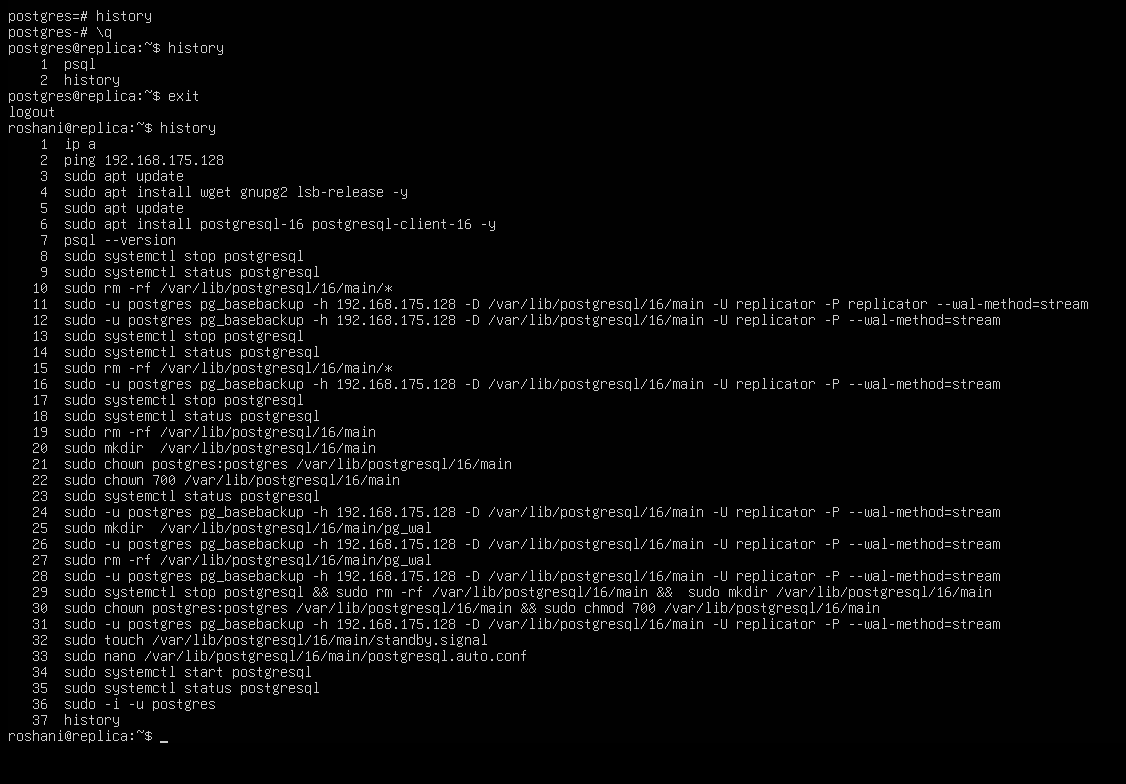
Primary Replica Snippets with commands





Replica VM snippet commands:



**1.Replication Status check from primary VM**

SSH/login into your **Primary VM** and run:

Sudo -i -u postgres

psql

SELECT client\_addr, state, sync\_state, write\_lag, flush\_lag, replay\_lag

FROM pg\_stat\_replication;

From replica:

SELECT pg\_is\_in\_recovery();

sudo systemctl stop postgresql

**2: From Replica VM (Standby) promoting it as Primary replica by failover:**

sudo -i -u postgres

pg\_ctl promote -D /var/lib/postgresql/16/main

psql

SELECT pg\_is\_in\_recovery();

**f = false** → means Replica is no longer in recovery → it's now **acting as Primary**.

CREATE TABLE demo\_failover(id INT);

INSERT INTO demo\_failover VALUES (1);

SELECT \* FROM demo\_failover;

