Vercel CLI Deploy from the terminal

Local folder .vercel link → `vercel` → **Preview** unique URL → `vercel promote` → **Production** live domains → `vercel rollback` → **Earlier deployment** previous build

CONCEPTS

A plain vercel is a preview deploy; only --prod (or the first deploy of a project) goes to production.

In CI, authenticate with the VERCEL_TOKEN environment variable rather than --token so it stays out of process lists.

stdout of a deploy is always the deployment URL, so vercel > url.txt works in scripts.

SETUP

npm i -g vercel
Install the CLI

`vercel` **login**
Sign in to your account

`vercel` **link**
Link folder to a Vercel project

`vercel` **whoami**
Show the signed-in user

`vercel` **switch** *<team>*
Change active team scope

`vercel` **--version**
Show CLI version

`vercel` **upgrade**
Update the CLI

DEPLOY

vercel
Deploy a preview

`vercel` **--prod**
Deploy to production

`vercel` **deploy --prebuilt**
Deploy output of vercel build

`vercel` **build**
Build locally into .vercel/output

`vercel` **--force**
Deploy without the build cache

`vercel` **--prod --skip-domain**
Deploy prod, keep domains on old one

`vercel` **--logs**
Print build logs while deploying

`vercel` **--yes**
Skip project setup questions

`vercel` **--no-wait**
Exit before the build finishes

`vercel` **--env** *<KEY>=<value>*
Set a runtime env var

`vercel` **redeploy** *<url>*
Rebuild an existing deployment

LOCAL & ENV

`vercel` **dev**
Run the project locally

`vercel` **pull**
Fetch env vars and settings to .vercel/

`vercel` **env ls**
List environment variables

`vercel` **env add** *<name> <environment>*
Add an env var

`vercel` **env update** *<name> <environment>*
Change an env var's value

`vercel` **env rm** *<name> <environment>*
Remove an env var

`vercel` **env pull** *<file>*
Write dev vars to a file

`vercel` **env pull --environment=preview**
Pull Preview variables instead

`vercel` **env run --** *<command>*
Run a command with the env vars

INSPECT & LOGS

`vercel` **list**
Recent deployments

INSPECT & LOGS (cont.)

`vercel` **inspect** *<url>*
Show deployment details

`vercel` **inspect** *<url>* **--logs**
Show its build logs

`vercel` **logs**
Runtime logs, last 30 minutes

`vercel` **logs --since** *<duration>*
Widen the time window

`vercel` **logs --deployment** *<url>*
Logs of one deployment

`vercel` **logs --query** *'<kql>'*
Filter with a KQL query

`vercel` **logs requests**
CDN request records

RELEASE

`vercel` **promote** *<url>*
Make a deployment the current one

`vercel` **rollback**
Return production to prior deployment

`vercel` **rollback** *<url>*
Roll back to a specific deployment

DOMAINS

`vercel` **domains ls**
List domains

`vercel` **domains add** *<domain> <project>*
Add a domain to a project

`vercel` **alias set** *<url> <domain>*
Point a domain at a deployment

OPTIONS

--cwd *<dir>*
Run against another directory

--scope *<team>*
Run as another team

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

--non-interactive
Never prompt (CI, scripts)

RECIPES

First deploy of a project

```
npm i -g vercel
vercel login
vercel link
vercel env pull
vercel --prod
```

Stage, check, then promote

```
vercel --prod --skip-domain
vercel inspect <url> --logs
vercel promote <url>
```