Docker Sandboxes API concepts

- Kits and sandbox images
  - Bundled kits
- Resource names
- Wait for an action to finish
  - Wait for kit setup
- Management and sandbox endpoints
- Read all results from a list
- Choose supported Cloud options
- Supply kit artifacts

An application uses the Docker Sandboxes API to create sandboxes, connect to them, and track their state. Choose the environment for your sandbox, then learn how to work with its resources throughout their lifecycle.

Kits and sandbox images

A sandbox kit defines an environment for an agent or tool, including its image, setup, network rules, and credential requirements. The SDK bundles a catalog of kits you can launch by name. Using kits from other sources requires preparing their content for the API.

You can also create a sandbox from a container image. Choose the source based on how much of the environment you want to configure yourself:

Source	What it provides	How to create a sandbox
Bundled kit	An image reference and configuration included in the SDK's catalog	`client.kits.launch('shell')`
Custom kit	An environment defined by a kit you obtain separately	`client.create()` with prepared kit artifacts

Source	What it provides	How to create a sandbox
Registry image (`imageRef`)	A container image to use with your own sandbox settings	`client.create({ imageRef: 'ubuntu:24.04', resources: 'small' })`
Image resource (`image`)	An image already prepared for Cloud Sandboxes, including its compute settings	`client.create({ image: 'images/<uid>' })`

The `imageRef` value is an image name in a registry. The `image` value is a resource name returned by the Sandboxes API. When you use `image`, omit `resources` because the image resource supplies its compute settings.

These creation methods don't wait for the sandbox to be running. See Wait for an action to finish before running commands.

Bundled kits

The npm package includes the following kit definitions and supporting files. Launch a bundled kit by its short name, such as `shell`:

Kit name	Environment
`shell`	A shell environment for running your own commands
`claude`	Claude Code
`codex`	Codex
`cursor`	Cursor
`devin`	Devin
`docker-agent`	Docker Agent
`gemini`	Gemini CLI
`opencode`	OpenCode

For example, `client.kits.launchAndWait('shell')` creates a shell sandbox and waits until it's running. The kit launch helpers default to `small` compute, with two CPUs and 4 GiB of memory. See Compute sizes to choose a different size.

Bundled kits are tied to the SDK release. Call `client.kits.list()` to see the catalog in your installed version. The kit definitions are included in the npm package, so the SDK doesn't download them from a registry. Cloud Sandboxes pulls their referenced container images as needed.

To run an AI agent, provide credentials for its model provider, such as an Anthropic API key for Claude Code. See Authenticate agents. The `shell` kit needs no provider key to run commands.

Resource names

Use a resource's returned `name` to refer to it in later requests. A sandbox name has the form `sandboxes/<uid>`. Pass the complete name, including the `sandboxes/` prefix, when reading or deleting it.

The server assigns the name, which stays the same throughout the resource's lifetime. The optional `displayName` is a label you can change without changing the resource's identity.

Wait for an action to finish

Wait until a sandbox is running before sending commands to it. In the SDK, `client.kits.launchAndWait()` creates a bundled kit's sandbox and waits for it to run. If you use `client.create()` or `client.kits.launch()`, call `waitUntilRunning()` on the returned sandbox and use the result to run commands.

For direct API requests, HTTP 202 means the action was accepted and is still in progress. Read the resource repeatedly until it reaches the state you need.

Sandbox creation can continue after your client stops waiting. Read the sandbox again to check its state, and inspect its `failure` field if it has failed. See Errors and retries for how to recover.

Deletion can also take time. The API returns HTTP 202 while the sandbox is being deleted and HTTP 204 when deletion is complete. After deletion, authorized reads return `notFound`.

Wait for kit setup

`waitUntilRunning()` and `kits.launchAndWait()` wait for the sandbox to reach the `running` state. Kit setup commands, such as installing tools or cloning a repository, can still be running at that point.

The SDK doesn't provide a helper that waits for all kit setup to finish. If your application depends on that setup, add a readiness check before starting its work. What you check depends on the kit and the task—for example, a completion marker written after a repository clone finishes, or a successful health check from a service.

Poll with a delay between checks and a timeout so your application stops waiting if setup fails.

Management and sandbox endpoints

Creating a sandbox and running a command inside it use different endpoints:

Endpoint	Use it to
Management API at `https://connect.docker.com/sandboxes`	Create, inspect, and delete sandboxes and manage related resources.
Sandbox API at the returned `core.endpoint.uri`	Run processes and read or write files inside that sandbox.

The SDK builds request URLs from these base URLs. If you make HTTP requests directly, append the `/v1` route to the base URL, preserving any existing path. For example, the management route `/v1/sandboxes` becomes `https://connect.docker.com/sandboxes/v1/sandboxes`.

Each sandbox endpoint requires a token that grants access to that sandbox. The SDK obtains this token when you use a sandbox's process or file methods. See Authentication and authorization for details.

A sandbox's endpoint can change when its runtime changes. Read the sandbox resource again before reconnecting to get its endpoint.

Read all results from a list

List requests return one page of results at a time. To retrieve the next page, pass the response's `nextPageToken` as the next request's `pageToken`. Keep the same page size, filter, and ordering. Continue until `nextPageToken` is empty, even if a page contains fewer items than you requested.

The default page size is 25 for Cloud sandbox, image, snapshot, volume, and secret lists. You can request up to 100 items per page.

Choose supported Cloud options

Cloud supports kits, sandbox timeouts, stored secrets, and volume attachments, subject to account permissions and feature availability. For example, volume access must be enabled for your account. An SDK method's presence doesn't guarantee that your account can use it.

Supply kit artifacts

To use a kit outside the bundled catalog, your application must load and prepare its content before calling `client.create()`. The npm SDK doesn't fetch kits from a registry. Its `kits.launch()` and `kits.launchAndWait()` helpers accept only bundled kit names.

For example, the Hermes agent kit is published as `docker.io/sbx/hermes-agent-kit:latest`. To use it through the SDK, you need code outside the SDK that loads the kit definition and its supporting files into the serialized v2 artifact format accepted by the API.

The kit artifacts described here use the v2 format. The `kits` array contains a sandbox kit and any mixins that add configuration to it. See the v2 kit reference for details. The bundled launch helpers prepare this same input for the kits in their catalog.

Pass the kit's source reference and prepared artifact bytes to `client.create()`:

```
function createFromKit(reference: string, artifactBytes: Uint8Array) {
  return client.create({
    resources: 'small',
    kits: [
      {
        artifact: {
          ref: { ref: reference, kind: 'sandbox' },
          inline: artifactBytes,
        },
      },
    ],
  });
}
```

The `ref` identifies the kit's source. It doesn't trigger a registry pull. The `inline` value contains the serialized artifact as a `Uint8Array`, including the kit's file content. Raw `spec.yaml`, ZIP files, and OCI manifests aren't valid inputs for this field.

To launch a public kit by registry reference without writing loading code, use the Docker Agentic Platform Console.

□□□ #2
□ □□□ □□□ 2026-10-31 18:54:46 UTC
□ □□□ □□□ 2026-10-31 18:54:47 UTC