

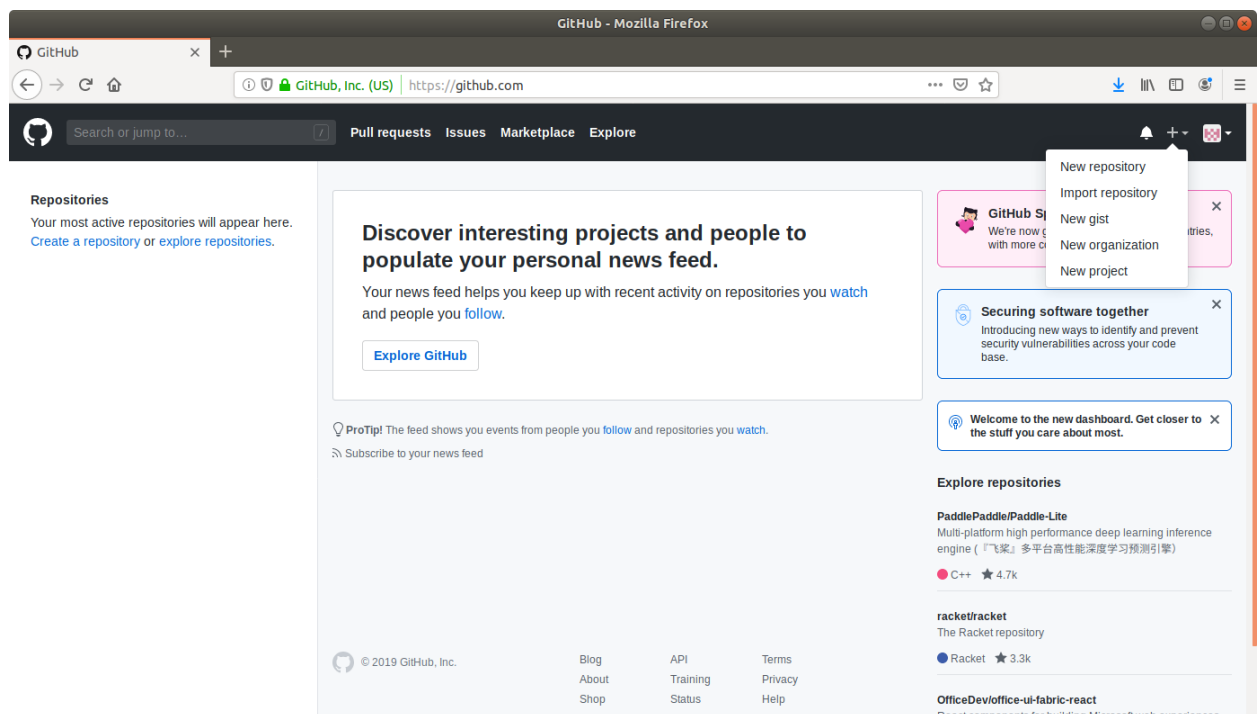
# Deploying Maven App to Tomcat Server

This section will guide you to:

- Create an EC2 instance
- Create and run a Tomcat instance on an EC2 VM
- Create a Maven webapp
- Configure a Jenkins build job to compile and deploy a Maven app to a Tomcat server

## Step 1: Creating a Git repository for the webapp

- Log in to your Github account.
- Click on the plus icon next to the profile picture and select *New repository* from the drop-down menu.





- Fill the required fields in the create repository form.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).



Owner

Repository name \*

 **judy-simplilearn** / **hooli-website** 

Great repository names are short and memorable. Need inspiration? How about **vigilant-guacamole**?

Description (optional)

- ☒  **Public**  
Anyone can see this repository. You choose who can commit.
- ☐  **Private**  
You choose who can see and commit to this repository.

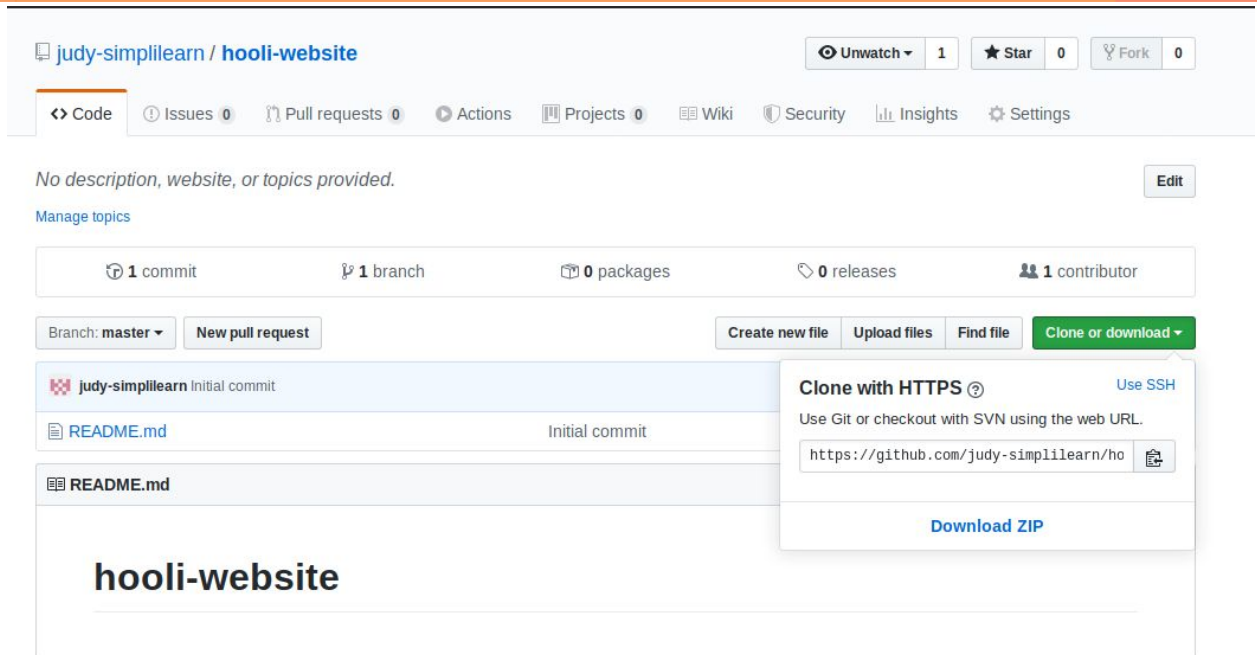
Skip this step if you're importing an existing repository.

- ☒ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **None** 

**Create repository**

- Click on the **Create Repository** button.
- Click on the **Clone or download** button and copy the URL.



