The Heroku CLI

Last updated 20 March 2020

Table of Contents

The Heroku Command Line Interface (CLI) makes it easy to create and manage your Heroku apps directly from the terminal. It's an essential part of using Heroku.

Download and install

The Heroku CLI requires **Git**, the popular version control system. If you don't already have Git installed, complete the following before installing the CLI:

- Git installation (https://git-scm.com/book/en/v2/Getting-Started-Installing-Git)
- First-time Git setup (https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup)

macOS

Download the installer (https://cli-assets.heroku.com/heroku.pkg)

Also available via Homebrew:

```
$ brew tap heroku/brew && brew install heroku
```

Windows

Download the appropriate installer for your Windows installation

64-bit installer (https://cli-assets.heroku.com/heroku-x64

32-bit installer (https://cli-assets.heroku.com/heroku-x86

Ubuntu 16+

Run the following from your terminal:

```
$ sudo snap install --classic heroku
```

Snap is available on other Linux OS's as well (https://snapcraft.io).

Currently, the macOS installer on Catalina displays an "unknown author" warning. We are working to address this.

Currently, the Windows installers may display a warning titled "Windows protected your PC". To run the installation when this warning is shown, click "More info", verify the publisher as "Heroku, Inc.", then click the "Run anyway" button.

Other installation methods

Standalone installation

The standalone install is a simple tarball with a binary that is useful in scripted environments or where there is restricted access (non-sudo). It contains its own node.js binary and will autoupdate like the above install methods. **We encourage this method inside docker containers.**

To quickly setup into `/usr/local/lib/heroku` and `/usr/local/bin/heroku`, run this script (script requires sudo and not Windows compatible):

```
$ curl https://cli-assets.heroku.com/install.sh | sh
```

Otherwise, download one of the tarballs below and extract it yourself.

Tarballs

These are available in `gz` or `xz` compression. `xz` is much smaller but `gz` is more compatible.

- macOS (https://cli-assets.heroku.com/heroku-darwin-x64.tar.gz)
- Linux (x64) (https://cli-assets.heroku.com/heroku-linux-x64.tar.gz)
- Linux (arm) (https://cli-assets.heroku.com/heroku-linux-arm.tar.gz)
- Windows (x64) (https://cli-assets.heroku.com/heroku-win32-x64.tar.gz)
- Windows (x86) (https://cli-assets.heroku.com/heroku-win32-x86.tar.gz)

Ubuntu / Debian apt-get

```
$ curl https://cli-assets.heroku.com/install-ubuntu.sh | sh
```

This version does not autoupdate and must be updated manually via `apt-get`. Use the snap or standalone installation for an autoupdating version of the CLI.

Arch Linux

This package (https://aur.archlinux.org/packages/heroku-cli) is community maintained and **not** by Heroku.

```
$ yay -S heroku-cli
```

npm

The CLI is built with Node.js and is installable via `npm`. This is a manual install method that can be used in environments where autoupdating is not ideal or where Heroku does not offer a prebuilt Node.js binary.

> It's strongly recommended to use one of the other installation methods if possible.
>
> This installation method does not autoupdate and requires you to use your system's version of Node.js, which may be older than the version Heroku develops the CLI against. Heroku uses very current releases of Node.js and does not back-support older versions.
>
> If you use any of the other installation methods the proper version of Node.js is already included, and it doesn't conflict with any other version on your system.
>
> Also, this method won't use the yarn lockfile for dependencies like the others do (even if you install with yarn). This may cause issues if the CLI's dependencies become incompatible in minor or patch releases.

This installation method is required for users on ARM and BSD. You must have `node` and `npm` installed already.

```
$ npm install -g heroku
```

Verifying your installation

To verify your CLI installation, use the `heroku --version` command:

```
$ heroku --version
heroku/7.0.0 (darwin-x64) node-v8.0.0
```

You should see `heroku/x.y.z` in the output. If you don't, but you have installed the Heroku CLI, it's possible you have an old `heroku` gem on your system. Uninstall it with these instructions.

