🚀Deploying Angular Application on EC2 Instance with RDS PostgreSQL Database using PM2 in AWS Infrastructure

Hello Folks ! 👏 Wanted to share my recent learning experience 🚀 in deploying a angular application on Amazon EC2 with RDS Postgres SQL.

👦 🏫 Jatin Shharma

It's been an incredible learning curve and I wanted to share my the steps I followed, challenges I faced and steps I took to overcome them.

Prerequisites:

✓ Basic knowledge of AWS services (EC2, RDS).

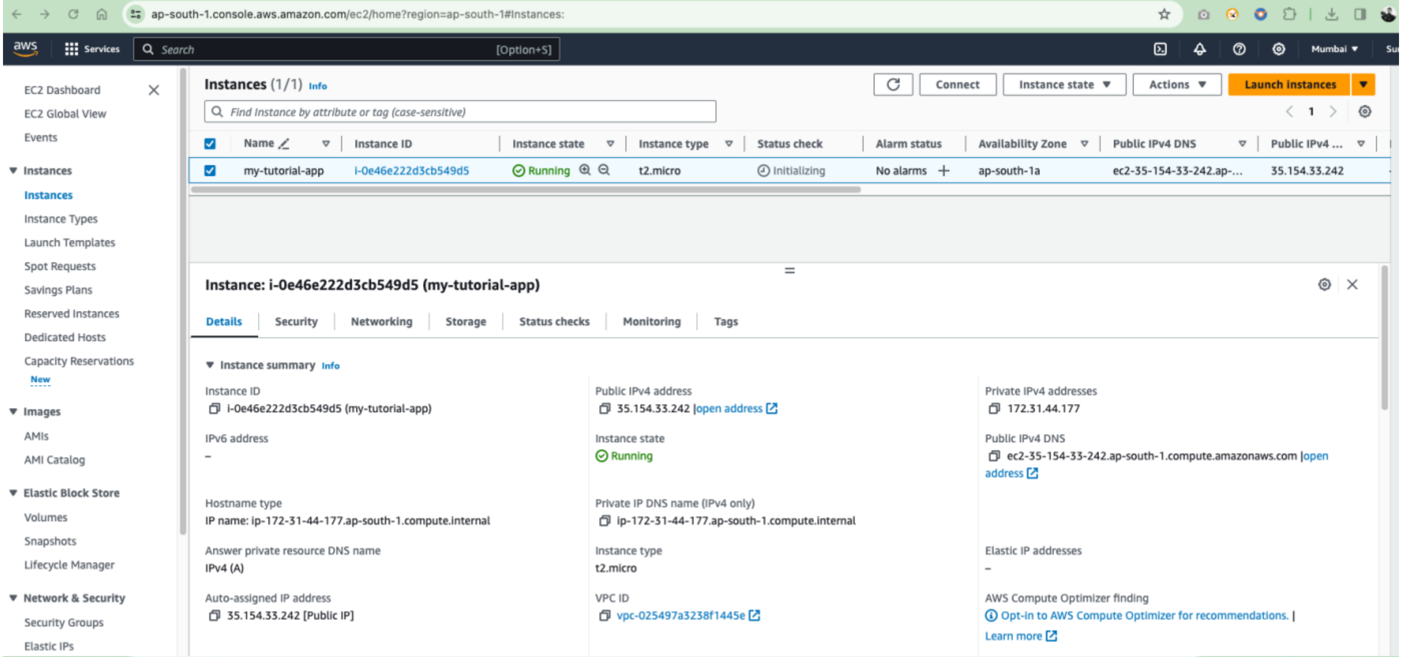
✓ Familiarity with Angular, Node.js, and npm.

✓ An AWS account with necessary permissions to create EC2 instances and RDS databases.

