Jenkins Introduction:

SDLC: two major steps: development + build\_deploy

Developer will write the code --> Developer pushes the code to GitHub repository --> code is available on GitHub --> Compile and Build the code (we have Maven tools) --> Code quality will be done (we have SonarQube to check the code quality) --> Packaging our application (Java Jar or War) --> Docker image will be created --> Application is deployed using Kubernetes. DevOps team’s responsibility is to build that code and compile that code. We have to repeat this process in multiple environments. Also when there is a new commit into GitHub then the same process you got to follow. To avoid manual build and deployment, we have one tool called as Jenkins. Jenkins will take care of entire Build and Deployment. Jenkins will take code from Git, will use Maven for compilation (Compile sourceCode). Jenkins will talk to Maven tool

Developer pushes code into Repo

Jenkins

GitHub

SonarQube for code quality check

Maven compile source code

Kubernetes for deployment

Docker to create image

Package App

You could manually do the same process but it is error-prone. For every new commit, you have to follow the process again and again. To automate process of Build and Deployment, Jenkins comes into picture. Alternative for Jenkins are GitLab and GitHub actions.

Jenkins is CI/CD tool, CI refers to Continuous Integration, CD refers to Continuous Deployment.

It is free and open-source software/tool, which is developed using Java language. CI/CD is an approach to automate project: build & deployment process.

Using Jenkins, we can deploy any type of projects, regardless of tech stacks used (Java, Python, .Net, React)

Software development involves many steps out of which two major steps are writing code and pushing code into GitHub then building and deploying that application.

To automate Build and Deployment process of our application, we can use tools such as GitLab, Jenkins

Build and Deployment process of an Application:

-> Taking latest code from Repo such as GitHub or BitBucket

-> Compile and Build source code using tools such as Maven

-> Perform code reviews using SonarQube

-> Upload project artifact using Nexus

-> Create a Docker image with Docker

-> Deploy Code/App into Server using K8s

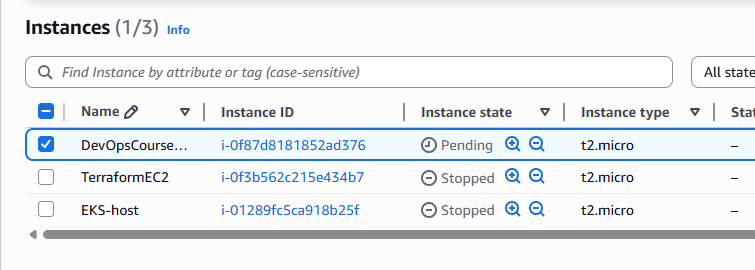
Note this process can be done manually, however it is a time-consuming process and manual process of build-deployment could lead to error hence to overcome limitation associated with manual build-deployment process we have tools such as Jenkins, GitLab.

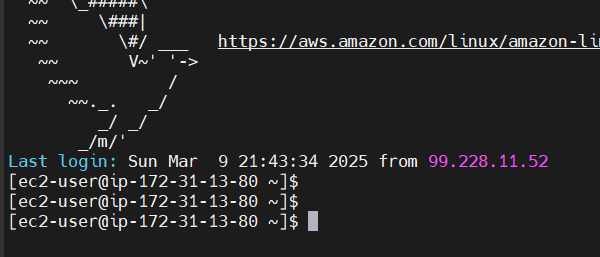
To automate the Build-Deployment process of our application we can use tools such as GitLab, Jenkins

Jenkins for Windows follow this link: https://youtu.be/1LE1llhafOE?si=jvOZYSfcNLvK3lay

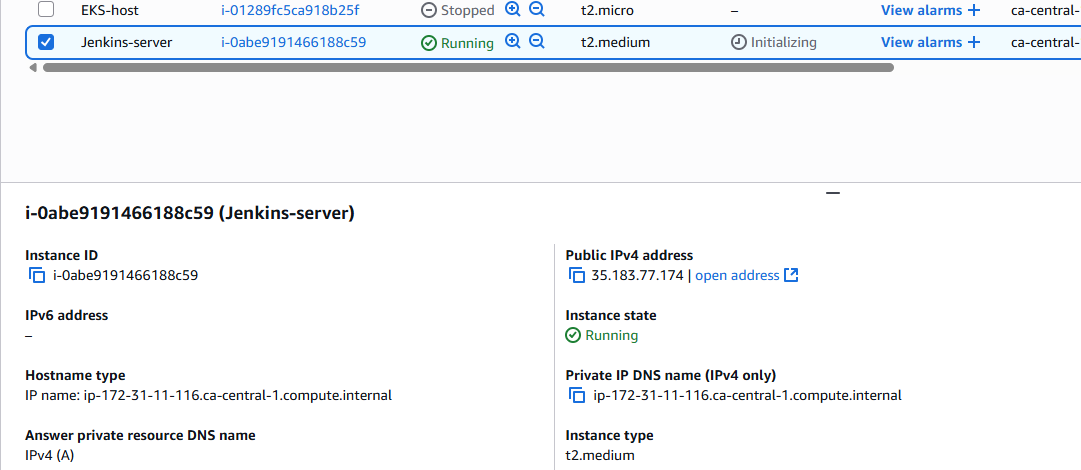
Jenkins installation setup in Linux VM

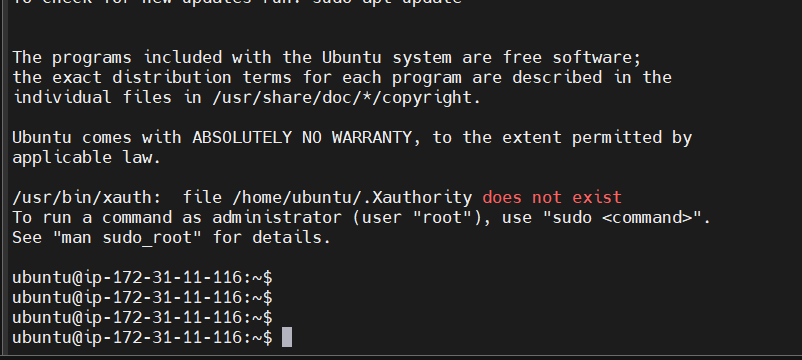
Get an instance



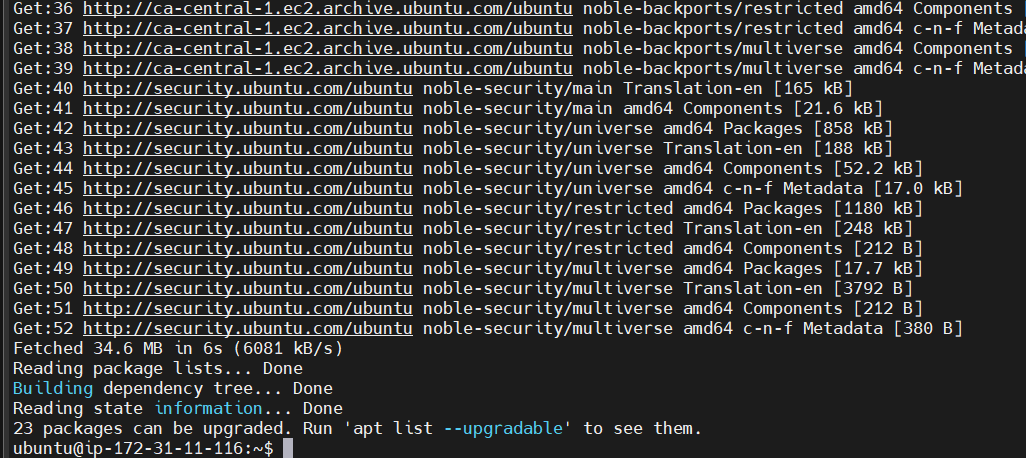


t2.medium is recommended for Jenkins





ubuntu@ip-172-31-11-116:~$ sudo apt update



To run Jenkins, which software is required: Java

ubuntu@ip-172-31-11-116:~$ sudo apt install default-jdk

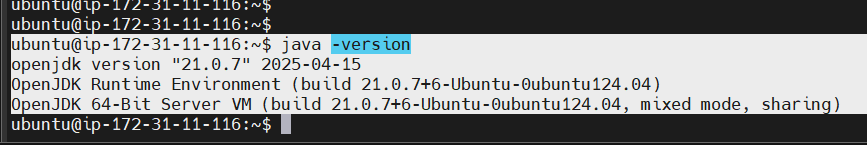


ubuntu@ip-172-31-11-116:~$ java -version

openjdk version "21.0.7" 2025-04-15

OpenJDK Runtime Environment (build 21.0.7+6-Ubuntu-0ubuntu124.04)

OpenJDK 64-Bit Server VM (build 21.0.7+6-Ubuntu-0ubuntu124.04, mixed mode, sharing)



1. Create Linux VM on AWS Cloud - Ubuntu (preferred to use t2.medium as instance type)

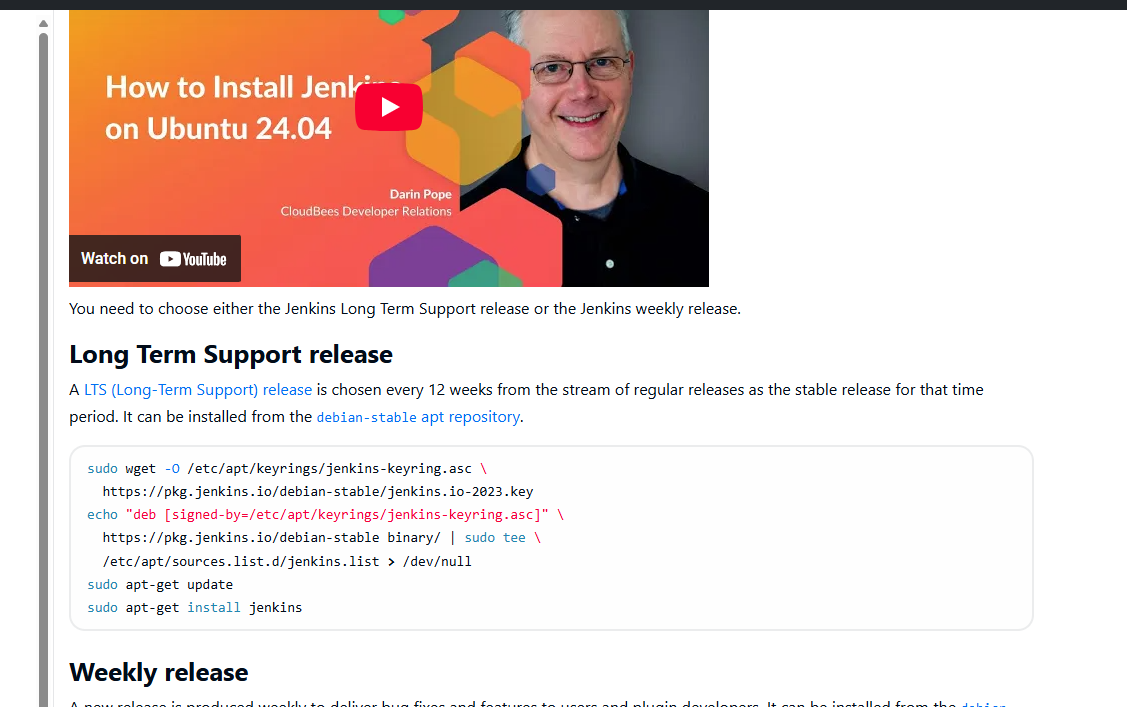
Get connected to Linux VM using SSH Gitbash, Terminal or any medium

1. Install Java
   1. ubuntu@ip-172-31-11-116:~$ sudo apt update
   2. ubuntu@ip-172-31-11-116:~$ sudo apt install default-jdk
   3. ubuntu@ip-172-31-11-116:~$ java -version

Go to <https://www.jenkins.io/>

<https://www.jenkins.io/download/>

<https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>



1. Install Jenkins

sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

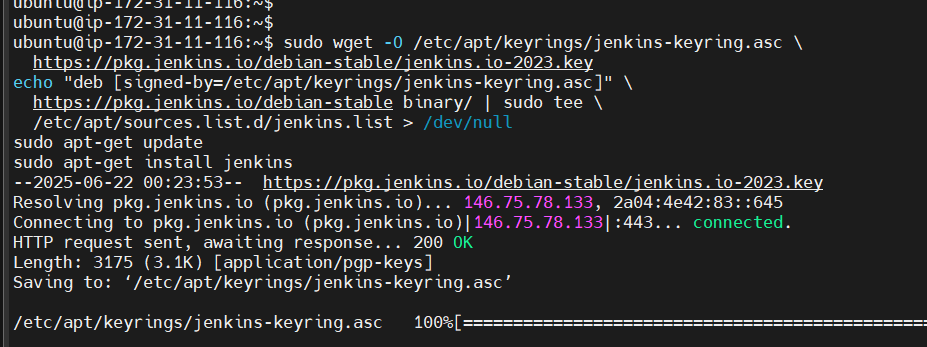
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins



1. Start and verify Jenkins

sudo systemctl enable jenkins

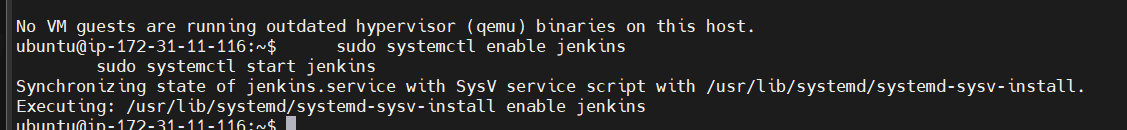
sudo systemctl start jenkins

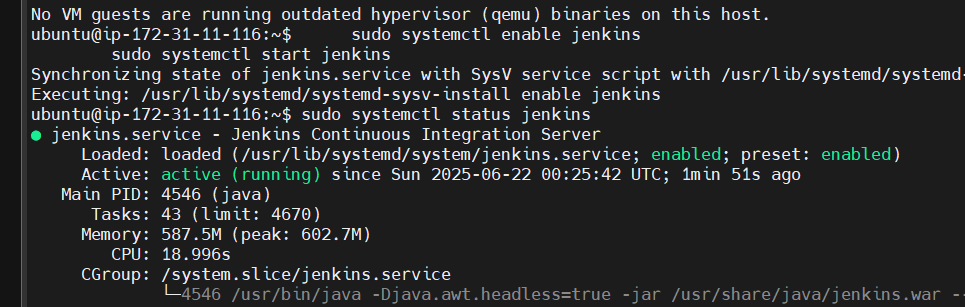
1. Verify Jenkins

sudo systemctl status jenkins

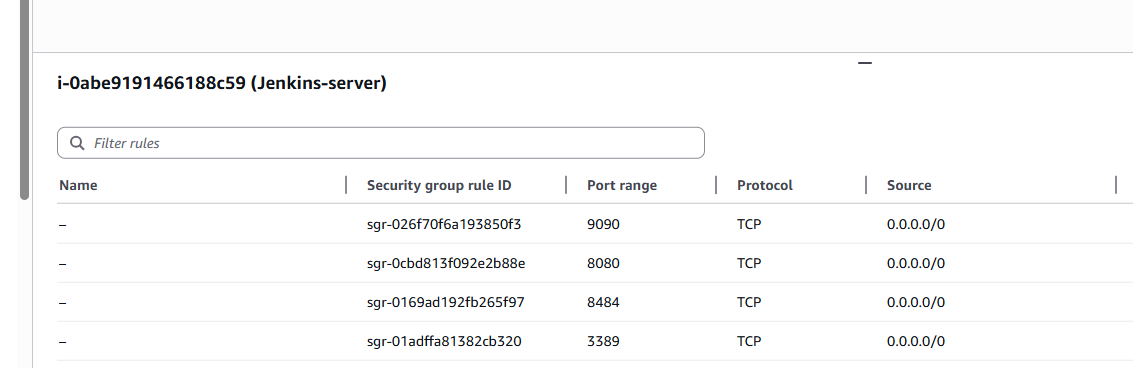
1. Open Jenkins server in browser (also make sure edit inbound rules and add 8080 in Security group)
   1. <http://public-ip:8080/>
2. Copy Jenkins admin password

