Exercise 7 -SWED

1.Compare Mainframe and Cloud Computing Models

Mainframe Computing:

Centralized model.

High reliability, availability, and security.

Expensive infrastructure & operation.

Limited scalability

Users access via terminals (thin clients).

Example: Banking systems

Cloud Computing:

Decentralized & distributed model.

On-demand resources via the internet.

Highly scalable

Used in modern web, mobile apps, IoT, and big data.

Why the Change Over Time?

Cost-efficiency: Cloud reduces CAPEX, only OPEX.

Scalability: Cloud allows dynamic scaling, ideal for fluctuating workloads.

Global Reach: Cloud services offer low-latency access worldwide.

Agility: Faster time-to-market and ease of deployment.

Maintenance: No hardware management by developers.

Innovation: New cloud-native services (AI, ML, serverless) not possible on mainframes.

2. Your Web Monitoring App (Exercise 6)

Features:

Monolithic architecture

Local database or single cloud instance

Limited fault tolerance and scalability

Basic or no real-time monitoring/alerting

Mostly synchronous, tightly coupled components

Cloud-native Architecture:

Features:

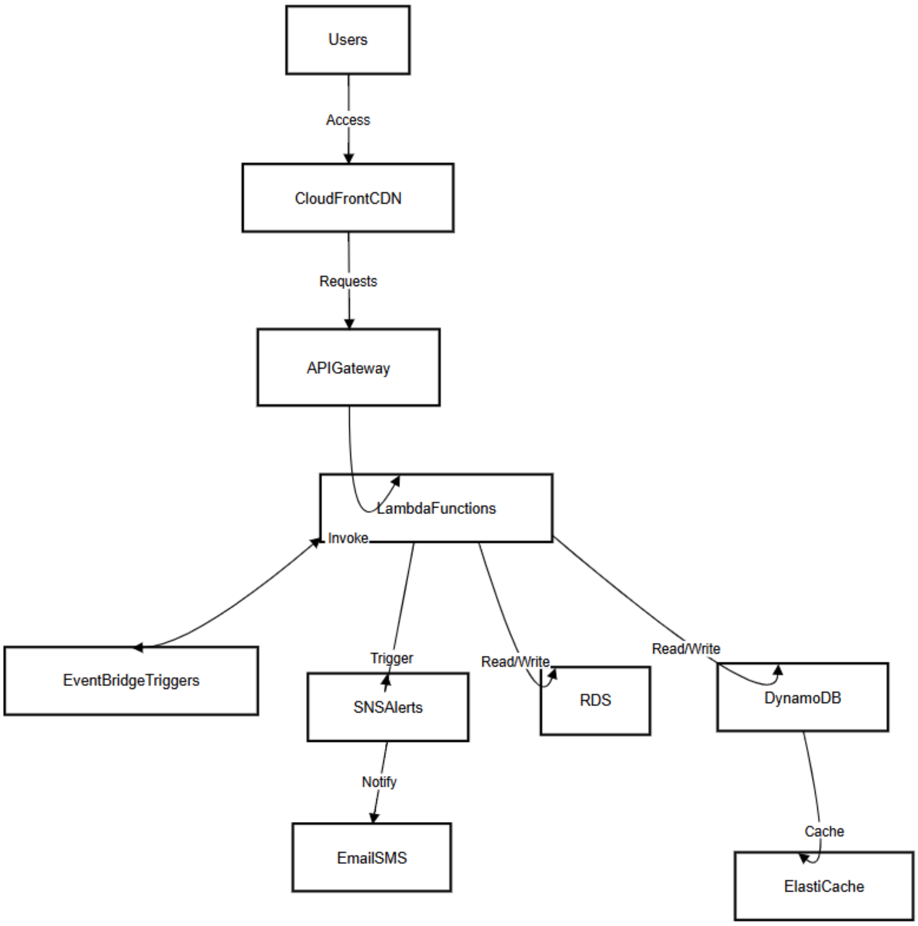
Scalability: Autoscaling groups

Architecture style : Microservices

Deployment : CI/CD with pipelines, containers

Availability:load balanced

3.



Users → CloudFront CDN
Users load the dashboard

CloudFront → API Gateway
Dynamic data requests (e.g. fetch latest checks) go through API Gateway.

API Gateway → Lambda
Lambda functions handle API calls (read/write config or results).

EventBridge → Lambda
Scheduled rules invoke the same (or a dedicated) Lambda every interval to run the site-check logic.

Lambda ↔ RDS / DynamoDB
Lambdas read endpoint configurations and write each check’s status and latency to the databases.

DynamoDB → ElastiCache
Recent results are cached for fast dashboard reads.

Lambda → SNS → Email/SMS
On failure or threshold breach, Lambda publishes to SNS, which pushes alerts to admins.

CloudWatch (Implicit)
All executions and metrics flow into CloudWatch for logging and dashboards.

4.

The halting problem

This is a classic problem in computer science that asks whether a given program will eventually halt or run forever. Alan Turing proved that there is no algorithm that can solve this problem for all possible inputs.