

# MODUL 5 PRAKTIKUM

**By Ahyar Muawwal S.Kom., M.T**

Google Cloud Storage adalah layanan penyimpanan dari Google Cloud yang ditujukan untuk menyimpan berkas berbasis object. **Object** adalah sebuah data individual yang dapat disimpan pada penyimpanan cloud.

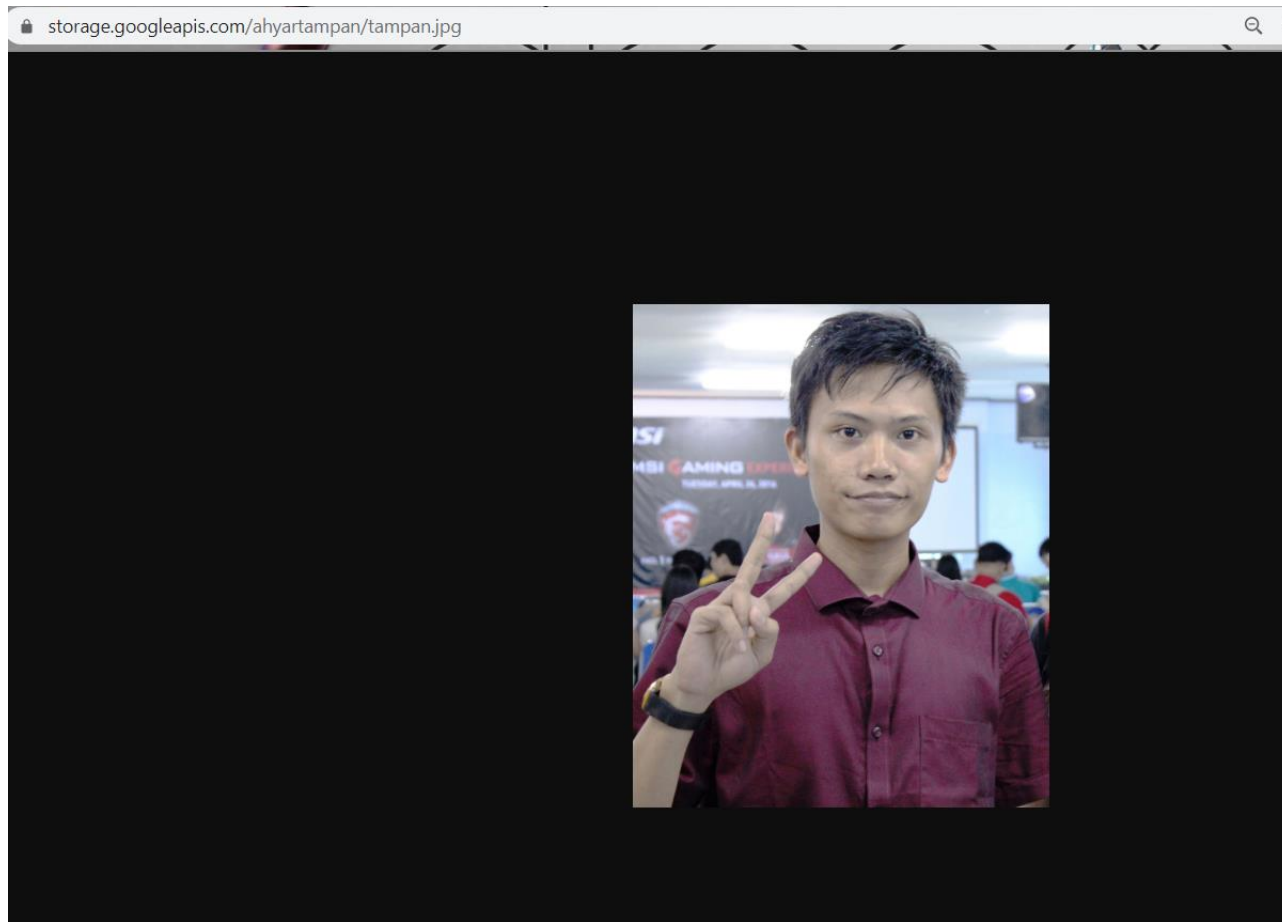
Layanan penyimpanan object seperti ini digunakan untuk menyimpan berkas-berkas yang digunakan pada aplikasi supaya tidak membebani penyimpanan database atau penyimpanan pada komputer/mesin virtual.

Untuk menyimpan berkas ke dalam GCS kita perlu terlebih dulu membuat **bucket** alias wadah penyimpan data kita.

## LATIHAN 1

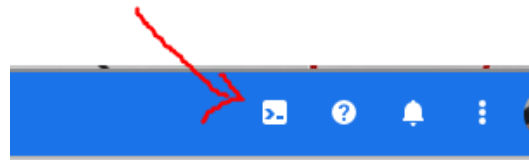
# Google Cloud Storage (GCS) with console

Dalam latihan ini kita akan membuat bucket, mengupload gambar dalam bucket, generate versioning menggunakan command line dan nantinya kita akan membuat gambar yang di upload bisa akses melalui link secara online kemudian mengaktifkan mode versioning dari storage yang kita kelola. Hasilnya sebagai berikut :



Silahkan perhatikan tutorial cara mudah dari kk ganteng dan lanjutkan step projectnya dibawah ni:

1. Silahkan login kedalam akun google cloud computing kalian masing – masing.
2. Pada GCP Console, di **Navigation menu** (☰), click **storage**.



3. Selanjutnya Selanjutnya masuk ke terminal untuk melakukan proses konfigurasi.
4. Kemudian jalankan perintah gsutil mb seperti berikut untuk membuat bucket.

```
gsutil mb -l region_or_multi-region gs://bucket_name
```

catatan :

Perlu diingat bahwa nama bucket haruslah unik. Pada perintah di atas, kita bisa menentukan lokasi penyimpanan yang kita inginkan baik secara regional maupun multi-regional. Sebagai contoh, Anda ingin membuat bucket yang tersimpan di server **Singapura**, maka Anda perlu memasukkan **-l asia-southeast1**. Sedangkan untuk penyimpanan **multi-regional di Asia** gunakan perintah **-l asia**. Cek daftar lokasi tersedia pada slide teori yang telah diberikan

## Regions

All regions are at least 100 miles apart.

Continent	Region Name	Region Description
North America		
	NORTHAMERICA-NORTHEAST1	Montréal
	US-CENTRAL1	Iowa
	US-EAST1	South Carolina
	US-EAST4	Northern Virginia
	US-WEST1	Oregon
	US-WEST2	Los Angeles
	US-WEST3	Salt Lake City
South America		
	SOUTHAMERICA-EAST1	São Paulo

## Regions ⇄

All regions are at least 100 miles apart.

Continent	Region Name	Region Description
Europe		
	EUROPE-NORTH1	Finland
	EUROPE-WEST1	Belgium
	EUROPE-WEST2	London
	EUROPE-WEST3	Frankfurt
	EUROPE-WEST4	Netherlands
	EUROPE-WEST6	Zürich
Asia		
	ASIA-EAST1	Taiwan
	ASIA-EAST2	Hong Kong
	ASIA-NORTHEAST1	Tokyo
	ASIA-NORTHEAST2	Osaka
	ASIA-NORTHEAST3	Seoul
	ASIA-SOUTH1	Mumbai
	ASIA-SOUTHEAST1	Singapore
Australia		
	AUSTRALIA-SOUTHEAST1	Sydney

## Multi-regions

Multi-Region Name	Multi-Region Description
ASIA	Data centers in Asia
EU	Data centers within <a href="#">member states</a> of the European Union <sup>1</sup>
US	Data centers in the United States

<sup>1</sup> Object data added to a bucket in the **EU** multi-region is not stored in the **EUROPE-WEST2** (London) or **EUROPE-WEST6** (Zurich) regions.

