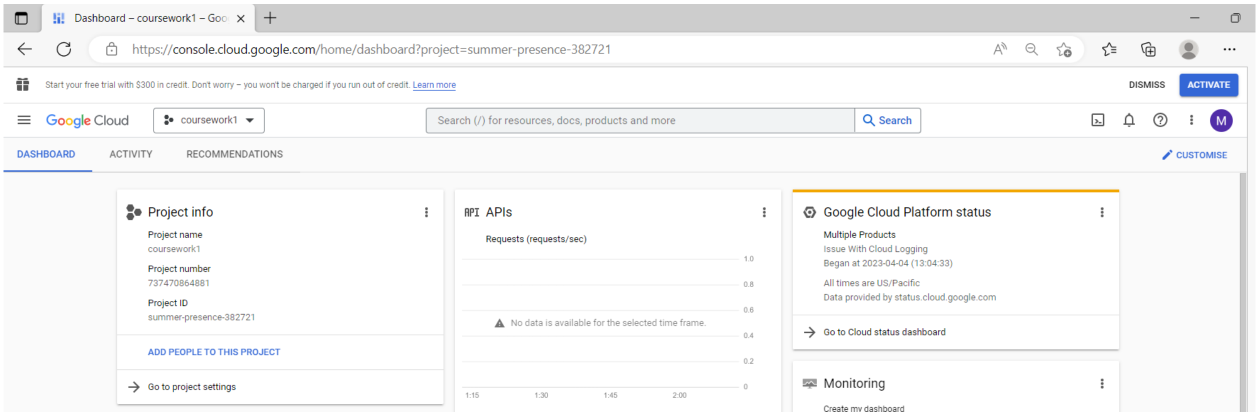
Task 1:

Point a: Low-cost virtual machine with appropriate settings:

Step 1: Open Google Cloud Console and Login with account which has credits provided by institute.

Now Created a new project:



Dashboard – coursework1 – Goo

https://console.cloud.google.com/home/dashboard?project=summer-presence-382721

Start your free trial with $300 in credit. Dont worry – you wont be charged if you run out of credit. Learn more

Search (/) for resources, docs, products and more

Project info

Project name
coursework1

Project number
737470864881

Project ID
summer-presence-382721

APIs

Requests (requests/sec)

No data is available for the selected time frame.

Google Cloud Platform status

Multiple Products
Issue With Cloud Logging
Began at 2023-04-04 (13:04:33)

All times are US/Pacific
Data provided by status.cloud.google.com

Monitoring

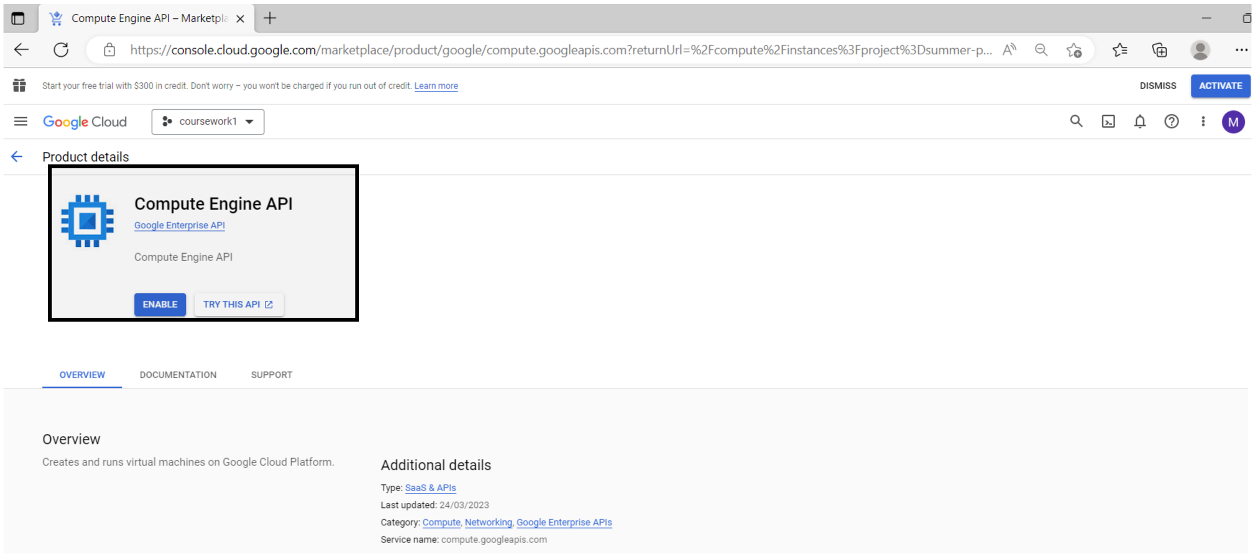
Here I want to create a new instance. For that I need it to connect with a compute engine API.

Google Compute Engine API is a cloud computing platform that offers a variety of benefits. It allows you to scale your computing resources up or down based on the demand for your services. The API provides flexibility in terms of machine types and configurations, enabling you to select the right configuration for your workload. Compute Engine API is highly reliable, runs on Google's infrastructure, which is designed to be highly available and fault tolerant. It is secure, cost-effective, and integrates with a wide range of other Google Cloud Platform services. Overall, Compute Engine API provides a powerful and flexible platform for running applications in the cloud, making it an attractive choice for many businesses and developers.

In this step I enable the compute engine and connected it with billed account which I have mentioned earlier.

In next step I am going to create a new instance.

Figure below provides the overview of compute engine API


Compute Engine API – Marketpla
https://console.cloud.google.com/marketplace/product/google/compute.googleapis.com?returnUrl=%2Fcompute%2Finstances%3Fproject%3Dsummer-p...
DISMISS
ACTIVATE
Google Cloud
coursework1
Product details
ENABLE
TRY THIS API
OVERVIEW
DOCUMENTATION
SUPPORT

Start your free trial with $300 in credit. Dont worry – you wont be charged if you run out of credit. Learn more

Compute Engine API

Google Enterprise API

Compute Engine API

Overview

Creates and runs virtual machines on Google Cloud Platform.

Additional details

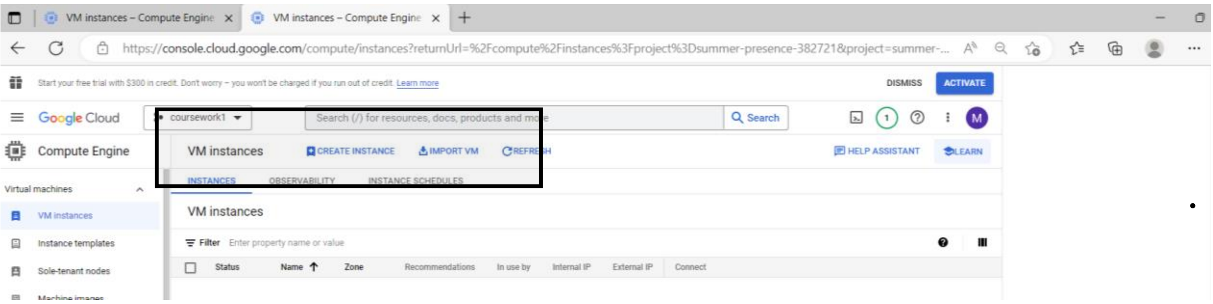
Type: SaaS & APIs

Last updated: 24/03/2023

Category: Compute, Networking, Google Enterprise APIs

Service name: compute.googleapis.com

In below diagram it can be observed that by VM instances have been created. Here we can observe the option of create instances. As highlighted



Start your free trial with $300 in credit. Dont worry – you wont be charged if you run out of credit. Learn more

As per my understanding following steps are required to complete the instances and their working.

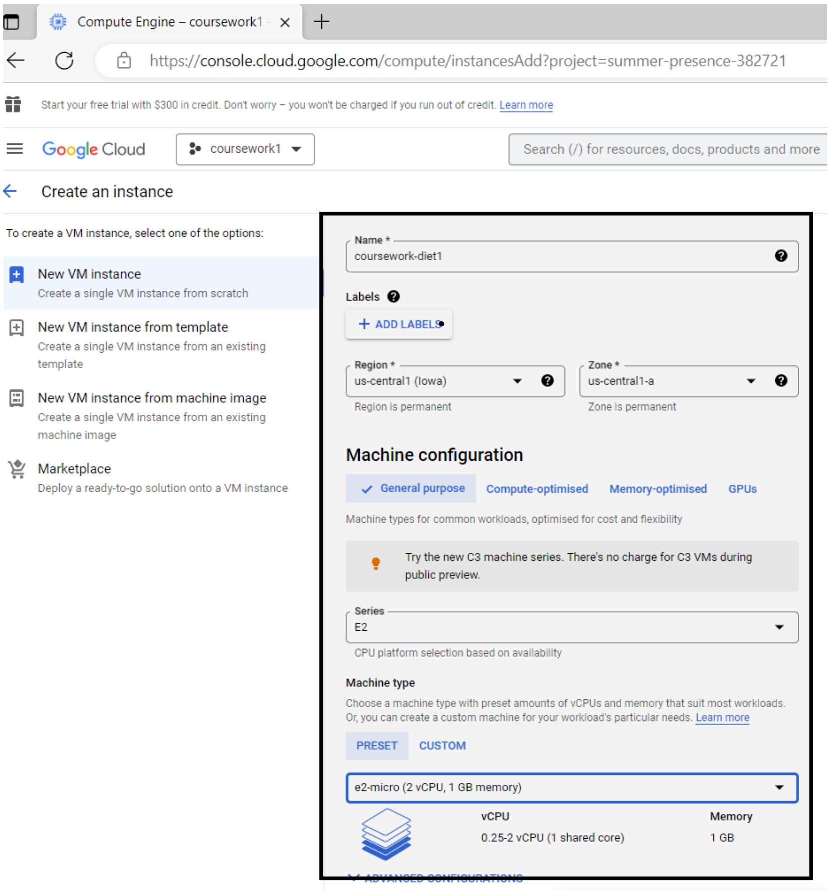
- To create an instance and install a web server, I need to follow a few steps.
- Firstly, log in to your Google Cloud Console and select Compute Engine from the main menu.
- Then, create an instance with a name of your choice and select a region within Europe. Ensure that you allow HTTP and HTTPS network traffic in the firewall settings.
- Next, choose a machine type and disk size for your instance and create it.
- Once your instance is up and running, connect to it via SSH and install on my preferred web server software.
- To test the web server, navigate to instance's IP address in a web browser, and if it's working correctly, I should see the default page for my web server.
- With these steps, I can successfully create an instance and install a web server.

From here, I am going to create and instance. I have used following setting in it.

1. Name: coursework-diet1
2. Label: None (selected as default)
3. Region: us-central1 (lowa)
4. Zone: us-central1-a
5. Machine Series: E2
6. Machine Type: e3-micro
7. Boot Disk Size: 10 GB
8. API Access: Default
9. HTTP Traffic: Allowed
10. HTTPs Traffic: Allowed

I have used above customized settings as per my understanding for a better, economical and efficient machine. Other zones and regions can also be selected. One more preferred region was to use Europe but I opted for US as it has a bit lower prices and it is low carbon zone as per google cloud information on regions and zones.

Figures below show my setting for this instance.



Compute Engine – coursework1

https://console.cloud.google.com/compute/instancesAdd?project=summer-presence-382721

Start your free trial with $300 in credit. Don't worry – you won't be charged if you run out of credit. Learn more

Create an instance

To create a VM instance, select one of the options:

New VM instance
Create a single VM instance from scratch

New VM instance from template
Create a single VM instance from an existing template

New VM instance from machine image
Create a single VM instance from an existing machine image

Marketplace
Deploy a ready-to-go solution onto a VM instance

Name *
coursework-diet1

Labels

Region *
us-central1 (Iowa)
Region is permanent

Zone *
us-central1-a
Zone is permanent

Machine configuration

Machine types for common workloads, optimised for cost and flexibility

Try the new C3 machine series. There's no charge for C3 VMs during public preview.

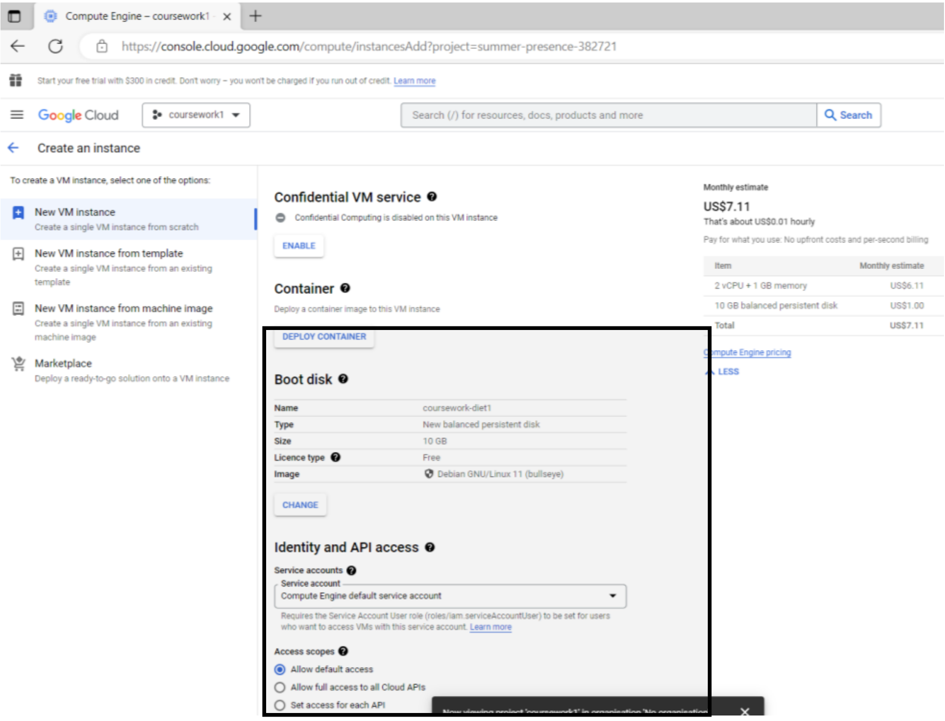
Series
E2
CPU platform selection based on availability

Machine type
Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. Learn more

e2-micro (2 vCPU, 1 GB memory)

vCPU	Memory
0.25-2 vCPU (1 shared core)	1 GB

Figure Below provides overview of Boot Disk:


Compute Engine – coursework1
https://console.cloud.google.com/compute/instancesAdd?project=summer-presence-382721
Start your free trial with $300 in credit. Don't worry – you won't be charged if you run out of credit. Learn more
Google Cloud
coursework1
Search (/) for resources, docs, products and more
Search
Create an instance
To create a VM instance, select one of the options:
New VM instance
Create a single VM instance from scratch
New VM instance from template
Create a single VM instance from an existing template
New VM instance from machine image
Create a single VM instance from an existing machine image
Marketplace
Deploy a ready-to-go solution onto a VM instance
ENABLE
DEPLOY CONTAINER
CHANGE
Compute Engine pricing
LESS

Confidential VM service

Confidential Computing is disabled on this VM instance

Container

Deploy a container image to this VM instance

Boot disk

Name	coursework-diet1
Type	New balanced persistent disk
Size	10 GB
Licence type	Free
Image	Debian GNU/Linux 11 (bullseye)

Identity and API access

Service accounts

Service account: Compute Engine default service account

Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. Learn more

Access scopes

● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Monthly estimate

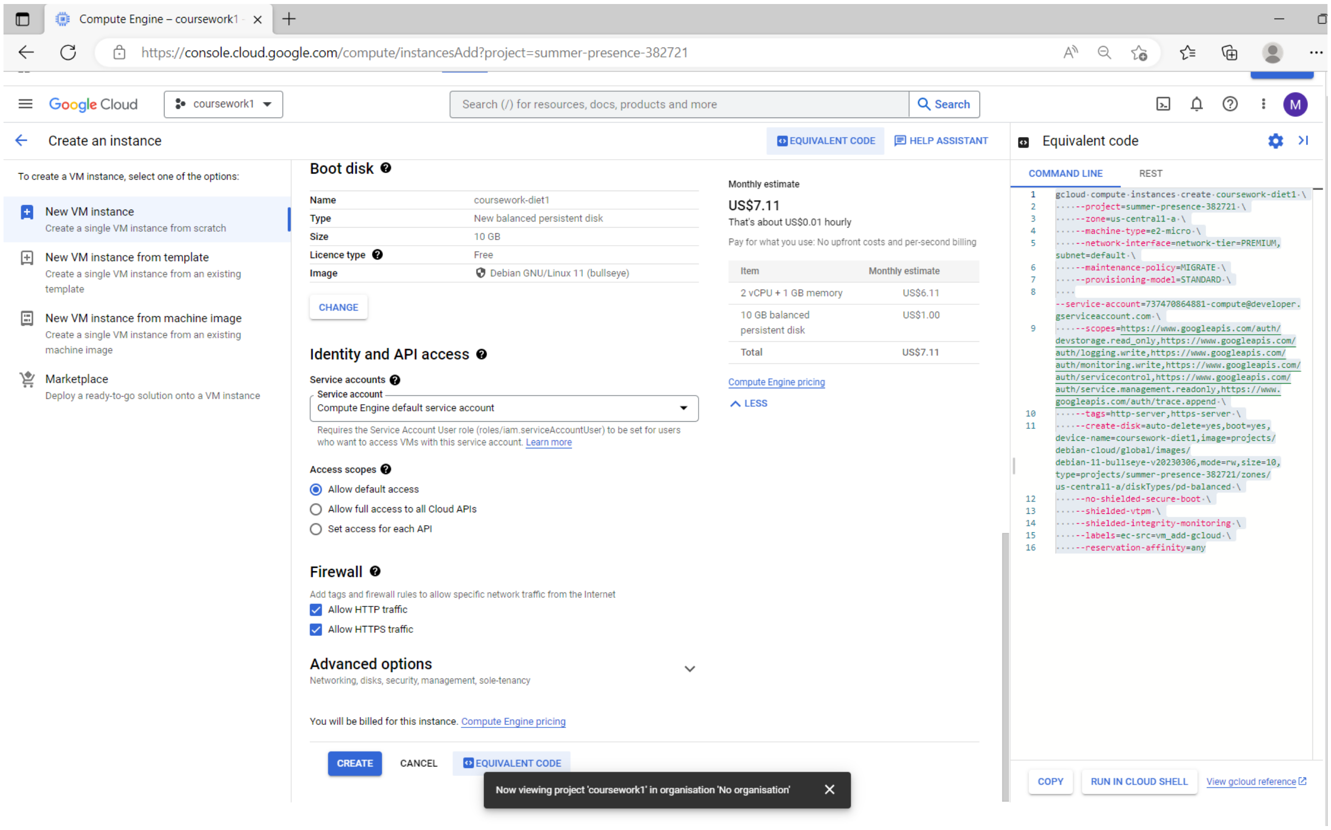
US$7.11

That's about US$0.01 hourly

Pay for what you use: No upfront costs and per-second billing

Item	Monthly estimate
2 vCPU + 1 GB memory	US$6.11
10 GB balanced persistent disk	US$1.00
Total	US$7.11

In figure below, It can be observed that I can selected https and http reequests. Moreover, on the right side equilvelent code can also be observed.


