

Kishan Bhakta

COMPSCI 4750

LAB 3

## 8.1 Concepts

There are 4 ways to manipulate buckets and objects and they are:

1. Via local machine
2. Via ssh from a local machine
3. Via direct ssh from the console
4. Directly into the bucket from a web browser.

To gain access privileges, one needs authentication via *gcloud auth application-default login* and *gcloud auth login*.

The screenshot displays the Google Cloud IAM & Admin console. The left sidebar shows the navigation menu with 'Service Accounts' highlighted. The main content area is titled 'Compute Engine default service account' and includes tabs for 'DETAILS', 'PERMISSIONS', 'KEYS', 'METRICS', and 'LOGS'. The 'DETAILS' tab is active, showing the 'Service account details' section with fields for 'Name' (Compute Engine default service account) and 'Description', both with 'SAVE' buttons. Below these, the 'Email' is listed as 483264854334-compute@developer.gserviceaccount.com and the 'Unique ID' is 110999075900040802093. The 'Service account status' section indicates the account is 'Account currently active' with a green checkmark and a 'DISABLE SERVICE ACCOUNT' button. At the bottom, there is a 'SHOW ADVANCED SETTINGS' link.

The service account had a default access to the VM instance.

engine

learning-demo

EDIT

RESET

CREATE MACHINE IMAGE

DETAILS

OBSERVABILITY

OS INFO

NEW

SCREENSHOT

Local disks

None

Additional disks

None

Security and access

Shielded VM

Secure Boot

Off

vTPM

On

Integrity Monitoring

On

SSH Keys

SSH keys

None

Block project-wide SSH keys

Off

API and identity management

Service account

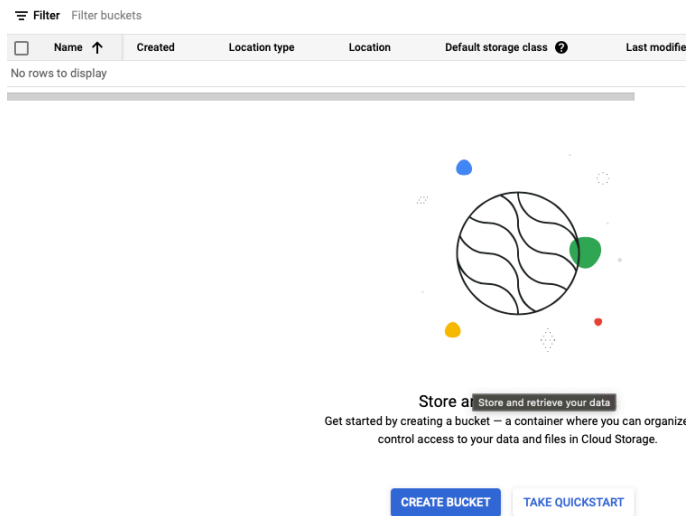
483264854334-compute@developer.gserviceaccount.com

Cloud API access scopes

Allow default access

Hence, we don't need to download the public key of the storage to learning-demo.

## 8.2 Storing data in Cloud Storage



### ✓ Name your bucket

Name: kishans-first-bucket

### ✓ Choose where to store your data

Location: us (multiple regions in United States)

Location type: Multi-region

### ✓ Choose a default storage class for your data

Default storage class: Standard

### ✓ Choose how to control access to objects

Public access prevention: Off

Access control: Uniform

### • Choose how to protect object data

Your data is always protected with Cloud Storage but you can also choose from these additional data protection options to prevent data loss. Note that object versioning and retention policies cannot be used together.

