Lab Exercise 6– Terraform Multiple tfvars Files

Objective:

Learn how to use multiple tfvars files in Terraform for different environments.

Prerequisites:

- Terraform installed on your machine.
- Basic knowledge of Terraform configuration and variables.

Steps:

1. Create a Terraform Directory:

```
mkdir terraform-multiple-tfvars
cd terraform-multiple-tfvars
```

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

# main.tf

```
provider "aws" {
  region = var.region
}

resource "aws_instance" "example" {
  ami           = var.ami
  instance_type = var.instance_type
}
```

- Create a file named variables.tf:

# variables.tf

```
variable "region" {
  description = "AWS region"
  default     = "us-west-2"
}

variable "ami" {
  description = "AMI ID"
  default     = "ami-0c55b159cbfafe1f0"
}

variable "instance_type" {
  description = "EC2 Instance Type"
  default     = "t2.micro"
}
```

2. Create Multiple tfvars Files:

- Create a file named dev.tfvars:

# dev.tfvars

```
region        = "us-west-2"
ami           = "ami-0123456789abcdef0"
instance_type = "t2.micro"
```

- Create a file named prod.tfvars:

# prod.tfvars

```
region        = "us-east-1"
ami           = "ami-9876543210fedcba0"
instance_type = "t2.large"
```

- In these files, provide values for the variables based on the environments.

3. Initialize and Apply for Dev Environment:

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

```
terraform init
terraform apply -var-file=dev.tfvars
```

4. Initialize and Apply for Prod Environment:

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

```
terraform init
terraform apply -var-file=prod.tfvars
```

5. Test and Verify:

- Observe how different tfvars files are used to set variable values for different environments during the apply process.
- Access the AWS Management Console or use the AWS CLI to verify the creation of resources in the specified regions and instance types.

6. Clean Up:

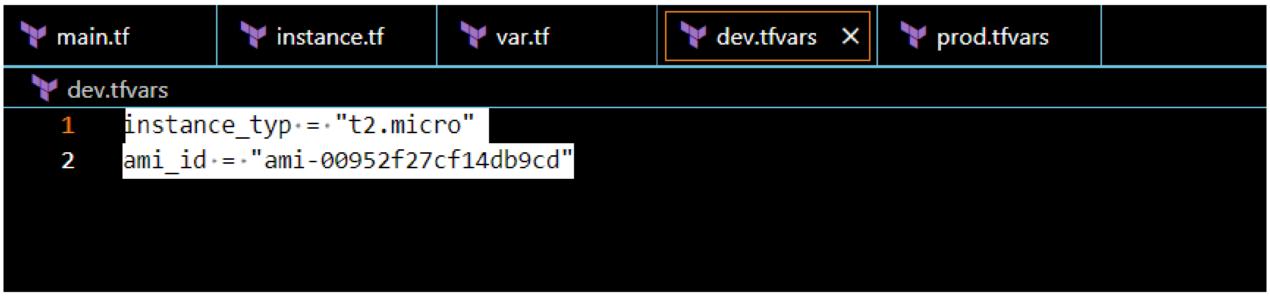
- After testing, you can clean up resources:

```
terraform destroy -var-file=dev.tfvars
terraform destroy -var-file=prod.tfvars
```

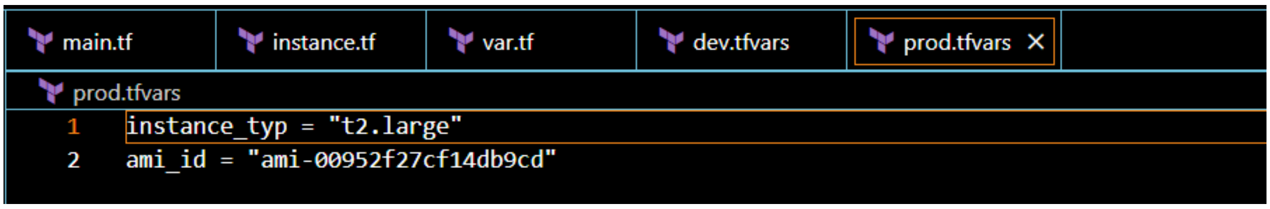
- Confirm the destruction by typing yes.

7. Conclusion:

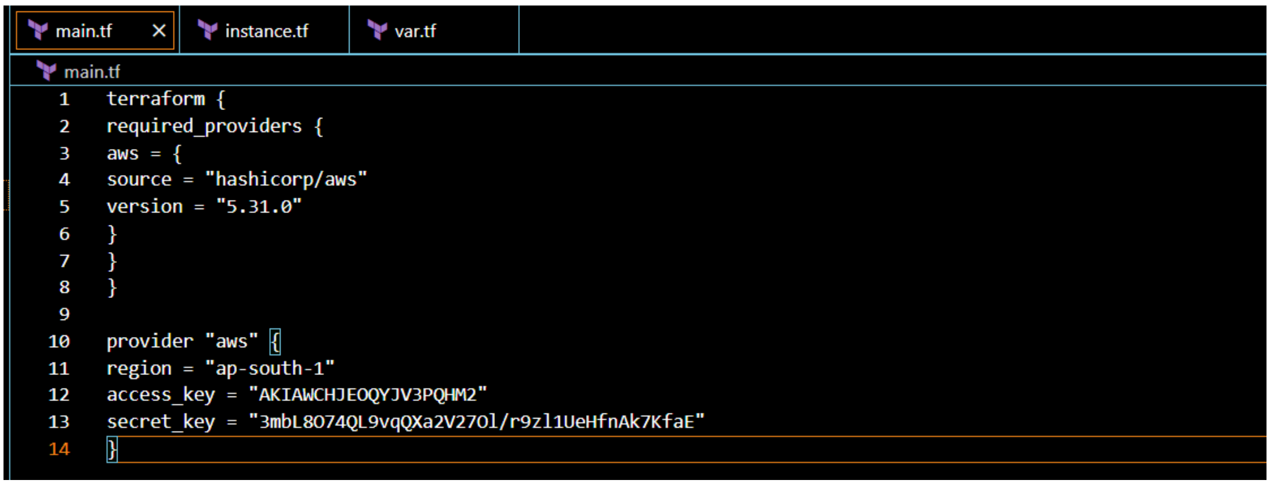
This lab exercise demonstrates how to use multiple tfvars files in Terraform to manage variable values for different environments. It allows you to maintain separate configuration files for different environments, making it easier to manage and maintain your infrastructure code. Experiment with different values in the dev.tfvars and prod.tfvars files to observe how they impact the infrastructure provisioning process for each environment.



