Google Cloud Speech API: Qwik Start

GSP119

Overview

The Google Cloud Speech API enables easy integration of Google speech recognition technologies into developer applications. The Speech API allows you to send audio and receive a text transcription from the service (see What is the Google Cloud Speech API? for more information).

What you'll do

- Create an API key

- Create a Speech API request

- Call the Speech API request

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).
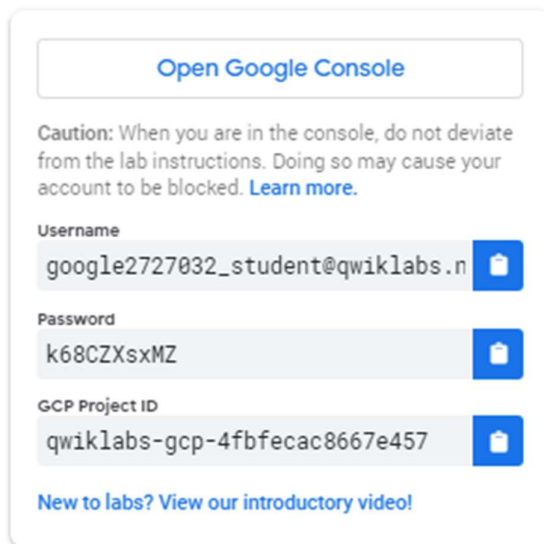- Time to complete the lab.
Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

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

Now that you've started your lab, you'll log in to the Google Cloud Shell console, then launch the command line tool.

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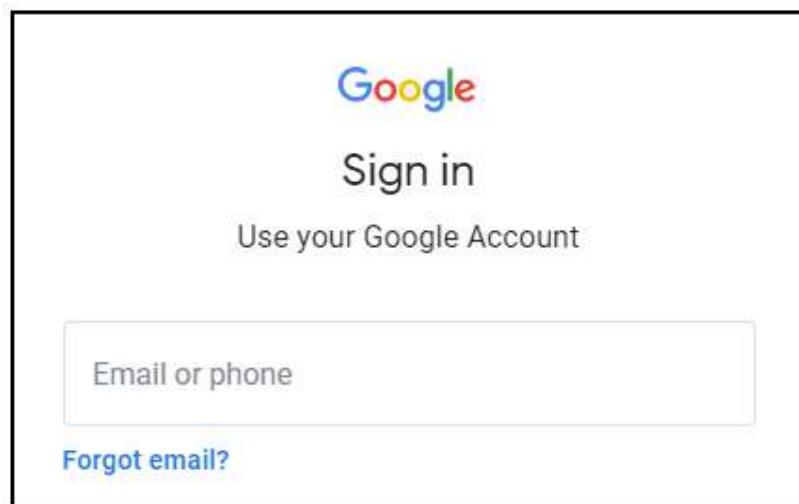
2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.

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-
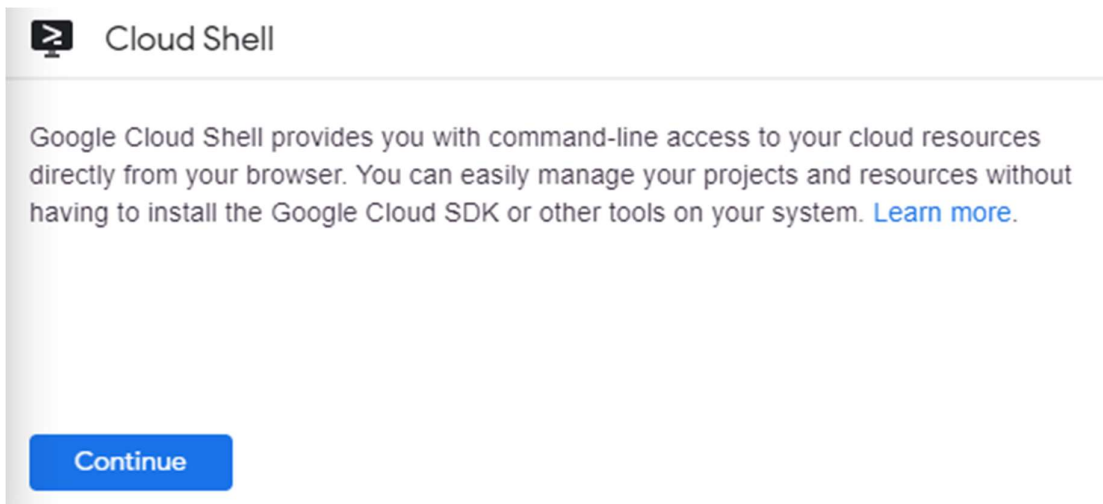left.

