

# Google Cloud Deployment of the Slim Viewer

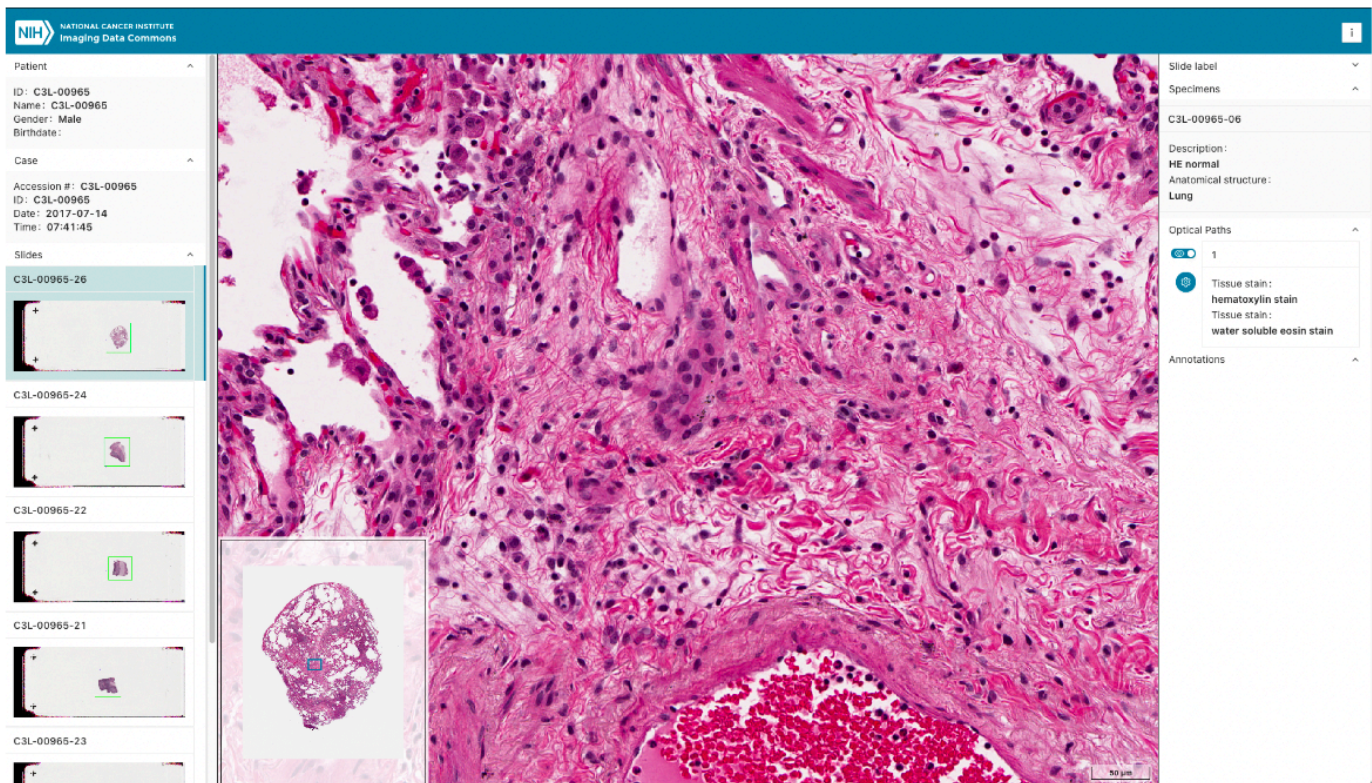
Short link to this document: <https://tinyurl.com/idc-slim-gcp>

Authors: Daniela Schacherer, Igor Octaviano, Andrey Fedorov ([NCI Imaging Data Commons](#)).

Revision history:

- Version for public release: August 2023

Slim is an interactive web application for visualization and annotation of digital whole slide microscopy images and derived image analysis results represented in standard DICOM format. The source code is publicly available on Github at <https://github.com/ImagingDataCommons/slim>.



This tutorial will show you how to deploy the Slim viewer as a hosted web application using [Firebase](#) and how it can be configured to access DICOM data from [Google Healthcare DICOM stores](#) using your user credentials. Some of the examples of the use cases that can be supported by such configuration are:

- Visualize results of analysis of DICOM images available in IDC using the same viewer as the one used in IDC;
- Share access to visualization of restricted access data with your collaborators;
- ...

