Name: Venkata Lahari Goruputi

Role: DevOps Intern

Topic: Jenkins pipeline for automating Terraform deployments

✓ What is Terraform:

- Terraform is an open-source infrastructure as code (IaC) tool created by Hashicorp that enables you to define, provision, and manage cloud and on-premises infrastructure through code. It allows for automated infrastructure management by defining your desired state in configuration files.

✓ What is Jenkins:

- Jenkins is an open-source automation server widely used for continuous integration (CI) and continuous delivery (CD). It helps automate various stages of the software development lifecycle, from building and testing code to deploying applications. Jenkins integrates with a wide range of tools and technologies, making it a flexible and powerful solution for automating software development processes.

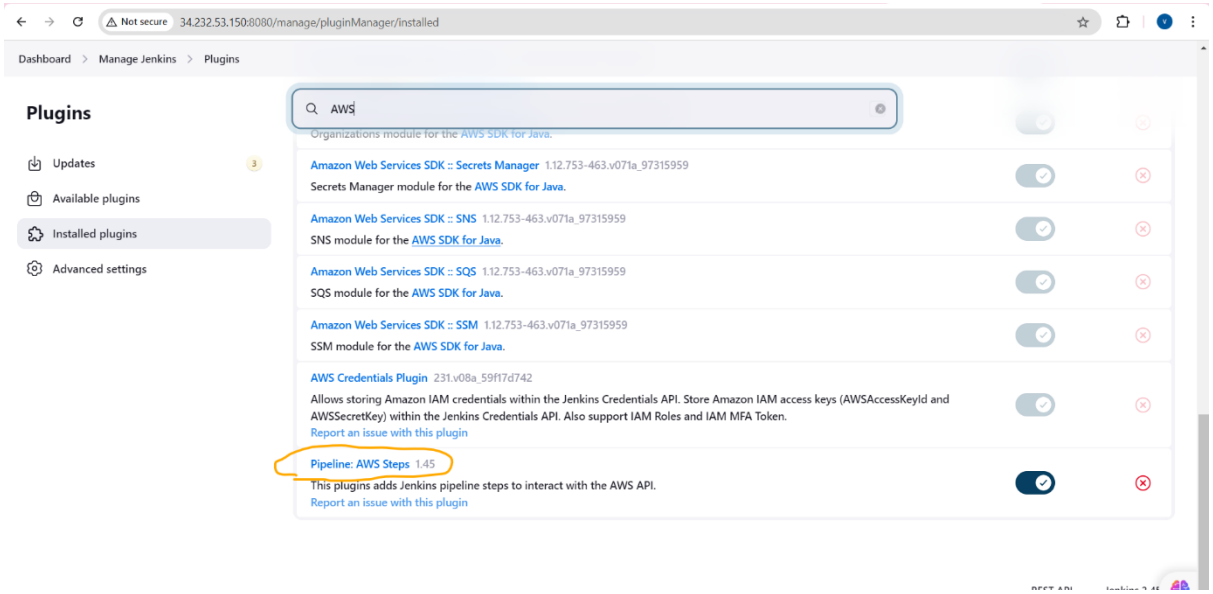
- The task to create infra from Terraform using terraform S3 backend with DynamoDB table from Jenkins.
- As part of this task, I have written terraform script which creates EC2 instance.
- To run this script, we need to install terraform and then we need to configure our AWS credentials, so that terraform will know about our AWS credentials and it will create the infra in the matching credentials.
- When we run **terraform init** command terraform will download provider details and required Apis which requires to create infra in AWS.
- When we run **terraform apply** it will create the infra and also will create a statefile which has all the details about our infra structure details which is **terraform.tfstate.**
- We can store this statefile in a common location which is called as **backend,** here we are using **S3** backend, and we can lock this statefile using DynamoDB table, this process is called statefile locking.
- After this I have setup Jenkins Master-Slave architecture in order make this terraform process automate.
- As part of Master-Slave setup I have created two EC2 instances and Installed Jenkins in one node which becomes Jenkins Master and then connected the other instance to Jenkins which becomes node (slave).
- After successfully connecting node to master I have installed **Terraform** in node, because to run the terraform commands we need to have terraform in the node where we want to execute the terraform script.

