Table of contents

What is RDS and why RDS :

- RDS stands for relational database system and it is a native service of AWS .
- RDS is a managed services . Managed service means AWS takes care of most of the components and we can focus on just the database part
- With RDS we only get an endpoint to connect to the database , we cannot ssh into the server .
- Our role from the infra side will be taking care of the provisioning and maintaning the RDS servers . There are tons of things which can be utilized from the database perspective which we just need to be aware of

RDS components

Subnet group

- Subnets are segments of a VPC's IP address range that you designate to group your resources based on security and operational needs. A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances using the CLI or API; if you use the console, you can just choose the VPC and subnets you want to use.

- This basically dictates which VPC and subnet the DB will be created in

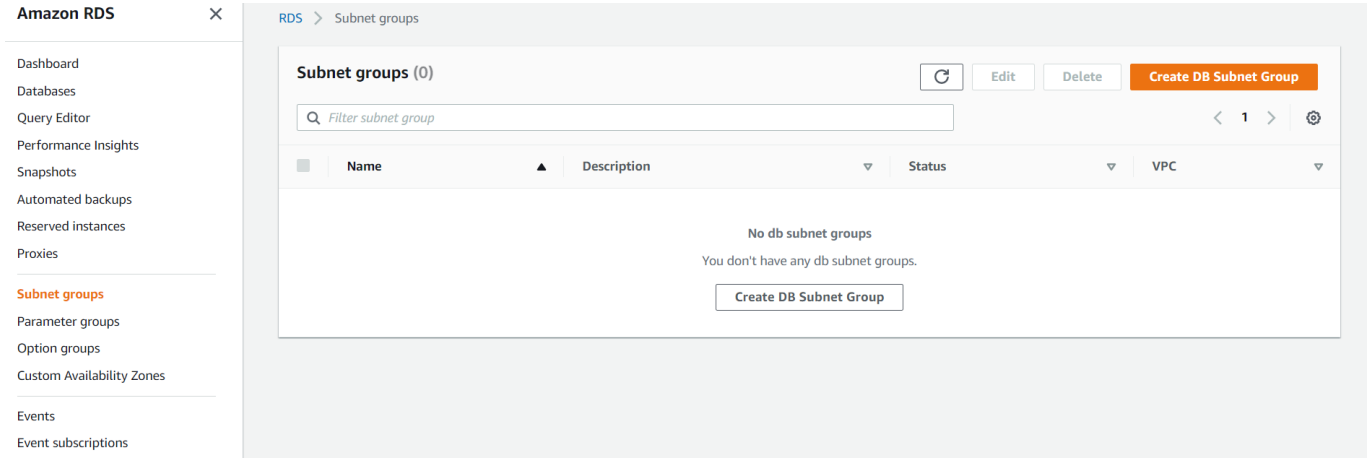
- Each DB subnet group should have subnets in at least two Availability Zones in a given AWS Region. When creating a DB instance in a VPC, you must choose a DB subnet group. Amazon RDS uses that DB subnet group and your preferred Availability Zone to choose a subnet and an IP address within that subnet to associate with your DB instance. If the primary DB instance of a Multi-AZ deployment fails, Amazon RDS can promote the corresponding standby and subsequently create a new standby using an IP address of the subnet in one of the other Availability Zones.

- The subnets in a DB subnet group are either public or private. They can't be a mix of both public and private subnets. The subnets are public or private, depending on the configuration that you set for their

