# CE308 – Cloud Computing


## SafeZoneSentient – A Cloud-Based War Crisis Management System

**Submitted by**:

Saad – 2022509 Aiza – 2022077

Ahmed – 2022054 Mustafa – 2022407

**Submitted to**: Ma'am Safia Baloch



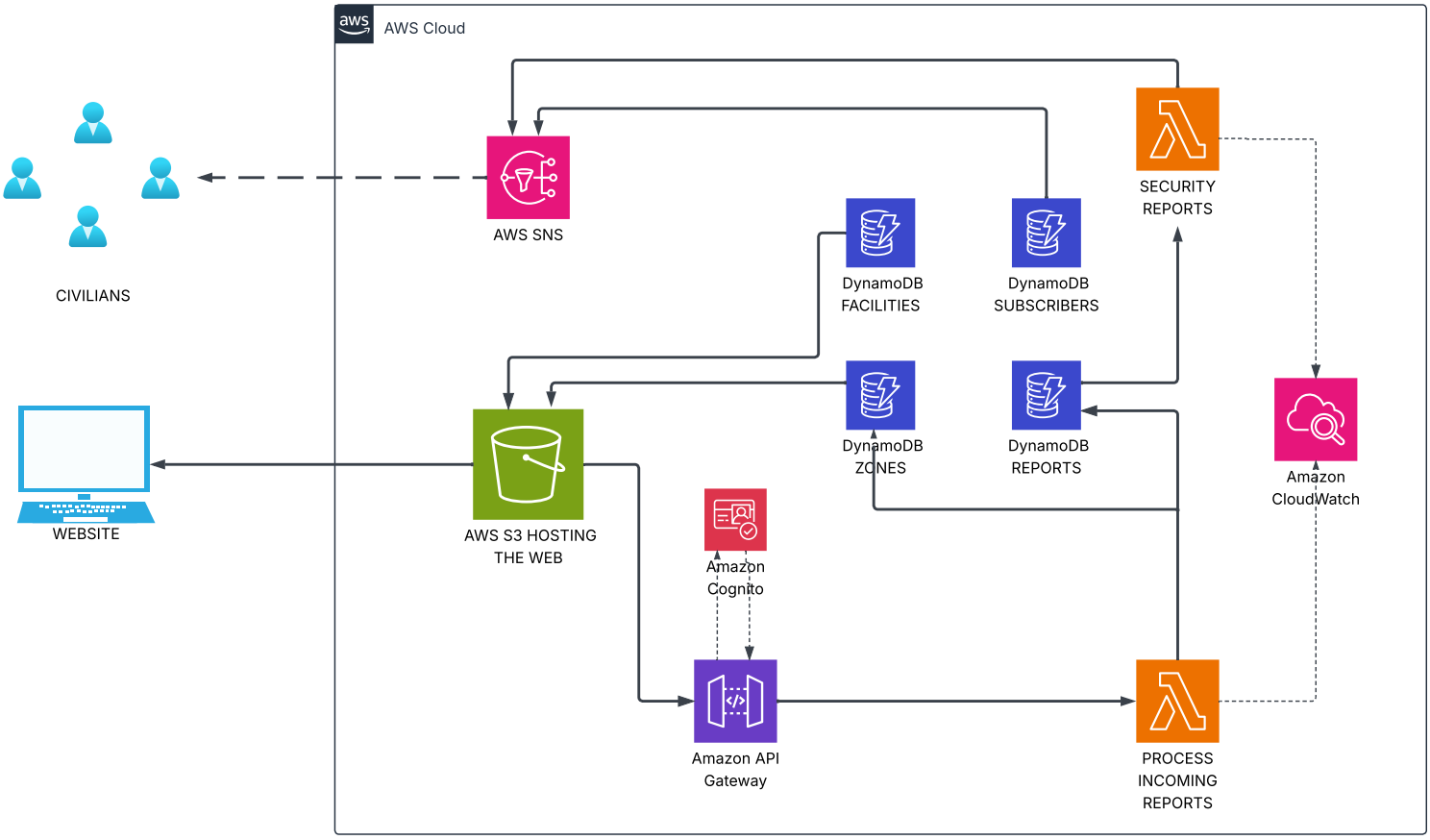

# SafeZoneSentient – A Cloud-Based War Crisis Management System - Report

## 1. Introduction

In high-stakes scenarios like war zones or large-scale disasters, access to real-time, **trustworthy information** can be the difference between safety and chaos. **SafeZoneSentient** is a **cloud-native crisis management system** built for such times — offering a public-facing platform to disseminate **verified alerts**, facilitate **civilian reporting**, and assist with **situation awareness** through map-based visualizations and real-time notifications.

