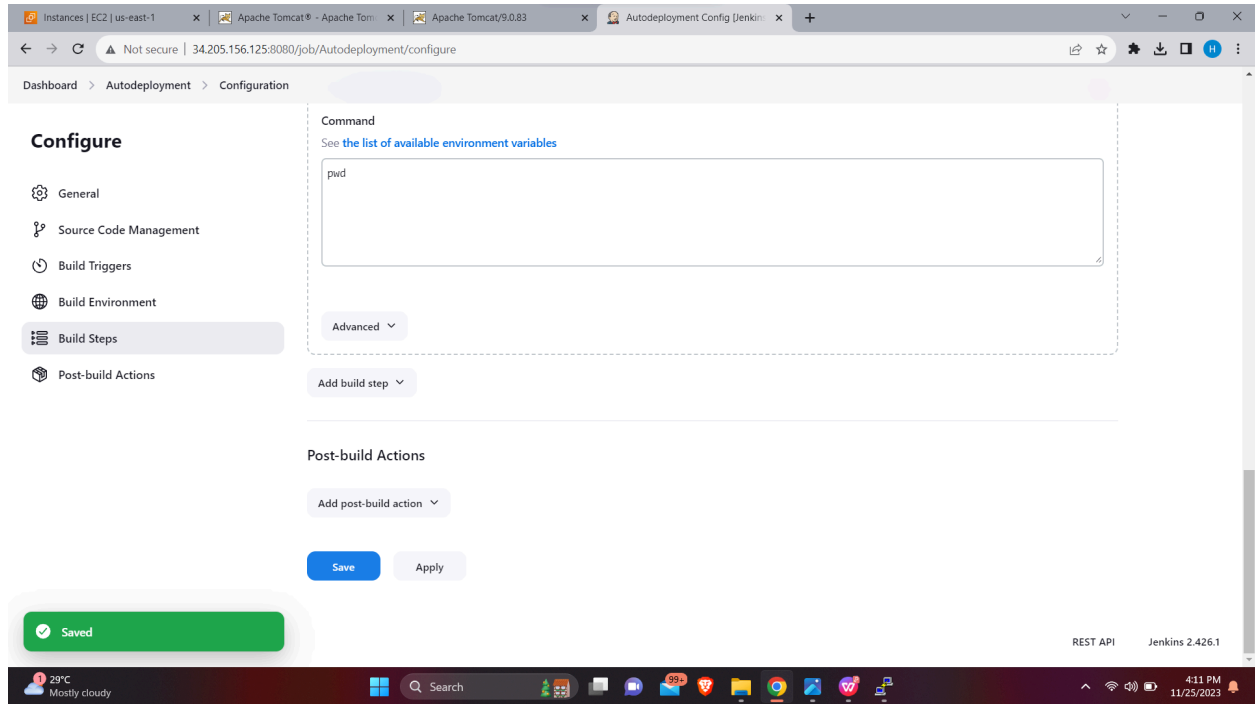
Now hit the tomcat server public along with port number in the browser. <server pub-ip>:9090



Login to the jenkins dashboard.
Create a new freestyle project job in name of AutoDeployment.

