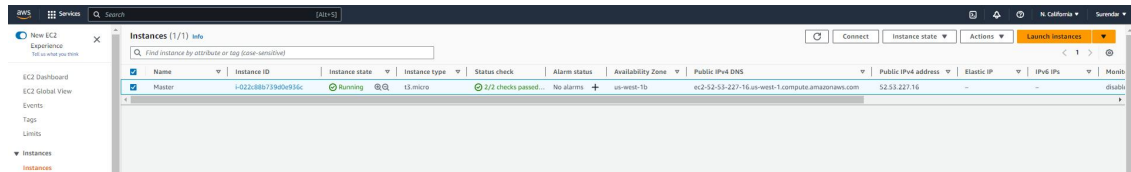


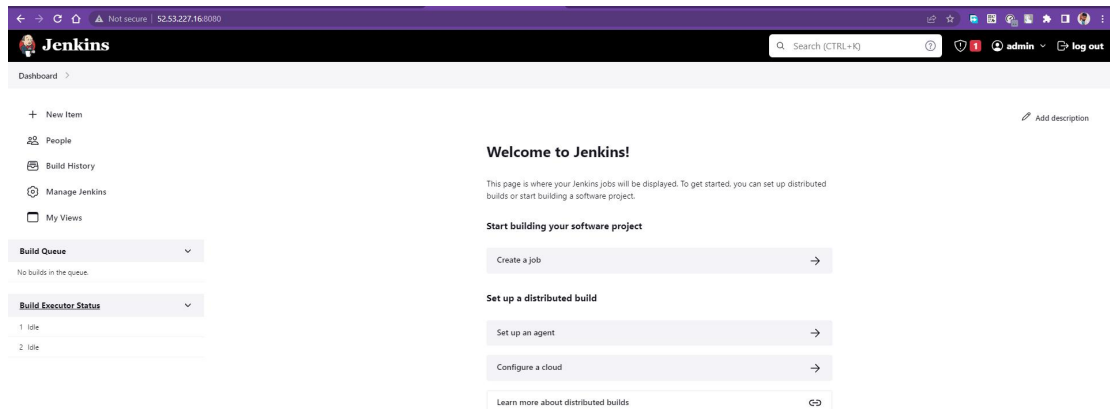
DevOps Project Task

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

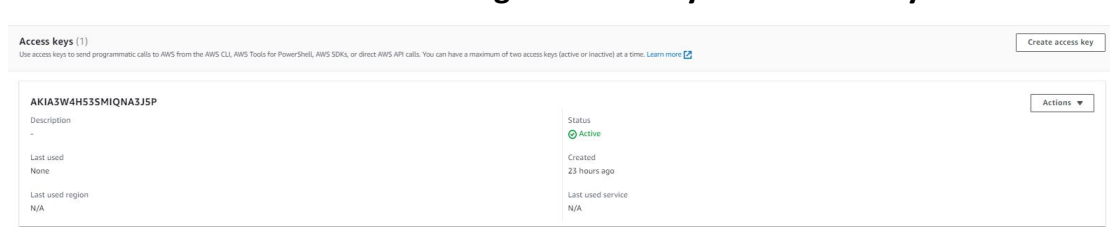
Launch an EC2 Instances - Ubuntu 20.04 - t3.micro - All TCP SG - Launch.