**Master username**

Postgres

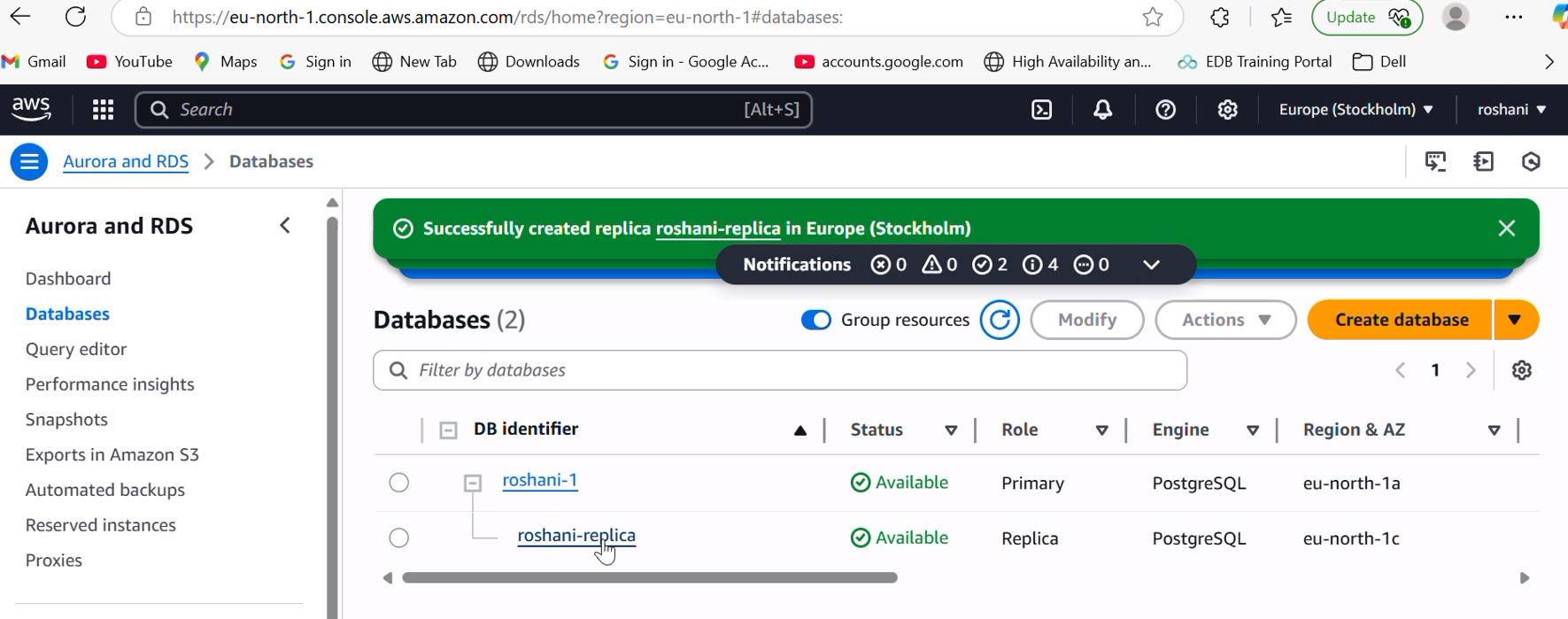
**AWS RDS Demo for Failover**

(Due to free tier subscription of AWS limited features were available for demonstrating the study)

