**In this project I designed an infrastructure, consisting of a VPC and its components and EC2 machines bootstrapped with necessary software installations and configurations, using Terraform to create a CI/CD workflow. In the pipeline, Jenkins will build and deploy a Java application into a Docker container hosted on a different machine and reached via the browser.**

Log in to your AWS account, configure your terminal with your access keys, and apply the Terraform template (can be downloaded from https://github.com/charleseke) to provision the infrastructure.

SSH into the Jenkins server and download Java JDK 8:

yum install java-1.8.0-openjdk-devel

*(Troubleshooting – If*  you want to switch between java versions run*: sudo alternatives --config java)*

Download/install Maven:

wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.tar.gz  
tar xzvf apache-maven-3.8.6-bin.tar.gz

copy the installation to the /opt directory

Edit the .bash\_profile:

Set the JAVA\_HOME path:

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

Set the MAVEN\_HOME path:

MAVEN\_HOME=/opt/apache-maven-3.6.3

M2=/opt/apache-maven-3.6.3/bin

PATH=$PATH:$HOME/bin:$JAVA\_HOME:$MAVEN\_HOME:$M2

Text

Description automatically generated

SSH into the Application server and create a password for the ec2-user who will manage deployments.

sudo su 🡪 passwd ec2-user

sudo su ec2-user 🡪 sudo service docker start

CREATE DOCKERFILE – (which installs Tomcat server and puts build artifact in the correct folder)

sudo vi Dockerfile

Text

Description automatically generated

**Configure Jenkins integration server**

Log on to the Jenkins dashboard by visiting the public ipv4 of the machine listening on port 8080 <ipaddress:8080>

Navigate to Manage Jenkins – System Configuration

In the Manage Plugins install the required plugins:

Publish Over SSH – enable connection between servers to send build artifacts

In the Global Tools Configuration provide the paths for the JDK, Git, and Maven

Graphical user interface, application

Description automatically generated

*Connect the Jenkins server to the Application server by providing the server name, IP address, and username/password of the ec2-user in the Application server.*

**Application

Description automatically generated with low confidence**In Configure System under Publish Over SSH, click Add SSH server

**Create Jenkins job to build and deploy artifact into Tomcat server of the Docker container**

Return to the Jenkins dashboard and create a Freestyle project.

Under the Source Code Management section, click the Git option and paste the GitHub link to integrate the repository containing the web application source code.

Under the Build section, click Add build step—Invoke top-level maven targets

Goals: test install (Maven commands to run so that the war file is created)

*--After building, the .war file will be present in the Target folder of the project.*

*--Create image from from Dockerfile*

*--Run container from image*

In the post build actions, Send build artifacts over SSH:

***Source files*** *webapp/target/webapp.war*

***Remove Prefix*** *webapp/target*

***Remote directory*** .

***Exec command***

sudo docker build -t customimage

sudo docker run -d -p 8080:8080 --name App1-container customimage

\_\_\_ IMAGE \_\_\_\_\_

Access the App through public Ip:port/webapp which is running on Container