TERRAFORM ASSIGNMENT -4

Name: Vikram
Assignment: Create VPC & Deploy EC2 using Terraform

Problem Statement

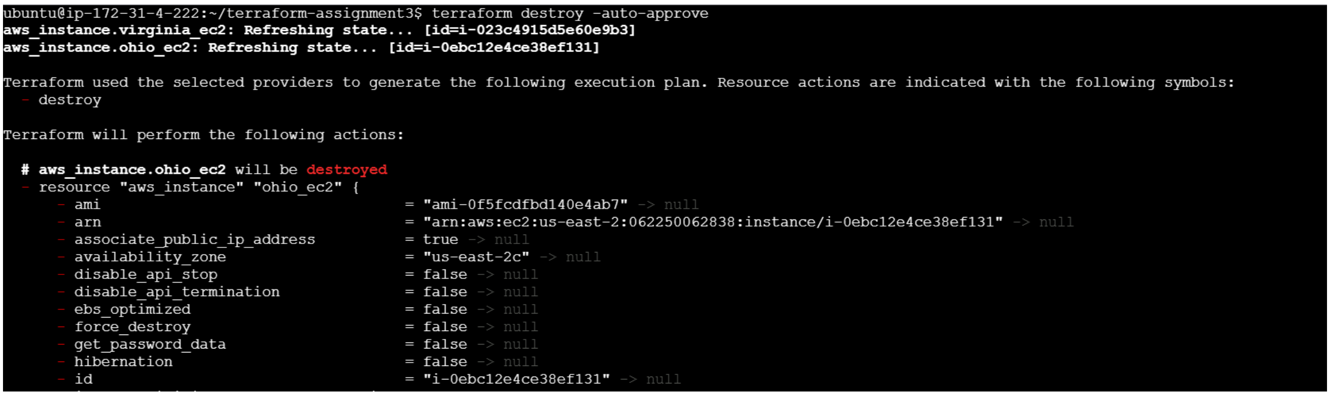
Perform the following using Terraform:

1. Destroy the previous deployment
2. Create a VPC with all required components
3. Deploy an EC2 instance inside the VPC

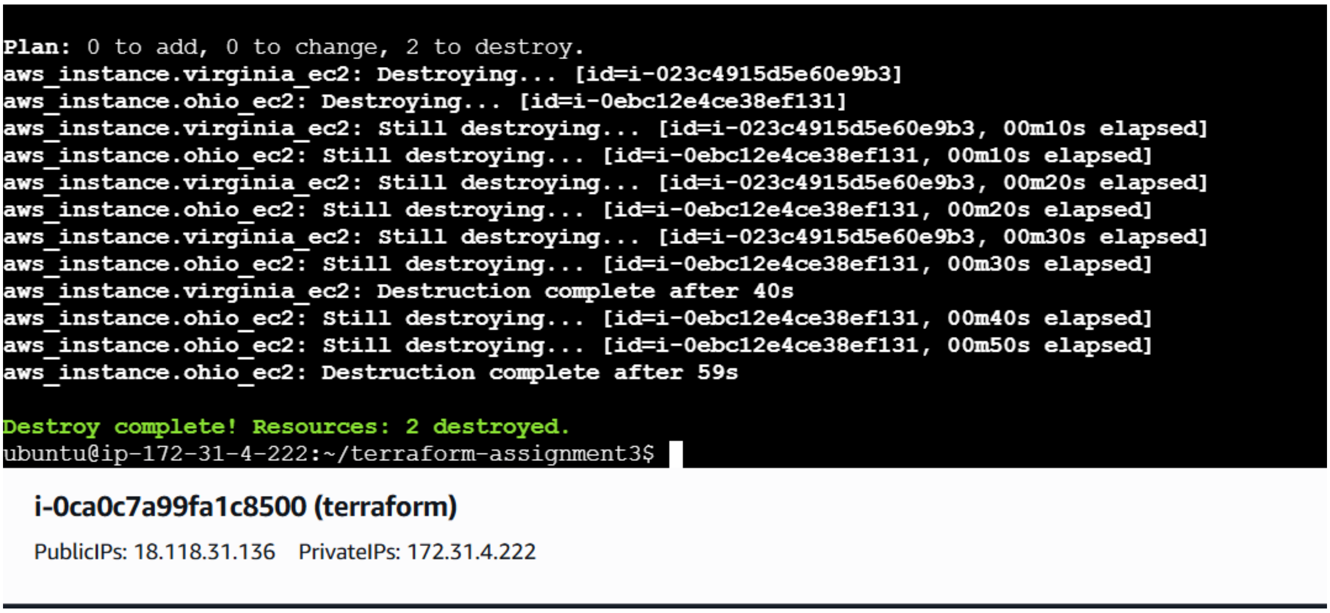
Steps Performed

Step 1 — Destroy Previous Deployment

terraform destroy



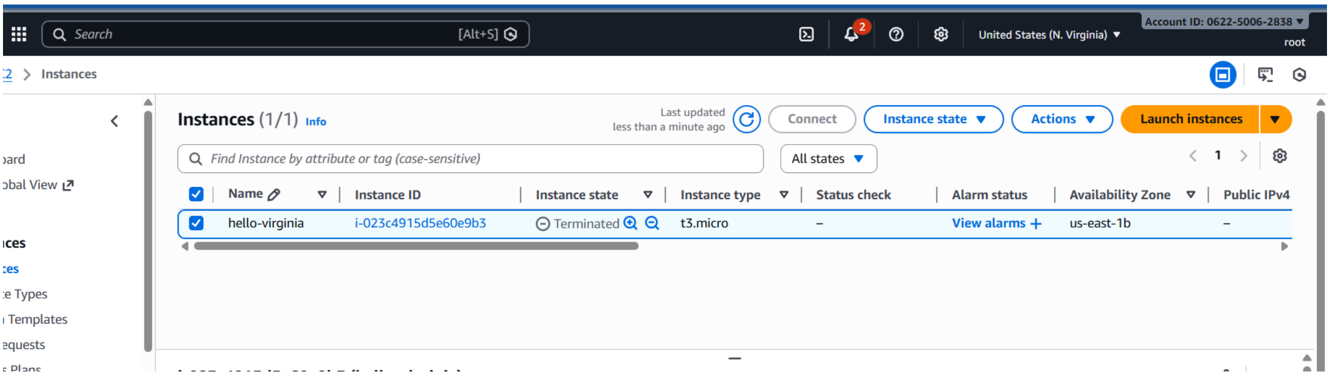
```
ubuntu@ip-172-31-4-222:~/terraform-assignment3$ terraform destroy -auto-approve
aws_instance.virginia_ec2: Refreshing state... [id=i-023c4915d5e60e9b3]
aws_instance.ohio_ec2: Refreshing state... [id=i-0ebc12e4ce38ef131]

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

Terraform will perform the following actions:

  # aws_instance.ohio_ec2 will be destroyed
  - resource "aws_instance" "ohio_ec2" {
      - ami                                  = "ami-0f5fcdfbd140e4ab7" -> null
      - arn                                  = "arn:aws:ec2:us-east-2:062250062838:instance/i-0ebc12e4ce38ef131" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-2c" -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - force_destroy                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-0ebc12e4ce38ef131" -> null
```



```
Plan: 0 to add, 0 to change, 2 to destroy.
aws_instance.virginia_ec2: Destroying... [id=i-023c4915d5e60e9b3]
aws_instance.ohio_ec2: Destroying... [id=i-0ebc12e4ce38ef131]
aws_instance.virginia_ec2: Still destroying... [id=i-023c4915d5e60e9b3, 00m10s elapsed]
aws_instance.ohio_ec2: Still destroying... [id=i-0ebc12e4ce38ef131, 00m10s elapsed]
aws_instance.virginia_ec2: Still destroying... [id=i-023c4915d5e60e9b3, 00m20s elapsed]
aws_instance.ohio_ec2: Still destroying... [id=i-0ebc12e4ce38ef131, 00m20s elapsed]
aws_instance.virginia_ec2: Still destroying... [id=i-023c4915d5e60e9b3, 00m30s elapsed]
aws_instance.ohio_ec2: Still destroying... [id=i-0ebc12e4ce38ef131, 00m30s elapsed]
aws_instance.virginia_ec2: Destruction complete after 40s
aws_instance.ohio_ec2: Still destroying... [id=i-0ebc12e4ce38ef131, 00m40s elapsed]
aws_instance.ohio_ec2: Still destroying... [id=i-0ebc12e4ce38ef131, 00m50s elapsed]
aws_instance.ohio_ec2: Destruction complete after 59s

Destroy complete! Resources: 2 destroyed.
ubuntu@ip-172-31-4-222:~/terraform-assignment3$
```

i-0ca0c7a99fa1c8500 (terraform)

PublicIPs: 18.118.31.136   PrivateIPs: 172.31.4.222

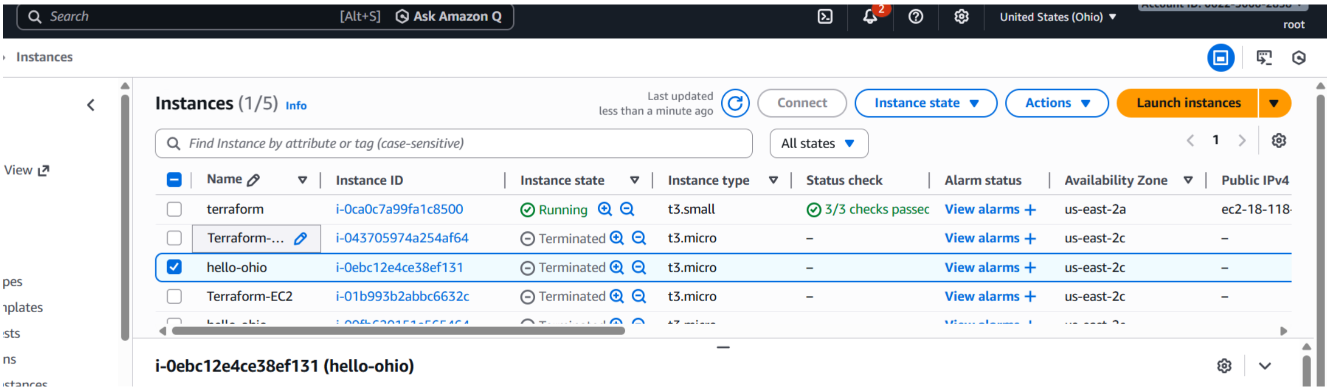


Search
[Alt+S]
United States (N. Virginia) ▼
Account ID: 0622-5006-2838 ▼
root
Instances
Connect
Instance state ▼
Actions ▼
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states ▼

Instances (1/1) Info

Last updated less than a minute ago

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	hello-virginia	i-023c4915d5e60e9b3	Terminated	t3.micro	–	View alarms +	us-east-1b	–



Instances (1/5) Info

Last updated less than a minute ago

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	terraform	i-0ca0c7a99fa1c8500	Running	t3.small	3/3 checks passec	View alarms +	us-east-2a	ec2-18-118-
□	Terraform-...	i-043705974a254af64	Terminated	t3.micro	–	View alarms +	us-east-2c	–
☑	hello-ohio	i-0ebc12e4ce38ef131	Terminated	t3.micro	–	View alarms +	us-east-2c	–
□	Terraform-EC2	i-01b993b2abbc6632c	Terminated	t3.micro	–	View alarms +	us-east-2c	–

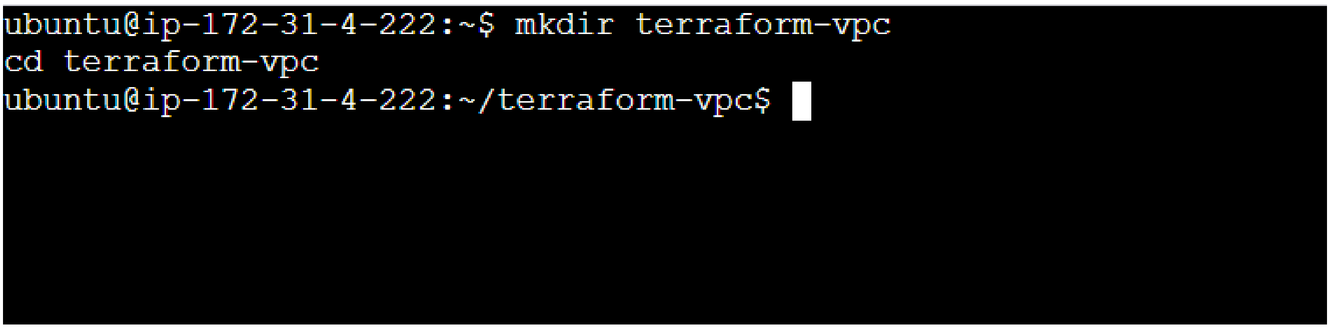
i-0ebc12e4ce38ef131 (hello-ohio)