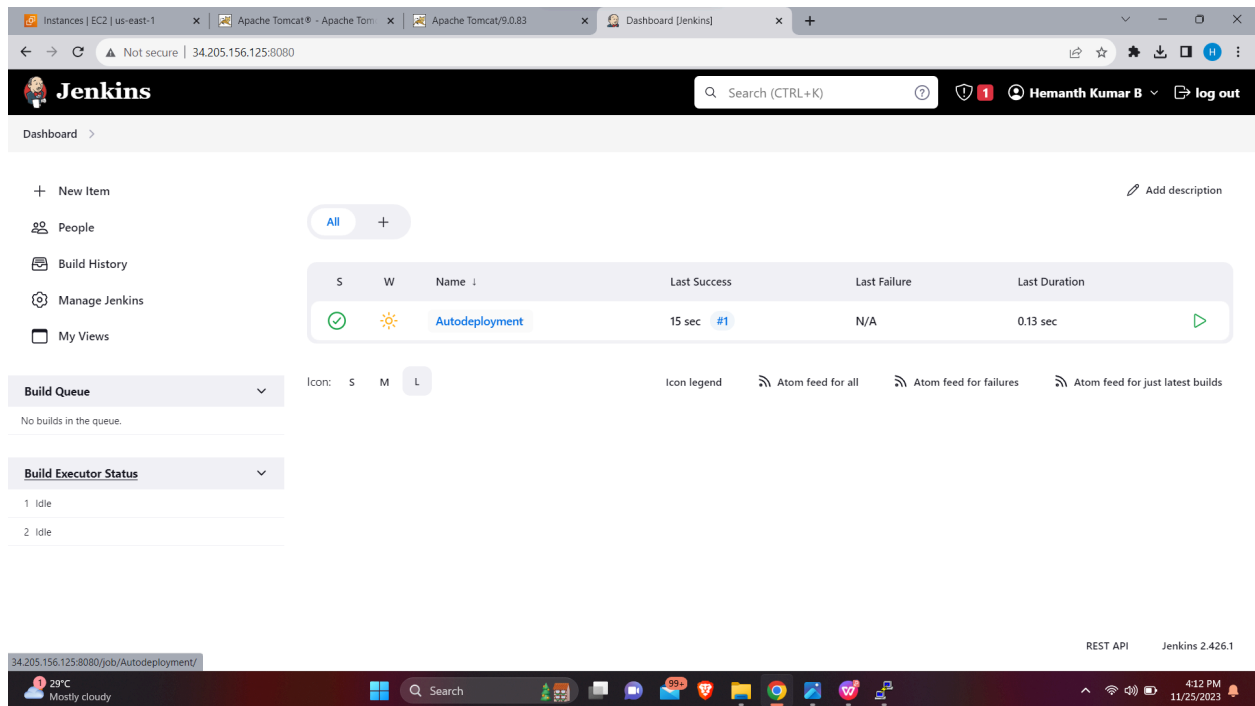


Select the created job & click configure - In build jobs - Choose execute a shell command - write a shell command (say pwd) - Then apply & save.



The screenshot shows the Jenkins 'Configure' page for the 'Autodeployment' job. The left sidebar contains a navigation menu with options: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Build Steps' section is active, showing a 'Command' field with the text 'pwd'. Below the command field is an 'Advanced' dropdown and an 'Add build step' button. The 'Post-build Actions' section is also visible, with an 'Add post-build action' button. At the bottom, there are 'Save' and 'Apply' buttons. A green 'Saved' notification is displayed at the bottom left. The browser's address bar shows the URL '34.205.156.125:8080/job/Autodeployment/configure'.

Build the Job Now - Ensure the Job deployed successfully.



The screenshot shows the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search bar, and the user name 'Hemanth Kumar B'. The left sidebar contains a navigation menu with options: New Item, People, Build History, Manage Jenkins, and My Views. The main content area displays a table of build history for the 'Autodeployment' job. The table has columns for 'S' (Status), 'W' (When), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'Autodeployment' job is listed with a status of 'Success' (green checkmark), a duration of '15 sec', and a build number of '#1'. Below the table, there is a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing '1 Idle' and '2 Idle' executors. The browser's address bar shows the URL '34.205.156.125:8080/job/Autodeployment/'.

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/
```

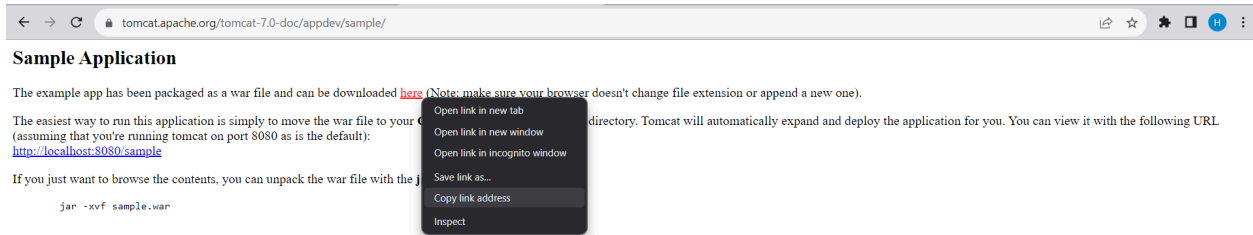
```
#ls
```

```

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 link in autodeployment directory.

