NC3 E-Learning Platform

Release 0.1.0

NC3-LU

Dec 18, 2024

Technical considerations

1 Prerequisites

1.1 Software

Generally speaking, requirements are the following:

- A GNU/Linux distribution. Tested on Ubuntu 22.04 LTS and 24.04 LTS ;
- Python version >= 3.10.x ;
- A PostgreSQL server for persistent storage. Tested with PostgreSQL 15.3 and 15.5;

For the Web server you can use Gunicorn, uWSGI, Apache or Nginx.

1.2 Hardware

The Django application is designed to operate efficiently, and it can run seamlessly on a Raspberry Pi when paired with Gunicorn and either Nginx or Apache to handle request proxying. It is advisable to allocate ample memory and disk space, particularly for the database, especially when it shares the same server. This proactive approach ensures smoother performance and mitigates potential resource constraints.

A decent configuration for a server would be:

- number of vCPU: 4;
- RAM (GB): 4;
- HDD (GB): 20.

The application will function seamlessly with these settings. Moreover, these values are relatively low when considering the capacity of modern servers.

1.3 Network

The deployment on the different servers requires an Internet connection since the updates are retrieved from the GitHub repository.

2 Installation

This section covers the installation steps of the sofware.

2.1 System packages

```
$ sudo apt install gettext curl npm libpango-1.0-0 libpangoft2-1.0-0
```

2.2 Poetry

```
$ curl -sSL https://install.python-poetry.org | python3 -
```

at the end of the *~/.bashrc* file add the line:

```
$ export PATH="/root/.local/bin:$PATH"
```

2.3 PostgreSQL

Install PostgreSQL, the version provided by default for your GNU/Linux distribution.

```
$ sudo apt-get install postgresql
```

Create a database, database user:

```
$ sudo -u postgres createuser <username>
$ sudo -u postgres createdb <database>
$ sudo -u postgres psql
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.
postgres=# alter user <username> with encrypted password '<password>';
ALTER ROLE
postgres=# grant all privileges on database <database> to <username>;
GRANT
postgres=# alter database <database> owner to <username>;
ALTER DATABASE
```

2.4 E-Learning Platform

```
git clone https://github.com/NC3-LU/eLearning.git
cd eLearning
npm install
```

Copy the config and adjust the DB connection and the other settings:

```
cp elearning/config_dev.py elearning/config.py
poetry install
poetry shell
python manage.py migrate
python manage.py collectstatic
poetry manage.py compilemessagess
```

2.4.1 Theme

In this case, the theme (static and templates) of the sofware will be cloned into the `theme` folder. You can replace it by your own. Currently one theme is available:

- https://github.com/NC3-LU/eLearning_daaz_theme (default theme, DAAZ Theme)

Exemple:

```
cd eLearning
git clone https://github.com/NC3-LU/eLearning_daaz_theme theme
```

2.4.2 Configuration

In the configuration file `elearning/config.py` , ensures that you have configured:

- `PUBLIC_URL`
- `ALLOWED_HOSTS`
- `SITE_NAME`
- `DATABASES`
- `HASH_KEY` and `SECRET_KEY`
- `DEBUG`: must be set to `False` in a production environment
- etc.

You **must really** set **your** secret keys.

Here is an example for the Fernet hash key (`HASH_KEY`):

```
$ python -c 'from cryptography.fernet import Fernet; print(Fernet.generate_key())'
b'Xaj5lFGAPiy2Ovzi4YmlWh-s4HHikFV4AswilOPPYN8='
```

For the Django secret key (`SECRET_KEY`), you can for example do:

```
$ python -c 'import secrets; print(secrets.token_hex())'
9cf5c7b13e469e6f6a9403b33410589031cfe927df6471a1cbdef1d4deb57c37
```

2.5 Populating the DB

If you want to populate the database with an existing data, see the *Populating or updating the DB from json file* (page 7) documentation

2.6 Launch the Django application

```
poetry run python manage.py runserver 127.0.0.1:8000
```

Of course, do not do that for a production environment.

