UPES

UNIVERSITY OF TOMORROW

System Provisioning & Configuration Management

ASSIGNMENT-1

SUBMITTED BY:

NAME: BHAVYA JAIN

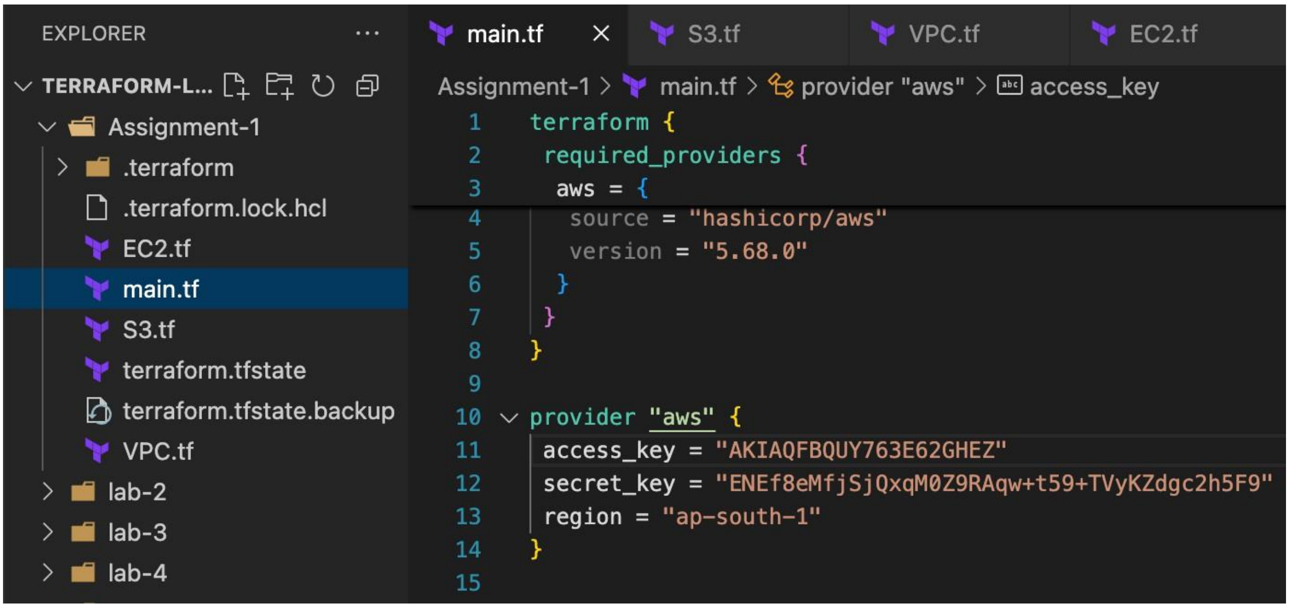
SAP ID: 500109495

BATCH: 2 (DevOps)

ROLL NO: R2142221391

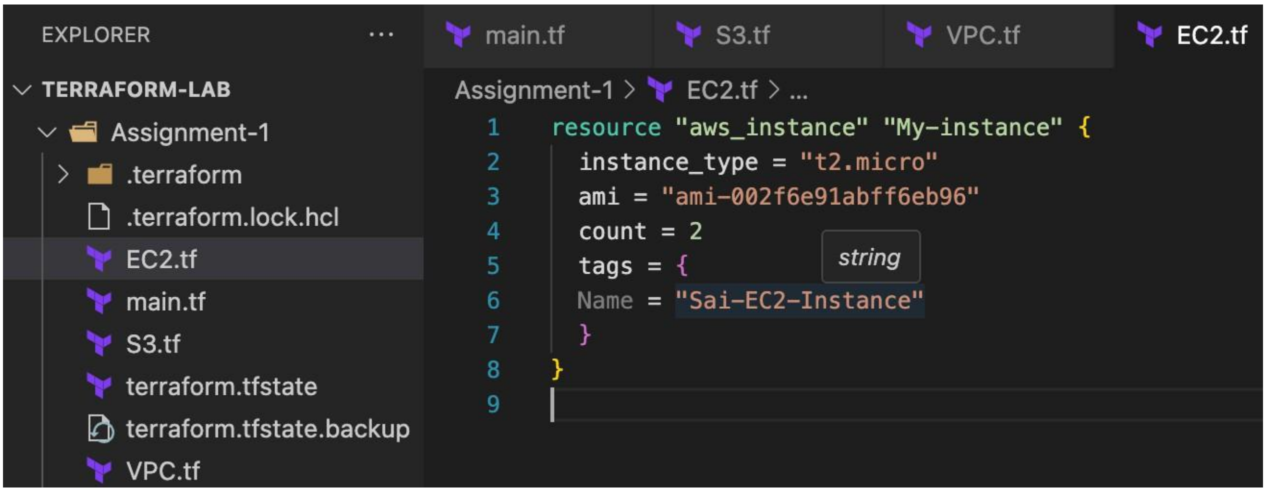
Write Terraform script to do perform following tasks on AWS cloud Platform

Step 1: Create Terraform Configuration File (main.tf):



