CLOUD EXAM

Alessandro Minutolo

March 2024

1 Introduction

This project aims to address the increasing demand for efficient, secure, and scalable file storage by implementing a cloud-based solution. The primary objective is to enable file upload, download, and management functionalities while ensuring data privacy and security and, consequentially, to test the overall system in terms of scalability and cost-efficiency.

2 NextCloud

Nextcloud stands out due to its open-source nature, extensive documentation, and a robust community of developers and users that contribute to its continuous improvement. It offers a versatile set of features beyond basic file storage, including end-to-end encryption, collaboration tools, and customization options. This makes Nextcloud not just a file storage solution but a comprehensive platform for data management and collaboration.

The decision was further influenced by Nextcloud's compatibility with Docker, which simplifies the deployment and scaling process. Its emphasis on privacy and security aligns with our project's objectives, providing granular control over data access and user authentication. Moreover, the platform's flexibility in integrating with existing infrastructure and third-party applications offers the potential to extend functionality and enhance user experience.

3 Deployment

The deployment of our Nextcloud-based file storage system utilizes Docker and docker-compose, enabling a straightforward and efficient setup process on a laptop. This section outlines the deployment plan, leveraging the provided **docker-compose.yml** file to create a local containerized environment.

- **Preparation**: Ensure Docker and docker-compose are installed and running on the laptop. Place the docker-compose.yml file in a dedicated project directory.

- **Configuration**: The docker-compose.yml file defines two main services: db (MariaDB database) and app (Nextcloud application). It also specifies volumes for persistent storage and a dedicated network for inter-service communication.
- **Deployment Execution**: Navigate to the project directory in a terminal and run docker-compose up -d. This command pulls the necessary images, creates volumes for data persistence, sets up the network, and starts the services in detached mode. Once the containers are running, Nextcloud is accessible via http://localhost:8080, where you can complete the setup through the web interface.

4 User Features

4.1 User Authentication and Authorization

User authentication and authorization are foundational to ensuring that our Nextcloud deployment remains secure and functional while providing a seamless user experience. In Nextcloud, these aspects are managed directly through its web interface, which offers a comprehensive set of tools and options for admin users to effectively control access and user roles. Here's how Nextcloud facilitates these critical functionalities:

- **User Sign-Up, Log-In, and Log-Out**: Nextcloud allows users to create their accounts, sign in to access their files, and log out when finished. The process is streamlined for ease of use, ensuring that even users with minimal technical knowledge can navigate these steps effortlessly. While Nextcloud does not enable public user registration by default, plugins such as the Registration app can be installed to allow users to sign up independently, with email verification for added security.
- **Role-Based Access Control**: Nextcloud differentiates users primarily into two categories: regular users and administrators. Each category has distinct permissions and access levels within the platform. Administrators can assign and modify these roles through the Users section of the Nextcloud admin interface.
- **Private Storage for Regular Users**: Upon account creation, regular users are allocated private storage space where they can securely store their files. This space is isolated from other users, ensuring privacy and data protection. Administrators can adjust the storage quota for each user, managing the system's capacity and ensuring fair use across all accounts.
- **Administrative User Management**: One of the key responsibilities of Nextcloud administrators is managing user accounts. This includes creating new user accounts, modifying existing ones, resetting passwords, and

deleting users when necessary. Additionally, in order to manage common settings among users, a user group was created for which the admins are fully responsible.

4.2 Security Measures

NextCloud offers a variety of features and tools that administrators can implement to protect data and ensure a secure user environment. Many of these security measures can be easily managed and installed directly from Nextcloud's web interface.

- **Two-Factor Authentication (2FA)**: To add an extra layer of security to user accounts, Nextcloud supports two-factor authentication. Administrators can enable 2FA from the security settings in the web interface and users can choose from several second-factor methods, including TOTP (Time-based One-Time Password) apps, SMS, or hardware tokens.
- **File Access Control**: Nextcloud's File Access Control app allows administrators to create rules that restrict file access based on factors like user groups, file type, or size. This can help prevent the sharing of sensitive data and ensure that users only have access to the files they are authorized to view or edit.
- **Brute Force Protection**: Nextcloud includes a brute force protection feature that detects and slows down repeated failed login attempts, making it more difficult for attackers to gain unauthorized access through password guessing.
- **Server-side Encryption**: Nextcloud allows administrators to enable server-side encryption through its web interface, encrypting data stored on the server to protect it from unauthorized access.

