



Group Members:

1. CYNTHIA MOKGOEBO
2. PALESA PHAKATHI
3. JOAO AMARAL
4. MAGAU AZWIFARWI
5. MIKELANI MALULEKA
6. PHUMLA SHARON MASHABANE

AWS Architecture for 3D E-Commerce Platform

Introduction

Our team has designed a cloud architecture on AWS to support an interactive 3D e-commerce platform highlighting on the AWS services to support a platform that delivers 3D models and web applications with high reliability and efficiency.

When deciding on the services, we focused on five essentials:

High availability – keeping the platform up and running 24/7.

Scalability – adjusting to user demand without manual intervention.

Performance – fast page loads, responsive rendering, and quick interactions.

Security – protecting both our users and our data.

Cost optimization – staying efficient and avoiding unnecessary spend.

1. 3D Model Storage & Management – Amazon S3

Amazon S3 is the best place to store our 3D models. It's durable, infinitely scalable, and easy to connect with CloudFront for global delivery. On top of that, it gives us different storage classes, so we don't pay for more than we need.



Trade-Offs: The downside is that S3 isn't the fastest when it comes to data retrieval compared to block storage. If we use archive tiers like Glacier, pulling files out can also get expensive. We'll need to set up strict IAM policies to avoid accidental overexposure.

2. Content Delivery – Amazon CloudFront

It caches content in edge locations around the world, reducing lag and improving the overall user experience. It also has built-in protection against DDoS attacks, which adds another layer of security.

Trade-Offs: CloudFront can be tricky to manage at times—cache invalidations and TTL settings require some attention. If not tuned properly, users could end up seeing outdated content. And while AWS has a strong global edge network, coverage in certain regions is still limited.

3. Web Application – AWS Lambda

It's serverless, meaning we don't have to manage infrastructure, and it automatically scales to meet demand. This is cost-effective too, since we only pay when functions run.

Trade-Offs: Its functions can only run for up to 15 minutes, which might not fit every task. There's also the "cold start" issue, where infrequently used functions can take a little longer to respond. At very high usage, costs could outpace traditional servers.

4. Database – Amazon RDS

Amazon RDS for our relational data needs (user accounts, metadata, etc.). It's fully managed, handles backups and replication, and keeps things simple while still being reliable. With Multi-AZ deployment and read replicas, we'll have both resilience and scalability covered.

Trade-Offs: RDS isn't as elastic as DynamoDB or Aurora Serverless, and costs go up with replicas and failover setups. Schema management and query optimization are still on us as a team.



5. Security & Access – AWS IAM

AWS IAM is built into the AWS ecosystem and lets us control access at a very fine-grained level. We can enforce least-privilege policies across users, services, and groups without additional costs.

Trade-Offs: IAM can get complicated when policies pile up, and a misstep could open unwanted access. It's also AWS-specific, so if we want to connect outside systems, we'll likely need to add something like SSO.

6. Performance Optimization – AWS Auto Scaling

Auto Scaling ensures we're never under- or over-provisioned. It automatically adjusts resources to match demand and replaces unhealthy instances when needed. This helps us maintain performance during heavy traffic without overspending.

Trade-Offs: Scaling isn't always instant—it takes time to kick in. If our policies aren't tuned well, we could see "thrashing," where resources scale up and down too quickly. Keeping a small baseline of reserved resources may still be necessary for stability.

7. Cost Management – AWS Cost Explorer

It gives us visibility into where our money is going, highlights unused resources, and helps us optimize spend over time.

Trade-Offs: Cost Explorer only gives us insights it won't take action for us. Some reports can also lag by up to 24 hours. We must consider third-party tools later.

8. Monitoring & Logging – Amazon CloudWatch

It integrates smoothly with all AWS services, letting us monitor performance, set up alarms, and log events in one place. This helps us catch issues early and troubleshoot faster.



Trade-Offs: The challenge is cost—large amounts of logs and metrics can add up quickly. Analysing logs deeply might also require extra tools, and setting up advanced dashboards isn't always straightforward.

Below is our 3D architecture Diagram for the selected services :

