***Steps to execute tasks***

***Git path for all task:*** [***https://github.com/angelin-mariya/course7\_project.git***](https://github.com/angelin-mariya/course7_project.git)

Task1

Subtask 1

Aws cli was installed and configured (AWS configure command) in Linux machine.

S3 bucket was initialized to store the tfsate file of terraform.

Subtask 2

AWS VPC, 1 IGW 1 NAT-GW in AZ-a, 2 public Subnets, 2 privates (1 each in AZ-a &b), Route Tables for both subnets was created using Terraform code.

CIDR Block/16 for vpc and CIDR Block/24 for subnets

Subtask 3

Security group was created for -

• Bastion host SG - Allow self ip to ssh to ‘bastion’ instance and allow all egress.

• Private Instances SG - Allow all incoming traffic from within VPC and all egress.

• Public Web SG - Allow incoming to port 80 from self IP and all egress.

To access self ip sh has to be exected .

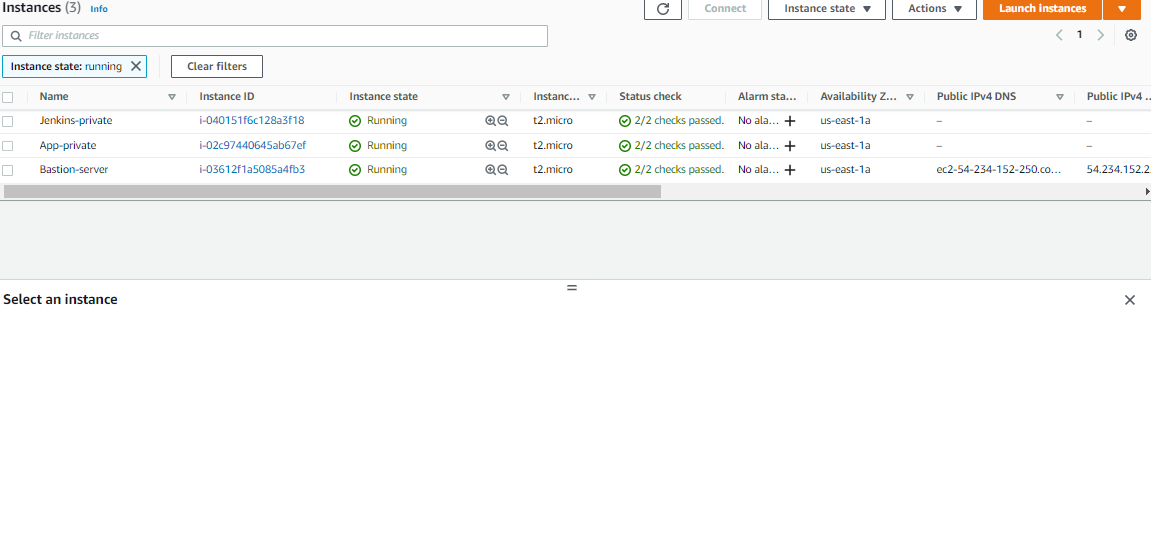
For that

sudo apt-get install jq

chmod 764 whatismyip.sh (Gives user full access, group read and write access, and other read access)

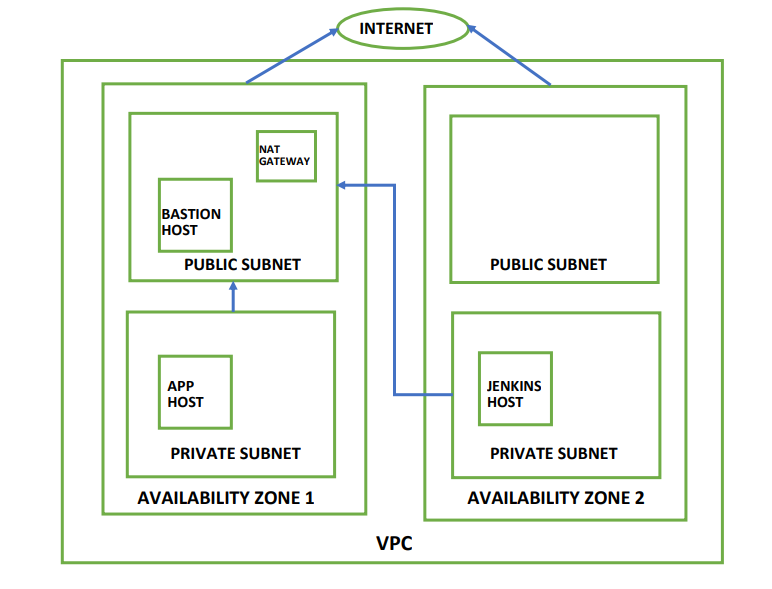
Subtask 4

EC2 instances - bastion (Bastion host SG), (Jenkins Private Instances SG), app (Private Instances SG) was created with Ubuntu-20 official AMI using terraform.