/var/lib/jenkins/secrets/initialAdminPassword

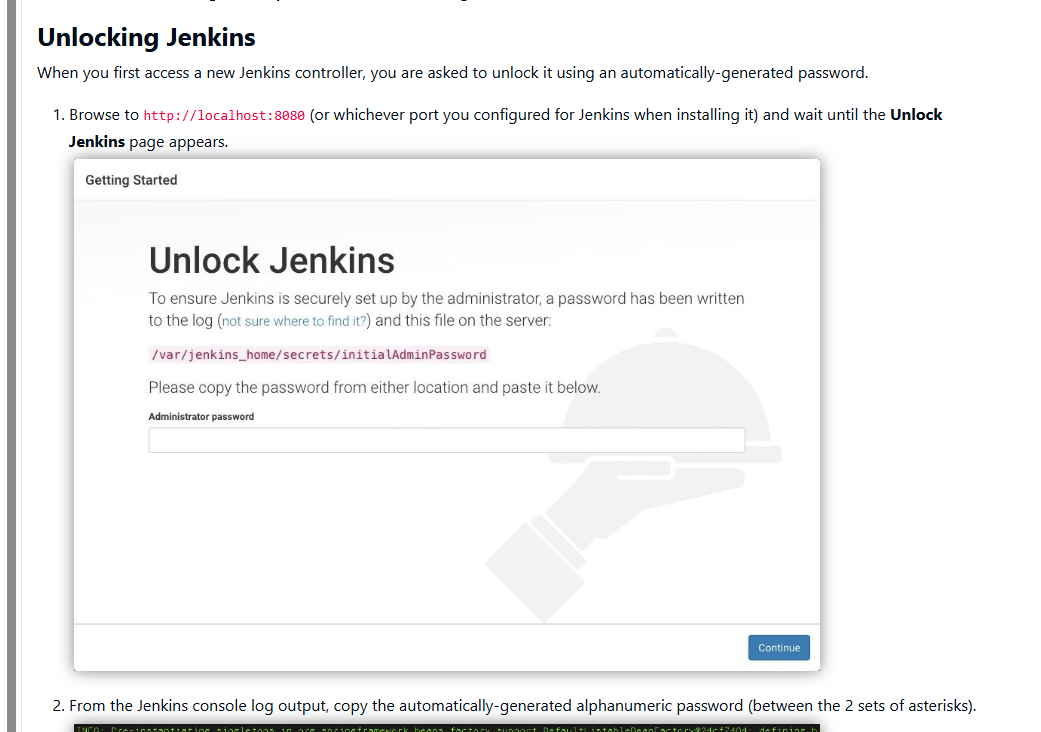




Make sure 8080 is enabled in the Security group



Scroll down copy the command

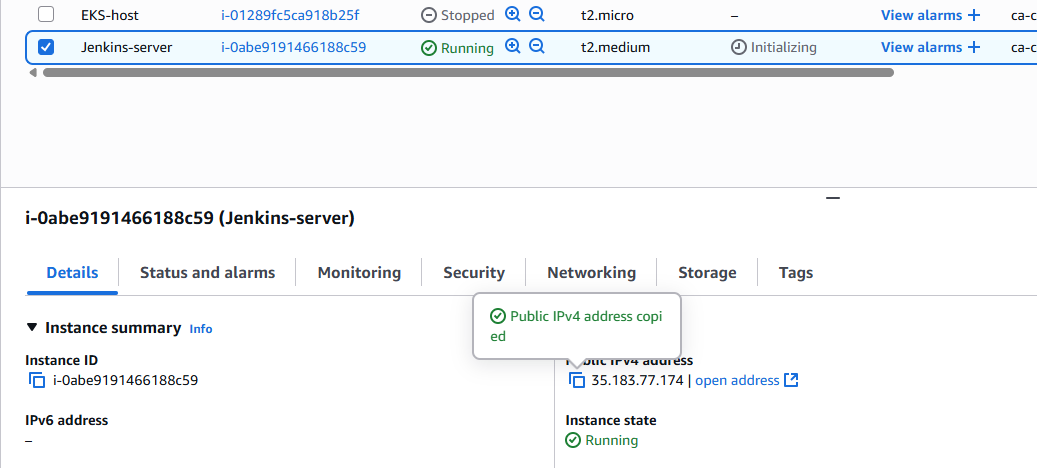


The command: sudo cat /var/lib/jenkins/secrets/initialAdminPassword

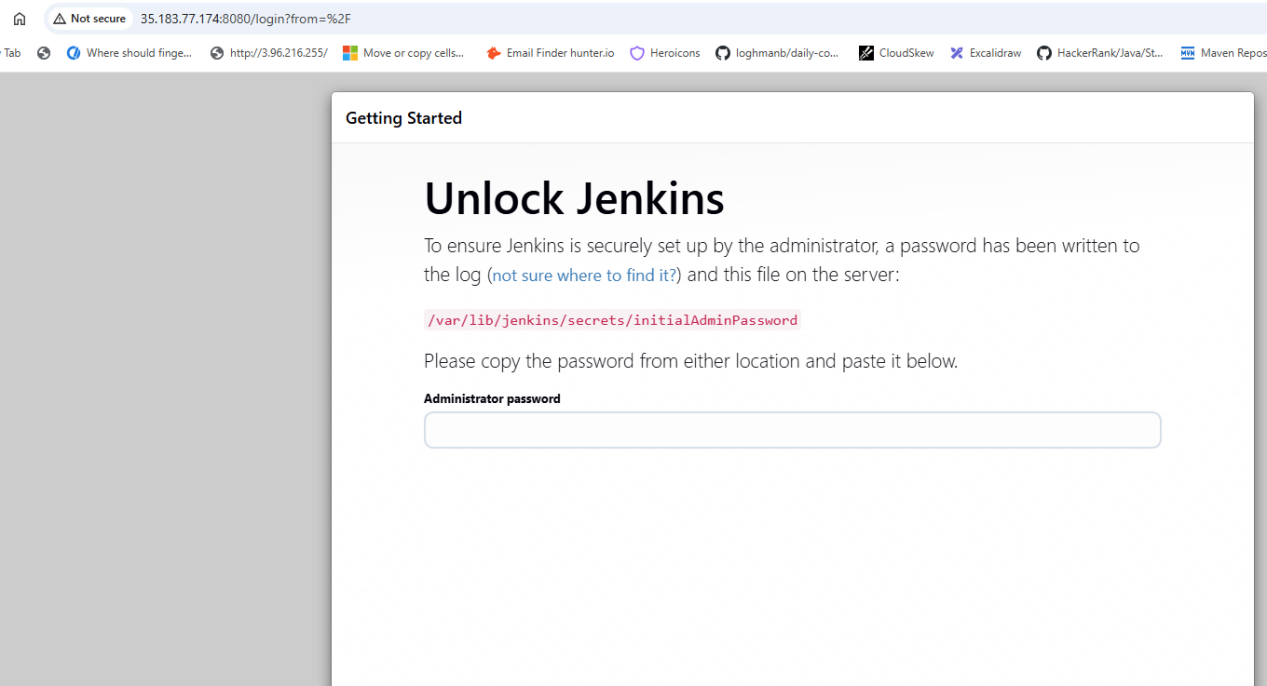
ubuntu@ip-172-31-11-116:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

cat: /var/lib/jenkins/secrets/initialAdminPassword : No such file or directory

ubuntu@ip-172-31-11-116:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

XXXXXXXXXXXXXXXXXXXXXXXXXXX

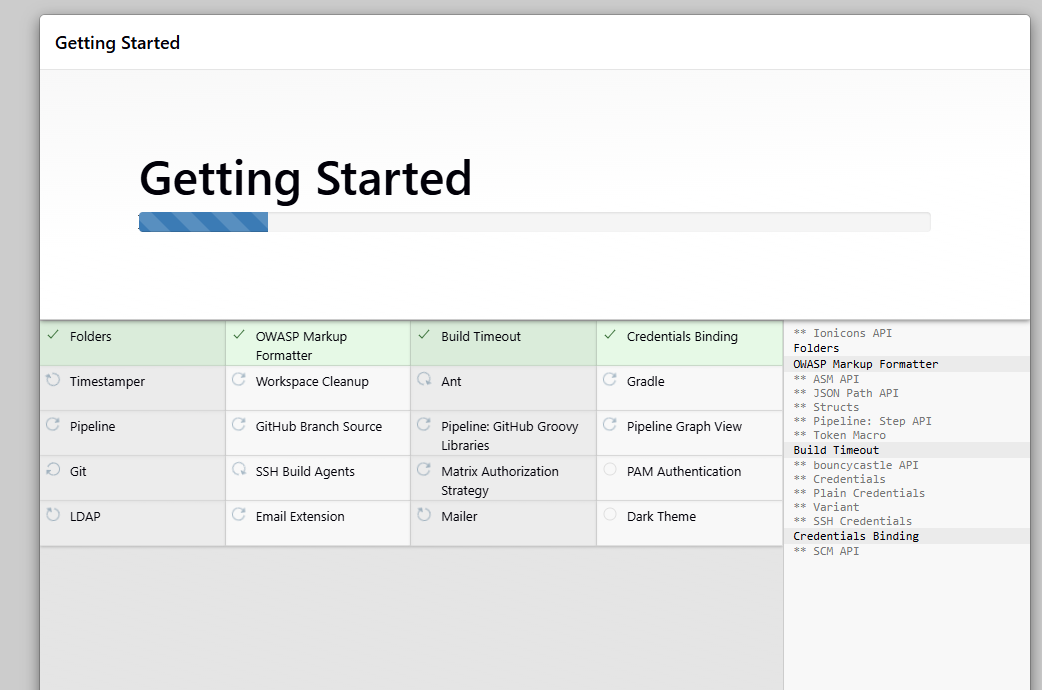
http://35.183.77.174:8080/login?from=%2F

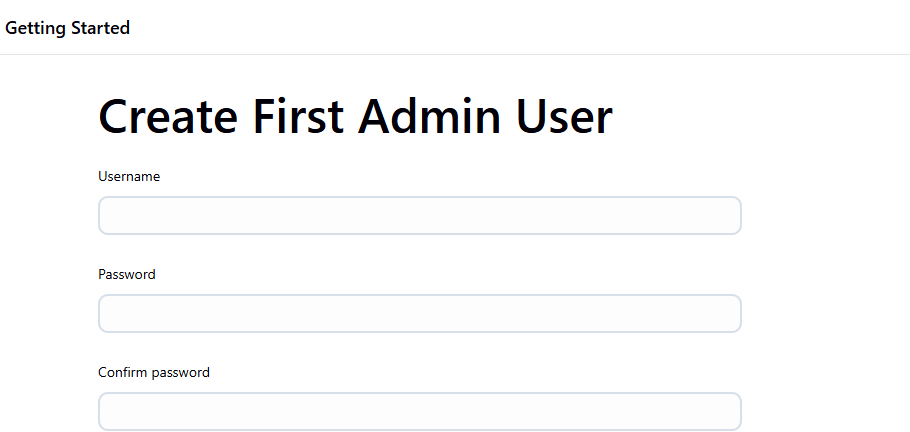


Copy paste password: XXXXXXXXXXXXXXXXXXX

Select Install Plugins

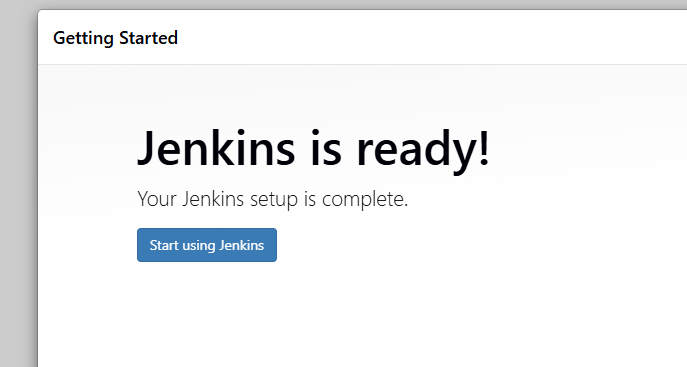
Click Install button in User Management and Security





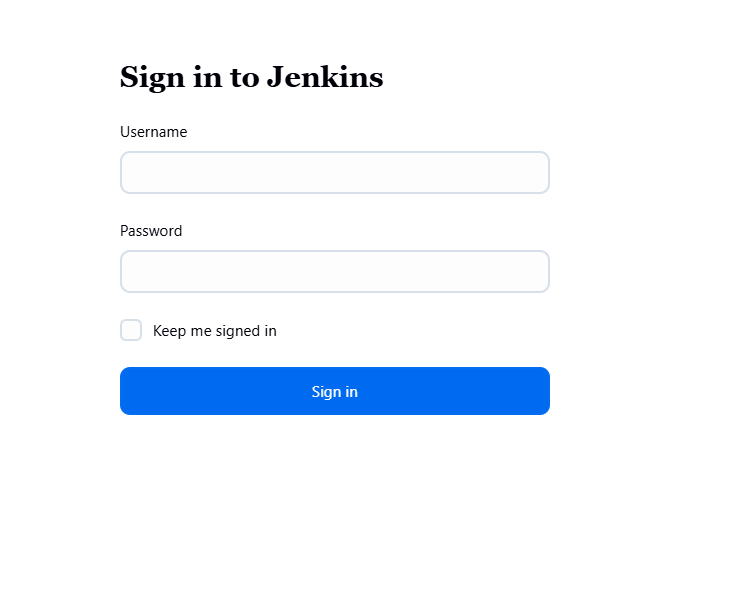
Jenkins URL: <http://35.183.77.174:8080/>

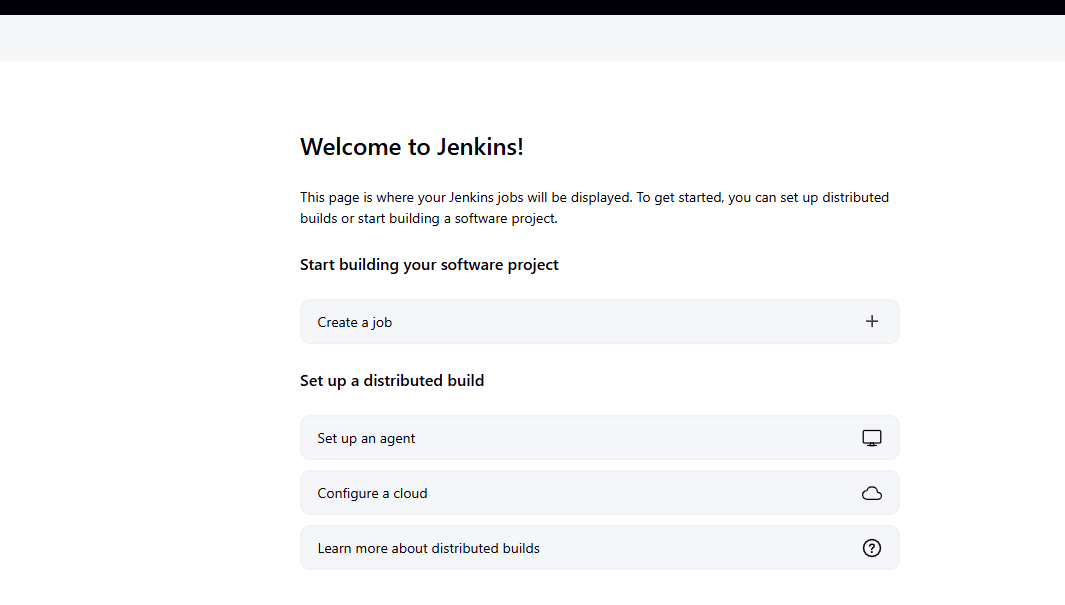
Click Save and Finish



Start using Jenkins

Click Logout





What’s Jobs in Jenkins?

It refers to set of steps that are assigned to Jenkins to perform task

-> Taking code from Repo such as GitHub

-> Perform Maven build

-> Build Docker image

-> Deploy app/docker image in K8s cluster

Creating First Job

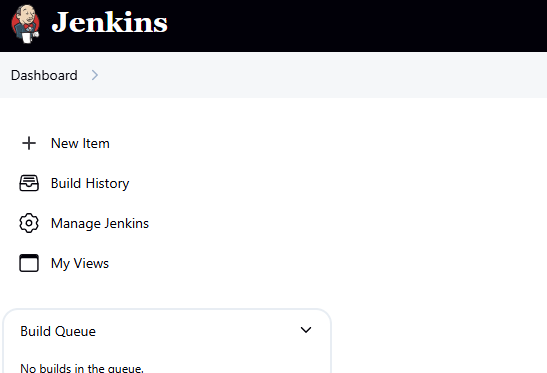
-> Go to Jenkins dashboard

-> Click on New Item -> Enter the Name -> Select Freestyle -> Enter some description --> add build step -> select execute shell -> enter shell script -> echo “Demo First Job” -> touch alien.txt -> echo “Demo first job completed”

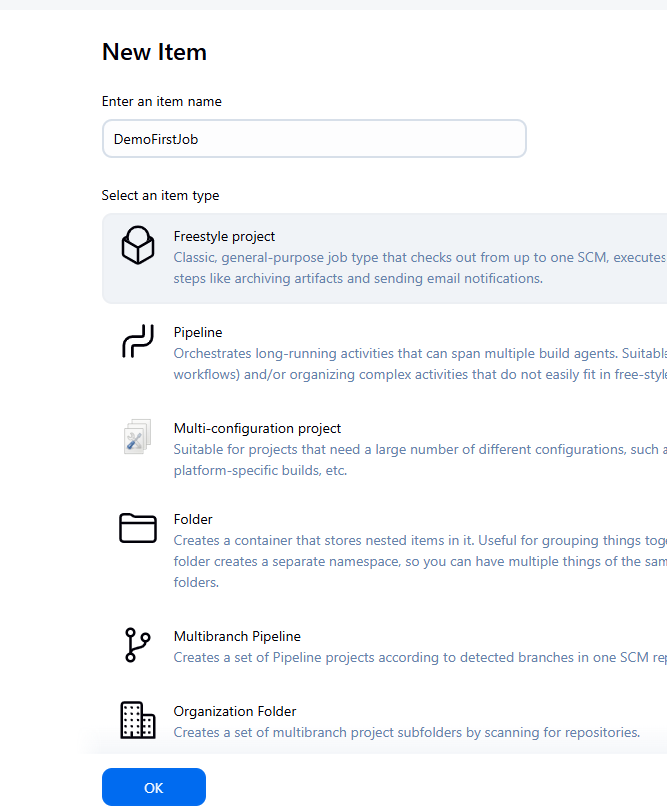
-> Apply and Save

See the job in Dashboard and click that job --> Build now --> Console output to see the job execution details

-> cd /var/lib/jenkins/workspace/DemoFirstJob

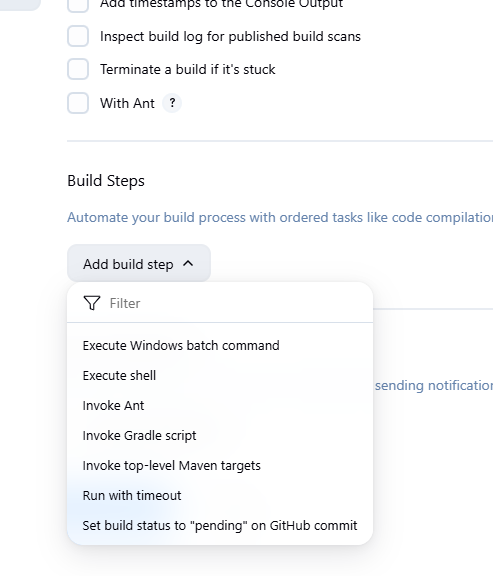


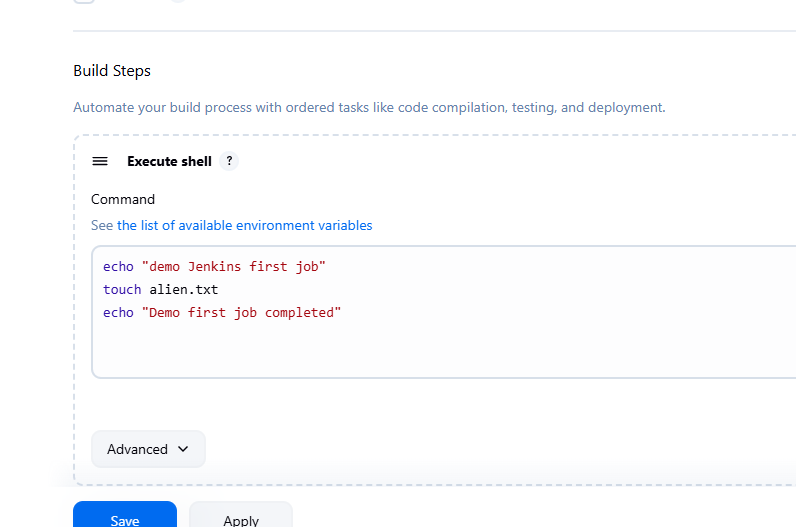
Click on New Item



Click on Freestyle project then Ok

In Build steps click Execute shell



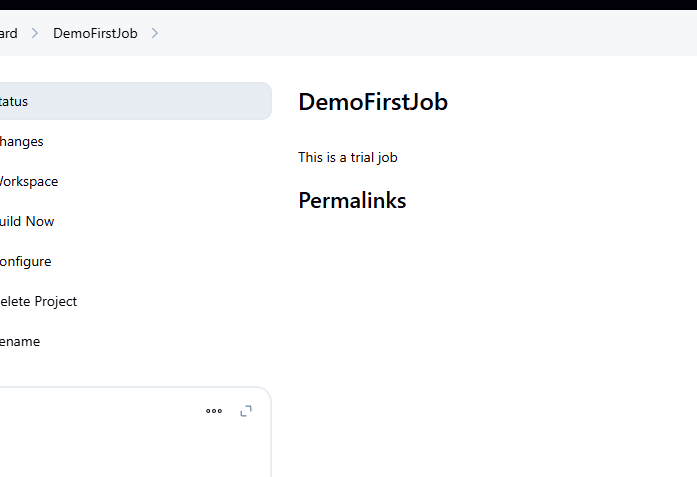


echo "demo Jenkins first job"

