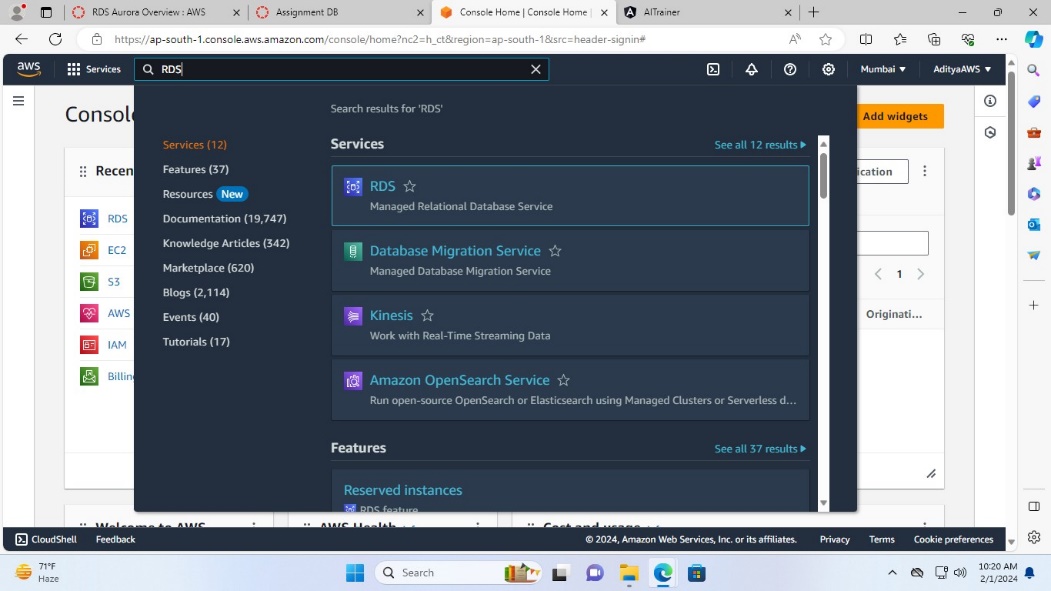
**Task 1. Create a new Amazon RDS instance with a database engine of your choice (PostgreSQL):**

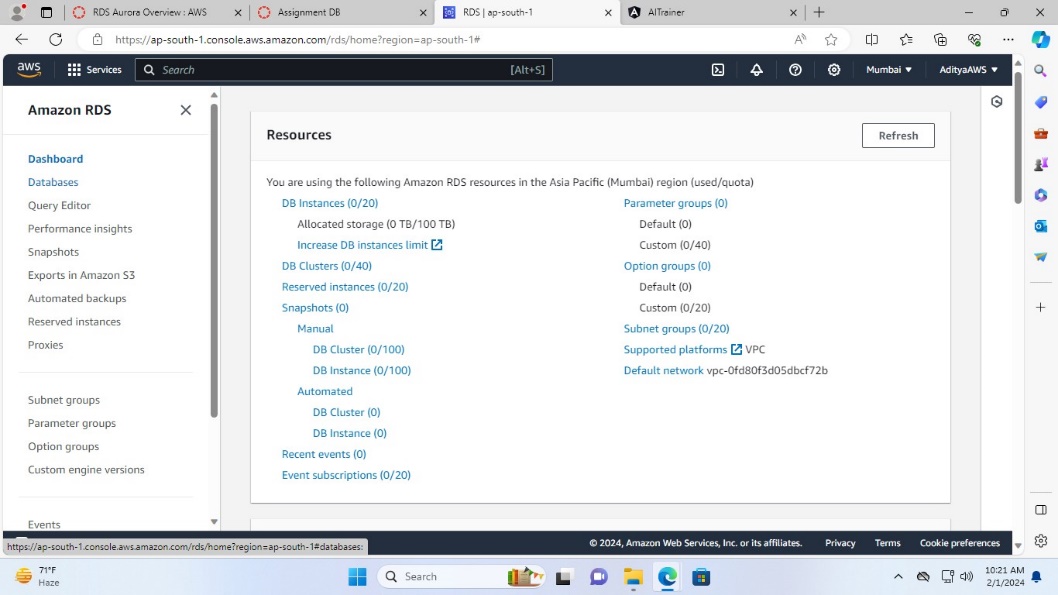
* 1. **Configure the instance with appropriate settings, including the master username and password.**
  2. **Take a manual snapshot of your RDS instance.**
  3. **Do PG Dump of RDS using connection string or Connect to the DB using connection string.**

**Steps to create Amazon RDS instance with database engine PostgreSQL:**

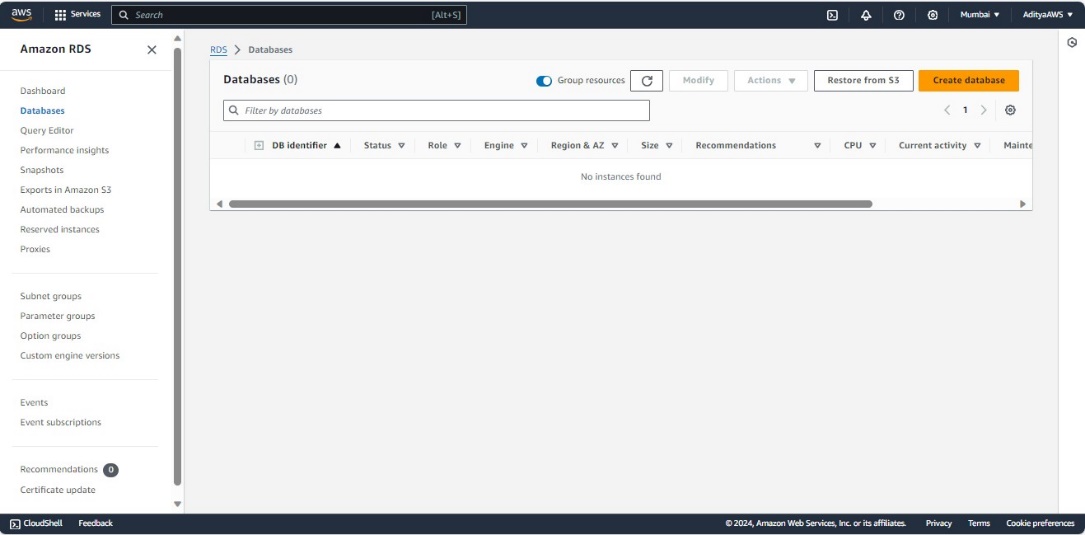
1. In the console home search RDS and navigate to the RDS.



