



USE CASE

CLOUD MIGRATION – CLOUD TO CLOUD – LIFT & SHIFT – GCP TO AZURE

OVERVIEW

On our tech exploration journey, we discovered an innovator rooted in a thriving metropolis, a beacon of forward-thinking and adaptability in the tech realm. Their location, pulsating with life and dynamism, seemed symbolic of their approach: always moving, always evolving.

This company wasn't merely riding the waves of innovation; they were making deliberate strides, evident in their strategic cloud decisions. Their commitment extended beyond enhancing performance and reliability. It was about cultivating development efficiency and, crucially, fortifying their digital foundation.

The company's philosophy resonated with us. It wasn't just about adapting to technological changes but leading the charge, perpetually exploring and innovating. Their interest in varied database strategies during migration revealed their relentless drive for growth and distinction.

MIGRATION OBJECTIVE

The company aims to improve performance, reliability, and development efficiency by migrating its core application from Google Cloud Platform (GCP) to Microsoft Azure. They plan to adopt a lift-and-shift approach with automation along the way, including database migration.

The company also wants to explore different database approaches during the migration process.

Migration objectives include:

- Migrating the core application from GCP to Azure using a lift and shift approach to minimize downtime and complexity.
- Utilizing automation tools and scripts to streamline the migration process and ensure a seamless transition.
- Evaluating different database migration approaches to find the most suitable option for each database.

MIGRATION STRATEGY

Infrastructure assessment and provisioning:

- Conduct a detailed assessment of the existing infrastructure on GCP, including virtual machines, storage accounts, networking components, and security groups.
- Set up the Azure environment, creating virtual networks, storage accounts, and other necessary resources to align with the GCP setup.
- Utilize Infrastructure-as-Code (IaC) tools like Terraform or Azure Resource Manager (ARM) templates to automate resource deployment.

Data migration:

- Choose the appropriate database migration approach based on each database's size and complexity.
- For smaller databases, opt for lift-an-shift migration, transferring the entire database schema and data to Azure.
- For larger databases, set up replication between GCP and Azure databases to keep data synchronized during migration.
- Ensure data consistency and integrity throughout the migration process.

Application migration:

- Package the core application and its dependencies for deployment on Azure.
- Utilize automation scripts to streamline the deployment process and ensure compatibility with Azure services.
- Conduct thorough testing and validation in a staging environment to identify and resolve any issues before production deployment.

MIGRATION RESULTS

Lift-and-shift migration:

Reduced downtime during migration due to the lift-and-shift approach, minimizing disruptions to end-users and business operations.

Automated deployment:

Streamlined migration process with automation, resulting in faster and error-free resource provisioning on Azure.

Database migration:

Successful migration of databases using suitable approaches, ensuring data consistency and preserving database integrity (setting up replication between GCP and Azure databases).

TAKEAWAYS

Lift-and-shift approach:

The lift-and-shift strategy facilitated a smoother migration process with minimal downtime and complexity, enabling a seamless transition from GCP to Azure.

Automation benefits:

Leveraging automation tools and scripts simplified resource provisioning and deployment, contributing to a more efficient migration experience.

Database migration flexibility:

By evaluating different database migration approaches, we ensured that each database was migrated using the most suitable and efficient method.

CONCLUSION

Migration from GCP to Azure successfully achieved our partner's objectives of improving performance, reliability, and development efficiency. The lift-and-shift approach, combined with automation, resulted in reduced downtime and streamlined resource provisioning.

Moreover, the careful consideration of database migration approaches ensured data integrity and consistency throughout the migration process. By migrating to Azure, the company is well-positioned to leverage the platform's capabilities, enhancing its core application's performance and enabling future growth and innovation.

EXECUTIVE SUMMARY

Industry	Technology Accessories and Innovation
Partnership Duration	14 months
Architecture	Migrating core application from Google Cloud Platform (GCP) to Microsoft Azure
Platforms	Web, Windows, Mac
Team Structure	Cloud Architect, Database Architect, DevSecOps engineer, Client and Backend Developers, Full Stack, QA engineers, BA, Data Engineer, Security Engineer, Network Engineer, Analytics Specialist, Cloud Cost Analyst, Release Manager
Tech Stack	Cloud Provider: Google Cloud Platform (GCP) and Microsoft Azure Development Languages: JavaScript (Node.js), Python, Java, Go (Golang), Ruby, C# (.NET Core) DevOps: CI/CD pipelines, Jenkins/GitLab CI, automated testing Infrastructure as Code (IaC) Tools: Terraform or Azure Resource Manager (ARM) templates