Lab Exercise 9– Creating Multiple EC2 Instances with for each in Terraform

Objective:

Learn how to use for each in Terraform to create multiple AWS EC2 instances with specific settings for each instance.

Prerequisites:

- Terraform installed on your machine.
- AWS CLI configured with the necessary credentials.

Steps:

1. Create a Terraform Directory:

```
mkdir terraform-ec2-for-each
cd terraform-ec2-for-each
```

- Create Terraform Configuration Files:
- Create a file named main.tf:

# main.tf

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

variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1" = {
      ami           = "ami-0c55b159cbfafe1f0"
      instance_type = "t2.micro"
    },
    "instance2" = {
      ami           = "ami-0123456789abcdef0"
      instance_type = "t2. micro "
    },
    "instance3" = {
      ami           = "ami-9876543210fedcba0"
      instance_type = "t2. micro "
    }
  }
}

resource "aws_instance" "ec2_instances" {
  for_each = var.instances
  ami           = var.instances[each.key].ami
  instance_type = var.instances[each.key].instance_type
  tags = {
    Name = "EC2-Instance-${each.key}"
  }
}
```

- Replace "your-key-pair-name" and "your-subnet-id" with your actual key pair name and subnet ID.
- In this configuration, we define a variable instances as a map containing settings for each EC2 instance. The aws_instance resource is then used with for_each to create instances based on the map.

2. Initialize and Apply:

- Run the following Terraform commands to initialize and apply the configuration:

```
terraform init
terraform apply
```

- Terraform will prompt you to confirm the creation of EC2 instances. Type yes and press Enter.

3. Verify Instances in AWS Console:

- Log in to the AWS Management Console and navigate to the EC2 service.
- Verify that the specified EC2 instances with the specified names and settings have been created.

4. Update Instance Configuration:

- If you want to modify the EC2 instance configuration, update the main.tf file with the desired changes.
- Rerun the terraform apply command to apply the changes:

```
terraform apply
```

5. Clean Up:

- After testing, you can clean up the EC2 instances:

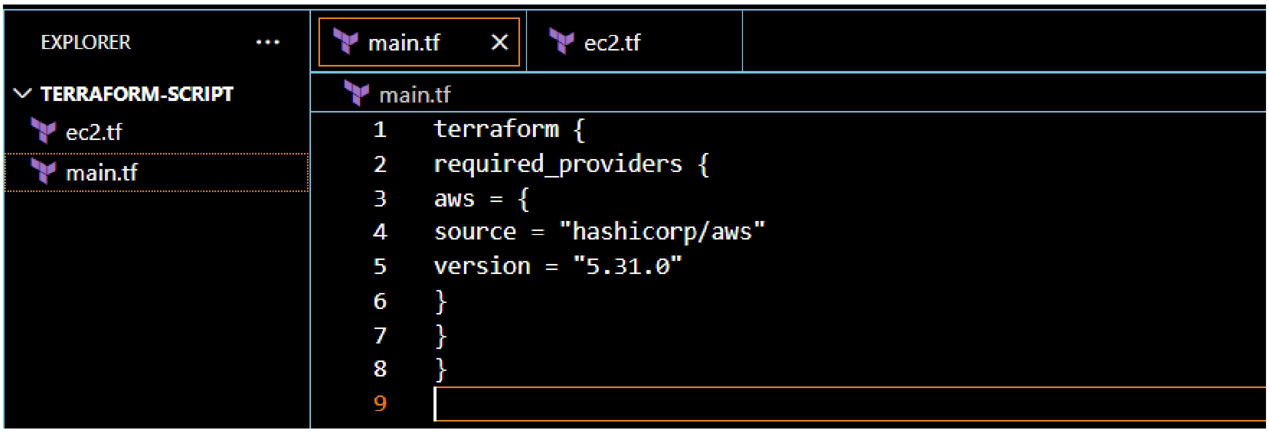
```
terraform destroy
```

- Confirm the destruction by typing yes.

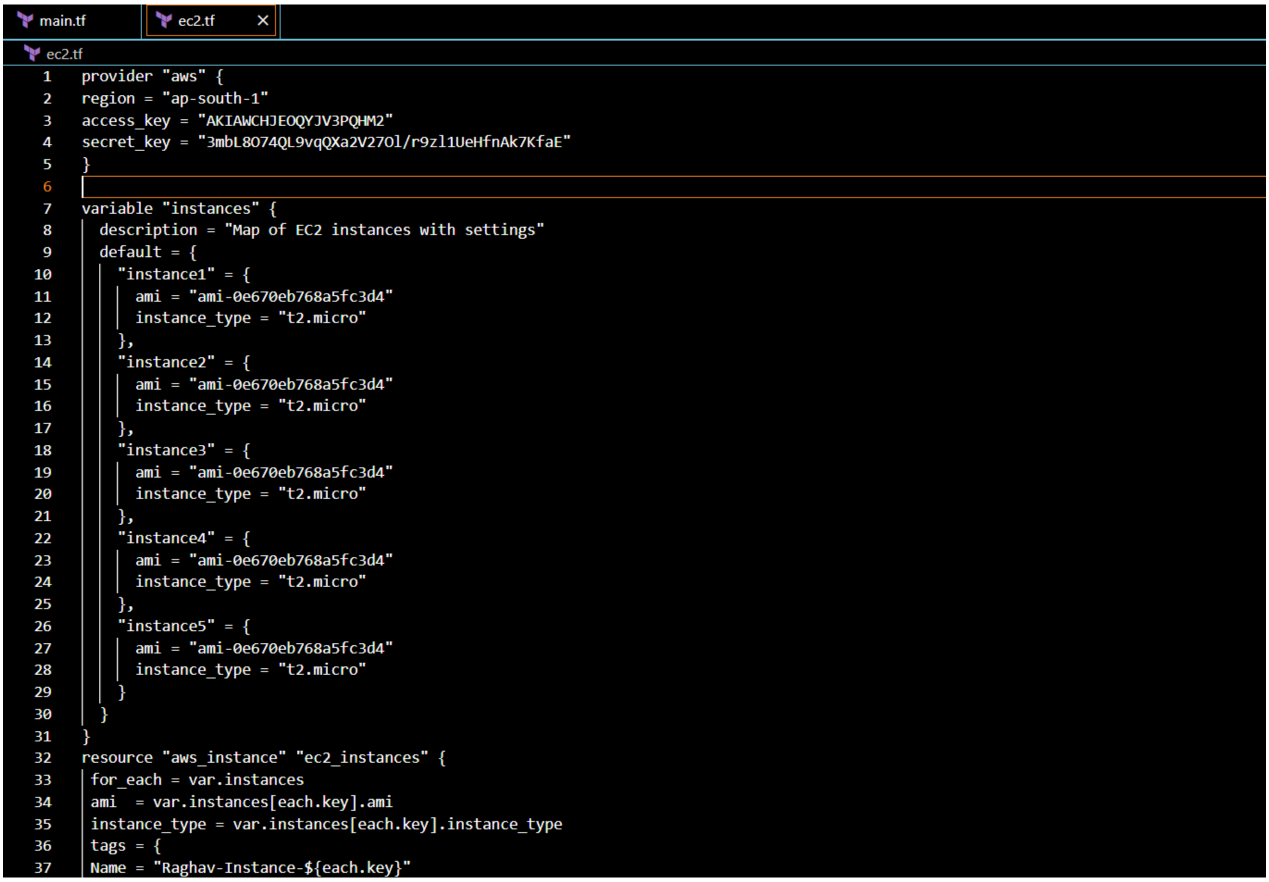
6. Conclusion:

This lab exercise demonstrates how to use the for_each construct in Terraform to create multiple AWS EC2 instances with specific settings for each instance. The use of a map allows you to define and manage settings for each instance individually.

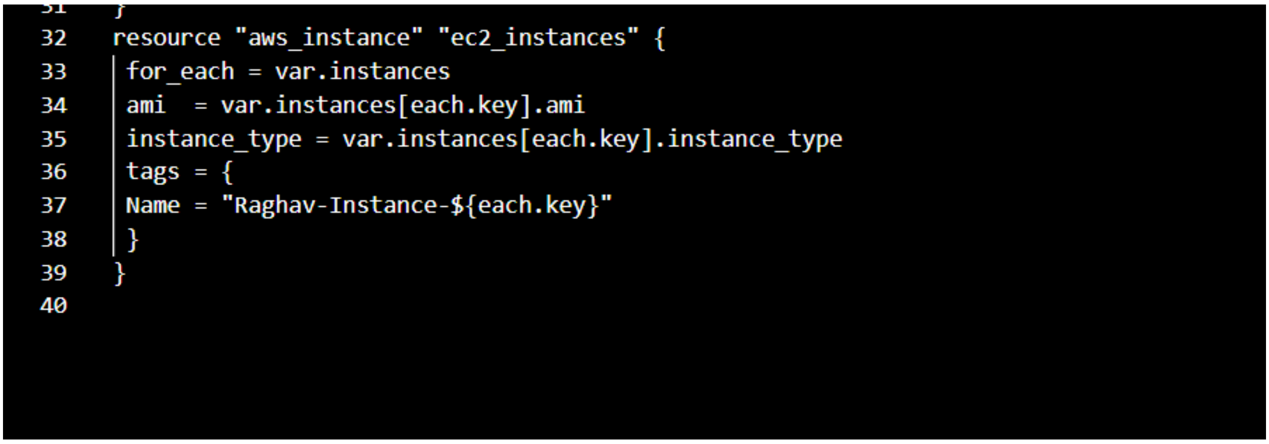
Experiment with different instance types, AMIs, and settings in the main.tf file to observe how Terraform provisions resources based on your configuration.



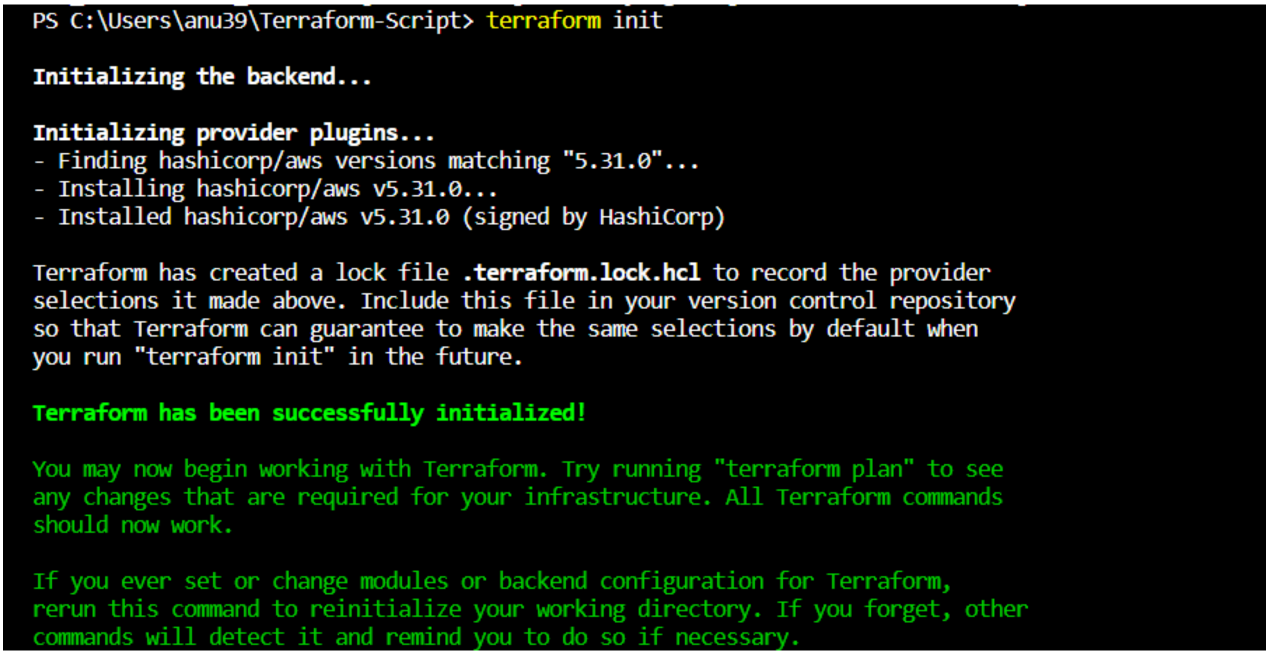
```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}
```



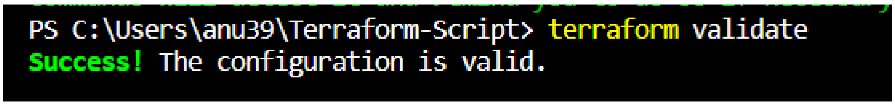
```
provider "aws" {
region = "ap-south-1"
access_key = "AKIAWCHJEOQYJV3PQHM2"
secret_key = "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"
}

variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1" = {
      ami = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    },
    "instance2" = {
      ami = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    },
    "instance3" = {
      ami = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    },
    "instance4" = {
      ami = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    },
    "instance5" = {
      ami = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    }
  }
}
resource "aws_instance" "ec2_instances" {
 for_each = var.instances
 ami  = var.instances[each.key].ami
 instance_type = var.instances[each.key].instance_type
 tags = {
 Name = "Raghav-Instance-${each.key}"
```



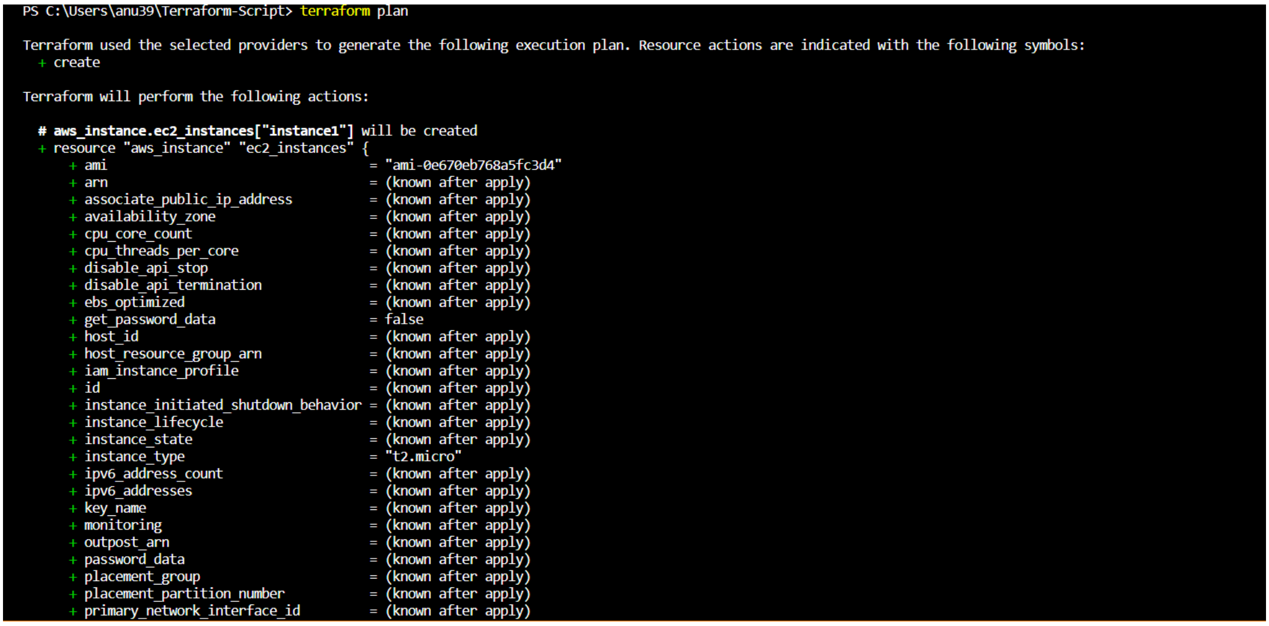
```
}
resource "aws_instance" "ec2_instances" {
  for_each = var.instances
  ami   = var.instances[each.key].ami
  instance_type = var.instances[each.key].instance_type
  tags = {
  Name = "Raghav-Instance-${each.key}"
  }
}
```



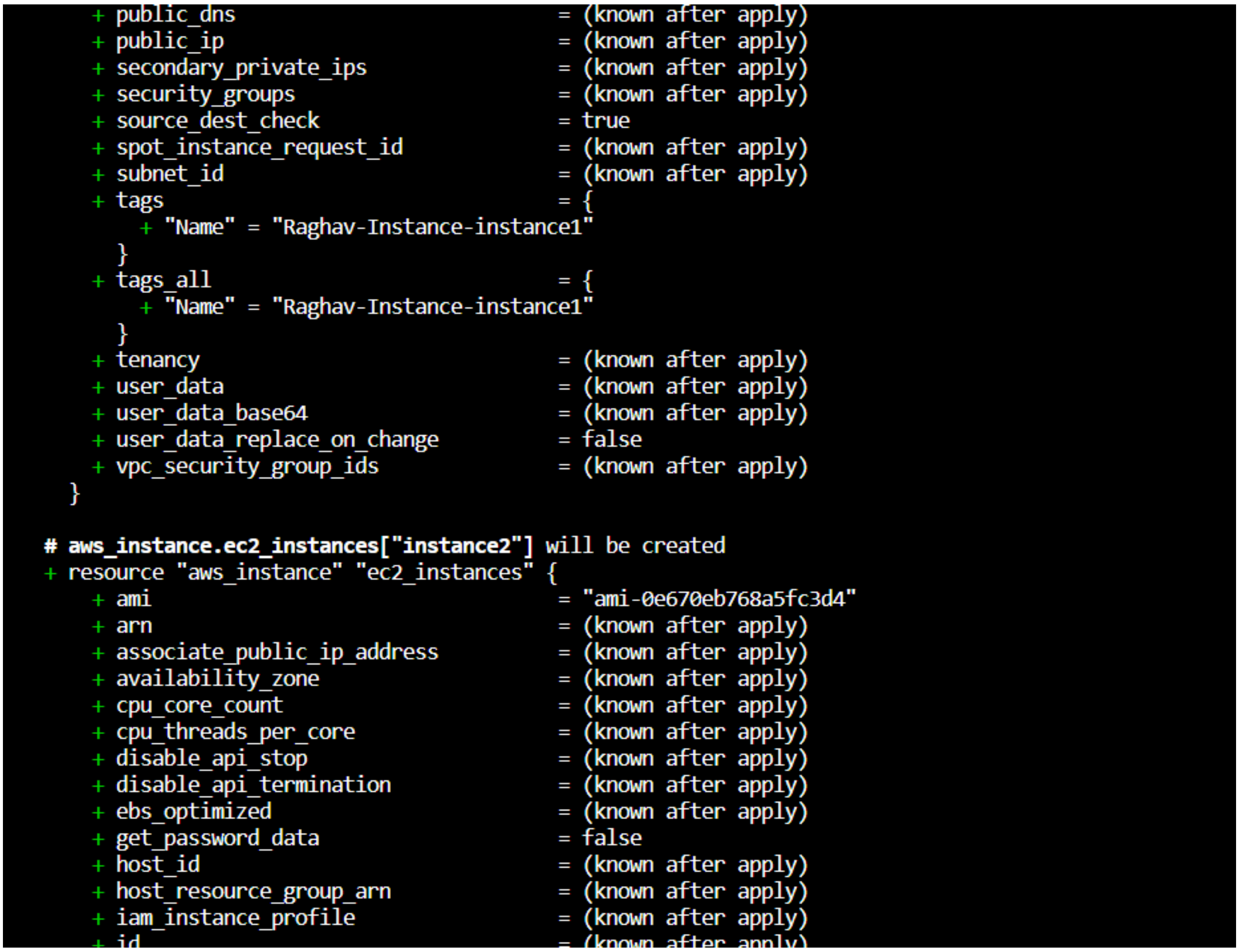
```
PS C:\Users\anu39\Terraform-Script> terraform init

