

# Project Documentation: CI/CD Pipeline Using Jenkins and Terraform on AWS

---

## 1. Project Overview

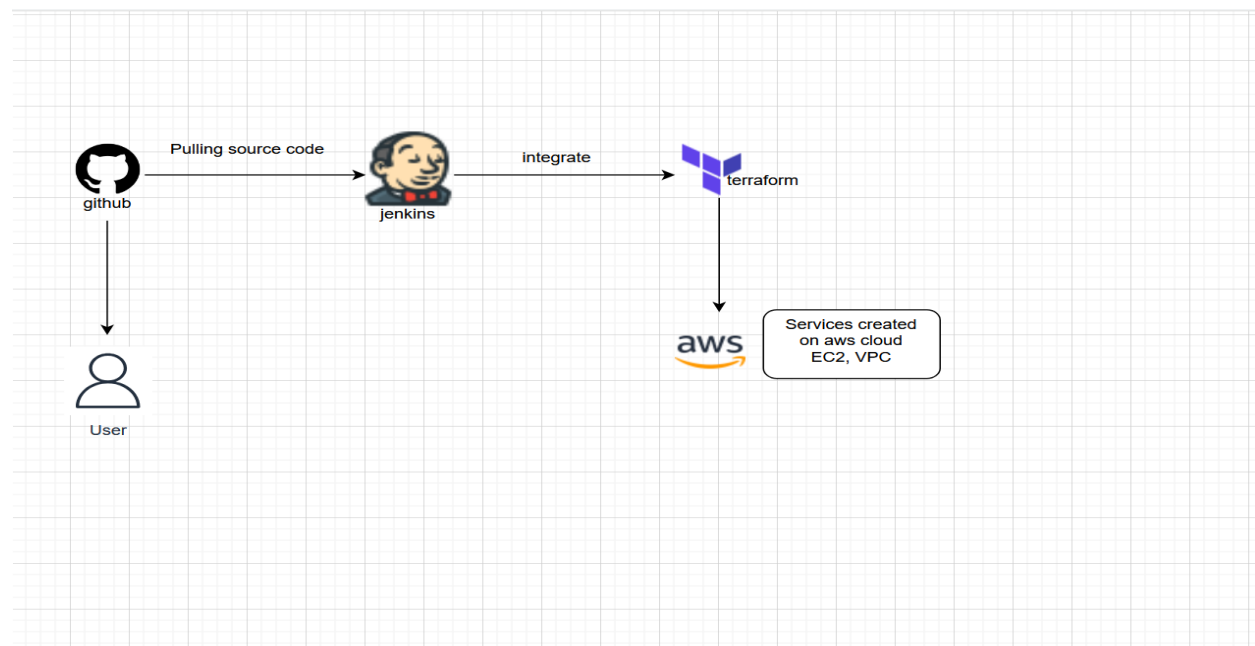
This project demonstrates how to automate the provisioning of AWS infrastructure using **Jenkins** and **Terraform** in a CI/CD pipeline.

The goal is to integrate Infrastructure as Code (IaC) into a continuous delivery workflow, allowing infrastructure to be created, validated, and managed automatically from a Jenkins pipeline using Terraform and AWS services.

### Key Components:

- **Jenkins** → Automation server for CI/CD.
  - **Terraform** → Infrastructure as Code tool for AWS resource provisioning.
  - **AWS** → Cloud provider hosting the infrastructure (EC2, VPC, Subnet, S3).
  - **S3 Bucket** → Used as Terraform backend for remote state management.
  - **GitHub** → Version control system for storing Terraform and Jenkins files.
- 

## 2. Architecture Diagram



### 3. Prerequisites

Before starting the setup, ensure the following requirements are met:

- **AWS Account**
- **IAM User** with permissions:
  - AdministratorAccess
  - AmazonS3FullAccess
- **S3 Bucket** (for Terraform state file) → infra-terraform-state-aniket-2025
- **EC2 Instance** (Ubuntu 22.04) for Jenkins and Terraform
- **Key Pair** → account-a (used by Terraform for SSH access)
- **Public GitHub Repository** → jenkins-terraform-aws-cicd

