ASSIGNMENT 01

Qno. 01

● Create one IAM user and one IAM Group using Terraform.

● Make sure you will use variables for names of IAM users and Group.

● Note: - Below files are required. - main.tf - variables.tf - your_name_custom.tfvars

Answer:

- ➔ Create main.tf file:

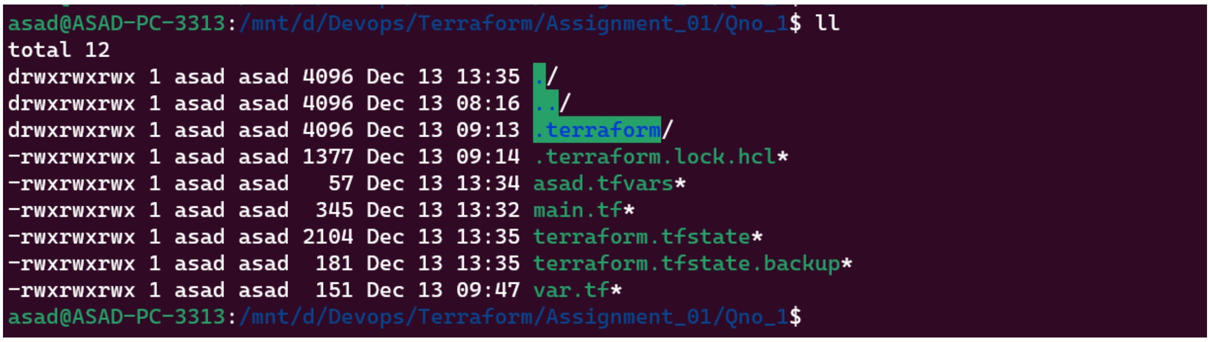
  This file will contain the main Terraform configuration, including the definition of the IAM user and IAM group.

- ➔ Create variables.tf file:
  This file will contain variable declarations that specify the names of the IAM user and IAM group.

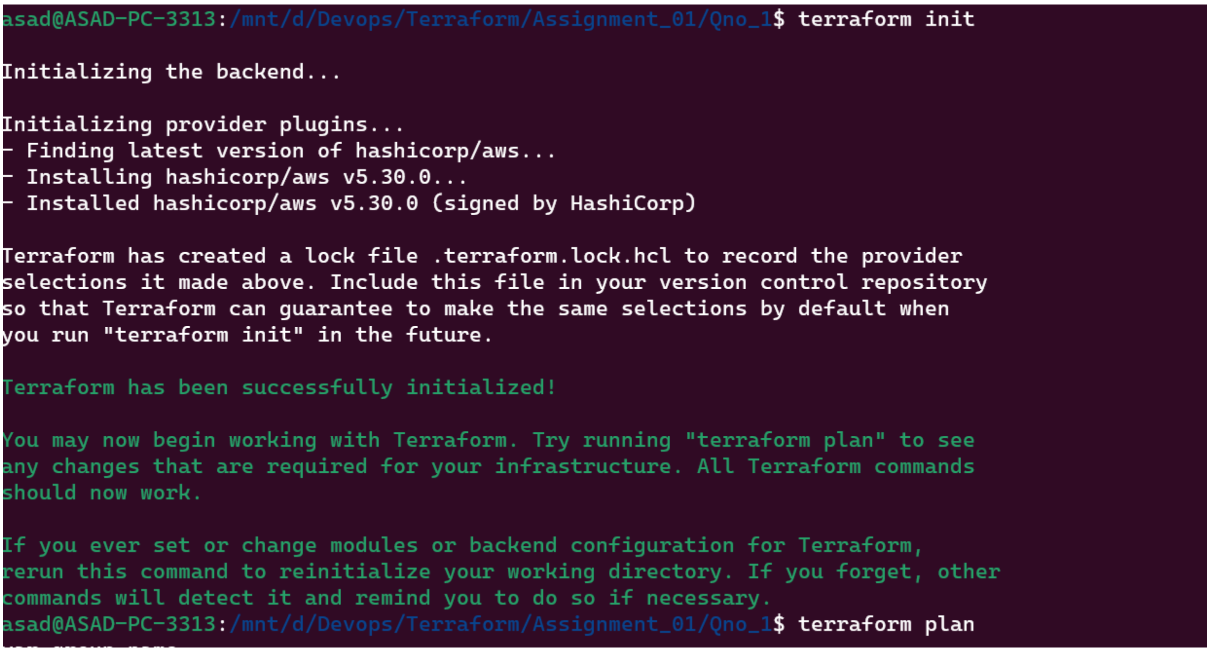
- ➔ Create your_name_custom.tfvars file:

  This file will contain the actual values for the variables declared in variables.tf. Replace "your_name" with your actual name in the filename.

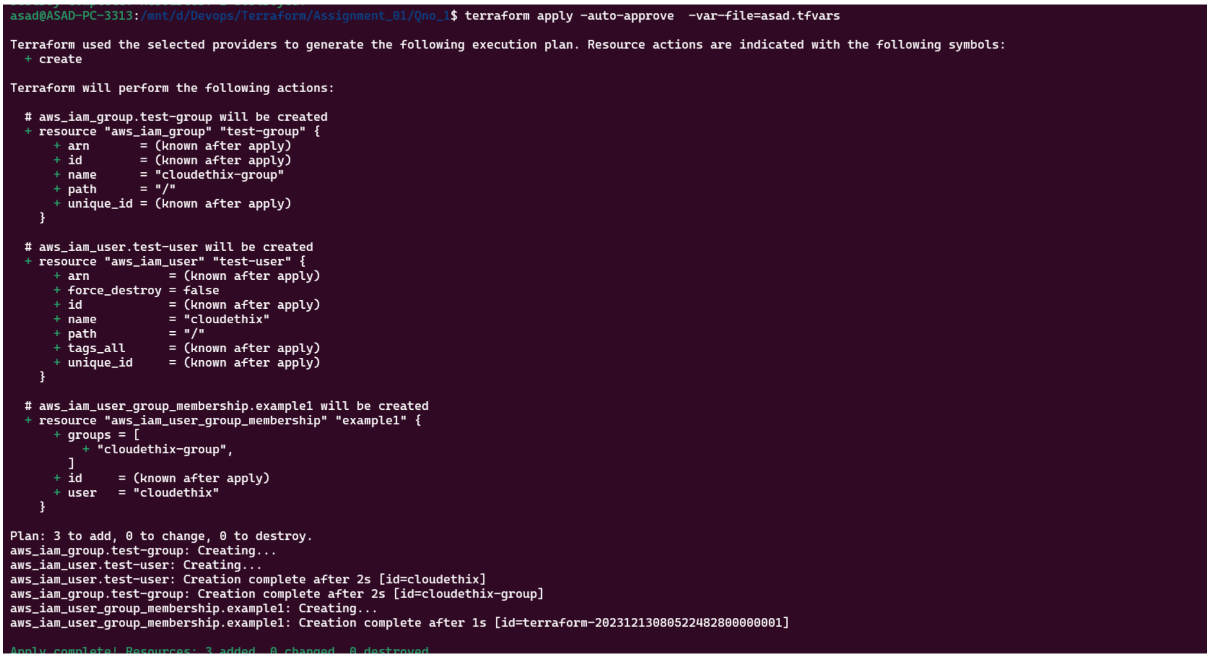
- ➔ Run Terraform commands:
- ➔ Initialize Terraform with terraform init.
- ➔ Apply the configuration using terraform apply -var-file=asad.tfvars.



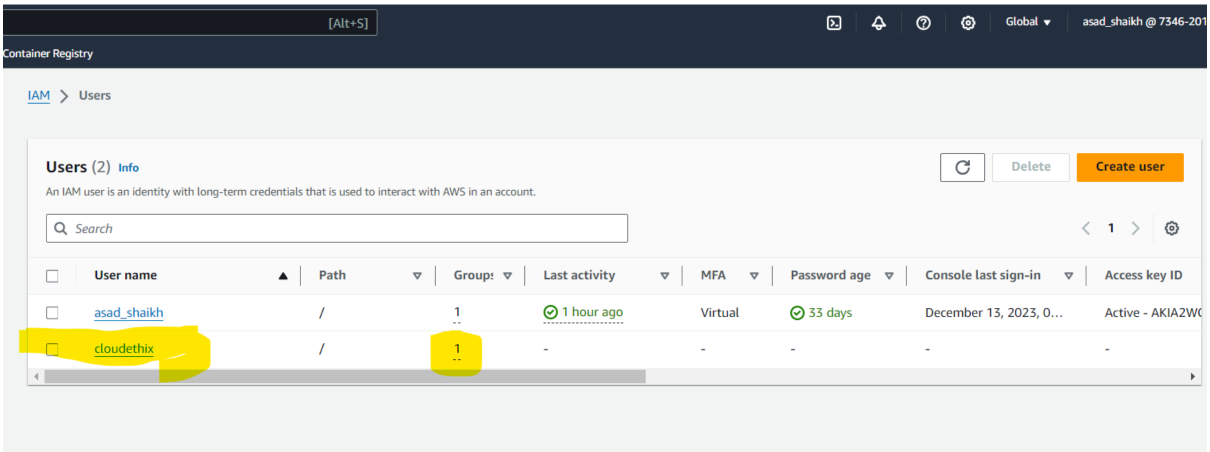
```
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_1$ ll
total 12
drwxrwxrwx 1 asad asad 4096 Dec 13 13:35 ./
drwxrwxrwx 1 asad asad 4096 Dec 13 08:16 ../
drwxrwxrwx 1 asad asad 4096 Dec 13 09:13 .terraform/
-rwxrwxrwx 1 asad asad 1377 Dec 13 09:14 .terraform.lock.hcl*
-rwxrwxrwx 1 asad asad   57 Dec 13 13:34 asad.tfvars*
-rwxrwxrwx 1 asad asad  345 Dec 13 13:32 main.tf*
-rwxrwxrwx 1 asad asad 2104 Dec 13 13:35 terraform.tfstate*
-rwxrwxrwx 1 asad asad  181 Dec 13 13:35 terraform.tfstate.backup*
-rwxrwxrwx 1 asad asad  151 Dec 13 09:47 var.tf*
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_1$
```



```
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_1$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.30.0...
- Installed hashicorp/aws v5.30.0 (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.
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_1$ terraform plan
```