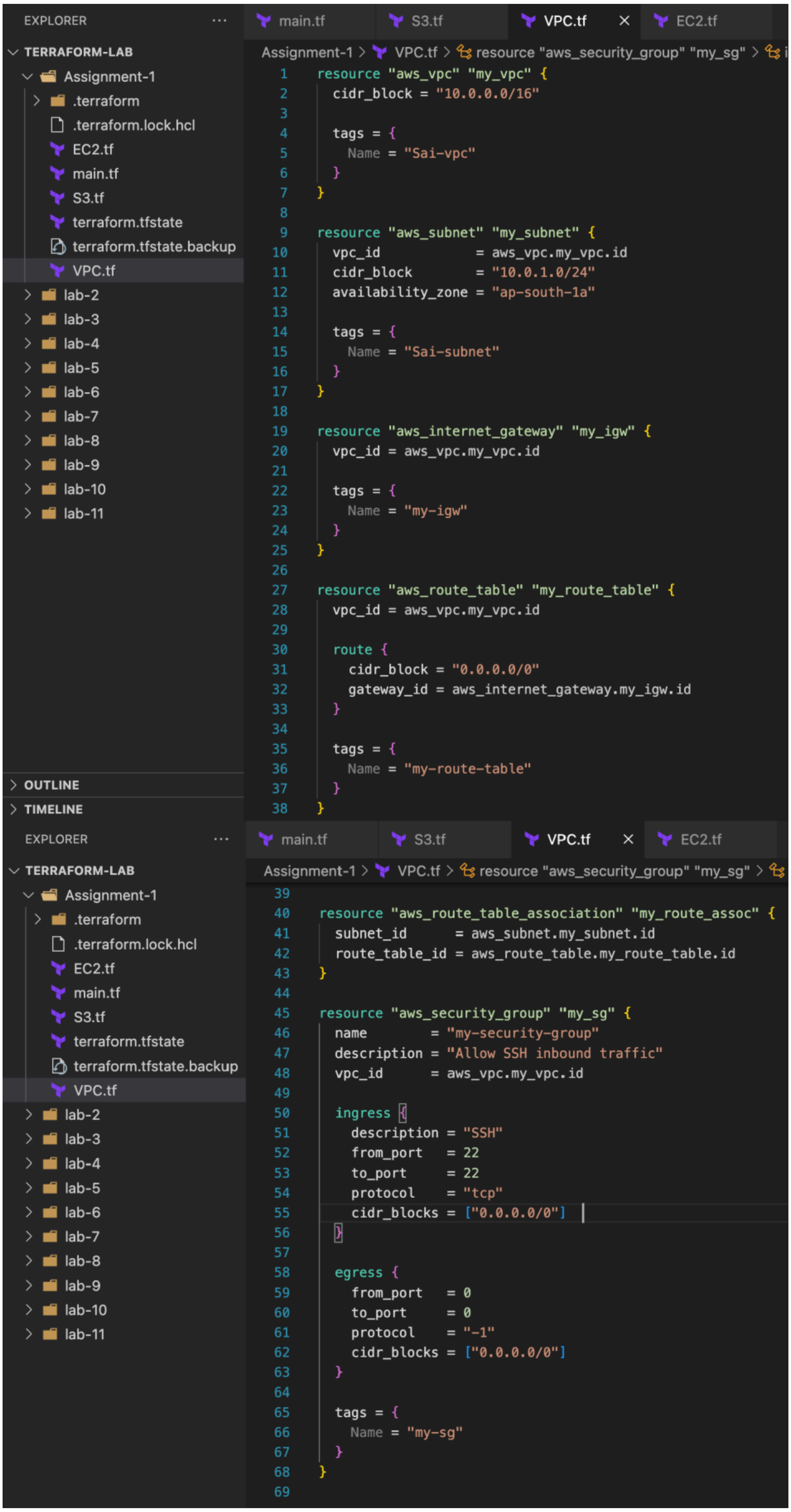
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.68.0"
    }
  }
}

provider "aws" {
  access_key = "AKIAQFBQUY763E62GHEZ"
  secret_key = "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"
  region = "ap-south-1"
}
```

Step 2: To create two T2 Micro EC2 Instances create Ec2.tf :



```
resource "aws_instance" "My-instance" {
  instance_type = "t2.micro"
  ami = "ami-002f6e91abff6eb96"
  count = 2
  tags = {
  Name = "Sai-EC2-Instance"
  }
}
```

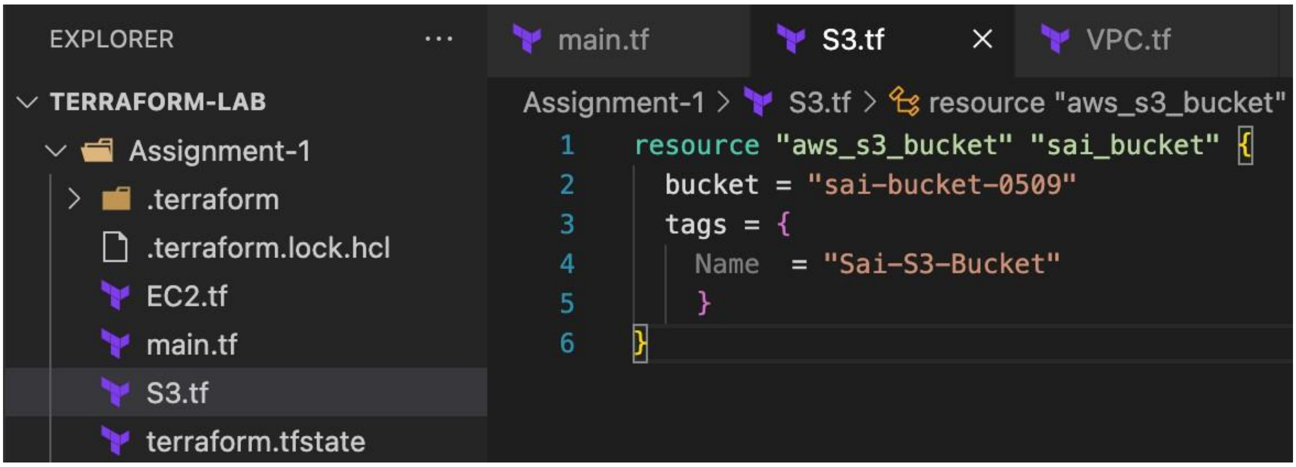
Step 3: To Create a VPC on AWS create VPC.tf :



