CONTINUOUS INTEGRATION, JENKINS

Installation of Tomcat on AWS ec2 linux & integration with Jenkins

AUGUST 13, 2018 ANUSHA SHARMA 1 COMMENT

This blog will deal with the Installation of Tomcat on AWS ec2 linux & integration with Jenkins will furthermore be demonstrated by deploying a simple Java WAR package on to the Tomcat server.

Pre-requisite:

Click on the following links to get into the details of each pre-requisite lab.

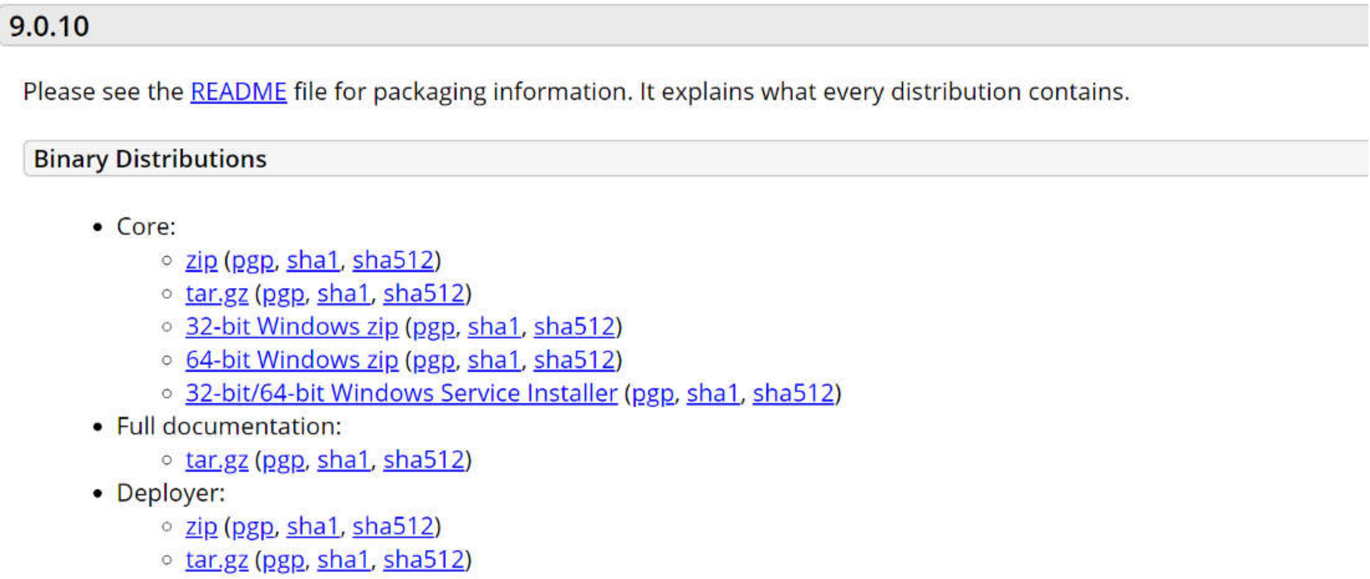
1. AWS account and ec2 Linux (Amazon Linux AMI) installed.
2. Java installation on AWS ec2 linux instance.
3. Jenkins installation on AWS ec2 linux instance.

Installation of Tomcat on AWS ec2 linux instance

Following are the step-by-step guide to install Tomcat on AWS ec2 linux instance:

Step 1: Download Tomcat package

Go to browser –> https://tomcat.apache.org/download-90.cgi (to download tomcat9) –> Copy ***tar.gz***from core section.



9.0.10

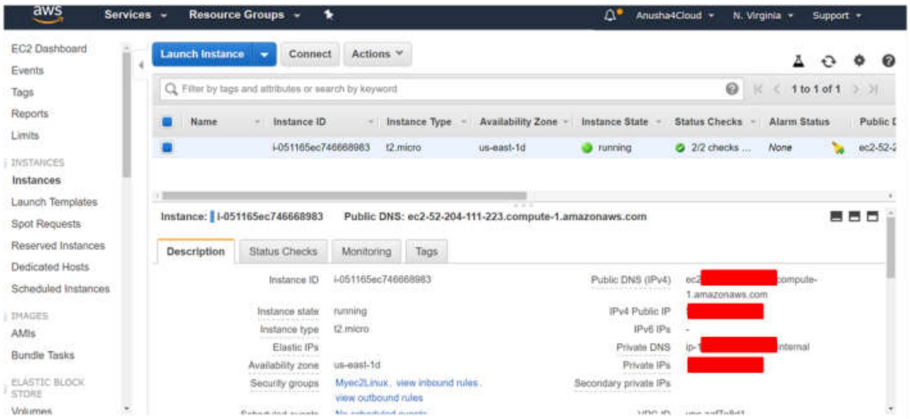
Please see the README file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
  - zip (pgp, sha1, sha512)
  - tar.gz (pgp, sha1, sha512)
  - 32-bit Windows zip (pgp, sha1, sha512)
  - 64-bit Windows zip (pgp, sha1, sha512)
  - 32-bit/64-bit Windows Service Installer (pgp, sha1, sha512)
- Full documentation:
  - tar.gz (pgp, sha1, sha512)
- Deployer:
  - zip (pgp, sha1, sha512)
  - tar.gz (pgp, sha1, sha512)

Apache Tomcat download page

Step 2: Log in to AWS ec2 instance & load putty session (for window users)



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public D
	i-051165ec746668983	t2.micro	us-east-1d	running	2/2 checks ...	None	ec2-52-2

Instance: i-051165ec746668983 Public DNS: ec2-52-204-111-223.compute-1.amazonaws.com

Instance ID i-051165ec746668983

Instance state running

Instance type t2.micro

Elastic IPs

Availability zone us-east-1d

Security groups Myec2Linux, view inbound rules, view outbound rules

Public DNS (IPv4) ec2[redacted]compute-1.amazonaws.com

IPv4 Public IP [redacted]

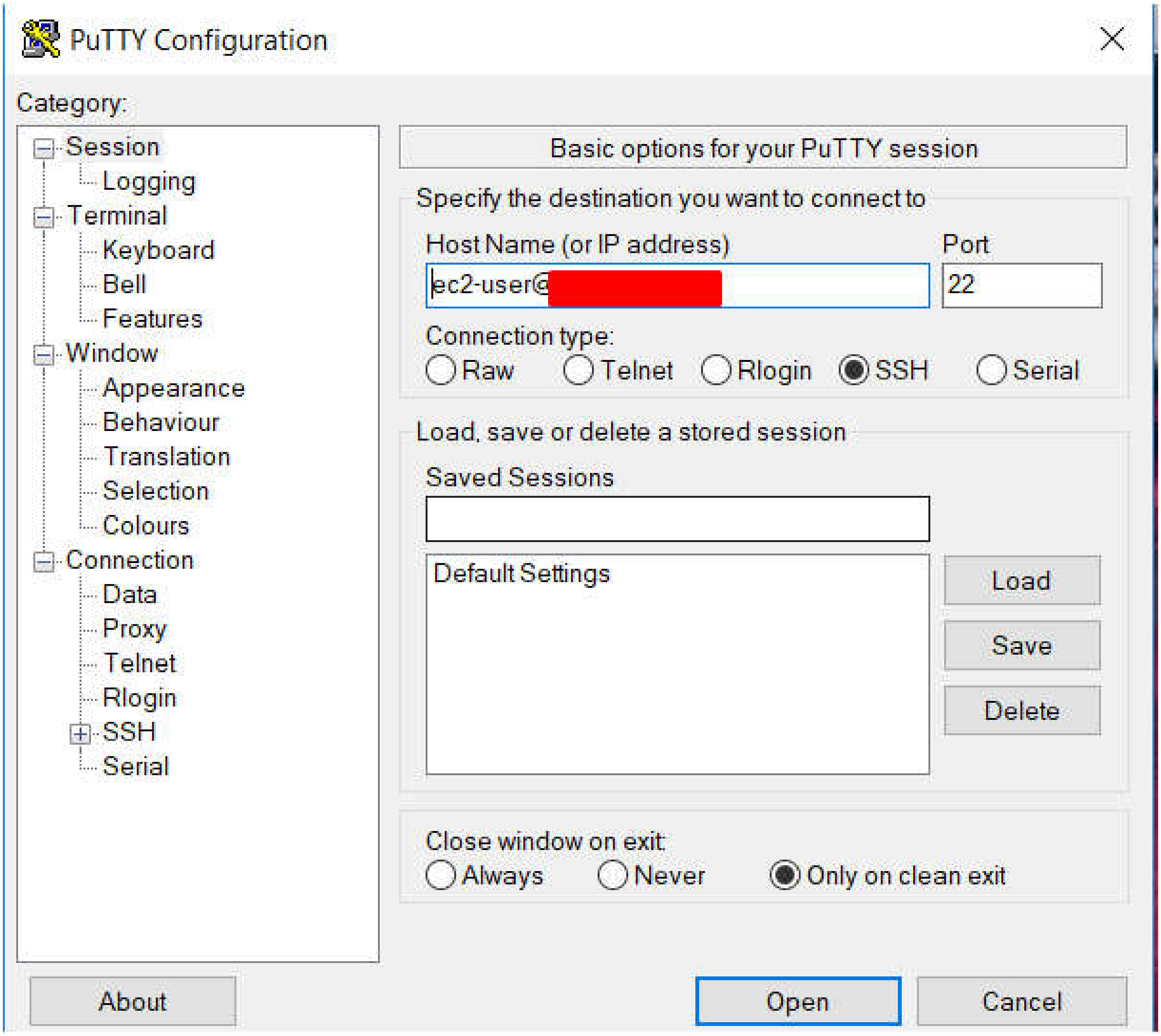
IPv6 IPs -

Private DNS ip-[redacted]nternal

Private IPs [redacted]

Secondary private IPs

AWS ec2 linux instance



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ec2-user@[redacted]

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Loading putty session

Step 3: Install & unzip the tar.gz folder

Go to the terminal & type the following commands:

```
[ec2-user@ip-xxx-xx-xx-xx] sudo -i
[root@ip-xxx-xx-xx-xx]$ wget <paste the tar.gz address copied in step1>
```

Now Tomcat has been downloaded, check with “ls” command. Further Unzip the folder:

```
[root@ip-xxx-xx-xx-xx]$ tar -zvxf apache-tomcat-9.0.10.tar.gz
```

