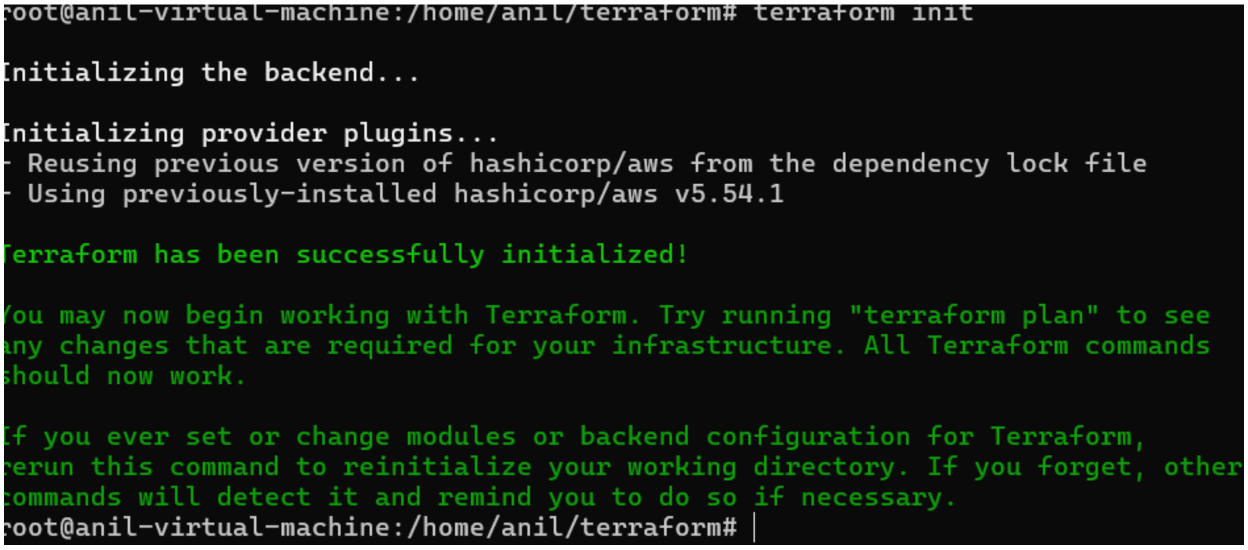
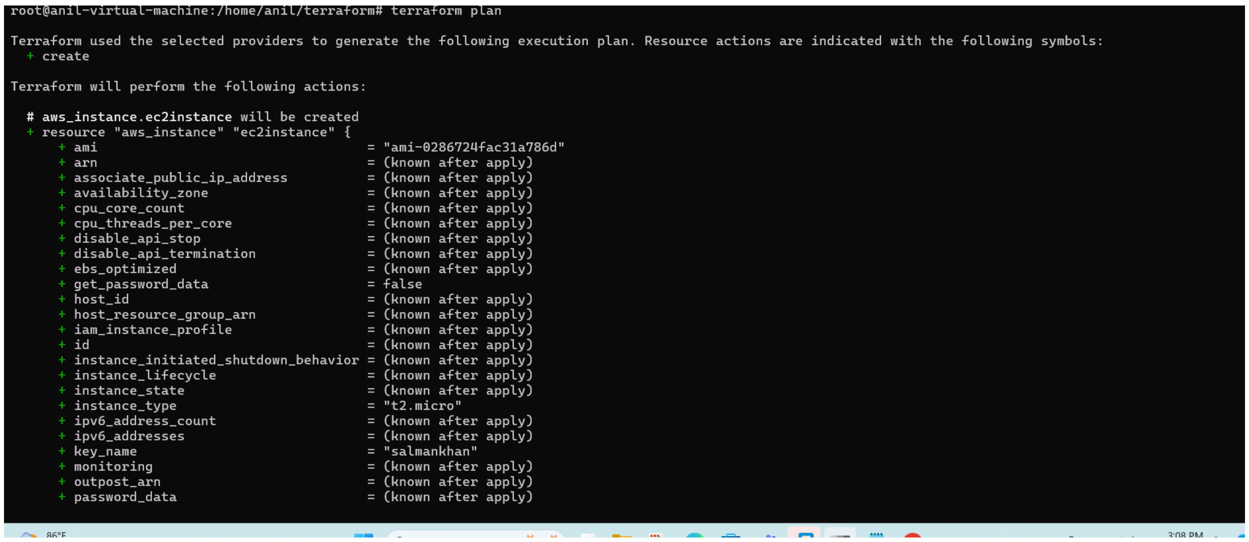
**Lunching EC2 and hosting an node application from Terraform.**

We initialize the terraform and it is initilized successfully.



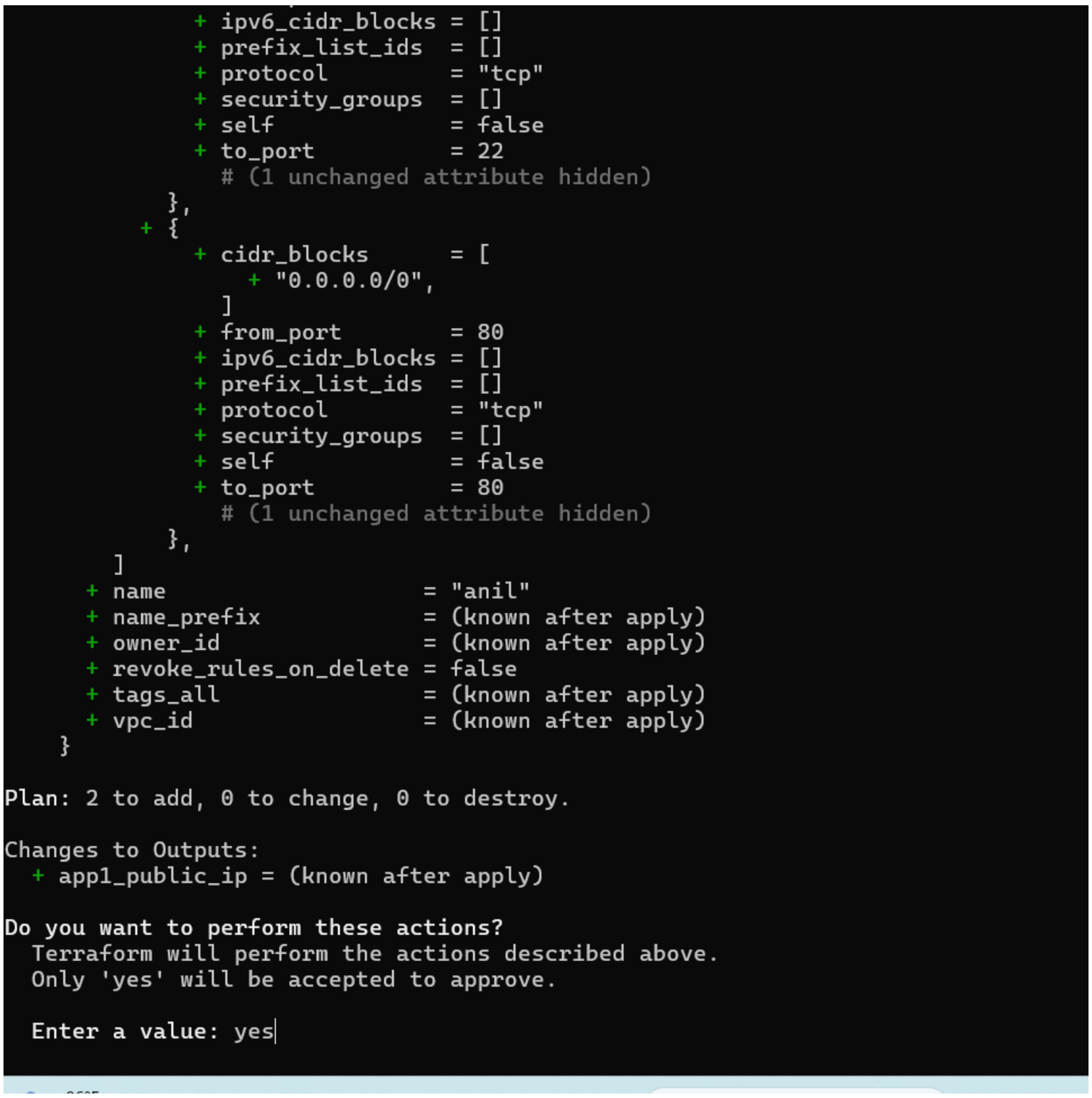
```
root@anil-virtual-machine:/home/anil/terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.54.1

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@anil-virtual-machine:/home/anil/terraform#
```

