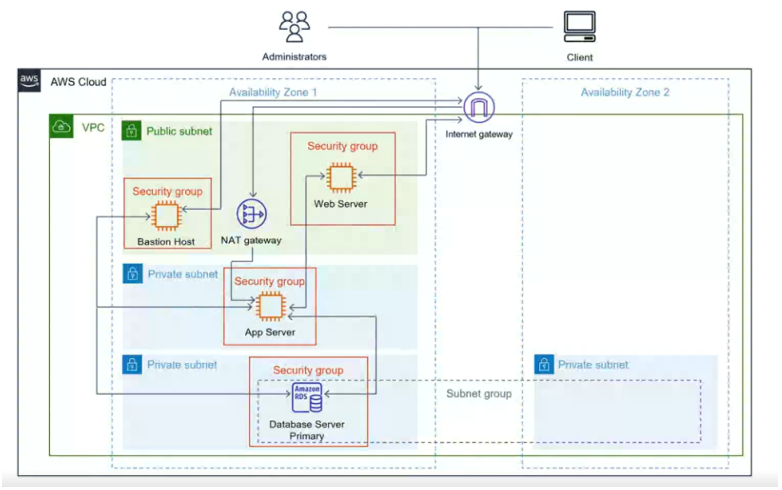
Design and configure a high available 3-tier Architecture on AWS

Type	Post Graduation
Tasks Achieved	• Tier 1 - User/Presentation Tier • Tier 2 - Application Tier • Tier 3 - Data Tier

Tasks Achieved



▼ VPC - CIDR Block be 10.1.0.0/16

- 4 subnets (1 public, 3 private)
- Enable in subnet settings public IP addresses
- Make it highly available (use 2 availability zones, the final private subnet can be the only one in a different subnet)
- Allocate an Elastic IP
- Create a NAT gateway
- Create an internet gateway and attach it to your VPC
- Make route tables for your public and private subnets and attach an internet gateway and NAT gateway to them respectively
- Make security groups for Bastion Host, web server, app server, and database
- Make sure to go back to security groups after making them and adding security groups to link them together, for example in the app server security group adding a rule for the database security group after creating the database security group.
  - If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes DNS hostnames and DNS resolution.

▼ EC2 Instances

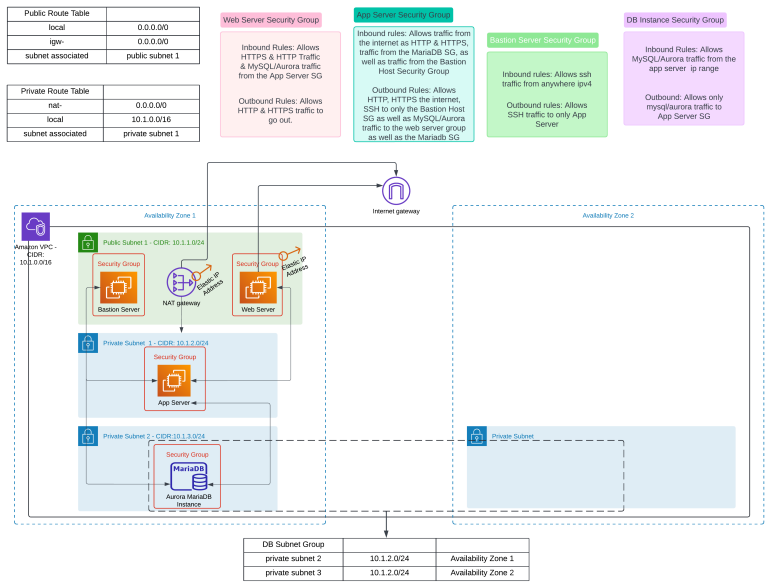
Bastion Host

- Amazon Linux 2 ami
- T2 Micro
- Use VPC & public subnet
- Use security group enable ssh

Web Server

- Amazon Linux 2 ami
- T2 Micro

▼ Built Architecture



▼ POCs

```
Using username "ec2-user".
Authenticating with public key "bastion ho
st key pair"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.co
m/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-1-1-155 ~]$ ls
[ec2-user@ip-10-1-1-155 ~]$ touch appserve
r.pem
[ec2-user@ip-10-1-1-155 ~]$ chmod 400 apps
erver.pem
[ec2-user@ip-10-1-1-155 ~]$ ls -la appserv
er.pem
```

- Use VPC & public subnet
- Use security group created in the VPC SETUP
- User Data

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install -y lamp-
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

App Server

- Amazon Linux 2 ami
- T2 Micro
- Use VPC & public subnet
- Use security group created in the VPC setup enable ssh
- User Data

```
#!/bin/bash
sudo yum install -y mariadb-server
sudo service mariadb start
```

Create DB Instance

- Create a subnet group
- DB Instance
  - Standard create
  - Mariadb
  - Free tier
  - Disable backups & encryption

```
user = root
password = Re:Start!9
initial Database: mydb
```

```
-r--------. 1 ec2-user ec2-user 0 Oct  6 1
1:51 appserver.pem
[ec2-user@ip-10-1-1-155 ~]$ chmod +w appse
rver.pem
[ec2-user@ip-10-1-1-155 ~]$ sudo vi appser
ver.pem
[ec2-user@ip-10-1-1-155 ~]$ sudo ssh -i ap
pserver.pem ec2-user@10.1.2.222
The authenticity of host '10.1.2.222 (10.
1.2.222)' can't be established.
ED25519 key fingerprint is SHA256:FVW12hvf
OTsFvxC0flmKVST38qUjmCnIPrgPRJKZlMU.
This key is not known by any other names
Are you sure you want to continue connecti
ng (yes/no/[fingerprint])? yes
Warning: Permanently added '10.1.2.222' (E
D25519) to the list of known hosts.
```

```
[ec2-user@ip-10-1-2-222 ~]$ mysql -u root
-h mariadb-3tier.cafrbt5swsel.us-west-2.rd
s.amazonaws.com -p
Enter password:
Welcome to the MariaDB monitor.  Commands
end with ; or \g.
Your MariaDB connection id is 51
Server version: 10.11.8-MariaDB managed by
https://aws.amazon.com/rds/

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

Type 'help;' or '\h' for help. Type '\c' t
o clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.001 sec)

MariaDB [(none)]>
```

Web Server HTTP Connection