This tutorial contains a lot of steps, and may look intimidating. BUT you only need to do this once to have your own instance of the Slim viewer that you can configure to your liking, that you can use with any of the Google Healthcare DICOM stores, which you can share with your friends and family - so it might be worth your time!

If you're just looking for an explanation on how to **set-up and populate a Google Healthcare DICOM store**, you can simply jump to the section [Creating and populating a Google Healthcare DICOM store](#).

*If you have questions or feedback about this tutorial, please leave comments using the Google Docs "Add comment" feature, or post your question in the [IDC user forum](#)!*

# Table of contents

[Table of contents](#)

[Prerequisites](#)

[Costs](#)

[1. Creating a Firebase project](#)

[2. Enabling APIs in Google Cloud Platform](#)

[3. Creating OAuth consent screen in Google Cloud Platform](#)

[4. Creating OAuth credentials in the Google Cloud Platform](#)

[5. Configuring and deploying the Slim viewer application with Firebase](#)

[6. Updating OAuth Client and Consent Screen configurations](#)

[7. Authorizing Slim to access user information](#)

[8. Creating and populating a Google Healthcare DICOM store](#)

[Granting the necessary permissions](#)

[Creating and populating a DICOM store](#)

[9. Configuring the Slim viewer with your own DICOM store](#)

[10. Developing the Slim viewer](#)

[11. Cleaning up](#)

[Deleting the project](#)

[Deleting the Google storage bucket and DICOM store](#)

[Further resources](#)

## Prerequisites

- [Google account](#)
- [Create or select a Google Cloud project](#)
- [Enable billing for your project](#)
  - You will need to enable billing in order to enable the APIs used for creating the OAuth credentials, and in order to create DICOM stores with data.
  - All first time Google Cloud users have the opportunity to access \$300 of free cloud credits to support billing for your project

## Costs

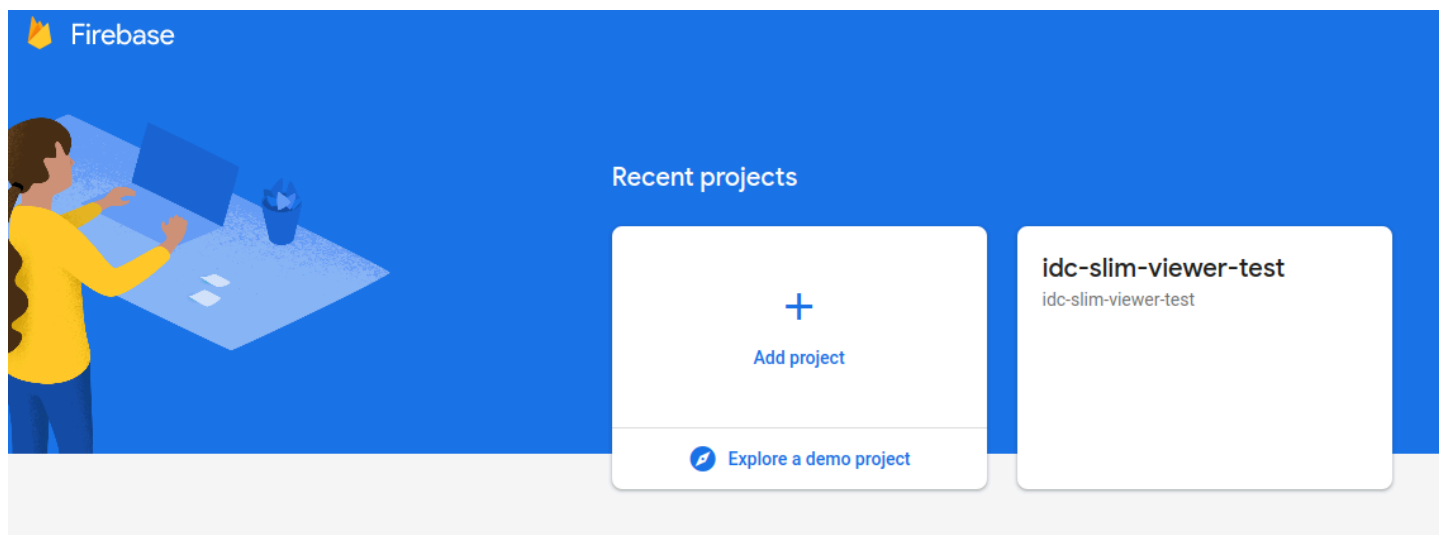
This tutorial uses the following billable components of Google Cloud:

- [Cloud Storage](#): Cloud storage is only needed to populate the DICOM store with content and can be released once the data is imported into a DICOM store
- [Healthcare API](#)



## 1. Creating a Firebase project

First of all, go to the [Firebase console](#) and select the option to create a new project. When prompted for the name of the project, **do not** select any of your existing Google Cloud projects, since that will result in associating your project with your billing account, and you will be forced to use the “Blaze - Pay as you go” plan. Instead, type a new name in the “Enter your project name” field as creating a new project will allow you to use the free “Spark” plan. Then proceed with the following steps, choose your desired Google Analytics configuration and complete the creation of the project. When you return to the Firebase console, you should see a tile for your project. Make note of your project name, as you will need it again later.



