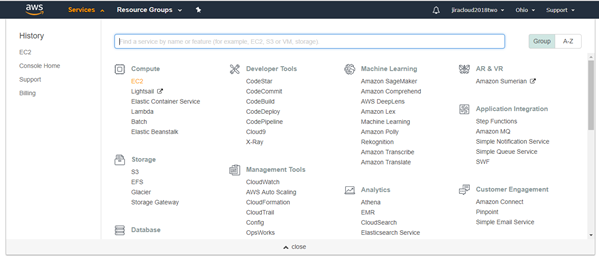
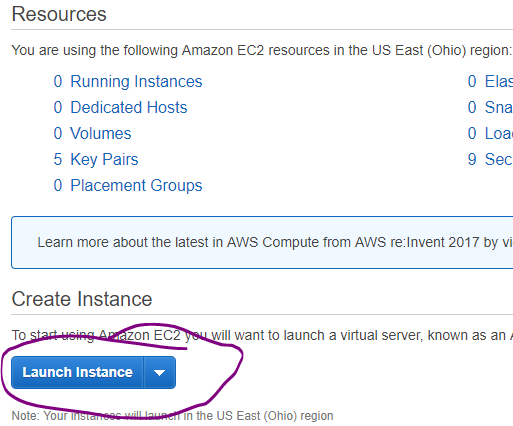
|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Pages** |
| A | Launch the EC2 redhat instance in AWS & connect to the instance. | 1-6 |
| **B** | **Tomcat Start-up.** | 7-11 |
| **C** | **Jenkins Start-up** & sample job configuration. | 12-25 |
| **D** | **Automatic build trigger – Configuration setup** | 25-27 |

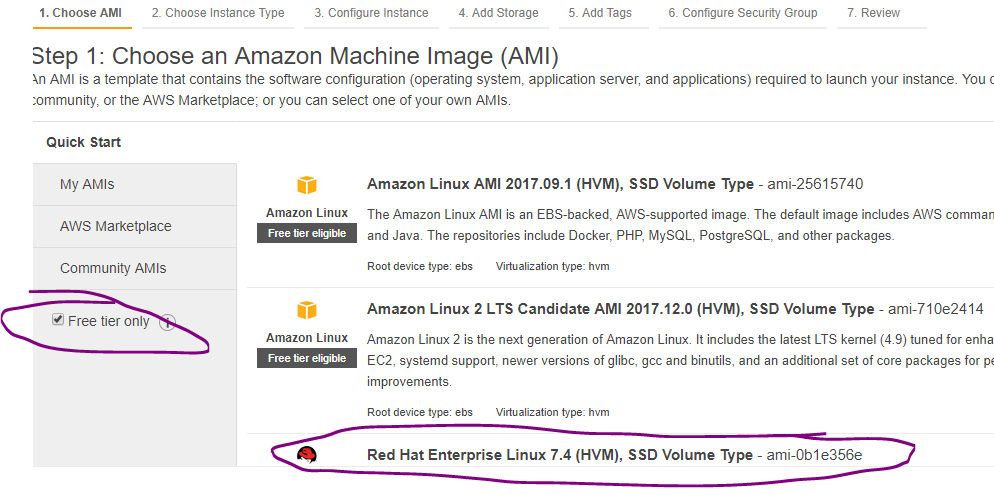
1. Launch the EC2 redhat instance in AWS & connect to the instance.
2. Login to AWS 🡪Services (top left side) 🡪 EC2.



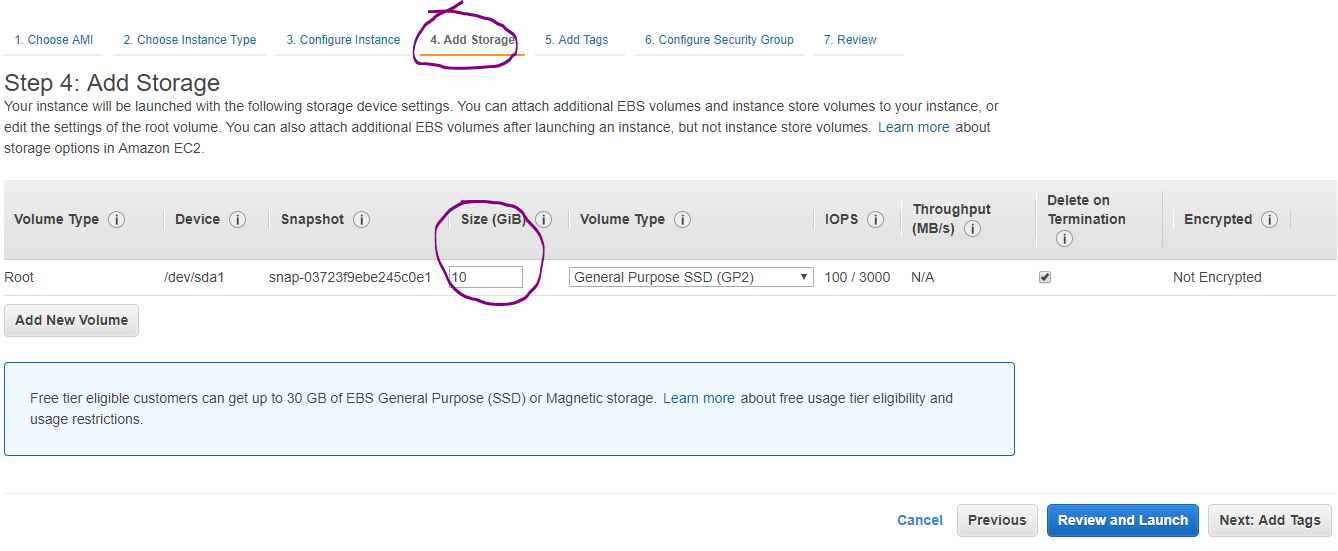
1. Click on “Launch Instance”.



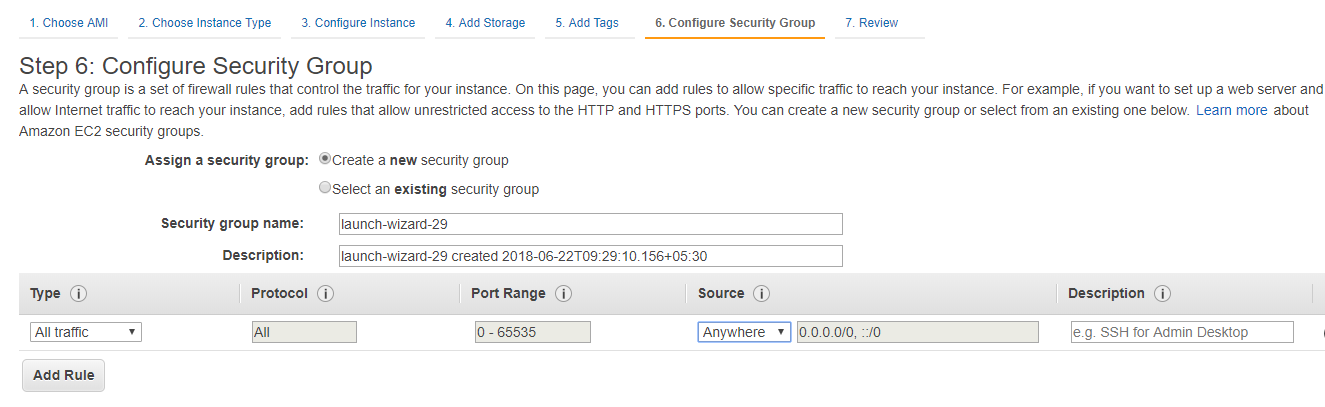
Choose “Free Tier Only”🡪Choose RHEL machine.