5 Locust Testing

To assess the performance and scalability of our Nextcloud deployment under realistic user load conditions, we employ Locust, a versatile open-source load testing tool. Our Locust test script is built to simulate a variety of user actions that mimics typical interactions with the Nextcloud platform. In particular, the script performs a **'PROPFIND'** request to list files, a **'PUT'** request to upload files of different sizes (1KB, 1MB, and 1GB) and a **'GET'** request to read a Readme.md file shared among all users. Each task utilizes **HTTPBasicAuth** for authentication, ensuring each request is performed as a unique user with proper credentials.

In order to simulate user actions on NextCloud, 70 users were created through a bash script (**create_users.sh** in the 'scripts' directory), which automatically creates 70 users with 3GB of private storage. Before performing each locust test

we want to be sure that all the users' private storage is empty through another bash script (**clear_user_storage.sh** in the 'scripts' directory)

5.1 Results

We start testing with a modest number of users to establish baseline performance metrics under normal operating conditions. This provides insight into the system's default handling of user requests. To stress-test the system, we simulate a high number of concurrent users performing tasks as defined in the script. After examining the Locust test reports for simulations with 10, 30, 50, and 70 users, several patterns emerge that highlight the scalability and performance characteristics of the Nextcloud deployment under test. As the number of simulated users increases, so does the strain on the system, which is reflected in various metrics including response times, failure rates, and system throughput.

- **10-users test**: the system under test displayed commendable performance and stability with 10 users. The average response times were generally low, and the system managed to handle requests efficiently with minimal errors.

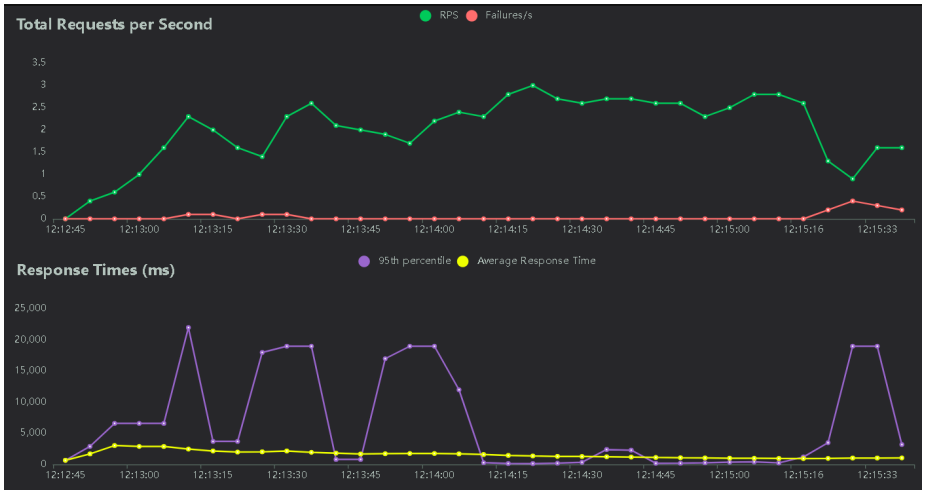

Figure 1: 10-users test

- **30-users test**: increasing the user count to 30 introduced a slight uptick in response times, particularly for more resource-intensive operations like uploading 1GB files. While the system still maintained a relatively low failure rate, the increased load began to reveal limitations in handling concurrent large file uploads, evident from the occasional HTTP 507 errors.