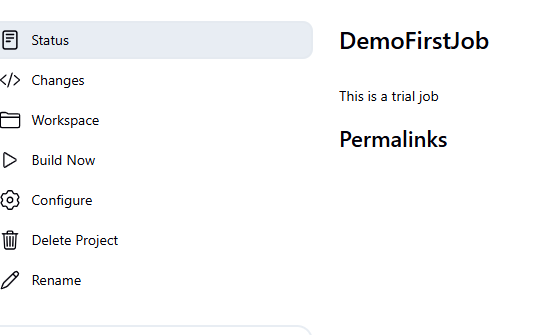
touch alien.txt

echo "Demo first job completed"

Click Apply then Save

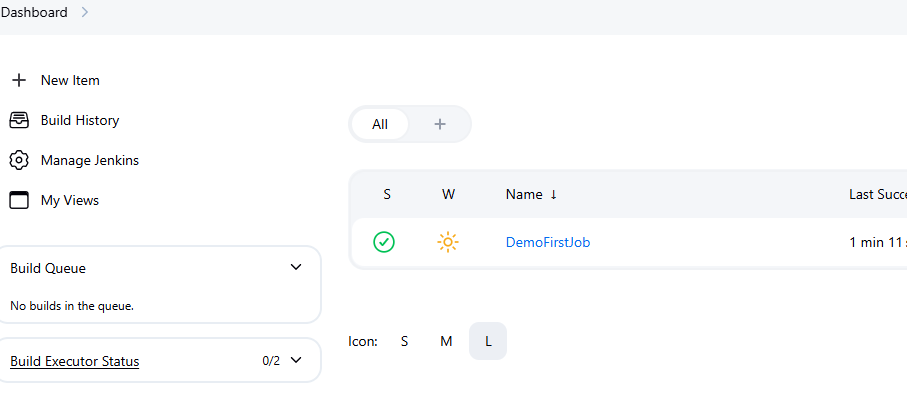


Click on Build Now

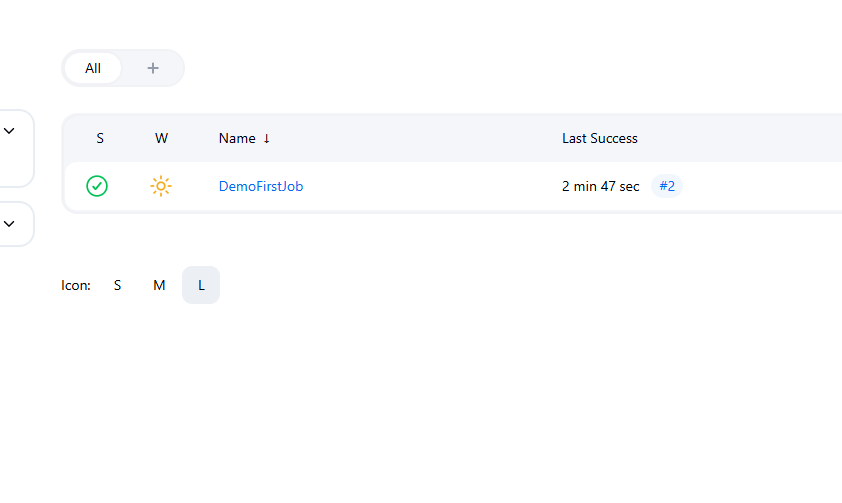


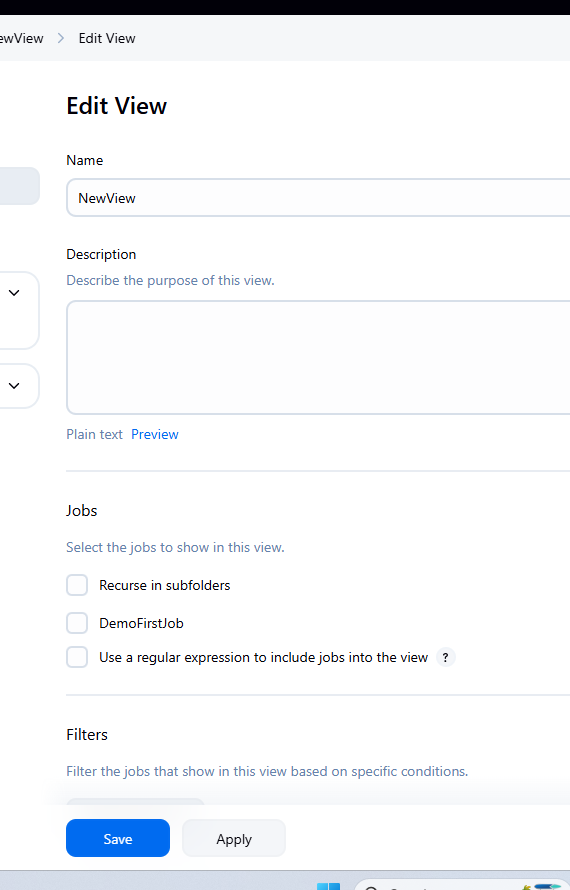
Click Build Now

In the Dashboard we have this



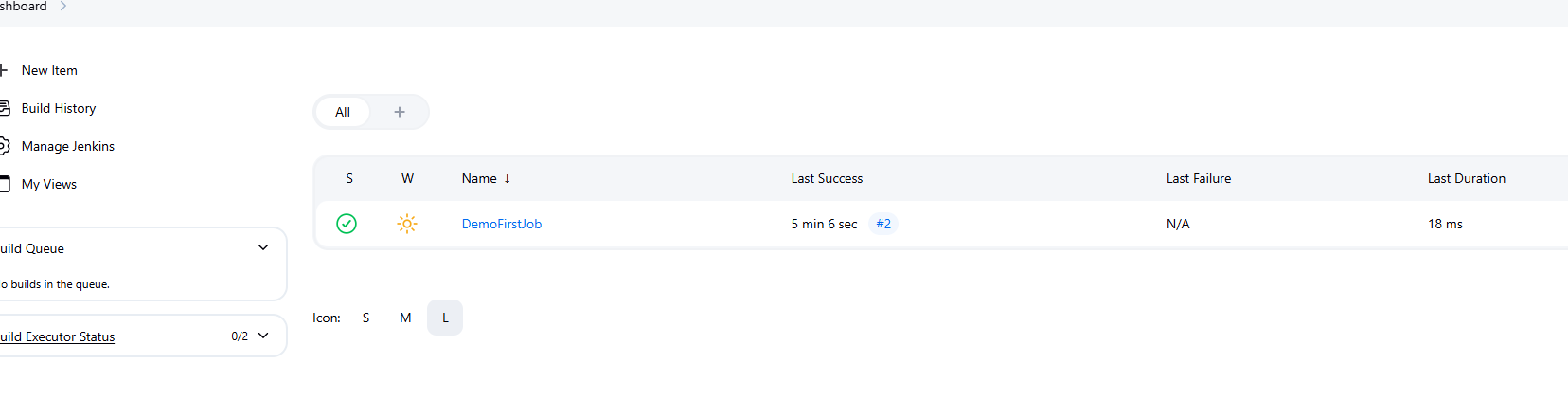
Click on the + button





We can add new Columns also

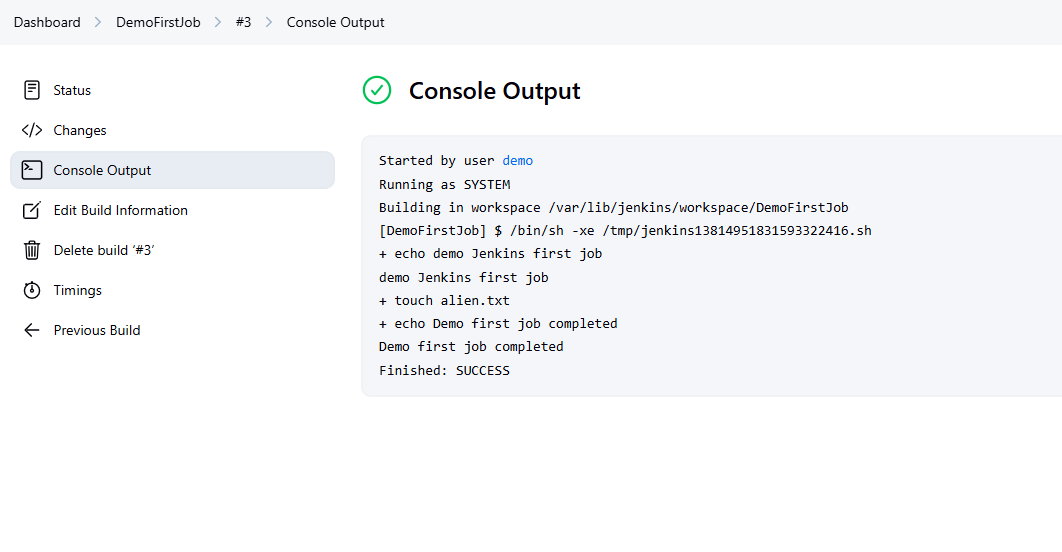
This is the default view: shows columns Name, Last Success, Last Failure, Last Duration



Once you click Build Now



Click on Console Output



We can see Demo first job completed

The file is also created in this workspace

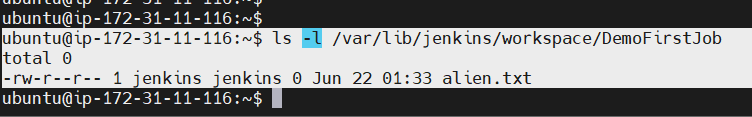
Building in workspace /var/lib/jenkins/workspace/DemoFirstJob

Copy paste this location

ubuntu@ip-172-31-11-116:~$ ls -l /var/lib/jenkins/workspace/DemoFirstJob

total 0

-rw-r--r-- 1 jenkins jenkins 0 Jun 22 01:33 alien.txt



Jenkins job with GitHub repo + Maven

Jenkins job with GitHub repo + Maven + Tomcat server

Jenkins job with GitHub Repo + Maven

1. Install Git client in Jenkins server (a Linux VM where Jenkins is installed)
2. Configuring Maven tool (Jenkins dashboard -> Manage Jenkins -> Tools -> Add Maven
3. Create a Jenkins job
   1. New item : enter name -> select freestyle job -> enter description -> Select Git (source code management)
   2. Enter the GitHub project link (https://github.com/Haider7214/SpringApp.git) -> Build steps -> Invoke top level Maven target

Maven Version

Goals

clean compile test package

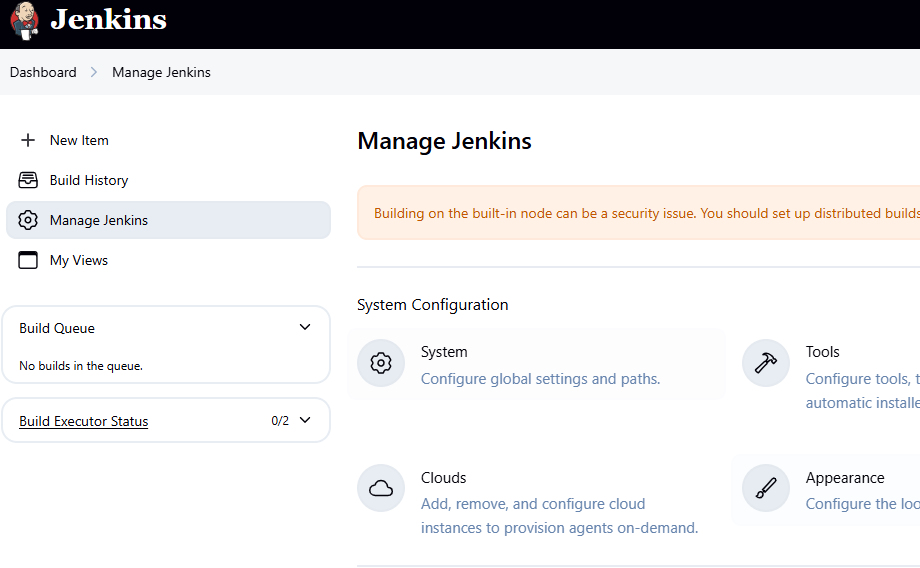
--> apply and save

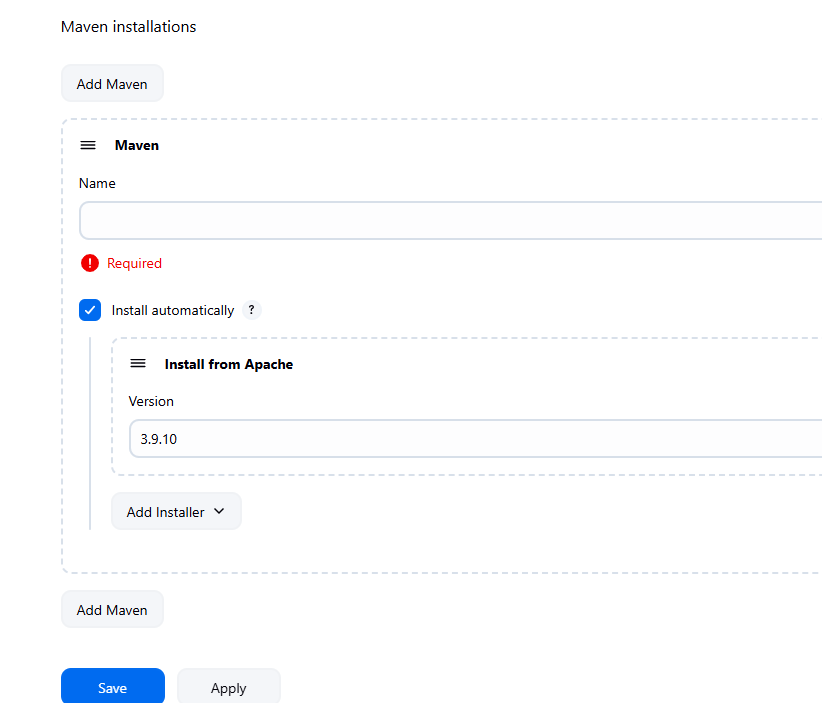
--> Jenkins Dashboard --> Get into job --> build now --> Jenkins home directory : cd /var/lib/Jenkins/workspace

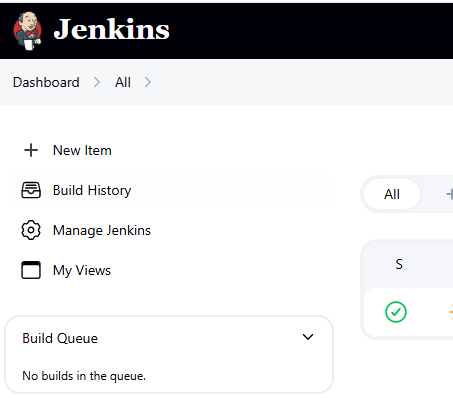
Verify the target folder within the job and you can notice war file is created

