API Gateway: Qwik Start — Lab Documentation

Lab ID: GSP872

Duration: 1 Hour

Level: Introductory

Platform: Google Cloud Skills Boost (Qwiklabs)

Status: Completed

Badge: [API Gateway: Qwik Start Skill Badge]

1. Overview

This lab demonstrates how to deploy and secure a REST API using **Google Cloud API Gateway**.

API Gateway provides a unified and consistent API endpoint to securely expose backend services, such as **Cloud Functions**, to consumers. It enables scalability, access control, and simplified API lifecycle management.

The goal of this lab was to:

- Deploy a backend **Cloud Function**
- Create and configure an **API Gateway**
- Secure API access using an **API key**

2. Environment Setup

1. Launched lab environment through Qwiklabs and logged in using **temporary Google Cloud credentials**.
2. Opened **Google Cloud Console** in an Incognito window to prevent conflicts with personal accounts.
3. Activated **Cloud Shell** to execute all CLI operations.
4. Set the active region for resource deployment.

```
gcloud config set compute/region REGION
```

1. Enabled the **API Gateway API** and verified access permissions.

3. Task 1 – Deploying the API Backend

Created and deployed a **Cloud Function** named `helloGET` using **Node.js 20** runtime.

Function Code (index.js):

```
exports.helloGET = (req, res) ⇒ {
  res.send('Hello World!');
};
```

Deployment Command:

```
gcloud functions deploy helloGET \
--runtime nodejs20 \
--trigger-http \
--allow-unauthenticated \
--region REGION
```

Verification:

After deployment, tested the Cloud Function endpoint using:

```
curl -v https://REGION-PROJECT_ID.cloudfunctions.net/helloGET
```

Output:

```
Hello World!
```

4. Task 2 – Creating the API Definition

Defined an **OpenAPI Specification** (`openapi2-functions.yaml`) for the API Gateway to route traffic to the Cloud Function backend.

Sample Specification:

```
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://REGION-PROJECT_ID.cloudfunctions.net/helloGET
      responses:
        '200':
          description: A successful response
          schema:
            type: string
```

Environment variables were dynamically replaced for project and region values.

5. Task 3 – Creating and Deploying the Gateway

Created a new API Gateway named **Hello Gateway** using the previously defined OpenAPI specification.

Steps:

1. Navigated to **API Gateway → Create Gateway**.
2. Created a new API:
   - **API Name:** Hello World API
   - **API ID:** hello-world-[random]
3. Uploaded `openapi2-functions.yaml`.
4. Selected **Compute Engine default service account** for execution.
5. Waited for deployment to complete (~10 minutes).

Verification:

Retrieved Gateway URL:

```
export GATEWAY_URL=$(gcloud api-gateway gateways describe hello-gateway --location REGION --format json | jq -r .defaultHostname)
curl -s https://$GATEWAY_URL/hello
```

Output:

```
Hello World!
```

6. Task 4 – Securing the API using API Keys

Generated and attached an API key for authorized access.

Steps:

1. In Cloud Console, navigated to:

   ```
   APIs & Services → Credentials → Create Credentials → API Key
   ```

2. Stored key in Cloud Shell:

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

3. Modified the OpenAPI definition to enforce API key validation:

```
securityDefinitions:
  api_key:
    type: apiKey
    name: key
    in: query
paths:
  /hello:
    get:
      summary: Greet a user
      operationId: hello
      x-google-backend:
        address: https://REGION-PROJECT_ID.cloudfunctions.net/helloGET
      security:
        - api_key: []
```

1. Uploaded the updated file (`openapi2-functions2.yaml`) and created a new **API Config** under the same Gateway.

7. Task 5 – Testing Secured API

Without API Key:

```
curl -sL $GATEWAY_URL/hello
```

Response:

```
UNAUTHENTICATED: Method doesn't allow unregistered callers...
```

With API Key:

```
curl -sL "$GATEWAY_URL/hello?key=$API_KEY"
```

Response:

```
Hello World!
```

This confirmed that the API Gateway correctly enforced key-based authentication.

8. Outcomes

Successfully completed and validated all objectives of the lab:

- Created and deployed **Cloud Function backend**
- Configured and deployed **API Gateway**

- Implemented **secure API access** using API keys
- Verified end-to-end communication through **cURL tests**

9. Skills Gained

- Configuring and deploying APIs using Google Cloud API Gateway
- Integrating API Gateway with Cloud Functions backend
- Implementing API key–based authentication and security policies
- Managing OpenAPI specifications in Google Cloud environments
- Using **Cloud Shell**, **gcloud CLI**, and **IAM service accounts**