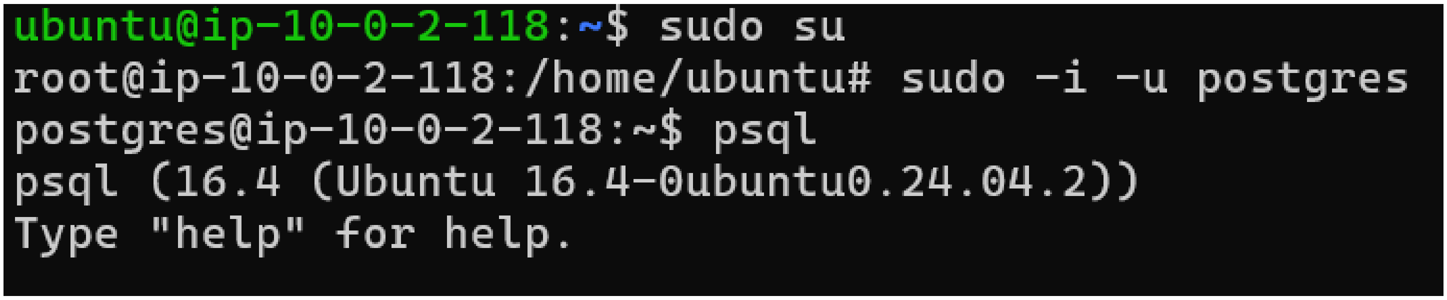
PostgreSQL Database server

sudo apt update

sudo apt install postgresql postgresql-contrib -y

sudo su

sudo -i -u postgres



```
ubuntu@ip-10-0-2-118:~$ sudo su
root@ip-10-0-2-118:/home/ubuntu# sudo -i -u postgres
postgres@ip-10-0-2-118:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.
```

psql

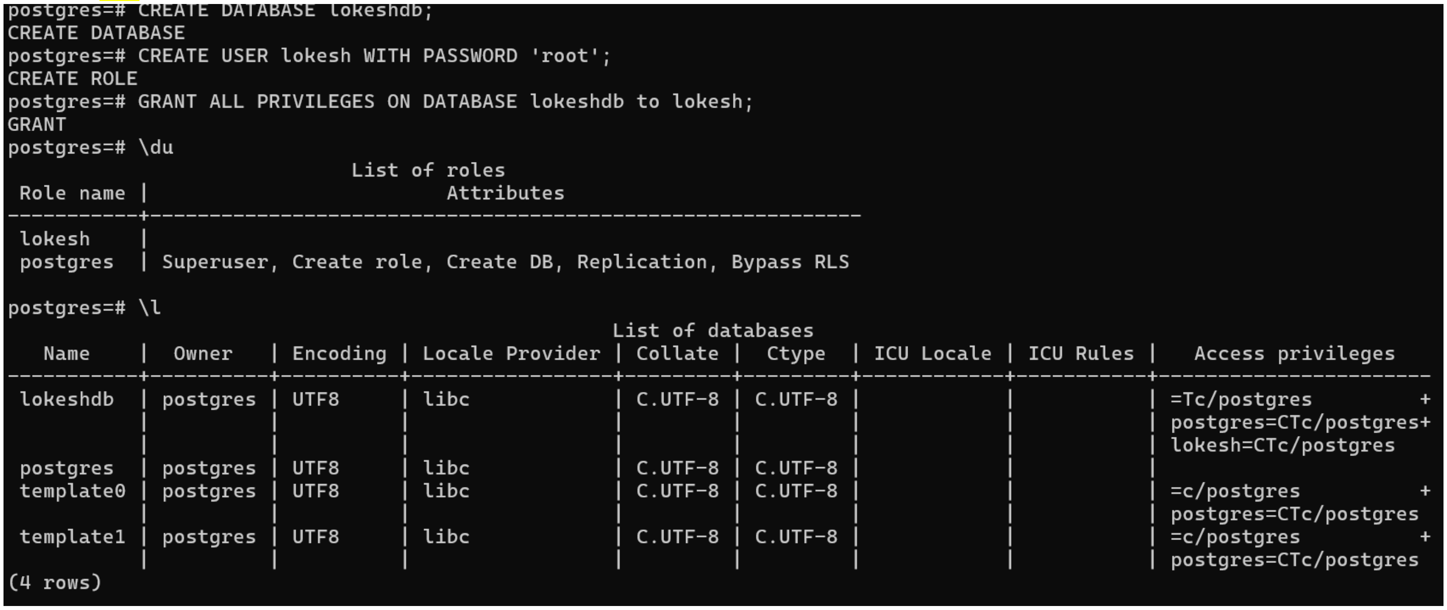
CREATE DATABASE lokeshdb;