## Step 2: Adding the code for the webapp to the repository

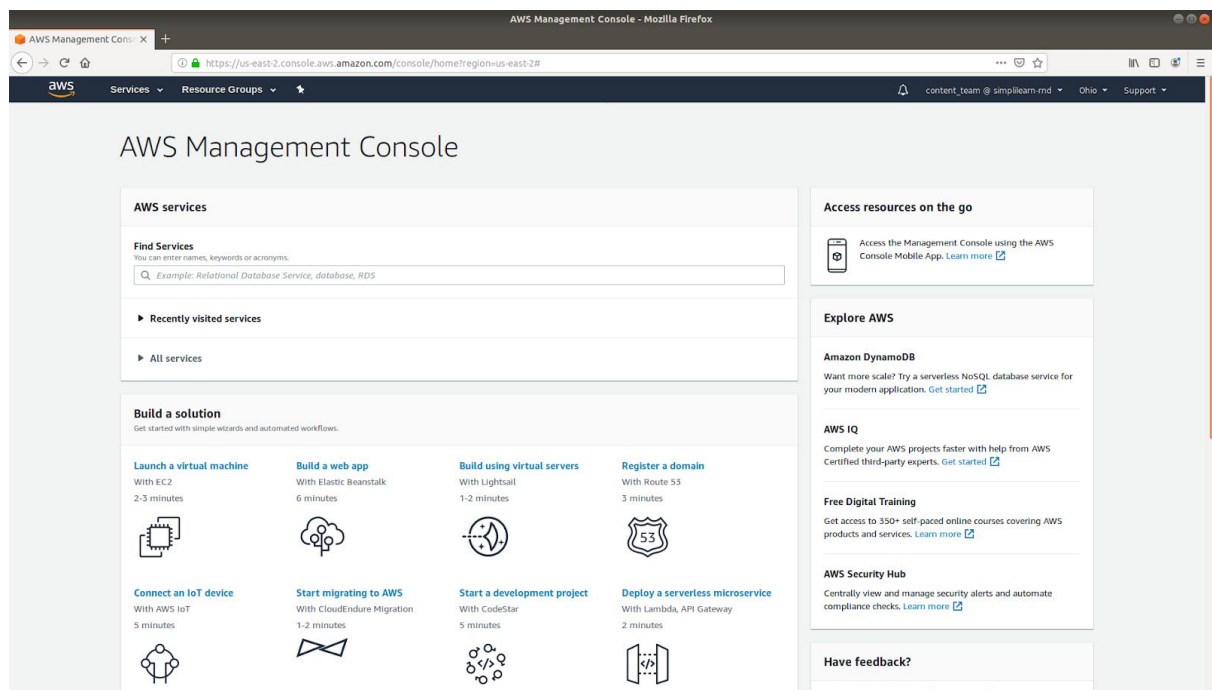
- Open the terminal and navigate to an appropriate location.
- Run **`mvn archetype:generate -DgroupId=com.hooli.app -DartifactId=welcome-app -DarchetypeArtifactId=maven-archetype-webapp -DarchetypeVersion=1.4 -DinteractiveMode=false`** to generate a maven project.
- Navigate inside the Maven app and open the index.jsp file.
- Add the following content to file and save it:
 

```
<html>
<body>
<h2>Welcome to Hooli!</h2>
</body>
</html>
```
- Navigate outside the Maven directory.
- Run **`git clone [URL]`** to clone the repository.
- Move the Maven repository files inside the clone directory.
- Commit the changes to the remote SCM.
- Run **`git add .`**
- Run **`git commit -m "Add new files"`**
- Run **`git push -u origin master`**

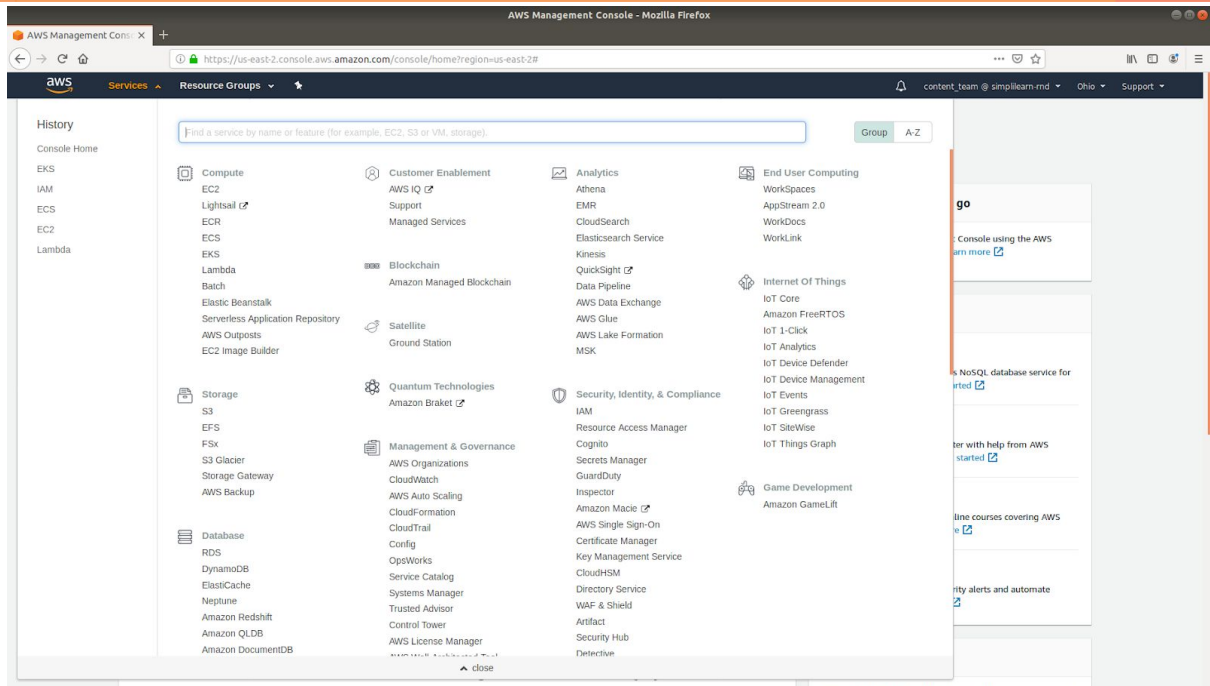
```
judy@SSPL-LP-DNS-0060:~/Downloads/hooli-website$ ls
pom.xml  README.md  src
judy@SSPL-LP-DNS-0060:~/Downloads/hooli-website$ cat src/main/webapp/index.jsp
<html>
<body>
<h2>Welcome to Hooli!</h2>
</body>
</html>
judy@SSPL-LP-DNS-0060:~/Downloads/hooli-website$ git add .
judy@SSPL-LP-DNS-0060:~/Downloads/hooli-website$ git commit -m "New files"
[master 437eeb2] New files
5 files changed, 79 insertions(+), 35 deletions(-)
create mode 100644 pom.xml
delete mode 100644 src/main/java/com/simplilearn/hooli/HooliApplication.java
delete mode 100644 src/main/java/com/simplilearn/hooli/HooliHomeController.java
create mode 100644 src/main/webapp/WEB-INF/web.xml
create mode 100644 src/main/webapp/index.jsp
judy@SSPL-LP-DNS-0060:~/Downloads/hooli-website$ git push -u origin master
```

