Creation of the customized VPC with Terraform able to communicate EC2 to RDS communication

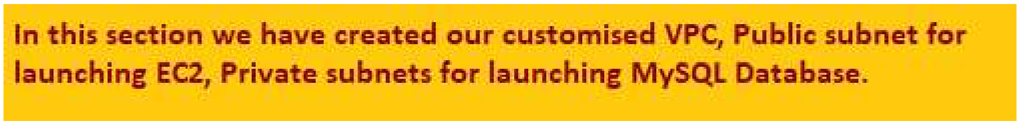
Amazon RDS

Amazon Relational Database Service (Amazon RDS) is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity, while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.

variable.tf

```
variable "access_key" {
   type = string
   description = "accesskey"
   default= "AKIA6LFFBWBRQ7YNMV2"
}

variable "region" {
   default = "us-east-1"
}

variable "secret_key" {
   type = string
   description = "secretkey"
   default= "ewjax303Mtck8MOJ28dUDdZKYmooUaDVPgQ19I3"
}
```

VPC.tf

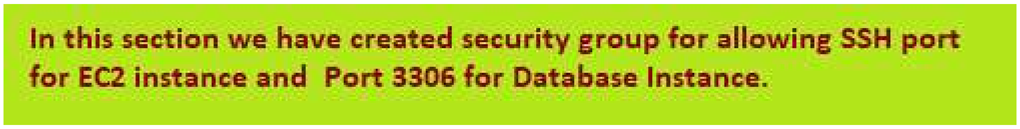


In this section we have created our customised VPC, Public subnet for launching EC2, Private subnets for launching MySQL Database.

```
resource "aws_vpc" "vpc" {
  cidr_block = "192.178.0.0/16"

  tags = {
     Name = "MY-VPC"
   }
```

```
}
#PUBLIC SUBNET
resource "aws_subnet" "public" {
  vpc_id     = aws_vpc.vpc.id
  map_public_ip_on_launch = true
  cidr_block = "192.178.1.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "public subnet"
  }
}

# internet gateway for public subnet

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.vpc.id

  tags = {
    Name = "internet-gateway-vpc"
  }
}

#root table and assosiation with subnet

resource "aws_route_table" "route-public" {
  vpc_id = aws_vpc.vpc.id

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

  tags = {
    Name = "public-route-table"
  }
}

resource "aws_route_table_association" "public" {
  subnet_id      = aws_subnet.public.id
  route_table_id = aws_route_table.route-public.id
}

#Private subnet
resource "aws_subnet" "private" {
  vpc_id     = aws_vpc.vpc.id
  cidr_block = "192.178.2.0/24"
  availability_zone = "us-east-1b"
  map_public_ip_on_launch = false

  tags = {
    Name = "Private subnet"
  }
}
resource "aws_subnet" "private2" {
  vpc_id     = aws_vpc.vpc.id
  cidr_block = "192.178.3.0/24"
  availability_zone = "us-east-1c"
  map_public_ip_on_launch = false

  tags = {
    Name = "Private subnet2"
  }
}
resource "aws_route_table" "private" {
  vpc_id = aws_vpc.vpc.id

}
resource "aws_route" "private_nat_gateway" {
  route_table_id = aws_route_table.private.id
  destination_cidr_block = "0.0.0.0/0"
  nat_gateway_id = aws_nat_gateway.nat.id
}
resource "aws_nat_gateway" "nat" {
  connectivity_type = "private"
  subnet_id     = aws_subnet.public.id

}
resource "aws_route_table_association" "private1" {
  subnet_id      = aws_subnet.private.id
  route_table_id = aws_route_table.private.id
}
resource "aws_route_table_association" "private2" {
  subnet_id      = aws_subnet.private2.id
  route_table_id = aws_route_table.private.id
}
```

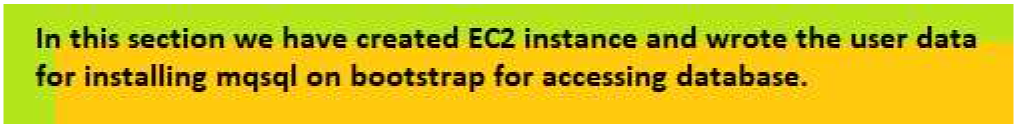
SecurityGroup.tf



In this section we have created security group for allowing SSH port for EC2 instance and Port 3306 for Database Instance.

```
resource "aws_security_group" "sg_vpc" {
  name        = "allow_SSH"
  description = "Allow SSH inbound traffic"
  vpc_id      = aws_vpc.vpc.id

  ingress {
    # SSH Port 22 allowed from any IP
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

   ingress {
    # Mysql Port 3306 allowed from any IP
    from_port   = 3306
    to_port     = 3306
    protocol    = "tcp"
    cidr_blocks = ["192.178.0.0/16"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

EC2.tf



In this section we have created EC2 instance and wrote the user data for installing mqsql on bootstrap for accessing database.