## Dual-regions

Dual-Region Name	Dual-Region Description
EUR4	EUROPE-NORTH1 and EUROPE-WEST4.
NAM4	US-CENTRAL1 and US-EAST1.

Sebagai contoh disini saya akan deploy pada server **singapura** dan nama bucket saya adalah **ahyartampan** maka dalam command line dituliskan sebagai berikut

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to model-night-271300.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
ahyar@cloudshell:~ (model-night-271300)$ gsutil mb -l asia-southeast1 gs://ahyartampan
Creating gs://ahyartampan/...
ahyar@cloudshell:~ (model-night-271300)$
```

**Silahkan deploy pada server apapun yang kalian inginkan, kemudian buat bucket dengan nama kalian masing2**

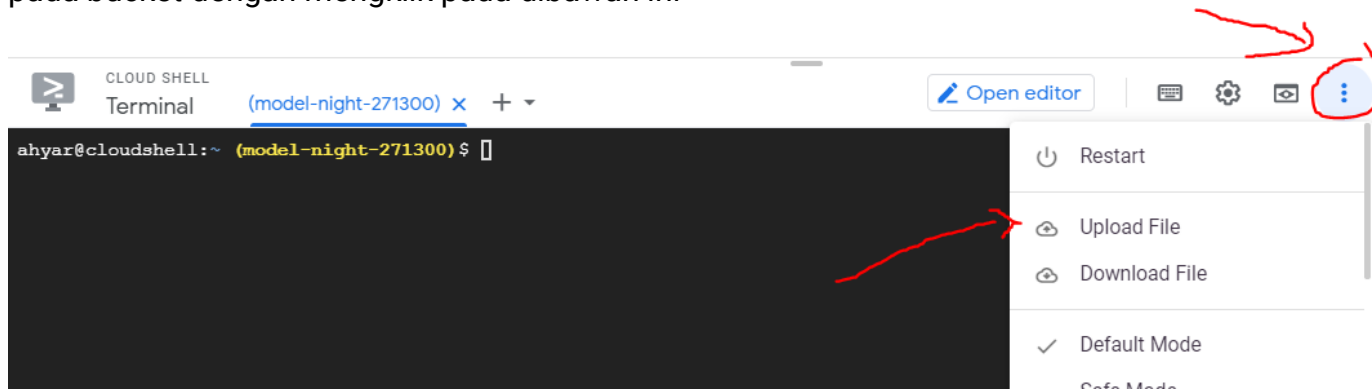
5. Untuk melihat daftar bucket yang ada pada project Anda dan memastikan bahwa bucket Anda sudah terbuat jalankan

```
gsutil ls
```

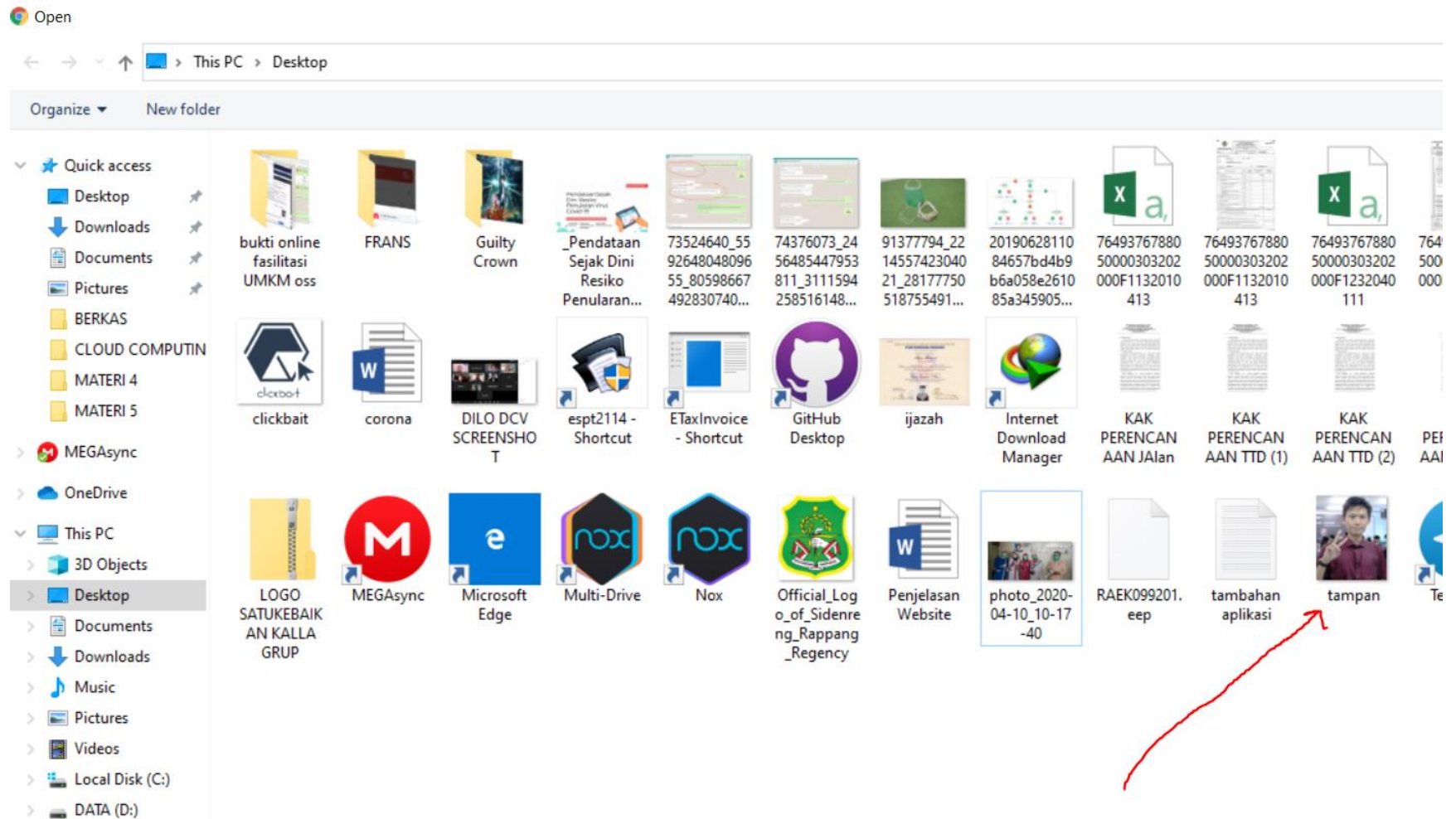
maka hasilnya

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil ls
gs://ahyartampan/
ahyar@cloudshell:~ (model-night-271300)$
```

6. Selanjutnya siapkan sendiri gambar yang akan kalian upload dalam bucket, kita akan mengupload sebuah gambar pada bucket dengan mengklik pada dibawah ini

