

Deploying your app

CompactHive's deploy flow is push-to-deploy: once it's set up, you don't run any deploy commands yourself.

The flow

1. **Connect a repo.** In the portal, connect a GitHub or GitLab repository, or use the built-in Gitea instance if you'd rather not bring your own. GitHub can be connected with one click via the CompactHive GitHub App, which auto-detects your project and generates a CI pipeline for you.
2. **Push your code.** Your repo needs a `Dockerfile` describing how to build your app. Push to your default branch (or open a PR, depending on your pipeline config).
3. **CI builds it.** Woodpecker CI picks up the push and builds your image using Kaniko (a container-friendly image builder). Registry credentials are auto-injected into the pipeline — there's no manual registry login to configure.
4. **Image goes to your registry.** The built image is pushed to your private container registry, tagged with the commit SHA and `latest`.
5. **Auto-deploy rolls it out.** If the app has auto-deploy enabled, the pipeline redeploys it automatically — typically in under a minute from push. Prefer to review a build before it goes live? Turn auto-deploy off and use "Release latest build" in the portal when you're ready.

A live timeline in the portal (Pushed → Built → Deployed → Live) shows exactly where each push is in this pipeline.

App kinds

When you create an app in the portal, you choose a kind:

- **web** — a long-running service exposed on a port, with a public URL.
- **cron** — a scheduled job that runs on a cron schedule you define.
- **worker** — a long-running background process with no public URL.

Where to go for detail

This page is the overview. Once you're signed in, your portal has a **Developer Guide** tailored to your tenant — it shows your actual registry URL, app URL pattern, and a pipeline template pre-filled for your setup. That's the reference to use once you're actually wiring up a repo.

Revision #1

Created 2026-07-08 21:01:56 UTC by Admin

Updated 2026-07-08 21:01:56 UTC by Admin