App Deployment

Architecture Components:

- **Instance 1 (Database):** PostgreSQL on EC2
- **Instance 2 (Backend):** Django application on EC2

Step 1: Set Up PostgreSQL on Database Instance

Launch the Database EC2 Instance

1. Choose an **Ubuntu 22.04 AMI**.
2. Configure the security group to allow:
    - **SSH (port 22)**
    - **PostgreSQL (port 5432)** from the IP of the Django instance.

Connect to the Database EC2 Instance

ssh -i path_to_your_key.pem ubuntu@your_database_instance_public_ip

Switch to root user and updating package index

sudo su

sudo apt update

```
ubuntu@ip-172-31-1-154:~$ sudo su
root@ip-172-31-1-154:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2106 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [363 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.9 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2568 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [444 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1132 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [265 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1854 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [26.3 
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
```

Install PostgreSQL

sudo apt update sudo apt install postgresql postgresql-contrib -y

```
root@ip-172-31-1-154:/home/ubuntu# apt install postgresql postgresql-contrib -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libsensors-config libsensors5
  postgresql-client-14 postgresql-client-common postgresql-common ssl-cert sysstat
Suggested packages:
  lm-sensors postgresql-doc postgresql-doc-14 isag
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libsensors-config libsensors5
  postgresql-14 postgresql-client-14 postgresql-client-common postgresql-common postgresql-contrib
0 upgraded, 16 newly installed, 0 to remove and 12 not upgraded.
```

Access PostgreSQL

sudo -u postgres psql

Create Database and User

CREATE DATABASE fundoo_notes_db;

CREATE USER fundoo_user WITH PASSWORD 'your_password';

GRANT ALL PRIVILEGES ON DATABASE fundoo_notes_db TO fundoo_user;

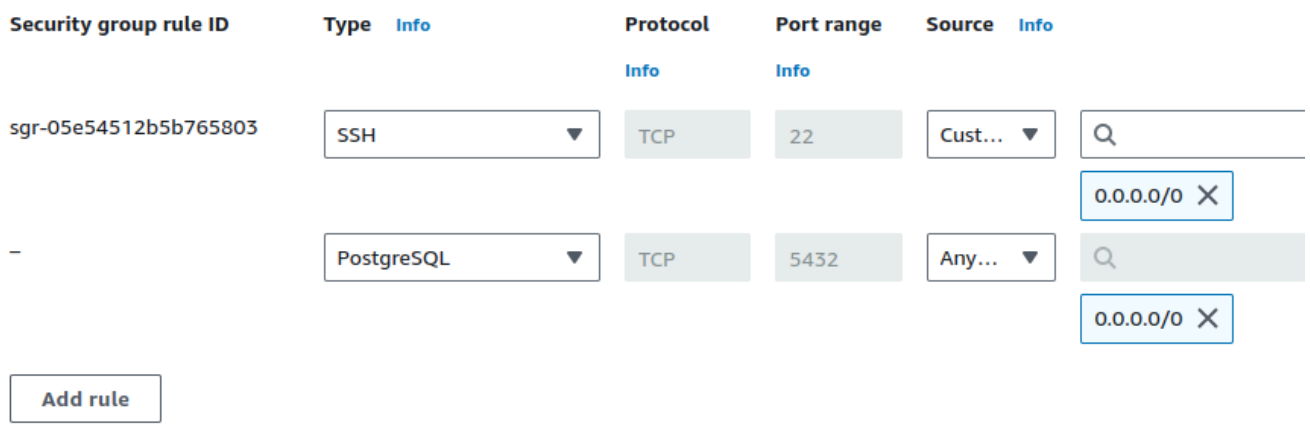
\q

```
root@ip-172-31-1-154:/home/ubuntu# sudo -u postgres psql
could not change directory to "/home/ubuntu": Permission denied
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE fundoo_database;
CREATE DATABASE
postgres=# CREATE USER fundoouser WITH PASSWORD 'deven1234';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE fundoo_database TO fundoouser;
GRANT
postgres=# \q
root@ip-172-31-1-154:/home/ubuntu#
```

```
postgres=# \l
                                  List of databases
      Name        |  Owner   | Encoding | Collate |  Ctype  |    Access privileges
------------------+----------+----------+---------+---------+-----------------------
 fundoo_database  | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =Tc/postgres          +
                  |          |          |         |         | postgres=CTc/postgres +
                  |          |          |         |         | fundoouser=CTc/postgres
 postgres         | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 template0        | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres           +
                  |          |          |         |         | postgres=CTc/postgres
 template1        | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres           +
                  |          |          |         |         | postgres=CTc/postgres
(4 rows)

postgres=#
```