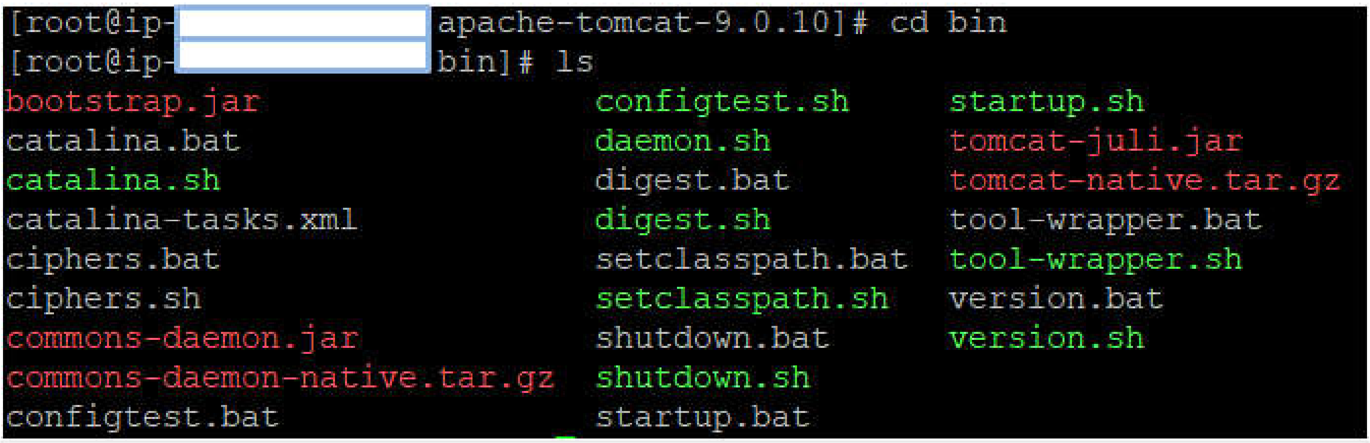
*In case wget is not installed then firstly employ this step:

```
[root@ip-xxx-xx-xx-xx]$ yum install wget -y
```

Step 4: Start Tomcat service

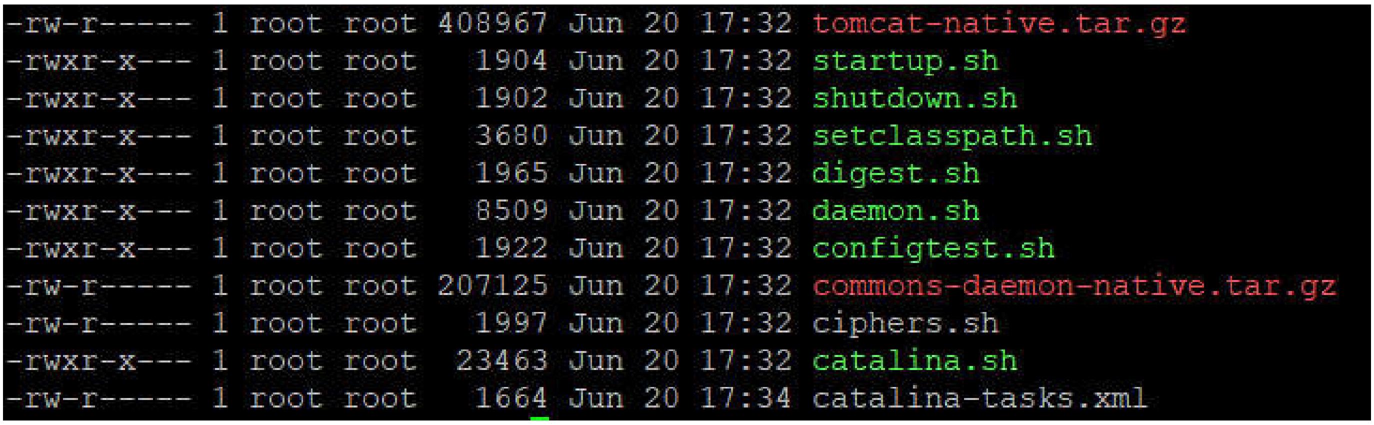
Under Apache Tomcat folder, there exists two files, namely; startup.sh and shutdown.sh

- Browse to the bin folder



```
[root@ip-[redacted] apache-tomcat-9.0.10]# cd bin
[root@ip-[redacted] bin]# ls
bootstrap.jar                 configtest.sh      startup.sh
catalina.bat                  daemon.sh          tomcat-juli.jar
catalina.sh                   digest.bat         tomcat-native.tar.gz
catalina-tasks.xml            digest.sh          tool-wrapper.bat
ciphers.bat                   setclasspath.bat   tool-wrapper.sh
ciphers.sh                    setclasspath.sh    version.bat
commons-daemon.jar            shutdown.bat       version.sh
commons-daemon-native.tar.gz  shutdown.sh
configtest.bat                startup.bat
```

```
[root@ip-xxx-xx-xx-xx bin]$ ls -ltr
//to check the status of the startup services
```



```
-rw-r----- 1 root root 408967 Jun 20 17:32 tomcat-native.tar.gz
-rwxr-x--- 1 root root   1904 Jun 20 17:32 startup.sh
-rwxr-x--- 1 root root   1902 Jun 20 17:32 shutdown.sh
-rwxr-x--- 1 root root   3680 Jun 20 17:32 setclasspath.sh
-rwxr-x--- 1 root root   1965 Jun 20 17:32 digest.sh
-rwxr-x--- 1 root root   8509 Jun 20 17:32 daemon.sh
-rwxr-x--- 1 root root   1922 Jun 20 17:32 configtest.sh
-rw-r----- 1 root root 207125 Jun 20 17:32 commons-daemon-native.tar.gz
-rw-r----- 1 root root   1997 Jun 20 17:32 ciphers.sh
-rwxr-x--- 1 root root  23463 Jun 20 17:32 catalina.sh
-rw-r----- 1 root root   1664 Jun 20 17:34 catalina-tasks.xml
```

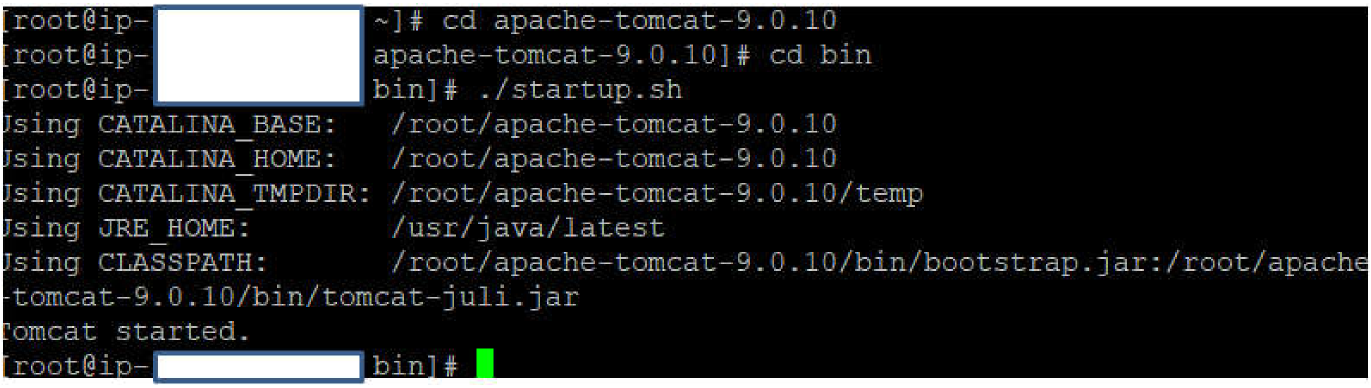
**No full permission to execute startup & shutdown services*

```
[root@ip-xxx-xx-xx-xx bin]$ chmod +x startup.sh

[root@ip-xxx-xx-xx-xx bin]$ chmod +x shutdown.sh

//For all users to execute this script
```

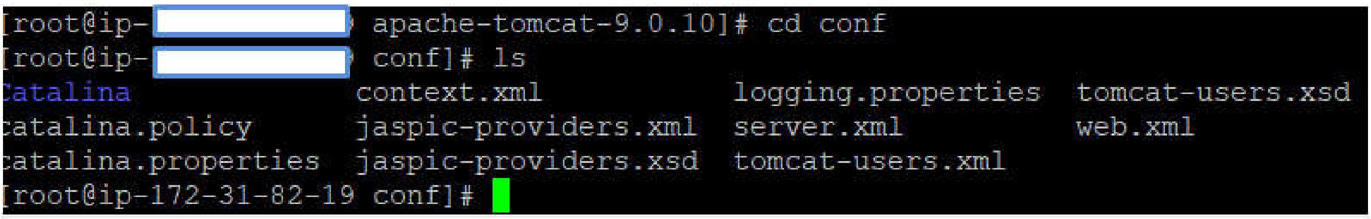
```
//Now lets start tomcat service
[root@ip-xxx-xx-xx-xx bin]$ ./startup.sh
```



```
[root@ip-[redacted] ~]# cd apache-tomcat-9.0.10
[root@ip- apache-tomcat-9.0.10]# cd bin
[root@ip- bin]# ./startup.sh
Using CATALINA_BASE:   /root/apache-tomcat-9.0.10
Using CATALINA_HOME:   /root/apache-tomcat-9.0.10
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.10/temp
Using JRE_HOME:        /usr/java/latest
Using CLASSPATH:       /root/apache-tomcat-9.0.10/bin/bootstrap.jar:/root/apache
-tomcat-9.0.10/bin/tomcat-juli.jar
Tomcat started.
[root@ip-[redacted] bin]#
```