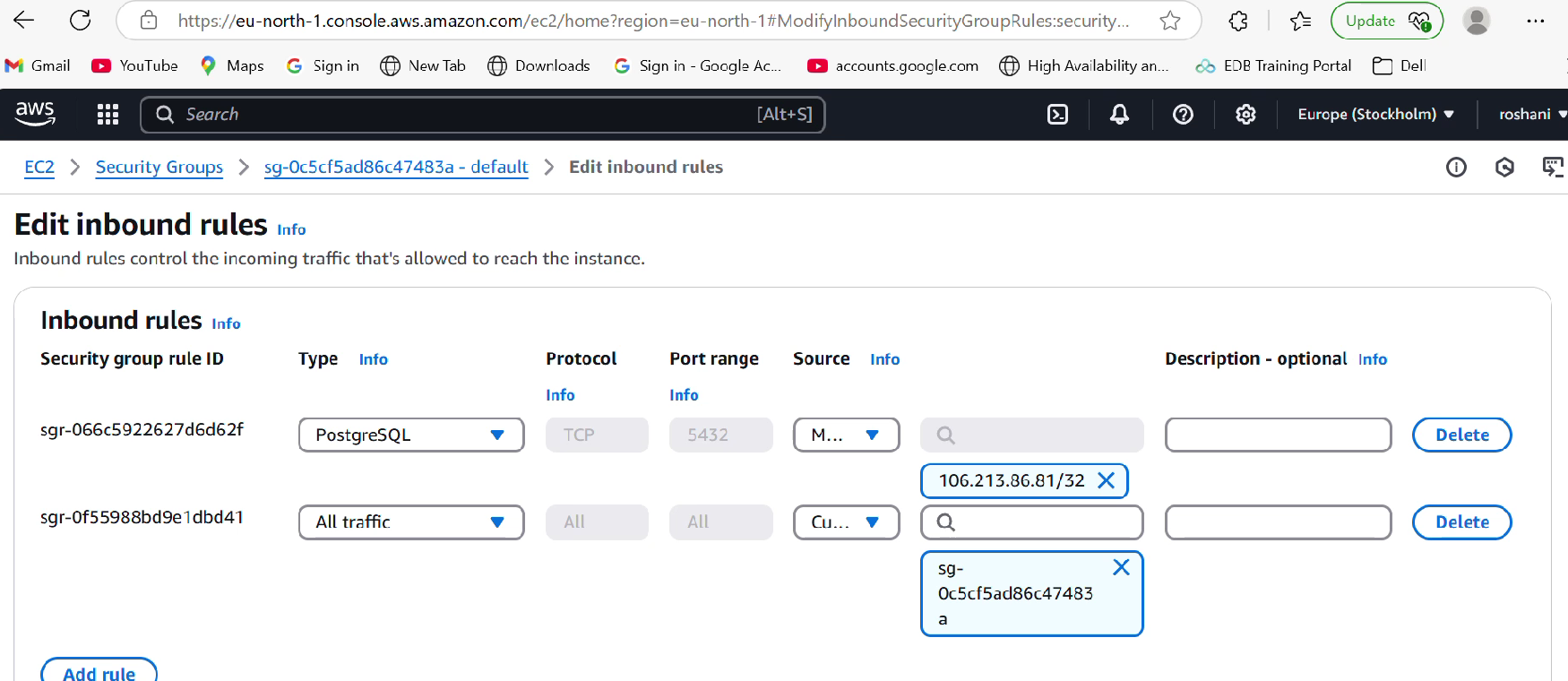
**1.Created two VM in different region**

Primary VM - roshani-1.cf4g24se079r.eu-north-1.rds.amazonaws.com