Subtask Bonus

VPC Diagram



TASK-2

Subtask 1

log on to bastion server

1. Installed Ansible using

$ sudo apt update

$ sudo apt install software-properties-common -y

$ sudo add-apt-repository --yes --update ppa:ansible/ansible

$ sudo apt install ansible -y

2. check connectivity to Jenkins and App hosts

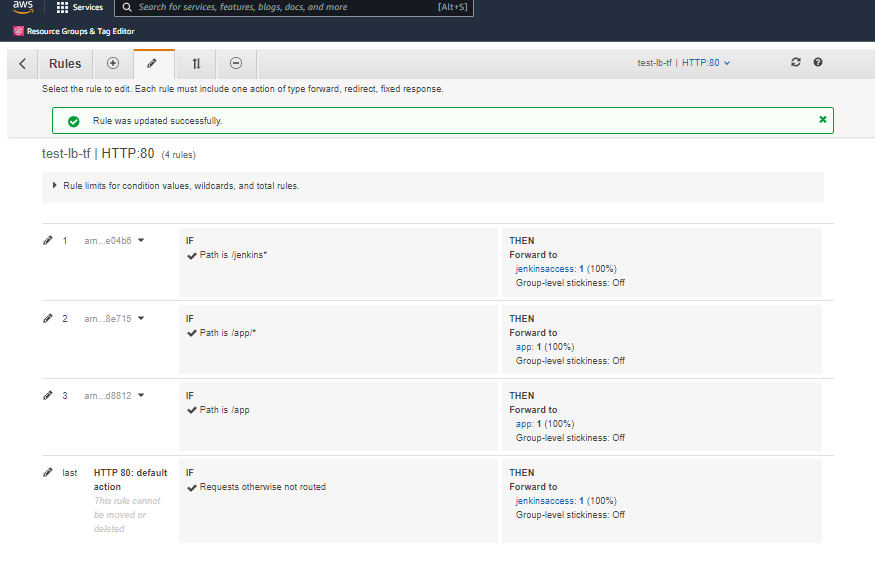
ansible all -m ping -i inventory

3.Install docker using Ansible script

ansible-playbook -i inventory install\_docker.yml

Subtask 2

ALB was created using terraform and forwards /jenkins, /jenkins/\* to a Target Group having Jenkins host 🡪 port 8080 and /app, /app/\* to a Target Group having app host 🡪 port 8080



