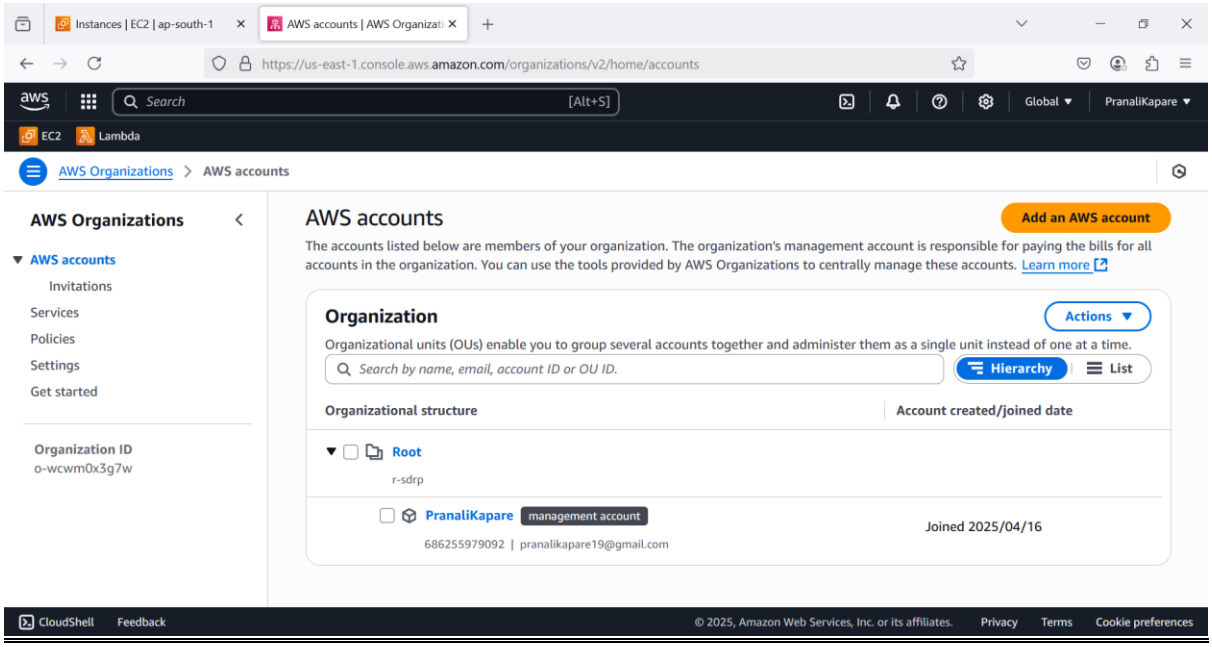
NAME: PRANALI SACHIN KAPARE

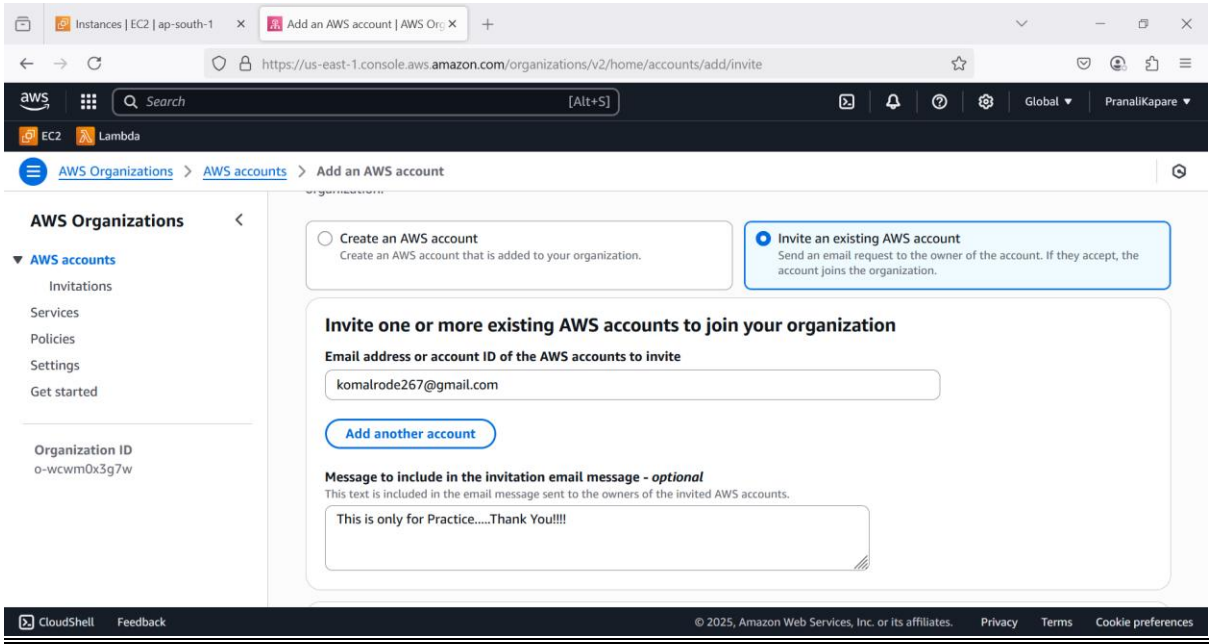
BATCH: MCA 10 JUNE

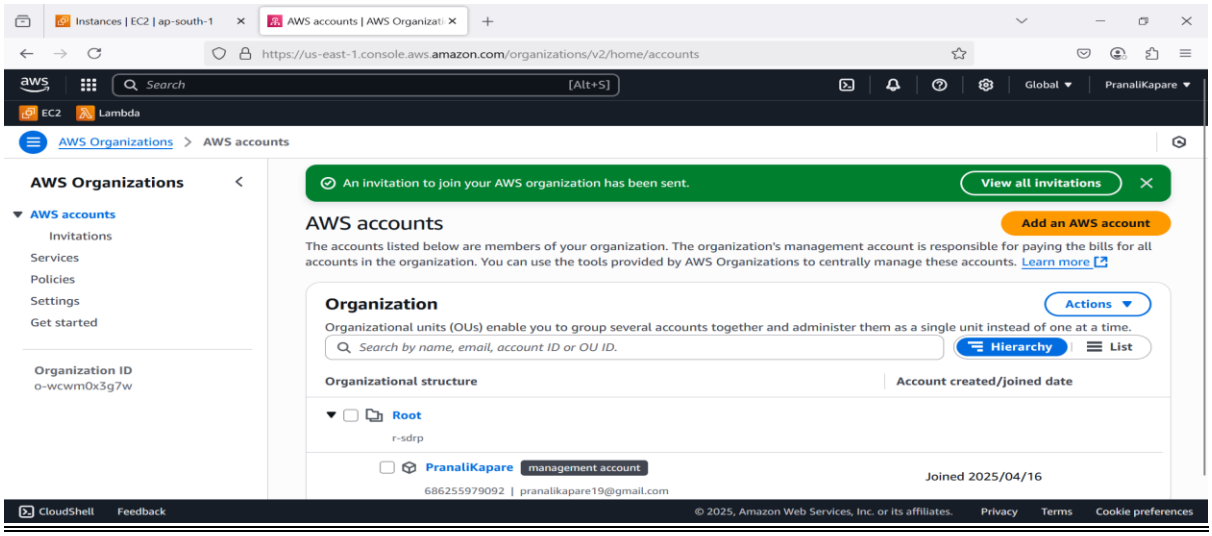
SET B1

Task 1: Set up AWS Organizations for managing multiple AWS accounts.

ANS:







Task 2: What is the purpose of use of AWS-CLI? What are the key requirements to access AWS account using AWS-CLI.

Create a working custom VPC. Launch one server in each subnet & connect private server with bastion host using CLI.

ANS:

🧰 Purpose of AWS CLI

The AWS CLI (Command Line Interface) allows you to interact with AWS services from your terminal or shell using simple commands.

✅ Key Requirements to Access AWS Account Using AWS CLI

1. **Install AWS CLI**
    - Download and install the AWS Command Line Interface tool on your system.
2. **IAM User with Programmatic Access**
    - You need an IAM user in your AWS account.
    - When creating the user, enable programmatic access (this gives you the Access Key and Secret Key).
    - The user should have the appropriate permissions or policies (like AmazonEC2FullAccess, AdministratorAccess, etc.).
3. **Access Key and Secret Access Key**
    - After creating the IAM user, you'll get:
        - AWS_ACCESS_KEY_ID

- AWS_SECRET_ACCESS_KEY

4. **Configure the CLI**

   - Run this command in your terminal:

     aws configure

   - Enter the following when prompted:

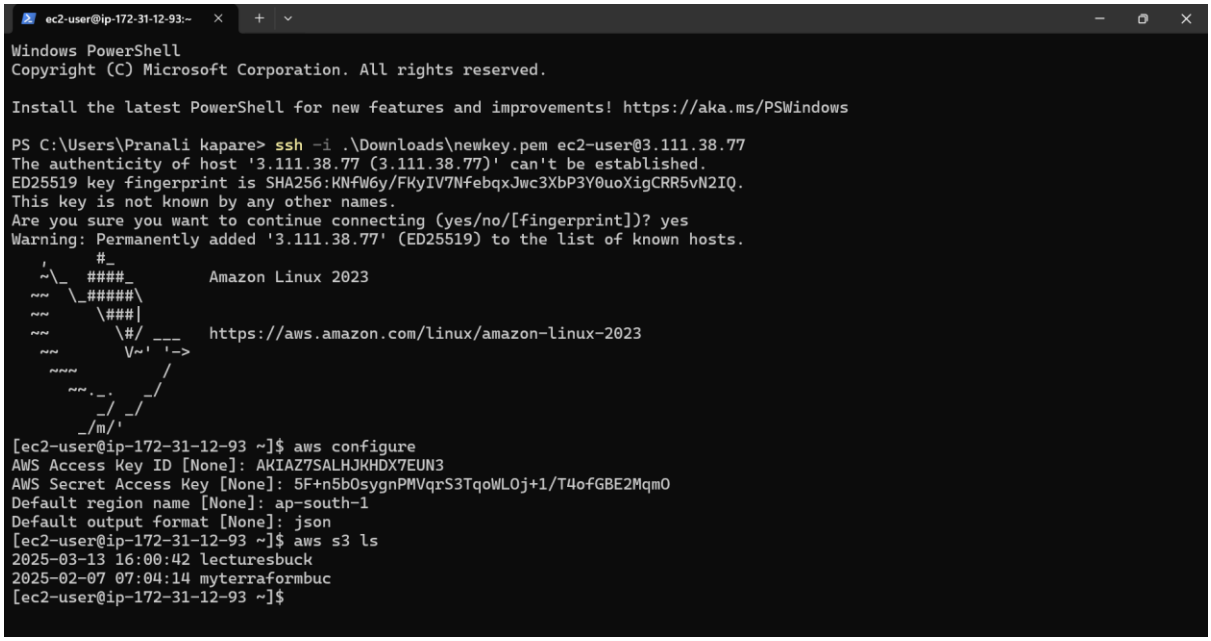
     Access Key ID

     Secret Access Key

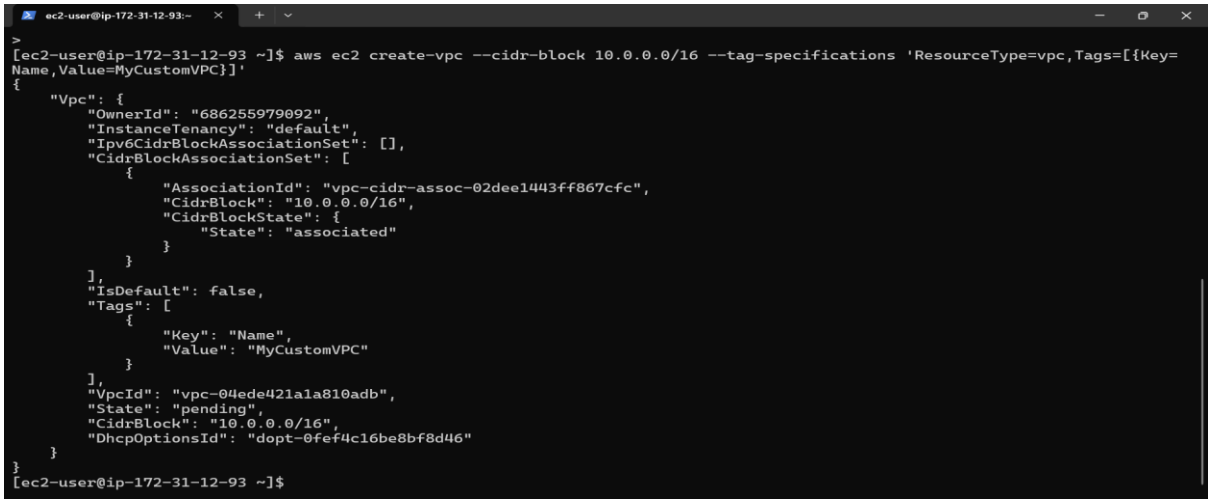
     Default Region (e.g., ap-south-1 for Mumbai)

     Default Output Format (e.g., json)

➢ AWS Configure



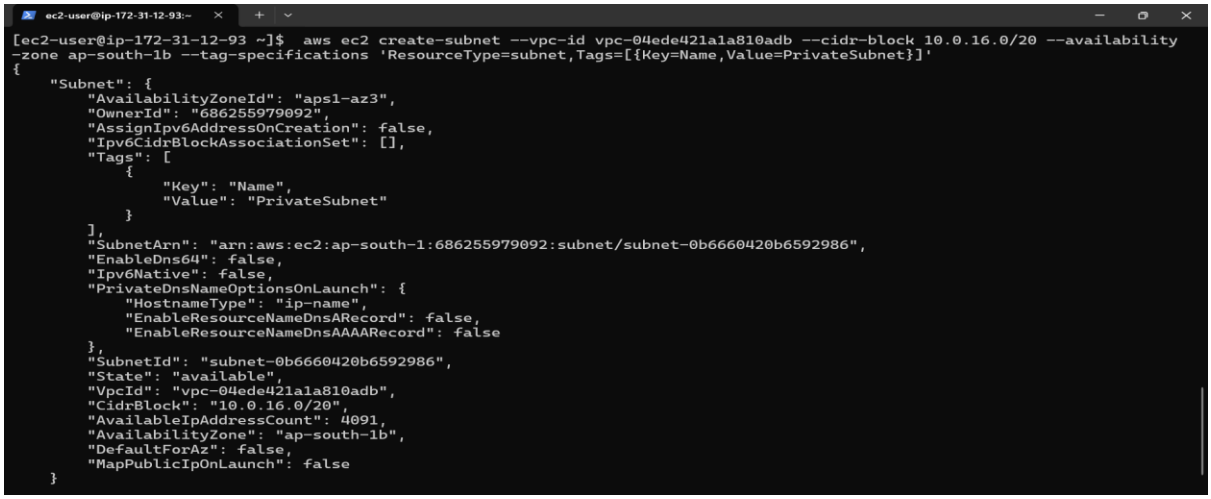
➢ Create Custom VPC



- Create Public Subnet

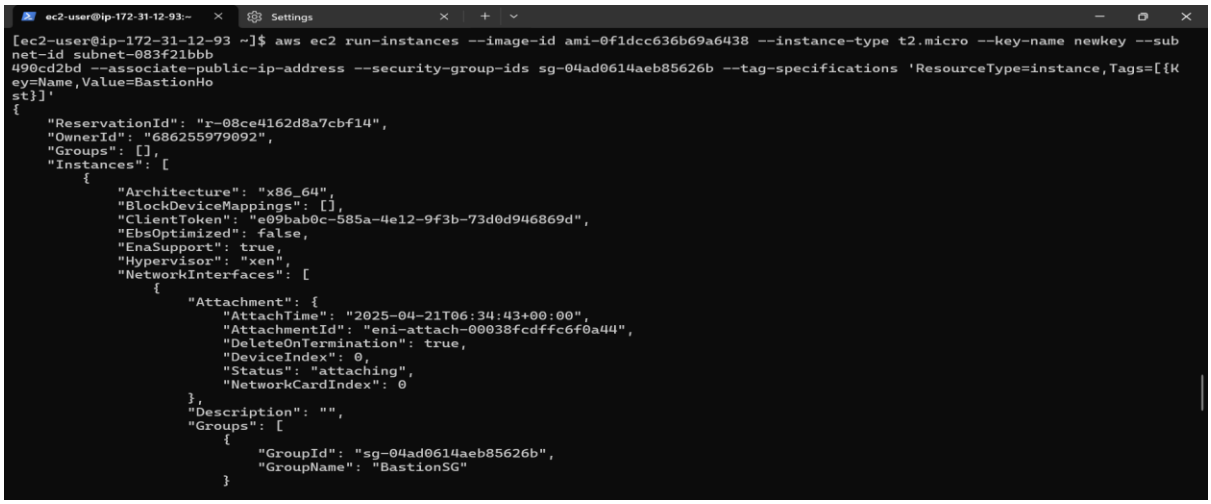
```
[ec2-user@ip-172-31-12-93 ~]$ aws ec2 create-subnet --vpc-id vpc-04ede421a1a810adb --cidr-block 10.0.1.0/20 --availability-z
one ap-south-1a --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet}]'
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az1",
        "OwnerId": "686255979092",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "Tags": [
            {
                "Key": "Name",
                "Value": "PublicSubnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:ap-south-1:686255979092:subnet/subnet-083f21bbb490cd2bd",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-083f21bbb490cd2bd",
        "State": "available",
        "VpcId": "vpc-04ede421a1a810adb",
        "CidrBlock": "10.0.0.0/20",
        "AvailableIpAddressCount": 4091,
        "AvailabilityZone": "ap-south-1a",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
```

- Create Private Subnet



```
[ec2-user@ip-172-31-12-93 ~]$  aws ec2 create-subnet --vpc-id vpc-04ede421a1a810adb --cidr-block 10.0.16.0/20 --availability
-zone ap-south-1b --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet}]'
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az3",
        "OwnerId": "686255979092",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "Tags": [
            {
                "Key": "Name",
                "Value": "PrivateSubnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:ap-south-1:686255979092:subnet/subnet-0b6660420b6592986",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-0b6660420b6592986",
        "State": "available",
        "VpcId": "vpc-04ede421a1a810adb",
        "CidrBlock": "10.0.16.0/20",
        "AvailableIpAddressCount": 4091,
        "AvailabilityZone": "ap-south-1b",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
```

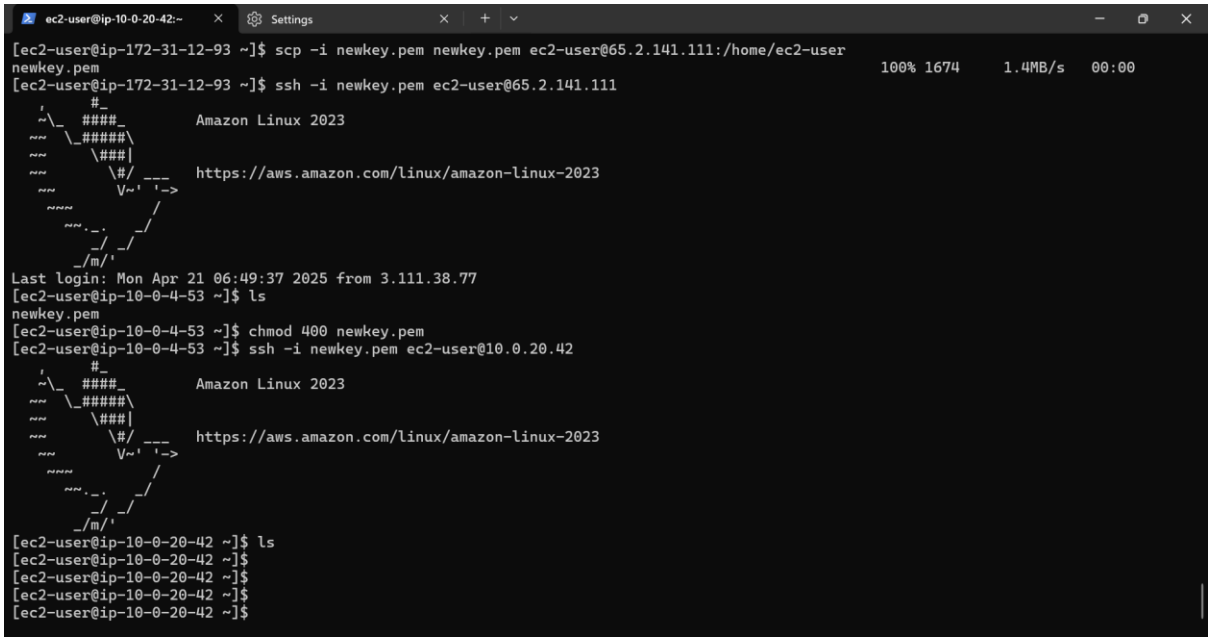
- Create BastionHost



```
[ec2-user@ip-172-31-12-93 ~]$ aws ec2 run-instances --image-id ami-0f1dcc636b69a6438 --instance-type t2.micro --key-name newkey --sub
net-id subnet-083f21bbb
490cd2bd --associate-public-ip-address --security-group-ids sg-04ad0614aeb85626b --tag-specifications 'ResourceType=instance,Tags=[{K
ey=Name,Value=BastionHo
st}]'
{
    "ReservationId": "r-08ce4162d8a7cbf14",
    "OwnerId": "686255979092",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "e09bab0c-585a-4e12-9f3b-73d0d946869d",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-04-21T06:34:43+00:00",
                        "AttachmentId": "eni-attach-00038fcdffc6f0a44",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-04ad0614aeb85626b",
                            "GroupName": "BastionSG"
                        }
```

- Create Private Server

- Connect Private Server with Bastion Host



Task 3: Write a short note on S3. What is difference between Static website and Dynamic website?

