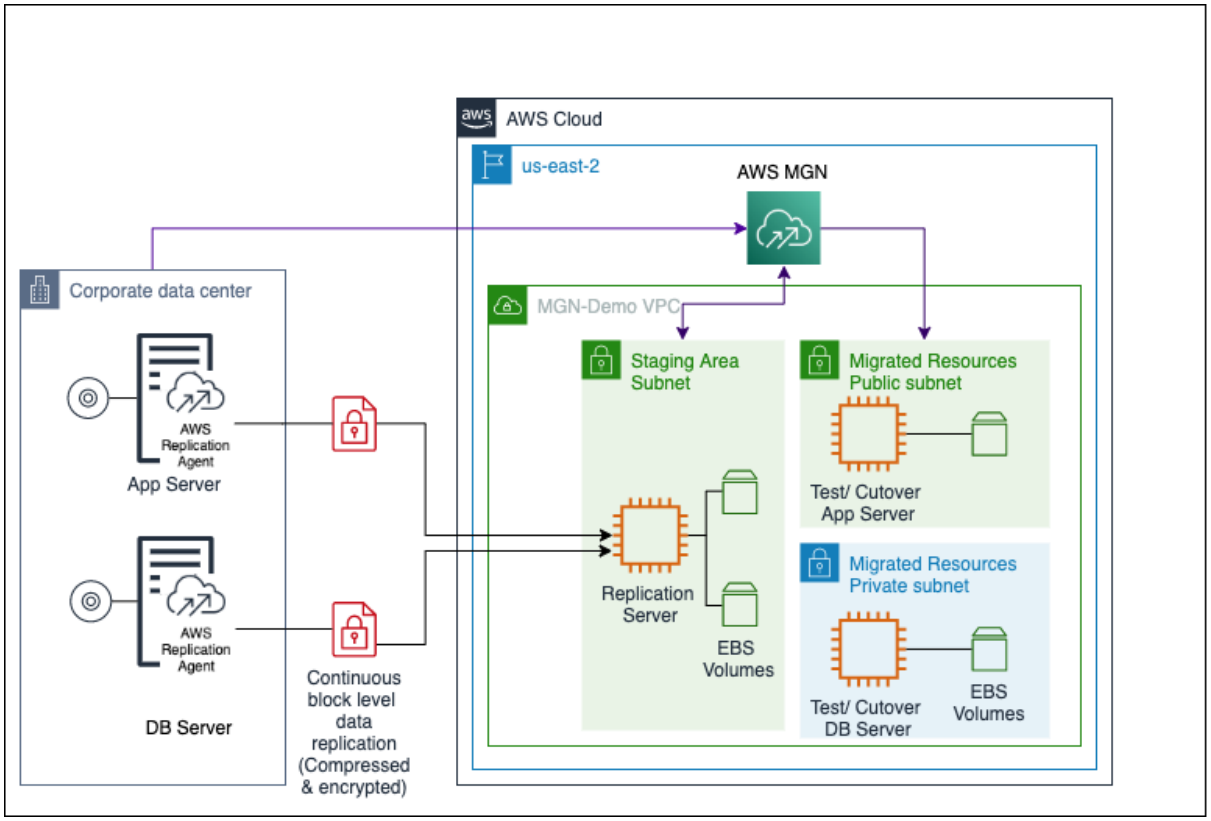
AWS Migration for On-Premises Workloads

Project Overview

In this project, we will design a solution to migrate an on-premises three-tier application and a data analytics workload to AWS (Amazon Web Services). The three-tier application consists of a frontend, backend, and database, while the data analytics workload involves Apache Hadoop. Our goal is to leverage AWS services to improve scalability, availability, and disaster recovery capabilities while optimizing costs. We will explore two migration approaches: a cloud-native architecture and a lift-and-shift approach.

Architecture Diagram



Components:

1. **Internet Gateway**: Serves as the entry point for user traffic from the internet.
2. **Load Balancer**: AWS Elastic Load Balancer (ELB) distributes incoming traffic across multiple frontend instances for high availability and scalability.
3. **Frontend**: Deployed on Amazon Elastic Beanstalk, the frontend is a cloud-native web application. It consists of HTML, CSS, and JavaScript components. Elastic Beanstalk handles the deployment, scaling, and load balancing automatically.
4. **Backend**: The backend is hosted on Amazon Elastic Container Service (ECS) with Fargate launch type. It runs the Apache Web Server and Java application in containers. ECS ensures efficient resource utilization and scalability.
5. **Database**: Amazon RDS (Relational Database Service) hosts the MySQL database. It provides managed database services with automated backups, scaling, and high availability.
6. **Data Analytics Workload**: The on-premises Hadoop workload will be migrated to Amazon EMR (Elastic MapReduce). EMR is a fully managed big data service that simplifies cluster provisioning and scaling.
7. **Data Ingestion**: AWS Glue is used for data ingestion. It automates data extraction, transformation, and loading (ETL) processes, making it easy to ingest data into EMR for analysis.
8. **Data Storage**: Amazon S3 (Simple Storage Service) stores both raw and processed data. It provides durability, scalability, and low-cost storage options.
9. **Data Visualization**: Amazon Quick Sight is employed for data visualization and deriving insights. Quick Sight offers easy-to-use dashboards and analytics for business intelligence.

Written Explanation

Migration Approaches

We considered two migration approaches:

1. **Cloud-Native Approach**: We chose this approach for the frontend and backend components of the three-tier application. By deploying the frontend on Elastic Beanstalk and the backend on ECS with Fargate, we take full advantage of cloud-native services. This allows for automatic scaling, load balancing, and cost optimization based on actual usage. It also enhances availability and simplifies deployment.
2. **Lift-and-Shift Approach**: For the MySQL database and the Hadoop analytics workload, we opted for a lift-and-shift approach. We utilized Amazon RDS to host the MySQL database with minimal code changes. Similarly, we migrated the Hadoop workload to Amazon EMR, which offers Hadoop compatibility and simplifies cluster management. This approach minimizes the need for code refactoring while benefiting from AWS-managed services.

Conclusion

This project outlines a comprehensive migration plan for both the three-tier application and the data analytics workload to AWS. By leveraging cloud-native services and lift-and-shift strategies, we ensure scalability, availability, and cost efficiency while minimizing disruption to existing systems. AWS services such as Elastic Beanstalk, ECS, RDS, EMR, Glue, S3, and Quick Sight play pivotal roles in achieving a successful migration.