

[Serverless Computing](https://cloud.google.com/products/serverless/) (https://cloud.google.com/products/serverless/)

[Cloud Run: Serverless Computing](https://cloud.google.com/run/) (https://cloud.google.com/run/)

[Documentation](https://cloud.google.com/run/docs/) (https://cloud.google.com/run/docs/) [Guides](#)

# Quickstart: Deploy to Cloud Run for Anthos on Google Cloud

This page shows you how to create a cluster enabled for Cloud Run for Anthos on Google Cloud and then deploy a prebuilt sample container to the cluster.

You can also follow this quickstart with a demo account [on Qwiklabs](https://www.qwiklabs.com/focuses/5147?parent=catalog) (https://www.qwiklabs.com/focuses/5147?parent=catalog).

## Before you begin

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Google Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[GO TO THE PROJECT SELECTOR PAGE](https://console.cloud.google.com/projectselect) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECT)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](https://cloud.google.com/billing/docs/how-to/modify-project) (https://cloud.google.com/billing/docs/how-to/modify-project).
4. [Install cURL](https://curl.haxx.se/) (https://curl.haxx.se/) if you want to send a request to the deployed sample service.
5. Set up the `gcloud` command line, described next.

## Setting up the `gcloud` command line

To set up the `gcloud` command line for Cloud Run for Anthos on Google Cloud:

1. Install and initialize Cloud SDK (<https://cloud.google.com/sdk/docs/>).
2. Set the platform to `gke` and set your default project setting for `gcloud` to the one you just created:

```
gcloud config set run/platform gke
gcloud config set project PROJECT-ID
```

Replace **PROJECT-ID** with the project ID of the project you created.

3. Set the desired zone (<https://cloud.google.com/compute/docs/regions-zones/#available>) for the new cluster. You can use any zone where GKE is supported, for example:

```
gcloud config set compute/zone ZONE
```

Replace **ZONE** with your zone.

4. Enable the following APIs for the project, which are needed to create a cluster, build and publish a container into the Google Container registry:

```
gcloud services enable container.googleapis.com containerregistry.googleapis.com
```

5. Update installed `gcloud` components:

```
gcloud components update
```

## Creating a GKE cluster with Cloud Run enabled

To create a cluster and enable it for Cloud Run for Anthos on Google Cloud:

1. Create a new cluster using the command:

```
gcloud container clusters create CLUSTER_NAME \  
  --addons=HorizontalPodAutoscaling,HttpLoadBalancing,CloudRun \  
  --machine-type=n1-standard-4 \  
  --enable-stackdriver-kubernetes \  
  --zone=ZONE \  
  --scopes cloud-platform
```

Replace

- **CLUSTER\_NAME** with the name you want for your cluster.
- **ZONE** with the zone you are using for your cluster, for example, `us-central1-a`. You can use any zone supported by GKE.

Note that although these instructions don't enable [cluster autoscaling](https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-autoscaler) (<https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-autoscaler>) to resize clusters for demand, Cloud Run for Anthos on Google Cloud automatically scales instances within the cluster.

2. Wait for the cluster creation to complete. During the creation process, you should see messages similar to the following:

```
Creating cluster my-cluster...done.  
Created [https://container.googleapis.com/v1beta1/ \\  
projects/my-project/zones/us-central1-b/clusters/my-cluster].
```

where `my-project` is your own project ID. You have just created a new Google Kubernetes Engine cluster named `my-cluster` in the project `my-project`

3. Set `gcloud` defaults to use your new cluster and cluster location, to avoid having to specify these when you use the `gcloud` command line:

```
gcloud config set run/cluster CLUSTER_NAME  
gcloud config set run/cluster_location us-central1-a
```

Replace **CLUSTER\_NAME** with the name you used for your cluster, and if necessary replace `us-central1-a` with the supported cluster location of your choice.

## Deploying a sample container

To deploy a container to the cluster you just created:

1. [GO TO CLOUD RUN](https://console.cloud.google.com/run) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/RUN](https://console.cloud.google.com/run))
2. Click **Create service** to display the *Create service* form. In the form,
  - a. Use the sample `gcr.io/cloudrun/hello` as the container image.
  - b. Select Cloud Run for Anthos on Google Cloud as your development platform.

- c. Use the name *hello* as the service name. You can use some other name, but if you do, make sure you use that name later when you use the cURL command to access the service (#accessing). These instructions assume you use *hello*.
  - d. In the available clusters dropdown menu, select the cluster you just created in the previous steps.
  - e. Select **External** under *Connectivity*, so you can invoke the service.
3. Click **Create** and wait for the deployment to finish. This creates the service and deploys it to Cloud Run for Anthos on Google Cloud.

Congratulations! You have just deployed a service to a Cloud Run for Anthos on Google Cloud enabled cluster.

## Accessing your deployed service

After you deploy your service, you can use cURL to send a request and verify the service is working, using the external IP address of the Load Balancer service created for the Istio ingress controller:

1. Go to the Google Kubernetes Engine page in the Cloud Console:

**[GO TO GOOGLE KUBERNETES ENGINE](https://console.cloud.google.com/kubernetes)** (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/KUBERNETES)

2. Click **Services** in the left navigation panel to display a list of services.
3. If your cluster version is `1.15.3-gke.19` or greater, `1.14.3-gke.12` or greater, or `1.13.10-gke.8` or greater, scroll down to the `istio-ingress` service. For all other cluster versions, scroll down to the `istio-ingressgateway` service.
4. Copy the external IP address shown next to the Load Balancer, WITHOUT THE PORT setting, if there is one. For example, if the IP is `XX.XXX.XX.XXX:15020`, omit the `:15020`. Ignore the other IP addresses listed.
5. Invoke cURL, making sure to replace *hello* in the Host string with the name of your service, if you used a service name other than *hello*:

```
curl -v -H "Host: hello.default.example.com" YOUR-IP
```



Replace **YOUR-IP** with the IP address you obtained in the previous step.

## Clean up

### Disabling Cloud Run for Anthos on Google Cloud

To disable Cloud Run for Anthos on Google Cloud in your cluster:

1. Go to the Google Kubernetes Engine page in the Cloud Console:

**[GO TO GOOGLE KUBERNETES ENGINE](https://console.cloud.google.com/kubernetes/lis)** ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/KUBERNETES/LIS](https://console.cloud.google.com/kubernetes/lis))

2. Click the cluster where you want to disable Cloud Run for Anthos on Google Cloud .
3. Click **Edit**.
4. Scroll down to *Anthos Features* and from the *Cloud Run for Anthos* dropdown, select **Disable**.
5. Click **Save**.

## What's next

To learn how to build a container from code source, push to Container Registry, and deploy, see:

- [Building Containers](https://cloud.google.com/run/docs/building/containers) (<https://cloud.google.com/run/docs/building/containers>)

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated December 9, 2019.*