Create a static website using S3 with minimum 3 pages having images and videos.

ANS:

✅ Amazon S3 (Simple Storage Service)

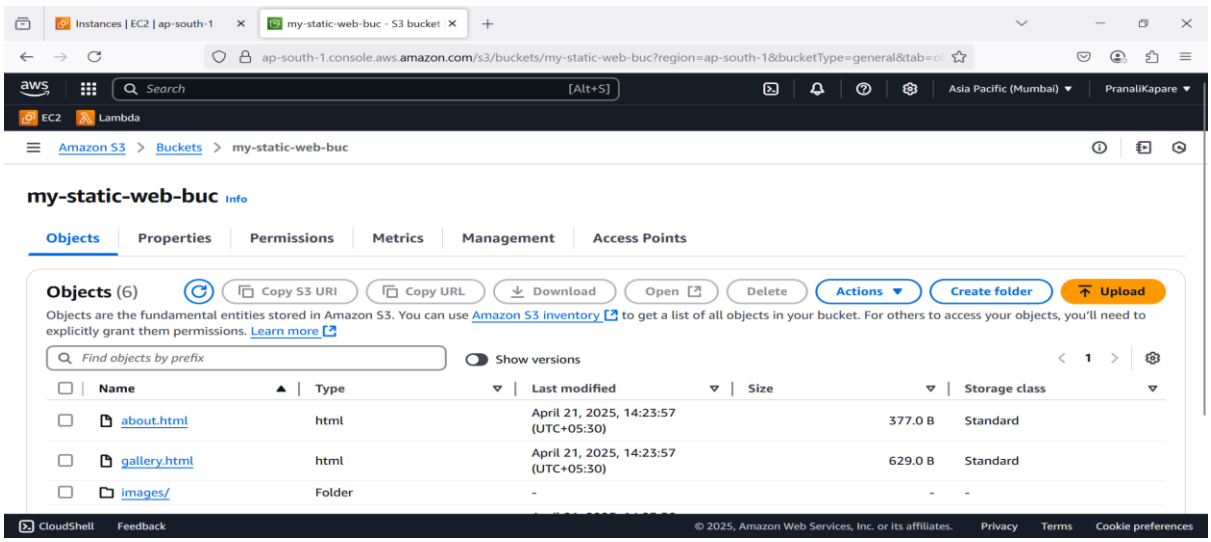
Amazon S3 is a scalable object storage service provided by AWS. It is used to store and retrieve any amount of data at any time, from anywhere on the web.

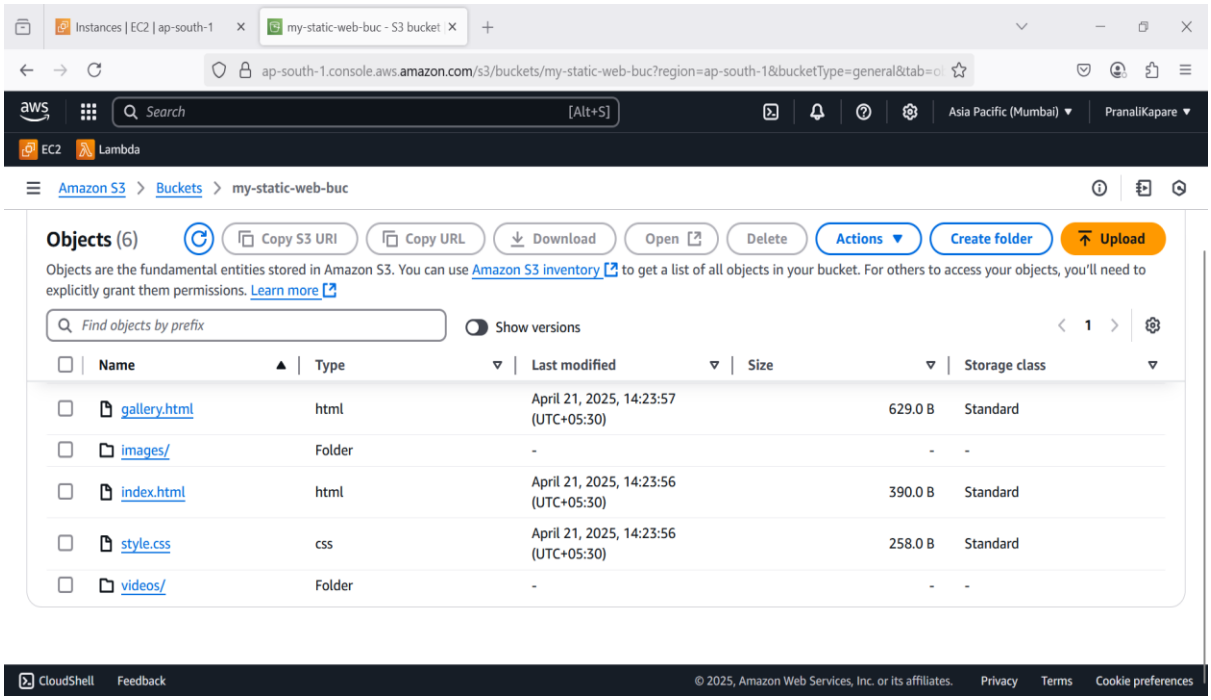
🔹 Key Features:

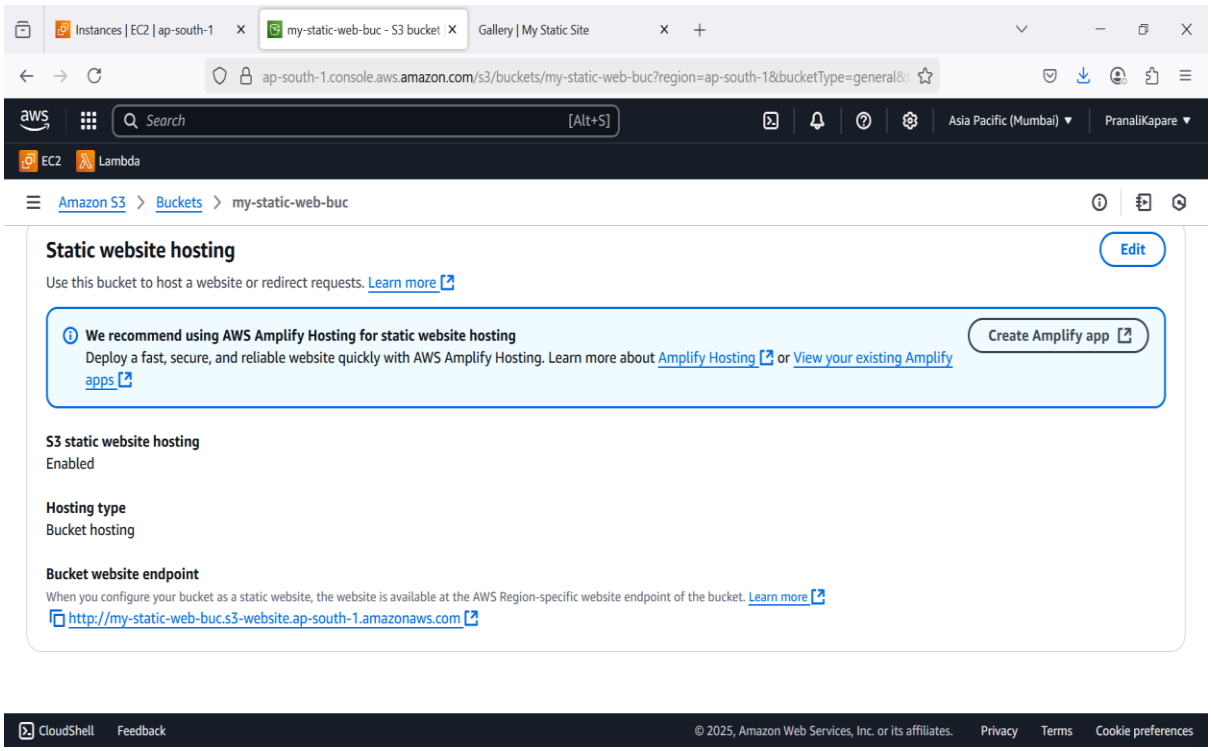
- Stores objects (files) in buckets
- Supports versioning, encryption, and lifecycle policies
- Common use cases: backups, website hosting, media storage, big data analytics
- Offers high durability (99.999999999%) and availability

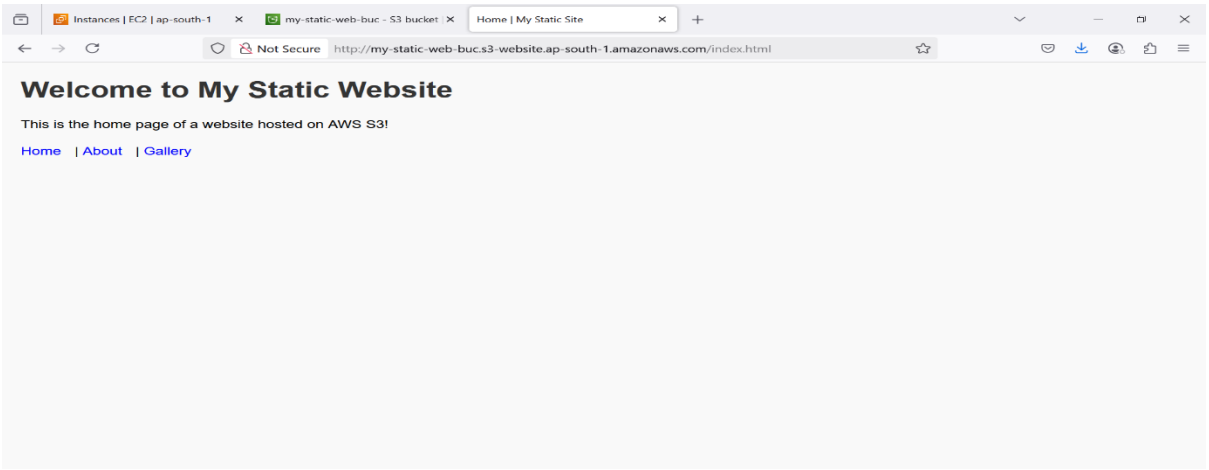
✅ Difference Between Static Website and Dynamic Website

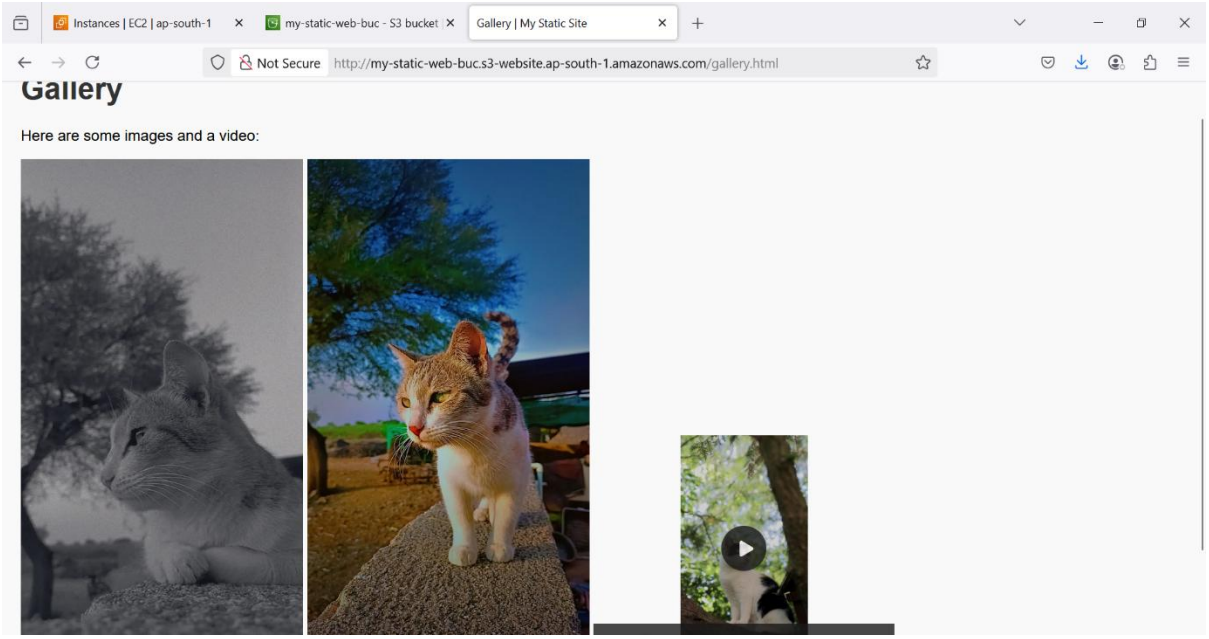
Feature	Static Website	Dynamic Website
Content	Same for all users	Changes based on user input or server logic
Technologies	HTML, CSS, JavaScript	PHP, Python, Node.js + Database (MySQL, MongoDB, etc.)
Hosting	Simple hosting (e.g., AWS S3)	Needs server (e.g., EC2, web hosting with backend)
Interactivity	Limited or none	High — based on user interactions
Use Cases	Portfolios, blogs, landing pages	E-commerce sites, dashboards, login-based applications
Speed	Very fast, no server processing needed	Slower, depends on server performance




Instances | EC2 | ap-south-1
my-static-web-buc - S3 bucket
ap-south-1.console.aws.amazon.com/s3/buckets/my-static-web-buc?region=ap-south-1&bucketType=general&tab=o
Search [Alt+S]
Asia Pacific (Mumbai)
PranaliKapare
EC2
Lambda
Amazon S3 > Buckets > my-static-web-buc
Objects (6)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
1
Name
Type
Last modified
Size
Storage class
gallery.html
html
April 21, 2025, 14:23:57 (UTC+05:30)
629.0 B
Standard
images/
Folder
-
-
-
index.html
html
April 21, 2025, 14:23:56 (UTC+05:30)
390.0 B
Standard
style.css
css
April 21, 2025, 14:23:56 (UTC+05:30)
258.0 B
Standard
videos/
Folder
-
-
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Instances | EC2 | ap-south-1
my-static-web-buc - S3 bucket
Gallery | My Static Site
ap-south-1.console.aws.amazon.com/s3/buckets/my-static-web-buc?region=ap-south-1&bucketType=general&t
Search [Alt+S]
Asia Pacific (Mumbai)
PranaliKapare
EC2
Lambda
Amazon S3 > Buckets > my-static-web-buc
Static website hosting
Edit
Use this bucket to host a website or redirect requests. Learn more
We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps
Create Amplify app
S3 static website hosting
Enabled
Hosting type
Bucket hosting
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://my-static-web-buc.s3-website.ap-south-1.amazonaws.com
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences





Task 4: What is Instance Refresh? Demonstrate the use of Instance Refresh**.**

ANS:

✅ Instance Refresh

Instance Refresh is a feature of **Amazon EC2 Auto Scaling Groups (ASG)** that enables you to **safely update the group's instances** without replacing or recreating the ASG itself.

It helps apply changes such as:

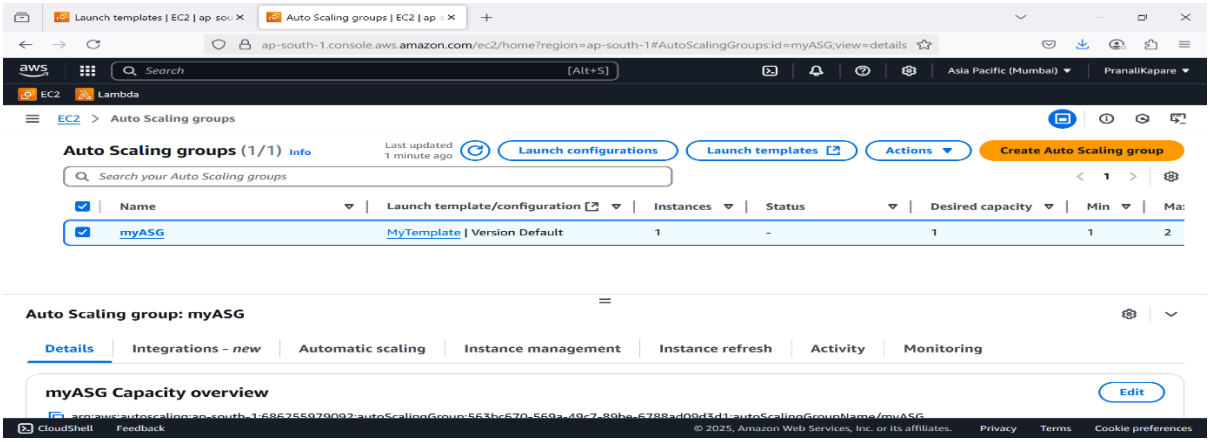
- New Amazon Machine Image (AMI)
- Updated launch template or launch configuration
- Configuration changes (e.g., instance type, user data)

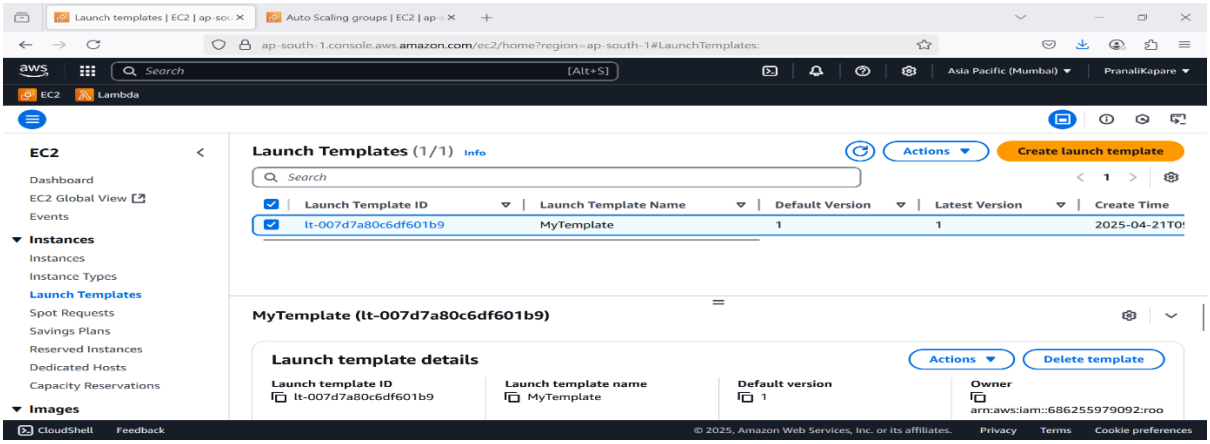
The refresh happens in a **rolling update manner**, ensuring **high availability** and **minimal disruption**.

