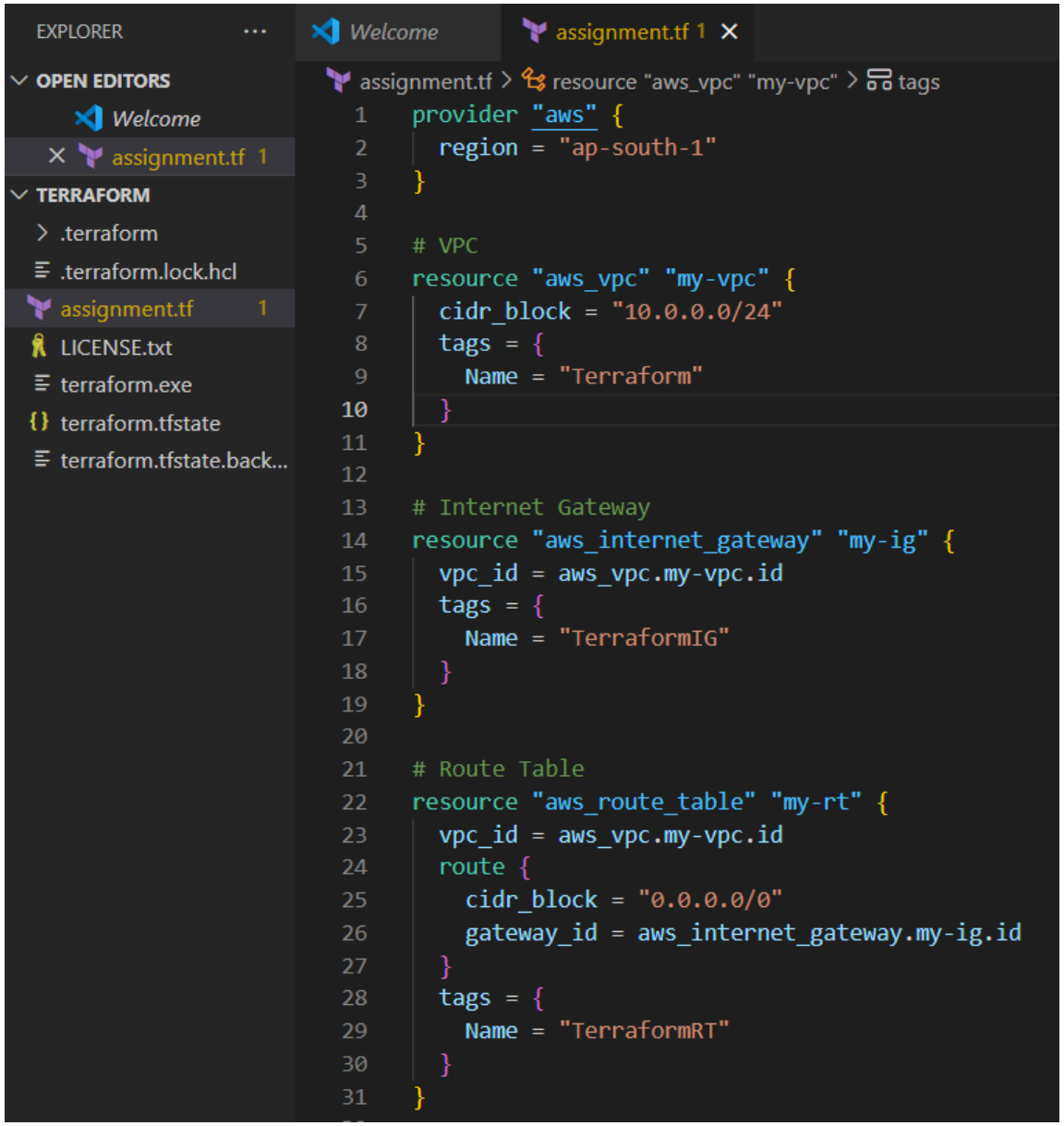
L1

1



2

```
# Subnet
resource "aws_subnet" "my-sn" {
  vpc_id                  = aws_vpc.my-vpc.id
  map_public_ip_on_launch = false
  cidr_block              = "10.0.0.0/25"
  tags = {
    Name = "TerraformSN"
  }
}

# Security Group ports (22, 80, 443)
resource "aws_security_group" "my-sg" {
  name   = "my-securitygroup"
  vpc_id = aws_vpc.my-vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

3

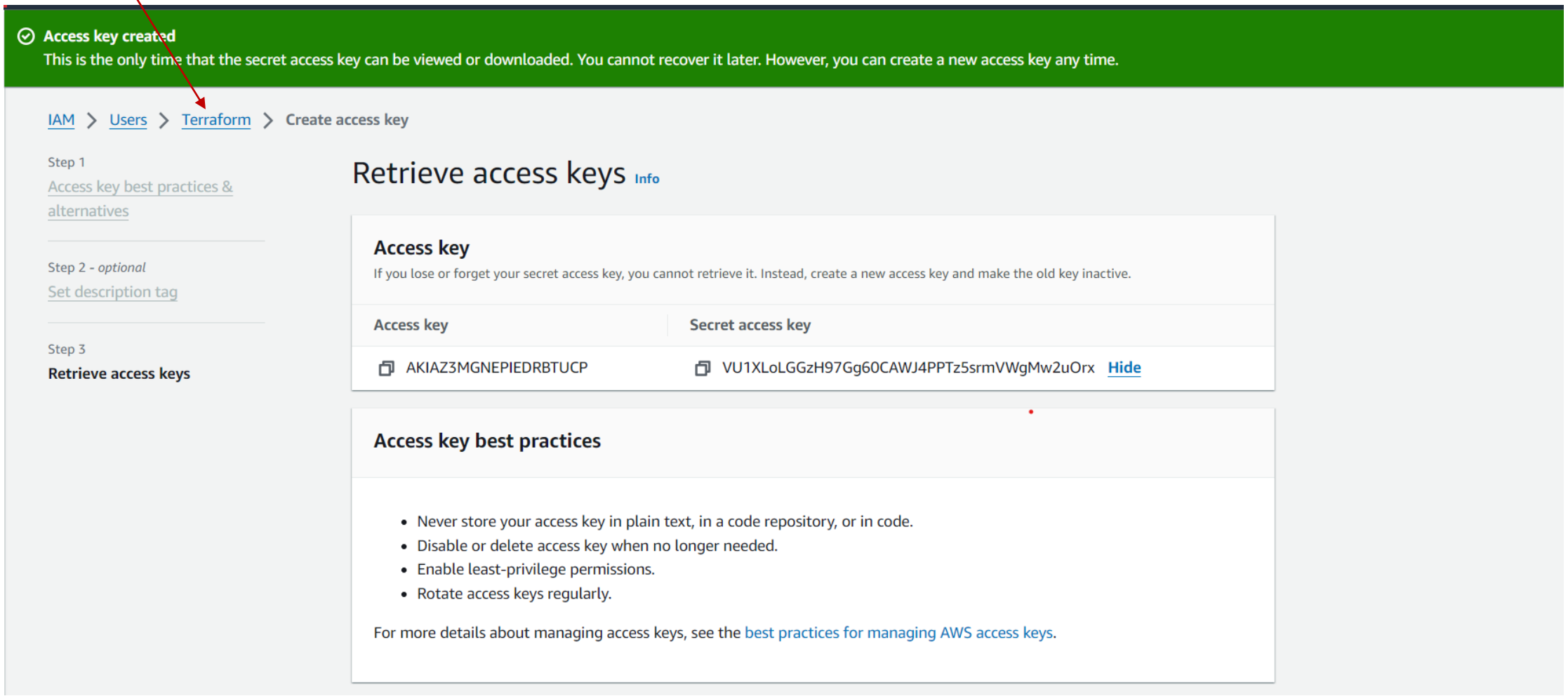
```
# Network Interface setup/creation
resource "aws_network_interface" "my-ni" {
  subnet_id       = aws_subnet.my-sn.id
  security_groups = [aws_security_group.my-sg.id]
  tags = {
    Name = "terraformSN"
  }
}

# Elastic IP
resource "aws_eip" "eip-ni" {
  vpc = true
  tags = {
    Name = "My-EIP"
  }
}

# Elastic IP Association
resource "aws_eip_association" "eip-association" {
  allocation_id       = aws_eip.eip-ni.allocation_id
  network_interface_id = aws_network_interface.my-ni.id
}
```

4