Compute Engine – coursework1
https://console.cloud.google.com/compute/instancesAdd?project=summer-presence-382721
Google Cloud
coursework1
Search (/) for resources, docs, products and more
Search
M
Create an instance
EQUIVALENT CODE
HELP ASSISTANT
To create a VM instance, select one of the options:
New VM instance
Create a single VM instance from scratch
New VM instance from template
Create a single VM instance from an existing template
New VM instance from machine image
Create a single VM instance from an existing machine image
Marketplace
Deploy a ready-to-go solution onto a VM instance
CHANGE
Compute Engine pricing
LESS
CREATE
CANCEL
EQUIVALENT CODE
Equivalent code
COMMAND LINE
REST
COPY
RUN IN CLOUD SHELL
View gcloud reference

Boot disk

Name	coursework-diet1
Type	New balanced persistent disk
Size	10 GB
Licence type	Free
Image	Debian GNU/Linux 11 (bullseye)

Identity and API access

Service accounts

Service account: Compute Engine default service account

Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. Learn more

Access scopes

● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

☑ Allow HTTP traffic

☑ Allow HTTPS traffic

Advanced options

Networking, disks, security, management, sole-tenancy

You will be billed for this instance. Compute Engine pricing

Now viewing project 'coursework1' in organisation 'No organisation'

Monthly estimate

US$7.11

That's about US$0.01 hourly

Pay for what you use: No upfront costs and per-second billing

Item	Monthly estimate
2 vCPU + 1 GB memory	US$6.11
10 GB balanced persistent disk	US$1.00
Total	US$7.11

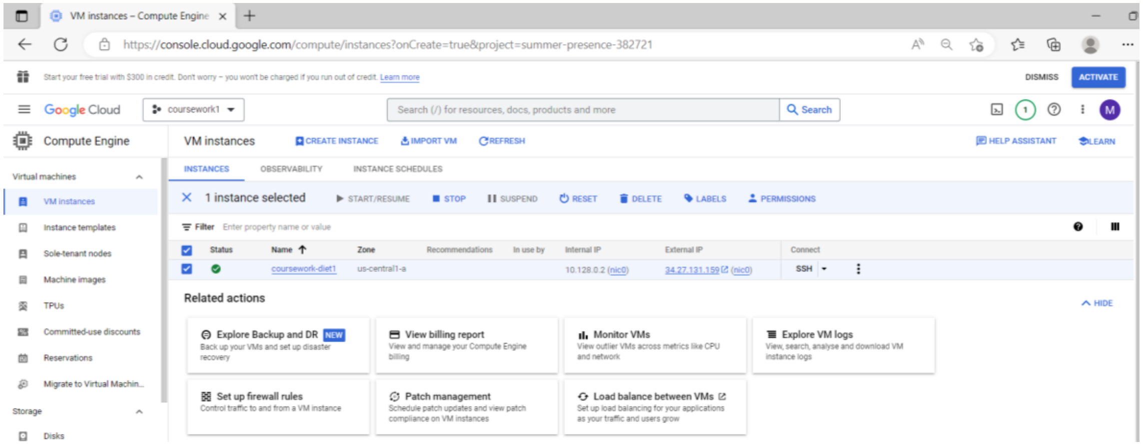
```
gcloud compute instances create coursework-diet1 \
    --project=summer-presence-382721 \
    --zone=us-central1-a \
    --machine-type=e2-micro \
    --network-interface=network-tier=PREMIUM,subnet=default \
    --maintenance-policy=MIGRATE \
    --provisioning-model=STANDARD \
    --service-account=737470864881-compute@developer.gserviceaccount.com \
    --scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append \
    --tags=http-server,https-server \
    --create-disk=auto-delete=yes,boot=yes,device-name=coursework-diet1,image=projects/debian-cloud/global/images/debian-11-bullseye-v20230306,mode=rw,size=10,type=projects/summer-presence-382721/zones/us-central1-a/diskTypes/pd-balanced \
    --no-shielded-secure-boot \
    --shielded-vtpm \
    --shielded-integrity-monitoring \
    --labels=ec-src=vm_add-gcloud \
    --reservation-affinity=any
```

Equivalent Code:

The purpose of equivalent code when creating instances is to enable the creation of instances using code instead of the Google Cloud Console. This approach allows for automation and the ability to repeat deployments of instances using infrastructure-as-code (IaC) methods. Writing equivalent code allows the creation of instances in various programming languages such as Python, Java, Go, and Ruby using either Google Cloud SDK or third-party tools like Terraform. With equivalent code, one can create instances, specify machine types and regions, configure networking, and install software in a systematic and automated way, making it easier to manage infrastructure at scale.

```
gcloud compute instances create coursework-diet1 \
    --project=summer-presence-382721 \
    --zone=us-central1-a \
    --machine-type=e2-micro \
    --network-interface=network-tier=PREMIUM,subnet=default \
    --maintenance-policy=MIGRATE \
    --provisioning-model=STANDARD \
    --service-account=737470864881-compute@developer.gserviceaccount.com \
    --
scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https
://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleap
is.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append \
    --tags=http-server,https-server \
    --create-disk=auto-delete=yes,boot=yes,device-name=coursework-diet1,image=projects/debian-
cloud/global/images/debian-11-bullseye-v20230306,mode=rw,size=10,type=projects/summer-presence-382721/zones/us-
central1-a/diskTypes/pd-balanced \
    --no-shielded-secure-boot \
    --shielded-vtpm \
    --shielded-integrity-monitoring \
    --labels=ec-src=vm_add-gcloud \
    --reservation-affinity=any
```

Figure below provides and overview of the created instance. It can be observed that its status is green which indicates that it is connected and running.



Start your free trial with $300 in credit. Don't worry – you won't be charged if you run out of credit. Learn more

☑	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
☑	✓	coursework-diet1	us-central1-a			10.128.0.2 (nic0)	34.27.131.159 (nic0)	SSH

Related actions

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

View billing report
View and manage your Compute Engine billing

Monitor VMs
View outlier VMs across metrics like CPU and network

Explore VM logs
View, search, analyse and download VM instance logs

Set up firewall rules
Control traffic to and from a VM instance

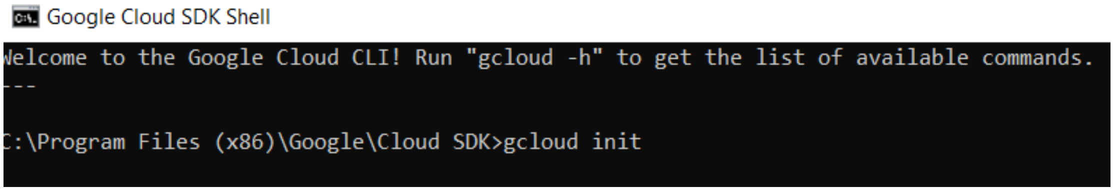
Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance between VMs
Set up load balancing for your applications as your traffic and users grow

So far, I have created project and adjusted instance by connecting it to virtual machine. Meanwhile, now I will show the work done in the google cloud SDK Shell.

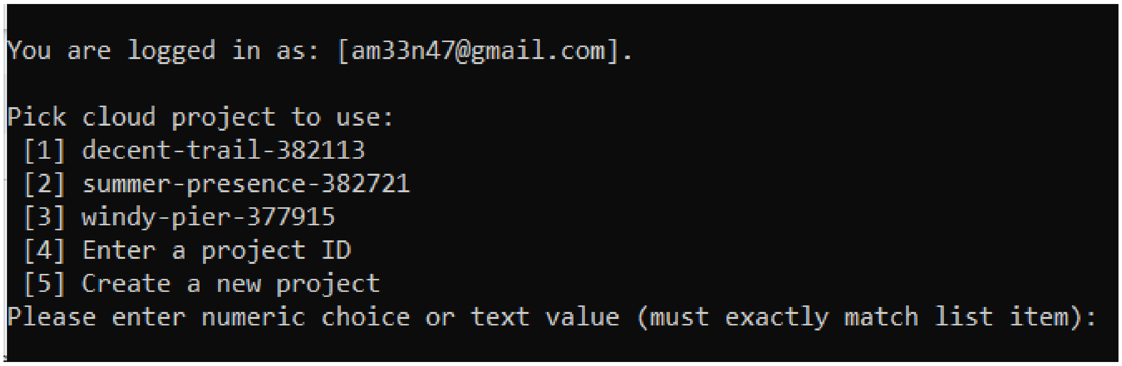
Step 1: Run *gcloud init*

The purpose of running the "gcloud init" command in the Google Cloud SDK shell is to initialize and authenticate the SDK with my Google Cloud account. This command helps us configure our local environment and set default settings for various parameters, such as project ID, zone, and region. The command prompts us to select a Google Cloud project to use with the SDK and authenticate using our Google Cloud credentials. Once we have completed the initialization process, we can use the SDK to interact with our Google Cloud resources from our local environment. This can include creating and managing instances, storage buckets, and other Google Cloud Platform services.



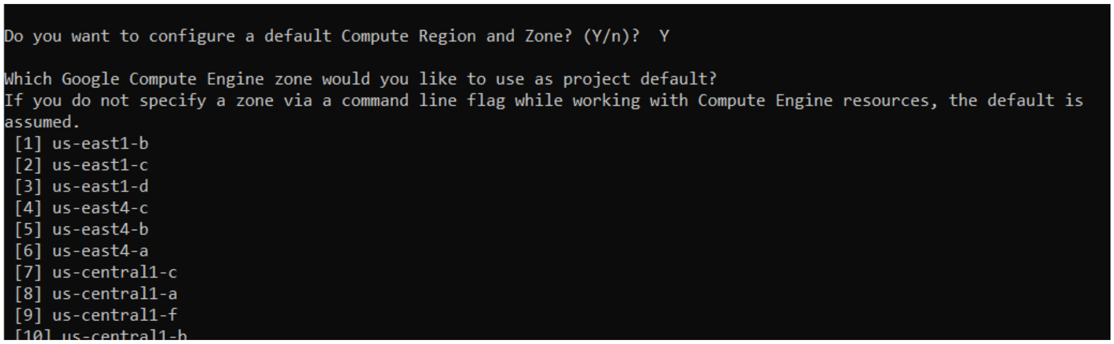
```
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Program Files (x86)\Google\Cloud SDK>gcloud init
```

Step 2: Selecting my project summer-presence-382721



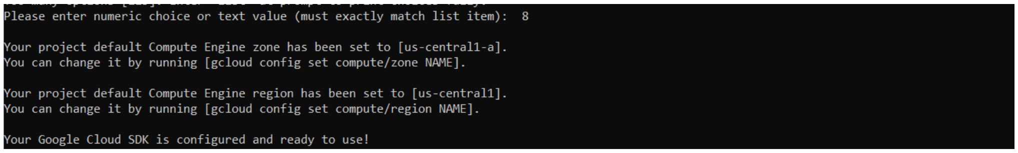
```
You are logged in as: [am33n47@gmail.com].

Pick cloud project to use:
 [1] decent-trail-382113
 [2] summer-presence-382721
 [3] windy-pier-377915
 [4] Enter a project ID
 [5] Create a new project
Please enter numeric choice or text value (must exactly match list item):
```

Step 3: Reconfiguring time zone 8 which is the us-central1-a, as I have selected in the instances.



```
Do you want to configure a default Compute Region and Zone? (Y/n)?  Y

Which Google Compute Engine zone would you like to use as project default?
If you do not specify a zone via a command line flag while working with Compute Engine resources, the default is
assumed.
 [1] us-east1-b
 [2] us-east1-c
 [3] us-east1-d
 [4] us-east4-c
 [5] us-east4-b
 [6] us-east4-a
 [7] us-central1-c
 [8] us-central1-a
 [9] us-central1-f
 [10] us-central1-b
```

Step 4: Now my google cloud console is all set.



```
Please enter numeric choice or text value (must exactly match list item):  8

Your project default Compute Engine zone has been set to [us-central1-a].
You can change it by running [gcloud config set compute/zone NAME].

Your project default Compute Engine region has been set to [us-central1].
You can change it by running [gcloud config set compute/region NAME].

Your Google Cloud SDK is configured and ready to use!
```

Point B: Installing a webserver of my choosing

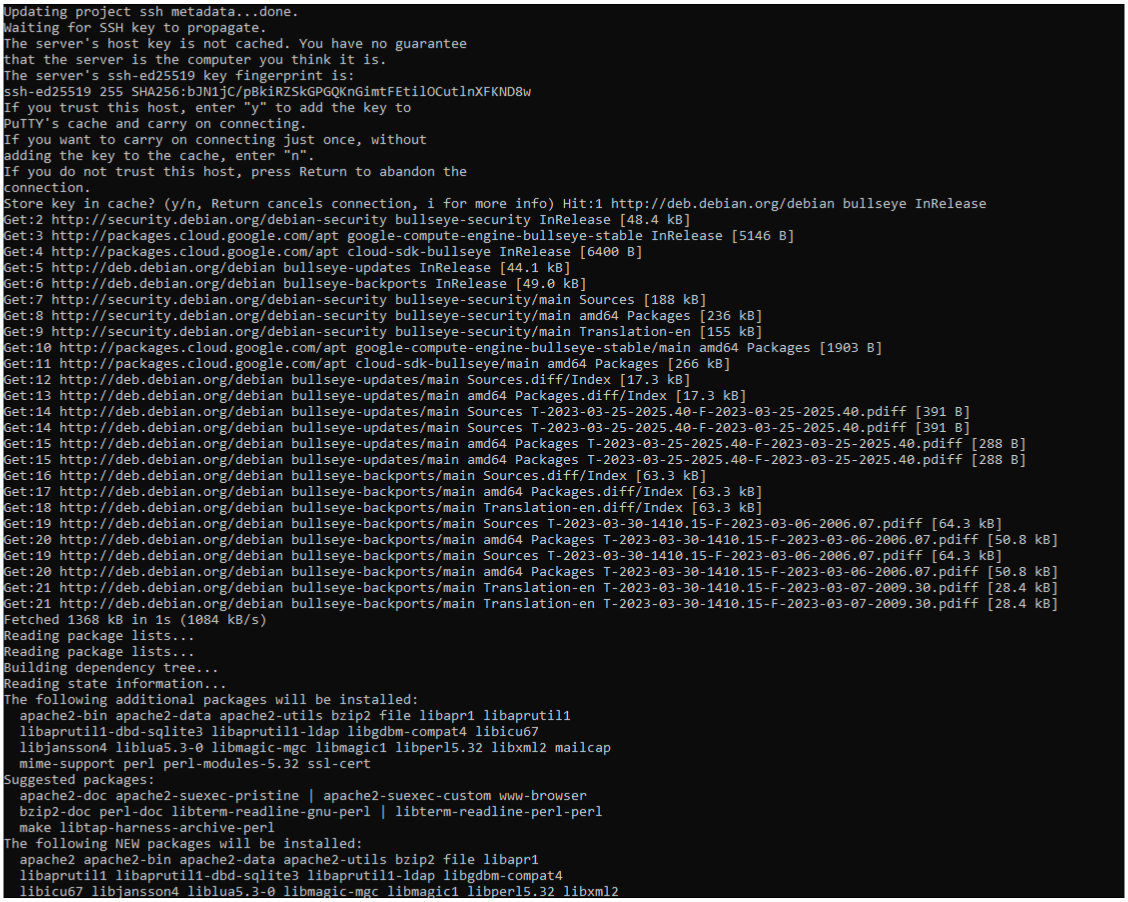
Web Server: apache2

Apache2 is a widely-used web server software that offers several benefits. It is free, open-source, and highly customizable, making it a popular choice for developers and businesses alike. It supports multiple programming languages, has robust security features, and can handle a high volume of web traffic, making it a reliable and scalable solution for hosting websites and web applications.

Step 1: Installation

Run: gcloud compute ssh coursework-diet1 --zone=us-central1-a --command='sudo apt-get update && sudo apt-get install -y apache2'

To deploy the web server on instances this process is being done. In this, The "gcloud compute ssh" command is used to connect to a virtual machine instance in the Google Cloud Platform (GCP) via SSH. In this specific command, the instance named "coursework-diet1" is being accessed, and the zone "us-central1-a" is specified. Additionally, the command being executed on the instance is "sudo apt-get update && sudo apt-get install -y apache2", which updates the package lists and installs the Apache2 web server on the instance.



```
Updating project ssh metadata...done.
Waiting for SSH key to propagate.
The server's host key is not cached. You have no guarantee
that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:bJN1jC/pBkiRZSkGPGQKnGimtFEtilOCutlnXFKND8w
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n, Return cancels connection, i for more info) Hit:1 http://deb.debian.org/debian bullseye InRelease
Get:2 http://security.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://packages.cloud.google.com/apt google-compute-engine-bullseye-stable InRelease [5146 B]
Get:4 http://packages.cloud.google.com/apt cloud-sdk-bullseye InRelease [6400 B]
Get:5 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:6 http://deb.debian.org/debian bullseye-backports InRelease [49.0 kB]
Get:7 http://security.debian.org/debian-security bullseye-security/main Sources [188 kB]
Get:8 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [236 kB]
Get:9 http://security.debian.org/debian-security bullseye-security/main Translation-en [155 kB]
Get:10 http://packages.cloud.google.com/apt google-compute-engine-bullseye-stable/main amd64 Packages [1903 B]
Get:11 http://packages.cloud.google.com/apt cloud-sdk-bullseye/main amd64 Packages [266 kB]
Get:12 http://deb.debian.org/debian bullseye-updates/main Sources.diff/Index [17.3 kB]
Get:13 http://deb.debian.org/debian bullseye-updates/main amd64 Packages.diff/Index [17.3 kB]
Get:14 http://deb.debian.org/debian bullseye-updates/main Sources T-2023-03-25-2025.40-F-2023-03-25-2025.40.pdiff [391 B]
Get:14 http://deb.debian.org/debian bullseye-updates/main Sources T-2023-03-25-2025.40-F-2023-03-25-2025.40.pdiff [391 B]
Get:15 http://deb.debian.org/debian bullseye-updates/main amd64 Packages T-2023-03-25-2025.40-F-2023-03-25-2025.40.pdiff [288 B]
Get:15 http://deb.debian.org/debian bullseye-updates/main amd64 Packages T-2023-03-25-2025.40-F-2023-03-25-2025.40.pdiff [288 B]
Get:16 http://deb.debian.org/debian bullseye-backports/main Sources.diff/Index [63.3 kB]
Get:17 http://deb.debian.org/debian bullseye-backports/main amd64 Packages.diff/Index [63.3 kB]
Get:18 http://deb.debian.org/debian bullseye-backports/main Translation-en.diff/Index [63.3 kB]
Get:19 http://deb.debian.org/debian bullseye-backports/main Sources T-2023-03-30-1410.15-F-2023-03-06-2006.07.pdiff [64.3 kB]
Get:20 http://deb.debian.org/debian bullseye-backports/main amd64 Packages T-2023-03-30-1410.15-F-2023-03-06-2006.07.pdiff [50.8 kB]
Get:19 http://deb.debian.org/debian bullseye-backports/main Sources T-2023-03-30-1410.15-F-2023-03-06-2006.07.pdiff [64.3 kB]
Get:20 http://deb.debian.org/debian bullseye-backports/main amd64 Packages T-2023-03-30-1410.15-F-2023-03-06-2006.07.pdiff [50.8 kB]
Get:21 http://deb.debian.org/debian bullseye-backports/main Translation-en T-2023-03-30-1410.15-F-2023-03-07-2009.30.pdiff [28.4 kB]
Get:21 http://deb.debian.org/debian bullseye-backports/main Translation-en T-2023-03-30-1410.15-F-2023-03-07-2009.30.pdiff [28.4 kB]
Fetched 1368 kB in 1s (1084 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 file libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libgdbm-compat4 libicu67
  libjansson4 liblua5.3-0 libmagic-mgc libmagic1 libperl5.32 libxml2 mailcap
  mime-support perl perl-modules-5.32 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  bzip2-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl
  make libtap-harness-archive-perl
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 file libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libgdbm-compat4
  libicu67 libjansson4 liblua5.3-0 libmagic-mgc libmagic1 libperl5.32 libxml2
```

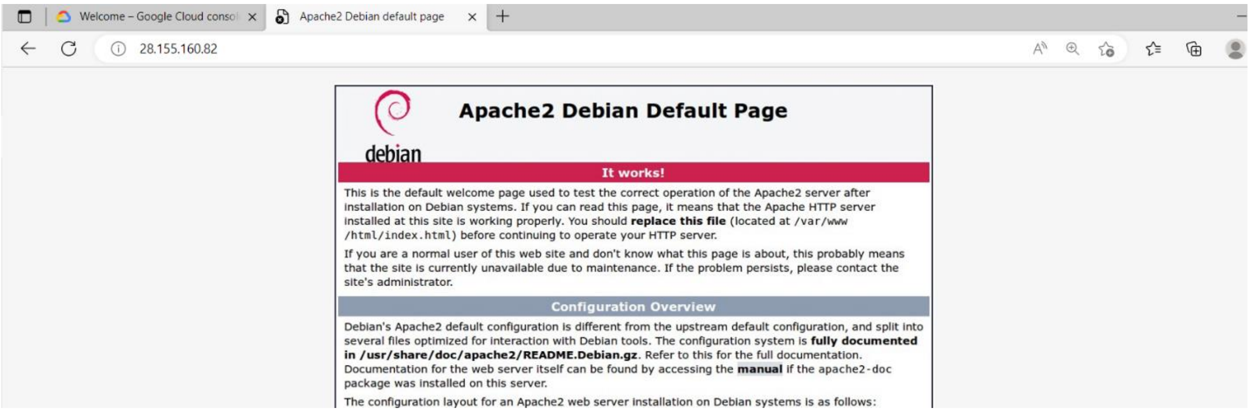
Step 2: Extract IP Address:

Run: `gcloud compute instances describe coursework-diet1 --zone=us-central1-a --format json | jq -r '.networkInterfaces[].accessConfigs[].natIP'`

This command retrieves the external IP address of a virtual machine instance named "coursework-diet1" in the Google Cloud Platform (GCP) in the "us-central1-a" zone. The command uses the "gcloud compute instances describe" command to retrieve detailed information about the instance in JSON format, and then pipes the output to the "jq" tool. The "jq" tool is used to extract the external IP address of the instance from the JSON output by using the ".networkInterfaces[].accessConfigs[].natIP" filter. The "-r" flag is used with "jq" to ensure that the output is in raw format, without quotes or other formatting.

