



School of Information Technology and Engineering

**Cloud Application Development
Assignment 4**

Title: Application Security Best Practices and Scaling Applications on Google Cloud

**Done by: Altair Kabdrakhmanov
4th year student, ID: 21B030829,
Information Systems
Date of submission: 24th November**

Almaty, Fall 2024

Executive Summary

This report shows how to improve security and scalability for apps on Google Cloud. For security was used IAM to control access, encrypted data with Google Cloud KMS, and added HTTPS for safe data transfer. Also it was set up monitoring, tested the app for security issues, and made a plan to handle security problems.

For scalability, it was used horizontal and vertical scaling to handle more users and traffic. Tools like Kubernetes auto-scaling and load balancing helped keep the app running smoothly. To save costs, we chose the right storage, used serverless options, and set alerts to track usage and spending.

Table of Contents

- 1. Introduction
- 2. Application Security Best Practices
 - 2.1. Overview of Cloud Security
 - 2.2. IAM Configuration
 - 2.3. Data Protection
 - 2.4. Security Testing
 - 2.5. Monitoring and Logging
 - 2.6. Incident Response
- 3. Scaling Applications on Google Cloud
 - 3.1. Overview of Scalability
 - 3.2. Application Design
 - 3.3. Scaling Methods
 - 3.4. Load Balancing
 - 3.5. Auto-Scaling Implementation
 - 3.6. Performance Monitoring
 - 3.7. Cost Optimization Strategies
- 4. Conclusion
- 5. Recommendations
- 6. References
- 7. Appendices

1. Introduction

Google Cloud is a powerful platform that helps businesses run applications and store data on cloud. It offers many services, like computing power, storage, and tools for managing apps. Google Cloud is used by companies around the world because it is reliable, flexible, and easy to scale.

In cloud applications, security and scalability are very important. Security keeps data safe from unauthorized access, while scalability ensures the app can handle more users and work as needed. Without good security, sensitive data can be exposed, and without scalability, the app can become slow or stop working during busy times.

The purpose of this report is to explore the best practices for security and scaling applications on Google Cloud. It will explain how to secure an app using Google Cloud's tools and how to make the app scale easily to handle growing demand. The scope of this report includes setting up security, testing for vulnerabilities, monitoring, handling incidents, and optimizing costs while scaling the app on Google Cloud.

To start the project we need to sign in gcloud use `gcloud auth login` command

```
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS zsh + - ×

(base) kabdrakhman@MacBook-Pro-Altair Assignment-4 % gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=3255940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhos
t%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%
2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=xu3e1k4fRzzbN5Btg7REHpn7fB1rw&access_type=offline&code_challenge=3uyqSYCv
-2QjevjxVSg1GEaR0oQdMSAsWiHhUhx40I&code_challenge_method=S26

You are now logged in as [altairkabdrakhmanov@gmail.com].
Your current project is [None]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
```

and to create a project with this command: `gcloud projects create --name="CloudAppDev-Assignment-4"`

```
+ gcloud config set project PROJECT_ID
(base) kabdrakhman@MacBook-Pro-Altair Assignment-4 % gcloud projects create --name="CloudAppDev-Assignment-4"
No project ID provided.

Use [cloudappdev-assignment-4] as project ID (Y/n)? y

Create in progress for [https://clouddre sourcemanager.googleapis.com/v1/projects/cloudappdev-assignment-4].
Waiting for [operations/cp.8682396706275599641] to finish...done.
Enabling service [cloudapis.googleapis.com] on project [cloudappdev-assignment-4]...
Operation "operations/acat.p2-624543893269-59897604-0b1a-43c4-8ba8-b52a8be0ce78" finished successfully.
(base) kabdrakhman@MacBook-Pro-Altair Assignment-4 %
```