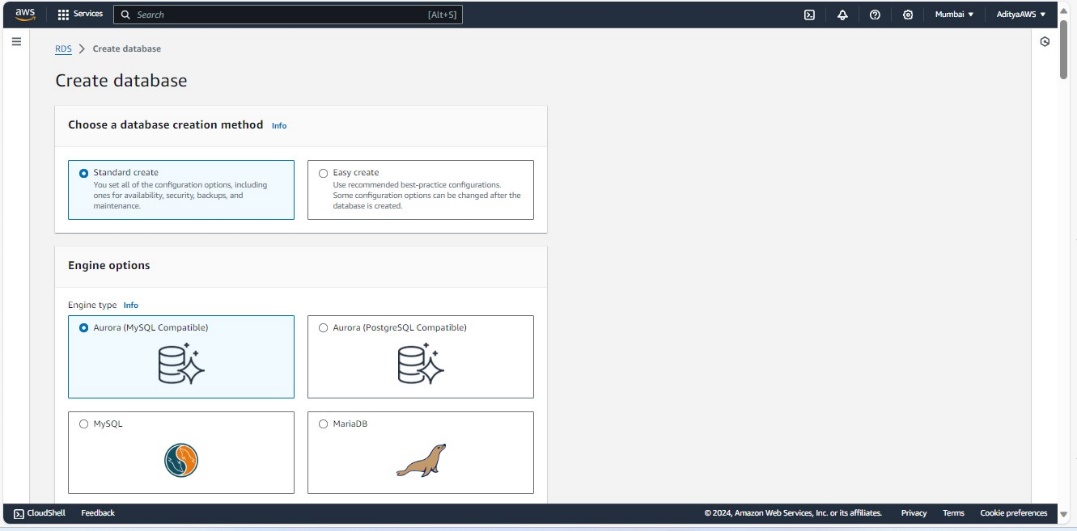
1. Now in the RDS console click on “Databases” in the dashboard menu or click on the “DB instances” in the resources section.



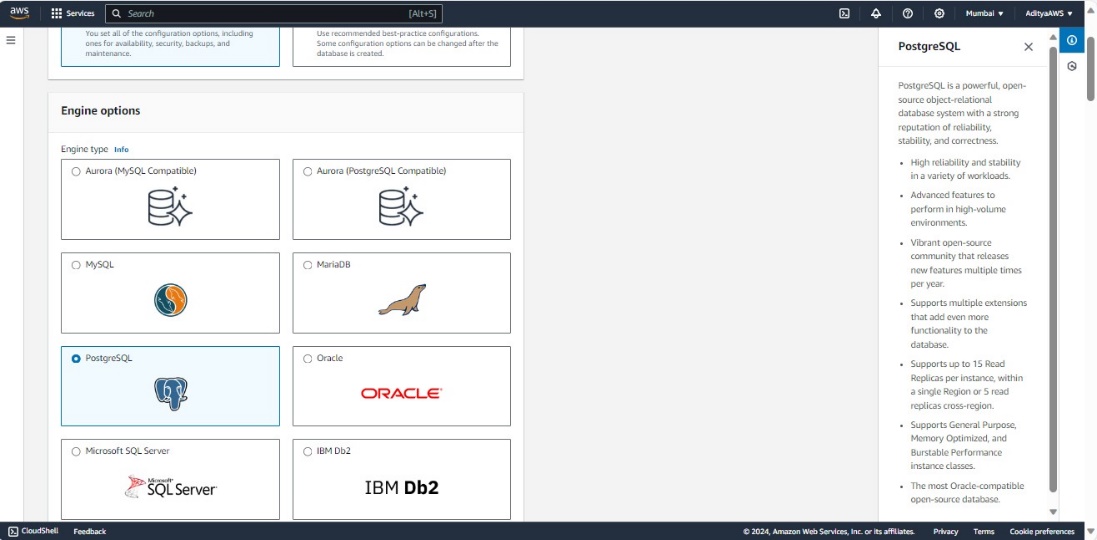
1. Now, click on “Create database”.



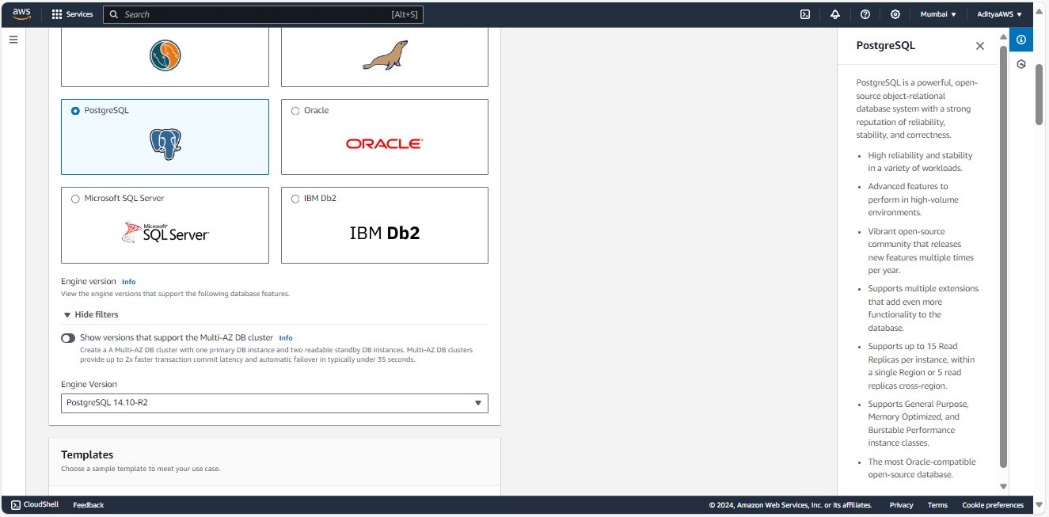
1. Choose the “Standard create” creation method.



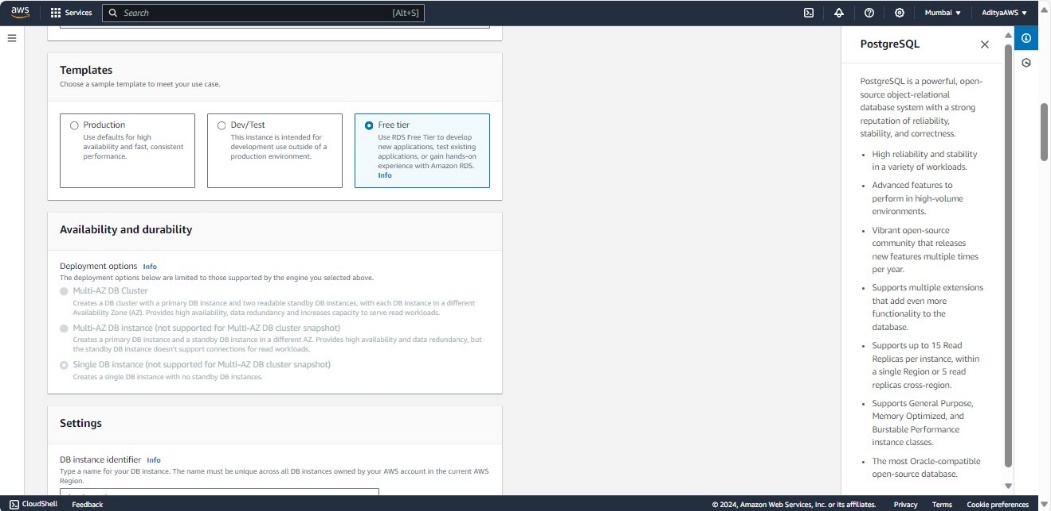
1. Select the PostgreSQL in the engine options.



