



FREE

STANDALONE

NO DOCKER DESKTOP REQUIRED

Docker Sandboxes

MicroVM-powered isolation for AI coding agents. Local or cloud, one CLI.

The problem

AI agents need real development environments: Docker builds, shell access, system packages, multi-container services. Every existing sandbox forces a tradeoff: VMs are slow, containers leak privileges, WASM can't run a real OS. So most developers skip sandboxing, run agents on the host, and babysit permission prompts.

The solution

Docker Sandboxes gives every agent session a dedicated microVM with its own kernel, a private Docker daemon, workspace sync, and no path back to the host. It runs any coding agent on any machine where it is installed. It is a free, standalone product from Docker and does not require Docker Desktop.



Dedicated microVM per session

Each sandbox gets its own kernel. A compromised or runaway agent hits a hardware boundary, not your machine.



Full Docker inside the sandbox

Native docker build, run, and compose from a private engine. No socket mounting, no host privileges.



Infrastructure-enforced boundaries

Files, network, and secrets are defined before the agent runs. The boundary comes from infrastructure, not a system prompt.

How it compares

	Docker Sandboxes	VM (full virtual machine)	Container-based approaches (e.g. Modal)	WASM-based (e.g. Cloudflare)	Bare (no sandboxing)
Depth of isolation/safety	Full kernel isolation	Generally considered enough isolation	Namespace isolation, not strong enough (kernel) boundary	Not true OS isolation based on v8 isolates; very limited things you can do in it	Completely at risk; no guardrails
Speed of spin up/tear down	Fast spin up and tear down	Very long spin up and tear down times; needs management overhead	Faster than VMs, slower than WASM, speed due to no kernel boot (trade off for true isolation)	Generally considered fast	
Ability to let agents use Docker containers	Supports Docker containers within sandbox	Can spin up containers within VM	Can't do this without elevated privileges; runs into Docker-in-Docker problem	No, JS/WASM only; no OS at all	

"You don't trust agents with security. You build walls around them. Docker has been ahead of the curve on exactly this, and Docker Sandboxes is what that looks like at the infrastructure level."

Gavriel Cohen, Creator of NanoClaw

Autonomy without the babysitting

Agents do their best work with freedom. Because the microVM is the safety boundary, permissive modes are the default: hand the agent a task and walk away. No permission prompts, no reviewing every command, and your machine stays untouched.

Run unattended

The sandbox is the safety, not the prompt. Agents work at full speed without approval loops.

Disposable by default

Off the rails? Delete it and start fresh in seconds. Nothing to clean up, nothing to roll back.

Parallel work

Run several agents on several tasks at once, each in its own isolated environment.

Key capabilities

✓ Full development environment

Clone repos, install dependencies, run tests, build images, open PRs. Nothing stubbed or simulated.

✓ Scoped access

You decide which files, endpoints, and secrets each agent gets. Credentials are injected at runtime, never baked in.

✓ Workspace sync

Your project files sync into the sandbox and results sync back. The agent works on real code.

✓ Cross-platform

A purpose-built VMM runs natively on macOS (Hypervisor.framework), Windows (WHP), and Linux (KVM). One codebase, zero translation layers.

✓ Fast cold starts

No performance reason to run outside the sandbox.

✓ Local to cloud and back

Move a sandbox between your laptop and the cloud with one command.

Agent compatibility

Works with Claude Code, Codex, Cursor, Devin, OpenCode, GitHub Copilot, Docker Agent, and next-generation autonomous systems. Same isolation, same speed, one sandbox model.

Why a new VMM

Existing microVM runtimes like Firecracker were designed for cloud, Linux-only environments. Coding agents run on developer laptops. Docker built a new VMM from scratch on each OS's native hypervisor: no translation layers, no platform-specific caveats, no friction that makes developers skip sandboxing.

Customizing sandboxes

EARLY ACCESS

Two ways to shape a sandbox beyond the built-in defaults, so teams standardize how agents run without rebuilding environments each time.

Templates

Reusable sandbox images with tools, packages, and config baked in. Build once, or save a running sandbox, so every session starts fast.

Kits

EXPERIMENTAL

One declarative YAML file gives an agent its tools, credentials, network rules, and config at runtime, or defines a new agent from scratch. The agent boots ready, the same way, every time.

Templates and kits compose: ship one heavy template, then layer thin kits on top. Config, secrets, and capabilities change without rebuilding the image.



Cloud sandboxes

NEW

When a job outgrows your machine, run the identical microVM model on Docker-hosted infrastructure. Same CLI, same trust model, same Kits: add the `--cloud` flag to `sbx` commands. The isolation boundary does not change in either direction.

```
$ sbx --cloud run claude # sandbox  
ready: microVM, same as local
```

```
$ sbx move my-project --  
to cloud # filesystem captured,  
cloud sandbox created
```

```
$ sbx move my-project  
--to local # snapshot imported,  
back on your laptop
```

Burst up

Heavy builds and long test suites run on cloud hardware, not your battery.

Same isolation

The same microVM boundary on both sides. No downgrade either way.

Carry it over

`sbx move` recreates a sandbox's filesystem on the other side, in either direction. The work comes with it.

Keep working

Detach and close the laptop. The agent keeps going.

"Docker Sandboxes let agents have the autonomy to do long-running tasks without compromising safety. We're excited to integrate Sandboxes into Warp so developers can **run agents freely with a consistent environment**, whether they're running locally or in the cloud."

Ben Navetta, Engineering Lead, Warp

Deployment

Individual developers: Install and run standalone. Docker Desktop is not required. Sign up for a free Docker account to start.

Teams & enterprise: Centralized filesystem and network policies enforced across the organization. Contact Docker to learn about enterprise deployment.

Then `sbx run claude` or `sbx run codex`.

macOS

```
$ brew trust docker/tap && brew  
install docker/tap/sbx
```

Windows

```
> winget install  
Docker.sbx
```

Linux

```
$ curl -fsSL https://get.docker.com |  
sudo REPO_ONLY=1 sh  
$ sudo apt-get install docker-sbx
```

No tradeoff. VM-grade isolation. Fast starts. Full Docker support.

Your agents run autonomously, with guardrails. On your laptop or in the cloud.

