Continues integration and continues deployment

Pre-requisite:

1.GitHub for continues development

2.Jenkins for continues integration

3.S3 for storage

4.Docker for containerization

5.Chef for continues deployment

Description:

Deploying .war files we need Jenkins for that we need to setup the Jenkins services. Before installing the Jenkins in our server, we need to install java on that server because Jenkins in dependent on java. For java installation we follow the bellow commands.

* 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/8u151-b12/e758a0de34e24606bca991d704f6dcbf/jdk-8u151-linux-x64.tar.gz>
* tar xzf jdk-8u151-linux-x64.tar.gz
* cd /opt/jdk1.8.0\_151/
* alternatives --install /usr/bin/java java /opt/jdk1.8.0\_151/bin/java 2
* alternatives --config java
* here we need to enter 2
* alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_151/bin/jar 2
* alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_151/bin/javac 2
* alternatives --set jar /opt/jdk1.8.0\_151/bin/jar
* alternatives --set javac /opt/jdk1.8.0\_151/bin/javac
* java -version

Now we need to setup Jenkins server for that following are the commands:

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

yum install jenkins

Know we need to start the Jenkins server. After then go to aws console open 8080 port because Jenkins default port. Login to the Jenkins server.

To integrate git in Jenkins we need to install git on Jenkins cli.

* Yum install git -y

Take another server in that we need to install docker. For docker installation we use this command

* curl -fsSL get.docker.com -o get-docker.sh

To execute this shell script we execute this command

* sh get-docker.sh

Now we need to check the docker version by using this command

* docker –version

First we need to install ubuntu image on the docker engine.

Write a docker file for our customized image. In that docker file we need to write the following script.

* Vim Dockerfile

From ubuntu

RUN apt-get update -y && apt-get install git -y

RUN apt-get install default-jre -y

RUN apt-get install default-jdk -y

RUN apt-get install vim -y

RUN apt-get install sudo -y

RUN apt-get install wget -y

RUN apt-get install tar -y

RUN wget http://redrockdigimark.com/apachemirror/tomcat/tomcat-7/v7.0.82/bin/apache-tomcat-7.0.82.tar.gz

RUN tar xvzf apache-tomcat-7.0.82.tar.gz

EXPOSE 8080

To build our customized image on our docker file we need to execute this command

* docker build -t tom1 .

Note:In above command we need to put . it represent the current working directory and the docker file is present in that directory

Now we need to run the container on our customized image (tom1) to run the container we use this command

* Docker run -dti --name tomcnt -p 9090:8080 tom1 /bin/bash

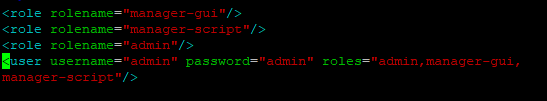
Now we need to login to the container for that the command is

Docker attach tomcnt

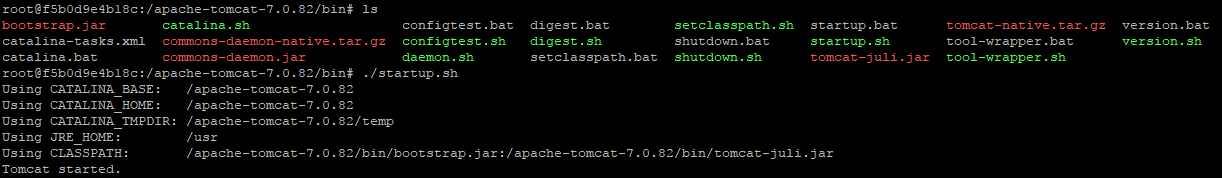
Now we are logged into the container now we can see that tomcat is installed on our container on we need to start the tomcat service for that we go to following directory.

we need to create the tomcat users for that we need to do the following changes in the container

