

# Data Loss Prevention: Qwik Start - JSON

**GSP107**



The Data Loss Prevention API provides programmatic access to a powerful detection engine for personally identifiable information (PII) and other privacy-sensitive data in unstructured data streams.

The DLP API provides fast, scalable classification and optional redaction for sensitive data elements like credit card numbers, names, social security numbers, passport numbers, and phone numbers. The API supports text and images – just send data to the API or specify data stored on your Cloud Storage, BigQuery, and Cloud Datastore instances.

In this lab, you will set up a JSON file with to analyze, send it to the Data Loss Prevention API, to inspect a string of data for sensitive information, then redact any sensitive information that was found.

# 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 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:

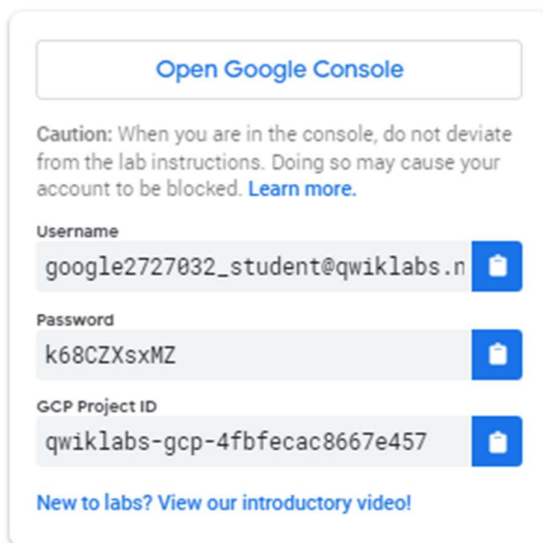
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

**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

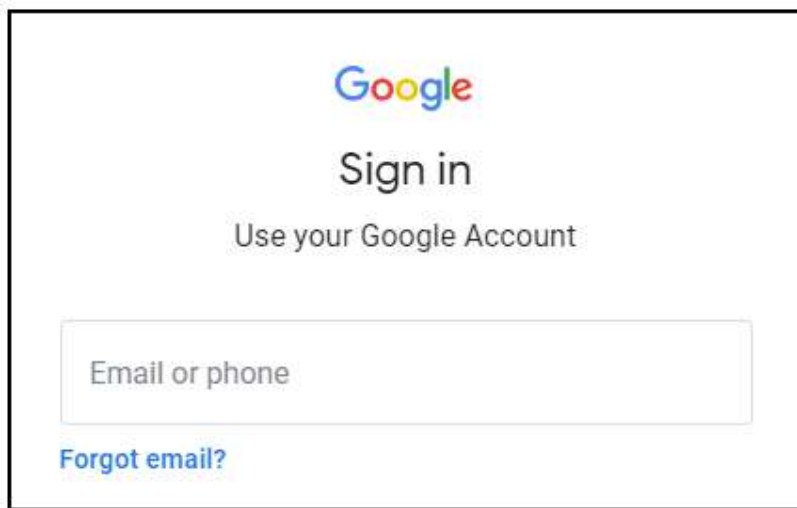
1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



The screenshot shows a sign-in panel with the following elements:

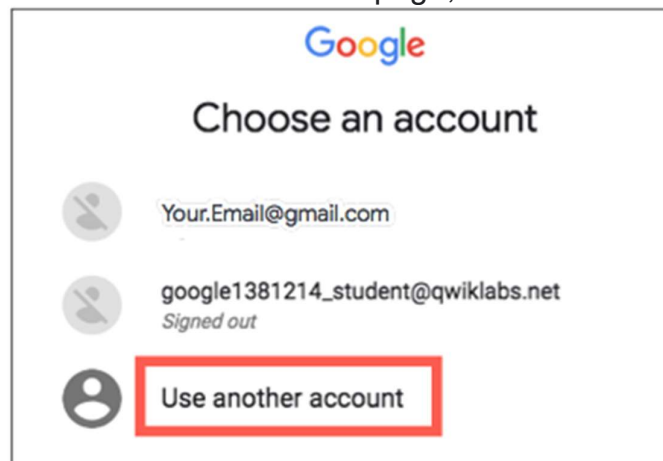
- A button at the top labeled "Open Google Console".
- A caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)"
- Three input fields, each with a copy icon to its right:
  - Username:** google2727032\_student@qwiklabs.n
  - Password:** k68CZXsxMZ
  - GCP Project ID:** qwiklabs-gcp-4fbfecac8667e457
- A link at the bottom: "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).