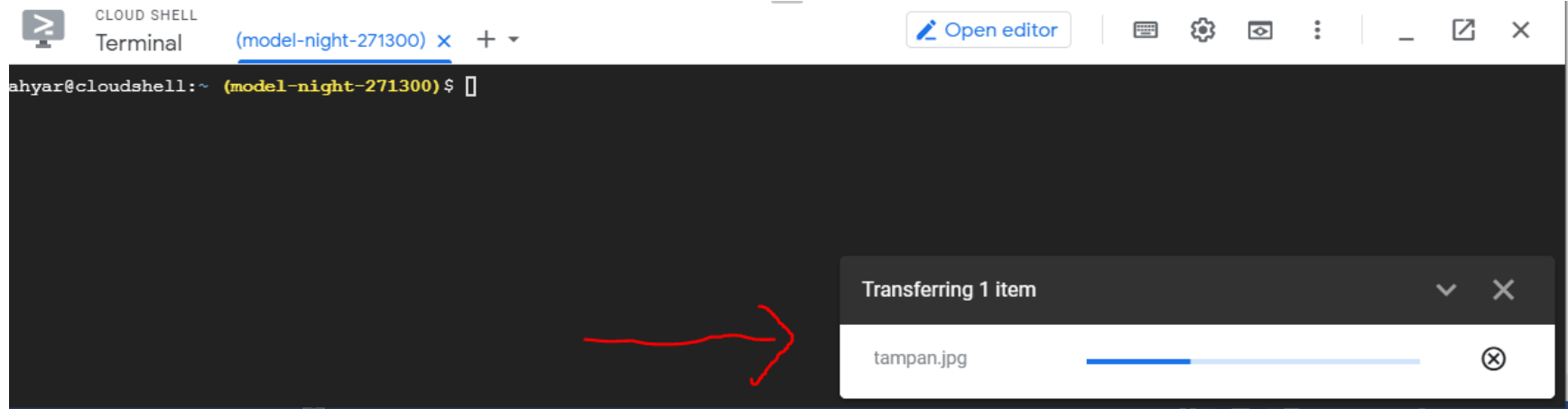


7. Selanjutnya silahkan **memilih file yang anda upload** dalam hal ini saya akan mengupload file gambar dengan nama **tampan.jpg**

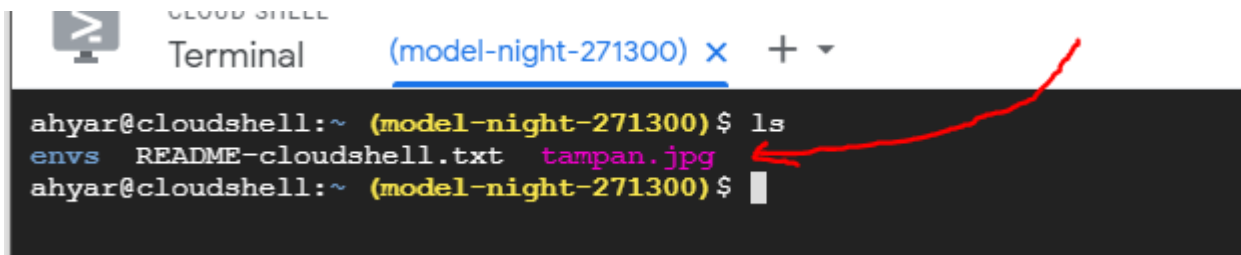




8. Tunggu proses upload hingga selesai.



9. Lakukan pengecekan apakah file sudah terupload dalam console dengan perintah **ls**.



Harap perhatikan nama gambar yang di upload lengkap dengan ekstensinya masing – masing. Karena akan digunakan untuk proses copy file.

10. Selanjutnya kita copy file yang kita upload kedalam bucket yang kita sudah buat dengan perintah .

```
gsutil cp file_name gs://bucket_name
```

sehingga hasilnya

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil cp tampan.jpg gs://ahyartampan
Copying file://tampan.jpg [Content-Type=image/jpeg]...
- [1 files][ 1.5 MiB/ 1.5 MiB]
Operation completed over 1 objects/1.5 MiB.
ahyar@cloudshell:~ (model-night-271300)$
```

11. Selanjutnya kita cek isi bucket kita apakah gambar sudah masuk dengan perintah.

```
gsutil ls gs://bucket_name
```

sehingga hasilnya

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil ls gs://ahyartampan/
gs://ahyartampan/tampan.jpg
ahyar@cloudshell:~ (model-night-271300)$
```

Terlihat bahwa kita telah berhasil mengupload sebuah file kedalam bucket.

12. Selanjutnya kita cek isi bucket kita apakah gambar sudah masuk dengan perintah.

```
gsutil acl ch -u AllUsers:R gs://bucket_name/file_name
```

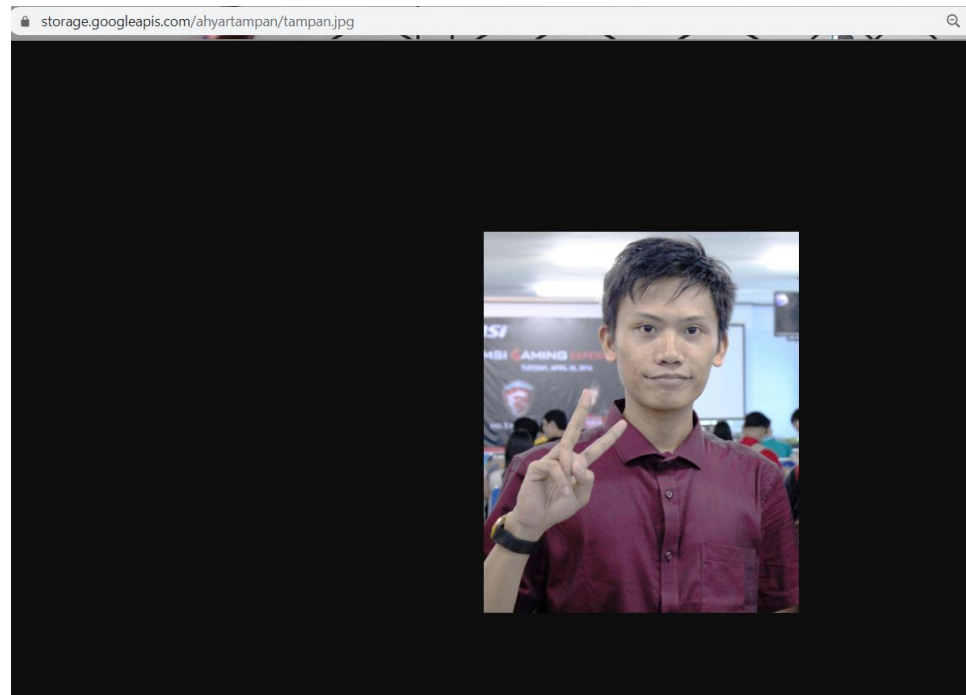
sehingga hasilnya

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil acl ch -u AllUsers:R gs://ahyartampan/tampan.jpg
Updated ACL on gs://ahyartampan/tampan.jpg
ahyar@cloudshell:~ (model-night-271300)$
```

13. Setelah menambahkan read permission, kita bisa mengakses object tadi melalui tautan seperti berikut:

```
https://storage.googleapis.com/bucket_name/file_name
```

sehingga hasilnya <https://storage.googleapis.com/ahyartampan/tampan.jpg> :



14. Selanjutnya kita akan mempelajari untuk **setting object versioning** dalam google cloud storage, Terdapat 2 cara untuk mengaktifkan object versioning, yaitu dengan menggunakan **perintah gsutil** atau dengan **REST API**. Kali ini kita akan belajar bagaimana **menggunakan object versioning melalui cloud shell atau terminal dengan perintah gsutil**. Untuk mengaktifkan versioning, jalankan perintah berikut :

```
gsutil versioning set on gs://bucket_name
```

sehingga hasilnya :

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil versioning set on gs://ahyartampan
Enabling versioning for gs://ahyartampan/...
ahyar@cloudshell:~ (model-night-271300)$
```

15. Untuk memastikan bahwa object versioning sudah aktif, jalankan perintah berikut:

```
gsutil versioning get gs://bucket_name
```

sehingga hasilnya :