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.
```



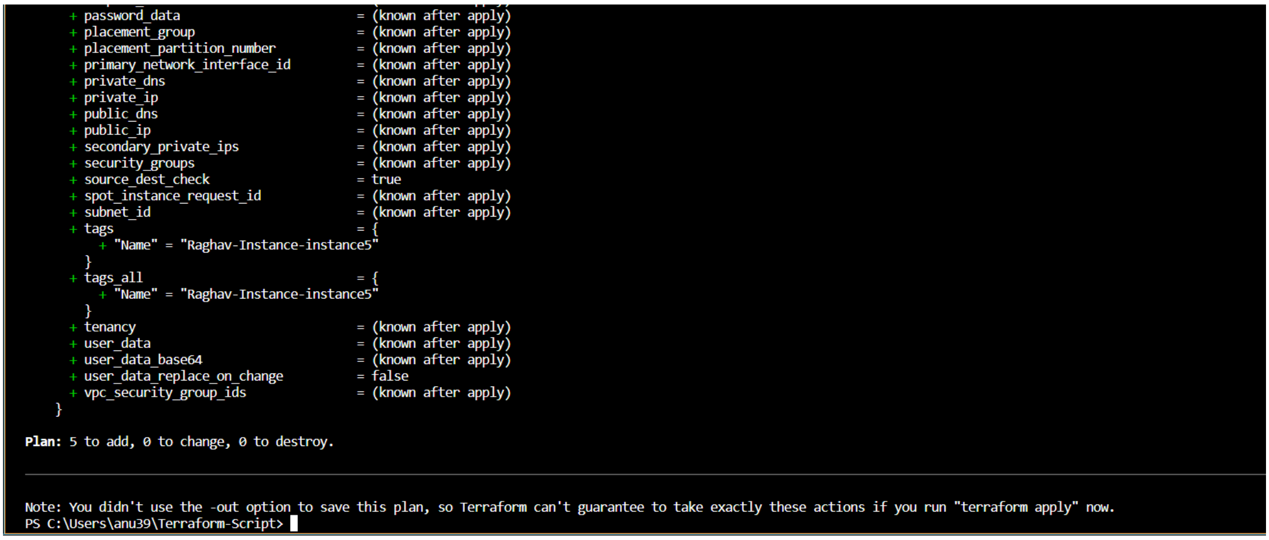
```
PS C:\Users\anu39\Terraform-Script> terraform validate
Success! The configuration is valid.
```



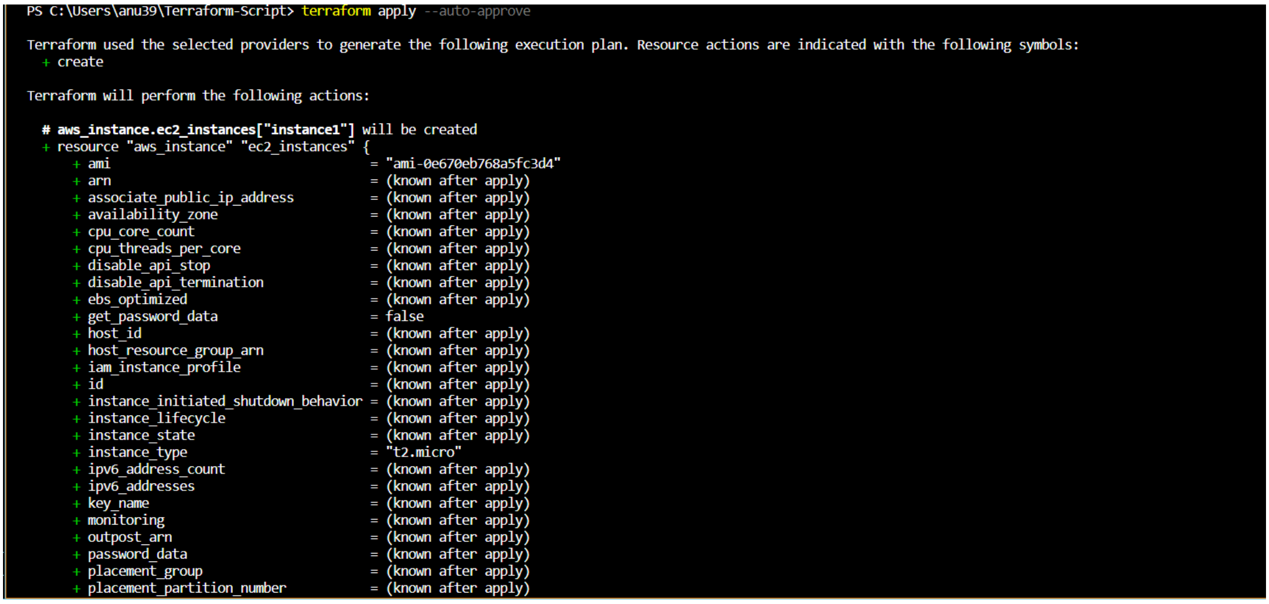
```
PS C:\Users\anu39\Terraform-Script> terraform plan