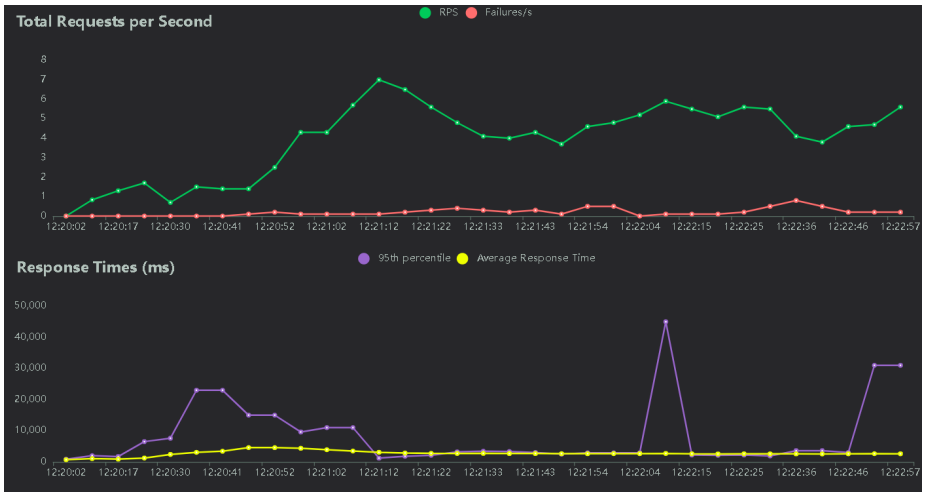

Figure 2: 30-users test

- **50-users test**: At 50 users, the strain on the system became more pronounced. The average response times for operations, especially file uploads, showed a marked increase. The system encountered a broader array of errors, including **'RemoteDisconnected'** and **'HTTPError'** responses, indicating potential bottlenecks in network throughput or server processing capabilities.

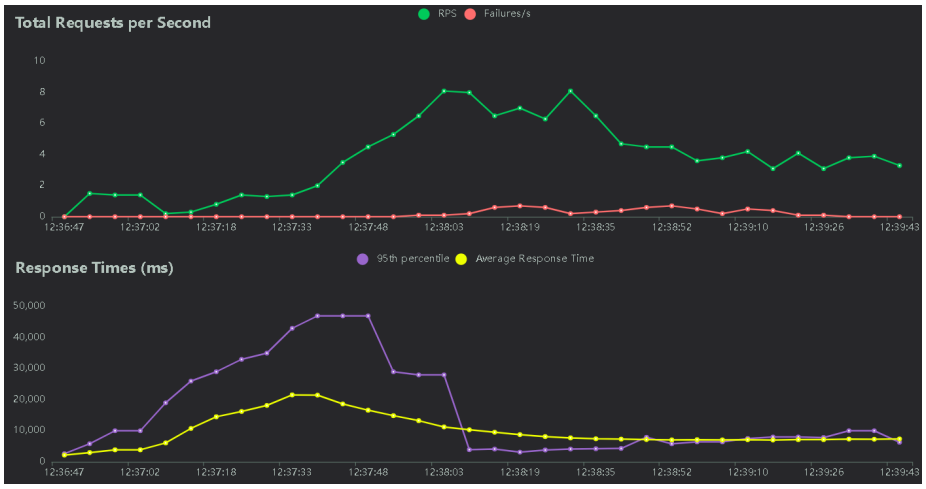

Figure 3: 50-users test

- **70-users test**: The test with 70 users presented the most significant challenges for the system. Response times increased substantially across all

tasks, and the failure rates spiked, underscoring the system's difficulty in managing simultaneous connections and requests. The prevalence of **'RemoteDisconnected'** errors suggests that the server was overwhelmed, possibly due to limitations in handling concurrent HTTP connections or due to the web server configuration.

