# AWS Certified Solutions Architect – Professional Exam Guide (SAP-C02)

## Preface:

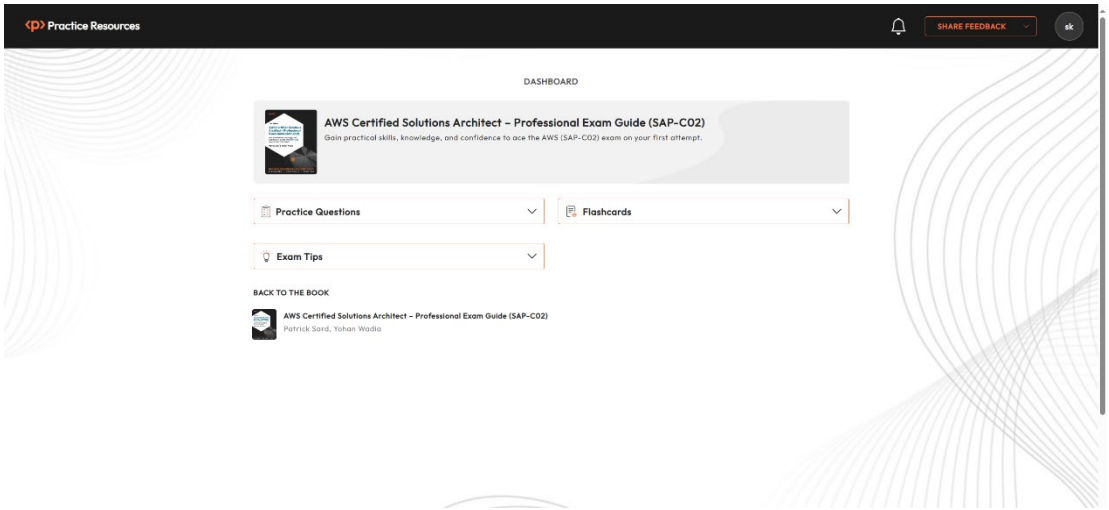


Figure 0.1: Online exam-prep platform on a desktop device

## Chapter 1: Determining an Authentication and Access Control Strategy for Complex Organizations

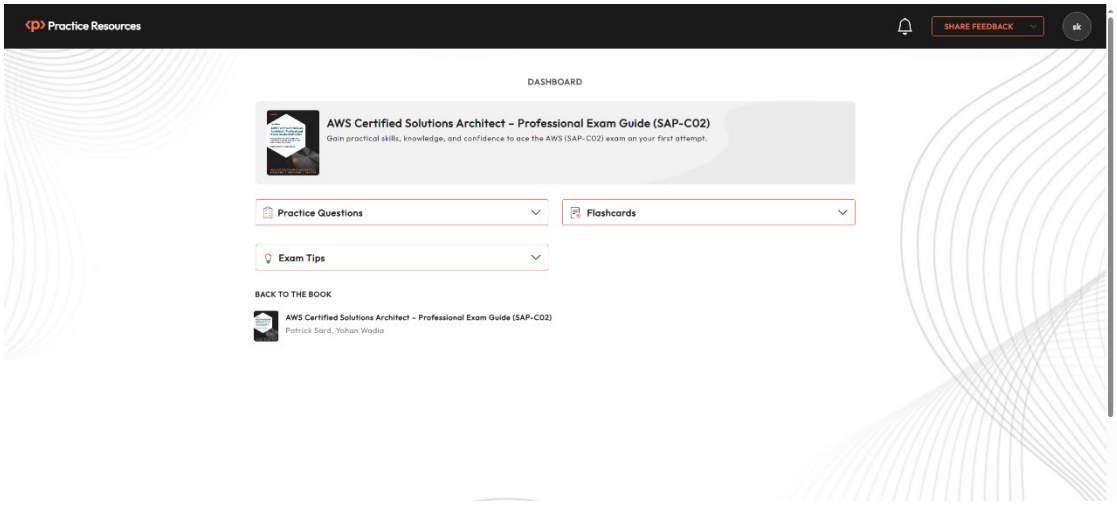


Figure 1.1: Dashboard interface of the online practice resources

# Chapter 2: Designing Networks for Complex Organizations

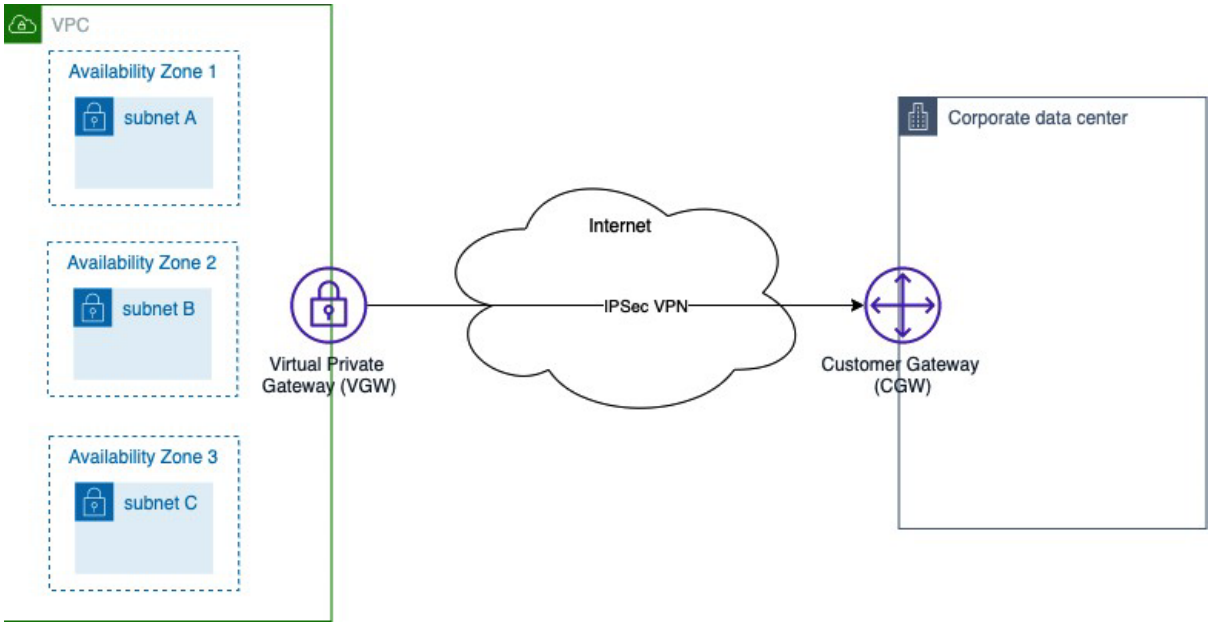


Figure 2.1: VPN connection between single VPC and on-premises equipment

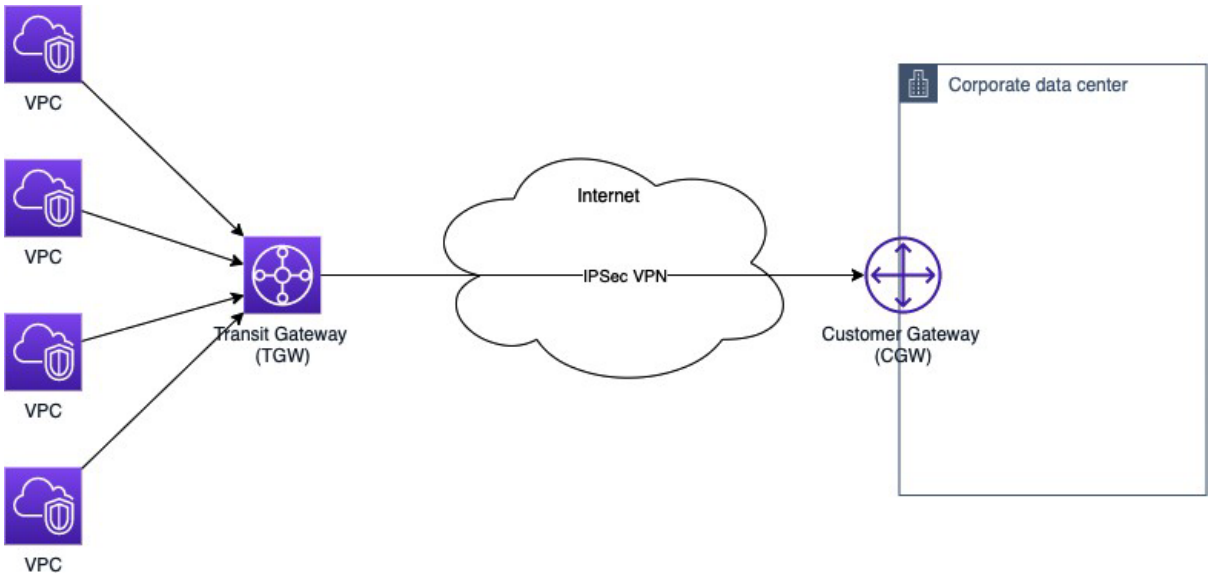


Figure 2.2: VPN connection between TGW and on-premises equipment

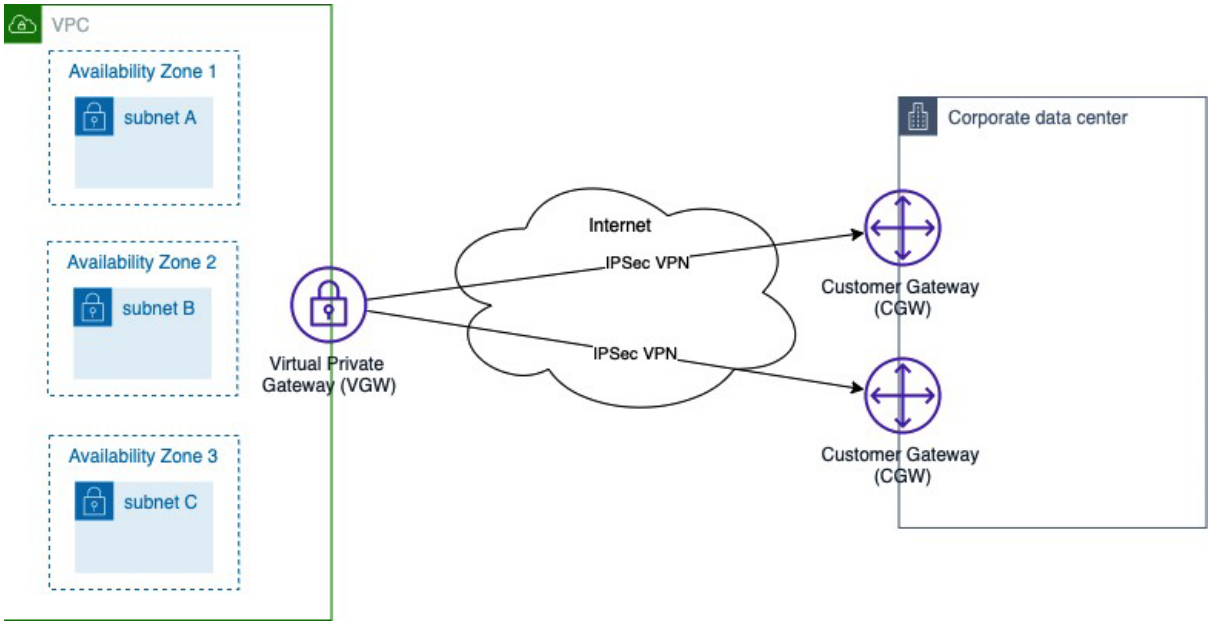


Figure 2.3: VPN connection redundancy for failover

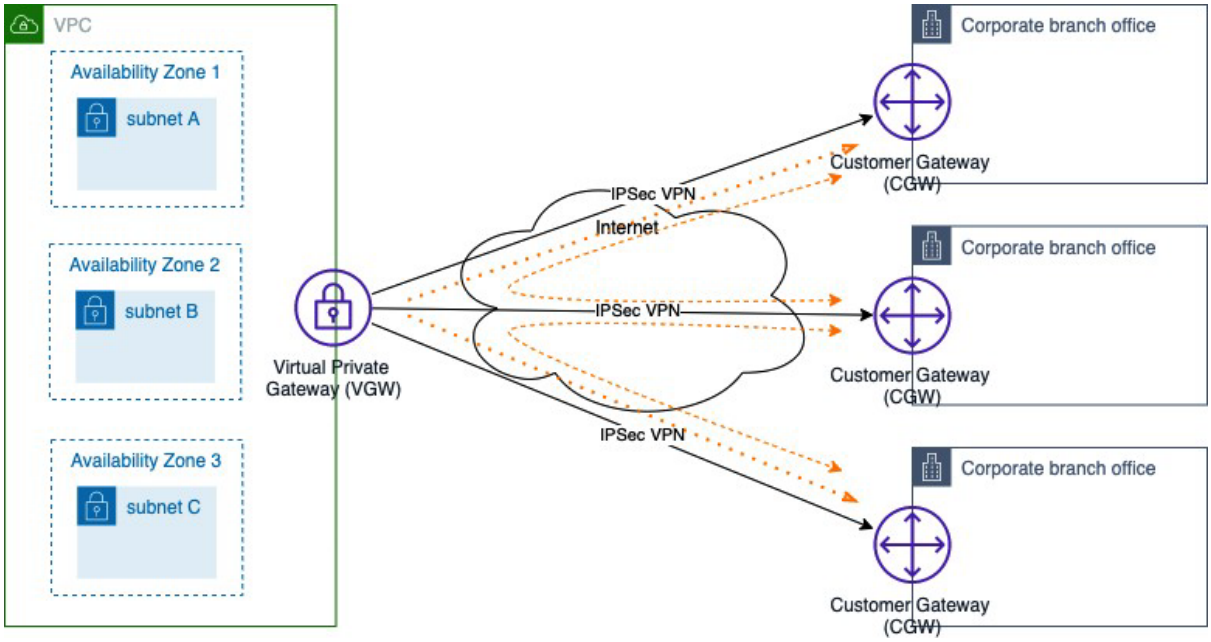


Figure 2.4: Hub-and-spoke VPN

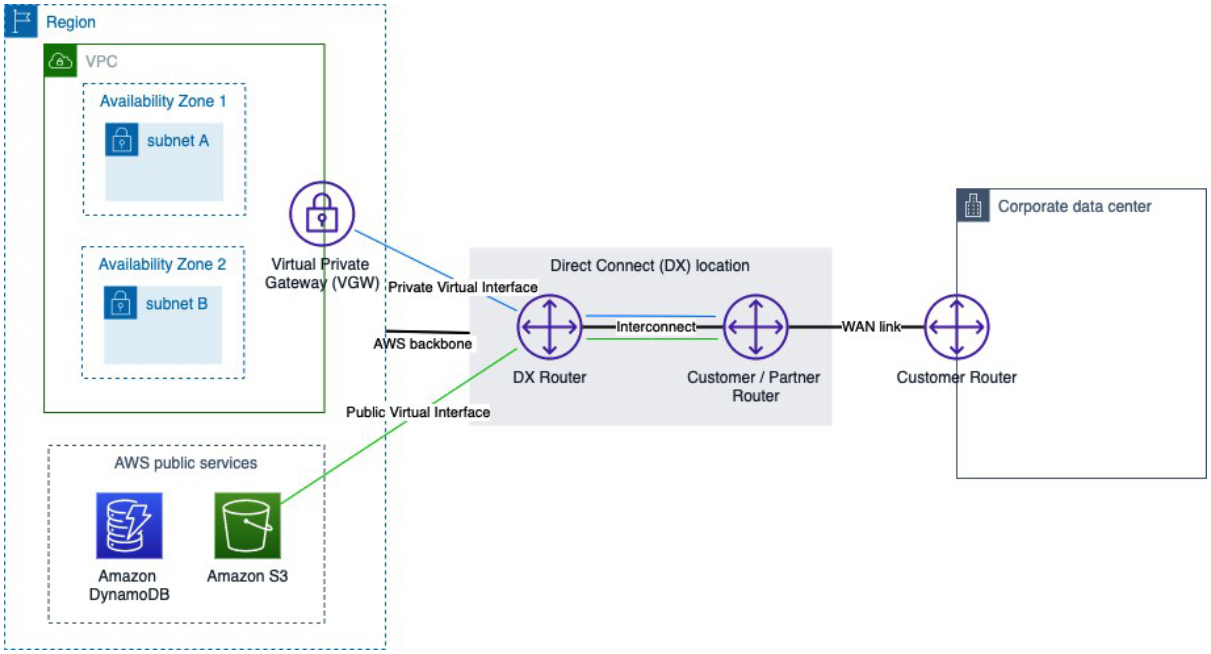


Figure 2.5: Public and private VIFs

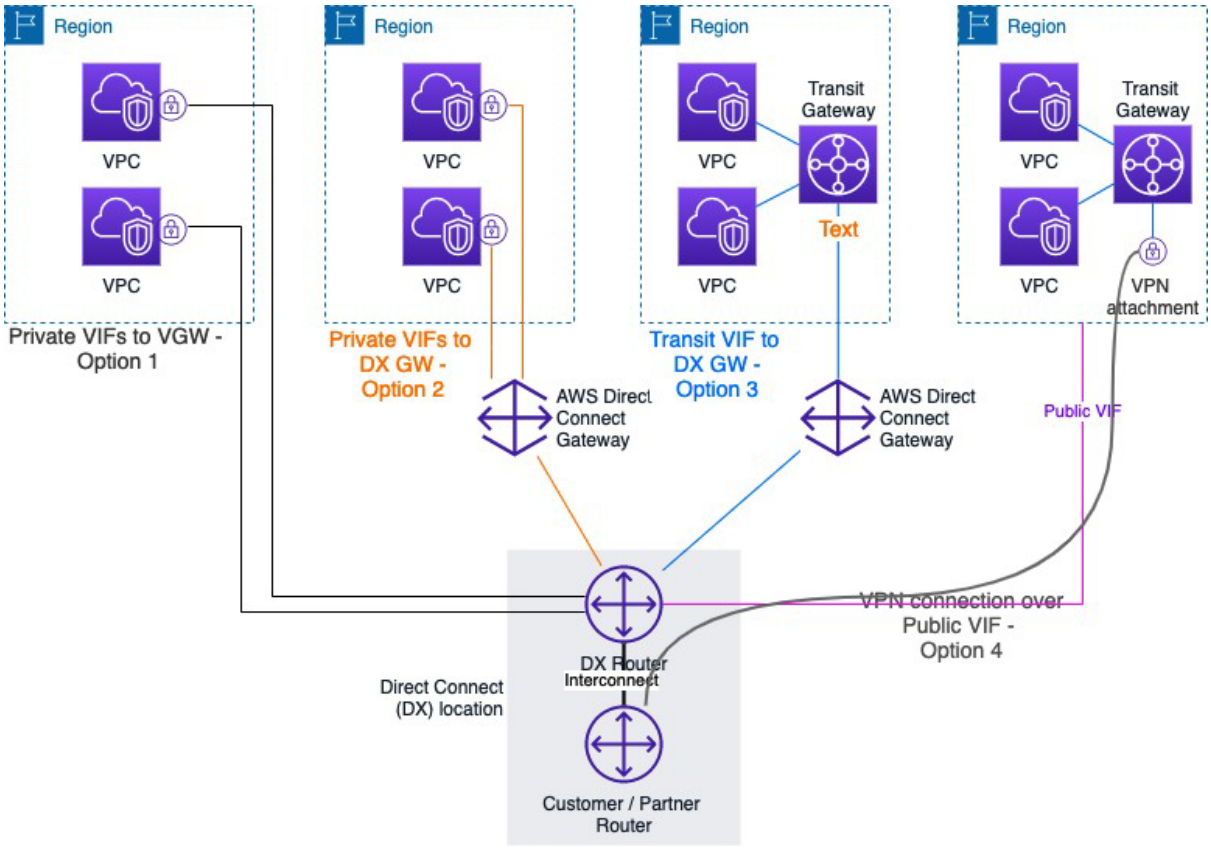


Figure 2.6: DX options summary

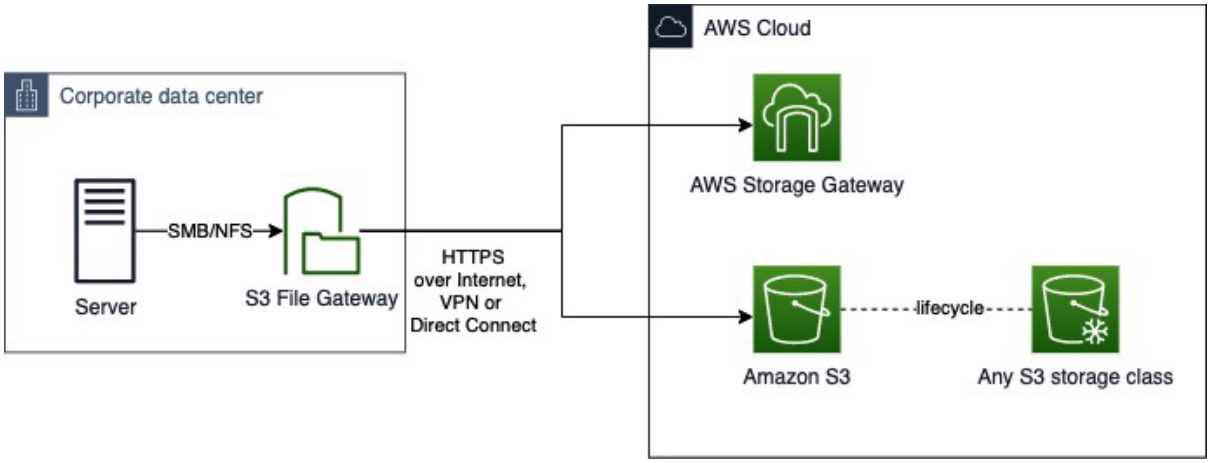


Figure 2.7: Amazon S3 File Gateway

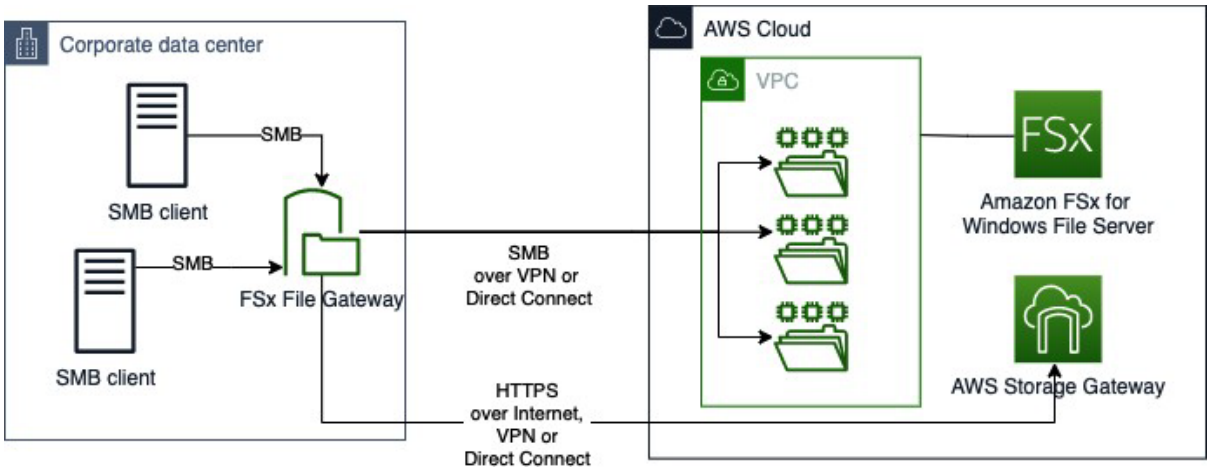


Figure 2.8: FSx File Gateway

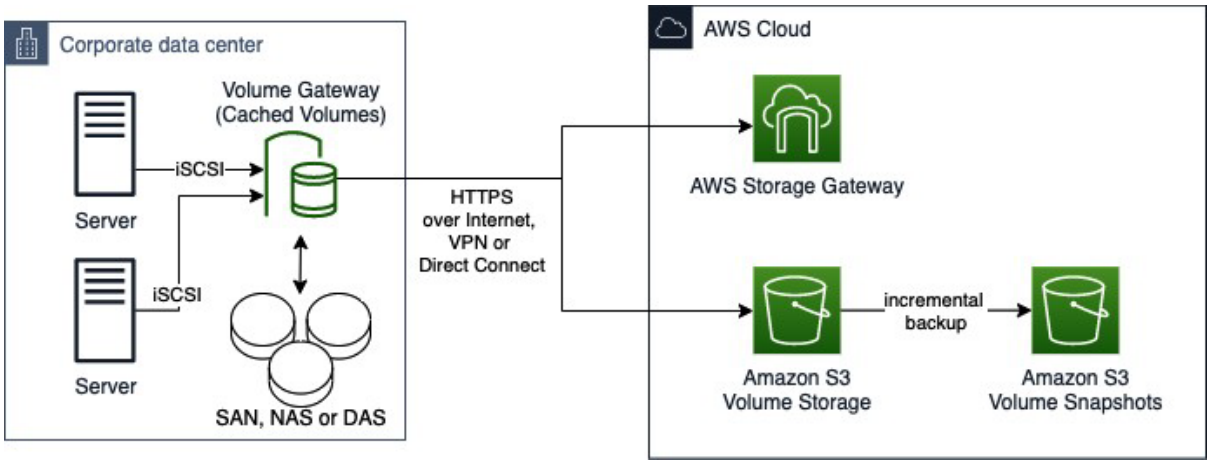


Figure 2.9: Volume Gateway (cached volumes)

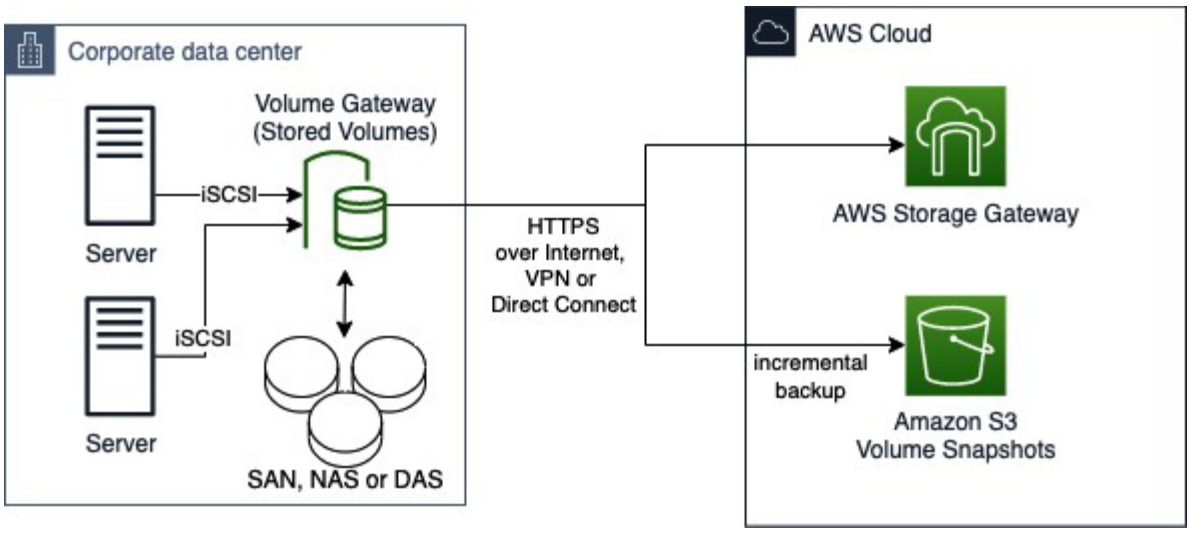


Figure 2.10: Volume Gateway (stored volumes)

