SPCM - LAB

6th Sem

Submitted To:

Dr. Hitesh Kumar Sharma

School of Computer Science

Submitted By:

Ansh Tyagi

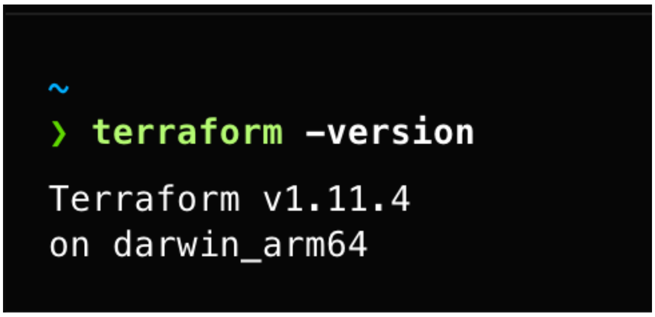
R2142220033

500105272

Batch 2-NH

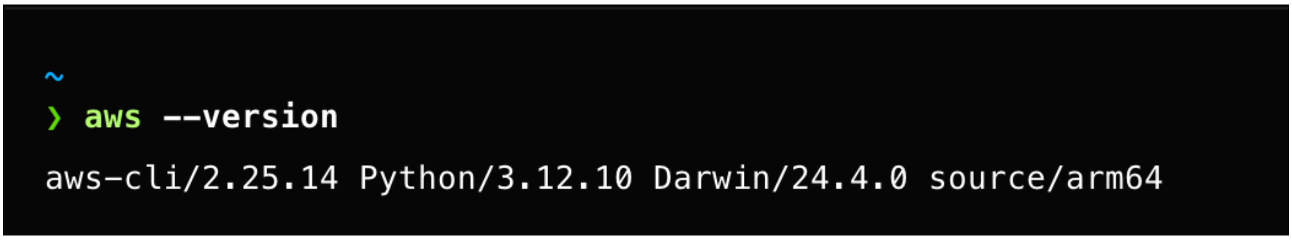
Step 1: Set Up Your Environment

1. Install Terraform



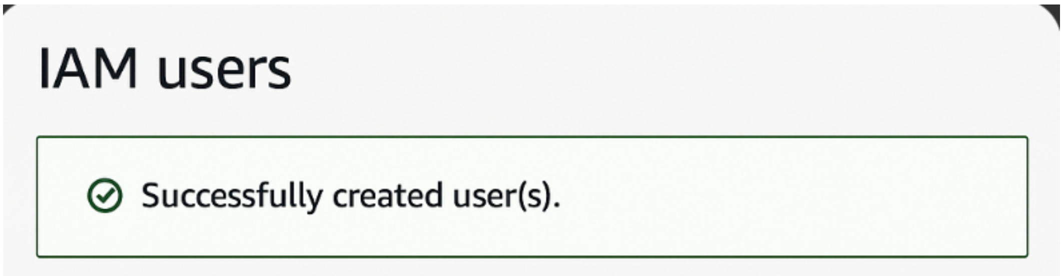
```
~
❯ terraform -version
Terraform v1.11.4
on darwin_arm64
```

2. Install AWS CLI



```
~
❯ aws --version
aws-cli/2.25.14 Python/3.12.10 Darwin/24.4.0 source/arm64
```

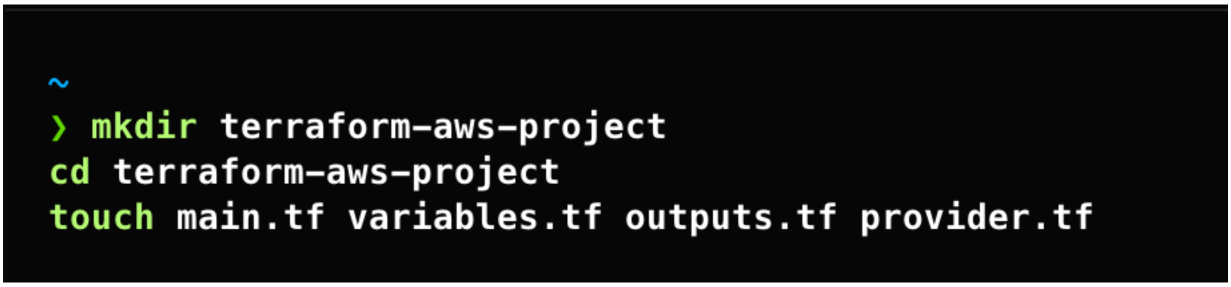
3. Create an IAM User with programmatic access and Administrator permissions (for testing purposes).



IAM users

Successfully created user(s).