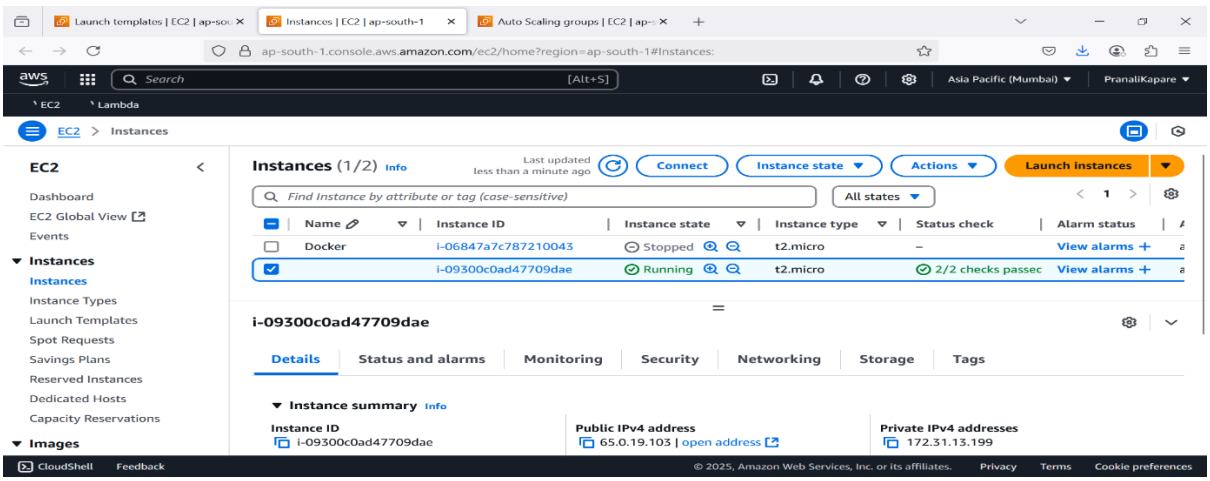
🛠️ Demonstration: Using Instance Refresh via AWS Console

Make sure you already have:

- An **Auto Scaling Group (ASG)** created
- A **Launch Template** attached to it
- At least one running EC2 instance managed by the ASG

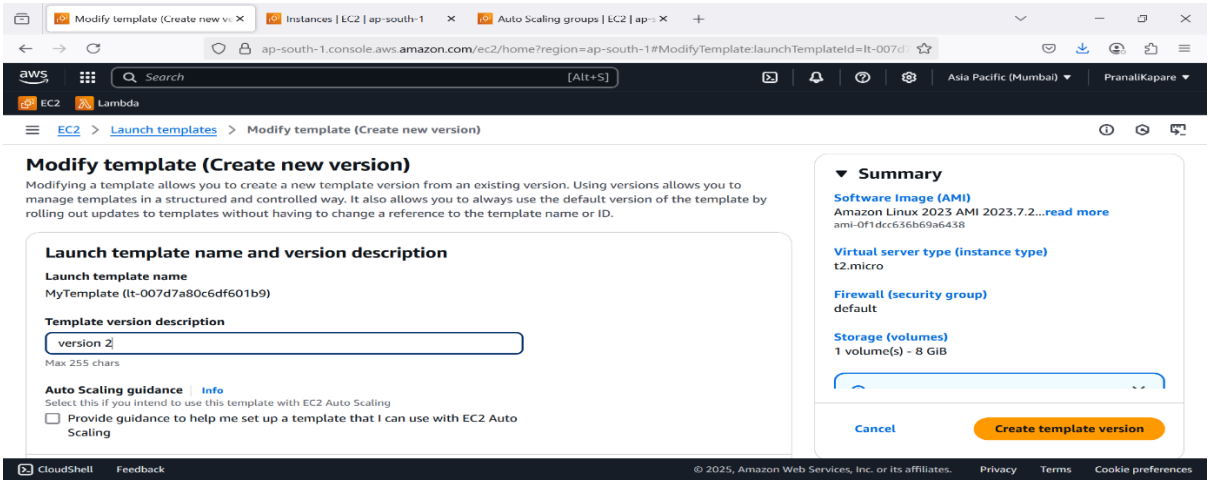


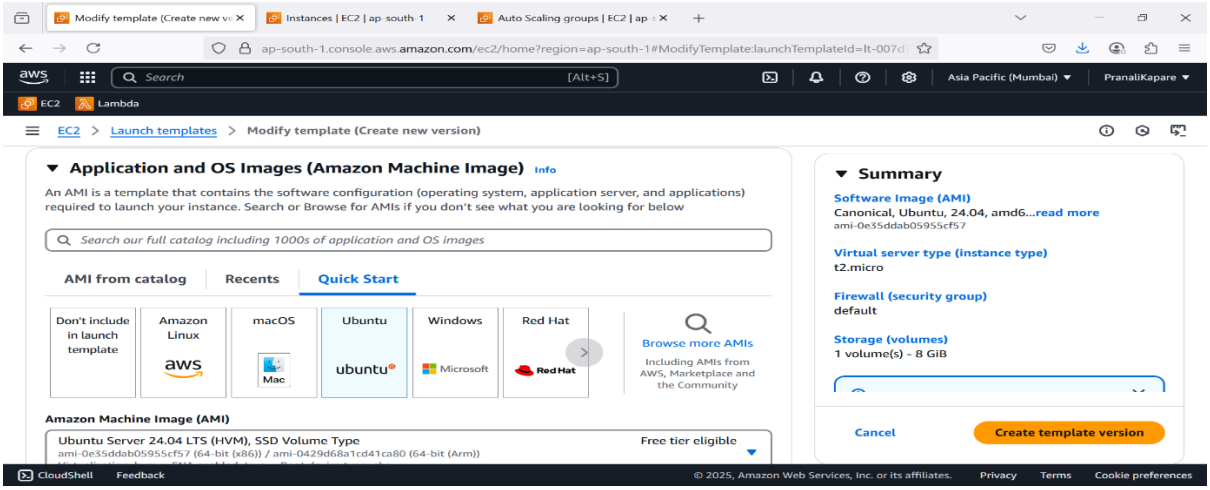




✅ Step 1: Update Launch Template

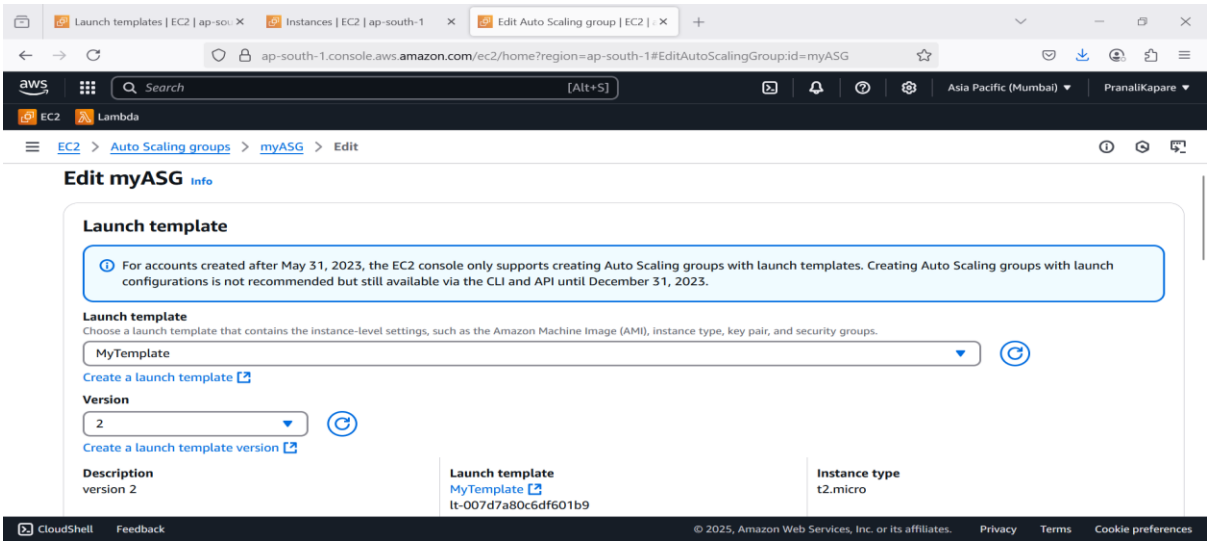
1. Go to **EC2 Dashboard → Launch Templates**
2. Select your existing Launch Template
3. Click on **Actions → Create new version**
4. Update configuration (e.g., use a new AMI)
5. Click **Create launch template version**





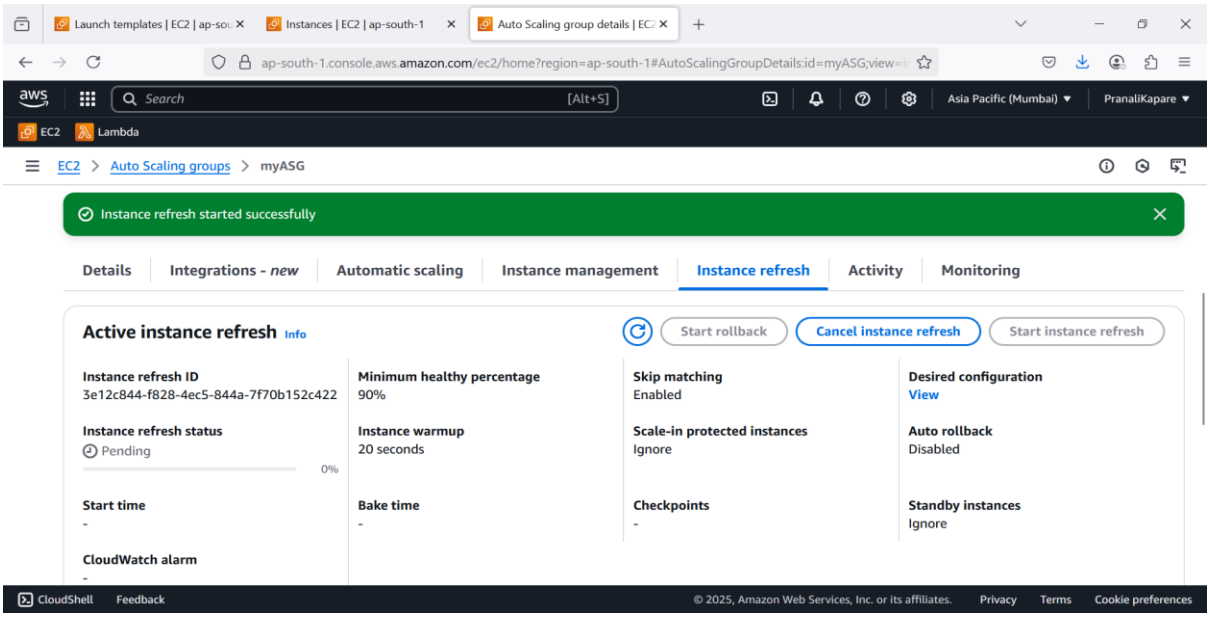
✅ Step 2: Attach Updated Launch Template to ASG

1. Go to **EC2 Dashboard → Auto Scaling Groups**
2. Click on your Auto Scaling Group
3. Under the **Details** tab, click **Edit**
4. Select the new **Launch Template version** (e.g., Version 2)
5. Click **Update**

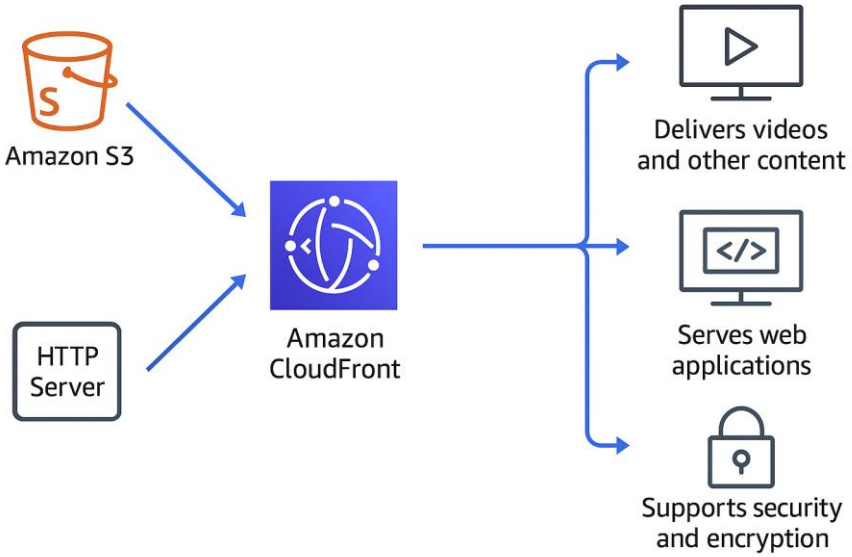


✅ Step 3: Start Instance Refresh

1. Inside your Auto Scaling Group page
2. Navigate to the **Instance Refresh** tab
3. Click **Start instance refresh**
4. You can configure:
    - **Minimum healthy percentage** (e.g., 90%)
    - **Instance warm-up time** (e.g., 20 seconds)
5. Click **Start**



Task 5: Create a architectural diagram of uses of cloud front.



SET D

1. **How to install Prometheus and Grafana on you cluster using Terraform and helm.**

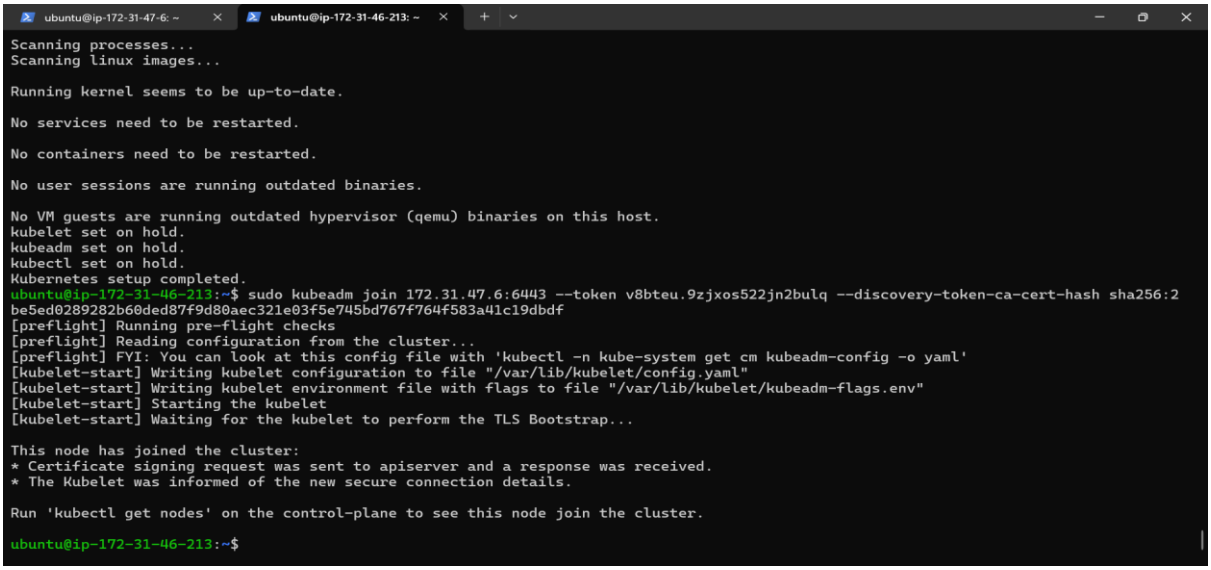
   ANS:

   - Install Kubernetes

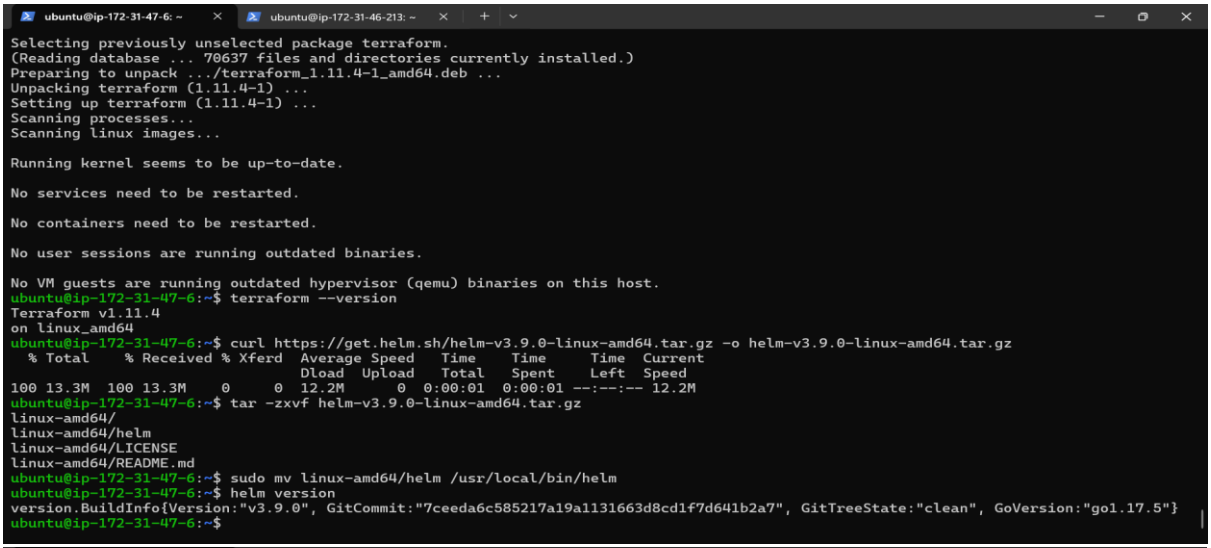
     On Master Node:

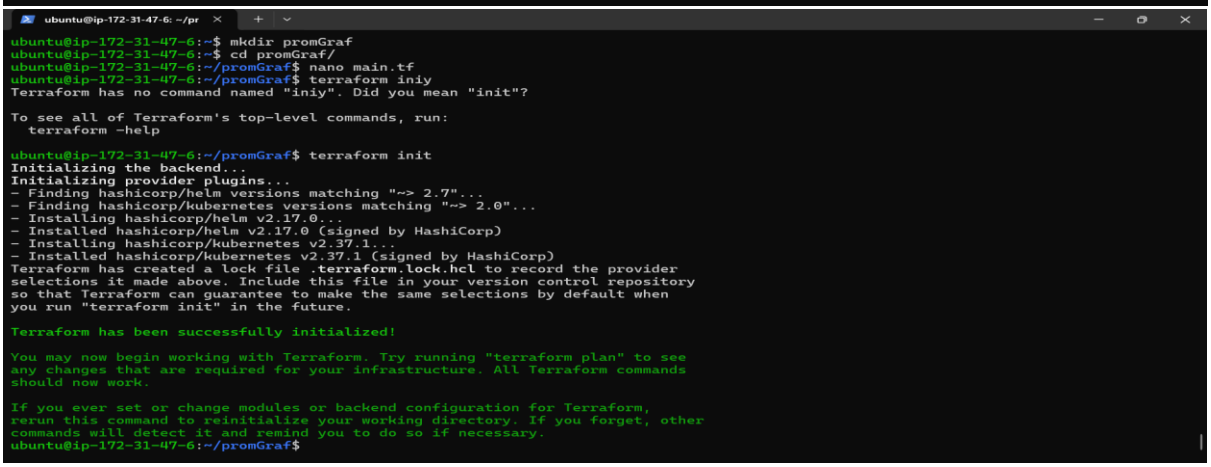
```
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
Generating join command for worker nodes...
kubeadm join 172.31.47.6:6443 --token v8bteu.9zjxos522jn2bulq --discovery-token-ca-cert-hash sha256:2be5ed0289282b60ded87f9d80aec321e
03f5e745bd767f764f583a41c19dbdf
Kubernetes Master Node setup complete.
ubuntu@ip-172-31-47-6:~$ kubectl get nodes
NAME              STATUS     ROLES           AGE     VERSION
ip-172-31-46-213  NotReady   <none>          15s     v1.29.15
ip-172-31-47-6    Ready      control-plane   3m18s   v1.29.15
ubuntu@ip-172-31-47-6:~$ kubectl get nodes
NAME              STATUS     ROLES           AGE     VERSION
ip-172-31-46-213  NotReady   <none>          32s     v1.29.15
ip-172-31-47-6    Ready      control-plane   3m35s   v1.29.15
ubuntu@ip-172-31-47-6:~$ kubectl get nodes
NAME              STATUS   ROLES           AGE     VERSION
ip-172-31-46-213  Ready    <none>          3m22s   v1.29.15
ip-172-31-47-6    Ready    control-plane   6m25s   v1.29.15
ubuntu@ip-172-31-47-6:~$
```

