2025

AnyGroupLLC- AWS Cloud Migration

PREPARED BY: GROUP 21

JIBIN JOY (JJOY283)

TARAK PATEL (TPAT586)

SHUBHAM GAIROLA (SGAI126)

SAIKAT BAIDYA (SBAI858)

Contents

1. Introduction

This document outlines the proposed Amazon Web Services (AWS) cloud architecture to support AnyGroupLLC's expansion into the New Zealand market. The design is based on a detailed analysis of the business requirements gathered from the CTO, CFO, CISO, and IT Manager. The primary objectives of this solution are to ensure **high availability, robust security, cost-effective scalability,** and **operational efficiency** for the new e-commerce platform and supporting systems.

The proposed solution leverages core AWS services to create a resilient, secure, and modern foundation that addresses immediate business needs while providing a clear pathway for future innovation using microservices and AI/ML.

2. Business Requirement Analysis

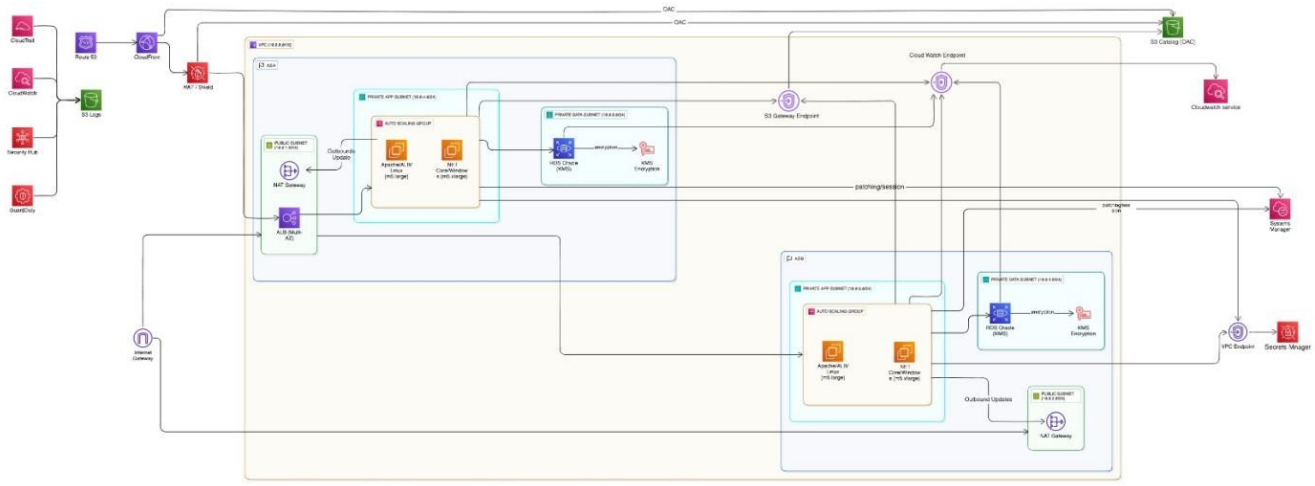
The design is driven by the following key business requirements:

- **CTO:** High availability for customer experience and scalability to handle 500k+ daily visits and holiday sales spikes.
- **CFO:** Cost management through a shift from Capital Expenditure (CapEx) to Operational Expenditure (OpEx) and solutions for better demand forecasting.
- **CISO:** A strong security posture to prevent DDoS attacks and data breaches, ensuring PCI DSS compliance for online payments.
- **IT Manager:** A manageable solution for a team of 15, reducing time spent on patching and maintenance to free up capacity for innovation.

3. Proposed Architecture Overview

The proposed architecture is a multi-tier, highly available system hosted across two Availability Zones (AZs) within a single AWS Region. The core components include:

- A secure Amazon Virtual Private Cloud (VPC) with isolated subnets.
- Web and application servers hosted on Amazon EC2 in an Auto Scaling Group behind an Application Load Balancer.
- A managed Oracle database using Amazon RDS deployed in a Multi-AZ configuration.
- Amazon S3 and Amazon CloudFront for scalable, global image delivery.
- AWS WAF and Shield for application security and DDoS mitigation.



