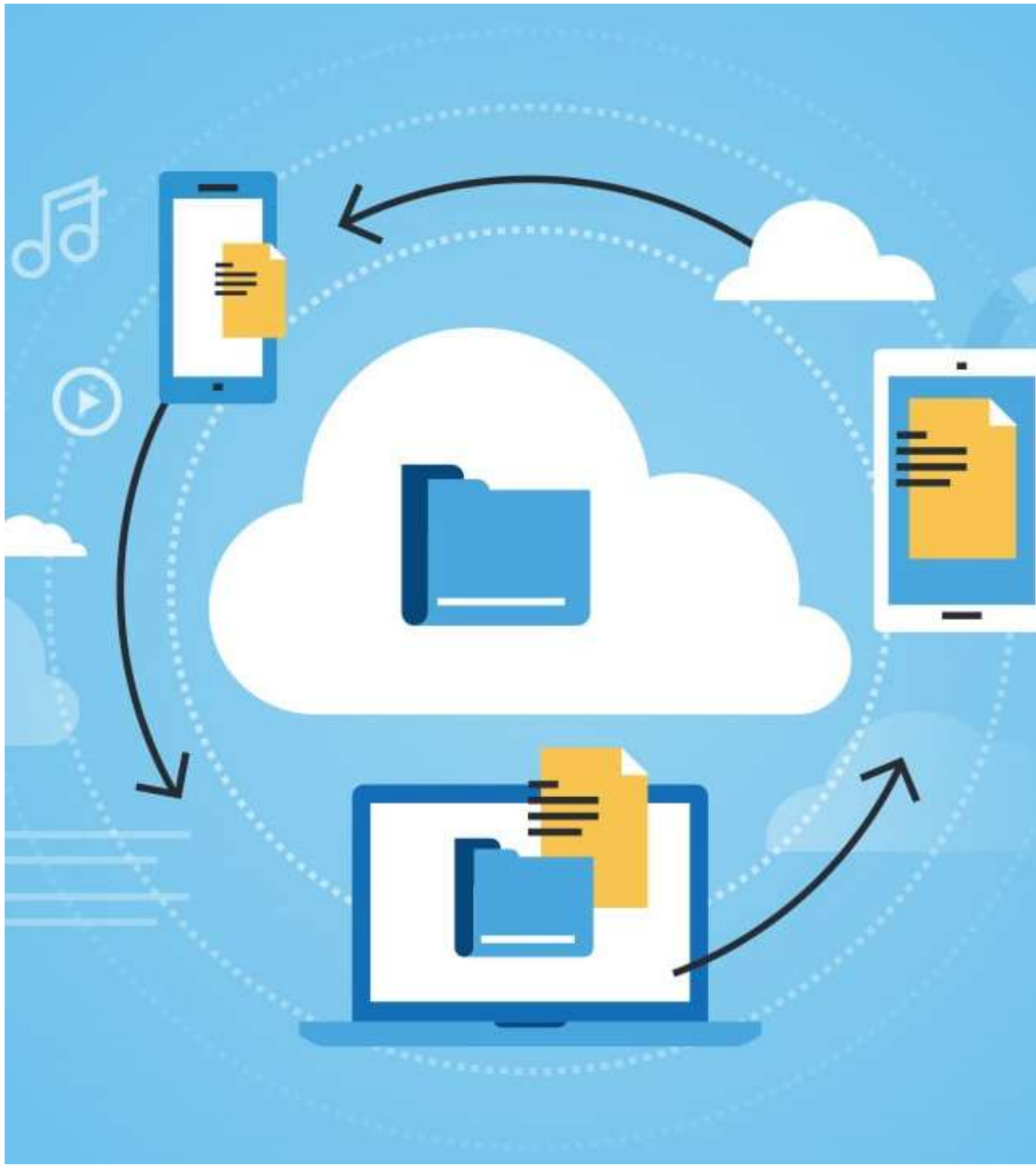
CLOUD COMPUTING FINAL PROJ ECT

CLOUD BASED FILE STORAGE SYSTEM

Marta Lucas (SM800043)

INTRODUCTION

This project consist in the deployment and implementation of a cloud-based file storage system.



DESIGN AND DEPLOYMENT

Nextcloud is an open-source cloud storage solution that allows users to save and share files in a private storage space.

It has a user-friendly interface and several built-in security features.

MariaDB is a relational database management system that is fully open-source and known for its performances and scalability.