1️⃣Launch an EC2 Instance

Refer below article for step-by-step process of creating EC2 Instance.

https://www.linkedin.com/pulse/deploying-static-website-ec2-apache2-ubuntu-suraj-shrikant-khopkar-9e51f?trackingId=FrXfEdX5R1CLvQE0fRvLJg%3D%3D&lipi=urn%3Ali%3Apage%3Ad_flagship3_detail_base%3BcBFlPbbdSTKqViqZTirsmg%3D%3D



Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
my-tutorial-app	i-0e46e222d3cb549d5	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-35-154-33-242.ap-...	35.154.33.242

Instance: i-0e46e222d3cb549d5 (my-tutorial-app)

Instance summary Info

Instance ID: i-0e46e222d3cb549d5 (my-tutorial-app)

Public IPv4 address: 35.154.33.242 |open address

Private IPv4 addresses: 172.31.44.177

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-35-154-33-242.ap-south-1.compute.amazonaws.com |open address

Hostname type: IP name: ip-172-31-44-177.ap-south-1.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-44-177.ap-south-1.compute.internal

Answer private resource DNS name: IPv4 (A)

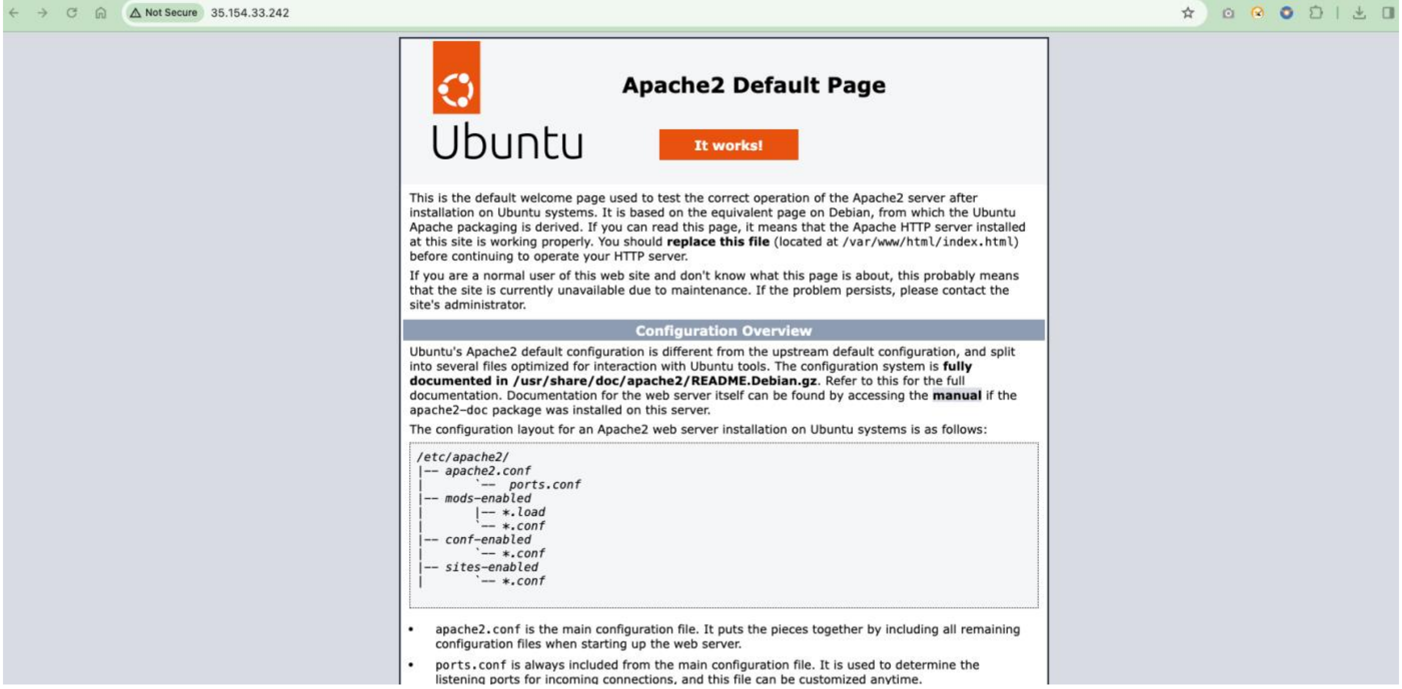
Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 35.154.33.242 [Public IP]

