Kiro

- Quick start
- Authentication
  - Device flow login
  - Manual login
  - Authentication persistence
- Configuration
  - Default startup command
- Base image

{{% include "sandboxes-local-scope.md" %}}

This guide covers authentication, configuration, and usage of Kiro in a sandboxed environment.

Official documentation: Kiro CLI

Quick start

Create a sandbox and run Kiro for a project directory:

```
$ sbx run kiro ~/my-project
```

The workspace parameter is optional and defaults to the current directory:

```
$ cd ~/my-project
$ sbx run kiro
```

On first run, Kiro prompts you to authenticate using device flow.

Authentication

Kiro uses device flow authentication, which requires interactive login through a web browser. This method provides secure authentication without storing API keys directly.

Device flow login

When you first run Kiro, it prompts you to authenticate:

1. Kiro displays a URL and a verification code
2. Open the URL in your web browser
3. Enter the verification code
4. Complete the authentication flow in your browser
5. Return to the terminal - Kiro proceeds automatically

The authentication session is persisted in the sandbox and doesn't require repeated login unless you destroy and recreate the sandbox.

Manual login

You can trigger the login flow manually:

```
$ sbx run kiro --name <sandbox-name> -- login --use-device-flow
```

This command initiates device flow authentication without starting a coding session.

Authentication persistence

Kiro stores authentication state in `~/.local/share/kiro-cli/data.sqlite3` inside the sandbox. This database persists as long as the sandbox exists. If you destroy the sandbox, you'll need to authenticate again when you recreate it.

Configuration

Sandboxes don't pick up user-level configuration from your host. Only project-level configuration in the working directory is available inside the sandbox. See Why doesn't the sandbox use my user-level agent configuration? for workarounds.

Kiro requires minimal configuration. The agent runs with trust-all-tools mode by default, which lets it execute commands without repeated approval prompts.

Default startup command

Without extra args, the sandbox runs:

```
kiro chat --trust-all-tools
```

When the first argument after `--` is a flag (begins with `-`), it's added after the defaults — for example, `sbx run kiro -- --resume` runs `kiro chat --trust-all-tools --resume`. When the first argument is a bare word, it replaces the defaults, which is why `sbx run kiro -- login --use-device-flow` runs the login subcommand on its own. To run `chat` with extra arguments of your own, include the subcommand:

```
$ sbx run kiro -- chat --trust-all-tools --resume
```

Base image

Template: `docker/sandbox-templates:kiro`

Authentication state is persisted across sandbox restarts.

See Customize to pre-install tools or customize this environment.

□□□ #2
□ □□□ □□□ 2026-10-31 18:57:57 UTC
□ □□□ □□□ 2026-10-31 18:57:58 UTC