

[Solutions](https://cloud.google.com/solutions/) (<https://cloud.google.com/solutions/>) [Solutions](#)

Deploying internal services using Cloud Run for Anthos deployed on GKE

This tutorial demonstrates how to expose services deployed to [Cloud Run for Anthos deployed on GKE](https://cloud.google.com/run/docs/quickstarts/prebuilt-deploy-gke) (<https://cloud.google.com/run/docs/quickstarts/prebuilt-deploy-gke>) on your internal network. This type of configuration allows other resources in your network to communicate with the service using a private, internal ([RFC 1918](https://tools.ietf.org/html/rfc1918) (<https://tools.ietf.org/html/rfc1918>)) IP address. Exposing services on an internal network is useful for enterprises that provide internal apps to their staff, and for services that are used by clients that run outside the Cloud Run for Anthos deployed on GKE cluster.

[Cloud Run for Anthos deployed on GKE](https://cloud.google.com/run/) (<https://cloud.google.com/run/>) provides a developer-focused experience for deploying and serving apps and functions running on GKE. By default, Cloud Run for Anthos deployed on GKE exposes services outside the cluster on a public IP address by using Istio's [ingress gateway](https://istio.io/docs/tasks/traffic-management/ingress/) (<https://istio.io/docs/tasks/traffic-management/ingress/>). This gateway is a Kubernetes service of type [LoadBalancer](https://kubernetes.io/docs/concepts/services-networking/service/#loadbalancer) (<https://kubernetes.io/docs/concepts/services-networking/service/#loadbalancer>), which means it's exposed on a public IP address using [Network Load Balancing](https://cloud.google.com/load-balancing/docs/network/) (<https://cloud.google.com/load-balancing/docs/network/>).

Istio also provides an internal load balancing (ILB) gateway. This gateway provides a way for other resources in the same region to access your services by using an internal IP address in your VPC network with [Internal TCP/UDP Load Balancing](https://cloud.google.com/load-balancing/docs/internal/) (<https://cloud.google.com/load-balancing/docs/internal/>). The Istio add-on for GKE doesn't install the ILB gateway, but you can add it as an extra component. This tutorial shows you how to use this ILB gateway for services deployed to Cloud Run for Anthos deployed on GKE.

Objectives

- Create a GKE cluster with Cloud Run enabled.
- Install the Istio ILB gateway.
- Update Cloud Run for Anthos deployed on GKE to use the ILB gateway.
- Test the app by deploying a sample service to Cloud Run for Anthos deployed on GKE.

Costs

This tutorial uses the following billable components of Google Cloud:

- [Cloud Run for Anthos deployed on GKE](https://cloud.google.com/run/pricing#cloudrun-gke-pricing)
(<https://cloud.google.com/run/pricing#cloudrun-gke-pricing>)
- [Compute Engine](https://cloud.google.com/compute/pricing) (<https://cloud.google.com/compute/pricing>)
- [Stackdriver](https://cloud.google.com/stackdriver/pricing) (<https://cloud.google.com/stackdriver/pricing>)

To generate a cost estimate based on your projected usage, use the [pricing calculator](https://cloud.google.com/products/calculator) (<https://cloud.google.com/products/calculator>). New Google Cloud users might be eligible for a [free trial](https://cloud.google.com/free-trial) (<https://cloud.google.com/free-trial>).

When you finish this tutorial, you can avoid continued billing by deleting the resources you created. For more information, see [Cleaning up](#) (#clean-up).

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/projectselector) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECTOR](https://console.cloud.google.com/projectselector))

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. In the Cloud Console, go to Cloud Shell.

GO TO CLOUD SHELL (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/?CLOUDSHELL=TRUE)

At the bottom of the Cloud Console, a [Cloud Shell](https://cloud.google.com/shell/docs/features) (https://cloud.google.com/shell/docs/features) session opens and displays a command-line prompt. Cloud Shell is a shell environment with the Cloud SDK already installed, including the [gcloud](https://cloud.google.com/sdk/gcloud/) (https://cloud.google.com/sdk/gcloud/) command-line tool, and with values already set for your current project. It can take a few seconds for the session to initialize.