VPC ID: vpc-025497a3238f1445e

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

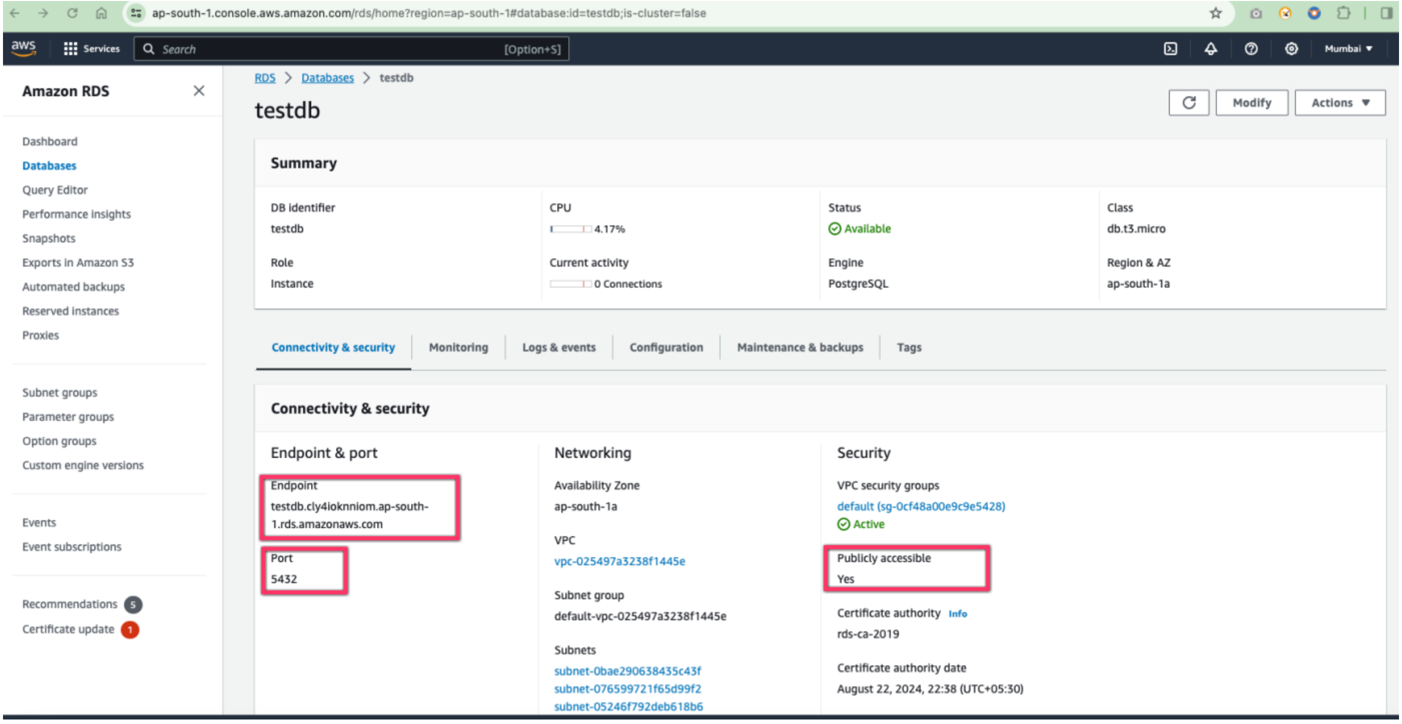
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