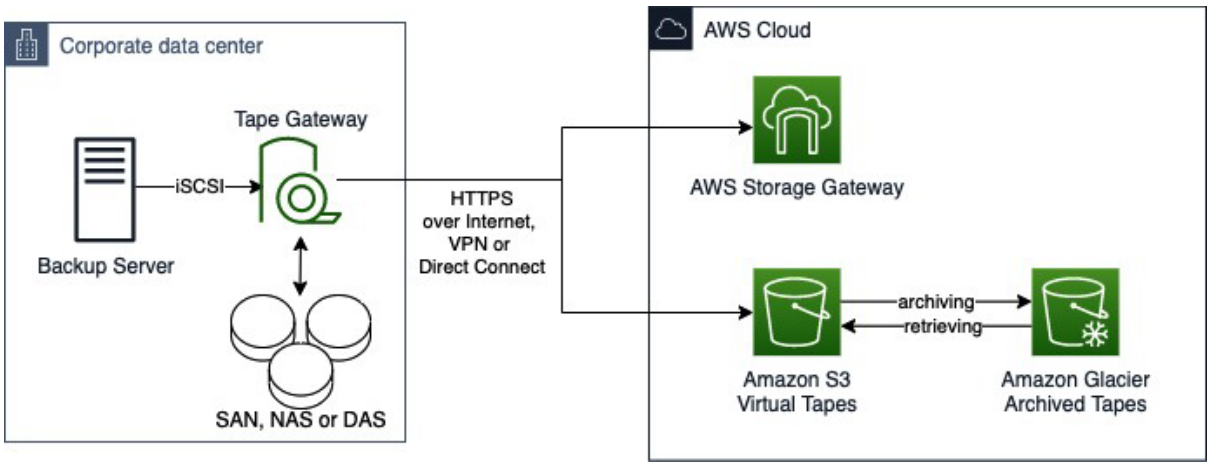


Figure 2.11: Tape gateway

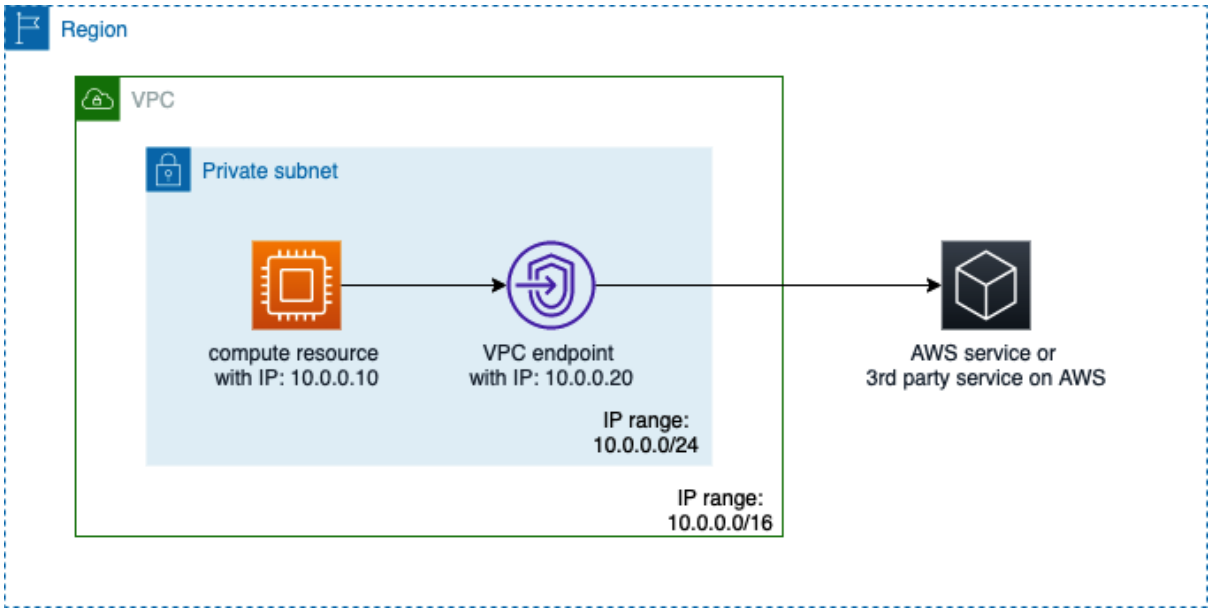


Figure 2.12: VPC endpoint

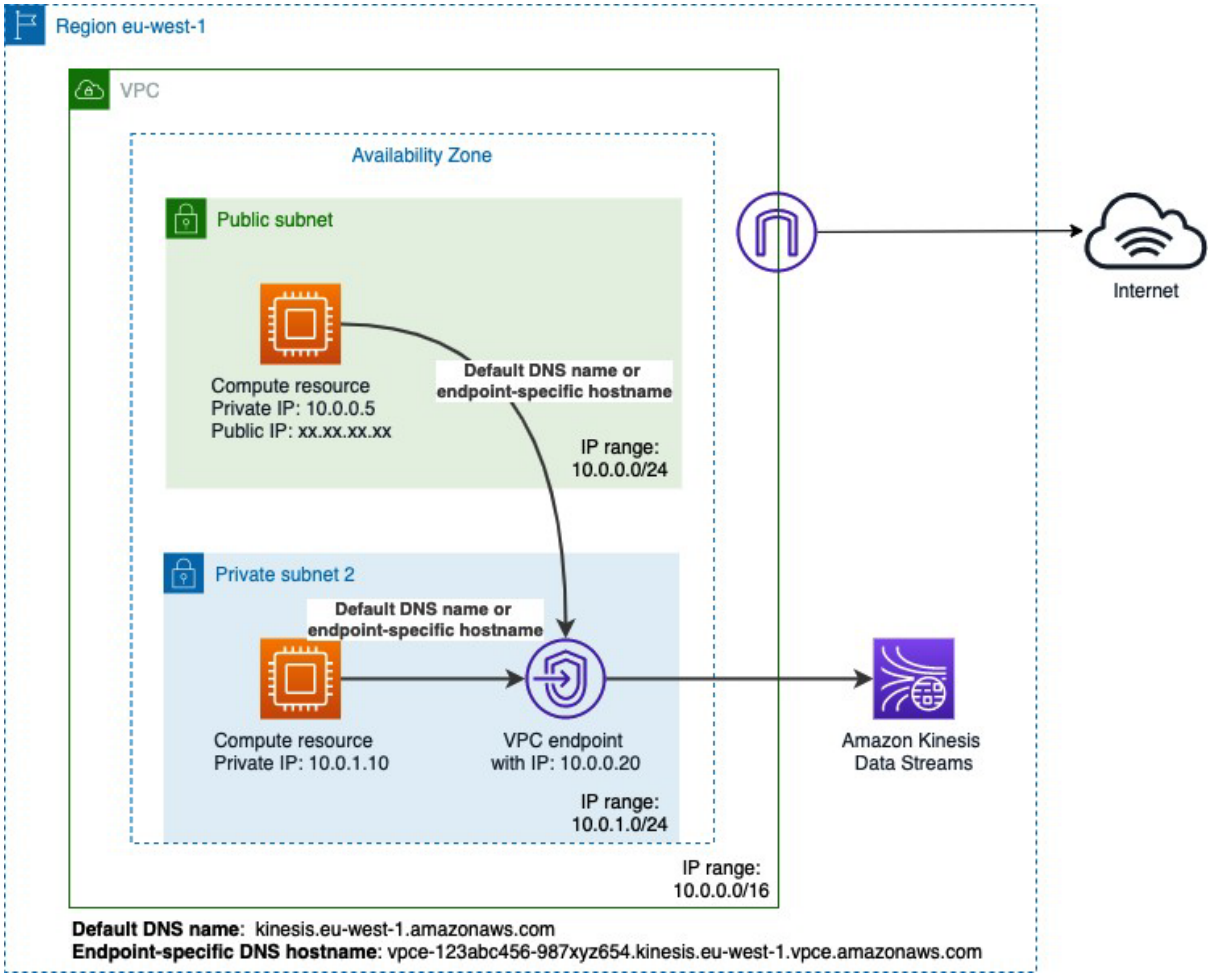


Figure 2.13: VPC interface endpoints and DNS names

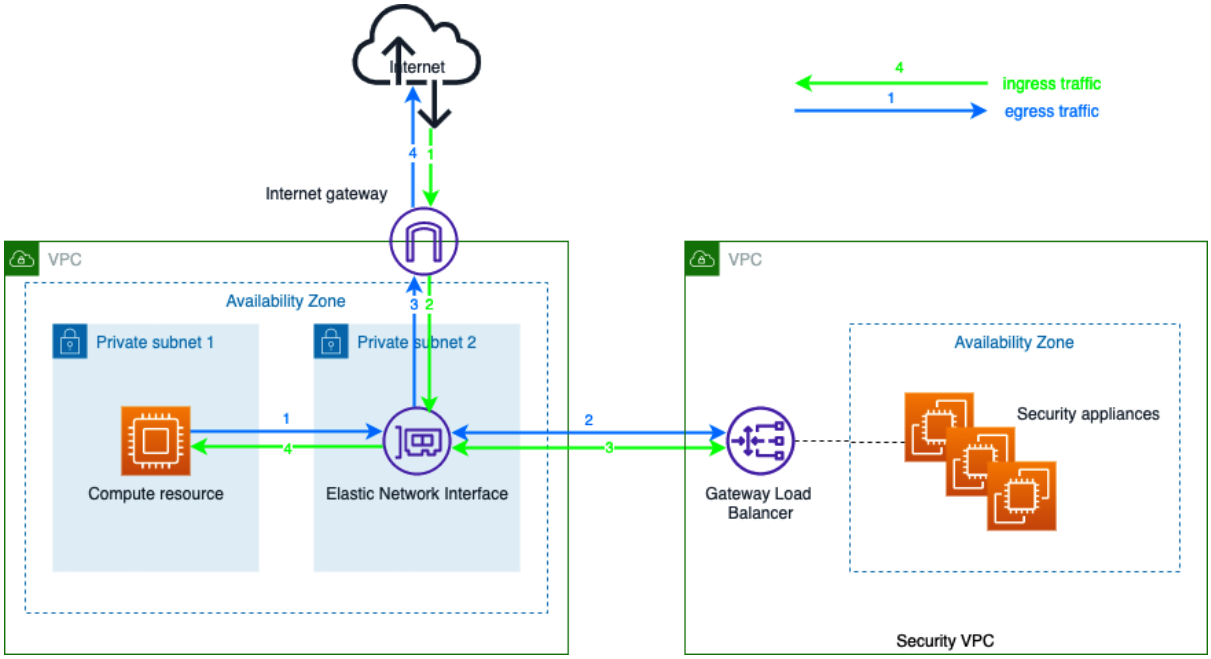


Figure 2.14: GWLB endpoint

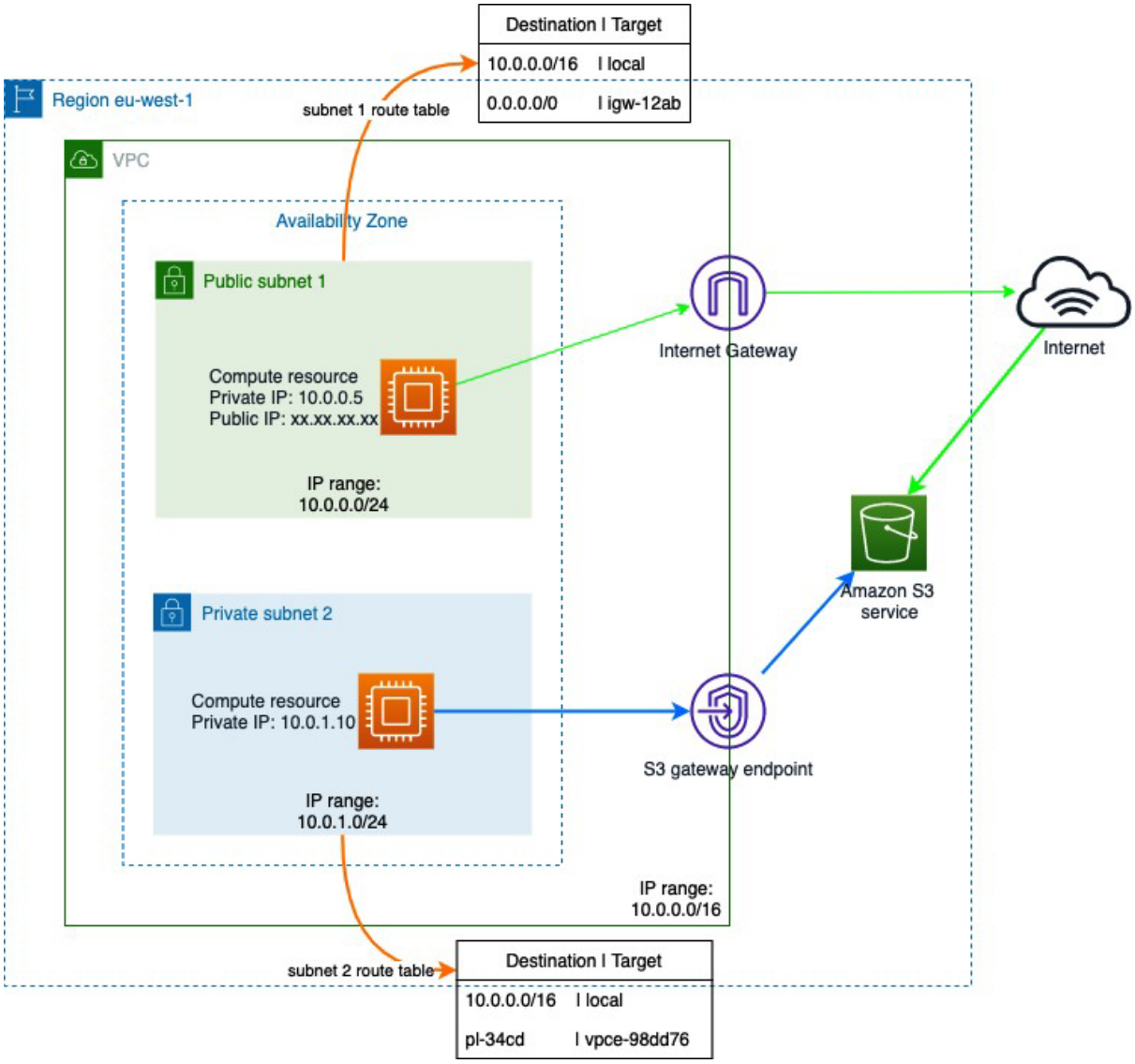


Figure 2.15: VPC gateway endpoint

## Chapter 3: Designing a Multi-Account AWS Environment for Complex Organizations

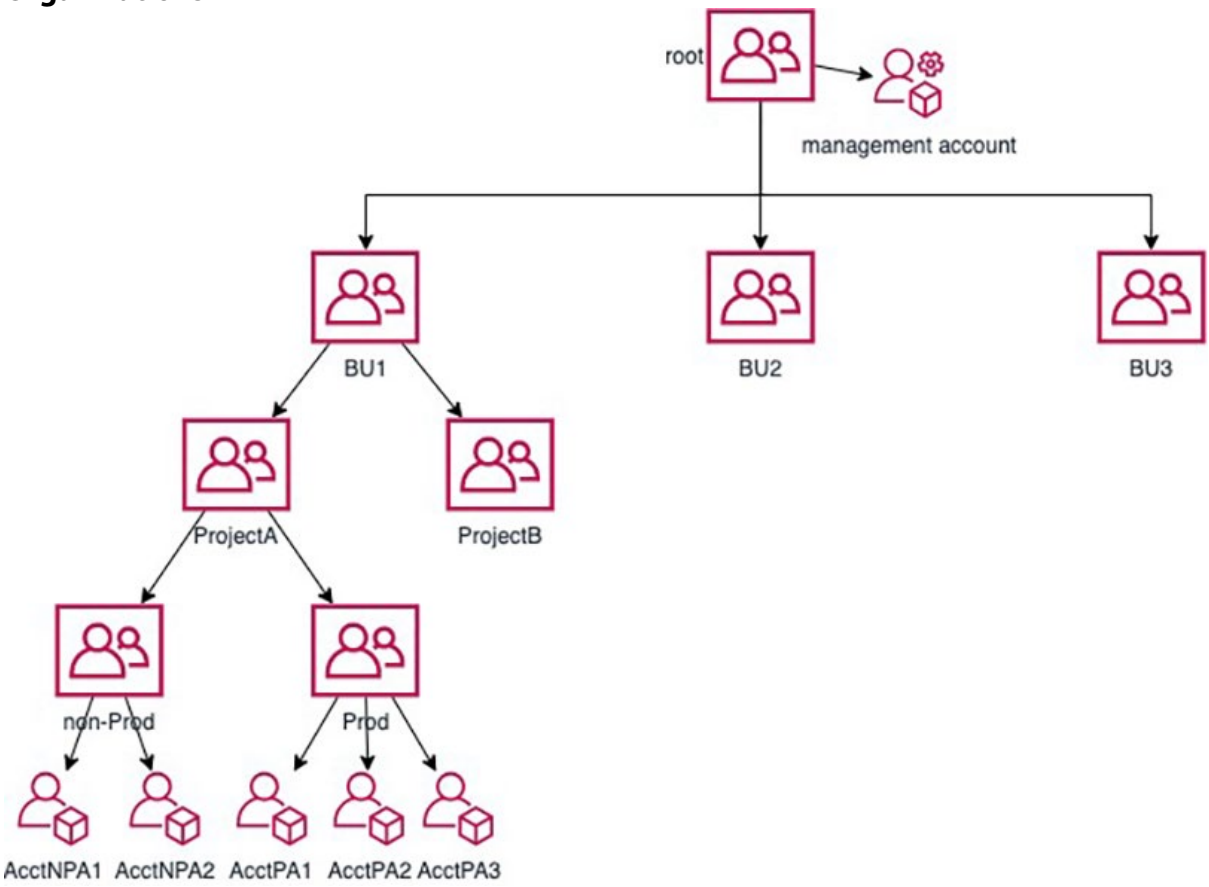


Figure 3.1: Example of account structure

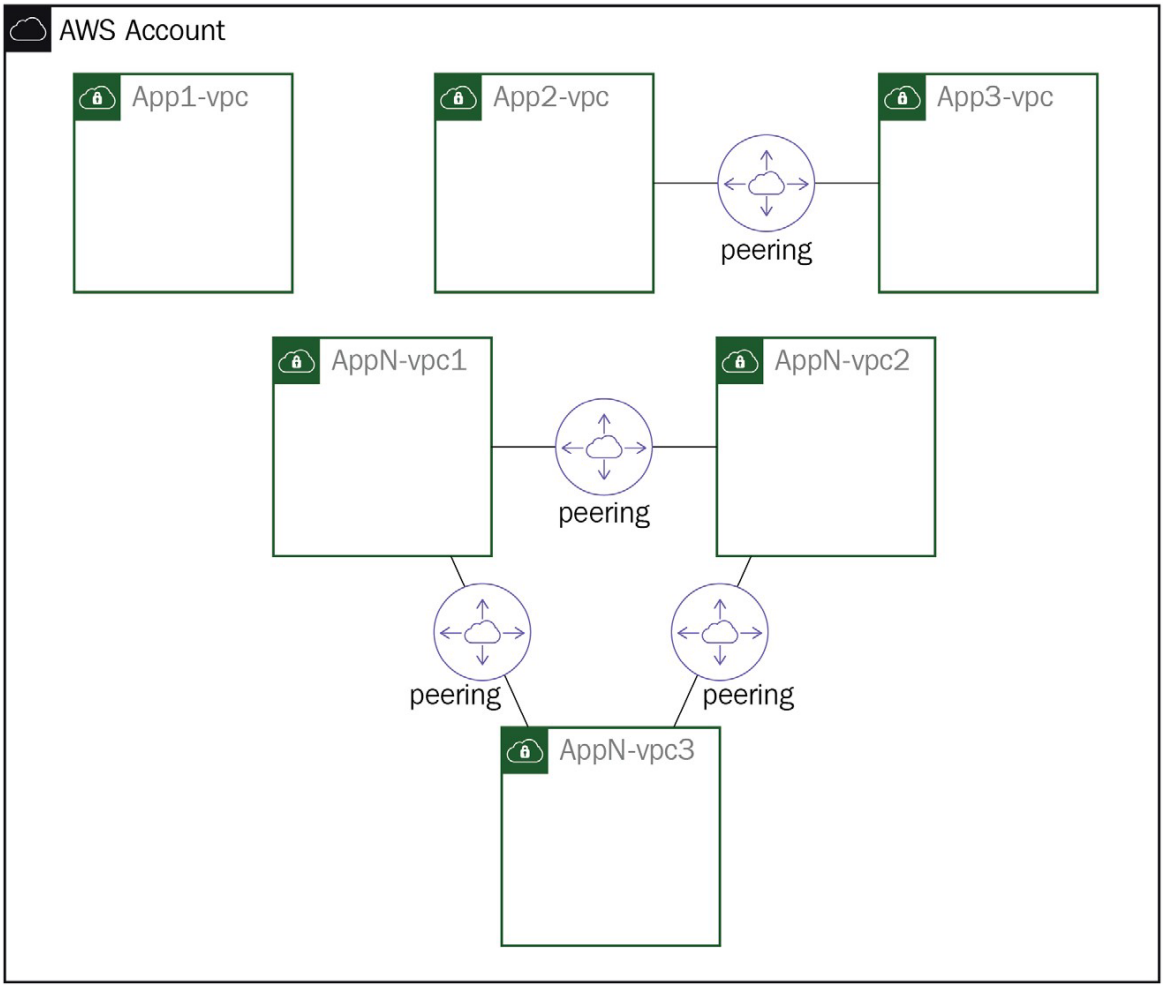


Figure 3.2: Example of VPC organization

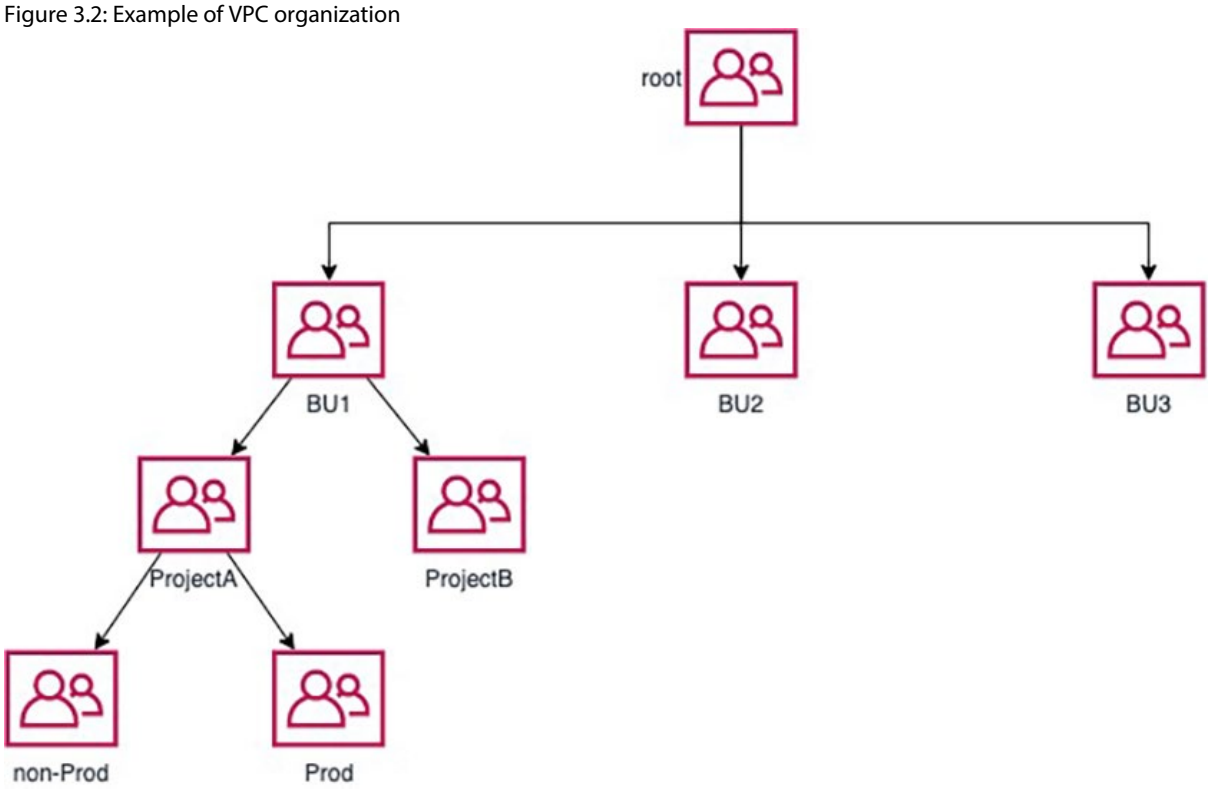


Figure 3.3: BU style

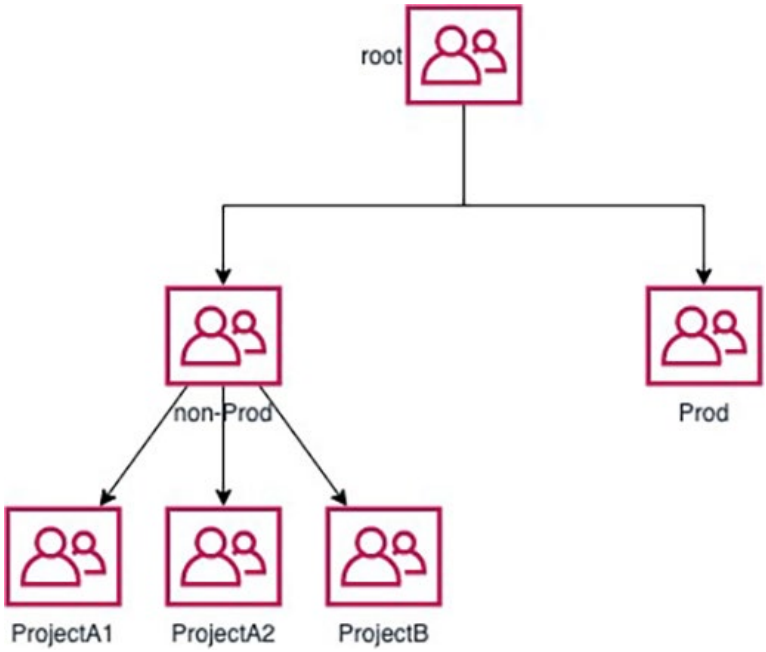


Figure 3.4: Environment style

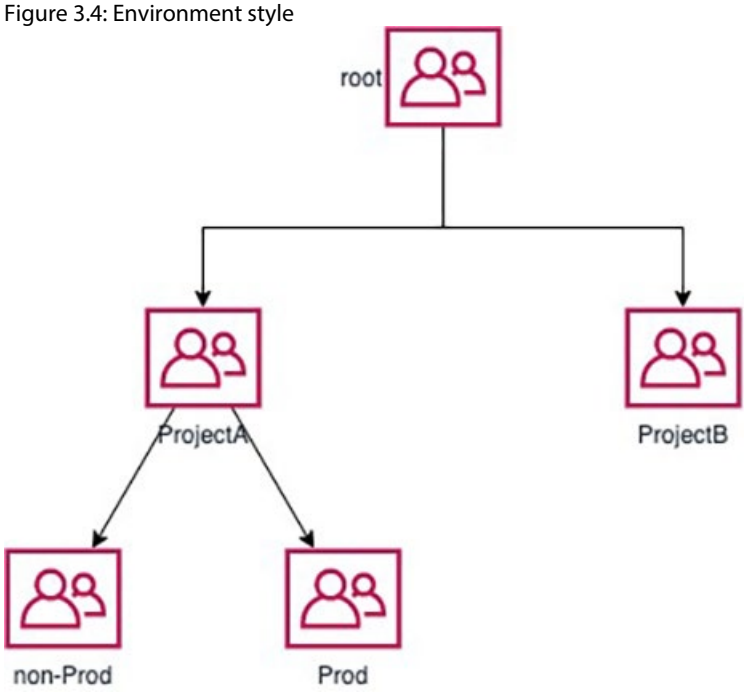


Figure 3.5: Project style

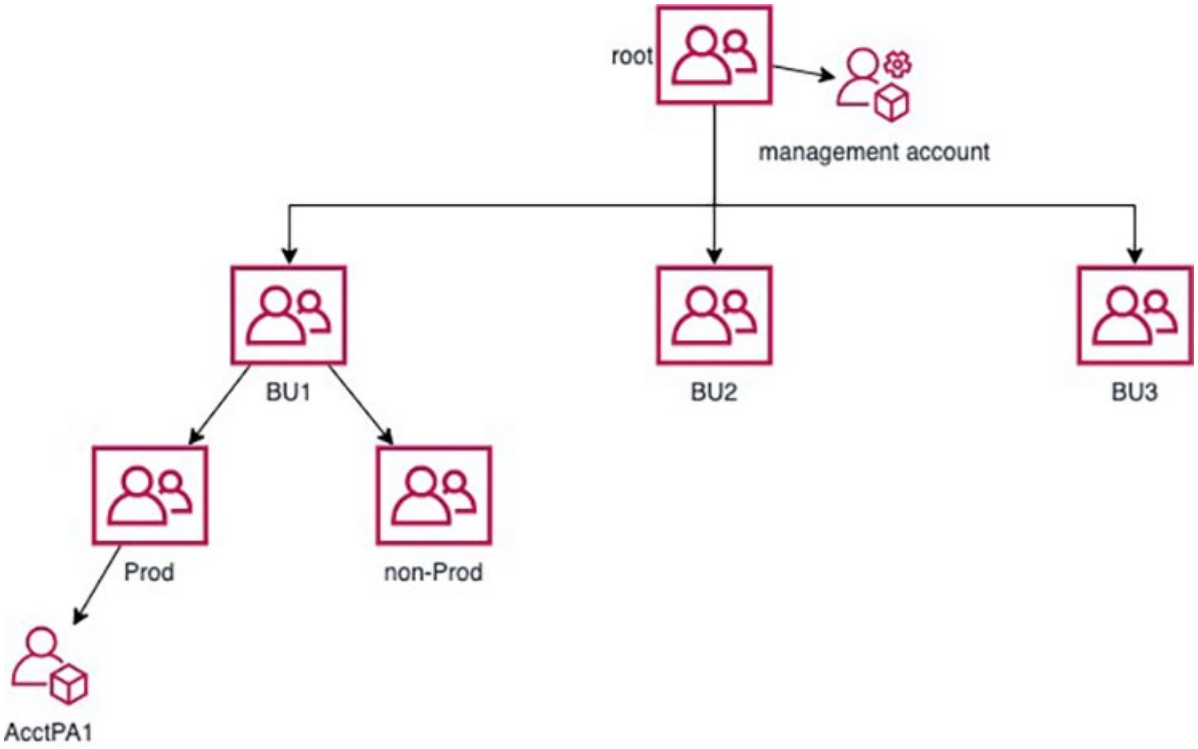


Figure 3.6 – Example organizational structure

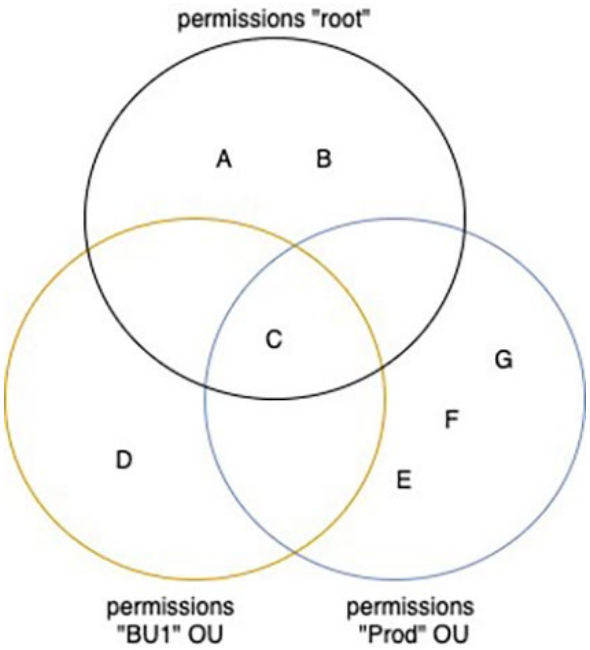


Figure 3.7: OU permissions intersection

# Chapter 4: Ensuring Cost Optimization

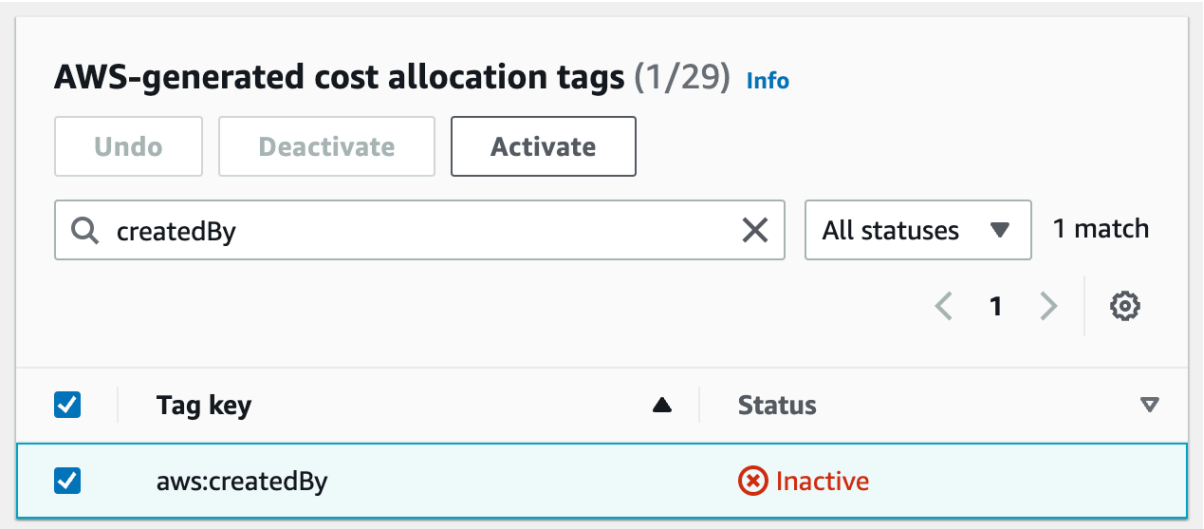