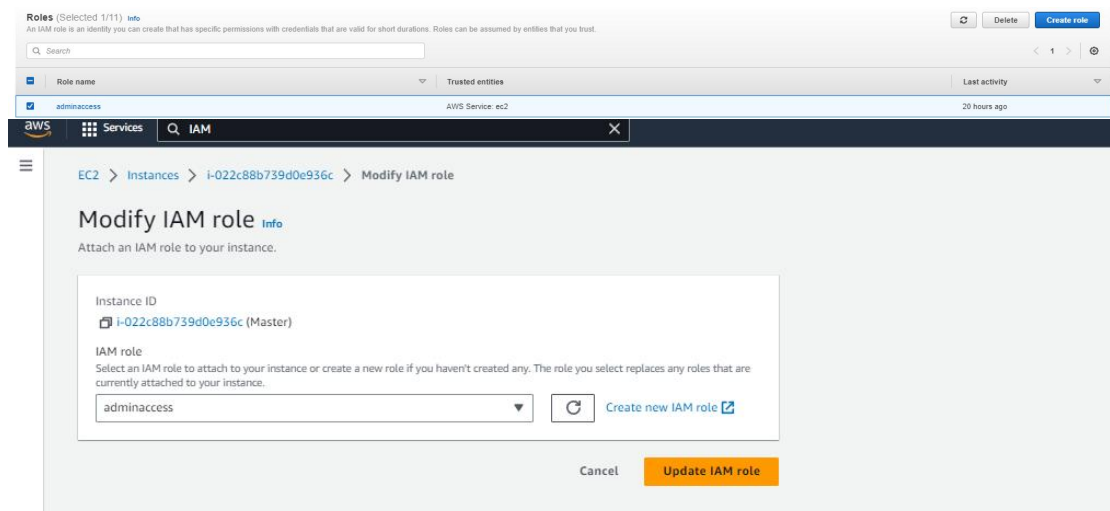
In Master Server- Install Java and Jenkins packages then host a jenkins dashboard.



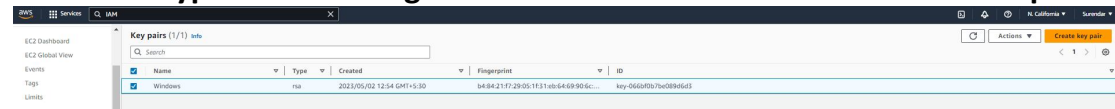
Then in IAM - Create an IAM user get Access key and Secret Key credentials.



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



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

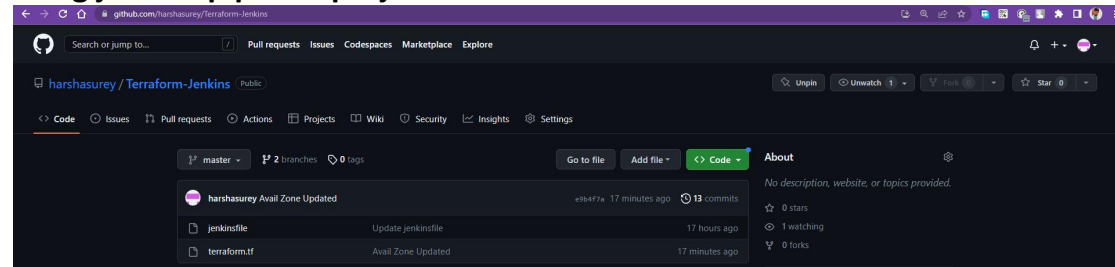


The screenshot shows the AWS IAM console 'Key pairs' page. A table lists one key pair:

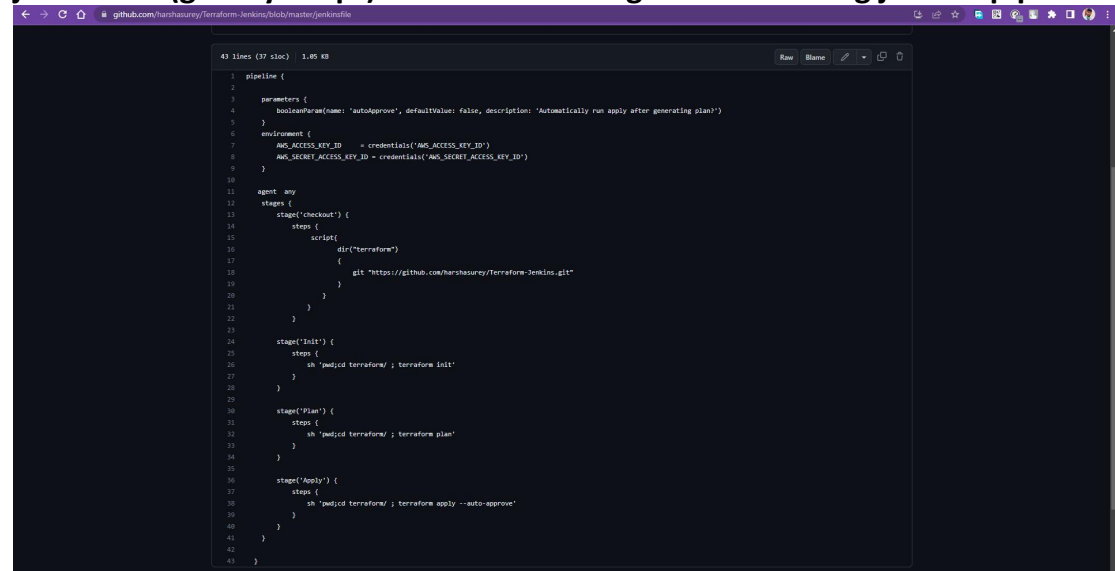
Name	Type	Created	Fingerprint	ID
rsa	rsa	2023/05/02 12:54 GMT+5:30	848421f7-20051f31ab6469-906c...	key-066d02b7uo0896d3

