Deploy Django Backend on EC2 Instance

Architecture Components

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

Configuring Instance 1 (Database)

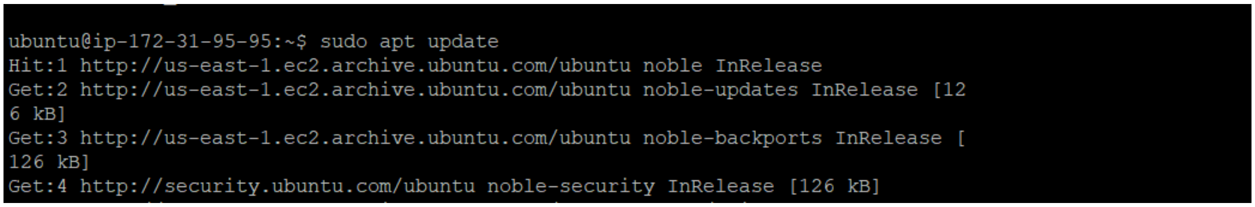
Connect to the Database EC2 Instance

ssh -i path_to_your_key.pem ubuntu@your_database_instance_public_ip

Note: Alternatively, you can use Putty

Update package index

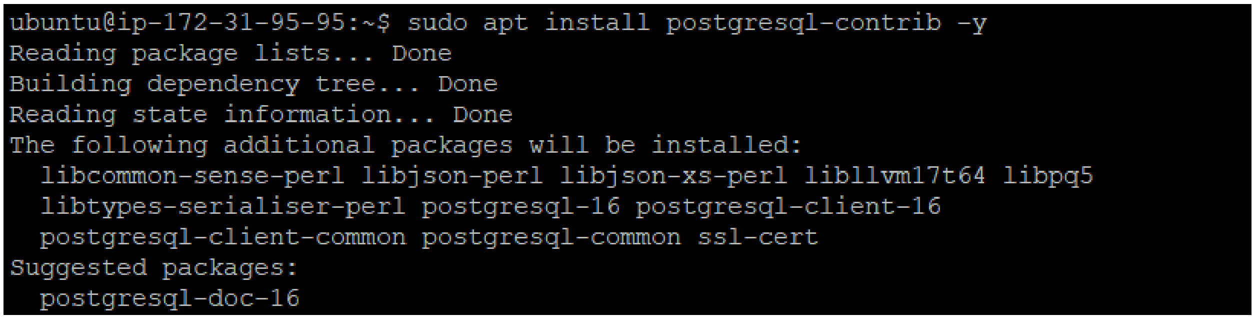
sudo apt update



```
ubuntu@ip-172-31-95-95:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [
126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
```

Install PostgreSQL

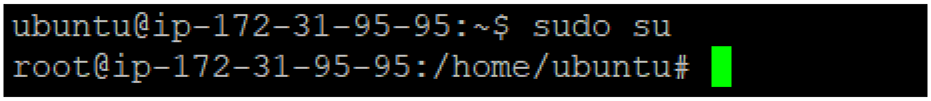
sudo apt update sudo apt install postgresql postgresql-contrib -y



```
ubuntu@ip-172-31-95-95:~$ sudo apt install 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 libllvm17t64 libpq5
  libtypes-serialiser-perl postgresql-16 postgresql-client-16
  postgresql-client-common postgresql-common ssl-cert
Suggested packages:
  postgresql-doc-16
```

Switch to root user

Sudo su

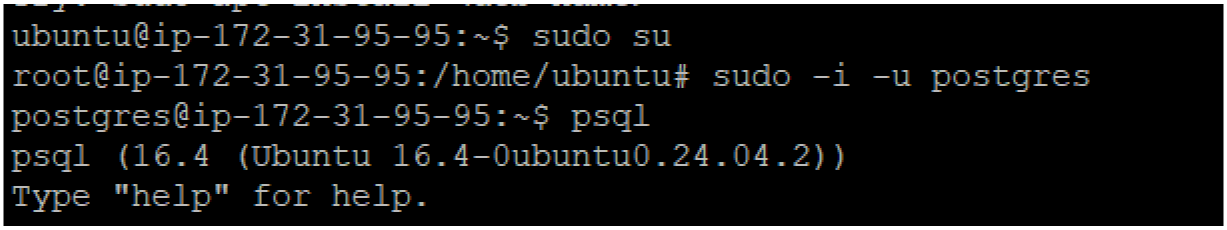


```
ubuntu@ip-172-31-95-95:~$ sudo su
root@ip-172-31-95-95:/home/ubuntu#
```

Create a new User

Creating a new user for postgresql with the name postgres

sudo -i -u postgres

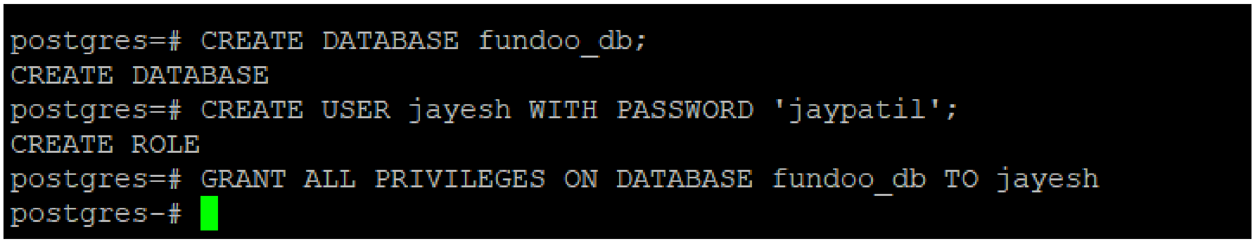


```
ubuntu@ip-172-31-95-95:~$ sudo su
root@ip-172-31-95-95:/home/ubuntu# sudo -i -u postgres
postgres@ip-172-31-95-95:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.
```

