UPES

UNIVERSITY OF TOMORROW

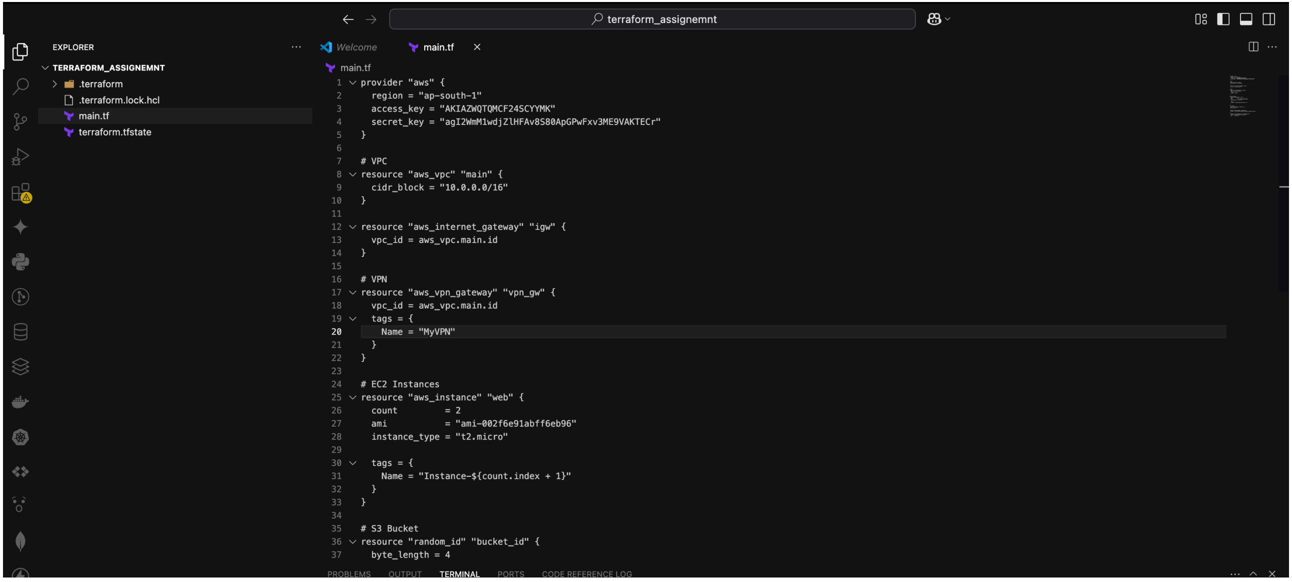
SYSTEM PROVISIONING AND CONFIGURATION

NAME:ARYAN BANSAL
ROLL NO:R2142220237
SAP-ID:500101700

1SUBMITTED TO:
Dr. Hitesh Kumar Sharma

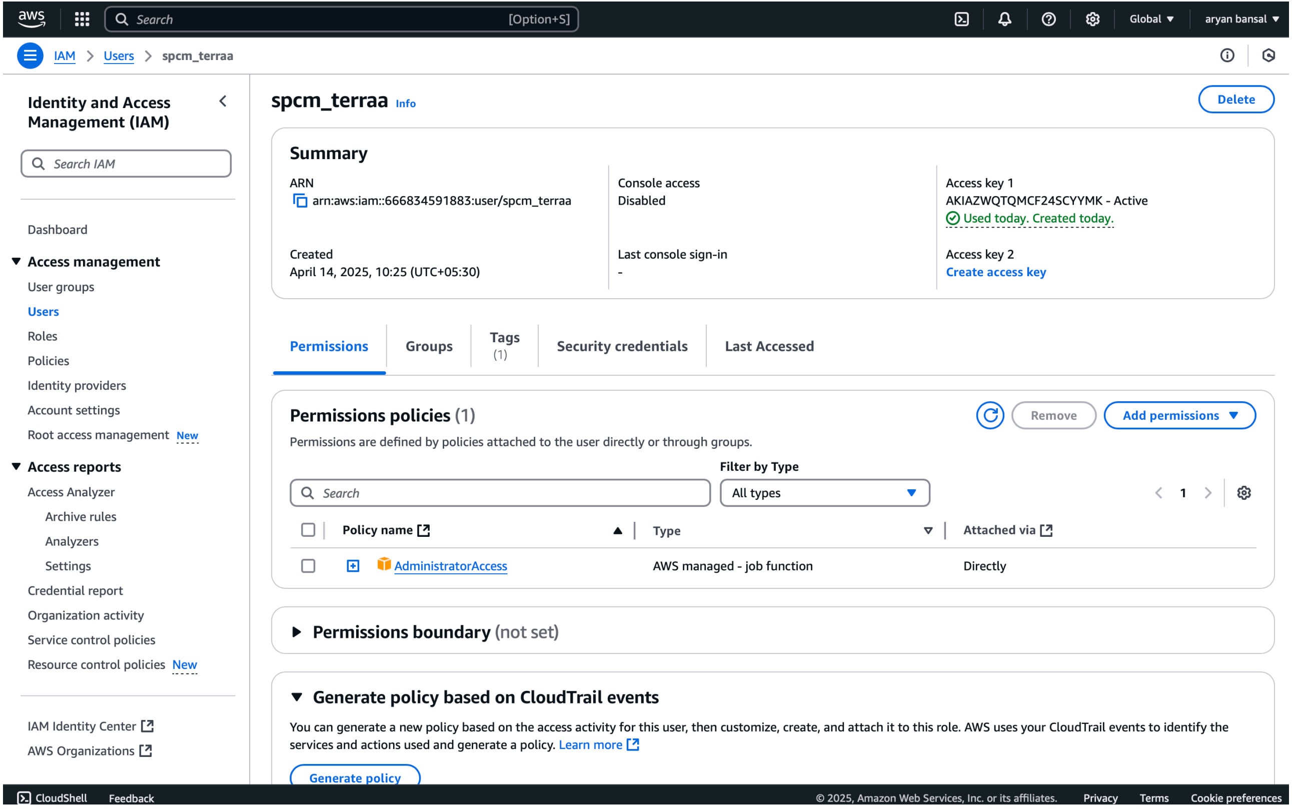
ASSIGNMENT—>1

STEP 1: Terraform Setup



```
provider "aws" {
  region = "ap-south-1"
  access_key = "AKIAZWQTQMCF24SCYYMK"
  secret_key = "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"
}

# VPC
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id
}

# VPN
resource "aws_vpn_gateway" "vpn_gw" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "MyVPN"
  }
}

# EC2 Instances
resource "aws_instance" "web" {
  count         = 2