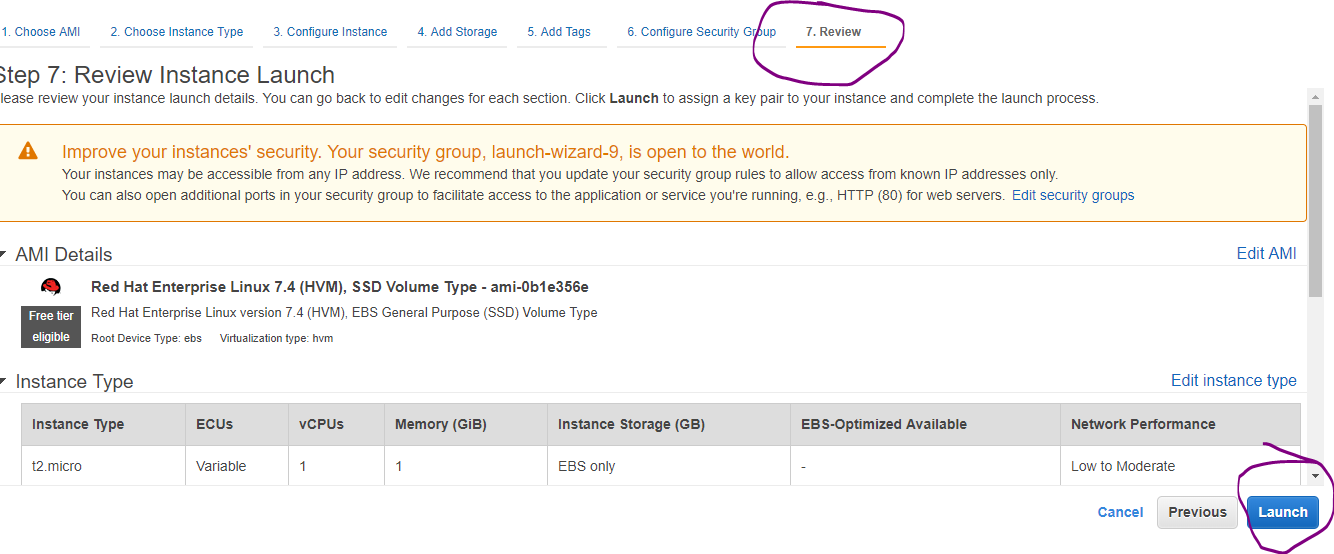
Default size is 10GB (its more enough for us) 🡪 click on “Review and Launch”.



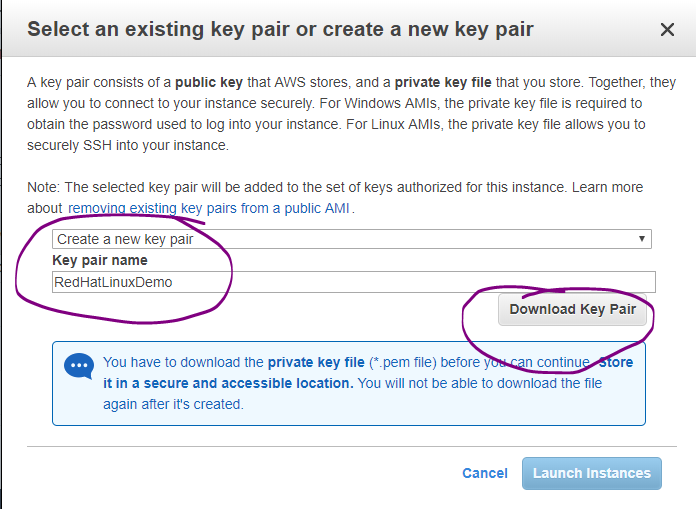
Click Configure security group and update the option as per the below screen. This option is required to access the IP outside of the network ( Public).



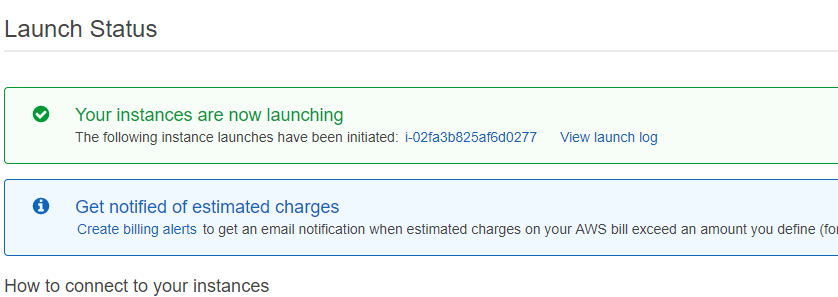
Click on Launch.

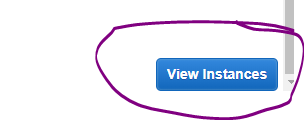


Select “Create a new key pair” 🡪 give the file name🡪 download🡪 file will be downloaded to your local machine. 🡪 click on Launch Instances.

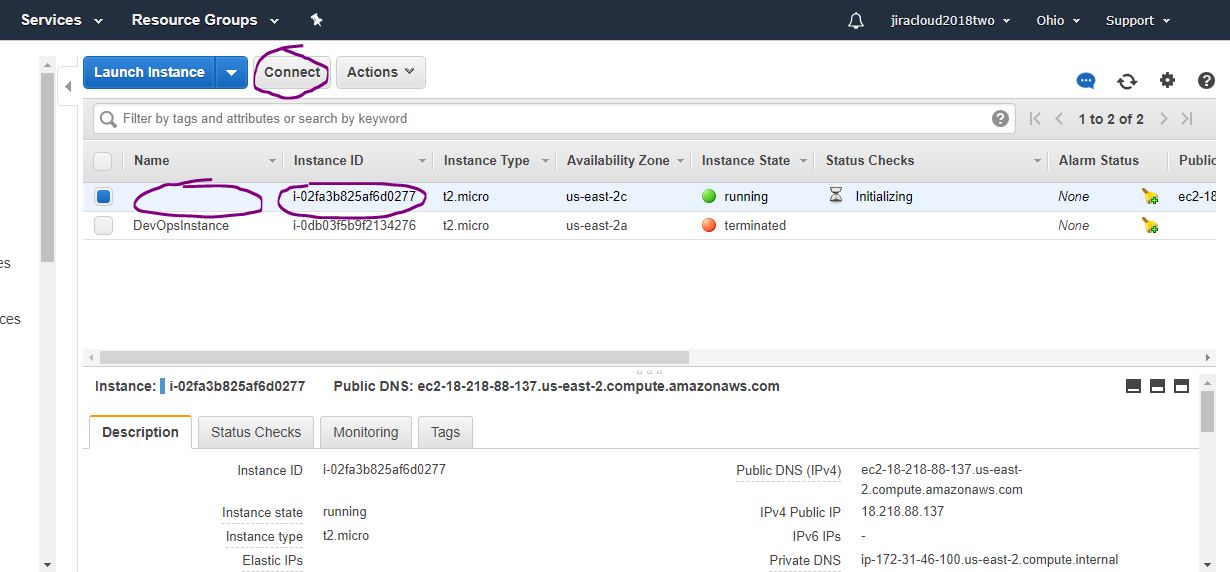


Scroll down in this page and click on view launch

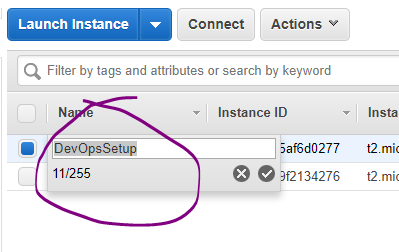




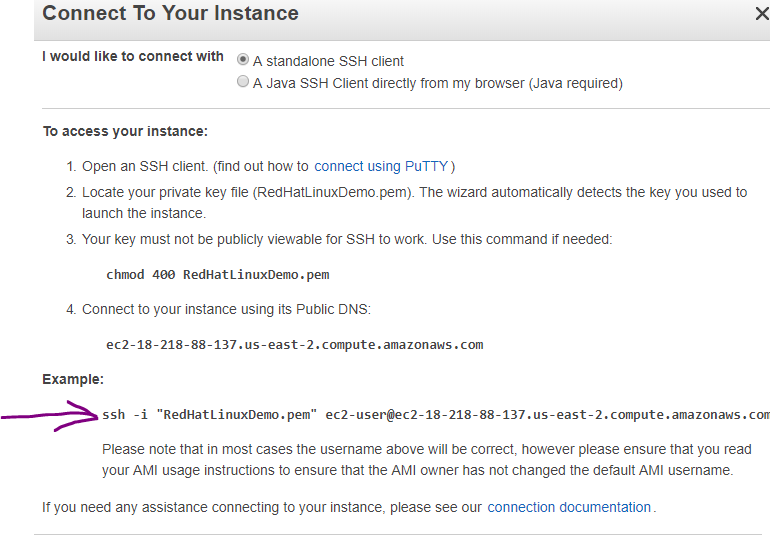
You can give the instance name which is showing empty below. Choose the instance & click on connect if you want to connect.



