Case Study on

Amazon RDS



**Introduction:**

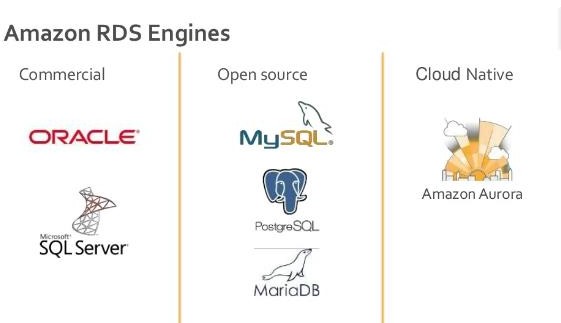
* Amazon Relational Database Service (Amazon RDS) is a service used to build and operate relational databases in the AWS Cloud. Amazon RDS is a managed database service
* Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks
* Amazon RDS provides the following specific advantages over database deployments that aren't fully managed:

1. You can use the database products you are already familiar with: MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server.
2. Amazon RDS manages backups, software patching, automatic failure

detection, and recovery.

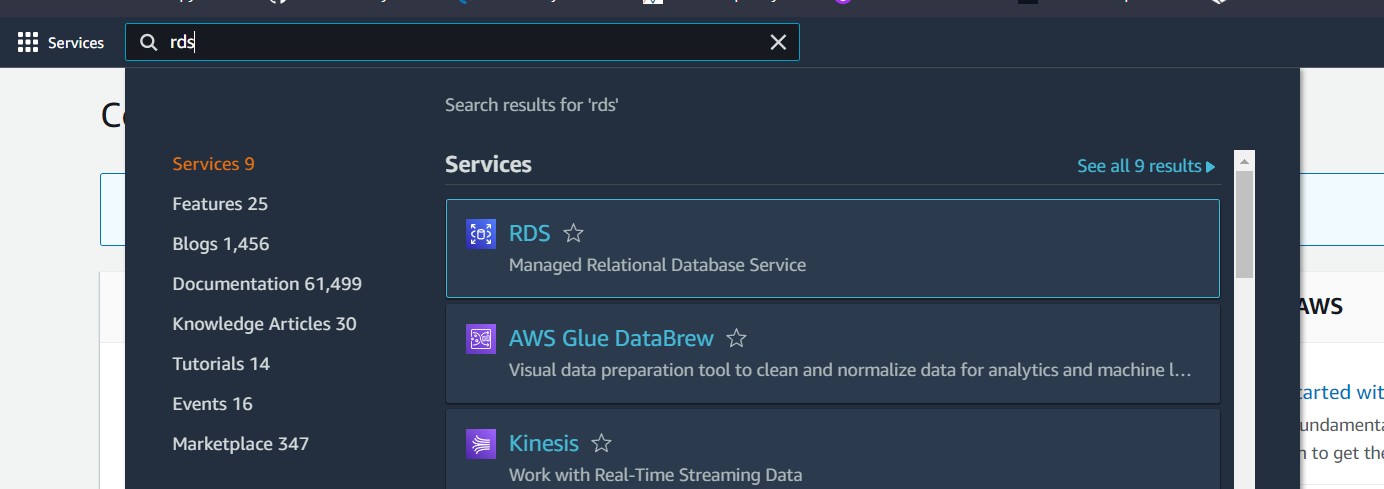
1. You can turn on automated backups, or manually create your own backup snapshots. You can use these backups to restore a database. The Amazon RDS restore process works reliably and efficiently.
2. You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use read replicas to increase read scaling.
3. In addition to the security in your database package, you can help control who can access your RDS databases by using AWS Identity and Access Management (IAM) to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud (VPC).

Amazon RDS supported engines

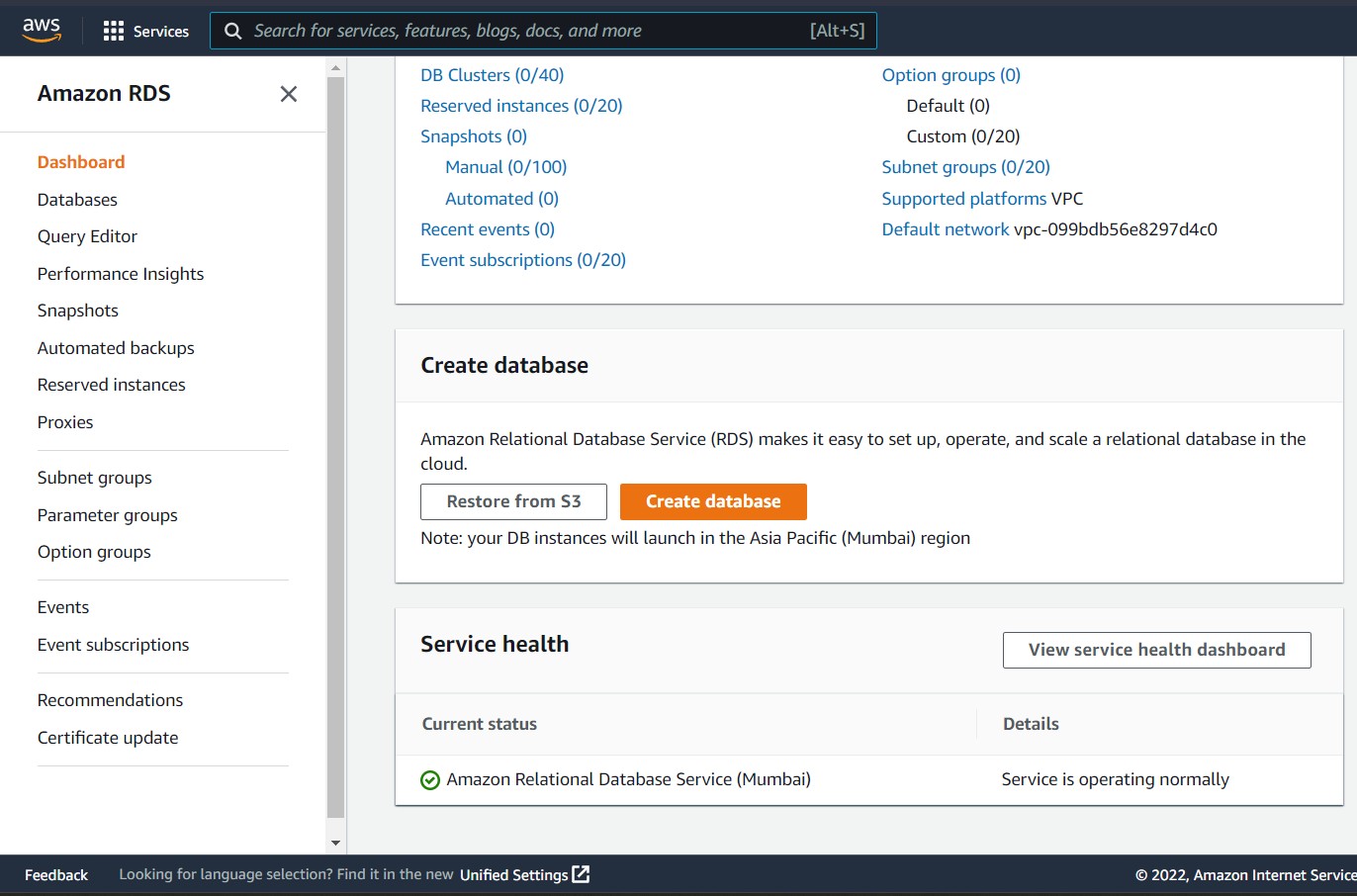


**Hands on**

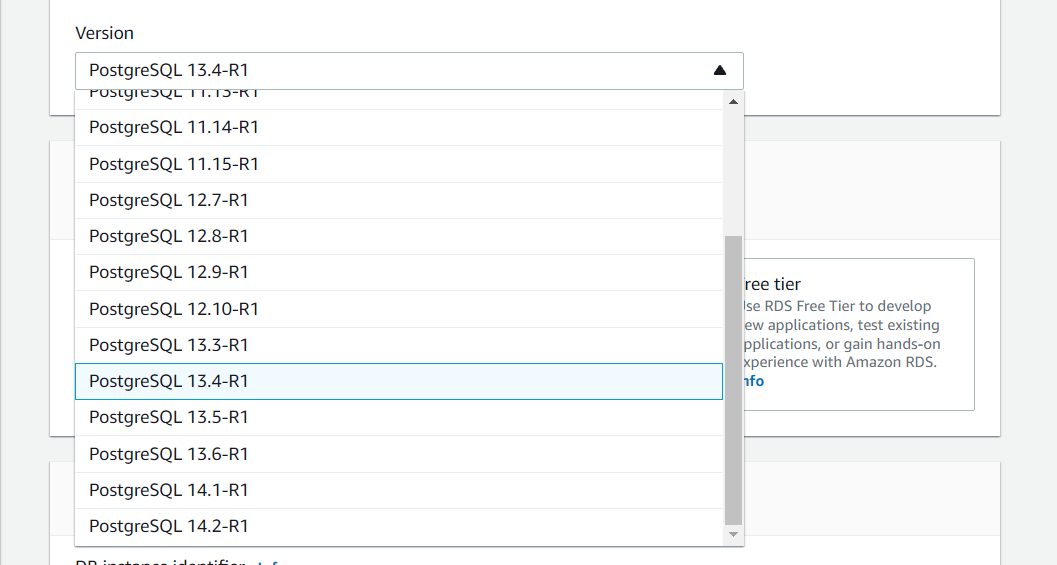
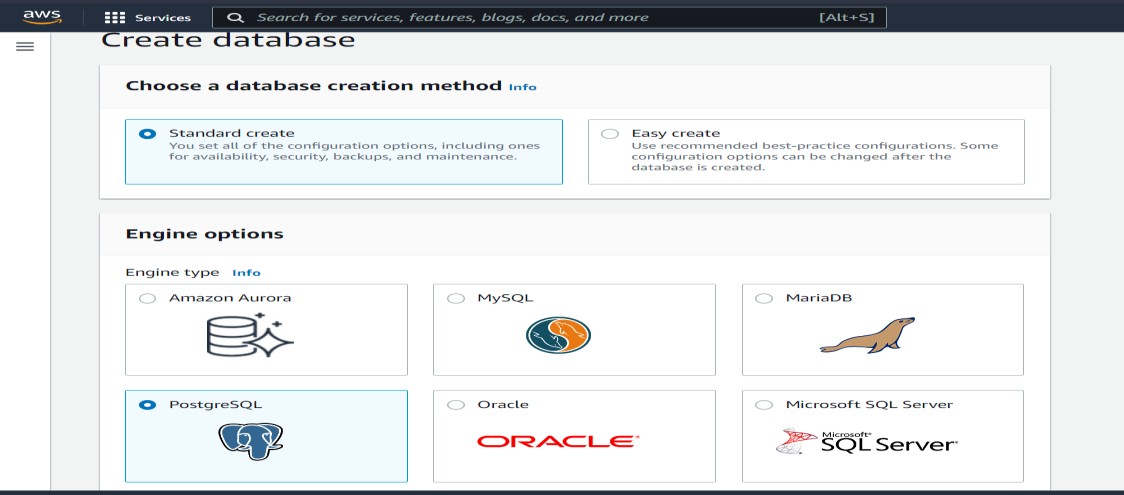
* + **Step 1**: Type RDS on management console search bar.



