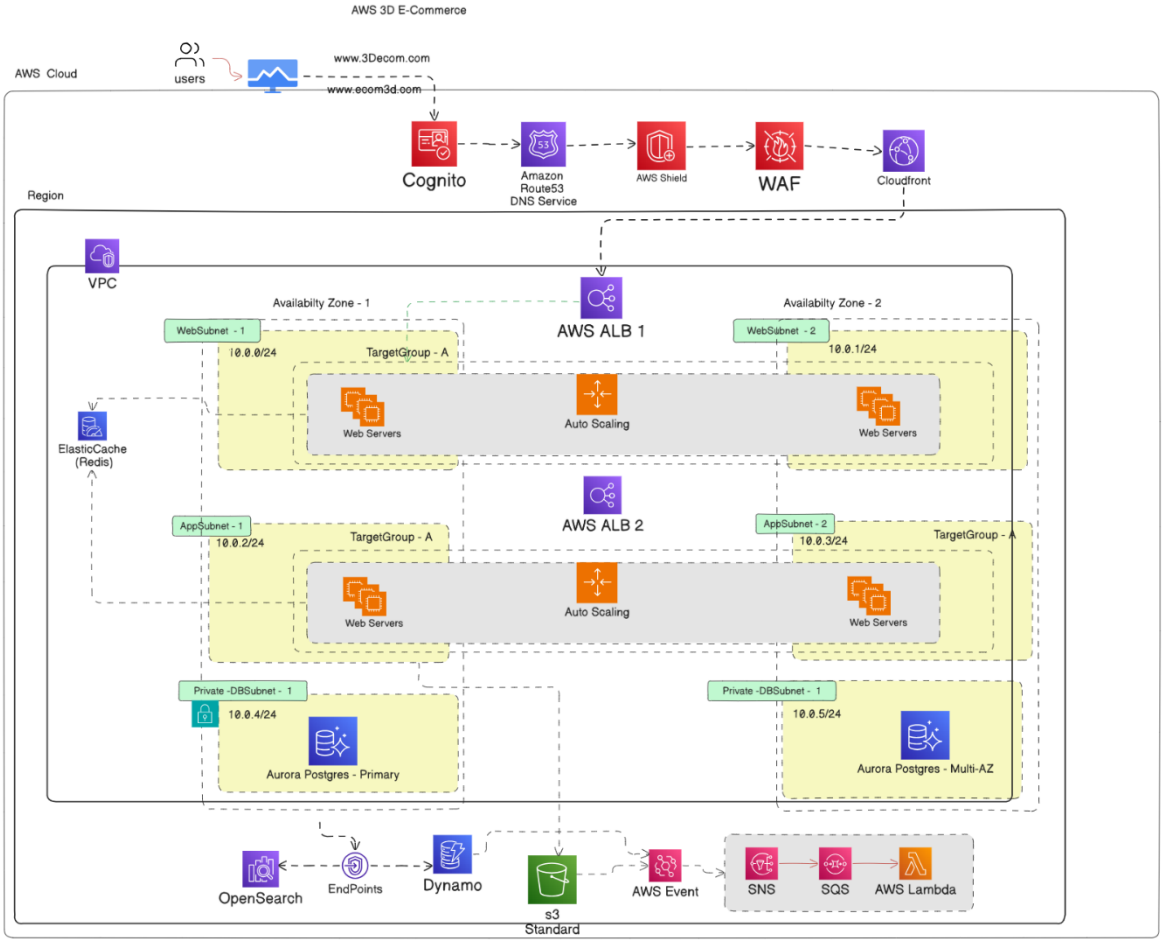
3D E-Commerce Platform on AWS — Architecture Documentation

1. OVERVIEW

This platform enables users to interact with 3D models of products globally. It is designed to deliver high performance, availability, and security while being cost-efficient. The architecture leverages AWS services for static asset storage, dynamic APIs, real-time services, and global distribution.