2.7 Apache

The `mod_wsgi` package provides an Apache module that implements a WSGI compliant interface for hosting Python based web applications on top of the Apache web server. Install Apache and this module.

```
$ sudo apt install apache2 libapache2-mod-wsgi-py3
```

Note

Only in the case you can not use the version of mod_wsgi from your GNU/Linux distribution:

```
$ sudo apt install apache2 apache2-dev # apxs2
$ wget https://github.com/GrahamDumpleton/mod_wsgi/archive/refs/tags/5.0.0.tar.gz
$ tar -xzvf 5.0.0.tar.gz
$ cd mod_wsgi-5.0.0/
$ ./configure --with-apxs=/usr/bin/apxs2 --with-python=/home/<user>/.pyenv/shims/
↪python
$ make
$ sudo make install
```

Then in `/etc/apache2/apache2.conf` add the lines:

```
LoadFile /home/<user>/.pyenv/versions/3.11.0/lib/libpython3.11.so
LoadModule wsgi_module /usr/lib/apache2/modules/mod_wsgi.so
```

Restart Apache:

```
sudo systemctl restart apache2.service
```

For the next steps you must have a valid domain name.

2.7.1 Example of VirtualHost configuration file

Modify the <install_path>, <user>, <virtualenv_path> tags as appropriate

For <virtualenv_path> check using :

```
$ cd <install_path>
$ poetry env info
```

```
<VirtualHost *:80>
    ServerAdmin info@nc3.lu
    ServerName elearning.nc3.lu
    DocumentRoot /var/www/html
    RewriteEngine on
    RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin info@nc3.lu
    ServerName elearning.nc3.lu
    DocumentRoot <install_path>
    WSGIDaemonProcess elearning python-path=<install_path> python-home=<virtualenv_path>/
↪lib/python3.10/site-packages/
    WSGIProcessGroup elearning
    WSGIScriptAlias / <install_path>/elearning/wsgi.py

    <Directory "<install_path>/elearning/">
        <Files "wsgi.py">
            Require all granted
        </Files>
        WSGIApplicationGroup %{GLOBAL}
        WSGIPassAuthorization On

        Options Indexes FollowSymLinks
        Require all granted
    </Directory>

    Alias /static <install_path>/elearning/static
    <Directory <install_path>/static>
        Require all granted
    </Directory>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/elearning.nc3.lu_access.log combined
    ErrorLog ${APACHE_LOG_DIR}/elearning.nc3.lu_error.log

    # Let's Encrypt configuration
    SSLCertificateFile /etc/letsencrypt/live/elearning.nc3.lu/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/elearning.nc3.lu/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf
```

(continues on next page)

(continued from previous page)

```
</VirtualHost>
```

Then configure HTTPS properly. If you want to use Let's Encrypt:

```
sudo apt install certbot python3-certbot-apache
sudo certbot certonly --standalone -d elearning.nc3.lu
sudo a2enmod rewrite
sudo systemctl restart apache2.service
```

Verify that the certificate will be automatically updated:

```
$ cat /etc/letsencrypt/renewal/elearning.nc3.lu.conf
# Options used in the renewal process
[renewalparams]
account = <-account-id->
authenticator = apache
server = https://acme-v02.api.letsencrypt.org/directory
```

3 Updating the application

3.1 Run update bash script

All you have to do is:

<install_path> for exemple : /home/<user>/eLearning

```
$ cd <install_path>
$ ./scripts/update.sh -a
```

Usage: ./scripts/update.sh --help

```
$ ./scripts/update.sh --help
Usage: scripts/update.sh [options]
Options:
-u, --update-repositories   Update git repositories
-npm, --update-npm-packages Update npm packages
-p, --update-python-packages Update python packages
-m, --migrate-database      Migrate database
-c, --compile-translations  Compile translations
-s, --collect-static        Collect static files
-a, --update-all            Update all components
--help                      Display this help message
```

3.2 Or manually

```
$ cd <install_path>/theme/
$ git pull origin master
$ cd ..
$ git pull origin master
$ npm install
$ poetry install
$ poetry shell
```

(continues on next page)

(continued from previous page)

```
$ python manage.py migrate
$ python manage.py compilemessages
$ python manage.py collectstatic
```

3.3 Restart Apache server

Finally, restart Apache:

```
$ sudo systemctl restart apache2.service
```

4 Populating or updating the DB from json file

4.1 Dumping an existing database

Run the follow script

<install_path> for exemple : /home/<user>/eLearning

```
$ cd <install_path>
$ ./scripts/dump_data.sh
```

4.2 Loadind data from json file

Run the follow script

<install_path> for exemple : /home/<user>/eLearning

```
$ cd <install_path>
$ ./scripts/load_data.sh
```

5 Presentation

The E-Learning Platform is Django-based project provides an extensive platform for online learning management.

It integrates various features essential for e-learning systems including user management, course creation, and interactive learning tools.