PROFESSIONAL DATA ENGINEER *- Google Cloud Platform*

TITLE: "Introduction to Google Cloud - Sign Temproary URL"

Author:

- *Name: "Vignesh Sekar S"*
- *Designation: "Multi Cloud Architect"*
- *Tags: [Google Cloud, DataEngineer, Python, PySpark, SQL, BigData]*

gsutil tool documentation link

- https://cloud.google.com/storage/docs/gsutil (https://cloud.google.com/storage/docs/gsutil)

SignUrl - Create a signed URL

- The signurl command will generate a signed URL that embeds authentication data so the URL can be used by someone who does not have a Google account.
- A signed URL is a URL that provides limited permission and time to make a request.
- Signed URLs contain authentication information in their query string, allowing users without credentials to perform specific actions on a resource
- Temporary access
- you can give access to user who doesn't have Google Account. URL expired after time period defined.
- Max period for which URL is valid is 7 days.
- **gsutil signurl -d 10m -u gs:///**

When should you use a signed URL?

- In some scenarios, you might not want to require your users to have a Google account in order to access Cloud Storage, but you still want to control access using your application-specific logic.
- The typical way to address this use case is to provide a signed URL to a user, which gives the user read, write, or delete access to that resource for a limited time.

    - You specify an expiration time when you create the signed URL.
    - Anyone who knows the URL can access the resource until the expiration time for the URL is reached or the key used to sign the URL is rotated.

- The most common uses for signed URLs are uploads and downloads, because in such requests, object data moves between requesters and Cloud Storage.
- In most other cases, such as copying objects, composing objects, deleting objects, or editing metadata, creating a signed URL and giving it to someone to use is an unnecessary extra step. Instead, you should consider a design in which the entity responsible for creating the signed URL directly makes the desired request to Cloud Storage.

Options for generating a signed URL

- Cloud Storage supports several methods for generating a signed URL:
    - V4 signing with service account authentication.
    - Signing with HMAC authentication.
    - V2 signing with service account authentication.

Commands:

- Create new bucket for demo. Use existing compute engine service account.
- upload Service Account json key to gcp cloud platform.
    - Sudo pip3 install pyopenssl – pyopenssl, gsutil will internally use this for signed url.
    - sudo apt-get install libssl-dev
    - Sudo pip3 install –upgrade pip
    - Sudo pip3 install pyopenssl

- Provide storage object viewer role to that SA.
- gsutil signurl -d 60s . /home/cloudaianalytics/
- gsutil signurl -d 10m -r us-central1 zeta-matrix-377816-aa0a6fbe9836.json gs://zeta-matrix-377816-data100/hamlet.txt
- gsutil signurl -d 10s zinc-forge-380121-444ecd3a7c3c.json gs://gsutildemo-pde-0002/titanic.csv
- gsutil signurl -d 1m zinc-forge-380121-444ecd3a7c3c.json gs://gsutildemo-pde-0002/desktopwallpaper1.jpeg

---