```hcl
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "Sai-vpc"
  }
}

resource "aws_subnet" "my_subnet" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "ap-south-1a"

  tags = {
    Name = "Sai-subnet"
  }
}

resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id

  tags = {
    Name = "my-igw"
  }
}

resource "aws_route_table" "my_route_table" {
  vpc_id = aws_vpc.my_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.my_igw.id
  }

  tags = {
    Name = "my-route-table"
  }
}
```

```hcl

resource "aws_route_table_association" "my_route_assoc" {
  subnet_id      = aws_subnet.my_subnet.id
  route_table_id = aws_route_table.my_route_table.id
}

resource "aws_security_group" "my_sg" {
  name        = "my-security-group"
  description = "Allow SSH inbound traffic"
  vpc_id      = aws_vpc.my_vpc.id

  ingress {
    description = "SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "my-sg"
  }
}

```

Step 4: To Create a S3 Bucket create S3.tf :

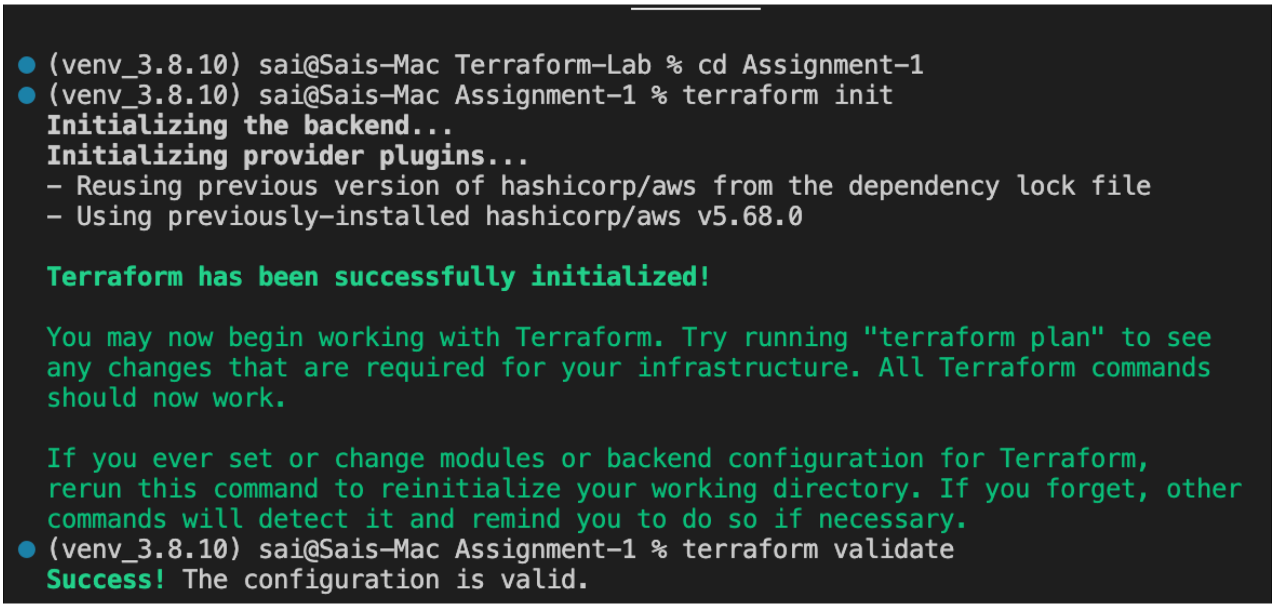


