Clone a cloud sandbox

- Read the source configuration {#1-read-the-source-configuration}
- Create a fresh sandbox {#2-create-a-fresh-sandbox}

Create another sandbox using selected settings from an existing one. This example copies the image source and environment, not the sandbox's files or running processes. For a saved filesystem or memory state, use a snapshot.

You need an authenticated client, the source sandbox's resource name, and a display name for the new sandbox.

Read the source configuration {#1-read-the-source-configuration}

Get the source sandbox and inspect its image and environment. Review environment values before copying them; they may contain credentials or settings specific to the original job.

Do not treat the complete response as a create request. It includes read-only state and connection details.

{{< tabs >}} {{< tab name="TypeScript" >}}

```
const sandbox = await client.get(name);
return sandbox.core;
```

Complete TypeScript example: clone/inspect.ts

```
import type { Sandboxes } from '@docker/sandboxes';

export async function inspectSandbox(client: Sandboxes, name: string) {
  const sandbox = await client.get(name);
  return sandbox.core;
}
```

{{< /tab >}} {{< /tabs >}}

Create a fresh sandbox {#2-create-a-fresh-sandbox}

The example copies the image source and environment. For a registry image it also copies resource settings. It waits for the new sandbox to run.

The call returns the new sandbox's handle, ready for running commands or transferring files.

This is a limited configuration copy, not a complete clone. It does not copy kits, attached policies, secrets, volumes, files, or process state. A copied image alone may not reproduce a kit's setup. For repeated agent environments, launch the same named kit with the options your application saved.

Attach required network policies and credentials before running work in the new sandbox. The source is unchanged, and both sandboxes need their own cleanup.

{{< tabs >}} {{< tab name="TypeScript" >}}

```
const copy = await client.create(
  {
    displayName: name,
    ...(core.image
      ? { image: core.image, agent: core.agent }
      : {
          imageRef: core.imageRef!,
          resources: core.resources,
        }),
    environment: core.environment,
  },
```

```
  { timeoutMs: 300_000, idempotencyKey: requestId },
);
return copy.waitUntilRunning();
```

Complete TypeScript example: clone/rebuild.ts

```
import type { Sandbox, Sandboxes } from '@docker/sandboxes';

export async function rebuildSandbox(
  client: Sandboxes,
  core: Sandbox['core'],
  name: string,
  requestId: string,
) {
  if (!core.image && !core.imageRef)
    throw new TypeError('Source has no image');
  const copy = await client.create(
    {
      displayName: name,
      ...(core.image
        ? { image: core.image, agent: core.agent }
        : {
            imageRef: core.imageRef!,
            resources: core.resources,
          }),
      environment: core.environment,
    },
    { timeoutMs: 300_000, idempotencyKey: requestId },
  );
  return copy.waitUntilRunning();
}
```

{{< /tab >}} {{< /tabs >}}

□□□ #2
□ □□□ □□□ 2026-10-31 18:55:22 UTC
□ □□□ □□□ 2026-10-31 18:55:23 UTC