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.ec2_instances["instance1"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0e670eb768a5fc3d4"
      + 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"
      + 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)
```



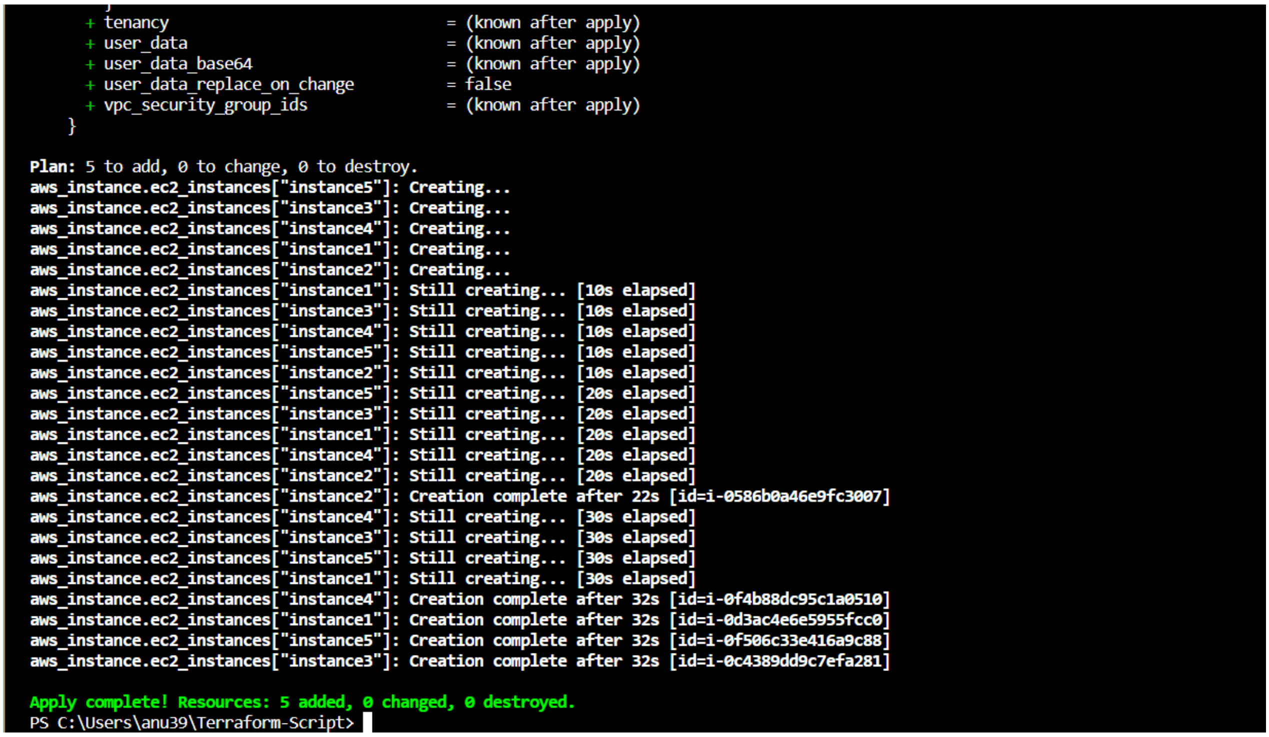
```
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "Raghav-Instance-instance1"
        }
      + tags_all                             = {
          + "Name" = "Raghav-Instance-instance1"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