Obtained IP: 28.155.160.82

Step 3: IP Testing



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

Here, webserver has been established and ready to use.

Now, in next part. I am going to upload an image on this web server through a random URL of my choice.

Point C: Copying Image to *coursework-diet1* and retrieving it on *28.155.160.82*

Step 1: Copying image gcu.jpg to webserver from my device.

Run: `gcloud compute scp C:/Users/Amin/Downloads/gcu.jpg coursework-diet1:/var/www/html`

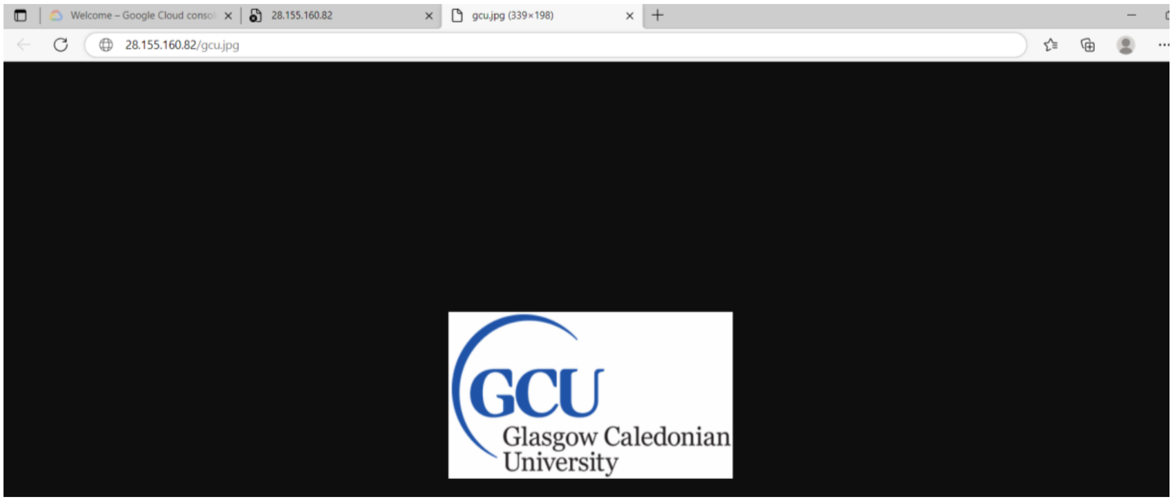
The "gcloud compute scp" command is used to securely copy files between a local machine and a virtual machine instance in the Google Cloud Platform (GCP). In this specific command, the file "gcu.jpg" located on the local machine at "C:/Users/Amin/Downloads/" is being copied to the virtual machine instance named "coursework-diet1" in the "/var/www/html" directory using the "gcloud compute scp" command. The file will be securely transferred over SSH and will be accessible in the specified directory on the virtual machine instance. This command can be useful when you need to transfer files to or from a GCP virtual machine instance.

Step 2: Accessing copied image

Run: gcloud compute ssh coursework-diet1 --zone= us-central1-a --command='sudo service apache2 restart'

Copied Image:

URL: https://28.155.160.82/gcu.jpg



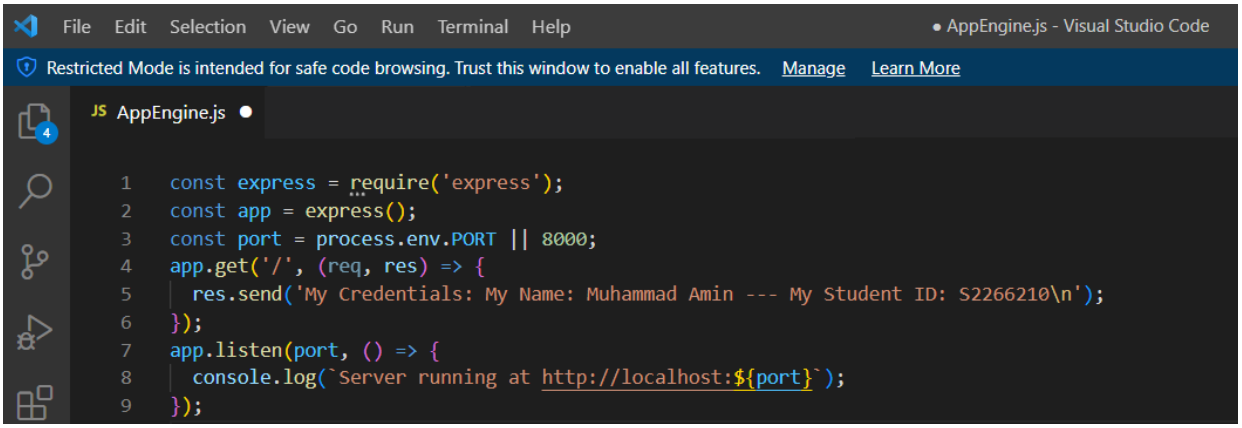
Point D: App Development

Step 1: Creating App Engine

Code:

```
const express = require('express');
const app = express();
const port = process.env.PORT || 8000;
app.get('/', (req, res) => {
  res.send('My Credentials: My Name: Muhammad Amin --- My Student ID:
S2266210\n');
});
app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

This code creates a new Express.js application using the express module and listens for incoming requests on the specified port. When a request is received with the HTTP GET method and the root path /, the server sends a response with the desired message using the res.send() method. The process.env.PORT statement works the same way as before, reading the value of the PORT environment variable. The console.log statement outputs a message to the console indicating that the server is running and listening for requests.



Restricted Mode is intended for safe code browsing. Trust this window to enable all features.

```javascript
const express = require('express');
const app = express();
const port = process.env.PORT || 8000;
app.get('/', (req, res) => {
  res.send('My Credentials: My Name: Muhammad Amin --- My Student ID: S2266210\n');
});
app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

Step 2: Initialization through Google Cloud SDK

Run: gcloud app create --region=Europe-west1

The command gcloud app create --region=Europe-west1 creates a new App Engine application in the specified region. The --region flag specifies the region where the application will be deployed. In this case, Europe-west1 refers to the App Engine region in Western Europe, located in Belgium.

Step 3: App Deployment

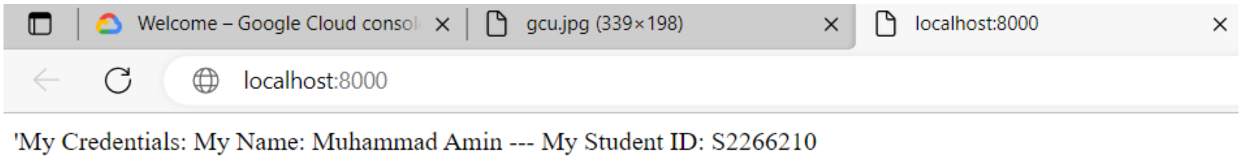
Run: gcloud app deploy

Step 4: App Testing

Run: npm start

Step 5: Test App on Local Host

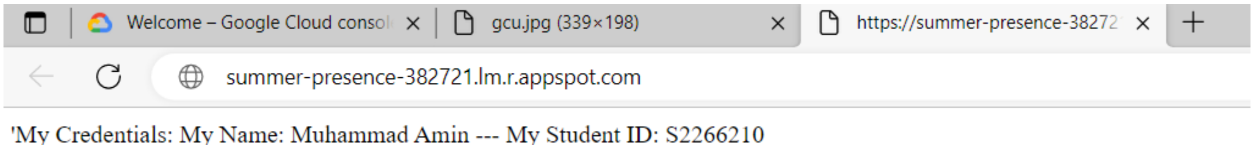
Link: localhost:8000



'My Credentials: My Name: Muhammad Amin --- My Student ID: S2266210

Step 5: Test App Remotely

URL: https://summer-presence-382721.lm.r.appspot.com/


Welcome – Google Cloud consol
gcu.jpg (339×198)
https://summer-presence-38272
summer-presence-382721.lm.r.appspot.com

'My Credentials: My Name: Muhammad Amin --- My Student ID: S2266210

Task 2:

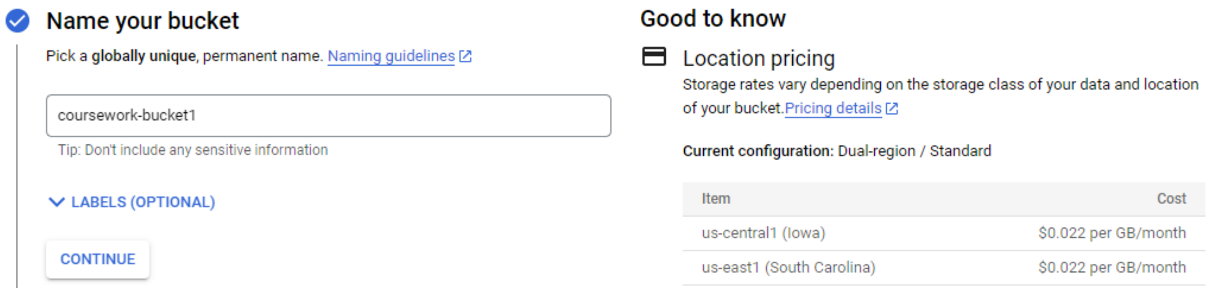
Point a: Cloud Storage Bucket

Following are the general step involved in the creation of cloud bucket

- Open the Google Cloud Console
- Select Project
- Name the bucket
- Select Data Storage
- Select Data Storage Class
- Set the default access control
- Create Bucket

In the following steps, I have created a coursework-bucket1

Step 1: Naming Bucket



Name your bucket

Pick a **globally unique**, permanent name. Naming guidelines

coursework-bucket1

Tip: Don't include any sensitive information

Good to know

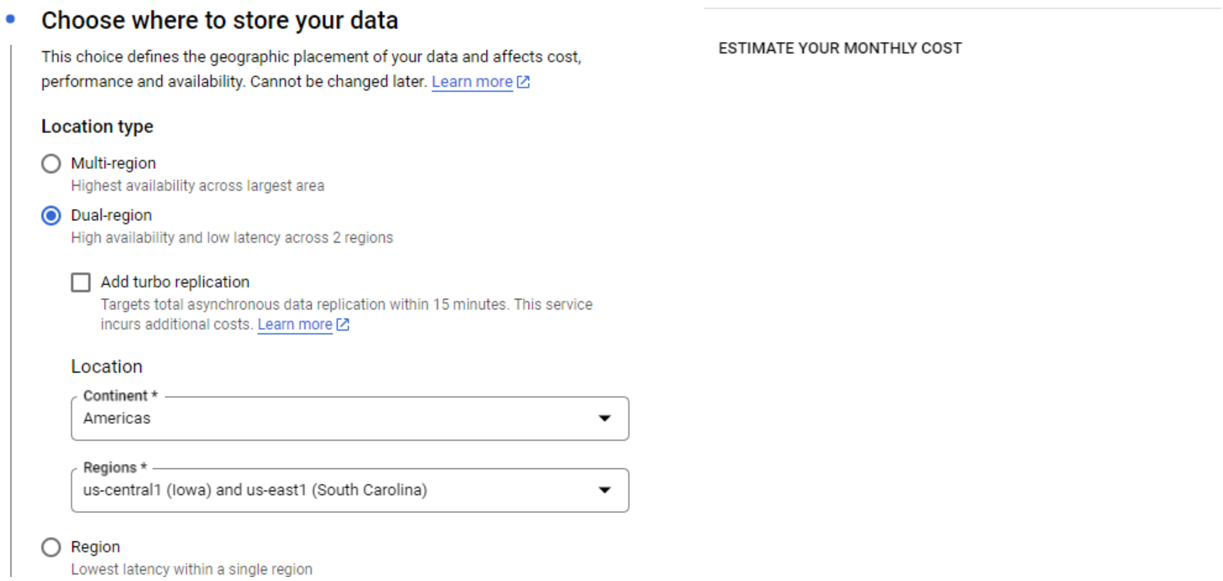
Location pricing

Storage rates vary depending on the storage class of your data and location of your bucket.Pricing details

Current configuration: Dual-region / Standard

Item	Cost
us-central1 (Iowa)	$0.022 per GB/month
us-east1 (South Carolina)	$0.022 per GB/month

Step 2: Where to store Data



Choose where to store your data

This choice defines the geographic placement of your data and affects cost, performance and availability. Cannot be changed later. Learn more

Location type

○ Multi-region
Highest availability across largest area

● Dual-region
High availability and low latency across 2 regions

□ Add turbo replication
Targets total asynchronous data replication within 15 minutes. This service incurs additional costs. Learn more

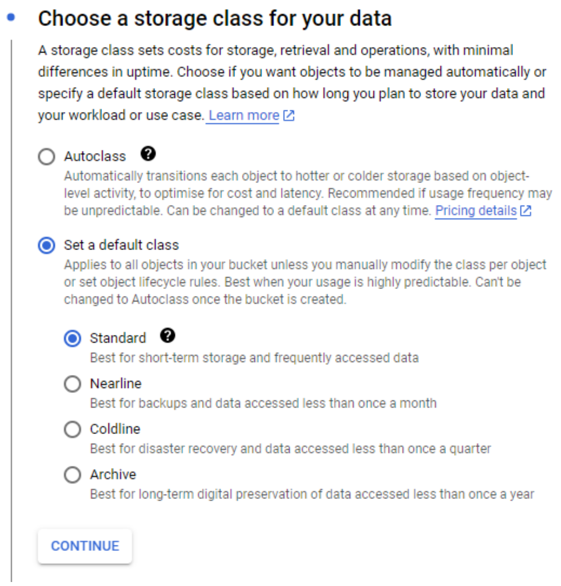
Location

Continent * Americas

Regions * us-central1 (Iowa) and us-east1 (South Carolina)

○ Region
Lowest latency within a single region

Step 3: Storage Class



Choose a storage class for your data

A storage class sets costs for storage, retrieval and operations, with minimal differences in uptime. Choose if you want objects to be managed automatically or specify a default storage class based on how long you plan to store your data and your workload or use case. Learn more

○ Autoclass
Automatically transitions each object to hotter or colder storage based on object-level activity, to optimise for cost and latency. Recommended if usage frequency may be unpredictable. Can be changed to a default class at any time. Pricing details

● Set a default class
Applies to all objects in your bucket unless you manually modify the class per object or set object lifecycle rules. Best when your usage is highly predictable. Can't be changed to Autoclass once the bucket is created.

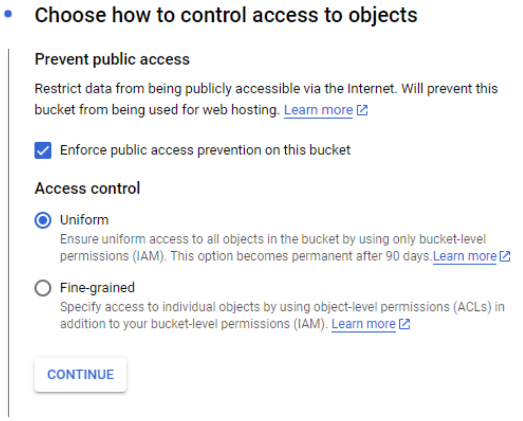
● Standard
Best for short-term storage and frequently accessed data

○ Nearline
Best for backups and data accessed less than once a month

○ Coldline
Best for disaster recovery and data accessed less than once a quarter

○ Archive
Best for long-term digital preservation of data accessed less than once a year

Step 4: Defining Control Access



Choose how to control access to objects

Prevent public access

Restrict data from being publicly accessible via the Internet. Will prevent this bucket from being used for web hosting. Learn more

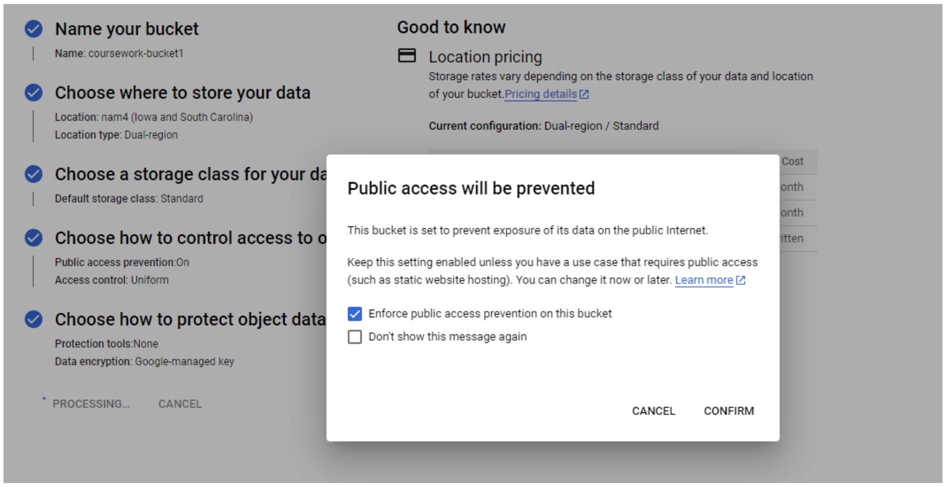
☑ Enforce public access prevention on this bucket

Access control

● Uniform
Ensure uniform access to all objects in the bucket by using only bucket-level permissions (IAM). This option becomes permanent after 90 days. Learn more

○ Fine-grained
Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). Learn more

Step 5: Public Access Prevention



Name your bucket
Name: coursework-bucket1

Choose where to store your data
Location: nam4 (Iowa and South Carolina)
Location type: Dual-region

Choose a storage class for your da
Default storage class: Standard

Choose how to control access to o
Public access prevention:On
Access control: Uniform

Choose how to protect object data
Protection tools:None
Data encryption: Google-managed key

Good to know

Location pricing
Storage rates vary depending on the storage class of your data and location of your bucket.Pricing details

Current configuration: Dual-region / Standard

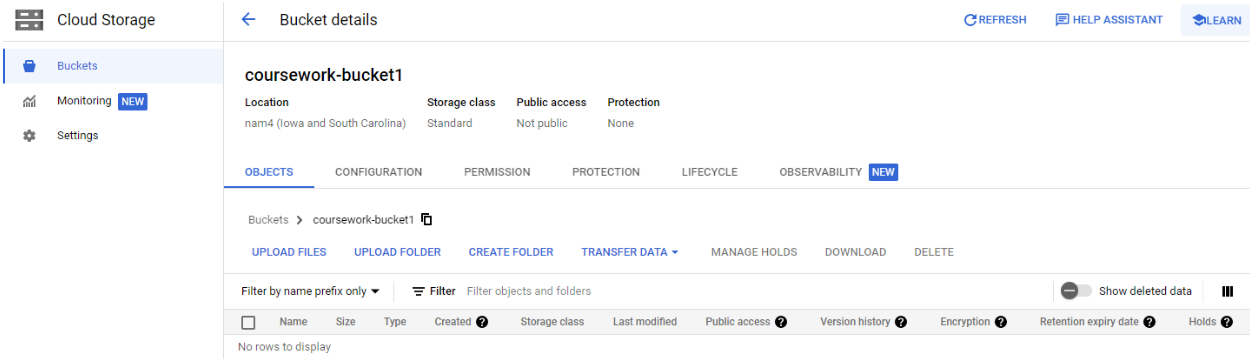
Public access will be prevented

This bucket is set to prevent exposure of its data on the public Internet.

