CLOUD BASED FILE STORAGE USING NEXTCLOUD

EMANUELE RUOPPOLO – SM3800049

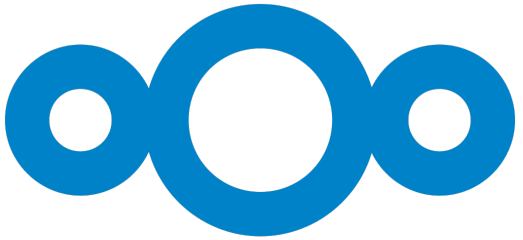
GOALS OF THE PROJECT

- Implementing a file storage that ensures:
  - **Broad network accessibility;**
  - **Users' facilities;**
  - **Security measures;**

- Using Nextcloud:
  - **Intuitive interface;**
  - **Full customizable in its features;**
  - **Highly compatible for the deployment in a Docker container for its setup;**

SETUP OF DOCKER CONTAINER

- Nextcloud has been deployed in a Docker container using a Docker-compose file:
  - A file `.yml` has been written with the needed specifics:
    - 2 volumes, one for Nextcloud one for a database;
    - 1 network for enabling communications between the volumes;
    - 2 services: Nextcloud, MariaDb, each with its specifics, for Nextcloud:
      - Port `8080:80` enabled between the container and the local host
      - Admin user and password set to login
  - Container created by running: `docker compose up –d`
  - Container shut down by running `docker compose down`
  - Nextcloud accessible from local browser typing `http://localhost:8080` and using admin credentials set in the compose file



Nextcloud

NEXTCLOUD DEPLOYED FEATURES

Nextcloud allows for a full personal configuration, admins can easily install apps to guarantee many services in terms of security, accessibility and users' commodities. In our case these are the main features deployed:

- **Registration and authentication**: Nextcloud has a clear interface where it's easy login in, sign up and log out;
- **Authorizations management**: different roles within the system have been imposed, admin and users, each with different permissions. Users have also been grouped together to ease communication with them;
- **File management**: Users have an assigned Quota of storage, based on their needs, and they can manage this storage by uploading, downloading and deleting files, also a feature has been implemented that notices a user when its storage is almost full.

SECURITY MEASURES

To ensure users' security and isolation Nextcloud issues an access token for each client to use for any HTTP request and **encrypts** the clients' passwords in its database, it has a **Logging** page on its interface to check all the suspicious activities and allows for antiviruses that control the integrity of users' files.