```
resource "aws_s3_bucket" "sai_bucket" {
  bucket = "sai-bucket-0509"
  tags = {
    Name  = "Sai-S3-Bucket"
    }
}
```

Step 5: Initialize Terraform

Run the following command to initialize your Terraform working directory:

Terraform init

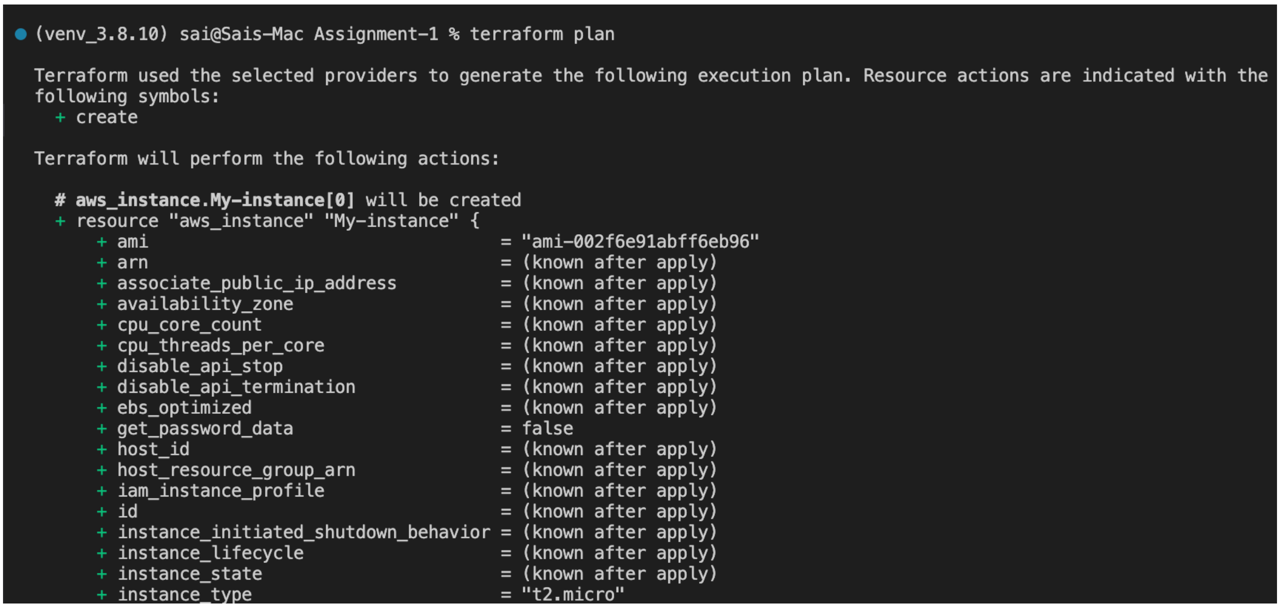


```
● (venv_3.8.10) sai@Sais-Mac Terraform-Lab % cd Assignment-1
● (venv_3.8.10) sai@Sais-Mac Assignment-1 % terraform init
  Initializing the backend...
  Initializing provider plugins...
  - Reusing previous version of hashicorp/aws from the dependency lock file
  - Using previously-installed hashicorp/aws v5.68.0

  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.
● (venv_3.8.10) sai@Sais-Mac Assignment-1 % terraform validate
  Success! The configuration is valid.
```

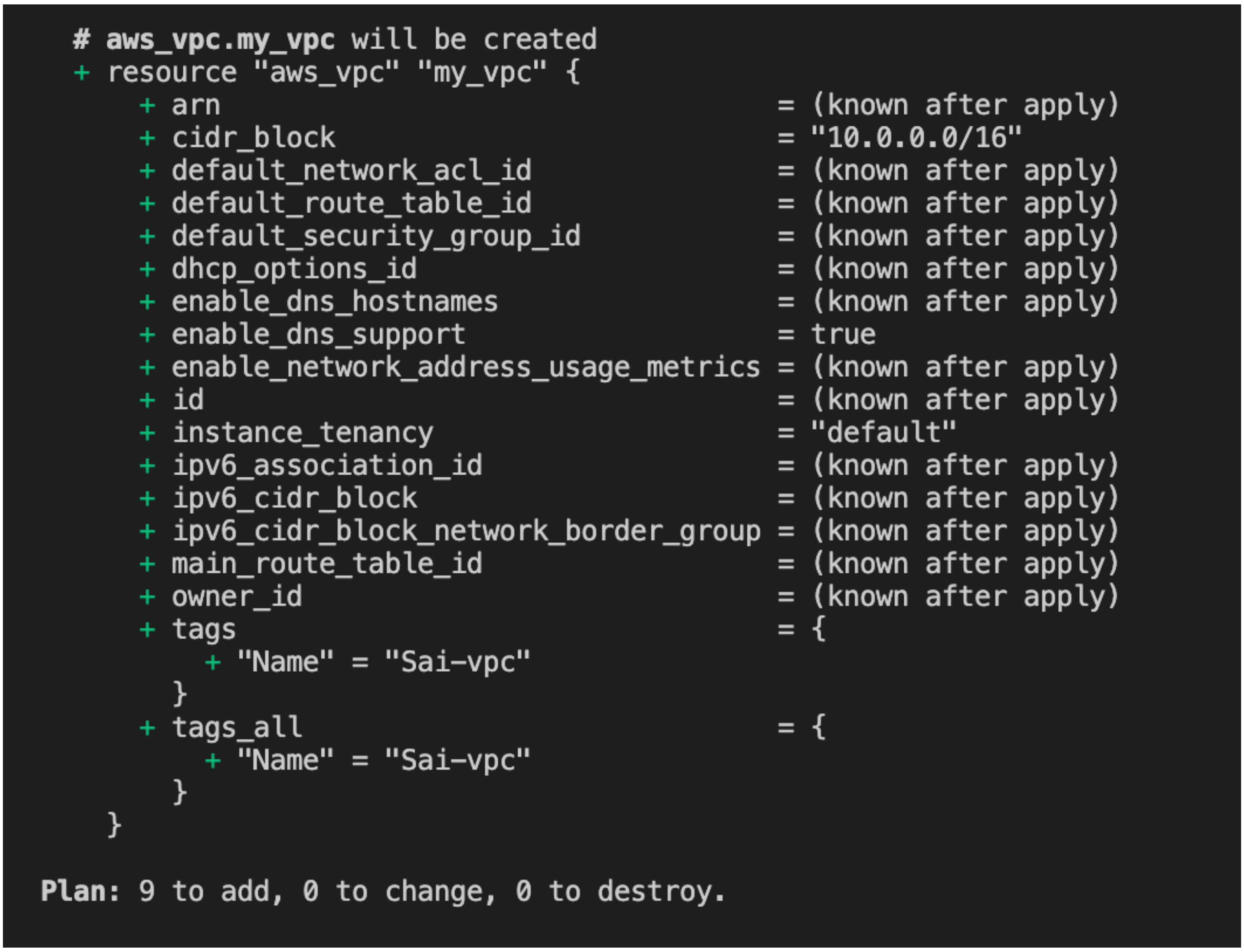
Step 6: Review Plan

Run the following command to see what Terraform will do:

Terraform plan