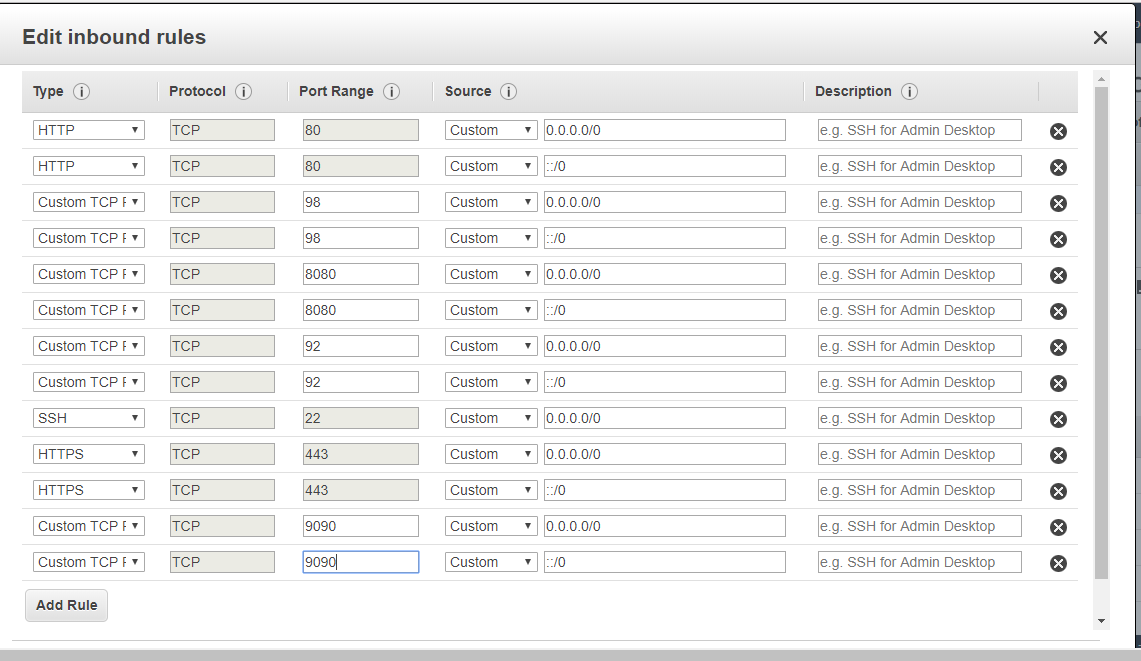
vi /apache-tomcat-7.0.82/conf/tomcat-user.xml