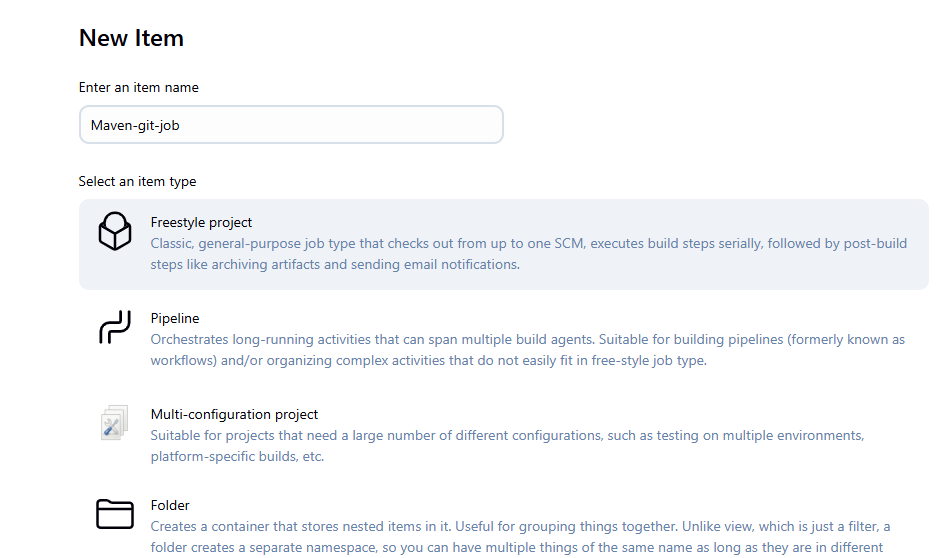
sudo apt install git -y

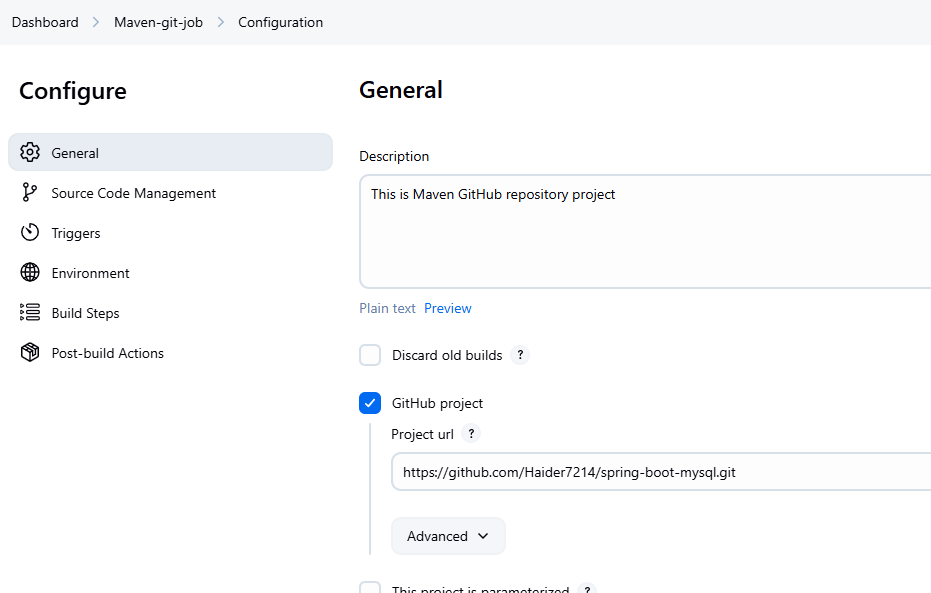
Click on Manage Jenkins

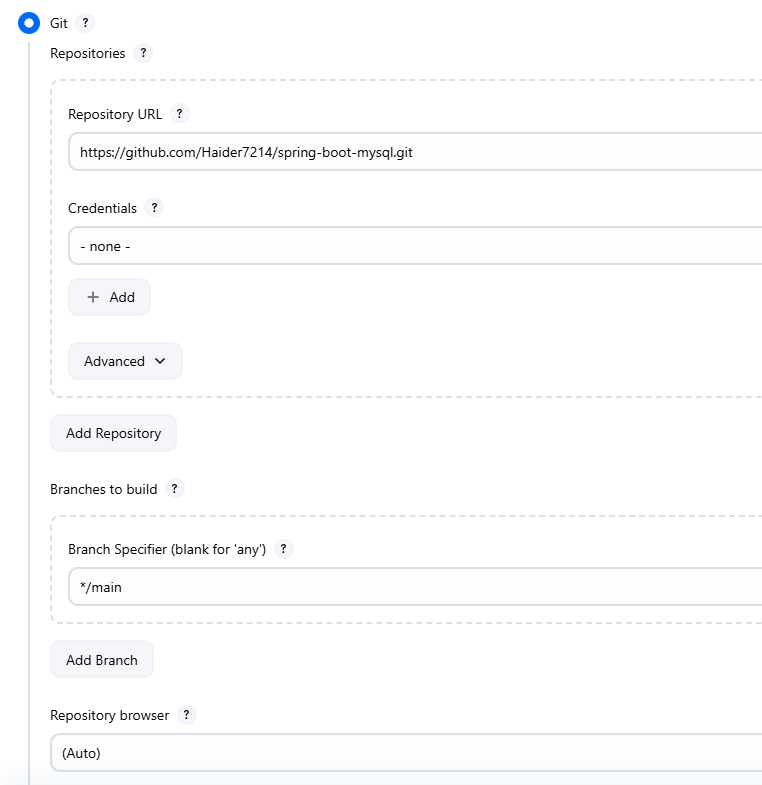


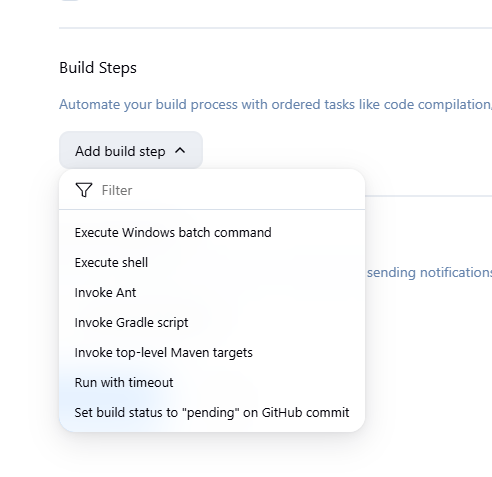


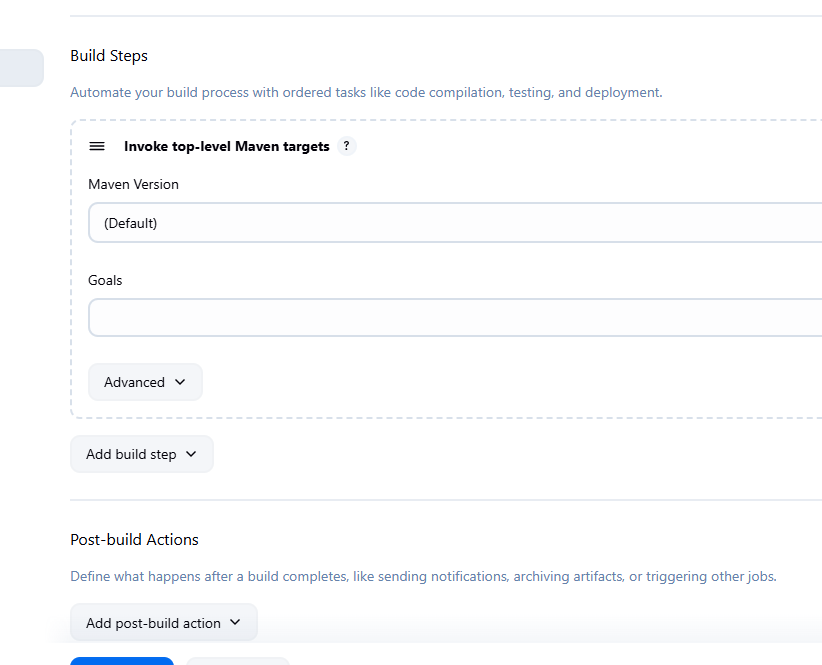


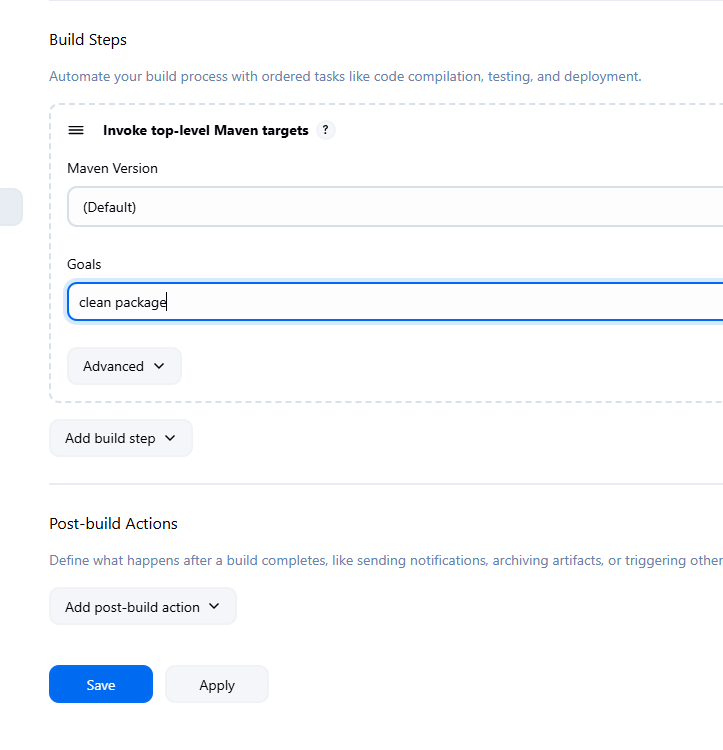




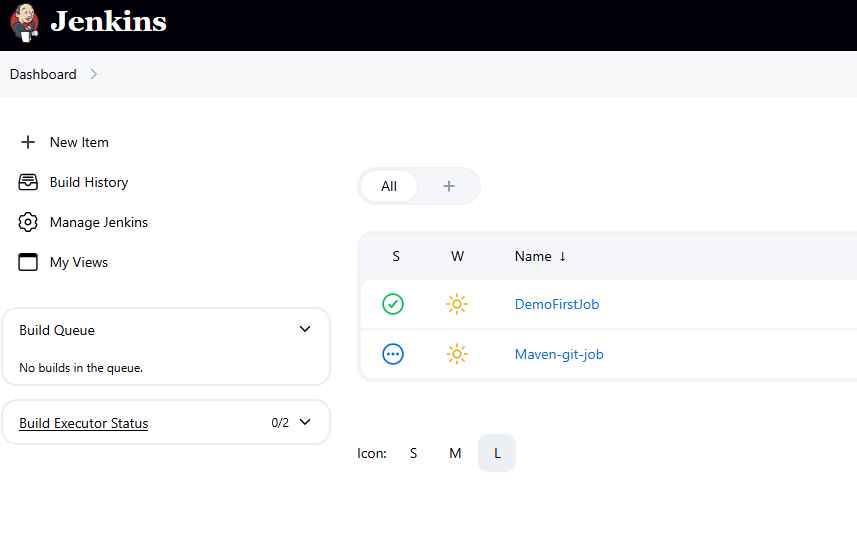








Click Apply and Save



ubuntu@ip-172-31-11-116:~$ cd /var/lib/jenkins/workspace/

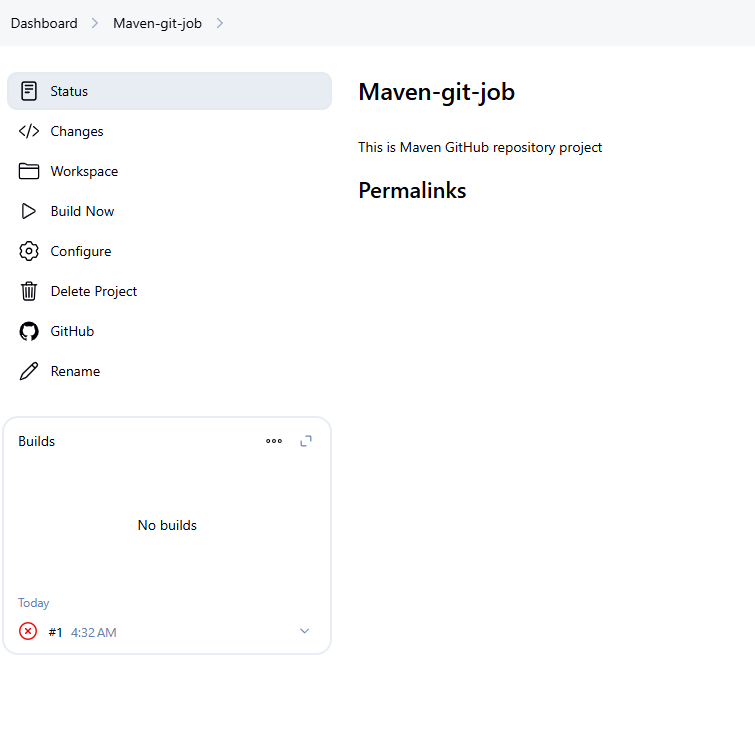
ubuntu@ip-172-31-11-116:/var/lib/jenkins/workspace$ ls

DemoFirstJob

Only one job we executed that’s present here

1:30

We click Build Now and it failed



sudo systemctl enable jenkins

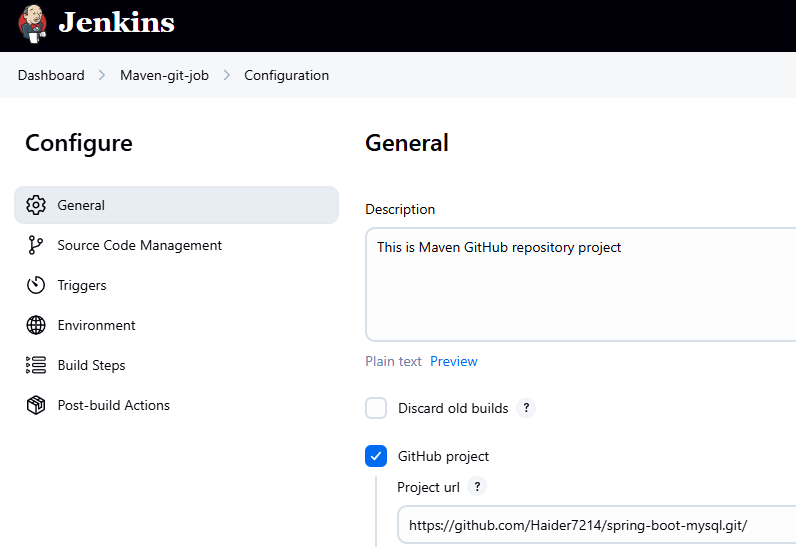
sudo systemctl start jenkins

<http://3.99.132.101:8080/login?from=%2Fjob%2FMaven-git-job%2F>

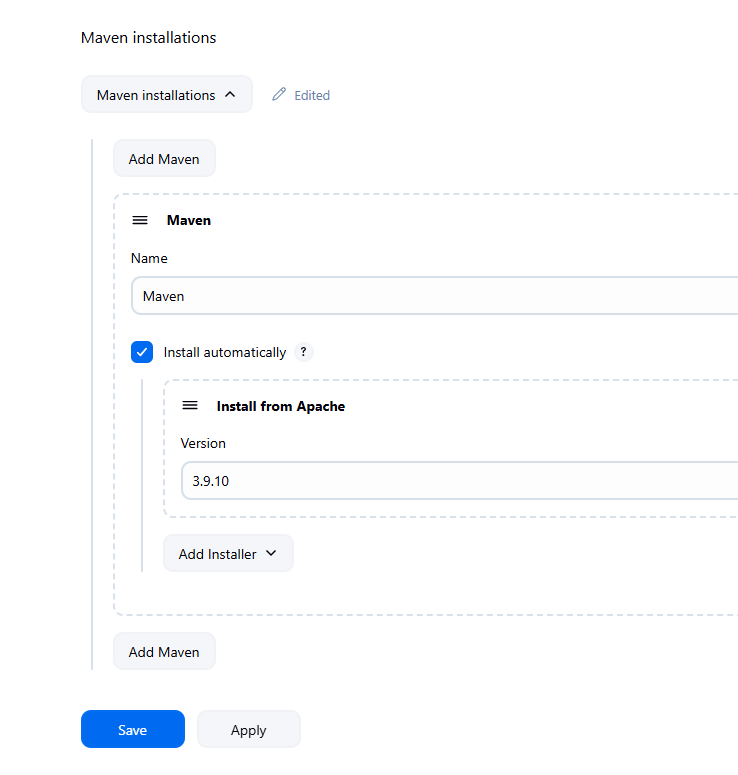
Console output

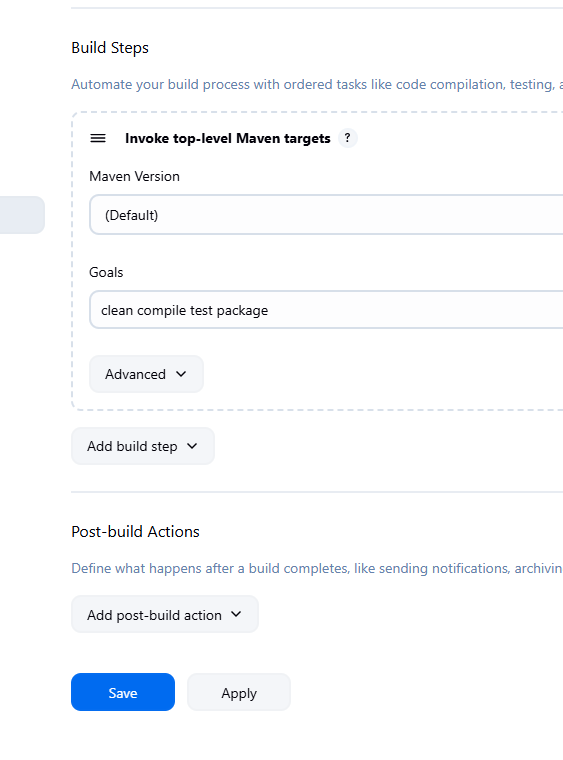


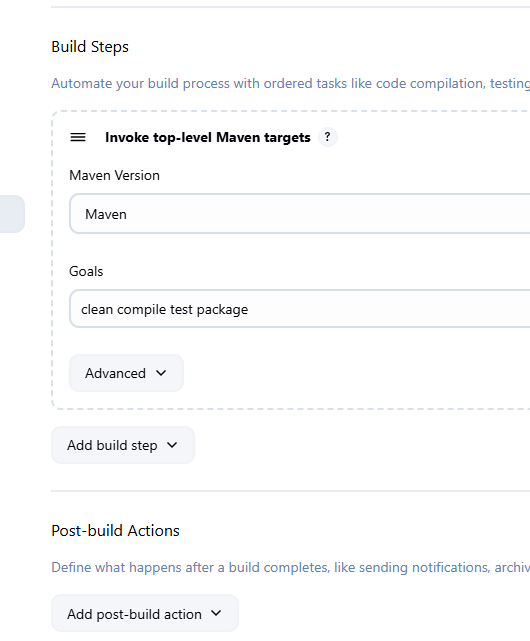
Click Configure



<http://3.99.132.101:8080/manage/configureTools/>





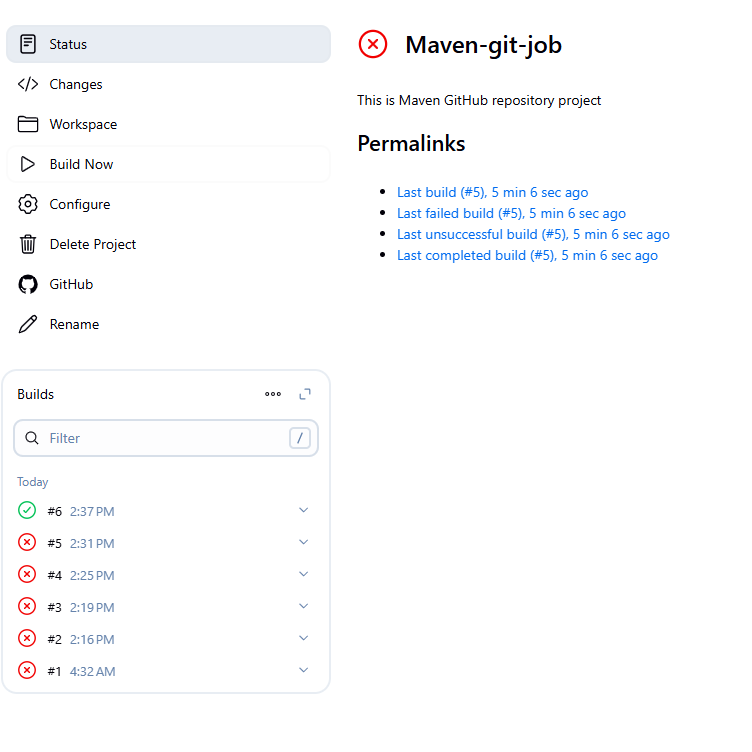


I am changing the git repo

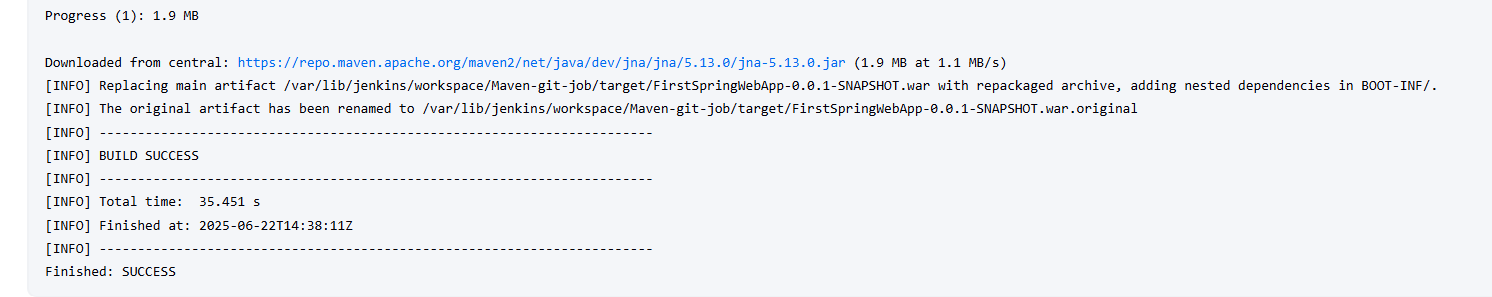
<https://github.com/Haider7214/SpringApp.git>