#### Protection tools

☒ None

☐ Object versioning (best for data recovery)

For restoring deleted or overwritten objects. To minimize the cost of storing versions, we recommend limiting the number of noncurrent versions per object and scheduling them to expire after a number of days. [Learn more](#)

☐ Retention policy (best for compliance)

For preventing the deletion or modification of the bucket's objects for a specified minimum duration of time after being uploaded. [Learn more](#)

#### ✓ DATA ENCRYPTION

CREATE

CANCEL

```

kb@kb-virtualbox:~$ gsutil ls
gs://kishans-first-bucket/
kb@kb-virtualbox:~$ gcloud compute instances list
NAME                                ZONE                MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP
learning-demo                       us-central1-a      e2-medium    10.128.0.2
kb@kb-virtualbox:~$
kb@kb-virtualbox:~$ echo "This is my first file!" > my_first_file.txt
kb@kb-virtualbox:~$ gsutil ls
gs://kishans-first-bucket/
kb@kb-virtualbox:~$ gsutil cp my_first_file.txt gs://kishans-first-bucket/
Copying file:///my_first_file.txt [Content-Type=text/plain]...
- [1 files][ 23.0 B/ 23.0 B]
Operation completed over 1 objects/23.0 B.
kb@kb-virtualbox:~$

```

## kishans-first-bucket

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

**OBJECTS** CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE

Buckets > kishans-first-bucket

[UPLOAD FILES](#)
[UPLOAD FOLDER](#)
[CREATE FOLDER](#)
[MANAGE HOLDS](#)
[DOWNLOAD](#)
[DELETE](#)

Filter by name prefix only ▼	Filter	Filter objects and folders	Show deleted data	
<input type="checkbox"/> Name	Size	Type	Created ?	Storage
<input type="checkbox"/> my_first_file.txt	23 B	text/plain	Dec 7, 20...	Standard

```

kb@kb-virtualbox:~$ gsutil ls
gs://kishans-first-bucket/
kb@kb-virtualbox:~$ npm install @google-cloud/storage
[ ] ..... / extract:stream-events: sill extract mime-types@2.1

```

```

kb@kb-virtualbox:~$ cat list_script.js
const {Storage}=require('@google-cloud/storage');
const storage = new Storage({projectId: 'qualified-smile-334417'});
const bucket = storage.bucket('kishans-first-bucket');
bucket.GetFiles()
    .then(files=>{
        files[0].forEach(
            file => console.log('found a file: ', file.name));
    });

kb@kb-virtualbox:~$ node list_script.js
found a file: my_first_file.txt
kb@kb-virtualbox:~$ echo "My second file." > my_second_file.txt
kb@kb-virtualbox:~$ gsutil cp my_second_file.txt gs://kishans-first-bucket/
Copying file://my_second_file.txt [Content-Type=text/plain]...
/ [1 files][ 16.0 B/ 16.0 B]
Operation completed over 1 objects/16.0 B.
kb@kb-virtualbox:~$ node list_script.js
found a file: my_first_file.txt
found a file: my_second_file.txt

```

### kishans-first-bucket

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

[OBJECTS](#) [CONFIGURATION](#) [PERMISSIONS](#) [PROTECTION](#) [LIFECYCLE](#)

Buckets > kishans-first-bucket

[UPLOAD FILES](#) [UPLOAD FOLDER](#) [CREATE FOLDER](#) [MANAGE HOLDS](#) [DOWNLOAD](#) [DELETE](#)

Filter by name prefix only Filter Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created ?	Storage class	Last modified	Public access
<input type="checkbox"/>	my_first_file.txt	23 B	text/plain	Dec 7, 20...	Standard	Dec 7, 202...	Not public
<input type="checkbox"/>	my_second_file.txt	16 B	text/plain	Dec 9, 20...	Standard	Dec 9, 202...	Not public

```

kb@kb-virtualbox:~$ echo "My third file." > my_third_file.txt
kb@kb-virtualbox:~$ vim upload_script.js
kb@kb-virtualbox:~$ cat upload_script.js
const {Storage}=require('@google-cloud/storage');
const storage = new Storage({projectId: 'qualified-smile-334417'});
const bucket = storage.bucket('kishans-first-bucket');
bucket.upload('my_third_file.txt', (err, file) => {
    if(err){
        console.log('There was an error: ', err);}
    else{
        console.log('Uploaded file: ', file.name);}
});


kb@kb-virtualbox:~$ node upload_script.js
Uploaded file: my_third_file.txt
kb@kb-virtualbox:~$ node list_script.js
found a file: my_first_file.txt
found a file: my_second_file.txt
found a file: my_third_file.txt
kb@kb-virtualbox:~$

