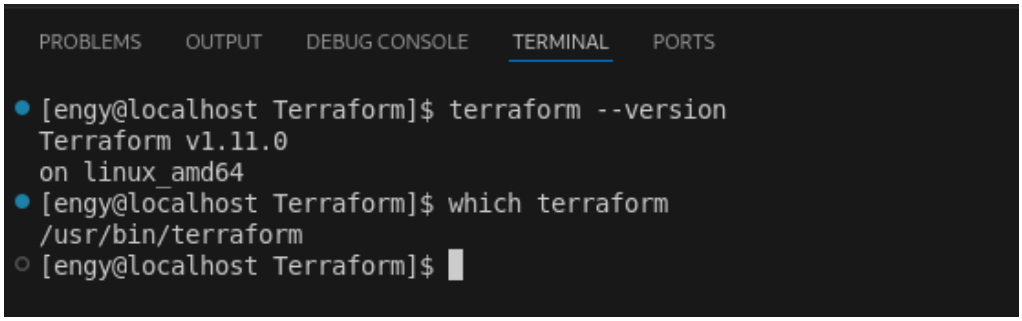
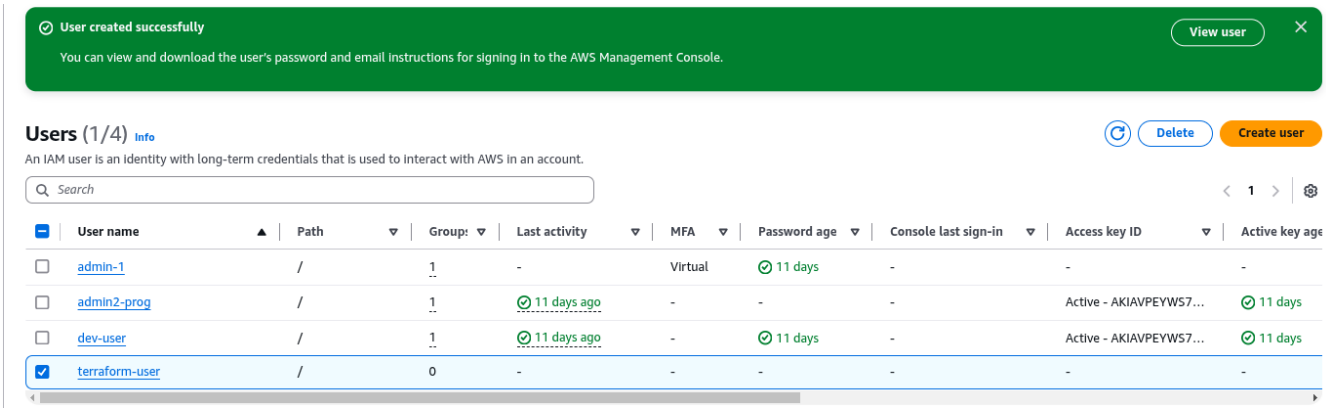
Lab 1

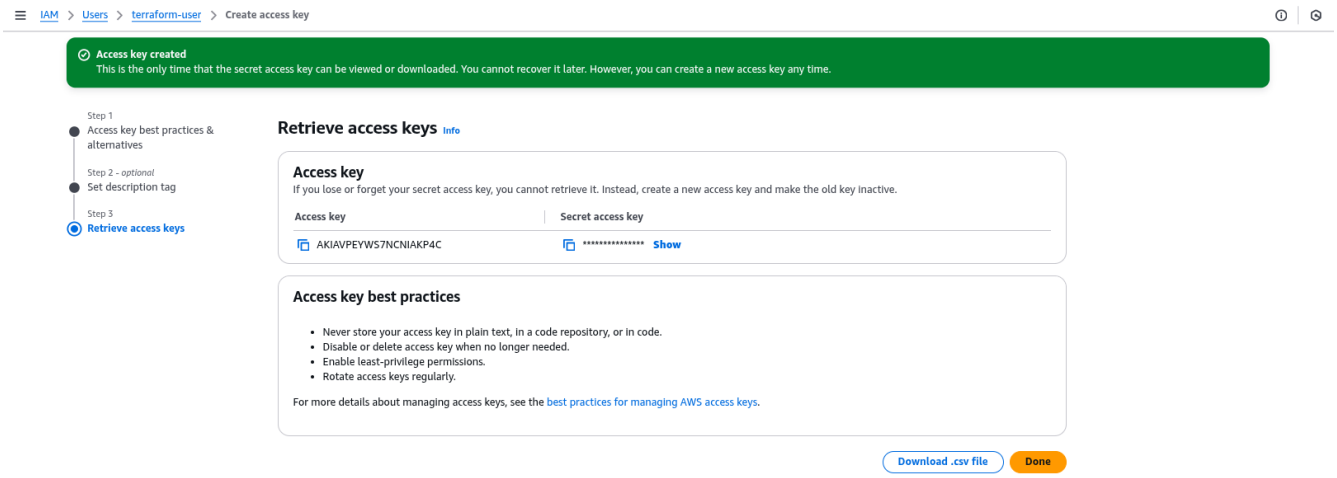
- Terraform installed successfully



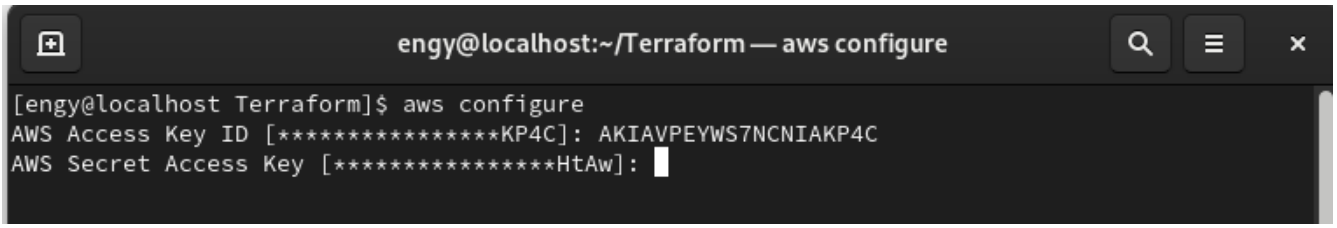
- Created a user name terraform-user in AWS Console and gave it Administrator access