  ami           = "ami-002f6e91abff6eb96"
  instance_type = "t2.micro"

  tags = {
    Name = "Instance-${count.index + 1}"
  }
}

# S3 Bucket
resource "random_id" "bucket_id" {
  byte_length = 4
```

Step 2: Create IAM user get credential



spcm_terraa Info

Summary

ARN
arn:aws:iam::666834591883:user/spcm_terraa

Created
April 14, 2025, 10:25 (UTC+05:30)

Console access
Disabled

Last console sign-in
-

Access key 1
AKIAZWQTQMCF24SCYYMK - Active
✓ Used today. Created today.

Access key 2
Create access key

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

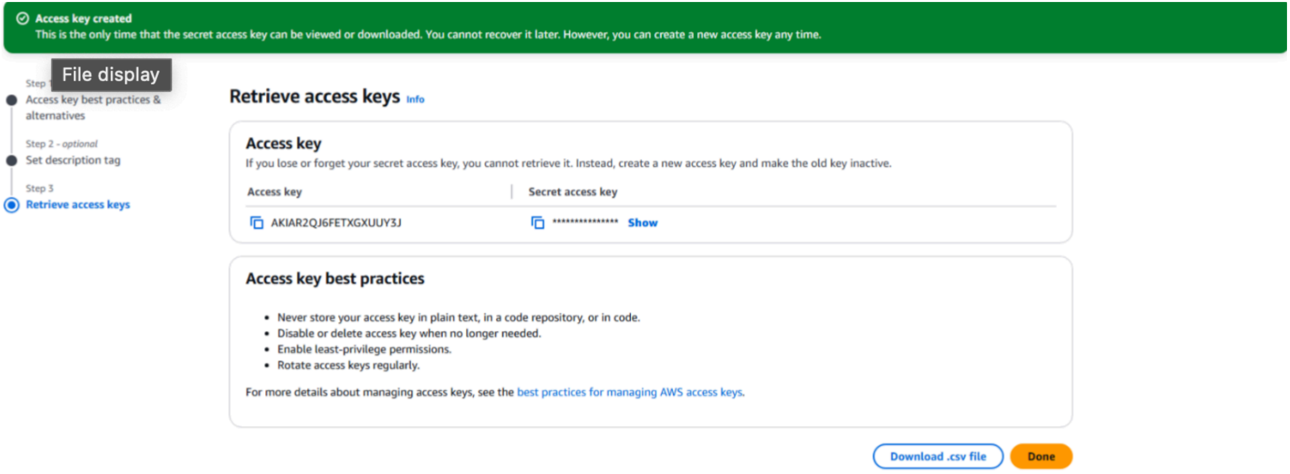
□	Policy name	Type	Attached via
□	AdministratorAccess	AWS managed - job function	Directly