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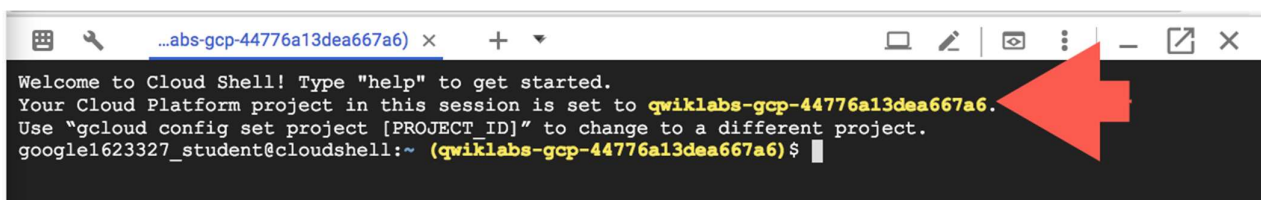
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.
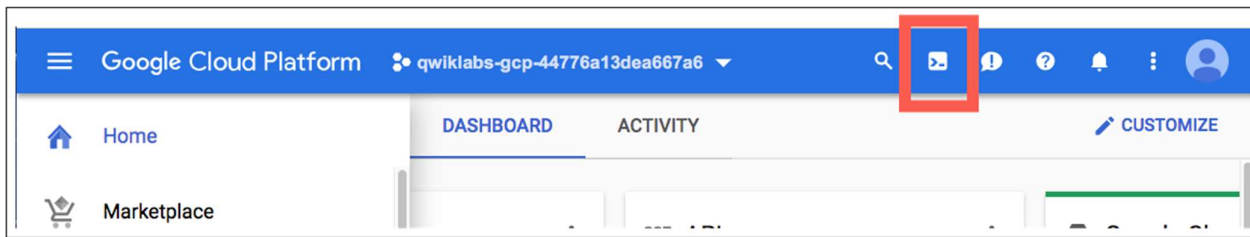
Click **Continue**.

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:

`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>
```
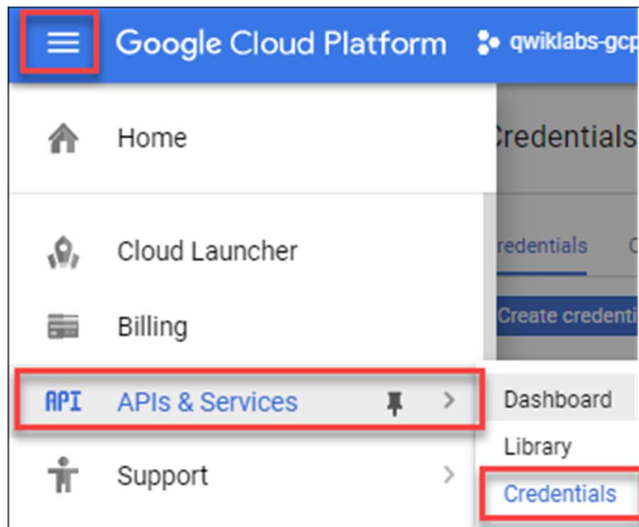
(Example output)

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

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

Create an API Key

Since you'll be using `curl` to send a request to the Speech API, you'll need to generate an API key to pass in our request URL.
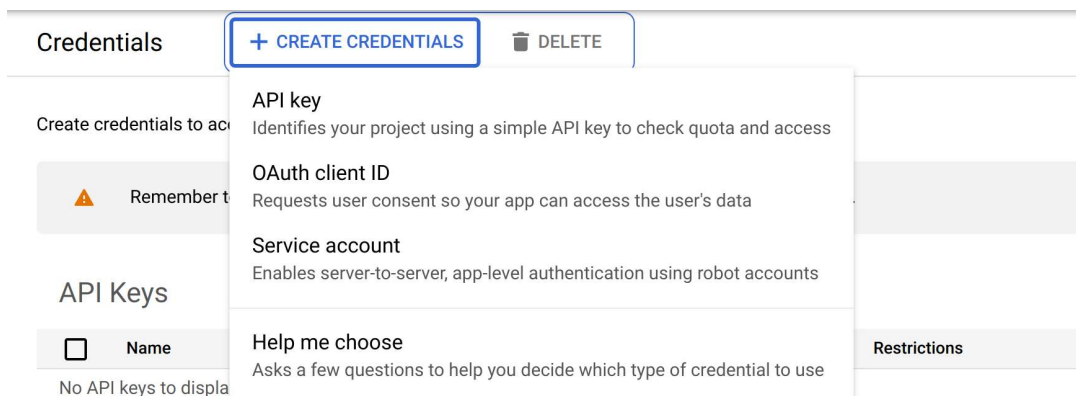
To create an API key, click **Navigation menu** > **APIs & services** > **Credentials**:

Then click **Create credentials**:

In the drop down menu, select **API key**:

Copy the key you just generated and click **Close**.

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

Create an API Key

Now that you have an API key, you will save it as an environment variable to avoid having to insert the value of your API key in each request.

In order to perform next steps please connect to the instance provisioned for you via ssh. Open the **Navigation menu** and select **Compute Engine**. You should see the following provisioned linux instance:
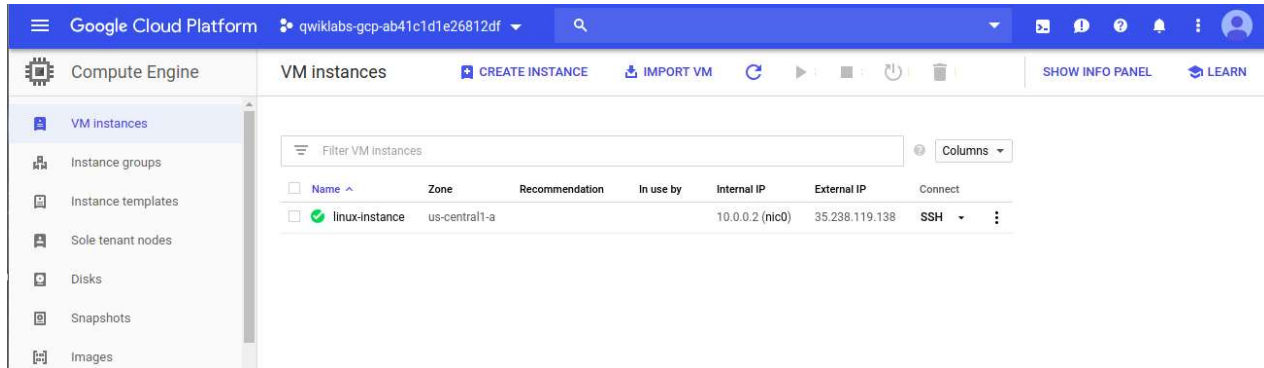
Click on the **SSH** button. You will be brought to an interactive shell. In the command line, enter in the following, replacing `<YOUR_API_KEY>` with the key you just copied:

```
export API_KEY=<YOUR_API_KEY>
```

Remain in this SSH session for the rest of the lab.

Create your Speech API request

Note: You will use a pre-recorded file that's available on Cloud Storage: `gs://cloud-samples-tests/speech/brooklyn.flac`. You can listen to this file before sending it to the Speech API [here](#).

Create `request.json` in SSH command line. You'll use this to build your request to the speech API:.

```
touch request.json
```

Now open the `request.json` using your preferred command line editor (`nano`, `vim`, `emacs`) or `gcloud`. Add the following to your `request.json` file, using the `uri` value of the sample raw audio file:

```
{
  "config": {
      "encoding":"FLAC",
      "languageCode": "en-US"
  },
  "audio": {
      "uri":"gs://cloud-samples-tests/speech/brooklyn.flac"
  }
}
```

The request body has a `config` and `audio` object.

In `config`, you tell the Speech API how to process the request:

- The `encoding` parameter tells the API which type of audio encoding you're using while the file is being sent to the API. `FLAC` is the encoding type for .raw files (here is [documentation](#) for encoding types for more details).
  There are other parameters you can add to your `config` object, but `encoding` is the only required one.

In the `audio` object, you pass the API the uri of the audio file in Cloud Storage.

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

Create your Speech API request

Check my progress

Now you're ready to call the Speech API!

Call the Speech API

Pass your request body, along with the API key environment variable, to the Speech API with the following `curl` command (all in one single command line):

```
curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json \
"https://speech.googleapis.com/v1/speech:recognize?key=${API_KEY}"
```

Your response should look something like this:

```
{
  "results": [
    {
      "alternatives": [
        {
          "transcript": "how old is the Brooklyn Bridge",
          "confidence": 0.98267895
        }
      ]
    }
  ]
}
```

The `transcript` value will return the Speech API's text transcription of your audio file, and the `confidence` value indicates how sure the API is that it has accurately transcribed your audio.

You'll notice that you called the `syncrecognize` method in the request above. The Speech API supports both synchronous and asynchronous speech to text transcription. In this example you sent it a complete audio file, but you can also use the `syncrecognize` method to perform streaming speech to text transcription while the user is still speaking.

You created an Speech API request then called the Speech API. Run the following command to save the response in a `result.json` file:

```
curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json \
"https://speech.googleapis.com/v1/speech:recognize?key=${API_KEY}" > result.json
```
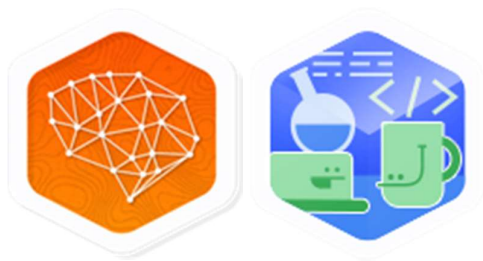
Click **Check my progress** to verify the objective.

Call the Speech API

Check my progress

Congratulations!

This concludes the self-paced lab, Google Cloud Speech API: Qwik Start. You integrated speech recognition into an app, and then generated transcription from the service.

Finish Your Quest

Continue with your [Baseline: Data, ML, AI](#) or [Intro to ML: Language Processing](#) Quest. A Quest is a series of related labs that form a learning path. Completing a 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 either of the above Quests and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).

Take Your Next Lab

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!

Next Steps /Learn More

- Check out our [Quests](#) for a series of labs to concentrate on one area. For example [Networking in the Google Cloud](#).
- Or, learn about something completely different with [Google Maps Web Services Proxy for Mobile Applications](#).

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 January 13, 2021

Lab Last Tested January 13, 2021