Access the Postgresql

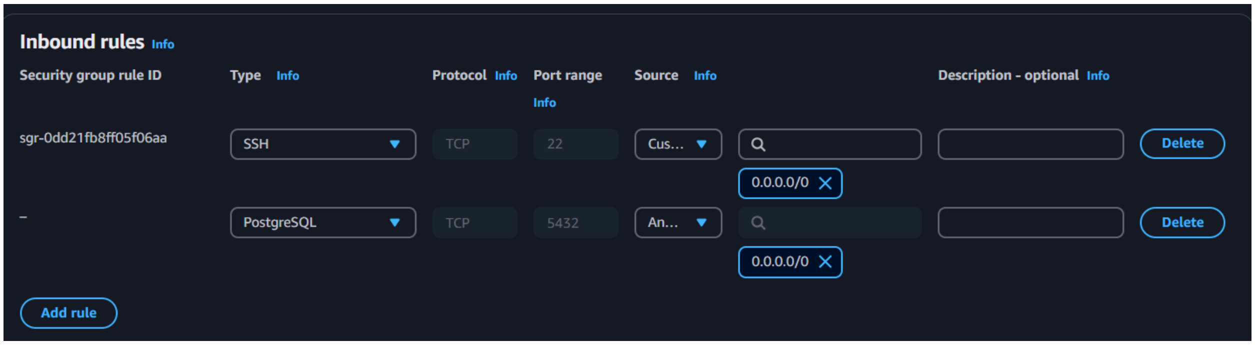
Psql

Create Database, User and Grant Privileges



```
postgres=# CREATE DATABASE fundoo_db;
CREATE DATABASE
postgres=# CREATE USER jayesh WITH PASSWORD 'jaypatil';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE fundoo_db TO jayesh
postgres-#
```

Configure EC2 Security Group



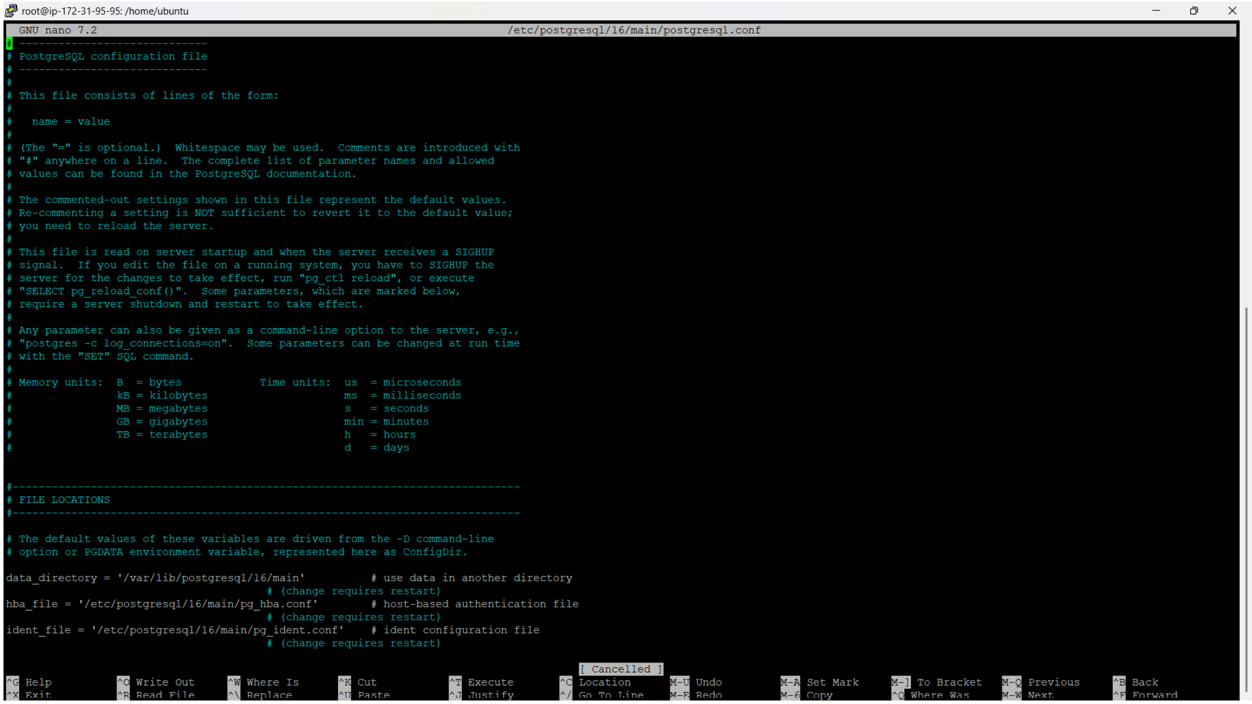
Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0dd21fb8ff05f06aa	SSH ▼	TCP	22	Cus... ▼	🔍 0.0.0.0/0 ✕		Delete
–	PostgreSQL ▼	TCP	5432	An... ▼	🔍 0.0.0.0/0 ✕		Delete

Configure postgresql.conf

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 = '*'



