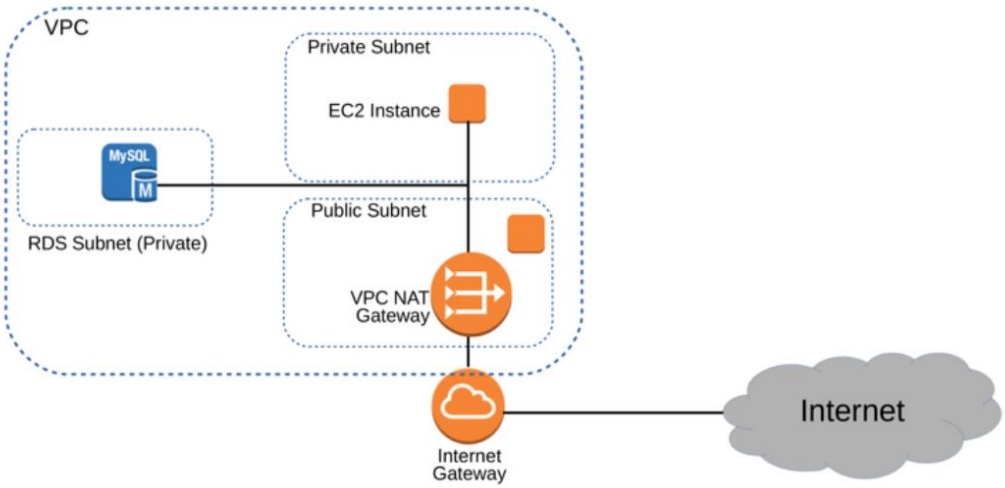
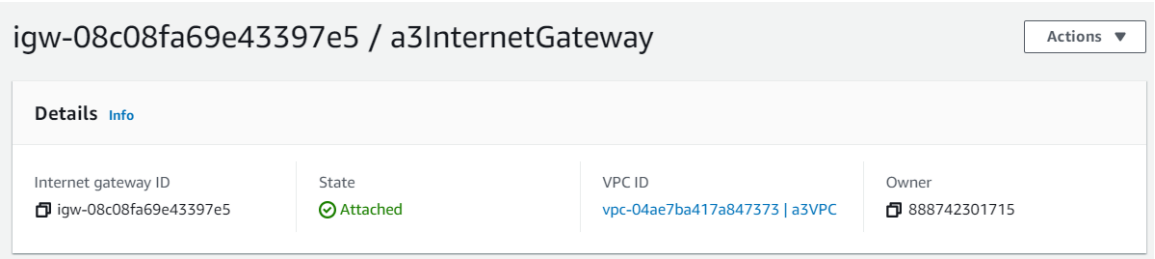
Hosting a website from an instance in a private subnet



1. Create a new VPC and name it a3VPC

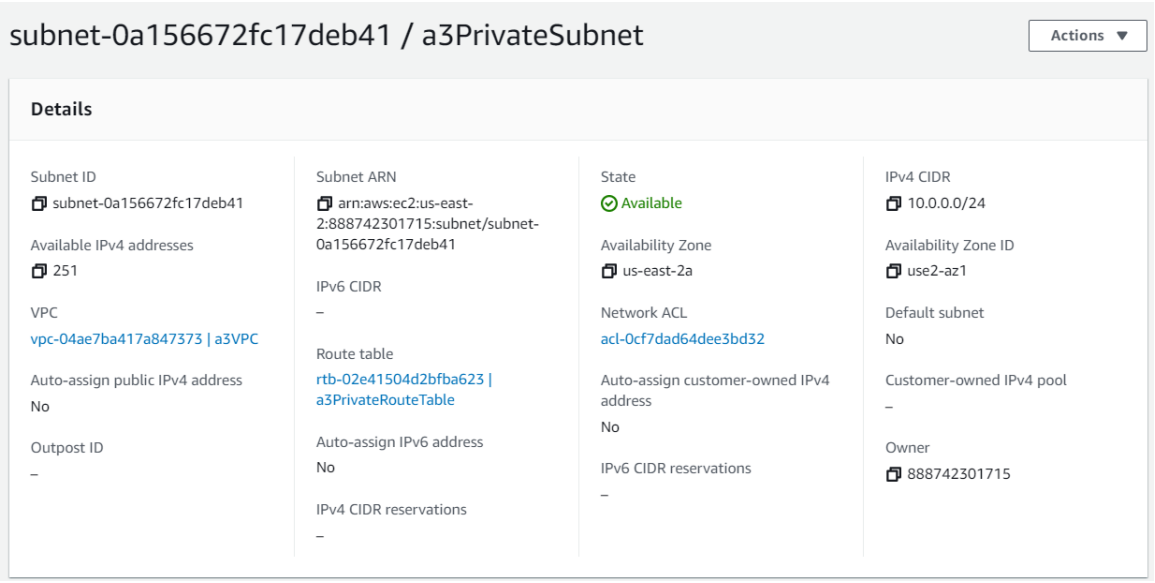
- Log in to AWS Management Console
- Go to VPC Dashboard
- Click 'Create VPC'
    - Name tag: 'a3VPC'
    - IPv4 CIDR block: 10.0.0.0/16
    - No IPv6 CIDR block
    - Tenancy: Default
    - DNS hostnames: Enabled

