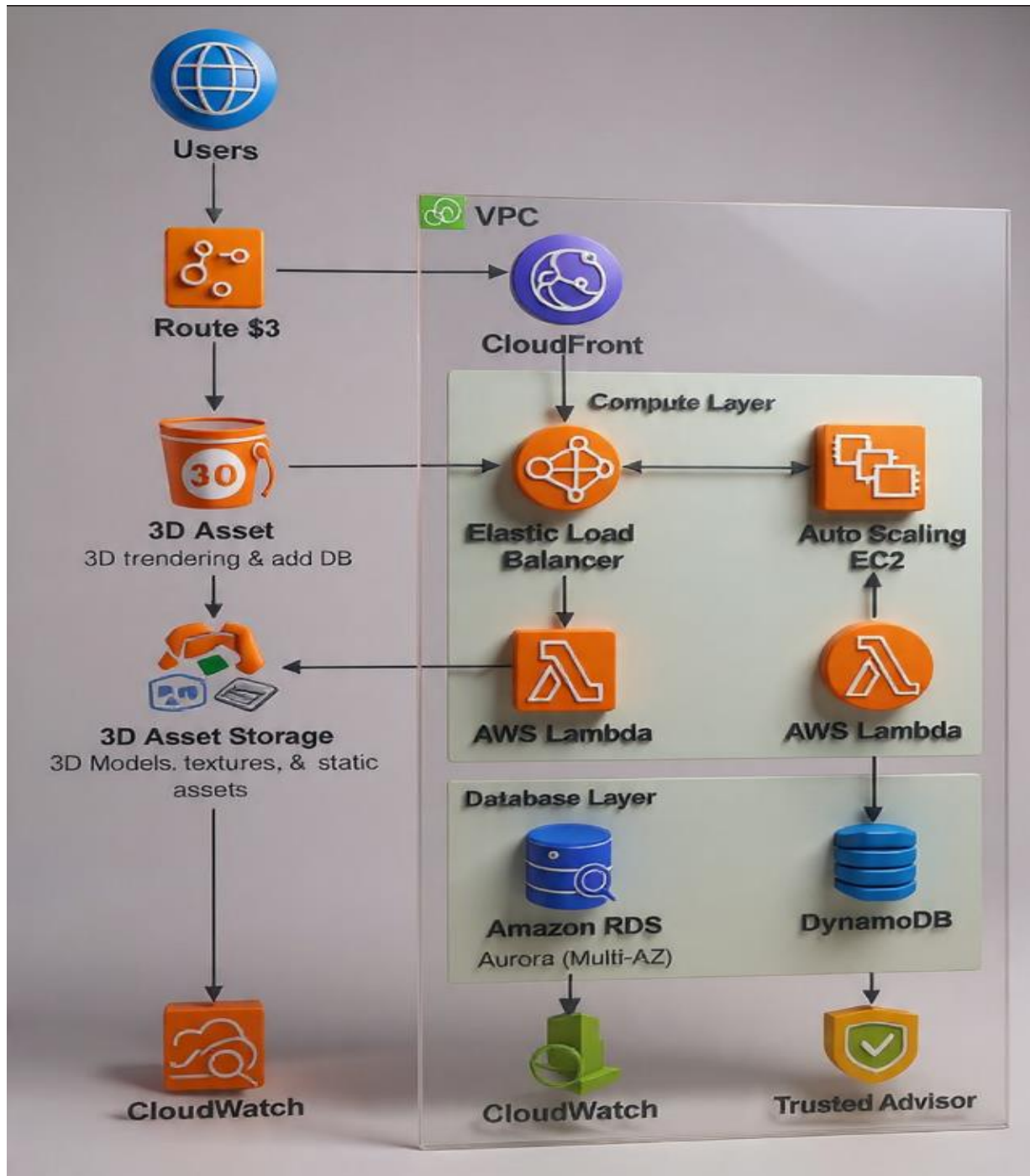


## 3D E-Commerce Platform Architecture on AWS: High Availability Design



## AWS Cloud Practitioner Project – High Availability Design

**Team Members:** Kwanele, Neo, Madimetja, Leah, Keketso

---

## Overview & Core Design Principles

This architecture supports a global 3D e-commerce platform where users interact with high-resolution 3D product models. The design strictly adheres to the AWS Well-Architected Framework, focusing on the following pillars:

<u>Pillar</u>	<u>Focus Area</u>
High Availability	No single point of failure (SPOF); multi-AZ deployments.
Scalability	Automatic scaling to handle peak global traffic.
Performance	Low latency delivery of large 3D assets worldwide.
Security	Defence-in-depth approach; encryption and least-privilege access.
Cost Efficiency	Utilizing serverless and pay-as-you-go services to minimize operational expenditure (OpEx).