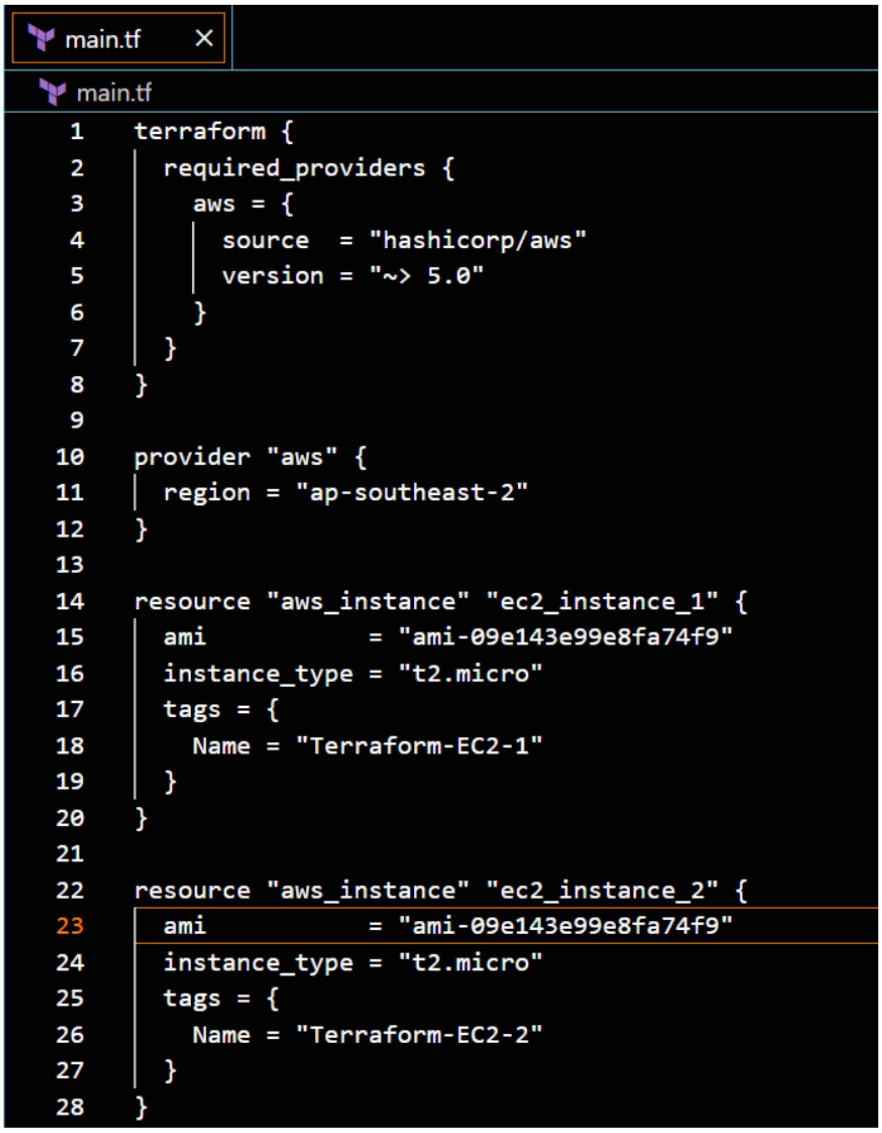
Step 2: Create a Project Directory



```
~
❯ mkdir terraform-aws-project
cd terraform-aws-project
touch main.tf variables.tf outputs.tf provider.tf
```