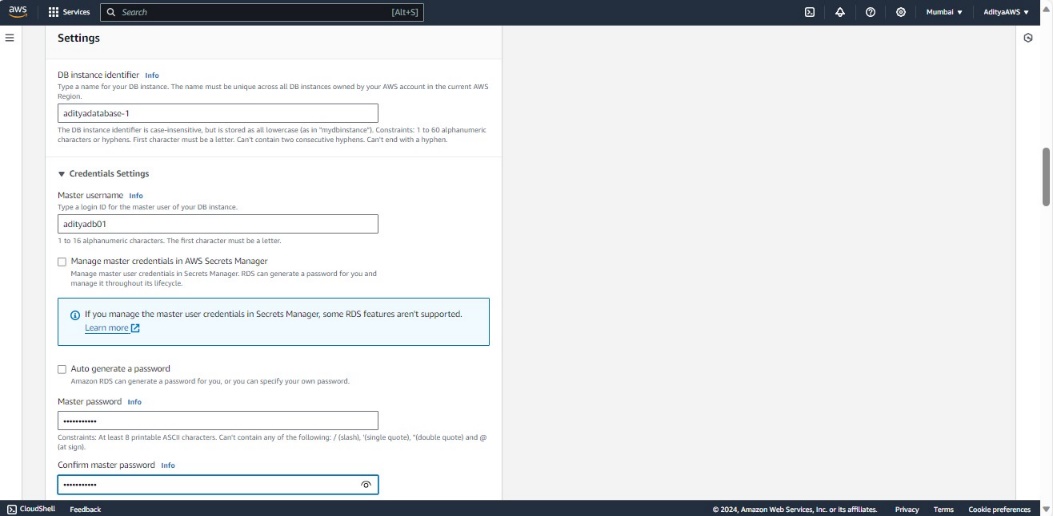
1. Then, select the engine version (I have selected PostgreSQL 14.10-R2).



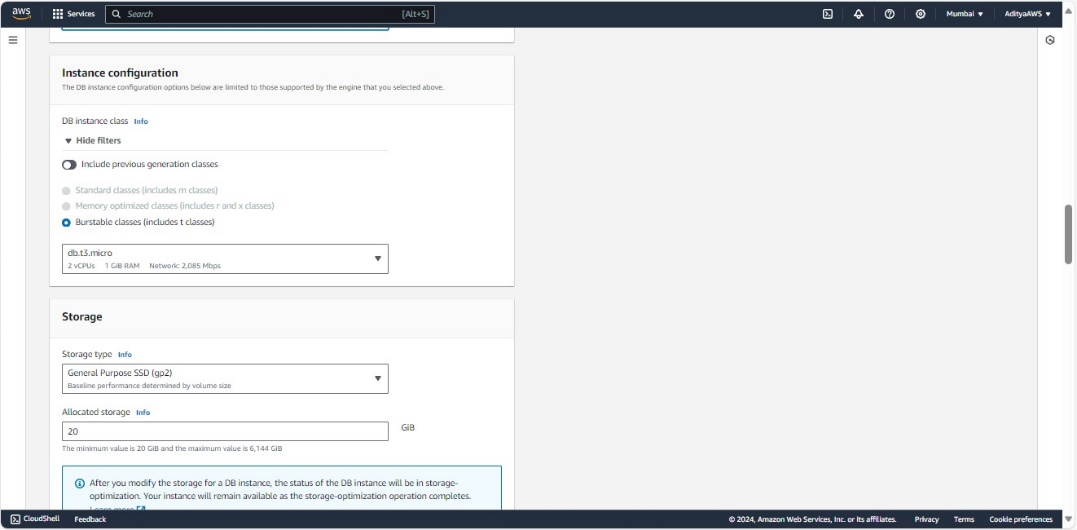
1. Select the “Free tier” template.



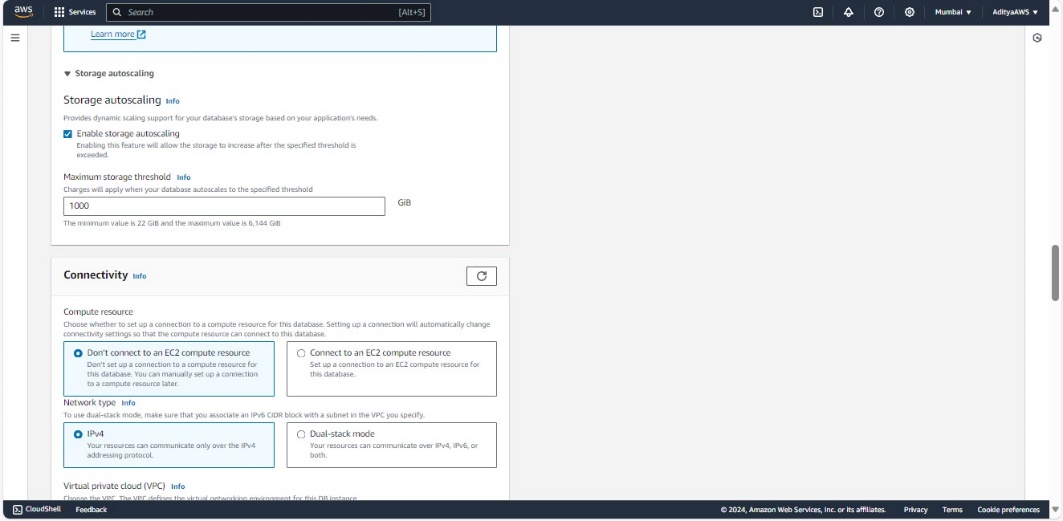
1. Now, in the settings section set the “DB instance identifier”, “Master username”, and “Master password”.