```
provider "aws" {
    region= var.region
    access_key = var.access_key
    secret_key = var.secret_key
}
resource "aws_instance" "ec2" {
 ami = "ami-05fa00d4c63e32376"
 instance_type = "t2.micro"
 subnet_id = aws_subnet.public.id
 # Security group assign to instance
 vpc_security_group_ids = [aws_security_group.sg_vpc.id]

 # key name
 key_name = "08-09-2022"

 user_data = <<EOF
                        #! /bin/bash
   # Update all packages
   yum -y update
   # Install mysql client
   yum -y install mysql
              EOF

 tags = {
   Name = "Ec2 "
 }
}
```

main.tf

In this section we have created MySQL Database with customised private subnets username "mysqldb" and Password "mysqldb20200". Also we have created subnet group containing 2 private subnets.

```
resource "aws_db_instance" "my_db" {
 allocated_storage    = 10
 engine               = "mysql"
 engine_version       = "5.7"
 instance_class       = "db.t3.micro"
 db_name              = "mysqldb"
 username             = "mysqldb"
 password             = "mysqldb2022"
 skip_final_snapshot  = true
 db_subnet_group_name = aws_db_subnet_group.private.name
 vpc_security_group_ids = [aws_security_group.sg_vpc.id]
}

resource "aws_db_subnet_group" "private" {
   name        = "subnet_group_for_rds "
   description = "Our main group of subnets"
```

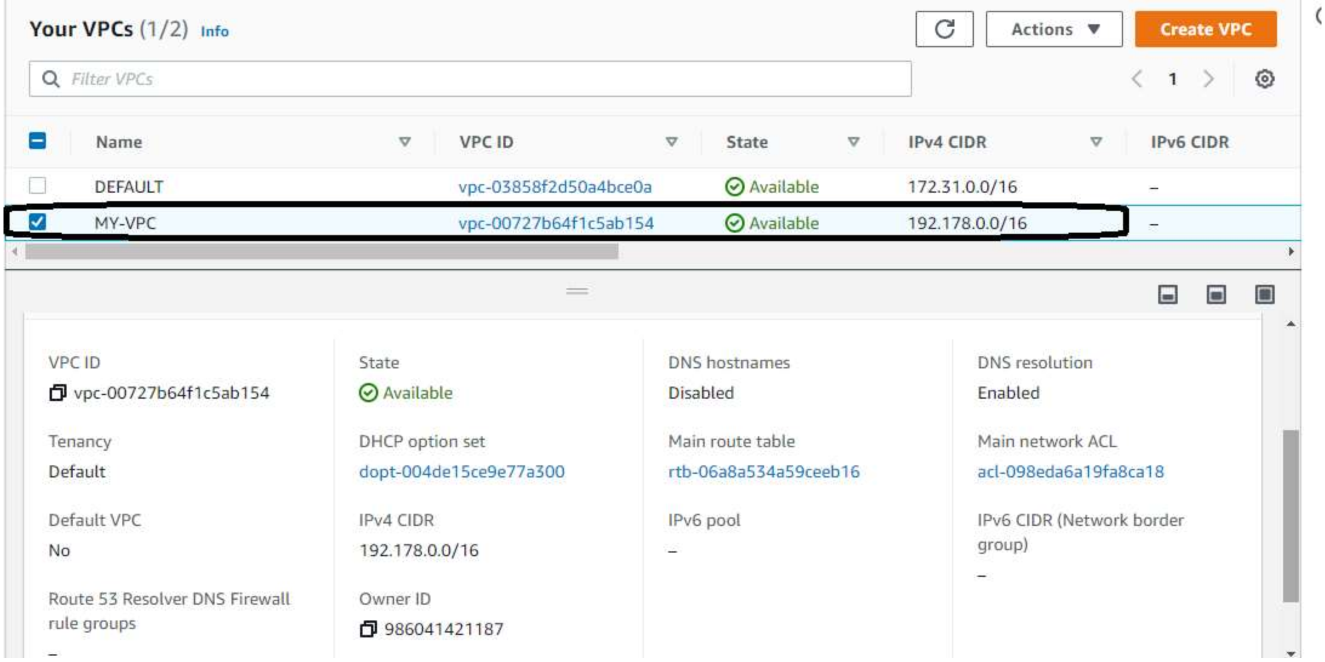
```
  subnet_ids  = [aws_subnet.private.id, aws_subnet.private2.id]
}
```

Output.tf

In this section we get required output of resources like, Public IP of EC2 Instance, endpoint for connecting RDS Database

