ASSIGNMENT 1
Write Terraform script to do perform following tasks on AWS cloud Platform

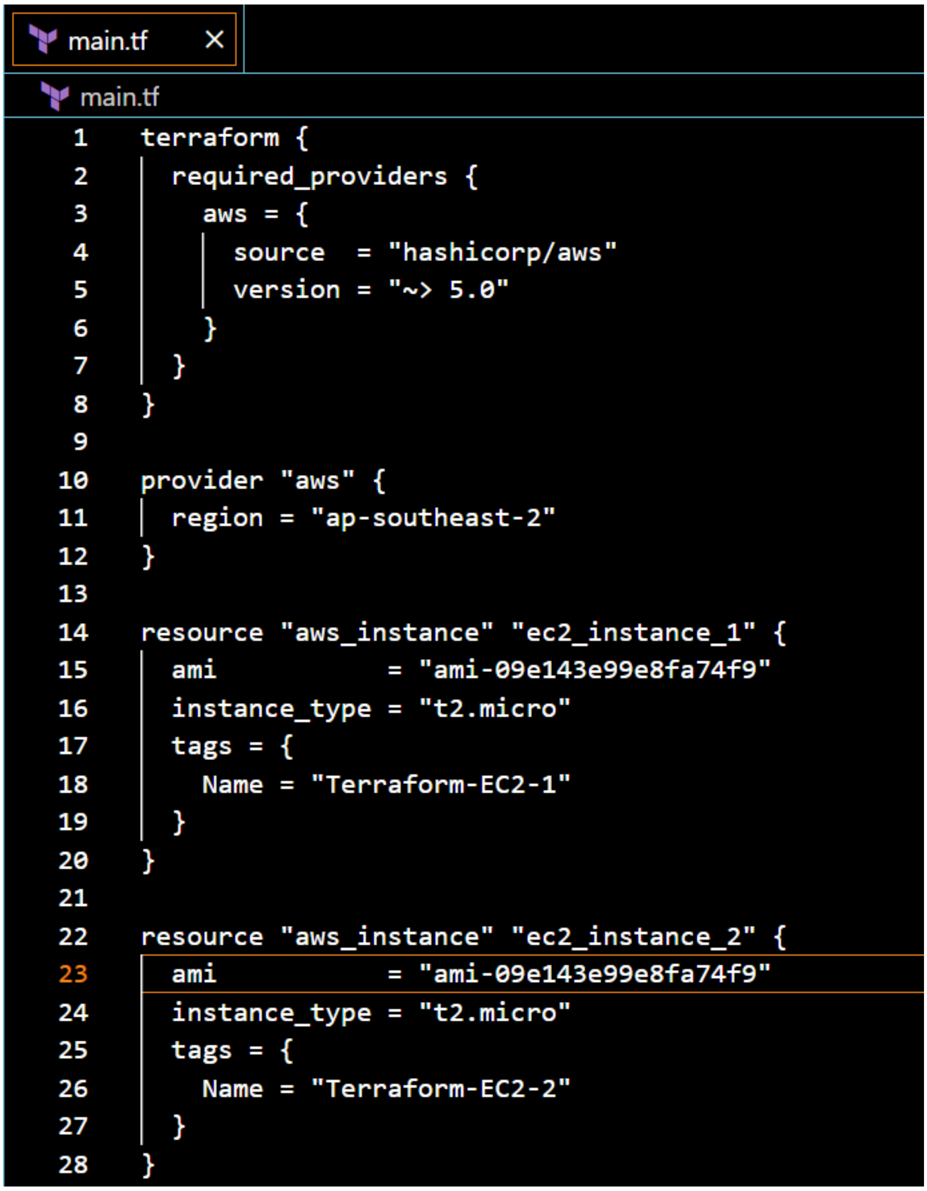
NAME: Jahnavi Anand

ROLL NO.: R2142220089

BATCH: DevOps B1-NH

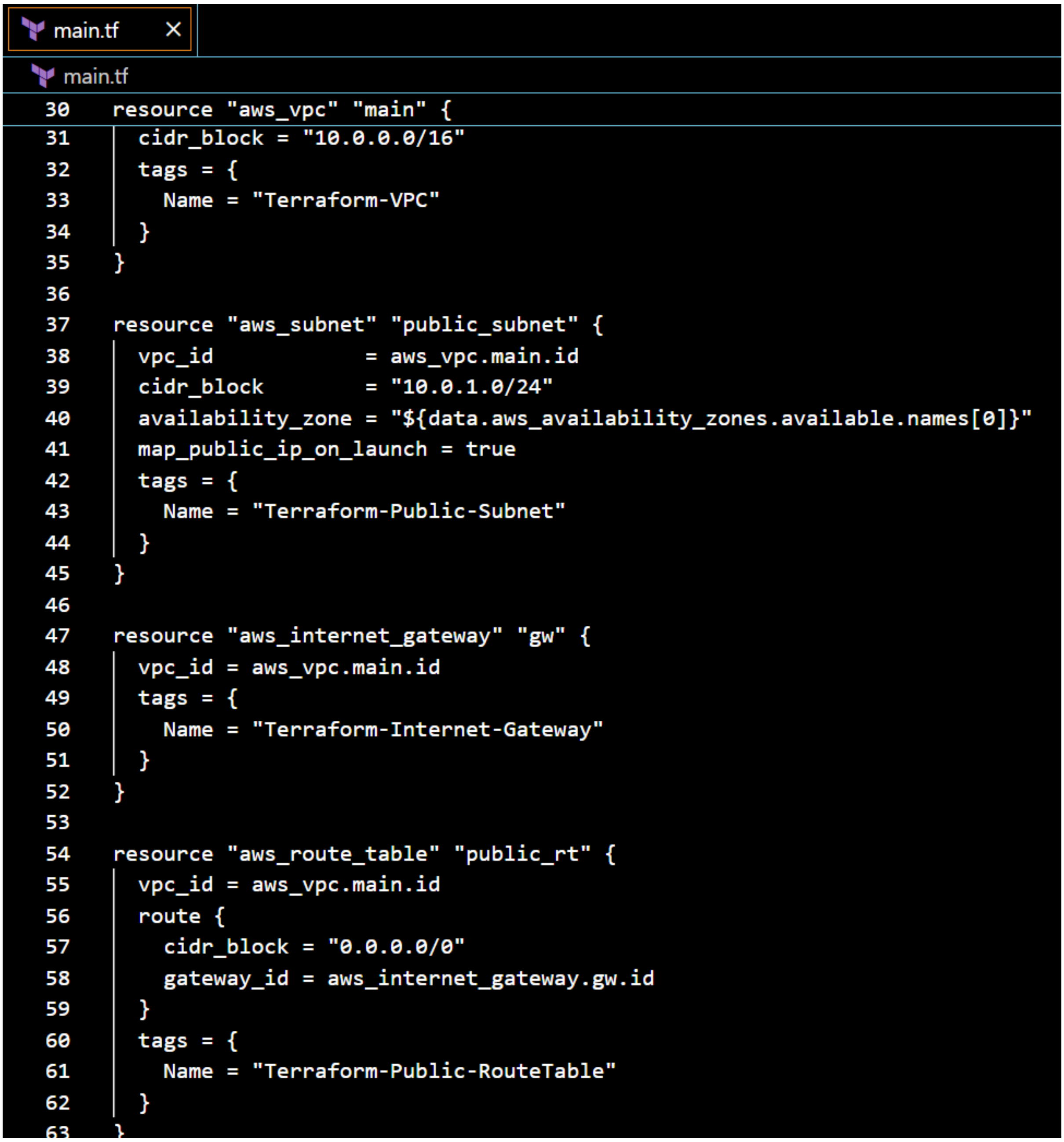
SUBMITTED TO: Prof. Hitesh Sharma

Step 1: Create two T2 Micro EC2 Instances.



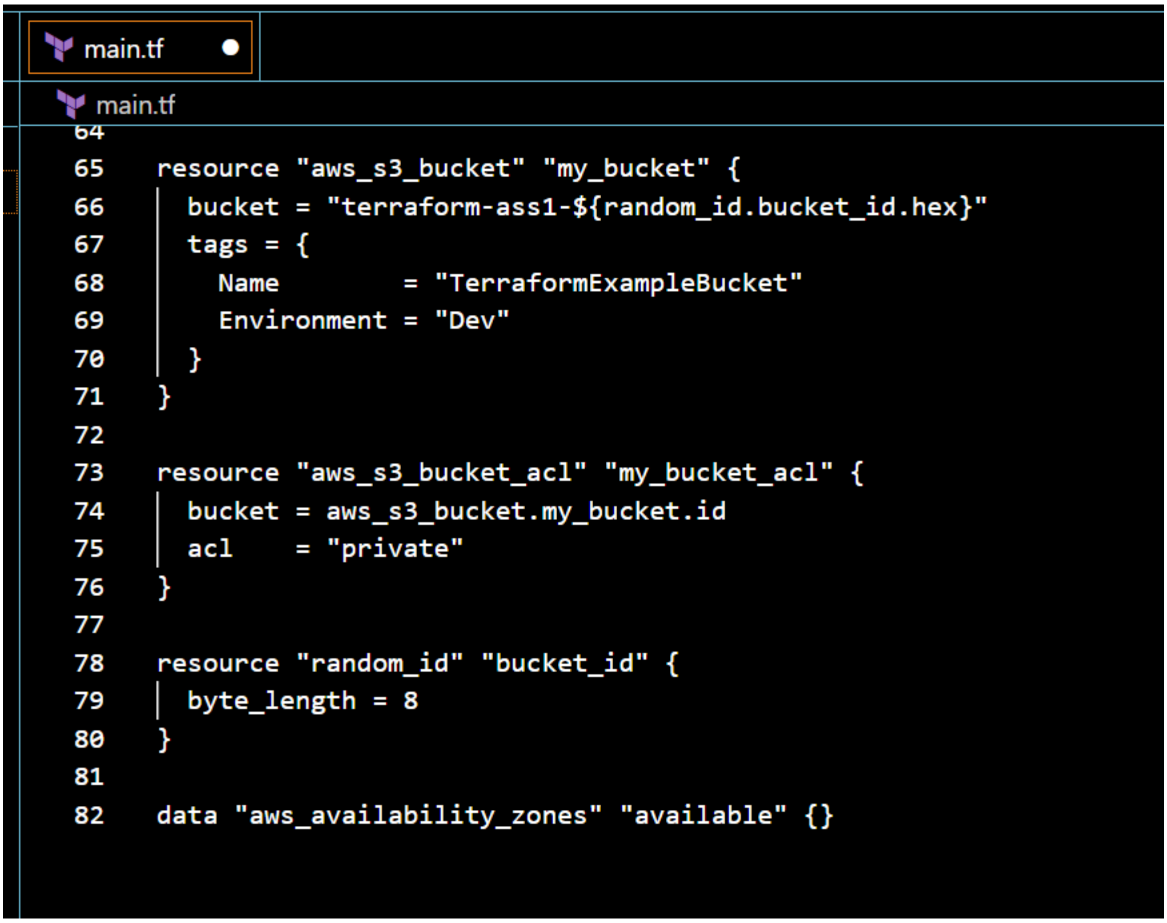
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = "ap-southeast-2"
}

resource "aws_instance" "ec2_instance_1" {
  ami           = "ami-09e143e99e8fa74f9"
  instance_type = "t2.micro"
  tags = {
    Name = "Terraform-EC2-1"
  }
}

resource "aws_instance" "ec2_instance_2" {
  ami           = "ami-09e143e99e8fa74f9"
  instance_type = "t2.micro"
  tags = {
    Name = "Terraform-EC2-2"
  }
}
```

Step2: Create a VPN on AWS