```
● (venv_3.8.10) sai@Sais-Mac Assignment-1 % 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.My-instance[0] will be created
    + resource "aws_instance" "My-instance" {
        + 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)
        + 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"
```

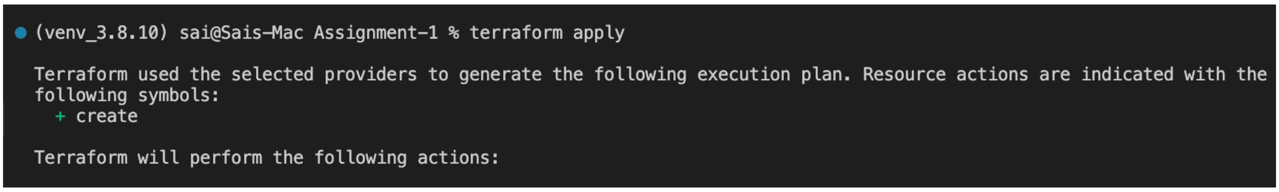


```
  # aws_vpc.my_vpc will be created
  + resource "aws_vpc" "my_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (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" = "Sai-vpc"
        }
      + tags_all                             = {
          + "Name" = "Sai-vpc"
        }
    }

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

Step 7: Apply Changes:

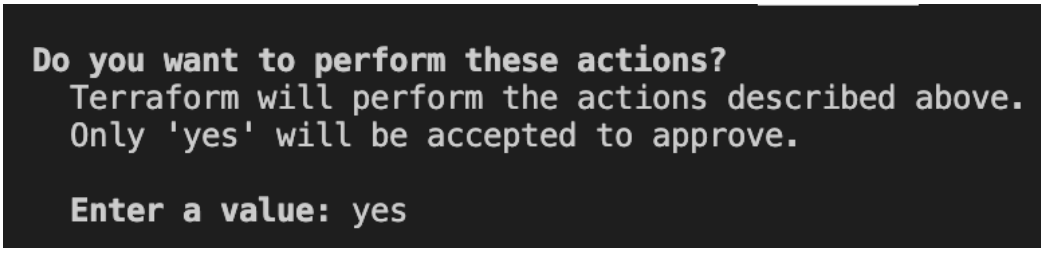
Apply the changes to create the AWS resources:

Terraform apply



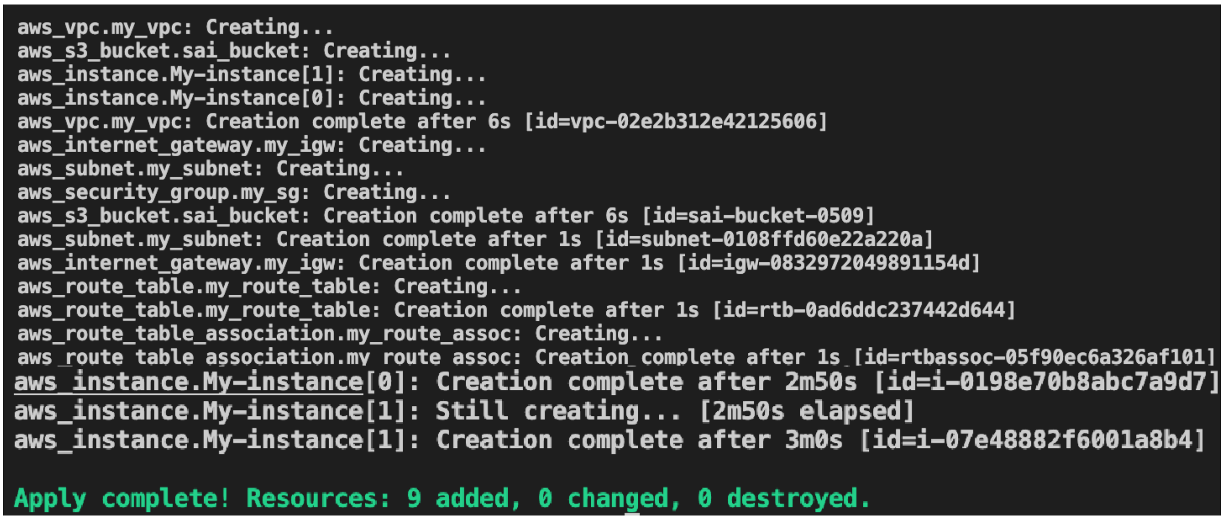
```
● (venv_3.8.10) sai@Sais-Mac Assignment-1 % terraform apply

  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:
```

It will ask for approval before creating, enter “yes” to continue.



```
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
```