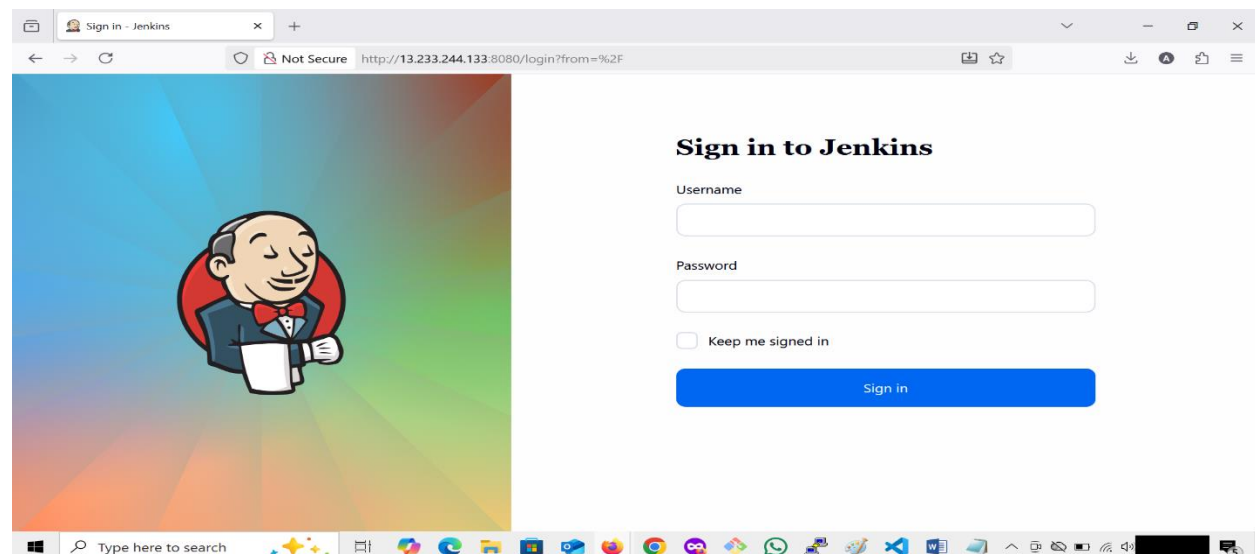
### 4. Environment Setup

Jenkins Installation on EC2

```
root@ip-172-31-32-166: /home/ubuntu
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-32-166:/home/ubuntu# sudo systemctl start jenkins
root@ip-172-31-32-166:/home/ubuntu# sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-32-166:/home/ubuntu# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-11-11 05:47:36 UTC; 1min 41s ago
     Main PID: 5282 (java)
       Tasks: 43 (limit: 4670)
      Memory: 582.0M
         CPU: 18.53s
    CGroup: /system.slice/jenkins.service
            └─5282 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Nov 11 05:47:31 ip-172-31-32-166 jenkins[5282]: [LF]> This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Nov 11 05:47:31 ip-172-31-32-166 jenkins[5282]: [LF]>
Nov 11 05:47:31 ip-172-31-32-166 jenkins[5282]: [LF]> *****
Nov 11 05:47:31 ip-172-31-32-166 jenkins[5282]: [LF]> *****
Nov 11 05:47:36 ip-172-31-32-166 jenkins[5282]: 2025-11-11 05:47:36.821+0000 [id=37] INFO jenkins.InitReactorRunner$1#onAttained: Completed in
Nov 11 05:47:36 ip-172-31-32-166 jenkins[5282]: 2025-11-11 05:47:36.842+0000 [id=30] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully
Nov 11 05:47:36 ip-172-31-32-166 systemd[1]: Started Jenkins Continuous Integration Server.
Nov 11 05:47:38 ip-172-31-32-166 jenkins[5282]: 2025-11-11 05:47:38.332+0000 [id=56] INFO h.m.DownloadService$Downloadable#load: Obtained the
Nov 11 05:47:38 ip-172-31-32-166 jenkins[5282]: 2025-11-11 05:47:38.334+0000 [id=56] INFO hudson.util.Retrier#start: Performed the action check

root@ip-172-31-32-166:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
90a8eb0821b94296acb362891a64ea59
root@ip-172-31-32-166:/home/ubuntu# history
1 clear
2 sudo apt update
3 sudo apt install fontconfig openjdk-21-jre -y
4 sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
5 echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
6 sudo apt update
7 sudo apt install jenkins -y
8 sudo systemctl start jenkins
9 sudo systemctl enable jenkins
10 sudo systemctl status jenkins
11 cat /var/lib/jenkins/secrets/initialAdminPassword
12 history
root@ip-172-31-32-166:/home/ubuntu#
```



## Terraform Installation on Jenkins EC2

```
root@ip-172-31-32-166: /home/ubuntu
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-32-166:/home/ubuntu# unzip terraform_1.7.0_linux_amd64.zip
Archive:  terraform_1.7.0_linux_amd64.zip
  inflating: terraform
root@ip-172-31-32-166:/home/ubuntu# sudo mv terraform /usr/local/bin/
root@ip-172-31-32-166:/home/ubuntu# terraform -v
Terraform v1.7.0
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.13.5. You can update by downloading from https://www.terraform.io/downloads.html
root@ip-172-31-32-166:/home/ubuntu# history
1  clear
2  sudo apt update
3  sudo apt install fontconfig openjdk-21-jre -y
4  sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
5  echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenki
ns.list > /dev/null
6  sudo apt update
7  sudo apt install jenkins -y
8  sudo systemctl start jenkins
9  sudo systemctl enable jenkins
10 sudo systemctl status jenkins
11 cat /var/lib/jenkins/secrets/initialAdminPassword
12 history
13 clear
14 wget https://releases.hashicorp.com/terraform/1.7.0/terraform_1.7.0_linux_amd64.zip
15 unzip terraform_1.7.0_linux_amd64.zip
16 apt install unzip
17 unzip terraform_1.7.0_linux_amd64.zip
18 sudo mv terraform /usr/local/bin/
19 terraform -v
20 history
root@ip-172-31-32-166:/home/ubuntu#
```