```
# EC2 instance setup
resource "aws_instance" "demo" {
  ami           = "ami-08ee1453725d19cdb"
  instance_type = "t2.micro"
  key_name      = "amazon-linux"

  user_data = <<-EOF
    #!/bin/bash
    sudo yum install git -y
    EOF

  tags = {
    Name = "terraformDemoInstance"
  }

  network_interface {
    device_index          = 0
    network_interface_id = aws_network_interface.my-ni.id
  }
}
```

Configure the Creds

Using user Access Key



Aws configure

```
PS O:\Terraform> aws configure
AWS Access Key ID [****************TUCP]: AKIAZ3MGNEPIC2MDTWHI
AWS Secret Access Key [****************uOrx]: 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
Default region name [us-east-1]: ap-south-1
Default output format [json]: json
```

```
PS O:\Terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.64.0...
- Installed hashicorp/aws v5.64.0 (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 InIt

Terraform validate

```
PS O:\Terraform> terraform validate

Warning: Argument is deprecated

  with aws_eip.eip-ni,
  on assignment.tf line 71, in resource "aws_eip" "eip-ni":
  71:   vpc = true

use domain attribute instead

Success! The configuration is valid, but there were some validation warnings as shown above.
```

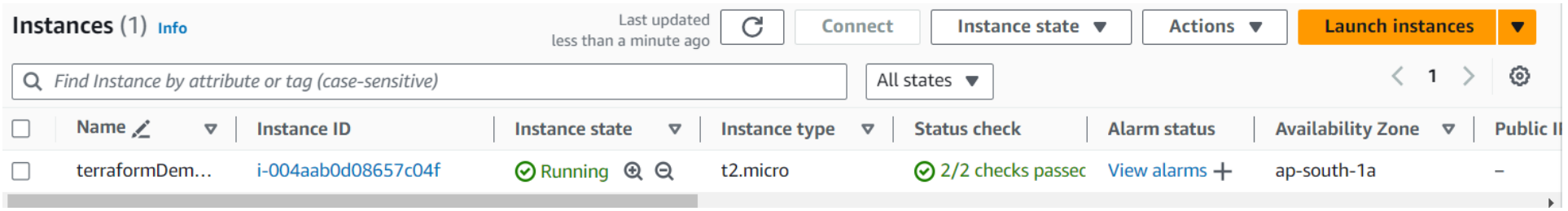
Terraform Plan

```
PS O:\Terraform> terraform plan

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

Terraform will perform the following actions:

  # aws_eip.eip-ni will be created
  + resource "aws_eip" "eip-ni" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = (known after apply)
      + 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)
      + ptr_record           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags                 = {
          + "Name" = "My-EIP"
        }
      + tags_all             = {
          + "Name" = "My-EIP"
        }
      + vpc                  = true
    }

  # aws_eip_association.eip-association will be created
  + resource "aws_eip_association" "eip-association" {
      + 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.demo will be created
  + resource "aws_instance" "demo" {
      + ami                                  = "ami-0e86e20dae9224db8"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = "ap-south-1b"
      + 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                             = "ubuntu"
      + 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" = "terraformDemoInstance"
        }
      + tags_all                             = {
          + "Name" = "terraformDemoInstance"
        }
      + tenancy                              = (known after apply)
      + user_data                            = "0d6cf0609ba7a0357bbf4405cf05041302b1a7f2"
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

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

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

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

  # aws_network_interface.my-ni will be created
  + resource "aws_network_interface" "my-ni" {
      + 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)
                                    (known after apply)
                                    (known after apply)
                        _enabled  = false
                                    (known after apply)
      + owner_id                  = (known after apply)
      + private_dns_name          = (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
      + subnet_id                 = (known after apply)
      + tags                      = {
          + "Name" = "terraformSN"
        }
      + tags_all                  = {
          + "Name" = "terraformSN"
        }

      + attachment (known after apply)
    }

  # aws_route_table.my-rt will be created
  + resource "aws_route_table" "my-rt" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = [
          + {
              + cidr_block                = "0.0.0.0/0"
              + gateway_id                = "aws_internet_gateway.my-ig.id"
                # (11 unchanged attributes hidden)
            },
        ]
      + tags             = {
          + "Name" = "TerraformRT"
        }
      + tags_all         = {
          + "Name" = "TerraformRT"
        }
      + vpc_id           = (known after apply)
    }

  # aws_security_group.my-sg will be created
  + resource "aws_security_group" "my-sg" {
      + arn                    = (known after apply)
      + description            = "Managed by Terraform"
      + egress                 = (known after apply)
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
                # (1 unchanged attribute hidden)
            },
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 443
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 443
                # (1 unchanged attribute hidden)
            },
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "my-securitygroup"
      + 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_subnet.my-sn will be created
  + resource "aws_subnet" "my-sn" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = (known after apply)
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.0.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_customer_owned_ip_on_launch                = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.my-vpc will be created
  + resource "aws_vpc" "my-vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/24"
      + 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" = "Terraform"
        }
      + tags_all                             = {
          + "Name" = "Terraform"
        }
    }

Plan: 9 to add, 0 to change, 0 to destroy.

Warning: Argument is deprecated

  with aws_eip.eip-ni,
  on assignment.tf line 71, in resource "aws_eip" "eip-ni":
  71:   vpc = true

use domain attribute instead

(and one more similar warning elsewhere)
```

