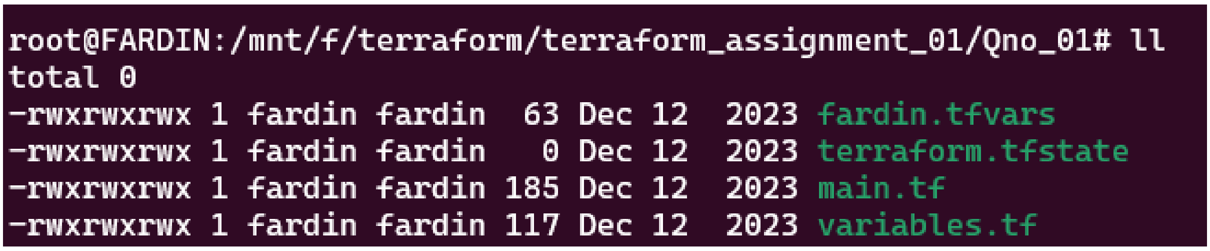
ASSIGNMENT NO. 01

01.

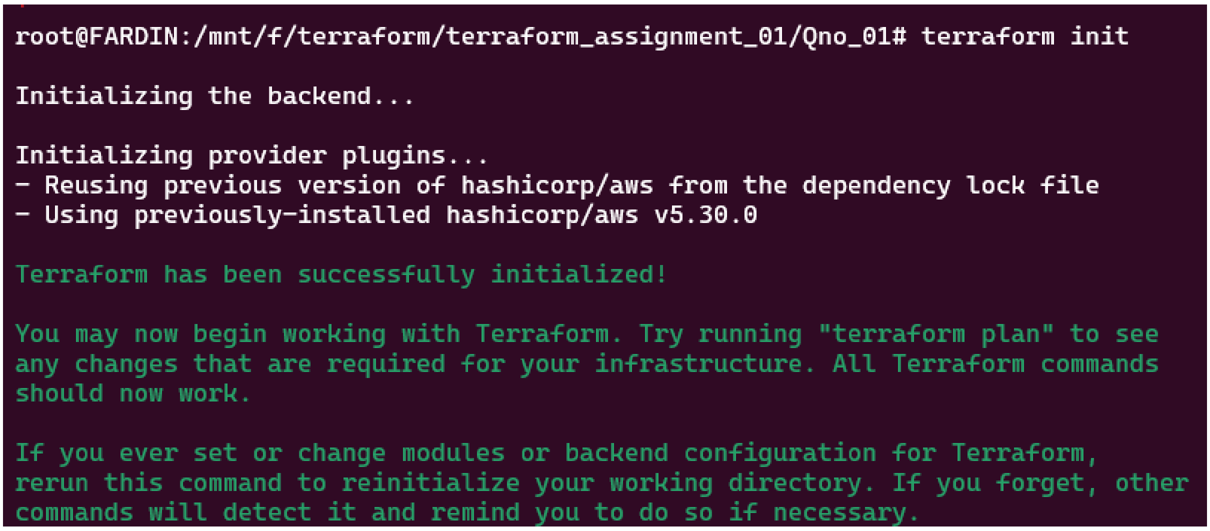
- Create one IAM user and one IAM Group using Terraform.
- Make sure you will use variables for names of IAM users and Group.
- Note: - Below files are required.
  - main.tf
  - variables.tf
  - your_name_custom.tfvars

ANS:-



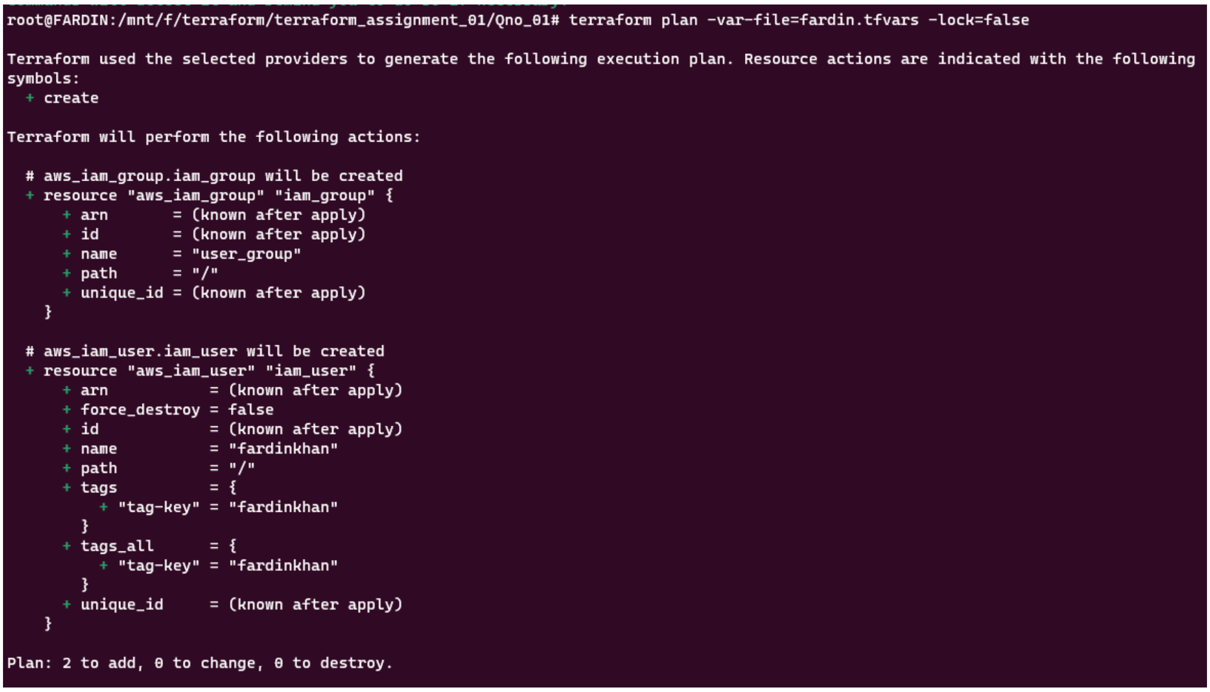
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_01# ll
total 0
-rwxrwxrwx 1 fardin fardin  63 Dec 12  2023 fardin.tfvars
-rwxrwxrwx 1 fardin fardin   0 Dec 12  2023 terraform.tfstate
-rwxrwxrwx 1 fardin fardin 185 Dec 12  2023 main.tf
-rwxrwxrwx 1 fardin fardin 117 Dec 12  2023 variables.tf
```

terraform init



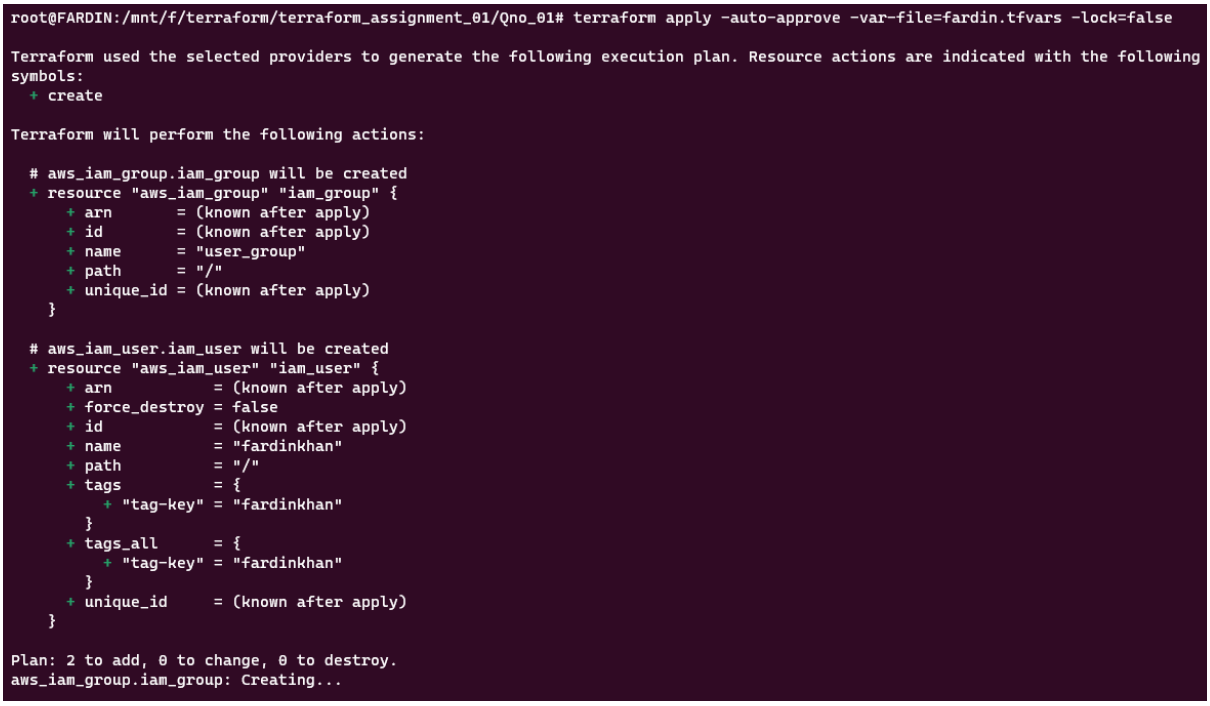
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_01# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.30.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

terraform plan

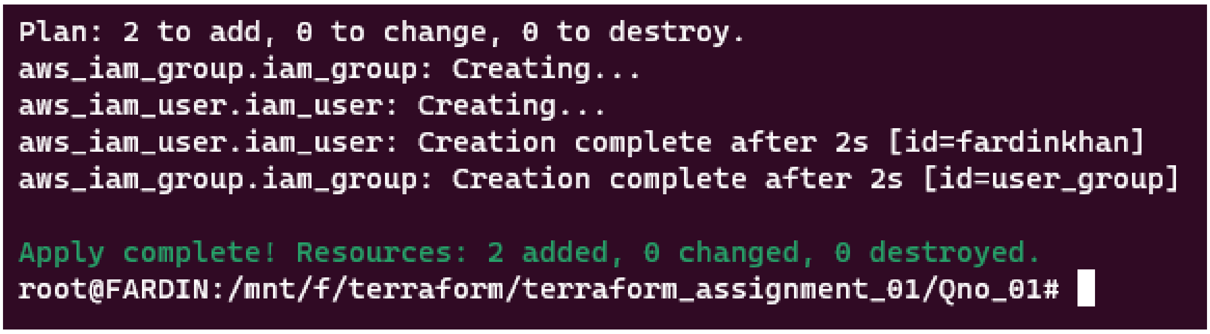


```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_01# terraform plan -var-file=fardin.tfvars -lock=false

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.iam_group will be created
  + resource "aws_iam_group" "iam_group" {
      + arn       = (known after apply)
      + id        = (known after apply)
      + name      = "user_group"
      + path      = "/"
      + unique_id = (known after apply)
    }

  # aws_iam_user.iam_user will be created
  + resource "aws_iam_user" "iam_user" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "fardinkhan"
      + path          = "/"
      + tags          = {
          + "tag-key" = "fardinkhan"
        }
      + tags_all      = {
          + "tag-key" = "fardinkhan"
        }
      + unique_id     = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
```

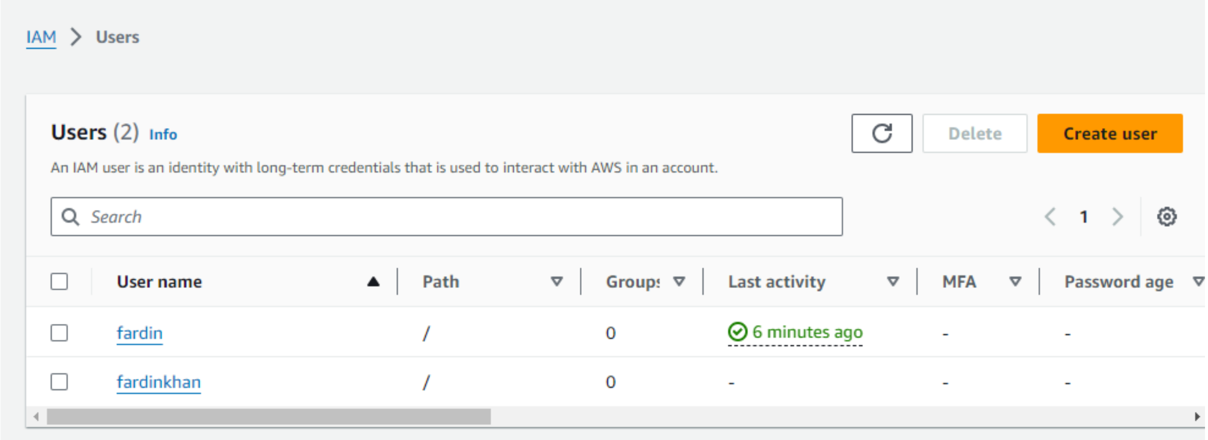
terraform apply



```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_01# terraform apply -auto-approve -var-file=fardin.tfvars -lock=false

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.iam_group will be created
  + resource "aws_iam_group" "iam_group" {
      + arn       = (known after apply)
      + id        = (known after apply)
      + name      = "user_group"
      + path      = "/"
      + unique_id = (known after apply)
    }

  # aws_iam_user.iam_user will be created
  + resource "aws_iam_user" "iam_user" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "fardinkhan"
      + path          = "/"
      + tags          = {
          + "tag-key" = "fardinkhan"
        }
      + tags_all      = {
          + "tag-key" = "fardinkhan"
        }
      + unique_id     = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_iam_group.iam_group: Creating...
```



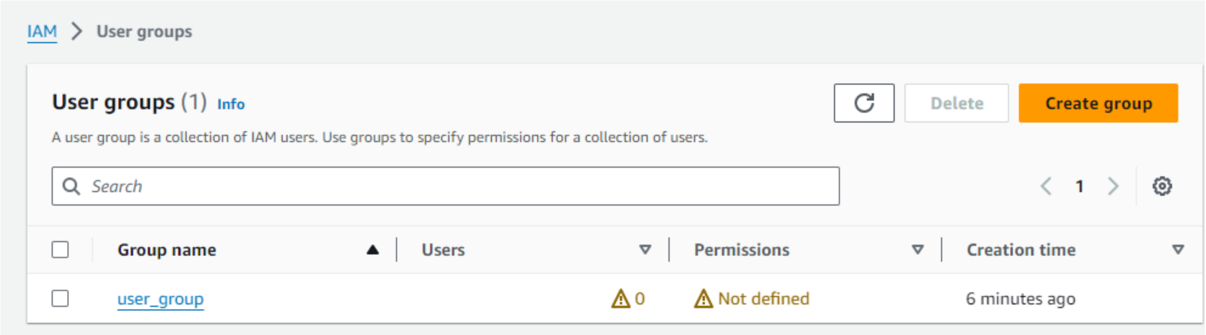
```
Plan: 2 to add, 0 to change, 0 to destroy.
aws_iam_group.iam_group: Creating...
aws_iam_user.iam_user: Creating...
aws_iam_user.iam_user: Creation complete after 2s [id=fardinkhan]
aws_iam_group.iam_group: Creation complete after 2s [id=user_group]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_01#
```



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
□	fardin	/	0	✓ 6 minutes ago	-	-
□	fardinkhan	/	0	-	-	-



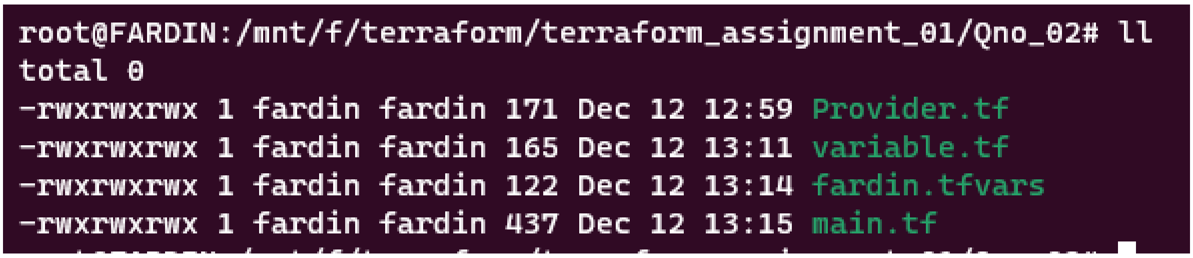
User groups (1) Info

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
□	user_group	⚠ 0	⚠ Not defined	6 minutes ago

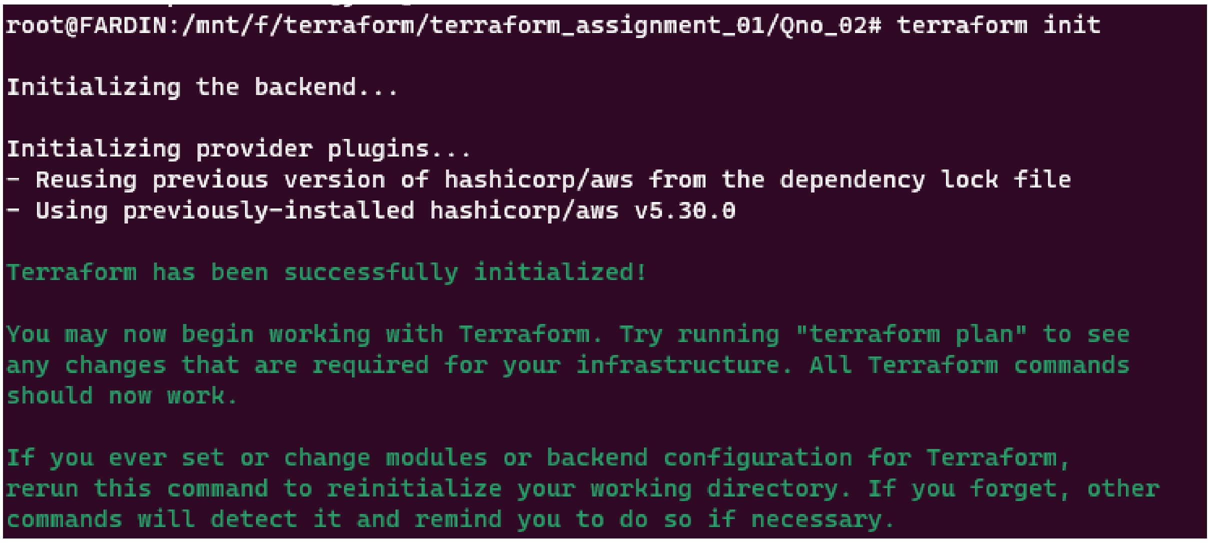
02.

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



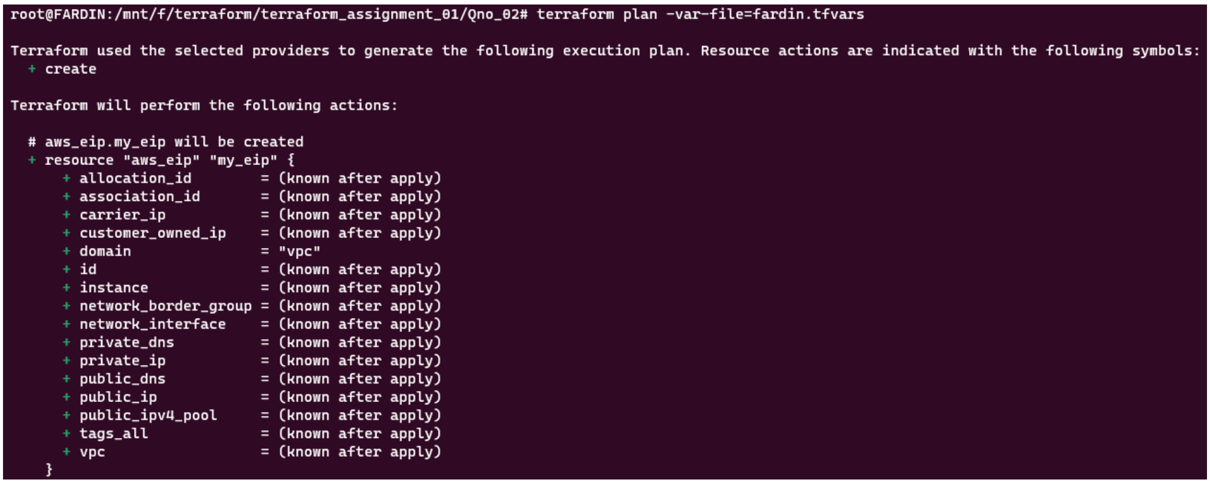
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_02# ll
total 0
-rwxrwxrwx 1 fardin fardin 171 Dec 12 12:59 Provider.tf
-rwxrwxrwx 1 fardin fardin 165 Dec 12 13:11 variable.tf
-rwxrwxrwx 1 fardin fardin 122 Dec 12 13:14 fardin.tfvars
-rwxrwxrwx 1 fardin fardin 437 Dec 12 13:15 main.tf
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_02#
```

