

App Dev-Storing Image and Video Files in Cloud Storage-Java

GSP168



Overview

Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download.

In this lab you'll configure an application to use Cloud Storage to store and retrieve application data. The application is an online Quiz, the data is the form data, including an image you upload from your local machine.

Objectives

In this lab, you learn how to perform the following tasks:

- Set up Cloud Shell as your development environment
- Update the application code to integrate Cloud Datastore
- Use the Quiz application to upload an image file into Cloud Storage and view the image in the Quiz

Setup

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 Qwiklabs 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.

What you need

To complete this lab, you need:

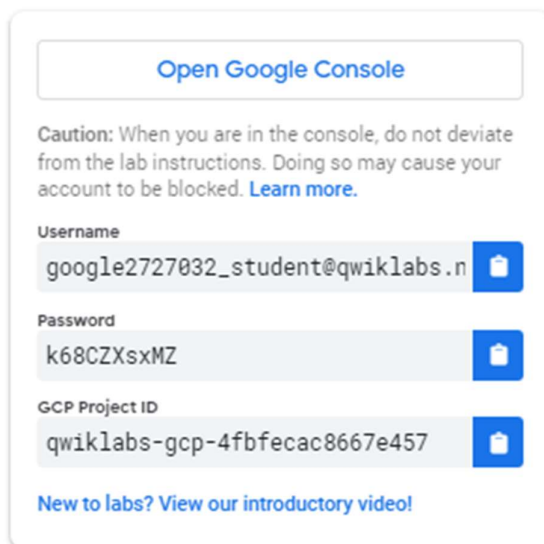
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



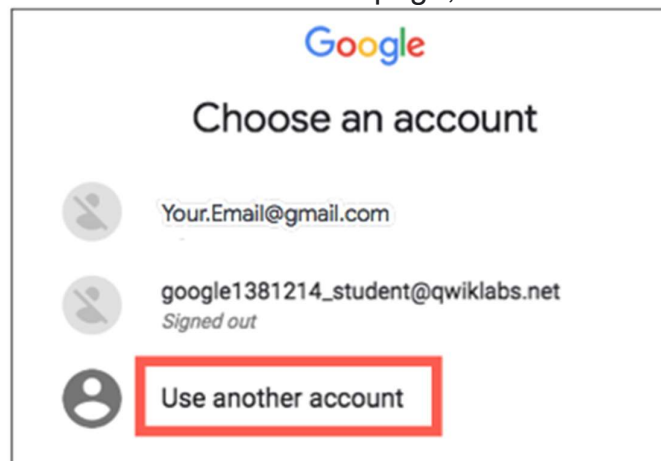
The screenshot shows a sign-in panel for the Google Cloud Console. At the top is a button labeled "Open Google Console". Below it is a caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)". The panel contains three input fields, each with a copy icon to its right: "Username" with the value "google2727032_student@qwiklabs.n", "Password" with the value "k68CZXsxMZ", and "GCP Project ID" with the value "qwiklabs-gcp-4fbfecac8667e457". At the bottom is a link: "New to labs? View our introductory video!"

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another**



Account.

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

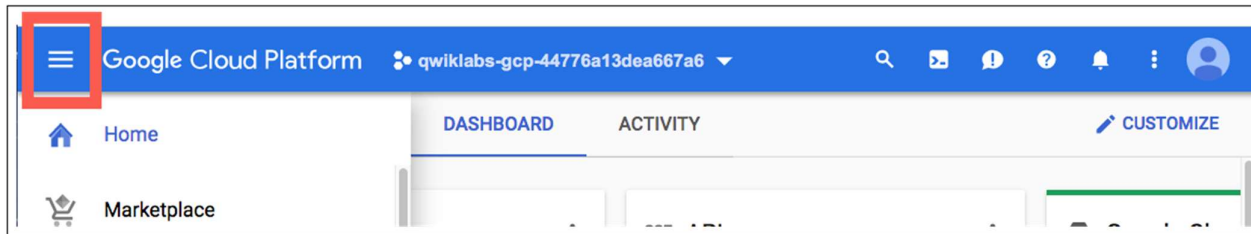
4. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-

