**JENKINS INSTALLATION ON CENOS**

**Prerequists :**

1. Java installation
2. Jenkins installation
3. Maven installation
4. Git installation

*Java installation*

Go to serach java installation oncentos tecadmin or

Execute thease commands on centos

Follow thease steps

**Step 1 – Download Latest Java Archive**

cd /opt/

wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u161-b12/2f38c3b165be4555a1fa6e98c45e0808/jdk-8u161-linux-x64.tar.gz"

tar xzf jdk-8u161-linux-x64.tar.gz

**Step 2 – Install Java 8 with Alternatives**

cd /opt/jdk1.8.0\_161/

alternatives --install /usr/bin/java java /opt/jdk1.8.0\_161/bin/java 2

alternatives --config java

select option 2

alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_161/bin/jar 2

alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_161/bin/javac 2

alternatives --set jar /opt/jdk1.8.0\_161/bin/jar

alternatives --set javac /opt/jdk1.8.0\_161/bin/javac

**Step 3 – Check Installed Java Version**

java -version

**Step 4 – Setup Java Environment Variables**

export JAVA\_HOME=/opt/jdk1.8.0\_161

export JRE\_HOME=/opt/jdk1.8.0\_161/jre

export PATH=$PATH:/opt/jdk1.8.0\_161/bin:/opt/jdk1.8.0\_161/jre/bin

**JENKINS INSTALLATION**

Go to genkins.io

Copy below commands and phaste in your mechine



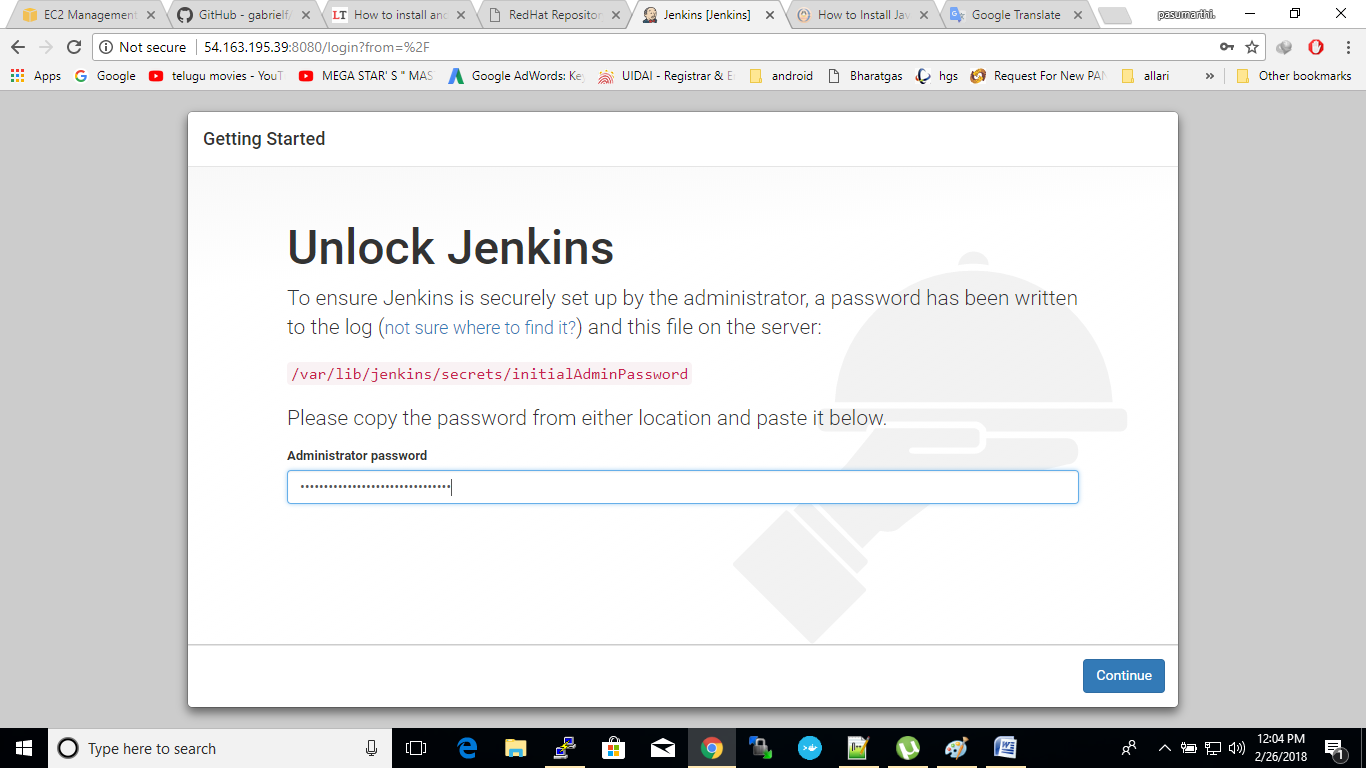


Next we need to start service use this command **service jenkins start**

Go to browser and **IPADDRESS:8080 example** [**http://54.163.195.39:8080**](http://54.163.195.39:8080)

How to change default port 8080

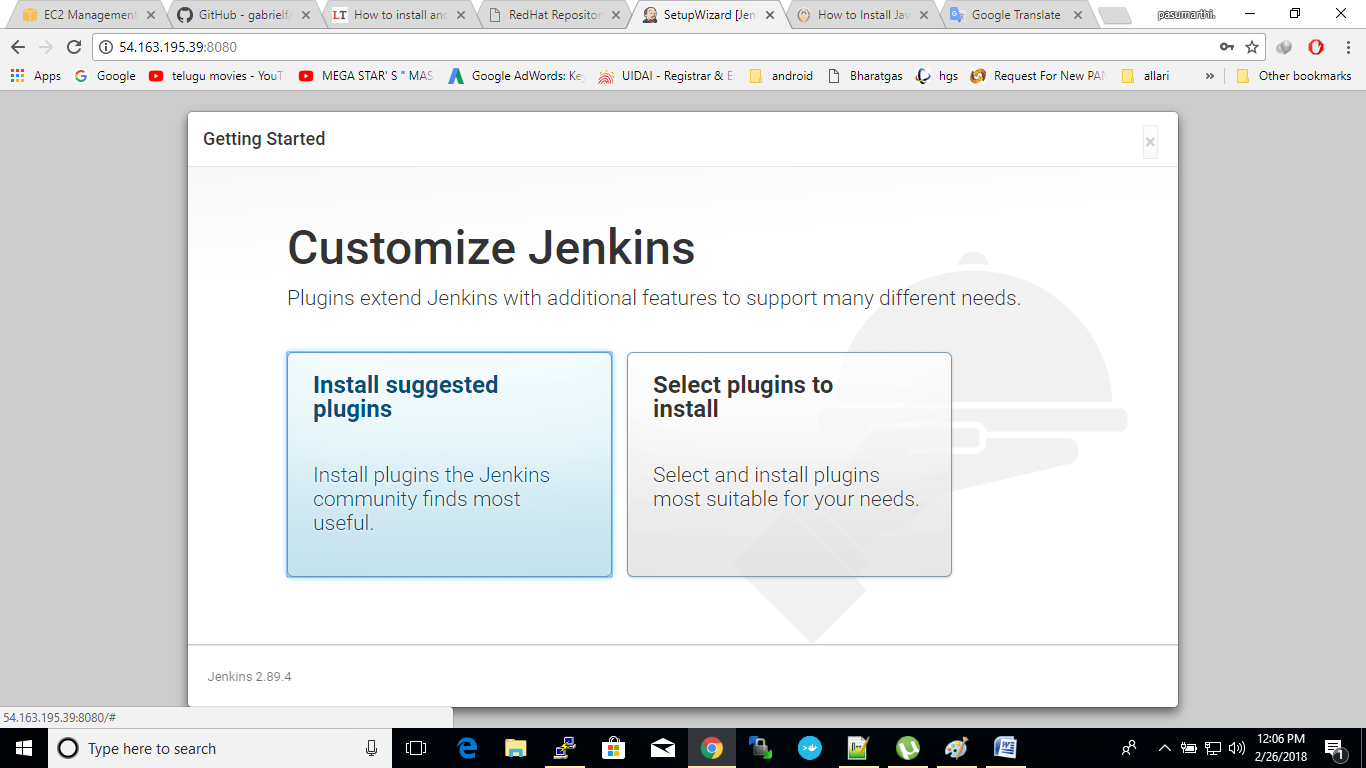
vi /etc/ sysconfig/Jenkins and search JENKINS\_PORT="8080" then change (centos)



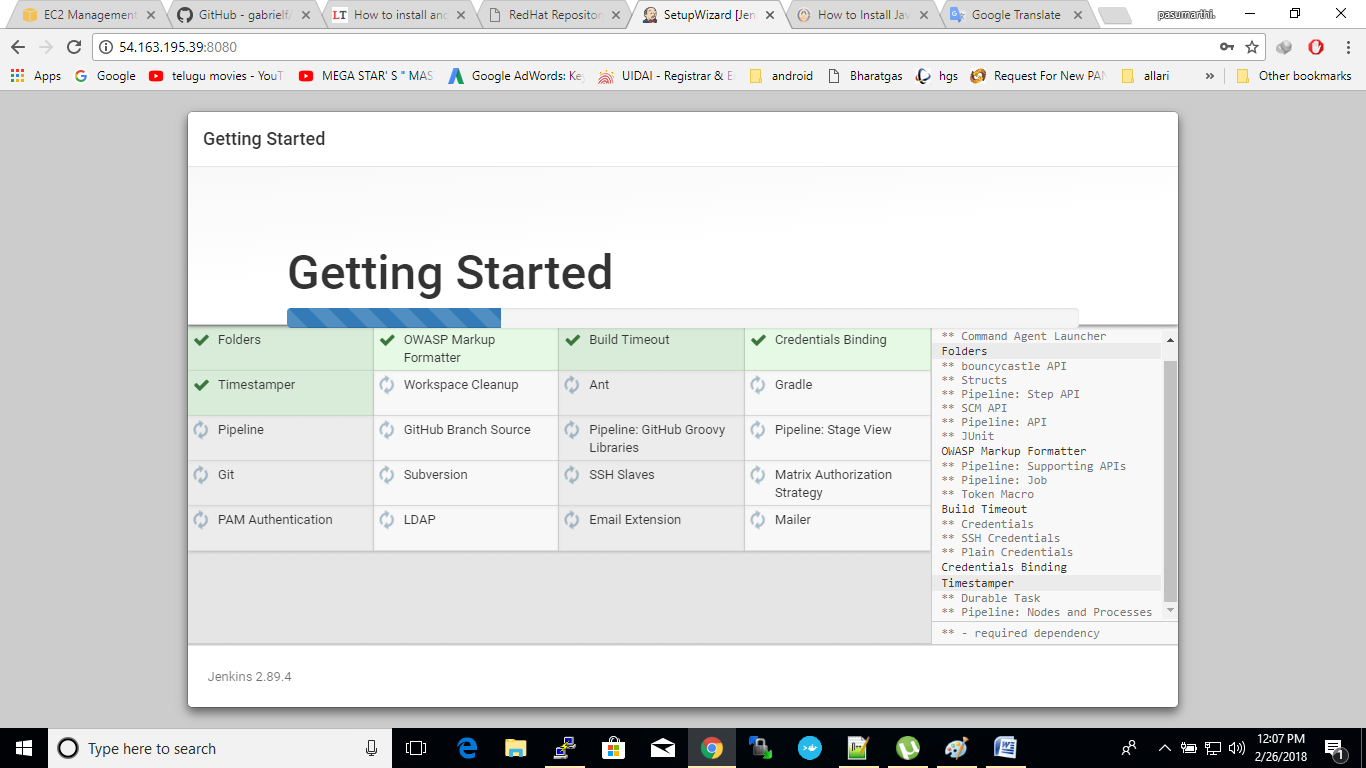
Use this **cat /var/lib/jenkins/secrets/initialAdminPassword**