```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "Terraform-VPC"
  }
}

resource "aws_subnet" "public_subnet" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "${data.aws_availability_zones.available.names[0]}"
  map_public_ip_on_launch = true
  tags = {
    Name = "Terraform-Public-Subnet"
  }
}

resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "Terraform-Internet-Gateway"
  }
}

resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.main.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
  tags = {
    Name = "Terraform-Public-RouteTable"
  }
}
```

Step 3: Create a S3 Bucket



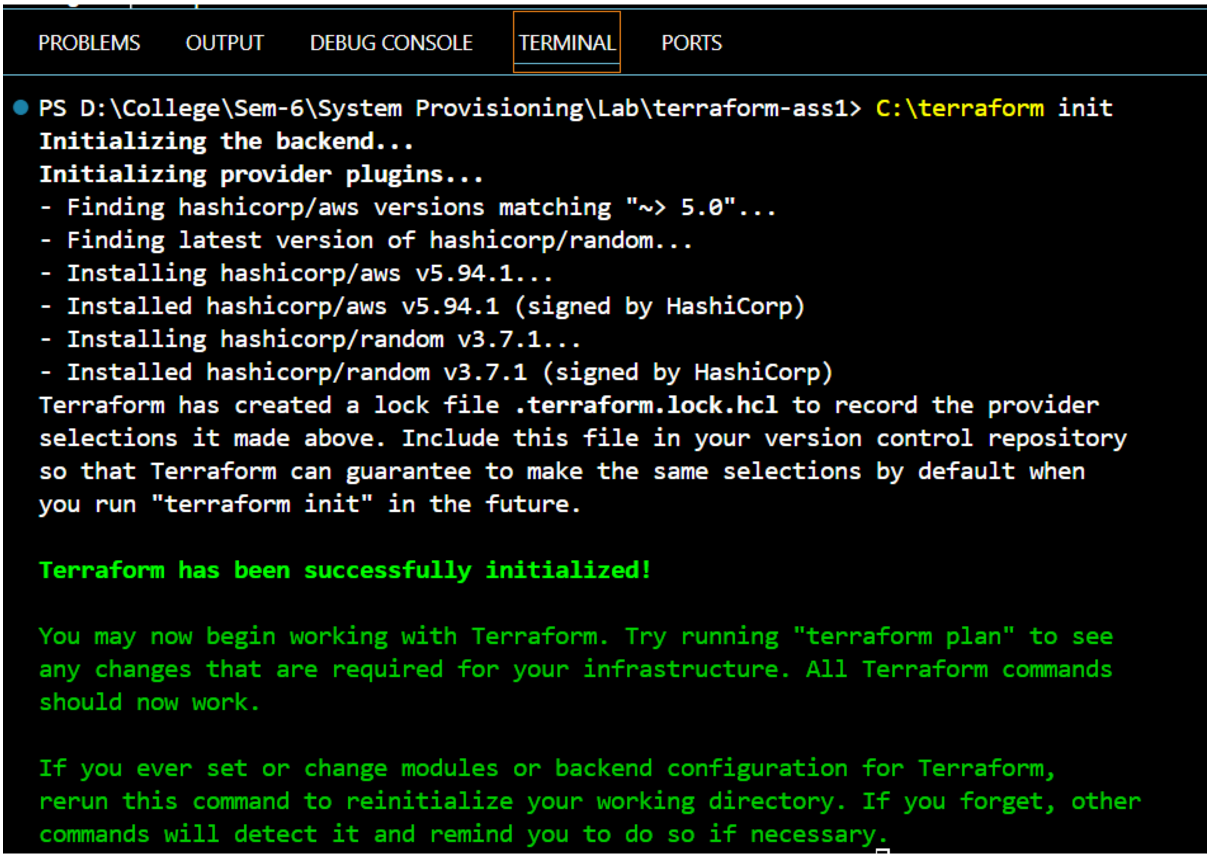
```
resource "aws_s3_bucket" "my_bucket" {
  bucket = "terraform-ass1-${random_id.bucket_id.hex}"
  tags = {
    Name        = "TerraformExampleBucket"
    Environment = "Dev"
  }
}

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

resource "random_id" "bucket_id" {
  byte_length = 8
}

data "aws_availability_zones" "available" {}
```

Step 4: Write the code for step 1,2 and 3 in a IaC terraform file and run terraform commands to execute these steps.

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}
provider "aws" {
  region = "ap-southeast-2"
}
resource "aws_instance" "ec2_instance_1" {
  ami           = "ami-09e143e99e8fa74f9"
  instance_type = "t2.micro"
  tags = {
    Name = "Terraform-EC2-1"
  }
}
resource "aws_instance" "ec2_instance_2" {
  ami           = "ami-09e143e99e8fa74f9"
  instance_type = "t2.micro"
  tags = {
    Name = "Terraform-EC2-2"
  }
}
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "Terraform-VPC"
  }
}
resource "aws_subnet" "public_subnet" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.1.0/24"
  availability_zone =
"${data.aws_availability_zones.available.names[0]}"
  map_public_ip_on_launch = true
  tags = {
    Name = "Terraform-Public-Subnet"
  }
}
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "Terraform-Internet-Gateway"
  }
```