- Generate the access keys for this user



- AWS Configure

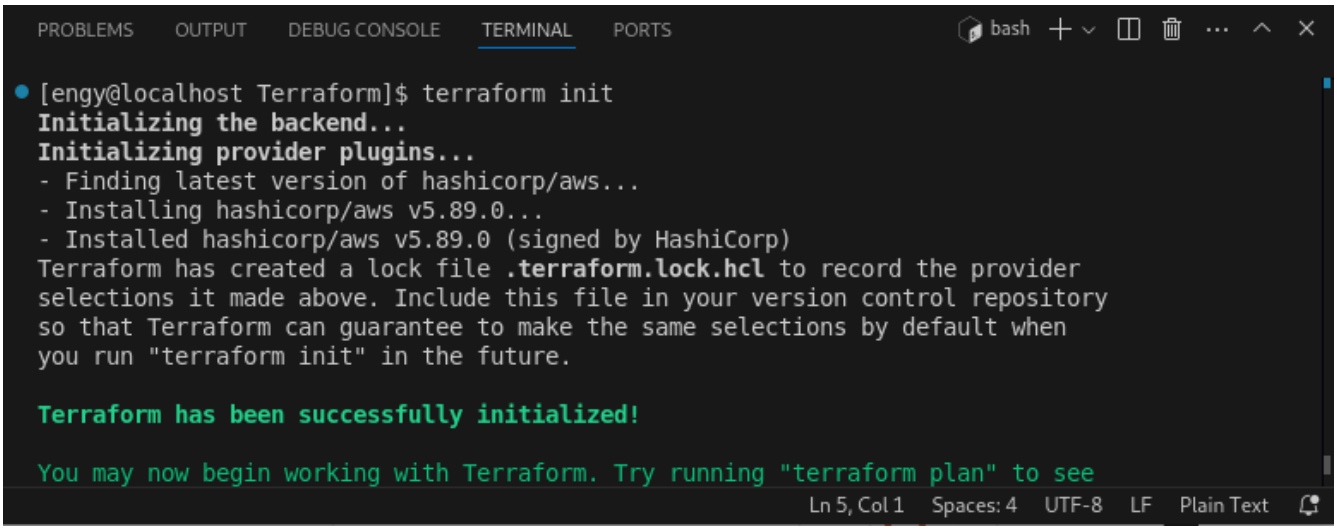


- Made a profile named terraform in my ~/.aws/credentials file

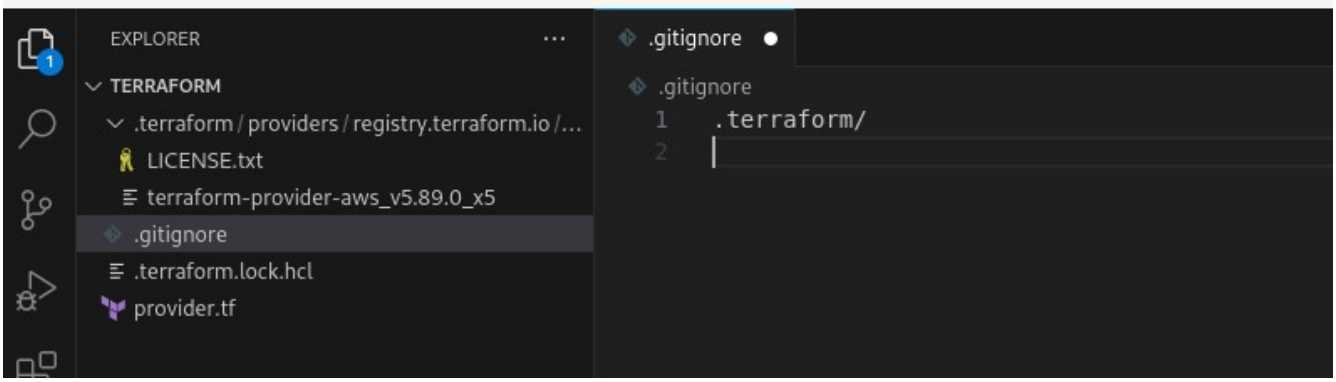
```
[engy@localhost Terraform]$ cd ~/.aws
[engy@localhost .aws]$ ls
config  credentials
[engy@localhost .aws]$ vi credentials
```

```
[terraform]
aws_access_key_id = AKIAV
aws_secret_access_key = F
```

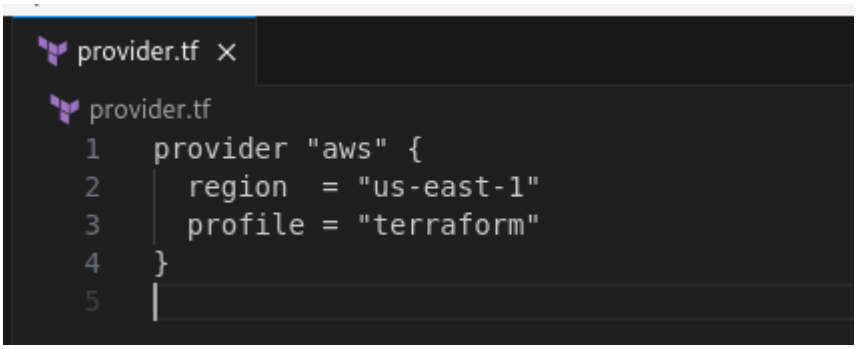
- Used terraform init to initialize my terraform plugins