Configure PostgreSQL for Remote Access

1. **Edit pg_hba.conf:**

   sudo nano /etc/postgresql/14/main/pg_hba.conf

   Add the following line (replace your_django_instance_ip):

   host all all your_django_instance_ip/32 md5

```
root@ip-172-31-9-246:/home/ubuntu# sudo nano /etc/postgresql/16/main/postgresql.conf
root@ip-172-31-9-246:/home/ubuntu# sudo nano /etc/postgresql/14/main/postgresql.conf
root@ip-172-31-9-246:/home/ubuntu#
```

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             0.0.0.0/0               md5
# 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
```

2. **Edit postgresql.conf:**

   sudo nano /etc/postgresql/14/main/postgresql.conf

   Change listen_addresses to: **listen_addresses = '*'**

```
root@ip-172-31-9-246:/home/ubuntu# sudo nano /etc/postgresql/14/main/pg_hba.conf
root@ip-172-31-9-246:/home/ubuntu#
```

```
# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for
                                # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of dire
                                # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                # (change requires restart)
```

3. **Restart PostgreSQL:** **sudo service postgresql restart**

Step 2 : Run the Django Application Locally on EC2

Connect to Your EC2 Instance

ssh -i path_to_your_key.pem ubuntu@your_instance_public_ip

Switch to root user and updating package index

sudo su

sudo apt update

```
ubuntu@ip-172-31-1-175:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRele
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-update
128 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backpo
 [127 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/univer
ages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [
```

Navigate to Your Django Project Directory

If you haven't cloned your repository yet, do that first: git clone **https://github.com/Deven5656/fundoo-notes**

cd fundoo_notes

```
(myenv) ubuntu@ip-172-31-1-175:~$ git clone -b dev https://github.com/ayush-prajapati01/
Cloning into 'fundoo-notes-copy'...
remote: Enumerating objects: 130, done.
remote: Counting objects: 100% (130/130), done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 130 (delta 29), reused 127 (delta 29), pack-reused 0 (from 0)
Receiving objects: 100% (130/130), 135.23 KiB | 7.12 MiB/s, done.
Resolving deltas: 100% (29/29), done.
(myenv) ubuntu@ip-172-31-1-175:~$ ls
fundoo-notes-copy  myenv
(myenv) ubuntu@ip-172-31-1-175:~$ cd fundoo-notes-copy/
(myenv) ubuntu@ip-172-31-1-175:~/fundoo-notes-copy$
```

Set Up a Virtual Environment

If you haven't already set up a virtual environment: sudo apt update sudo apt install **python3-pip python3-dev -y**

pip3 install virtualenv

virtualenv venv

source venv/bin/activate

```
ubuntu@ip-172-31-1-175:~$ python3 -m venv myenv
ubuntu@ip-172-31-1-175:~$ source myenv/bin/activate
(myenv) ubuntu@ip-172-31-1-175:~$
```

Install Required Packages

If you have a requirements.txt file, install the necessary packages:

pip install -r requirements.txt

```
(myenv) ubuntu@ip-172-31-1-175:~/fundoo-notes-copy$ pip install -r requirements.txt
Collecting amqp==5.2.0
  Downloading amqp-5.2.0-py3-none-any.whl (50 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.9/50.9 KB 1.7 MB/s eta 0:00:00
Requirement already satisfied: asgiref==3.8.1 in /home/ubuntu/myenv/lib/python3.10/site-
Collecting billiard==4.2.0
  Downloading billiard-4.2.0-py3-none-any.whl (86 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.7/86.7 KB 5.4 MB/s eta 0:00:00
Collecting celery==5.4.0
  Downloading celery-5.4.0-py3-none-any.whl (425 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 426.0/426.0 KB 23.1 MB/s eta 0:00:00
Collecting click==8.1.7
```

Configure Database Settings

Ensure your settings.py is configured to connect to your PostgreSQL database. Update the database settings as follows:

DATABASES = { 'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'fundoo_notes_db',

'USER': 'fundoo_user',

'PASSWORD': 'your_password',

'HOST': 'your_database_instance_private_ip',

'PORT': '5432', }

}

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'fundoo_database',
        'USER': 'fundoouser',
        'PASSWORD': 'deven1234',
        'HOST': '172.31.1.154',
        'PORT': '5432',
    }
}
```