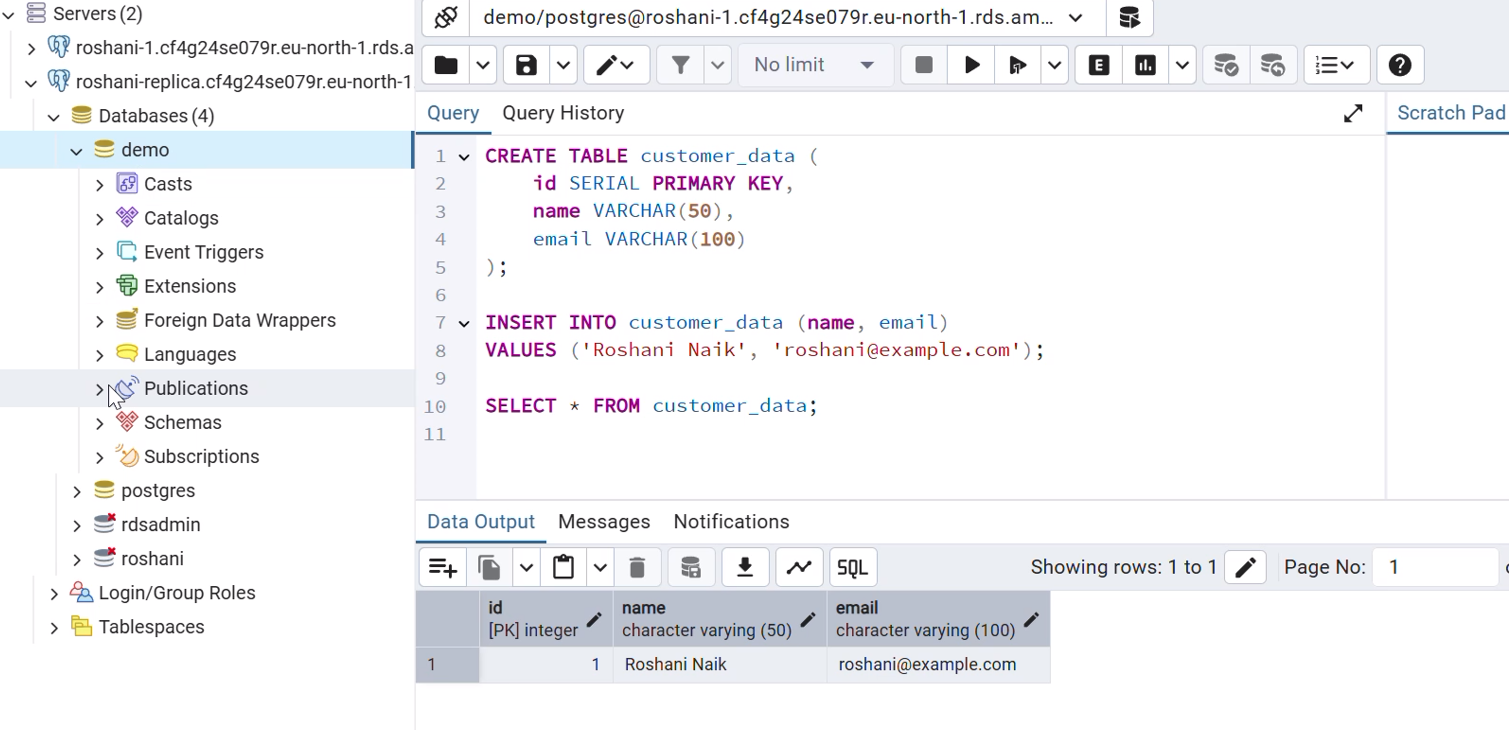
Replica VM - roshani-replica.cf4g24se079r.eu-north-1.rds.amazonaws.com

