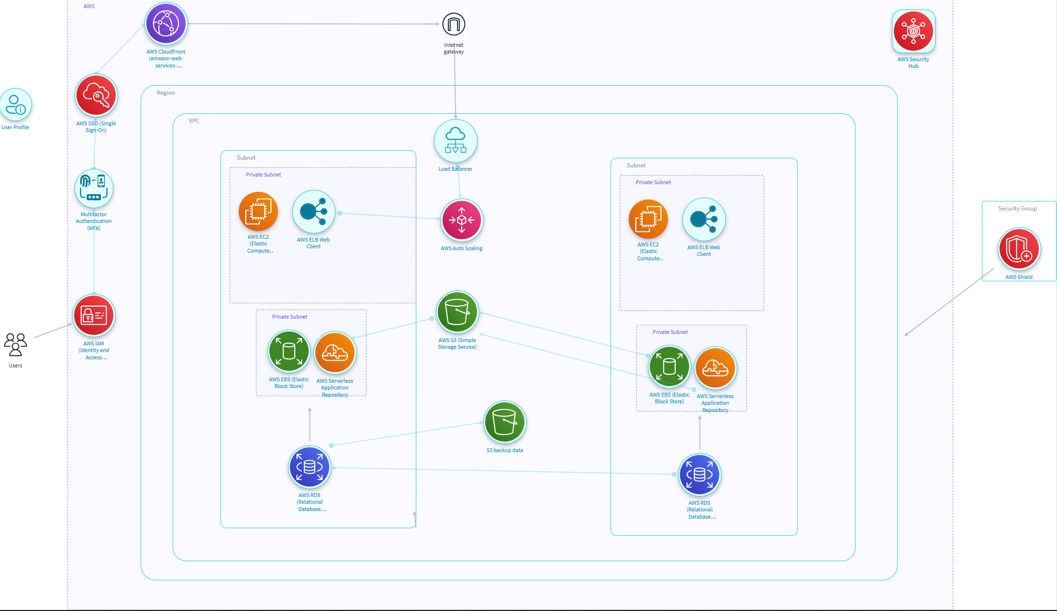
FINAL REPORT ON ARCHITECTURE

CLOUD SECURITY

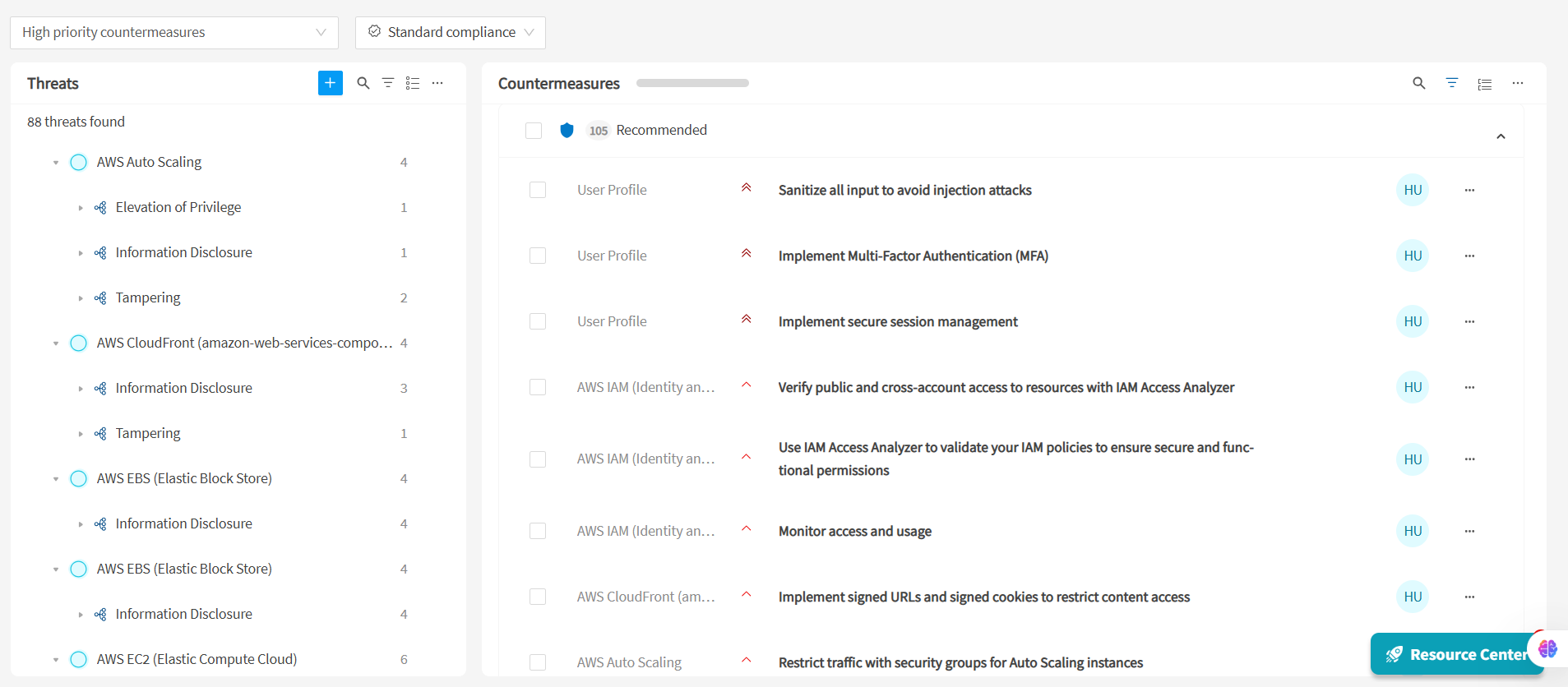
Author: K21-3559 Huzaifa

Created an ecommerce store system architecture using draw.io, added AWS services that will be used in the backend and how the whole application will be hosted on AWS with all the functionalities handled on AWS.

Initial Architecture design before threat model vulnerabilities highlighted



Threats highlighted by the **IrusRisk** Threat Modeler in our architecture diagram



**Threat Modeling Report - E-Commerce Cloud Architecture**

**Spoofing (Identity Impersonation):**

1. IAM Roles & Users: Attackers may attempt to spoof AWS IAM credentials to gain access to AWS services.
2. Web Server (Public Subnet): Risk of spoofed IP addresses or header injection to bypass access controls.
3. Admin/Customer Authentication: Brute-force or credential stuffing attacks to impersonate users or admins.
4. Mitigations: Enforce MFA, strong passwords, use WAF, apply rate limiting, use AWS Cognito.

**Tampering (Data or Resource Modification):**

1. S3 Buckets: Improperly configured buckets may allow unauthorized file modifications.
2. Application Servers: Attackers could inject malicious code (e.g., via SQLi, XSS) to tamper with data.
3. CI/CD Pipeline: Malicious actor may inject vulnerable code if the pipeline is exposed.
4. Mitigations: Use S3 bucket policies and encryption, sanitize inputs, secure CI/CD with least privilege.

**Repudiation (Denial of Actions):**

1. Application Logs: Attackers may perform actions and deny responsibility due to lack of auditing.
2. Load Balancer / API Gateway: Without request-level logging, hard to trace malicious activity.
3. Mitigations: Enable CloudTrail, VPC Flow Logs, CloudWatch, AWS Config.

**Information Disclosure (Data Leak):**

1. S3: Public access misconfiguration may expose private data.
2. RDS: Sensitive user data may leak due to SQLi or misconfiguration.
3. TLS/SSL: Without TLS, sensitive data in transit can be sniffed.
4. Mitigations: Use S3 encryption and ACLs, encrypt RDS, enforce HTTPS.

**Denial of Service (DoS):**

1. Load Balancer / Web Tier: Subject to traffic spikes or malicious DoS attacks.
2. NAT Gateway / EIP: Resource exhaustion can lead to service outage.
3. RDS: Heavy query load may exhaust resources.
4. Mitigations: Use AWS Shield, auto-scaling, rate limiting, RDS monitoring.

