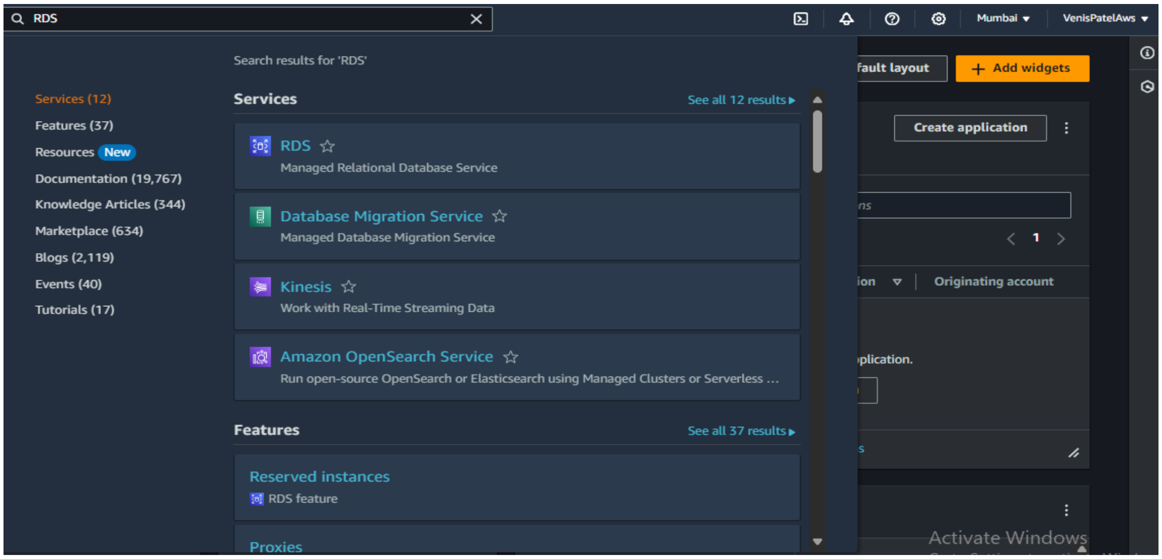
TASK 1: Create a new Amazon RDS instance with a database engine of your choice (PostgreSQL)

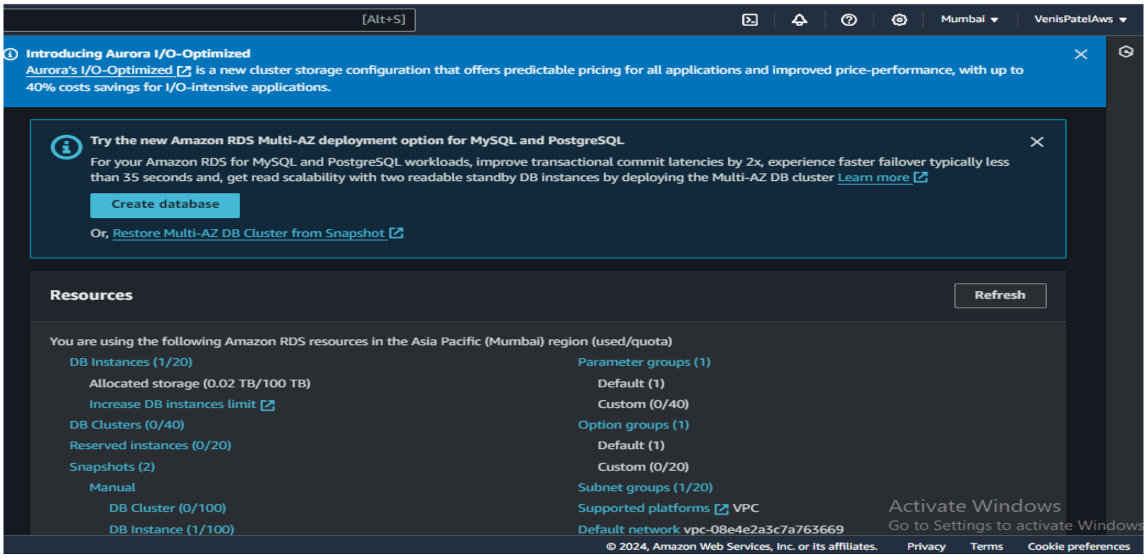
1. **Configure the instance with appropriate settings, including the master username and password.**
2. **Take a manual snapshot of your RDS instance.**
3. **Do PG Dump of RDS using connection string or Connect to the DB using connection string.**

- ➢ **Steps to create RDS PostgreSQL database:**

1. Search for RDS in AWS Console.



2. In RDS click on “Create Database”.



[Alt+S]

Mumbai ▾ VenisPatelAws ▾

Introducing Aurora I/O-Optimized

Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL

For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional commit latencies by 2x, experience faster failover typically less than 35 seconds and, get read scalability with two readable standby DB instances by deploying the Multi-AZ DB cluster Learn more