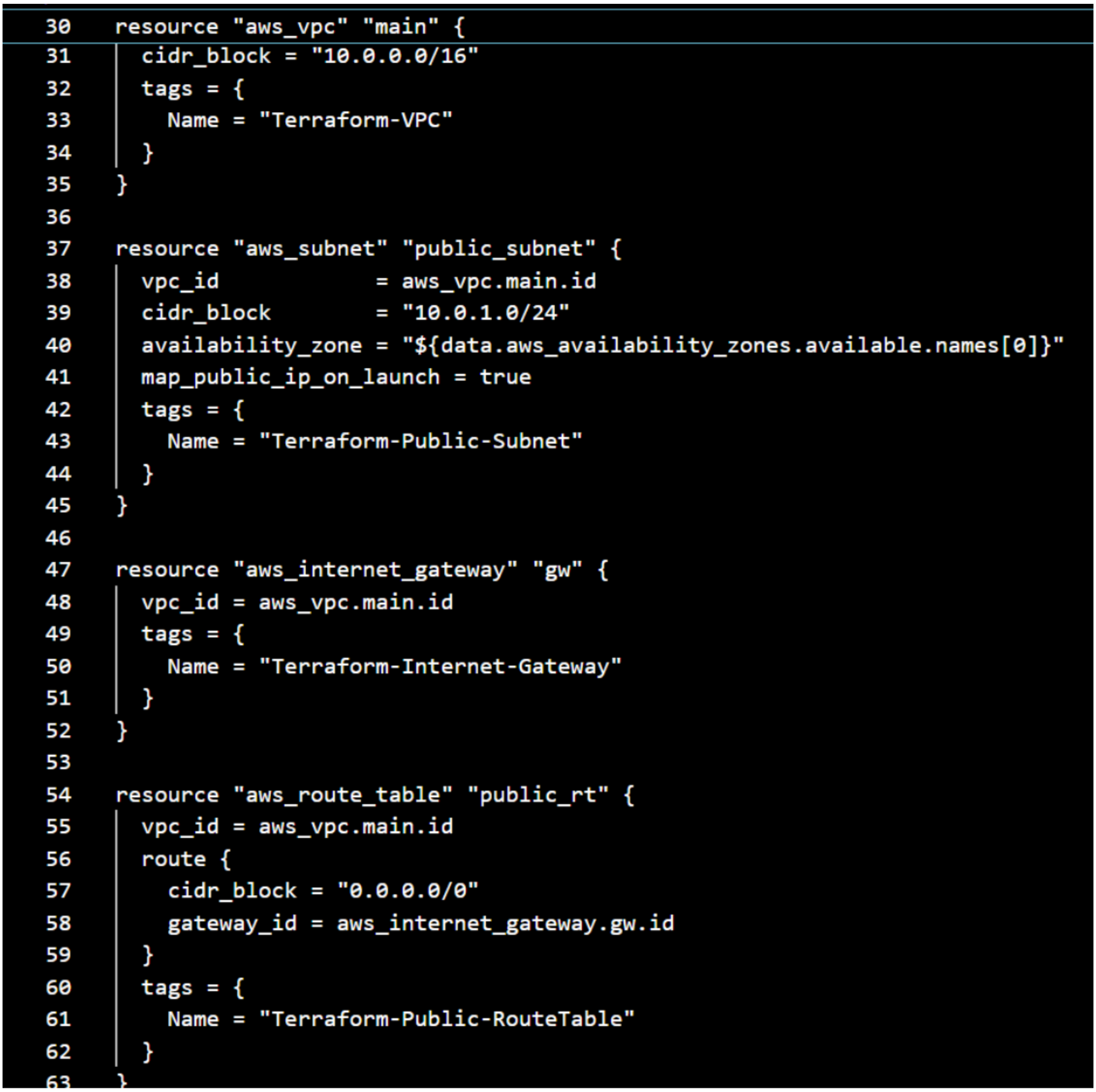
Step 3: Write Terraform Code

1. Create two T2 Micro EC2 Instances.



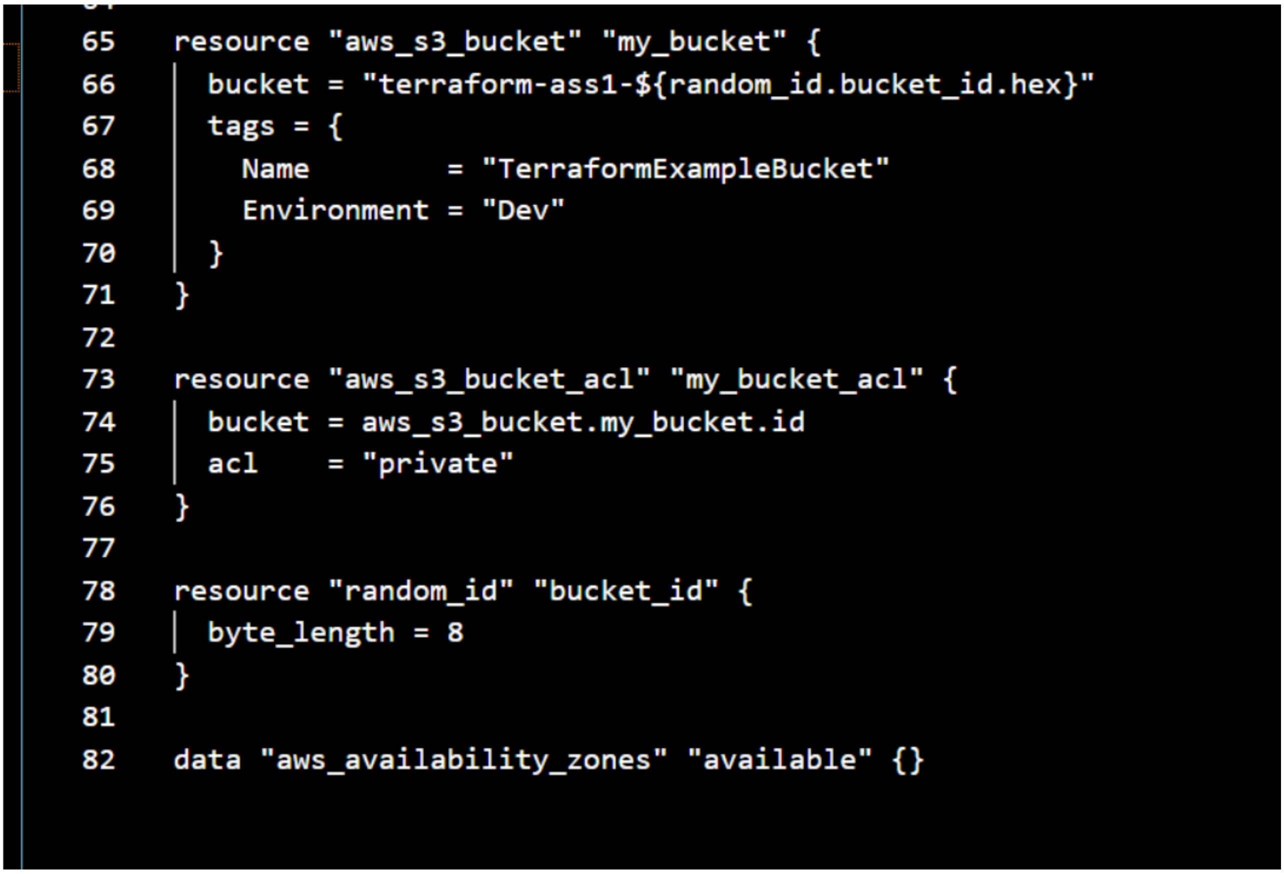
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

