flyctl Fly.io CLI

CONCEPTS

fly and flyctl are the same binary. Most commands read the app name from fly.toml in the current directory; use -a <app> elsewhere.

launch app + fly.toml
↓ deploy
Release image, rolling update
↓ Machines
Machines run in regions
↓ logs / ssh / scale

fly deploy builds on a remote builder by default; --local-only uses your Docker daemon.

ACCOUNT AND APPS

`fly auth login`
Log in via the browser

`fly auth whoami`
Show the logged-in user

`fly apps list`
List your apps

`fly apps create <name>`
Create an app without deploying

`fly apps destroy <name>`
Delete an app

LAUNCH AND DEPLOY

`fly launch`
Detect project, create app and fly.toml

`fly launch --no-deploy`
Configure only, deploy later

`fly launch --name <app> --region <code>`
Choose name and region

`fly deploy`
Build and deploy from the current directory

`fly deploy -a <app>`
Deploy a specific app

`fly deploy --image <image>`
Deploy a prebuilt image

`fly deploy --strategy <name>`
rolling, canary, bluegreen or immediate

STATUS AND LOGS

`fly status`
App, Machines and their state

`fly logs`
Stream logs

`fly logs --no-tail`
Print buffered logs and exit

`fly logs --machine <id>`
Logs of one Machine

SECRETS

`fly secrets set <NAME>=<value>`
Set encrypted secrets

`fly secrets list`
Names and digests, never values

MACHINES AND SCALING

`fly scale count <n>`
Set the number of Machines

`fly scale show`
Current VM size and count

`fly scale vm <size>`
Change size, e.g. shared-cpu-1x

`fly machine list`
List Machines

ACCESS AND DEBUGGING

`fly ssh console`
Interactive shell on a Machine

`fly ssh console -C <cmd>`
Run one command

`fly proxy <local>:<remote>`
Tunnel a local port to the app

VOLUMES

`fly volumes create <name> -r <code> -s <gb>`
Create a volume

`fly volumes list`
List volumes

DOMAINS AND IPS

`fly certs add <domain>`
Request a TLS certificate

`fly certs check <domain>`
Check DNS and certificate state