Redacting Sensitive Data with the DLP API

### **Objectives**

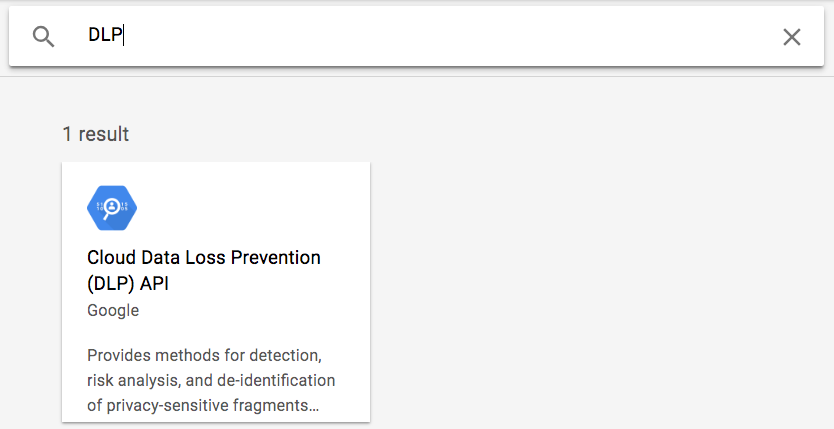
In this lab, you will learn how to do the following:

* Enable the DLP API.
* Install the Node JS DLP API and sample.
* Inspect string data for sensitive data.
* Redact sensitive data from string data and images.

## Task 1. Enable the DLP API

In this task, you enable the DLP API through APIs & Services.

1. Go to **Navigation menu > APIs & Services**.
2. Click the **Enable APIs and Services** button.
3. In the **Search for APIs & Services** field, enter DLP and then click on the **Cloud Data Loss Prevention (DLP) API** title.



1. Click the **Enable** button to enable the DLP API. If the API is already enabled, you will see a **Manage** button instead, along with an API enabled message. In that case you do not need to do anything.

## Task 2. Install the DLP API and Node JS samples

In this task, you download the Node JS DLP API and samples and install the required dependencies.

1. On the Google Cloud Console tile bar, click **Activate Cloud Shell** (Activate Cloud Shell icon) to open Cloud Shell. When prompted, click **Continue**. After a moment, in the lower part of the browser window, the Terminal appears.
2. Run the following command to create the GCLOUD\_PROJECT environment variable and set it to the project ID:

export GCLOUD\_PROJECT=$DEVSHELL\_PROJECT\_ID

Copied!

content\_copy

1. Run the following command in Cloud Shell to download the Node JS DLP API and samples:

git clone https://github.com/googleapis/nodejs-dlp.git

Copied!

content\_copy

1. Once the download is complete, change into the **nodejs-dlp/samples** directory:

cd nodejs-dlp/samples

Copied!

content\_copy

There are several Node JS DLP sample programs in this folder. Before you run them, you need to install the dependencies.

1. Run the following command to install the required dependencies:

npm install @google-cloud/dlp

npm install yargs

npm install mime

Copied!

content\_copy

## Task 3. Inspect and redact sensitive data

In this task, you inspect and mask sensitive information from the string also use the DLP API to redact sensitive data from an image.

### **Inspect a string for sensitive information**

1. In Cloud Shell run the command below. (If prompted, click **Authorize**.)

node inspectString.js $GCLOUD\_PROJECT "My email address is joe@example.com."

Copied!

content\_copy

You should receive the following output.

**Output:**

Findings:

Info type: PERSON\_NAME

Likelihood: POSSIBLE

Info type: EMAIL\_ADDRESS

Likelihood: LIKELY

The result shows what sensitive data was found, what type of information it is, and how certain the API is about that info type.

1. In Cloud Shell, run the following command:

node inspectString.js $GCLOUD\_PROJECT "My phone number is 555-555-5555."

Copied!

content\_copy

You should receive the following output.

**Output:**

Findings:

Info type: PHONE\_NUMBER

Likelihood: VERY\_LIKELY

1. Feel free to experiment with different input to the **inspectString.js** program. For example, try passing in values like 1234-5678-9876-5432 or 123-45-6789.

### **Mask sensitive information from a string**

* In Cloud Shell, run the following command:

node deidentifyWithMask.js $GCLOUD\_PROJECT "My phone number is 555-555-5555."