terraform plan and its result is in below snapshot:



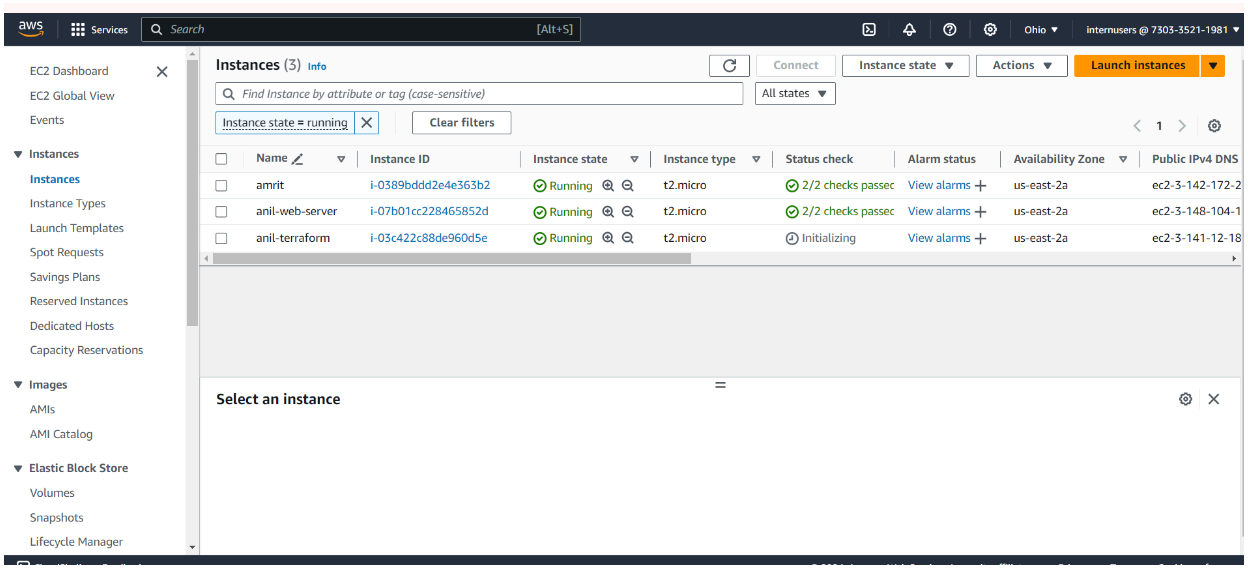
```
root@anil-virtual-machine:/home/anil/terraform# terraform plan

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_instance.ec2instance will be created
  + resource "aws_instance" "ec2instance" {
      + ami                                  = "ami-0286724fac31a786d"
      + 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                             = "salmankhan"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
```

terraform apply and its result is as below shown in snapshot:



```
              + 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        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "anil"
      + 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)
    }

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

Changes to Outputs:
  + app1_public_ip = (known after 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
```

Here is our ec2 which is created with name anil-terraform as shown in below snapshot:

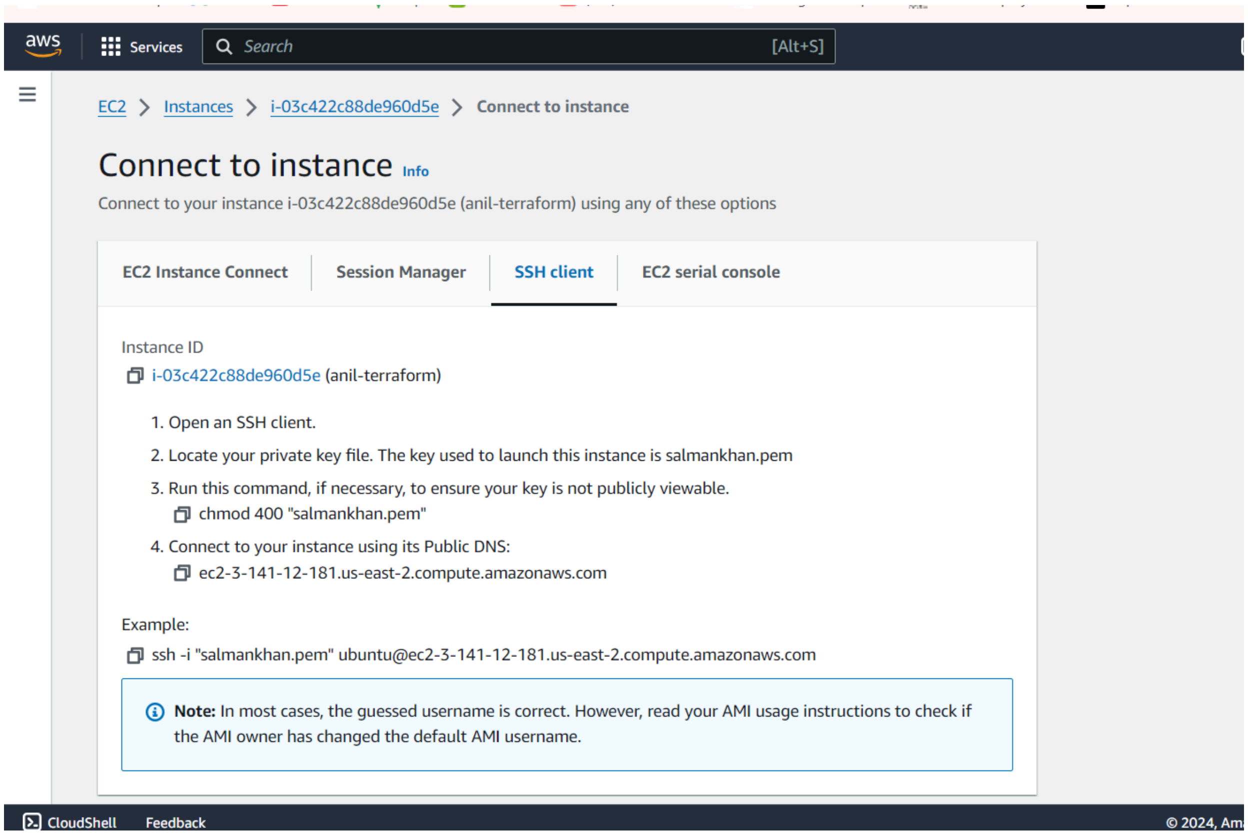


Instances (3) Info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|---|---|---|---|---|---|---|---|
| amrit | i-0389bddd2e4e363b2 | Running | t2.micro | 2/2 checks passec | View alarms + | us-east-2a | ec2-3-142-172-2 |
| anil-web-server | i-07b01cc228465852d | Running | t2.micro | 2/2 checks passec | View alarms + | us-east-2a | ec2-3-148-104-1 |
| anil-terraform | i-03c422c88de960d5e | Running | t2.micro | Initializing | View alarms + | us-east-2a | ec2-3-141-12-18 |

Select an instance

We can connect to the aws console with the ssh -i command we can copy and paste in the console:



Connect to instance Info

Connect to your instance i-03c422c88de960d5e (anil-terraform) using any of these options

Instance ID

i-03c422c88de960d5e (anil-terraform)

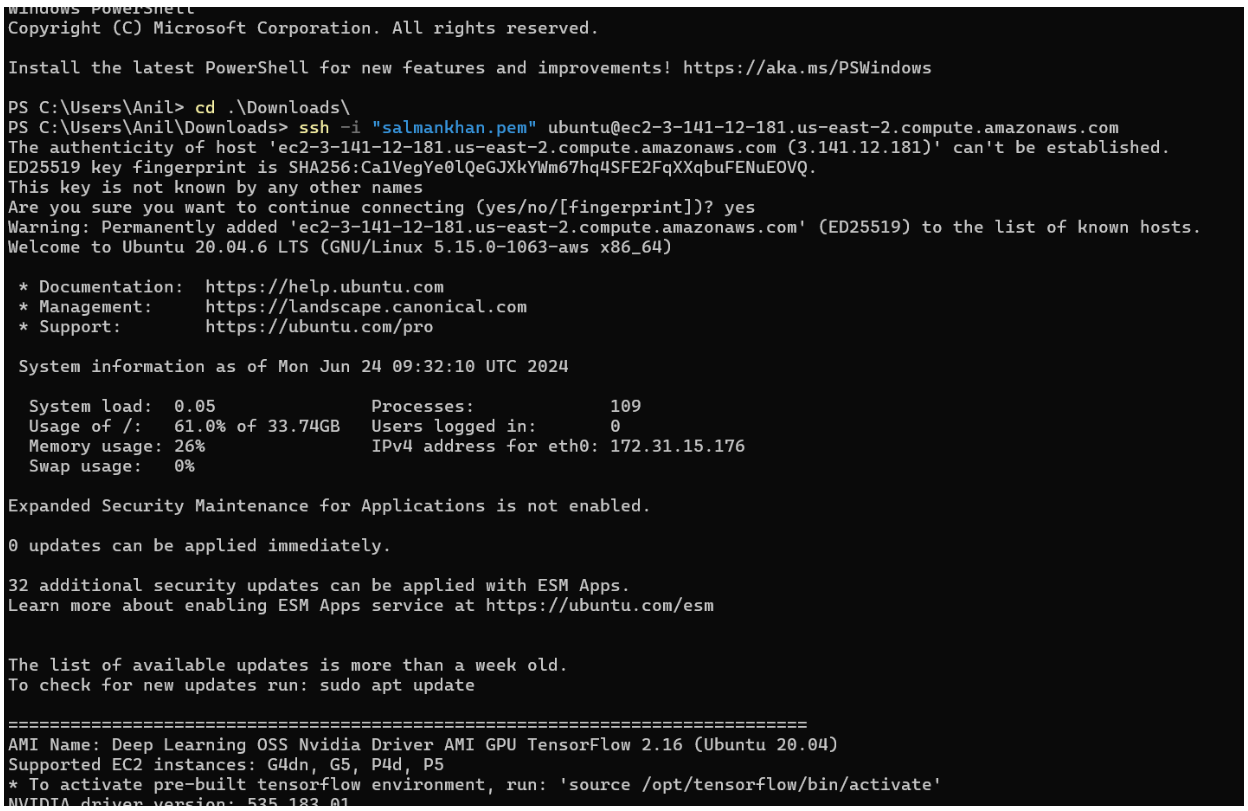
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is salmankhan.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "salmankhan.pem"
4. Connect to your instance using its Public DNS:
   ec2-3-141-12-181.us-east-2.compute.amazonaws.com

Example:

```
ssh -i "salmankhan.pem" ubuntu@ec2-3-141-12-181.us-east-2.compute.amazonaws.com
```

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Here we can see that its connected successfully:



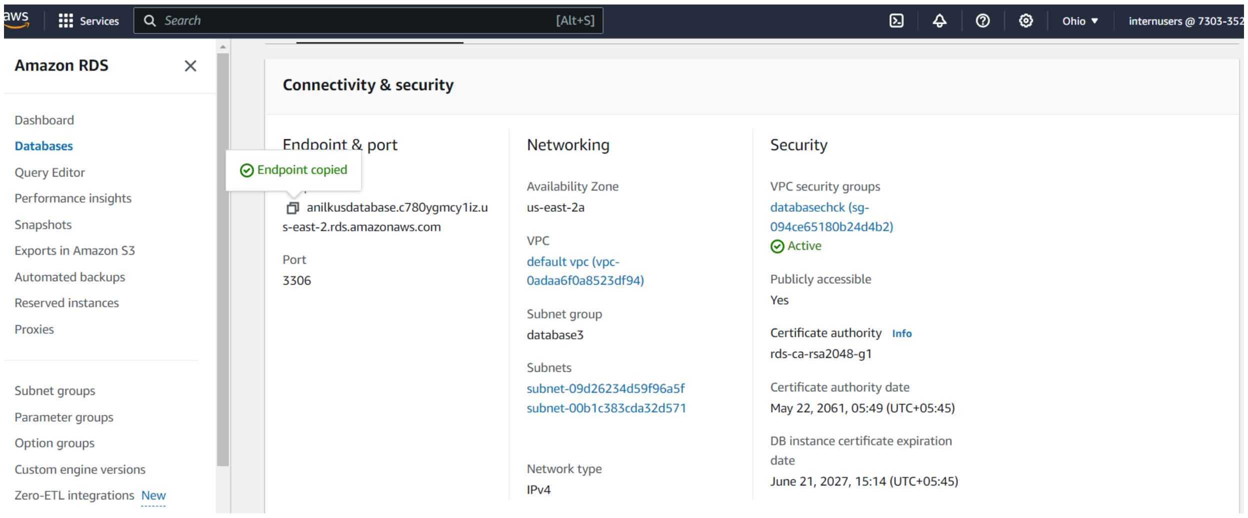
```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Anil> cd .\Downloads\
PS C:\Users\Anil\Downloads> ssh -i "salmankhan.pem" ubuntu@ec2-3-141-12-181.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-141-12-181.us-east-2.compute.amazonaws.com (3.141.12.181)' can't be established.
ED25519 key fingerprint is SHA256:Ca1VegYe0lQeGJXkYWm67hq4SFE2FqXXqbuFENuEOVQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-141-12-181.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1063-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Jun 24 09:32:10 UTC 2024

  System load:  0.05               Processes:             109
  Usage of /:   61.0% of 33.74GB   Users logged in:       0
  Memory usage: 26%                IPv4 address for eth0: 172.31.15.176
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

32 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

=============================================================================
AMI Name: Deep Learning OSS Nvidia Driver AMI GPU TensorFlow 2.16 (Ubuntu 20.04)
Supported EC2 instances: G4dn, G5, P4d, P5
* To activate pre-built tensorflow environment, run: 'source /opt/tensorflow/bin/activate'
```

## RDS

After we create the .tf we need to check it and apply it for that the commands are as follow

1.`terraform init` [it initialize the .tf code]
2.`terraform plan` [This command shows you what changes Terraform will make]
3.`terraform apply` [This command applies the changes]

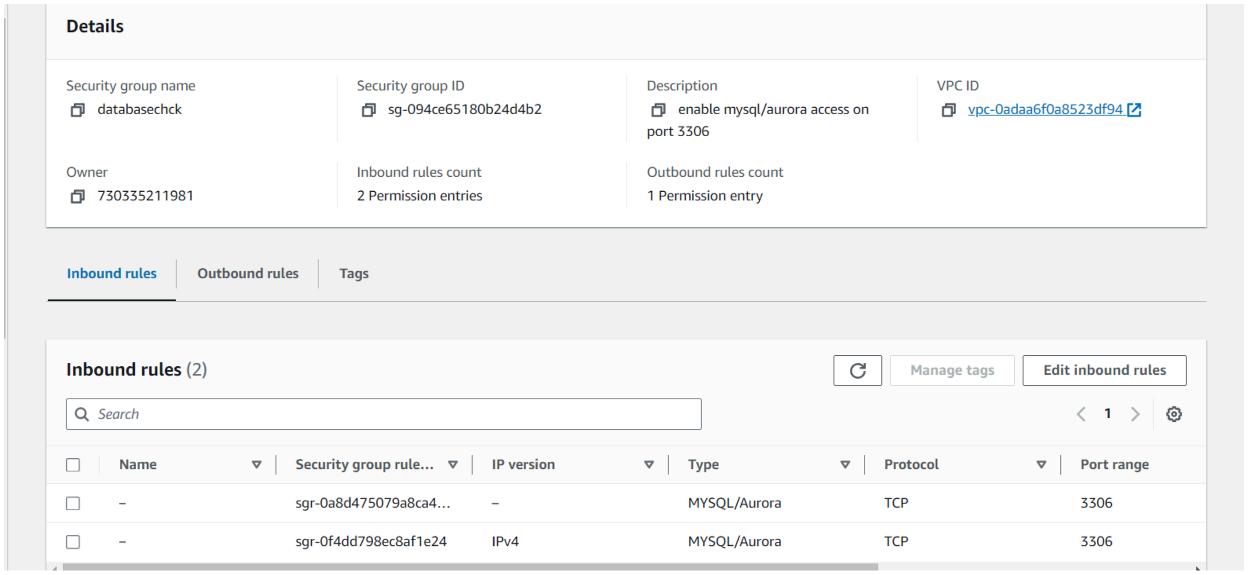
We have successfully created the RDS and deployed on the AWS:
We have created a database as shown on the below snapshot:
Here to connect to the ec2 we have copy Endpoint &port:



Connectivity & security

| Endpoint & port | Networking | Security |
|---|---|---|
| anilkusdatabase.c780ygmcy1iz.us-east-2.rds.amazonaws.com | Availability Zone: us-east-2a | VPC security groups: databasechck (sg-094ce65180b24d4b2) Active |
| Port: 3306 | VPC: default vpc (vpc-0adaa6f0a8523df94) | Publicly accessible: Yes |
| | Subnet group: database3 | Certificate authority Info: rds-ca-rsa2048-g1 |
| | Subnets: subnet-09d26234d59f96a5f subnet-00b1c383cda32d571 | Certificate authority date: May 22, 2061, 05:49 (UTC+05:45) |
| | Network type: IPv4 | DB instance certificate expiration date: June 21, 2027, 15:14 (UTC+05:45) |

In the below screen we have created the security group and it has been created successfully. We have added port 3306 here to get access from any network.