In this project it’s used as Nexcloud’s backend database.

DESIGN AND DEPLOYMENT

Nextcloud and MariaDB were deployed with docker compose through a .yml file.

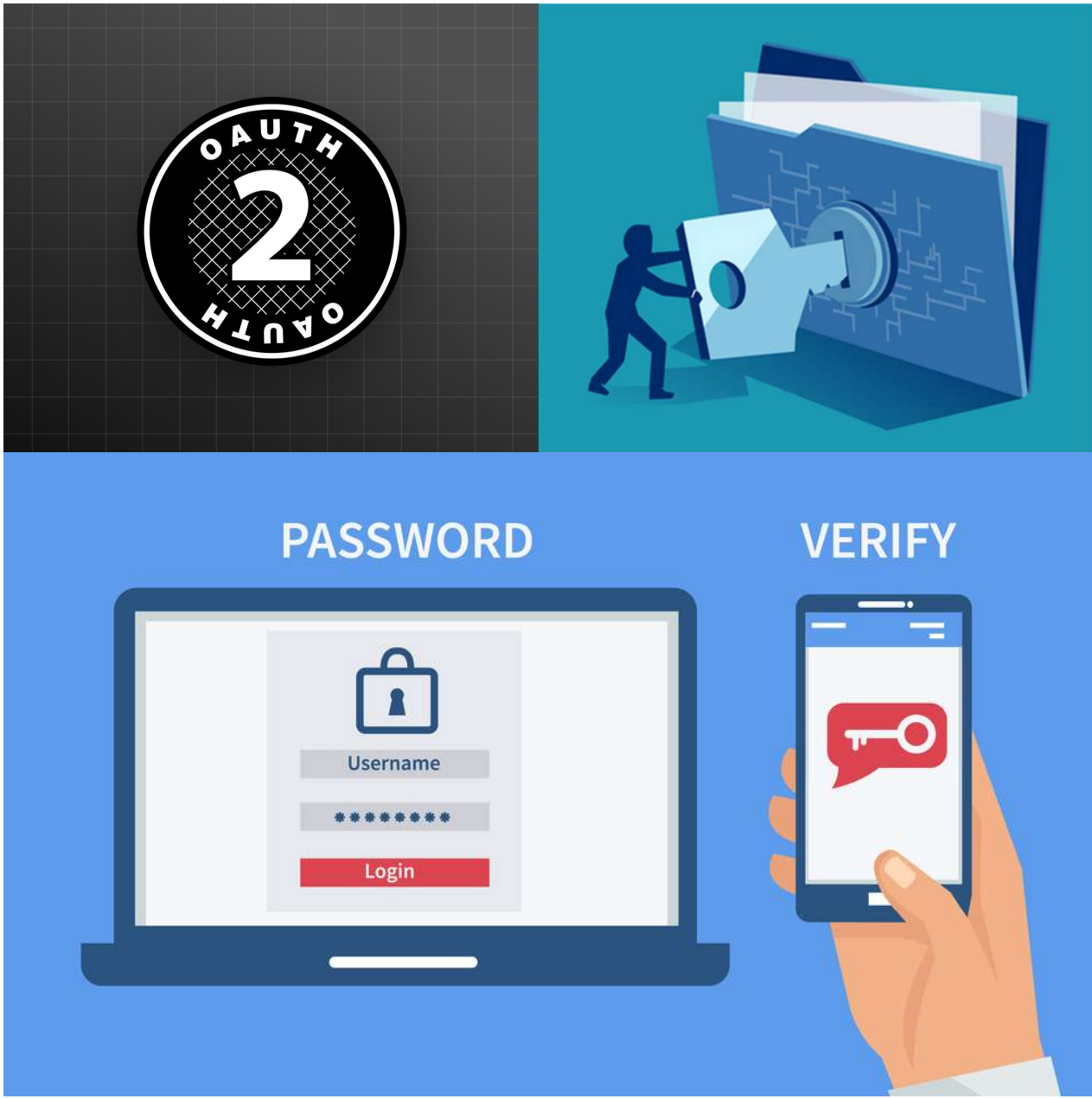
This tool allows to handle multiple containers at once.

Two different volumes were saved to persist data when containers are removed or stopped.