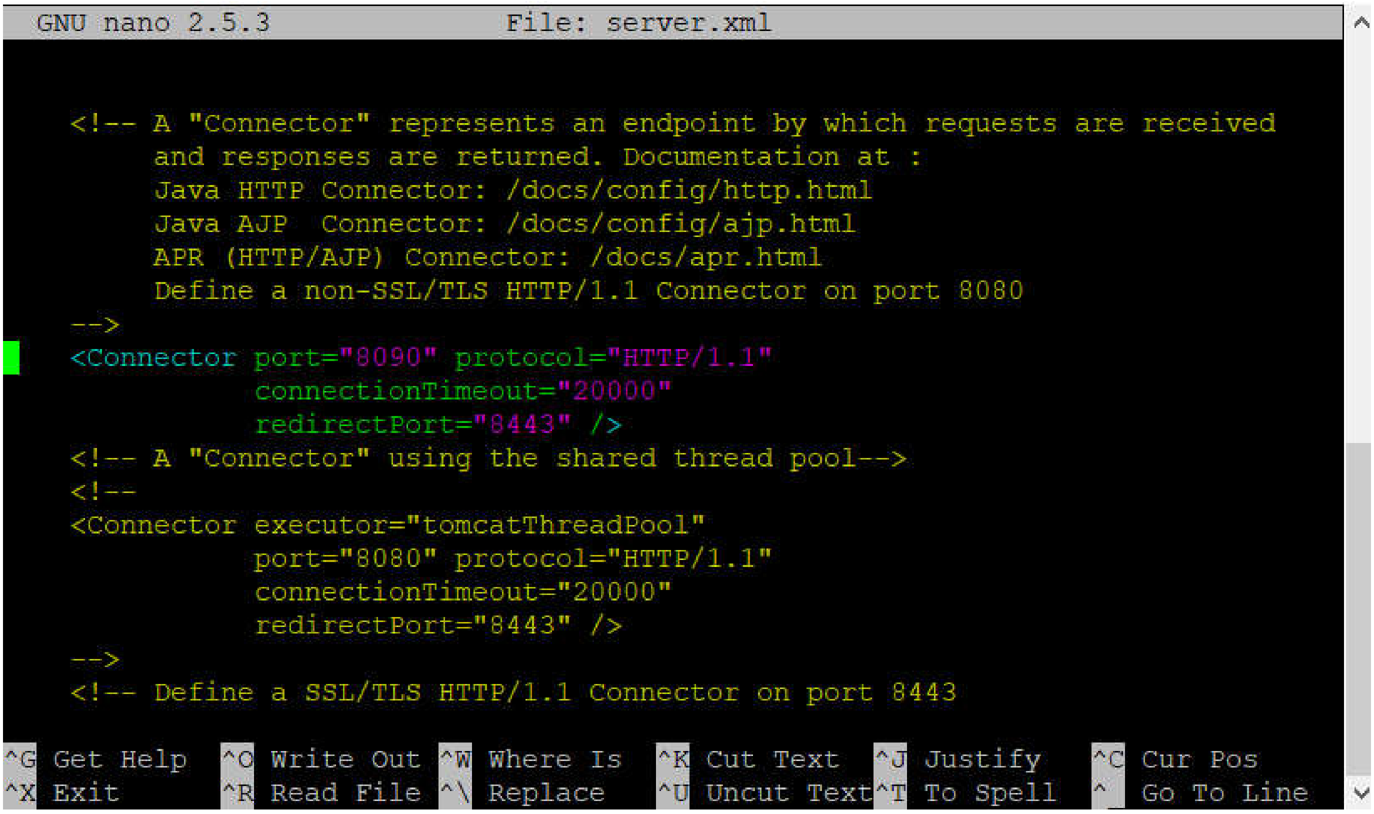
Step 5: Change port number from 8080 to 8090 (as Our Jenkins on AWS is also listening to the port 8080)

Browse to conf sub-directory under Tomcat directory and open server.xml file for editing using ‘nano’ command (vi command can also be used).



```
[root@ip-[redacted] apache-tomcat-9.0.10]# cd conf
[root@ip-[redacted] conf]# ls
Catalina             context.xml          logging.properties  tomcat-users.xsd
catalina.policy      jaspic-providers.xml  server.xml          web.xml
catalina.properties  jaspic-providers.xsd  tomcat-users.xml
[root@ip-172-31-82-19 conf]#
```

```
[root@ip-xxx-xx-xx-xx conf]$ nano server.xml
```



```
GNU nano 2.5.3              File: server.xml

    <!-- 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="8090" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

server.xml

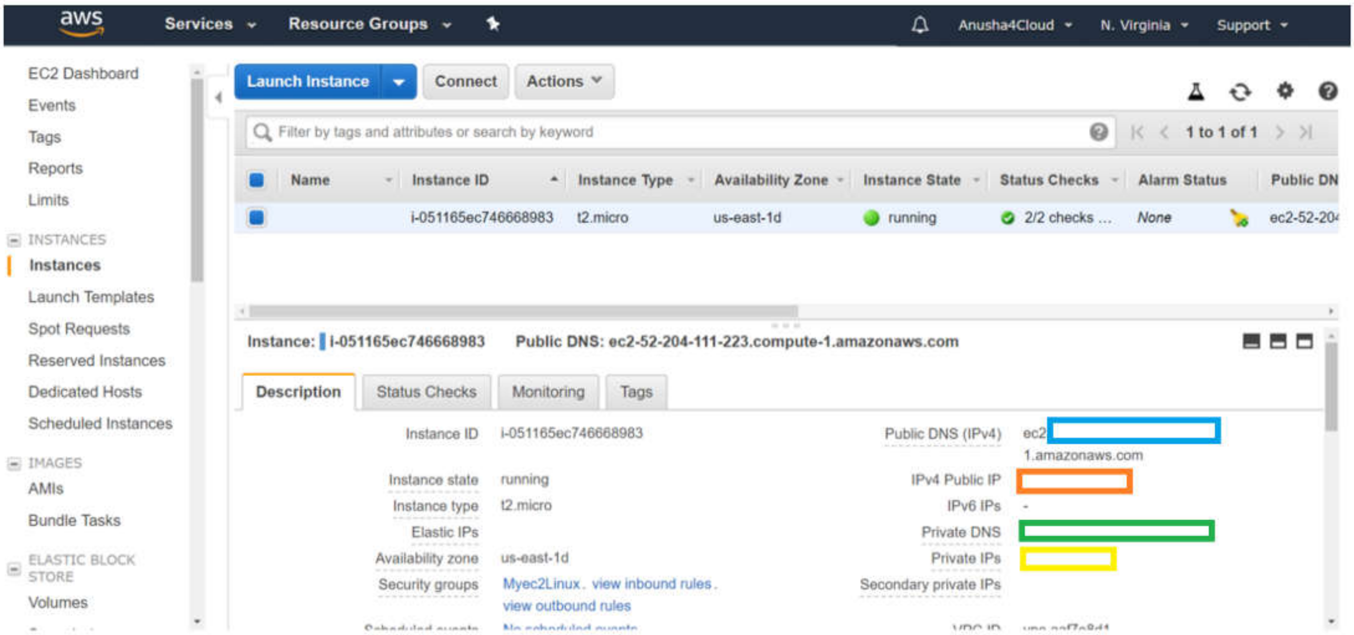
Restart the tomcat service (browse to the bin folder)

