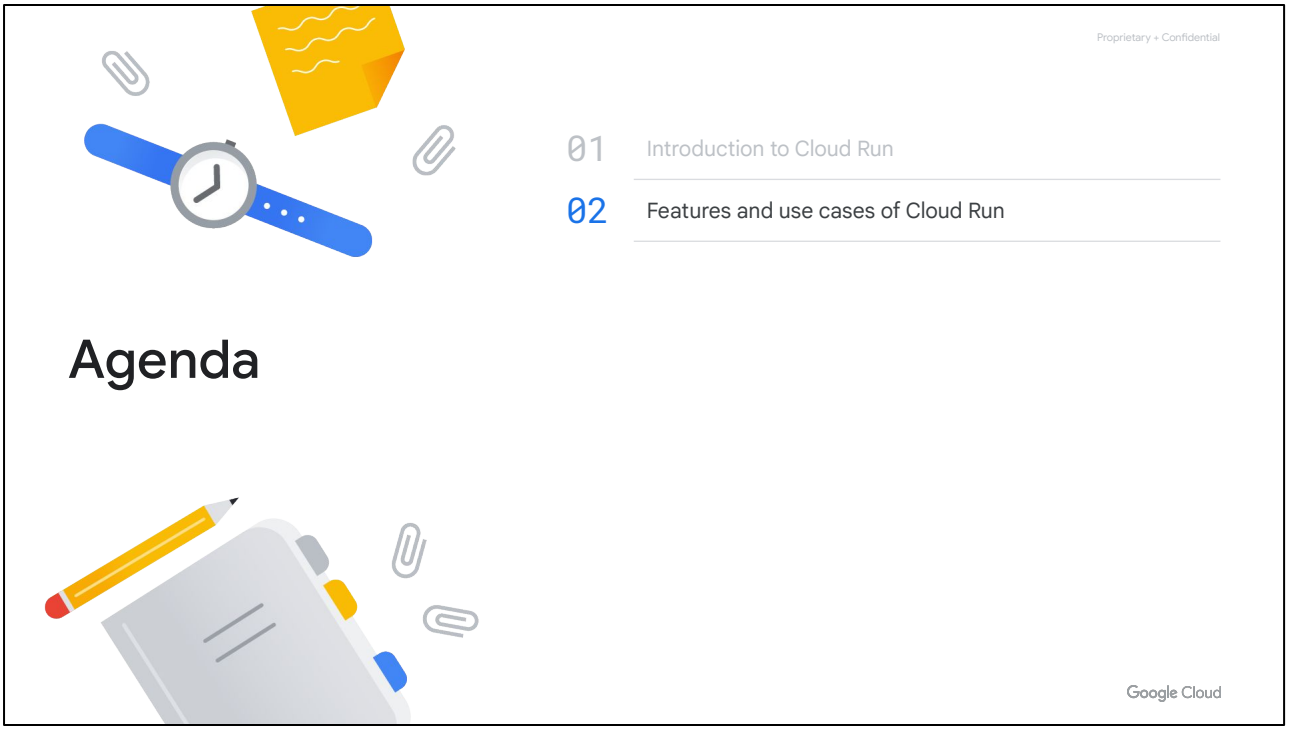
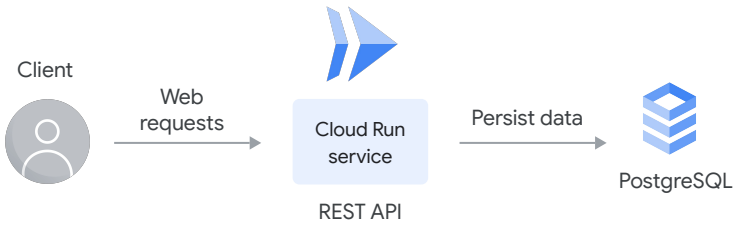


Let’s now discuss some of the use cases for Cloud Run.