#### Step 4: Creating an EC2 instance

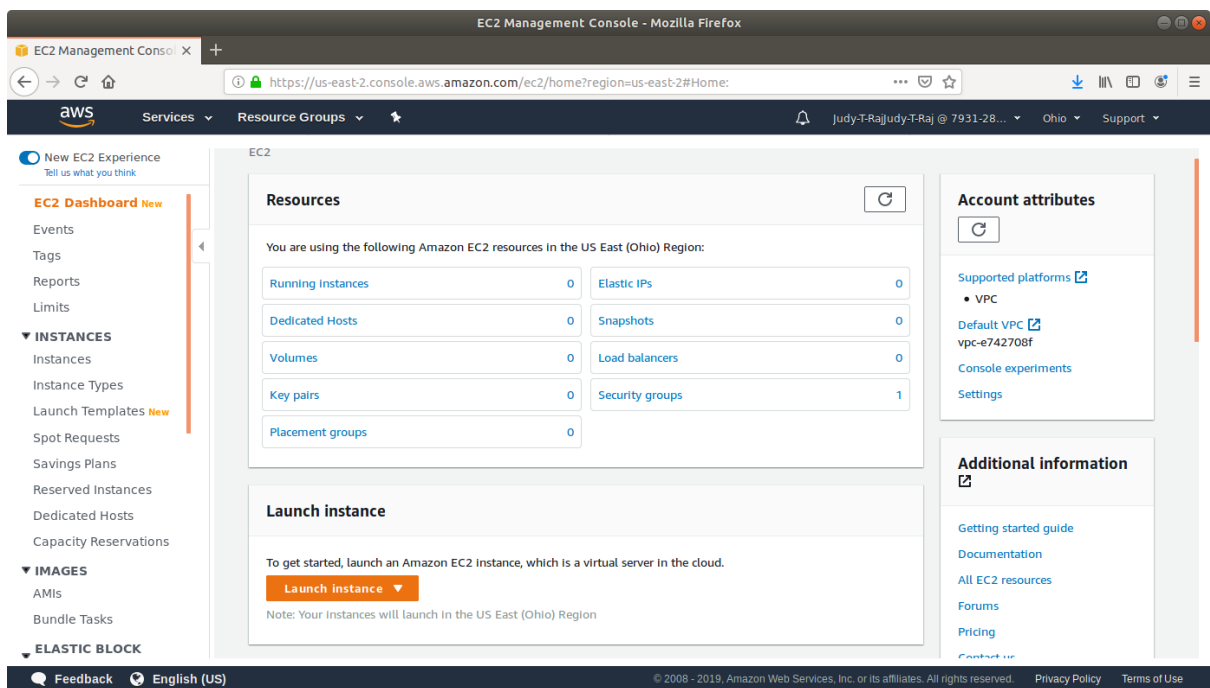
- Log in to the AWS lab account provided. You will then be able to see the following screen:



- Click on Services at the top left to view the drop-down list of resources.
- Click on EC2 under the Compute menu from the drop-down list.



- Click on Launch Instance button and select Launch Instance from the menu.



- Choose an Amazon Machine Image (AMI) from the list of AMIs and click on Select.

Launch Instance wizard | EC2 Management Console - Mozilla Firefox

Launch instance wizard | x +

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Judy-T-Raj@Judy-T-Raj @ 7931-28... Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

**Red Hat**  
Free tier eligible  
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0520e698dd500b1d1 (64-bit x86) / ami-0099847d600887c9f (64-bit Arm)  
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

**SUSE Linux**  
Free tier eligible  
SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-052a6e77572eba9a9 (64-bit x86) / ami-034ecb883363663c5 (64-bit Arm)  
SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

**Ubuntu Server 18.04 LTS**  
Free tier eligible  
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0d5d9d301c853a04a (64-bit x86) / ami-0fb0129cd568fe35f (64-bit Arm)  
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

**Are you launching a database instance? Try Amazon RDS.**  
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Choose an Instance Type and click Review and Launch.

Launch Instance wizard | EC2 Management Console - Mozilla Firefox

Launch instance wizard | x +

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Judy-T-Raj@Judy-T-Raj @ 7931-28... Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click on Launch.

**Step 7: Review Instance Launch**

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**AMI Details** [Edit AMI](#)

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0d5d9d301c853a04a**

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

**Instance Type** [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

**Security Groups** [Edit security groups](#)

Security group name: launch-wizard-1  
Description: launch-wizard-1 created 2019-12-17T15:56:06.249+05:30

Type	Protocol	Port Range	Source	Description

[Cancel](#) [Previous](#) [Launch](#)

[Feedback](#) [English \(US\)](#) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- In the pop-up menu, select Create a new key-value pair.
- Click on Download Key. You'll need this key to SSH to the VM later.

**Select an existing key pair or create a new key pair**

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name: my\_key

[Download Key Pair](#)

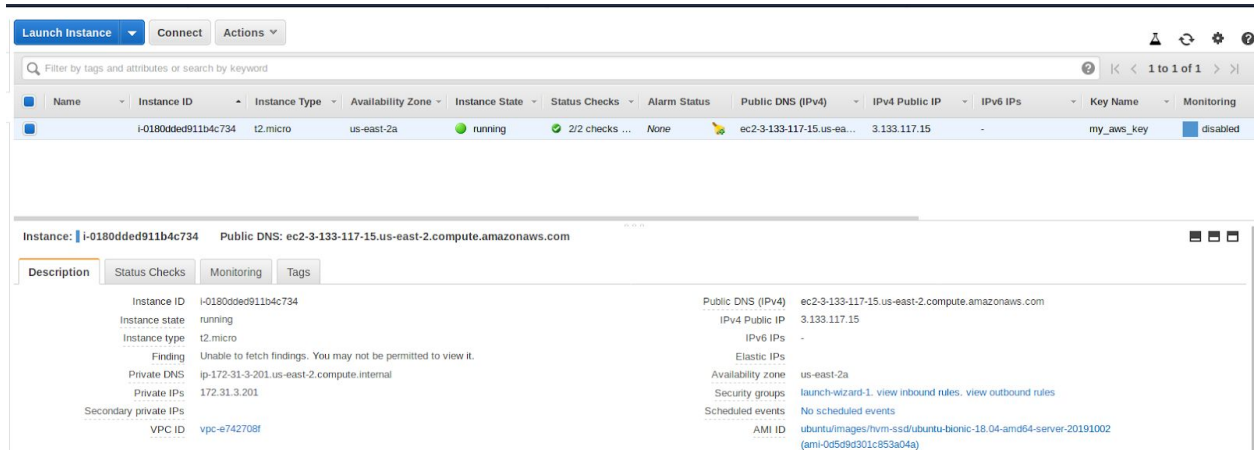
You have to download the **private key file** (\*.pem file) before you can continue. Store it in a **secure and accessible location**. You will not be able to download the file again after it's created.