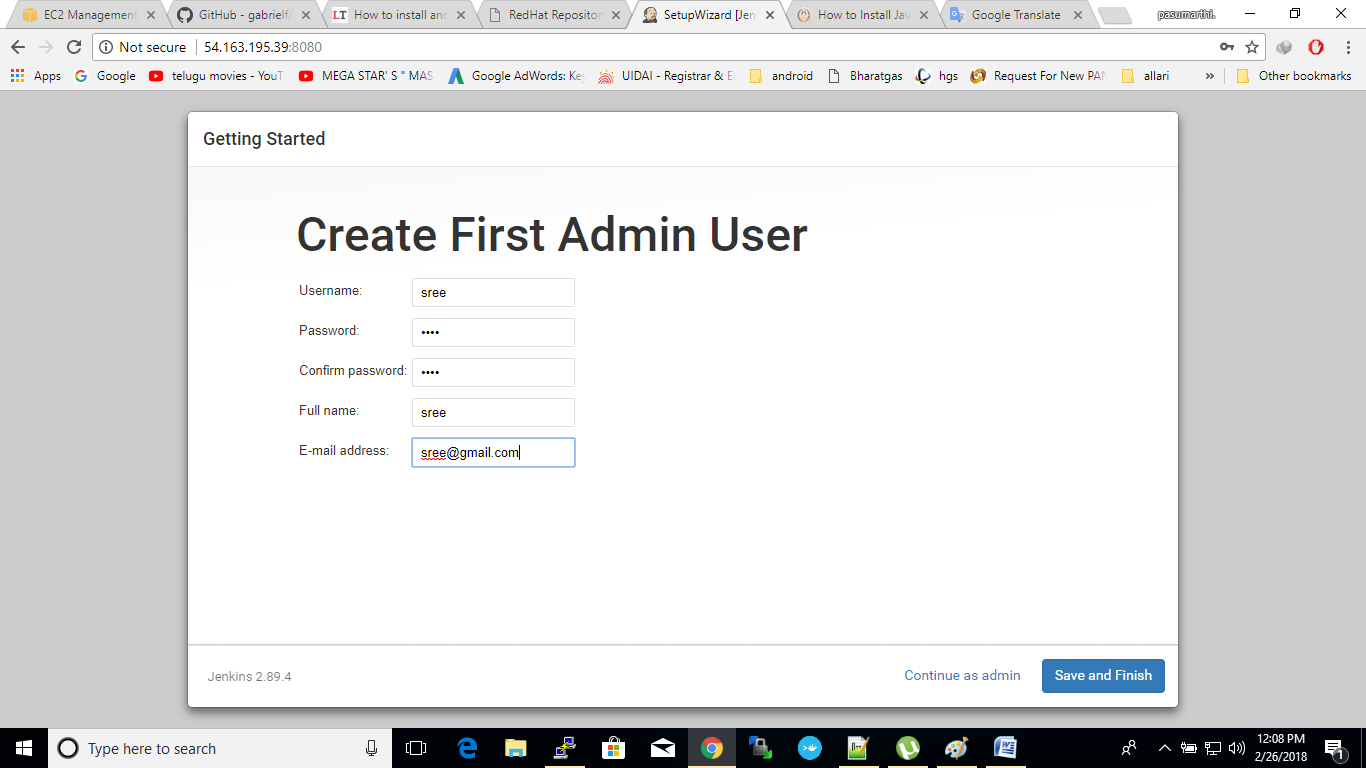
After this command get one token copy that tocken id and phaste in **Administrator password and click on continue**

Example that id 5c0d38b1a41045a090064b2c7f81afbf

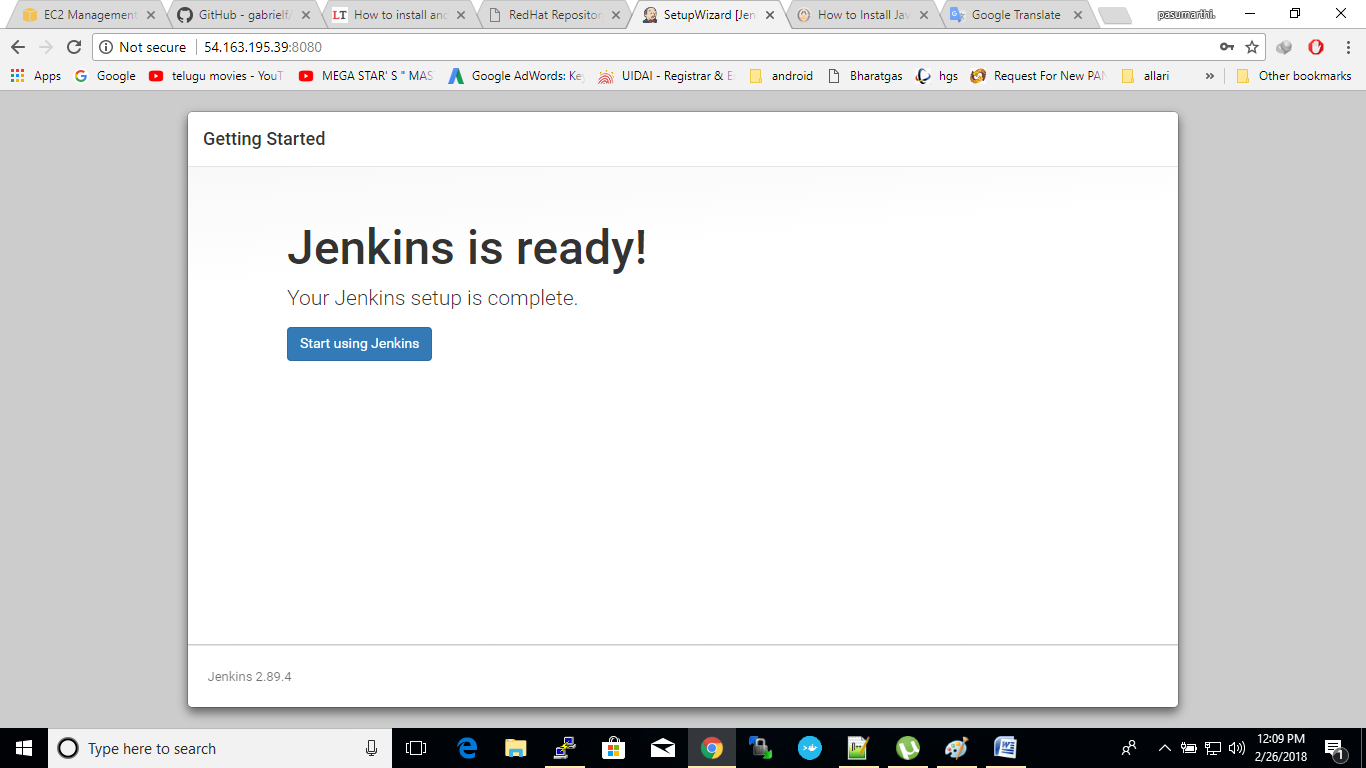


**Select install suggested plugins**

****

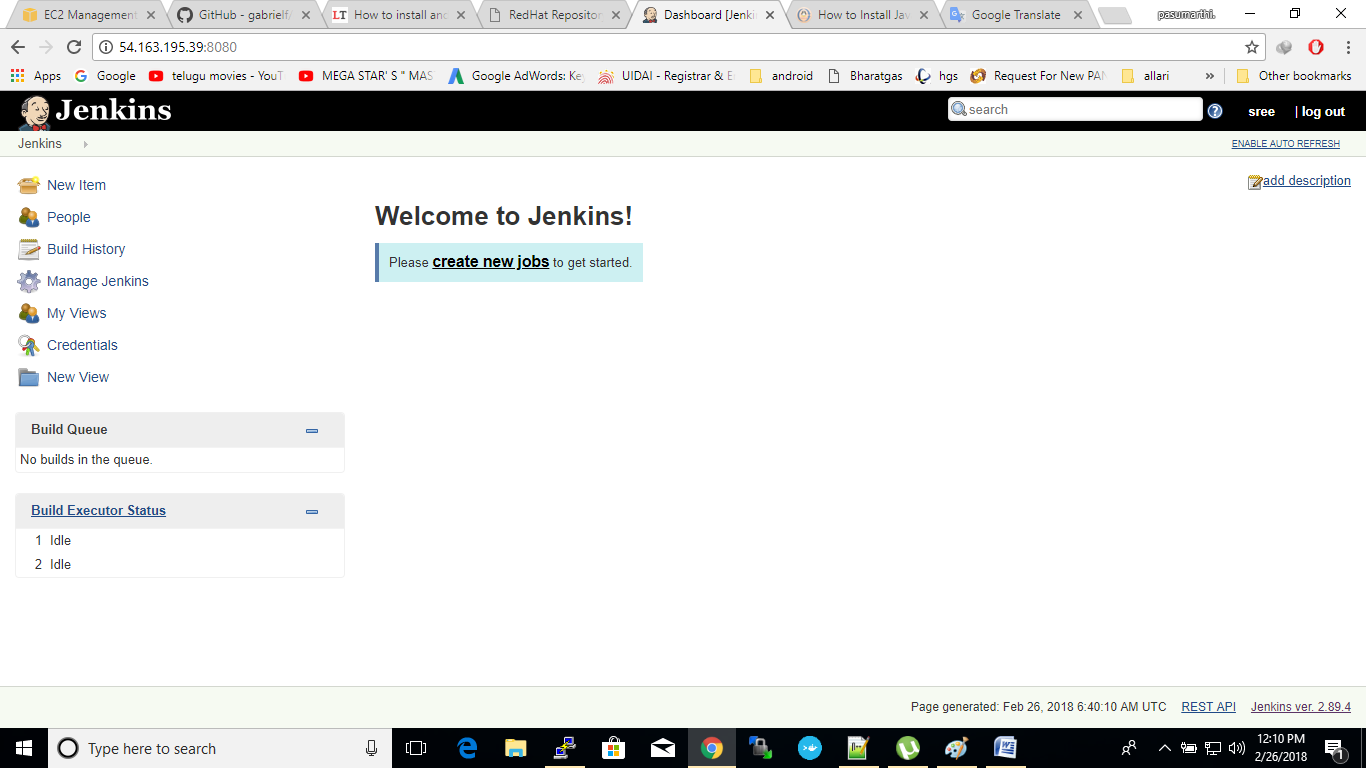
****

**Save and finish**

****

**Select start using Jenkins**

**After that show bellow window**

****

**Maven installation**

Go to search maven installation on centos tecadmin or

**Install Apache Maven**

cd /usr/local

wget http://www-eu.apache.org/dist/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz

sudo tar xzf apache-maven-3.5.2-bin.tar.gz

sudo ln -s apache-maven-3.5.2 maven

Setup Environment Variables

sudo vi /etc/profile.d/maven.sh

export M2\_HOME=/usr/local/maven

export PATH=${M2\_HOME}/bin:${PATH}

source /etc/profile.d/maven.sh

**Check Version**

mvn -version

**Finally, remove the downloaded archive file from the system to free disk space.**

rm -f /usr/local/apache-maven-3.5.2-bin.tar.gz

maven default path **/usr/local/maven**

**Git installation**

**Yum install git**

**Create new job for maven project :**

Manage Jenkins 🡪 Manage plugins select Available and search **“Maven Integration”** then install without restart