## 2. Enabling APIs in Google Cloud Platform

Slim will use two APIs: The Cloud Resource Manager API to access the list of projects available to the user of the viewer, and the Cloud Healthcare API to access the listing of the available DICOM stores and retrieve the content of those stores. Both of those APIs are disabled by default, and can only be enabled in a project with billing enabled. To enable those APIs, visit the Marketplace pages for the [Cloud Resource Manager API](#) and the [Healthcare API](#), respectively. For both make sure that your correct project with billing enabled is selected in the upper left corner and then click the “ENABLE” button.

## 3. Creating OAuth consent screen in Google Cloud Platform

The [OAuth Consent screen](#) provides the user of your application with information about the developer of the application, the application itself and the permissions (scopes) that the application is requesting access to. You can locate the [OAuth Consent screen page](#) in the Google Cloud console by searching for “oauth consent screen” or navigating to API & Services > OAuth consent screen.

Unless you have a Google organization within which you want to share the application, you will need to choose “External” as User type in the preparation step for the consent screen registration (see left image below).

## 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

## Edit app registration

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

In (1) **OAuth consent screen** assign a name to your app and specify the contact emails.

In (2) **Scopes**, click “Add or remove scopes” and search for the Healthcare and Cloud Resource Manager APIs to add those. For the Cloud Resource Manager API select the “../auth/cloudplatformprojects.readonly” scope (as a rule of thumb, choose the minimum scopes required by the application - in this case, we only want the Slim viewer to be able to see the list of projects under the user account).

✕ Update selected scopes

? Only scopes for enabled APIs are listed below. To add a missing scope to your app, enable the API in the [Google API Library](#) or use the Pasted Scopes text box below. Remember to enable the APIs you enable from the Library.

Filter healthcare

API	Scope	User-facing description
<input type="checkbox"/>	Cloud Healthcare API	View your primary Google A...
<input type="checkbox"/>	https://www.googleapis.com/auth/cloud-healthcare	Read, write and manage healthcare data
<input type="checkbox"/>	openid	Associate you with your pe...
<input type="checkbox"/>	https://www.googleapis.com/auth/healthcare	View and manage your dat...

Filter Cloud Resource Manager API + Enter property name or value ✕ ?

API	Scope	User-facing description
<input type="checkbox"/>	Cloud Resource Manager API	../auth/cloudplatformprojects Manage your Cloud Platform projects
<input checked="" type="checkbox"/>	Cloud Resource Manager API	../auth/cloudplatformprojects.readonly View your Cloud Platform projects
<input type="checkbox"/>	Cloud Resource Manager API	../auth/cloudplatformorganizations Manage your Cloud Platform organizations

If everything was done correctly, you should see the following scopes configuration for your consent screen:

## 🔒 Your sensitive scopes

Sensitive scopes are scopes that request access to private user data.

API ↑	Scope	User-facing description	
Cloud Healthcare API	.../auth/cloud-healthcare	Read, write and manage healthcare data	🗑️
Cloud Resource Manager API	.../auth/cloudplatformprojects.readonly	View your Cloud Platform projects	🗑️

In the **(3) Test users** add the Google account email you will be using to access the Slim viewer. This is also the place where you will later be able to add emails of other users you want to be able to access the viewer.

## 4. Creating OAuth credentials in the Google Cloud Platform

You can locate the [Credentials page](#) in the Google cloud console by searching for “Credentials” or by navigating to API & Services > Credentials. On that page, click “+ Create credentials”, and choose “OAuth Client ID”. Set the application type to “Web application”. You can leave everything else as it is for now and click “Create”. Once the OAuth client is created, take note of the **Client ID**, which should look like a sequence of characters ending with `.apps.googleusercontent.com`.