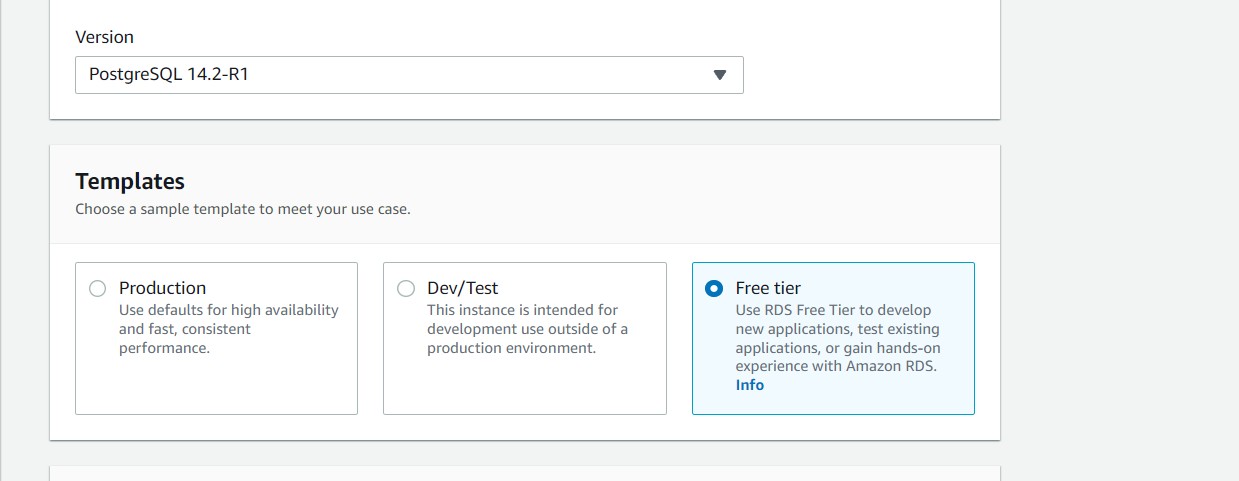
* + **Step 2**: Below interference will open. Now in order to create database click on Create database.



