



Cloud Final Project



2021/2022

Contents

| | |
|--|---|
| Use cases..... | 2 |
| A customer at the frontend can do the following: | 2 |
| Mandatory | 2 |
| Bonus | 2 |
| Docker environment specification:..... | 3 |
| The following is exposed to internet: | 3 |
| The following is defined in an internal network: | 3 |
| Restrictions on the used technologies:..... | 3 |
| Submission | 3 |
| Reading material | 3 |

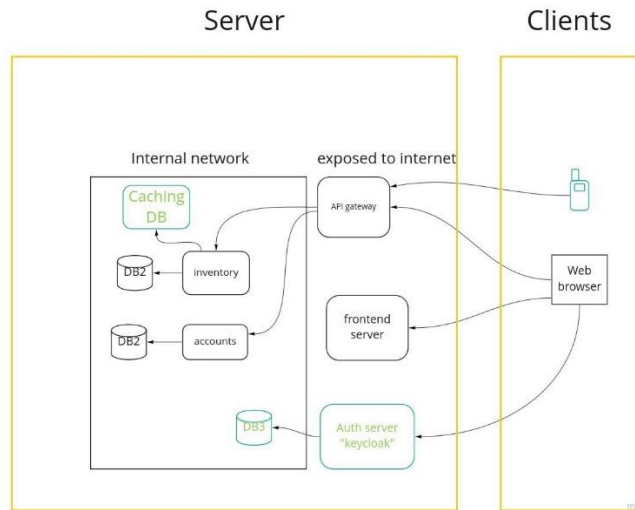


Figure 1 Microservice diagram for the required website

Use cases

- Navigation Bar
- Footer
- Home

A customer at the frontend can do the following:

- Register for a new account Sign up Page
- Login to an existing account Login Page
- Get his/her account information Profile Page
- Get a list of all products in the database (with filtering) Categories and ordering by price
- Search for a product One Code In Navigation Bar
- Get the details of a certain product Product Page
- Add a product to the cart Cart Page
- Checkout Cart Page

Mandatory

- Frontend server,
- API gateway,
- Inventory service,
- Account service, and
- Uploading the projects on Amazon Web Services: Elastic BeanStalk (AWS EB)

Bonus

- Mobile application that covers all the above use cases (Flutter only):
- Caching database
- Authentication server
- AI recommendation service

Docker environment specification:

The following is exposed to internet:

- API gateway
- Frontend server
- Authentication server

The following is defined in an internal network:

- Inventory service
- Accounts service
- Caching database

You may add any extra services you need.

Restrictions on the used technologies:

- Mobile application: Flutter
- Web application: React
- Frontend server: any
- API gateway: any
- Authentication server: any
- Service implementation: Node.js, Python, PHP, Java, Golang, Dart
- Database allowed: MongoDB, NoSQL DB, MySQL, PostgreSQL, MariaDB
- AI recommendation service: any
- Caching database: any

Submission

Provide a public GitHub repository that contains everything: the used images (**Dockerfiles**), **docker-composed-yaml file**, any source codes, and the link to your website.

Reading material

<https://www.nginx.com/blog/building-microservices-using-an-api-gateway/>