Keep this setting enabled unless you have a use case that requires public access (such as static website hosting). You can change it now or later. Learn more

☑ Enforce public access prevention on this bucket
☐ Don't show this message again

Step 6: Bucket Created



coursework-bucket1

Location	Storage class	Public access	Protection
nam4 (Iowa and South Carolina)	Standard	Not public	None

□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiry date	Holds

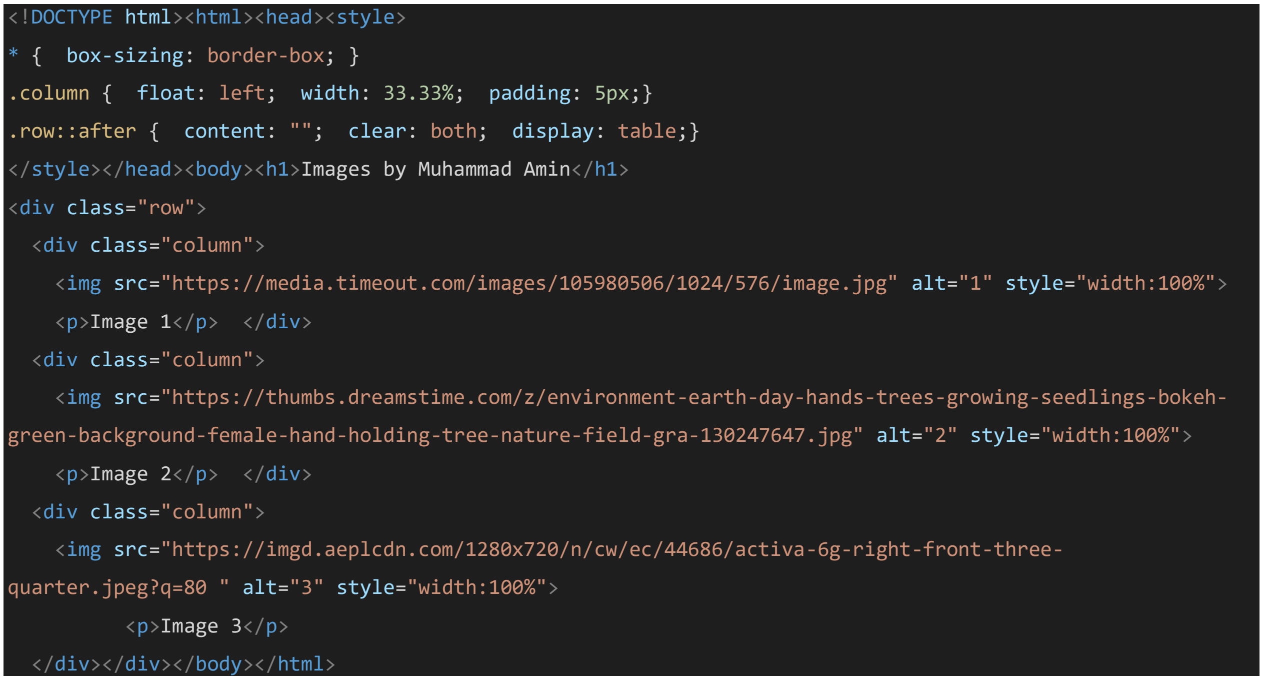
No rows to display

Point b: Uploading three photos and making public access True

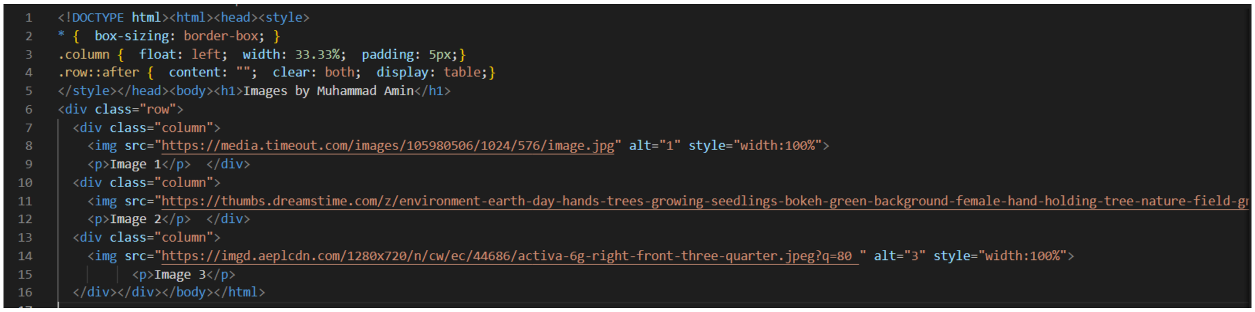
Step 1: Upload Three Photos to coursework-bucket1
gsutil cp 1.jpg gs://coursework-bucket1/1.jpg
gsutil cp 1. jpg gs://coursework-bucket1/2.jpg
gsutil cp 1. jpg gs://coursework-bucket1/3.jpg
Step 2: Assigning Public Access
gsutil acl ch -u AllUsers:R gs://coursework-bucket1/1.jpg
gsutil acl ch -u AllUsers:R gs://coursework-bucket1/2.jpg
gsutil acl ch -u AllUsers:R gs://coursework-bucket1/3.jpg

Point C: HTML File for these images and hosting on previous webserver 28.155.160.82

Step 1: HTML Code: File name task2.html



```
<!DOCTYPE html><html><head><style>
* {  box-sizing: border-box; }
.column {  float: left;  width: 33.33%;  padding: 5px;}
.row::after {  content: "";  clear: both;  display: table;}
</style></head><body><h1>Images by Muhammad Amin</h1>
<div class="row">
  <div class="column">
    <img src="https://media.timeout.com/images/105980506/1024/576/image.jpg" alt="1" style="width:100%">
    <p>Image 1</p>  </div>
  <div class="column">
    <img src="https://thumbs.dreamstime.com/z/environment-earth-day-hands-trees-growing-seedlings-bokeh-
green-background-female-hand-holding-tree-nature-field-gra-130247647.jpg" alt="2" style="width:100%">
    <p>Image 2</p>  </div>
  <div class="column">
    <img src="https://imgd.aeplcdn.com/1280x720/n/cw/ec/44686/activa-6g-right-front-three-
quarter.jpeg?q=80 " alt="3" style="width:100%">
          <p>Image 3</p>
  </div></div></body></html>
```

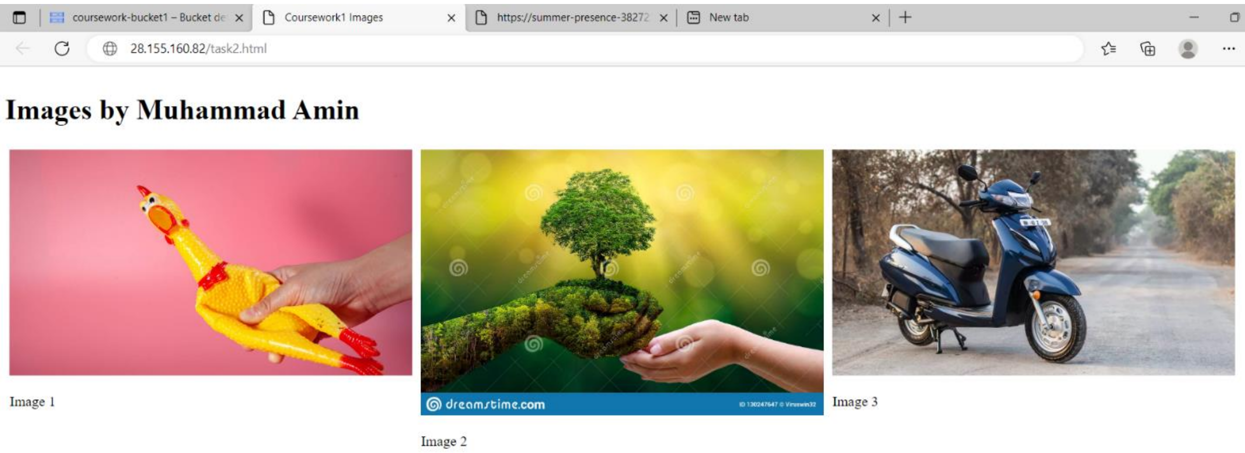


```
<!DOCTYPE html><html><head><style>
* {  box-sizing: border-box; }
.column {  float: left;  width: 33.33%;  padding: 5px;}
.row::after {  content: "";  clear: both;  display: table;}
</style></head><body><h1>Images by Muhammad Amin</h1>
<div class="row">
  <div class="column">
    <img src="https://media.timeout.com/images/105980506/1024/576/image.jpg" alt="1" style="width:100%">
    <p>Image 1</p>  </div>
  <div class="column">
    <img src="https://thumbs.dreamstime.com/z/environment-earth-day-hands-trees-growing-seedlings-bokeh-green-background-female-hand-holding-tree-nature-field-gr
    <p>Image 2</p>  </div>
  <div class="column">
    <img src="https://imgd.aeplcdn.com/1280x720/n/cw/ec/44686/activa-6g-right-front-three-quarter.jpeg?q=80 " alt="3" style="width:100%">
          <p>Image 3</p>
  </div></div></body></html>
```

Step 2: Uploading task2.html on apache2 directory

gcloud compute scp C:/Users/Amin/Documents/task2.html coursework-diet1:/var/www/html

Step 3: Accessing file



Images by Muhammad Amin

Image 1

Image 2

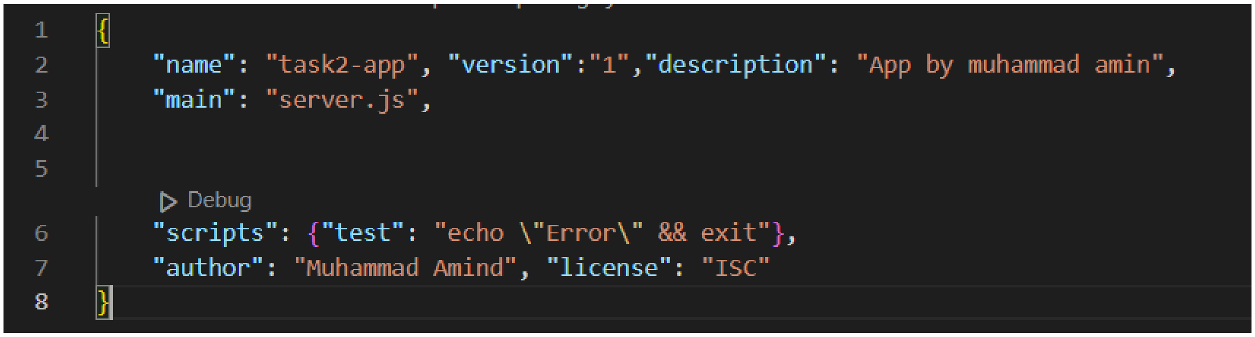
Image 3

Point D: App Development and Testing

Step 1: Initialization

Run: npm init

Json Code for task2 app:



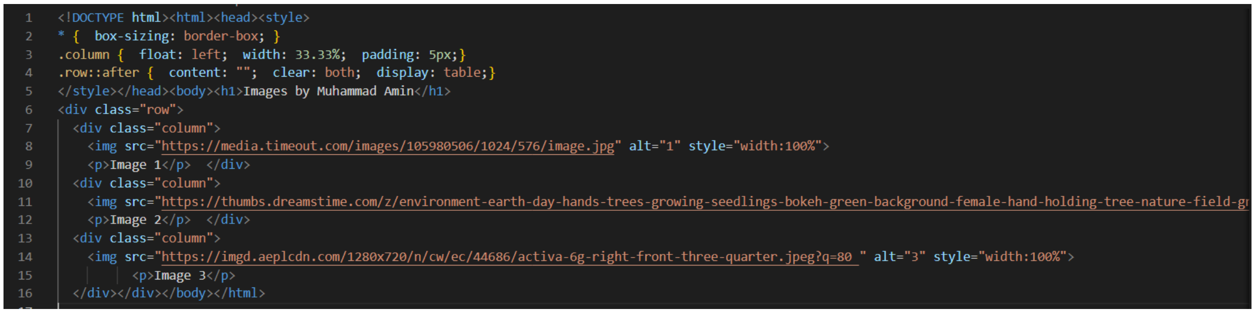
```
{
    "name": "task2-app", "version":"1","description": "App by muhammad amin",
    "main": "server.js",


    "scripts": {"test": "echo \"Error\" && exit"},
    "author": "Muhammad Amind", "license": "ISC"
}
```

Step 2: npm install exp

Code: runtime: nodejs14

Task2.html

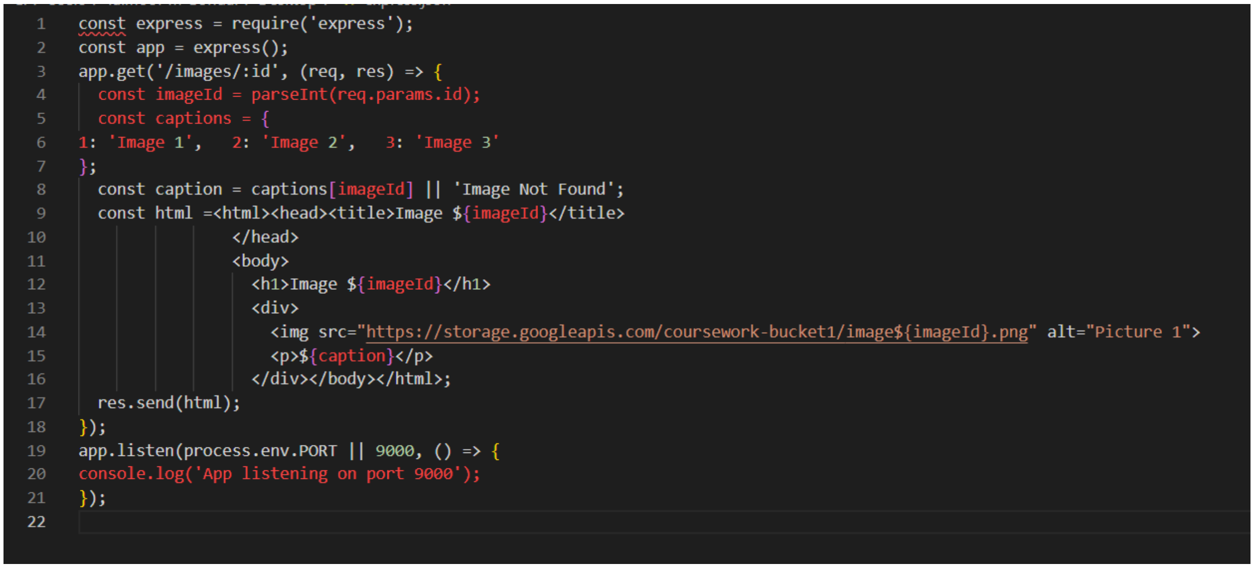
```html
<!DOCTYPE html><html><head><style>
* {  box-sizing: border-box; }
.column {  float: left;  width: 33.33%;  padding: 5px;}
.row::after {  content: "";  clear: both;  display: table;}
</style></head><body><h1>Images by Muhammad Amin</h1>
<div class="row">
  <div class="column">
    <img src="https://media.timeout.com/images/105980506/1024/576/image.jpg" alt="1" style="width:100%">
    <p>Image 1</p>  </div>
  <div class="column">
    <img src="https://thumbs.dreamstime.com/z/environment-earth-day-hands-trees-growing-seedlings-bokeh-
green-background-female-hand-holding-tree-nature-field-gra-130247647.jpg" alt="2" style="width:100%">
    <p>Image 2</p>  </div>
  <div class="column">
    <img src="https://imgd.aeplcdn.com/1280x720/n/cw/ec/44686/activa-6g-right-front-three-
quarter.jpeg?q=80 " alt="3" style="width:100%">
          <p>Image 3</p>
  </div></div></body></html>
```



```html
<!DOCTYPE html><html><head><style>
* {  box-sizing: border-box; }
.column {  float: left;  width: 33.33%;  padding: 5px;}
.row::after {  content: "";  clear: both;  display: table;}
</style></head><body><h1>Images by Muhammad Amin</h1>
<div class="row">
  <div class="column">
    <img src="https://media.timeout.com/images/105980506/1024/576/image.jpg" alt="1" style="width:100%">
    <p>Image 1</p>  </div>
  <div class="column">
    <img src="https://thumbs.dreamstime.com/z/environment-earth-day-hands-trees-growing-seedlings-bokeh-green-background-female-hand-holding-tree-nature-field-gr
    <p>Image 2</p>  </div>
  <div class="column">
    <img src="https://imgd.aeplcdn.com/1280x720/n/cw/ec/44686/activa-6g-right-front-three-quarter.jpeg?q=80 " alt="3" style="width:100%">
          <p>Image 3</p>
  </div></div></body></html>
```

Server Code:

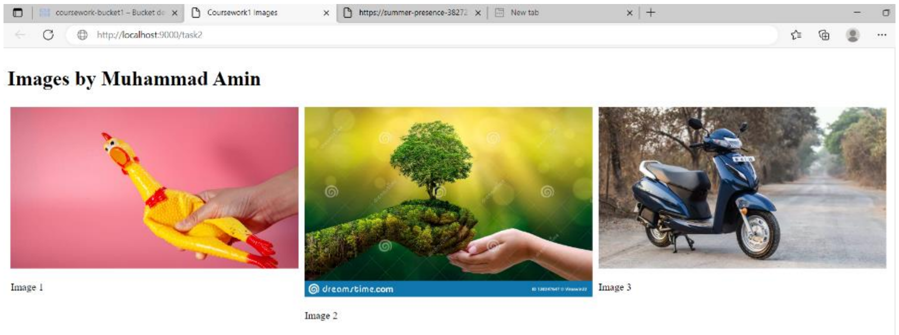
```javascript
const express = require('express'); const app = express();
app.get('/images/:id', (req, res) => {
  const imageId = parseInt(req.params.id);
  const captions = { 1: 'Image 1',   2: 'Image 2',   3: 'Image 3'};
  const caption = captions[imageId] || 'Image Not Found';
  const html =<html><head><title>Image ${imageId}</title></head><body>
  <h1>Image ${imageId}</h1><div>
  <img src="https://storage.googleapis.com/coursework-
bucket1/image${imageId}.png" alt="Picture 1"><p>${caption}</p>
  </div></body></html>;res.send(html);
});
app.listen(process.env.PORT || 9000, () => {
console.log('App listening on port 9000');
});
```



```
const express = require('express');
const app = express();
app.get('/images/:id', (req, res) => {
  const imageId = parseInt(req.params.id);
  const captions = {
1: 'Image 1',   2: 'Image 2',   3: 'Image 3'
};
  const caption = captions[imageId] || 'Image Not Found';
  const html =<html><head><title>Image ${imageId}</title>
              </head>
              <body>
                <h1>Image ${imageId}</h1>
                <div>
                  <img src="https://storage.googleapis.com/coursework-bucket1/image${imageId}.png" alt="Picture 1">
                  <p>${caption}</p>
                </div></body></html>;
  res.send(html);
});
app.listen(process.env.PORT || 9000, () => {
console.log('App listening on port 9000');
});
```

Now I want to access locally my uploaded images:

Link: http://localhost:9000/task2



coursework-bucket1 – Bucket de
Coursework1 Images
https://summer-presence-38272
New tab
http://localhost:9000/task2
dreamstime.com

Images by Muhammad Amin

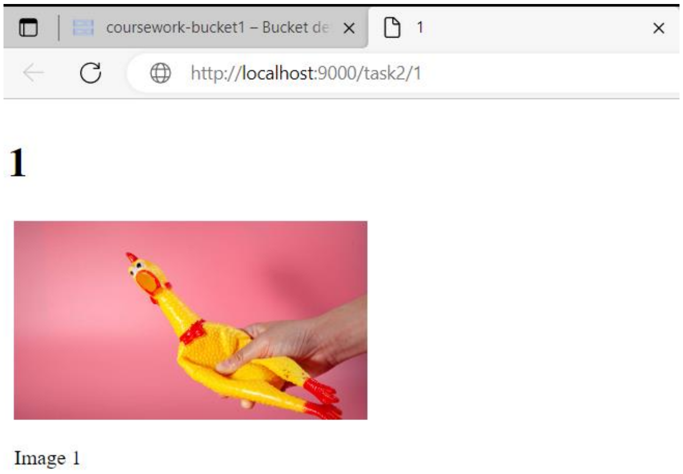
Image 1

Image 2

Image 3

Lets Access Images one by one:

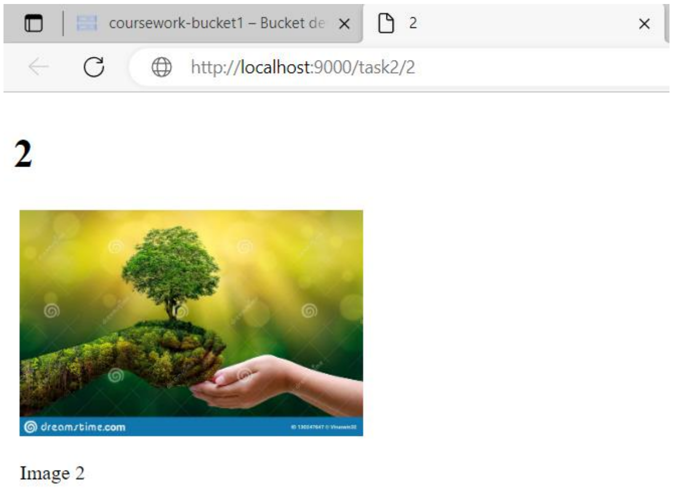
Image 1: http://localhost:9000/task2/1



1

Image 1

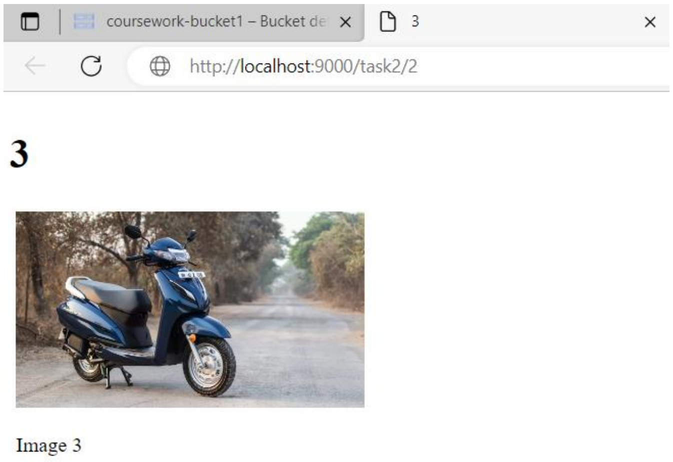
Image 2: http://localhost:9000/task2/2



2

Image 2

Image 3: http://localhost:9000/task2/3

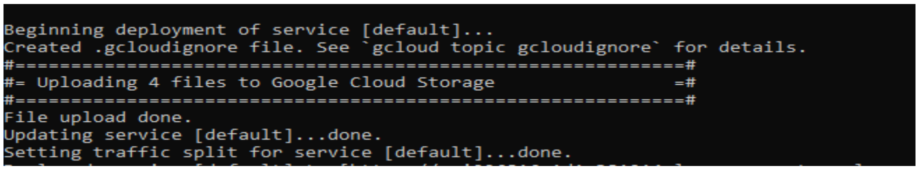


3

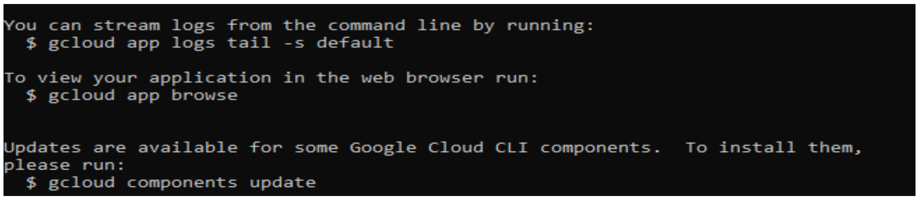
Image 3

Now App Deployment to Google Cloud

Run: gcloud app deploy



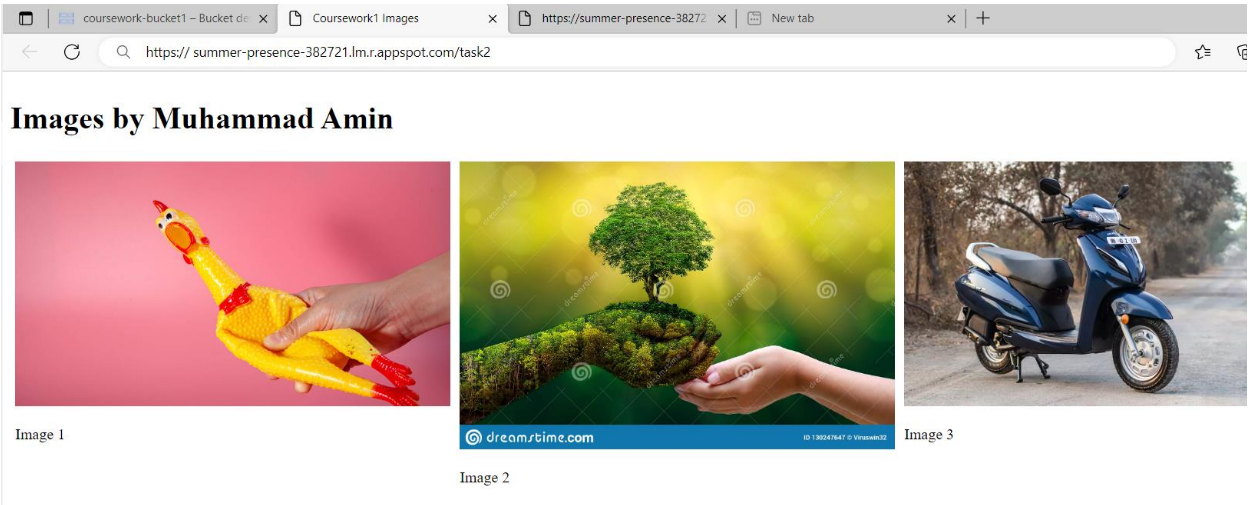
```
Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
#============================================================#
#= Uploading 4 files to Google Cloud Storage                =#
#============================================================#
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
```



```
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


Updates are available for some Google Cloud CLI components.  To install them,
please run:
  $ gcloud components update
```

Lets Access Images one by one:

Link: https:// summer-presence-382721.lm.r.appspot.com/task2



https:// summer-presence-382721.lm.r.appspot.com/task2

Images by Muhammad Amin

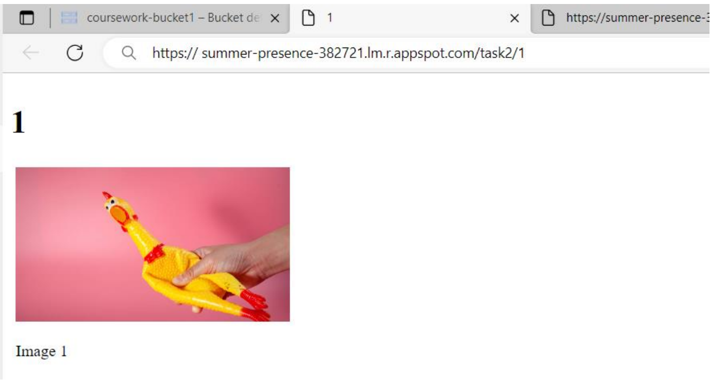
Image 1

Image 2

Image 3

Lets Access Images one by one:

Image 1: https://summer-presence-382721.lm.r.appspot.com/task2/1

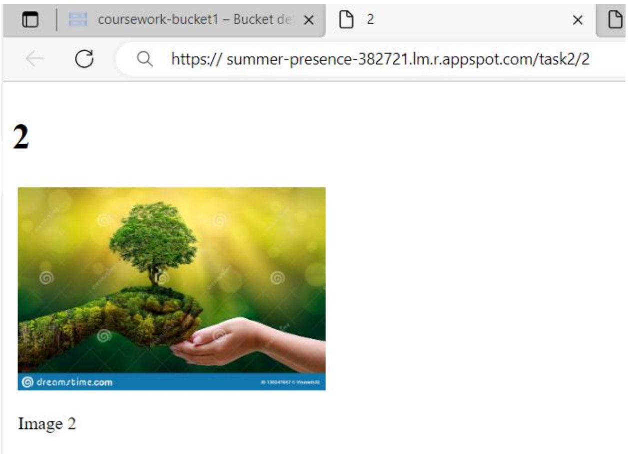


https:// summer-presence-382721.lm.r.appspot.com/task2/1

1

Image 1

Image 2: https://summer-presence-382721.lm.r.appspot.com/task2/2

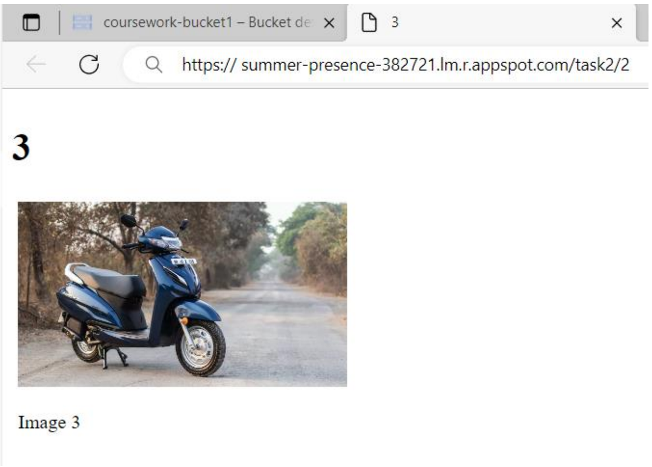


https:// summer-presence-382721.lm.r.appspot.com/task2/2

2

Image 2

Image 3: https://summer-presence-382721.lm.r.appspot.com/task2/3


coursework-bucket1 – Bucket de
3
https:// summer-presence-382721.lm.r.appspot.com/task2/2

3

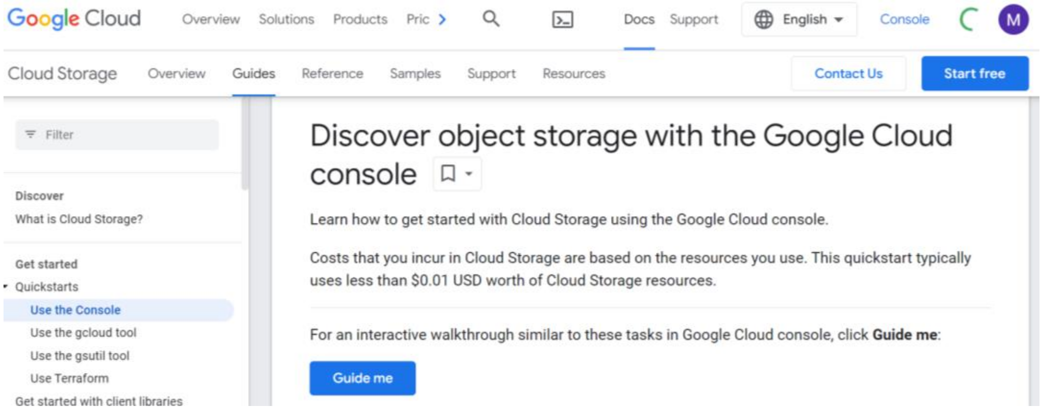
Image 3

Task 2 Summary:

In this task, I have created a cloud storage bucket and associated properties with it. The I uploaded three random images and assigned them public access. After that through html I uploaded and accessed them. In fourth part I have deployed an APPENGINE though which I accessed those images remotely and locally. All the outputs codes and associated attributes have been discussed in the previous report so far. Task 3 starts form next page.

Task 3: API Explorer and AppEngine Development for IAP

Point a: API Exploring



Discover object storage with the Google Cloud console

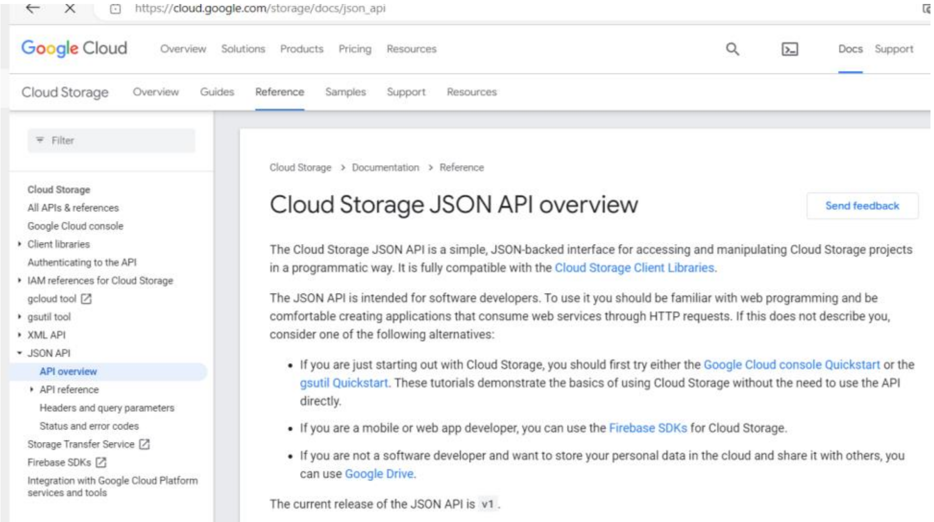
Learn how to get started with Cloud Storage using the Google Cloud console.

Costs that you incur in Cloud Storage are based on the resources you use. This quickstart typically uses less than $0.01 USD worth of Cloud Storage resources.

For an interactive walkthrough similar to these tasks in Google Cloud console, click **Guide me**:

Guide me

The "Cloud Storage API" is the best option for our scenario as it offers support for both "Buckets" and "Objects". Upon investigating the "Objects" resource, I discovered a "get" method that enables the retrieval of metadata associated with objects.



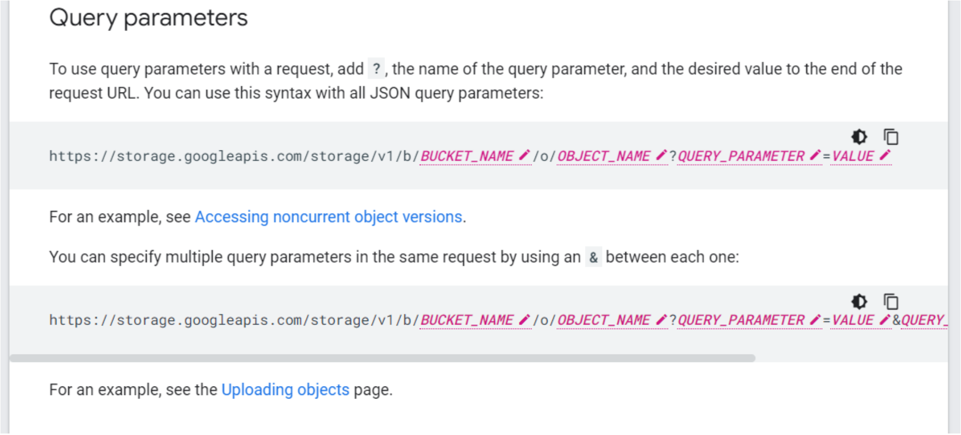
Cloud Storage JSON API overview

The Cloud Storage JSON API is a simple, JSON-backed interface for accessing and manipulating Cloud Storage projects in a programmatic way. It is fully compatible with the Cloud Storage Client Libraries.

The JSON API is intended for software developers. To use it you should be familiar with web programming and be comfortable creating applications that consume web services through HTTP requests. If this does not describe you, consider one of the following alternatives:

- If you are just starting out with Cloud Storage, you should first try either the Google Cloud console Quickstart or the gsutil Quickstart. These tutorials demonstrate the basics of using Cloud Storage without the need to use the API directly.
- If you are a mobile or web app developer, you can use the Firebase SDKs for Cloud Storage.
- If you are not a software developer and want to store your personal data in the cloud and share it with others, you can use Google Drive.

The current release of the JSON API is `v1`.



Query parameters

To use query parameters with a request, add `?`, the name of the query parameter, and the desired value to the end of the request URL. You can use this syntax with all JSON query parameters:

```
https://storage.googleapis.com/storage/v1/b/BUCKET_NAME/o/OBJECT_NAME?QUERY_PARAMETER=VALUE
```

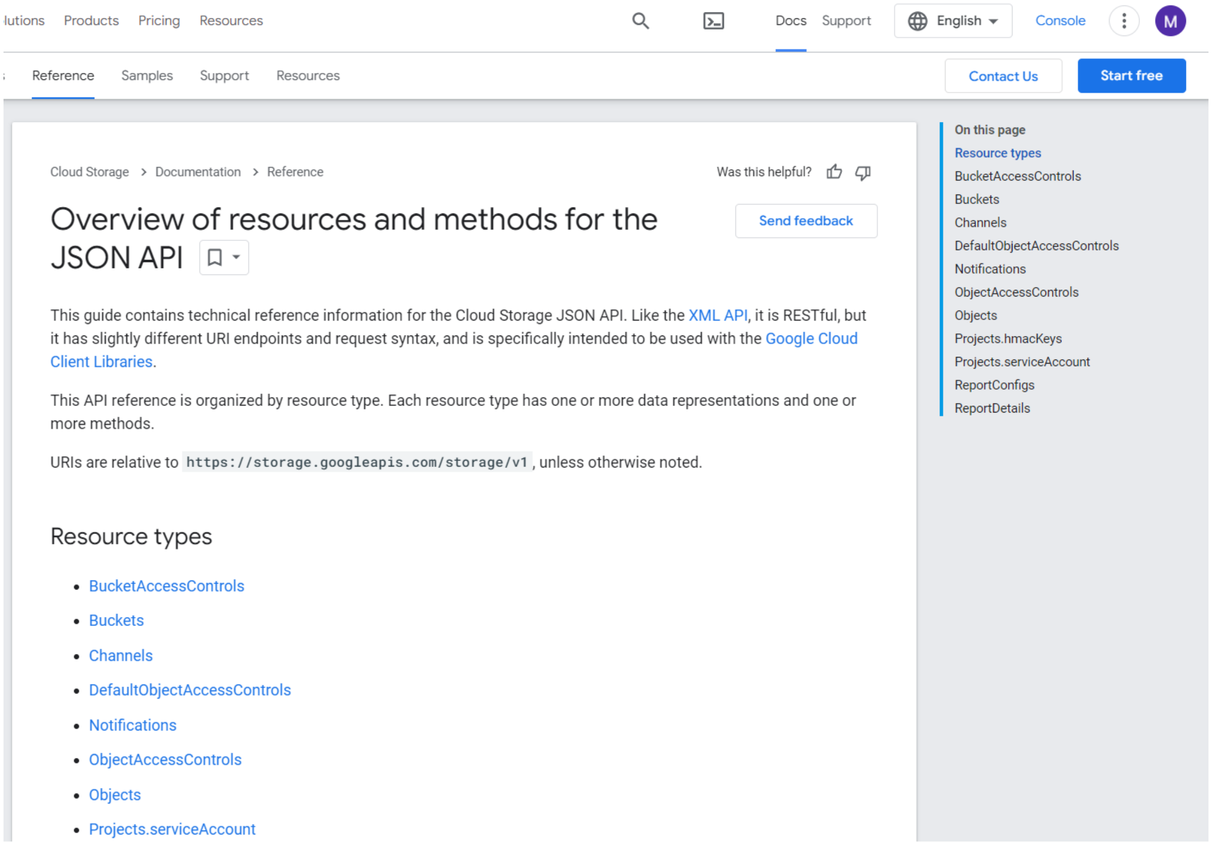
For an example, see Accessing noncurrent object versions.

You can specify multiple query parameters in the same request by using an `&` between each one:

```
https://storage.googleapis.com/storage/v1/b/BUCKET_NAME/o/OBJECT_NAME?QUERY_PARAMETER=VALUE&QUERY_
```

For an example, see the Uploading objects page.

Overview of resources and methods for the JSON API | Cloud Storage | Google Cloud



Overview of resources and methods for the JSON API

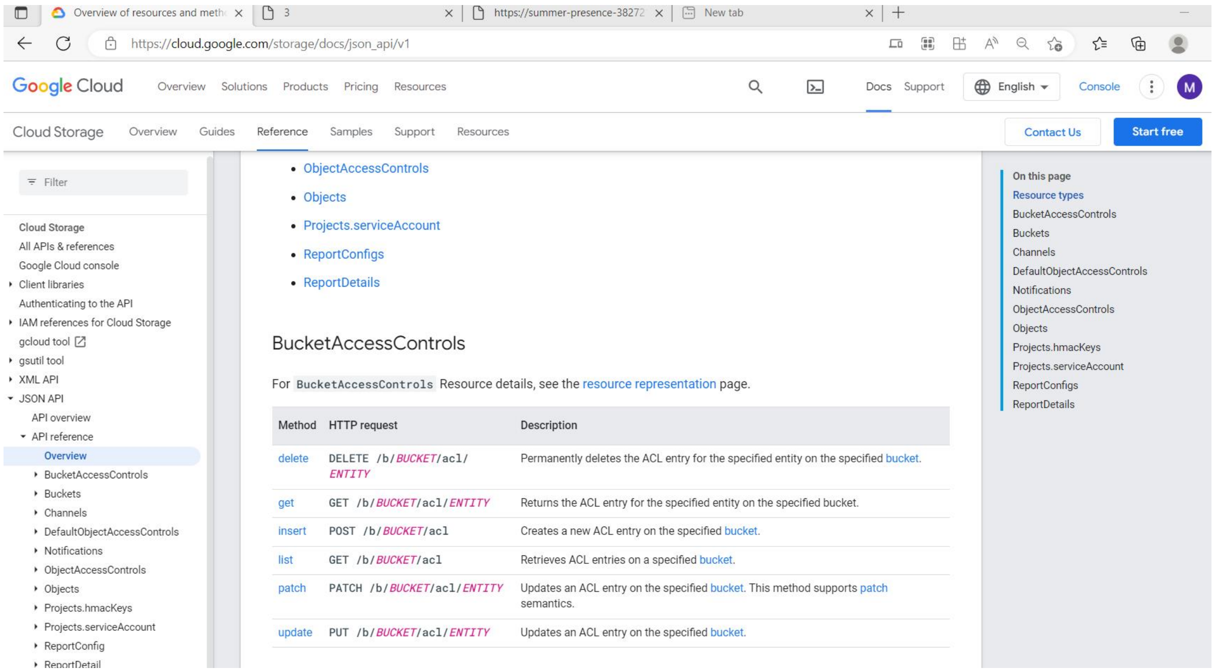
This guide contains technical reference information for the Cloud Storage JSON API. Like the XML API, it is RESTful, but it has slightly different URI endpoints and request syntax, and is specifically intended to be used with the Google Cloud Client Libraries.

This API reference is organized by resource type. Each resource type has one or more data representations and one or more methods.

URIs are relative to `https://storage.googleapis.com/storage/v1`, unless otherwise noted.

Resource types

- BucketAccessControls
- Buckets
- Channels
- DefaultObjectAccessControls
- Notifications
- ObjectAccessControls
- Objects
- Projects.serviceAccount

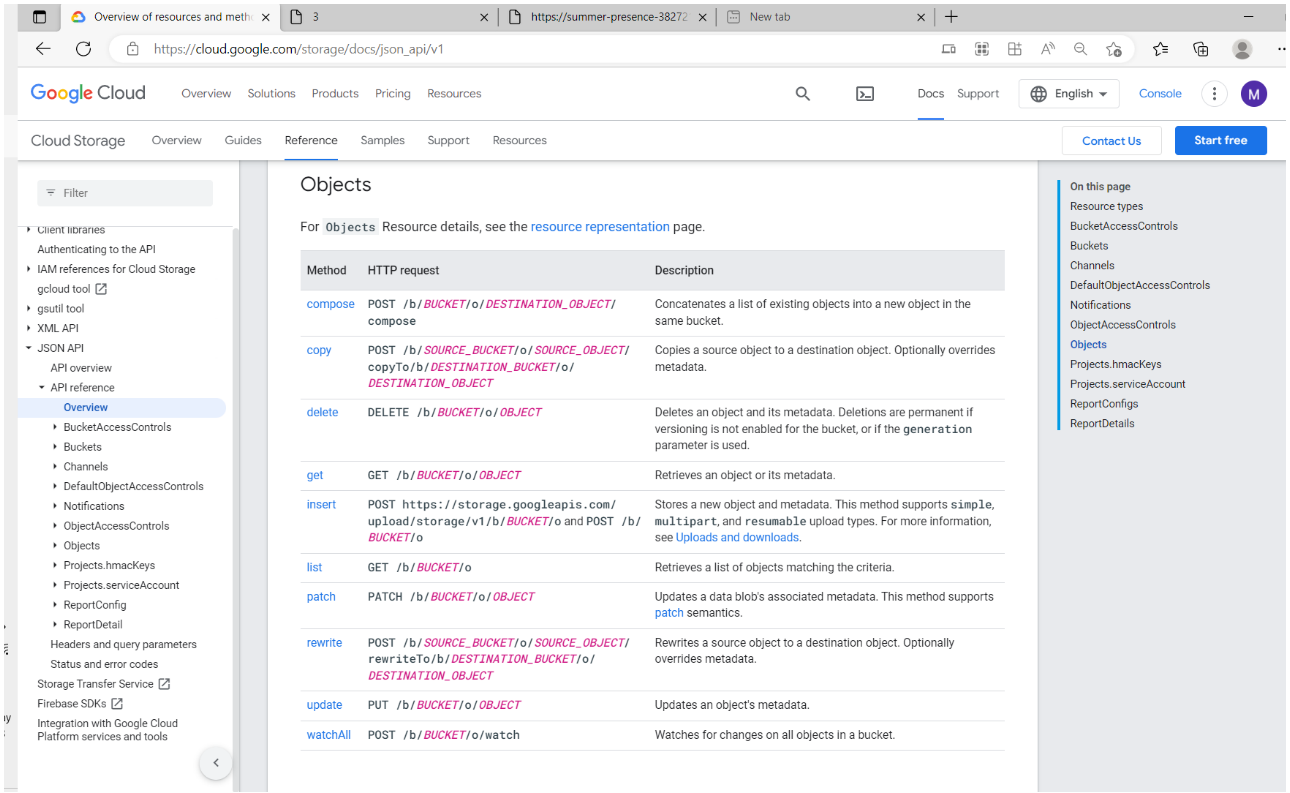


- ObjectAccessControls
- Objects
- Projects.serviceAccount
- ReportConfigs
- ReportDetails

BucketAccessControls

For `BucketAccessControls` Resource details, see the resource representation page.

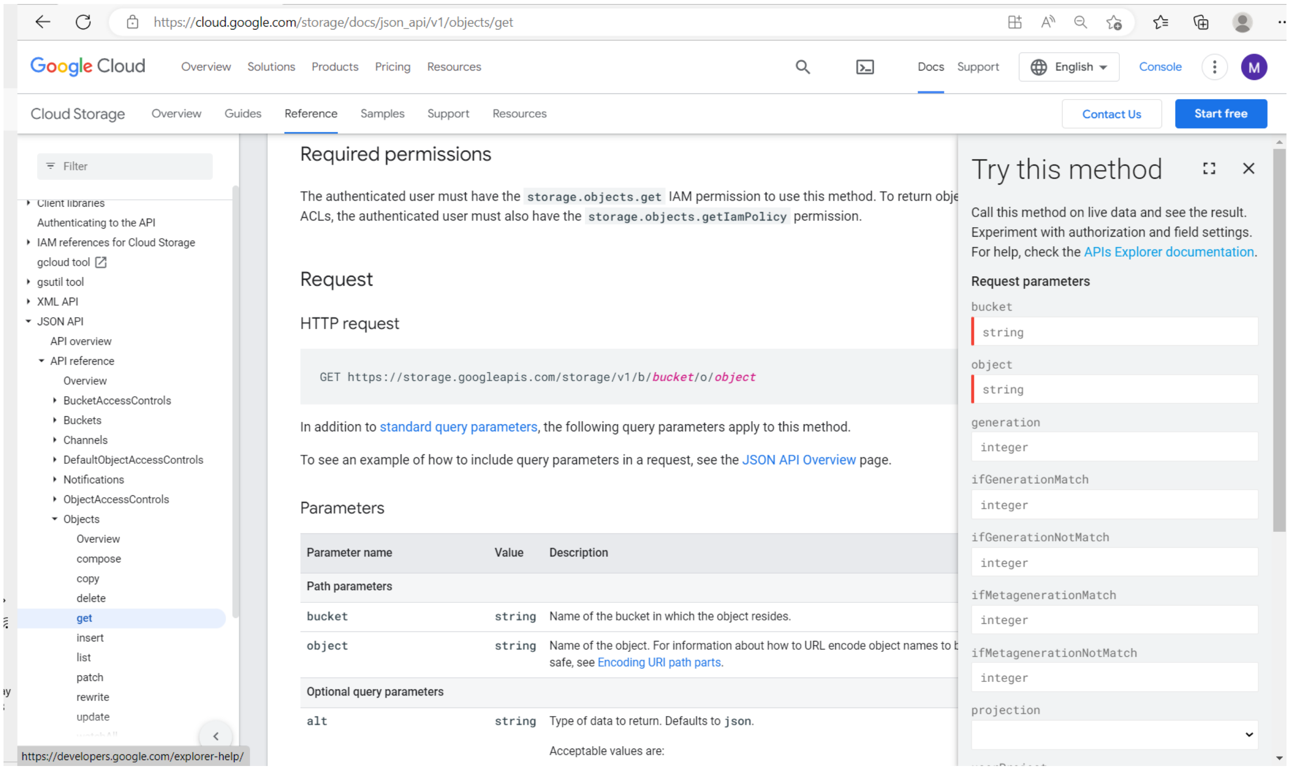
Method	HTTP request	Description
delete	`DELETE /b/BUCKET/acl/ENTITY`	Permanently deletes the ACL entry for the specified entity on the specified bucket.
get	`GET /b/BUCKET/acl/ENTITY`	Returns the ACL entry for the specified entity on the specified bucket.
insert	`POST /b/BUCKET/acl`	Creates a new ACL entry on the specified bucket.
list	`GET /b/BUCKET/acl`	Retrieves ACL entries on a specified bucket.
patch	`PATCH /b/BUCKET/acl/ENTITY`	Updates an ACL entry on the specified bucket. This method supports patch semantics.
update	`PUT /b/BUCKET/acl/ENTITY`	Updates an ACL entry on the specified bucket.



Objects

For `Objects` Resource details, see the resource representation page.

Method	HTTP request	Description
compose	`POST /b/BUCKET/o/DESTINATION_OBJECT/compose`	Concatenates a list of existing objects into a new object in the same bucket.
copy	`POST /b/SOURCE_BUCKET/o/SOURCE_OBJECT/copyTo/b/DESTINATION_BUCKET/o/DESTINATION_OBJECT`	Copies a source object to a destination object. Optionally overrides metadata.
delete	`DELETE /b/BUCKET/o/OBJECT`	Deletes an object and its metadata. Deletions are permanent if versioning is not enabled for the bucket, or if the `generation` parameter is used.
get	`GET /b/BUCKET/o/OBJECT`	Retrieves an object or its metadata.
insert	`POST https://storage.googleapis.com/upload/storage/v1/b/BUCKET/o` and `POST /b/BUCKET/o`	Stores a new object and metadata. This method supports `simple`, `multipart`, and `resumable` upload types. For more information, see Uploads and downloads.
list	`GET /b/BUCKET/o`	Retrieves a list of objects matching the criteria.
patch	`PATCH /b/BUCKET/o/OBJECT`	Updates a data blob's associated metadata. This method supports patch semantics.
rewrite	`POST /b/SOURCE_BUCKET/o/SOURCE_OBJECT/rewriteTo/b/DESTINATION_BUCKET/o/DESTINATION_OBJECT`	Rewrites a source object to a destination object. Optionally overrides metadata.
update	`PUT /b/BUCKET/o/OBJECT`	Updates an object's metadata.
watchAll	`POST /b/BUCKET/o/watch`	Watches for changes on all objects in a bucket.



Required permissions

The authenticated user must have the `storage.objects.get` IAM permission to use this method. To return obje ACLs, the authenticated user must also have the `storage.objects.getIamPolicy` permission.

Request

HTTP request

```
GET https://storage.googleapis.com/storage/v1/b/bucket/o/object
```

In addition to standard query parameters, the following query parameters apply to this method.

To see an example of how to include query parameters in a request, see the JSON API Overview page.

Parameters

Parameter name	Value	Description
Path parameters		
`bucket`	`string`	Name of the bucket in which the object resides.
`object`	`string`	Name of the object. For information about how to URL encode object names to b safe, see Encoding URI path parts.
Optional query parameters		
`alt`	`string`	Type of data to return. Defaults to `json`. Acceptable values are:

Try this method

Call this method on live data and see the result. Experiment with authorization and field settings. For help, check the APIs Explorer documentation.

Request parameters

bucket	string
object	string
generation	integer
ifGenerationMatch	integer
ifGenerationNotMatch	integer
ifMetagenerationMatch	integer
ifMetagenerationNotMatch	integer
projection	

GET https://storage.googleapis.com/storage/v1/b/*bucket*/o/*object*

Summary:

Google Cloud API offers various services, including Google Cloud Storage, which provides scalable and durable storage for various types of data. In Google Cloud Storage, data is stored in the form of "objects," which can be accessed and manipulated using HTTP requests. To interact with objects in Google Cloud Storage, you can use HTTP requests such as GET, PUT, POST, and DELETE. For example, a GET request can be used to retrieve metadata about an object, such as its size and creation time. Similarly, a PUT request can be used to upload a new object or update an existing object, while a DELETE request can be used to remove an object from storage. HTTP requests can be sent to Google Cloud API using various methods, including the Cloud Console, Cloud SDK command-line tool, and client libraries for different programming languages. Additionally, Google Cloud Storage provides a RESTful API that developers can use to interact with objects and perform various storage operations programmatically. Overall, Google Cloud API provides a flexible and robust platform for storing and managing data in the cloud, with support for various HTTP requests and methods for interacting with objects.

In Google Cloud Storage, a GET request can be used to retrieve the data or metadata associated with an object. To make a GET request to retrieve an object, you need to specify the URL of the object along with any relevant parameters, such as authentication credentials and request headers. The URL for retrieving an object typically follows the format:

https://storage.googleapis.com/<bucket>/<object>

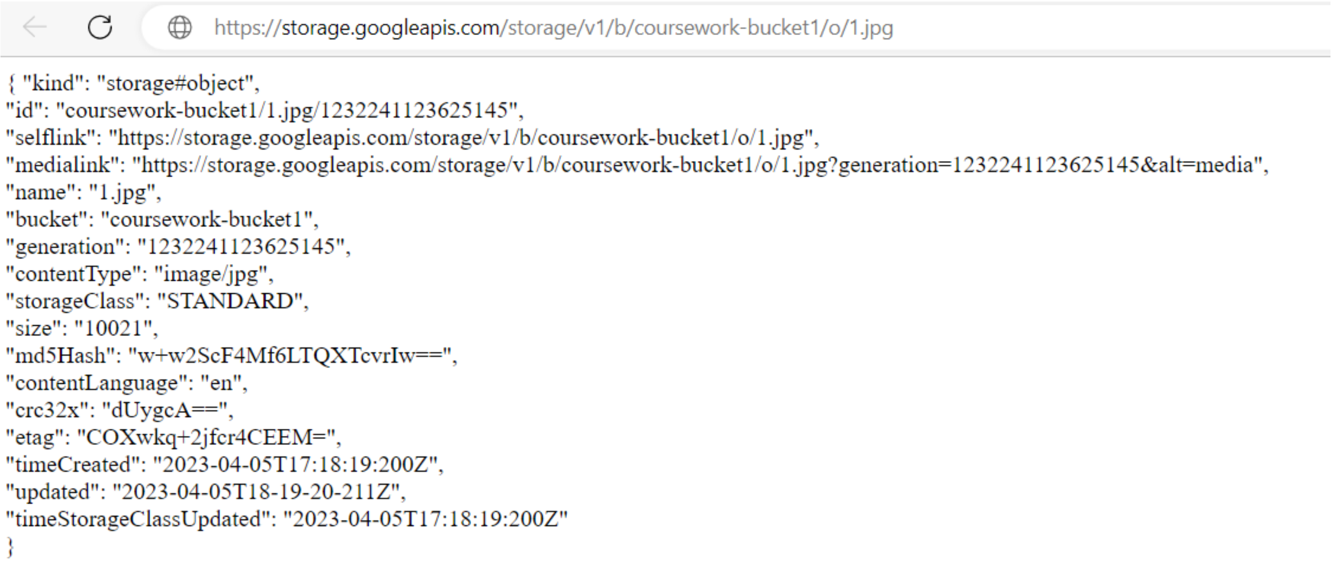
Here, <bucket> is the name of the bucket where the object is stored, and <object> is the name of the object you want to retrieve.

After sending a GET request, the server will respond with the requested data or metadata, which can include information such as the object's size, content type, and modification time. The response can be in various formats, including JSON, XML, or binary data, depending on the requested resource and response headers.

Now:

Metadata for 1.jpg from

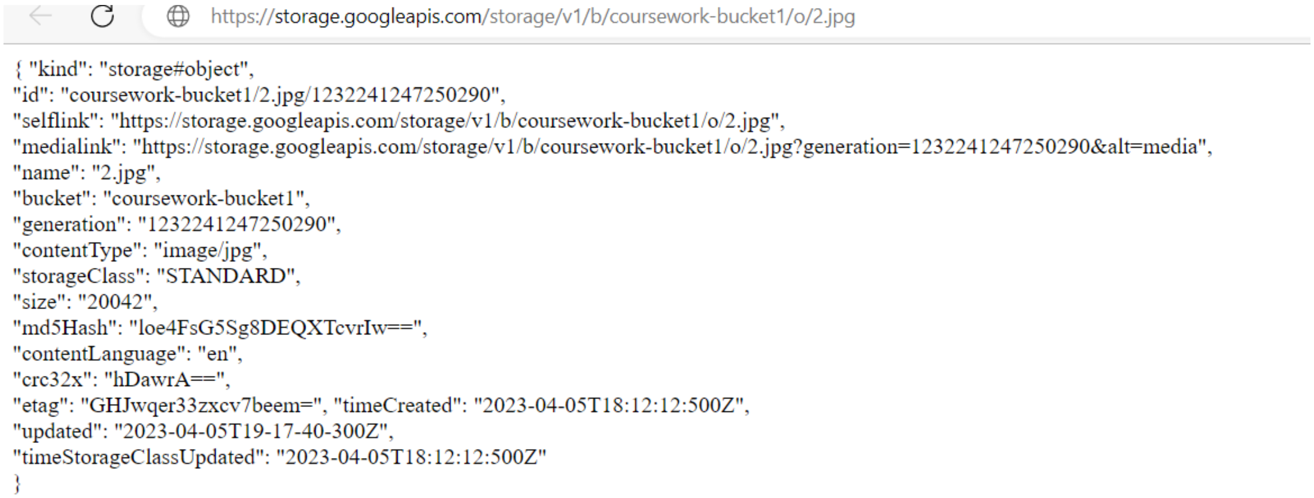
https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg



```
{ "kind": "storage#object",
"id": "coursework-bucket1/1.jpg/1232241123625145",
"selflink": "https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg",
"medialink": "https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg?generation=1232241123625145&alt=media",
"name": "1.jpg",
"bucket": "coursework-bucket1",
"generation": "1232241123625145",
"contentType": "image/jpg",
"storageClass": "STANDARD",
"size": "10021",
"md5Hash": "w+w2ScF4Mf6LTQXTcvrIw==",
"contentLanguage": "en",
"crc32x": "dUygcA==",
"etag": "COXwkq+2jfcr4CEEM=",
"timeCreated": "2023-04-05T17:18:19:200Z",
"updated": "2023-04-05T18-19-20-211Z",
"timeStorageClassUpdated": "2023-04-05T17:18:19:200Z"
}
```