```
[root@ip-xxx-xx-xx-xx bin]$ ./shutdown.sh

[root@ip-xxx-xx-xx-xx conf]$ ./startup.sh
```

Step 7. Allow port no 8090 under security group in AWS

- Go to Your AWS account –> ec2 linux instance



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DN
	i-051165ec746668983	t2.micro	us-east-1d	running	2/2 checks ...	None	ec2-52-204

Instance: i-051165ec746668983   Public DNS: ec2-52-204-111-223.compute-1.amazonaws.com

Instance ID i-051165ec746668983

Instance state running

Instance type t2.micro

Elastic IPs

Availability zone us-east-1d

Security groups Myec2Linux. view inbound rules. view outbound rules

Scheduled events No scheduled events

Public DNS (IPv4) ec2[redacted] 1.amazonaws.com

IPv4 Public IP [redacted]

IPv6 IPs -

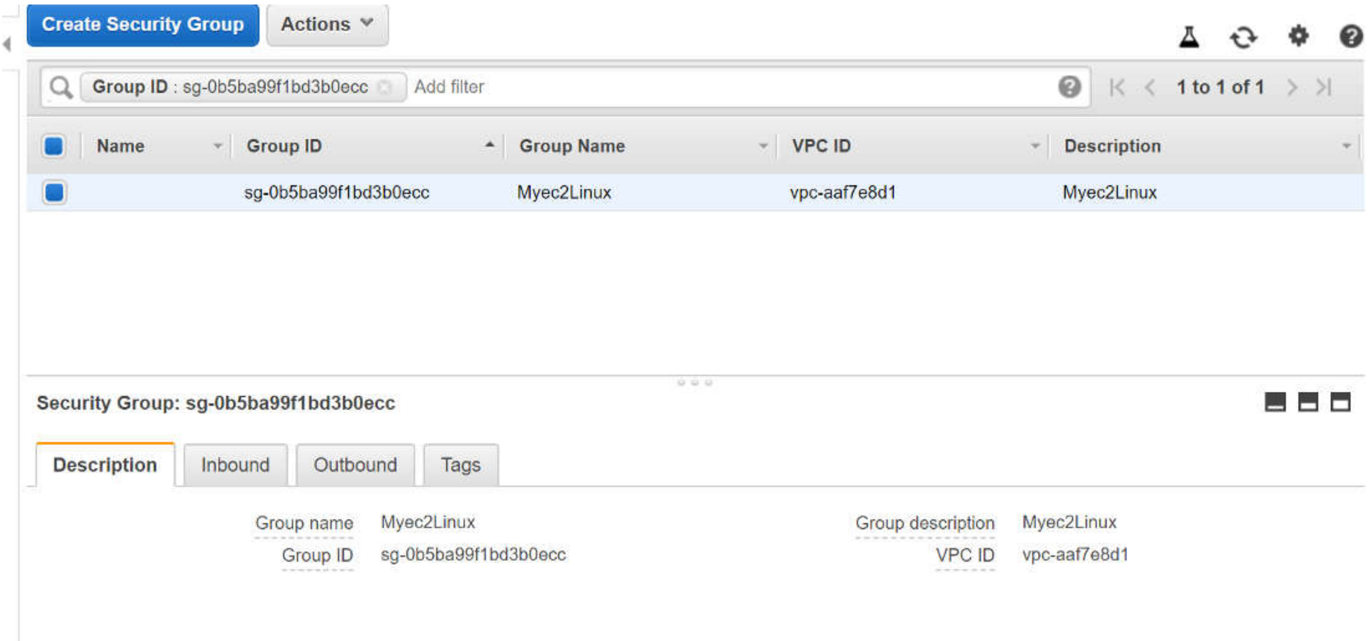
Private DNS [redacted]

Private IPs [redacted]

Secondary private IPs

VPC ID vpc-aaf7e8d1

- Got to the related security group (in this case: Myec2Linux)

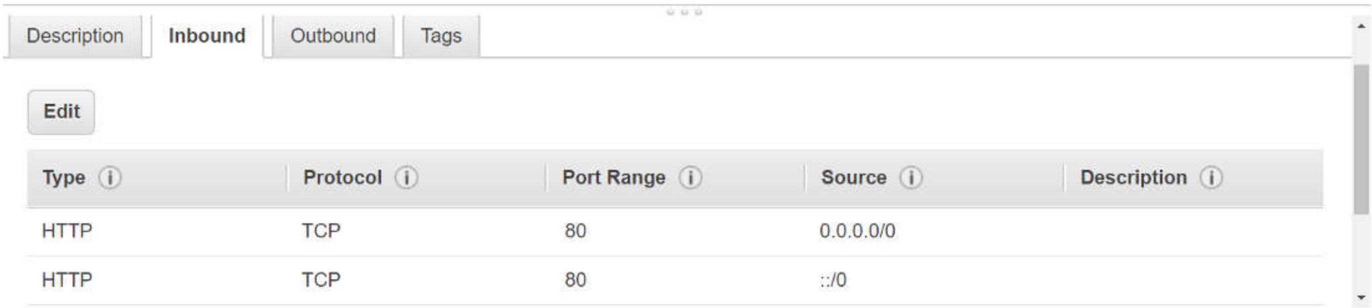


Name	Group ID	Group Name	VPC ID	Description
	sg-0b5ba99f1bd3b0ecc	Myec2Linux	vpc-aaf7e8d1	Myec2Linux

Security Group: sg-0b5ba99f1bd3b0ecc

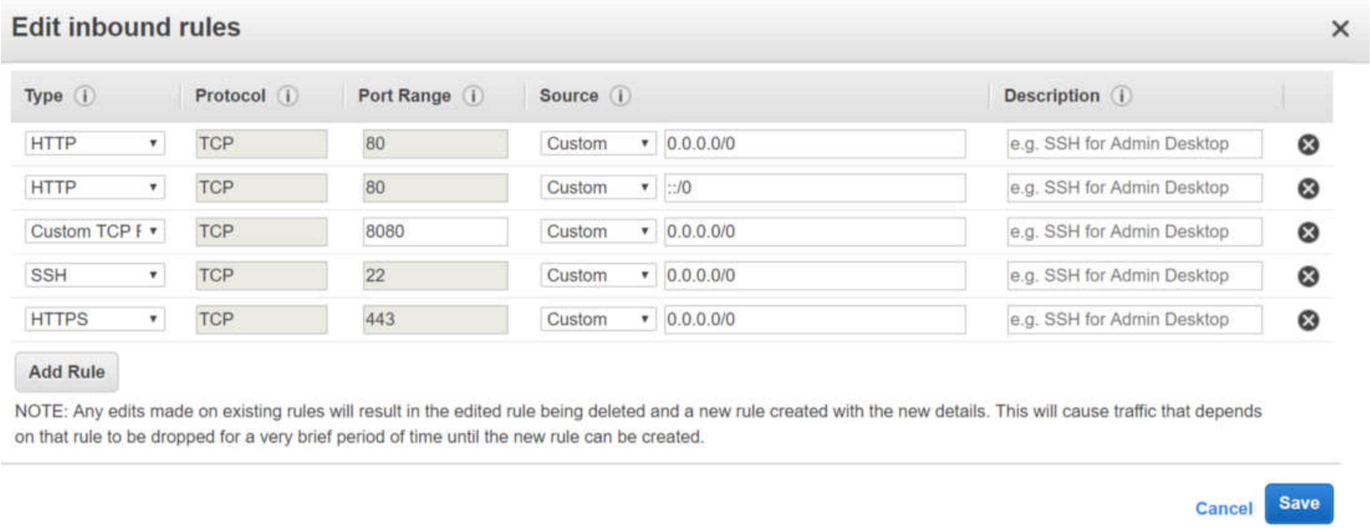
Group name Myec2Linux
Group ID sg-0b5ba99f1bd3b0ecc
Group description Myec2Linux
VPC ID vpc-aaf7e8d1

- Click Inbound Rules



Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	

- Edit Inbound rules

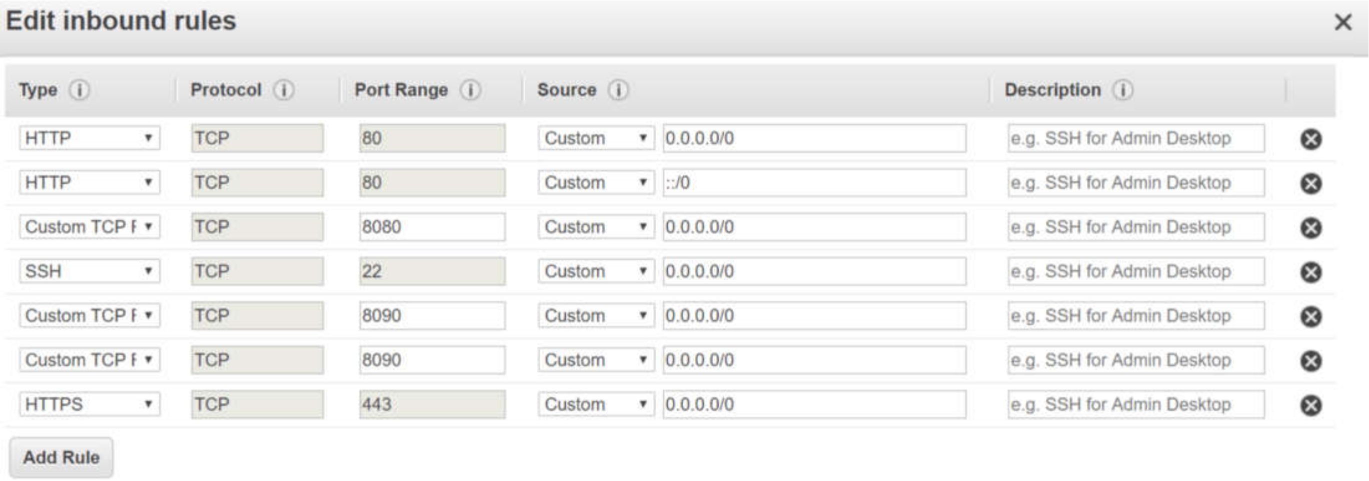


Edit inbound rules

Type	Protocol	Port Range	Source		Description
HTTP	TCP	80	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	::/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

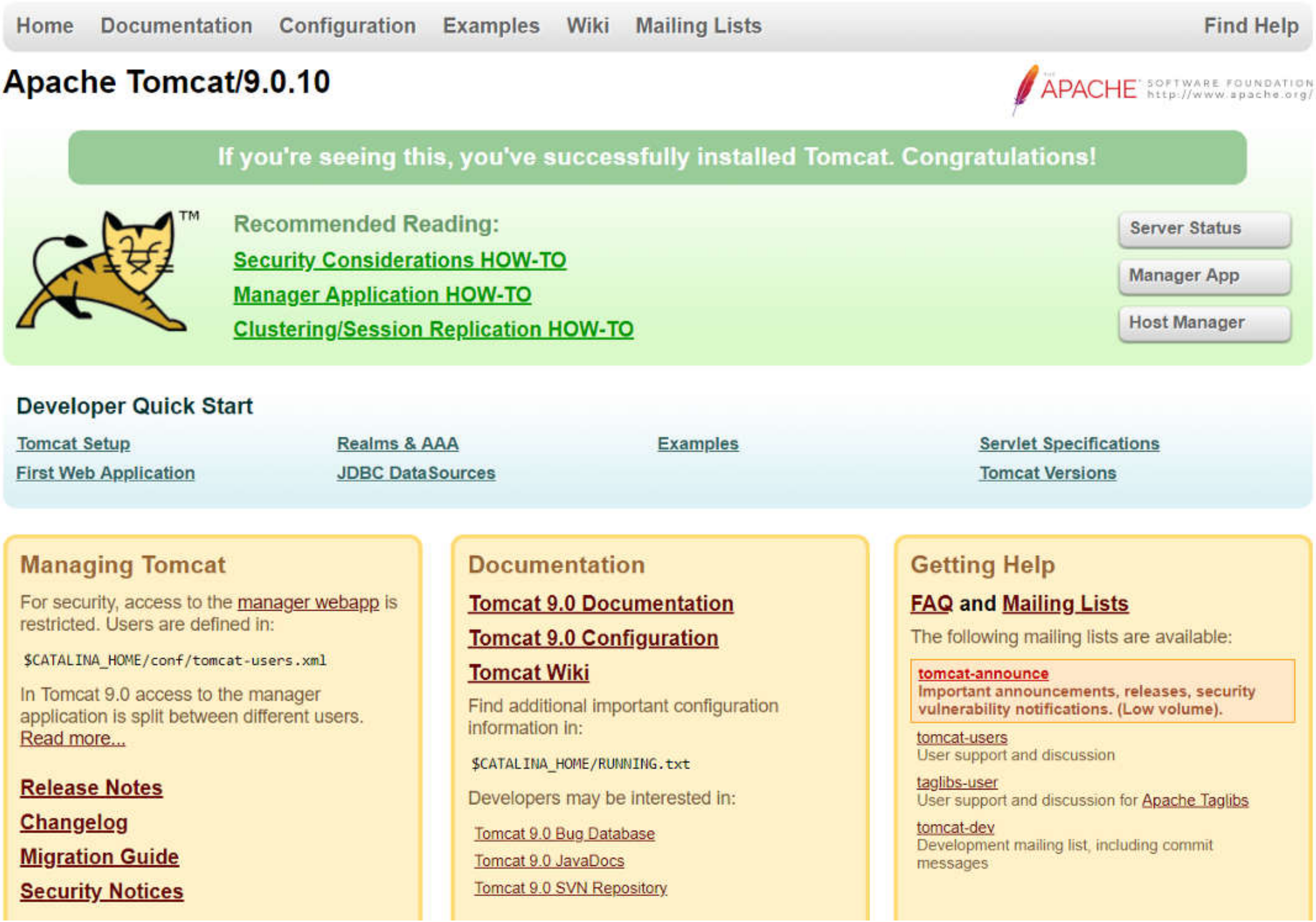
- Add 8090 port no and allow it to be public



Edit inbound rules ×

Type	Protocol	Port Range	Source		Description
HTTP	TCP	80	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	::/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8090	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8090	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

- Go to Browser and type- http://<ip_address>:8090



Apache Tomcat/9.0.10

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

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

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

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-users.xml

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

$CATALINA_HOME/RUNNING.txt

Developers may be interested in:

Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

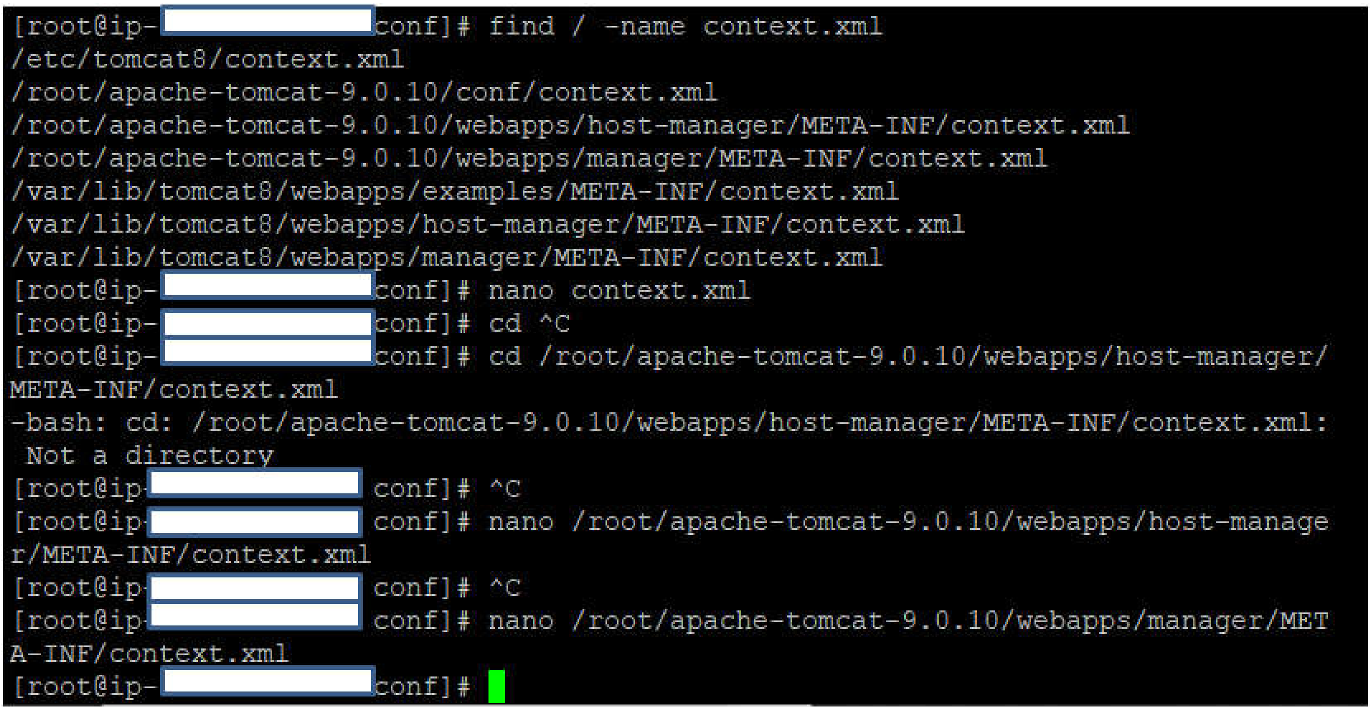
Step 8: Edit the context.xml

By default the manager is only accessible from a browser running on the same machine as Tomcat. Therefore to modify this restriction, go to context.xml file and comment out the default IP address.

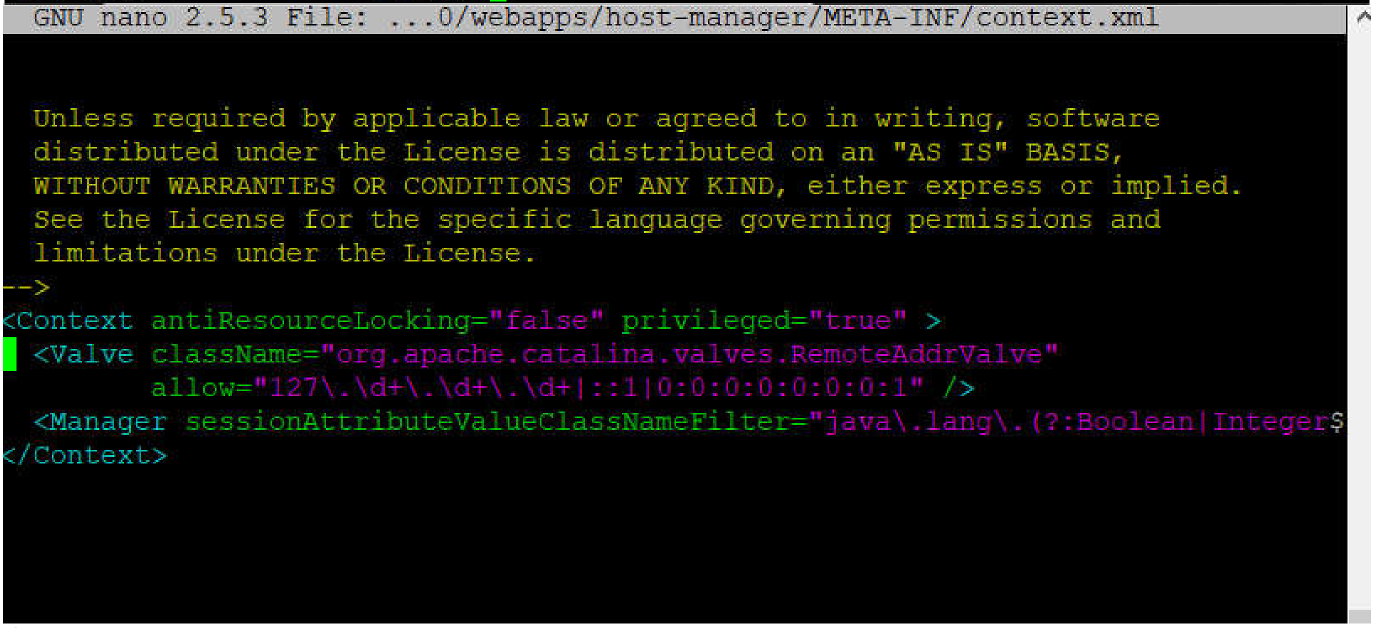
Use the ‘find’ command to find the context.xml

