Amar Bay Backend Deployment

1. **Update and install Python in system**

```
root@primary-node:/ecom# apt update -y
root@primary-node:/ecom# apt install -y
software-properties-common
root@primary-node:/ecom# add-apt-repository
ppa:deadsnakes/ppa
root@primary-node:/ecom# apt update -y
root@primary-node:/ecom# apt install -y python3.8
python3.8-venv python3.8-dev python3.8-distutils
```

2. **Download the project from Git**

```
root@primary-node:/ecom# git clone
https://github.com/ForhadPython/amarbay_backend.git
User:forhadPython
Password: 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
```

3. **Create Virtual Environment for amarbay_backend**

```
root@primary-node:/ecom# ls
amarbay_backend  amarbay_frontend
root@primary-node:/ecom# cd amarbay_backend/
root@primary-node:/ecom/amarbay_backend# python3.8 -m venv
venv3.8
root@primary-node:/ecom/amarbay_backend# source
venv3.8/bin/activate
(venv) root@primary-node:/ecom/amarbay_backend# pip
install -r requirements.txt
(venv) root@primary-node:/ecom/amarbay_backend# cd
amarbay/
```

Note:

If the database connection is not present in the "**settings.py**" file then add this or already added then just change the DB name, DB User, Password, Host and Port. Or if present then keep it remaining as it is. Finally add the allowed host.

4. Add the DB connection in Django "settings.py" file

```
(venv) root@primary-node:/ecom/amarbay_backend/amarbay#
vim settings.py

ALLOWED_HOSTS = ["192.168.144.128", "127.0.0.1"]

 'default': {
        'ENGINE':
'django.db.backends.postgresql_psycopg2',
        'NAME': 'amarbay_com',
        'USER': 'website',
        'PASSWORD': '@m@rb@yweb$1t',
        'HOST': 'localhost',
        'PORT': '5432',  # Default port for PostgreSQL
    }
```

5. Deactivate the Virtual Environment

```
(venv3.8) root@primary-node:/ecom/amarbay_backend/amarbay#
cd ..
(venv3.8) root@primary-node:/ecom/amarbay_backend# ls
amarbay  contact  customer  custompage  home  __init__.py
logo.png  manage.py  order  package.json
package-lock.json  product  ProductDetails.jsx  README.md
requirements.txt  venv3.8

(venv3.8) root@primary-node:/ecom/amarbay_backend#
deactivate
```

6. Database (Postgresql) Configuration

```
root@primary-node:/ecom/amarbay_backend# su - postgres
postgres@primary-node:~$ psql
postgres=#
postgres=# create database amarbay_com;
CREATE DATABASE

postgres=# create user website with password
'@m@rb@yweb$1t';
CREATE ROLE

postgres=# grant all privileges on database amarbay_com to
website;
GRANT

postgres=# GRANT ALL PRIVILEGES ON SCHEMA public TO
website;
GRANT

postgres=# ALTER USER website WITH SUPERUSER;
ALTER ROLE

postgres=# \q
```

Take a another tab into the terminal and test the DB connection

```
postgres@primary-node:~$ psql -h localhost -U website -d
amarbay_com
```

Note: If the DB is connected properly that means no issue from Django to connect the DB

7. Active the Virtual Environment and Migrate Python

```
root@primary-node:/ecom/amarbay_backend# source
venv3.8/bin/activate
(venv3.8) root@primary-node:/ecom/amarbay_backend# ls
amarbay  contact  customer  custompage  home  __init__.py
logo.png  manage.py  order  package.json
package-lock.json  product  ProductDetails.jsx  README.md
requirements.txt  venv3.8

(venv3.8) root@primary-node:/ecom/amarbay_backend# pip
install psycopg2-binary
(venv3.8) root@primary-node:/ecom/amarbay_backend# python
manage.py migrate
(venv3.8) root@primary-node:/ecom/amarbay_backend# python
manage.py collectstatic --noinput
(venv3.8) root@primary-node:/ecom/amarbay_backend# pip
install gunicorn
```

Test the Gunicorn Connection

```
(venv3.8) root@primary-node:/ecom/amarbay_backend#
gunicorn amarbay.wsgi:application
```

Note: If "**gunicorn**" is connected that means it working properly with the environment then press "**ctrl+c**"

8. Create Gunicorn Socket and Service File

Gunicorn Socket

```
root@primary-node:/home/rudra# vim
/etc/systemd/system/gunicorn.socket

[Unit]
Description=gunicorn socket for amarbay
[Socket]
ListenStream=/run/gunicorn.sock
SocketUser=rudra
SocketGroup=www-data
[Install]
WantedBy=sockets.target
```

Gunicorn Service

```
root@primary-node:/home/rudra# vim
/etc/systemd/system/gunicorn.service

[Unit]
Description=gunicorn daemon for amarbay
Requires=gunicorn.socket
After=network.target

[Service]
User=rudra
Group=www-data
WorkingDirectory=/ecom/amarbay_backend
ExecStart=/ecom/amarbay_backend/venv3.8/bin/gunicorn \
          --access-logfile - \
          --workers 3 \
          --bind unix:/run/gunicorn.sock \
          amarbay.wsgi:application

Restart=always

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

9. Enable and Start Gunicorn Service

```
root@primary-node:/home/rudra# systemctl daemon-reload
root@primary-node:/home/rudra# systemctl enable --now
gunicorn.socket
Created symlink
/etc/systemd/system/sockets.target.wants/gunicorn.socket →
/etc/systemd/system/gunicorn.socket.