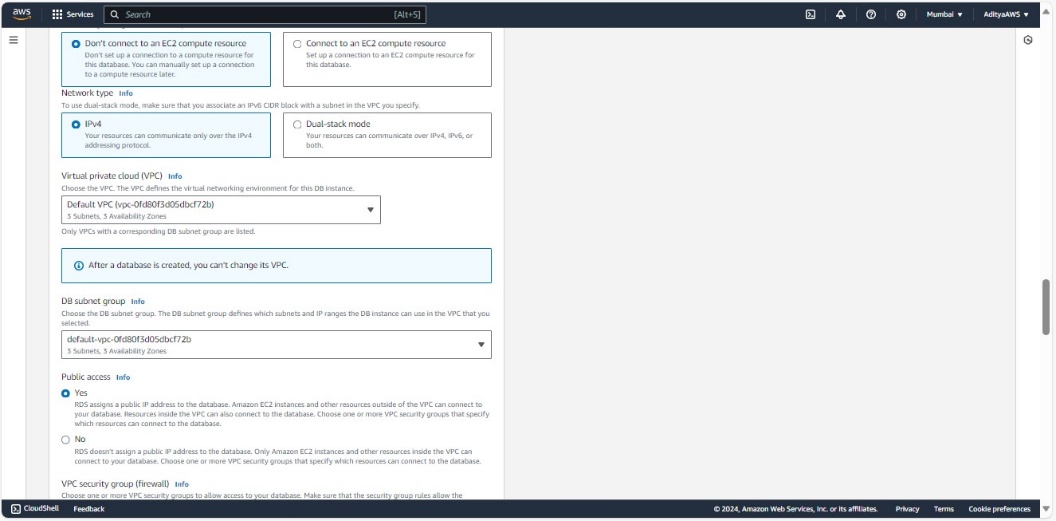
1. In instance configuration select “db.t3.micro” and leave others as it is.



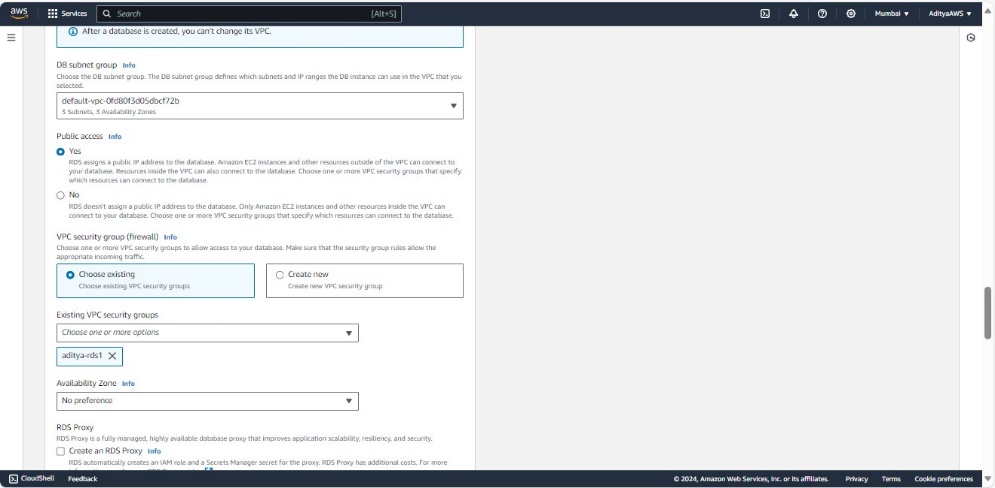
1. Enable storage autoscaling.



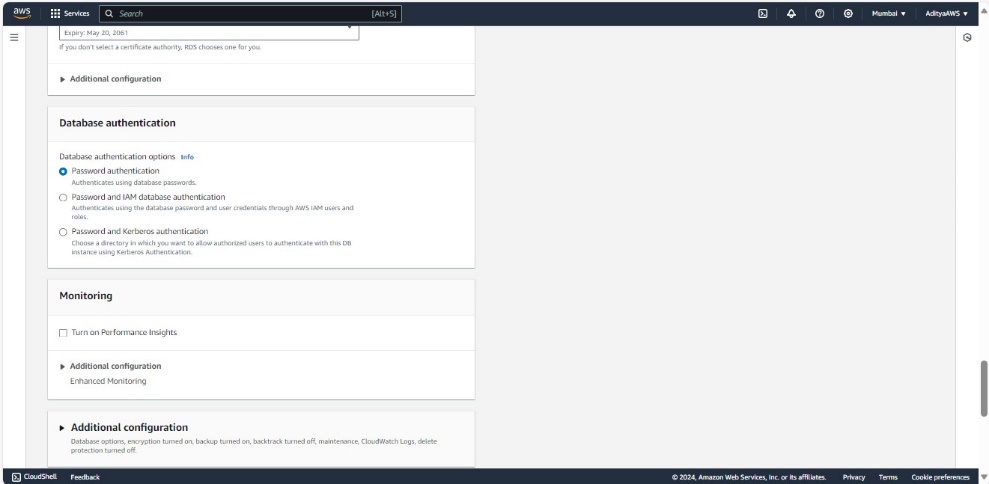
1. Enable the public access and leave others as default.



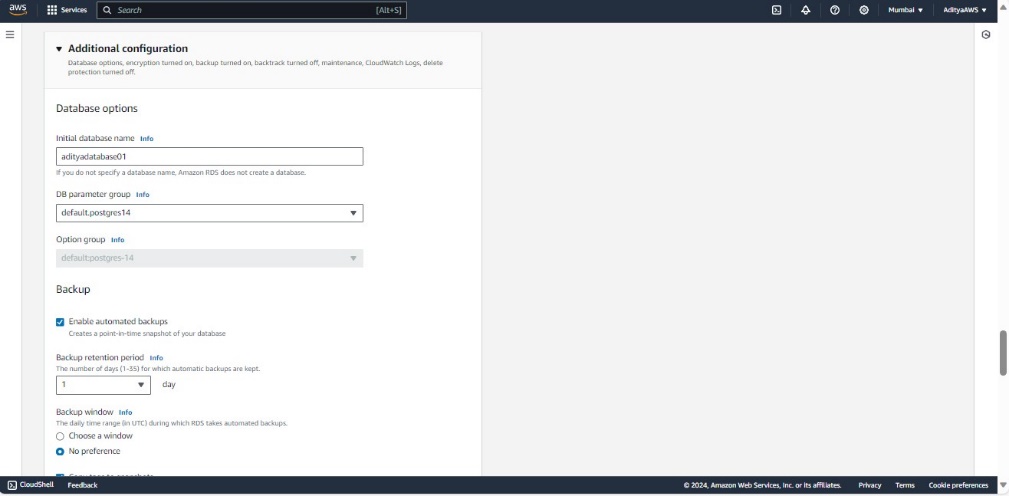
1. In VPC security group choose the VPC security group or create the new one as I have already created a security group “aditya-rds1” for this DB instance so, I have chosen it from the existing.



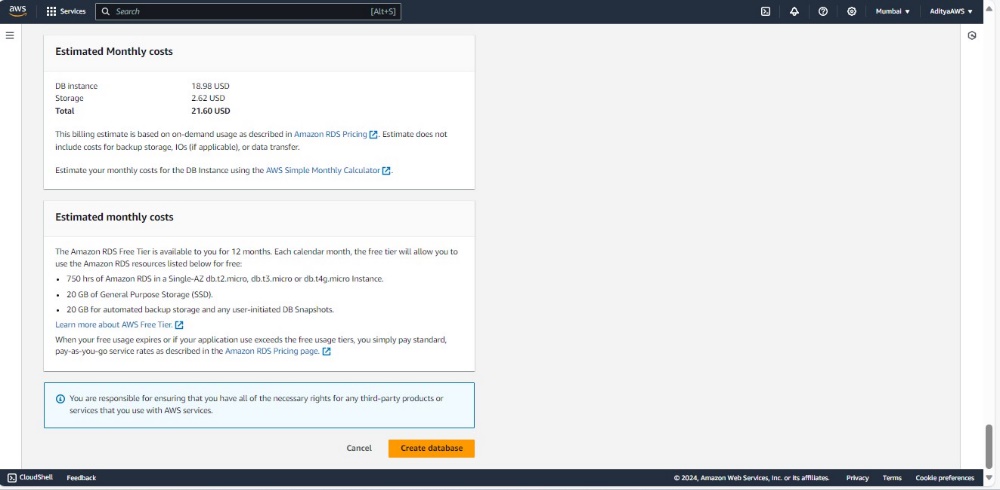
1. In database authentication select the “Password authentication”.