Step 2 — Create Terraform Project

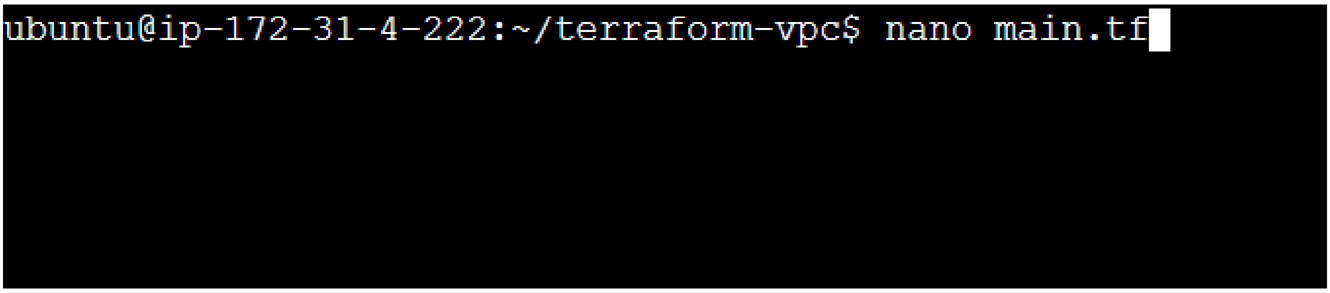
mkdir terraform-vpc

cd terraform-vpc

nano main.tf



```
ubuntu@ip-172-31-4-222:~$ mkdir terraform-vpc
cd terraform-vpc
ubuntu@ip-172-31-4-222:~/terraform-vpc$
```



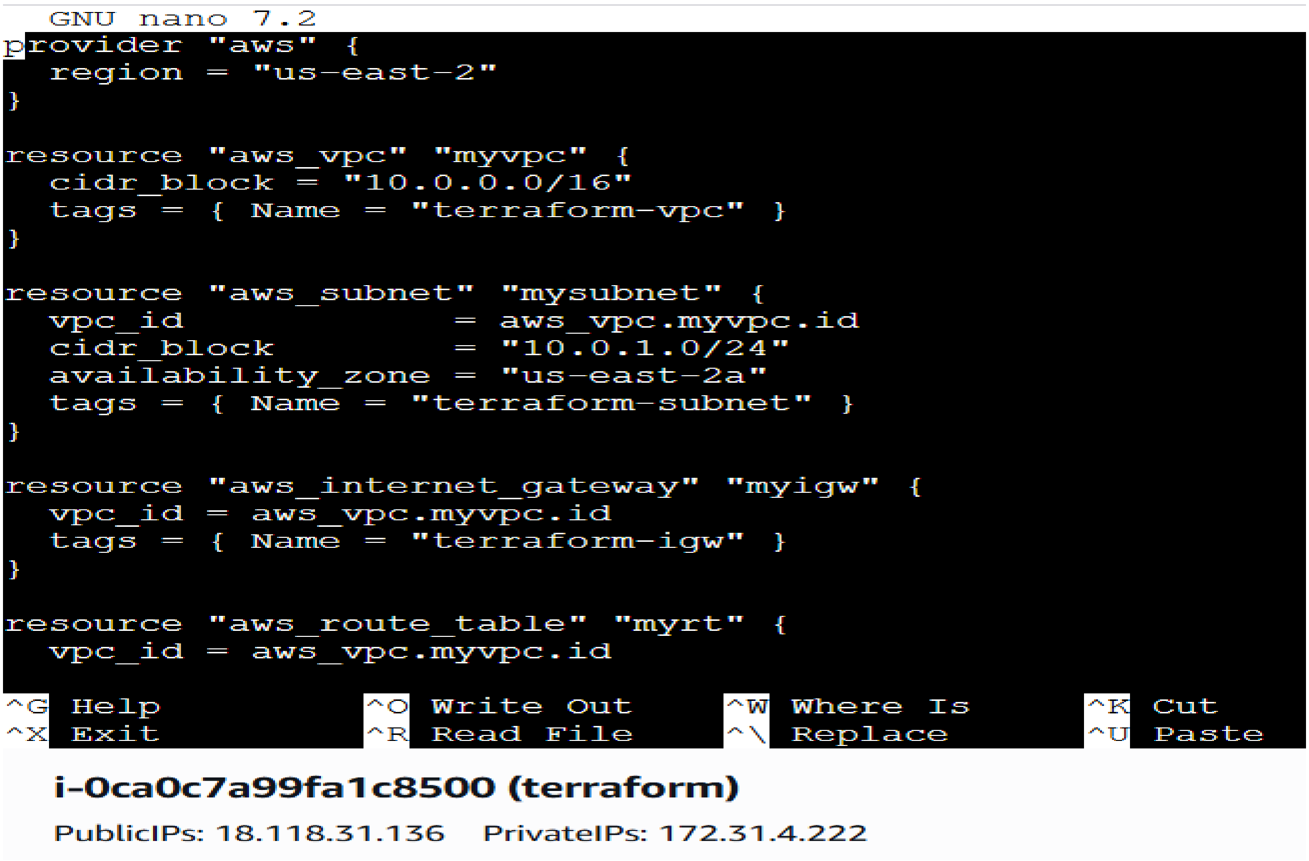
```
ubuntu@ip-172-31-4-222:~/terraform-vpc$ nano main.tf
```

Step 3 — Terraform Configuration (main.tf)

```
provider "aws" {
  region = "us-east-2"
}

resource "aws_vpc" "myvpc" {
  cidr_block = "10.0.0.0/16"
  tags = { Name = "terraform-vpc" }
}

resource "aws_subnet" "mysubnet" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-2a"
  tags = { Name = "terraform-subnet" }
}

resource "aws_internet_gateway" "myigw" {
  vpc_id = aws_vpc.myvpc.id
  tags = { Name = "terraform-igw" }
}

resource "aws_route_table" "myrt" {
  vpc_id = aws_vpc.myvpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myigw.id
  }
}

resource "aws_route_table_association" "rta" {
  subnet_id      = aws_subnet.mysubnet.id
  route_table_id = aws_route_table.myrt.id
}
```

```
resource "aws_security_group" "mysg" {
  vpc_id = aws_vpc.myvpc.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   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
resource "aws_instance" "myec2" {
  ami                         = "ami-0f5fcdfbd140e4ab7"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.mysubnet.id
  vpc_security_group_ids      = [aws_security_group.mysg.id]
  associate_public_ip_address = true
  tags = { Name = "terraform-ec2" }
}
```