SECURITY

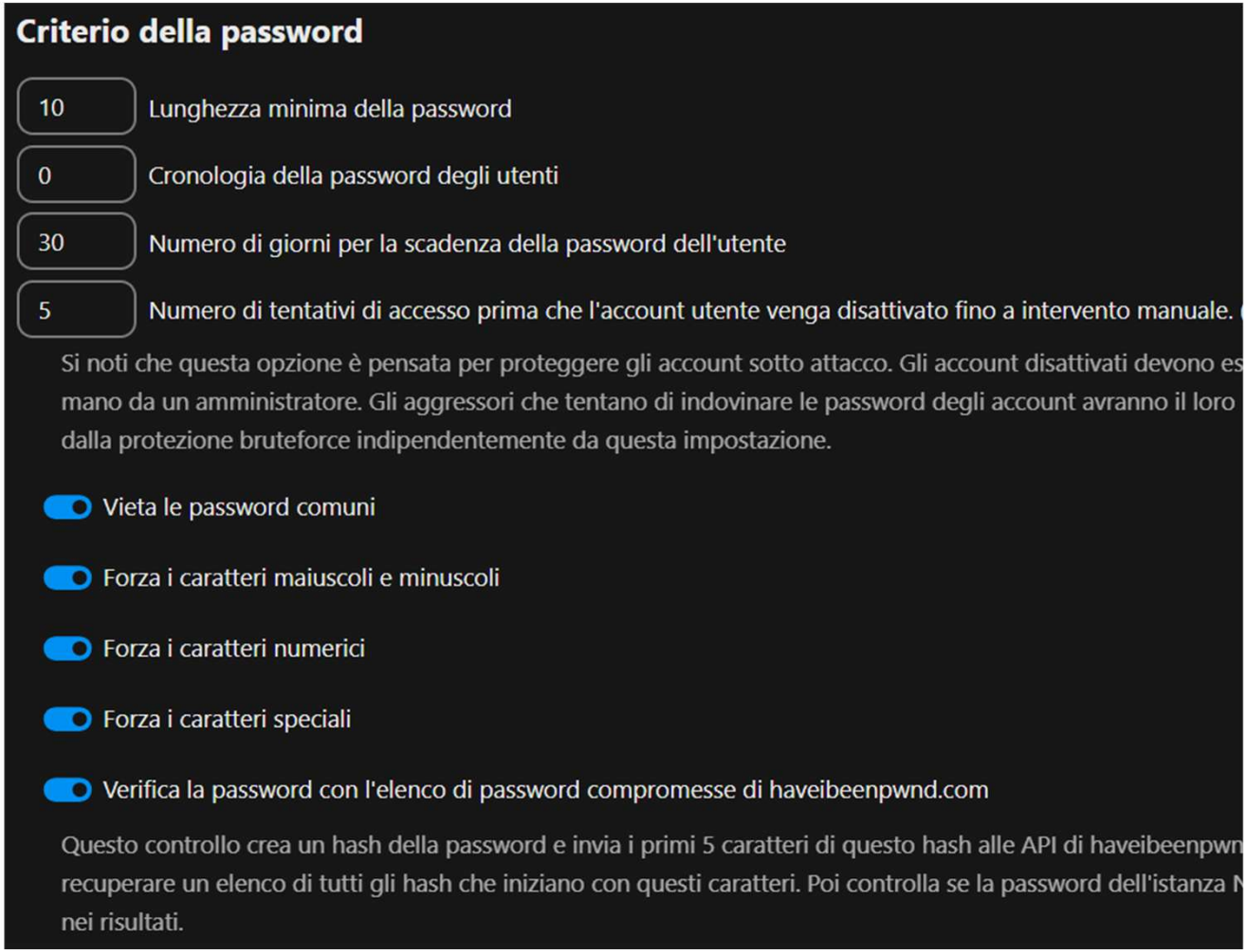
The following features are available on Nextcloud, but weren't activated:

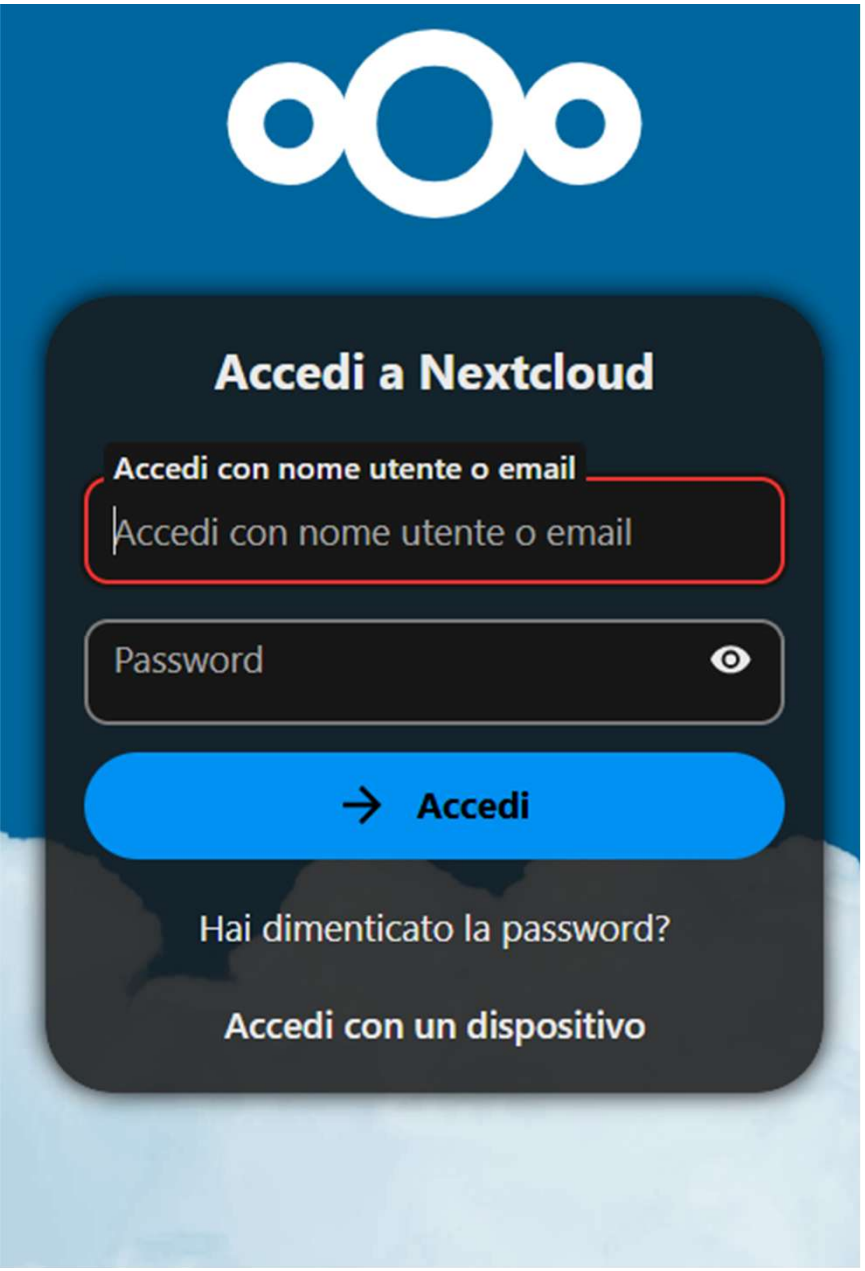
- Two factor authentication
- OAuth 2.0
- Server side encryption



PASSWORD POLICY

- Minimum password length: 10 characters
- Uppercase and lowercase letters
- At least one number
- At least one special character
- Failed login attempts before account is blocked: 5
- Password expiration: 30 days





USER MANAGEMENT

Scripts:

- create_user.sh: Creates 100 test users
- delete_files.sh: Deletes files in each test user's folder
- delete_user.sh: Deletes the test users

Access:

Username : Admin password: SecureAdmin15$

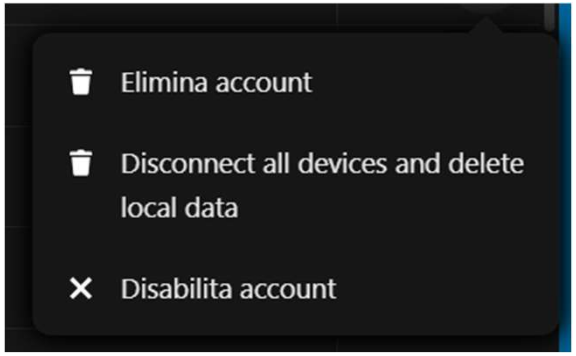
Username: test_userX password: Test_passwordX!