[Cancel](#) [Launch Instances](#)

[Feedback](#) [English \(US\)](#) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- Click on Launch.
- Navigate to the security groups console .





- Add a rule to the security group to which the instance belongs to allow SSH, with the following settings:

**Type:** SSH

**Protocol:** TCP

**Port Range:** 22

**Source:** Anywhere 0.0.0.0/0

- Add a rule to the security group to which the instance belongs to allow http traffic to port 8080, with the following settings:

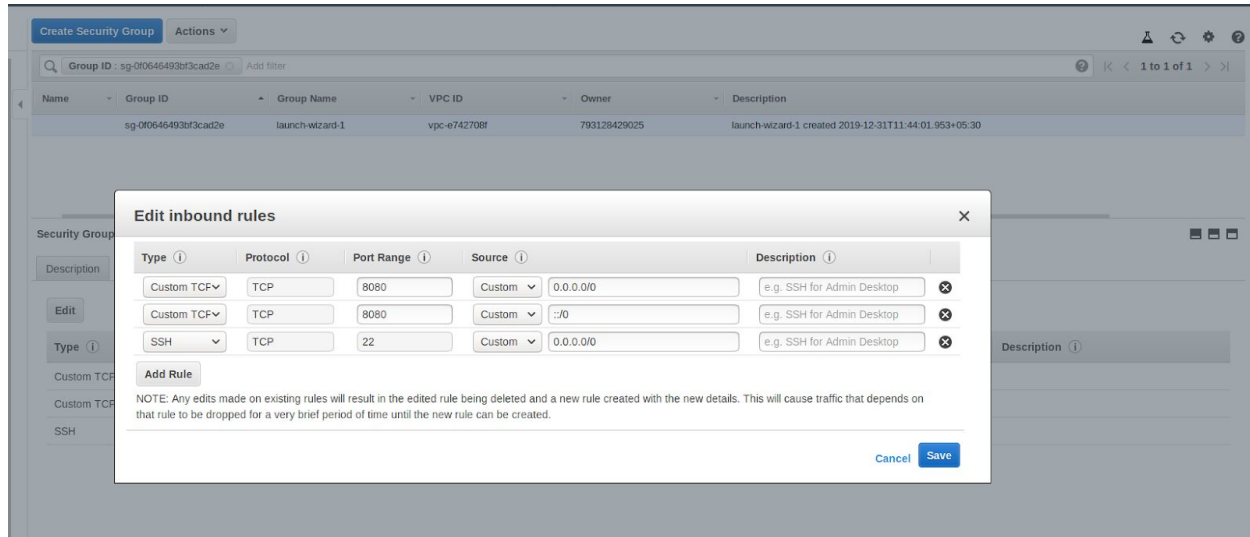
**Type:** Custom TCP Rule

**Protocol:** TCP

**Port Range:** 8080

**Source:** Anywhere 0.0.0.0/0





## Step 5: Installing Tomcat on EC2

- Open the terminal.
- Navigate to the location where the AWS key is stored.
- Make the key file executable with the command **chmod 400 <key-name>.pem**
- SSH to the EC2 instance with the command **sudo ssh -i <key-name>.pem ubuntu@<public-dns>**

```
judy@SSPL-LP-DNS-0060:~/Downloads$ chmod 400 my_aws_key.pem
judy@SSPL-LP-DNS-0060:~/Downloads$ sudo ssh -i "my_aws_key.pem" ubuntu@ec2-3-133-117-15.us-east-2.compute.amazonaws.com
[sudo] password for judy:
The authenticity of host 'ec2-3-133-117-15.us-east-2.compute.amazonaws.com (3.133.117.15)' can't be established.
ECDSA key fingerprint is SHA256:QuclPccAiVK9XoH0Rdv++ysA1nKerThNdtLMBctMZaE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-133-117-15.us-east-2.compute.amazonaws.com,3.133.117.15' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Dec 31 06:29:31 UTC 2019

System load:  0.0          Processes:      86
Usage of /:   13.8% of 7.69GB   Users logged in:  0
Memory usage: 17%          IP address for eth0: 172.31.3.201
Swap usage:   0%

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

- Run the following commands to install Java and Tomcat and grant permissions to the Tomcat user:

```
sudo apt-get update
```

```
sudo apt-get install default-jdk
```

```
sudo groupadd tomcat
```

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

```
cd /tmp
```

```
curl -O
```

```
http://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.30/bin/apache-tomcat-9.0.30.tar.gz
```

```
sudo mkdir /opt/tomcat
```

```
sudo tar xzvf apache-tomcat-9*.tar.gz -C /opt/tomcat --strip-components=1
```

```
cd /opt/tomcat
```

```
sudo chgrp -R tomcat /opt/tomcat
```

```
sudo chmod -R g+r conf
```

```
sudo chmod g+x conf
```

```
sudo chown -R tomcat /opt/tomcat
```

```
sudo chown -R tomcat webapps/ work/ temp/ logs/
```

- Find the path to Java with the following command:

```
sudo update-java-alternatives -l
```

- Open the tomcat.service file with the command **sudo nano /etc/systemd/system/tomcat.service**
- Add the following content to the file. Replace the JAVA\_HOME value with the value obtained in the previous step:

```
Description=Apache Tomcat Web Application Container
```

```
After=network.target
```

```
[Service]
```

**Type=forking**

**Environment=JAVA\_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64/jre**

**Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid**

**Environment=CATALINA\_HOME=/opt/tomcat**

**Environment=CATALINA\_BASE=/opt/tomcat**

**Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'**

**Environment='JAVA\_OPTS=-Djava.awt.headless=true  
-Djava.security.egd=file:/dev/./urandom'**

**ExecStart=/opt/tomcat/bin/startup.sh**

**ExecStop=/opt/tomcat/bin/shutdown.sh**

**User=tomcat**

**Group=tomcat**

**UMask=0007**

**RestartSec=10**

**Restart=always**

**[Install]**

**WantedBy=multi-user.target**

- Add the following content to the file. Replace the JAVA\_HOME value with the value obtained in the previous step.

```
Description=Apache Tomcat Web Application Container
After=network.target
[Service]
Type=forking
Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64/jre
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
```

- Open the server.xml file with the command **sudo nano conf/server.xml**.
- Add **address="0.0.0.0"** to connector and save the file.

