Secure AI Dev Toolbox — User Guide

Reusable, network-restricted Docker environment with the aidev launcher

Version: 1.3 • Generated: August 30, 2025

Published image

Use the prebuilt image: docker.io/pdrittenhouse/secure-ai-dev:1

1. Quick Start (use the published image)

```
mkdir -p "$HOME/.secure-ai-dev"
echo 'SECURE_AI_IMAGE="docker.io/pdrittenhouse/secure-ai-dev:1"' >> "$HOME/.secure-ai-dev/config
# install aidev (from your repo) and ensure allowlist
```

VS Code devcontainers example:

```
{
  "image": "docker.io/pdrittenhouse/secure-ai-dev:1",
  "workspaceFolder": "/workspaces/app",
  "workspaceMount": "source=${localWorkspaceFolder},target=/workspaces/app,type=bind",
  "runArgs": ["--cap-add=NET_ADMIN","--cap-add=NET_RAW","--add-host=host.docker.internal:host-ga
  "mounts": ["source=${env:HOME}/.secure-ai-dev/security/allowlist,target=/opt/security/allowlis
  "remoteUser": "vscode",
  "postStartCommand": "sudo /opt/security/setup-firewall.sh"
}
```

2. Build from Source (this repo)

Use these when you’re developing the base image itself or want to test local changes before publishing.

Local single-arch build

```
docker build -t pdrittenhouse/secure-ai-dev:dev .
echo 'SECURE_AI_IMAGE="pdrittenhouse/secure-ai-dev:dev"' >> "$HOME/.secure-ai-dev/config"
aidev rm 2>/dev/null || true
aidev && aidev doctor
```

Multi-arch build & push (Docker Hub)

```
docker buildx create --use 2>/dev/null || true
docker login
docker buildx build --platform linux/amd64,linux/arm64   -t docker.io/pdrittenhouse/secure-ai-de
docker buildx imagetools inspect docker.io/pdrittenhouse/secure-ai-dev:1
```

Load a specific arch locally (no push)

```
# Apple Silicon
docker buildx build --platform linux/arm64 -t pdrittenhouse/secure-ai-dev:dev --load .
# x86_64
docker buildx build --platform linux/amd64 -t pdrittenhouse/secure-ai-dev:dev --load .
```

3. Allowlists & Security Notes

Your host controls networking via allowlists. Global allowlist is mounted read-only at /opt/security/allowlist. Per-project file is ./.allowed-domains.txt.

```
# Example global list
registry.npmjs.org
github.com
api.github.com
objects.githubusercontent.com
codeload.github.com
nodejs.org
api.openai.com
api.anthropic.com
host.docker.internal
# Project list
aidev domains add n8n.example.com
aidev domains test api.openai.com n8n.example.com
aidev reload
```

4. `aidev` Reference & Tips

Command	What it does
aidev [start]	Start or reattach a secure container for the current folder.
aidev sh	Open a bash shell inside the container.
aidev stop	Stop the container for this folder.
aidev rm	Remove (force) the container for this folder.
aidev reload	Re-run /opt/security/setup-firewall.sh.
aidev doctor [--verbose]	Diagnostics; with --verbose, per-domain IP checks.
aidev domains add/remove/list/test	Manage per■project allowlist from the host.

5. Git Hygiene & Release Flow

Commit the Dockerfile, security scripts, and the aidev launcher. Do not commit operator-specific allowlists.

```
.gitignore:
/.allowed-domains.txt
.DS_Store
```

Release steps:

```
1) Update Dockerfile / security scripts
2) Build & push a new tag (e.g., :2)
3) Update docs to reference the new tag
4) Users: aidev rm && aidev  (or Dev Containers → Rebuild)
```