Create a 2 file in your own github repository

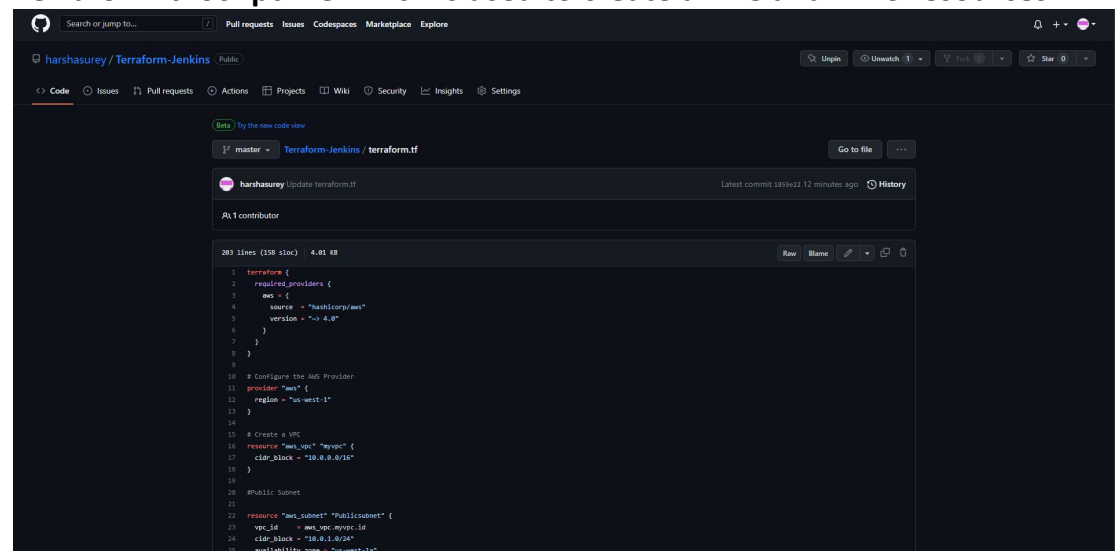
Because fetching the script code from github and deploy the code in AWS using jenkins pipeline project.



