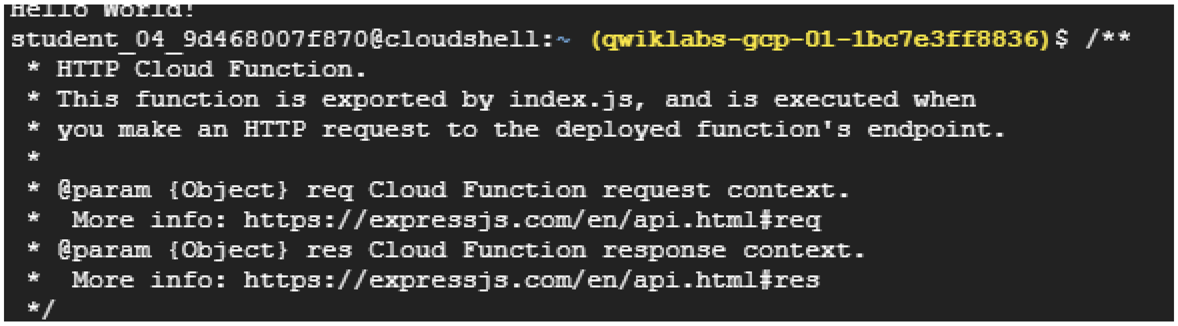
API Gateway: Qwik Start

ask 1. Deploying an API backend

API Gateway sits in front of a deployed backend service and handles all incoming requests. In this lab, API Gateway routes incoming calls to a Cloud Function backend named **helloGET** that contains the function shown below:

/**

- HTTP Cloud Function.
- This function is exported by index.js, and is executed when
- you make an HTTP request to the deployed function's endpoint.
-
- @param {Object} req Cloud Function request context.
- More info: https://expressjs.com/en/api.html#req
- @param {Object} res Cloud Function response context.
- More info: https://expressjs.com/en/api.html#res

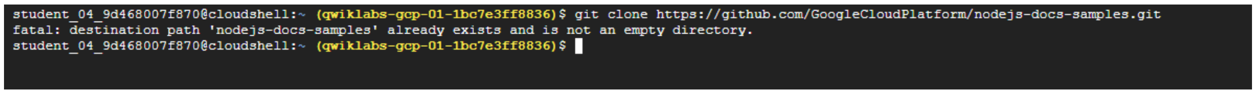


```
Hello World!
student_04_9d468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ /**
 * HTTP Cloud Function.
 * This function is exported by index.js, and is executed when
 * you make an HTTP request to the deployed function's endpoint.
 *
 * @param {Object} req Cloud Function request context.
 *  More info: https://expressjs.com/en/api.html#req
 * @param {Object} res Cloud Function response context.
 *  More info: https://expressjs.com/en/api.html#res
 */
```

*/ exports.helloGET = (req, res) => { res.send('Hello World!'); };

1. In Cloud Console, clone the Cloud Function sample repository:

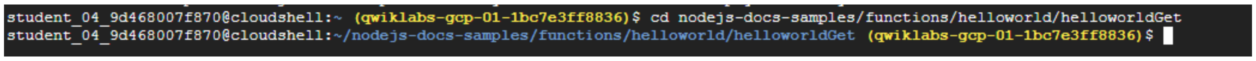
git clone https://github.com/GoogleCloudPlatform/nodejs-docs-samples.git



```
student_04_9d468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ git clone https://github.com/GoogleCloudPlatform/nodejs-docs-samples.git
fatal: destination path 'nodejs-docs-samples' already exists and is not an empty directory.
student_04_9d468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ 
```

2. Change to the directory that contains the Cloud Functions sample code:

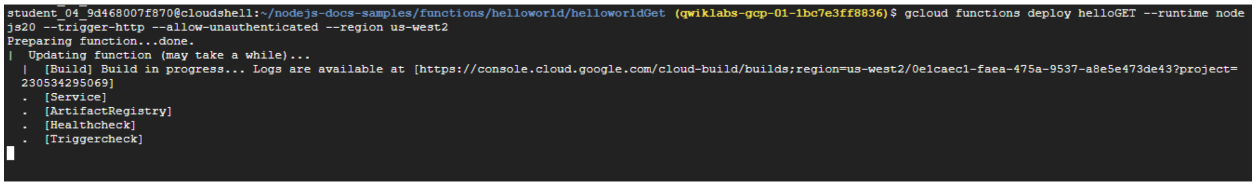
cd nodejs-docs-samples/functions/helloworld/helloworldGet



```
student_04_9d468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ cd nodejs-docs-samples/functions/helloworld/helloworldGet
student_04_9d468007f870@cloudshell:~/nodejs-docs-samples/functions/helloworld/helloworldGet (qwiklabs-gcp-01-1bc7e3ff8836)$ 
```