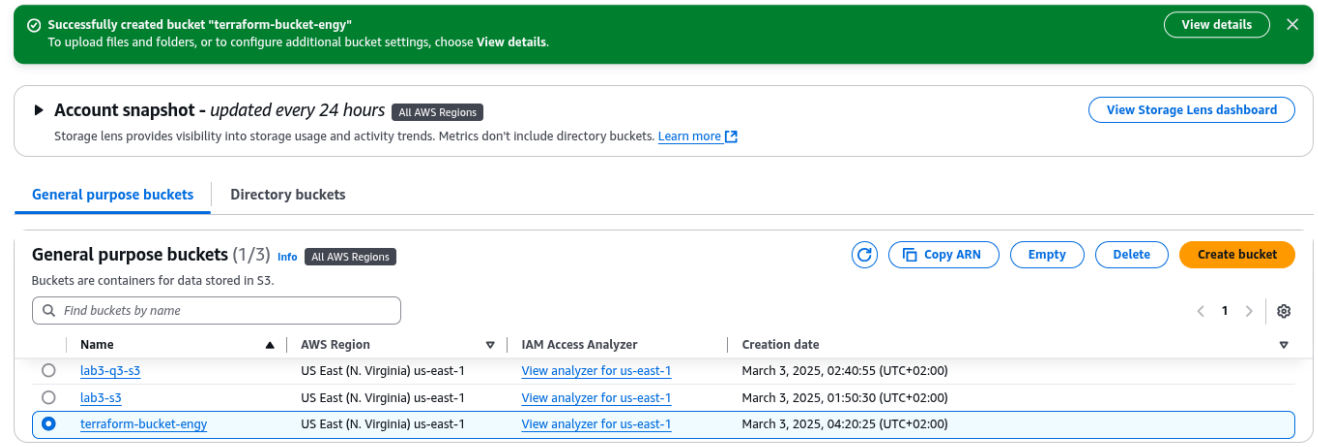
- Created gitignore file and added in it all the plugins



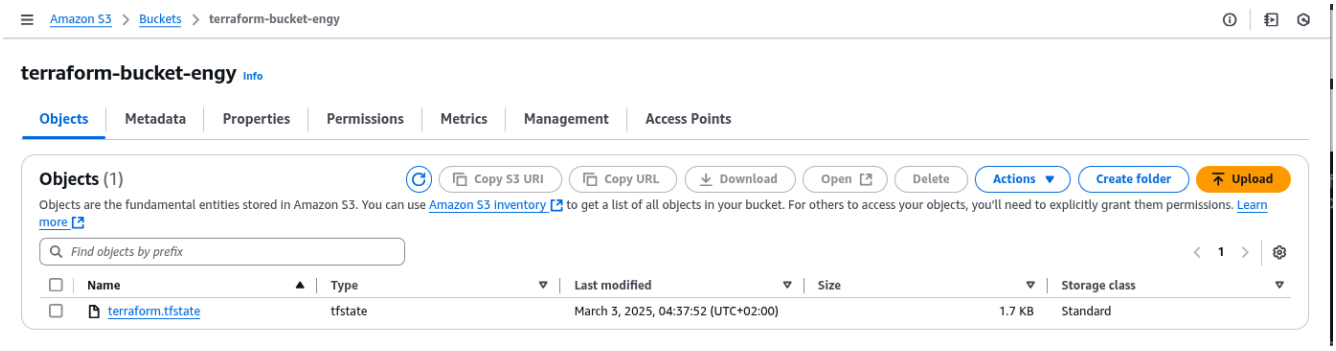
- Created provider file and used in it the terraform profile that has the accesskeys that we added.



- Created a s3 bucket named terraform-bucket-engy



- Created a backend.tf file and added my bucket name “terraform-bucket-engy “ and the key that will be created inside of it .

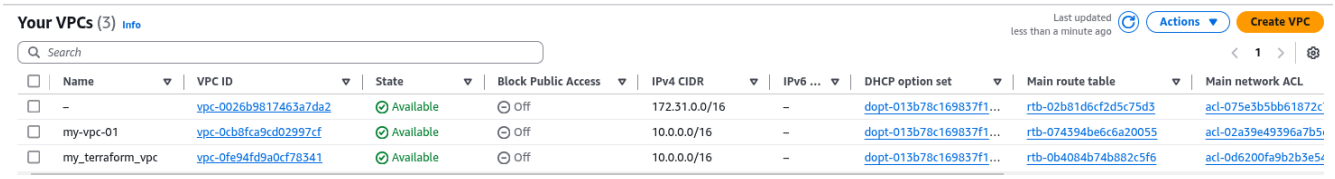


→ **was created after terraform apply command**

1- create vpc

```
[engy@localhost Terraform]$ terraform apply

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

Terraform will perform the following actions:

  # aws_vpc.main will be created
  + resource "aws_vpc" "main" {
      + 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)
```

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

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

  Enter a value: yes

aws_vpc.main: Creating...
aws_vpc.main: Creation complete after 3s [id=vpc-0fe94fd9a0cf78341]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[engy@localhost Terraform]$
```



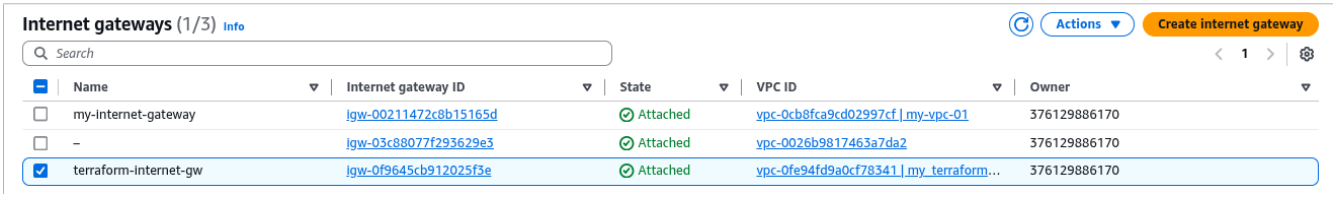
2- create internet gateway

internet_gateway.tf