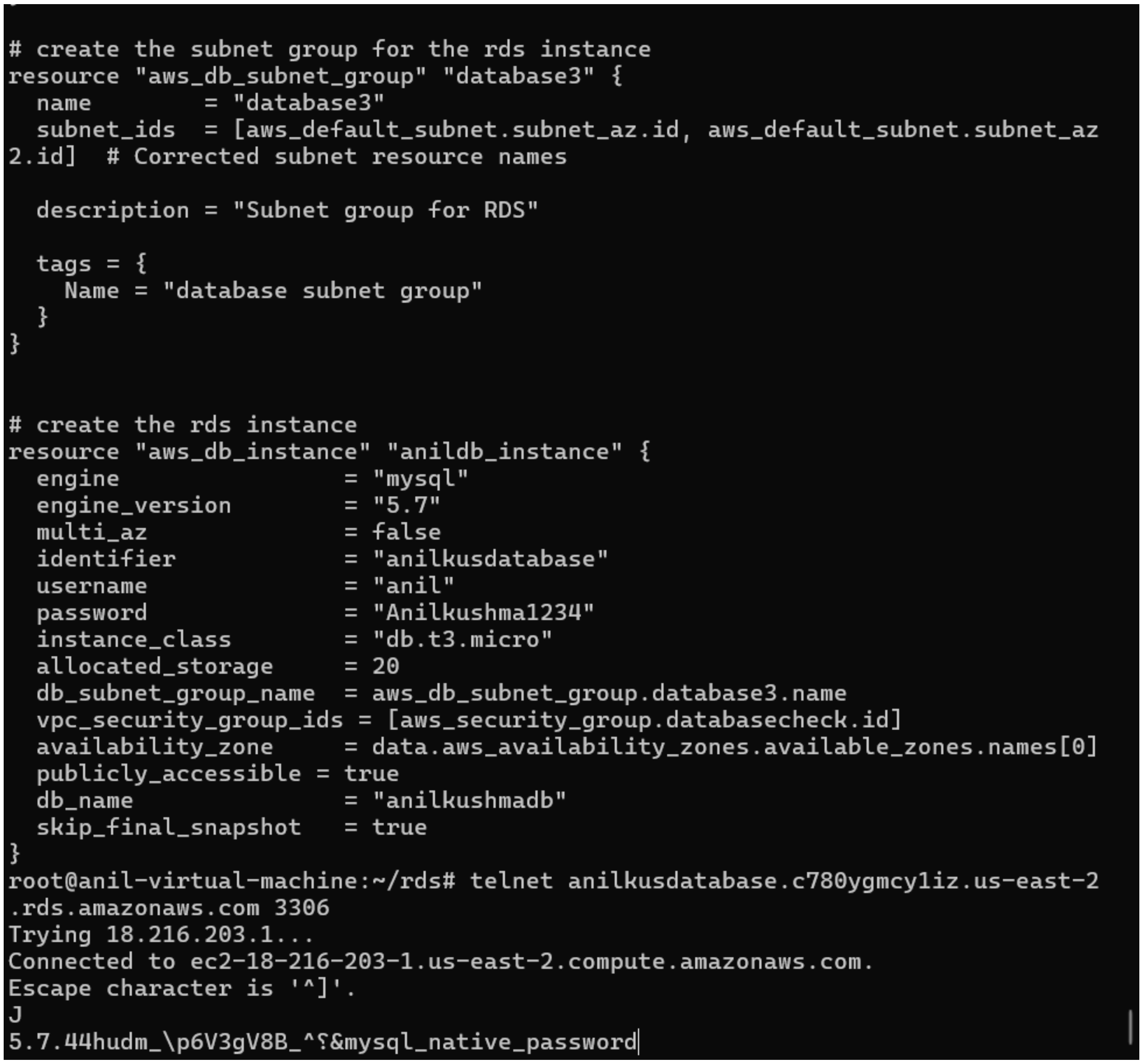
Details

| Security group name | Security group ID | Description | VPC ID |
|---|---|---|---|
| databasechck | sg-094ce65180b24d4b2 | enable mysql/aurora access on port 3306 | vpc-0adaa6f0a8523df94 |
| **Owner** | **Inbound rules count** | **Outbound rules count** | |
| 730335211981 | 2 Permission entries | 1 Permission entry | |

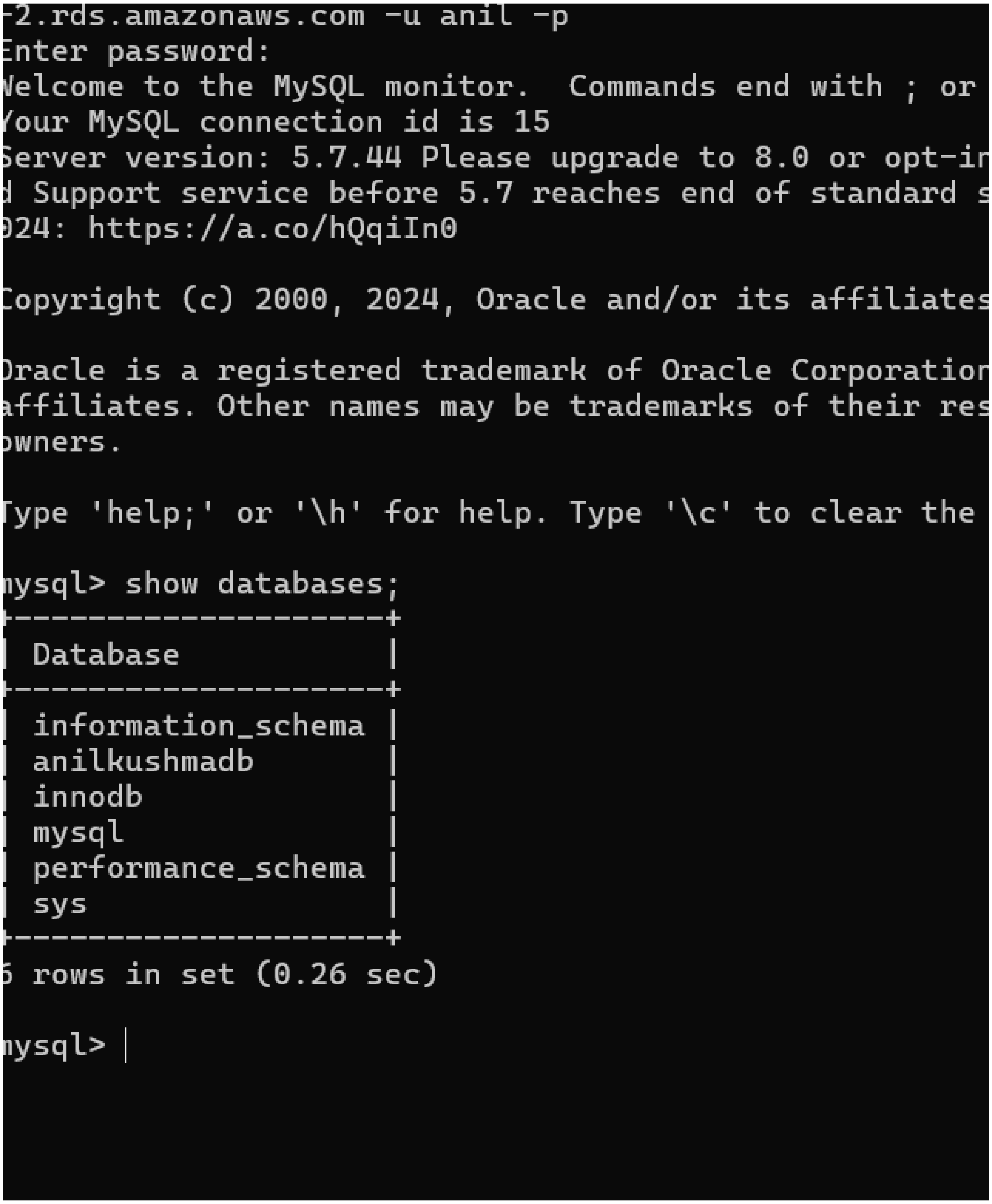
Inbound rules (2)

| Name | Security group rule... | IP version | Type | Protocol | Port range |
|---|---|---|---|---|---|
| – | sgr-0a8d475079a8ca4... | – | MYSQL/Aurora | TCP | 3306 |
| – | sgr-0f4dd798ec8af1e24 | IPv4 | MYSQL/Aurora | TCP | 3306 |