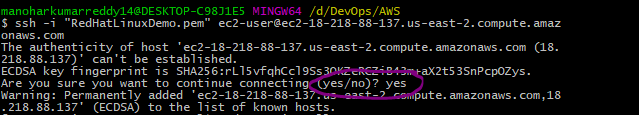
You can give the instance name as showing below.



Choose the instance & click on connect. Copy the ssh command & paste it on “git bash” to connect to the instance.



Use Gitbash or putty to connect the machine. For Git bash, the keys should be available in the current folder.



Refer the tools installation document to install the required tools in the VM.

#Installation Procedure: Refer the attachment below and follow the below procedure.

#Step-1: Copy the lines from 1 to 182, and then paste it on your terminal. All the commands will be executed one by one.

#Step-2: Run the below commands make sure java, git, maven installed properly.

Type the below commands in the git bash or any command prompt and check the whether the tools are installed or not.

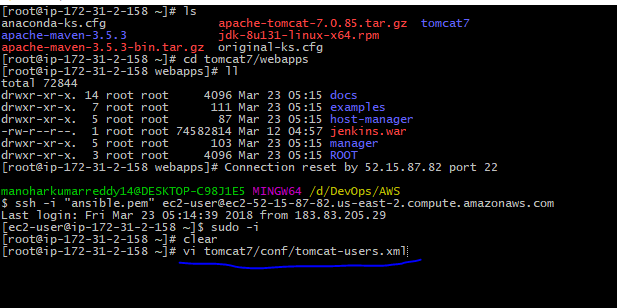
#java -version, mvn -v, git --version

#Step-3: Run this command, jenkins.war should be there: ll tomcat7/webapps/

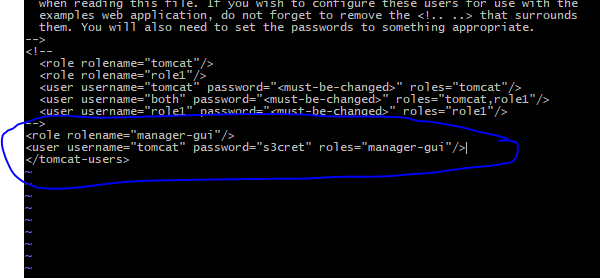


1. **Tomcat Start-up**

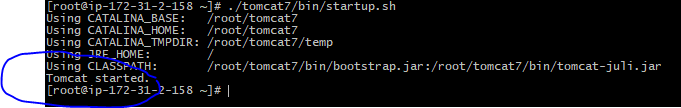
Update tomcat-users.xml file: vi tomcat7/conf/tomcat-users.xml



|  |
| --- |
| Update tomcat-users.xml file with below content.<role rolename="manager-gui"/>  <user username="tomcat" password="s3cret" roles="manager-gui"/> |

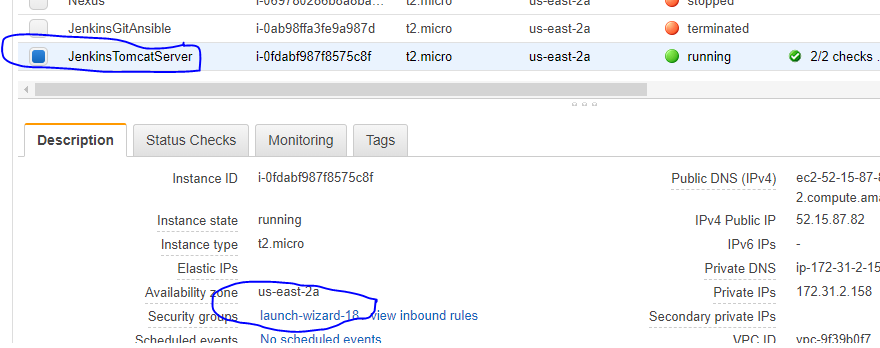


Startup the tomcat server: ./tomcat7/bin/startup.sh

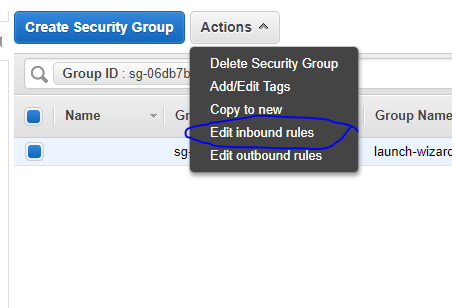


To launch the tomcat url on web browser, we have to free all the security restrictions.

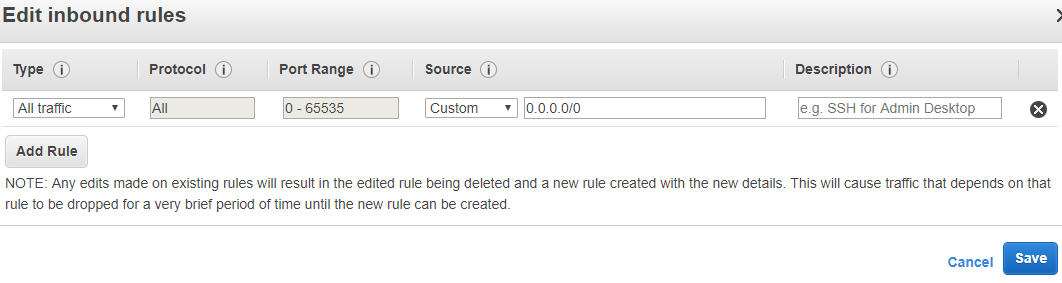
For that go to AWS, choose the instance where tomcat installed. Clock on the security group as shown below.



Clock on Actions🡪Edit Inbound Rules.



Change the “Type” from the default value “ssh” to “All traffic” and then click on save.

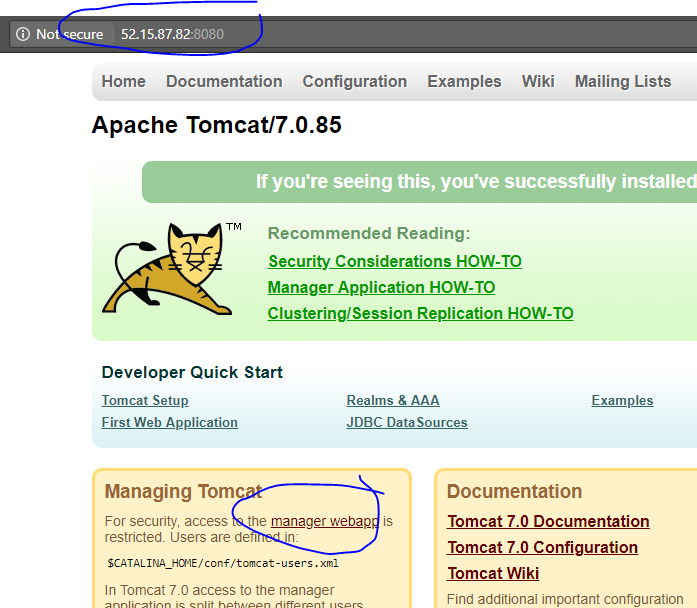


Go o instances 🡪 choose your instance🡪 copy the public IP.



http://<publicIp>:8080 ex: <http://52.15.87.82:8080> 🡪 it will navigate to tomcat server home page.

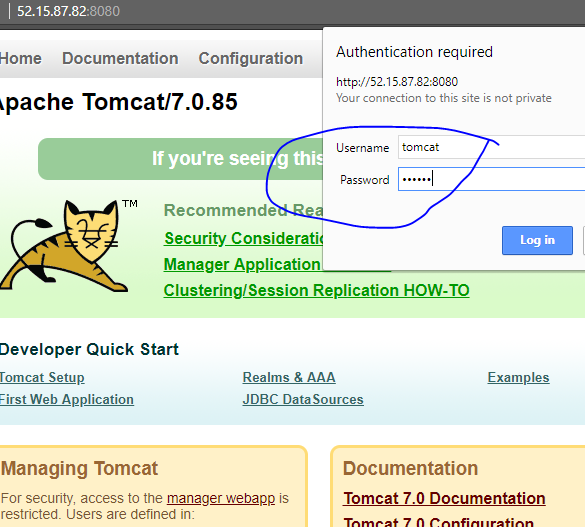
And click on “manager webapp”.



