Service identity

Runtime service account

During its execution, a Cloud Run revision uses a service account as its identity. This means that when your code uses Google Cloud client libraries (https://cloud.google.com/apis/docs/cloud-client-libraries), it automatically obtains and uses credentials from the runtime service account of the current Cloud Run revision. This strategy is called "Application Default Credentials" (https://cloud.google.com/docs/authentication/production#providing_credentials_to_your_application).

By default, Cloud Run revisions are using the **Compute Engine default service account** (`PROJECT_NUMBER-compute@developer.gserviceaccount.com`), which has the *Project > Editor* IAM role. This means that by default, your Cloud Run revisions have read and write access to all resources in your Google Cloud project. While this is very convenient, we recommend granting more granular permissions to each of your Cloud Run services by assigning dedicated service accounts (#per-service-identity) with more restricted IAM roles.

Using per-service identity

Important: the following sections on setting the service account on your service currently applies only to Cloud Run (fully managed).

It is recommended that you give each of your services a dedicated identity and restrict what it is able to access by granting it a minimal set of permissions using IAM (https://cloud.google.com/iam). You can do this by assigning a named service account that has the correct IAM role(s). You can only use service accounts in the same project as the Cloud Run (fully managed) service.

Permissions required to use non-default identities

In order to deploy a service with a non-default service account, the deployer must have the `iam.serviceAccounts.actAs` permission on the service account being deployed.

If a user creates a service account, that user is automatically granted this permission; otherwise, a user with the correct permissions must grant the deployer this permission on the service account in order for the user to deploy.

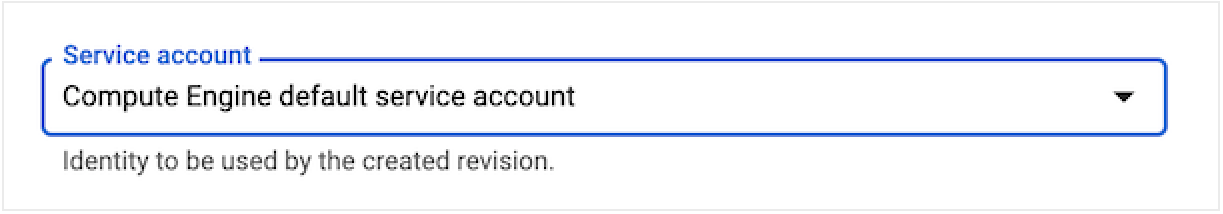
Deploying a new service with a non-default identity

Before you deploy a service with a new identity, make sure that the service account you want to use is already created. If not, learn how to create and manage service accounts (https://cloud.google.com/iam/docs/creating-managing-service-accounts).

You can set environment variables using the Cloud Console or the gcloud command line when you create a new service (https://cloud.google.com/run/docs/deploying#service) or deploy a new revision (https://cloud.google.com/run/docs/deploying#revision):

CONSOLE UI GCLOUD

1. GO TO CLOUD RUN (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/RUN)
2. Click **CREATE SERVICE** if you are configuring a service account on a new service you are deploying to. If you are configuring a service account on an existing service, then click on the service, then click **DEPLOY NEW REVISION**.
3. Click **SHOW OPTIONAL SETTINGS**.


Service account
Compute Engine default service account
Identity to be used by the created revision.

4. Click the **Service account** dropdown and select the desired service account.
5. Click **Create** or **Deploy**.

Fetching identity and access tokens

When your code runs on Cloud Run (fully managed) it can use the Compute Metadata Server (https://cloud.google.com/compute/docs/storing-retrieving-metadata) to fetch identity tokens and access tokens. You cannot query the metadata server directly from your local computer.

Identity tokens

You use identity tokens when calling other Cloud Run (fully managed) services or any other service that can validate an identity token (https://developers.google.com/identity/sign-in/web/backend-auth#verify-the-integrity-of-the-id-token).

You can use the Compute Metadata Server to fetch identity tokens (https://cloud.google.com/compute/docs/instances/verifying-instance-identity#request_signature) with a specific audience as follows:

```
curl "http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/d
  -H "Metadata-Flavor: Google"
```

Where ***`AUDIENCE`*** is the JWT Audience requested, for example: the URL of a service you're invoking, such as **`https://service.domain.com`**, or the OAuth Client ID of an IAP protected resource, such as **`1234567890.apps.googleusercontent.com`**.

Access tokens

You use access tokens when calling Google APIs.

By default, access tokens have the **`cloud-platform`** scope, which allows access to all Google Cloud Platform APIs, assuming IAM also allows access. In order to access other Google or Google Cloud APIs, you will need to fetch an access token with the appropriate scope.

You can use the Compute Metadata Server to fetch access tokens (https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances#applications)
.

If you need an access token with a specific scope, you can generate one as follows:

```
curl "http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/d
  -H "Metadata-Flavor: Google"
```

Where `SCOPES` is a comma separated list of OAuth scopes requested, for example: `https://www.googleapis.com/auth/drive,https://www.googleapis.com/auth/spreadsheets`.

Consult the full list of Google OAuth scopes (https://developers.google.com/identity/protocols/googlescopes) to find which scopes you need.

Note: the `?scopes=` query parameter is only available on App Engine, Cloud Functions, and Cloud Run.

Next steps

Learn how to manage access (https://cloud.google.com/run/docs/securing/managing-access) to or securely authenticate developers, services, and end-users (https://cloud.google.com/run/docs/securing/authenticating) to your services.