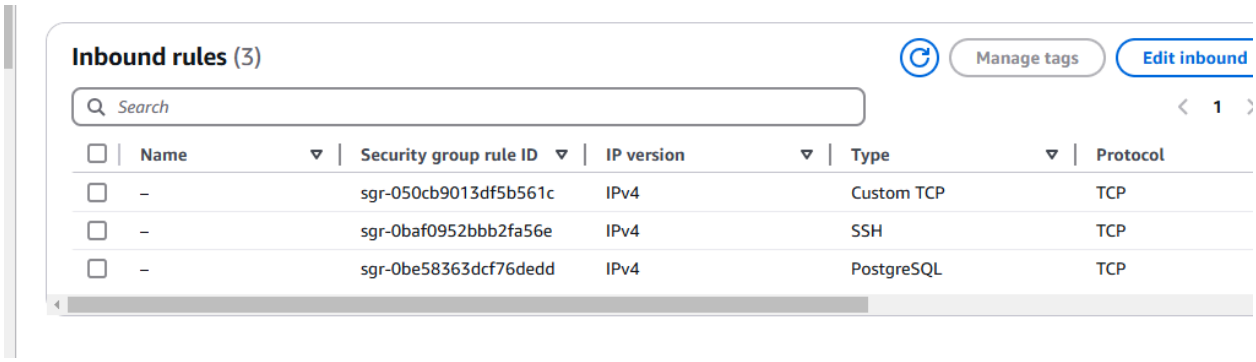
**Deploy Fundo_notes application in AWS:**