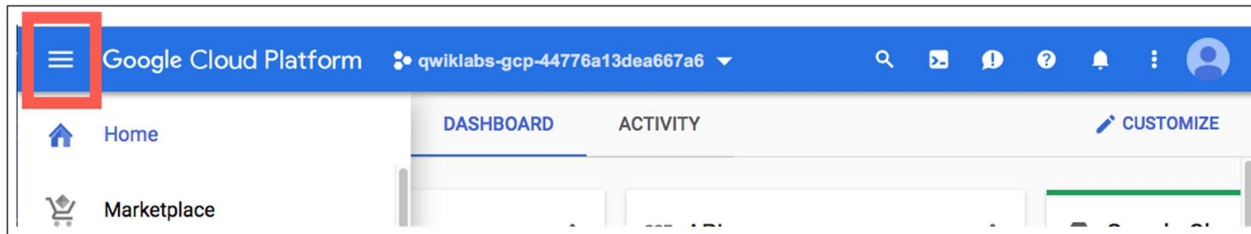
4. Click through the subsequent pages:

- 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-

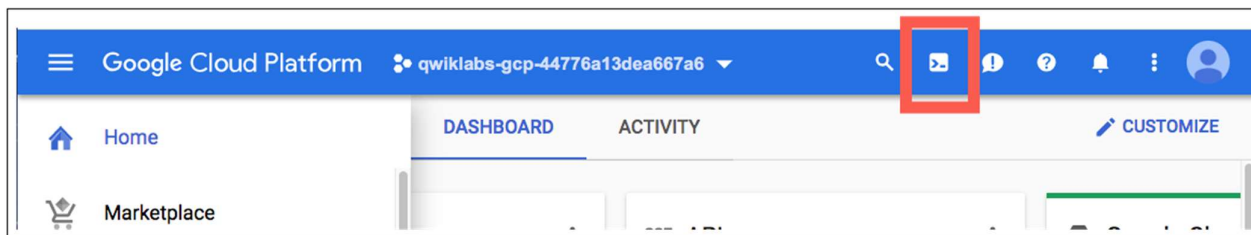
left.