Built entirely on **Amazon Web Services (AWS)**, the solution is highly available, **fully serverless,** and designed with **event-driven principles**, making it resilient under unpredictable, large-scale usage.

### WORKFLOW DIAGRAM



# 2. Use Case Summary

**SafeZoneSentient** was designed under a hypothetical collaboration with **ISPR Pakistan** to act as a digital guardian for civilians. The platform serves the following core roles:

- **Admins (ISPR)** can:
  - Post verified alerts
  - Update zone data
  - Trigger mass alerts
- **Civilians** can:
  - View nearby safe zones, shelters, or danger zones
  - Submit danger or help reports
  - Receive email alerts if they're subscribed
- **All users** interact through:
  - A **real-time, interactive web map**
  - **Form-based reporting**
  - **Email alerts** triggered by new events

# 3. Cloud-Native Architecture Overview

The platform is composed of several AWS services working in synergy to support live interactivity, security, and scalability:

### A. Amazon S3 – Static Website Hosting 

- Hosts the **React-based frontend**
- Provides **HTTPS-based secure content delivery**
- Public-read bucket with **cloudfront** support
- **Cross-region replication** for geo-resilience

### B. React Frontend & UX 


The frontend is built with:

- **React.js** for dynamic UI rendering

- **Leaflet.js** for interactive maps
- **Axios** for API integration
- **Tailwind CSS** for responsive styling

**Key Frontend Features**:

- Interactive map filters
- Civilian reporting forms
- Admin dashboard with alert push controls
- Real-time zone status visualization

### C. Amazon API Gateway – Backend Entry Point


- RESTful APIs for:
  - /alerts, /reports, /zones, /shelters
- Supports:
  - POST from civilians (reports)
  - PUT/POST from admins (alerts)
  - GET for real-time frontend data fetching
- Fully secured with **CORS** and throttling
- Directly integrated with Lambda and DynamoDB

### D. Amazon DynamoDB – NoSQL Storage Layer


- Tables:
  - Reports
  - Facilites
  - Zones
  - Shelters
- Key benefits:
  - High throughput with auto-scaling
  - Stream-based event triggering
  - IAM policy-controlled access
- Used in a **table-per-feature** pattern for modularity

## E. AWS Lambda – Backend Logic Layer 

Two crucial Lambda functions power the core logic:

### 1. Lambda: ProcessingIncomeReports

- **Trigger**: API Gateway
- **Function**:
  - Validates and processes incoming civilian reports
  - Inserts them into the Reports table in DynamoDB

### 2. Lambda: ProcessingSecurityReports

- **Trigger**: DynamoDB Stream
- **Function**:
  - Detects new reports
  - Classifies them based on priority
  - Publishes alerts to SNS if critical

## F. Amazon SNS – Notification and Alert System 

- SNS topic war-zone-alerts
- Subscribers: Instructor and all class members
- Receives alerts via Lambda and sends:
  - **Email notifications** for verified crisis reports
  - **Instant delivery** through event-driven design

## G. Cross-Region & Disaster Recovery

- **S3 buckets** use cross-region replication
- **DynamoDB** supports multi-region tables (optionally enabled)
- Entire infrastructure is region-agnostic
- Ensures uptime and continuity during localized outages

# 4. Frontend UX Features

### Interactive Map (Leaflet.js)

- Real-time visualization of:
  - Red Zones
  - Safe Zones
  - Relief Points (Food, Water, NGOs)

### Civilian Reporting

- Submit:
  - Location-based reports
  - Descriptions of threats or needs
- Processed via Lambda → DynamoDB → SNS (if critical)