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

```
#ll
```

```

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#

```

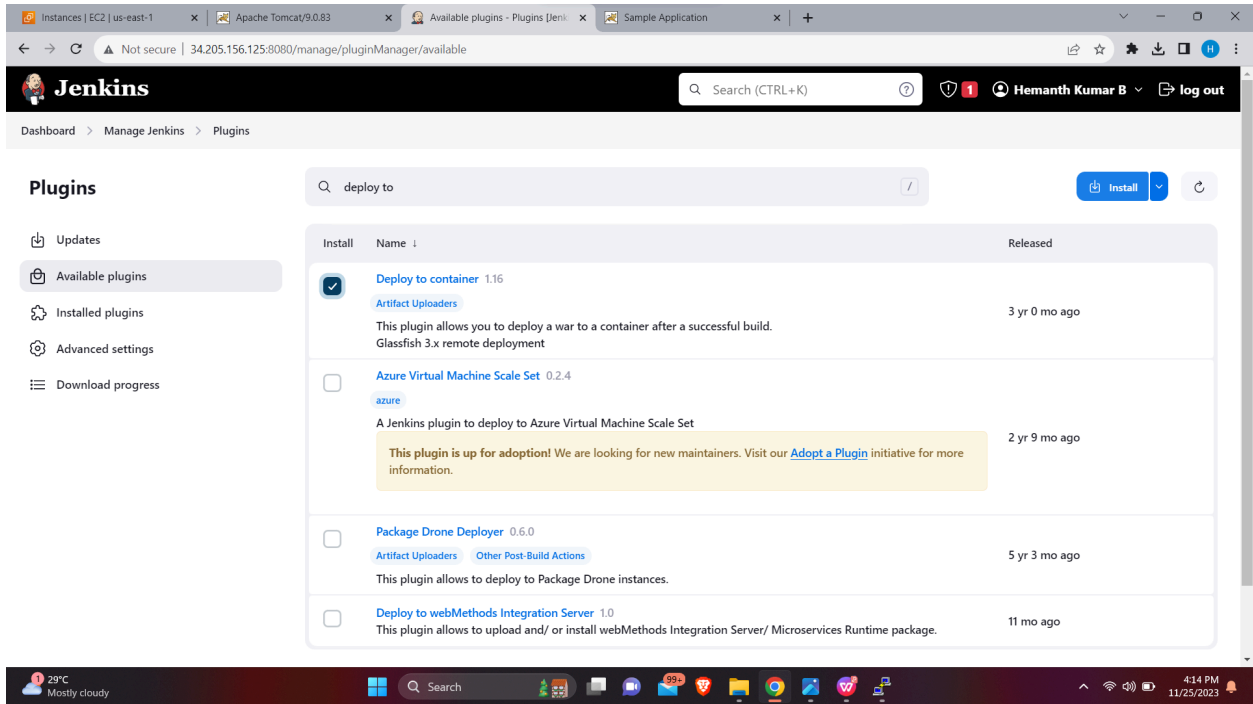
```

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

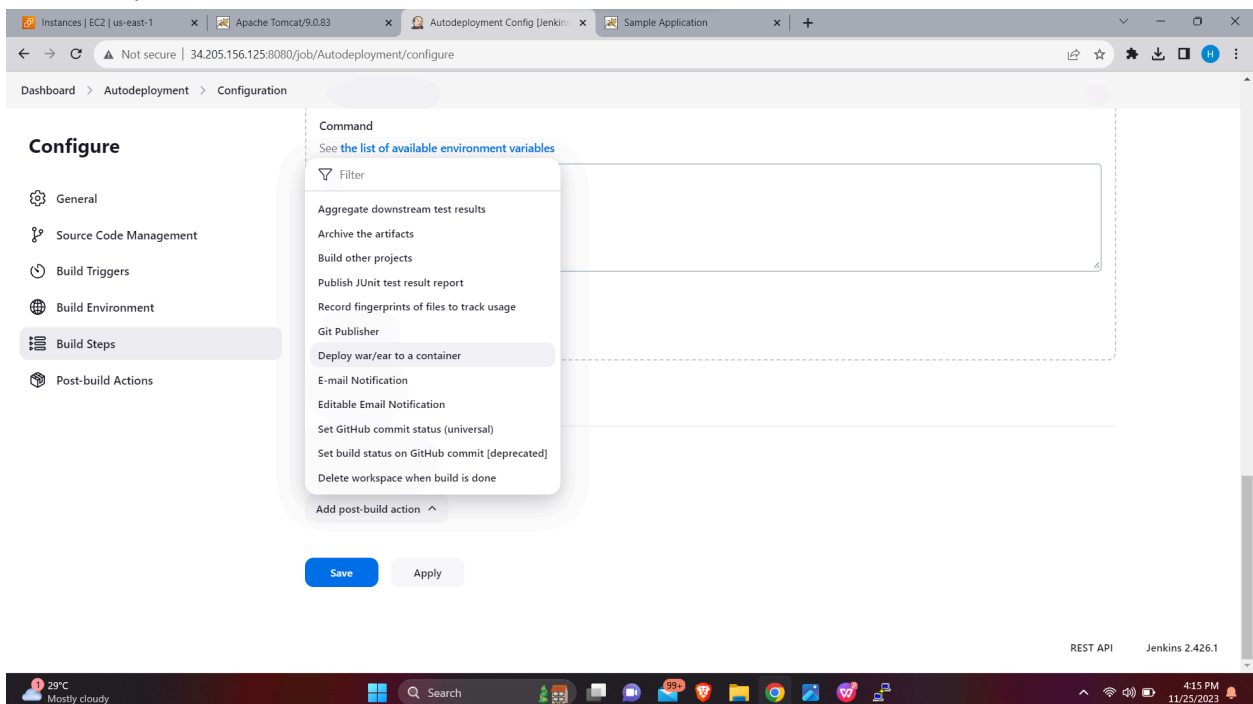
```

