

# INTRODUCTION

This document provides an overview of the technical architecture for a next-generation 3D e-commerce web application hosted on AWS. The platform allows users to interact with 3D models of products before purchasing. With millions of users expected globally, the architecture must deliver fast performance, high availability, security, and cost efficiency. This document explains why specific AWS services were chosen, how they address the five core requirements, and the design trade-offs or challenges encountered.

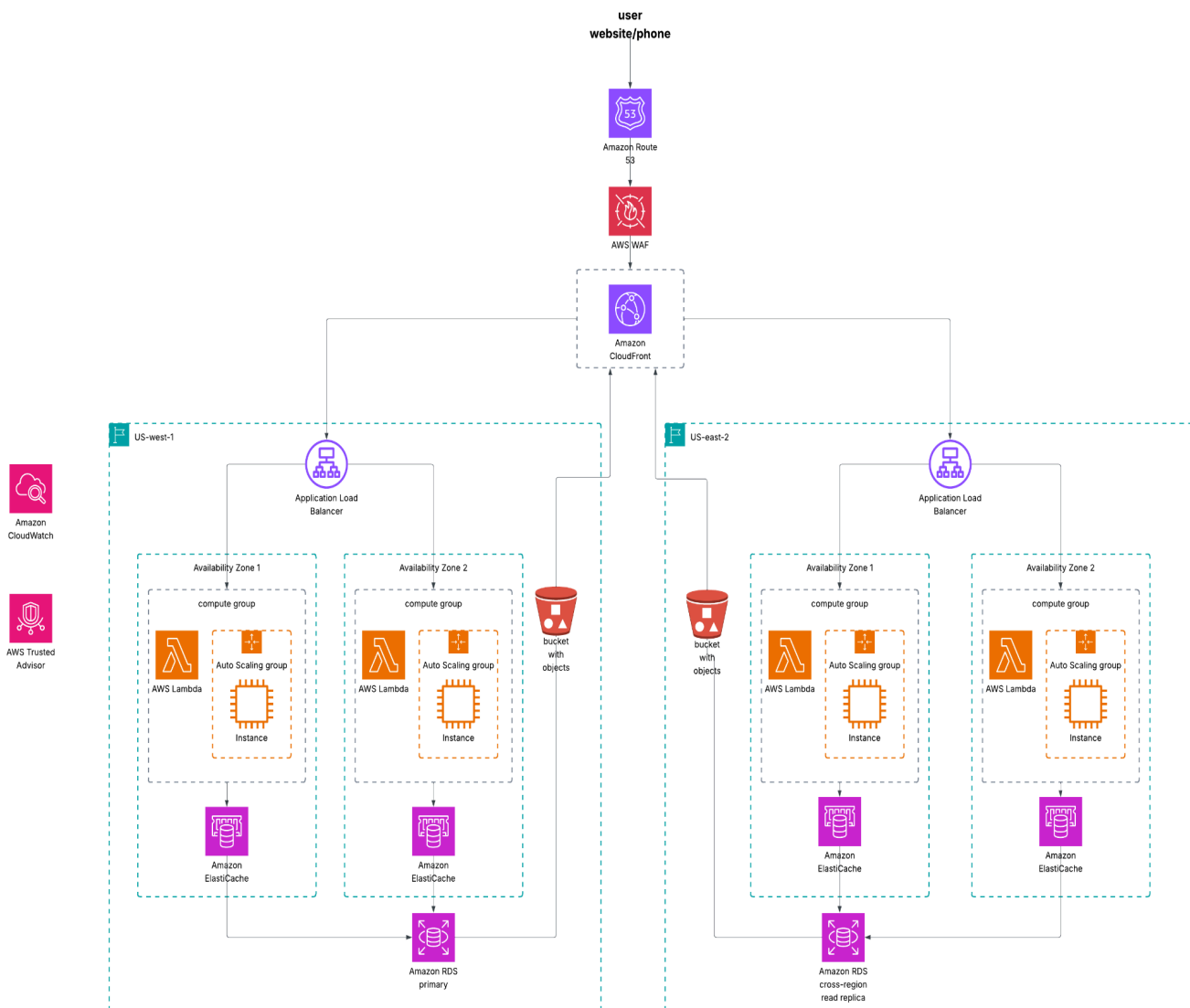


Figure 1: 3D e-commerce aws architecture

## CHOSEN AWS SERVICES & JUSTIFICATION

### **Amazon Simple Storage Service (S3)**

Stores 3D assets, product images, and static website content.