CREATE USER lokesh WITH PASSWORD 'root';

GRANT ALL PRIVILEGES ON DATABASE lokeshdb to lokesh;

\du ---- checks user

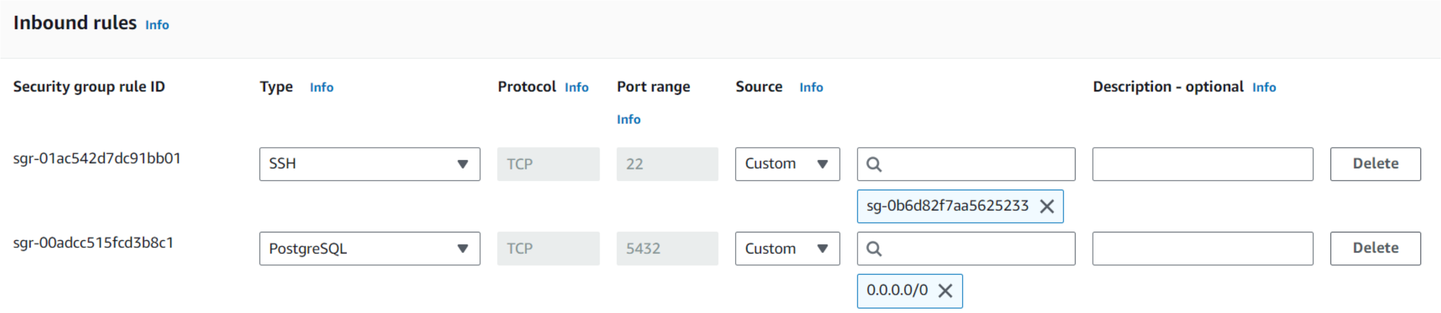
\l ---list



```
postgres=# CREATE DATABASE lokeshdb;
CREATE DATABASE
postgres=# CREATE USER lokesh WITH PASSWORD 'root';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE lokeshdb to lokesh;
GRANT
postgres=# \du
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 lokesh    |
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS

postgres=# \l
                                                  List of databases
   Name    |  Owner   | Encoding | Locale Provider | Collate |  Ctype  | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+---------+---------+------------+-----------+-----------------------
 lokeshdb  | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           | =Tc/postgres         +
           |          |          |                 |         |         |            |           | postgres=CTc/postgres+
           |          |          |                 |         |         |            |           | lokesh=CTc/postgres
 postgres  | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |         |         |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |         |         |            |           | postgres=CTc/postgres
(4 rows)
```

