AWS 3D E-Commerce Architecture Design

Group Members: [**Mokgadi**], [**Kazizwe**], [**Tiisetso**], [**Lufhuno**] & [**Ivyn**]

Course: AWS re-Start Cloud computing Programme

Date given: 01 December 2025

Date due: 05 December 2025

---

1. **Introduction**

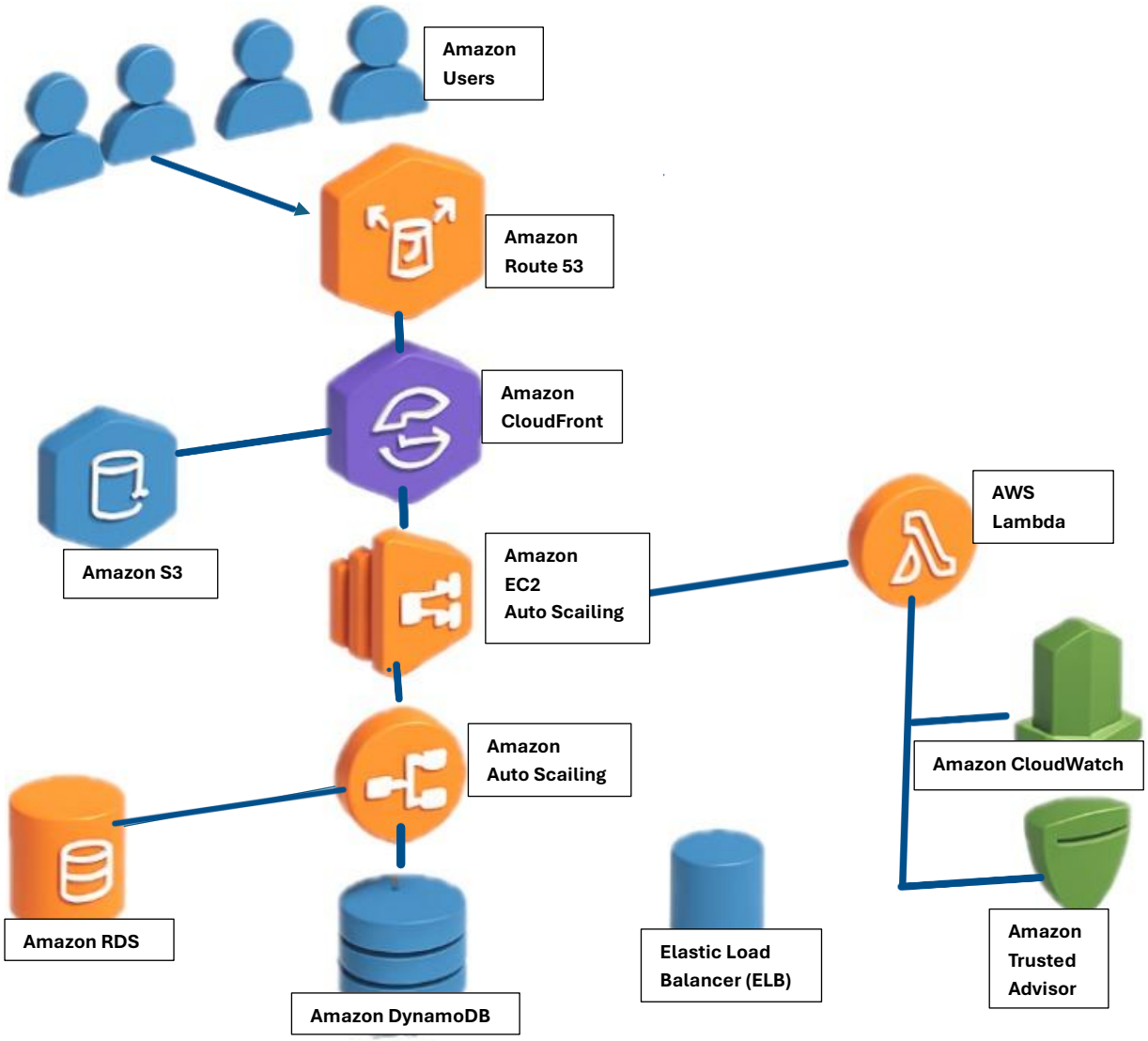
We are a startup team building a new kind of online shop where customers can see and play with 3D models of products (furniture, gadgets, clothes, etc.) before they buy.