Run Migrations

If you haven't done this already:

python manage.py makemigrations

python manage.py migrate

```
------------+------------------------------------------------------------+-----------
 fundoouser |                                                            | {}
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

fundoo_database=> exit
(venv) root@ip-172-31-6-72:/home/ubuntu/fundoo-notes/fundoo_notes# ls
app  fundoo_notes  label  logs  manage.py  notes  pytest.ini  results  templates  transport  user_auth  utils
(venv) root@ip-172-31-6-72:/home/ubuntu/fundoo-notes/fundoo_notes# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, django_celery_beat, label, notes, sessions, user_auth
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
  Applying user_auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying django_celery_beat.0001_initial... OK
  Applying django_celery_beat.0002_auto_20161118_0346... OK
  Applying django_celery_beat.0003_auto_20161209_0049... OK
  Applying django_celery_beat.0004_auto_20170221_0000... OK
  Applying django_celery_beat.0005_add_solarschedule_events_choices... OK
  Applying django_celery_beat.0006_auto_20180322_0932... OK
  Applying django_celery_beat.0007_auto_20180521_0826... OK
  Applying django_celery_beat.0008_auto_20180914_1922... OK
  Applying django_celery_beat.0006_auto_20180210_1226... OK
  Applying django_celery_beat.0006_periodictask_priority... OK
```

Start the Django Development Server

Use the following command to run the development server:

python manage.py runserver 0.0.0.0:8000

Note - After following the above steps your application with run in *Foreground* i.e if you close the terminal ,the application also close

Hence to make application run in *background* you need to create a systemd service file (daemon file)

Step 3 : creating systemd (Daemon) service file

Navigate to system service directory

the following command will go to directory where all daemon service file is available:

```
cd  /etc/systemd/system
```

Create a new daemon service file using .service extension

the following command will open a vim editor for **fundooauto.service** file :

```
sudo vim /lib/systemd/system/fundooauto.service
```

Add below configuration :-

```
[Unit]
Description=Fundoo Notes Service
After=network.target
[Service]
User=root
Group=root
# EnvironmentFile=/etc/fundoo_notes/env.conf
WorkingDirectory=/home/ubuntu/fundoo-notes/fundoo_notes
ExecStart=/bin/bash -c "cd /home/ubuntu/fundoo-notes && source myenv/bin/activate
&& python3 /home/ubuntu/fundoo notes/fundoo_notes/manage.py runserver
0.0.0.0:8000"
[Install]
WantedBy=multi-user.target
```

Systemd Service Management Commands

The following commands are used to manage a systemd service named `fundooauto.service` on a Linux system.

Systemd is a system and service manager for Linux operating systems.

Reload systemd manager configuration

sudo systemctl daemon-reload

Enable the service to start at boot

sudo systemctl enable fundooauto.service

Start the service immediately

sudo systemctl start fundooauto.service

Check the status of the service

sudo systemctl status fundooauto.service

```
root@ip-172-31-6-72:/etc/systemd/system# vim fundooauto.service
root@ip-172-31-6-72:/etc/systemd/system# sudo systemctl daemon-reload
root@ip-172-31-6-72:/etc/systemd/system# sudo systemctl enable fundooauto.service
root@ip-172-31-6-72:/etc/systemd/system# sudo systemctl start fundooauto.service
root@ip-172-31-6-72:/etc/systemd/system# sudo systemctl status fundooauto.service
● fundooauto.service - Fundoo Notes Service
     Loaded: loaded (/etc/systemd/system/fundooauto.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-10-16 12:17:36 UTC; 2s ago
   Main PID: 5237 (python3)
      Tasks: 3 (limit: 1130)
     Memory: 95.8M
        CPU: 1.620s
     CGroup: /system.slice/fundooauto.service
             ├─5237 python3 /home/ubuntu/fundoo-notes/fundoo_notes/manage.py runserver 0.0.0.0:8000
             └─5240 /home/ubuntu/fundoo-notes/venv/bin/python3 /home/ubuntu/fundoo-notes/fundoo_notes/manage.py runserver 0.0.0.0:8000

Oct 16 12:17:36 ip-172-31-6-72 systemd[1]: Started Fundoo Notes Service.
Oct 16 12:17:37 ip-172-31-6-72 bash[5240]: Watching for file changes with StatReloader
root@ip-172-31-6-72:/etc/systemd/system#
```