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil versioning get gs://ahyartampan
gs://ahyartampan: Enabled
ahyar@cloudshell:~ (model-night-271300)$
```

16. Cobalah untuk mengunggah berkas dengan nama yang sama (sama seperti langkah no 10 diatas) pastikan nama file dan typenya sama, misalnya berkas teks dengan beberapa perubahan atau gambar yang diubah namun nama yang sama. Lalu jalankan perintah berikut untuk menampilkan semua objek, baik yang versi live maupun arsip, jalankan perintah berikut:

```
gsutil ls -a gs://bucket_name
```

sehingga hasilnya :

```
ahyar@cloudshell:~ (model-night-271300)$ gsutil ls -a gs://ahyartampan
gs://ahyartampan/tampan.jpg#1586668871005515
gs://ahyartampan/tampan.jpg#1586673411576143
```

Perhatikan diatas setiap nama gambar diatas memiliki penamaan yang sama, namun memiliki versi yang berbeda dengan generation number yang di ikuti dibelakangnya

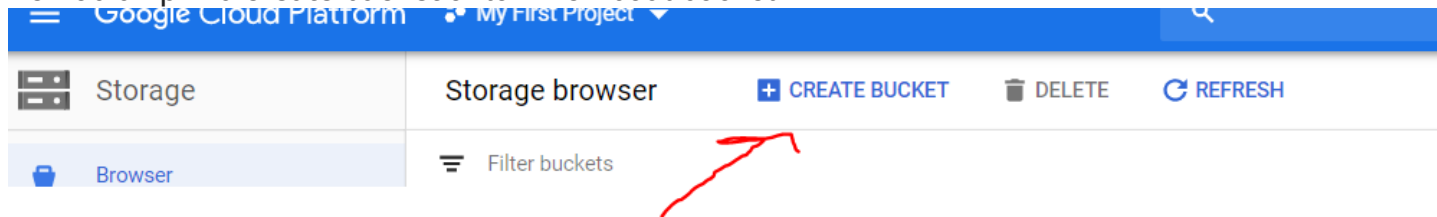
## **LATIHAN 2**

# **Google Cloud Storage (GCS) with Web Command and implement in HTML web NGINX**

Dalam latihan ini kita akan membuat bucket, mengupload gambar dalam bucket, dengan menggunakan web command dan generate akses linknya untuk dapat diakses melalui website yang nantinya dibuat dari nginx berdasarkan percobaan sebelumnya.

## Upload Image dalam Bucket dan generate link

1. Silahkan login kedalam akun google cloud computing kalian masing – masing.
2. Pada GCP Console, di **Navigation menu** (☰), click **storage**.
3. Kemudian pilih **create bucket** untuk membuat bucket.



4. Selanjutnya mengisi nama bucket dalam hal ini saya membuat bucket\_name = ahyartampan2, kemudian pencet continue

✓ **Name your bucket**

Pick a globally unique, permanent name. [Naming guidelines](#)

ahyartampan2

Tip: Don't include any sensitive information

CONTINUE

5. Kemudian menyimpan bucket di **tokyo**, dengan tipe lokasinya adalah **region** kemudian pencet continue

✓ **Name your bucket**

- **Choose where to store your data**  
This permanent choice defines the geographic placement of your data and affects cost, performance and availability. [Learn more](#)

**Location type**

☒ **Region**  
Lowest latency within a single region

☐ **Dual-region**  
High availability and low latency across 2 regions

☐ **Multi-region**  
Highest availability across largest area

**Location**

asia-northeast1 (Tokyo)

CONTINUE

### Catatan:

- > silahkan simpan bucket kalian ditempat berbeda
- > buat penamaan bucket menggunakan nama kalian masing – masing (ingat UNIK)

6. Kemudian memilih model standart sebagai penyimpanan, selanjutnya pencet **CREATE** pada bagian bawah

The screenshot shows the Google Cloud Platform interface for creating a new bucket. The top navigation bar includes the Google Cloud Platform logo, the project name 'My First Project', and a search bar. The left sidebar shows the 'Storage' section with options like 'Browser', 'Transfer', 'Transfer for on-premises', 'Transfer Appliance', and 'Settings'. The main content area is titled 'Create a Bucket' and contains a series of steps:

- Name your bucket** (checked)
- Choose where to store your data** (checked)
- Choose a default storage class for your data** (checked)

Under the third step, a description states: 'A storage class sets costs for storage, retrieval and operations. Pick a default storage class based on how long you plan to store your data and how often it will be accessed. [Learn more](#)'. Four storage classes are listed with radio buttons:

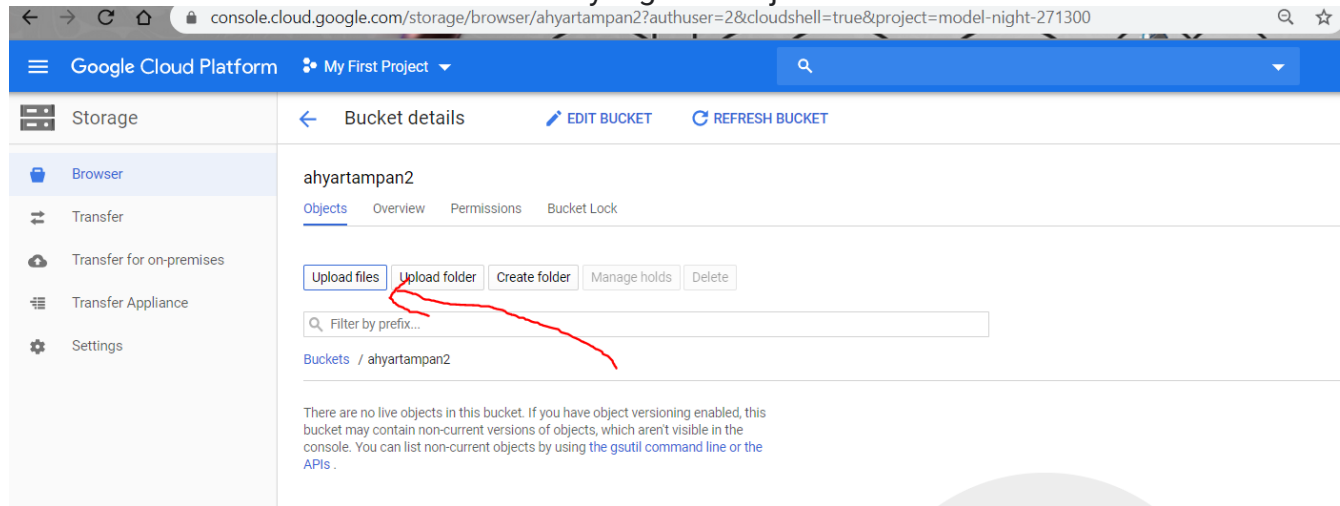
- Standard** (selected): Best for short-term storage and frequently accessed data
- Nearline**: Best for backups and data accessed less than once a month
- Coldline**: Best for disaster recovery and data accessed less than once a quarter
- Archive**: Best for long-term digital preservation of data accessed less than once a year

A 'CONTINUE' button is located below the storage class options. Below this, two more steps are listed:

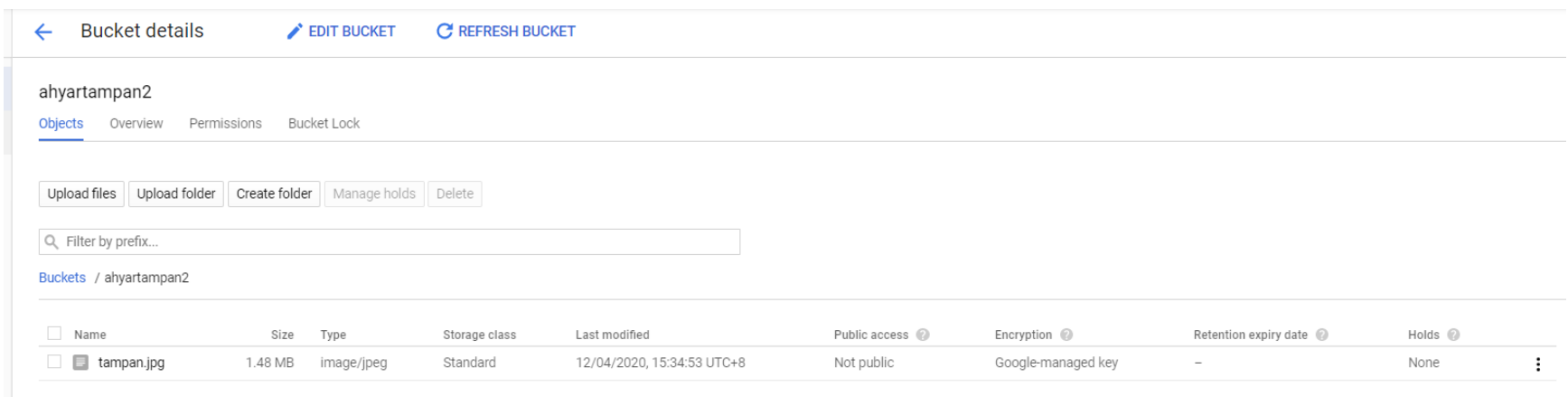
- Choose how to control access to objects**
- Advanced settings (optional)**

At the bottom, there are two buttons: 'CREATE' (highlighted with a red arrow) and 'CANCEL'.

7. Kemudian tunggu hingga proses selesai dan menampilkan **bucket details** kemudian pilih **upload files** untuk memasukkan file kedalam bucket yang baru saja kita buat.

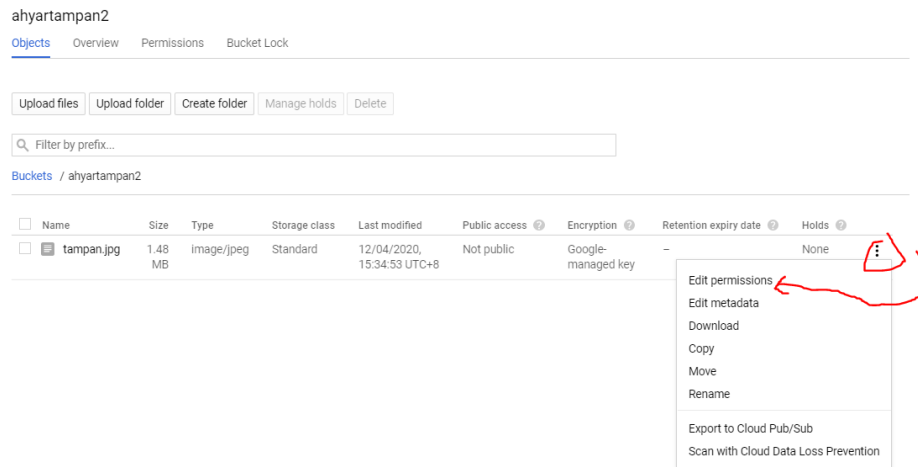


8. Kemudian pilih file yang digunakan, dalam hal ini saya menggunakan gambar yang sama pada latihan 1 sebelumnya yaitu **tampan.jpg** (silahkan gunakan file gambar kalian masing – masing). Tunggu proses upload selesai dan Setelah selesai mengupload maka tampilan **bucket details** akan menampilkan nama gambar dalam table dibawah untuk kita lakukan pengeditan lebih lanjut