## AWS CLI Installation

```
root@ip-172-31-32-166: /home/ubuntu
creating: aws/dist/awscli/customizations/wizard/wizards/events/
creating: aws/dist/awscli/customizations/wizard/wizards/iam/
creating: aws/dist/awscli/customizations/wizard/wizards/lambda/
inflating: aws/dist/awscli/customizations/wizard/wizards/dynamodb/new-table.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/lambda/new-function.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/events/new-rule.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/iam/new-role.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/configure/_main.yml
inflating: aws/dist/awscli/customizations/aso/index.html
inflating: aws/dist/awscli/topics/s3-faq.rst
inflating: aws/dist/awscli/topics/s3-config.rst
inflating: aws/dist/awscli/topics/ddb-expressions.rst
inflating: aws/dist/awscli/topics/topic-tags.json
inflating: aws/dist/awscli/topics/config-vars.rst
inflating: aws/dist/awscli/topics/return-codes.rst
inflating: aws/dist/awscli/data/metadata.json
inflating: aws/dist/awscli/data/ac.index
inflating: aws/dist/awscli/data/cli.json
creating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/METADATA
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/RECORD
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/AUTHORS.rst
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
root@ip-172-31-32-166:/home/ubuntu# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
root@ip-172-31-32-166:/home/ubuntu# history
1 sudo apt update -y
2 sudo apt install unzip -y
3 curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
4 unzip awscliv2.zip
5 sudo ./aws/install
6 history
root@ip-172-31-32-166:/home/ubuntu#
```

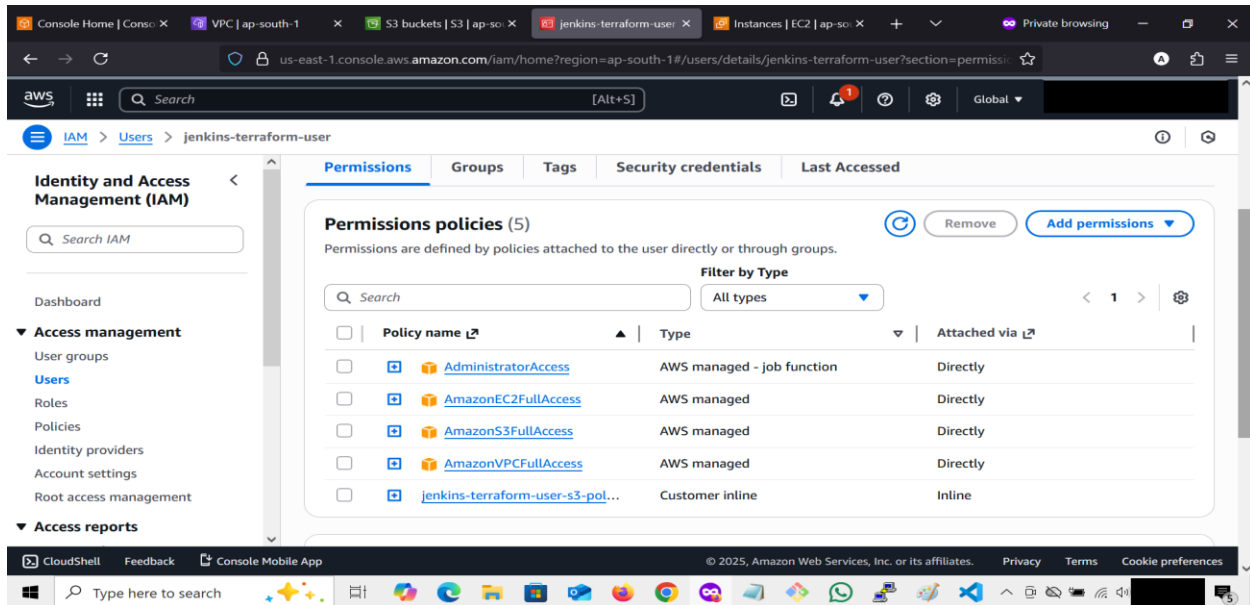
## 5. AWS Configuration

### IAM User and Permissions

Created a dedicated IAM user jenkins-terraform-user with these policies:

- AdministratorAccess
- AmazonS3FullAccess

Used for Jenkins → Terraform AWS authentication.