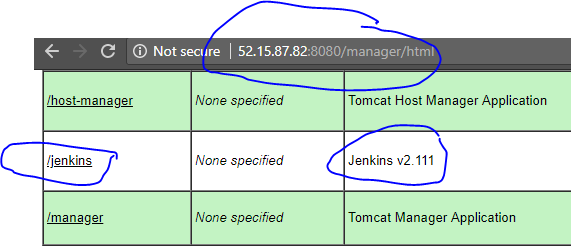
Enter the username & pwd which were configured in tomcat-users.xml file 🡪 click on Log in.

<role rolename="manager-gui"/>

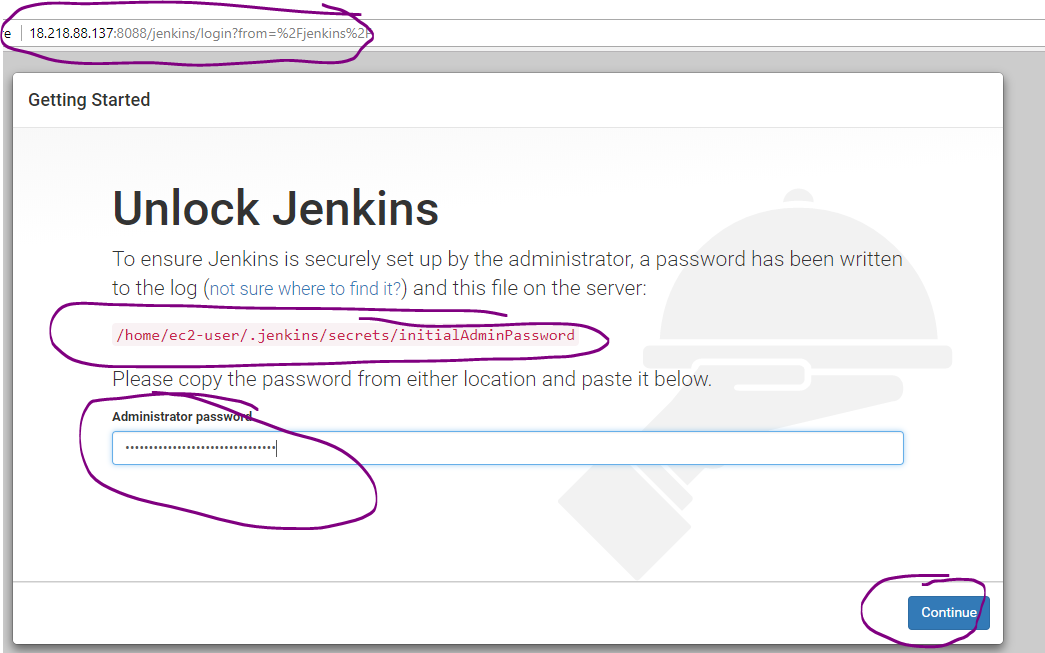
<user username="tomcat" password="s3cret" roles="manager-gui"/>

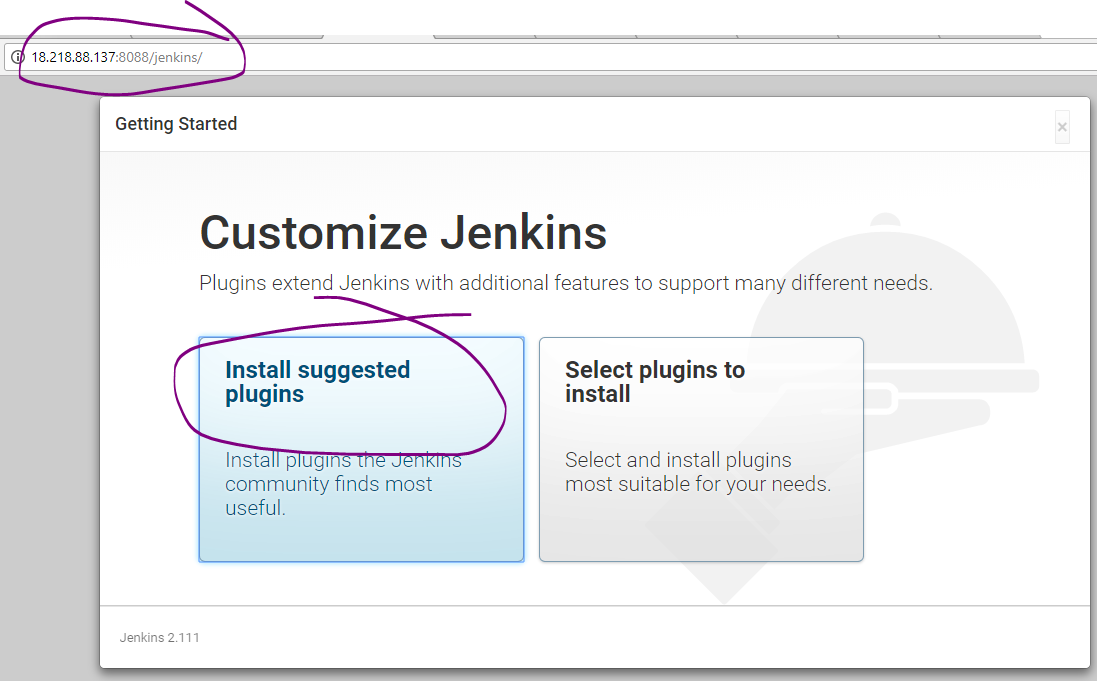


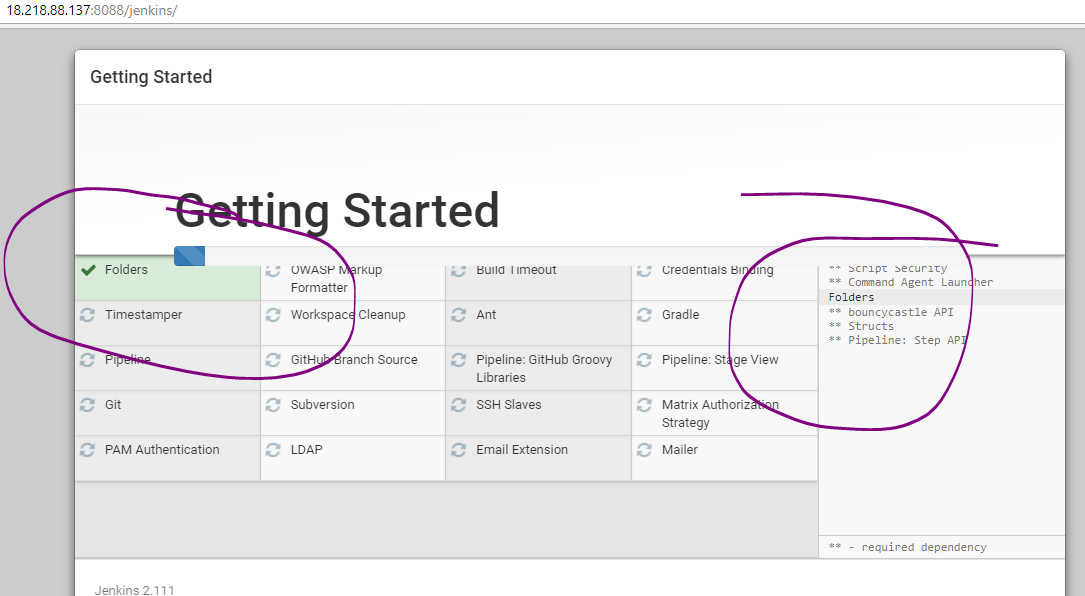
Here you will find the Jenkins🡪 click on it.