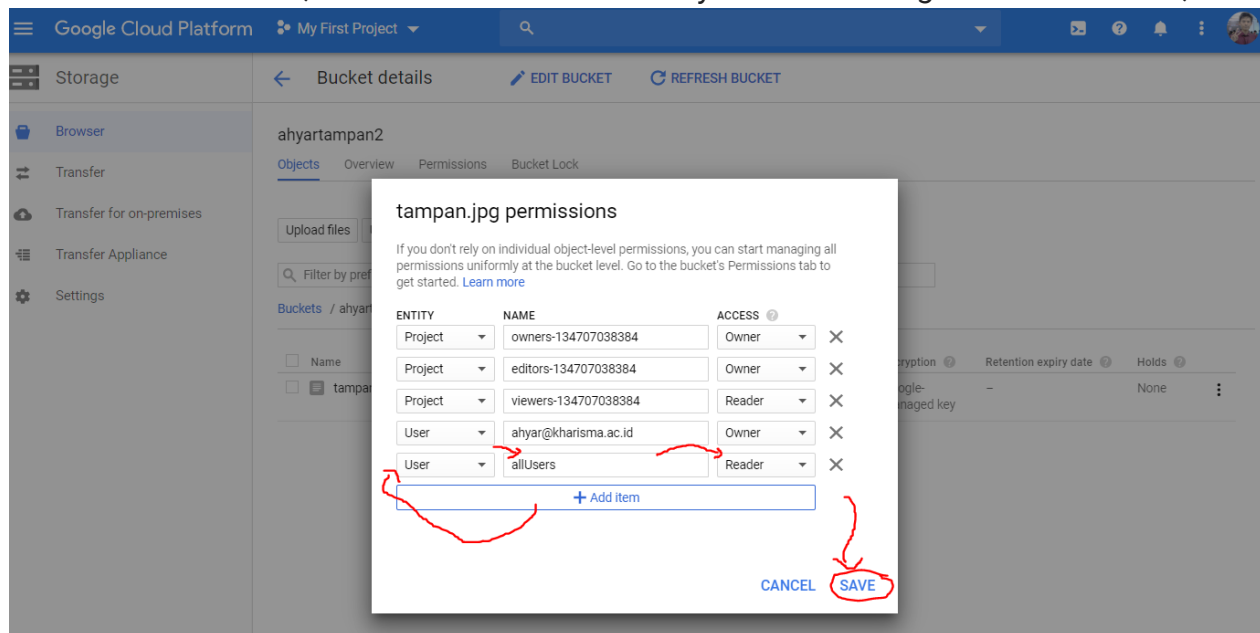





9. Untuk membuat berkas menjadi public, pilih menu opsi yang ada di bagian paling kanan list file, kemudian pilih **Edit permissions**.



10. Pilih **Add item**, kemudian tambahkan entity **user** baru dengan nama **allUsers**, berikan akses **Reader**, lalu klik **Save**.



11. Anda akan melihat pada bagian **public access** bahwa file akan memiliki **public link**. Klik tautan tersebut (tanda ). Anda akan diarahkan ke tab baru yang menampilkan berkas Anda dengan url yang lebih rapi dan mudah dibaca.









ahyartampan2

[Objects](#) [Overview](#) [Permissions](#) [Bucket Lock](#)

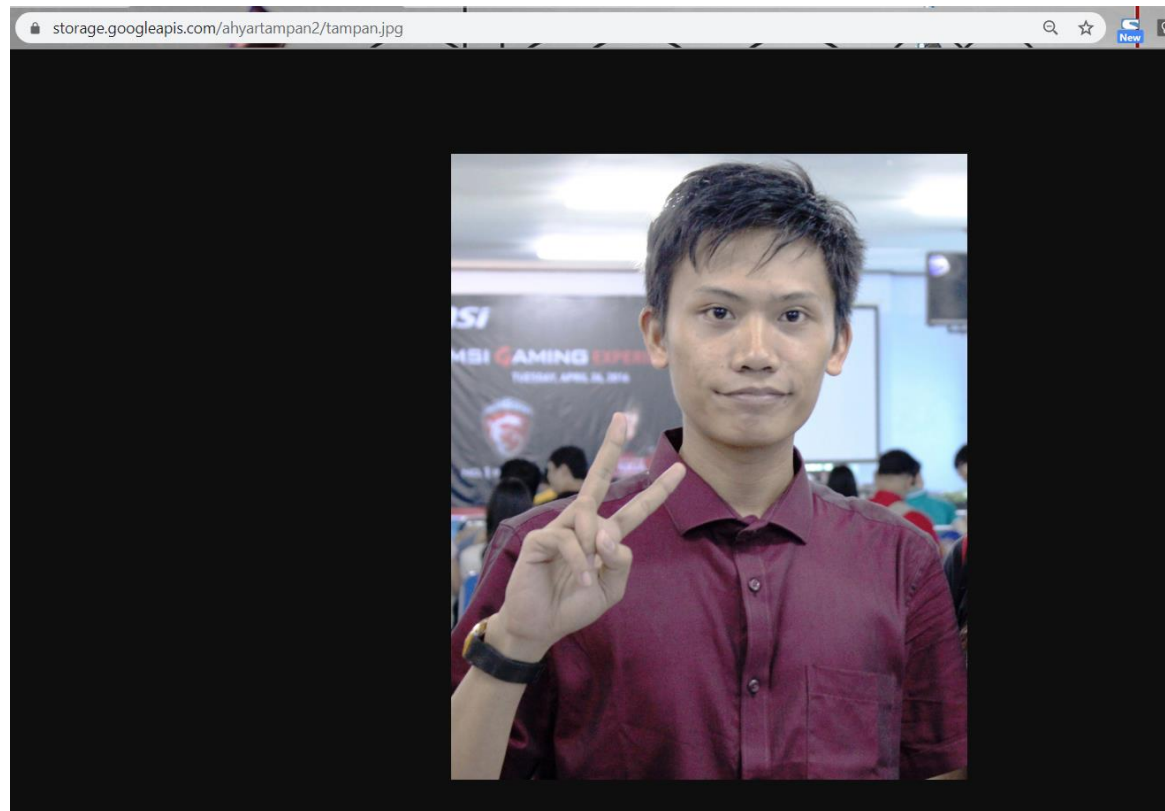
[Upload files](#) [Upload folder](#) [Create folder](#) [Manage holds](#) [Delete](#)

 Filter by prefix...

[Buckets](#) / ahyartampan2

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified	Public access 	Encryption 	Retention expiry date 	Holds 
<input type="checkbox"/>	 tampan.jpg	1.48 MB	image/jpeg	Standard	12/04/2020, 15:41:21 UTC+8	 Public to internet 	Google-managed key	–	None 

## HASILNYA

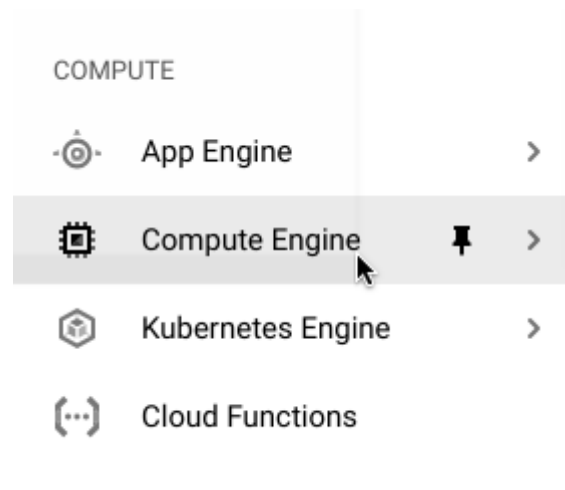


Seperti yang anda lihat diatas linknya adalah <https://storage.googleapis.com/ahyartampan2/tampan.jpg> hasil ini sama dengan perintah console yang kita lakukan (pada latihan 1)

Harap menyimpan link ini sebagai contoh <https://storage.googleapis.com/ahyartampan2/tampan.jpg> dimana kita akan lanjutkan untuk memunculkan gambar tersebut pada NGINX sesuai pada praktikum pada minggu sebelumnya.

**Buat Website dengan Link Cloud Storage (bucket) dengan terlebih dahulu menggunakan compute engine (SAMA SEPERTI MODUL 2 / PRAKTIKUM 2 SEBELUMNYA)**

12. Buka halaman Compute Engine pada navigation menu, pada bagian Compute pilih **Compute Engine**.



13. Untuk membuat instance baru, pilih **Create Instance**.

14. Selanjutnya Anda harus mengisi properti untuk VM Anda, dengan entri sebagai berikut:

- a. **Name:** Nama VM yang akan dibuat.
- b. **Region & Zone:** Zone yang akan menjalankan VM Anda.
- c. **Machine type:** Jenis mesin yang akan digunakan untuk menjalankan mesin virtual Anda. Bisa dikustomisasi sesuai dengan kebutuhan.
- d. **Container:** Centang jika Anda ingin men-*deploy container* untuk VM Anda.
- e. **Boot disk:** Pilihan sistem operasi yang akan berjalan pada VM.
- f. **Identity and API access:** Pengaturan *service account* dan hak akses ke API yang disediakan Google Cloud.
- g. **Firewall:** Pengaturan izin untuk *network traffic* dari internet ke VM dan sebaliknya. Pada latihan ini centang opsi **Allow HTTP traffic** dan **Allow HTTPS traffic** karena kita akan melakukan akses ke VM melalui protokol HTTP.

- h. **Management, security, disks, networking, sole tenancy:** Pengaturan tambahan yang meliputi keamanan, penyimpanan, dan jaringan pada VM.

Sebagai contoh :

**Name** <sup>?</sup>  
Name is permanent

**Labels** <sup>?</sup> (Optional)

**Region** <sup>?</sup>


**Zone** <sup>?</sup>  
Zone is permanent

**Machine configuration**

**Machine family**  
    
Machine types for common workloads, optimised for cost and flexibility


**Series**  
  
Powered by Intel Skylake CPU platform or one of its predecessors

**Machine type**

	vCPU	Memory
	1	3.75 GB

⌵ CPU platform and GPU

**Container** <sup>?</sup>  
☐ Deploy a container image to this VM instance. [Learn more](#)

**Boot disk** <sup>?</sup>  
 New 10 GB standard persistent disk  
Image  
Debian GNU/Linux 9 (stretch)

**Identity and API access** <sup>?</sup>  
**Service account** <sup>?</sup>

**Access scopes** <sup>?</sup>  
☒ Allow default access  
☐ Allow full access to all Cloud APIs  
☐ Set access for each API

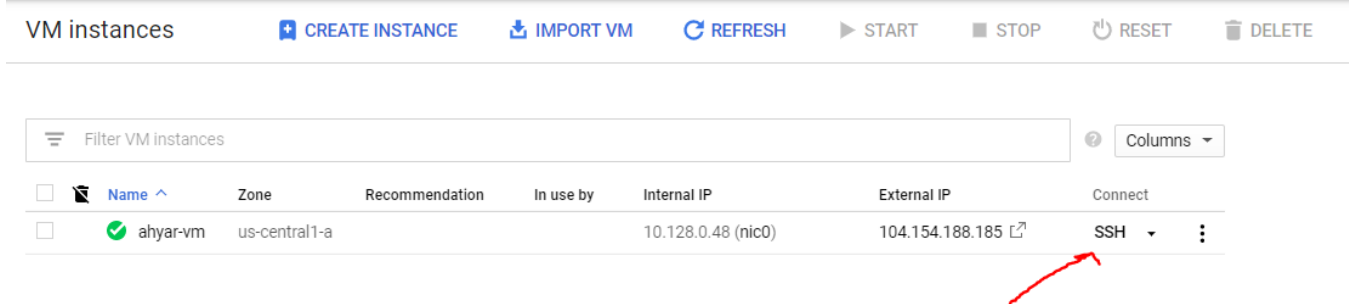
**Firewall** <sup>?</sup>  
Add tags and firewall rules to allow specific network traffic from the Internet.  
☒ Allow HTTP traffic  
☒ Allow HTTPS traffic

⌵ Management, security, disks, networking, sole tenancy

You will be billed for this instance. [Compute Engine pricing](#) <sup>?</sup>

Equivalent [REST](#) or [command line](#)

15. Selanjutnya kita akan masuk kedalam virtual machine **ahyar-vm** dengan mengklik ssh pada bagian connect



16. Tunggu hingga jendela control ssh ditampilkan kemudian kita install nginx dengan perintah

**sudo apt-get install nginx**

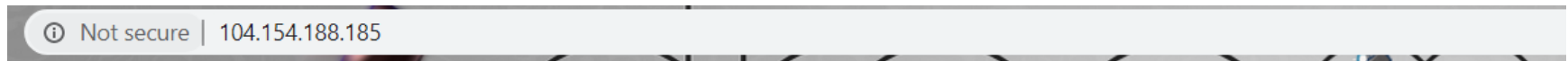
jangan lupa pilih **Y** untuk konfirmasi pemasangan nginx sehingga hasilnya

