**Title** : **TedxNTUA(2020) Python Application Deployment**

**Project - Repo**: <https://github.com/TEDxNTUA/tedxntua2020>

**Objective**:

The objective is to deploy a python application (from repo) locally and then containarize the same.

**Theory** :

# **Project structure**

* docs/  
  Project-level documentation needs to be stored here and be written in Markdown. Module-specific documentation may be located only in source files.
* [bundles/](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/webpack/structure.md)
* assets/  
  Project-level CSS and JS scripts. For documentation on the bundling process, check [here](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/webpack/index.md).
* etc/  
  Extra files that accompany the project, such as configuration of external tools etc.
* project/
  + [about/](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/about/index.md)
  + [home/](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/home/index.md)
  + [partners/](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/partners/index.md)
  + [program/](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/program/index.md)
  + [team/](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/team/index.md)
  + static/  
    Project-level static files (e.g. logo) live here.
  + settings/  
    Settings module that reads from *.env* and exports configuration parameters to Django.
  + urls.py  
    Top-level routing instructions.
* media/  
  User-generated media files will be stored here. More info [here](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/static_management/index.md).
* static/  
  Static files (CSS, JS, images, etc.) from STATICFILES\_DIRS are gathered here. Do not place or edit any files in it, since all the management is done automatically by Django's collectstatic command. More info [here](https://github.com/TEDxNTUA/tedxntua2020/blob/master/docs/static_management/index.md).
* manage.py  
  Django command-line tool.
* env.sample  
  Example of a *.env* configuration file.
* \_version.py  
  Contains a \_\_version\_\_ variable to indicate the current version of the website. Calendar versioning is used.
* setup.py  
  Setup script. Dependencies must also be included here.
* requirements.txt  
  Pinned production dependencies.
* requirements-dev.txt  
  Pinned development dependencies.

**Installation**:

The project follows the [GoodCode.io](https://goodcode.io/articles/django-env-settings/) guide on storing configuration in the environment and the [12factor](https://12factor.net/) methodology. Thus, each deployment needs to have its own isolated environment and its own configuration, which shall be stored in the *.env* file.

## **Steps**

1. Setup and activate a virtual environment with python3.6. A useful guide on how to do this can be found [here](https://docs.python-guide.org/dev/virtualenvs/" \l "lower-level-virtualenv).
2. Install a MySQL client library for your system (sudo apt install default-libmysqlclient-dev for Linux).
3. Run pip install -e . if in a production environment, or pip install -e .[dev] if in a development environment.
4. Copy *env.sample* to *.env* and edit *.env* to customize the configuration for your local deployment.
5. Using MySQL for the database is highly recommended. Create a database and a user and write the credentials to .env in the form of DATABASE\_URL=mysql://user:pass@host:port/dbname (omit :port to use the default MySQL port).

# Webpack: Installation

## Steps

1. [Install NodeJS](https://nodejs.org/en/). This will also install Node's package manager, npm.
2. Navigate to the bundles/ directory:  
   cd bundles/
3. Run npm install.

**Implementation (Practical) :**

Step 1: setup the virtual environment [https://docs.python-guide.org/dev/virtualenvs/#lower-level-virtualenv](https://docs.python-guide.org/dev/virtualenvs/" \l "lower-level-virtualenv)

**sudo virtualenv venv**

**source venv/bin/activate**

Step 2: Install a MySQL client library for your system ( for Linux).

**sudo apt install default-libmysqlclient-dev**

Step 3: **pip install -r requirements.txt**

Step 3: **Run pip install -e .** if in a production environment, or pip install -e .[dev] if in a development environment.

Step 4: Copy env.sample to .env and edit .env to customise the configuration for your local deployment.

Step 5: Using MySQL for the database is highly recommended. Create a database and a user and write the credentials to .env in the form of DATABASE\_URL=mysql://user:pass@host:port/dbname (omit :port to use the default MySQL port).

**Database\_Url=mysql://admin:hadiyaproduct@hadiya-1.cdee8uw0q3dt.us-east-1.rds.amazonaws.com:3306/hadiya**

Step 6:

**Webpack: Installation**

1. [Install NodeJS](https://nodejs.org/en/). This will also install Node's package manager, npm.
2. Navigate to the bundles/ directory:  
   cd bundles/
3. Run npm install.

Step 7: python manage.py makemigrations

Step 8: python manage.py migrate

Step 9: python manage.py collectstatic

Step 10: python manage.py runserver 0.0.0.0:8000  
  
Step 11: python freeze > requirements1.txt

**Containerization Steps:**

**Dockerfile:**

FROM python:3.11

WORKDIR /app

COPY requirements1.txt .

RUN pip install pip --upgrade

RUN apt-get update && apt-get install -y \

libgirepository1.0-dev \

&& rm -rf /var/lib/apt/lists/\*

RUN pip install --no-cache-dir -r requirements1.txt

COPY . .