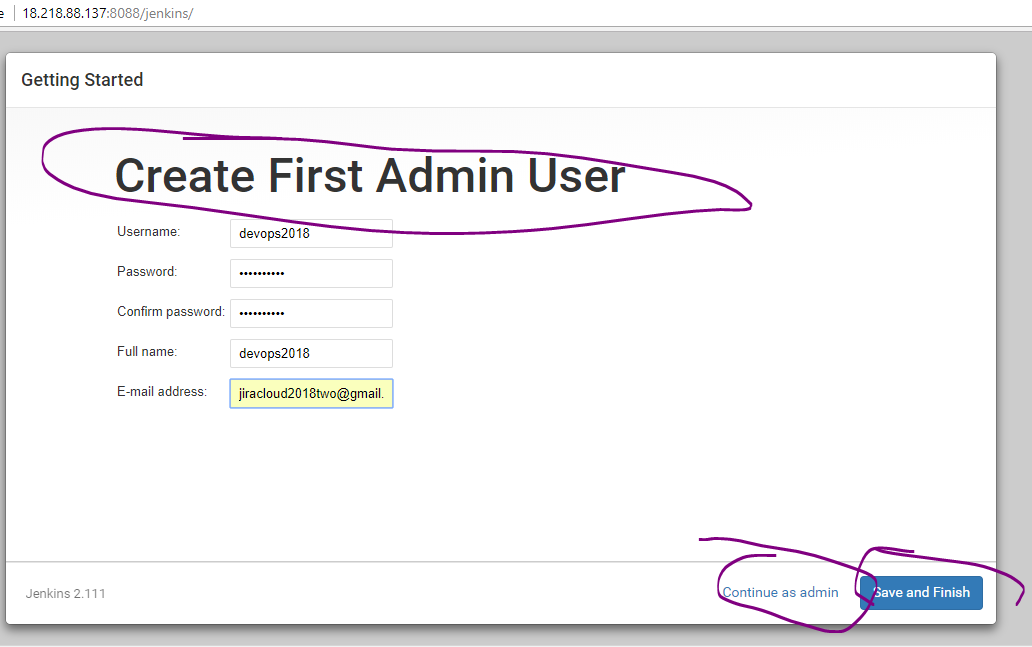


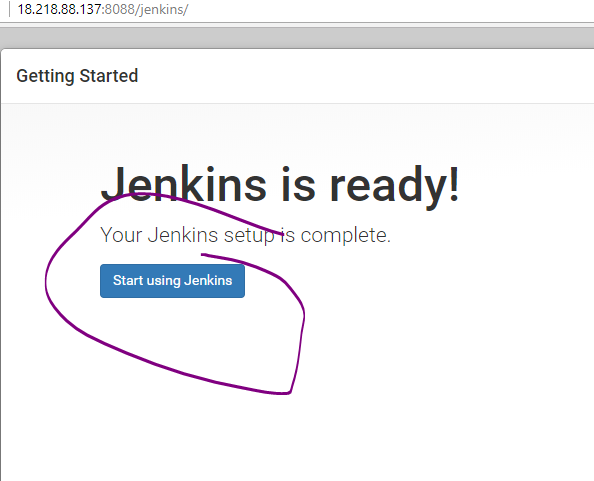
1. **Jenkins Start-up** & sample job configuration:



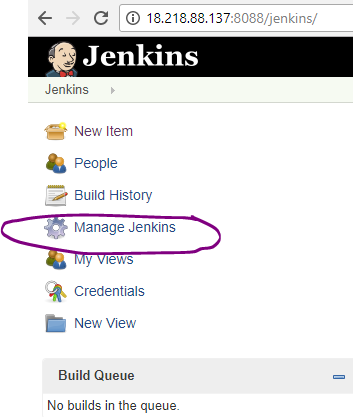


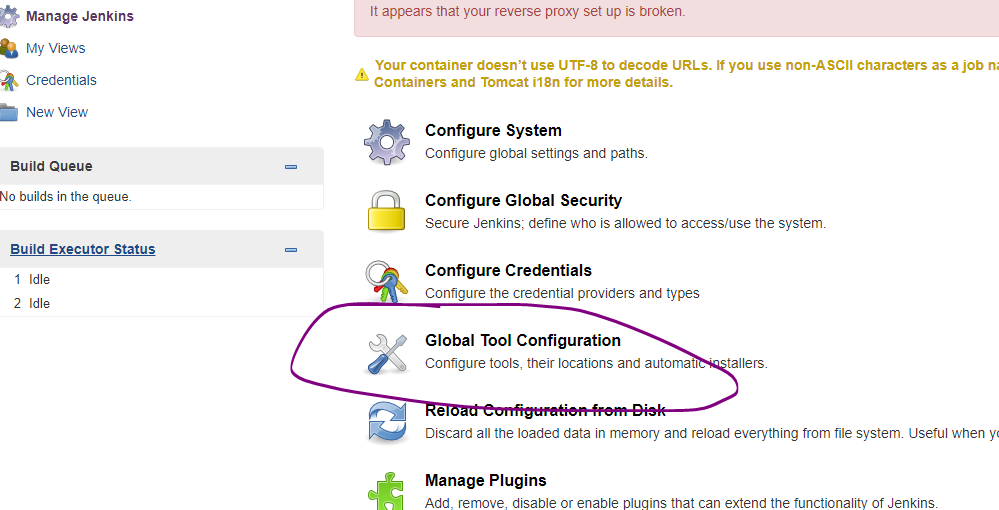




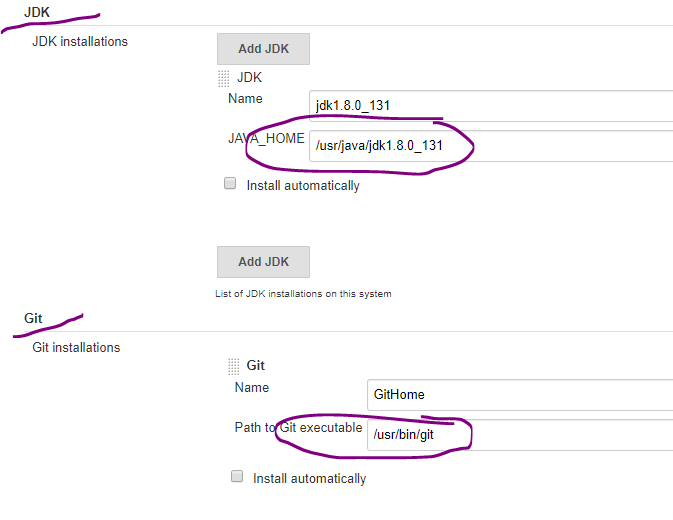


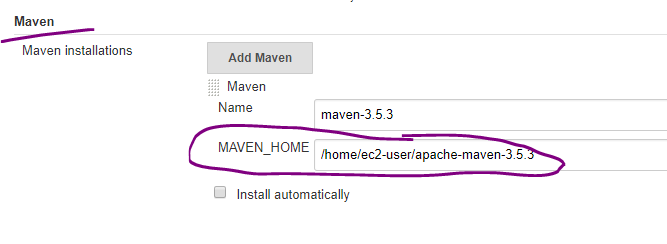
Global Tools configuration:

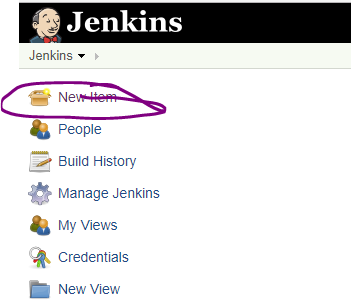


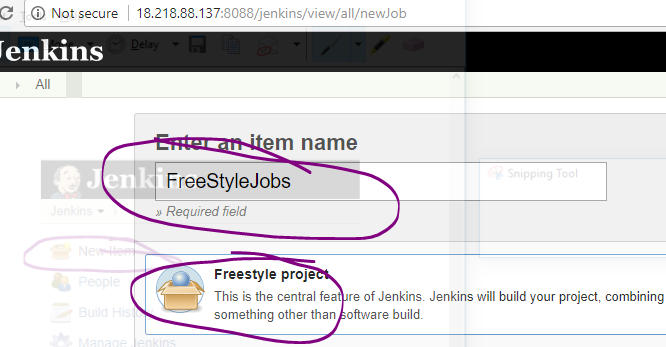


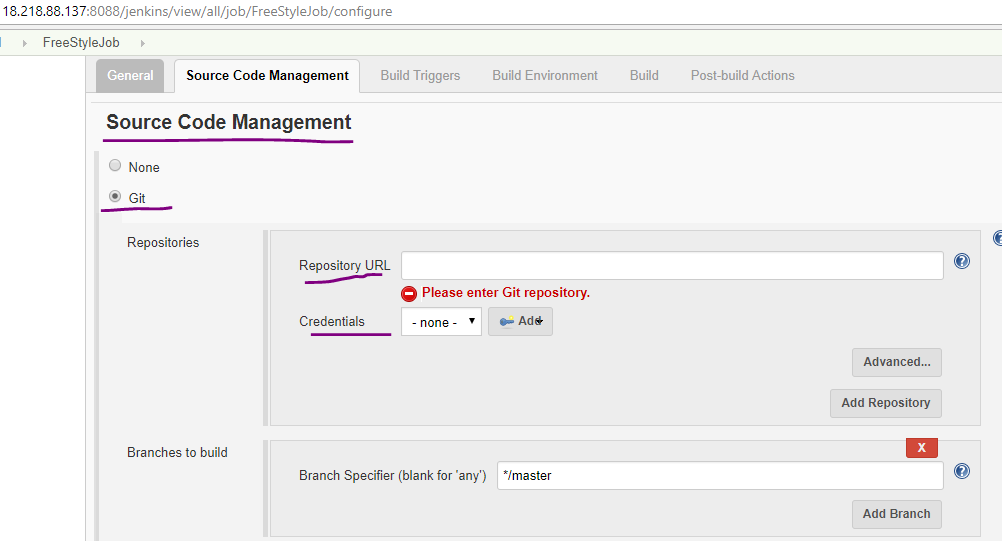
Add maven, java, git home paths

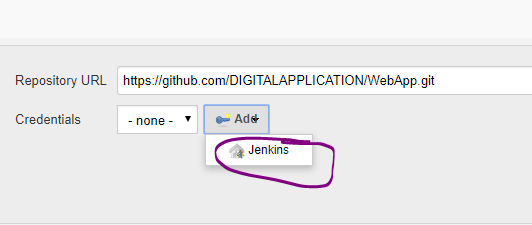


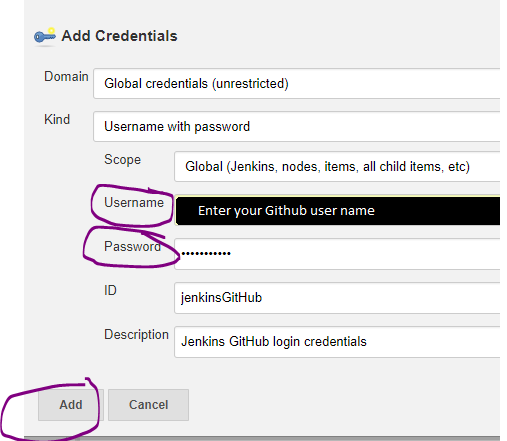




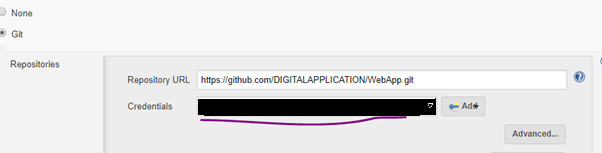




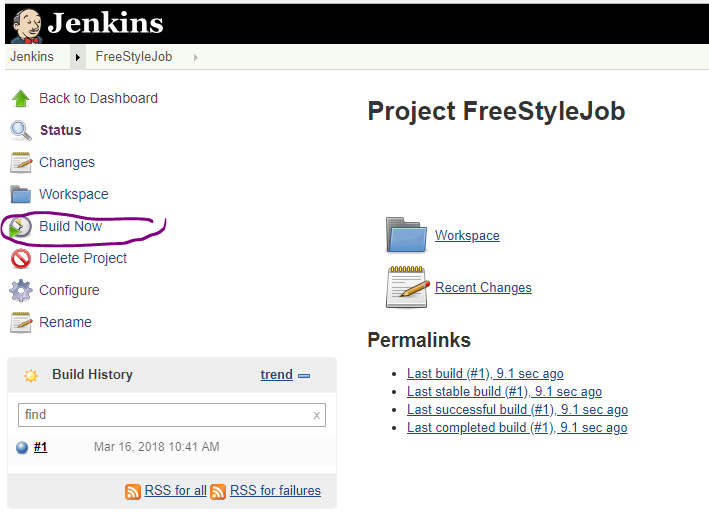


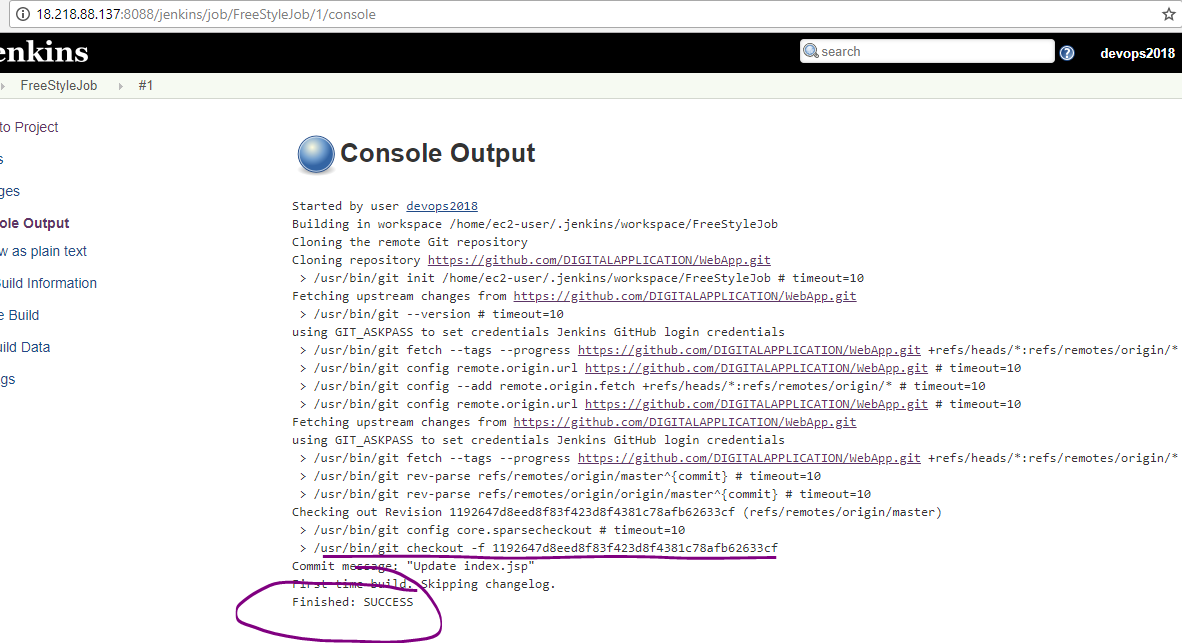


Select your account name in the credentials drop down box :



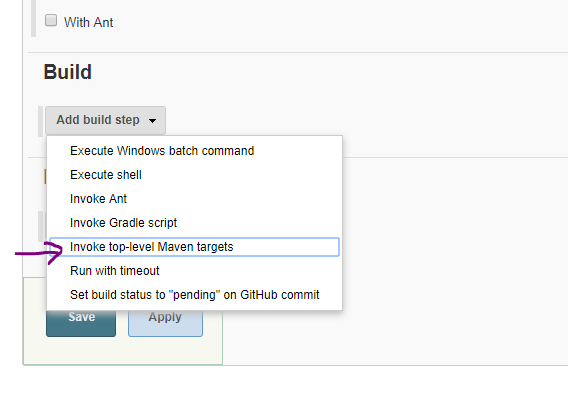
Check whether the code is being cloned / checkout or not.

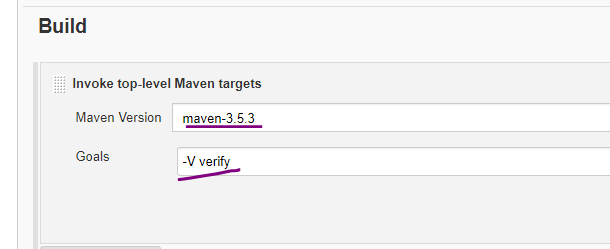




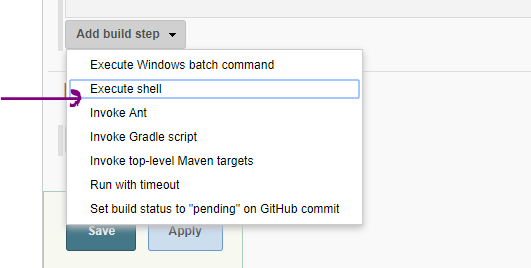
Go to created job and select configure.

