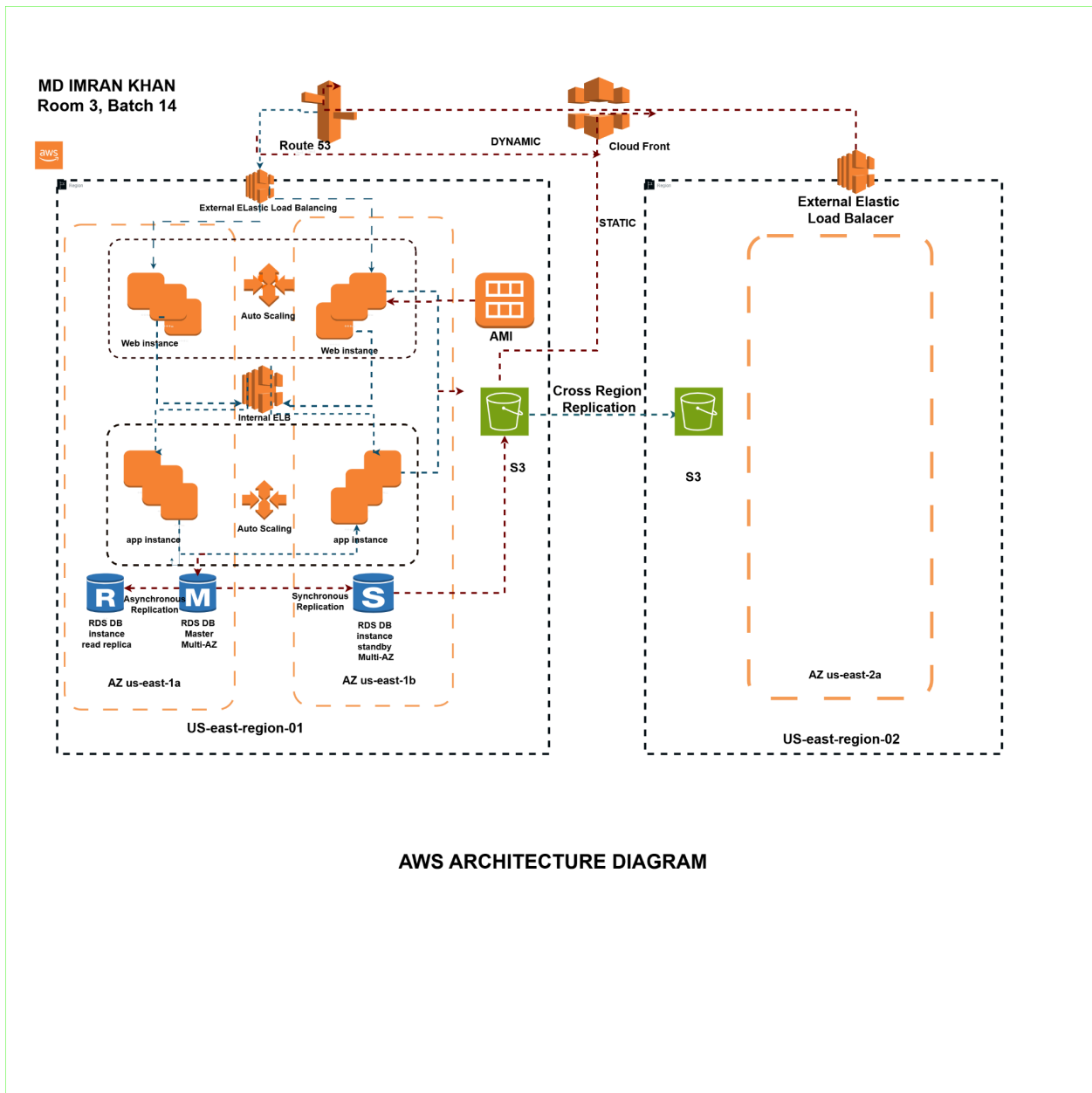


AWS Architecture Diagram



AWS ARCHITECTURE DIAGRAM

Scenario

Imagine you're running a large-scale **e-commerce platform** — let's call it **ShopMaster.com** — which sells products globally. You want your application to:

- Be **highly available** (no downtime if a server or AZ fails)

- Be **scalable** (handle sudden spikes during sales events like Black Friday)
 - Be **fault-tolerant** (data is safe even if an AZ goes down)
 - Deliver **static content quickly worldwide**
-

Step-by-Step Explanation of the Diagram

1. Users & DNS Routing

- **Route 53** (top left)
Your domain (**shopmaster.com**) is hosted in Route 53, which routes traffic to your infrastructure.
 - **Dynamic content** (API calls, database-driven pages) → goes through **External Elastic Load Balancing**.
 - **Static content** (product images, CSS, JavaScript) → served via **CloudFront** CDN.
-

2. Web Layer

- **External Elastic Load Balancer (ELB)**
 - Distributes incoming traffic across **Web Instances** in **multiple AZs** (**us-east-1a** and **us-east-1b**) to avoid a single point of failure.
 - Example: If AZ **us-east-1a** fails, users still reach the site via **us-east-1b**.
- **EBS (Elastic Block Storage)**
 - Stores app code/data for EC2 instances.
 - Snapshots are taken and stored in S3 for backup.
- **Auto Scaling**
 - Automatically launches more web instances if traffic spikes.
 - Example: Black Friday sale → Auto Scaling adds 10 more EC2 instances in seconds.

3. Application Layer

- **Internal ELB**
 - Balances traffic between **App Instances** (business logic layer) across AZs.
 - Web layer calls API endpoints here for cart management, order processing, etc.
- **Auto Scaling (App Layer)**
 - Scales application servers independently from the web layer.
 - Example: If checkout requests spike, more app servers are launched.

4. Database Layer

- **RDS Multi-AZ Setup**
 - **Master DB** (RDS DB Master instance) in **us-east-1a**.
 - **Standby DB** (RDS DB Standby instance) in **us-east-1b** — synchronous replication ensures zero data loss on failover.
 - **Read Replica** (RDS DB Read Replica) handles read-heavy queries asynchronously (useful for product catalog browsing).

5. Backup & Disaster Recovery

- **EBS Snapshots & RDS Snapshots** → stored in S3.
- **Cross-Region Replication** to another AWS Region (**us-east-2**) for disaster recovery.
 - If the **entire us-east-1 region** goes down, you can spin up your app in **us-east-2** using the replicated data.

6. CloudFront (CDN)

- **Static content** (images, CSS, JS) cached at AWS Edge locations worldwide.
 - Reduces latency for global users (e.g., someone in Europe loads the site faster even though servers are in the US).
-

Real-Life Flow

1. A user visits `shopmaster.com` → Route 53 decides where to send them.
2. Dynamic content → ELB → Web EC2 (auto-scaled) → Internal ELB → App EC2 → RDS database.
3. Static content → CloudFront → cached copy from nearest edge location.
4. If AZ `us-east-1a` fails → ELB automatically sends traffic to `us-east-1b` instances.
5. If entire `us-east-1` fails → restore system in `us-east-2` from S3 cross-region backups.