▶ **Permissions boundary** (not set)

▼ **Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

Create the access key



Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

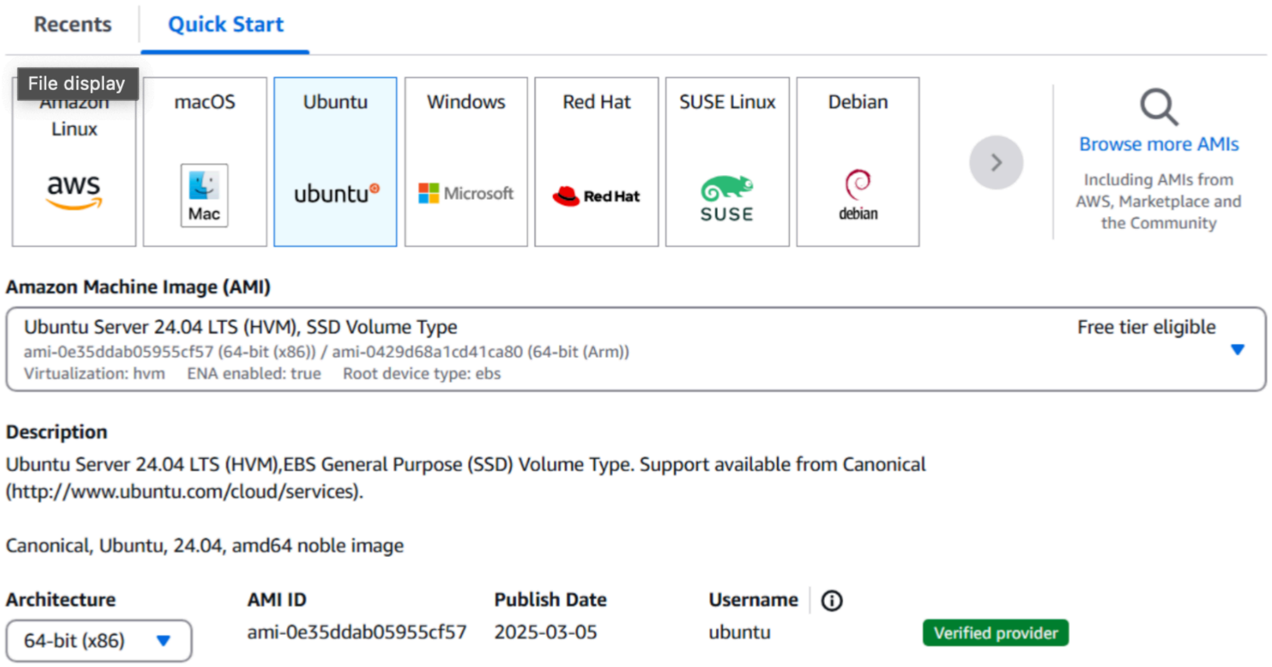
Access key	Secret access key
AKIAR2QJ6FETXGXUUY3J	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

STEP 3:Get the Amazon Linux AMI ID



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0e35ddab05955cf57 (64-bit (x86)) / ami-0429d68a1cd41ca80 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