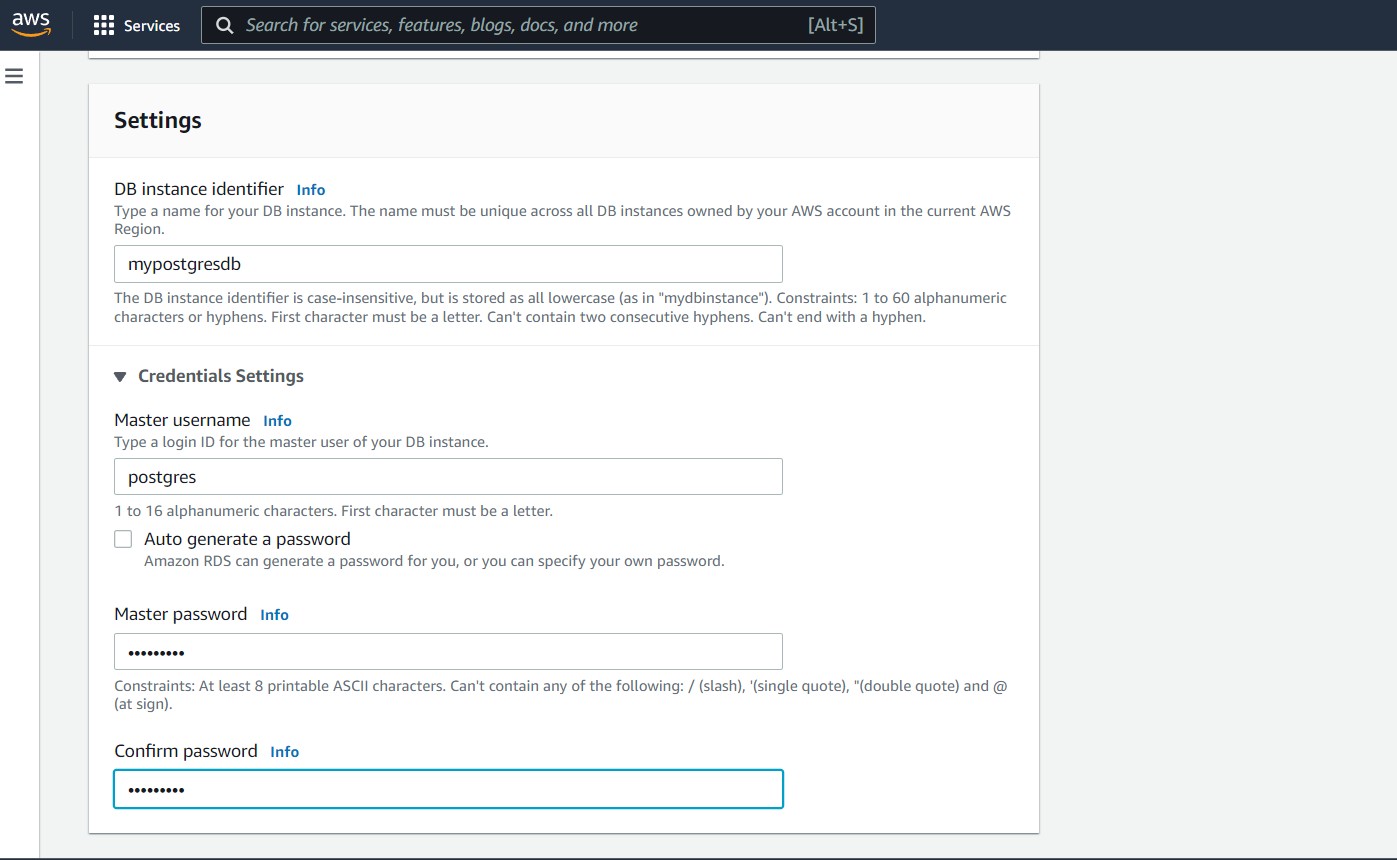
* + **Step 3**: choose standard create mode and select engine type which you want. for my system I have choose postgresql and its version



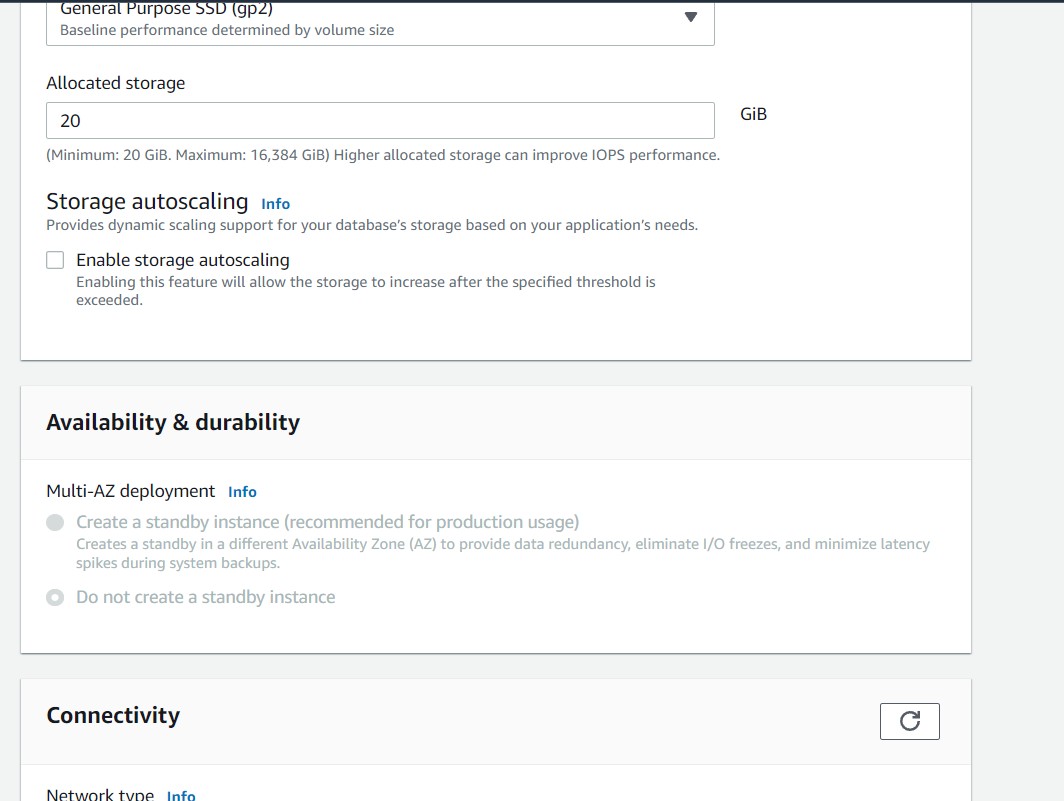
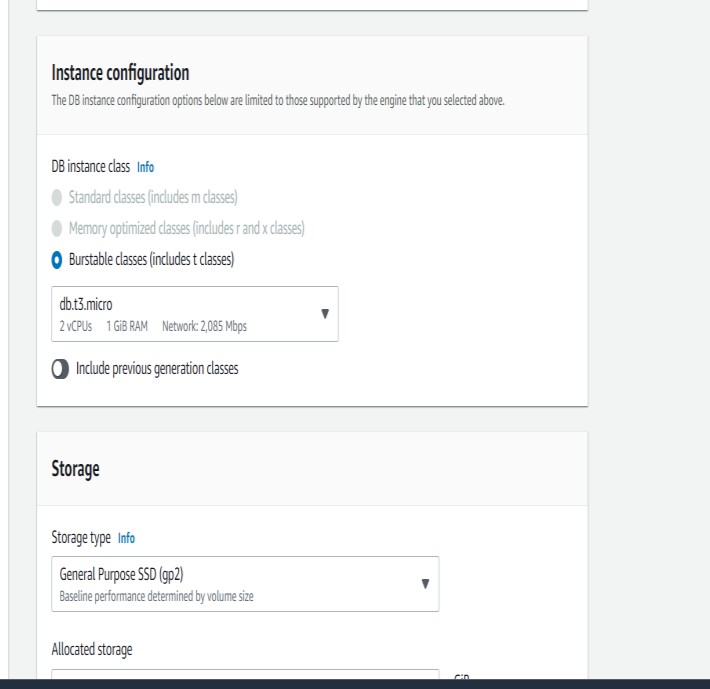
* **Step 4**: Now for demo purpose I am choosing free tier as it wont be costing money for using RDS.