Save and exit



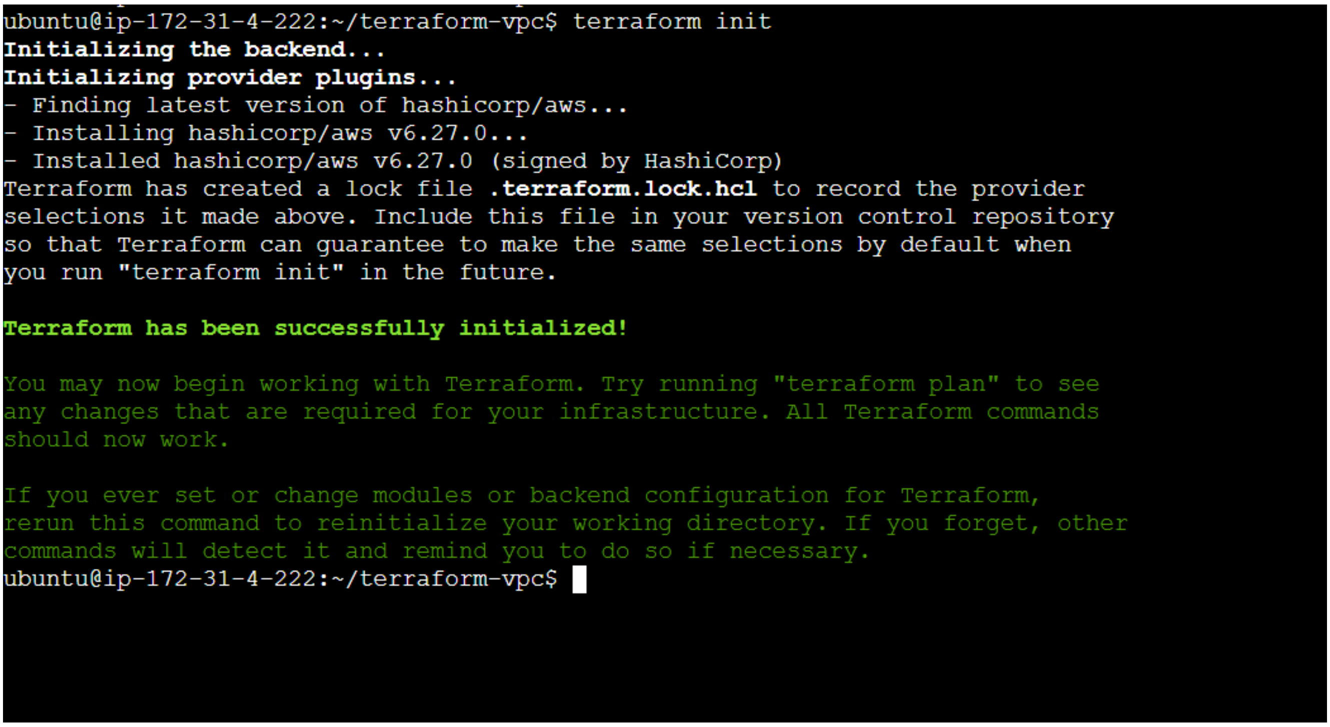
```
provider "aws" {
  region = "us-east-2"
}

resource "aws_vpc" "myvpc" {
  cidr_block = "10.0.0.0/16"
  tags = { Name = "terraform-vpc" }
}

resource "aws_subnet" "mysubnet" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-2a"
  tags = { Name = "terraform-subnet" }
}

resource "aws_internet_gateway" "myigw" {
  vpc_id = aws_vpc.myvpc.id
  tags = { Name = "terraform-igw" }
}

resource "aws_route_table" "myrt" {
  vpc_id = aws_vpc.myvpc.id
```

i-0ca0c7a99fa1c8500 (terraform)

PublicIPs: 18.118.31.136   PrivateIPs: 172.31.4.222

Step 4 — Initialize Terraform

terraform init



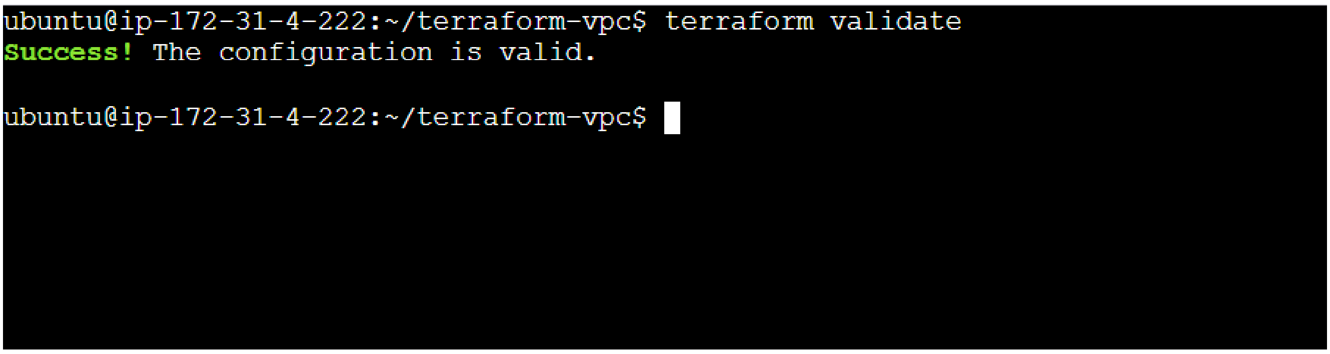
```
ubuntu@ip-172-31-4-222:~/terraform-vpc$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl 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.
ubuntu@ip-172-31-4-222:~/terraform-vpc$
```

Step 5 — Validate & Apply

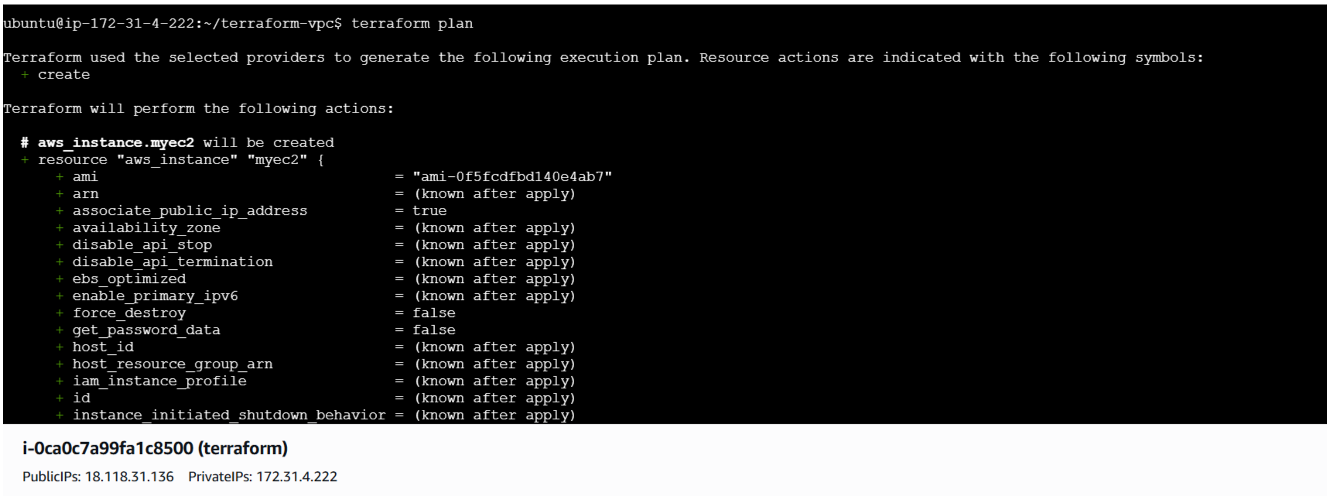
terraform validate

terraform apply

Type yes when prompted.