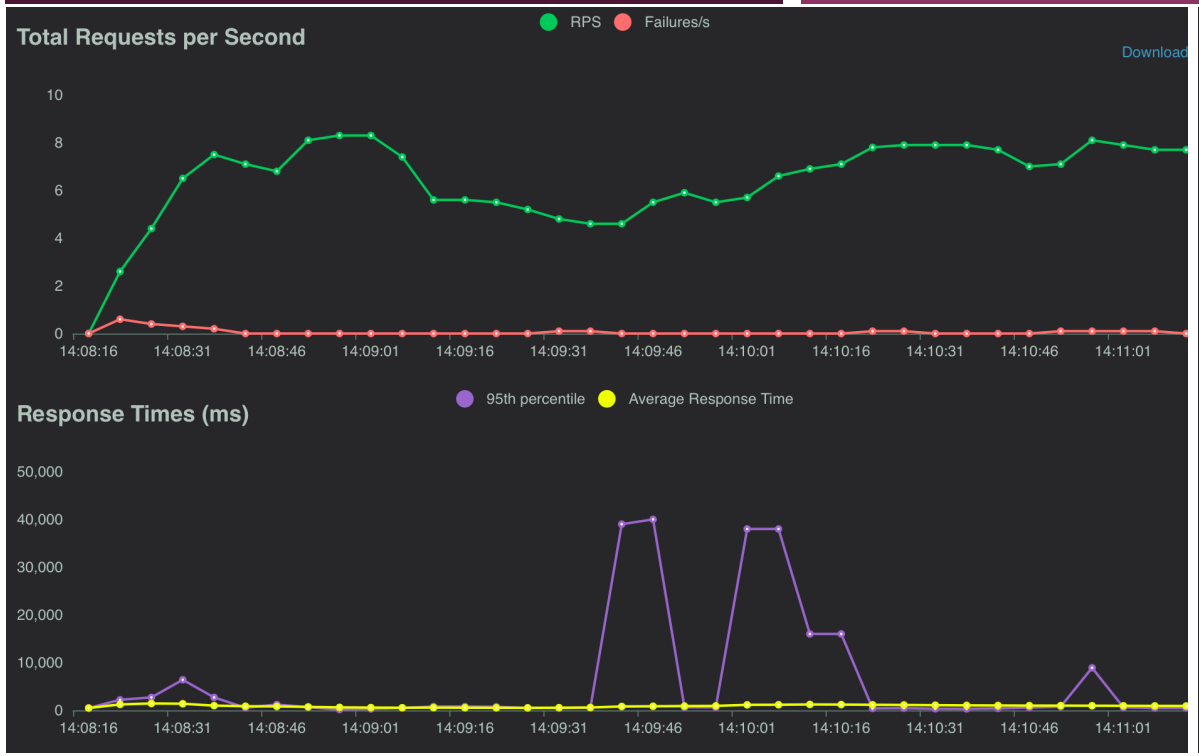
- **Registration** via the web interface has been made possible via e-mail-based sign-up with verification link;
- **Data encryption server-side** can be enabled;
- **Two factors authentication (2FA)** has been set in order to ensure the login;
- **Password recovery** via e-mail has been enabled;
- **Password constraints** such as minimum length, capital letters, special characters and numbers presence have been imposed

PERFORMANCES EVALUATION WITH LOCUST

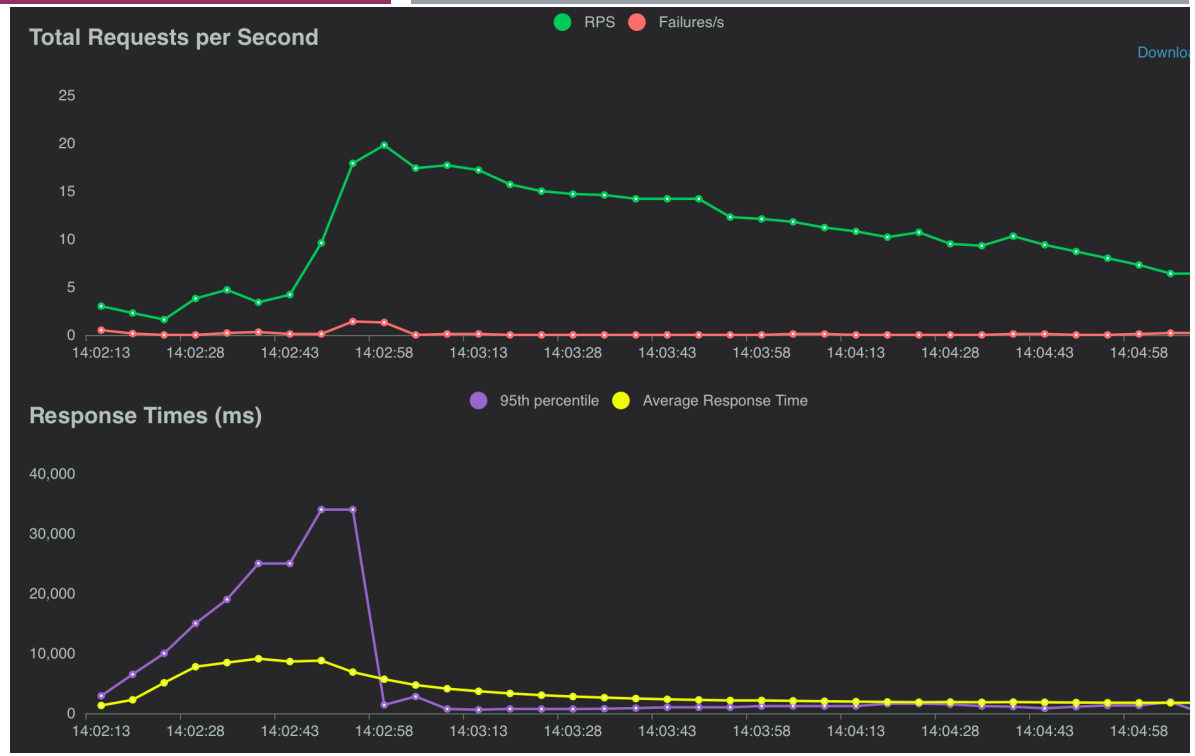
- Locust stands as an open-source tool rooted in Python, dedicated to performance testing. It simulates vast user numbers to measure the scalability and performance of web applications;
- It allows to simulate scenarios where multiple user interactions are done simultaneously;
- A load is generated to rigorously test the application or service in question;