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"
  }
}
```

2: Create a VPN on AWS



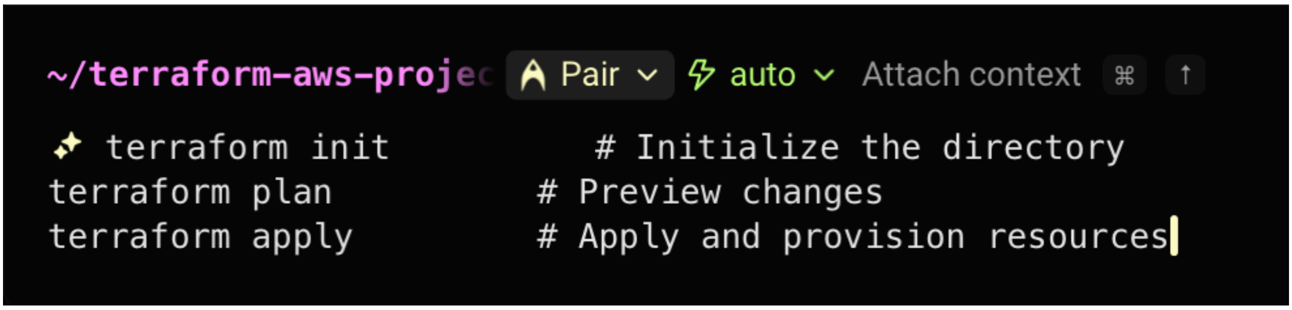
```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "Terraform-VPC"
  }
}

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"
  }
}
```

3: Create a S3 Bucket

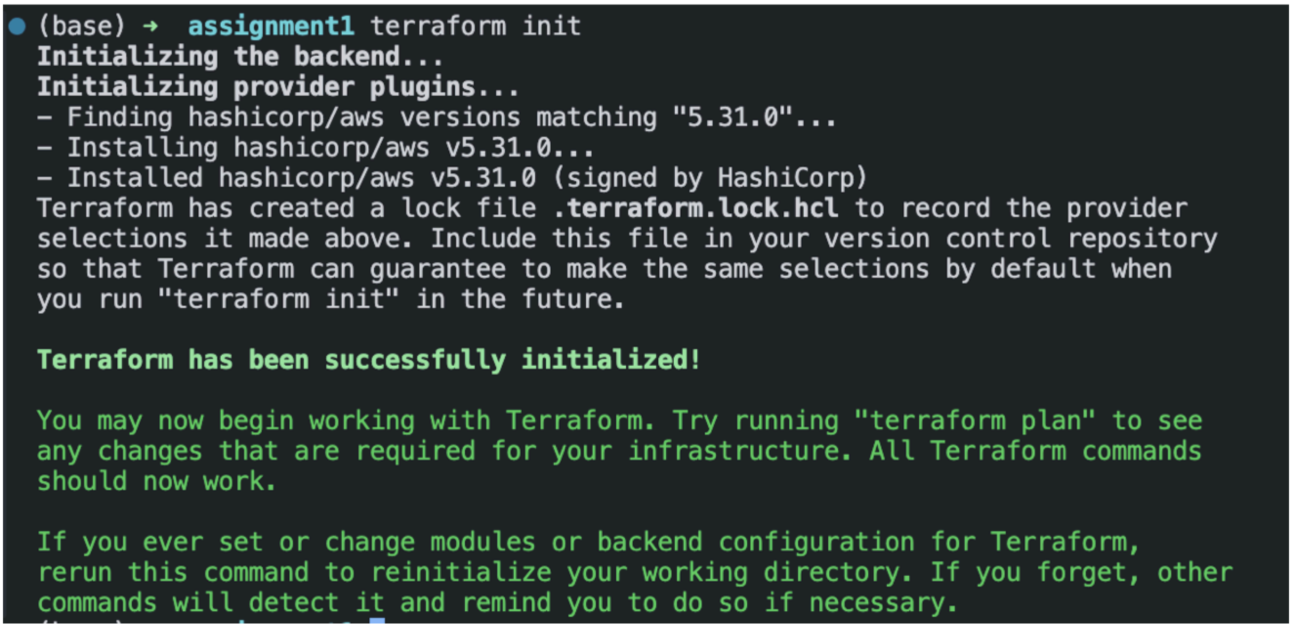


```
resource "aws_s3_bucket" "my_bucket" {
  bucket = "terraform-ass1-${random_id.bucket_id.hex}"
  tags = {
    Name        = "TerraformExampleBucket"
    Environment = "Dev"
  }
}