Make sure you have configures security groups { postgresql – 5432 - anywhere}

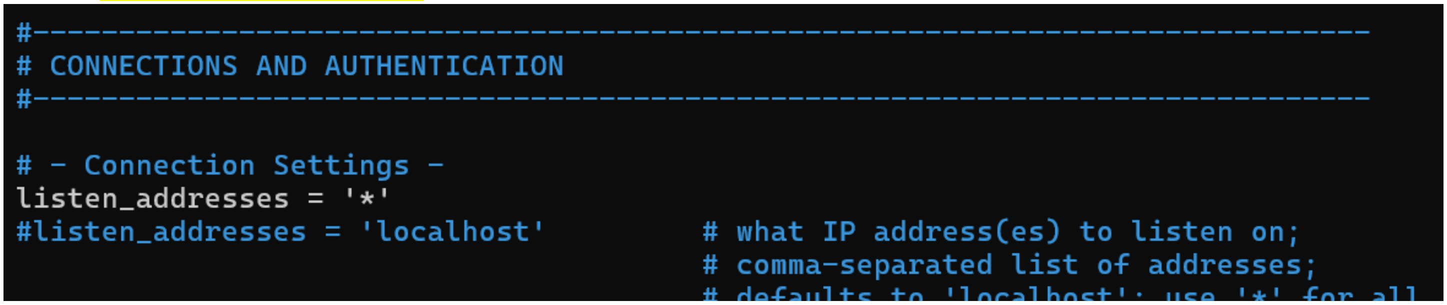


Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-01ac542d7dc91bb01	SSH ▼	TCP	22	Custom ▼	sg-0b6d82f7aa5625233 ✕		Delete
sgr-00adcc515fcd3b8c1	PostgreSQL ▼	TCP	5432	Custom ▼	0.0.0.0/0 ✕		Delete

sudo nano /etc/postgresql/”version no”/main/postgresql.conf

listen_addresses = ’*’

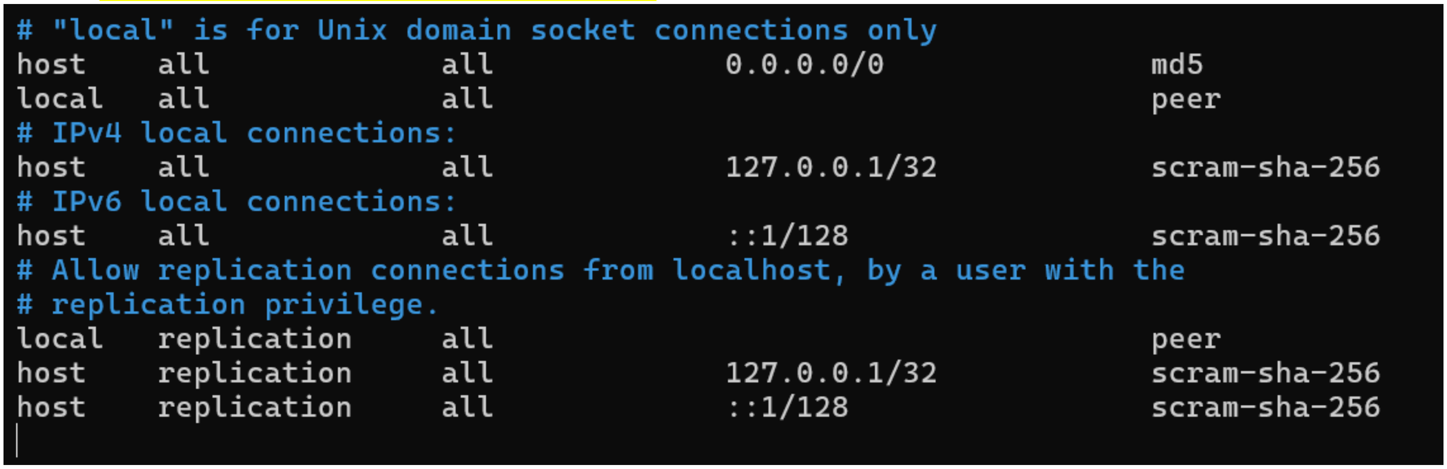


```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -
listen_addresses = '*'
#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
```

