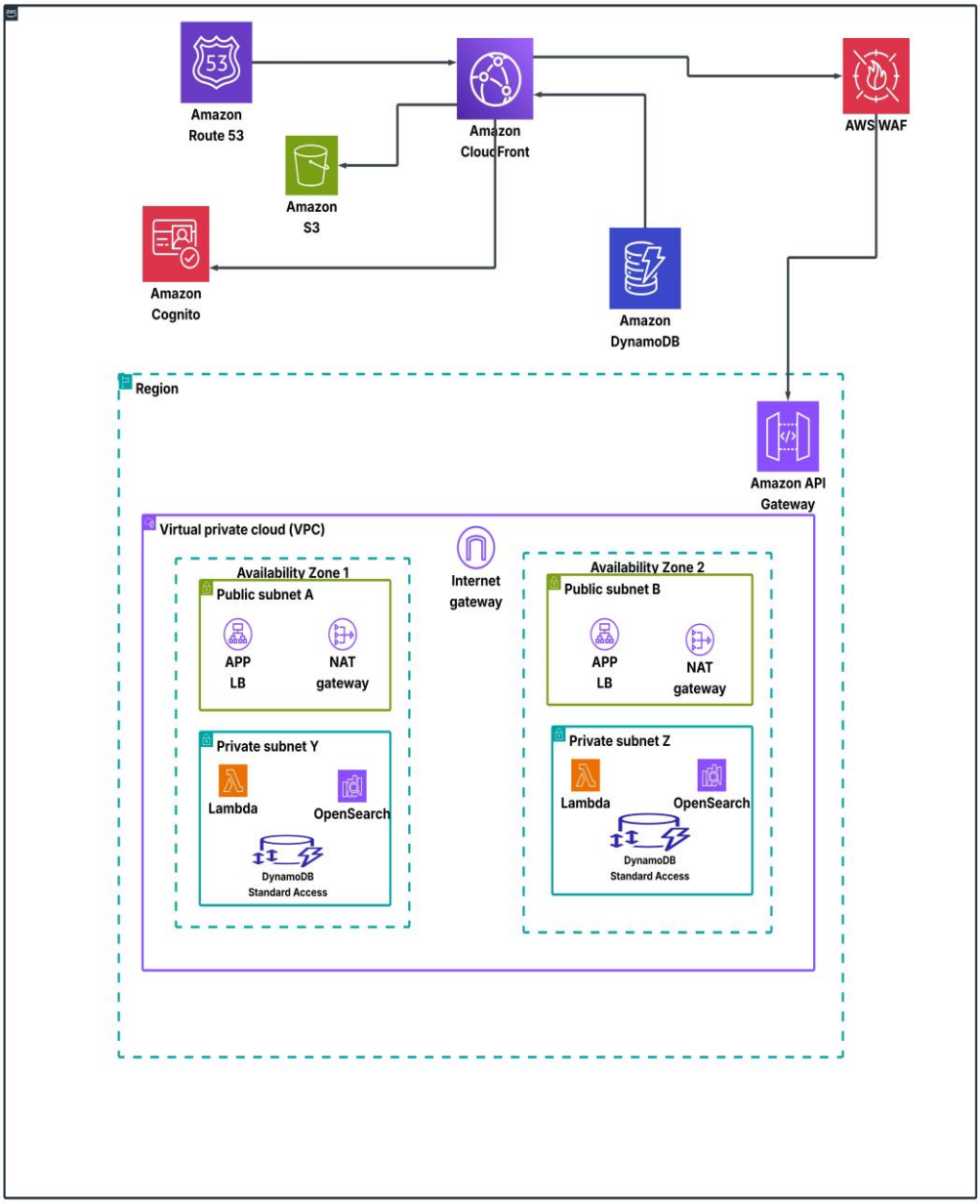
3D Sports Warehouse Platform Architecture on AWS



This design proposes a serverless, highly available, and cost-efficient 3D Sports Warehouse Platform architecture on AWS that enables customers to interact with 3D models of sporting goods (equipment, apparel, footwear) before purchase, while also supporting warehouse operations such as inventory management, picking, and returns.

The architecture leverages Amazon CloudFront for global content delivery, AWS Lambda for backend compute, Amazon DynamoDB for low-latency product and user data, and Amazon OpenSearch Service for catalog search. User authentication is handled by Amazon Cognito, and warehouse operations integrate directly with backend APIs for order fulfillment.

