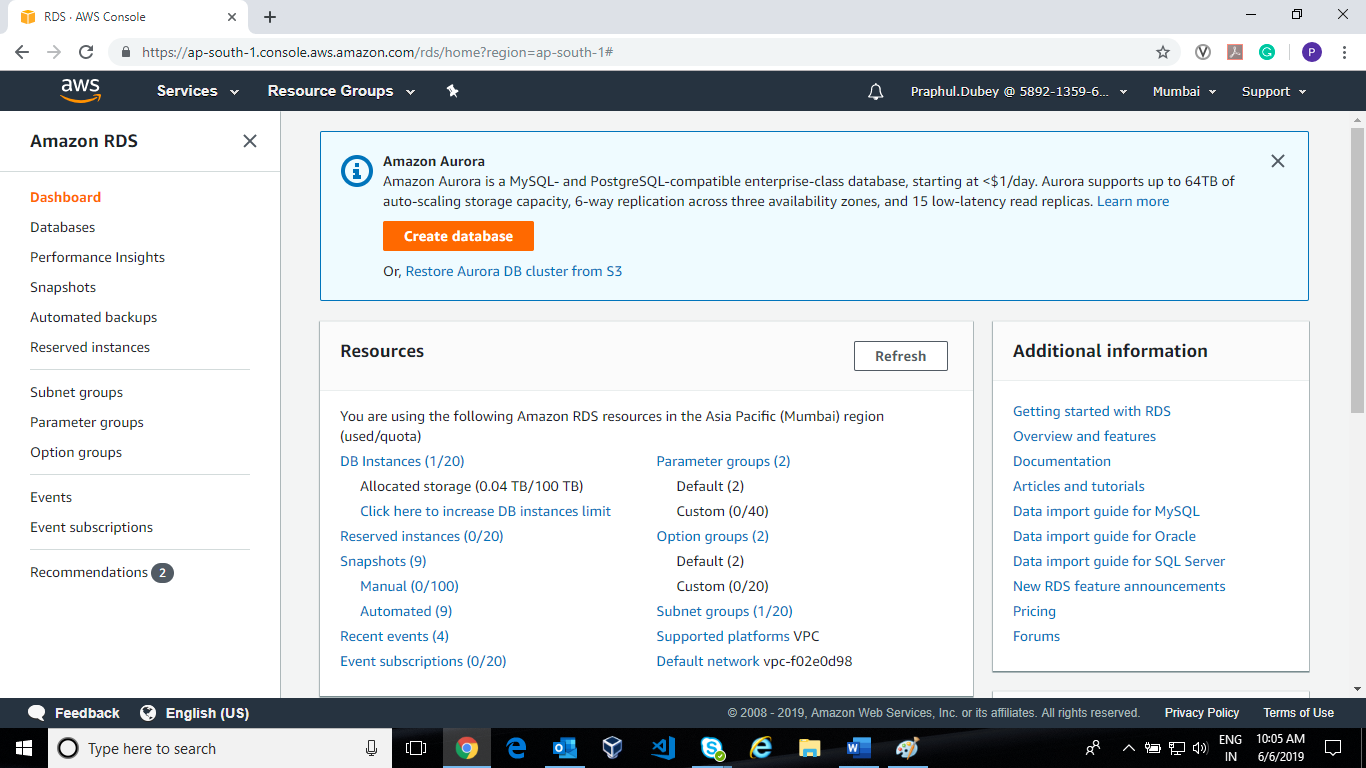
SONARQUBES

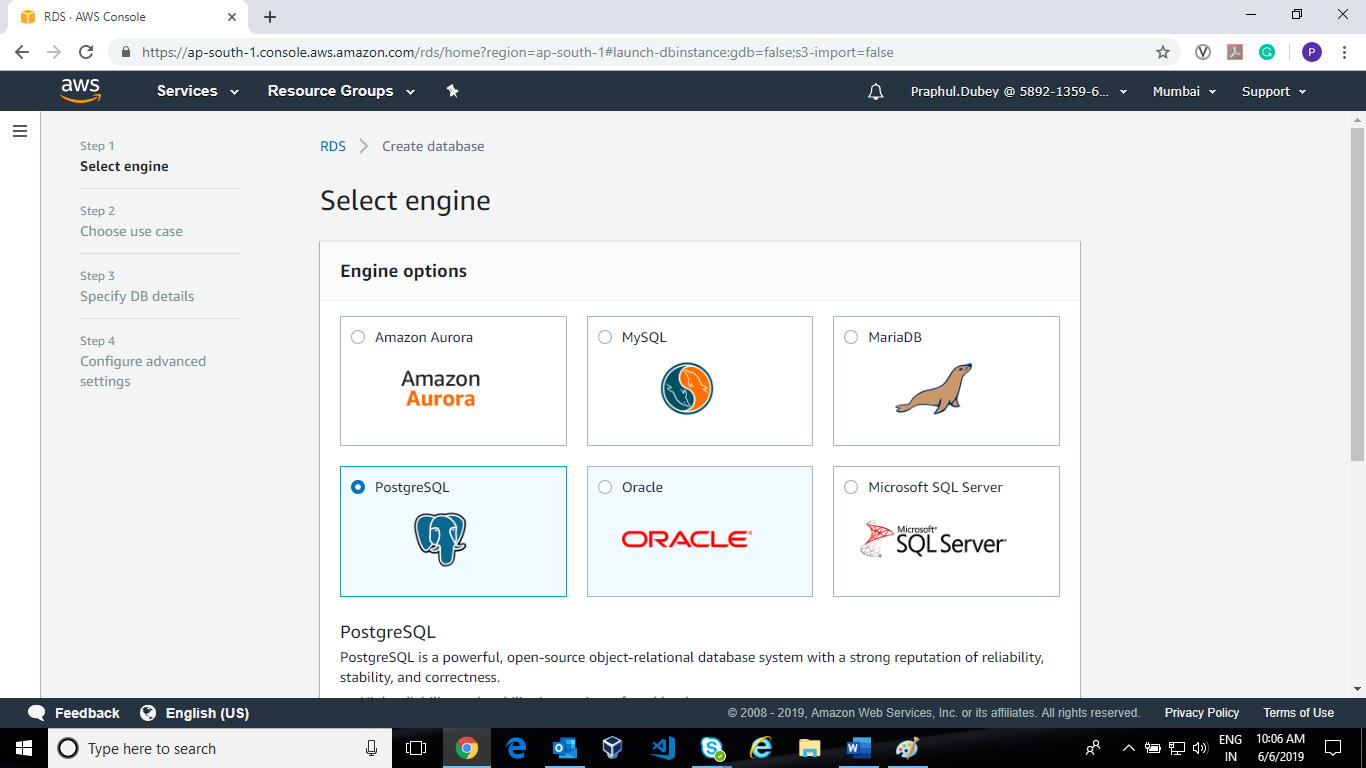
Steps to install SonarQube for enterprise.