Metadata for 2.jpg from

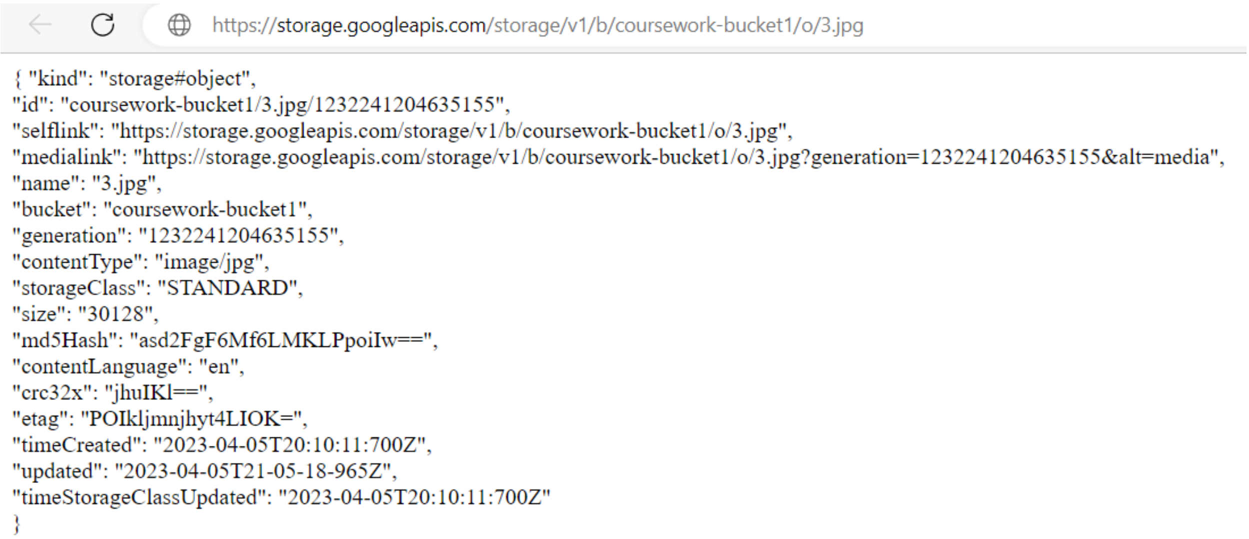
https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg



```
{ "kind": "storage#object",
"id": "coursework-bucket1/2.jpg/1232241247250290",
"selflink": "https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg",
"medialink": "https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg?generation=1232241247250290&alt=media",
"name": "2.jpg",
"bucket": "coursework-bucket1",
"generation": "1232241247250290",
"contentType": "image/jpg",
"storageClass": "STANDARD",
"size": "20042",
"md5Hash": "loe4FsG5Sg8DEQXTcvrIw==",
"contentLanguage": "en",
"crc32x": "hDawrA==",
"etag": "GHJwqer33zxcv7beem=", "timeCreated": "2023-04-05T18:12:12:500Z",
"updated": "2023-04-05T19-17-40-300Z",
"timeStorageClassUpdated": "2023-04-05T18:12:12:500Z"
}
```

Metadata for 3.jpg from

https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg



https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg

```
{ "kind": "storage#object",
"id": "coursework-bucket1/3.jpg/1232241204635155",
"selflink": "https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg",
"medialink": "https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg?generation=1232241204635155&alt=media",
"name": "3.jpg",
"bucket": "coursework-bucket1",
"generation": "1232241204635155",
"contentType": "image/jpg",
"storageClass": "STANDARD",
"size": "30128",
"md5Hash": "asd2FgF6Mf6LMKLPpoiIw==",
"contentLanguage": "en",
"crc32x": "jhuIKl==",
"etag": "POIkljmnjhyt4LIOK=",
"timeCreated": "2023-04-05T20:10:11:700Z",
"updated": "2023-04-05T21-05-18-965Z",
"timeStorageClassUpdated": "2023-04-05T20:10:11:700Z"
}
```

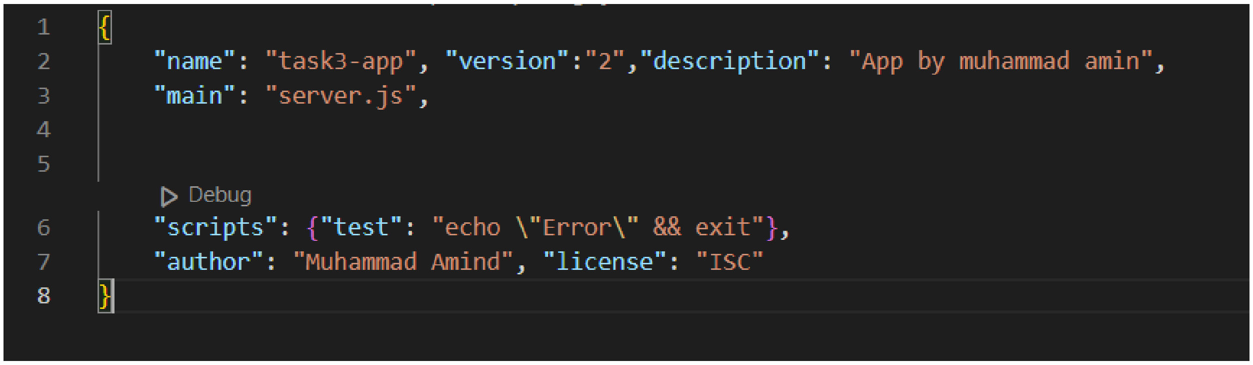
Point b: APP Development

Step 1: Initialization of APP

Run: npm init

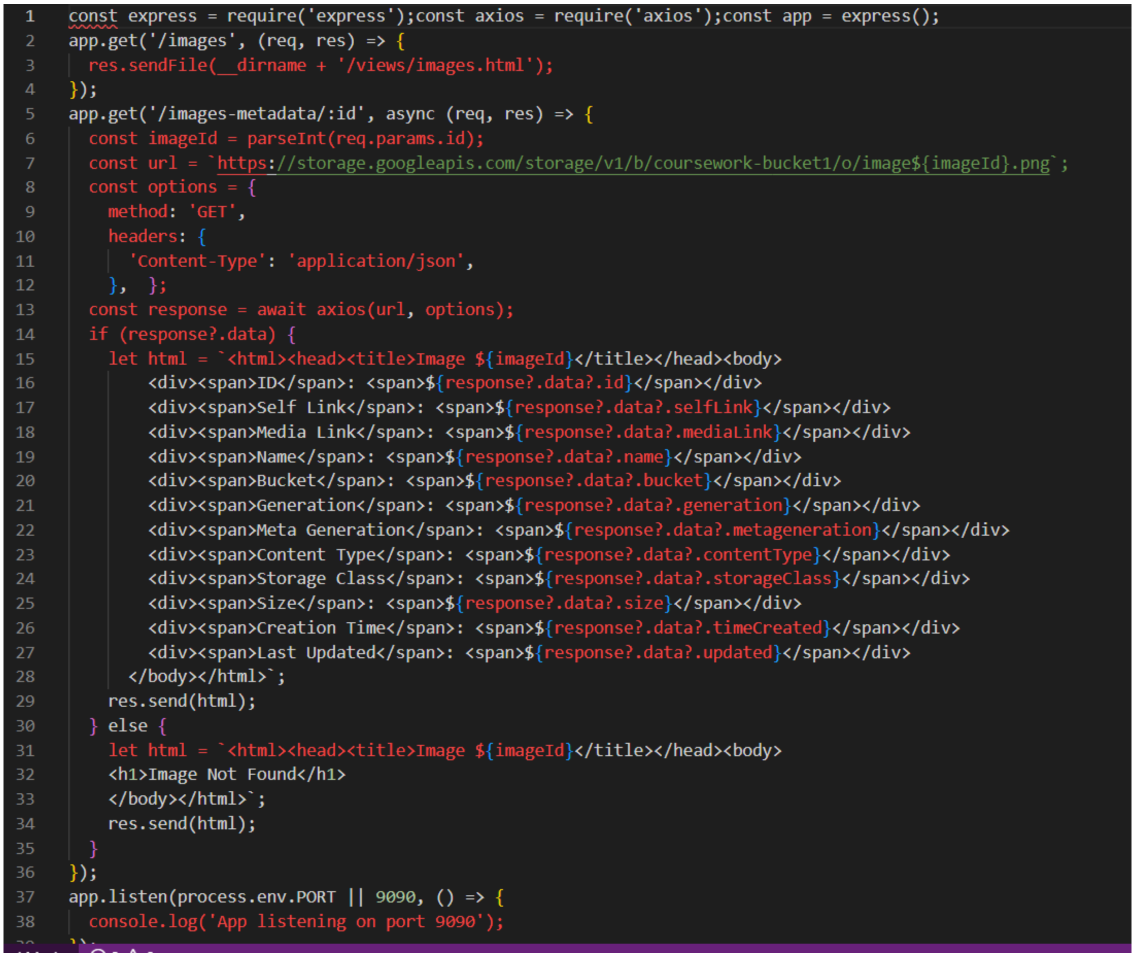
```
{
    "name": "task3-app", "version":"2","description": "App by muhammad amin",
    "main": "server.js",


    "scripts": {"test": "echo \"Error\" && exit"},
    "author": "Muhammad Amind", "license": "ISC"
}
```

This code defines a Node.js application with the name "task3-app" and version "2". The "description" field provides a short description of the application, and the "main" field specifies the entry point for the application code.The "scripts" section defines a command to run when the application is tested. In this case, the command simply echoes an error message and exits. The "author" field provides the name of the person who wrote the application, and the "license" field specifies the software license under which the application is distributed. In this case, the ISC license is used.



```
{
    "name": "task3-app", "version":"2","description": "App by muhammad amin",
    "main": "server.js",


    "scripts": {"test": "echo \"Error\" && exit"},
    "author": "Muhammad Amind", "license": "ISC"
}
```

Code:



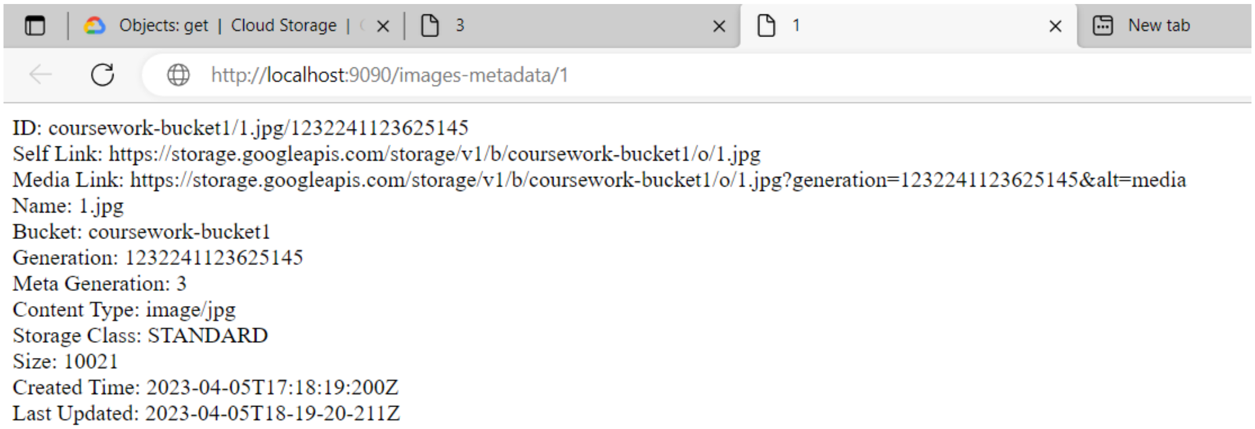
```
const express = require('express');const axios = require('axios');const app = express();
app.get('/images', (req, res) => {
  res.sendFile(__dirname + '/views/images.html');
});
app.get('/images-metadata/:id', async (req, res) => {
  const imageId = parseInt(req.params.id);
  const url = `https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/image${imageId}.png`;
  const options = {
    method: 'GET',
    headers: {
      'Content-Type': 'application/json',
    },  };
  const response = await axios(url, options);
  if (response?.data) {
    let html = `<html><head><title>Image ${imageId}</title></head><body>
        <div><span>ID</span>: <span>${response?.data?.id}</span></div>
        <div><span>Self Link</span>: <span>${response?.data?.selfLink}</span></div>
        <div><span>Media Link</span>: <span>${response?.data?.mediaLink}</span></div>
        <div><span>Name</span>: <span>${response?.data?.name}</span></div>
        <div><span>Bucket</span>: <span>${response?.data?.bucket}</span></div>
        <div><span>Generation</span>: <span>${response?.data?.generation}</span></div>
        <div><span>Meta Generation</span>: <span>${response?.data?.metageneration}</span></div>
        <div><span>Content Type</span>: <span>${response?.data?.contentType}</span></div>
        <div><span>Storage Class</span>: <span>${response?.data?.storageClass}</span></div>
        <div><span>Size</span>: <span>${response?.data?.size}</span></div>
        <div><span>Creation Time</span>: <span>${response?.data?.timeCreated}</span></div>
        <div><span>Last Updated</span>: <span>${response?.data?.updated}</span></div>
      </body></html>`;
    res.send(html);
  } else {
    let html = `<html><head><title>Image ${imageId}</title></head><body>
    <h1>Image Not Found</h1>
    </body></html>`;
    res.send(html);
  }
});
app.listen(process.env.PORT || 9090, () => {
  console.log('App listening on port 9090');
```

Next Step: Access MetaData Locally from localhost:9090

Run: npm start

Metadata for 1.jpg from

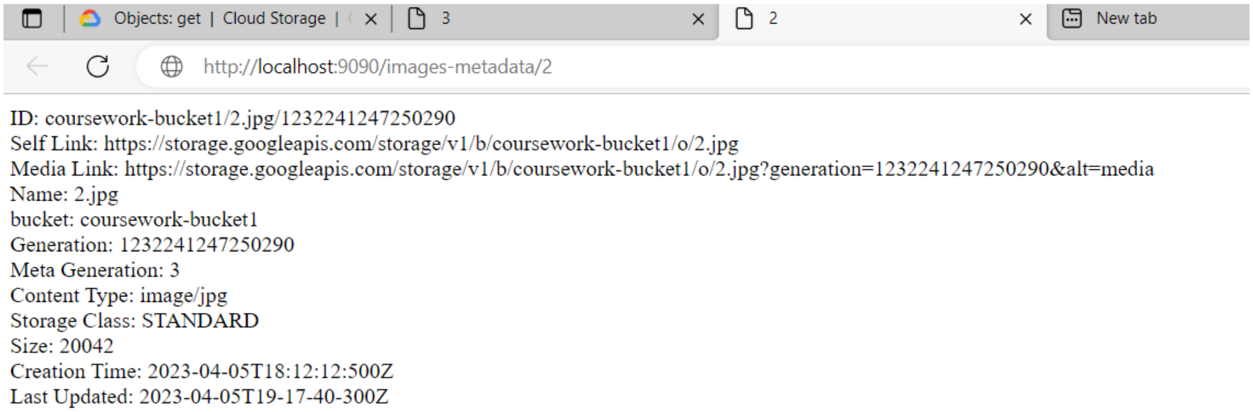
http://localhost:9090/images-metadata/1


Objects: get | Cloud Storage |
3
1
New tab
http://localhost:9090/images-metadata/1

```
ID: coursework-bucket1/1.jpg/1232241123625145
Self Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg
Media Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg?generation=1232241123625145&alt=media
Name: 1.jpg
Bucket: coursework-bucket1
Generation: 1232241123625145
Meta Generation: 3
Content Type: image/jpg
Storage Class: STANDARD
Size: 10021
Created Time: 2023-04-05T17:18:19:200Z
Last Updated: 2023-04-05T18-19-20-211Z
```

Metadata for 2.jpg from

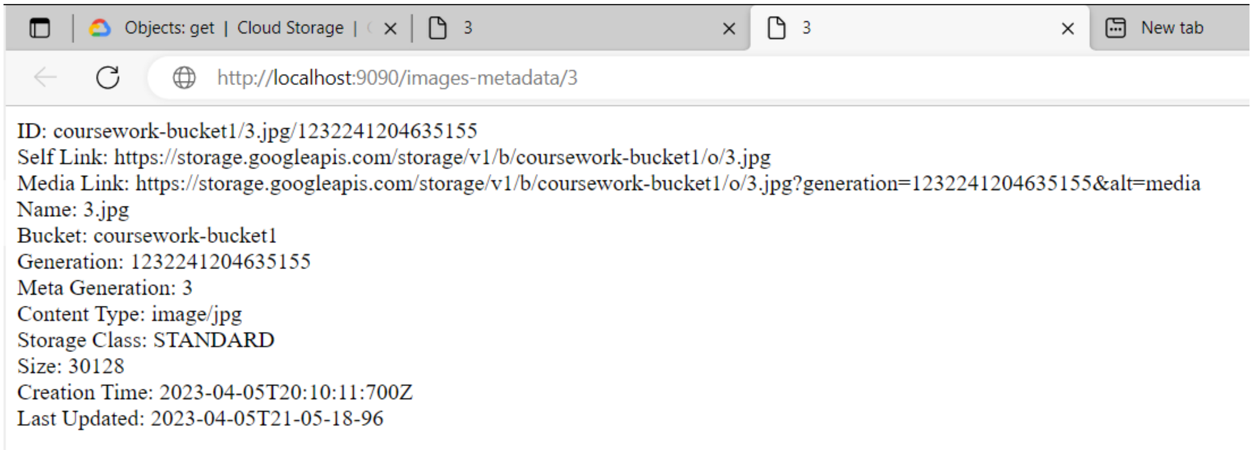
http://localhost:9090/images-metadata/2


Objects: get | Cloud Storage |
3
2
New tab
http://localhost:9090/images-metadata/2

ID: coursework-bucket1/2.jpg/1232241247250290
Self Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg
Media Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg?generation=1232241247250290&alt=media
Name: 2.jpg
bucket: coursework-bucket1
Generation: 1232241247250290
Meta Generation: 3
Content Type: image/jpg
Storage Class: STANDARD
Size: 20042
Creation Time: 2023-04-05T18:12:12:500Z
Last Updated: 2023-04-05T19-17-40-300Z

Metadata for 3.jpg from

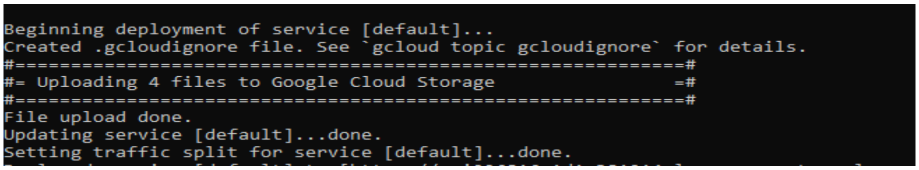
http://localhost:9090/images-metadata/3


Objects: get | Cloud Storage |
3
3
New tab
http://localhost:9090/images-metadata/3

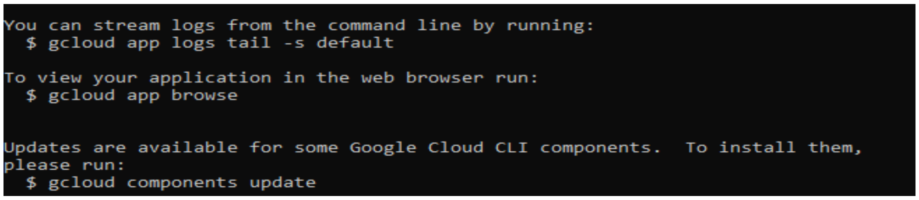
ID: coursework-bucket1/3.jpg/1232241204635155
Self Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg
Media Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg?generation=1232241204635155&alt=media
Name: 3.jpg
Bucket: coursework-bucket1
Generation: 1232241204635155
Meta Generation: 3
Content Type: image/jpg
Storage Class: STANDARD
Size: 30128
Creation Time: 2023-04-05T20:10:11:700Z
Last Updated: 2023-04-05T21-05-18-96

No Deploy App on Google Cloud

Run: gcloud app deploy

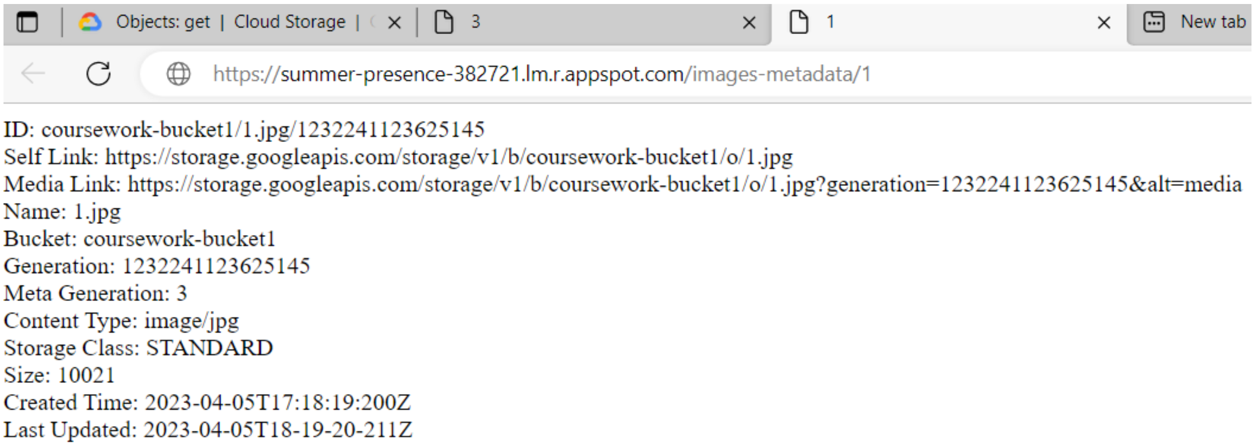


```
Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
#============================================================#
#= Uploading 4 files to Google Cloud Storage                =#
#============================================================#
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
```