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

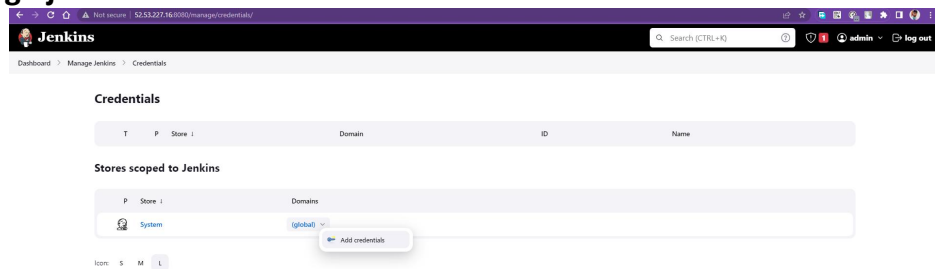


Terraform.tf Script file which is used to create a VPC and AWS resources

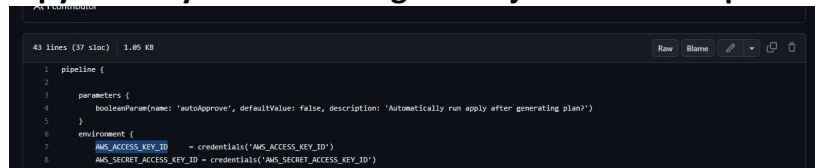


In jenkins dashboard

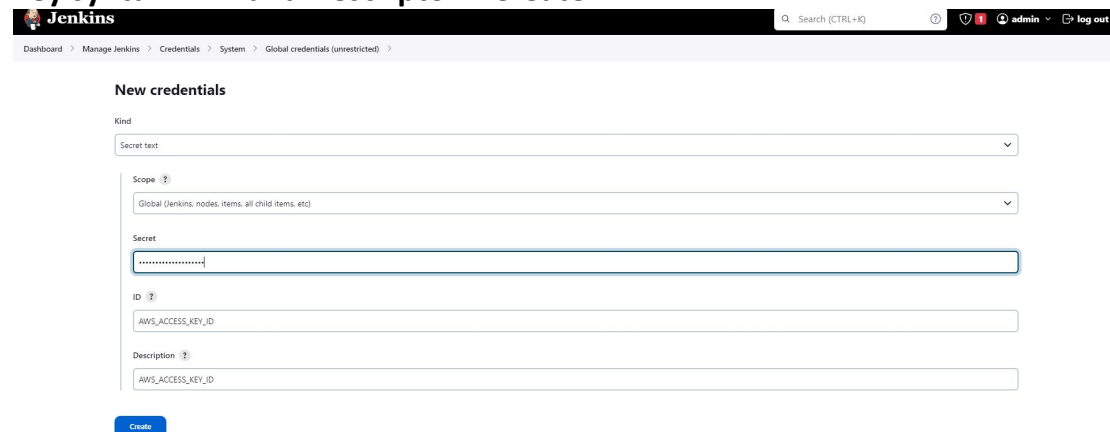
Manage jenkins - Credentials - Global Add credentials.



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



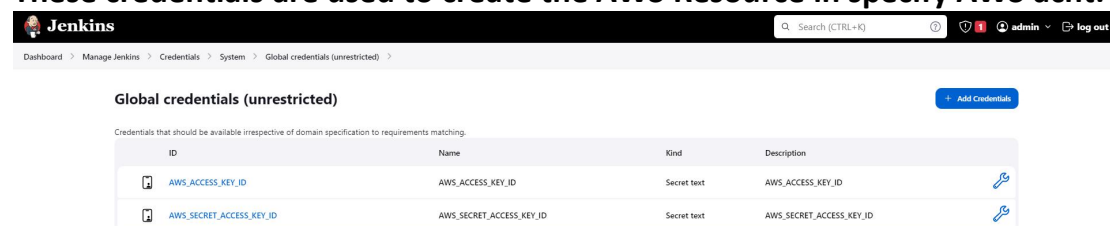
Choose Secret text - Paste Access Key credential in Secret - Paste the Access key syntax in ID and Description - Create.



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 acct.



In Master Server

Download terraform zip package using wget cmd.

