Cloud Computing & Backend Development

- Group Members
- Harshal Gavali -2003040
- Adharva Kumar Thodupunooru -2003042
- Gourav Powar -2003044
- Rohan Chinchkar -2003046

INTRODUCTION

- Cloud computing is a technology that provides computing services over the internet, making powerful computing resources accessible without investing in hardware.

- Backend development is the process of building and maintaining the server-side of web applications using a programming language such as Java, which is fast, secure, and scalable.

- Together, cloud computing and Java backend development enable businesses to build and deploy web applications that are robust, scalable, and secure, while also being flexible and cost-effective.

Problem Statement 1

Title: "Scalable Cloud-Based Java Application Architecture for High Performance Storage"

Reference: IEEE International Conference on Load Balancing Framework for Cross-Region Tasks in Cloud Computing (2021)

Description : Designing a scalable and high-performance cloud-based Java application architecture for high-performance computing.

Challenges involved in designing such an architecture, such as load balancing, fault tolerance, and scalability.

Problem Statement 2

Title: "Performance and Scalability Challenges in Cloud-Based Java Enterprise Applications"

Reference: IEEE International Conference on High performance cloud computing on multicore computers (2018)

Description : Overcome the performance and scalability challenges faced by cloud-based Java enterprise applications.

Challenges involved, such as database performance, network latency, and resource allocation.

Problem Statement 3

Title: "Towards Reliable and Efficient Backend-as-a-Service with Java"

Reference: IEEE International Conference on Research on Auto-Scaling of Web Applications in Cloud (2019)

Description : Design a reliable and efficient Backend-as-a-Service (BaaS) using Java.

 Challenges involved in designing a BaaS, such as scalability, security, and data consistency.

Choosing problem Statement

Title: "Scalable Cloud-Based Java Application Architecture for High Performance Storage"
 We know that this is an important problem to solve due to the increasing demand for scalable and high-performance computing and storage.

Benefits of Scalable Cloud-Based Java Application Architecture

Creating a scalable cloud-based Java application architecture for high-performance storage.

Architecture can provide high availability, improved performance, easy scalability, and reduced costs.

Businesses that have successfully implemented this architecture, such as Netflix and Airbnb, and have seen significant improvements in their operations.

Challenges of Building Scalable Cloud-Based Java Application Architecture

Challenges associated with building a scalable cloud-based Java application architecture for high-performance storage.

Challenges include load balancing, fault tolerance, and scalability.

Challenges can be overcome through careful design and implementation, such as using load balancers and implementing fault tolerance mechanisms.

Technology we will use:

1. Spring Framework:

Spring is an open-source framework that provides a comprehensive programming and configuration model for Java-based applications. It provides solutions for various enterprise applications like web, mobile, and cloud. Spring framework provides several modules like Spring Core, Spring MVC, Spring Data, and more.

2. Spring Boot:

Spring Boot is an extension of the Spring framework, which simplifies the developments of Spring-based applications. It provides a set of starter dependencies and auto-configurations to reduce boilerplate code. Spring Boot is widely used to develop microservices, web applications, and RESTful APIs.

3. REST API:

REST (Representational State Transfer) is a software architectural style that defines a set of constraints to be used when creating web services. RESTful APIs are designed to work with HTTP protocols like GET, POST, PUT, and DELETE. They enable the development of scalable and flexible web services.

Project Development Steps:

1.Set up the Development Environment :

First, we need to set up the development environment. We can use any IDE like Eclipse, IntelliJ IDEA, or NetBeans for development. We also need to install Maven for dependency management.

2. Create a Spring Boot Project :

Next, we need to create a Spring Boot project using the Spring Initializer. We can choose the required dependencies like Spring Web, Spring Data JPA, and H2 Database.

3. Implement RESTful API:

We need to create RESTful API endpoints for file upload, download, and deletion. We can use the Spring Web module to implement the API endpoints.

4. Implement File Storage:

We need to implement file storage to store the uploaded files. We can use the Java NIO package to handle file operations.

5.Add Security:

As the data is confidential, we need to add security to the API endpoints. We can use Spring Security to provide authentication and authorization.

6. Testing:

We need to test the application by uploading, downloading, and deleting files. We can use tools like Postman or Swagger to test the API endpoints.

Conclusion:

In conclusion, developing a private cloud storage project using Spring, Spring Boot, and REST API is a great way to address the requirement for secure data storage. The project development involves setting up the development environment, creating a Spring Boot project, implementing RESTful API endpoints, file storage, adding security, and testing. By following these steps, we can develop a robust private cloud storage project.

References

1. Load Balancing Framework for Cross-Region Tasks in Cloud Computing. Article in Computers, Materials and Continua · September 2021 DOI: 10.32604/cmc.2022.019344
2. High performance cloud computing on multicore computers, Spring 5-31-2018, Jianchen Shan New Jersey Institute of Technology
3. IEEE International Conference on Research on Auto-Scaling of Web Applications in Cloud (2019), DOI 10.12694/scpe.v20i2.1537 ISSN 1895-1767
4. Analysis of Critical Factors in Manufacturing by Adopting a Cloud Computing Service, DOI:10.32604/csse.2022.021767