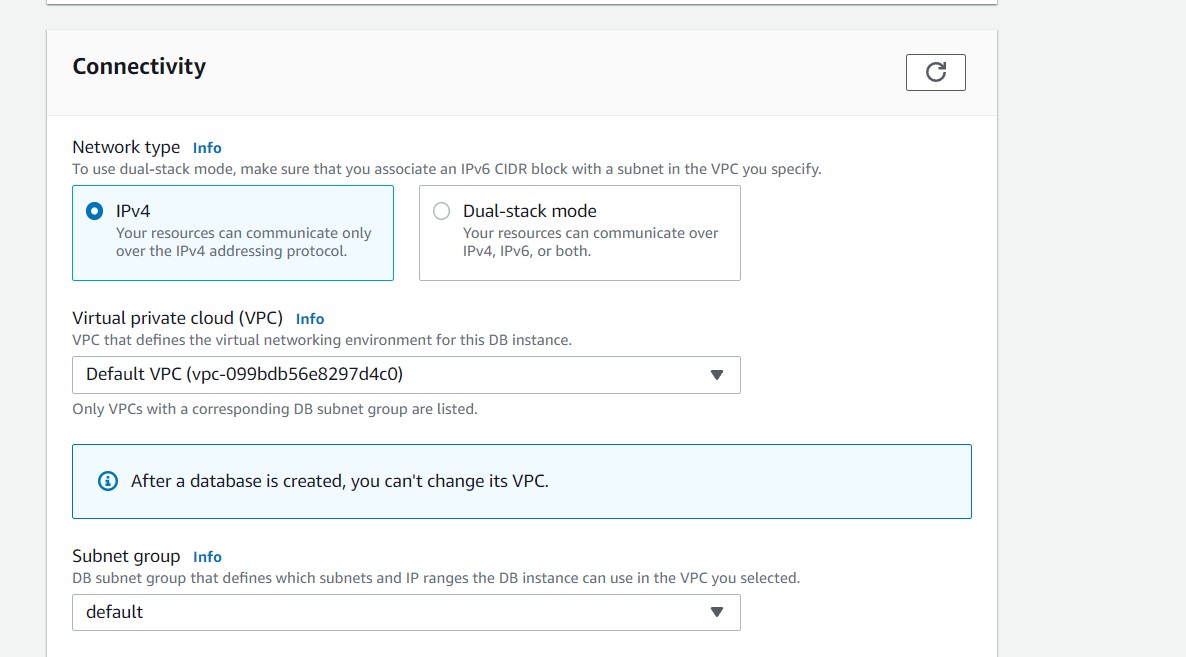
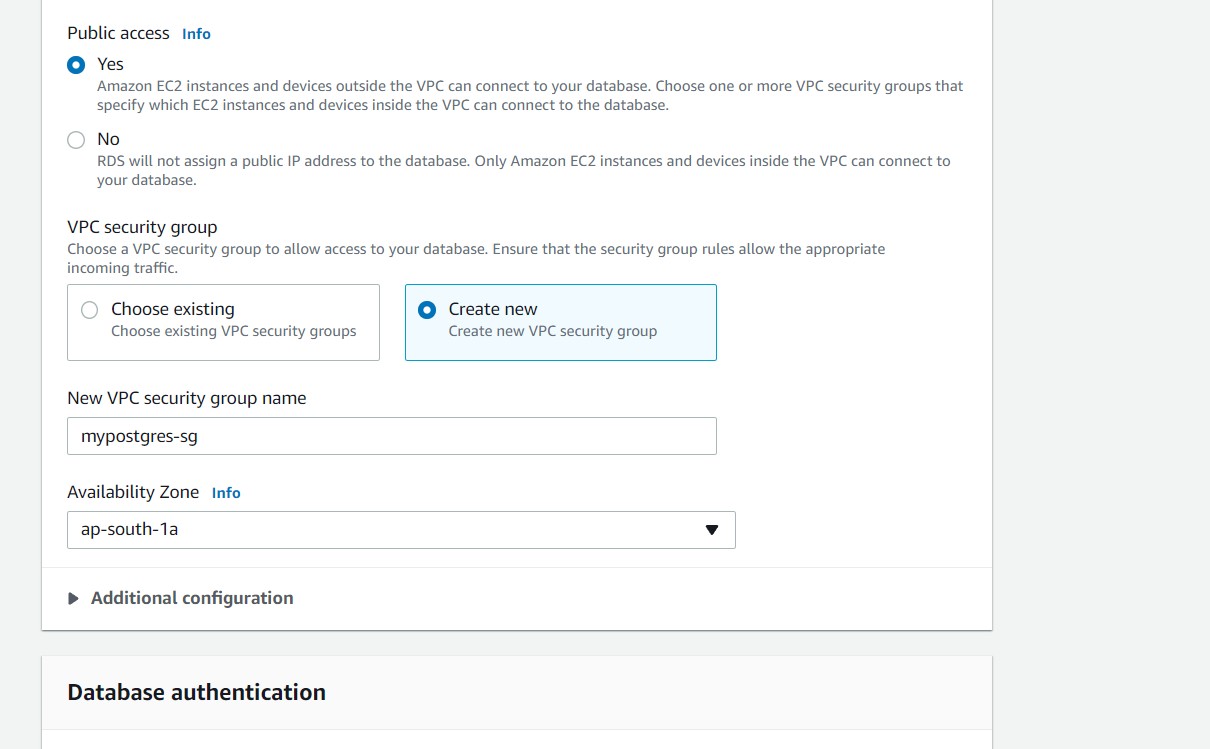