### S3 Bucket for Terraform State

#### What is a Terraform State File?

When Terraform creates or updates infrastructure, it keeps track of all deployed resources in a **state file** named terraform.tfstate.

This file acts as a **single source of truth** for Terraform, it records details about every AWS resource created (like EC2 IDs, subnet IDs, VPC IDs, etc.), allowing Terraform to understand the current infrastructure state and make only necessary changes in the future.

Without this file, Terraform would have no idea what's already deployed, leading to duplicated or mismatched resources.

#### Why Use a Remote Backend (S3)?

By default, Terraform stores its state file locally (on the Jenkins server).

However, this is not ideal for collaborative or CI/CD environments.

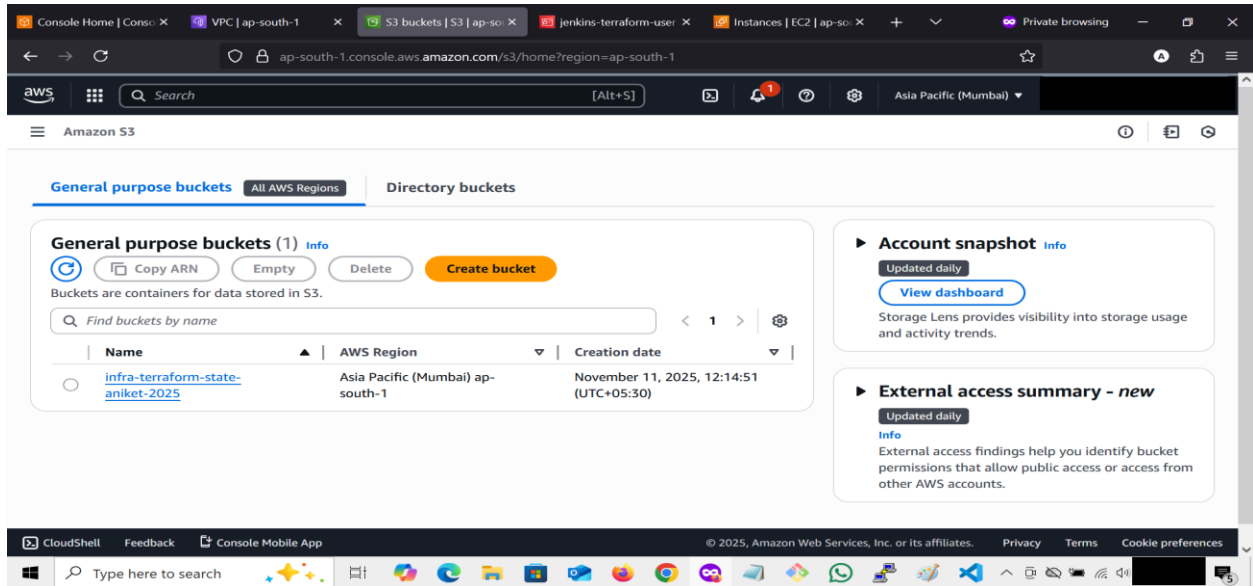
That's why we use an **AWS S3 bucket** as a **remote backend** to

S3 bucket name: **infra-terraform-state-aniket-2025**

Region: ap-south-1

Used to store Terraform state files remotely for consistency and collaboration.

S3 bucket view in AWS Console



## 6. Jenkins Configuration

Required Jenkins Plugins

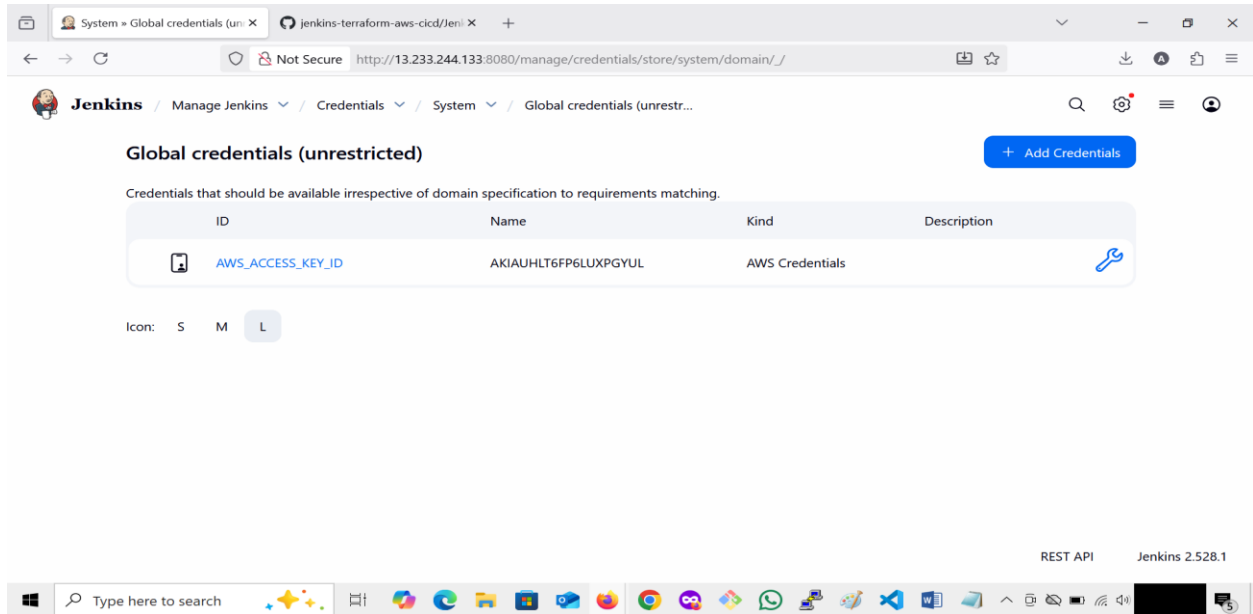
Installed via **Manage Jenkins** → **Manage Plugins** → **Available**:

- **AWS Credentials Plugin**
- **Git Plugin**
- **Pipeline Plugin**
- **Terraform Plugin (optional)**

Jenkins Credentials

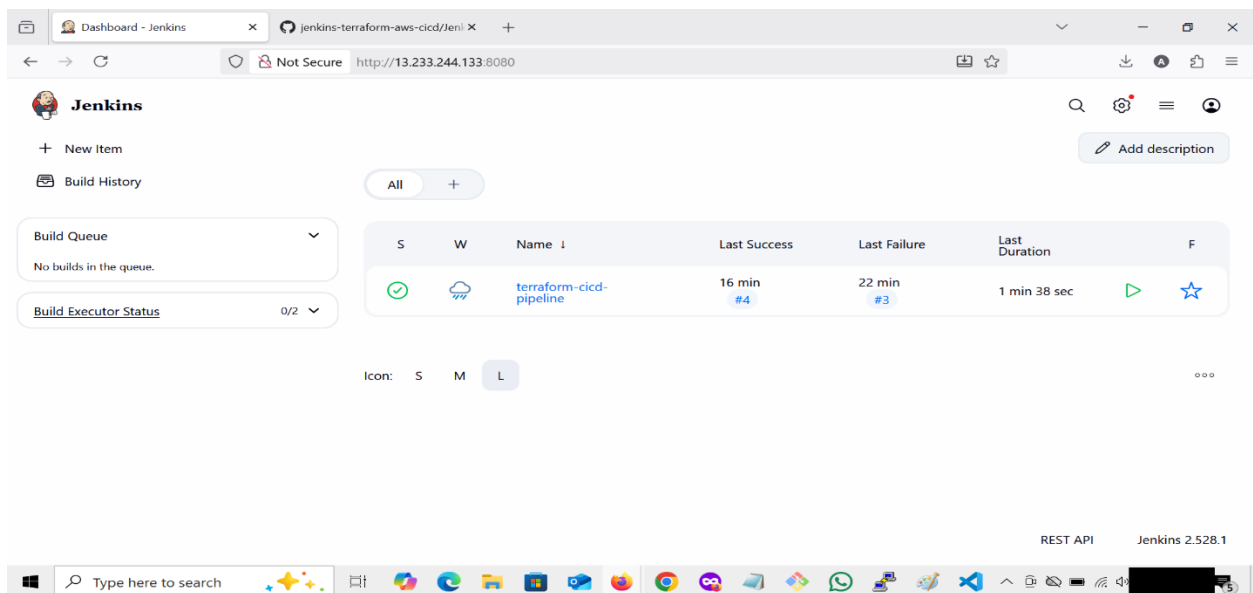
Stored AWS keys securely:

- Go to **Manage Jenkins** → **Manage Credentials**
- Add New:
  - **Kind**: AWS Credentials
  - **ID**: AWS\_ACCESS\_KEY\_ID
  - **Access Key / Secret Key**: from IAM user



## Jenkins Pipeline Configuration

1. Go to **Jenkins** → **New Item** → **Pipeline**
2. Set:
  - **Definition:** Pipeline script from SCM
  - **SCM:** Git
  - **Repository URL:** <https://github.com/anikettalwekar/jenkins-terraform-aws-cicd.git>
  - **Branch:** main



## 7. Terraform Configuration Files (Stored in GitHub)

- main.tf: Defines AWS VPC, Subnet, Security Group, and EC2 instance.
- variables.tf: Defines variable for key\_name = "account-a".
- outputs.tf: Outputs instance details (ID, public IP, VPC ID).
- provider.tf: Configures AWS provider region and backend (S3).