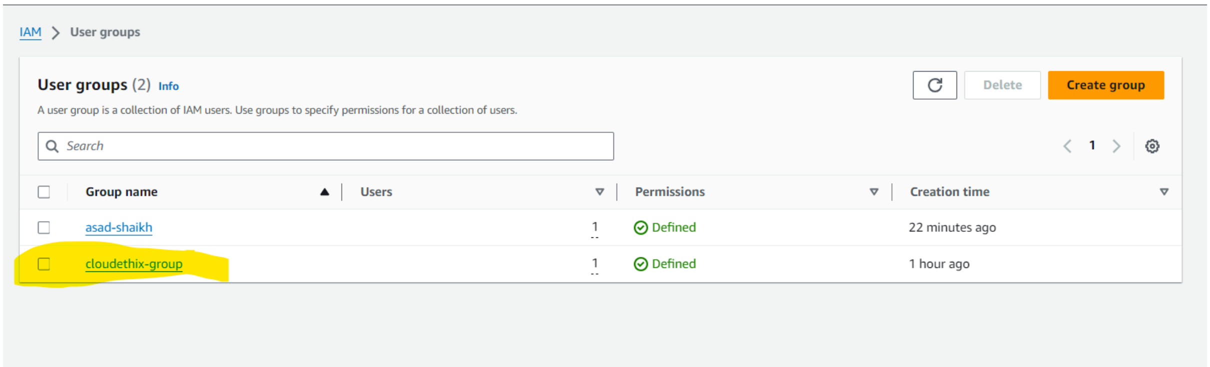
```
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_1$ terraform apply -auto-approve  -var-file=asad.tfvars

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_iam_group.test-group will be created
  + resource "aws_iam_group" "test-group" {
      + arn       = (known after apply)
      + id        = (known after apply)
      + name      = "cloudethix-group"
      + path      = "/"
      + unique_id = (known after apply)
    }

  # aws_iam_user.test-user will be created
  + resource "aws_iam_user" "test-user" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "cloudethix"
      + path          = "/"
      + tags_all      = (known after apply)
      + unique_id     = (known after apply)
    }

  # aws_iam_user_group_membership.example1 will be created
  + resource "aws_iam_user_group_membership" "example1" {
      + groups = [
          + "cloudethix-group",
        ]
      + id     = (known after apply)
      + user   = "cloudethix"
    }

Plan: 3 to add, 0 to change, 0 to destroy.
aws_iam_group.test-group: Creating...
aws_iam_user.test-user: Creating...
aws_iam_user.test-user: Creation complete after 2s [id=cloudethix]
aws_iam_group.test-group: Creation complete after 2s [id=cloudethix-group]
aws_iam_user_group_membership.example1: Creating...
aws_iam_user_group_membership.example1: Creation complete after 1s [id=terraform-20231213080522482800000001]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```



Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name ▲	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
□	asad_shaikh	/	1	✓ 1 hour ago	Virtual	✓ 33 days	December 13, 2023, 0...	Active - AKIA2WC
□	cloudethix	/	1	-	-	-	-	-



IAM > User groups

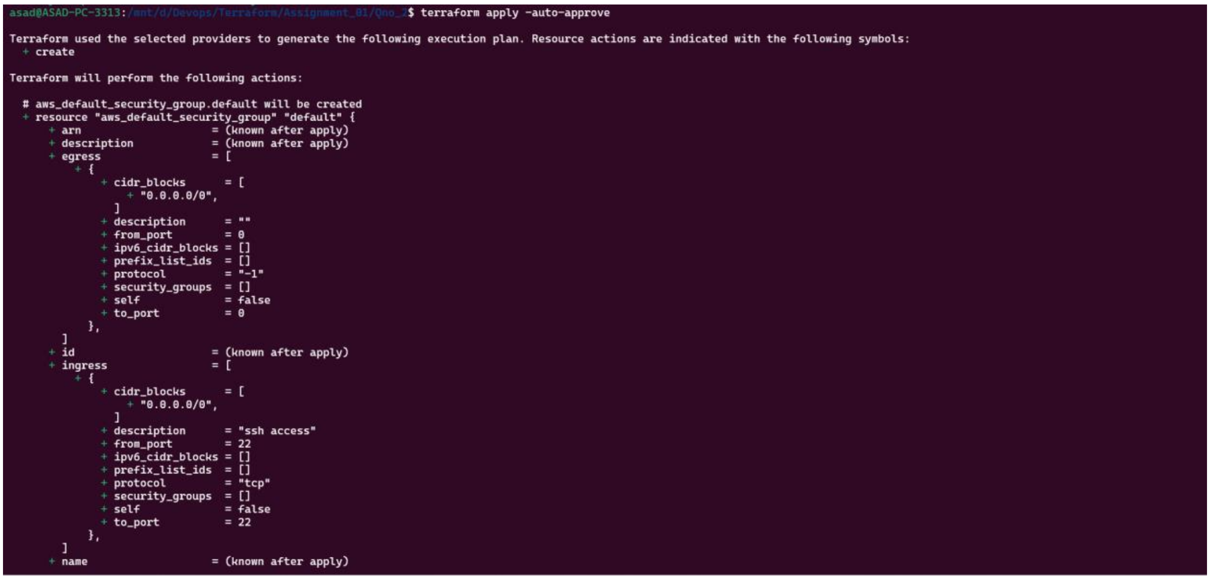
User groups (2) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

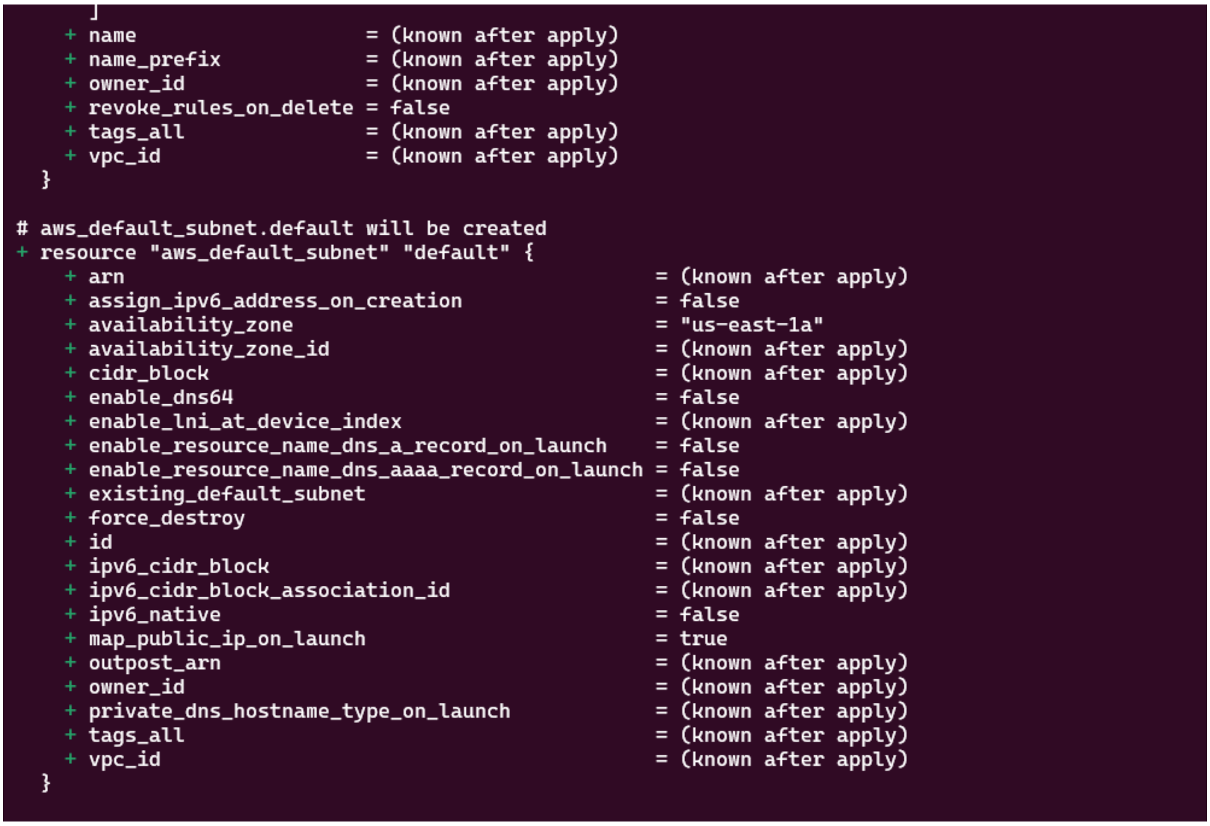
□	Group name	Users	Permissions	Creation time
□	asad-shaikh	1	✓ Defined	22 minutes ago
□	cloudethix-group	1	✓ Defined	1 hour ago

Qno. 2

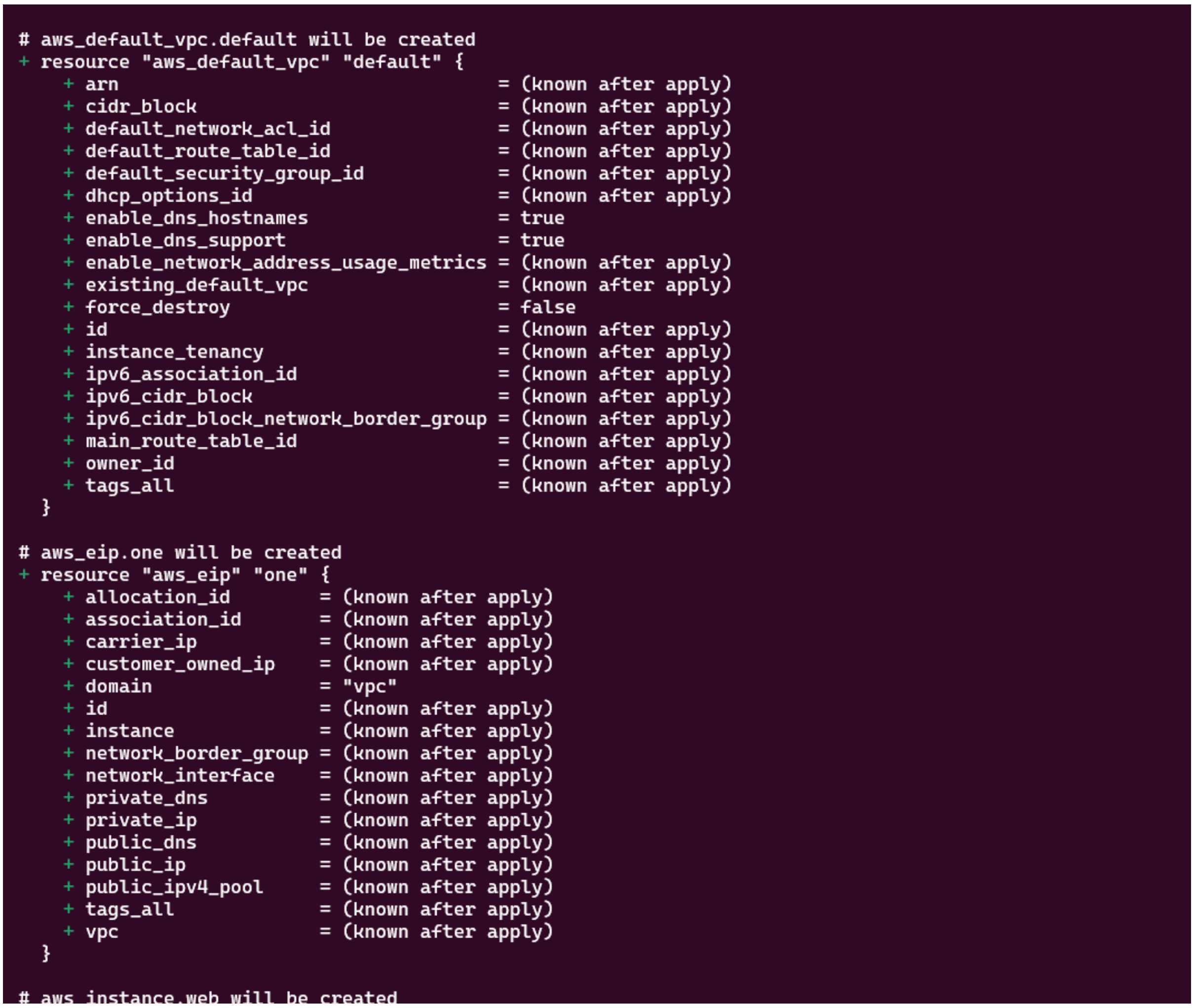
- Create one EC2 Instance and Elastic IP using Terraform
- Map elastic IP with EC2 instance



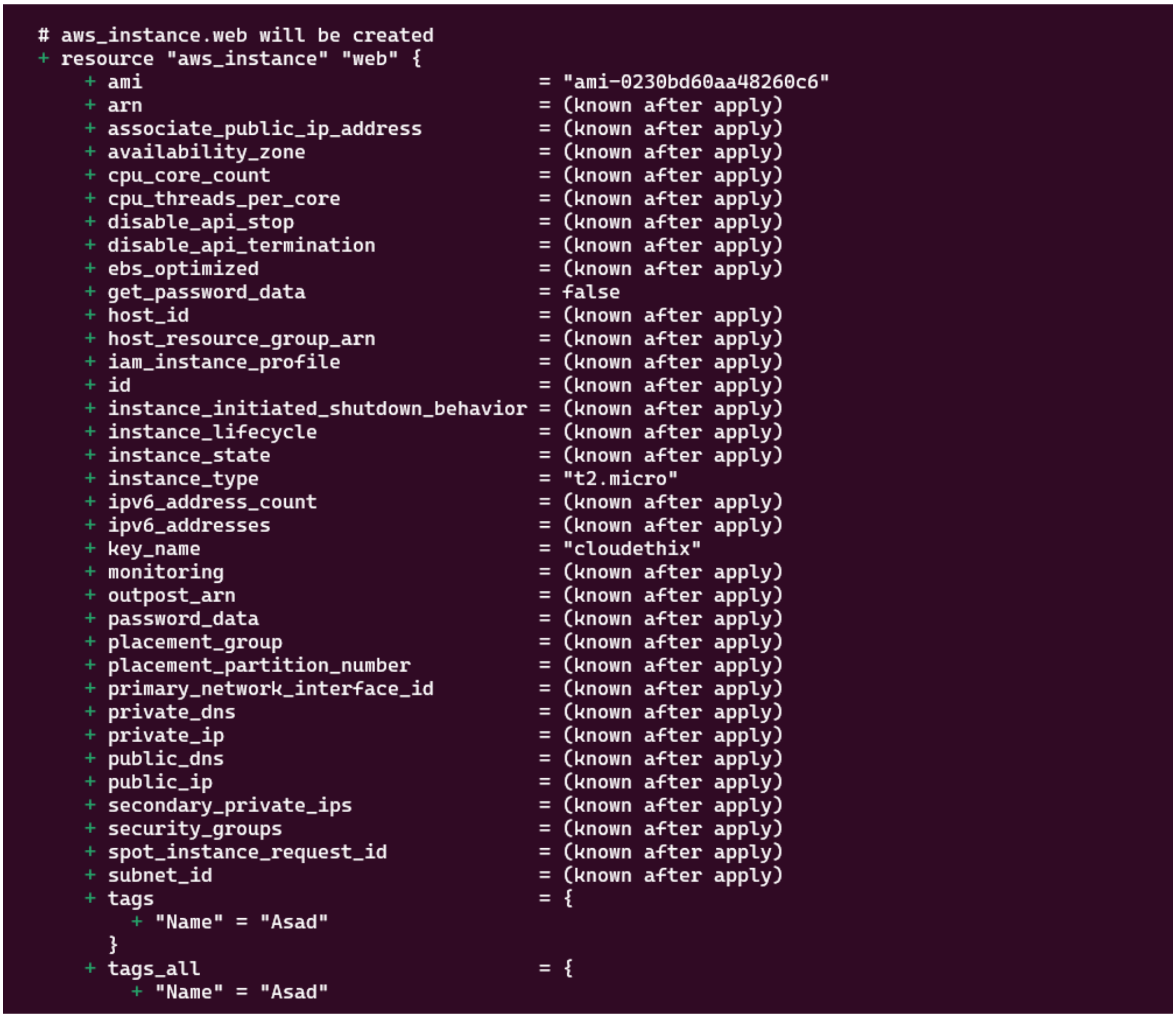
```
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_2$ terraform apply -auto-approve

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_default_security_group.default will be created
  + resource "aws_default_security_group" "default" {
      + arn                    = (known after apply)
      + description            = (known after apply)
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = "ssh access"
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
            },
        ]
      + name                   = (known after apply)
```



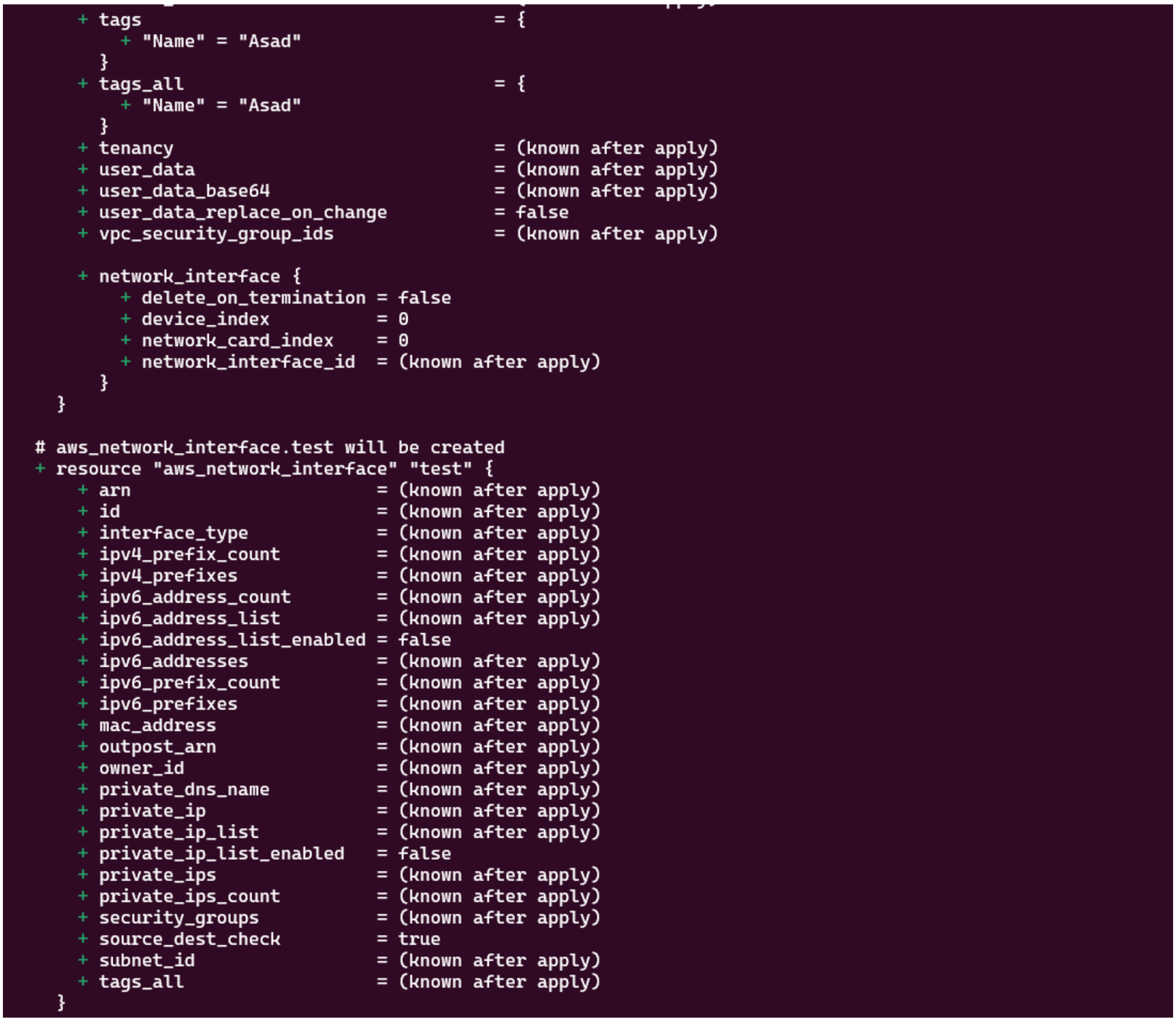
```
      ]
    + name                   = (known after apply)
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags_all               = (known after apply)
    + vpc_id                 = (known after apply)
  }

# aws_default_subnet.default will be created
+ resource "aws_default_subnet" "default" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = (known after apply)
    + enable_dns64                                   = false
    + enable_lni_at_device_index                     = (known after apply)
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + existing_default_subnet                        = (known after apply)
    + force_destroy                                  = false
    + id                                             = (known after apply)
    + ipv6_cidr_block                                = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = true
    + outpost_arn                                    = (known after apply)
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags_all                                       = (known after apply)
    + vpc_id                                         = (known after apply)
  }
```



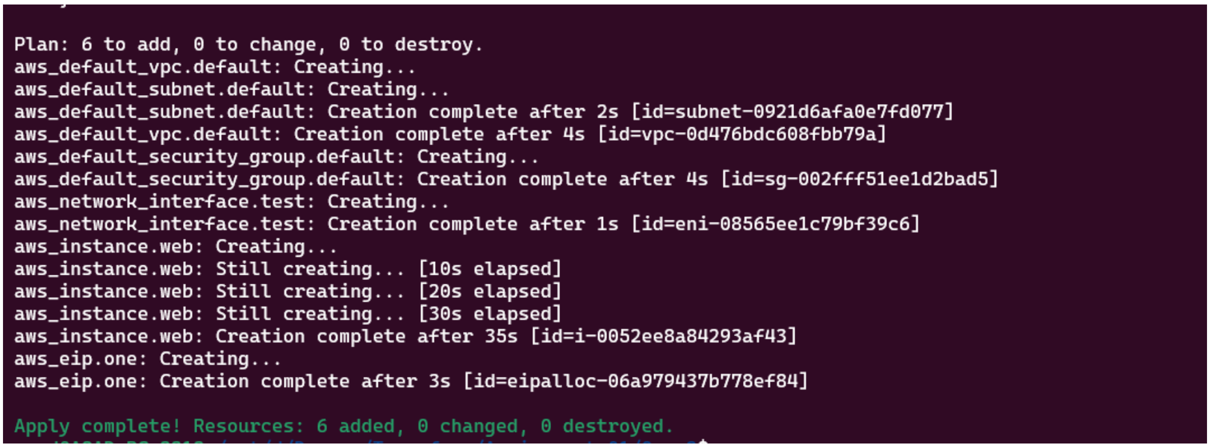
```
# aws_default_vpc.default will be created
+ resource "aws_default_vpc" "default" {
    + arn                                  = (known after apply)
    + cidr_block                           = (known after apply)
    + default_network_acl_id               = (known after apply)
    + default_route_table_id               = (known after apply)
    + default_security_group_id            = (known after apply)
    + dhcp_options_id                      = (known after apply)
    + enable_dns_hostnames                 = true
    + enable_dns_support                   = true
    + enable_network_address_usage_metrics = (known after apply)
    + existing_default_vpc                 = (known after apply)
    + force_destroy                        = false
    + id                                   = (known after apply)
    + instance_tenancy                     = (known after apply)
    + 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_all                             = (known after apply)
  }