Getting started

After you install the CLI, run the `heroku login` command. You'll be prompted to enter any key to go to your web browser to complete login. The CLI will then log you in automatically.

```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit
 ›   Warning: If browser does not open, visit
 ›   https://cli-auth.heroku.com/auth/browser/***
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

If you'd prefer to stay in the CLI to enter your credentials, you may run `heroku login -i`

```
$ heroku login -i
heroku: Enter your login credentials
Email: me@example.com
Password: ***************
Two-factor code: ********
Logged in as me@heroku.com
```

The CLI saves your email address and an API token to `~/.netrc` for future use. For more information, see Heroku CLI Authentication (https://devcenter.heroku.com/articles/authentication).

Now you're ready to create your first Heroku app:

```
$ cd ~/myapp
$ heroku create
Creating app... done, ⬢ sleepy-meadow-81798
https://sleepy-meadow-81798.herokuapp.com/ | https://git.heroku.com/sleepy-meadow-81798.git
```

Check out your preferred language's getting started guide (https://devcenter.heroku.com/start) for a comprehensive introduction to deploying your first app.

Staying up to date

The Heroku CLI keeps itself and its plugins (except linked plugins) up to date automatically, *unless* you installed the Debian/Ubuntu package or used `npm install`.

When you run a `heroku` command, a background process checks for the latest available version of the CLI. If a new version is found, it's downloaded and stored in `~/.local/share/heroku/client`. This background check happens at most once every 4 hours.

The `heroku` binary checks for an up-to-date client in `~/.local/share/heroku/client` before using the originally installed client.

Latest release SHAs

Darwin

- x64 (https://cli-assets.heroku.com/darwin-x64)

Linux

- arm (https://cli-assets.heroku.com/linux-arm)
- x64 (https://cli-assets.heroku.com/linux-x64)

Windows

- x64 (https://cli-assets.heroku.com/win32-x64)
- x86 (https://cli-assets.heroku.com/win32-x86)

Useful CLI plugins

CLI plugins allow you to extend your CLI installation. Install a CLI plugin with `heroku plugins:install someplugin`. See Using CLI Plugins (https://devcenter.heroku.com/articles/using-cli-plugins) for more information on plugin management.

Here are some useful plugins you might want to try:

- heroku-builds (https://github.com/heroku/heroku-builds) — create builds from tarballs
- heroku-repo (https://github.com/heroku/heroku-repo) — purge build cache, reset repo, and other commands to manipulate the Heroku git repository
- api (https://github.com/heroku/heroku-api-plugin) — make ad-hoc API requests (such as `heroku api GET /account`)
- heroku-pg-extras (https://github.com/heroku/heroku-pg-extras) — Add extra `heroku pg:*` commands
- heroku-slugs (https://github.com/heroku/heroku-slugs) — Downloads app slugs
- heroku-kafka (https://github.com/heroku/heroku-kafka-jsplugin) — Manage Heroku Kafka
- heroku-papertrail (https://github.com/papertrail/papertrail-heroku-plugin) — Display, tail, and search for logs with Papertrail

CLI architecture

The Heroku CLI is built with the Open CLI Framework (oclif (https://oclif.io)), developed within Heroku / Salesforce. oclif is available as a framework for any developer to build a large or a small CLI. The framework includes a CLI generator, automated documentation creation, and testing infrastructure.

The code for the Heroku CLI is also open source (https://github.com/heroku/cli). It does *not* require Node.js or any other dependencies to run. Unless you install the Debian/Ubuntu package or use `npm install`, the CLI contains its own Node.js binary that does not conflict with other applications.

Troubleshooting

If you're having issues with the CLI, first ensure that you're using the latest version (http://cli-assets.heroku.com/version). If you're not, try updating with `heroku update`.

Not all methods of installation support `heroku update`.

- If you installed the CLI with `apt`, you need to use `sudo apt-get update && sudo apt-get upgrade heroku` instead.
- If you installed the CLI with `npm` or `yarn`, you need to use `npm upgrade -g heroku` or `yarn global upgrade heroku` instead.

If the CLI fails to update, try uninstalling it with the instructions below, then reinstalling it. Ensure that you don't have the legacy Heroku Toolbelt or Heroku Ruby gem installed by using `which heroku` or `where heroku` (on Windows) to confirm what the `heroku` command points to. You might need to modify your `PATH` to include `/usr/local/bin/heroku` (for most installations).

If you're still encountering an issue, you can set the following debugging environment variables to help diagnose it:

Environment Variable	Description
`HEROKU_DEBUG=1`	Shows debugging information mostly related to Heroku API interactions
`HEROKU_DEBUG_HEADERS=1`	Alongside `HEROKU_DEBUG=1`, shows HTTP headers
`DEBUG=*`	Shows very verbose debugging information

You can also check the CLI's error logfile, which is stored at one of the following locations depending on your operating system:

OS	Location
macOS	`~/Library/Caches/heroku/error.log`
Windows	`%LOCALAPPDATA%\heroku\error.log`
Linux/Other	`~/.cache/heroku/error.log` (or `XDG_CACHE_HOME` if set)

If you continue to have problems and the CLI is up to date, or if updating fails for other reasons, you can reset the CLI by deleting its user directories. These directories are replaced automatically and you will not be logged out, but you *will* lose any installed plugins.

First, run `heroku plugins` to list your installed plugins so you can make sure to reinstall them.

Then, delete the following directories:

Windows:

- `%LOCALAPPDATA%\heroku`

macOS/Linux/Other:

- `~/.local/share/heroku` (or `XDG_DATA_HOME` if set)
- Either `~/Library/Caches/heroku` on macOS, or `~/.cache/heroku` on Linux/Other (or `XDG_CACHE_HOME` if set)

Login issues

If you are experiencing issues with logging in, try moving your `.netrc` file. This is where the CLI stores credentials:

```
$ mv ~/.netrc ~/.netrc.backup
$ heroku login
```

On Windows, the file is named `_netrc`.

Homebrew-specific issues

If you get legacy warnings even though you installed the latest homebrew version of heroku, this is happening because the binary `heroku` command in your `PATH` environment variable is not pointing to the version that brew installed.

First, run `which heroku` to see what binary `heroku` is pointing to. If it is not `/usr/local/bin/heroku`, you need to either delete the binary it is pointing to, or make `/usr/local/bin/` higher up in your `PATH` environment variable by modifying your `~/.bashrc` file or equivalent.

Next, run `brew link --overwrite heroku` to make sure that `/usr/local/bin/heroku` is pointing to the new CLI. If you continue to have trouble, run `brew doctor` which should point out any issues with your system.

Uninstalling the Heroku CLI

Note that this also deletes all CLI plugins.

macOS

On macOS, you can uninstall the CLI by typing:

```
$ rm -rf /usr/local/heroku /usr/local/lib/heroku /usr/local/bin/heroku ~/.local/share/heroku ~/Library/Caches/heroku
```

Homebrew

If you installed the Heroku CLI using Homebrew, you can uninstall the CLI by typing:

```
$ brew uninstall heroku
$ rm -rf ~/.local/share/heroku ~/Library/Caches/heroku
```

Linux

Standalone installs

For standalone installs, you can uninstall the CLI by typing:

```
$ rm /usr/local/bin/heroku
$ rm -rf /usr/local/lib/heroku /usr/local/heroku
$ rm -rf ~/.local/share/heroku ~/.cache/heroku
```

Debian and Ubuntu installs

For Debian/Ubuntu, you can uninstall the CLI by typing:

```
$ sudo apt-get remove heroku heroku-toolbelt
$ sudo rm /etc/apt/sources.list.d/heroku.list
```

If you have `$XDG_DATA_HOME`, and/or `$XDG_CACHE_HOME` it will use those variables instead of `~/.local/share`, and `~/.cache`.

You can remove the release key by running these commands:

```
$ sudo apt-key list
$ sudo apt-key del KEYFROMABOVE
```

Windows

On Windows, to uninstall the Heroku CLI:

1. Click **`Start > Control Panel > Programs > Programs and Features`**.
2. Select **`Heroku CLI`**, and then click **`Uninstall`**. (Note that the uninstaller is unsigned)

The Windows uninstaller is *not* automatically updated alongside the CLI. If it's been a while since you first installed the CLI and you're attempting to uninstall it to fix an issue, you might first need to manually install the latest version of the CLI to obtain an up-to-date uninstaller.

If this is unsuccessful, manually delete `%LOCALAPPDATA%\heroku` along with the directory in Program Files.

Uninstalling the legacy heroku gem

To find out where the executable is located, run `which`.

```
$ which heroku
/usr/local/heroku/bin/heroku
```

The path to the `heroku` command should not be a Ruby gem directory.

If it is, uninstall it and any other `heroku` gems:

```
$ gem uninstall heroku --all
```