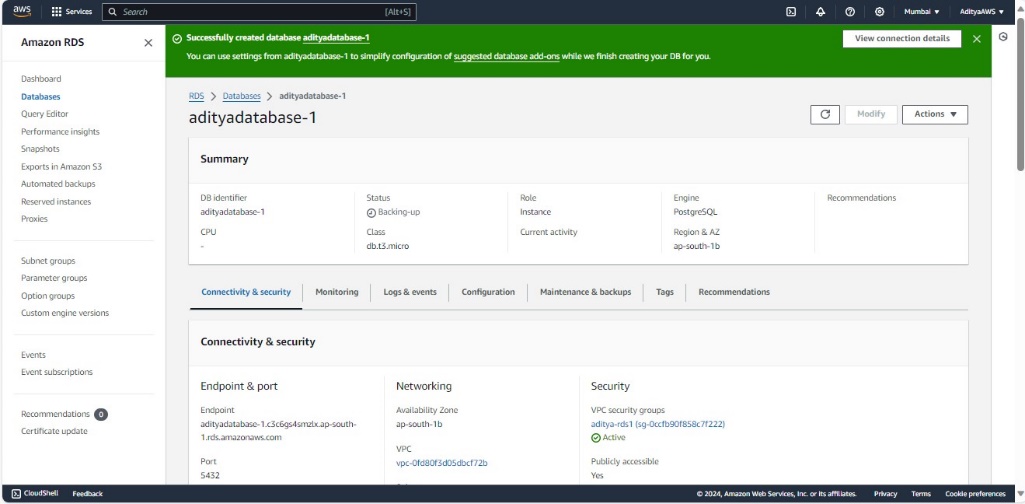
1. Click “Additional configuration” for database options then, set the initial database name and leave others as it is.



1. Finally, click on “Create database” to create the database.

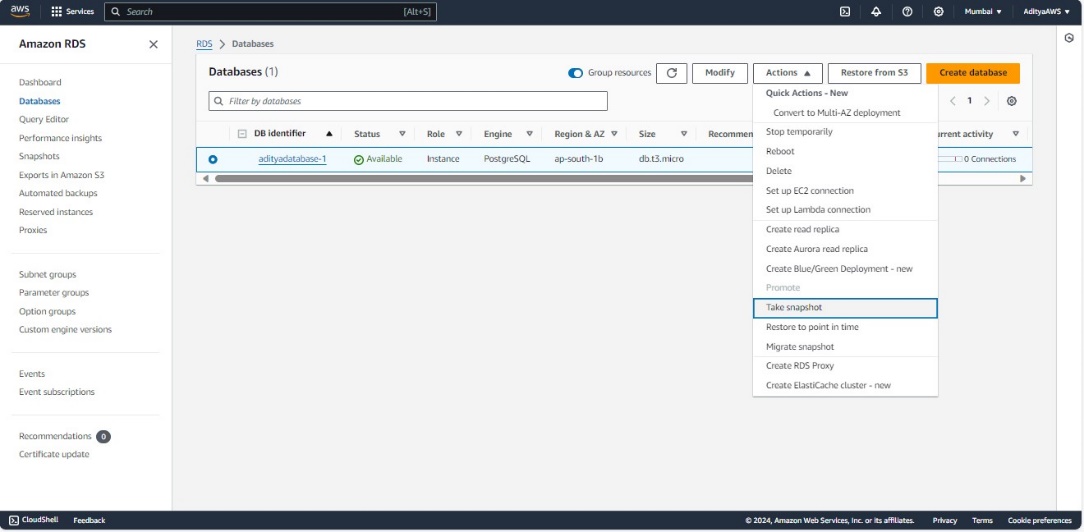


Now, the database “adityadatabase-1” has been created successfully.