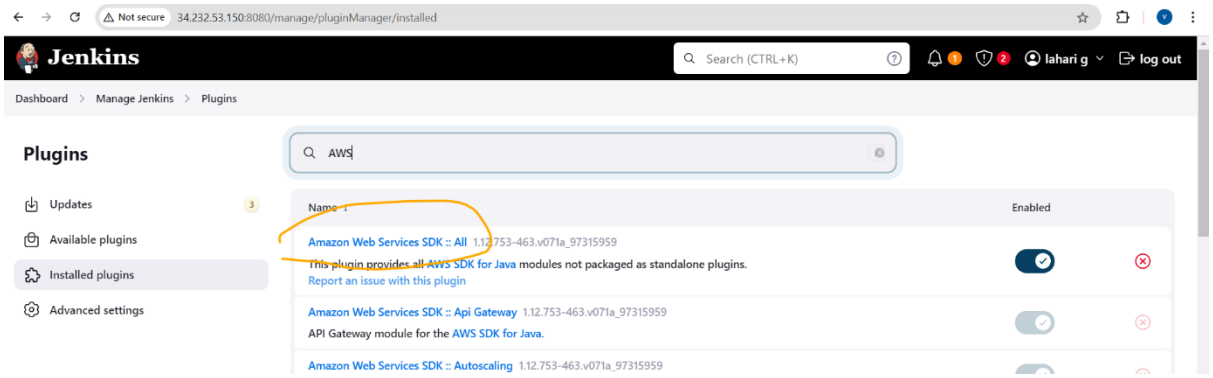
- Then I have written the Jenkins pipeline which has the script to run the terraform script in node.

```
pipeline {
  agent {
    label 'Lahari'
  }
  triggers {
    pollSCM('* * * * *')
  }
  environment {
    AWS_DEFAULT_REGION = 'us-east-1'
  }
  stages {
    stage('AWS Credentials') {
      steps {
        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding', credentialsId: 'aws-
credentials']]) {
          script {
            sh 'aws s3 ls'
          }
        }
      }
    }
    stage('terraform-execution') {
      steps {
        script {
          sh '''
            cd terraform-jenkins/
            terraform init
            terraform fmt
            terraform validate
            terraform plan
            terraform apply -auto-approve
            terraform destroy -auto-approve
          '''
        }
      }
    }
  }
}
```

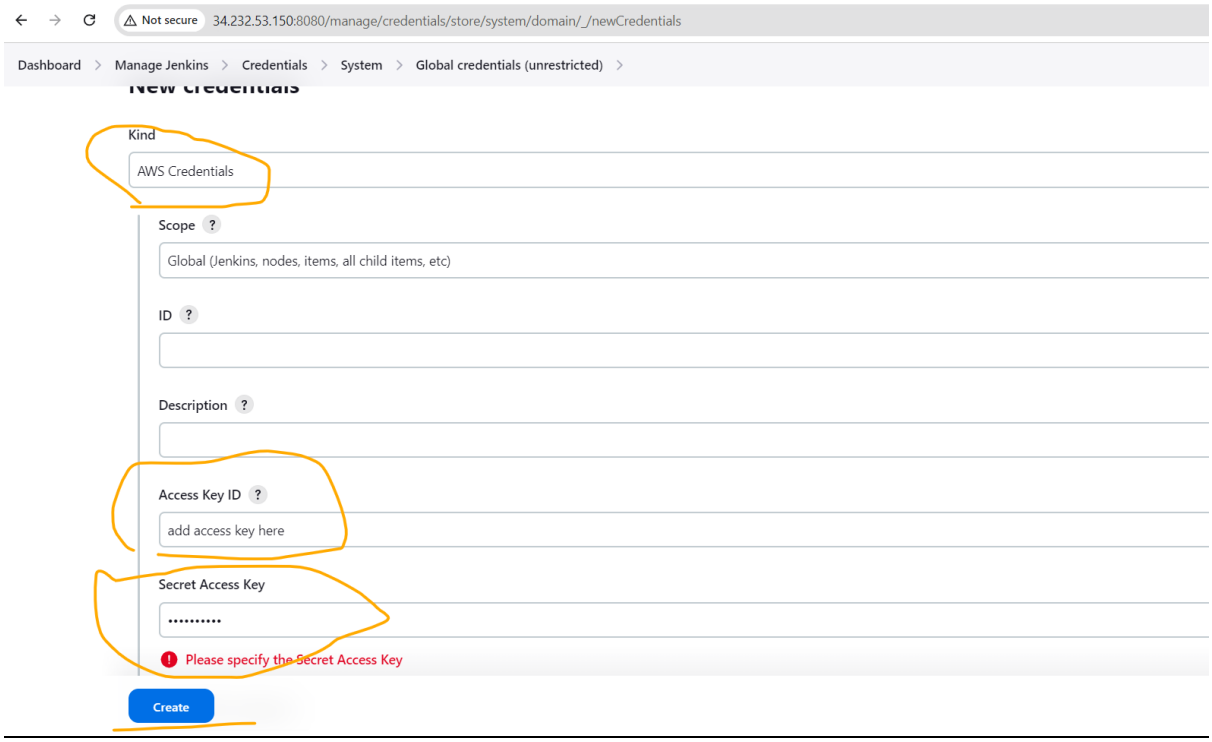
- Above are the pipeline I have used to run the terraform script.
- Then after writing the pipeline script push the terraform script and Jenkinsfile to remote repository to run the pipeline
- Create a pipeline project in Jenkins and do the necessary configuration.

