C3 CLI

- Create a new application
- Web frameworks
- Deploy
- Creating a new Pages project that is connected to a git repository
- CLI Arguments
- Telemetry

import { Render, TabItem, Tabs, Type, MetaInfo, PackageManagers, } from "~/components";

Cloudflare provides a CLI command for creating new Workers and Pages projects — `npm create cloudflare`, powered by the `create-cloudflare` package.

Create a new application

Open a terminal window and run:

Running this command will prompt you to install the `create-cloudflare` package, and then ask you questions about the type of application you wish to create.

:::note To create a Pages project you must now specify the `--platform=pages` arg, otherwise C3 will always create a Workers project. :::

Web frameworks

If you choose the "Framework Starter" option, you will be prompted to choose a framework to use. The following frameworks are currently supported:

- Angular
- Astro
- Docusaurus

- Gatsby
- Hono
- Next.js static exports
- Nuxt
- Qwik
- React
- Redwood
- Remix
- SolidStart
- SvelteKit
- Vue

When you use a framework, `npm create cloudflare` directly uses the framework's own command for generating a new projects, which may prompt additional questions. This ensures that the project you create is up-to-date with the latest version of the framework, and you have all the same options when creating you project via `npm create cloudflare` that you would if you created your project using the framework's tooling directly.

Deploy

Once your project has been configured, you will be asked if you would like to deploy the project to Cloudflare. This is optional.

If you choose to deploy, you will be asked to sign into your Cloudflare account (if you aren't already), and your project will be deployed.

Creating a new Pages project that is connected to a git repository

To create a new project using `npm create cloudflare`, and then connect it to a Git repository on your Github or Gitlab account, take the following steps:

1. Run `npm create cloudflare@latest`, and choose your desired options
2. Select `no` to the prompt, "Do you want to deploy your application?". This is important — if you select `yes` and deploy your application from your terminal (Direct Upload), then it will

not be possible to connect this Pages project to a git repository later on. You will have to create a new Cloudflare Pages project.

3. Create a new git repository, using the application that `npm create cloudflare@latest` just created for you.
4. Follow the steps outlined in the Git integration guide

CLI Arguments

C3 collects any required input through a series of interactive prompts. You may also specify your choices via command line arguments, which will skip these prompts. To use C3 in a non-interactive context such as CI, you must specify all required arguments via the command line.

This is the full format of a C3 invocation alongside the possible CLI arguments:

- `DIRECTORY`
  - The directory where the application should be created. The name of the application is taken from the directory name.
- `NESTED ARGS..`
  - CLI arguments to pass to eventual third party CLIs C3 might invoke (in the case of full-stack applications).
- `--category`
  - The kind of templates that should be created.
  - The possible values for this option are:
    - `hello-world`: Hello World example
    - `web-framework`: Framework Starter
    - `demo`: Application Starter
    - `remote-template`: Template from a GitHub repo
- `--type`
  - The type of application that should be created.
  - The possible values for this option are:
    - `hello-world`: A basic "Hello World" Cloudflare Worker.
    - `hello-world-durable-object`: A Durable Object and a Worker to communicate with it.
    - `common`: A Cloudflare Worker which implements a common example of routing/proxying functionalities.
    - `scheduled`: A scheduled Cloudflare Worker (triggered via Cron Triggers).
    - `queues`: A Cloudflare Worker which is both a consumer and produced of Queues .
    - `openapi`: A Worker implementing an OpenAPI REST endpoint.
    - `pre-existing`: Fetch a Worker initialized from the Cloudflare dashboard.

- `--framework`
  - The type of framework to use to create a web application (when using this option, `--type` is ignored).
  - The possible values for this option are:
    - `angular`
    - `astro`
    - `docusaurus`
    - `gatsby`
    - `hono`
    - `next`
    - `nuxt`
    - `qwik`
    - `react`
    - `redwood`
    - `remix`
    - `solid`
    - `svelte`
    - `vue`
- `--template`
  - Create a new project via an external template hosted in a git repository
  - The value for this option may be specified as any of the following:
    - `user/repo`
    - `git@github.com:user/repo`
    - `https://github.com/user/repo`
    - `user/repo/some-template` (subdirectories)
    - `user/repo#canary` (branches)
    - `user/repo#1234abcd` (commit hash)
    - `bitbucket:user/repo` (BitBucket)
    - `gitlab:user/repo` (GitLab)

    See the `degit` docs for more details.

    At a minimum, templates must contain the following:
    - `package.json`
    - Wrangler configuration file
    - `src/` containing a worker script referenced from the Wrangler configuration file

    See the templates folder of this repo for more examples.
- `--deploy`
  - Deploy your application after it has been created.
- `--lang`
  - The programming language of the template.
  - The possible values for this option are:
    - `ts`
    - `js`
    - `python`
- `--ts`
  - Use TypeScript in your application. Deprecated. Use `--lang=ts` instead.

- `--git`
  - Initialize a local git repository for your application.
- `--open`
  - Open with your browser the deployed application (this option is ignored if the application is not deployed).
- `--existing-script`
  - The name of an existing Cloudflare Workers script to clone locally. When using this option, `--type` is coerced to `pre-existing`.
  - When `--existing-script` is specified, `deploy` will be ignored.
- `-y`, `--accept-defaults`
  - Use all the default C3 options each can also be overridden by specifying it.
- `--auto-update`
  - Automatically uses the latest version of C3.
- `-v`, `--version`
  - Show version number.
- `-h`, `--help`
  - Show a help message.

:::note

All the boolean options above can be specified with or without a value, for example `--open` and `--open true` have the same effect, prefixing `no-` to the option's name negates it, so for example `--no-open` and `--open false` have the same effect.

:::

Telemetry

Cloudflare collects anonymous usage data to improve `create-cloudflare` over time. Read more about this in our data policy.

You can opt-out if you do not wish to share any information.

Alternatively, you can set an environment variable:

```
export CREATE_CLOUDFLARE_TELEMETRY_DISABLED=1
```

You can check the status of telemetry collection at any time.

You can always re-enable telemetry collection.

□□□ #2
□ □□□ □□□ 2026-11-01 03:03:05 UTC
□ □□□ □□□ 2026-11-01 03:03:06 UTC