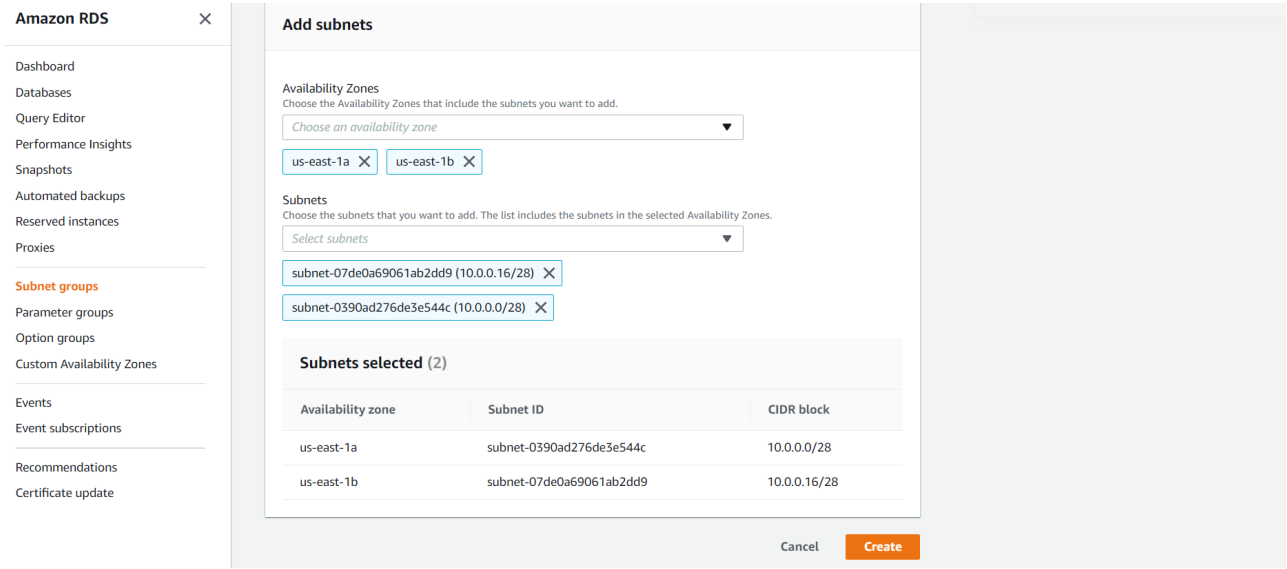
network access control lists (network ACLs) and routing tables

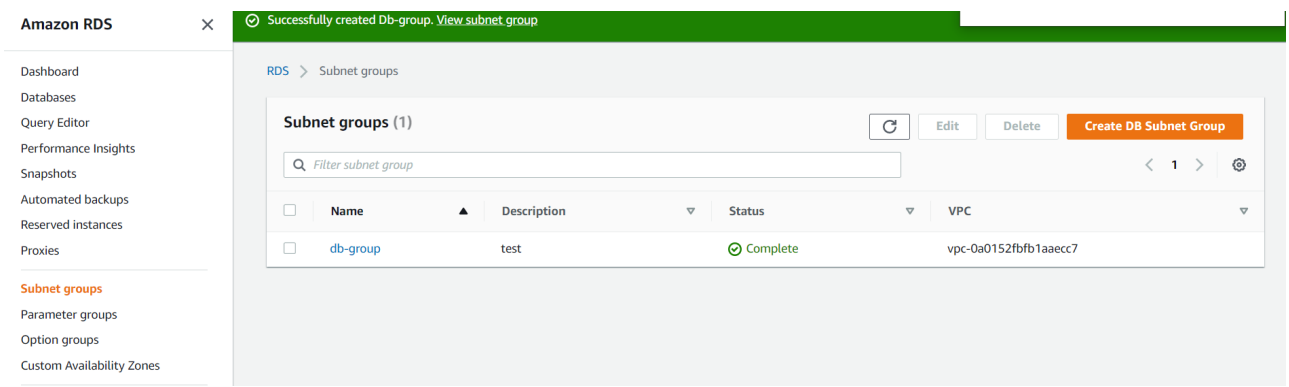
Creating a subnet group

- Navigate to RDS service . Select subnet group



- Click on create DB subnet group . Grant a name to the group and description
- Select a VPC
- While selecting the availability zone, make sure you select at least two AZs .
- Once the AZs are selected , select the subnets associated with it . You must select private subnets only as we do not want our RDS to be accessible from the internet