**Prerequists :**

1. Set the java path in global tool configuration
2. Set the maven path in global tool configuration

Go to managejenkins🡪 [Global Tool Configuration](http://54.163.195.39:8080/configureTools)

Select Add JDK and JDK name JAVA\_HOME and uncheck the Install automatically

echo $JAVA\_HOME

/opt/jdk1.8.0\_161 phaste this url

**MAVEN PATH**

Name MAVEN\_HOME

Uncheck the Install automatically

/usr/local/maven

Apply and save

Need to install **maven integration**

Need to install git **yum install git -y**

Goto managejenkins 🡪 manage Jenkins 🡪 Available search **maven integration**

Select new item

**Enter an item name** any name (maven\_project)

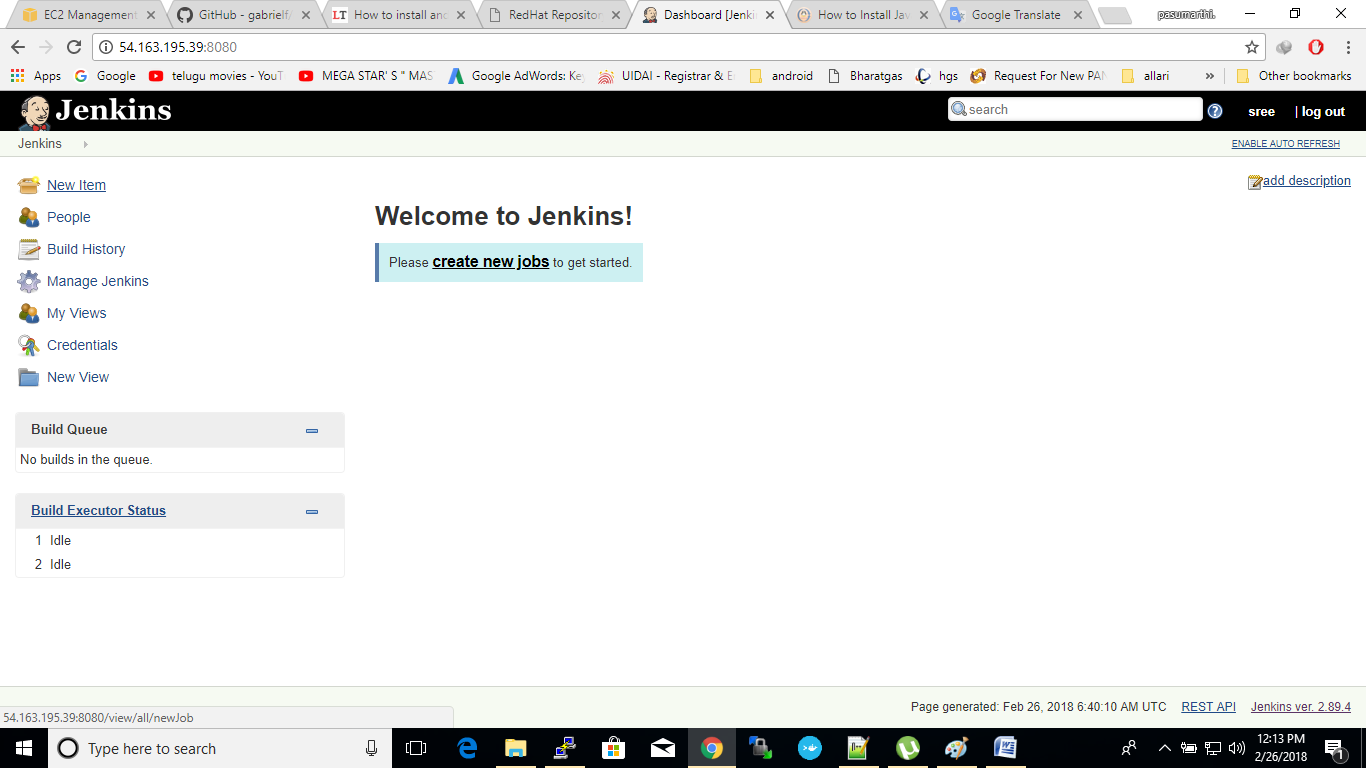
Select maven project then click on ok

Select git and phaste git url

**Build**

Goals and options clean install or clean package and save and apply

Select Build now

****

**APACHE INSTALLATION**

Go to[**https://tomcat.apache.org/**](https://tomcat.apache.org/)select any one version

Click on tomcat 7 and select zip (copy link address)

Apache default port 8080 if u want to change port number go to **cd apache-tomcat-7.0.85 conf/**

**ls**

**vi server.xml**

Edit server.xml.

Search "Connector port"

Replace "8080" by your port number. Like 8088 or 9090

**wget** [**http://www-eu.apache.org/dist/tomcat/tomcat-7/v7.0.85/bin/apache-tomcat-7.0.85.zip**](http://www-eu.apache.org/dist/tomcat/tomcat-7/v7.0.85/bin/apache-tomcat-7.0.85.zip)

**unzip apache-tomcat-7.0.85.zip**

**ls**

**cd apache-tomcat-7.0.85**

**cd conf/**

**vi tomcat\_user\_xml and remove that code and phaste below code**

**<?xml version="1.0" encoding="UTF-8"?>**

**<tomcat-users>**

**<role rolename="manager-gui"/>**

**<role rolename="manager-script"/>**

**<role rolename="manager-jmx"/>**

**<role rolename="manager-status"/>**

**<role rolename="admin-gui"/>**

**<role rolename="admin-script"/>**

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

**</tomcat-users>**

**cd bin/**

**chmod 777 catalina.sh**

**./** **catalina.sh start**

**Goto browser IPADDRESS:8080 ex** [**http://34.227.148.41:8088**](http://34.227.148.41:8088)

**War file deploye in tomcat through Jenkins**

Go to manage Jenkins 🡪 manage plugins🡪available and search Deploy to container and

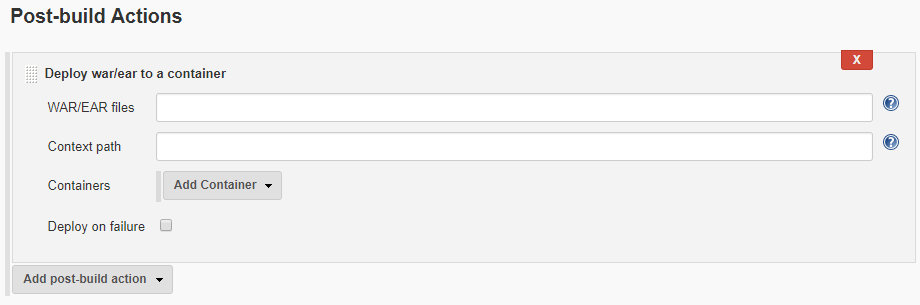
Install

Create new maven project

Goto **POSTBUILDACTIONS**

Click Add postbuilds-action

Select Deploy war/ear to a container



WAR/EAR files \*\*/\*.war

Context path any name ex sri

Containers click Add Container then select tomcat version

Credentials click Add select Jenkins

Username: admin

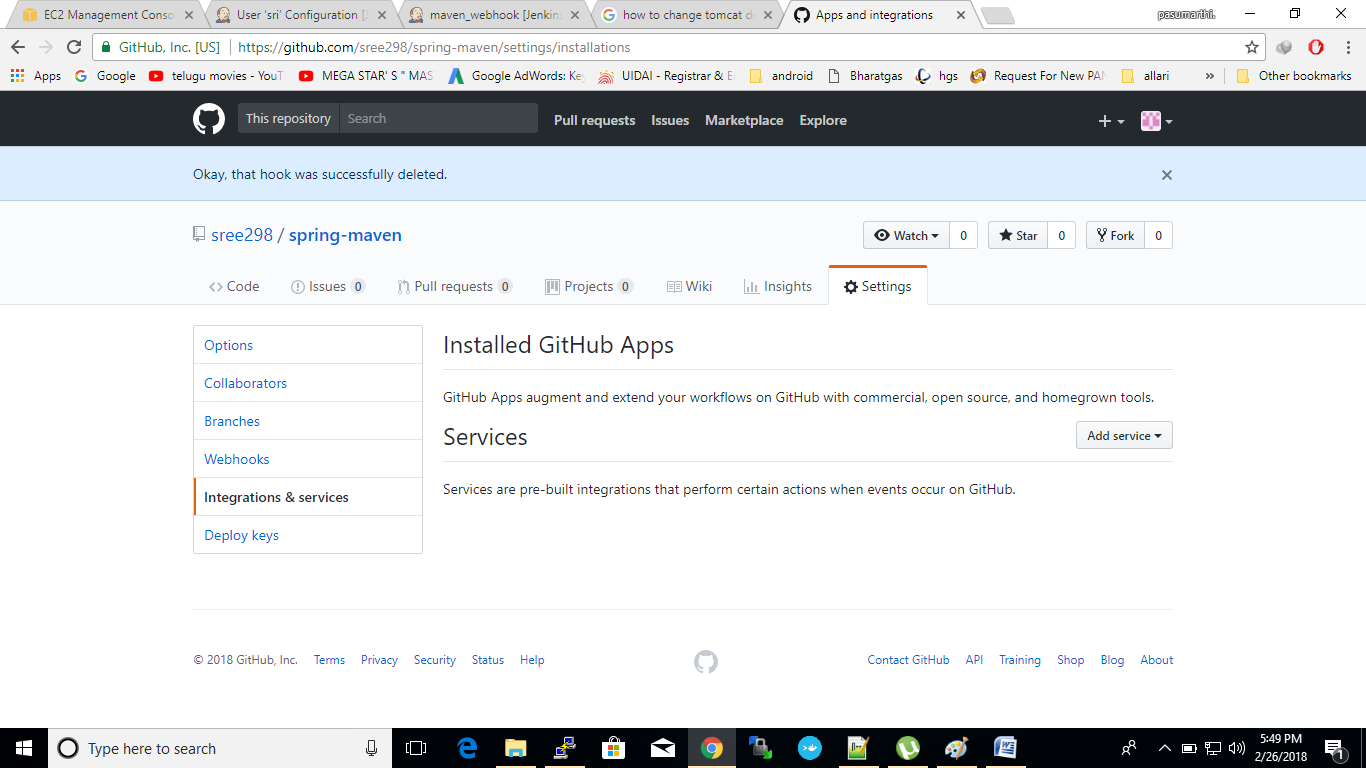
Password: admin click on add in Credentials select admin

