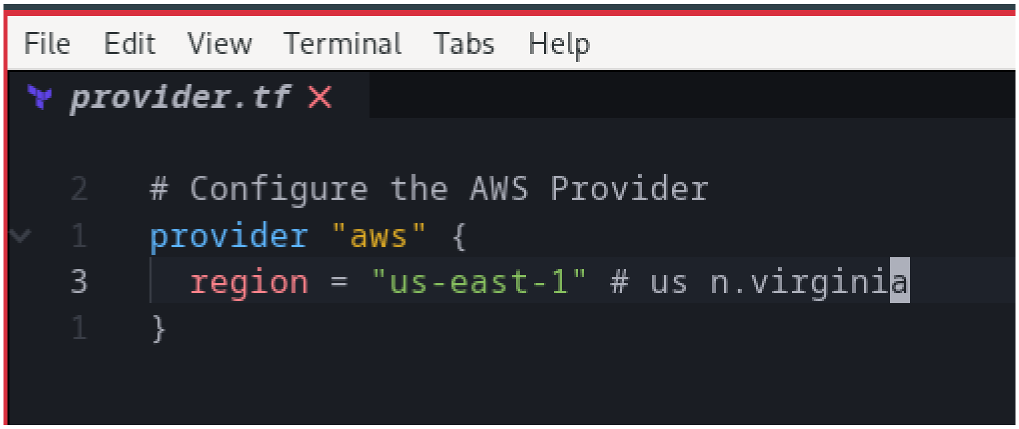
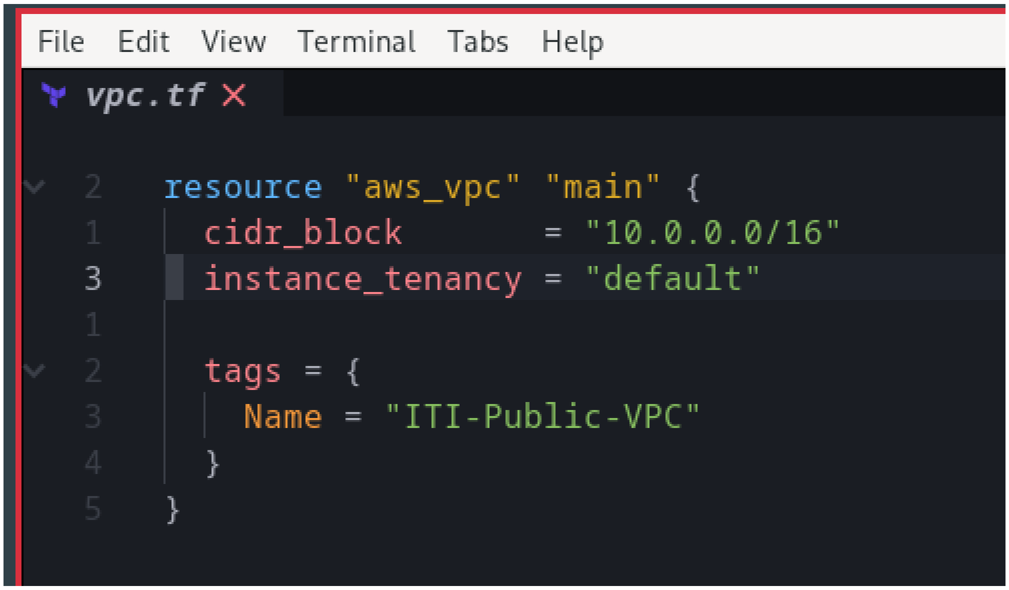
Networking

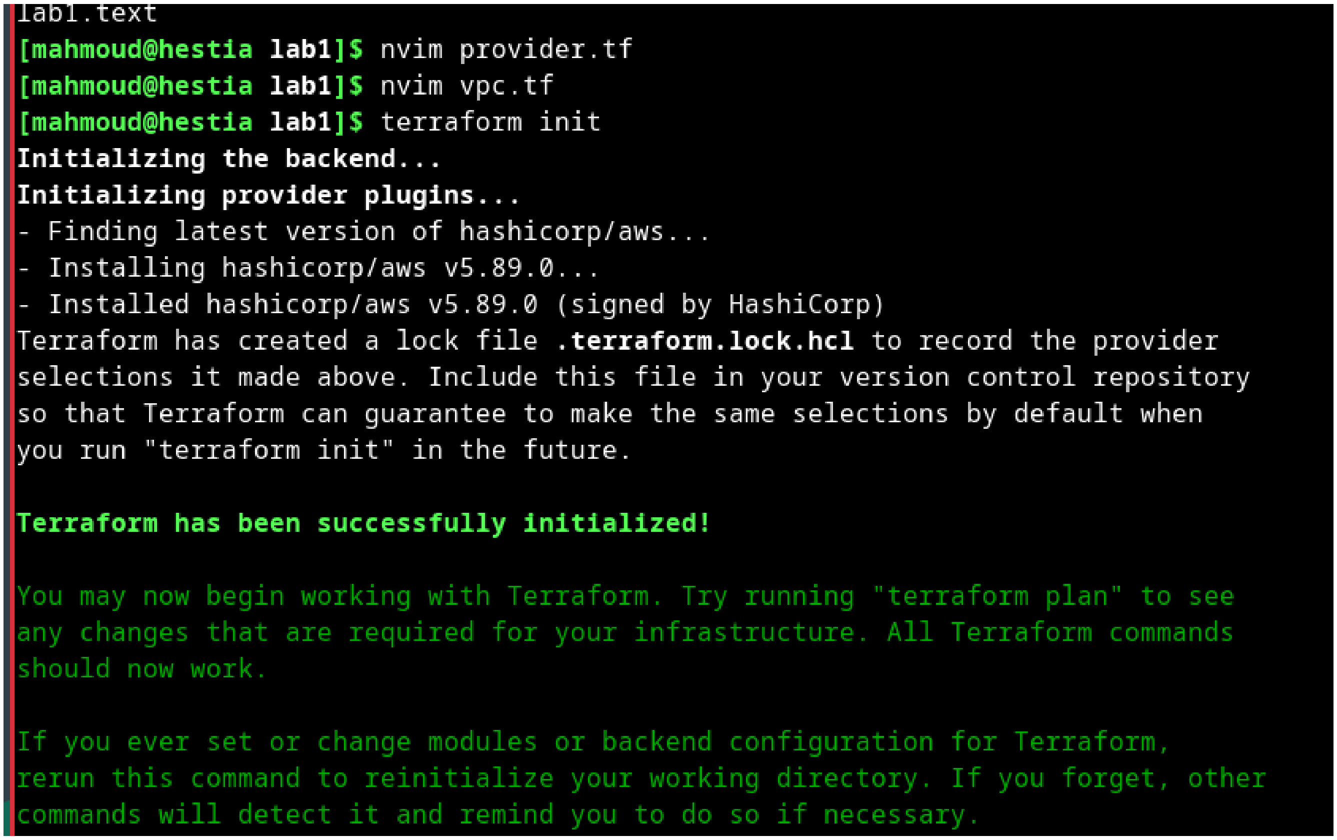
1- create vpc



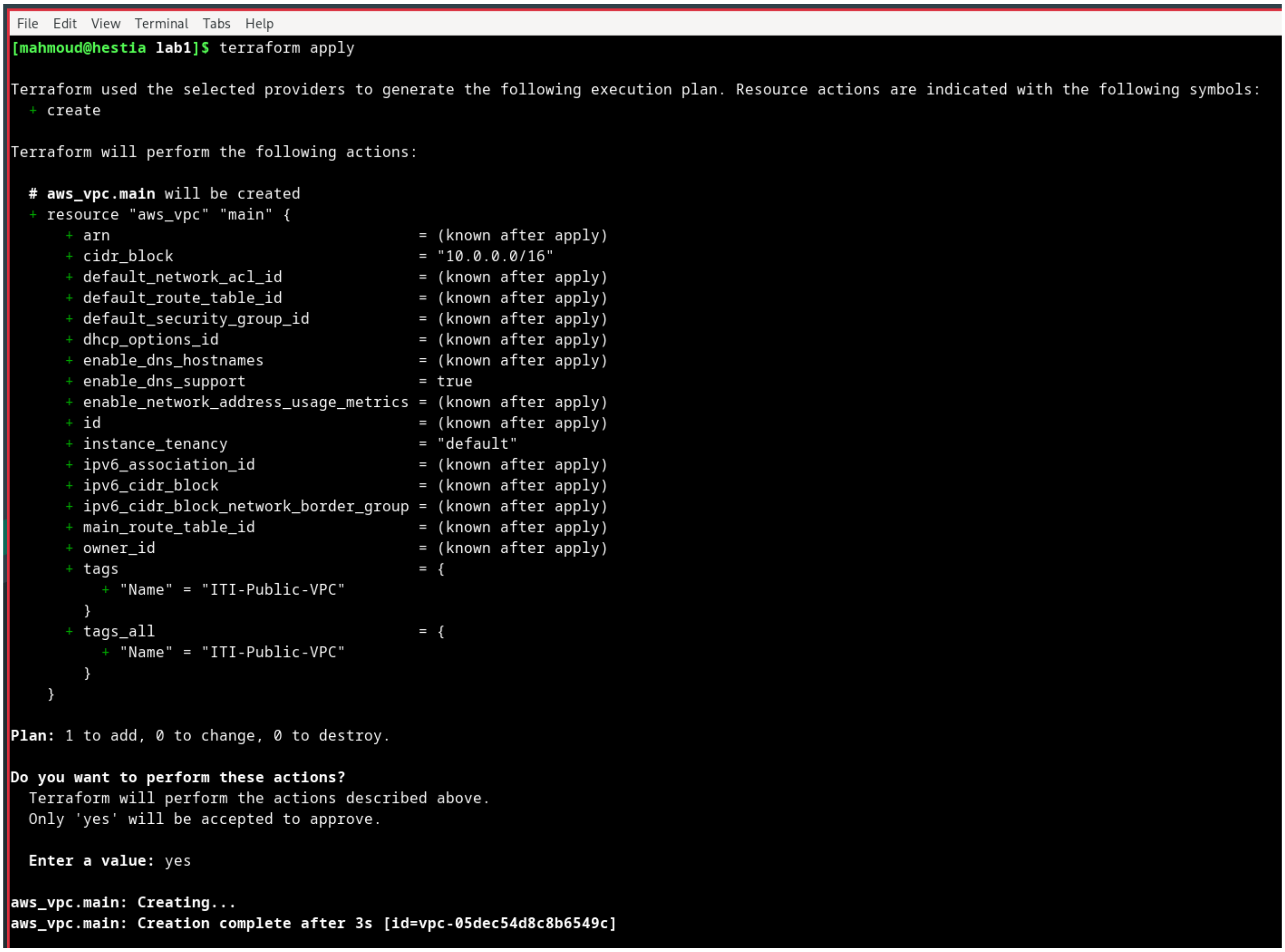
```
# Configure the AWS Provider
provider "aws" {
  region = "us-east-1" # us n.virginia
}
```



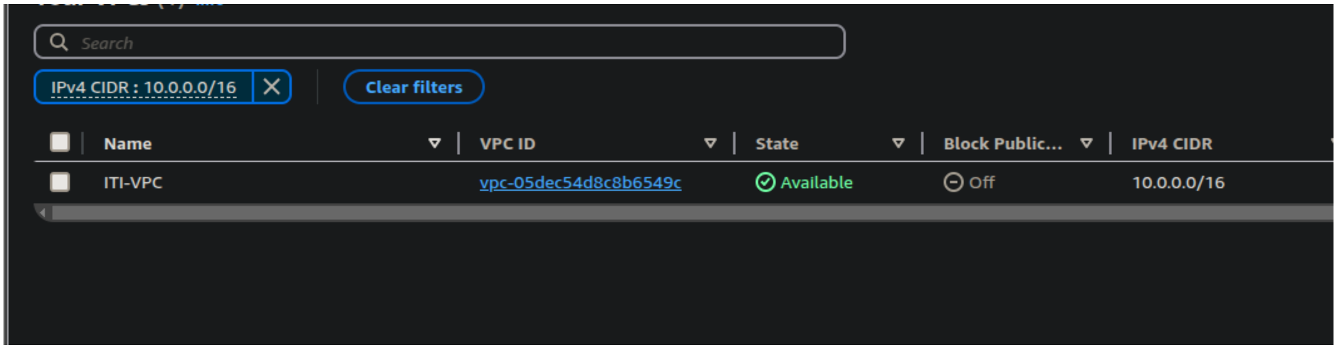
```
resource "aws_vpc" "main" {
  cidr_block       = "10.0.0.0/16"
  instance_tenancy = "default"

  tags = {
    Name = "ITI-Public-VPC"
  }
}
```



```
lab1.text
[mahmoud@hestia lab1]$ nvim provider.tf
[mahmoud@hestia lab1]$ nvim vpc.tf
[mahmoud@hestia lab1]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.89.0...
- Installed hashicorp/aws v5.89.0 (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.
```