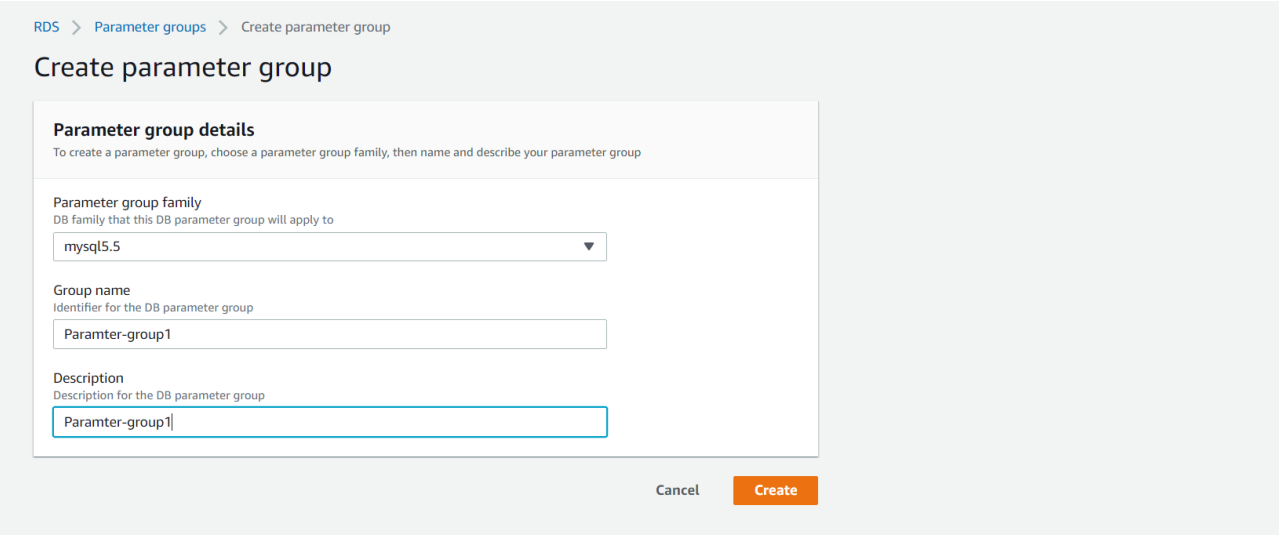




- We will use this subnet group later while creating the RDS

Paramter group

- You manage your DB engine configuration by associating your DB instances with parameter groups. Amazon RDS defines parameter groups with default settings that apply to newly created DB instances
- A DB parameter group acts as a container for engine configuration values that are applied to one or more DB instances.
- If you create a DB instance without specifying a DB parameter group, the DB instance uses a default DB parameter group. Each default DB parameter group contains database engine defaults and Amazon RDS system defaults based on the engine, compute class, and allocated storage of the instance. You can't modify the parameter settings of a default parameter group. Instead, you create your own parameter group where you choose your own parameter settings. Not all DB engine parameters can be changed in a parameter group that you create.
- If you want to use your own parameter group, you create a new parameter group and modify the parameters that you want to. You then modify your DB instance to use the new parameter group. If you update parameters within a DB parameter group, the changes apply to all DB instances that are associated with that parameter group.
- Paramater group along with option group is primarily meant for database developers and DBAs . From the infra side we provision them according to the requirement
- **Creating parameter group**
- Select paramater group from the navigation pane
- Click create parameter group
- Select the database engine that you are planning to use for ex - mysql 5.5
- give a suitable name and click on create



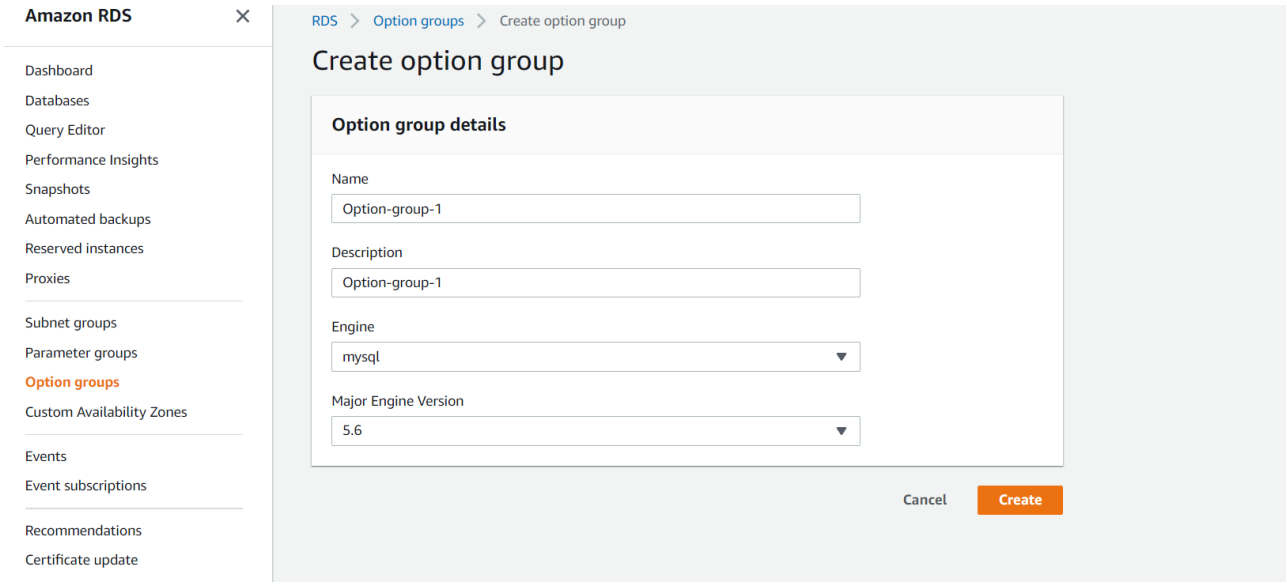
- Once the parameter group is created , it can be modified to add various mysql db engine level parameters for ex. autocommit