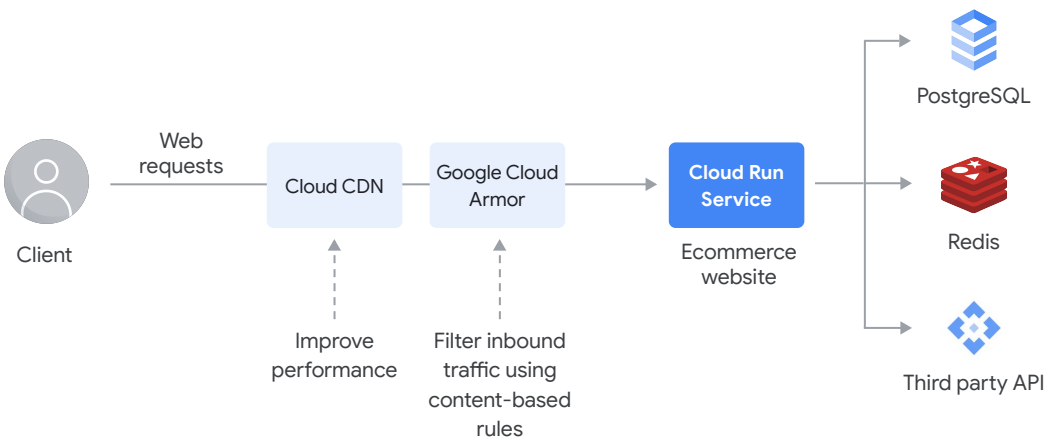
Serving a REST API with Cloud Run



A common use case for Cloud Run is to deploy a service that provides a REST API. You can use the service to provide an API, a website, or a web application.

If required, you can connect the service to a database to persist data handled by the API or web application.

An ecommerce site on Cloud Run

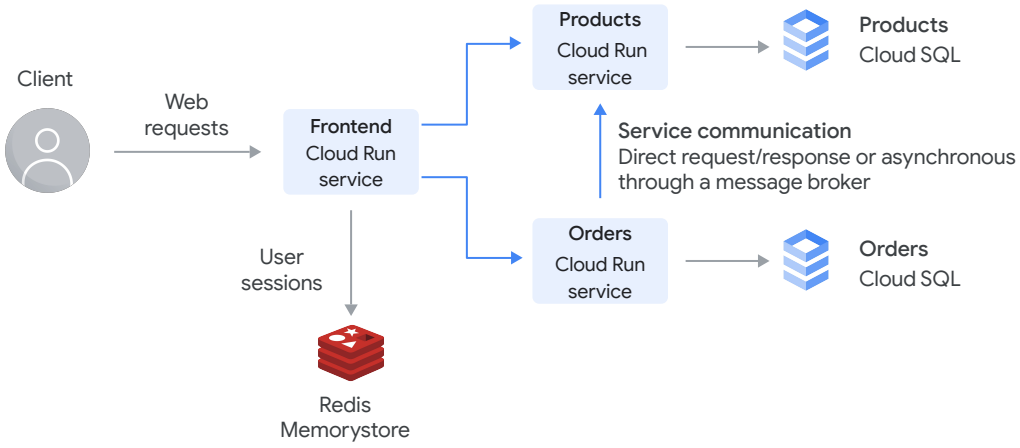


You can build a more complex public website, for example, an e-commerce site on Cloud Run.
In this case, you could also:

- Enable Cloud CDN to improve performance,
- Add Google Cloud Armor to filter malicious inbound traffic using content-based policies.

In the backend, you can connect with a relational database, a Redis store for user sessions, and connect with third-party APIs.

