WHAT IS GOOGLE CLOUD RUN?

Surprising no one, Google has created another cloud service. If you'd like to run an already built application in the cloud, use Cloud Run. It is good for all-in-one software bundles, such as open source software or something like a simple Flask app.

With no infrastructure to manage, a team can focus on writing the application itself—letting Cloud Run handle provisioning resources to meet demand. So, let's look at how Cloud Run works, when to use it, pros and cons, and steps to get started.

How Cloud Run works

When it comes to managing servers, problems abound:

- Provisioning servers
- Scaling servers up/down to meet demands
- Overpaying when more resources are allocated than necessary

Fortunately, Cloud Run automates most of this process. (Developers rejoice!) A person simply tells Cloud Run where their container is, then presses "Create". Cloud Run takes in a container with your pre-built app on it, deploying it in a serverless environment. Cloud Run eliminates the need to manage resources or create an infrastructure, allowing DevOps teams to stay in the code.

There are some stipulations: the logic inside the container has to be stateless, and you must specify a combination of memory vs CPU resources as well as allowed concurrencies. The cloud container

will delete data that is stored on it. A persistent volume must be attached, or other data transfer methods used while the container is running, to prevent any stored data from being deleted.

Afterwards, Cloud Run automatically adds or removes servers to meet user demand, only charging for what is used to the nearest tenth of a second. Cloud Run provides the user the http endpoint, container usage information, and control over the container's billing.

The Cloud Run usage can be monitored easily in the GUI Metrics, along with logs from the container. Container deployment also has a built-in versioning feature, so a rollback to a previous container version is easy.

For those familiar with the AWS environment, Google Cloud run is not unlike AWS Lambda, though there are plenty of differences.

Cloud Run pros and cons

A developer gets a lot of freedom to create when using a container. They can specify the app requirements in a docker file, using any library and any coding language. So, by and large, developers like containers.

Cloud Run makes container deployment even easier. It's particularly good for:

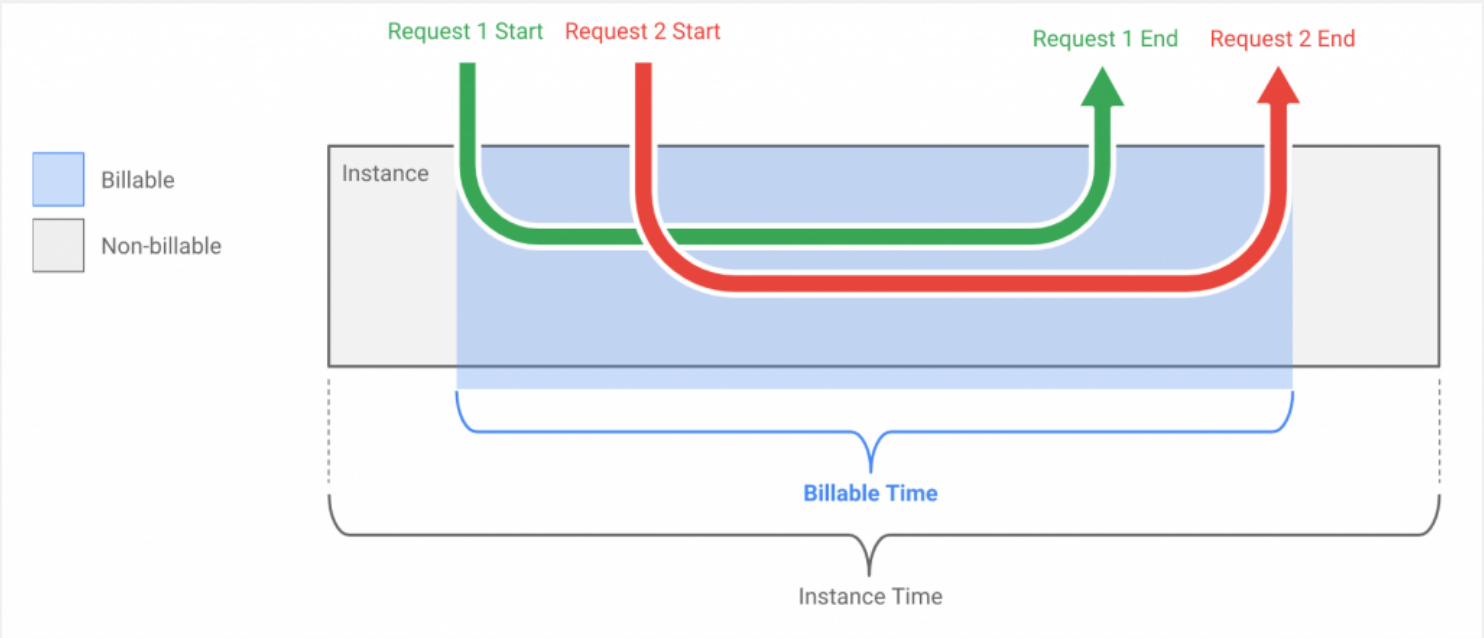
- Developing software in cloud applications
- Delivering web apps, APIs, background jobs