```
[root@ip-xxx-xx-xx-xx conf]$ find / -name context.xml
```

You will get a list, and edit context.xml within webapp, both under host-manager and manager.



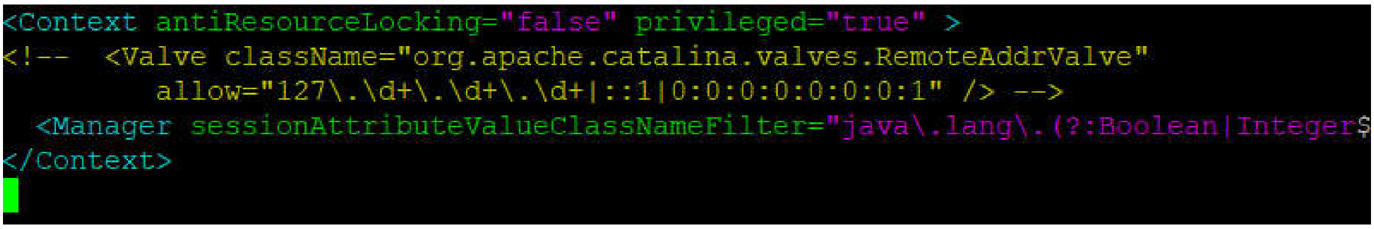
```
[root@ip-[redacted]conf]# find / -name context.xml
/etc/tomcat8/context.xml
/root/apache-tomcat-9.0.10/conf/context.xml
/root/apache-tomcat-9.0.10/webapps/host-manager/META-INF/context.xml
/root/apache-tomcat-9.0.10/webapps/manager/META-INF/context.xml
/var/lib/tomcat8/webapps/examples/META-INF/context.xml
/var/lib/tomcat8/webapps/host-manager/META-INF/context.xml
/var/lib/tomcat8/webapps/manager/META-INF/context.xml
[root@ip-[redacted]conf]# nano context.xml
[root@ip-[redacted]conf]# cd ^C
[root@ip-[redacted]conf]# cd /root/apache-tomcat-9.0.10/webapps/host-manager/
META-INF/context.xml
-bash: cd: /root/apache-tomcat-9.0.10/webapps/host-manager/META-INF/context.xml:
 Not a directory
[root@ip[redacted] conf]# ^C
[root@ip[redacted] conf]# nano /root/apache-tomcat-9.0.10/webapps/host-manage
r/META-INF/context.xml
[root@ip[redacted] conf]# ^C
[root@ip[redacted] conf]# nano /root/apache-tomcat-9.0.10/webapps/manager/MET
A-INF/context.xml
[root@ip-[redacted]conf]# 
```