```
ahyar@ahyar-vm: ~ - Google Chrome
ssh.cloud.google.com/projects/model-night-271300/zones/us-central1-a/instances/ahyar-vm?authuser=2&hl=en_GB&projectNumber=13470703
Processing triggers for systemd (232-25+deb9u12) ...
Setting up libnginx-mod-http-auth-pam (1.10.3-1+deb9u3) ...
Setting up libnginx-mod-http-dav-ext (1.10.3-1+deb9u3) ...
Setting up libnginx-mod-mail (1.10.3-1+deb9u3) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up libnginx-mod-http-xslt-filter (1.10.3-1+deb9u3) ...
Setting up libnginx-mod-http-upstream-fair (1.10.3-1+deb9u3) ...
Setting up libxdmcp6:amd64 (1:1.1.2-3) ...
Setting up xml-core (0.17) ...
Setting up libnginx-mod-http-geoip (1.10.3-1+deb9u3) ...
Setting up libx11-data (2:1.6.4-3+deb9u1) ...
Setting up libxau6:amd64 (1:1.0.8-1) ...
Setting up libwebp6:amd64 (0.5.2-1) ...
Setting up fontconfig-config (2.11.0-6.7) ...
Setting up libnginx-mod-stream (1.10.3-1+deb9u3) ...
Setting up libnginx-mod-http-echo (1.10.3-1+deb9u3) ...
Setting up libxcb1:amd64 (1.12-1) ...
Setting up libfontconfig1:amd64 (2.11.0-6.7+b1) ...
Setting up libx11-6:amd64 (2:1.6.4-3+deb9u1) ...
Setting up libxpm4:amd64 (1:3.5.12-1) ...
Setting up libgd3:amd64 (2.2.4-2+deb9u5) ...
Setting up libnginx-mod-http-image-filter (1.10.3-1+deb9u3) ...
Setting up nginx-full (1.10.3-1+deb9u3) ...
Setting up nginx (1.10.3-1+deb9u3) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
Processing triggers for sgml-base (1.29) ...
ahyar@ahyar-vm:~$
```

17. Selanjutnya kembali ke halaman vm instances pastikan copy **External IP** kedalam browser untuk melihat apakah nginx telah berhasil berjalan atau tidak

VM instances							
<div>CREATE INSTANCEIMPORT VMREFRESHSTARTSTOPRESETDELET</div>							
Filter VM instances							Columns
	Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> ahyar-vm	us-central1-a			10.128.0.48 (nic0)	104.154.188.185	SSH

Ketika External ip dicopy dan membuka browser maka akan ditampilkan sebagai berikut :



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

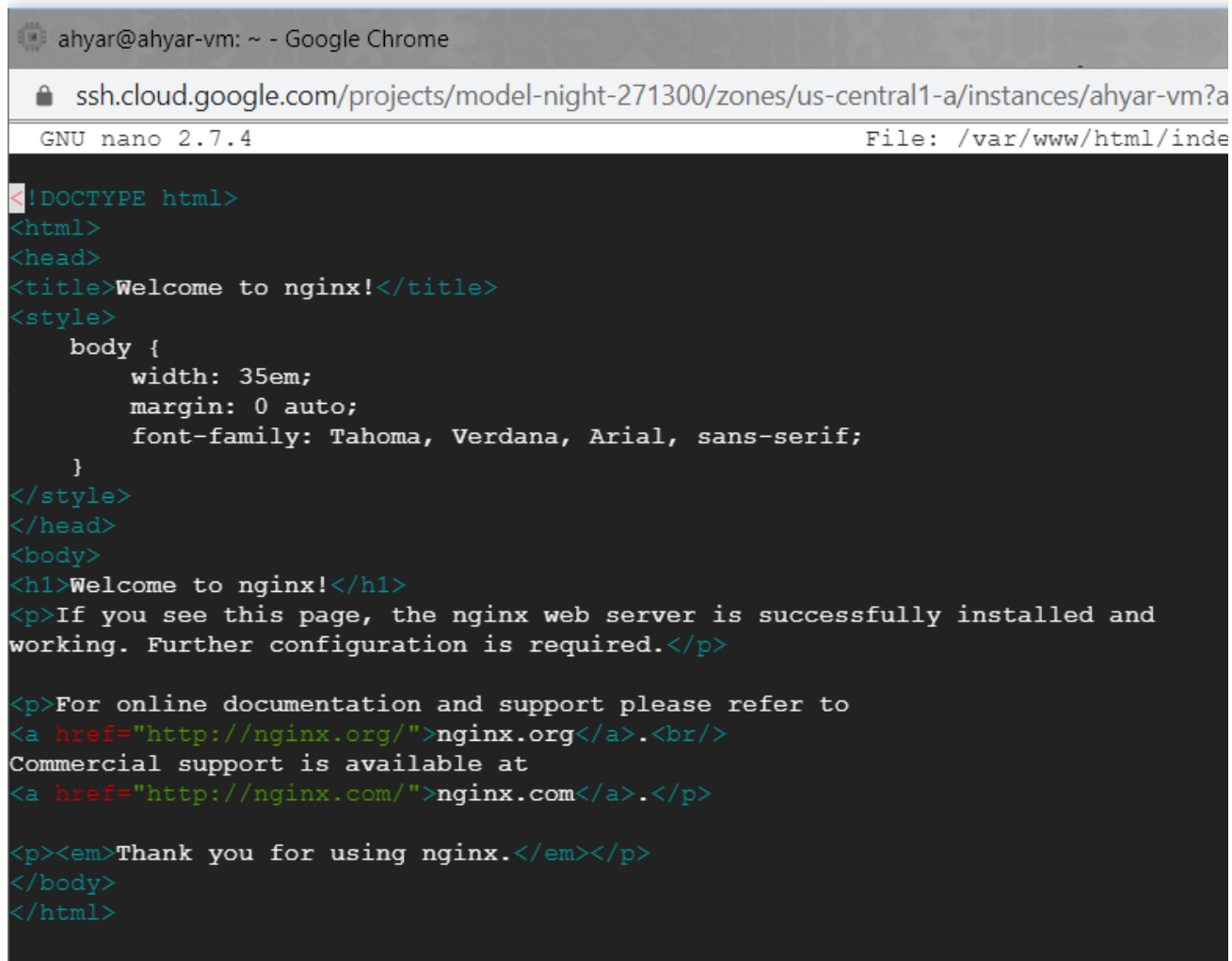
For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

18. Kembali ke halaman tab SSH yang terbuka selanjutnya mengedit isi html NGINX dengan perintah nano

