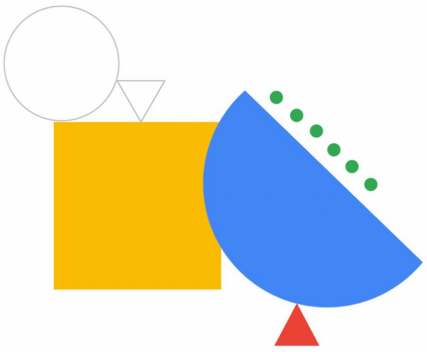
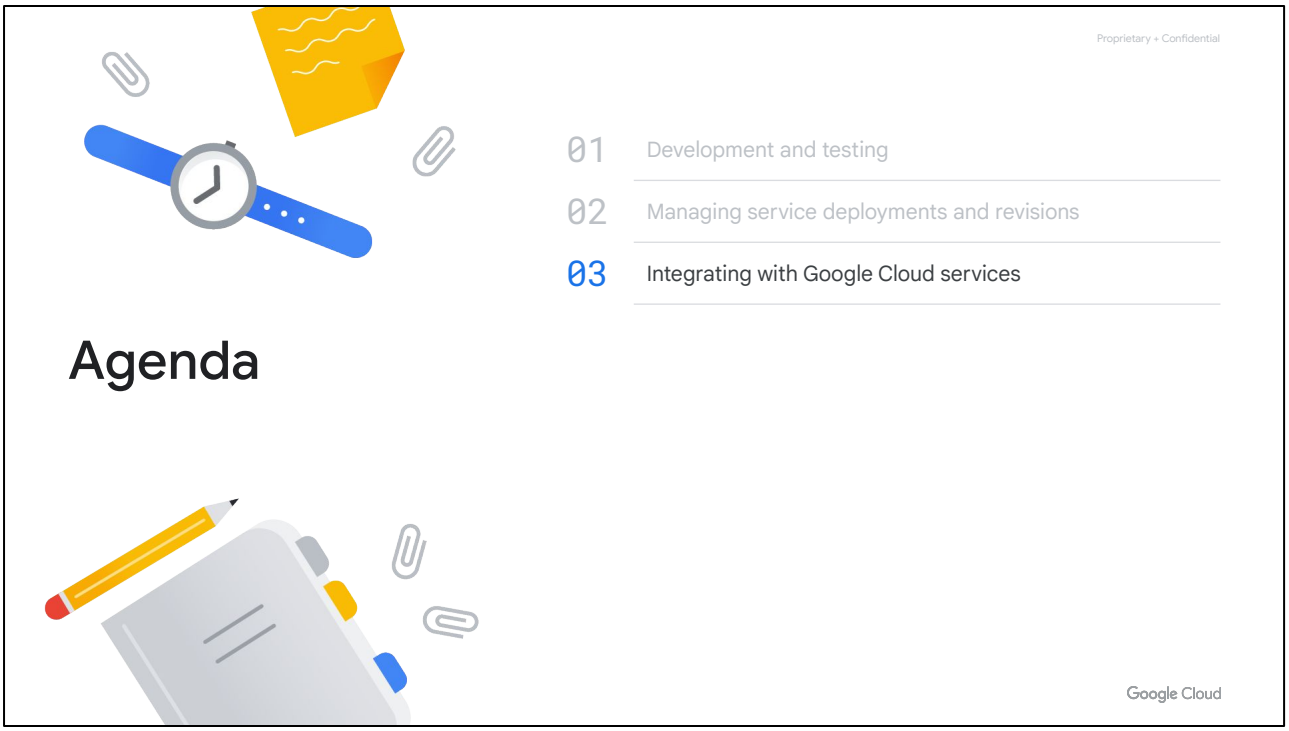
Google Cloud

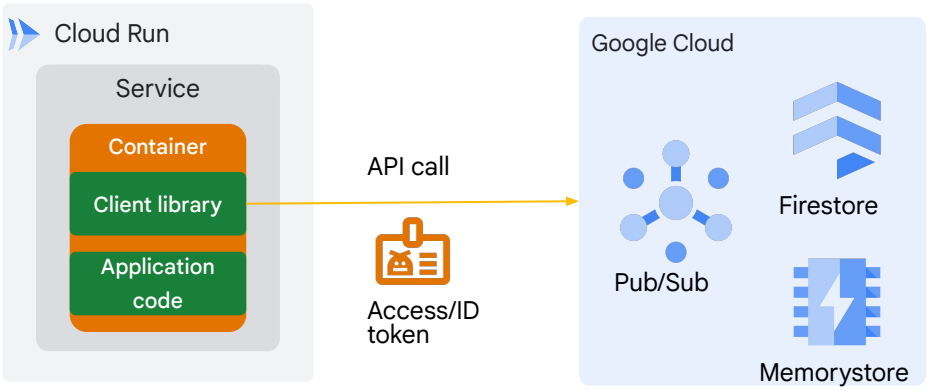
Application Development, Testing, and Integration





In this topic, we briefly discuss how you can connect and integrate your Cloud Run service with other services in Google Cloud.