Subtask 3

1. Install Jenkins (prereq – java)

$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

$ sudo apt upgrade -y

$ sudo apt update

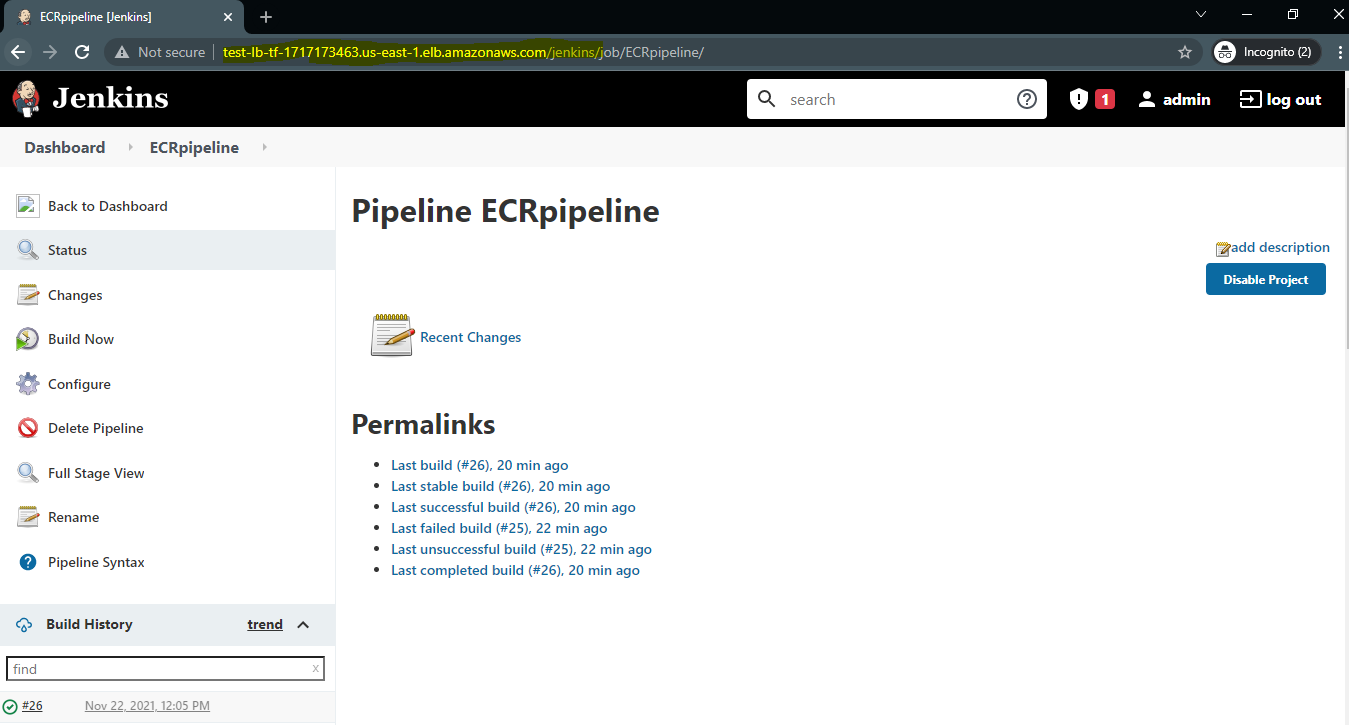
$ sudo apt-get install default-jdk -y

$ sudo apt-get install default-jre -y

$sudo apt-get install jenkins -y

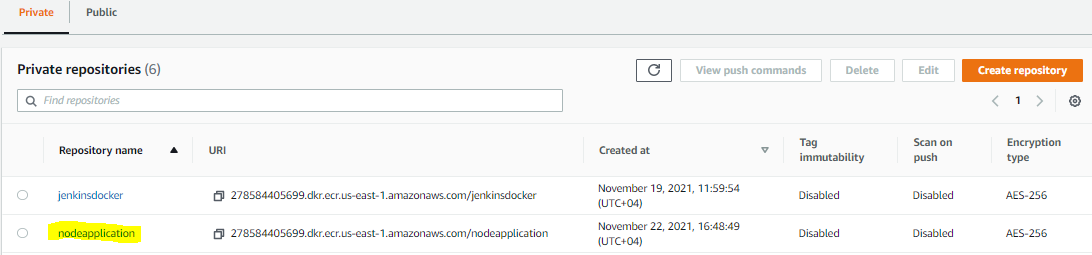
Jenkins was accessible over alb as shown below