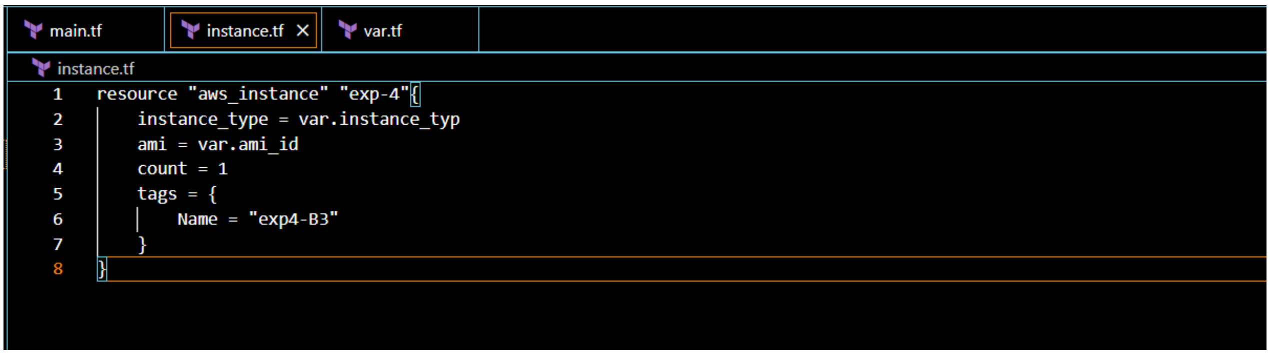
```
instance_typ = "t2.micro"
ami_id = "ami-00952f27cf14db9cd"
```



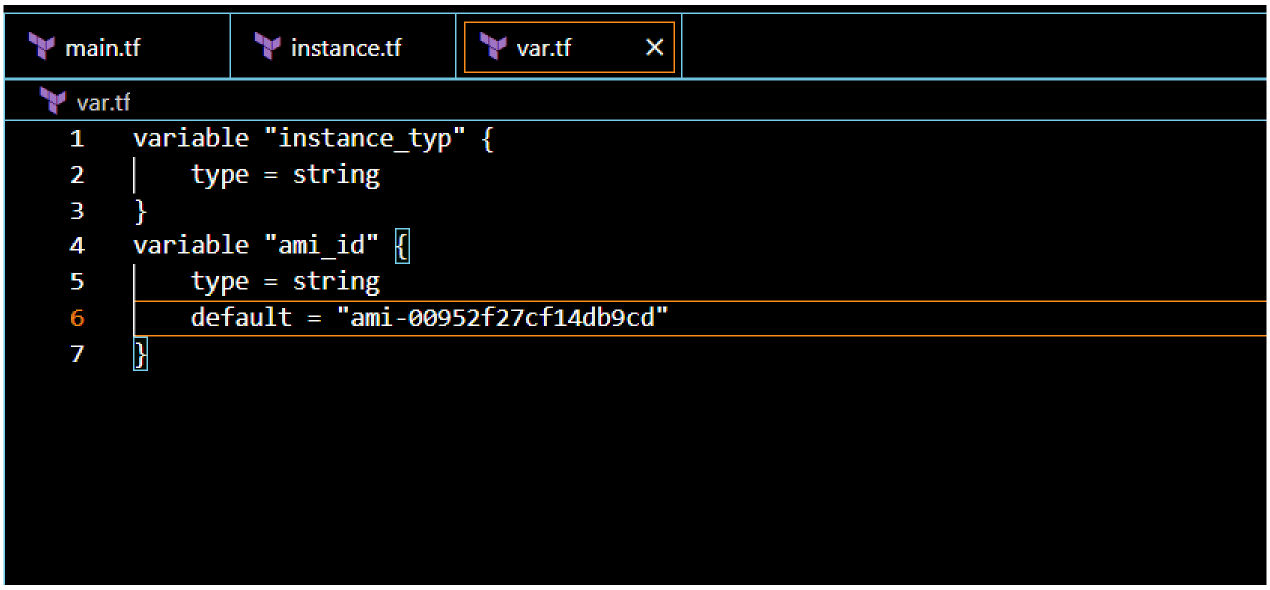
```
instance_typ = "t2.large"
ami_id = "ami-00952f27cf14db9cd"
```