```
  GNU nano 2.5.3 File: ...0/webapps/host-manager/META-INF/context.xml


  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" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer$
</Context>
```

Comment out the value section:

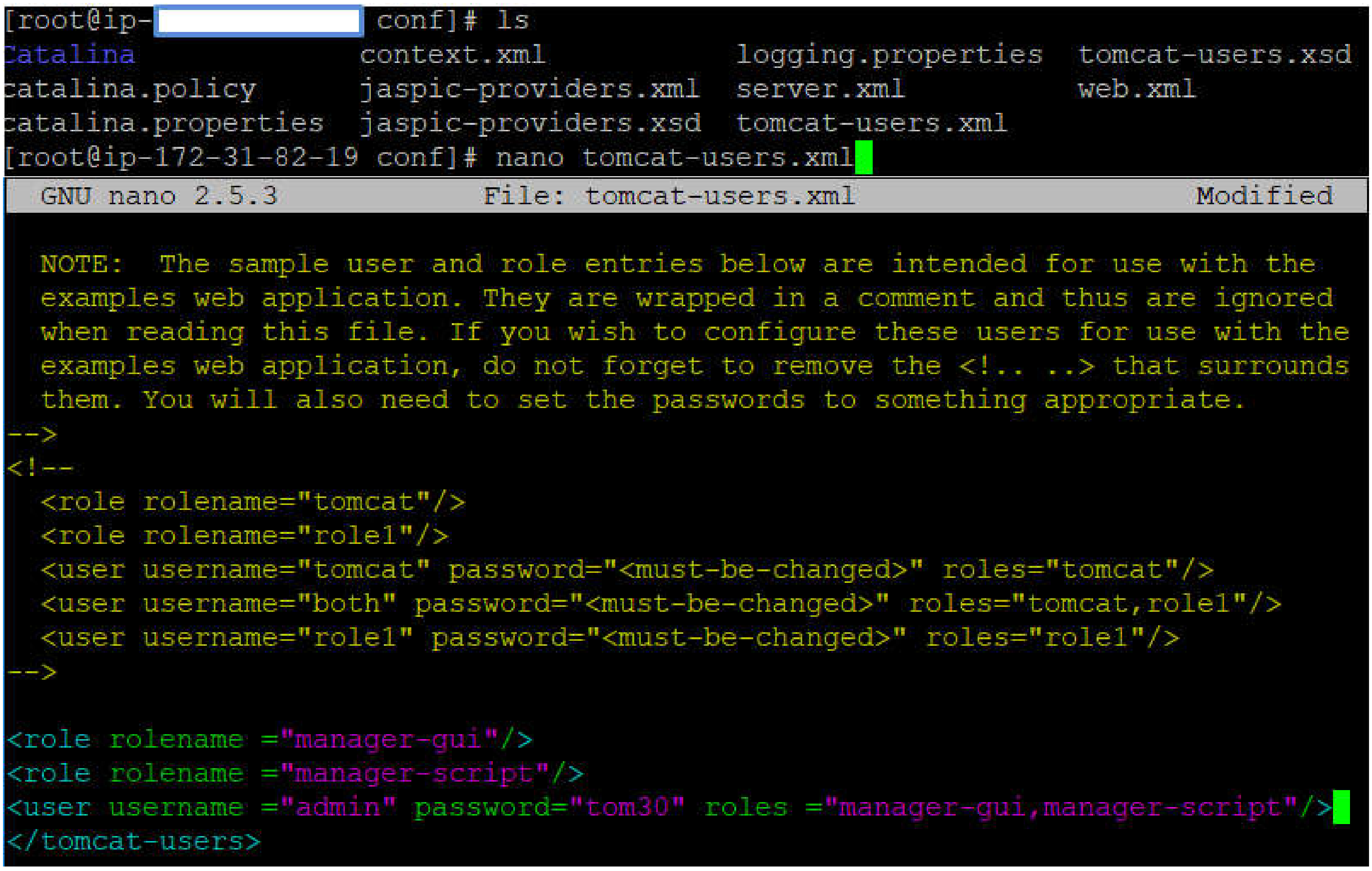


```
<Context antiResourceLocking="false" privileged="true" >
<!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer$
</Context>
```

Step 9. Specify the roles and the users:

Browse to the conf directory and open the tomcat-users.xml for editing.

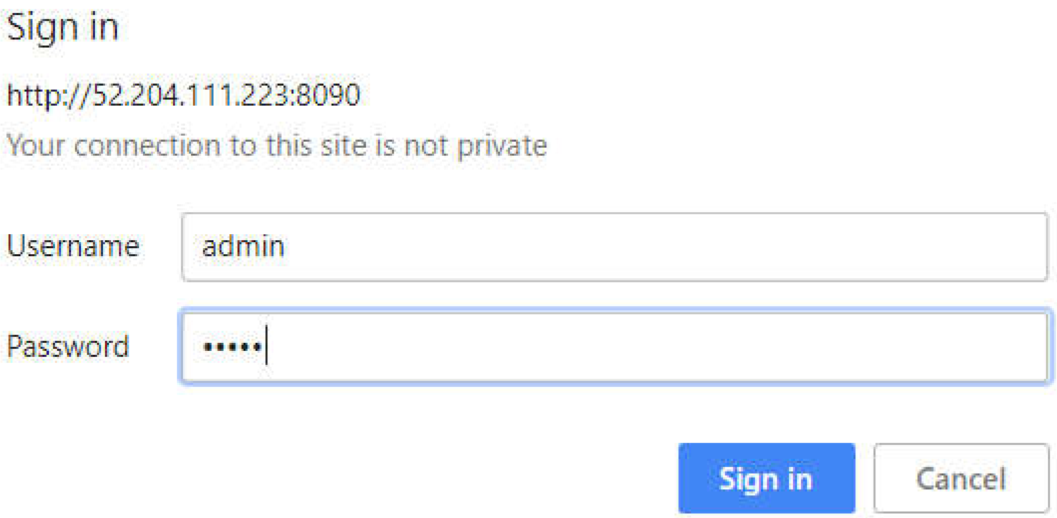
```
[root@ip-xxx-xx-xx-xx conf]$ find / -name context.xml
```



```
[root@ip-[redacted] conf]# ls
Catalina                 context.xml              logging.properties   tomcat-users.xsd
catalina.policy          jaspic-providers.xml     server.xml           web.xml
catalina.properties      jaspic-providers.xsd     tomcat-users.xml
[root@ip-172-31-82-19 conf]# nano tomcat-users.xml
  GNU nano 2.5.3               File: tomcat-users.xml                          Modified