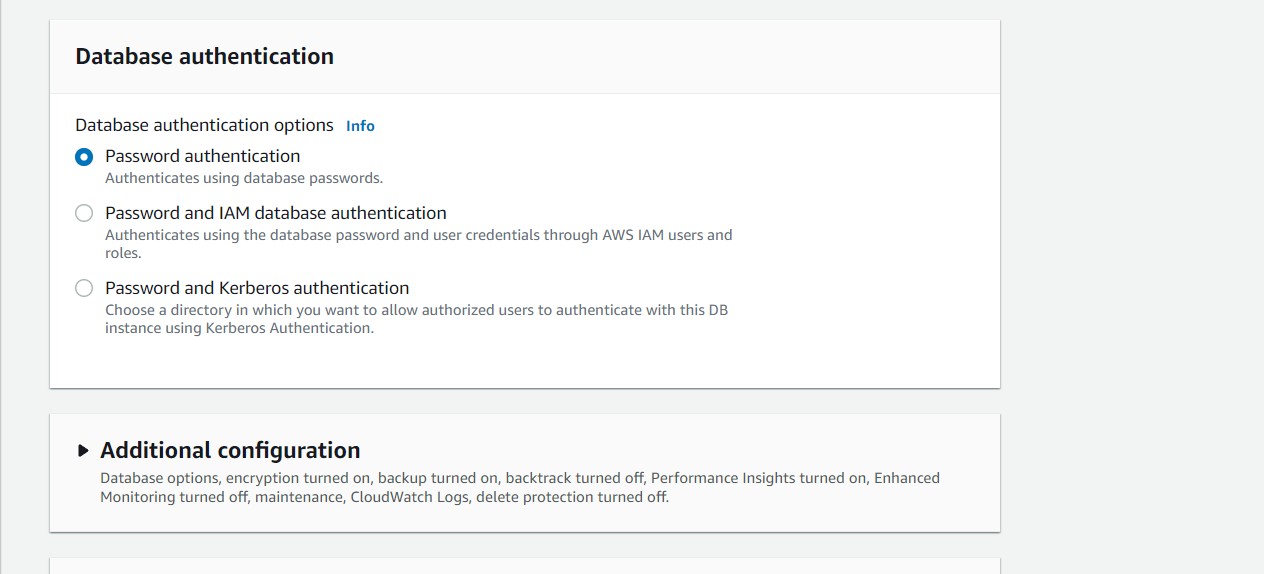
* **Step 5**: give the name of instance db classifier as per your choice and in credential give username and password.