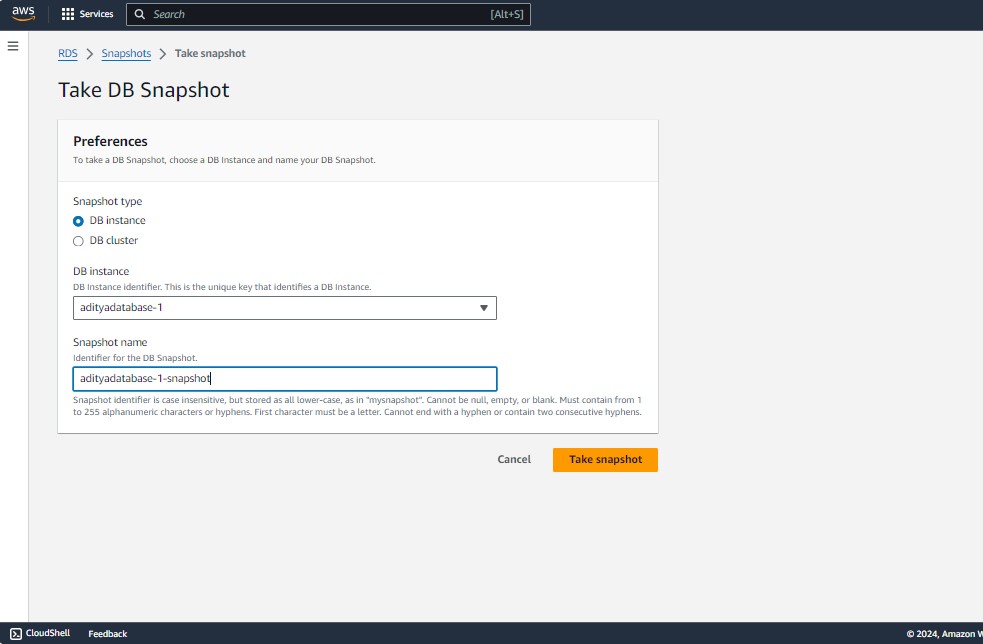


**In order to take the manual snapshot of this RDS instance:**

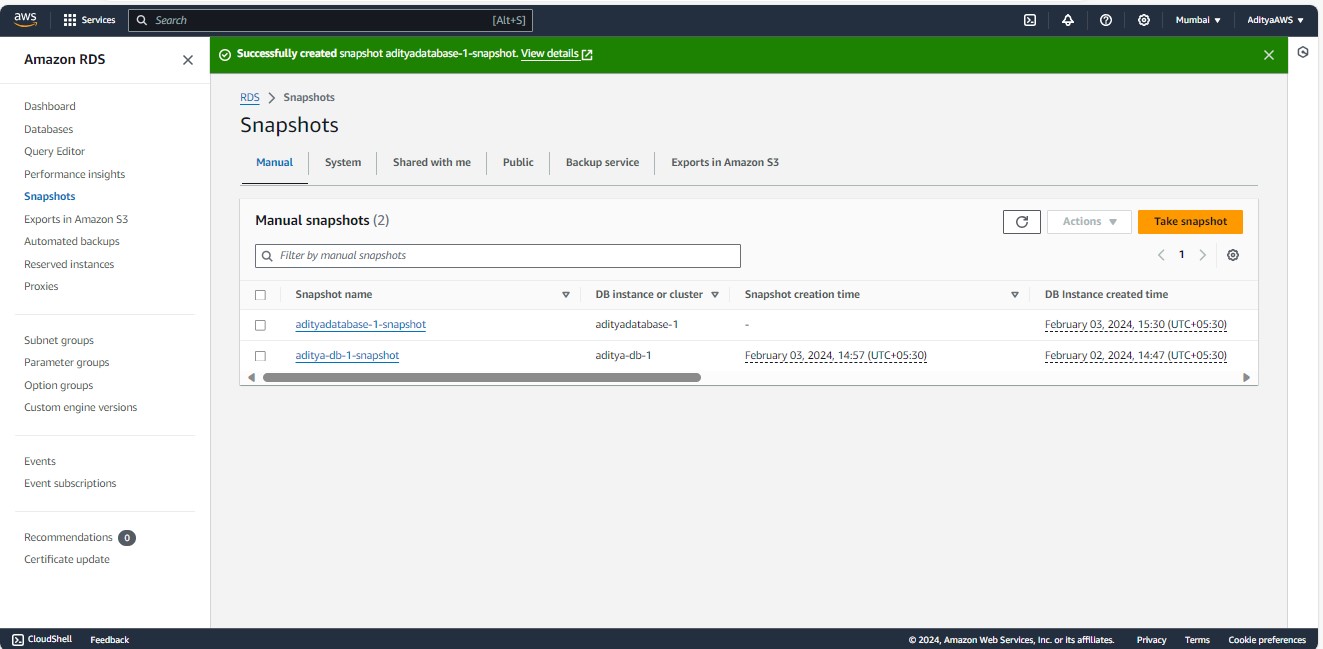
1. When the status of the RDS shows available then, select the database and click on “Actions” and in the dropdown click on “Take snapshot”.



1. Now, name the snapshot (as I have named it “adityadatabase-1-snapshot”) and click on “Take snapshot”.

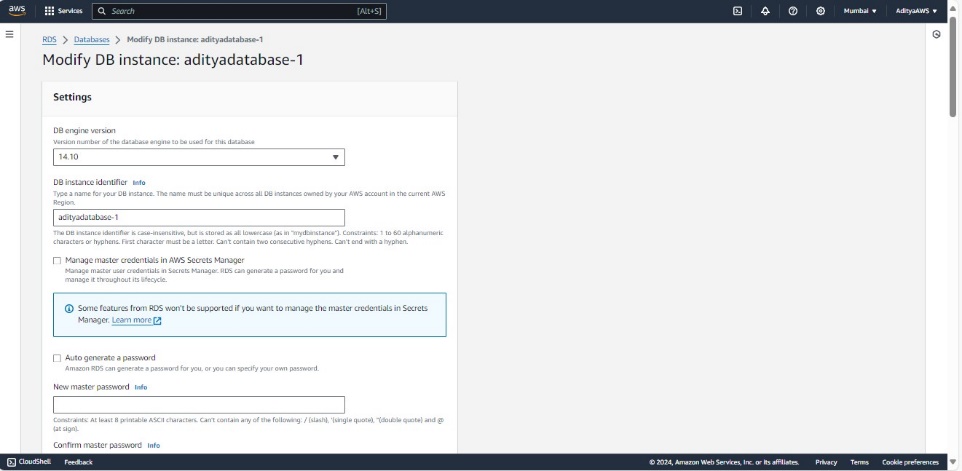
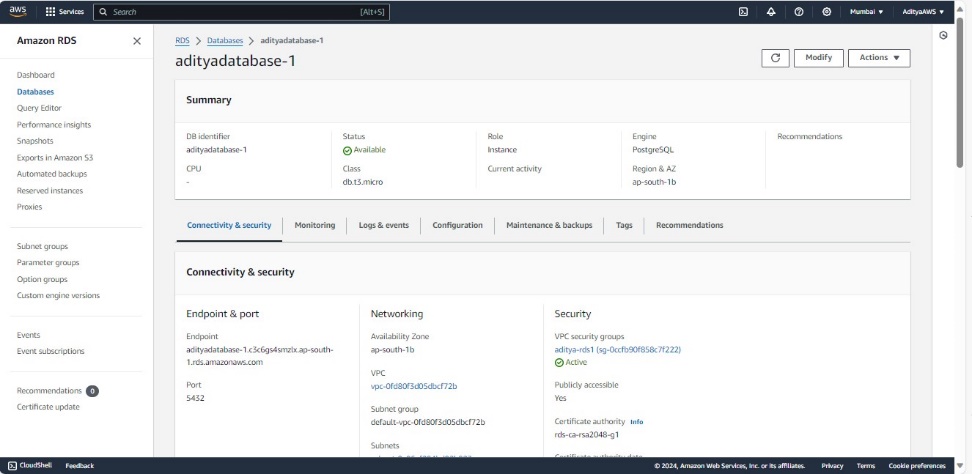


Now, it is created successfully

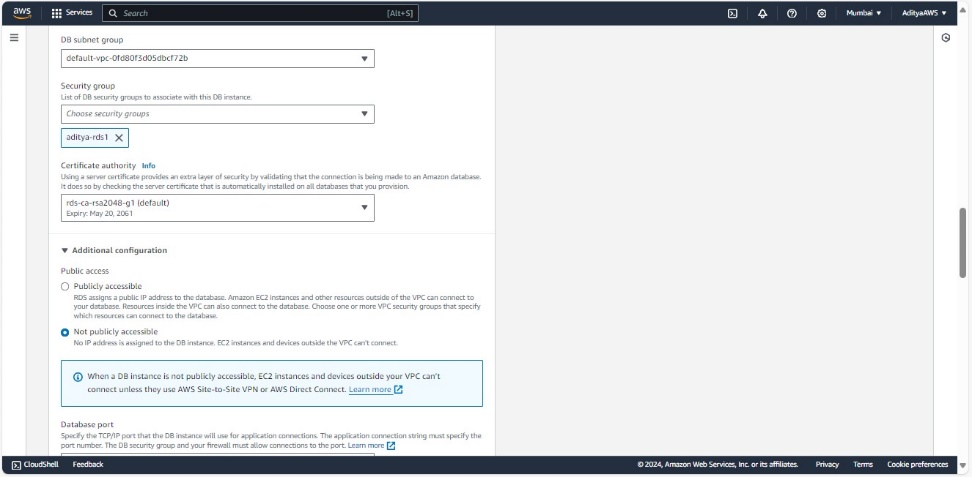


**Now, I am modifying the database (disabling the public access) and editing the inbound rules and outbound rules of the VPC security group which is associated with this database.**