File	Purpose
main.tf	Core infrastructure (VPC, subnet, EC2, SG)
variables.tf	Dynamic input parameters (key_name, region, etc.)
outputs.tf	Returns useful resource details
provider.tf	Connects Terraform to AWS region and backend

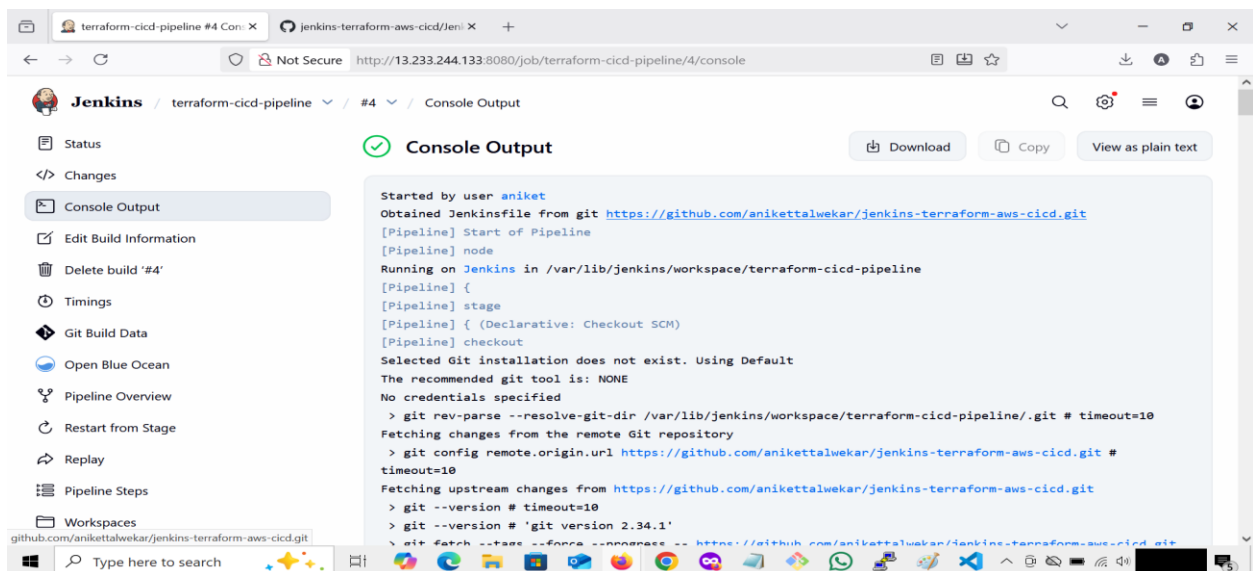
## 8. Jenkinsfile Explanation

Jenkinsfile defines all pipeline stages:

This provides **automated, repeatable infrastructure delivery**.

Stage	Purpose
Checkout Code	Pulls latest Terraform files from GitHub
Verify AWS Credentials	Validates IAM access using AWS CLI
Terraform Init	Initializes backend and providers
Terraform Validate	Checks syntax and configuration
Terraform Plan	Previews resource creation
Terraform Apply	Provisions AWS resources
Terraform Destroy (Optional)	Removes infrastructure

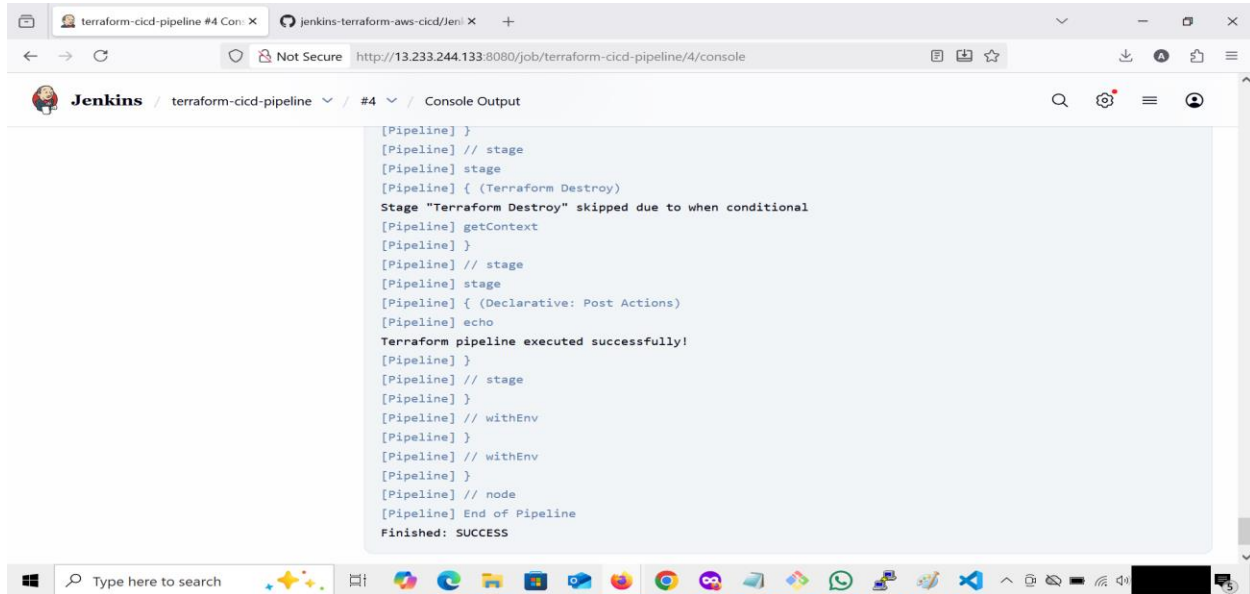
Jenkins console (pipeline start)