For this assiggmnet was used simple Flask application “Phonebook” with CRUD Operations. To deploy project used command `gcloud app deploy`

```
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS zsh - Assignment-4

source: [/Users/kabdrakhman/Documents/KBTU/7th semester/Cloud-Application-Development/Assignment-4]
target project: [cloudappdev-assignment-4]
target service: [default]
target version: [20241124t200822]
target url: [https://cloudappdev-assignment-4.uc.r.appspot.com]
target service account: [cloudappdev-assignment-4@appspot.gserviceaccount.com]

Do you want to continue (Y/n)? y
Beginning deployment of service [default]...
[ Uploading 0 files to Google Cloud Storage ]
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://cloudappdev-assignment-4.uc.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
(venv) (base) kabdrakhman@MacBook-Pro-Altair Assignment-4 %
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF

2. Application Security Best Practices

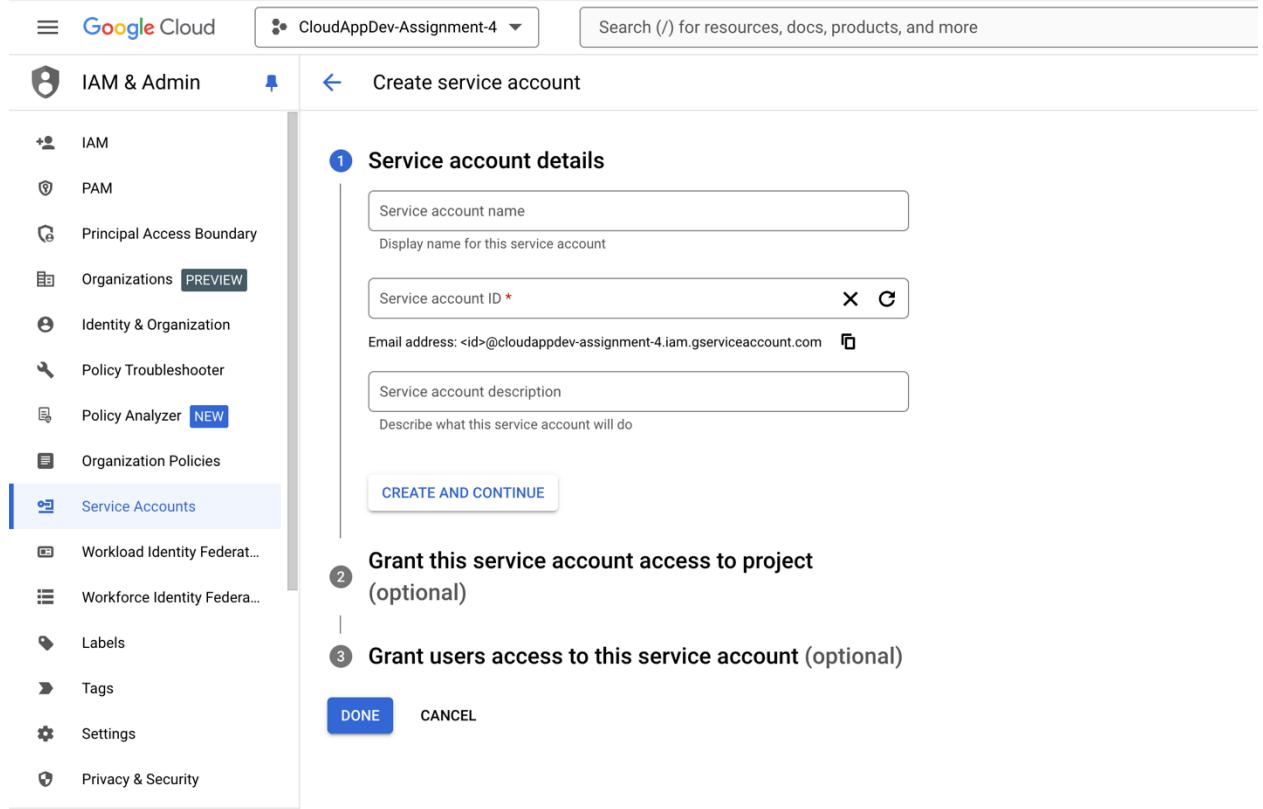
2.1. Overview of Cloud Security

Cloud security is set of practices and technologies used to protect data, applications, and infrastructure in cloud environments. It focuses on data confidentiality, integrity, and availability. Google Cloud provides built-in tools like Identity and Access Management (IAM), encryption, monitoring, and logging to implement security.

2.2. IAM Configuration

Identity and Access Management (IAM) is a Google Cloud tool that helps control who can access your resources and what actions they can perform. It allows you to create roles with specific permissions, follow the principle of least privilege to give only the necessary access, and set conditions like time or location to limit access further.

Now move to creation of a service account for my application and assign the principle of least privilege we need to go to the following link <https://console.cloud.google.com/iam-admin>. Select “Service Accounts” option on the left table. Click on “CREATE SERVICE ACCOUNT”.



You will see this window. Write name and go the second option.

I provide Role Viewers for this service account. That means it can view everything in my project, but cannot change anything.

The screenshot shows the Google Cloud IAM & Admin interface. On the left, a sidebar lists various IAM-related services: IAM, PAM, Principal Access Boundary, Organizations (PREVIEW), Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, Service Accounts (selected), Workload Identity Federat..., Workforce Identity Feda..., Labels, Tags, Settings, and Privacy & Security. The main panel is titled "Create service account". It has three steps: 1. Service account details (marked with a checkmark), 2. Grant this service account access to project (optional) (marked with a question mark), and 3. Grant users access to this service account (optional). Step 2 is currently active. In the "Grant this service account access to project (optional)" section, a dropdown menu under "Role" is set to "Viewer". Below it, a note says "View most Google Cloud resources. See the list of included permissions." A "CONTINUE" button is visible. Step 3 is partially visible below.

As result we can see list of available service accounts in our project.

The screenshot shows the Google Cloud IAM & Admin interface with the "Service Accounts" tab selected. The main panel displays a table of service accounts for the project "CloudAppDev-Assignment-4". The table columns are: Email, Status, Name, Description, Key ID, Key creation date, OAuth 2 Client ID, and Actions. One service account is listed:

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
service-account-for-assignment@cloudappdev-assignment-4.iam.gserviceaccount.com	Enabled	service-account-for-assignment4	This service account is created explicitly for assignment 4	No keys		117873366926502482584	

Now we can configure IAM Conditions to Restrict Access Based on Attributes (ABAC). Select “IAM” option and click edit button on just created service account name.

Click on IAM condition (optional) button and create own logic to restrict access. I provide access to service account until 31th December, then it will not available. Click save.

Add condition

Principal	Project
service-account-for-assignment@cloudappdev-assignment-4.iam.gserviceaccount.com	CloudAppDev-Assignment-4

CONDITION BUILDER

CONDITION EDITOR

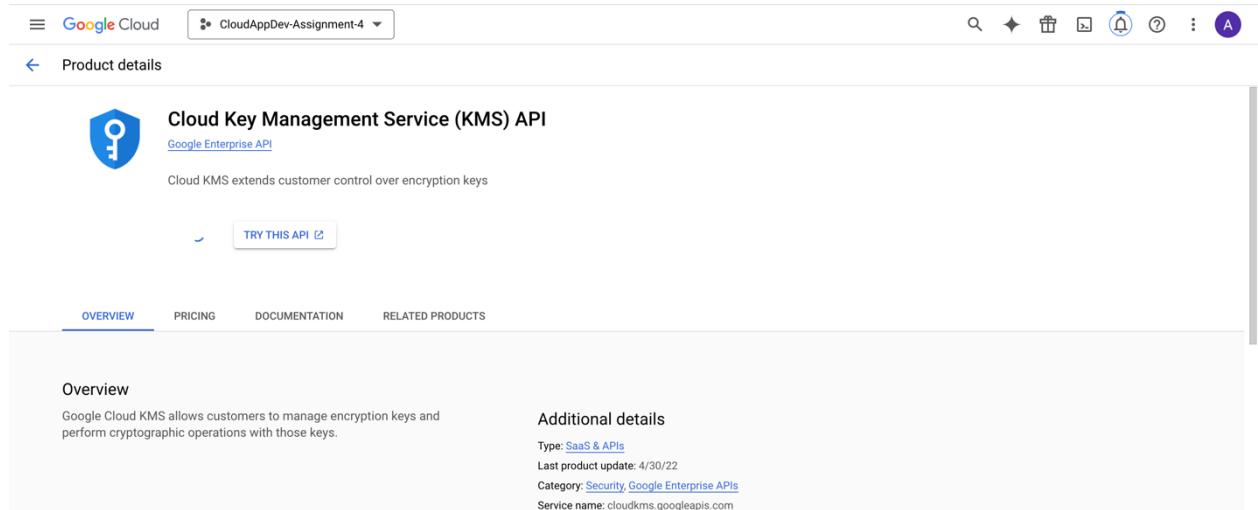
Condition type 1
Service
Operator is
Resource Service * storage.googleapis.com

Condition type 2
Expiring Access
Operator by
Time * 12/31/2024, 12:00 AM NPT

Buttons: AND, OR, ADD, SAVE, CANCEL

2.3. Data Protection

We will use **Google Cloud KMS** to set up encryption for data at rest. First we need to enable Cloud KMS API.

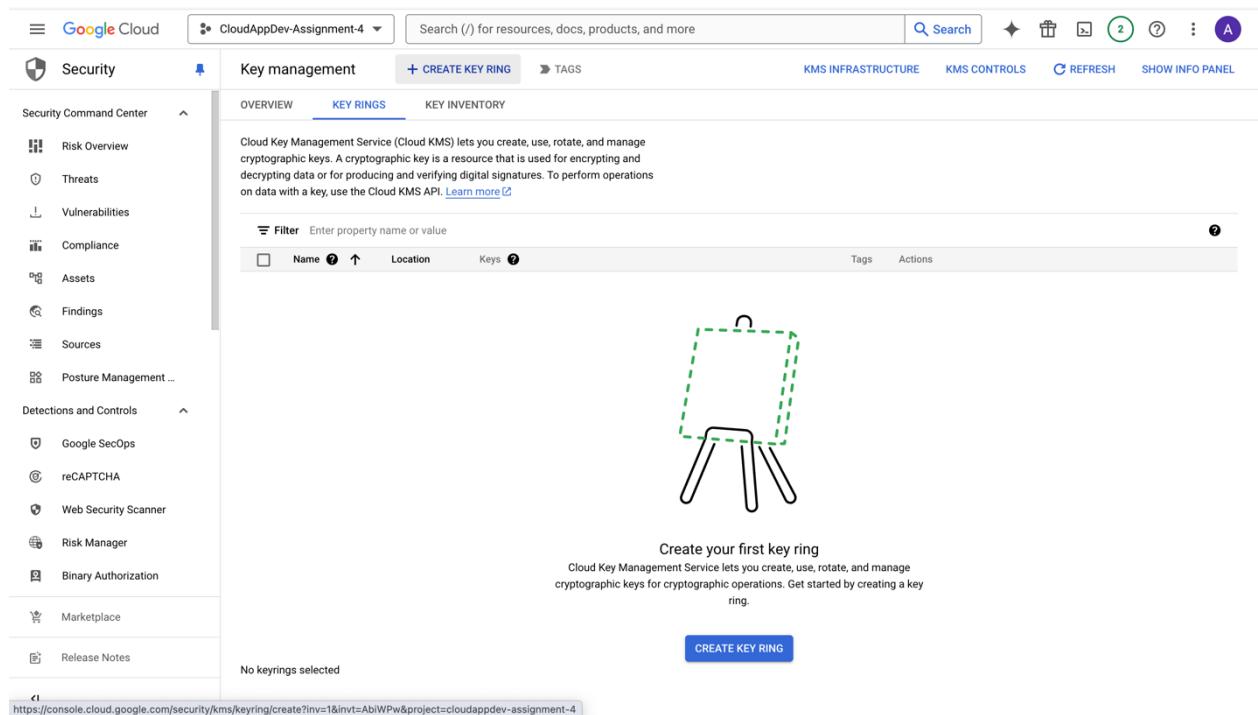


The screenshot shows the Google Cloud Cloud Key Management Service (KMS) API page. At the top, there is a navigation bar with 'Google Cloud' and a dropdown showing 'CloudAppDev-Assignment-4'. To the right are search, filter, and other navigation icons. Below the navigation is a back arrow labeled 'Product details'. The main content area features a blue shield icon with a key symbol, followed by the text 'Cloud Key Management Service (KMS) API' and 'Google Enterprise API'. A subtext states 'Cloud KMS extends customer control over encryption keys'. Below this is a 'TRY THIS API' button. At the bottom of the main content are tabs for 'OVERVIEW' (which is selected), 'PRICING', 'DOCUMENTATION', and 'RELATED PRODUCTS'.

Overview
Google Cloud KMS allows customers to manage encryption keys and perform cryptographic operations with those keys.

Additional details
Type: SaaS & APIs
Last product update: 4/30/22
Category: Security, Google Enterprise APIs
Service name: cloudkms.googleapis.com

Select here “CREATE KEY RING”



The screenshot shows the Google Cloud Security Command Center Key management page. The left sidebar has sections like Security Command Center, Risk Overview, Threats, Vulnerabilities, Compliance, Assets, Findings, Sources, and Posture Management. Under Detections and Controls, there are Google SecOps, reCAPTCHA, Web Security Scanner, Risk Manager, Binary Authorization, Marketplace, and Release Notes. The main content area has tabs for OVERVIEW, KEY RINGS (which is selected), and KEY INVENTORY. It includes a brief description of Cloud Key Management Service and a 'CREATE KEY RING' button. Below the button is a section titled 'Create your first key ring' with a subtext about creating a key ring. At the bottom, it says 'No keyrings selected'.

Here write name of ring and select region.