- A VPC requires an internet gateway
- In the VPC Dashboard, click 'Internet Gateways", "Create internet gateway"
    - Name tag: a3InternetGateway
- Click "Action", "Attach to VPC"
    - Select: a3VPC

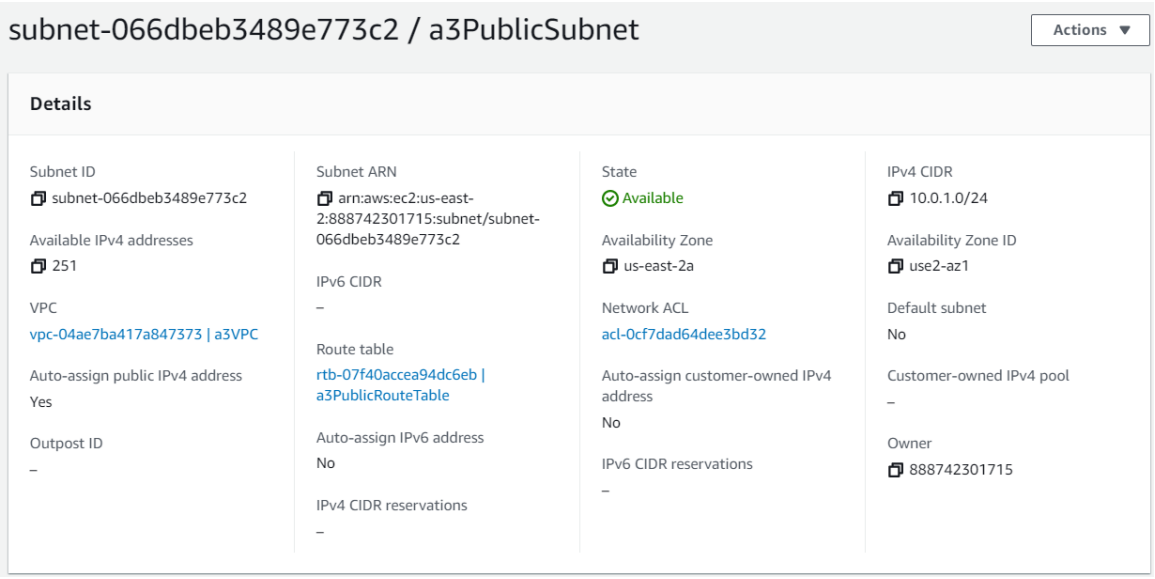


- The route table rtb-02e41504d2bfba623 has been automatically created and attached to a3VPC
    - I will use this route table for the private subnet
    - Click “Route Tables”, name this default route table “a3PrivateRouteTable”
- Create another route table for the public subnet
    - Name: a3PublicRouteTable
    - VPC: a3VPC
- Screenshots of these route tables will be taken later after subnet associations.

