Cloudflare Sandbox SDK — Training Doc

What Is It?

Cloudflare Sandbox lets you run untrusted code securely in isolated containers on Cloudflare's edge network. Think of it as giving your users a full Linux computer in the cloud, spun up in seconds, running at the edge — and you control it all through a simple API from your Worker.

The Architecture (Three Products in One)

```
Your Worker (application logic) ↓ Durable Objects (persistent sandbox identity) ↓ Containers (isolated Linux environment where code runs)
```

This is the first Cloudflare product that combines Workers + Durable Objects + Containers into a single developer experience.

What We Built

An interactive code playground on saltwaterbrc.com/playground.html where visitors can:

- Write and run **shell commands** in an isolated Linux container
- Execute **Python** scripts
- Run **Node.js / JavaScript** code
- See the full architecture: Worker → Durable Object → Container

The Worker (sandbox-worker)

Three endpoints:

- `/exec` — Execute a shell command in the sandbox container
- `/run-code` — Write a file (Python, JS, or Bash) and execute it
- `/info` — Returns the architecture explanation (great for demos)

Safety filters block destructive commands (fork bombs, rm -rf /, etc.).

The Dockerfile

```
FROM docker.io/cloudflare/sandbox:0.7.4 # Install Python and Node.js for the playground RUN apt-get update && apt-get install -y --no-install-recommends \ python3 \ nodejs \ && rm -rf /var/lib/apt/lists/* EXPOSE 8080
```

The base `cloudflare/sandbox` image is bare-bones Linux. We added Python and Node to support multi-language execution.

The Wrangler Config

```
{ "name": "saltwaterbrc-sandbox", "containers": [{ "class_name": "Sandbox", "image": "./Dockerfile", "instance_type": "lite", "max_instances": 3 }], "durable_objects": { "bindings": [{ "class_name": "Sandbox", "name": "Sandbox" }] } }
```

Key pieces: `containers` defines the Docker image and instance type. `durable_objects` binds the Sandbox class. The SDK handles the plumbing between DO and Container automatically.

How It Works (Step by Step)

1. **Visitor types code** in the browser on playground.html 2. **Frontend sends POST** to `saltwaterbrc-sandbox.saltwaterbrc.workers.dev/exec` 3. **Worker receives request**, validates input, checks safety filters 4. **Worker calls** `getSandbox()` — routes to a Durable Object 5. **Durable Object manages the container** — spins up or reuses an isolated Linux environment 6. **Code executes inside the container** — completely isolated filesystem, processes, network 7. **stdout/stderr returned** to the Worker, then back to the browser

First request is slower (container cold start). Subsequent requests are fast because the container stays warm.

Development Workflow

Prerequisites

- **Docker Desktop** — required to build container images locally
- **Wrangler CLI** — deploys Worker + container to Cloudflare
- **Node.js** — for npm/wrangler

Local Development

```
cd sandbox-worker npm install npm run dev # Starts local dev server + builds Docker container
```

First run takes 2-3 minutes (Docker image build). After that, fast reloads.

Deployment

```
npm run deploy # Builds Docker image, pushes to Cloudflare registry, deploys Worker
```

Docker must be running. First deploy pushes all image layers (~200MB+). Subsequent deploys are incremental.

Gotchas We Hit

- **Workers Paid plan required** — Containers/Sandbox needs the paid tier, not just the free plan
- **Docker Desktop networking** — Large image pushes can timeout. Fix: set `"max-concurrent-uploads": 1` in Docker Engine settings
- **Platform mismatch** — If you install `node_modules` on Linux (or in a VM) and try to run on macOS, workerd binary won't match. Fix: delete `node_modules` and reinstall on the target platform
- **Base image is bare-bones** — The `cloudflare/sandbox:0.7.4` image has almost nothing installed. Need to add languages in the Dockerfile
- **Python image variant** — Cloudflare offers `cloudflare/sandbox:0.7.4-python` but we used the base image + apt-get for more control

Customer Use Cases (By Vertical)

Gaming Industry

- **Sandbox for game logic testing**: Developers test game server code in isolated environments at the edge
- **Mod environments**: Let players or creators run custom scripts safely
- **Real-time multiplayer testing**: Spin up sandboxed game instances globally

Financial Services

- **Financial modeling**: Run Python-based financial models in sandboxed containers
- **AI-powered risk analysis**: Run risk models in sandboxed environments — no data leaves the edge
- **Client-specific compute**: Spin up isolated environments per fund/client

Healthcare

- **Secure medical data processing**: Run analysis in isolated containers with strong security boundaries
- **AI diagnostic tools**: AI models running in sandboxed environments with compliance-grade isolation
- **Research computing**: Sandboxed environments for research teams

Automotive

- **Connected vehicle simulation**: Test vehicle API logic in sandboxes
- **OTA update testing**: Sandbox firmware/software updates before deployment
- **Developer environments**: Remote dev environments for global engineering teams

Any Company Building AI Agents

- **Code execution for AI**: Let AI agents (Claude, GPT, etc.) write and run code safely
- **Interactive tutorials**: Users learn to code in sandboxed environments
- **CI/CD at the edge**: Run build and test pipelines in isolated containers

The Full Stack on saltwaterbrc.com

```
saltwaterbrc.com ■■■ Cloudflare Pages → Hosts the site (Phase 2 ■) ■■■ Cloudflare Workers → API logic at the edge (Phase 3 ■) ■■■ Durable Objects → Persistent state / visitor counter (Phase 3 ■) ■■■ R2 Storage → Zero-egress object storage (Phase 3 ■) ■■■ Workers AI + Vectorize → "Ask This Blog" RAG feature (Phase 4 ■) ■■■ AI Gateway → AI observability & control (Phase 4 ■) ■■■ Sandbox SDK → Interactive code playground (Phase 4 ■)
```

The Sales Pitch (One Paragraph)

"Cloudflare Sandbox lets your developers run code in isolated containers at the edge — no infrastructure to manage, strong security boundaries, and it spins up in seconds. It's built on Workers, Durable Objects, and Containers, so it integrates natively with everything else on the platform. Use cases range from AI agent code execution to interactive dev environments to secure data processing. I built a live playground on my own site — you can try it right now."

Resources

- Documentation: https://developers.cloudflare.com/sandbox/
- GitHub: https://github.com/cloudflare/sandbox-sdk
- Getting started: https://developers.cloudflare.com/sandbox/get-started/
- API reference: https://developers.cloudflare.com/sandbox/api/
- Live demo: https://saltwaterbrc.com/playground.html