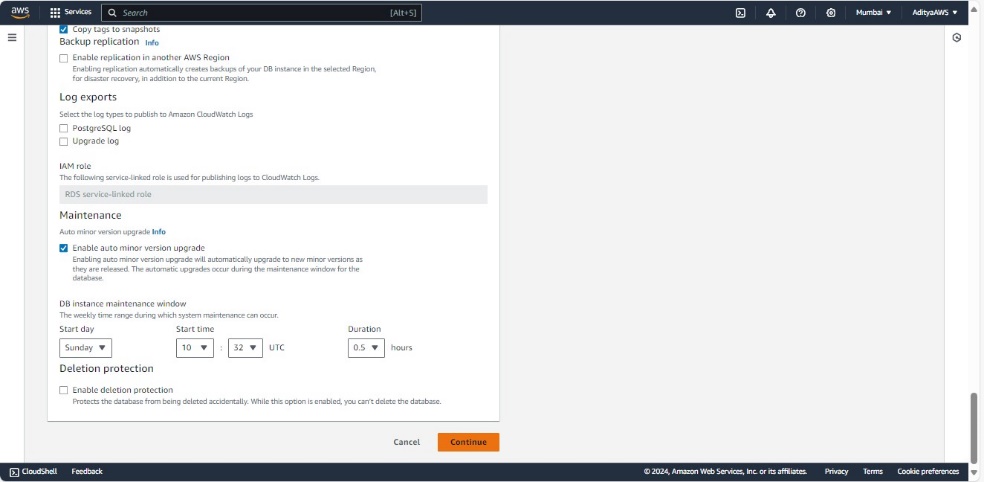
1. Click on “Modify”



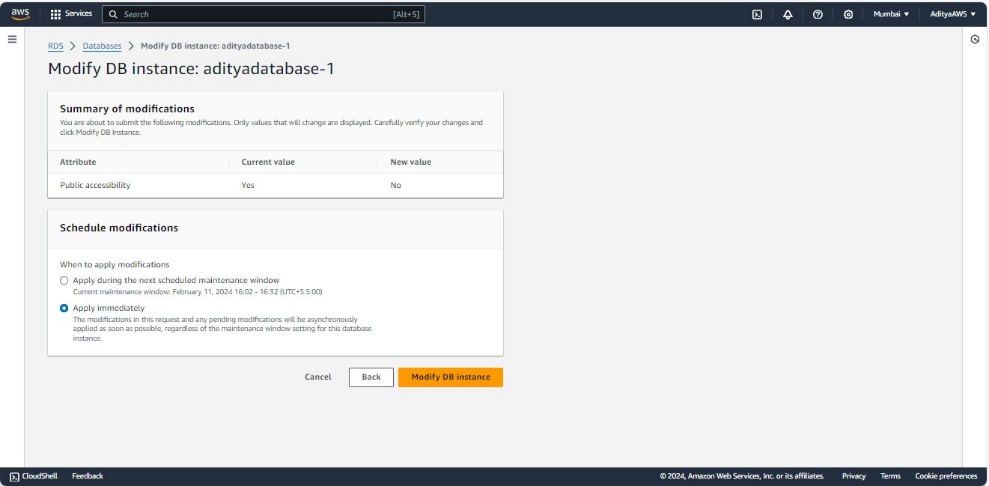
1. In the additional configuration select “Not publicly accessible” in public access section.



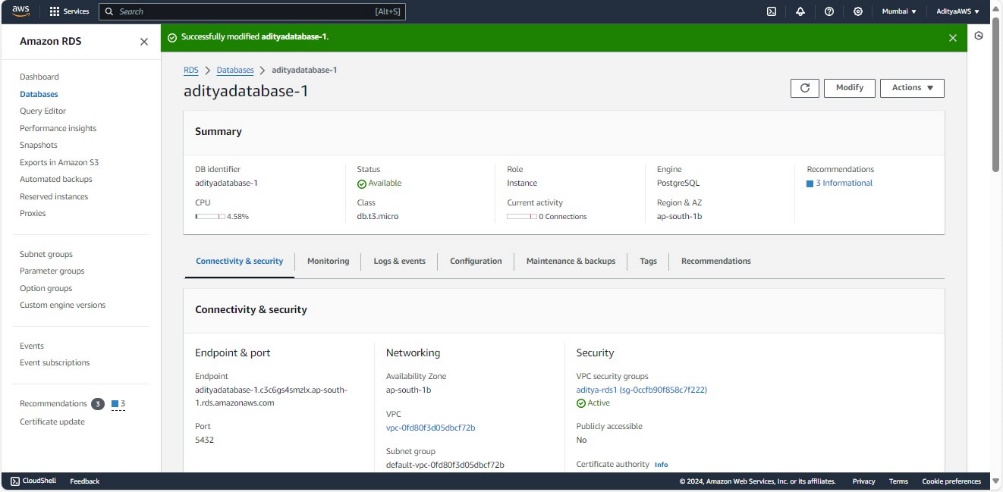
1. Then, click “Continue”.



1. Now, check the summary and schedule the modifications (I have selected “Apply immediately”). Click on “Modify DB instance”.

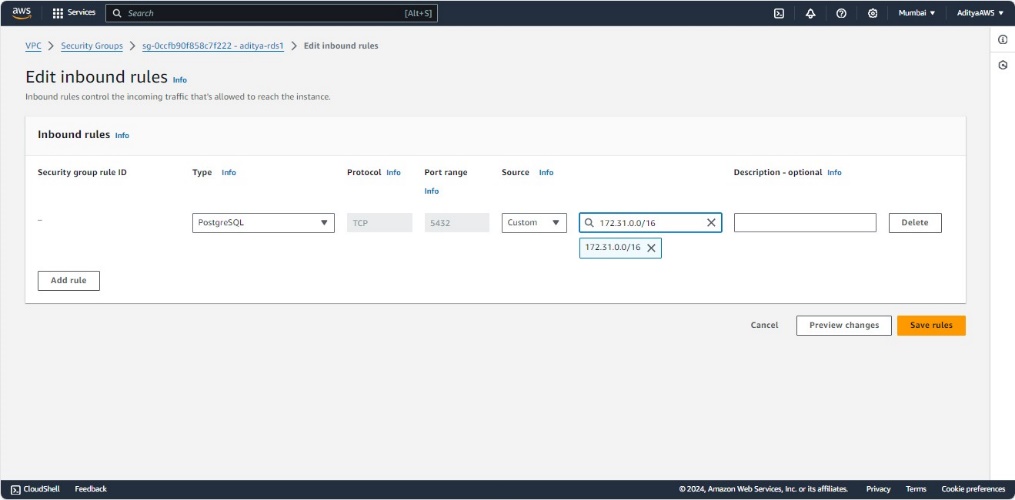


Successfully modified “adityadatabase-1”.

