Heroku CLI

Heroku platform CLI

START HERE

`heroku login`
Log in via the browser

`heroku config:set` *`<KEY>`*`=`*`<value>`*
Set an env var

`heroku create` *`<app>`*
New app plus git remote

`heroku ps:scale web=1`
Run one web dyno

`git push heroku main`
Build and deploy

`heroku run bash`
Shell in a one-off dyno

`heroku logs --tail`
Stream logs

`heroku open`
Open the app in a browser

git push source to Heroku → **buildpacks** → **Build** slug → **+ config vars** → **Release** v1, v2, ... → **Procfile** → **Dynos** web, worker, one-off

CONCEPTS

Most commands need an app: pass -a <app>, set HEROKU_APP, or run inside a git repo with a heroku remote (-r <remote> picks another remote).

Changing config vars, add-ons or the slug creates a new release and restarts all dynos.

LOGIN AND APPS

`heroku login`
Log in via the browser

`heroku login -i`
Log in at the prompt (no browser)

`heroku auth:whoami`
Show the logged-in user

`heroku apps`
List your apps

`heroku create` *`<app>`*
Create an app and a heroku git remote

`heroku create` *`<app>`* **`--region eu`**
Create in the EU region

`heroku apps:info -a` *`<app>`*
Stack, region, URL, dyno counts

`heroku open -a` *`<app>`*
Open the app URL in a browser

`heroku apps:destroy -a` *`<app>`* **`--confirm`** *`<app>`*
Delete an app for good

`heroku access:add` *`<email>`* **`-a`** *`<app>`*
Add a collaborator

`heroku maintenance:on -a` *`<app>`*
Show the maintenance page

`heroku maintenance:off -a` *`<app>`*
Back to serving traffic

GIT DEPLOY

`git push heroku main`
Deploy the main branch

`git push heroku` *`<branch>`***`:main`**
Deploy another local branch

`heroku git:remote -a` *`<app>`*
Add the heroku remote to this repo

`heroku git:remote -a` *`<app>`* **`-r`** *`<remote>`*
Add a remote under another name

CONFIG VARS

`heroku config -a` *`<app>`*
List all config vars

`heroku config:set` *`<KEY>`*`=`*`<value>`* **`-a`** *`<app>`*
Set one or more vars

`heroku config:get` *`<KEY>`* **`-a`** *`<app>`*
Print one value

`heroku config:unset` *`<KEY>`* **`-a`** *`<app>`*
Remove a var

`heroku config -s -a` *`<app>`*
Output as KEY=value lines

DYNOS

`heroku ps -a` *`<app>`*
List dynos and their state

`heroku ps:scale web=`*`<n>`* **`-a`** *`<app>`*
Set the number of web dynos

`heroku ps:scale web=0 -a` *`<app>`*
Stop all web dynos

`heroku ps:scale worker=1:Standard-2X -a` *`<app>`*
Count and size in one go

`heroku ps:type -a` *`<app>`*
Show dyno sizes

`heroku ps:type` *`<size>`* **`-a`** *`<app>`*
Resize all process types

`heroku ps:restart -a` *`<app>`*
Restart all dynos

`heroku ps:restart -p web -a` *`<app>`*
Restart one process type

`heroku ps:exec -a` *`<app>`*
SSH into a running dyno

LOGS

`heroku logs --tail -a` *`<app>`*
Stream logs live

`heroku logs -n` *`<lines>`* **`-a`** *`<app>`*
Print the last n lines

`heroku logs -p web -a` *`<app>`*
Only one process type

ONE-OFF DYNOS

`heroku run bash -a` *`<app>`*
Interactive shell in a new dyno
A fresh dyno with the current release, not one of the running ones; file changes are lost on exit.

ONE-OFF DYNOS (cont.)

`heroku run` *`<cmd>`* **`-a`** *`<app>`*
Run a command, e.g. migrations

`heroku run:detached` *`<cmd>`* **`-a`** *`<app>`*
Run in the background

RELEASES

`heroku releases -a` *`<app>`*
Release history

`heroku rollback -a` *`<app>`*
Back to the previous release
Rolls back code and config vars, not database schema or add-ons.

`heroku rollback v`*`<n>`* **`-a`** *`<app>`*
Back to a given release

ADD-ONS

`heroku addons -a` *`<app>`*
List add-ons of an app

`heroku addons:create` *`<service>`***`:`***`<plan>`* **`-a`** *`<app>`*
Provision an add-on

`heroku addons:attach` *`<addon>`* **`-a`** *`<other-app>`*
Share an add-on with another app

`heroku addons:destroy` *`<addon>`* **`-a`** *`<app>`*
Remove an add-on and its data

POSTGRES

`heroku pg:info -a` *`<app>`*
Plan, size, connections

`heroku pg:psql -a` *`<app>`*
Open psql on the database

`heroku pg:backups -a` *`<app>`*
List backups

`heroku pg:backups:capture -a` *`<app>`*
Take a manual backup

`heroku pg:backups:download -a` *`<app>`*
Download the latest as latest.dump

DOMAINS AND CERTS

`heroku domains -a` *`<app>`*
List domains and DNS targets

`heroku domains:add` *`<hostname>`* **`-a`** *`<app>`*
Add a custom domain
Point a CNAME at the DNS target it prints.

DOMAINS AND CERTS (cont.)

`heroku certs:auto:enable -a` *`<app>`*
Free Let's Encrypt certs (ACM)

`heroku certs:auto -a` *`<app>`*
ACM status per domain

PIPELINES

`heroku pipelines:create` *`<name>`* **`-a`** *`<app>`*
Create a pipeline with this app

`heroku pipelines:add` *`<pipeline>`* **`-a`** *`<app>`* **`-s`** *`<stage>`*
Add an app as staging or production

`heroku pipelines:promote -a` *`<app>`*
Promote the slug downstream

BUILDPACKS AND LOCAL

`heroku buildpacks -a` *`<app>`*
Show the app's buildpacks

`heroku buildpacks:set` *`<buildpack>`* **`-a`** *`<app>`*
Replace the buildpack

`heroku local`
Run the Procfile locally

RECIPES

First deploy

```
heroku login
heroku create <app>
heroku config:set
SECRET_KEY=<value>
git push heroku main
heroku ps:scale web=1
heroku open
```

Staging and production

```
heroku create <staging-app> -r
staging
heroku create <prod-app> -r prod
heroku pipelines:create <name> -
a <staging-app> -s staging
heroku pipelines:add <name> -a
<prod-app> -s production
git push staging main
heroku pipelines:promote -r
staging
```

Database backup to local

```
heroku pg:backups:capture -a
<app>
heroku pg:backups:download -a
<app>
pg_restore --no-owner -d
<localdb> latest.dump
```

Bad deploy

```
heroku releases -a <app>
heroku logs -n 200 -a <app>
heroku rollback -a <app>
heroku run -x <migrate cmd> -a
<app>
```