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" {}
```

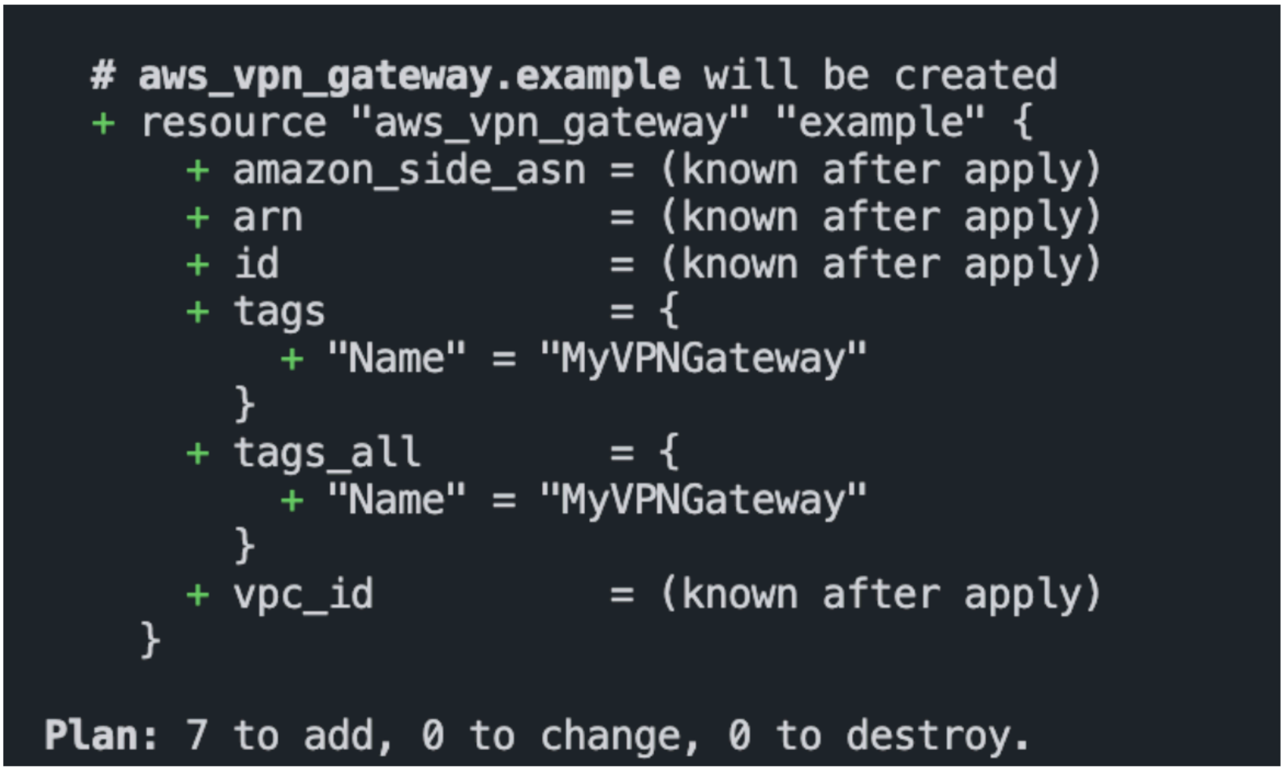
Step 4: Run Terraform Commands



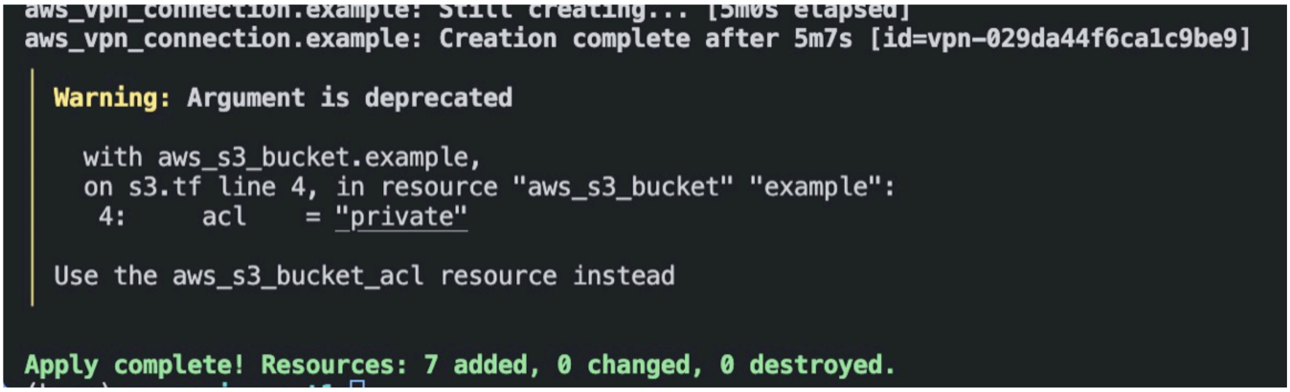
```
~/terraform-aws-projec
 terraform init          # Initialize the directory
terraform plan          # Preview changes
terraform apply         # Apply and provision resources
```



```
(base) → assignment1 terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.
```



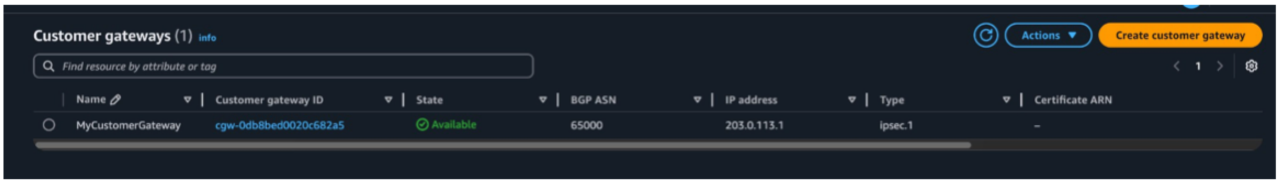
```
  # aws_vpn_gateway.example will be created
  + resource "aws_vpn_gateway" "example" {
      + amazon_side_asn = (known after apply)
      + arn             = (known after apply)
      + id              = (known after apply)
      + tags            = {
          + "Name" = "MyVPNGateway"
        }
      + tags_all        = {
          + "Name" = "MyVPNGateway"
        }
      + vpc_id          = (known after apply)
    }