```

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="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    address="0.0.0.0"/>
  <!-- A "Connector" using the shared thread pool-->
  <!--
  <Connector executor="tomcatThreadPool"
    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  -->
  <!-- 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.
  -->
  <!--
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150" SSLEnabled="true">
    <SSLHostConfig>
      <Certificate certificateKeystoreFile="conf/localhost-rsa.jks"
        type="RSA" />
    </SSLHostConfig>
  </Connector>
  -->

```

- Open the users file with the following command: **sudo nano /opt/tomcat/conf/tomcat-users.xml**
- Add the following lines right before the last line and save the file:

```
<user username="tomcatmanager" password="password" roles="manager-gui"/>
```

```
<user username="deployer" password="password" roles="manager-script"/>
```



```
<?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">
  <!--
  NOTE: 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. It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
  -->
  <!--
  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"/>
  -->
  <user username="tomcatmanager" password="password" roles="manager-gui"/>
  <user username="deployer" password="password" roles="manager-script"/>
</tomcat-users>
```

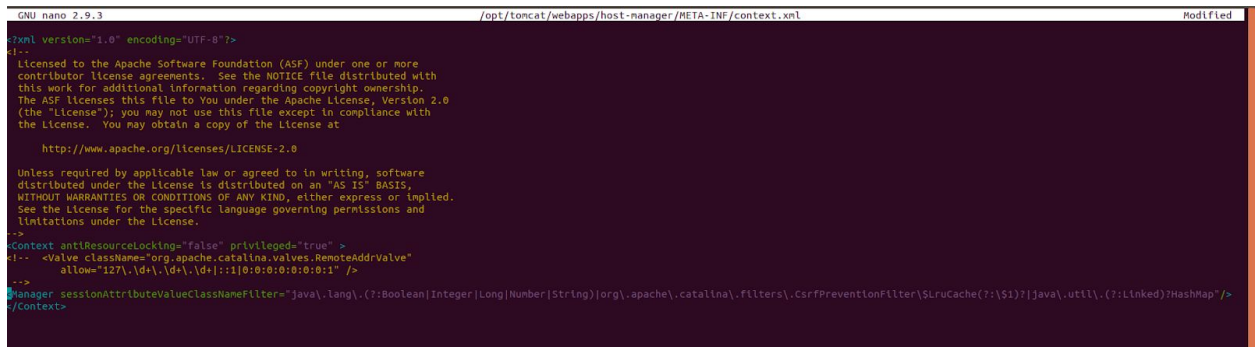
- By default, newer versions of Tomcat restrict access to the Manager and Host Manager apps to connections coming from the server itself. Since we are installing on a remote machine, you will need to remove or alter this restriction. To change the IP address

restrictions on these, open the appropriate context.xml files with the following commands:

**sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml and**

**sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml**

- Comment out the IP address restriction to allow connections from anywhere.



```
GNU nano 2.9.3 /opt/tomcat/webapps/host-manager/META-INF/context.xml Modified
<?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" >
  <!-- <Value 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|Long|Number|String|org\.apache\.catalina\.filters\.CsrfPreventionFilter|SLRUCache(?:|String|java\.util\.(?:Linked)?HashMap)" />
  -->
</context>
```

\

- Start and verify the Tomcat server with the following commands:

**sudo systemctl daemon-reload**

**sudo systemctl start tomcat**

**sudo systemctl status tomcat**

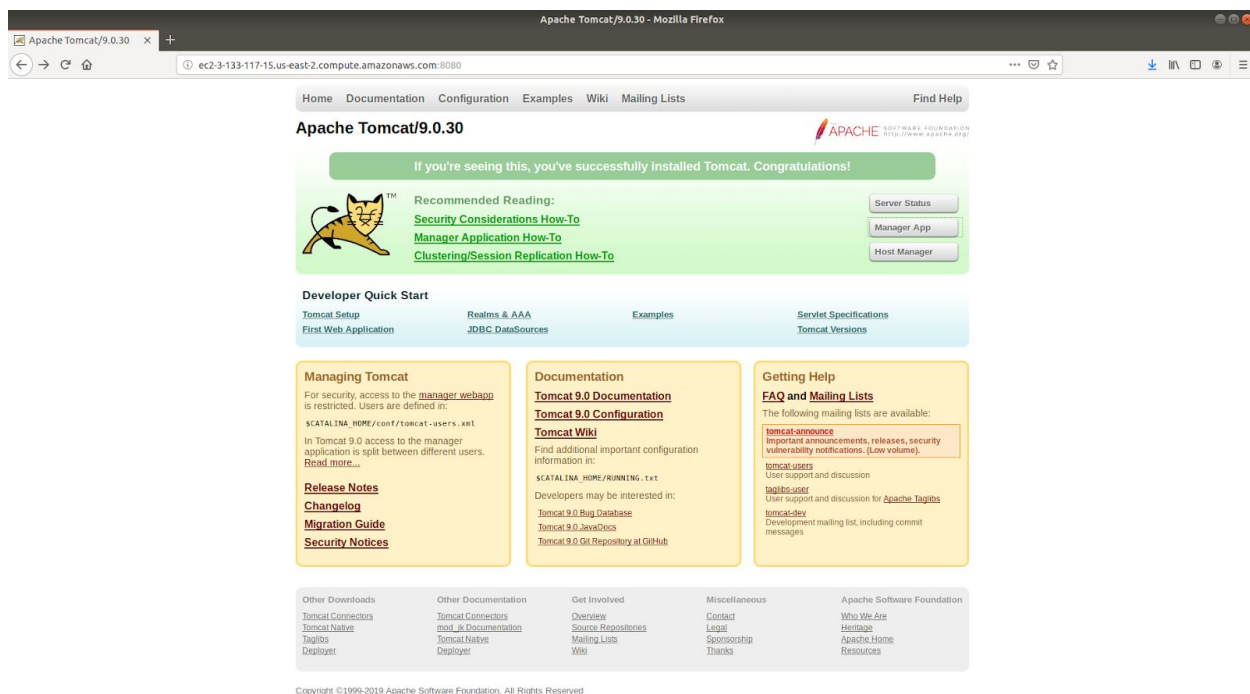
- Allow traffic at 8080 with the following command:

**sudo ufw allow 8080**