AWS Architecture Diagram

4. Design Choices

4.1 Network Design (VPC, Subnets, CIDR)

Design: A single VPC with a 10.0.0.0/16 CIDR block is provisioned, spanning two Availability Zones. Each AZ contains:

- A **public subnet** (10.0.1.0/24 & 10.0.2.0/24) for resources that must be internet-facing.
- A **private application subnet** (10.0.10.0/24 & 10.0.20.0/24) for backend servers.
- A **private data subnet** (10.0.100.0/24 & 10.0.200.0/24) for databases.

Justification

- **Two AZs:** This is the foundational requirement for high availability and fault tolerance. If one data center fails, the application remains operational in the second AZ.
- **Subnet Isolation:** The tiered subnet design is a core security best practice. It allows for strict control of traffic flow between layers. Public subnets have a route to an Internet Gateway, while private subnets use NAT Gateways for controlled outbound access. Data subnets have no internet access, creating a secure enclave for sensitive data.
- **Large CIDR Block:** The /16 block provides over 65,000 IP addresses, ensuring ample room for future growth and resource provisioning.

4.1.1. VPC Endpoints for Enhanced Security and Cost Control

Design: To minimize data egress costs and keep control-plane traffic within the AWS network, enhancing security, the VPC is configured with several endpoints:

- A **Gateway Endpoint** for S3, allowing instances in private subnets to access S3 without a NAT Gateway.
- **Interface Endpoints (AWS PrivateLink)** for Secrets Manager, CloudWatch Logs, CloudWatch Monitoring, and AWS Systems Manager (SSM).

Justification: This ensures that sensitive operations like retrieving database credentials and sending logs never traverse the public internet, reducing the attack surface and improving performance. It also reduces data processing costs associated with NAT Gateways.

4.2. Security Design

Design: Security follows a defense-in-depth model.

- **Edge Security:** The site is fronted by **Amazon CloudFront** with **AWS WAF** and **AWS Shield Standard** (Advanced optional), enforcing TLS 1.2+ and filtering common web exploits and volumetric DDoS traffic.
- **Load Balancer Security:** The Application Load Balancer (ALB) security group is configured to only allow inbound HTTPS traffic from the **CloudFront origin IP prefix list**, and egresses only to the application tier on port 443.
- **Application & Data Security:** Application instances run in private subnets and accept traffic only from the ALB; they connect to the database on the Oracle port. The database security group only allows inbound from the application security group.
- **Encryption:** All data is encrypted at rest with AWS KMS (EBS, RDS, S3) and in transit (TLS). **AWS Secrets Manager** stores and rotates database credentials; instances retrieve secrets through a VPC interface endpoint.
- **Monitoring & Audit: AWS CloudTrail, AWS Config, GuardDuty, Security Hub, Inspector, and CloudWatch** provide comprehensive audit, threat detection, and alerting. ALB, WAF, CloudFront, and VPC Flow Logs are stored in S3 under defined retention policies.

Justification: This multi-layered approach ensures that a breach in one layer does not compromise the entire system. The specific configuration of allowing only CloudFront to communicate with the ALB is a critical security hardening measure. The use of managed security services provides enterprise-grade protection and automated compliance reporting, directly addressing the CISO's requirements for a strong security posture and PCI DSS compliance.

4.3. Compute Resources

Design: Amazon EC2 is chosen for the web and application tiers.

- **Web Tier (Frontend):** Apache web servers running on Linux. Instance type: m5.large (2 vCPU, 8 GiB RAM).
- **Application Tier (Backend):** .NET Core application servers running on Windows. Instance type: m5.xlarge (4 vCPU, 16 GiB RAM).
- **Auto Scaling:** Both tiers are placed in Auto Scaling Groups (ASGs) across two AZs.