Create an EC2 instance and add security group which contain below inbound rules



**Inbound rules** (3)

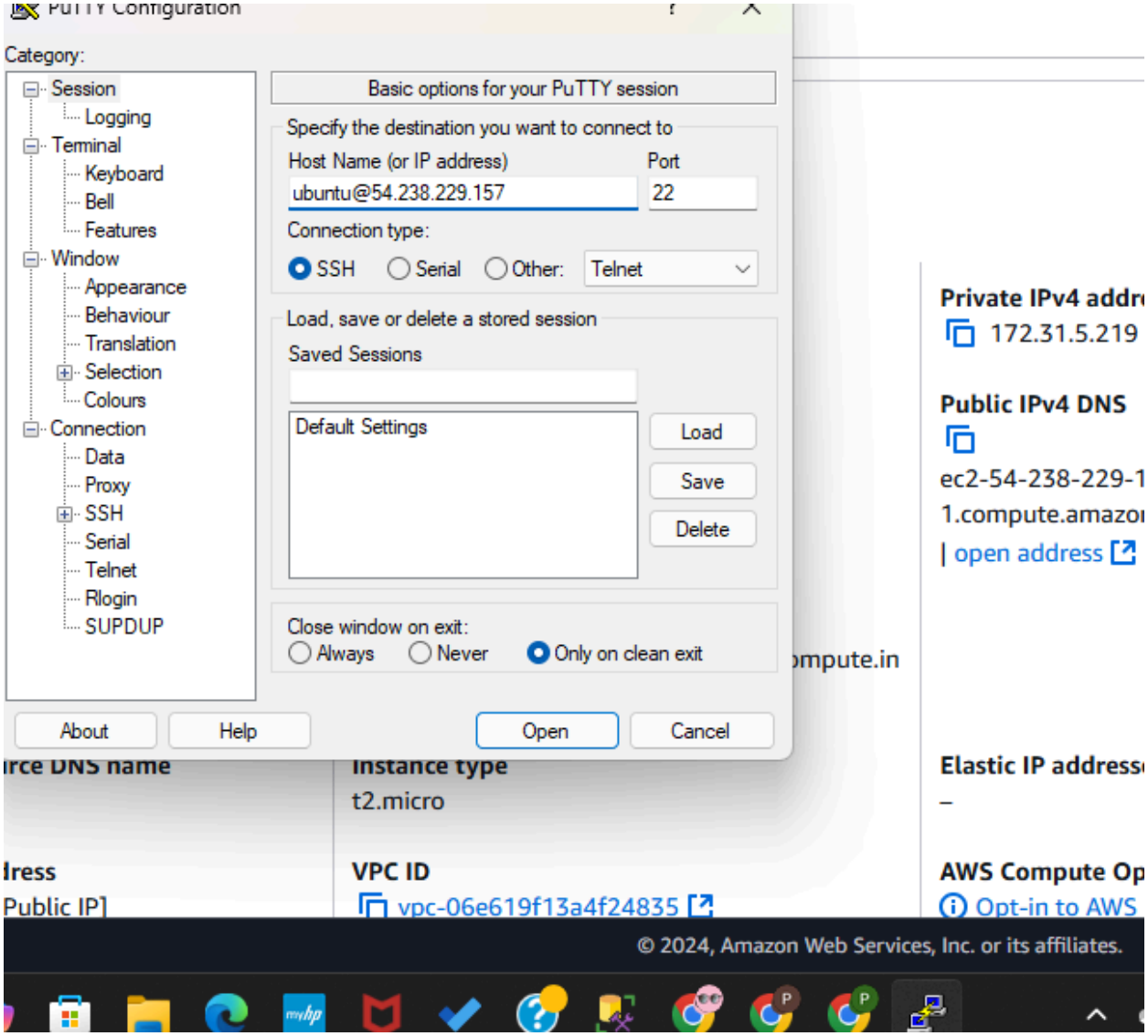
Manage tags | Edit inbound

Search

1

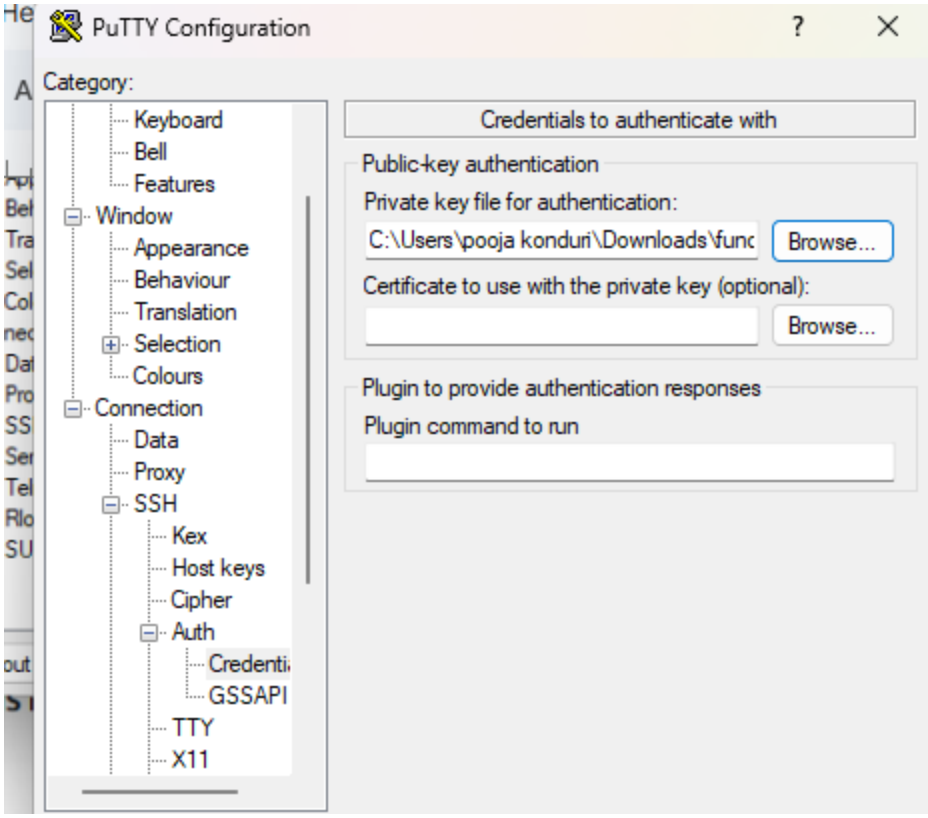
| Name | Security group rule ID | IP version | Type | Protocol |
|---|---|---|---|---|
| – | sgr-050cb9013df5b561c | IPv4 | Custom TCP | TCP |
| – | sgr-0baf0952bbb2fa56e | IPv4 | SSH | TCP |
| – | sgr-0be58363dcf76dedd | IPv4 | PostgreSQL | TCP |

Cpy public ipv4 from ur instances and paste it




Ssh -> auth -> credentials

Upload key dowloaded




Open and accept

Terminal will be opened

Update package index

**sudo apt update**

```
Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted amd
adata [424 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd
[12.2 kB]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse Tra
[2940 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd
s [212 B]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd
adata [356 B]
Fetched 30.8 MB in 15s (2086 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
58 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-5-219:~$
```

**Install PostgreSQL**

**sudo apt install postgresql postgresql-contrib -y**

```
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-5-219:~$
```