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

#apt-get install unzip

```
root@ip-172-31-18-104:/opt# cd /opt
root@ip-172-31-18-104:/opt# wget https://releases.hashicorp.com/terraform/1.0.7/terraform_1.0.7_linux_amd64.zip
--2023-05-02 07:53:15-- https://releases.hashicorp.com/terraform/1.0.7/terraform_1.0.7_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 108.138.246.79, 108.138.246.85, 108.138.246.27, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|108.138.246.79|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32671441 (31M) [application/zip]
Saving to: 'terraform_1.0.7_linux_amd64.zip'

terraform_1.0.7_linux_amd64.zip 100%[=====] 31.1GM --.-KB/s in 0.1s

2023-05-02 07:53:15 (225 MB/s) - 'terraform_1.0.7_linux_amd64.zip' saved [32671441/32671441]

root@ip-172-31-18-104:/opt# apt-get install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
unzip
The following NEW packages will be installed:
unzip
0 upgraded, 1 newly installed, 0 to remove and 32 not upgraded.
Need to get 168 kB of archives.
After this operation, 593 kB of additional disk space will be used.
Get:1 https://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 unzip amd64 6.0-25ubuntu1.1 [168 kB]
Fetched 168 kB in 0s (9406 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 63003 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-25ubuntu1.1_amd64.deb ...
Unpacking unzip (6.0-25ubuntu1.1) ...
Setting up unzip (6.0-25ubuntu1.1) ...
Processing triggers for man-db (2.9.1-1) ...
root@ip-172-31-18-104:/opt#
```

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

Enter an item name

Terraform-VPC

* Required field

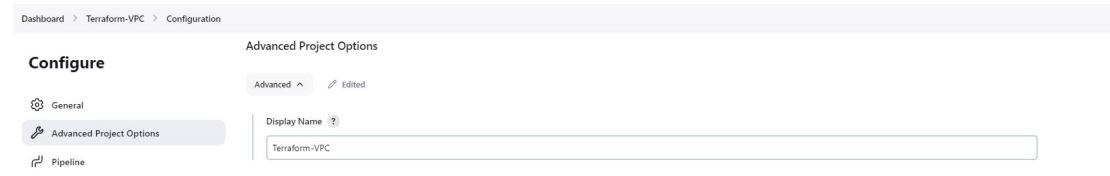
- Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Then in Display Name

Enter the name how you need to display the pipeline project.

Ex: Terraform-VPC



Dashboard > Terraform-VPC > Configuration

Configure

Advanced Project Options

Advanced ^ Edited

General

Advanced Project Options

Pipeline

Display Name ?

Terraform-VPC

In pipeline

Choose definition

Pipeline script from SCM (Source Control Management)

SCM

Choose Git

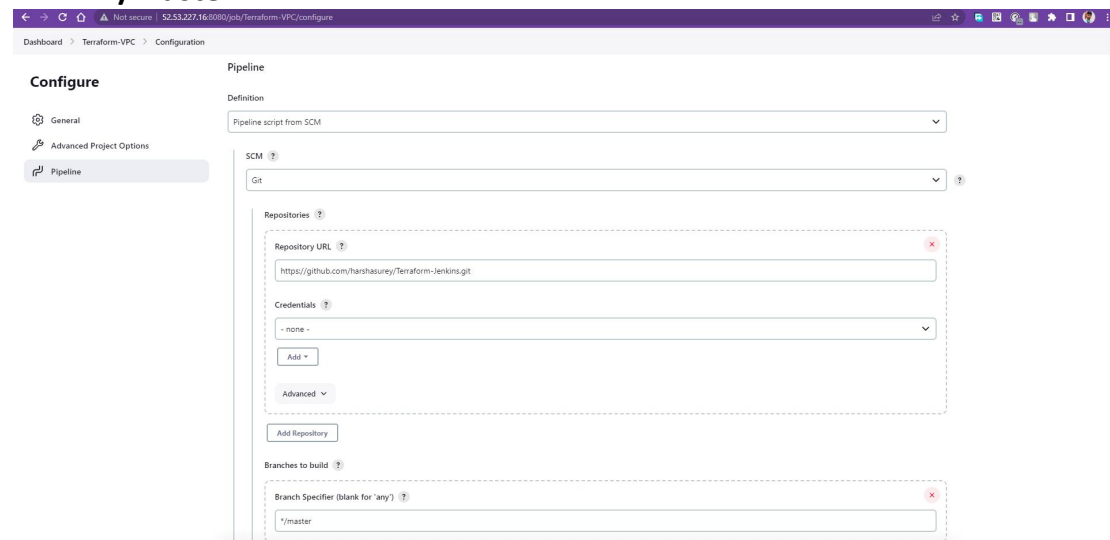
Repository URL

<https://github.com/harshasurey/Terraform-Jenkins.git>

Branch

Choose the branch where the script files are present

./master



Dashboard > Terraform-VPC > Configuration

Configure

General

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

<https://github.com/harshasurey/Terraform-Jenkins.git>

Credentials ?

none

Add +

Advanced v

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

In Script path

Choose the jenkinsfile

Apply and Save.