```
output "instance_public_ip" {
 description = "Public IP address of the EC2 instance"
 value       = aws_instance.ec2.public_ip
}
output "db_endpoint" {
 description = "Database Endpoint"
 value = aws_db_instance.my_db.endpoint
}
```

Screenshots



Your VPCs (1/2)

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	DEFAULT	vpc-03858f2d50a4bce0a	✓ Available	172.31.0.0/16	–
☑	MY-VPC	vpc-00727b64f1c5ab154	✓ Available	192.178.0.0/16	–

VPC ID
vpc-00727b64f1c5ab154

State
✓ Available

DNS hostnames
Disabled

DNS resolution
Enabled

Tenancy
Default

DHCP option set
dopt-004de15ce9e77a300

Main route table
rtb-06a8a534a59ceeb16

Main network ACL
acl-098eda6a19fa8ca18

Default VPC
No

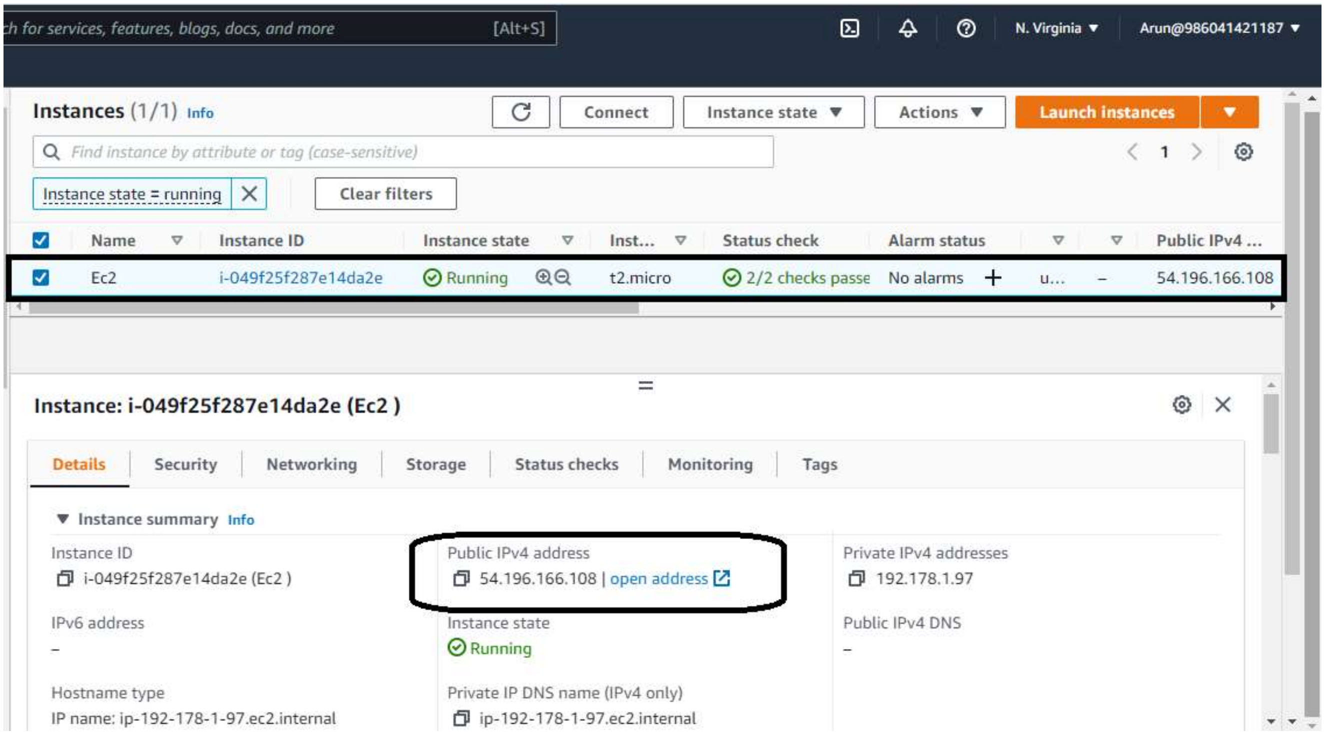
IPv4 CIDR
192.178.0.0/16

IPv6 pool
–

IPv6 CIDR (Network border group)
–

Route 53 Resolver DNS Firewall rule groups
–

Owner ID
986041421187



Instances (1/1) Info

Instance state = running

☑	Name	Instance ID	Instance state	Inst...	Status check	Alarm status			Public IPv4 ...
☑	Ec2	i-049f25f287e14da2e	✓ Running	t2.micro	✓ 2/2 checks passe	No alarms +	u...	–	54.196.166.108

Instance: i-049f25f287e14da2e (Ec2)

▼ Instance summary Info

Instance ID
i-049f25f287e14da2e (Ec2)