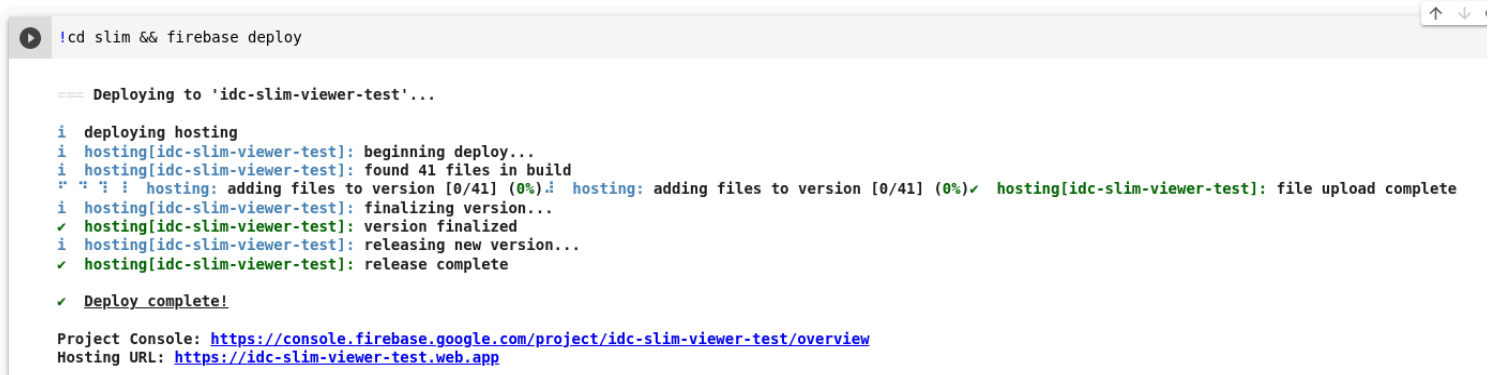
We will return to the configuration of the OAuth client after deploying the viewer to Firebase to configure the “Authorized redirect URI” section.

## 5. Configuring and deploying the Slim viewer application with Firebase

The following steps are documented in a [step-by-step Google Colab notebook](#) that you can execute to complete the deployment. Make sure you have your **GCP Project ID**, the **Client ID from the previous step** and the **name of your Firebase project** from Step 1 available as you will need it to update Slim’s configurations.

In the end, you should have two links (though no need to click on them now, we are not yet done): one to the project console and a hosting URL.

Finally, we deploy the app.



```
!cd slim && firebase deploy

=== Deploying to 'idc-slim-viewer-test'...

i deploying hosting
i hosting[idc-slim-viewer-test]: beginning deploy...
i hosting[idc-slim-viewer-test]: found 41 files in build
" " " " : hosting: adding files to version [0/41] (0%): hosting: adding files to version [0/41] (0%)✓ hosting[idc-slim-viewer-test]: file upload complete
i hosting[idc-slim-viewer-test]: finalizing version...
✓ hosting[idc-slim-viewer-test]: version finalized
i hosting[idc-slim-viewer-test]: releasing new version...
✓ hosting[idc-slim-viewer-test]: release complete

✓ Deploy complete!

Project Console: https://console.firebase.google.com/project/idc-slim-viewer-test/overview
Hosting URL: https://idc-slim-viewer-test.web.app
```

## 6. Updating OAuth Client and Consent Screen configurations

Now that you have the hosting URL for your viewer deployment, we need to update the OAuth client configuration we created earlier as well as the consent screen to include that URL as an authorized domain.

Proceed again to the [Credentials page](#), and click the pencil icon / “Edit OAuth client”. Here add the fully qualified hosting URL in the “Authorized JavaScript origins” section, and also in the “Authorized redirect URIs” section. For the latter, **post-fix the URL with a slash**. Don’t forget to save your changes.

### Authorized JavaScript origins

For use with requests from a browser

URIs 1 \*

[+ ADD URI](#)

### Authorized redirect URIs

For use with requests from a web server

URIs 1 \*

[+ ADD URI](#)

Ideally, adding the hosting URL to the OAuth client should also update the OAuth consent screen’s authorized domains. To check, proceed again to the [OAuth consent screen](#), and click “Edit app”. If you can’t find the hosting URL in **(1) OAuth consent screen** in the “Authorized domains” section add it manually, **omitting the “https://” prefix**. Don’t forget to save your changes.

### Edit app registration

[1 OAuth consent screen](#) — [2 Scopes](#) — [3 Test users](#) — [4 Summary](#)

### Authorised domains

When a domain is used on the consent screen or in an OAuth client’s configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorised. [Learn more](#) about the authorised domain limit.