Create database

Or, Restore Multi-AZ DB Cluster from Snapshot

Resources Refresh

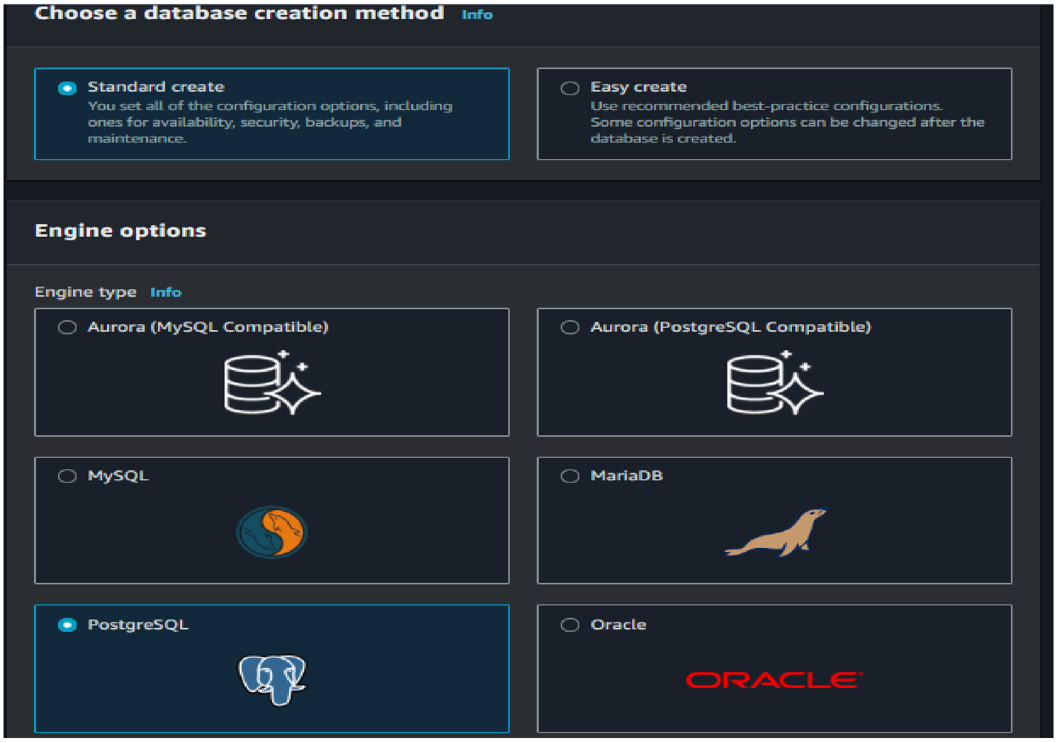
You are using the following Amazon RDS resources in the Asia Pacific (Mumbai) region (used/quota)

DB Instances (1/20)	Parameter groups (1)
Allocated storage (0.02 TB/100 TB)	Default (1)
Increase DB instances limit	Custom (0/40)
DB Clusters (0/40)	Option groups (1)
Reserved instances (0/20)	Default (1)
Snapshots (2)	Custom (0/20)
Manual	Subnet groups (1/20)
DB Cluster (0/100)	Supported platforms VPC
DB Instance (1/100)	Default network vpc-08e4e2a3c7a763669

Activate Windows
Go to Settings to activate Windows

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. Now select "Standard Create" in choose database creation, Select postgresql in Engine options. Select version PostgreSQL.



Choose a database creation method Info

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

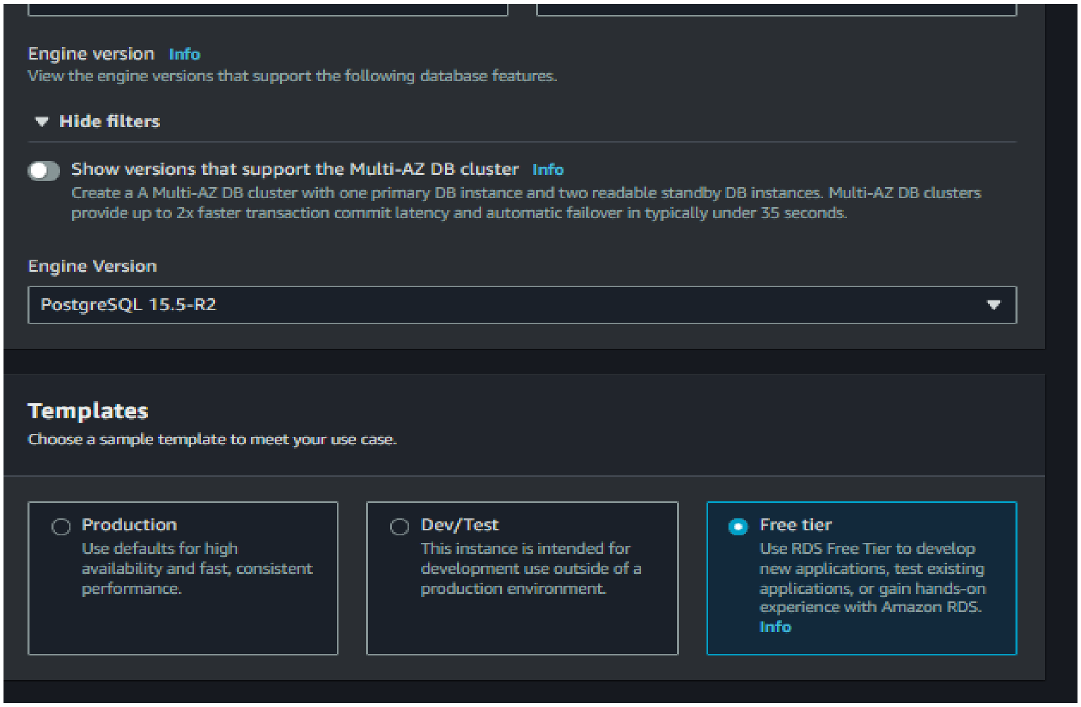
○ MySQL

○ MariaDB

● PostgreSQL

○ Oracle

4. Select Postgres version 15.5-R2 and select Free tier Templates.



Engine version Info

View the engine versions that support the following database features.

▼ Hide filters

○ Show versions that support the Multi-AZ DB cluster Info

Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Engine Version

PostgreSQL 15.5-R2 ▼

Templates

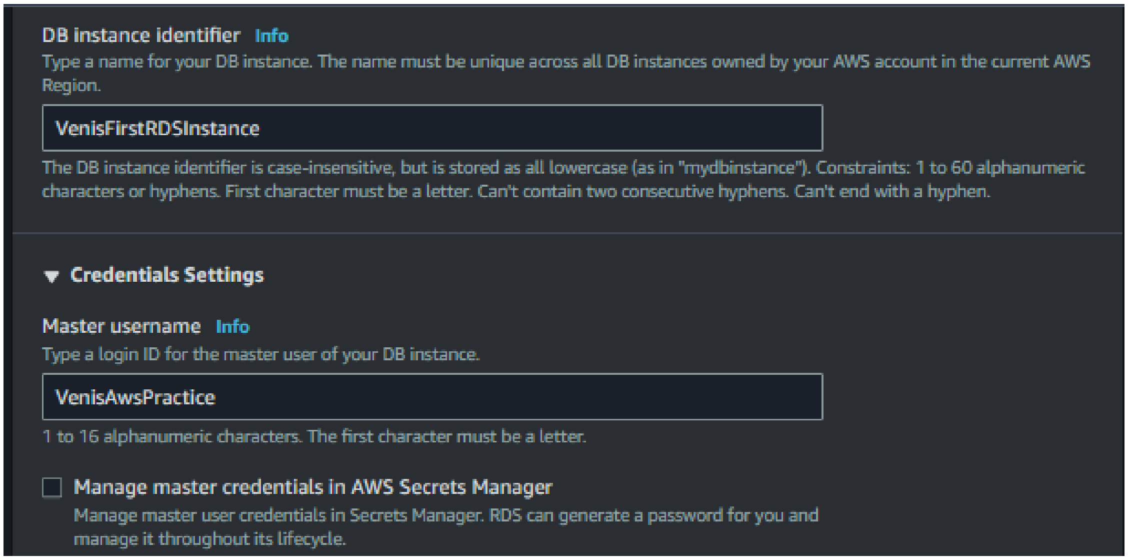
Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

