Deploy FastAPI application on Ubuntu + Nginx

Testing on Ubuntu Server 20.04 LTS

Steps:

1. setup nginx and python 3
2. creating virtual host
3. Create and set up fastapi app
4. install gunicorn
5. Create and set up fastapi app
6. Create the Server Block Files
7. Enable the New Server Block Files
8. Open a port and Permission
9. Testing

---

1. setup nginx and python 3

by default : installed Python 3 (python3)

```
sudo apt update
sudo apt-get install python3 python3-pip python-is-python3
```

*check

```
python3 --version
pip3 --version

sudo update-alternatives --config python
```

Install nginx

```
sudo apt-get install nginx
```

---

2. creating virtual host

Run

```
sudo mkdir -p /var/www/fastapi
sudo chown -R $USER:$USER /var/www/fastapi
```

```
sudo chmod -R 755 /var/www
```

---

3. Create and set up fastapi app

copy app

```
cd /var/www/fastapi
cp -a /home/agusk/fastapi/. .
```

install

```
pip install fastapi sqlalchemy python-multipart python-jose bcrypt
```

---

4. uvicorn and gunicorn

install

```
sudo pip3 install uvicorn gunicorn
```

test

```
gunicorn --bind 0.0.0.0:8000 -k uvicorn.workers.UvicornWorker
section5.main:app
```

5. Creating service

create a service file

```
sudo nano /etc/systemd/system/fastapi.service
```

++++++++++++++++++++++++++++++++++++++

```
[Unit]
Description=Running Python FastAPI on Ubuntu Server
After=network.target

[Service]
WorkingDirectory=/var/www/fastapi
```

```
ExecStart=/usr/local/bin/gunicorn --bind 0.0.0.0:8000 -k
uvicorn.workers.UvicornWorker section5.main:app
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=fastapi-app
User=agusk


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

++

** check

```
sudo systemd-analyze verify fastapi.service
```

** enable

```
sudo systemctl enable fastapi.service
```

start / stop

```
sudo systemctl start fastapi.service
sudo systemctl status fastapi.service
```

** stop/disable

```
sudo systemctl stop fastapi.service
sudo systemctl disable fastapi.service
```

check logs

```
journalctl -u fastapi.service
```

---

test

```
curl http://localhost:8000/users/
```

6. Create the Server Block Files

create a file

```
sudo nano /etc/nginx/sites-available/fastapi
```

/etc/nginx/sites-available/fastapi

```
server {
    listen  8083 default_server;
    listen [::]:8083 default_server;

    index index.html index.htm;

    location / {
        proxy_pass http://localhost:8000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_read_timeout 300s;
        proxy_connect_timeout 75s;
    }
}
```

7. Enable the New Server Block Files

```
sudo ln -s -f /etc/nginx/sites-available/fastapi /etc/nginx/sites-enabled/
```

**set permission (optional)

```
sudo setsebool -P httpd_can_network_connect 1
```

8. Open a port and Permission

open port 8083 ** open port on Azure VM

** restart

```
sudo nginx -t
sudo systemctl restart nginx
```

9. Test

http://:8083/users/