Figure 4.1: AWS-generated cost allocation tag enablement

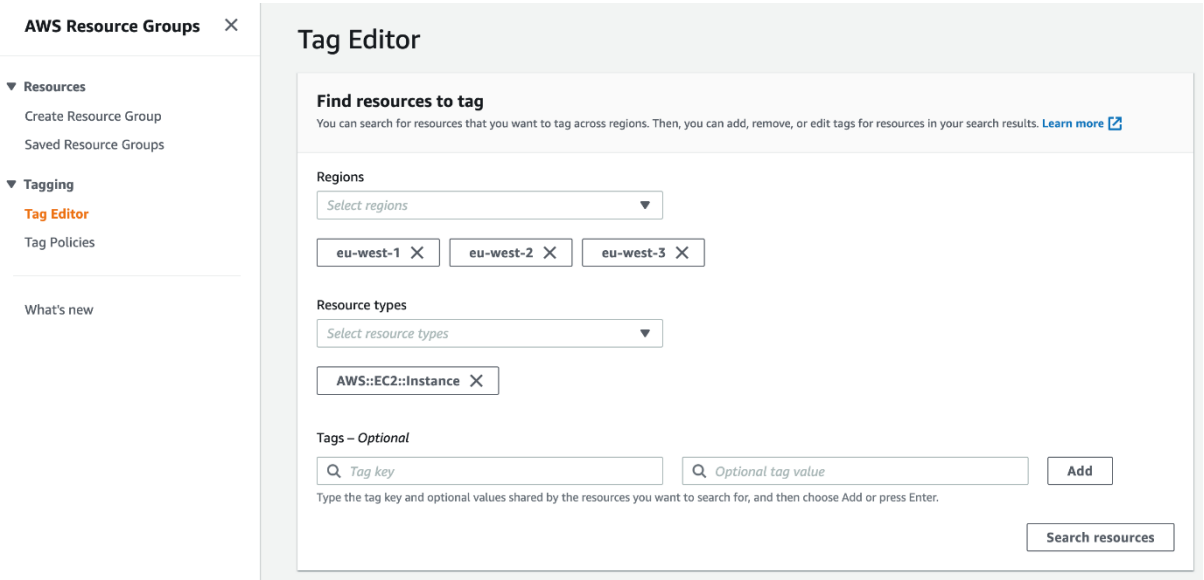


Figure 4.2: AWS Tag Editor dashboard

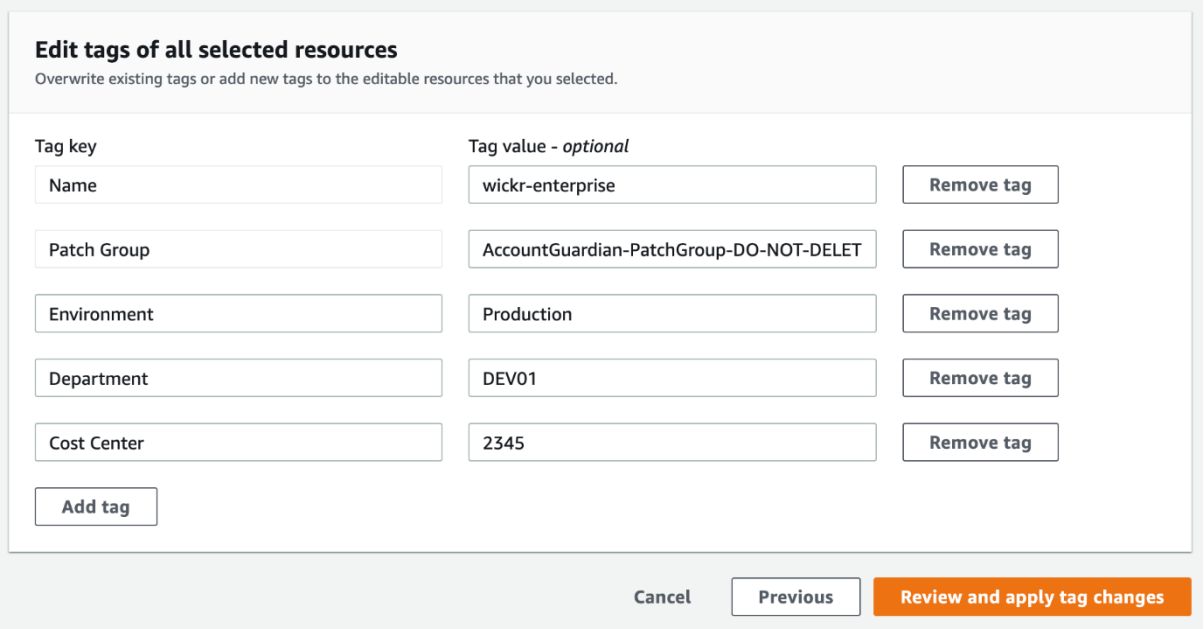


Figure 4.3: Applying tags to resources

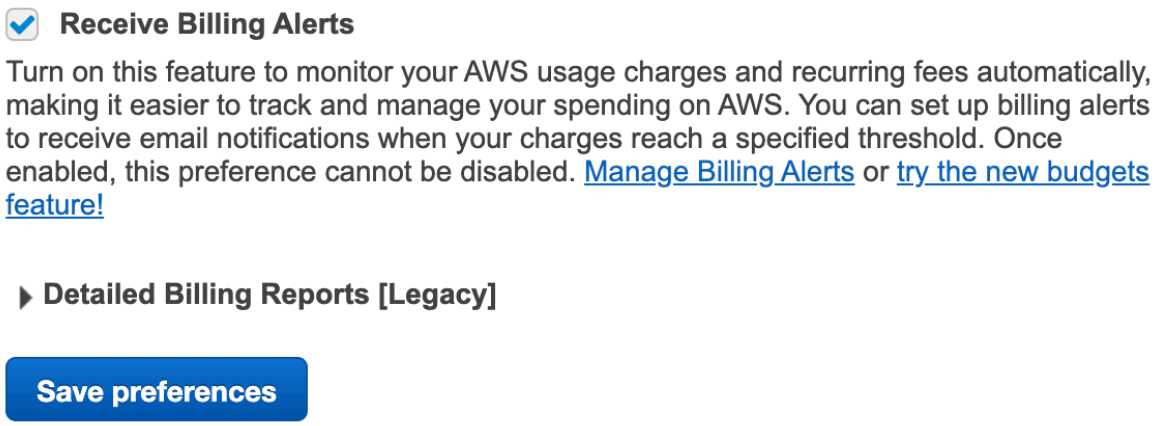


Figure 4.4: Enabling billing alerts using the AWS Billing dashboard

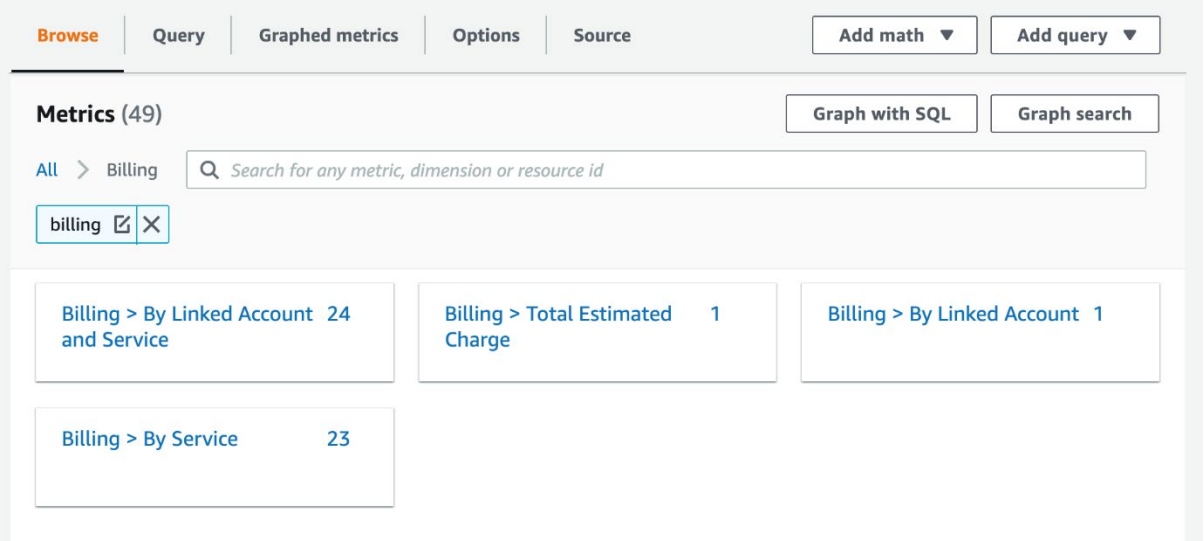


Figure 4.5: Total estimate billing alarm configuration

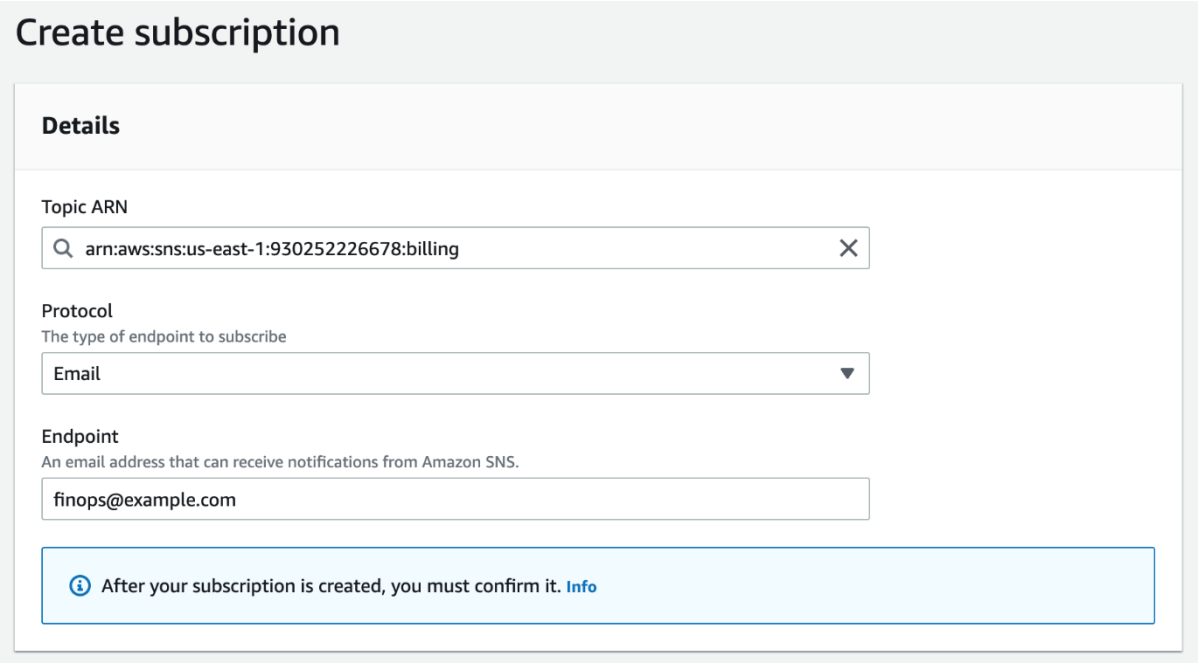


Figure 4.6: Billing notification configuration using Amazon SNS

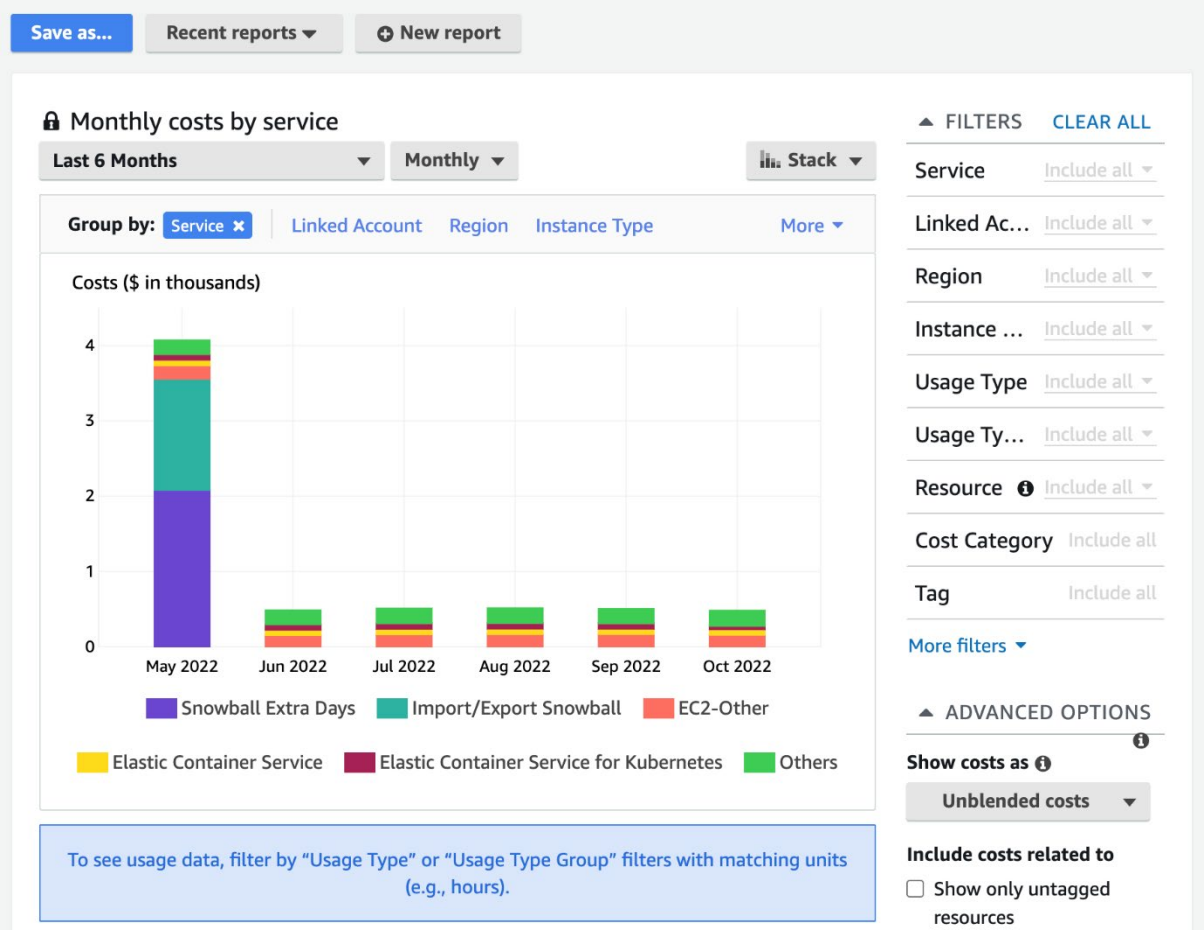


Figure 4.7: Cost Explorer "Monthly costs by service" view

## Chapter 5: Determining Security Requirements and Controls

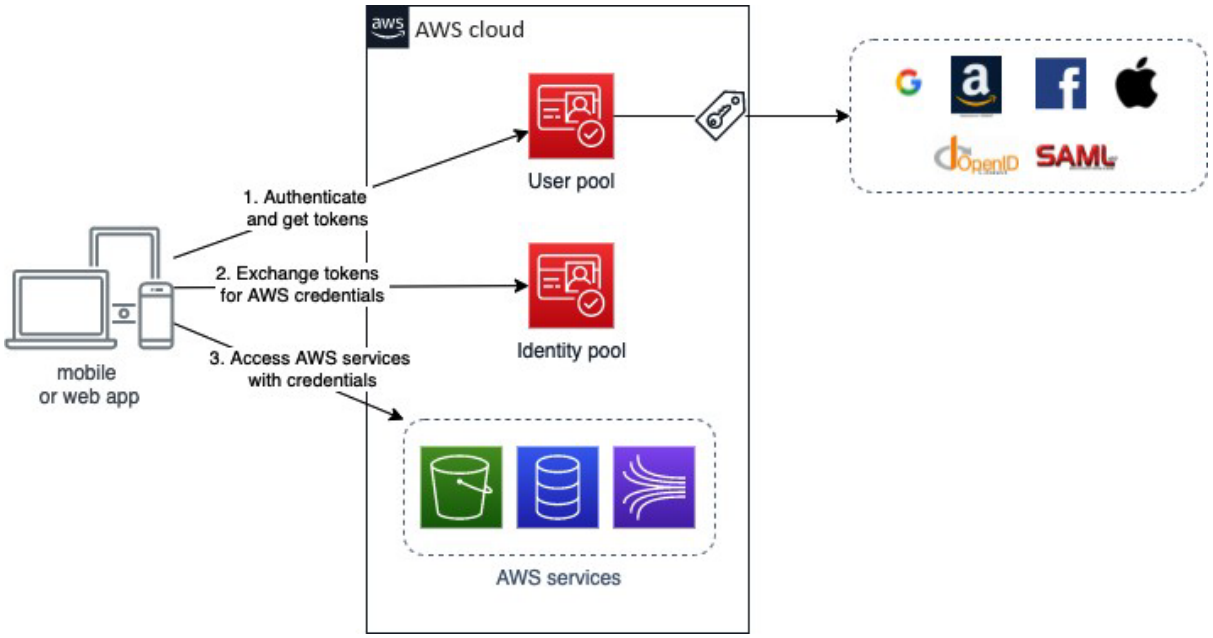


Figure 5.1: Amazon Cognito concepts overview

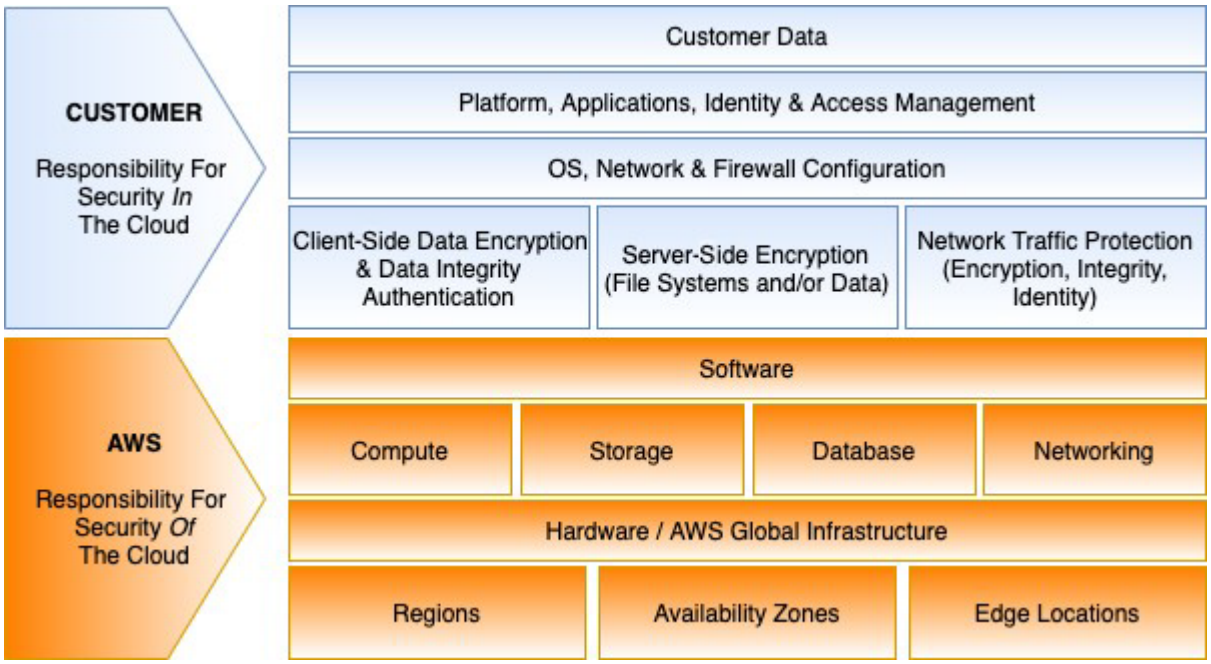


Figure 5.2: AWS shared responsibility model

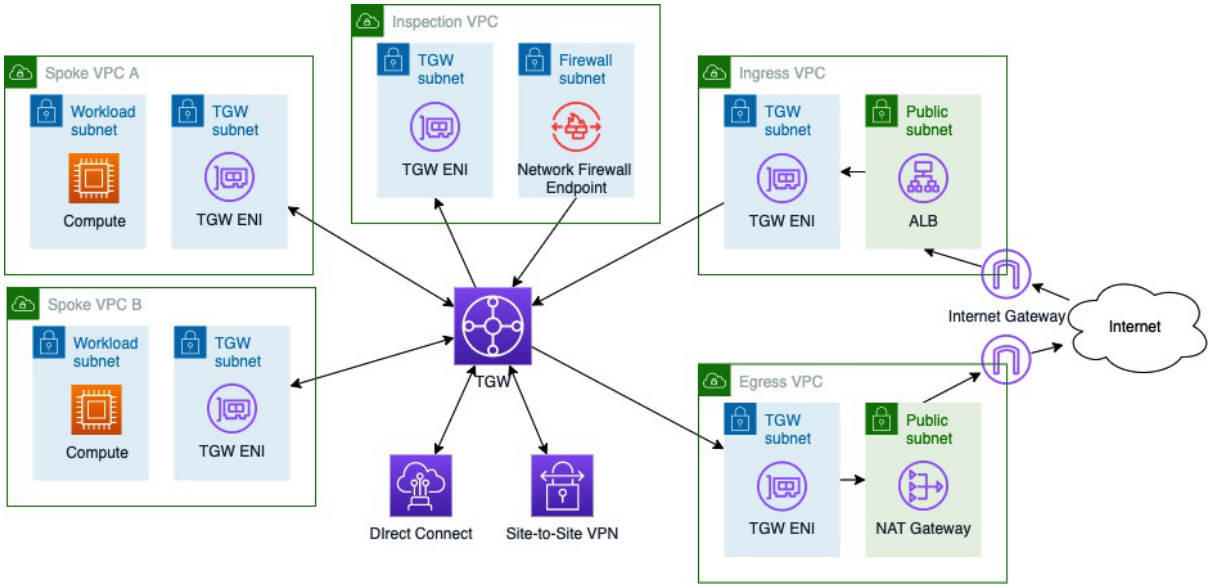


Figure 5.3: Illustration of a centralized setup with TGW and inspection via Network Firewall (NFW)