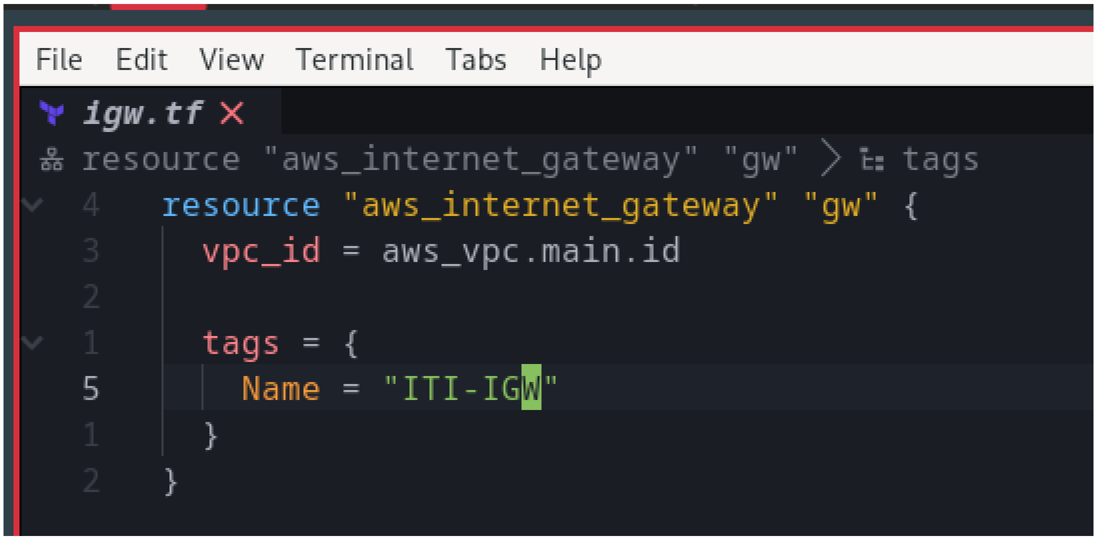


```
[mahmoud@hestia lab1]$ terraform apply

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)
      + 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" = "ITI-Public-VPC"
        }
      + tags_all                             = {
          + "Name" = "ITI-Public-VPC"
        }
    }

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-05dec54d8c8b6549c]
```



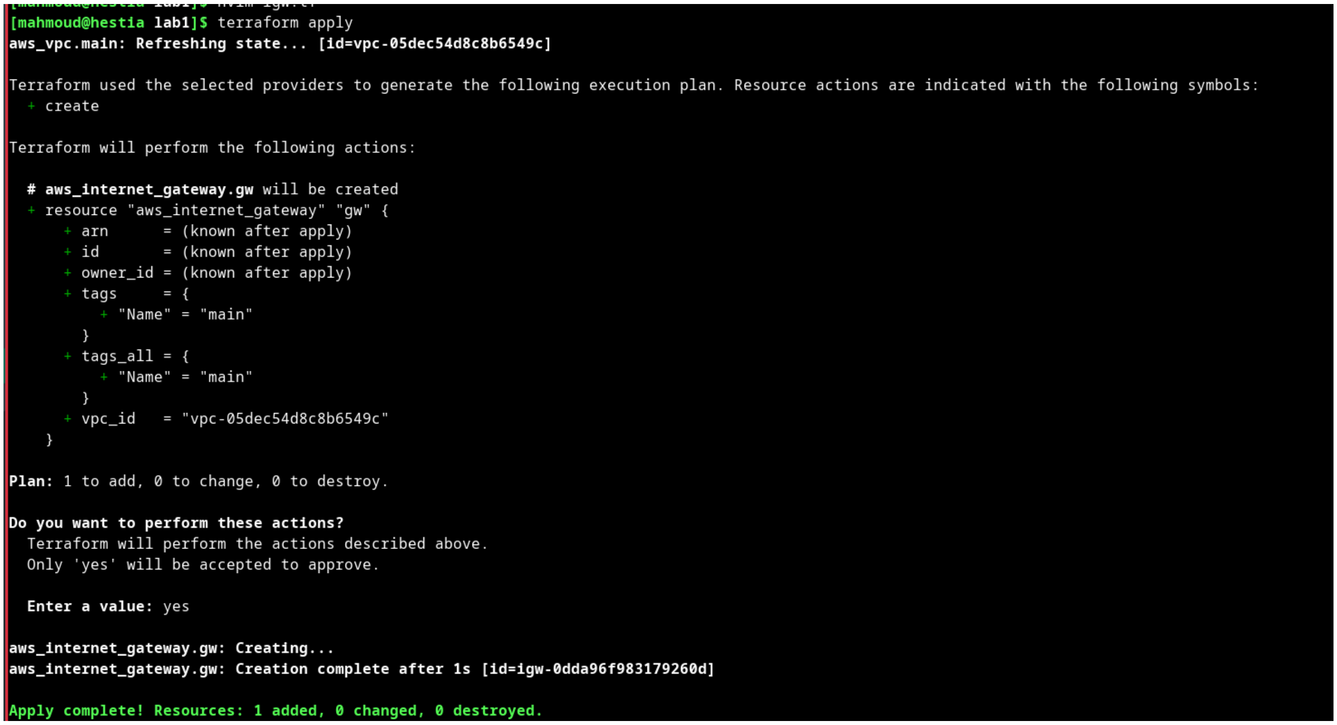
□	Name	VPC ID	State	Block Public...	IPv4 CIDR
□	ITI-VPC	vpc-05dec54d8c8b6549c	✓ Available	⊖ Off	10.0.0.0/16

2- create internet gateway

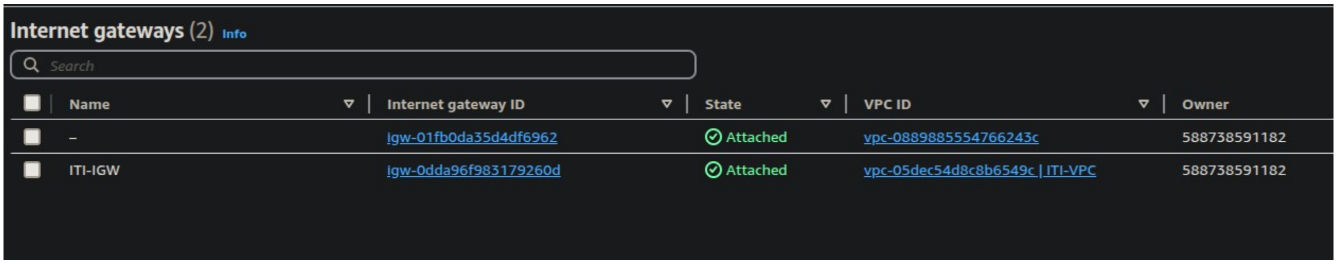


```
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "ITI-IGW"
  }
}
```

Terraform is attaching this to the needed vpc directly through the vpc_id

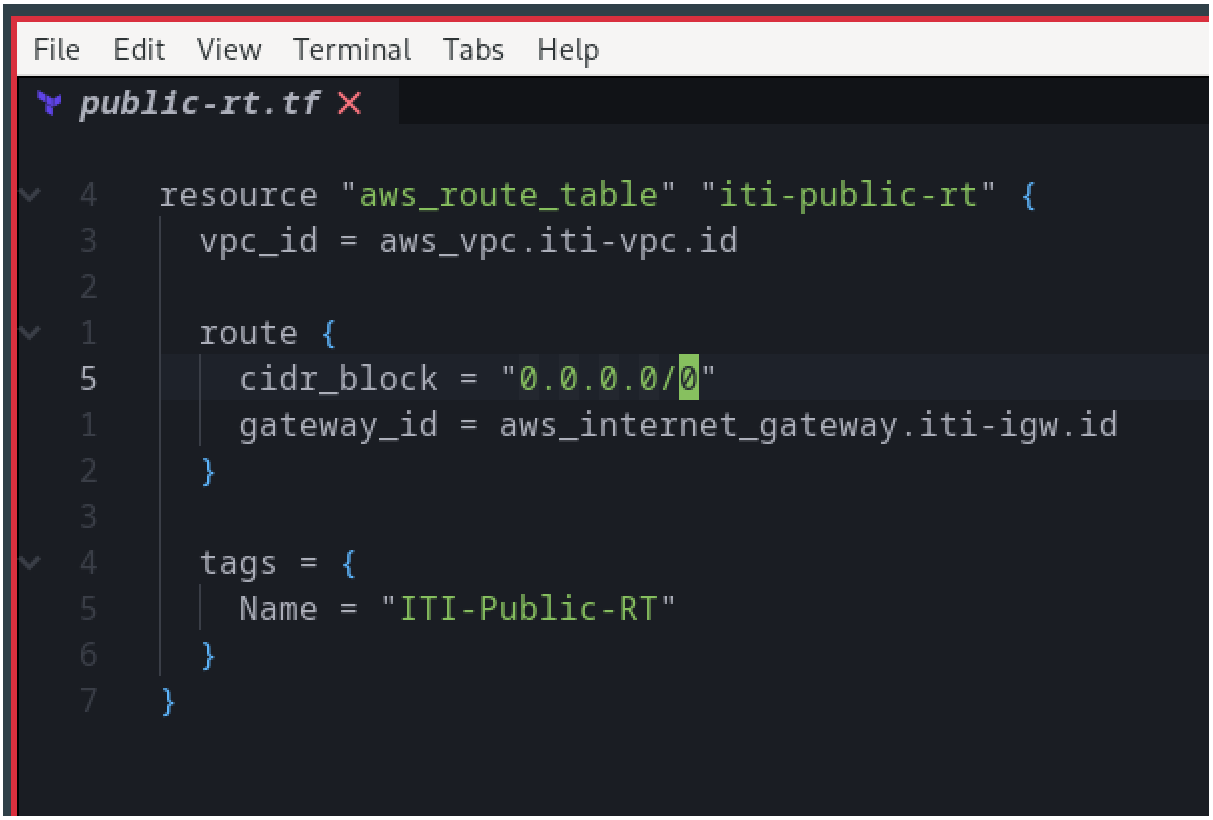


```
[mahmoud@hestia lab1]$ nvim igw.tf
[mahmoud@hestia lab1]$ terraform apply
aws_vpc.main: Refreshing state... [id=vpc-05dec54d8c8b6549c]

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" = "main"
        }
      + tags_all = {
          + "Name" = "main"
        }
      + vpc_id   = "vpc-05dec54d8c8b6549c"
    }

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 1s [id=igw-0dda96f983179260d]

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

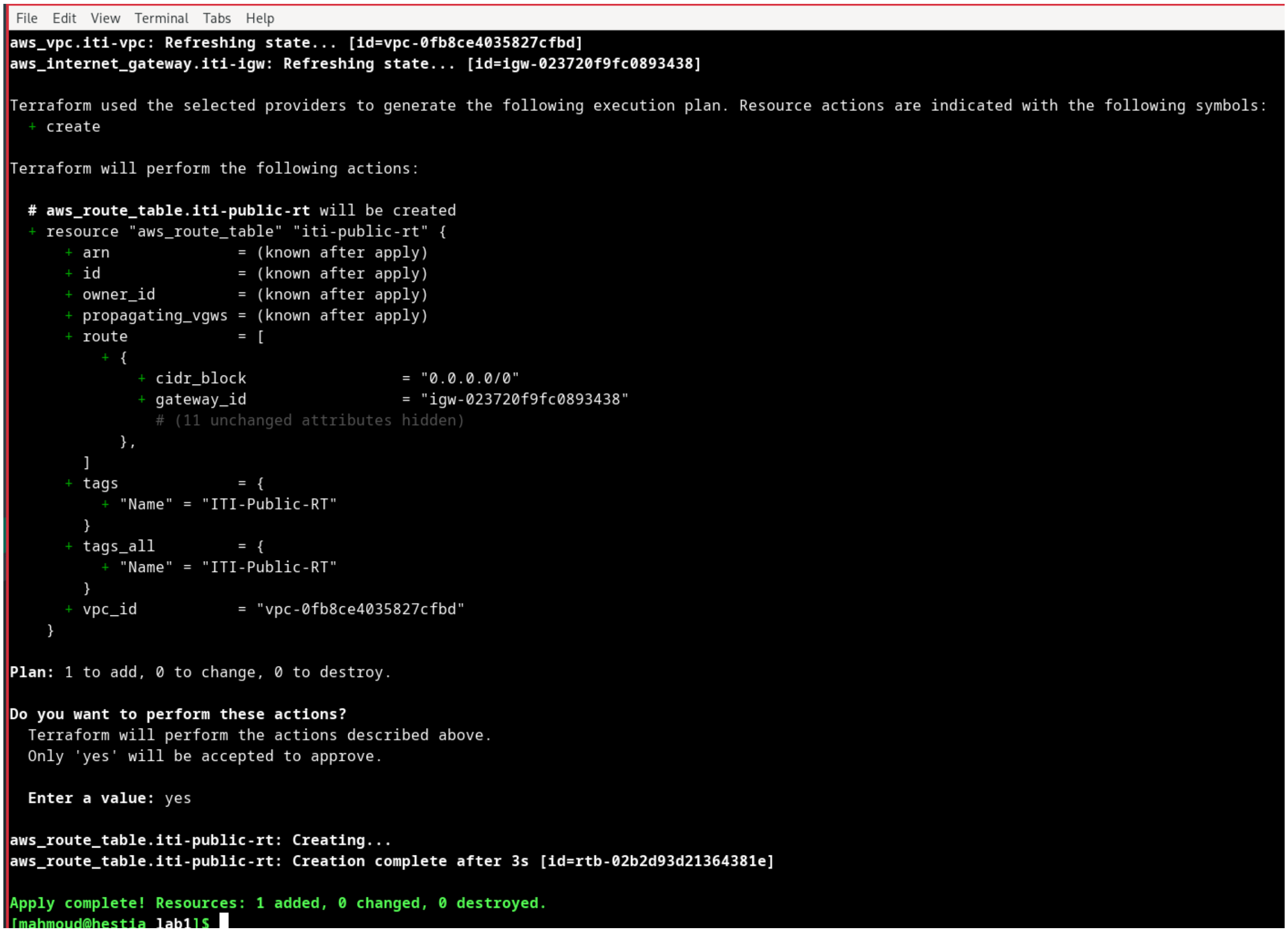


□	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-01fb0da35d4df6962	✓ Attached	vpc-0889885554766243c	588738591182
□	ITI-IGW	igw-0dda96f983179260d	✓ Attached	vpc-05dec54d8c8b6549c | ITI-VPC	588738591182

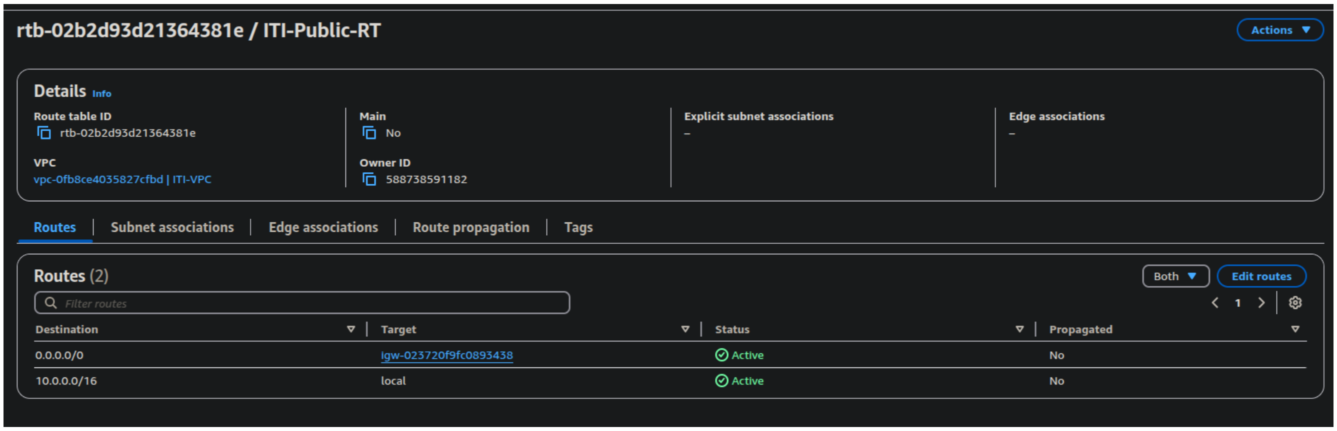
3- create public route table