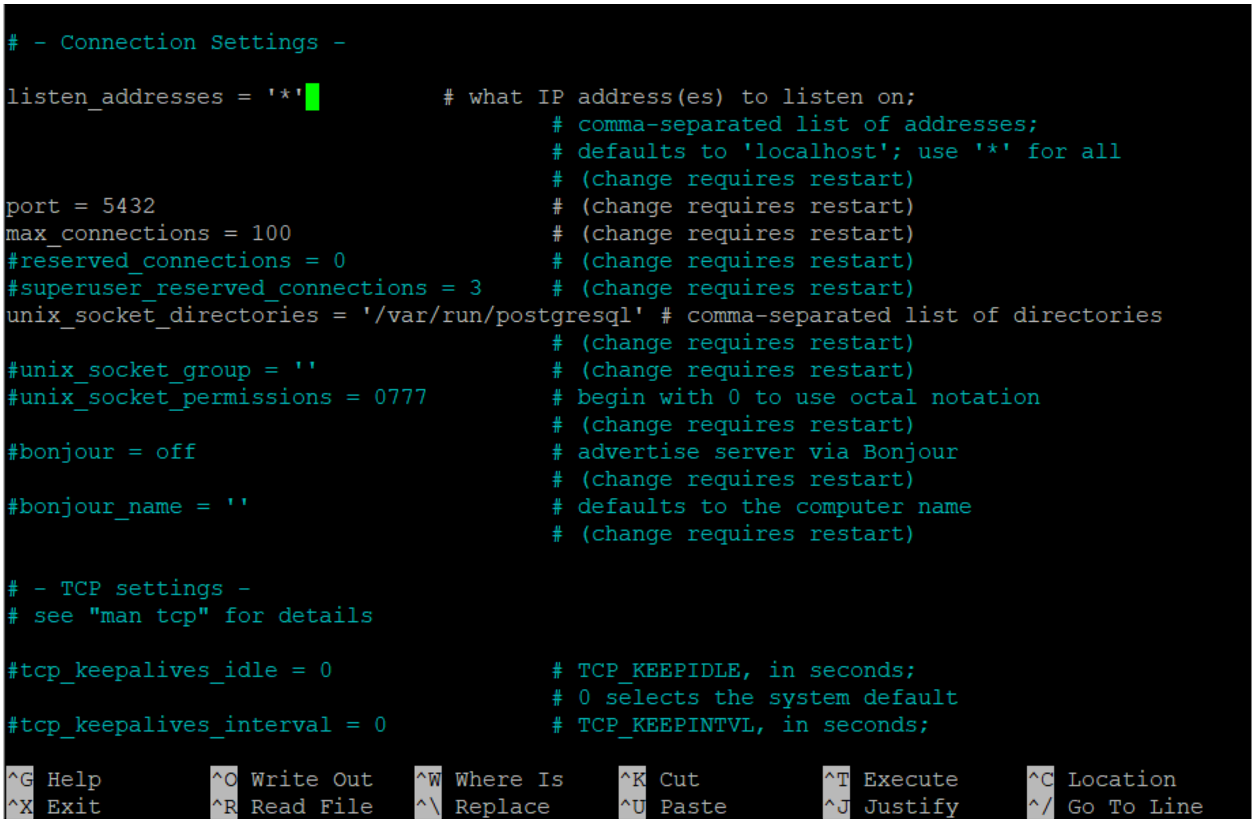
```
  GNU nano 7.2                                           /etc/postgresql/16/main/postgresql.conf
# -----------------------------
# PostgreSQL configuration file
# -----------------------------
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, run "pg_ctl reload", or execute
# "SELECT pg_reload_conf()".  Some parameters, which are marked below,
# require a server shutdown and restart to take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  B  = bytes            Time units:  us  = microseconds
#                kB = kilobytes                     ms  = milliseconds
#                MB = megabytes                     s   = seconds
#                GB = gigabytes                     min = minutes
#                TB = terabytes                     h   = hours
#                                                   d   = days


#------------------------------------------------------------------------------
# FILE LOCATIONS
#------------------------------------------------------------------------------

# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

data_directory = '/var/lib/postgresql/16/main'          # use data in another directory
                                        # (change requires restart)
hba_file = '/etc/postgresql/16/main/pg_hba.conf'        # host-based authentication file
                                        # (change requires restart)
ident_file = '/etc/postgresql/16/main/pg_ident.conf'    # ident configuration file
                                        # (change requires restart)

                                                       [ Cancelled ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket M-Q Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was   M-W Next       ^F Forward
```

Edit `postgresql.conf`:

Check for the version:
Change `listen_addresses` to: `listen_addresses = '*'`

Edit `pg_hba.conf`:

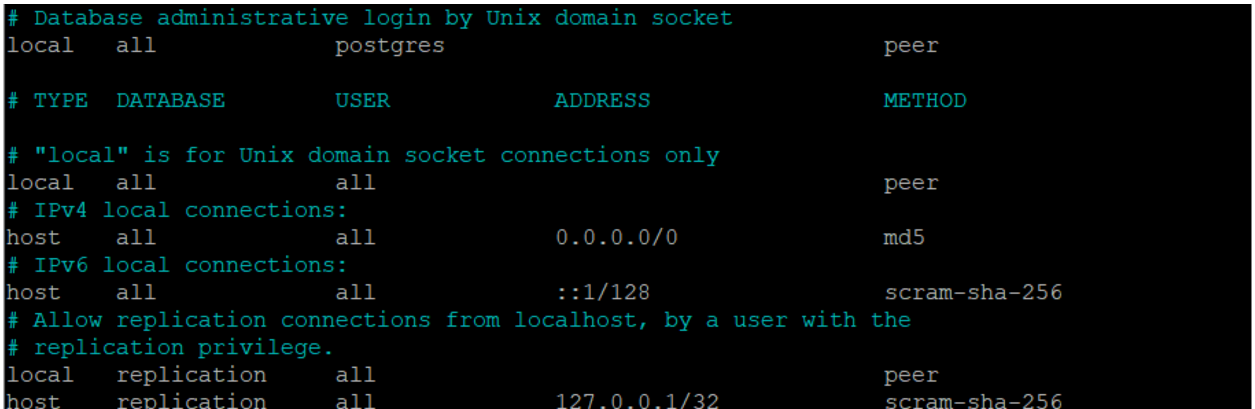
```
sudo nano /etc/postgresql/16/main/pg_hba.conf
```



```
# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#reserved_connections = 0               # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                                        # (change requires restart)

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0                # TCP_KEEPIDLE, in seconds;
                                        # 0 selects the system default
#tcp_keepalives_interval = 0            # TCP_KEEPINTVL, in seconds;

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line
```

Add the following line at the end of the file:

```
host    all             all             0.0.0.0/0               md5
```

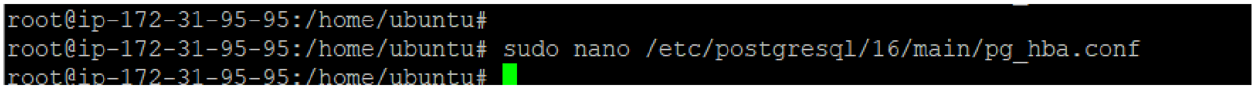


```
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# 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
```

Enable PostgreSQL to start on boot

To Enable PostgreSQL to run on ec2 instance startup

sudo systemctl enable postgresql

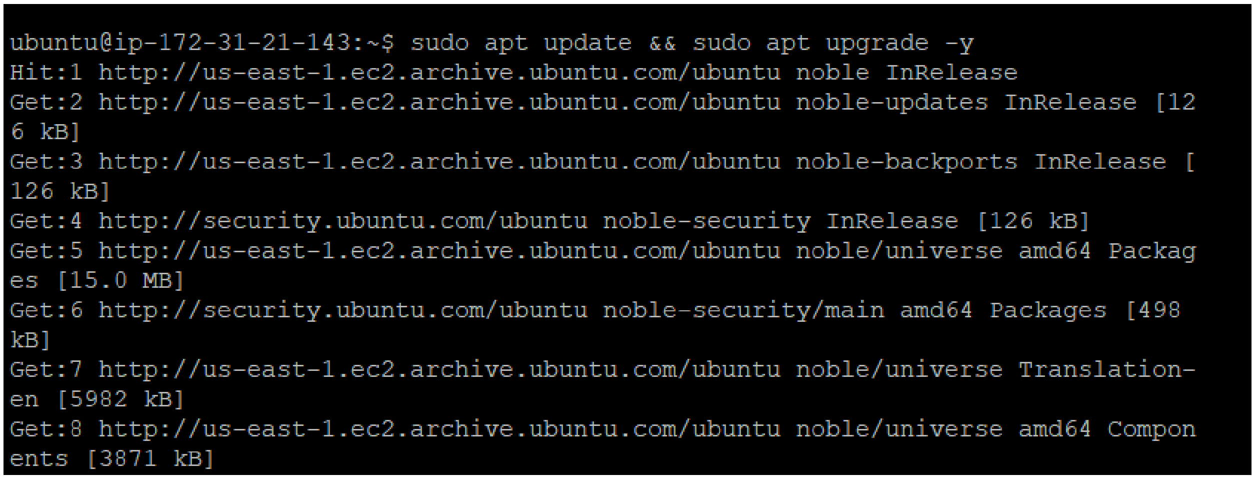


```
root@ip-172-31-95-95:/home/ubuntu#
root@ip-172-31-95-95:/home/ubuntu# sudo nano /etc/postgresql/16/main/pg_hba.conf
root@ip-172-31-95-95:/home/ubuntu#
```

Configuring Instance 2 (Backend)

Update package index

sudo apt update && sudo apt upgrade -y

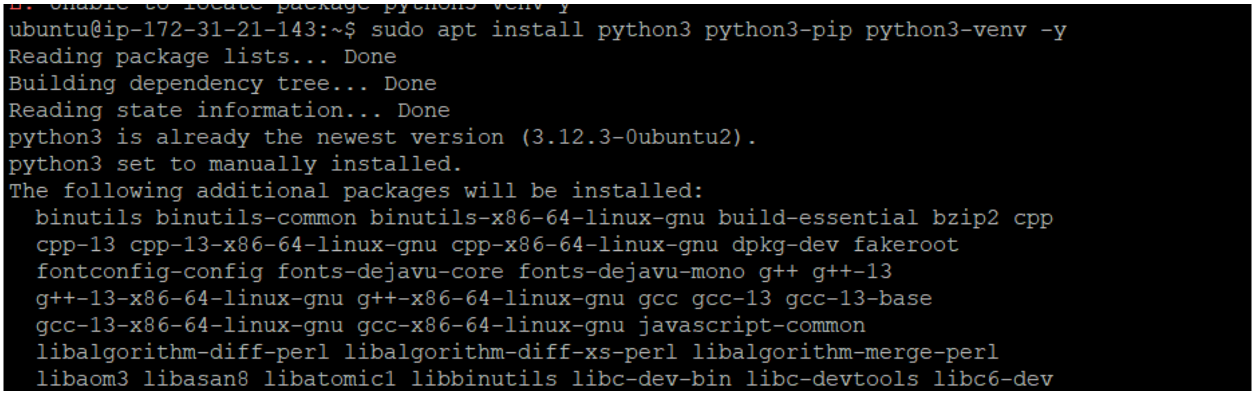


```
ubuntu@ip-172-31-21-143:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [
126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packag
es [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [498
kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-
en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Compon
ents [3871 kB]
```