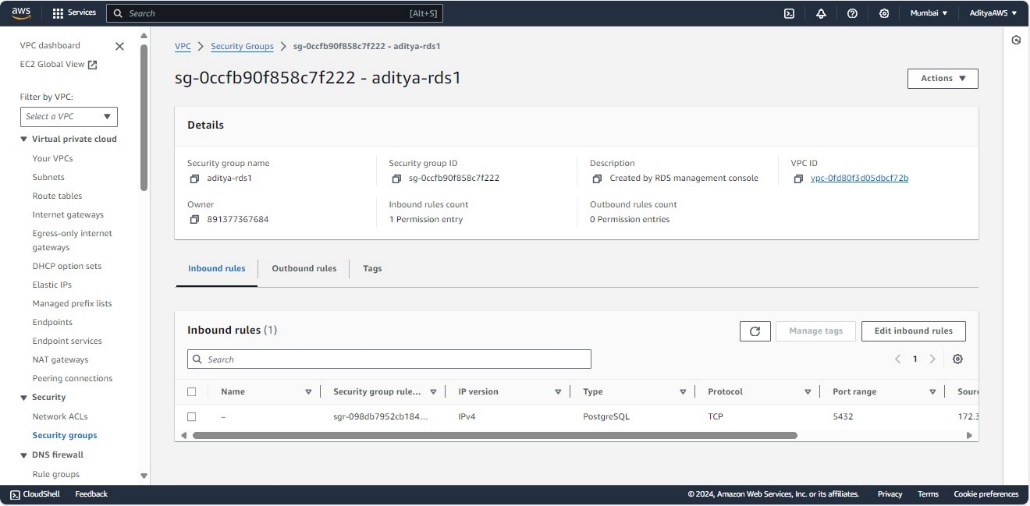


Now, editing the inbound rules and outbound rules of the VPC security group(aditya-rds1)

1. Firstly, I have edited the inbound rules.

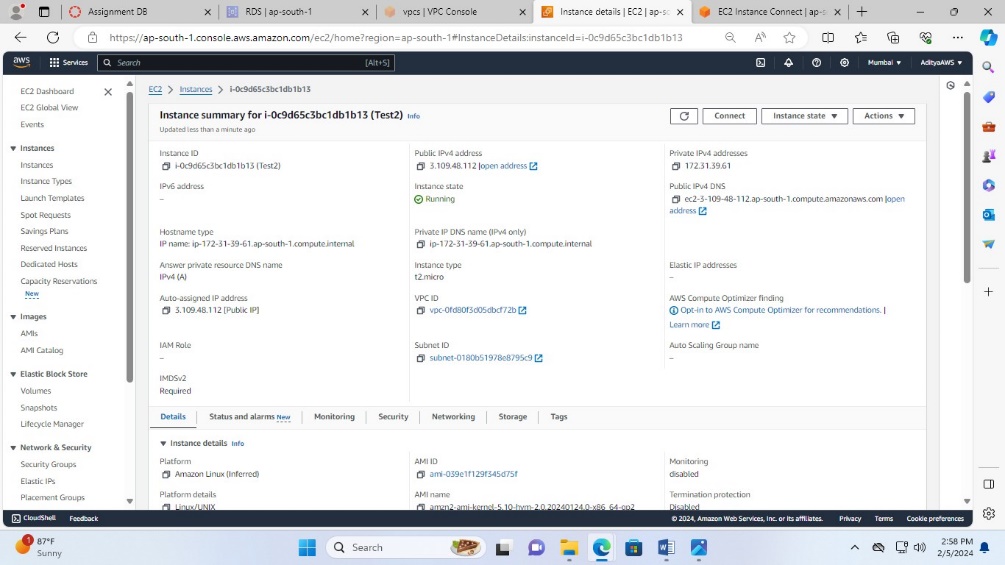


1. And removed the outbound rules (no permission entries for outbound rules).

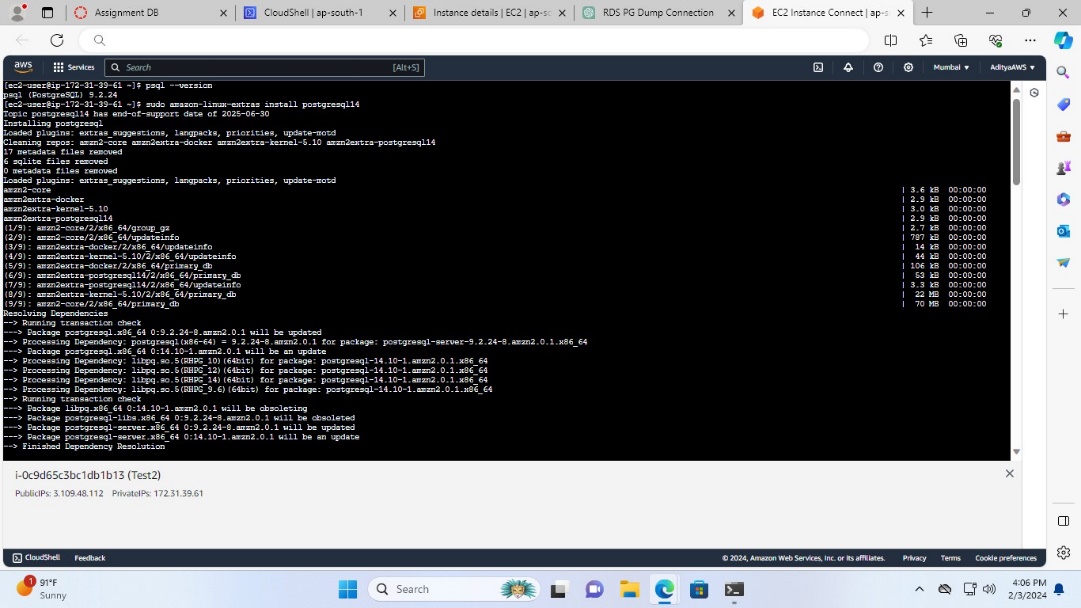


**In order to do PG Dump of RDS using connection string:**

1. Firstly, connect to an EC2 instance (I have connected to “Test2” instance). This instance platform is Amazon Linux,



1. So, after the connection I have installed the postgresql14 with the command **sudo amazon-linux-extras install postgresql14.**



1. Now, to do the PG Dump, run the command

**pg\_dump -h adityadatabase-1.c3c6gs4smzlx.ap-south-1.rds.amazonaws.com -U adityadb01 -Fc adityadatabase01 > backup.dump**

then, enter the master password of the RDS

and to check that backup.dump is created or not run the **ls** command.