```
sudo nano /var/www/html/index.nginx-debian.html
```

maka file html nginx akan terbuka sebagai berikut



The screenshot shows a Google Chrome browser window with the address bar displaying the SSH connection URL: `ssh.cloud.google.com/projects/model-night-271300/zones/us-central1-a/instances/ahyar-vm?a`. The browser window is titled "ahyar@ahyar-vm: ~ - Google Chrome". Below the browser window, the GNU nano 2.7.4 text editor is open, editing the file `/var/www/html/index.nginx-debian.html`. The editor shows the following HTML content:

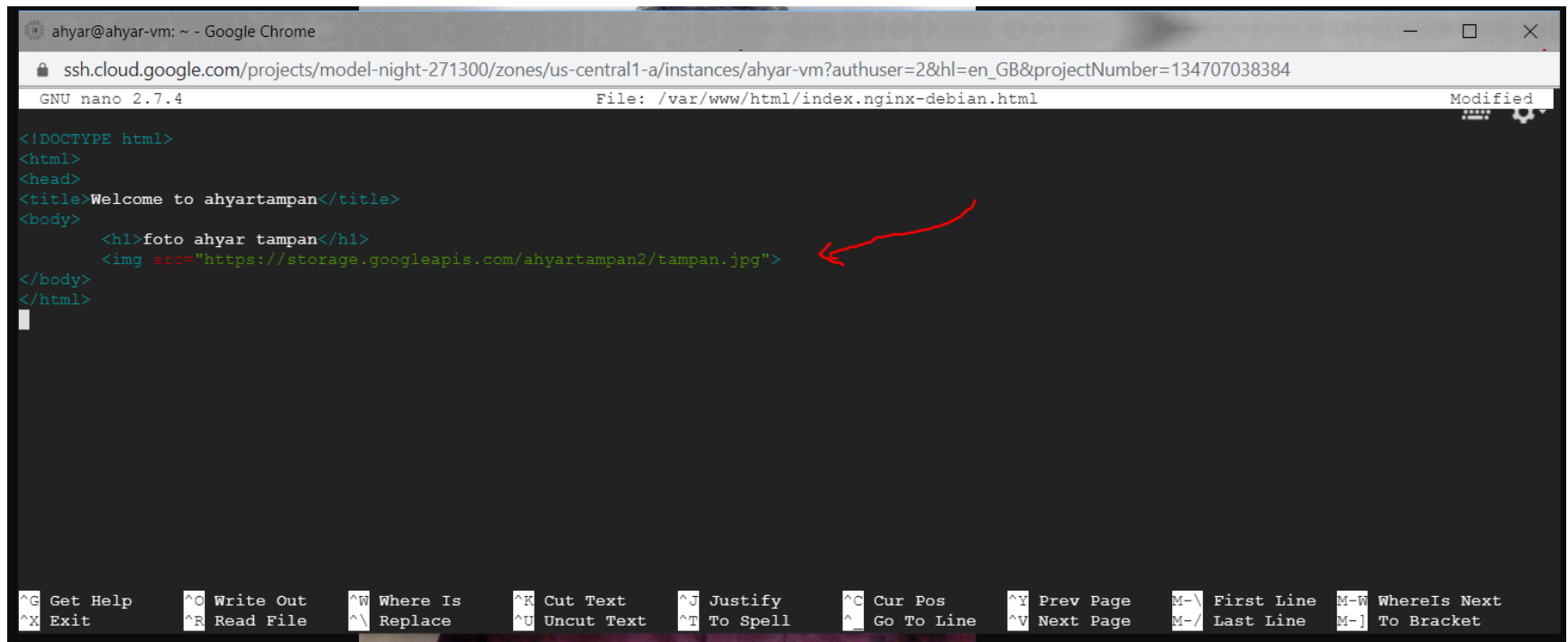
```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```