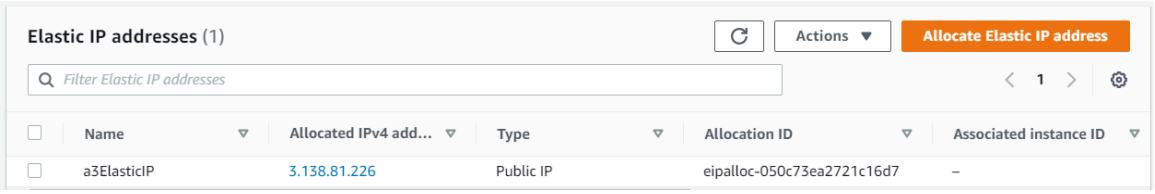
- Create a private subnet
    - Navigate to VPC Dashboard, click ‘Subnets’, click ‘Create subnet’
    - VPC ID: a3VPC
    - Subnet name: a3PrivateSubnet
    - Availability Zone: US East (Ohio) / us-east-2a
    - IPv4 CIDR block: 10.0.0.0/24
    - Auto-assign public IPv4 address: No
    - Route table association: a3PrivateRouteTable



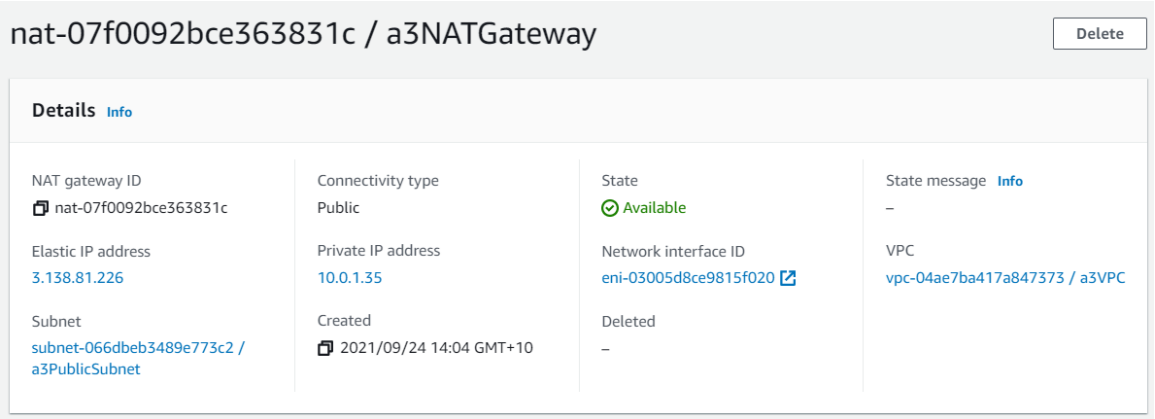
- Create a public subnet
    - VPC ID: a3VPC
    - Subnet name: a3PublicSubnet
    - Availability Zone: US East (Ohio) / us-east-2a
    - IPv4 CIDR block: 10.0.1.0/24
    - Auto-assign public IPv4 address: Yes
    - Route table association: a3PublicRouteTable



- Allocate an elastic IP address for the NAT Gateway
  - Navigate to VPC Dashboard, click “Elastic IPs”, “Allocate Elastic IP address”
  - Public IPv4 address pool: Amazon’s pool of IPv4 addresses
  - Add tag: Key: Name, Value: a3ElasticIP