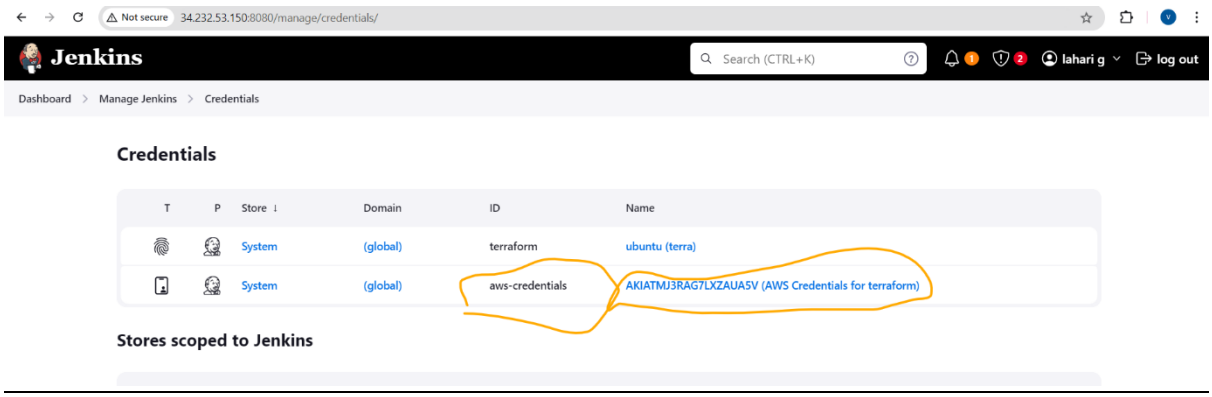
- In this we need to add our AWS user credentials in jenkins credentials to let Jenkins and terraform know where to create infra.
- To store these credentials, we need to install the below plugins in Jenkins.
    - **AWS Steps plugin**
    - **Amazon Web Services SDK plugin**



