

[Start Lab](#)

01:52:30

De-identifying DICOM Data with the Healthcare API

 Lab  1 hour 15 minutes  No cost  Intermediate**GSP626****Google Cloud Self-Paced Labs**

Lab instructions and tasks

GSP626

Overview

Setup and requirements

Task 1. Create Healthcare dataset

Task 2. Set up IAM permissions

Task 3. Enable data access logs on Cloud Healthcare

Task 4. Define variables needed

Task 5. Create data stores

Task 6. Import to DICOM datasets

Task 7. Configure OHIF Viewer

Task 8. Using De-



Overview

In this lab you will discover and use the de-identification functionality of Cloud Healthcare API using Digital Imaging and Communications in Medicine (DICOM) data model.

In this lab, you will:

- Gain a general understanding of Cloud Healthcare API and its role in managing healthcare data.
- Learn how to create Cloud Healthcare API datasets and stores.
- Import and Export DICOM data using the Cloud Healthcare API.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause

extra charges incurred to your personal account.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

Project_ID, **PROJECT_ID**. The output contains a line that declares the **Project_ID** for this session:

```
Your Cloud Platform project in this session is set to "PROJECT_ID"
```

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

2. (Optional) You can list the active account name with this command:

```
gcloud auth list
```



3. Click **Authorize**.

Output:

```
ACTIVE: *
ACCOUNT: "ACCOUNT"

To set the active account, run:
$ gcloud config set account `ACCOUNT`
```

4. (Optional) You can list the project ID with this command:

```
gcloud config list project
```



Output:

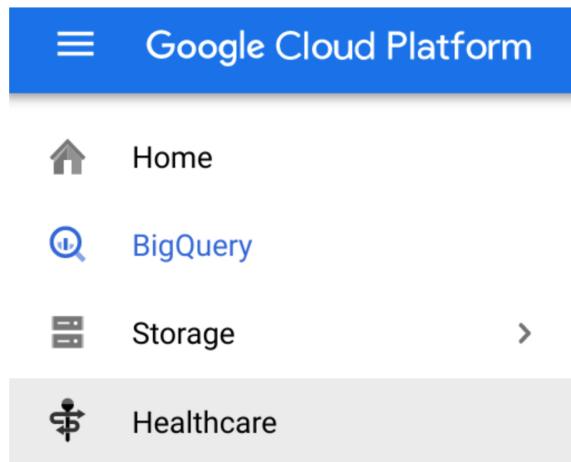
```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to [the gcloud CLI overview guide](#).

Task 1. Create Healthcare dataset

In this exercise you will use the UI to create a Cloud Healthcare API dataset.

1. Under the **Navigation Menu** (≡), select **Healthcare > Browser** and then **Enable** the API.



2. Once the API is enabled, in the Healthcare browser select **Create Dataset**.

3. Name the dataset **dataset1** within region **REGION** and click **Create**.

A screenshot of the "Create dataset" form. At the top, it shows the "Dataset properties" section. In the "Name" field, "dataset1" is entered. Below it, the "Region" dropdown is set to "us-central1". At the bottom of the form are two buttons: "CREATE" (in blue) and "CANCEL". A note below the form says "Click Check my progress to verify the objective." followed by a "Check my progress" button. The entire form is contained within a light gray box.

Task 2. Set up IAM permissions

1. From the **Navigation menu** (≡), go to **IAM & admin > IAM**.
2. In the **IAM** page, select the **Include Google-provided role grants** checkbox.
3. Edit the permissions for your **Healthcare Service Agent** by locating the service agent under the IAM list and selecting the pencil icon. The service account will have the Domain `@gcp-sa-healthcare.iam.gserviceaccount.com`.
4. Click **Add another role** to add additional roles to the Healthcare Service Agent account.

account.

5. Click inside the Select a role box and choose the following roles:

- Cloud Storage > Storage Object Admin
- Cloud Healthcare > Healthcare Dataset Administrator
- Cloud Healthcare > Healthcare DICOM Editor

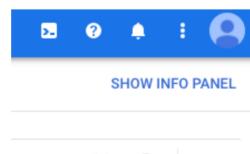
6. After all of the roles are added, select **Save** to commit your updates.

Task 3. Enable data access logs on Cloud Healthcare

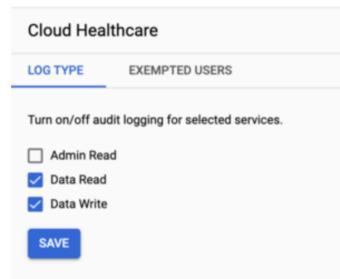
1. From the IAM & Admin menu, navigate to **Audit Logs**.

2. Scroll or use the filter box to locate **Cloud Healthcare**, then check the box next to it to select.

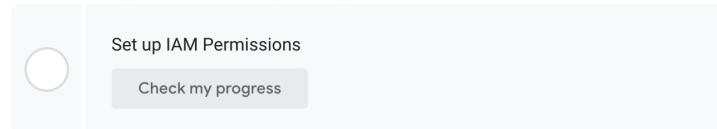
3. If the info panel isn't already open on the right side of the interface, click the **Show Info Panel** link.



4. Select **Data Read** and **Data Write**, then click **Save**.



Click *Check my progress* to verify the objective.



Task 4. Define variables needed

- In Cloud Shell, export the variables needed for the lab:

```
export PROJECT_ID='gcloud config get-value project'  
export REGION="REGION"  
export DATASET_ID=dataset1  
export DICOM_STORE_ID=dicomstore1
```

Task 5. Create data stores

Data in Cloud Healthcare API datasets and stores can be accessed and managed using a REST API that identifies each store using its project, location, dataset, store type and store name. This API implements modality-specific standards for access that are consistent with industry standards for that modality. For example, the Cloud Healthcare DICOM API natively provides operations for reading DICOM studies and series that are consistent with the DICOMweb standard, and supports the DICOM DIMSE C-STORE protocol via an [open-source adapter](#).

1. Call the API to create a DICOM store:

```
gcloud beta healthcare dicom-stores create $DICOM_STORE_ID --  
dataset=$DATASET_ID --location=$REGION
```

The server returns a path to the newly created store.

Users can also use the `curl` utility to issue Cloud Healthcare API calls. `curl` is pre-installed in your Cloud Shell machine. By default, `curl` does not show HTTP status codes or session-related information; if you would like to see this information please add the `-v` option to all commands in this tutorial.

2. Try creating a secondary FHIR store by using the below command:

```
curl -X POST \  
-H "Authorization: Bearer "$(sudo gcloud auth print-access-  
token) \  
-H "Content-Type: application/json; charset=utf-8" \  
"https://healthcare.googleapis.com/v1beta1/projects/$PROJECT_ID/loc  
dicomStoreId=dicomstore2"
```

Operations that access a modality-specific store use a request path that is comprised of two pieces: a base path, and a modality-specific request path.

Administrative operations—which generally operate only on locations, datasets and stores—may only use the base path. Data modality-specific retrieval operations use both the base path (for identifying the store to be accessed) and request path (for identifying the actual data to be retrieved).

Click *Check my progress* to verify the objective.



Create data stores

[Check my progress](#)

Note: If this check fails, wait a minute and try again. It often takes a minute or two for the import operation to be logged.

Task 6. Import to DICOM datasets

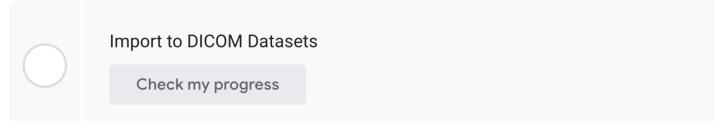
In this section you will be importing data from the NIH Chest x-ray data set to a DICOM store. For more information on the public dataset, visit the [NIH Chest X-ray dataset documentation](#).

- Call the API to use the import functionality:

```
gcloud beta healthcare dicom-stores import gcs $DICOM_STORE_ID -  
-dataset=$DATASET_ID --location=$REGION --gcs-  
uri=gs://splx/gsp626/LungCT-Diagnosis/R_004/*
```



Click *Check my progress* to verify the objective.



Task 7. Configure OHIF Viewer

The Open Health Imaging Foundation (OHIF) Viewer is an open source, web-based, medical imaging viewer. You will use **OHIF Viewer** in this lab to view your DICOM dataset.

The following steps will walk through setting up **OHIF Viewer** to view your dataset:

1. First, select **APIs & Services > OAuth Consent Screen** from the Navigation Window to create an OAuth Consent screen:

The screenshot shows the Google Cloud Platform navigation bar at the top with the title 'Google Cloud Platform' and a user icon. Below the navigation bar, there's a sidebar with various links: Home, Marketplace, Billing, APIs & Services, Support, Getting started, Admin, and Compliance. The 'APIs & Services' link is selected, and its dropdown menu is open, showing 'Cloud Healthcare' and several sub-options: Title ↑, Cloud Healthcare, Dashboard, Library, Credentials, OAuth consent screen (which is highlighted with a green box), Domain verification, and Page usage agreements.

2. At the **OAuth Consent Screen**, select **Internal** and click **Create**:

User Type

Internal [?](#)

Only available to users within your organization. You will not need to submit your app for verification.

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](#).

CREATE

3. Fill out the following on the **Edit app registration** window:

- App name: **QL-de-identify**
- User support email: **YOUR STUDENT EMAIL** (this is provided by the lab)
- Developer contact information: **YOUR STUDENT EMAIL** (same value as *user support email*)

App name *
QL-de-identify
The name of the app asking for consent

User support email *
student-01-faa0e3dba6d9@qwiklabs.net
For users to contact you with questions about their consent

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

App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page
Provide users a link to your home page

Application privacy policy link
Provide users a link to your public privacy policy

Application terms of service link
Provide users a link to your public terms of service

Authorized 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 authorized. [Learn more](#) about the authorized domain limit.

+ ADD DOMAIN

Developer contact information

Email addresses *
student-01-faa0e3dba6d9@qwiklabs.net 
These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE

CANCEL

4. Click **Save and Continue**.

5. At the **Scopes** tab, click the **Add or Remove Scopes** button.

6. Scroll to the bottom of the pop-up window to the **Manually add scopes** section.

7. Add the following scopes:

`https://www.googleapis.com/auth/cloudplatformprojects.readonly`
`https://www.googleapis.com/auth/cloud-healthcare`



Manually add scopes

If the scopes you would like to add do not appear in the table above, you can enter them here. Each scope should be on a new line or separated by commas. Please provide the full scope string (beginning with "https://"). When you are finished, click "Add to table".

`https://www.googleapis.com/auth/cloudplatformprojects.readonly`
`https://www.googleapis.com/auth/cloud-healthcare`

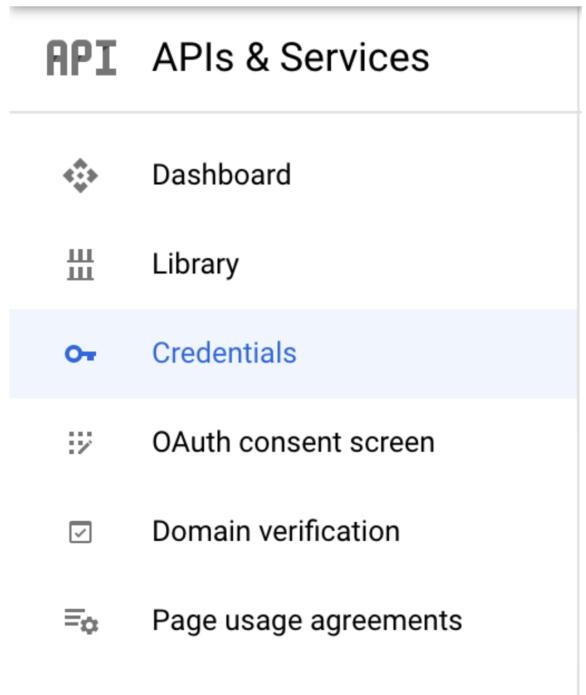
ADD TO TABLE

UPDATE

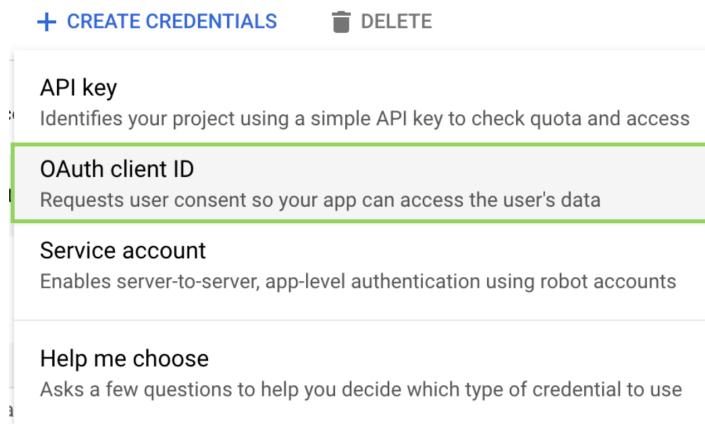
8. Click **Add to table** and then click **Update**.
9. Scroll to the bottom of the **Scopes** tab and click **Save and Continue**.

Next, you'll need an **OAuth Client ID** to connect **OHIF Viewer** to your **Cloud Healthcare** resources.

1. Select **Credentials** from the **APIS & Services** menu:



2. In the **Credentials** page, click **+ Create Credentials** > **OAuth Client ID**:



3. For your **Application Type**, choose **Web application**.

You will need to return to your client ID and fill out the domains once your **OHIF Viewer** application has been launched.

4. So, for now, leave everything as default and click **Create**.

You'll now see your **Client ID** and **Client Secret** in the next window.

5. Click **OK** to close the window.

Now, deploy the **OHIF Viewer** container to **Cloud Run** and connect it with your OAuth Client ID.

To simplify the setup, the **OHIF Viewer** docker image already exists in **container registry** in a project you have access to, so you can directly deploy the container to **Cloud Run**.

1. In **Cloud Shell**, deploy the **OHIF Viewer** container to **Cloud Run** with this command substituting PASTE-CLIENT-ID-HERE with the **Client ID** of the OAuth Client you just created:

```
gcloud run deploy ohif-viewer --image=gcr.io/qwiklabs-resources/ohif-viewer:latest --platform=managed --region="REGION" --allow-unauthenticated --set-env-vars=CLIENT_ID=[PASTE-CLIENT-ID-HERE] --max-instances=3
```

Note: You can view and copy your Client ID in the **Credentials** tab:

OAuth 2.0 Client IDs				
<input type="checkbox"/>	Name	Creation date	Type	Client ID
<input type="checkbox"/>	Web client 1	Jan 23, 2021	Web application	567555637566-1ct4... 

2. If asked to enable the **Cloud Run API**, enter **y** and continue.

Once your Cloud Run deployment completes, you will be given a **unique** service URL that looks similar to this:

Service **URL**: <https://ohif-viewer-ratpkirjdq-uc.a.run.app>

3. You can now return to your OAuth Client ID and update the domains with this **Service URL**.

4. If you're not still on the **Credentials** page, select **APIs & Services > Credentials** from the Navigation Menu in your Cloud Console.

5. Edit your Client ID by clicking the pencil icon.

OAuth 2.0 Client IDs				
<input type="checkbox"/>	Name	Creation date	Type	Client ID
<input type="checkbox"/>	Web client 1	Jan 6, 2021	Web application	1674324282798-dp... 

6. Add your **unique** service URL to **Authorized Javascript Origins**.

7. Add your **unique** service URL + `/callback` to **Authorized Redirect URIs**.

8. Click **Save**.

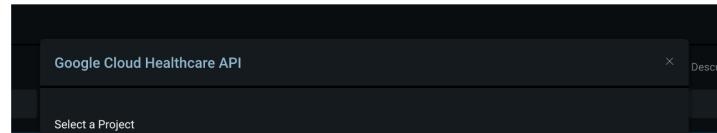
Task 8. Using De-identification

De-identification (redacting or transformation) of sensitive data elements is often an important step in pre-processing healthcare data so that it can be made available for analysis, machine learning models, and other use cases. Cloud Healthcare API has the capability to de-identify data stored in the service, facilitating analysis by researchers or machine learning analysis for advanced anomaly scans.

1. First, navigate to the service URL of your **ohif-viewer** Cloud Run app and sign in using your lab credentials. If you've lost track of your service URL, you can find it again with this command:

```
gcloud run services list --platform managed
```

2. Once on the **OHIF-Viewer** page, select your **Project ID** for the **Project**.



Project	ID
qwiklabs-gcp-03-af95a9cb5336	qwiklabs-gcp-03-af95a9cb5336
dw-workshop	dw-workshop
hcls-testing-data	hcls-testing-data
bq-taw-data	bq-taw-data
Qwiklabs Resources	qwiklabs-resources
Google Cloud Shell	cloudshell-images
Cloud Solutions Project	cloud-solutions-group

3. Select **REGION** for the location.

4. Select **dataset1** for your dataset.

5. Select **dicomstore1** in the DICOM Store window.

You'll see one entry, **R_004** with info for its ID number, Study Date, and Description:

6. Click on the entry to inspect it further and view the associated images.

7. This dataset contains pre-surgery images of a chest. You can scroll through them to view them all.

8. When you're done looking at it, press the **Back** button on your browser to return to the **OHIF-Viewer** main menu.

Next, you will de-identify this dataset.

1. Navigate back to Cloud Shell and issue the following request to de-identify the dataset:

```
curl -X POST \
  -H "Authorization: Bearer "$(gcloud auth print-access-token)
  \
  -H "Content-Type: application/json; charset=utf-8" \
  --data "{
    'destinationDataset':
      'projects/$PROJECT_ID/locations/$REGION/datasets/de-id',
    'config':
      {
        'dicom':
          {
            'filterProfile':
              'ATTRIBUTE_CONFIDENTIALITY_BASIC_PROFILE'
          },
        'image':
          {
            'textRedactionMode': 'REDACT_NO_TEXT'
          }
      }
  }"
  "https://healthcare.googleapis.com/v1beta1/projects/$PROJECT_ID/loc
```

With our small dataset, this operation will be done quickly, but on a larger dataset this operation can take a few minutes.

2. You can issue a rest request to check the status of a long running operation, replacing **<operation-ID>** with the operations ID issued in the previous output:

```
curl -X GET \
  "https://healthcare.googleapis.com/v1beta1/projects/$PROJECT_ID/loc
  id" \
  -H "Authorization: Bearer "$(sudo gcloud auth print-access-
  token) \
  -H 'Content-Type: application/json; charset=utf-8'
```

If you see "done": true in the output of the previous command, you can be sure that your operation is complete.

Once the operation is complete a new **de-id** dataset will appear on the Healthcare UI [here in the Console](#).

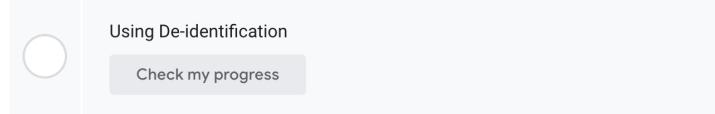
page in the Console.

1. Confirm the identifiable information has been redacted by returning to your OHIF-Viewer browser tab and selecting the **Change DICOM Store** button.
2. In the window that pops up, select your Qwiklabs **Project ID** as the Project.
3. Select **REGION** for the location.
4. Select **de-id** as the dataset.
5. Select **dicomstore1** for the DICOM Store.

You'll now see one entry in the DICOM Store, but the outward facing information/tags have been removed:

6. Select the entry to confirm it's the same images copied from the previous dataset but with most of its information removed.

Click *Check my progress* to verify the objective.



Task 9. Converting DICOM Images

From the **Navigation menu**, navigate to **Cloud Storage > Buckets**.

1. Click **Create bucket**.
2. Fill out the first box with a unique name and click **Continue**.
3. Set the Location type to **Region** and select the region **REGION**.
4. Click **Create**.
5. Using Cloud Shell export the variable for your newly created bucket, replacing **<name of bucket>** with your bucket's name:

```
export BUCKET_ID=<name of bucket>
```



Now you can export the DICOM images into JPEG or PNG using a `gcloud` command.

6. Export the DICOM images into JPEG:

```
gcloud beta healthcare dicom-stores export gcs $DICOM_STORE_ID -  
-dataset=$DATASET_ID --gcs-uri-prefix=gs://$BUCKET_ID/ --mime-  
type="image/jpeg; transfer-syntax=1.2.840.10008.1.2.4.50" --  
location=$REGION
```



OR

Export the DICOM images into PNG:

```
gcloud beta healthcare dicom-stores export gcs $DICOM_STORE_ID -  
-dataset=$DATASET_ID --gcs-uri-prefix=gs://$BUCKET_ID/ --mime-  
type="image/png" --location=$REGION
```

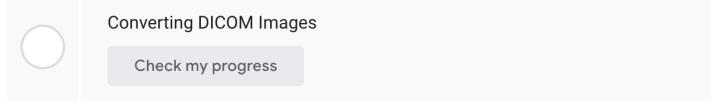


7. In the Console, from the **Navigation menu** navigate to **Cloud Storage** and click on your bucket.

8. Select a folder, click on an image, then click on the Link URL. This will download the image.

9. You can check the file extension to verify your file is correct or click the image to view.

Click *Check my progress* to verify the objective.



Lab review

Cloud Healthcare API provides a comprehensive facility for ingesting, storing, managing, and securely exposing healthcare data in FHIR, DICOM, and HL7 v2 formats. Using Cloud Healthcare API, you can ingest and store data from electronic health records systems (EHRs), radiological information systems (RISs), and custom healthcare applications. You can then immediately make that data available to applications for analysis, machine learning prediction and inference, and consumer access.

Cloud Healthcare API enables application access to healthcare data via widely-accepted, standards-based interfaces such as FHIR STU3 and DICOMweb. These APIs allow data ingestion into modality-specific data stores, which support data retrieval, update, search and other functions using familiar standards-based interfaces.

Further, the API integrates with other capabilities in Google Cloud through two primary mechanisms:

- *Cloud Pub/Sub*, which provides near-real-time updates when data is ingested into a Cloud Healthcare API data store, and
- *Import/export APIs*, which allow you to integrate Cloud Healthcare API into both Google Cloud Storage and Google BigQuery.

Using Cloud Pub/Sub with Google Cloud Functions enables you to invoke machine learning models on healthcare data, storing the resulting predictions back in Cloud Healthcare API data store. A similar integration with Cloud Dataflow supports transformation and cleansing of healthcare data prior to use by applications.

To support healthcare research, Cloud Healthcare API offers de-identification capabilities for FHIR and DICOM. This feature allows customers to share data with researchers working on new cutting-edge diagnostics and medicines.

Congratulations!

In this lab you:

- Gained a general understanding of Cloud Healthcare API and its role in managing healthcare data.
- Learned how to create datasets and stores for FHIR and DICOM data.
- Imported FHIR and DICOM data.

Finish your quest

This self-paced lab is part of the Cloud Healthcare API quest. A quest is a series of related labs that form a learning path. Completing this quest earns you a badge to recognize your achievement. You can make your badge or badges public and link to

them in your online resume or social media account. Enroll in any quest that contains this lab and get immediate completion credit. See the [Google Cloud Skills Boost catalog](#) to see all available quests

Take your next lab

Continue your quest with [Ingesting FHIR Data with the Healthcare API](#) or try one of these suggestions:

- [Ingesting HL7v2 Data with the Healthcare API](#)
- [Ingesting DICOM Data with the Healthcare API](#)

Manual Last Updated October 26, 2023

Lab Last Tested October 27, 2023

Copyright 2024 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.