Offers high durability, global availability, and lifecycle policies for cost optimization.

### **Amazon Route 53**

Provides domain registration, DNS management, and traffic routing.

Ensures global availability with latency-based routing to direct users to the nearest AWS region.

### **AWS Lambda**

Serverless compute to handle backend logic such as API requests, order processing, and event-driven workflows.

Automatically scales without provisioning servers.

### **Amazon CloudWatch**

Monitors application performance, resource utilization and logs.

Provides alerts, dashboards, and automated responses for system health.

### **Amazon ElastiCache**

In-memory caching to reduce database load and deliver fast response times for frequently accessed product data. Reducing database load, giving customers sub-millisecond response times.

### **Application Load Balancer (ALB)**

Distributes incoming traffic across multiple EC2 instances or Lambda functions.

Supports advanced routing (e.g., path-based, host-based) for microservices. For example if a lot of traffic comes into the web application, since we are expecting millions, the load balancer will distribute the traffic across the availability zone and/trigger the creation of more EC2 instances.

### **Amazon EC2**

Provides compute power for workloads requiring more control, such as rendering 3D previews or heavy API processing for example if a user customizes furniture by changing color or size, an EC2 GPU instance generates the preview in real time. This serves as flexible compute power for advanced rendering tasks as a clear cut benefit.

### **Amazon EC2 Auto Scaling**

Automatically adds or removes EC2 instances based on traffic demand.

Ensures consistent performance and optimizes cost.

### **AWS Trusted Advisor**

Provides recommendations on cost optimization, security, fault tolerance, service limits and performance.

### **Amazon RDS (Relational Database Service)**

Stores structured e-commerce data (customers, orders, transactions). For example when a user completes checkout, the order details are stored in the RDS primary database. A read replica in another region supports global users checking their order history. Benefit to this it offers Multi-AZ failover ensures orders are not lost even if one database instance fails.

### **AWS WAF(Web Application Firewall)**

It is a cloud-native security service that monitors and filters HTTP and HTTPS requests to your web applications and APIs, it enforces access control based on customizable rules. It lets you specify conditions on incoming web traffic(like IP addresses, URL paths, queries and known attack signatures) and then allow, block or count requests that match those conditions. For example, our 3D ecommerce web application is going to block requests containing SQL injection patterns or known malicious user agents, while allowing normal user traffic. By default, a web ACL in AWS WAF either allows or blocks all requests, but you can attach custom rules so that specific unwanted traffic is filtered out.

## **REQUIREMENTS**

### **High Availability**

- Achieved with Route 53 global DNS failover, ALB across multiple Availability Zones, RDS Multi-AZ, and S3's built-in redundancy.

### **Scalability**

- Lambda scales automatically for event-driven workloads.
- EC2 Auto Scaling adjusts capacity during peak shopping times.
- S3 and CloudFront scale seamlessly for asset delivery.

### **Performance**

- ElastiCache speeds up data retrieval.
- ALB ensures traffic is routed efficiently.
- CloudWatch identifies bottlenecks early.
- Amazon CloudFront CDN can be added to accelerate global delivery of 3D assets and ensure low latency.
- EC2 instances optimized for rendering.

### **Security**

- AWS IAM for identity and access management.
- Trusted Advisor highlights security misconfigurations.
- S3 bucket policies and encryption protect assets.
- RDS encryption at rest and SSL/TLS for data in transit.
- ALB + WAF protects against common web attacks.

## **Cost Optimization**

- Auto Scaling reduces unused EC2 costs.
- Lambda follows a pay-per-use model.
- S3 lifecycle rules move older assets to cheaper storage classes.
- Trusted Advisor provides ongoing cost-saving recommendations.

## **DESIGN TRADE-OFFS OR CHALLENGES**

- The design balances performance with cost by using a mix of EC2 instances and AWS Lambda, while EC2 provides consistent performance for heavy workloads(host application logic) and Lambda reduces costs during low traffic periods.
- Additionally, using both Amazon RDS and CloudFront allows flexibility and increases complexity in data management.
- Multi-region and multi-AZ designs improve resilience but add cost and operational challenges in managing failovers.
- Cloudwatch provides insights but detailed logs across regions can be costly and excessive alerts may overwhelm operations teams.
- Strict WAF/IAM rules enhance protection but risk blocking legitimate users if misconfigured.