```
resource "aws_route_table" "iti-public-rt" {
  vpc_id = aws_vpc.iti-vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.iti-igw.id
  }

  tags = {
    Name = "ITI-Public-RT"
  }
}
```



```
aws_vpc.iti-vpc: Refreshing state... [id=vpc-0fb8ce4035827cfbd]
aws_internet_gateway.iti-igw: Refreshing state... [id=igw-023720f9fc0893438]

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.iti-public-rt will be created
  + resource "aws_route_table" "iti-public-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                 = "igw-023720f9fc0893438"
                # (11 unchanged attributes hidden)
            },
        ]
      + tags             = {
          + "Name" = "ITI-Public-RT"
        }
      + tags_all         = {
          + "Name" = "ITI-Public-RT"
        }
      + vpc_id           = "vpc-0fb8ce4035827cfbd"
    }

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.iti-public-rt: Creating...
aws_route_table.iti-public-rt: Creation complete after 3s [id=rtb-02b2d93d21364381e]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[mahmoud@hestia lab1]$
```



rtb-02b2d93d21364381e / ITI-Public-RT

Details Info

Route table ID: rtb-02b2d93d21364381e

Main: No

Explicit subnet associations: –

Edge associations: –

VPC: vpc-0fb8ce4035827cfbd | ITI-VPC

Owner ID: 588738591182

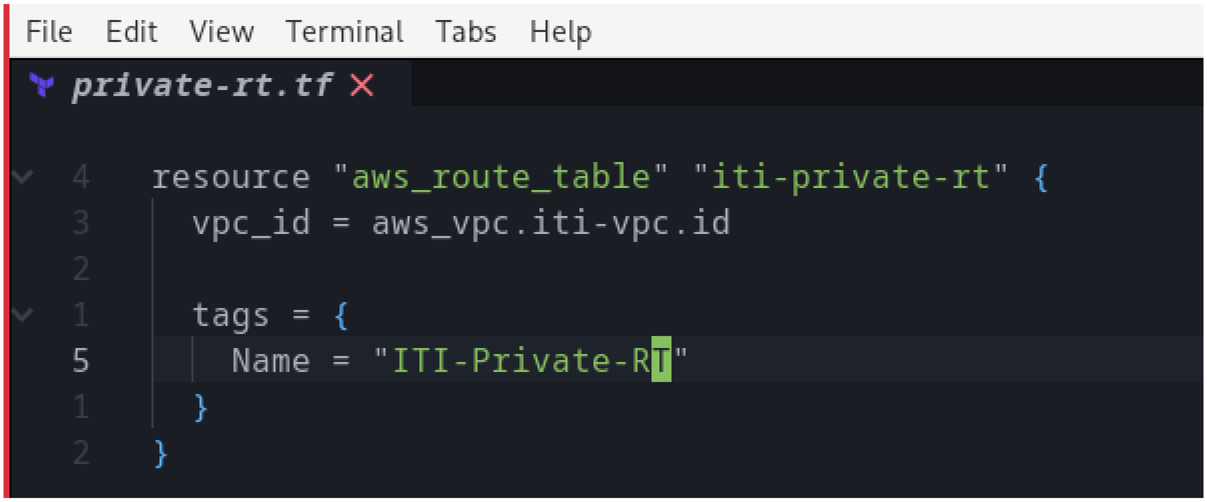
Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

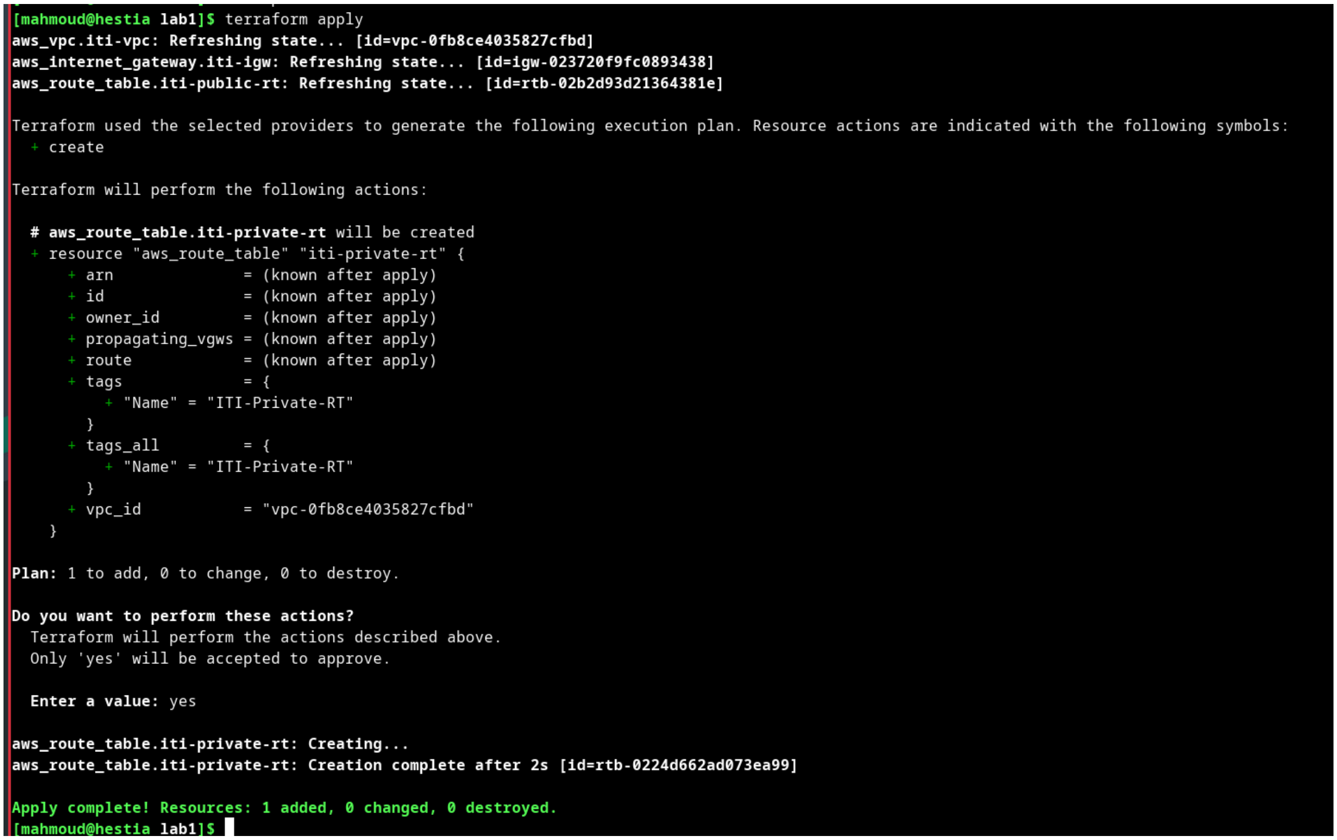
Both ▼
Edit routes
Filter routes
1

Destination	Target	Status	Propagated
0.0.0.0/0	igw-023720f9fc0893438	✓ Active	No
10.0.0.0/16	local	✓ Active	No

4- create private route table



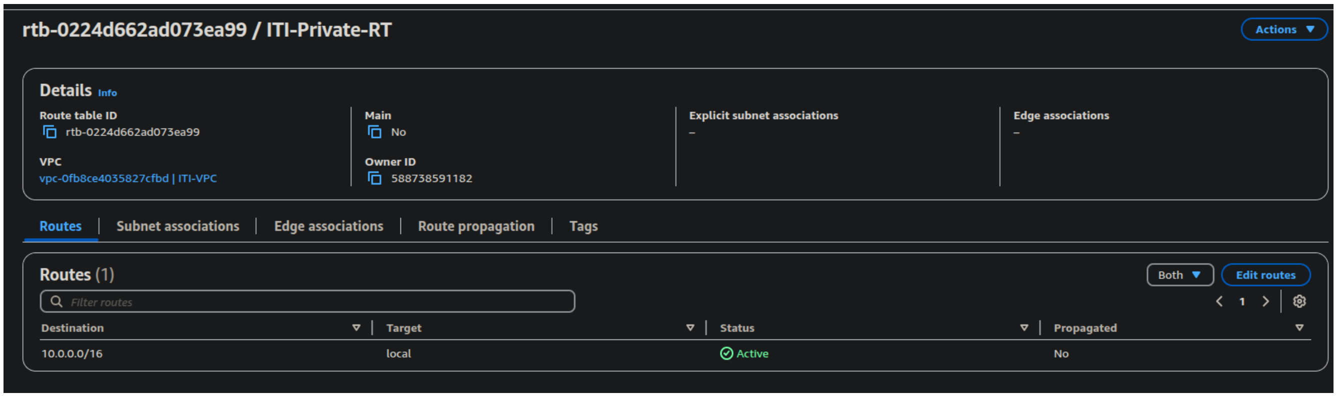
```
resource "aws_route_table" "iti-private-rt" {
  vpc_id = aws_vpc.iti-vpc.id

  tags = {
    Name = "ITI-Private-RT"
  }
}
```



```
[mahmoud@hestia lab1]$ terraform apply
aws_vpc.iti-vpc: Refreshing state... [id=vpc-0fb8ce4035827cfbd]
aws_internet_gateway.iti-igw: Refreshing state... [id=igw-023720f9fc0893438]
aws_route_table.iti-public-rt: Refreshing state... [id=rtb-02b2d93d21364381e]

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.iti-private-rt will be created
  + resource "aws_route_table" "iti-private-rt" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = (known after apply)
      + tags             = {
          + "Name" = "ITI-Private-RT"
        }
      + tags_all         = {
          + "Name" = "ITI-Private-RT"
        }
      + vpc_id           = "vpc-0fb8ce4035827cfbd"
    }

