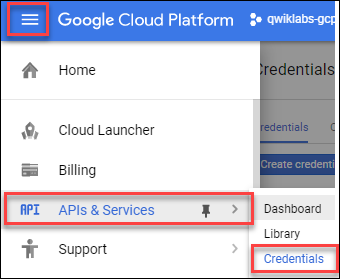
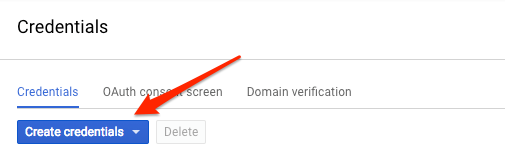
**Create an API Key**

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.

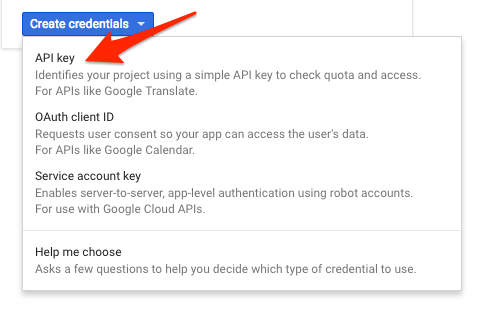
To create an API key, click **Navigation menu** > **APIs & services** > **Credentials**:



Then click **Create credentials**:



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



Copy the key you just generated.

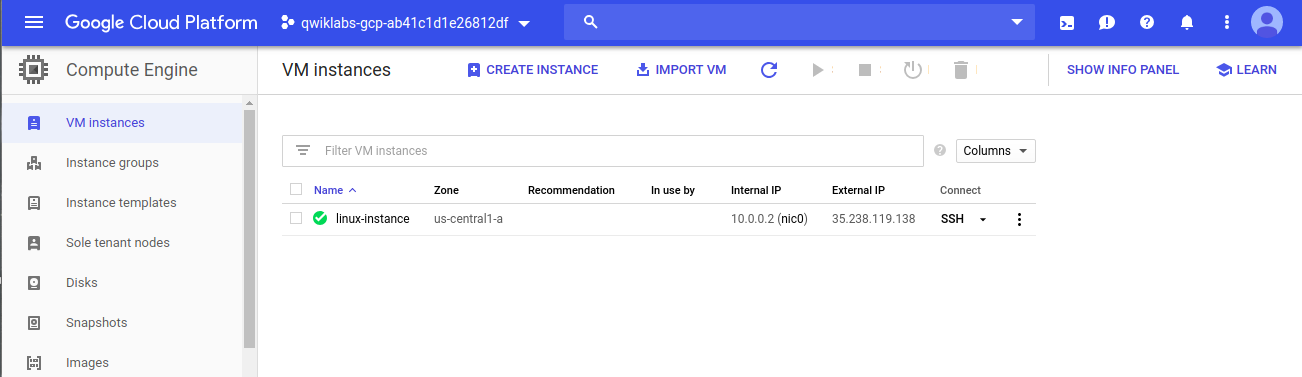
Click **Check my progress** to verify the objective.

Create an API Key

Check my progress

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](https://storage.cloud.google.com/cloud-samples-tests/speech/brooklyn.flac).

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](https://cloud.google.com/speech/reference/rest/v1/RecognitionConfig) 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](https://google.qwiklabs.com/quests/34) or [Intro to ML: Language Processing](https://google.qwiklabs.com/quests/82) 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](http://google.qwiklabs.com/catalog).

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](https://google.qwiklabs.com/catalog) to find the next lab you'd like to take!