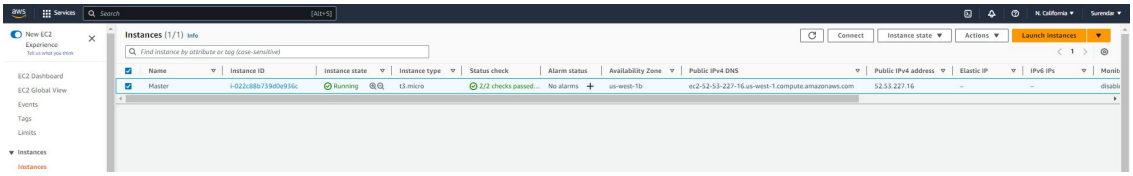
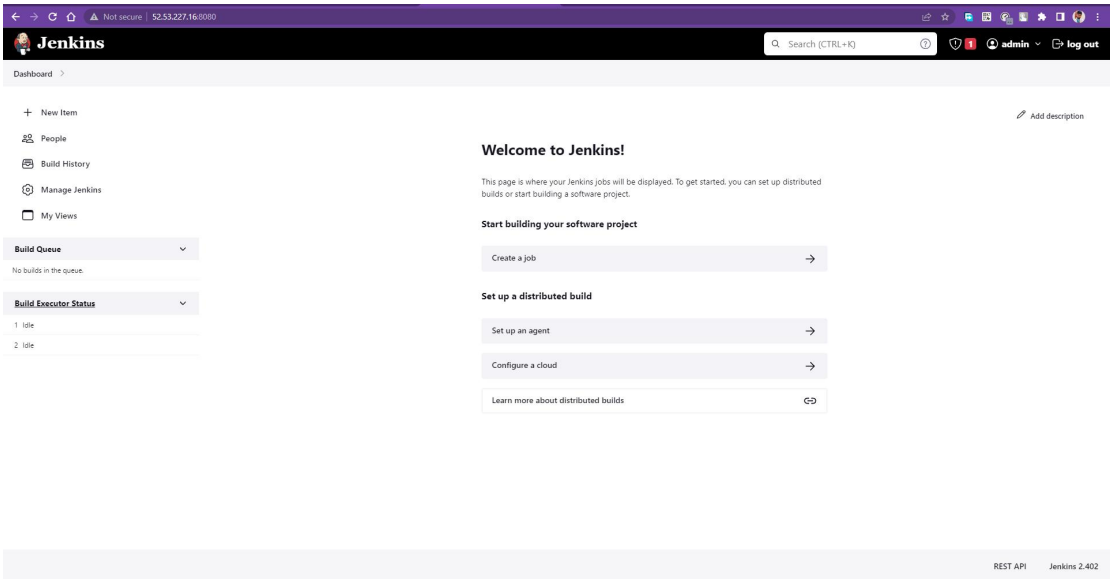
DevOps Project Task
Steps to Create VPC resources in AWS Infra with Terraform and Jenkins

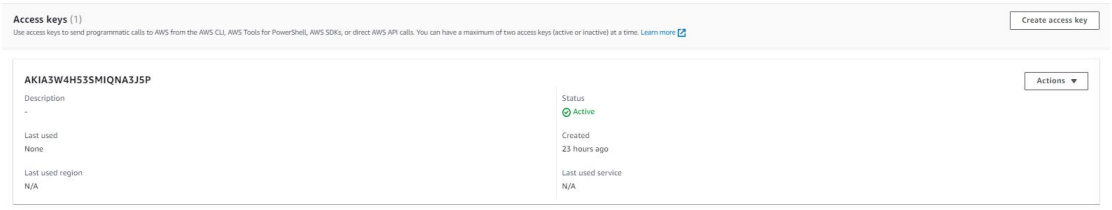
Launch an EC2 Instances - Ubuntu 20.04 - t3.micro - All TCP SG - Launch.