Plan: 7 to add, 0 to change, 0 to destroy.
```



```
aws_vpn_connection.example: Still creating... [5m0s elapsed]
aws_vpn_connection.example: Creation complete after 5m7s [id=vpn-029da44f6ca1c9be9]
│
│ Warning: Argument is deprecated
│
│   with aws_s3_bucket.example,
│   on s3.tf line 4, in resource "aws_s3_bucket" "example":
│    4:     acl    = "private"
│
│ Use the aws_s3_bucket_acl resource instead
│

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
```

Step 5: SHOW OUTPUTS

1. Customer Gateway



Customer gateways (1)

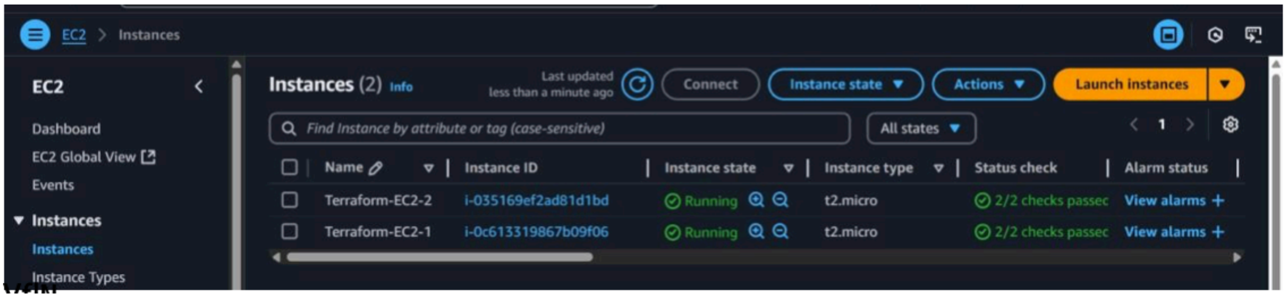
	Name	Customer gateway ID	State	BGP ASN	IP address	Type	Certificate ARN
○	MyCustomerGateway	cgw-0db8bed0020c682a5	Available	65000	203.0.113.1	ipsec.1	–

2. EC2 Instances

Dashboard Location: EC2 > Instances

What to check:

- Two instances named **EC2-Instance-1** and **EC2-Instance-2**
- State should be **running**
- Instance type should be **t2.micro**

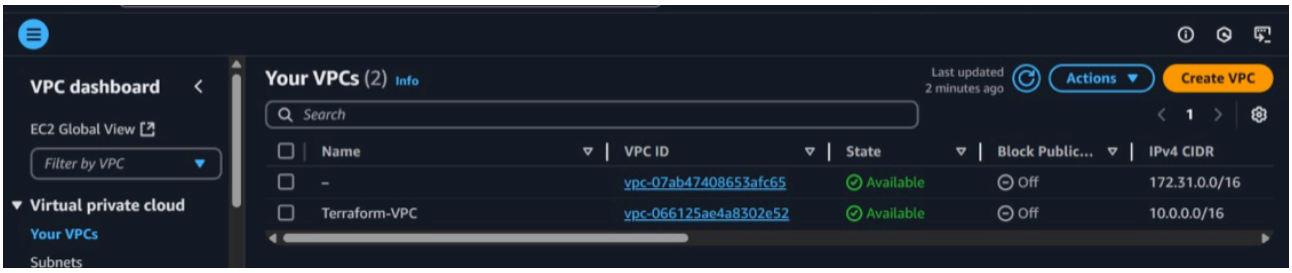