**Build (i.e., compile the source code & package):**









**Deployment**:

#Deploy the war to tomcat server.

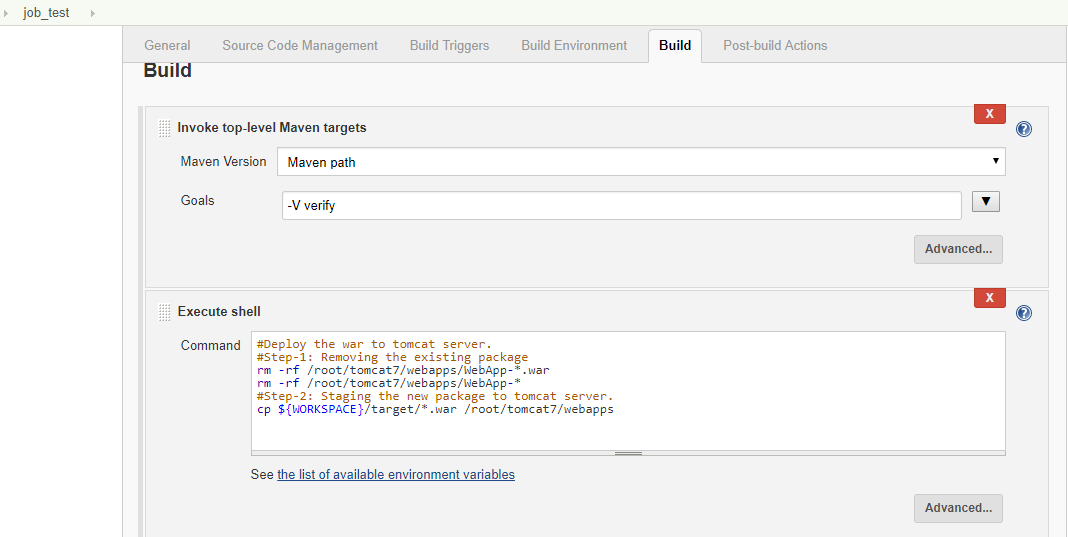
#Step-1: Removing the existing package

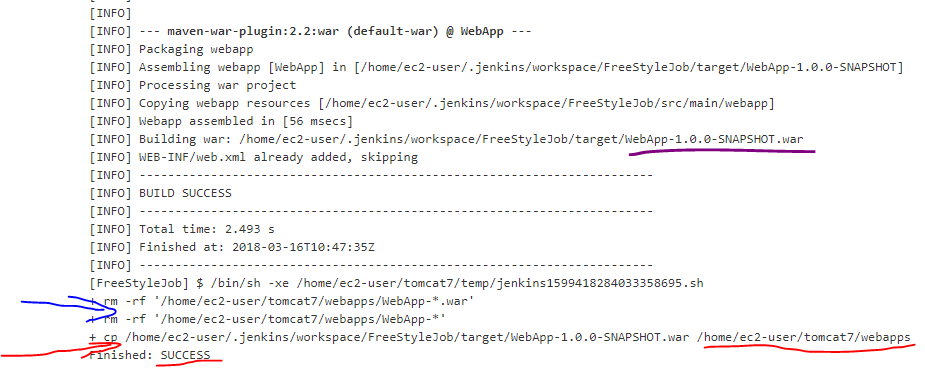
rm -rf /root/tomcat7/webapps/WebApp-\*.war

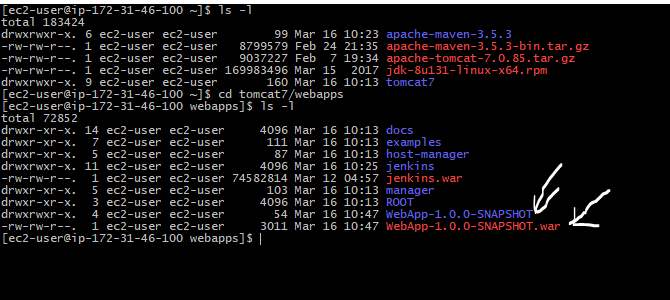
rm -rf /root/tomcat7/webapps/WebApp-\*

#Step-2: Staging the new package to tomcat server.

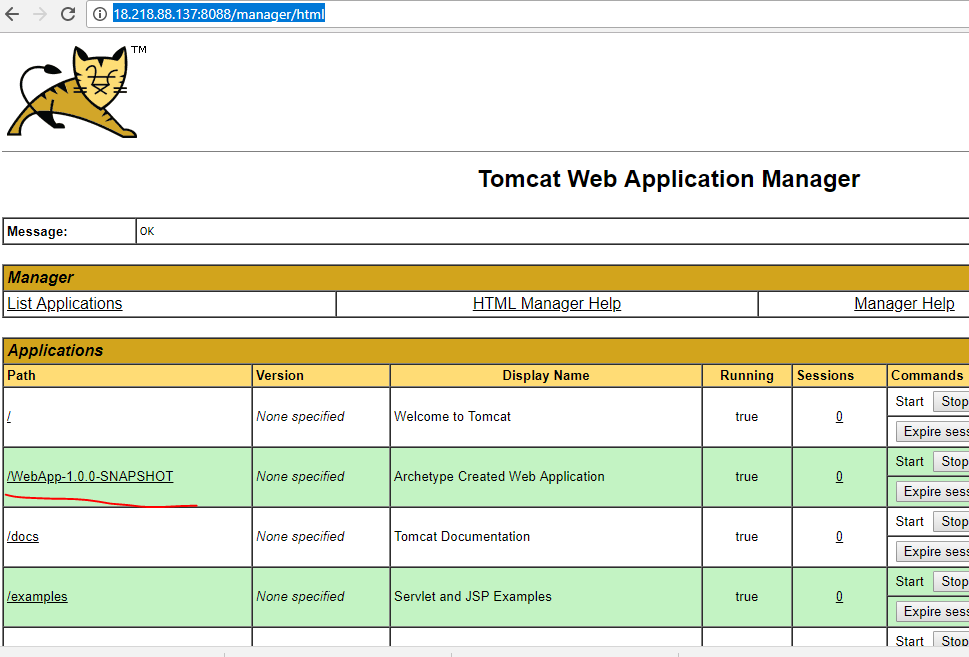
cp ${WORKSPACE}/target/\*.war /root/tomcat7/webapps

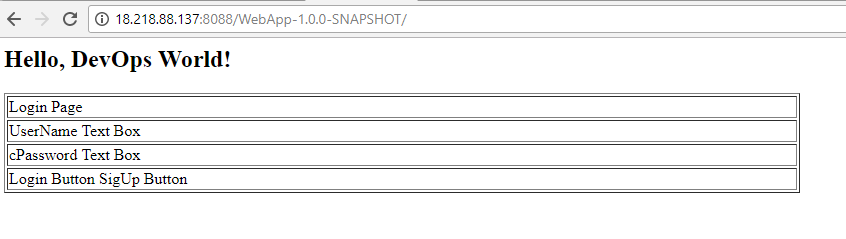






<http://18.218.88.137:8088/manager/html>





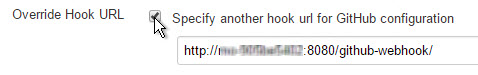
**Jenkins\_Configuation\_Build\_Weburl**

Configuring Jenkins to Run a Build Automatically on Code Push

Go to: Manage Jenkins > Configure System

Under GitHub Plugin Configuration, Click on ‘Advanced

Check ‘Specify another hook url for GitHub configuration’