  # aws_instance.ec2_instances["instance2"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0e670eb768a5fc3d4"
      + 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)
```



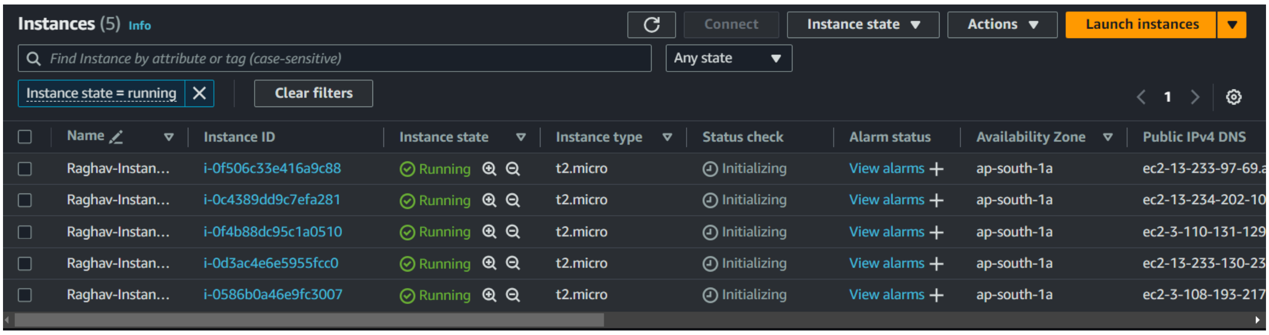
```
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "Raghav-Instance-instance5"
        }
      + tags_all                             = {
          + "Name" = "Raghav-Instance-instance5"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\anu39\Terraform-Script>
```



```
PS C:\Users\anu39\Terraform-Script> terraform apply --auto-approve

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_instances["instance1"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0e670eb768a5fc3d4"
      + 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"
      + 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)
```

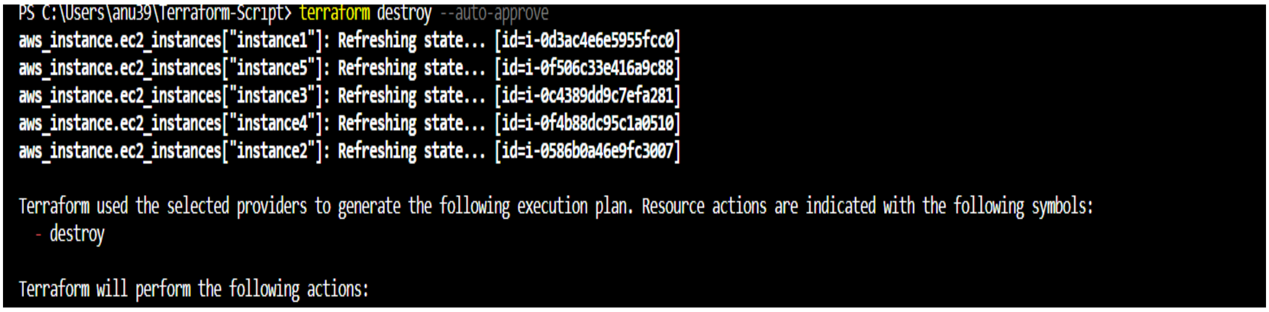