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.iti-private-rt: Creating...
aws_route_table.iti-private-rt: Creation complete after 2s [id=rtb-0224d662ad073ea99]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[mahmoud@hestia lab1]$
```

5- create public route

and the route table is connecting the local network by default



rtb-0224d662ad073ea99 / ITI-Private-RT

Details

Route table ID: rtb-0224d662ad073ea99

Main: No

Explicit subnet associations: –

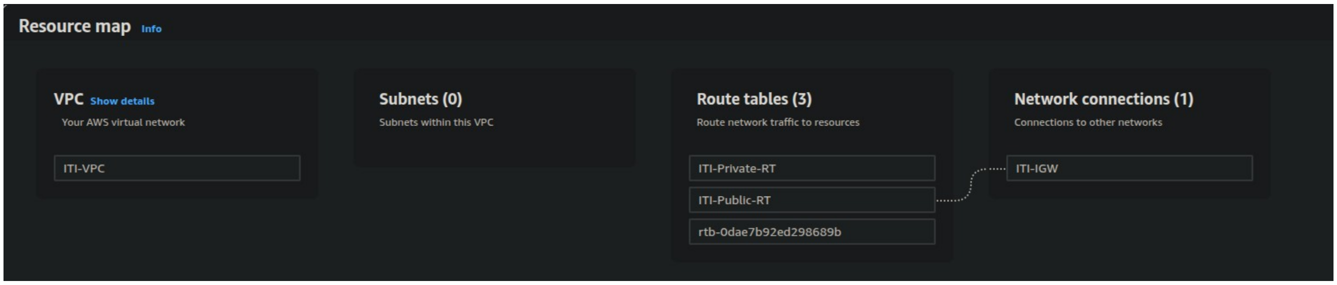
Edge associations: –

VPC: vpc-0fb8ce4035827cfbd | ITI-VPC

Owner ID: 588738591182

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	✓ Active	No



Resource map

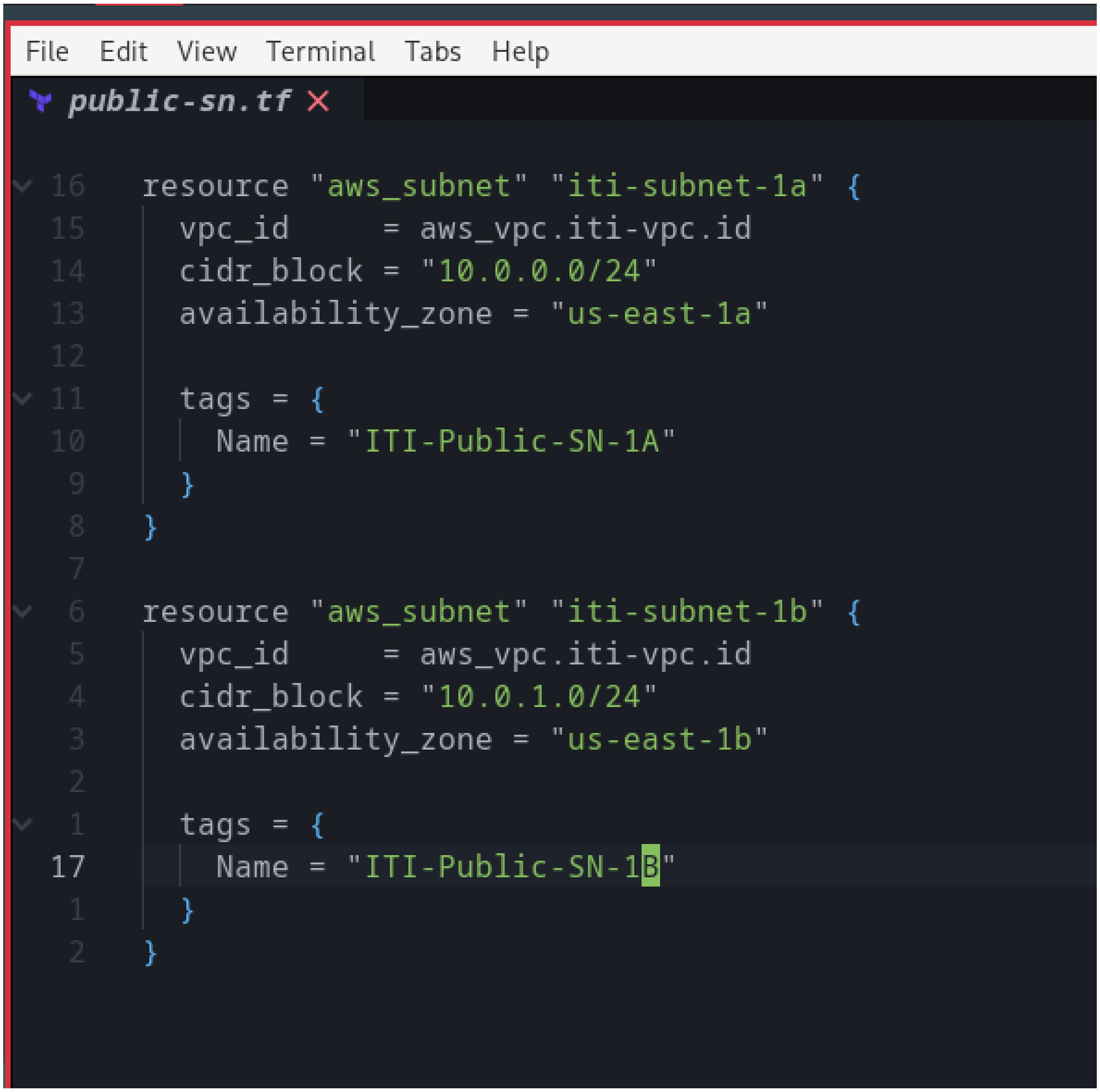
VPC
Your AWS virtual network
ITI-VPC

Subnets (0)
Subnets within this VPC

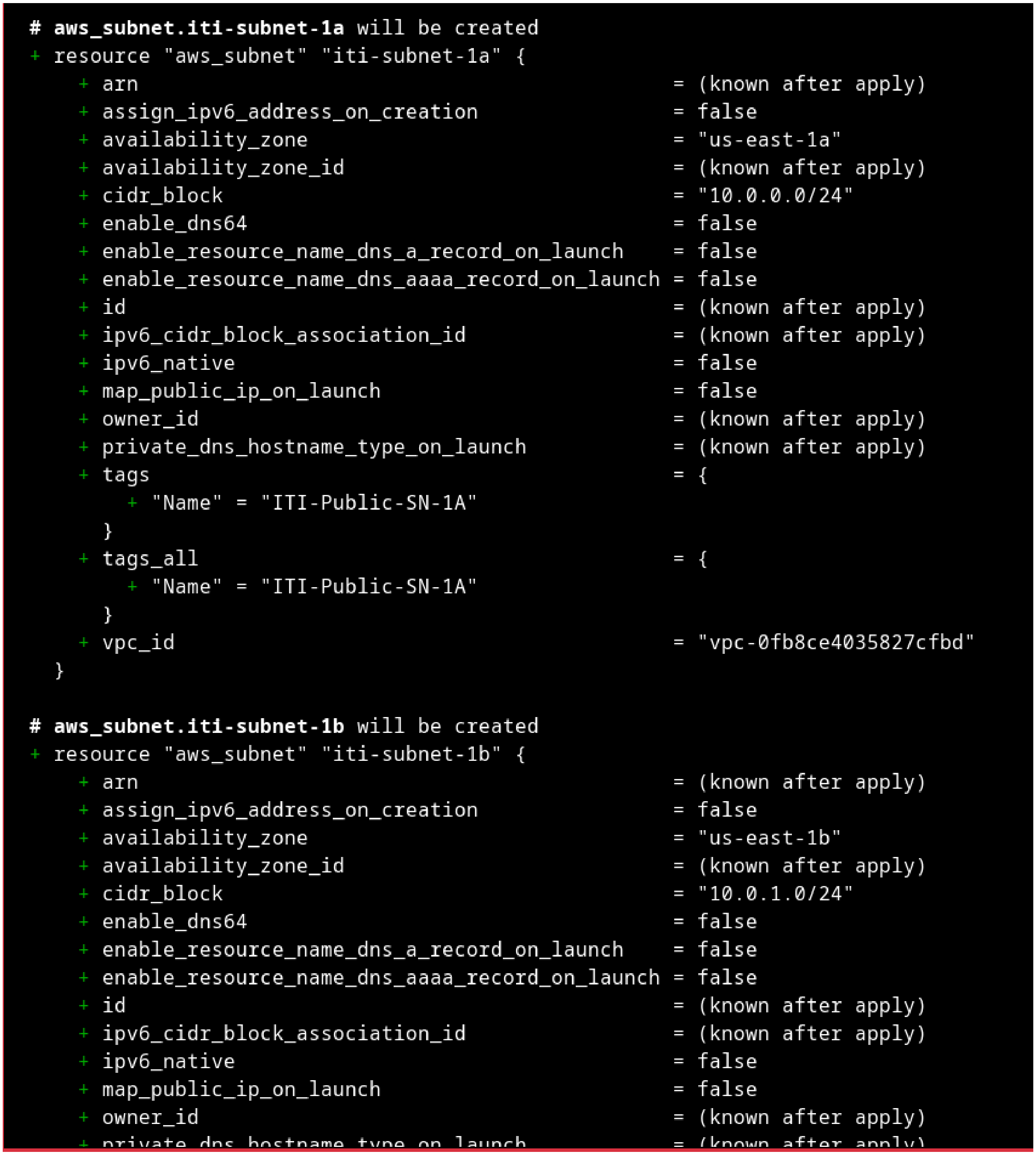
Route tables (3)
Route network traffic to resources
ITI-Private-RT
ITI-Public-RT
rtb-0dae7b92ed298689b

Network connections (1)
Connections to other networks
ITI-IGW

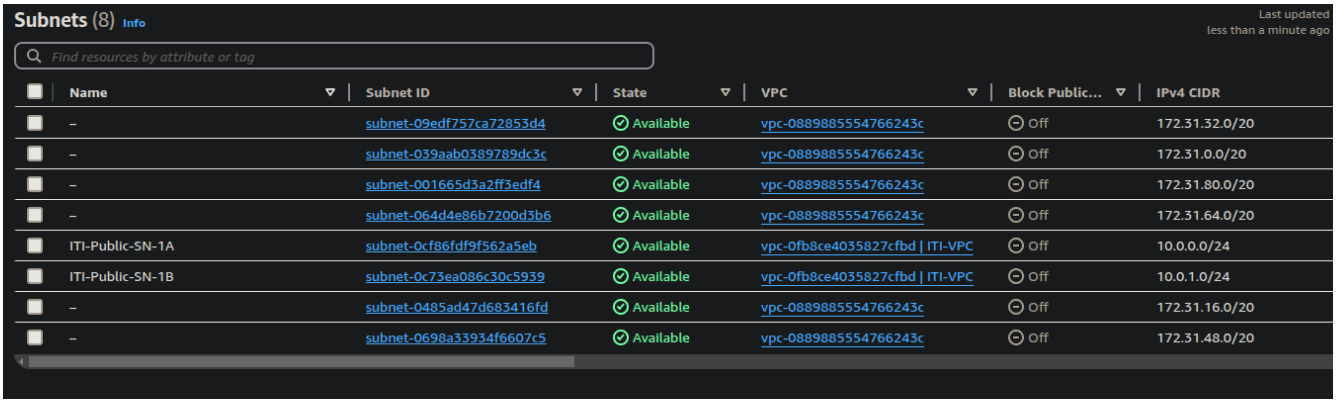
6- attach public route table to subnets



```
resource "aws_subnet" "iti-subnet-1a" {
  vpc_id     = aws_vpc.iti-vpc.id
  cidr_block = "10.0.0.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "ITI-Public-SN-1A"
  }
}

resource "aws_subnet" "iti-subnet-1b" {
  vpc_id     = aws_vpc.iti-vpc.id
  cidr_block = "10.0.1.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "ITI-Public-SN-1B"
  }
}
```



```
  # aws_subnet.iti-subnet-1a will be created
  + resource "aws_subnet" "iti-subnet-1a" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.0.0/24"
      + 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" = "ITI-Public-SN-1A"
        }
      + tags_all                                       = {
          + "Name" = "ITI-Public-SN-1A"
        }
      + vpc_id                                         = "vpc-0fb8ce4035827cfbd"
    }