```
1  resource "aws_internet_gateway" "gw" {
2    vpc_id = aws_vpc.main.id
3
4    tags = {
5      Name = "terraform-lab-gw"
6    }
7  }
8
```

```
[engy@localhost Terraform]$ terraform apply
aws_vpc.main: Refreshing state... [id=vpc-0fe94fd9a0cf78341]

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_internet_gateway.gw will be created
  + resource "aws_internet_gateway" "gw" {
      + arn      = (known after apply)
      + id       = (known after apply)
      + owner_id = (known after apply)
      + tags     = {
          + "Name" = "terraform-internet-gw"
        }
      + tags_all = {
          + "Name" = "terraform-internet-gw"
        }
```

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

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

  Enter a value: yes

aws_internet_gateway.gw: Creating...
aws_internet_gateway.gw: Creation complete after 2s [id=igw-0f9645cb912025f3e]

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



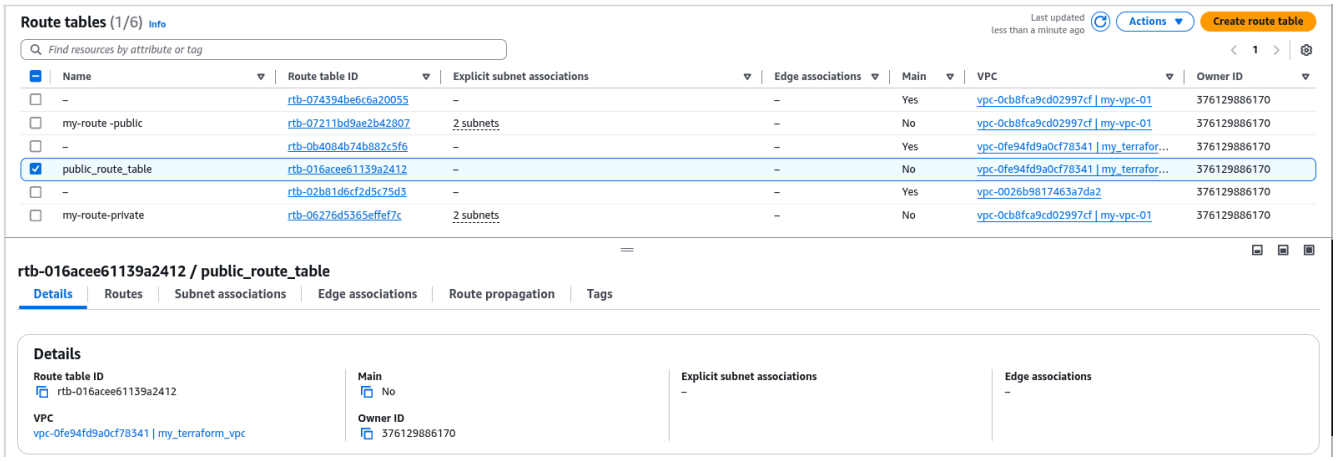
3- create public route table

public_route-table

```
public_route-table
1   resource "aws_route_table" "public_route" {
2     vpc_id = aws_vpc.main.id
3
4     route {
5       cidr_block = "0.0.0.0/0"
6       gateway_id = aws_internet_gateway.gw.id
7     }
8
9     tags = {
10      Name = "public_route_table"
11    }
12  }
13
```

```
[engy@localhost Terraform]$ terraform apply
aws_vpc.main: Refreshing state... [id=vpc-0fe94fd9a0cf78341]
aws_internet_gateway.gw: Refreshing state... [id=igw-0f9645cb912025f3e]

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_route_table.public_route will be created
  + resource "aws_route_table" "public_route" {
      + 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                = "igw-0f9645cb912025f3e"
                # (11 unchanged attributes hidden)
            },
```

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

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

  Enter a value: yes

aws_route_table.public_route: Creating...
aws_route_table.public_route: Creation complete after 2s [id=rtb-016acee61139a2412]

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



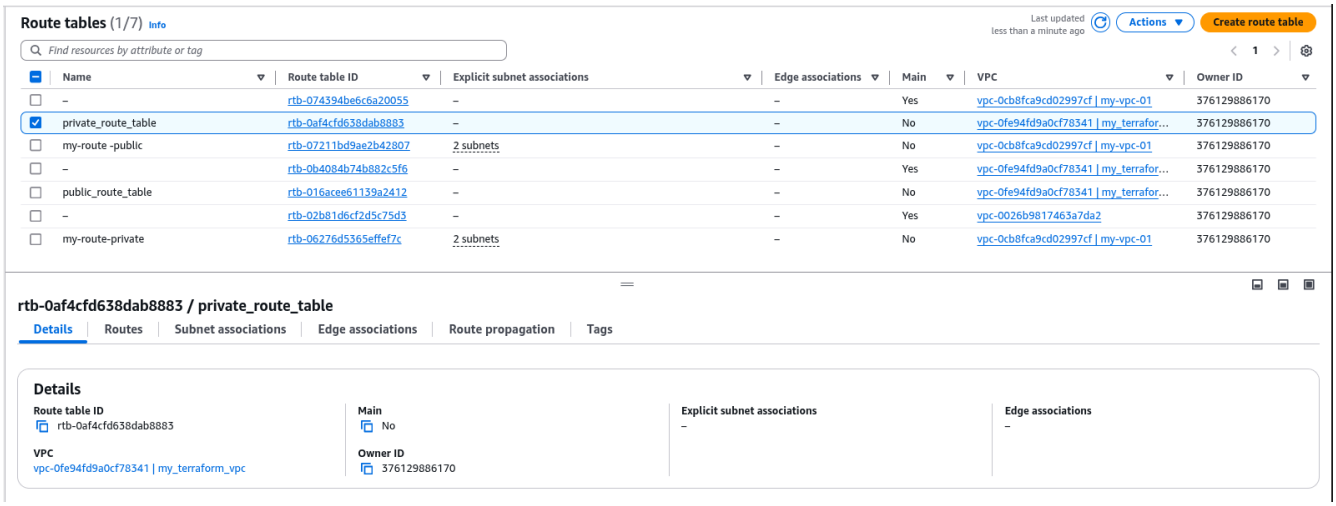
4- create private route table

private_route_table.tf