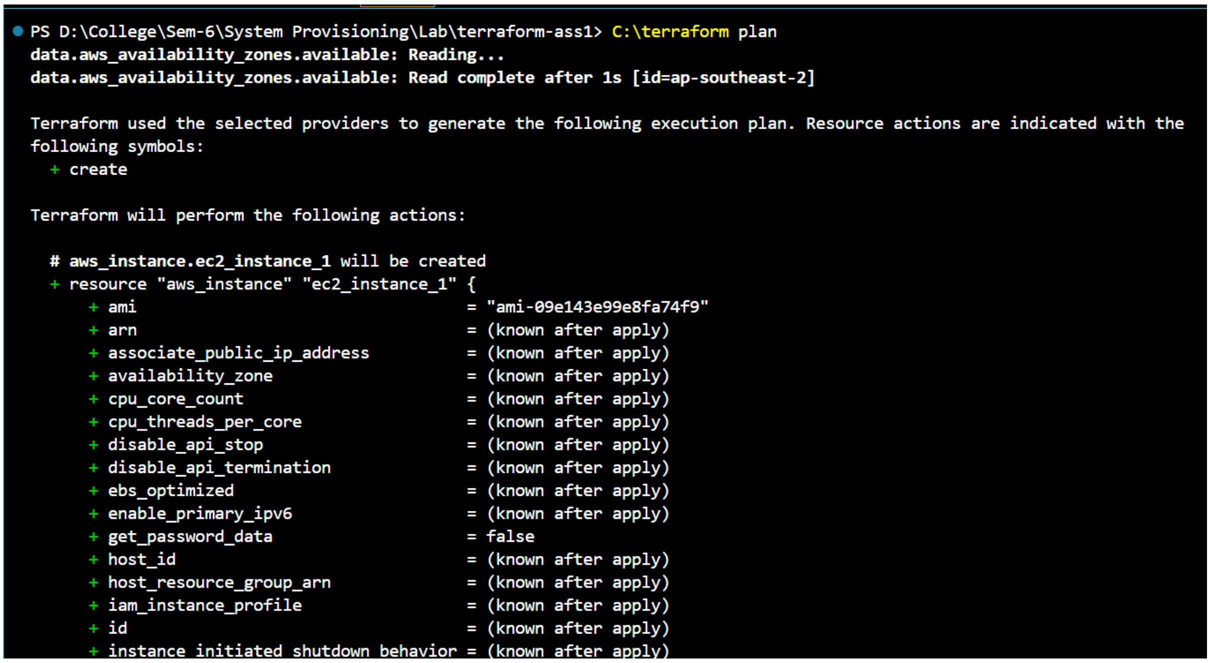
```
}
resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.main.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
  tags = {
    Name = "Terraform-Public-RouteTable"
  }
}
resource "aws_s3_bucket" "my_bucket" {
  bucket = "terraform-ass1-${random_id.bucket_id.hex}"
  tags = {
    Name        = "TerraformExampleBucket"
    Environment = "Dev"
  }
}
resource "aws_s3_bucket_acl" "my_bucket_acl" {
  bucket = aws_s3_bucket.my_bucket.id
  acl    = "private"
}
resource "random_id" "bucket_id" {
  byte_length = 8
}
data "aws_availability_zones" "available" {}
```

Terraform init:

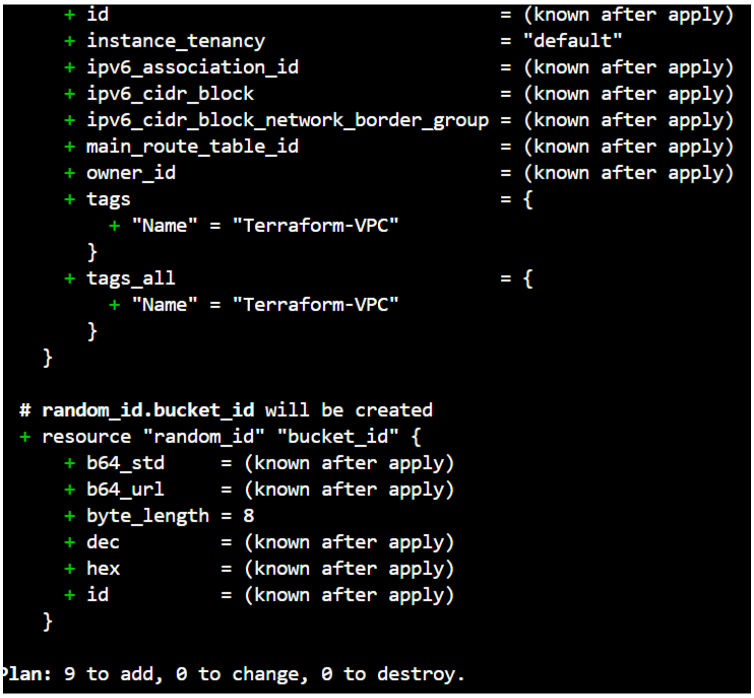


```
● PS D:\College\Sem-6\System Provisioning\Lab\terraform-ass1> C:\terraform init
  Initializing the backend...
  Initializing provider plugins...
  - Finding hashicorp/aws versions matching "~> 5.0"...
  - Finding latest version of hashicorp/random...
  - Installing hashicorp/aws v5.94.1...
  - Installed 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.
```

Terraform plan:

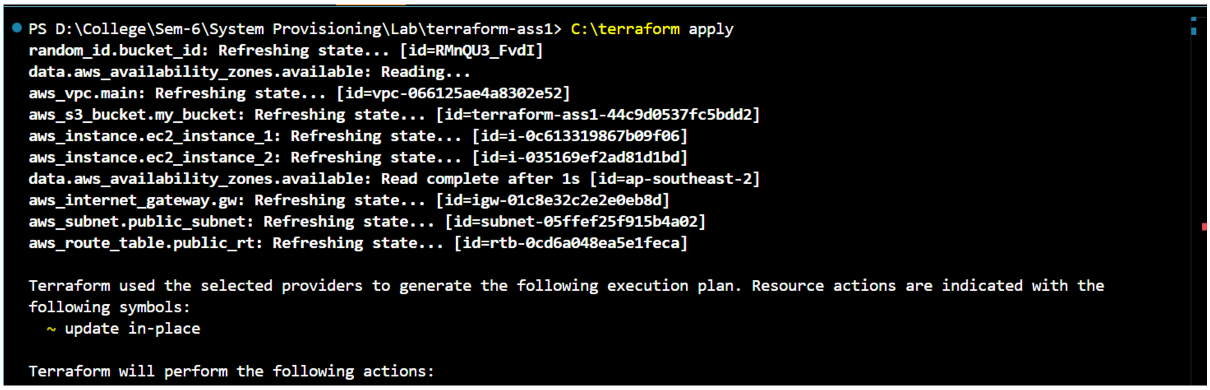


```
● PS D:\College\Sem-6\System Provisioning\Lab\terraform-ass1> C:\terraform plan
  data.aws_availability_zones.available: Reading...
  data.aws_availability_zones.available: Read complete after 1s [id=ap-southeast-2]

  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.ec2_instance_1 will be created
    + resource "aws_instance" "ec2_instance_1" {
        + ami                                  = "ami-09e143e99e8fa74f9"
        + 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)
```

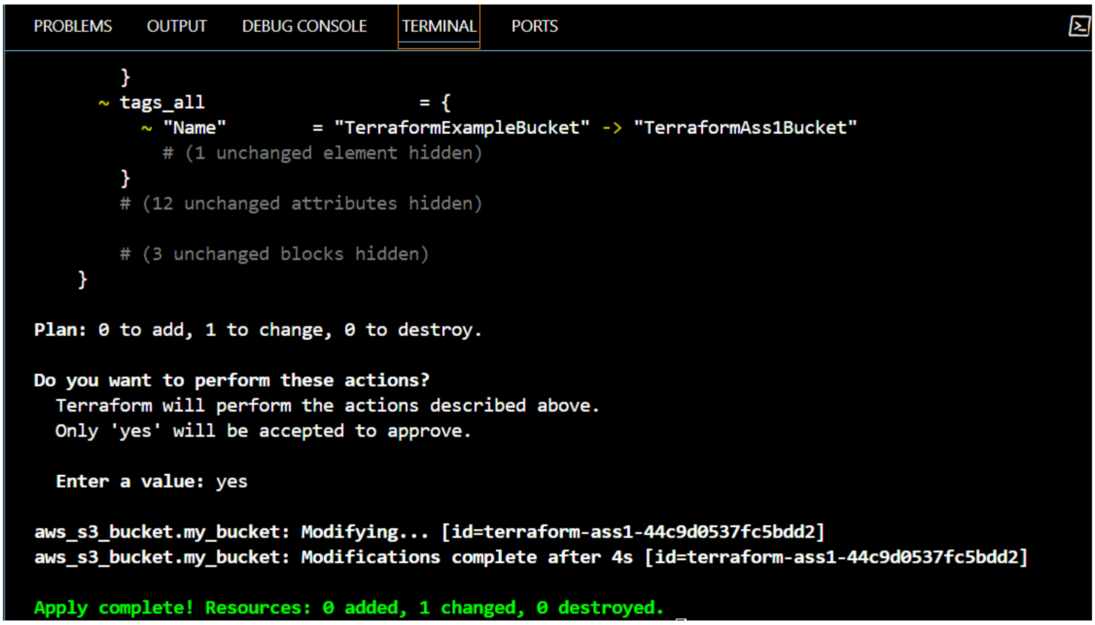


```
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "Terraform-VPC"
        }
      + tags_all                             = {
          + "Name" = "Terraform-VPC"
        }
    }

  # random_id.bucket_id will be created
  + resource "random_id" "bucket_id" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 8
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 9 to add, 0 to change, 0 to destroy.
```

Terraform apply:



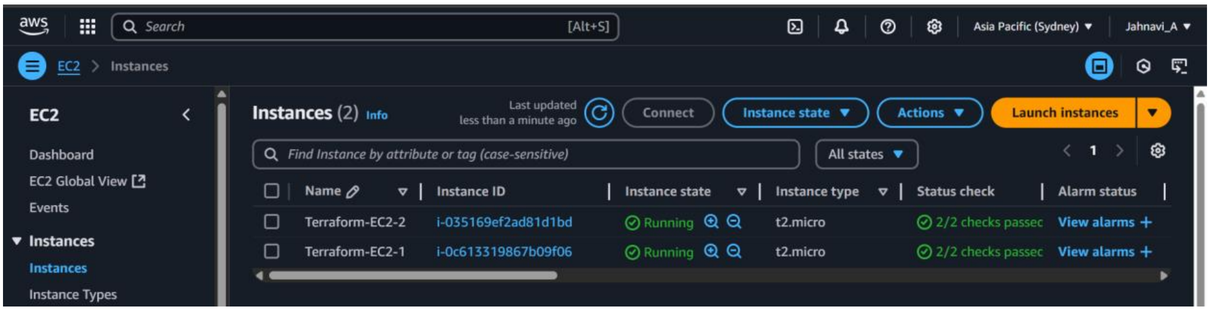
```
● PS D:\College\Sem-6\System Provisioning\Lab\terraform-ass1> C:\terraform apply
  random_id.bucket_id: Refreshing state... [id=RMnQU3_FvdI]
  data.aws_availability_zones.available: Reading...
  aws_vpc.main: Refreshing state... [id=vpc-066125ae4a8302e52]
  aws_s3_bucket.my_bucket: Refreshing state... [id=terraform-ass1-44c9d0537fc5bdd2]
  aws_instance.ec2_instance_1: Refreshing state... [id=i-0c613319867b09f06]
  aws_instance.ec2_instance_2: Refreshing state... [id=i-035169ef2ad81d1bd]
  data.aws_availability_zones.available: Read complete after 1s [id=ap-southeast-2]
  aws_internet_gateway.gw: Refreshing state... [id=igw-01c8e32c2e2e0eb8d]
  aws_subnet.public_subnet: Refreshing state... [id=subnet-05ffef25f915b4a02]
  aws_route_table.public_rt: Refreshing state... [id=rtb-0cd6a048ea5e1feca]

  Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
  following symbols:
    ~ update in-place

  Terraform will perform the following actions:
```



```
          }
        ~ tags_all                    = {
            ~ "Name"        = "TerraformExampleBucket" -> "TerraformAss1Bucket"
              # (1 unchanged element hidden)
          }
          # (12 unchanged attributes hidden)

          # (3 unchanged blocks hidden)
      }

Plan: 0 to add, 1 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

aws_s3_bucket.my_bucket: Modifying... [id=terraform-ass1-44c9d0537fc5bdd2]
aws_s3_bucket.my_bucket: Modifications complete after 4s [id=terraform-ass1-44c9d0537fc5bdd2]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

AWS Console Output:

Instance-