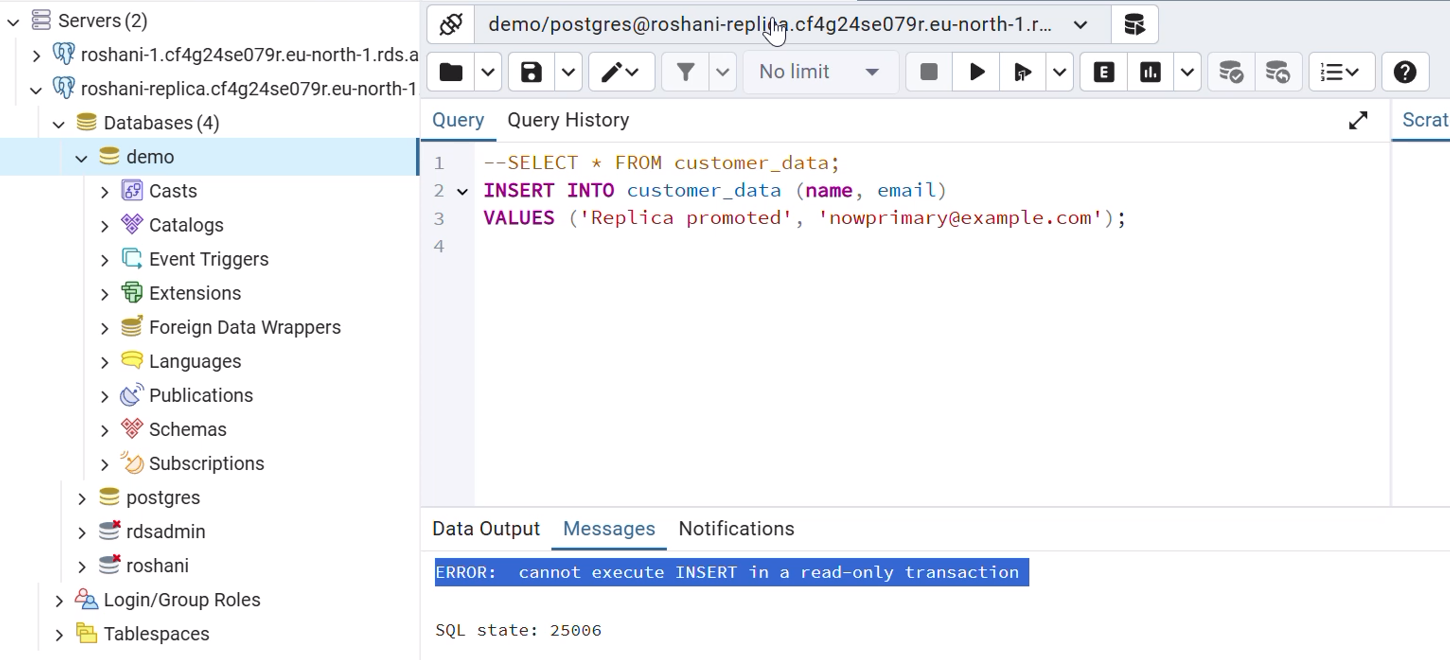


**2. Security configurations for allowing connections to the machines created via PGADMIN tool**

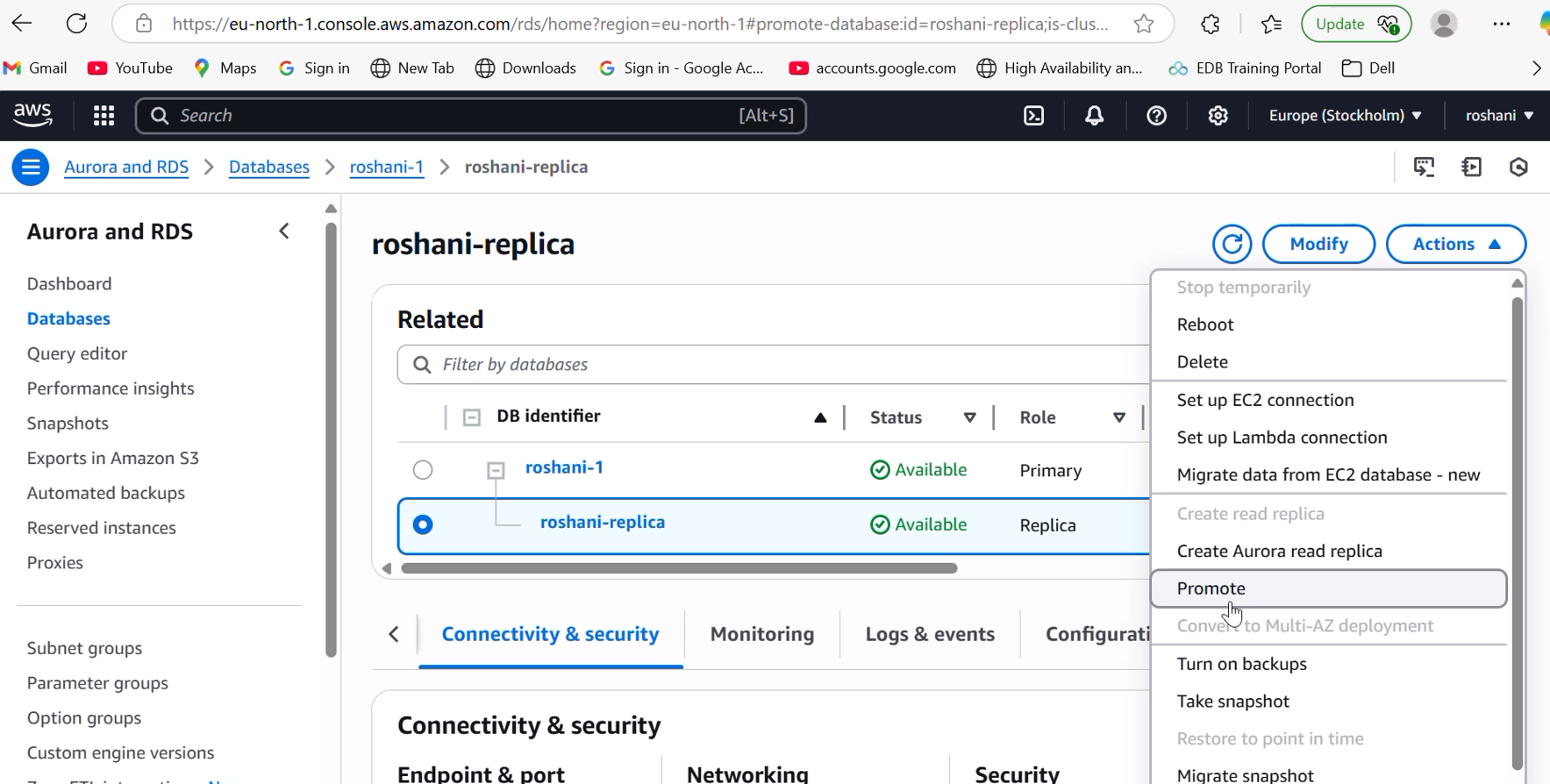


**3.Accessing VM’s created through PGAdmin tool for testing and validation**





**4. Promoting the replica read only VM to be primary for read write connections**



**5.Replica VM promoted to Primary accepting write operation**