Tomcat URL <http://34.227.148.41:8088>

**When we code commit new build trigger process**

Go to github server

Select integration&service

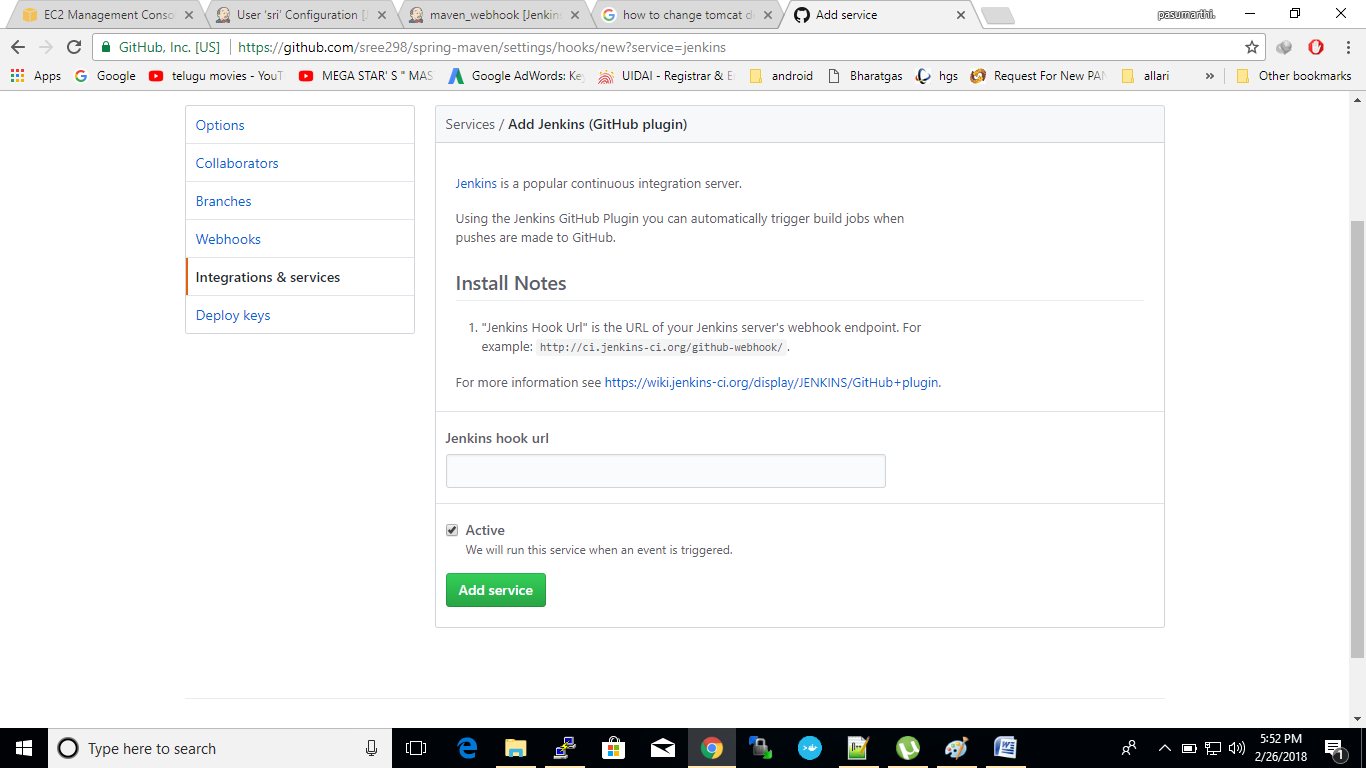


Click on Add Service and search Jenkins github pligin

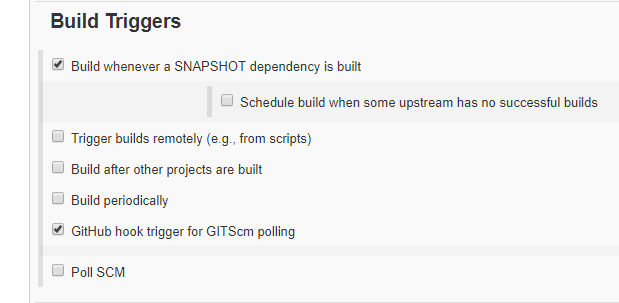
**Jenkins hook url** <http://jenkinsurl/github-webhook/>

Ex <http://34.227.148.41:8080/github-webhook/>

Click on Addservice



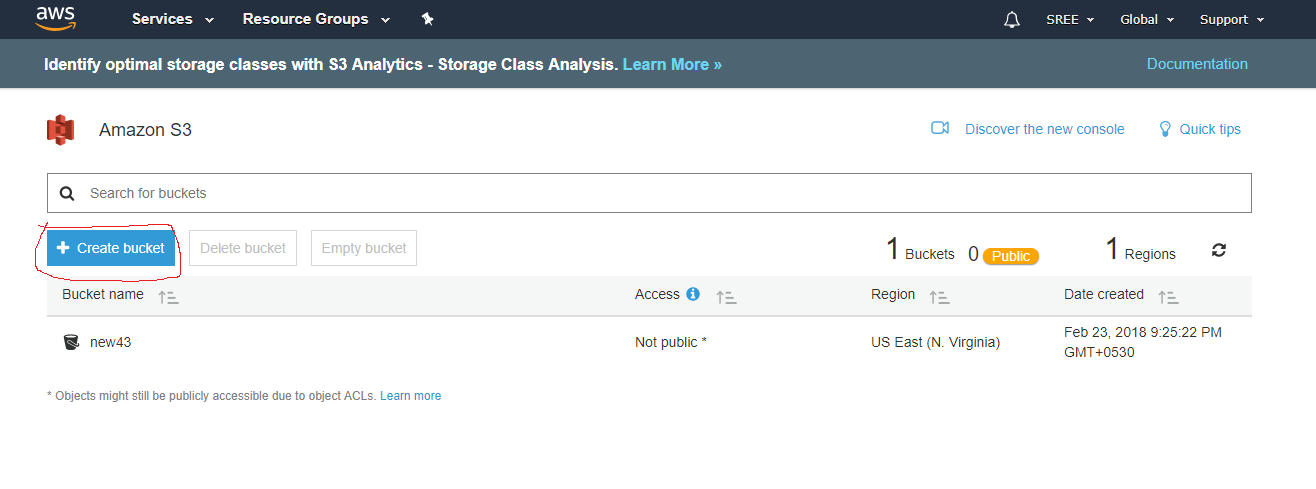
Create new maven project and select GitHub hook trigger for GitScm polling