# aws_eip.one will be created
+ resource "aws_eip" "one" {
    + allocation_id        = (known after apply)
    + association_id       = (known after apply)
    + carrier_ip           = (known after apply)
    + customer_owned_ip    = (known after apply)
    + domain               = "vpc"
    + id                   = (known after apply)
    + instance             = (known after apply)
    + network_border_group = (known after apply)
    + network_interface    = (known after apply)
    + private_dns          = (known after apply)
    + private_ip           = (known after apply)
    + public_dns           = (known after apply)
    + public_ip            = (known after apply)
    + public_ipv4_pool     = (known after apply)
    + tags_all             = (known after apply)
    + vpc                  = (known after apply)
  }

# aws_instance.web will be created
```



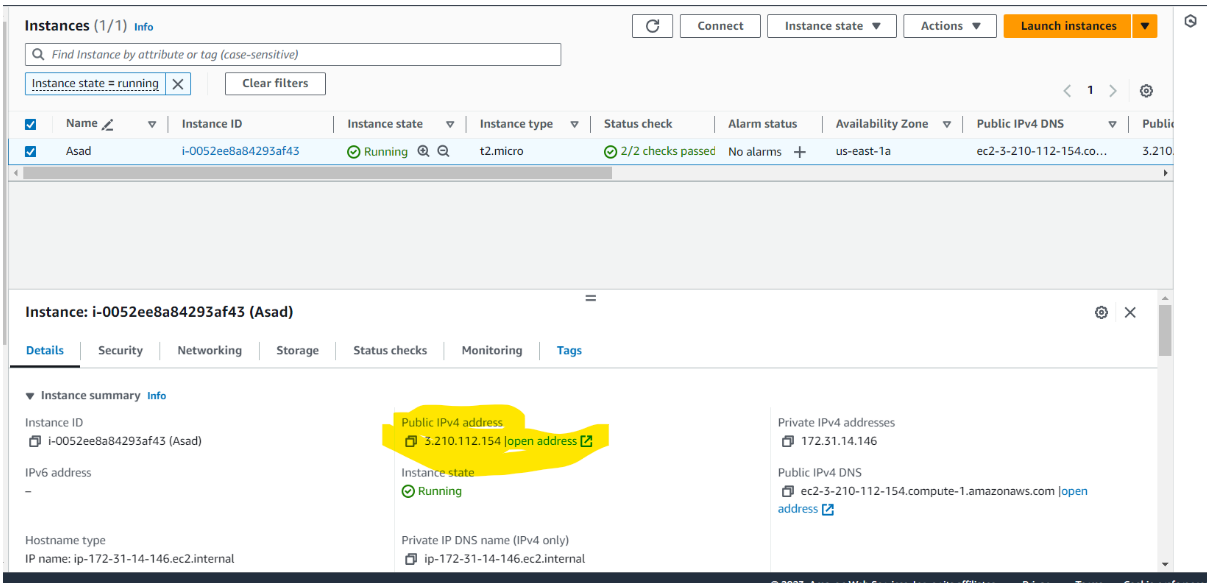
```
# aws_instance.web will be created
+ resource "aws_instance" "web" {
    + ami                                  = "ami-0230bd60aa48260c6"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (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                             = "cloudethix"
    + 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)
    + spot_instance_request_id             = (known after apply)
    + subnet_id                            = (known after apply)
    + tags                                 = {
        + "Name" = "Asad"
      }
    + tags_all                             = {
        + "Name" = "Asad"
```



```
      + tags                                 = {
          + "Name" = "Asad"
        }
      + tags_all                             = {
          + "Name" = "Asad"
        }
      + 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)

      + network_interface {
          + delete_on_termination = false
          + device_index          = 0
          + network_card_index    = 0
          + network_interface_id  = (known after apply)
        }
    }

  # aws_network_interface.test will be created
  + resource "aws_network_interface" "test" {
      + arn                       = (known after apply)
      + id                        = (known after apply)
      + interface_type            = (known after apply)
      + ipv4_prefix_count         = (known after apply)
      + ipv4_prefixes             = (known after apply)
      + ipv6_address_count        = (known after apply)
      + ipv6_address_list         = (known after apply)
      + ipv6_address_list_enabled = false
      + ipv6_addresses            = (known after apply)
      + ipv6_prefix_count         = (known after apply)
      + ipv6_prefixes             = (known after apply)
      + mac_address               = (known after apply)
      + outpost_arn               = (known after apply)
      + owner_id                  = (known after apply)
      + private_dns_name          = (known after apply)
      + private_ip                = (known after apply)
      + private_ip_list           = (known after apply)
      + private_ip_list_enabled   = false
      + private_ips               = (known after apply)
      + private_ips_count         = (known after apply)
      + security_groups           = (known after apply)
      + source_dest_check         = true
      + subnet_id                 = (known after apply)
      + tags_all                  = (known after apply)
    }
```



```
Plan: 6 to add, 0 to change, 0 to destroy.
aws_default_vpc.default: Creating...
aws_default_subnet.default: Creating...
aws_default_subnet.default: Creation complete after 2s [id=subnet-0921d6afa0e7fd077]
aws_default_vpc.default: Creation complete after 4s [id=vpc-0d476bdc608fbb79a]
aws_default_security_group.default: Creating...
aws_default_security_group.default: Creation complete after 4s [id=sg-002fff51ee1d2bad5]
aws_network_interface.test: Creating...
aws_network_interface.test: Creation complete after 1s [id=eni-08565ee1c79bf39c6]
aws_instance.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [20s elapsed]
aws_instance.web: Still creating... [30s elapsed]
aws_instance.web: Creation complete after 35s [id=i-0052ee8a84293af43]
aws_eip.one: Creating...
aws_eip.one: Creation complete after 3s [id=eipalloc-06a979437b778ef84]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
```



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Publi
☑	Asad	i-0052ee8a84293af43	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-210-112-154.co…	3.210.

Instance: i-0052ee8a84293af43 (Asad)

Instance summary Info

Instance ID: i-0052ee8a84293af43 (Asad)

Public IPv4 address: 3.210.112.154 |open address

Private IPv4 addresses: 172.31.14.146

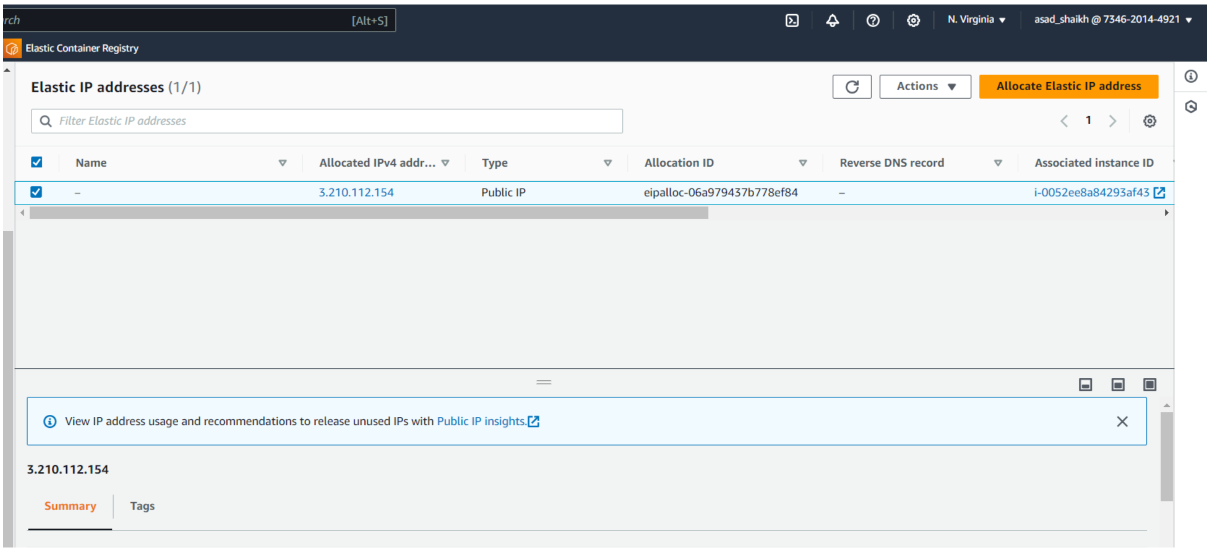
IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-210-112-154.compute-1.amazonaws.com |open address

Hostname type: IP name: ip-172-31-14-146.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-14-146.ec2.internal


rch [Alt+S]
N. Virginia ▼
asad_shaikh @ 7346-2014-4921 ▼
Elastic Container Registry
Actions ▼
Allocate Elastic IP address
Filter Elastic IP addresses
Summary
Tags

Elastic IP addresses (1/1)

☑	Name	Allocated IPv4 addr…	Type	Allocation ID	Reverse DNS record	Associated instance ID
☑	–	3.210.112.154	Public IP	eipalloc-06a979437b778ef84	–	i-0052ee8a84293af43

View IP address usage and recommendations to release unused IPs with Public IP insights

3.210.112.154

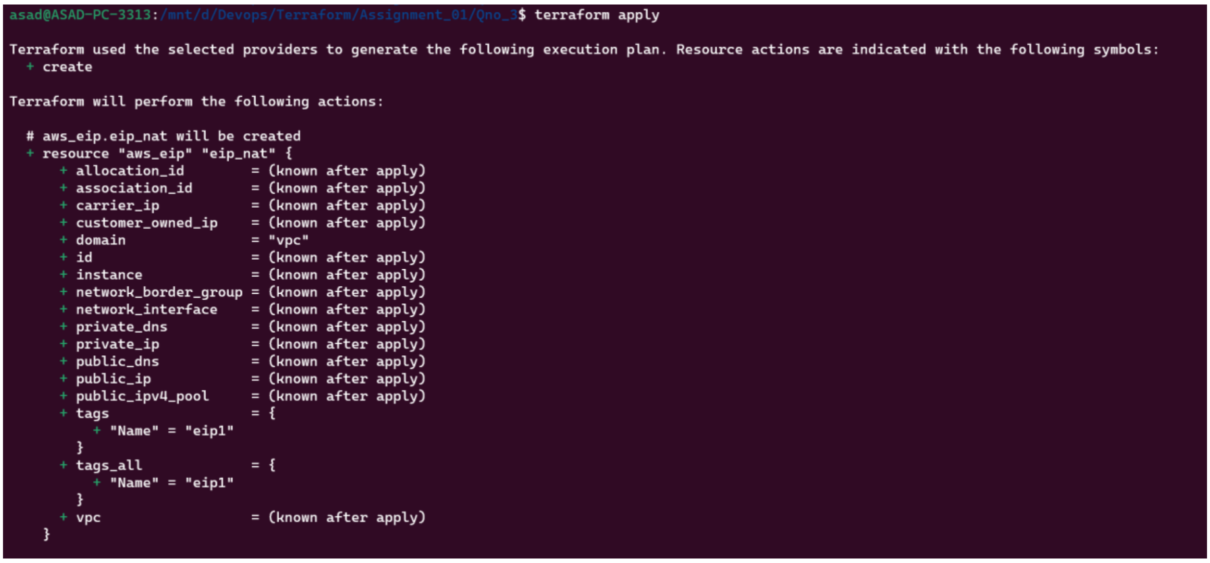
Qno. 3

- Create AWS VPC with Terraform.
- Please follow the given link for more on AWS VPC creation.

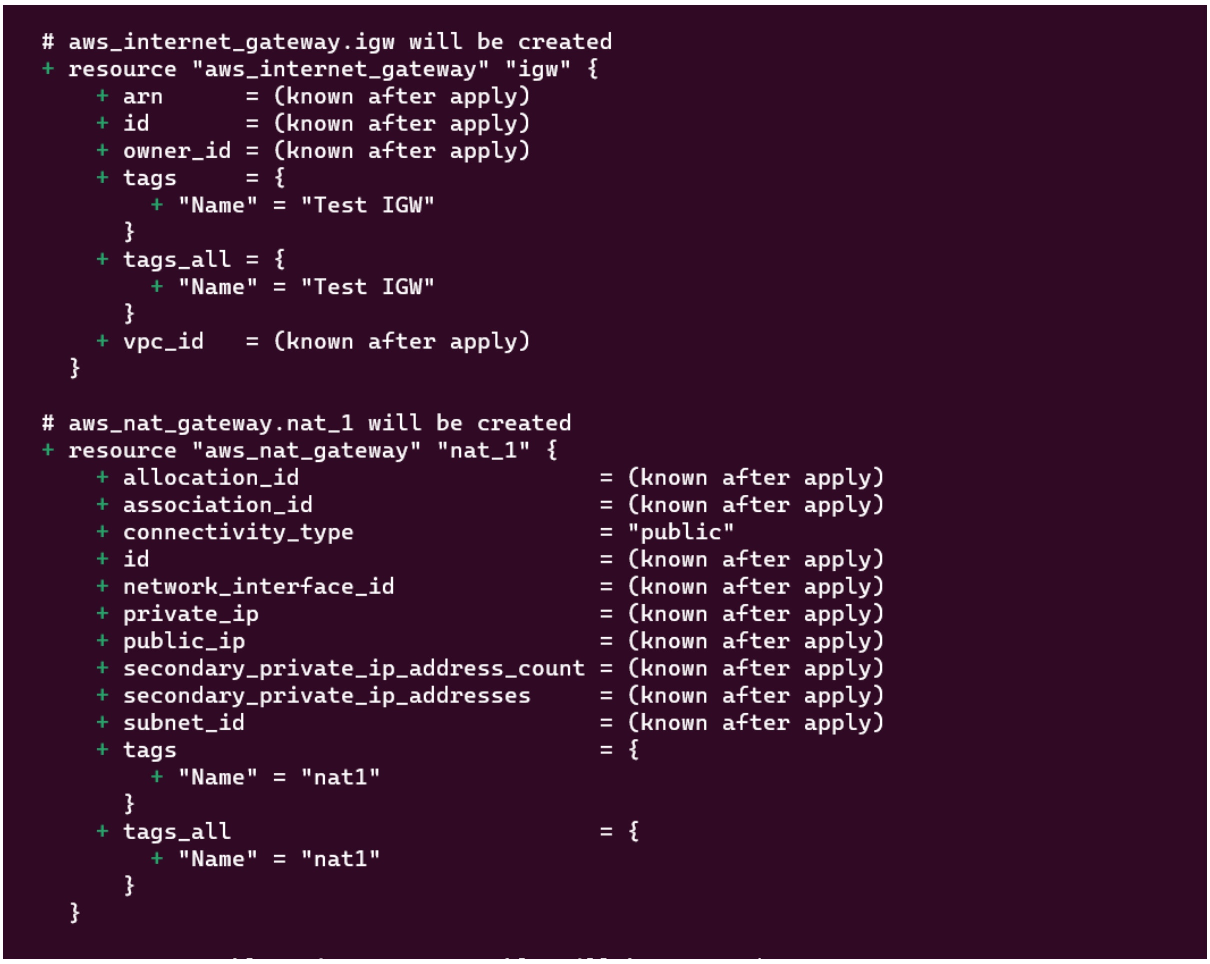
1. Create a VPC.
2. Create 2 Public Subnet & Create 2 Private Subnet.
3. Create IGW (Internet Gateway) & Attach to the VPC.
4. Create Public and Private Route Table.
5. Add IGW in Public Route table (0.0.0.0/0).
6. Add Public Subnet (1a & 1b) in Route table.
7. Create a NAT Gateway in Public Subnet.
8. Add NAT GW into the Private Route Table.
9. Add Private Subnet in Private Route Table.

Answer

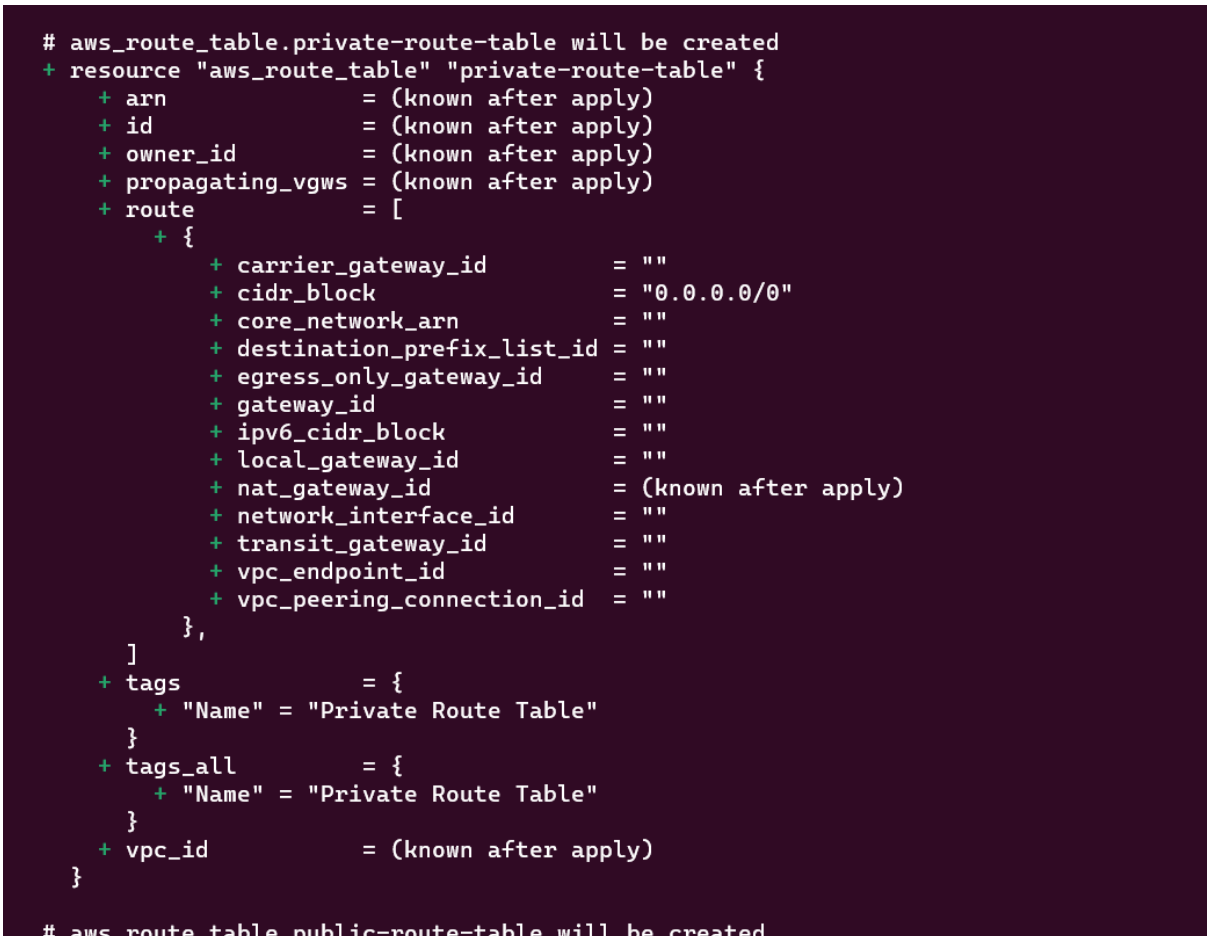
➔ Created VPC



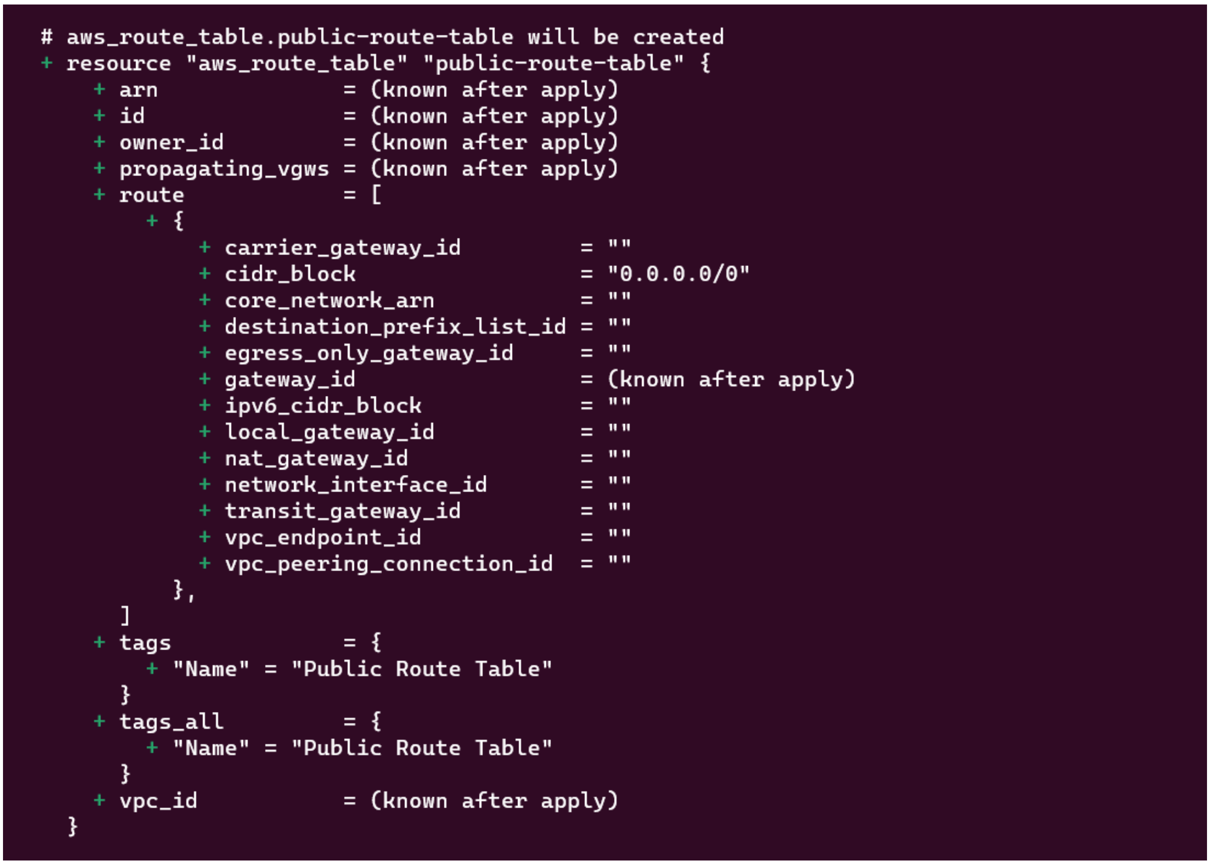
```
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_3$ terraform apply

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:

  # aws_eip.eip_nat will be created
  + resource "aws_eip" "eip_nat" {
      + allocation_id        = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = "vpc"
      + id                   = (known after apply)
      + instance             = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags                 = {
          + "Name" = "eip1"
        }
      + tags_all             = {
          + "Name" = "eip1"
        }
      + vpc                  = (known after apply)
    }
```



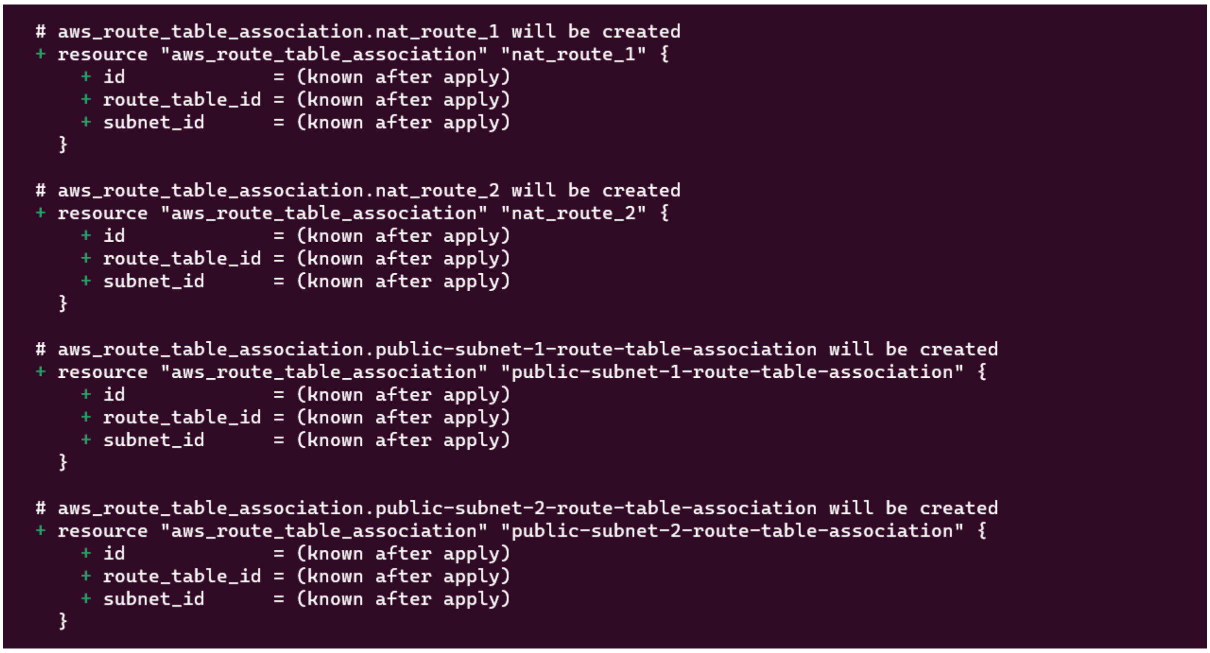
```
  # aws_internet_gateway.igw will be created
  + resource "aws_internet_gateway" "igw" {
      + arn      = (known after apply)
      + id       = (known after apply)
      + owner_id = (known after apply)
      + tags     = {
          + "Name" = "Test IGW"
        }
      + tags_all = {
          + "Name" = "Test IGW"
        }
      + vpc_id   = (known after apply)
    }

  # aws_nat_gateway.nat_1 will be created
  + resource "aws_nat_gateway" "nat_1" {
      + allocation_id                      = (known after apply)
      + association_id                     = (known after apply)
      + connectivity_type                  = "public"
      + id                                 = (known after apply)
      + network_interface_id               = (known after apply)
      + private_ip                         = (known after apply)
      + public_ip                          = (known after apply)
      + secondary_private_ip_address_count = (known after apply)
      + secondary_private_ip_addresses     = (known after apply)
      + subnet_id                          = (known after apply)
      + tags                               = {
          + "Name" = "nat1"
        }
      + tags_all                           = {
          + "Name" = "nat1"
        }
    }
```



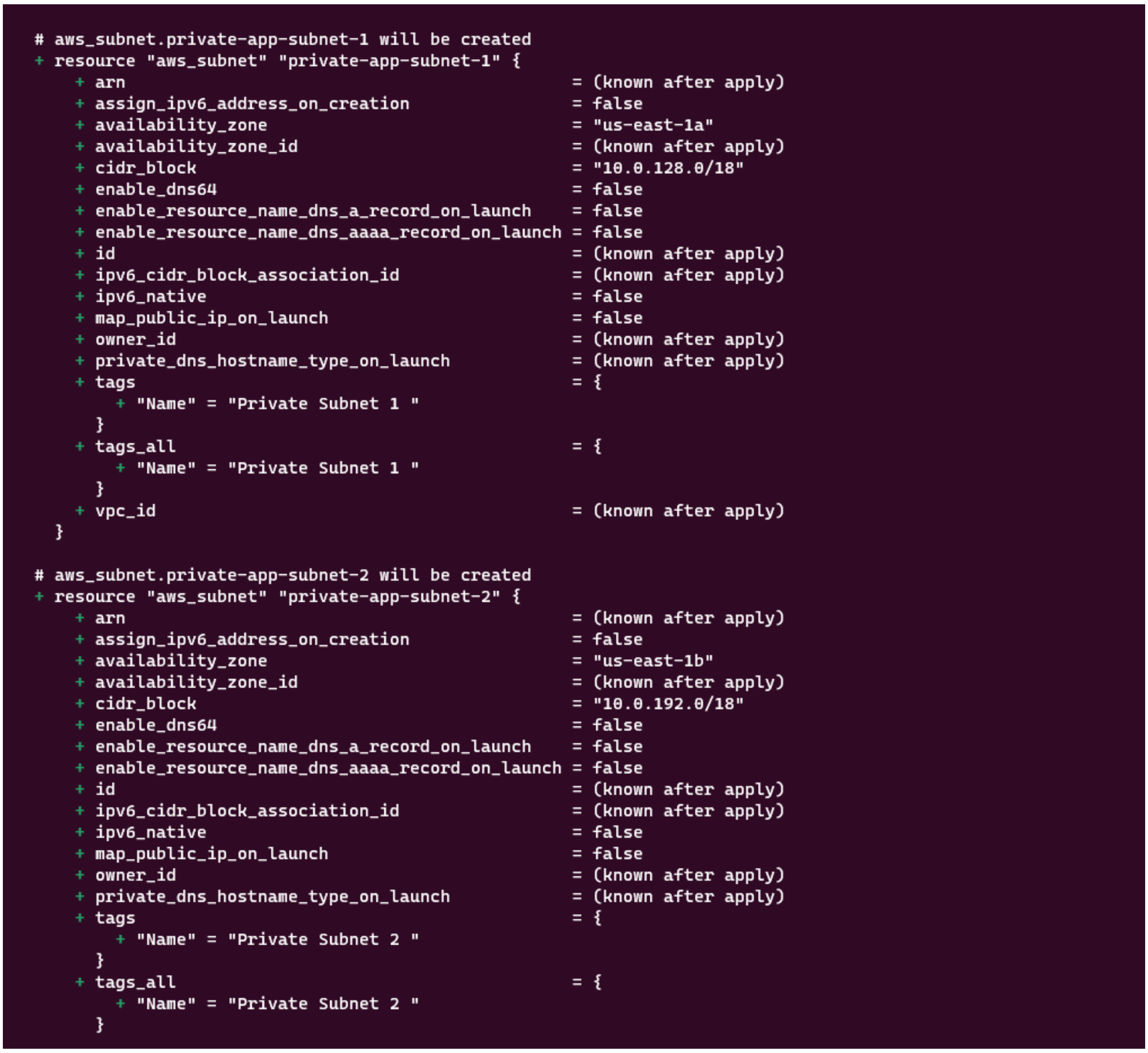
```
# aws_route_table.private-route-table will be created
+ resource "aws_route_table" "private-route-table" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + carrier_gateway_id         = ""
            + cidr_block                 = "0.0.0.0/0"
            + core_network_arn           = ""
            + destination_prefix_list_id = ""
            + egress_only_gateway_id     = ""
            + gateway_id                 = ""
            + ipv6_cidr_block            = ""
            + local_gateway_id           = ""
            + nat_gateway_id             = (known after apply)
            + network_interface_id       = ""
            + transit_gateway_id         = ""
            + vpc_endpoint_id            = ""
            + vpc_peering_connection_id  = ""
          },
      ]
    + tags             = {
        + "Name" = "Private Route Table"
      }
    + tags_all         = {
        + "Name" = "Private Route Table"
      }
    + vpc_id           = (known after apply)
  }