## Key AWS Services Used and Rationale

<b><u>AWS Service</u></b>	<b><u>Function</u></b>	<b><u>Rationale</u></b>
<b>Route 53</b>	DNS & Global Traffic Routing	Routes users to the nearest, healthy application endpoint using Latency-Based Routing. Provides robust health checks (100% SLA) for fast failover.
<b>Amazon CloudFront</b>	Global Content Delivery Network (CDN)	Caches large 3D models (OBJ/GLB files), textures, and static assets at Edge Locations to deliver content with millisecond latency globally. Uses OAC (Origin Access Control) for S3 security.
<b>Amazon S3</b>	Durable Storage for 3D Assets	Provides 11 nines of durability and virtually unlimited storage capacity for high-resolution 3D files. Its integration with CloudFront is seamless and secure.
<b>Application Load Balancer (ALB)</b>	Layer 7 Traffic Distribution	Distributes incoming HTTP/HTTPS traffic across multiple EC2 instances or containers within a Virtual Private Cloud (VPC). Manages SSL termination and detailed request routing.
<b>Compute Layer</b>	Backend Business Logic & Rendering	Option A: EC2 Auto Scaling Group for demanding tasks like real-time 3D model configuration/rendering APIs. Option B: AWS Lambda for event-driven, lightweight tasks (e.g., cart updates, user authentication).
<b>Database Layer</b>	Data Persistence	DynamoDB (NoSQL) for high-read, low-latency product catalog lookups. Amazon RDS (Aurora Multi-AZ) for highly transactional data (orders, payments, inventory, customer profiles) requiring ACID compliance.
<b>Amazon CloudWatch</b>	Monitoring & Observability	Centralized collection of metrics, logs, and events across all services. Crucial for defining auto-scaling policies and setting up alerts for performance degradation.
<b>AWS Trusted Advisor</b>	Optimization & Best Practices	Continuously scans the environment against AWS best practices to ensure ongoing cost optimization, security posture, and fault tolerance.

## How the Architecture Meets the 5 Requirements

### 1. High Availability

- Multi-AZ Deployment: RDS (Aurora) and the EC2 Auto Scaling Group span multiple Availability Zones within an AWS Region.
- Decentralized Delivery: CloudFront caches content at edge locations, reducing dependency on a single region or origin.
- Automatic Failover: Route 53 health checks automatically re-route traffic away from unhealthy endpoints.

### 2. Scalability

- Horizontal Scaling: EC2 Auto Scaling dynamically adds or removes instances based on demand (CPU utilization, queue length).
- Serverless Scaling: Lambda and DynamoDB offer near-instant, massive, and automatic scaling capabilities.
- Elastic Storage: S3 and CloudFront provide virtually unlimited, auto-scaling capacity for global asset storage and delivery.

### 3. Performance

- Edge Caching: CloudFront significantly reduces the load on the origin (S3/EC2) and provides the fastest possible delivery to end-users worldwide.
- Specialized Databases: DynamoDB is optimized for sub-10ms product lookups, keeping the user experience snappy.
- Efficient Routing: ALB intelligently distributes traffic for optimal server resource usage.

### 4. Security

- VPC Isolation: Backend resources (EC2, RDS) are isolated in private subnets, only accessible via the ALB.
- Least Privilege: IAM Roles and policies enforce granular access control for all services.

- **Data Protection:** RDS enforces data-at-rest encryption. CloudFront OAC restricts S3 access only to CloudFront.

## 5. Cost Optimization

- **Pay-as-you-go:** Utilizing Lambda and DynamoDB eliminates idle compute costs for sporadic workloads.
- **Caching Efficiency:** CloudFront minimizes expensive requests to the origin (EC2/S3) by serving content from cheaper edge caches.
- **Resource Right-Sizing:** The Auto Scaling Group and DynamoDB auto-scaling ensure resources precisely match the current load, preventing over-provisioning.

## Trade-offs and Challenges

Area	Trade-off / Challenge	Mitigation Strategy
<b>EC2 vs. Lambda</b>	Lambda has a time limit (up to 15 minutes) and is not ideal for long-running, complex 3D rendering jobs.	Use EC2 Spot Instances within the Auto Scaling Group for cost-effective heavy rendering tasks that can tolerate interruption.
<b>RDS vs. DynamoDB</b>	DynamoDB requires careful data modeling and lacks the flexibility of SQL for complex analytical queries.	Use Amazon Redshift (optional) to pull data from RDS for complex business intelligence (BI) analysis, keeping the operational databases lean.
<b>3D Asset Size</b>	High-resolution 3D models and textures can be hundreds of megabytes, impacting initial load times.	Implement CloudFront Compression and use multi-tiered caching (e.g., S3 Intelligent-Tiering) to manage storage costs for less-frequently accessed models.
<b>VPC Design</b>	Initial setup of private subnets, NAT Gateways, and Security Groups adds configuration complexity.	Use Infrastructure as Code (e.g., AWS CloudFormation) to define the VPC structure reliably and repeatably.

## Insight

This taught us how to integrate and optimize AWS services to solve the complex technical challenges of a global, high-demand platform as this research serves as a practical lesson in designing a modern, highly available, and scalable cloud application using AWS.

## **Conclusion**

The proposed architecture successfully leverages a suite of AWS managed services—particularly the combination of CloudFront, S3, Auto Scaling compute, and a specialized database layer (DynamoDB/RDS)—to create a global, highly available, and cost-effective foundation for a high-traffic 3D e-commerce platform. This design minimizes operational overhead while ensuring resilience against both sudden traffic spikes and component failures.