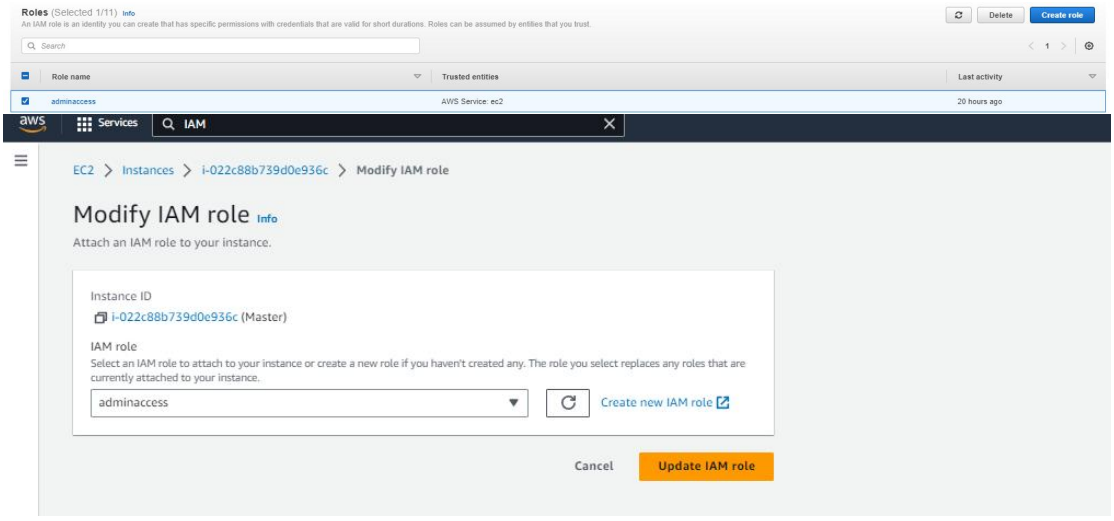
In Master Server- Install Java and Jenkins packages then host a jenkins dashboard.



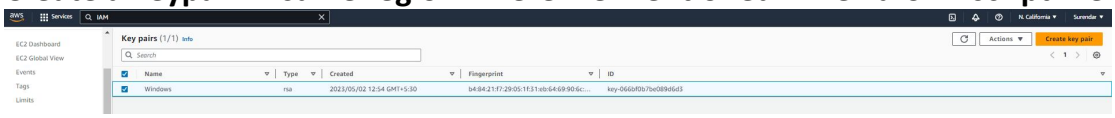
Then in IAM - Create an IAM user get Access key and Secret Key credentials.



Create Roles - Full Admin access - Attach with Master EC2 Server.

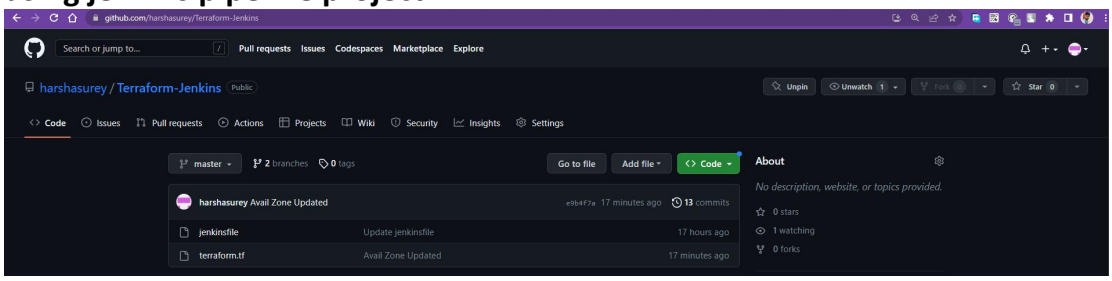


Create a keypair in same region where we mentioned in Terraform script file.