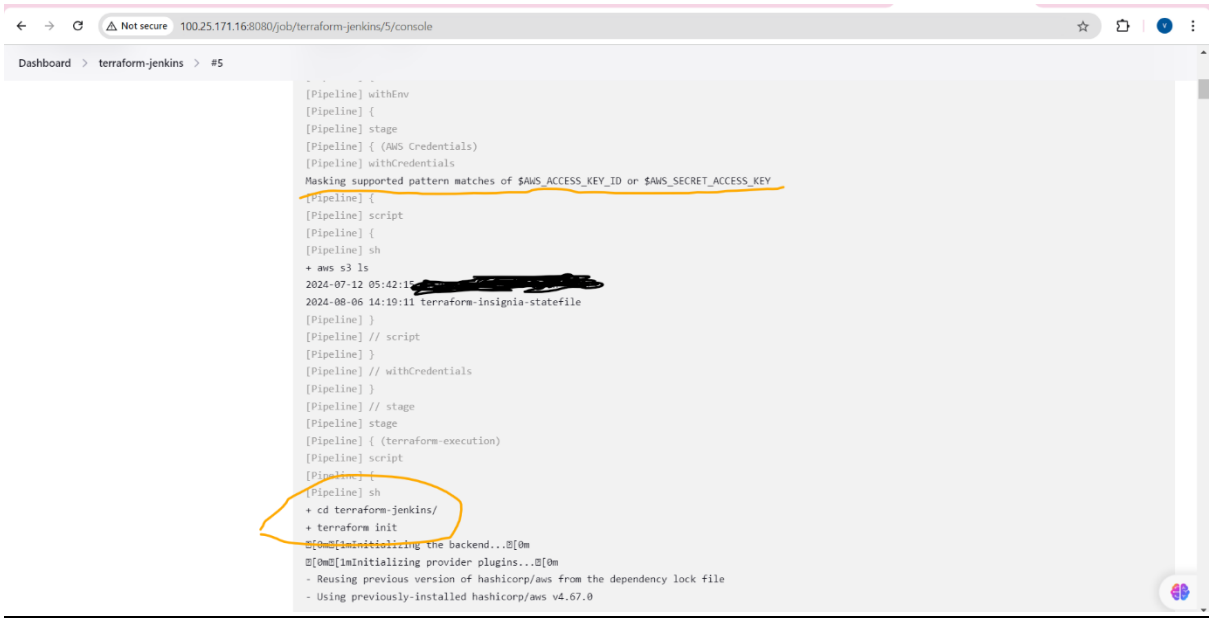


- After installing the plugins, we need to restart the Jenkins, then we can see an additional credentials storing option in Jenkins credentials as below.
- Then in this we need to give a name which is credentialsID in the credentials and add the Access Key and SecretAccessKey of our AWS credentials, then save the configuration.
    - **<u>Note:</u>**
        - We need to use the same credentialsID which have used to add the AWS credentials in the Jenkinsfile also.