  # aws_subnet.iti-subnet-1b will be created
  + resource "aws_subnet" "iti-subnet-1b" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1b"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + 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)
```



Subnets (8) Info

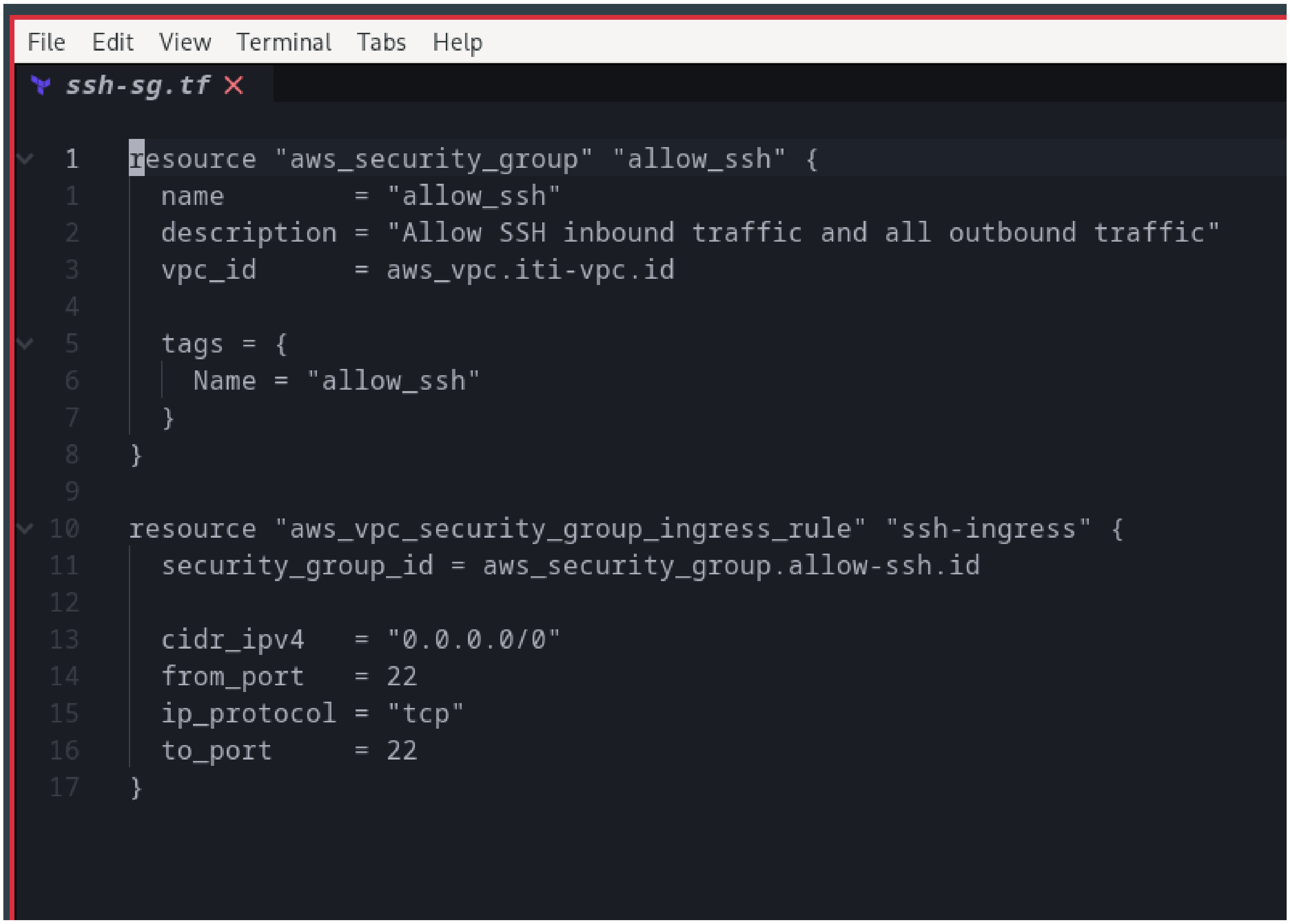
Last updated
less than a minute ago

Find resources by attribute or tag

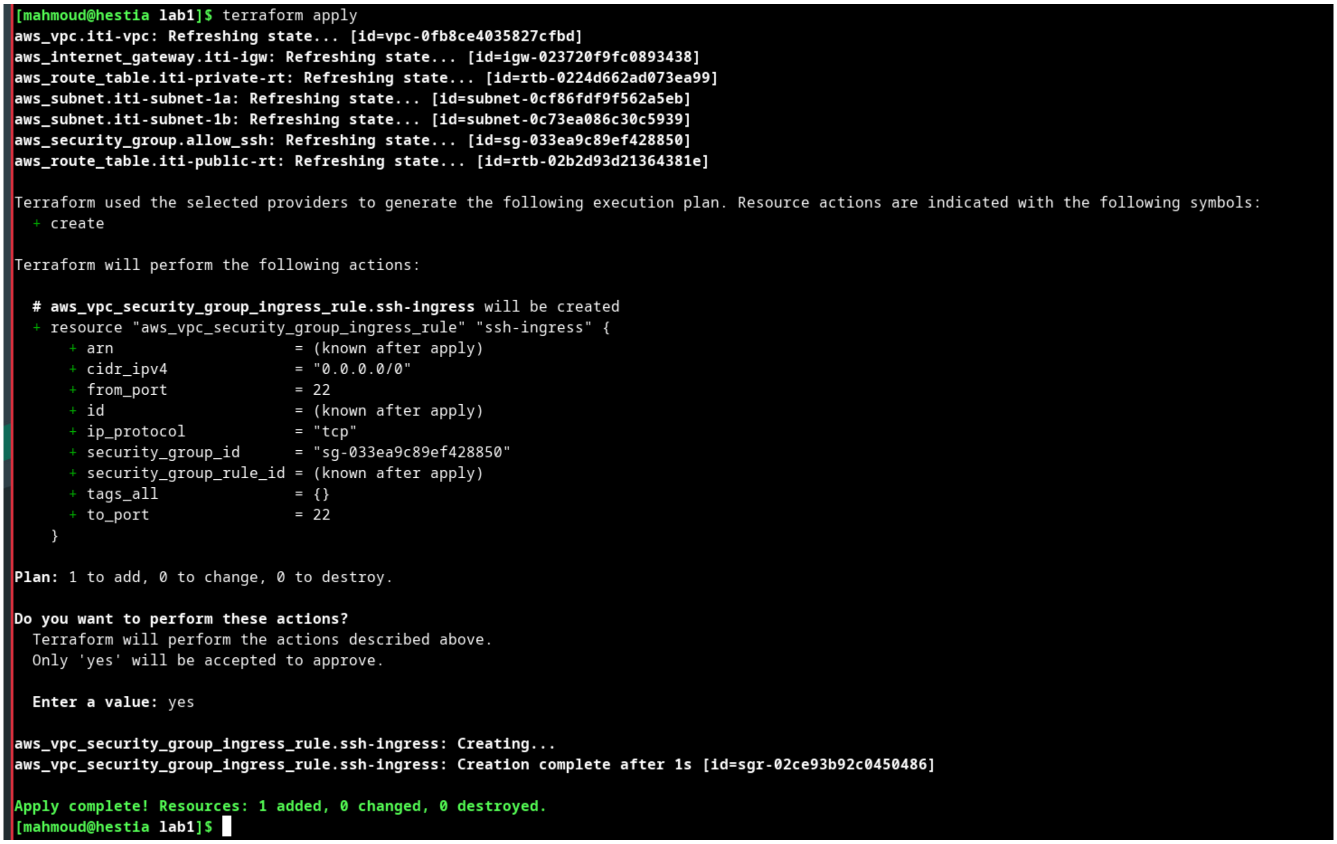
□	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
□	–	subnet-09edf757ca72853d4	✓ Available	vpc-0889885554766243c	⊖ Off	172.31.32.0/20
□	–	subnet-039aab0389789dc3c	✓ Available	vpc-0889885554766243c	⊖ Off	172.31.0.0/20
□	–	subnet-001665d3a2ff3edf4	✓ Available	vpc-0889885554766243c	⊖ Off	172.31.80.0/20
□	–	subnet-064d4e86b7200d3b6	✓ Available	vpc-0889885554766243c	⊖ Off	172.31.64.0/20
□	ITI-Public-SN-1A	subnet-0cf86fdf9f562a5eb	✓ Available	vpc-0fb8ce4035827cfbd | ITI-VPC	⊖ Off	10.0.0.0/24
□	ITI-Public-SN-1B	subnet-0c73ea086c30c5939	✓ Available	vpc-0fb8ce4035827cfbd | ITI-VPC	⊖ Off	10.0.1.0/24
□	–	subnet-0485ad47d683416fd	✓ Available	vpc-0889885554766243c	⊖ Off	172.31.16.0/20
□	–	subnet-0698a33934f6607c5	✓ Available	vpc-0889885554766243c	⊖ Off	172.31.48.0/20

- Compute

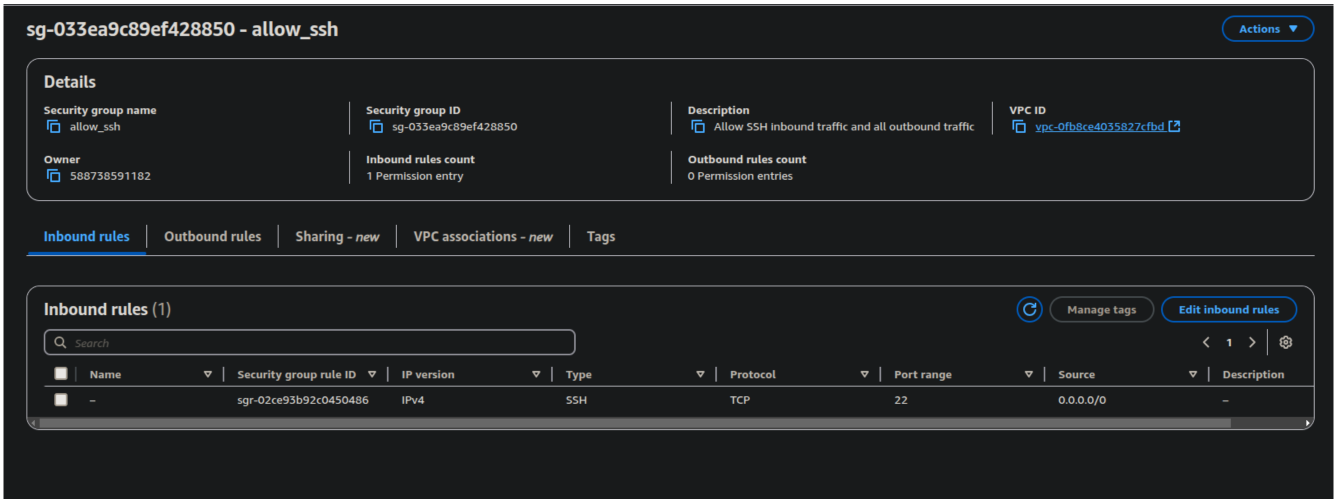
7- create security group which allow ssh from 0.0.0.0/0



```
resource "aws_security_group" "allow_ssh" {
  name        = "allow_ssh"
  description = "Allow SSH inbound traffic and all outbound traffic"
  vpc_id      = aws_vpc.iti-vpc.id

  tags = {
    Name = "allow_ssh"
  }
}

resource "aws_vpc_security_group_ingress_rule" "ssh-ingress" {
  security_group_id = aws_security_group.allow-ssh.id

  cidr_ipv4   = "0.0.0.0/0"
  from_port   = 22
  ip_protocol = "tcp"
  to_port     = 22
}
```



```
[mahmoud@hestia lab1]$ terraform apply
aws_vpc.iti-vpc: Refreshing state... [id=vpc-0fb8ce4035827cfbd]
aws_internet_gateway.iti-igw: Refreshing state... [id=igw-023720f9fc0893438]
aws_route_table.iti-private-rt: Refreshing state... [id=rtb-0224d662ad073ea99]
aws_subnet.iti-subnet-1a: Refreshing state... [id=subnet-0cf86fdf9f562a5eb]
aws_subnet.iti-subnet-1b: Refreshing state... [id=subnet-0c73ea086c30c5939]
aws_security_group.allow_ssh: Refreshing state... [id=sg-033ea9c89ef428850]
aws_route_table.iti-public-rt: Refreshing state... [id=rtb-02b2d93d21364381e]

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_security_group_ingress_rule.ssh-ingress will be created
  + resource "aws_vpc_security_group_ingress_rule" "ssh-ingress" {
      + arn                    = (known after apply)
      + cidr_ipv4              = "0.0.0.0/0"
      + from_port              = 22
      + id                     = (known after apply)
      + ip_protocol            = "tcp"
      + security_group_id      = "sg-033ea9c89ef428850"
      + security_group_rule_id = (known after apply)
      + tags_all               = {}
      + to_port                = 22
    }

