flyctl Fly.io CLI

START HERE

`fly` **`auth login`**
Log in in the browser

`fly` **`logs`**
Stream app logs

`fly` **`launch`**
Create app and fly.toml

`fly` **`secrets set`** *`<NAME>=<value>`*
Store a secret

`fly` **`deploy`**
Build and release the app

`fly` **`ssh console`**
Shell into a Machine

`fly` **`status`**
Machines and health

`fly` **`scale count`** *`<n>`*
Run n Machines

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

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.

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

Setting or unsetting secrets redeploys the app unless you pass --stage; apply staged ones with fly secrets deploy.

ACCOUNT AND APPS

`fly` **`auth login`**
Log in via the browser

`fly` **`auth whoami`**
Show the logged-in user

`fly` **`auth token`**
Print an API token

`fly` **`apps list`**
List your apps

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

`fly` **`apps destroy`** *`<name>`*
Delete an app

`fly` **`apps restart`** *`<name>`*
Restart all Machines of an app

`fly` **`platform regions`**
List region codes

`fly` **`doctor`**
Diagnose flyctl and connectivity

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 -c`** *`<file>`*
Use another config file

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

`fly` **`deploy --local-only`**
Build with the local Docker daemon

LAUNCH AND DEPLOY (cont.)

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

`fly` **`deploy --build-arg`** *`<NAME>=<value>`*
Pass a build argument

`fly` **`deploy --regions`** *`<a,b>`*
Limit to machines in these regions

`fly` **`deploy --detach`**
Return without watching the rollout

`fly` **`deploy --ha=false`**
Skip the extra standby Machine

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

`fly` **`logs --region`** *`<code>`*
Logs of one region

`fly` **`logs --json`**
Logs as JSON

`fly` **`releases`**
Release history

`fly` **`open`**
Open the app in a browser

`fly` **`dashboard`**
Open the web dashboard

`fly` **`config show`**
Print the config Fly has deployed

SECRETS

`fly` **`secrets set`** *`<NAME>=<value>`*
Set encrypted secrets
Triggers a redeploy unless --stage is given.

`fly` **`secrets list`**
Names and digests, never values

`fly` **`secrets unset`** *`<NAME>`*
Remove a secret

`fly` **`secrets set --stage`** *`<NAME>=<value>`*
Stage without redeploying

`fly` **`secrets deploy`**
Apply staged secrets

SECRETS (cont.)

`fly` **`secrets import <`** *`<file>`*
Set many from NAME=VALUE lines on stdin

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` **`scale memory`** *`<mb>`*
Set memory

`fly` **`machine list`**
List Machines

`fly` **`machine status`** *`<id>`*
Detail of one Machine

`fly` **`machine stop`** *`<id>`*
Stop a Machine

`fly` **`machine start`** *`<id>`*
Start a Machine

`fly` **`machine restart`** *`<id>`*
Restart a Machine

`fly` **`machine destroy`** *`<id>`*
Delete a Machine

`fly` **`machine clone`** *`<id>`*
Copy a Machine, e.g. into another region

`fly` **`machine update`** *`<id>`*
Change a Machine's config

`fly` **`machine exec`** *`<id> <cmd>`*
Run a command on a Machine

`fly` **`machine run`** *`<image>`*
Start a one-off Machine

ACCESS AND DEBUGGING

`fly` **`ssh console`**
Interactive shell on a Machine

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

`fly` **`ssh console --machine`** *`<id>`*
Pick a specific Machine

`fly` **`ssh console -s`**
Choose from a list

`fly` **`ssh sftp shell`**
SFTP session to a Machine

`fly` **`proxy`** *`<local>:<remote>`*
Tunnel a local port to the app
Reaches Machines over WireGuard, e.g. a private database.

`fly` **`proxy 5433:5432`** *`<host>`*
Forward to a specific host

VOLUMES

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

`fly` **`volumes list`**
List volumes

`fly` **`volumes show`** *`<id>`*
Volume details

`fly` **`volumes extend`** *`<id>`* **`-s`** *`<gb>`*
Grow a volume

`fly` **`volumes snapshots list`** *`<id>`*
List snapshots

`fly` **`volumes fork`** *`<id>`*
Copy a volume

`fly` **`volumes destroy`** *`<id>`*
Delete a volume

DOMAINS AND IPS

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

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

`fly` **`certs list`**
List certificates

`fly` **`certs remove`** *`<domain>`*
Remove a certificate

`fly` **`ips list`**
Show allocated IPs

`fly` **`ips allocate-v4`**
Allocate a dedicated IPv4

`fly` **`ips allocate-v6`**
Allocate an IPv6

RECIPES

First deploy

```
fly auth login
fly launch --no-deploy
fly secrets set
DATABASE_URL=<url>
fly deploy
fly status
```

Persistent data

```
fly volumes create data -r
<code> -s 3
# fly.toml: [mounts] source =
"data", destination = "/data"
fly deploy
```

Custom domain

```
fly certs add example.com
fly certs check example.com
# add the DNS records it prints
```

Debug a running app

```
fly logs --no-tail
fly ssh console -C "ls /app"
fly proxy 5433:5432
```