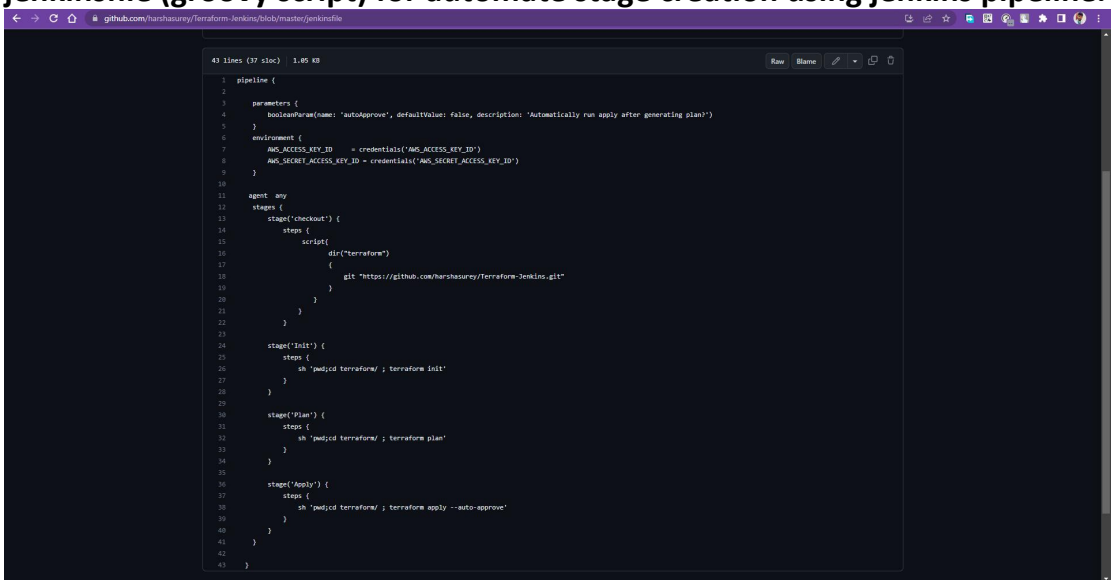


Create a 2 file in your own github repository

Because fetching the script code from github and deploy the code in AWS using jenkins pipeline project.

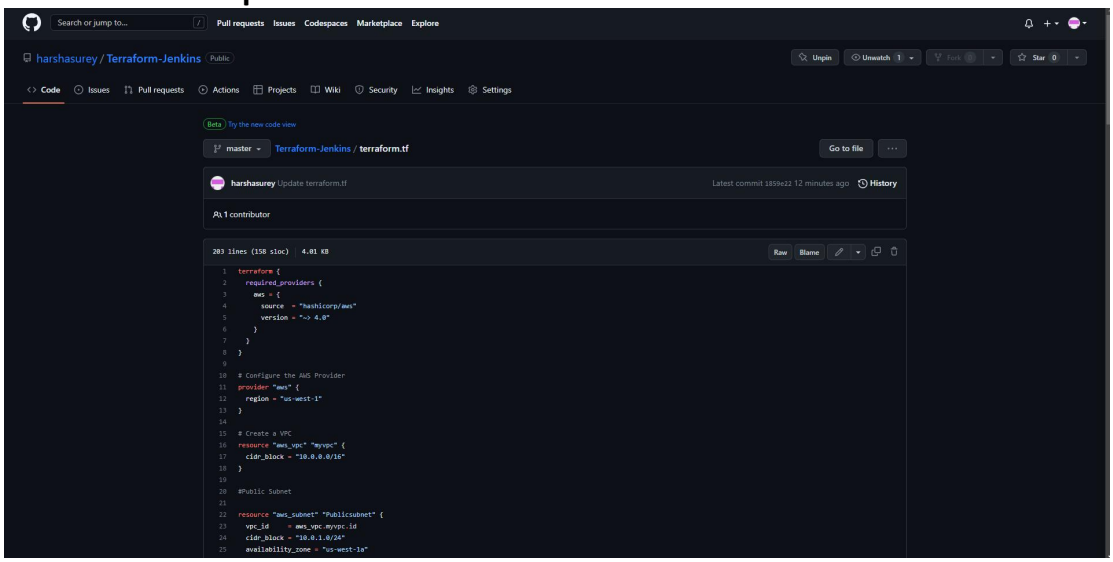


jenkinsfile (groovy script) for automate stage creation using jenkins pipeline.



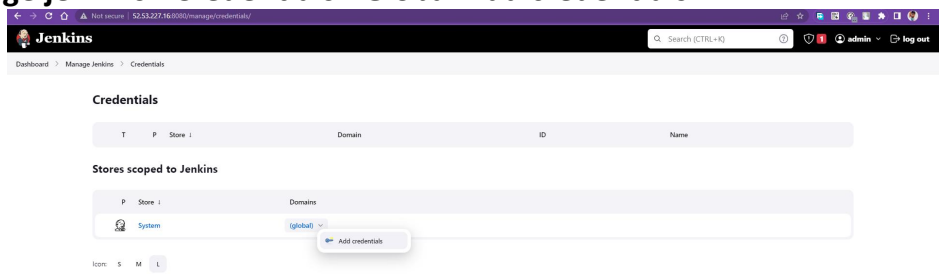
```
pipeline {

    parameters {
        booleanParam(name: 'autoApprove', defaultValue: false, description: 'Automatically run apply after generating plan?')
    }
    environment {
        AWS_ACCESS_KEY_ID     = credentials('AWS_ACCESS_KEY_ID')
        AWS_SECRET_ACCESS_KEY_ID = credentials('AWS_SECRET_ACCESS_KEY_ID')
    }

agent  any
    stages {
        stage('checkout') {
            steps {
                 script{
                        dir("terraform")
                        {
                            git "https://github.com/harshasurey/Terraform-Jenkins.git"
                        }
                    }
                }
            }

        stage('Init') {
            steps {
                sh 'pwd;cd terraform/ ; terraform init'
            }
        }

        stage('Plan') {
            steps {
                sh 'pwd;cd terraform/ ; terraform plan'
            }
        }

        stage('Apply') {
            steps {
                sh 'pwd;cd terraform/ ; terraform apply --auto-approve'
            }
        }
    }

}
```