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_security_group_ingress_rule.ssh-ingress: Creating...
aws_vpc_security_group_ingress_rule.ssh-ingress: Creation complete after 1s [id=sgr-02ce93b92c0450486]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[mahmoud@hestia lab1]$
```



sg-033ea9c89ef428850 - allow_ssh

Details

Security group name: allow_ssh

Security group ID: sg-033ea9c89ef428850

Description: Allow SSH inbound traffic and all outbound traffic

VPC ID: vpc-0fb8ce4035827cfbd

Owner: 588738591182

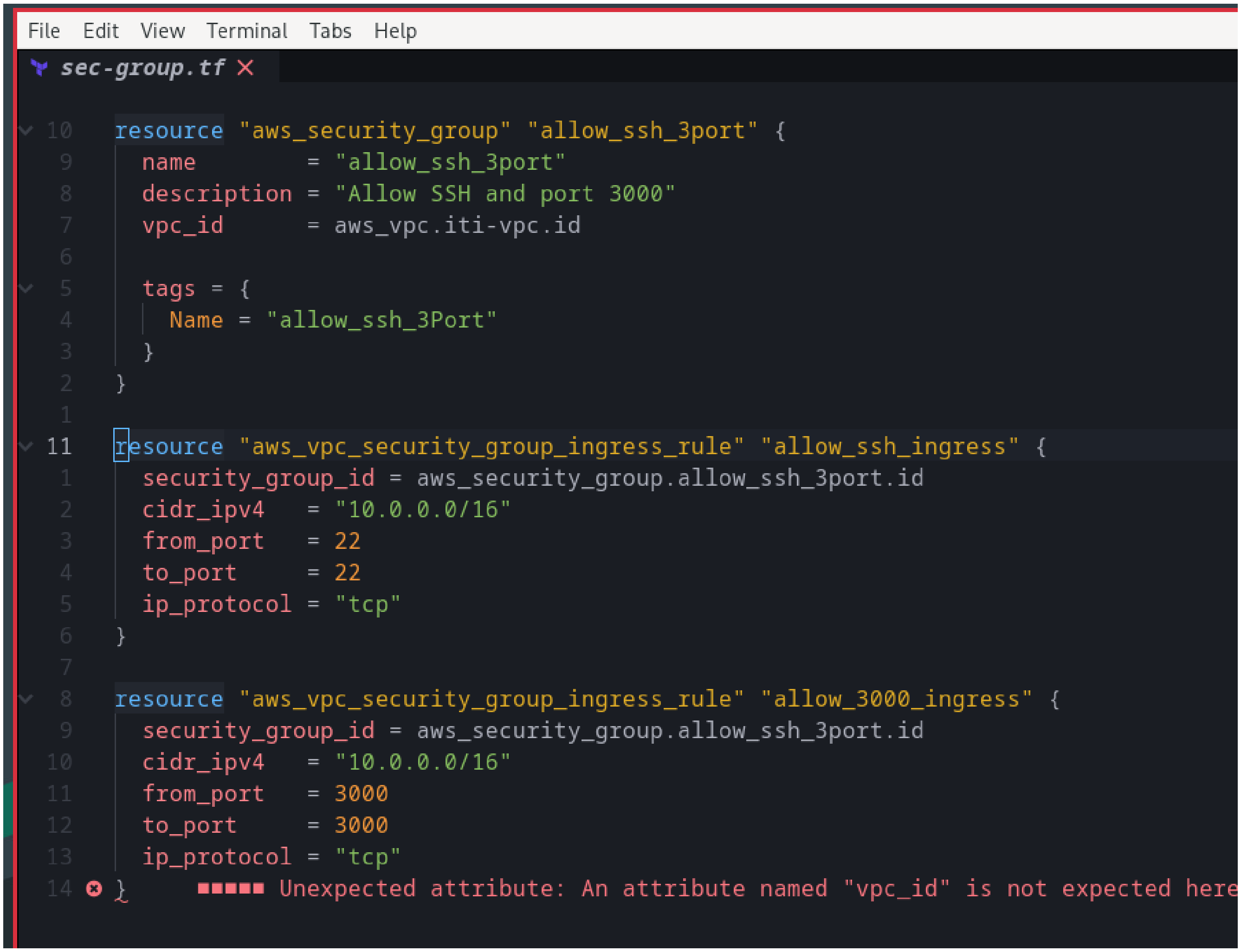
Inbound rules count: 1 Permission entry

Outbound rules count: 0 Permission entries

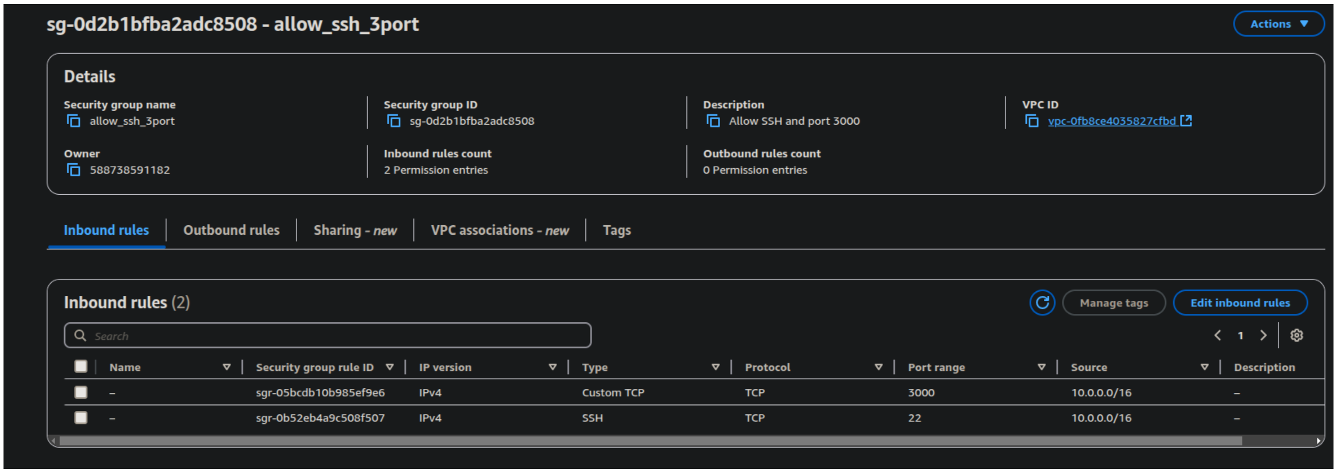
Inbound rules (1)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
□	–	sgr-02ce93b92c0450486	IPv4	SSH	TCP	22	0.0.0.0/0	–

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



```
resource "aws_security_group" "allow_ssh_3port" {
  name        = "allow_ssh_3port"
  description = "Allow SSH and port 3000"
  vpc_id      = aws_vpc.iti-vpc.id

  tags = {
    Name = "allow_ssh_3Port"
  }
}

resource "aws_vpc_security_group_ingress_rule" "allow_ssh_ingress" {
  security_group_id = aws_security_group.allow_ssh_3port.id
  cidr_ipv4   = "10.0.0.0/16"
  from_port   = 22
  to_port     = 22
  ip_protocol = "tcp"
}

resource "aws_vpc_security_group_ingress_rule" "allow_3000_ingress" {
  security_group_id = aws_security_group.allow_ssh_3port.id
  cidr_ipv4   = "10.0.0.0/16"
  from_port   = 3000
  to_port     = 3000
  ip_protocol = "tcp"
}     ■■■■■ Unexpected attribute: An attribute named "vpc_id" is not expected here
```



sg-0d2b1bfba2adc8508 - allow_ssh_3port

Details

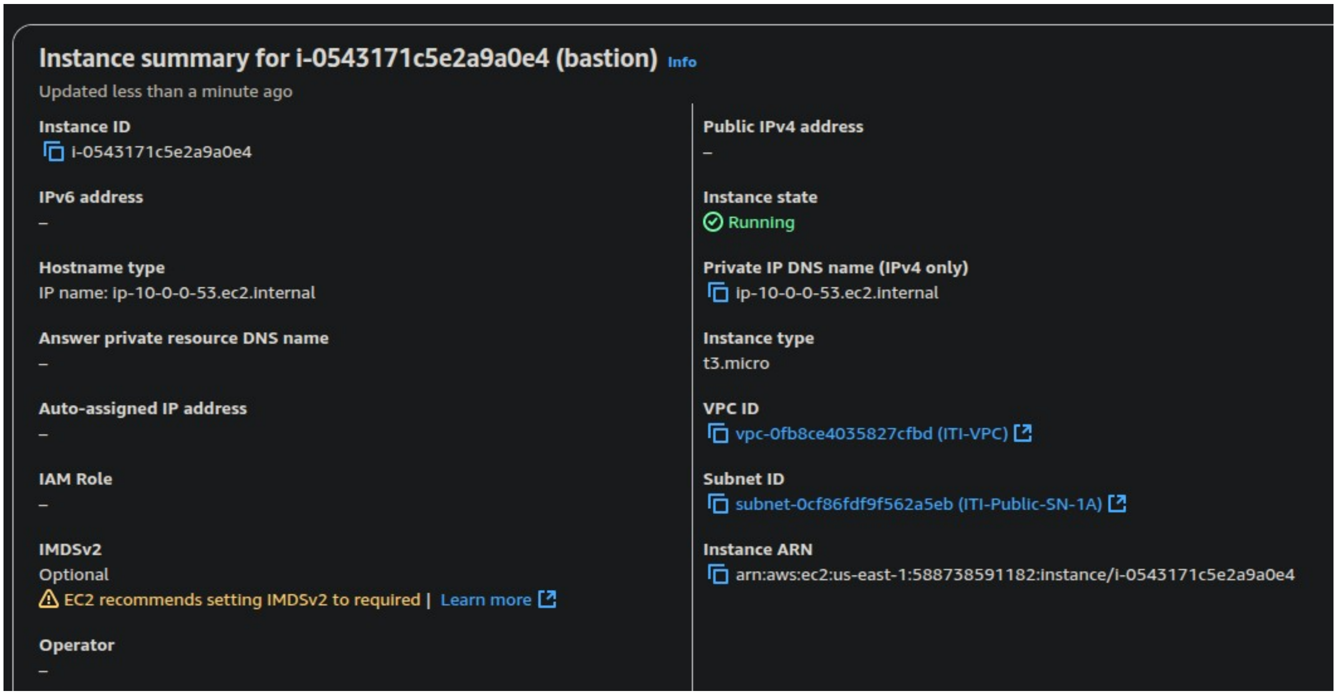
Security group name	Security group ID	Description	VPC ID
allow_ssh_3port	sg-0d2b1bfba2adc8508	Allow SSH and port 3000	vpc-0fb8ce4035827cfbd

Owner	Inbound rules count	Outbound rules count
588738591182	2 Permission entries	0 Permission entries

Inbound rules (2)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
□	–	sgr-05bcdb10b985ef9e6	IPv4	Custom TCP	TCP	3000	10.0.0.0/16	–
□	–	sgr-0b52eb4a9c508f507	IPv4	SSH	TCP	22	10.0.0.0/16	–

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



Instance summary for i-0543171c5e2a9a0e4 (bastion) Info

Updated less than a minute ago

Instance ID
i-0543171c5e2a9a0e4

IPv6 address
–

Hostname type
IP name: ip-10-0-0-53.ec2.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Optional
⚠ EC2 recommends setting IMDSv2 to required | Learn more

Operator
–

Public IPv4 address
–

Instance state
✓ Running

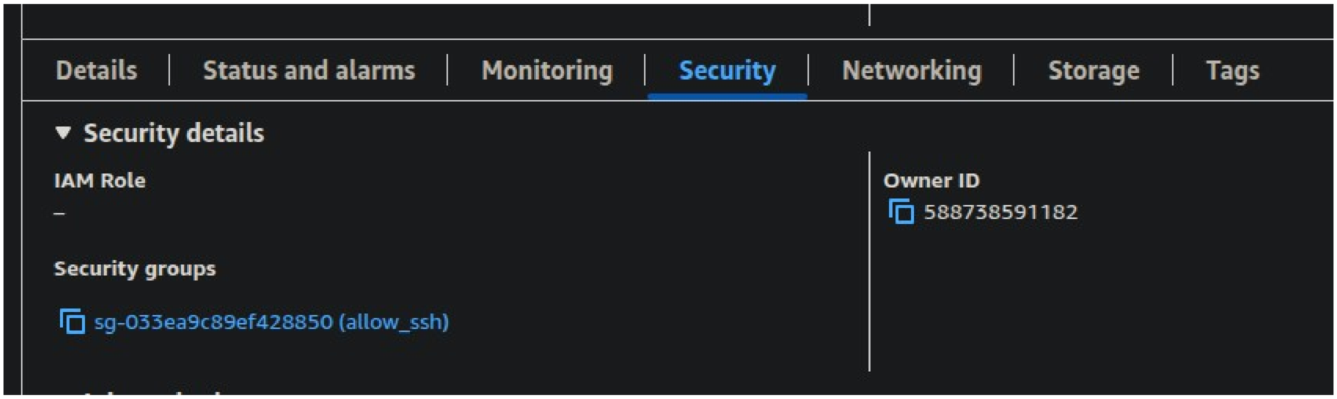
Private IP DNS name (IPv4 only)
ip-10-0-0-53.ec2.internal

Instance type
t3.micro

VPC ID
vpc-0fb8ce4035827cfbd (ITI-VPC)

Subnet ID
subnet-0cf86fdf9f562a5eb (ITI-Public-SN-1A)

Instance ARN
arn:aws:ec2:us-east-1:588738591182:instance/i-0543171c5e2a9a0e4

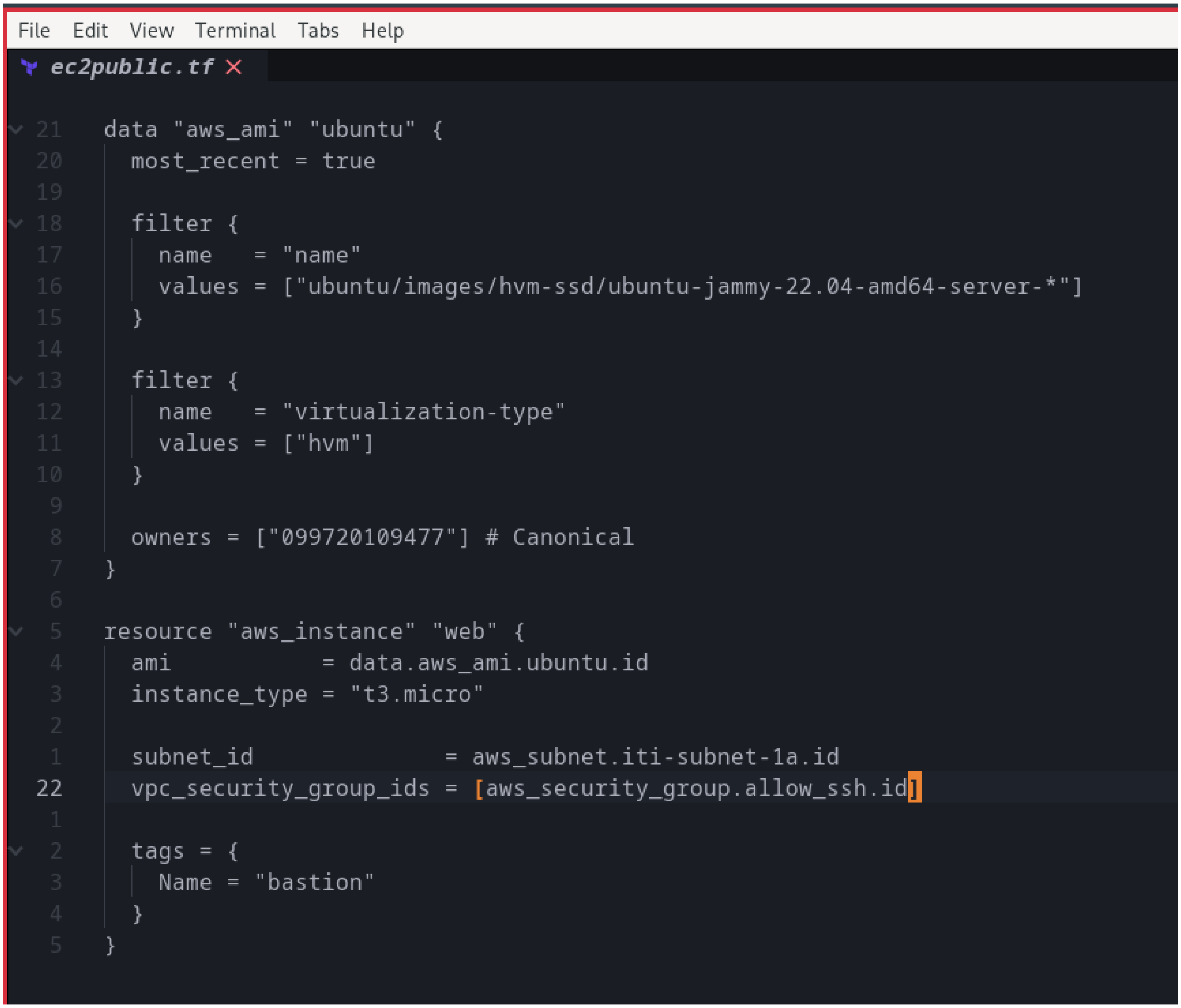


▼ Security details

IAM Role
–

Security groups
sg-033ea9c89ef428850 (allow_ssh)

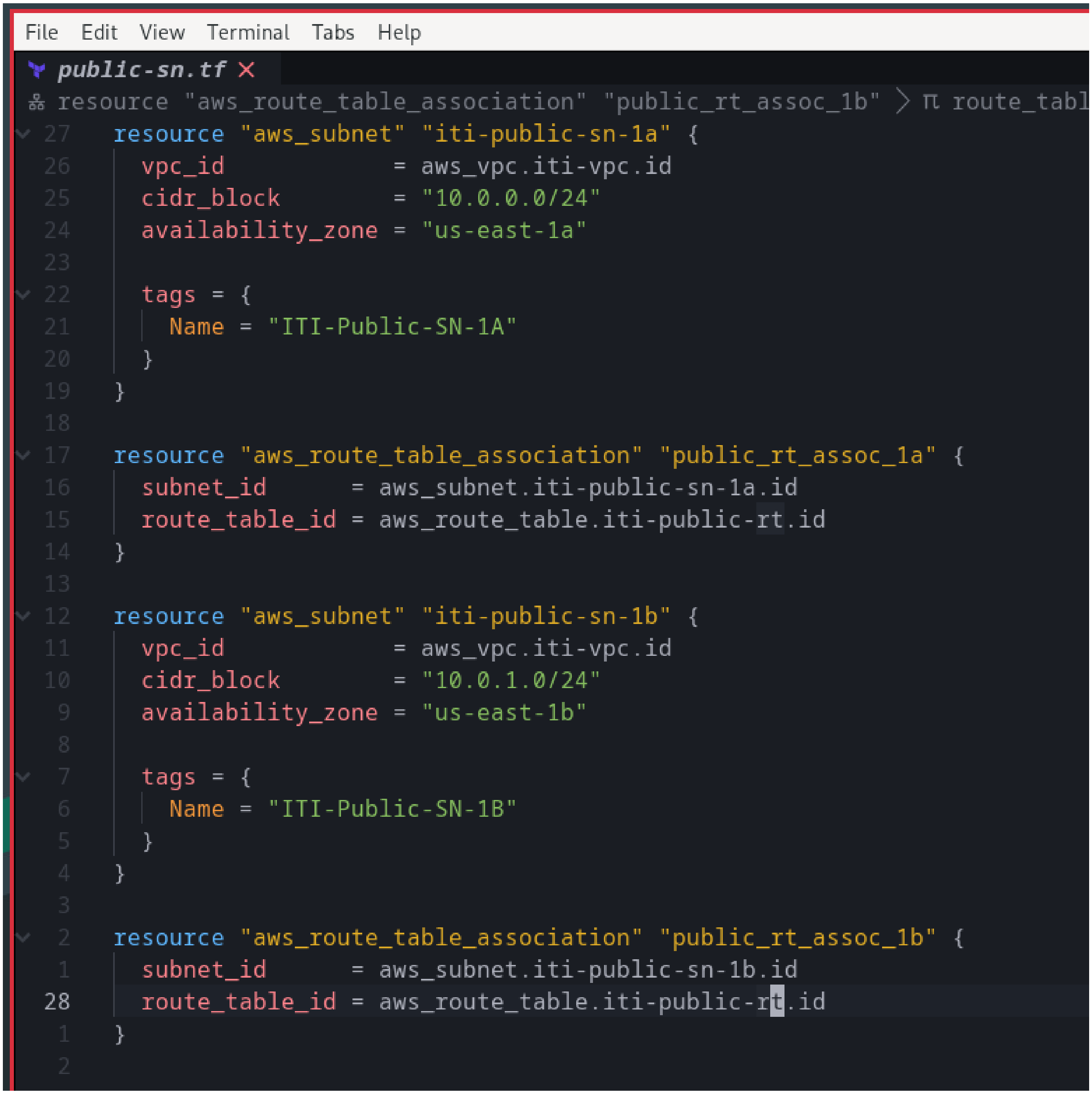
Owner ID
588738591182