Installed recommend plugins and also SSH agent , aws

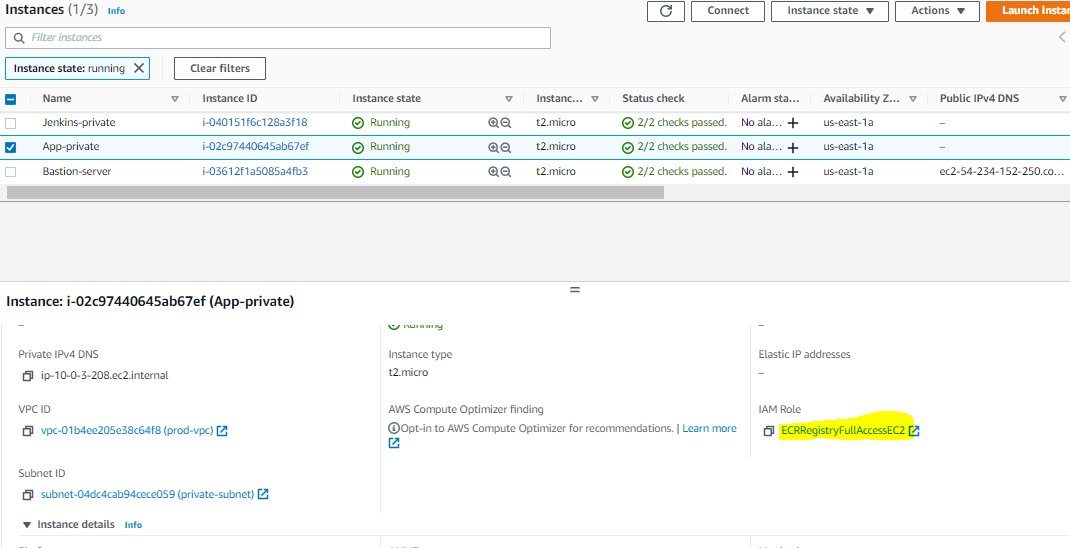


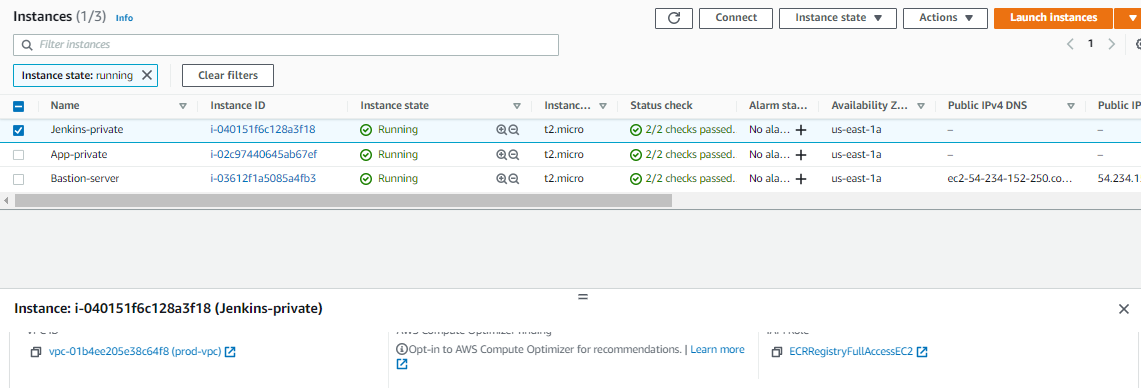
Subtask 4

Created an ECR repository to store the docker image of the node application.



Created attached IAM role to provide ECR access to ‘jenkins’ and ‘app’ hosts





‘jenkins’ and ‘app’ hosts are authenticated to use the ECR repository

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 278584405699.dkr.ecr.us-east-1.amazonaws.com

TASK 3

Subtask 1

Docker file was written for the node application and was pushed in to github.

FROM node:14.14.0-alpine

WORKDIR "/app"