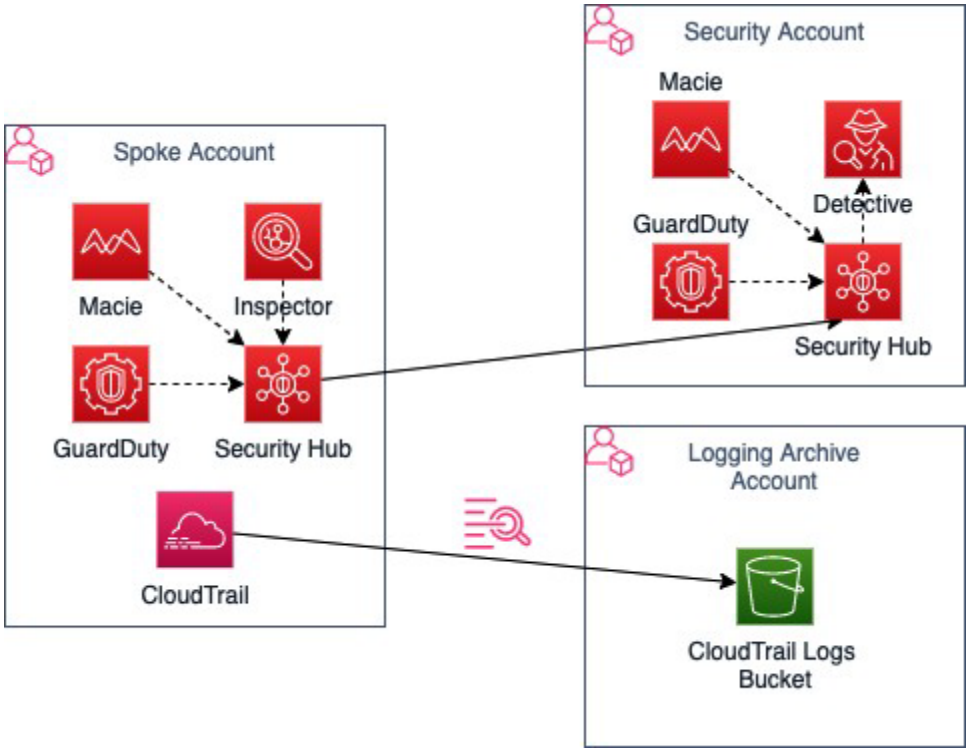


Figure 5.4: Incident detection and centralized CloudTrail logs

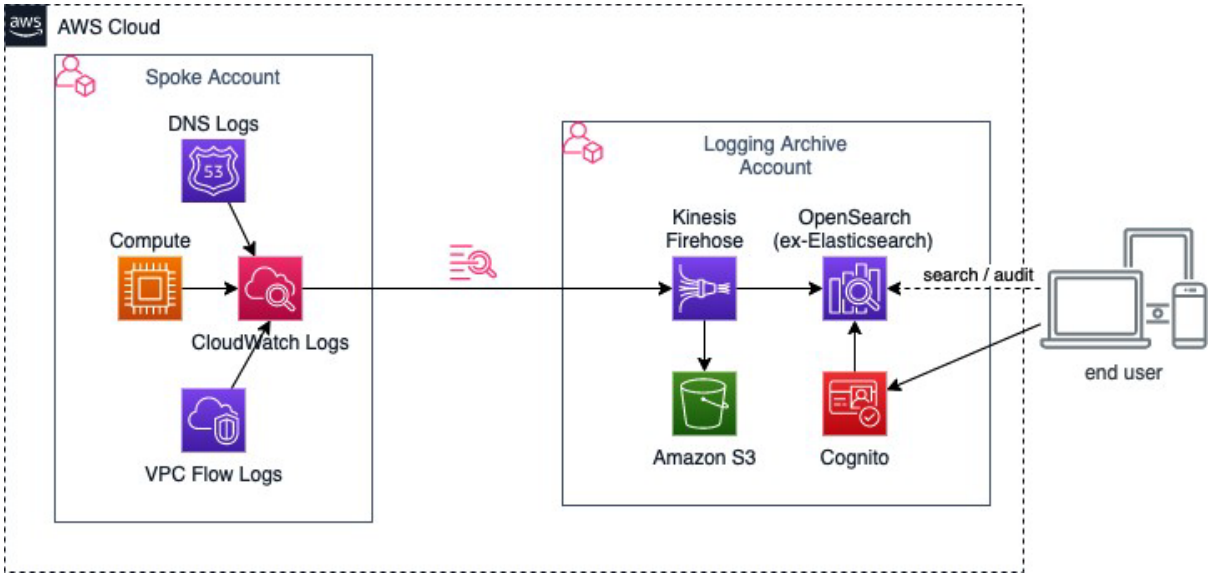


Figure 5.5 – Centralized logging and log indexing for search and audit

## Chapter 6: Meeting Reliability Requirements

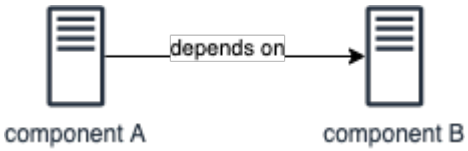


Figure 6.1: Tight coupling

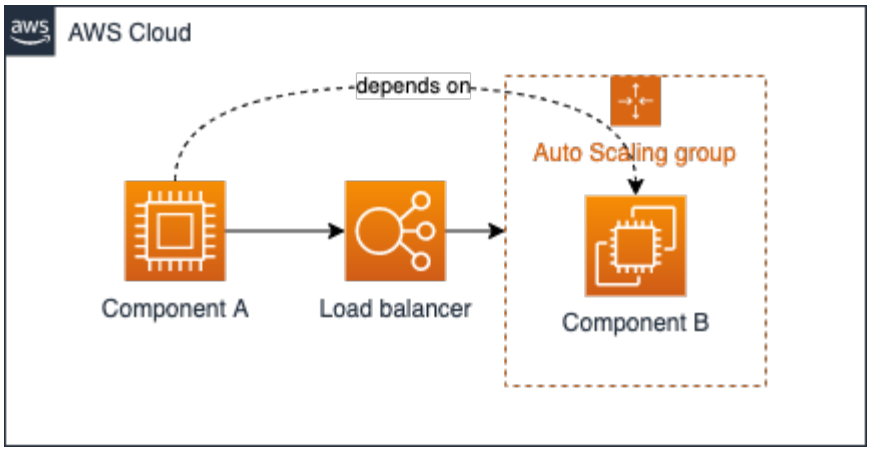


Figure 6.2: Loose coupling (synchronous communication)

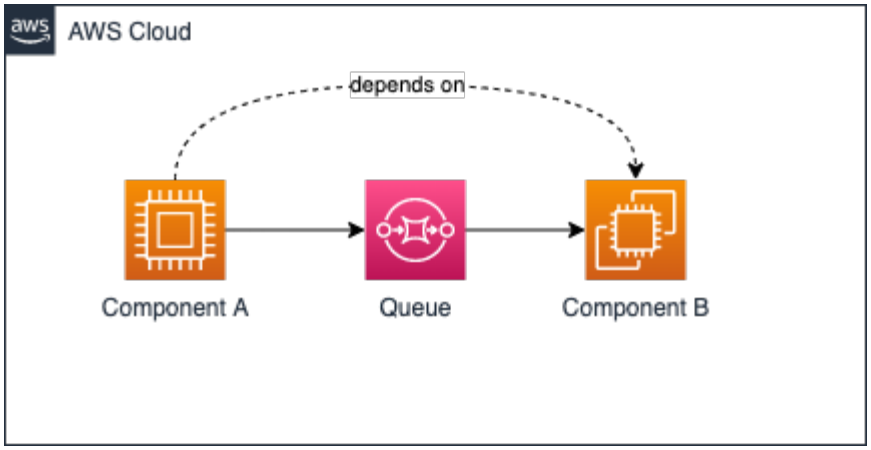


Figure 6.3: Loose coupling (asynchronous communication)

