Cloud-Based Ransomware Detection and Recovery System Report

By : Sanskar Solanki

Table of Contents

Introduction

1. Objective **2**
2. Scope **2**

Report Components

2.1 Detection System

1. Tools and Techniques **2**
2. Implementation Steps **3**

2.2 Mitigation System

1. Tools and Techniques **5**
2. Implementation Steps **5**

2.3 Recovery System

1. Tools and Techniques **7**
2. Implementation Steps **7**

2.4 Incident Response **8**

2.5 Environment Testing **9**

1. Introduction

Objective

The objective of this project is to develop a robust system for detecting, mitigating, and recovering from ransomware attacks in a cloud-based environment using Google Cloud Platform (GCP) services. The solution aims to leverage GCP's free-tier resources to build a cost-effective yet functional security framework.

Scope

This project will utilize GCP services to:

- Detect potential ransomware activity using anomaly detection in cloud logs.
- Mitigate ransomware spread through Identity and Access Management (IAM) policies and real-time monitoring.
- Enable rapid recovery through automated backups, file versioning, and disaster recovery mechanisms.

2. Report Components

2.1 Detection System

Tools and Techniques:

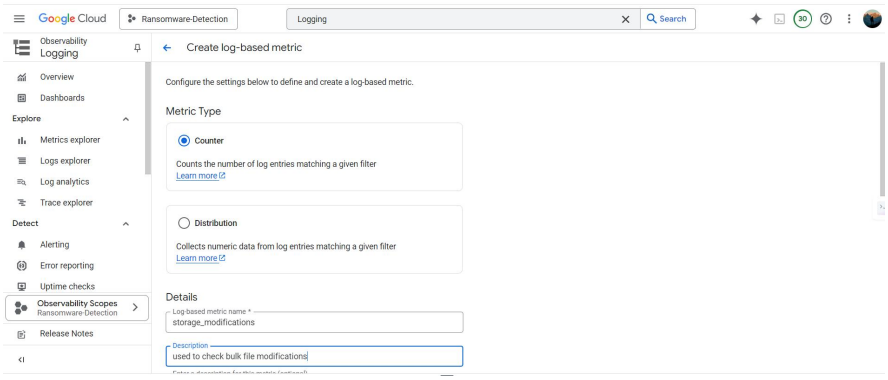
- **Cloud Logging:** Collect and analyze log data for unusual patterns, such as rapid file modifications or encryption activities.
- **Cloud Functions:** Automate alerts when suspicious activities occur, such as excessive failed login attempts or sudden spikes in resource utilization.
- **Job Scheduler :** Automatically backup sensitive data stored on cloud storage.
- **Pub/Sub Topics :** Triggers event , messages and alerts based on that , some action will take place to secure the system

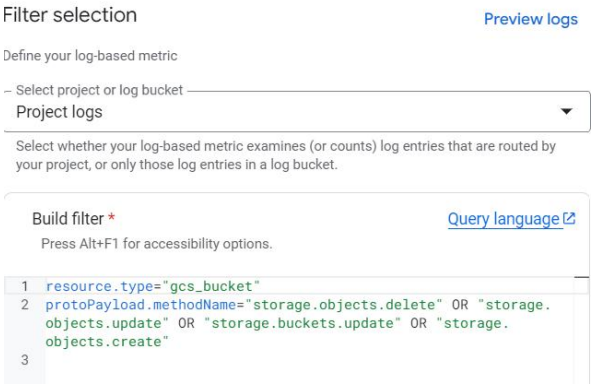
Implementation Steps:

Cloud Logging :

1. First , we navigate to Logging (one of the crucial component of operations suite) and move to **Logs Explorer** in Google Cloud Console.
2. Now we created **Log-Based Metric** , which will collect data from Logs and create metric to analyze large number of file modifications and Input /Output operations.
3. Here , we created metric to analyze cloud storage operations .

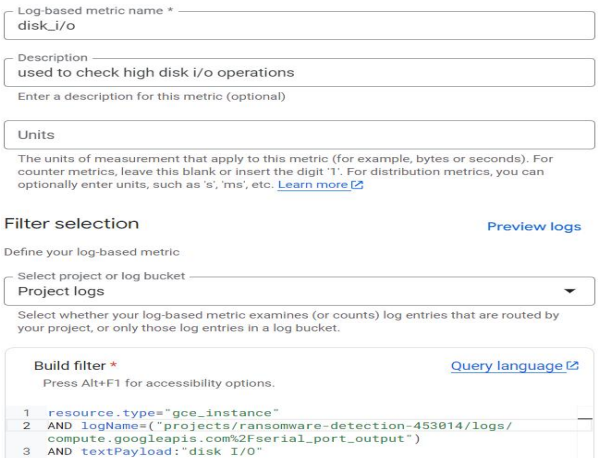
- **a)** **Metric Type** : Counter
- **b)** For Storage Bucket object operations





4. Next , we created metric high disk read/write operations in VM instances

- **a)** **Metric Type** : Counter

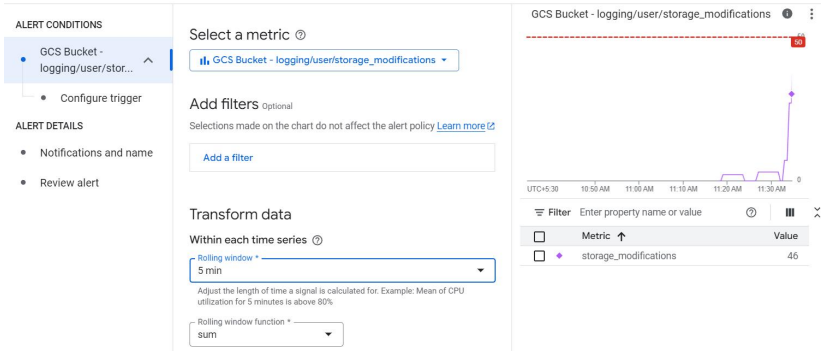


Cloud Monitoring :

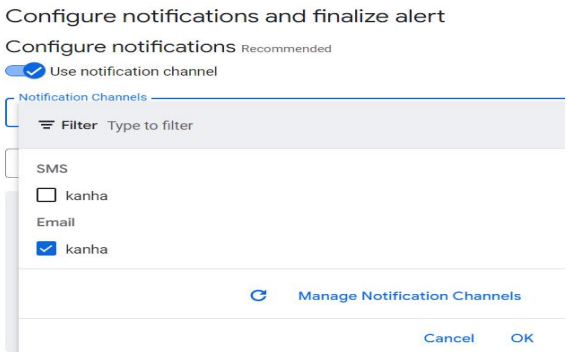
1. To use these log-based metric for generating alerts , we used alerting policies in cloud monitoring.

2. For **Storage based metric** :

- **a)** Select metric created above , set Rolling Window : 5 min with function : sum(total operations).
- **b)** With condition type of Threshold Value : 50



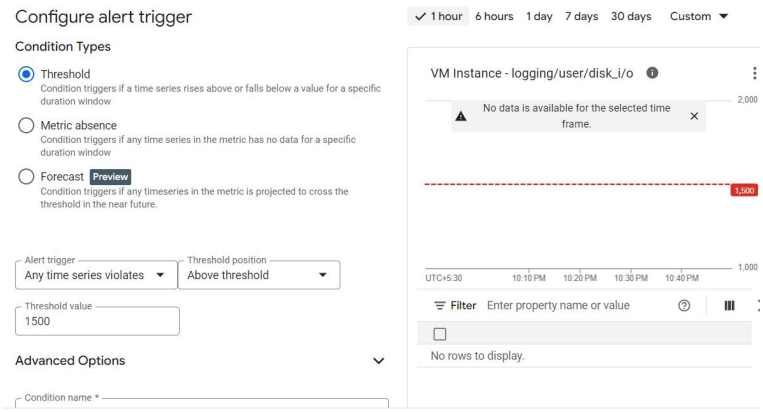
C) Also add , notification channel (email or phone). And give policy name.



3. For Disk based Metric

- **a)** Select metric , set Rolling Window : 5 min with function : sum.

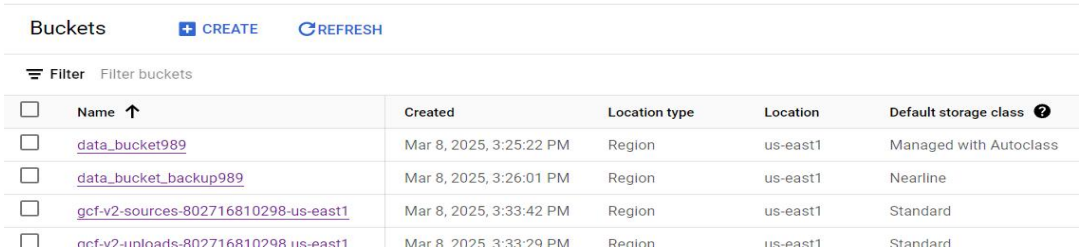
b) With condition Type of Threshold Value : 1500 and add notification channel for same.



Cloud Functions Implementation :

With this step , any file uploaded on Cloud Storage send alerts through log by Cloud Function triggered by event takes place on storage objects.

1. Enable Cloud Function API by navigating to API services through console or CLI.
2. Now select or create new bucket in cloud storage (in my case : data_bucket989) to upload files.



3. Create Cloud Run function files with : main.py and requirements.txt in same directory.

4. Deploy function in same region with cloud bucket :

```
sanskarsolanki417@cloudshell:~ (ransomware-detection-453014)$ gcloud functions deploy detecti
on --runtime python39 --trigger-event google.storage.object.finalize --trigger-resource=data_
bucket989 --entry-point=detect_ransomware --trigger-location=us-east1 --memory=256MB --region
=us-east1
```

5. Assign , default cloud storage service account with **Pub/Sub publisher** IAM role to publish message to cloud function.

```
sanskarsolanki417@cloudshell:~ (ransomware-detection-453014)$ gcloud projects add-iam-policy-
binding ransomware-detection-453014 --member=serviceAccount:802716810298-compute@developer.gs
erviceaccount.com --roles=pubsub.publisher
```

6. This will trigger cloud function , detection , which perform action on uploaded file and publish Pub/Sub Topic message .

2.2 Mitigation System

Tools and Techniques:

- **Identity and Access Management (IAM):** Implement least privilege principles to minimize access to sensitive data.
- **Firewall Rules:** Restrict unauthorized access to cloud resources.

Implementation Steps:

1. Define granular IAM roles and assign them to users and service accounts.
    1. Use manual service account with specific and least privilege role (**role/storage.editor**).
    2. Assign required permissions to default service account like default cloud function service account with **Cloud Function Invoker** role.
    3. Assign Role to default compute service account with required roles like **Pub/Sub Publisher**.
2. Define firewall rules to restrict traffic within boundary.
    1. Create Deny Egress Rule to malicious IP addresses like (**45.227.252.12/32**).
    2. Restrict incoming traffic only for required and allowed ports .

2.3 Recovery System

Tools and Techniques:

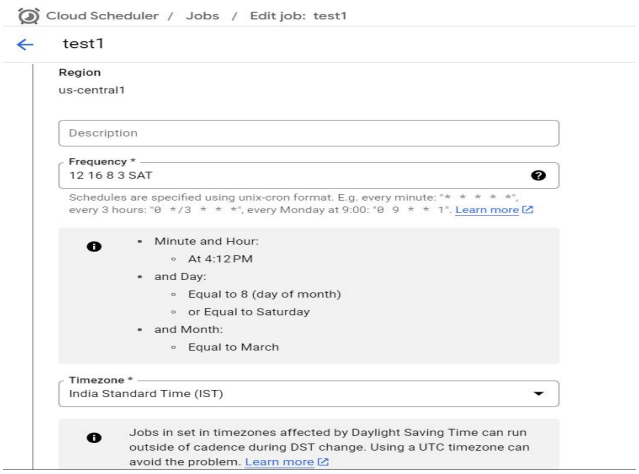
- **Cloud Storage with Versioning:** Maintain multiple versions of files to prevent data loss.
- **Cloud Scheduler:** Automate periodic backups to offsite storage.

Implementation Steps:

1. Enable Object Versioning in Cloud Storage buckets.

```
sanskarsolanki417@cloudshell:~ (ransomware-detection-453014)$ gsutil versioning set on gs://d
ata_bucket989
Enabling versioning for gs://data_bucket989/...
```

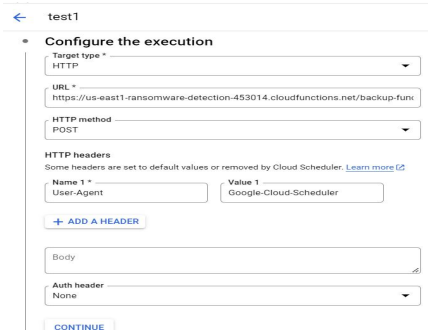
2. **Cloud Scheduler :** This will help to automate tasks like backup data for disaster recovery
    1. Enable Cloud Scheduler API.
    2. Create new Job with test1 name , set region (any) , frequency in correct format and timezone.



    3. Now to configure the execution , we need to trigger **HTTP request** requires Cloud Function URL . Ensure that **Job Scheduler's** service account have Cloud Functions Invoke role.

```
- members:
  - serviceAccount:cloud-storage-service-account@ransomware-detection-453014.iam.gservi
  - serviceAccount:service-802716810298@gcp-sa-cloudscheduler.iam.gserviceaccount.com
  role: roles/cloudfunctions.invoker
```

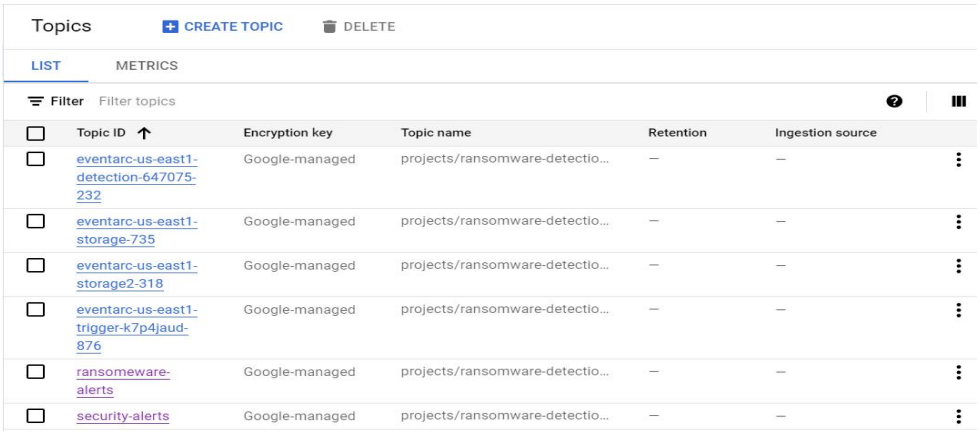
    4. Create **Cloud Function** : backup_function , copy url and add this in Job Scheduler with POST method.



5. Ensure that destination bucket is created.

2.4 Incident Response

1. In this we set up Pub/Sub for Alerts , create topic for **cloud function (**file detection**)** which will pubish messages with subscription.Create topic for, **Cloud Function (**test function for compromised instances**)** which will get triggered once , topic receives message as **subscriber**.

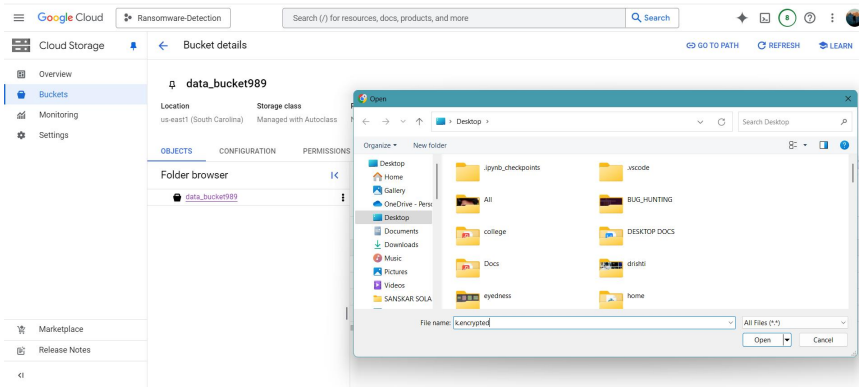


2. Implement **Cloud Function** to isolate vm instance , when it gets triggered by Pub/Sub (**security-alerts**). Function : isolate-vm-function and Requirements file : requirements

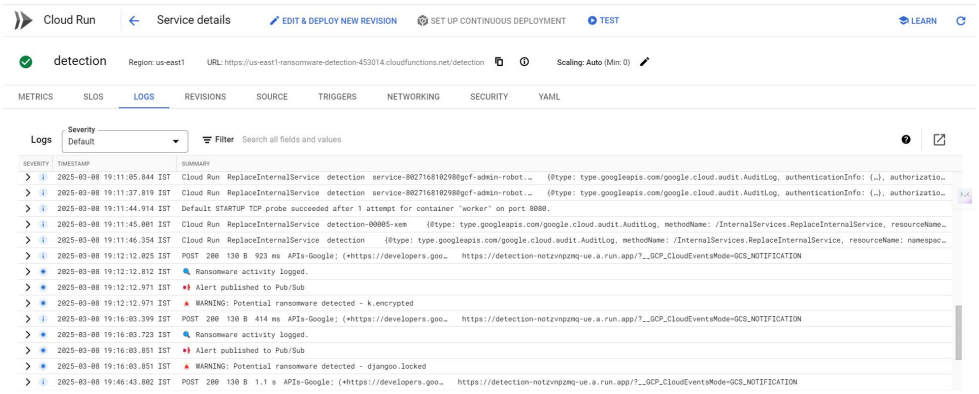
```
sanskarsolanki417@cloudshell:~ (ransomware-detection-453014)$ gcloud functions deploy isolate
-vm --trigger-topic=security-alerts --runtime python39 --entry-point=isolate_vm --memory=256M
B --region=us-east1
```

2.5 Environment Testing

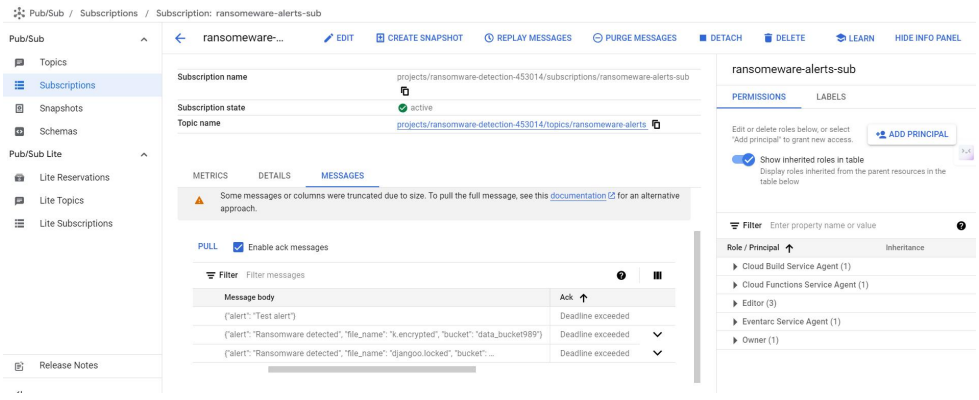
1. First by uploading any file containing malicious extension, to detect if , function writing logs and Publishes messages on Pub/Sub or not. (Here **k.encrypted**).



2. We can see , in Logs we received alert , detecting malicious file. We can also check it on Log Explorer.



3. Here , on ransomware-alerts topic of Pub/Sub we can pull messages published by cloud function.

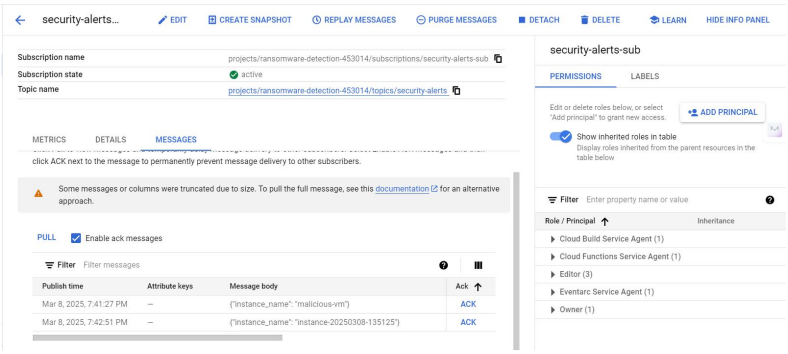


4. Now to check , if any message Published by **Security Command Center** when it detects malicious thing happened on instance , then that vm instance gets isolated by cloud function triggered through Pub/Sub Topic message publisher.

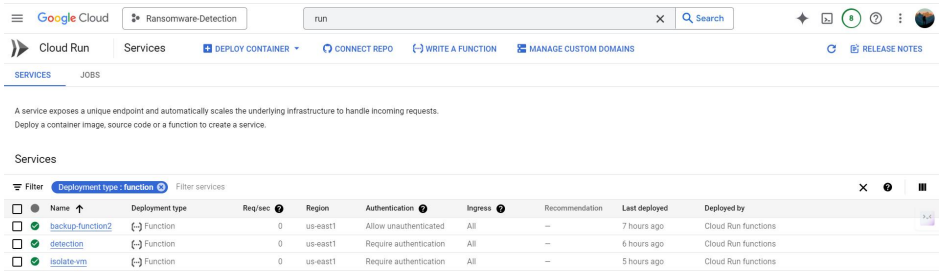
5. To use security command center , we need to be part of any organization . So here we use **CLI** to publish message on security-alerts topic.

```
sanskarsolanki417@cloudshell:~ (ransomware-detection-453014)$ gcloud pubsub topics publish se
curity-alerts --message='{"instance_name": "instance-20250308-135125"}'
```

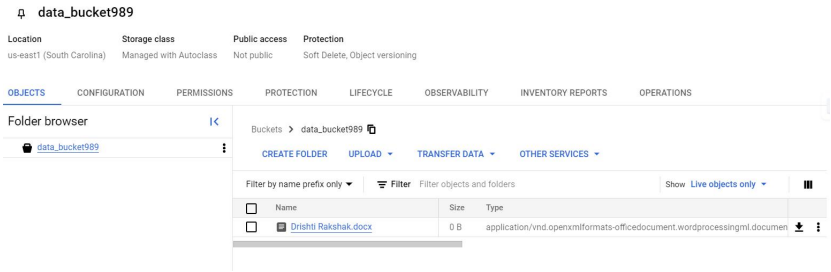
6. We can see, message is published and this will trigger cloud function to **isolate vm**



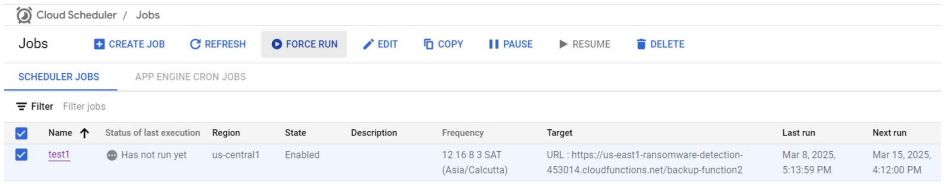
7. List of all cloud run functions.



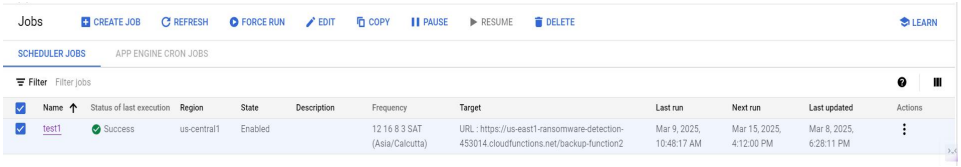
8. To check , if backup-function working correctly , we can forcely run it through Job Scheduler.
    1. First we upload file in data_bucket989 .



    2. Now , we can run scheduler forcely.

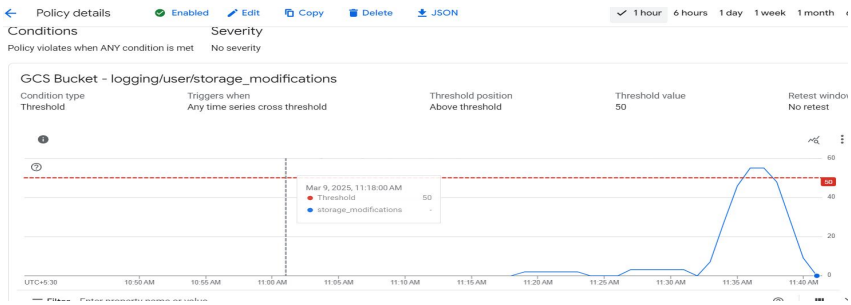


    3. And here we got success.



4. We can also check , if log-based metrics collecting data , and generating alerts or not manually.

```
sanskarsolanki417@cloudshell:~ (ransomware-detection-453014)$ for i in {1..55}; do
  echo "Test $i" > test-file-$i.txt
  gsutil cp test-file-$i.txt gs://data_bucket989/
done
```



5. And here we got mail from google cloud.