COPY ./package.json ./

RUN npm install

COPY ./ ./

EXPOSE 8081

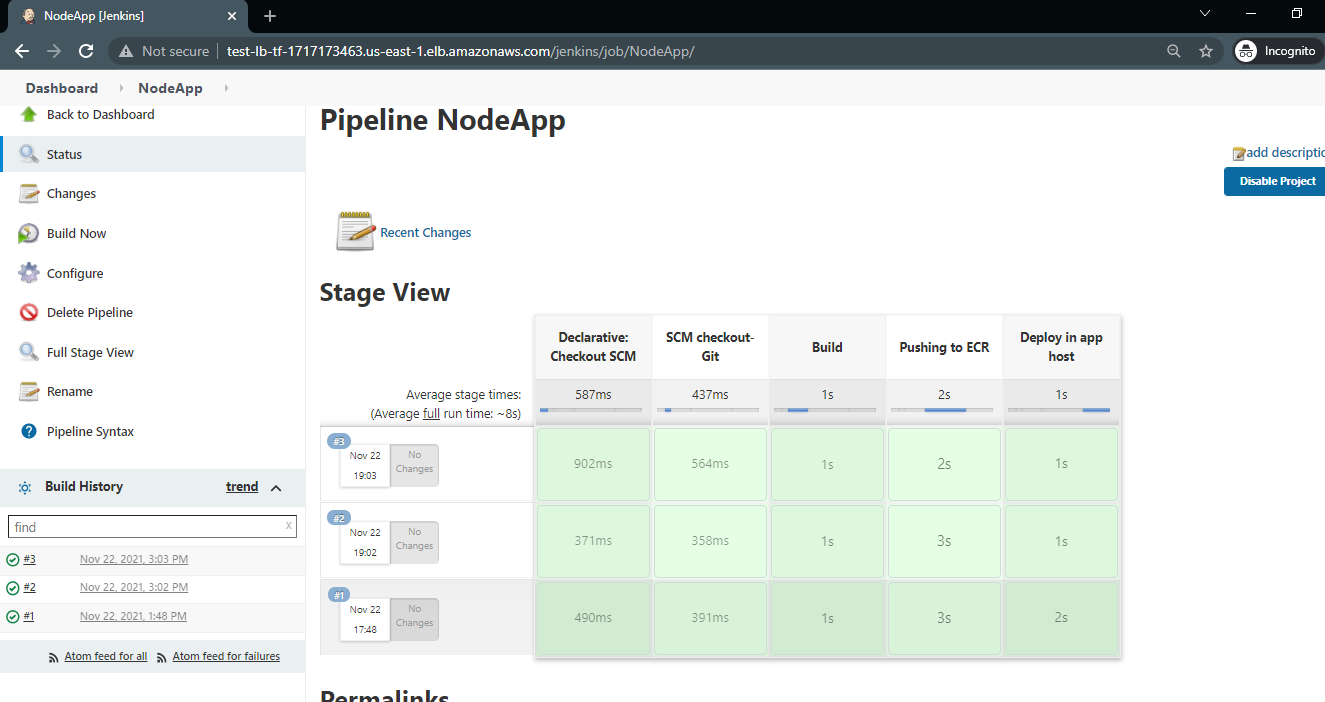
CMD ["npm", "start"]

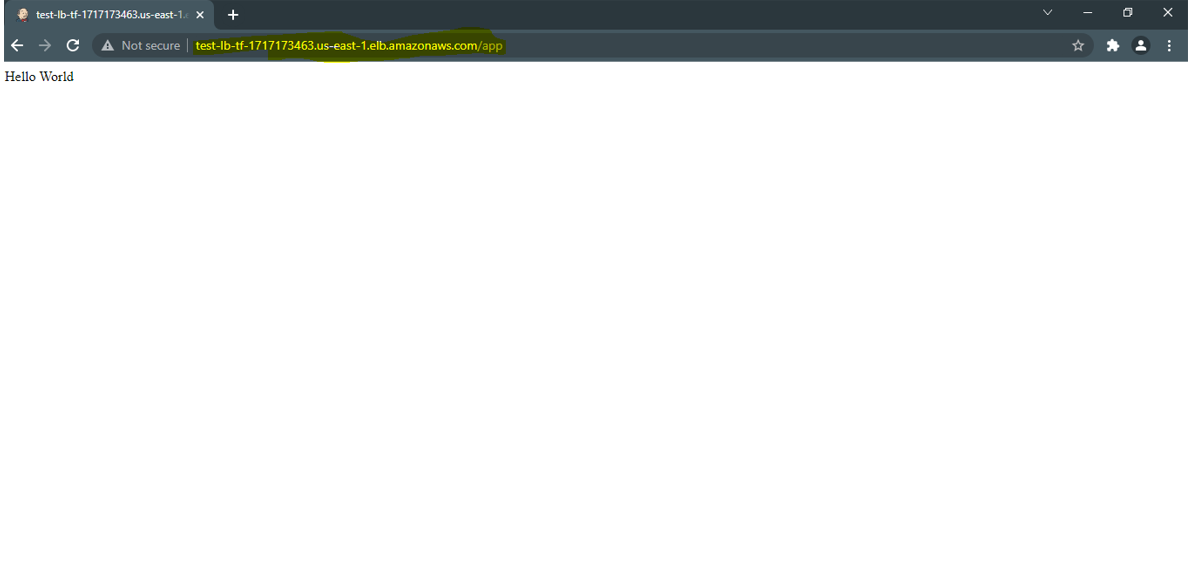
Subtask 2

Jenkins file was written to check out node application and build docker image and push it to ECR.

Subtask 3

Another stage was added to the Jenkins file to pull and run the docker image from ECR repository the node application to app host and then to check the image and run it .





Jenkins Job Console( pls zoom the word size to 250 %)

