Compute sizes and limits

- Compute sizes
- Account quotas
- Request rate limits

Choose a compute size for each sandbox and keep your application's resource usage within your account's quotas. Request rate limits also constrain how quickly your application can send API requests.

Compute sizes

Cloud Sandboxes supports these CPU and memory pairs:

SDK size name	CPUs	Memory	Memory in MiB
`micro`	1	2 GiB	2048
`small`	2	4 GiB	4096
`medium`	4	8 GiB	8192
`large`	8	16 GiB	16384
`xl`	16	32 GiB	32768

The kit launch helpers default to `small` when you omit `resources`. To select another size, pass its name:

```
const sandbox = await client.kits.launchAndWait('shell', {
  resources: 'medium',
});
```

For a registry image, specify a size in `client.create()`, such as `resources: 'small'`. You can also pass an explicit CPU and memory pair: `resources: { cpus: 2, memoryMib: 4096 }`.

Named sizes are an SDK convenience. In direct REST requests, supply both `resources.cpus` and `resources.memoryMib`. CPU and memory aren't independent settings: for example, 1 CPU with 1024

MiB is not a supported pair.

When creating from an existing image resource with `image`, omit resource settings because the image supplies them.

Your account access and available capacity determine whether a request can be accepted. See Billing for pricing and usage information.

Account quotas

The default quotas apply across an account:

Resource	Default limit
Concurrent sandboxes	10
Stored sandboxes	50
Volumes	100
Secrets	100
Images being prepared at the same time	3

Your account can have different quotas. Confirm your account's limits with Docker before planning a workload that depends on a particular allowance.

Stopping a sandbox releases its concurrency slot unless the sandbox is configured as always-on. Restarting a stopped sandbox requires a concurrency slot. A stopped sandbox still counts toward the stored sandbox quota. Delete sandboxes you no longer need to reduce stored usage.

Handle quota errors even if you checked usage before creating a resource. Other applications can consume the remaining allowance between requests. Reduce concurrency or remove unused resources before retrying.

Request rate limits

Rate limits control how quickly you can send requests and can vary by operation. When a request reaches a rate limit, wait before retrying and honor any delay specified by the server. Limit concurrent requests and set bounds on retry attempts and elapsed time.

Check the error details to distinguish a rate limit from a resource quota. Waiting can resolve a rate limit, but exceeding a quota requires reducing resource usage. See Errors and retries for how to retry without duplicating work.

□□□ #2
□ □□□ □□□ 2026-10-31 18:57:32 UTC
□ □□□ □□□ 2026-10-31 18:57:33 UTC