3. To deploy the function with an HTTP trigger, run the following command in the directory containing your function:

gcloud functions deploy helloGET --runtime nodejs20 --trigger-http --allow-unauthenticated --region us-west2



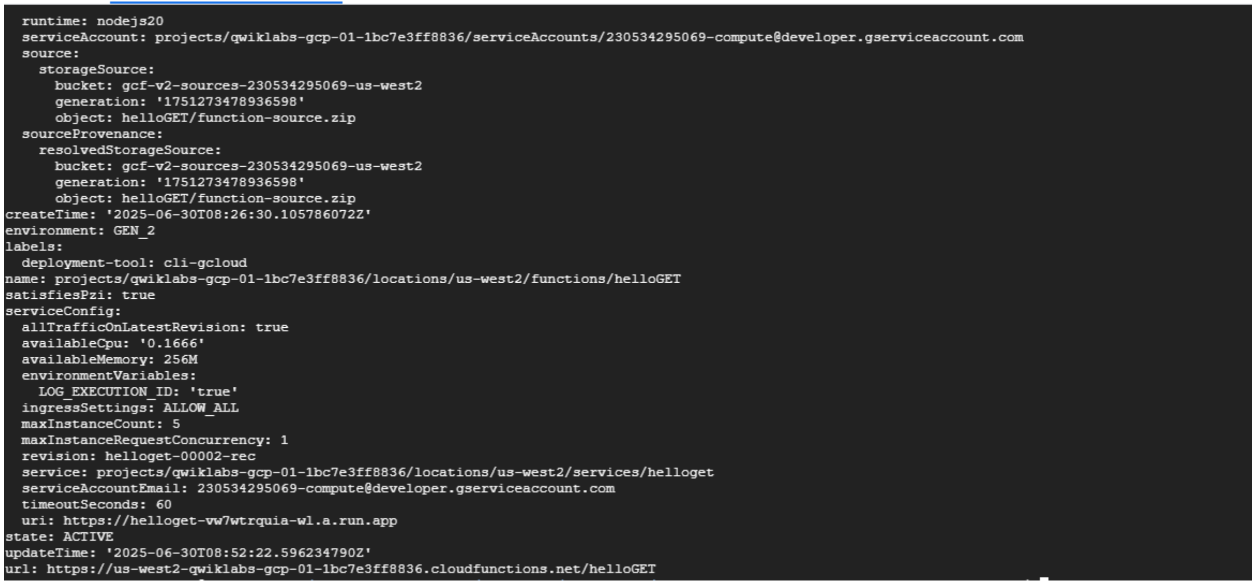
```
student_04_9d468007f870@cloudshell:~/nodejs-docs-samples/functions/helloworld/helloworldGet (qwiklabs-gcp-01-1bc7e3ff8836)$ gcloud functions deploy helloGET --runtime node
js20 --trigger-http --allow-unauthenticated --region us-west2
Preparing function...done.
|  Updating function (may take a while)...
  |  [Build] Build in progress... Logs are available at [https://console.cloud.google.com/cloud-build/builds;region=us-west2/0e1caec1-faea-475a-9537-a8e5e473de43?project=
  230534295069]
  .  [Service]
  .  [ArtifactRegistry]
  .  [Healthcheck]
  .  [Triggercheck]
```

- If prompted, enter **Y** to enable the API.