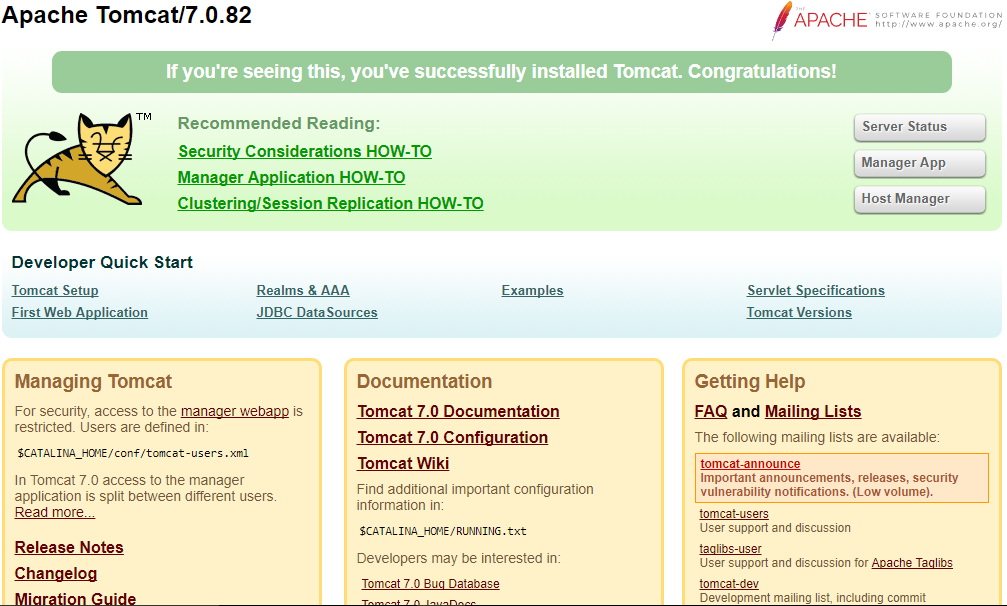


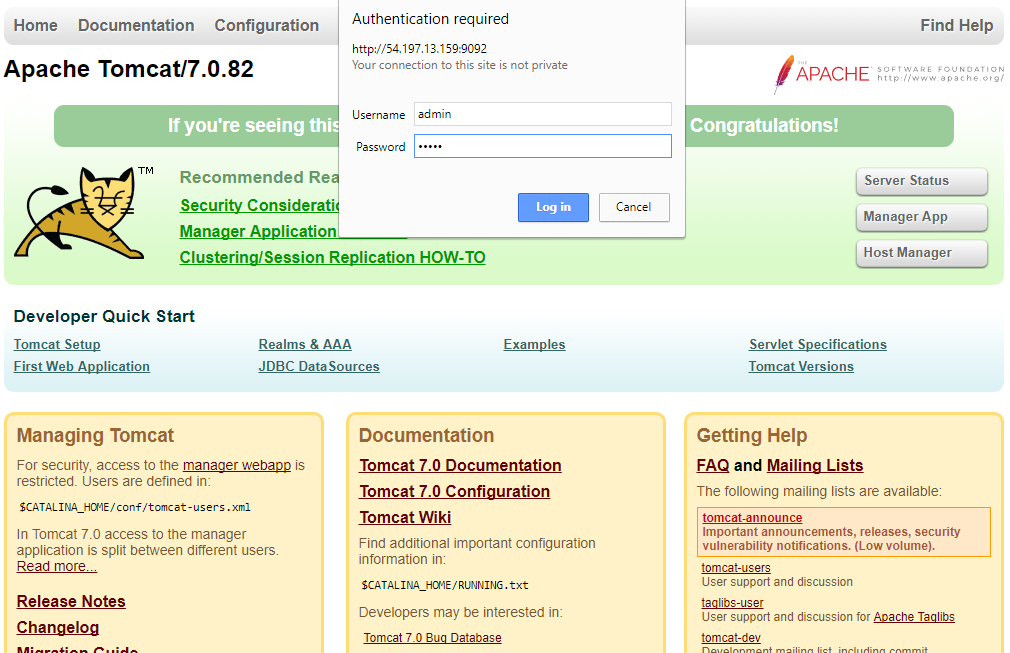
After creation of users we need to start the tomcat service.



After completion of this we need to go to aws console and open the port 9090



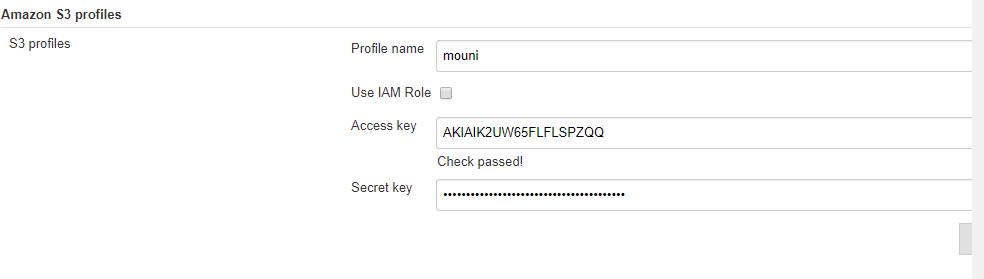




Now we need to configure the Jenkins

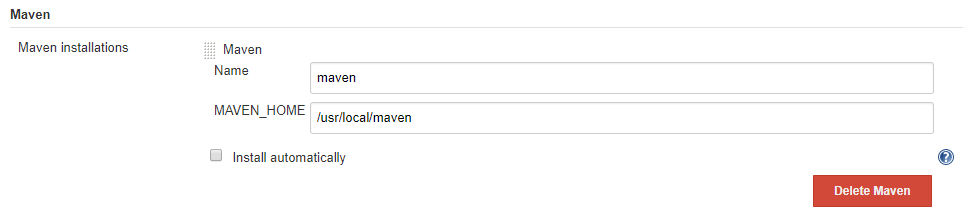
We need to install few plugins like deploy plugin, s3 publisher plugin, maven integration plugin.

Go to configure system do the following below



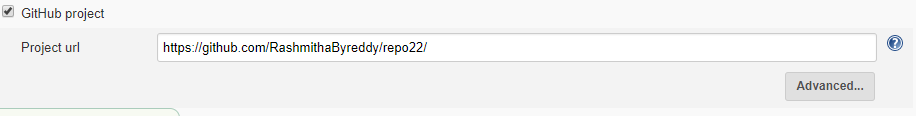
After completion of this go to global tool configuration setup the maven and jdk path