  NOTE:  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->

<role rolename ="manager-gui"/>
<role rolename ="manager-script"/>
<user username ="admin" password="tom30" roles ="manager-gui,manager-script"/>
</tomcat-users>
```



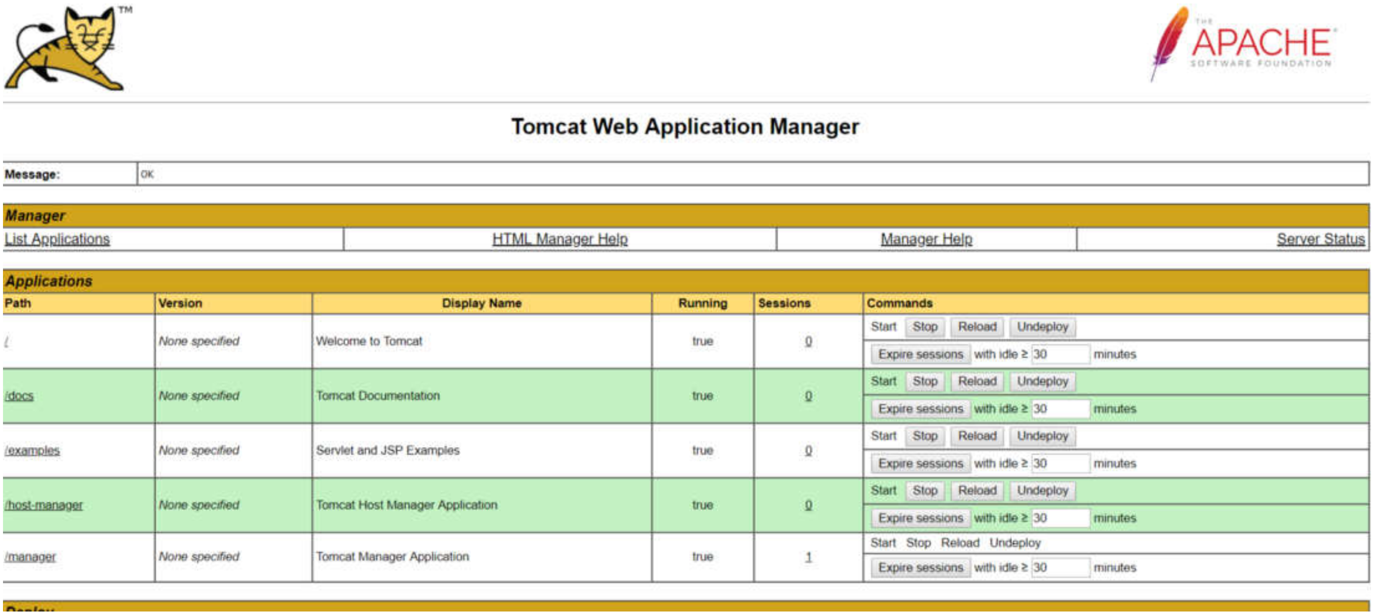
Sign in

http://52.204.111.223:8090

Your connection to this site is not private

Username: admin

Password: •••••



Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

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

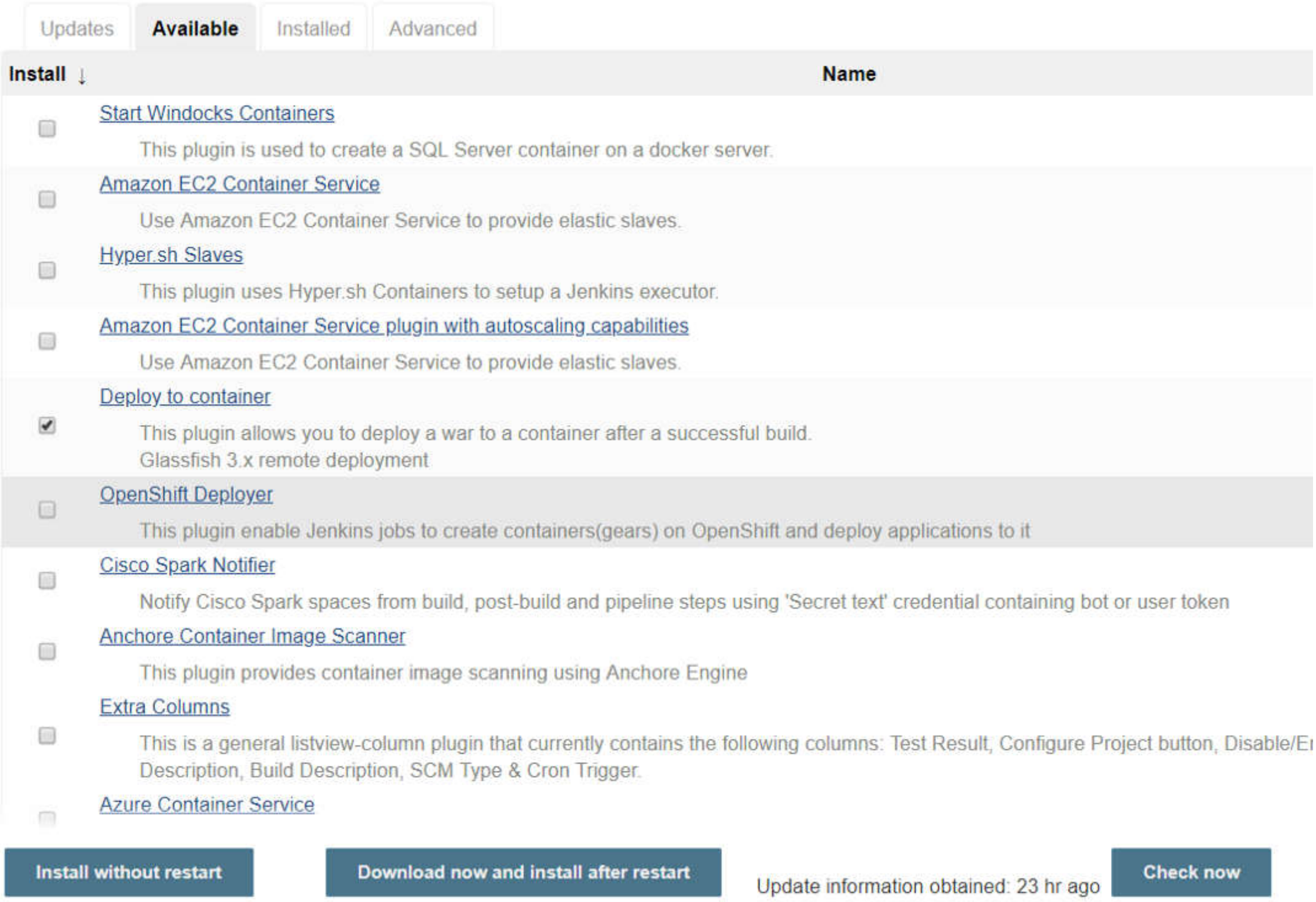
Step 10. Restart Tomcat service

Integration with Jenkins

Following are the steps to integrate with Jenkins:

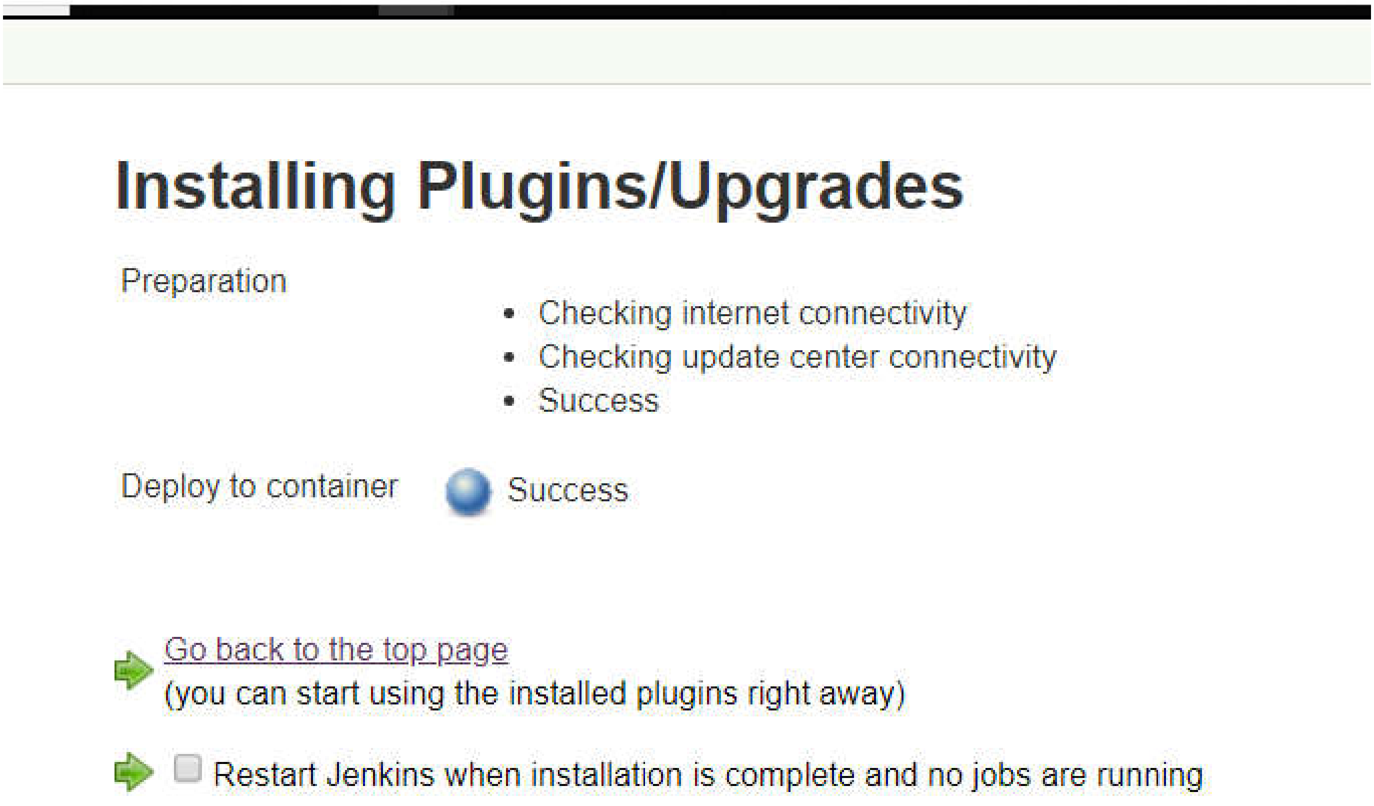
Step 1. Install Jenkins on AWS ec2 Linux (Clickhere).

Step 2. Install ‘Deploy to container’ plugin from Manage Jenkins –> Manage Plugins –> Available –> ‘Deploy to container’



Install ↓	Name
□	Start Windocks Containers This plugin is used to create a SQL Server container on a docker server.
□	Amazon EC2 Container Service Use Amazon EC2 Container Service to provide elastic slaves.
□	Hyper.sh Slaves This plugin uses Hyper.sh Containers to setup a Jenkins executor.
□	Amazon EC2 Container Service plugin with autoscaling capabilities Use Amazon EC2 Container Service to provide elastic slaves.
☑	Deploy to container This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment
□	OpenShift Deployer This plugin enable Jenkins jobs to create containers(gears) on OpenShift and deploy applications to it
□	Cisco Spark Notifier Notify Cisco Spark spaces from build, post-build and pipeline steps using 'Secret text' credential containing bot or user token
□	Anchore Container Image Scanner This plugin provides container image scanning using Anchore Engine
