About container instance autoscaling

In Cloud Run, each revision (https://cloud.google.com/run/docs/resource-model#revisions) is automatically scaled to the number of container instances needed to handle all incoming requests.

The number of instances scheduled is impacted by:

- The amount of CPU needed to process a request
- The concurrency setting (https://cloud.google.com/run/docs/about-concurrency)
- The maximum number of container instances setting (https://cloud.google.com/run/docs/configuring/max-instances)

In some cases you may want to limit the total number of container instances that can be started, for cost control reasons, or for better compatibility with other resources used by your service. For example, your Cloud Run service might interact with a database that can only handle a certain number of concurrent open connections.

About maximum container instances

You can use the maximum container instances setting to limit the total number of instances that can be started in parallel, as documented in Setting a maximum number of container instances (https://cloud.google.com/run/docs/configuring/max-instances).

Exceeding maximum instances

Under normal circumstances, your revision scales up by creating new instances to handle incoming traffic load. But when you set a maximum instances limit, in some scenarios there will be insufficient instances to meet that traffic load. In that case, incoming requests queue for up to 60 seconds. During this 60 second window, if an instance finishes processing requests, it becomes available to process queued requests. If no instances become available during the 60 second window, the request fails with a `429` error code on Cloud Run (fully managed).

Scaling guarantees

The maximum instances limit is an upper limit. Setting a high limit does not mean that your revision will scale up to the specified number of container instances. It only means that the number of container instances at any point in time should not exceed the limit.

Traffic spikes

In some cases, such as rapid traffic surges, Cloud Run may, for a short period of time, create slightly *more* container instances than the specified max instances value. If your service cannot tolerate this temporary behavior, you may want to factor in a safety margin and set a lower max instances value.

Deployments

When you deploy a new revision, Cloud Run gradually migrates traffic from the old revision to the new one. Because maximum instances limits are set for each revision, you may temporarily exceed the specified limit during the period after deployment.

Idle instances and minimizing cold starts

You are only billed when an instance is handling a request (https://cloud.google.com/run/pricing#billable_time), but this does not mean that Cloud Run immediately shuts down instances once they have handled all requests. To minimize the impact of cold starts, Cloud Run may keep some instances idle. These instances are ready to handle requests in case of a sudden traffic spike.

For example, when a container instance has finished handling requests, it may remain idle for a period of time in case another request needs to be handled. An idle container instance may persist resources, such as open database connections. However, for Cloud Run (fully managed), the CPU will not be available (https://cloud.google.com/run/docs/reference/container-contract#cpu)

What's next

To manage the maximum number of instances of your Cloud Run services, see Setting a maximum number of container instances (https://cloud.google.com/run/docs/configuring/max-instances).

To manage the maximum number of simultaneous requests handled by each container instance, see Setting concurrency (https://cloud.google.com/run/docs/configuring/concurrency).

To optimize your concurrency setting, see development tips for tuning concurrency (https://cloud.google.com/run/docs/tips#tuning-concurrency).

Last updated December 23, 2019.