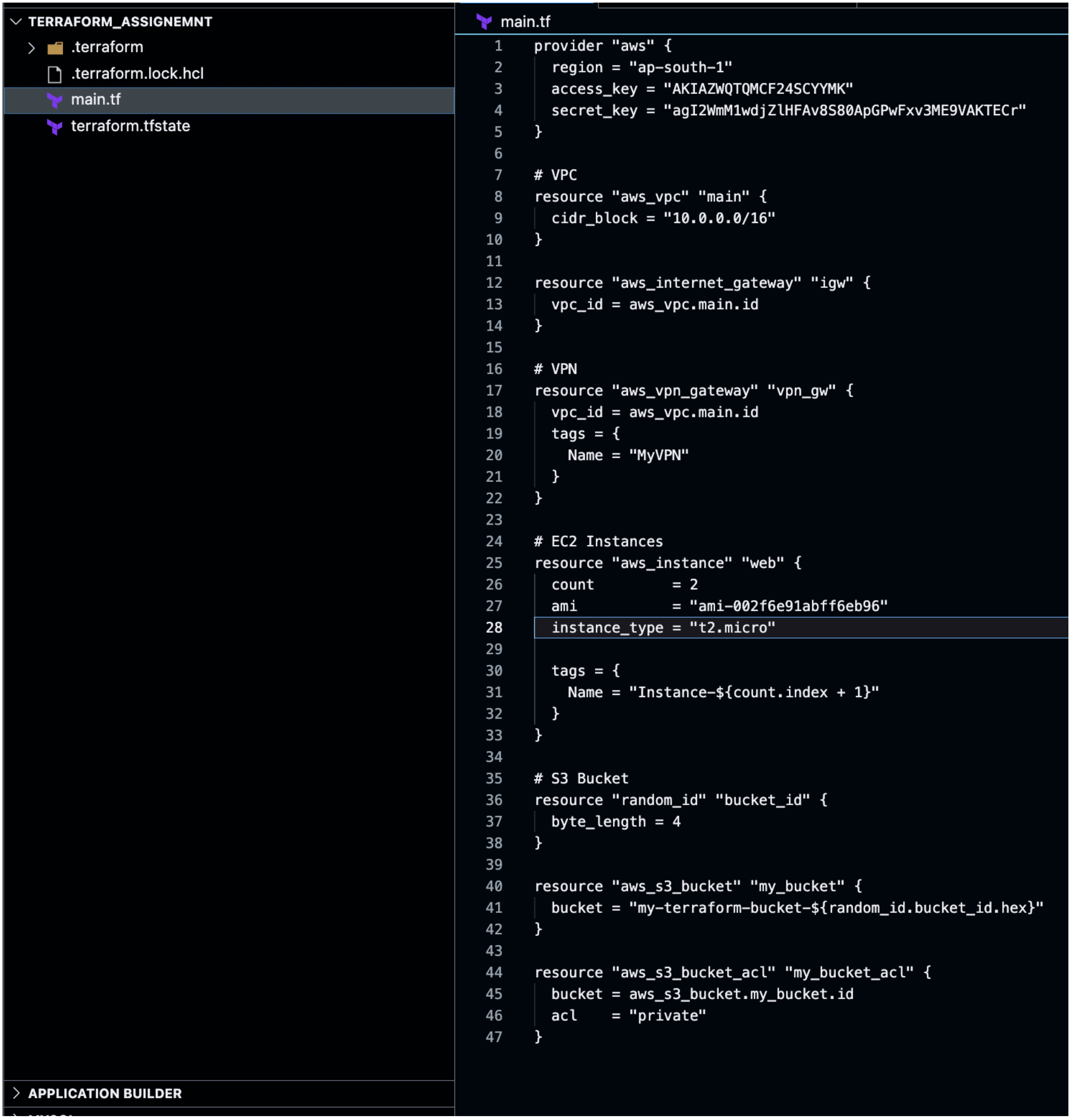
Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	
64-bit (x86)	ami-0e35ddab05955cf57	2025-03-05	ubuntu	Verified provider

