

Create a token

Railway	Project settings → Tokens. Account tokens: railway.com/account/tokens
Render	Dashboard → Account Settings → API key. Shown in full only once
Fly.io	<code>fly tokens create deploy -a <app> --name ci --expiry 48h</code> (copy all of it, FlyV1 prefix included)
Vercel	vercel.com/account/tokens . <code>vercel tokens add "CI deploy"</code> works only if you already use a classic account token; a <code>vercel login</code> session can't make one
Heroku	<code>heroku authorizations:create -d "ci" -e 86400 -S</code> (-e in seconds; no expiry if left out)
DigitalOcean	cloud.digitalocean.com/account/api/tokens
Cloudish	POST <code>/api/v1/keys</code> (cloudish.ai), no credential needed (see the Cloudish block)

Give the token to your script

Railway	<code>RAILWAY_TOKEN</code>	Project token: one environment in one project
Railway	<code>RAILWAY_API_TOKEN</code>	Account token (broadest) or workspace token (one workspace)
Render	<code>RENDER_API_KEY</code>	API key. Unlike <code>render login</code> tokens, it doesn't expire on its own
Fly.io	<code>FLY_API_TOKEN</code>	A deploy token covers one app
Vercel	<code>VERCEL_TOKEN</code>	A personal token from your account
Heroku	<code>HEROKU_API_KEY</code>	Overrides the token saved by <code>heroku login</code>
DigitalOcean	<code>DIGITALOCEAN_ACCESS_TOKEN</code>	An API token; or pass <code>--access-token</code> per command
Cloudish	<code>CLOUDISH_API_KEY</code>	One key; it can't spend past its prepaid balance

No browser on the machine?

Railway	<code>railway login --brow serless</code> (for SSH sessions)
Render	Skip <code>login</code> ; the token variable above is enough
Fly.io	Skip <code>login</code> ; the token variable above is enough
Vercel	Skip <code>login</code> ; the token variable above is enough
Heroku	Skip <code>login</code> ; the token variable above is enough
DigitalOcean	With <code>DIGITALOCEAN_ACCESS_TOKEN</code> set, <code>doctl auth init</code> saves it with no prompt
Cloudish	Nothing to log in to: the first call returns the key



Deploy	
Railway	<code>RAILWAY_TOKEN=xxx railway up</code> (add <code>--ci</code> to stream only build logs and exit when the build completes)
Render	<code>render deploys create <service-id> --output json --confirm --wait</code>
Fly.io	<code>fly deploy</code> (uses the settings in <code>fly.toml</code> ; builds remotely by default)
Vercel	<code>vercel deploy --prod --yes > url.txt</code> (stdout is the deploy URL)
Heroku	<code>heroku container:login</code> , then <code>heroku container:push web -a <app></code> , then <code>heroku container:release web -a <app></code>
DigitalOcean	First time: <code>doctl apps create --spec app.yaml --wait</code> . Later: <code>doctl apps create-deployment <app-id> --wait</code>
Cloudish	<code>POST /api/v1/projects</code> (cloudish.ai) with a tar.gz of the folder (see the Cloudish block)
Did it work?	
Railway	<code>railway up</code> exits 0 when the deploy reaches SUCCESS, 1 if it fails. <code>--ci</code> exits once the build completes. With <code>--detach</code> : <code>poll railway deployment list --json</code>
Render	<code>--wait</code> blocks; a failed deploy exits non-zero
Fly.io	Watches the deploy until the machines are healthy (<code>--wait-timeout</code> , default 5 min); <code>--detach</code> returns at once. Then: <code>fly status -a <app></code>
Vercel	Exit code not 0 = error; read stderr
Heroku	No wait flag. Check <code>heroku releases -a <app></code> and <code>heroku ps -a <app></code>
DigitalOcean	<code>--wait</code> gives up after 30 min by default (<code>--wait -ti meout 1h</code> for longer); the deploy itself keeps running
Cloudish	Poll <code>GET /api/v1/images/builds/<id></code> until status is succeeded or failed
Read logs	
Railway	<code>railway logs</code> · build: <code>railway logs --build</code> · last 100: <code>railway logs -n 100</code>
Render	<code>render logs --resources <service-id> --tail</code> (<code>--resources</code> is required in scripts)
Fly.io	<code>fly logs -a <app></code> streams; <code>--no-tail</code> prints the buffer and exits
Vercel	<code>vercel logs --deployment <url-or-id> --json</code> (default: last 30 min, 100 records)
Heroku	<code>heroku logs --tail -a <app></code> · <code>heroku logs -n 200 -a <app></code> (Cedar apps, the generation before Fir: up to 1,500 lines)
DigitalOcean	<code>doctl apps logs <app-id> web --type build</code> (build, deploy or run; add <code>--follow</code> to stream)
Cloudish	<code>GET /api/v1/docker/<alias>/<app>/logs</code>