# aws_route_table.public-route-table will be created
```



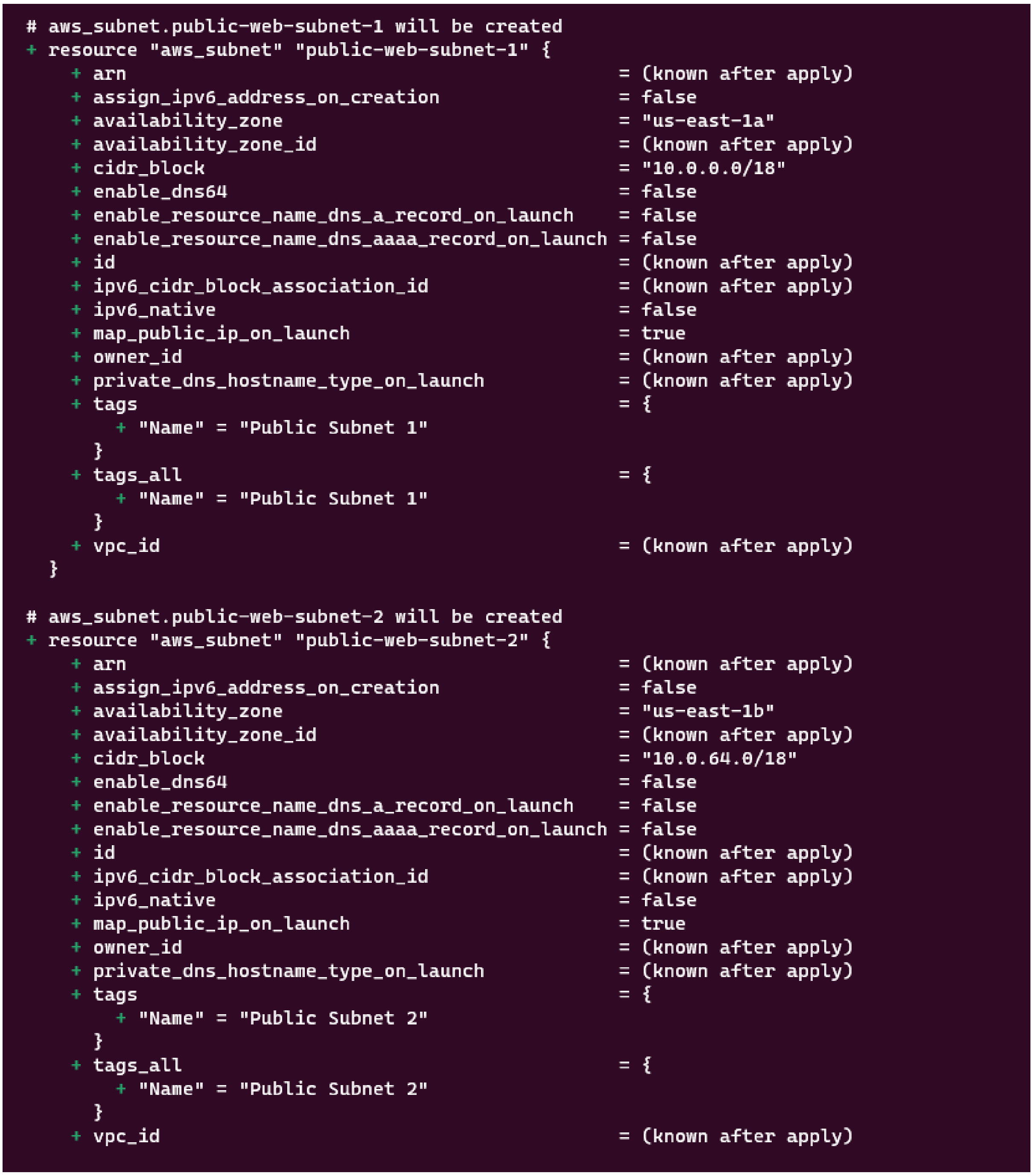
```
# aws_route_table.public-route-table will be created
+ resource "aws_route_table" "public-route-table" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + carrier_gateway_id         = ""
            + cidr_block                 = "0.0.0.0/0"
            + core_network_arn           = ""
            + destination_prefix_list_id = ""
            + egress_only_gateway_id     = ""
            + gateway_id                 = (known after apply)
            + ipv6_cidr_block            = ""
            + local_gateway_id           = ""
            + nat_gateway_id             = ""
            + network_interface_id       = ""
            + transit_gateway_id         = ""
            + vpc_endpoint_id            = ""
            + vpc_peering_connection_id  = ""
          },
      ]
    + tags             = {
        + "Name" = "Public Route Table"
      }
    + tags_all         = {
        + "Name" = "Public Route Table"
      }
    + vpc_id           = (known after apply)
  }
```



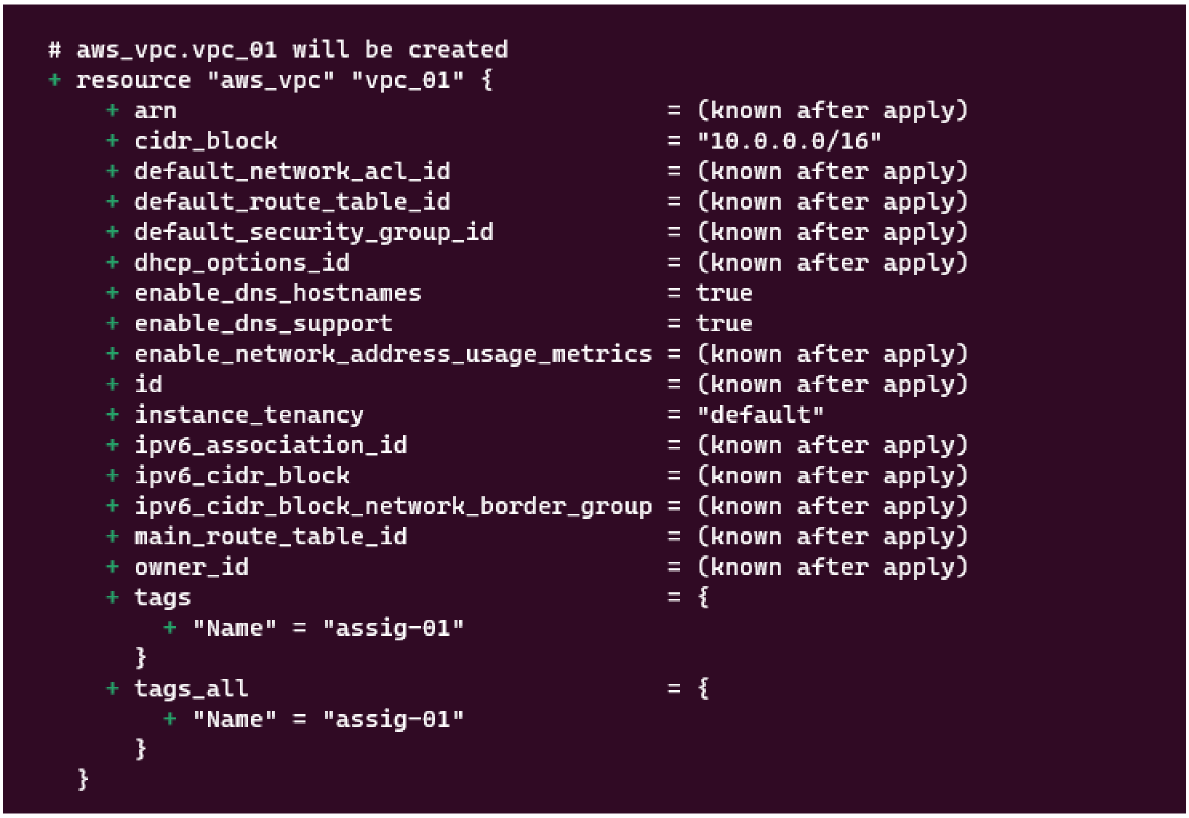
```
# aws_route_table_association.nat_route_1 will be created
+ resource "aws_route_table_association" "nat_route_1" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.nat_route_2 will be created
+ resource "aws_route_table_association" "nat_route_2" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.public-subnet-1-route-table-association will be created
+ resource "aws_route_table_association" "public-subnet-1-route-table-association" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.public-subnet-2-route-table-association will be created
+ resource "aws_route_table_association" "public-subnet-2-route-table-association" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }
```



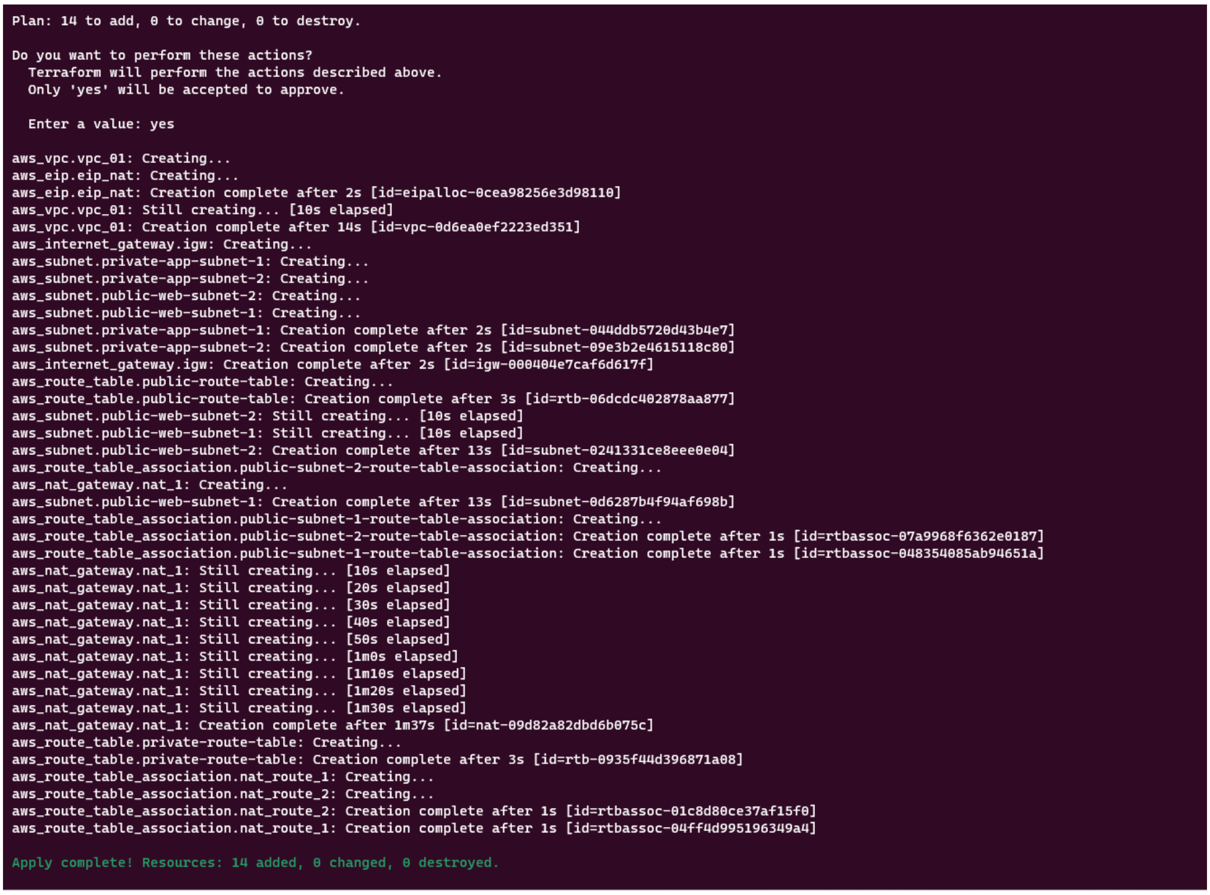
```
# aws_subnet.private-app-subnet-1 will be created
+ resource "aws_subnet" "private-app-subnet-1" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.128.0/18"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "Private Subnet 1 "
      }
    + tags_all                                       = {
        + "Name" = "Private Subnet 1 "
      }
    + vpc_id                                         = (known after apply)
  }

# aws_subnet.private-app-subnet-2 will be created
+ resource "aws_subnet" "private-app-subnet-2" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1b"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.192.0/18"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "Private Subnet 2 "
      }
    + tags_all                                       = {
        + "Name" = "Private Subnet 2 "
      }
```



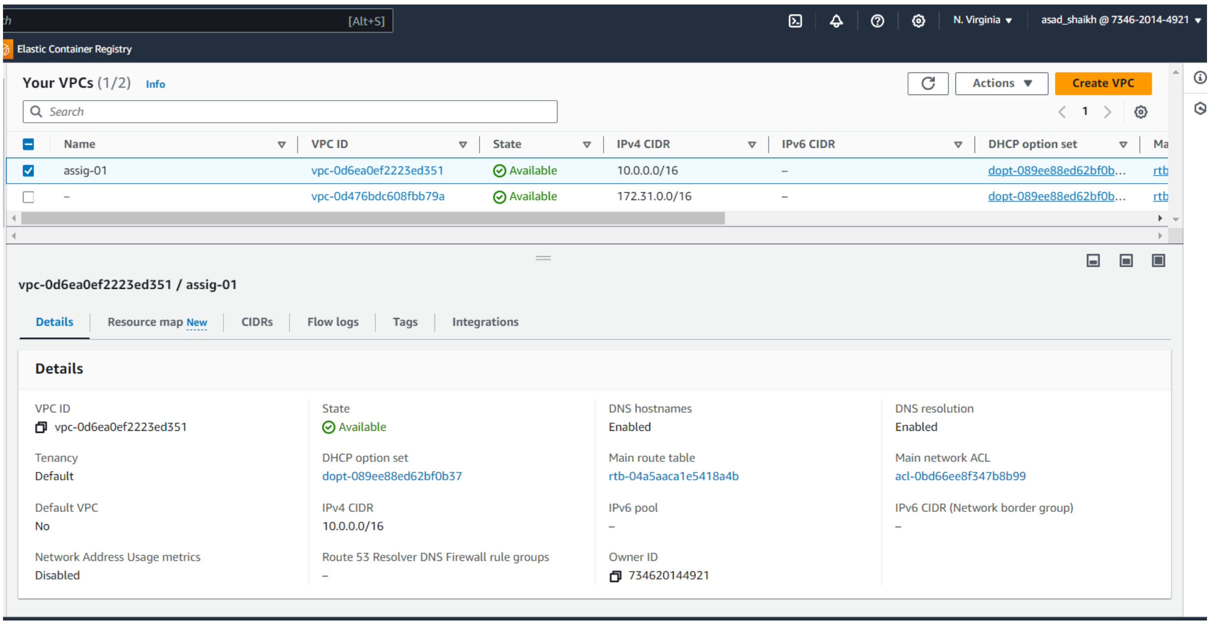
```
# aws_subnet.public-web-subnet-1 will be created
+ resource "aws_subnet" "public-web-subnet-1" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.0.0/18"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = true
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "Public Subnet 1"
      }
    + tags_all                                       = {
        + "Name" = "Public Subnet 1"
      }
    + vpc_id                                         = (known after apply)
  }

# aws_subnet.public-web-subnet-2 will be created
+ resource "aws_subnet" "public-web-subnet-2" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1b"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.64.0/18"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = true
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "Public Subnet 2"
      }
    + tags_all                                       = {
        + "Name" = "Public Subnet 2"
      }
    + vpc_id                                         = (known after apply)
```



```
  # aws_vpc.vpc_01 will be created
  + resource "aws_vpc" "vpc_01" {
      + 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                 = true
      + 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" = "assig-01"
        }
      + tags_all                             = {
          + "Name" = "assig-01"
        }
    }
```