Justification:

- **EC2:** The use of EC2 provides a direct "lift-and-shift" path for the existing applications, minimizing initial re-engineering effort and risk.
- **Instance Sizing:** The chosen instance types provide a performance parity or improvement over the existing physical servers while optimizing for cost on AWS.
- **Auto Scaling:** This is the critical component for meeting the CTO's scalability and the CFO's cost-efficiency goals. The ASG automatically adds instances during sales spikes (e.g., Halloween) and removes them during quieter periods, ensuring performance while minimizing cost.

4.3.1. Systems Management and Patching

Design: Access and patching are performed via **AWS Systems Manager**. Session Manager enables secure SSH/RDP connections without bastion hosts or open inbound ports. Patch Manager automates OS patching according to a defined schedule.

Justification: This eliminates a common attack vector (open management ports) and automates a time-consuming task for the IT team, directly freeing them up for innovation as requested by the IT Manager.

4.4. Storage and Database Resources

Design:

- **Database:** Amazon RDS for Oracle in a Multi-AZ deployment. Initial size: db.m5.2xlarge (8 vCPU, 32 GiB RAM) with 2 TB of storage.
- **Object Storage:** Amazon S3 is used to store all product catalogue images. Amazon CloudFront is used as a Content Delivery Network (CDN) to serve these images globally.

Justification:

- **RDS Multi-AZ:** This managed database service provides high availability, automated backups, and patching. The Multi-AZ feature ensures automatic failover in case of an AZ outage, providing database resilience. This reduces administrative overhead for the IT team.
- **S3 & CloudFront:** This combination is the industry standard for storing and delivering static content. It provides limitless, durable storage for images and ensures fast load times for customers anywhere in the world by caching content at edge locations.
- **CloudFront Configuration:** CloudFront defines two origins and behaviors: the path pattern /images/* routes to the private S3 bucket (secured by an Origin Access Control - OAC), while the default behavior routes to the ALB for dynamic content. This maximizes cache hit ratio, reduces origin load, and enhances security by ensuring S3 is not directly publicly accessible.

4.5. Cost Optimization

Design: After an initial baseline period, **Savings Plans** or **Reserved Instances** will be applied to stable components (e.g., RDS, baseline EC2 capacity). AWS Budgets and cost allocation tags will be implemented for detailed spend tracking and accountability.

Justification: This proactive cost management strategy ensures long-term cost efficiency. Savings Plans can reduce compute costs by up to 70% compared to On-Demand pricing, directly supporting the CFO's goal of better cost management and forecasting.

5. Addressing Business Requirements

- **Secure & Reliable:** The multi-AZ, multi-subnet design with WAF/Shield ensures the application is resilient to failures and malicious attacks, meeting the CISO's requirements.
- **Scalable & Highly Available:** Auto Scaling and Load Balancing allow the application to seamlessly handle from zero to millions of users, meeting the CTO's goals for launch and growth.
- **Cost-Optimized:** The move to an OpEx model and the use of Auto Scaling ensure AnyGroupLLC only pays for the IT resources they consume, addressing the CFO's primary objective. S3 is a highly cost-effective storage solution.
- **Operationally Efficient:** The use of managed services (RDS, ALB, S3) automates undifferentiated tasks like database patching and load balancer scaling, freeing the IT team to focus on innovation rather than maintenance, as requested by the IT Manager.

6. Design Pattern Recommendations

To ensure that AnyGroupLLC's cloud solution is secure, reliable, and scalable, three modern design patterns are recommended: the **Strangler Fig Pattern**, the **Event-Driven Architecture with Outbox**, and the **Bulkhead Isolation Pattern**. These patterns were chosen because they align directly with the concerns raised by the company's CTO, CFO, CISO, and IT Manager during the stakeholder interviews. Together, they provide a safe migration path from legacy systems, the ability to handle rapid demand spikes without over-provisioning, and resilience against failures or malicious attacks.