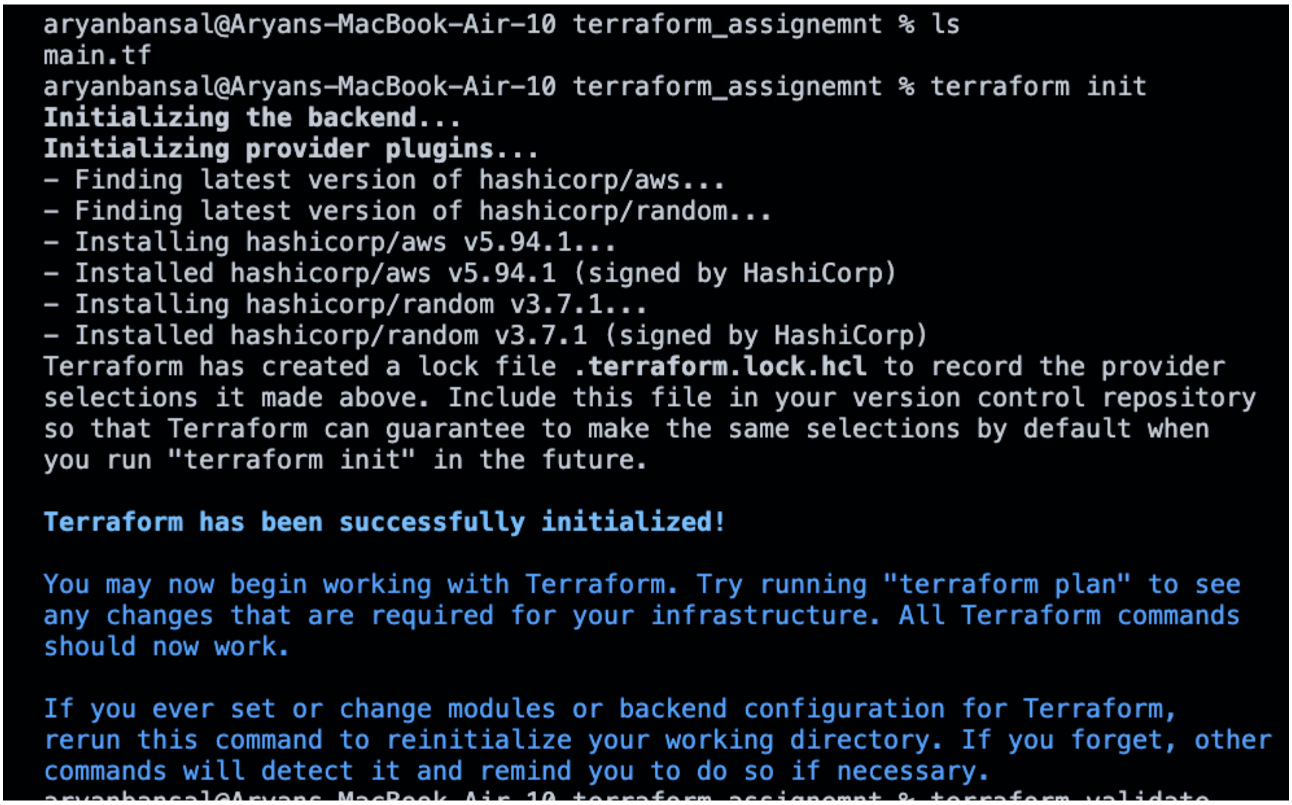
STEP 4:Terraform code (main.tf)



```
provider "aws" {
  region = "ap-south-1"
  access_key = "AKIAZWQTQMCF24SCYYMK"
  secret_key = "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"
}

# VPC
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id
}

# VPN
resource "aws_vpn_gateway" "vpn_gw" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "MyVPN"
  }
}

# EC2 Instances
resource "aws_instance" "web" {
  count         = 2
  ami           = "ami-002f6e91abff6eb96"
  instance_type = "t2.micro"

  tags = {
    Name = "Instance-${count.index + 1}"
  }
}

# S3 Bucket
resource "random_id" "bucket_id" {
  byte_length = 4
}

resource "aws_s3_bucket" "my_bucket" {
  bucket = "my-terraform-bucket-${random_id.bucket_id.hex}"
}

resource "aws_s3_bucket_acl" "my_bucket_acl" {
  bucket = aws_s3_bucket.my_bucket.id
  acl    = "private"
}
```