This time it succeeded. Click Build Now

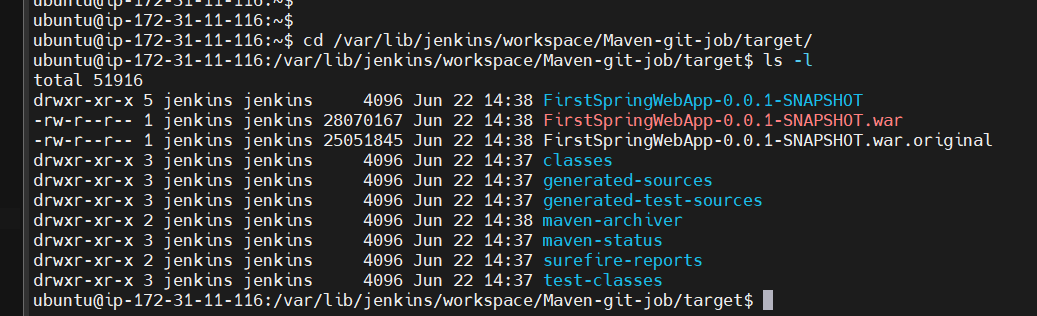


Click on Console Output



See a war file is created

ubuntu@ip-172-31-11-116:~$ cd /var/lib/jenkins/workspace/Maven-git-job/target/



Task 3:

Create Jenkins job Git repo + Maven + Tomcat server

Installing Tomcat into Linux server is one extra step

-> create new VM on EC2 --> Get connected to that Linux VM

-> Install Java

sudo apt update

sudo apt install openjdk-21-jdk

--> Install Tomcat

$ wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.8/bin/apache-tomcat-11.0.8.tar.gz

Dev team -> Push code into --> GitHub repo --> Jenkins

Jenkins should clone with project

Deploy

Clone

GitHub repo

Apache Tomcat

Jenkins

Dev team

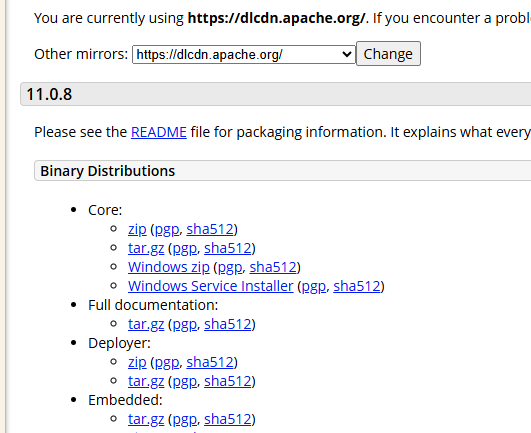
Push code

Build

Maven

For Tomcat go to

<https://tomcat.apache.org/download-11.cgi>



Copy tar link address

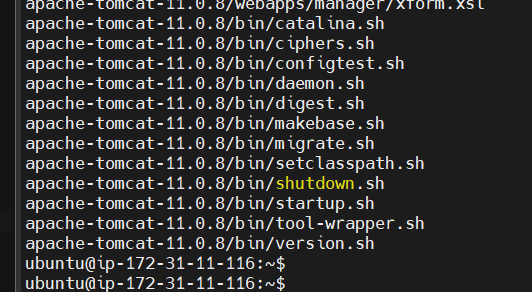
<https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.8/bin/apache-tomcat-11.0.8.tar.gz>

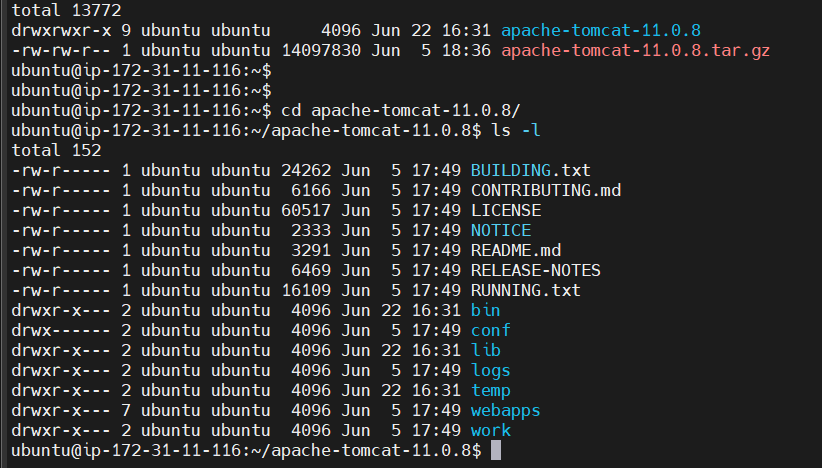
ubuntu@ip-172-31-11-116:/var/lib/jenkins/workspace/Maven-git-job/target$ wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.8/bin/apache-tomcat-11.0.8.tar.gz

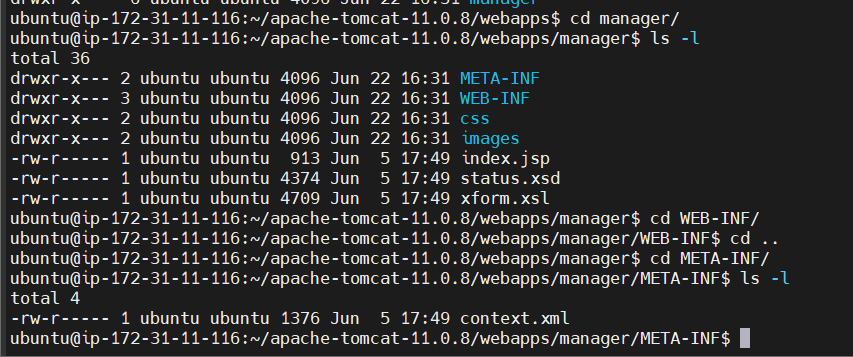


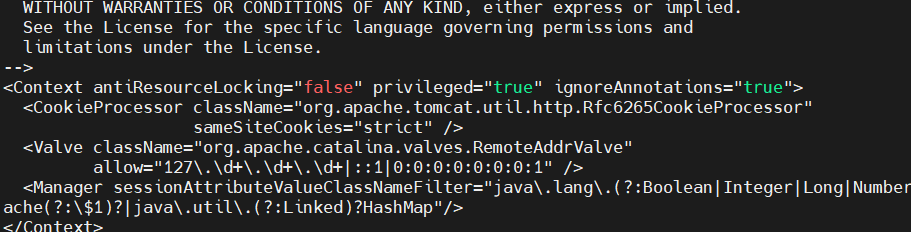
ubuntu@ip-172-31-11-116:~$ tar -xvf apache-tomcat-11.0.8.tar.gz

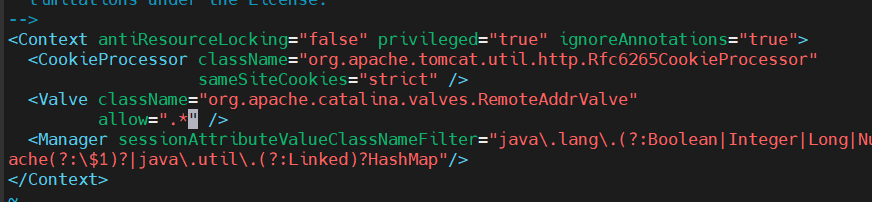
startup.sh for Linux and startup.bat for Windows











Change to “.\*”

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8$ ls -l

total 152

-rw-r----- 1 ubuntu ubuntu 24262 Jun 5 17:49 BUILDING.txt

-rw-r----- 1 ubuntu ubuntu 6166 Jun 5 17:49 CONTRIBUTING.md

-rw-r----- 1 ubuntu ubuntu 60517 Jun 5 17:49 LICENSE

-rw-r----- 1 ubuntu ubuntu 2333 Jun 5 17:49 NOTICE

-rw-r----- 1 ubuntu ubuntu 3291 Jun 5 17:49 README.md

-rw-r----- 1 ubuntu ubuntu 6469 Jun 5 17:49 RELEASE-NOTES

-rw-r----- 1 ubuntu ubuntu 16109 Jun 5 17:49 RUNNING.txt

drwxr-x--- 2 ubuntu ubuntu 4096 Jun 22 16:31 bin

drwx------ 2 ubuntu ubuntu 4096 Jun 5 17:49 conf

drwxr-x--- 2 ubuntu ubuntu 4096 Jun 22 16:31 lib

drwxr-x--- 2 ubuntu ubuntu 4096 Jun 5 17:49 logs

drwxr-x--- 2 ubuntu ubuntu 4096 Jun 22 16:31 temp

drwxr-x--- 7 ubuntu ubuntu 4096 Jun 5 17:49 webapps

drwxr-x--- 2 ubuntu ubuntu 4096 Jun 5 17:49 work

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8$

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8$

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8$ cd conf

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8/conf$ ls -l

total 212

-rw------- 1 ubuntu ubuntu 6636 Jun 5 17:49 catalina.properties

-rw------- 1 ubuntu ubuntu 1411 Jun 5 17:49 context.xml

-rw------- 1 ubuntu ubuntu 1149 Jun 5 17:49 jaspic-providers.xml

-rw------- 1 ubuntu ubuntu 2313 Jun 5 17:49 jaspic-providers.xsd

-rw------- 1 ubuntu ubuntu 4003 Jun 5 17:49 logging.properties

-rw------- 1 ubuntu ubuntu 6905 Jun 5 17:49 server.xml

-rw------- 1 ubuntu ubuntu 2756 Jun 5 17:49 tomcat-users.xml

-rw------- 1 ubuntu ubuntu 2558 Jun 5 17:49 tomcat-users.xsd

-rw------- 1 ubuntu ubuntu 172995 Jun 5 17:49 web.xml

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8/conf$ vi tomcat-users.xml

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="admin-gui"/>

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

<user username="admin" password="admin" roles="manager-script, admin-gui"/>

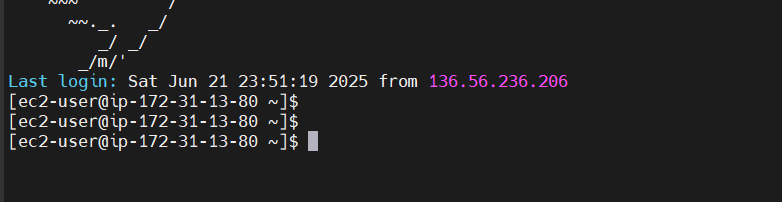


ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8/conf$ cd ..

ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8$ cd bin/

Lets launch a new EC2 instance for Tomcat-server due to some port availability issue

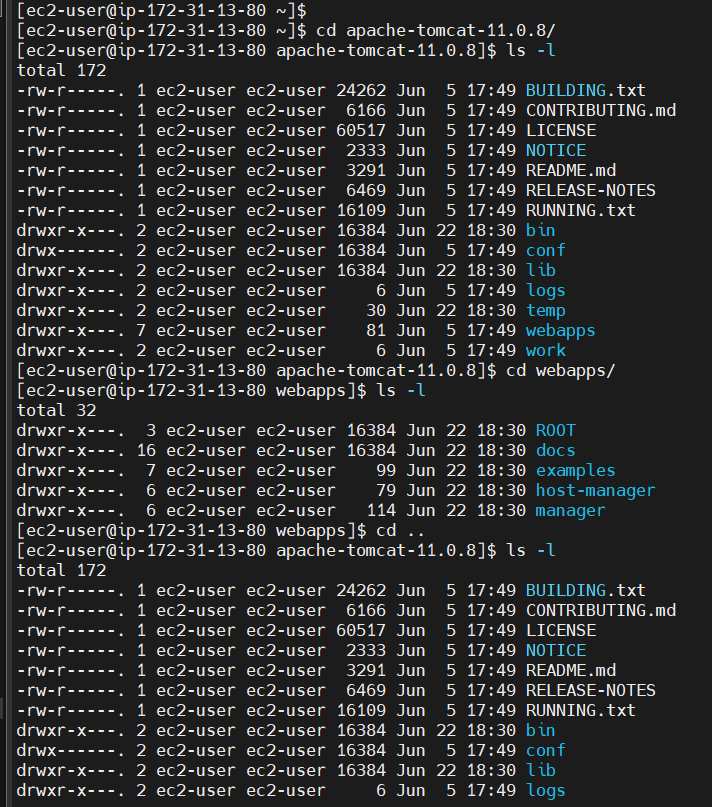
t2.medium is not needed

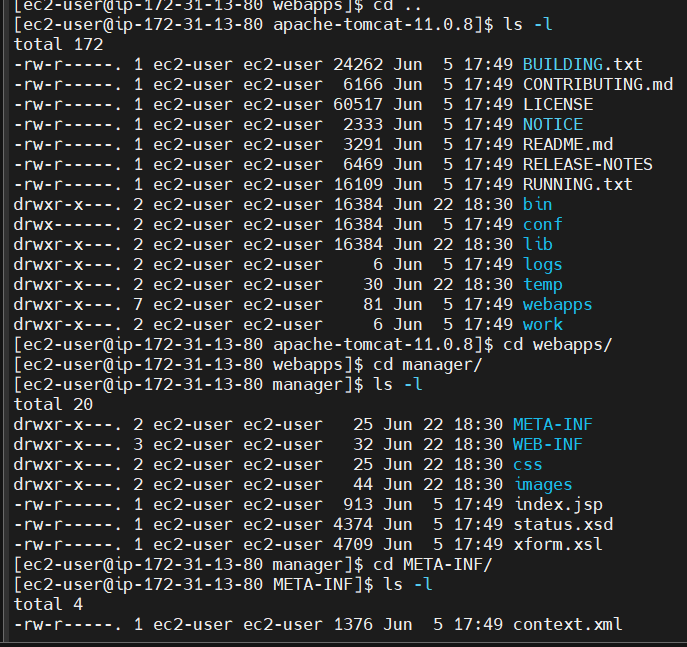


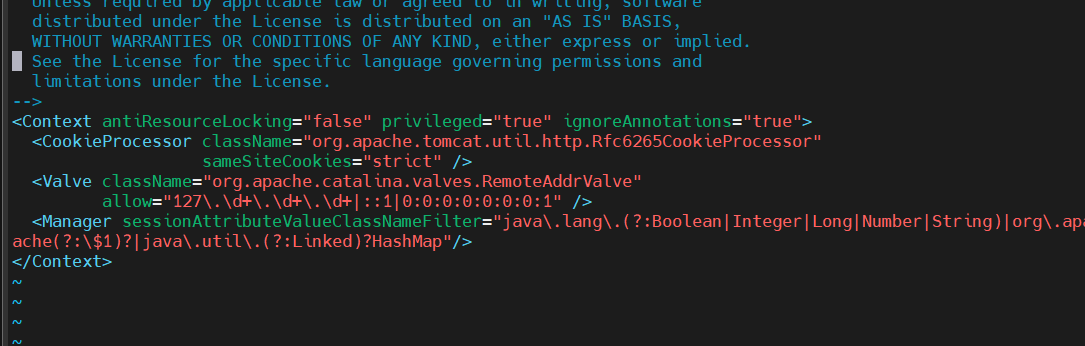
wget <https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.8/bin/apache-tomcat-11.0.8.tar.gz>

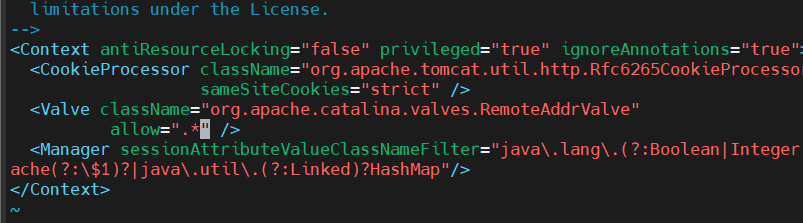
[ec2-user@ip-172-31-13-80 ~]$ wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.8/bin/apache-tomcat-11.0.8.tar.gz

[ec2-user@ip-172-31-13-80 ~]$ tar -xvf apache-tomcat-11.0.8.tar.gz









[ec2-user@ip-172-31-13-80 apache-tomcat-11.0.8]$ cd conf

[ec2-user@ip-172-31-13-80 conf]$ ls -l

total 212

-rw-------. 1 ec2-user ec2-user 6636 Jun 5 17:49 catalina.properties

-rw-------. 1 ec2-user ec2-user 1411 Jun 5 17:49 context.xml

-rw-------. 1 ec2-user ec2-user 1149 Jun 5 17:49 jaspic-providers.xml

-rw-------. 1 ec2-user ec2-user 2313 Jun 5 17:49 jaspic-providers.xsd

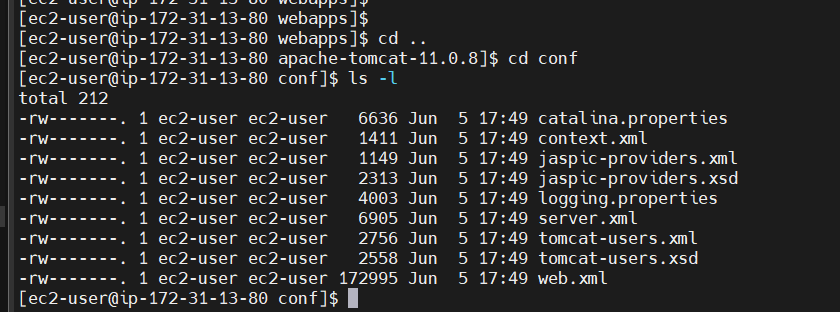
-rw-------. 1 ec2-user ec2-user 4003 Jun 5 17:49 logging.properties

-rw-------. 1 ec2-user ec2-user 6905 Jun 5 17:49 server.xml

-rw-------. 1 ec2-user ec2-user 2756 Jun 5 17:49 tomcat-users.xml

-rw-------. 1 ec2-user ec2-user 2558 Jun 5 17:49 tomcat-users.xsd

-rw-------. 1 ec2-user ec2-user 172995 Jun 5 17:49 web.xml



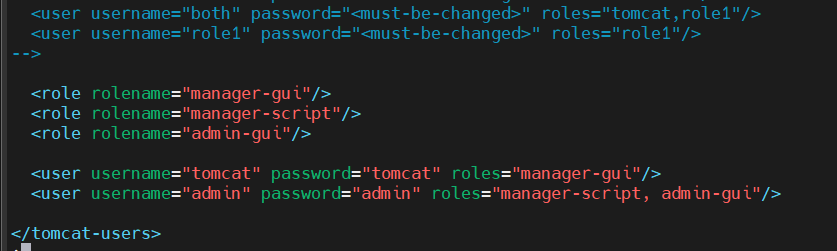
<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="admin-gui"/>