```


## kishans-first-bucket





Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

**OBJECTS** CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE

Buckets > kishans-first-bucket 

**UPLOAD FILES** **UPLOAD FOLDER** **CREATE FOLDER** **MANAGE HOLDS** **DOWNLOAD** **DELETE**

Filter by name prefix only ▼  **Filter** Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created 	Storage class	Last modified
<input type="checkbox"/>	 my_first_file.txt	23 B	text/plain	Dec 7, 20...	Standard	Dec 7, 202...
<input type="checkbox"/>	 my_second_fle.txt	16 B	text/plain	Dec 9, 20...	Standard	Dec 9, 202...
<input type="checkbox"/>	 my_third_file.txt	15 B	text/plain	Dec 9, 20...	Standard	Dec 9, 202...

As we can see, there are two methods to upload a file/object to the storage via script to gsutil.

Both (above used) methods to upload the files/objects work and objects are being listed in the bucket once the listing script is run.

### 8.3 Choosing the right storage class

There was nothing to do over here except for reading what GCP had as storage options for the customers.

Above we saw it from a local machine. Let's try ssh from the console method.

```

kbzz249@learning-demo: ~
ssh.cloud.google.com/projects/qualified-smile-334417/zones/us-central1-a/instances/learning-demo?authuser=0&hl=en_US&...
kbzz249@learning-demo:~$ gcloud auth application-default login

You are running on a Google Compute Engine virtual machine.
The service credentials associated with this virtual machine
will automatically be used by Application Default
Credentials, so it is not necessary to use this command.

If you decide to proceed anyway, your user credentials may be visible
to others with access to this virtual machine. Are you sure you want
to authenticate with your personal account?

Do you want to continue (Y/n)? y

Go to the following link in your browser:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=764086051850-6qr4p6gpi6hn506pt8ejuq83di341hur.apps.g...
oglesusercontent.com&redirect_uri=urn%3Aietf%3Aawg%3Aoauth%3A2.0%3Aaob&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuser...
info.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&st...
ate=6ORNPaiZGtRmkevy62Vof2kETyCnqS&prompt=consent&access_type=offline&code_challenge=h-Td25I3F5eeD2yiU9V39DRHn_uYeyL1npLAnWfni8F...
&code_challenge_method=S256

Enter verification code: 4/1AX4XfWgXmAzh9Keqci4BaiYuo_LHbaHS7V61UUidOdfLaVoADM8y7oA-yGs

Credentials saved to file: [/home/kbzz249/.config/gcloud/application_default_credentials.json]

These credentials will be used by any library that requests Application Default Credentials (ADC).

Quota project "qualified-smile-334417" was added to ADC which can be used by Google client libraries for billing and quota. Note
that some services may still bill the project owning the resource.
kbzz249@learning-demo:~$ gsutil ls
gs://kishans-first-bucket/
kbzz249@learning-demo:~$

```

If you get denied access, change the scopes in OAuth consent screen.