Here is the final report of our .tf and we can see our .tf is working successfully and we are able to connect to the databases.



```
# create the subnet group for the rds instance
resource "aws_db_subnet_group" "database3" {
  name        = "database3"
  subnet_ids  = [aws_default_subnet.subnet_az.id, aws_default_subnet.subnet_az
2.id]  # Corrected subnet resource names

  description = "Subnet group for RDS"

  tags = {
    Name = "database subnet group"
  }
}


# create the rds instance
resource "aws_db_instance" "anildb_instance" {
  engine                 = "mysql"
  engine_version         = "5.7"
  multi_az               = false
  identifier             = "anilkusdatabase"
  username               = "anil"
  password               = "Anilkushma1234"
  instance_class         = "db.t3.micro"
  allocated_storage      = 20
  db_subnet_group_name   = aws_db_subnet_group.database3.name
  vpc_security_group_ids = [aws_security_group.databasecheck.id]
  availability_zone      = data.aws_availability_zones.available_zones.names[0]
  publicly_accessible = true
  db_name                = "anilkushmadb"
  skip_final_snapshot    = true
}
root@anil-virtual-machine:~/rds# telnet anilkusdatabase.c780ygmcy1iz.us-east-2
.rds.amazonaws.com 3306
Trying 18.216.203.1...
Connected to ec2-18-216-203-1.us-east-2.compute.amazonaws.com.
Escape character is '^]'.
J
5.7.44hudm_\p6V3gV8B_^?&mysql_native_password
```



```
-2.rds.amazonaws.com -u anil -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or
Your MySQL connection id is 15
Server version: 5.7.44 Please upgrade to 8.0 or opt-ir
d Support service before 5.7 reaches end of standard s
024: https://a.co/hQqiIn0

Copyright (c) 2000, 2024, Oracle and/or its affiliates

Oracle is a registered trademark of Oracle Corporatior
affiliates. Other names may be trademarks of their res
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| anilkushmadb       |
| innodb             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.26 sec)

mysql>
```

# Vpc

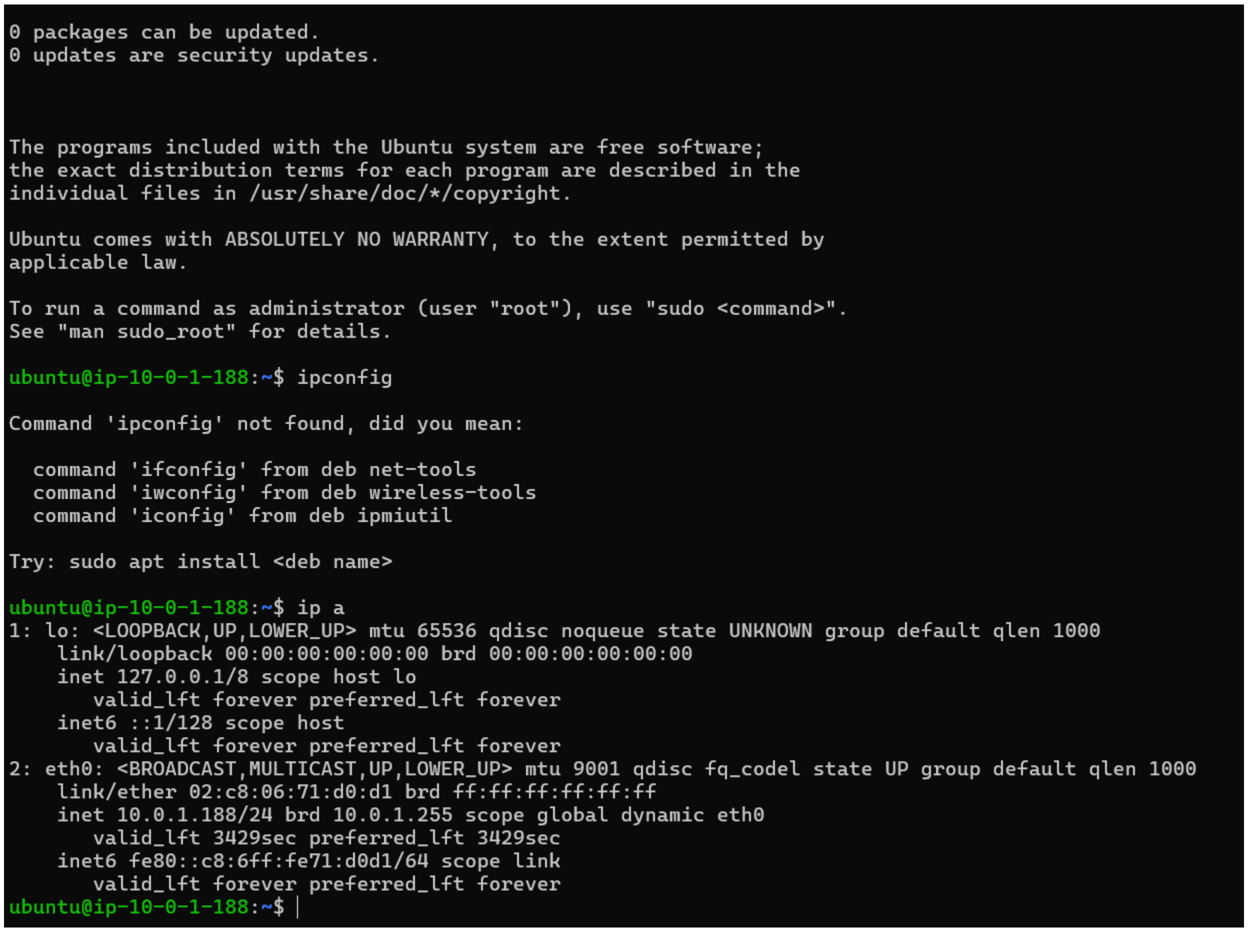
After defining your Terraform configuration file (`main.tf`), you can apply it using the following commands:

```
terraform init
```

```
terraform plan
```

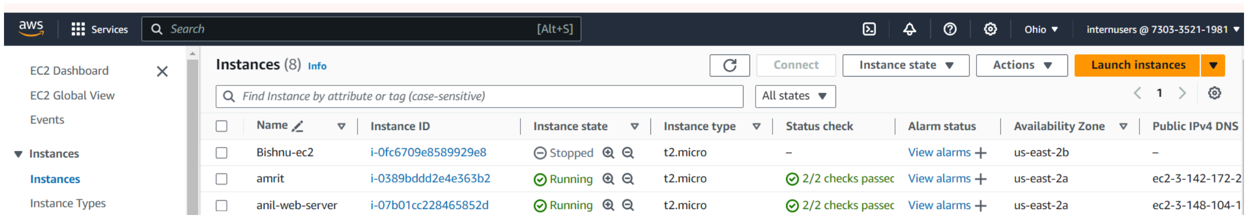
```
terraform apply
```

Once Terraform applies the configuration, it will create the VPC, subnets, and internet gateway as specified. You can verify the resources created by checking the AWS Management Console or by using AWS CLI commands .
Its connected successfully as shown on the below snapshot.



```
0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-188:~$ ipconfig

Command 'ipconfig' not found, did you mean:

  command 'ifconfig' from deb net-tools
  command 'iwconfig' from deb wireless-tools
  command 'iconfig' from deb ipmiutil

Try: sudo apt install <deb name>

ubuntu@ip-10-0-1-188:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:c8:06:71:d0:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.188/24 brd 10.0.1.255 scope global dynamic eth0
       valid_lft 3429sec preferred_lft 3429sec
    inet6 fe80::c8:6ff:fe71:d0d1/64 scope link
       valid_lft forever preferred_lft forever
ubuntu@ip-10-0-1-188:~$
```