**Switch to root user**
**sudo su**

```
ubuntu@ip-172-31-5-219:~$ sudo su
root@ip-172-31-5-219:/home/ubuntu#
```

**Create a new User**
Creating a new user for postgresql with the name postgres
**sudo -i -u postgres**

```
root@ip-172-31-5-219:/home/ubuntu# sudo -i -u postgres
postgres@ip-172-31-5-219:~$
```

**psql**

```
root@ip-172-31-5-219:/home/ubuntu# sudo -i -u
postgres@ip-172-31-5-219:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.

postgres=#
```

Create Database, User and Grant Privileges

**CREATE USER pooja WITH PASSWORD 'pooja';**
**CREATE DATABASE fundo_db;**
**GRANT ALL PRIVILEGES ON DATABASE fundo_db TO pooja;**

```
postgres=# CREATE USER pooja WITH PASSWORD 'pooja';
CREATE ROLE
```

```
postgres=# CREATE DATABASE fundo_db;
CREATE DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE fundo_db TO pooja;
GRANT
postgres=#
```

**\l** [slash L ]
You'll see tables and details

**Configure postgresql.conf**
I will be in postgree path
I have to go to root directory and type our command
**\q**
**Exit**

**sudo nano /etc/postgresql/16/main/postgresql.conf**

By default, PostgreSQL listens on localhost only. To allow remote connections, Find the line with listen_addresses and change it to **listen_addresses = '*'**

```
                                                  List of databases
   Name    |  Owner   | Encoding | Locale Provider | Collate |  Ctype  | ICU Local
e | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+---------+---------+----------
--+-----------+-----------------------
 fundo_db  | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |
  |          | =Tc/postgres         +
          |          |          |                 |         |         |
  |          | postgres=CTc/postgres+
          |          |          |                 |         |         |
  |          | pooja=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)
```

Scroll down with cursor and check for listen_addresses = 'local host'
Uncomment and change it to listen_addresses = '*'

**Ctrl + x , yes and enter**

**Configure pg_hba.conf**

**sudo nano /etc/postgresql/16/main/pg_hba.conf**

Add the following line at the end of the file to allow connections from any IP:
**Host all all 0.0.0.0/0 md5**

