**CI/CD Assignment**

**Deployed simple web application on to the AWS ec2 instance under ALB via Jenkins**

Prerequisites for deploying an application

1. Initialized 3 Aws instances with 4 CPU 16GB RAM
2. Created Helloworld.war web application
3. Configured Tomcat using Docker file
4. Created Application Load Balancer

**Steps for Continuation Integration and Deployment:**

For continuous integration and continuous deployment, we do have different approaches. We can automate the process using the pipeline concept. Here I have taken the simple approach as it our testing. I have taken Maven Project job and configured and deployed our application in the Aws server without any manual intervention with using the shell commands and SSH plugin.

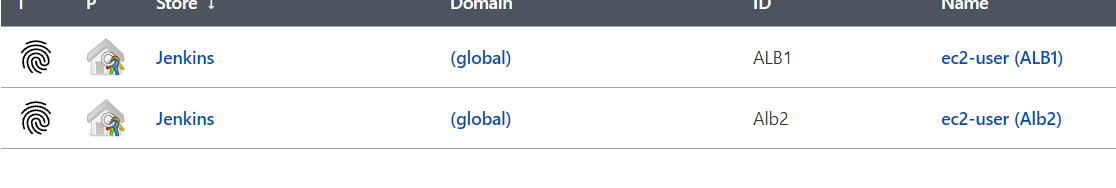
I have used 3 EC2 instances one for Jenkins’s server and another two is for deploying the web application and attaching them to Application Load Balancer so that traffic can distribute to the registered targets and to monitor the health of the registered targets.

1. Initialized one ec2 instance and installed &configured Jenkins’s server on top of that.
2. Created a Job with using Maven project and installed Git and Maven plugin in the Jenkins to build the job.
3. Written a Tomcat Dockerfile and uploaded in the remote repository Github.
4. Created a repository in the dockerhub for building and pushing the image into it.
5. Using the Execute shell in Jenkins in the post steps, builded the image and pushed into the docker hub.

Continuous Deployment:

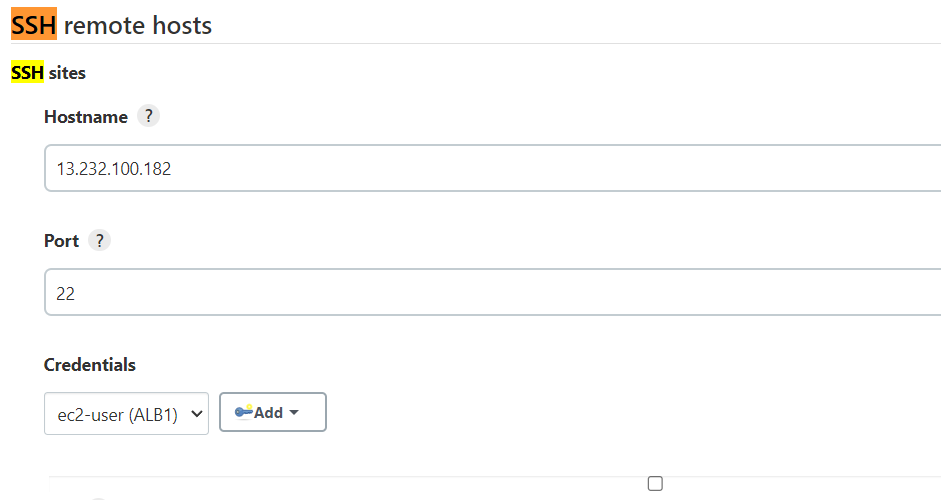
1. Installed SSH Plugin in Jenkins and add the credentials of another two Ec2 instances which named as ELB1 and ELB2. I have used kind: SSH username with private key. Private key has generated with the help of putty generator by taking PEM file.

Dashboard>>managejenkins>>manage credentials>>Global Credentials>>Add credentials>>Kind: SSH username with private key >>copied the privatekey>>save.



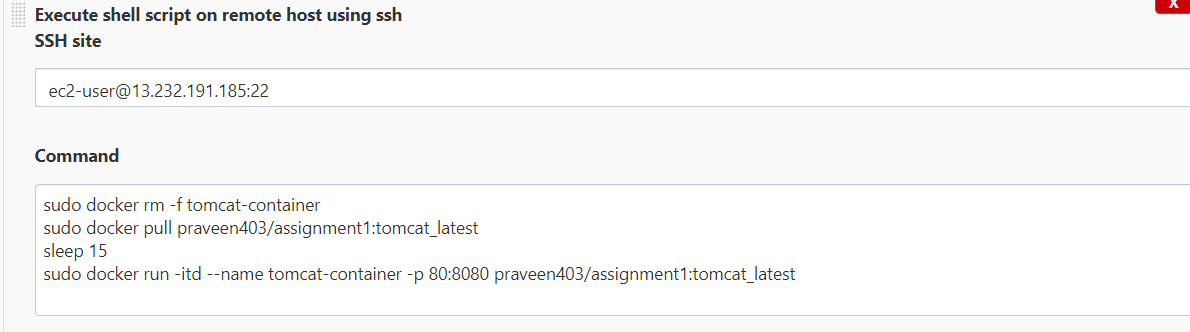
1. Go to configure system and configured the IP Address of ELB1 and ELB2 instances in the SSH remote hosts

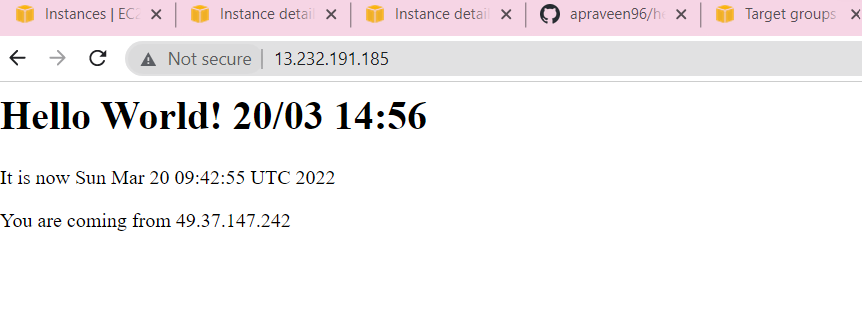
Dashboard>>configure system>>SSH remote hosts>>Ip address, credentials>>check connection>>save



1. Go to configure system of the job and execute the commands in the post build setup and add Execute Shell script on remote host using SSH after installing SSH plugin only we can able to see the Shell script on remote host using SSH.

Now we are connecting to the VM of EC2 instances.

1. Select the IP address with the instance username and use the docker commands to pull the image and create a container.
2. 
3. Select the ip address of EC2 instance with the port 80 and check the application is up and running.





1. Now Attach the EC2 instances to the Application Load balancer.