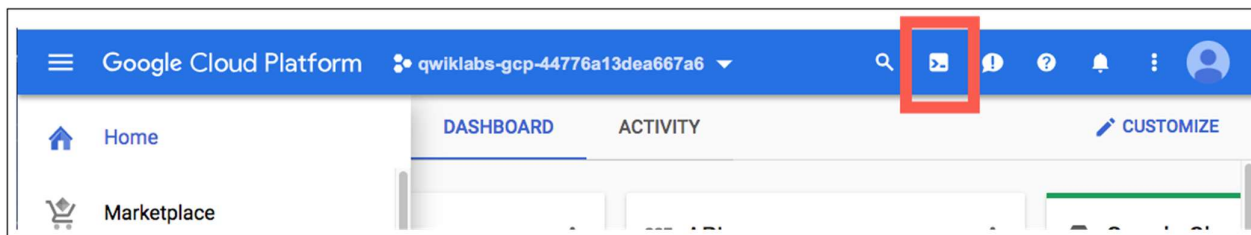
left.



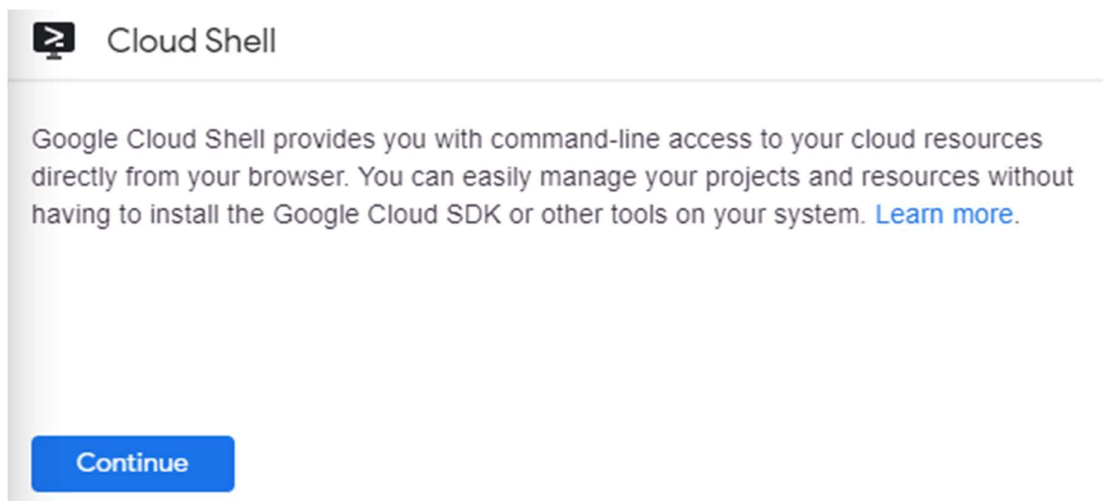
Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

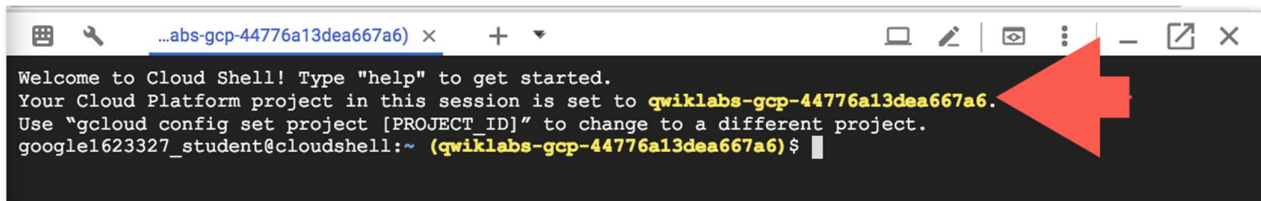
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



```
...abs-gcp-44776a13dea667a6) x + ▾
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6) $
```