Terraform.tf Script file which is used to create a VPC and AWS resources



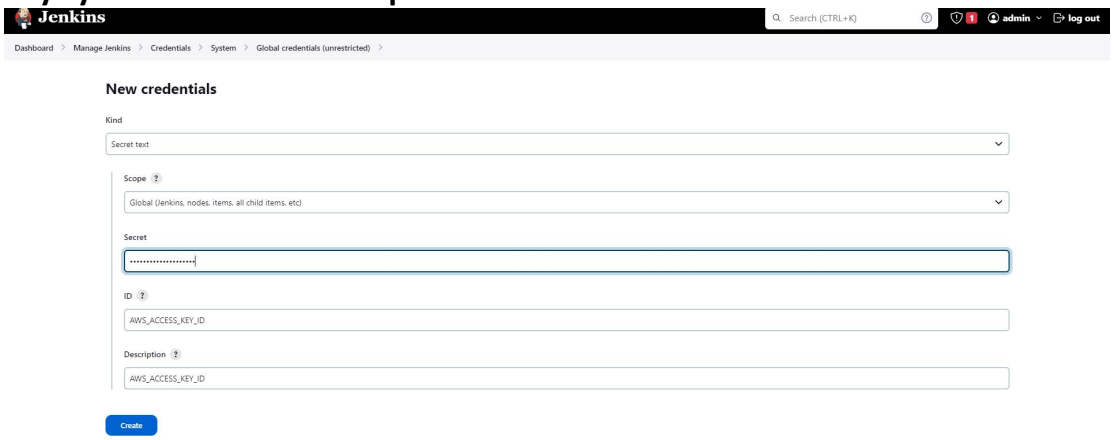
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-west-1"
}

