

2024 TSMC IT
CareerHack



Digital Excellence × Generative AI

以生成式 AI 技術實現 IT Infra 自動化監控與管理

C5 我想吃滷豆乾跟sudo

國立成功大學資訊工程學系 大二

李緒成 劉哲佑 李達安 張百鴻



OUTLINE

User Story

System Architecture

Application Implement

Demo

DevOps

Testing

Future



User Story (1/2)



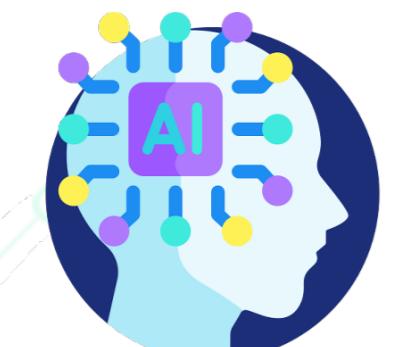
Senior SA

Jason



Junior SA

Henry



AI

Jerry

Jason:

We must closely monitor the consumer, as any errors could result in significant financial losses. We can only assign this task to our Junior SA Henry.

Henry:

It feels like such a waste of my skills, and monitoring around the clock is impossible. Handling these trivial tasks daily is incredibly boring. I'm seriously considering quitting : (

Jerry:

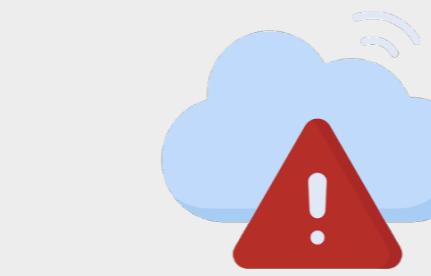
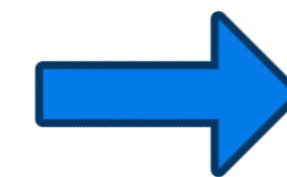
It's my show time!

User Story (2/2)

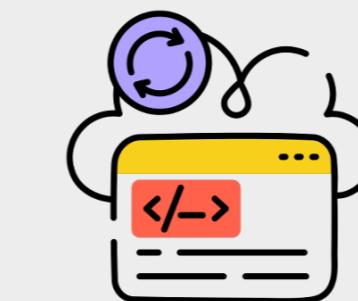


AI Monitor System

- Replace the role of a junior SA
- Analyze Consumer's log 
- Resolve the issue automatically



**System Error
Detected**



**Auto-Scale
Consumer
resources**



Discord Notification

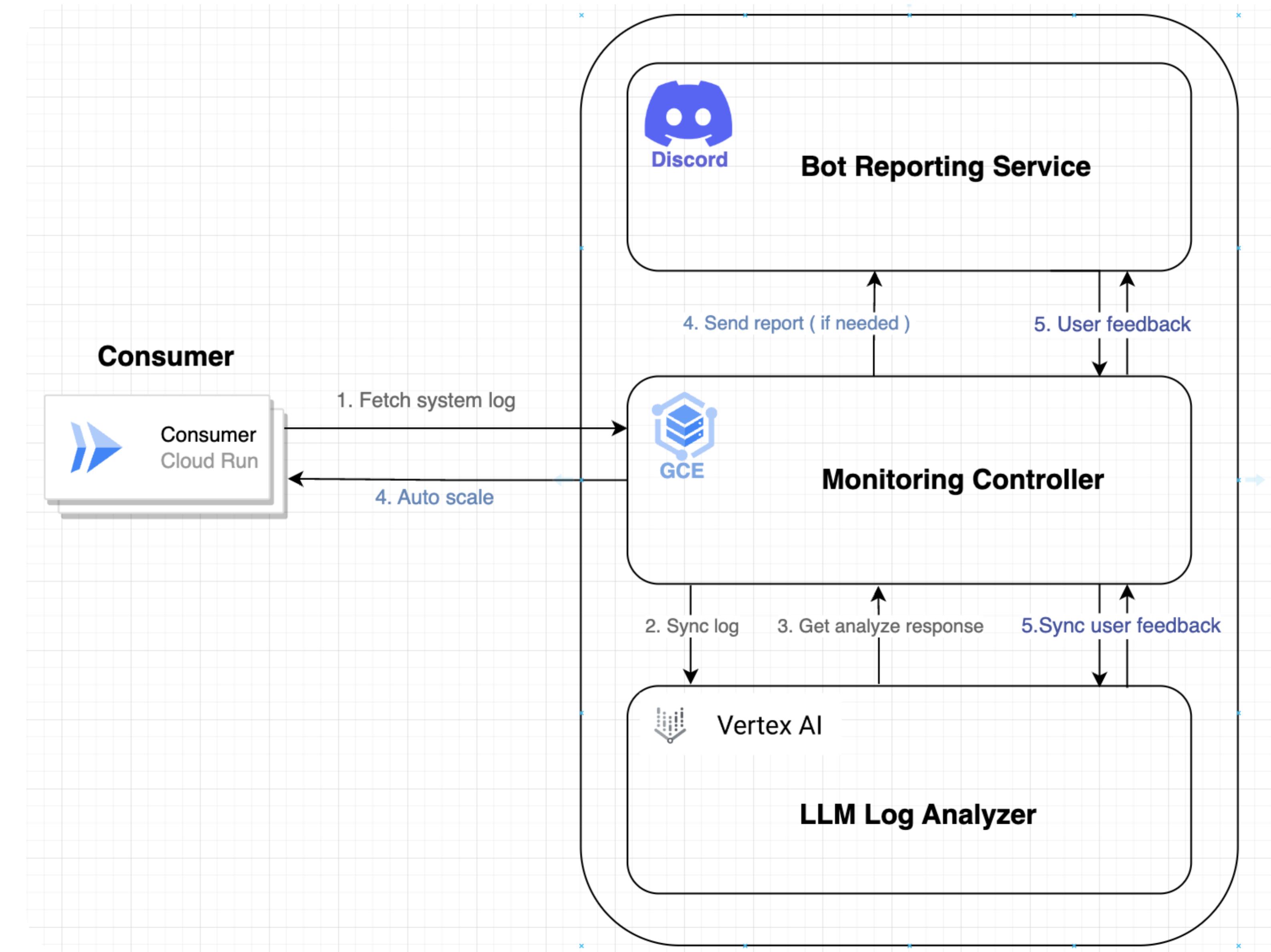
- By Discord bot
- Senior SA can send feedback to help generating better notification mess

OUTLINE

User Story
System Architecture
Application Implement
Demo
DevOps
Testing
Future



System Architecture



OUTLINE

User Story

System Architecture

Application Implement

Demo

DevOps

Testing

Future



Consumer

- stimulate async task
- stimulate error status
- feature metrics :
 - remain_count
 - avg_exe_time