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

You can list the active account name with this command:

```
gcloud auth list
content_copy
```

(Output)

```
Credentialed accounts:
- <myaccount>@<mydomain>.com (active)content_copy
```

(Example output)

```
Credentialed accounts:
- google1623327_student@qwiklabs.netcontent_copy
```

You can list the project ID with this command:

```
gcloud config list project
content_copy
```

(Output)

```
[core]
project = <project_ID>content_copy
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6content_copy
```

For full documentation of `gcloud` see the [gcloud command-line tool overview](#).

Review the Quiz application

In this section, you access Cloud Shell, clone the git repository containing the Quiz application, and run the application.

Clone source code in Cloud Shell

Clone the repository for the class.

```
git clone https://github.com/GoogleCloudPlatform/training-data-analystcontent_copy
```

Configure and run the Quiz application

1. Change the working directory:

```
cd ~/training-data-analyst/courses/developingapps/java/cloudstorage/startcontent_copy
```

2. Configure the application:

```
./prepare_environment.shcontent_copy
```

This script file:

- Creates an App Engine application
- Exports an environment variable `G_CLOUD_PROJECT`
- Runs `mvn clean install`
- Prints out the Project ID.

When the application has been configured, you'll see output similar to the following:

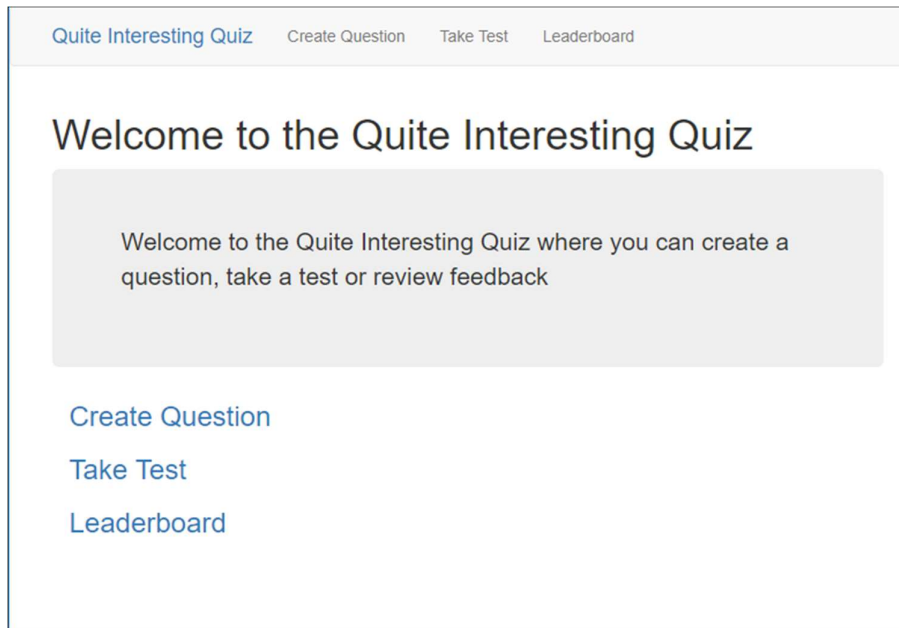
```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 35.702 s
[INFO] Finished at: 2018-05-20T21:22:33-04:00
[INFO] Final Memory: 35M/86M
[INFO] -----
Project ID: qwiklabs-gcp-a3b32951ac6cfb59content_copy
```