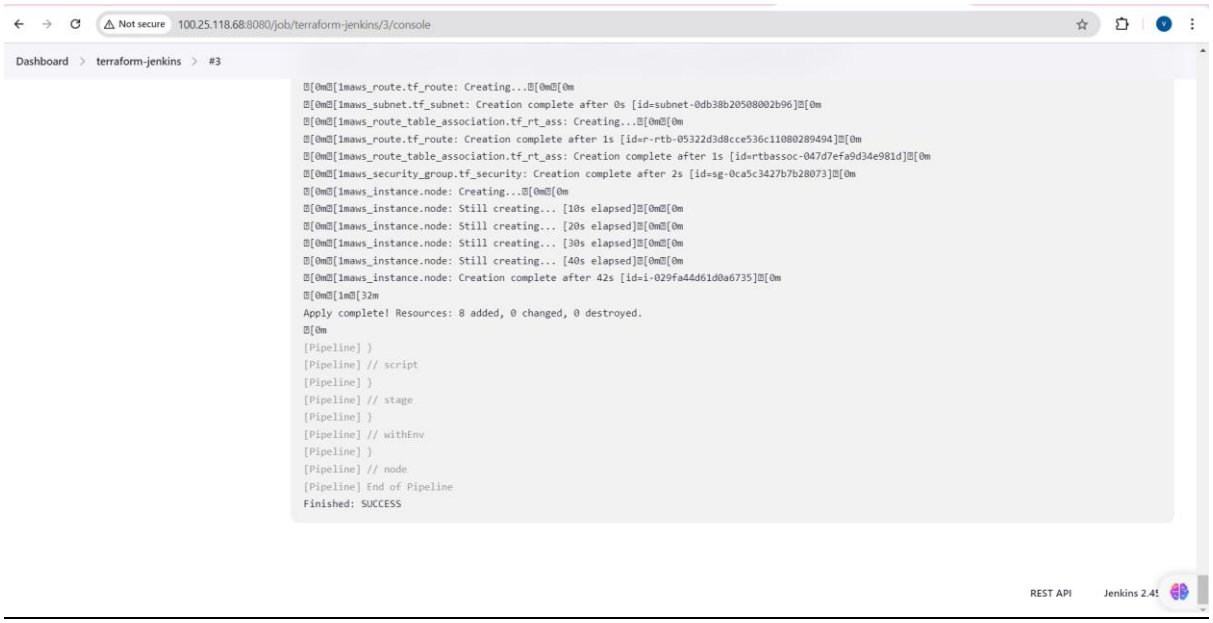




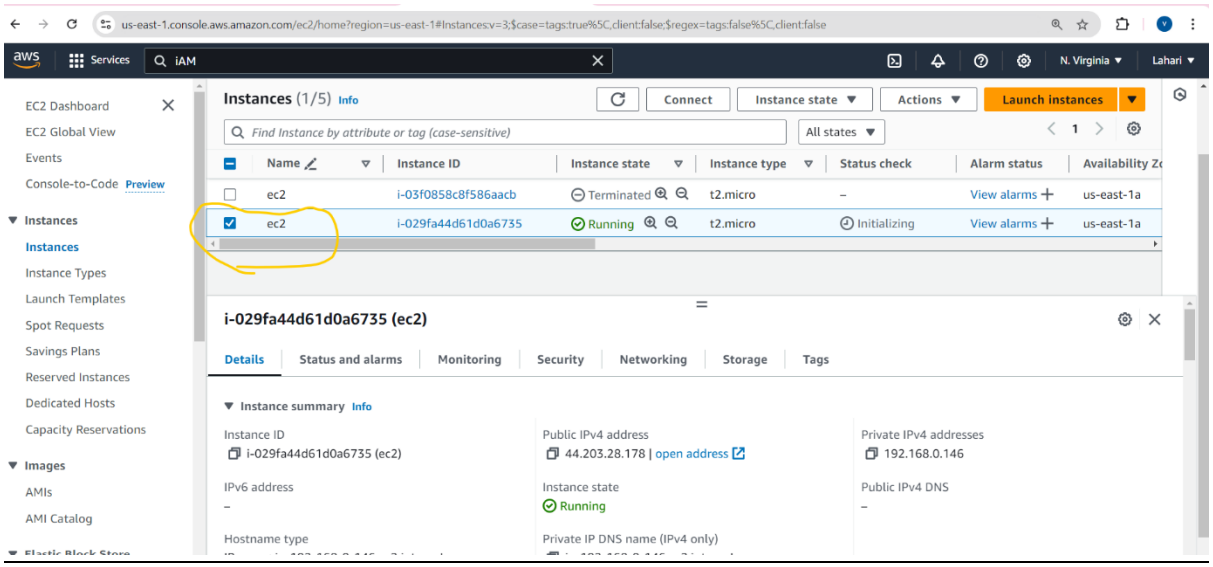
- Then After adding the credentials, run the pipeline, if all the configuration is correct then the pipeline will execute the terraform script from Jenkins Node and creates the infra in AWS and stores the statefile in S3 bucket and locks the statefile with DynamoDB.
- Below Images shows that the pipeline is picking up AWS credentials from Jenkins credentials.