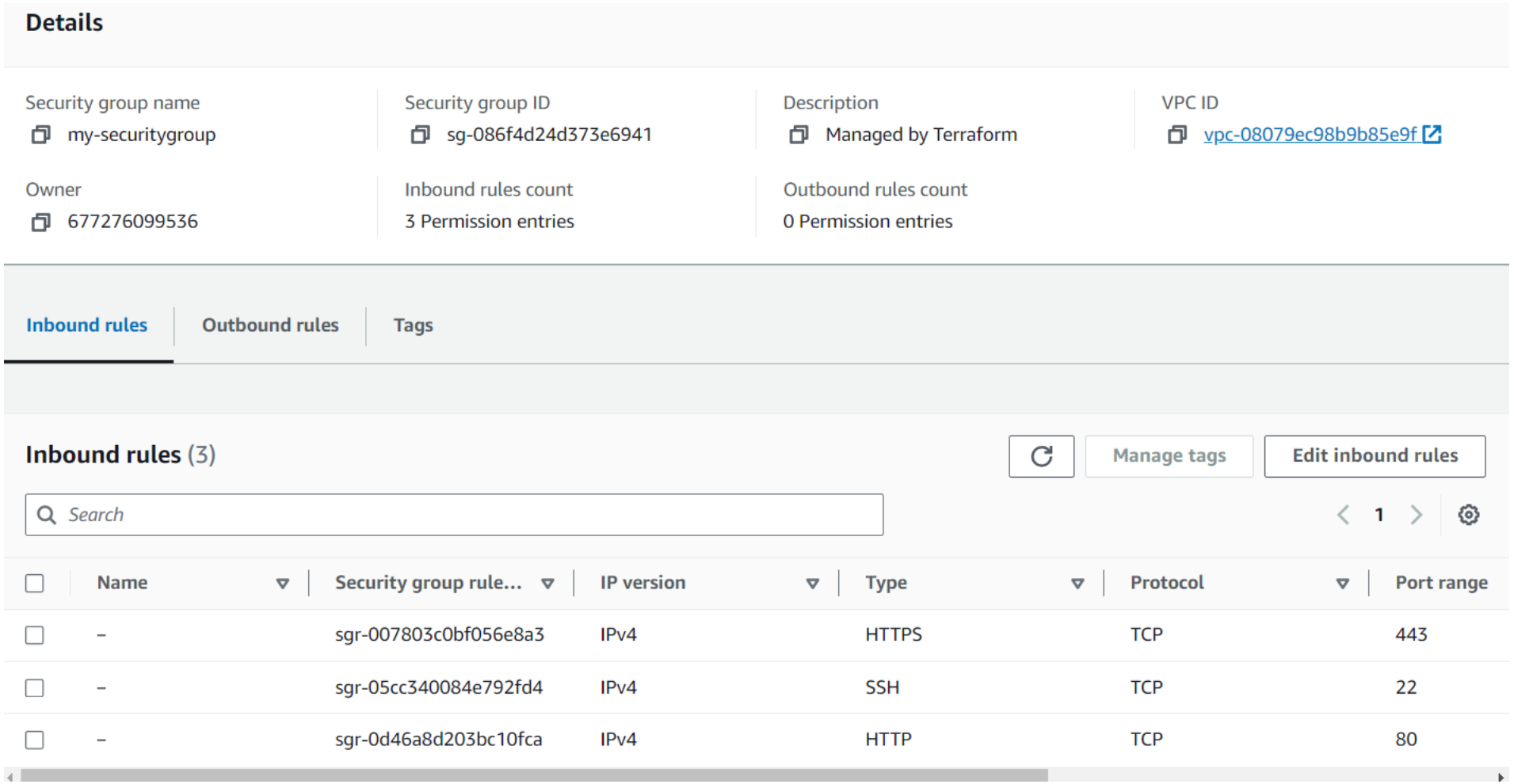
Terraform Apply

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.demo: Creating...
aws_instance.demo: Still creating... [10s elapsed]
aws_instance.demo: Still creating... [20s elapsed]
aws_instance.demo: Still creating... [30s elapsed]
aws_instance.demo: Still creating... [40s elapsed]
aws_instance.demo: Creation complete after 47s [id=i-004aab0d08657c04f]