1. Install managed AWS Postgres DB

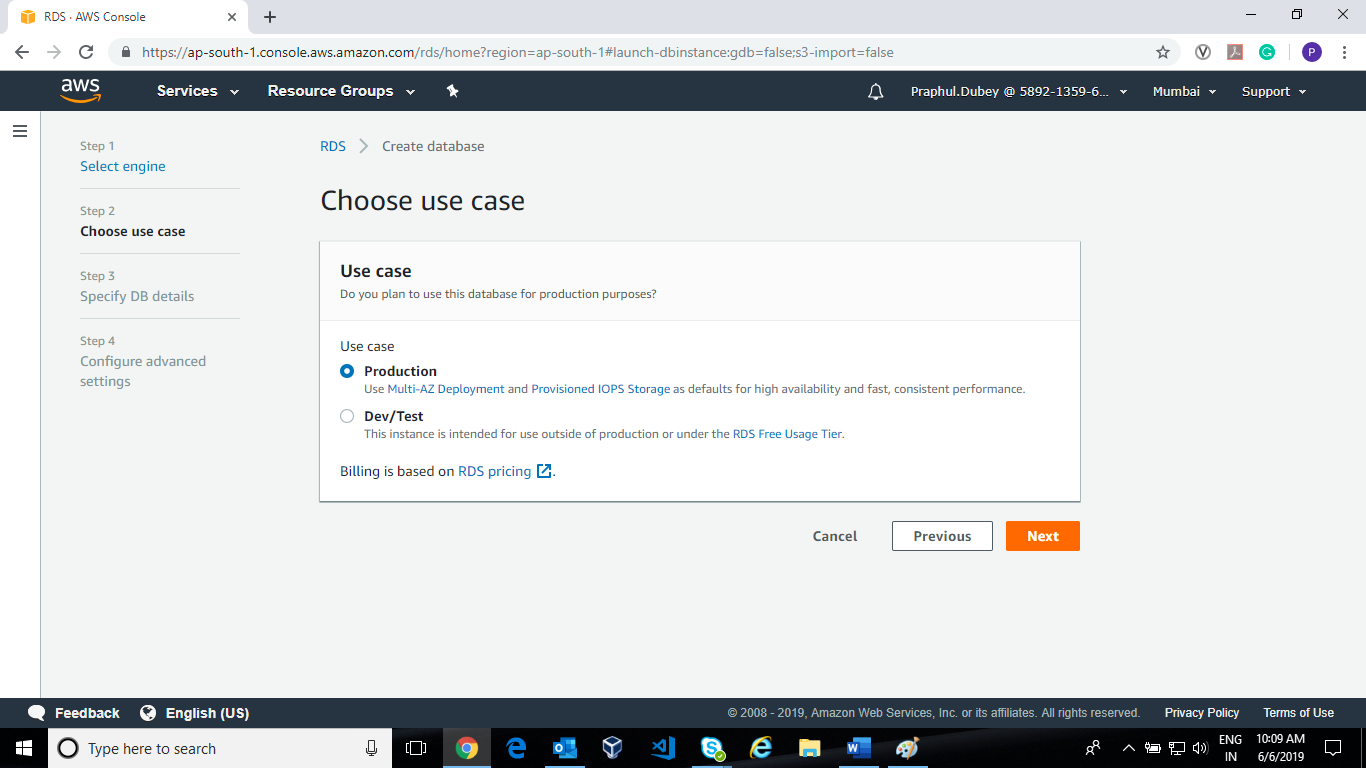
Go to RDS Service and click on create Database



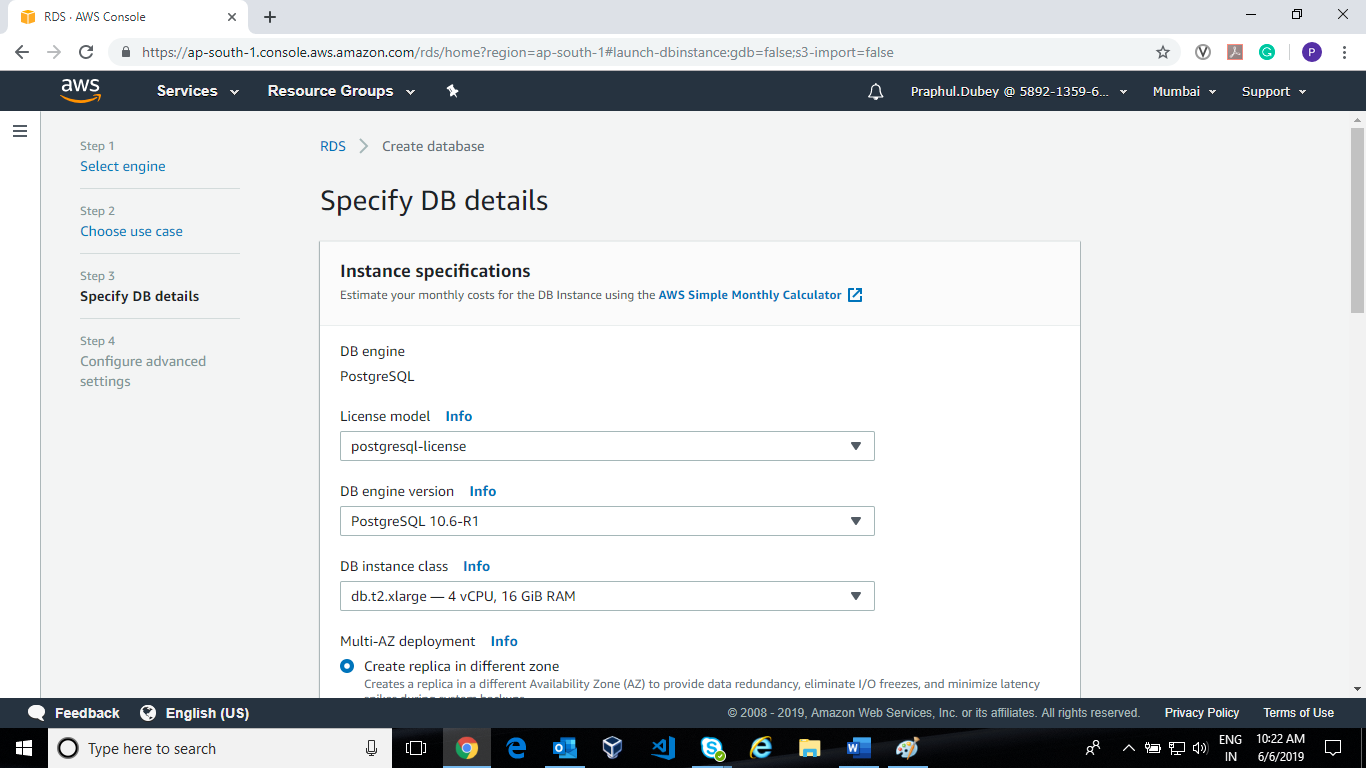
1. Click on **Create database select PostgreSQL** and click Next**.**