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

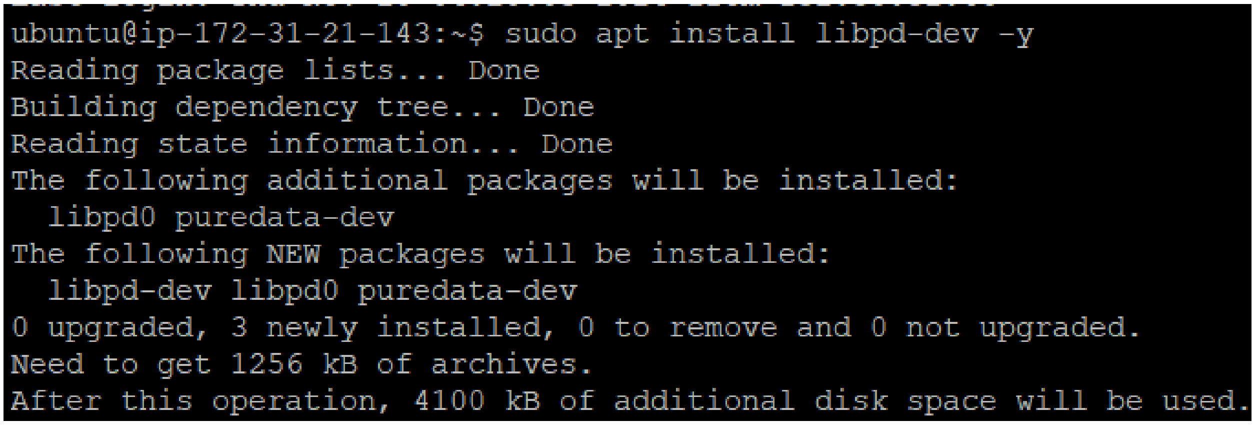


```
E: Unable to locate package python3-venv y
ubuntu@ip-172-31-21-143:~$ sudo apt install python3 python3-pip python3-venv -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.12.3-0ubuntu2).
python3 set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp
  cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu dpkg-dev fakeroot
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono g++ g++-13
  g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base
  gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libaom3 libasan8 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev
```

Install PostgreSQL Development Libraries

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

sudo apt install libpq-dev -y

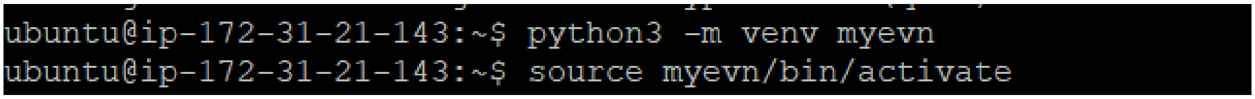


```
ubuntu@ip-172-31-21-143:~$ sudo apt install libpd-dev -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpd0 puredata-dev
The following NEW packages will be installed:
  libpd-dev libpd0 puredata-dev
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 1256 kB of archives.
After this operation, 4100 kB of additional disk space will be used.
```

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

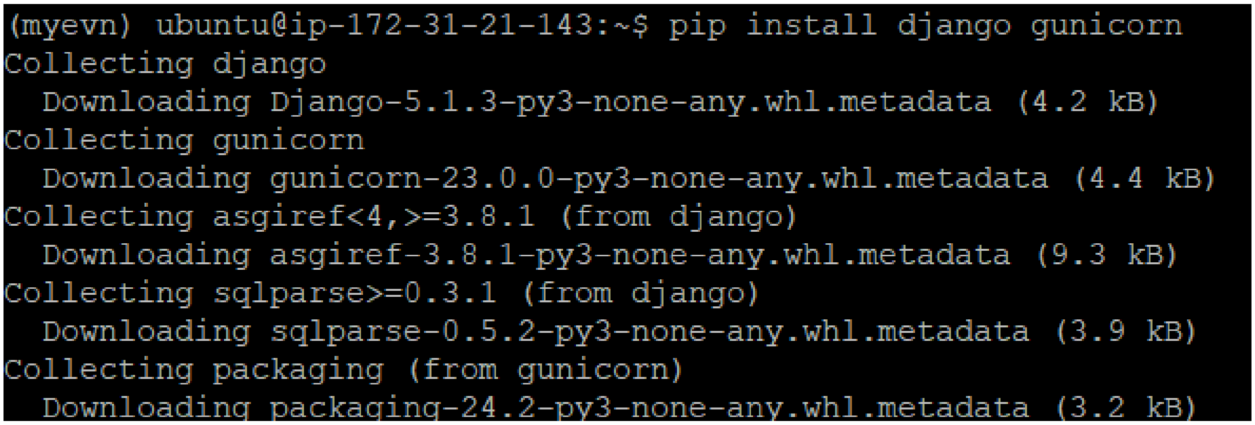


```
ubuntu@ip-172-31-21-143:~$ python3 -m venv myevn
ubuntu@ip-172-31-21-143:~$ source myevn/bin/activate
```

Install Django and Gunicorn

Install Django and Gunicorn (the production WSGI server)

pip install django gunicorn

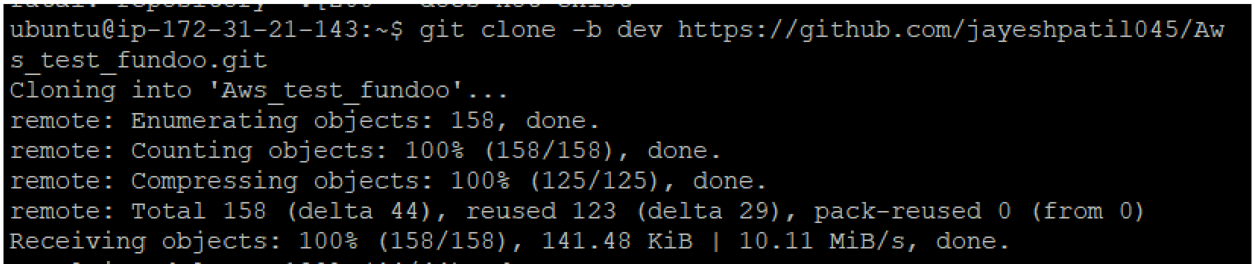


```
(myevn) ubuntu@ip-172-31-21-143:~$ pip install django gunicorn
Collecting django
  Downloading Django-5.1.3-py3-none-any.whl.metadata (4.2 kB)
Collecting gunicorn
  Downloading gunicorn-23.0.0-py3-none-any.whl.metadata (4.4 kB)
Collecting asgiref<4,>=3.8.1 (from django)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Downloading sqlparse-0.5.2-py3-none-any.whl.metadata (3.9 kB)
Collecting packaging (from gunicorn)
  Downloading packaging-24.2-py3-none-any.whl.metadata (3.2 kB)
```

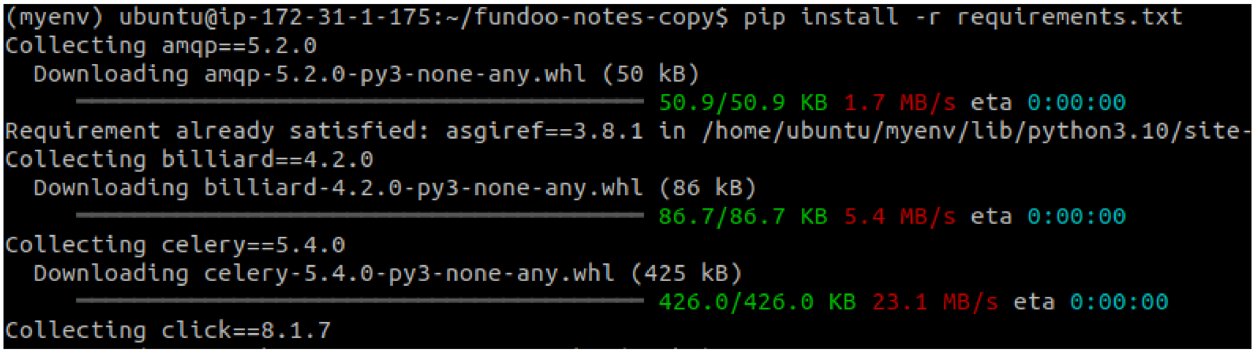
Clone the Django project from Github

git clone -b <branch-name> <repo-link>

Install requirements.txt



```
ubuntu@ip-172-31-21-143:~$ git clone -b dev https://github.com/jayeshpatil045/Aw
s_test_fundoo.git
Cloning into 'Aws_test_fundoo'...
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 | 10.11 MiB/s, done.
```



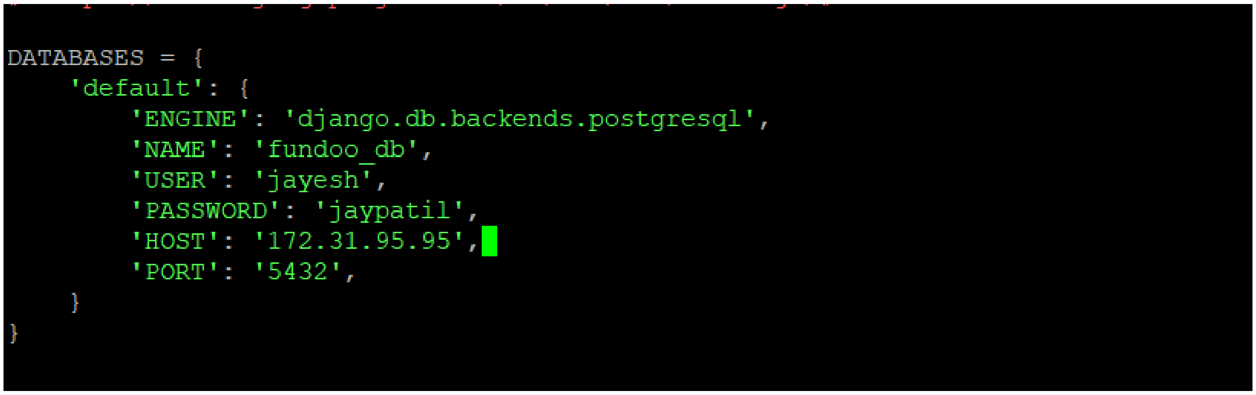
```
(myenv) ubuntu@ip-172-31-1-175:~/fundoo-notes-copy$ pip install -r requirements.txt
Collecting amqp==5.2.0
  Downloading amqp-5.2.0-py3-none-any.whl (50 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.9/50.9 KB 1.7 MB/s eta 0:00:00
Requirement already satisfied: asgiref==3.8.1 in /home/ubuntu/myenv/lib/python3.10/site-
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 PostgreSQL in Django Settings

(myenv)
ubuntu@ip-172-31-1-175:~/fundoo-notes-copy/fundoo_notes/fundoo_notes$ nano settings.py

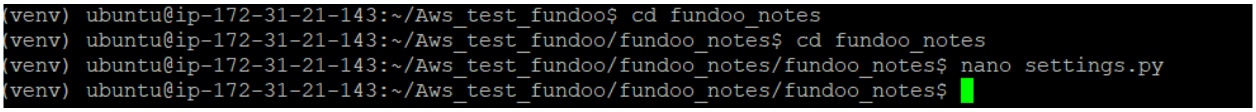
Allow all host and Change databases settings

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.postgresql',
'NAME': 'django_db',
'USER': 'shiv_database_user',
'PASSWORD': 'strongpassword',
'HOST': 'your_postgres_ec2_instance_private_ip', # Use private IP of EC2 instance 1
'PORT': '5432',
}
}



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'fundoo_db',
        'USER': 'jayesh',
        'PASSWORD': 'jaypatil',
        'HOST': '172.31.95.95',
        'PORT': '5432',
    }
}
```