terraform init

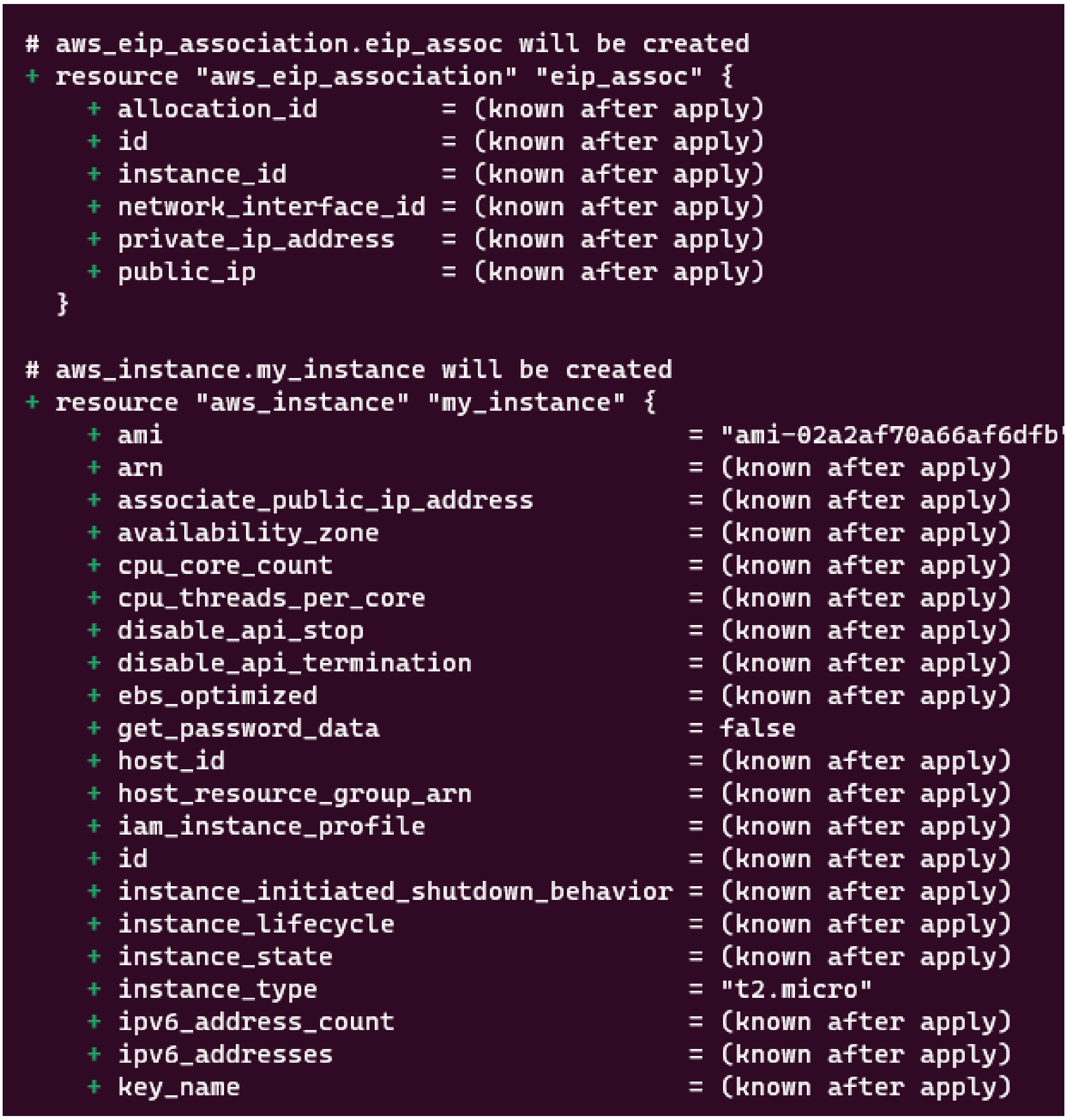


```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_02# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.30.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

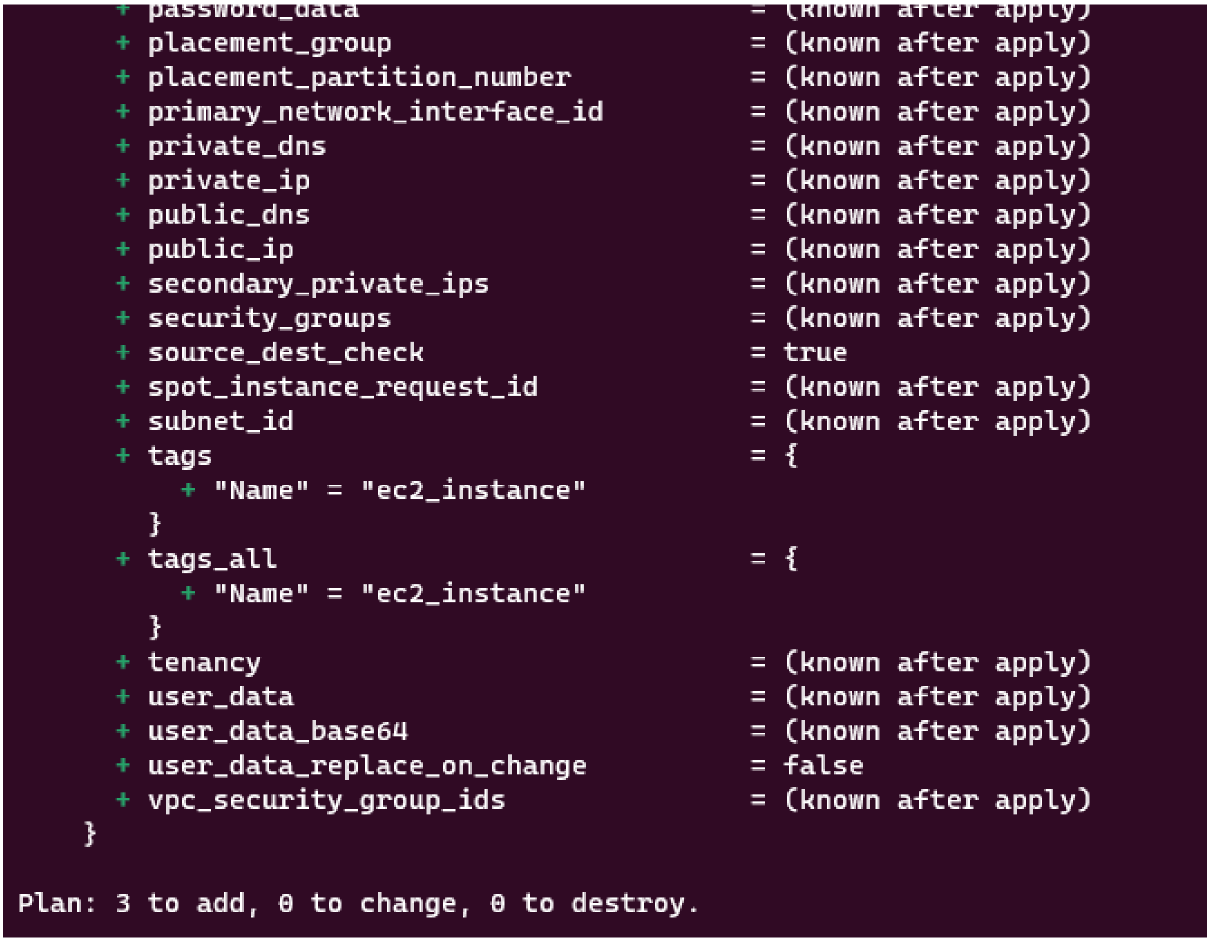
terraform plan



```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_02# terraform plan -var-file=fardin.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_eip.my_eip will be created
  + resource "aws_eip" "my_eip" {
      + 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)
    }
```

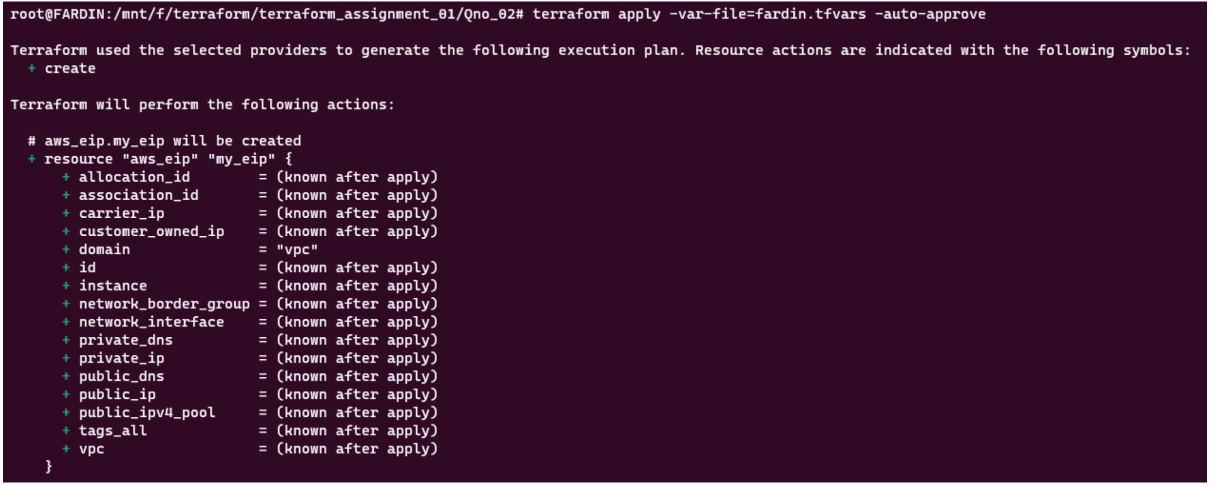


```
# aws_eip_association.eip_assoc will be created
+ resource "aws_eip_association" "eip_assoc" {
    + allocation_id        = (known after apply)
    + id                   = (known after apply)
    + instance_id          = (known after apply)
    + network_interface_id = (known after apply)
    + private_ip_address   = (known after apply)
    + public_ip            = (known after apply)
  }

# aws_instance.my_instance will be created
+ resource "aws_instance" "my_instance" {
    + ami                                  = "ami-02a2af70a66af6dfb"
    + 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)
```

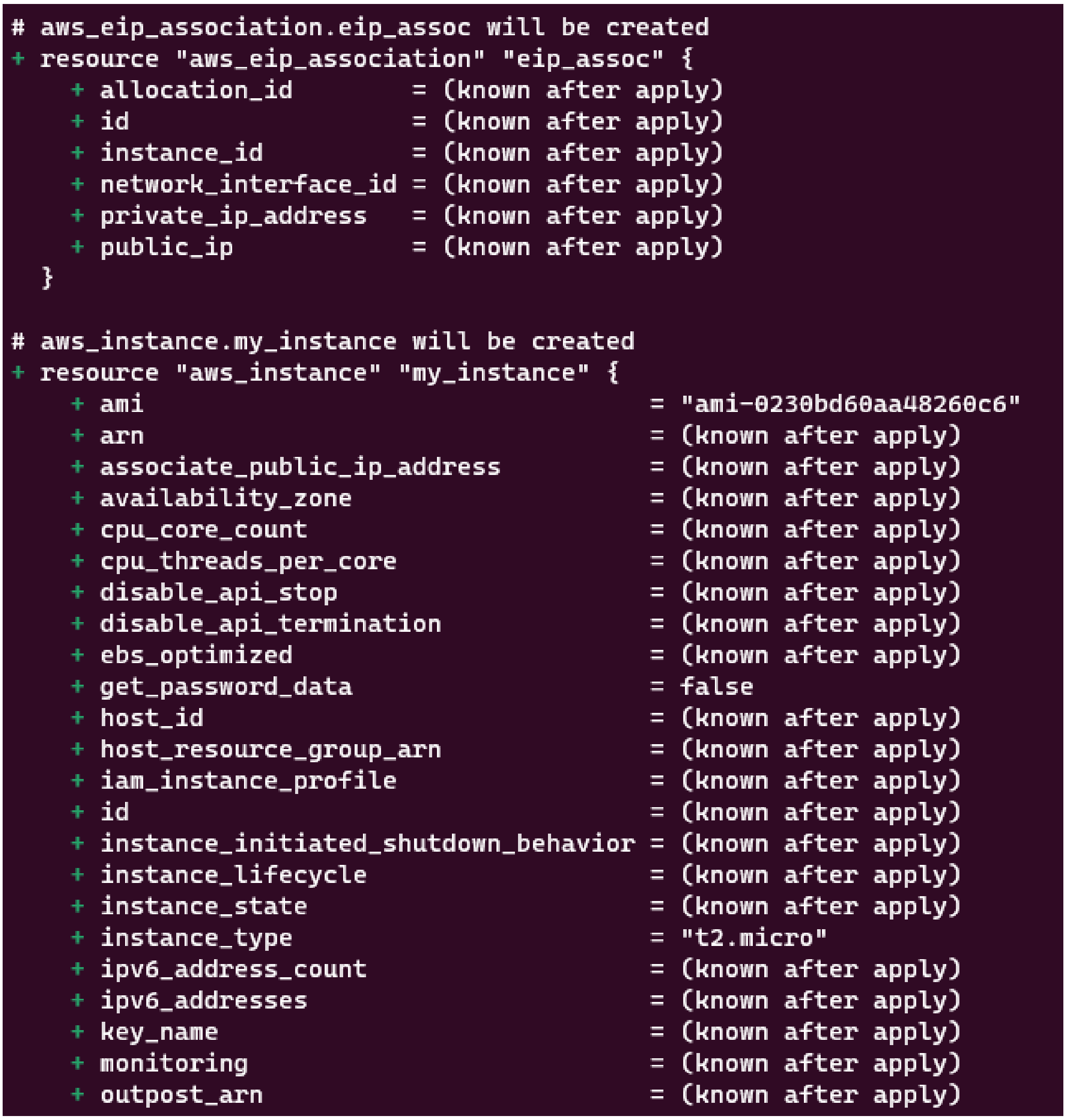


```
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (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" = "ec2_instance"
        }
      + tags_all                             = {
          + "Name" = "ec2_instance"
        }
      + 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: 3 to add, 0 to change, 0 to destroy.
```

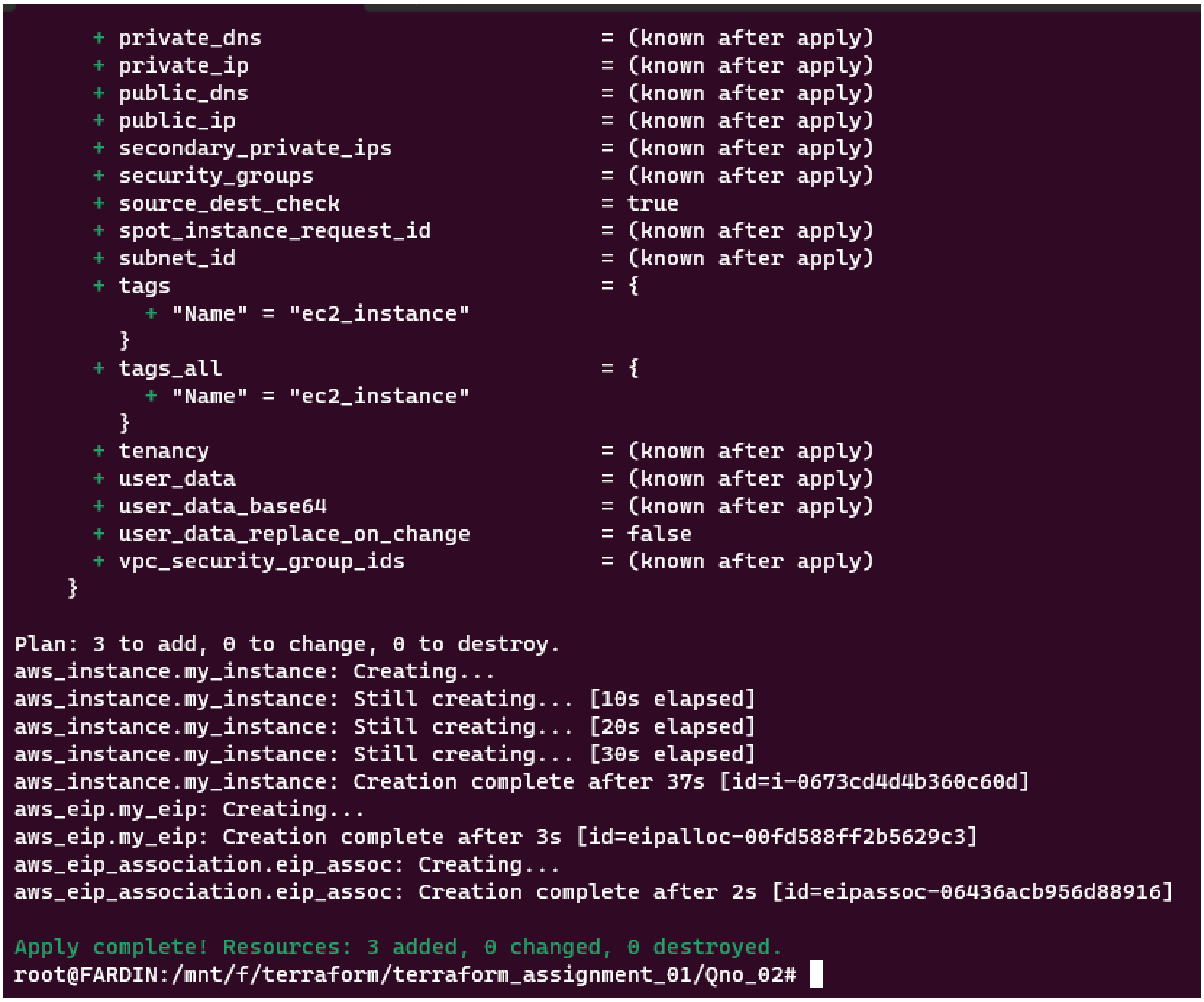
terraform apply



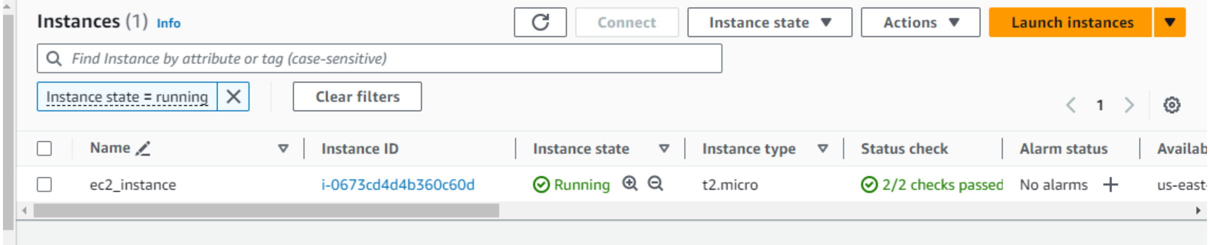
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_02# terraform apply -var-file=fardin.tfvars -auto-approve

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_eip.my_eip will be created
  + resource "aws_eip" "my_eip" {
      + 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)
    }
```



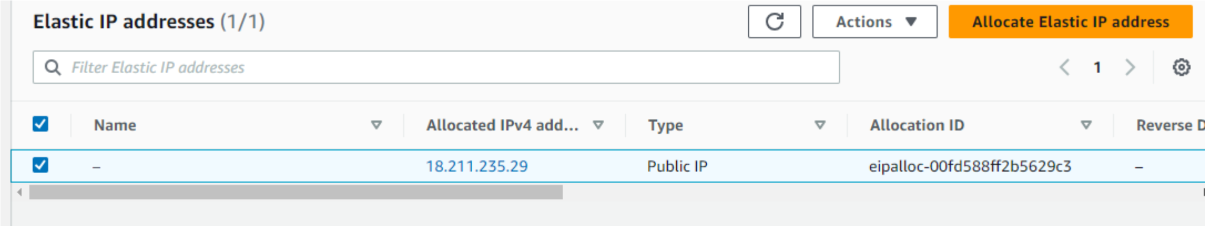
```
# aws_eip_association.eip_assoc will be created
+ resource "aws_eip_association" "eip_assoc" {
    + allocation_id        = (known after apply)
    + id                   = (known after apply)
    + instance_id          = (known after apply)
    + network_interface_id = (known after apply)
    + private_ip_address   = (known after apply)
    + public_ip            = (known after apply)
  }

# aws_instance.my_instance will be created
+ resource "aws_instance" "my_instance" {
    + 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)
```



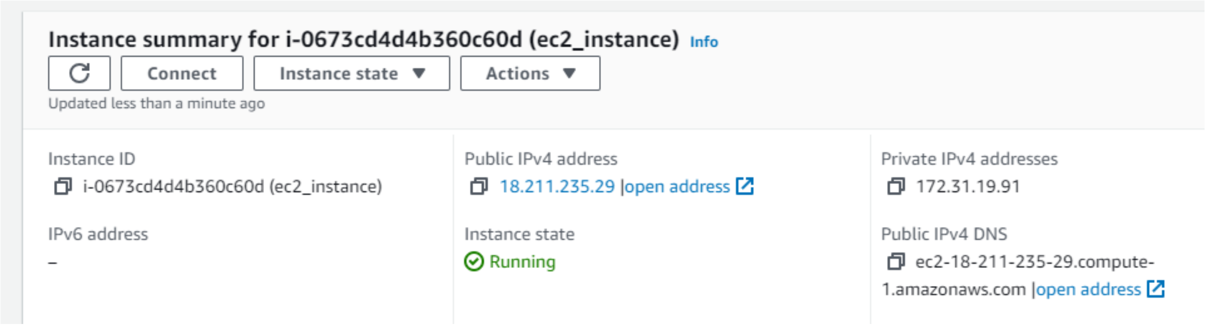
```
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (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" = "ec2_instance"
        }
      + tags_all                             = {
          + "Name" = "ec2_instance"
        }
      + 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: 3 to add, 0 to change, 0 to destroy.
aws_instance.my_instance: Creating...
aws_instance.my_instance: Still creating... [10s elapsed]
aws_instance.my_instance: Still creating... [20s elapsed]
aws_instance.my_instance: Still creating... [30s elapsed]
aws_instance.my_instance: Creation complete after 37s [id=i-0673cd4d4b360c60d]
aws_eip.my_eip: Creating...
aws_eip.my_eip: Creation complete after 3s [id=eipalloc-00fd588ff2b5629c3]
aws_eip_association.eip_assoc: Creating...
aws_eip_association.eip_assoc: Creation complete after 2s [id=eipassoc-06436acb956d88916]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_02#
```



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availab
□	ec2_instance	i-0673cd4d4b360c60d	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east


Elastic IP addresses (1/1)
Actions ▼
Allocate Elastic IP address
Filter Elastic IP addresses
< 1 >

☑	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse D
☑	–	18.211.235.29	Public IP	eipalloc-00fd588ff2b5629c3	–



Instance summary for i-0673cd4d4b360c60d (ec2_instance) Info

Updated less than a minute ago

Instance ID
i-0673cd4d4b360c60d (ec2_instance)

Public IPv4 address
18.211.235.29 |open address

Private IPv4 addresses
172.31.19.91

IPv6 address
–

Instance state
Running

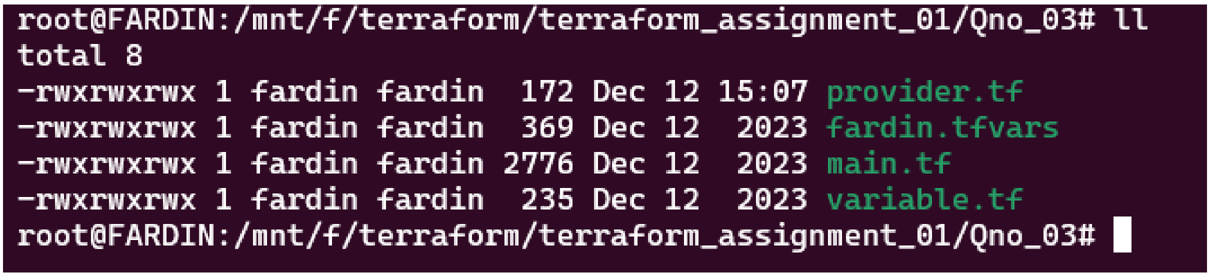
Public IPv4 DNS
ec2-18-211-235-29.compute-1.amazonaws.com |open address

03.