Authorised domain 1 \*

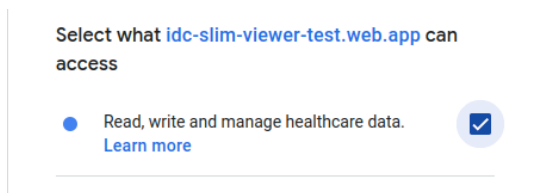
[+ ADD DOMAIN](#)

## 7. Authorizing Slim to access user information

At this point, your Slim viewer instance should be ready to be used.

However, as a user, you need to give it the permissions to access the the content of the Google Healthcare DICOM stores, so that the DICOM data can be retrieved and visualized in the browser. To do this, proceed to the hosting URL and choose the Google account you want to use with this application. Note that the account will need to be listed as a “Test user” account in the configuration of the Consent screen, as we discussed in the [“Creating OAuth consent screen”](#) section.

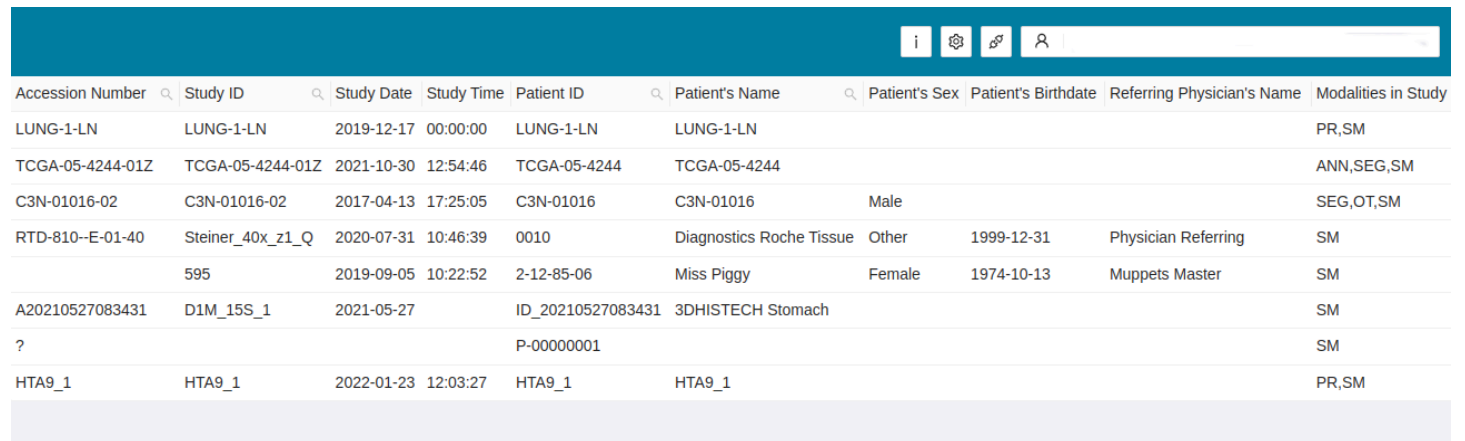
You will be prompted that the application is [not verified by Google](#), and will need to decide if you want to continue with this application. If you do decide to proceed, you will be prompted with the listing of the scopes the application will need. You will have to check the “Read, write and manage healthcare data” scope we discussed earlier, which will not be checked by default.



Select what `idc-slim-viewer-test.web.app` can access

Read, write and manage healthcare data. [Learn more](#)

If everything worked as expected, after completing this step, you should be prompted with the listing of studies you can visualize. Note that this deployment does access a default DICOM store. You will learn in the following section how to set-up and access your own Google Cloud Healthcare DICOM store.



Accession Number	Study ID	Study Date	Study Time	Patient ID	Patient's Name	Patient's Sex	Patient's Birthdate	Referring Physician's Name	Modalities in Study
LUNG-1-LN	LUNG-1-LN	2019-12-17	00:00:00	LUNG-1-LN	LUNG-1-LN				PR,SM
TCGA-05-4244-01Z	TCGA-05-4244-01Z	2021-10-30	12:54:46	TCGA-05-4244	TCGA-05-4244				ANN,SEG,SM
C3N-01016-02	C3N-01016-02	2017-04-13	17:25:05	C3N-01016	C3N-01016	Male			SEG,OT,SM
RTD-810--E-01-40	Steiner_40X_z1_Q	2020-07-31	10:46:39	0010	Diagnostics Roche Tissue	Other	1999-12-31	Physician Referring	SM
	595	2019-09-05	10:22:52	2-12-85-06	Miss Piggy	Female	1974-10-13	Muppets Master	SM
A20210527083431	D1M_15S_1	2021-05-27		ID_20210527083431	3DHISTECH Stomach				SM
?				P-00000001					SM
HTA9_1	HTA9_1	2022-01-23	12:03:27	HTA9_1	HTA9_1				PR,SM