Connect to Google Cloud services



You can use Cloud Run to connect to supported Google Cloud services from your application using the client libraries that those services provide. The client libraries use the built-in service account to transparently authenticate with Google Cloud services. This service account that has the *Project* > *Editor* role, which means it's able to call all Google Cloud APIs and have read and write access on all resources in your Google Cloud project.

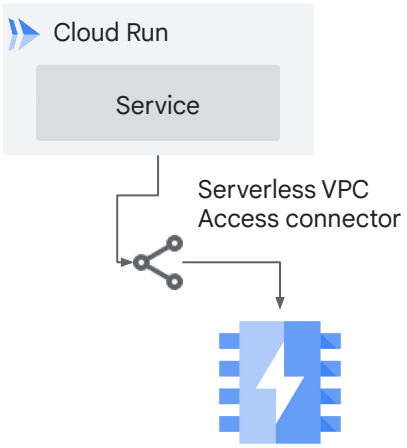
As mentioned previously, you should use a per-service identity to restrict the APIs and resources that your Cloud Run service can access. You can do this by assigning a service account with a minimal set of permissions to your Cloud Run service. For example, if your Cloud Run service is only reading data from Firestore, you should assign it a service account that only has the *Firestore User* IAM role.

For a full list of Google Cloud services with which you can integrate your Cloud Run service, refer to the documentation.

Connect Cloud Run to Memorystore

To access Memorystore from Cloud Run:

1. Determine your Redis instance's authorized VPC network.
2. Create a Serverless VPC Access connector.
3. Attach the connector to the Redis instance's authorized VPC network.



Let's review how you can connect to a Memorystore for Redis instance from your Cloud Run service.

Memorystore is a Google Cloud service that provides a highly available, scalable, and secure in-memory cache solution for Redis and Memcached.

To connect to a Memorystore for Redis instance from your Cloud Run service, you use Serverless VPC Access.

To connect to your Redis instance, your Cloud Run service needs access to the Redis instance's authorized VPC network. To enable this access, you need a Serverless VPC Access connector.

Once you've determined your Redis instance's authorized VPC network, create a Serverless VPC Access connector in the same region as your Cloud Run service. Then, attach the connector to the Redis instance's authorized VPC network.

Read the documentation to learn about creating a Serverless VPC Access connector.

Connect Cloud Run to Memorystore

Deploy the service:

- Specify the name of the connector.
- Set environment variables for the service application code to connect to the Redis instance's host and port.

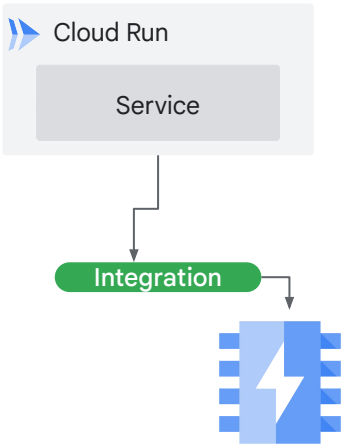
```
gcloud run deploy \
--image my-container-image \
--platform managed \
--region us-central1 \
--vpc-connector my-connector \
--set-env-vars \
REDISHOST=[REDIS_IP],REDISPORT=[REDIS_PO
RT]
```


Deploy the service to Cloud Run specifying the name of the connector, and environment variables for the Redis instance's host IP address and port.

Your application code can then use these environment variables to instantiate a client that connects to the Redis instance.

Cloud Run integrations

1. Create an integration to connect a Cloud Run service to a Memorystore for Redis cache.
2. A fully configured Redis cache is automatically created with a configurable memory size.
3. A new Cloud Run service revision is created for the service.
4. Networking and environment variables are also configured for the service to access the Redis cache.



The integrations feature provides a simple Google Cloud console UI and gcloud CLI command that create and configure the resources and services needed for the specific integration, eliminating the complicated steps that would otherwise be required.