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

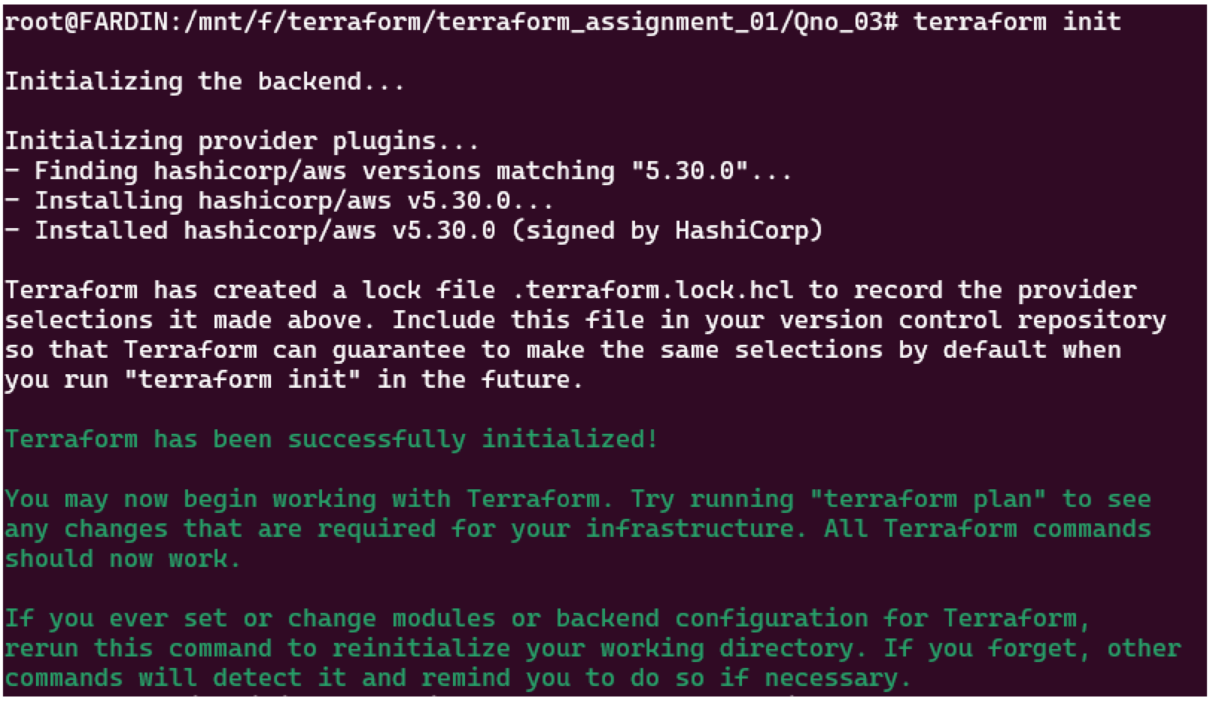
    1. Create a VPC.
    2. Create 2 Public Subnet & Create 2 Private Subnet.
    3. Create IGW (Internet Gateway) & Attach to the VPC.
    4. Create Public and Private Route Table.
    5. Add IGW in Public Route table (0.0.0.0/0).
    6. Add Public Subnet (1a & 1b) in Route table.
    7. Create a NAT Gateway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

ANS:-



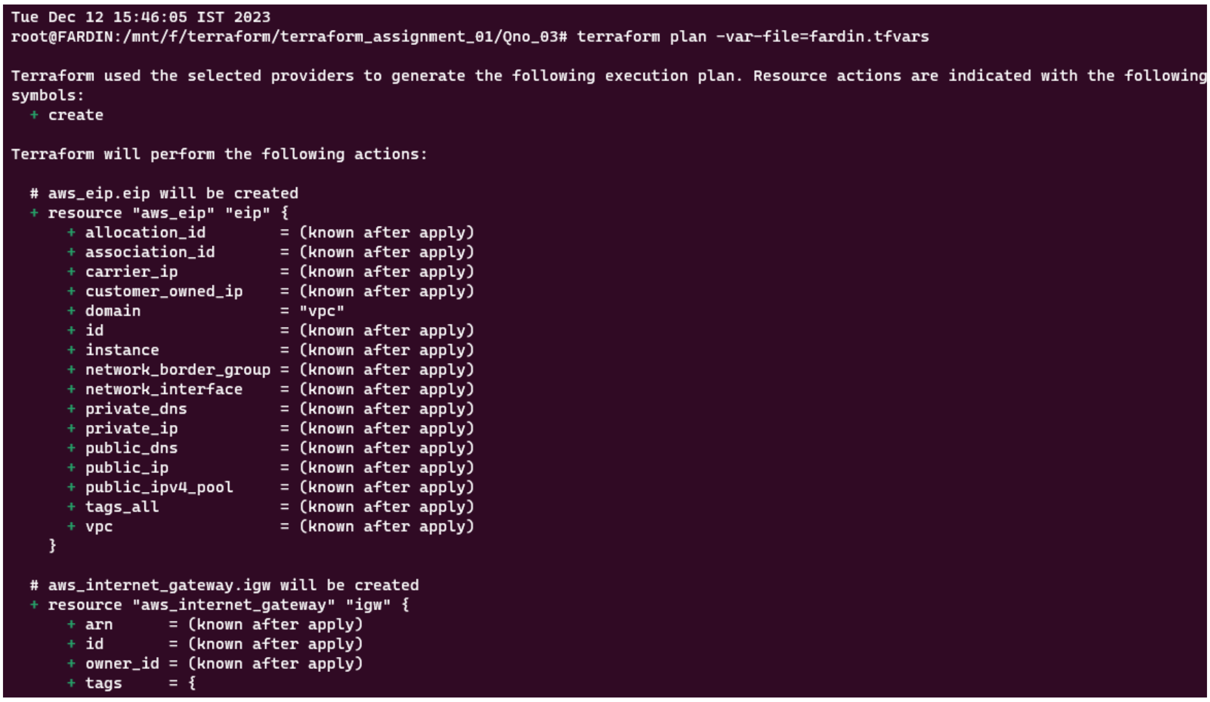
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_03# ll
total 8
-rwxrwxrwx 1 fardin fardin  172 Dec 12 15:07 provider.tf
-rwxrwxrwx 1 fardin fardin  369 Dec 12  2023 fardin.tfvars
-rwxrwxrwx 1 fardin fardin 2776 Dec 12  2023 main.tf
-rwxrwxrwx 1 fardin fardin  235 Dec 12  2023 variable.tf
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_03# 
```

Terraform init

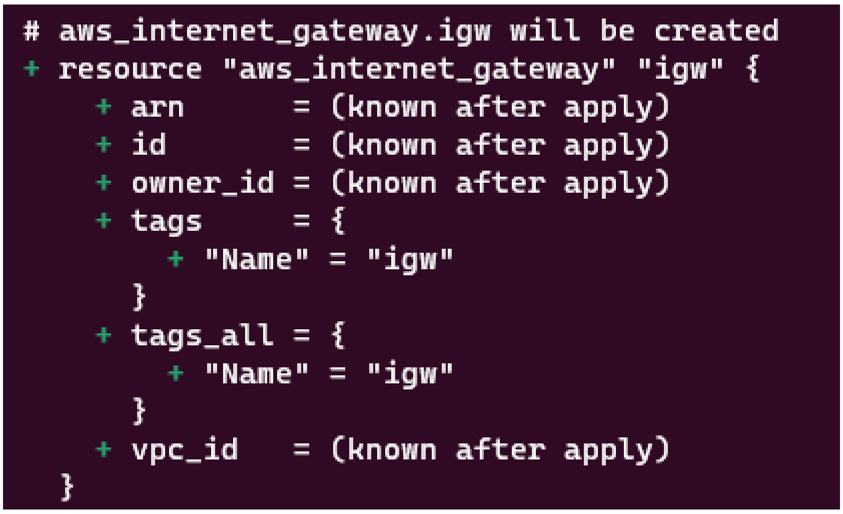


```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_03# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.30.0"...
- 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.
```

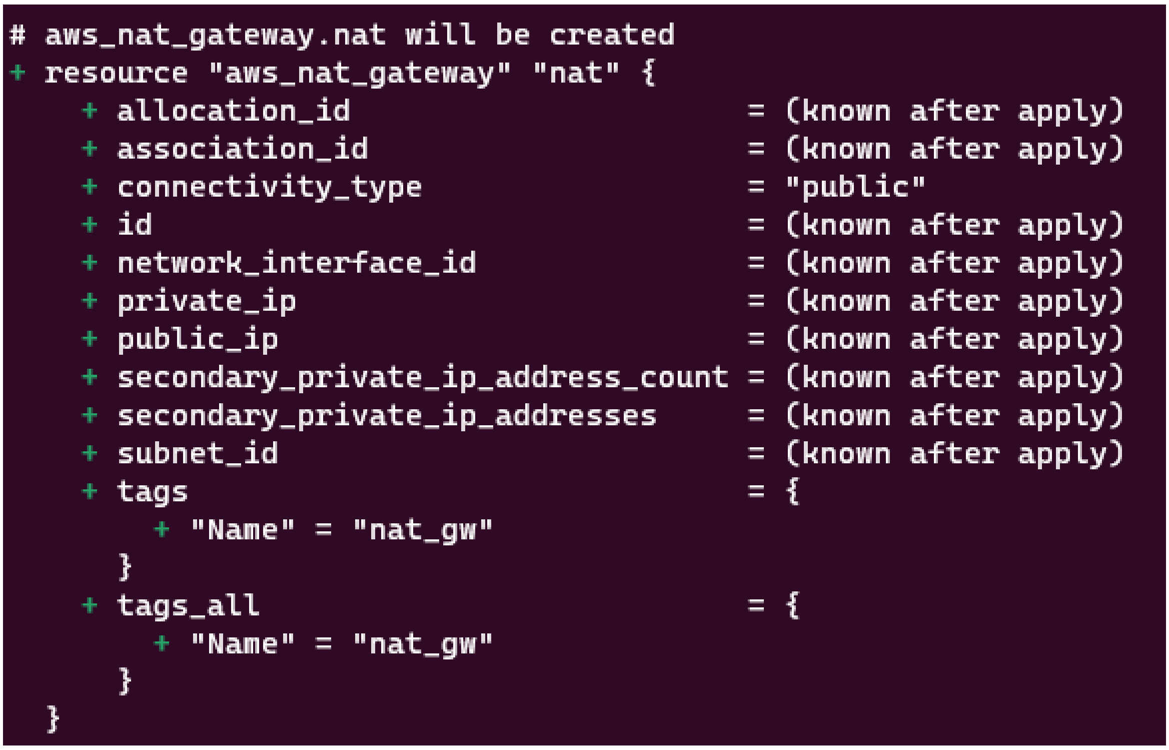
Terraform plan



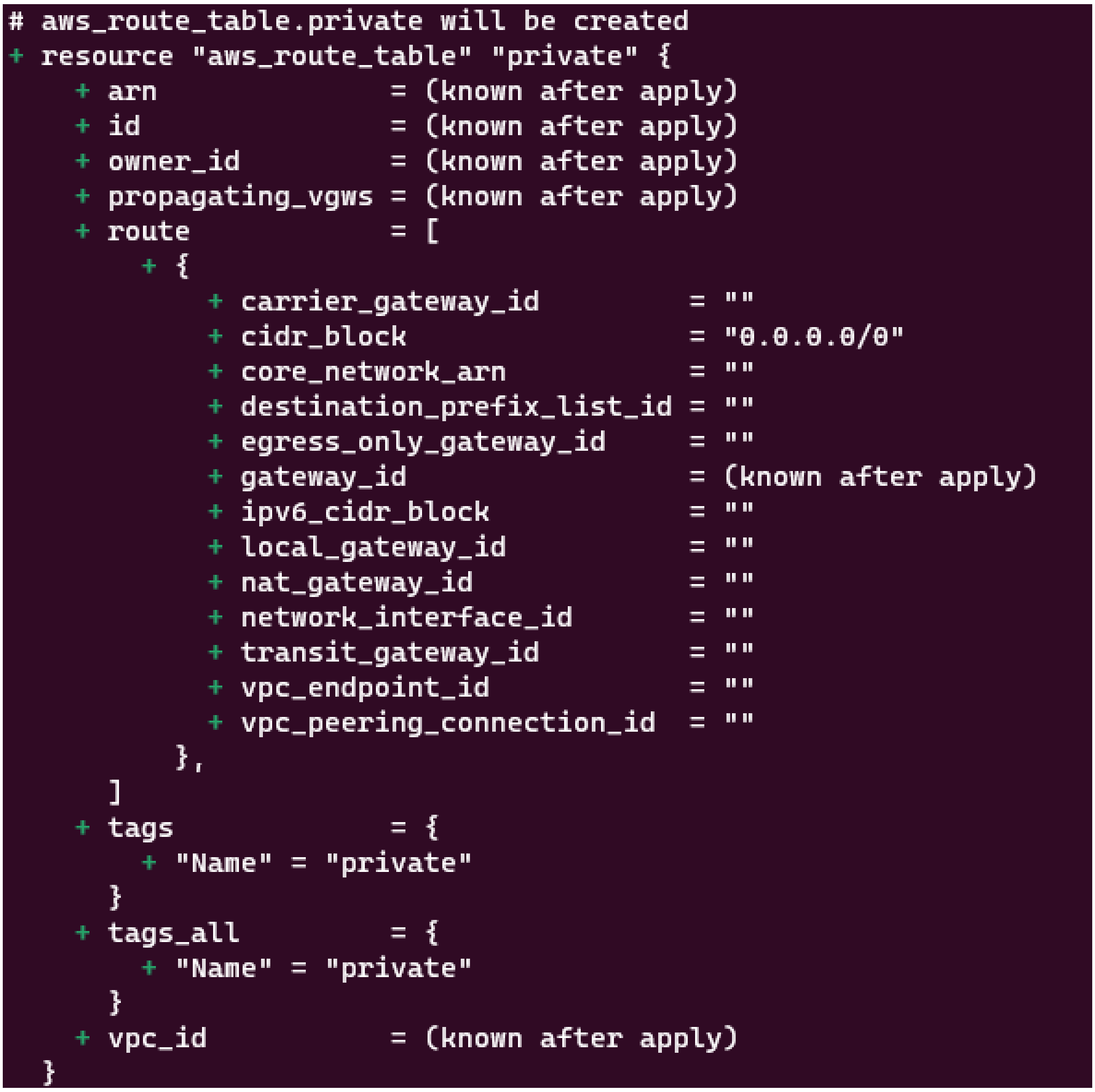
```
Tue Dec 12 15:46:05 IST 2023
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_03# terraform plan -var-file=fardin.tfvars

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_eip.eip will be created
  + resource "aws_eip" "eip" {
      + 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_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     = {
```



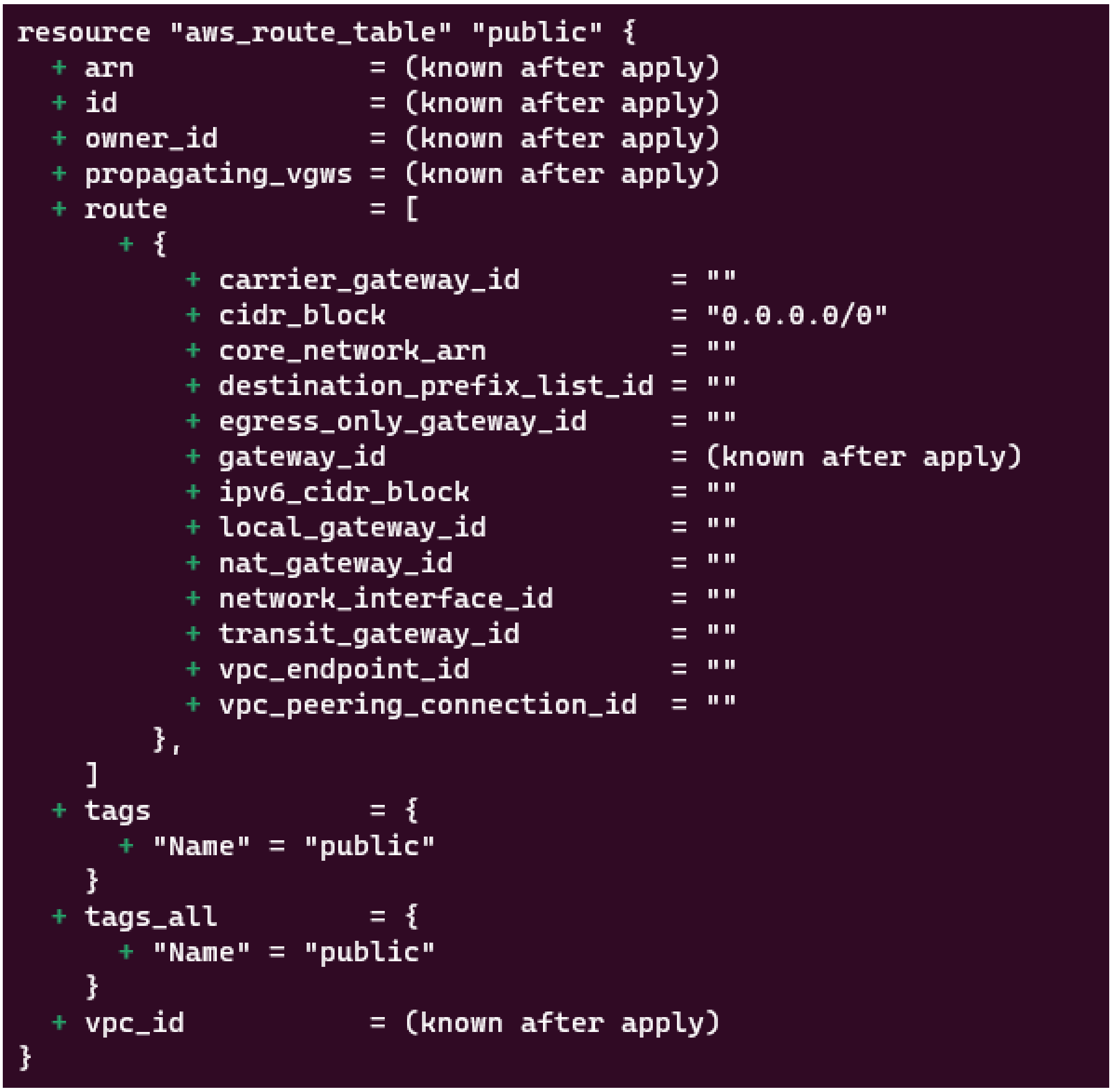
```
# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "igw"
      }
    + tags_all = {
        + "Name" = "igw"
    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



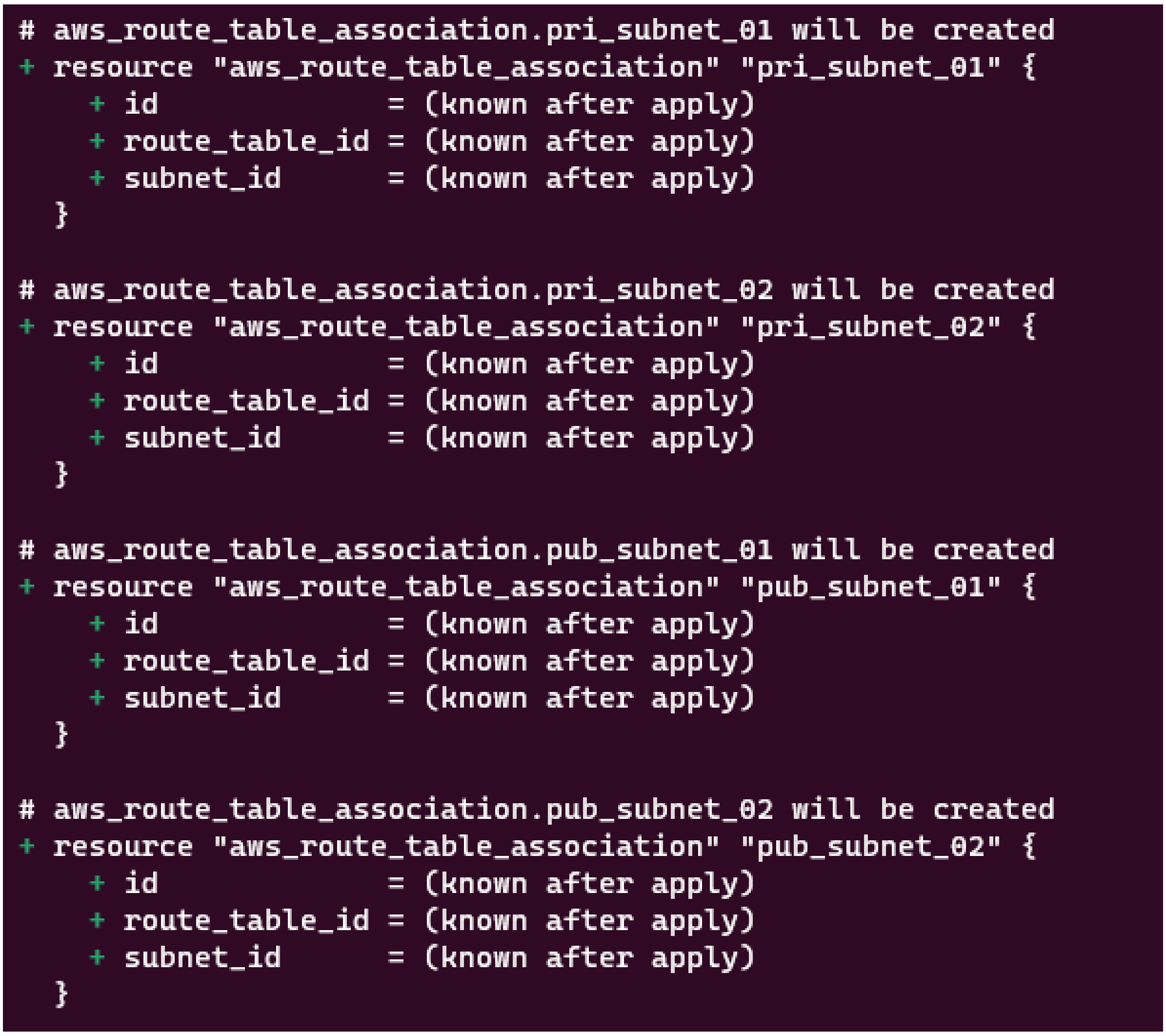
```
# aws_nat_gateway.nat will be created
+ resource "aws_nat_gateway" "nat" {
    + allocation_id                      = (known after apply)
    + association_id                     = (known after apply)
    + connectivity_type                  = "public"
    + id                                 = (known 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" = "nat_gw"
      }
    + tags_all                           = {
        + "Name" = "nat_gw"
      }
  }
```



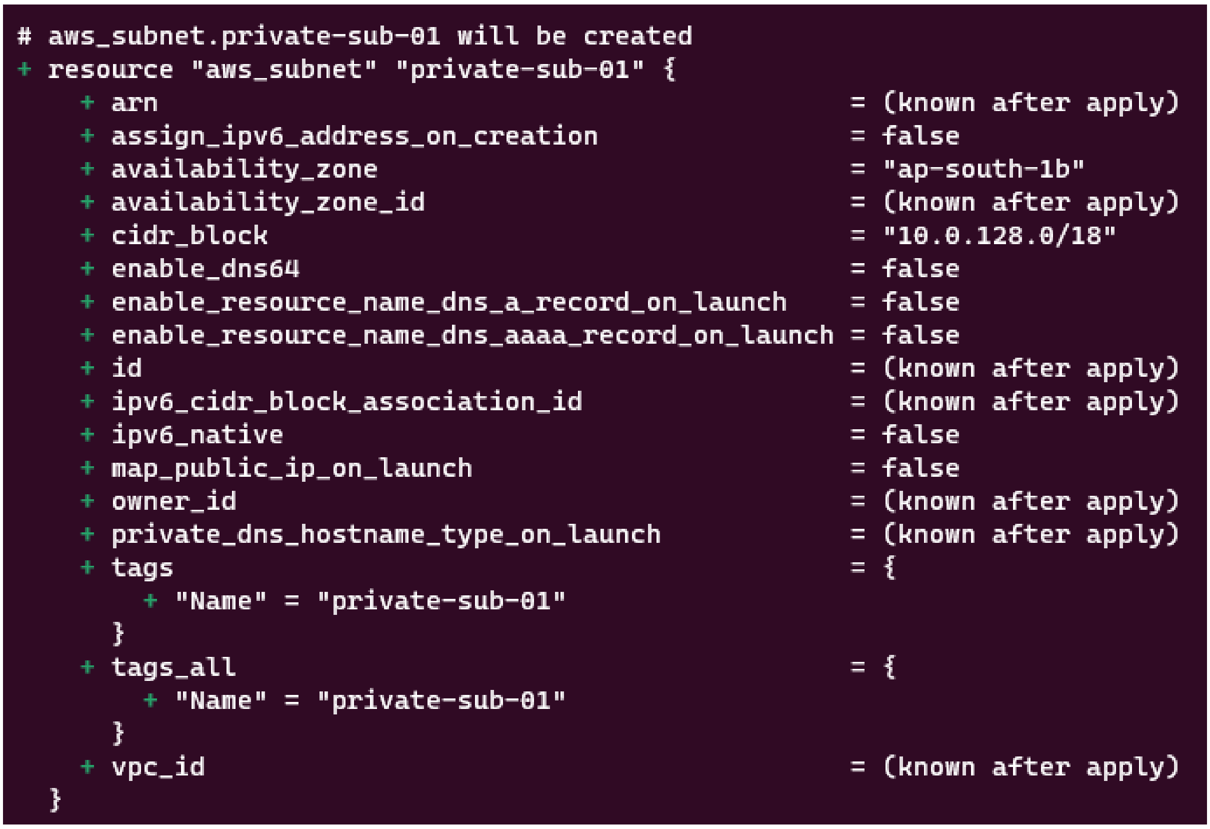
```
# aws_route_table.private will be created
+ resource "aws_route_table" "private" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + 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" = "private"
      }
    + tags_all         = {
        + "Name" = "private"
      }
    + vpc_id           = (known after apply)
  }
```



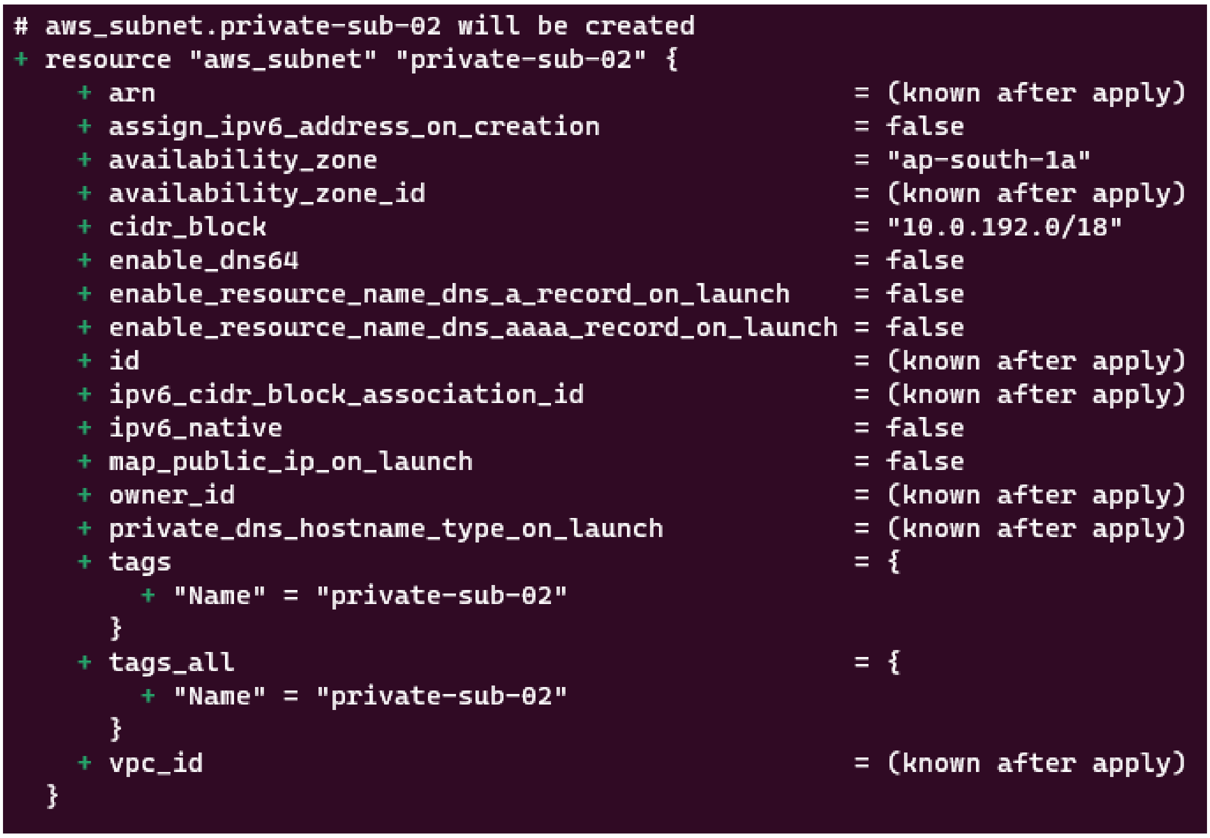
```
resource "aws_route_table" "public" {
  + arn              = (known after apply)
  + id               = (known after apply)
  + owner_id         = (known after apply)
  + propagating_vgws = (known after apply)
  + route            = [
      + {
          + carrier_gateway_id         = ""
          + 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"
    }
  + tags_all         = {
      + "Name" = "public"
    }
  + vpc_id           = (known after apply)
}
```



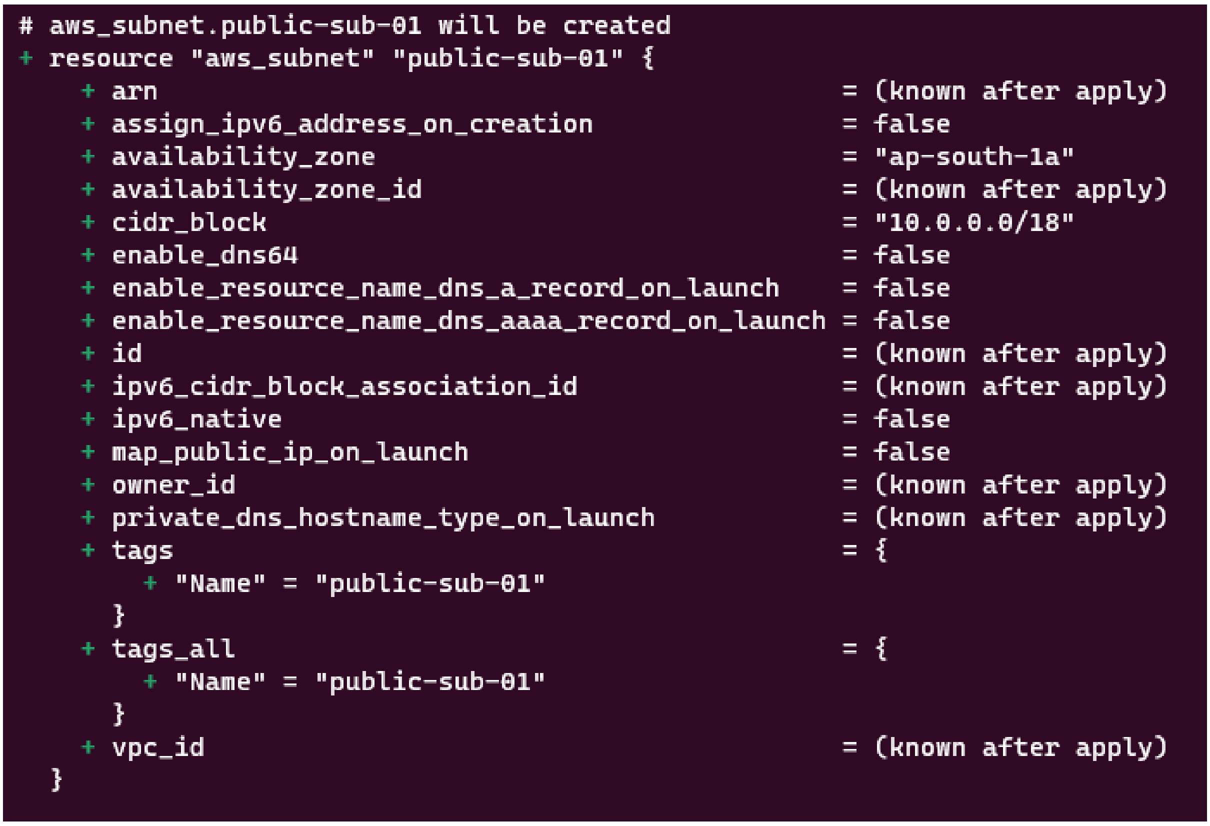
```
# aws_route_table_association.pri_subnet_01 will be created
+ resource "aws_route_table_association" "pri_subnet_01" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.pri_subnet_02 will be created
+ resource "aws_route_table_association" "pri_subnet_02" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.pub_subnet_01 will be created
+ resource "aws_route_table_association" "pub_subnet_01" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.pub_subnet_02 will be created
+ resource "aws_route_table_association" "pub_subnet_02" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }
```



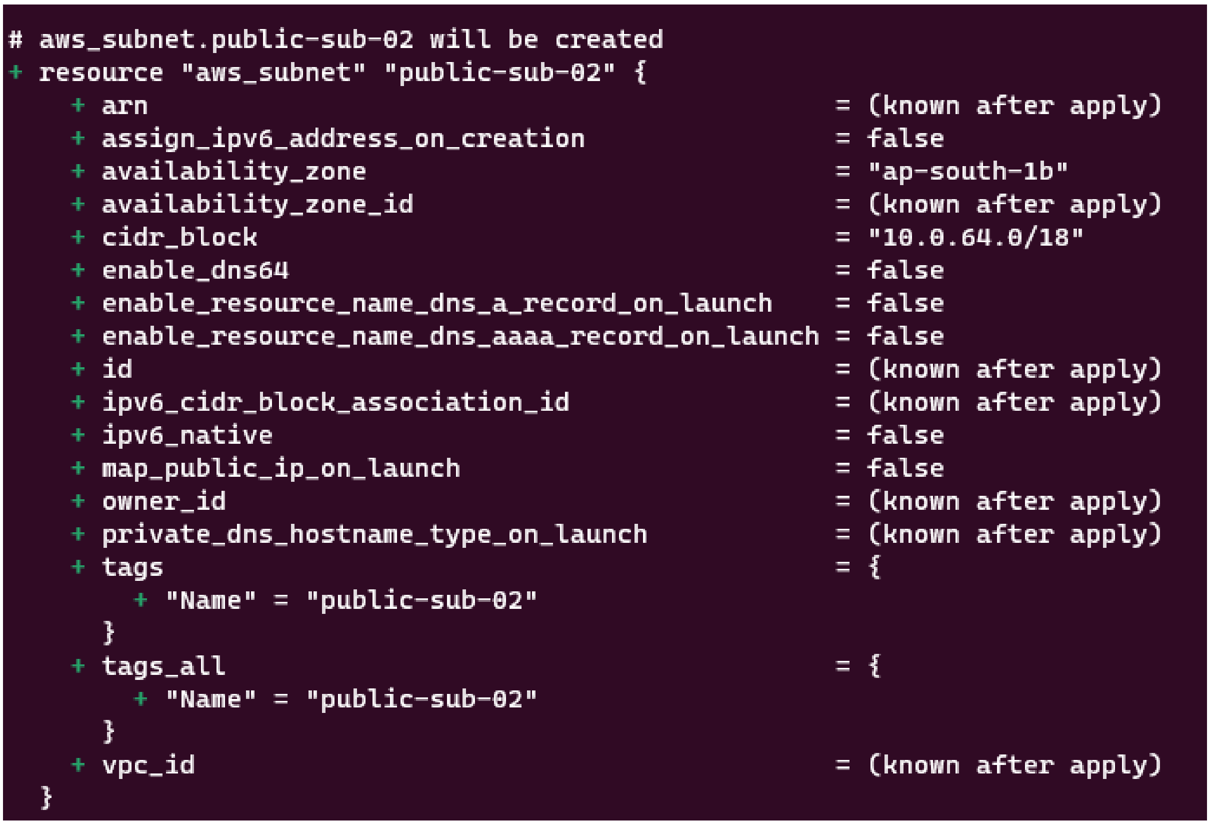
```
# aws_subnet.private-sub-01 will be created
+ resource "aws_subnet" "private-sub-01" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1b"
    + 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-sub-01"
      }
    + tags_all                                       = {
        + "Name" = "private-sub-01"
      }
    + vpc_id                                         = (known after apply)
  }
```



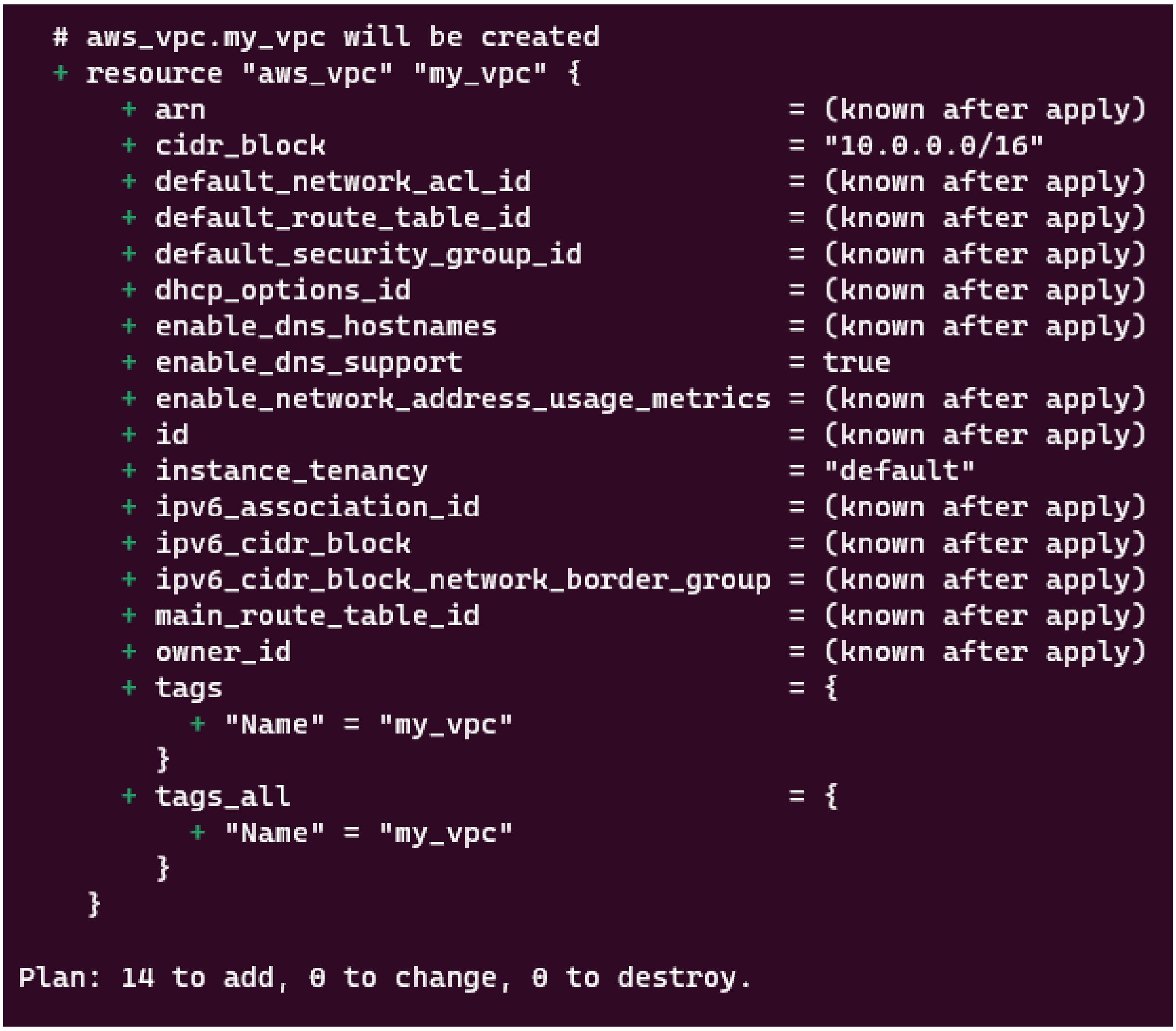
```
# aws_subnet.private-sub-02 will be created
+ resource "aws_subnet" "private-sub-02" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1a"
    + 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-sub-02"
      }
    + tags_all                                       = {
        + "Name" = "private-sub-02"
      }
    + vpc_id                                         = (known after apply)
  }
```



```
# aws_subnet.public-sub-01 will be created
+ resource "aws_subnet" "public-sub-01" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-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                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "public-sub-01"
      }
    + tags_all                                       = {
        + "Name" = "public-sub-01"
      }
    + vpc_id                                         = (known after apply)
  }
```

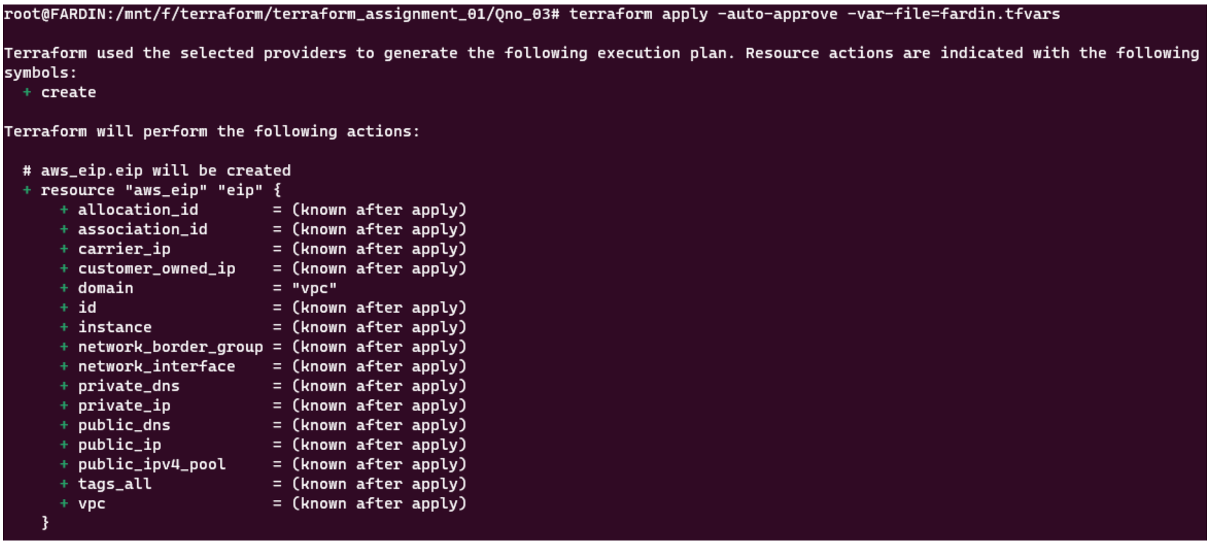


```
# aws_subnet.public-sub-02 will be created
+ resource "aws_subnet" "public-sub-02" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1b"
    + availability_zone_id                           = (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                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "public-sub-02"
      }
    + tags_all                                       = {
        + "Name" = "public-sub-02"
      }
    + vpc_id                                         = (known after apply)
  }
```

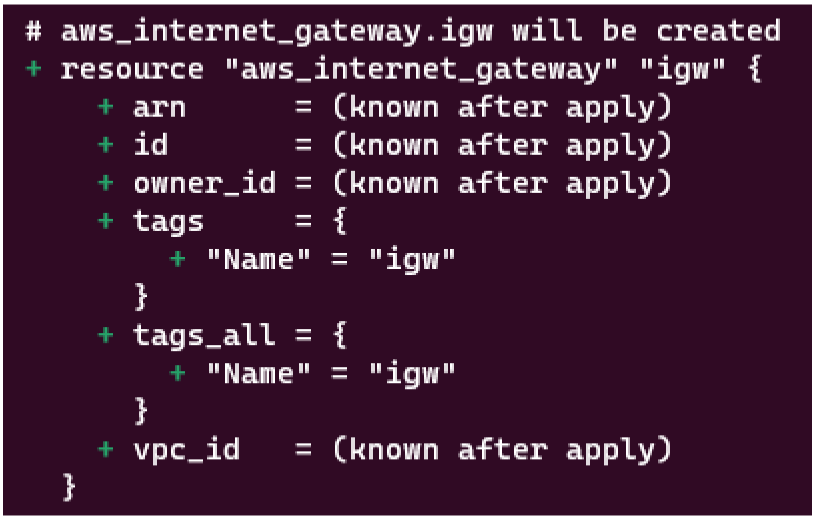


```
  # aws_vpc.my_vpc will be created
  + resource "aws_vpc" "my_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "my_vpc"
        }
      + tags_all                             = {
          + "Name" = "my_vpc"
        }
    }

Plan: 14 to add, 0 to change, 0 to destroy.
```

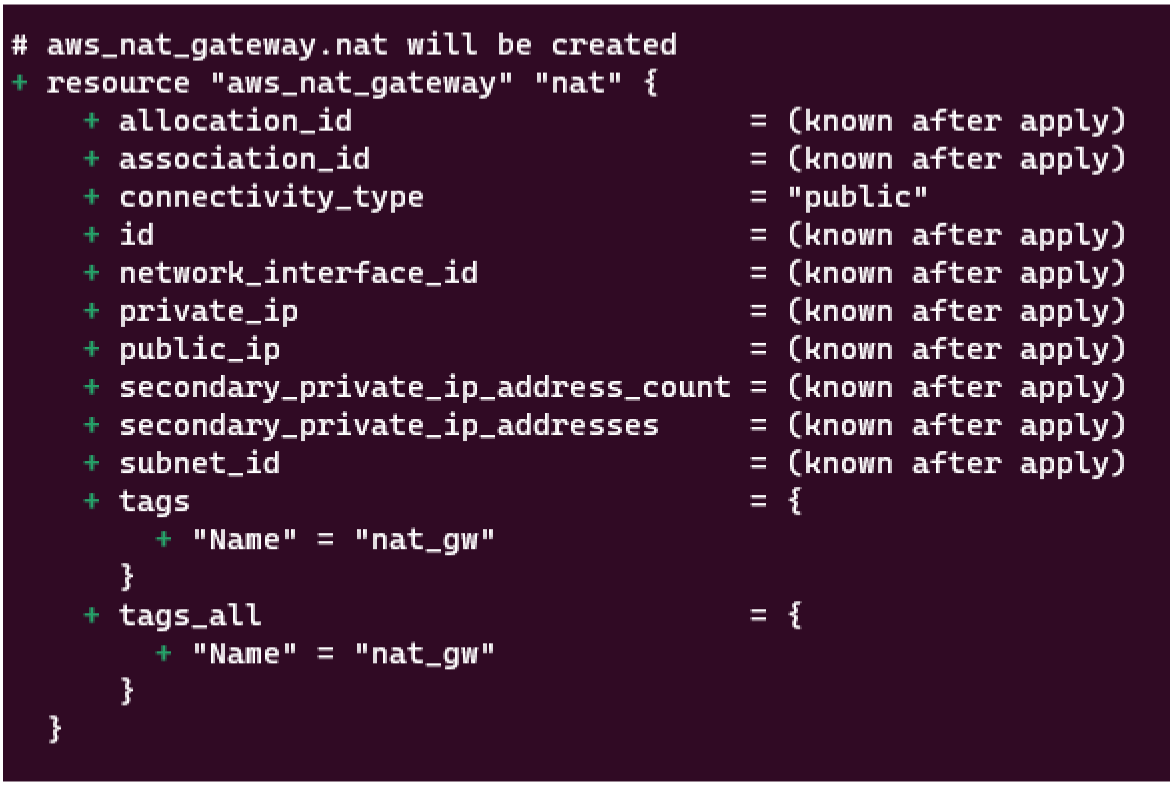
Terraform apply



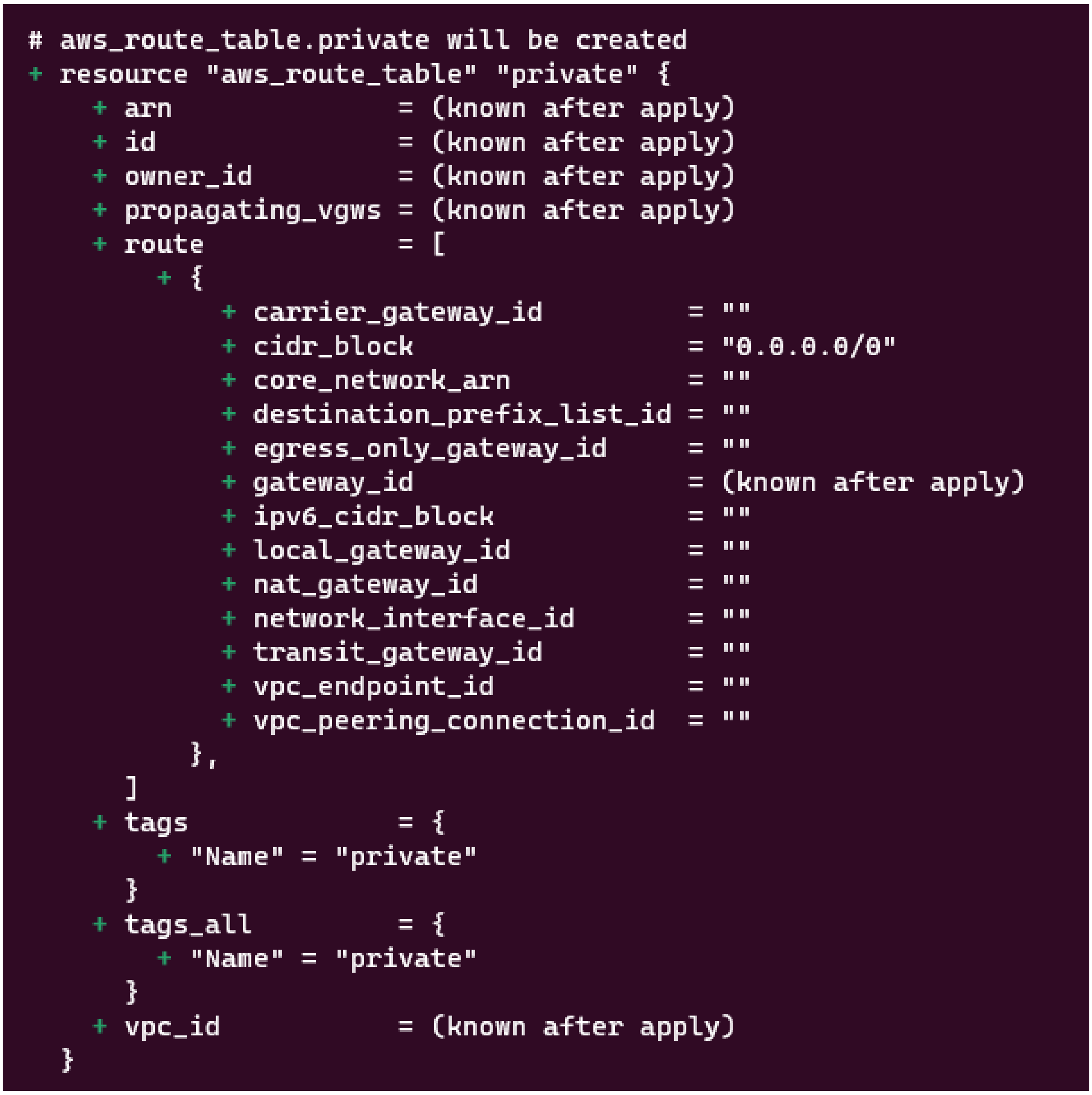
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_03# terraform apply -auto-approve -var-file=fardin.tfvars

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_eip.eip will be created
  + resource "aws_eip" "eip" {
      + 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)
    }
```



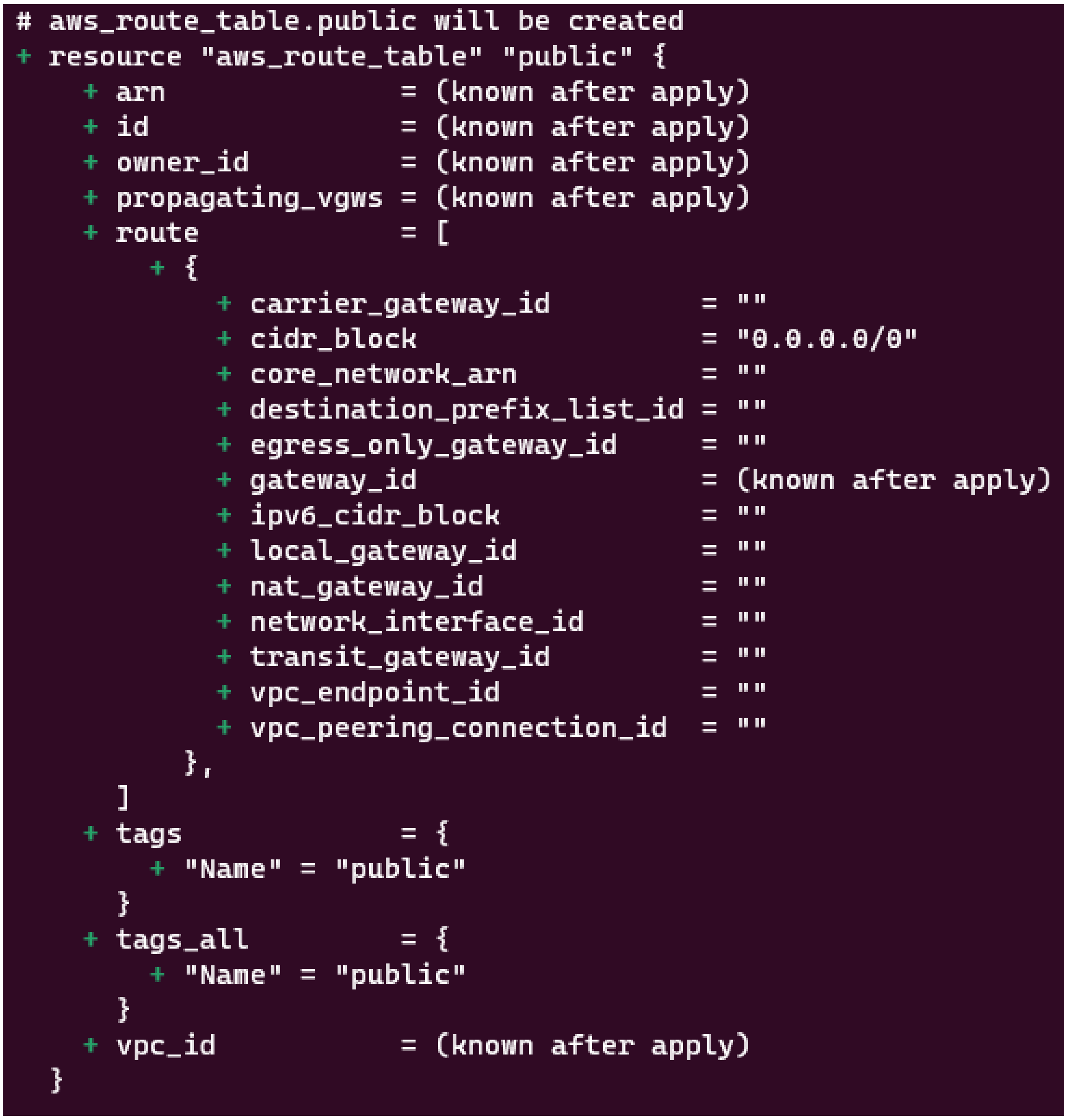
```
# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "igw"
      }
    + tags_all = {
        + "Name" = "igw"
    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



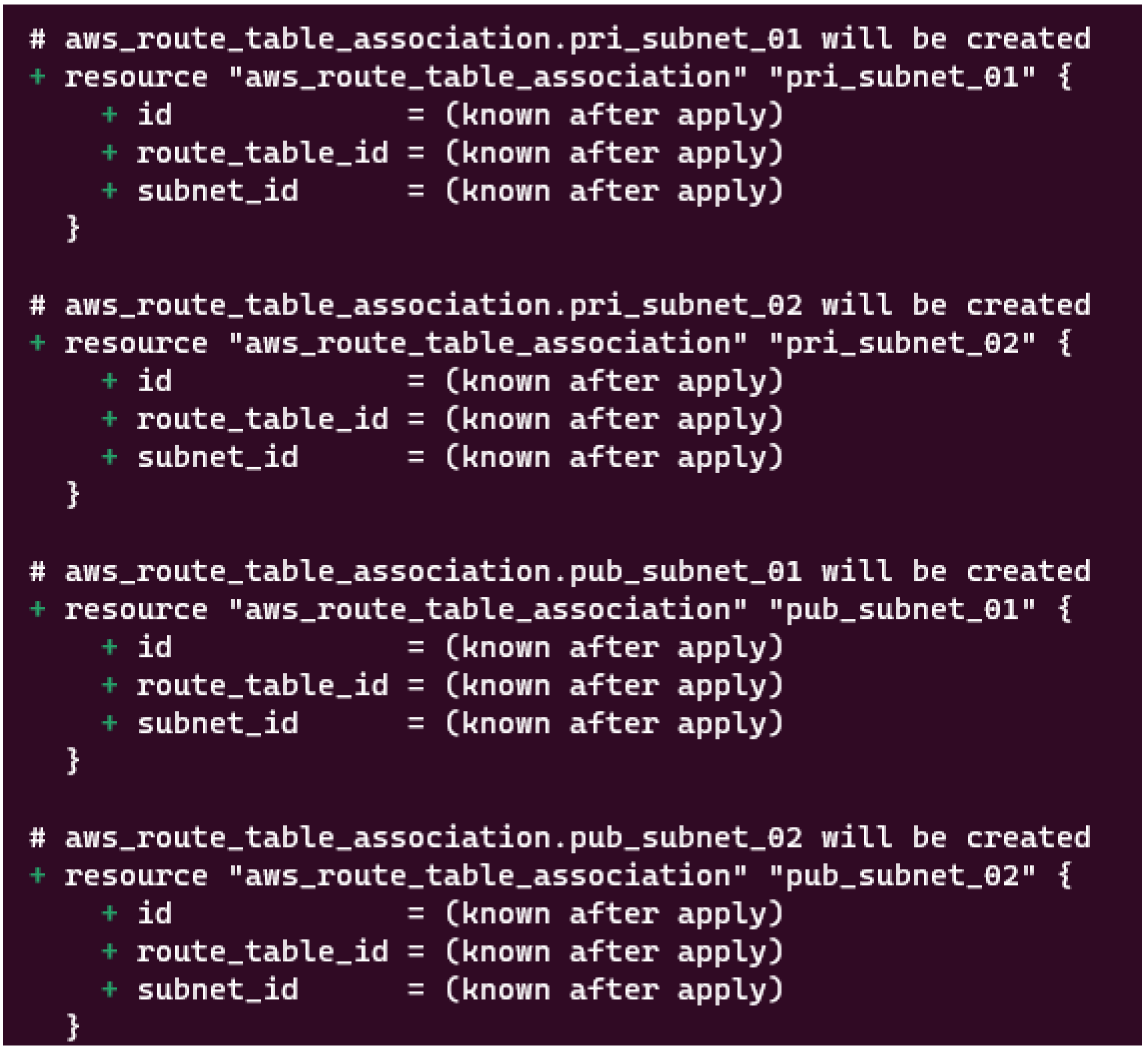
```
# aws_nat_gateway.nat will be created
+ resource "aws_nat_gateway" "nat" {
    + allocation_id                      = (known after apply)
    + association_id                     = (known after apply)
    + connectivity_type                  = "public"
    + id                                 = (known 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" = "nat_gw"
      }
    + tags_all                           = {
        + "Name" = "nat_gw"
      }
  }
```



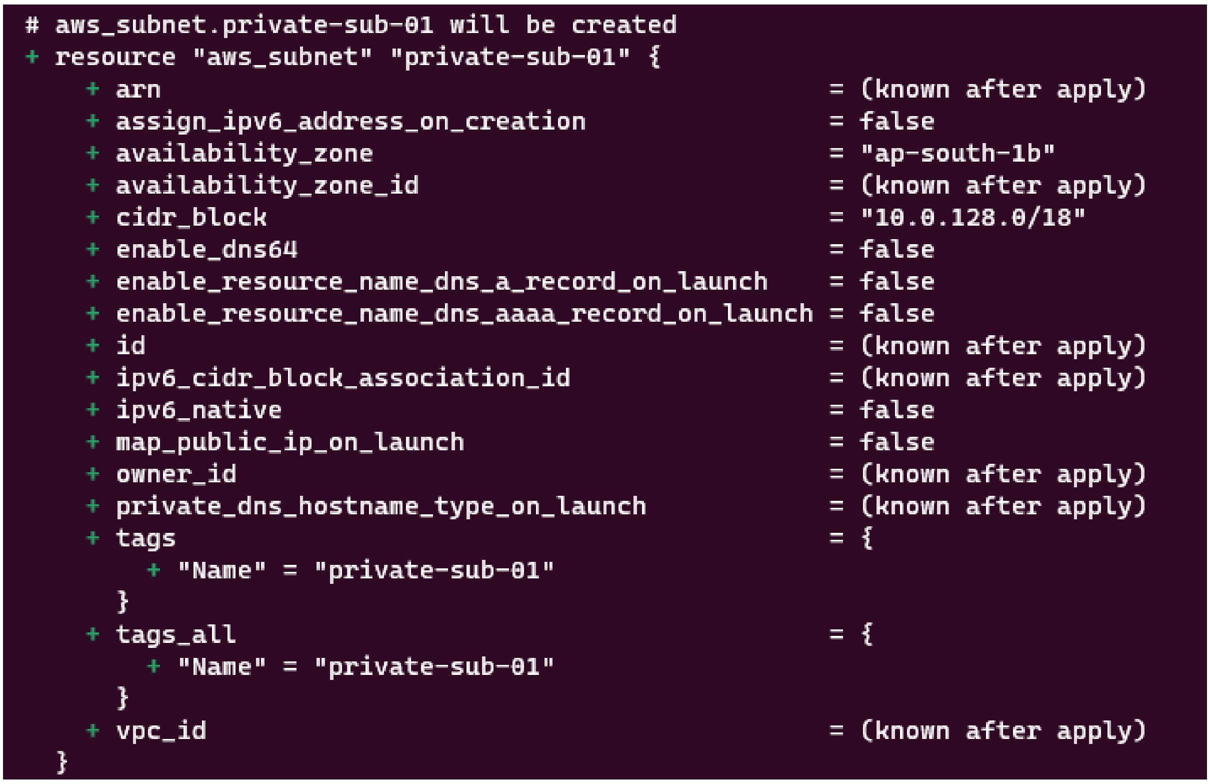
```
# aws_route_table.private will be created
+ resource "aws_route_table" "private" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + 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" = "private"
      }
    + tags_all         = {
        + "Name" = "private"
      }
    + vpc_id           = (known after apply)
  }
```



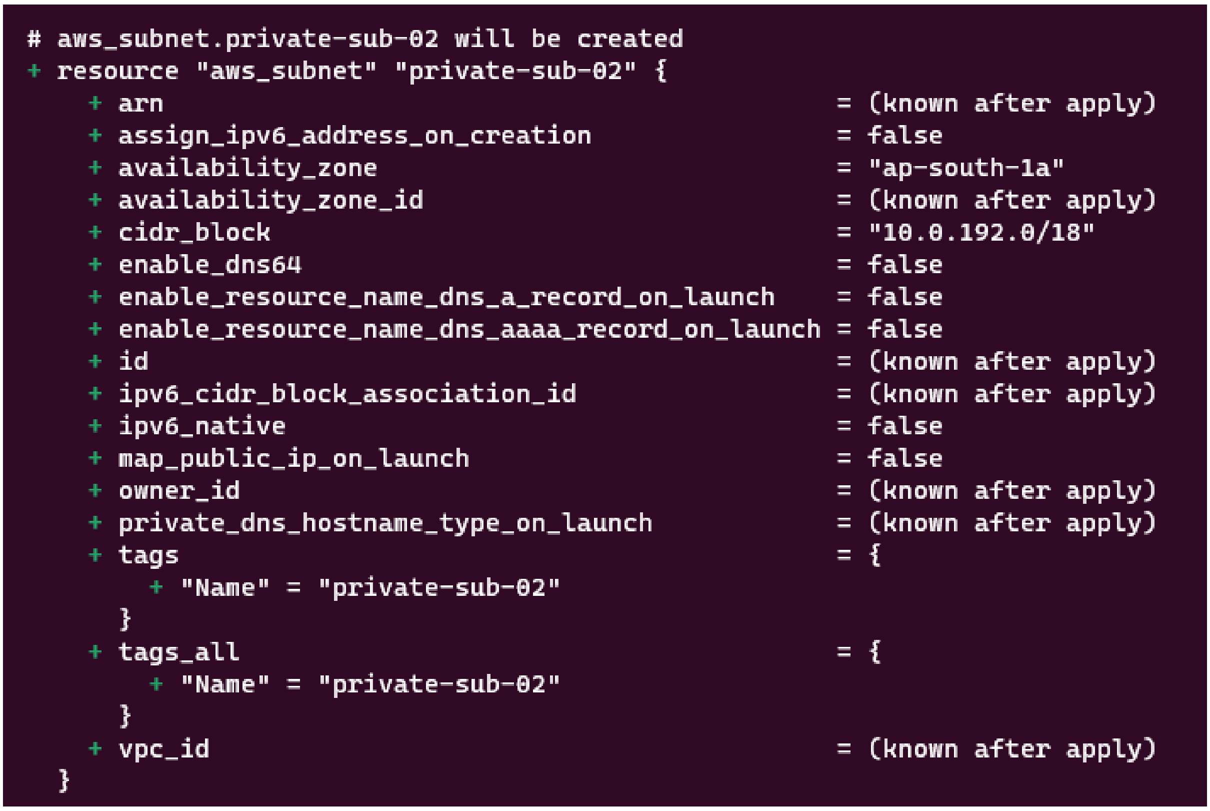
```
# aws_route_table.public will be created
+ resource "aws_route_table" "public" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + 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"
      }
    + tags_all         = {
        + "Name" = "public"
      }
    + vpc_id           = (known after apply)
  }
```