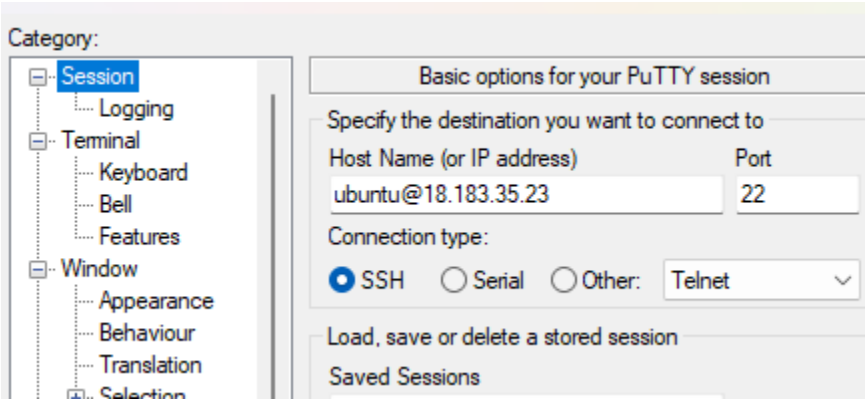
```
# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             0.0.0.0/0               md5
host    all             all             127.0.0.1/32            scram-sha-256
# IPv6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

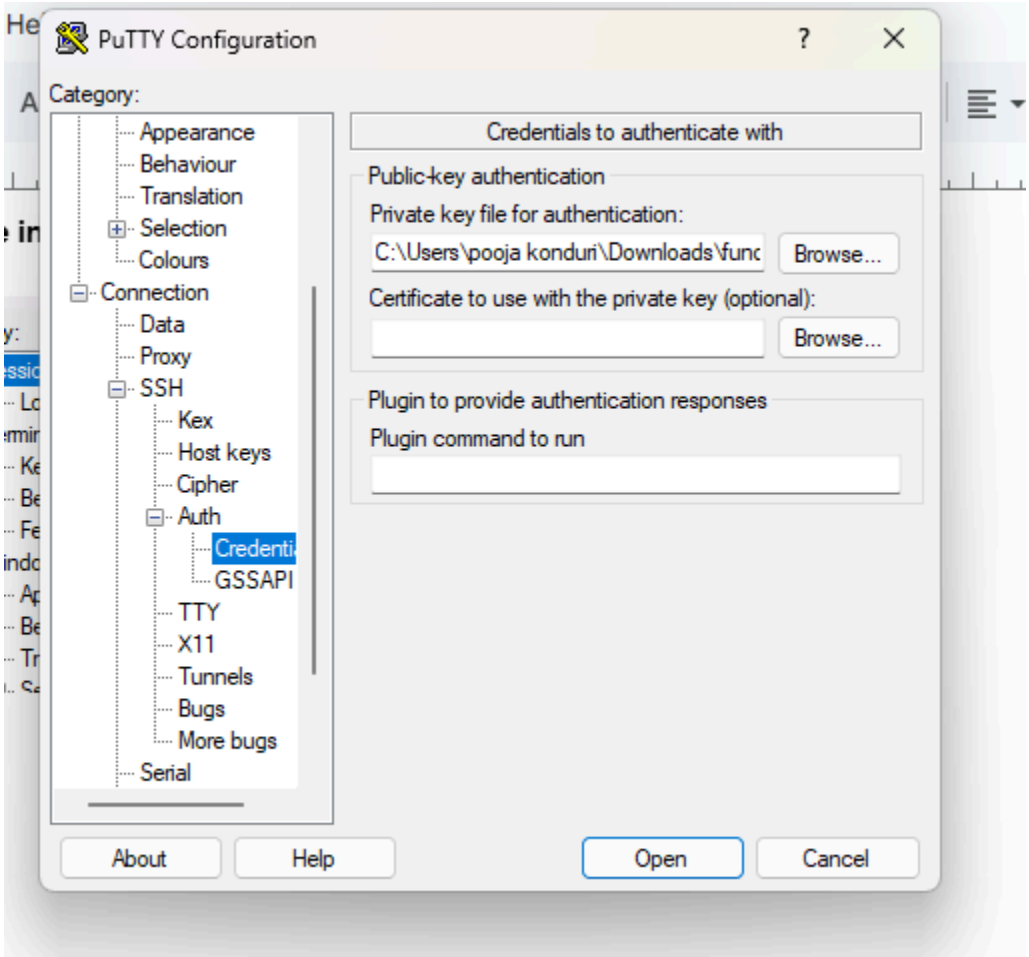
**Enable PostgreSQL to start on boot**
To Enable PostgreSQL to run on ec2 instance startup

**sudo systemctl enable postgresq**

```
root@ip-172-31-5-219:/home/ubuntu# sudo systemctl enable postgresql
Synchronizing state of postgresql.service with SysV service script with /usr/lib/s
ystemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable postgresql
root@ip-172-31-5-219:/home/ubuntu#
```

**Create instance 2 for app (Backend)**
Same procedure as instance 1




**Open -> accept = your command prompt will be open**

**Update package index**

**sudo apt update && sudo apt upgrade -y**

Make sure your command is with particular spaces

```
sudo apt update && sudo apt upgrade-y
```

```
No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: apt[1742], sshd[995]
 ubuntu @ user manager service: systemd[1000]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-10-212:~$
```

**Install Python and pip Django requires Python,**

so install Python and pip (Python's package installer)

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

```
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[995]
 ubuntu @ user manager service: systemd[1000]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-10-212:~$
```

**Install PostgreSQL Development Libraries**

Install PostgreSQL development headers and libraries (necessary for connecting Django to PostgreSQL)

**sudo apt install libpq-dev -y**

```
$  sudo apt install libpq-dev -y
```

```
No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[995]
ubuntu@ip-172-31-10-212:~$
```

**Set Up a Python Virtual Environment**

It’s best practice to use a virtual environment for your Django app to manage dependencies

**python3 -m venv myenv**
**source myenv/bin/activate**

```
ubuntu@ip-172-31-10-212:~$ python3 -m venv myenv
ubuntu@ip-172-31-10-212:~$ source myenv/bin/activate
(myenv) ubuntu@ip-172-31-10-212:~$
```

**install Django and Gunicorn (the production WSGI server)**
**pip install django gunicorn**

