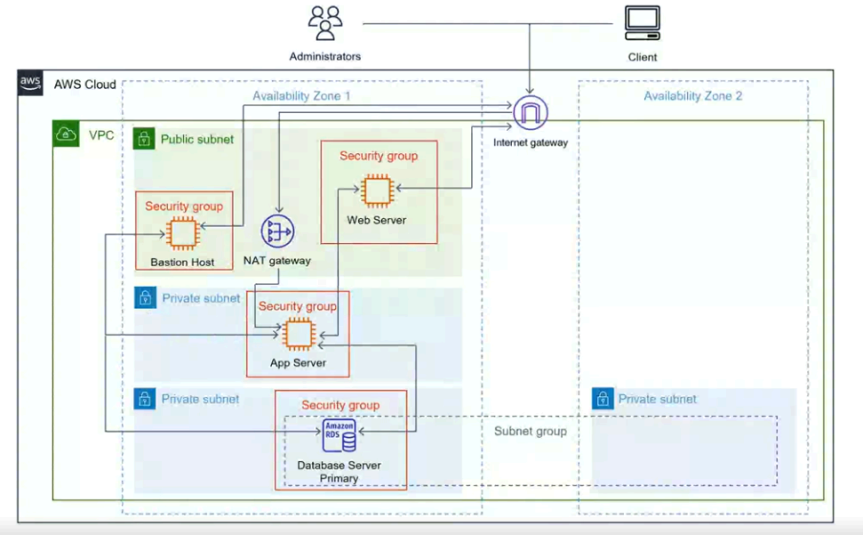
Design and configure a high available 3-tier Architecture on AWS

Type	AWS ARCHITECTURE
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)

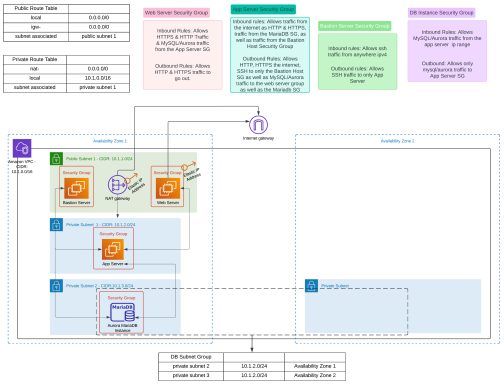
▼ Built Architecture

- 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



▼ POCs

```
Using username "ec2-user".
Authenticating with public ke
y "bastion host key pair"
  ,     #_
  ~\_  ####_       Amazon Lin
ux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.a
mazon.com/linux/amazon-lin
ux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-1-1-155 ~]$
ls
[ec2-user@ip-10-1-1-155 ~]$
touch appserver.pem
[ec2-user@ip-10-1-1-155 ~]$
chmod 400 appserver.pem
[ec2-user@ip-10-1-1-155 ~]$
ls -la appserver.pem
-r--------. 1 ec2-user ec2-us
er 0 Oct  6 11:51 appserver.pe
m
[ec2-user@ip-10-1-1-155 ~]$
```

Web Server

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

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras ir
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·
sudo service mariadb start
```

Create DB Instance

- Create a subnet group
- DB Instance
  - Standard create
  - Mariadb

```
chmod +w appserver.pem
[ec2-user@ip-10-1-1-155 ~]$
sudo vi appserver.pem
[ec2-user@ip-10-1-1-155 ~]$
sudo ssh -i appserver.pem e
c2-user@10.1.2.222
The authenticity of host '10.1.
2.222 (10.1.2.222)' can't be e
stablished.
ED25519 key fingerprint is S
HA256:FVW12hvfOTsFvxC0fl
mKVST38qUjmCnIPrgPRJKZI
MU.
This key is not known by any
other names
Are you sure you want to con
tinue connecting (yes/no/[fin
gerprint])? yes
Warning: Permanently added
'10.1.2.222' (ED25519) to the l
ist of known hosts.
```

```
[ec2-user@ip-10-1-2-222 ~]
$ mysql -u root -h <hostnam
e> -p
Enter password:
Welcome to the MariaDB mon
itor.  Commands end with ; or
\g.
Your MariaDB connection id i
s 51
Server version: 10.11.8-Maria
DB managed by https://aws.a
mazon.com/rds/

Copyright (c) 2000, 2018, Or
acle, MariaDB Corporation A
b and others.
```

- Free tier
- Disable backups & encryption

```
user = <username>
password = <password>
initial Database: mydb
```

```
Type 'help;' or '\h' for help. T
ype '\c' to clear the current in
put statement.

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

MariaDB [(none)]>
```

Web Server HTTP Connection