**Elevation of Privilege (Unauthorized Access):**

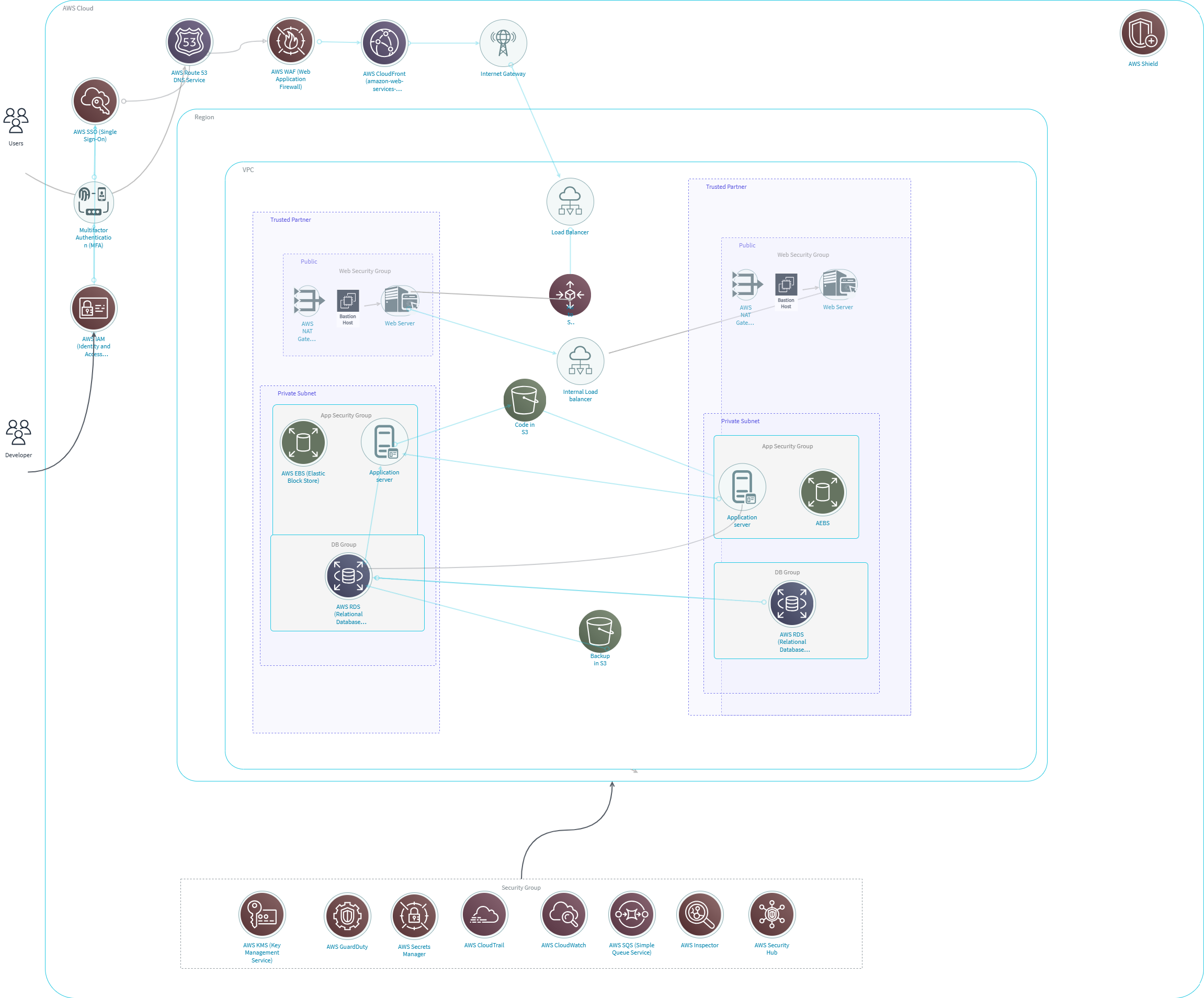
1. Application Layer / IAM Roles: Exploiting vulnerabilities to gain unauthorized access to resources.
2. EC2 / Lambda: Improper IAM role assignments can escalate privileges.
3. Mitigations: Implement least privilege IAM policies, use VPC Endpoint policies, monitor with GuardDuty.

**Additional Recommendations**

1. Security Groups: Restrict inbound traffic tightly; use different SGs for each tier.
2. WAF ACLs: Set geo-blocking, SQLi/XSS filters, and rate limits.
3. CloudFront: Use signed URLs or cookies for access control if needed.
4. Monitoring & Alerts: Set up CloudWatch alarms for suspicious activity and billing anomalies.