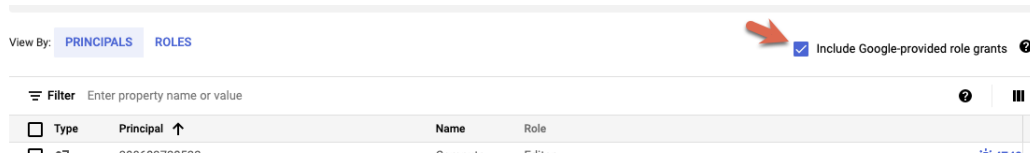
## 8. Creating and populating a Google Healthcare DICOM store

### *Granting the necessary permissions*

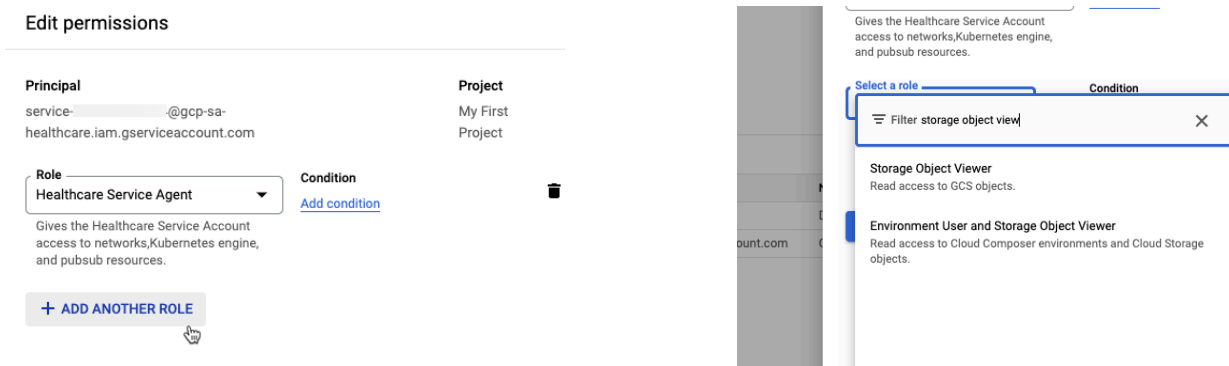
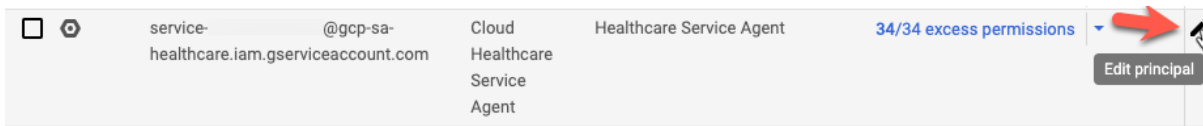
In order to complete this step you will first need to configure the Cloud Healthcare Service Agent (SA) account to have the Storage Object Viewer role. This is needed to import DICOM files from a GCP Storage Bucket into a Google Healthcare DICOM store.



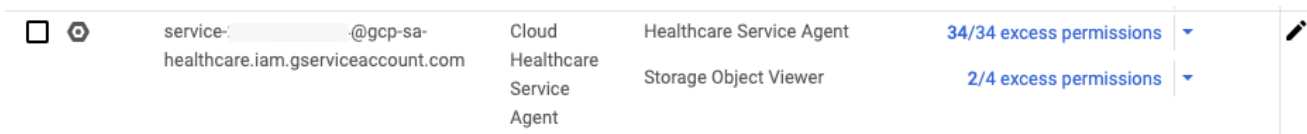
To add this role, navigate to the IAM & Admin > IAM section in the cloud console. There you will see the list of accounts and the respective permissions given to those accounts. Include the Google-provided role grants by checking the checkbox in the upper right:



Next, look for the row corresponding to the Cloud Healthcare Service Agent, click the pencil icon to edit the permissions and then click “Add Another Role”. Search and select the “Storage Object Viewer” role and hit “Save”.