```
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.
aws_instance.ec2_instances["instance5"]: Creating...
aws_instance.ec2_instances["instance3"]: Creating...
aws_instance.ec2_instances["instance4"]: Creating...
aws_instance.ec2_instances["instance1"]: Creating...
aws_instance.ec2_instances["instance2"]: Creating...
aws_instance.ec2_instances["instance1"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance4"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance5"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance5"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance1"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance4"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance2"]: Creation complete after 22s [id=i-0586b0a46e9fc3007]
aws_instance.ec2_instances["instance4"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance5"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance1"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance4"]: Creation complete after 32s [id=i-0f4b88dc95c1a0510]
aws_instance.ec2_instances["instance1"]: Creation complete after 32s [id=i-0d3ac4e6e5955fcc0]
aws_instance.ec2_instances["instance5"]: Creation complete after 32s [id=i-0f506c33e416a9c88]
aws_instance.ec2_instances["instance3"]: Creation complete after 32s [id=i-0c4389dd9c7efa281]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
PS C:\Users\anu39\Terraform-Script>
```

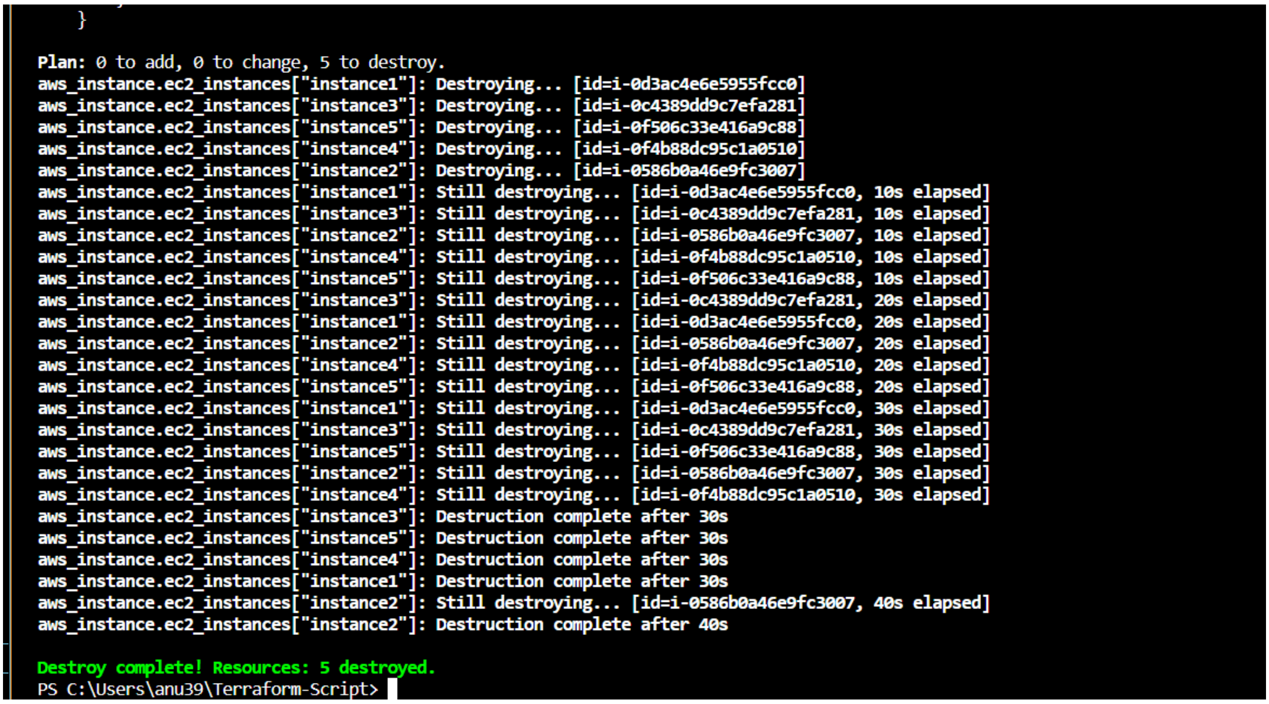


Instance state = running

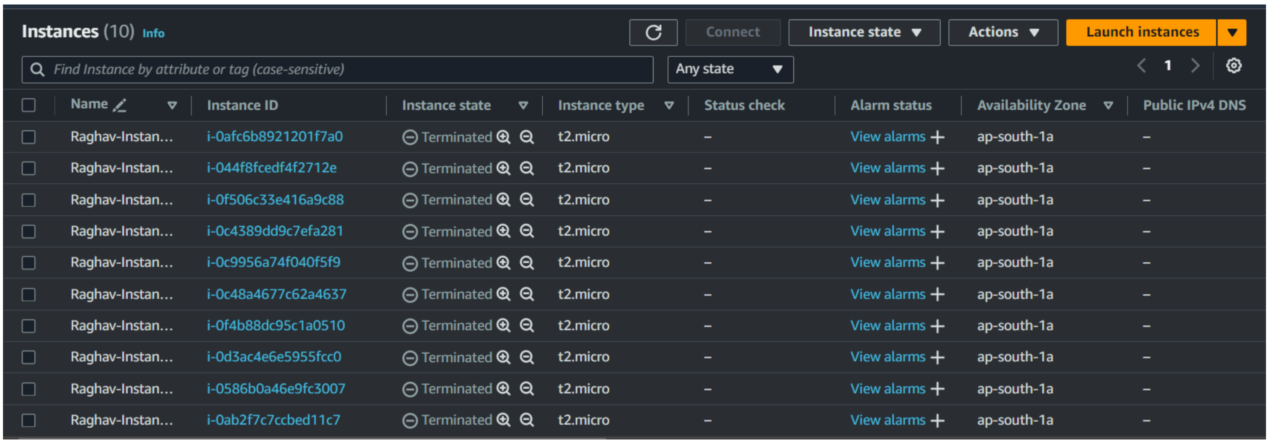
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Raghav-Instan...	i-0f506c33e416a9c88	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-233-97-69.a
□	Raghav-Instan...	i-0c4389dd9c7efa281	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-234-202-10
□	Raghav-Instan...	i-0f4b88dc95c1a0510	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-3-110-131-129
□	Raghav-Instan...	i-0d3ac4e6e5955fcc0	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-233-130-23
□	Raghav-Instan...	i-0586b0a46e9fc3007	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-3-108-193-217