Task 2. Test the API backend

1. When the function finishes deploying, take note of the httpsTrigger's url property or find it using the following command:

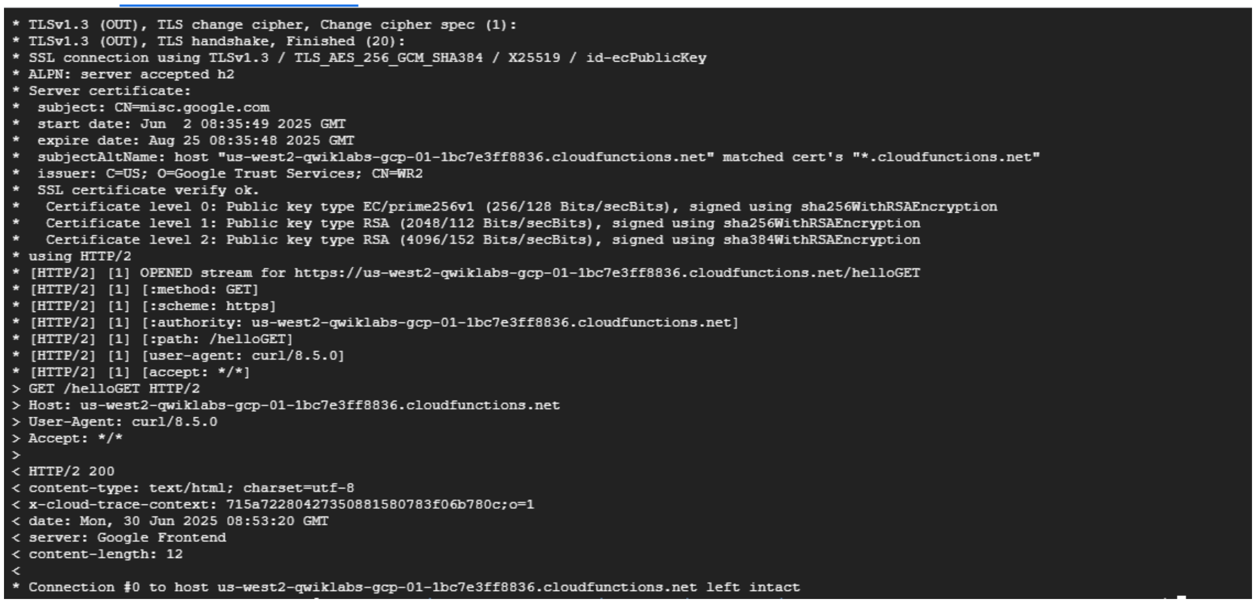
   gcloud functions describe helloGET --region us-west2



```
  runtime: nodejs20
  serviceAccount: projects/qwiklabs-gcp-01-1bc7e3ff8836/serviceAccounts/230534295069-compute@developer.gserviceaccount.com
  source:
    storageSource:
      bucket: gcf-v2-sources-230534295069-us-west2
      generation: '1751273478936598'
      object: helloGET/function-source.zip
  sourceProvenance:
    resolvedStorageSource:
      bucket: gcf-v2-sources-230534295069-us-west2
      generation: '1751273478936598'
      object: helloGET/function-source.zip
createTime: '2025-06-30T08:26:30.105786072Z'
environment: GEN_2
labels:
  deployment-tool: cli-gcloud
name: projects/qwiklabs-gcp-01-1bc7e3ff8836/locations/us-west2/functions/helloGET
satisfiesPzi: true
serviceConfig:
  allTrafficOnLatestRevision: true
  availableCpu: '0.1666'
  availableMemory: 256M
  environmentVariables:
    LOG_EXECUTION_ID: 'true'
  ingressSettings: ALLOW_ALL
  maxInstanceCount: 5
  maxInstanceRequestConcurrency: 1
  revision: helloget-00002-rec
  service: projects/qwiklabs-gcp-01-1bc7e3ff8836/locations/us-west2/services/helloget
  serviceAccountEmail: 230534295069-compute@developer.gserviceaccount.com
  timeoutSeconds: 60
  uri: https://helloget-vw7wtrquia-wl.a.run.app
state: ACTIVE
updateTime: '2025-06-30T08:52:22.596234790Z'
url: https://us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net/helloGET
```