Microservices on Cloud Run

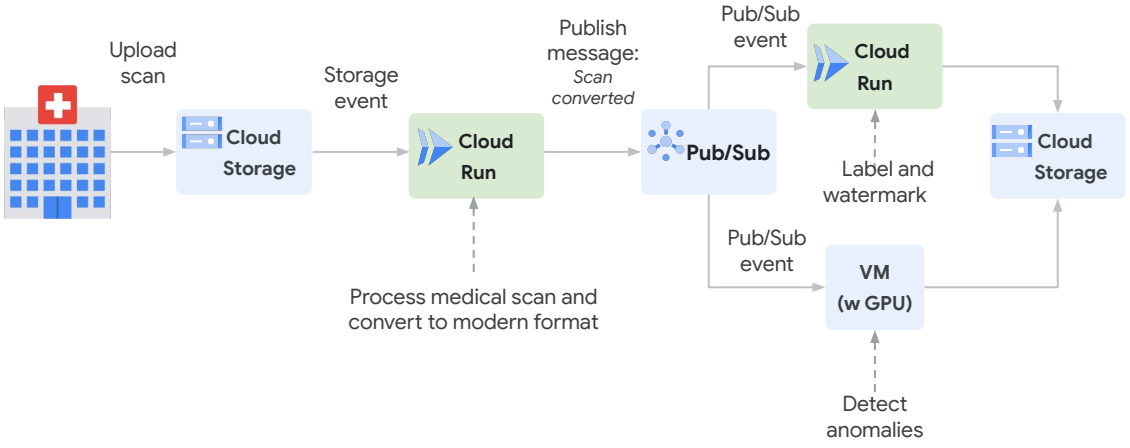


You can deploy and run an application that is composed of many microservices on Cloud Run.

Services on Cloud Run can communicate with each other using REST APIs or gRPC.

Using Pub/Sub, you can send and receive asynchronous messages between services with guaranteed delivery. Pub/Sub is well integrated with Cloud Run using push subscriptions. Pub/Sub forwards and optionally authenticates messages as HTTP requests to the endpoint of your Cloud Run service.

Event processing on Cloud Run

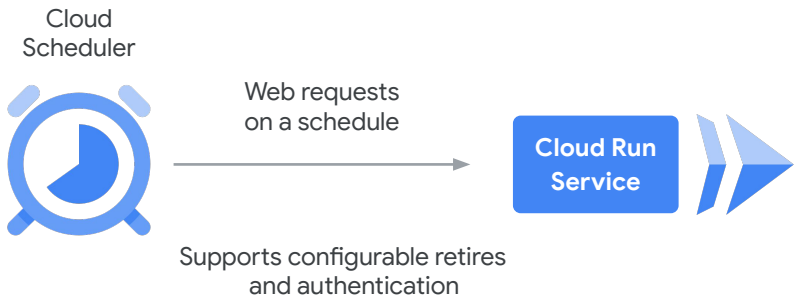


Cloud Run integrates with various Google Cloud services such as Cloud Storage, Cloud Build, Pub/Sub, Eventarc, and others that generate events from your cloud infrastructure.

This enables you to build event processing workflows with Cloud Run. The example of such a workflow is shown.

When an image of a medical scan is uploaded to Cloud Storage, a Cloud Run service is triggered to process the scanned image and convert it into a modern format. The service then pushes a message to Pub/Sub that triggers another Cloud Run service to label and watermark the converted image, and another VM application that detects anomalies in the scan data. Both services generate output that is stored back in Cloud Storage.

Scheduling a Cloud Run service with Cloud Scheduler



You can use Cloud Scheduler to securely trigger a Cloud Run service on a schedule. Cloud Scheduler is a fully-managed cron job scheduler.

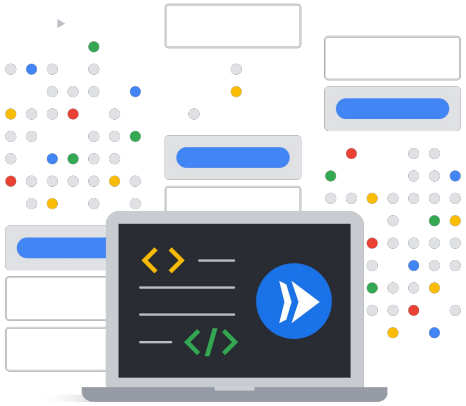
Some examples of scheduled services include generating invoices, or rebuilding a search index.