	Nome visualizzato Nome utente	Password	Posta elettronica	Gruppi
A	**admin** admin			admin
T1	**TestUser 1** test_user1			Users
T1	**TestUser 10** test_user10			Users
T1	**TestUser 100** test_user100			Users
T1	**TestUser 11** test_user11			Users
T1	**TestUser 12** test_user12			Users
T1	**TestUser 13** test_user13			Users
T1	**TestUser 14** test_user14			Users
T1	**TestUser 15** test_user15			Users
T1	**TestUser 16** test_user16			Users
T1	**TestUser 17** test_user17			Users

USER MANAGEMENT

ADMIN

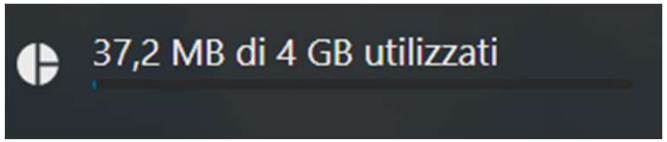
- Privileges:
  - Setting security measures
  - Setting user quotas
  - Disabling suspicious accounts
  - Managing user groups
  - Creating/Deleting users



USER MANAGEMENT
TEST USERS

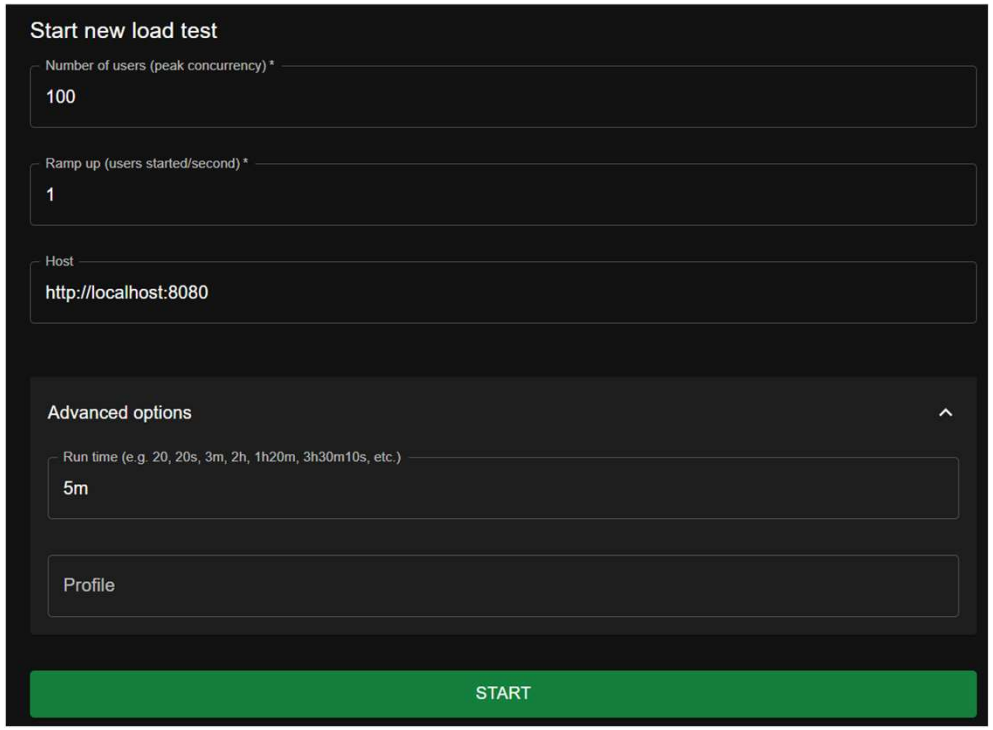
Privileges:
- Upload/download files to/from storage
- Change personal interface settings
- Change password/personal information