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

**Requirement.txt:**

distro==1.7.0

dj-database-url==0.5.0

Django==2.2.10

django-active-link==0.1.6

django-extensions==2.2.1

django-parler==2.0

django-versatileimagefield==1.10

django-webpack-loader==0.6.

gunicorn==21.2.0

Markdown==3.3.4

MarkupSafe==2.0.1

martor==1.6.14

more-itertools==8.10.0

mysqlclient==1.4.4

packaging==23.2

paramiko==3.4.0

pexpect==4.8.0

Pillow==5.4.1

platformdirs==4.1.0

ptyprocess==0.7.0

pyasn1==0.4.8

pyasn1-modules==0.2.1

pycparser==2.21

Pygments==2.11.2

PyGObject==3.42.1

PyHamcrest==2.0.2

PyJWT==2.3.0

PyNaCl==1.5.0

pyOpenSSL==21.0.0

pyparsing==2.4.7

pyrsistent==0.18.1

pyserial==3.5

python-dateutil==2.8.1

python-debian==0.1.43

python-dotenv==0.19.2

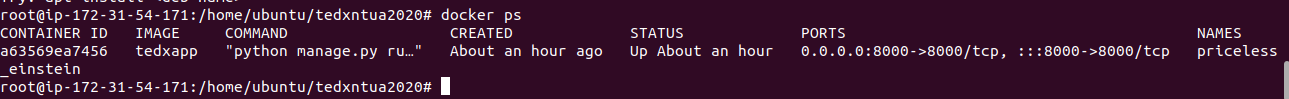
python-magic==0.4.24

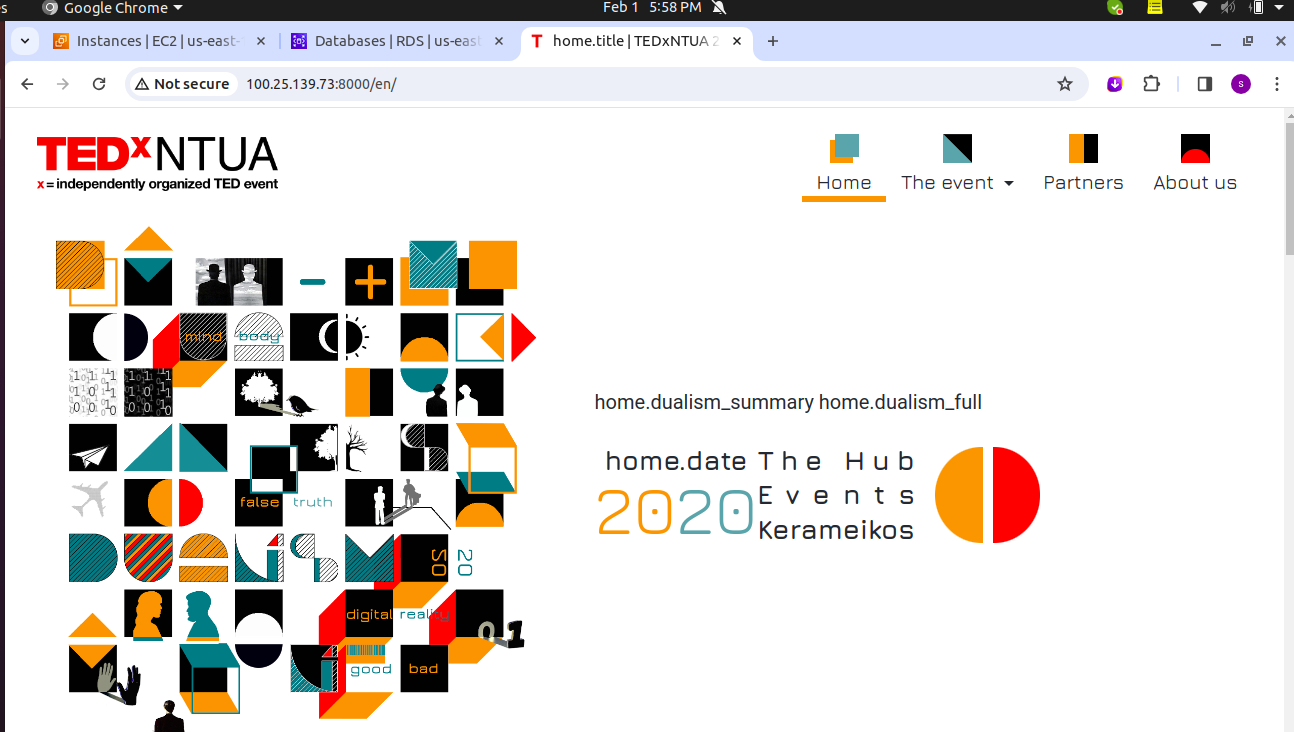
sqlparse==0.4.4

docker build -t tedxapp .

Docker run -itd -p 8000:8000 tedxapp

Need to open 8000 port on ec2





FAQS For Interview: