PostgreSQL

- sudo apt update
- sudo apt install postgresql postgresql-contrib
- sudo systemctl start postgresql
- sudo systemctl enable postgresql
- sudo systemctl status postgresql

```
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
     Active: active (exited) since Thu 2024-10-24 14:24:22 UTC; 14min ago
    Process: 699 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 699 (code=exited, status=0/SUCCESS)
        CPU: 2ms

Oct 24 14:24:22 DataBase systemd[1]: Starting PostgreSQL RDBMS...
Oct 24 14:24:22 DataBase systemd[1]: Finished PostgreSQL RDBMS.
~
~
```

- sudo -u postgres psql *optional(sudo -i -u postgres)*

```
ubuntu@DataBase:~$ 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=#
```

- psql
  - CREATE DATABASE fundoodb;
  - CREATE USER eran WITH PASSWORD '123456789';
  - GRANT ALL PRIVILEGES ON DATABASE fundoodb TO eran;
  - \q
  - Exit
- Find the line #listen_addresses = 'localhost' and change it to:
- sudo nano /etc/postgresql/12/main/postgresql.conf

- listen_addresses = '*'

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

# - 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)
```

- (This change allows PostgreSQL to listen for connections from any IP address, not just localhost.)
- sudo nano /etc/postgresql/16/main/pg_hba.conf

  host all all 0.0.0.0/0 md5

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

- sudo systemctl restart postgresql

- To access the Database from the Remote server
- psql -h your-database-instance-ip> -U myuser -d mydatabase
  *(install sudo apt install postgresql-client)

Backend Server:-

- cd /
- Sudo git clone -b dev https://github.com/Aniket26559/Aws_test.git

```
ubuntu@Backend:~$ ls -l /
total 68
drwxr-xr-x   5 eran eran  4096 Oct 23 11:36 Aws_test
lrwxrwxrwx   1 root root     7 Sep 27 02:09 bin -> usr/bin
```

- sudo apt update && sudo apt upgrade -y
- cd /Aws_test
- sudo apt-get install python3 python3-pip python3-dev libpq-dev build-essential python3 -m venv myenv
- source myenv/bin/activate

```
eran@Backend:/Aws_test$ source myenv/bin/activate
(myenv) eran@Backend:/Aws_test$
```

- pip install -r requirements.txt
- sudo nano /etc/fundoo/env.conf *(create a env.conf file to avoid Hard coding)

  DB_NAME=<database name>

  DB_USER=<username>

  DB_PASSWORD=<database password>

  DB_HOST=<data base ip >

  DB_PORT=5432

- sudo nano settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': os.environ['DB_NAME'],                  # Re
        'USER': os.environ['DB_USER'],                  # Re
        'PASSWORD': os.environ['DB_PASSWORD'],          # Re
        'HOST': os.environ['DB_HOST'],                  # Re
        'PORT': os.environ['DB_PORT'],                  # Re
    }
}
```

- python3 manage.py makemigrations
- python3 manage.py migrate

```
(myenv) eran@Backend:/Aws_test/fundoo_notes$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, django_celery_beat,
abel, notes, sessions, user_auth
Running migrations:
  No migrations to apply.
(myenv) eran@Backend:/Aws_test/fundoo_notes$
```

Run the server manually :- python3 manage.py runserver 0.0.0.0:8000

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

```
(myenv) eran@Backend:/Aws_test/fundoo_notes$ python3 manage.py runserver
 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 24, 2024 - 15:15:32
Django version 5.1, using settings 'fundoo_notes.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Create a service file for gunicorn:-

- sudo nano /etc/system/system/fundoo.service

```
[Unit]
Description=Fundoo Service
After=network.target

[Service]
User=eran
Group=eran
EnvironmentFile=/etc/fundoo/env.conf
WorkingDirectory=/Aws_test/fundoo_notes
ExecStart=/bin/bash -c 'source /Aws_test/fundoo_notes/myenv/bin/activate && exec gunicorn --
workers 3 --bind 0.0.0.0:8000 fundoo_notes.wsgi:appli>

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

```
[Unit]
Description=Fundoo Service
After=network.target

[Service]
User=eran
Group=eran
EnvironmentFile=/etc/fundoo/env.conf
WorkingDirectory=/Aws_test/fundoo_notes
ExecStart=/bin/bash -c 'source /Aws_test/fundoo_notes/myenv/bin/activate && exec gunicorn --workers 3 --bind 0.0.0.0:8000 fundoo_notes.wsgi:appli

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

- sudo systemctl daemon-reload
- sudo systemctl start fundoo.service
- sudo systemctl enable fundoo.service
- sudo systemctl status fundoo.service

