



System Provisioning & Configuration Management

ASSIGNMENT-1

SUBMITTED BY:

NAME: SAIRANJAN SUBUDHI

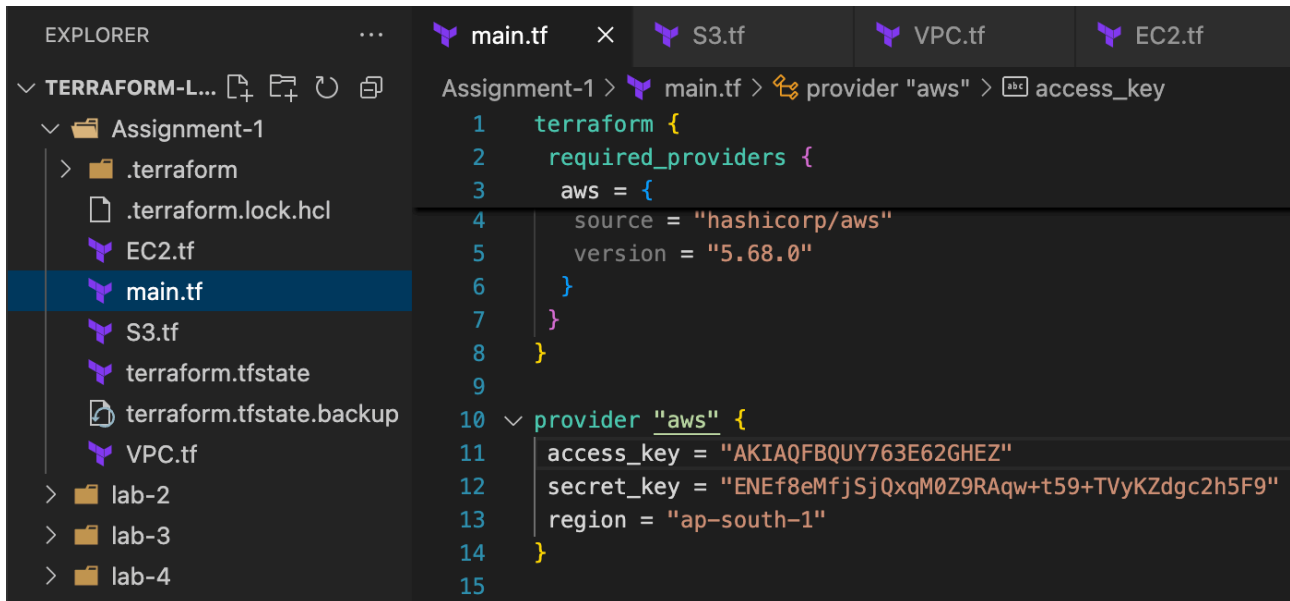
SAP ID: 500101861

BATCH: 2 (DevOps)

ROLL NO: R2142220816

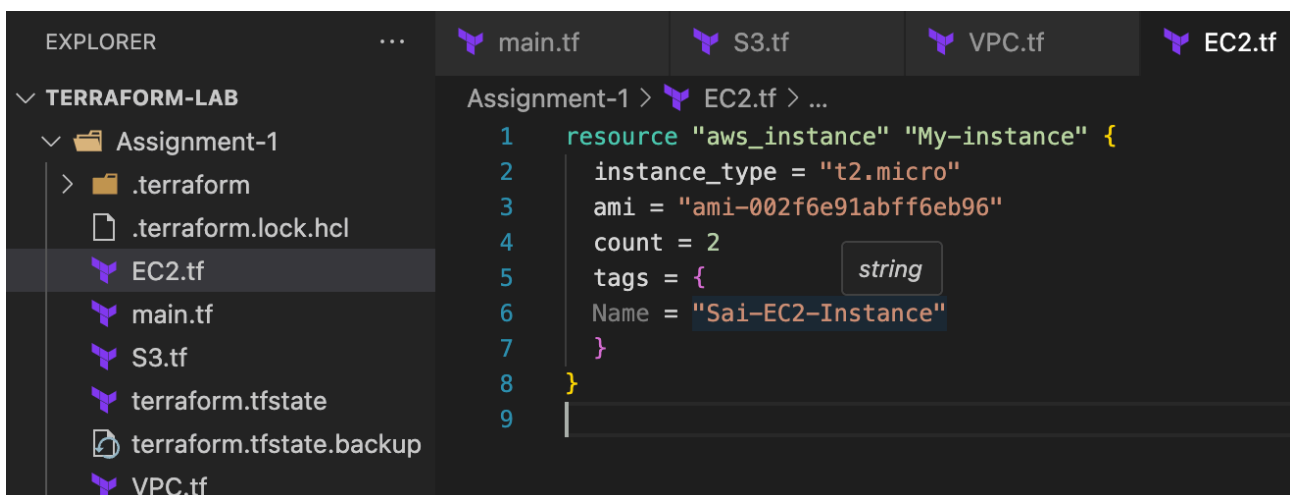
Write Terraform script to do perform following tasks on AWS cloud Platform