3. To run the application, execute the following command:

```
mvn spring-boot:runcontent_copy
```

Review the Quiz application

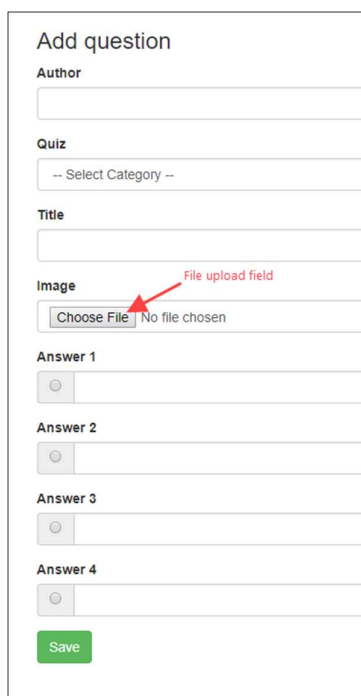
1. To view the application, click **Web preview > Preview on port 8080**.



2. Click **Create Question** in the toolbar.

You should see a simple form that contains textboxes for the question and answers and radio buttons to select the correct answer.

The form has a new file upload field that is used to upload an image or video file. In this lab you upload an image file; you would use the same process to upload a video file.

The screenshot shows a form titled 'Add question'. It contains several input fields: 'Author' (text), 'Quiz' (dropdown menu with '-- Select Category --'), 'Title' (text), and 'Image' (file upload field). The 'Image' field has a red arrow pointing to it with the label 'File upload field'. Below the 'Image' field, there are four radio button options labeled 'Answer 1', 'Answer 2', 'Answer 3', and 'Answer 4'. At the bottom of the form, there is a green 'Save' button.

Examining the Quiz application code

In this section, you use the Cloud Shell text editor to review the Quiz application code.

Launch the Cloud Shell text editor

1. From Cloud Shell, click **Launch the code editor** icon (looks like a pencil) to launch the code editor.



The code editor launches in a separate tab of your browser, along with Cloud Shell.

2. Navigate to `/training-data-analyst/courses/developingapps/java/cloudstorage/start` folder using the file browser panel on the left side of the editor.

Review the Spring Boot Web application

1. Select `new_question.html` file in the `.../src/main/resources/templates` folder.

This file contains the Thyme template for the Create Question form. Notice how the form uses `multipart/form-data` as the `enc-type`, and the two form controls for images and videos:

- A file upload control called `image`
- A hidden field called `imageUrl`

2. Select the `.../src/main/java/com/google/training/appdev` folder.

Java file paths are relative to this folder.

3. Select `.../web/QuestionsController.java`.

In this file the POST handler invokes a method on an image service.

4. Select the `.../services/gcp/cloudstorage/ImageService.java` file.

This is the file where you write code to save image file data into Cloud Storage.

Creating a Cloud Storage Bucket

In this section, you create a Cloud Storage bucket and export an environment variable that references it.

Create a Cloud Storage bucket

1. Return to Cloud Shell and stop the application by pressing **Ctrl+C**.
2. To create a Cloud Storage bucket named `<Project ID>-media`, execute the following command:

```
gsutil mb gs://$DEVSHELL_PROJECT_ID-mediacontent_copy
```

You can create a bucket using the `gsutil mb` command, passing through the name of the bucket as `gs://BUCKET_NAME`. You can use `$DEVSHELL_PROJECT_ID` as the bucket name prefix followed by `-media`.

Click **Check my progress** to verify the objective.

Create a storage bucket

Check my progress

3. To export the Cloud Storage bucket name as an environment variable named `G_CLOUD_BUCKET`, execute the following command:

```
export G_CLOUD_BUCKET=$DEVSHELL_PROJECT_ID-mediacontent_copy
```

Recall that the application makes use of environment variables for configuration. This allows the development team to deploy the application into development, test, staging, and production just by changing these variables.

Adding Objects to Cloud Storage

In this section, you write code to save uploaded files into Cloud Storage.

Important: Update code within the sections marked as follows: `// TODO //` `END TODO` To maximize your learning, review the code, inline comments, and related [Cloud Storage APIs & Reference documentation](#).