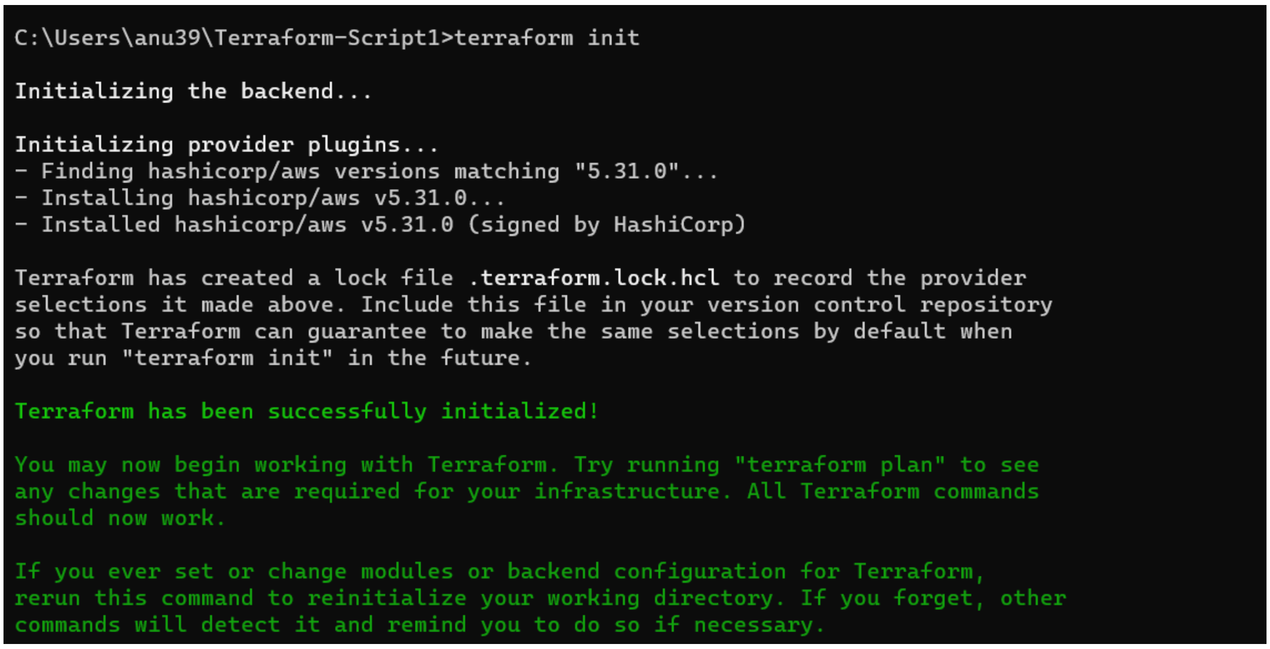
```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}

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

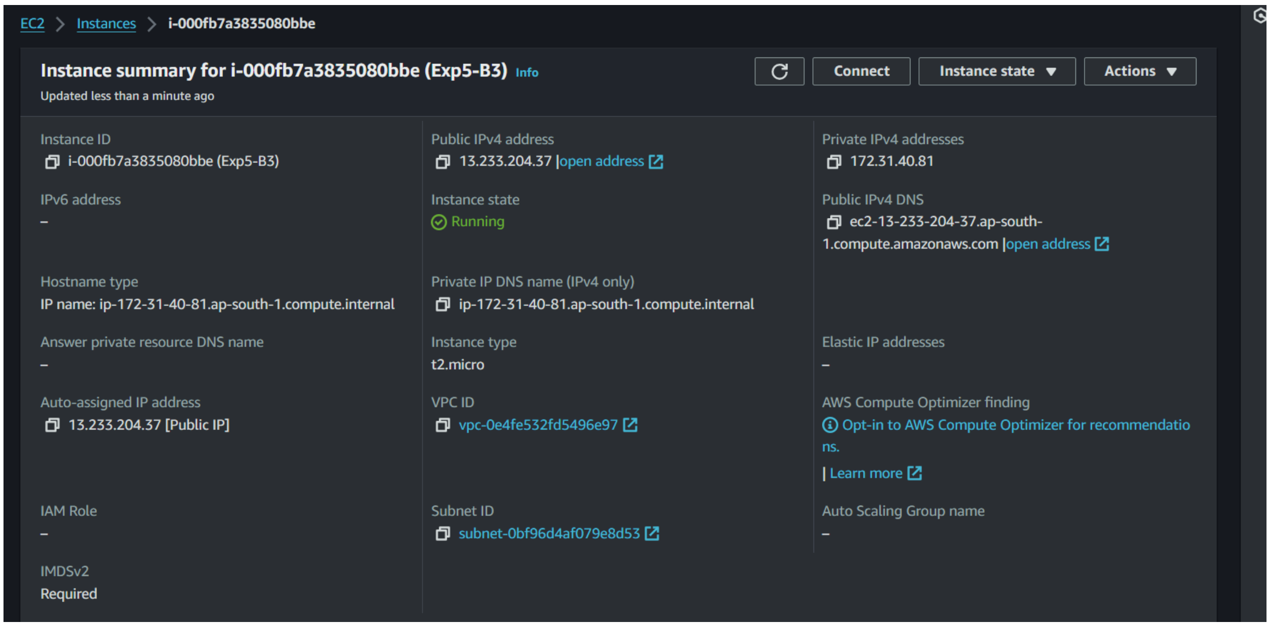

main.tf | instance.tf | var.tf

instance.tf
```
resource "aws_instance" "exp-4"{
    instance_type = var.instance_typ
    ami = var.ami_id
    count = 1
    tags = {
        Name = "exp4-B3"
    }
}
```


main.tf | instance.tf | var.tf

var.tf
```
variable "instance_typ" {
    type = string
}
variable "ami_id" {
    type = string
    default = "ami-00952f27cf14db9cd"
}
```