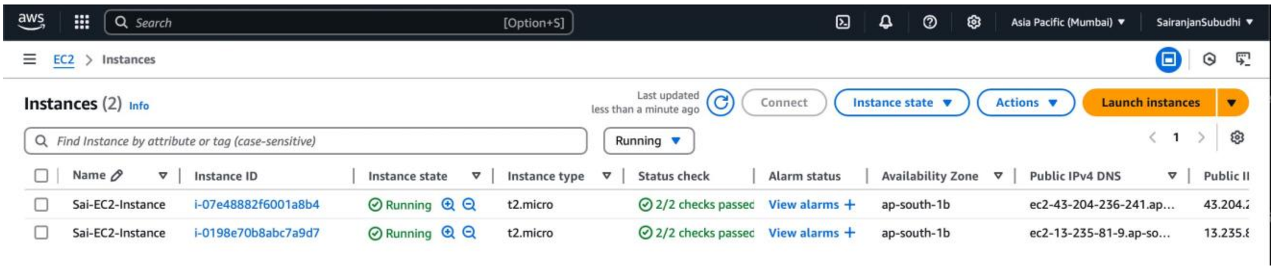
After approval, it will start creating.



```
aws_vpc.my_vpc: Creating...
aws_s3_bucket.sai_bucket: Creating...
aws_instance.My-instance[1]: Creating...
aws_instance.My-instance[0]: Creating...
aws_vpc.my_vpc: Creation complete after 6s [id=vpc-02e2b312e42125606]
aws_internet_gateway.my_igw: Creating...
aws_subnet.my_subnet: Creating...
aws_security_group.my_sg: Creating...
aws_s3_bucket.sai_bucket: Creation complete after 6s [id=sai-bucket-0509]
aws_subnet.my_subnet: Creation complete after 1s [id=subnet-0108ffd60e22a220a]
aws_internet_gateway.my_igw: Creation complete after 1s [id=igw-0832972049891154d]
aws_route_table.my_route_table: Creating...
aws_route_table.my_route_table: Creation complete after 1s [id=rtb-0ad6ddc237442d644]
aws_route_table_association.my_route_assoc: Creating...
aws_route_table_association.my_route_assoc: Creation complete after 1s [id=rtbassoc-05f90ec6a326af101]
aws_instance.My-instance[0]: Creation complete after 2m50s [id=i-0198e70b8abc7a9d7]
aws_instance.My-instance[1]: Still creating... [2m50s elapsed]
aws_instance.My-instance[1]: Creation complete after 3m0s [id=i-07e48882f6001a8b4]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.
```

You can verify by logging into the AWS Console,

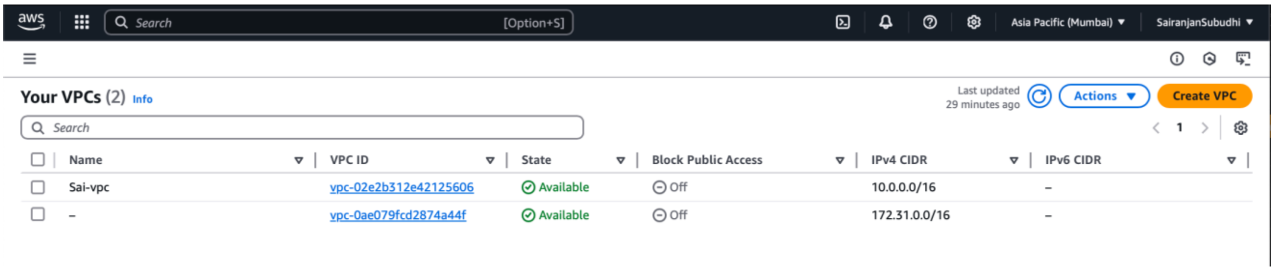
- Creation of 2 instances



Instances (2) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public II
☐	Sai-EC2-Instance	i-07e48882f6001a8b4	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-43-204-236-241.ap...	43.204.2
☐	Sai-EC2-Instance	i-0198e70b8abc7a9d7	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-235-81-9.ap-so...	13.235.8