Install Postgresql Client



```
(venv) ubuntu@ip-172-31-21-143:~/Aws_test_fundoo$ cd fundoo_notes
(venv) ubuntu@ip-172-31-21-143:~/Aws_test_fundoo/fundoo_notes$ cd fundoo_notes
(venv) ubuntu@ip-172-31-21-143:~/Aws_test_fundoo/fundoo_notes/fundoo_notes$ nano settings.py
(venv) ubuntu@ip-172-31-21-143:~/Aws_test_fundoo/fundoo_notes/fundoo_notes$ 
```

Test the Connection with Database

Test the database connection with the following command
psql -U shiv_database_user -d fundoo_db -h 172.31.9.246

Migrate the Database

python manage.py migrate

Run Django Locally to Test

python manage.py runserver 0.0.0.0:8000

Configure the daemon service file

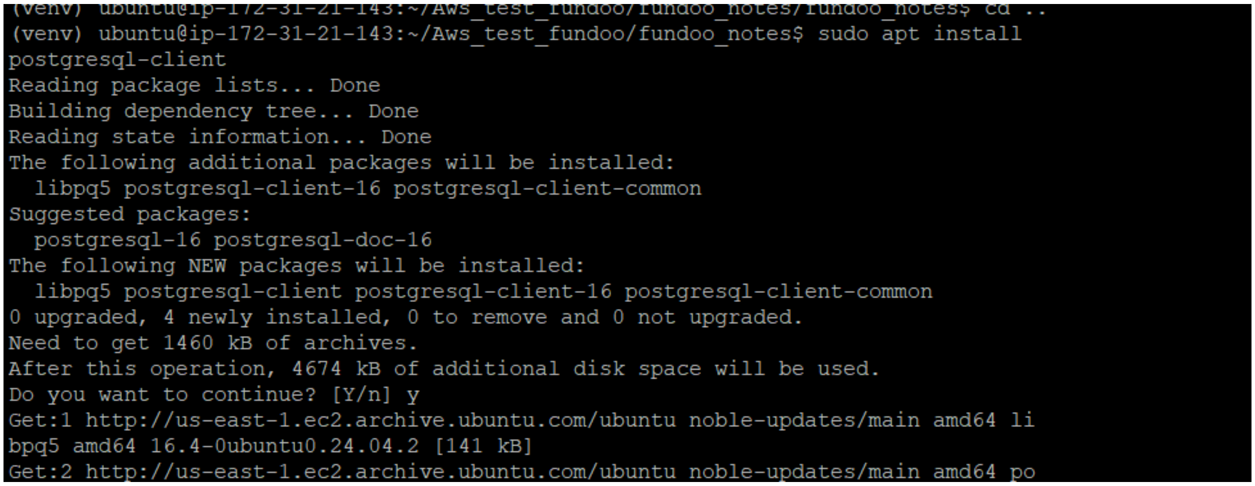
We will create a service file so that the django app can run in the background

Create a Service File:

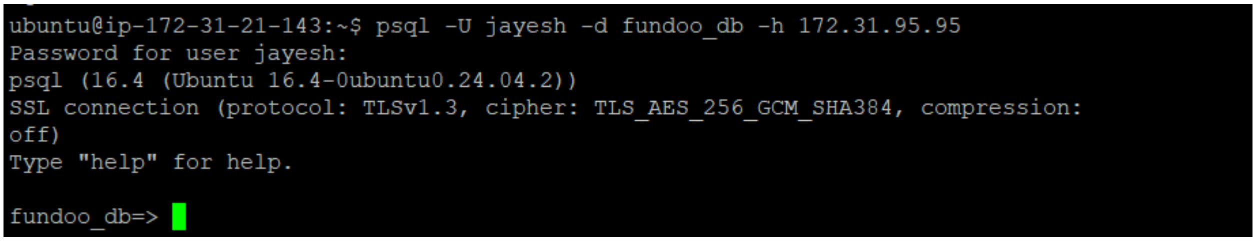
The service files are usually located in /etc/systemd/system/. You'll create your custom service file there.
sudo nano /etc/systemd/system/<name>.service