```
C:\Users\anu39\Terraform-Script1>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.
```


EC2 > Instances > i-000fb7a3835080bbe

Instance summary for i-000fb7a3835080bbe (Exp5-B3) Info

Updated less than a minute ago

Connect | Instance state | Actions

Instance ID
i-000fb7a3835080bbe (Exp5-B3)

Public IPv4 address
13.233.204.37 | open address

Private IPv4 addresses
172.31.40.81

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-13-233-204-37.ap-south-1.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-40-81.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-40-81.ap-south-1.compute.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
13.233.204.37 [Public IP]

VPC ID
vpc-0e4fe532fd5496e97

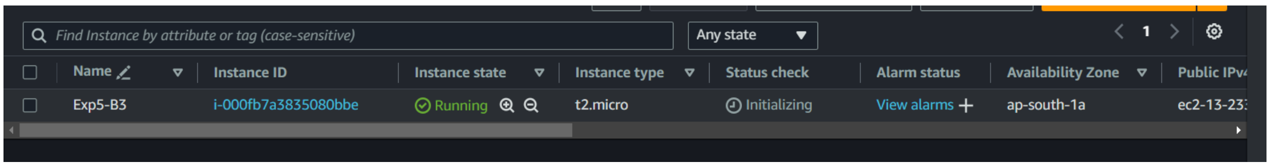
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

IAM Role
–

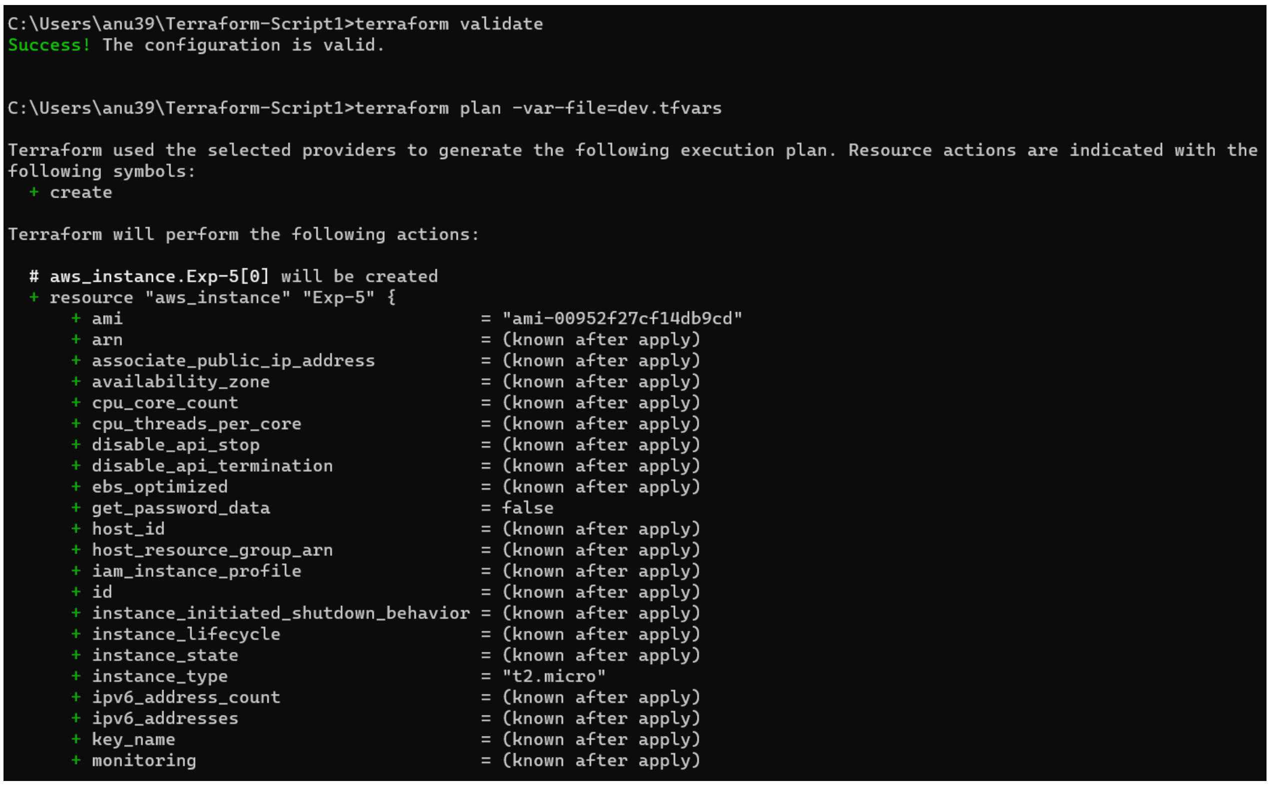
Subnet ID
subnet-0bf96d4af079e8d53

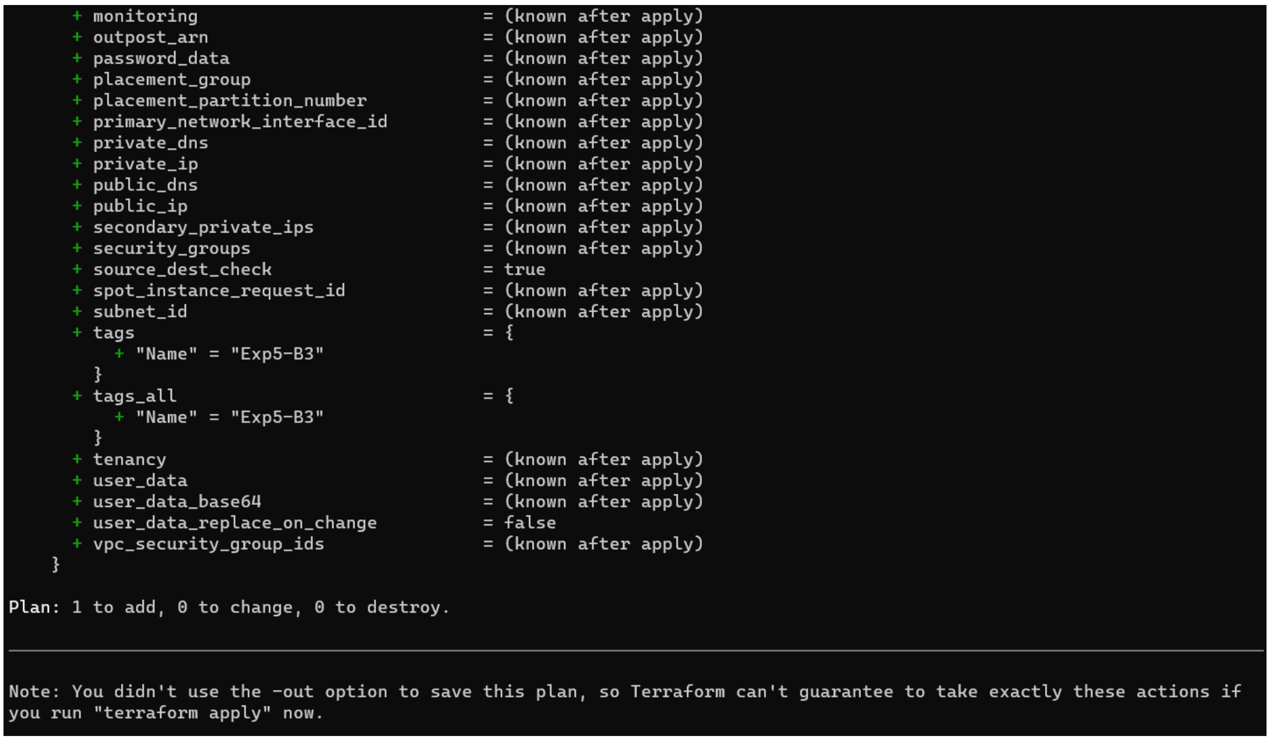
Auto Scaling Group name
–

IMDSv2
Required

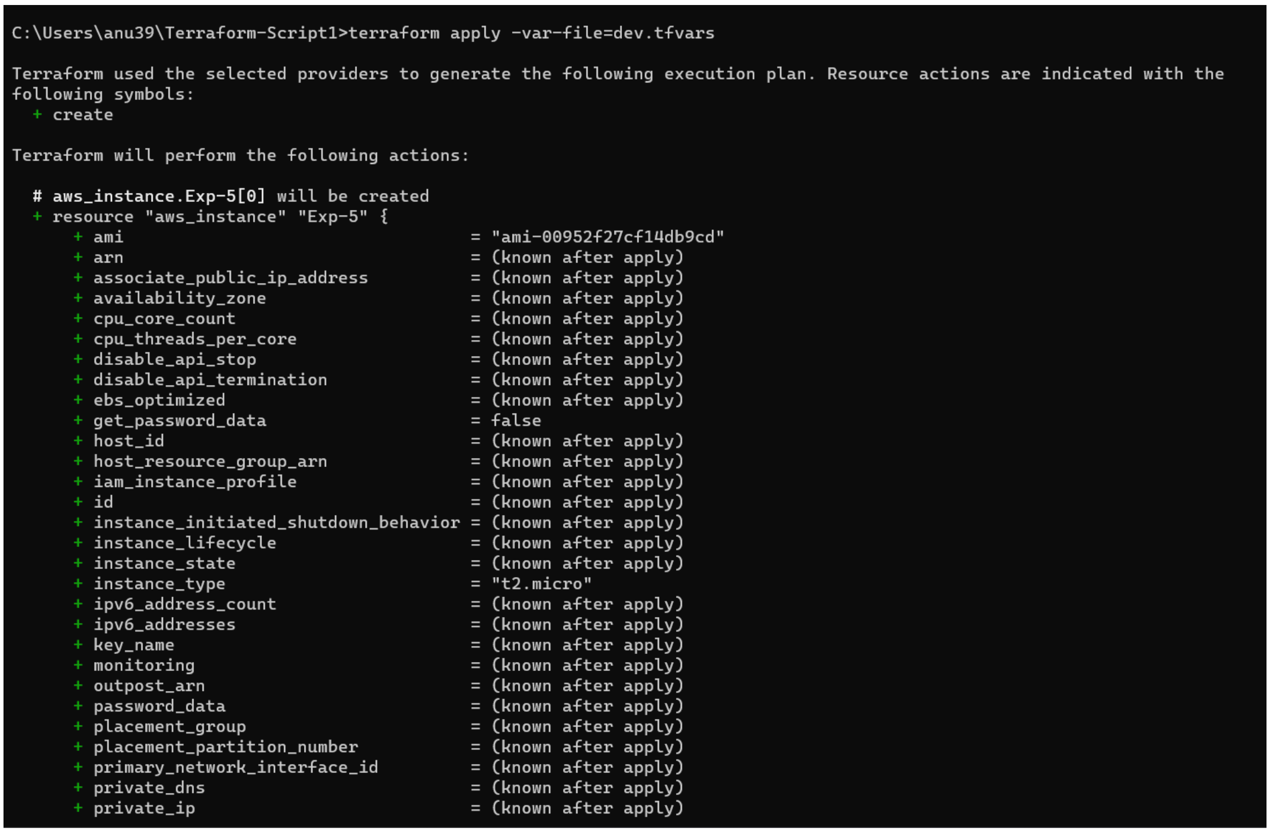

Find Instance by attribute or tag (case-sensitive) | Any state

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Exp5-B3	i-000fb7a3835080bbe	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-23

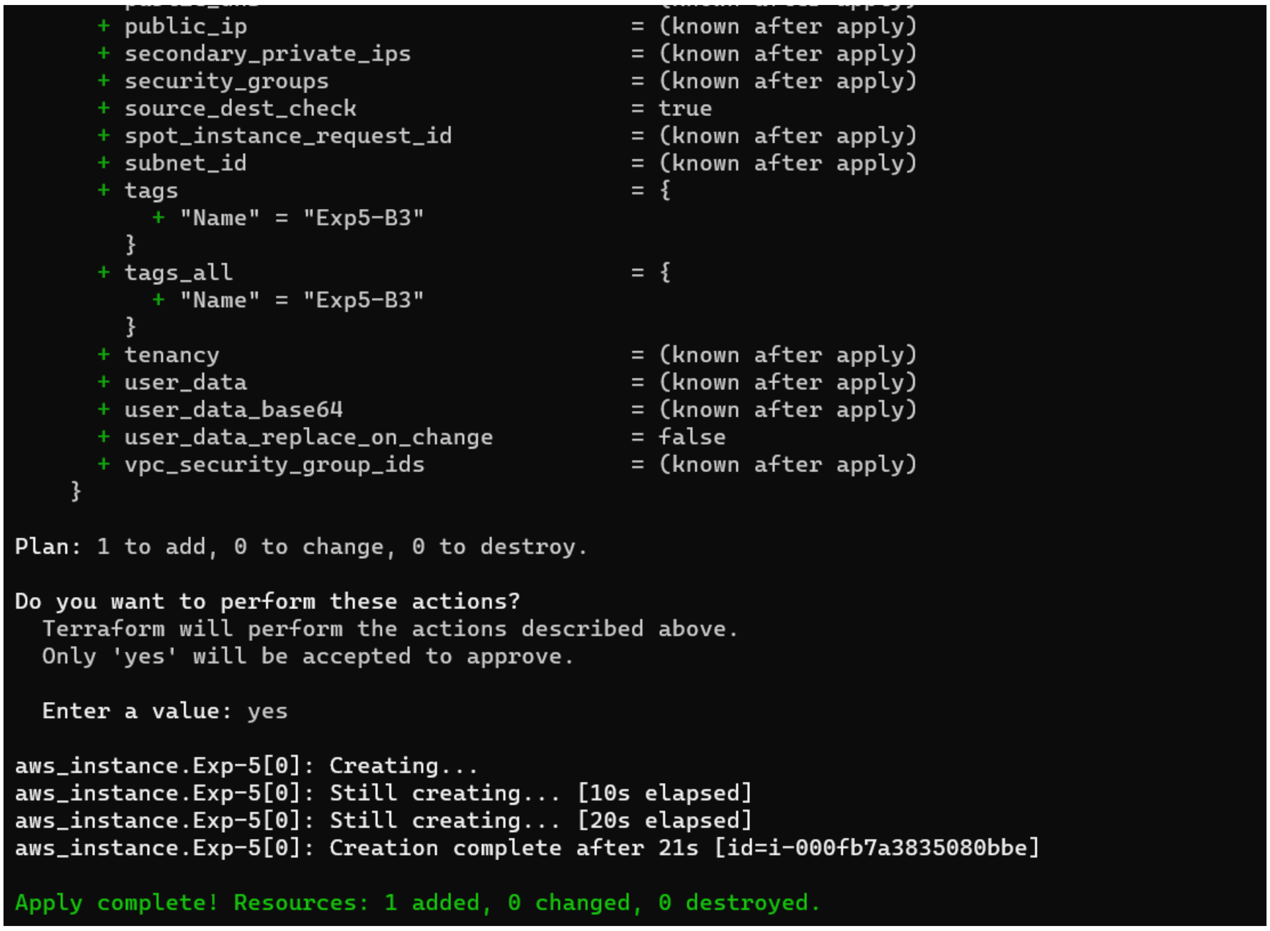