The limitation of running a scheduled job in the container itself is that the lifetime of a container is only guaranteed while it's handling requests. If you schedule tasks on a container to run later, the container might be shut down or stopped by the time the task has to run.

Note that the Cloud Run service must complete its task within the configured request timeout.

Design HA applications with Cloud Run

1. **Incremental application updates**, switching traffic gradually with easy rollback.
2. **Automatic scaling** of the number of containers to handle all incoming requests.
3. **Load balancing** across zones and regions.

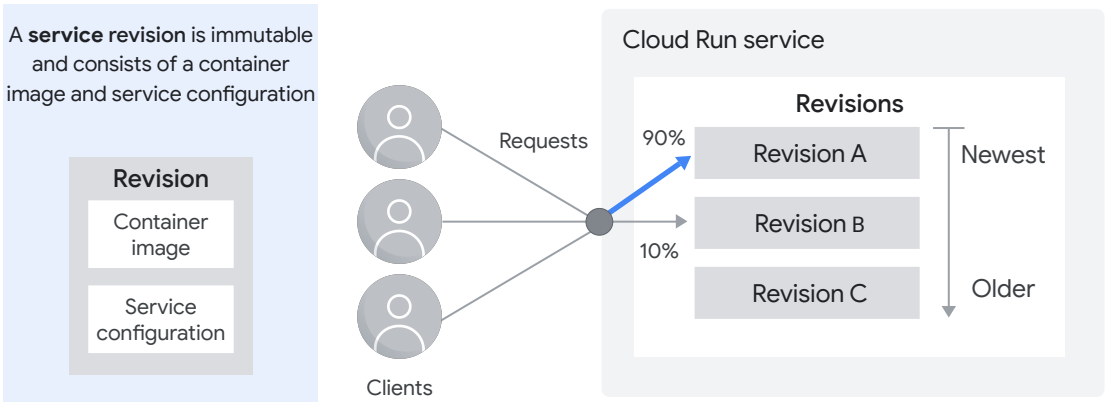


Cloud Run helps you design applications that are highly available. To support high availability, Cloud Run provides:

- Incremental application updates
- Autoscaling
- Load Balancing across zones and regions

Let's review these features in more detail.

Incremental application updates with service revisions



A common cause of service disruptions is often application updates, which affect the availability of your application.

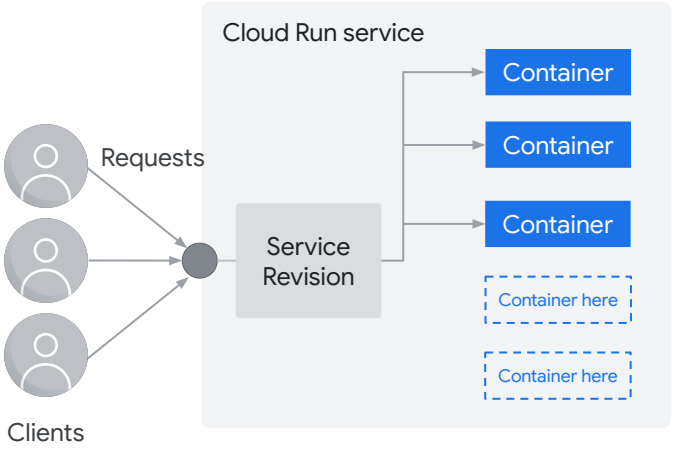
On Cloud Run, each deployment of your container image to a service creates a new revision.
A service revision is immutable and cannot be modified. If you make a change to your application and deploy it, Cloud Run creates a new revision of your service.

A service *revision* consists of:

- Your container image, and
- The service configuration that includes settings such as environment variables, memory limits, and other configuration values.

You can reduce the impact of request processing failures by splitting request traffic between the new and previous revisions of your service, by specifying the percentage of requests that should be sent to the new revision. This lets you roll back to a previous stable revision if there is a high rate of request failures, or gradually send 100% of request traffic to the new revision.