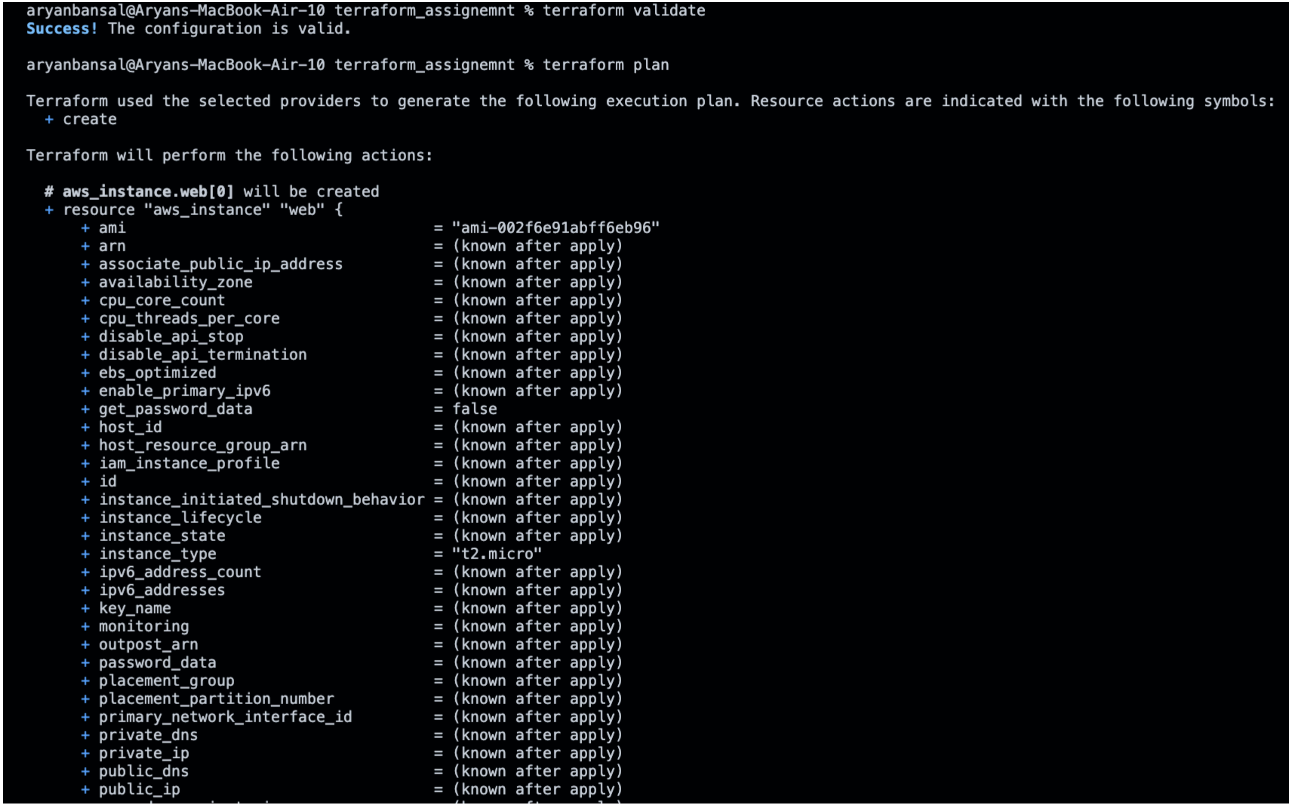
STEP 5: Terraform Commands

Terraform init



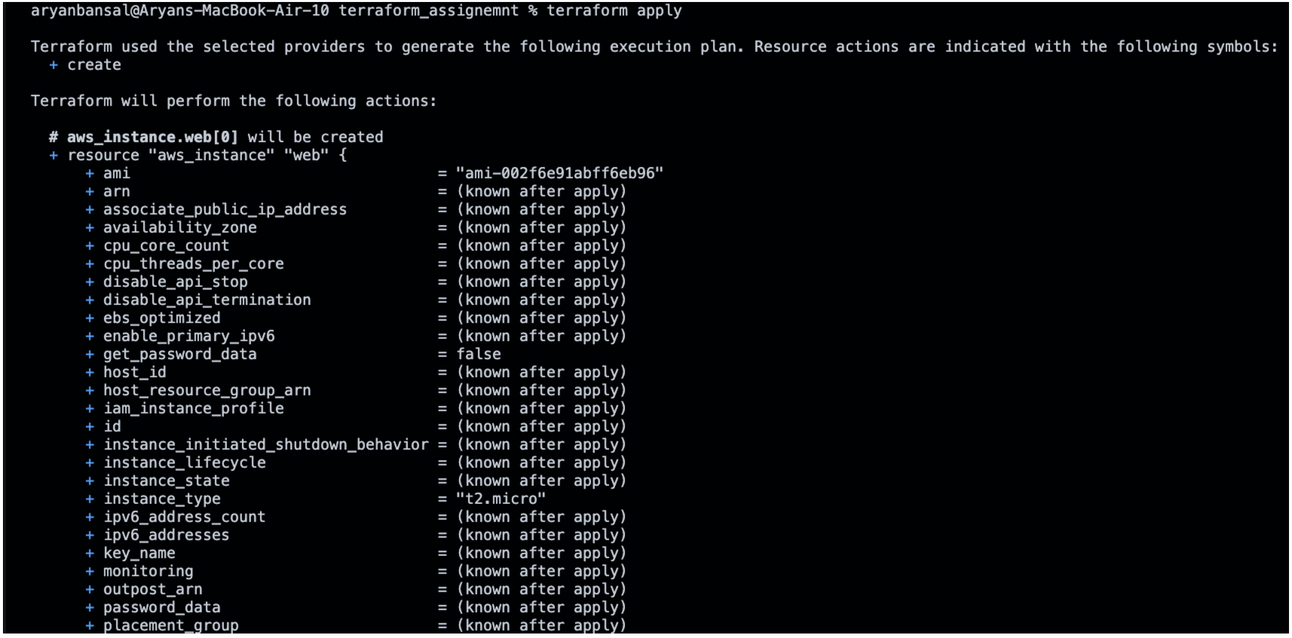
```
aryanbansal@Aryans-MacBook-Air-10 terraform_assignemnt % ls
main.tf
aryanbansal@Aryans-MacBook-Air-10 terraform_assignemnt % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/random...
- Installing hashicorp/aws v5.94.1...
- Installed hashicorp/aws v5.94.1 (signed by HashiCorp)
- Installing hashicorp/random v3.7.1...
- Installed hashicorp/random v3.7.1 (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.
aryanbansal@Aryans-MacBook-Air-10 terraform_assignemnt % terraform validate
```