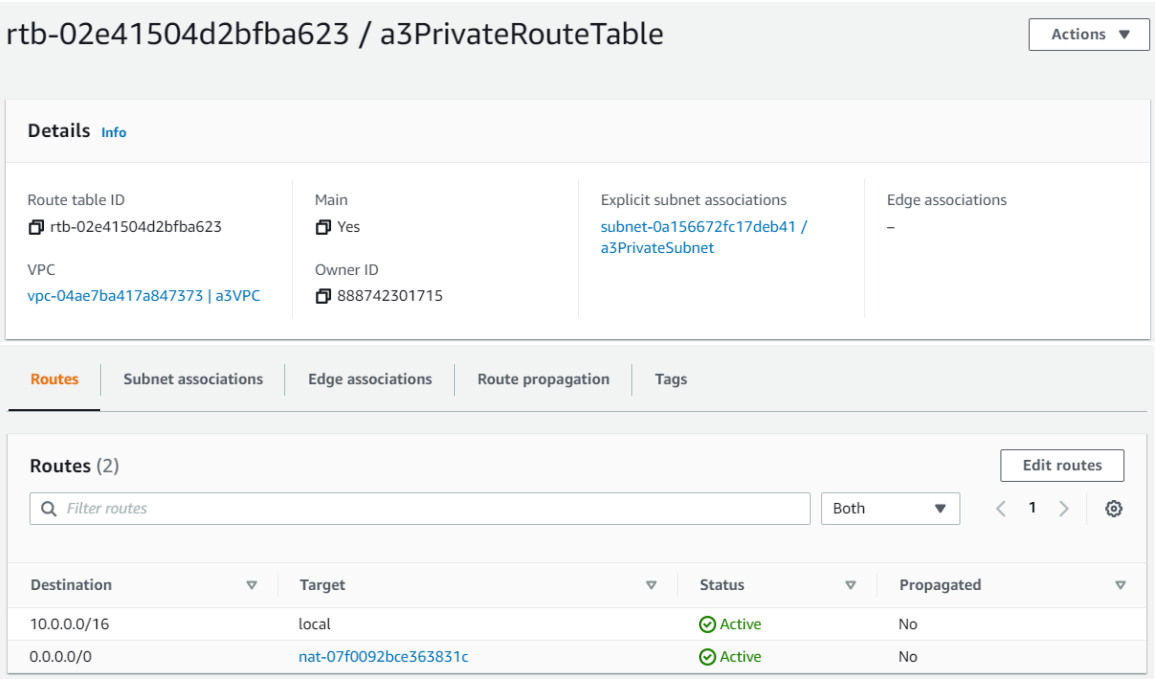


- Create a NAT Gateway
  - Navigate to VPC Dashboard, click “NAT Gateways”, “Create NAT gateway”
  - Name: a3NATGateway
  - Subnet: a2PublicSunet
  - Connectivity type: Public
  - allocate elastic IP address: a3ElasticIP

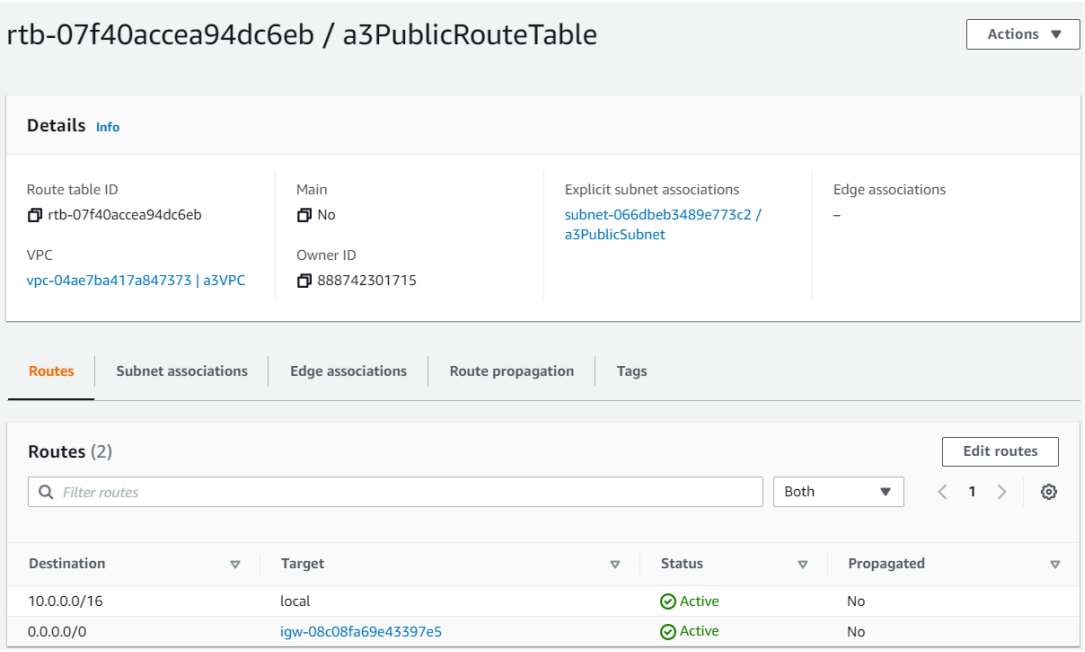


- Edit a3PrivateRouteTable to point internet-bound traffic to a3NATGateway
  - Select a3RouteTable, click “Actions”, “Edit routes”, “Add routes”:
    - Destination: 0.0.0.0/0, Target: a3InternetGateway