Add +

Script Path ?

jenkinsfile

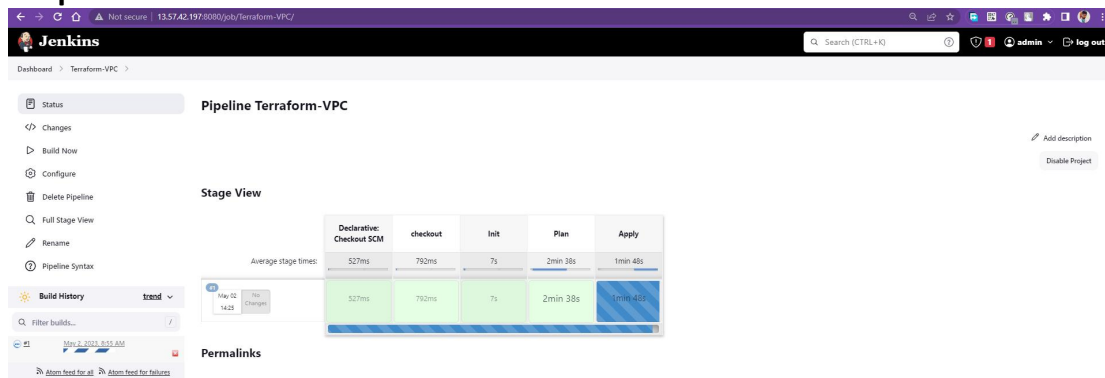
☒ Lightweight checkout ?

Pipeline Syntax

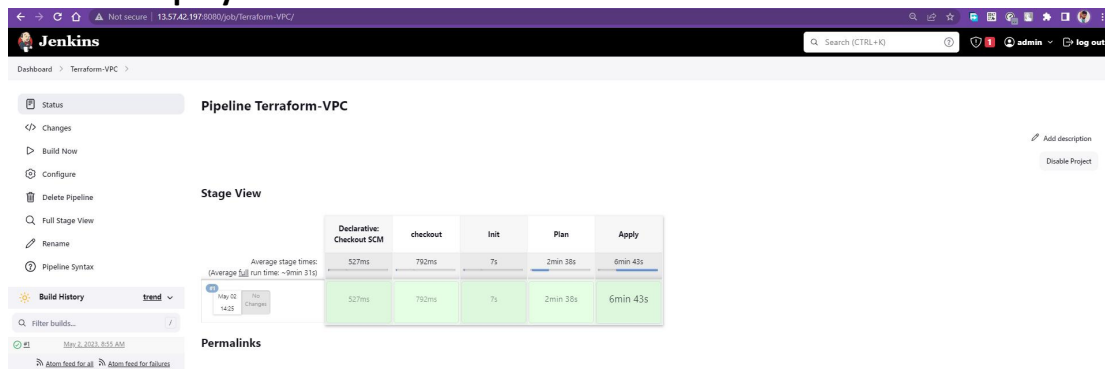
Save Apply

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.



