Netlify CLI

Deploy from the terminal

START HERE

`netlify` **`login`**
Sign in via browser

`netlify` **`deploy`**
Deploy a draft

`netlify` **`link`**
Tie folder to a project

`netlify` **`deploy --prod`**
Deploy to production

`netlify` **`dev`**
Run locally

`netlify` **`env:list`**
Show variables

`netlify` **`build`**
Build like Netlify does

`netlify` **`logs`**
Read function logs

Local folder .netlify/ state.json → `netlify deploy` → **Draft deploy** unique URL → `--prod` → **Production** live site → `netlify watch` → **Deploy finished** status in terminal

CONCEPTS

A plain netlify deploy makes a draft with its own URL; only --prod changes the live site.

netlify deploy builds first by default; pass --no-build when the output already exists.

In CI, authenticate with the NETLIFY_AUTH_TOKEN variable and pick the project with --site or NETLIFY_SITE_ID.

SETUP

`npm i -g netlify-cli`
Install the CLI

`netlify` **`login`**
Sign in via browser

`netlify` **`link`**
Link folder to an existing project

`netlify` **`init`**
Set up continuous deployment

`netlify` **`link --name`** *`<name>`*
Link by project name

`netlify` **`link --id`** *`<id>`*
Link by project ID

`netlify` **`sites:create --name`** *`<name>`*
Create an empty project

`netlify` **`clone`** *`<owner/repo>`*
Clone a repo and link it

`netlify` **`unlink`**
Unlink this folder

`netlify` **`switch`**
Switch active account

`netlify` **`status`**
Show login and linked project

`netlify` **`open`**
Open project settings in the browser

LOCAL DEV

`netlify` **`dev`**
Run the site with functions locally

`netlify` **`dev --port`** *`<port>`*
Use another port

`netlify` **`dev --command`** **"***`<cmd>`***"**
Set the framework dev command

`netlify` **`dev --live`**
Share a public tunnel URL

`netlify` **`dev --no-open`**
Do not open the browser

`netlify` **`dev --offline`**
Skip features needing the network

LOCAL DEV (cont.)

`netlify` **`dev --context`** *`<context>`*
Use another env var context

`netlify` **`dev --filter`** *`<app>`*
Pick an app in a monorepo

`netlify` **`serve`**
Production build, served locally

`netlify` **`build`**
Run the build locally

`netlify` **`build --dry`**
Show build steps without running

`netlify` **`build --context deploy-preview`**
Build with preview env vars

DEPLOY

`netlify` **`deploy`**
Deploy a draft

`netlify` **`deploy --prod`**
Deploy to production

`netlify` **`deploy --dir`** *`<path>`*
Folder to publish

`netlify` **`deploy --functions`** *`<folder>`*
Functions folder to ship

`netlify` **`deploy --no-build`**
Skip the build step

`netlify` **`deploy --message`** **"***`<msg>`***"**
Note in the deploy log

`netlify` **`deploy --alias`** *`<name>`*
Predictable draft subdomain

`netlify` **`deploy --json`**
Print deploy data as JSON

`netlify` **`deploy --prod-if-unlocked`**
Prod if unlocked, else draft

`netlify` **`deploy --site`** *`<name-or-id>`*
Target a specific project

`netlify` **`deploy --env`** *`<KEY>=<value>`*
Deploy-only var for functions

`netlify` **`deploy --secret-env`** *`<KEY>=<value>`*
Same, masked in UI and API

`netlify` **`deploy --allow-anonymous`**
Deploy without an account

`netlify` **`claim`**
Claim an anonymous deploy

`netlify` **`watch`**
Wait for a deploy to finish

ENV VARS

`netlify` **`env:list`**
List variables

ENV VARS (cont.)

`netlify` **`env:get`** *`<name>`*
Print one value

`netlify` **`env:set`** *`<name>`* *`<value>`*
Set a variable

`netlify` **`env:unset`** *`<name>`*
Remove a variable

`netlify` **`env:import`** *`<file>`*
Load variables from a .env file

`netlify` **`env:import`** *`<file>`* **`--replace-existing`**
Replace instead of merge

`netlify` **`env:set`** *`<name>`* *`<value>`* **`--context production`**
Limit to a deploy context

`netlify` **`env:set`** *`<name>`* *`<value>`* **`--scope builds`**
Limit to builds or functions

`netlify` **`env:set`** *`<name>`* *`<value>`* **`--secret`**
Make the value unreadable later

`netlify` **`env:list --context`** *`<context>`*
List one context (default dev)

`netlify` **`env:list --plain`**
Print as plain text

`netlify` **`env:clone --to`** *`<project-id>`*
Copy variables to another project

Deploy contexts

Context	Used for
`production`	Production deploys
`deploy-preview`	Pull request previews
`branch-deploy`	Branch deploys
`dev`	netlify dev and CLI default
`branch:<name>`	One specific branch

FUNCTIONS

`netlify` **`functions:create`** *`<name>`*
Scaffold a new function

`netlify` **`functions:list`**
List local functions

`netlify` **`functions:invoke`** *`<name>`*
Call a function in netlify dev

`netlify` **`functions:invoke`** *`<name>`* **`--payload`** **'***`<json>`***'**
Send a JSON body

`netlify` **`functions:invoke`** *`<name>`* **`--querystring`** **"***`<q>`***"**
Add a query string

`netlify` **`functions:serve`**
Serve functions only

`netlify` **`functions:build`**
Build functions locally

FUNCTIONS (cont.)

`netlify` **`functions:create --language`** *`<lang>`* **`--template`** *`<name>`*
Skip the template picker

LOGS & PROJECTS

`netlify` **`logs`**
Recent function logs (10 min)

`netlify` **`logs --follow`**
Stream logs live

`netlify` **`logs --since`** *`<time>`*
Widen the time window

`netlify` **`logs --function`** *`<name>`*
Only one function

`netlify` **`logs --source deploy --deploy-id`** *`<id>`*
Build logs of a deploy

`netlify` **`logs --level error`**
Only given levels

`netlify` **`logs --json`**
JSON Lines output

`netlify` **`sites:list`**
List your projects

`netlify` **`sites:search`** *`<name>`*
Find projects by name

`netlify` **`sites:delete`** *`<id>`*
Delete a project for good

GLOBAL OPTIONS

`--site` *`<name-or-id>`*
Run against another project

`--auth` *`<token>`*
Authenticate with a token

`--filter` *`<app>`*
Pick an app in a monorepo

`netlify` **`help`** *`<command>`*
Help for one command

RECIPES

First deploy of a project

```
npm i -g netlify-cli
netlify login
netlify link
netlify env:import .env
netlify deploy --prod --dir
dist
```

CI deploy with a token

```
export
NETLIFY_AUTH_TOKEN=<token>
export NETLIFY_SITE_ID=<site-
id>
netlify deploy --build --prod
```

Preview, check, then publish

```
netlify deploy --dir dist --
message "<msg>"
# open the draft URL and check
it
netlify deploy --dir dist --
prod --no-build
```