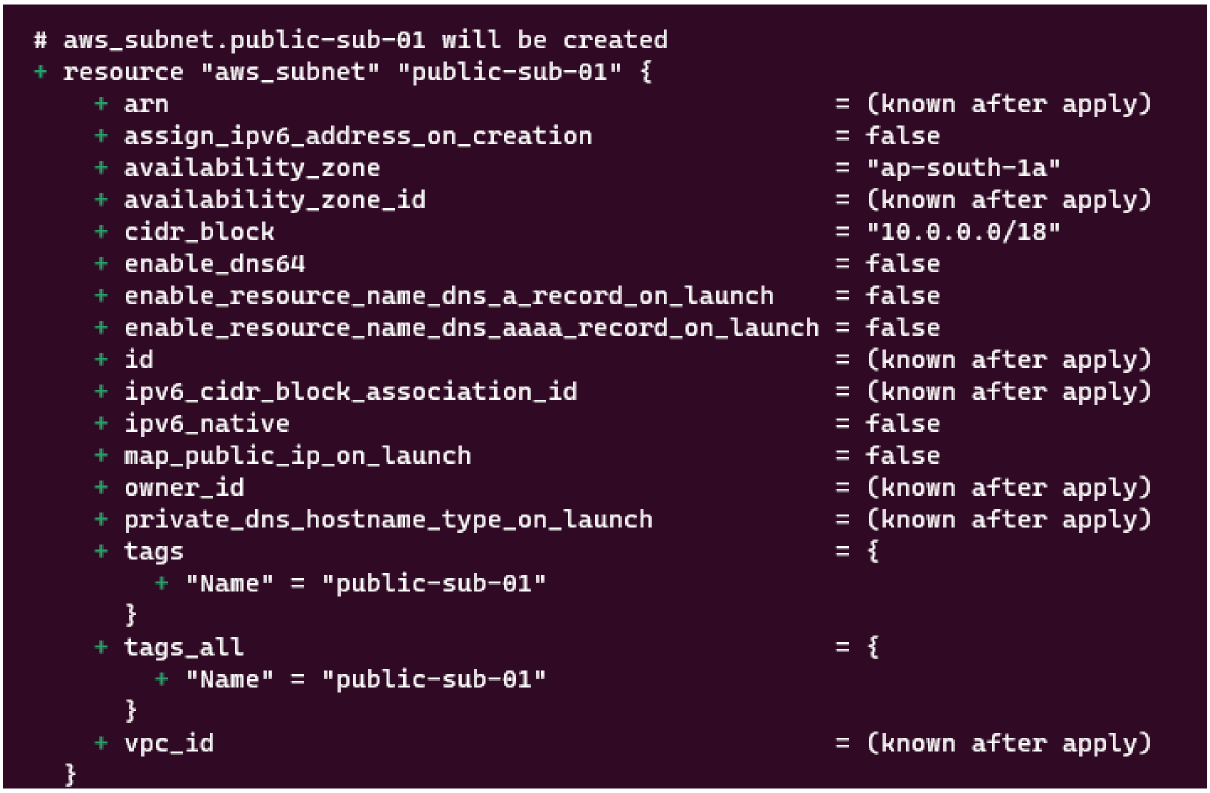
```
# aws_route_table_association.pri_subnet_01 will be created
+ resource "aws_route_table_association" "pri_subnet_01" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.pri_subnet_02 will be created
+ resource "aws_route_table_association" "pri_subnet_02" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.pub_subnet_01 will be created
+ resource "aws_route_table_association" "pub_subnet_01" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.pub_subnet_02 will be created
+ resource "aws_route_table_association" "pub_subnet_02" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }
```



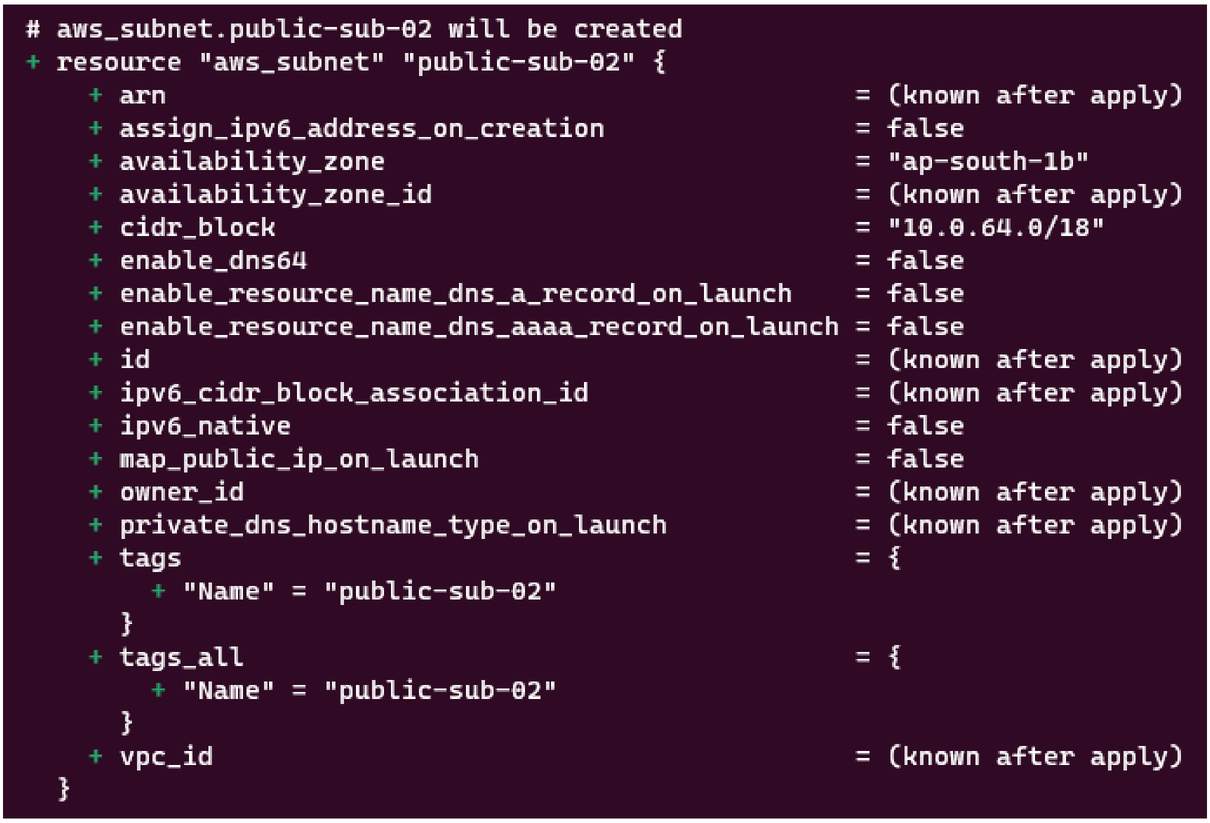
```
# aws_subnet.private-sub-01 will be created
+ resource "aws_subnet" "private-sub-01" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1b"
    + 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-sub-01"
      }
    + tags_all                                       = {
        + "Name" = "private-sub-01"
      }
    + vpc_id                                         = (known after apply)
  }
```



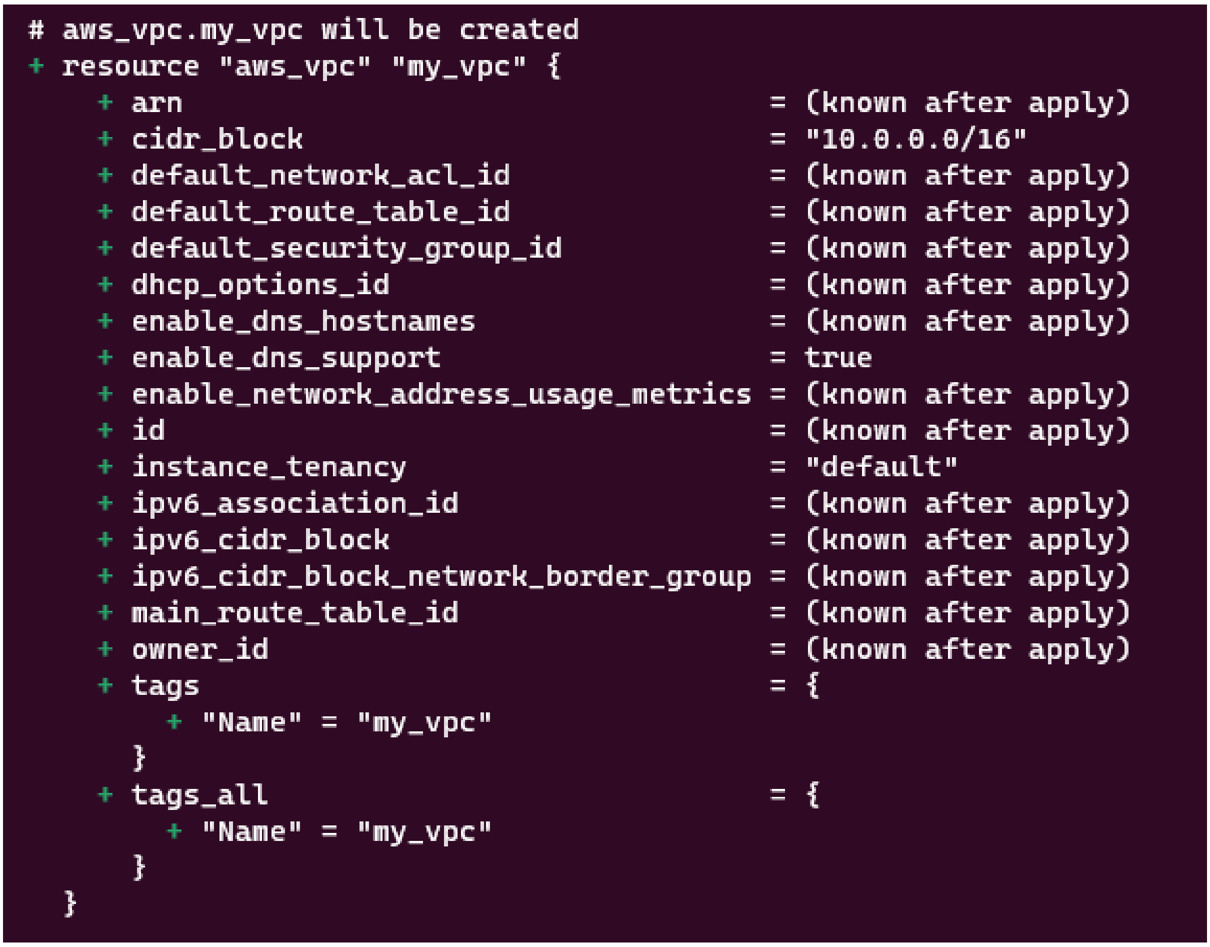
```
# aws_subnet.private-sub-02 will be created
+ resource "aws_subnet" "private-sub-02" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1a"
    + 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-sub-02"
      }
    + tags_all                                       = {
        + "Name" = "private-sub-02"
      }
    + vpc_id                                         = (known after apply)
  }
```



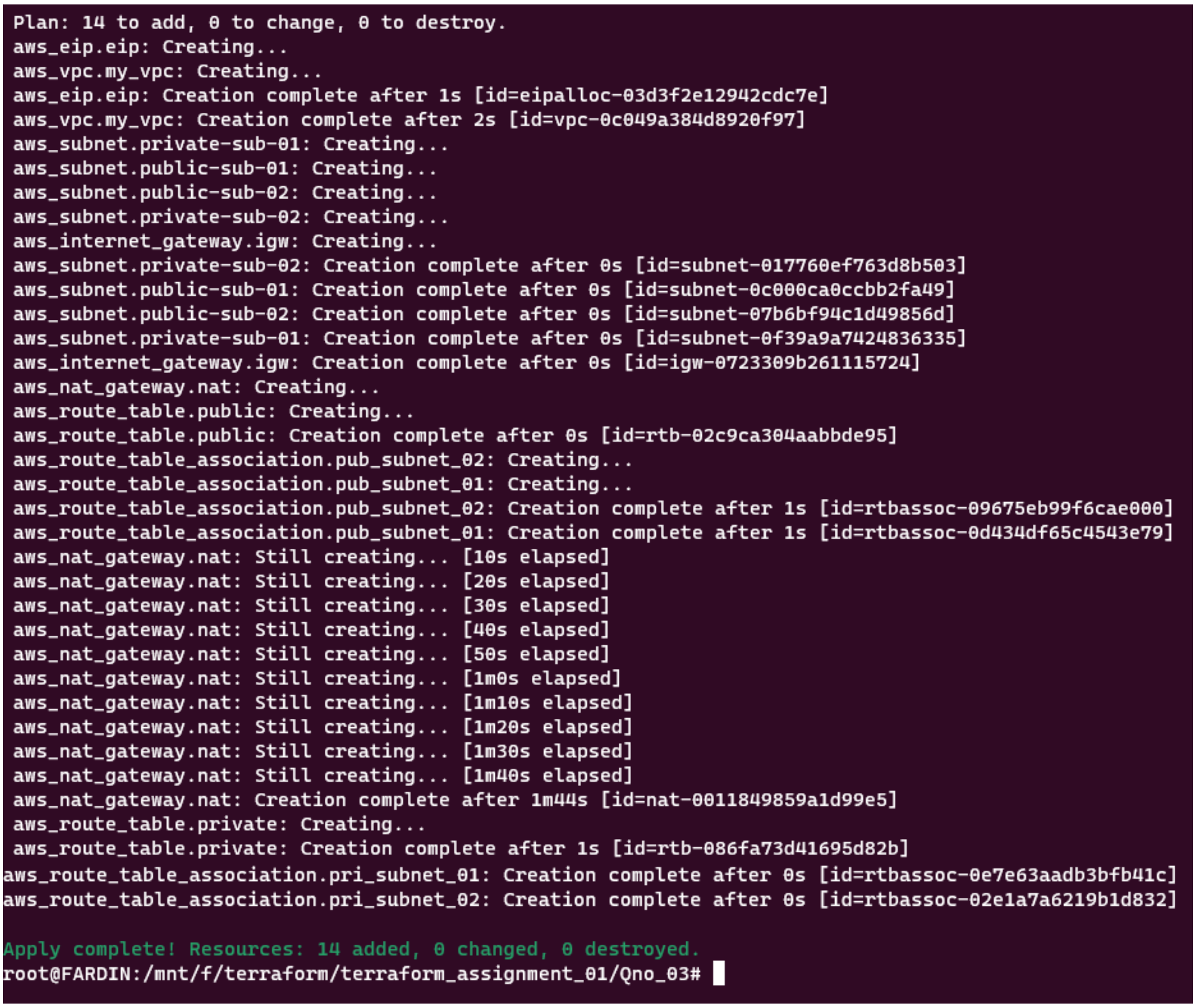
```
# aws_subnet.public-sub-01 will be created
+ resource "aws_subnet" "public-sub-01" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1a"
    + availability_zone_id                           = (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                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "public-sub-01"
      }
    + tags_all                                       = {
        + "Name" = "public-sub-01"
      }
    + vpc_id                                         = (known after apply)
  }
```



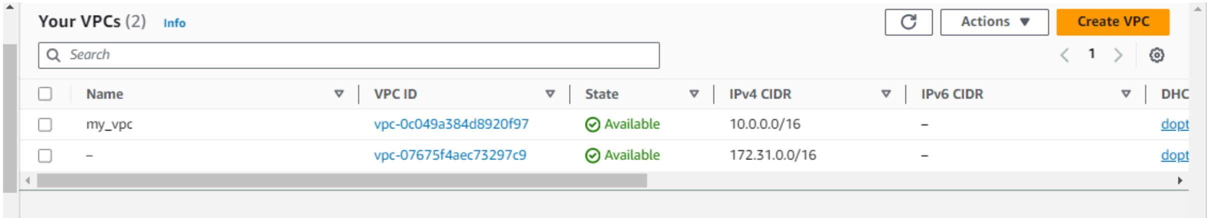
```
# aws_subnet.public-sub-02 will be created
+ resource "aws_subnet" "public-sub-02" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "ap-south-1b"
    + availability_zone_id                           = (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                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "public-sub-02"
      }
    + tags_all                                       = {
        + "Name" = "public-sub-02"
      }
    + vpc_id                                         = (known after apply)
  }
```



```
# aws_vpc.my_vpc will be created
+ resource "aws_vpc" "my_vpc" {
    + arn                                  = (known after apply)
    + cidr_block                           = "10.0.0.0/16"
    + default_network_acl_id               = (known after apply)
    + default_route_table_id               = (known after apply)
    + default_security_group_id            = (known after apply)
    + dhcp_options_id                      = (known after apply)
    + enable_dns_hostnames                 = (known after apply)
    + enable_dns_support                   = true
    + enable_network_address_usage_metrics = (known after apply)
    + id                                   = (known after apply)
    + instance_tenancy                     = "default"
    + ipv6_association_id                  = (known after apply)
    + ipv6_cidr_block                      = (known after apply)
    + ipv6_cidr_block_network_border_group = (known after apply)
    + main_route_table_id                  = (known after apply)
    + owner_id                             = (known after apply)
    + tags                                 = {
        + "Name" = "my_vpc"
      }
    + tags_all                             = {
        + "Name" = "my_vpc"
      }
  }
```

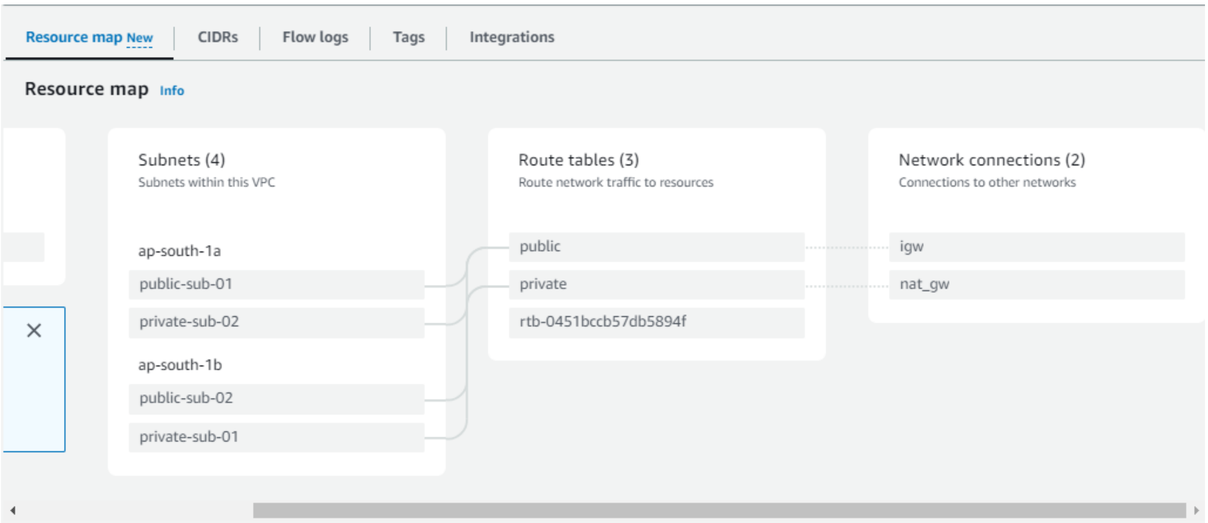


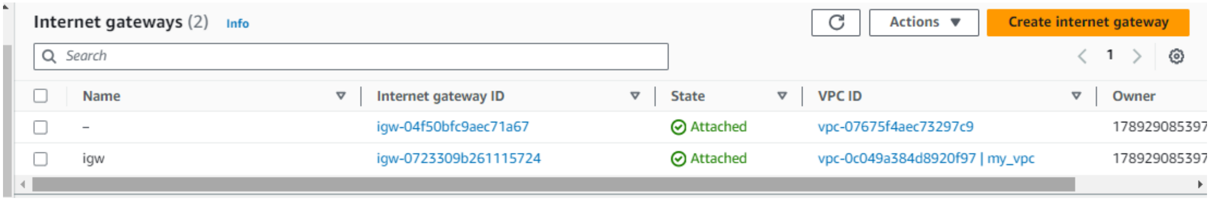
```
Plan: 14 to add, 0 to change, 0 to destroy.
aws_eip.eip: Creating...
aws_vpc.my_vpc: Creating...
aws_eip.eip: Creation complete after 1s [id=eipalloc-03d3f2e12942cdc7e]
aws_vpc.my_vpc: Creation complete after 2s [id=vpc-0c049a384d8920f97]
aws_subnet.private-sub-01: Creating...
aws_subnet.public-sub-01: Creating...
aws_subnet.public-sub-02: Creating...
aws_subnet.private-sub-02: Creating...
aws_internet_gateway.igw: Creating...
aws_subnet.private-sub-02: Creation complete after 0s [id=subnet-017760ef763d8b503]
aws_subnet.public-sub-01: Creation complete after 0s [id=subnet-0c000ca0ccbb2fa49]
aws_subnet.public-sub-02: Creation complete after 0s [id=subnet-07b6bf94c1d49856d]
aws_subnet.private-sub-01: Creation complete after 0s [id=subnet-0f39a9a7424836335]
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0723309b261115724]
aws_nat_gateway.nat: Creating...
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 0s [id=rtb-02c9ca304aabbde95]
aws_route_table_association.pub_subnet_02: Creating...
aws_route_table_association.pub_subnet_01: Creating...
aws_route_table_association.pub_subnet_02: Creation complete after 1s [id=rtbassoc-09675eb99f6cae000]
aws_route_table_association.pub_subnet_01: Creation complete after 1s [id=rtbassoc-0d434df65c4543e79]
aws_nat_gateway.nat: Still creating... [10s elapsed]
aws_nat_gateway.nat: Still creating... [20s elapsed]
aws_nat_gateway.nat: Still creating... [30s elapsed]
aws_nat_gateway.nat: Still creating... [40s elapsed]
aws_nat_gateway.nat: Still creating... [50s elapsed]
aws_nat_gateway.nat: Still creating... [1m0s elapsed]
aws_nat_gateway.nat: Still creating... [1m10s elapsed]
aws_nat_gateway.nat: Still creating... [1m20s elapsed]
aws_nat_gateway.nat: Still creating... [1m30s elapsed]
aws_nat_gateway.nat: Still creating... [1m40s elapsed]
aws_nat_gateway.nat: Creation complete after 1m44s [id=nat-0011849859a1d99e5]
aws_route_table.private: Creating...
aws_route_table.private: Creation complete after 1s [id=rtb-086fa73d41695d82b]
aws_route_table_association.pri_subnet_01: Creation complete after 0s [id=rtbassoc-0e7e63aadb3bfb41c]
aws_route_table_association.pri_subnet_02: Creation complete after 0s [id=rtbassoc-02e1a7a6219b1d832]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_03#
```



Your VPCs (2) Info

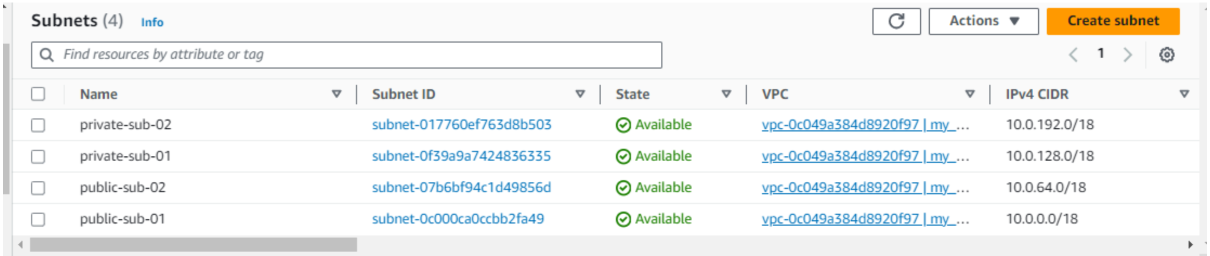
□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHC
□	my_vpc	vpc-0c049a384d8920f97	✓ Available	10.0.0.0/16	–	dopt
□	–	vpc-07675f4aec73297c9	✓ Available	172.31.0.0/16	–	dopt