```
Plan: 14 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_vpc.vpc_01: Creating...
aws_eip.eip_nat: Creating...
aws_eip.eip_nat: Creation complete after 2s [id=eipalloc-0cea98256e3d98110]
aws_vpc.vpc_01: Still creating... [10s elapsed]
aws_vpc.vpc_01: Creation complete after 14s [id=vpc-0d6ea0ef2223ed351]
aws_internet_gateway.igw: Creating...
aws_subnet.private-app-subnet-1: Creating...
aws_subnet.private-app-subnet-2: Creating...
aws_subnet.public-web-subnet-2: Creating...
aws_subnet.public-web-subnet-1: Creating...
aws_subnet.private-app-subnet-1: Creation complete after 2s [id=subnet-044ddb5720d43b4e7]
aws_subnet.private-app-subnet-2: Creation complete after 2s [id=subnet-09e3b2e4615118c80]
aws_internet_gateway.igw: Creation complete after 2s [id=igw-000404e7caf6d617f]
aws_route_table.public-route-table: Creating...
aws_route_table.public-route-table: Creation complete after 3s [id=rtb-06dcdc402878aa877]
aws_subnet.public-web-subnet-2: Still creating... [10s elapsed]
aws_subnet.public-web-subnet-1: Still creating... [10s elapsed]
aws_subnet.public-web-subnet-2: Creation complete after 13s [id=subnet-0241331ce8eee0e04]
aws_route_table_association.public-subnet-2-route-table-association: Creating...
aws_nat_gateway.nat_1: Creating...
aws_subnet.public-web-subnet-1: Creation complete after 13s [id=subnet-0d6287b4f94af698b]
aws_route_table_association.public-subnet-1-route-table-association: Creating...
aws_route_table_association.public-subnet-2-route-table-association: Creation complete after 1s [id=rtbassoc-07a9968f6362e0187]
aws_route_table_association.public-subnet-1-route-table-association: Creation complete after 1s [id=rtbassoc-048354085ab94651a]
aws_nat_gateway.nat_1: Still creating... [10s elapsed]
aws_nat_gateway.nat_1: Still creating... [20s elapsed]
aws_nat_gateway.nat_1: Still creating... [30s elapsed]
aws_nat_gateway.nat_1: Still creating... [40s elapsed]
aws_nat_gateway.nat_1: Still creating... [50s elapsed]
aws_nat_gateway.nat_1: Still creating... [1m0s elapsed]
aws_nat_gateway.nat_1: Still creating... [1m10s elapsed]
aws_nat_gateway.nat_1: Still creating... [1m20s elapsed]
aws_nat_gateway.nat_1: Still creating... [1m30s elapsed]
aws_nat_gateway.nat_1: Creation complete after 1m37s [id=nat-09d82a82dbd6b075c]
aws_route_table.private-route-table: Creating...
aws_route_table.private-route-table: Creation complete after 3s [id=rtb-0935f44d396871a08]
aws_route_table_association.nat_route_1: Creating...
aws_route_table_association.nat_route_2: Creating...
aws_route_table_association.nat_route_2: Creation complete after 1s [id=rtbassoc-01c8d80ce37af15f0]
aws_route_table_association.nat_route_1: Creation complete after 1s [id=rtbassoc-04ff4d995196349a4]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.
```



Your VPCs (1/2) Info

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Ma
☑	assig-01	vpc-0d6ea0ef2223ed351	✓ Available	10.0.0.0/16	–	dopt-089ee88ed62bf0b...	rtb
□	–	vpc-0d476bdc608fbb79a	✓ Available	172.31.0.0/16	–	dopt-089ee88ed62bf0b...	rtb

vpc-0d6ea0ef2223ed351 / assig-01

Details

VPC ID
vpc-0d6ea0ef2223ed351

State
✓ Available

DNS hostnames
Enabled

DNS resolution
Enabled

Tenancy
Default

DHCP option set
dopt-089ee88ed62bf0b37

Main route table
rtb-04a5aaca1e5418a4b

Main network ACL
acl-0bd66ee8f347b8b99

Default VPC
No

IPv4 CIDR
10.0.0.0/16

IPv6 pool
–

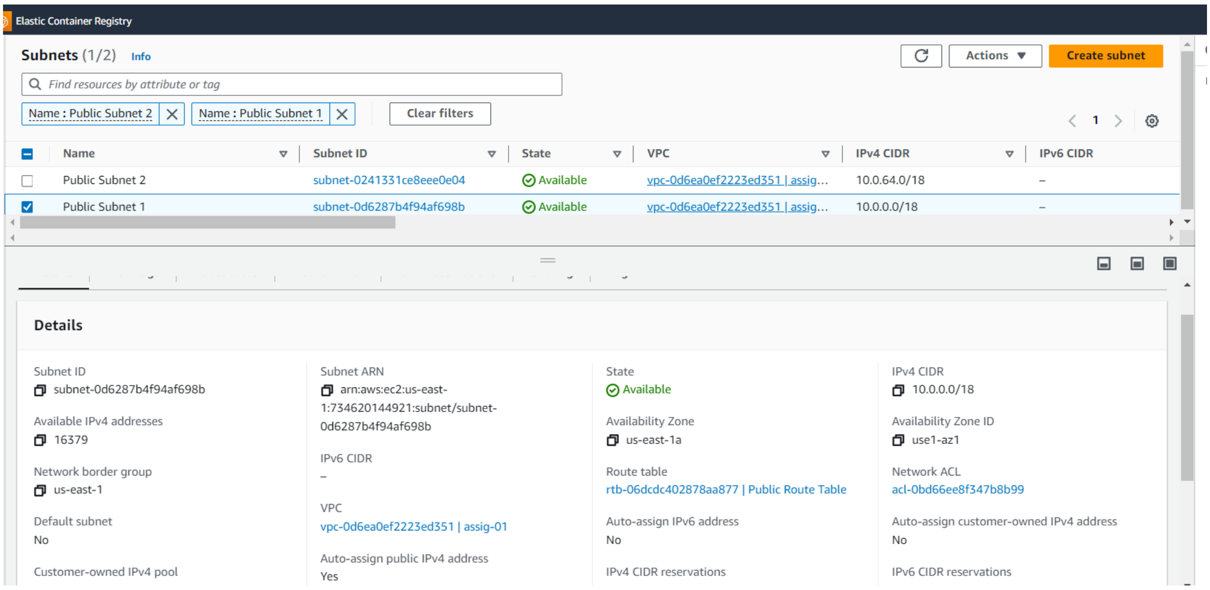
IPv6 CIDR (Network border group)
–

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
–

Owner ID
734620144921

Created 2 public and 2 Private subnets



Subnets (1/2) Info

	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
□	Public Subnet 2	subnet-0241331ce8eee0e04	✓ Available	vpc-0d6ea0ef2223ed351 | assig...	10.0.64.0/18	–
☑	Public Subnet 1	subnet-0d6287b4f94af698b	✓ Available	vpc-0d6ea0ef2223ed351 | assig...	10.0.0.0/18	–

Details

Subnet ID
subnet-0d6287b4f94af698b

Subnet ARN
arn:aws:ec2:us-east-1:734620144921:subnet/subnet-0d6287b4f94af698b

State
✓ Available

IPv4 CIDR
10.0.0.0/18

Available IPv4 addresses
16379

IPv6 CIDR
–

Availability Zone
us-east-1a

Availability Zone ID
use1-az1

Network border group
us-east-1

Route table
rtb-06dcdc402878aa877 | Public Route Table

Network ACL
acl-0bd66ee8f347b8b99

Default subnet
No

VPC
vpc-0d6ea0ef2223ed351 | assig-01

Auto-assign IPv6 address
No

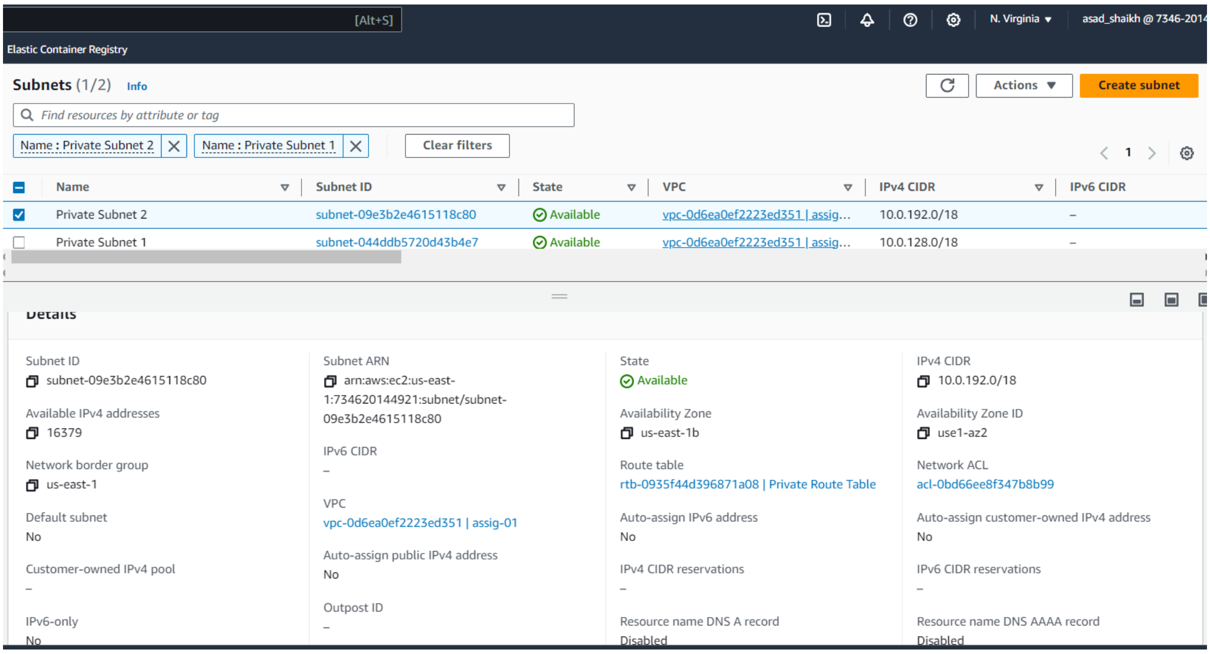
Auto-assign customer-owned IPv4 address
No

Customer-owned IPv4 pool

Auto-assign public IPv4 address
Yes

IPv4 CIDR reservations

IPv6 CIDR reservations



Subnets (1/2) Info

☑	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
☑	Private Subnet 2	subnet-09e3b2e4615118c80	Available	vpc-0d6ea0ef2223ed351 | assig…	10.0.192.0/18	–
□	Private Subnet 1	subnet-044ddb5720d43b4e7	Available	vpc-0d6ea0ef2223ed351 | assig…	10.0.128.0/18	–

Details

Subnet ID
subnet-09e3b2e4615118c80

Available IPv4 addresses
16379

Network border group
us-east-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

Subnet ARN
arn:aws:ec2:us-east-1:734620144921:subnet/subnet-09e3b2e4615118c80

IPv6 CIDR
–

VPC
vpc-0d6ea0ef2223ed351 | assig-01

Auto-assign public IPv4 address
No

Outpost ID
–

State
Available

Availability Zone
us-east-1b

Route table
rtb-0935f44d396871a08 | Private Route Table

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

IPv4 CIDR
10.0.192.0/18

Availability Zone ID
use1-az2