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



```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.68.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   access_key = "AKIAQFBQUY763E62GHEZ"
12   secret_key = "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"
13   region = "ap-south-1"
14 }
15
```

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



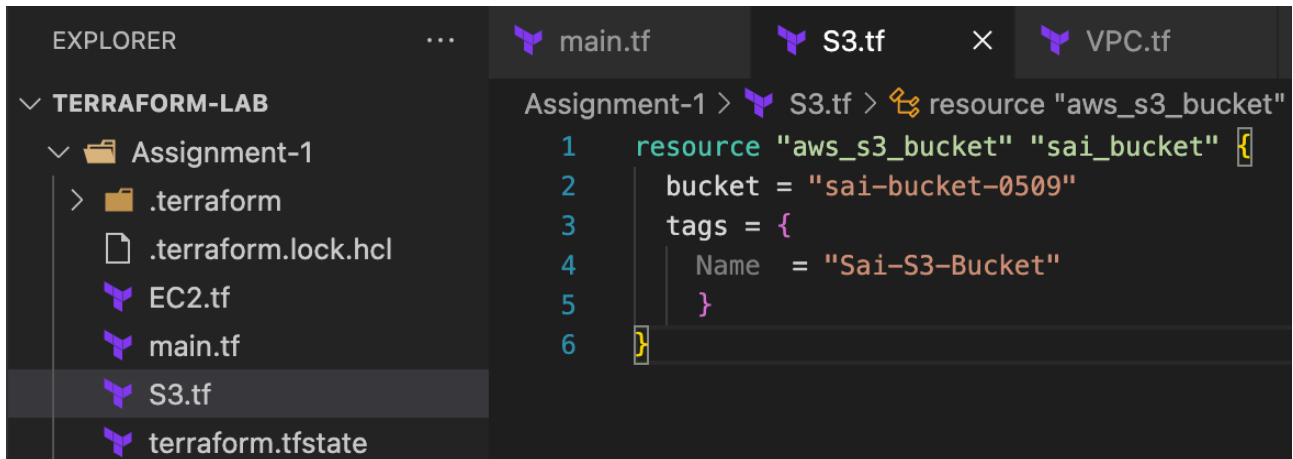
```
1 resource "aws_instance" "My-instance" {
2   instance_type = "t2.micro"
3   ami = "ami-002f6e91abff6eb96"
4   count = 2
5   tags = {
6     Name = "Sai-EC2-Instance"
7   }
8 }
9
```

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

The image shows a Terraform IDE interface with two panels. The left panel displays the file explorer for a project named 'TERRAFORM-LAB', showing a directory structure with 'Assignment-1' containing files like '.terraform.lock.hcl', 'EC2.tf', 'main.tf', 'S3.tf', 'terraform.tfstate', and 'terraform.tfstate.backup'. The right panel shows the 'VPC.tf' file being edited, with the following Terraform configuration:

```
1 resource "aws_vpc" "my_vpc" {
2   cidr_block = "10.0.0.0/16"
3
4   tags = {
5     Name = "Sai-vpc"
6   }
7 }
8
9 resource "aws_subnet" "my_subnet" {
10  vpc_id      = aws_vpc.my_vpc.id
11  cidr_block  = "10.0.1.0/24"
12  availability_zone = "ap-south-1a"
13
14  tags = {
15    Name = "Sai-subnet"
16  }
17 }
18
19 resource "aws_internet_gateway" "my_igw" {
20  vpc_id = aws_vpc.my_vpc.id
21
22  tags = {
23    Name = "my-igw"
24  }
25 }
26
27 resource "aws_route_table" "my_route_table" {
28  vpc_id = aws_vpc.my_vpc.id
29
30  route {
31    cidr_block = "0.0.0.0/0"
32    gateway_id = aws_internet_gateway.my_igw.id
33  }
34
35  tags = {
36    Name = "my-route-table"
37  }
38 }
39
40 resource "aws_route_table_association" "my_route_assoc" {
41  subnet_id      = aws_subnet.my_subnet.id
42  route_table_id = aws_route_table.my_route_table.id
43 }
44
45 resource "aws_security_group" "my_sg" {
46  name        = "my-security-group"
47  description = "Allow SSH inbound traffic"
48  vpc_id      = aws_vpc.my_vpc.id
49
50  ingress {
51    description = "SSH"
52    from_port   = 22
53    to_port     = 22
54    protocol    = "tcp"
55    cidr_blocks = ["0.0.0.0/0"]
56  }
57
58  egress {
59    from_port = 0
60    to_port   = 0
61    protocol  = "-1"
62    cidr_blocks = ["0.0.0.0/0"]
63  }
64
65  tags = {
66    Name = "my-sg"
67  }
68 }
69
```