```
C:\Users\anu39\Terraform-Script1>terraform validate
Success! The configuration is valid.


C:\Users\anu39\Terraform-Script1>terraform plan -var-file=dev.tfvars

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.Exp-5[0] will be created
  + resource "aws_instance" "Exp-5" {
      + ami                                  = "ami-00952f27cf14db9cd"
      + 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)
```



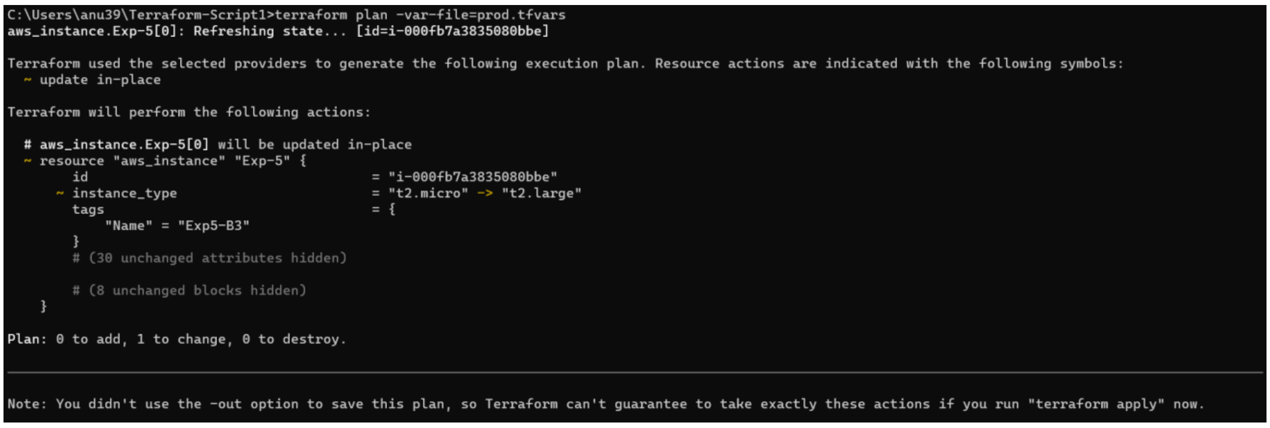
```
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + 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" = "Exp5-B3"
        }
      + tags_all                             = {
          + "Name" = "Exp5-B3"
        }
      + 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: 1 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.
```



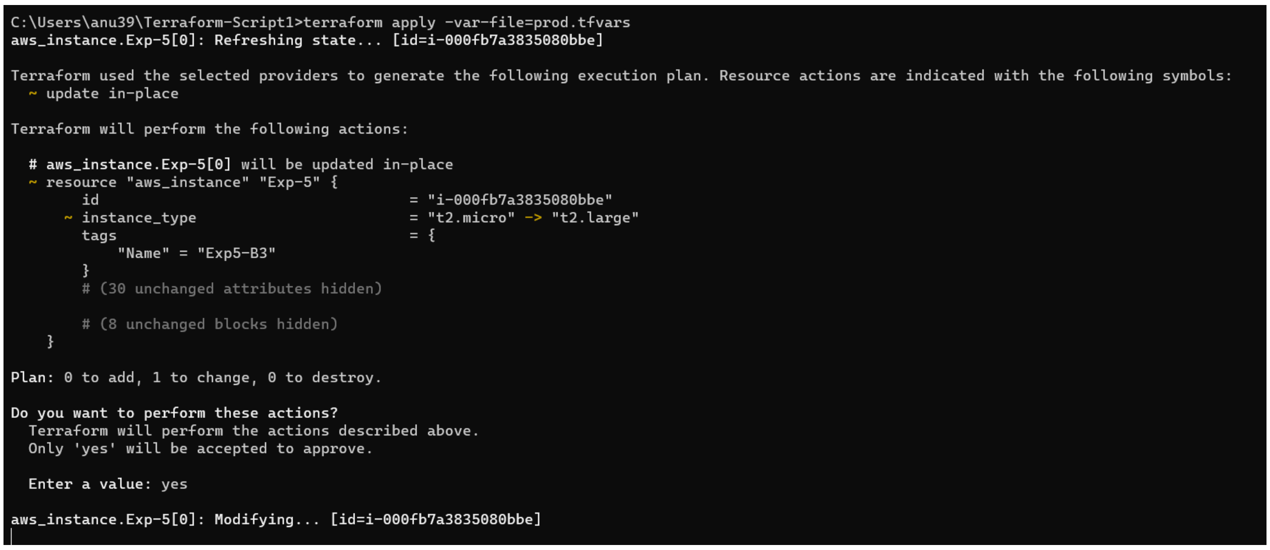
```
C:\Users\anu39\Terraform-Script1>terraform apply -var-file=dev.tfvars