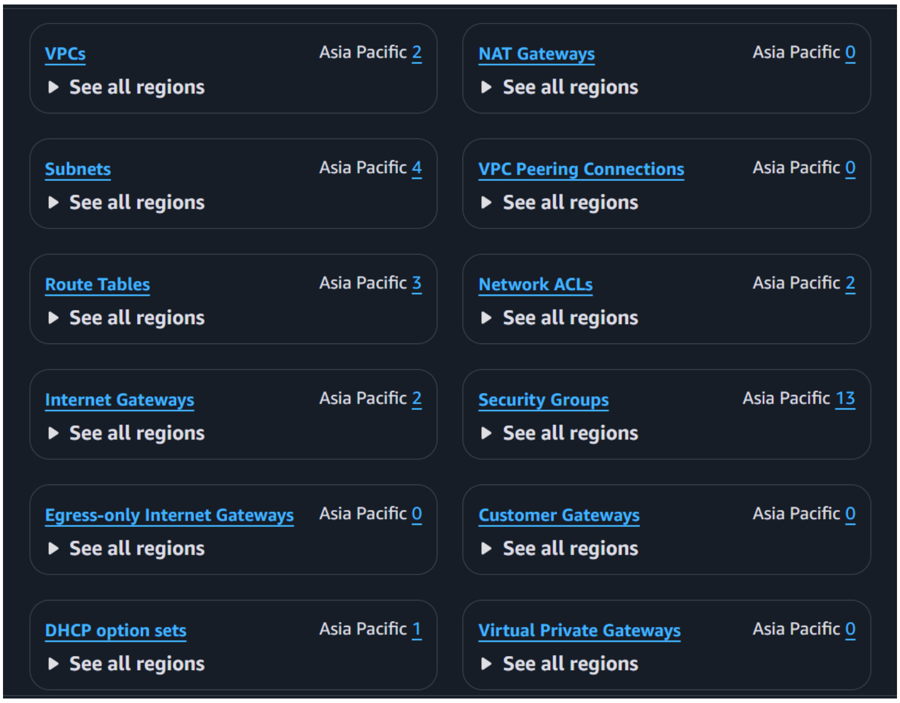


Instances (2) Info

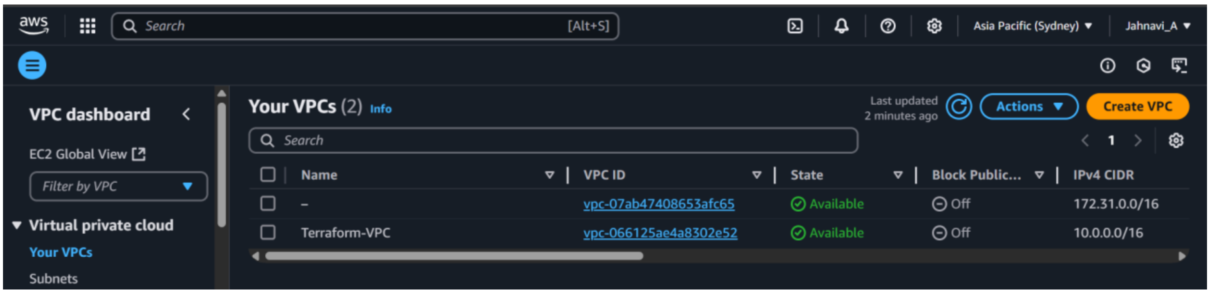
Find Instance by attribute or tag (case-sensitive)

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	Terraform-EC2-2	i-035169ef2ad81d1bd	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +
☐	Terraform-EC2-1	i-0c613319867b09f06	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +

VPN-



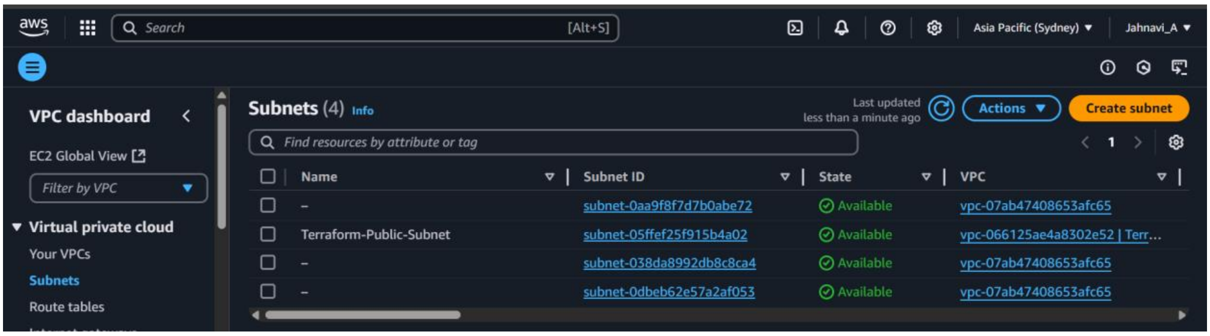
Resource	Asia Pacific
VPCs	2
NAT Gateways	0
Subnets	4
VPC Peering Connections	0
Route Tables	3
Network ACLs	2
Internet Gateways	2
Security Groups	13
Egress-only Internet Gateways	0
Customer Gateways	0
DHCP option sets	1
Virtual Private Gateways	0



Your VPCs (2) Info

Search

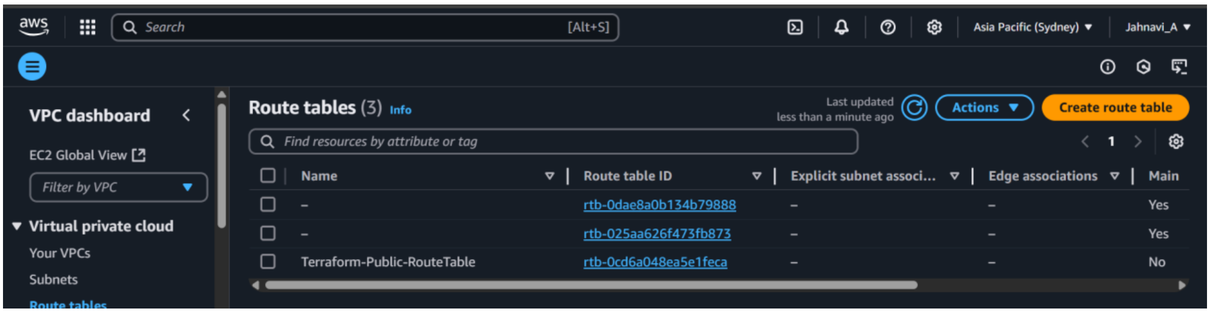
☐	Name	VPC ID	State	Block Public...	IPv4 CIDR
☐	–	vpc-07ab47408653afc65	✓ Available	⊖ Off	172.31.0.0/16
☐	Terraform-VPC	vpc-066125ae4a8302e52	✓ Available	⊖ Off	10.0.0.0/16



Subnets (4) Info

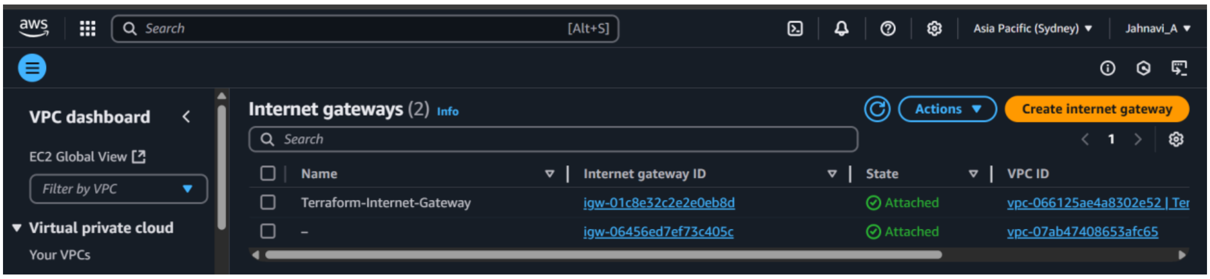
Find resources by attribute or tag

☐	Name	Subnet ID	State	VPC
☐	–	subnet-0aa9f8f7d7b0abe72	✓ Available	vpc-07ab47408653afc65
☐	Terraform-Public-Subnet	subnet-05ffef25f915b4a02	✓ Available	vpc-066125ae4a8302e52 | Terr...
☐	–	subnet-038da8992db8c8ca4	✓ Available	vpc-07ab47408653afc65
☐	–	subnet-0dbeb62e57a2af053	✓ Available	vpc-07ab47408653afc65



Route tables (3) Info

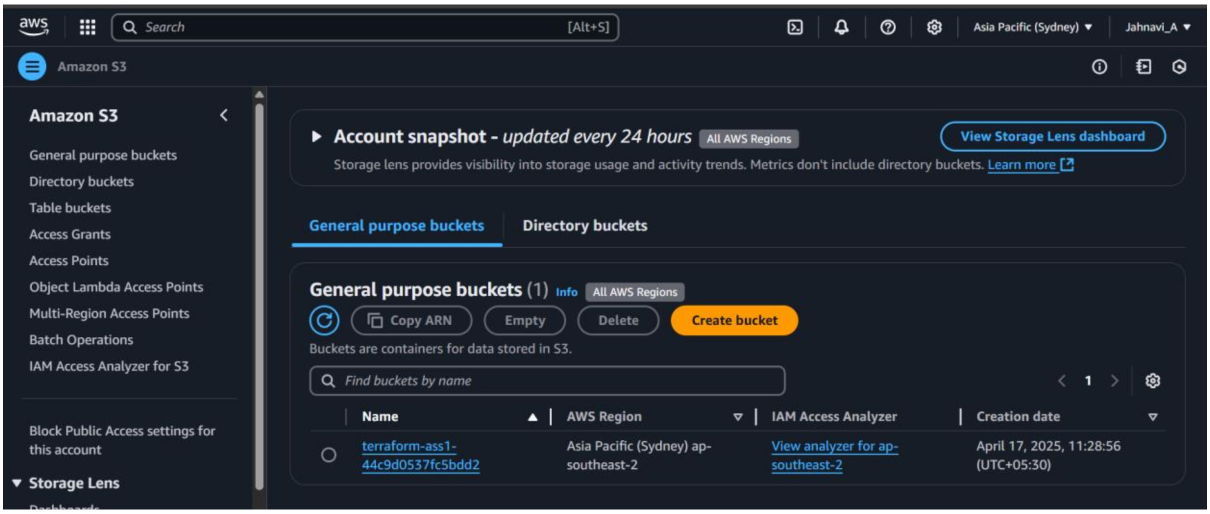
□	Name	Route table ID	Explicit subnet associ...	Edge associations	Main
□	–	rtb-0dae8a0b134b79888	–	–	Yes
□	–	rtb-025aa626f473fb873	–	–	Yes
□	Terraform-Public-RouteTable	rtb-0cd6a048ea5e1feca	–	–	No



