The project name is: Deploy end to end CI/CD Infrastructure using terraform, AWS EKS, Jenkins and Kubernetes deployment.

In this project we will be creating Infrastructure to deploy Jenkin server and then use Jenkin Pipeline and using Terraform create EKS Cluster and deploy a nginx server inside a Kubernetes deployment pod and create a Load balancer for high availability.

1. We will create a Jenkin server using code in terraform and deploy the Jenkin server in EC2 instance and install git, kubernetes and terraform in the same.
2. To connect to our EC2 instance we use keypair authentication to connect and access. We will need this because after deploying Jenkin Server we need admin password to access, and you will be only able to access if you have key pair to connect to your Ec2 instance.
3. We will save out terraform.tfstate file in remote location i.e. S3 bucket.
4. Once you login Jenkin server we will now connect AWS user and Git account to fetch automatically the tf files for creation of EKS cluster and then deploy high availability Nginx server inside a POD.
5. Now we will just build using Build now in Jenkins for our resources to be build.
6. You need to use two repository Shubham-Devops and Jenkins-deploy-eks

Let’s Go deep in what is Shubham Devops Repository.

1. In this Repository you will find files to create Jenkins Server.
2. Backend.tf: This is used to initialize the backend S3 bucket and region to be deployed.

A screenshot of a computer

Description automatically generated with medium confidence

1. Jenkins-server-scripts.sh: This file is shell bash script file , which will specify what will get installed and how inside an EC2 instance.
2. Outputs.tf: This file is just used to print the public IP address of our Jenkin Server.
3. Provider.tf: This file contains necessary information for the tool to connect to AWS env. Like which cloud provider, region, version etc.
4. Terraform.tfvars: This file defines the variables value used in the terraform code.
5. Variables.tf : same as above.
6. Jenkin-server.tf: This file is used to create Ec2 instance.

* To create first we will mention which aws ami image to be used, version, owner, name
* Then make a resource to create AWS instance and provide the resource configuration block and specify various attributes and values that are specific to the particular resource type.
* We will also provide user data in this which nothing but the shell script to run.

A screenshot of a computer

Description automatically generated with medium confidence

1. Vpc.tf: This file is used to create a VPC network, subnets, internet gateway, route table, security group to give access to ports to send and receive traffic. In short, the networking capability.
2. After creating this you will be able to access Jenkins.

A screenshot of a computer

Description automatically generated with medium confidence

1. Now find the Jenkins password by accessing the EC2 using Key pair or AWS CLI to the given location above in the picture.

A screenshot of a computer

Description automatically generated with medium confidence

1. Once your Jenkins is ready you will see this message

A screenshot of a computer

Description automatically generated

Now let’s See another Repo which we will be using for the next part that is CI/CD “Jenkins-deploy-eks”

In this repo we have

1. Kubernetes Folder:
2. Nginx deployment.yaml file: This this deployment manifest yaml file to deploy nginx.
3. Nginx.service.yaml : This file is used to create a service and to define and configure a service in a Kubernetes cluster. The Service will then act as a stable endpoint to route traffic to the pods with the label app: nginx, typically used to expose and load balance traffic to a set of nginx pods.
4. Terrafom Folder:
5. Backend.tf: As mentioned above
6. Eks-cluster.tf: In this we have used EKS module of terraform to create eks cluster. We will specify the eks cluster name, vpc config which will tell the subnet id to be used, node groups config.
7. Terraform.tfvars: variables value to be used
8. Variables.tf:
9. Vpc.tf: In this we will VPC network for the EKS cluster. We will create as mentioned in guideline to create a VPC for high availability zone. We will create public, private subnet,nat gateway, internet gateway, network addressing CICDR blockand Subnets for Availability zones.
10. Jenkinsfile: In this we write the pipeline what Jenkins needs to follow. In this we have defined 3 stages; first Terraform to initialize then EKS cluster to create and in last stage to create Pod in Cluster. Now this all needs to be done in AWS so at start we mention the environment and AWS access key, secret key and default region.
11. The next Step is to set Up Git Credentials and AWS Credentials in the Jenkins to access your Git and AWS resources.

A screenshot of a computer

Description automatically generated with medium confidence

1. Then we select the pipeline and use pipeline Script SCM and provide Git repository URL and use login and password mentioned in above. Here he URL we will use is For another Repository which is “Jenkins-deploy-eks”.

A screenshot of a computer

Description automatically generated with low confidence

1. Now Build: It takes 15-20 mins to build

A screenshot of a computer

Description automatically generated

1. EKS Cluster Created:

A screenshot of a computer

Description automatically generated

1. Availability Zones:

A screenshot of a computer

Description automatically generated

1. Load Balancer and DNS name to access the NGINX server

A screenshot of a computer

Description automatically generated with medium confidence

1. Here we go !!!

A screenshot of a computer

Description automatically generated with medium confidence