Storage quota: 4GB



LOCUST TESTING

An open source tool to simulate concurrent user behaviour



Five minute tests, with spawn rate of 1 user per second

Light test: 1 KB 80 users task frequency 2-4 s

Medium test: 1 KB and 1 MB 80 users
task frequency 1-2 s

Heavy test: 1 GB, 1 KB, 1 MB 10 users
task frequency 2-4 s

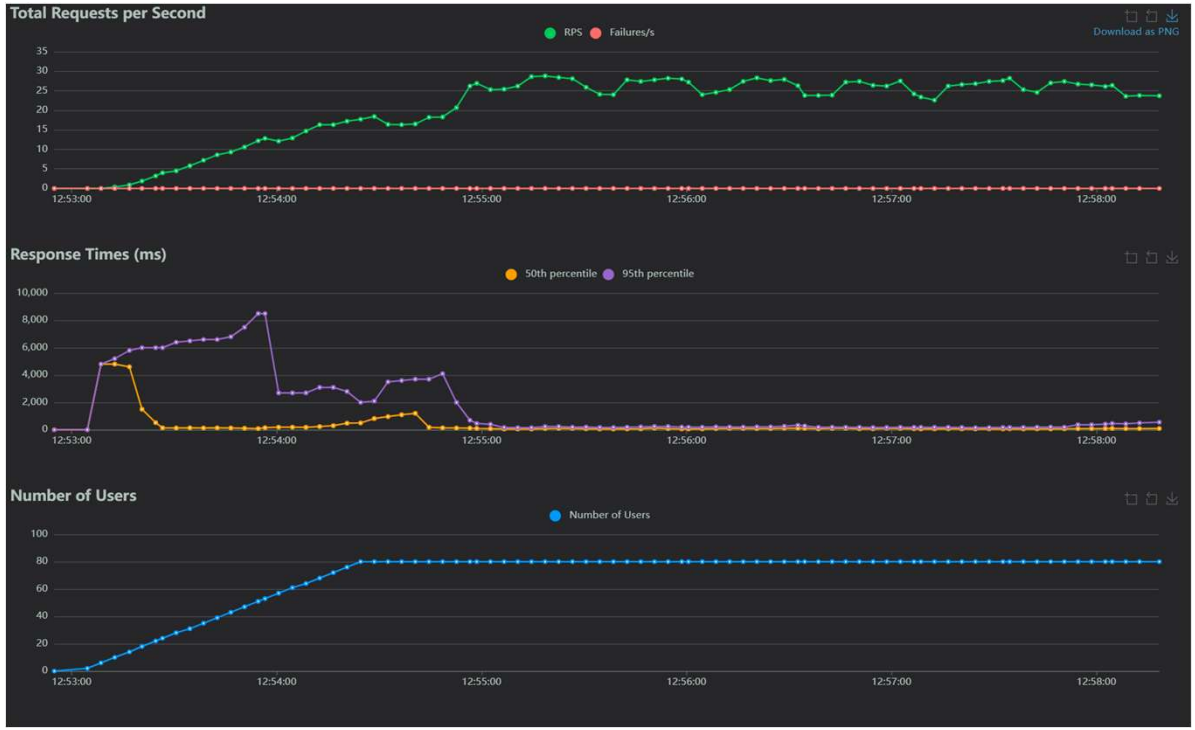
LOCUST TASKS

Task name	Description
Authentication (HEAD)	Verifies user credentials and server availability
Search (PROPFIND)	Lists the contents of the user's root directory
Read (GET)	Retrieves the contents of the Readme.md file (included by default)
Upload (PUT)	Uploads files of various sizes, depending on the load scenario
Deletion (PUT + Delete)	Uploads a temporary file and then immediately deletes it

On_start: initializes each virtual user randomly from the test pool and sets up an empty list to track the paths of uploaded files during the test.

On_stop: at the end of the simulations iterates over the uploaded file paths and deletes them

