Creating an API config

cloud.google.com/api-gateway/docs/creating-api-config

Prerequisites

Before you can create an API config, ensure that you have:

- Prepared your development environment as described in Configuring your development environment.
- Created an API definition as an OpenAPI spec.
- Optionally created an API as described in Creating an API. If the API does not already exist, then creating the API config creates it.

API config ID requirements

Many of the `gcloud` commands shown below require you to specify the ID of the API config, in the form: *CONFIG_ID*. API Gateway enforces the following requirements for the API config ID:

- Must have a maximum length of 63 characters.
- Must contain only lowercase letters, numbers, or dashes.
- Must not start with a dash.
- Must not contain an underscore.

Use the `gcloud` command-line tool to upload your API definition to create an API config. When you upload the API definition, you are required to specify the name of the API. If the API does not already exist in API Gateway then this command also creates it.

Note: API Gateway supports creation of API definitions as OpenAPI specs for all available backend services. API Gateway also supports the use of gRPC service definitions and configurations for gRPC services on Cloud Run.
To create an API config:

1. Change directory to the directory containing your API definition.

   For more on creating the OpenAPI spec for your API definition, see OpenAPI overview and Quickstart: Deploy an API on API Gateway.

   For more on creating a gRPC service definition and configuration for your API definition, see Configuring a gRPC service and Getting started with API Gateway and Cloud Run for gRPC.

2. Validate the project ID returned from the following command to make sure that the service isn't created in the wrong project.

   ```
   gcloud config list project
   ```

   If you need to change the default project, run the following command and replace *PROJECT_ID* with the Google Cloud project ID in which you want to create the service::

   ```
   gcloud config set project PROJECT_ID
   ```

3. View help for the `api-configs create` command:

   ```
   gcloud api-gateway api-configs create --help
   ```

4. Run the following command to create the API config:

   ```
   gcloud api-gateway api-configs create CONFIG_ID \
     --api=API_ID --openapi-spec=API_DEFINITION \
     --project=PROJECT_ID --backend-auth-service-account=SERVICE_ACCOUNT_EMAIL
   ```

   where:

   - *CONFIG_ID* specifies the ID of the new API config.
   - *API_ID* specifies the ID of the API Gateway API associated with this API config. If the API does not already exist then this command creates it.
   - *API_DEFINITION* specifies the name of the OpenAPI spec containing the API definition.
   - *PROJECT_ID* specifies the Google Cloud project ID.
   - *SERVICE_ACCOUNT_EMAIL* specifies the service account created explicitly for creating API configs. See Configuring a service account for more details.

   As it is creating the API and API config, API Gateway outputs information to the terminal. This operation may take several minutes to complete as the API config is propagated to downstream systems. Creation of a complex API config could take up to ten minutes to complete successfully.

5. On successful completion, you can use the following command to view details about the new API config:

   ```
   gcloud api-gateway api-configs describe CONFIG_ID \
     --api=API_ID --project=PROJECT_ID
   ```

   This command returns the following:

   ```
   createTime: '2020-02-04T18:33:11.882707149Z'
   displayName: CONFIG_ID
   gatewayConfig:
     backendConfig:
       googleServiceAccount: 1111111@developer.gserviceaccount.com
   labels: ''
   name: projects/PROJECT_ID/locations/global/apis/API_ID/configs/CONFIG_ID
   serviceRollout:
     rolloutId: 2020-02-04r2
   state: ACTIVE
   updateTime: '2020-02-04T18:33:12.219323647Z'
   ```

6. Enable the API:

   ```
   gcloud services enable API_ID-HASH.apigateway.PROJECT_ID.cloud.goog
   ```

   You only have to run this command once when you create the API. If you later modify the API, you do not have to rerun the command.

The `gcloud` command-line tool takes many options, including those described in the gcloud Reference. In addition, for API Gateway, you can set the following options when creating an API config:

- `--async` : Return control to the terminal immediately, without waiting for the operation to complete.
- `--display-name=NAME` : Specifies the display name of the API config, meaning the name shown in the UI. Do not use spaces in the name. Use hyphens and underscores instead. The default value is *CONFIG_ID*.
- `--labels=KEY1=VALUE1,KEY2=VALUE2,...` : Specifies labels associated with the API config.

For example:

```
gcloud api-gateway api-configs create CONFIG_ID \
  --api=API_ID --openapi-spec=API_DEFINITION \
  --project=PROJECT_ID --backend-auth-service-account=SERVICE_ACCOUNT_EMAIL \
  --async --display-name=MyConfig --labels=a=1,b=2
```

You can see the labels in the output of the `describe` command shown above, or in the `list` command by including the `--format` option:

```
gcloud api-gateway api-configs list \
  --api=API_ID --project=PROJECT_ID --format="table(name, labels)"
```

Listing API configs

To list API configs for a specific project:

```
gcloud api-gateway api-configs list --project=PROJECT_ID
```

This command returns the following:

```
NAME                                                                DISPLAY_NAME
 ROLLOUT_ID    STATE   CREATE_TIME
projects/PROJECT_ID/locations/global/apis/API_ID/configs/CONFIG_ID  CONFIG_ID
2020-02-04r0  ACTIVE  2020-02-04T16:18:02.369859863Z
```

To list API configs for a specific API in a project:

```
gcloud api-gateway api-configs list --api=API_ID --project=PROJECT_ID
```

Use the project, API, and config IDs to obtain detailed information about the API config:

```
gcloud api-gateway api-configs describe CONFIG_ID \
  --api=API_ID --project=PROJECT_ID
```

Updating an API config

You cannot modify an existing API config other than to update its labels and its display name.

Note: Every time you upload an API definition, API Gateway creates a new API config. To update an API definition:

1. Update the OpenAPI spec containing the API definition.
2. Upload the API definition to create a **new** API config.

Use the same command to create the **new** API config as you used to create the initial API config, as shown above in Creating an API config.

Use the following `gcloud` to update an existing API config:

- `--display-name`
- `--update-labels`
- `--clear-labels`
- `--remove-labels`

For example:

```
gcloud api-gateway api-configs update CONFIG_ID \
  --api=API_ID  --project=PROJECT_ID \
  --update-labels=a=1,b=2
```

Use the following command to view all update options:

```
gcloud api-gateway api-configs update --help
```

Deleting an API config

Warning: If you deployed the API config to a gateway, the API config may not be deleted until the gateway is no longer using that API config.
Before you delete an API config that is in use, you must either:

- Deploy a different API config to the gateway.
- Delete the gateway.

See Deploying an API to a gateway for more.

Use the following `gcloud` command to delete an existing API config:

```
gcloud api-gateway api-configs delete CONFIG_ID --api=API_ID --project=PROJECT_ID
```

What's next

Deploying an API to a gateway

Rate and review