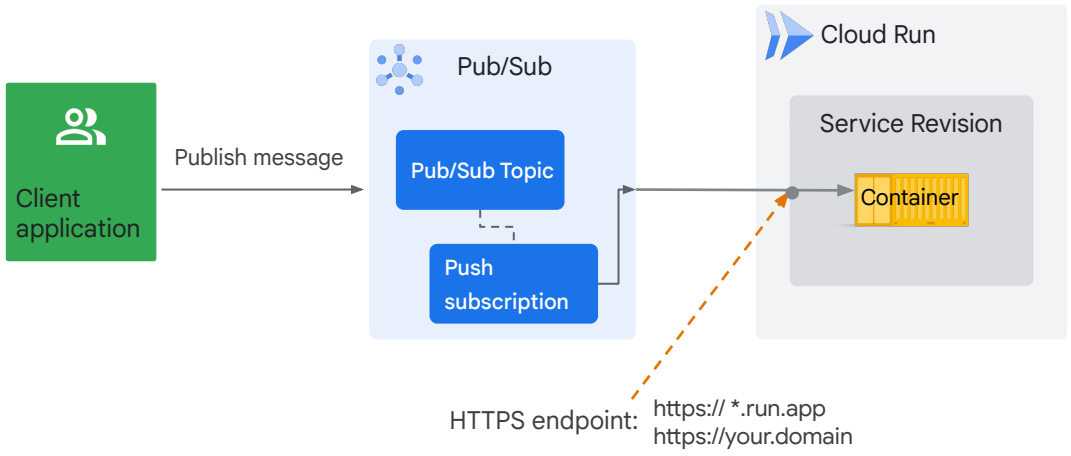
Integrations currently let you:

- Map custom domains to Cloud Run services.
- Connect a Cloud Run service to a Memorystore for Redis instance.

There are more integrations that will be supported in the future.

For more information, refer to the documentation on connecting to a Memorystore instance from Cloud Run using integrations.