Cloudish: the whole flow in curl

```
# 1. Get a key. No account, card or browser.
curl -X POST $API/keys \
  -H "content-type: application/json" \
  -d '{"alias": "my-app"}'
# -> "key": "cld_...", "apiKey": {"alias": ...}
export CLOUDISH_API_KEY=cld_...
# Keep the key out of chat and logs.
ALIAS= my-app # the apiKey.alias in the answer
AUTH="Authorization: Bearer $CLOUDISH_API_KEY"

# 2. Pack the folder (Dockerfile at its root),
# minus secrets.
tar --exclude='.env*' --exclude='.git' \
  -czf context.tar.gz .

# 3. Build on Cloudish's servers and deploy.
# No local Docker.
curl -X POST $API/projects -H "$AUTH" \
  -F name=my-app -F port=8080 \
  -F context=context.tar.gz
# -> "build": {"id": 123, "status": "pending"}

# 4. Wait for the build, then URL and logs.
curl $API/images/builds/123 -H "$AUTH"
curl $API/projects/$ALIAS/my-app -H "$AUTH"
# The URL is in subdomain.url.
curl $API/docker/$ALIAS/my-app/logs -H "$AUTH"
```

Cloudish errors you'll meet

503 "nothing accepts connections on port ..."	port doesn't match the port your server listens on (default 8080)
"Job has reached the specified backoff limit"	The build most likely ran out of CPU or memory; retry with <code>buildCpuCores/buildMemoryGb</code>
402	The key's balance is zero. Builds and deploys answer 402; running apps stop within a minute, volumes kept

Gotchas by host

Railway	A script never creates a project on its own. Link one first (<code>railway up --new</code> creates one), or the run fails with <code>NOT_AUTHENTICATED</code> or a no-project error
Railway	Set only one of <code>RAILWAY_TOKEN</code> and <code>RAILWAY_API_TOKEN</code> ; both at once is an error
Railway	Raw API calls with a project token use the <code>Project-Access-Token</code> header, not <code>Authorization: Bearer</code>
Fly.io	Deploy tokens last 20 years unless you set <code>--expiry</code>
Vercel	If <code>--token</code> and <code>VERCEL_TOKEN</code> are both set, the flag wins



Gotchas by host (cont)

Heroku	<code>container:push</code> needs Docker on your machine. Heroku documents this flow for Cedar-generation apps only
Heroku	<code>heroku auth:token</code> gives a token valid for 1 year; use <code>authorizations:create</code> for CI
DigitalOcean	<code>doctl apps create</code> needs App Platform to have access to your repo
Cloudish	The <code>claimUrl</code> (which lets a person take over the key) works once and expires in 30 minutes

Keep tokens safe

- Put the token in an environment variable, not a flag. Vercel's docs warn that command-line arguments "can be visible in process lists and logs".
- Pick the narrowest token: a Railway project token over an account token, a Fly deploy token over an org token.
- Give it an end date: Fly `--expiry 48h`, Heroku `-e 86400`. Render API keys don't expire on their own, so revoke them in the dashboard when done.
- Revoke: fly `tokens list`, then `fly tokens revoke <id>`. Vercel: `vercel tokens rm <id>`.
- Never paste a key into chat, code or logs, and keep `.env` files out of anything you upload.

Which host fits

You need a fully automated workflow	Cloudish: the only one of the seven where the script creates its own key. On the others, a person signs up and makes the token first. (Without an account, a person can still try two others: <code>vercel deploy --temporary</code> puts a folder online, not in CI, until someone claims it; <code>ssh railway.new</code> gives a Linux VM to claim within 24 hours.)
You need a managed Postgres database	Railway (<code>railway add --database postgres</code>) or Render (Render Postgres). Cloudish has none; you run the database inside your container on a volume
A person iterates on a frontend app	Vercel

Sources (as of 2026-10-26)

- Railway: CLI · up · login · API · free trial
 - Render: CLI · CLI reference · API
 - Fly.io: tokens · GitHub Actions · fly deploy · fly logs
 - Vercel: CLI · deploy · logs · global options
 - Heroku: authentication · CLI commands · container registry · logging · generations
 - DigitalOcean: auth init · apps create · create-deployment · apps logs
 - Cloudish: `skill.md` · `llms.txt`
- Made by Cloudish (cloudish.ai/), by Hal9 (hal9.com/). Cloudish is one of the seven hosts on this sheet. Spotted something out of date? Leave a comment.

