Invoking with an HTTPS Request

You can send HTTPS requests from anything able to make HTTPS requests to trigger a Cloud Run-hosted service. Note that all Cloud Run services have a stable HTTPS URL.

Some of the use cases include:

- Custom RESTful web API
- Private microservice
- Web proxy middleware
- Pre-packaged web application

Creating public services

Creating a public service on Cloud Run requires:

- Access to the service from the public internet
- A URL intended for public use

To make a service public, set your service to allow unauthenticated (public) access when you deploy (https://cloud.google.com/run/docs/deploying#service), or at any time after you deploy (https://cloud.google.com/run/docs/authenticating/public).

You can use the stable, auto-assigned URL provided on the first deployment of your service as the public URL on fully managed Cloud Run.

You can also use your own custom domain that maps to the service (https://cloud.google.com/run/docs/mapping-custom-domains). For fully managed Cloud Run, this automatically provides managed SSL certificates.

Creating private services

Creating a private service on Cloud Run requires you to limit access to the service.

For managed Cloud Run, you can limit access to your service using Cloud IAM (https://cloud.google.com/iam/) as discussed in Managing Access via Cloud IAM (https://cloud.google.com/run/docs/securing/managing-access).

The easiest way for developers to test private services is to use a tool like `curl` and pass an auth token in the `Authorization` header:

```
curl -H "Authorization: Bearer $(gcloud auth print-identity-token)" SERVICE_URL
```

Note that a Cloud Run service can call another managed Cloud Run service with service-to-service authentication (https://cloud.google.com/run/docs/authenticating/service-to-service).

For Cloud Run for Anthos on Google Cloud, you can use the internal connectivity setting to prevent external requests from routing to your service.

In addition to the above listed ways to limit access, you can also limit access to a service using application-level authorization and authentication mechanism, for example, using Identity Platform (https://cloud.google.com/identity-platform). For a tutorial on doing this with Cloud Run for Anthos on Google Cloud, refer to Authenticating Cloud Run on GKE end users using Istio and Identity Platform (https://cloud.google.com/solutions/authenticating-cloud-run-on-gke-end-users-using-istio-and-identity-platform)

.

Using a middleware to enhance your service

HTTPS proxies can offload common functionality from an HTTP service, such as caching, request validation, or authorization. For microservices, many HTTP proxies are part of an API Gateway (https://en.wikipedia.org/wiki/API_management) solution or a service mesh such as Istio (https://istio.io/).

Google Cloud products that you can use to enhance your Cloud Run service include:

- Cloud Endpoints (https://cloud.google.com/endpoints/docs/openapi/get-started-cloud-run), which you can use to deploy your own endpoint to Cloud Run and configure it as an API Gateway proxy to other Cloud Run services.

- Firebase Hosting (https://firebase.google.com/docs/hosting), which you can use to build a web application frontend to use with Cloud Run as a dynamic backend (https://firebase.google.com/docs/hosting/cloud-run).

Last updated December 6, 2019.