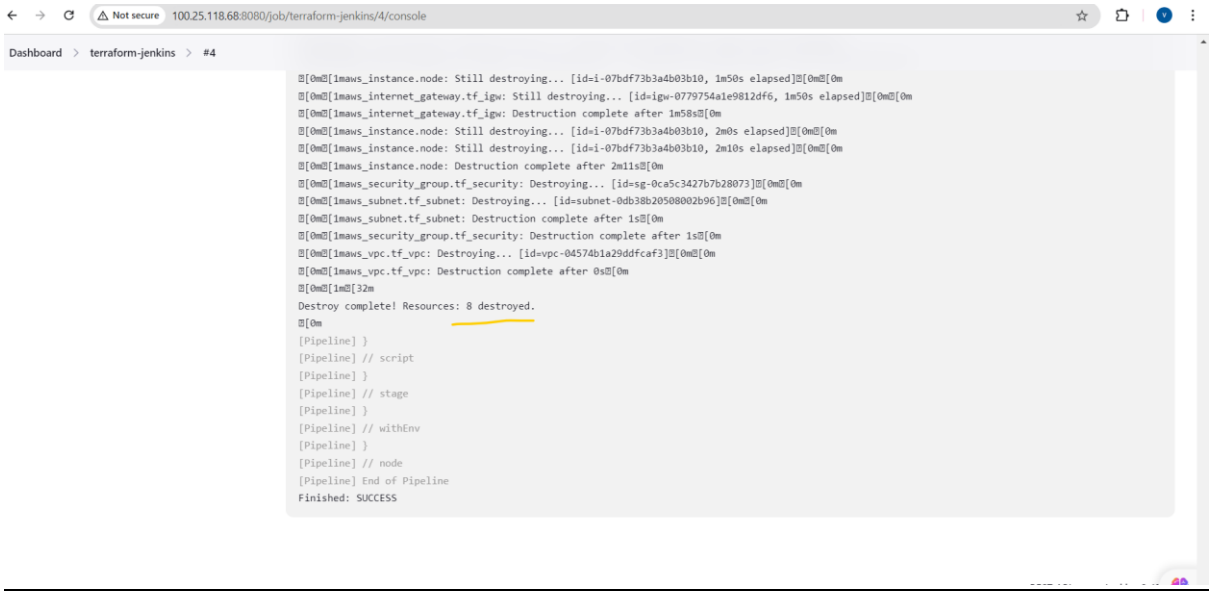
- Below image shows the creation of infra from pipeline is success.



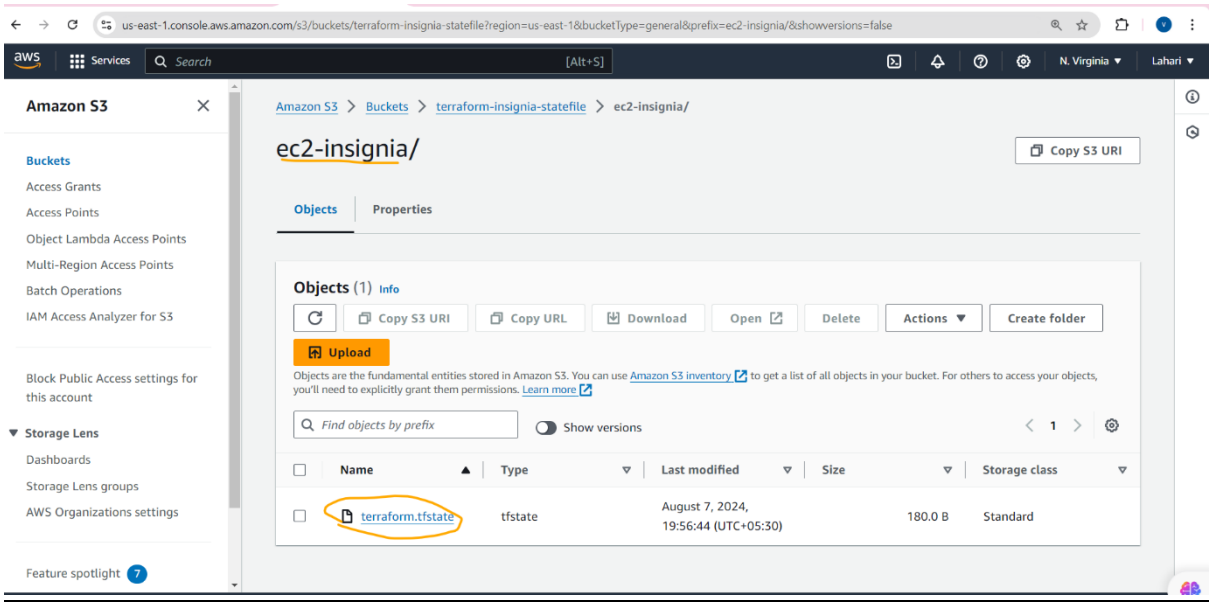
- Below image indicates the creation of EC2 instance.



- Below image indicates that the resources were destroyed after the creation from pipeline.



- The below pic indicates that the statefile were stored in S3 bucket with the given key.



- This is the process that I have followed to automate the terraform script from Jenkins.

Thanks,

Lahari G