sudo nano /etc/postgresql/”version no”/main/pg_hba.conf

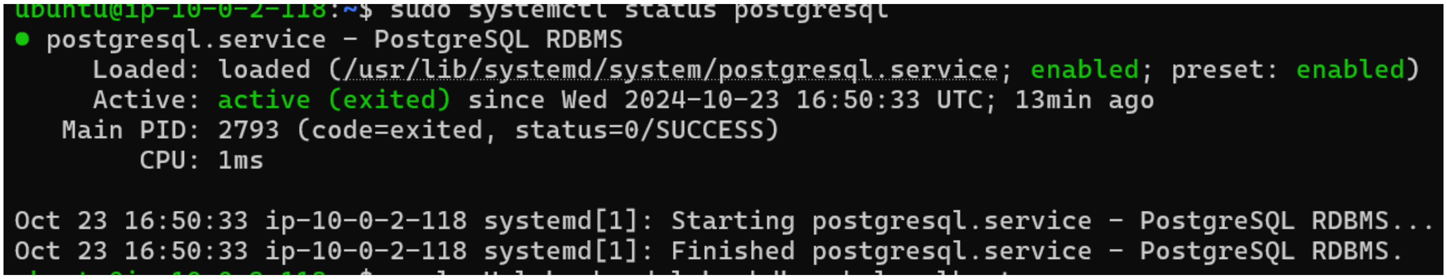
host   all   all   0.0.0.0/0   md5



```
# "local" is for Unix domain socket connections only
host    all             all             0.0.0.0/0               md5
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            scram-sha-256
# IPv6 local connections:
host    all             all             ::1/128                 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            scram-sha-256
host    replication     all             ::1/128                 scram-sha-256
```

sudo systemctl enable postgresql

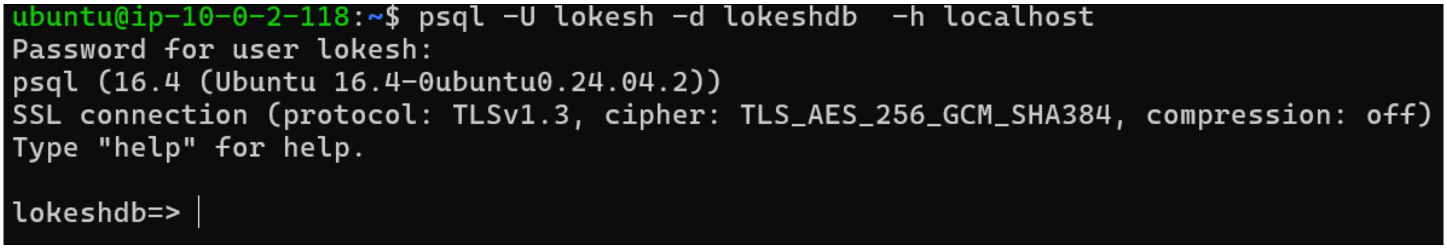
sudo systemctl status postgresql



```
ubuntu@ip-10-0-2-118:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Wed 2024-10-23 16:50:33 UTC; 13min ago
   Main PID: 2793 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Oct 23 16:50:33 ip-10-0-2-118 systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Oct 23 16:50:33 ip-10-0-2-118 systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
```

psql -U lokesh -d lokeshdb -h localhost



```
ubuntu@ip-10-0-2-118:~$ psql -U lokesh -d lokeshdb  -h localhost
Password for user lokesh:
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

lokeshdb=>
```

set permissions for posgres – set password

sudo -u postgres psql

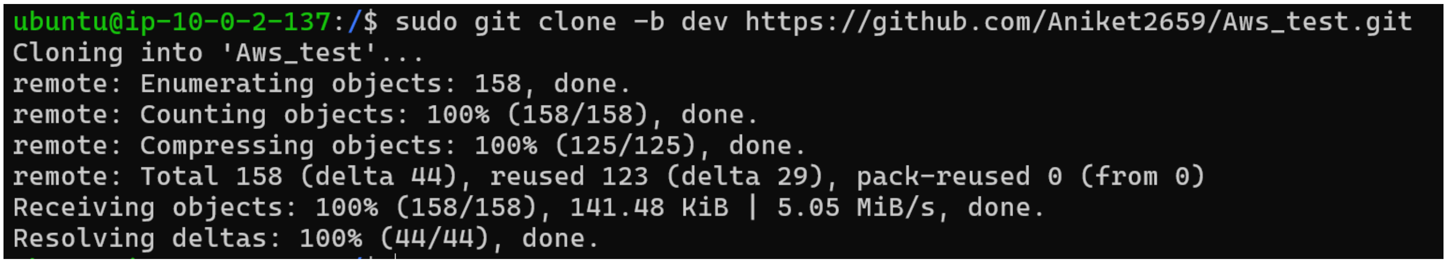
psql -U postgres -h <host> -p <port>;