19. Selanjutnya edit file tersebut dan masukkan link alamat gambar yang sudah kita buat dalam bucket <https://storage.googleapis.com/ahyartampan2/tampan.jpg> kedalam sintaks `<img src="">` secara lengkap seperti berikut



```
ahyar@ahyar-vm: ~ - Google Chrome
ssh.cloud.google.com/projects/model-night-271300/zones/us-central1-a/instances/ahyar-vm?authuser=2&hl=en_GB&projectNumber=134707038384
GNU nano 2.7.4 File: /var/www/html/index.nginx-debian.html Modified
<!DOCTYPE html>
<html>
<head>
<title>Welcome to ahyartampan</title>
<body>
  <h1>foto ahyar tampan</h1>
  
</body>
</html>
```

Kemudian simpan dengan **ctrl+O**, dan **ctrl + X** untuk keluar dari editor

20. Setelah selesai buka ip kembali dan refresh untuk melihat hasilnya

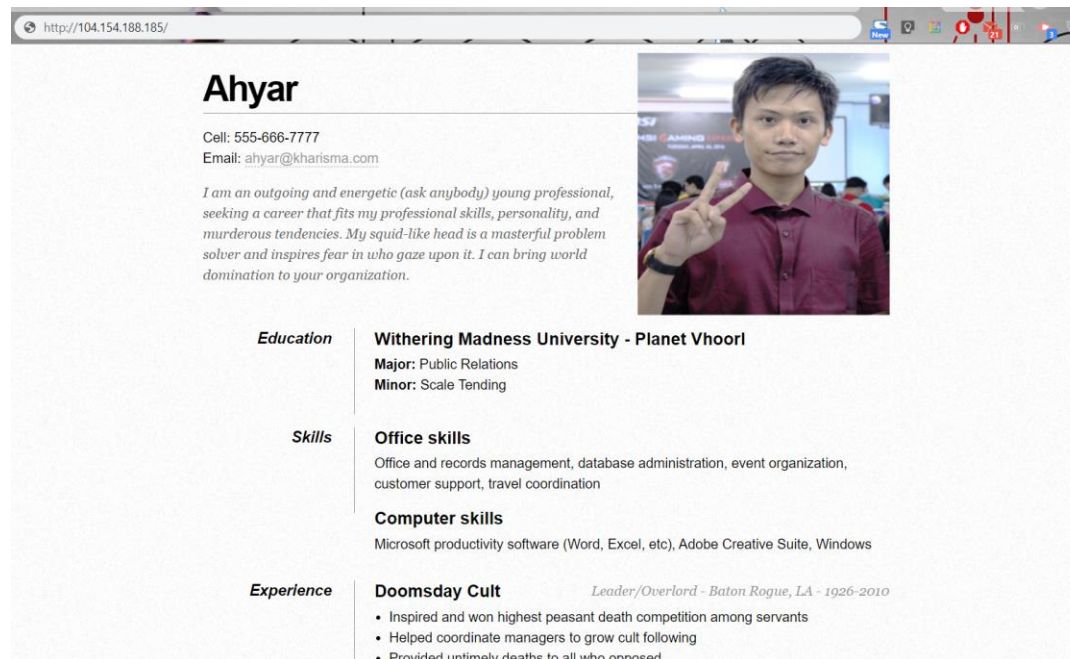


## HADIAH

Silahkan deploy website profile kalian masing – masing menggunakan bucket (Google Cloud Storage / GCS) dan NGINX pada layanan Google Compute Engine dengan menggunakan bahan one page resume yang telah di download dari link github berikut ini :

<https://github.com/ahyarmuawwal/OnePageResume>

## CONTOH HASILNYA



## Bukti upload :

1. foto hasil website seperti diatas lengkap dengan biodata kalian yang di isi lengkap, dan foto kalian masing - masing
2. foto hasil bucket details yang menampilkan bahwa gambar anda telah berhasil di upload.