# Create a VPC
resource "aws_vpc" "myvpc" {
  cidr_block = "10.0.0.0/16"
}

#Public Subnet

resource "aws_subnet" "Publicsubnet" {
  vpc_id     = aws_vpc.myvpc.id
  cidr_block = "10.0.1.0/24"
  availability_zone = "us-west-1a"
```

In jenkins dashboard
Manage jenkins - Credentials - Global Add credentials.



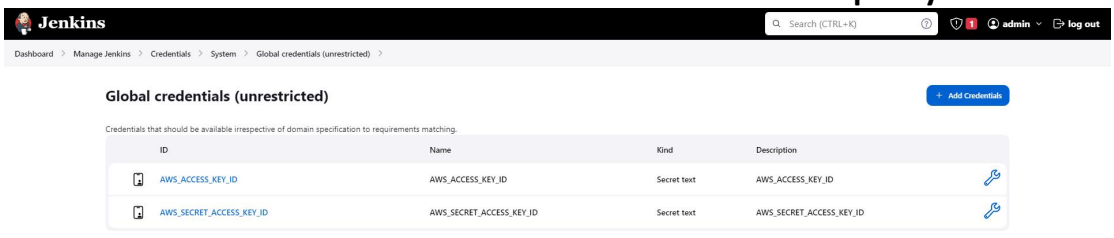
Enter the copy same syntax which is given in jenkins file script.

Choose Secret text - Paste Access Key credential in Secret -Paste the Access key syntax in ID and Descripton - Create.



Same as like Access key choose secret text - Paste Secret key credential - Paste secret key same syntax in ID and Description - Create.

Successfully added both credentials in Global credentials.
These credentials are used to create the AWS Resource in specify AWS acnt.

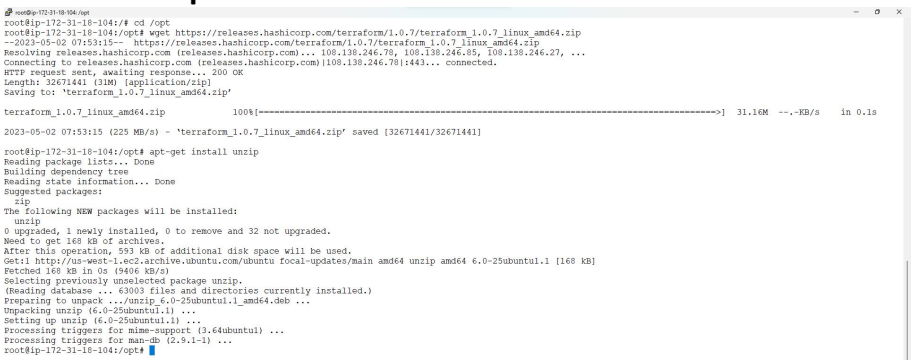


In Master Server

Download terraform zip package using wget cmnd.

wget https://releases.hashicorp.com/terraform/1.0.7/terraform_1.0.7_linux_amd64.zip

#apt-get install unzip



Unzip the terraform zip package

Then Move the Terraform script file to /usr/local/bin

#mv terraform /usr/local/bin

Then check the terraform is active using

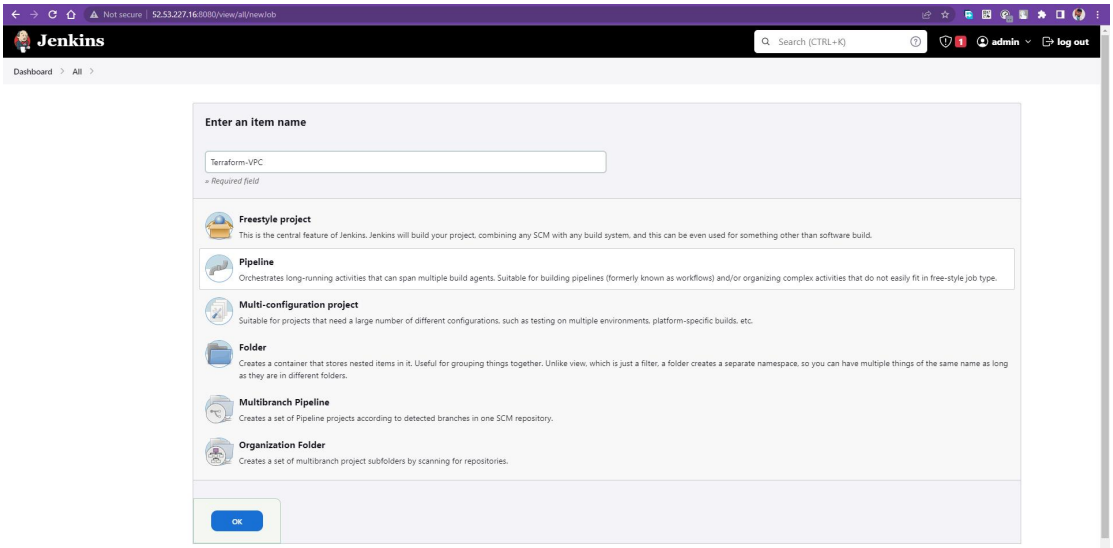
#terraform --version