Automatic scaling with Cloud Run



To maintain the capacity to handle incoming requests to your service, Cloud Run automatically increases the number of container instances of a service revision when necessary. This feature is known as autoscaling.

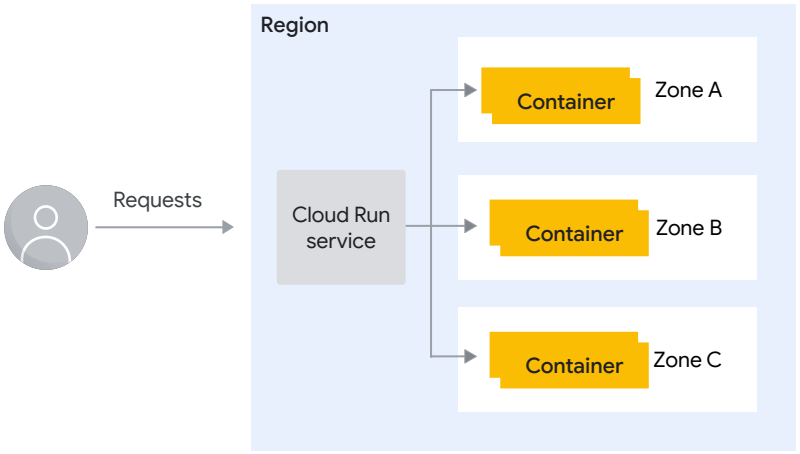
Requests to a service revision are distributed across the group of container instances.

- If all container instances are busy, Cloud Run adds additional instances.
- When demand decreases, Cloud Run stops sending traffic to some instances and shuts them down.
- Note that a container instance can receive many requests at the same time. With the concurrency setting, you can set the maximum number of requests that can be sent in parallel to a given container instance.

In addition to the rate of incoming requests to your service, the number of container instances is impacted by:

- The CPU utilization of existing instances when they are processing requests (with a target of 60% of utilization).
- The maximum concurrency setting.
- The minimum and maximum number of container instances setting.

Regions and zones



A **region** is a geographic location where cloud resources are hosted. (Iowa, North America)

A region has three or more **zones.** A zone is a deployment area for cloud resources within a region.

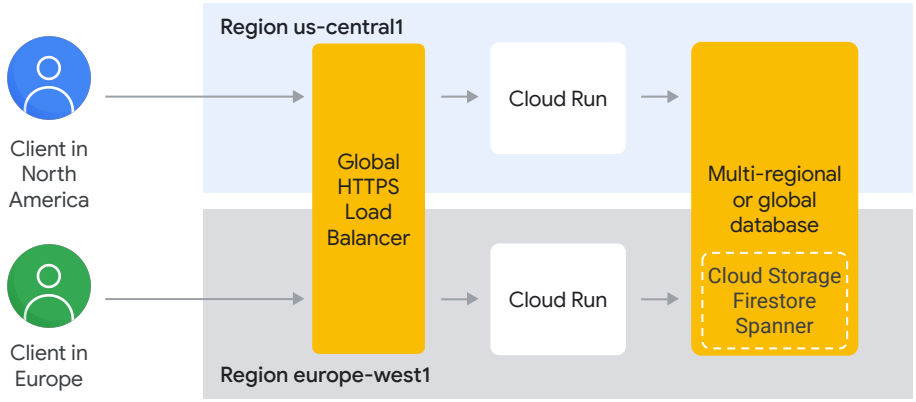
Cloud Run is a regional service that lets you choose a region where your containers are deployed. A region is a specific geographical location where your Google Cloud resources are hosted.

A region consists of three or more zones. Zones and regions are logical abstractions of underlying physical resources that are provided in one or more data centers. An example of a region is *us-central1* in Iowa, North America.

A zone is a deployment area for cloud resources within a region. Zones are considered to single failure domains within a region.