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.Exp-5[0] will be created
  + resource "aws_instance" "Exp-5" {
      + ami                                  = "ami-00952f27cf14db9cd"
      + 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)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
```



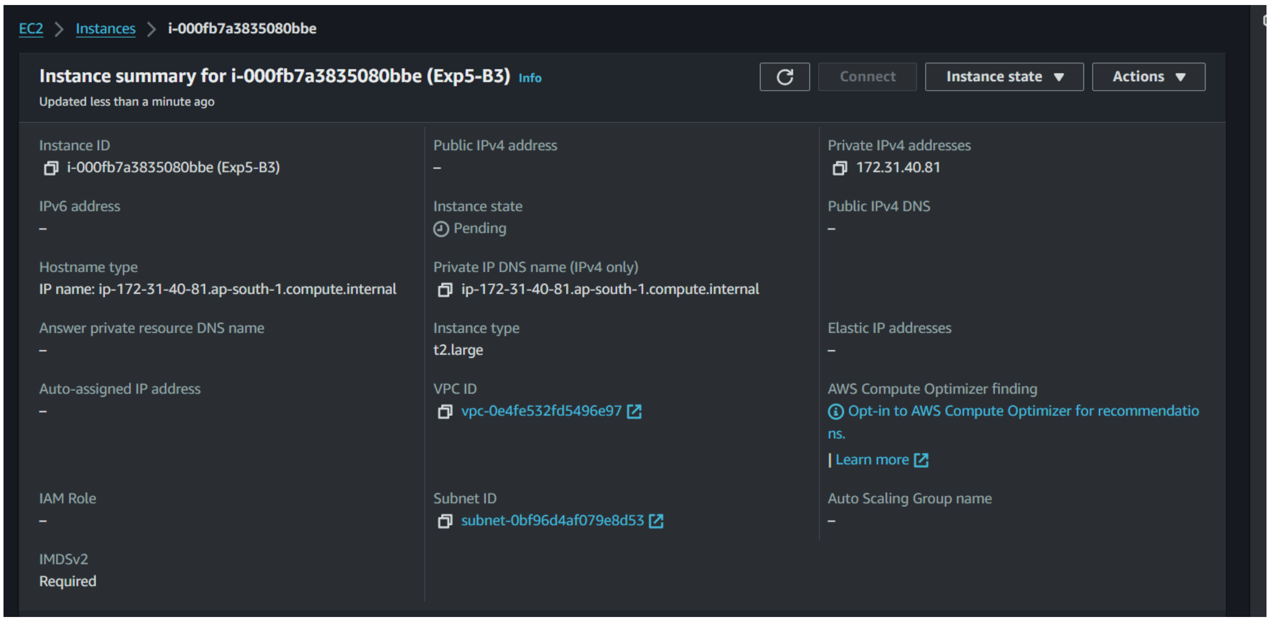
```
      + public_ip                                = (known after apply)
      + secondary_private_ips                    = (known after apply)
      + security_groups                          = (known after apply)
      + source_dest_check                        = true
      + spot_instance_request_id                 = (known after apply)
      + subnet_id                                = (known after apply)
      + tags                                     = {
          + "Name" = "Exp5-B3"
        }
      + tags_all                                 = {
          + "Name" = "Exp5-B3"
        }
      + 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: 1 to add, 0 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_instance.Exp-5[0]: Creating...
aws_instance.Exp-5[0]: Still creating... [10s elapsed]
aws_instance.Exp-5[0]: Still creating... [20s elapsed]
aws_instance.Exp-5[0]: Creation complete after 21s [id=i-000fb7a3835080bbe]

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



```
C:\Users\anu39\Terraform-Script1>terraform plan -var-file=prod.tfvars
aws_instance.Exp-5[0]: Refreshing state... [id=i-000fb7a3835080bbe]

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:

  # aws_instance.Exp-5[0] will be updated in-place
  ~ resource "aws_instance" "Exp-5" {
        id                                   = "i-000fb7a3835080bbe"
      ~ instance_type                        = "t2.micro" -> "t2.large"
        tags                                 = {
            "Name" = "Exp5-B3"
        }
        # (30 unchanged attributes hidden)

        # (8 unchanged blocks hidden)
    }

Plan: 0 to add, 1 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.
```



```
C:\Users\anu39\Terraform-Script1>terraform apply -var-file=prod.tfvars
aws_instance.Exp-5[0]: Refreshing state... [id=i-000fb7a3835080bbe]

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:

  # aws_instance.Exp-5[0] will be updated in-place
  ~ resource "aws_instance" "Exp-5" {
        id                                   = "i-000fb7a3835080bbe"
      ~ instance_type                        = "t2.micro" -> "t2.large"
        tags                                 = {
            "Name" = "Exp5-B3"
        }
        # (30 unchanged attributes hidden)

        # (8 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_instance.Exp-5[0]: Modifying... [id=i-000fb7a3835080bbe]
```



EC2 > Instances > i-000fb7a3835080bbe

Instance summary for i-000fb7a3835080bbe (Exp5-B3) Info

Updated less than a minute ago

Instance ID
i-000fb7a3835080bbe (Exp5-B3)

Public IPv4 address
–

Private IPv4 addresses
172.31.40.81

IPv6 address
–

Instance state
Pending

Public IPv4 DNS
–

Hostname type
IP name: ip-172-31-40-81.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-40-81.ap-south-1.compute.internal