Procedure:

- Multiple users (90) have been created (and similarly deleted) with a docker dedicated command, to each of them have been assigned the same storage Quota (3GB);
- Files of different sizes (1KB, 1MB, 1GB, .jpg) have been created and added to a load data local repository;
- A `tasks.py` file has been created to make Locust available to test the system with different HTTP requests: **head, propfind, get, put, delete**.



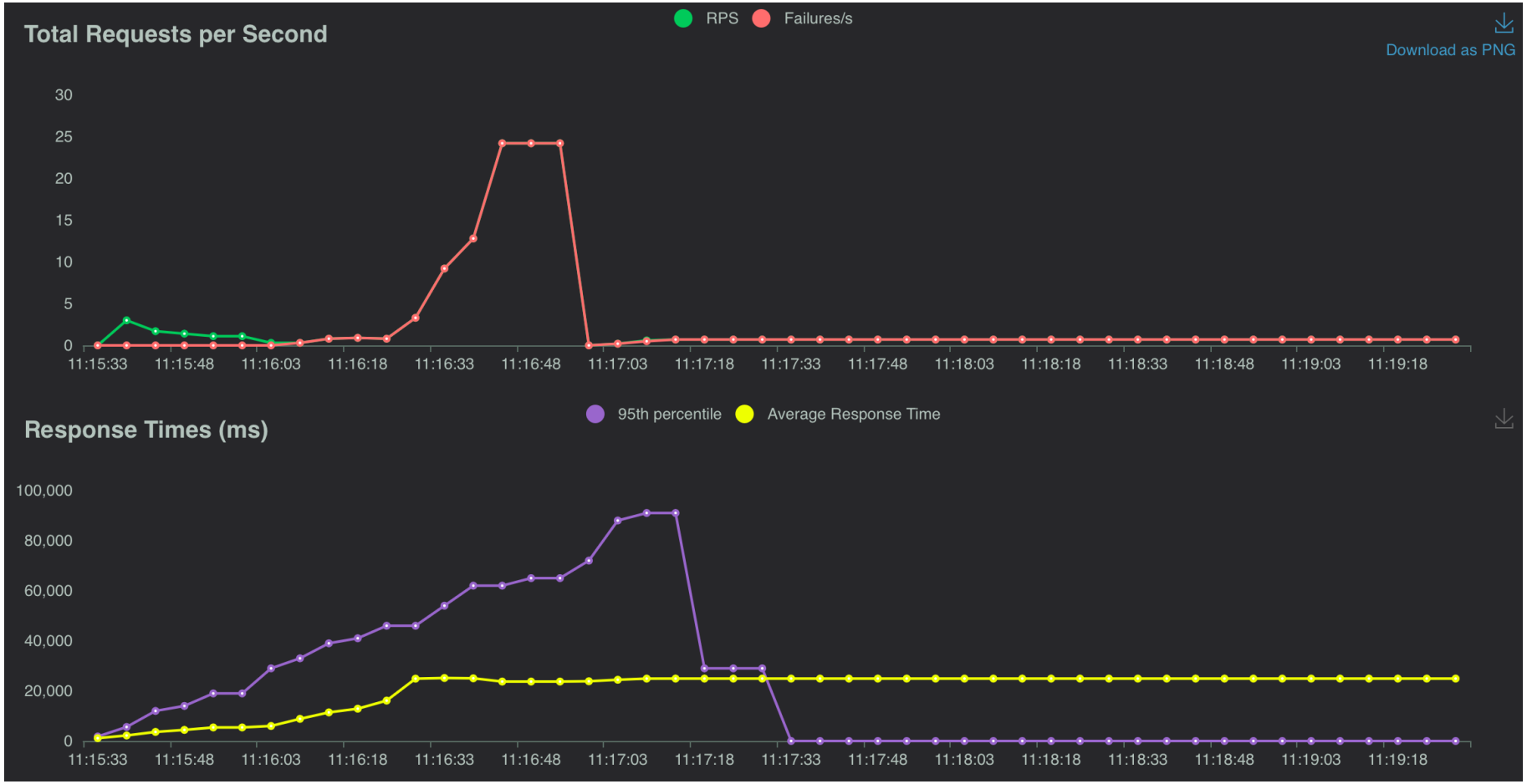
30 users



90 users

RESULTS…

… AND FAILURES



180 users

EXPLORING SPAWN RATE AND FILE WEIGHT DEPENDENCIES

Two other tasks files have been made in order to evaluate the performances of the system when stressed in a certain scenario, like allowing more users logging in simultaneously uploading small size files, or less users uploading bigger files. The results are here presented:

Method	50%ile (ms)	66%ile (ms)	75%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
`HEAD`	18	22	25	30	44	3400	6200	7600
`PROPFIND`	21	26	30	35	55	1800	5800	7100
`DELETE`	31	33	34	35	40	330	6000	7700
`PUT`	36	42	47	51	73	870	4900	6800
`GET`	25	30	35	38	58	670	5500	8000
Aggregated	30	33	37	40	58	670	5800	8000

Table 1: Test performed for small files with 60 users, spawn rate = 10, for $\simeq$ 2 min.

Method	50%ile (ms)	66%ile (ms)	75%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
`HEAD`	28	32	45	36	50	520	1300	1300
`PROPFIND`	33	46	51	53	56	59	1100	1200
`DELETE`	33	35	35	36	42	94	1500	2400
`PUT`	64	77	79	81	84	550	1600	1900
`GET`	39	52	60	61	63	65	1200	1200
Aggregated	37	49	56	61	79	120	1400	2400

Table 2: Test performed for big files with 20 users, spawn rate = 5, for $\simeq$ 2 min.