```
Collecting packaging (from gunicorn)
  Downloading packaging-24.2-py3-none-any.whl.metadata (3.2 kB)
Downloading Django-5.1.3-py3-none-any.whl (8.3 MB)
   ---------------------------------------- 8.3/8.3 MB 10.9 MB/s eta 0:00:00
Downloading gunicorn-23.0.0-py3-none-any.whl (85 kB)
   ---------------------------------------- 85.0/85.0 kB 9.9 MB/s eta 0:00:00
Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Downloading sqlparse-0.5.2-py3-none-any.whl (44 kB)
   ---------------------------------------- 44.4/44.4 kB 5.7 MB/s eta 0:00:00
Downloading packaging-24.2-py3-none-any.whl (65 kB)
   ---------------------------------------- 65.5/65.5 kB 10.4 MB/s eta 0:00:00
Installing collected packages: sqlparse, packaging, asgiref, gunicorn, django
Successfully installed asgiref-3.8.1 django-5.1.3 gunicorn-23.0.0 packaging-24.2 sqlparse-0.5.2
(myenv) ubuntu@ip-172-31-10-212:~$
```

**Clone the Django project from Github**

**git clone-b <branchname> <link>**

```
12:~$ git clone -b dev https://github.com/antimaYAD/FUNDOO-NOTES
.
240, done.
% (240/240), done.
```

**Install requirement**
**Pip install -r requirements.txt**

**Error**

```
Resolving deltas: 100% (147/147), done.
(myenv) ubuntu@ip-172-31-10-212:~$ pip install -r requirements.txt
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
(myenv) ubuntu@ip-172-31-10-212:~$
```

**Path cd F tab goes to FUNDO-NOTES**

```
myenv) ubuntu@ip-172-31-10-212:~$ cd FUNDOO-NOTES/
myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ pip install -r requirements.txt
ollecting amqp==5.2.0 (from -r requirements.txt (line 1))
 Downloading amqp-5.2.0-py3-none-any.whl.metadata (8.9 kB)
```

**Configure PostgreSQL in Django Settings (myenv)**
**nano settings.py**

```
note: This error originates from a subprocess, and is likely not a problem with pip
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ ls
README.md  fundoonote  label  manage.py  notes  pytest.ini  requirements.txt  user
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$
```

**Cd go inside fundonote**

**And type nano settings.py**

**comment load_dotenv #**
**Put name , password of ours , copy private ipv4 of fundo_db and paste in HOST**

```
# https://docs.djangoproject.com/en/5.1/ref/

DATABASES = {
     "default": {
        "ENGINE": "django.db.backends.postgr
        "NAME": 'fundo_db',
        "USER": 'pooja',
        "PASSWORD": 'pooja',
        "HOST" : '172.31.5.219',
        "PORT" : '5432',
    }
}

^G Help        ^O Write Out    ^W Where Is
^X Exit        ^R Read File    ^\ Replace
```

**Install Postgresql Client**
**sudo apt install postgresql-client**
**y**

```
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES/fundoonote$ sudo apt install postgresql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  postgresql-client-16 postgresql-client-common
Suggested packages:
  postgresql-16 postgresql-doc-16
The following NEW packages will be installed:
  postgresql-client postgresql-client-16 postgresql-client-common
0 upgraded, 3 newly installed, 0 to remove and 1 not upgraded.
Need to get 1319 kB of archives.
After this operation, 4235 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

**Test the Connection with Database**

Test the database connection with the following command

**psql -U pooja -d fundo_db -h 172.31.5.219**

**Ip = private fundo_db**

```
d..: command not found
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ cd ..
(myenv) ubuntu@ip-172-31-10-212:~$ psql -U pooja -d fundo_db -h 172.31.5.219
psql: error: connection to server at "172.31.5.219", port 5432 failed: Connection refused
        Is the server running on that host and accepting TCP/IP connections?
(myenv) ubuntu@ip-172-31-10-212:~$
```

**In db**

```
postgres=# \l
postgres=# select current_database();
current_database
----------------
postgres
(1 row)

postgres=# \c fundo_db
You are now connected to database "fundo_db" as user "postgres".
fundo_db=#
```

**ERROR :** connection refused, i rectified by doing some commands **in db (granting acces commands) and commented load_env**

**After making changes in db start the below command in db and check for connection**

```
ubuntu@ip-172-31-5-219:~$ sudo nano /etc/postgresql/16/main/pg_hba.conf
ubuntu@ip-172-31-5-219:~$ sudo systemctl restart postgresql
ubuntu@ip-172-31-5-219:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
```

**In 2nd  instance(app)**

```
(myenv) ubuntu@ip-172-31-10-212:~$ cd FUNDOO-NOTES/
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ psql -U pooja -d fundo_db -h 172.31.5.219
Password for user pooja:
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.

fundo_db=# \q
```