Option group

- Some DB engines offer additional features that make it easier to manage data and databases, and to provide additional security for your database. Amazon RDS uses option groups to enable and configure these features. An option group can specify features, called options, that are available for a particular Amazon RDS DB instance. Options can have settings that specify how the option works. When you associate a DB instance with an option group, the specified options and option settings are enabled for that DB instance.

Creating option group

- Similar to parameter group , click on option groups from the pane .
- Click on create option group
- After givig the name and description select the db engine and the version for ex mysql and engine version 5.6
- Click on create



- Similar to the paramter group , option group could be utilized for adding additional options for ex audit file path

Creating a database instance

- As we have seen already RDS is a managed service . Where we as users only get a database to use and AWS takes care of the rest of the things . for ex : OS level patching etc
- Go to RDS service and click on create database
- Chose standard create
- As of now we'll stick to mysql as the db engine option
- For the version , we'll go with 5.7.31
- On templates , there are 3 options . Production , Dev/test, Free tier . These are predefined standard practices recomended by aws for various use cases
- As of now lets select free tier .
- Keep the username and password as desired
- For DB instance size , let us keep the default value of db.t2.micro . This is exactly similar to chosing your ec2 size .

- For storage , this again is similar to your EC2. We have an additional feature of automatically scaling the storage .
- For allocated storage we can keep the default 20 GB as storage . We can tick the "Enable storage autoscaling " and set the maximum limit as 1000. It basically means that the initial storage will be 20Gb , and the storage can scale upto 1000 Gbs. Remember it is onlly the storage power that is scaling and not the processing power .
- Avalability and durability dictates if Multi-AZ deployment where the database basically launches a standby instance which automaticaly comes up when primary server goes down . For free tier this option is disabled.
- The RDS similar to EC2 is launched in the VPC . Under connectivity select the VPC and click on "additional connectivity configuration "
- Select the subnet group we have created before . This dictates in which subnet the RDS will be launched .
- Always make sure the "Public Access" setting is always set to No . As we'd never want our database to be accessed by internet
- Select the security group . As of now any will do . WE'll modify it later
- Avalability zone we can keep as no preference . It will be selected from the subnet group
- Database port by default for mysql is 3306 . It can be changed if needed. We'll keep it default
- For database Authentication we'll stick to password authentication.
- Expand additional configuration
- We can grant a name to the database , it is optional . If we dont give it any name then a default database will be created
- Here we have option to set our paramamter group and option group which are optional
- Backup we can enable "Enable automatic backups " to have automated snapshots taken of yur database . These are same as the volume snapshots . Using these we can directly restore an RDS .
- The Backup retention period is similar to the lifecycle of snapshots
- Backup window is a time frame we can provide when the backup will be taken . This should be always non business hours as the snapshot process takes away some amount of processing power and we do not want our users to face latency
- Under monitoring
    - Enhanced monitoring enables extra graphs for detailed insights
    - Log exports sends out logs to cloudwatch logs . There are multiple types of logs from which we can decide
    - Maintenence decides when the patching activity and other maintenence level activities are done . This is something AWS takes care of
    - "Enable auto-minor update " will allow AWS to update the db engine version whenever its about to expire
    - Maintenence window is similar to backup window. The updates and patching will be done in this window only
    - Deletion protection is again similar to Ec2 . It protects from accidental termination . If it is desabled only then RDS will be terminated
    - Click on create database