# Chapter 7: Ensuring Business Continuity

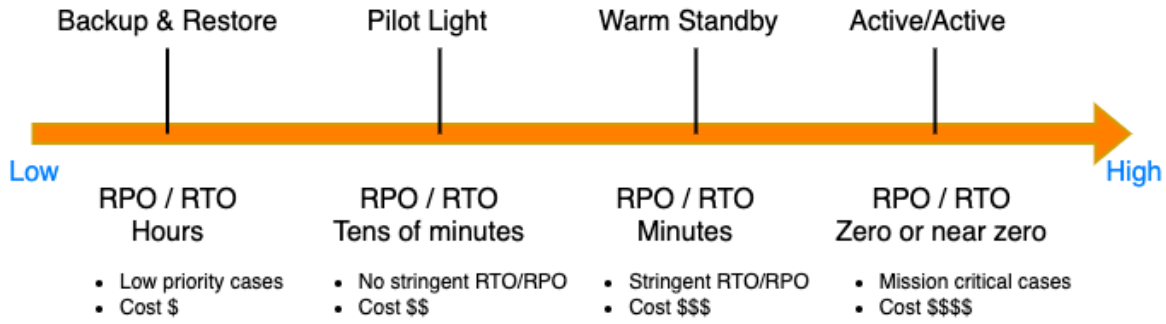


Figure 7.1: DR strategies cost and complexity

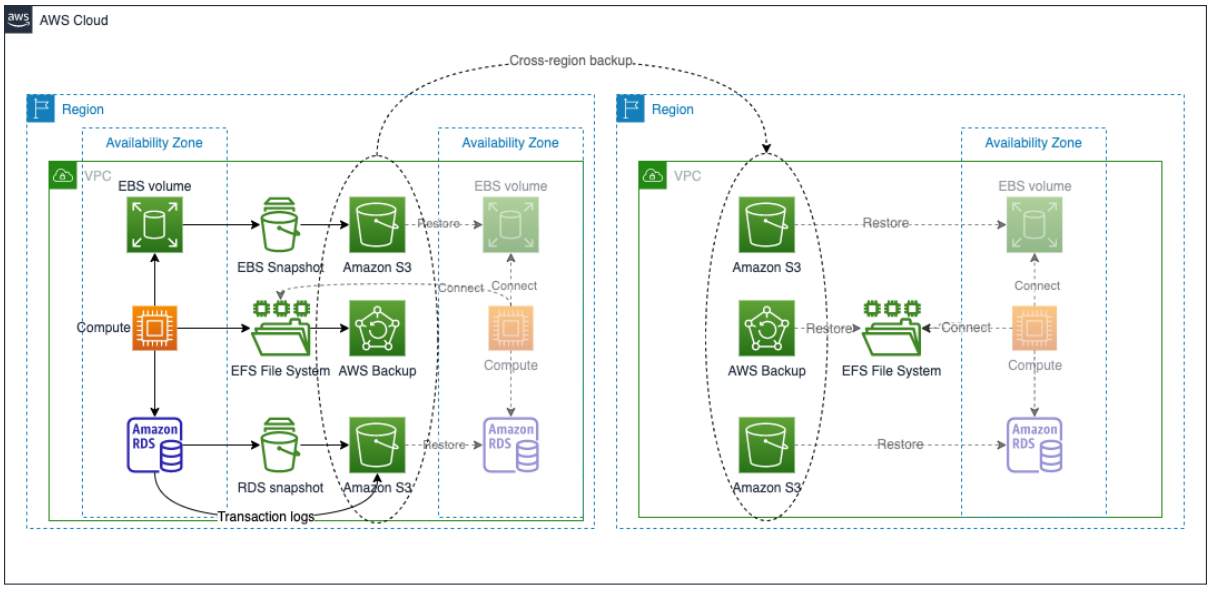


Figure 7.2: Backup and restore approach

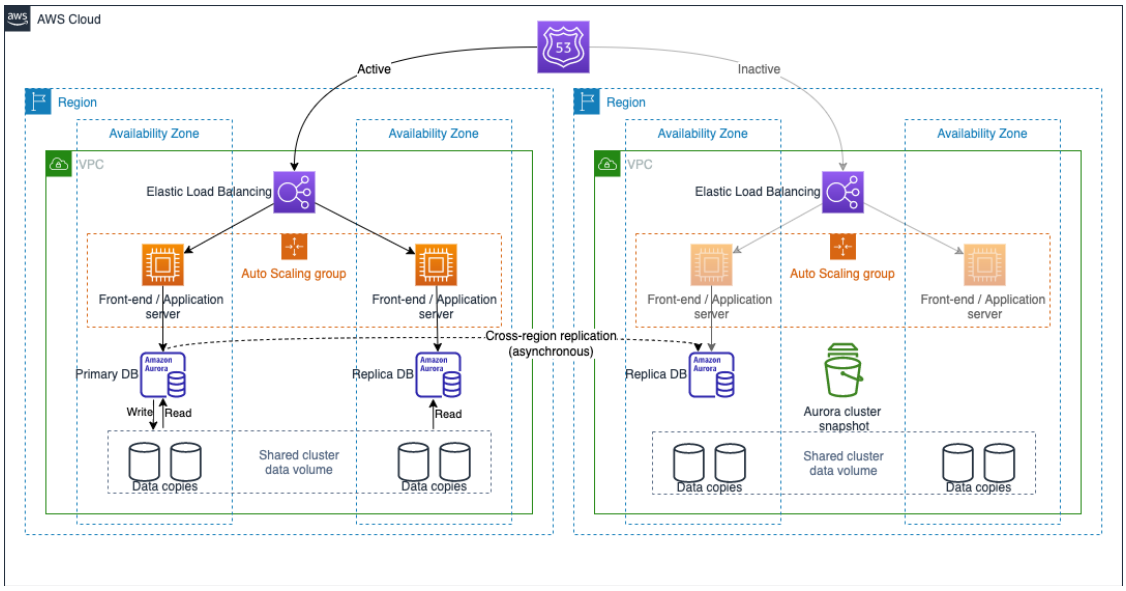


Figure 7.3: Pilot light approach

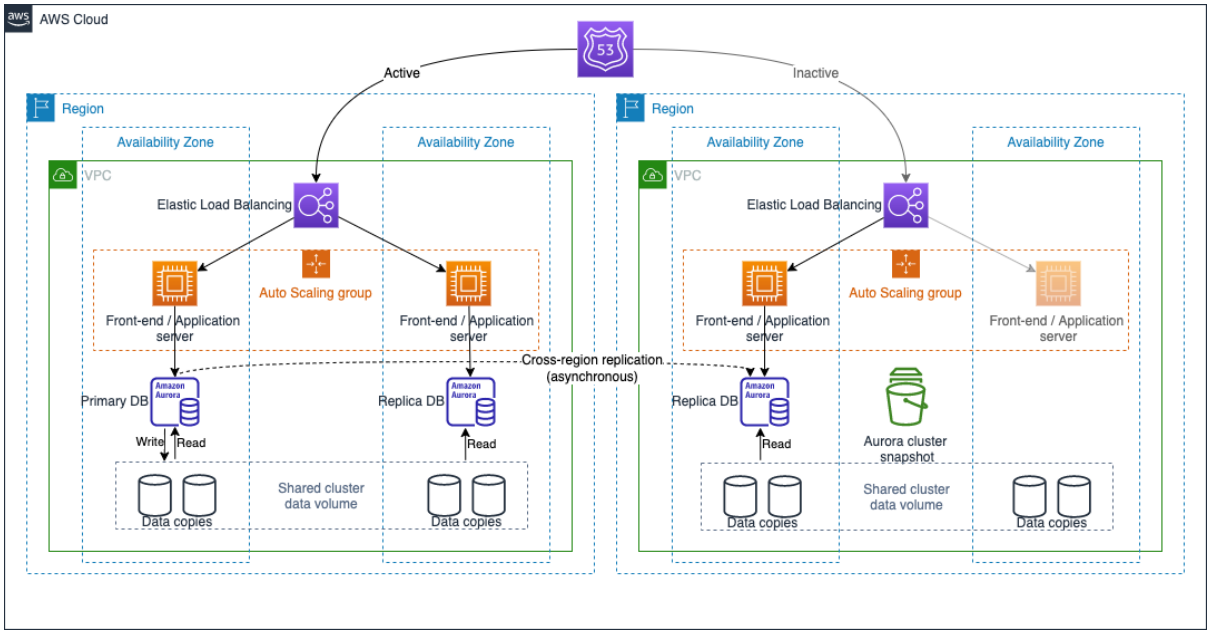


Figure 7.4: Warm standby approach

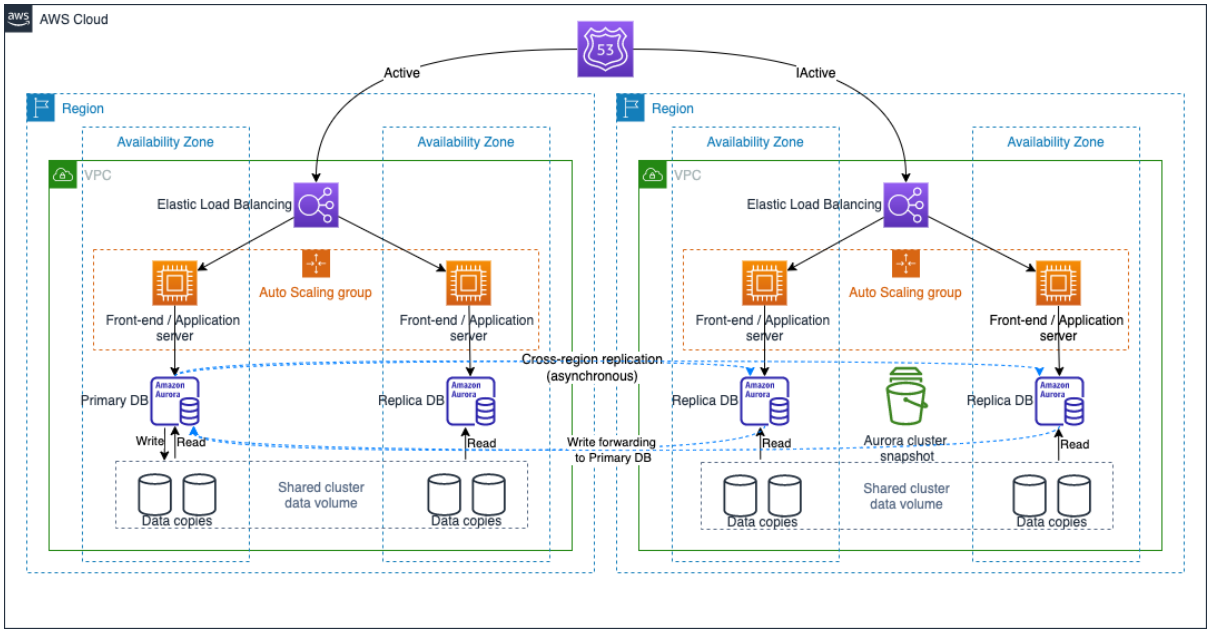


Figure 7.5: Active-active approach

## Chapter 8: Meeting Performance Objectives

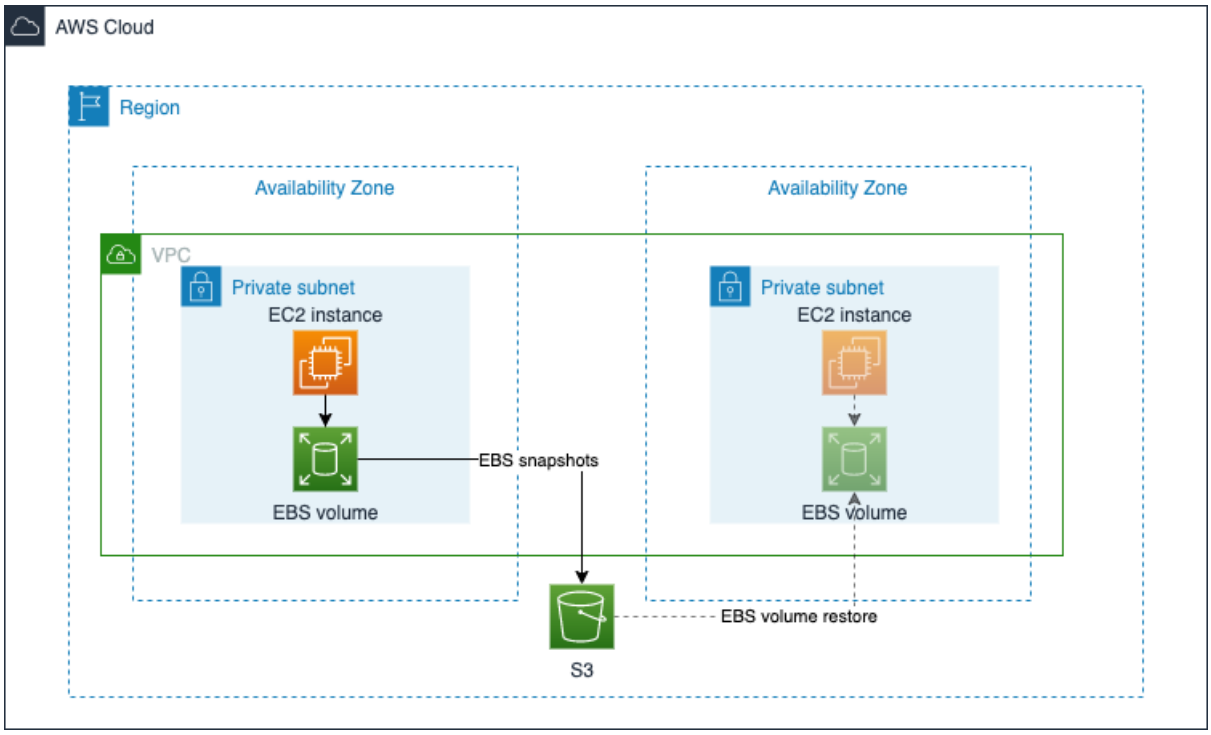


Figure 8.1: EBS volume attach, snapshot, and restore

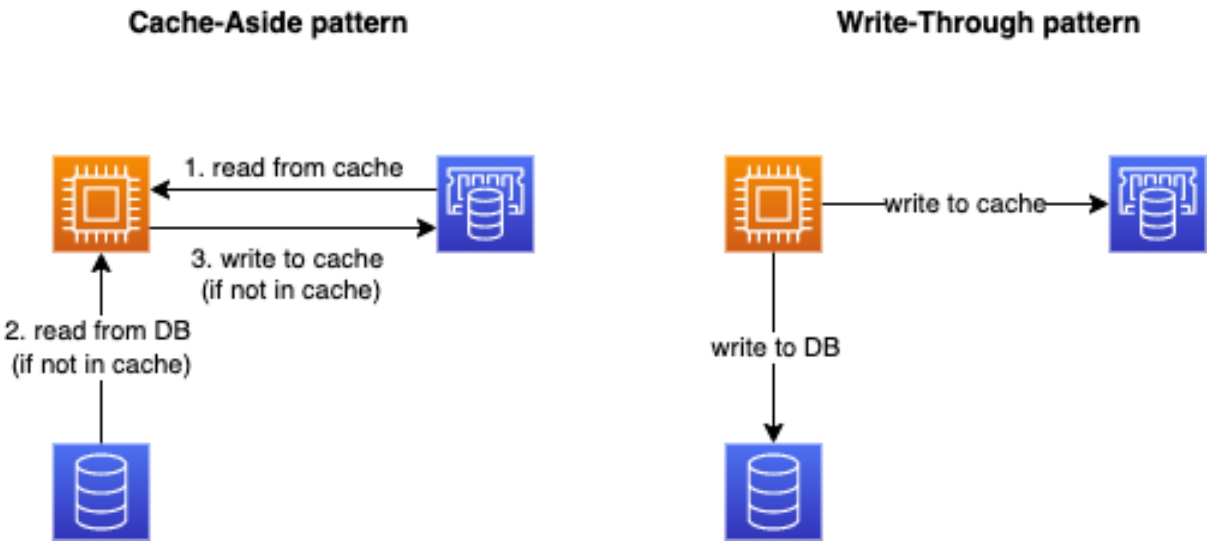


Figure 8.2: Cache patterns: cache-aside and write-through

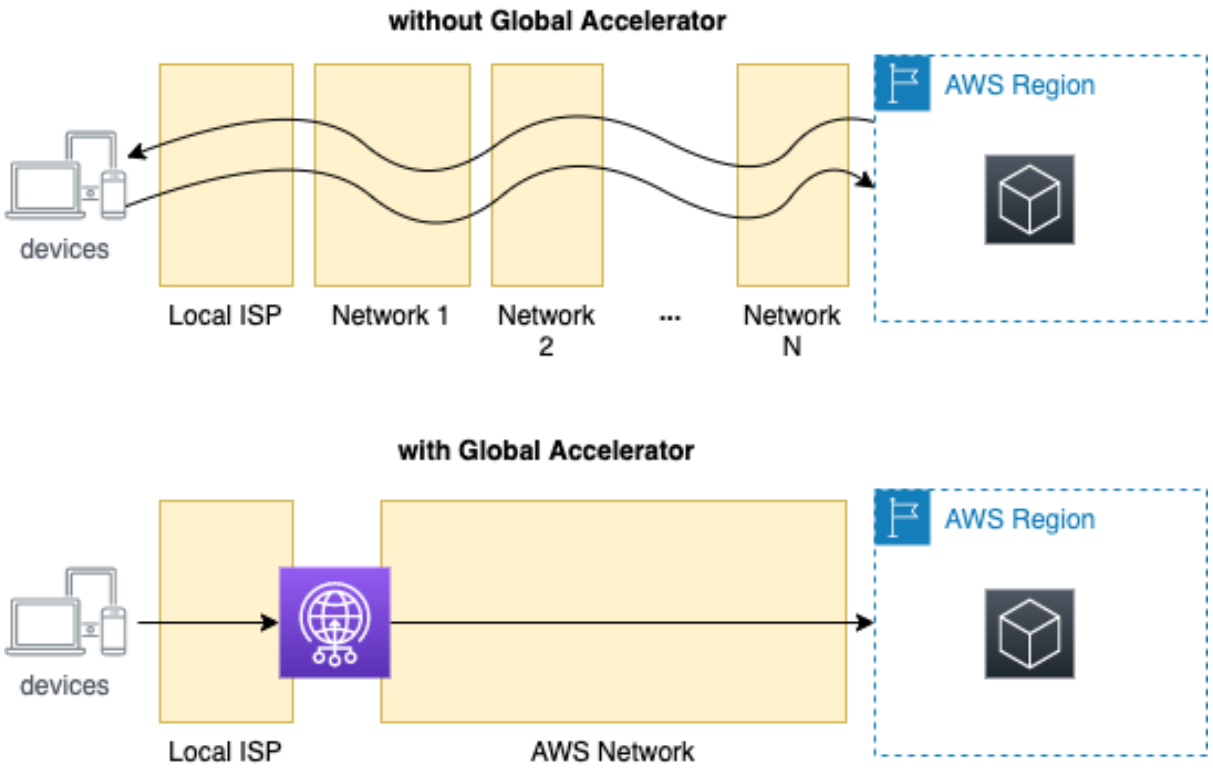


Figure 8.3: Side-by-side visualization: with and without AWS Global Accelerator

## Chapter 10: Designing for Cost Efficiency

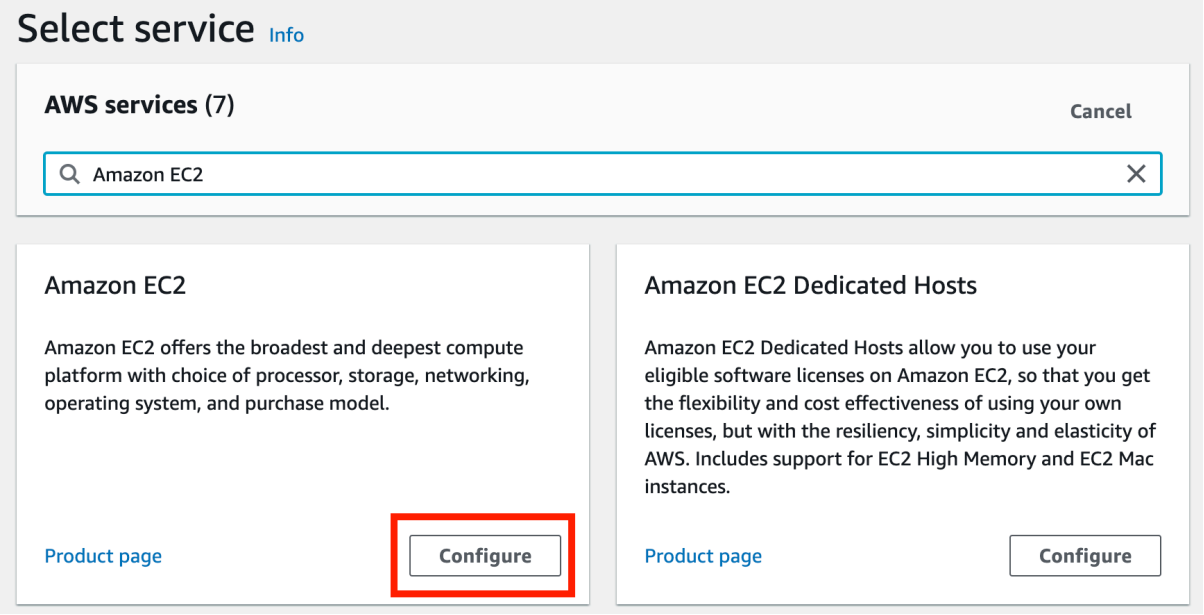


Figure 10.1: Service selection using AWS Pricing Calculator

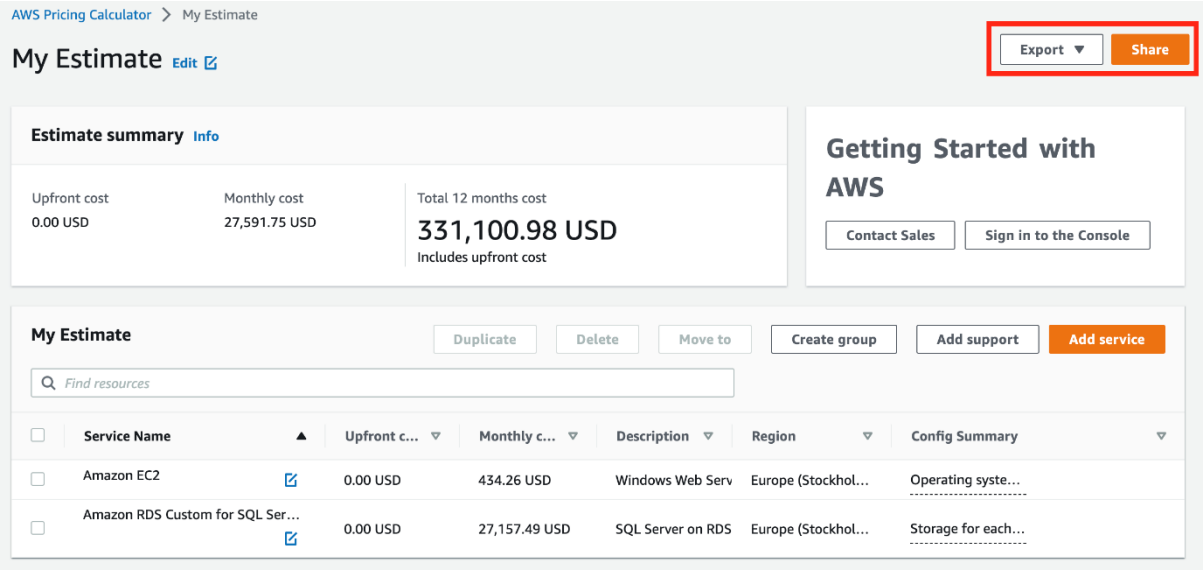


Figure 10.2: The estimates dashboard

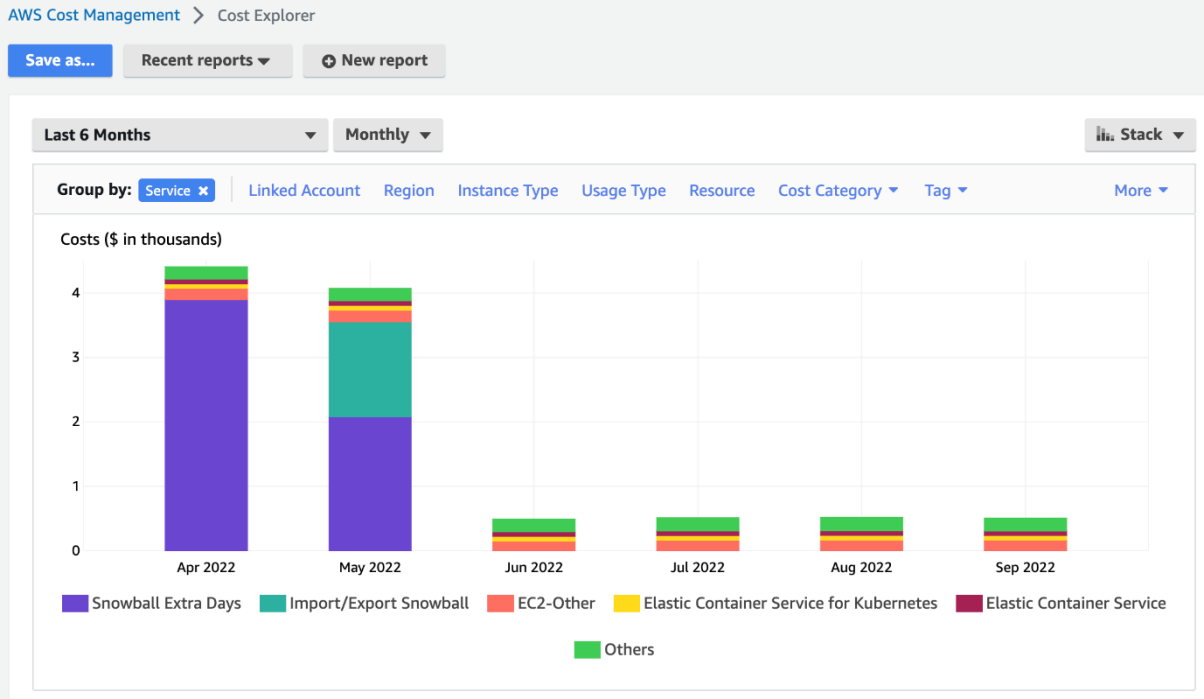


Figure 10.3: The AWS Cost Explorer dashboard

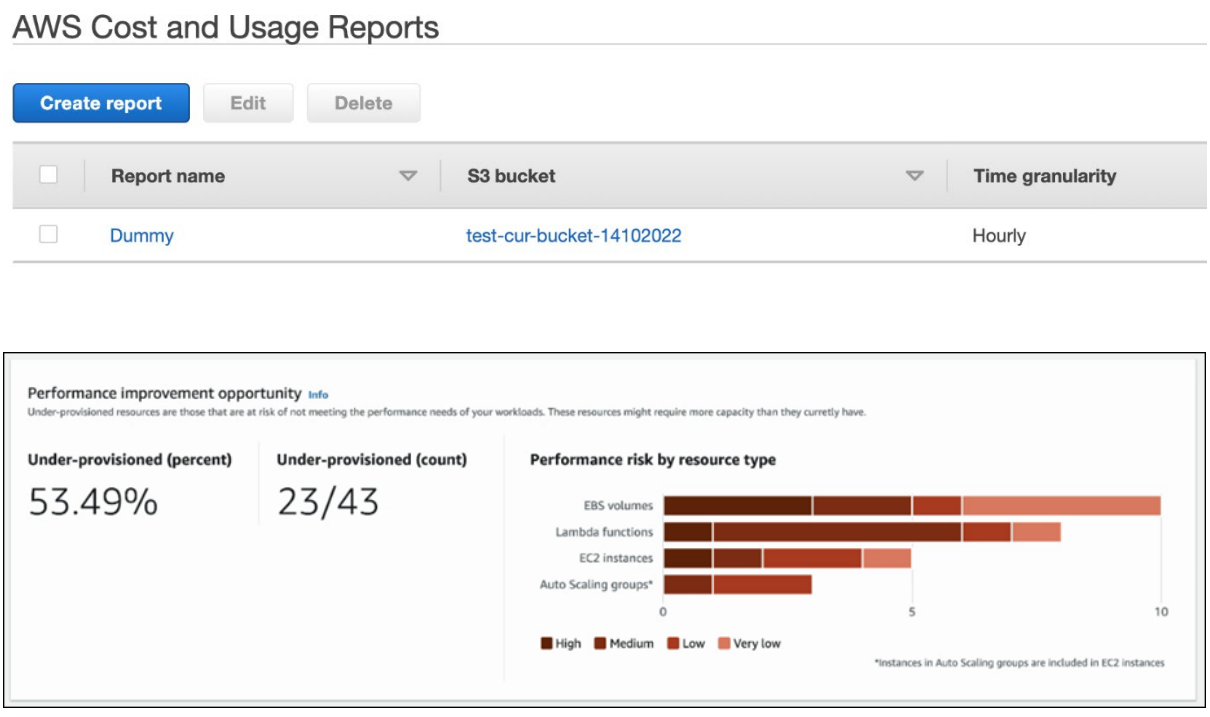





Figure 10.4: AWS CUR

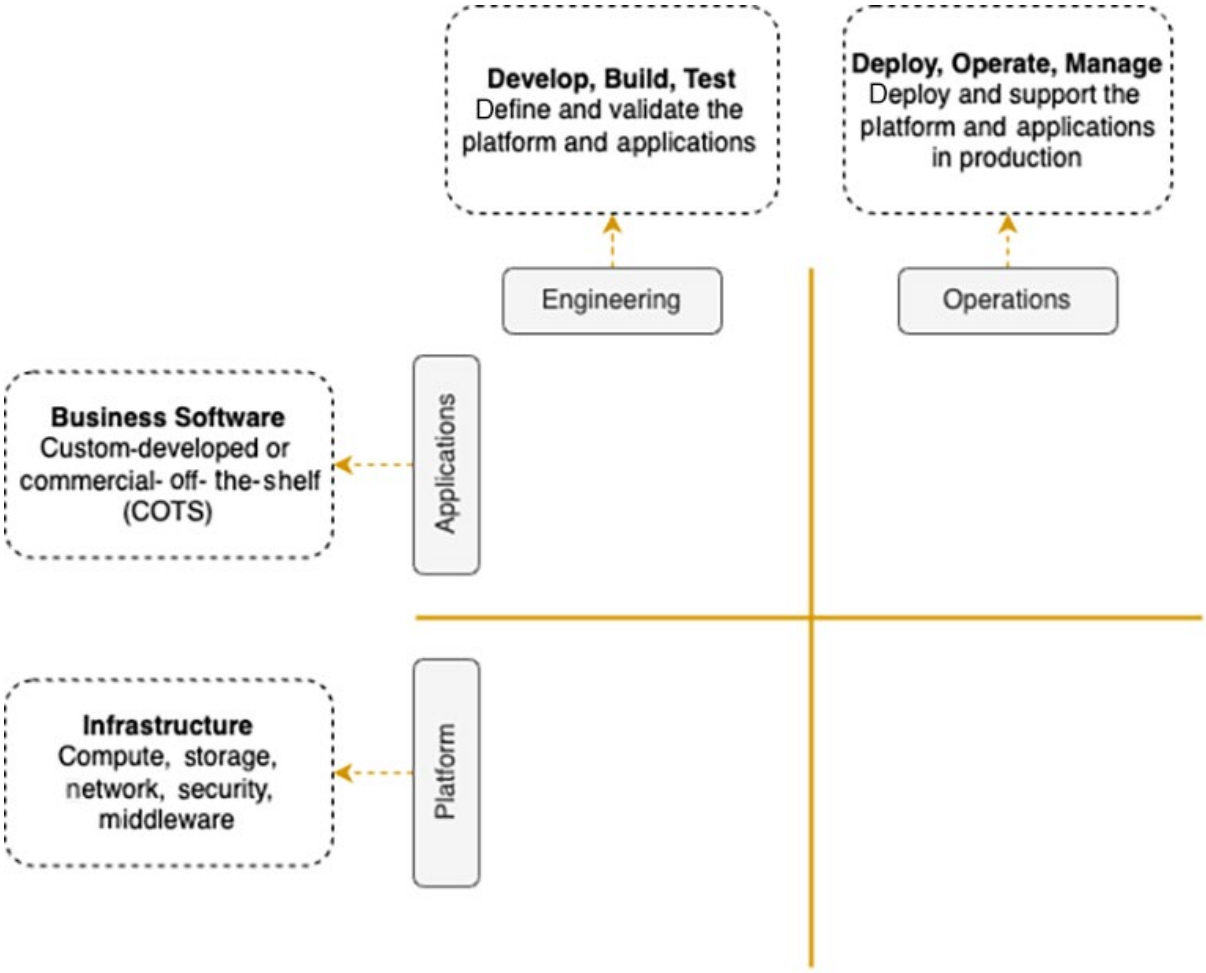


Figure 11.1: A 2x2 template representing an operating model graphical representation

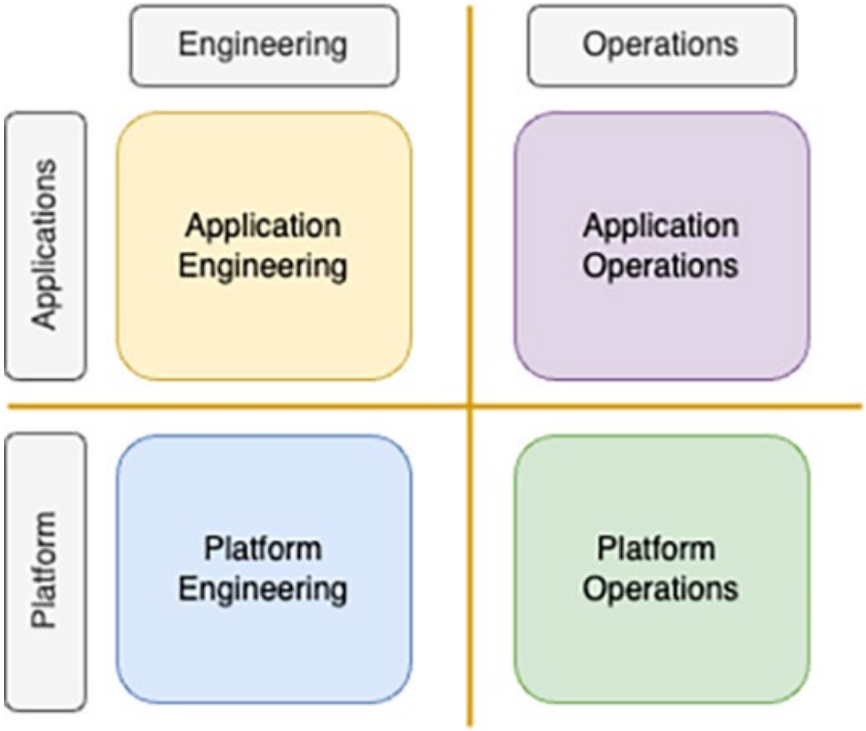


Figure 11.2: A fully separated operating model

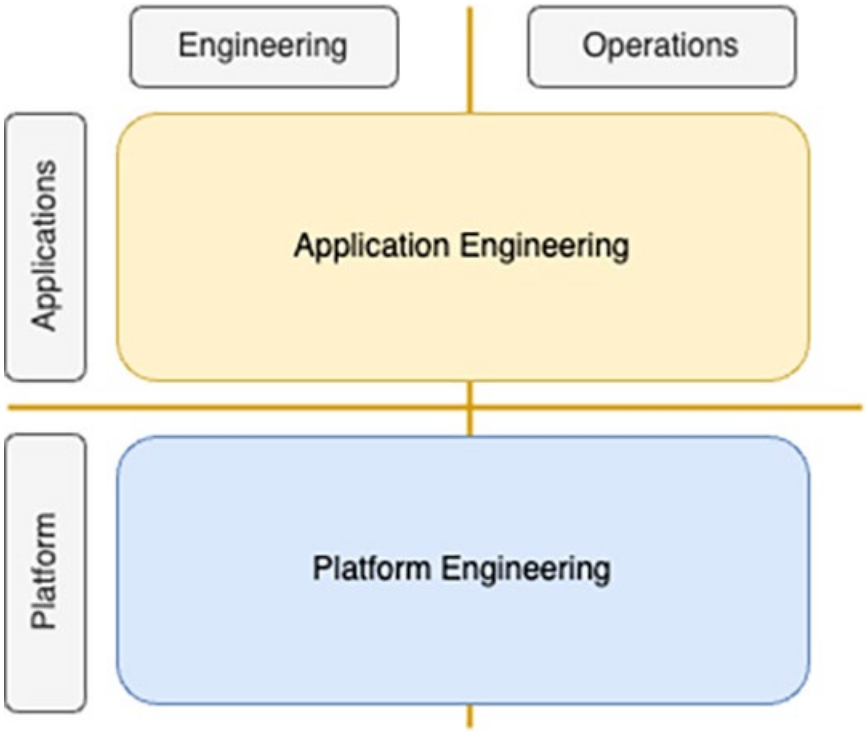


Figure 11.3: Separated AEO and PEO with centralized governance

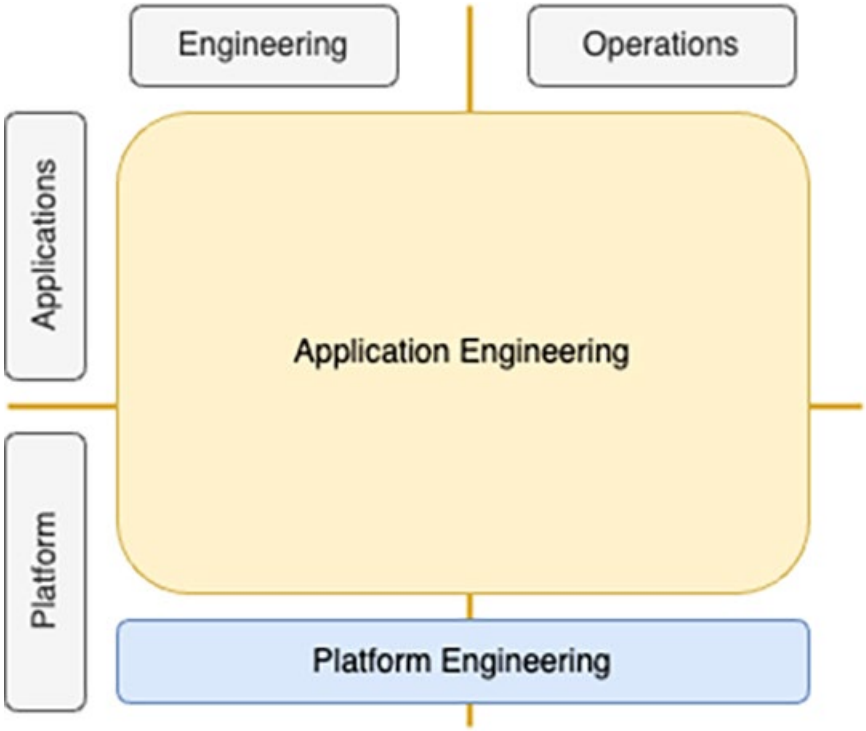


Figure 11.4: Separated AEO and PEO with decentralized governance

## Chapter 12: Improving Reliability

# Improvement plan

**Improvement item summary**

High risk: 10

Medium risk: 3

| Pillar | High risk | Medium risk |
|---|---|---|
| Reliability | 10 | 3 |
| Security | 0 | 0 |
| Operational Excellence | 0 | 0 |
| Performance Efficiency | 0 | 0 |
| Cost Optimization | 0 | 0 |
| Sustainability | 0 | 0 |

Figure 12.1: Example of a Well-Architected Review results overview

4. How do you design interactions in a distributed system to prevent failures?

High risk

**Selected choice(s)**

- Implement loosely coupled dependencies

**Not selected choice(s)**

- Identify which kind of distributed system is required
- Do constant work
- Make all responses idempotent
- None of these

**Best Practices marked as Not Applicable**

-

**Notes**

-

**Improvement plan**

- Identify which kind of distributed system is required
- Do constant work
- Make all responses idempotent

Ask an expert

Figure 12.2: Example of a high-risk issue

# Chapter 14: Improving Security

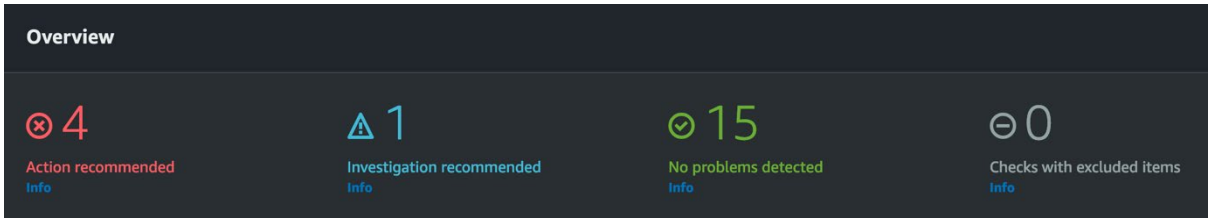


Figure 14.1: Summary of Trusted Advisor checks

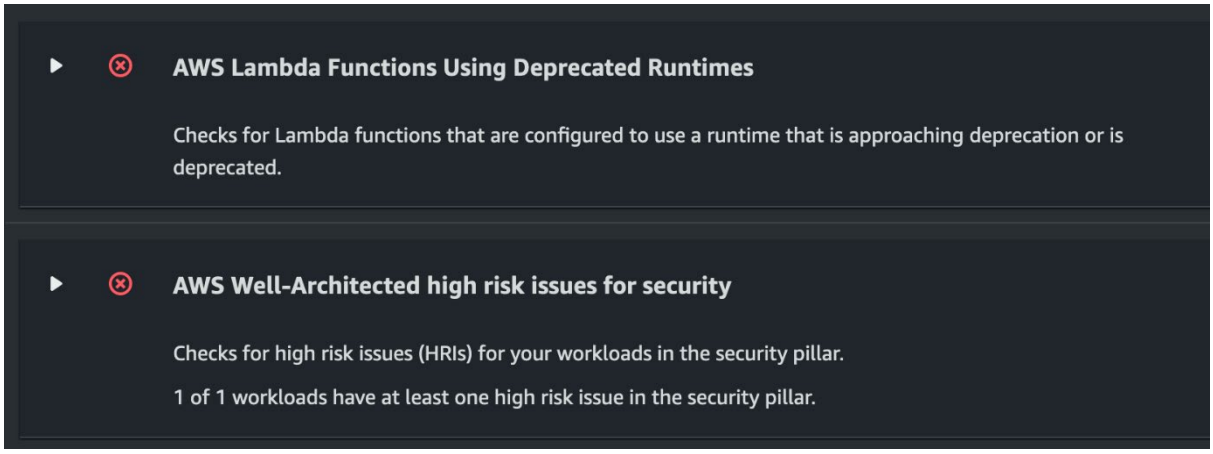


Figure 14.2: Sample list of checks from Trusted Advisor

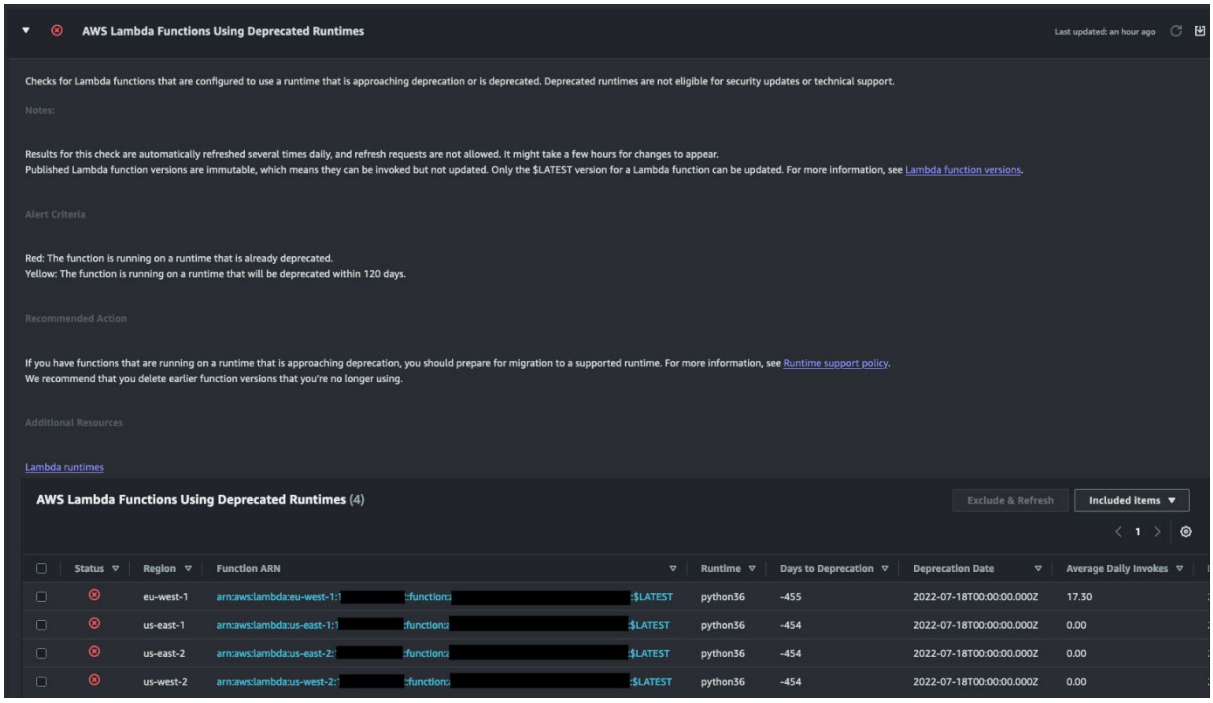


Figure 14.3: Details of a sample check from Trusted Advisor

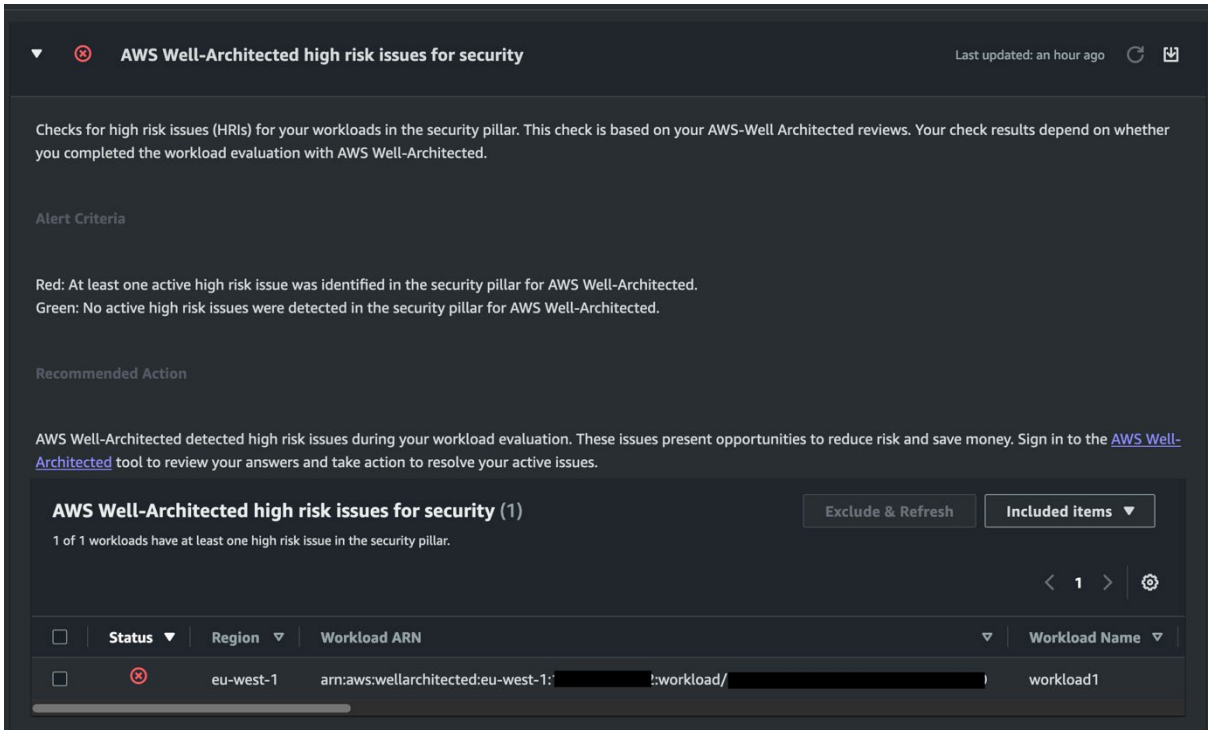


Figure 14.4: Details of a TrustedAdvisor check

## Chapter 17: Selecting Existing Workloads and Processes to Migrate


Figure 17.1: The Cloud Adoption Framework's perspectives

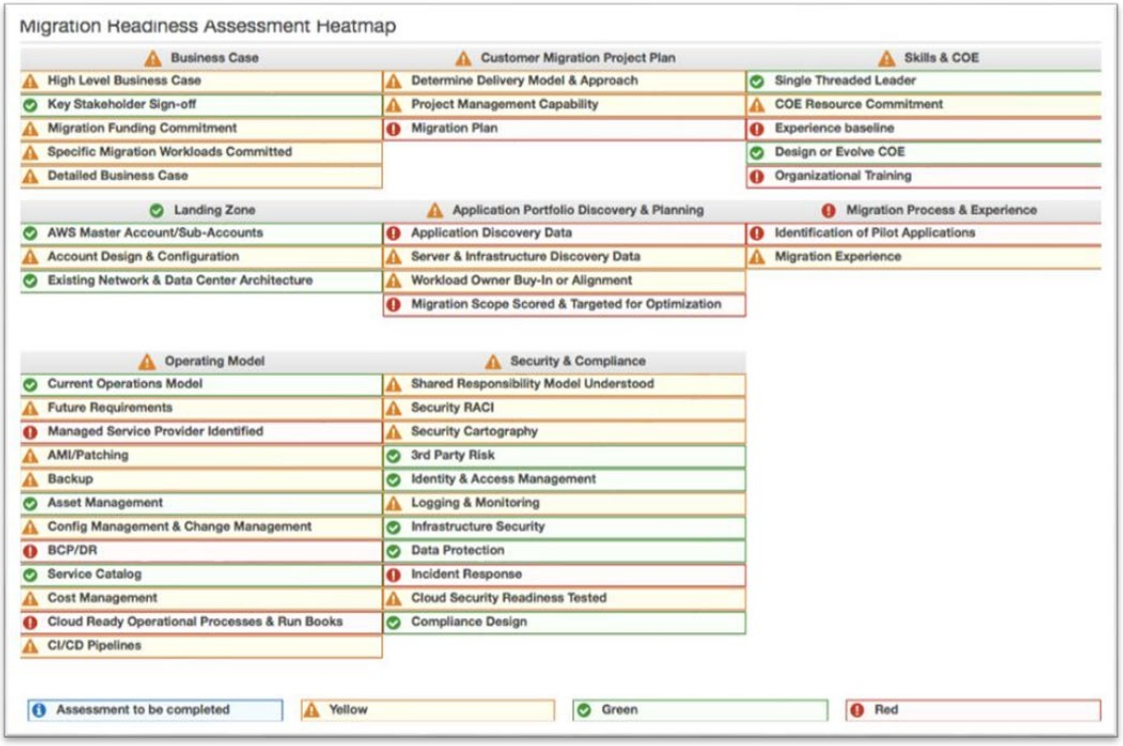


Figure 17.2: A sample migration readiness assessment output in the form of a heatmap