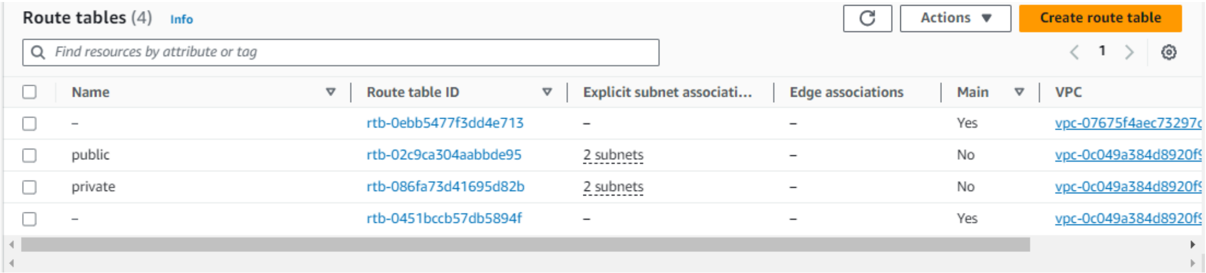
Internet gateways (2) Info

□	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-04f50bfc9aec71a67	✓ Attached	vpc-07675f4aec73297c9	178929085397
□	igw	igw-0723309b261115724	✓ Attached	vpc-0c049a384d8920f97 | my_vpc	178929085397



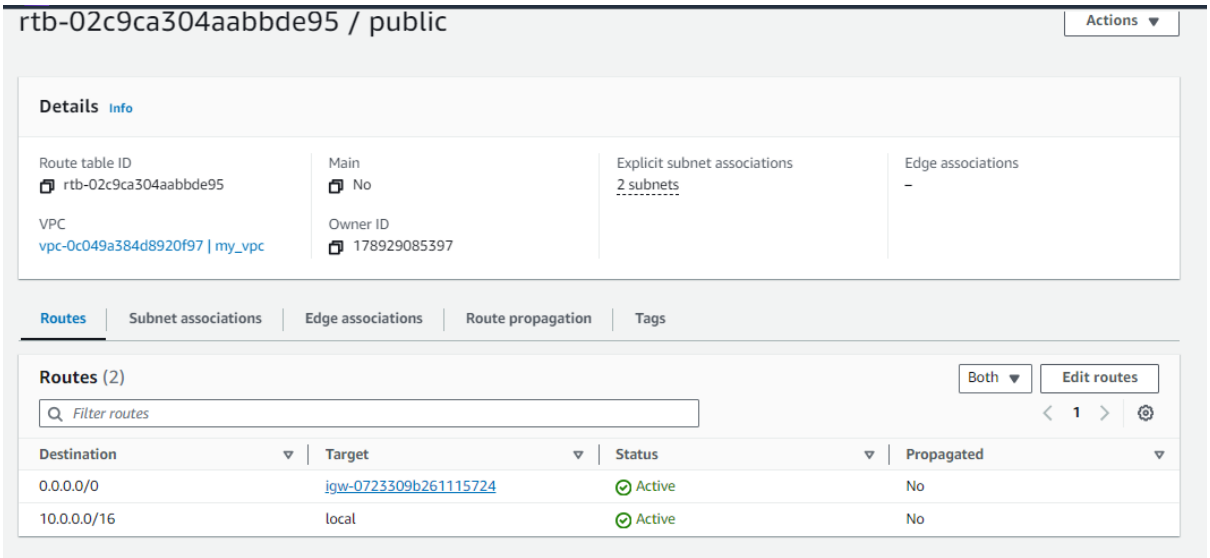
Subnets (4) Info

□	Name	Subnet ID	State	VPC	IPv4 CIDR
□	private-sub-02	subnet-017760ef763d8b503	✓ Available	vpc-0c049a384d8920f97 | my_...	10.0.192.0/18
□	private-sub-01	subnet-0f39a9a7424836335	✓ Available	vpc-0c049a384d8920f97 | my_...	10.0.128.0/18
□	public-sub-02	subnet-07b6bf94c1d49856d	✓ Available	vpc-0c049a384d8920f97 | my_...	10.0.64.0/18
□	public-sub-01	subnet-0c000ca0ccbb2fa49	✓ Available	vpc-0c049a384d8920f97 | my_...	10.0.0.0/18



Route tables (4) Info

□	Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC
□	–	rtb-0ebb5477f3dd4e713	–	–	Yes	vpc-07675f4aec73297c
□	public	rtb-02c9ca304aabbde95	2 subnets	–	No	vpc-0c049a384d8920f9
□	private	rtb-086fa73d41695d82b	2 subnets	–	No	vpc-0c049a384d8920f9
□	–	rtb-0451bccb57db5894f	–	–	Yes	vpc-0c049a384d8920f9



rtb-02c9ca304aabbde95 / public

Details Info

Route table ID
rtb-02c9ca304aabbde95

Main
No

Explicit subnet associations
2 subnets

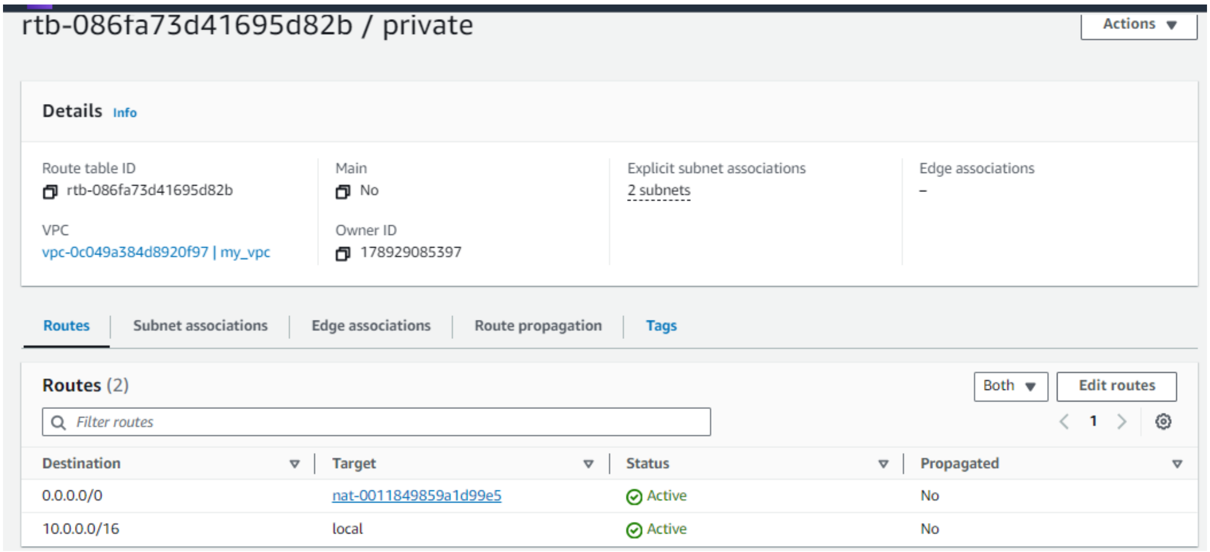
Edge associations
–

VPC
vpc-0c049a384d8920f97 | my_vpc

Owner ID
178929085397

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0723309b261115724	✓ Active	No
10.0.0.0/16	local	✓ Active	No



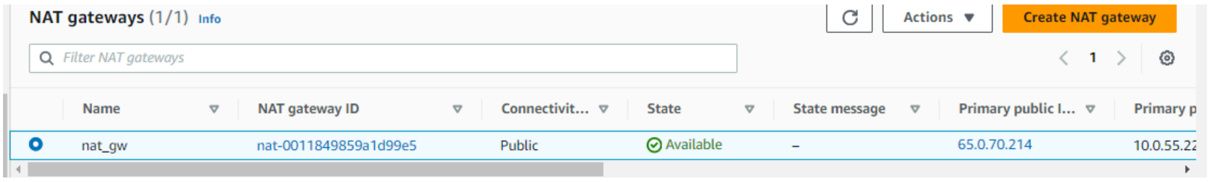
rtb-086fa73d41695d82b / private

Details

Route table ID rtb-086fa73d41695d82b	Main No	Explicit subnet associations 2 subnets	Edge associations –
VPC vpc-0c049a384d8920f97 | my_vpc	Owner ID 178929085397		

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0011849859a1d99e5	✓ Active	No
10.0.0.0/16	local	✓ Active	No



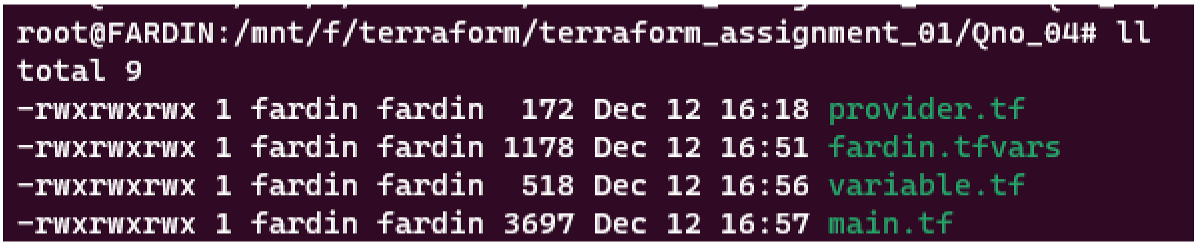
NAT gateways (1/1)

	Name	NAT gateway ID	Connectivit...	State	State message	Primary public I...	Primary p
●	nat_gw	nat-0011849859a1d99e5	Public	✓ Available	–	65.0.70.214	10.0.55.22

04.

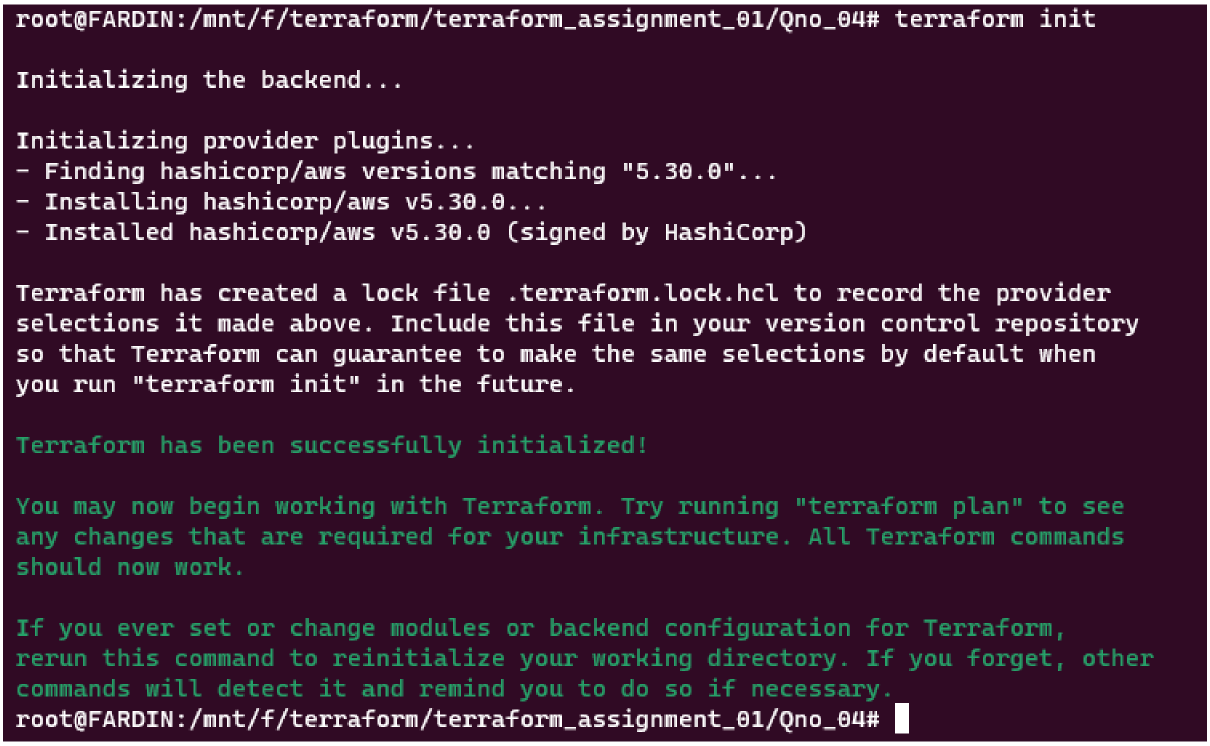
- Create EC2 instance one of the public Subnets of VPC that you have created
- Validate your Connection using SSH.

ANS:-



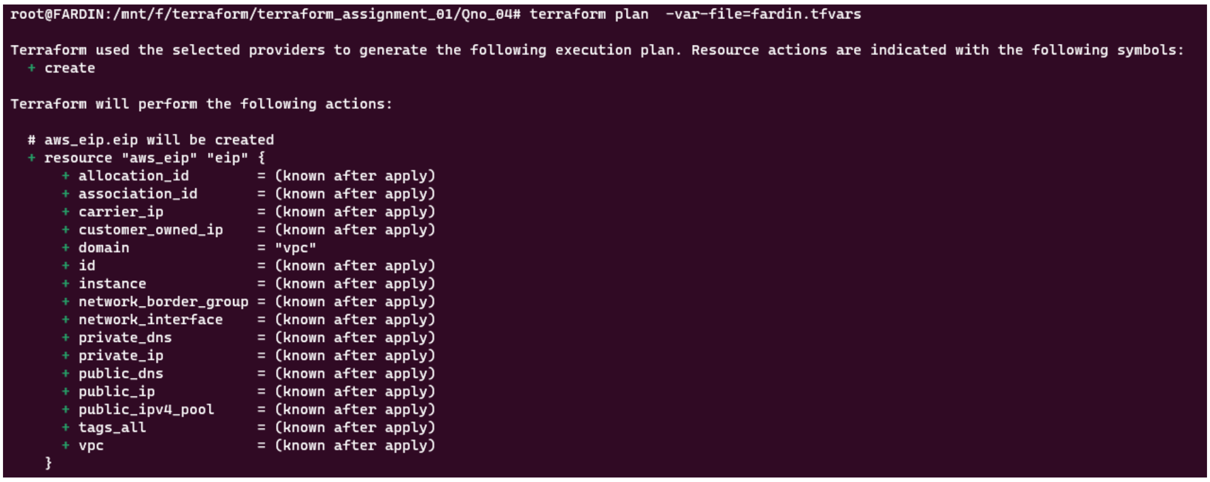
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04# ll
total 9
-rwxrwxrwx 1 fardin fardin  172 Dec 12 16:18 provider.tf
-rwxrwxrwx 1 fardin fardin 1178 Dec 12 16:51 fardin.tfvars
-rwxrwxrwx 1 fardin fardin  518 Dec 12 16:56 variable.tf
-rwxrwxrwx 1 fardin fardin 3697 Dec 12 16:57 main.tf
```

Terraform init

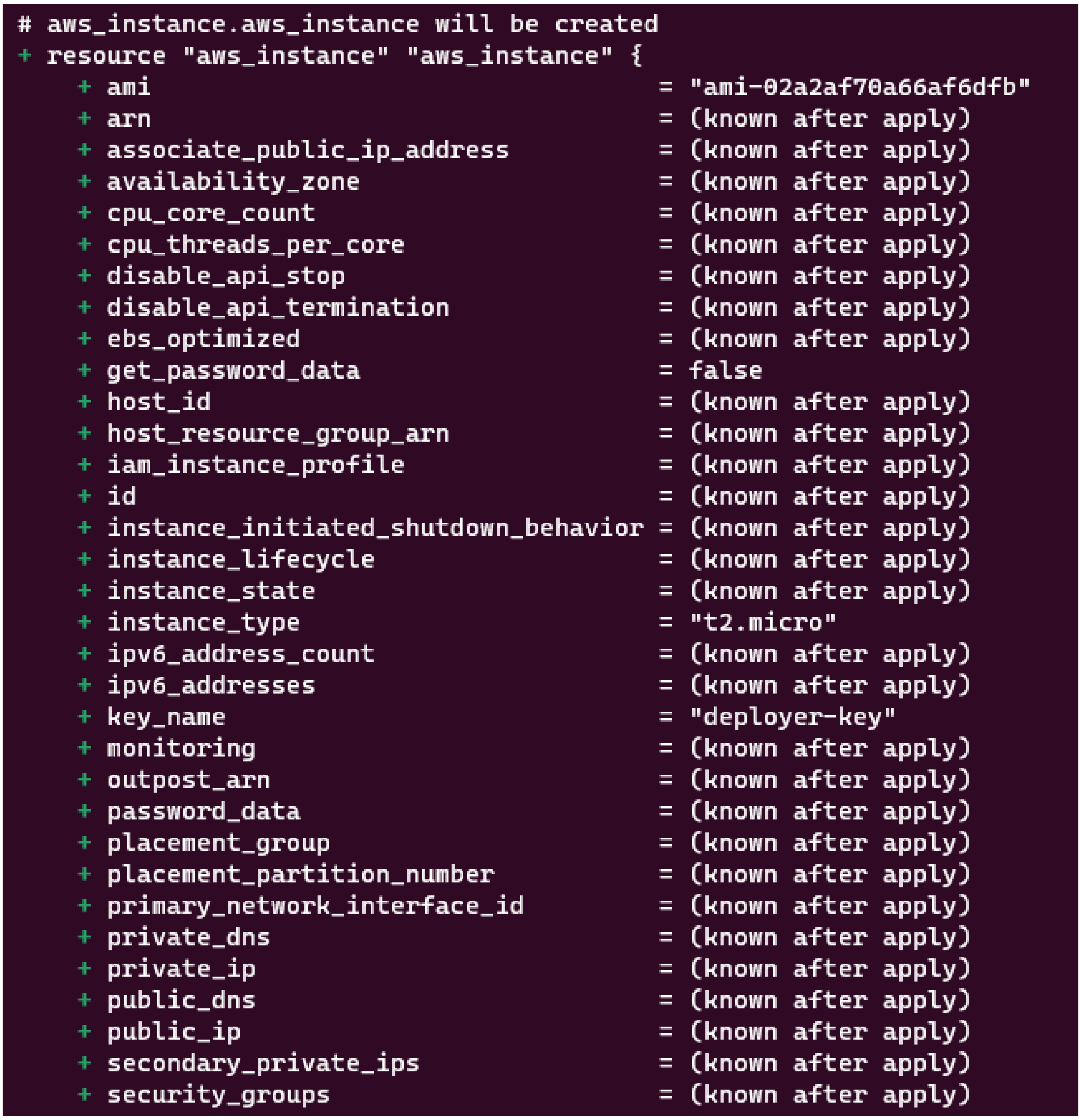


```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.30.0"...
- 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.
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04# 
```

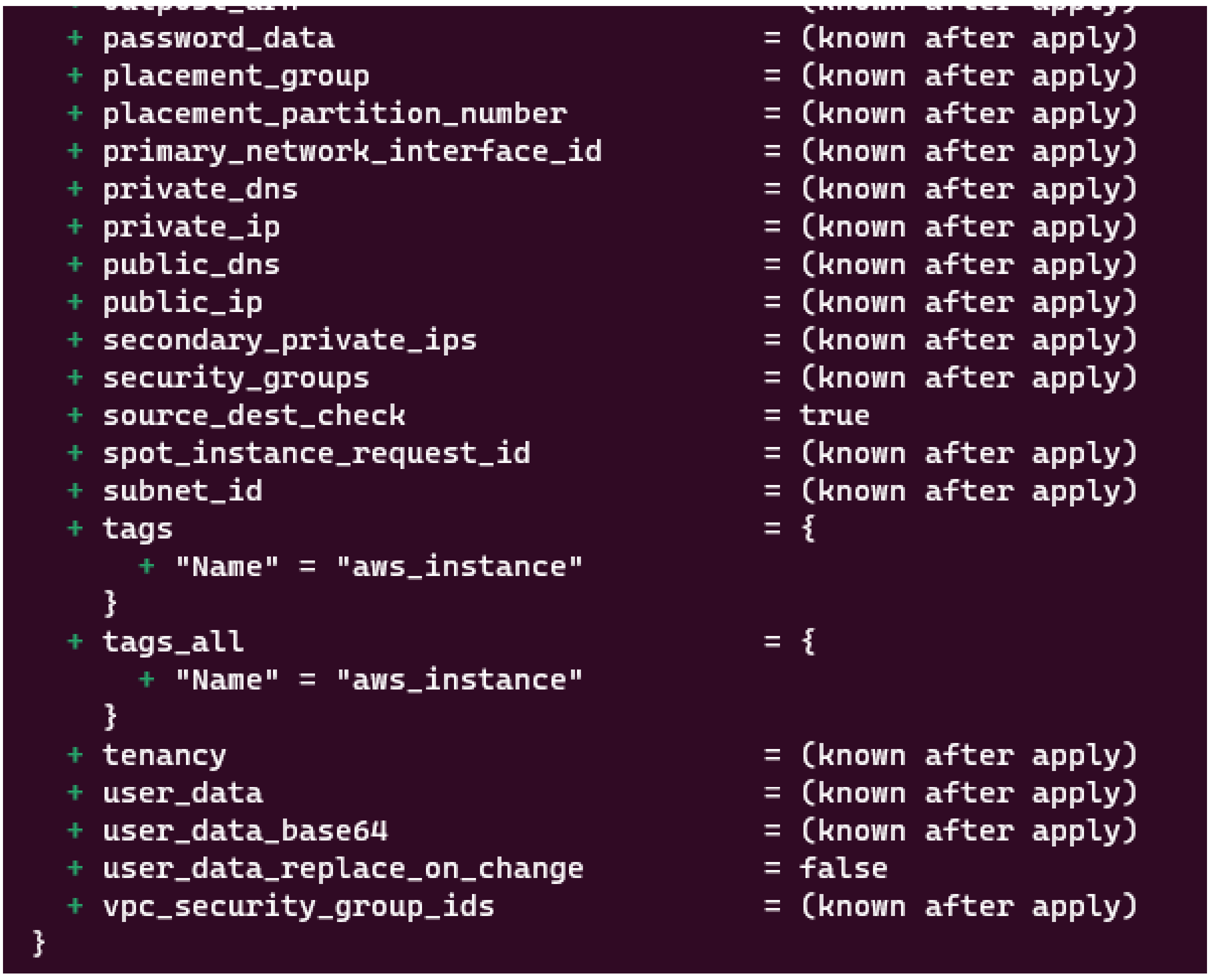
Terraform plan



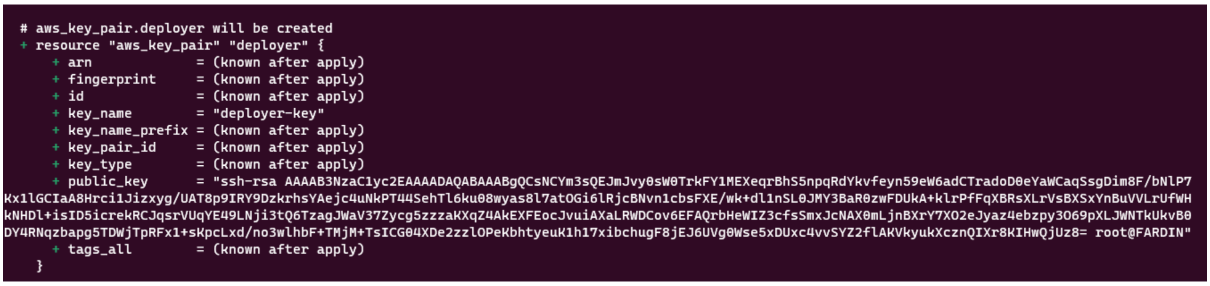
```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04# terraform plan  -var-file=fardin.tfvars