- Destination: 0.0.0.0/0, Target: a3NATGateway

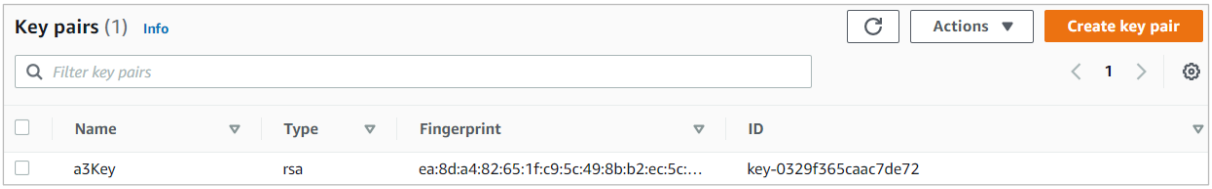


- Edit a3PublicRouteTable to point internet-bound traffic to a3InternetGateway
  - Select a3RouteTable, click “Actions”, “Edit routes”, “Add routes”:
    - Destination: 0.0.0.0/0, Target: a3InternetGateway
    - Destination: 0.0.0.0/0, Target: a3InternetGateway



2. Create an instance in a private subnet

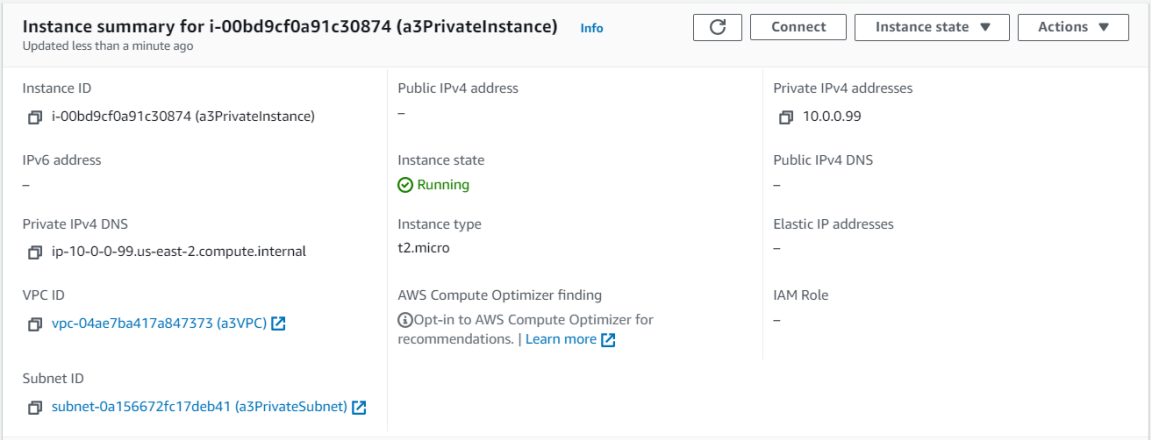
- Create a key pair in the EC2 Dashboard
    - Name: a3Key
    - Key pair type: RSA
    - File format: ppk



- Launch an instance in a3PrivateSubnet
- Navigate to EC2 Dashboard and click 'Launch instance'
- Step 1: Select 'Amazon Linux 2 AMI (HVM) SSD Volume Type 64-bit (x86)'
- Step 2: Choose the Instance Type 't2.micro'
- Step 3: Configure instance details:
    - Network: a3VPC
    - Subnet: a3PrivateSubnet
    - Auto-assign Public IP: Use subnet setting (Disable)
    - Advanced Details: User data: insert the following code:

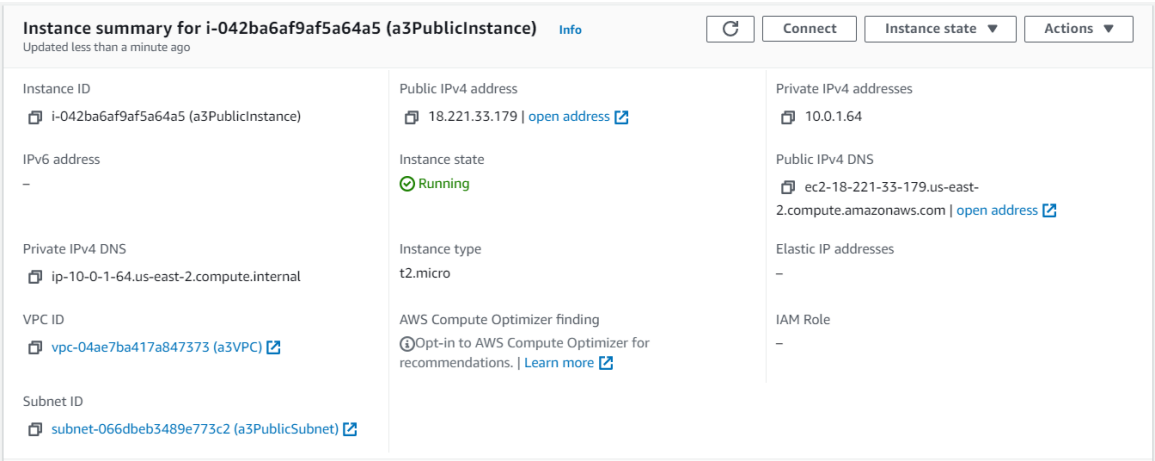
```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
mkdir var
cd /var
mkdir www
cd /var/www
mkdir html
cd /var/www/html
touch index.html
cat << EOF > index.html
<!DOCTYPE html>
<html lang="en">
<body>
<h1 style="color:MediumTurquoise; font-family:Broadway; font-size:500%; text-
align:center">Private Subnet</h1>
</body>
</html>
EOF
```

- Step 4: Add Storage: Keep the default selection of 8 GiB General Purpose SSD (gp2)
- Step 5: Add Tags: Key 'Name', Value 'a3PrivateInstance'
- Step 6: Select an existing security group: sg-0d559e64f02375dbc (default)
- Step 7: Review Instance Launch
- Select the key pair 'a3Key'
- Launch a3PrivateInstance



3. Create an instance in a public subnet

- Launch an instance in a3PublicSubnet
- Step 1: Select 'Amazon Linux 2 AMI (HVM) SSD Volume Type 64-bit (x86)'
- Step 2: Choose the Instance Type 't2.micro'
- Step 3: Configure instance details:
    - Network: a3VPC
    - Subnet: a3PublicSubnet
    - Auto-assign Public IP: Use subnet setting (Enable)
- Step 4: Add Storage: Keep the default selection of 8 GiB General Purpose SSD (gp2)
- Step 5: Add Tags: Key 'Name', Value 'a3PublicInstance'
- Step 6: Select an existing security group: sg-0d559e64f02375dbc (default)
- Step 7: Review Instance Launch
- Select the key pair 'a3Key' and launch a3PublicInstance



- Access a3PrivateInstance from a3PublicInstance
- On my Windows device, open Pageant and add the a3Key key pair
- Open SSH PuTTY:
    - Host Name: use the public DNS address of a3PublicInstance: ec2-18-221-33-179.us-east-2.compute.amazonaws.com
    - In SSH-Auth, click 'Allow agent forwarding' and add the key pair a3Key
    - In the PuTTY terminal, log in as 'ec2-user'
    - Access a3PrivateInstance using its private IP address: ssh ec2-user@10.0.0.99

```
ec2-user@ip-10-0-0-99:~
login as: ec2-user
Authenticating with public key "a3Key" from agent
Last login: Sat Sep 25 01:47:21 2021 from 1.144.111.146

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-64 ~]$ ssh ec2-user@10.0.0.99
Last login: Sat Sep 25 01:47:37 2021 from ip-10-0-1-64.us-east-2.compute.interna
l

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-99 ~]$
```

- Display the directory structure for the webserver

```
ec2-user@ip-10-0-0-99:/var/www/html
[ec2-user@ip-10-0-0-99 ~]$ cd /var/www/html
[ec2-user@ip-10-0-0-99 html]$ ls
index.html
[ec2-user@ip-10-0-0-99 html]$
```

- Display the HTML code for the index file

vim index.html

```
ec2-user@ip-10-0-0-99:/var/www/html
<!DOCTYPE html>
<html lang="en">
<body>
<h1 style="color:MediumTurquoise; font-family:Broadway; font-size:500%; text-ali
gn:center">Private Subnet</h1>
</body>
</html>
```

4. Create an RDS instance in a private subnet

- An RDS instance requires a Subnet Group with subnets in at least two availability zones
- Create another private subnet in a new availability zone
    - Navigate to VPC Dashboard, click “Subnets”, “Create subnet”
        - VPC: a3VPC
        - Availability Zone: US East (Ohio) / us-east-2b
        - IPv4 CIDR block: 10.0.2.0/24
        - Name: a3PrivateSubnetB