```
ubuntu@ip-172-31-9-241:/tmp$ cd /opt/tomcat
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo chgrp -R tomcat /opt/tomcat
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo chmod -R g+r conf
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo chmod g+x conf
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo chown -R tomcat /opt/tomcat
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo chown -R tomcat webapps/ work/ temp/ logs/
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo update-java-alternatives -l
java-1.11.0-openjdk-amd64 1111 /usr/lib/jvm/java-1.11.0-openjdk-amd64
ubuntu@ip-172-31-9-241:/opt/tomcat$ ^C
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo nano /etc/systemd/system/tomcat.service
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo nano conf/server.xml
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo nano /opt/tomcat/conf/tomcat-users.xml
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo ufw allow 8080
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-9-241:/opt/tomcat$ sudo /opt/tomcat/bin/startup.sh
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
ubuntu@ip-172-31-9-241:/opt/tomcat$
```

- Navigate to [http://<Public DNS \(IPv4\)>:8080](http://<Public DNS (IPv4)>:8080) to view the tomcat server.



## Step 6: Creating a deployment pipeline in Jenkins

- Go to Jenkins dashboard.
- Click on *Manage Jenkins* and select *Manage Plugins*.

- From the available plugins, install *Deploy to container*.

Name	Version
<a href="#">Deploy to container</a> This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment	1.15

- Click on *Install without restart*.

# Installing Plugins/Upgrades

## Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

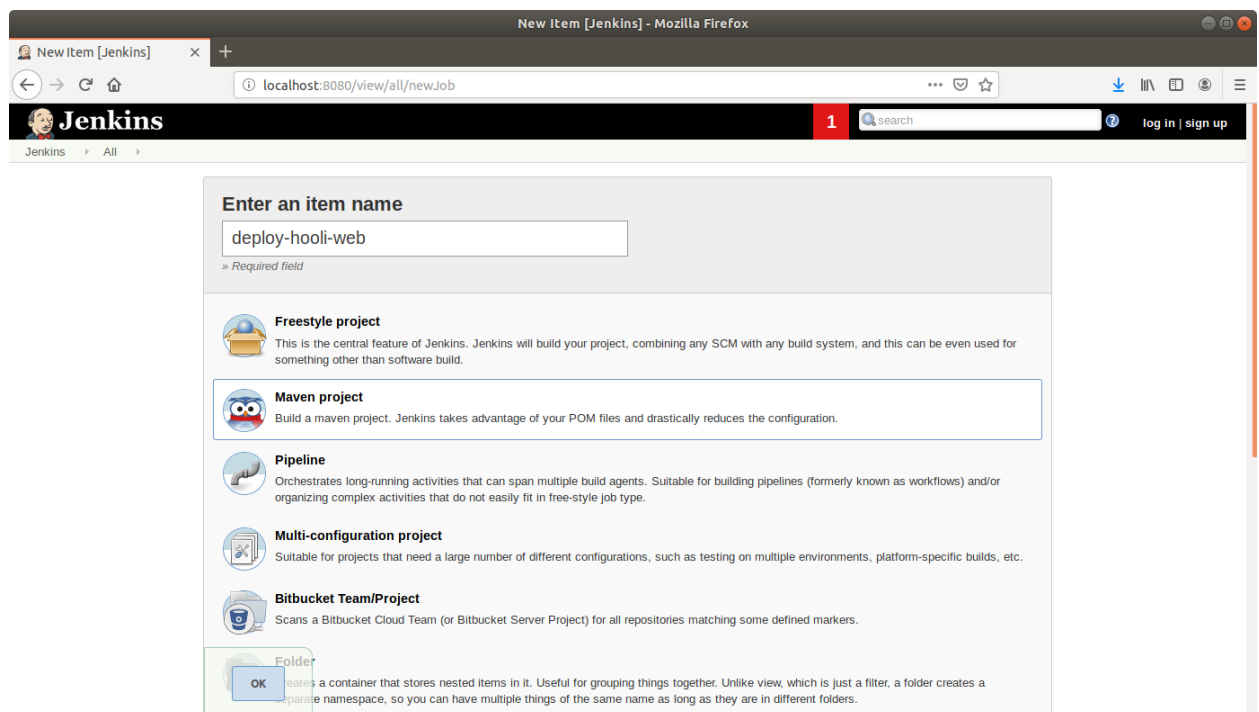
Deploy to container  Success

Loading plugin extensions  Success

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

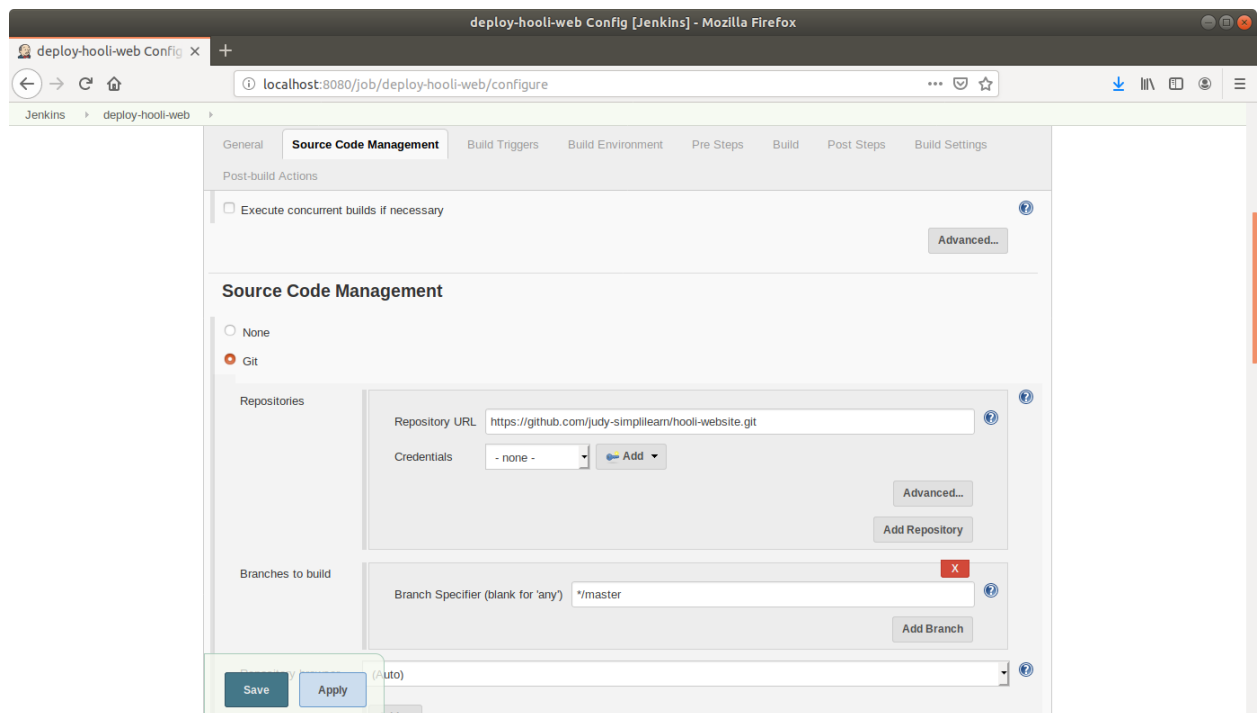
➡ ☐ Restart Jenkins when installation is complete and no jobs are running

- Click on *New Item*.
- Enter a name for your build job.
- Select *Maven Project* as the build job type.

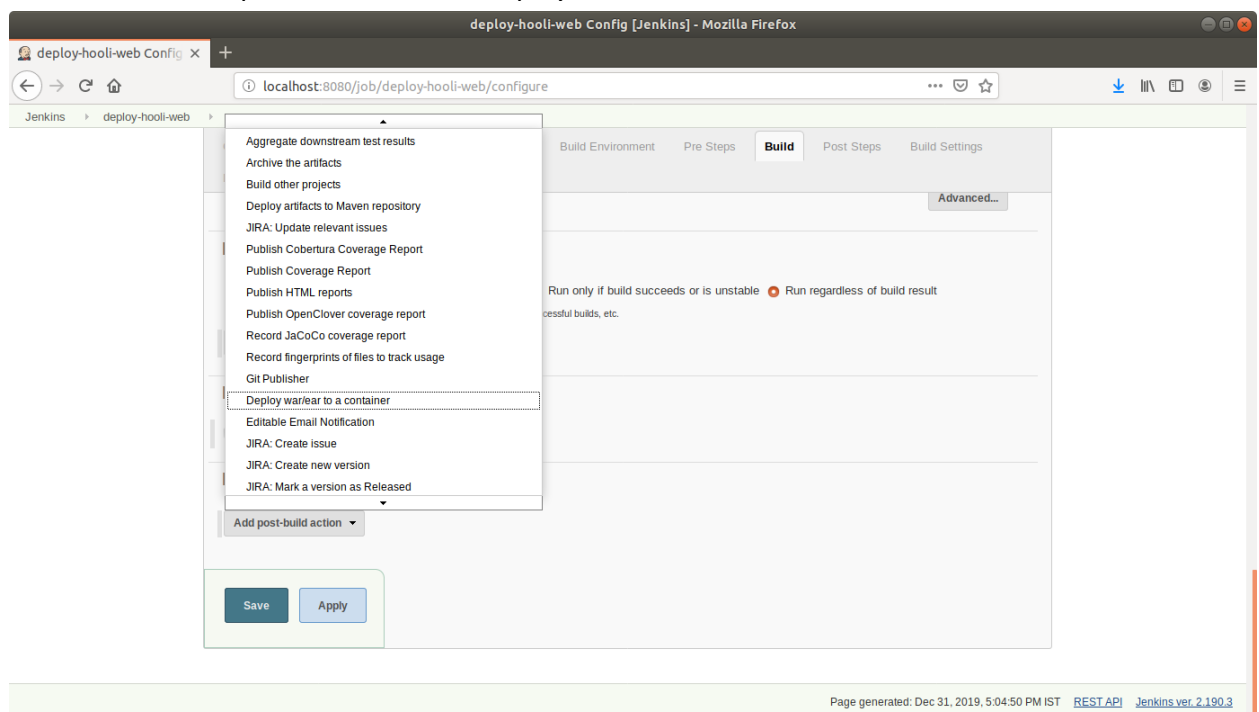


- Click OK.
- On the configuration page, scroll down to the Source Code Management section.

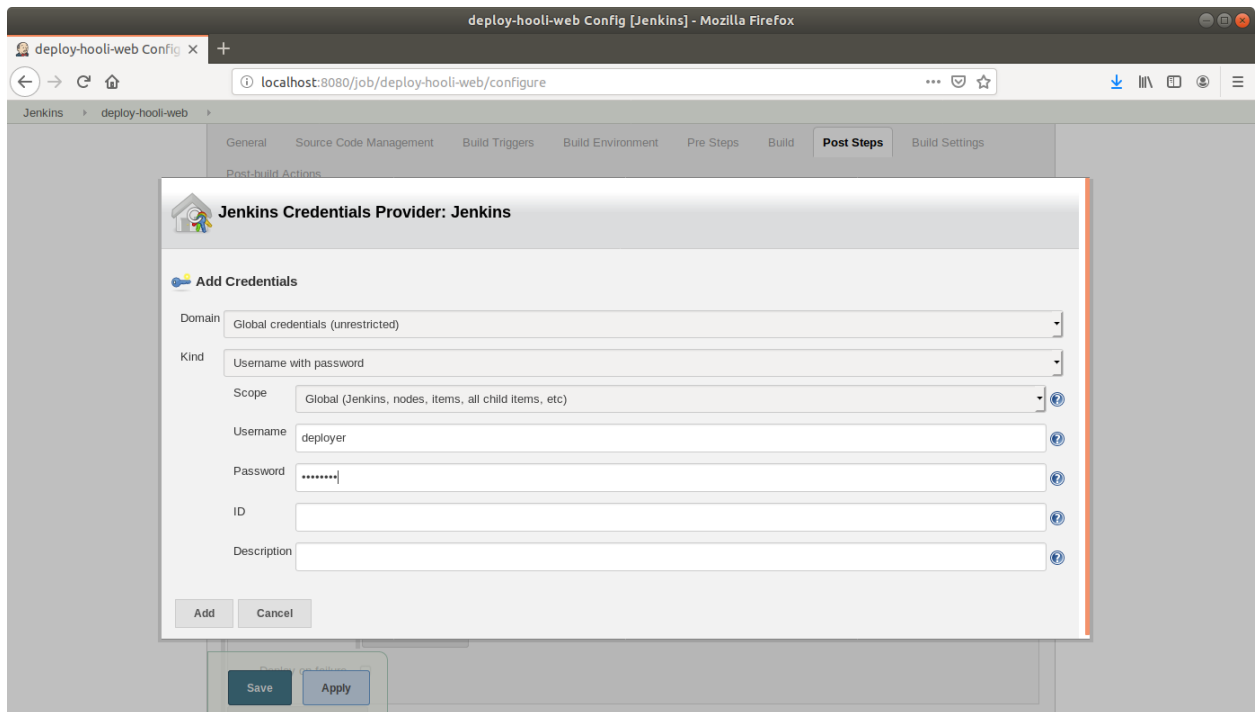
- Select *Git in SCM*.
- Add the repository URL.



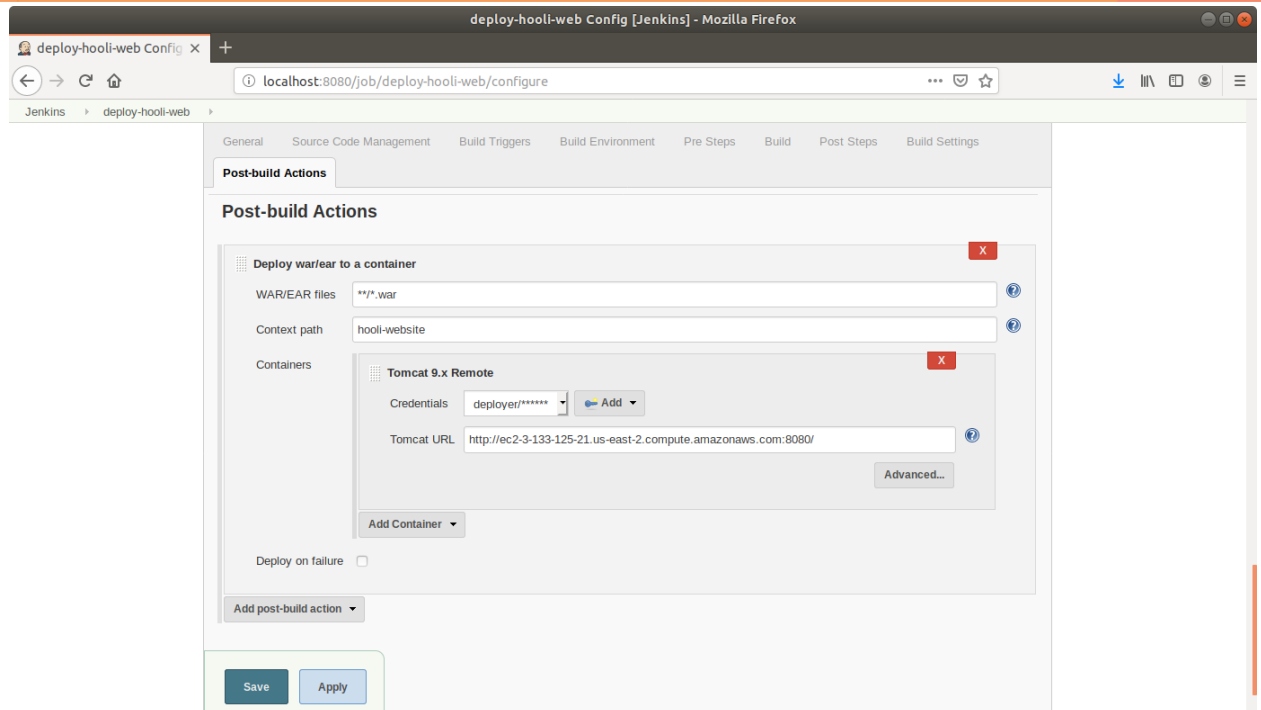
- Drag to the bottom and go to the *Post-build Actions* section.
- Click on *Add post-build action* button.
- On the available options click on the *Deploy war/ear to container*.



- Add the credentials.



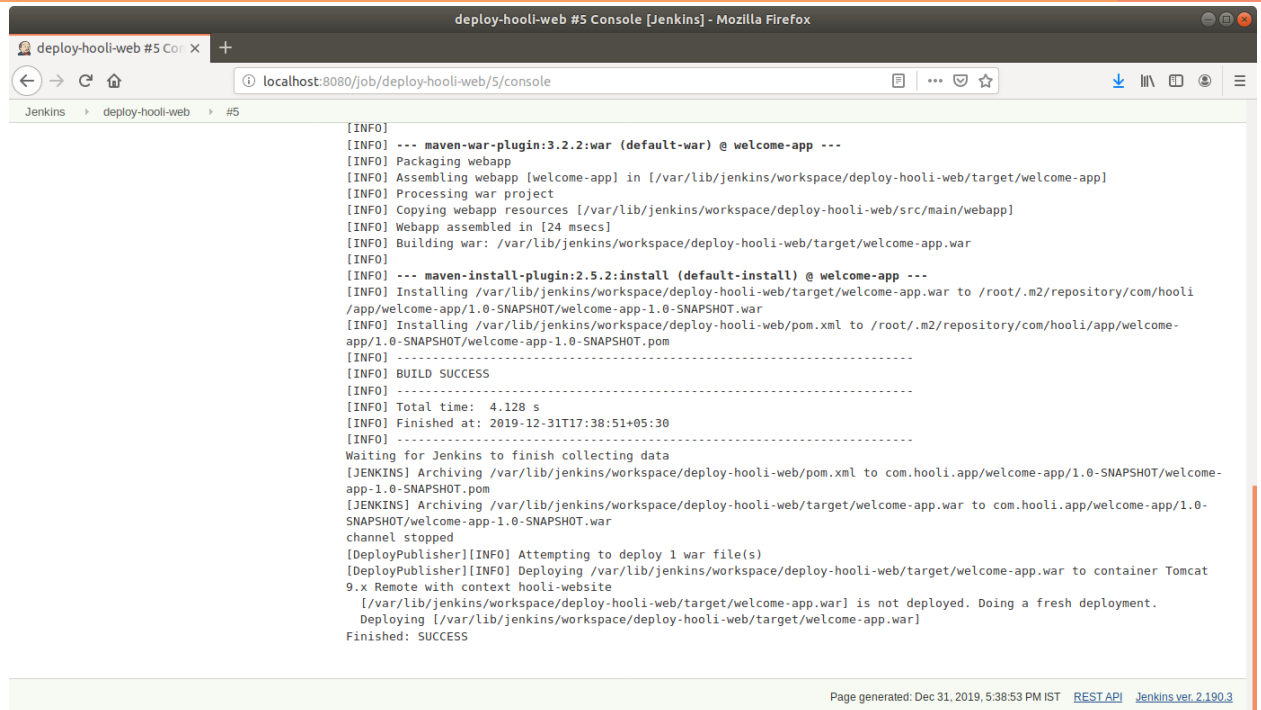
- Fill the required parameters for the plugin. Use the following screenshot as a reference:



- Choose the Context Path in which the application should be installed. It would rename the WAR file before deploying to the server and thereby the application context root would be changed.
- Click Save.

### Step 7: Running a deployment pipeline in Jenkins

- Click on *Build Now* in the project window.
- Jenkins will now build your pipeline and output the logs.