2 Install required dependencies on the server i.e. Java, Apache2, nodejs, npm, n.

*(** n is installed as we need specific version of nodejs i.e. 16)*

3 Setting up RDS Postgres Database server.

- Navigate to RDS in the AWS console and click "Create Database."
- Select PostgreSQL as the engine and configure your database settings.
- Make a note of the database endpoint, username, and password for later use.



testdb

Summary

DB identifier	CPU	Status	Class
testdb	4.17%	Available	db.t3.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	PostgreSQL	ap-south-1a

Connectivity & security

Endpoint & port

Endpoint
testdb.cly4iokniom.ap-south-1.rds.amazonaws.com

Port
5432

Networking

Availability Zone
ap-south-1a

VPC
vpc-025497a3238f1445e

Subnet group
default-vpc-025497a3238f1445e

Subnets
subnet-0bae290638435c43f
subnet-076599721f65d99f2
subnet-05246f792deb618b6

Security

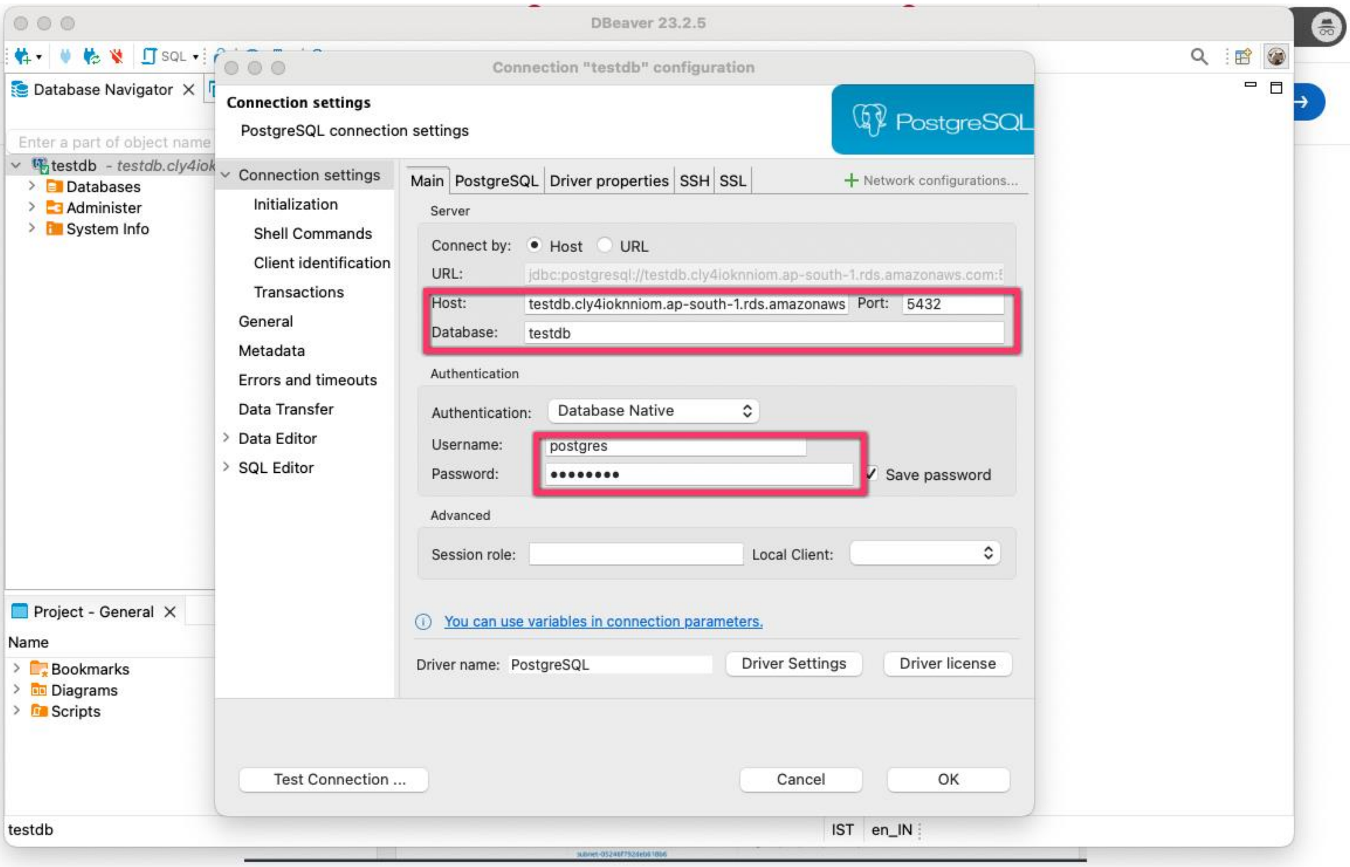
VPC security groups
default (sg-0cf48a00e9c9e5428)
Active