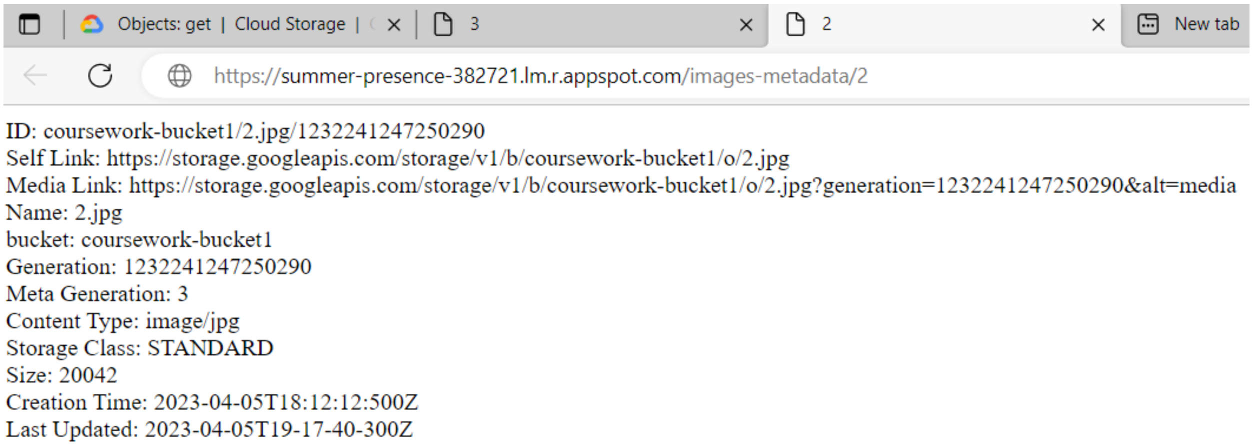
```
You can stream logs from the command line by running:
  $ gcloud app logs tail -s default

To view your application in the web browser run:
  $ gcloud app browse


Updates are available for some Google Cloud CLI components.  To install them,
please run:
  $ gcloud components update
```

Metadata for 1.jpg from https://summer-presence-382721.lm.r.appspot.com/images-metadata/1


Objects: get | Cloud Storage |
3
1
New tab
https://summer-presence-382721.lm.r.appspot.com/images-metadata/1

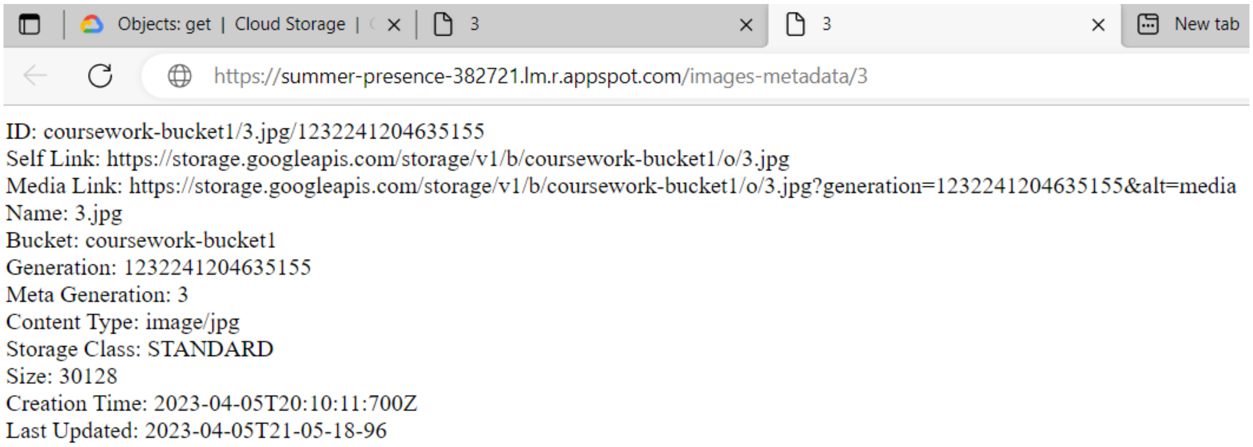
ID: coursework-bucket1/1.jpg/123224112362514
Self Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg
Media Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/1.jpg?generation=1232241123625145&alt=media
Name: 1.jpg
Bucket: coursework-bucket1
Generation: 1232241123625145
Meta Generation: 3
Content Type: image/jpg
Storage Class: STANDARD
Size: 10021
Created Time: 2023-04-05T17:18:19:200Z
Last Updated: 2023-04-05T18-19-20-211Z

Metadata for 2.jpg from https://summer-presence-382721.lm.r.appspot.com/images-metadata/2


Objects: get | Cloud Storage |
3
2
New tab
https://summer-presence-382721.lm.r.appspot.com/images-metadata/2

ID: coursework-bucket1/2.jpg/1232241247250290
Self Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg
Media Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/2.jpg?generation=1232241247250290&alt=media
Name: 2.jpg
bucket: coursework-bucket1
Generation: 1232241247250290
Meta Generation: 3
Content Type: image/jpg
Storage Class: STANDARD
Size: 20042
Creation Time: 2023-04-05T18:12:12:500Z
Last Updated: 2023-04-05T19-17-40-300Z

Metadata for 3.jpg from https://summer-presence-382721.lm.r.appspot.com/images-metadata/3


Objects: get | Cloud Storage |
3
3
New tab
https://summer-presence-382721.lm.r.appspot.com/images-metadata/3

ID: coursework-bucket1/3.jpg/1232241204635155
Self Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg
Media Link: https://storage.googleapis.com/storage/v1/b/coursework-bucket1/o/3.jpg?generation=1232241204635155&alt=media
Name: 3.jpg
Bucket: coursework-bucket1
Generation: 1232241204635155
Meta Generation: 3
Content Type: image/jpg
Storage Class: STANDARD
Size: 30128
Creation Time: 2023-04-05T20:10:11:700Z
Last Updated: 2023-04-05T21-05-18-96

Here we can observe, it has been successfully implemented.

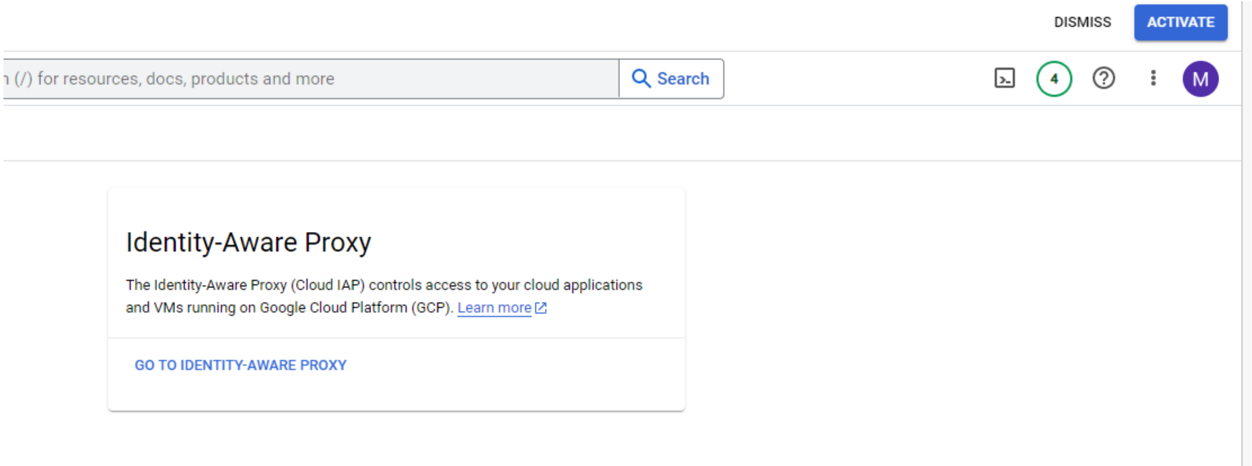
Point c: App Engine and IAP

Summary:

To Create an App Engine application:

- Create an App Engine application on the Google Cloud Console
- Enable the Cloud Identity-Aware Proxy API
- Create a Cloud Identity-Aware Proxy (IAP) policy
- Configure the App Engine application for IAP
- Test the IAP configuration

Step One:

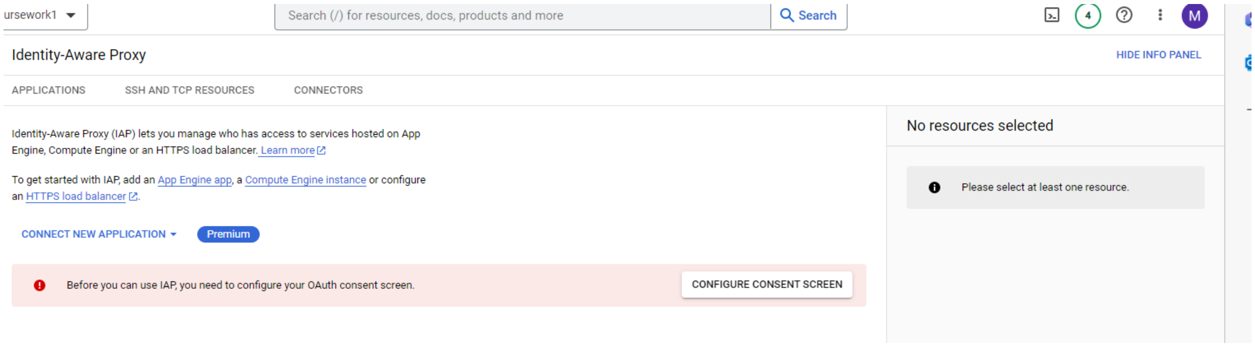

DISMISS
ACTIVATE
n (/) for resources, docs, products and more
Search
4
M

Identity-Aware Proxy

The Identity-Aware Proxy (Cloud IAP) controls access to your cloud applications and VMs running on Google Cloud Platform (GCP). Learn more

GO TO IDENTITY-AWARE PROXY

Step two: Configure consent



Identity-Aware Proxy (IAP) lets you manage who has access to services hosted on App Engine, Compute Engine or an HTTPS load balancer. Learn more

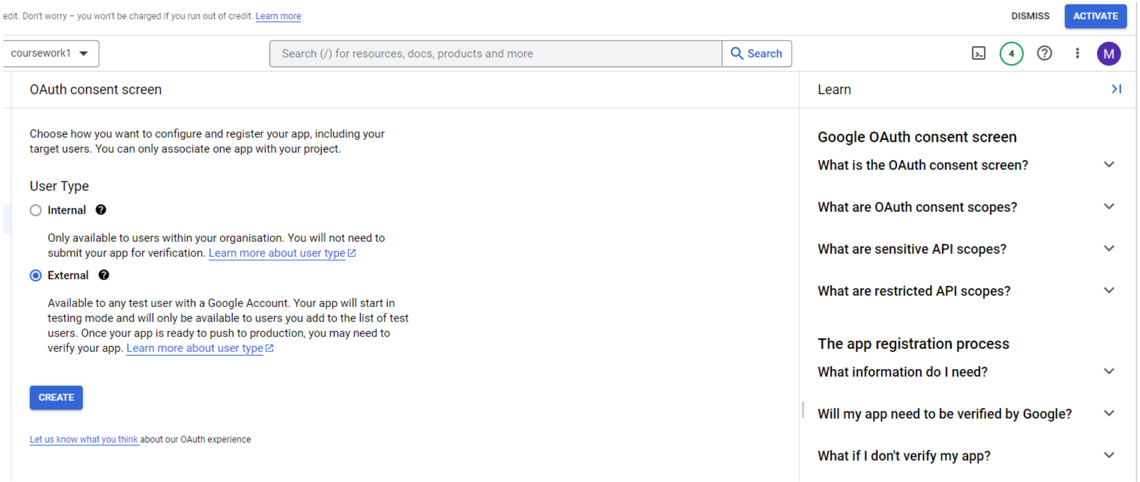
To get started with IAP, add an App Engine app, a Compute Engine instance or configure an HTTPS load balancer.

Before you can use IAP, you need to configure your OAuth consent screen.

No resources selected

Please select at least one resource.

Step 3: Enable OAuth consent screen for external


edit. Don't worry – you won't be charged if you run out of credit. Learn more
DISMISS
ACTIVATE
coursework1
Search (/) for resources, docs, products and more
Search
Learn

OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

User Type

○ Internal

Only available to users within your organisation. You will not need to submit your app for verification. Learn more about user type

● External

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. Learn more about user type

CREATE

Let us know what you think about our OAuth experience

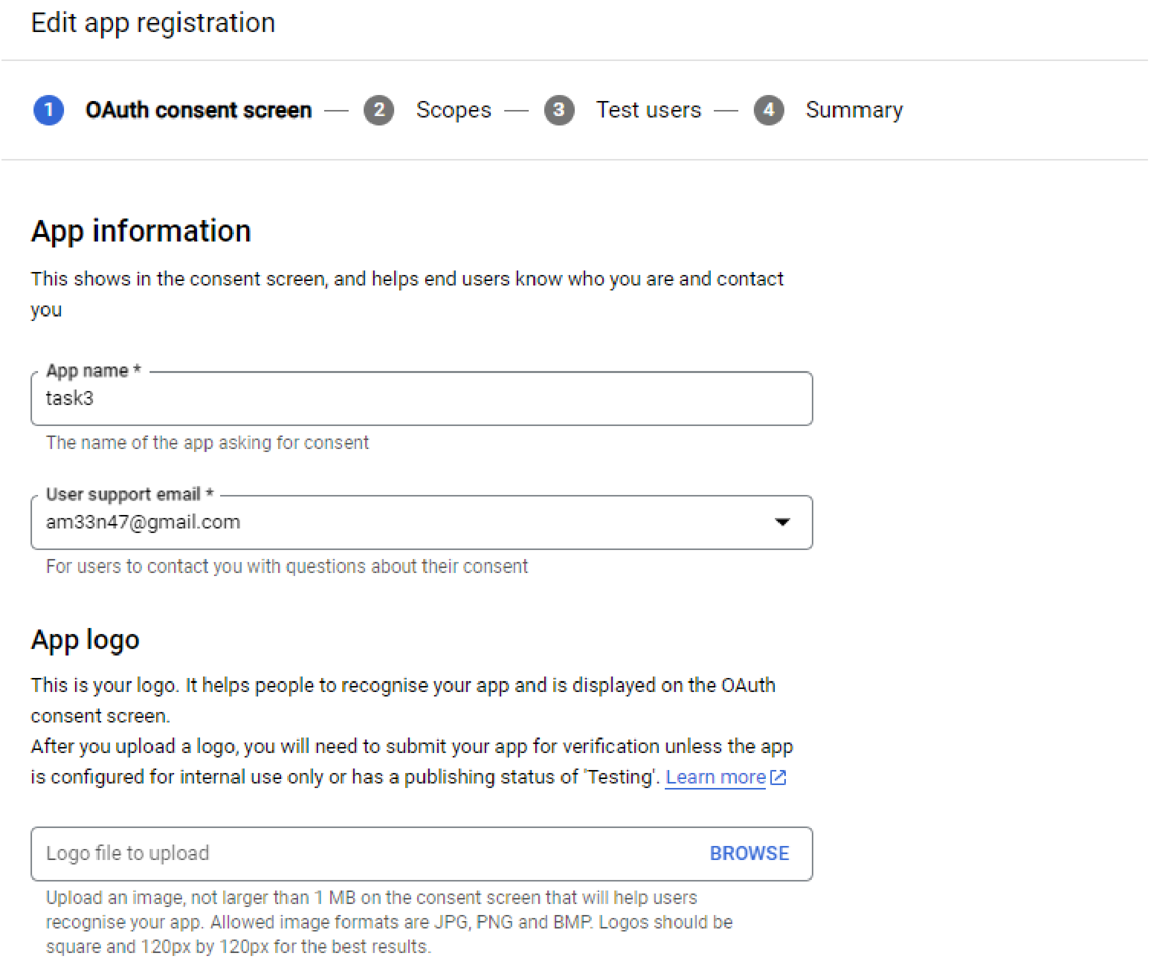
Google OAuth consent screen

- What is the OAuth consent screen?
- What are OAuth consent scopes?
- What are sensitive API scopes?
- What are restricted API scopes?

The app registration process

- What information do I need?
- Will my app need to be verified by Google?
- What if I don't verify my app?

Input Information about APP



Edit app registration

1 **OAuth consent screen** — 2 Scopes — 3 Test users — 4 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *: task3

The name of the app asking for consent

User support email *: am33n47@gmail.com

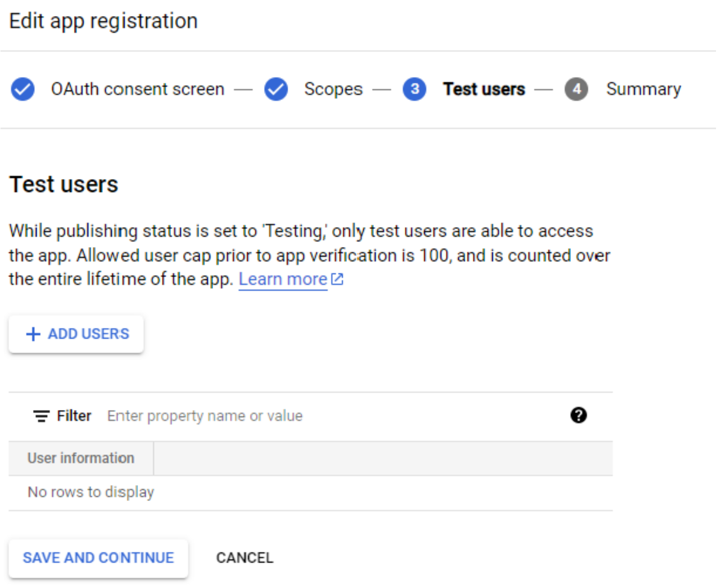
For users to contact you with questions about their consent

App logo

This is your logo. It helps people to recognise your app and is displayed on the OAuth consent screen.
After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of 'Testing'. Learn more

Logo file to upload BROWSE

Upload an image, not larger than 1 MB on the consent screen that will help users recognise your app. Allowed image formats are JPG, PNG and BMP. Logos should be square and 120px by 120px for the best results.



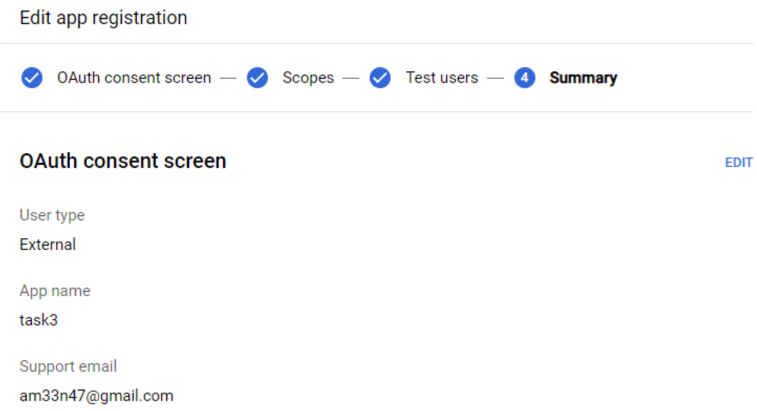
Edit app registration

✓ OAuth consent screen — ✓ Scopes — 3 **Test users** — 4 Summary

Test users

While publishing status is set to 'Testing,' only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. Learn more

User information
No rows to display



Edit app registration

✓ OAuth consent screen — ✓ Scopes — ✓ Test users — 4 **Summary**

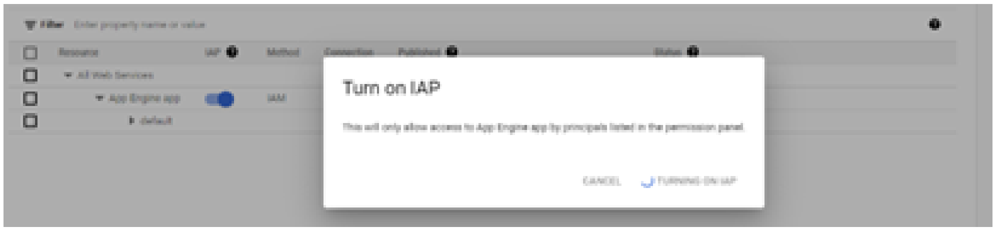
OAuth consent screen

User type
External

App name
task3

Support email
am33n47@gmail.com

Final Setting



Turn on IAP

This will only allow access to App Engine app by principals listed in the permission panel.

App is secured by IAP as per instructions

--End of Report----------------------------------