□	Extra Columns This is a general listview-column plugin that currently contains the following columns: Test Result, Configure Project button, Disable/E Description, Build Description, SCM Type & Cron Trigger.
□	Azure Container Service

Update information obtained: 23 hr ago



Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

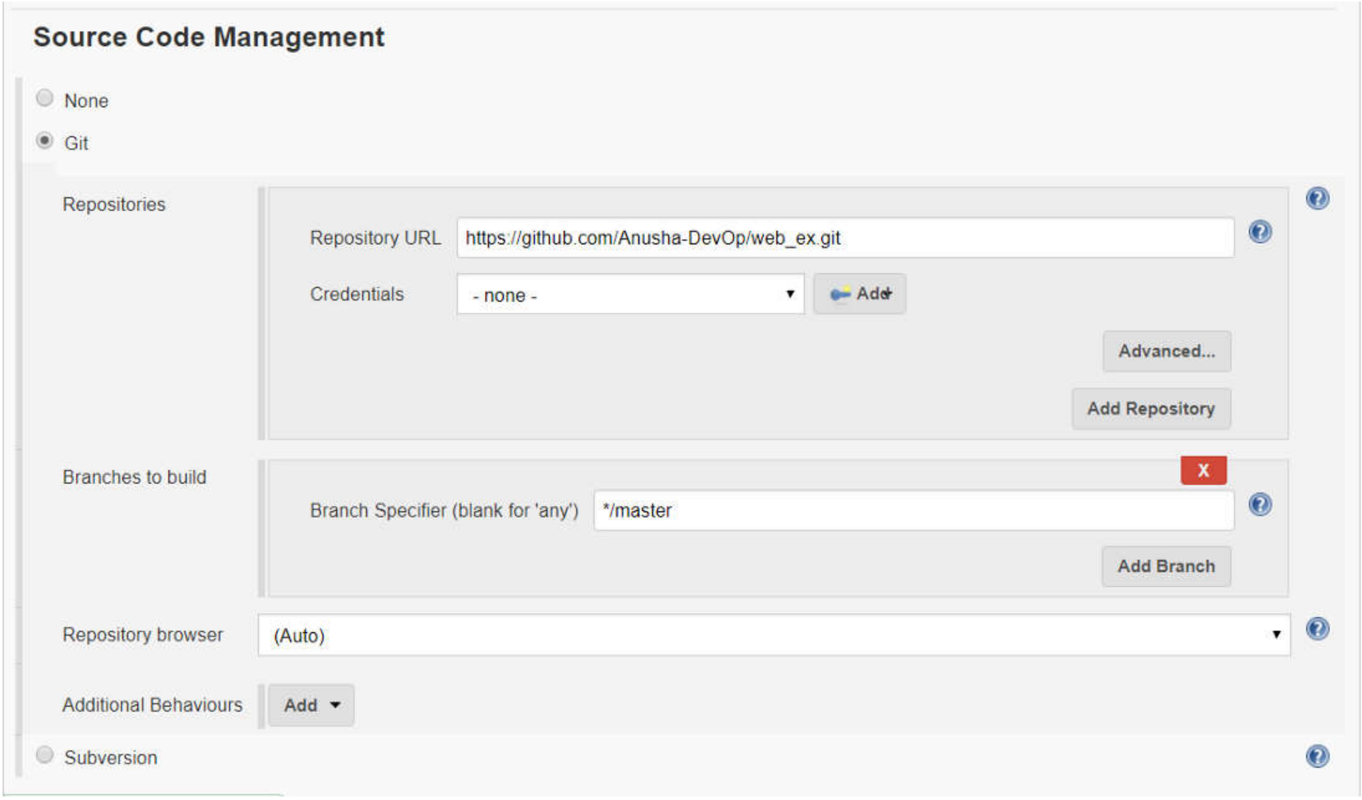
Deploy to container ● Success

Go back to the top page
(you can start using the installed plugins right away)

□ Restart Jenkins when installation is complete and no jobs are running

Step 3. Create a freestyle project

- Check out the Java code (that will create WAR) from github.



Source Code Management

○ None

● Git

Repositories

Repository URL: https://github.com/Anusha-DevOp/web_ex.git

Credentials: - none -

Branches to build

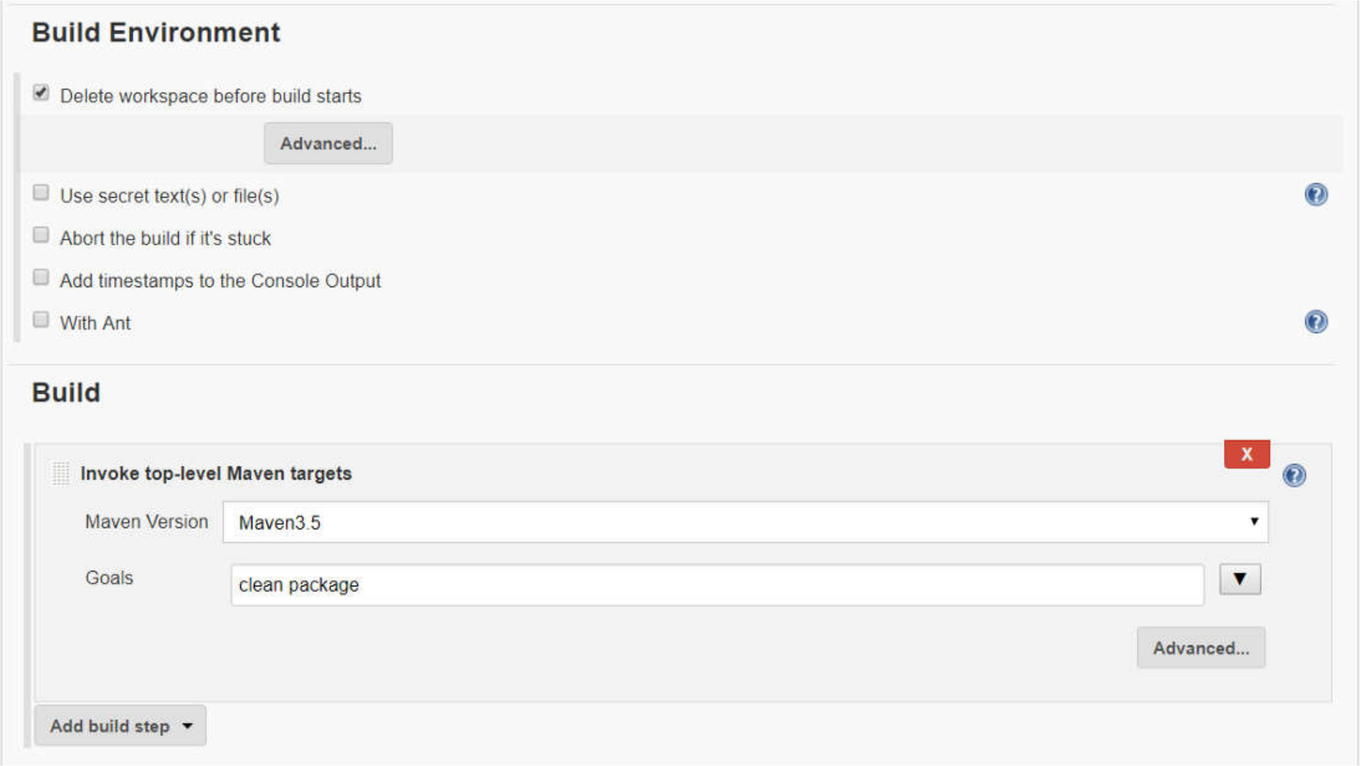
Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

○ Subversion

- Invoke top-level Maven target



Build Environment

☑ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ With Ant

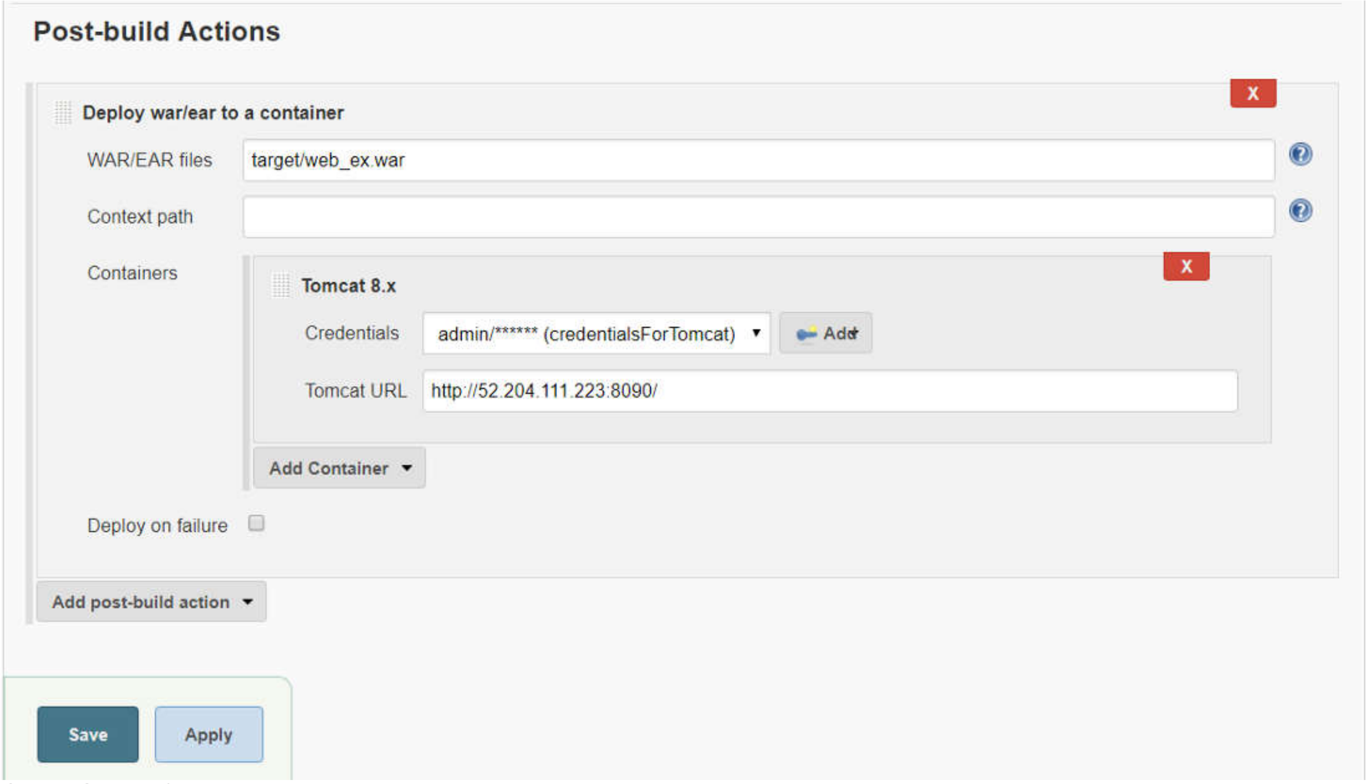
Build

Invoke top-level Maven targets

Maven Version: Maven3.5

Goals: clean package

- Add post-build action : 'Deploy war/ear to container'



Post-build Actions

Deploy war/ear to a container

WAR/EAR files: target/web_ex.war

Context path:

Containers: Tomcat 8.x

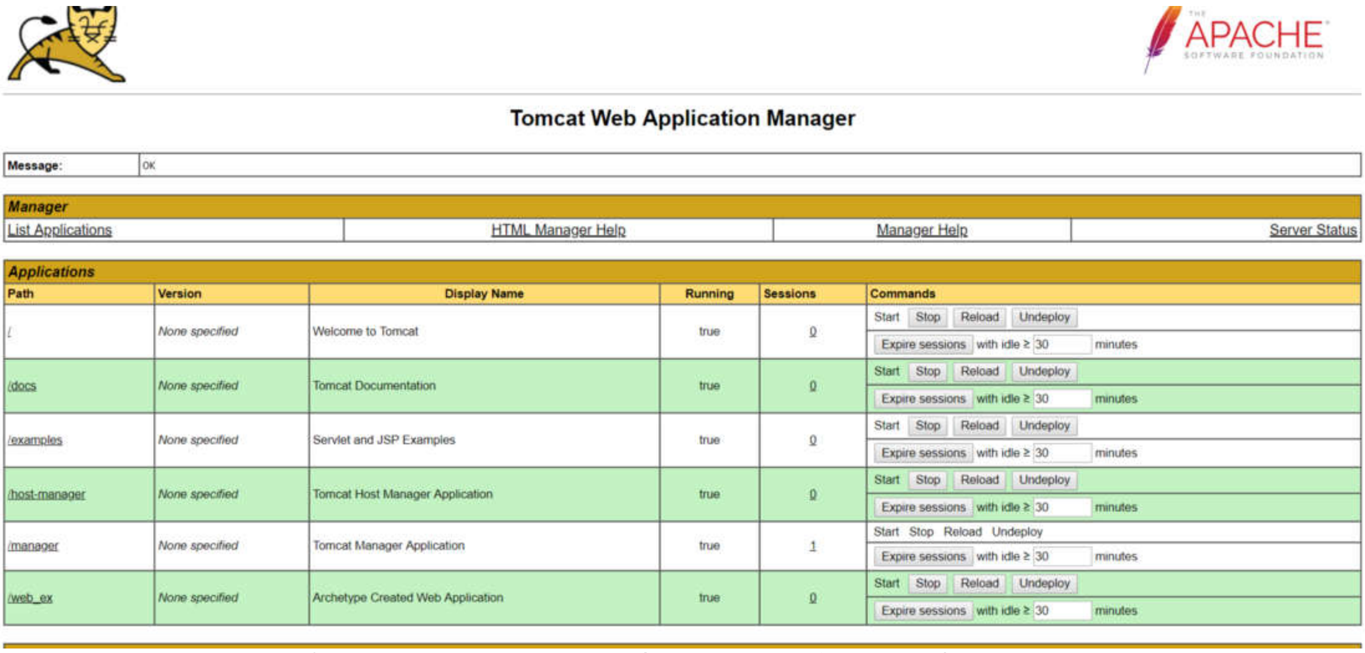
Credentials: admin/****** (credentialsForTomcat)

Tomcat URL: http://52.204.111.223:8090/

Deploy on failure ☐

Click Save & build now.

- Go to browser : http://<ip_address>:8090 and log in manager app



Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

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

You will see your war file deployed to tomcat (in this case ‘web_ex’).

Voila!! Congratulations !!

You have done a great job. All steps meticulously followed.