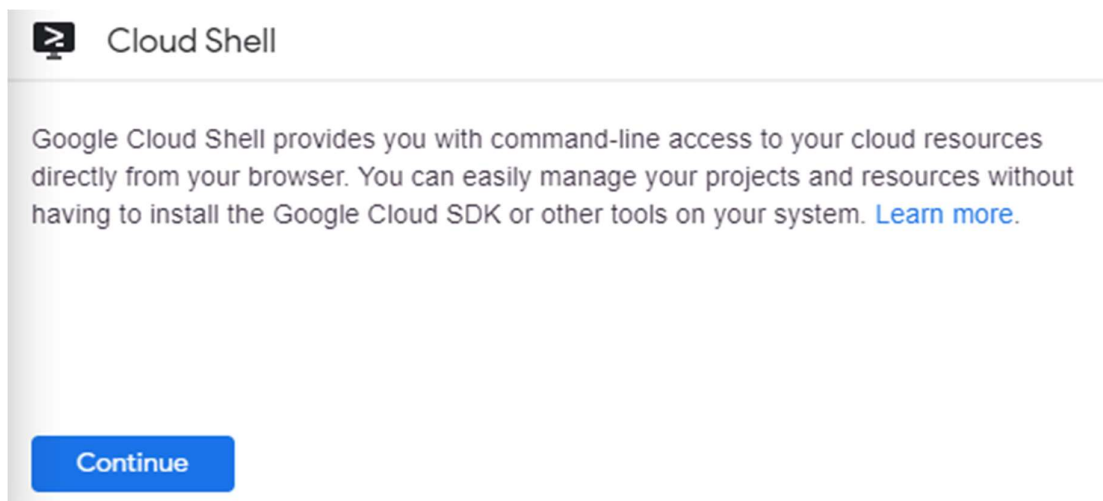
## Activate Cloud Shell

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.

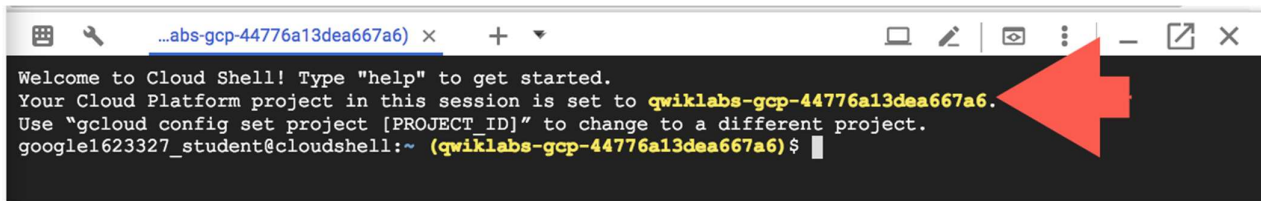
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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
```

(Output)

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

(Example output)

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

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]
project = <project ID>
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

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

# Create a service account key

Set an environment variable for your project ID:

```
export PROJECT_ID=$DEVSHHELL_PROJECT_ID
```

Create a Service Account to access the Google Cloud APIs when testing locally:

```
gcloud iam service-accounts create qwiklab \
  --display-name "Qwiklab Service Account"
```

Give your newly created Service Account appropriate permissions:

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member serviceAccount:qwiklab@${PROJECT_ID}.iam.gserviceaccount.com \
  --role roles/owner
```

After creating your Service Account, create a Service Account key:

```
gcloud iam service-accounts keys create ~/key.json \
  --iam-account qwiklab@${PROJECT_ID}.iam.gserviceaccount.com
```

This command generates a service account key stored in a JSON file named *key.json* in your home directory.

## Inspect a string for sensitive information

This section shows you how to ask the service to scan sample text using the [projects.content.inspect](#) REST method. The JSON file you will make contains an [InspectConfig](#) and a [ContentItem](#) object.

Using your preferred editor (*nano*, *vim*, etc.) or Cloud Shell, create a JSON request file with the following text, and save it as `inspect-request.json`.

```
{
  "item":{
    "value":"My phone number is (206) 555-0123."
  },
  "inspectConfig":{
    "infoTypes":[
      {
        "name":"PHONE_NUMBER"
      },
      {
        "name":"US_TOLLFREE_PHONE_NUMBER"
      }
    ]
  }
}
```

```
    "minLikelihood": "POSSIBLE",
    "limits": {
      "maxFindingsPerItem": 0
    },
    "includeQuote": true
  }
}
```

In Cloud Shell, authenticate to your service account, passing the location of your service account key file:

```
gcloud auth activate-service-account --key-file=key.json
```

Obtain an authorization token using your service account:

```
gcloud auth print-access-token
```

A huge string will get returned. You will need this token in the next step.