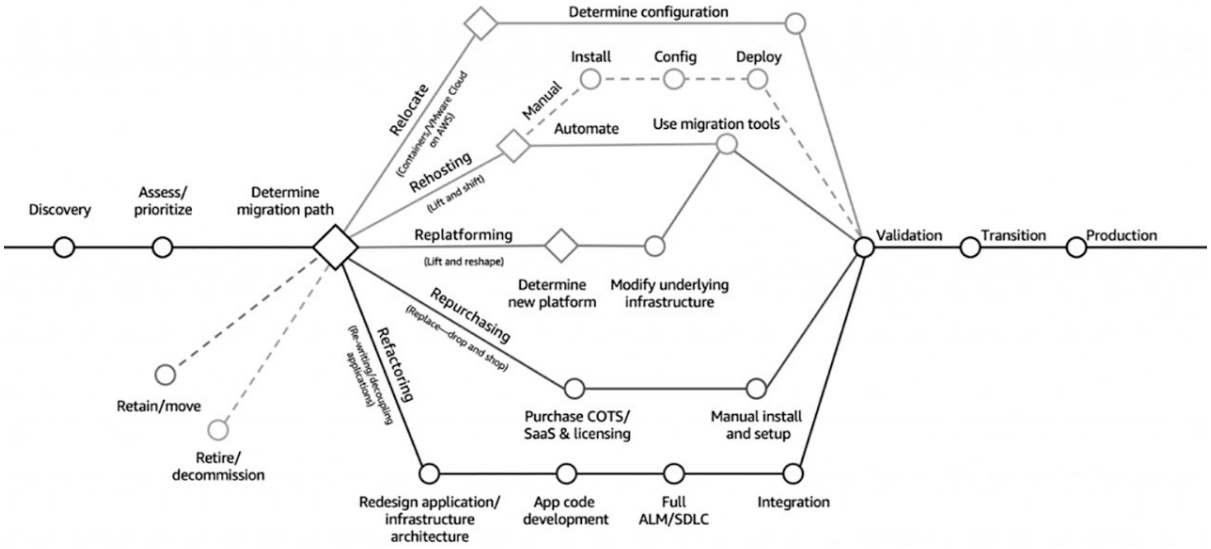


Figure 17.3: Application migration process


Figure 17.4: Cloud migration strategies and their complexities compared

# Chapter 18: Selecting Migration Tools and Services

| WQF Categories | Migration Strategies | Effort required | Oppotunity to optimize |
|---|---|---|---|
| 1 - ODBC/JDBC Workloads | Refactor | | |
| 2 - Light, proprietary features workloads | Refactor | | |
| 3 - heavy,proprietary feature workloads | Refactor / Replatform | | |
| 4 - Engine-specific workloads | Replatform / Rehost | | |
| 5 - COTS/non-portable workloads | Replatform/ Rehost | | |


Figure 18.1: Migration and optimization effort calibration

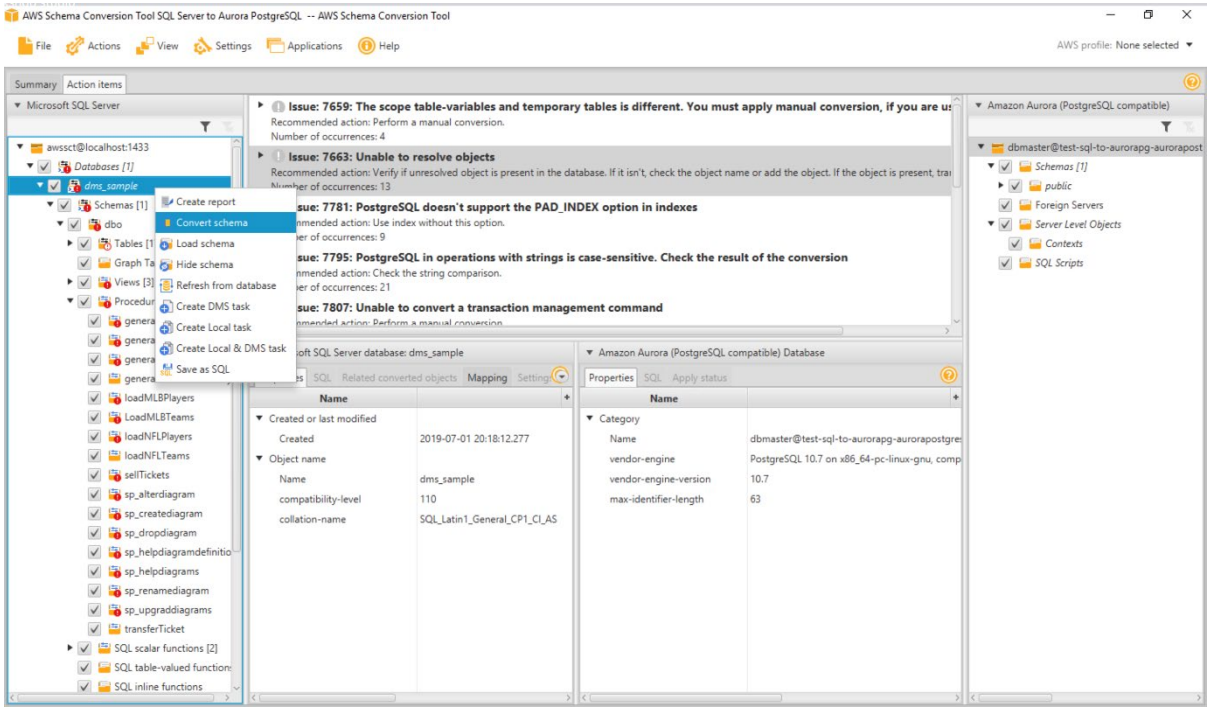


Figure 18.2: Schema conversion using SCT

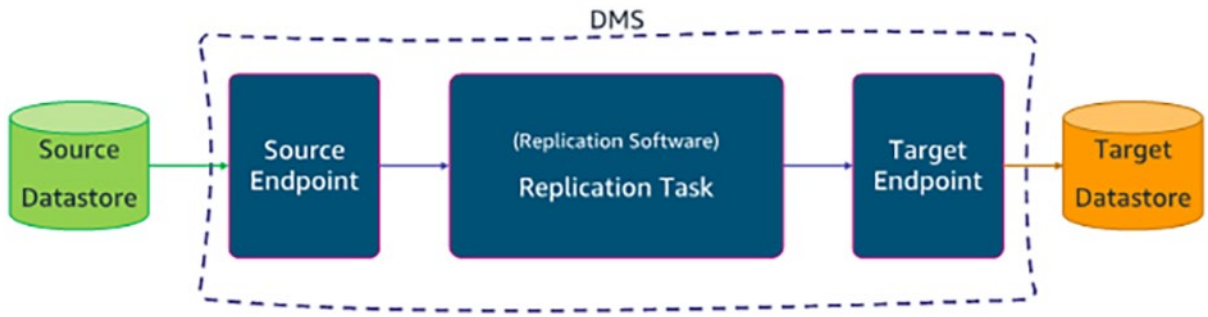


Figure 18.3: High-level overview of DMS

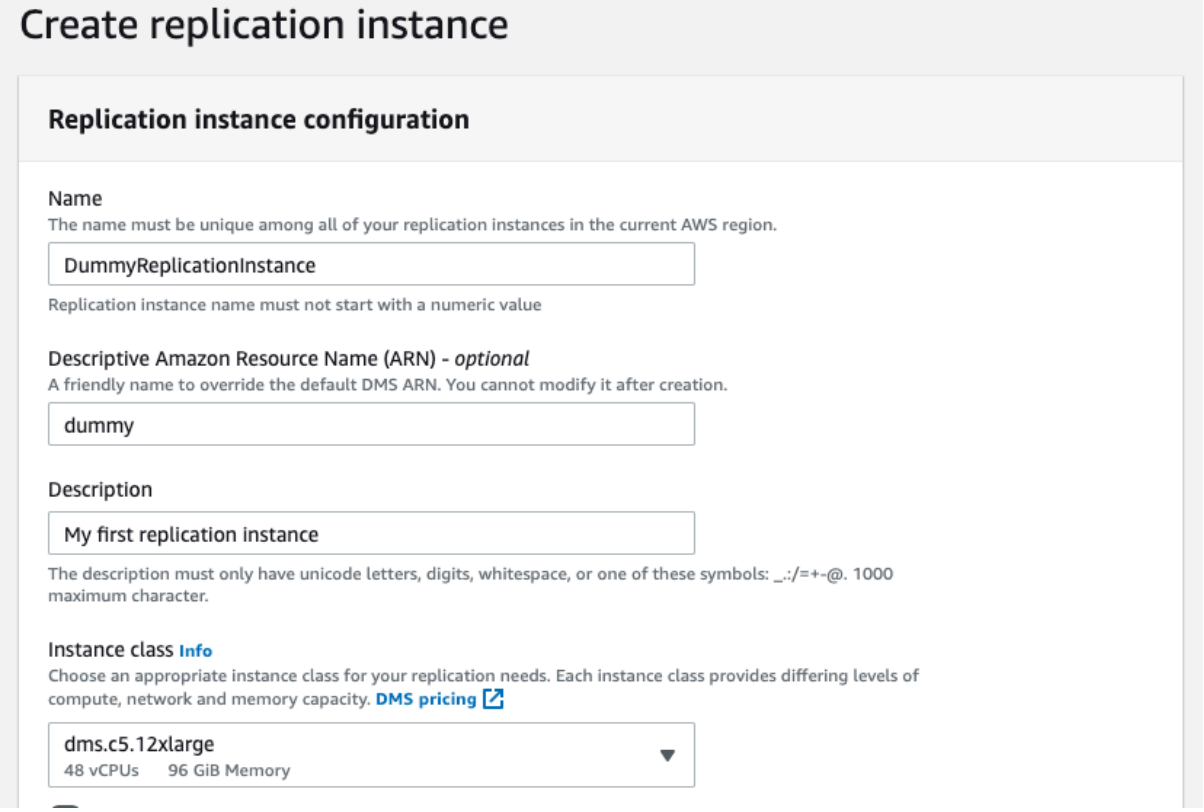


Figure 18.4: Creating a replication instance using the DMS management console

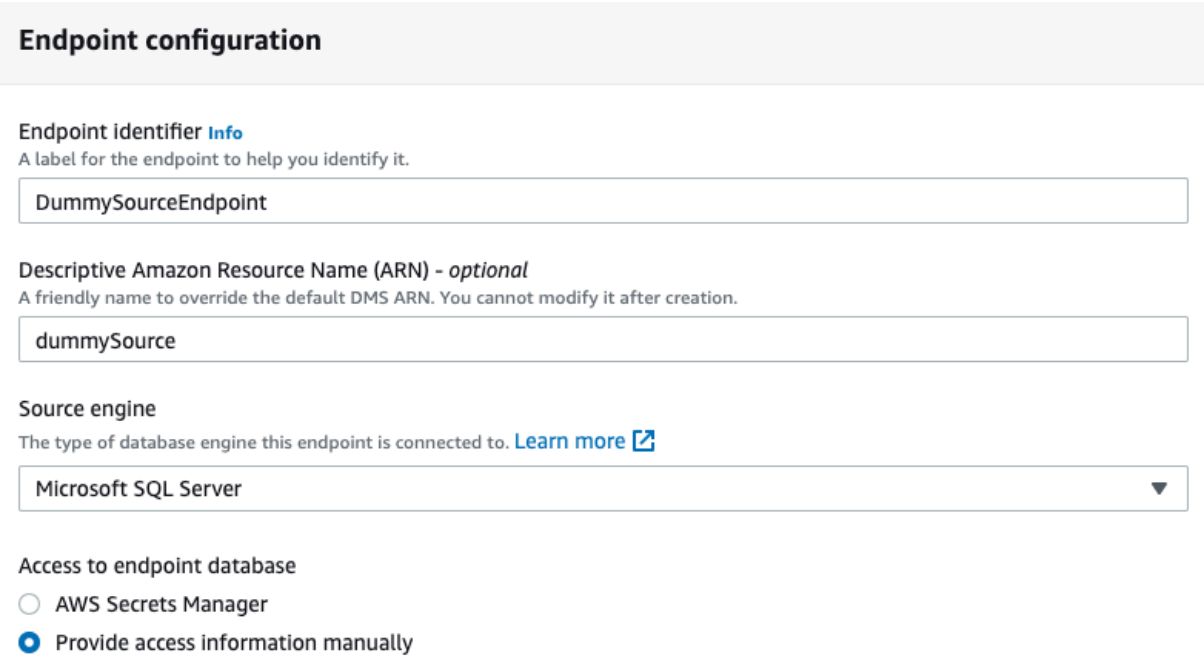


Figure 18.5: Endpoint creation

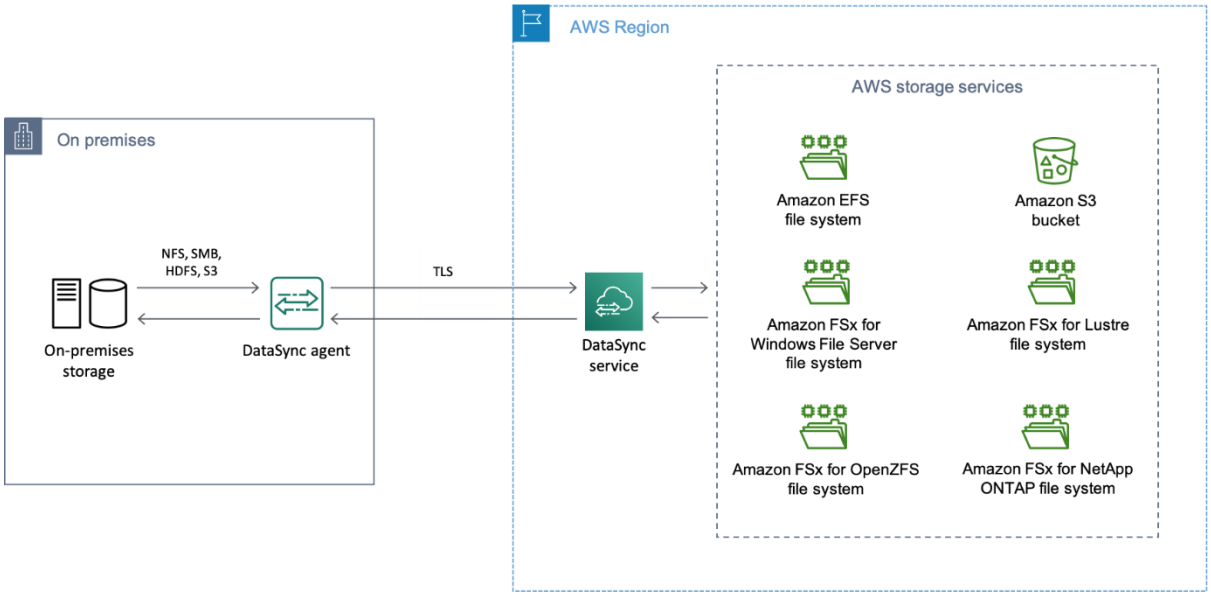


Figure 18.6: DataSync data migration from on-premises systems to AWS storage services

Figure 18.7: A Snowball Edge device

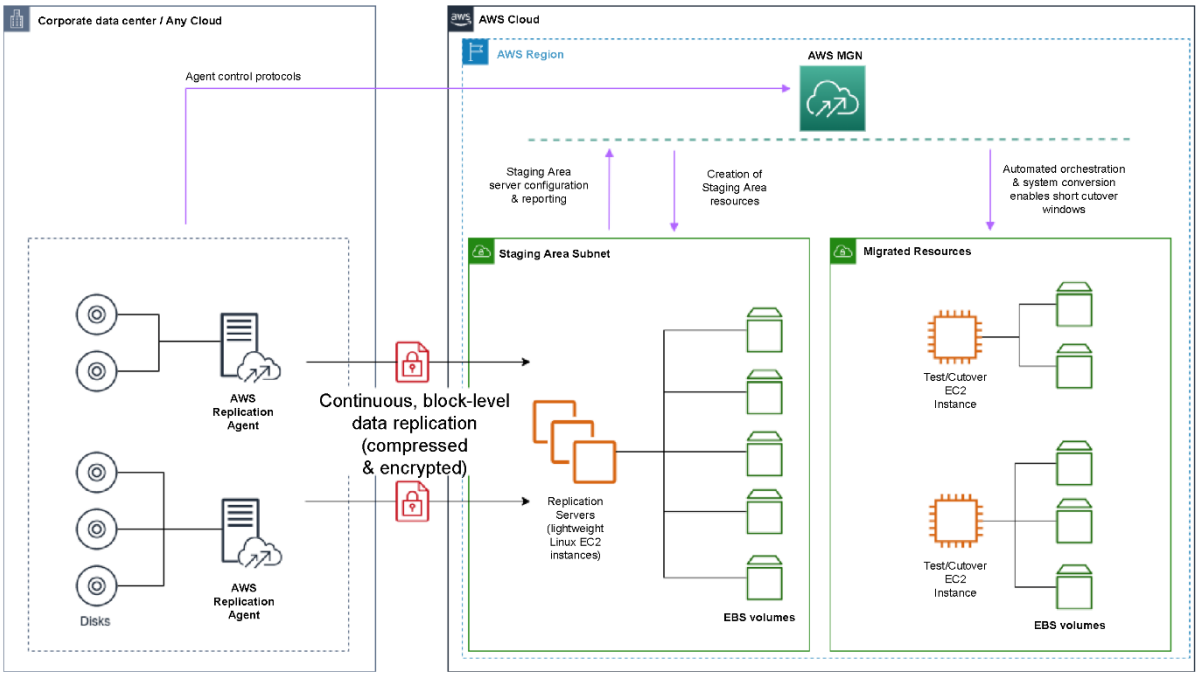


Figure 18.8: AWS Application Migration Service – under the hood

## Chapter 19: Determining a New Architecture for Existing Workloads

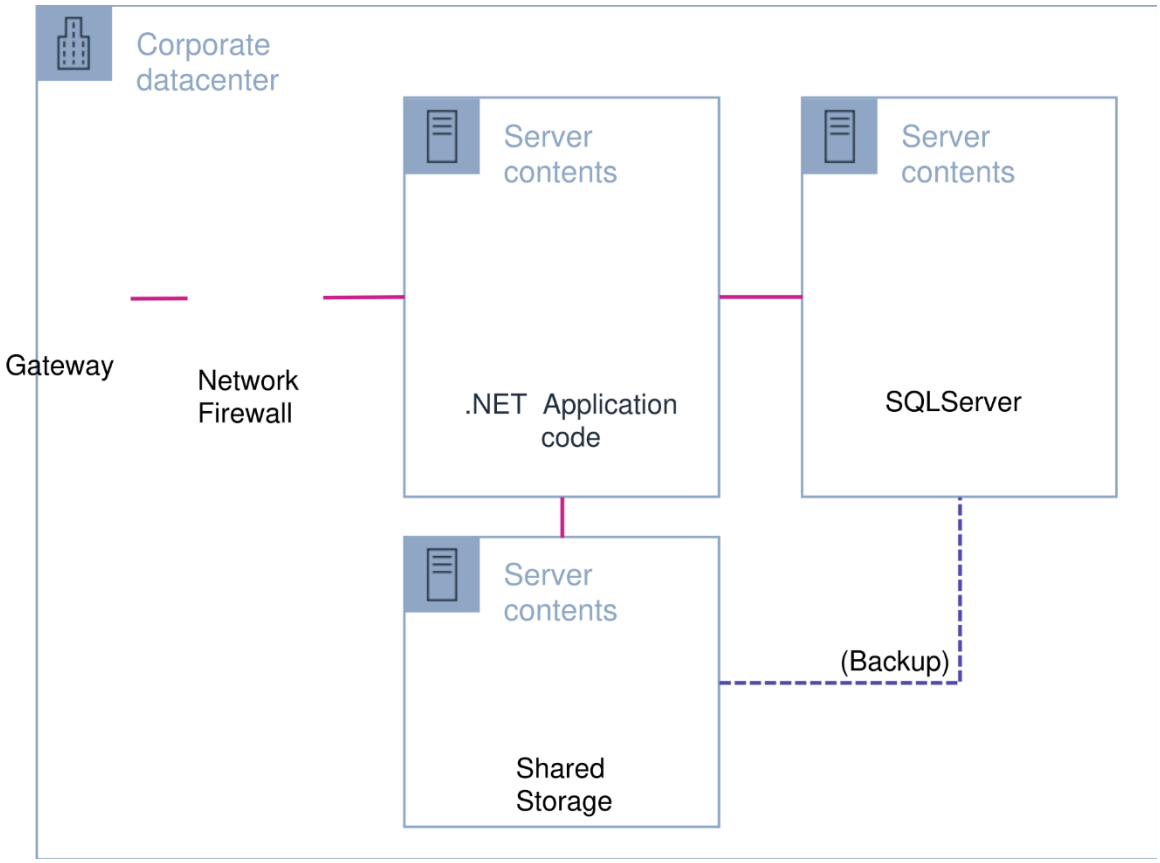


Figure 19.1: Typical legacy web application design hosted in a physical data center

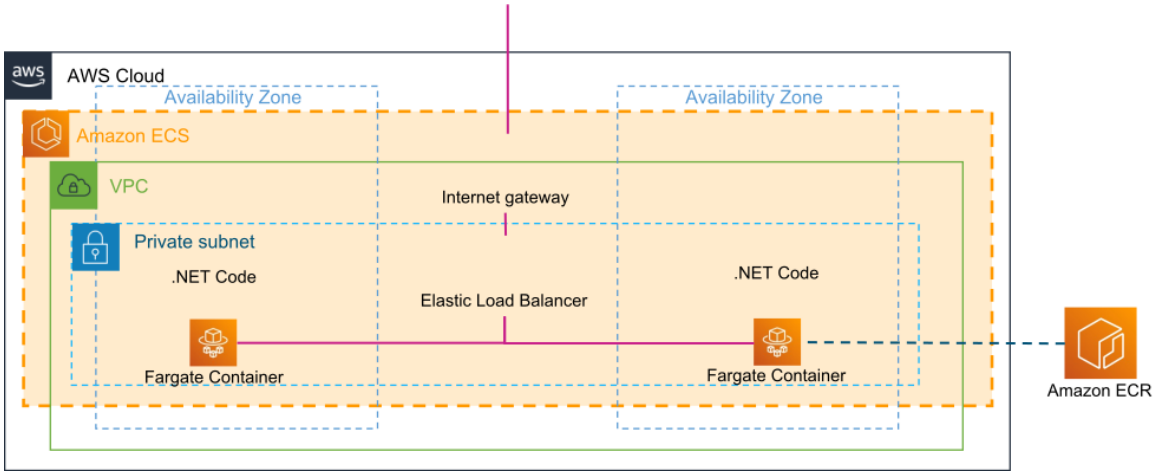


Figure 19.2: Containerized application hosting on AWS Fargate

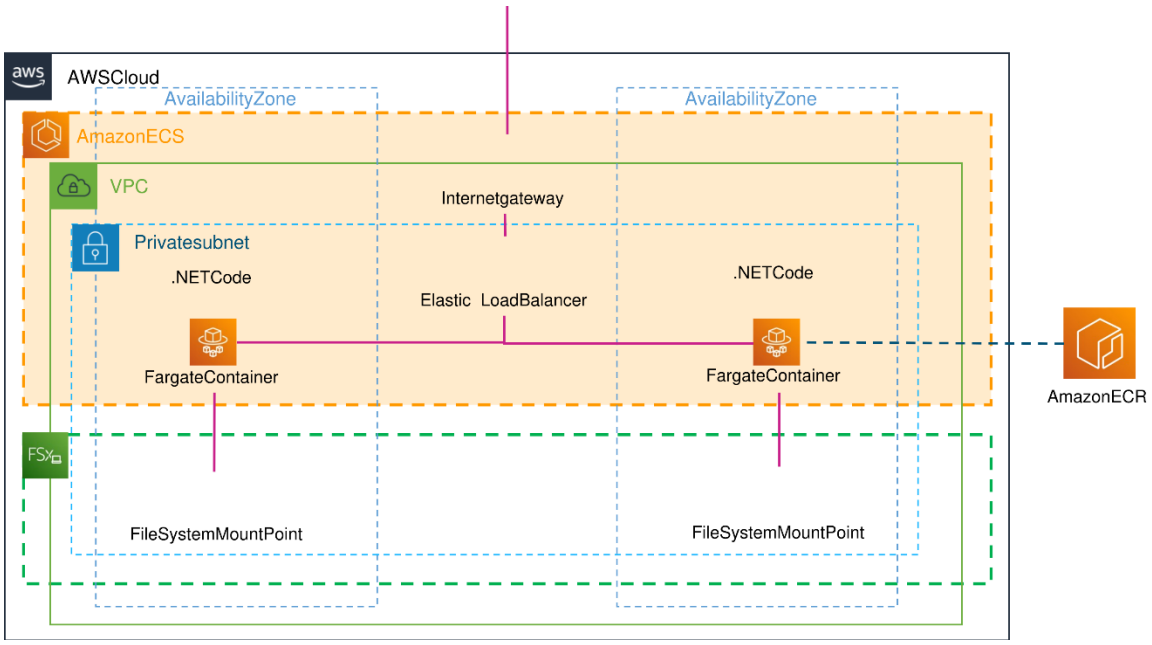


Figure 19.3: Containerized application with backend storage on Amazon FSx for Windows File Server