```
resource "aws_route_table" "private_route" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "private_route_table"
  }
}
```

```
[engy@localhost Terraform]$ terraform apply
aws_vpc.main: Refreshing state... [id=vpc-0fe94fd9a0cf78341]
aws_internet_gateway.gw: Refreshing state... [id=igw-0f9645cb912025f3e]
aws_route_table.public_route: Refreshing state... [id=rtb-016acee61139a2412]

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_route_table.private_route will be created
  + resource "aws_route_table" "private_route" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = (known after apply)
      + tags             = {
          + "Name" = "private_route_table"
        }
      + tags_all         = {
          + "Name" = "private route table"
```

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

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

  Enter a value: yes

aws_route_table.private_route: Creating...
aws_route_table.private_route: Creation complete after 2s [id=rtb-0af4cfd638dab8883]

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



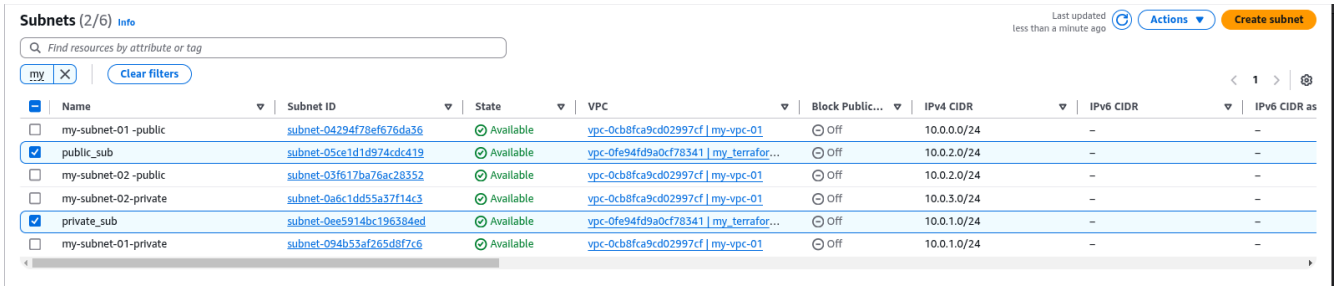
5- create public route

```
route {
  cidr_block = "0.0.0.0/0"
  gateway_id = aws_internet_gateway.gw.id
}
```

- Create two subnets (1 private and 1 public)

```
# aws_subnet.private_sub will be created
+ resource "aws_subnet" "private_sub" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.1.0/24"
    + 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"
      }
    + tags_all                                       = {
        + "Name" = "private_sub"
      }
    + vpc_id                                         = "vpc-0fe94fd9a0cf78341"
  }

# aws_subnet.public_sub will be created
+ resource "aws_subnet" "public_sub" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.2.0/24"
    + 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                                           = {
```

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

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

  Enter a value: yes

aws_subnet.public_sub: Creating...
aws_subnet.private_sub: Creating...
aws_subnet.private_sub: Creation complete after 2s [id=subnet-0ee5914bc196384ed]
aws_subnet.public_sub: Creation complete after 3s [id=subnet-05ce1d1d974cdc419]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```



6- Attach public route table to subnets

public_route_table.tf

```
resource "aws_route_table" "public_route" {
  vpc_id = aws_vpc.main.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }

  tags = {
    Name = "public_route_table"
  }
}
resource "aws_route_table_association" "public_route_association" {
  subnet_id      = aws_subnet.public_sub.id
  route_table_id = aws_route_table.public_route.id
}
```

```
[engy@localhost Terraform]$ terraform apply
aws_vpc.main: Refreshing state... [id=vpc-0fe94fd9a0cf78341]
aws_route_table.private_route: Refreshing state... [id=rtb-0af4cfd638dab8883]
aws_internet_gateway.gw: Refreshing state... [id=igw-0f9645cb912025f3e]
aws_subnet.private_sub: Refreshing state... [id=subnet-0ee5914bc196384ed]
aws_subnet.public_sub: Refreshing state... [id=subnet-05ce1d1d974cdc419]
aws_route_table.public_route: Refreshing state... [id=rtb-016acee61139a2412]

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_route_table_association.public_route_association will be created
  + resource "aws_route_table_association" "public_route_association" {
      + id             = (known after apply)
      + route_table_id = "rtb-016acee61139a2412"
      + subnet_id      = "subnet-05ce1d1d974cdc419"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