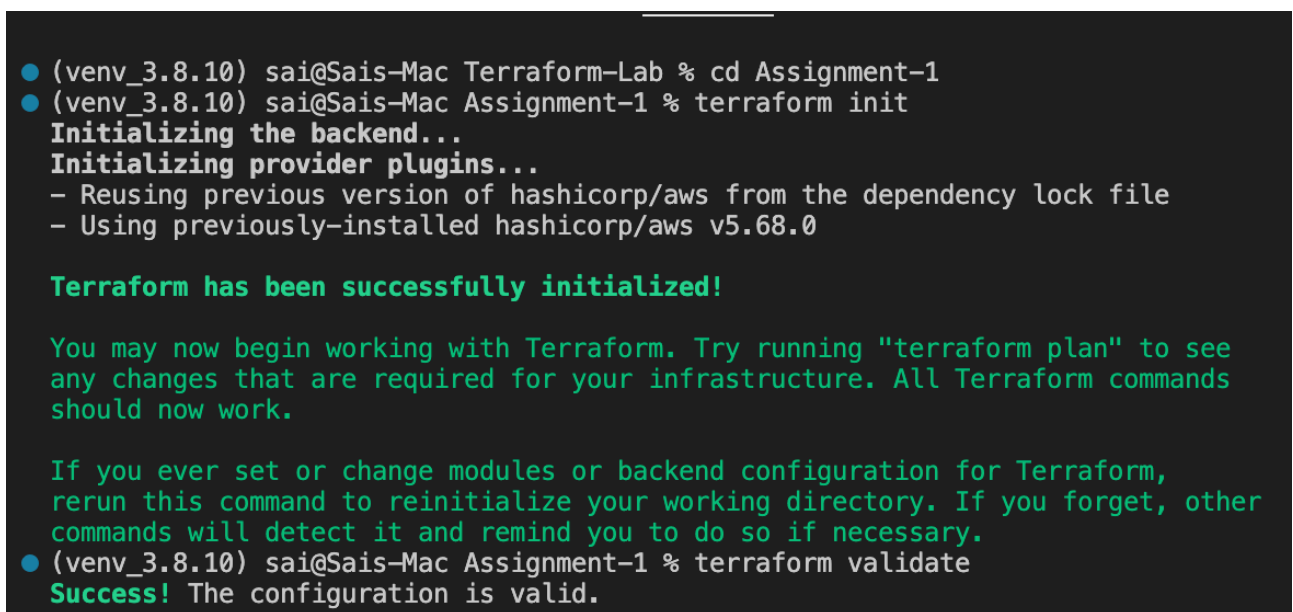
Step 4: To Create a S3 Bucket create S3.tf :



Step 5: Initialize Terraform

Run the following command to initialize your Terraform working directory:

Terraform init



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
```

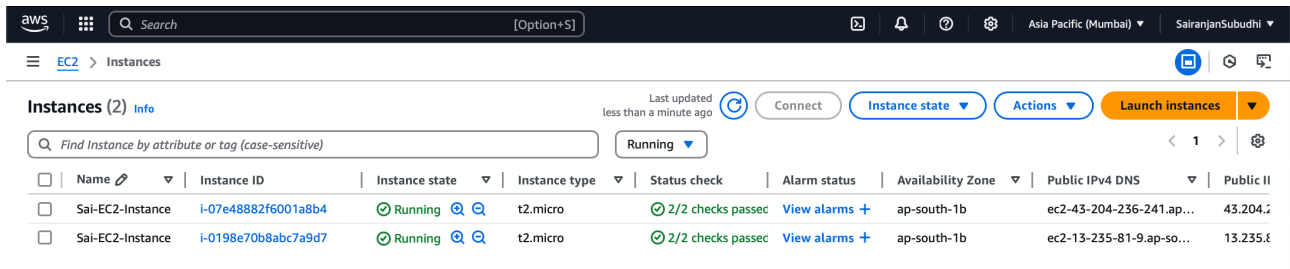
After approval, it will start creating.

