############ PROJECT1 ###############

Text

Description automatically generated with medium confidence

Text, letter

Description automatically generated

Project1 – Part1 – High level implementation

Graphical user interface, text, application, chat or text message

Description automatically generated

Below are the detailed steps needed to performGraphical user interface, text, application

Description automatically generated

Created required Vars.tf to create two VMs in Azure

Text

Description automatically generatedText

Description automatically generated

Created main.tf to create two Linux VMsText

Description automatically generated

Used Terraform Provisioner to install JDK and Jenkins in VM1

Text

Description automatically generated

Use Terraform Provisioner to install JDK, Maven, Ansible, Docker, AzureCli and Git

Text

Description automatically generated

Init, Plan and Apply Terraform Script to create VM’S in Azure

I already created VMS using terraform before. currently both VM'S are up and running in Azure.

Graphical user interface, text, application

Description automatically generated

A picture containing text, screenshot, computer, computer

Description automatically generated

Manually Started Jenkins and configured required Plug-ins and Master Slave Configuration

Manually Started Jenkins

Graphical user interface, text

Description automatically generated

**Able to Login to Jenkins**

Graphical user interface, application

Description automatically generated

Completed Master Slave Configuration

Graphical user interface

Description automatically generated

Configured required Plug-ins

A screenshot of a computer

Description automatically generated with medium confidence

Project1 – Part2 – High level implementation

Text

Description automatically generated

**Below are Detailed steps needed to perform in Part-2**

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, website

Description automatically generated

Graphical user interface, text, application, website

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated



**Created Maven Project with Archetype as web application in eclipse**

Graphical user interface, text, application, email

Description automatically generated

**Modify Index.jsp under src/main/web content to display a custom message**

Graphical user interface, text, application

Description automatically generated

**Generated Docker file under project folder of app and Modified FROM statement to use tomcat as base image**Graphical user interface, text, application

Description automatically generated

**Created a GitHub repository and copied repo URL, In Eclipse converted the app into a local repo from Team menu share Project Option and performed Commit and Pushed the code to the remote repo.**

A screenshot of a computer

Description automatically generated

**In build server configured Ansible manually**

A picture containing text, indoor, screenshot, close

Description automatically generated

**Created a playbook1(final1.yaml) to create a VM in azure**

Text

Description automatically generated

Text

Description automatically generated

**Created the playbook2(Docker2.yaml) to install Docker engine on the VM**

Text

Description automatically generated

**Created a shell script(ip.sh) to get VM IP and updating it in Inventory File**

A screenshot of a computer

Description automatically generated with medium confidence

**Push fianl1.yaml, docket2.yaml2 and shell script to Remote Git repo created in Phase3**

A screenshot of a computer

Description automatically generated with medium confidence

**From Build server able to run all 2 playbooks and one sh script successfully.**

A picture containing text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Project1 – Pipelines - High level implementation

Text

Description automatically generated

Below are the detailed steps needed to perform

Graphical user interface, text, application, email

Description automatically generated

Created Repo in hub.docker.com

Graphical user interface, text, application, email

Description automatically generated

Configured Global tool configurations in Jenkins to use JDK,Maven and Git

Graphical user interface, text, application

Description automatically generated

**Configured Git credentials in Jenkins Vault**

Graphical user interface

Description automatically generated

PIPELINE1: **Created Pipeline1 using Freestyle project in Jenkins**

Graphical user interface, text, application

Description automatically generated

PIPELINE1: **In Jenkins SCM stage Pulled code form Remote Repo**

Graphical user interface, text, application, email

Description automatically generated

PIPELINE1: **In Build Stage, Step1: used maven top level target to build**

Graphical user interface, text, application, email

Description automatically generated

PIPELINE1:**in Build Stage Step 2: Used Docker build to create an image to push that image to the docker hub.Graphical user interface, text, application, email

Description automatically generated**

PIPELINE1: **Once pipeline1 configuration is complete and started build.**

Graphical user interface, text, application

Description automatically generated

PIPELINE1: **It started building the image**

Text

Description automatically generated

PIPELINE1: **after image creation, it’s pushing the image to Docker central HUB**

A picture containing text

Description automatically generated

PIPELINE1: **Finally Build is successful**

Graphical user interface, application

Description automatically generated

**Created Pipeline2 using Freestyle project in Jenkins**

Graphical user interface, text, application, email

Description automatically generated

PIPELINE2: In SCM stage pulled code form Remote Repo

Graphical user interface, text, application, email

Description automatically generated

PIPELINE2**: In Build Stage Step 1: Selected ansible Playbook1(Final1.yaml)**

Graphical user interface, text, application, email

Description automatically generated

PIPELINE2**: In Build Stage Step 2: Selected Shell Script(IP.sh)**

Graphical user interface, text, application

Description automatically generated

PIPELINE2: In Build Stage Step 3: Selected Ansible Playbook2(Docke2.yaml)

Graphical user interface, text, application, email

Description automatically generated

PIPELINE2: In Build Stage Step 4: Selected Ansible Playbook3(Docker3.yaml)

Graphical user interface, text, application, email

Description automatically generated

PIPELINE2: Pipeline2 was configured successfully and started the Build.Graphical user interface, text

Description automatically generated

PIPELINE2:Pipeline2 Creating VM on Azure using Playbook1**(Final1.yaml)**

Table

Description automatically generated

PIPELINE2: Pipeline created VM using ansible playbook1 and using shell script its updated ansible hosts file automatically, also using playbook2 its installing docker on newly created VM in Azure.

Text

Description automatically generated with low confidence

PIPELINE2: Finally Build is successful, Pipeline2 is able to successfully deploy the application by taking the image from Docker Hub.

Text

Description automatically generated

PIPELINE2 SUCCESS: After Build success, I got this IP 13.68.130.175

and when I try to browse the IP, I am getting the below message, which means my

Pipeline is successful.

Text

Description automatically generated

**Below is the code on my local, it means my Project is a success**

Graphical user interface, text, application

Description automatically generated

Final step: Configured the webhook on the GitHub repository, just to ensure that every time a developer commits a code to GitHub, our build will be triggered.

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Additional step: Also Enabled Build after other projects are built option in pipeline2.

so, once pipeline1 is successful and the second pipeline will get triggered.

Graphical user interface, text, application, email

Description automatically generated

FINAL STEP – TESTING CI/CD

Now I am going to update the code on a local system and push code to GitHub. Once I push the code, Pipeline1 will be triggered automatically, and after that Pipeline2 will be executed automatically.

Below are the pipeline tasks.

**Pipeline1** - it will take app code from GitHub, build a docker image and will push to docker hub, and it will trigger pipeline2.

**Pipeline2** - it will deploy VM on Azure and install docker on that VM. Also, using Ansible playbooks, it will get an image from the docker hub and will deploy on that Azure VM using the playbook.

This is a continuous integration and deployment.

Updated code on my local

Graphical user interface, text

Description automatically generated

Pushed the code

Text

Description automatically generated

**Pipeline1** - it will take app code from GitHub, build a docker image and will push to docker hub, and it will trigger pipeline2.

Graphical user interface, text, application

Description automatically generated

**Pipeline2** - it will deploy VM on Azure and install docker on that VM. Also, using Ansible playbooks, it will get an image from the docker hub and will deploy on that Azure VM using the playbook.

Text

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

This is a continuous integration and deployment.

############### THE END ##################