Public IPv4 address
54.196.166.108 | open address

Private IPv4 addresses
192.178.1.97

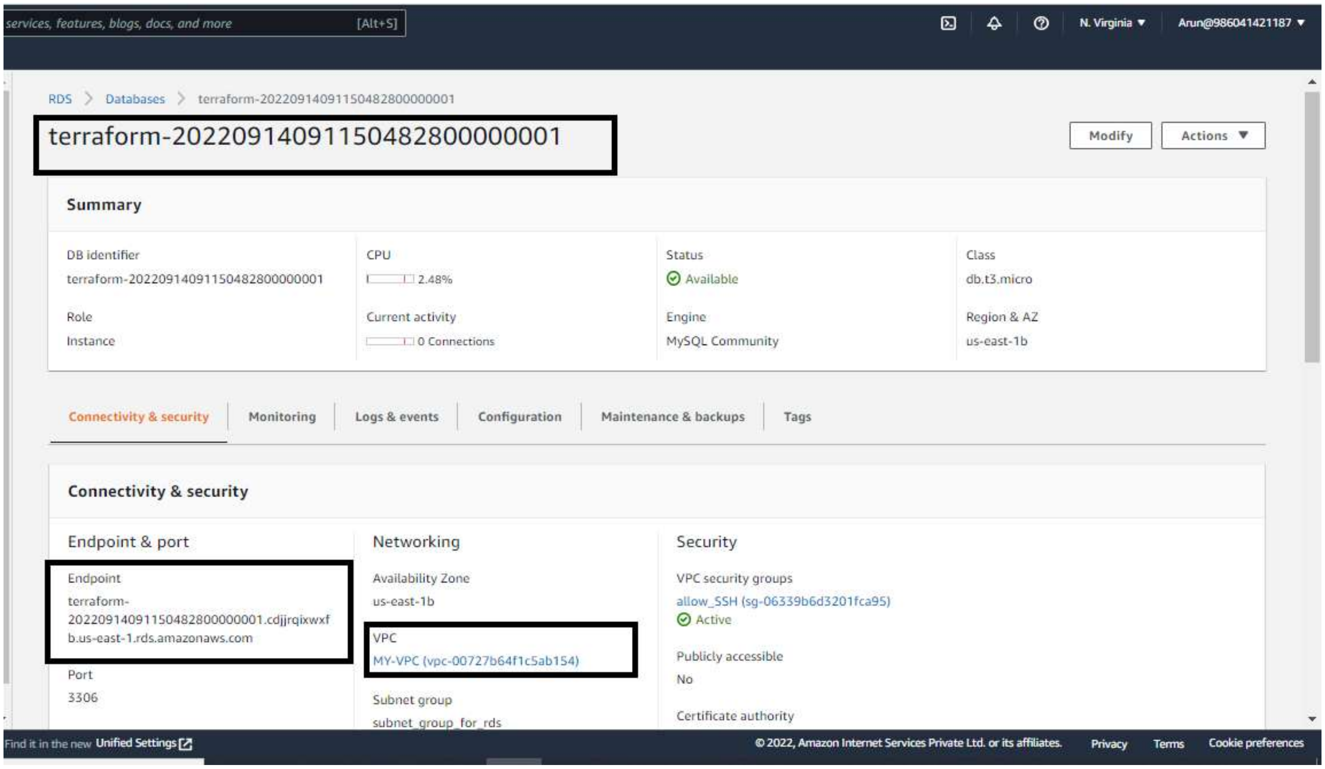
IPv6 address
–

Instance state
✓ Running

Public IPv4 DNS
–

Hostname type
IP name: ip-192-178-1-97.ec2.internal

Private IP DNS name (IPv4 only)
ip-192-178-1-97.ec2.internal



services, features, blogs, docs, and more [Alt+S]
N. Virginia ▼
Arun@986041421187 ▼
RDS > Databases > terraform-20220914091150482800000001
Modify
Actions ▼
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

terraform-20220914091150482800000001

Summary

DB identifier
terraform-20220914091150482800000001

CPU
2.48%

Status
✓ Available

Class
db.t3.micro

Role
Instance

Current activity
0 Connections

Engine
MySQL Community

Region & AZ
us-east-1b

Connectivity & security

Endpoint & port

Endpoint
terraform-20220914091150482800000001.cdjjrqixwxfb.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1b

VPC
MY-VPC (vpc-00727b64f1c5ab154)

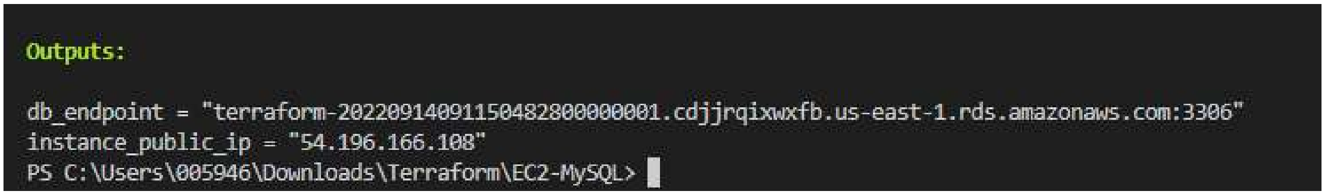
Subnet group
subnet_group_for_rds

Security

VPC security groups
allow_SSH (sg-06339b6d3201fca95)
✓ Active

Publicly accessible
No

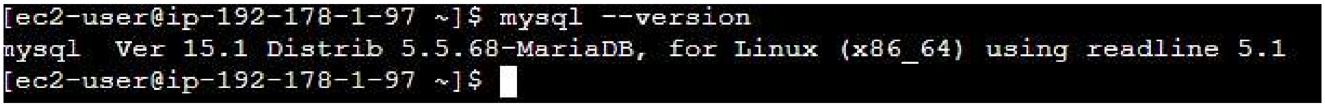
Certificate authority



