

Test Project

Cloud Computing

Independent Test Project Designers: Shawn Xu, Xiaowei Wang, Qiang Wang, Xu Chen
Independent Test Project Validator: Chit Boon Lee, SCM

Contents

Task Description	3
Background	3
Initial state	4
Three Stages	4
Stage One: Data Ingestion	4
Stage Two: Data Preprocessing	5
Stage Three: Data Extraction and Storage	5
<i>Tasks</i>	5
Technical details	5
Service details	5
<i>DataBase</i>	6
<i>Applications</i>	6
<i>Reference Document</i>	6

Contents

This Test Project consists of the following documentation/files:

1. WSC2024_TP53_main_document_actual_en
2. **WSC2024_TP53_day1_actual_en**
3. WSC2024_TP53_day2_actual_en
4. WSC2024_TP53_day3_actual_en

Task Description

UnicornTech has identified an anomaly in the data returned by devices deployed in customer environments. These devices are generating non-standard, alphanumeric information codes which cannot be processed by the existing system.

To facilitate comprehensive data analysis and utilization, a cloud-based batch processing solution is required. This system must efficiently ingest, process, and categorize the anomalous data for subsequent analysis and troubleshooting.

Effective resource management, AWS cost optimization, and cloud resource organization are essential. Secure integration with AWS services, robust network traffic monitoring, and performance optimization are critical. Additionally, stringent data security measures, including regulatory compliance, access controls, data loss prevention, and regular backups, must be implemented.

Successful deployment of this system will enhance customer satisfaction and drive product improvement.

Background

UnicornTech, a leading provider of intelligent equipment, is dedicated to delivering high-quality, innovative solutions to its customers. The company's advanced machinery and production lines are deployed across diverse customer environments, generating valuable operational data. In the highly competitive landscape of intelligent equipment, maintaining a competitive edge demands not only superior product offerings but also a profound understanding of customer feedback and equipment performance. By leveraging sophisticated data processing systems, UnicornTech aims to elevate service quality, optimize product performance, and solidify its market position.

UnicornTech has encountered a challenge in processing abnormal data generated by its intelligent equipment. These data sets, composed of non-standard alphanumeric characters, are not effectively handled by the current system. Recognizing the potential insights embedded within these anomalous data, UnicornTech seeks to extract critical information to drive product enhancements and elevate customer satisfaction. To realize this potential, a robust data processing solution is imperative.

The company's strategic objective is to deploy a cloud-based batch data processing system. This system will be engineered to effectively handle the complexities of non-standard data, ensuring rapid and accurate data ingestion, processing, and categorization. By empowering data analysts with the necessary tools, UnicornTech will expedite issue identification and resolution, resulting in enhanced service quality and customer satisfaction.

Aligned with UnicornTech's commitment to continuous improvement and customer-centricity, the successful implementation of this batch data processing system signifies a pivotal advancement. This initiative will not only streamline the management of non-standard data but also bolster the company's broader goals of product excellence

and market leadership. By augmenting its data processing capabilities, UnicornTech is positioned to gain deeper customer insights, refine product offerings, and fortify its industry leadership.

CAUTION

Please use **us-east-1** region.

Initial state

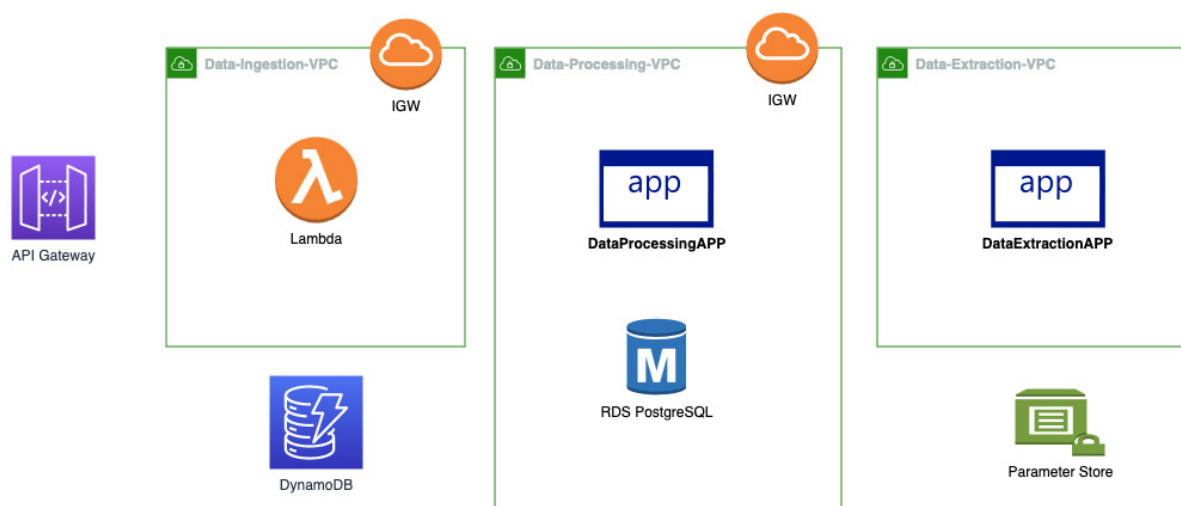
A cloud-based batch processing system must be implemented to efficiently handle the anomalous equipment information codes, incorporating scheduling mechanisms to automate data collection and processing.