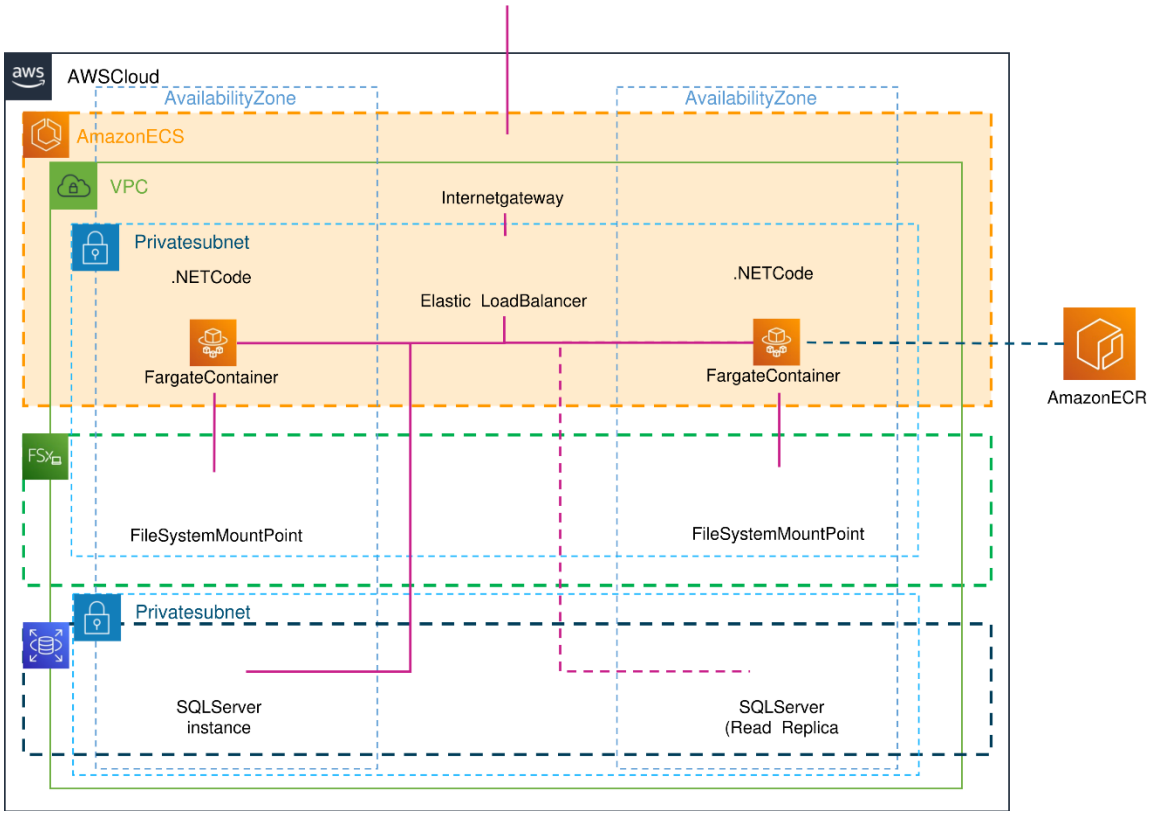


Figure 19.4: Managed SQL Server database using Amazon RDS

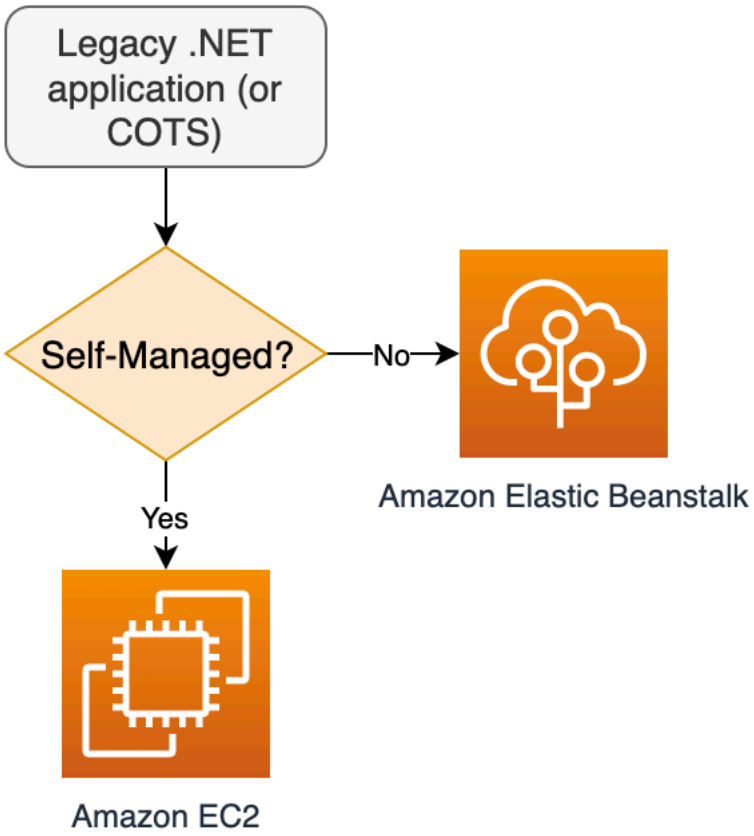


Figure 20.1: Rehosting pattern

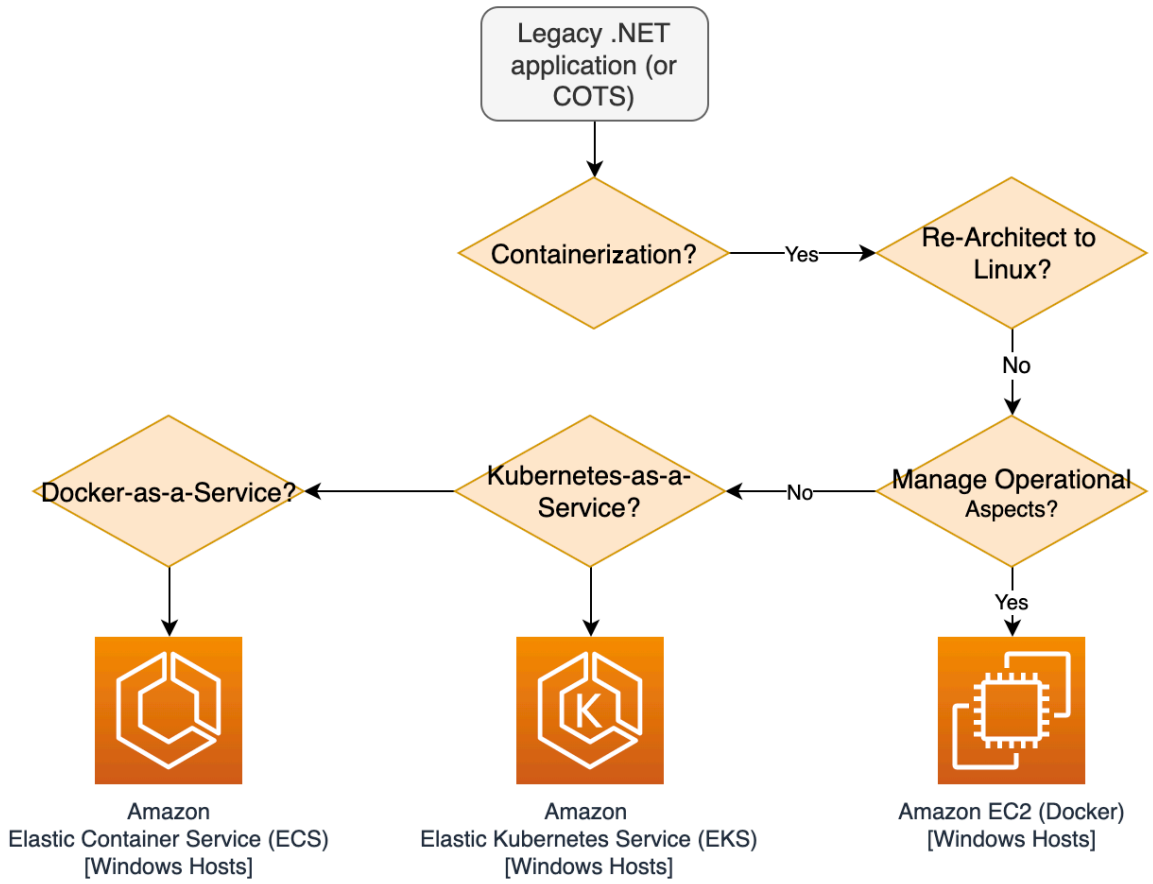


Figure 20.2: Replatform pattern

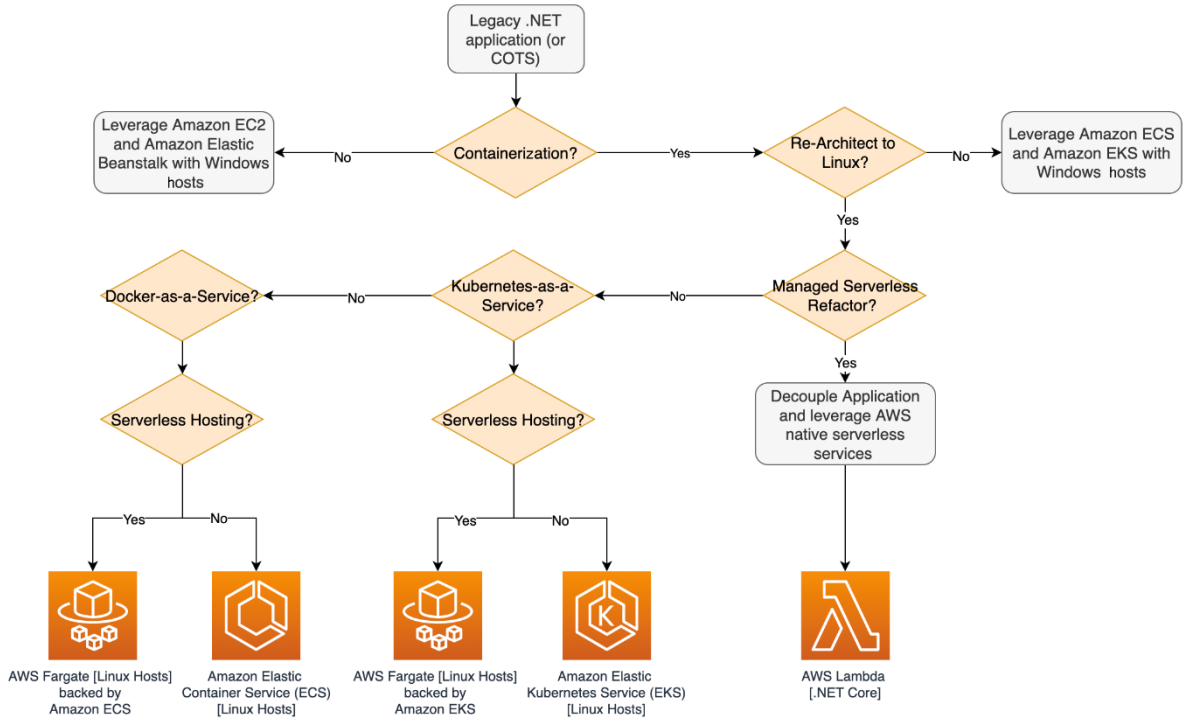


Figure 20.3: Refactoring pattern

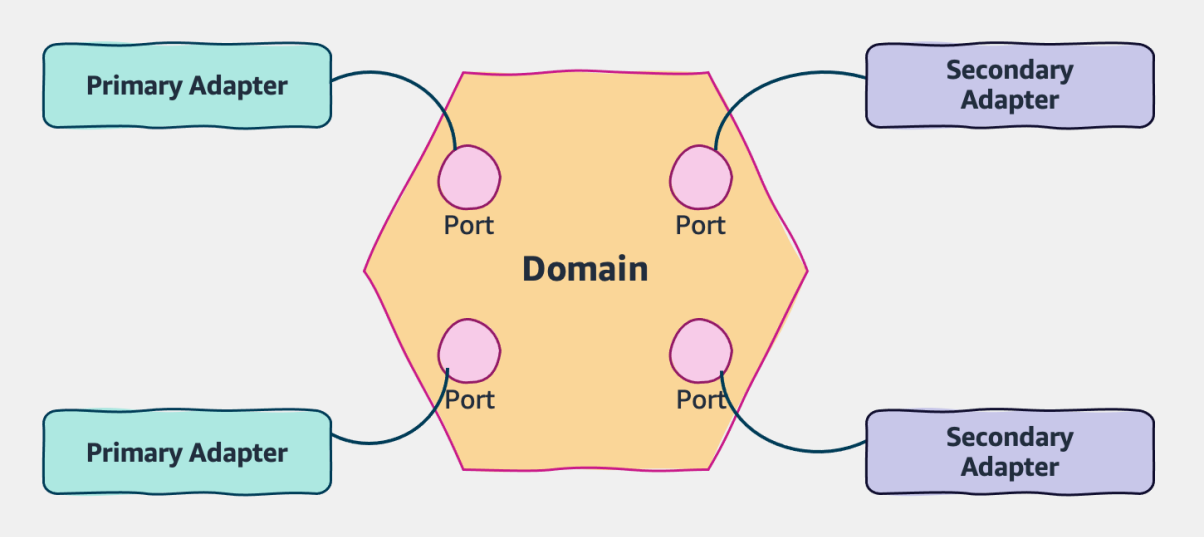

Figure 20.4: Anatomy of hexagonal architecture

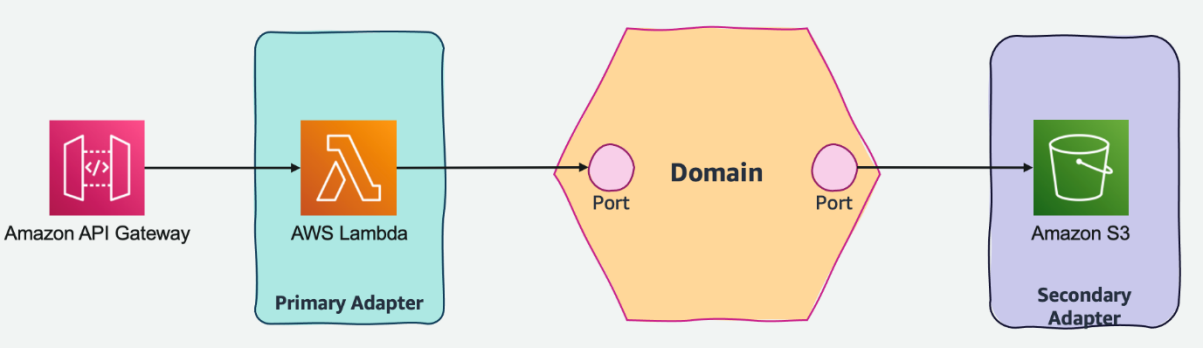

Figure 20.5: Starting out with a hexagonal architecture implementation for our sample application

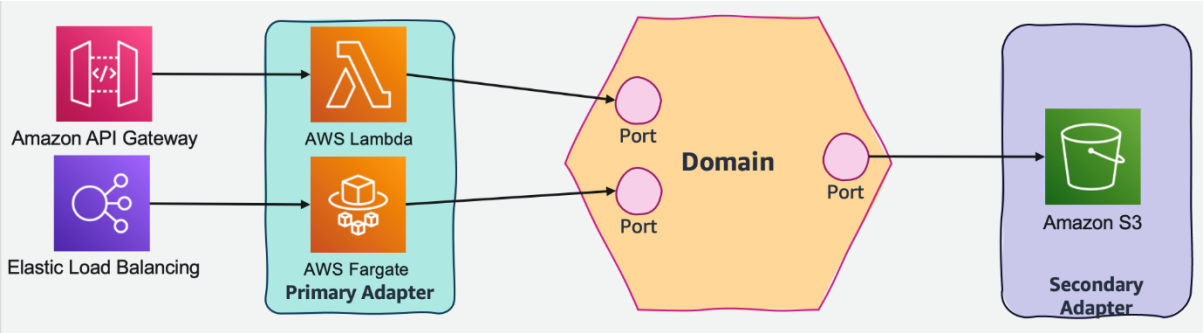

Figure 20.6: Adding more clients and adapters

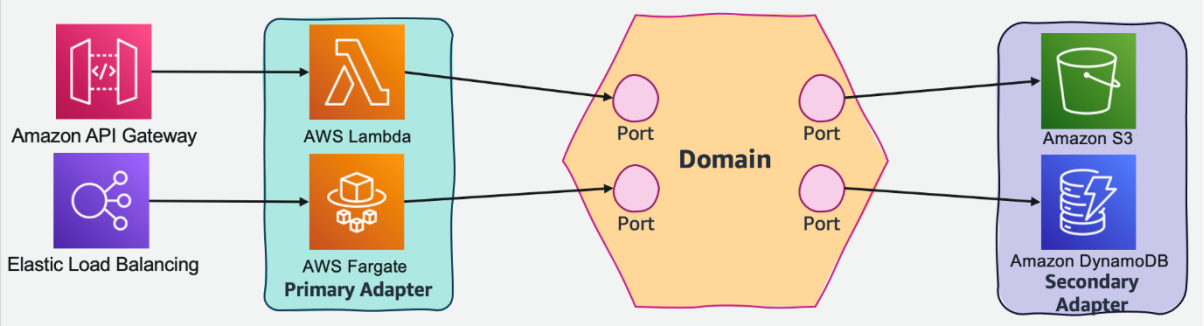


Figure 20.7: Adding new adapters and ports to extend the application's functionality

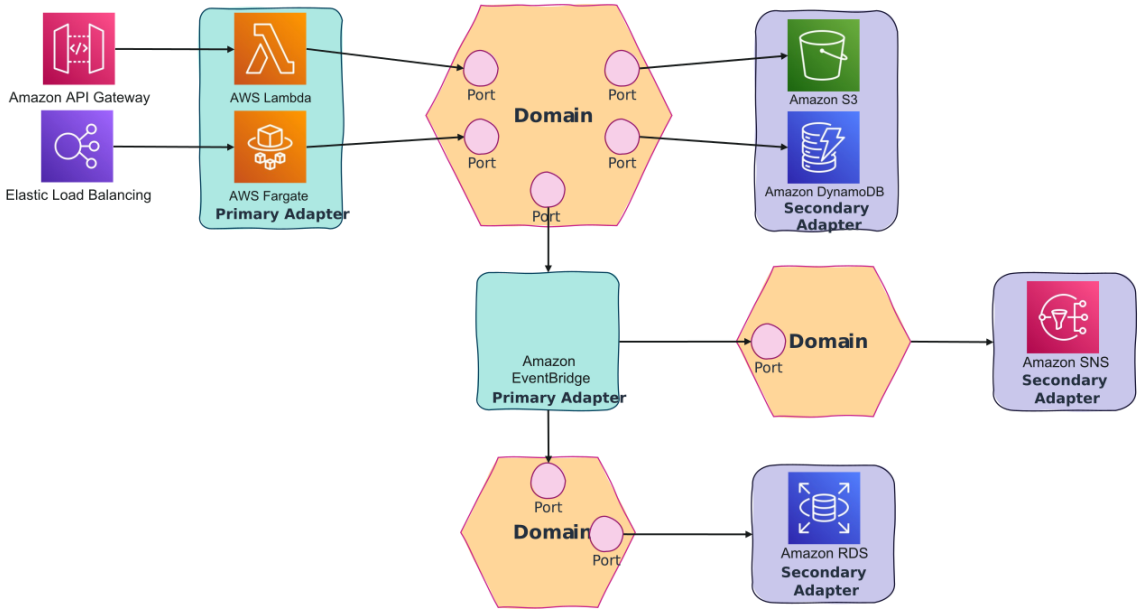


Figure 20.8: Expanding domains as per application requirements

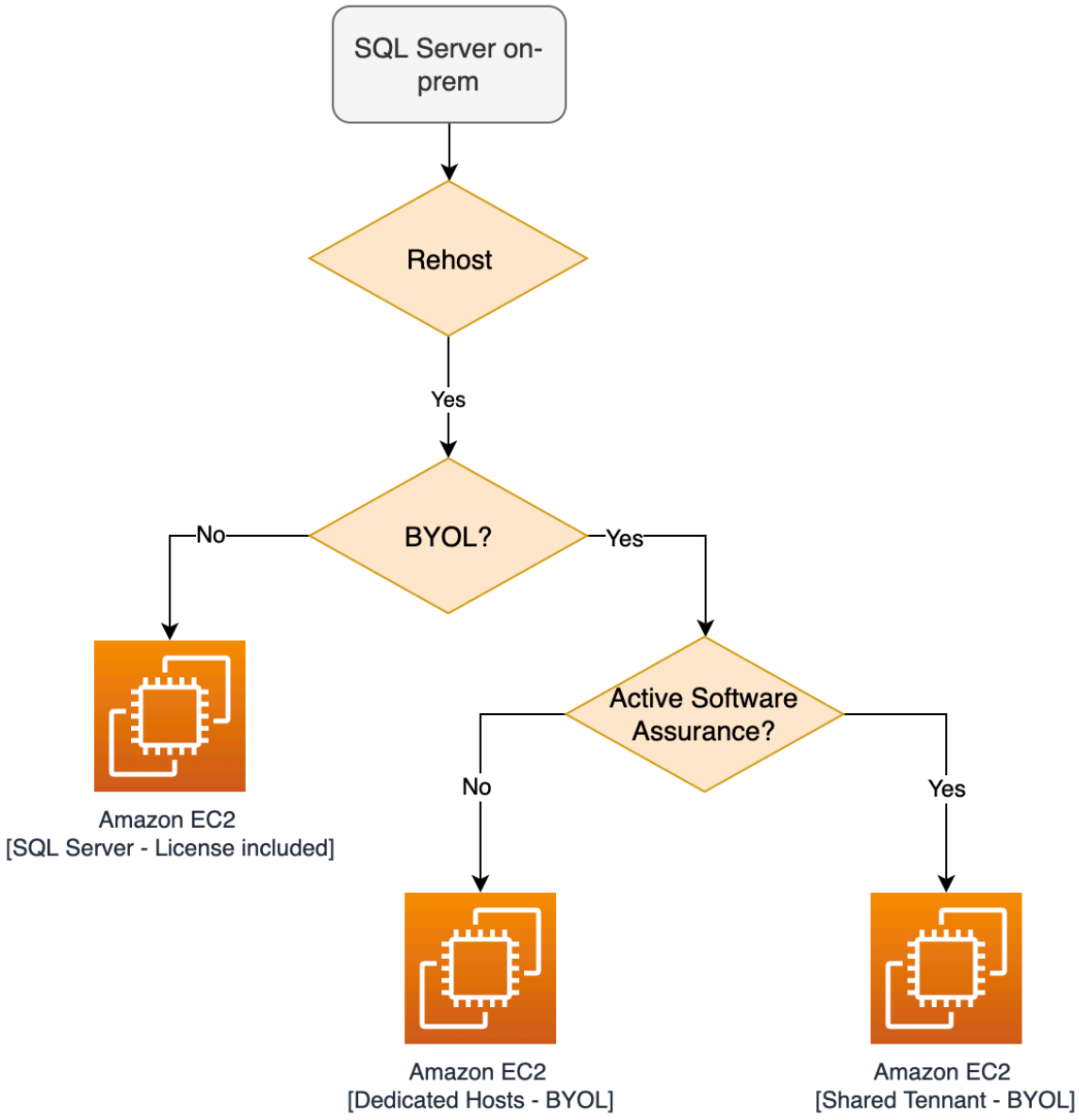


Figure 20.9: Rehosting opportunity for SQL Server

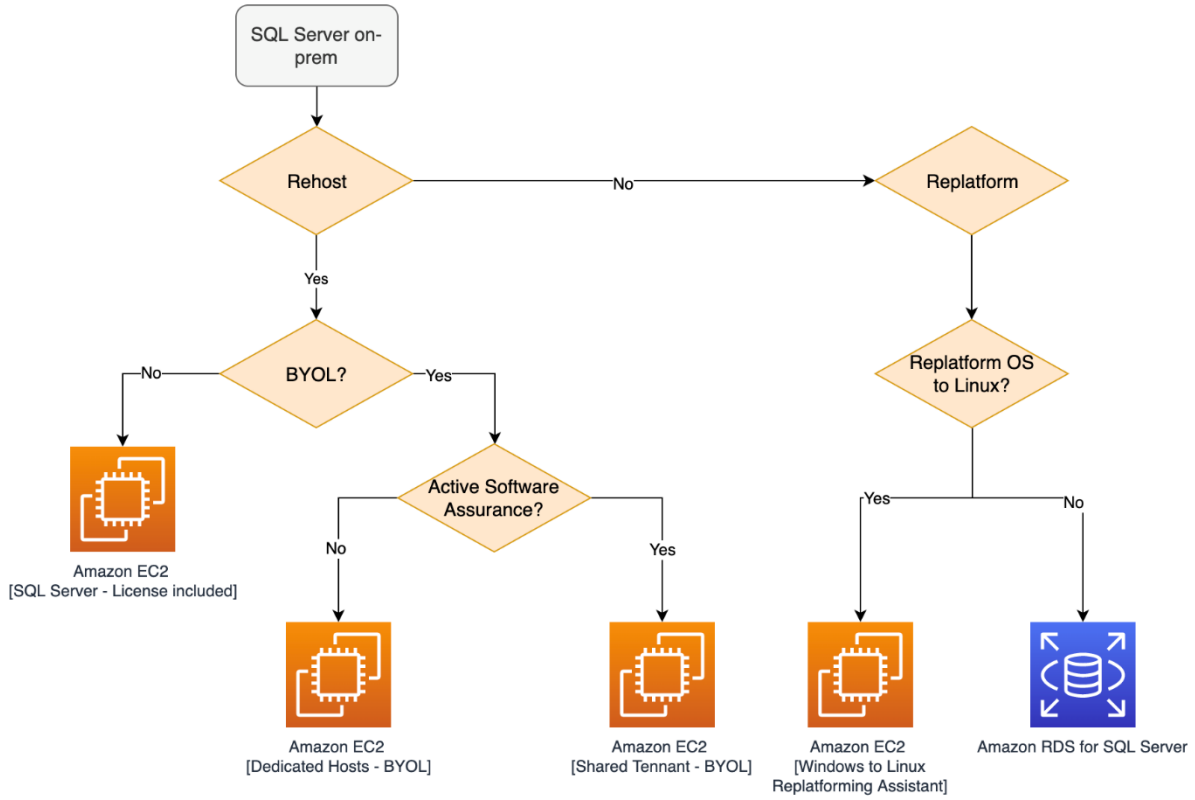


Figure 20.10: Replatforming opportunity for SQL Server

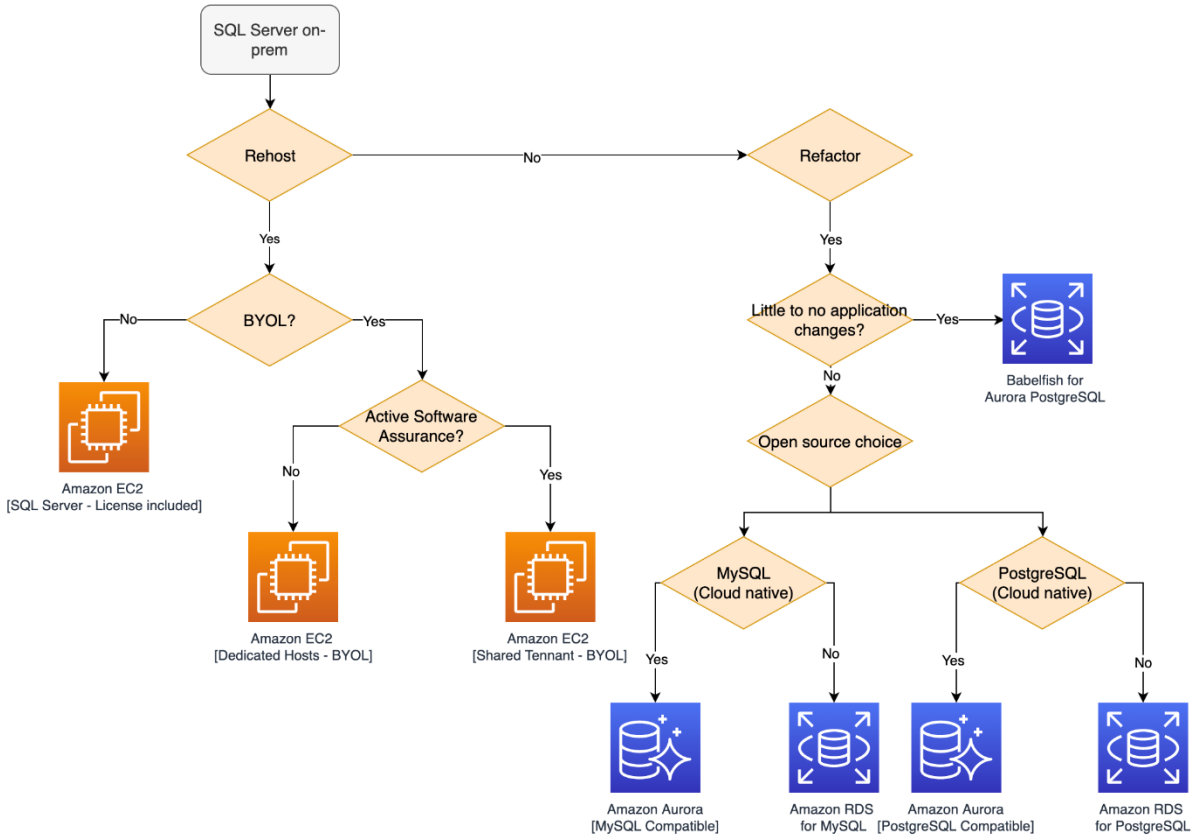


Figure 20.11: Refactoring opportunities for SQL Server on AWS

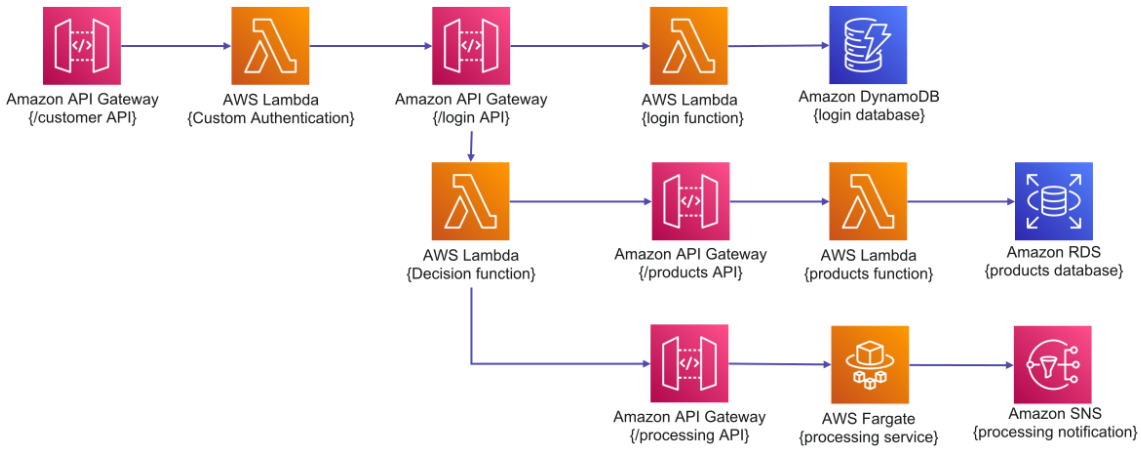


Figure 20.12: API Gateway pattern with single and multiple API gateways implemented within the same architecture