5. Now enter the following details:
    a. Db Instance Identifier : VenisFirstRDSInstance
    b. Master Username: VenisAwsPractice
    c. Master Password



DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

VenisFirstRDSInstance

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username Info

Type a login ID for the master user of your DB instance.

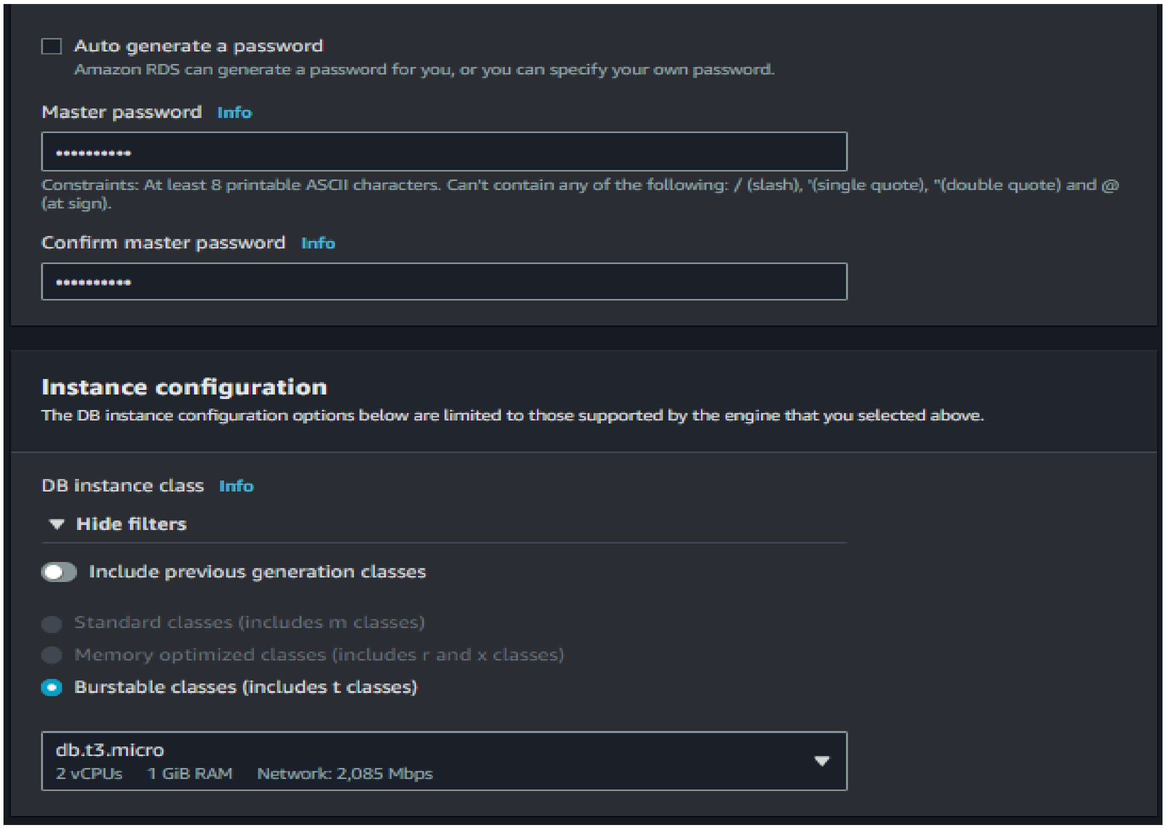
VenisAwsPractice

1 to 16 alphanumeric characters. The first character must be a letter.

□ Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

6. Select Instance configuration as db.t3.micro (covered under free tier).



□ Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

••••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

••••••••••

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class Info

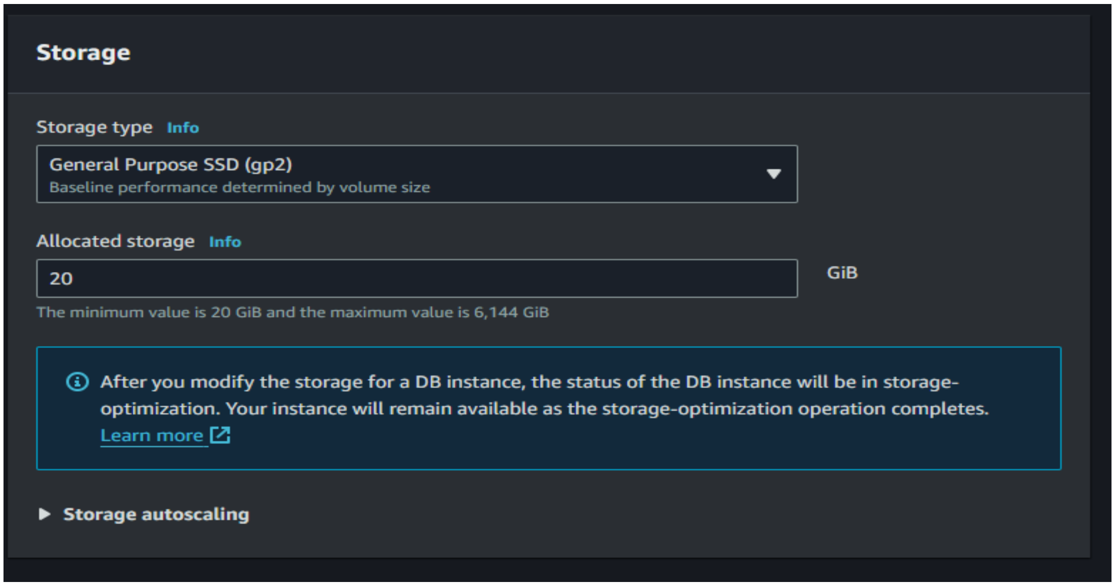
▼ Hide filters

Include previous generation classes

○ Standard classes (includes m classes)
○ Memory optimized classes (includes r and x classes)
● Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