**Error = no module dotenv**

**Go to env virtual environment and install for all packages asked for**

**python3-m venv myenv**
**source myenv/bin/activate**

Install all modules

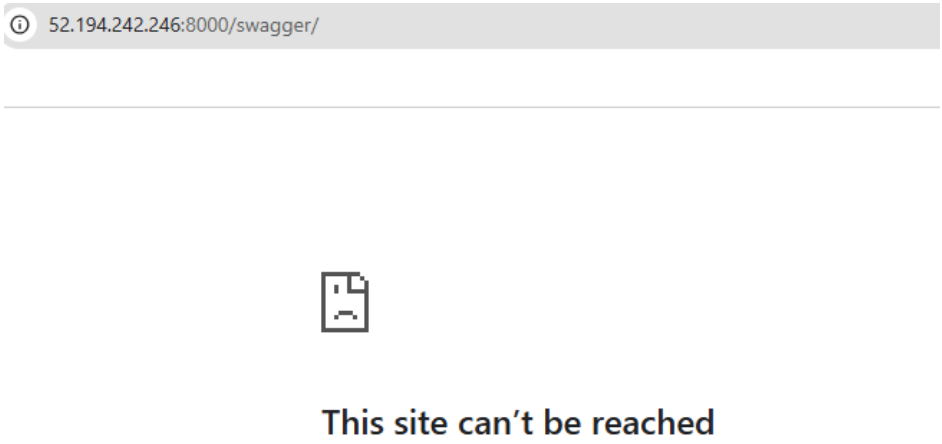
**pip install python-dotenv,pip install celery,pip install loguru,pip install django-celery-beat pip install django-celery-results,pip install djangorestframework,pip install djangorestframework-simplejwt,pip install drf-yasg,pip install psycopg2,pip install psycopg, pip install django-redis, pip install Pillow**

**Run the server**

**python manage.py runserver**

```
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 60 unapplied migration(s). Your project may not work properly until you appl
celery_results, label, notes, sessions, user.
Run 'python manage.py migrate' to apply them.
November 29, 2024 - 10:57:48
Django version 5.1.3, using settings 'fundoonote.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

**Copy pubic ipv4 of 2nd instance and paste in browser**




**sudo apt install redis-server -y  in env**

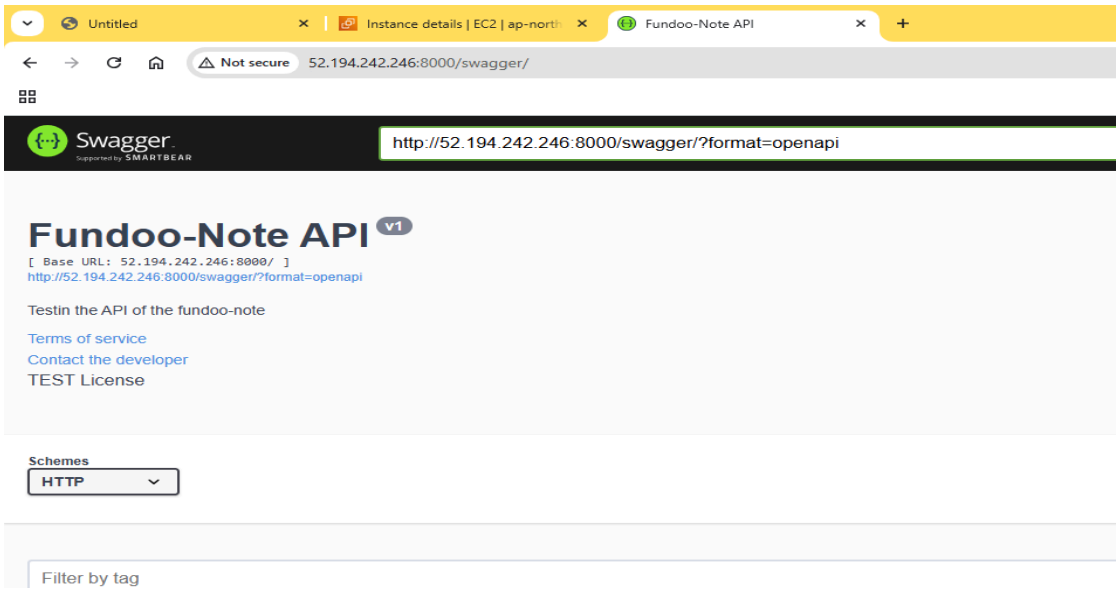
**Migrate the Database**
python manage.py migrate

```
^C(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$  python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, django_celery_beat, django_celery_results, la
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0001_initial... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
```

**Run Django Locally to Test**
python manage.py runserver 0.0.0.0:8000

```
^C(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ python manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 29, 2024 - 11:14:59
Django version 5.1.3, using settings 'fundoonote.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

**Go to browser and type**
**' your_publicipv4' : 8000/swagger**
http://52.194.242.246:8000/swagger/




**Configure the daemon service file**
We will create a service file so that the django app can run in the background Create a Service File**:**
The service files are usually located in /etc/systemd/system/. You'll create your custom service file there.