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

  Enter a value: yes

aws_route_table_association.public_route_association: Creating...
aws_route_table_association.public_route_association: Creation complete after 2s [id=rtbassoc-071543b81fc68
1ce6]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[engy@localhost Terraform]$
```

Route tables (1/7) Info

Last updated less than a minute ago | Actions | Create route table

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
–	rtb-074394be6c6a20055	–	–	Yes	vpc-0cb8fca9cd02997cf | my-vpc-01	376129886170
private_route_table	rtb-0af4cfd638dab8883	–	–	No	vpc-0fe94fd9a0cf78341 | my_terrafor…	376129886170
my-route -public	rtb-07211bd9ae2b42807	2 subnets	–	No	vpc-0cb8fca9cd02997cf | my-vpc-01	376129886170
–	rtb-0b4084b74b882c5f6	–	–	Yes	vpc-0fe94fd9a0cf78341 | my_terrafor…	376129886170
public_route_table	rtb-016acee61139a2412	subnet-05ce1d1d974cdc419 / public_sub	–	No	vpc-0fe94fd9a0cf78341 | my_terrafor…	376129886170
–	rtb-02b81d6cf2d5c75d3	–	–	Yes	vpc-0026b9817463a7da2	376129886170
my-route-private	rtb-06276d5365effef7c	2 subnets	–	No	vpc-0cb8fca9cd02997cf | my-vpc-01	376129886170

rtb-016acee61139a2412 / public_route_table

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-016acee61139a2412	No	subnet-05ce1d1d974cdc419 / public_sub	–

VPC	Owner ID
vpc-0fe94fd9a0cf78341 | my_terraform_vpc	376129886170

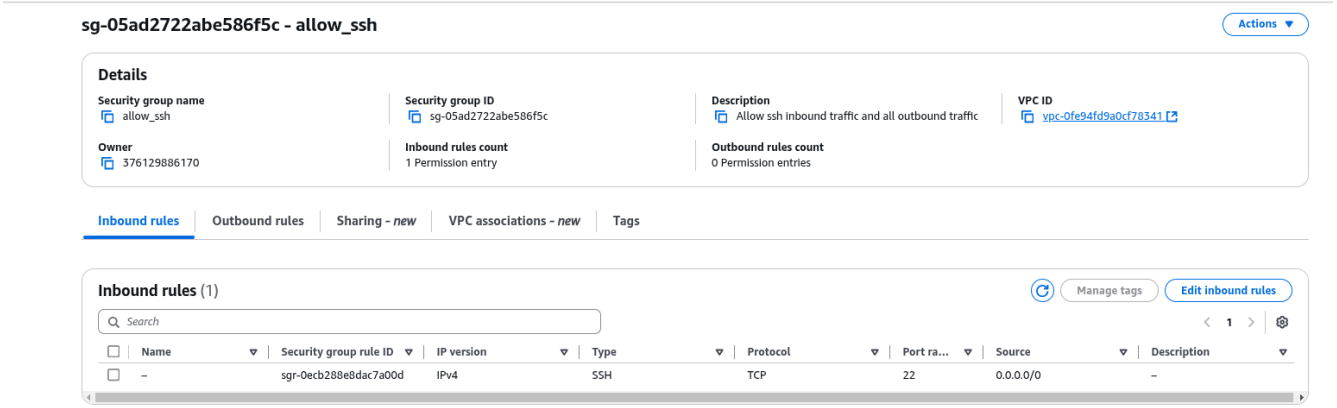
Compute

7- create security group which allow ssh from 0.0.0.0/0

sg_1.tf

```hcl
resource "aws_security_group" "sg_1_terraform" {
  name        = "allow_ssh"
  description = "Allow ssh inbound traffic and all outbound traffic"
  vpc_id      = aws_vpc.main.id

  tags = {
    Name = "allow_ssh"
  }
}

resource "aws_vpc_security_group_ingress_rule" "allow_ssh_ipv4" {
  security_group_id = aws_security_group.sg_1_terraform.id
  cidr_ipv4         = "0.0.0.0/0"
  from_port         = 22
  ip_protocol       = "tcp"
  to_port           = 22
}
```

```
Terraform will perform the following actions:

  # aws_security_group.sg_1_terraform will be created
  + resource "aws_security_group" "sg_1_terraform" {
      + arn                    = (known after apply)
      + description            = "Allow ssh inbound traffic and all outbound traffic"
      + egress                 = (known after apply)
      + id                     = (known after apply)
      + ingress                = (known after apply)
      + name                   = "allow_ssh"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags                   = {
          + "Name" = "allow_ssh"
        }
      + tags_all               = {
          + "Name" = "allow_ssh"
        }
      + vpc_id                 = "vpc-0fe94fd9a0cf78341"
    }

  # aws_vpc_security_group_ingress_rule.allow_ssh_ipv4 will be created
  + resource "aws_vpc_security_group_ingress_rule" "allow_ssh_ipv4" {
      + arn                    = (known after apply)
      + cidr_ipv4              = "0.0.0.0/0"
      + from_port              = 22
      + id                     = (known after apply)
      + ip_protocol            = "tcp"
      + security_group_id      = (known after apply)
      + security_group_rule_id = (known after apply)
      + tags_all               = {}
      + to_port                = 22
    }

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

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

  Enter a value: yes

aws_security_group.sg_1_terraform: Creating...
aws_security_group.sg_1_terraform: Creation complete after 4s [id=sg-05ad2722abe586f5c]
aws_vpc_security_group_ingress_rule.allow_ssh_ipv4: Creating...
aws_vpc_security_group_ingress_rule.allow_ssh_ipv4: Creation complete after 1s [id=sgr-0ecb288e8dac7a00d]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```



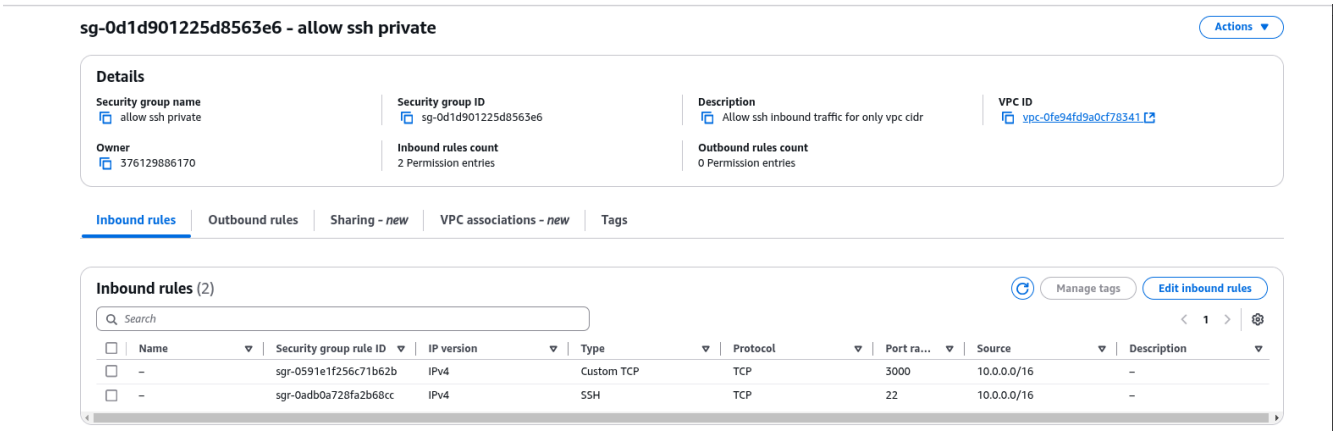
8- create security group that allow ssh and port 3000 from vpc cidr only