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_eip.eip will be created
  + resource "aws_eip" "eip" {
      + 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.aws_instance will be created
+ resource "aws_instance" "aws_instance" {
    + ami                                  = "ami-02a2af70a66af6dfb"
    + 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                             = "deployer-key"
    + 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)
```

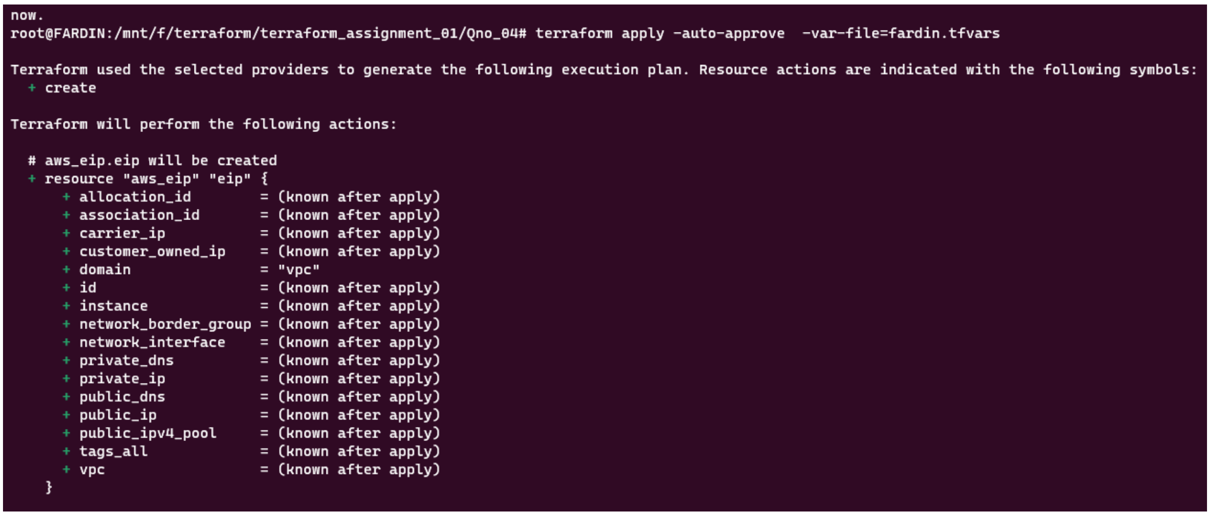


```
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (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" = "aws_instance"
        }
      + tags_all                             = {
          + "Name" = "aws_instance"
        }
      + 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)
    }
```

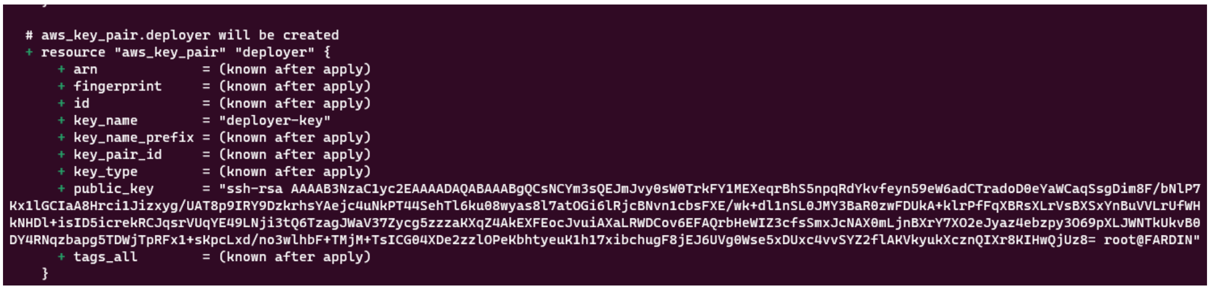


```
  # aws_key_pair.deployer will be created
  + resource "aws_key_pair" "deployer" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "deployer-key"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCsNCYm3sQEJmJvy0sW0TrkFY1MEXeqrBhS5npqRdYkvfeyn59eW6adCTradoD0eYaWCaqSsgDim8F/bNlP7
Kx1lGCIaA8Hrci1Jizxyg/UAT8p9IRY9DzkrhsYAejc4uNkPT44SehTl6ku08wyas8l7atOGi6lRjcBNvn1cbsFXE/wk+dl1nSL0JMY3BaR0zwFDUkA+klrPfFqXBRsXLrVsBXSxYnBuVVLrUfWH
kNHDl+isID5icrekRCJqsrVUqYE49LNji3tQ6TzagJWaV37Zycg5zzzaKXqZ4AkEXFEocJvuiAXaLRWDCov6EFAQrbHeWIZ3cfsSmxJcNAX0mLjnBXrY7XO2eJyaz4ebzpy3O69pXLJWNTkUkvB0
DY4RNqzbapg5TDWjTpRFx1+sKpcLxd/no3wlhbF+TMjM+TsICG04XDe2zzlOPeKbhtyeuK1h17xibchugF8jEJ6UVg0Wse5xDUxc4vvSYZ2flAKVkyukXcznQIXr8KIHwQjUz8= root@FARDIN"
      + tags_all        = (known after apply)
    }
```

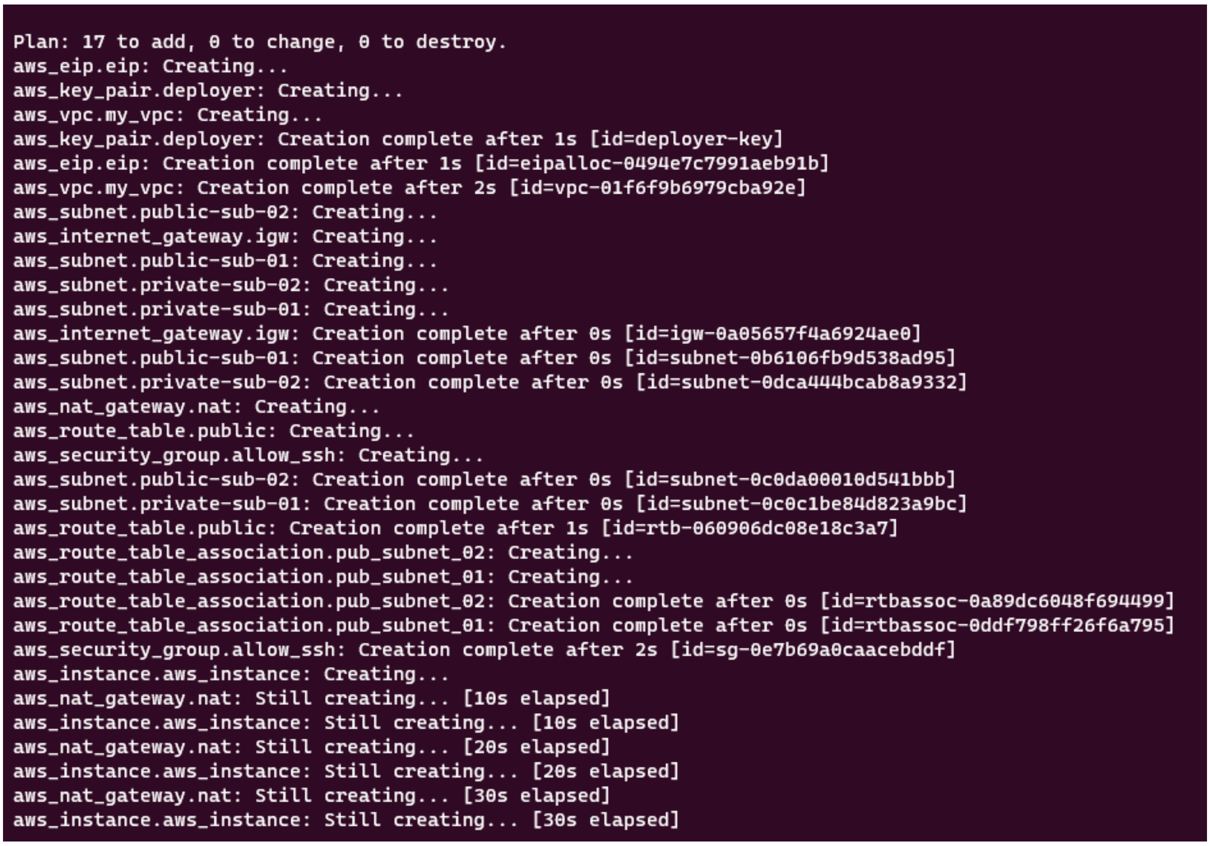
Terraform apply



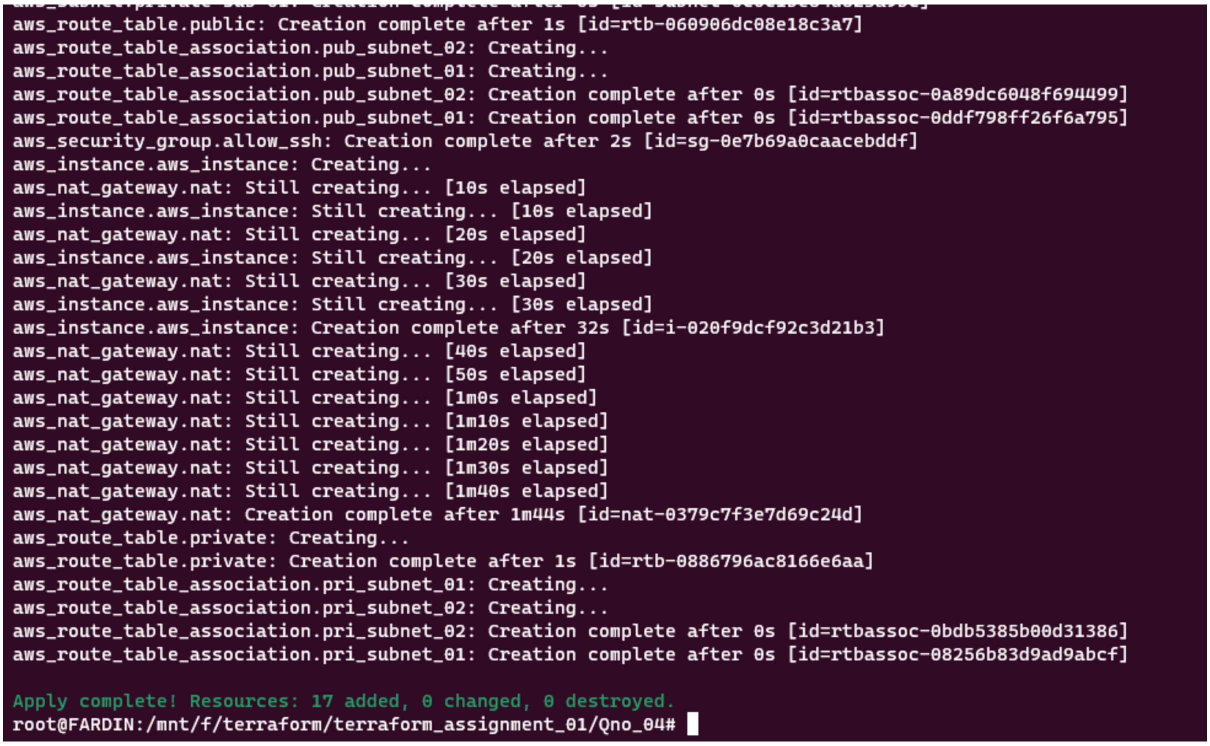
```
now.
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04# terraform apply -auto-approve  -var-file=fardin.tfvars

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_eip.eip will be created
  + resource "aws_eip" "eip" {
      + 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)
    }
```



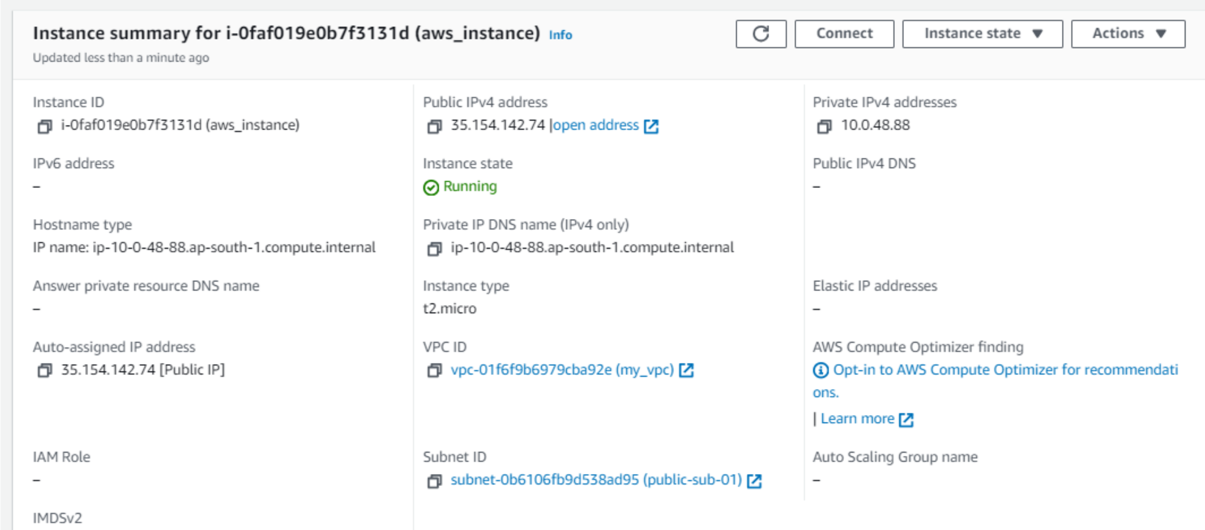
```
  # aws_key_pair.deployer will be created
  + resource "aws_key_pair" "deployer" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "deployer-key"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCsNCYm3sQEJmJvy0sW0TrkFY1MEXeqrBhS5npqRdYkvfeyn59eW6adCTradoD0eYaWCaqSsgDim8F/bNlP7
Kx1lGCIaA8Hrci1Jizxyg/UAT8p9IRY9DzkrhsYAejc4uNkPT44SehTl6ku08wyas8l7atOGi6lRjcBNvn1cbsFXE/wk+dl1nSL0JMY3BaR0zwFDUkA+klrPfFqXBRsXLrVsBXSxYnBuVVLrUfWH
kNHDl+isID5icrekRCJqsrVUqYE49LNji3tQ6TzagJWaV37Zycg5zzzaKXqZ4AkEXFEocJvuiAXaLRWDCov6EFAQrbHeWIZ3cfsSmxJcNAX0mLjnBXrY7XO2eJyaz4ebzpy3O69pXLJWNTkUkvB0
DY4RNqzbapg5TDWjTpRFx1+sKpcLxd/no3wlhbF+TMjM+TsICG04XDe2zzlOPeKbhtyeuK1h17xibchugF8jEJ6UVg0Wse5xDUxc4vvSYZ2flAKVkyukXcznQIXr8KIHwQjUz8= root@FARDIN"
      + tags_all        = (known after apply)
    }
```



```
Plan: 17 to add, 0 to change, 0 to destroy.
aws_eip.eip: Creating...
aws_key_pair.deployer: Creating...
aws_vpc.my_vpc: Creating...
aws_key_pair.deployer: Creation complete after 1s [id=deployer-key]
aws_eip.eip: Creation complete after 1s [id=eipalloc-0494e7c7991aeb91b]
aws_vpc.my_vpc: Creation complete after 2s [id=vpc-01f6f9b6979cba92e]
aws_subnet.public-sub-02: Creating...
aws_internet_gateway.igw: Creating...
aws_subnet.public-sub-01: Creating...
aws_subnet.private-sub-02: Creating...
aws_subnet.private-sub-01: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0a05657f4a6924ae0]
aws_subnet.public-sub-01: Creation complete after 0s [id=subnet-0b6106fb9d538ad95]
aws_subnet.private-sub-02: Creation complete after 0s [id=subnet-0dca444bcab8a9332]
aws_nat_gateway.nat: Creating...
aws_route_table.public: Creating...
aws_security_group.allow_ssh: Creating...
aws_subnet.public-sub-02: Creation complete after 0s [id=subnet-0c0da00010d541bbb]
aws_subnet.private-sub-01: Creation complete after 0s [id=subnet-0c0c1be84d823a9bc]
aws_route_table.public: Creation complete after 1s [id=rtb-060906dc08e18c3a7]
aws_route_table_association.pub_subnet_02: Creating...
aws_route_table_association.pub_subnet_01: Creating...
aws_route_table_association.pub_subnet_02: Creation complete after 0s [id=rtbassoc-0a89dc6048f694499]
aws_route_table_association.pub_subnet_01: Creation complete after 0s [id=rtbassoc-0ddf798ff26f6a795]
aws_security_group.allow_ssh: Creation complete after 2s [id=sg-0e7b69a0caacebddf]
aws_instance.aws_instance: Creating...
aws_nat_gateway.nat: Still creating... [10s elapsed]
aws_instance.aws_instance: Still creating... [10s elapsed]
aws_nat_gateway.nat: Still creating... [20s elapsed]
aws_instance.aws_instance: Still creating... [20s elapsed]
aws_nat_gateway.nat: Still creating... [30s elapsed]
aws_instance.aws_instance: Still creating... [30s elapsed]
```



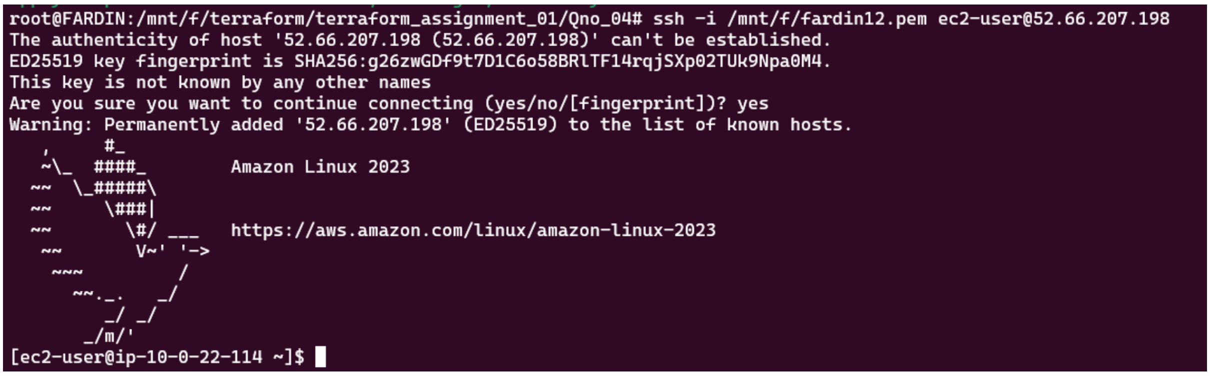
```
aws_route_table.public: Creation complete after 1s [id=rtb-060906dc08e18c3a7]
aws_route_table_association.pub_subnet_02: Creating...
aws_route_table_association.pub_subnet_01: Creating...
aws_route_table_association.pub_subnet_02: Creation complete after 0s [id=rtbassoc-0a89dc6048f694499]
aws_route_table_association.pub_subnet_01: Creation complete after 0s [id=rtbassoc-0ddf798ff26f6a795]
aws_security_group.allow_ssh: Creation complete after 2s [id=sg-0e7b69a0caacebddf]
aws_instance.aws_instance: Creating...
aws_nat_gateway.nat: Still creating... [10s elapsed]
aws_instance.aws_instance: Still creating... [10s elapsed]
aws_nat_gateway.nat: Still creating... [20s elapsed]
aws_instance.aws_instance: Still creating... [20s elapsed]
aws_nat_gateway.nat: Still creating... [30s elapsed]
aws_instance.aws_instance: Still creating... [30s elapsed]
aws_instance.aws_instance: Creation complete after 32s [id=i-020f9dcf92c3d21b3]
aws_nat_gateway.nat: Still creating... [40s elapsed]
aws_nat_gateway.nat: Still creating... [50s elapsed]
aws_nat_gateway.nat: Still creating... [1m0s elapsed]
aws_nat_gateway.nat: Still creating... [1m10s elapsed]
aws_nat_gateway.nat: Still creating... [1m20s elapsed]
aws_nat_gateway.nat: Still creating... [1m30s elapsed]
aws_nat_gateway.nat: Still creating... [1m40s elapsed]
aws_nat_gateway.nat: Creation complete after 1m44s [id=nat-0379c7f3e7d69c24d]
aws_route_table.private: Creating...
aws_route_table.private: Creation complete after 1s [id=rtb-0886796ac8166e6aa]
aws_route_table_association.pri_subnet_01: Creating...
aws_route_table_association.pri_subnet_02: Creating...
aws_route_table_association.pri_subnet_02: Creation complete after 0s [id=rtbassoc-0bdb5385b00d31386]
aws_route_table_association.pri_subnet_01: Creation complete after 0s [id=rtbassoc-08256b83d9ad9abcf]

Apply complete! Resources: 17 added, 0 changed, 0 destroyed.
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04#
```



Instance summary for i-0faf019e0b7f3131d (aws_instance) Info

Updated less than a minute ago

Instance ID i-0faf019e0b7f3131d (aws_instance)	Public IPv4 address 35.154.142.74 |open address	Private IPv4 addresses 10.0.48.88
IPv6 address –	Instance state Running	Public IPv4 DNS –
Hostname type IP name: ip-10-0-48-88.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-48-88.ap-south-1.compute.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 35.154.142.74 [Public IP]	VPC ID vpc-01f6f9b6979cba92e (my_vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0b6106fb9d538ad95 (public-sub-01)	Auto Scaling Group name –
IMDSv2		



```
root@FARDIN:/mnt/f/terraform/terraform_assignment_01/Qno_04# ssh -i /mnt/f/fardin12.pem ec2-user@52.66.207.198
The authenticity of host '52.66.207.198 (52.66.207.198)' can't be established.
ED25519 key fingerprint is SHA256:g26zwGDf9t7D1C6o58BRlTF14rqjSXp02TUk9Npa0M4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.66.207.198' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-22-114 ~]$
```