Azure Cloud Adoption Reasoning

Contents

Introduction

The objective of this document is to explain the reasoning behind the choice of cloud provider – Microsoft Azure. The decision is important for aligning with the project's goals of scalability, performance, and security measures.

Project Overview

My project, Movimingle, is an online software that is designed to improve the experience of movie selection and collaborative watching better. It uses highly scalable, secure, and efficient cloud infrastructure. The application will be able to handle vast amounts of user interactions, thus the need for cloud services is an integral component of the architecture.

Technical Requirements and Alignment with Azure

Microservices Architecture

From the beginning of the project the idea of the microservices architecture was part of it, so this is how I developed it. I used Kubernetes for the orchestration and for that the Azure Kubernetes Service was used when it came time to deploy it in the cloud. The benefits are the good management of Kubernetes, simplifying deployment and operations of the microservices.

Database Services

As I have noted in the database selection document, initially I was planning on using two different databases for the backend. This, however proved to be unnecessary since I am not taking advantage of all of the features of the PostgreSQL database. This was only noted down in the documentation and the actual implementation still uses PostgreSQL and Cassandra. Azure supports this dual-database strategy, as well as just the Cassandra one. It uses Azure Cosmos DB for Cassandra API and Azure Database for PostgreSQL, which are scalable, secure, and highly available database services.

Backend Technologies

The backend, developed with Spring Boot, needs a reliable environment for deployment and scaling. Azure's support for Java applications, integrated with Azure Spring Cloud, simplify this requirement.

Frontend Deployment

Separate Deployment for React Application

When I was developing the software, I was not sure whether to deploy the frontend separately or in the cluster. Locally I did both and both were working fine, however I encountered a number of challenges when deploying the frontend in the cluster. In the end I succeeded, but I did not have much time to deploy the application in Azure, so I decided not to risk being late for the delivery and demo and went with the easier (for me) solution, as I did not know what changes I may need to make to the configuration in the cloud. The React-based frontend is now deployed as a standalone Azure Static Web App, outside of the Kubernetes cluster. As well as the previously mentioned reasons, this separation offers several advantages:

- **Scalability and Isolation:** Deploying the frontend separately allows for independent interface scaling, which could be cost effective and efficient.
- **Simplified Management:** The features that Azure provides for the Static Web Apps simplify the management experience. Some of the features are automatic scaling, patching, and CI/CD integration.
- **Improved Performance:** By isolating the frontend, we can improve the delivery of static content using Azure's global distribution capabilities, resulting in shorter load times and a better user experience.
- **Security:** The separate environment improves the security by limiting the exposure of backend services.

Performance

The load testing was performed on my local Kubernetes cluster on my laptop, this was done to reduce the costs of the cloud services. By doing this, I have established a baseline for the performance under predictable load conditions. The future plans are to do cloud-based testing in Azure, which will leverage its scalable environment to simulate real-world usage scenarios. When done with the tests I will be able to optimize the performance of the application when deployed in the cloud.

Security and Compliance

Advanced Threat Protection

Azure Security Center provides multiple tools that are integrated and provide monitoring, threat detection etc. These integrated tools continuously monitor security health and have advanced threat detection using machine learning and behavioral analytics. These tools are important for maintaining the integrity and security of the cloud resources and data.

Data Protection and Regulatory Compliance

Azure makes sure that data stored in its cloud services complies with legal and regulatory requirements relevant to various industries and regions, such as GDPR. Azure maintains a list of a lot of compliance certifications, which makes it a suitable platform for projects requiring stringent data protection controls.

Encryption and Network Security

Sensitive data is protected on Azure, because they provide andvanced encryption features for data at rest and in transit. It also offers firewall configurations, link services, and virtual networks that isolate and protect resources from unauthorized access. All this improves the network security.

Cost Management and Scalability

Cost-effective Resource Management

Azure’s pricing model is pay-as-you go, which means that you have a complete control over the resource usage, minimizing wastage and reducing costs. The optimization of costs is further improved by the ability to scale services and resources according to precise demand. All of this means that the user satisfaction will be high and the efficiency of the operations also. As well as that Azure has detailed insights into spending patterns and offers recommendations for cost optimization.

Auto-scaling and Performance Optimization

As mentioned in the above paragraph, Azure has autoscaling capabilities that scale the application based on the load without affecting uptime or user experience, done by adjusting the resources.

Conclusion

Azure was selected because it aligns with the project’s goals of scalability, security, and innovation. Its good infrastructure, combined with advanced security measures, cost management tools, and strong compliance with international standards, offer a great environment that supports both current needs and future growth. With the capabilities of Azure and its use in the project the Movimingle software will remain efficient and secure.