```
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]

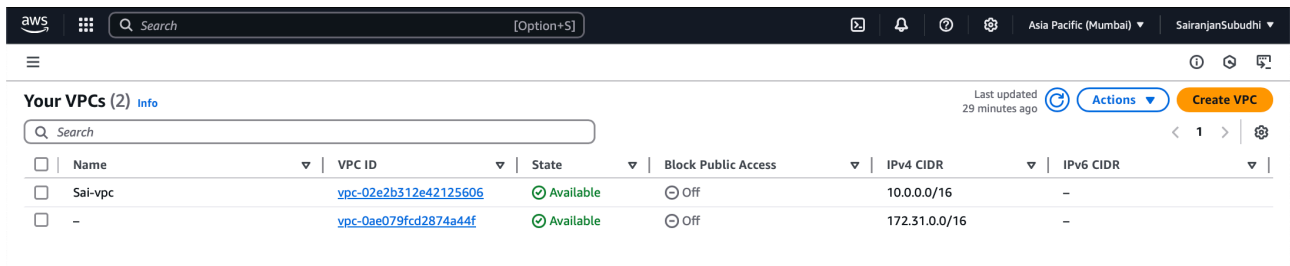
Apply complete! Resources: 9 added, 0 changed, 0 destroyed.
```

You can verify by logging into the AWS Console,

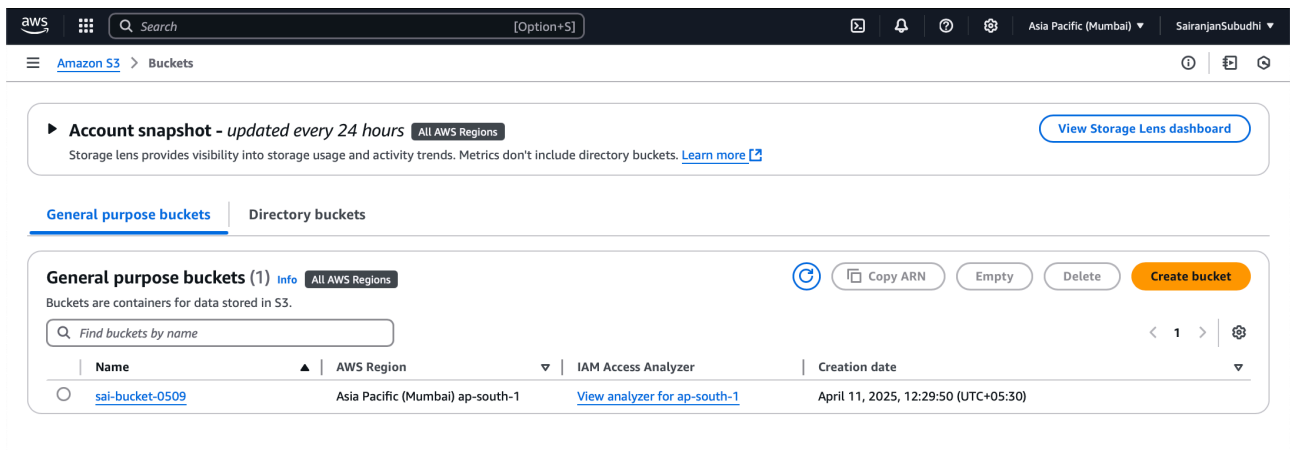
- Creation of 2 instances



- Creation of a VPC



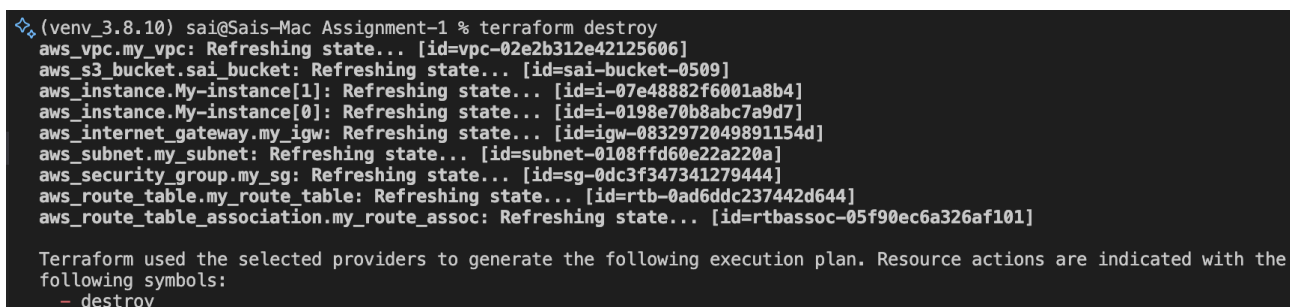
- Creation of S3 Bucket



Step 8: Cleanup Resources

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

Terraform 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.
```