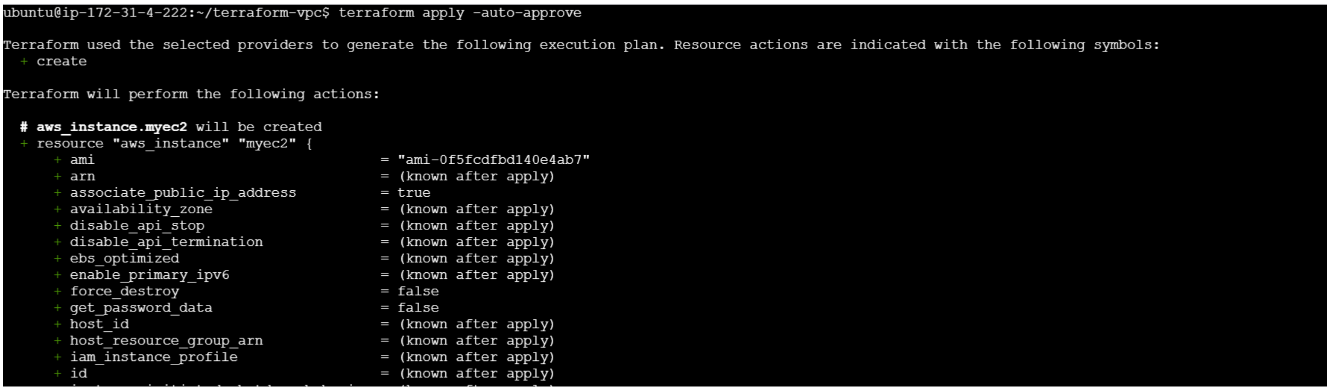
```
ubuntu@ip-172-31-4-222:~/terraform-vpc$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-4-222:~/terraform-vpc$
```



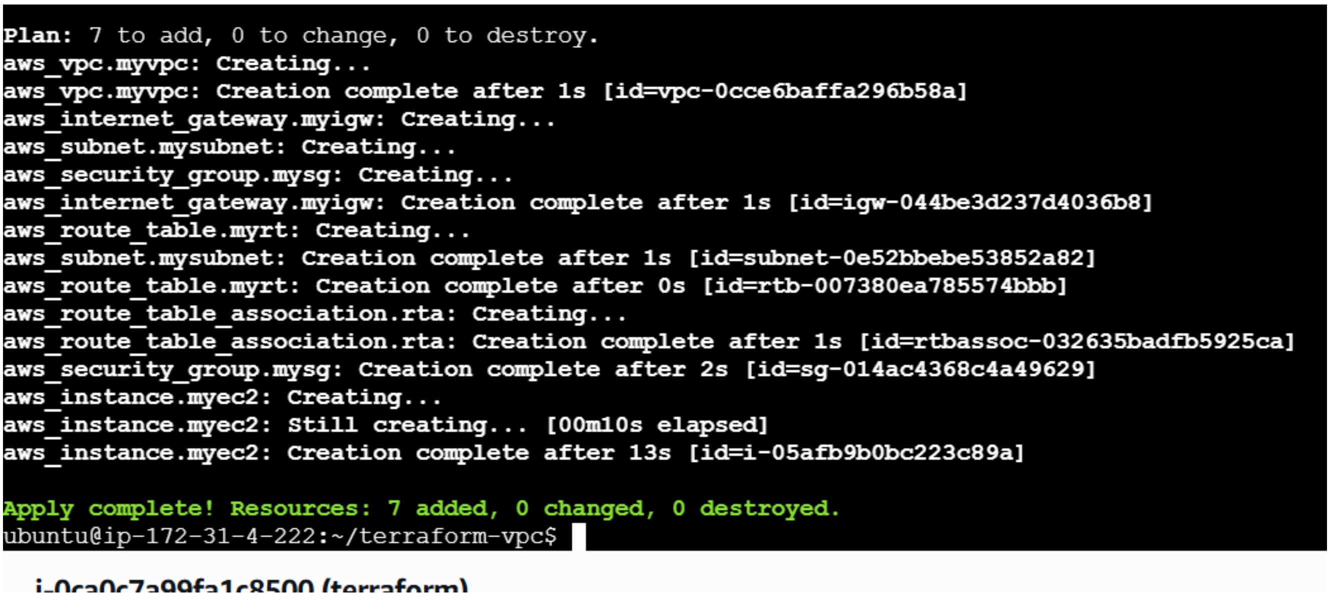
```
ubuntu@ip-172-31-4-222:~/terraform-vpc$ terraform plan

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_instance.myec2 will be created
  + resource "aws_instance" "myec2" {
      + ami                                  = "ami-0f5fcdfbd140e4ab7"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + 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)
```

i-0ca0c7a99fa1c8500 (terraform)

PublicIPs: 18.118.31.136 PrivateIPs: 172.31.4.222



```
ubuntu@ip-172-31-4-222:~/terraform-vpc$ terraform apply -auto-approve

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

Terraform will perform the following actions:

  # aws_instance.myec2 will be created
  + resource "aws_instance" "myec2" {
      + ami                                  = "ami-0f5fcdfbd140e4ab7"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + 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)
```



```
Plan: 7 to add, 0 to change, 0 to destroy.
aws_vpc.myvpc: Creating...
aws_vpc.myvpc: Creation complete after 1s [id=vpc-0cce6baffa296b58a]
aws_internet_gateway.myigw: Creating...
aws_subnet.mysubnet: Creating...
aws_security_group.mysg: Creating...
aws_internet_gateway.myigw: Creation complete after 1s [id=igw-044be3d237d4036b8]
aws_route_table.myrt: Creating...
aws_subnet.mysubnet: Creation complete after 1s [id=subnet-0e52bbebe53852a82]
aws_route_table.myrt: Creation complete after 0s [id=rtb-007380ea785574bbb]
aws_route_table_association.rta: Creating...
aws_route_table_association.rta: Creation complete after 1s [id=rtbassoc-032635badfb5925ca]
aws_security_group.mysg: Creation complete after 2s [id=sg-014ac4368c4a49629]
aws_instance.myec2: Creating...
aws_instance.myec2: Still creating... [00m10s elapsed]
aws_instance.myec2: Creation complete after 13s [id=i-05afb9b0bc223c89a]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-4-222:~/terraform-vpc$
```

i-0ca0c7a99fa1c8500 (terraform)