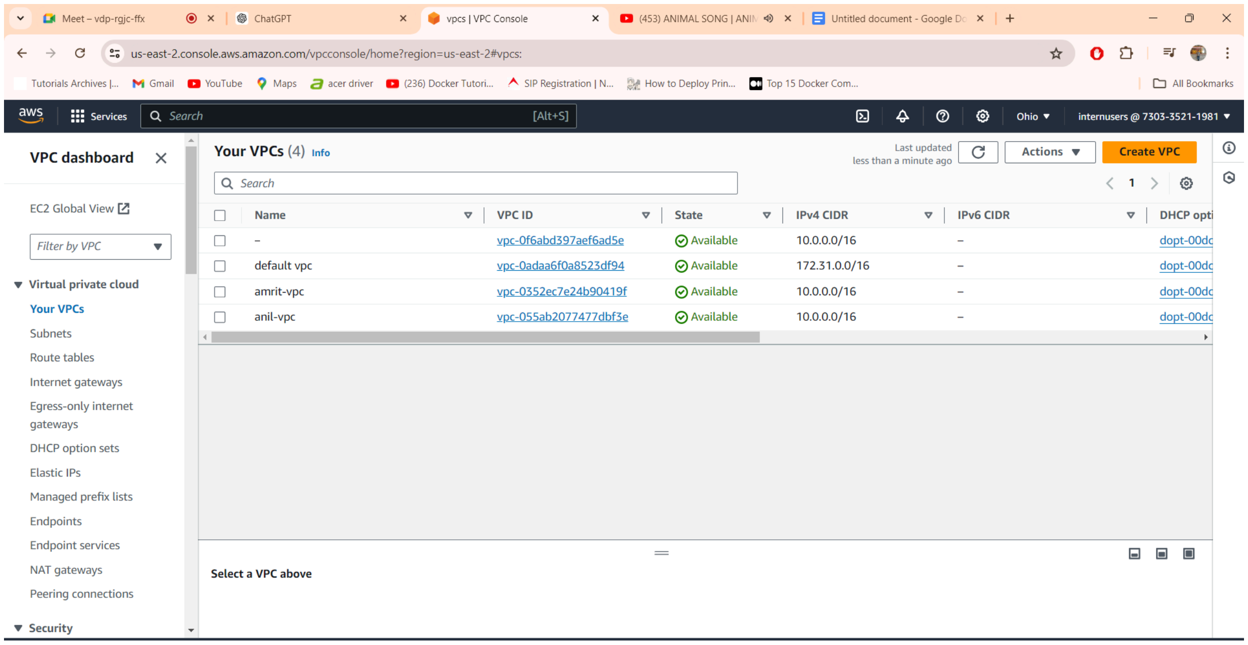
Here is the EC2 that we have made to check is our vpc is woking or not



Instances (8)

| | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|---|---|---|---|---|---|---|---|---|
| ☐ | Bishnu-ec2 | i-0fc6709e8589929e8 | Stopped | t2.micro | – | View alarms + | us-east-2b | – |
| ☐ | amrit | i-0389bddd2e4e363b2 | Running | t2.micro | 2/2 checks passec | View alarms + | us-east-2a | ec2-3-142-172-2 |
| ☐ | anil-web-server | i-07b01cc228465852d | Running | t2.micro | 2/2 checks passec | View alarms + | us-east-2a | ec2-3-148-104-1 |

Here we have created the vpc as shown on the below snapshot:



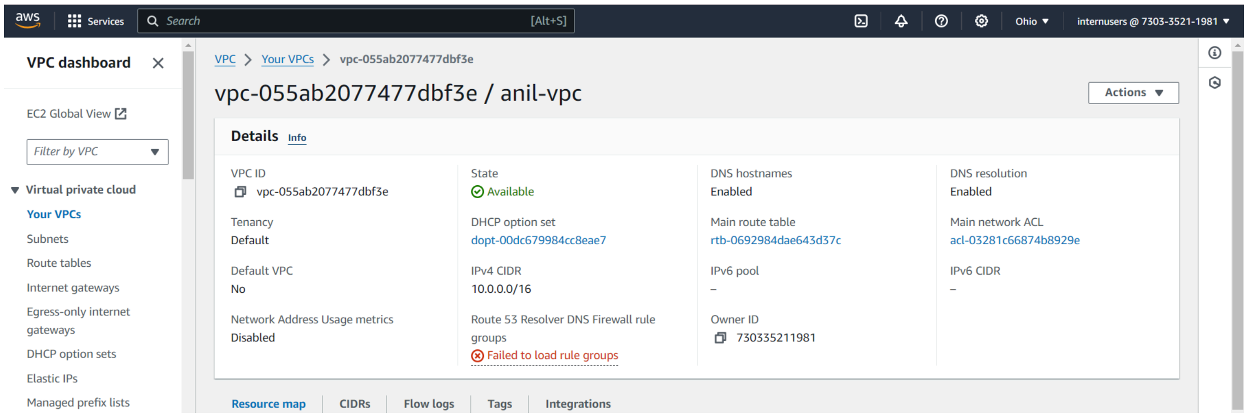
us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#vpcs:

Your VPCs (4)

| | Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR | DHCP opti |
|---|---|---|---|---|---|---|
| ☐ | – | vpc-0f6abd397aef6ad5e | Available | 10.0.0.0/16 | – | dopt-00dc |
| ☐ | default vpc | vpc-0adaa6f0a8523df94 | Available | 172.31.0.0/16 | – | dopt-00dc |
| ☐ | amrit-vpc | vpc-0352ec7e24b90419f | Available | 10.0.0.0/16 | – | dopt-00dc |
| ☐ | anil-vpc | vpc-055ab2077477dbf3e | Available | 10.0.0.0/16 | – | dopt-00dc |

Select a VPC above

Here are the details about the vpc that we have created :