As for what's missing, Cloud Run does come with the limitation that comes with using containers. Containers handle all aspects of the service within a single container. Serverless design and pay-as-you-go pricing benefits by splitting tasks into different pieces of logic. That design cannot occur when one container handles everything. Google services like Cloud Functions can be used to split a container's functions into their own services (FaaS).

Cloud Run pricing

With Cloud Run, you only pay for what you use. Billed time is broken down to the nearest 0.1 of a second.

Google provides a nice chart that shows how billing works when two requests are made and their timing overlaps. The billing period will begin when the first request is made and when the last request is returned.



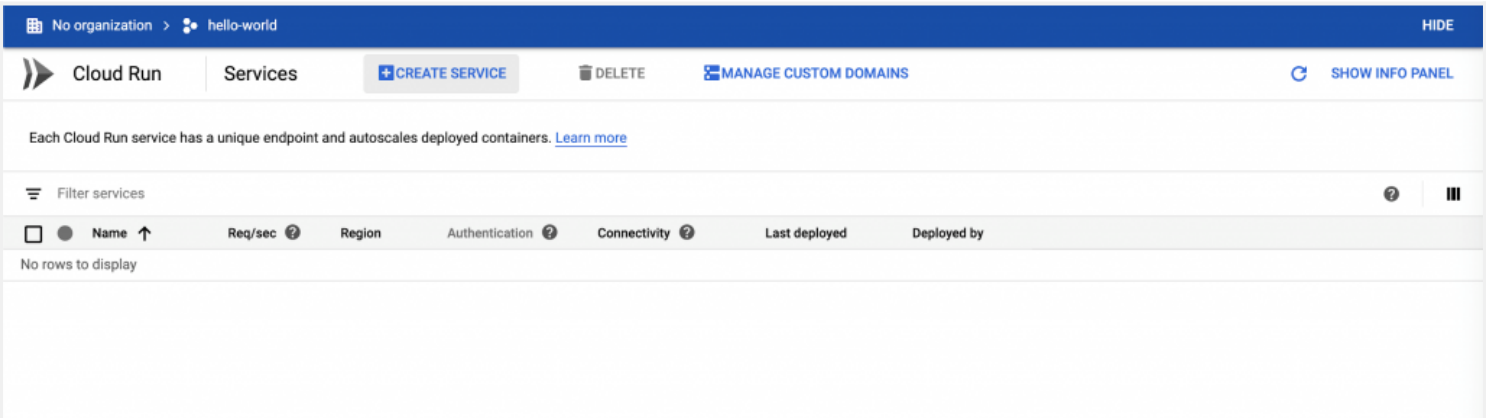
The free-tier allows for 180,000 vCPU-Seconds free, the first 360,000 GB-seconds of memory free, 2 million requests free, and 1 GB free egress within North America. Remember, Google gives $300 in GCP credit when you sign up.