Trigger from Pub/Sub

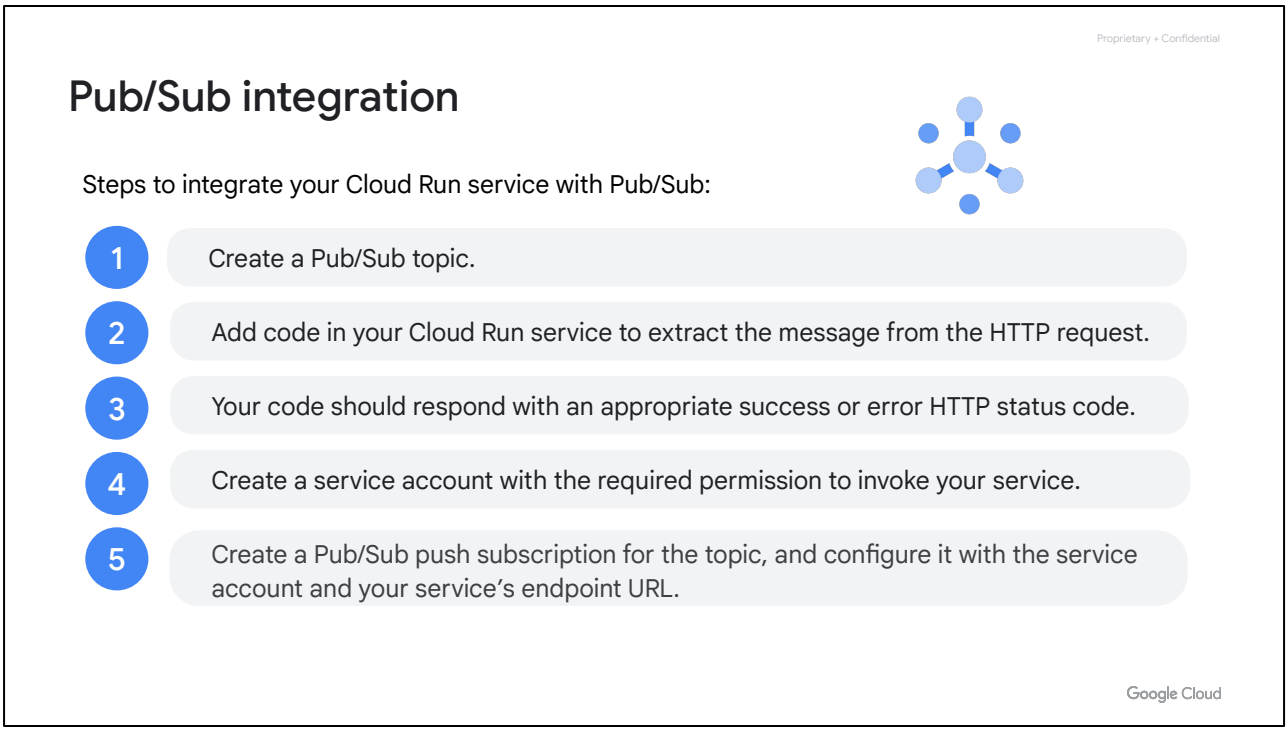


We discussed how you can publish a message to a Pub/Sub topic from a Cloud Run service in the previous module.

Let's talk about how you can trigger a Cloud Run service by a message from a topic in Pub/Sub.

You can use Pub/Sub to *push* messages to the endpoint of your Cloud Run service, where the messages are subsequently delivered to containers as HTTP requests. The endpoint can be protected using IAM and doesn't need to be public.

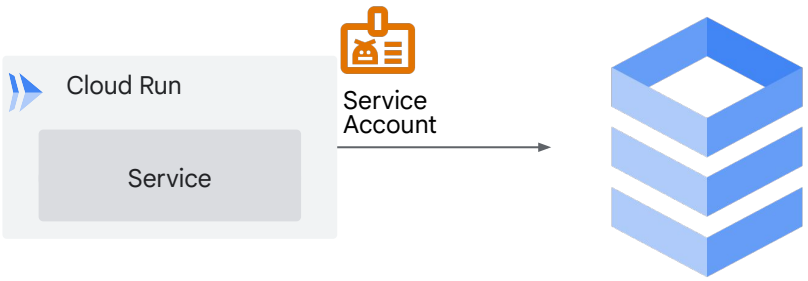
Your Cloud Run service must acknowledge the Pub/Sub message by returning a response within 600 seconds (maximum acknowledgement deadline), otherwise Pub/Sub will redeliver the message, causing a your Cloud Run service to be triggered again.



To integrate your service with Pub/Sub:

- Create a Pub/Sub topic.
- Add code in your Cloud Run service to respond to the Pub/Sub messages sent to the topic that you created.
    a. Your service must extract the message from the request and return an expected HTTP status response code.
    b. Success codes, such as HTTP 200 or 204, acknowledge complete processing of the Pub/Sub message.
    c. Error codes, such as HTTP 400 or 500, indicate that the message will be retried.
- Create a service account with the required permission (role: Cloud Run Invoker) to invoke your Cloud Run service.
- Create a Pub/Sub push subscription for the topic that you created, and associate it with the service account. Provide your service's URL as the endpoint URL. This subscription will send any message that is published to the topic to your service.

Connect Cloud Run to Cloud SQL




```
gcloud run services update my-service
--add-cloudsql-instances=my-sql-instance-connection
```