On Worker Node:



- Terraform and Helm Installation:



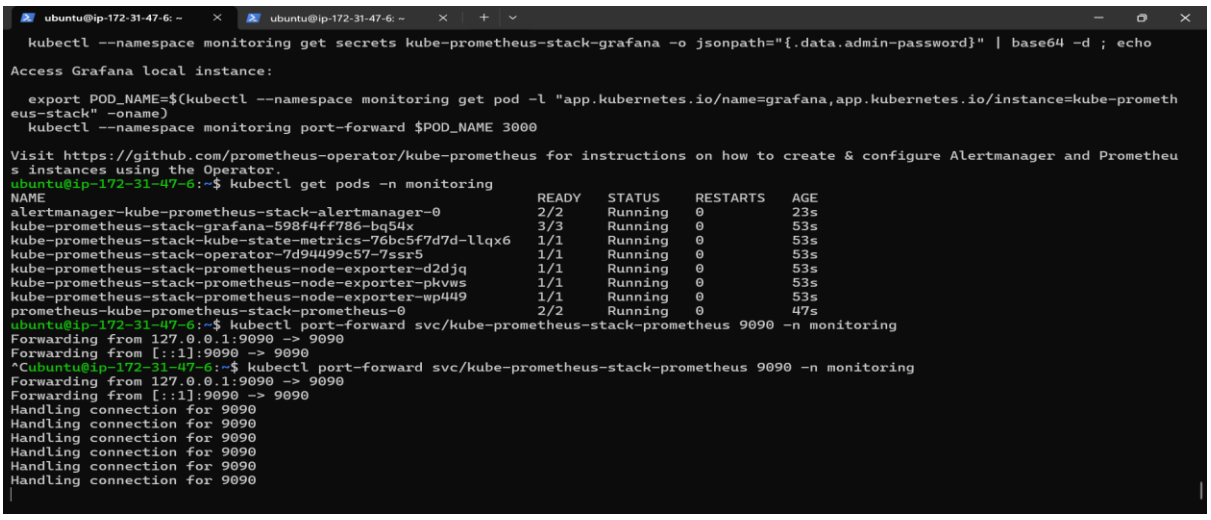


Main.tf

```
    terraform {
  required_providers {
    kubernetes = {
      source  = "hashicorp/kubernetes"
      version = "~> 2.0"
    }
    helm = {
      source  = "hashicorp/helm"
      version = "~> 2.7"
    }
  }
}

provider "kubernetes" {
  config_path = "~/.kube/config"
}

provider "helm" {
  kubernetes {
    config_path = "~/.kube/config"
  }
}

resource "helm_repository" "prometheus" {
  name = "prometheus-community"
  url  = "https://prometheus-community.github.io/helm-charts"
}
resource "helm_repository" "grafana" {
  name = "grafana"
  url  = "https://grafana.github.io/helm-charts"
}

resource "kubernetes_namespace" "monitoring" {
  metadata {
    name = "monitoring"
  }
}
```

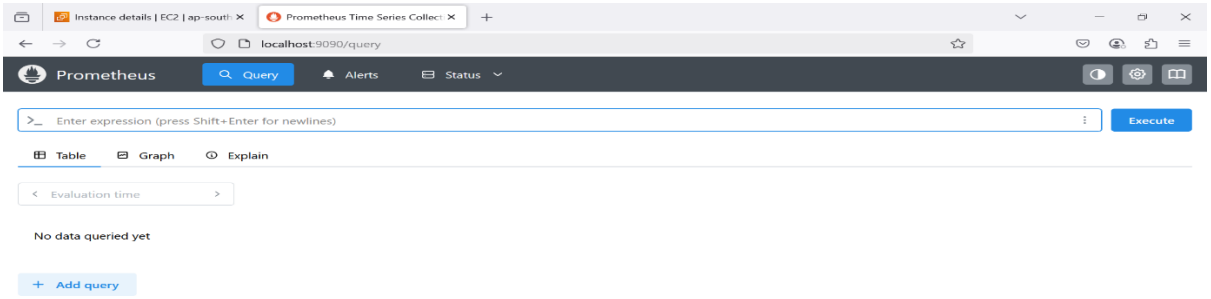
```
resource "helm_release" "prometheus_stack" {
  name             = "kube-prometheus-stack"
  repository       = helm_repository.prometheus.url
  chart            = "kube-prometheus-stack"
  namespace        = kubernetes_namespace.monitoring.metadata[0].name
  create_namespace = false

  values = [
    file("prometheus-values.yaml")  # optional, remove if you don't have this file
  ]

  depends_on = [kubernetes_namespace.monitoring]
}
resource "helm_release" "grafana" {
  name             = "grafana"
  repository       = helm_repository.grafana.url
  chart            = "grafana"
  namespace        = kubernetes_namespace.monitoring.metadata[0].name
  create_namespace = false

  values = [
    file("grafana-values.yaml")
  ]

  depends_on = [kubernetes_namespace.monitoring]
}
```

- Prometheus Port Forwarding



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Pranali kapare> ssh -i .\Downloads\newkey.pem -L 9090:localhost:9090 ubuntu@52.66.204.141
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri May 23 09:51:01 UTC 2025

  System load:  0.7               Processes:             168
  Usage of /:   30.8% of 18.33GB  Users logged in:       1
  Memory usage: 30%               IPv4 address for enX0: 172.31.47.6
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

55 updates can be applied immediately.
12 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Fri May 23 09:20:58 2025 from 152.58.30.178
ubuntu@ip-172-31-47-6:~$
```

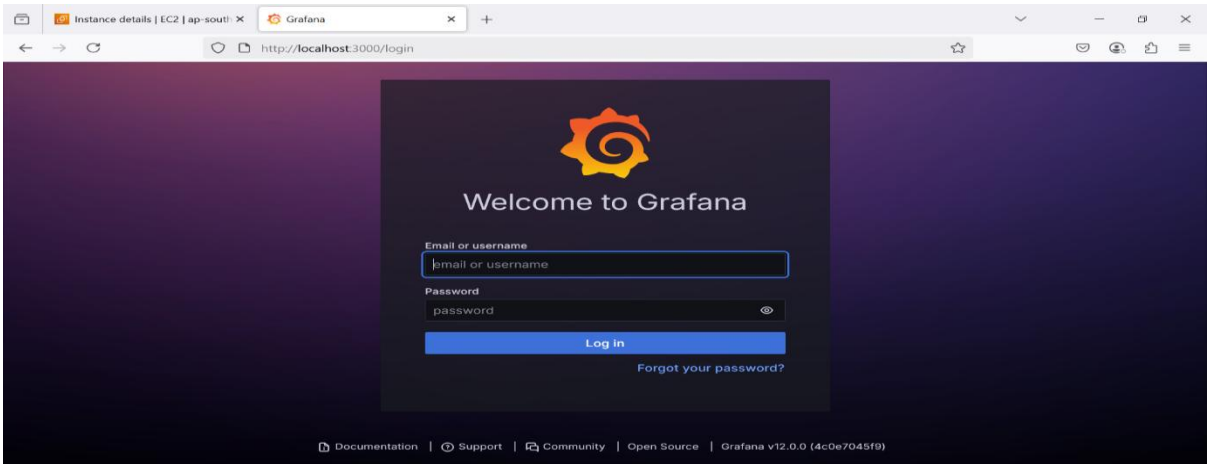
Access Prometheus UI on Browser



- ## Grafana Port Forwarding