```
(myenv) eran@Backend:~$ sudo systemctl status fundoo.service
● fundoo.service - Fundoo Service
     Loaded: loaded (/etc/systemd/system/fundoo.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-10-24 15:24:59 UTC; 5s ago
   Main PID: 1067 (gunicorn)
      Tasks: 4 (limit: 1130)
     Memory: 137.9M
        CPU: 1.784s
     CGroup: /system.slice/fundoo.service
             ├─1067 /Aws_test/fundoo_notes/myenv/bin/python3 /Aws_test/fundoo_notes/myenv/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 f
             ├─1068 /Aws_test/fundoo_notes/myenv/bin/python3 /Aws_test/fundoo_notes/myenv/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 f
             ├─1069 /Aws_test/fundoo_notes/myenv/bin/python3 /Aws_test/fundoo_notes/myenv/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 f
             └─1070 /Aws_test/fundoo_notes/myenv/bin/python3 /Aws_test/fundoo_notes/myenv/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 f

Oct 24 15:24:59 Backend systemd[1]: Started Fundoo Service.
Oct 24 15:24:59 Backend bash[1067]: [2024-10-24 15:24:59 +0000] [1067] [INFO] Starting gunicorn 23.0.0
Oct 24 15:24:59 Backend bash[1067]: [2024-10-24 15:24:59 +0000] [1067] [INFO] Listening at: http://0.0.0.0:8000 (1067)
Oct 24 15:24:59 Backend bash[1067]: [2024-10-24 15:24:59 +0000] [1067] [INFO] Using worker: sync
Oct 24 15:24:59 Backend bash[1068]: [2024-10-24 15:24:59 +0000] [1068] [INFO] Booting worker with pid: 1068
Oct 24 15:24:59 Backend bash[1069]: [2024-10-24 15:24:59 +0000] [1069] [INFO] Booting worker with pid: 1069
Oct 24 15:24:59 Backend bash[1070]: [2024-10-24 15:24:59 +0000] [1070] [INFO] Booting worker with pid: 1070
lines 1-20/20 (END)
```

Test the application in local host

- curl localhost:8000/home/

1. Setting Up Nginx for Frontend (Public Subnet)

- **Update your server**:
  sudo apt update && sudo apt upgrade -y
- **Install Nginx**:
  sudo apt install nginx -y
- **Start and enable Nginx** to ensure it runs on boot:
  sudo systemctl start nginx
  sudo systemctl enable nginx

```
ubuntu@ip-10-0-0-91:~$ sudo systemctl start nginx
ubuntu@ip-10-0-0-91:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-10-19 07:55:06 UTC; 4min 20s ago
       Docs: man:nginx(8)
    Process: 15618 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process o>
    Process: 15620 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=>
   Main PID: 15621 (nginx)
      Tasks: 2 (limit: 1130)
     Memory: 1.7M (peak: 1.9M)
        CPU: 12ms
     CGroup: /system.slice/nginx.service
             ├─15621 "nginx: master process /usr/sbin/nginx -g daemon on; master_pro>
             └─15622 "nginx: worker process"

Oct 19 07:55:06 ip-10-0-0-91 systemd[1]: Starting nginx.service - A high performance>
Oct 19 07:55:06 ip-10-0-0-91 systemd[1]: Started nginx.service - A high performance >

ubuntu@ip-10-0-0-91:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/s
ystemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
ubuntu@ip-10-0-0-91:~$ sudo nano /etc/nginx/sites-available/fundoo-frontend
ubuntu@ip-10-0-0-91:~$
```

- **Check Nginx** is working by visiting the instance's public IP in a browser (e.g.,http://your_public_frontend_ip). You should see the default Nginx welcome page.

➢ **Configure Nginx**:

You need to create a configuration file for your app. Let's assume you have a frontend directory with static files. Edit the Nginx configuration:

- sudo nano /etc/nginx/sites-available/fundoo-frontend

- Add this configuration:

```
server {
listen 80;
server_name _default;
location / {
include proxy_params;
proxy_pass http://10.0.2.177:8000;
    }
}
```

- In summary, this Nginx server block is set up to listen for HTTP requests on port 80 directed to your public IP address (13.233.143.26). When a user visits the root URL, Nginx will serve the index.html file located in the specified directory. This setup is typical for serving static websites or the frontend of web applications.
- A **configuration file** is a file used to define settings and preferences for a software application, service, or system. Instead of hard-coding options into the program, configuration files provide a way to customize the behavior of the software without modifying its source code. These files are often written in text format, making them easy to read and modify by both humans and programs.
- Configuration files allow flexible control over how an application behaves without modifying the underlying code.

2. **Link the configuration**:

- Sudo unlink /etc/nginx/sites-enabled/default (optional)
- sudo ln -s /etc/nginx/sites-available/fundoo-frontend /etc/nginx/sites-enabled/

Purpose of the Command

- In Nginx, the convention is to keep all site configuration files in the sites-available directory. Only the configurations in the sites-enabled directory are loaded by Nginx when it starts.

By running this command, you:

- Enable your Nginx configuration for your frontend application by creating a link to it in the sites-enabled directory.
- Keep your configurations organized, allowing you to easily manage and enable/disable different sites.

3. Restart Nginx:

```
sudo systemctl restart nginx
```

4. **Check the connection from Host**

- curl <backend ip>:8000/home/