Consumer



Health

GET / Health

Job

GET /api/job/start Start Normal Behavior

GET /api/job/stop Stop Normal Behavior

GET /api/job/status Status Normal Behavior

GET /api/job/sleep/{seconds} Sleep

Full

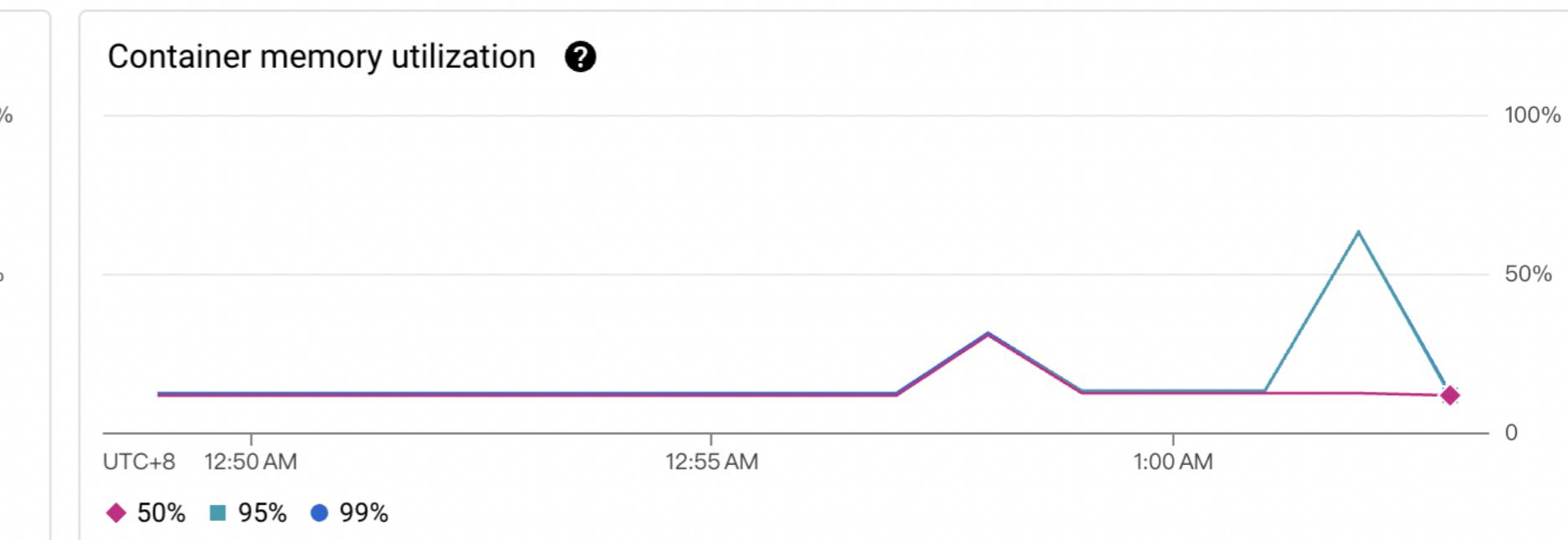
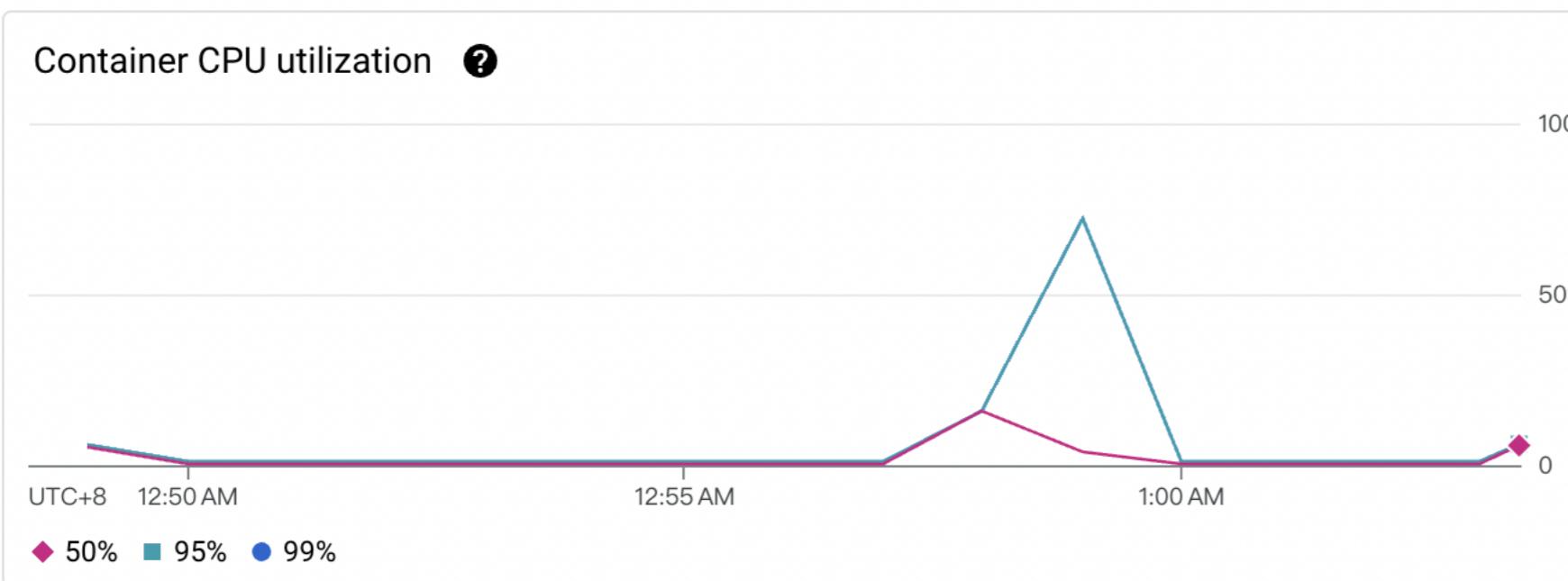
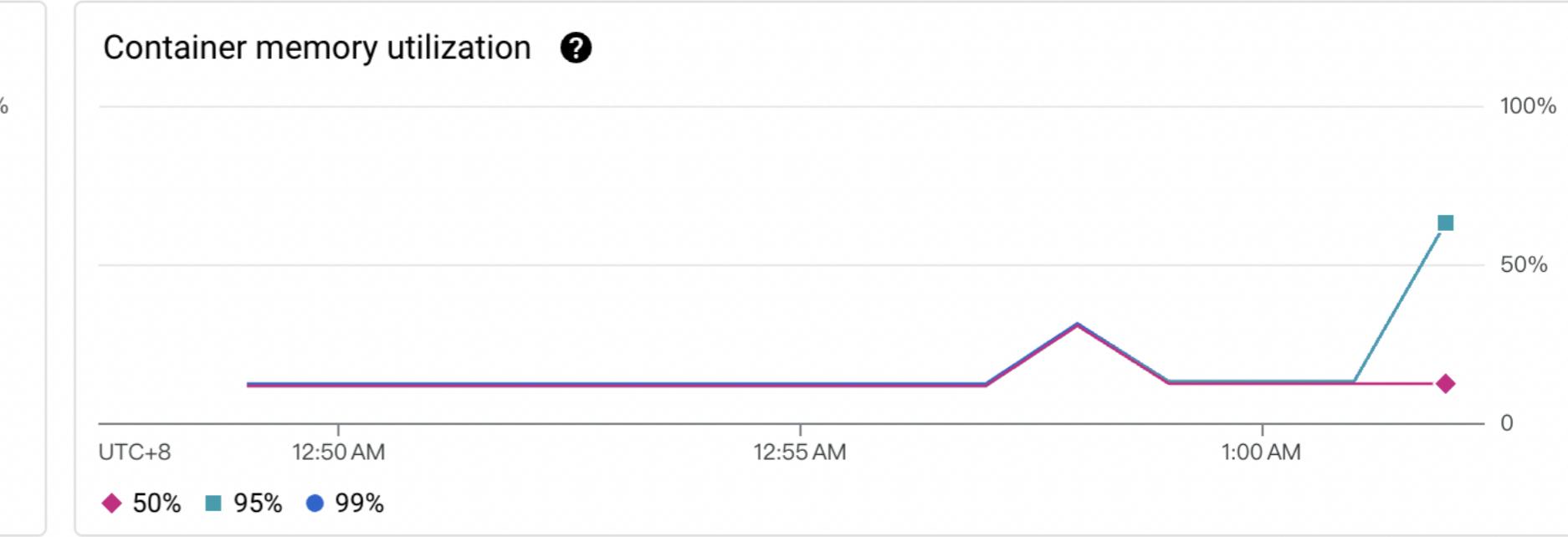
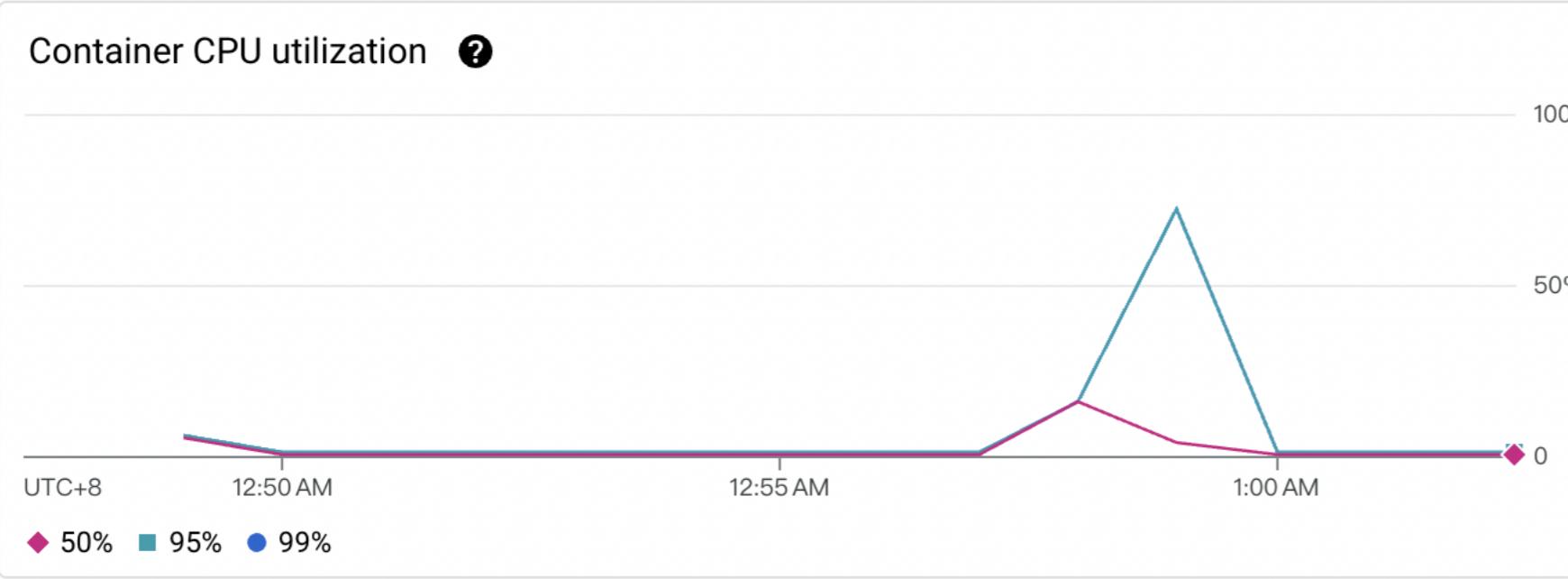
GET /api/full/cpu/duration/{duration} Full Cpu

GET /api/full/ram/duration/{duration} Full Ram

GET /api/full/enque/{num} Enque

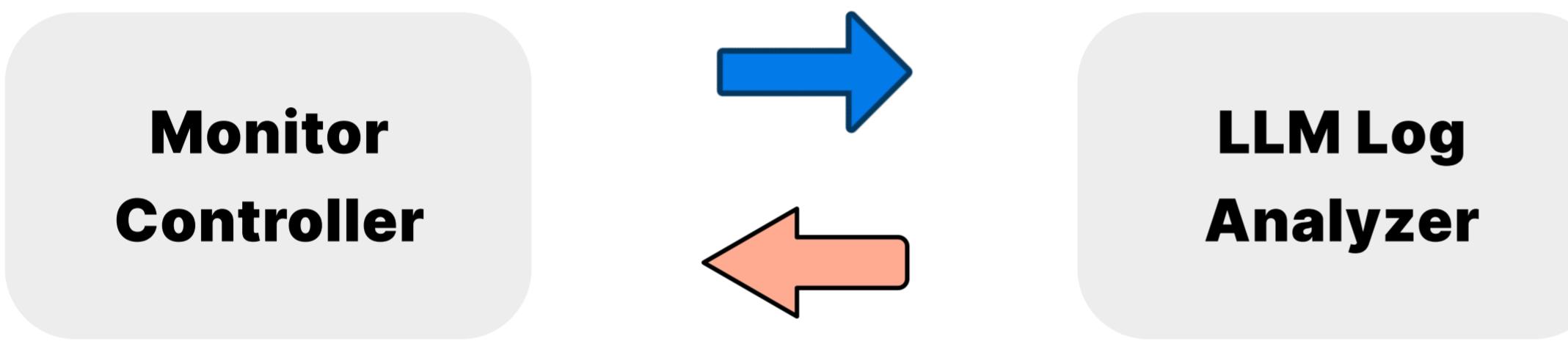
GET /api/full/error Error

Consumer



GenAI - Analyze Flow

1. send log data
to analyzer



2. send feedback
to report and
scale

Problem:

- **AI could give the wrong suggestion**
- **No ways for it to improve**
- **We need a way for AI to learn from experience**

GenAI - Analyze Flow with memory

1. send log data
to analyzer

2. retrieve similar log from past
and its analysis feedback



4. send feedback
to report and
scale

3. Store log data and newly
generated analysis feedback

GenAI - Feedback Response

After monitor controller send report to user(senior SA) ...

Provide feedback



Send Response

Update AI & Human transaction

GenAI - Prompt Engineering

1. Detect error/warning thresholds by rule. (heuristic_analysis)
2. Retrieve similar log and analysis from Pinecone. (memory)
3. Formulate prompt with 1. and 2. and ask GenAI for analysis
4. Parse the analysis result into JSON format for scaling and reporting

The following text contains log data for a Google Cloud Run application. \
This data is presented in CSV format and encompasses the most recent {time_span} minutes:
{log_data}

{memory}

Some heuristic analysis has been performed beforehand, the following text is a summary of the analysis:
{heuristic_analysis}

Your task is to provide an in-depth analysis based on the provided log data and heuristic analysis feedback, \
and scale the application by providing the number of CPUs, amount of memory in MB, and amount of instances to add(positive) or remove(negative).
The system will automatically scale the application based on your feedback.

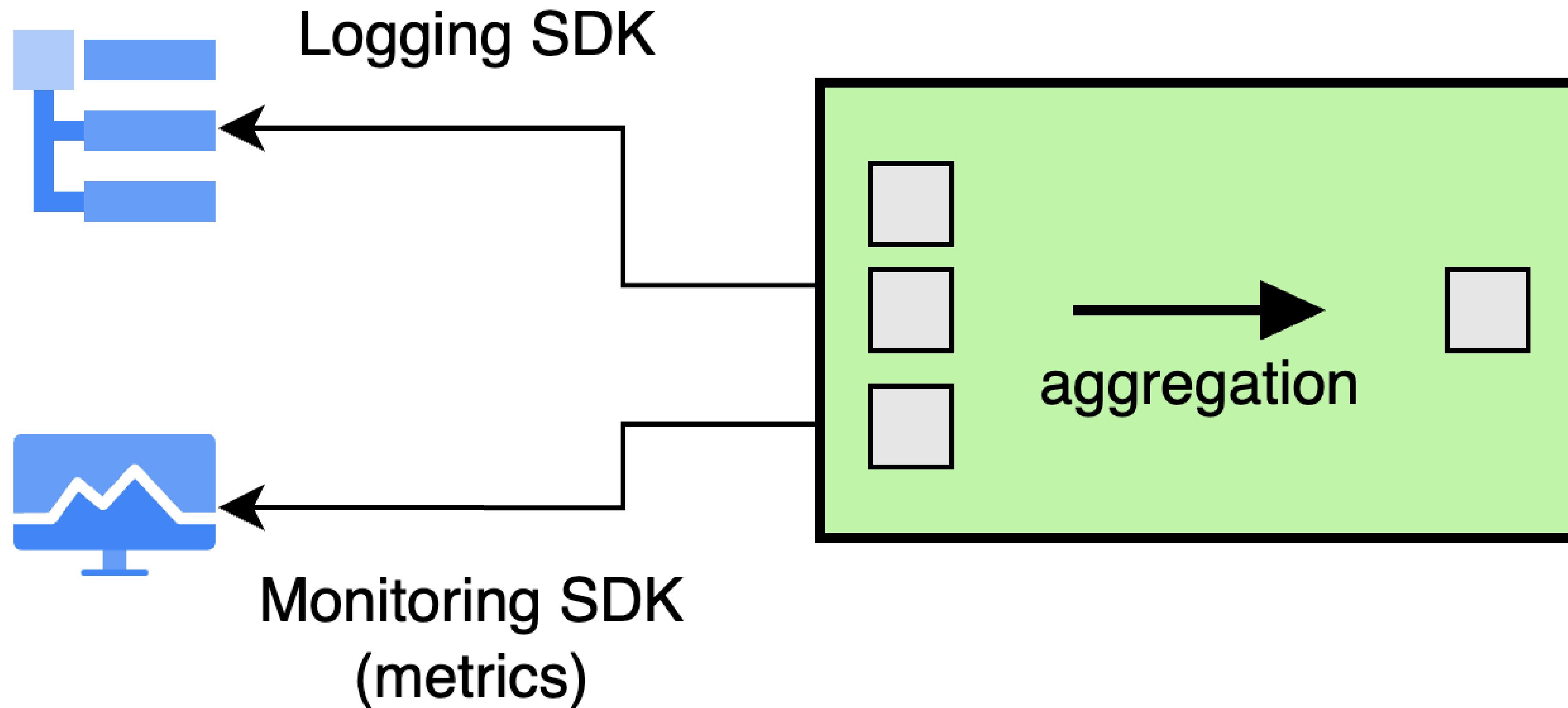
{format_instruction}

Monitor Controller - Logs & Metrics

Application Log and Metrics Real-Time Retrieval

- Application Log Structure
 - Remaining Task Count in Queue
 - Average Task Execution Time
 - ...
- What metrics we retrieved?
 - Container CPU Utilization (%)
 - Container Memory Utilization (%)
 - Instance Count (active)
 - Instance Count (idle)

Monitor Controller - Logs & Metrics



Discord bot

FLOW

1. Monitor send **Report Request**
2. Analyze notification data and transfer it into **message**. Open a **thread** at the same time for feedback.
3. If there are any **feedback**. Send it to **GenAI** and wait for **response**.

Discord bot

TSMC System Bot 機器人 昨天 21:51

ERROR

The application is experiencing high latency and is unable to keep up with the demand. The number of tasks in the queue has been above 100 for the past 5 minutes and the average task execution time has been above 30 seconds. I recommend increasing the number of instances by 1.

Adjustment

- CPU has to be decreased -1.
- Instance has to be increased 1.

ERROR Message by your system buddy • 昨天 19:05

Feedback Discussion 3則訊息，
機器人 TSMC System Bot AI response 12小時以前

TSMC System Bot 機器人 今天 02:30

INFO

The application is running smoothly and does not require any scaling action. The average task execution time is high, but this is likely due to the fact that the application is under heavy load. It is recommended to monitor the application closely and scale up if necessary.

INFO Message by your system buddy • 今天 02:30

Feedback Discussion 1則訊息，
機器人 TSMC System Bot Send me message if you have any s... 9小時以前

TSMC System Bot 機器人 今天 10:34

WARNING

The average task execution time is above 30 seconds, which may indicate that the application is under load. However, the container CPU and memory utilization are both low, so it is not clear what is causing the slow performance. It is recommended to monitor the application closely and scale up if necessary.

WARNING Message by your system buddy • 今天 10:33

Feedback Discussion 1則訊息，
機器人 TSMC System Bot Send me message if you have any su... 1小時以前

Feedback Discussion

開啟者 TSMC System Bot

2024年1月27日

TSMC System Bot 機器人 今天 11:45

WARNING

The average task execution time is above 30 seconds, which may indicate that the application is under load. However, the container CPU and memory utilization are both low, so it is not clear what is causing the slow performance. It is recommended to monitor the application closely and scale up if necessary.

WARNING Message by your system buddy • 今天 11:45

TSMC System Bot 機器人 今天 11:45

Send me message if you have any suggestion!

peter lee 今天 11:45

coud you give me the log

TSMC System Bot 機器人 今天 11:45

The following text contains log data for a Google Cloud Run application. This data is presented in CSV format and encompasses the most recent 1 minutes:

Time,Remaining Task Count in Queue,Average Task Execution Time,Container CPU Utilization (%),Container Memory Utilization (%)
2024-01-27 03:45:03+00:00,434,146.83017834431027,0.9986112510343546,17.52471923828135

+ 傳訊息到 "Feedback Discussion"

OUTLINE

User Story

System Architecture

Application Implement

Demo

DevOps

Testing

Future

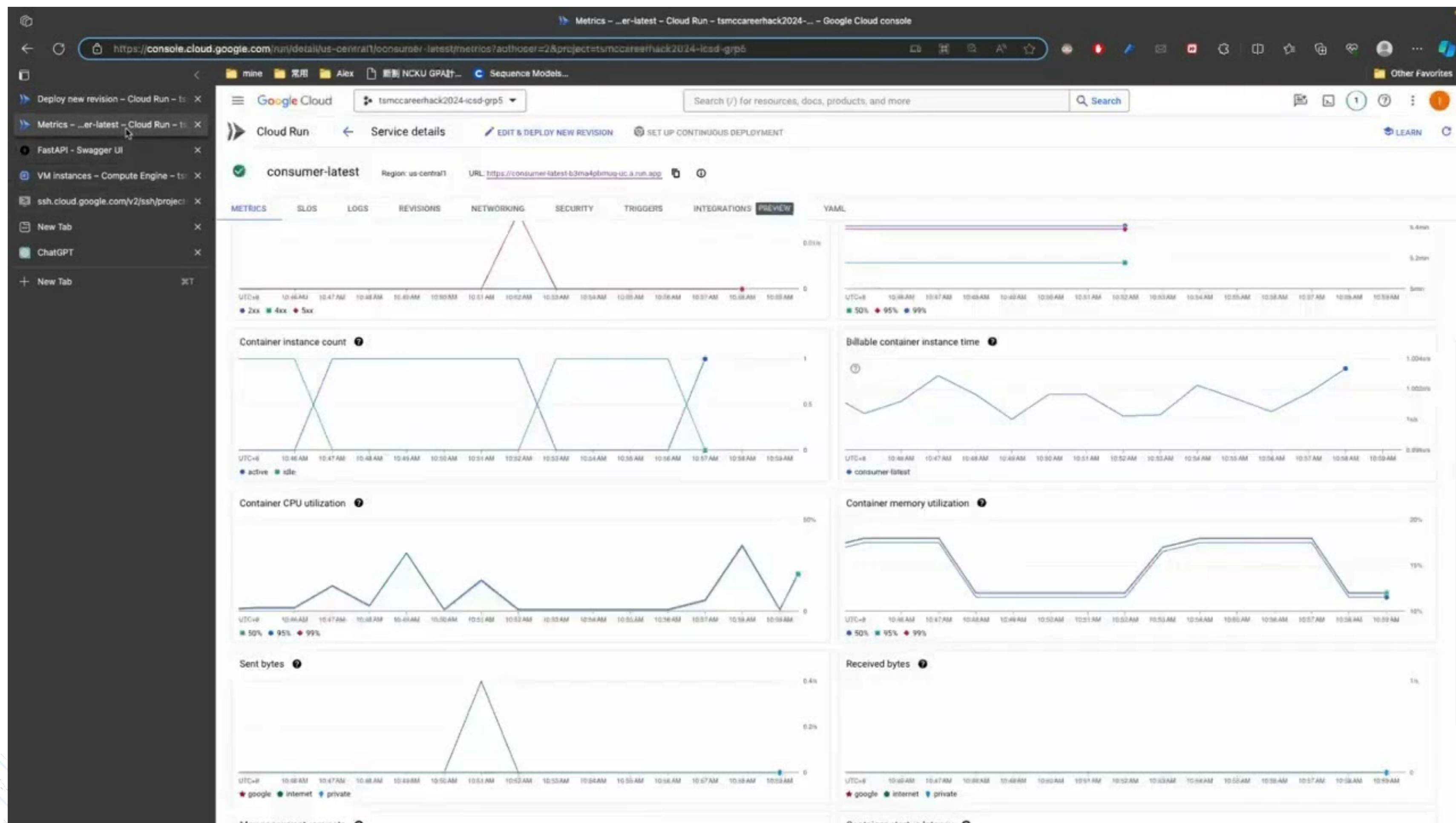


Demo - Testing Dataset

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows the project structure for "TSMC-HACKATHON-2024-IT-INFRA".
- Code Editor:** Displays the content of `simulate_runner.py`. The code performs the following steps:
 - Imports necessary modules: `pd`, `asyncio`, `argparse`, `discord`, and `os`.
 - Defines a class `RealTimeDataSimulator` with a method `is_end` that checks if the end of the data is reached.
 - An asynchronous function `main` takes a `data_directory` as input. It preprocesses metric data, starts a bot in a thread, and then enters a loop where it prints analysis results every 6 seconds. The results include timestamp, CPU utilization, memory utilization, instance count, and request count. It also sends alerts via discord.
 - If the script is run directly (`__name__ == "__main__"`), it creates an argument parser to handle the `--data` option, which specifies the directory of data to preprocess.
- Terminal:** Shows the command `python simulate_runner.py --data ./data/test/` being run, followed by log output indicating the bot has connected to the gateway.
- Bottom Status Bar:** Shows the current file path as `/dev/s17`, the Python version as `v0.1.0`, and the session ID as `v3.11.3 (tsmc-hackathon-2024-it-infra-py3.11)`.

Demo - Real time simulation



OUTLINE

User Story

System Architecture

Application Implement

Demo

DevOps

Testing

Future



DevOps Outline

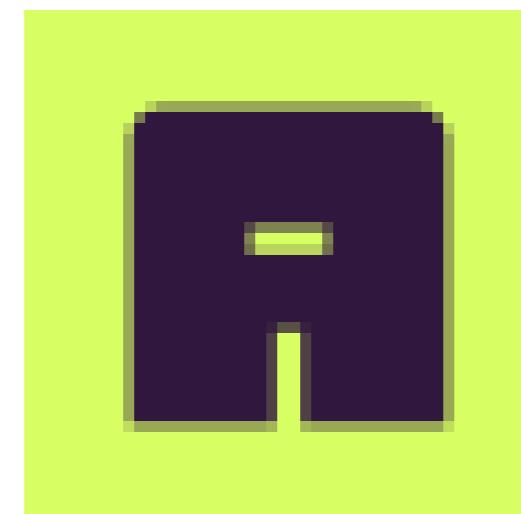
- Linting
- CI/CD
- Demo

Linting



Pre Commit

Auto trigger on commit



Ruff

Python linter and code formatter, written in Rust.

Linting - commit

The screenshot shows a dark-themed code editor interface with several tabs at the top: `docker-compose.yml`, `! test-monitor.yaml M`, `! wheel-ci-test.yaml M X`, `! test.yaml M`, and `service.py`. The main area displays a GitHub Workflow YAML file:

```
name: Consumer CI

on:
  push:
    branches: [ feature/wheel_ci ]
    tags:
      - "*"

env:
  WHEEL_CI_SERVICE_
  WHEEL_CI_SERVICE_

jobs:
  docker:
    runs-on: ubuntu
    steps:
      - name: ruff
        uses:
```

A tooltip is displayed over the word `Failed` in the `steps` section of the workflow. The tooltip contains the following information:

- Icon: A yellow triangle with an exclamation mark and a blue icon.
- Text: **Git:**
- Text: **ruff.....**
- Text: **.....Failed**
- Buttons:
 - Open Git Log** (highlighted)
 - Show Command Output**
 - Cancel**

At the bottom of the screen, there is a terminal window showing log output:

```
INFO:test-monitor:Output: /Users/jason/Desktop/TSMC-Hackathon-2024-IT-Infra/monitor/tests/test_c
32 0 100%
INFO:test-monitor:Output: /Users/jason/Desktop/TSMC-Hackathon-2024-IT-Infra/monitor/tests/test_c
py 32 0 100%
INFO:test-monitor:Output: -
```

Linting - Ruff

```
import yaml
import sys

def check_new_task():
    # if ./tasks/* exists
    # using `ls ./tasks`
    # then run the task

    ls_output = subprocess.check_output(["ls", "./tasks"], env=os.environ)
    ls_output = ls_output.decode("utf-8")

    if len(ls_output) > 0:
        print(ls_output)

    else:
        print("No tasks found in ./tasks directory")
```

→

```
5 import yaml
6
7
8 def check_new_task():
9     # if ./tasks/* exists
10    # using `ls ./tasks`
11    # then run the task
12
13+ ls_output = subprocess.check_output(
14+     ["ls", "./tasks"], env=os.environ.copy(), stderr=subprocess.STDOUT
15+ )
16    ls_output = ls_output.decode("utf-8")
17
18    if len(ls_output) > 0:
```

CI/CD - Attempt

- Github Action with GCP
- Github Action with SSH
- Drone CI



Fail: Due to GCP Permission

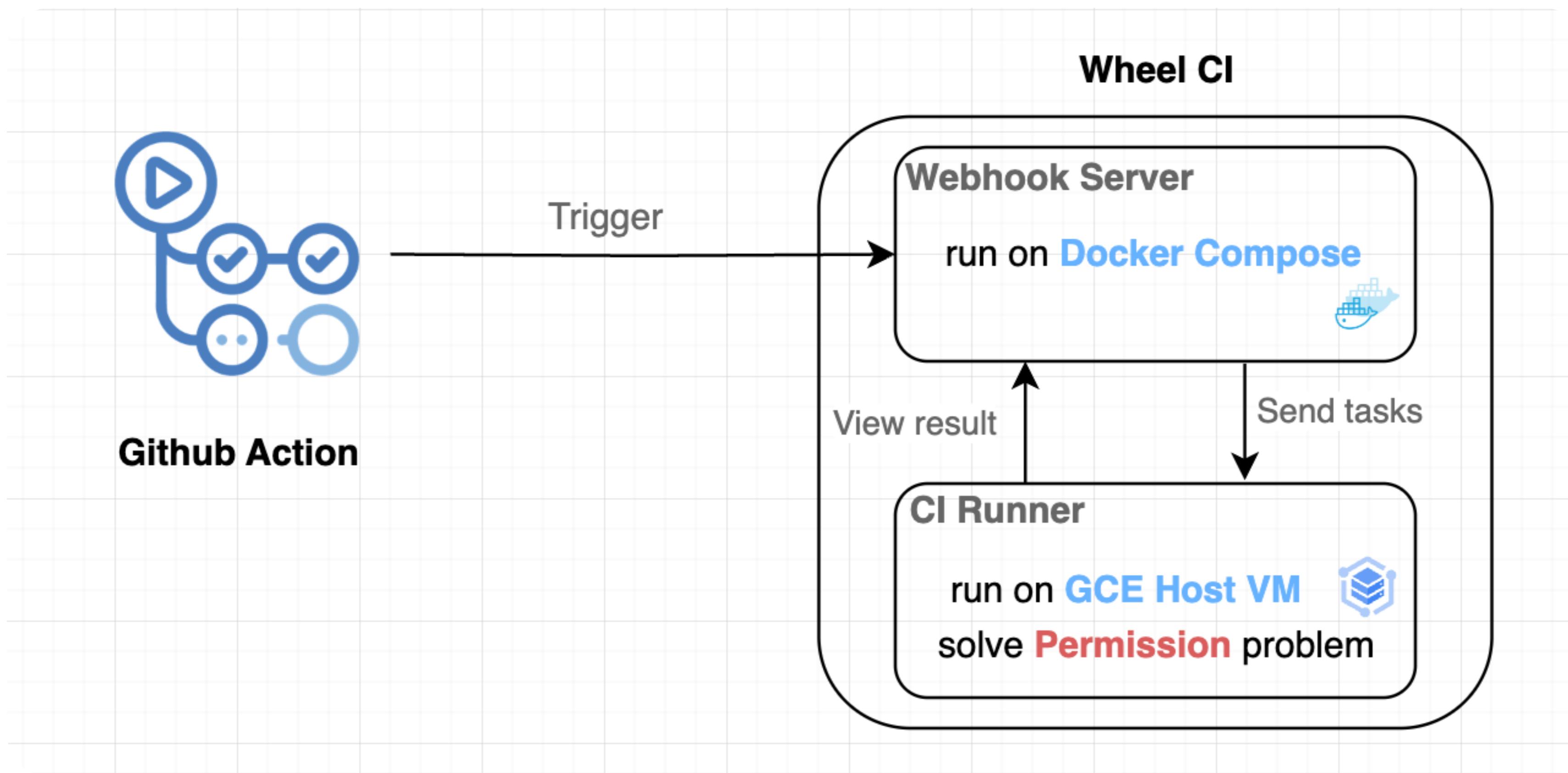
The screenshot shows a dark-themed CI/CD pipeline interface. On the left, a sidebar lists 'Summary', 'Jobs' (with 'test' selected), 'Run details', 'Usage', and 'Workflow file'. The main area displays the 'test' job, which failed 15 hours ago in 52s. The job's steps are listed as follows:

- > ✓ Set up job
- > ✓ Checkout code
- > ✓ Auth with Google Cloud
- > ✓ Install poetry
- > ✓ Run actions/setup-python@v5
- > ✓ Install dependencies
- > ✘ Run Monitor Tests
 - 1 ► Run poetry run coverage run -m pytest monitor/tests/test_cloud_run_manager.py
 - 19 ===== test session starts =====
 - 20 platform linux -- Python 3.11.7, pytest-7.4.4, pluggy-1.4.0
 - 21 rootdir: /home/runner/work/TSMC-Hackathon-2024-IT-Infra/TSMC-Hackathon-2024-IT-Infra

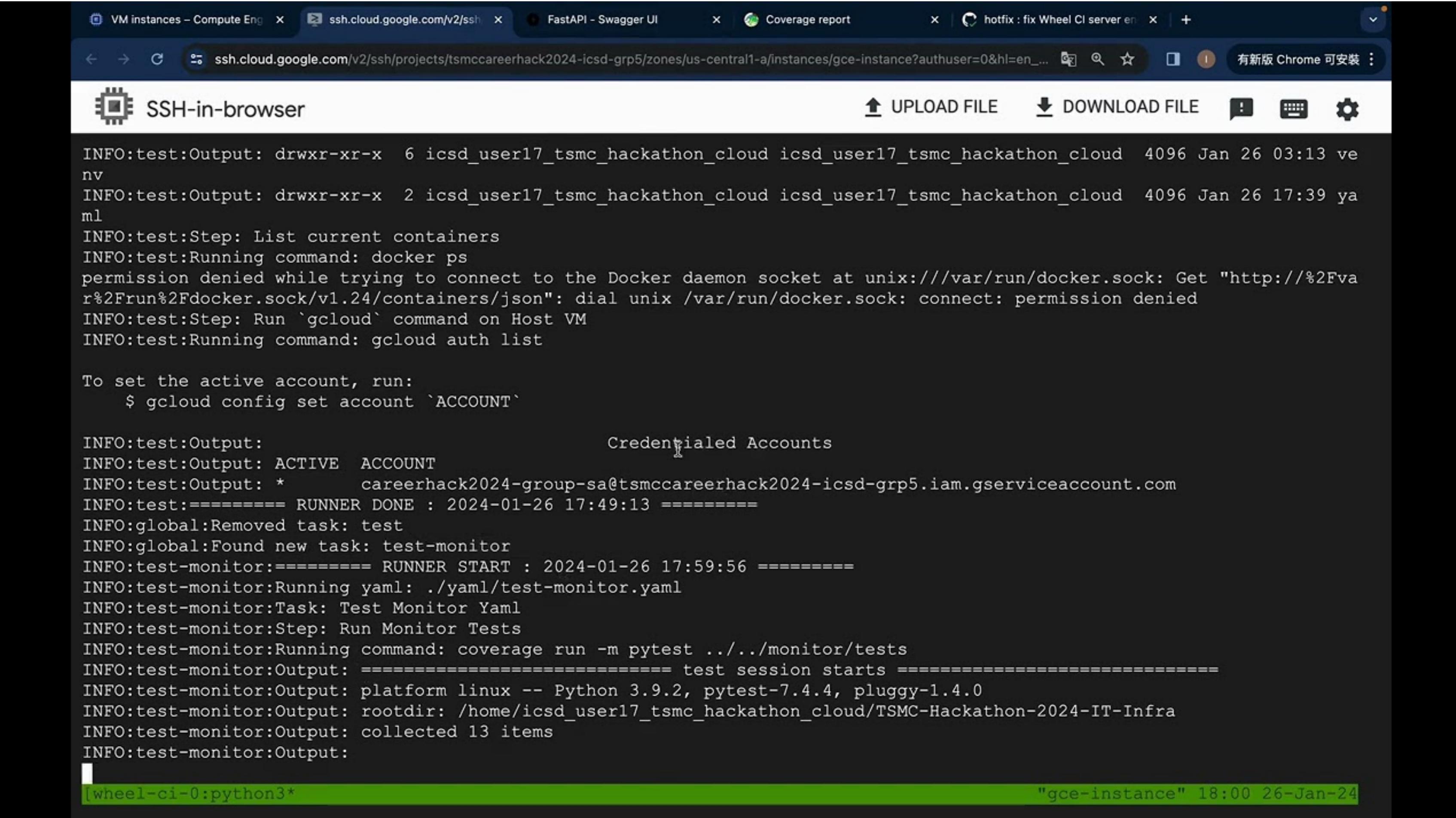
```
grp5/locations/us-central1/services/consumer-test' (or resource may not exist).
510 FAILED monitor/tests/test_cloud_run_manager.py::test_cloud_run_manager_adjust_instance_count -
    google.api_core.exceptions.PermissionDenied: 403 Permission 'run.services.get' denied on resource 'projects/tsmccareerhack2024-icsd-
    grp5/locations/us-central1/services/consumer-test' (or resource may not exist).
511 FAILED monitor/tests/test_cloud_run_manager.py::test_cloud_run_manager_increase_instance_count -
    google.api_core.exceptions.PermissionDenied: 403 Permission 'run.services.get' denied on resource 'projects/tsmccareerhack2024-icsd-
    grp5/locations/us-central1/services/consumer-test' (or resource may not exist).
512 FAILED monitor/tests/test_cloud_run_manager.py::test_cloud_run_manager_get_metrics - google.api_core.exceptions.PermissionDenied:
    403 Permission 'run.services.get' denied on resource 'projects/tsmccareerhack2024-icsd-grp5/locations/us-central1/services/consumer-
    test' (or resource may not exist).
513 ===== 6 failed, 1 passed in 3.11s =====
514 Error: Process completed with exit code 1.
```

CI/CD - Self-made CI

> build our own Wheel



CI/CD Workflow Demo



The screenshot shows a browser window with multiple tabs at the top, including "VM instances - Compute Eng", "ssh.cloud.google.com/v2/ssh", "FastAPI - Swagger UI", "Coverage report", and "hotfix : fix Wheel CI server en...". The main content area is titled "SSH-in-browser" and displays a terminal session log. The log includes:

- INFO: test: Output: drwxr-xr-x 6 icsd_user17_tsmc_hackathon_cloud icsd_user17_tsmc_hackathon_cloud 4096 Jan 26 03:13 ve nv
- INFO: test: Output: drwxr-xr-x 2 icsd_user17_tsmc_hackathon_cloud icsd_user17_tsmc_hackathon_cloud 4096 Jan 26 17:39 ya ml
- INFO: test: Step: List current containers
- INFO: test: Running command: docker ps
- permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fva r%2Frun%2Fdocker.sock/v1.24/containers/json": dial unix /var/run/docker.sock: connect: permission denied
- INFO: test: Step: Run `gcloud` command on Host VM
- INFO: test: Running command: gcloud auth list

To set the active account, run:

```
$ gcloud config set account `ACCOUNT`
```

INFO: test: Output: Credentialed Accounts

INFO: test: Output: ACTIVE ACCOUNT

INFO: test: Output: * careerhack2024-group-sa@tsmccareerhack2024-icsd-grp5.iam.gserviceaccount.com

INFO: test: ===== RUNNER DONE : 2024-01-26 17:49:13 =====

INFO: global: Removed task: test

INFO: global: Found new task: test-monitor

INFO: test-monitor: ===== RUNNER START : 2024-01-26 17:59:56 =====

INFO: test-monitor: Running yaml: ./yaml/test-monitor.yaml

INFO: test-monitor: Task: Test Monitor Yaml

INFO: test-monitor: Step: Run Monitor Tests

INFO: test-monitor: Running command: coverage run -m pytest ../../monitor/tests

INFO: test-monitor: Output: ===== test session starts =====

INFO: test-monitor: Output: platform linux -- Python 3.9.2, pytest-7.4.4, pluggy-1.4.0

INFO: test-monitor: Output: rootdir: /home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra

INFO: test-monitor: Output: collected 13 items

INFO: test-monitor: Output:

[wheel-ci-0:python3* "gce-instance" 18:00 26-Jan-24]

OUTLINE

User Story

System Architecture

Application Implement

Demo

DevOps

Testing

Future



Testing

- Log Analyzer
- Monitor Controller

Testing - Log Analyzer

```
Coverage for ai2/log_analyzer.py: 97%
66 statements 64 run 2 missing 0 excluded
« prev ^ index » next coverage.py v7.4.0, created at 2024-01-27 11:19 +0800

1 import pandas as pd
2 from langchain.output_parsers import ResponseSchema, StructuredOutputParser
3 from langchain.prompts import PromptTemplate
4 from langchain_google_vertexai import VertexAI
5 from pinecone import Pinecone
6 from vertexai.language_models import TextEmbeddingModel
7
8
9 class LLMLogAnalyzer:
10     """
11         LLMLogAnalyzer is a class that performs log analysis and provides scaling recommendations for a Google Cloud Run application.
12
13     Args:
14         pinecone_api_key (str): The API key for accessing the Pinecone service.
15         index_name (str): The name of the Pinecone index.
16         llm_args (dict): Additional arguments for initializing the VertexAI model.
17
18     Attributes:
19         llm (VertexAI): The VertexAI model for performing log analysis.
20         db (Pinecone.Index): The Pinecone index for storing log embeddings.
21         embedding_model (TextEmbeddingModel): The text embedding model for generating log embeddings.
22         output_parser (StructuredOutputParser): The output parser for parsing the analysis feedback.
23         format_instruction (str): The format instruction for providing scaling recommendations.
24         prompt_template (PromptTemplate): The prompt template for generating analysis prompts.
25
26     Methods:
27         analyze_log: Analyzes the log data and provides scaling recommendations.
28         store_memory: Stores the analysis feedback and log data in the Pinecone index.
29         chat: Performs a conversation with the AI based on the stored analysis feedback.
30
31     """
32
33     def __init__(self, pinecone_api_key: str, index_name: str, llm_args: dict) -> None:
34         """
35             Initializes the LLMLogAnalyzer.
36
37         Args:
38             pinecone_api_key (str): The API key for accessing the Pinecone service.
39             index_name (str): The name of the Pinecone index.
40             llm_args (dict): Additional arguments for initializing the VertexAI model.
41
42         """
43         self.llm = VertexAI(**llm_args)
44
45         pc = Pinecone(api_key=pinecone_api_key)
46         self.db = pc.Index(index_name)
47
48         self.embedding_model = TextEmbeddingModel.from_pretrained(
49             "textembedding-gecko@003"
50         )
51
52         # Define output parser and generate format instruction
53         severity_schema = ResponseSchema(
```

Total Coverage:

- 98%

Coverage report: 98%

coverage.py v7.4.0, created at 2024-01-27 11:19 +0800

Module	statements	missing	excluded	coverage
ai2/log_analyzer.py	66	2	0	97%
ai2/tests/__init__.py	0	0	0	100%
ai2/tests/conftest.py	15	0	0	100%
ai2/tests/log_analyzer_test.py	14	0	0	100%
Total	95	2	0	98%

coverage.py v7.4.0, created at 2024-01-27 11:19 +0800

Testing - Monitor Controller

Total Coverage:

- 82%

Module	statements	missing	excluded	coverage
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/__init__.py	0	0	0	100%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/service/__init__.py	0	0	0	100%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/service/cloudrun.py	151	45	0	70%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/service/conversation_manager.py	33	0	0	100%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/tests/__init__.py	0	0	0	100%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/tests/conftest.py	9	0	0	100%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/tests/test_cloud_run_manager.py	32	0	0	100%
/home/icsd_user17_tsmc_hackathon_cloud/TSMC-Hackathon-2024-IT-Infra/monitor/tests/test_conversation_manager.py	32	0	0	100%
Total	257	45	0	82%

coverage.py v7.4.0, created at 2024-01-26 18:01 +0000

OUTLINE

User Story

System Architecture

Application Implement

Demo

DevOps

Testing

Future



Future Improvements

GenAI

- Allow user to choose different language model
- Used advanced chain to enable additional functionalities
 - Ex. Generate code, Visualize data
- Incorporate RAG to learn from documents

Thank you for your attention