LIGHT LOAD TEST

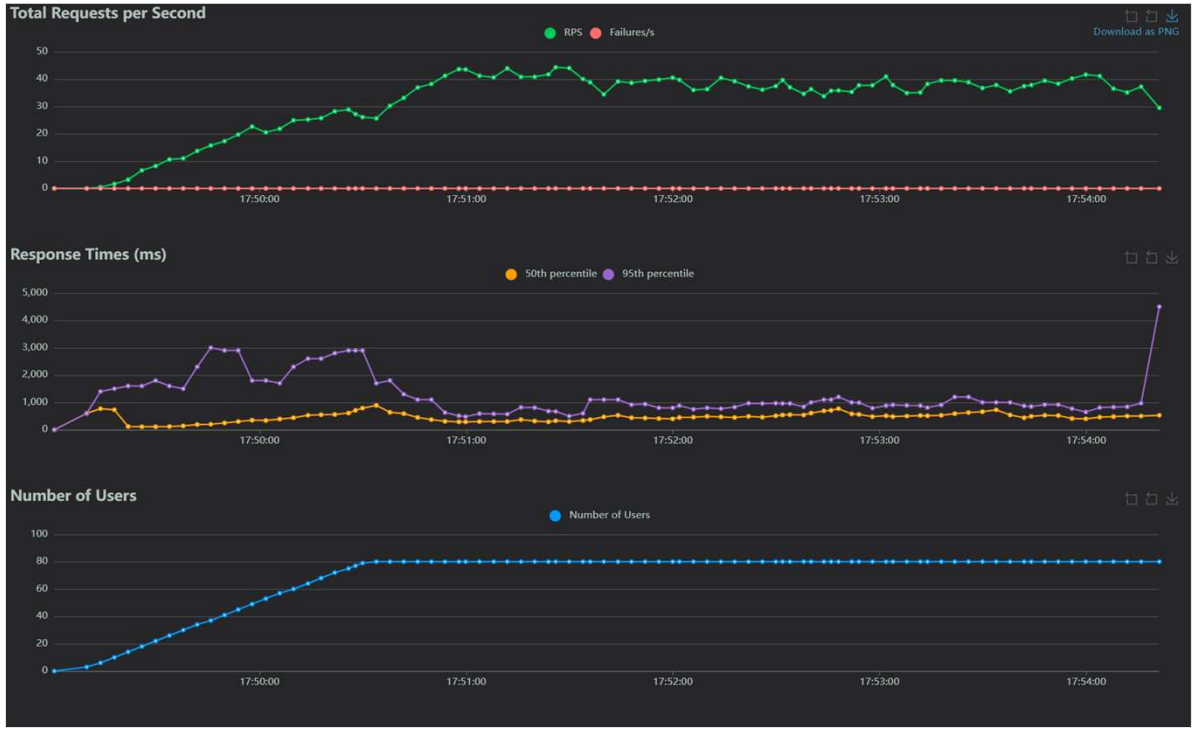


Failures: None

Average response time: 653.01 ms

Average upload response time: 438.05 ms

MEDIUM LOAD TEST

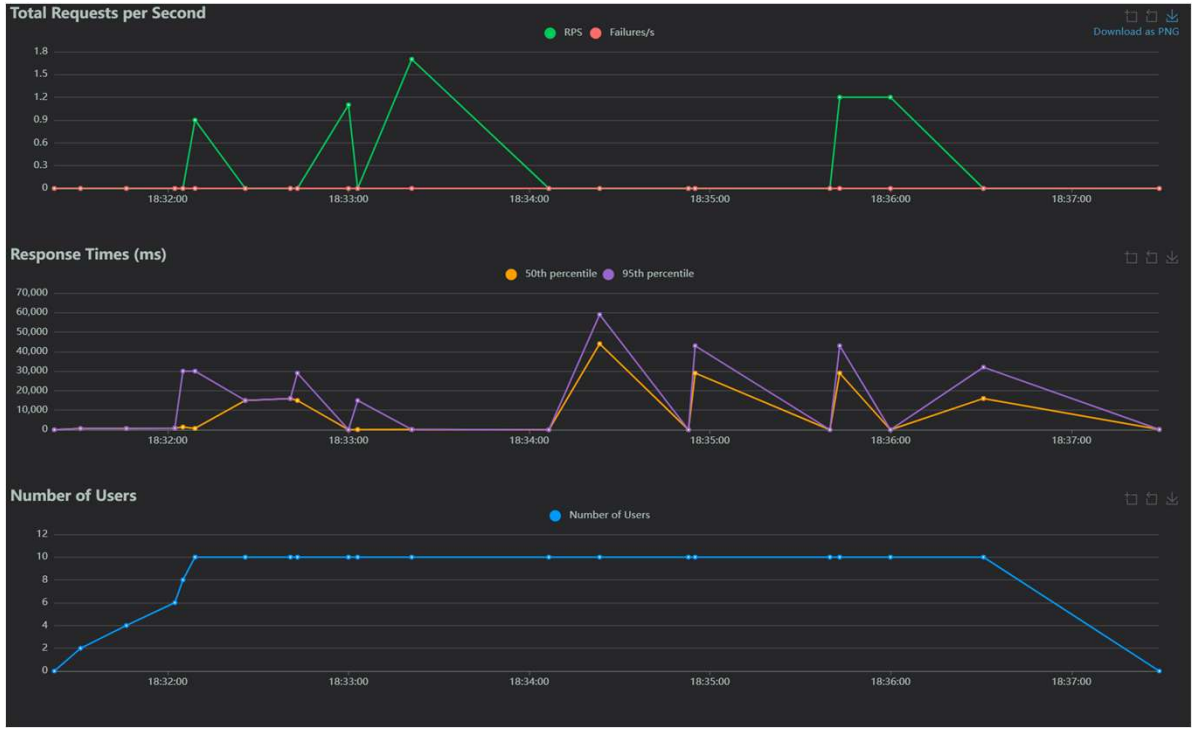


Failures: 4 Remote Disconnected errors

Average response time: 1038.79 ms

Average upload response time: 631.7 ms

HEAVY LOAD TEST

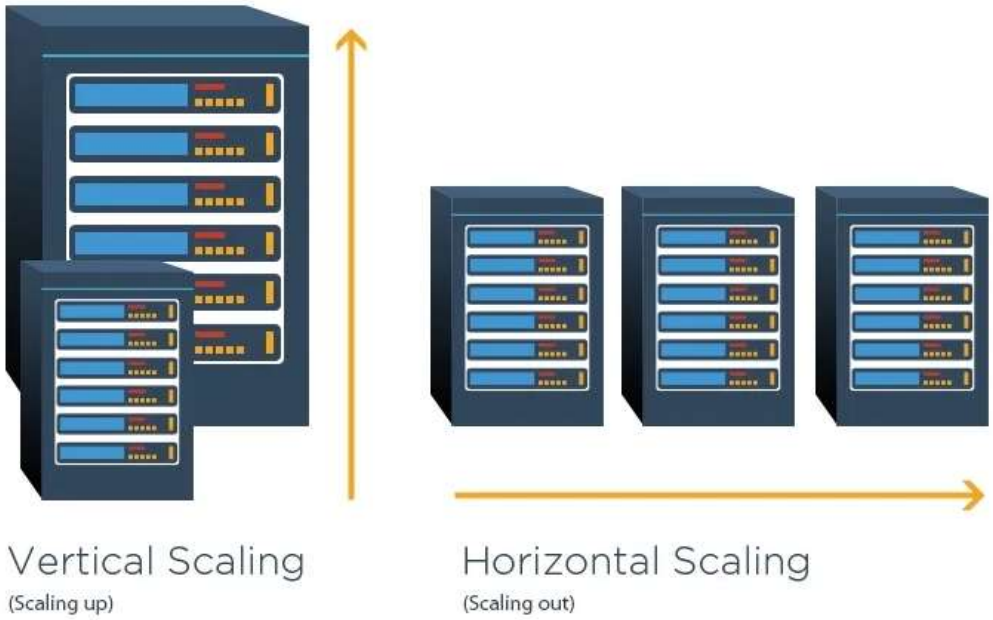


Failures: None

Average response time: 10452.42 ms

Average upload response time: 19437.61 ms

SCALABILITY AND COST EFFICIENCY



Cost saving measures:

- Storing under-accessed files in cheaper storage
- Monitoring resource usage (Grafana, Prometheus)
- Using a more optimized database

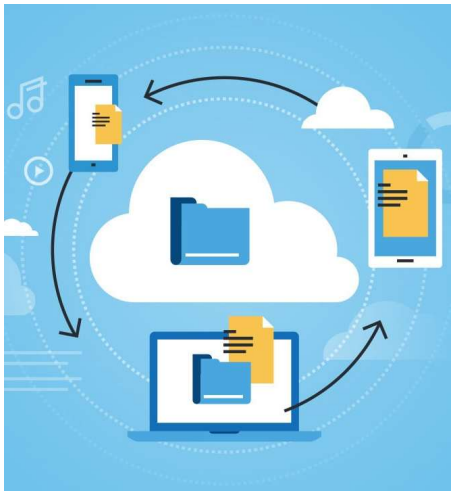
CONCLUSIONS

The cloud-based file storage system was successfully implemented.

The simulations with Locust revealed that the system performs well under **light and medium loads**, maintaining adequate latency and reliability.

However, under **heavy load**, performance degraded significantly, indicating the need for further optimization.

To address these challenges, **horizontal scaling** with multiple Nextcloud instances behind a load balancer is recommended.



THANKS FOR THE ATTENTION!

REFERENCES

Nextcloud Documentation, https://docs.nextcloud.com/

MariaDB, https://mariadb.org/

Locust Documentation, https://docs.locust.io/en/stable/

Docker Documentation, https://docs.docker.com/

Docker Compose Documentation, https://docs.docker.com/compose/