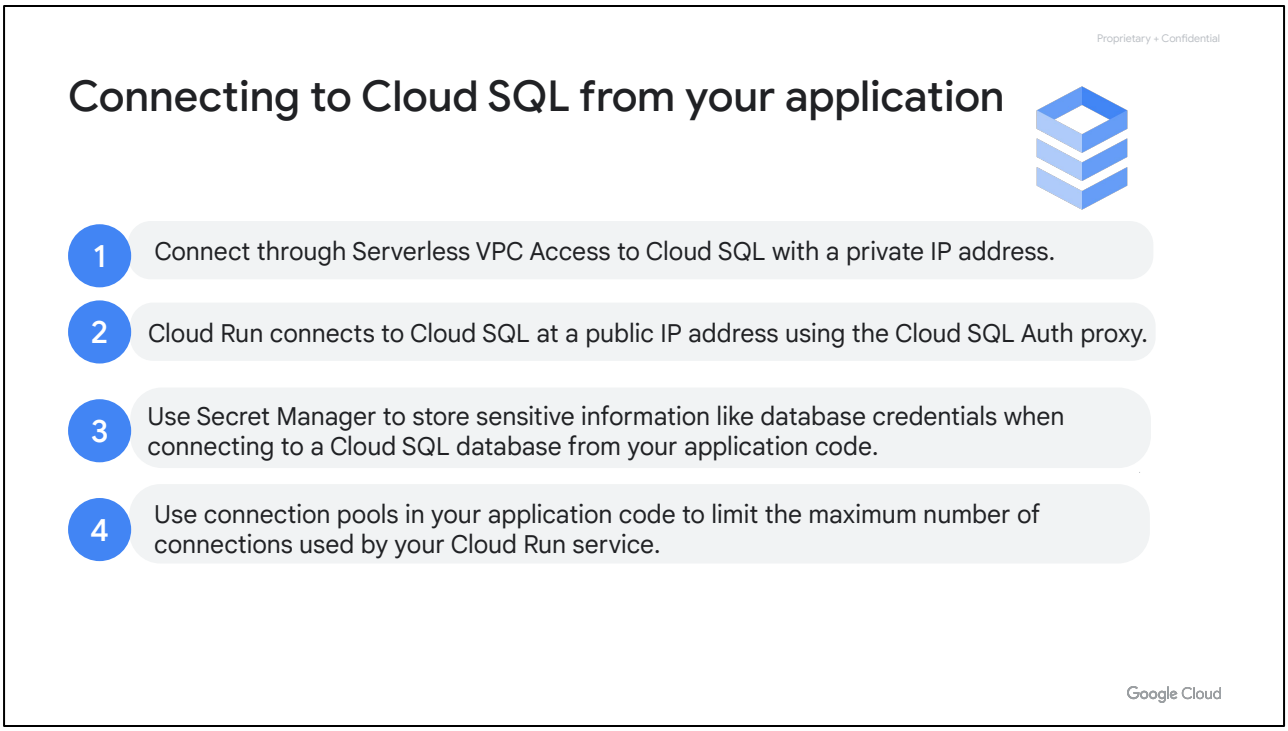

Let's now review how you can connect to a Cloud SQL instance from a service running in Cloud Run.

Cloud SQL is a fully-managed database service for MySQL, PostgreSQL, and SQL Server that lets you set up, manage, and administer relational databases in Google Cloud.

By default, Cloud SQL assigns a public IP address when you create a new instance. To connect to the instance from your Cloud Run service:

- The service account used by the service must have the appropriate Cloud SQL roles and permissions. (one of Cloud SQL Client, Cloud SQL Admin)
- Deploy or update your Cloud Run service with the instance connection name of your Cloud SQL instance. You can do this in the Google Cloud console, with the gcloud CLI, or with Terraform.

You also have the option to assign a private IP address to your Cloud SQL instance. With a private IP address, you can route all egress traffic from your Cloud Run service to the Cloud SQL instance using a Serverless VPC Access connector. To do this, you configure your Cloud Run service to use the connector when you deploy or update the service.



As mentioned previously, when connecting to a Cloud SQL instance with a private IP address, your application will connect directly through Serverless VPC Access.

For the public IP path, Cloud Run provides encryption and connects using the Cloud SQL Auth proxy using network sockets or a Cloud SQL connector.

Cloud SQL connectors are language specific libraries that provide encryption and IAM-based authorization when connecting to a Cloud SQL instance.

Your application code will need access to the Cloud SQL instance connection name, database name, and credentials. It's recommended to use Secret Manager to store the sensitive database credentials, and pass this information to your application as environment variables or mounted as a volume in Cloud Run.

You should also use a client library that supports connection pools that automatically reconnect broken client connections to the Cloud SQL database.

By using a connection pool, you can also limit the maximum number of connections used by the service. Cloud Run services are limited to 100 connections per service to a Cloud SQL database. There are also quotas and limits imposed by Cloud SQL. For more information, refer to the documentation.

Remember

1. Connect to supported Google Cloud services from your Cloud Run application using client libraries.
2. Use a per-service identity to restrict the APIs and resources that your Cloud Run service can access.
3. Connect to a Memorystore for Redis instance from your Cloud Run service with Serverless VPC Access.
4. Pub/Sub messages are delivered to your container in Cloud Run with HTTP requests.
5. Use Secret Manager to store sensitive database credentials, and pass them to your service as environment variables or mounted as a volume in Cloud Run.

In summary:

- Use Cloud Run to connect to supported Google Cloud services from your application using the client libraries that those services provide.
- Use a per-service identity (service account) to restrict the APIs and resources that your Cloud Run service can access.
- Connect to a Memorystore for Redis instance from your Cloud Run service with Serverless VPC Access.
- Use Pub/Sub to *push* messages to the endpoint of your Cloud Run service, where the messages are then delivered to containers as HTTP requests.
- Use Secret Manager to store sensitive database credentials, and pass this information to your service as environment variables or mounted as a volume in Cloud Run.