eraser

2. ARCHITECTURE COMPONENTS

Layer / Domain	Service(s)	Role & Justification
FRONT-END & GLOBAL DELIVERY	Users (Web & Mobile via HTTP/3 over QUIC with TLS)	Entry point for clients, leveraging modern, secure protocols for faster connections.
	Amazon Route 53	DNS resolution, traffic routing, and failover to ensure availability across AZs and regions.
	Amazon CloudFront	Global CDN caching + WAF integration, reduces latency and protects applications before requests reach ALBs.
STATIC & DYNAMIC CONTENT	Amazon S3	Durable object storage for static assets, backups, and uploads with versioning, lifecycle policies, and SDK integration.
COMPUTE	Elastic Load Balancer (ALB/NLB)	Distributes HTTP/HTTPS traffic. ALB supports Layer 7 routing, WebSocket, and content-based routing for efficient tiering.
	Application Load Balancer (ALB x2)	One ALB for web tier, one for app tier, balancing across multi-AZ deployments.
	Auto Scaling	Dynamically scales EC2/web servers behind ALBs for performance and cost efficiency.
	AWS Batch	Event-driven and batch processing (e.g., asset pipelines), integrated with SQS, SNS, and EventBridge.
DATA LAYER	Amazon DynamoDB (Global Tables)	Serverless NoSQL DB for catalogs, carts, and sessions with multi-region replication for global low latency.
	Amazon Aurora (Global Database)	Relational DB for orders/payments with ACID compliance, strong consistency, and global failover.
	Amazon ElastiCache (Redis)	In-memory cache for low-latency queries and reducing DB load, improving rendering performance.
	Amazon OpenSearch	Full-text search and analytics for product discovery and filtering.
SECURITY & IDENTITY	Amazon Cognito	Authentication & authorization with MFA, social login, and JWTs.

	AWS KMS & Secrets Manager	Encryption at rest (KMS) + secure credential management and rotation (Secrets Manager).
	AWS WAF & AWS Shield	WAF for OWASP protection, Shield for managed DDoS defense.
	Networking Controls (Private Subnets, SGs, NACLs, VPC Endpoints)	Isolates workloads, enforces least privilege, secures private access to services.
OBSERVABILITY	Amazon CloudWatch	Metrics, logging, and alarms for system health monitoring.
	AWS X-Ray	Distributed tracing for identifying performance bottlenecks.
	AWS CloudTrail	API call logging for audit/compliance.
	AWS Trusted Advisor	Best-practice checks for cost, security, performance, and fault tolerance.

3. HOW THE ARCHITECTURE MEETS KEY REQUIREMENTS

REQUIREMENT	Implementation	Justification
HIGH AVAILABILITY	Multi-AZ deployments, CloudFront edge caching, Route 53 failover, DynamoDB Global Tables, Aurora Global DB	Ensures resilience across regions, reduces downtime, and provides global access to data with minimal disruption.
SCALABILITY	Auto-scaling (Lambda), CloudFront, on-demand DynamoDB, Aurora Serverless v2, SQS decoupling	Supports unpredictable workloads by scaling resources dynamically while preventing bottlenecks.
PERFORMANCE	Optimized 3D assets, HTTP/3 delivery via CloudFront, ElastiCache, global DB replicas	Improves speed and responsiveness for global users by reducing latency and delivering cached content.
SECURITY	IAM least-privilege, Cognito with MFA, WAF & Shield, KMS encryption, private subnets	Protects system and data with strong authentication, encryption, and network isolation.
COST OPTIMIZATION	Serverless, S3 Intelligent-Tiering + Glacier, Spot Instances, CloudFront caching, Savings Plans	Reduces waste by only paying for what we use, automating storage tiers, and leveraging discounted resources.

4. DESIGN TRADE-OFFS & CONSIDERATIONS

Area	Trade-Off / Challenge	Mitigation / Rationale
Compute Choice (EC2 vs Lambda)	EC2 provides high control and handles heavy workloads but can be costly and slow to scale. Lambda is cost-efficient and auto-scales but has execution time and memory limits.	Use EC2 for compute-heavy rendering or batch tasks; use Lambda for lightweight, event-driven tasks. This hybrid approach balances cost and performance.
Database Design (DynamoDB vs Aurora)	DynamoDB is ultra-fast for NoSQL operations but may be limited for complex relational queries. Aurora supports relational data and ACID transactions but scaling multi-region is more expensive and complex.	Use DynamoDB for catalog, sessions, and carts for low-latency reads. Use Aurora Global Database for orders and payments requiring strong consistency.
Global Delivery vs Consistency	CloudFront caching improves performance but may serve slightly stale 3D asset versions until cache invalidation.	Implement proper cache invalidation and versioning of assets in S3 to minimize inconsistencies.
Security vs Usability	Strong security (MFA, WAF, private subnets) may slightly increase friction for developers and users.	Balance usability with security by providing social login, Cognito user pools, and streamlined private networking for internal services.
Cost Optimization vs Performance	Using Spot Instances, serverless, or Glacier reduces costs but may introduce latency or availability trade-offs for critical tasks.	Critical workloads run on On-Demand or Reserved Instances, while non-critical processing and long-term storage use Spot/Glacier.
3D Rendering Performance	Serving large 3D assets globally can be bandwidth-intensive, causing latency for users in remote regions.	Use CloudFront edge caching, optimized file formats (glTF, Draco compression), and streaming techniques to reduce load times.
Monitoring Complexity	Observability across multiple services (EC2, Lambda, Aurora, DynamoDB, CloudFront) can be complex.	Use CloudWatch dashboards, X-Ray for distributed tracing, and CloudTrail for audits to maintain centralized visibility.