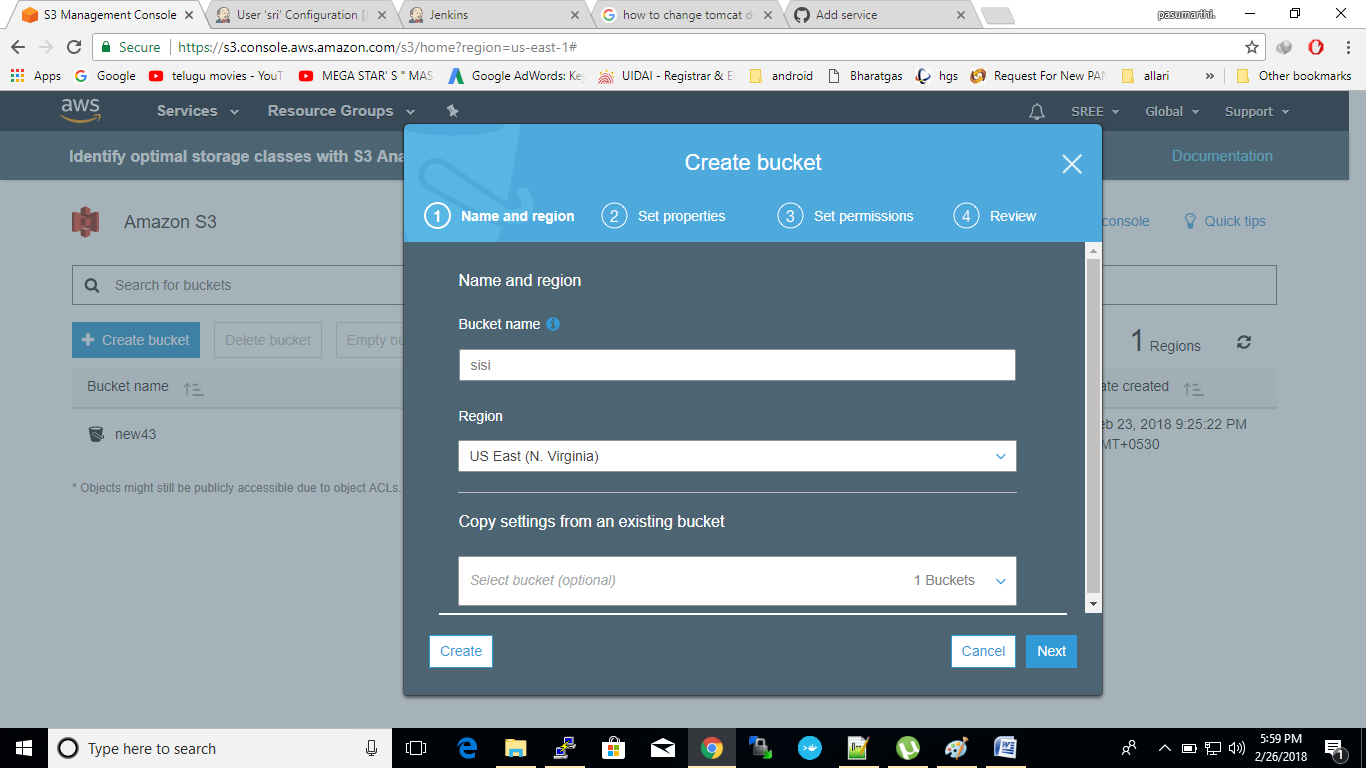


**War file store in s3 bucket through Jenkins**

Go to managejenkins🡪managepulgin search S3publisher plugin and install

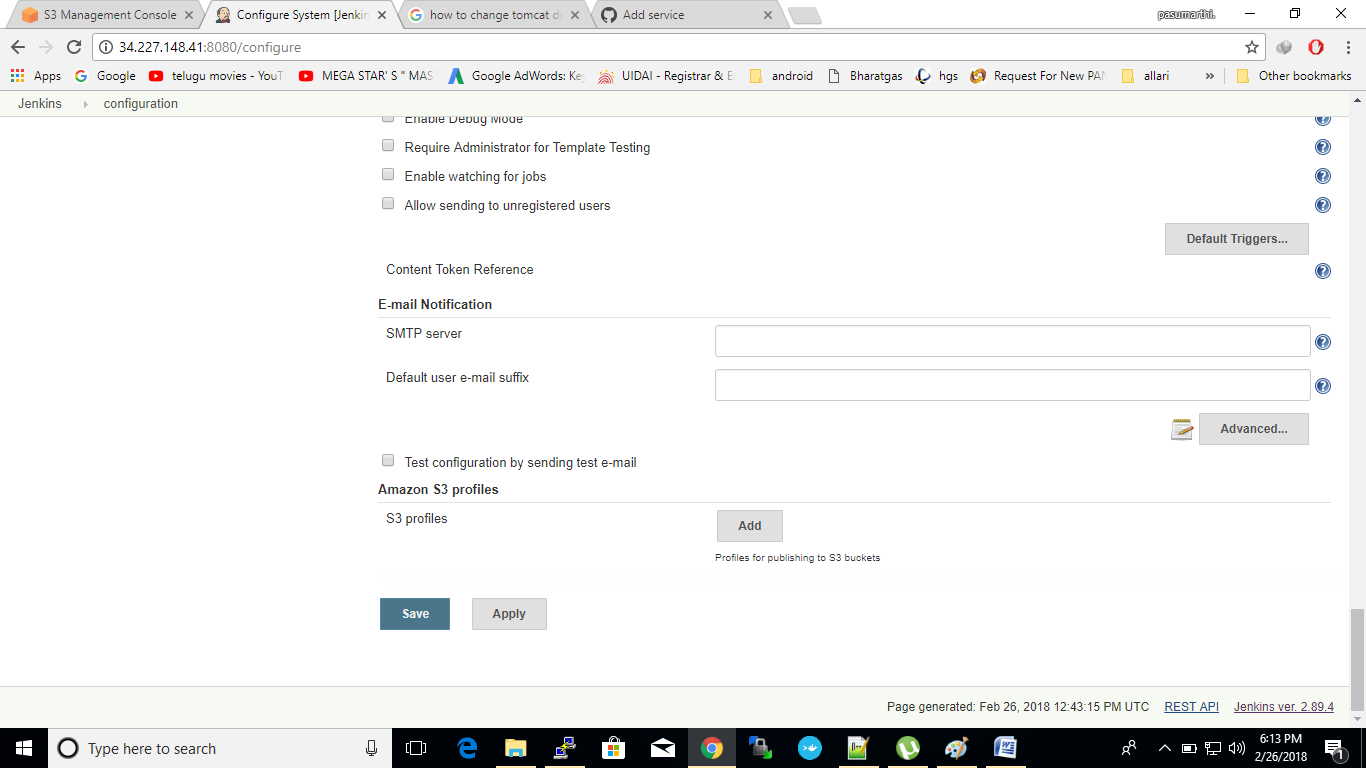
Create new bucket in aws



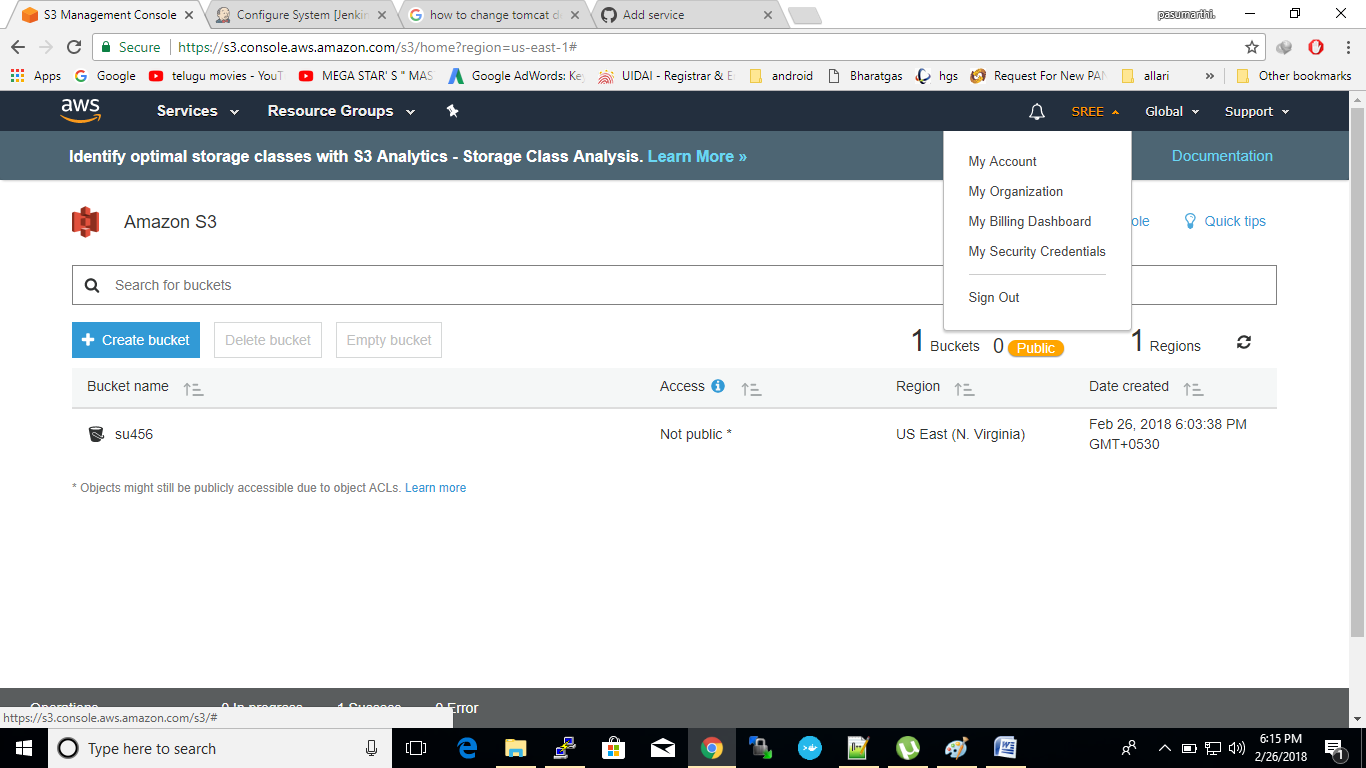


Click on next and select versioning and then select enable versioning and click on next and click on next then click on create bucket

