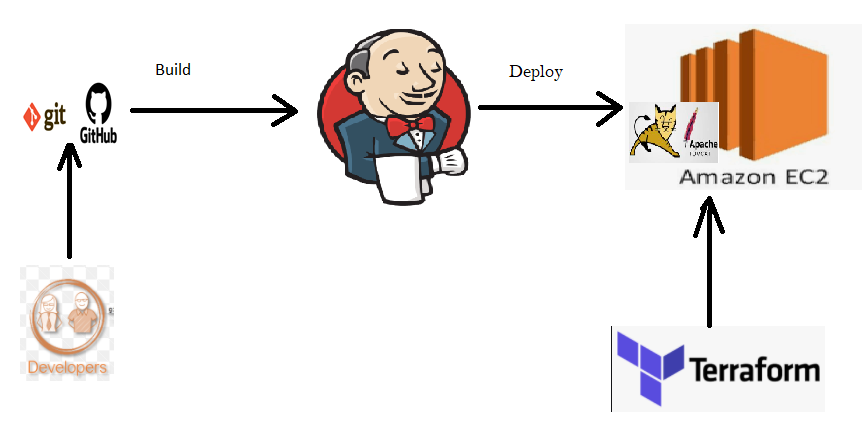
**Hello world web Application deployed on Tomcat using Jenkins in AWS server**



**Tools used:**

* Infrastructure tool - Terraform
* Programming Language - Java
* CI/CD tool - Jenkins
* Build tool - Maven
* Compute Resources - EC2
* SCM - Git hub

**Summary:**

1. Provisioned both EC2 for Tomcat and Jenkins using Terraform scripts and all the necessary tools had been installed by passing into terraform script as a User data shell script
2. Java code has been pulled from my GIT repository (SCM)
3. Code build has been done through maven in Jenkins
4. Deployed to Tomcat Apache server and Hello world Application is accessible on port 8090 on Apache server

**STEP 1 -**

Manually provisioned a Linux EC2 instance as a base machine in AWS Management console in order to install Terraform tool on the same and placed .tf(Terraform files) which provisioned two Ubuntu machine as Jenkins and Tomcat server

1. Please refer the below commands in order to install Terraform on base machine

wget <https://releases.hashicorp.com/terraform/0.14.5/terraform_0.14.5_linux_amd64.zip>

unzip terraform\_0.14.5\_linux\_amd64.zip

sudo mv terraform /usr/bin/ && rm terraform\_0.14.5\_linux\_amd64.zip

cd /usr/bin

terraform -v

1. Configured AWS access key and Secret access key to map terraform with AWS account
2. Created a directory and inside this directory, created .tf files and provided all the required sh commands to provision EC2 and the script installed Jenkins and tomcat on these EC2 Instances – The detailed terraform script files attached below provision the installation of Tomcat and Jenkins on the Instances created

****

**STEP 2 -**

**Apache Tomcat Server Verification**

1. Once installation got completed, A**pache Tomcat installation on tomcat server** was verified on port 8080 on web browser using tomcatup soft link provisioned in the user script

sudo su

ps -ef | grep tomcat

tomcatup

1. After proper installation was ensured, modified configuration on three files as follows

cd /opt

ls -lrt

cd /apache-tomcat-9.0.41

ls -lrt

cd conf

* **server.xml file for port connection**

ls -lrt

vi server.xml (Update the connector port number as 8090 to run tomcat on port 8090)

* **context.xml for Manager link access**

find / -name context.xml

vi /opt/apache-tomcat-9.0.41/webapps/host-manager/META-INF/context.xml (comment the valve class name)

vi /opt/apache-tomcat-9.0.41/webapps/manager/META-INF/context.xml(comment the valve class name)

* **users.xml for role access**

ls -lrt

vi tomcat-users.xml (Below lines highlighted in brown were added in order to use deployer/deployer credentials at Jenkins)

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

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

<user username="deployer" password="deployer" roles="manager-script"/>

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

Tomcatdown

Tomcat up

1. Apache tomcat Server on port 8090 was checked and verified to be running successfully

**STEP 3 -**

**Jenkins Server verification**

a)VerifiedInstallation of Java and Jenkins on provisioned EC2 Instance & Jenkins UI on web browser

b) Configured JAVA\_HOME path and Maven path(M2\_HOME) in .profile file under root directory for Jenkins to interact with Java and Maven which is highlighted in brown

cd /root

ls -la

vi .profile(Add the brown highlighted code to this file and save)

root@ip-172-31-86-2:~# cat .profile

# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

fi

mesg n || true

**JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64**

**M2\_HOME=/opt/apache-maven-3.6.3**

**M2=$M2\_HOME/bin**

**export $PATH**

**Jenkins UI Manage Configure**

1. Deployer/deployer credential were added to credential store for accessing Apache tomcat servlet container
2. Configured JDK installation by adding JAVA\_HOME path and Maven Installation by adding Maven path as noted below

**JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64**

**M2\_HOME=/opt/apache-maven-3.6.3**

1. Below plugins were installed from the available plugin store

**Deploy to container**

**Maven Integration**

**Maven Invoker**

**Configuring Jenkins job**

1. Created a new hello-world maven Project from New Item. Under Source Code Management, Git Configuration details were provisioned with Git hub repository URL and Git Credentials
2. Deploy .ear/.war to container option was chosen as post build action. Apache tomcat Container configuration were provisioned with credential details and Container URL (http://publicIP:8090)
3. This Hello-world maven project pulled the source code from Git hub and deployed (Hello-World) application to the Tomcat apache server and it is accessible by below URL -

[http://publicIP:8090**/**](http://publicIP:8090/)

**Summary:**

I had a Java code repository. This code has been pulled up from Jenkins and it is been built through Maven prior to that I did install few required plugins from Jenkins UI and did some Credentials setup for my Tomcat manager script and created a Jenkins job to perform all these steps and once this got successfully built and deployed through CI/CD process my Java application (Hello-World) is accessible on my Tomcat web-browser.