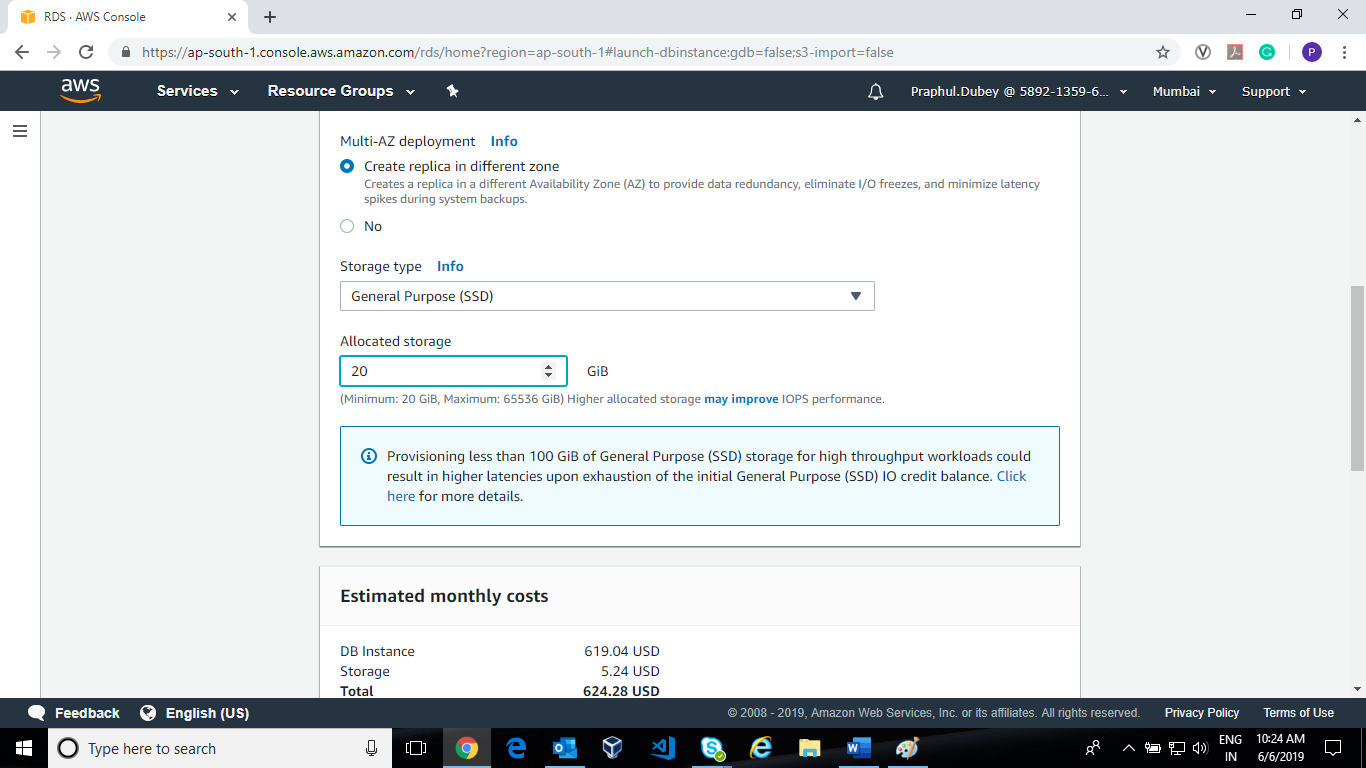
1. Choose use case as **Production and** click next**.**



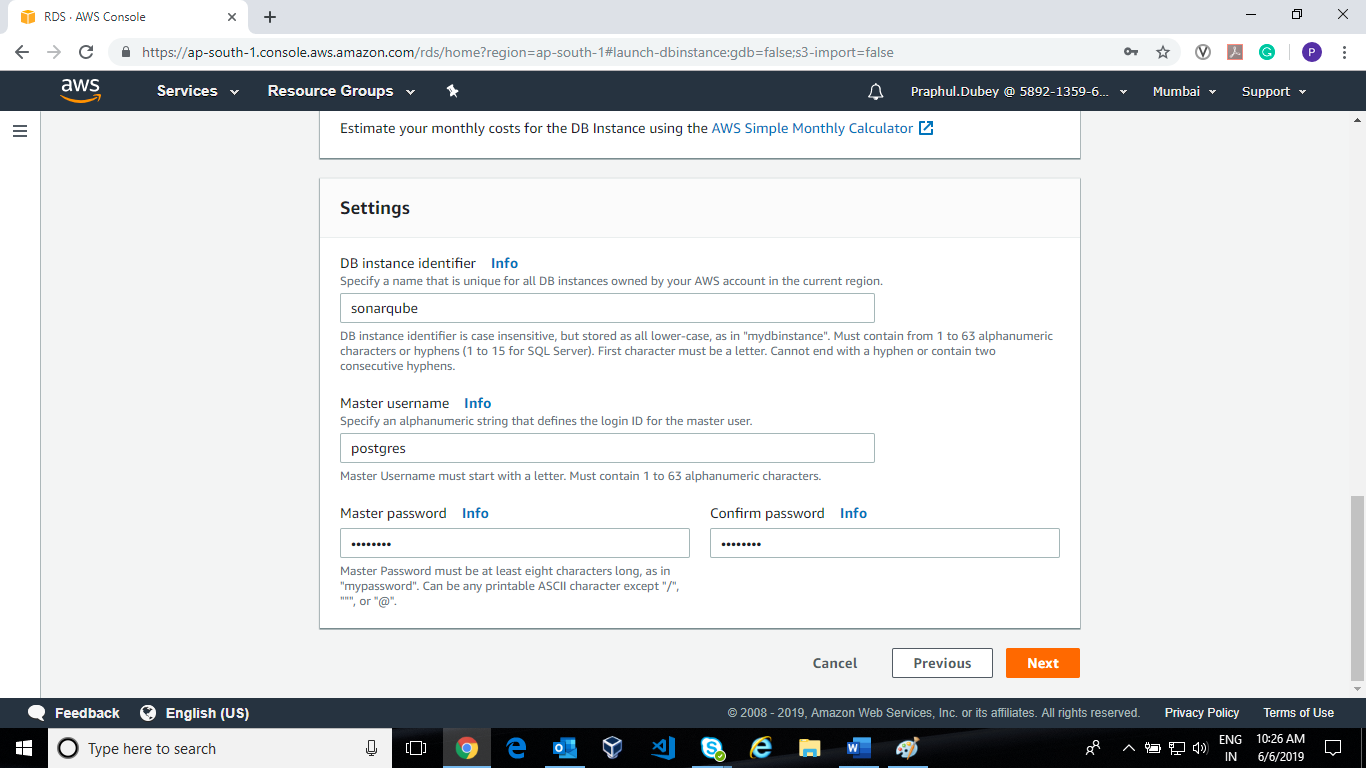
1. Now fill the specific DB details. Choose **the DB instance class** as per your requirement.



Allocated storage as per your requirement by default it is 20GB



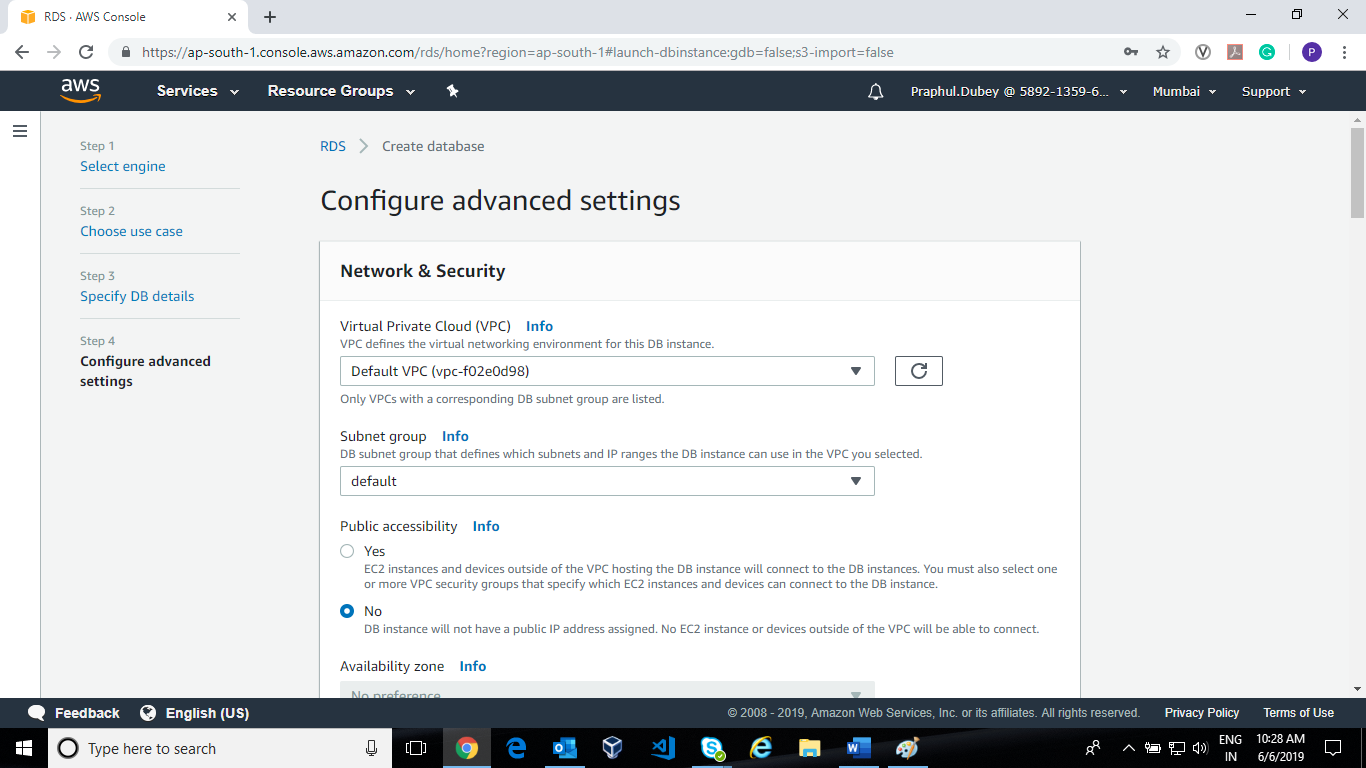
Provide DB Instance identifier and Master Username and Password and click Next.