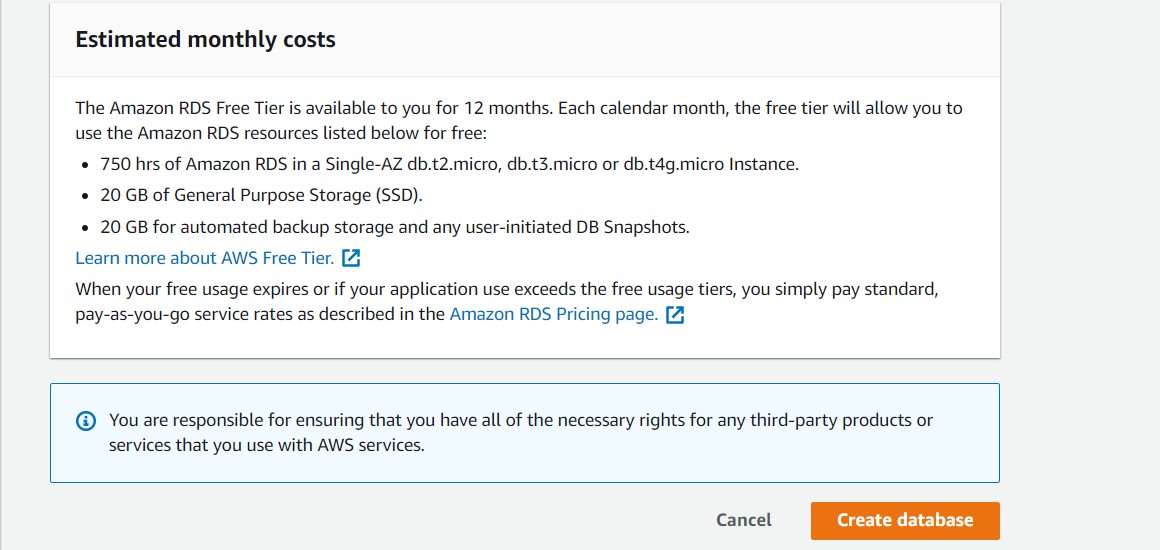
* **Step 6**: choose the instance class and storage as selected



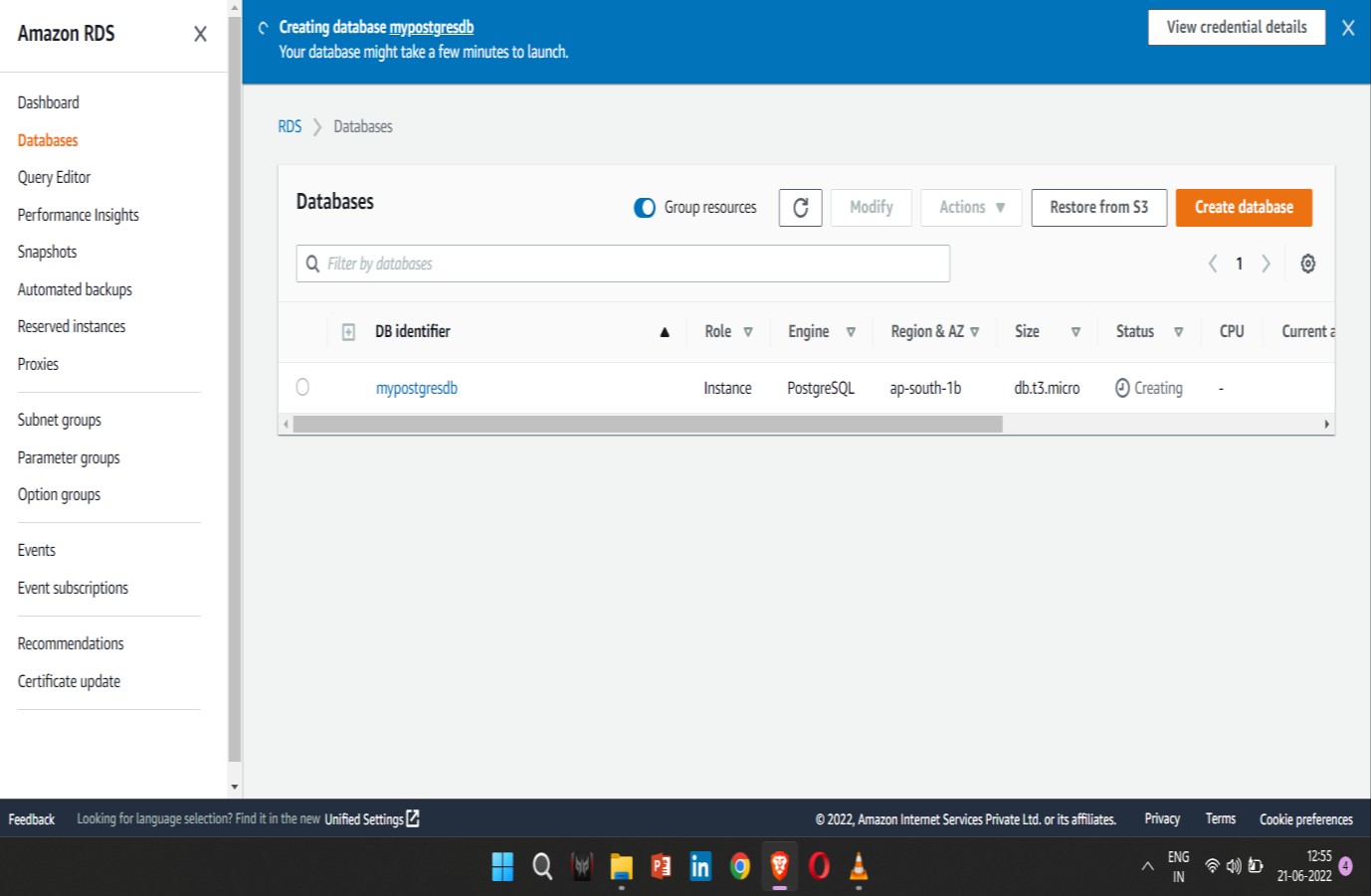
* **Step 7**: In connectivity choose network type as below you can choose default VPC and subnet as well as you can create your custom VPC
* **Step 8**: Give Public access to Yes and you can existing security group or can create new security group
  + **Step 9**: choose password authentication for database authentication and keep additional configuration as default