7. Then select storage type "gp2" and allocate 20gb of storage.



Storage

Storage type Info

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage Info

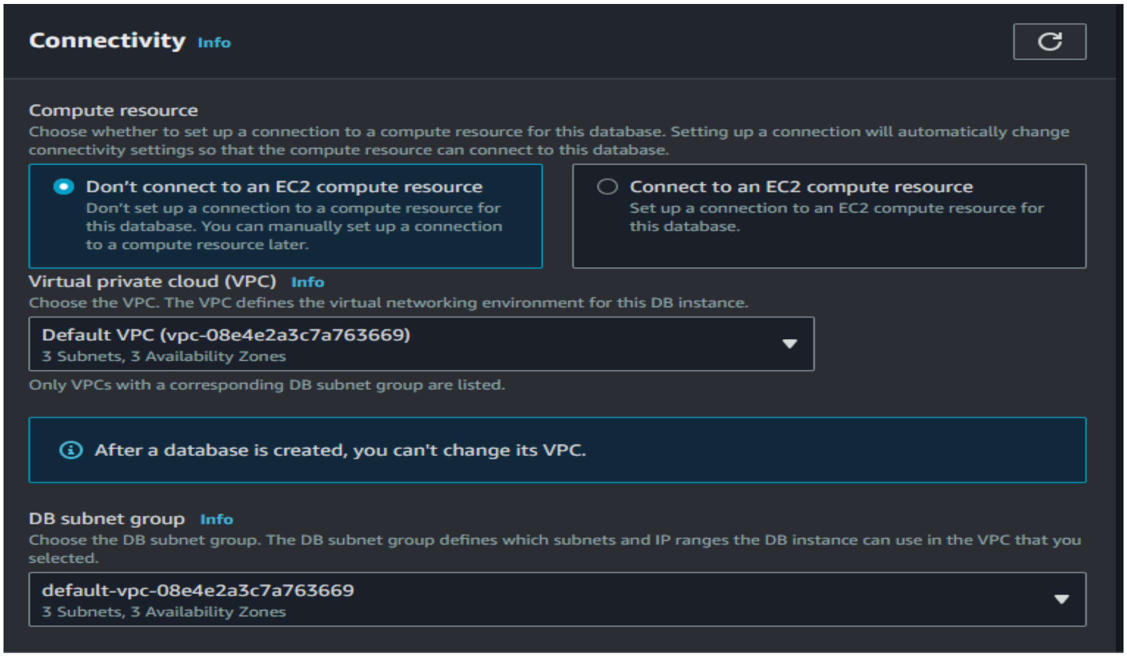
20 GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.
Learn more

▶ Storage autoscaling

8. Under connectivity select "Don’t Connect to an EC2 compute resource", Then select Default VPC. And select default-vpc in subnet group.



Connectivity Info

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

● Don’t connect to an EC2 compute resource
Don’t set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

○ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-08e4e2a3c7a763669)
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

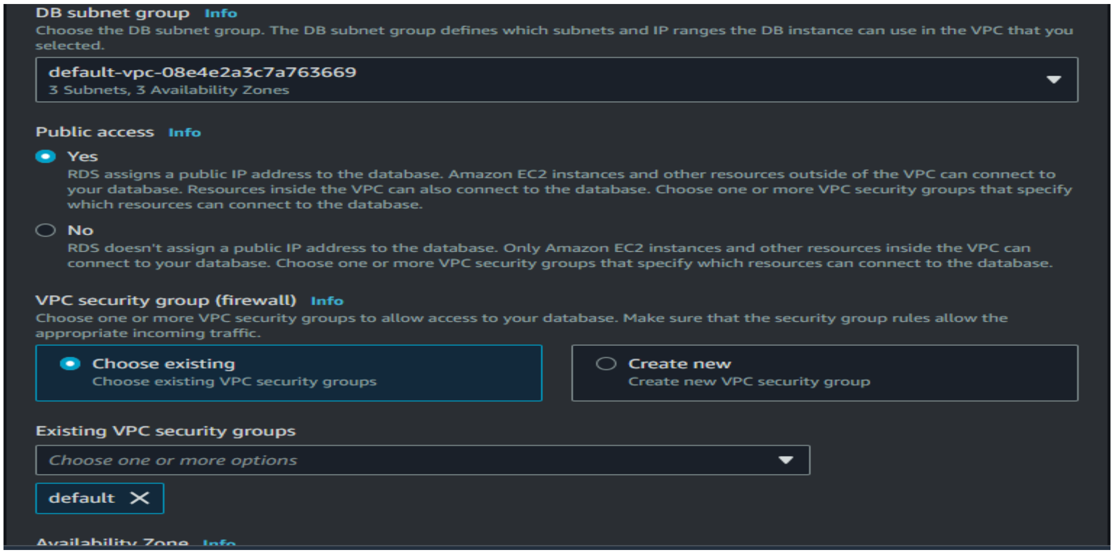
ⓘ After a database is created, you can't change its VPC.

DB subnet group Info

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-08e4e2a3c7a763669
3 Subnets, 3 Availability Zones

9. We don’t want to allow RDS database to be accessed but just for tutorial purpose, we are making it public . Select Existing “default” security groups.



DB subnet group Info

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-08e4e2a3c7a763669
3 Subnets, 3 Availability Zones

Public access Info

● Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

○ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

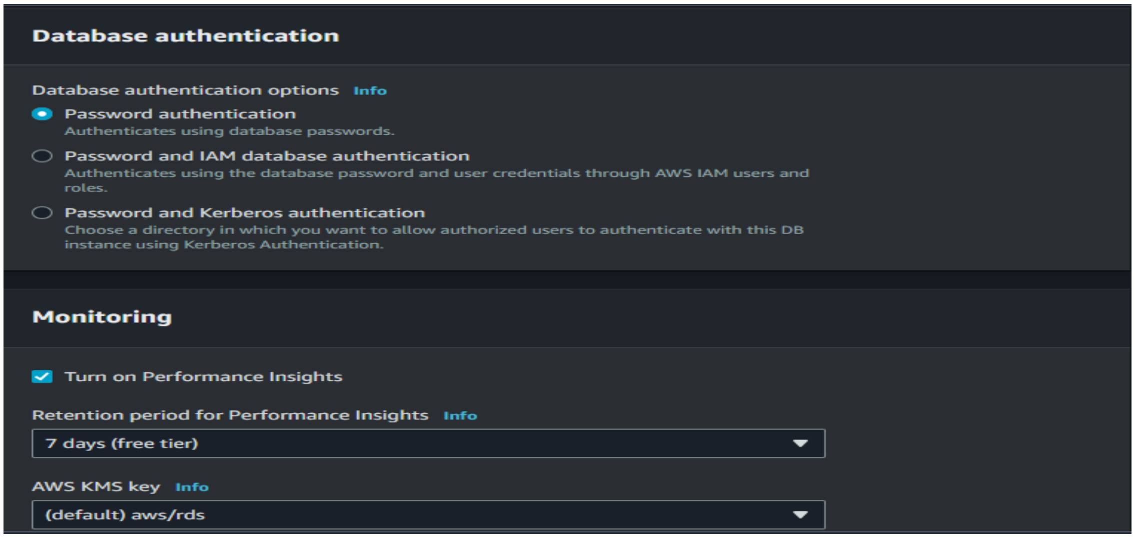
Existing VPC security groups

Choose one or more options

default ✕

Availability Zone Info

10. Select “Password Authentication” as Database authentication and keep everything same.



Database authentication

Database authentication options Info

● Password authentication
Authenticates using database passwords.

○ Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

○ Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

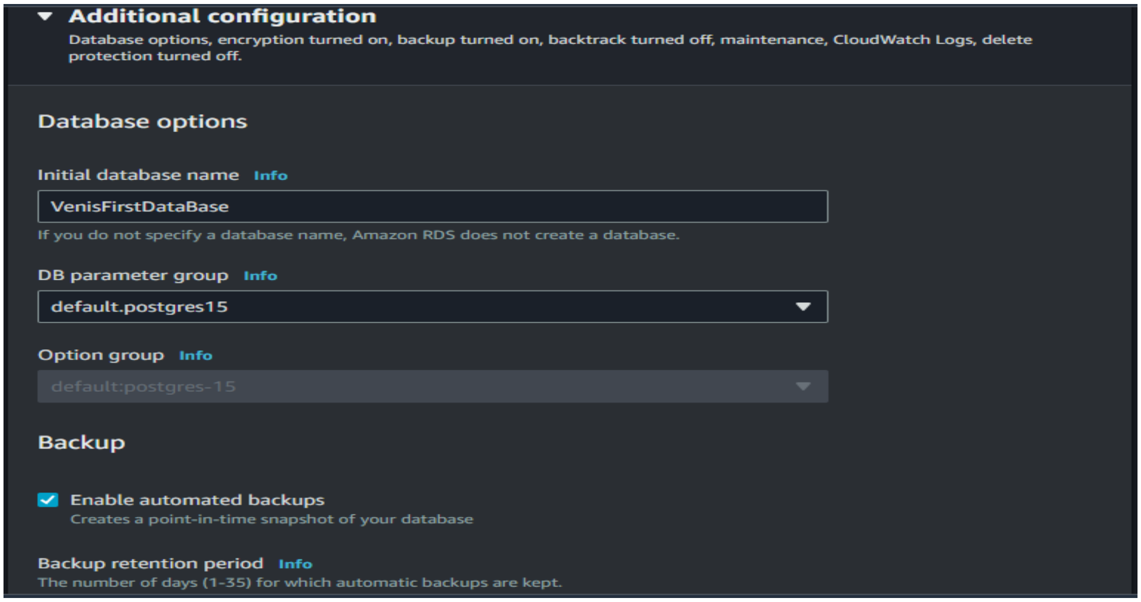
Monitoring

☑ Turn on Performance Insights

Retention period for Performance Insights Info
7 days (free tier)

AWS KMS key Info
(default) aws/rds

11. Now Go into Additional configuration and give Initial database name as “VenisFirstDataBase” and keep everything default.



Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name Info
VenisFirstDataBase
If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group Info
default.postgres15

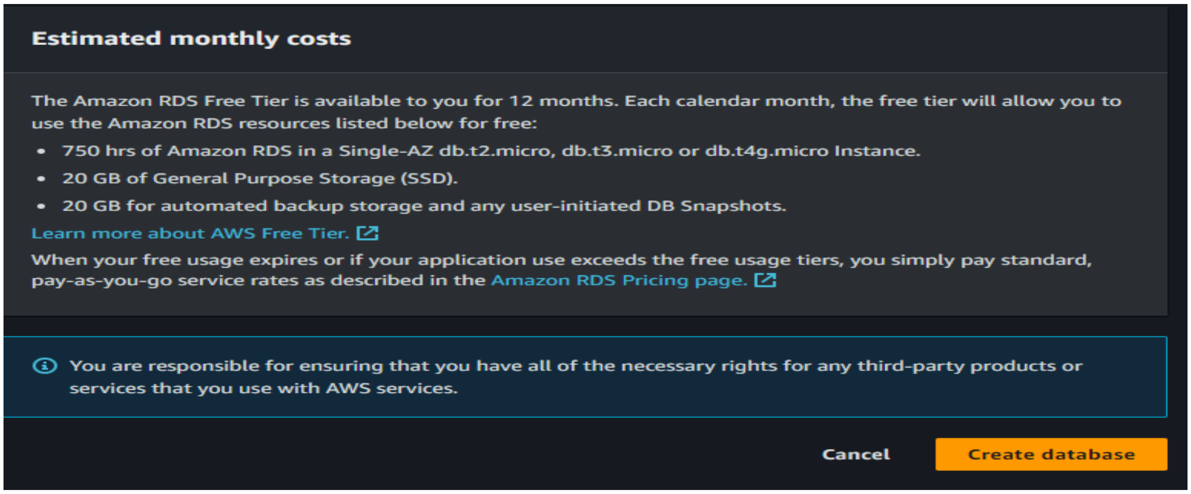
Option group Info
default:postgres-15

Backup

☑ Enable automated backups
Creates a point-in-time snapshot of your database

Backup retention period Info
The number of days (1-35) for which automatic backups are kept.

12. Now Select “Create database”.



Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

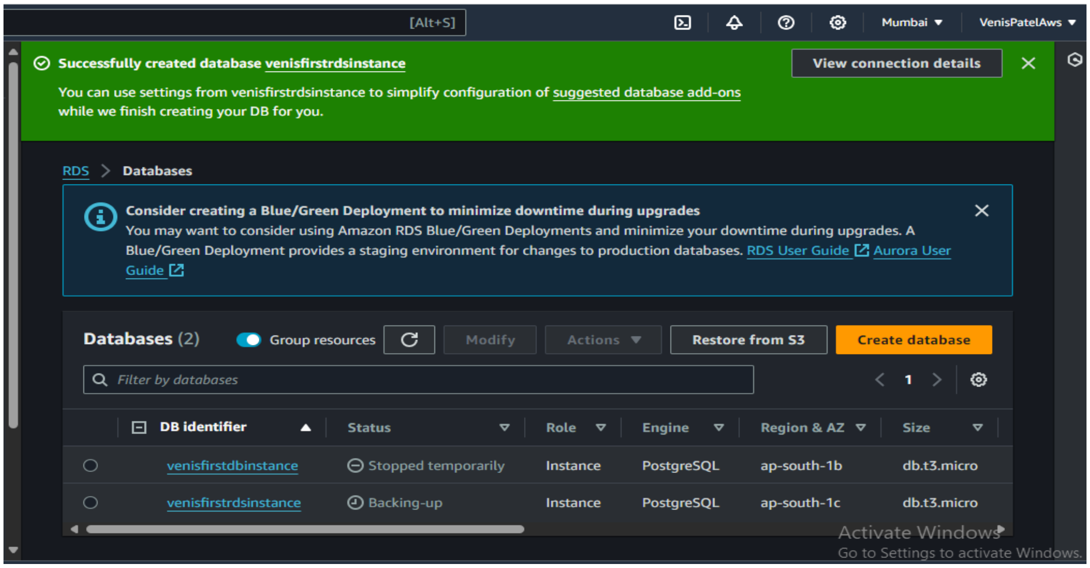
- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

Learn more about AWS Free Tier.

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page.

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

13. Now, here we can see that RDS has been created.



Successfully created database venisfirstrdsinstance

You can use settings from venisfirstrdsinstance to simplify configuration of suggested database add-ons while we finish creating your DB for you.

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

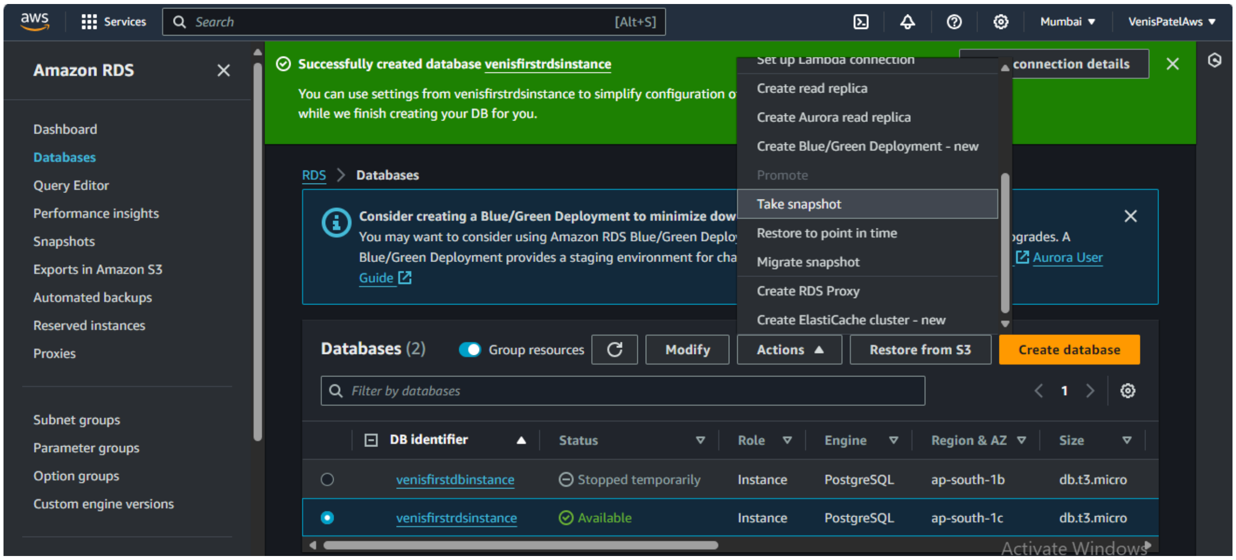
Databases (2)

	DB identifier	Status	Role	Engine	Region & AZ	Size
○	venisfirstdbinstance	Stopped temporarily	Instance	PostgreSQL	ap-south-1b	db.t3.micro
○	venisfirstrdsinstance	Backing-up	Instance	PostgreSQL	ap-south-1c	db.t3.micro

Activate Windows
Go to Settings to activate Windows.

➢ Steps For Taking Snapshots.

1. Now wait for Available status of DB Instance and then select that instance and go to Actions and Select “Take Snapshot”.



Successfully created database venisfirstrdsinstance

You can use settings from venisfirstrdsinstance to simplify configuration o while we finish creating your DB for you.