ALTER USER postgres PASSWORD 'new_password';

\q

Backend Server -Django Application

git clone <link>



```
ubuntu@ip-10-0-2-137:/$ sudo git clone -b dev https://github.com/Aniket2659/Aws_test.git
Cloning into 'Aws_test'...
remote: Enumerating objects: 158, done.
remote: Counting objects: 100% (158/158), done.
remote: Compressing objects: 100% (125/125), done.
remote: Total 158 (delta 44), reused 123 (delta 29), pack-reused 0 (from 0)
Receiving objects: 100% (158/158), 141.48 KiB | 5.05 MiB/s, done.
Resolving deltas: 100% (44/44), done.
```

cd Aws_test/

sudo apt update && sudo apt upgrade -y

sudo apt install python3 python3-pip python3-venv -y

sudo apt install libpq-dev -y

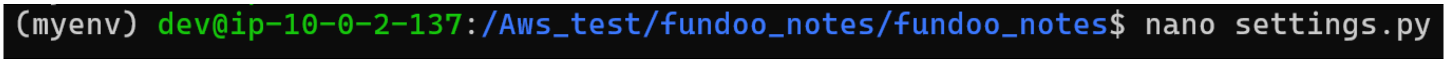
python3 -m venv myenv

source myenv/bin/activate

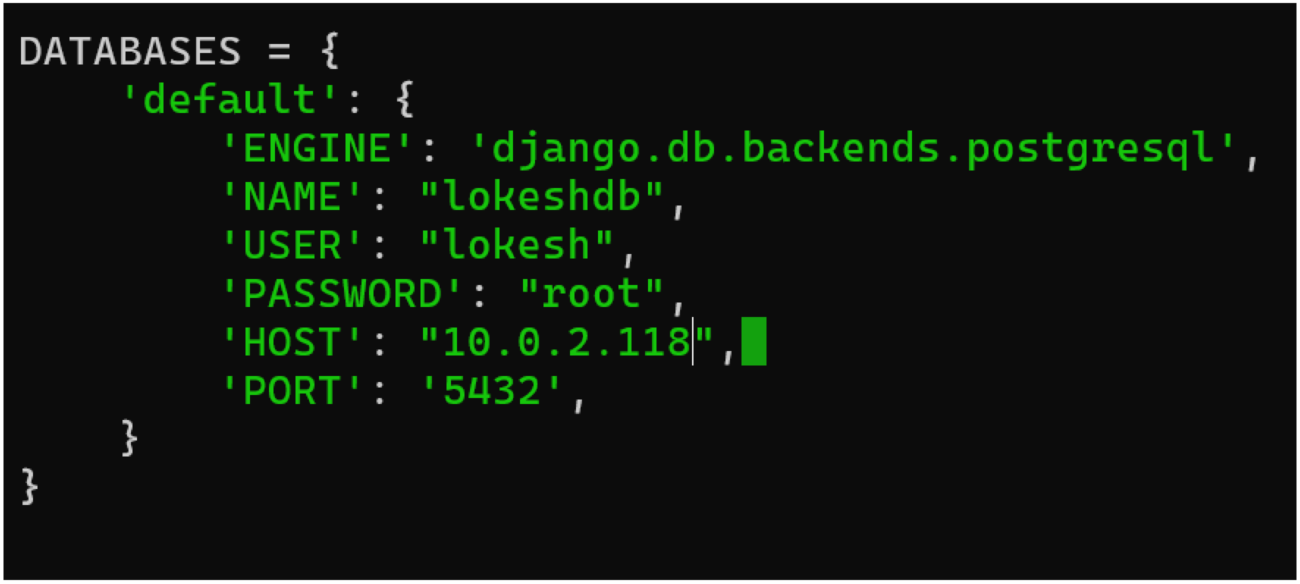
pip install django gunicorn

pip install -r requirements.txt

nano settings.py



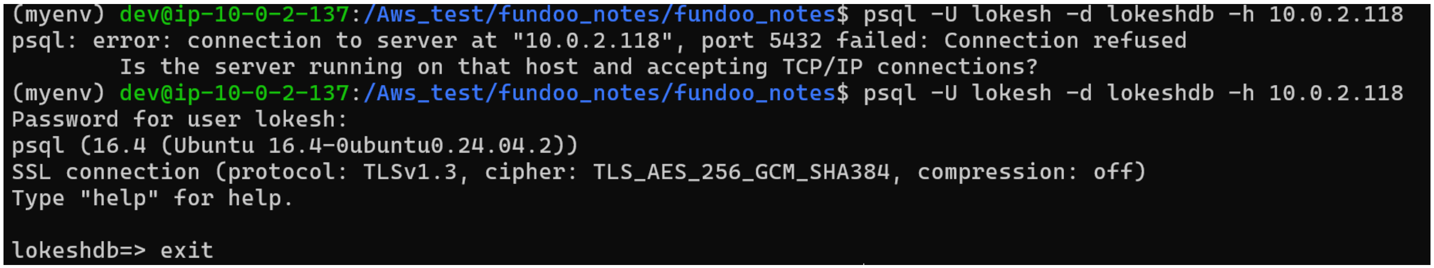
```
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes/fundoo_notes$ nano settings.py
```



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': "lokeshdb",
        'USER': "lokesh",