```
data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami           = data.aws_ami.ubuntu.id
  instance_type = "t3.micro"

  subnet_id              = aws_subnet.iti-subnet-1a.id
  vpc_security_group_ids = [aws_security_group.allow_ssh.id]

  tags = {
    Name = "bastion"
  }
}
```

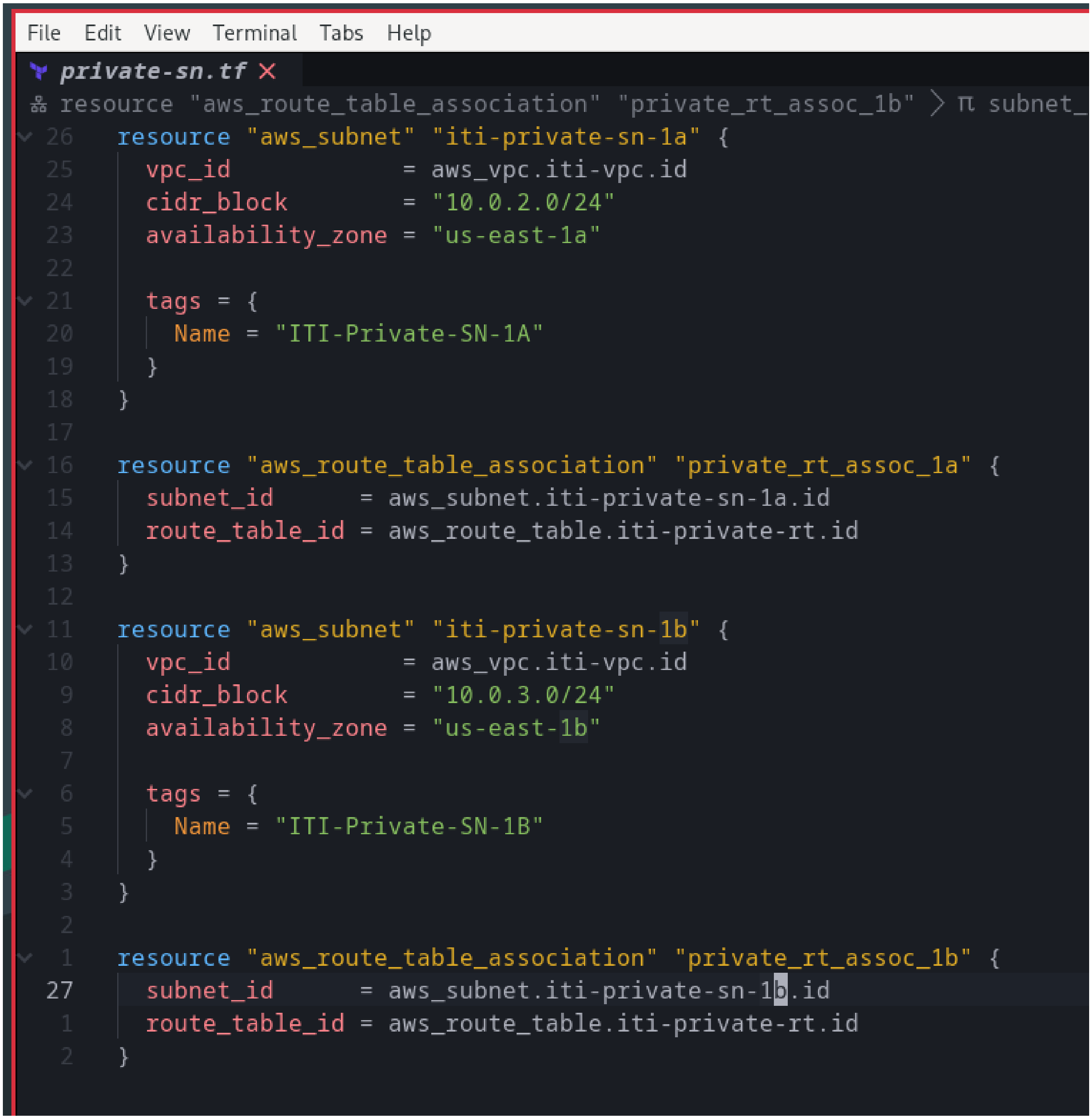
10- create ec2(application) private subnet with security group from 8

fixing the subnets to route to the correct route table


File
Edit
View
Terminal
Tabs
Help
public-sn.tf
resource "aws_route_table_association" "public_rt_assoc_1b" > route_tabl

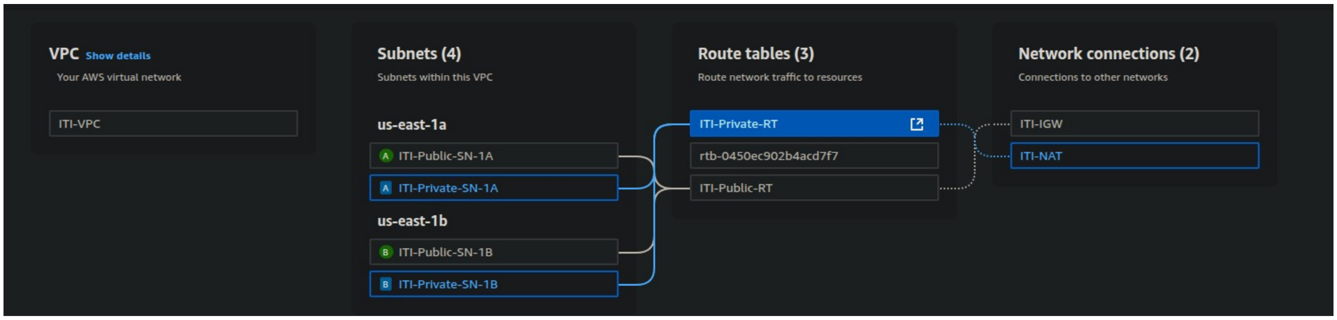
```
resource "aws_subnet" "iti-public-sn-1a" {
  vpc_id            = aws_vpc.iti-vpc.id
  cidr_block        = "10.0.0.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "ITI-Public-SN-1A"
  }
}

resource "aws_route_table_association" "public_rt_assoc_1a" {
  subnet_id      = aws_subnet.iti-public-sn-1a.id
  route_table_id = aws_route_table.iti-public-rt.id
}

resource "aws_subnet" "iti-public-sn-1b" {
  vpc_id            = aws_vpc.iti-vpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "ITI-Public-SN-1B"
  }
}

resource "aws_route_table_association" "public_rt_assoc_1b" {
  subnet_id      = aws_subnet.iti-public-sn-1b.id
  route_table_id = aws_route_table.iti-public-rt.id
}
```

and also fixing the private one

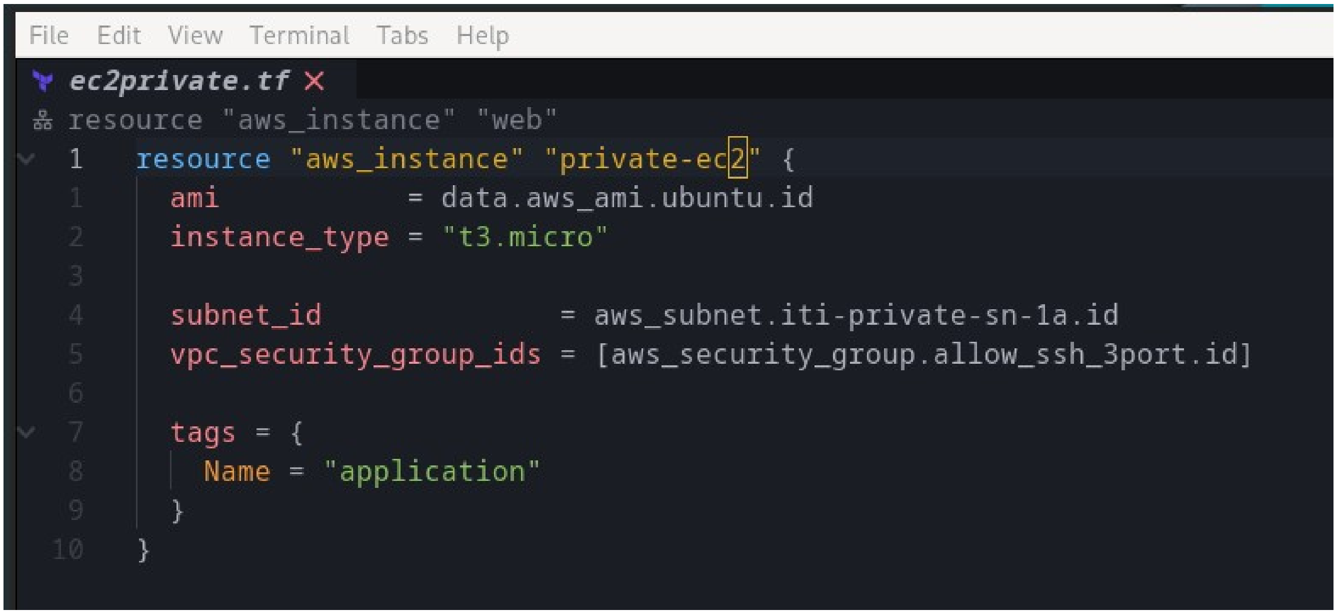

File
Edit
View
Terminal
Tabs
Help
private-sn.tf
resource "aws_route_table_association" "private_rt_assoc_1b" > π subnet_