An initial, yet incomplete, architectural blueprint has been developed in collaboration with engineering colleagues.

The task involves configuring cloud services and establishing network connectivity to eliminate single points of failure and mitigate potential bandwidth constraints. Comprehensive network traffic monitoring and security measures are essential components of this infrastructure.

This system architecture is fundamental to the efficient processing and secure storage of abnormal equipment information codes for subsequent analysis and utilization.

The architecture diagram is for reference only



Three Stages

Stage One: Data Ingestion

Implement a secure RESTful API to receive and store abnormal information codes. The API must incorporate robust security measures to protect against malicious attacks. Ingested data will be persisted in a DynamoDB table. Ensure the Lambda function handling the API is securely integrated within the Data-Ingestion-VPC.

Stage Two: Data Preprocessing

Establish a data preprocessing service within the Data-Processing-VPC. Utilize the provided binary file (detailed specifications in subsequent documentation) to process abnormal information codes extracted from the DynamoDB table. Processed data will be stored in an RDS database.

Stage Three: Data Extraction and Storage

Configure a data extraction service within the Data-Extraction-VPC. Employ the provided binary file (detailed specifications in subsequent documentation) to process abnormal information codes from the RDS database. Final processed data will be stored in the Parameter Store.

Tasks

1. Access the Cloud Raiser platform and thoroughly review the competition guidelines.
2. Integrate the Lambda function with the Data-Ingestion-VPC.
3. Establish data preprocessing and extraction services within the Data-Processing-VPC and Data-Extraction-VPC respectively.
4. Implement cost-effective strategies to minimize system operational expenses.
5. Prioritize API security to prevent data breaches.
6. Design the architecture to withstand potential Chaos attacks.
7. Earn points based on system performance as outlined in Score Events and Scoreboard.

Technical details

1. The server application is a precompiled binary that must remain unmodified. Using any altered or unofficial binary will result in disqualification.
2. The server application is designed for x86 architecture and optimized for Amazon Linux 2 and specific Docker containers. Consider computing services that minimize infrastructure management.
3. The system will send abnormal equipment information codes to your API every two minutes. Analyze traffic patterns to optimize performance.
4. Successful processing and storage of message codes in the Parameter Store will earn points.
5. Points will be awarded for the creation of expected cloud services.
6. Strict adherence to VPC naming conventions, avoidance of new VPC creation, and prohibition of Internet Gateway modifications are enforced.
7. Manual database or Parameter Store modifications will result in significant point deductions.
8. The use of a single Lambda function bound to the Data-Ingestion-VPC is mandatory.

Service details

IAM ROLES

Preconfigured IAM roles and instance profiles are available for your use.

- game-ec2-profile
- game-ec2-role
- game-cloudwatch-event-role
- game-lambda-role
- game-ecs-role
- game-ecs-execution-role
- game-scheduler-role

- game-codedeploy-role
- game-codepipeline-role

DataBase

1. A highly available DynamoDB table named 'FeedbackTable' must be deployed to store abnormal information codes. The table schema is as follows:

KEY NAME	KEY TYPE	DATA TYPE
id	Partition Key	String
message	Sort Key	String

2. A relational database, compatible with the **PostgreSQL** engine, must be deployed to store preprocessed abnormal messages. Given the anticipated data volume, bi-weekly backups are sufficient. **Serverless databases are prohibited.**

The table name must be 'DataProcessingAPP'. Use the following SQL statement to create the table:

```
CREATE TABLE IF NOT EXISTS DataProcessingAPP (
    id SERIAL PRIMARY KEY,
    message VARCHAR(255)
);
```

Applications

Binary Files and Configuration: The development team has provided pre-compiled binary files, DataProcessingAPP and DataExtractionAPP, along with a configuration file, config.ini. Deploy DataProcessingAPP in the Data-Processing-VPC and DataExtractionAPP in the Data-Extraction-VPC.

Game Assets: Download necessary game assets from the S3 bucket **s3://tp53-day1-game-assets-2024**. Modify the **config.ini** file to run the binary files. Avoid modifying or deleting bucket contents. For redundancy, upload essential files to your competition account's S3 bucket, leveraging its versioning and cost-effective storage capabilities.

CI/CD Implementation: Implementing a CI/CD pipeline with a review process for the Lambda function in the Data-Ingestion-VPC is strongly recommended to streamline updates while ensuring system integrity and security. For efficient and reliable deployments, using CodeDeploy is also highly advised.

Reference Document

- <https://docs.aws.amazon.com/>
- <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>
- <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>