Use `curl` to make a `content:inspect` request, replacing `ACCESS_TOKEN` with the string that was returned in the previous step and `PROJECT_ID` with your Qwiklabs project ID:

```
curl -s \
-H "Authorization: Bearer ACCESS_TOKEN" \
-H "Content-Type: application/json" \
https://dlp.googleapis.com/v2/projects/$PROJECT_ID/content:inspect \
-d @inspect-request.json
```

**Here's what's going on:** To pass a filename to `curl` you use the `-d` option (for "data") and precede the filename with an `@` sign. This file should be in the same directory in which you execute the `curl` command.

You should see a response similar to the following:

```
{
  "result": {
    "findings": [
      {
        "quote": "(206) 555-0123",
        "infoType": {
          "name": "PHONE_NUMBER"
        },
        "likelihood": "LIKELY",
        "location": {
          "byteRange": {
            "start": "19",
            "end": "33"
          },
          "codepointRange": {
            "start": "19",
            "end": "33"
          }
        },
        "createTime": "2018-07-03T02:20:26.043Z"
      }
    ]
  }
}
```

# Redacting sensitive data from text content

The Data Loss Prevention API can automatically redact sensitive data from text files instead of giving you a list of findings.

Try sending the API JSON file using [deidentifyConfig](#) object, so sensitive information is redacted from the output.

Create a new JSON file (called new-inspect-file) that includes the following:

```
{
  "item": {
    "value": "My email is test@gmail.com",
  },
  "deidentifyConfig": {
    "infoTypeTransformations": {
      "transformations": [
        {
          "primitiveTransformation": {
            "replaceWithInfoTypeConfig": {}
          }
        }
      ]
    }
  },
  "inspectConfig": {
    "infoTypes": {
      "name": "EMAIL_ADDRESS"
    }
  }
}
```

Use `curl` to make a `content:deidentify` request, replacing `ACCESS_TOKEN` with the string that was returned earlier and `PROJECT_ID` with your Qwiklabs project ID. Also, replace the filename with the JSON file you configured in the previous step:

```
curl -s \
-H "Authorization: Bearer ACCESS_TOKEN" \
-H "Content-Type: application/json" \
https://dlp.googleapis.com/v2/projects/$PROJECT_ID/content:deidentify \
-d @new-inspect-file.json
```

Sample Output:

```
{
  "item": {
    "value": "My email is [EMAIL_ADDRESS]"
  },
  "overview": {
    "transformedBytes": "14",
    "transformationSummaries": [
      {
        "infoType": {
          "name": "EMAIL_ADDRESS"
        },
        "transformation": {
          "replaceWithInfoTypeConfig": {}
        },
        "results": [
          {
            "count": "1",

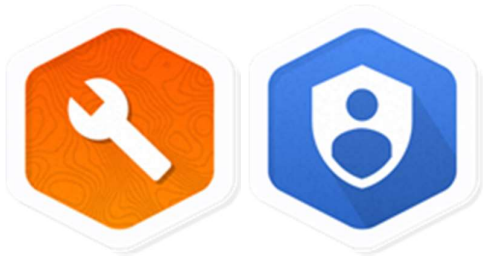
```



```
      "code": "SUCCESS"
    }
  ],
  "transformedBytes": "14"
}
]
```

You've sent your first request to the Data Loss Prevention API and redacted sensitive information from output!

# Congratulations!



## Finish Your Quest

This self-paced lab is part of the [Baseline: Deploy & Develop](#) and [Security & Identity Fundamentals](#). 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 a Quest and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).

## Take Your Next Lab

Continue your Quest with these suggestions:

- [Cloud Function: Qwik Start - Command Line](#)
- [Getting Started with Cloud KMS](#)
- [Setting up a Private Kubernetes Cluster](#)
- [Video Intelligence: Qwik Start](#)

## Next Steps / Learn More

This lab is also part of a series of labs called Qwik Starts. These labs are designed to give you a little taste of the many features available with Google Cloud. Search for "Qwik Starts" in the [lab catalog](#) to find the next lab you'd like to take!

## 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 May 16, 2019

Lab Last Tested September 27, 2018

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.