2. Set your PROJECT_ID as a variable:

   export PROJECT_ID=qwiklabs-gcp-01-1bc7e3ff8836



```
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / X25519 / id-ecPublicKey
* ALPN: server accepted h2
* Server certificate:
*  subject: CN=misc.google.com
*  start date: Jun  2 08:35:49 2025 GMT
*  expire date: Aug 25 08:35:48 2025 GMT
*  subjectAltName: host "us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net" matched cert's "*.cloudfunctions.net"
*  issuer: C=US; O=Google Trust Services; CN=WR2
*  SSL certificate verify ok.
*   Certificate level 0: Public key type EC/prime256v1 (256/128 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 2: Public key type RSA (4096/152 Bits/secBits), signed using sha384WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net/helloGET
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net]
* [HTTP/2] [1] [:path: /helloGET]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> GET /helloGET HTTP/2
> Host: us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/2 200
< content-type: text/html; charset=utf-8
< x-cloud-trace-context: 715a72280427350881580783f06b780c;o=1
< date: Mon, 30 Jun 2025 08:53:20 GMT
< server: Google Frontend
< content-length: 12
<
* Connection #0 to host us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net left intact
```

3. Visit the URL to invoke the Cloud Function. You should see the message Hello World! as the response:

   curl -v https://us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net/helloGET

   Create the API definition

   API Gateway uses an API definition to route calls to the backend service. You can use an OpenAPI spec that contains specialized annotations to define the desired API Gateway behavior. The OpenAPI spec for this quickstart contains routing instructions to the Cloud Function backend.

1. From Cloud Shell, navigate back to your home directory:

   cd ~

2. Create a new file named openapi2-functions.yaml:

   touch openapi2-functions.yaml

3. Copy and paste the contents of the OpenAPI spec shown below into the newly created file:

```
swagger: '2.0'
info:
  title: API_ID description
  description: Sample API on API Gateway with a Google Cloud Functions backend
  version: 1.0.0
```

```
schemes:
  - https
produces:
  - application/json
paths:
  /hello:
    get:
      summary: Greet a user
      operationId: hello
      x-google-backend:
        address: https://us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net/helloGET
      responses:
       '200':
          description: A successful response
          schema:
            type: string
```

Copied!

4. Set the following environment variables:

```
export API_ID="hello-world-$(cat /dev/urandom | tr -dc 'a-z' | fold -w ${1:-8} | head -n 1)"
```

5. Run the following commands to replace the variables set in the last step in the OpenAPI spec file:

```
sed -i "s/API_ID/${API_ID}/g" openapi2-functions.yaml
sed -i "s/PROJECT_ID/$PROJECT_ID/g" openapi2-functions.yaml
```

Task 3. Creating a gateway

Now you are ready to create and deploy a gateway on API Gateway.

1. In the top search bar enter **API Gateway** and select it from the options that appear.
2. Click **Create Gateway**. Then, in the **APIs** section:

- Ensure the **Select an API** input is set to **Create new API**.
- For **Display Name** enter Hello World API

- For **API ID**, run the following command to once again obtain the API ID and enter it into the **API ID** field:

  export API_ID="hello-world-$(cat /dev/urandom | tr -dc 'a-z' | fold -w ${1:-8} | head -n 1)"

  echo $API_ID

3. In the **API Config** section:

- Ensure the **Select a Config** input is set to **Create new API config**.

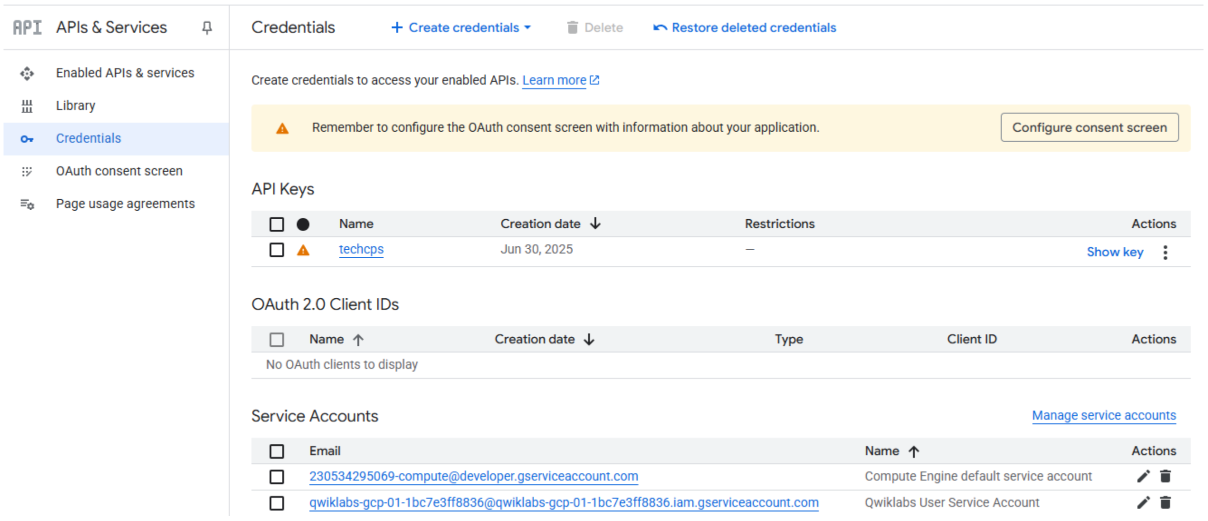
- Do the following to upload the openapi2-functions.yaml file previously created.

4. In Cloud Shell, run the following command:

   cloudshell download $HOME/openapi2-functions.yaml

5. Click **Download**.

6.



Create credentials to access your enabled APIs. Learn more

Remember to configure the OAuth consent screen with information about your application.

API Keys

	Name	Creation date	Restrictions	Actions
⚠	techcps	Jun 30, 2025	–	Show key

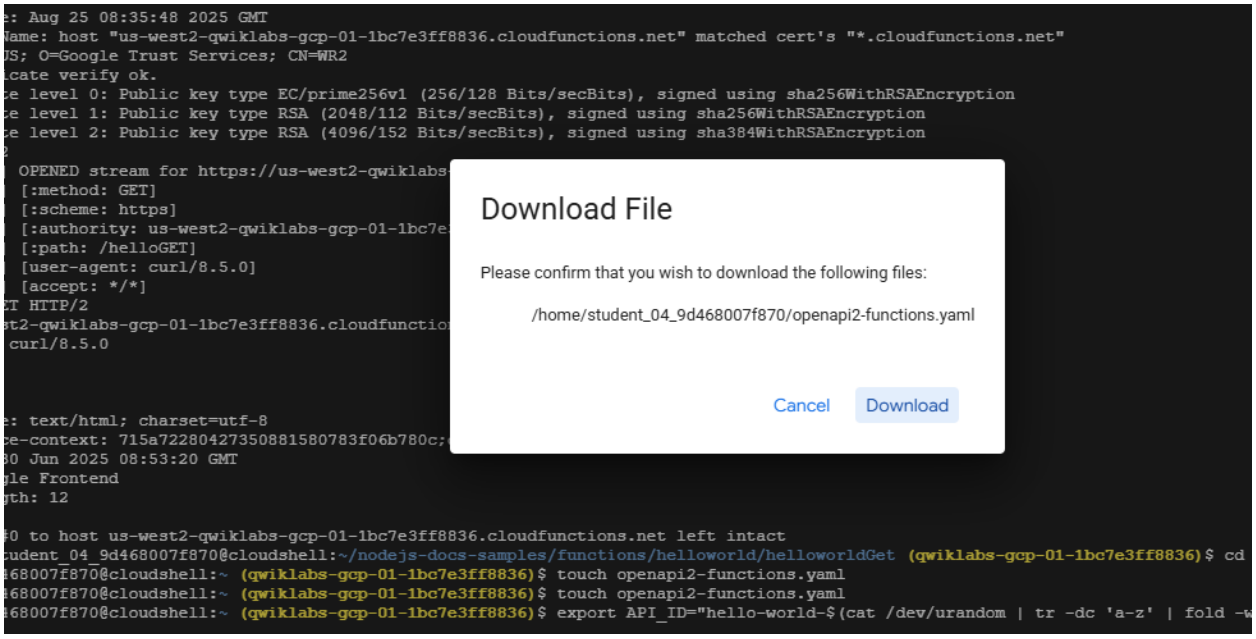
OAuth 2.0 Client IDs

Name	Creation date	Type	Client ID	Actions
No OAuth clients to display				

Service Accounts

Email	Name	Actions
230534295069-compute@developer.gserviceaccount.com	Compute Engine default service account	
qwiklabs-gcp-01-1bc7e3ff8836@qwiklabs-gcp-01-1bc7e3ff8836.iam.gserviceaccount.com	Qwiklabs User Service Account	