For high availability, Cloud Run distributes your containers over multiple zones in a region, making your application resilient against the failure of a zone.

Global Load Balancing

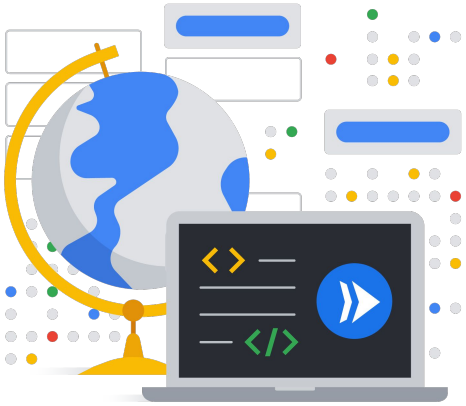


Cloud Run integrates with the Google Cloud external HTTP(S) global load balancer (GLB), which lets you expose a single, global IP address in front of multiple, regional, Cloud Run services.

The global load balancer routes requests from a client to the region closest to them. In addition to improving application availability, the GLB decreases latency for clients worldwide.

Application portability

1. A container image contains your application and **everything** that your application needs to run.
2. Containers are inherently portable and run in any container-based environment.
3. The Cloud Run platform is compatible with **Knative,** which implements the same container runtime contract as Cloud Run.



Portability is important for application developers. Here are a couple of use cases where portability is important:

- The application needs to run in a geographical region where Google Cloud has no physical presence, and you're required to run it there (for data sovereignty).
- The developer wants to avoid vendor lock-in.

Applications on Cloud Run are portable in two ways:

- Cloud Run uses containers. You already learned that containers can run anywhere, which makes your application inherently portable.
- The Cloud Run platform is API-compatible with Knative, an open source project, enabling serverless applications to easily run in Kubernetes-based environments.

Considerations when using Cloud Run

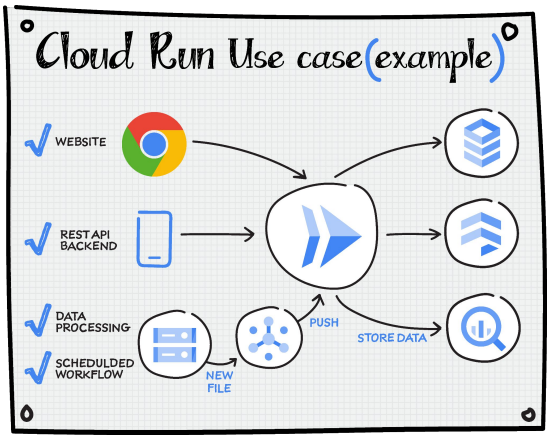
- Autoscaling costs
- Scaling mismatch with downstream systems
- Workload migration

There are some aspects of running your applications on Cloud Run which you should consider:

- If you deploy a service that scales up to many container instances, you will incur costs for running those containers. To limit the number of instances during autoscaling, you can set the maximum number of container instances for your Cloud Run service.
- If your Cloud Run service scales up to many container instances in a short period of time, your downstream systems might not be able to handle the additional traffic load. You'll need to understand the throughput capacity of those downstream systems when configuring your Cloud Run service.
- As part of your application modernization strategy, you'll need to create a migration plan and use tools to migrate VM-based workloads into containers that will run on Cloud Run or Google Kubernetes Engine.

Remember

1. Cloud Run runs and autoscales your application on-demand.
2. Use Cloud Run for applications that serve web requests, including microservices, event processing workflows, and scheduled tasks.
3. Automatic scaling, incremental application updates, and built-in load balancing help you build highly available applications.
4. Cloud Run is designed to make developers more productive.



In summary:

- Cloud Run runs and autoscales your application on-demand.
- Use Cloud Run for applications that serve web requests, including microservices, event processing workflows, and scheduled tasks or jobs.
- Automatic scaling, incremental application updates, and built-in load balancing help you build highly available applications.
- Cloud Run is designed to make developers more productive.