```
root@ip-172-31-18-104:/opt# ls
terraform_1.0.7_linux_amd64.zip
root@ip-172-31-18-104:/opt# unzip terraform_1.0.7_linux_amd64.zip
Archive:  terraform_1.0.7_linux_amd64.zip
  inflating: terraform
root@ip-172-31-18-104:/opt# ls
terraform  terraform_1.0.7_linux_amd64.zip
root@ip-172-31-18-104:/opt# mv terraform /usr/local/bin/
root@ip-172-31-18-104:/opt# terraform --version
Terraform v1.0.7
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.4.6. You can update by downloading from https://www.terraform.io/downloads.html
root@ip-172-31-18-104:/opt#
```

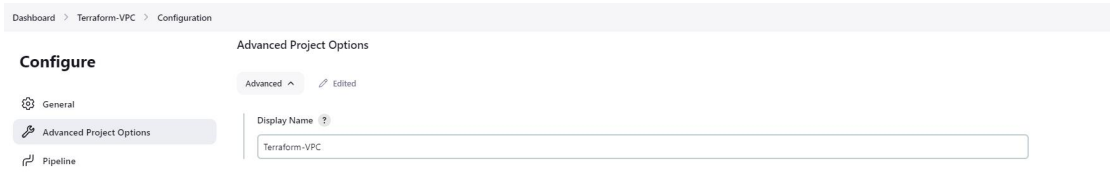
In Jenkins dashboard

Create a pipeline project

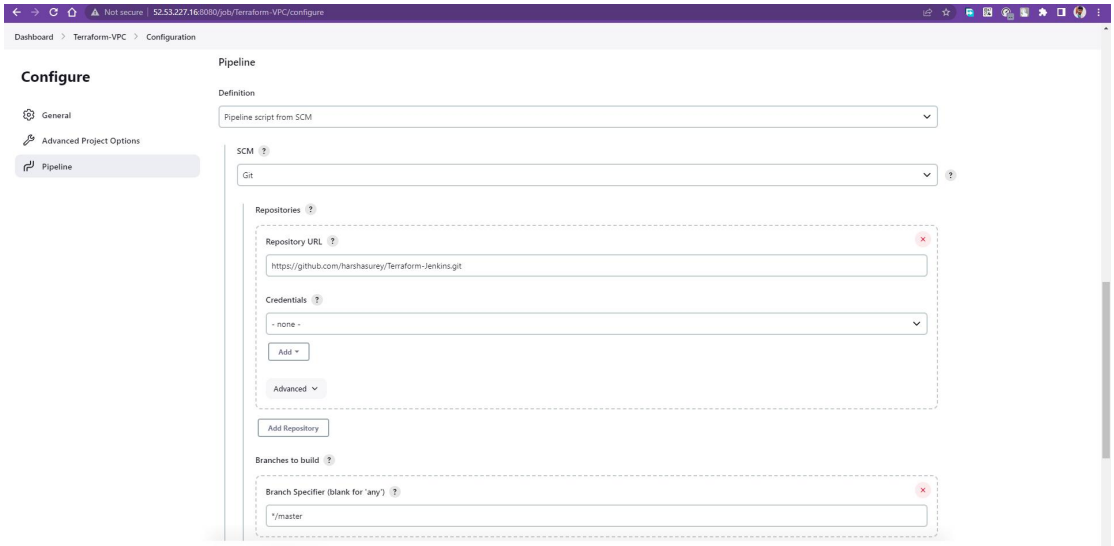
Terraform-VPC



Then in Display Name
Enter the name how you need to display the pipeline project.
Ex: Terraform-VPC



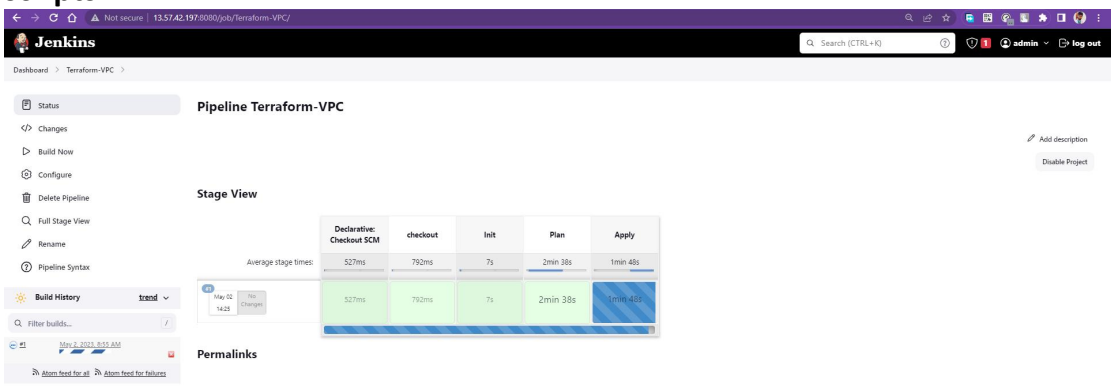
In pipeline
Choose definition
Pipeline script from SCM (Source Control Management)
SCM
Choose Git