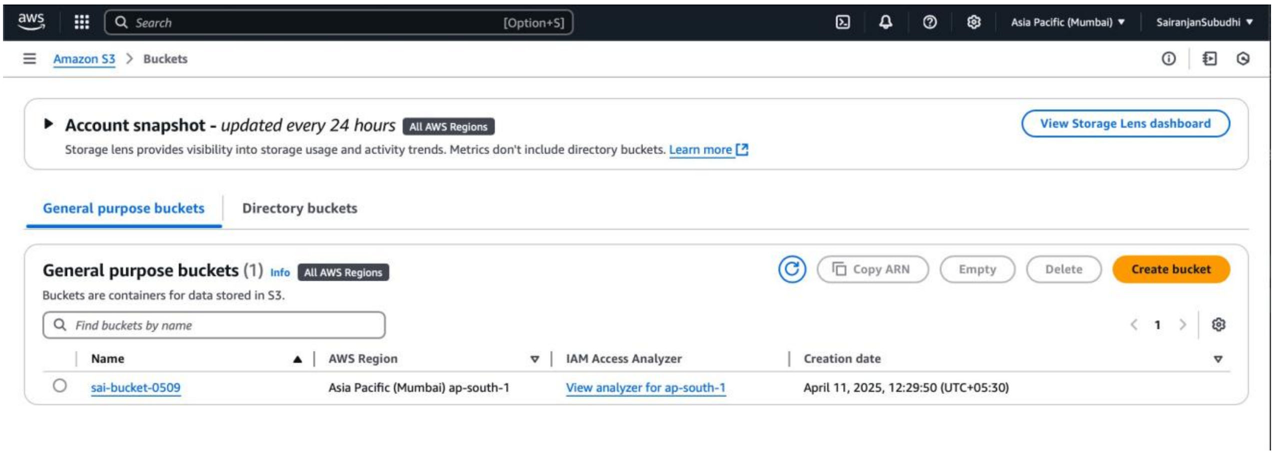
- Creation of a VPC



Your VPCs (2) Info

☐	Name	VPC ID	State	Block Public Access	IPv4 CIDR	IPv6 CIDR
☐	Sai-vpc	vpc-02e2b312e42125606	Available	Off	10.0.0.0/16	–
☐	–	vpc-0ae079fcd2874a44f	Available	Off	172.31.0.0/16	–

- Creation of 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

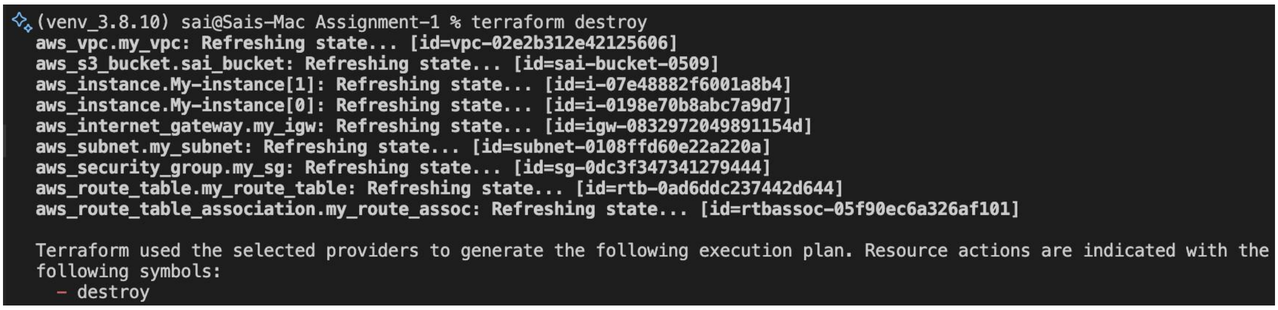
Buckets are containers for data stored in S3.

○	Name	AWS Region	IAM Access Analyzer	Creation date
○	sai-bucket-0509	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 11, 2025, 12:29:50 (UTC+05:30)

Step 8: Cleanup Resources

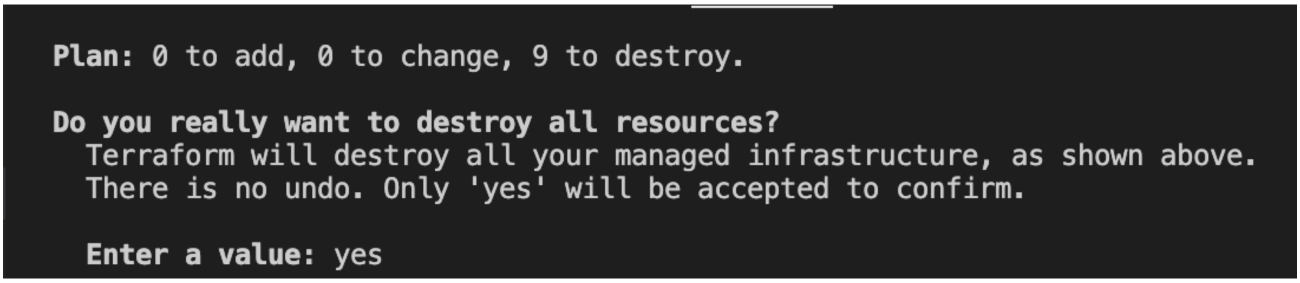
When you are done experimenting, run the following command to destroy the created resources:

Terraform destroy



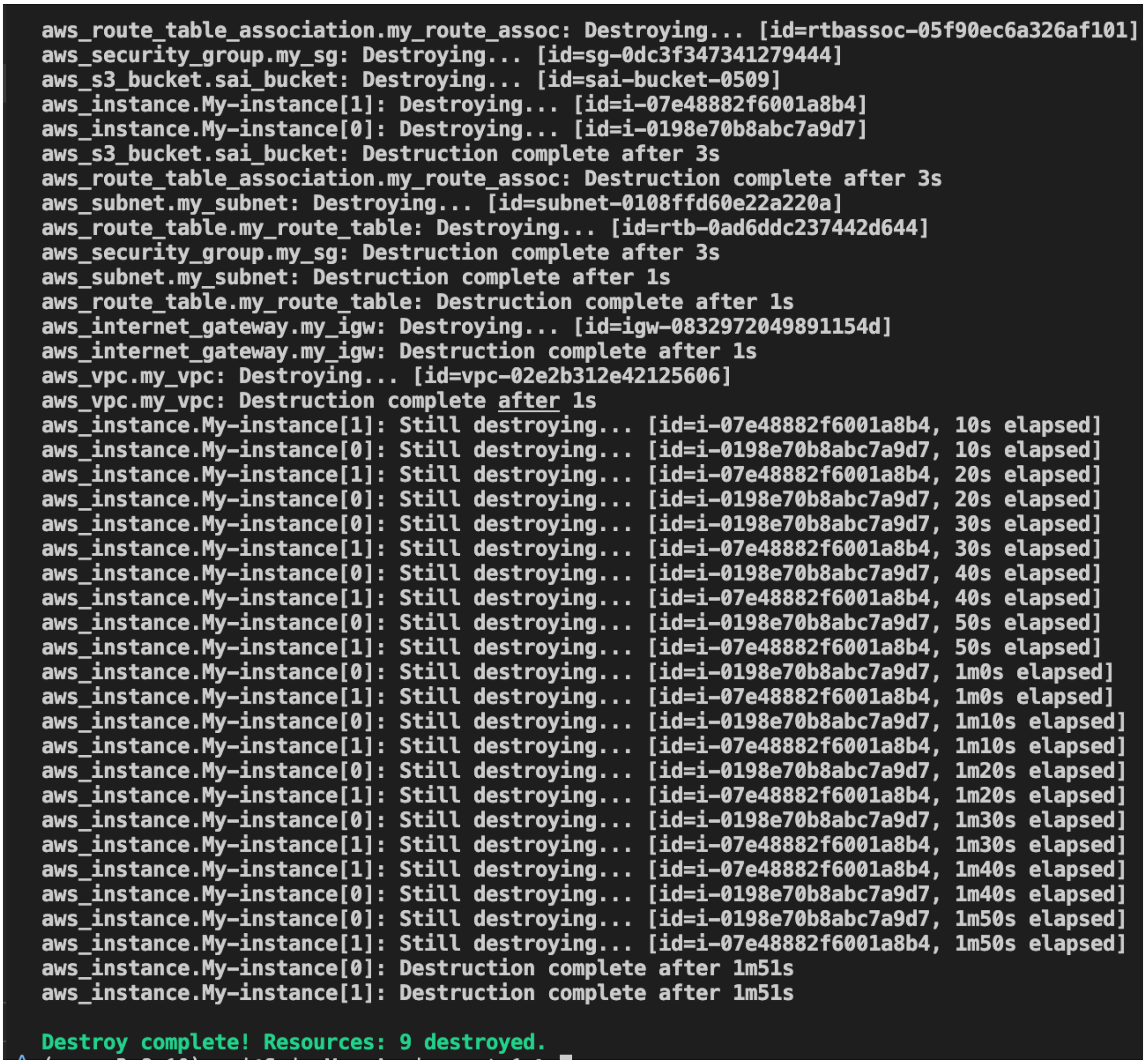
```
(venv_3.8.10) sai@Sais-Mac Assignment-1 % terraform destroy
 aws_vpc.my_vpc: Refreshing state... [id=vpc-02e2b312e42125606]
 aws_s3_bucket.sai_bucket: Refreshing state... [id=sai-bucket-0509]
 aws_instance.My-instance[1]: Refreshing state... [id=i-07e48882f6001a8b4]
 aws_instance.My-instance[0]: Refreshing state... [id=i-0198e70b8abc7a9d7]
 aws_internet_gateway.my_igw: Refreshing state... [id=igw-0832972049891154d]
 aws_subnet.my_subnet: Refreshing state... [id=subnet-0108ffd60e22a220a]
 aws_security_group.my_sg: Refreshing state... [id=sg-0dc3f347341279444]
 aws_route_table.my_route_table: Refreshing state... [id=rtb-0ad6ddc237442d644]
 aws_route_table_association.my_route_assoc: Refreshing state... [id=rtbassoc-05f90ec6a326af101]

 Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
 following symbols:
   - destroy
```

It will ask for approval before destroying, enter “yes” to continue.



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

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes
```

After approval, it will start destroying.



```
aws_route_table_association.my_route_assoc: Destroying... [id=rtbassoc-05f90ec6a326af101]
aws_security_group.my_sg: Destroying... [id=sg-0dc3f347341279444]
aws_s3_bucket.sai_bucket: Destroying... [id=sai-bucket-0509]
aws_instance.My-instance[1]: Destroying... [id=i-07e48882f6001a8b4]
aws_instance.My-instance[0]: Destroying... [id=i-0198e70b8abc7a9d7]
aws_s3_bucket.sai_bucket: Destruction complete after 3s
aws_route_table_association.my_route_assoc: Destruction complete after 3s
aws_subnet.my_subnet: Destroying... [id=subnet-0108ffd60e22a220a]
aws_route_table.my_route_table: Destroying... [id=rtb-0ad6ddc237442d644]
aws_security_group.my_sg: Destruction complete after 3s
aws_subnet.my_subnet: Destruction complete after 1s
aws_route_table.my_route_table: Destruction complete after 1s
aws_internet_gateway.my_igw: Destroying... [id=igw-0832972049891154d]
aws_internet_gateway.my_igw: Destruction complete after 1s
aws_vpc.my_vpc: Destroying... [id=vpc-02e2b312e42125606]
aws_vpc.my_vpc: Destruction complete after 1s
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 10s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 10s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 20s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 20s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 30s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 30s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 40s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 40s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 50s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 50s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 1m0s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 1m0s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 1m10s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 1m10s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 1m20s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 1m20s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 1m30s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 1m30s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 1m40s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 1m40s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0198e70b8abc7a9d7, 1m50s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-07e48882f6001a8b4, 1m50s elapsed]
aws_instance.My-instance[0]: Destruction complete after 1m51s
aws_instance.My-instance[1]: Destruction complete after 1m51s

Destroy complete! Resources: 9 destroyed.
```