WEB PROJECT
Deliverable 1

Alejandro Clavera Poza

Jordi Rafael Lazo Florensa

Guillem Arbiol López de Zamora

Didac Colominas Abalde

Miguel Ángel Barraza

Pau Escolà Barragán

10 d'abril de 2021

Computer Engineering Degree

Universitat de Lleida
Escola Politècnica Superior

1. Repository URL

https://github.com/alejandroclavera/web-project

2. Proposal to deploy the solution using multiple servers in a n-layer (web, application, database)

1.1 Number and function of servers

The application will consist of:

- A front end NGINX web server that will be in charge of receiving and making user requests and communicating directly with the Django application.
- Three GUNICORN servers, each running an instance of the django application.
- A MEMCACHE server for the database.
- A DATABASE server, which runs PostgresSQL.
- A REDIS server that stores user sessions.

Regarding the gunicorn servers, as they are scalable, if necessary the number of these can be increased in the future.

1.2 Connections and dependences amongst them

The client makes an HTTP request and this is managed by the NGINX server, which transmits it to the gunicorn servers that are running the django application. REDIS is connected to the django framework and it stores user sessions. The django server accesses the DATABASE and once the queries are made, they are stored in the MEMCACHE for faster access in future requests.

1.3 Optional and obligatory servers

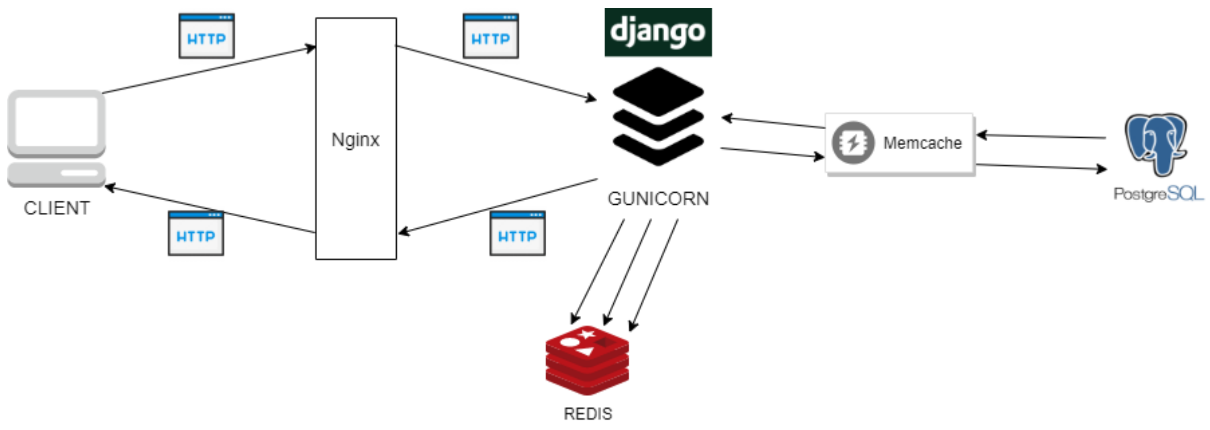
The states that are obligatory are:

- NGINX server.
- GUNICORN server.
- DATABASE server.
- REDIS server.

The states that are optional:

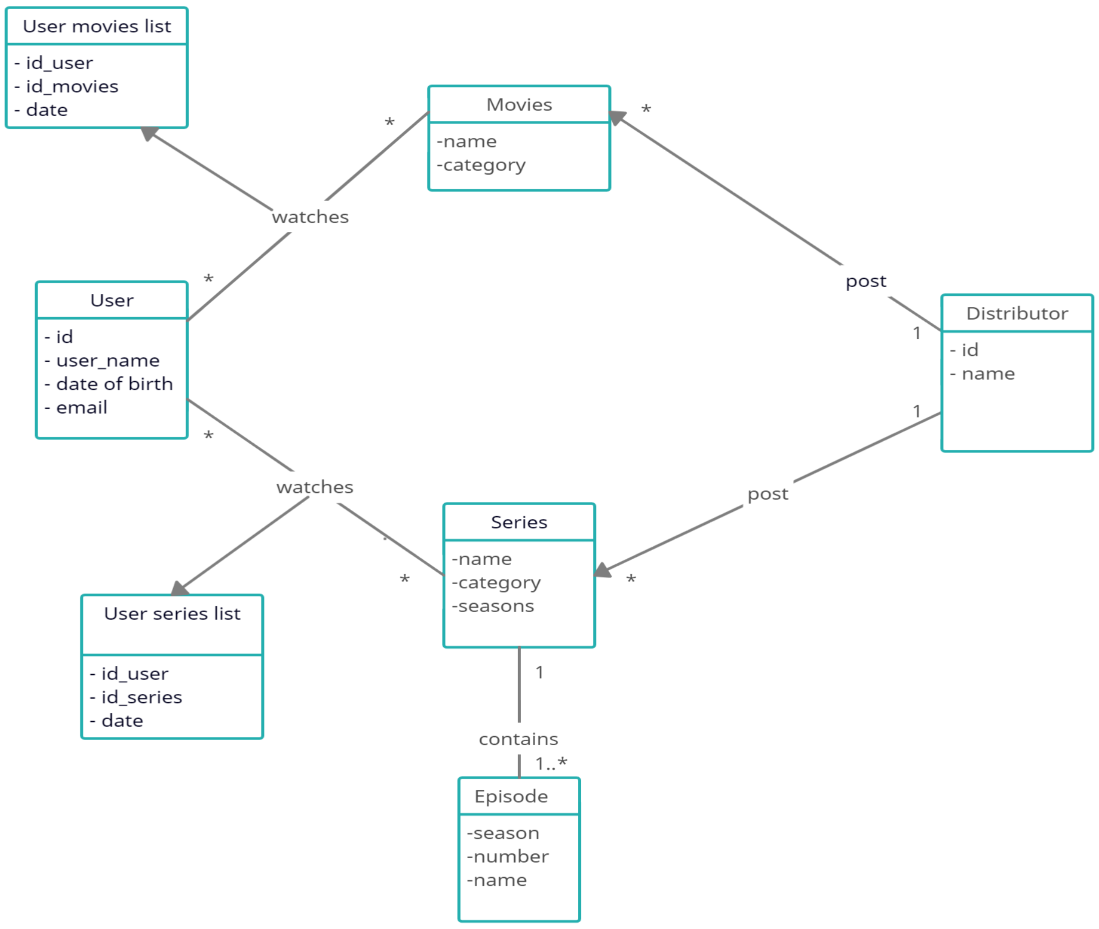
- MEMCACHE server.

1.4 Scheme of the deployment



3. Changes we have made in the project proposal

When creating the models in django, the data model design delivered in the project proposal has been completed by adding the relationship called UserSeriesList that stores the list of User and UserMoviesList series, which store the list of movies that the user is watching.



On the other hand, it has been decided to use the User model implemented by django to facilitate the work when authenticating the user.

4.Deplyoment schema for heroku and docker

You can find the instructions of how to deploy the application on heroku and how to run the application as a docker container orchestration using docker-compose in the **readme.md**.