```
  name: kube-prometheus-stack-grafana
  namespace: monitoring
  resourceVersion: "282075"
  uid: 5601abfb-2391-4231-98bb-f1b8c48414ed
spec:
  clusterIP: 10.100.94.51
  clusterIPs:
  - 10.100.94.51
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - name: http-web
    port: 80
    protocol: TCP
    targetPort: 3000
  selector:
    app.kubernetes.io/instance: kube-prometheus-stack
    app.kubernetes.io/name: grafana
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
ubuntu@ip-172-31-47-6:~$ kubectl get pods -n monitoring | grep grafana
kube-prometheus-stack-grafana-598f4ff786-2f74g              3/3     Running   0          17m
ubuntu@ip-172-31-47-6:~$ kubectl port-forward kube-prometheus-stack-grafana-598f4ff786-2f74g 3000:3000 -n monitoring
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
```

Access Grafana UI on Browser

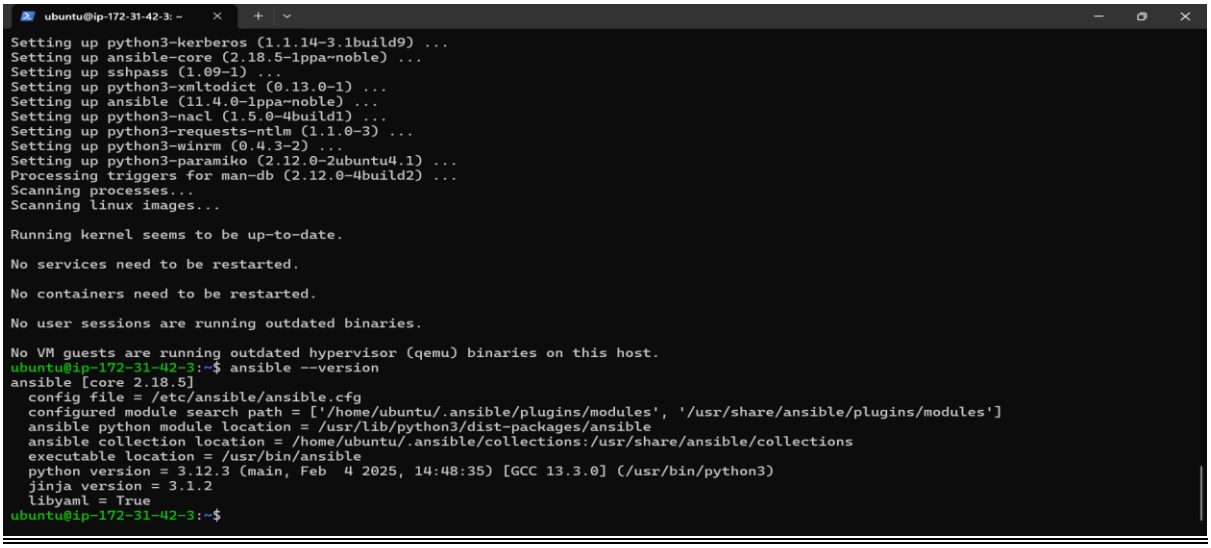


---

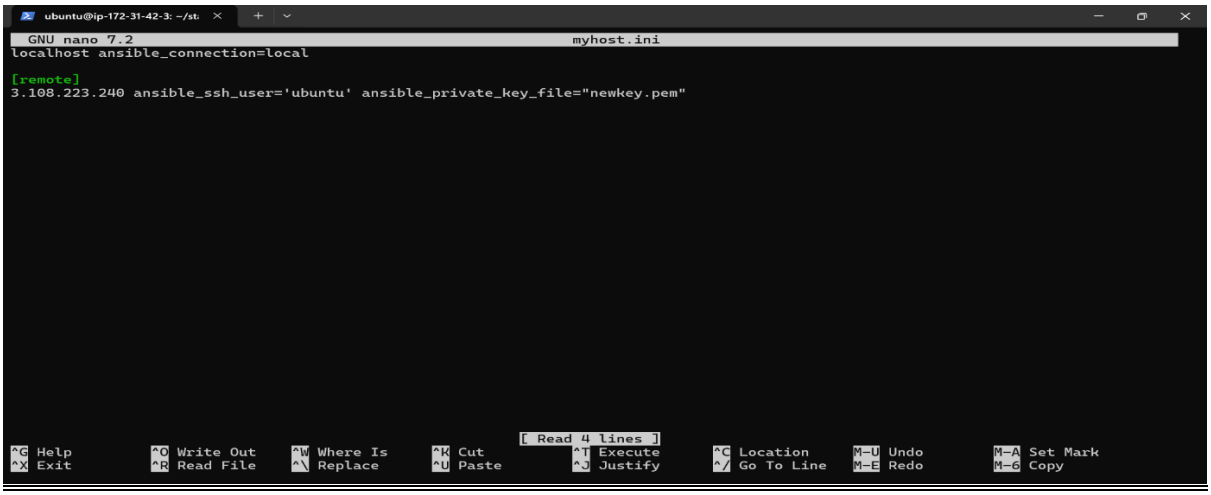
2. **Using Ansible to deploy a static web application onto a remote server.**

   ANS:

- Install Ansible

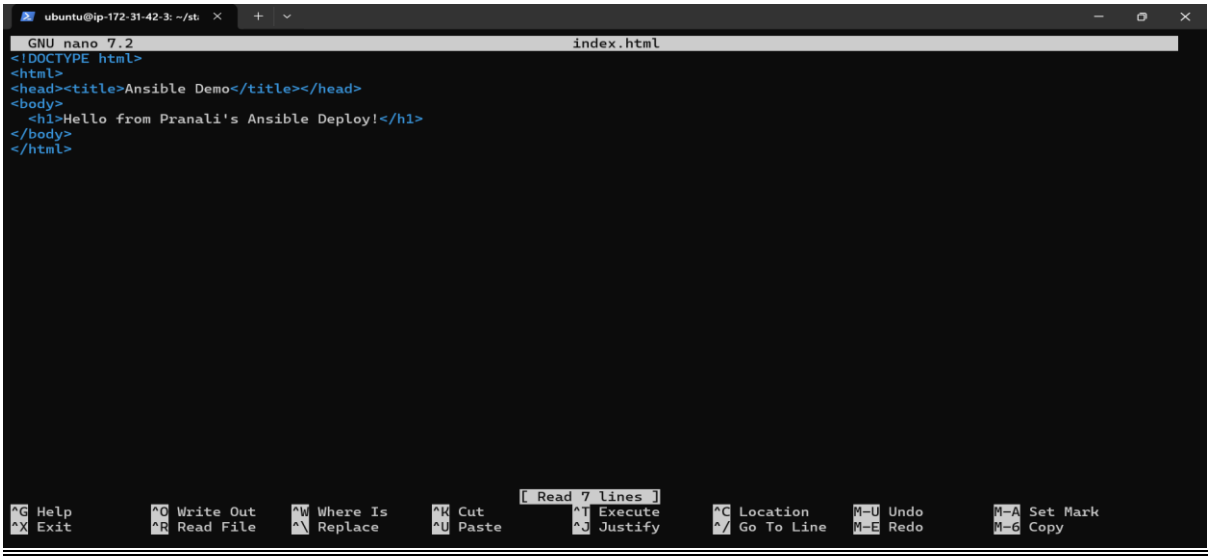


- Myhost.ini

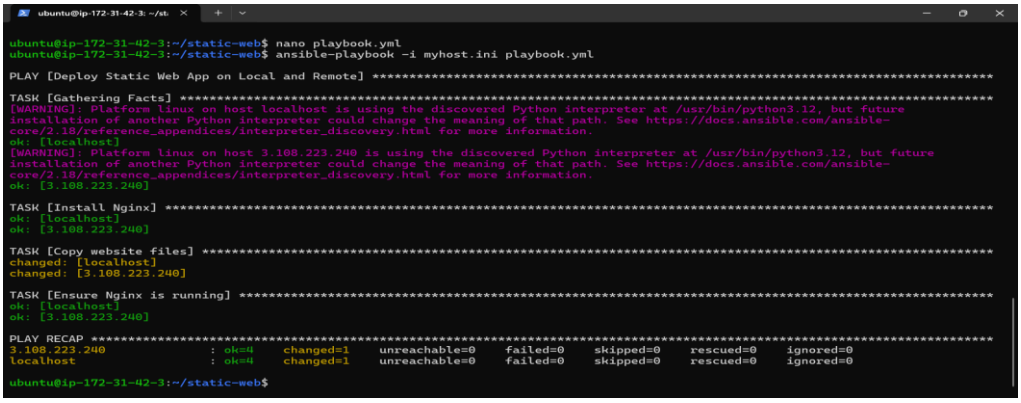


- Playbook.yml

- Index.html



- Execute the Ansible Playbook



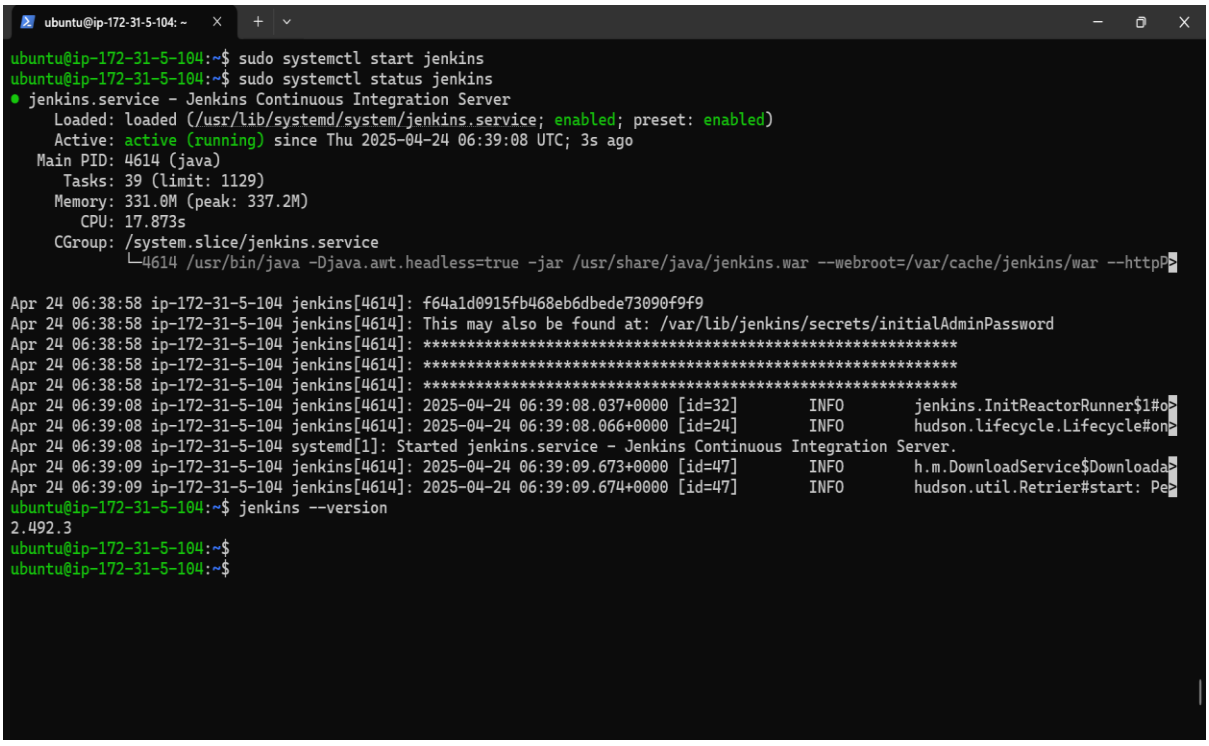
➢ Output on Browser

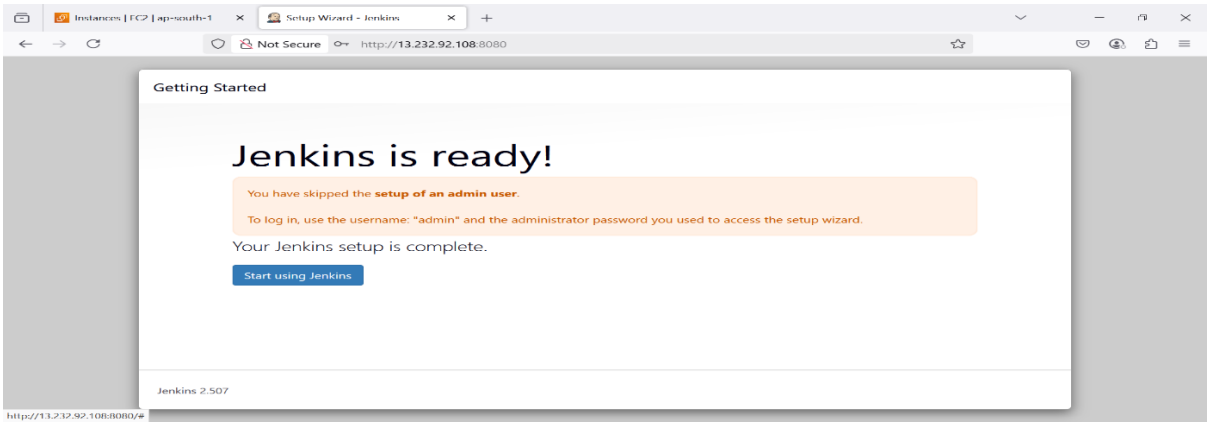
---

3. Jenkins

a. Install Jenkins from scratch on an ec2 instance with default plungins.

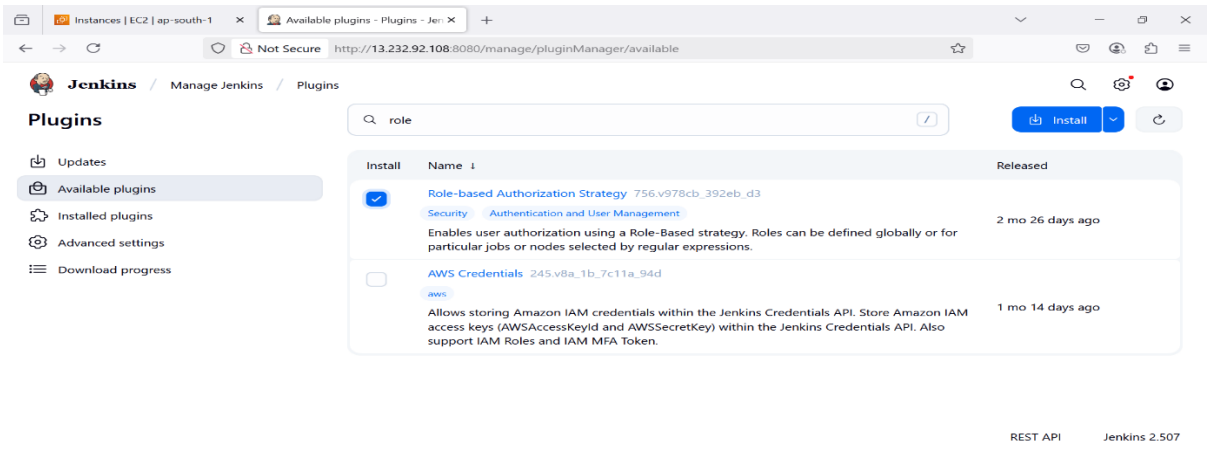
➢ Install Jenkins



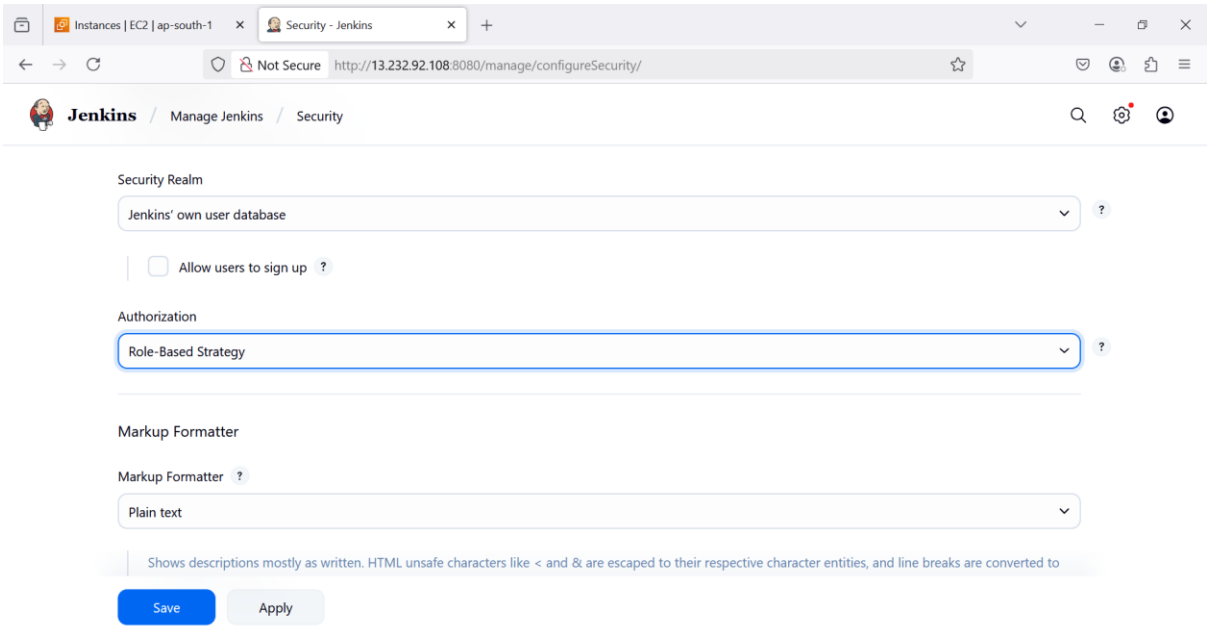


b. Create 3 users in Alice/Bob/Mary with different roles.

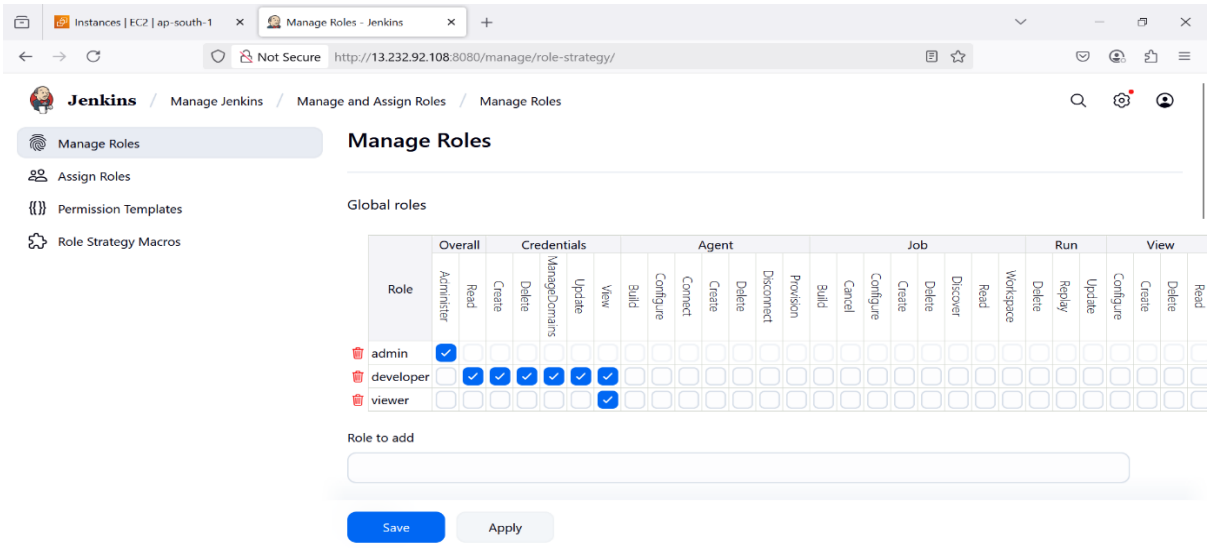
➢ Install Plugin



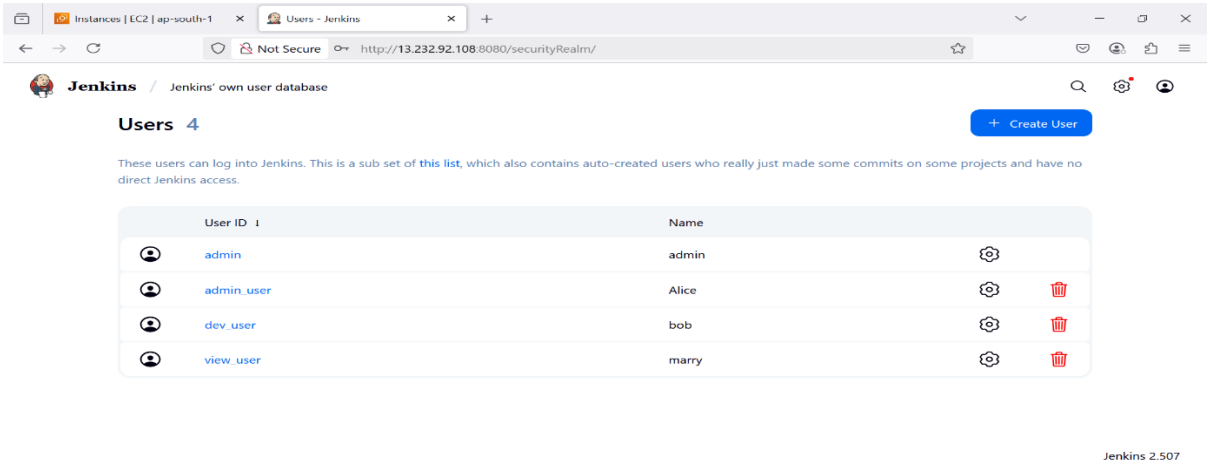
➢ Set Authorization



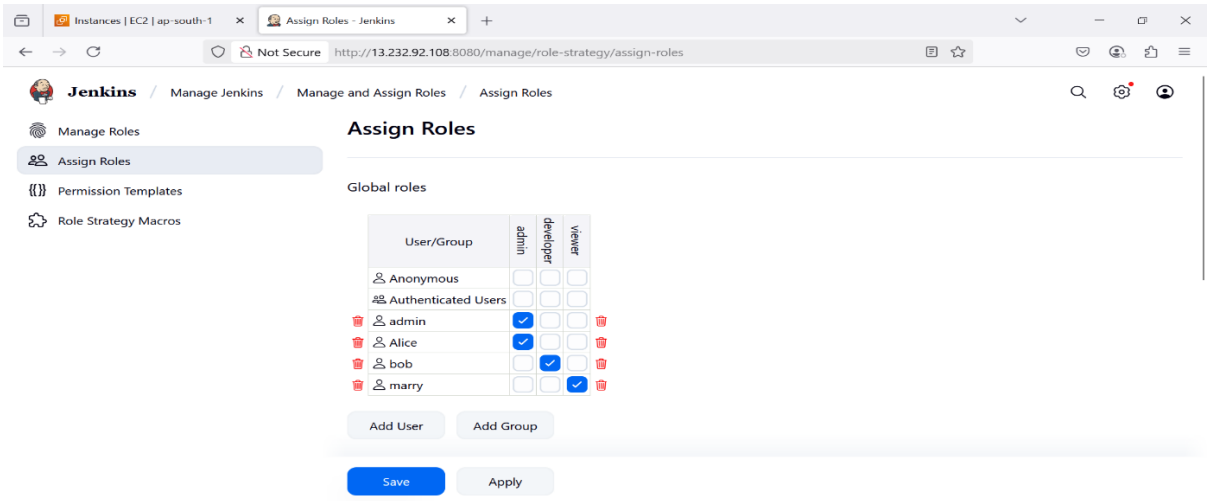
➤ Manage Roles



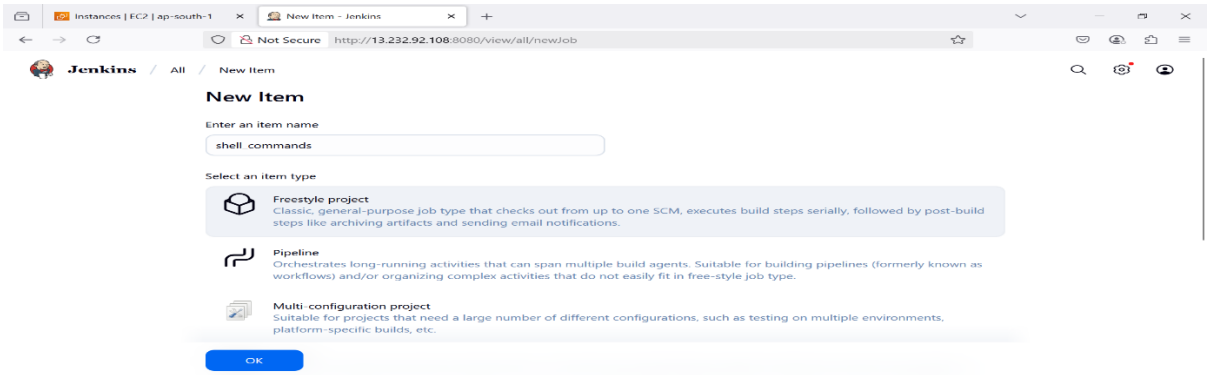
➤ Create Users

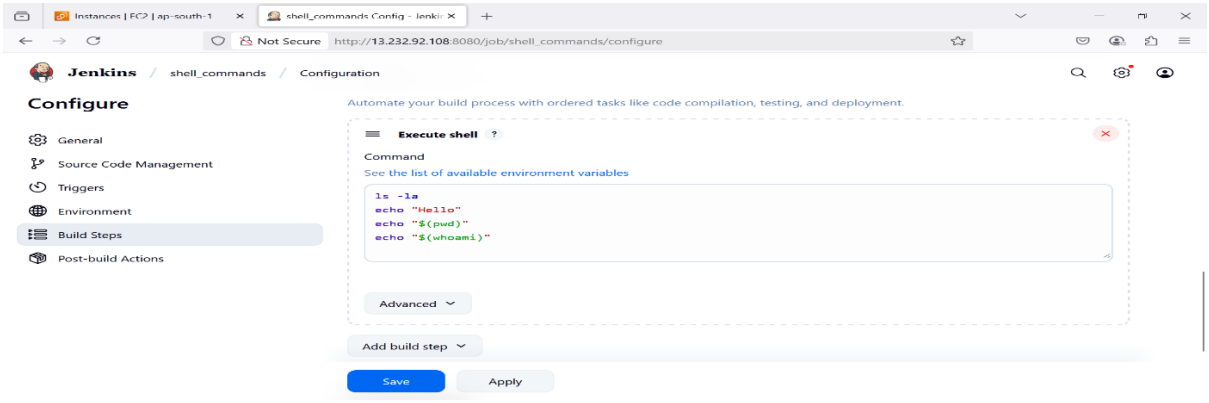


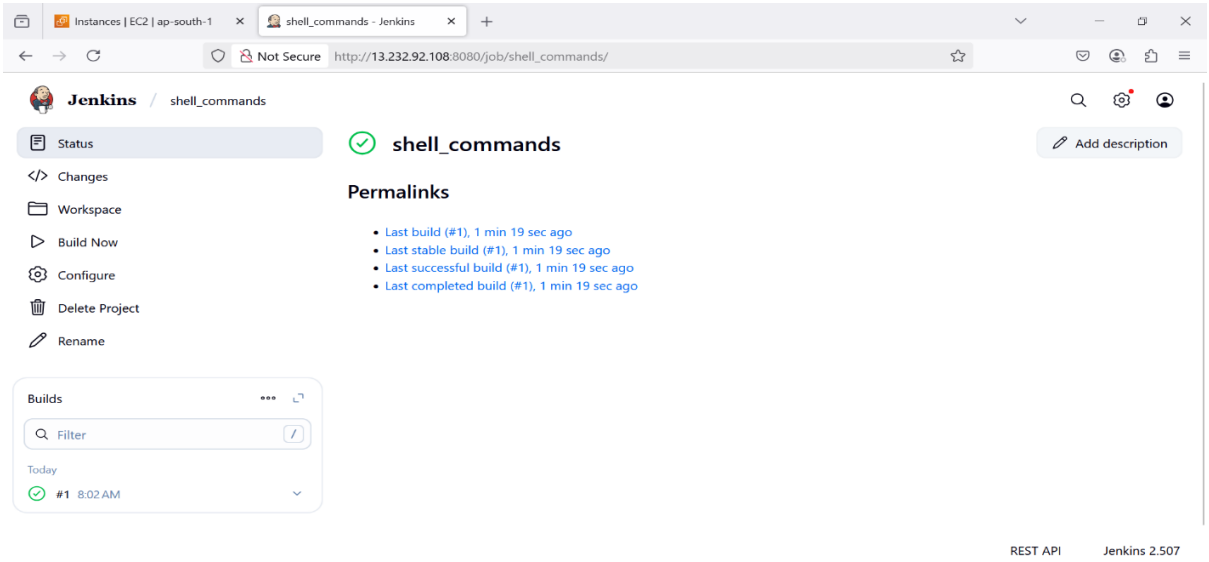
➤ Assign Roles to Users

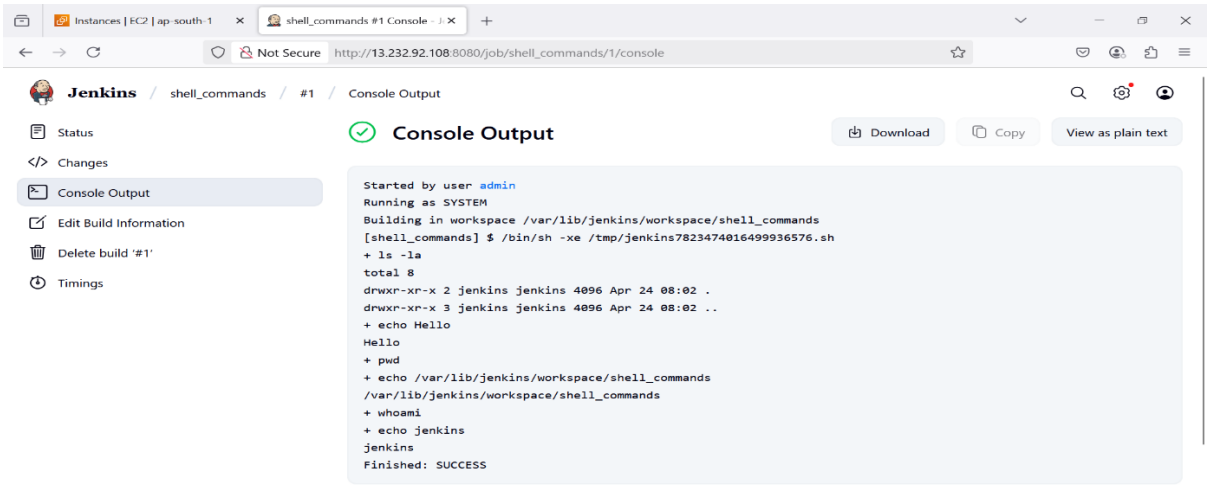


c. create free style project and execute various Linux commands in Build > Execute shell

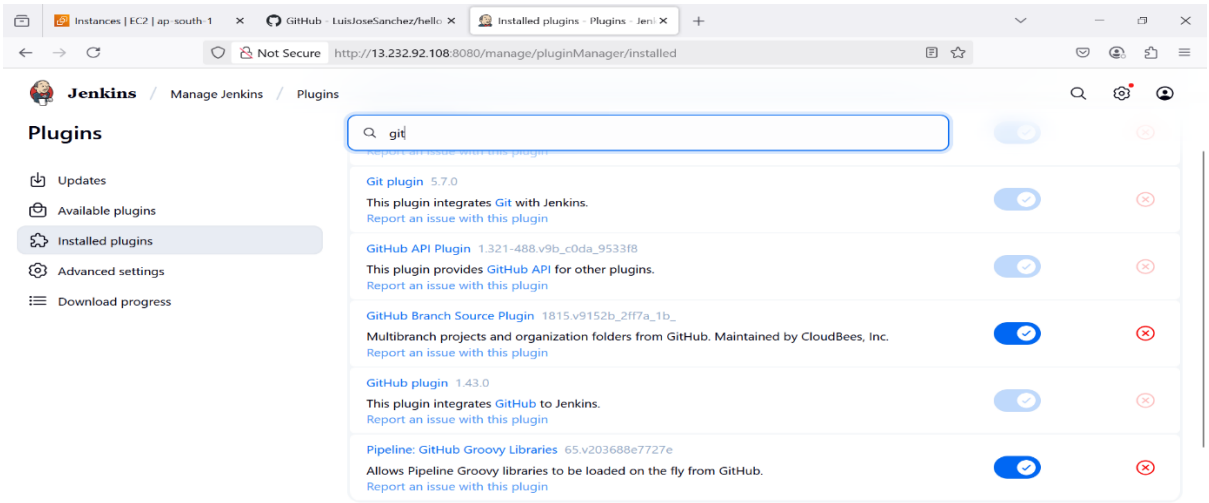


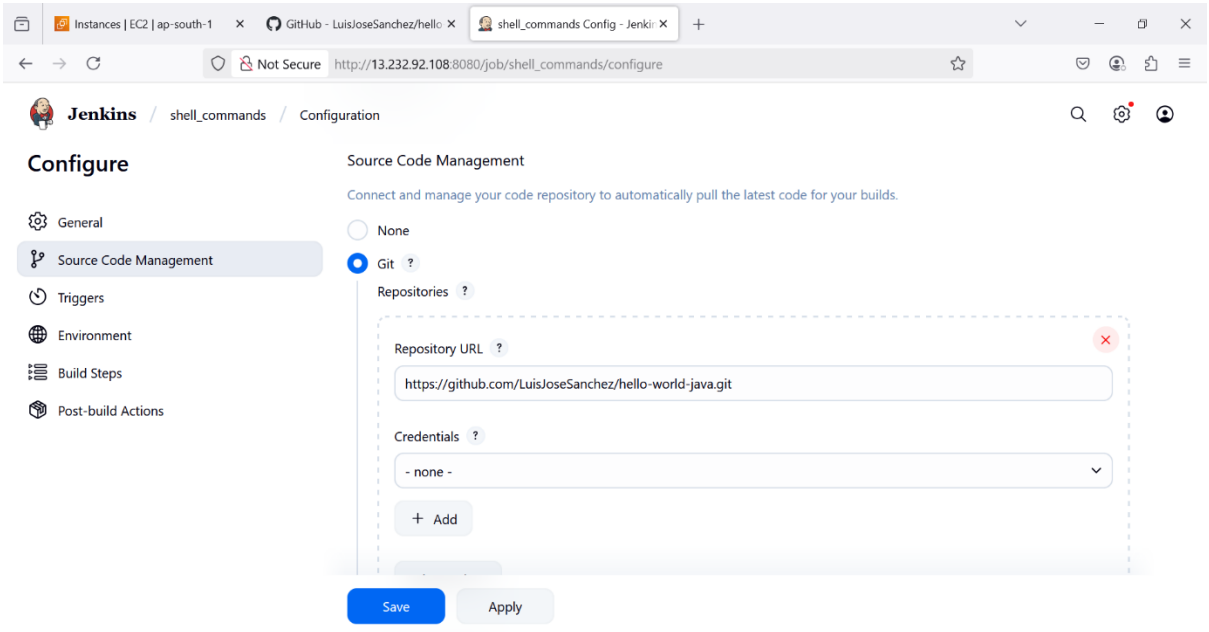




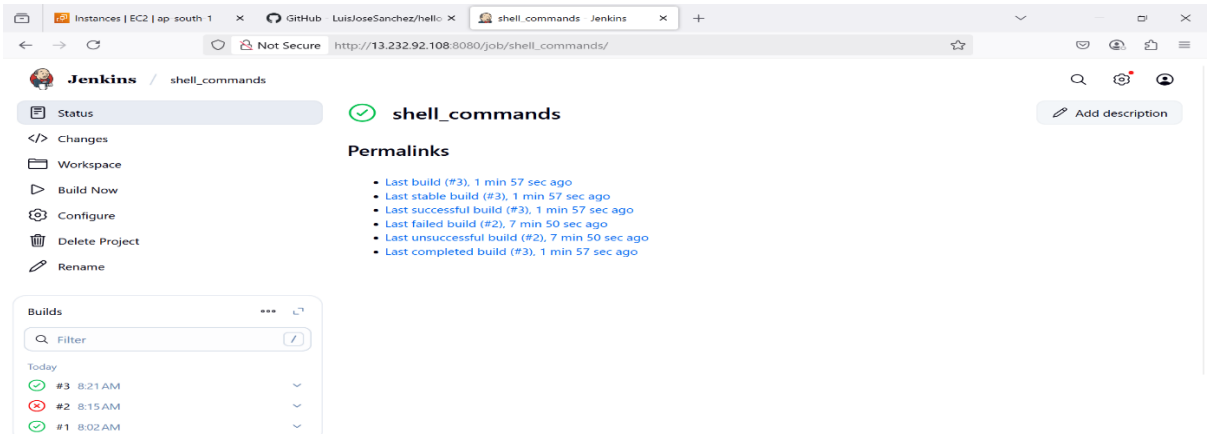


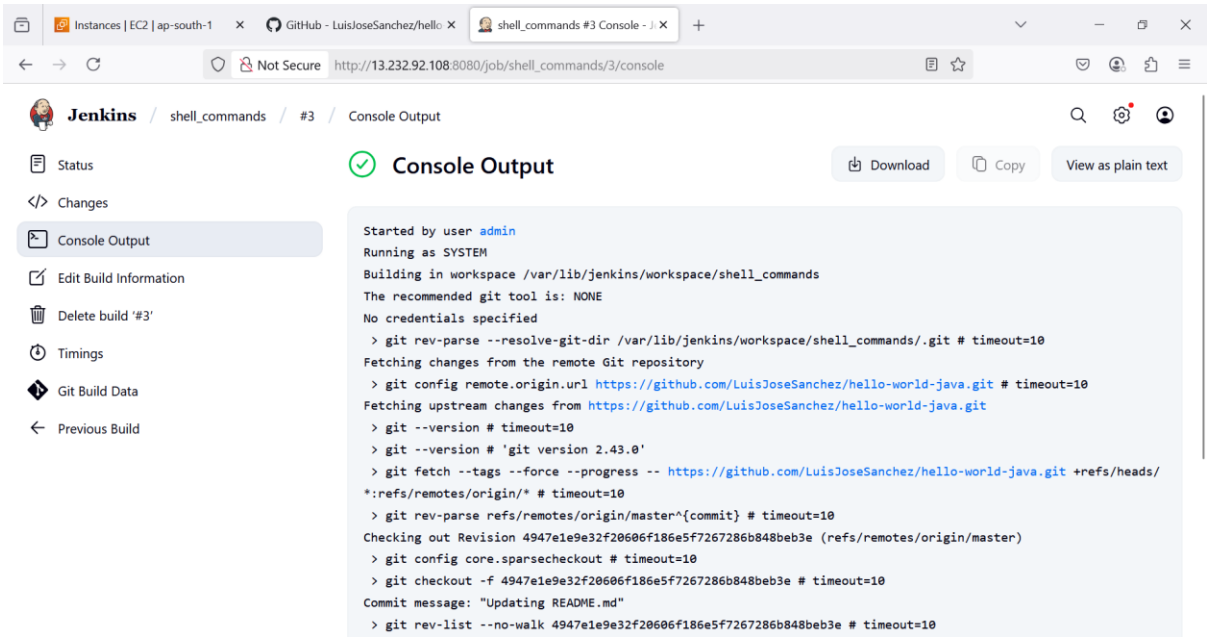
d. install a git plugin and configure repo LuisJoseSanchez/hello-world-java: Hello world with Java (github.com) into it.

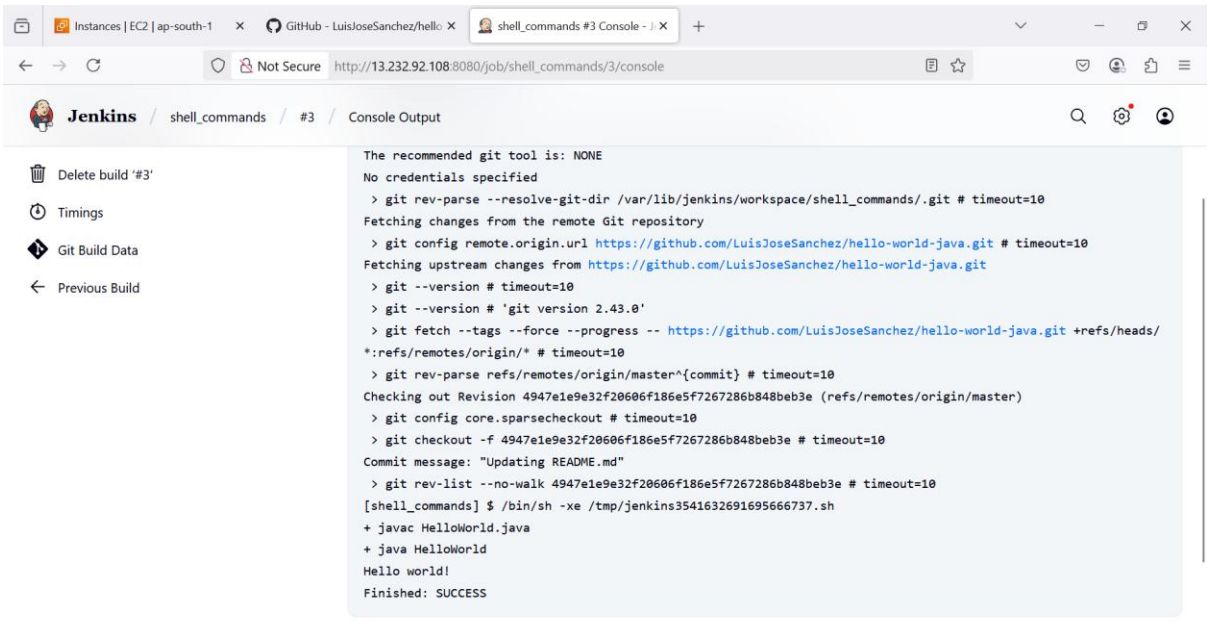




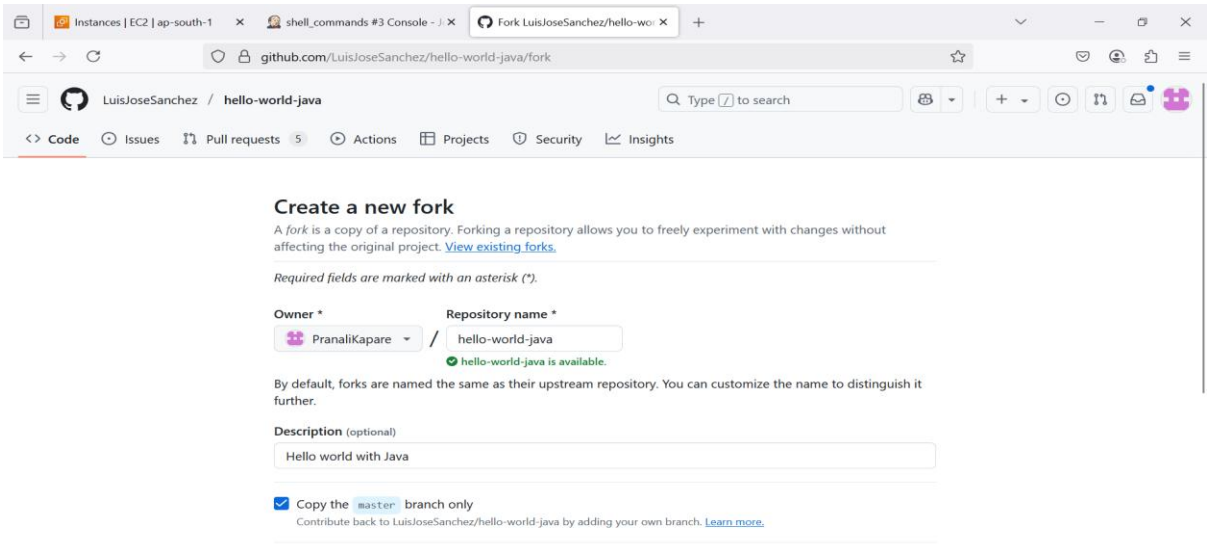
e. Execute the commands given in the repo description and the output should be printed into jenkins console.

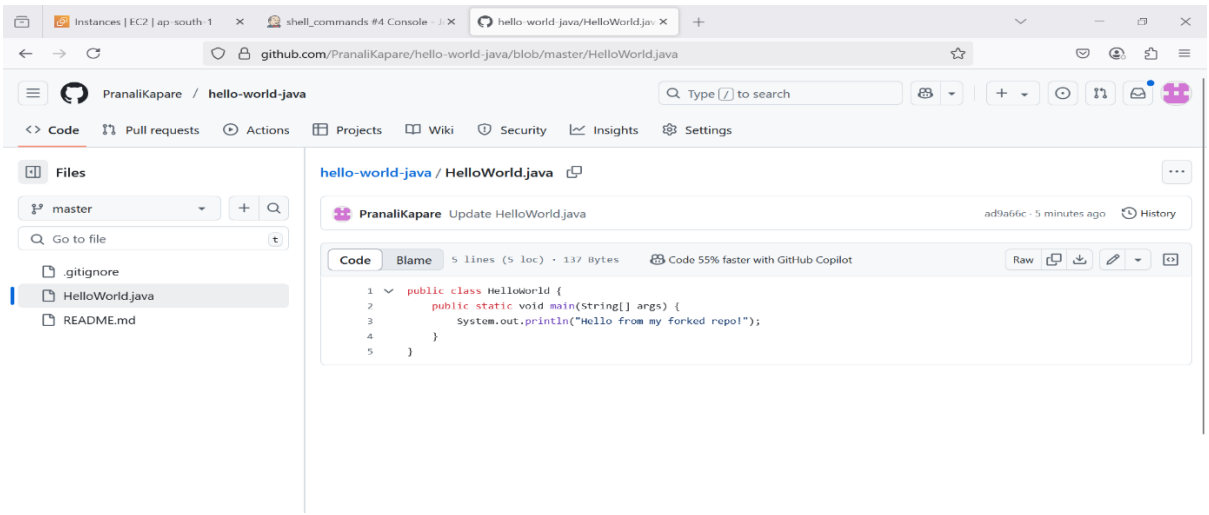


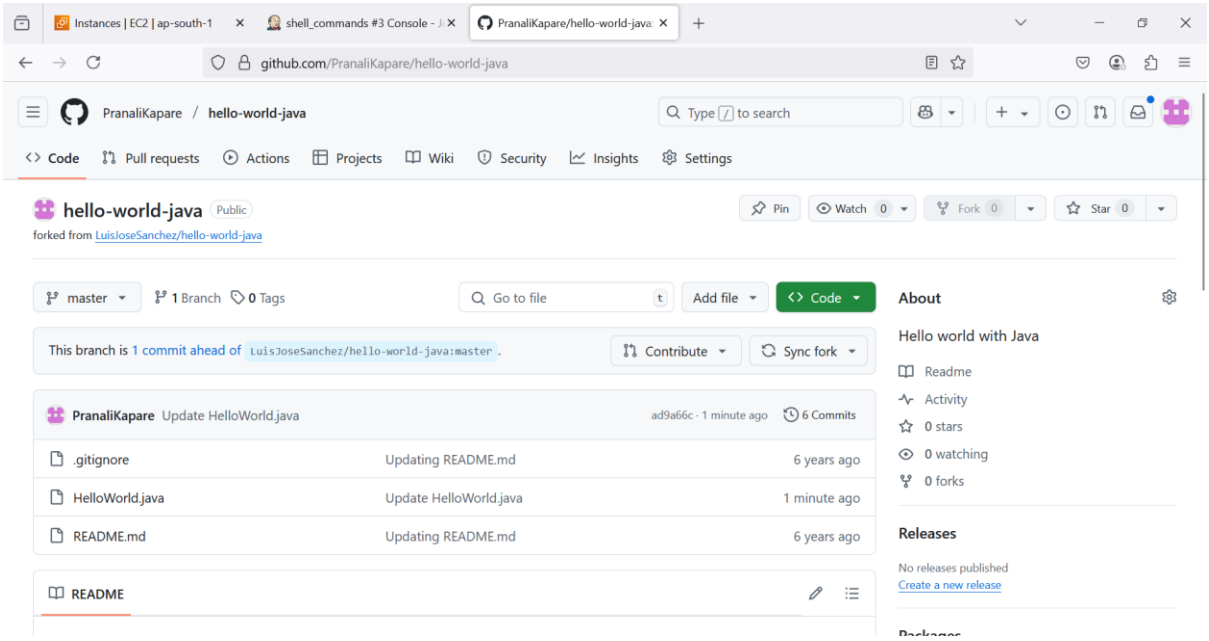


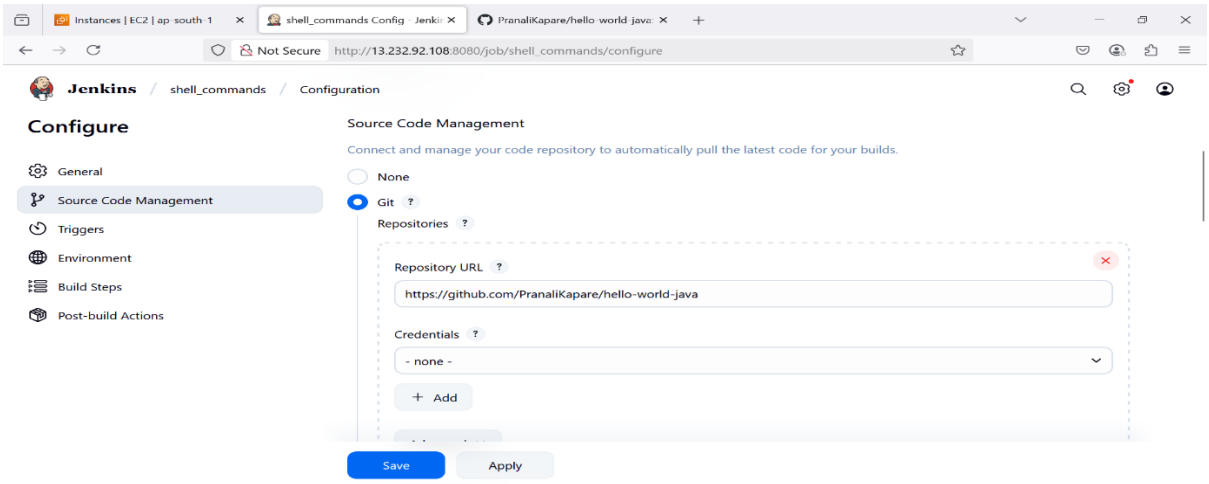


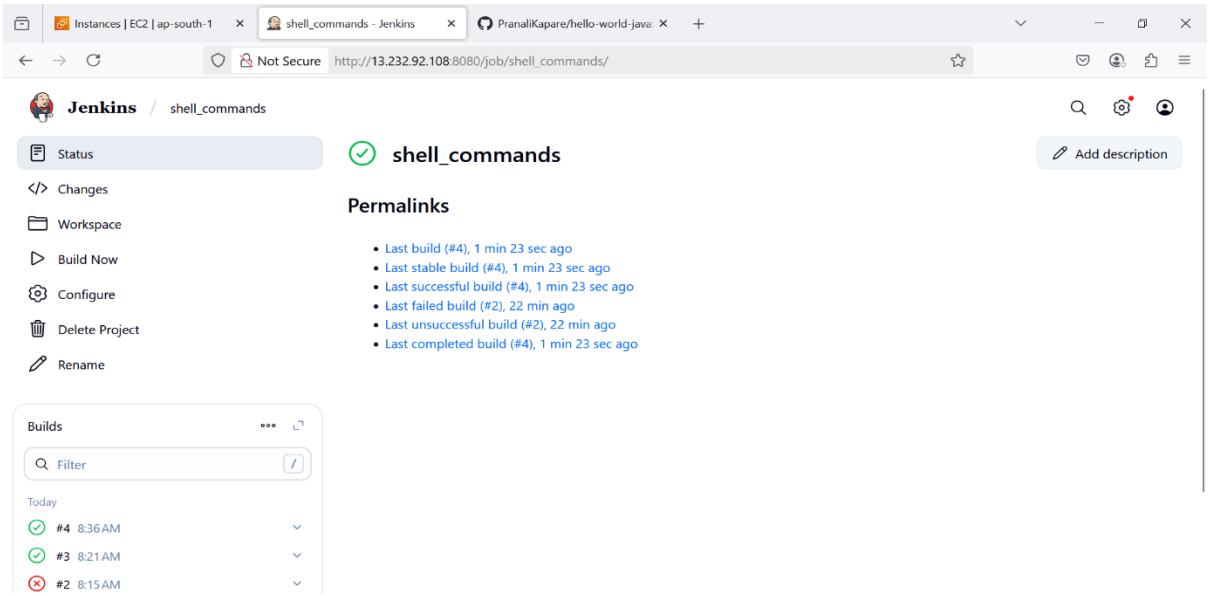
f. Fork the repo, changes the description in print statement into code and rerun the Jenkins job the updated output should be shown into jenkins console log.

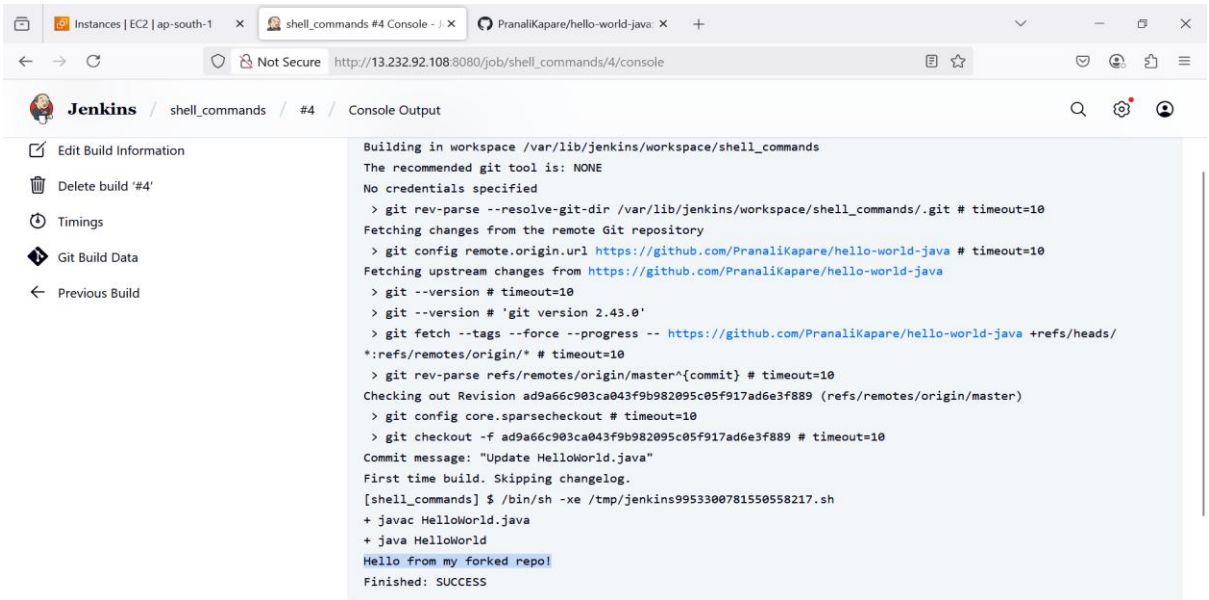






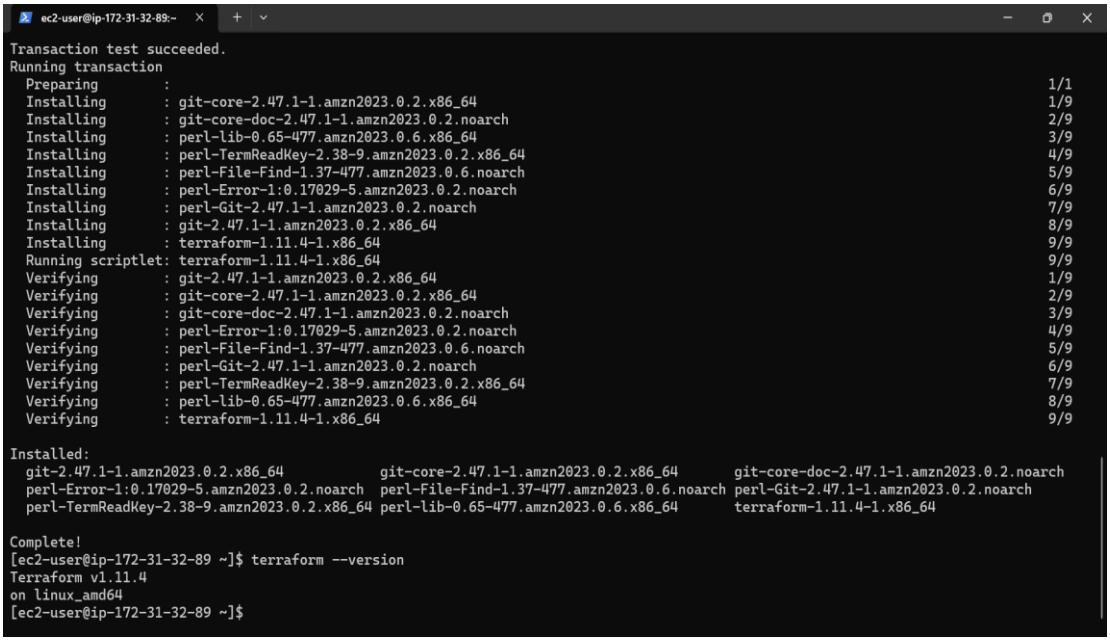

Instances | EC2 | ap-south-1
shell_commands Config
PranaliKapare/hello-world-java
Not Secure
http://13.232.92.108:8080/job/shell_commands/configure
Jenkins / shell_commands / Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
Source Code Management
Connect and manage your code repository to automatically pull the latest code for your builds.
None
Git
Repositories
Repository URL
https://github.com/PranaliKapare/hello-world-java
Credentials
- none -
Add
Save
Apply


Instances | EC2 | ap-south-1
shell_commands - Jenkins
PranaliKapare/hello-world-java
Not Secure
http://13.232.92.108:8080/job/shell_commands/
Jenkins / shell_commands
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
shell_commands
Add description
Permalinks
Last build (#4), 1 min 23 sec ago
Last stable build (#4), 1 min 23 sec ago
Last successful build (#4), 1 min 23 sec ago
Last failed build (#2), 22 min ago
Last unsuccessful build (#2), 22 min ago
Last completed build (#4), 1 min 23 sec ago
Builds
Filter
Today
#4 8:36 AM
#3 8:21 AM
#2 8:15 AM

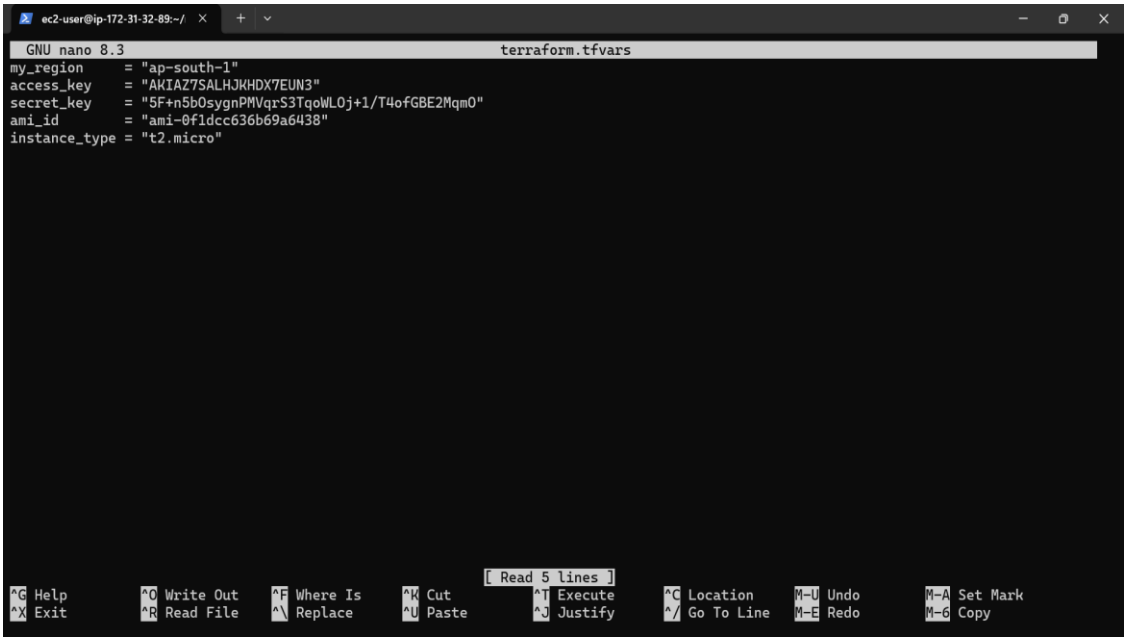

Instances | EC2 | ap-south-1
shell_commands #4 Console
PranaliKapare/hello-world-java
Not Secure
http://13.232.92.108:8080/job/shell_commands/4/console
Jenkins / shell_commands / #4 / Console Output
Edit Build Information
Delete build '#4'
Timings
Git Build Data
Previous Build
Building in workspace /var/lib/jenkins/workspace/shell_commands
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/shell_commands/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/PranaliKapare/hello-world-java # timeout=10
Fetching upstream changes from https://github.com/PranaliKapare/hello-world-java
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/PranaliKapare/hello-world-java +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision ad9a66c903ca043f9b982095c05f917ad6e3f889 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f ad9a66c903ca043f9b982095c05f917ad6e3f889 # timeout=10
Commit message: "Update HelloWorld.java"
First time build. Skipping changelog.
[shell_commands] $ /bin/sh -xe /tmp/jenkins9953300781550558217.sh
+ javac HelloWorld.java
+ java HelloWorld
Hello from my forked repo!
Finished: SUCCESS

4. Terraform Installation and Setup

- Install Terraform on your local machine or a designated server.



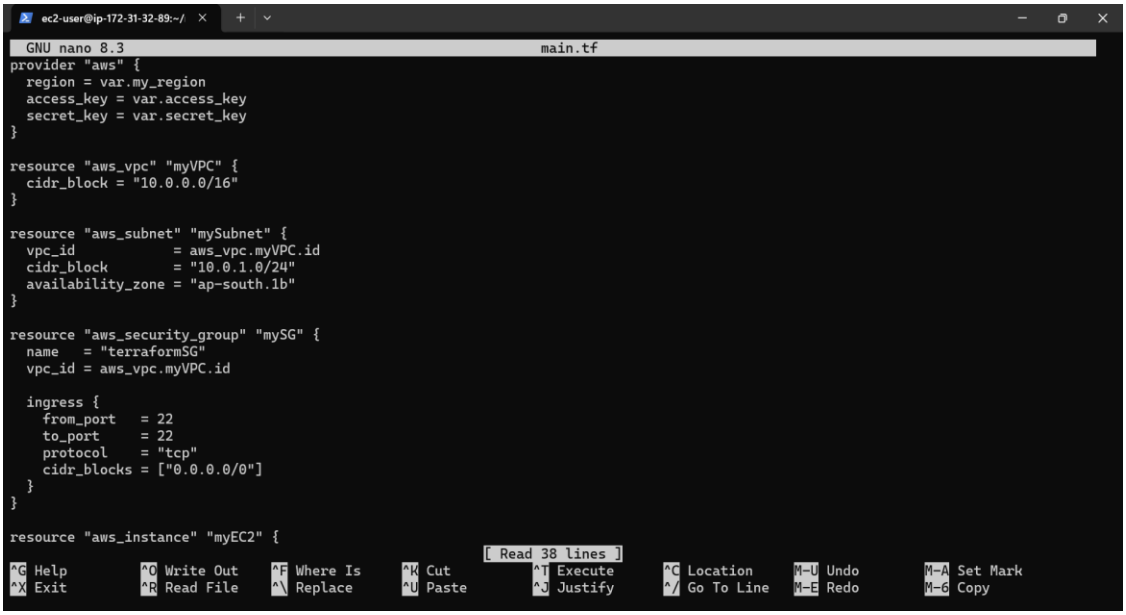
- Configure your cloud provider's credentials for Terraform.



Define Infrastructure Resources

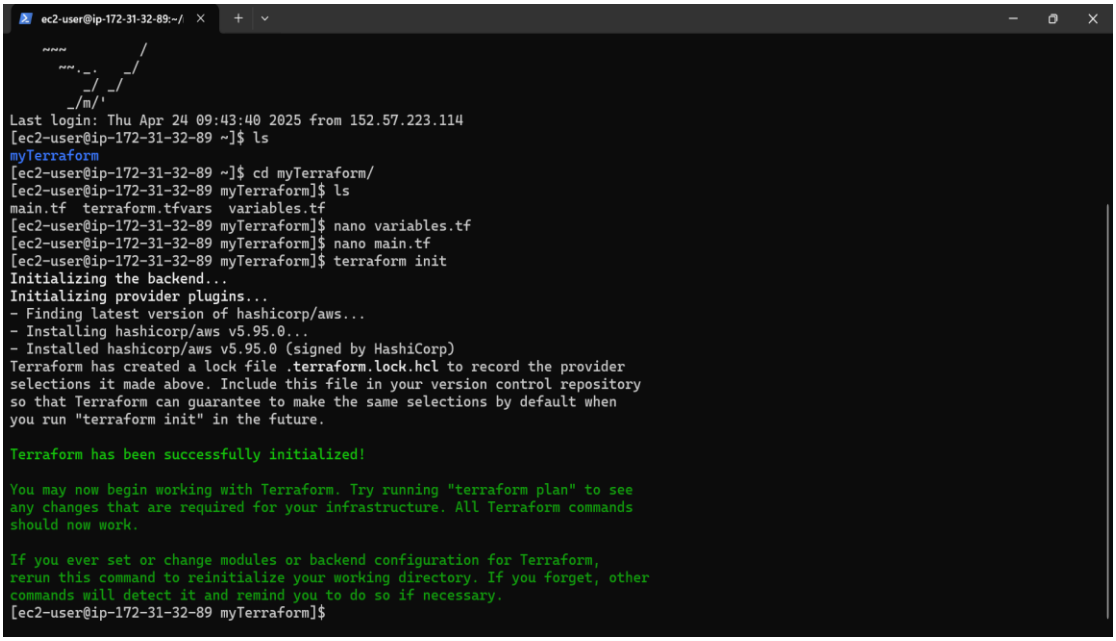
- Define Terraform configuration files (.tf) to create infrastructure resources using Terraform's HashiCorp Configuration Language (HCL).

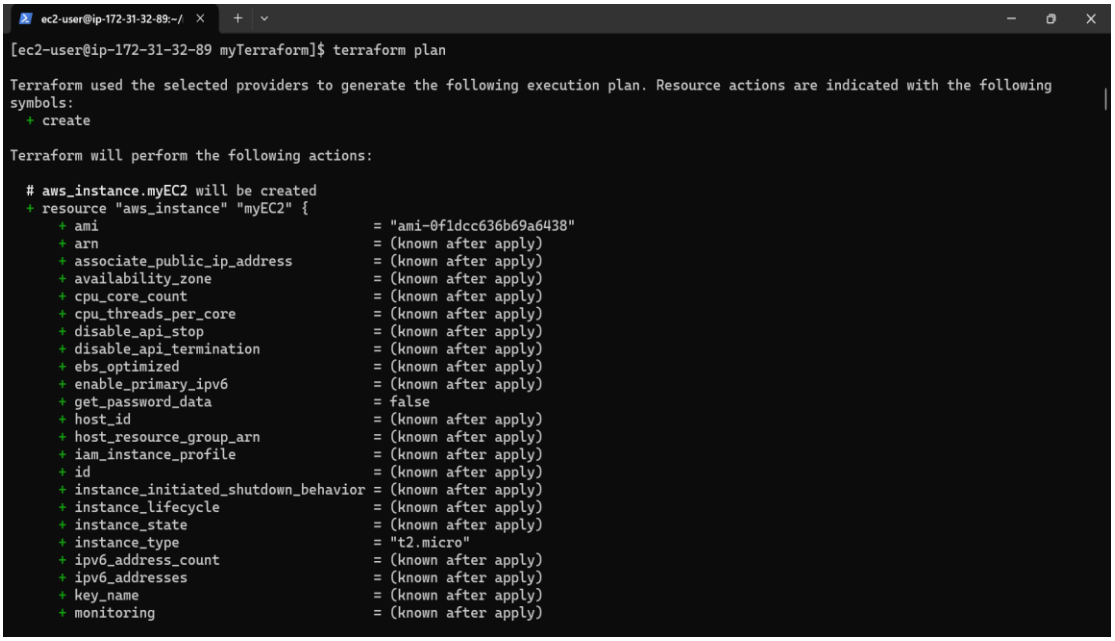
- Create resources such as virtual machines, storage, networking components, and security groups. Use a cloud AWS.
- Use variables and data sources to parameterize your configuration.

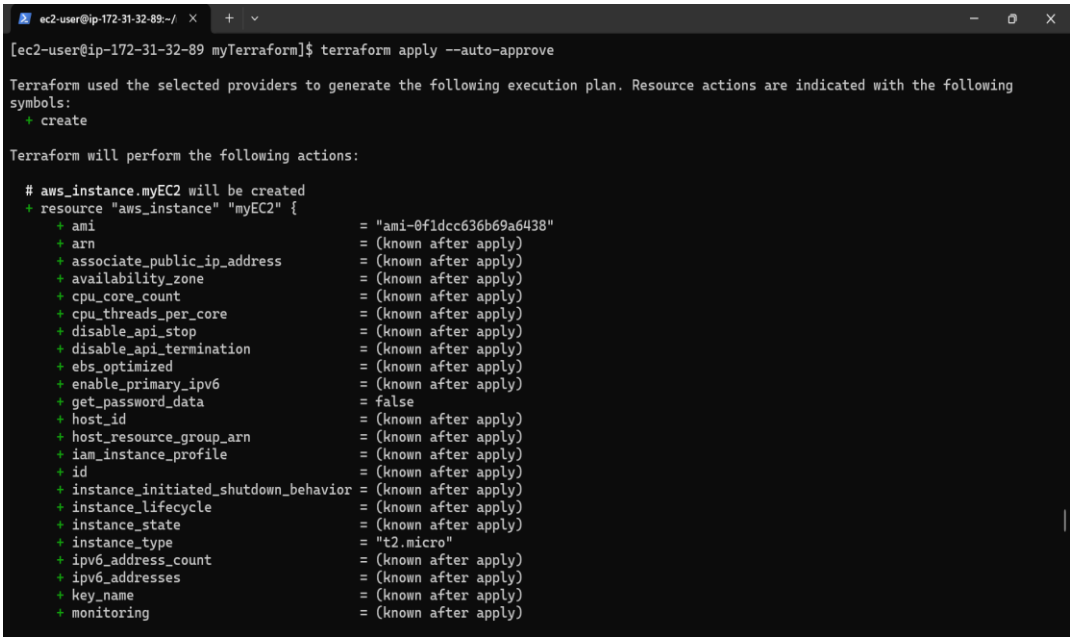


Provision and Apply Infrastructure:

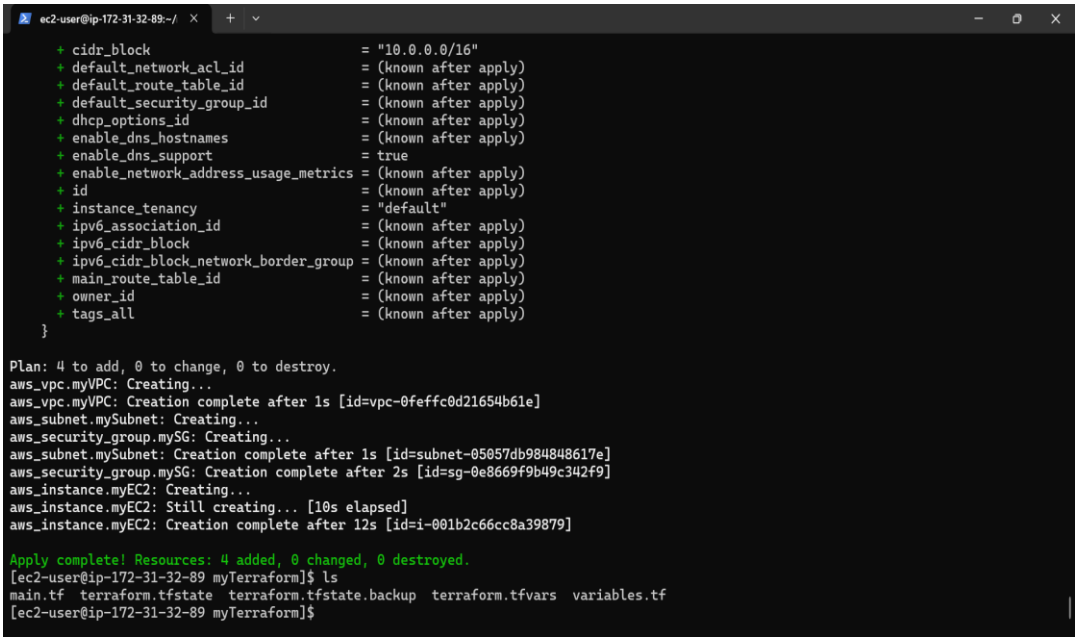
- Use the terraform init, terraform plan, and terraform apply commands to initialize and provision your infrastructure.


ec2-user@ip-172-31-32-89:~/
Last login: Thu Apr 24 09:43:40 2025 from 152.57.223.114
[ec2-user@ip-172-31-32-89 ~]$ ls
myTerraform
[ec2-user@ip-172-31-32-89 ~]$ cd myTerraform/
[ec2-user@ip-172-31-32-89 myTerraform]$ ls
main.tf terraform.tfvars variables.tf
[ec2-user@ip-172-31-32-89 myTerraform]$ nano variables.tf
[ec2-user@ip-172-31-32-89 myTerraform]$ nano main.tf
[ec2-user@ip-172-31-32-89 myTerraform]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.95.0...
- Installed hashicorp/aws v5.95.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-32-89 myTerraform]$


ec2-user@ip-172-31-32-89:~/
[ec2-user@ip-172-31-32-89 myTerraform]$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
+ create
Terraform will perform the following actions:
# aws_instance.myEC2 will be created
+ resource "aws_instance" "myEC2" {
+ ami = "ami-0f1dcc636b69a6438"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ monitoring = (known after apply)

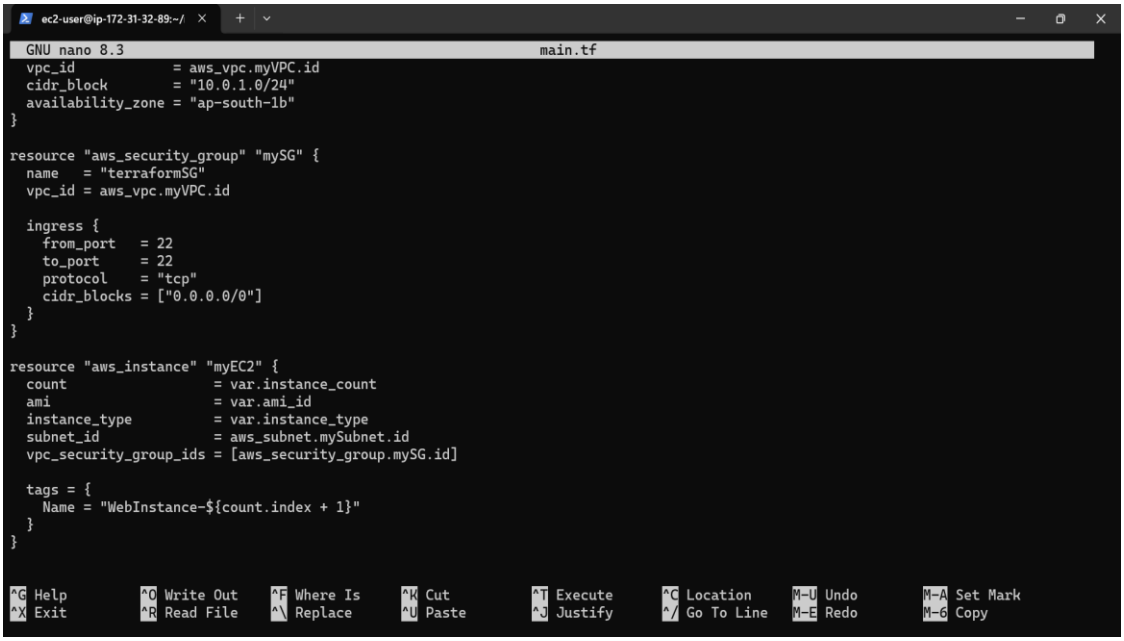


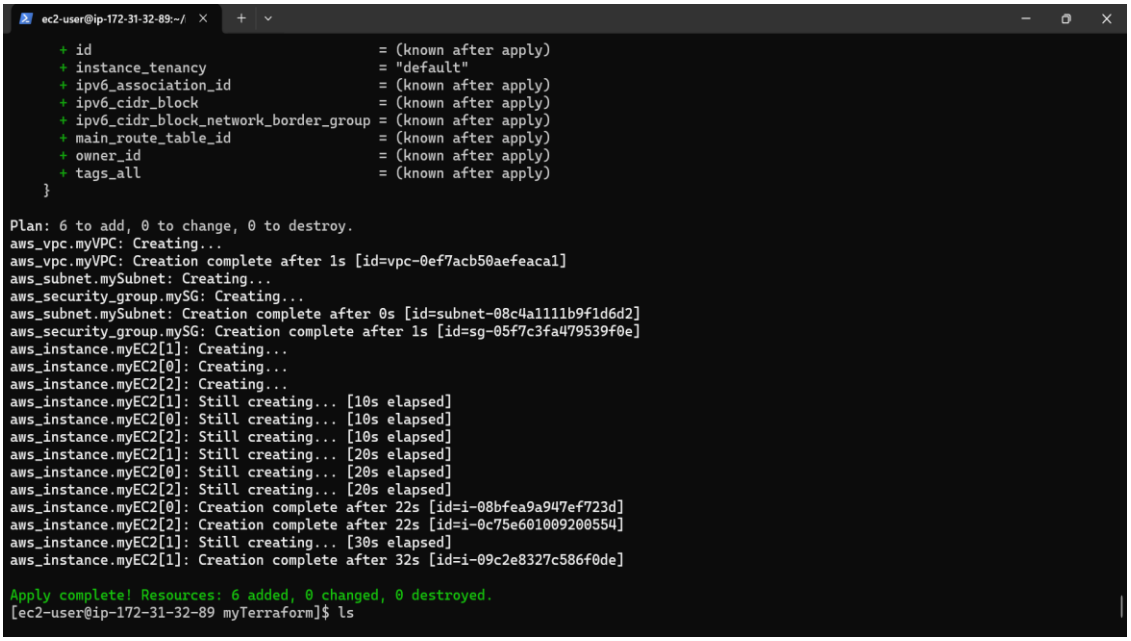
- Make sure that Terraform successfully provisions the defined resources and saves their state.



Scaling and Variable Expansion :

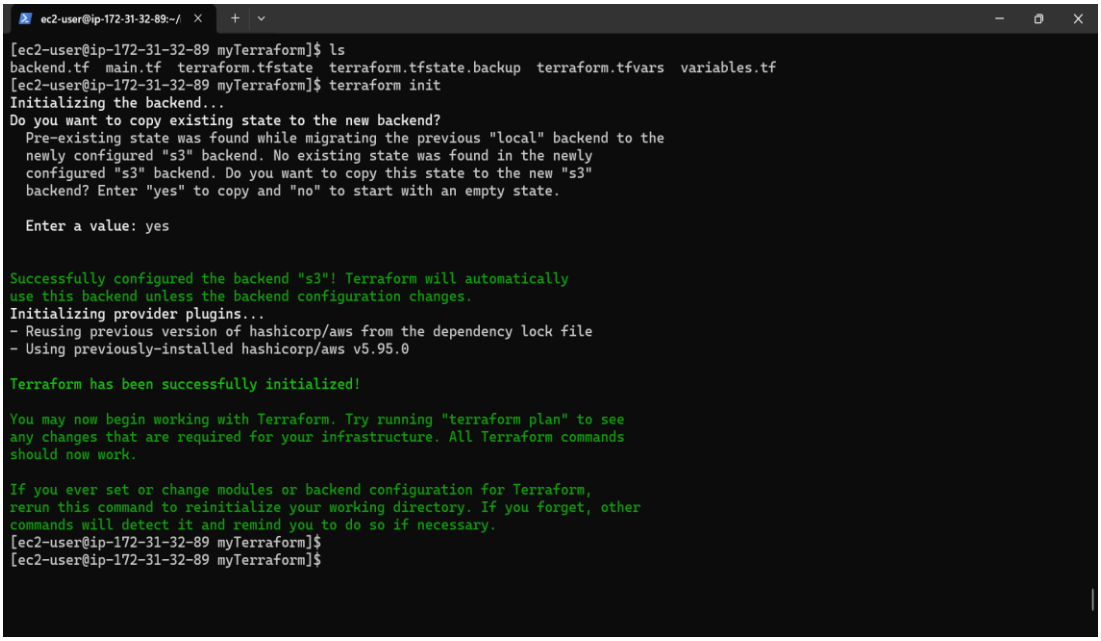
- Extend your Terraform configuration to allow for horizontal scaling of resources (e.g., autoscaling groups or instance counts).
- Implement variable expansion and dynamic resource creation based on the number of instances required.

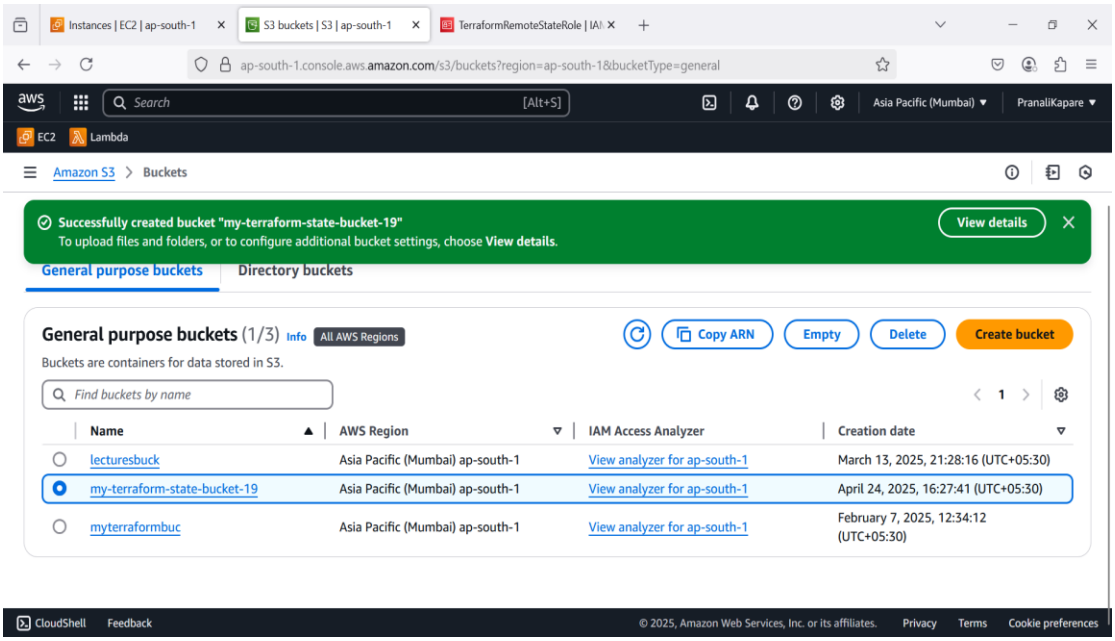


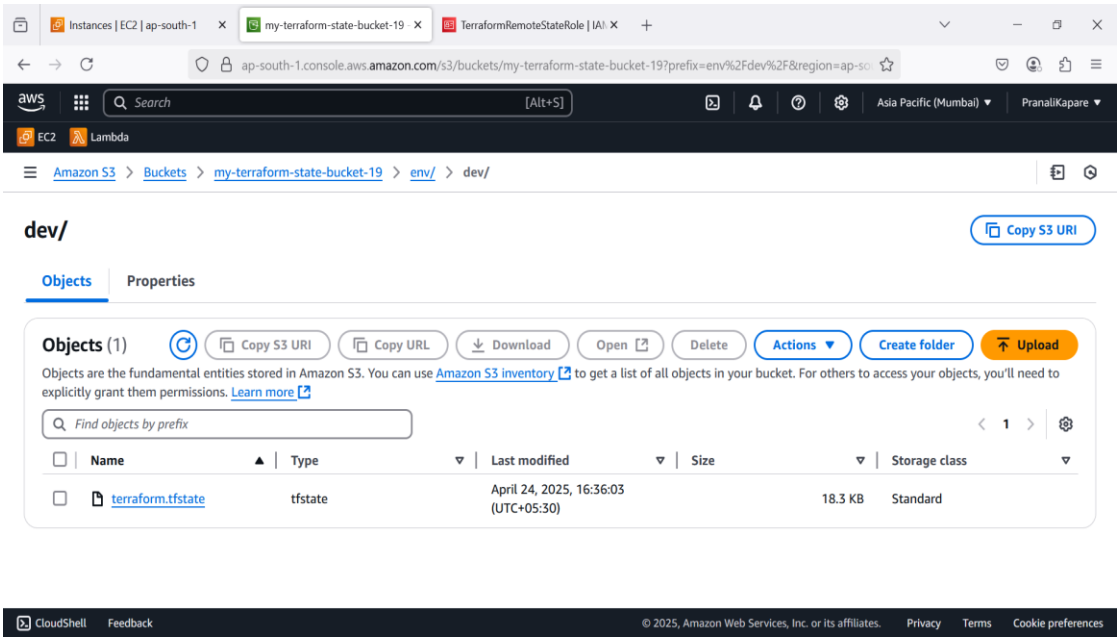


Remote State Management :

- Set up remote state management using a remote backend, such as an S3 bucket for AWS or a similar solution for your cloud provider.
- Ensure that Terraform state files are stored remotely and securely.

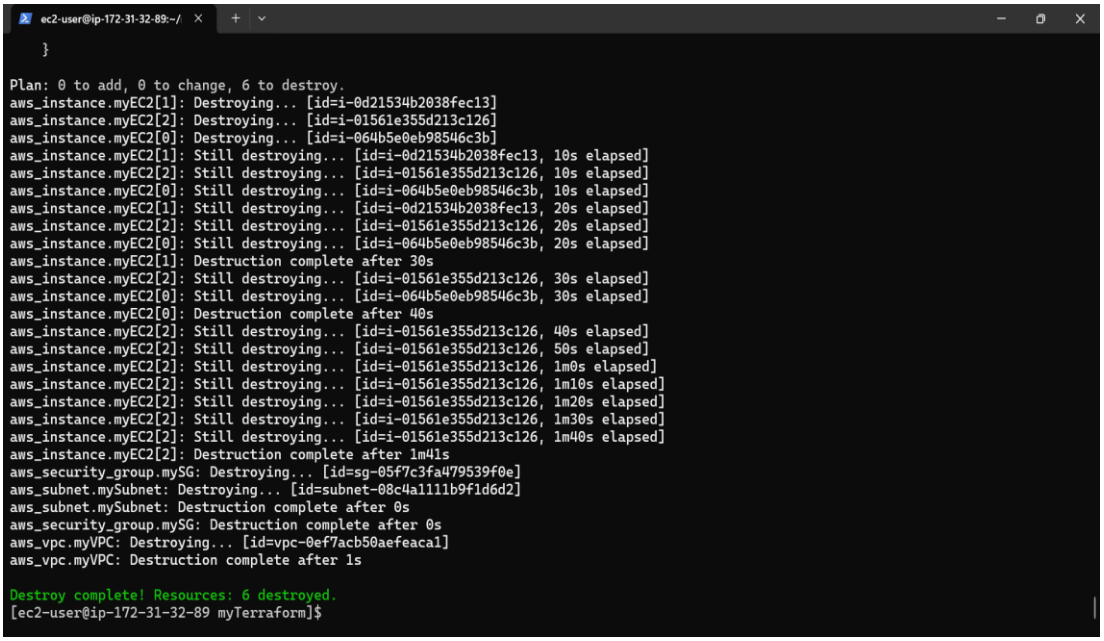

ec2-user@ip-172-31-32-89:~/
[ec2-user@ip-172-31-32-89 myTerraform]$ ls
backend.tf main.tf terraform.tfstate terraform.tfstate.backup terraform.tfvars variables.tf
[ec2-user@ip-172-31-32-89 myTerraform]$ terraform init
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.
  Enter a value: yes
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.95.0
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-32-89 myTerraform]$
[ec2-user@ip-172-31-32-89 myTerraform]$


Instances | EC2 | ap-south-1
S3 buckets | S3 | ap-south-1
TerraformRemoteStateRole | IA
ap-south-1.console.aws.amazon.com/s3/buckets?region=ap-south-1&bucketType=general
Search
[Alt+S]
Asia Pacific (Mumbai)
PranaliKapare