Once completed, you should see the newly added role corresponding to the Cloud Healthcare Service Agent as shown below:



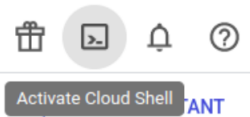
## Creating and populating a DICOM store

After granting the permissions needed, you can proceed to the actual creation and population of your own DICOM store. You might either do this by executing the steps outlined in [this notebook](#) (although it is part of a tutorial for the radiology OHIF viewer, most parts apply to our case as well) or via the Google Cloud console as described in the following.

### Step 1: Creation and population of your temporary storage bucket

- First of all, you need to download your sample DICOM dataset from IDC into a (temporary) GCP storage bucket. Please go to Cloud Storage > Buckets and click “+Create”. Name your bucket to your liking and choose “Region” as location type and select a region of your liking. Confirm by clicking “Create”.

- Next, activate the Google Cloud Shell and use gsutil or s5cmd to copy your DICOM files to the storage bucket. You can find more detailed information on how to do so in the above mentioned notebook.



## Step 2: Creation of your DICOM store

- Before you can create a DICOM store you need to create a Dataset first. Thus, go to Healthcare > Browser and click “+ Create Dataset”. Provide a name, choose “Region” as location type, select a region of your choice and hit the “Create” button. Then return to the Healthcare Browser.
- Click on the dataset you just created and then on “+ Create Data Store”. Select DICOM as “Type” and provide an ID for your DICOM store. Everything else can be left as it is and you can hit the “Create” button. Return

### ← Create dataset

#### Dataset properties

Name \*  
test\_dicom\_dataset

A permanent identifier for this dataset

#### Location

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

#### Location type

- Region  
Lower latency within a single region
- Multi-region  
Highest availability across largest area

Region \*  
us-central1 (Iowa)

CREATE CANCEL

### ← Create Data Store

Datasets / single-dicom-annotation-test (us-central1)

#### • Choose a data store type and ID

Data stores hold modality-specific healthcare data in the Cloud Healthcare API. Each modality has different structural and processing characteristics, but they are all backed by a standards-compliant data store that provides read, write, search, and other operations on the data.

Type \*  
DICOM

ID \*  
test\_dicom\_store

Only numbers, letters, underscores, hyphens, and periods are allowed

NEXT

- Stream resource changes to BigQuery
- Receive Cloud Pub/Sub notifications
- Add labels to organize your data stores

CREATE CANCEL

\* indicates required field

## Step 3: Population of your DICOM store

- To populate your DICOM store, return once more to the Healthcare Browser and click on your dataset. You should now see the DICOM store you just created being listed. On the right, open the “Actions” drop-down menu and select “Import”.
- You can then specify from where to import your DICOM data, which in this case is the GCP bucket your created in Step 1. When you are done, hit the “Import” button.

### Cloud Storage Source

Bulk import DICOM files (.dcm) from Cloud Storage to a DICOM store.

Project \*  
global-pagoda-294415 BROWSE

Location \* BROWSE

### Advanced Options

IMPORT CANCEL

\* indicates required field

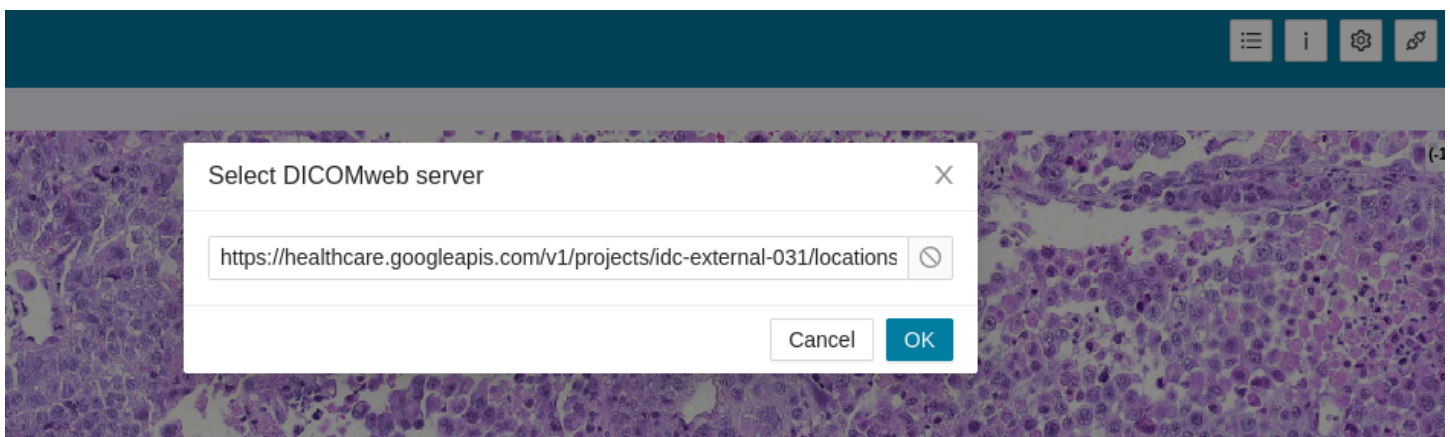
## 9. Configuring the Slim viewer with your own DICOM store