sg_1.tf sg_2.tf

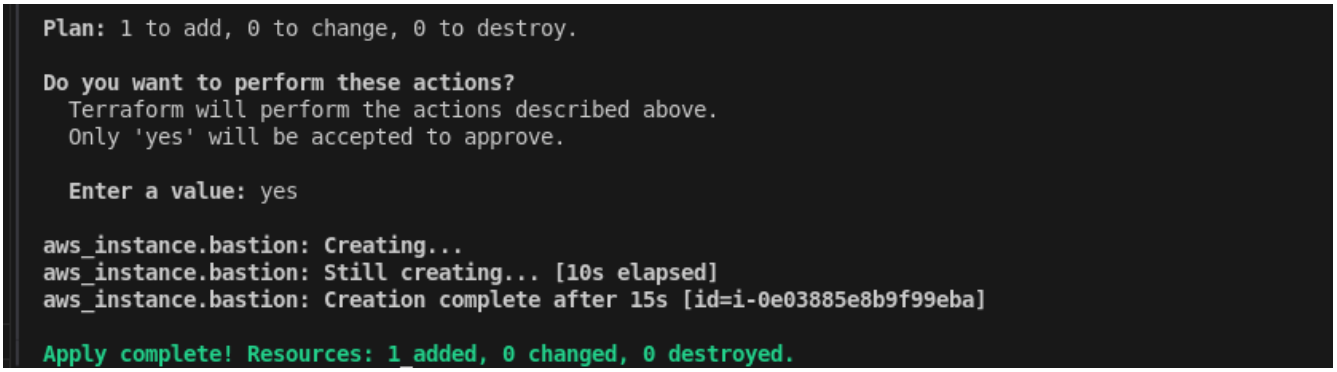
sg_2.tf

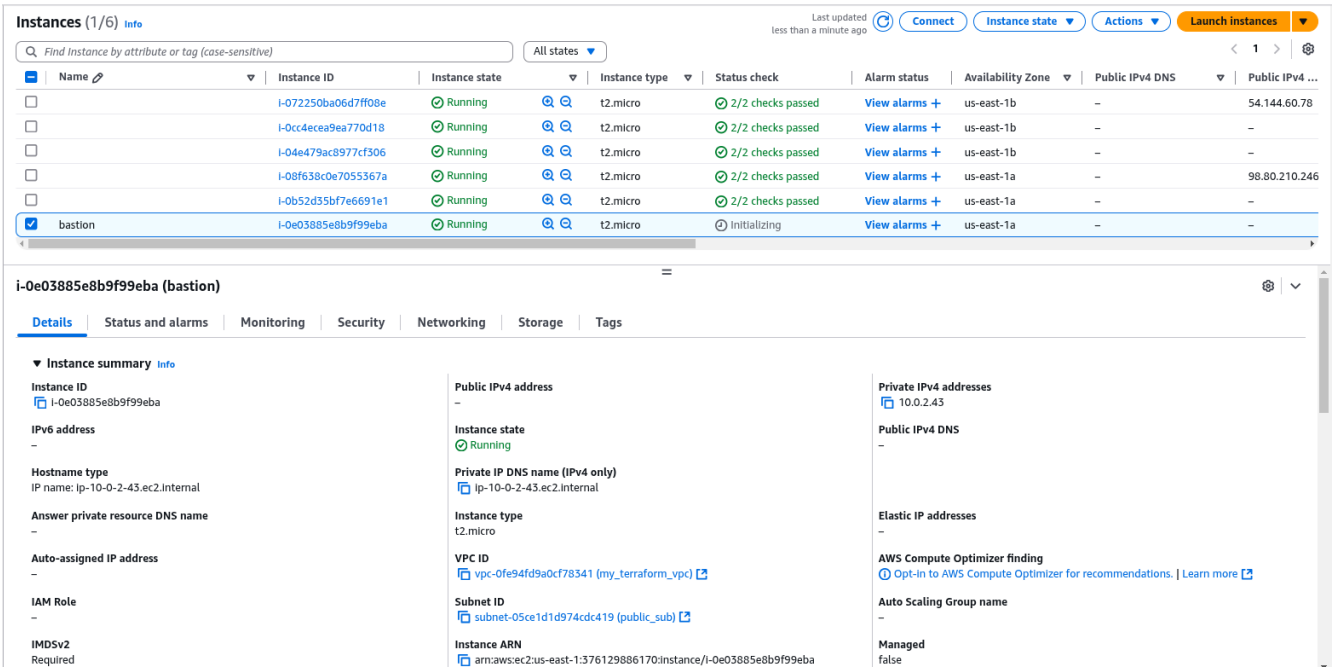
```
resource "aws_security_group" "sg_2_terraform" {
  name        = "allow ssh private"
  description = "Allow ssh inbound traffic for only vpc cidr"
  vpc_id      = aws_vpc.main.id

  tags = {
    Name = "allow_ssh_private"
  }
}

resource "aws_vpc_security_group_ingress_rule" "allow_ssh_ipv4_2" {
  security_group_id = aws_security_group.sg_2_terraform.id
  cidr_ipv4         = aws_vpc.main.cidr_block
  from_port         = 22
  ip_protocol       = "tcp"
  to_port           = 22
}

resource "aws_vpc_security_group_ingress_rule" "allow_3000_ipv4_2" {
  security_group_id = aws_security_group.sg_2_terraform.id
  cidr_ipv4         = aws_vpc.main.cidr_block
  from_port         = 3000
  ip_protocol       = "tcp"
  to_port           = 3000
}
```

```
Plan: 3 to add, 0 to change, 0 to destroy.

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

  Enter a value: yes

aws_security_group.sg_2_terraform: Creating...
aws_security_group.sg_2_terraform: Creation complete after 4s [id=sg-0d1d901225d8563e6]
aws_vpc_security_group_ingress_rule.allow_3000_ipv4_2: Creating...
aws_vpc_security_group_ingress_rule.allow_ssh_ipv4_2: Creating...
aws_vpc_security_group_ingress_rule.allow_3000_ipv4_2: Creation complete after 1s [id=sgr-0591e1f256c71b62b
]
aws_vpc_security_group_ingress_rule.allow_ssh_ipv4_2: Creation complete after 1s [id=sgr-0adb0a728fa2b68cc]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
[engy@localhost Terraform]$
```