Verification

From AWS Console:

✓ New VPC created

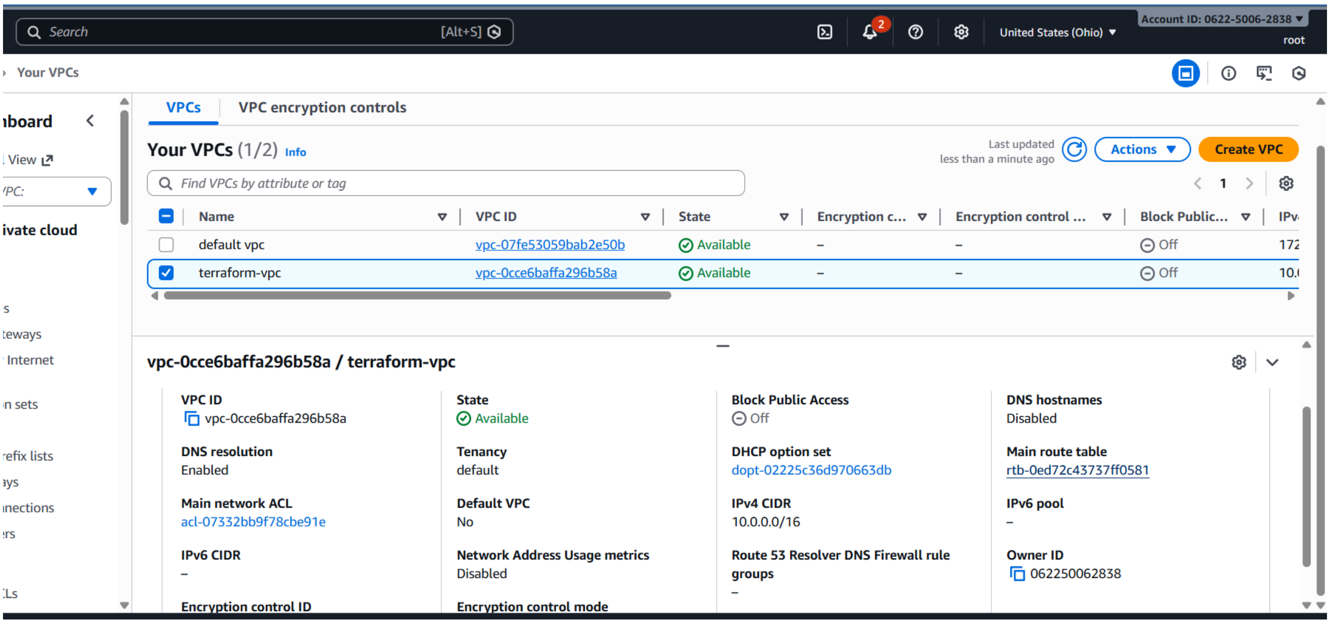
✓ Subnet created

✓ Internet Gateway attached

✓ Route table configured

✓ Security Group created

✓ EC2 instance launched inside the VPC

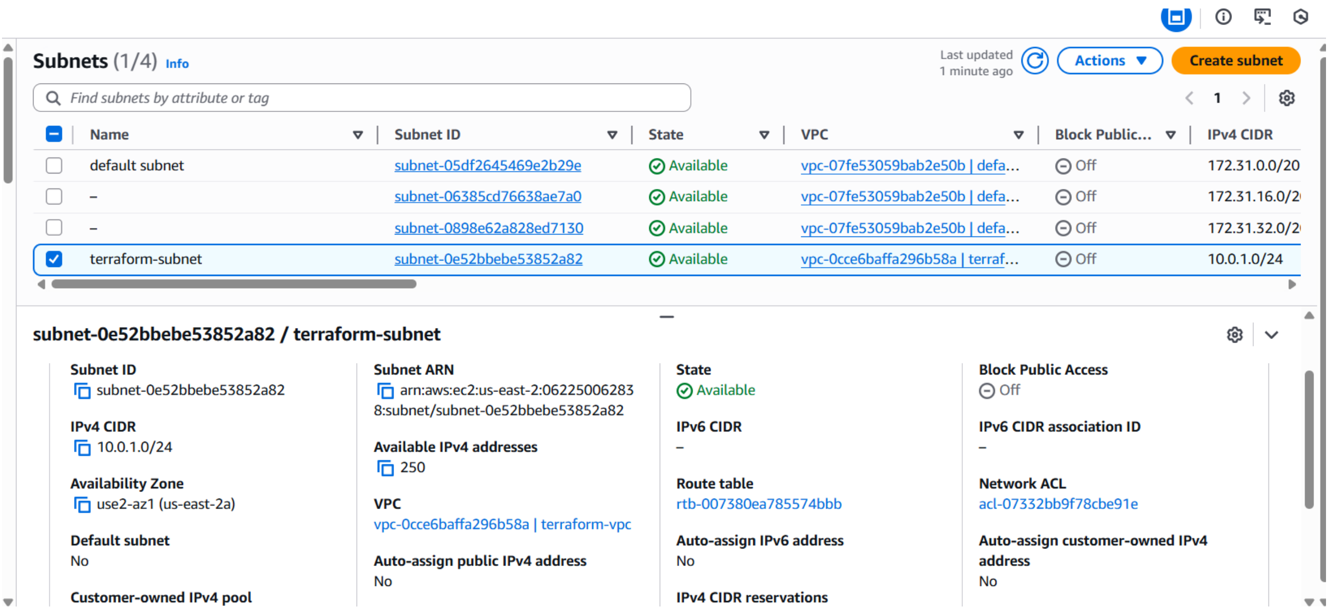


Your VPCs (1/2) Info

	Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv
☐	default vpc	vpc-07fe53059bab2e50b	Available	–	–	Off	172
☑	terraform-vpc	vpc-0cce6baffa296b58a	Available	–	–	Off	10.(

vpc-0cce6baffa296b58a / terraform-vpc

VPC ID vpc-0cce6baffa296b58a	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-02225c36d970663db	Main route table rtb-0ed72c43737ff0581
Main network ACL acl-07332bb9f78cbe91e	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –
IPv6 CIDR –	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 062250062838
Encryption control ID	Encryption control mode		

