Azure Migration Plan

Overview

Goal: Move the e-commerce application from AWS to Azure efficiently.
My Background: I hold the Microsoft Certified: Azure Administrator (AZ-104) certification, so I am familiar with the Azure ecosystem and can map AWS resources to their Azure equivalents effectively.

Phase 1: Infrastructure (Terraform)

Since I built the current infrastructure using Terraform (IaC), I will refactor the code to use the azurerm provider instead of aws.

- **Network:** Create **Azure VNet** instead of VPC.
- **Compute:** Deploy **Azure Linux VMs** instead of EC2.
- **Security:** Use **Network Security Groups (NSGs)** for traffic filtering.

Phase 2: Database Migration (Learning Approach)

For the MySQL database, I know that simply dumping and restoring data might cause too much downtime.

- **Proposed Solution:** I plan to use the **Azure Database Migration Service (DMS)**.
- **Implementation:** To be honest, I haven't used DMS in a production environment before. However, I am confident I can implement it by following the official **Microsoft Learn documentation**.
- **Plan:** I will research how to configure DMS in **"Online Mode"** to replicate data continuously from AWS to Azure while the site is active, ensuring minimal impact on users.

Phase 3: Application & CI/CD

Since the application is containerized with **Docker**, the logic remains the same.

- **GitHub Actions:** I will simply update the repository **Secrets** (IP addresses and SSH keys) to point to the new Azure VMs.
- **Deployment:** The same docker-compose commands will work on the Azure VMs without code changes.

Phase 4: Cutover Strategy

To finalize the move:

1. Announce a short maintenance window.
2. Pause the application on AWS to stop new data entries.
3. Let Azure DMS finish the final data sync.
4. Update the DNS records to point to the new Azure IP.