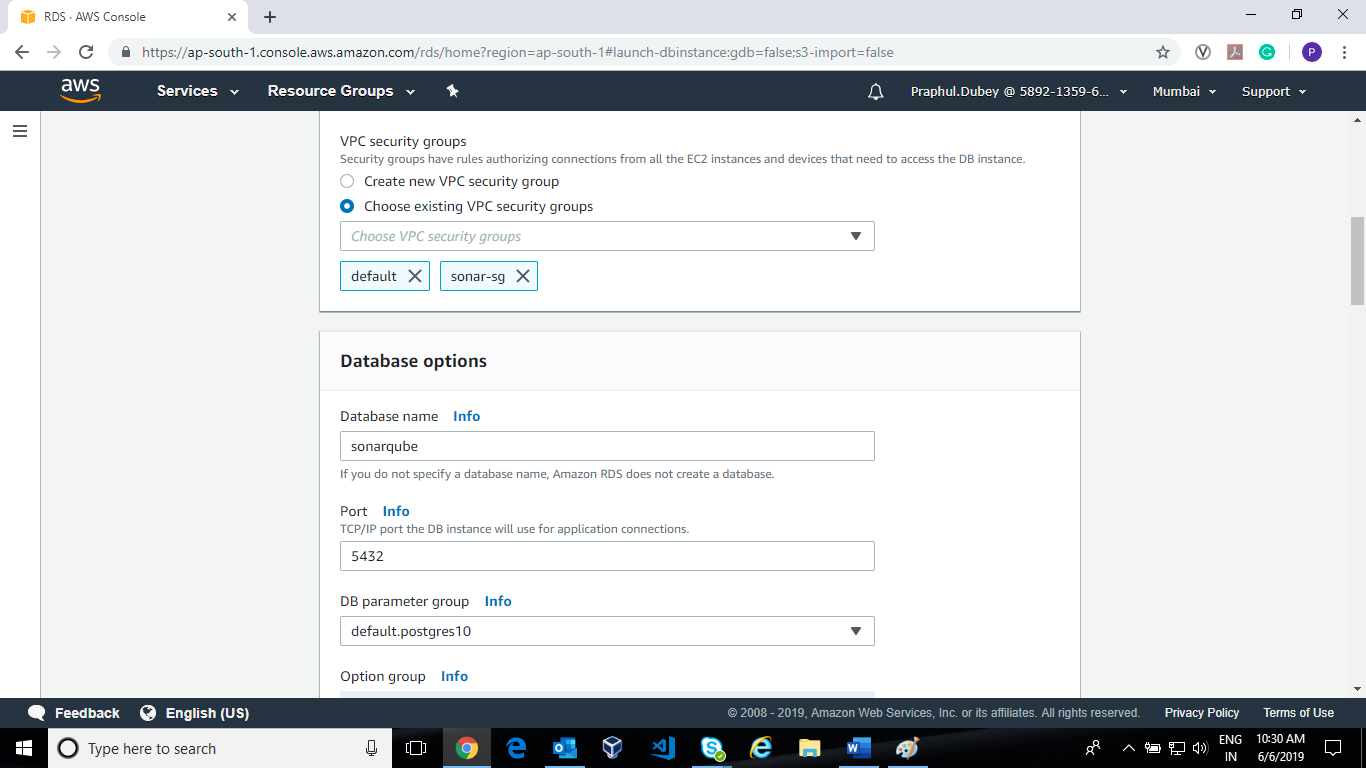


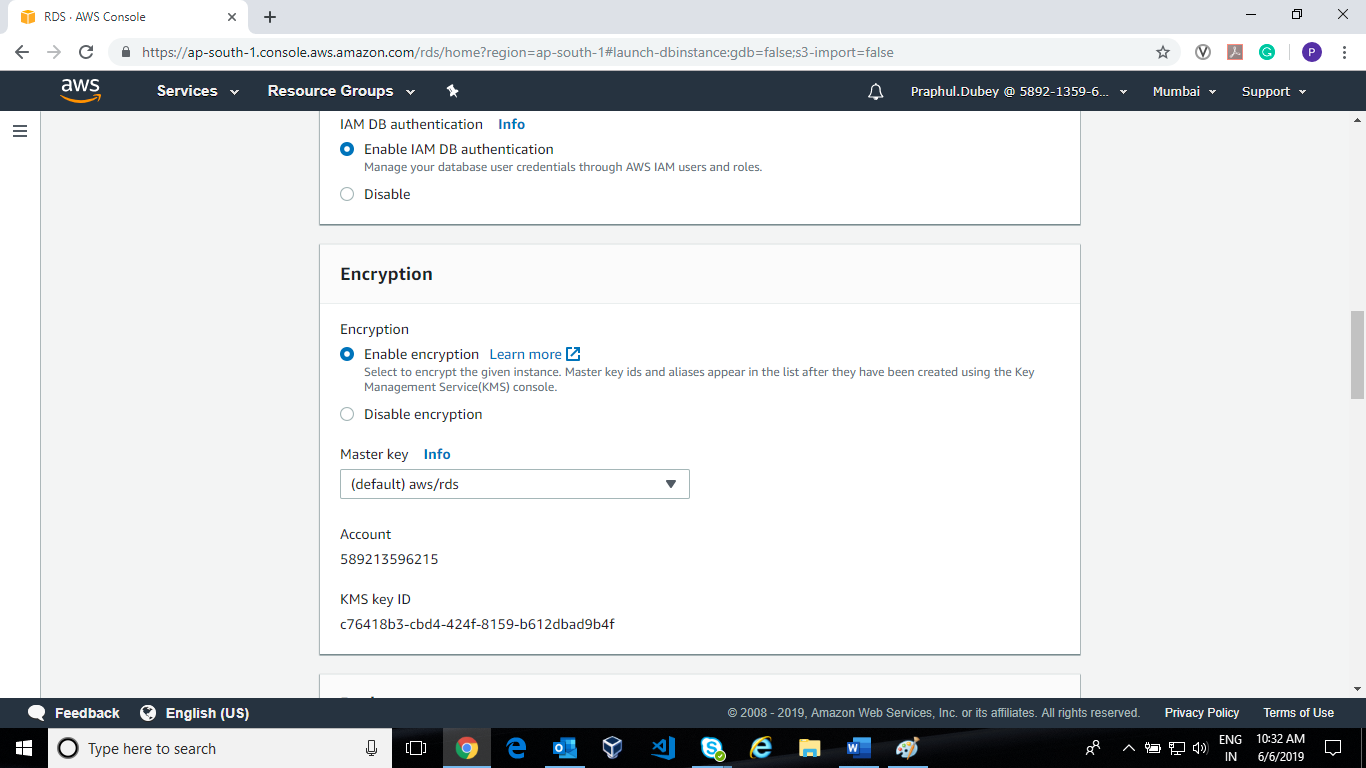
1. Configure Advanced Setting.