root@primary-node:/home/rudra# systemctl start
gunicorn.socket
root@primary-node:/home/rudra# systemctl status
gunicorn.socket
● gunicorn.socket - gunicorn socket for amarbay
     Loaded: loaded (/etc/systemd/system/gunicorn.socket;
enabled; preset: enabled)
     Active: active (listening) since Sun 2025-09-21
16:37:04 +06; 20s ago
```

10. Web Server “Nginx” Installation and Configuration

```
root@primary-node:/home/rudra# apt install nginx -y
root@primary-node:/home/rudra# cd
/etc/nginx/sites-available/
root@primary-node:/etc/nginx/sites-available# vim
amarbay_backend

server {
    listen 80;
    server_name 192.168.144.128;

    # Django static files
    location /static/ {
        alias /ecom/amarbay_backend/static/root/;
        expires 30d;
    }

    location /media/ {
        alias /ecom/amarbay_backend/media/;
        expires 30d;
    }

    # Pass everything else to Gunicorn
    location / {
        proxy_read_timeout 36000;
        proxy_connect_timeout 36000;
        proxy_send_timeout 36000;
        include proxy_params;
        proxy_pass http://unix:/run/gunicorn.sock;
    }

    access_log /var/log/nginx/amarbay.access.log;
    error_log /var/log/nginx/amarbay.error.log;
}
```

Enable and Restart Nginx

```
root@primary-node:/etc/nginx/sites-available# rm -rf
default
root@primary-node:/etc/nginx/sites-available# nginx -t
root@primary-node:/etc/nginx/sites-available# ln -s
/etc/nginx/sites-available/amarbay_backend
/etc/nginx/sites-enabled/
root@primary-node:/etc/nginx/sites-available# systemctl
restart nginx
```

```
root@primary-node:/etc/nginx/sites-available# systemctl
status nginx
```

11. Go to The Browser

```
http://192.168.144.128/admin
```

======O Backend Deployment Done O======

FrontEnd Deployment

1. Download the project from Git

```
root@primary-node:/ecom# git clone
https://github.com/ForhadPython/amarbay_frontend.git
User:forhadPython
Password: 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
```

2. Make ready the environment for Frontend

```
root@primary-node:/ecom/amarbay_frontend# npm install
root@primary-node:/ecom/amarbay_frontend# npm run build
root@primary-node:/ecom/amarbay_frontend# npm run dev
```

Note: After running the "**npm run dev**" command, there will be a probability to get version related errors. The errors are as following

```
You are using Node.js 18.19.1. Vite requires Node.js version
20.19+ or 22.12+. Please upgrade your Node.js version.
error when starting dev server:
TypeError: crypto.hash is not a function

Error Means: We are using a backdated Node.js version.

To mitigate the issue you have to follow the bellow steps
```

3. **Check the node version and install the required version**

```
root@primary-node:/ecom/amarbay_frontend# node -v
v18.19.1
root@primary-node:/ecom/amarbay_frontend# ls
dist                 index.html    package.json       public
src
eslint.config.js  node_modules  package-lock.json
README.md  vite.config.js

root@primary-node:/ecom/amarbay_frontend# rm -rf
node_modules/
root@primary-node:/ecom/amarbay_frontend# ls
dist                 index.html    package-lock.json
README.md  vite.config.js eslint.config.js  package.json
public               src

root@primary-node:/ecom/amarbay_frontend# curl -fsSL
https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/insta
ll.sh | bash
root@primary-node:/ecom/amarbay_frontend# source ~/.bashrc
root@primary-node:/ecom/amarbay_frontend# nvm install 22
root@primary-node:/ecom/amarbay_frontend# node -v
V22.19.0

root@primary-node:/ecom/amarbay_frontend# npm -v
10.9.3

root@primary-node:/ecom/amarbay_frontend# rm -rf
package-lock.json
root@primary-node:/ecom/amarbay_frontend# npm install
root@primary-node:/ecom/amarbay_frontend# ls
dist                 index.html    package.json       public
src eslint.config.js   node_modules  package-lock.json
README.md  vite.config.js
root@primary-node:/ecom/amarbay_frontend# npm run build
root@primary-node:/ecom/amarbay_frontend# npm run dev
```

4. Configure Nginx for amarbay_frontend

```
root@primary-node:/etc/nginx/sites-available# vim
amarbay_frontend

server {
    listen 8080;
    server_name 192.168.144.128;

    root /ecom/amarbay_frontend/dist;
    index index.html;

    # Serve static assets with long cache
    location /assets/ {
        add_header Cache-Control "public,
max-age=31536000, immutable";
    }

    # Handle frontend routes (SPA fallback)
    location / {
        try_files $uri /index.html;
    }

    access_log /var/log/nginx/amarbay_frontend.access.log;
    error_log /var/log/nginx/amarbay_frontend.error.log;
}

root@primary-node:/etc/nginx/sites-available# ln -s
/etc/nginx/sites-available//amarbay_frontend
/etc/nginx/sites-enabled/

root@primary-node:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax
is ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful
```

```
root@primary-node:/etc/nginx/sites-available# systemctl
restart gunicorn
root@primary-node:/etc/nginx/sites-available# systemctl
restart gunicorn.socket
root@primary-node:/etc/nginx/sites-available# systemctl
status gunicorn
root@primary-node:/etc/nginx/sites-available# systemctl
status gunicorn.socket
```

```
root@primary-node:/etc/nginx/sites-available# systemctl
restart nginx
root@primary-node:/etc/nginx/sites-available# systemctl
status nginx
```

5. Go to The Browser

```
http://192.168.144.128:8080
```

======O Frontend Deployment Done O======