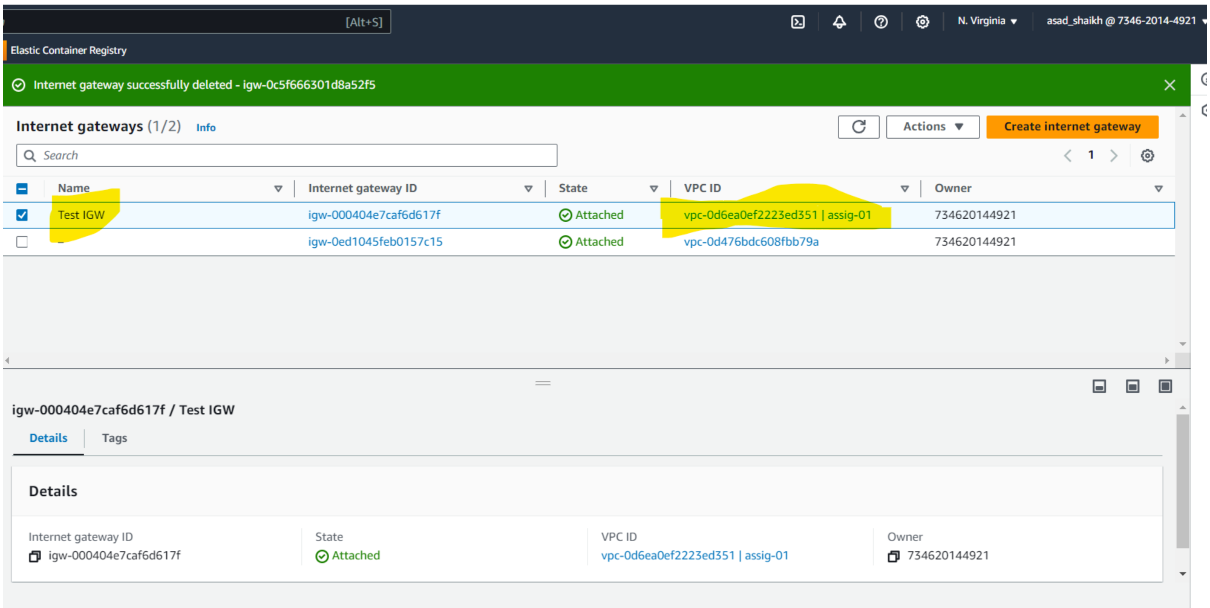
Network ACL
acl-0bd66ee8f347b8b99

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled

Created Internet Gateway



Internet gateway successfully deleted - igw-0c5f666301d8a52f5

Internet gateways (1/2) Info

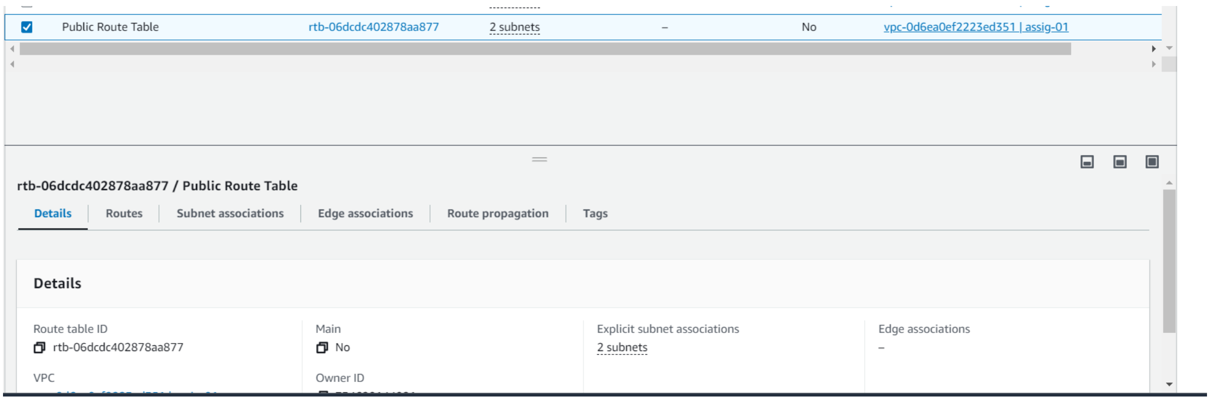
☑	Name	Internet gateway ID	State	VPC ID	Owner
☑	Test IGW	igw-000404e7caf6d617f	Attached	vpc-0d6ea0ef2223ed351 | assig-01	734620144921
□	–	igw-0ed1045feb0157c15	Attached	vpc-0d476bdc608fbb79a	734620144921

igw-000404e7caf6d617f / Test IGW

Details

Internet gateway ID	State	VPC ID	Owner
igw-000404e7caf6d617f	Attached	vpc-0d6ea0ef2223ed351 | assig-01	734620144921

Public and Private route table



☑	Public Route Table	rtb-06dcdc402878aa877	2 subnets	–	No	vpc-0d6ea0ef2223ed351 | assig-01

rtb-06dcdc402878aa877 / Public Route Table

Details

Route table ID: rtb-06dcdc402878aa877

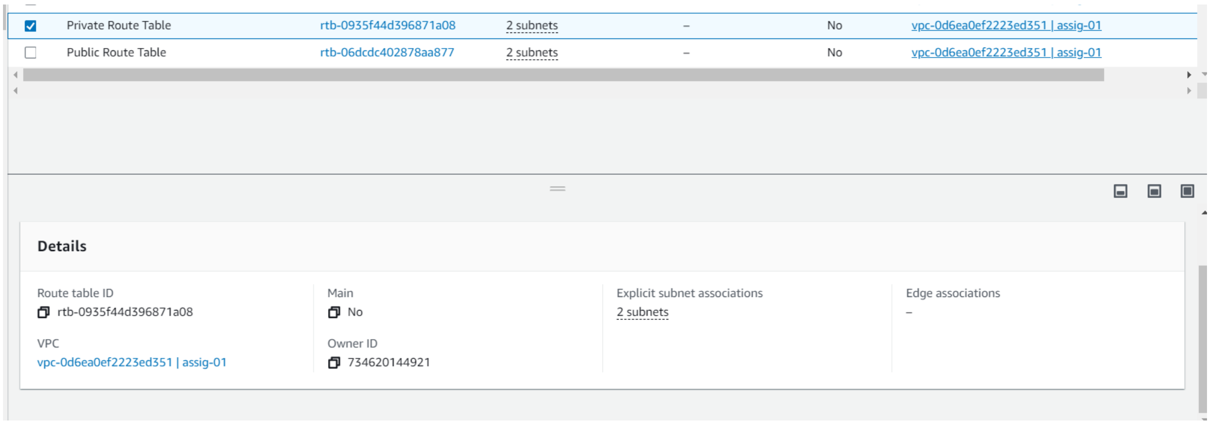
Main: No

Explicit subnet associations: 2 subnets

Edge associations: –

VPC

Owner ID



☑	Private Route Table	rtb-0935f44d396871a08	2 subnets	–	No	vpc-0d6ea0ef2223ed351 | assig-01
□	Public Route Table	rtb-06dcdc402878aa877	2 subnets	–	No	vpc-0d6ea0ef2223ed351 | assig-01

Details

Route table ID: rtb-0935f44d396871a08

Main: No

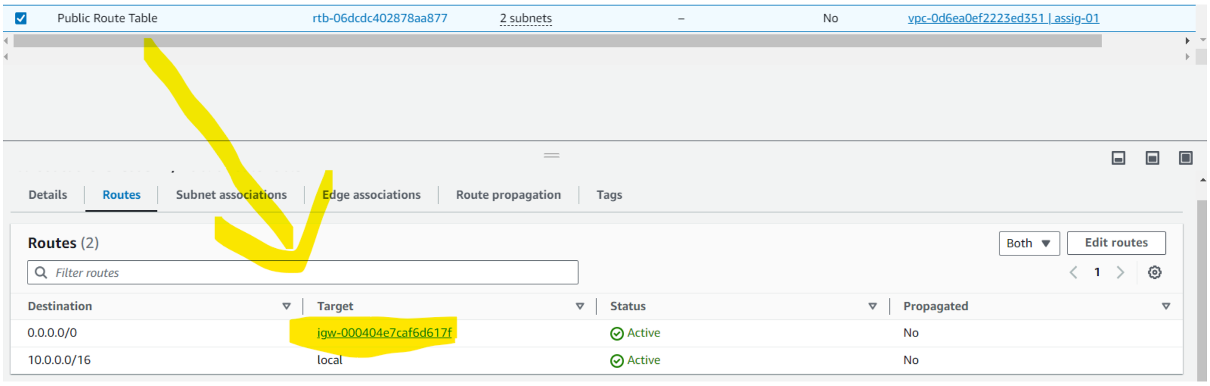
Explicit subnet associations: 2 subnets

Edge associations: –

VPC: vpc-0d6ea0ef2223ed351 | assig-01

Owner ID: 734620144921

Added Internet gateway in Public route table

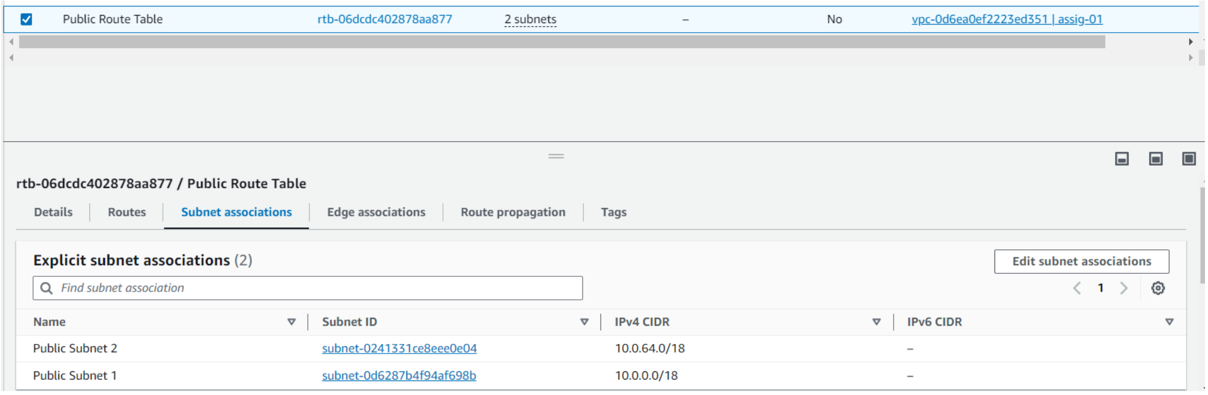


☑	Public Route Table	rtb-06dcdc402878aa877	2 subnets	–	No	vpc-0d6ea0ef2223ed351 | assig-01

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-000404e7caf6d617f	Active	No
10.0.0.0/16	local	Active	No

Added 2 Public Subnet in Public Route Table



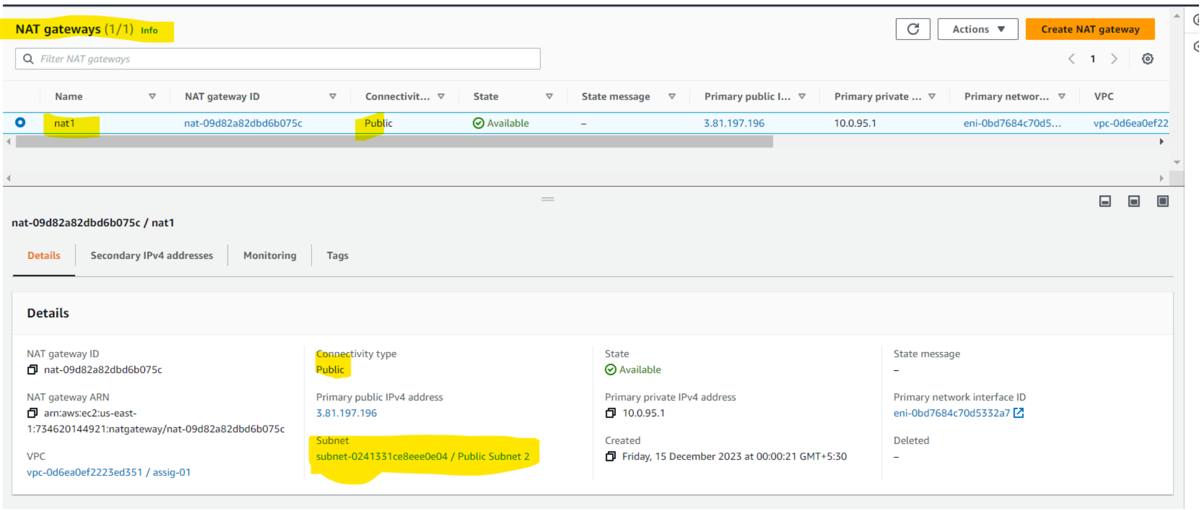
☑	Public Route Table	rtb-06dcdc402878aa877	2 subnets	–	No	vpc-0d6ea0ef2223ed351 | assig-01

rtb-06dcdc402878aa877 / Public Route Table

Explicit subnet associations (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Public Subnet 2	subnet-0241331ce8eee0e04	10.0.64.0/18	–
Public Subnet 1	subnet-0d6287b4f94af698b	10.0.0.0/18	–

Created NAT gateway in the public subnet



NAT gateways (1/1) Info

	Name	NAT gateway ID	Connectivit...	State	State message	Primary public I...	Primary private ...	Primary networ...	VPC
●	nat1	nat-09d82a82dbd6b075c	Public	✓ Available	–	3.81.197.196	10.0.95.1	eni-0bd7684c70d5...	vpc-0d6ea0ef22

nat-09d82a82dbd6b075c / nat1

Details

NAT gateway ID
nat-09d82a82dbd6b075c

NAT gateway ARN
arn:aws:ec2:us-east-1:734620144921:natgateway/nat-09d82a82dbd6b075c

VPC
vpc-0d6ea0ef2223ed351 / assig-01

Connectivity type
Public

Primary public IPv4 address
3.81.197.196

Subnet
subnet-0241331ce8eee0e04 / Public Subnet 2

State
✓ Available

Primary private IPv4 address
10.0.95.1

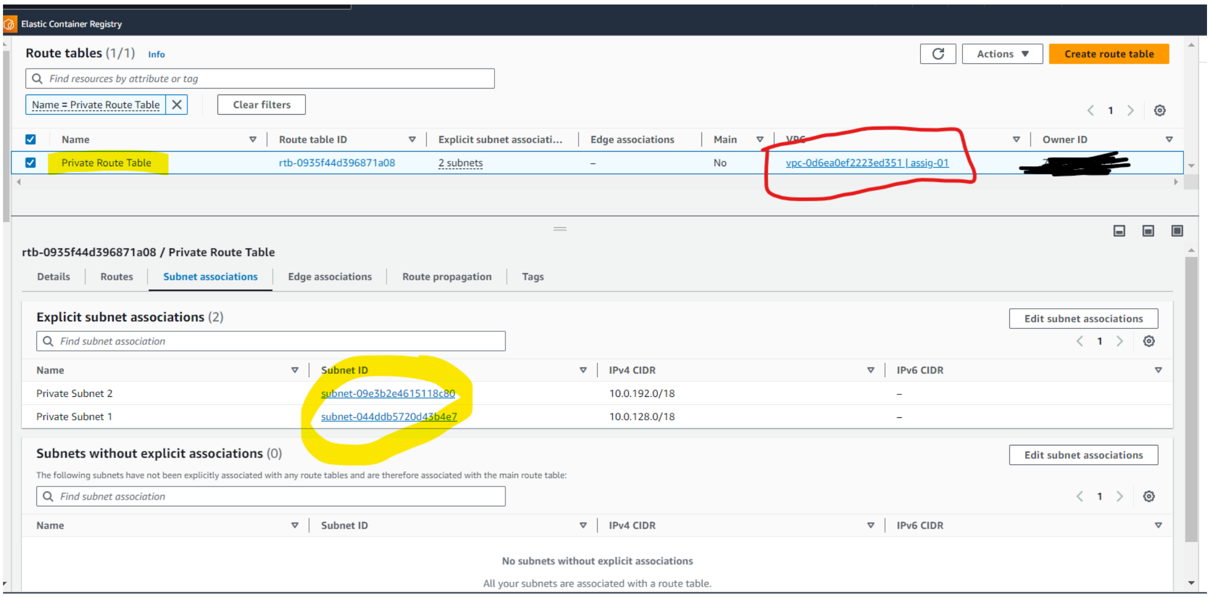
Created
Friday, 15 December 2023 at 00:00:21 GMT+5:30

State message
–

Primary network interface ID
eni-0bd7684c70d5332a7

Deleted
–

Add Private subnets to Private route table



Route tables (1/1) Info

Actions ▼
Create route table
Find resources by attribute or tag
Name = Private Route Table
Clear filters

☑	Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC	Owner ID
☑	Private Route Table	rtb-0935f44d396871a08	2 subnets	–	No	vpc-0d6ea0ef2223ed351 | assig-01	[redacted]

rtb-0935f44d396871a08 / Private Route Table

Explicit subnet associations (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Private Subnet 2	subnet-09e3b2e4615118c80	10.0.192.0/18	–
Private Subnet 1	subnet-044ddb5720d43b4e7	10.0.128.0/18	–

Subnets without explicit associations (0)

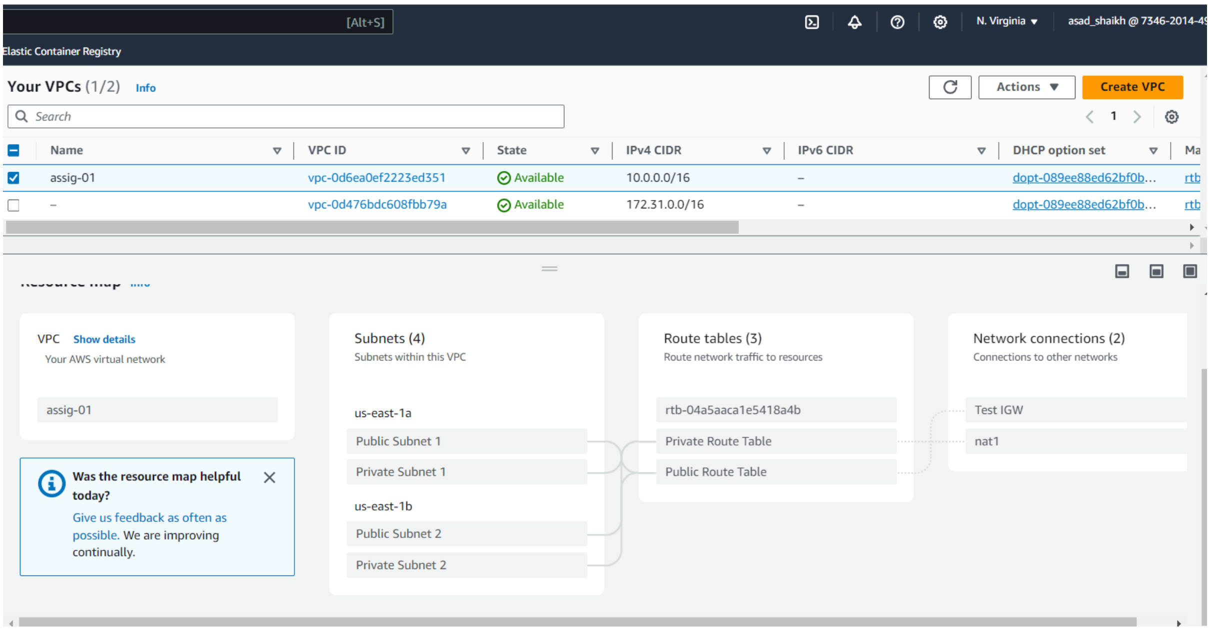
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR

No subnets without explicit associations

All your subnets are associated with a route table.

Below Map indicates the VPC Subnet Flow



Your VPCs (1/2) Info

☐	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Ma
☑	assig-01	vpc-0d6ea0ef2223ed351	✓ Available	10.0.0.0/16	–	dopt-089ee88ed62bf0b…	rtb
☐	–	vpc-0d476bdc608fbb79a	✓ Available	172.31.0.0/16	–	dopt-089ee88ed62bf0b…	rtb

Resource map Info

VPC Show details
Your AWS virtual network
assig-01

Was the resource map helpful today?
Give us feedback as often as possible. We are improving continually.

Subnets (4)
Subnets within this VPC
us-east-1a
Public Subnet 1
Private Subnet 1
us-east-1b
Public Subnet 2
Private Subnet 2

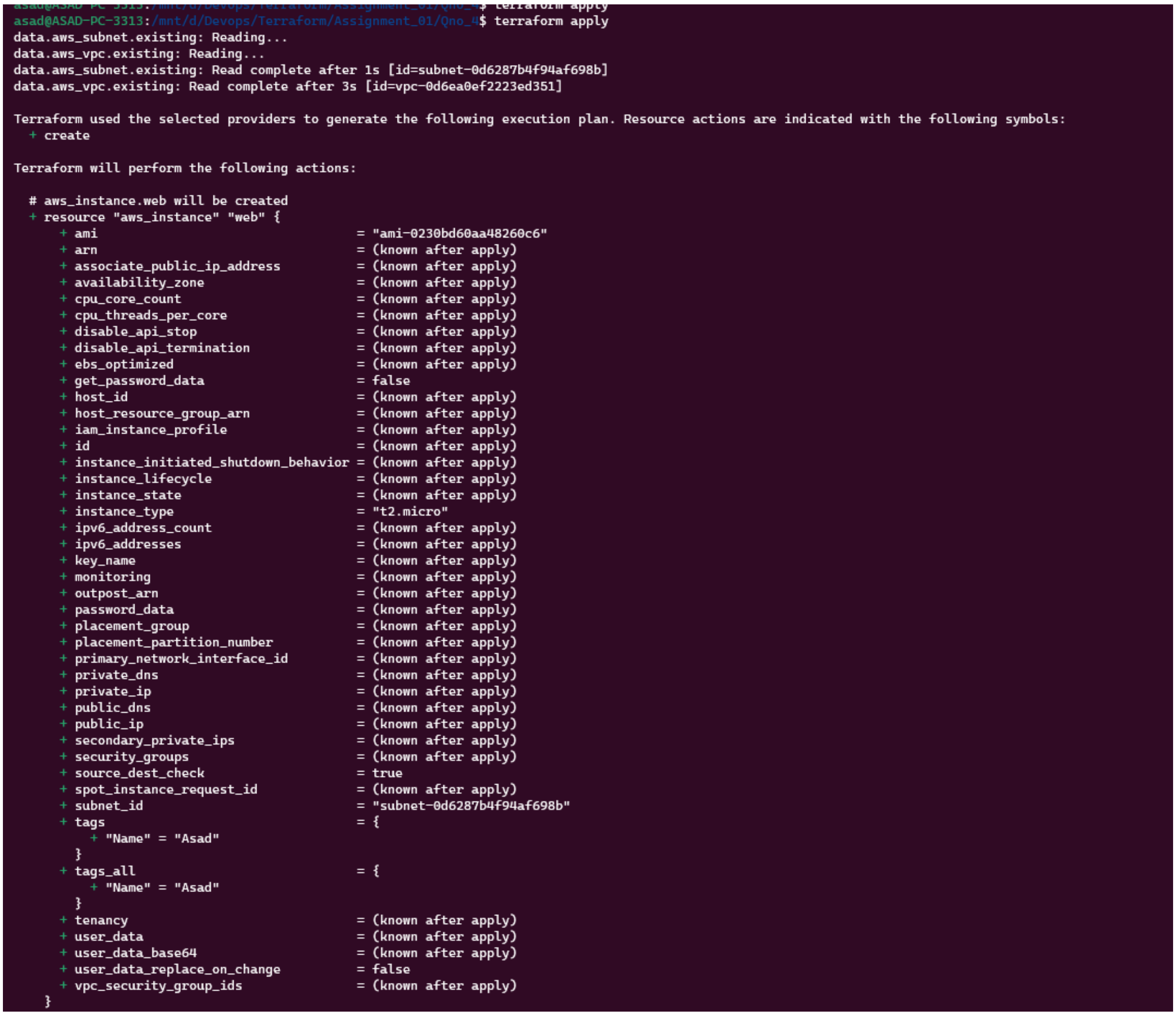
Route tables (3)
Route network traffic to resources
rtb-04a5aaca1e5418a4b
Private Route Table
Public Route Table

Network connections (2)
Connections to other networks
Test IGW
nat1

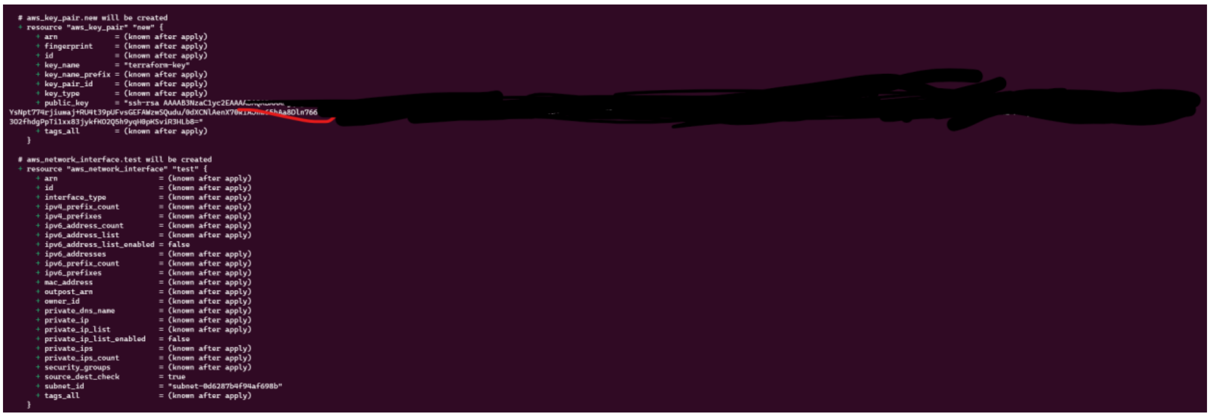
Qno 4

Create EC2 instance one of the public Subnets of VPC that you have created ● Validate your Connection using SSH

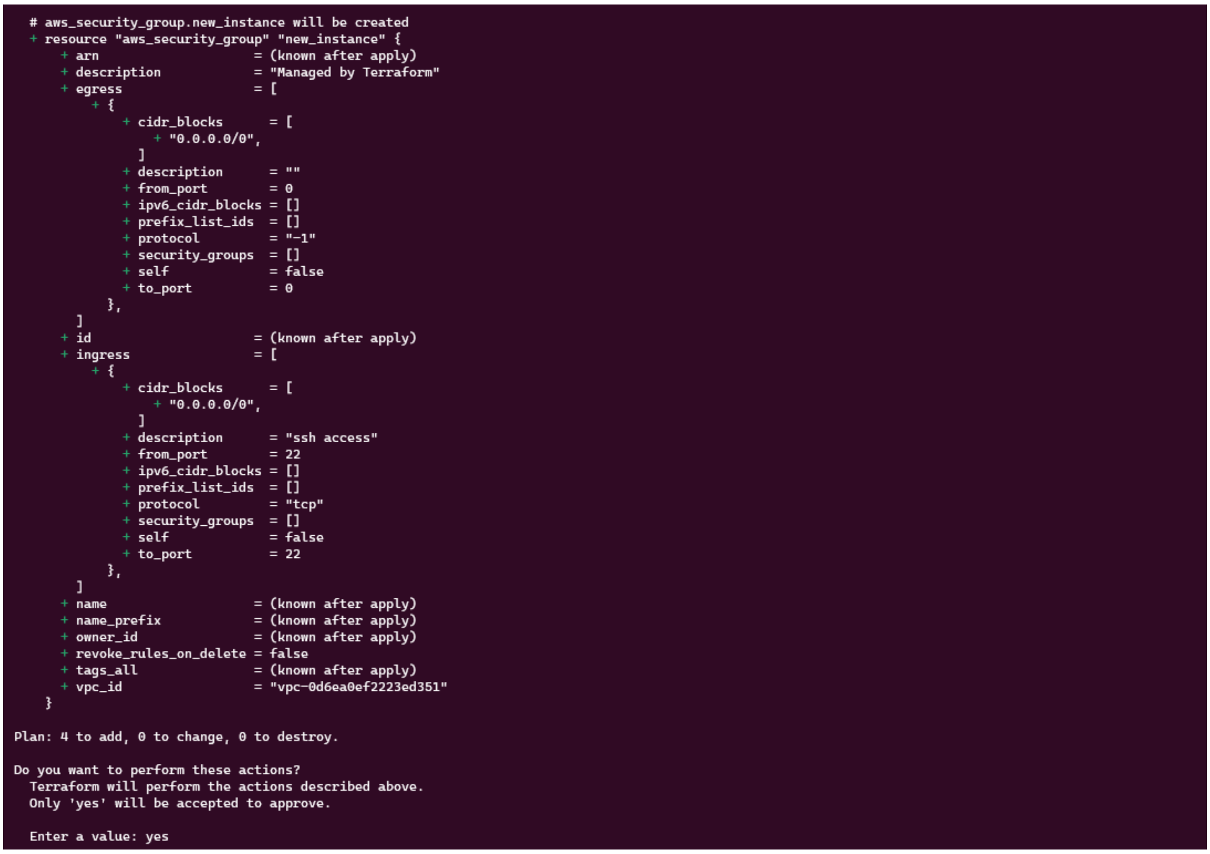
- ➔ Created ec2 instance in previously created vpc and used existing public subnet in it Via data block



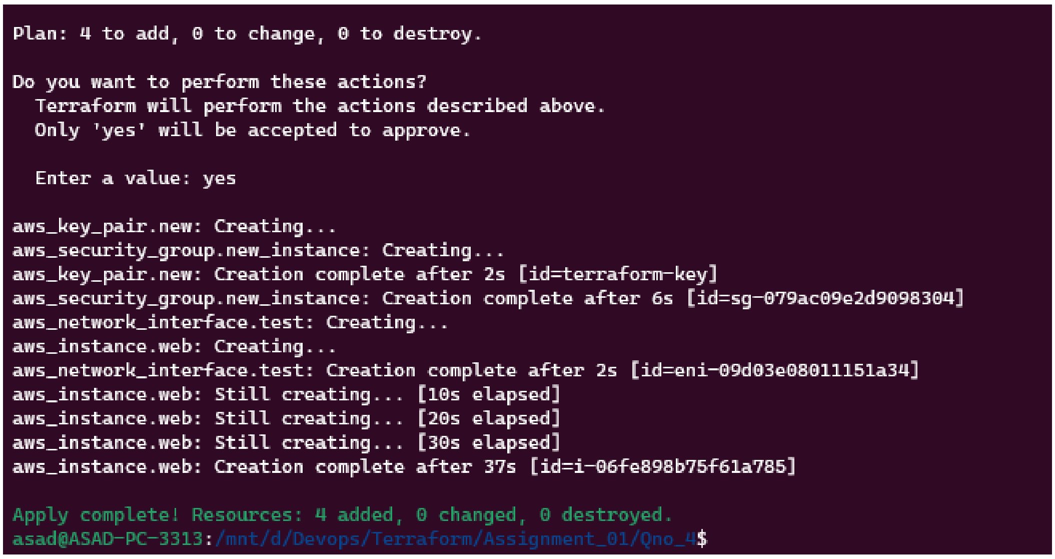
```
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_4$ terraform apply
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_4$ terraform apply
data.aws_subnet.existing: Reading...
data.aws_vpc.existing: Reading...
data.aws_subnet.existing: Read complete after 1s [id=subnet-0d6287b4f94af698b]
data.aws_vpc.existing: Read complete after 3s [id=vpc-0d6ea0ef2223ed351]