Repository URL
https://github.com/harshasurey/Terraform-Jenkins.git
Branch
Choose the branch where the script files are present
./master



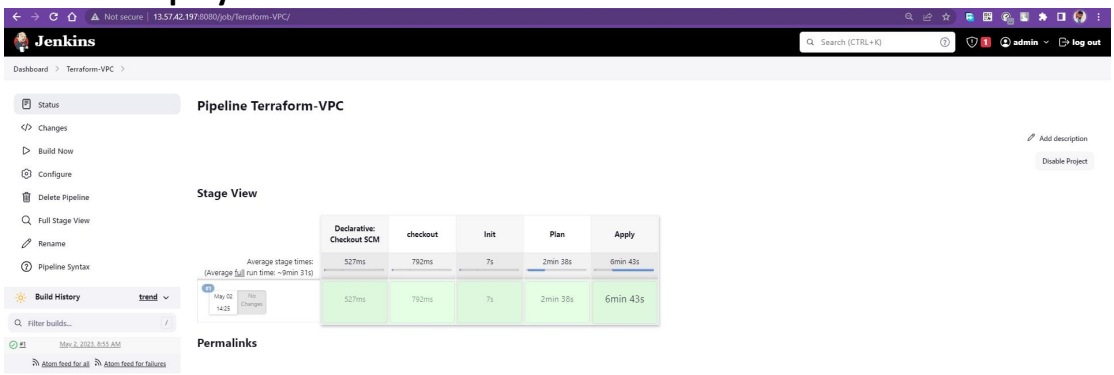
In Script path
Choose the jenkinsfile
Apply and Save.

Then Build the pipeline project.

Can able to view the stages are auto deploying as following jenkinsfile scripts.

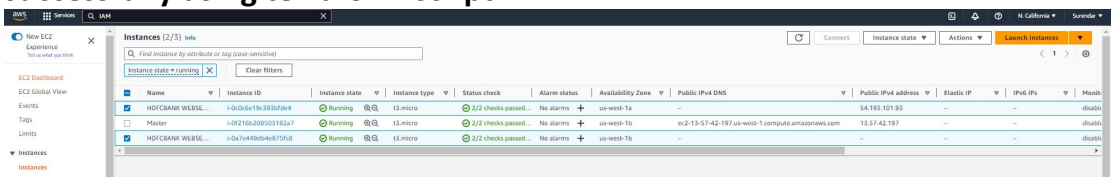


After Successfully Created all the stages it means VPC and AWS resources are auto deployed.

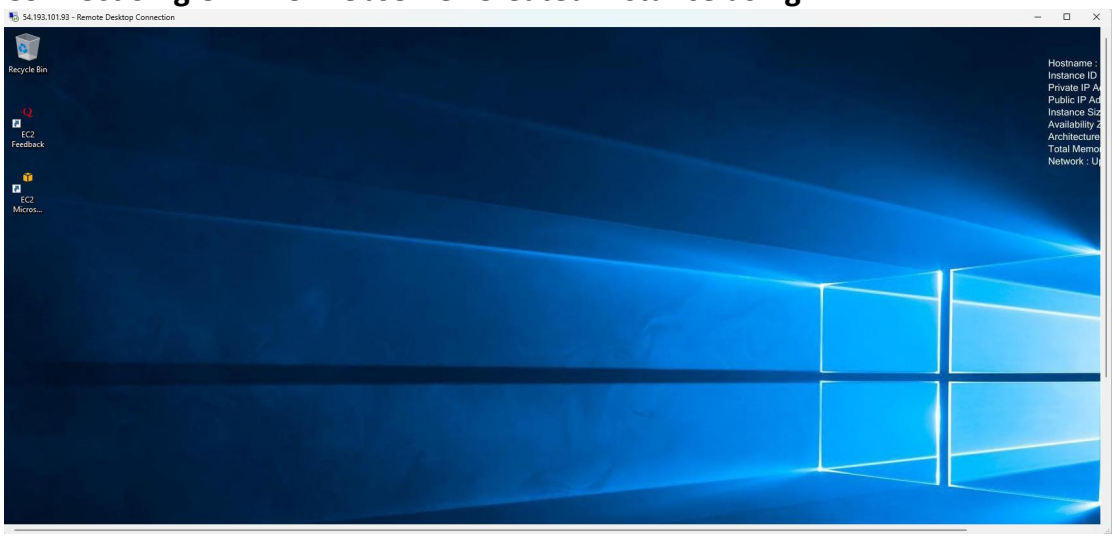


In EC2 Instances

Can able to view that the HDFC Webserver both Instances are launched successfully using terraform script.



Connect Single HDFC Webserver Created Instance using RDP



After successfully connected one window server.
Open RDP and enter the Second HDFC webServer Instances Credentials and Connected another window Server.

Successfully Created VPC and AWS instances using Terraform script in Jenkins.