The screenshot shows the Jenkins web interface. The left sidebar contains navigation links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#4', Timings, Git Build Data, Open Blue Ocean, Pipeline Overview, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The main area displays the 'Console Output' for build #4. The output text is as follows:

```
Started by user aniket
Obtained Jenkinsfile from git https://github.com/aniketallwekar/jenkins-terraform-aws-cicd.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/terraform-cicd-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/terraform-cicd-pipeline/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/aniketallwekar/jenkins-terraform-aws-cicd.git #
timeout=10
Fetching upstream changes from https://github.com/aniketallwekar/jenkins-terraform-aws-cicd.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/aniketallwekar/jenkins-terraform-aws-cicd.git
```

## Jenkins console (pipeline end)

A screenshot of a web browser displaying the Jenkins console output for a pipeline named 'terraform-cicd-pipeline'. The browser's address bar shows the URL 'http://13.233.244.133:8080/job/terraform-cicd-pipeline/4/console'. The Jenkins interface shows the pipeline's console output, which includes stage definitions and a final success message. The output text is as follows:

```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Terraform Destroy)
Stage "Terraform Destroy" skipped due to when conditional
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Terraform pipeline executed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## 9. AWS Infrastructure Verification

After the successful Jenkins pipeline execution, Terraform automatically provisioned AWS resources as defined in the Terraform configuration files (main.tf, variables.tf, and provider.tf).

### Resources Created by Terraform

Resource Type	Name	Description
VPC	infra-project-vpc	Custom Virtual Private Cloud created to host network resources
Subnet	infra-project-subnet	Public subnet within the VPC for EC2 instance deployment
Security Group	infra-project-sg	Allows inbound SSH (22) and HTTP (8080) traffic
EC2 Instance	infra-project-ec2	EC2 instance automatically created by Terraform through Jenkins pipeline



## EC2

Successfully initiated termination (deletion) of i-0b2ea7167f0541bf3

**Instances (2)** Info Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch Instances](#)

Find Instance by attribute or tag (case-sensitive)

[running](#) [Clear filters](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	
<input type="checkbox"/>	jenkins-terraform-server	i-07a162d6eee733efe	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1a	e
<input type="checkbox"/>	infra-project-ec2	i-0d34f1b140a489d44	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1a	e

Select an instance

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

## VPC

**Your VPCs (2)** Info Last updated less than a minute ago [Actions](#) [Create VPC](#)

Find VPCs by attribute or tag

	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	<a href="#">vpc-02311120e9f9a1a85</a>	Available	Off	172.31.0.0/16	-
<input type="checkbox"/>	infra-project-vpc	<a href="#">vpc-0385396d770ec109b</a>	Available	Off	10.0.0.0/16	-

Select a VPC above