We chose AWS because it can help us serve millions of people worldwide, to keep the site fast, safe, and it is not too expensive.

2. **What the project needs**

- Always online (24/7)
- Can handle huge traffic spikes (for example Black Friday)
- Super fast loading of 3D models
- Very secure (protect customer data and payments)
- Not waste money

3. **Our 3D Diagram-Architecture**



Main AWS pieces we use:

- Amazon S3 → stores all 3D models and pictures
- Amazon CloudFront → sends files from the closest city to the user
- Amazon EC2 → powerful computers that show the 3D models (with GPU)
- AWS Lambda → does quick jobs without keeping servers on all the time
- Amazon RDS → keeps product details, orders, customer info
- Amazon DynamoDB → stores shopping carts and fast-changing data
- Elastic Load Balancer (ELB) → shares traffic between EC2 servers
- Amazon Route 53 → our website address and automatic backup routing

- Amazon CloudWatch & Trusted Advisor → watches everything and tells us how to save money

4. **Why we chose each service**

- S3: cheap, safe, and can hold huge 3D files
- CloudFront: makes 3D models load in seconds anywhere in the world
- EC2: we need strong GPUs for smooth 3D experience
- Lambda: perfect for small tasks, we only pay when it runs
- RDS: good for normal tables (products, orders)
- DynamoDB: super fast for shopping carts and live data
- ELB: keeps the site alive even if some servers crash
- Route 53: smart DNS that switches to a backup if needed
- CloudWatch & Trusted Advisor: help us see problems and lower the bill

5. **How our design meets the requirements**

Requirement:	**How we do it**
Always online	Everything is in 2–3 Availability Zones + Route 53 automatic failover
Can grow	EC2 Auto Scaling adds servers when busy; Lambda grows by itself
Fast	CloudFront caches files near users + powerful GPU servers
Secure	Encryption everywhere, least-privilege permissions, WAF, Guard Duty
Cheap	Lambda + Auto Scaling + Spot instances + Trusted Advisor tips

6. **Trade-offs we noticed**

- EC2 with GPUs is fast but costs more → we only use it when really needed
- Lambda is cheap but can be slow the first time (cold start) → we use it only for short jobs
- CloudFront makes the site fast but adds a small monthly cost → totally worth it
- Using two databases (RDS + DynamoDB) is a bit more work, but gives the best speed

7. **What we learned as a team**

- Speed for 3D shopping is everything – caching and CDN are a must
- Never build everything yourself; AWS managed services save tons of time
- Auto Scaling and serverless are the best way to survive traffic spikes without spending too much
- Security and cost must be planned from day one, not added later
- CloudWatch is like a free doctor for your system – it tells you exactly what's wrong

8. **Quick look at AWS Well-Architected Framework:**

We checked our design against the official AWS rules (6 pillars):

Pillar	Our Score	Why we got this score
Operational Excellence	5/5	Everything automated with CDK and Code Pipeline
Security	5/5	Encryption + IAM + WAF + Cognito
Reliability	5/5	Multi-AZ + failover + backups
Performance Efficiency	5/5	CloudFront + GPU servers + caching
Cost Optimization	4.5/5	Lambda + Auto Scaling + Spot – still room for small tweaks
Sustainability	4/5	Good use of CloudFront and Auto Scaling

9. **Conclusion**

With S3, CloudFront, EC2, Lambda, RDS, DynamoDB, ELB, Route 53, and CloudWatch we built a real-world 3D shopping platform that is fast, always online, secure, and doesn't cost a fortune.

This design can grow with our startup and give customers an amazing experience.

10. **References**

- AWS Well-Architected Framework – https://aws.amazon.com/architecture/well-architected/
- AWS Documentation – https://docs.aws.amazon.com/

Submission Statement

We, the group 2 members listed above, confirm that this is our own group work. All sources are cited and we all contributed to the design and document.

END!!