7.



```
e: Aug 25 08:35:48 2025 GMT
Name: host "us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net" matched cert's "*.cloudfunctions.net"
US; O=Google Trust Services; CN=WR2
icate verify ok.
ce level 0: Public key type EC/prime256v1 (256/128 Bits/secBits), signed using sha256WithRSAEncryption
ce level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
ce level 2: Public key type RSA (4096/152 Bits/secBits), signed using sha384WithRSAEncryption
2
] OPENED stream for https://us-west2-qwiklabs-
] [:method: GET]
] [:scheme: https]
] [:authority: us-west2-qwiklabs-gcp-01-1bc7e
] [:path: /helloGET]
] [user-agent: curl/8.5.0]
] [accept: */*]
ET HTTP/2
st2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctio
 curl/8.5.0


e: text/html; charset=utf-8
ce-context: 715a7228042735088158078 3f06b780c;
30 Jun 2025 08:53:20 GMT
gle Frontend
gth: 12

#0 to host us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net left intact
tudent_04_9d468007f870@cloudshell:~/nodejs-docs-samples/functions/helloworld/helloworldGet (qwiklabs-gcp-01-1bc7e3ff8836)$ cd
468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ touch openapi2-functions.yaml
468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ touch openapi2-functions.yaml
468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ export API_ID="hello-world-$(cat /dev/urandom | tr -dc 'a-z' | fold -w
```

Download File

Please confirm that you wish to download the following files:

/home/student_04_9d468007f870/openapi2-functions.yaml

8. Select **Browse** and select the file from the browser's download location:

- Enter Hello World Config in the **Display Name** field.
- Ensure the **Select a Service Account** input is set to **Compute Engine default service account**.

7. In the **Gateway details** Section:

- Enter Hello Gateway in the **Display Name** field.
- Set the **Location** drop down to us-west2.

8. Click **Create Gateway**.

   Testing your API Deployment

   Now you can send requests to your API using the URL generated upon deployment of your gateway.

1. In Cloud Shell, enter the following command to

   export GATEWAY_URL=$(gcloud api-gateway gateways describe hello-gateway --location us-west2 --format json | jq -r .defaultHostname)

2. Run the following command to ensure that the GATEWAY_URL environment variable is set:

   echo $GATEWAY_URL

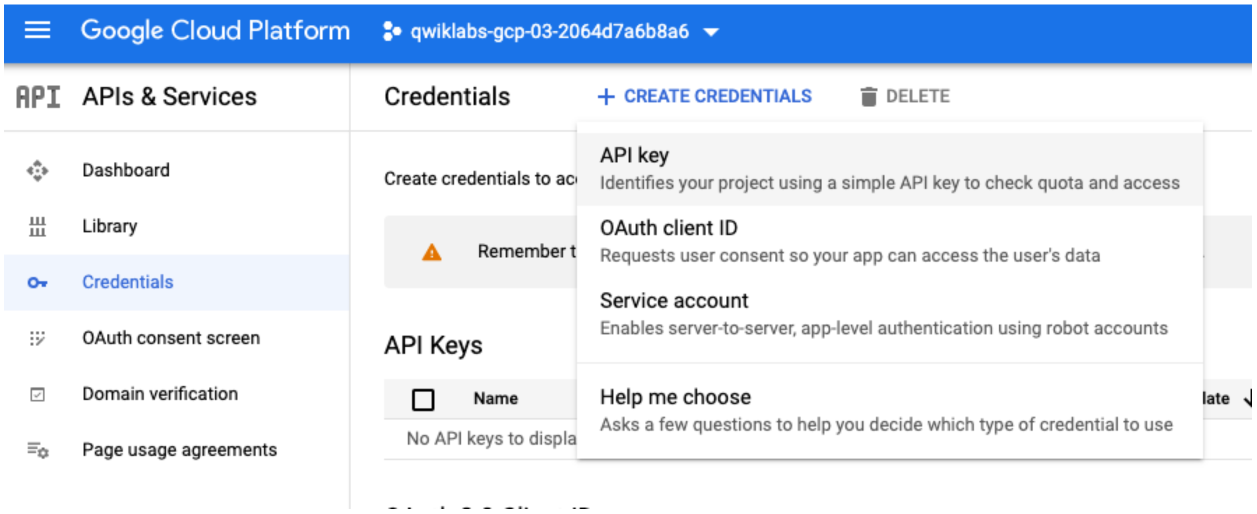
3. Run the following curl command and validate that the response returned is Hello World!:

   ```
   curl -s -w "\n" https://$GATEWAY_URL/hello
   ```

   Task 4. Securing access by using an API key

   To secure access to your API backend, you can generate an API key associated with your project and grant that key access to call your API. To create an API Key you must do the following:

1. In the Cloud Console, navigate to **APIs & Services** > **Credentials**.
2. Select **Create credentials**, then select **API Key** from the dropdown menu. The **API key created** dialog box displays your newly created key.


Google Cloud Platform
qwiklabs-gcp-03-2064d7a6b8a6
APIs & Services
Credentials
CREATE CREDENTIALS
DELETE
Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements
Create credentials to ac
Remember t
API Keys
Name
No API keys to displa
API key
Identifies your project using a simple API key to check quota and access
OAuth client ID
Requests user consent so your app can access the user's data
Service account
Enables server-to-server, app-level authentication using robot accounts
Help me choose
Asks a few questions to help you decide which type of credential to use

3. Copy the API Key from the dialog, then click on **close**.
4. Store the API Key value in Cloud Shell by running the following command:

```
export API_KEY=REPLACE_WITH_COPIED_API_KEY
```

1. In Cloud Shell, obtain the name of the Managed Service you just created using the following command:

   ```
   MANAGED_SERVICE=$(gcloud api-gateway apis list --format json | jq -
   r .[0].managedService | cut -d'/' -f6)
   ```

echo $MANAGED_SERVICE

2. Then, using the Managed Service name of the API you just created, run this command to **enable** API key support for the service:

   gcloud services enable $MANAGED_SERVICE

   Modify the OpenAPI Spec to leverage API Key Security

   In this section, modify the API config of the deployed API to enforce an API key validation security policy on all traffic.

1. Add the security type and securityDefinitions sections to a new file called openapi2-functions2.yaml file as shown below:

   touch openapi2-functions2.yaml

3. Run the following commands to replace the variables set in the last step in the OpenAPI spec file:

   sed -i "s/API_ID/${API_ID}/g" openapi2-functions2.yaml sed -i "s/PROJECT_ID/$PROJECT_ID/g" openapi2-functions2.yaml

4. Download the updated API spec file, you will use it to update the Gateway config in the next step:

   cloudshell download $HOME/openapi2-functions2.yaml

5. Click Download.

   Task 5. Create and deploy a new API config to your existing gateway

1. Open the **API Gateway** page in Cloud Console. (Click **Navigation Menu > API Gateway**.)
2. Select your API from the list to view details.
3. Select the **Gateways** tab.
4. Select Hello Gateway from the list of available **Gateways**.
5. Click on Edit at the top of the Gateway page.
6. Under **API Config** change the drop down to Create new API config.
7. Click **Browse** in the **Upload an API Spec** input box and select the openapi2-functions2.yaml file.
8. Enter Hello Config for **Display Name**.

