

Exercise for the course Cloud Computing Basic.

This is the exercise for Prof. Taffoni and Ruggero of the 2023/2024 Cloud Computing course.

Version 1.0 : This document can be modified several times in the next few days to improve the clarity of the information and provide a better understanding of what we are asking.

Rules

- Exercise should be done individually: no group, please!
- Materials (code/scripts/pictures and final report) should be prepared on a GitHub repository, starting with this one and sharing it with the teachers.
- A report should be sent by e-mail to the teachers at least five days in advance: the name of the file should `YOURSURNAME_report.pdf`
- Results and numbers of the exercises should be presented (also with the help of slides** in a maximum 10-minute presentation: this will be part of the exam). A few more questions on the topic of the courses will be asked at the end of the presentation.

deadlines

You should send us the e-mail at least one week before the exam. For the first two scheduled "appelli" this means:

- exam scheduled at 1.02.2024 **deadline 28.01.2023 at midnight**
- exam scheduled at 23.02.2024 **deadline 20.02.2023 at midnight**

The report should clearly explain which software stack we should use to deploy the developed infrastructure and run all the programs you used in your exercises. Providing well-done Makefiles/Dockerfiles/scripts to automatize the work is highly appreciated.

The exercise: Cloud-Based File Storage System

You are tasked with identifying, deploying, and implementing a cloud-based file storage system. The system should allow users to upload, download, and delete files. Each user should have a private storage space. The system should be scalable, secure, and cost-efficient. Suggested solutions to use for the exam are Nextcloud and MinIO.

Requirements

The deployed platform should be able to:

Manage User Authentication and Authorization:

- Users should be able to sign up, log in, and log out.
- Users should have different roles (e.g., regular user and admin).
- Regular users should have their private storage space.
- Admins should have the ability to manage users.

Manage File Operations:

- Users should be able to upload files to their private storage.
- Users should be able to download files from their private storage.
- Users should be able to delete files from their private storage.

Address Scalability:

- Design the system to handle a growing number of users and files.
- Discuss theoretically how you would handle increased load and traffic.

Address Security:

- Implement secure file storage and transmission.
- Discuss how you would secure user authentication.
- Discuss measures to prevent unauthorized access.

Discuss Cost-Efficiency:

- Consider the cost implications of your design.
- Discuss how you would optimize the system for cost efficiency.

Deployment:

- Provide a deployment plan for your system in a containerized environment on your laptop based on docker and docker-compose.
- Discuss how you would monitor and manage the deployed system.
- Choose a cloud provider that could be used to deploy the system in production and justify your choice.

Test your infrastructure:

- Consider the performance of your system in terms of load and IO operations

Submission details

Documentation:

- Submit a detailed design document explaining your choices and describing the platform's architecture, including components, databases, and their interactions.
- Include a section on the security measures taken.

Code:

- Submit the Docker files and any code eventually developed/modified for your cloud-based file storage system.
- Include a README file with instructions on how to deploy and use your system.

Presentation:

- Prepare a short presentation summarizing your design, implementation, and any interesting challenges you faced.
- Be ready to answer questions about your design choices and on the topics discussed during the Cloud Course Lectures

Evaluation Criteria

- Design Clarity: Is the system design well-documented and clear?
- Functionality: Does the system meet the specified requirements?
- Scalability: How well does the system handle increased load? How does the system perform on small files (a few KB), large files (GBs), and average (MBs)
- Security: Are appropriate security measures implemented?

- Cost-Efficiency: Has the student considered cost implications and optimized the system accordingly?