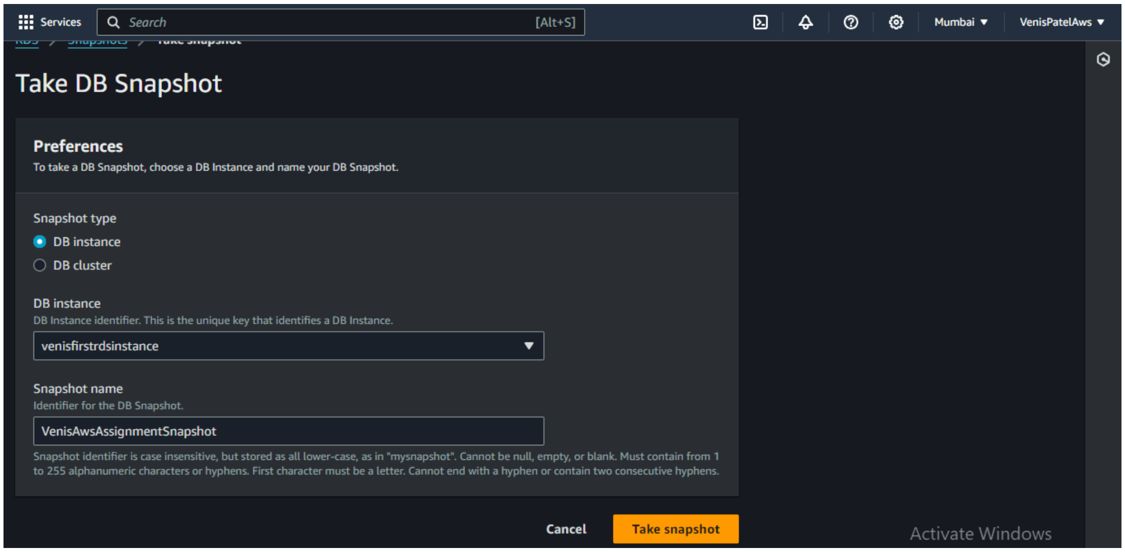
Consider creating a Blue/Green Deployment to minimize dow

You may want to consider using Amazon RDS Blue/Green Deplo ɔgrades. A Blue/Green Deployment provides a staging environment for cha Aurora User Guide

Databases (2)

	DB identifier ▲	Status	Role	Engine	Region & AZ	Size
○	venisfirstdbinstance	⊖ Stopped temporarily	Instance	PostgreSQL	ap-south-1b	db.t3.micro
●	venisfirstrdsinstance	✓ Available	Instance	PostgreSQL	ap-south-1c	db.t3.micro

2. Now select DB Instance and give DB instance “venisfirstinstance” and now give Snapshot name “VenisAwsAssignmentSnapshot”.



Take DB Snapshot

Preferences

To take a DB Snapshot, choose a DB Instance and name your DB Snapshot.

Snapshot type

● DB instance

○ DB cluster

DB instance

DB Instance identifier. This is the unique key that identifies a DB Instance.

venisfirstrdsinstance ▾

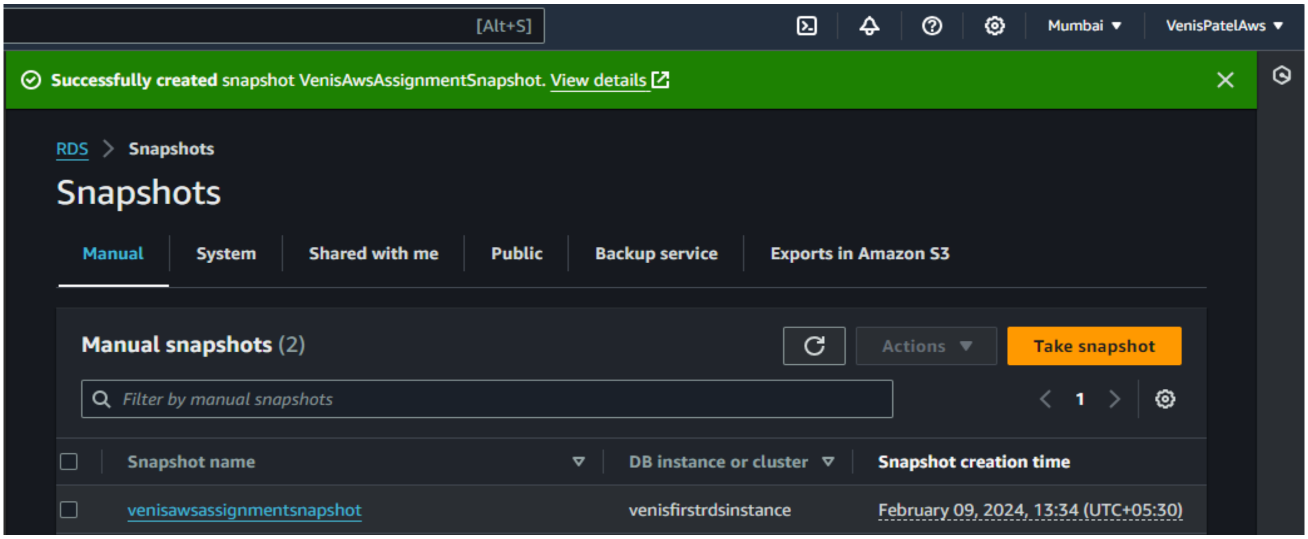
Snapshot name

Identifier for the DB Snapshot.

VenisAwsAssignmentSnapshot

Snapshot identifier is case insensitive, but stored as all lower-case, as in "mysnapshot". Cannot be null, empty, or blank. Must contain from 1 to 255 alphanumeric characters or hyphens. First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