Instances (2)

Last updated less than a minute ago

	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 +

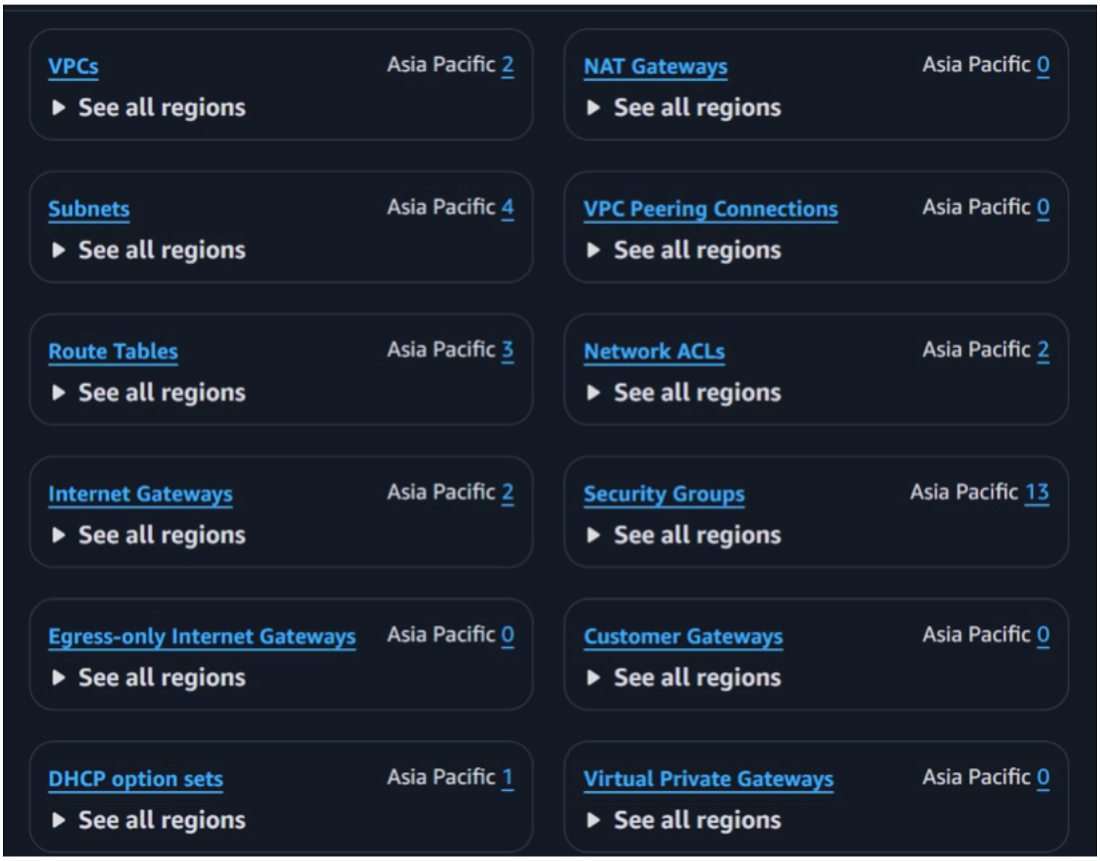
3. Vpc



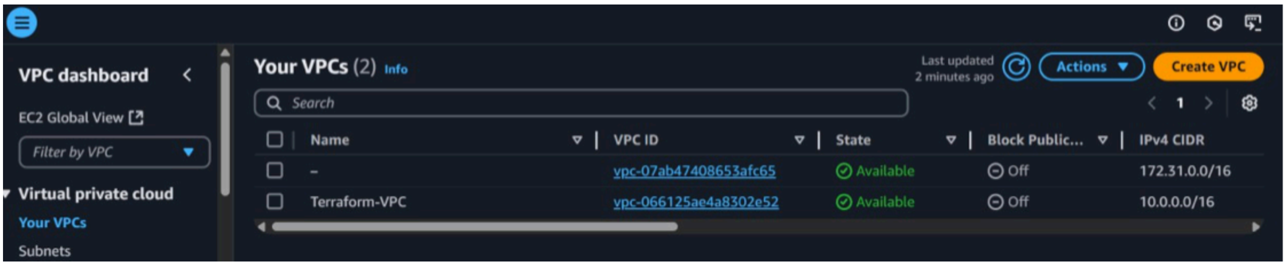
Your VPCs (2)

Last updated 2 minutes ago

	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



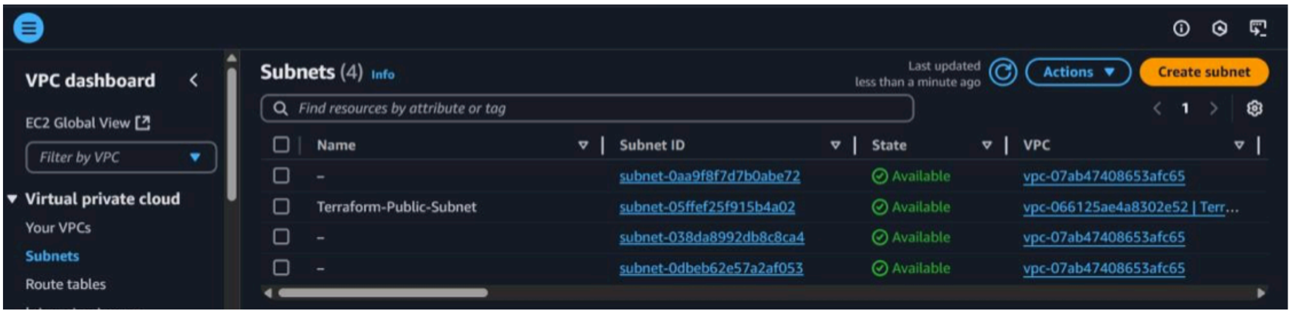
VPCs Asia Pacific 2 ▶ See all regions	NAT Gateways Asia Pacific 0 ▶ See all regions
Subnets Asia Pacific 4 ▶ See all regions	VPC Peering Connections Asia Pacific 0 ▶ See all regions
Route Tables Asia Pacific 3 ▶ See all regions	Network ACLs Asia Pacific 2 ▶ See all regions
Internet Gateways Asia Pacific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



Your VPCs (2) Info

Last updated 2 minutes ago

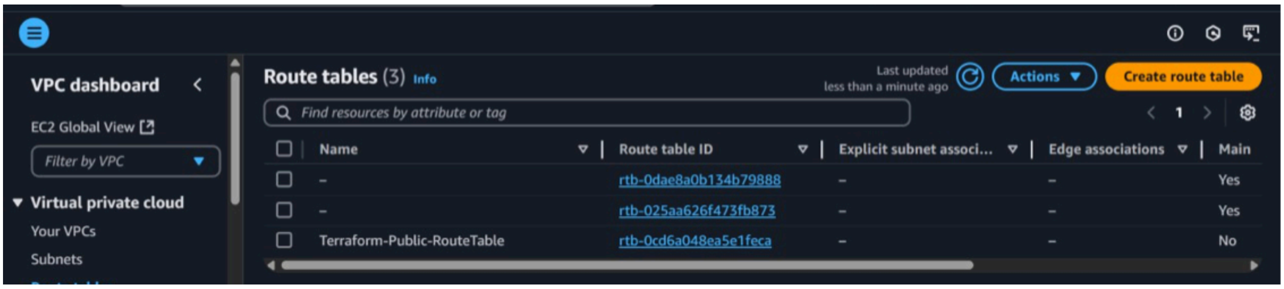
☐	Name	VPC ID	State	Block Public...	IPv4 CIDR
☐	–	vpc-07ab47408653afc65	✓ Available	Off	172.31.0.0/16
☐	Terraform-VPC	vpc-066125ae4a8302e52	✓ Available	Off	10.0.0.0/16