        'PASSWORD': "root",
        'HOST': "10.0.2.118",
        'PORT': '5432',
    }
}
```

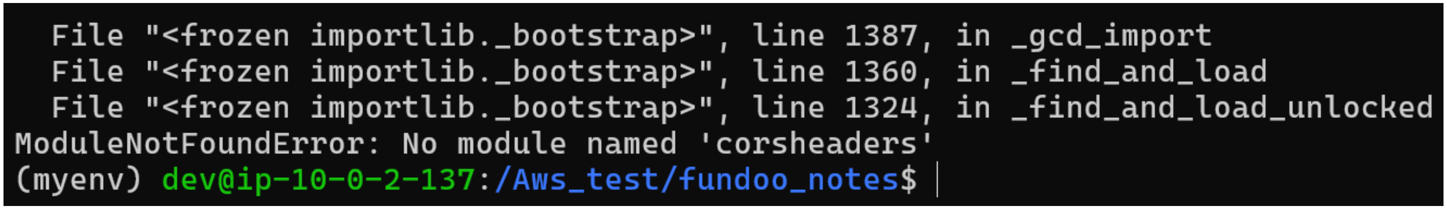
sudo apt install postgresql-client

psql -U lokesh -d lokeshdb -h 10.0.2.118 => in case error make sure security groups has posgresql port



```
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes/fundoo_notes$ psql -U lokesh -d lokeshdb -h 10.0.2.118
psql: error: connection to server at "10.0.2.118", port 5432 failed: Connection refused
        Is the server running on that host and accepting TCP/IP connections?
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes/fundoo_notes$ psql -U lokesh -d lokeshdb -h 10.0.2.118
Password for user lokesh:
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

lokeshdb=> exit
```



```
  File "<frozen importlib._bootstrap>", line 1387, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1360, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1324, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'corsheaders'
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes$
```

pip install django-cors-headers

psql -U postgres -h 10.0.2.118

```
GRANT ALL PRIVILEGES ON SCHEMA public TO lokesh;
GRANT ALL PRIVILEGES ON DATABASE lokeshdb TO lokesh;
\q
```

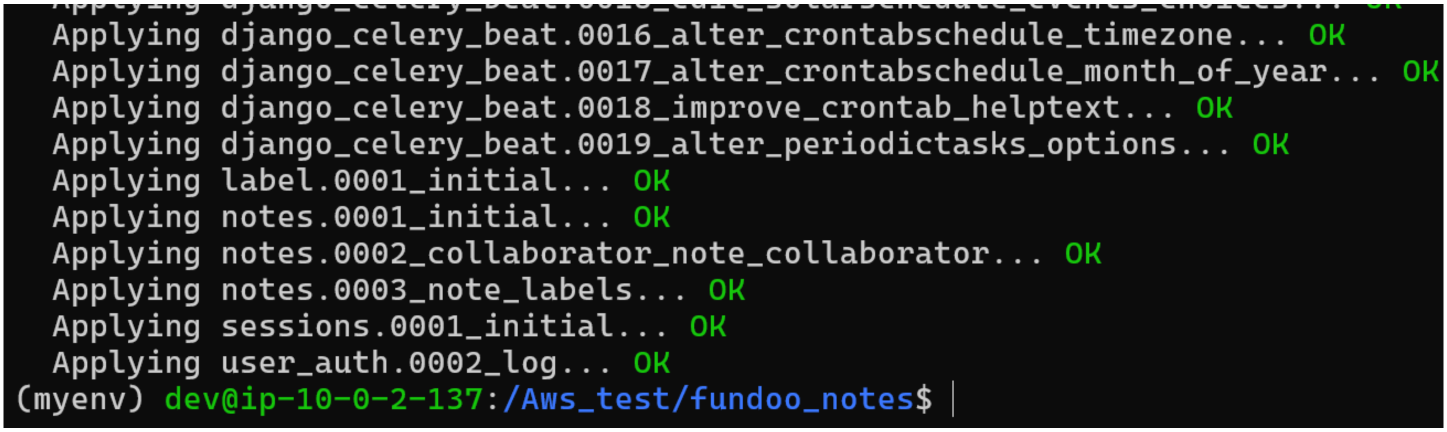
python3 manage.py migrate

if it doesn’t work try alternate give additional permissions

psql -U postgres -h 10.0.2.118

```
ALTER SCHEMA public OWNER TO lokesh;
GRANT ALL ON SCHEMA public TO lokesh;
GRANT USAGE, CREATE ON SCHEMA public TO lokesh;
\du lokesh
ALTER USER lokesh WITH SUPERUSER;
\q
```

python3 manage.py migrate – it should work

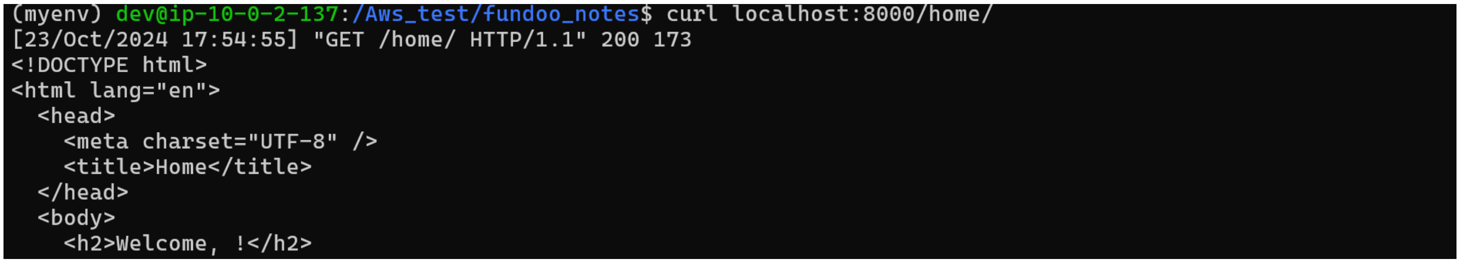


```
  Applying django_celery_beat.0016_alter_crontabschedule_timezone... OK
  Applying django_celery_beat.0017_alter_crontabschedule_month_of_year... OK
  Applying django_celery_beat.0018_improve_crontab_helptext... OK
  Applying django_celery_beat.0019_alter_periodictasks_options... OK
  Applying label.0001_initial... OK
  Applying notes.0001_initial... OK
  Applying notes.0002_collaborator_note_collaborator... OK
  Applying notes.0003_note_labels... OK
  Applying sessions.0001_initial... OK
  Applying user_auth.0002_log... OK
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes$
```

python manage.py runserver 0.0.0.0:8000 -> run this in the backend

ctrl + Z >> bg 1 >> it will run in the backend

curl localhost:8000/home/