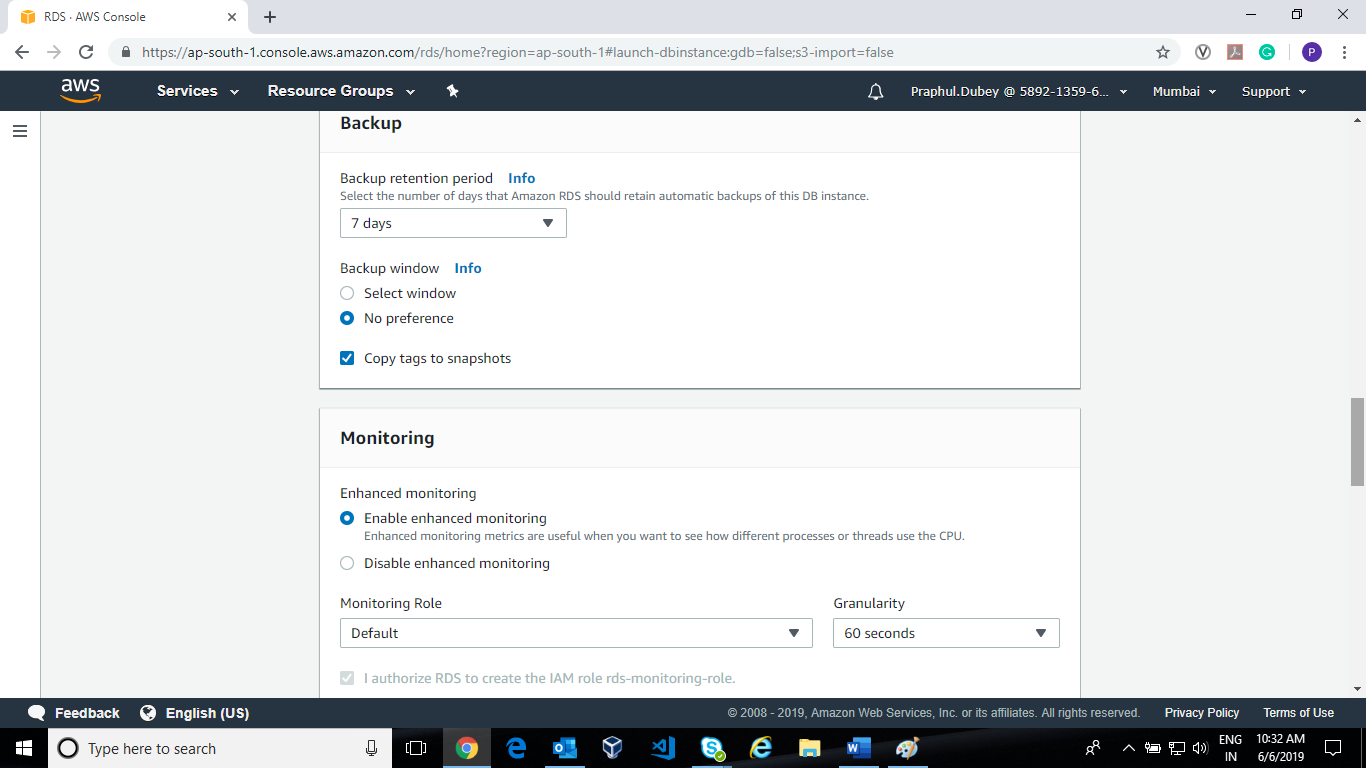
Network & Security

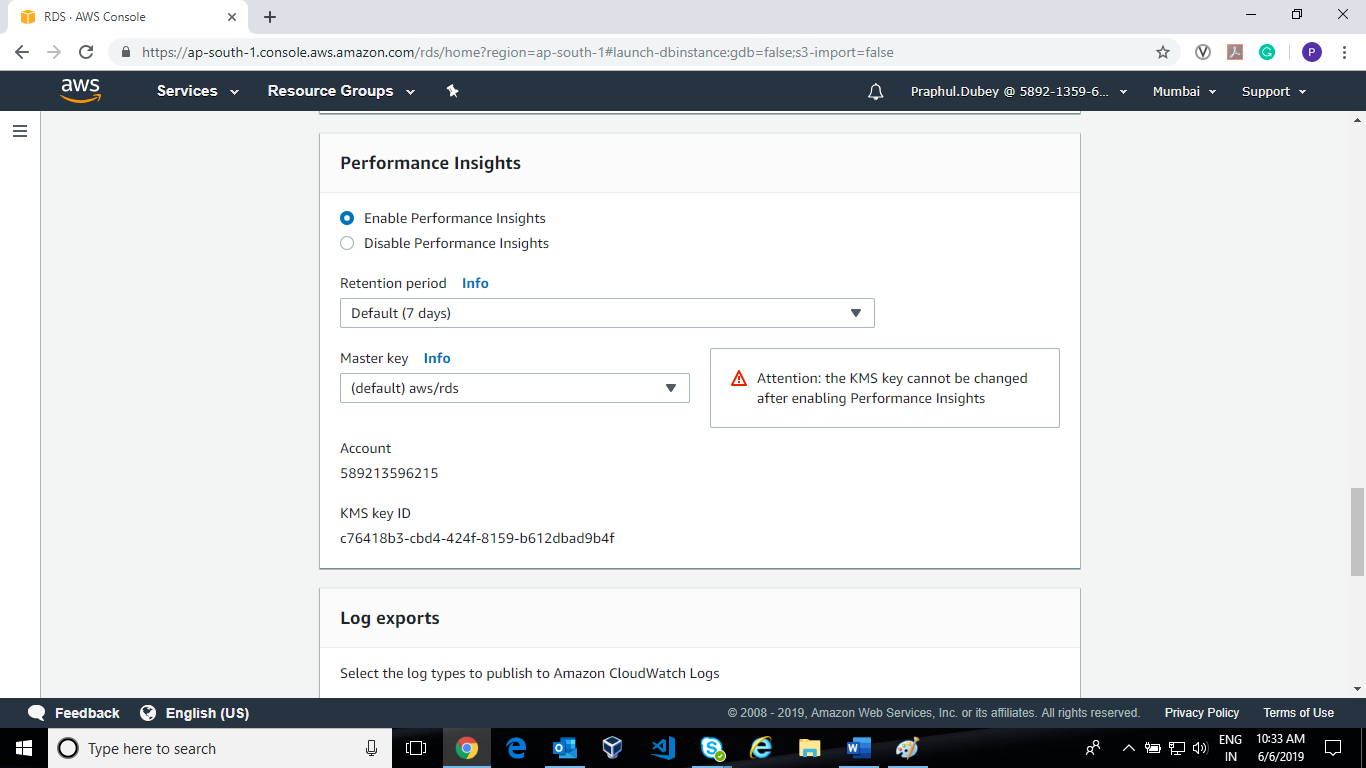
Select VPC subnet Group and Public accessibility **NO**

