## Connect to database instance

psql -h testdatabase-gp-instance-1.cgwbzekfvqhi.us-east-1.rds.amazonaws.com -p 5432 -U greenpg postgres -> Aurora post

psql -h testdatabase-gp1.cgwbzekfvqhi.us-east-1.rds.amazonaws.com -p 5432 -U greenpg1 postgres ->Post standalone

## Steps to install postgres in Amazon Linux

https://installvirtual.com/install-postgresql-10-on-amazon-ec2/

Step 1. Install PostgreSQL 10

Firstly we need to add the pgdg repository in Amazon Linux

amazon-linux-extras install postgresql10 vim epel -y

Step 2. Install PostgreSQL 10

yum install -y postgresql-server postgresql-devel

Create a new Postgres database cluster

/usr/bin/postgresql-setup --initdb

\* Initializing database in '/var/lib/pgsql/data'

\* Initialized, logs are in /var/lib/pgsql/initdb\_postgresql.log

Step 4. Start the services of Postgres

Now execute the following command to start and enable the postgres services.

systemctl enable postgresql

systemctl start postgresql

How to install postgresql 9.6 on Mac

Step 5. Login into Database

PostgreSQL installation is done. If you want to use the Database log in Postgres account by typing following command

su - postgres

You can get into Postgres console by typing

psql

##

AWSTemplateFormatVersion: 2010-09-09

Description: A basic CloudFormation template for an RDS Aurora PostgreSQL cluster.

Parameters:

DatabaseInstanceType:

Default: db.r4.large

AllowedValues:

- db.r4.large

- db.r4.xlarge

- db.r4.2xlarge

- db.r4.4xlarge

- db.r4.8xlarge

- db.r4.16xlarge

Description: The instance type to use for the database.

Type: String

DatabasePassword:

AllowedPattern: "[a-zA-Z0-9]+"

ConstraintDescription: must contain only alphanumeric characters.

Description: The database admin account password.

MaxLength: '41'

MinLength: '8'

NoEcho: 'true'

Type: String

DatabaseUsername:

AllowedPattern: "[a-zA-Z0-9]+"

ConstraintDescription: must contain only alphanumeric characters.

Description: The database admin account user name.

MaxLength: '16'

MinLength: '1'

Type: String

DatabaseClusterParameterGroupName:

Type: String

Default: "default.aurora-postgresql9.6"

DatabaseBackupRetentionPeriod:

Type: String

Default: 35

Description: The database backup retention period in days.

DatabaseVPC:

Description: The VPC where your Instance will be placed

Type: List<AWS::EC2::VPC::Id>

DatabaseSubnets:

Description: The subnets to place database instances in.

Type: List<AWS::EC2::Subnet::Id>

DatabaseSecurityGroups:

Type: List<AWS::EC2::SecurityGroup::Id>

Description: Security groups to apply to the RDS cluster.

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: default.aurora-postgresql9.6

Parameters:

- DatabaseInstanceType

- DatabaseName

- DatabaseUsername

- DatabasePassword

- DatabaseVPC

- DatabaseSubnets

- DatabaseSecurityGroups

- DatabaseBackupRetentionPeriod

- DatabaseClusterParameterGroupName

ParameterLabels:

DatabaseInstanceType:

default: Database Instance Type

DatabaseClusterParameterGroupName:

default: DB cluster parameter group

DatabasePassword:

default: Database Password

DatabaseUsername:

default: Database Username

DatabaseBackupRetentionPeriod:

default: Database Backup Retention Period

DatabaseSubnets:

default: Database Subnets

DatabaseSecurityGroups:

default: Database Security Groups

Resources:

DatabaseSubnetGroup:

Type: AWS::RDS::DBSubnetGroup

Properties:

DBSubnetGroupDescription: CloudFormation managed DB subnet group.

SubnetIds:

Ref: DatabaseSubnets

DatabaseCluster:

Type: AWS::RDS::DBCluster

Properties:

Engine: aurora-postgresql

MasterUsername:

Ref: DatabaseUsername

MasterUserPassword:

Ref: DatabasePassword

BackupRetentionPeriod:

Ref: DatabaseBackupRetentionPeriod

DBClusterParameterGroupName:

Ref: DatabaseClusterParameterGroupName

PreferredBackupWindow: 02:00-03:00

PreferredMaintenanceWindow: mon:03:00-mon:04:00

DBSubnetGroupName:

Ref: DatabaseSubnetGroup

VpcSecurityGroupIds:

Ref: DatabaseSecurityGroups

DatabasePrimaryInstance:

Type: AWS::RDS::DBInstance

Properties:

Engine: aurora-postgresql

DBClusterIdentifier:

Ref: DatabaseCluster

DBInstanceClass:

Ref: DatabaseInstanceType

DBSubnetGroupName:

Ref: DatabaseSubnetGroup

DatabaseReplicaInstance:

Type: AWS::RDS::DBInstance

Properties:

Engine: aurora-postgresql

DBClusterIdentifier:

Ref: DatabaseCluster

DBInstanceClass:

Ref: DatabaseInstanceType

DBSubnetGroupName:

Ref: DatabaseSubnetGroup

DatabaseReplicaInstance2:

Type: AWS::RDS::DBInstance

Properties:

Engine: aurora-postgresql

DBClusterIdentifier:

Ref: DatabaseCluster

DBInstanceClass:

Ref: DatabaseInstanceType

DBSubnetGroupName:

Ref: DatabaseSubnetGroup

CREATE TABLE COMPANY(

ID INT PRIMARY KEY NOT NULL,

NAME TEXT NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR(50),

SALARY REAL

);

INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) VALUES (1, 'Paul', 32, 'California', 20000.00);

Created a postgresql database

Added 1 table and records

Took asnapshot

Migrated the sanp to the new Aurora-postgresql cluster

Problem-

The snap is getting migrated to a brand new Aurora cluster, no option for existing cluster.

Need to create a database in existing postgres

creating table inside the database

and then importing this in Aurora