The screenshot shows the Google Cloud Security Command Center interface. On the left, there's a sidebar with various security modules like Security Command Center, Risk Overview, Threats, Vulnerabilities, Compliance, Assets, Findings, Sources, and Posture Management. The main area is titled "Create key ring". It has a brief description about key rings, a "Project name" field set to "cloudappdev-assignment-4", a "Key ring name" field containing "assignemnt4", a "Location type" section with "Region" selected (lower latency within a single region), and a "Region" dropdown set to "us-central1 (Iowa)". At the bottom are "CREATE" and "CANCEL" buttons.

Then we can create encryption key

The screenshot shows the "Create key" page within the Google Cloud Security Command Center. The sidebar includes modules like Security Command Center, Risk Overview, Threats, Vulnerabilities, Compliance, Assets, Findings, Sources, and Posture Management. The main form is titled "Create key" and contains several sections: "Name and protection level" (selected), "Key material" (selected), "Purpose and algorithm" (selected), "Versions" (selected), and "Additional settings (optional)". Under "Additional settings", there's a field for "Duration of 'scheduled for destruction' state" set to "90 days". To the right, a "Key creation details" panel shows "Project name: cloudappdev-assignment-4", "Location: us-central1", and "Key ring: assignemnt4". At the bottom are "CREATE" and "CANCEL" buttons.

Use HTTPS for data in transit and configure load balancers to enforce SSL.
To use https is necessary to create SSL certificate. Go to security page and “Certificate

Manager” section and click on button “CREATE SSL CERTIFICATE”.

The screenshot shows the Google Cloud Security interface. On the left, there's a sidebar with 'Security' selected. Under 'Certificate Manager', the 'CLASSIC CERTIFICATES' tab is active. A large dashed cloud icon is centered on the page. Below it, a message says 'No SSL certificates have been created.' with a link to 'Learn more'.

Fill the necessary fields.

The screenshot shows the 'Create certificate' form. In the 'Name' field, 'assignment4' is entered. The 'Description' field contains 'SSL Certificate for assignment4'. Under 'Location', 'Global' is selected. In the 'Scope' section, 'Default' is chosen. Under 'Certificate type', 'Create Self-managed certificate' is selected. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

Where Domain name write the application domain name. For me it is `cloudappdev-assignment-4.uc.r.appspot.com`

Google Cloud CloudAppDev-Assignment-4 Search (S) for resources, docs, products, and more

Security

Create certificate

Location
Certificate can be used globally or regionally.
 Global
 Regional

Scope
Key distribution scope defines on which data center cert will be deployed.
Default

Certificate type
You can have Google issue and manage the certificate or you can upload a certificate issued by a third-party.
 Create Self-managed certificate
 Create Google-managed certificate

Certificate Authority type
You can have Google issue and manage the certificate or you can upload a certificate issued by a third-party.
 Public
 Private

Domain names *
Use the input box to specify your domain name(s), use comma to separate different ones.
cloudappdev-assignment-4.uc.r.appspot.com

Authorization type
 DNS Authorization
 Load Balancer Authorization

Labels
+ ADD LABEL

CREATE CANCEL

The next is **configure Load Balancer** for my application. Go to search and find Network Services – Load balancing. Click on “CREATE LOAD BALANCER”.

Google Cloud CloudAppDev-Assignment-4 load bal Search (S) X

Network Services

Load balancing

+ CREATE LOAD BALANCER REFRESH DELETE LEARN

Load balancing

Cloud DNS Cloud CDN Cloud NAT Cloud Service Mesh (Traffic ...) Service Directory Cloud Domains Private Service Connect SSL policies Service Extensions Marketplace Release Notes

Get started with real-time analytics

Use Network Intelligence Center for comprehensive monitoring and troubleshooting. [Learn more](#)

- ✓ Visualize your network resources
- ✓ Diagnose and prevent connectivity issues
- ✓ View packet loss and latency metrics
- ✓ Keep your firewall rules strict and efficient

TRY NOW REMIND ME LATER

LOAD BALANCERS	BACKENDS	FRONTENDS	LB POLICIES
Filter Enter property name or value			
Name	Load balancer type	Access type	Protocols Region Backends

Load balancing

The configuration for our load balancer.

Google Cloud CloudAppDev-Assignment-4 load balancer

Create a load balancer

Type of load balancer
Application Load Balancer

Public facing or internal
Public facing (external)

Global or single region deployment
Global workloads

Load balancer generation
Global external Application Load Balancer

Create load balancer

You are about to create an Application Load Balancer with following features:

- Public facing (external)
- Global

CONFIGURE CANCEL

On frontend write name and select before created SSL certificate

Network Services Create global external Application Load Balancer

Load balancing

Frontend configuration

Backend configuration

Review and finalize (optional)

New Frontend IP and port

Name: frontend-balancer
Description:
Protocol: HTTPS (includes HTTP/2 and HTTP/3)
Network Service Tier: Premium
IP version: IPv4
IP address: Ephemeral
Port: 443
Certificate: assignment-4

ADDITIONAL CERTIFICATES

SSL policy: GCP default
HTTP/2 (QUIC) negotiation:

2.4. Security Testing

We need to add a security scanning tool to the CI/CD pipeline. For that I will use GitHub Actions with OWASP ZAP tool. When we make commits, the tool will analyze the code and create a report for us.

For that go to your git repository. Select actions and find by search OWASP. Click configure and continue

The screenshot shows a GitHub repository page for 'Cloud-Application-Development'. The 'Actions' tab is selected. A search bar at the top contains the query 'owasp'. Below the search bar, there is a message: 'Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.' There is also a link to 'Skip this and set up a workflow yourself →'. On the left, there is a sidebar with categories: Deployment, Security, Continuous integration, Automation, and Pages. The main area displays a single workflow card titled 'EthicalCheck' by APISec. The card describes the service as providing free & automated API security testing. It includes a 'Configure' button and a 'Code scanning' button with a blue dot indicating it is active. At the bottom of the page, there is a footer with links to GitHub terms, privacy, security, status, docs, contact, manage cookies, and a 'Do not share my personal information' option.

Used configuration for that:

```
Assignment-4 > 📄 owasp-zap.yml
1   name: OWASP ZAP
2
3   on:
4     push:
5       branches:
6         - master
7
8   jobs:
9     zap_scan:
10    runs-on: ubuntu-latest
11
12   steps:
13     - name: Checkout code
14       uses: actions/checkout@v2
15
16     - name: Set up OWASP ZAP
17       uses: zaproxy/action-full-scan@v0.1.0
18       with:
19         target: "https://cloudappdev-assignment-4.uc.r.appspot.com"
20         docker_name: "owasp/zap2docker-stable"
21         cmd_options: "-r zap_report.html"
22
23     - name: Upload OWASP ZAP Report
24       uses: actions/upload-artifact@v3
25       with:
26         name: zap-report
27         path: zap_report.html
```

With new commit it shows github workflow that scan our application.

The screenshot shows the GitHub Actions page for the repository 'altair2503 / Cloud-Application-Development'. The 'Actions' tab is selected. On the left, there's a sidebar with sections for Actions, Management (Caches, Attestations, Runners), and a link to 'EthicalCheck-Workflow'. The main area displays 'All workflows' with a heading 'All workflow runs'. There are two runs listed:

- test pipeline**: Status: Queued, Started: now, Last updated: 6 minutes ago, Actor: ...
Event: Commit 25e4a14 pushed by altair2503
- Create ethicalcheck.yml**: Status: Queued, Started: 6 minutes ago, Last updated: 1m 12s, Actor: ...
Event: Commit 74ca19e pushed by altair2503

Now when we do commit it generates report in github actions. The report itself:

Risk Level	Number of Alerts
High	0
Medium	0
Low	2
Informational	4
False Positives:	0

Alerts			
Name	Risk Level	Number of Instances	
Content Security Policy (CSP) Header Not Set	Medium	0	
Missing Anti-clickjacking Header	Medium	0	
Permissions Policy Header Not Set	Low	0	
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	0	
X-Content-Type-Options Header Missing	Low	2	
Storable and Cacheable Content	Informational	4	

2.5. Monitoring and Logging

First we need to enable Google Cloud Audit Logs for my project. To do that search logs explorer and activate it. If you did `gcloud app deploy` (as I did before) it will automatically enable logging for you applicatin. Below logs explorer for my project.

Severity	Time	Summary
> i	2024-11-24 22:08:17.938	compute.googleapis.com v1.compute.securityPolicies.insert _cies/default-security-policy-for-backend altairkabdrakhmanov@gmail..
> i	2024-11-24 22:08:21.629	compute.googleapis.com .beta.compute.backendServices.insert _a1/backendServices/load-balancer-backend altairkabdrakhmanov@gmail..
> i	2024-11-24 22:08:41.444	compute.googleapis.com .beta.compute.backendServices.insert _a1/backendServices/load-balancer-backend altairkabdrakhmanov@gmail..
> i	2024-11-24 22:08:44.893	compute.googleapis.com .beta.compute.backendServices.setSecurityPolicy _a1/backendServices/load-balancer-backend altairkabdrakhmanov@gmail..
> i	2024-11-24 22:08:53.568	compute.googleapis.com .beta.backendServices.setSecurityPolicy _a1/backendServices/load-balancer-backend altairkabdrakhmanov@gmail..
> i	2024-11-24 22:08:57.021	compute.googleapis.com v1.compute.urlMaps.insert _global:urlMaps/load-balancer-assignment4 altairkabdrakhmanov@gmail.. audit_-
> i	2024-11-24 22:09:02.779	compute.googleapis.com v1.compute.urlMaps.insert _global:urlMaps/load-balancer-assignment4 altairkabdrakhmanov@gmail.. audit_-
> i	2024-11-24 22:09:05.179	compute.googleapis.com .l.compute.targetHttpsProxies.insert _/load-balancer-assignment4-target-proxy altairkabdrakhmanov@gmail..
> i	2024-11-24 22:09:07.285	compute.googleapis.com .compute.globalForwardingRules.insert _global/forwardingRules/frontend-balancer altairkabdrakhmanov@gmail..
> i	2024-11-24 22:09:25.257	compute.googleapis.com .compute.globalForwardingRules.insert _global/forwardingRules/frontend-balancer altairkabdrakhmanov@gmail..
> i	2024-11-24 22:11:26.998	2024-11-24 16:26:26.989 UTC [1825]: [-1] db=,user= LOG: automatic vacuum of table "cloudsqladmin.public.heartbeat": index scans: 0 -
> i	2024-11-24 22:11:26.993	2024-11-24 16:26:26.992 UTC [1825]: [-1] db=,user= LOG: automatic analyze of table "cloudsqladmin.public.heartbeat" avg read rate: -
> i	2024-11-24 22:12:04.168	2024-11-24 16:27:04.167 UTC [18]: [41-1] db=,user= LOG: checkpoint starting: time
> i	2024-11-24 22:12:04.499	2024-11-24 16:27:04.499 UTC [18]: [42-1] db=,user= LOG: checkpoint complete: wrote 4 buffers (0.0%); 0 WAL file(s) added, 0 removed, -

The next is to set up alerts using Google Cloud Monitoring based on specific security events. To do that search policies minotring and create alert. For my case I create an alert for memory

usage.

The screenshot shows the 'Create alerting policy' interface for a GAE Application - Memory usage metric. The left sidebar lists 'ALERT CONDITIONS' (GAE Application - Memory usage selected) and 'ALERT DETAILS' (Notifications and name, Review alert). The main area shows 'Policy configuration mode' (Builder selected), 'Select a metric' (GAE Application - Memory usage), and 'Add filters' (optional). A chart displays memory usage over time, showing a sharp drop from 150MB to 50MB at 9:00 PM UTC+5. A filter panel shows 'Metric ↑ usage' with a value of 0. Estimated monthly cost is listed as '\$1.53 \$0.00'. Buttons at the bottom include 'Create Policy', 'Provide feedback', and 'Cancel'.

Set alert to 1GB usage.

The screenshot shows the 'Create alerting policy' interface for a GAE Application - Memory usage metric. The left sidebar lists 'ALERT CONDITIONS' (GAE Application - Memory usage selected, with a note: 'Some form fields are incorrect') and 'ALERT DETAILS' (Review alert). The main area shows 'Configure alert trigger' under 'Condition Types' (Threshold selected). It shows a chart with a red dashed line at 1.000MB and a blue line dropping to 953.674MB at 9:00 PM UTC+5. A filter panel shows 'Metric ↑ usage' with a value of 0. Estimated monthly cost is listed as '\$1.53 \$0.00'. Buttons at the bottom include 'Create Policy', 'Provide feedback', and 'Cancel'.

As notification channel I chose my email.

Google Cloud CloudAppDev-Assignment-4 policies

[Create alerting policy](#) [+ Add alert condition](#) [Delete alert condition](#)

ALERT CONDITIONS

- GAE Application - Memory usage
- Configure trigger

ALERT DETAILS

- Notifications and name
- Review alert

Configure notifications and finalize alert

Configure notifications Recommended

Use notification channel

Notification Channels [my email](#)

Notification subject line [Alert memory usage](#)

We recommend that you create multiple notification channels for redundancy purposes. Google has no control of many of the delivery systems after we have passed the notification to that system. Additionally, a single Google service supports Cloud Console Mobile App, PagerDuty, Webhooks, and Slack. If you use one of these notification channels, then use email, SMS, or Pub/Sub as the redundant channel.

[Learn more](#)

Notify on incident closure

Incident autoclose duration [1 hour](#) [6 hours](#) [1 day](#) [1 week](#) [1 month](#) [6 weeks](#)

If data is absent, select a duration after which Incident will automatically close.

Policy user labels Recommended

Policy user labels allow you to add your own labels to alert policies for organization. The labels are included in the notification and incident details.

As result here my “Memory usage” alert.

Google Cloud CloudAppDev-Assignment-4 policies

[Search](#) [1 hour](#) [6 hours](#) [1 day](#) [1 week](#) [1 month](#) [6 weeks](#)

Observability Monitoring

- Overview
- Dashboards
- Metrics explorer
- Logs explorer
- Log analytics
- Trace explorer
- Alerting
- Error reporting
- Uptime checks
- Synthetic monitoring
- SLOs

Configure

- Integrations

Observability Scopes
CloudAppDev-Assignment- > 4

Memory usage

Conditions **Severity**
Policy violates when ANY condition is met No severity

GAE Application - Memory usage

Condition type	Triggers when	Threshold position	Threshold value	Retest window
Threshold	Any time series cross threshold	Above threshold	1000000000 B	No retest

Filter Enter property name or value

Metric ↑ usage

Alert policy Memory usage saved

2.6. Incident Response

Incident response is the common practice for organization to respond to critical failures of the system. It can be referred as part of Business Continuity Plan (BCP). The process of identifying, managing, and resolving security breaches is very important to manage system availability. The key components of a response plan include:

1. **Preparation:** Document response procedures and ensure all team members are trained.
2. **Detection and Analysis:** Use tools like Google Cloud Monitoring and Audit Logs to identify unusual activities.
3. **Containment:** Isolate the affected systems to prevent the spread of the breach.
4. **Eradication:** Identify and remove the cause of the incident, such as malicious code or compromised accounts.
5. **Recovery:** Restore normal operations by patching vulnerabilities and verifying system integrity.
6. **Post-Incident Review:** Conduct a review to analyze the cause of the incident, evaluate the response, and implement improvements.

To demonstrate incident response, simulate a security incident by triggering a fake vulnerability. After that make all steps described above as training of incident response.

3. Scaling Applications on Google Cloud

3.1. Overview of Scalability

Scalability is the ability of a system to handle more work or support more users when needed. It is important because it helps applications run smoothly even when there are more requests or higher demand.

There are two main types of scalability:

1. Horizontal Scalability means adding more machines or instances to share the workload. It is good for applications that do not keep data on the instance (stateless), so any instance can handle any request. Benefits: Better performance, higher availability, and less chance of failure.
2. Vertical Scalability means increasing the power of one machine, like adding more CPU or memory. It works well for applications that keep important data in the instance (stateful). Benefits: Simpler to set up, but there is a limit to how much you can upgrade one machine.

Both methods can be used together depending on the needs of the application. In cloud systems like Google Cloud, tools like auto-scaling can make scaling automatic, saving both time and effort.

3.2. Application Design

For this purpose it is used simple phonebook application written in Flask framework.

With CRUD operations:

```
11  # Create a new contact
12  @app.route('/contacts', methods=['POST'])
13  def create_contact():
14      global next_contact_id
15      contact_data = request.json
16
17      # Proceed to create the contact if validation succeeds
18      contact = {
19          'id': next_contact_id,
20          'name': contact_data['name'],
21          'phone': contact_data['phone']
22      }
23      phonebook[next_contact_id] = contact
24      next_contact_id += 1 # Increment the ID for the next contact
25
26
27      return jsonify(contact), 201
28
29  # Read all contacts
30  @app.route('/contacts', methods=['GET'])
31  def get_contacts():
32      return jsonify(list(phonebook.values())), 200
33
34  # Read a specific contact by ID
35  @app.route('/contacts/<int:contact_id>', methods=['GET'])
36  def get_contact(contact_id):
37      contact = phonebook.get(contact_id)
38      if not contact:
39          return jsonify({'message': 'Contact not found'}), 404
40      return jsonify(contact), 200
41
42  # Update a contact
43  @app.route('/contacts/<int:contact_id>', methods=['PUT'])
44  def update_contact(contact_id):
45      contact = phonebook.get(contact_id)
46      if not contact:
47          return jsonify({'message': 'Contact not found'}), 404
48      updated_data = request.json
49      contact['name'] = updated_data.get('name', contact['name'])
50      contact['phone'] = updated_data.get('phone', contact['phone'])
51
52      return jsonify(contact), 200
53
54  # Delete a contact
55  @app.route('/contacts/<int:contact_id>', methods=['DELETE'])
56  def delete_contact(contact_id):
57      contact = phonebook.pop(contact_id, None)
58      if not contact:
59          return jsonify({'message': 'Contact not found'}), 404
60
61      return jsonify({'message': 'Contact deleted'}), 200
62
```

Also it is used Google Kubernetes Engine that supports containerizing and autoscaling.
But for we need to create image of application and enable GKE. So to do that, I create

Dockerfile:

```
Assignment-4 > Dockerfile
1  # Use the official Python image as a base
2  FROM python:3.9-slim
3
4  # Set the working directory inside the container
5  WORKDIR /app
6
7  # Copy the requirements file into the container
8  COPY requirements.txt .
9
10 # Install the required packages
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Copy the rest of the application code into the container
14 COPY main.py .
15
16 # Expose the port the app runs on
17 EXPOSE 8080
18
19 # Command to run the application using gunicorn
20 CMD ["gunicorn", "--bind", "0.0.0.0:8080", "main:app"]
21
```

Build the image: docker build -t phonebook-app .

```
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS zsh - As
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 421 bytes | 421.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/altair2503/Cloud-Application-Development
  25e4a14..ae44a5f main -> main
(base) kabdrakhman@MacBook-Pro-Altair Cloud-Application-Development % cd Assignment-4
(base) kabdrakhman@MacBook-Pro-Altair Assignment-4 % docker build -t phonebook-app .
[+] Building 20.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 548B
=> [internal] load .dockerrignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397
=> => resolve docker.io/library/python:3.9-slim@sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397
=> => sha256:193e8340ba1de799bd9476ee14a9c4e24d0f51ccfe4c57c82bb7e6f7f807526e 14.83MB / 14.83MB
=> => sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397085d 10.41kB / 10.41kB
=> => sha256:da365f61478902de7be5ff083369c8c38be5a9cc924b1c6504bdbcd497d9e60c3 1.75kB / 1.75kB
=> => sha256:d5a37ad066668ba1af6d9691600454b12a51f1c92f19858dfda2b00c48d579c2 5.43kB / 5.43kB
=> => sha256:6d29a096dd42e5e003949f934fa6b1a3ec8e076dd8cfc2a85a4e750a3639bf7a 29.16MB / 29.16MB
=> => sha256:6fab32a80202b33cfda04522829fe2e08e9c3a1a4a548a71a09d35720709a5ba 3.33MB / 3.33MB
=> => sha256:7e9a72ce45a2f47cd2b49aa49edba5d1d4c3b2fe18309912991d48b62ebccaa8 249B / 249B
=> => extracting sha256:6d29a096dd42e5e003949f934fa6b1a3ec8e076dd8cfc2a85a4e750a3639bf7a
=> => extracting sha256:6fab32a80202b33cfda04522829fe2e08e9c3a1a4a548a71a09d35720709a5ba
=> => extracting sha256:193e8340ba1de799bd9476ee14a9c4e24d0f51ccfe4c57c82bb7e6f7f807526e
=> => extracting sha256:7e9a72ce45a2f47cd2b49aa49edba5d1d4c3b2fe18309912991d48b62ebccaa8
=> [internal] load build context
=> => transferring context: 2.15kB
=> [2/5] WORKDIR /app
=> [3/5] COPY requirements.txt .
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> [5/5] COPY main.py .
=> exporting to image
=> => exporting layers
=> => writing image sha256:224a2aa191967be15ded94ca38f248f45393d5ce57542ec8e37202838f4584c2
=> => naming to docker.io/library/phonebook-app

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
(base) kabdrakhman@MacBook-Pro-Altair Assignment-4 %
```

Ln 7, Col 44 Spaces: 4

Prior pushing image to GCR, I tested locally with this command: `docker run -p 8080:8080 phonebook-app`

```
○ (base) kabdrakhman@MacBook-Pro-Altair Assignment-4 % docker run -p 8080:8080 phonebook-app
[2024-11-24 16:50:55 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-11-24 16:50:55 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2024-11-24 16:50:55 +0000] [1] [INFO] Using worker: sync
[2024-11-24 16:50:55 +0000] [7] [INFO] Booting worker with pid: 7
```

Ln 13, Col 17

To have permission to add images into your remote docker registry on google cloud you need to execute this command: `gcloud auth configure-docker`

To push my docker images I used this command to tag phonebook with my google cloud registry (GCR): `docker tag phonebook-app gcr.io/cloudappdev-assignment-4/phonebook-app:latest`
Only then I could push image to GCR `docker push gcr.io/ cloudappdev-assignment-4/phonebook-app:latest`

```
getcloud credentials: [REDACTED] already registered correctly.
● (base) kabdrakhman@MacBook-Pro-Altair Assignment-4 % docker tag phonebook-app gcr.io/cloudappdev-assignment-4/phonebook-app:latest
● (base) kabdrakhman@MacBook-Pro-Altair Assignment-4 % docker push gcr.io/cloudappdev-assignment-4/phonebook-app:latest
The push refers to repository [gcr.io/cloudappdev-assignment-4/phonebook-app]
94b63cbd4acd: Pushed
6f4c54009c8c: Pushed
33b038fbdc5c: Pushed
2d032a168907: Pushed
00490679d83e: Layer already exists
8dc6549c207d: Layer already exists
678d187bf5a8: Layer already exists
077584c0c75a: Layer already exists
latest: digest: sha256:54ad10b9f070e309d4cc5403cea9a0903cb38808232883cace3b8d53761f0a67 size: 1990
○ (base) kabdrakhman@MacBook-Pro-Altair Assignment-4 %
```

Ln 13, Col 17 Spaces: 4 UTF-8 LF Plain Text ⓘ

To create Kubernetes enable api - Kubernetes Engine API

The screenshot shows the Google Cloud Platform interface. At the top, there is a navigation bar with 'Google Cloud' and a dropdown menu labeled 'CloudAppDev-Assignment-4'. Below the navigation bar, there is a back arrow and the text 'Product details'. The main content area features a blue hexagonal icon representing the Kubernetes Engine API. The title 'Kubernetes Engine API' is displayed in large blue text, with 'Google Enterprise API' in smaller blue text below it. A description states: 'Builds and manages container-based applications, powered by the open source Kubernetes technology.' Below the description are three buttons: 'MANAGE' (blue), 'TRY THIS API' (white with blue text), and 'API Enabled' (white with green checkmark). At the bottom of the page, there are tabs for 'OVERVIEW' (which is underlined in blue), 'DOCUMENTATION', and 'RELATED PRODUCTS'. The 'OVERVIEW' section contains a 'Overview' heading, a brief description, and a 'Additional details' button.

and to create cluster used code:

```
gcloud container clusters create assignment4-cluster \
--zone us-central1-a \
--num-nodes 1
```

To enable kubectl for my cluster used command:

```
gcloud container clusters get-credentials assignment4-cluster --zone us-central1-a
```

3.3. Scaling Methods

We have two types of scaling methods is to use horizontal or vertical scaling. The difference is that:

- Horizontal scaling is used when we increase number of instance. It is preferred to use when the application is stateless and requires high availability or load distribution across multiple instances. That means it doesn't share its states with other apps. The implementation with 3 instances

```
1  apiVersion: Development/.github/workflows/ethicalcheck.yml
2  kind: Deployment
3  metadata:
4    name: phonebook-app
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: phonebook-app
10   template:
11     metadata:
12       labels:
13         app: phonebook-app
14     spec:
15       containers:
16         - name: phonebook-app
17           image: gcr.io/cloudappdev-assignment-4/phonebook-app:latest
18         ports:
19           - containerPort: 8080
20
```

- Vertical scaling is used when we increase capacity of our machine. It is preferred to use when the application is stateful or requires more resources than a single instance can handle. That means when instance saves its state. The implementation with high number of memory and CPUs.

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: phonebook-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: phonebook-app
10   template:
11     metadata:
12       labels:
13         app: phonebook-app
14   spec:
15     containers:
16       - name: phonebook-app
17         image: gcr.io/cloudappdev-assignment-4/phonebook-app:latest
18         ports:
19           - containerPort: 8080
20         resources:
21           requests:
22             cpu: "500m"
23             memory: "256Mi" |
24