After the creation and population of your own DICOM store, you create the URL to access the DICOMweb endpoint of your DICOM store by updating the string below:

- MY\_PROJECT\_ID: project under which you created the Healthcare dataset
- LOCATION: location you chose for your Healthcare dataset
- DATASET\_ID: dataset ID/name you specified earlier while creating the dataset
- STORE\_ID: DICOM store ID you specified earlier

**[https://healthcare.googleapis.com/v1/projects/MY\\_PROJECT\\_ID/locations/LOCATION/datasets/DATASET\\_ID/dicomStores/STORE\\_ID/dicomWeb](https://healthcare.googleapis.com/v1/projects/MY_PROJECT_ID/locations/LOCATION/datasets/DATASET_ID/dicomStores/STORE_ID/dicomWeb)**

Go back to your Slim webpage, click the “Select server” button in the upper right corner and input the updated URL you just put together.



YOU'RE DONE! You should now be able to visualize all your DICOM data that you have in your DICOM store.

## 10. Developing the Slim viewer

Every time you complete the steps in the notebook as specified in the aforementioned [step-by-step Google Colab notebook](#), the version corresponding to the checked out source code will be deployed to your Firebase project.

## 11. Cleaning up

After you've finished this tutorial, you can clean up the resources that you created on Google Cloud so that you won't be billed for them in the future. The following sections describe how to delete or turn off these resources.

### *Deleting the project*

The easiest way to eliminate billing is to delete the project you created for the tutorial. If you don't want to delete the project, delete the individual resources, as described in the next section.

**Warning:** Deleting a project has the following consequences:

- If you used an existing project, you'll also delete any other work you've done in the project.
- You can't reuse the project ID of a deleted project. If you created a custom project ID that you plan to use in the future, you should delete the resources inside the project instead. This ensures that URLs that use the project ID, such as an appspot.com URL, remain available.

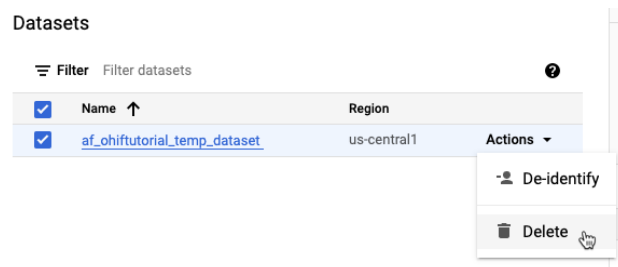
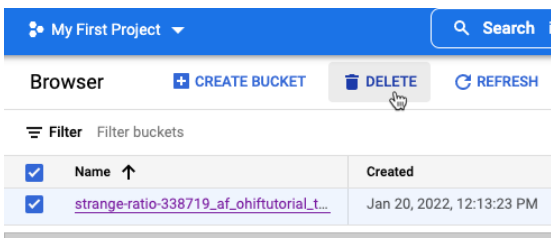
If you are exploring multiple tutorials and quickstarts, reusing projects instead of deleting them prevents you from exceeding project quota limits.

To delete the project:

1. In the Cloud Console, go to the [Projects page](#).
2. Click the checkbox next to the project you want to delete.
3. Click the **Delete** button at the top of the page.
4. In the dialog, type the project ID, and then click **Shut down** to delete the project.

### *Deleting the Google storage bucket and DICOM store*

Navigate to Cloud Storage > Buckets in the console, select the bucket you created and hit "DELETE". Then navigate to Healthcare > Browser and select the "Delete" option from the drop-down on the right side of the dataset row.



## Further resources

GCP tutorial contribution guide: <https://cloud.google.com/community/tutorials/write>

Example tutorial: <https://cloud.google.com/community/tutorials/setting-up-lamp>

Tutorial template: <https://github.com/GoogleCloudPlatform/community/blob/master/tutorials/tutorial-template/index.md>