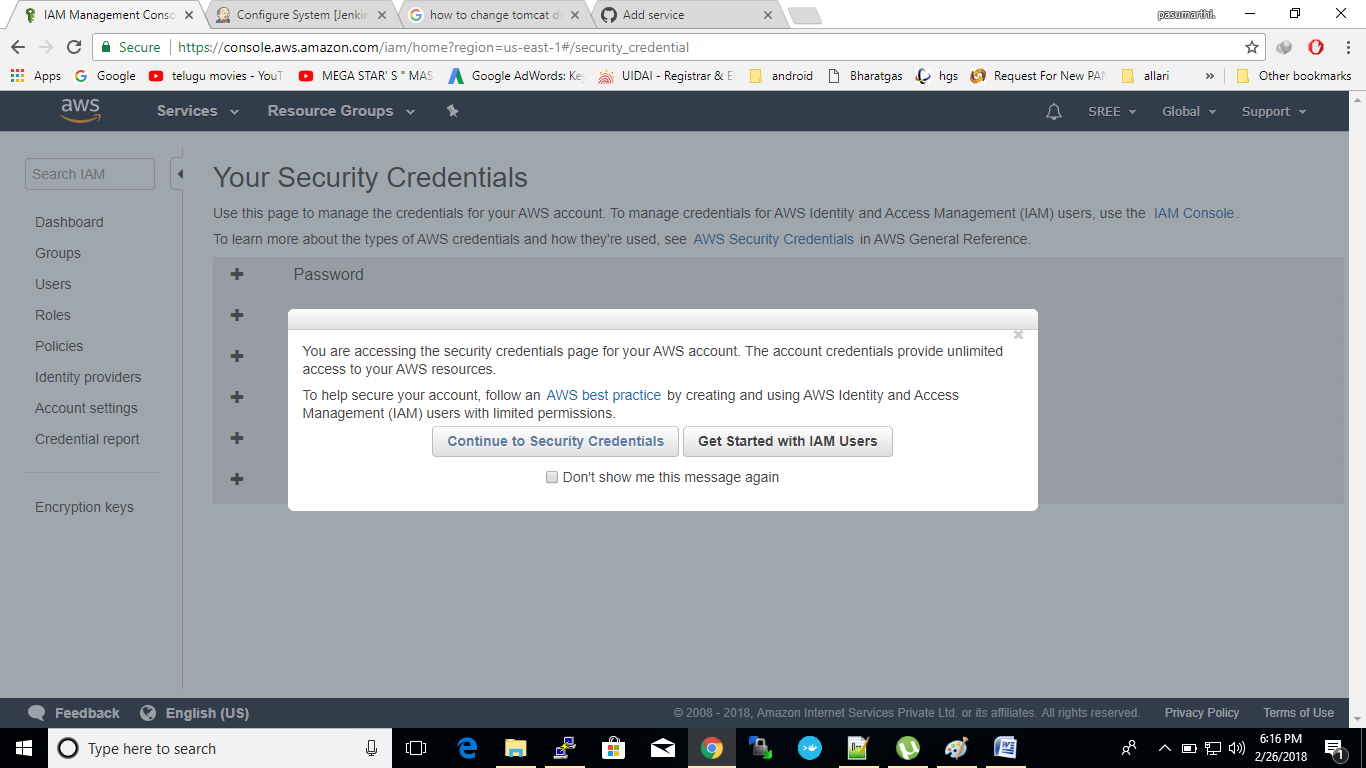
Go to managejenkins 🡪 configure system



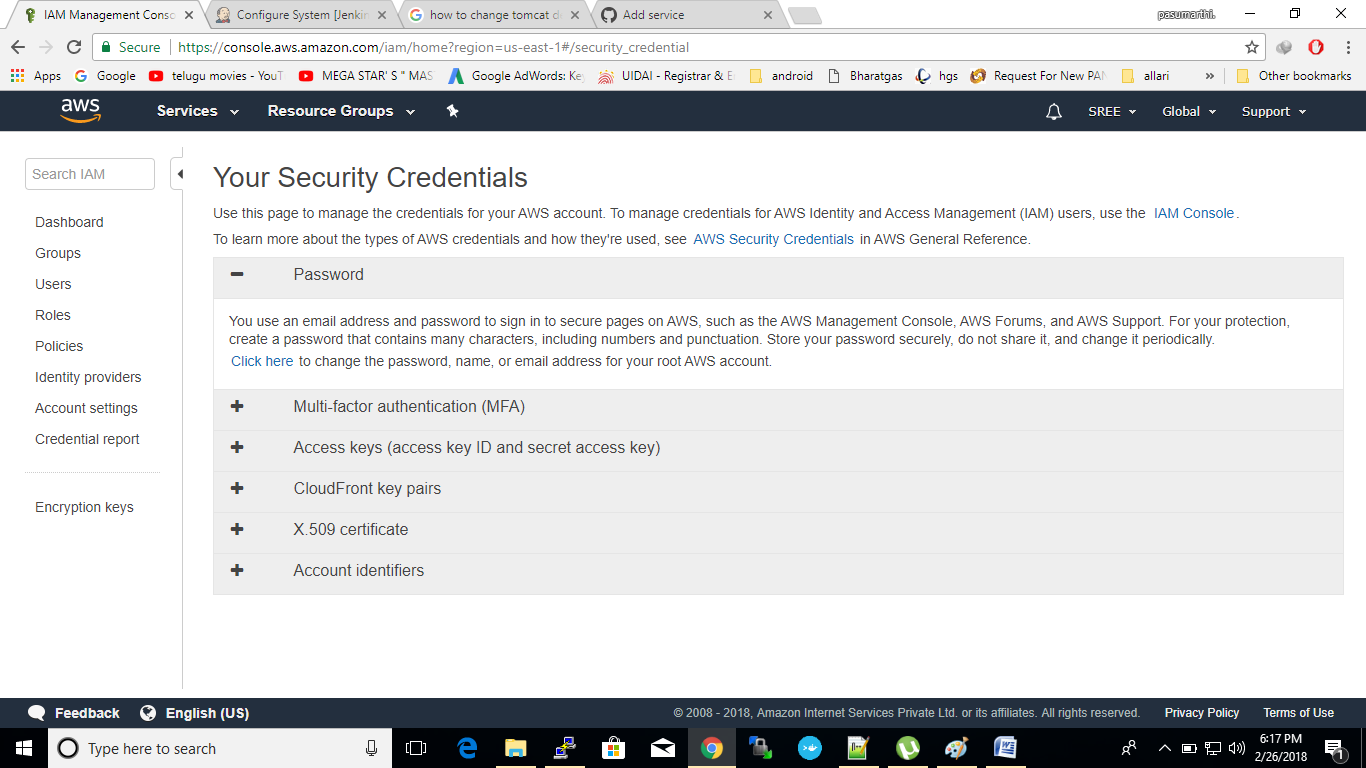
Click on Add in s3 profile



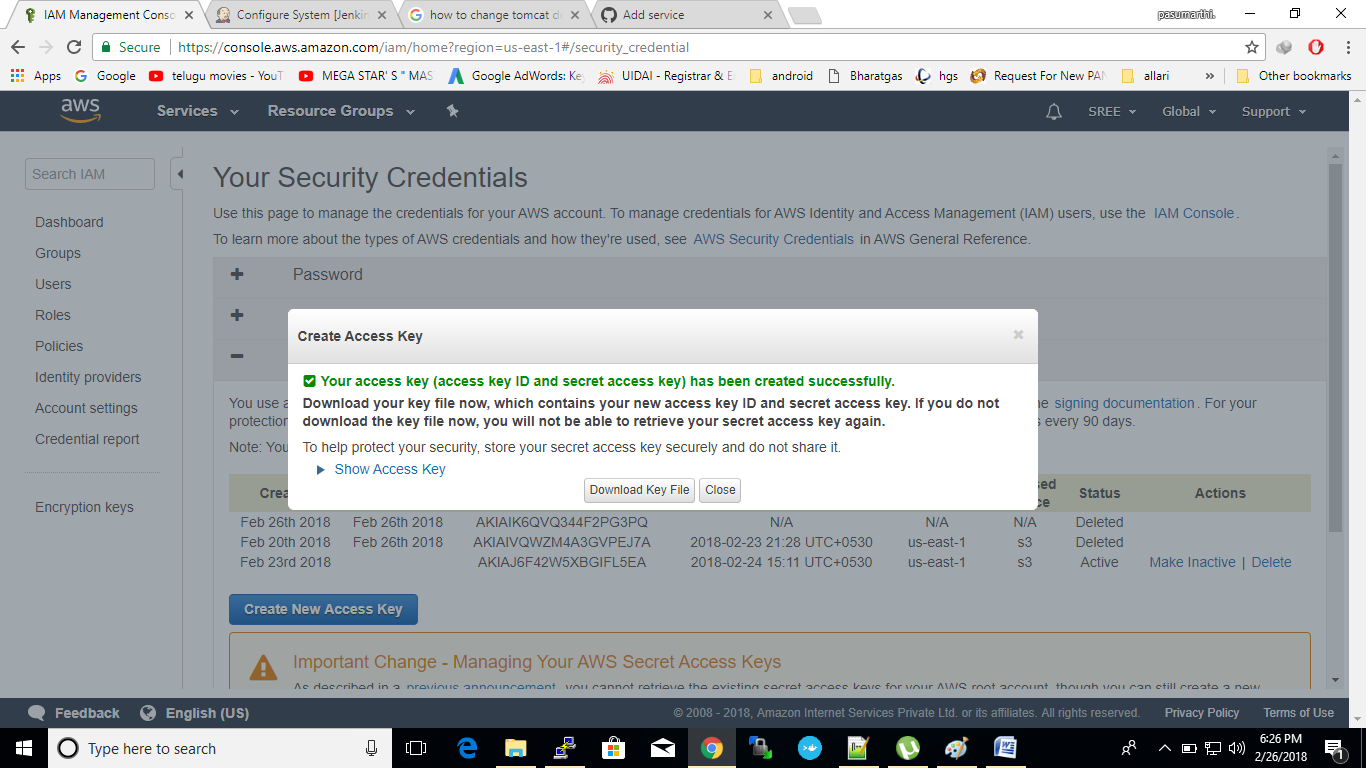
Go to mysecurity Credentionals



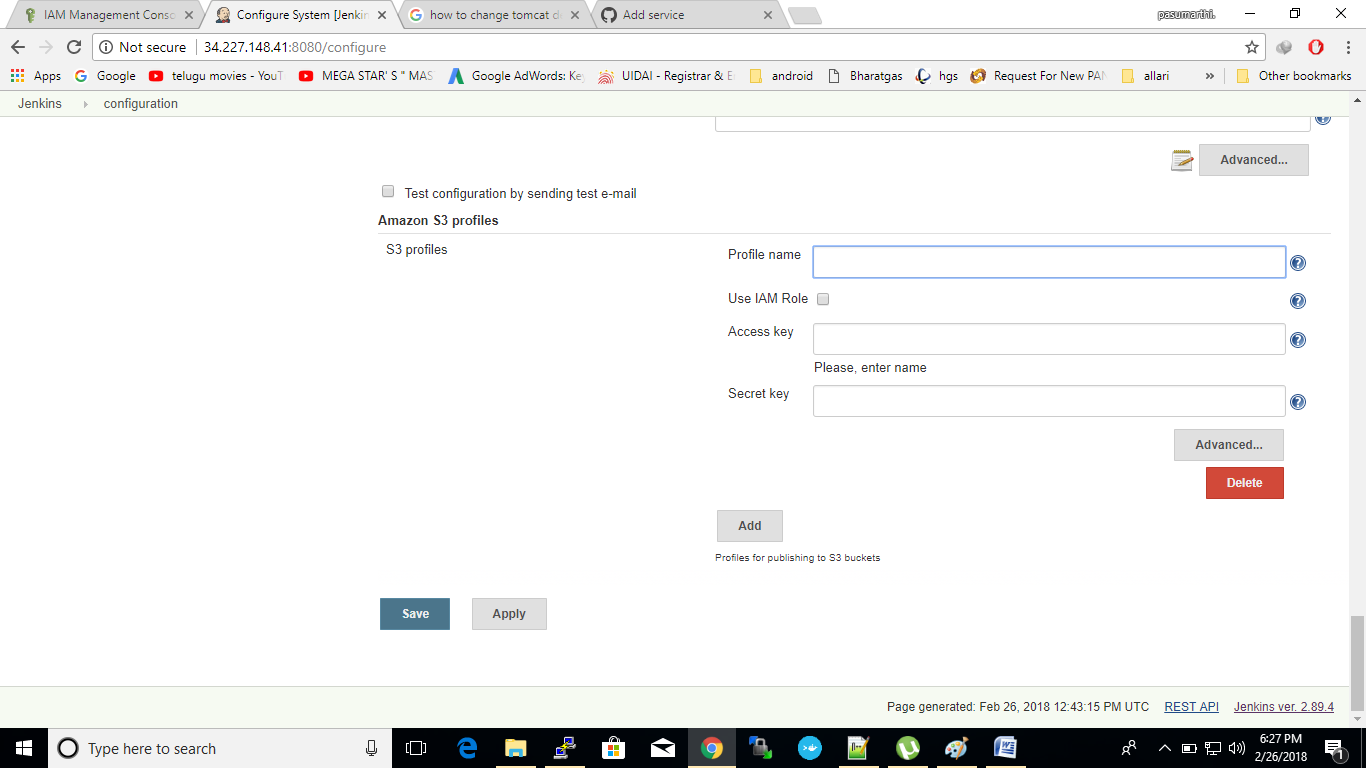
Select continue security Credentials



select Access keys (access key ID and secret access key) and then click on create new access key



Click on show access key or Download Key File



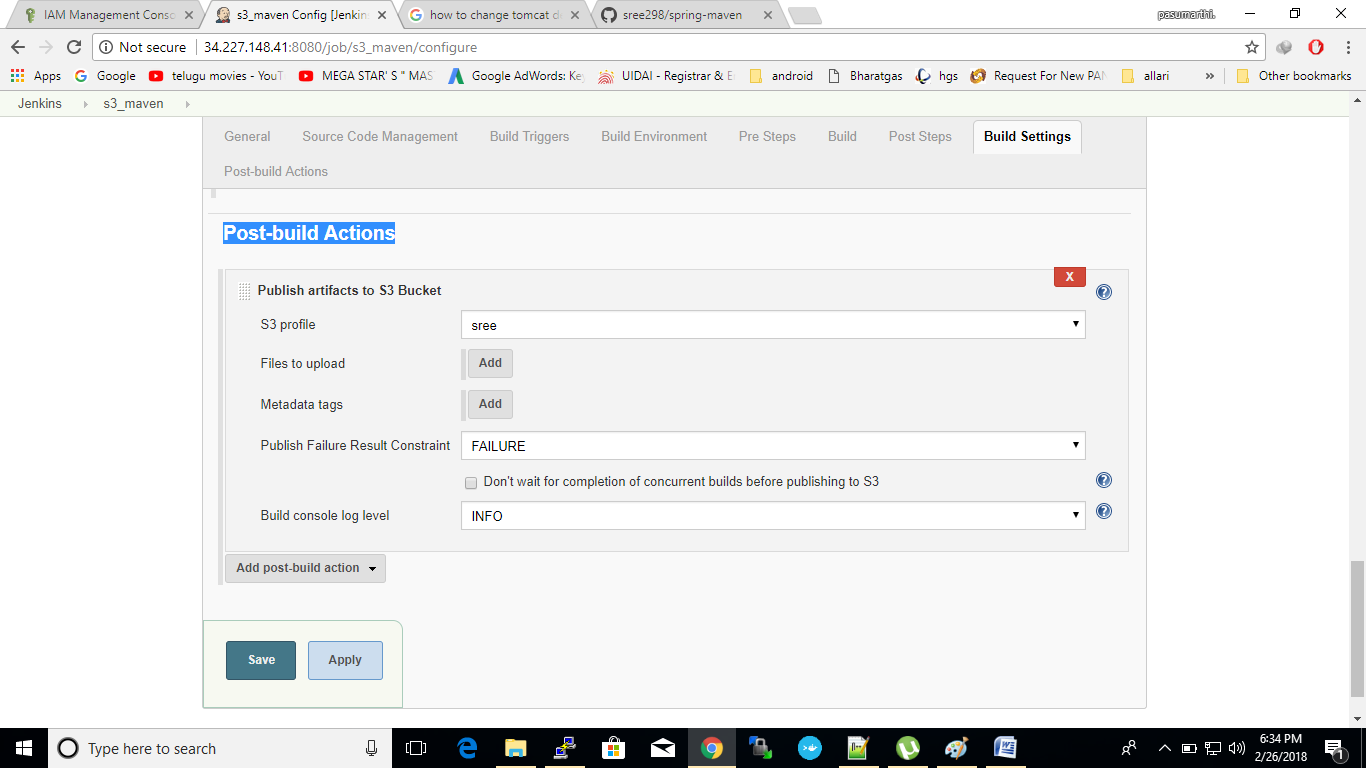
Profile name any name like sree

|  |  |
| --- | --- |
| Access | Key copy the access key and security key and save |
|  |
|  |  |