EC2
Lambda
Amazon S3 > Buckets
Successfully created bucket "my-terraform-state-bucket-19"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details
General purpose buckets
Directory buckets
General purpose buckets (1/3) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
lecturesbuck
Asia Pacific (Mumbai) ap-south-1
View analyzer for ap-south-1
March 13, 2025, 21:28:16 (UTC+05:30)
my-terraform-state-bucket-19
Asia Pacific (Mumbai) ap-south-1
View analyzer for ap-south-1
April 24, 2025, 16:27:41 (UTC+05:30)
myterraformbuc
Asia Pacific (Mumbai) ap-south-1
View analyzer for ap-south-1
February 7, 2025, 12:34:12 (UTC+05:30)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Infrastructure Updates and Changes :

- Modify your Terraform configuration to simulate changes to the infrastructure, such as adding or modifying resources.
- Apply these changes using the terraform apply command and verify that Terraform updates the infrastructure accordingly.


ec2-user@ip-172-31-32-89:~/
}
Plan: 0 to add, 0 to change, 6 to destroy.
aws_instance.myEC2[1]: Destroying... [id=i-0d21534b2038fec13]
aws_instance.myEC2[2]: Destroying... [id=i-01561e355d213c126]
aws_instance.myEC2[0]: Destroying... [id=i-064b5e0eb98546c3b]
aws_instance.myEC2[1]: Still destroying... [id=i-0d21534b2038fec13, 10s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 10s elapsed]
aws_instance.myEC2[0]: Still destroying... [id=i-064b5e0eb98546c3b, 10s elapsed]
aws_instance.myEC2[1]: Still destroying... [id=i-0d21534b2038fec13, 20s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 20s elapsed]
aws_instance.myEC2[0]: Still destroying... [id=i-064b5e0eb98546c3b, 20s elapsed]
aws_instance.myEC2[1]: Destruction complete after 30s
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 30s elapsed]
aws_instance.myEC2[0]: Still destroying... [id=i-064b5e0eb98546c3b, 30s elapsed]
aws_instance.myEC2[0]: Destruction complete after 40s
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 40s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 50s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 1m0s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 1m10s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 1m20s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 1m30s elapsed]
aws_instance.myEC2[2]: Still destroying... [id=i-01561e355d213c126, 1m40s elapsed]
aws_instance.myEC2[2]: Destruction complete after 1m41s
aws_security_group.mySG: Destroying... [id=sg-05f7c3fa479539f0e]
aws_subnet.mySubnet: Destroying... [id=subnet-08c4a1111b9f1d6d2]
aws_subnet.mySubnet: Destruction complete after 0s
aws_security_group.mySG: Destruction complete after 0s
aws_vpc.myVPC: Destroying... [id=vpc-0ef7acb50aefeaca1]
aws_vpc.myVPC: Destruction complete after 1s
Destroy complete! Resources: 6 destroyed.
[ec2-user@ip-172-31-32-89 myTerraform]$