How to start Google Cloud Run

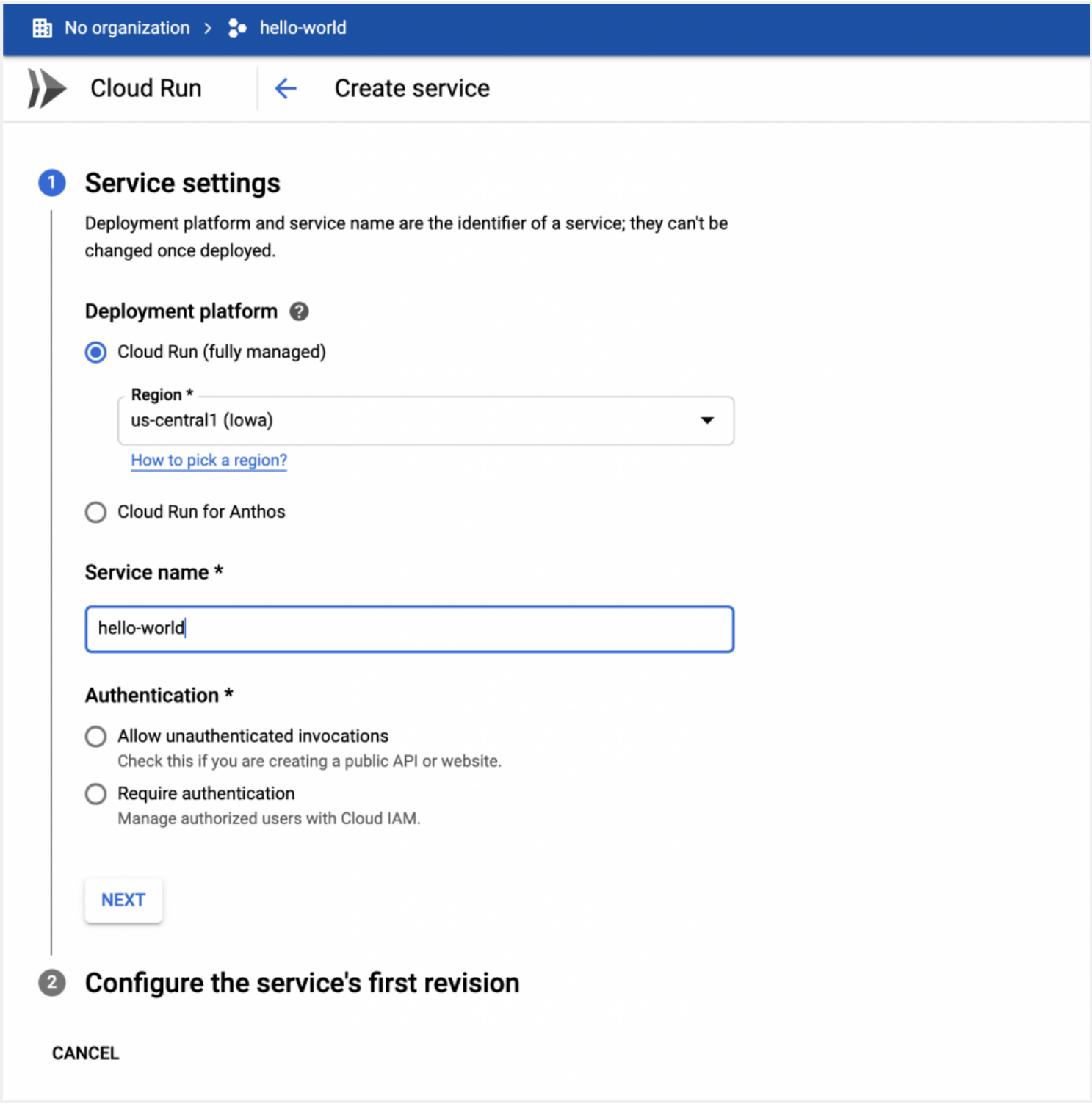
Getting started with Google Cloud Run is straightforward. They include a demo container on their site, so you can go through the steps and try it yourself. Remember to delete the Cloud Run Service after testing.