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.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-0230bd60aa48260c6"
      + 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_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                            = "subnet-0d6287b4f94af698b"
      + tags                                 = {
          + "Name" = "Asad"
        }
      + tags_all                             = {
          + "Name" = "Asad"
        }
      + 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_key_pair.new will be created
  + resource "aws_key_pair" "new" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "terraform-key"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-rsa AAAAB3NzaC1yc2EAAA[redacted]
YsNpt774rjiumaj+RU4t39pUFvsGEFAWzwSQudu/0dXCNlAenX70[redacted]
3O2fhdgPpTi1xx83jykfKO2Q5h9yqH8pKSviR3HLb8="
      + tags_all        = (known after apply)
    }

  # aws_network_interface.test will be created
  + resource "aws_network_interface" "test" {
      + arn                       = (known after apply)
      + id                        = (known after apply)
      + interface_type            = (known after apply)
      + ipv4_prefix_count         = (known after apply)
      + ipv4_prefixes             = (known after apply)
      + ipv6_address_count        = (known after apply)
      + ipv6_address_list         = (known after apply)
      + ipv6_address_list_enabled = false
      + ipv6_addresses            = (known after apply)
      + ipv6_prefix_count         = (known after apply)
      + ipv6_prefixes             = (known after apply)
      + mac_address               = (known after apply)
      + outpost_arn               = (known after apply)
      + owner_id                  = (known after apply)
      + private_dns_name          = (known after apply)
      + private_ip                = (known after apply)
      + private_ip_list           = (known after apply)
      + private_ip_list_enabled   = false
      + private_ips               = (known after apply)
      + private_ips_count         = (known after apply)
      + security_groups           = (known after apply)
      + source_dest_check         = true
      + subnet_id                 = "subnet-0d6287b4f94af698b"
      + tags_all                  = (known after apply)
    }
```

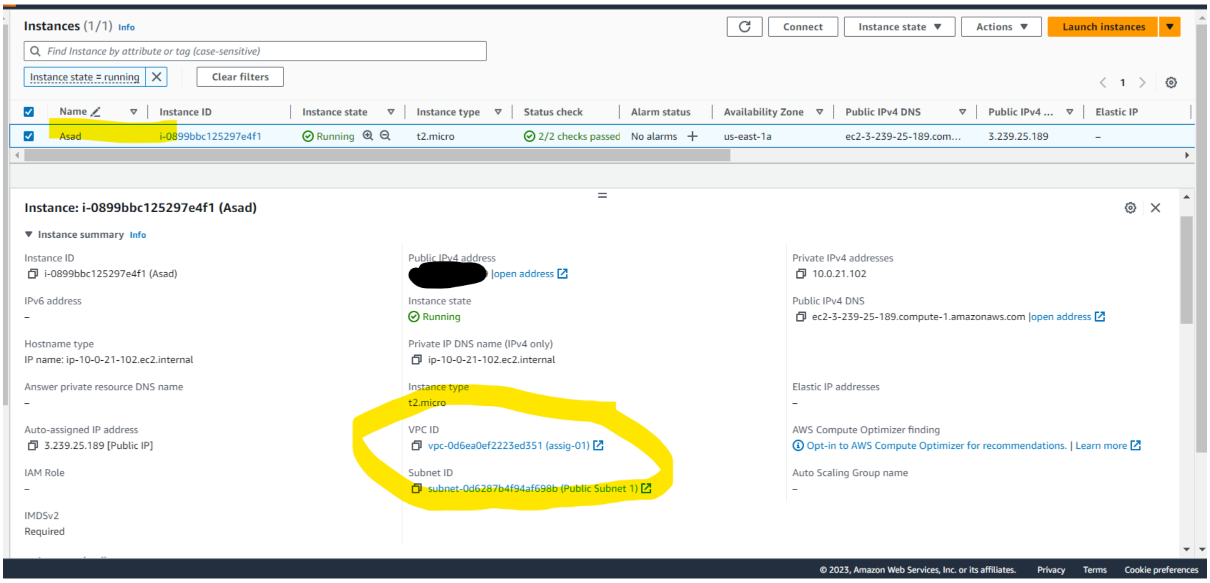


```
  # aws_security_group.new_instance will be created
  + resource "aws_security_group" "new_instance" {
      + arn                    = (known after apply)
      + description            = "Managed by Terraform"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = "ssh access"
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
            },
        ]
      + name                   = (known after apply)
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = "vpc-0d6ea0ef2223ed351"
    }

Plan: 4 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
```



```
Plan: 4 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_key_pair.new: Creating...
aws_security_group.new_instance: Creating...
aws_key_pair.new: Creation complete after 2s [id=terraform-key]
aws_security_group.new_instance: Creation complete after 6s [id=sg-079ac09e2d9098304]
aws_network_interface.test: Creating...
aws_instance.web: Creating...
aws_network_interface.test: Creation complete after 2s [id=eni-09d03e08011151a34]
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [20s elapsed]
aws_instance.web: Still creating... [30s elapsed]
aws_instance.web: Creation complete after 37s [id=i-06fe898b75f61a785]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
asad@ASAD-PC-3313:/mnt/d/Devops/Terraform/Assignment_01/Qno_4$
```

We can see that in console



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	Asad	i-0899bbc125297e4f1	Running	t2.micro	2/2 checks passed	No alarms +	us-east-1a	ec2-3-239-25-189.com...	3.239.25.189	–

Instance: i-0899bbc125297e4f1 (Asad)

▼ Instance summary Info

Instance ID
i-0899bbc125297e4f1 (Asad)

IPv6 address
–

Hostname type
IP name: ip-10-0-21-102.ec2.internal

Answer private resource DNS name
–

Auto-assigned IP address
3.239.25.189 [Public IP]

IAM Role
–

IMDSv2
Required

Public IPv4 address
[redacted] |open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-21-102.ec2.internal

Instance type
t2.micro

VPC ID
vpc-0d6ea0ef2223ed351 (assig-01)

Subnet ID
subnet-0d6287b4f94af698b (Public Subnet 1)

Private IPv4 addresses
10.0.21.102

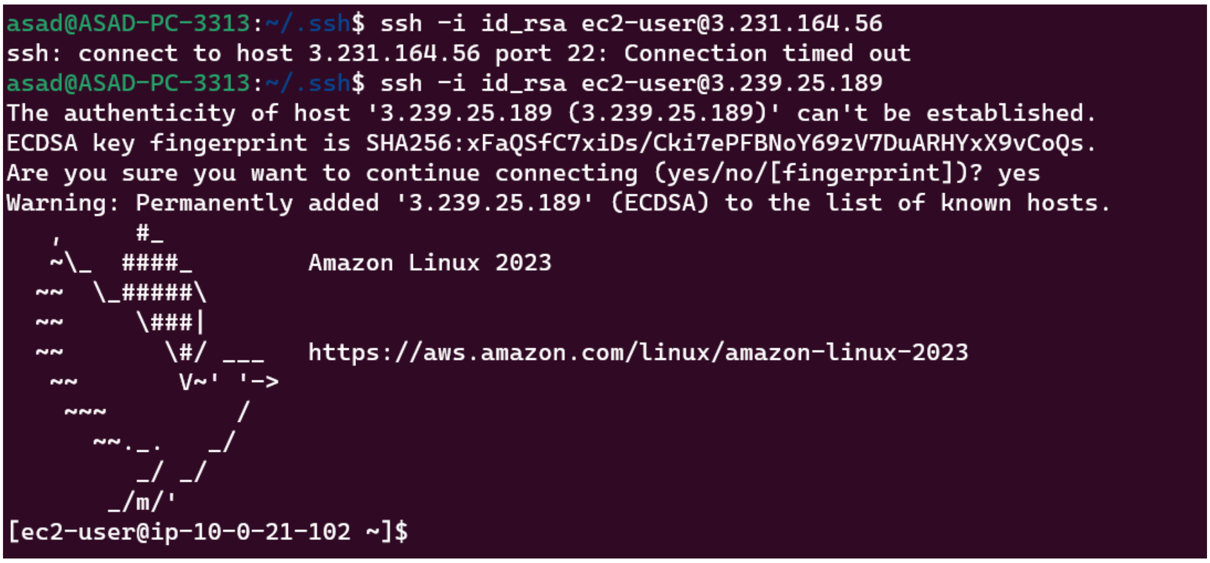
Public IPv4 DNS
ec2-3-239-25-189.compute-1.amazonaws.com |open address

Elastic IP addresses
–

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

Auto Scaling Group name
–

We can ssh into it as well



```
asad@ASAD-PC-3313:~/.ssh$ ssh -i id_rsa ec2-user@3.231.164.56
ssh: connect to host 3.231.164.56 port 22: Connection timed out
asad@ASAD-PC-3313:~/.ssh$ ssh -i id_rsa ec2-user@3.239.25.189
The authenticity of host '3.239.25.189 (3.239.25.189)' can't be established.
ECDSA key fingerprint is SHA256:xFaQSfC7xiDs/Cki7ePFBNoY69zV7DuARHYxX9vCoQs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.239.25.189' (ECDSA) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-21-102 ~]$
```

THANK YOU