SCALABILITY OF THE SYSTEM: ON-CLUSTER

The results have been obtained testing the system on a `small` machine. An idea for scaling the system in a much powerful environment could be deploy it **on-cluster**, a network that interconnects nodes with a network.

pros	cons
Full horizontal scalability	Complexity of the system set-up and effort in its maintenance
Data resilience due to built-in redundancy in case of failures	Costs of the infrastructure and its maintenance
Independent security policies	
Full control of the infrastructure	

SCALABILITY OF THE SYSTEM: ON-CLOUD

Another chances is to deploy the system on a cloud-based environment. There are many, one of these that seems reasonable is AWS S3, that provides a scalable infrastructure, ensuring data security and pay-as-you-go costs. In general these are the features of a on-cloud deployment:

pros	cons
On-demand scalability without hardware expenses	Data privacy due to the unknown management of the infrastructure
Global accessibility	Dependency to a third party, service outages
Pay-as-you go approach, minimizing costs and optimizing resource utilization	

CONCLUSIONS

- Nextcloud offers a user-friendly interface and useful services including robust **user authentication**, **role-based access control**, **private storage allocation**, and **comprehensive admin management capabilities**
- Nextcloud allows for the standards security measures of a cloud-based-storage;
- Locust tests demonstrate good performances in case of a small-environment deployment;
- Scalability can be addressed in either an on-cluster environment or a cloud-based one.