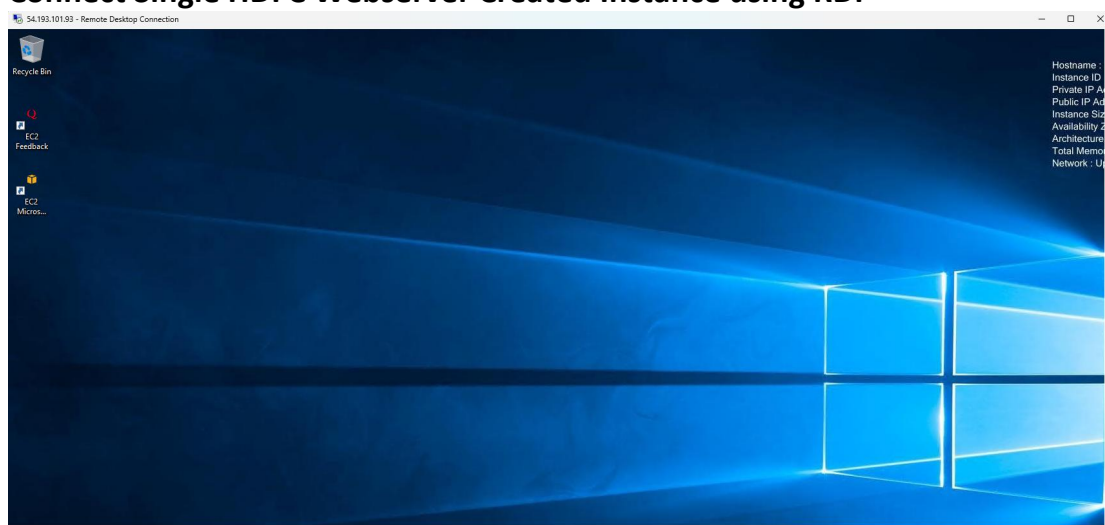
In EC2 Instances

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

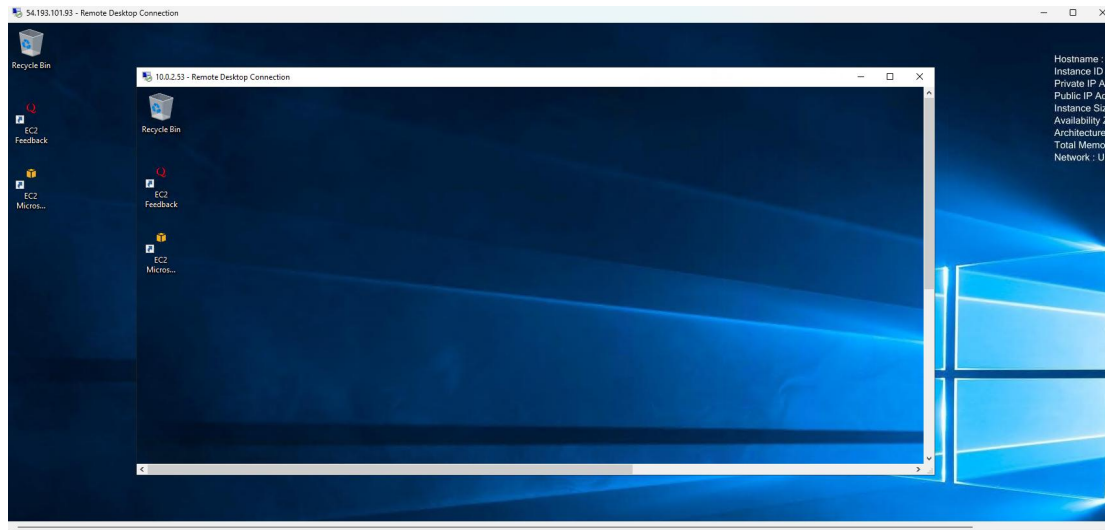
The screenshot shows the AWS Management Console 'Instances' page. It displays a table of EC2 instances, including their names, IDs, states, types, and public IP addresses.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Elastic IP	IPv6 IP	Monitor
HDFCBANK WEBS...	i-0c0d479c3930f6a4	Running	t3.micro	2/2 checks passed...	No alarms	us-west-1a	-	54.185.101.83	-	-	disabled
Master	i-0f216d208503182a7	Running	t3.micro	2/2 checks passed...	No alarms	us-west-1b	ec2-13.57.42-197-us-west-1-compute.amazonaws.com	13.57.42.197	-	-	disabled
HDFCBANK WEBS...	i-0a7e448b4d8795f8	Running	t3.micro	2/2 checks passed...	No alarms	us-west-1b	-	-	-	-	disabled

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.**