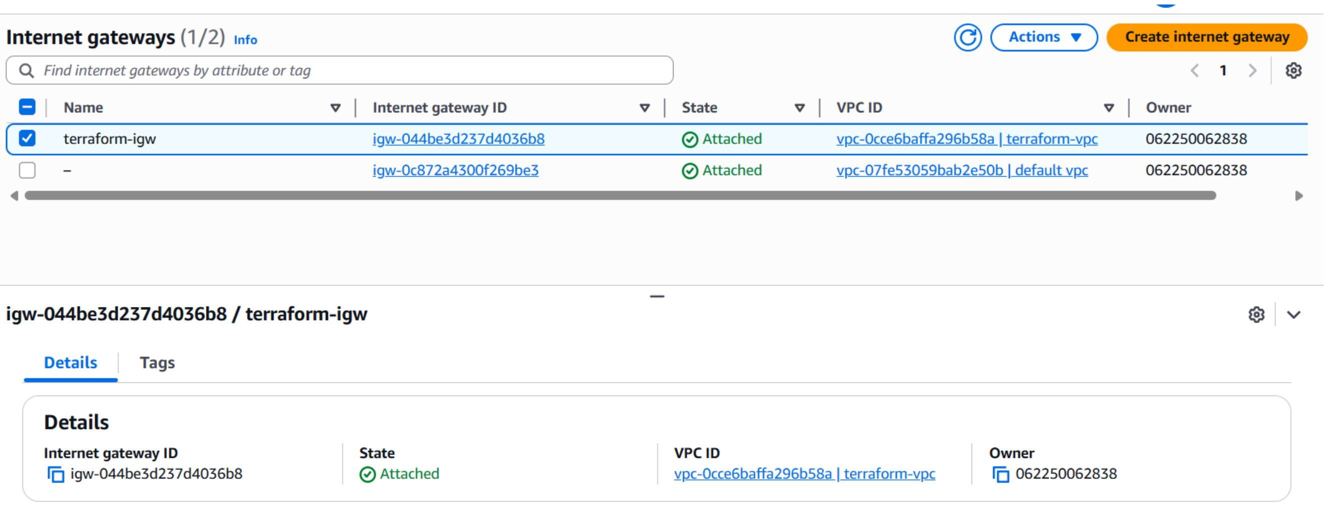


Subnets (1/4) Info

	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
☐	default subnet	subnet-05df2645469e2b29e	Available	vpc-07fe53059bab2e50b | defa...	Off	172.31.0.0/20
☐	–	subnet-06385cd76638ae7a0	Available	vpc-07fe53059bab2e50b | defa...	Off	172.31.16.0/2
☐	–	subnet-0898e62a828ed7130	Available	vpc-07fe53059bab2e50b | defa...	Off	172.31.32.0/2
☑	terraform-subnet	subnet-0e52bbebe53852a82	Available	vpc-0cce6baffa296b58a | terraf...	Off	10.0.1.0/24

subnet-0e52bbebe53852a82 / terraform-subnet

Subnet ID subnet-0e52bbebe53852a82	Subnet ARN arn:aws:ec2:us-east-2:062250062838:subnet/subnet-0e52bbebe53852a82	State Available	Block Public Access Off
IPv4 CIDR 10.0.1.0/24	Available IPv4 addresses 250	IPv6 CIDR –	IPv6 CIDR association ID –
Availability Zone use2-az1 (us-east-2a)	VPC vpc-0cce6baffa296b58a | terraform-vpc	Route table rtb-007380ea785574bbb	Network ACL acl-07332bb9f78cbe91e
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool		IPv4 CIDR reservations	



Internet gateways (1/2) Info
Actions ▼
Create internet gateway
Find internet gateways by attribute or tag

	Name	Internet gateway ID	State	VPC ID	Owner
☑	terraform-igw	igw-044be3d237d4036b8	✓ Attached	vpc-0cce6baffa296b58a | terraform-vpc	062250062838
□	–	igw-0c872a4300f269be3	✓ Attached	vpc-07fe53059bab2e50b | default vpc	062250062838