```

deploy-hooli-web #5 Console [Jenkins] - Mozilla Firefox
localhost:8080/job/deploy-hooli-web/5/console
Jenkins > deploy-hooli-web > #5

[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ welcome-app ---
[INFO] Packaging webapp
[INFO] Assembling webapp [welcome-app] in [/var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/deploy-hooli-web/src/main/webapp]
[INFO] Webapp assembled in [24 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app.war
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ welcome-app ---
[INFO] Installing /var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app.war to /root/.m2/repository/com/hooli/app/welcome-app/1.0-SNAPSHOT/welcome-app-1.0-SNAPSHOT.war
[INFO] Installing /var/lib/jenkins/workspace/deploy-hooli-web/pom.xml to /root/.m2/repository/com/hooli/app/welcome-app/1.0-SNAPSHOT/welcome-app-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.128 s
[INFO] Finished at: 2019-12-31T17:38:51+05:30
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/deploy-hooli-web/pom.xml to com.hooli.app/welcome-app/1.0-SNAPSHOT/welcome-app-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app.war to com.hooli.app/welcome-app/1.0-SNAPSHOT/welcome-app-1.0-SNAPSHOT.war
channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app.war to container Tomcat
9.x Remote with context hooli-website
[/var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/deploy-hooli-web/target/welcome-app.war]
Finished: SUCCESS

Page generated: Dec 31, 2019, 5:38:53 PM IST REST API Jenkins ver. 2.190.3

```

- Navigate to the URL on your browser to view your webapp.