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

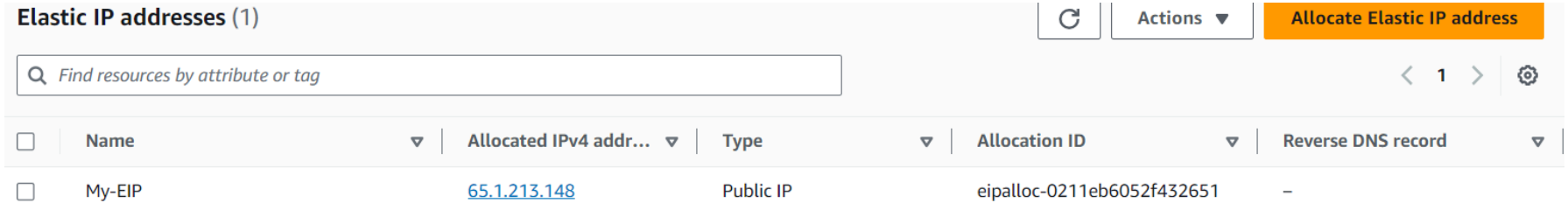
EC2



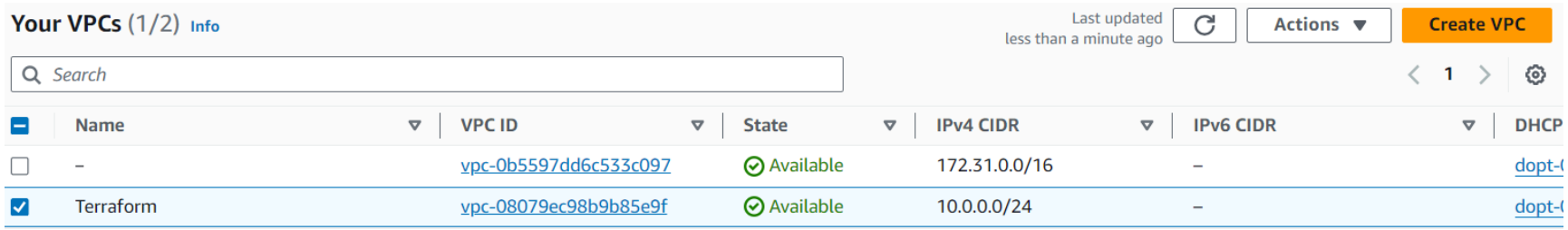
Security Group



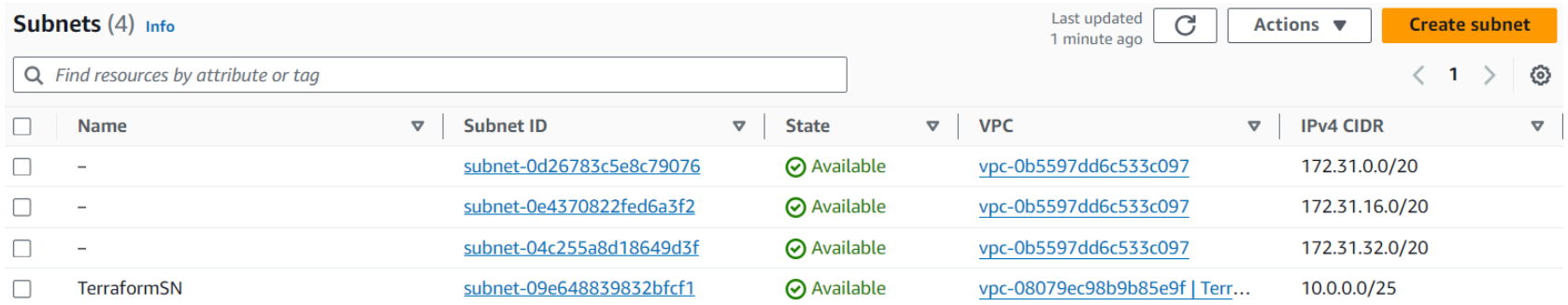
Elastic IP



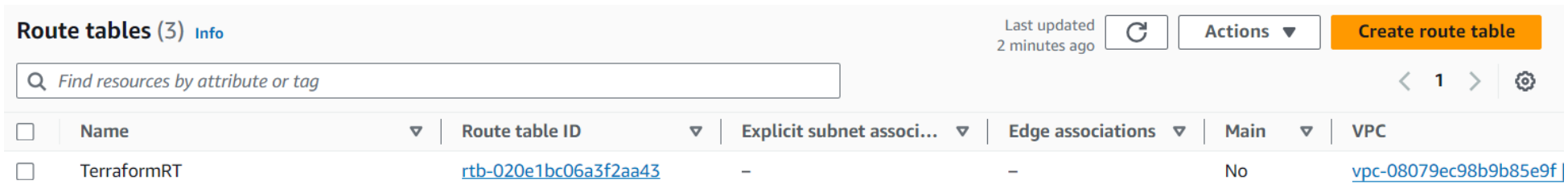
Vpc



Subnets



Route Tables



Internet Gateway

Internet gateways (2) Info

Actions ▼ | Create internet gateway

Search

1

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-045b8a48864ae9016	Attached	vpc-0b5597dd6c533c097	677276099536
TerraformIG	igw-08ab05e40a2a5e62c	Attached	vpc-08079ec98b9b85e9f | Terraform	677276099536

Terraform destroy

```
aws_eip_association.eip-association: Destruction complete after 4s
aws_eip.eip-ni: Destroying... [id=eipalloc-0211eb6052f432651]
aws_route_table.my-rt: Destruction complete after 6s
aws_internet_gateway.my-ig: Destroying... [id=igw-08ab05e40a2a5e62c]
aws_eip.eip-ni: Destruction complete after 4s
aws_instance.demo: Still destroying... [id=i-004aab0d08657c04f, 10s elapsed]
aws_internet_gateway.my-ig: Destruction complete after 7s
aws_instance.demo: Still destroying... [id=i-004aab0d08657c04f, 20s elapsed]
aws_instance.demo: Still destroying... [id=i-004aab0d08657c04f, 30s elapsed]
aws_instance.demo: Destruction complete after 37s
aws_network_interface.my-ni: Destroying... [id=eni-0db17d303c32b05b5]
aws_network_interface.my-ni: Destruction complete after 1s
aws_subnet.my-sn: Destroying... [id=subnet-09e648839832bfcf1]
aws_security_group.my-sg: Destroying... [id=sg-086f4d24d373e6941]
aws_security_group.my-sg: Destruction complete after 6s
aws_subnet.my-sn: Still destroying... [id=subnet-09e648839832bfcf1, 10s elapsed]
aws_subnet.my-sn: Destruction complete after 12s
aws_vpc.my-vpc: Destroying... [id=vpc-08079ec98b9b85e9f]
aws_vpc.my-vpc: Destruction complete after 2s

Destroy complete! Resources: 9 destroyed.
```