9. Select Qwiklabs User Service Account for **Select a Service Account**.

10. Click **Update**.

Task 6. Testing calls using your API key

1. To test using your API key run the following command:

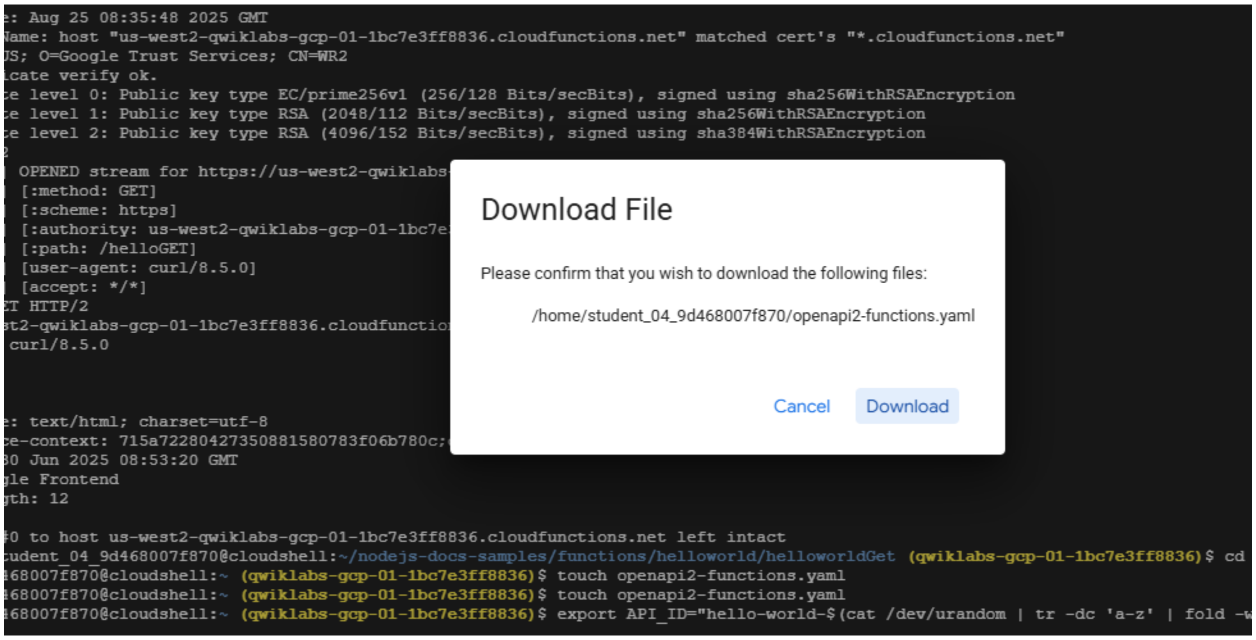
   export GATEWAY_URL=$(gcloud api-gateway gateways describe hello-gateway --location us-west2 --format json | jq -r .defaultHostname)

   curl -sL $GATEWAY_URL/hello

2. Run the following curl command with the key query parameter and use the API key previously created to call the API:

   curl -sL -w "\n" $GATEWAY_URL/hello?key=$API_KEY

   If you do not have the API_KEY environment variable set you can get your API key from the left menu by navigating **APIs & Services** > **Credentials**. The key will be available under the **API Keys** section.



```
e: Aug 25 08:35:48 2025 GMT
Name: host "us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net" matched cert's "*.cloudfunctions.net"
US; O=Google Trust Services; CN=WR2
icate verify ok.
ce level 0: Public key type EC/prime256v1 (256/128 Bits/secBits), signed using sha256WithRSAEncryption
ce level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
ce level 2: Public key type RSA (4096/152 Bits/secBits), signed using sha384WithRSAEncryption
2
] OPENED stream for https://us-west2-qwiklabs
] [:method: GET]
] [:scheme: https]
] [:authority: us-west2-qwiklabs-gcp-01-1bc7e
] [:path: /helloGET]
] [user-agent: curl/8.5.0]
] [accept: */*]
ET HTTP/2
st2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctio
 curl/8.5.0


e: text/html; charset=utf-8
ce-context: 715a722804273508815807 83f06b780c;
30 Jun 2025 08:53:20 GMT
gle Frontend
gth: 12

#0 to host us-west2-qwiklabs-gcp-01-1bc7e3ff8836.cloudfunctions.net left intact
tudent_04_9d468007f870@cloudshell:~/nodejs-docs-samples/functions/helloworld/helloworldGet (qwiklabs-gcp-01-1bc7e3ff8836)$ cd
468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ touch openapi2-functions.yaml
468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ touch openapi2-functions.yaml
468007f870@cloudshell:~ (qwiklabs-gcp-01-1bc7e3ff8836)$ export API_ID="hello-world-$(cat /dev/urandom | tr -dc 'a-z' | fold -w
```

Download File

Please confirm that you wish to download the following files:

/home/student_04_9d468007f870/openapi2-functions.yaml