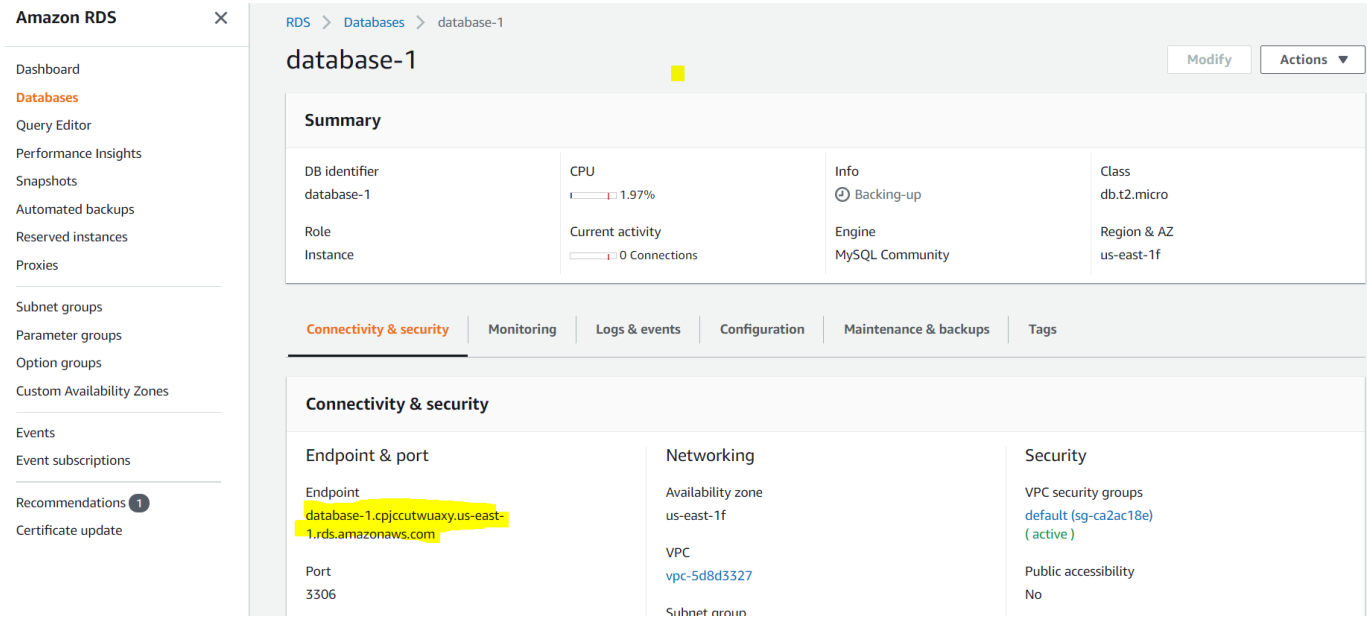
Connecting to an RDS

- Since we have selected public access as no , this RDS can be only accessed from within a VPC
- Create an EC2 machine . We will be using this machine to connect and operate RDS

- Before proceeding , whitelist the private ip address of this EC2 machine in the security group of the RDS for port 3306
- Login to EC2, and run below commands

```
#Install the mysql client on EC2
sudo yum install mysql -y
sudo yum install mysql-connector-java.noarch -y
```

- These commands will install the mysql client which helps us connect to any mysql database
- Now to connect to the RDS that we have created , navigate to the RDS console .
- Click on the RDS and copy the endpoint



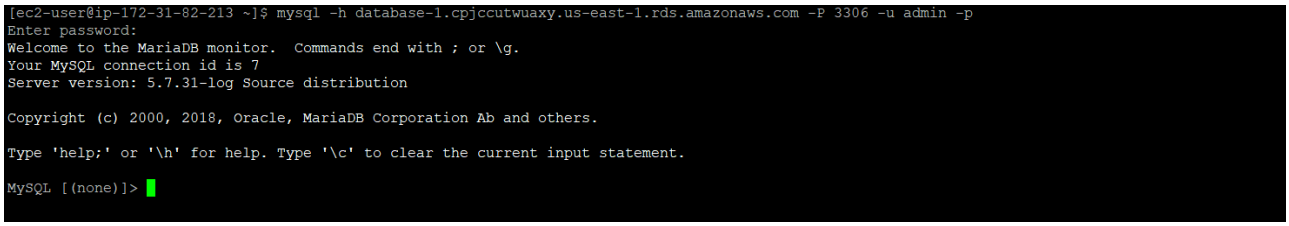
- Use below command from the EC2 terminal to connect to the server

```
mysql -h rds-endpoint-name -P 3306 -u username -p
```