```

3.4. Load Balancing

To enable load balancing it was created service instance attached to my pod with load balancing method. Here is my service:

```

24
25  apiVersion: v1
26  kind: Service
27  metadata:
28    name: phonebook-app-service
29  spec:
30    type: LoadBalancer
31    selector:
32      app: phonebook-app
33    ports:
34      - protocol: TCP
35        port: 80
36        targetPort: 8080

```

Also it is important to load traffic to working pods. For that it is necessary to have liveness and readiness implementation. To app was added:

```

7   next_contact_id = 1 # Start with ID 1
8
9   # Health Check Endpoints
10
11  # Liveness check endpoint
12  @app.route('/health', methods=['GET'])
13  def healthz():
14      # Just returns a 200 OK if the app is alive
15      return jsonify({"status": "ok"}), 200
16
17  # Readiness check endpoint
18  @app.route('/readiness', methods=['GET'])
19  def readiness():
20      # Check if phonebook is empty, which might be a simple readiness check (or use other logic)
21      if len(phonebook) >= 0: # You could make this check more complex if needed
22          return jsonify({"status": "ready"}), 200
23      else:
24          return jsonify({"status": "not ready"}), 503
25

```

And to main pod was added liveness and readiness fields:

```

17      image: gcr.io/cloudappdev-assignment-4/phonebook-app:latest
18
19      ports:
20          - containerPort: 8080
21
22      resources:
23          requests:
24              cpu: "500m" # Request 500m CPU for the container
25              memory: "256Mi" # Request 256Mi memory for the container
26
27      # Health checks (probes)
28      livenessProbe:
29          httpGet:
30              path: /health # Adjust the path based on your app's health endpoint
31              port: 8080
32              initialDelaySeconds: 10 # Time to wait before the first probe
33              periodSeconds: 5 # Time between subsequent probes
34              failureThreshold: 3 # Number of consecutive failures before marking the pod as unhealthy
35
36      readinessProbe:
37          httpGet:
38              path: /readiness # Adjust the path based on your app's readiness endpoint
39              port: 8080
40              initialDelaySeconds: 5 # Time to wait before starting readiness checks
41              periodSeconds: 5 # Time between subsequent probes
42              failureThreshold: 3 # Number of consecutive failures before stopping traffic to the pod

```

3.5. Auto-Scaling Implementation

To enable horizontal auto scaling it was defined HorizontalPodAutoscaler that will increase the number of instances if cpu usage exceeds more than 50 percents.

```
34  apiVersion: autoscaling/v2
35  kind: HorizontalPodAutoscaler
36  metadata:
37    name: phonebook-app-hpa
38    namespace: default
39  spec:
40    scaleTargetRef:
41      apiVersion: apps/v1
42      kind: Deployment
43      name: phonebook-app
44    minReplicas: 1 # Minimum number of replicas
45    maxReplicas: 10 # Maximum number of replicas
46    metrics:
47      - type: Resource
48        resource:
49          name: cpu
50          target:
51            type: Utilization
52            averageUtilization: 50 # Scale when average CPU usage exceeds 50%
```

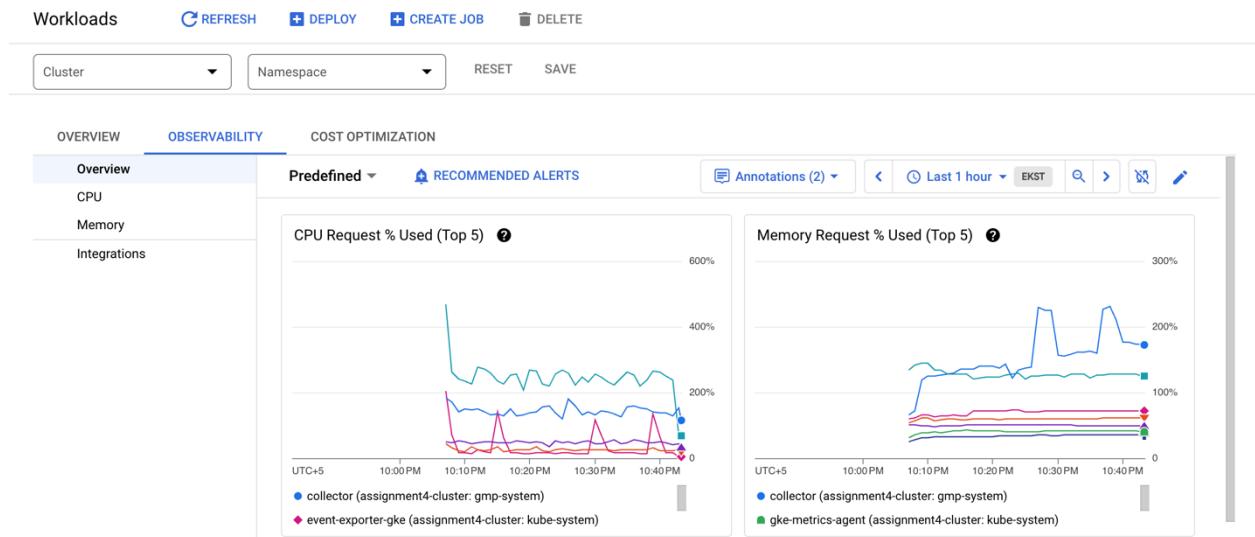
For vertical autoscaling it was added limits field to enable scaling memory and CPUs for my instance. It means it will use more than requested cpu and memory

Assignment-4 > phonebook.yaml

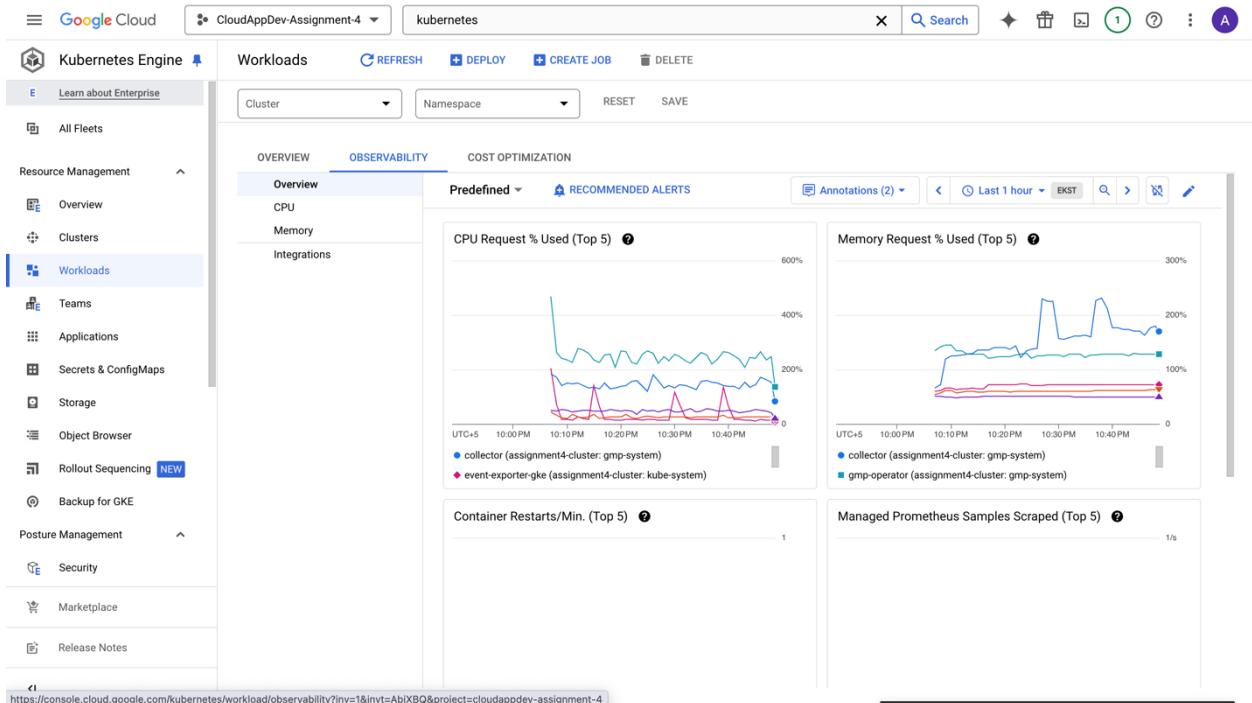
```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: phonebook-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: phonebook-app
10   template:
11     metadata:
12       labels:
13         app: phonebook-app
14     spec:
15       containers:
16         - name: phonebook-app
17           image: gcr.io/cloudappdev-assignment-4/phonebook-app:latest
18           ports:
19             - containerPort: 8080
20           resources:
21             requests:
22               cpu: "500m" # Request 500m CPU for the container
23               memory: "256Mi" # Request 256Mi memory for the container
24             limits:
25               cpu: "1" # Limit the container to 1 CPU
26               memory: "512Mi" # Limit the container to 512Mi memory
27 ---
```