Create new maven project

Go to **Post-build Actions**

Select publish artifacts to s3 bucket



Click an add

Source \*\*/\*.war

Destination bucket bucketname ( su456)

Bucket name su456/${JOB\_NAME}-${BUILD\_NUMBER (when you build now then not override in s3 create new war file)

Bucket Region select your aws bucket region like

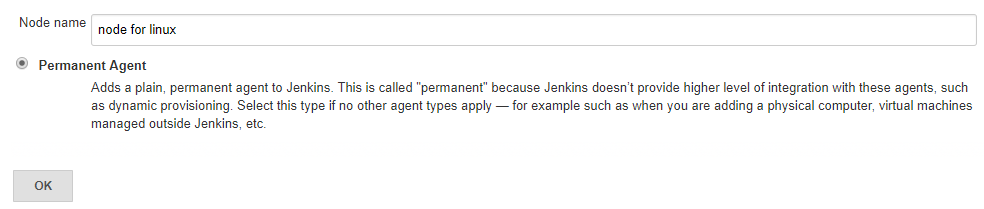
Click on save

Build now check in your s3 bucket

**Slave**

Go to managejenkins🡪managenodes and click on new node then give node name(ex node1)

Click on Permanent Agent then click on ok



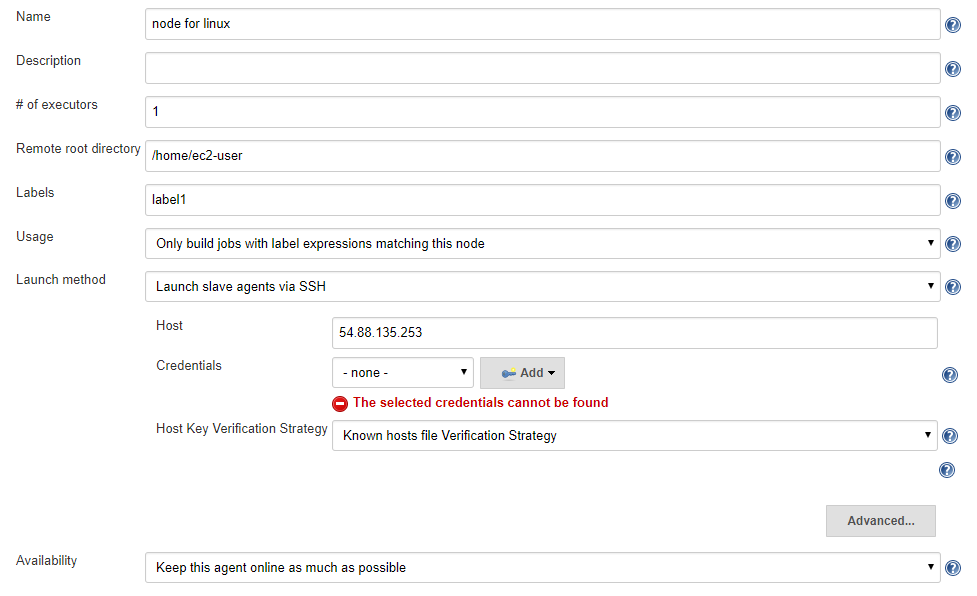
Executors : one or morethan one

Remote root directory : /home/ec2-user (depends on mechine)

Label : any name

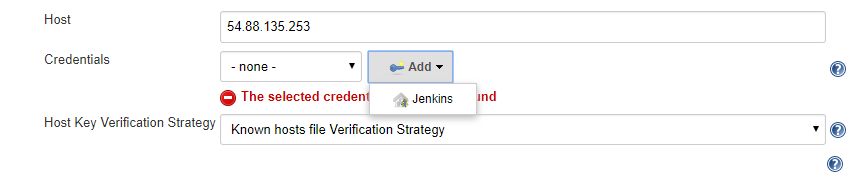
Usage : only build jobs with label expressions matching this node

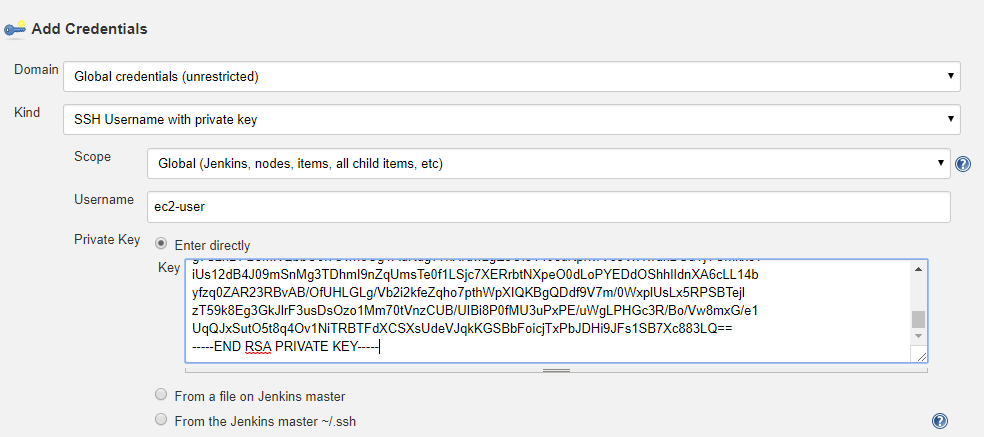
Launch method : Launch slave agent via SSH



Click on Add then select Jenkins

Give the your instance or node IPADDRESS



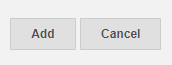


Select in kind : SSH Username with private key

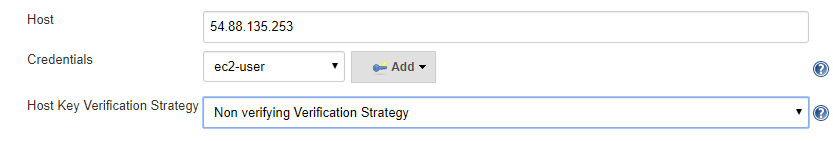
Usename : ece2-user

Private key : select Enter directly and open your pem key copy and phaste in entry directly

Click on Add



Select ec2-user in credentials



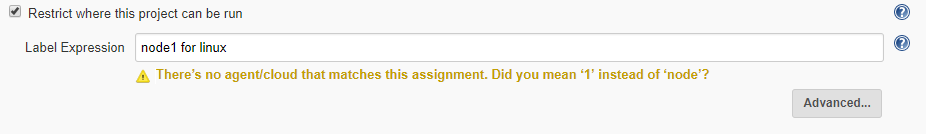
Non verifying Verfication Strategy option select in Host key Verification Strategy.

Then after create new project maven or freestyle

We need to install the java and maven in node server

Select Restrict where this project can be run

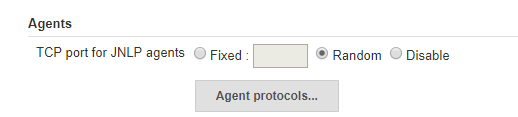
Label Expression: node name or label name



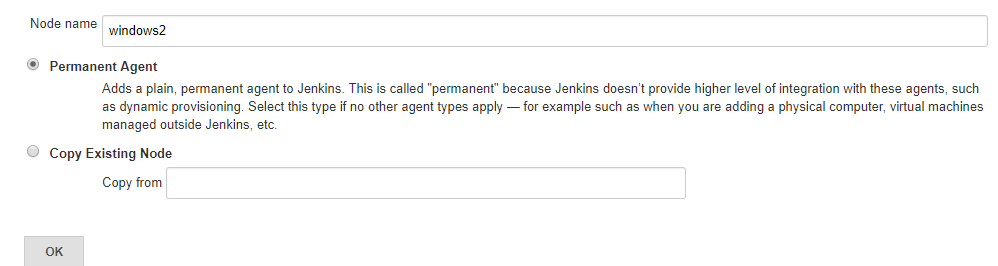
After build now check the node server

**SLAVE FOR WINDOWS**

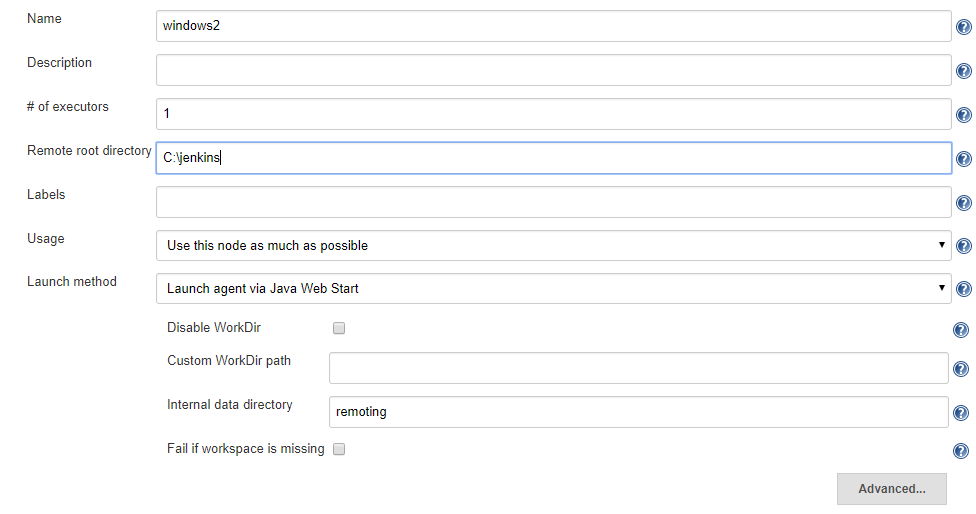
Goto managejenkins🡪 configure global security click on Random



ManageJenkins🡪managenode click on new node and give the node name

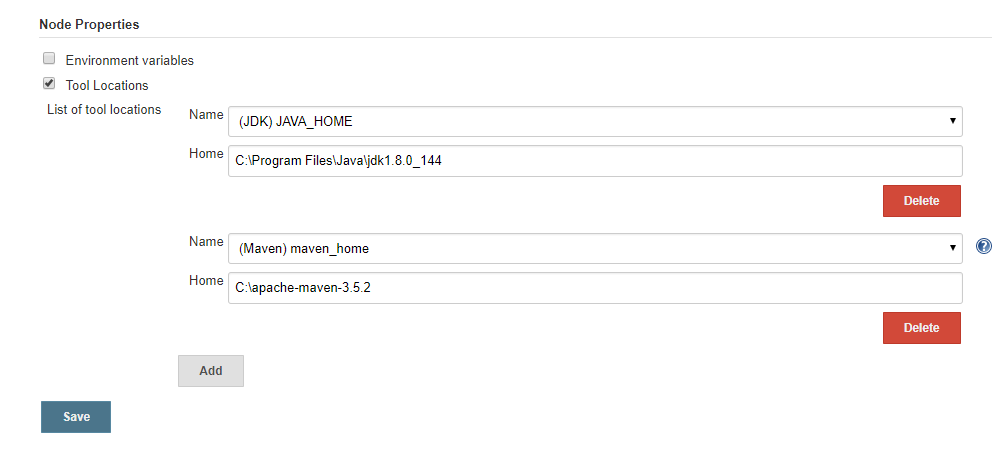


Click on ok



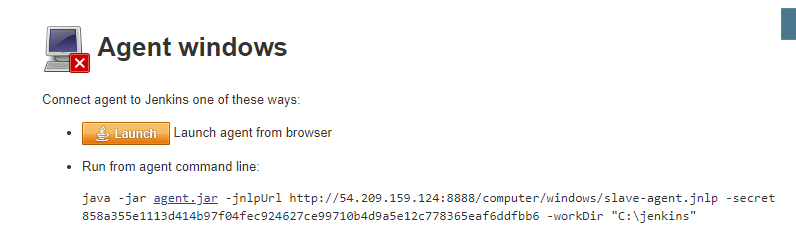
Remote root directory : C:\jenkins

Launch method : Lanch agent via Java Web Start



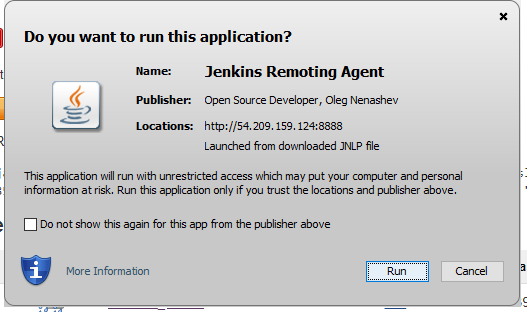
Select Tool Location

We need to add the javahome and maven home path (need to give slave paths)