Answer private resource DNS name
–

Instance type
t2.large

Elastic IP addresses
–

Auto-assigned IP address
–

VPC ID
vpc-0e4fe532fd5496e97

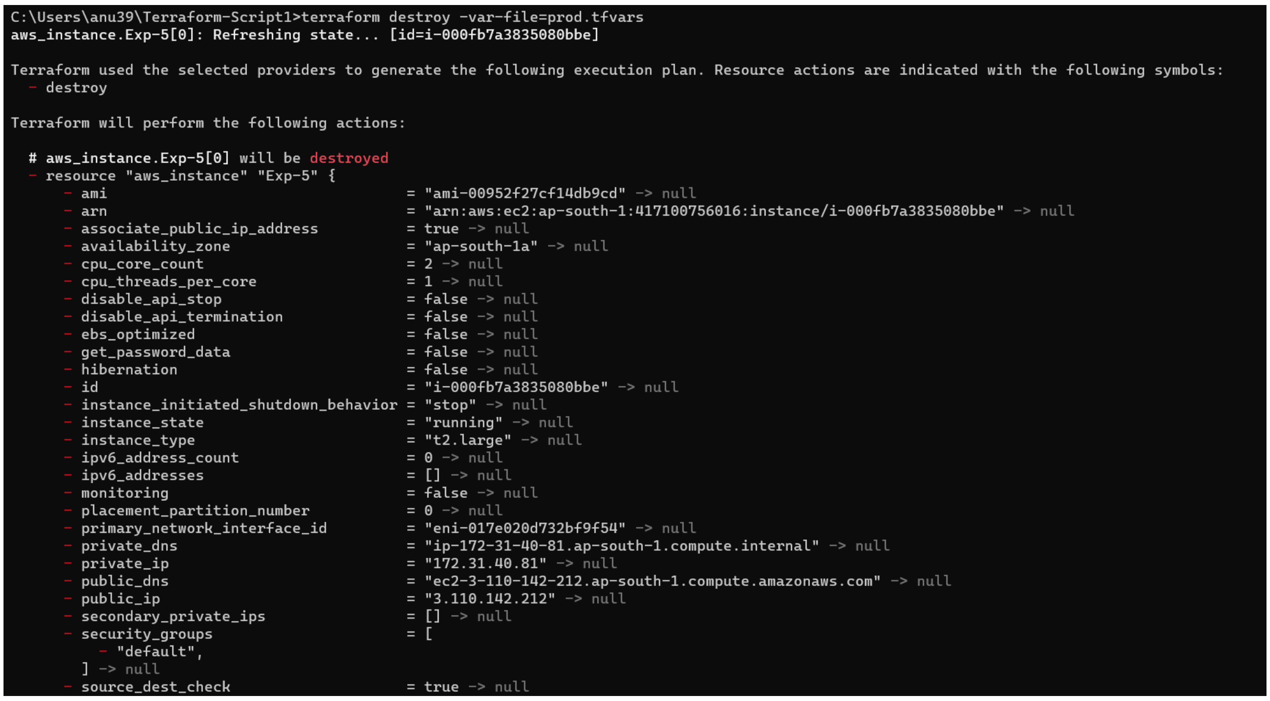
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

IAM Role
–

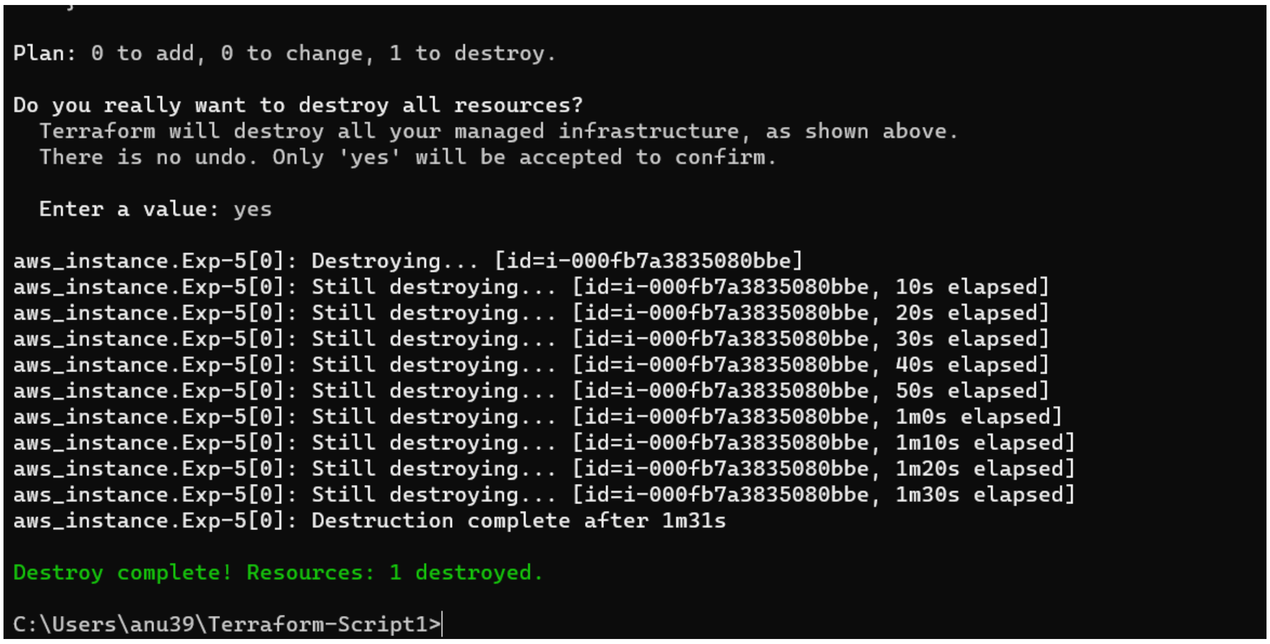
Subnet ID
subnet-0bf96d4af079e8d53

Auto Scaling Group name
–

IMDSv2
Required



```
C:\Users\anu39\Terraform-Script1>terraform destroy -var-file=prod.tfvars
aws_instance.Exp-5[0]: Refreshing state... [id=i-000fb7a3835080bbe]

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:

  # aws_instance.Exp-5[0] will be destroyed
  - resource "aws_instance" "Exp-5" {
      - ami                                  = "ami-00952f27cf14db9cd" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:417100756016:instance/i-000fb7a3835080bbe" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1a" -> null
      - cpu_core_count                       = 2 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-000fb7a3835080bbe" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.large" -> null
      - ipv6_address_count                   = 0 -> null
      - ipv6_addresses                       = [] -> null
      - monitoring                           = false -> null
      - placement_partition_number           = 0 -> null
      - primary_network_interface_id         = "eni-017e020d732bf9f54" -> null
      - private_dns                          = "ip-172-31-40-81.ap-south-1.compute.internal" -> null
      - private_ip                           = "172.31.40.81" -> null
      - public_dns                           = "ec2-3-110-142-212.ap-south-1.compute.amazonaws.com" -> null
      - public_ip                            = "3.110.142.212" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [
          - "default",
        ] -> null
      - source_dest_check                    = true -> null
```



```
Plan: 0 to add, 0 to change, 1 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

aws_instance.Exp-5[0]: Destroying... [id=i-000fb7a3835080bbe]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 10s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 20s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 30s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 40s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 50s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 1m0s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 1m10s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 1m20s elapsed]
aws_instance.Exp-5[0]: Still destroying... [id=i-000fb7a3835080bbe, 1m30s elapsed]
aws_instance.Exp-5[0]: Destruction complete after 1m31s

Destroy complete! Resources: 1 destroyed.

C:\Users\anu39\Terraform-Script1>
```