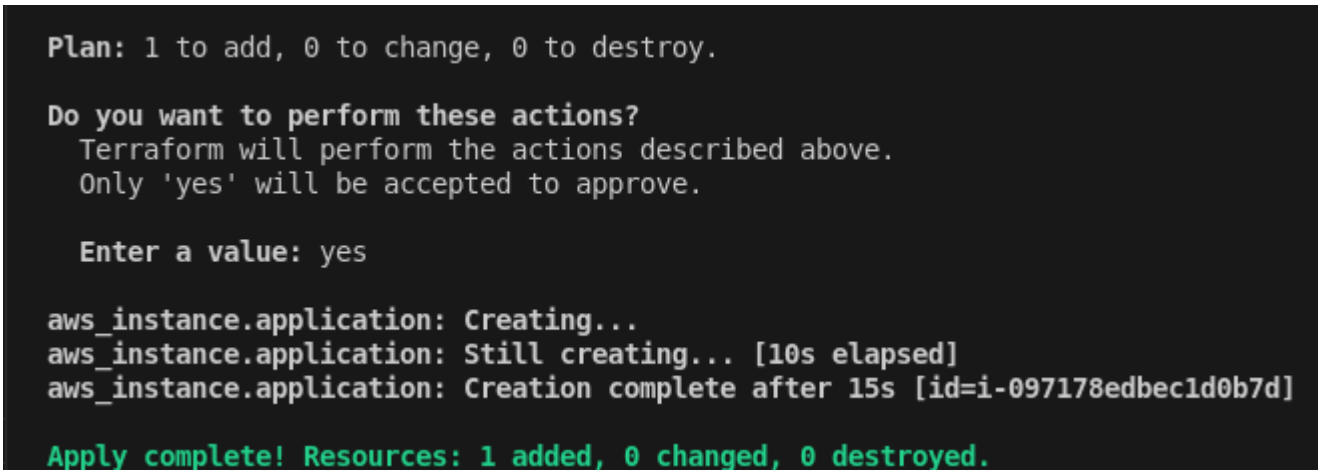


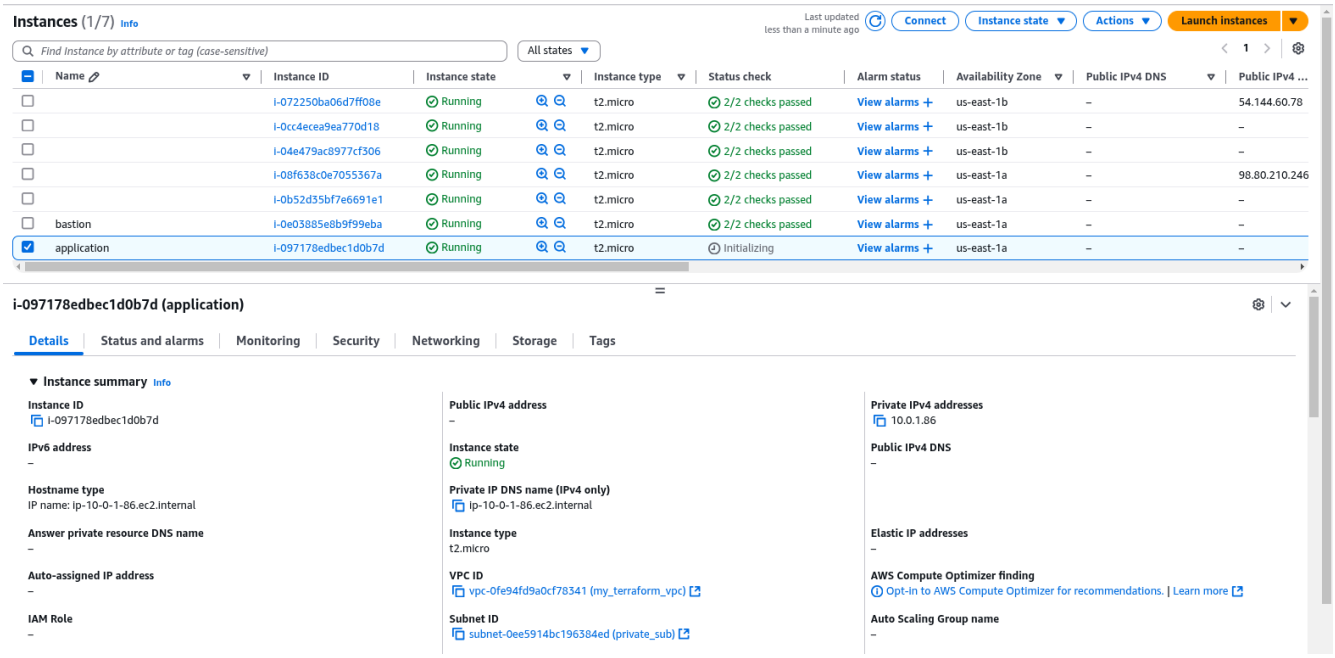
7- create ec2(bastion) in public subnet with security group from 7





8- create ec2(application) private subnet with security group from 8





Finally , that's all ! :)
You can access my github repo for this Lab through :
https://github.com/EngyElhamzawy/Terraform_Labs