Subnets (4) Info

Last updated less than a minute ago

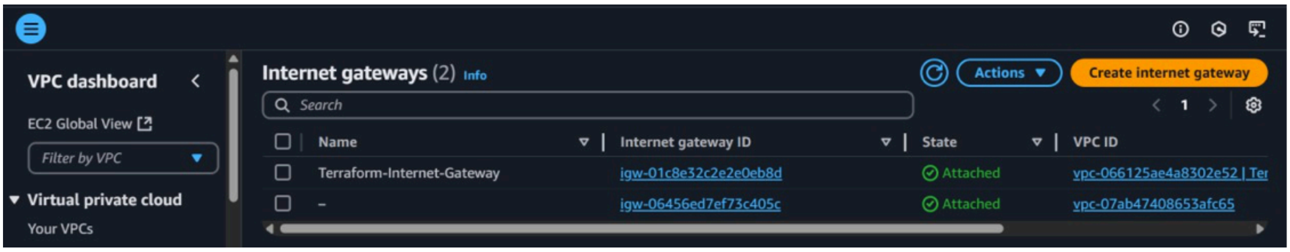
☐	Name	Subnet ID	State	VPC
☐	–	subnet-0aa9f8f7d7b0abe72	✓ Available	vpc-07ab47408653afc65
☐	Terraform-Public-Subnet	subnet-05ffef25f915b4a02	✓ Available	vpc-066125ae4a8302e52 | Terr...
☐	–	subnet-038da8992db8c8ca4	✓ Available	vpc-07ab47408653afc65
☐	–	subnet-0dbeb62e57a2af053	✓ Available	vpc-07ab47408653afc65



Route tables (3) Info

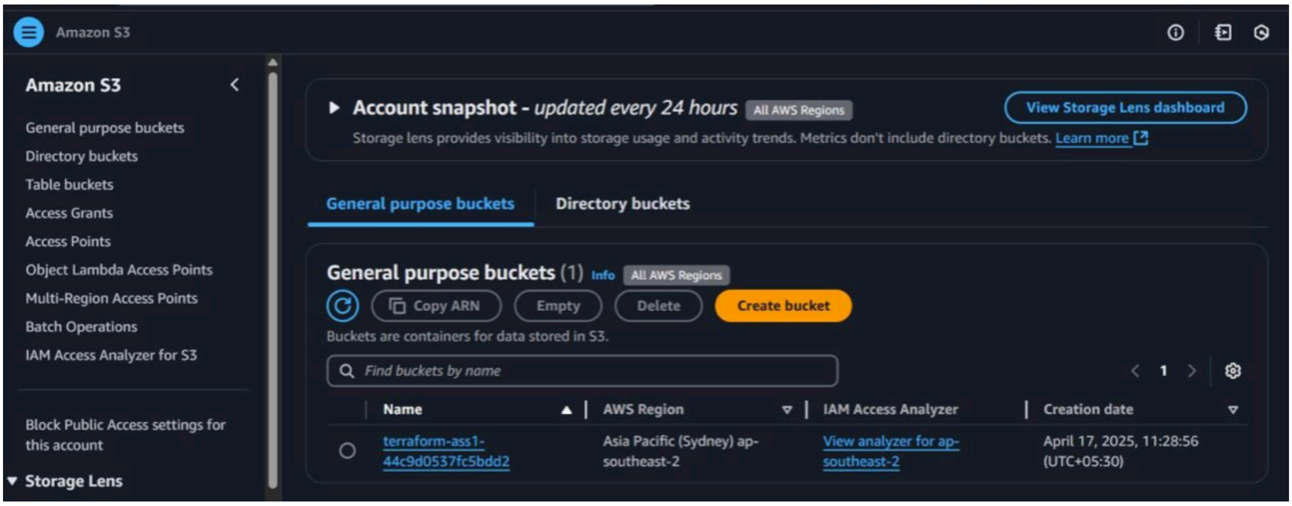
Last updated less than a minute ago

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



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

3. S3 Bucket



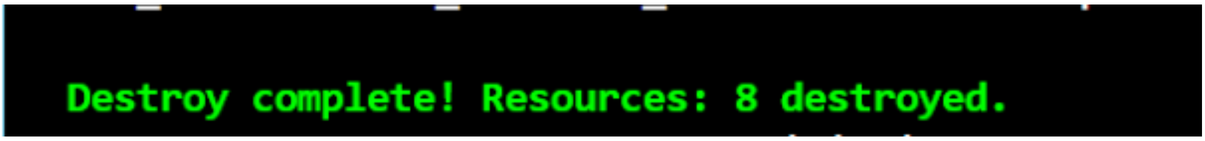
Account snapshot - *updated every 24 hours* All AWS Regions

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

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)

Step 5: Terraform Destroy



```
Destroy complete! Resources: 8 destroyed.
```