Import and use the Java Cloud Storage Package

1. In the code editor, move to the top of the `.../services/gcp/cloudstorage/ImageService.java` file.
2. Write a star import for the `com.google.cloud.storage.*` package.

```
3. // TODO: Write a start import for Cloud Storage
4.
5. import com.google.cloud.storage.*;
6.
7. // END TODO
content_copy
```

8. Construct a Cloud Storage client named `storage` using the `StorageOptions` class.

```
9. // TODO: Create the storage client
10. // The StorageOptions class has a getDefaultInstance()
11. // static method.
12. // Use the getService() method to get the storage client
13.
14. private static Storage storage = StorageOptions
15.     .getDefaultInstance()
16.     .getService();
17.
18. // END TODO
content_copy
```

19. Declare a String named `bucketName`, and annotate it with a Spring value annotation to retrieve the value from a property named `google.storage.bucket`.
20. Declare a constant named `bucket` to reference the Cloud Storage bucket.

```
// TODO: Get the name of the Cloud Storage bucket
// Use a Spring @Value annotation to get the value
// Get the value using ${google.storage.bucket}
// This references the GCLOUD_BUCKET environment variable

    @Value("${google.storage.bucket}")
    private String bucketName;

// END TODO
content_copy
```

ImageService.java

```

package com.google.training.appdev.services.gcp.cloudstorage;

// TODO: Write a start import for Cloud Storage

import com.google.cloud.storage.*;

// END TODO

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;

@Service
public class ImageService {

    // TODO: Create the storage client
    // The StorageOptions class has a getDefaultInstance()
    // static method.
    // Use the getService() method to get the storage client

    private static Storage storage = StorageOptions
        .getDefaultInstance()
        .getService();

    // END TODO

    // TODO: Get the name of the Cloud Storage bucket
    // Use a Spring @Value annotation to get the value
    // Get the value using ${google.storage.bucket}
    // This references the GLOUD_BUCKET environment variable

    @Value("${google.storage.bucket}")
    private String bucketName;

    // END TODO
content copy

```

Write code to send a file to Cloud Storage

Still updating `ImageService.java`, move to the `saveImage(MultipartFile file)` handler and then use the Cloud Storage client to upload a file to your Cloud Storage bucket and make it publicly available.

1. Declare a `BlobInfo` object and initialize it using the storage client object. Customize the `BlobInfo` object using its `Builder`. Use the option to set the Content Type and to set the ACL to allow unauthenticated Read access.

```

2. // TODO: Create a new Cloud Storage object
3. // Use the BlobInfo class to represent this object
4. // Use the BlobInfo.Builder to customize the Blob
5. // Set the content type from the file
6. // Set the object ACL to Public Read
7.
8. BlobInfo blobInfo = storage.create(
9.     BlobInfo.newBuilder(bucketName, fileName)
10.         .setContentType(file.getContentType())

```

```

11.         .setAcl(new ArrayList<>(
12.             Arrays.asList(Acl.of(Acl.User.ofAllUsers(),
13.                                   Acl.Role.READER))))
14.         .build(),
15.         file.getInputStream());
16.
17. // END TODO
content_copy

```

18. Return the public URL for the new Cloud Storage object as a string.

```

19. // TODO: Cloud Storage public URLs are in the form:
20. // https://storage.googleapis.com/[BUCKET]/[OBJECT]
21. // Use String concatenation to create return the URL
22.
23.     return "https://storage-download.googleapis.com/" + bucketName+ "/"
24.         +fileName;
25.
26. // END TODO
content_copy