3. Here we can see Snapshot has been taken.



Successfully created snapshot VenisAwsAssignmentSnapshot. View details

RDS > Snapshots

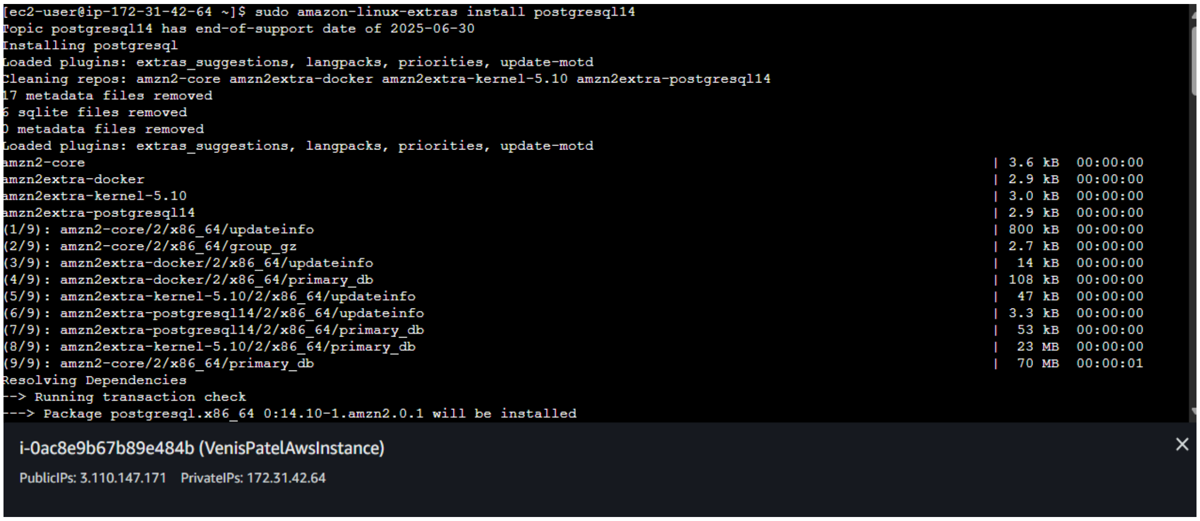
Snapshots

Manual snapshots (2)

□	Snapshot name	DB instance or cluster	Snapshot creation time
□	venisawsassignmentsnapshot	venisfirstrdsinstance	February 09, 2024, 13:34 (UTC+05:30)

➢ Steps For Taking backup using pg_dump

1. First of all connect a instance and Now we have to install Postgres so that we can access our RDS database.
   Run command : **sudo amazon-linux -extras install postgresql14**



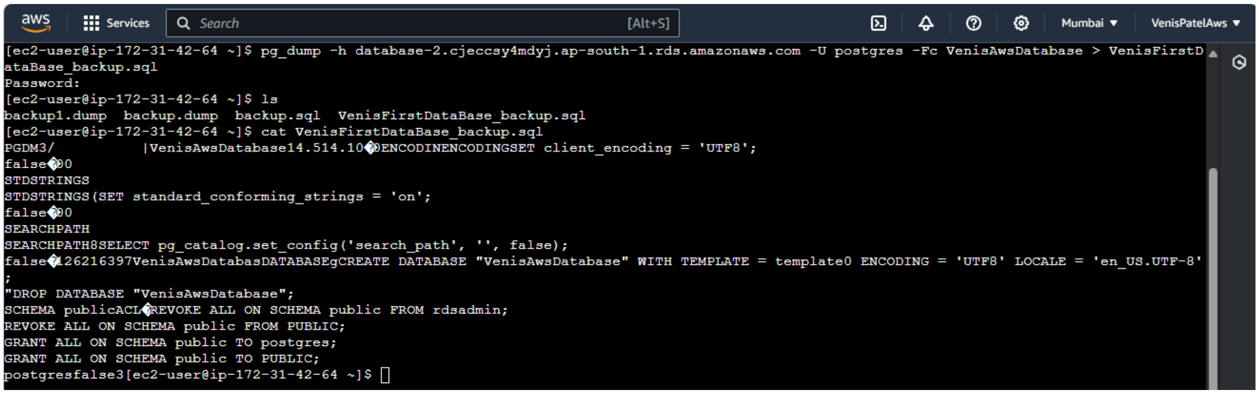
```
[ec2-user@ip-172-31-42-64 ~]$ sudo amazon-linux-extras install postgresql14
Topic postgresql14 has end-of-support date of 2025-06-30
Installing postgresql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-postgresql14
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                          | 3.6 kB  00:00:00
amzn2extra-docker                                                                   | 2.9 kB  00:00:00
amzn2extra-kernel-5.10                                                              | 3.0 kB  00:00:00
amzn2extra-postgresql14                                                             | 2.9 kB  00:00:00
(1/9): amzn2-core/2/x86_64/updateinfo                                               | 800 kB  00:00:00
(2/9): amzn2-core/2/x86_64/group_gz                                                 | 2.7 kB  00:00:00
(3/9): amzn2extra-docker/2/x86_64/updateinfo                                        |  14 kB  00:00:00
(4/9): amzn2extra-docker/2/x86_64/primary_db                                        | 108 kB  00:00:00
(5/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo                                   |  47 kB  00:00:00
(6/9): amzn2extra-postgresql14/2/x86_64/updateinfo                                  | 3.3 kB  00:00:00
(7/9): amzn2extra-postgresql14/2/x86_64/primary_db                                  |  53 kB  00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db                                   |  23 MB  00:00:00
(9/9): amzn2-core/2/x86_64/primary_db                                               |  70 MB  00:00:01
Resolving Dependencies
--> Running transaction check
---> Package postgresql.x86_64 0:14.10-1.amzn2.0.1 will be installed
```

i-0ac8e9b67b89e484b (VenisPatelAwsInstance)

PublicIPs: 3.110.147.171 PrivateIPs: 172.31.42.64

2. Now for taking backup of data , we have to run pg_dump command and after that we can see created backup file by using ls and cat commands .

pg_dump -h <hostname or Ip>-U <Mastername> -d <Initial Database name> -Fc -f <Output file>



```
[ec2-user@ip-172-31-42-64 ~]$ pg_dump -h database-2.cjeccsy4mdyj.ap-south-1.rds.amazonaws.com -U postgres -Fc VenisAwsDatabase > VenisFirstD
ataBase_backup.sql
Password:
[ec2-user@ip-172-31-42-64 ~]$ ls
backup1.dump  backup.dump  backup.sql  VenisFirstDataBase_backup.sql
[ec2-user@ip-172-31-42-64 ~]$ cat VenisFirstDataBase_backup.sql
PGDM3/          |VenisAwsDatabase14.514.10�0ENCODINENCODINGSET client_encoding = 'UTF8';
false�00
STDSTRINGS
STDSTRINGS(SET standard_conforming_strings = 'on';
false�00
SEARCHPATH
SEARCHPATH8SELECT pg_catalog.set_config('search_path', '', false);
false�126216397VenisAwsDatabasDATABASEgCREATE DATABASE "VenisAwsDatabase" WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE = 'en_US.UTF-8'
;
"DROP DATABASE "VenisAwsDatabase";
SCHEMA publicACL�REVOKE ALL ON SCHEMA public FROM rdsadmin;
REVOKE ALL ON SCHEMA public FROM PUBLIC;
GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO PUBLIC;
postgresfalse3[ec2-user@ip-172-31-42-64 ~]$ 
```