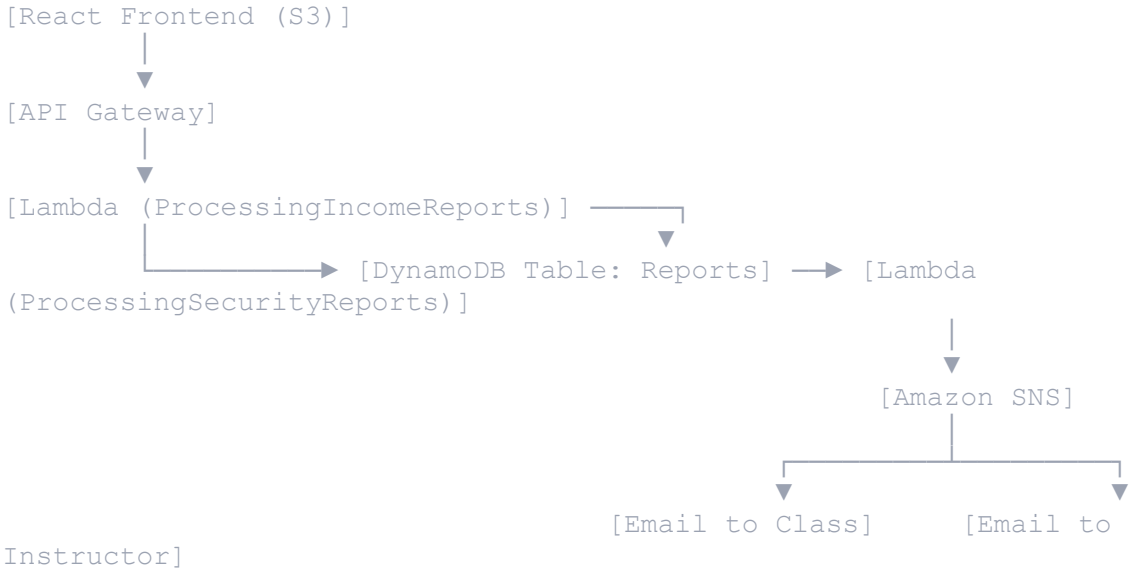
### Admin Panel

- Login-only access (future: Cognito auth)
- Add/update:
  - Zone colors
  - Alert messages
- Trigger SNS alerts manually or automatically

### Email Alerts

- Subscribed users get:
  - Verified ISPR warnings
  - Real-time updates if critical events occur nearby

# 5. End-to-End Architecture Diagram FOR REPORTING




# 6. Cloud-Native Design Benefits

| Capability | Implementation |
|---|---|
| Serverless Computing | Lambda, API Gateway, DynamoDB |
| Event-Driven Architecture | DynamoDB Stream → Lambda → SNS |
| Auto Scalability | Lambda, S3, DynamoDB auto-scale on-demand |
| High Availability | AWS multi-AZ managed services |
| Real-Time Communication | SNS email alerts |
| Zero Infrastructure Ops | No EC2, full serverless |
| Global Reach | S3 & DynamoDB region-agnostic + replication |
| Cost-Efficient | Pay-per-use Lambda, S3, API Gateway |
| Secure and Role-Based Access | IAM + HTTPS + CORS |

# 7. Future Scope & Enhancements

**In Development**

- Mobile app (Flutter / React Native)

- Push notifications (Firebase Cloud Messaging or SNS mobile)
- Real-time chat with responders (using WebSockets)

**Planned Additions**

- Amazon SageMaker for AI-powered report classification
- Offline mode with PWA caching
- User authentication via Amazon Cognito
- Localization support (Urdu, Pashto, Sindhi)

# 8. Conclusion

**SafeZoneSentient** is more than a project — it's a cloud-native blueprint for public safety. It demonstrates how **modern cloud platforms like AWS** can build life-saving systems that scale, adapt, and communicate in real-time. From serverless APIs and database triggers to resilient notifications and map-based data visualization, every part is built with a mission: **deliver verified crisis information quickly, securely, and scalably**.

This platform could serve as the basis for future national-level disaster management systems or NGO-driven emergency response tools.