CloudWatch Alarm Description, Load Balancer Selection & TLS/Domain Setup Requirement.

For our multi-tier web application, I selected AWS Application Load Balancer (ALB) to provide Layer-7 routing, deep ECS integration, and native HTTPS termination.

To ensure full observability and rapid response to issues, I've integrated ALB metrics and logs with Amazon CloudWatch, enabling real-time monitoring and alerting.

This document outlines the key factors behind my ALB choice, details the steps to associate a custom domain and provision TLS certificates via AWS Certificate Manager (ACM), and describes how to configure CloudWatch alarms to maintain performance and security.

1. **The table below outlines 5 different AWS CloudWatch Alarms, with a brief rationale to why each alarm was chosen.**

Alarm Name	Metric	Threshold	Notification Action	Rationale
High EC2 CPU	CPUUtilization	> 75% for 5 minutes	SNS topic	Sustained high CPU utilization often indicates overloaded instances or runaway processes. Alerting here lets us scale accordingly.
ALB High Latency	TargetResponseTime	> 1 s for 2 minutes	SNS topic	Elevated latency at the load balancer is a leading indicator of backend slowness or resource contention. Notifying on this metric ensures we can investigate

				and restore optimal user experience quickly.
RDS Connection Count	DatabaseConnections	> 80% of max for 5 minutes	SNS topic	Approaching the maximum allowed connections risks rejected requests and service outages. This alarm gives us lead time to tune connection pools or scale the database.
Low Free Storage (RDS)	FreeStorageSpace	< 20 GB	SNS topic	Running out of storage on RDS can halt writes and corrupt data. Early warning enables us to scale accordingly or clean up logs and other debugging options before critical failures occur.
ECS Task Count Drift	RunningTaskCount vs DesiredTaskCount	mismatch for 2 evaluation periods	SNS topic	A discrepancy between desired and running tasks signals failed deployments or unhealthy tasks. Alerting here helps trigger automatic

				remediation or manual intervention to maintain service availability.

2. **Load Balancer Choice:** ***Application Load Balancer (ALB)***

   Why Application Load Balancer (ALB)?

- **Layer-7 Routing**
  - Supports host- and path-based rules, enabling blue/green or canary deployments and microservices traffic steering.
- **Seamless ECS/Fargate Integration**
  - Automatically registers/deregisters Fargate tasks in target groups, simplifying service discovery and scaling.
- **Advanced Protocol Support**
  - Native support for HTTP/2 and WebSockets enhances modern web and real-time use cases.
- **Cost-Effective for HTTP Workloads**
  - Per-request pricing often yields lower costs compared to always-on Classic Load Balancer or Network Load Balancer (NLB).
- **Security & Extensibility**
  - Works with AWS WAF for application-layer threat protection.
  - TLS offload at the edge reduces CPU load on backend tasks.
- **Health Checks & Metrics**
  - Fine-grained HTTP(S) health checks ensure only healthy targets receive traffic.
  - Built-in CloudWatch metrics for latency, request counts, and HTTP codes.

Given our need for HTTP(S) routing, ECS integration, and WAF compatibility, ALB is the optimal choice.

3. **TLS / Domain Setup Requirements**

Domain Registration & DNS Configuration

1. **Register or Purchase Domain**
    - Via Route 53 or external registrar (e.g., GoDaddy).
2. **Create a Public Hosted Zone**
    - In Route 53, create a hosted zone matching your domain (e.g., Incode.com).
3. **Delegate Name Servers**
    - Ensure registrar's name servers point to the four NS records provided by Route 53.

Certificate Provisioning (ACM)

1. **Request for Certificate**
2. **DNS Validation**

- ACM returns one or more CNAME records.
- In Route 53, create the validation CNAME(s).
- Wait ~10 minutes for ACM to issue the certificate.

Note: You can use wildcard (*.Incode.com) to cover subdomains with a single certificate.

Attaching Certificate to ALB

1. Provision (or import) a valid ACM certificate and reference its ARN in an HTTPS listener.
2. Define an aws_lb_listener on port 443 with protocol = "HTTPS" and certificate_arn set to the ACM cert.
3. Configure the listener's default action to forward decrypted traffic to the appropriate target group.

HTTP → HTTPS Redirect

1. Create a second aws_lb_listener on port 80 with protocol = "HTTP".
2. Set its default_action to a redirect block targeting port 443, protocol HTTPS, and status code HTTP_301.

3. This ensures all incoming HTTP requests are automatically and securely rerouted to HTTPS.

Ongoing Certificate Renewal

1. ACM automatically renews certificates validated via DNS.
2. Route 53 CNAME records remain valid indefinitely—no manual intervention required.

SUMMARY:

- **Scalability**: Auto-scales to meet traffic demands without manual adjustment.
- **Security**: Centralized, managed TLS offload and WAF integration.
- **Observability**: Fine-grained metrics & logging via CloudWatch.
- **Maintainability**: Declarative Terraform configuration for repeatable deployments.