Terraform Validate && Terraform plan



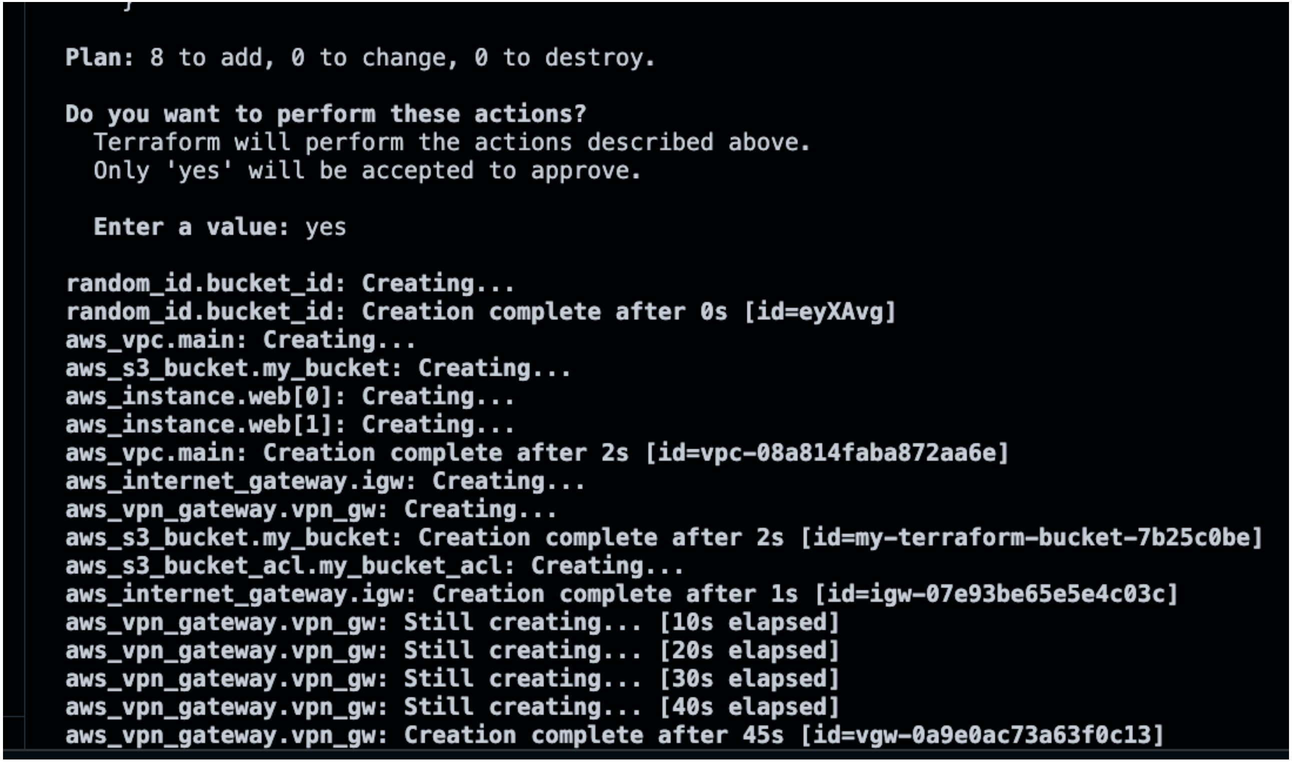
```
aryanbansal@Aryans-MacBook-Air-10 terraform_assignemnt % terraform validate
Success! The configuration is valid.

aryanbansal@Aryans-MacBook-Air-10 terraform_assignemnt % 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.web[0] will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-002f6e91abff6eb96"
      + 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)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
```

Terraform apply



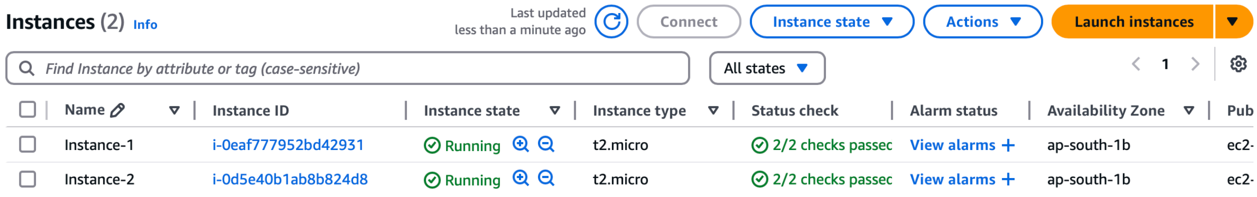
```
aryanbansal@Aryans-MacBook-Air-10 terraform_assignemnt % terraform apply

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.web[0] will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-002f6e91abff6eb96"
      + 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)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```

Output



```
    }

Plan: 8 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

random_id.bucket_id: Creating...
random_id.bucket_id: Creation complete after 0s [id=eyXAvg]
aws_vpc.main: Creating...
aws_s3_bucket.my_bucket: Creating...
aws_instance.web[0]: Creating...
aws_instance.web[1]: Creating...
aws_vpc.main: Creation complete after 2s [id=vpc-08a814faba872aa6e]
aws_internet_gateway.igw: Creating...
aws_vpn_gateway.vpn_gw: Creating...
aws_s3_bucket.my_bucket: Creation complete after 2s [id=my-terraform-bucket-7b25c0be]
aws_s3_bucket_acl.my_bucket_acl: Creating...
aws_internet_gateway.igw: Creation complete after 1s [id=igw-07e93be65e5e4c03c]
aws_vpn_gateway.vpn_gw: Still creating... [10s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [20s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [30s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [40s elapsed]
aws_vpn_gateway.vpn_gw: Creation complete after 45s [id=vgw-0a9e0ac73a63f0c13]
```

The created resources



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
□	Instance-1	i-0eaf777952bd42931	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-
□	Instance-2	i-0d5e40b1ab8b824d8	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-