igw-044be3d237d4036b8 / terraform-igw

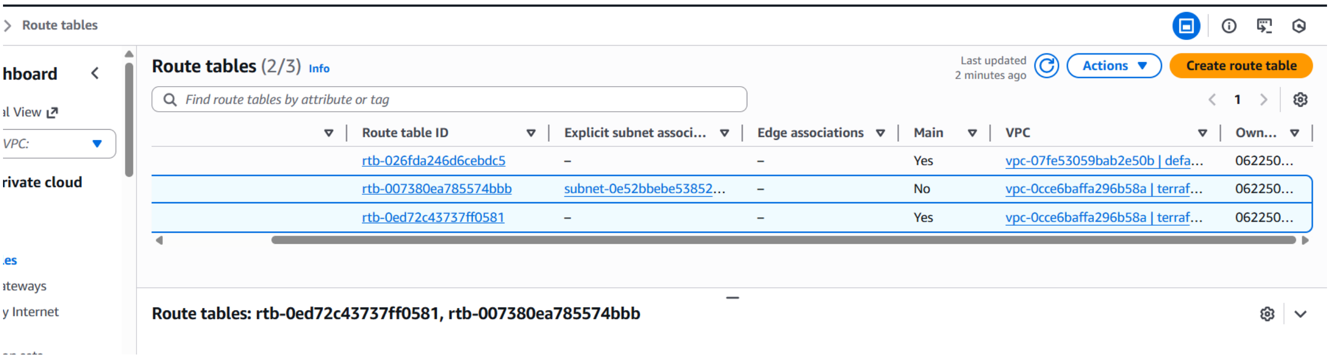
Details

Internet gateway ID
igw-044be3d237d4036b8

State
Attached

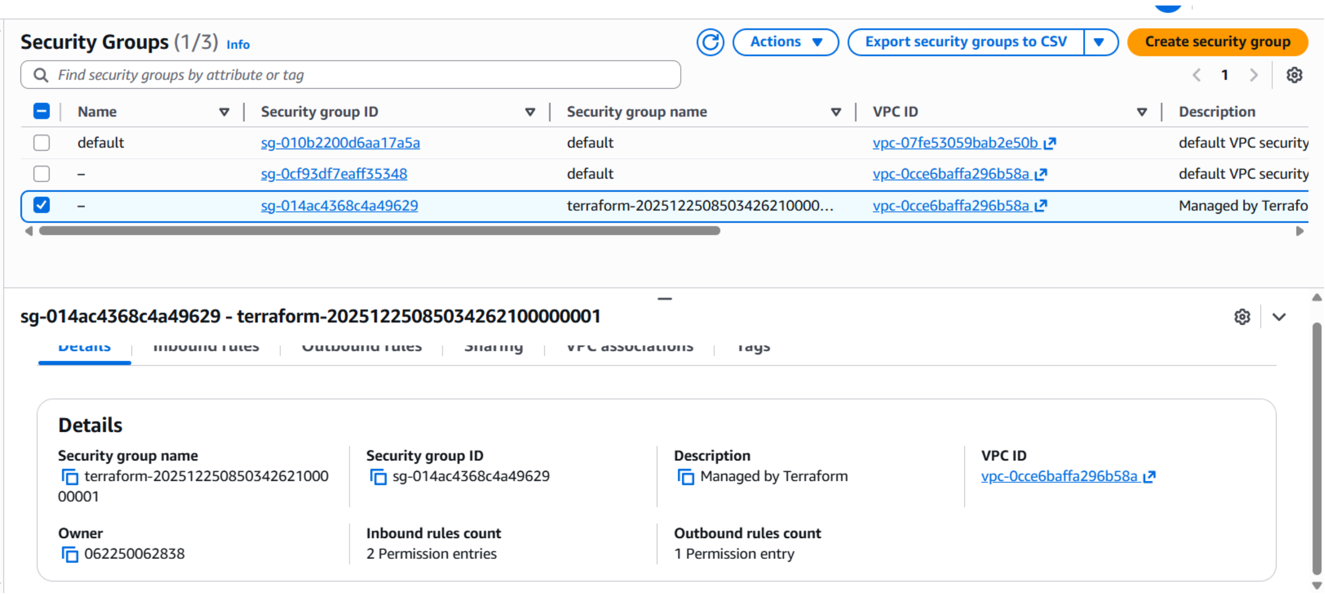
VPC ID
vpc-0cce6baffa296b58a | terraform-vpc

Owner
062250062838



	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Own...
	rtb-026fda246d6cebdc5	–	–	Yes	vpc-07fe53059bab2e50b | defa...	062250...
	rtb-007380ea785574bbb	subnet-0e52bbebe53852...	–	No	vpc-0cce6baffa296b58a | terraf...	062250...
	rtb-0ed72c43737ff0581	–	–	Yes	vpc-0cce6baffa296b58a | terraf...	062250...

Route tables: rtb-0ed72c43737ff0581, rtb-007380ea785574bbb



Security Groups (1/3) Info
Actions ▼
Export security groups to CSV
Create security group
Find security groups by attribute or tag

	Name	Security group ID	Security group name	VPC ID	Description
□	default	sg-010b2200d6aa17a5a	default	vpc-07fe53059bab2e50b	default VPC security
□	–	sg-0cf93df7eaff35348	default	vpc-0cce6baffa296b58a	default VPC security
☑	–	sg-014ac4368c4a49629	terraform-2025122508503426210000...	vpc-0cce6baffa296b58a	Managed by Terrafo

sg-014ac4368c4a49629 - terraform-20251225085034262100000001

Details

Security group name
terraform-20251225085034262100000001

Security group ID
sg-014ac4368c4a49629

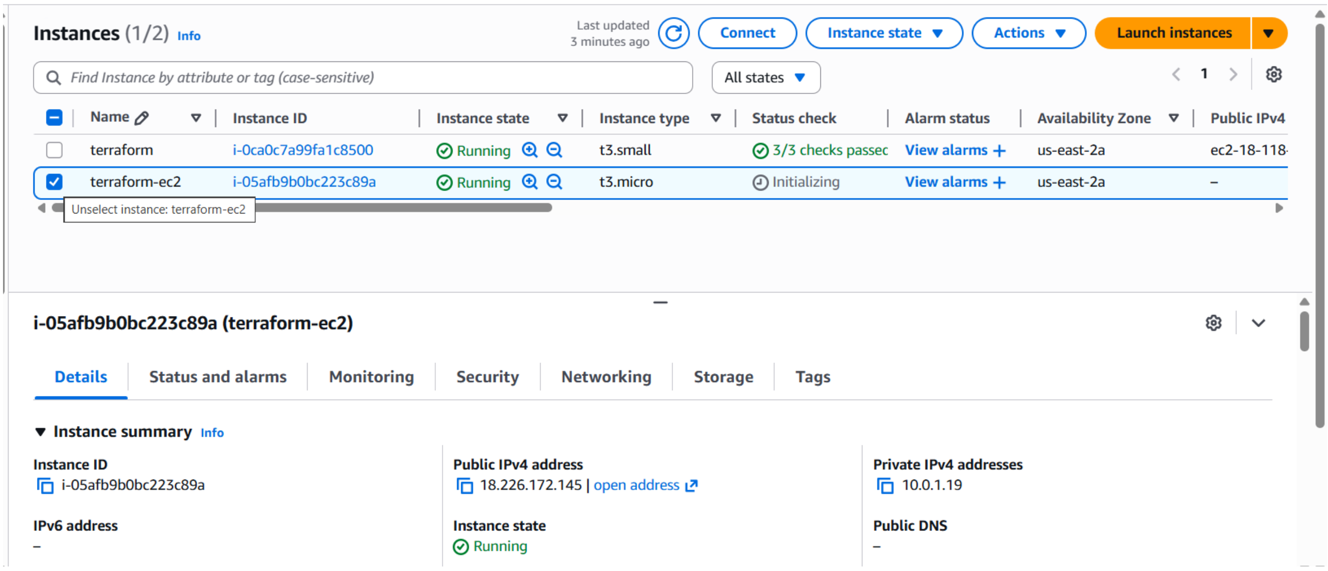
Description
Managed by Terraform

VPC ID
vpc-0cce6baffa296b58a

Owner
062250062838

Inbound rules count
2 Permission entries

Outbound rules count
1 Permission entry



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	terraform	i-0ca0c7a99fa1c8500	Running	t3.small	3/3 checks passec	View alarms +	us-east-2a	ec2-18-118-
☑	terraform-ec2	i-05afb9b0bc223c89a	Running	t3.micro	Initializing	View alarms +	us-east-2a	–

i-05afb9b0bc223c89a (terraform-ec2)

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-05afb9b0bc223c89a	18.226.172.145 | open address	10.0.1.19
IPv6 address	**Instance state**	**Public DNS**
–	Running	–

Conclusion

Successfully destroyed previous deployment, created a complete VPC architecture using Terraform, and deployed an EC2 instance inside the VPC.