Publicly accessible
Yes

Certificate authority Info
rds-ca-2019

Certificate authority date
August 22, 2024, 22:38 (UTC+05:30)

4 Test DataBase Connection



Connection settings

PostgreSQL connection settings

Server

Connect by: ● Host ○ URL

URL: jdbc:postgresql://testdb.cly4iokniom.ap-south-1.rds.amazonaws.com:

Host: testdb.cly4iokniom.ap-south-1.rds.amazonaws Port: 5432

Database: testdb

Authentication

Authentication: Database Native

Username: postgres

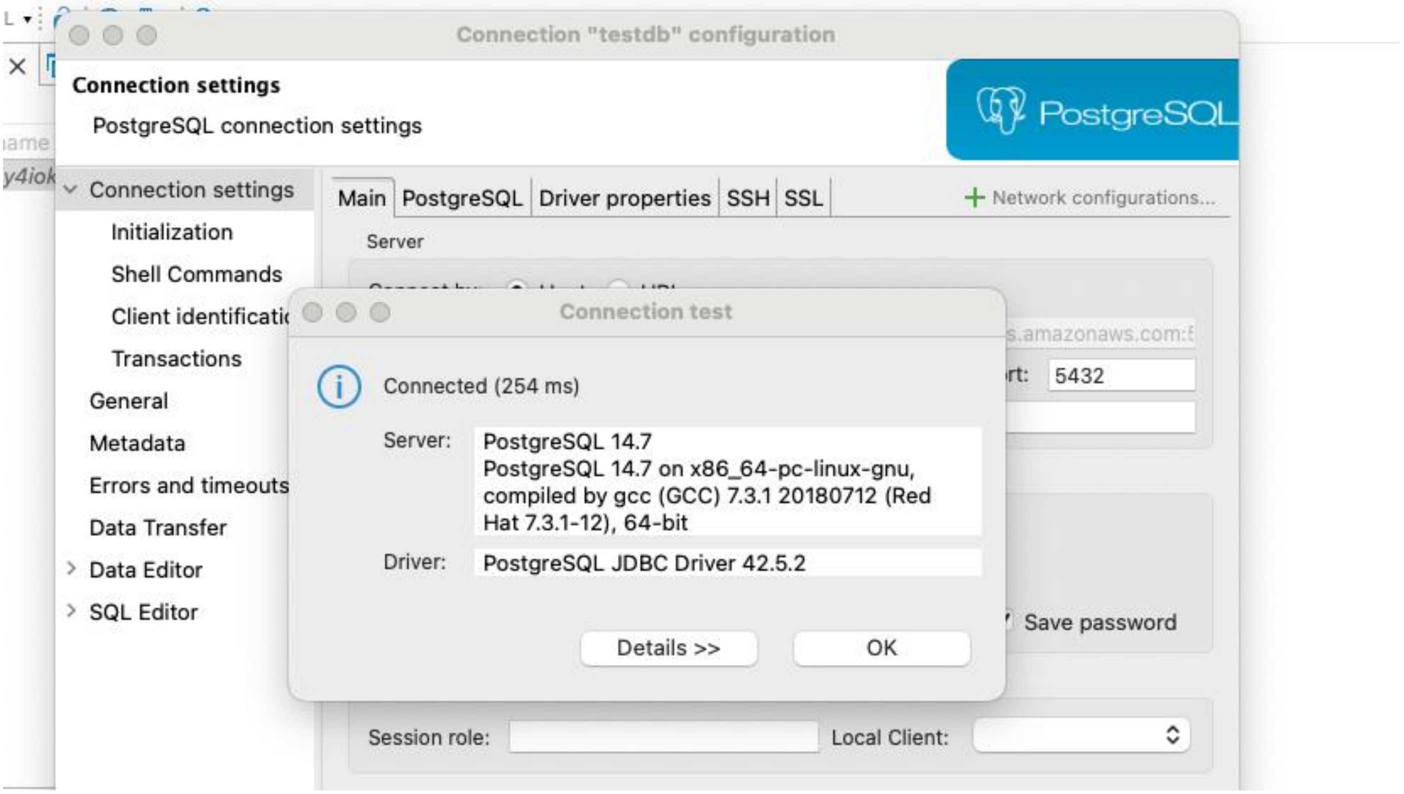
Password: •••••••• ✓ Save password

Advanced

Session role: Local Client:

You can use variables in connection parameters.

Driver name: PostgreSQL



Connected (254 ms)

Server: PostgreSQL 14.7
PostgreSQL 14.7 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 7.3.1 20180712 (Red Hat 7.3.1-12), 64-bit

Driver: PostgreSQL JDBC Driver 42.5.2

Port: 5432

Session role:

Local Client:

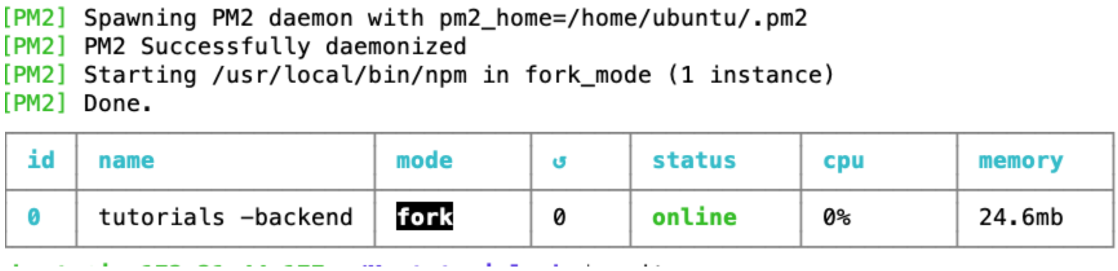
5 Transfer project files on the server.

- *scp -i "mytutorialkeypair.pem" My-tutorials-be.zip ubuntu@ec2-13-232-74-85.ap-south-1.compute.amazonaws.com:/home/ubuntu*
- *scp -i "mytutorialkeypair.pem" dist.zip ubuntu@ec2-13-232-74-85.ap-south-1.compute.amazonaws.com:/home/ubuntu*

6 Deploy Backend on the server

- Install PM2 globally on the instance using "npm install -g pm2."
- Start your backend application using "npm install" followed by "pm2 start npm -- start."
- Configure PM2 to start automatically on server reboot.

(PM2 is process manager tool which acts as a deamon process)



```
[PM2] Spawning PM2 daemon with pm2_home=/home/ubuntu/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /usr/local/bin/npm in fork_mode (1 instance)
[PM2] Done.
┌────┬────────────────────┬──────────┬──────┬───────────┬──────────┬──────────┐
│ id │ name               │ mode     │ ↺    │ status    │ cpu      │ memory   │
├────┼────────────────────┼──────────┼──────┼───────────┼──────────┼──────────┤
│ 0  │ tutorials -backend │ fork     │ 0    │ online    │ 0%       │ 24.6mb   │
└────┴────────────────────┴──────────┴──────┴───────────┴──────────┴──────────┘
```