Configuring the development environment

cloud.google.com/api-gateway/docs/configure-dev-env

This document describes how to configure your API Gateway development environment.

Prerequisites

Before you can create an API on API Gateway, ensure that you have:

- Created a Google Cloud project in which you have the **Editor** or **Owner** role. After the initial deployment, you can grant the more restrictive **Service Config Editor** role to a user, group, or service account.
- Prepared the Cloud SDK as described below.
- Enabled the required Google services as described below.
- Configure the service account used to create API configs as described below.

Preparing the Cloud SDK for deployment

To prepare `gcloud` for the deployment:

1. Install and initialize the Cloud SDK.
2. Update Cloud SDK:

   ```
   gcloud components update
   ```

3. Make sure that Cloud SDK is authorized to access your data and services:

   ```
   gcloud auth login
   ```

   A new browser tab opens and you are prompted to choose an account.

4. Set the default project. Replace *PROJECT_ID* with your Google Cloud project ID:

   ```
   gcloud config set project PROJECT_ID
   ```

Enabling required services

API Gateway requires that you enable the following Google services:

Name	Title
`apigateway.googleapis.com`	API Gateway API
`servicemanagement.googleapis.com`	Service Management API

Name	Title
`servicecontrol.googleapis.com`	Service Control API

To confirm that the required services are enabled:

```
gcloud services list
```

If you do not see the required services listed, enable them:

```
gcloud services enable apigateway.googleapis.com
gcloud services enable servicemanagement.googleapis.com
gcloud services enable servicecontrol.googleapis.com
```

For more information about the `gcloud` services, see gcloud services.

Configuring a service account

An API config deployed on a gateway executes with the permissions associated with the gateway service account.

As a best practice, create a separate service account in the same project as you are using for API Gateway. Then assign the service account only the permissions necessary to access the backend service. In that way, you limit the permissions associated with the API config.

For API Gateway the user creating or updating an API config or gateway requires the `iam.serviceAccounts.actAs` permission on the service account object. This permission is included in the Service Account User role.

The role and permission can be added to the service account for the user with the following command:

```
gcloud iam service-accounts SERVICE_ACCOUNT \
add-iam-policy-binding --member user:USER_EMAIL -role roles/iam.serviceAccountUser
```

In addition, the gateway service account requires the permissions necessary to access your backend service. For example, if your backend is implemented as a Cloud Function, then the service account should at least be assigned the role of **Cloud Functions Invoker**. For a Cloud Run backend, the role is **Cloud Run Invoker**. By limiting the permissions associated with the API config, you can better secure your backend systems.

For more information, see the Identity and Access Management (IAM) documentation.

After you create the service account, use the `--backend-auth-service-account` option to specify the email address of that service account when creating an 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
```

See Creating an API for more on creating API configs.

Using a default service account

Some GCP products define a *default* service account. For example, if you are using Compute Engine and have enabled the Compute Engine API for your project, a default Compute Engine service account is created for you. The default service account is identifiable by its email address:

```
PROJECT_NUMBER-compute@developer.gserviceaccount.com
```

If you assign the necessary permissions to the default service account, you can omit the `--backend-auth-service-account` option when creating an API config:

```
gcloud api-gateway api-configs create CONFIG_ID \
  --api=API_ID --openapi-spec=API_DEFINITION --project=PROJECT_ID
```

See Using the Compute Engine Default Service Account for more.

Using OpenID Connect

Requests from API Gateway to backend services may use authentication. These requests are secured using OpenID Connect (OIDC) tokens signed by the gateway's service account. You should confirm that your backend service is correctly configured to accept OIDC tokens for authentication and authorization. Cloud Functions, Cloud Run, and the Identity Aware Proxy (IAP) provide this option.

What's next

- Creating an API

Rate and review