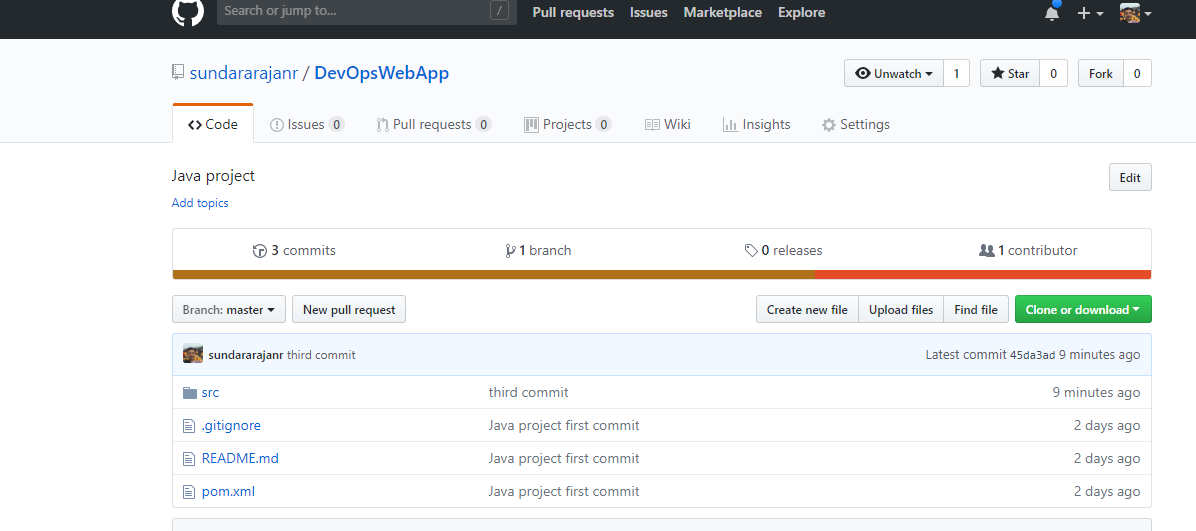
[](https://blogs.sap.com/wp-content/uploads/2015/12/img2_851355.jpg)

A textbox will appear with a hook URL. This is the Hook URL at which Jenkins will listen for POST requests. Copy this URL and enter into your Github webhook link.

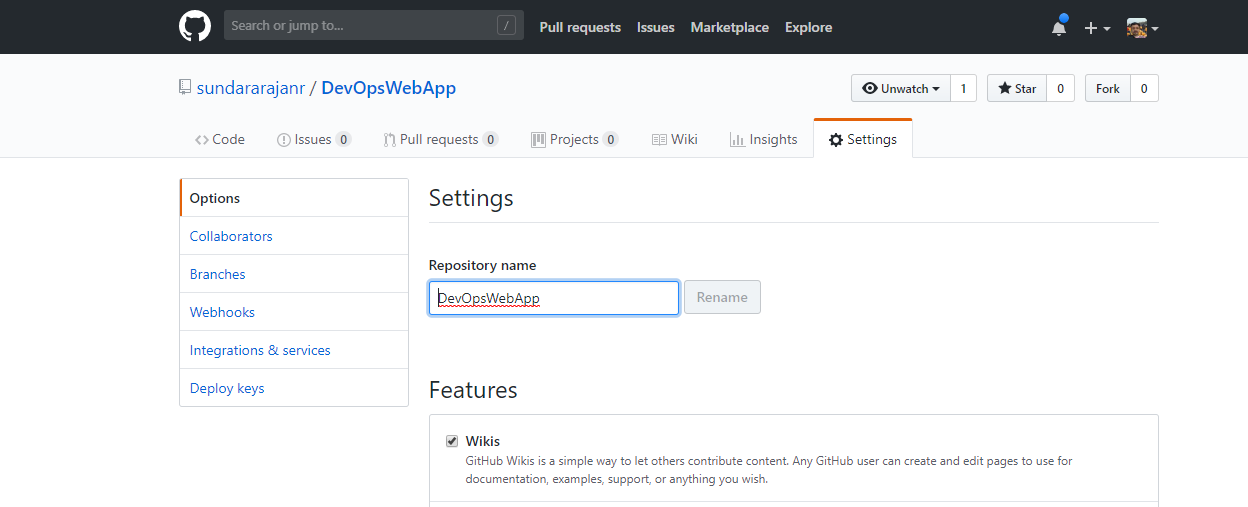
**Automatic build trigger using GitHub webbook**

Step 1: Login into GitHub and navigate to your project repository.

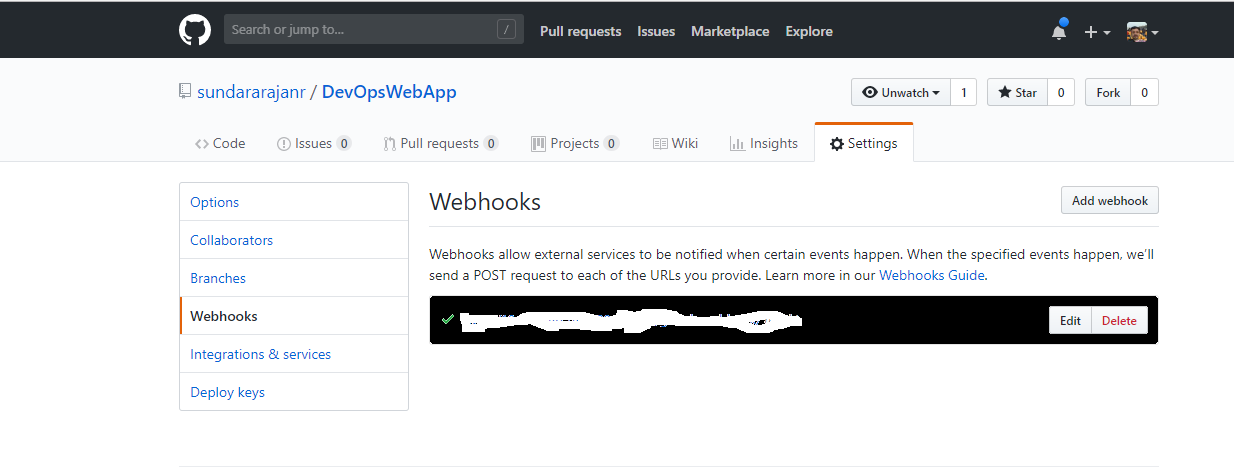
Step 2 : Click setting tab.



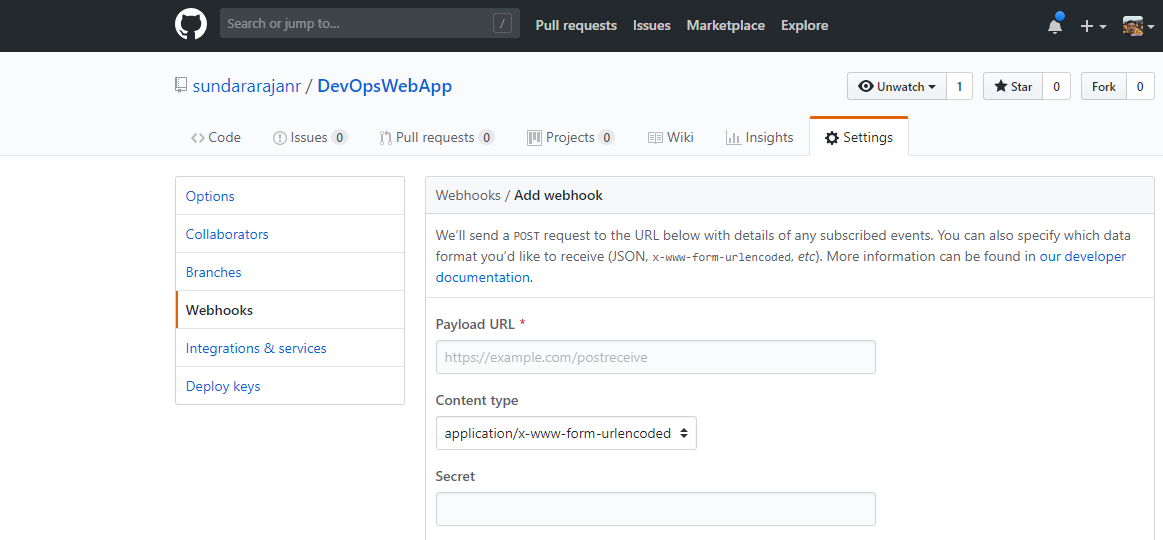
Step 3: you will able to see the settings tab now. Click Webhooks menu and it appears on left hand side.



Step 4 : Add webhook



Step 5 : Once you clicked the Add Webhook button it will display the below page. We have to enter the URL path of Jenkins. Refer the section Jenkins\_Configuation\_Build\_Weburl.



Now Go to “NexusSetup.sh” file and go through the section Note & Installation Procedure at the end of the file.