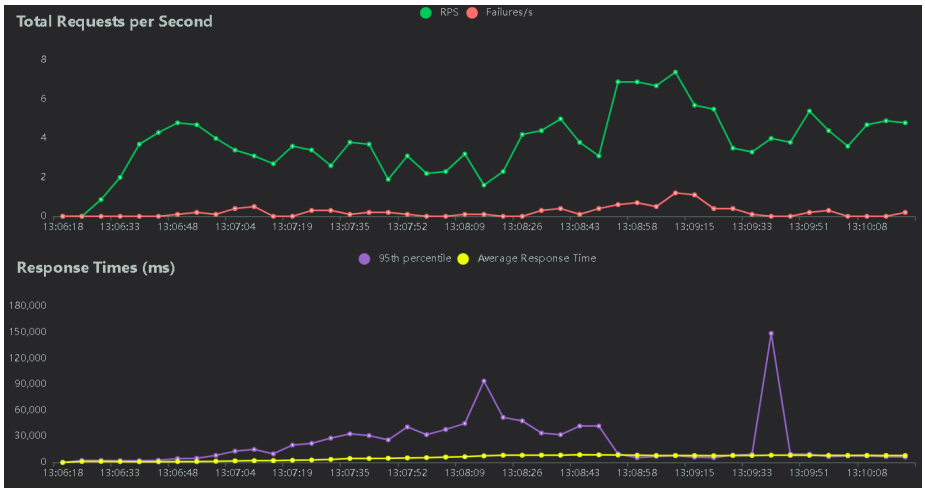

Figure 4: 70-users test

Initial tests with lower user counts showed promising system stability and efficiency, but as the load increased, so did the response times and error rates. This indicates that while the Nextcloud deployment is capable of handling small to moderate user loads effectively, it requires optimization or scaling strategies to improve performance under higher loads.

6 Scalability

Taking into account the performance results from the Locust tests and the broader context of our discussion, let's outline suitable solutions to optimize scalability and cost-efficiency for the Nextcloud deployment. Given the increase in response times and failure rates observed with higher user loads, particularly when dealing with file uploads and downloads, the following solutions are proposed:

- **Caching layers**: introducing caching mechanism can reduce network traffic by retaining recently accessed disk blocks in a cache, so that repeated accesses to the same information can be handled locally.
- **Horizontal scaling**: To handle an increasing load and traffic, we opt for a horizontal approach, we could create more instances of Nextlcoud and distribute the load between them.

- **Object Storage service**: There are many already deployed cloud providers that also offer auto-scaling solutions that dynamically adjust computing resources based on the current workload.

7 Cost-Efficiency

In order to discuss about the costs of this system, we have to understand the actual purposes of the project. In case we want to keep the system local, the costs would be close to zero; we would only want one or more external hard drive to increase the memory available. In case we want to make it open access, we would spend money on the domain service (usually prizes goes from 10$ to 50$), and, supposing this project becomes much wider than now, we would rely on a cloud service such as Google Cloud Storage or Amazon S3. Both of them charge primarily based on the amount of data stored, measured in GB per month.

8 Cloud Providers

In this section of the report, we focus on evaluating two prominent cloud providers, Amazon S3 and Google Cloud Storage, for hosting our Nextcloud deployment. By comparing these platforms, we aim to determine the most suitable option based on a detailed assessment of their features, cost structures, and overall compatibility with our requirements.

8.1 Cost-Efficiency

- **Amazon S3** offers a detailed pricing model that includes costs for storage per GB, requests, and data transfer. It features various storage classes such as S3 Standard, S3 Intelligent-Tiering, S3 One Zone-IA, and Glacier, each designed for different use cases and cost savings based on access frequency and data durability requirements.
- **GCS** also offers competitive pricing with a simple structure based on data storage, network usage, and operations. It includes storage classes like Standard, Nearline, Coldline, and Archive, each priced according to data access patterns and storage duration.

8.2 Scalability

- **Amazon S3** is designed to offer great scalability and handling of vast amounts of data and traffic, S3 supports extensive data operations with strong consistency and built-in redundancy.
- **GCS** benefits from Google's global infrastructure, which is optimized for high performance and rapid data access. Google's private network pro-

vides fast data transfer rates, which can be crucial for applications requiring quick access to large amounts of data.

8.3 Security

- **Amazon S3** provides detailed access controls, bucket policies, and integrates with AWS Identity and Access Management (IAM) for fine-grained security management.
- **GCS** also offers robust security measures. Google provides similar encryption capabilities and easy integration with Google Cloud's Identity and Access Management for managing permissions.

8.4 Solution

Finally, to choose between Amazon S3 and Google Cloud Storage for hosting Nextcloud, we should consider integration needs and data access patterns. We would opt for Amazon S3 if we require deep integration with other AWS services, sophisticated security controls, and scalable storage solutions. Google Cloud Storage is preferable for projects prioritizing high-speed global data access and cost efficiency, especially with infrequently accessed data. Both offer robust, secure, and scalable environments, but the right choice depends on our existing infrastructure and future developments.

9 Conclusion

In conclusion, the overall evaluation of our Nextcloud deployment project, including considerations for scalability, cost-efficiency, security, and provider selection, has provided us with a clear path for successful implementation of a file storage system through a cloud-based solution.