My First Project

scope

Edit app registration

Application privacy policy link

Not provided

Application terms of service link

Not provided

Authorized domains

Not provided

Contact email addresses

kbzz249@gmail.com

Scopes

API ↑	Scope	User-facing description
BigQuery	.../auth/devstorage	Manage your data in Cloud Storage
API	.read_write	Google Account

kbzz249@learning-demo: ~

ssh.cloud.google.com/projects/qualified-smile-334417/zones/us-central1-a/instances/learning-demo?authuser=0&hl=en\_US&...

kbzz249@learning-demo:~\$ echo "This is my first-ssh-file" > my\_first\_ssh\_file.txt
kbzz249@learning-demo:~\$ gsutil cp my\_first\_ssh\_file.txt gs://kishans-first-bucket/
Copying file://my\_first\_ssh\_file.txt [Content-Type=text/plain]...
AccessDeniedException: 403 Provided scope(s) are not authorized
kbzz249@learning-demo:~\$

My First Project

scope

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 organization. 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

```

kbzz249@learning-demo:~$ gsutil cp my_first_ssh_file.txt gs://kishans-first-bucket/
Copying file://my_first_ssh_file.txt [Content-Type=text/plain]...
/ [1 files][ 26.0 B/ 26.0 B]
Operation completed over 1 objects/26.0 B.
kbzz249@learning-demo:~$ vim listmyobjects_script.js
kbzz249@learning-demo:~$ cat listmyobjects_script.js
const {Storage} = require('@google-cloud/storage');
const storage = new Storage({projectId: 'qualified-smile-334417'});
const bucket = storage.bucket('kishans-first-bucket');
bucket.GetFiles()
    .then(files => {
        files[0].forEach(
            file => console.log('Found a file:', file.name));
    });
kbzz249@learning-demo:~$ npm install @google-cloud/storage
npm WARN npm npm does not support Node.js v10.24.0
npm WARN npm You should probably upgrade to a newer version of node as we
npm WARN npm can't make any promises that npm will work with this version.
npm WARN npm Supported releases of Node.js are the latest release of 4, 6, 7, 8, 9.
npm WARN npm You can find the latest version at https://nodejs.org/
[.....] / fetchMetadata: sill resolveWithNewModule wrappy@1.0.2 checking ins

```

ssh.cloud.google.com/projects/qualified-smile-334417/zones/us-central1-a/instances/lea

```

kbzz249@learning-demo:~$ echo "This is my second ssh file" > my_second_ssh_file.txt
kbzz249@learning-demo:~$ vim uploadsshfileobject.js
kbzz249@learning-demo:~$ cat uploadsshfileobject.js
const {Storage} = require('@google-cloud/storage');
const storage = new Storage({projectId: 'qualified-smile-334417'});
const bucket = storage.bucket('kishans-first-bucket');

bucket.upload('my_second_ssh_file.txt', (err, file) => {
    if(err){
        console.log('There was an error: ', err);
    }
    else{
        console.log('Uploaded file: ', file.name);
    }
});
kbzz249@learning-demo:~$ node uploadsshfileobject.js
Uploaded file: my_second_ssh_file.txt
kbzz249@learning-demo:~$ node listmyobjects_script.js
Found a file: my_first_ssh_file.txt
Found a file: my_second_ssh_file.txt
kbzz249@learning-demo:~$ █


```




## kishans-first-bucket




<b>Location</b>	<b>Storage class</b>	<b>Public access</b>	<b>Protection</b>
us (multiple regions in United States)	Standard	Not public	None

[OBJECTS](#)[CONFIGURATION](#)[PERMISSIONS](#)[PROTECTION](#)[LIFECYCLE](#)

Buckets > kishans-first-bucket 

[UPLOAD FILES](#)[UPLOAD FOLDER](#)[CREATE FOLDER](#)[MANAGE HOLDS](#)[DOWNLOAD](#)[DELETE](#)

Filter by name prefix only ▼  **Filter** Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created 	Storage class	Last modified	F
<input type="checkbox"/>	 my_first_ssh_file.txt	26 B	text/plain	Dec 12, 2...	Standard	Dec 12, 20...	N
<input type="checkbox"/>	 my_second_ssh_file.txt	27 B	text/plain	Dec 12, 2...	Standard	Dec 12, 20...	N

So, buckets and objects can be accessed in many ways if you have the authentication and the right tools.