Core Services and Why They Were Chosen

- **Amazon VPC (Networking & Security):** Provides secure, isolated networking across multiple Availability Zones. Public subnets host load balancers and API endpoints, while private subnets handle Lambda functions, DynamoDB, and OpenSearch.
- **Elastic Load Balancer (App LB):** Balances traffic between API Gateway integrations and backend services for warehouse management and fulfillment.
- **Amazon CloudFront (CDN):** Delivers 3D assets (e.g., .glb/.gltf files of sports gear) and media globally with low latency, enhancing user experience.
- **Amazon S3 (Static Content & Assets):** Stores website files, 3D product models, and promotional images/videos with high durability and cost efficiency.
- **Amazon Route 53:** Provides DNS routing and domain management; integrates with ACM for HTTPS/TLS.
- **Amazon API Gateway:** Exposes secure REST APIs for customer and warehouse operations (e.g., product catalog, shopping cart, order placement, returns).
- **AWS Lambda (Backend Compute):** Runs business logic server Lessly:
    - Customer-facing: product details, pricing, orders.
    - Warehouse-facing: stock updates, picking, return authorization.
- **Amazon DynamoDB:** Serves as the NoSQL datastore for product catalogs, inventory tracking, customer accounts, and shopping sessions. Standard Access mode balances cost and throughput.
- **Amazon OpenSearch Service:** Provides search and filtering capabilities for large catalogs (e.g., filtering shoes by size, rackets by grip, jerseys by team).
- **Amazon Cognito:** Manages secure user authentication for customers, staff, and warehouse operators.
- **AWS WAF:** Protects the API Gateway and CloudFront from web attacks (e.g., SQL injection, DDoS).

- **Amazon CloudWatch & AWS X-Ray:** Provide observability through logging, monitoring, and distributed tracing.
- **IAM & KMS:** Enforce fine-grained access control and encryption for sensitive data.

How the Architecture Meets the 5 Requirements

1. **High Availability**
    - Multi-AZ VPC deployment ensures resilience.
    - CloudFront caches assets at global edge locations.
    - Route 53 health checks enable failover for critical endpoints.
2. **Scalability**
    - Serverless Lambda + DynamoDB auto scale automatically during seasonal sales (e.g., Black Friday for sports gear).
    - OpenSearch scales horizontally for high-volume catalog queries.
    - API Gateway throttling prevents overload from flash sales.
3. **Performance**
    - CloudFront caches 3D assets and media near users for sub-second load times.
    - DynamoDB provides millisecond-level reads/writes for product lookups and inventory updates.
    - Provisioned concurrency on Lambda ensures low latency for cart and checkout operations.
4. **Security**
    - Cognito manages secure user access for customers and warehouse staff.
    - IAM enforces least-privilege policies; KMS encrypts data at rest in DynamoDB and S3.
    - WAF and Shield protect APIs and assets against malicious traffic.
5. **Cost Optimization**
    - Serverless pay-per-use model (Lambda, API Gateway) eliminates idle resource costs.
    - DynamoDB Standard Access mode optimizes storage/throughput balance.
    - S3 lifecycle policies and CloudFront caching reduce storage and bandwidth costs.

Design Trade-offs and Challenges

- **Lambda cold starts:** May add delay for critical APIs (checkout, warehouse updates). Mitigation: provisioned concurrency for hot paths.
- **Large 3D sports models:** Items like treadmills or bikes can be heavy files. Solution: use Level of Detail (LOD), progressive loading, or glTF streaming.
- **Inventory consistency:** DynamoDB's eventual consistency could risk overselling popular gear. Solution: use conditional writes or DynamoDB transactions.
- **Complex warehouse workflows:** Bulk updates, returns, or logistics integrations may require ECS/Fargate or AWS Batch for long-running processes.
- **NAT Gateways:** Since they charge per GB processed. If private subnet workloads are heavy, VPC endpoints will be used reduce data transfer costs.

References

Amazon Web Services (2025). Amazon Web Services (AWS) - Cloud Computing Services. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/.

AppMySite (2022). How to create an ecommerce shopping app like Amazon? - AppMySite. [online] AppMySite. Available at: https://www.appmysite.com/blog/how-to-create-an-ecommerce-app-like-amazon/.

Lucid Software Inc., 2025. Lucidchart. [Computer program] Available at: https://www.lucidchart.com [Accessed 10 September 2025].