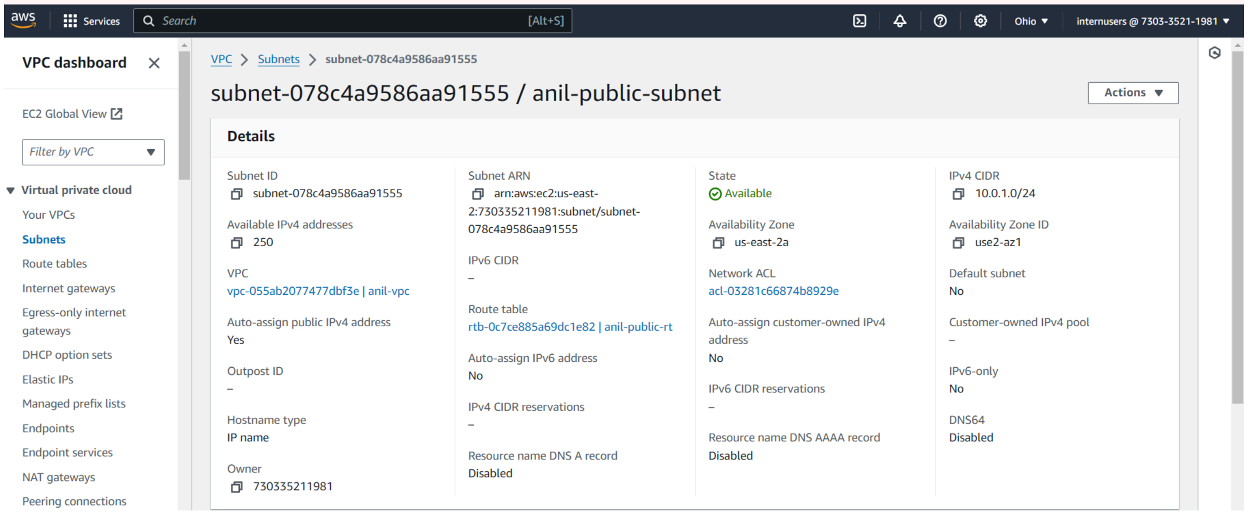
VPC > Your VPCs > vpc-055ab2077477dbf3e

vpc-055ab2077477dbf3e / anil-vpc

Details

| | | | |
|---|---|---|---|
| VPC ID<br>vpc-055ab2077477dbf3e | State<br>Available | DNS hostnames<br>Enabled | DNS resolution<br>Enabled |
| Tenancy<br>Default | DHCP option set<br>dopt-00dc679984cc8eae7 | Main route table<br>rtb-0692984dae643d37c | Main network ACL<br>acl-03281c66874b8929e |
| Default VPC<br>No | IPv4 CIDR<br>10.0.0.0/16 | IPv6 pool<br>– | IPv6 CIDR<br>– |
| Network Address Usage metrics<br>Disabled | Route 53 Resolver DNS Firewall rule groups<br>Failed to load rule groups | Owner ID<br>730335211981 | |

After we have details of vpc our .tf configuration have also created the subnet and it shown on the below snapshot:



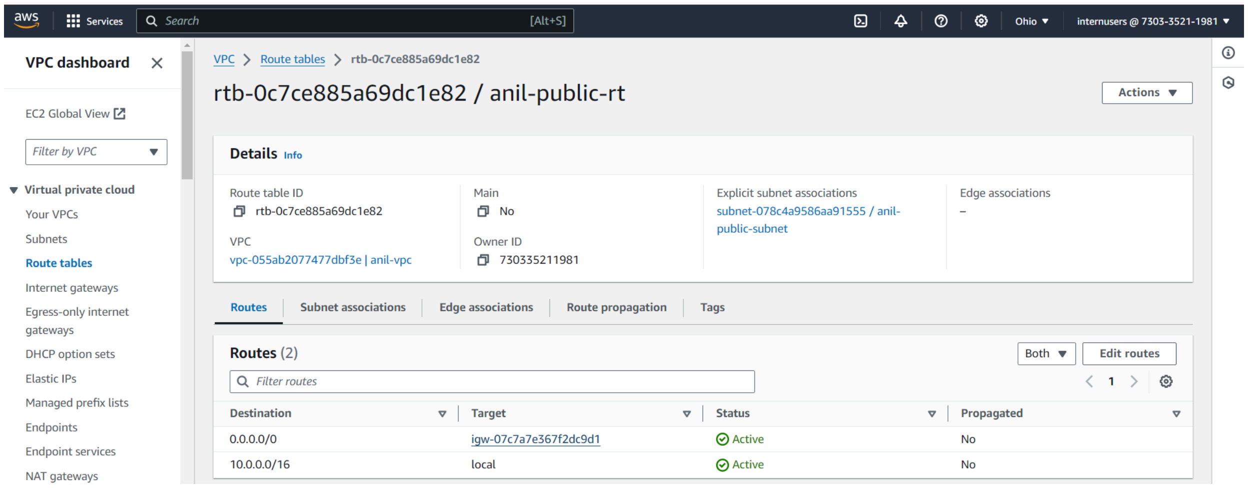
VPC > Subnets > subnet-078c4a9586aa91555

subnet-078c4a9586aa91555 / anil-public-subnet

Details

| | | | |
|---|---|---|---|
| Subnet ID: subnet-078c4a9586aa91555 | Subnet ARN: arn:aws:ec2:us-east-2:730335211981:subnet/subnet-078c4a9586aa91555 | State: Available | IPv4 CIDR: 10.0.1.0/24 |
| Available IPv4 addresses: 250 | IPv6 CIDR: – | Availability Zone: us-east-2a | Availability Zone ID: use2-az1 |
| VPC: vpc-055ab2077477dbf3e \| anil-vpc | Route table: rtb-0c7ce885a69dc1e82 \| anil-public-rt | Network ACL: acl-03281c66874b8929e | Default subnet: No |
| Auto-assign public IPv4 address: Yes | Auto-assign IPv6 address: No | Auto-assign customer-owned IPv4 address: No | Customer-owned IPv4 pool: – |
| Outpost ID: – | IPv4 CIDR reservations: – | IPv6 CIDR reservations: – | IPv6-only: No |
| Hostname type: IP name | Resource name DNS A record: Disabled | Resource name DNS AAAA record: Disabled | DNS64: Disabled |
| Owner: 730335211981 | | | |

Here are the details of the route tables too on the below snapshot:



VPC > Route tables > rtb-0c7ce885a69dc1e82

rtb-0c7ce885a69dc1e82 / anil-public-rt

Details Info

| | | | |
|---|---|---|---|
| Route table ID: rtb-0c7ce885a69dc1e82 | Main: No | Explicit subnet associations: subnet-078c4a9586aa91555 / anil-public-subnet | Edge associations: – |
| VPC: vpc-055ab2077477dbf3e \| anil-vpc | Owner ID: 730335211981 | | |

Routes (2)

| Destination | Target | Status | Propagated |
|---|---|---|---|
| 0.0.0.0/0 | igw-07c7a7e367f2dc9d1 | Active | No |
| 10.0.0.0/16 | local | Active | No |