**sudo nano /etc/systemd/system/<name>.service**

I gave name as **fundoo-service**

And type below

**[Unit]**
**Description=**Fundoo Django Application
**After=**network.target

**[Service]**
**User=y**our_system_user
**Group**=your_system_group
**WorkingDirectory**=/path/to/your/project  (**/home/ubuntu/FUNDOO-NOTES)**
**ExecStar**t=/path/to/your/virtualenv/bin/python /path/to/your/project/manage.py runserver 0.0.0.0:8000
**Restart**=always
**Environment=**"DJANGO_SETTINGS_MODULE=your_project.settings"

**[Install]**
**WantedBy=multi-user.target**

```
  GNU nano 7.2                                          /etc/systemd/system/fundoo-service.ser
[Unit]
Description=Fundoo Django Application
After=network.target

[Service]
User=ubuntu
Group=ubuntu
WorkingDirectory=/home/ubuntu/FUNDOO-NOTES/fundoonote
ExecStart=/home/ubuntu/myenv/bin/python /home/ubuntu/FUNDOO-NOTES/manage.py runserver 0.0.0.0:8000
Restart=always
Environment=PYTHONUNBUFFERED=1

[Install]
WantedBy=multi-user.target
```

**Come out of it ctrl+x**

Reload the systemd Daemon After creating the service file, reload systemd to recognize the new service.
**sudo systemctl daemon-reload**
Start the Service

**sudo systemctl start fundoo-service.service**
Enable the Service to Start on Boot To ensure the service starts automatically at boot

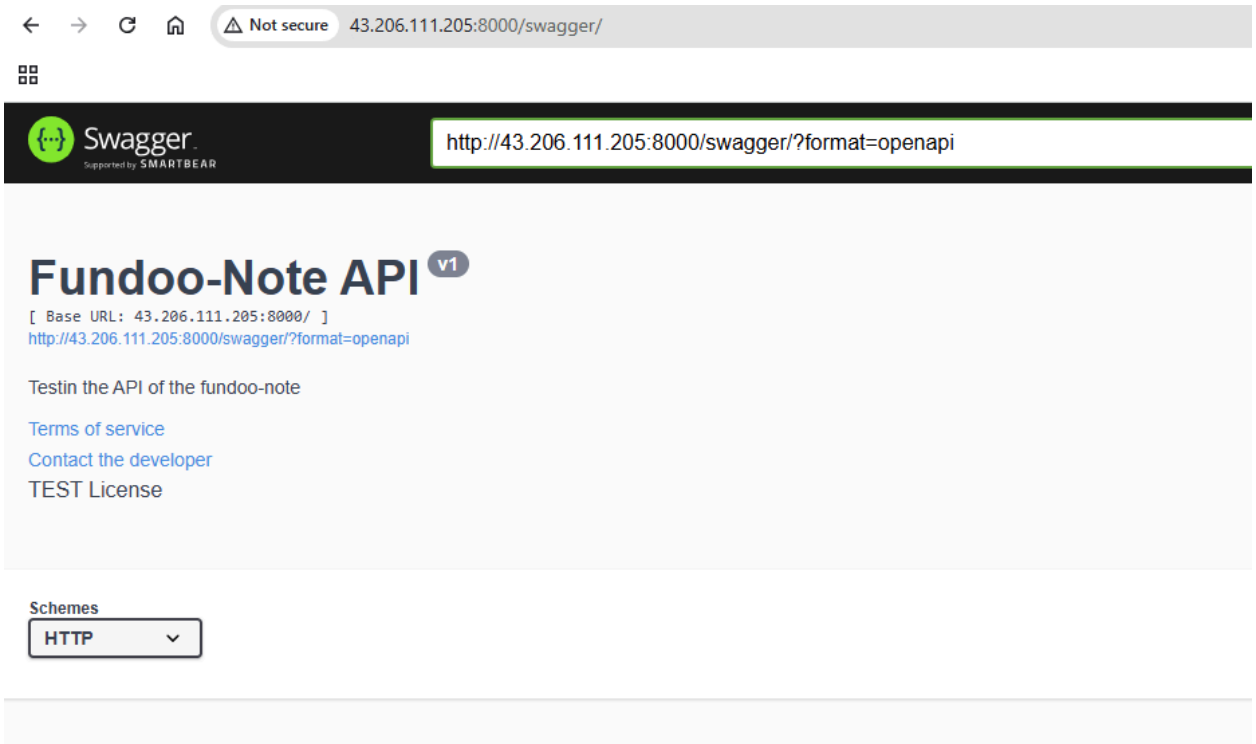
**sudo systemctl enable fundoo-service.service**
Check the Status of the Service Verify that the service is running correctly