replace the values in above command with your values for ex

```
mysql -h database-1.cpjccutwuaxy.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```

- After this command , there will be a password prompt . After entering , you'll be able to connect to the RDS



- Once we see that we are connected to mysql shell, we can start operating the database .
- Try out below commands to test out the DB

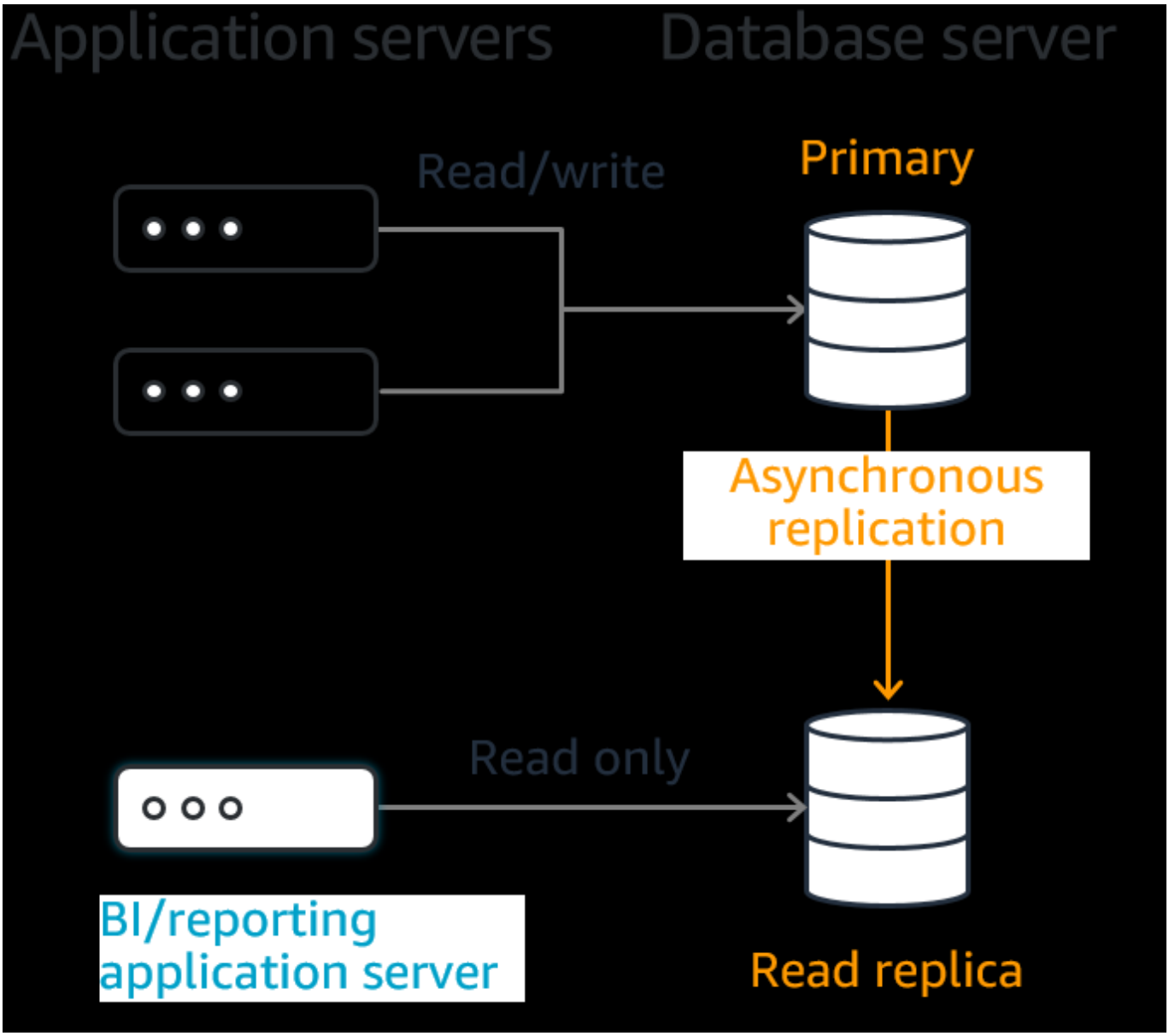
```
create database db1;
use db1;

CREATE TABLE customers
(id int(10),
 name varchar(50),
 city varchar(50),
 PRIMARY KEY (id )
);

insert into customers values(101,'rahul','amethi');
insert into customers values(102,'modi','india');