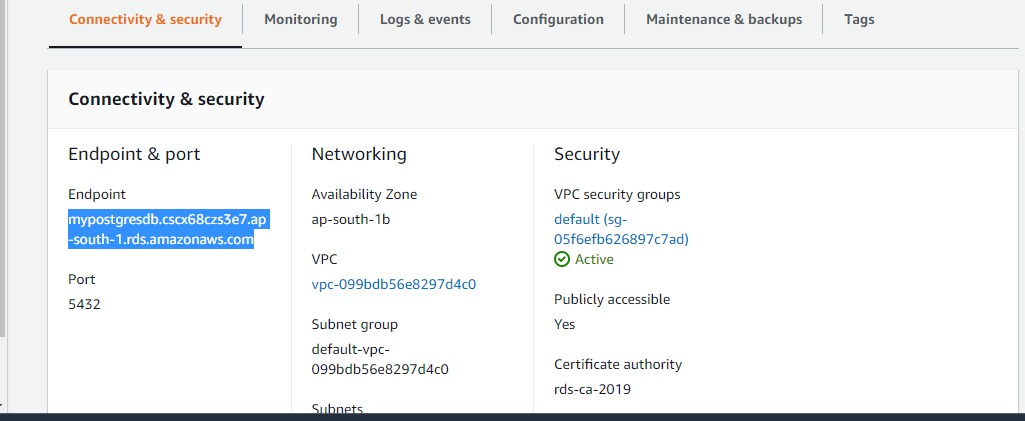


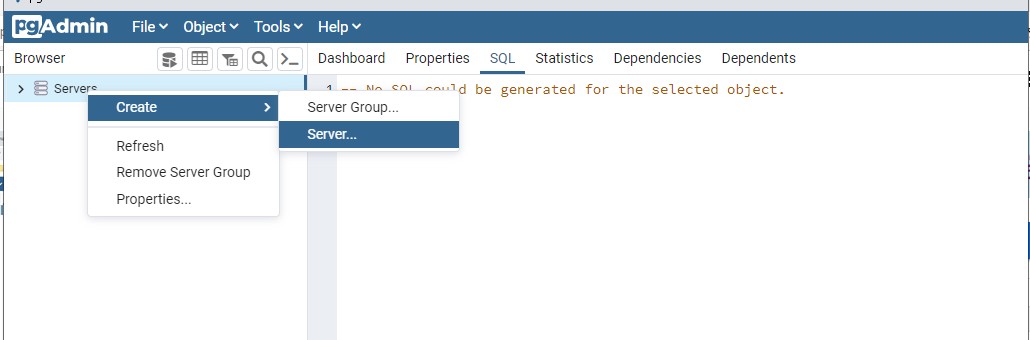
* + - **Step 10**: now click on create database



* + - **Step 11**: It will take 5 to 10 min to create your DB instance and once it is created you can connect your DB instance with your local SQL engine with the endpoint username and password



* + - **Step 12**: After clicking on your DB Identifier, you will get the endpoint . Just copy endpoint.
  + **Step 13**: Now Open Your PostgreSQL and right click on server then create and than again server as shown.



* + **Step 14**: Now enter the endpoint on host name and enter your username and password to establish connection . Save it and your connection will be established and you can create your database and start querying.