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.



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.



In WAR/EAR files:

Enter (Sample.war)

In context path:

Enter /sample

Containers:

Choose Tomcat version

Ex: Tomcat 9.x Remote

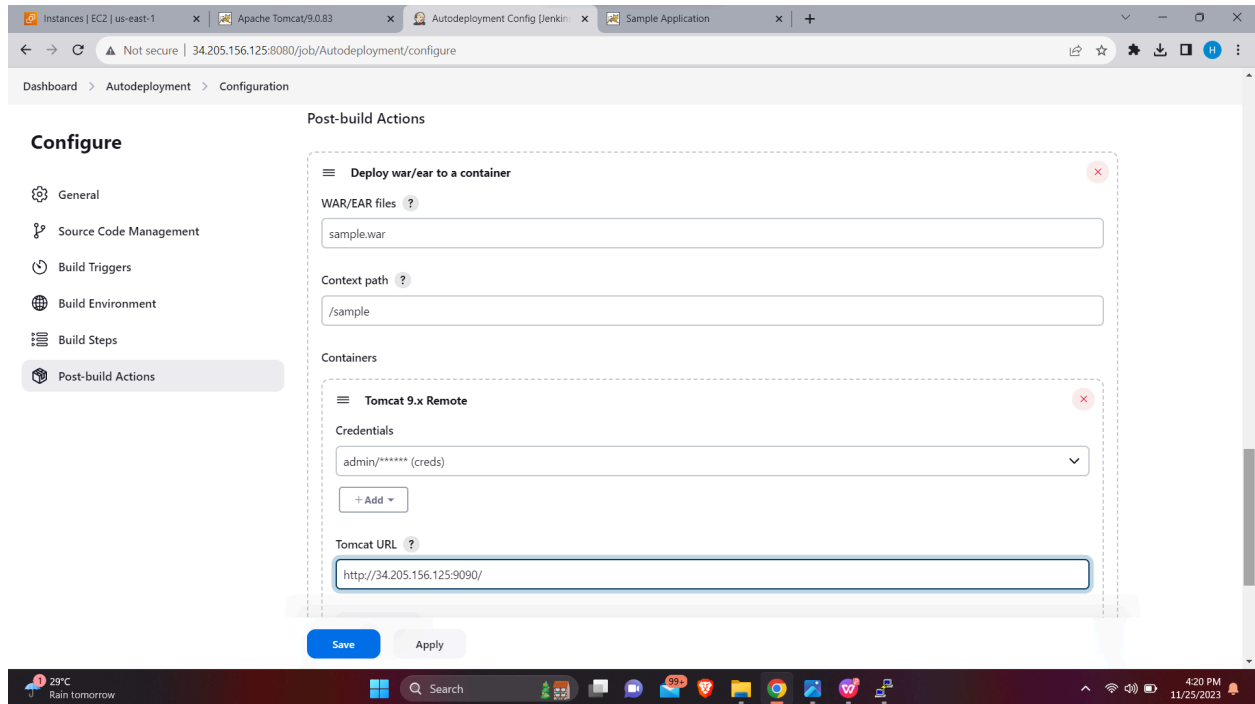
Credentials:

Add jenkins

Tomcat URL

Enter tomcat test page url

Leave the rest as default.



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
<?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.
-->
<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"/>
</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 --
26,1 Top
```

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/ja
va.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
Using CATALINA_OPTS:
Tomcat started.
```

Now run the job in Jenkins dashboard.

The screenshot shows the Jenkins Dashboard for a job named 'Autodeployment'. The job is in a failed state, indicated by a red 'X' icon. The table below shows the job's history:

S	W	Name	Last Success	Last Failure	Last Duration
⊗	☁	Autodeployment	12 min #1	2.8 sec #2	0.13 sec

The 'Build Queue' section shows 'No builds in the queue.' The 'Build Executor Status' section shows '1 idle' and '2 Autodeployment #2' (in a failed state).

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

The screenshot shows the console output for the 'Autodeployment #2' job. The output is as follows:

```
[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$AbstractBuildExecution.performAllBuildSteps(AbstractBuild.java:767)
    at hudson.model.Build$BuildExecution.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
```

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.xml
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.

```
root@ip-172-31-88-106: /opt/apache-tomcat-9.0.83/webapps/manager/META-INF
<?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="172\\.31\\.88\\.106|:::1|0:0:0:0:0:0:0:1" />
    <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.?(Boolean|Integer|Long|Number|String)|org\\.apache\\.catalina\\.filters\\.CsrftPreventionFilter|SruCache|?(\\$1)?|java\\.util\\.?(?!(Linked)?HashMap"/>
  -->
</Context>

:Wq!
```

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.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# ./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
va.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
Using CATALINA_OPTS:
Tomcat started.
```

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

The screenshot shows the Jenkins Dashboard interface. At the top, there's a search bar and user information for 'Hemanth Kumar B'. The left sidebar contains navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area displays a 'Weather report showing aggregated status of recent builds' table.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	Autodeployment	14 sec #3	9 min 43 sec #2	0.41 sec

Below the table, there are icons for 'S', 'M', and 'L'. To the right, there are links for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. On the left, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (2 idle).

Now the job is deployed successfully.

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

The screenshot shows a web browser displaying the 'Sample "Hello, World" Application' page. The page has a cat icon and a title. Below the title, it says: '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.' It then provides instructions: 'To prove that they work, you can execute either of the following links:' followed by two bullet points: 'To a JSP page.' and 'To a servlet.'

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