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

<user username="admin" password="admin" roles="manager-script, admin-gui"/>



[ec2-user@ip-172-31-13-80 bin]$ sh startup.sh

[ec2-user@ip-172-31-13-80 bin]$ sh startup.sh

Neither the JAVA\_HOME nor the JRE\_HOME environment variable is defined

At least one of these environment variable is needed to run this program

We have to install Java

[ec2-user@ip-172-31-13-80 ~]$ sudo apt install openjdk-21-jdk

sudo rpm --import https://yum.corretto.aws/corretto.key

sudo curl -Lo /etc/yum.repos.d/corretto.repo https://yum.corretto.aws/corretto.repo

sudo yum install -y java-21-amazon-corretto-devel

java -version

[ec2-user@ip-172-31-13-80 ~]$ cd apache-tomcat-11.0.8/

[ec2-user@ip-172-31-13-80 apache-tomcat-11.0.8]$ cd bin/

[ec2-user@ip-172-31-13-80 bin]$ sh startup.sh

Using CATALINA\_BASE: /home/ec2-user/apache-tomcat-11.0.8

Using CATALINA\_HOME: /home/ec2-user/apache-tomcat-11.0.8

Using CATALINA\_TMPDIR: /home/ec2-user/apache-tomcat-11.0.8/temp

Using JRE\_HOME: /usr

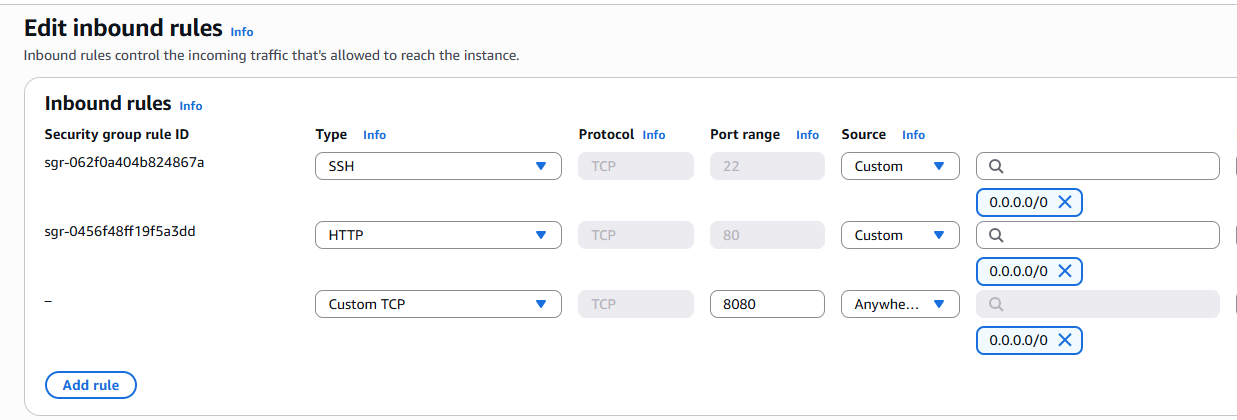
Using CLASSPATH: /home/ec2-user/apache-tomcat-11.0.8/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-11.0.8/bin/tomcat-juli.jar

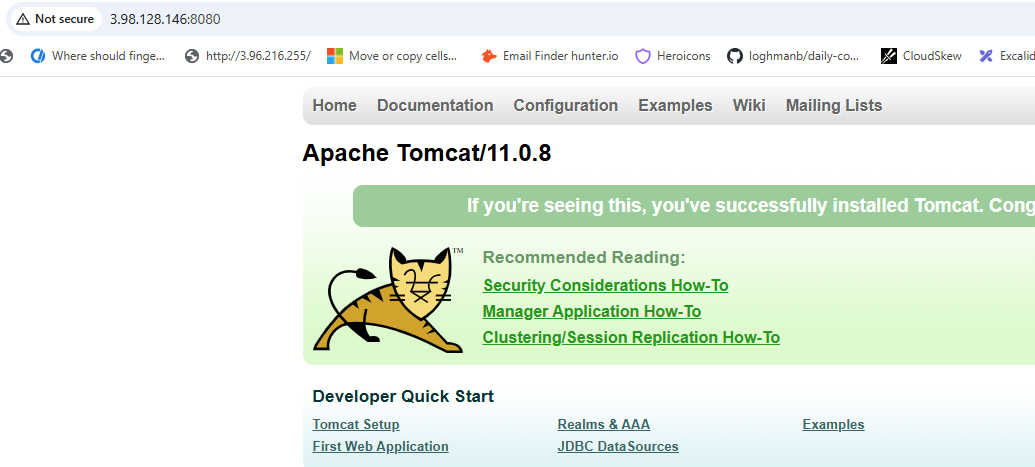
Using CATALINA\_OPTS:

Tomcat started.



<http://3.98.128.146:8080/>

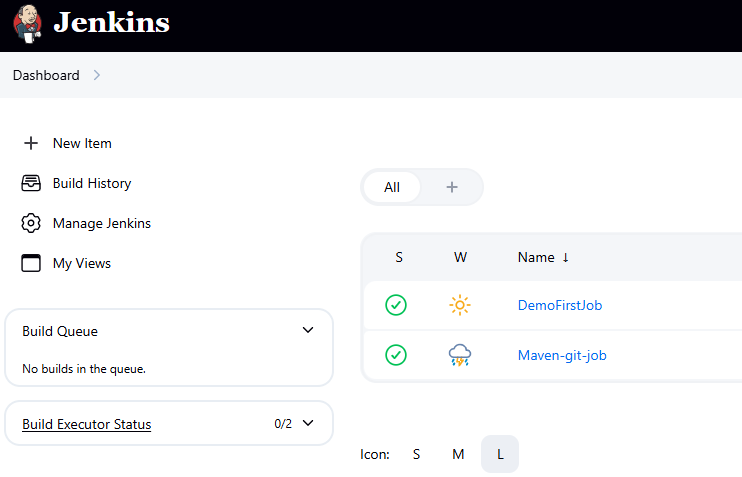




Go back to Jenkins

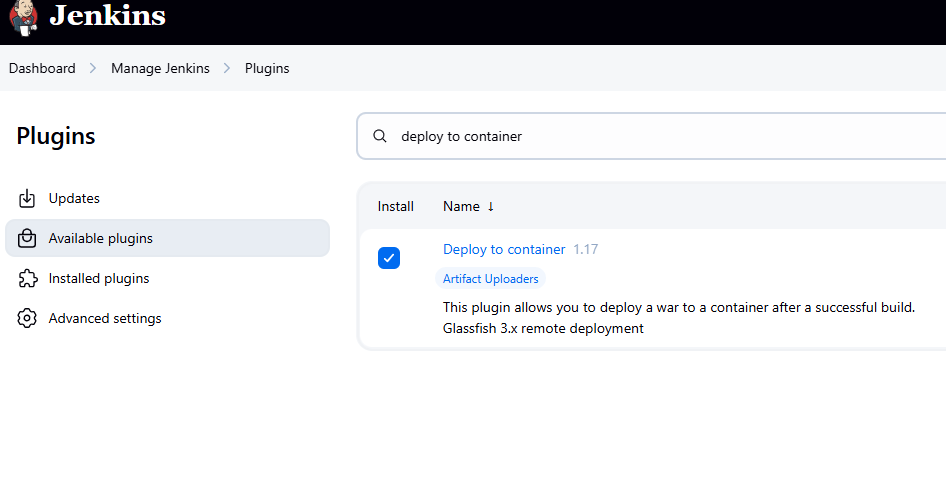
<http://3.99.132.101:8080/login?from=%2Fjob%2FMaven-git-job%2F6%2F>

Manage Jenkins

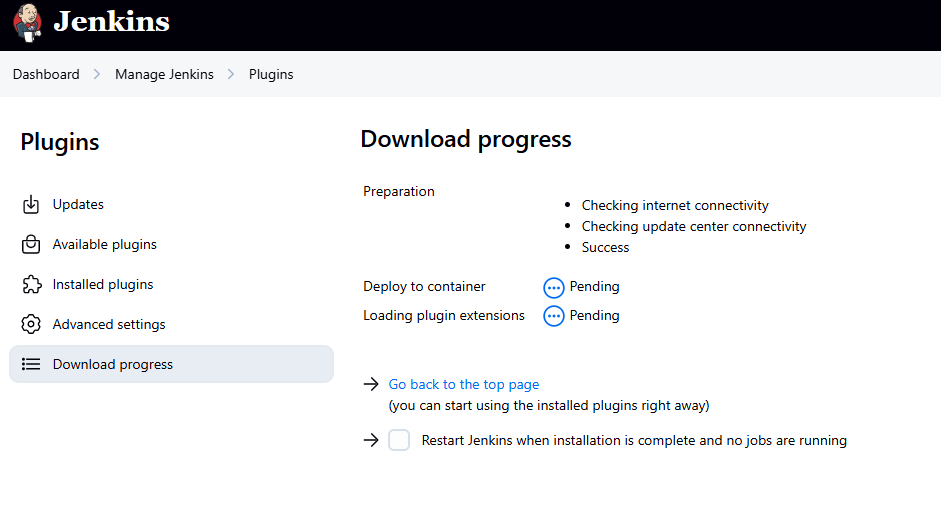


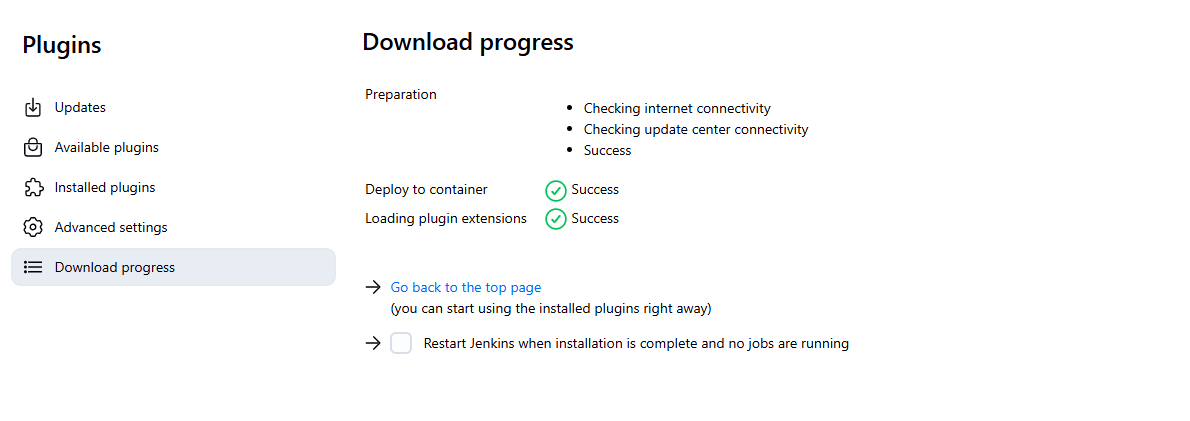
Click Plugins

Available Plugins --> Search “Deploy to container”