1. Go to your Cloud Run.

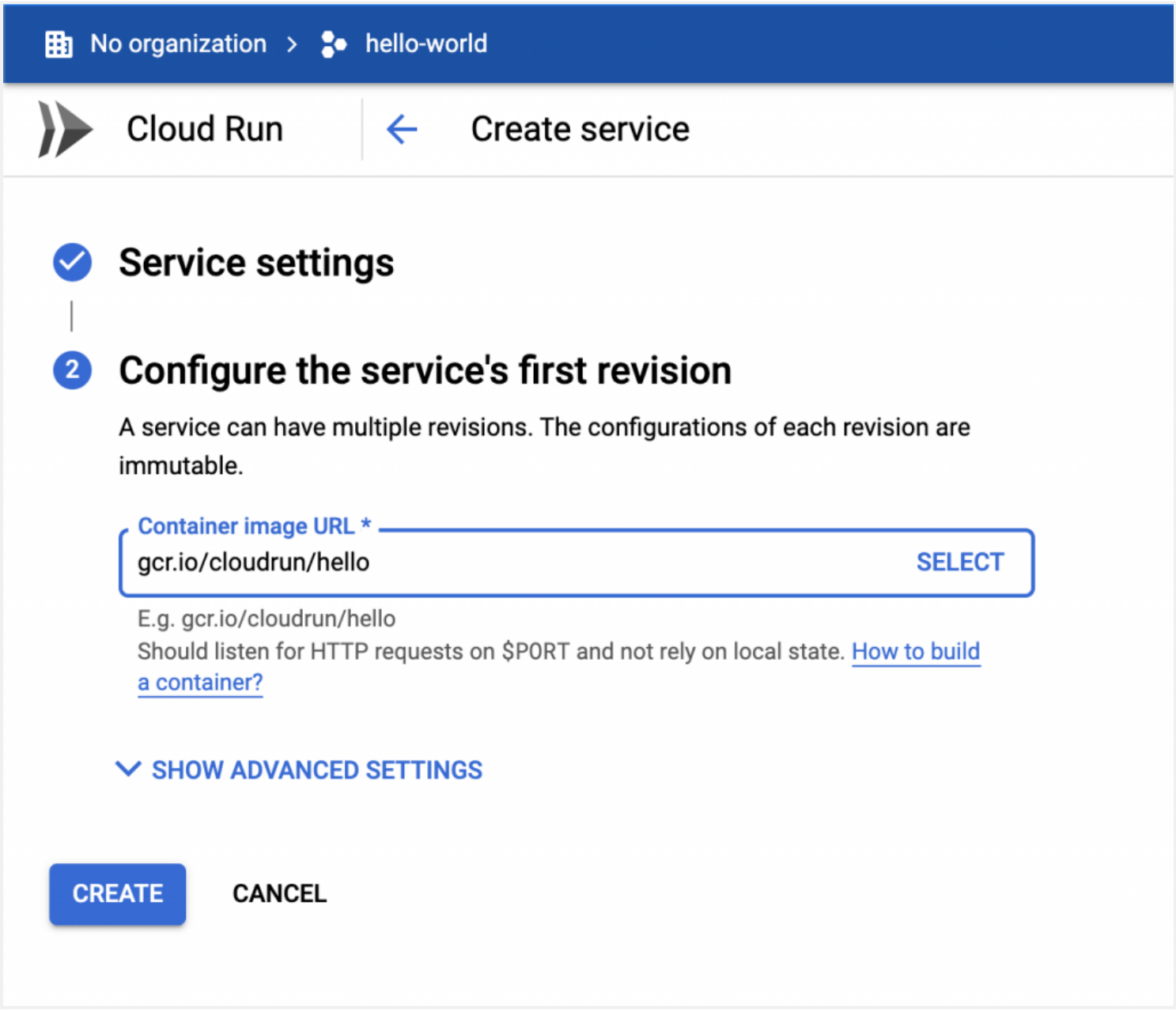
2. Click “Create Service”