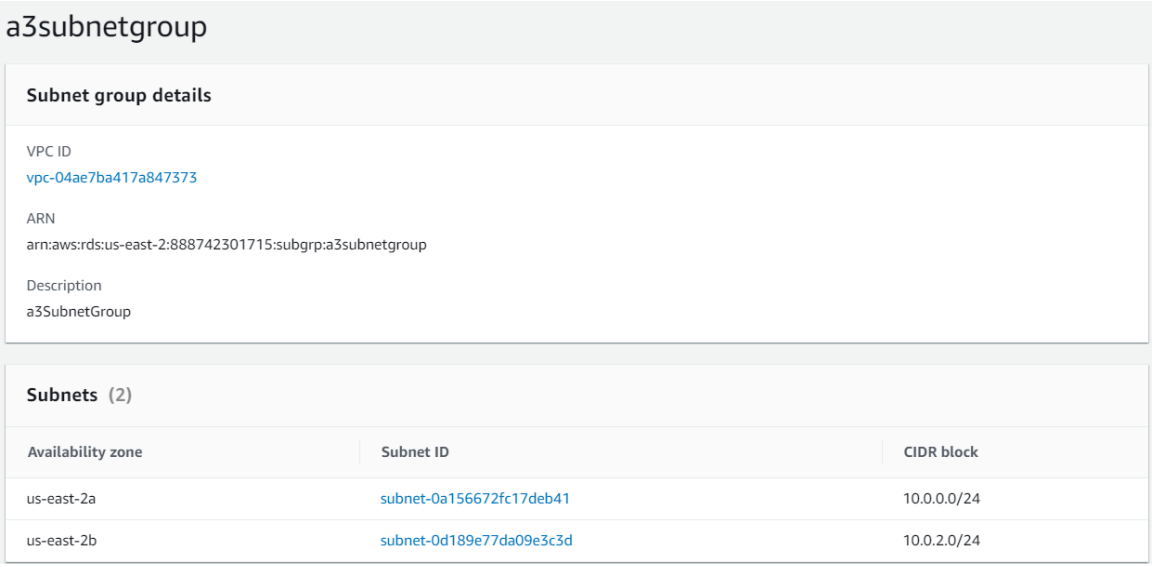
subnet-0d189e77da09e3c3d / a3PrivateSubnetB