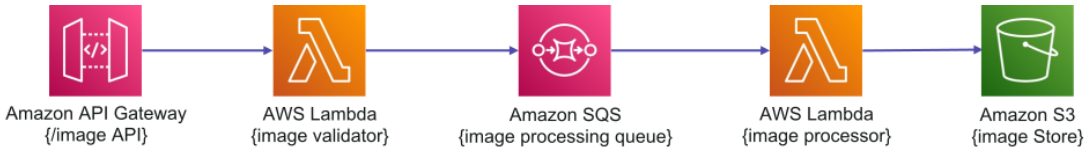


Figure 20.13: Messaging pattern implementation using Amazon SQS

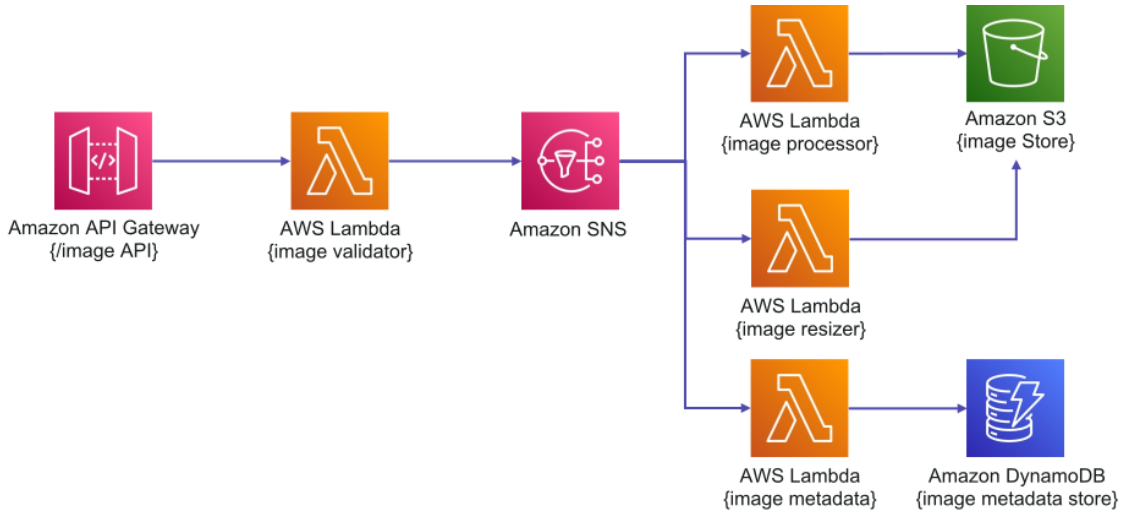


Figure 20.14: Pub/Sub pattern implementation using Amazon SNS

# Chapter 21: Accessing the Online Practice Resources

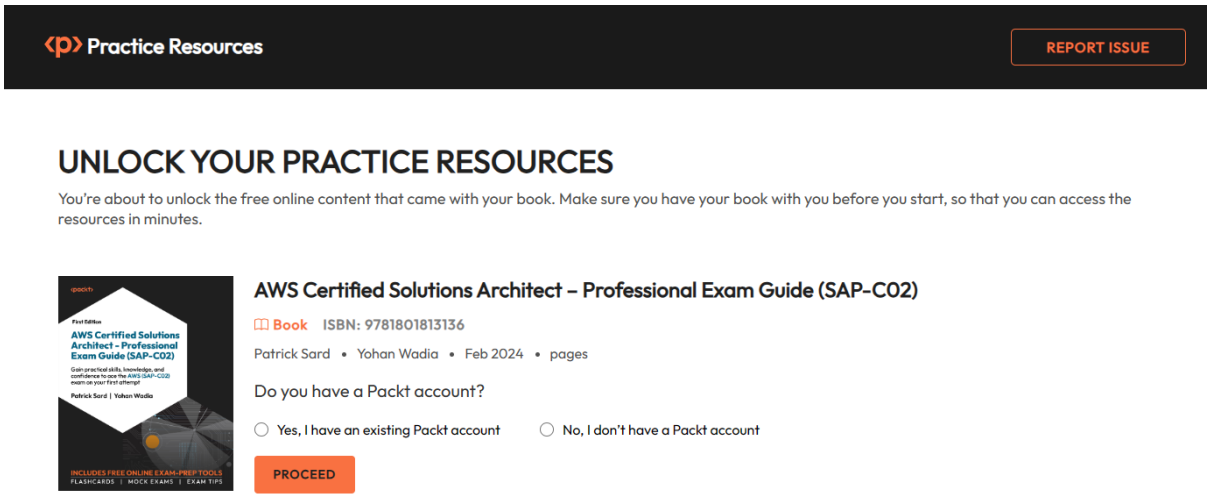


Figure 21.2 – Unlock page for the online practice resources

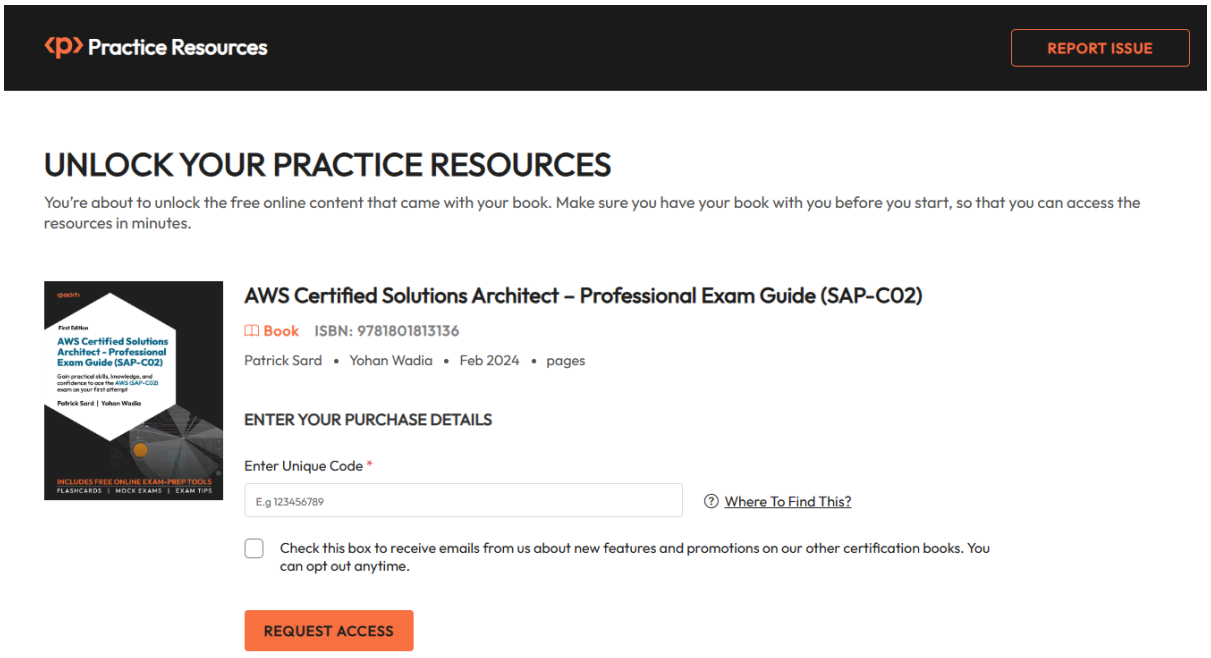


Figure 21.3 – Enter your unique sign-up code to unlock the resources

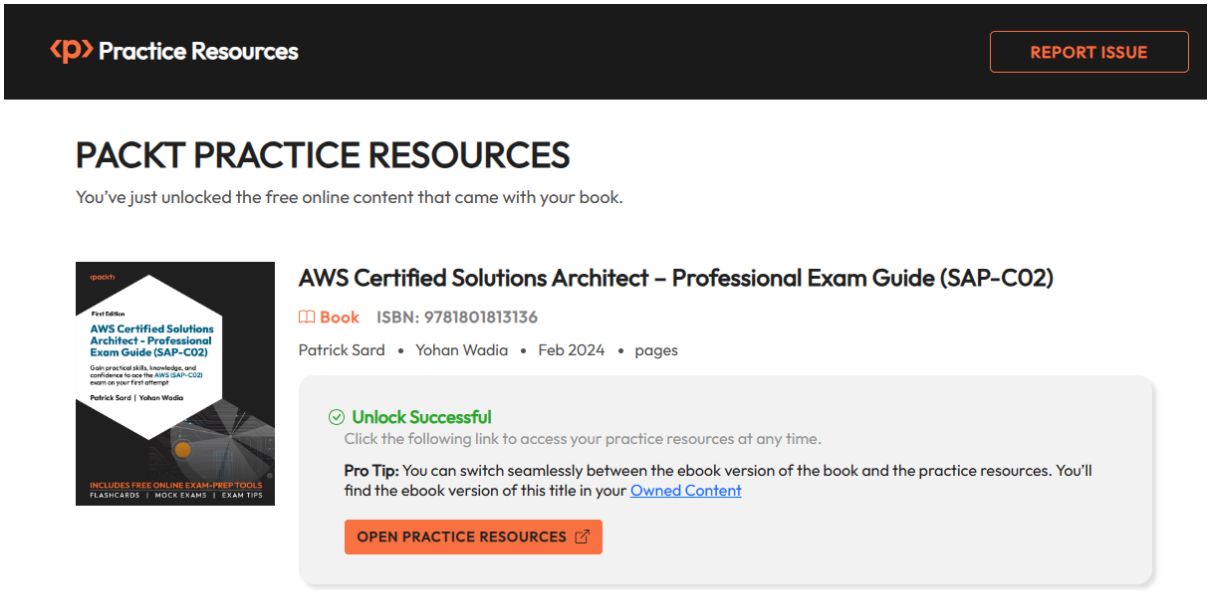


Figure 21.4 – Page that shows up after a successful unlock

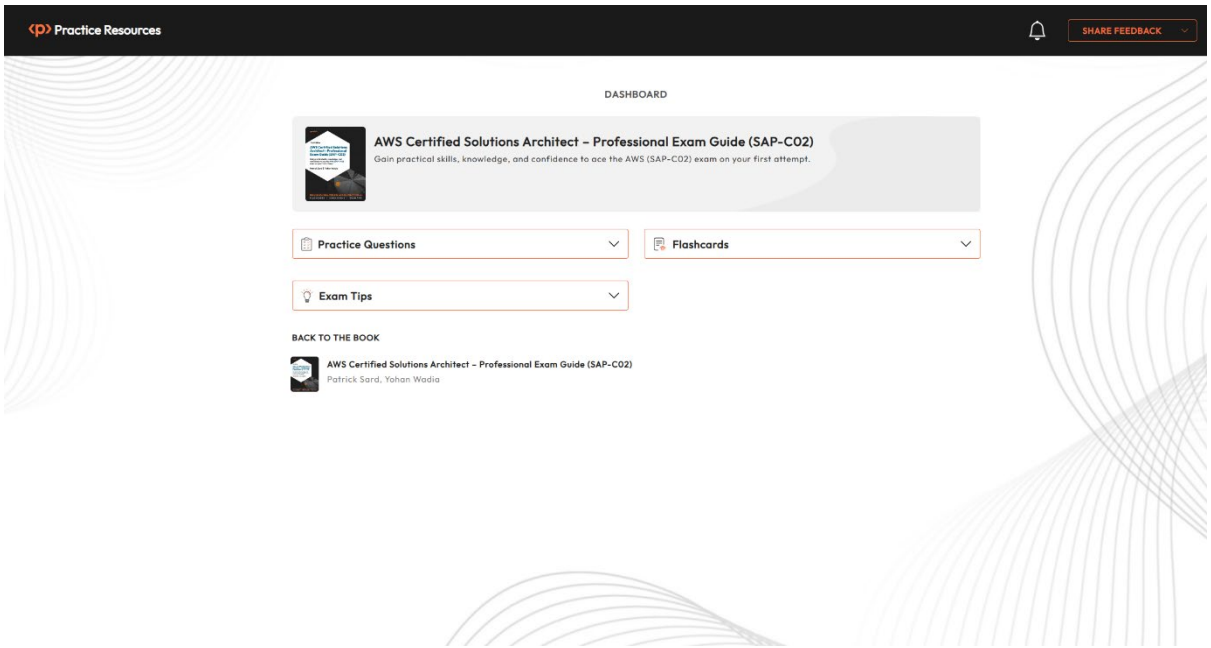


Figure 21.5 – Dashboard page for AWS (SAP-C02) practice resources

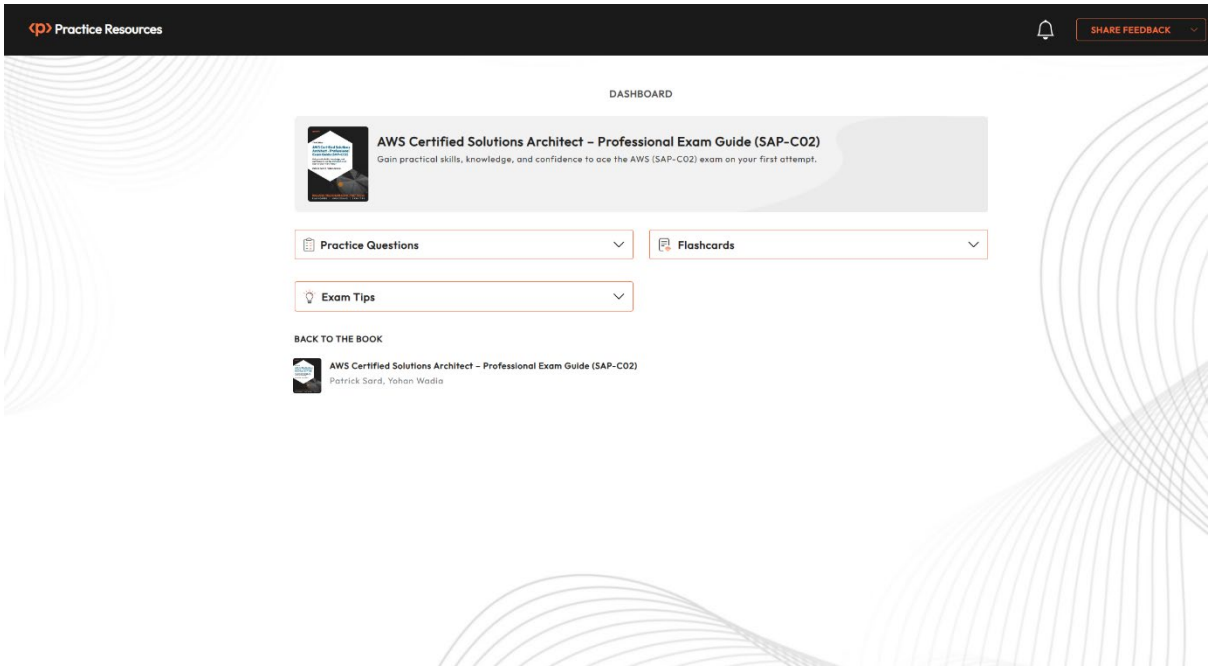


Figure 21.7 – Dashboard interface on a desktop device

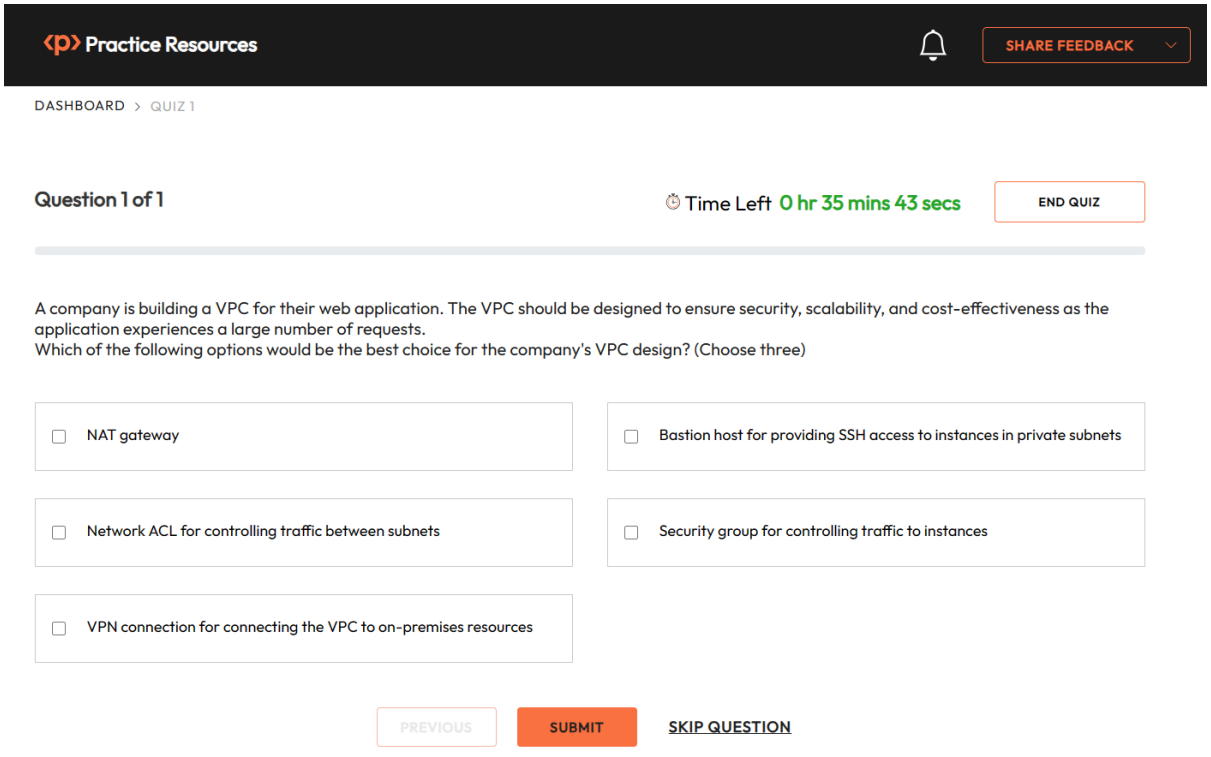


Figure 21.8: Practice questions interface on a desktop device

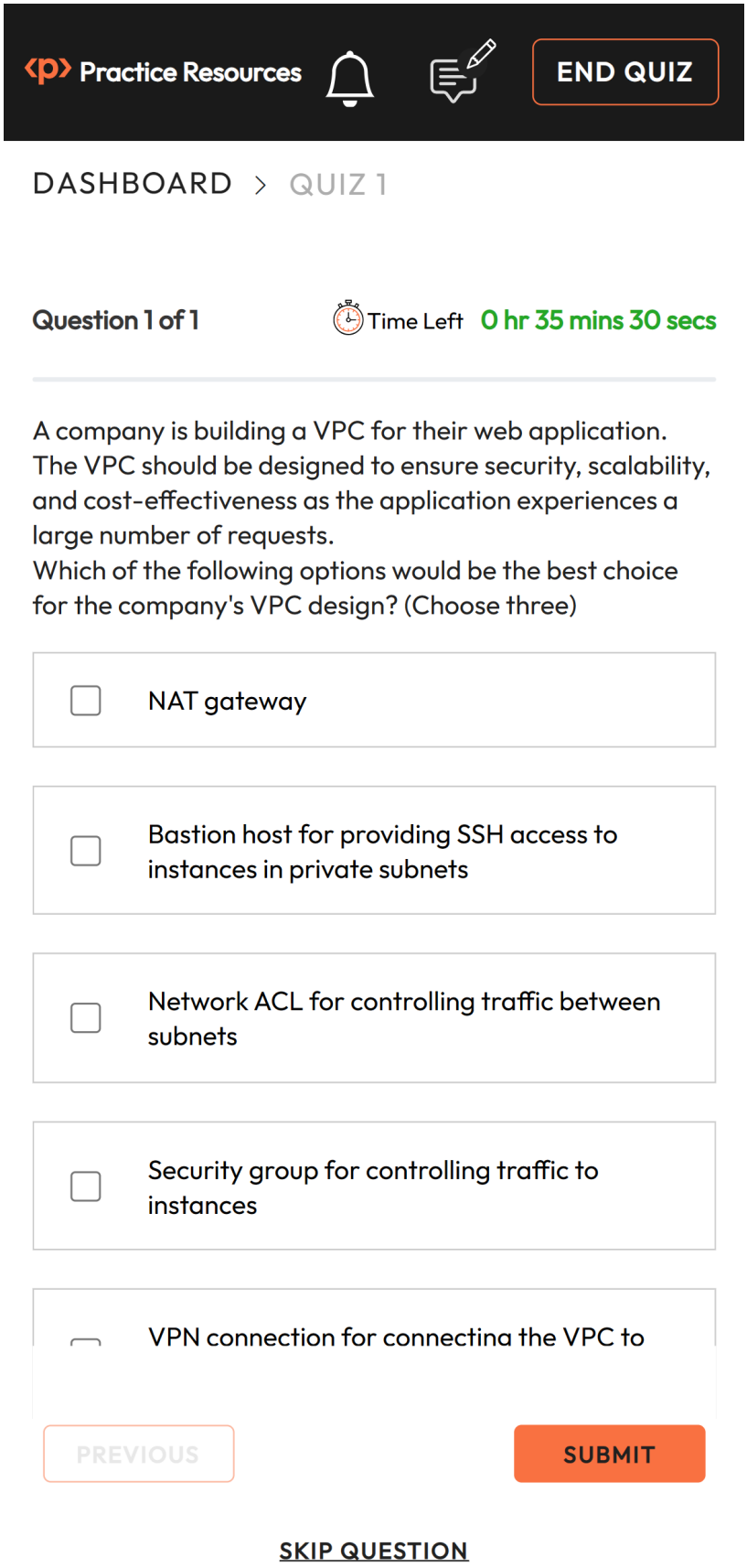


Figure 21.9: Quiz interface on a mobile device

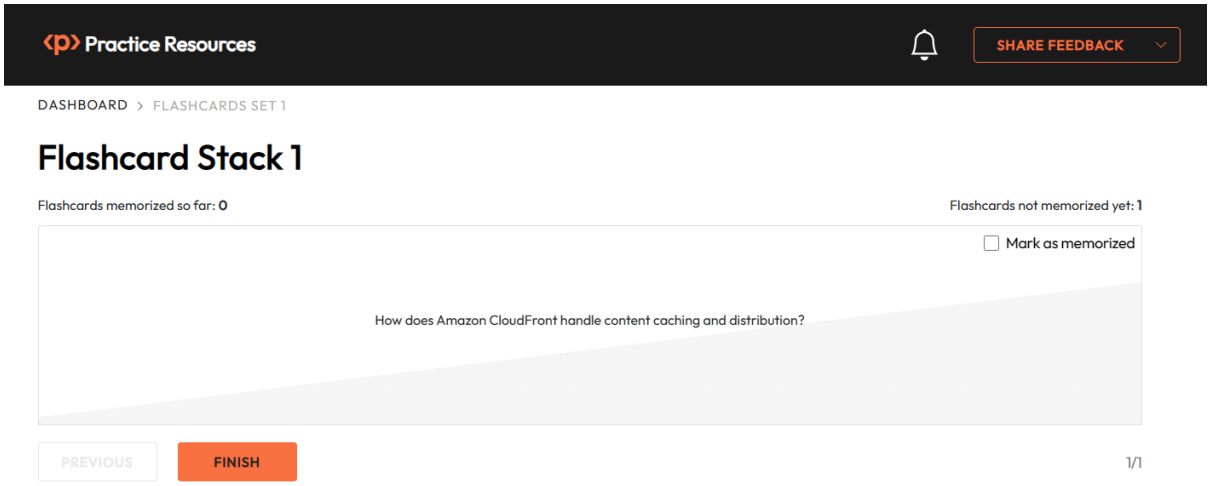


Figure 21.10: Flashcards interface

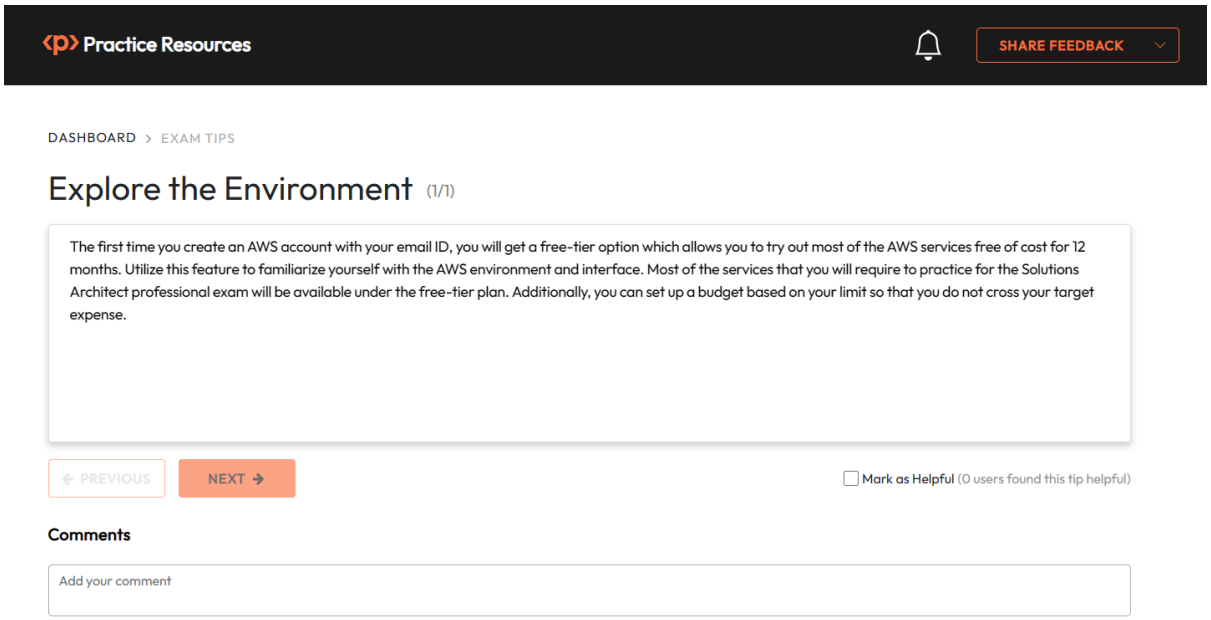


Figure 21.11 – Exam Tips Interface