SELECT * from customers;
update customers set name='bob', city='london' where id=101;

SELECT * from customers;
delete from customers where id=101;

SELECT * from customers;
ALTER TABLE customers  ADD age varchar(50) ;

SELECT * from customers;
update customers set age=40 where id=102;

drop table customers;
```

Backup and restore RDS using snapshot

- With RDS we get feature of automated backups in the form of snapshots .
- These snapshots are similar to that of EBS
- In order to test out the principal , let us take a snapshot .
- Click on the database name in RDS console and click on create snapshot
- You will be able to see the progress of the snapshot in the snapshots tab of RDS .
- Once it becomes available ,go to databases and delete the database instance .
- Go to the snapshot that we just created . Select it and click on actions .
- Select Restore snapshot
- For db identifier you can keep the same name as previosu database instance . Change the instance to t2.micro. Keep rest of the things as it is . And click on restore DB instance
- Once it launches , connect to it and verify if the data is still present .

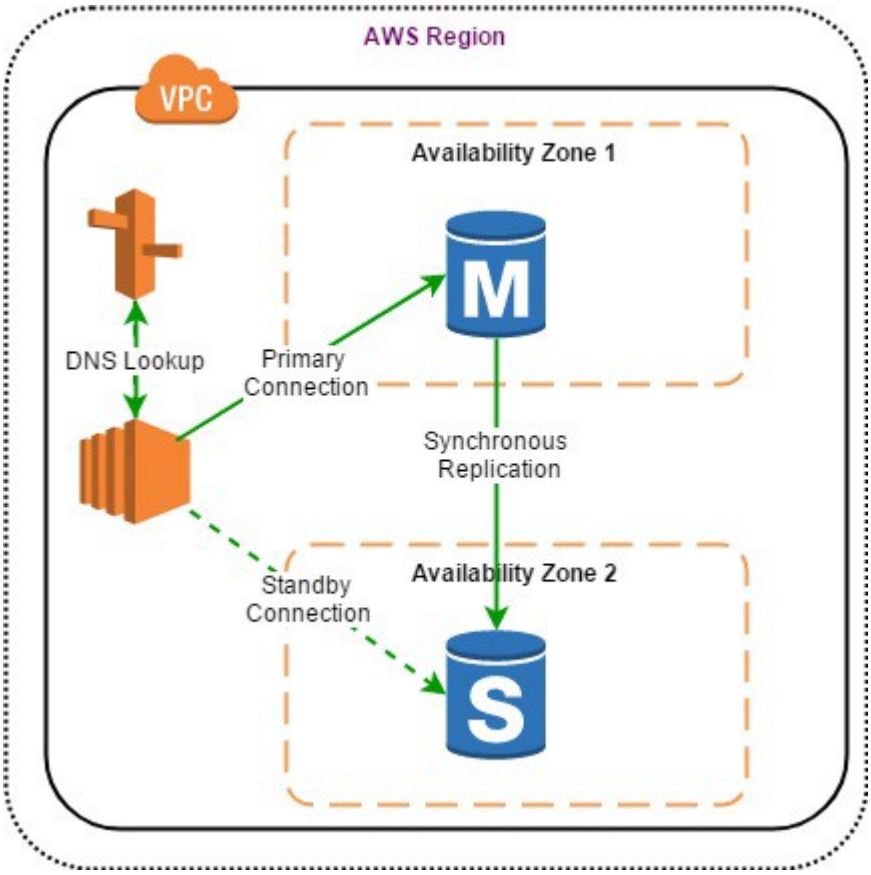
Read Replica

- Read replicas are a great way of segragating the traffic based on the usage . These are meant for all the select queries .
- We can create upto 5 read replicas per database.
- This is asynchronus replication
- Read replicas give a different endpoint to connect to it .
- Click on the database and select create read replica
- Follow the same steps which we used to create the database .
- Once the read replica is created , we cna use the same endpoint to conect to it from our EC2 .



Multi-AZ deployment

- As the name suggests multi-az deployment creates a snadby db instance in another AZ
- In a scenario of failure of primary db , the standby instance comes up
- The failover controls are automatic and require no manual intervention
- The endpoint will remain the same hence no downtime will be observed
- This is synchronus replication
- We can enable it while creating the database or can be modified once the db is created .



FAQ

- What if there is connection timed out while connecting to DB
    - Make sure the database and EC2 are in same VPC . And there are appropriate entries in the security group for connection