After implementing the necessary changes to the system architecture, we ran it in Iris-Risk once again after modifying our design and implementing the recommendations suggested by the threat modeler

**Updated design:**

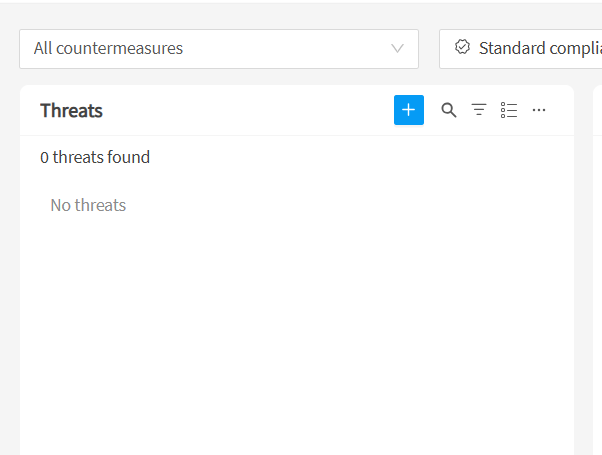


Changes made to the design:

Added proper security group with relevant security services to look after the entire architecture, data flow made reasonably constant. Proper trust zones created. Enhanced security in the design, created proper segregation between the public subnet applications and the private subnet applications. Proper labeling of the crucial services performed with their actual purpose rather generic names

**Re Running the modified design with threat model:**

Threat model threat detection after implementing the necessary changes recommended by the threat model. It gave 0 vulnerabilities and found no threat in the updated new design



**Why the architecture is best fit for the Ecommerce Store hosted on AWS**

1. **Scalability and High Availability**

Multi-AZ deployment ensures fault tolerance and availability even if one availability zone fails.

Using Auto Scaling Groups and Elastic Load Balancing distributes user traffic efficiently and dynamically scales the backend services.

CloudFront (not visible but typically assumed in e-commerce) ensures faster content delivery globally with CDN caching.

**2. Security-Centric Design**

Public and private subnets separate external-facing components (e.g., web servers) from sensitive backend services (e.g., databases).

Web Application Firewall (AWS WAF) and Security Groups ensure that only legitimate traffic is allowed through while blocking malicious actors and bot traffic.

NACLs and IAM roles/policies are enforceable for fine-grained control of data and resources.

**3. Modular and Microservices-Oriented Setup**

Web Tier (Web Servers in public subnet)

App Tier (Application Servers in private subnet)

Data Tier (RDS database in DB subnet)

promotes modularity, easier debugging, and independent scaling.

**4. Cost Optimization**

Resources like EC2 instances and RDS databases can be optimized with Reserved Instances or Spot Instances, helping lower long-term cost.

S3 for object storage ensures a durable, cost-effective way to store product images, invoices, logs, etc.

**5. Resilient Data Storage and Backup**

Amazon RDS (Multi-AZ) ensures automated failover for the database.

EBS Snapshots and S3 versioning can be used for backup and disaster recovery strategies. Supports encryption at rest and in transit for regulatory compliance and secure transactions.

**Benefits of AWS Services in This Architecture**

| **AWS Service** | **Justification / Benefit** |
| --- | --- |
| **Amazon VPC** | Enables full control over network configuration, including subnets, IP ranges, route tables, and gateways. |
| **EC2 Instances** | Scalable compute resources for the web and application servers. Easy to manage and customize for business logic. |
| **Elastic Load Balancer** | Distributes user traffic for better performance and availability of applications. |
| **Amazon RDS** | Fully managed relational database with automatic backups, failover, and security features (encryption, IAM auth). |
| **Amazon S3** | Scalable storage for static files like images, videos, and documents. Ideal for storing catalogs, receipts, and media. |
| **NAT Gateway** | Allows private instances to access the internet securely for updates without exposing them directly. |
| **Security Groups & NACLs** | Network-level security to control inbound/outbound traffic precisely. |
| **AWS WAF (Web ACL)** | Adds an additional layer of protection against common web exploits like SQL injection, XSS, etc. |
| **IAM** | Manages access to AWS services securely using fine-grained roles and permissions. |
| **CloudTrail and CloudWatch** | Monitor, log, and audit actions, helping with troubleshooting and security compliance. |

**Key Strengths for E-commerce Requirements**

Performance: Fast response with auto-scaling and load balancing.

Security: Strong perimeter defenses and internal segmentation.

Compliance: Meets standards like PCI DSS using AWS’s secure services.

Resilience: Auto-healing and fault-tolerant components ensure 24/7 availability.

Analytics Friendly: Integration with tools like Amazon Athena or Redshift for customer behavior analysis.