```

ImageService.java

```

public String saveImage(MultipartFile file)
    throws IOException {
// The existing code in the method creates a unique name
// based on the file's original name. It has a
// prefix generated using the current date and time.
// This should ensure that a new file upload won't
// overwrite an existing object in the bucket

    String fileName = System.nanoTime() +
        file.getOriginalFilename();

// TODO: Create a new Cloud Storage object
// Use the BlobInfo class to represent this object
// Use the BlobInfo.Builder to customize the Blob
// Set the content type from the file
// Set the object ACL to Public Read

    BlobInfo blobInfo = storage.create(
        BlobInfo.newBuilder(bucketName, fileName)
            .setContentType(file.getContentType())
            .setAcl(new ArrayList<>(
                Arrays.asList(Acl.of(Acl.User.ofAllUsers(),
                                    Acl.Role.READER))))
            .build(),
        file.getInputStream());

// END TODO

// TODO: Cloud Storage public URLs are in the form:
// https://storage.googleapis.com/[BUCKET]/[OBJECT]
// Use String concatenation to return the URL

    return "https://storage-download.googleapis.com/" +
        bucketName + "/" + fileName;

// END TODO
}
content_copy

```

Run the application and create a Cloud Storage object

1. Save `.../services/gcp/cloudstorage/ImageService.java` and then return to the Cloud Shell.

2. Start the application by typing

```
3. mvn spring-boot:run
content_copy
```

4. Download an image file to your local machine from [here](#).

5. In Cloud Shell, click **Web preview** > **Preview on port 8080** to preview the Quiz application.

6. Click the **Create Question** link.

7. Complete the form with the following values, and then click **Save**.

Field	Value
Author	Your name
Quiz	Google Cloud
Title	Which product does this logo relate to?
Image	Upload the <code>Google_Cloud_Storage_logo.png</code> file you previously downloaded
Answer 1	App Engine
Answer 2	Cloud Storage (and select the Answer 2 radio button to indicate this as the correct answer)
Answer 3	Compute Engine
Answer 4	Container Engine

8. Click **Check my progress** to verify the objective.

9. Add Objects to Cloud Storage

10. Check my progress

11. Return to the Cloud Console and navigate to **Navigation menu** > **Storage**.

12. On the **Storage** > **Browser** page, click the correct bucket (named `<Project ID>-media`).

You should see your new object

named `#UniqueNumber#Google_Cloud_Storage_logo.png`.

Click **Check my progress** to verify the objective.

Run the client application and test the Cloud Storage public URL

1. Add `/api/quizzes/gcp` to the end of the Quiz application's URL.

You should see that the application returns JSON data to the client corresponding to the Question you added in the web application.

```
{ "questions": [{"quiz": "gcp", "author": "Mary Smith", "title": "Which product does  
this logo relate to?", "correctAnswer": 2, "imageUrl": "https://storage-  
download.googleapis.com/qwiklabs-gcp-33f2cf3e36ee1794-  
media/2293278773582Websites_and_Web_Apps_GCP.png", "image": null, "id": "5629499534213  
120", "answer1": "App Engine", "answer2": "Cloud Storage", "answer3": "Compute  
Engine", "answer4": "Container Engine"}]}
```

[content copy](#)

The `imageUrl` property should have a value corresponding to the new object you created in Cloud Storage.

2. Return to the application home page and click the **Take Test** link.
3. Click **GCP**, and answer each question.

When you get to the question you just added, you should see the image has been formatted inside the client-side web application!

Congratulations!

This concludes the self-paced lab, App Dev: Storing Image and Video Files in Cloud Storage - Java. You configured a quiz application to use Cloud Storage to store and retrieve application data.

Finish your Quest



This lab is part of the [Application Development - Java](#) and [Cloud Development](#) Quests. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, 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 this Quest and get immediate completion credit if you've taken this lab. See other available [Qwiklabs Quests](#).

Next steps / learn more

- For more information about application servers, see [Using the Local Development Server](#).
- Check out [Building Scalable and Resilient Web Applications on Google Cloud](#).

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. [Our classes](#) include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. [Certifications](#) help you validate and prove your skill and expertise in Google Cloud technologies.

Manual last updated April 30, 2020

Lab last tested April 30, 2020

Copyright 2021 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.