Actions

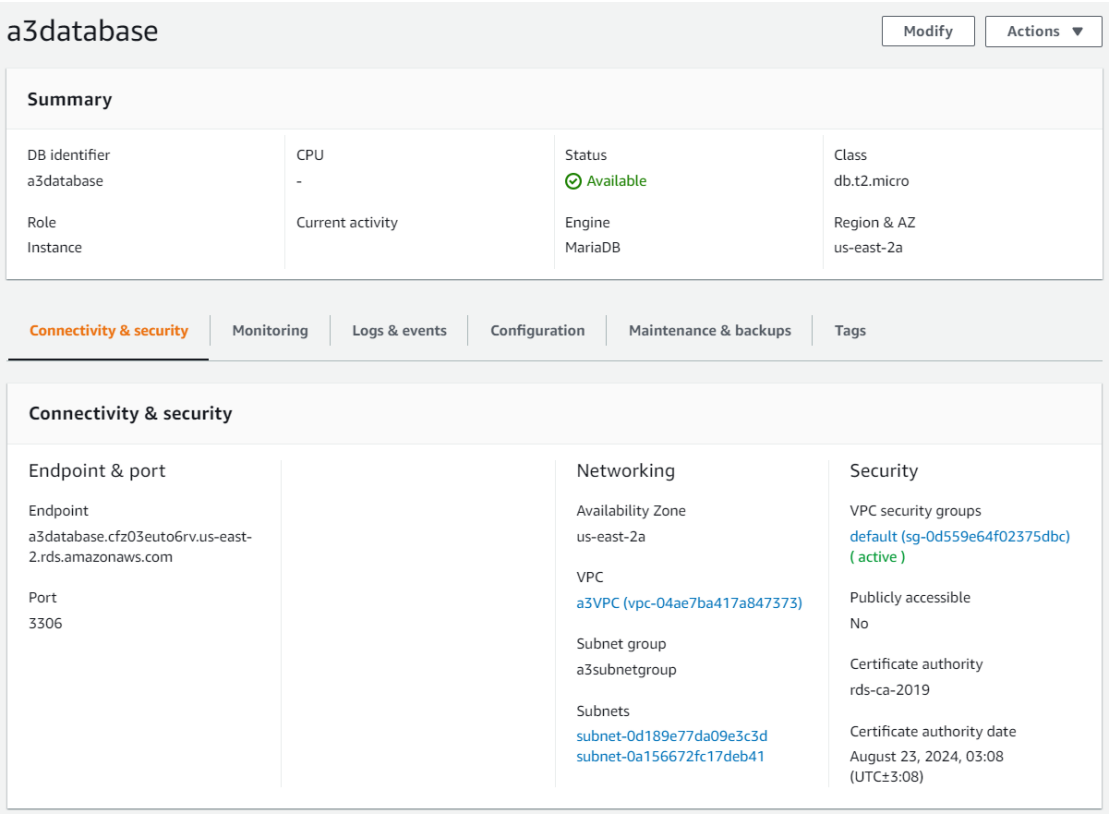
Details

Subnet ID subnet-0d189e77da09e3c3d	Subnet ARN arn:aws:ec2:us-east-2:888742301715:subnet/subnet-0d189e77da09e3c3d	State Available	IPv4 CIDR 10.0.2.0/24
Available IPv4 addresses 251	IPv6 CIDR –	Availability Zone us-east-2b	Availability Zone ID use2-az2
VPC vpc-04ae7ba417a847373 | a3VPC	Route table rtb-02e41504d2bfba623 | a3PrivateRouteTable	Network ACL acl-0cf7dad64dee3bd32	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 –
Outpost ID –	IPv4 CIDR reservations –	IPv6 CIDR reservations –	Owner 888742301715

- Navigate to RDS Dashboard, click “Subnet Groups”, “Create DB Subnet Group”
    - Name: a3SubnetGroup
    - Description: a3SubnetGroup
    - VPC: a3VPC
    - Availability Zones: us-east-2a & us-east-2b
    - Subnets: a3PrivateSubnet & a3PrivateSubnetB



- Navigate to RDS Dashboard, click “Databases”, “Create database”
    - Database creation method: Standard create
    - Engine options: MariaDB 10.4.13
    - Templates: Free tier
    - Settings:
        - DB instance identifier: a3Database
        - Master username: admin
        - Master password: ********
        - DB instance class: db.t2.micro
        - Storage: General Purpose SSD (gp2) 20 GiB
        - VPC: a3VPC
        - Subnet group: a3subnetgroup
        - Public access: No
        - VPC security group: Choose existing: default
        - Availability Zone: us-east-2a
        - Database port: 3306



5. Access the database and create a table

- I access a3Database from a3PublicInstance
- Open SSH PuTTY: Host Name: ec2-user@18.221.33.179
- Install the MySQL command-line client

```
sudo su
yum install mariadb
```

```
root@ip-10-0-1-64:/home/ec2-user
Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2

Complete!
[root@ip-10-0-1-64 ec2-user]#
```

- Connect to a3Database using the database endpoint

```
mysql -h a3database.cfz03euto6rv.us-east-2.rds.amazonaws.com -P 3306 -u admin -p
```

```
root@ip-10-0-1-64:/home/ec2-user
[root@ip-10-0-1-64 ec2-user]# mysql -h a3database.cfz03euto6rv.us-east-2.rds.ama
zonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.4.13-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

- Create a table

```
create database student_registrations;
use student_registrations;
create table students (Name varchar(255), sID int);
insert into students (Name, sID) values ('Joe', 20221177);
show tables;
select * from students;
```

```
root@ip-10-0-1-64:/home/ec2-user
MariaDB [student_registrations]> show tables;
+---------------------------------+
| Tables_in_student_registrations |
+---------------------------------+
| students                        |
+---------------------------------+
1 row in set (0.00 sec)

MariaDB [student_registrations]> select * from students;
+------+----------+
| Name | sID      |
+------+----------+
| Joe  | 20221177 |
+------+----------+
1 row in set (0.00 sec)

MariaDB [student_registrations]>
```