Copied!

content\_copy

You should receive the following output.

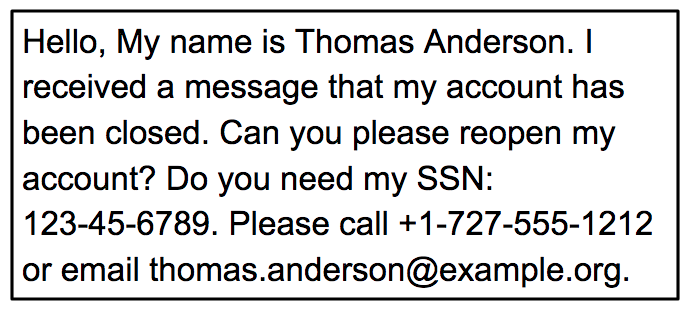
**Output:**

My phone number is \*\*\*\*\*\*\*\*\*\*\*\*.

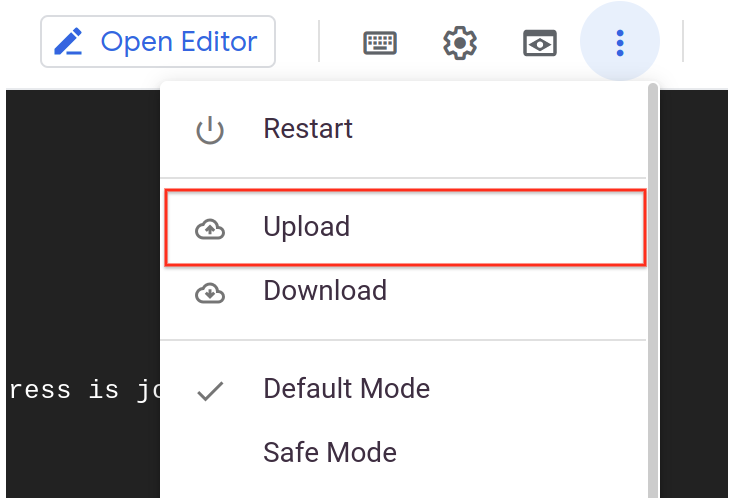
### **Redact sensitive data from images**

You will now use the DLP API to redact sensitive data from an image.

1. Right-click on the image below and select **Save image as**. Save it locally on your computer as dlp-input.png.



1. In the bar above the terminal, click the button at the top right with three vertical dots and select **Upload**.



If **Upload** is not clickable ("grayed out"), then click **Restart**. After the Cloud Shell environment is restarted, the **Upload** Link should be active.

Execute these commands commands before starting the next step in the lab.

export GCLOUD\_PROJECT=$DEVSHELL\_PROJECT\_ID

cd nodejs-dlp/samples

Copied!

content\_copy

1. Click **Choose Files**, select the downloaded **dlp-input.png** image file, and **Upload** it to Cloud Shell.
2. From **Cloud Shell**, click **Open Editor** . This will launch the Cloud Shell code editor, which includes a file browser.
3. In the Cloud Shell code editor, on the left, you should see the dlp-input.png file.
4. Click the **dlp-input.png** file to display the image and verify it was uploaded.
5. From **Cloud Shell**, click **Open Terminal** to return to the terminal window.
6. In the terminal, run the following command to redact the email address values from the image:

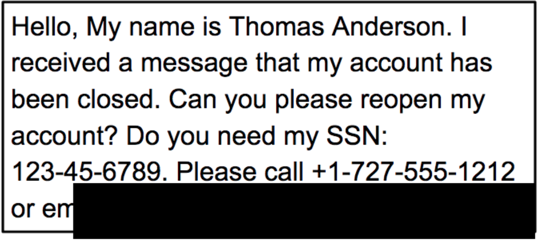
node redactImage.js $GCLOUD\_PROJECT ~/dlp-input.png "" EMAIL\_ADDRESS ~/dlp-redacted.png

Copied!

content\_copy

1. Open **Editor**.
2. In the Cloud Shell code editor, on the left, click the **dlp-redacted.png** file.

You will see the image with the domain name redacted.



When calling the redact API, you specified EMAIL\_ADDRESS as the infotype to redact. In the image, you should notice that the email address is no longer visible.