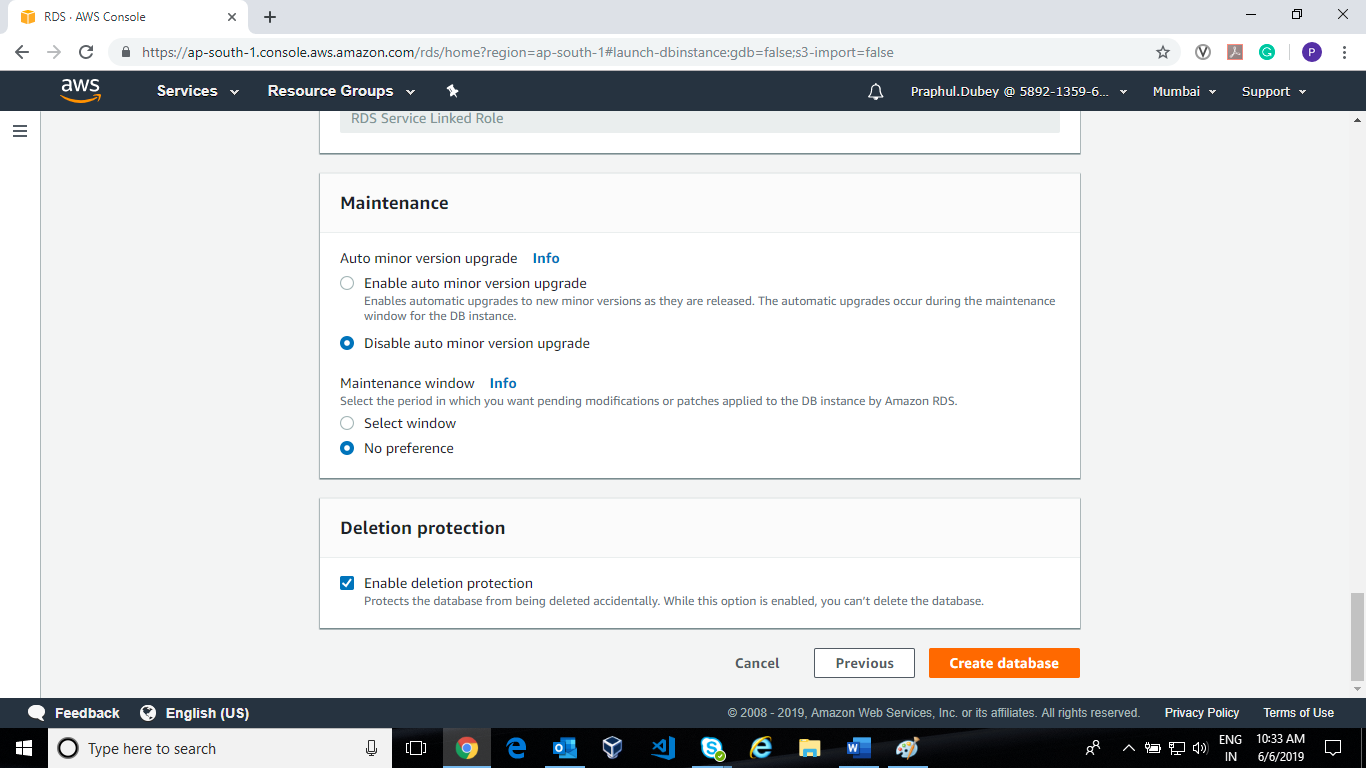








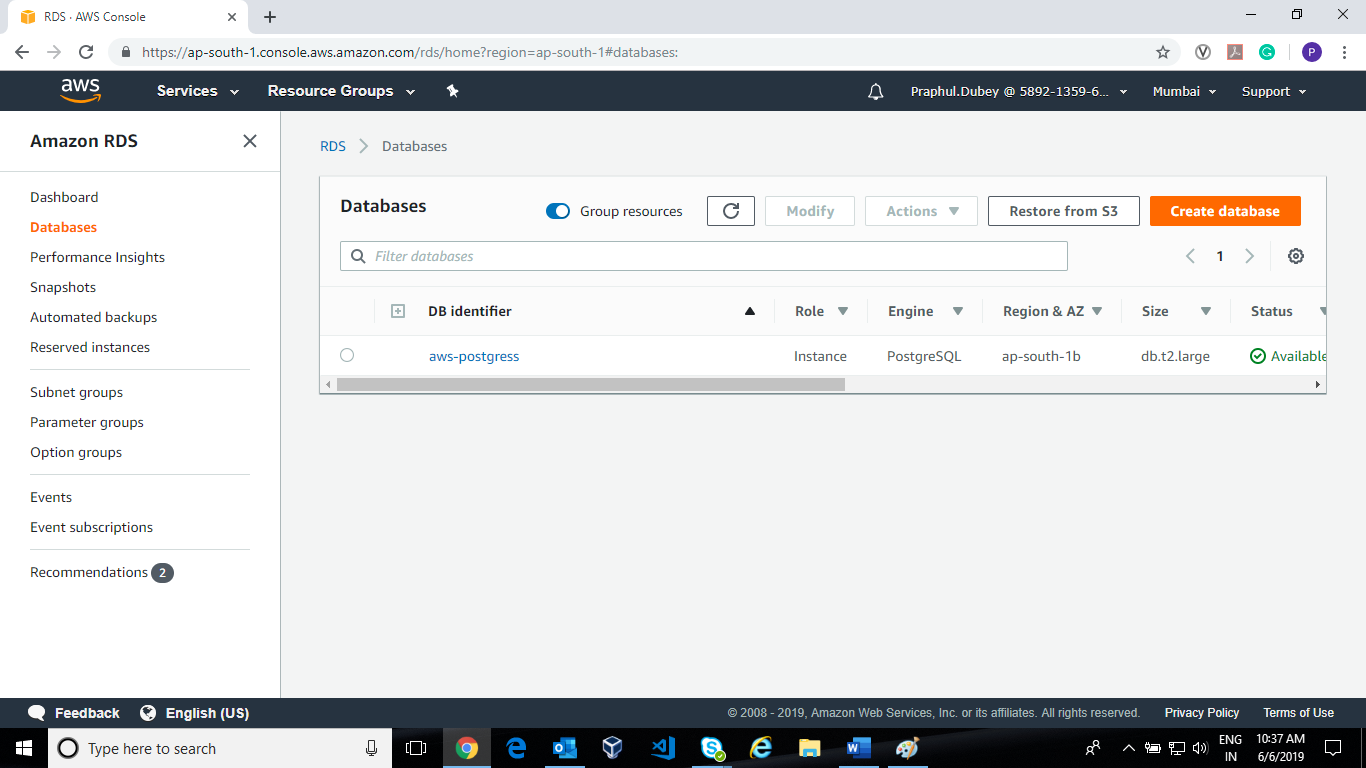




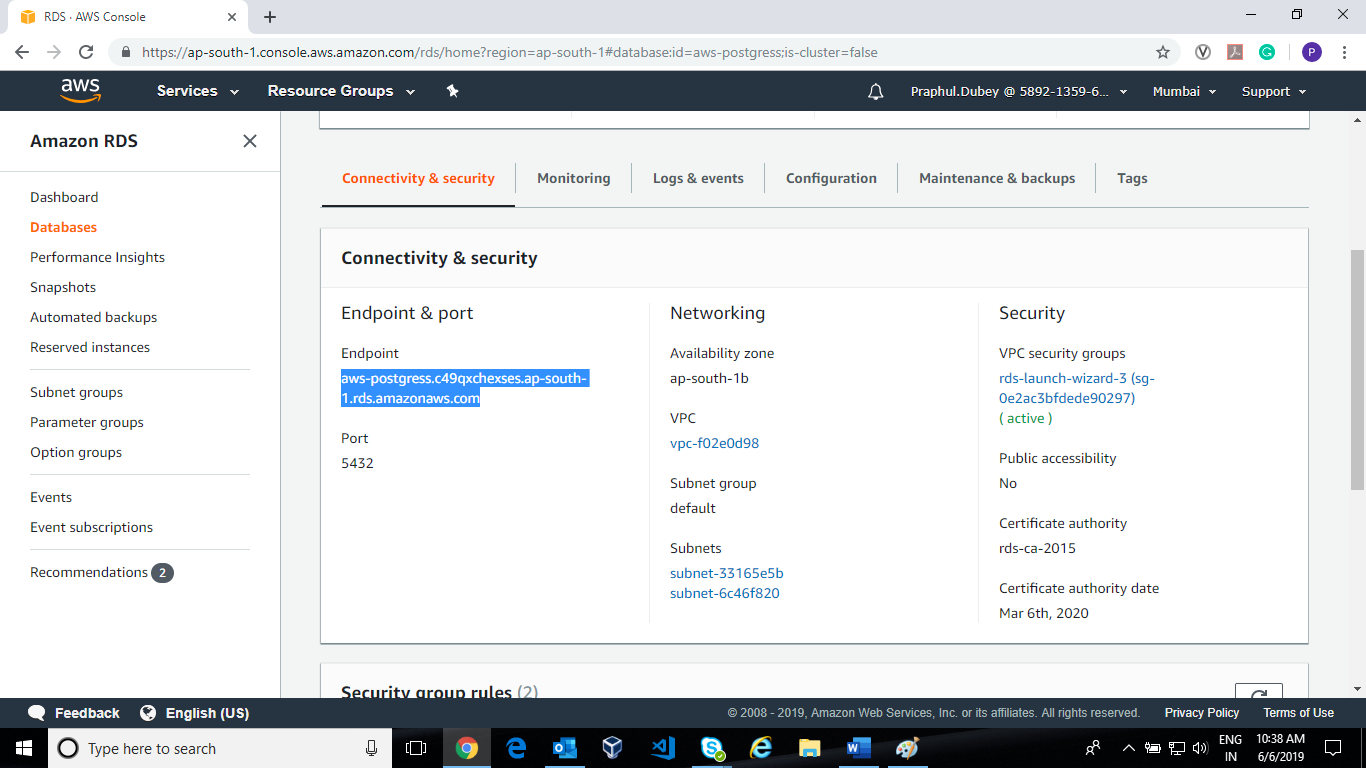
**Then click create database**

**NOTE: Backup, Monitoring,** **Performance Insights,** **Maintenance will vary according to your need.**

**Now the Database is creating and it will take several minutes to up.**



**Please verify the detail and check the Connectivity and Ports and find the Endpoint**



1. Now Create EC2 Instance and as per your requirements I have **selected t2.2xlarge your** instance should be in same VPC as selected in DB creation.

Security Group: port 22, 8080,9000, 5432 443,80 must be open.

1. Install some of the required packages like jdk8, curl, unzip

Install jdk8 or java

sudo yum update

sudo yum install wget curl unzip

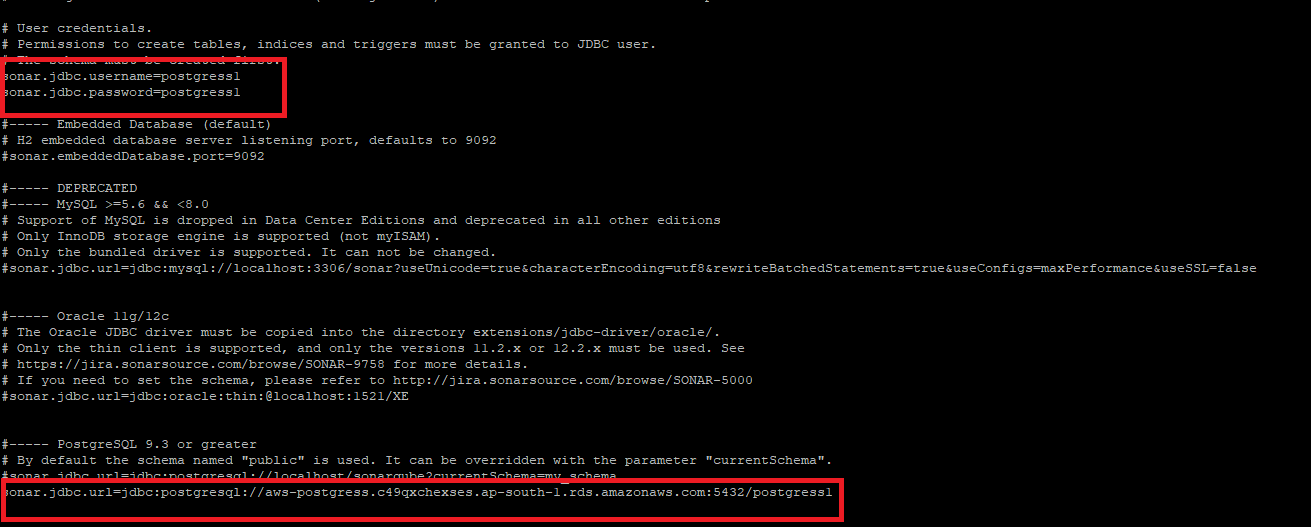
1. Now open your Sonarqube EC2 Instance in Terminal download

wget <https://binaries.sonarsource.com/CommercialDistribution/sonarqube-enterprise/sonarqube-enterprise-7.7.zip>