**sudo systemctl status fundoo-service.service**

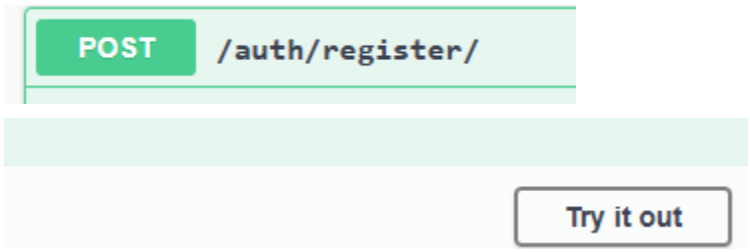
```
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ sudo nano /etc/systemd/system/fundoo-service.service
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$  sudo systemctl daemon-reload
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$  sudo systemctl start fundoo-service
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ sudo systemctl enable fundoo-service
Created symlink /etc/systemd/system/multi-user.target.wants/fundoo-service.service → /etc/systemd/system/fundoo-service.service.
(myenv) ubuntu@ip-172-31-10-212:~/FUNDOO-NOTES$ sudo systemctl status fundoo-service
● fundoo-service.service - Fundoo Django Application
     Loaded: loaded (/etc/systemd/system/fundoo-service.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-11-29 06:19:23 UTC; 32s ago
   Main PID: 2208 (python)
      Tasks: 3 (limit: 1130)
     Memory: 107.2M (peak: 107.4M)
        CPU: 1.794s
     CGroup: /system.slice/fundoo-service.service
             ├─2208 /home/ubuntu/myenv/bin/python /home/ubuntu/FUNDOO-NOTES/manage.py runserver 0.0.0.0:8000
             └─2211 /home/ubuntu/myenv/bin/python /home/ubuntu/FUNDOO-NOTES/manage.py runserver 0.0.0.0:8000

Nov 29 06:19:23 ip-172-31-10-212 systemd[1]: Started fundoo-service.service - Fundoo Django Application.
Nov 29 06:19:24 ip-172-31-10-212 python[2211]: Watching for file changes with StatReloader
Nov 29 06:19:24 ip-172-31-10-212 python[2211]: Performing system checks...
Nov 29 06:19:25 ip-172-31-10-212 python[2211]: System check identified no issues (0 silenced).
Nov 29 06:19:25 ip-172-31-10-212 python[2211]: November 29, 2024 - 11:49:25
Nov 29 06:19:25 ip-172-31-10-212 python[2211]: Django version 5.1.3, using settings 'fundoonote.settings'
Nov 29 06:19:25 ip-172-31-10-212 python[2211]: Starting development server at http://0.0.0.0:8000/
Nov 29 06:19:25 ip-172-31-10-212 python[2211]: Quit the server with CONTROL-C.
```

**Perform API testing**

We can perform api testing using swagger to confirm our applications is running perfectly




**Check register API**



Edit Value | Model


```
{
  "first_name": "pooja",
  "last_name": "konduri",
  "email": "poobbbn15@gmail.com",
  "password": "gmailpooja2"
}
```


Response body


```
{
  "message": "User created successfully",
  "status": "Success",
  "data": {
    "id": 2,
    "first_name": "pooja",
    "last_name": "konduri",
    "email": "poobbbn15@gmail.com"
  }
}
```


Now login API


```
{
  "Message": "Login successful",
  "status": "Success",
  "data": {
    "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
N2U3N5IsInVzZXJfaWQiOjJ9.mQZhOPVeLQlRNzyQB5HEY0-4wDkZpt
    "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
GFlIiwidXNlcl9pZCI6Mn0.N0sVnh7iOZtv5JQFeqEBjYh8eNzXbgu-
  }
}
```