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

## Subnet

The screenshot shows the AWS Management Console interface for a subnet. The breadcrumb navigation is **VPC > Subnets > subnet-0ef06d4d9deac9779**. The page title is **subnet-0ef06d4d9deac9779 / infra-project-subnet**. The left sidebar shows the **VPC dashboard** with a filter by VPC and a list of VPC resources including Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, and Peering connections.

**Details**

Property	Value
Subnet ID	subnet-0ef06d4d9deac9779
Subnet ARN	arn:aws:ec2:ap-south-1:290690313212:subnet/subnet-0ef06d4d9deac9779
State	Available
Block Public Access	Off
IPv4 CIDR	10.0.1.0/24
Availability Zone	aps1-az1 (ap-south-1a)
Available IPv4 addresses	250
IPv6 CIDR	-
Network ACL	acl-0484f620d07747137
Network border group	ap-south-1
VPC	vpc-0385396d770ec109b   infra-project-vpc
Auto-assign IPv6 address	No
Auto-assign customer-owned IPv4 address	No
Auto-assign public IPv4 address	Yes
Route table	rtb-0a15d051d6dc8ab71
Default subnet	No
Auto-assign IPv6 address	No
Customer-owned IPv4 pool	-
IPv4 CIDR reservations	-
Outpost ID	-
IPv6-only	No
Hostname type	IP name
Resource name DNS AAAA	-
Resource name DNS A record	Disabled

## Security Group

The screenshot shows the AWS Management Console interface for a security group. The breadcrumb navigation is **VPC > Security Groups > sg-06f6d2085983975e4 - infra-project-sg**. The page title is **sg-06f6d2085983975e4 - infra-project-sg**. The left sidebar shows the **VPC dashboard** with a filter by VPC and a list of VPC resources including Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, and Peering connections.

**Details**

Property	Value
Security group name	infra-project-sg
Security group ID	sg-06f6d2085983975e4
Description	Managed by Terraform
VPC ID	vpc-0385396d770ec109b
Owner	290690313212
Inbound rules count	2 Permission entries
Outbound rules count	1 Permission entry

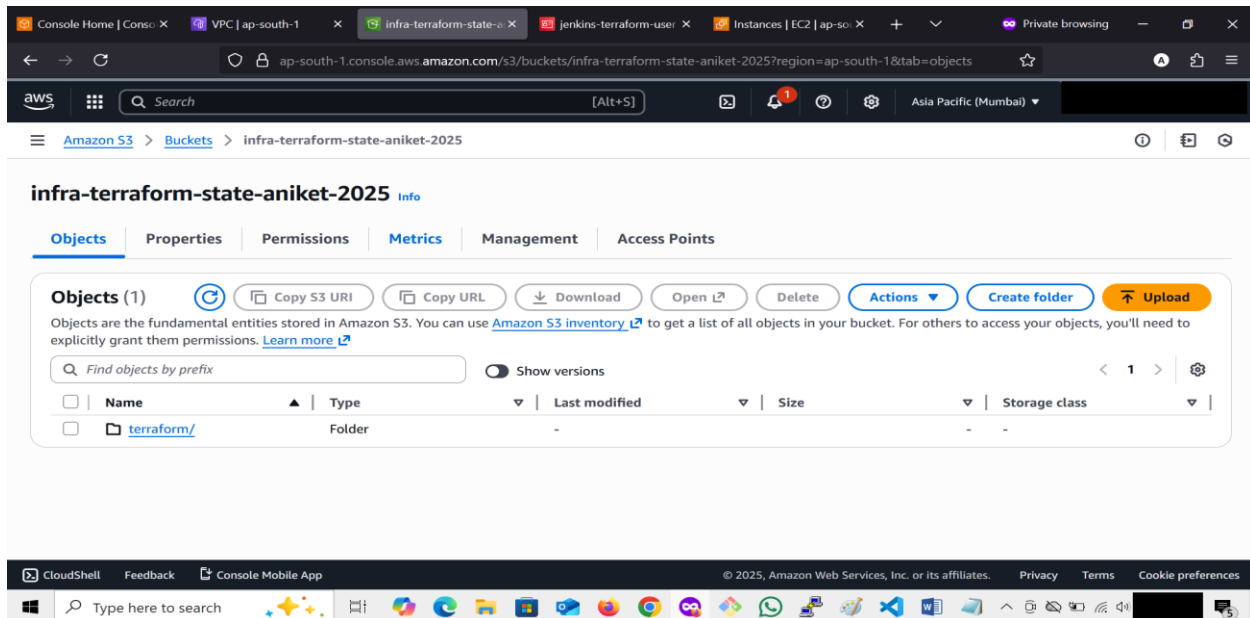
**Inbound rules (2)**

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0d35df7b969f2ce1a	IPv4	Custom TCP	TCP	8080
-	sgr-0e7bef3db8df64377	IPv4	SSH	TCP	22

S3 Bucket: infra-terraform-state-aniket-2025

Key: terraform/state.tfstate

Region: ap-south-1



this ensures consistent state even if the Jenkins instance is recreated.

## 10. CI/CD Execution Workflow

1. Developer pushes Terraform code to GitHub.
2. Jenkins pipeline is triggered.
3. Jenkins clones repo → verifies AWS creds → initializes Terraform.
4. Terraform provisions EC2, VPC, subnet, and SG on AWS.
5. Jenkins shows the build result and stores logs.
6. State file is uploaded to S3.
7. Destroy can be triggered for cleanup.

## 11. Destroy Infrastructure

Terraform destroy stage (optional) removes all created resources automatically:  
terraform destroy -auto-approve

## 12. Conclusion

This project demonstrates a complete CI/CD workflow integrating **Jenkins**, **Terraform**, **GitHub**, and **AWS**.

The system automates infrastructure deployment and management using Infrastructure as Code (IaC).

### Key Achievements:

- Jenkins automates Terraform workflows.
- Infrastructure is stored, versioned, and auditable.
- Remote state management with S3 ensures team consistency.
- Scalable foundation for multi-environment (Dev/Stage/Prod) CI/CD pipelines.