Documentation for Devops project course-4

Problem statement:

Implementing CICD pipeline for a node.js application by using Terraform for infrastructure provisioning and Ansible for configuration management. Node.js app has to be dockerized and should be deployed on to the target machines. The app should be accessible from a custom path using an AWS Application Load Balancer.

Technologies/Tools used: Aws, Docker, Terraform, Ansible, Jenkins

Task1 :

\*Installed Terraform and AWS CLI on the workstation and configured AWS credentials to my local workstation, Initialized a bucket in s3 for the backend state store, and used AWS Dynamodb table for remote state-lock.

\* Used terraform AWS resources to create the following components

AWS VPC

1 IGW

1 NAT-GW in AZ-a,

Subnets (2 public & 2 private, 1 each in AZ-a &b),

Route Tables for both subnets

\*Created security groups and 3, t2.micro instances using terraform resources and tagged them as Bastion\_host, Jenkins\_server and App\_host respectively.

\* Used Data Source for integrating public API [http://ipv4.icanhazip.com](http://ipv4.icanhazip.com/) to fetch my workstation machine IP address.

\* Terraform commands used: teraform init, terrafrom plan,terraform apply, terraform refresh,terraform destroy.

\* Configured a proxy-jump from git-bash ssh CLI using the below configuration. Here I created a public-private ssh keypair and copied the public key to both the bastion host and Jenkins server's authorized keys.

.ssh/config

Host bastion

User root

Hostname 59.161.64.15

Host jenkinsserver

User root

Hostname 10.0.14.141

ProxyJump bastion

Task2:

\* Installed ansible on Bastion\_host and created docker-playbook.yaml and jenkins-playbook.yaml file

\* Tweaked /etc/ansible/ansible.cfg and bypassed the host key checking by below configuration

[defaults]

host\_key\_checking = False

\* Created custom inventory file with below configuration

[jenkinsserver]

10.0.14.141 ansible\_ssh\_private\_key\_file='upgrad\_project.pem' ansible\_ssh\_user=ubuntu

[app]

10.0.3.125 ansible\_ssh\_private\_key\_file='upgrad\_project.pem' ansible\_ssh\_user=ubuntu

\* Installed docker and Jenkins on target machines using the below command ansible-playbook <playbook name> -i inventory.

\* Created ALB and target groups manually and configured custom path rules to access /jenkins and /app.

\* Logged in to Jenkins server and added jenkins user to docker group and added docker socket level permissions by using the below commands

- sudo usermod -aG docker jenkins

- sudo chown jenkins /var/run/docker.sock

- sudo chmod 644 /var/run/docker.sock

\* Created ECR repository with name upgrad\_project\_555.

\* Installed AWS cli on both Jenkins\_server and App\_host

\* Tweaked /etc/default/jenkins file for custom path access and added --prefix tag

- JENKINS\_ARGS="--webroot=/var/cache/$NAME/war --prefix=/jenkins --httpPort=$HTTP\_PORT"

\* Attached a IAM role to both the Jenkins server and app server with ECR publicfullaccess policy.

\* Created public-private ssh keypair for jenkins user and added the public key to app server's authorised\_keys and checked ssh connectivity.

Task3:

\* Created private Github repository by name upgrad\_final\_project\_555

\* Created Dockerfile.

\* Dockerised and tested node.js app locally.

\* Created Jenkinsfile

\* Added ssh, docker and git related plugins in Jenkins from Manage Jenkins sections

\* Configured Jenkins with two credentials i,e Personal Access Tokens and sshagent credentials.

\* Did a git-push of all relevant files and node-app code base to private GitHub repo named upgrad\_final\_project\_555.

\*Configured a jenkins pipline job and executed CICD pipeline using Jenkinsfile.

NOTE: Visibility of upgrad\_final\_project\_555, private GitHub repo has been changed to the public, for upGrad evaluation purposes.