Internet gateways (2) Info

□	Name	Internet gateway ID	State	VPC ID
□	Terraform-Internet-Gateway	igw-01c8e32c2e2e0eb8d	✓ Attached	vpc-066125ae4a8302e52 | Ter
□	–	igw-06456ed7ef73c405c	✓ Attached	vpc-07ab47408653afc65

S3 Bucket-



▶ Account snapshot - *updated every 24 hours* All AWS Regions

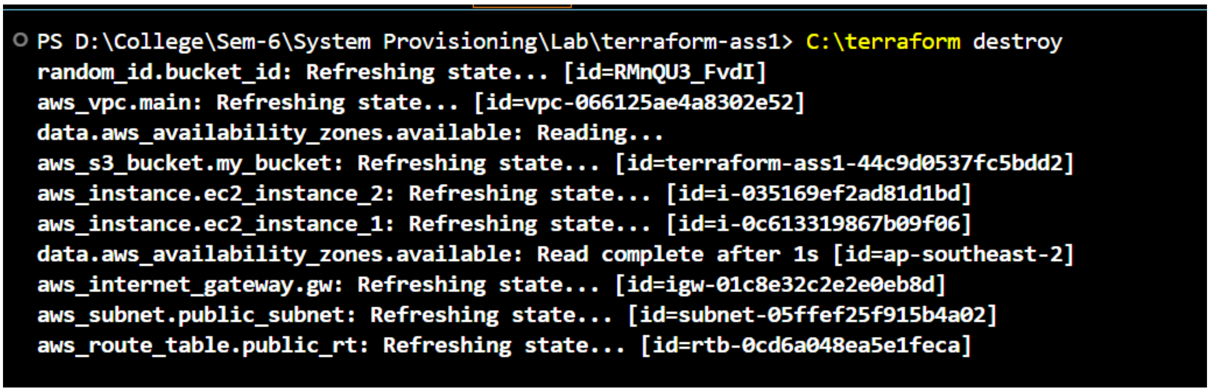
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (1) Info All AWS Regions

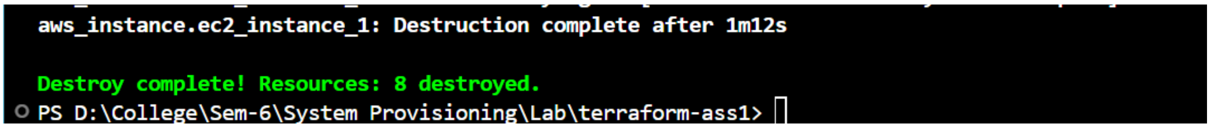
Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	terraform-ass1-44c9d0537fc5bdd2	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	April 17, 2025, 11:28:56 (UTC+05:30)

Terraform destroy:



```
PS D:\College\Sem-6\System Provisioning\Lab\terraform-ass1> C:\terraform destroy
random_id.bucket_id: Refreshing state... [id=RMnQU3_FvdI]
aws_vpc.main: Refreshing state... [id=vpc-066125ae4a8302e52]
data.aws_availability_zones.available: Reading...
aws_s3_bucket.my_bucket: Refreshing state... [id=terraform-ass1-44c9d0537fc5bdd2]
aws_instance.ec2_instance_2: Refreshing state... [id=i-035169ef2ad81d1bd]
aws_instance.ec2_instance_1: Refreshing state... [id=i-0c613319867b09f06]
data.aws_availability_zones.available: Read complete after 1s [id=ap-southeast-2]
aws_internet_gateway.gw: Refreshing state... [id=igw-01c8e32c2e2e0eb8d]
aws_subnet.public_subnet: Refreshing state... [id=subnet-05ffef25f915b4a02]
aws_route_table.public_rt: Refreshing state... [id=rtb-0cd6a048ea5e1feca]
```



```
aws_instance.ec2_instance_1: Destruction complete after 1m12s

Destroy complete! Resources: 8 destroyed.
PS D:\College\Sem-6\System Provisioning\Lab\terraform-ass1> 
```