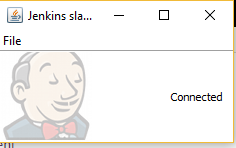


Click on  or copylinkaddress phaste in any browser

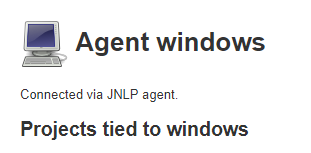
Then download click on that one get the below window



Click on run after that getting one window with port number that port number open In instance



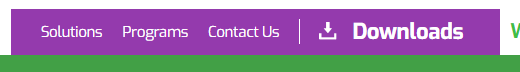
After refresh get the connection establish



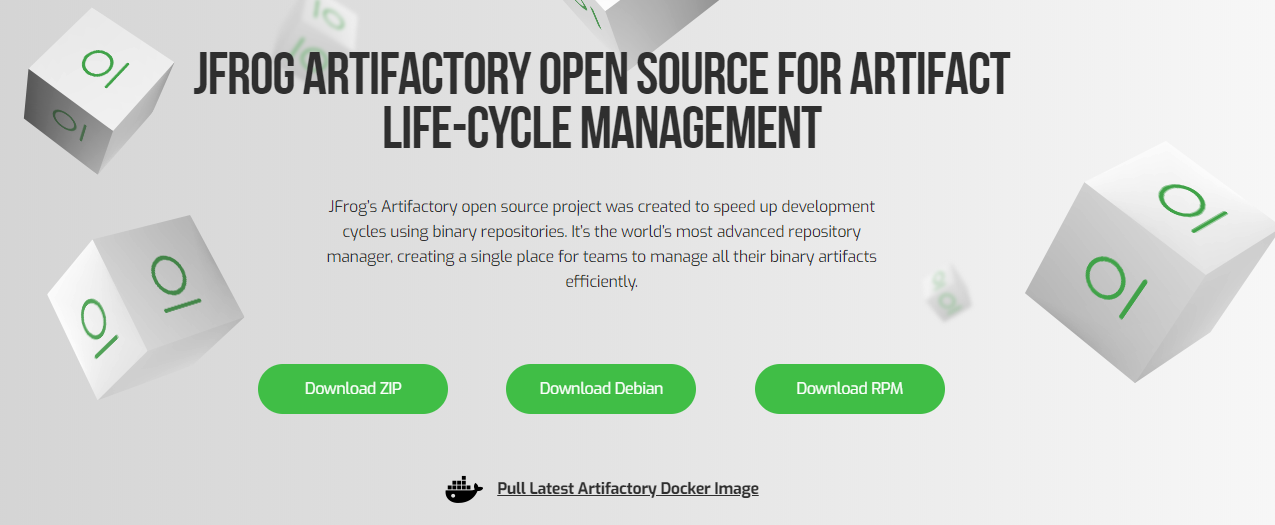
**JFROG**

Search jfrog open sorce or <https://jfrog.com/open-source/>

Click on download



Then after click on download Zip



After downloading copy that zip windows to aws mechine through winSCP

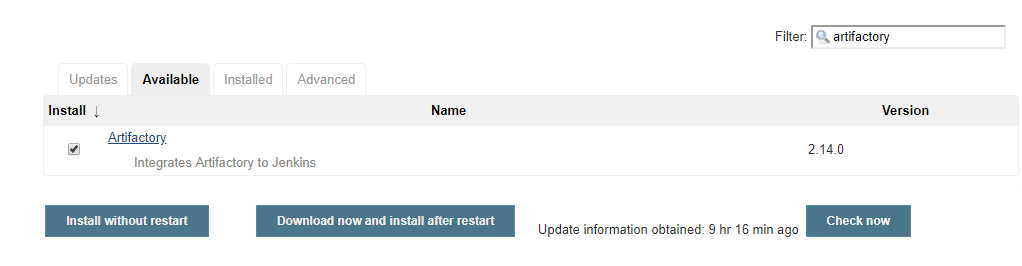
Unzip

IPADDRESS:8081 in browser

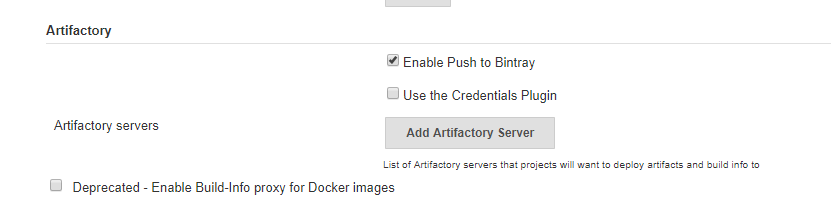
Ex: <http://34.226.139.26:8081>

Click on login then give the username : admin and password : password

Goto managjenkins🡪manageplugun🡪Available and search artifactory plugin



Goto managejenkins🡪 configure System



Click on Add Artifactory servers

Server Id : any name

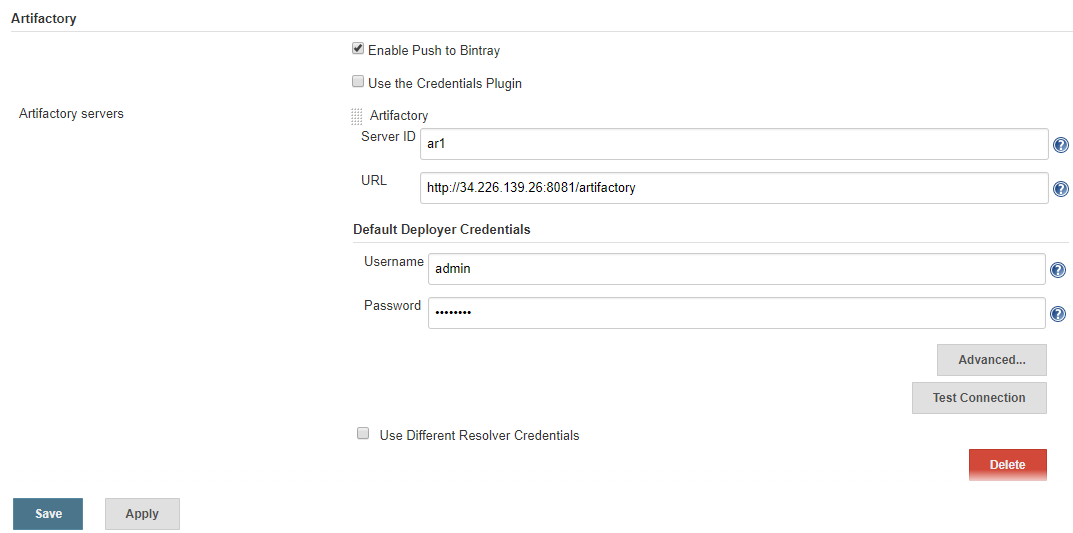
url IPADDRESS:8081/artifactory

Ex : <http://34.226.139.26:8081/artifactory>

Uername : admin password : password

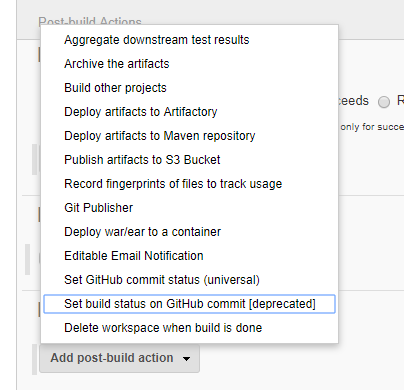
Clikc on Test connection display the Found Artifactory 5.9.0 ( when u enter correct username and password )

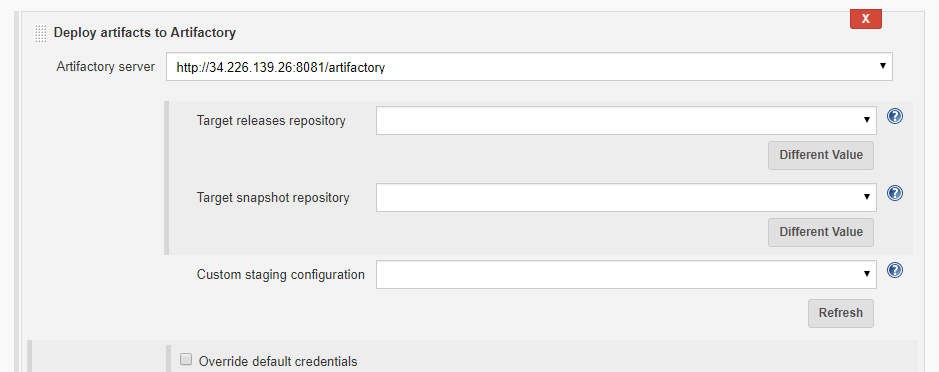
Click on save



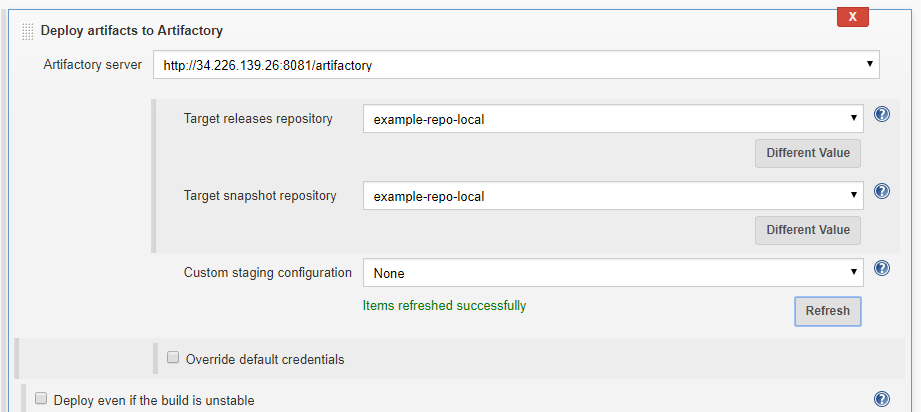
Create one maven project

Select **Deploy artifacts to Artifactory** in Add post build action





Click in refresh then getting information(Target releases repository ,snapshot repository)



Deploy Maven artifacts

Include Pattern \*\*/\*.war

|  |
| --- |
|  |
|  |  |

Then click on build now

**AMAZON EC2 PLUGIN**