```
resource "aws_subnet" "iti-private-sn-1a" {
  vpc_id            = aws_vpc.iti-vpc.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "ITI-Private-SN-1A"
  }
}

resource "aws_route_table_association" "private_rt_assoc_1a" {
  subnet_id      = aws_subnet.iti-private-sn-1a.id
  route_table_id = aws_route_table.iti-private-rt.id
}

resource "aws_subnet" "iti-private-sn-1b" {
  vpc_id            = aws_vpc.iti-vpc.id
  cidr_block        = "10.0.3.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "ITI-Private-SN-1B"
  }
}

resource "aws_route_table_association" "private_rt_assoc_1b" {
  subnet_id      = aws_subnet.iti-private-sn-1b.id
  route_table_id = aws_route_table.iti-private-rt.id
}
```

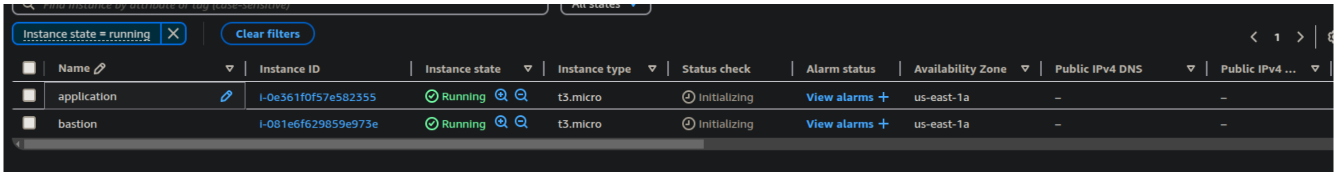
the ending result of the vpc



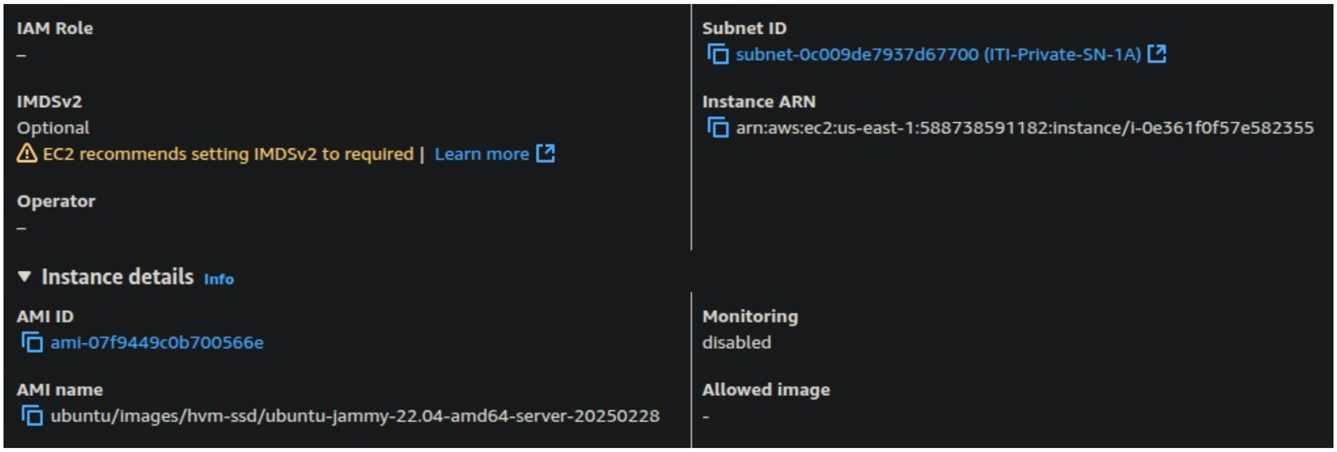
the ec2 terraform file



```
resource "aws_instance" "private-ec2" {
  ami           = data.aws_ami.ubuntu.id
  instance_type = "t3.micro"

  subnet_id              = aws_subnet.iti-private-sn-1a.id
  vpc_security_group_ids = [aws_security_group.allow_ssh_3port.id]

  tags = {
    Name = "application"
  }
}
```



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
application	i-0e361f0f57e582355	Running	t3.micro	Initializing	View alarms +	us-east-1a	–	–
bastion	i-081e6f629859e973e	Running	t3.micro	Initializing	View alarms +	us-east-1a	–	–



IAM Role
–

IMDSv2
Optional
⚠ EC2 recommends setting IMDSv2 to required | Learn more

Operator
–

Subnet ID
subnet-0c009de7937d67700 (ITI-Private-SN-1A)

Instance ARN
arn:aws:ec2:us-east-1:588738591182:instance/i-0e361f0f57e582355

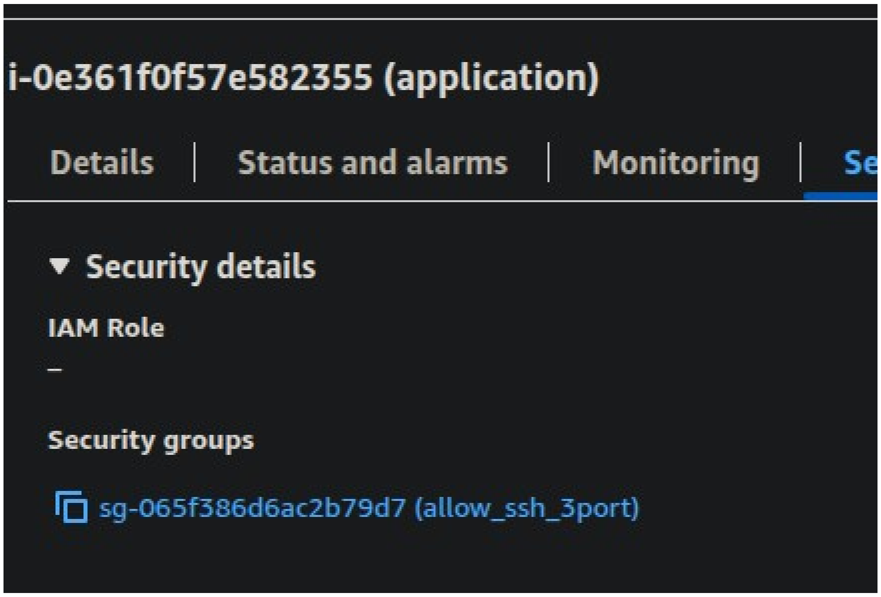
▼ Instance details Info

AMI ID
ami-07f9449c0b700566e

AMI name
ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20250228

Monitoring
disabled

Allowed image
-

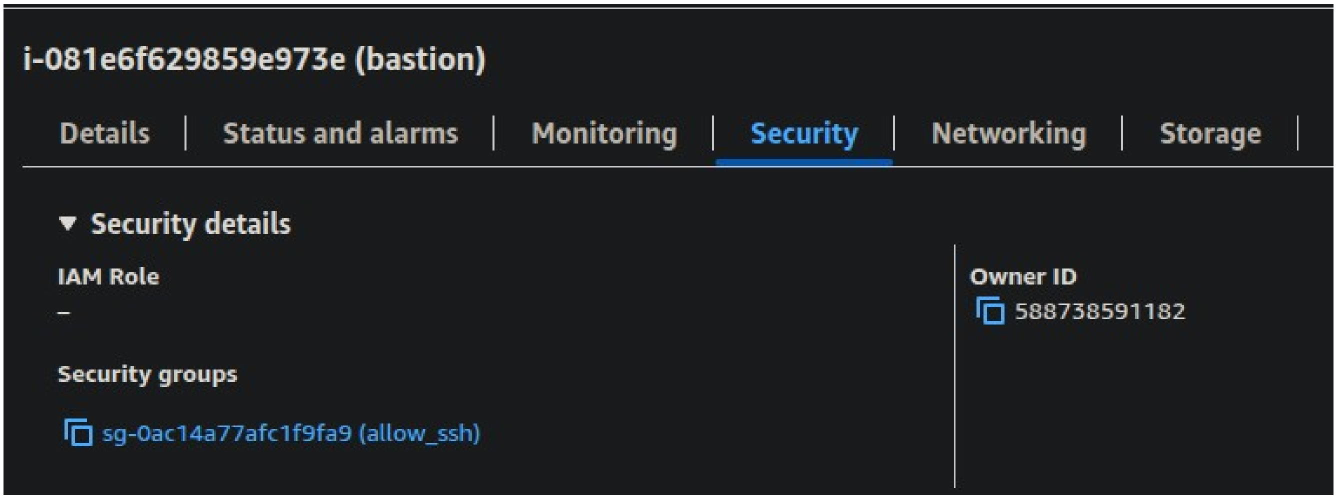


i-0e361f0f57e582355 (application)

▼ Security details

IAM Role
–

Security groups
sg-065f386d6ac2b79d7 (allow_ssh_3port)



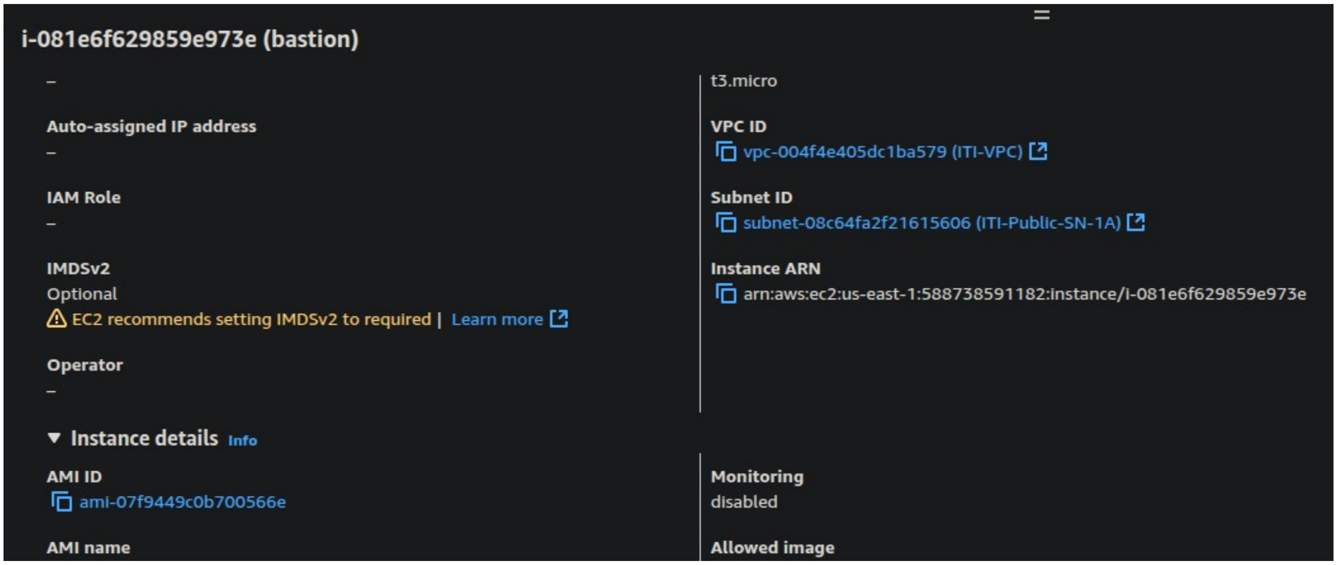
i-081e6f629859e973e (bastion)

▼ Security details

IAM Role
–

Security groups
sg-0ac14a77afc1f9fa9 (allow_ssh)

Owner ID
588738591182



i-081e6f629859e973e (bastion)

–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Optional
⚠ EC2 recommends setting IMDSv2 to required | Learn more

Operator
–

t3.micro

VPC ID
vpc-004f4e405dc1ba579 (ITI-VPC)

Subnet ID
subnet-08c64fa2f21615606 (ITI-Public-SN-1A)

Instance ARN
arn:aws:ec2:us-east-1:588738591182:instance/i-081e6f629859e973e

▼ **Instance details** Info

AMI ID
ami-07f9449c0b700566e

AMI name

Monitoring
disabled

Allowed image