```
PS C:\Users\anu39\Terraform-Script> terraform destroy --auto-approve
aws_instance.ec2_instances["instance1"]: Refreshing state... [id=i-0d3ac4e6e5955fcc0]
aws_instance.ec2_instances["instance5"]: Refreshing state... [id=i-0f506c33e416a9c88]
aws_instance.ec2_instances["instance3"]: Refreshing state... [id=i-0c4389dd9c7efa281]
aws_instance.ec2_instances["instance4"]: Refreshing state... [id=i-0f4b88dc95c1a0510]
aws_instance.ec2_instances["instance2"]: Refreshing state... [id=i-0586b0a46e9fc3007]

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

Terraform will perform the following actions:
```



```
    }

Plan: 0 to add, 0 to change, 5 to destroy.
aws_instance.ec2_instances["instance1"]: Destroying... [id=i-0d3ac4e6e5955fcc0]
aws_instance.ec2_instances["instance3"]: Destroying... [id=i-0c4389dd9c7efa281]
aws_instance.ec2_instances["instance5"]: Destroying... [id=i-0f506c33e416a9c88]
aws_instance.ec2_instances["instance4"]: Destroying... [id=i-0f4b88dc95c1a0510]
aws_instance.ec2_instances["instance2"]: Destroying... [id=i-0586b0a46e9fc3007]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-0d3ac4e6e5955fcc0, 10s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0c4389dd9c7efa281, 10s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0586b0a46e9fc3007, 10s elapsed]
aws_instance.ec2_instances["instance4"]: Still destroying... [id=i-0f4b88dc95c1a0510, 10s elapsed]
aws_instance.ec2_instances["instance5"]: Still destroying... [id=i-0f506c33e416a9c88, 10s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0c4389dd9c7efa281, 20s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-0d3ac4e6e5955fcc0, 20s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0586b0a46e9fc3007, 20s elapsed]
aws_instance.ec2_instances["instance4"]: Still destroying... [id=i-0f4b88dc95c1a0510, 20s elapsed]
aws_instance.ec2_instances["instance5"]: Still destroying... [id=i-0f506c33e416a9c88, 20s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-0d3ac4e6e5955fcc0, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0c4389dd9c7efa281, 30s elapsed]
aws_instance.ec2_instances["instance5"]: Still destroying... [id=i-0f506c33e416a9c88, 30s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0586b0a46e9fc3007, 30s elapsed]
aws_instance.ec2_instances["instance4"]: Still destroying... [id=i-0f4b88dc95c1a0510, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Destruction complete after 30s
aws_instance.ec2_instances["instance5"]: Destruction complete after 30s
aws_instance.ec2_instances["instance4"]: Destruction complete after 30s
aws_instance.ec2_instances["instance1"]: Destruction complete after 30s
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0586b0a46e9fc3007, 40s elapsed]
aws_instance.ec2_instances["instance2"]: Destruction complete after 40s

Destroy complete! Resources: 5 destroyed.
PS C:\Users\anu39\Terraform-Script>
```



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Raghav-Instan...	i-0afc6b8921201f7a0	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-044f8fcedf4f2712e	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0f506c33e416a9c88	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0c4389dd9c7efa281	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0c9956a74f040f5f9	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0c48a4677c62a4637	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0f4b88dc95c1a0510	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0d3ac4e6e5955fcc0	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0586b0a46e9fc3007	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
□	Raghav-Instan...	i-0ab2f7c7ccbed11c7	Terminated	t2.micro	–	View alarms +	ap-south-1a	–