You run all commands in this tutorial from Cloud Shell.

5. In Cloud Shell, enable the Cloud Run API, GKE API, and Cloud APIs:

```
gcloud services enable \  
  cloudapis.googleapis.com \  
  container.googleapis.com \  
  run.googleapis.com
```



Setting up the environment

- In Cloud Shell, define environment variables and the `gcloud` tool defaults for the Compute Engine zone and GKE cluster name that you want to use for this tutorial:

```
ZONE=us-central1-f  
CLUSTER=cloudrun-gke-ilb-tutorial  
  
gcloud config set compute/zone $ZONE  
gcloud config set run/cluster $CLUSTER  
gcloud config set run/cluster_location $ZONE
```



The examples in this tutorial use `us-central1-f` as the zone and `cloudrun-gke-ilb-tutorial` as the cluster name. You can use different values. For more information, see [Geography and regions](https://cloud.google.com/docs/geography-and-regions) (https://cloud.google.com/docs/geography-and-regions).

Creating a GKE cluster with Cloud Run enabled

1. In Cloud Shell, create a GKE cluster with the Cloud Run and Istio add-ons:

```
gcloud beta container clusters create $CLUSTER \  
  --machine-type n1-standard-4 \  
  --enable-stackdriver-kubernetes \  
  --enable-ip-alias \  
  --addons HttpLoadBalancing,Istio,CloudRun
```



2. Add yourself as a cluster admin so that you can install extra Istio components:

```
kubectl create clusterrolebinding cluster-admin-binding \  
  --clusterrole cluster-admin \  
  --user $(gcloud config get-value core/account)
```



Installing the Istio ILB gateway

1. In Cloud Shell, inspect your GKE cluster to find the version of Istio used:

```
ISTIO_PACKAGE=$(kubectl -n istio-system get deployments istio-pilot \  
  -o jsonpath="{.spec.template.spec.containers[0].image}" | \  
  cut -d':' -f2)  
  
ISTIO_VERSION=$(echo $ISTIO_PACKAGE | cut -d'-' -f1)
```



2. Download and extract Istio:

```
wget https://github.com/istio/istio/releases/download/$ISTIO_VERSION/istio-$ISTIO_VERSION-linux.tar.gz  
tar xzf istio-$ISTIO_VERSION-linux.tar.gz
```



3. Download and extract Helm:

```
HELM_VERSION=v2.14.3  
  
wget https://storage.googleapis.com/kubernetes-helm/helm-$HELM_VERSION-linux-amd64.tar.gz  
tar xzf helm-$HELM_VERSION-linux-amd64.tar.gz
```



4. Use [Helm's local template rendering](https://helm.sh/docs/helm/#helm-template) (<https://helm.sh/docs/helm/#helm-template>) to create a Kubernetes manifest that installs the Istio ILB gateway, `istio-ilbgateway`:

```
./linux-amd64/helm template \  
  --set galley.enabled=false \  
  --set gateways.enabled=true \  
  --set gateways.istio-ingressgateway.enabled=false \  
  --set gateways.istio-egressgateway.enabled=false \  
  --set gateways.istio-ilbgateway.enabled=true \  
  --set gateways.istio-ilbgateway.ports[0].name=https \  
  --set gateways.istio-ilbgateway.ports[0].port=443 \  
  --set gateways.istio-ilbgateway.ports[1].name=http \  
  --set gateways.istio-ilbgateway.ports[1].port=80 \  
  --set global.hub=gcr.io/gke-release/istio \  
  --set global.omitSidecarInjectorConfigMap=true \  
  --set global.tag=$ISTIO_PACKAGE \  
  --set mixer.enabled=false \  
  --set mixer.policy.enabled=false \  
  --set mixer.telemetry.enabled=false \  
  --set pilot.enabled=false \  
  --set prometheus.enabled=false \  
  --set security.enabled=false \  
  --set sidecarInjectorWebhook.enabled=false \  
  --namespace istio-system \  
istio-$ISTIO_VERSION/install/kubernetes/helm/istio \  
> istio-ilbgateway.yaml
```

This command sets several template flags to `false` to create a manifest file that contains only the objects required to add the ILB gateway to an existing Istio installation. Applying this manifest file doesn't disable any existing Istio components.

5. Apply the ILB gateway manifest file:

```
kubectl apply -f istio-ilbgateway.yaml
```

Note: When using the Istio add-on for GKE, upgrades to the Istio control plane components installed by the add-on are managed as part of the [GKE upgrade process](https://cloud.google.com/istio/docs/istio-on-gke/overview#how_does_the_upgrade_process_work) (https://cloud.google.com/istio/docs/istio-on-gke/overview#how_does_the_upgrade_process_work). These upgrades don't include extra Istio components installed manually, such as the ILB gateway. Repeat the steps in this section to upgrade the ILB gateway when GKE upgrades the version of Istio in your cluster.

Creating TLS certificate for the ILB gateway

1. In Cloud Shell, create a self-signed Transport Layer Security (TLS) certificate and private key to allow TLS termination by the ILB gateway:

```
openssl req -x509 -nodes -newkey rsa:2048 -days 365 \  
  -keyout privkey.pem -out cert.pem -subj "/CN=*.default.example.com"
```

2. Create a Kubernetes secret called `istio-ilbgateway-certs` to store the TLS certificate and private key:

```
kubectl -n istio-system create secret tls istio-ilbgateway-certs \  
  --key privkey.pem --cert cert.pem \  
  --dry-run -o yaml | kubectl apply -f -
```

Note: In this tutorial, you use a self-signed certificate. This approach isn't recommended in a production environment. You should instead obtain a certificate from a trusted certificate authority.

Configuring Cloud Run for Anthos deployed on GKE to use the ILB gateway

1. In Cloud Shell, create a patch file to configure TLS settings and to update `knative-ingress-gateway` to point to the internal Istio `ilbgateway` instead of the external Istio `ingressgateway`:

```
cat << EOF > knative-ingress-gateway-patch.yaml  
spec:  
  selector:  
    istio: ilbgateway  
  servers:  
  - hosts:  
    - '*'  
    port:  
      name: http  
      number: 80  
      protocol: HTTP  
  - hosts:  
    - '*'  
    port:  
      name: https  
      number: 443
```

```
    protocol: HTTPS
  tls:
    mode: SIMPLE
    privateKey: /etc/istio/ilbgateway-certs/tls.key
    serverCertificate: /etc/istio/ilbgateway-certs/tls.crt
EOF
```

2. Apply the patch:

```
kubectl -n knative-serving patch gateway knative-ingress-gateway \
  --type merge -p "$(cat knative-ingress-gateway-patch.yaml)"
```



Test the service

1. In Cloud Shell, deploy a service called `sample` to Cloud Run for Anthos deployed on GKE in the `default` namespace:

```
gcloud beta run deploy sample \
  --image gcr.io/knative-samples/simple-api \
  --namespace default \
  --platform gke
```



2. Create a Compute Engine virtual machine (VM) instance in the same zone as the GKE cluster:

```
VM=cloudrun-gke-ilb-tutorial-vm

gcloud compute instances create $VM
```



3. Store the ILB gateway IP address in an environment variable called `ILB_IP` and a file called `ilb-ip.txt`:

```
export ILB_IP=$(kubectl -n istio-system get service istio-ilbgateway \
  -o jsonpath='{.status.loadBalancer.ingress[0].ip}' | tee ilb-ip.txt)
```



4. Copy the file containing the IP address of the ILB gateway to the VM:

```
gcloud compute scp ilb-ip.txt $VM:~
```



5. Connect to the instance using SSH:

```
gcloud compute ssh $VM
```



6. While in the SSH session, test the service using HTTPS:

```
curl -s -k -w '\n' \  
-H Host:sample.default.example.com https://$(cat ilb-ip.txt)/
```



The output is as follows:

```
OK
```

Note: You use the `-k` parameter in the preceding command because you used a self-signed certificate in this tutorial.

7. Test the service using HTTP:

```
curl -s -w '\n' \  
-H Host:sample.default.example.com http://$(cat ilb-ip.txt)/
```



8. Leave the SSH session:

```
exit
```



9. Send a request to verify that you can't connect from outside the virtual private cloud (VPC) network:

```
curl -s -v -k --connect-timeout 5 \  
-H Host:sample.default.example.com https://$ILB_IP/
```



The output is similar to the following:

```
* Trying [$ILB_IP]...  
* TCP_NODELAY set  
* Connection timed out after 5003 milliseconds  
* stopped the pause stream!  
* Closing connection 0
```



This connection fails as expected because Cloud Shell runs outside your VPC network.

Troubleshooting

If you run into problems with this tutorial, review the following documents:

- [Cloud Run for Anthos deployed on GKE troubleshooting](https://cloud.google.com/run/docs/gke/troubleshooting) (https://cloud.google.com/run/docs/gke/troubleshooting)
- [GKE troubleshooting](https://cloud.google.com/kubernetes-engine/docs/troubleshooting) (https://cloud.google.com/kubernetes-engine/docs/troubleshooting)
- [Istio Operations Guide](https://istio.io/help/ops/) (https://istio.io/help/ops/)
- [Troubleshooting Kubernetes clusters](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-cluster/) (https://kubernetes.io/docs/tasks/debug-application-cluster/debug-cluster/)

Cleaning up

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial:


Delete the project

Caution: Deleting a project has the following effects:

- **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
- **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such as an **appspot.com** URL, delete selected resources inside the project instead of deleting the whole project.

1. In the Cloud Console, go to the **Manage resources** page.

[GO TO THE MANAGE RESOURCES PAGE](https://console.cloud.google.com/iam-admin/projects) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/IAM-ADMIN/PROJ)

2. In the project list, select the project you want to delete and click **Delete** .

3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

Delete the individual resources

If you want to keep the GCP project you used in this tutorial, delete the individual resources:

1. Delete the GKE cluster:

```
gcloud container clusters delete $CLUSTER --quiet --async
```



2. Delete the Compute Engine instance:

```
gcloud compute instances delete $VM --quiet
```



What's next

- Learn how to [authenticate end users of Cloud Run for Anthos deployed on GKE with Istio and Cloud Identity Platform](https://cloud.google.com/solutions/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>)
- Learn how to [authorize access to Cloud Run for Anthos deployed on GKE using Istio](https://cloud.google.com/solutions/authorizing-access-to-cloud-run-on-gke-services-using-istio) (<https://cloud.google.com/solutions/authorizing-access-to-cloud-run-on-gke-services-using-istio>).
- Understand how to [access Internal TCP/UDP Load Balancing IP addresses from connected networks](https://cloud.google.com/load-balancing/docs/internal/internal-lb-and-other-networks) (<https://cloud.google.com/load-balancing/docs/internal/internal-lb-and-other-networks>).
- Read [Cloud Run how-to guides](https://cloud.google.com/run/docs/how-to) (<https://cloud.google.com/run/docs/how-to>).
- Explore [Knative](https://www.knative.dev/) (<https://www.knative.dev/>), the open source project that underpins Cloud Run for Anthos deployed on GKE.
- Try out other Google Cloud Platform features for yourself. Have a look at our [tutorials](https://cloud.google.com/docs/tutorials) (<https://cloud.google.com/docs/tutorials>).

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>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated September 25, 2019.