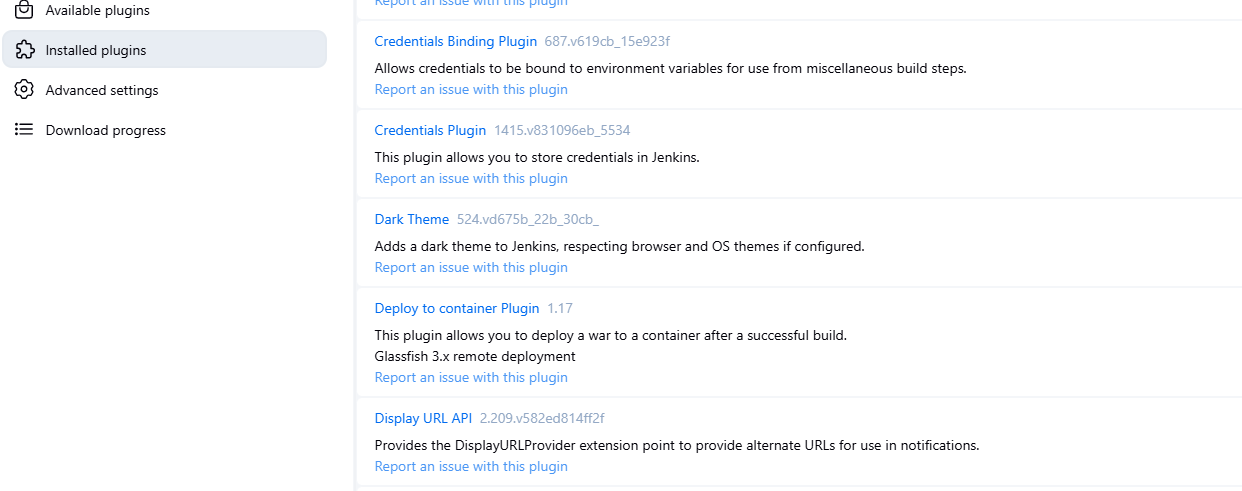


Click Install

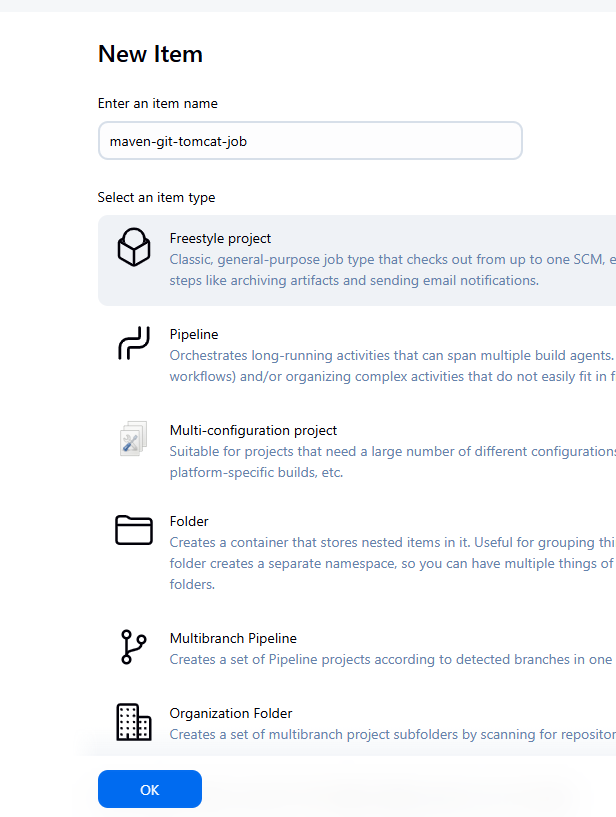


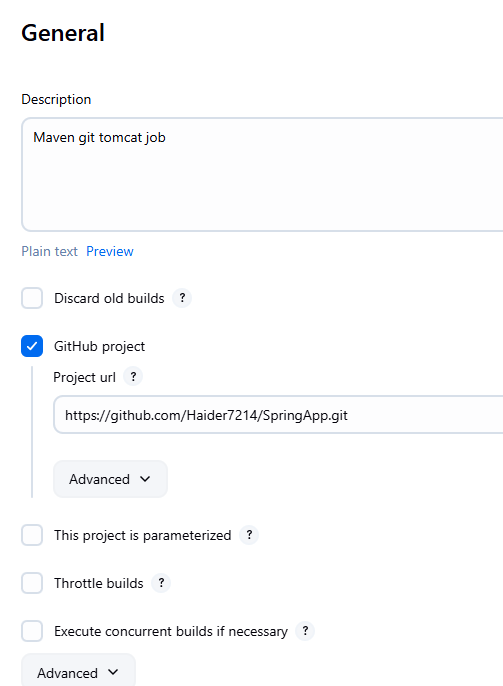


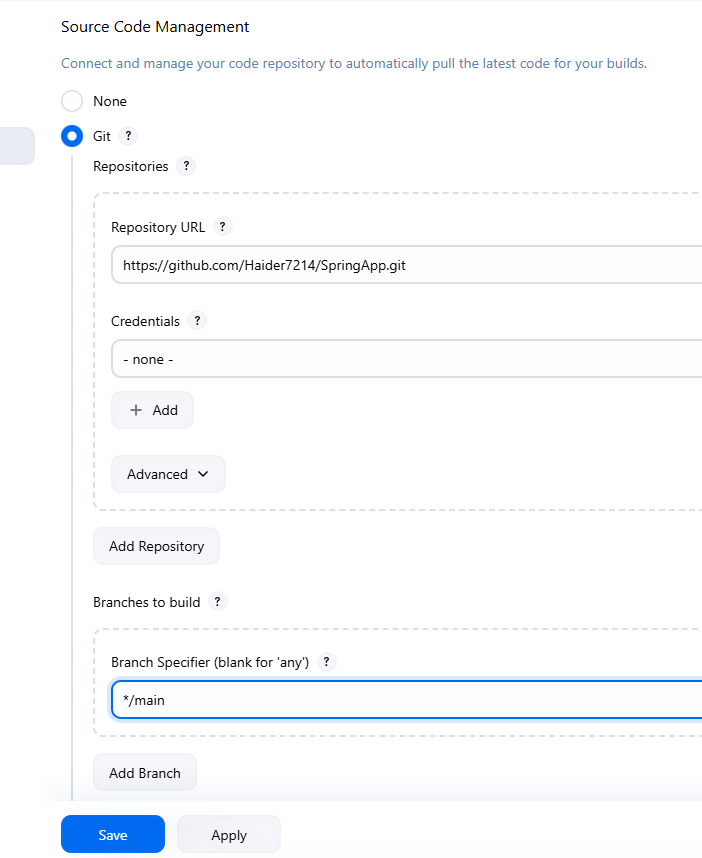
Installed Plugin

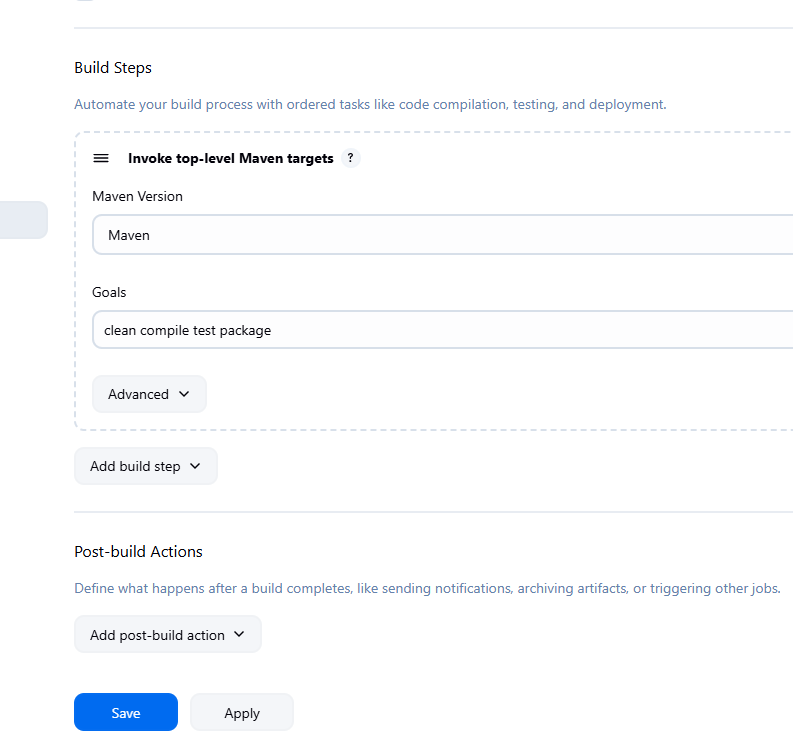


Dashboard --> New Item

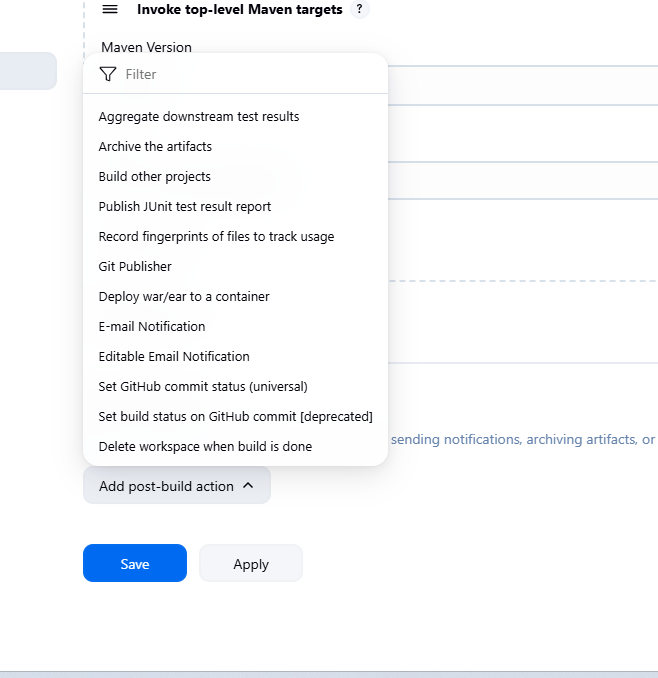




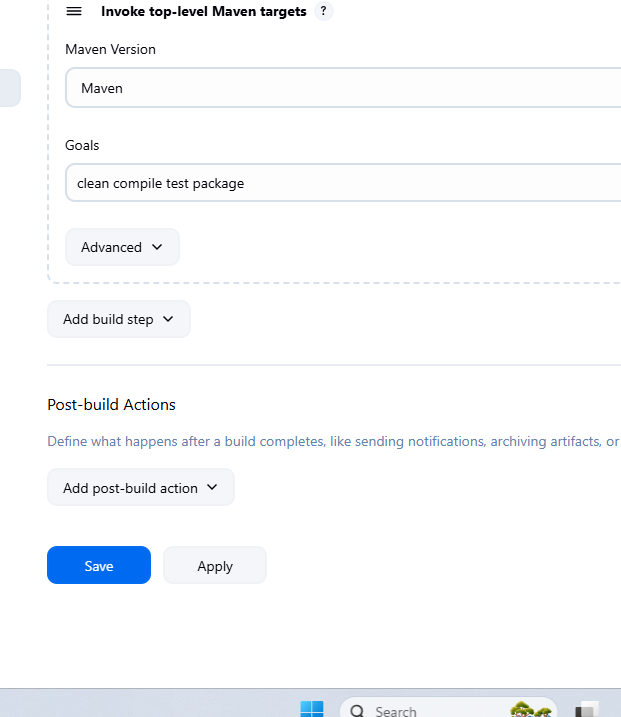


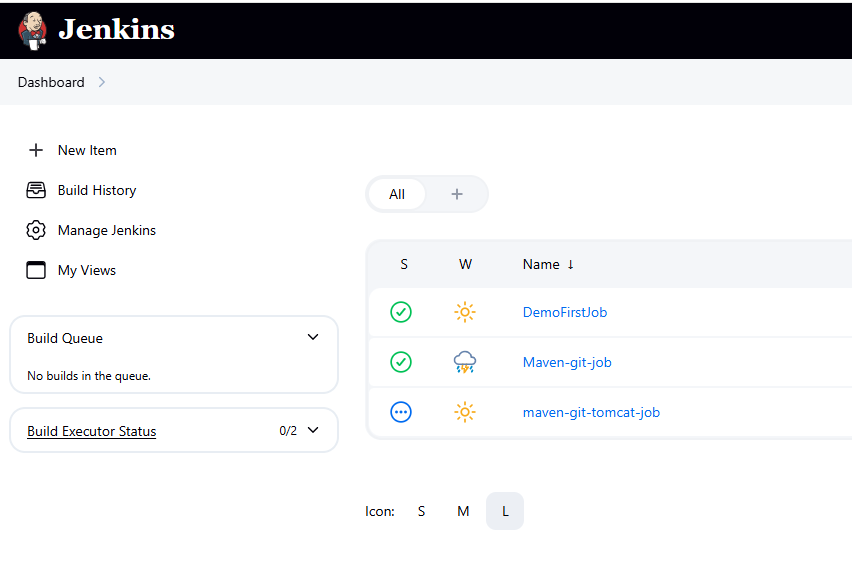


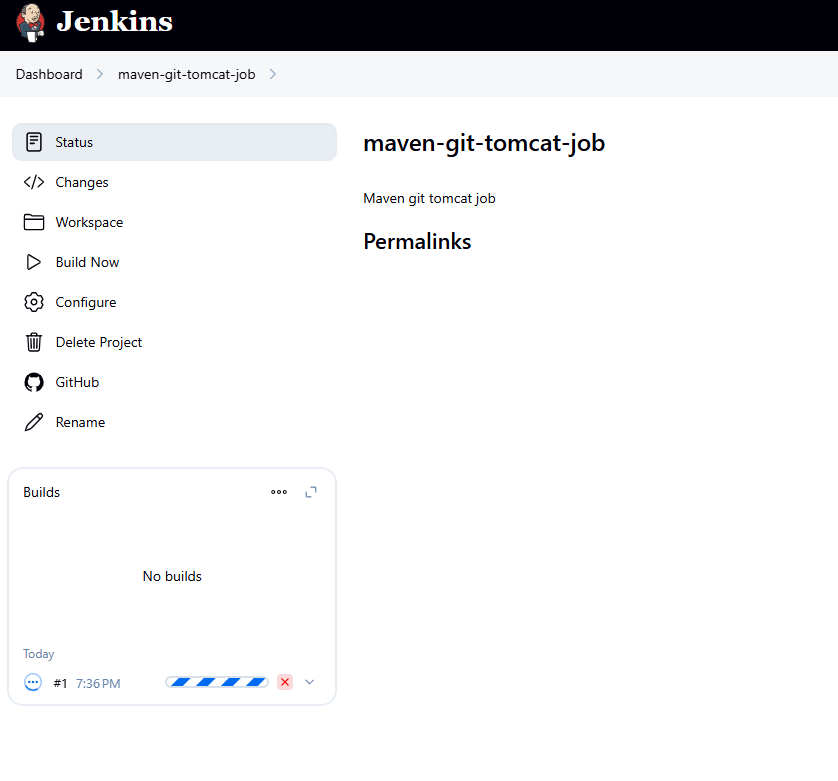
Deploy war to a container

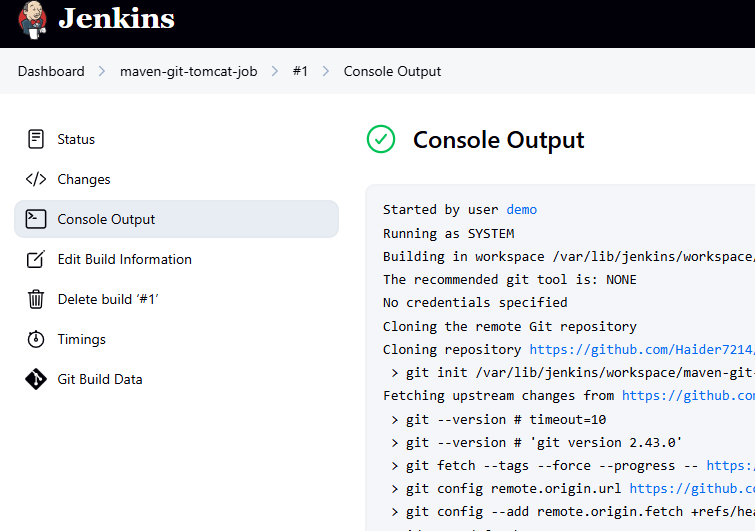


No we are not doing Deploy war to container







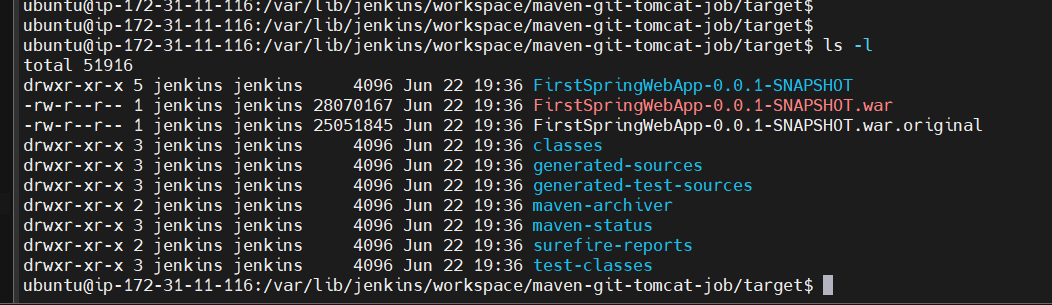




[INFO] The original artifact has been renamed to /var/lib/jenkins/workspace/maven-git-tomcat-job/target/FirstSpringWebApp-0.0.1-SNAPSHOT.war.original

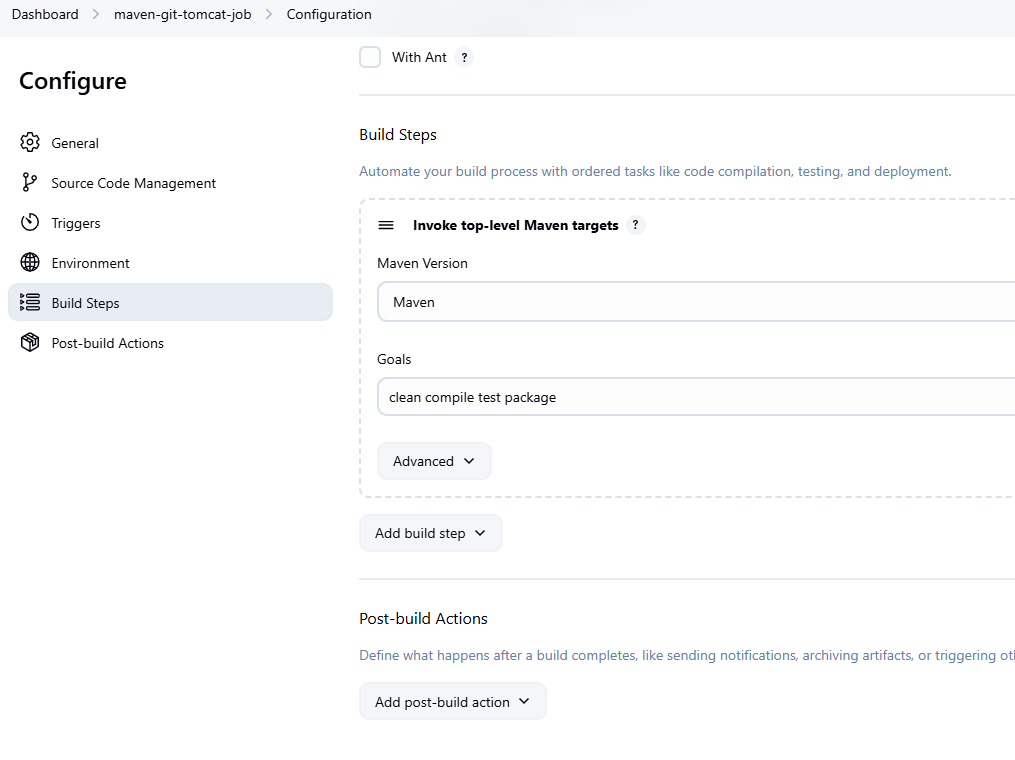
ubuntu@ip-172-31-11-116:~/apache-tomcat-11.0.8/bin$ cd /var/lib/jenkins/workspace/maven-git-tomcat-job/target/

ubuntu@ip-172-31-11-116:/var/lib/jenkins/workspace/maven-git-tomcat-job/target$



Go back to the Job on Jenkins

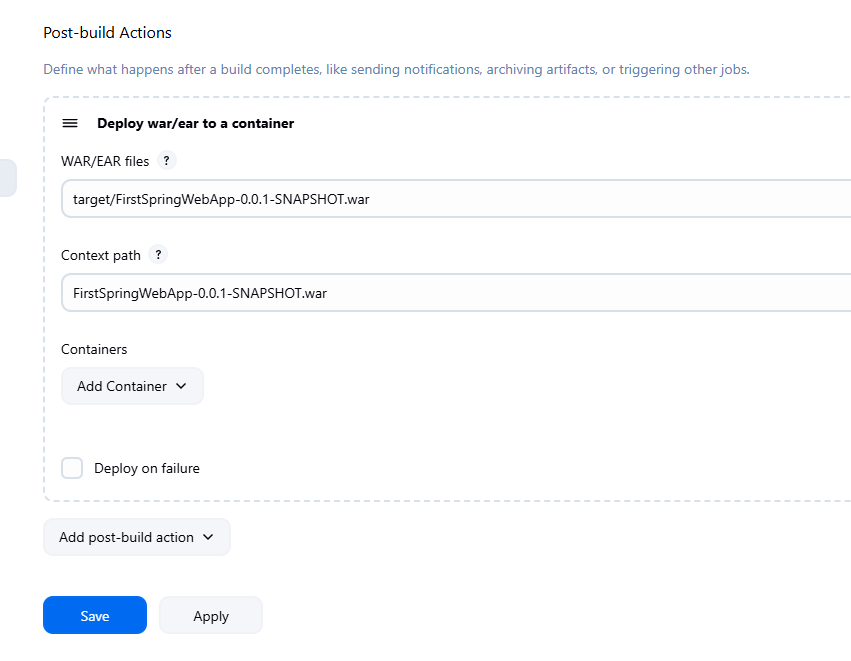
Configure



Copy war filename name from Console log

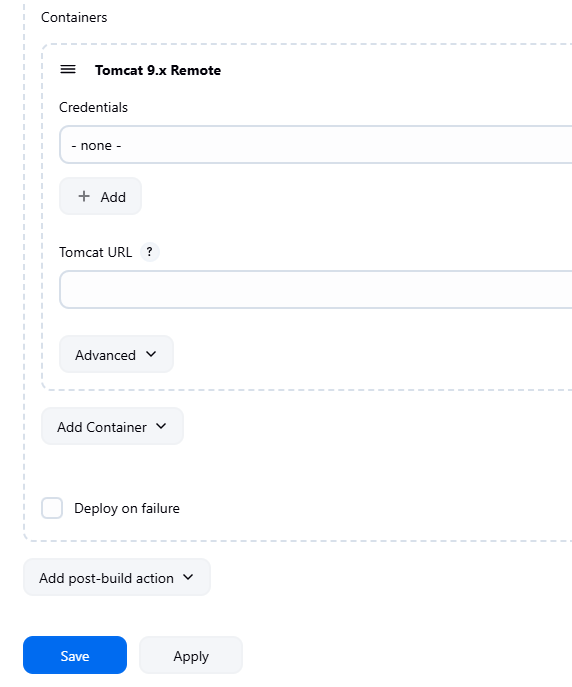
[INFO] Replacing main artifact /var/lib/jenkins/workspace/maven-git-tomcat-job/target/FirstSpringWebApp-0.0.1-SNAPSHOT.war

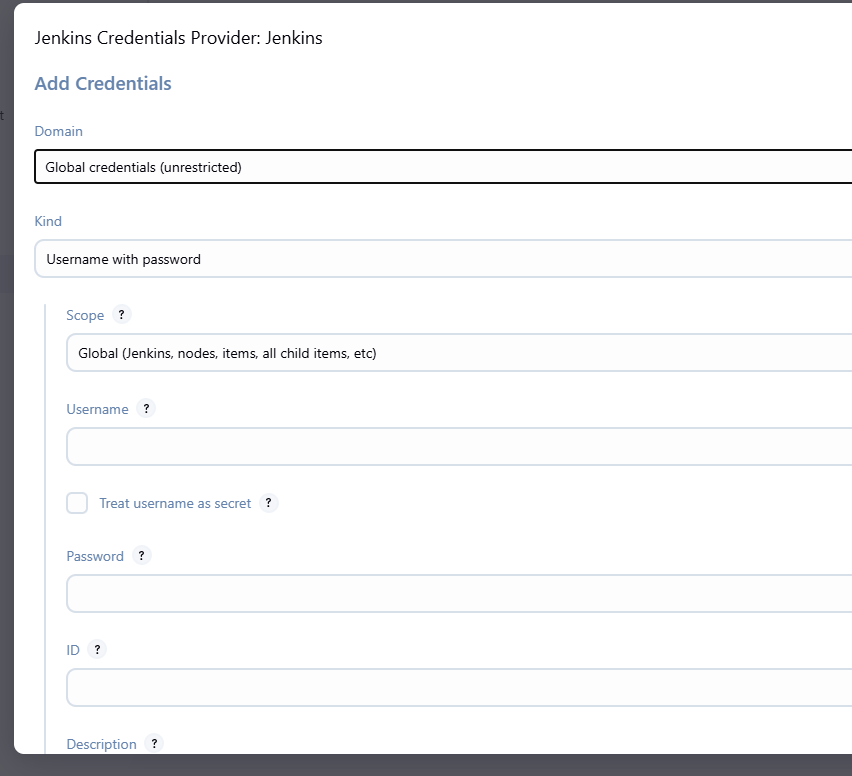
FirstSpringWebApp-0.0.1-SNAPSHOT.war

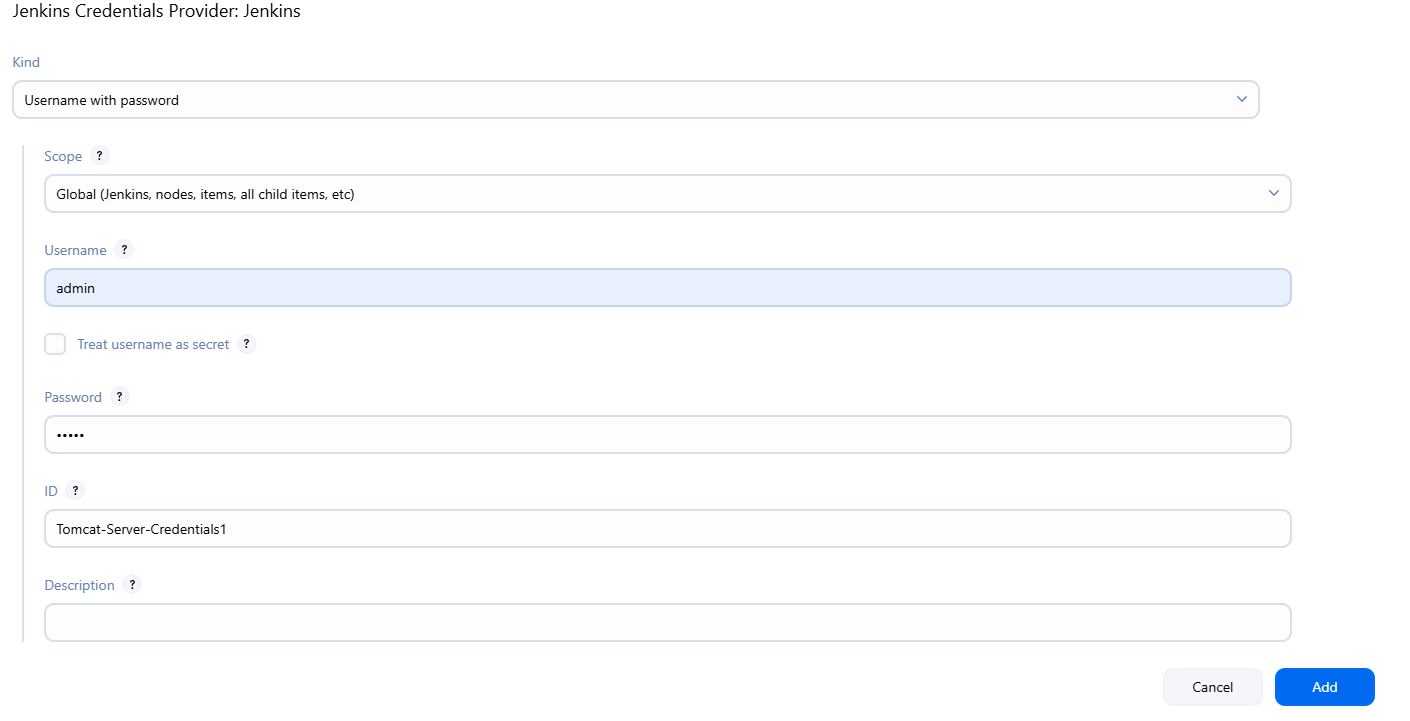


Add Container

We have installed Tomcat 11 in the other VM but only tomcat 9.x is available here