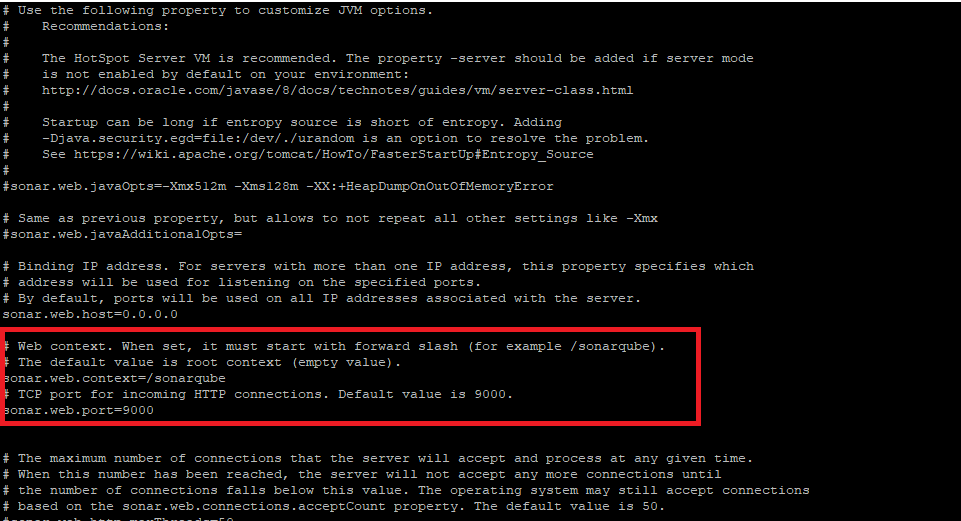
unzip sonarqube-enterprise-7.7.zip

1. Now open sonar.properties file in sonarqube/conf directory

Enter the username and password for you DB instance



Open the port and **sonar.web.host** as 0.0.0.0 as to access from every where



1. Now Start the sonar application

[**ec2-user@ip-172-31-30-245 linux-x86-64]$ pwd**

/**home/ec2-user/sonarqube/bin/linux-x86-64**

**[ec2-user@ip-172-31-30-245 linux-x86-64]$ sh sonar.sh start**

1. Access your Sonar application on

<http://ec2-13-233-68-207.ap-south-1.compute.amazonaws.com:9000/sonarqube>

**Adding SSL Certificate Using Certbot**

Steps for Creating SSL certificate and Reverse Proxy using Nginx

1. Install Nginx on RHEL 7

sudo yum update

sudo vi /etc/yum.repos.d/nginx.repo

add below line in nginx.repo file

**[nginx]**

name**=**nginx repo

baseurl**=**http://nginx.org/packages/mainline/rhel/7/$basearch/

gpgcheck**=**0

enabled**=**1

sudo yum install nginx

sudo systemctl enable nginx

sudo systemctl start nginx

1. Install certbot using below link

<https://certbot.eff.org/lets-encrypt/centosrhel7-nginx>

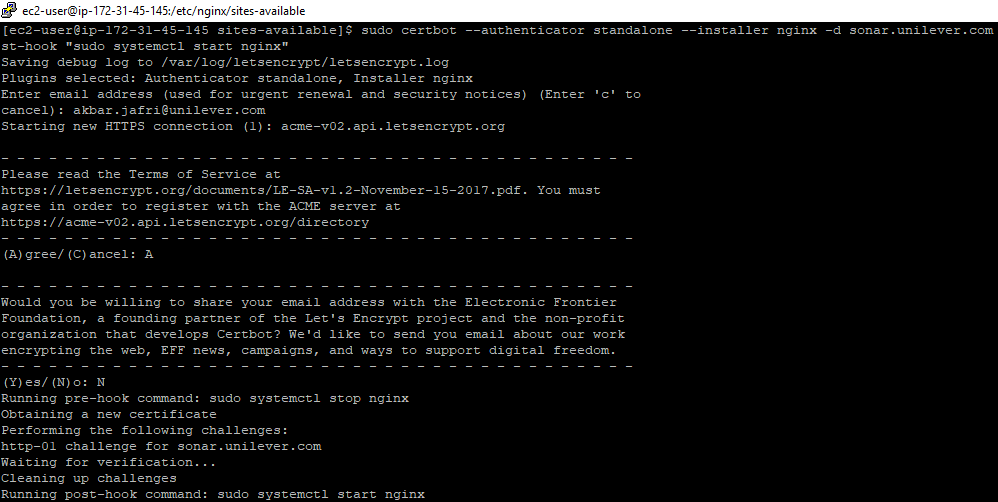
1. Steps to configure reverse proxy
2. sudo systemctl stop nginx
3. cd /etc/nginx/conf.d/ and remove default.conf

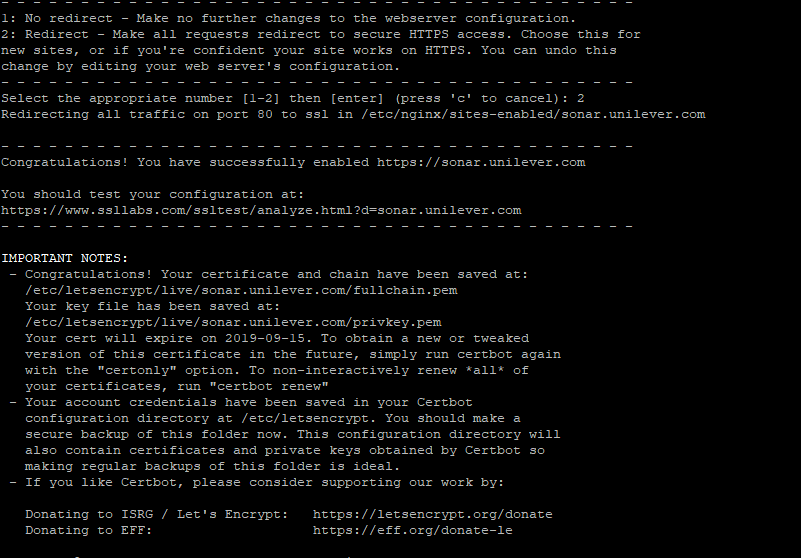
sudo rm default.conf

1. sudo mkdir -p /etc/nginx/sites-avaliable/
2. sudo mkdir -p /etc/nginx/sites-enabled/
3. cd /etc/nginx/sites-avaliable/
4. touch “domain-name”

Example: sudo touch sonar-dev.unilever.com<**domain-nam**e>

1. sudo ln -fs /etc/nginx/sites-available/sonar-dev.unilever.com /etc/nginx/sites-enabled/sonar-dev.unilever.com
2. sudo certbot --authenticator standalone --installer nginx -d **sonar-dev.unilever.com** --pre-hook "sudo systemctl stop nginx" --post-hook "sudo systemctl start nginx"





1. sudo vi sonar-dev.unilever.com

Add the following line with change in **<domain name>** for eg here “**sonar-dev.unilever.com”**

**server {**

**listen 443 ssl; # managed by Certbot**

**root /usr/share/nginx/html/;**

**index index.html index.htm;**

**server\_name sonar-dev.unilever.com;**

**location / {**

**try\_files $uri @proxy;**

**}**

**ssl\_certificate /etc/letsencrypt/live/sonar-dev.unilever.com/fullchain.pem; # managed by Certbot**

**ssl\_certificate\_key /etc/letsencrypt/live/sonar-dev.unilever.com/privkey.pem; # managed by Certbot**

**include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot**

**ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot**

**location @proxy {**

**proxy\_pass http://<172.31.46.242:9000>; //sonar private ip**

**proxy\_pass\_header Server;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection "upgrade";**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

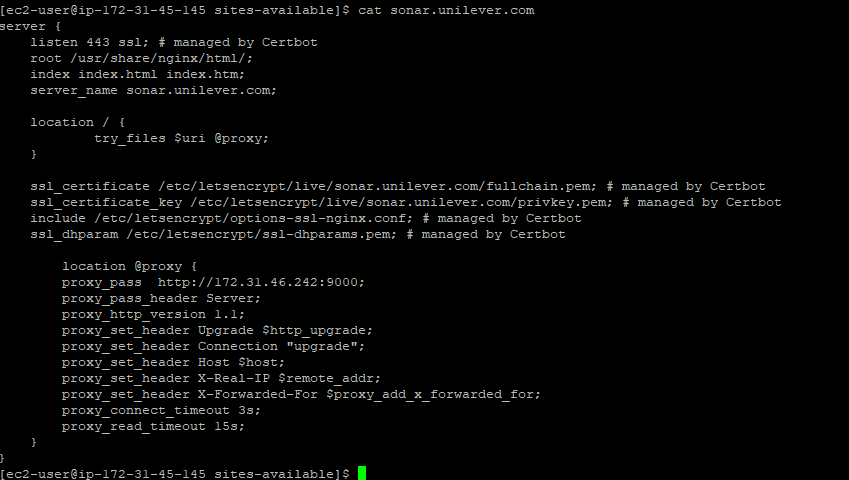
**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

**proxy\_connect\_timeout 3s;**

**proxy\_read\_timeout 15s;**

**}**

**}**



1. sudo vi nginx.conf

add the below line in nginx.conf file

include /etc/nginx/sites-enabled/\*;

1. sudo systemctl restart nginx