```
Outputs:

db_endpoint = "terraform-20220914091150482800000001.cdjjrqixwxfb.us-east-1.rds.amazonaws.com:3306"
instance_public_ip = "54.196.166.108"
PS C:\Users\005946\Downloads\Terraform\EC2-MySQL>
```

Connect to EC2 instance and check version of mysql

mysql --version



```
[ec2-user@ip-192-178-1-97 ~]$ mysql --version
mysql  Ver 15.1 Distrib 5.5.68-MariaDB, for Linux (x86_64) using readline 5.1
[ec2-user@ip-192-178-1-97 ~]$
```

connect to databse

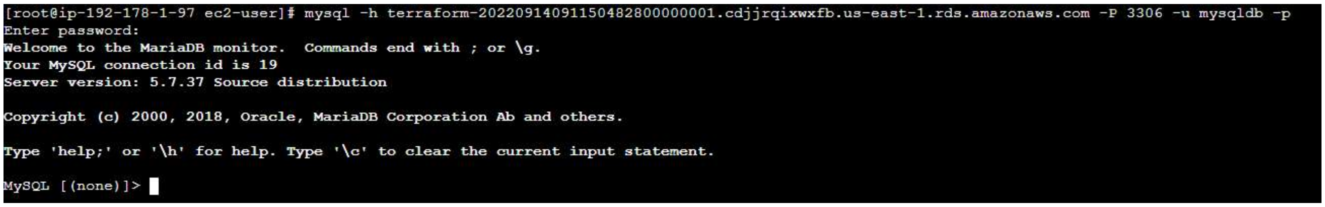
mysql -h <endpoint> -P 3306 -u <username> -p

endpoint -- terraform-20220914091150482800000001.cdjjrqixwxfb.us-east-1.rds.amazonaws.com

username --mysqldb

Password -- mysqldb2022

mysql -h terraform-20220914091150482800000001.cdjjrqixwxfb.us-east-1.rds.amazonaws.com -P 3306 -u mysqldb -p



```
[root@ip-192-178-1-97 ec2-user]# mysql -h terraform-20220914091150482800000001.cdjjrqixwxfb.us-east-1.rds.amazonaws.com -P 3306 -u mysqldb -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.7.37 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.

MySQL [(none)]>
```

Successfully Completed