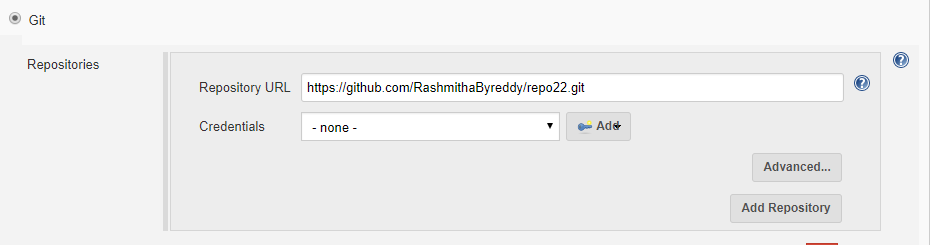




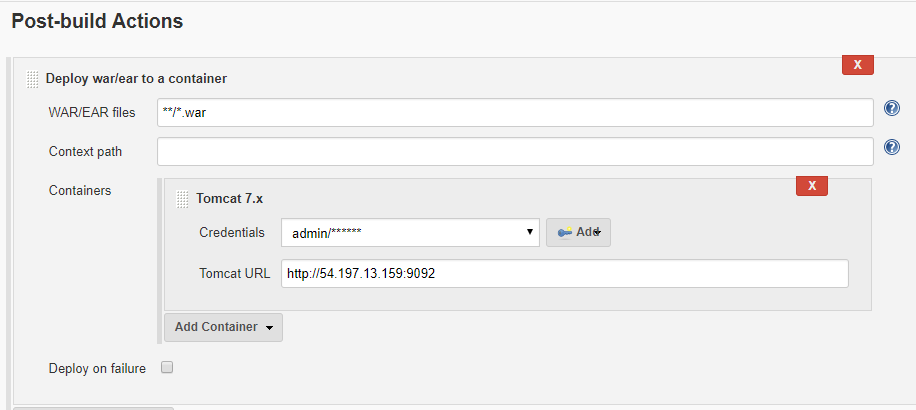
Apply and save

Now we need to create a maven job in that configure the steps given below





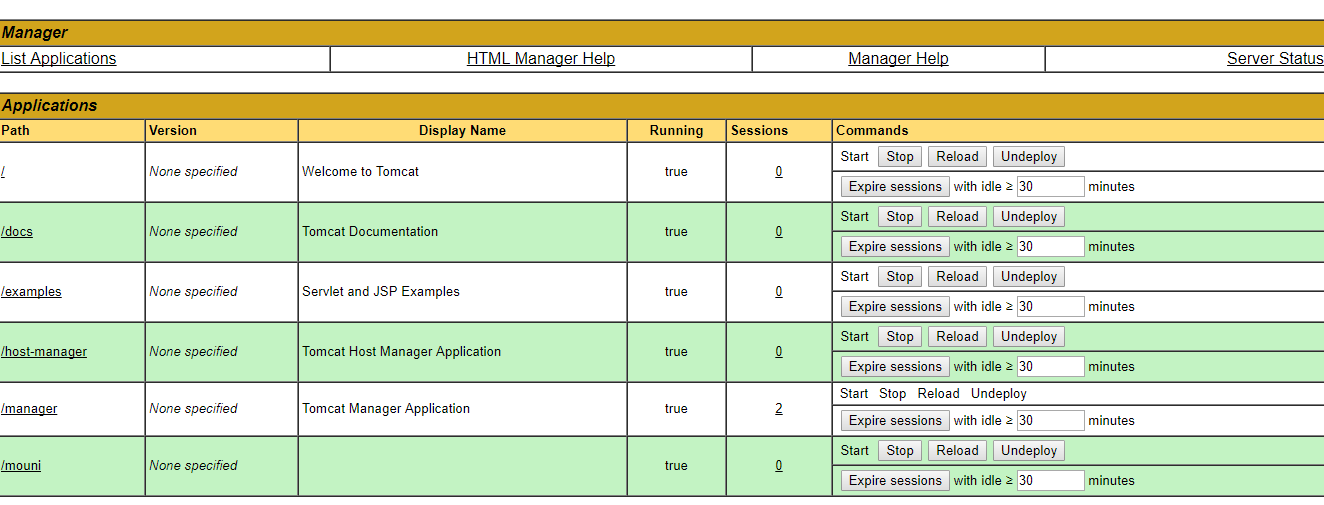




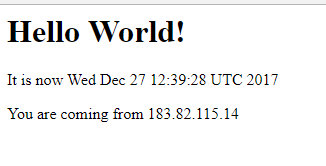
Apply and save

Finally build the job

Now our war file is deployed in the docker container



Here we can see the mouni folder in that our war file is deployed



Here we need to deploy this war file into the production servers for that we are using chef.To get that war file into the chefworkstation we need to copy the url of the s3 bucket and type this command

Wget url of the bucket

Now our war file present in chef-repo. After that move our war file into files by using move command. Write the recipe as the following

package 'apache2' do

action :install

end

service 'apache2' do

action :start

end

cookbook\_file '/apache-tomcat-7.0.82/webapps/manager/WEB-INF/hello-world-war-1.0.0.war' do

source 'hello-world-war-1.0.0.war'

end

Now upload the cookbook and add to the run list

Bootstrap the node then our war file is deployed into the production servers

Note: In our nodes tomcat should be installed