Define the Service Configuration

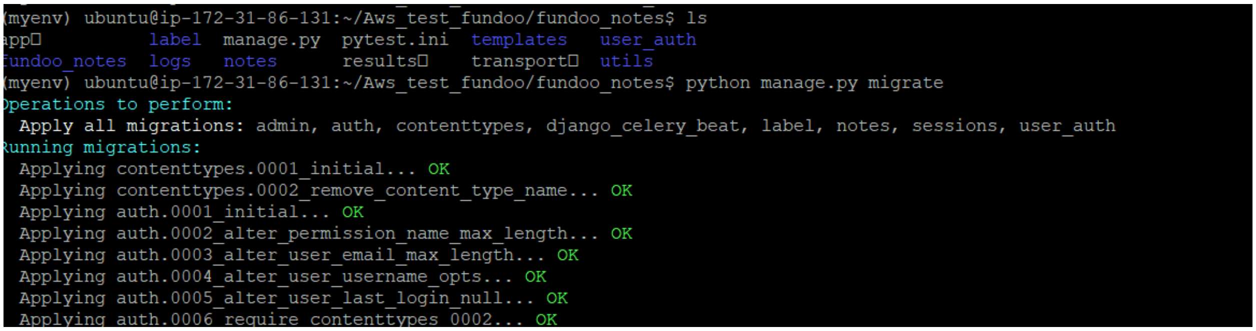
sudo vim fundoo-service.service



```
(venv) ubuntu@ip-172-31-21-143:~/Aws_test_fundoo/fundoo_notes/fundoo_notes$ cd ..
(venv) ubuntu@ip-172-31-21-143:~/Aws_test_fundoo/fundoo_notes$ sudo apt install
postgresql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpq5 postgresql-client-16 postgresql-client-common
Suggested packages:
  postgresql-16 postgresql-doc-16
The following NEW packages will be installed:
  libpq5 postgresql-client postgresql-client-16 postgresql-client-common
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 1460 kB of archives.
After this operation, 4674 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 li
bpq5 amd64 16.4-0ubuntu0.24.04.2 [141 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 po
```



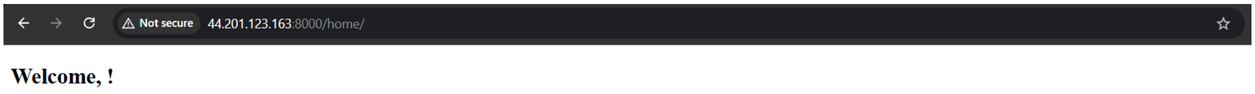
```
ubuntu@ip-172-31-21-143:~$ psql -U jayesh -d fundoo_db -h 172.31.95.95
Password for user jayesh:
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression:
off)
Type "help" for help.

fundoo_db=>
```



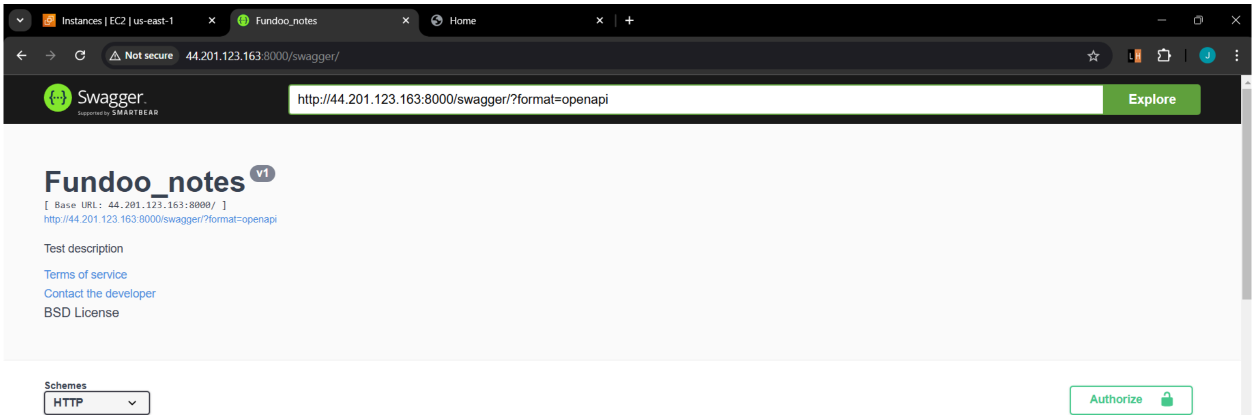
```
(myenv) ubuntu@ip-172-31-86-131:~/Aws_test_fundoo/fundoo_notes$ ls
app          label  manage.py  pytest.ini  templates   user_auth
fundoo_notes logs   notes      results     transport   utils
(myenv) ubuntu@ip-172-31-86-131:~/Aws_test_fundoo/fundoo_notes$ python manage.py migrate
Operations to perform:
  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
```

Verify Deployment

Once the setup is complete, verify that your Django application is running correctly by accessing it via its public IP address or domain name.



Welcome, !


Instances | EC2 | us-east-1
Fundoo_notes
Home
Not secure
44.201.123.163:8000/swagger/
Swagger
Supported by SMARTBEAR
http://44.201.123.163:8000/swagger/?format=openapi
Explore

Fundoo_notes v1

`[ Base URL: 44.201.123.163:8000/ ]`

http://44.201.123.163:8000/swagger/?format=openapi

Test description

Terms of service

Contact the developer

BSD License

Schemes: HTTP