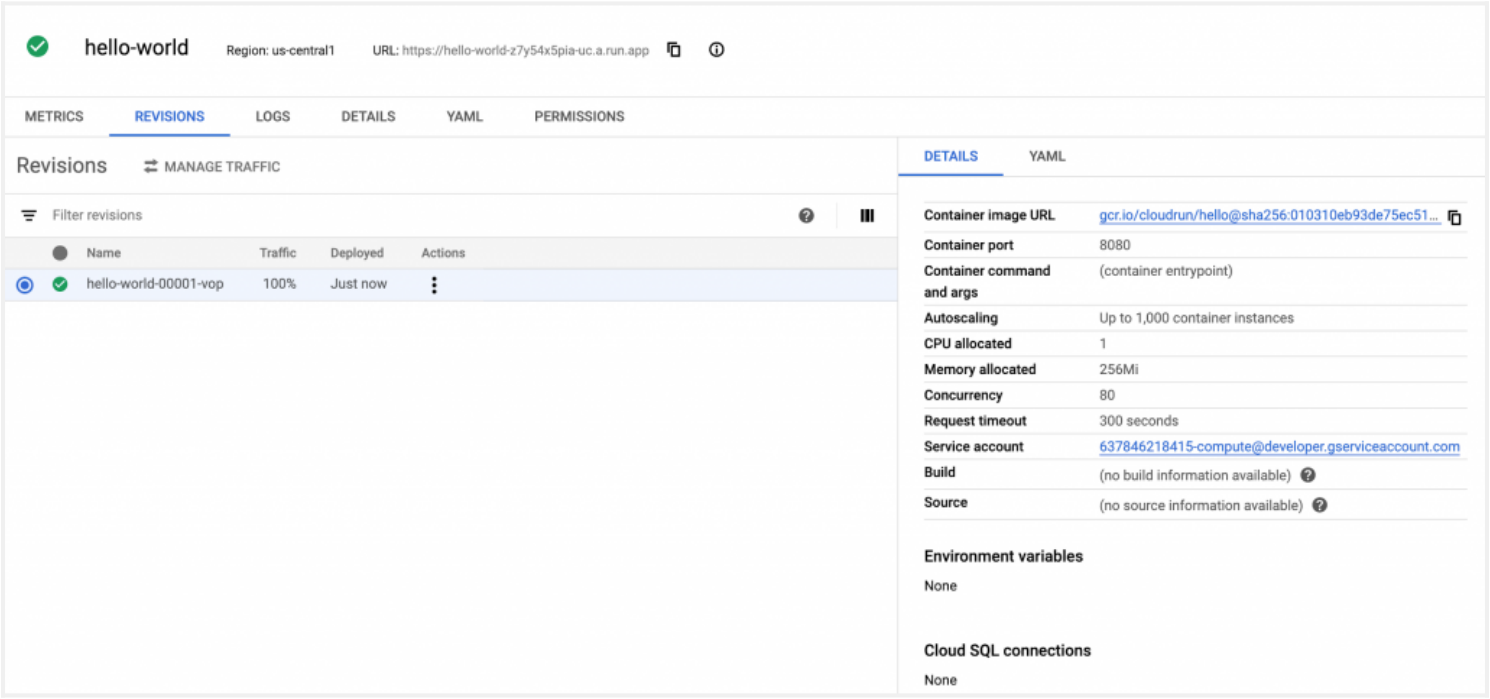
3. Choose your region and select a service name.



4. Add the container URL and hit “Create”.



5. See your container deploy and access its URL.



Additional resources

For more on using containers in the DevOps lifecycle, browse our BMC DevOps Blog or check out these BMC Blogs:

- Container Management Platforms: Which Are Most Popular?
- Container Security Best Practices
- How Containers Fit in a DevOps Delivery Pipeline
- Containers in the Multi-Cloud
- Containers Aren't Always the Solution