SJSU SAN JOSÉ STATE UNIVERSITY

Contents

Purpose of Learning 2

Hackathon Context 3

Tier-0 Application Overview 3

Device Data Simulation 3

User Data Simulation 4

Site Camera Data 4

Device Diagnostics and Log Analysis 5

SJSU SAN JOSÉ STATE UNIVERSITY

Enterprise Distributed Systems COE - SRE AI & Agentic Hackathon

This project demonstrates Tier-0 Enterprise Reliability Engineering principles through a simulated global application environment.
After completing the Enterprise Software Technologies Program at San José State University, you joined a Fortune 500 organization operating mission-critical applications that demand 99.99999% (seven-nines) availability.

The initiative centers on designing, deploying, and validating a Tier-0-class application that integrates real-time IoT telemetry, active user monitoring, and AI-driven visual intelligence across geographically distributed sites.
Through the simulation, you will apply high-availability architecture, message-driven data ingestion, and cognitive analytics to ensure operational continuity and reliability at enterprise scale.

Purpose of Learning

The purpose of this exercise is to **translate theoretical reliability and distributed-systems knowledge into practical implementation**.
By completing this scenario, learners will:

1. **Understand Tier-0 system requirements**—including uptime expectations, SLA design, and failure-domain isolation.
2. **Develop message-based data pipelines** using MQTT and message queues for scalable ingestion.
3. **Apply AI/ML techniques** such as image embeddings and semantic search for contextual site analysis.
4. **Integrate multiple data domains** (IoT telemetry, user sessions, and site imagery) into a unified observability framework.
5. **Simulate reliability tests** and evaluate system performance against HA and DR objectives.

Hackathon Context

Tier-0 Application Overview

The assigned application supports **real-time user access** through a centralized **Redis cache datastore**, maintaining continuous telemetry and user session state. IoT devices—such as **gas turbines, drilling bits, and industrial sensors**—transmit field data via **MQTT topics**, ensuring seamless communication across distributed sites.

- **SLA Target:** 99.99999% availability
- **Core Components:**
    - Front End: Web API & interface
    - Middleware: FastAPI orchestration layer
    - Backend: Redis cache acting as proxy to persistent database
    - Database: SQL/NoSQL datastore

Before executive rollout, the deployment must undergo **reliability and availability simulation tests** to validate system resilience.

Device Data Simulation

Oil & Gas MQTT Simulation: Simulate 100,000 field devices publishing telemetry to MQTT topics for 10 global sites. Device categories include:

Turbines

Thermal Engines

Electrical Rotors

Oil & Gas Connected Devices

Topic Convention: og/field/{site_id}/{device_type}/{device_id}

Each message follows a uniform JSON schema for consistent ingestion and monitoring across the enterprise pipeline.

User Data Simulation

The site also simulates **active web-application users**.
Metrics include:

- Number of active users currently logged in
- Number of active backend data connections
- Detailed user session lists with timestamps and regions

All user-activity data is streamed through a **message queue** (e.g., RabbitMQ) to enable **real-time analytics and performance monitoring**.

Site Camera Data

Each site contains **security cameras** that periodically capture and upload images.
At the **SRE layer**, images are processed using **Cohere embedding models**, converting visual data into **vector embeddings** for:

- **Image classification**
- **Semantic description**
- **Anomaly detection**
- **Contextual search and monitoring**

Images and embeddings can be stored in **MongoDB** (NoSQL) or **Redis** (in-memory cache).

When users click **"Describe Image Context"**, the system generates a **word cloud** summarizing image content on the site.

Device Diagnostics and Log Analysis

Device and system log data are available to the **SRE team** for diagnostics.
Using **Retrieval-Augmented Generation (RAG)**, **code parsing**, or **AI-assisted distributed-system analysis**, you must identify which **IP addresses generate the most requests or errors** within system logs.

Example **Natural-Language Queries:**

- "Give me turbine sites that have workers **without hats**."
- "Give me turbine sites that have workers **with high safety compliance**."
- "Get sites where an **engineer has a hard hat and tablet in hand**."

Prompts for deeper insights:

- "How many safety incidents occurred in BP operations in 2024?"
- "Describe BP Oil Drill Operations and Hard Hat requirements."
- "List the most frequent IP devices generating **Error 400**."