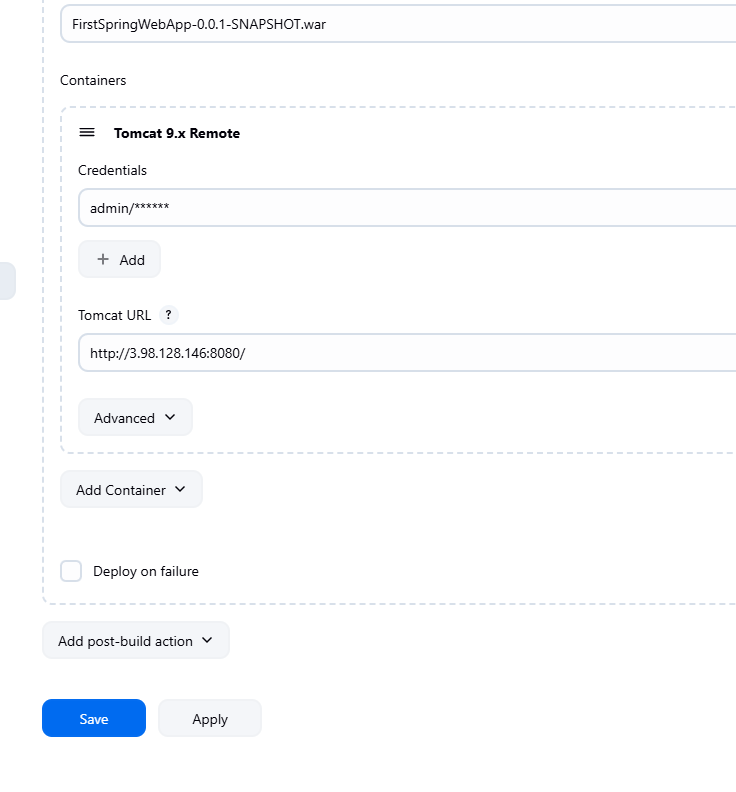




Click Add

Copy Tomcat URL from the other VM

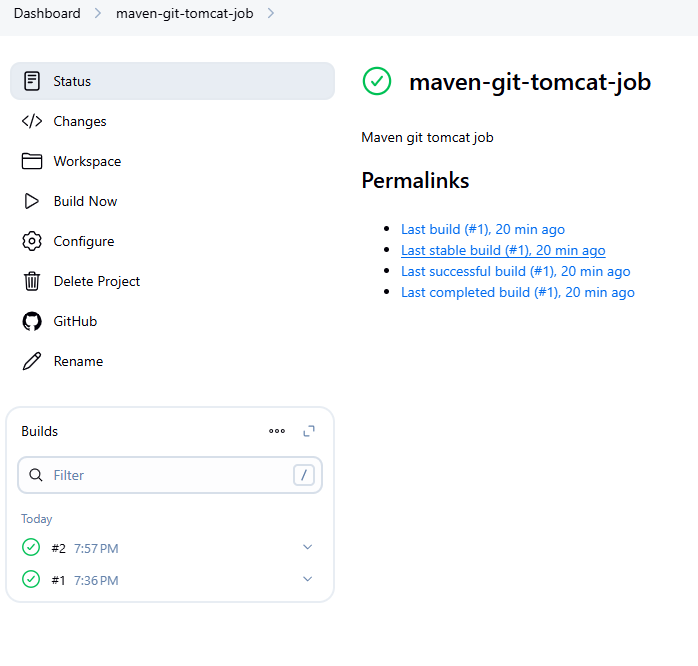
<http://3.98.128.146:8080/>



Click Apply then Save

Click Build Now

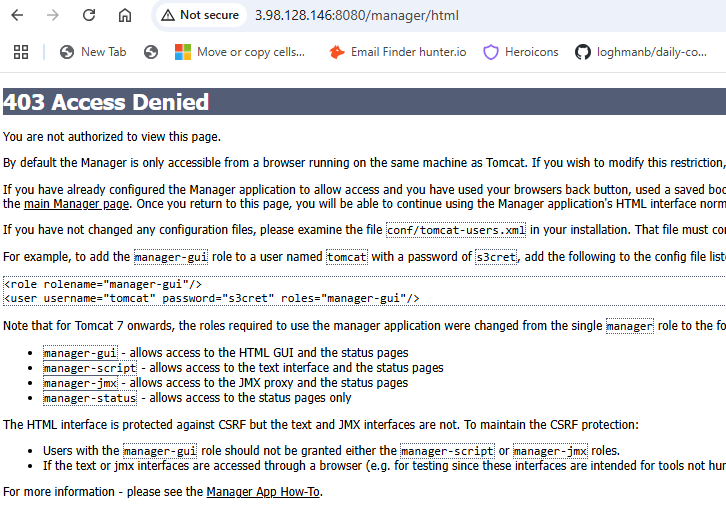
It built successfully

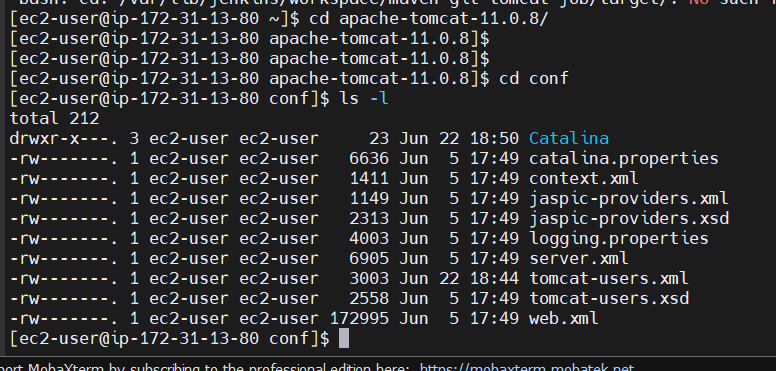


Go back to Tomcat IP

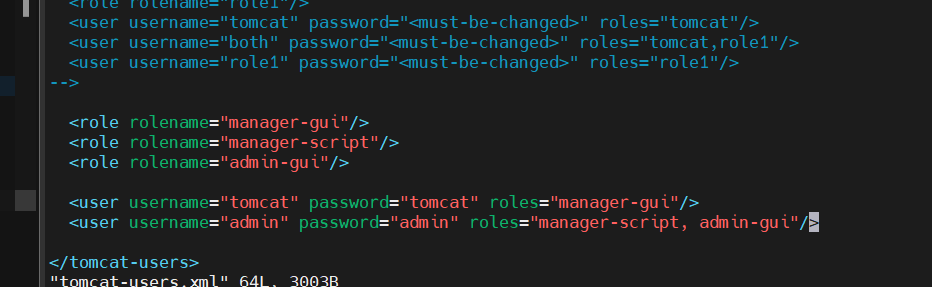
Click on Manager App

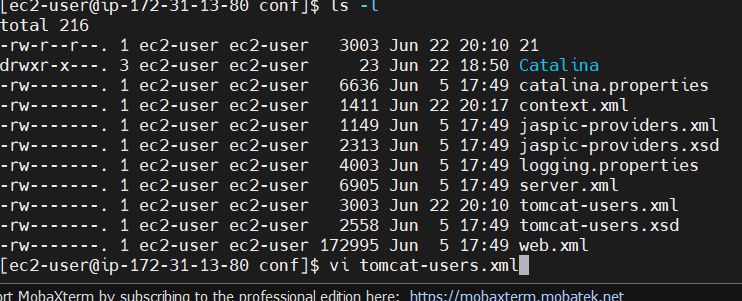
For some reasons it is not opening

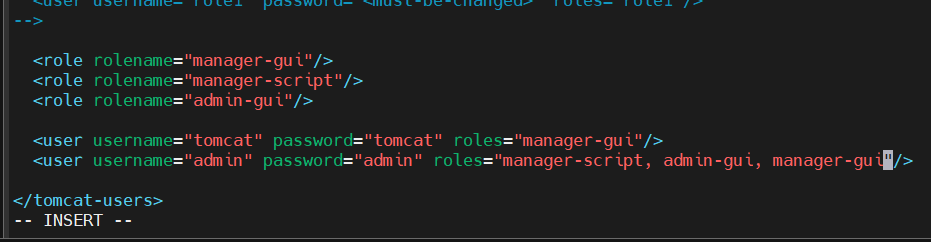


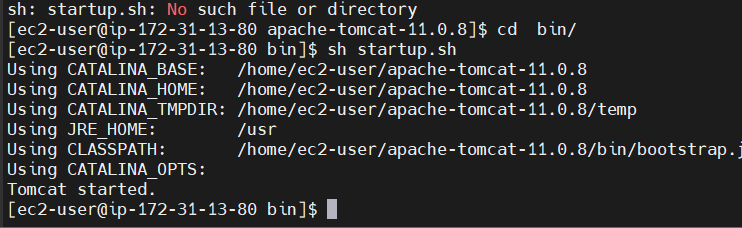


Checking



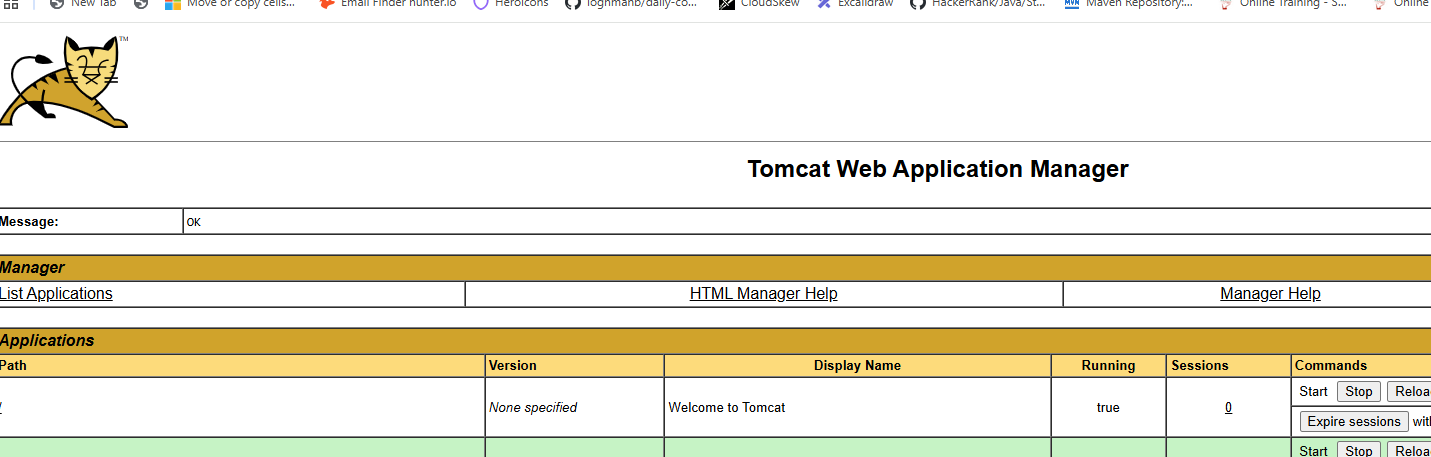




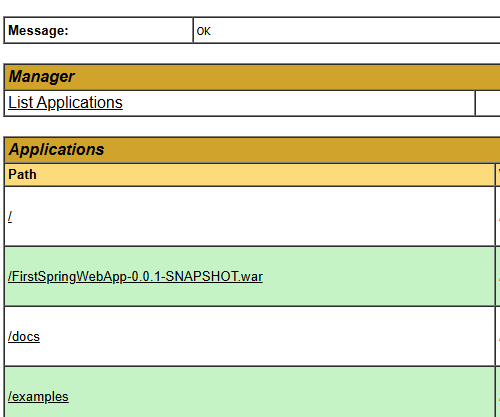


Click on Manager App

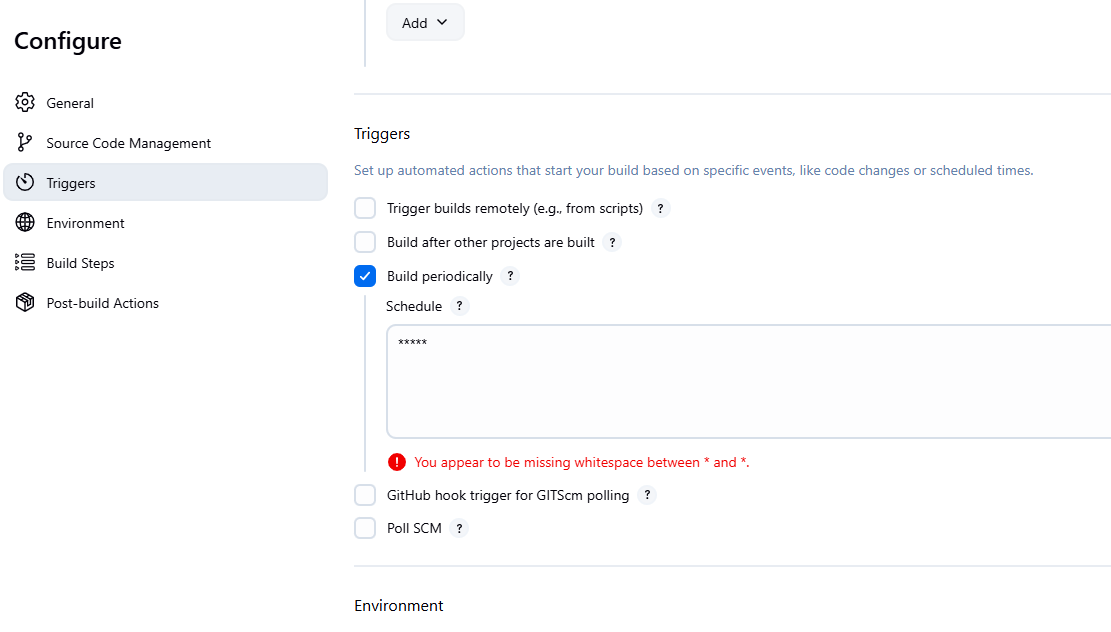




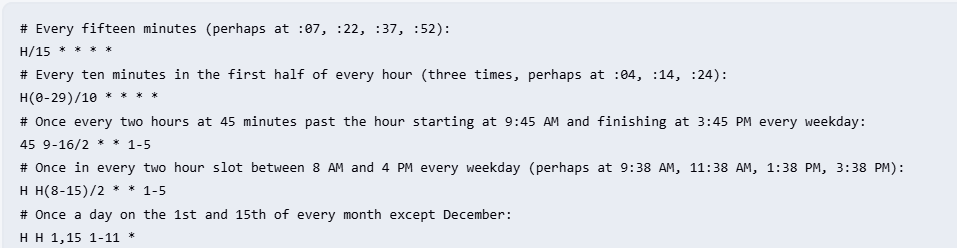
First Web Application is deployed



See our application is deployed

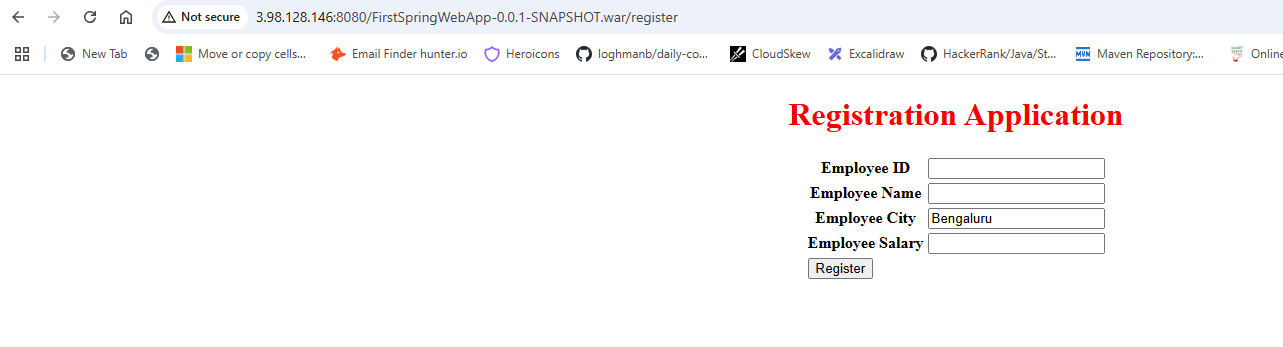


Triggers 5 stars means every 1 minute this job will be triggered



http://3.98.128.146:8080/FirstSpringWebApp-0.0.1-SNAPSHOT.war/register

See the deployed App



In Triggers, Poll SCM will start only if there is a new re-build