3.6. Performance Monitoring

GKE automatically enable Google Cloud Monirong tool for my cluster that monitor CPU and memory usage:



It can be found in workloads tab, observability option.



3.7. Cost Optimization Strategies

Cost optimization means using resources in a smart way to reduce expenses while keeping good performance. Here are some tips to save money on Google Cloud:

1. **Use the Right Resources:** Check how much CPU, memory, and storage your app uses and adjust instance size to avoid oversized resources.
2. **Auto-Scaling:** Set up auto-scaling to adjust resources based on demand and avoid wasting money during low-usage times.
3. **Storage Optimization:** Choose the right storage class—Standard for frequent use, Coldline for rare use, and Archive for long-term—and automate moving old data to cheaper storage.
4. **Serverless and Managed Services:** Use serverless options like Cloud Functions to pay only for usage and managed services like BigQuery for automatic scaling.
5. **Monitor Usage:** Use Google Cloud Monitoring to track costs and resource usage, and set alerts for unexpected cost increases.
6. **Optimize Networking:** Keep resources in the same region to reduce data transfer costs and use Cloud CDN to cache content closer to users.
7. **Remove Unused Resources:** Identify and delete unused instances, disks, or IP addresses, and turn off non-essential resources during off-hours.
8. **Review and Automate:** Regularly review your cloud costs and usage, and use tools like Google Cloud Recommender to automate savings.

We can reduce costs while keeping our application scalable and reliable. It helps make cloud resources more efficient and avoids spending money on unnecessary items.

4. Conclusion

In this report, we have discussed the best practices for improving security and scalability of applications on Google Cloud. We learned that using tools like IAM for access control, Google Cloud KMS for data encryption, and HTTPS for secure data transfer are important for

keeping applications safe. Security testing and continuous monitoring also help identify and fix issues quickly.

For scalability, we learnt horizontal and vertical scaling methods to manage user traffic and ensure the app works smoothly. Using load balancing and auto-scaling, we can make the app perform well even during high traffic. Cost optimization strategies also help reduce unnecessary spending while maintaining performance.

5. Recommendations

After completing the assignment I come up to the following recommendations:

1. **IAM Roles:** By using the principle of least privilege it makes only the necessary permissions are granted to users and service accounts. Reducing the risk of vulnerability.
2. **Data Protection:** Data encryption is important for maintaining confidentiality and integrity. Always encrypt sensitive data at rest using tools like Google Cloud Key Management Service (KMS).
3. **Secure Network Protocols:** Implement SSL certificates and configure the load balancer make possible that all data in transition between clients and the server is securely encrypted.
4. **Load Balancers:** Load balancing is essential for distributing network traffic efficiently across application, makes the application's availability and reliability.
5. **Auto-Scaling:** Both horizontal and vertical scaling strategies should be used depending on the application's architecture ensuring zero-down time for our app.
6. **Monitoring and Alerting:** Use Google Cloud's monitoring and logging tools to continuously track application performance and security events.
7. **Security and Performance Reviews:** Using tools like OWASP ZAP for security testing and Google Cloud Monitoring for tracking resource usage makes avoiding of issues in production

With these recommendations, the application will have enhanced security, scalability, and cost optimization, ensuring it is resilient to both current and future demands on Google Cloud.

6. References

- IAM documentation – <https://cloud.google.com/iam/docs/>
- Google KMS - <https://cloud.google.com/kms/docs>
- Load Balancer - <https://cloud.google.com/load-balancing/docs>
- Kubernetes - <https://cloud.google.com/kubernetes-engine/docs>
- Horizontal Autoscaling - <https://cloud.google.com/kubernetes-engine/docs/how-to/scaling-apps>
- Alerting - <https://cloud.google.com/retail/docs/monitor>
- OWASP ZAP - <https://github.com/zaproxy/zaproxy>

7. Appendices

Project structure

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure under "CLOUD-APPLICATION-DEVELOPMENT".
 - Assignment-4 folder contains:
 - venv
 - ethicalcheck.yml
 - Dockerfile
 - app.yaml
 - main.py
 - owasp-zap.yml
 - phonebook.yaml
 - requirements.txt
 - Midterm folder.
- Editor:** The main editor window displays the "main.py" file content. The code is a Flask application for a phonebook API.

```
from flask import Flask, request, jsonify
app = Flask(__name__)
# In-memory phonebook and ID counter
phonebook = {}

# Liveness check endpoint
@app.route('/health', methods=['GET'])
def health():
    # Just returns a 200 OK if the app is alive
    return jsonify({"status": "ok"}), 200

# Readiness check endpoint
@app.route('/readiness', methods=['GET'])
def readiness():
    # Check if phonebook is empty, which might be a simple readiness check (or use other logic)
    if len(phonebook) == 0: # You could make this check more complex if needed
        return jsonify({"status": "ready"}), 200
    else:
        return jsonify({"status": "not ready"}), 503

# Create a new contact
@app.route('/contacts', methods=['POST'])
def create_contact():
    global next_contact_id
    contact_data = request.json
    # Proceed to create the contact if validation succeeds
    contact = {
        'id': next_contact_id,
        'name': contact_data['name'],
        'phone': contact_data['phone']
    }
    phonebook[next_contact_id] = contact
    next_contact_id += 1 # Increment the ID for the next contact
    return jsonify(contact), 201
```
- Open Editors:** Shows multiple tabs for "main.py" across different assignments.
- Bottom Bar:** Includes tabs for PROBLEMS, DEBUG CONSOLE, OUTPUT, TERMINAL, and PORTS. The TERMINAL tab shows a command-line session with an error message: "error: unexpected error when reading response body. Please retry. Original error: net/http: request canceled (Client.Timeout or context cancel)".
- Status Bar:** Shows "zsh - Assignment-4" in the terminal tab, and "Ln 12, Col 20 Spaces:4 UTF-8 LF Python" in the bottom right corner.