```
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes$ curl localhost:8000/home/
[23/Oct/2024 17:54:55] "GET /home/ HTTP/1.1" 200 173
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Home</title>
  </head>
  <body>
    <h2>Welcome, !</h2>
```

For setting the environment variables

sudo nano /etc/bash.bashrc

```
        export FUNDOODB='lokeshdb'
        export FUNDOOUSER='lokesh'
        export FUNDOODBPASSWORD='root'
        export FUNDOODBHOST='10.0.2.118'
        source /etc/bash.bashrc
```

alternatively u can create a file

sudo nano /etc/fundoo/env.conf

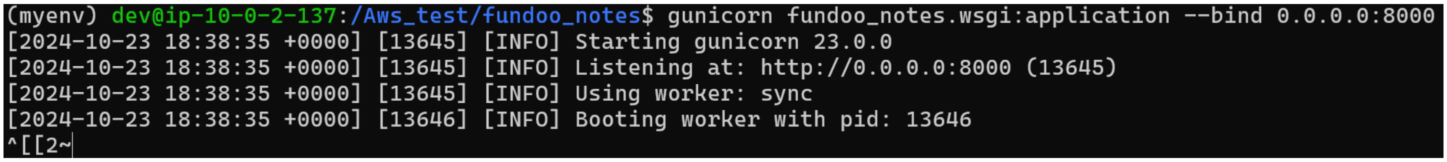
```
        export FUNDOODB='lokeshdb'
        export FUNDOOUSER='lokesh'
        export FUNDOODBPASSWORD='root'
        export FUNDOODBHOST='10.0.2.118'
        source /etc/fundoo/env.conf
```

python manage.py runserver 0.0.0.0:8000 -> run this in the backend

ctrl + Z >> bg 1 >> it will run in the backend

curl localhost:8000/home/

gunicorn fundoo_notes.wsgi:application --bind 0.0.0.0:8000



```
(myenv) dev@ip-10-0-2-137:/Aws_test/fundoo_notes$ gunicorn fundoo_notes.wsgi:application --bind 0.0.0.0:8000
[2024-10-23 18:38:35 +0000] [13645] [INFO] Starting gunicorn 23.0.0
[2024-10-23 18:38:35 +0000] [13645] [INFO] Listening at: http://0.0.0.0:8000 (13645)
[2024-10-23 18:38:35 +0000] [13645] [INFO] Using worker: sync
[2024-10-23 18:38:35 +0000] [13646] [INFO] Booting worker with pid: 13646
^[[2~
```

Creating a .service file to run indefinetly

sudo vim /etc/systemd/system/fundoo.service

```
        [Unit]
        Description=Chatapp Service
        After=network.target
        [Service]
        User=fundoo
        Group=fundoo
        EnvironmentFile=/etc/fundoo/env.conf
        WorkingDirectory=/Aws_test/fundoo_notes
        ExecStart=/bin/bash -c "cd /Aws_test && source venv/bin/activate && cd
        /Aws_test/fundoo_notes && /app/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:8000
        fundoo_notes.wsgi:application"

        [Install]
         WantedBy=multi-user.target
sudo systemctl daemon-reload
sudo systemctl enable fundoo.service
sudo systemctl start fundoo.service
netstat -taupne | grep LIST   => test application
curl http://localhost:8000
```

Frontend- server configuration

```
sudo apt-get update && apt-get install nginx
sudo vim /etc/nginx/sites-available/fundoo.conf
        server {
                        listen 80;
                        server_name _default;
                        location / {
                                include proxy_params;
                                proxy_pass http://10.0.2.137:8000;
                        }
                }
cd /etc/nginx/sites-enabled/
sudo unlink default
sudo ln -s /etc/nginx/sites-available/fundoo.conf fundoo.conf
sudo systemctl restart nginx
sudo systemctl enable nginx
curl http://10.0.2.137:8000/home/
```