Strangler Fig Pattern is particularly important for AnyGroupLLC because the company is moving from an on-premises stack consisting of Apache web servers, .NET Core applications, and an Oracle database to a modern AWS-based solution. Instead of replacing everything in a risky "big-bang" migration, the Strangler approach allows new services such as the product catalogue, checkout, or image storage to be built on AWS while the legacy application continues to operate. A routing layer, such as an Application Load Balancer or API Gateway, directs some requests to the old system and others to the new cloud services. Over time, the legacy components are retired as more functionality is transferred. This pattern ensures that the launch in New Zealand can proceed smoothly without major downtime, while also reducing the IT Manager's concern about continuous firefighting and patching. By leveraging managed AWS services, the team can spend more time innovating rather than maintaining legacy infrastructure. Additionally, AWS Managed Services and Training resources help reduce the upskilling burden on the IT team and ensure they can adopt and operate the new platform efficiently with minimal overhead. Over time, these patterns enable a microservices architecture where individual business capabilities such as checkout, product catalogue, and inventory can be developed and deployed independently using container services like Amazon ECS or EKS.

Event-Driven Architecture with Outbox Pattern addresses the scalability and cost challenges that AnyGroupLLC faces, especially during seasonal events and holiday promotions. In an event-driven model, business processes such as order placement, inventory updates, and customer notifications generate events that are published to a messaging backbone like Amazon EventBridge or SNS and consumed asynchronously by downstream services running on AWS Lambda or ECS. This decoupling allows each service to scale independently and absorb sudden surges in traffic without overwhelming the system. The Outbox Pattern adds reliability by ensuring that events are never lost: when an order is written to the database, an associated event is also written to an outbox table in the same transaction, and a relay service publishes it reliably to the event bus. For AnyGroupLLC, this means that stock levels, sales forecasts, and customer communications stay consistent even under heavy load. It also supports the CFO's goals of shifting IT spending to a usage-based model, as compute resources scale only when needed, and provides clean data streams that can later feed AI and machine learning models for demand forecasting and personalization.

Bulkhead Isolation Pattern strengthens the company's resilience and security posture, which is a key concern for the CISO after the website's previous DDoS attack. In this pattern, different parts of the system are isolated into separate resource pools so that a failure or

traffic spike in one component does not cascade into others. For example, the checkout process can be placed in its own autoscaling group and database instance, while images are served separately from Amazon S3 and CloudFront. If a surge of image requests occurs during a marketing campaign, it will not affect the ability of customers to complete purchases. Similarly, API Gateway and WAF rules can apply different rate limits for catalog browsing versus payment processing, ensuring that critical transactions are always prioritized. By containing failures and protecting essential workflows, this pattern ensures that availability targets are met even under attack or extreme load, and it gives the business predictable costs and service-level guarantees. For physical store security systems, IoT devices such as cameras and sensors can be integrated through AWS IoT Core and Kinesis Video Streams to ensure that monitoring workloads remain isolated from customer-facing systems.

In addition to these patterns, the adoption of an event-driven architecture naturally enables AI/ML innovation. The clean data streams from the Outbox Pattern can feed into services such as Amazon SageMaker for personalized recommendations and Amazon Forecast for demand prediction. This allows AnyGroupLLC to pilot new technologies in New Zealand with minimal risk, aligning with the executive team's vision of using AI/ML to drive customer experience and operational efficiency.

Taken together, these three design patterns form a cohesive approach that addresses the company's business and technical requirements. The Strangler Fig Pattern provides a safe and incremental path to cloud migration; the Event-Driven Architecture with Outbox ensures scalability, cost efficiency, and reliable business processes; and the Bulkhead Isolation Pattern safeguards against cascading failures and security incidents. By adopting these patterns, AnyGroupLLC can deliver a secure, reliable, and scalable cloud platform for its New Zealand operations while laying the foundation for future innovation in analytics, machine learning, and global product integration.

7. Conclusion

The proposed AWS architecture provides a secure, reliable, and scalable foundation for AnyGroupLLC's New Zealand market entry. It directly addresses all stated business and technical requirements while establishing a modern platform that can evolve to incorporate advanced capabilities in AI/ML and microservices, ensuring the company's long-term competitive advantage.