

Getting Started with CompactHive

A short introduction to CompactHive: what it is, your first steps, included services, deploying your app, Hive Code, and trial/support details.

- [What is CompactHive](#)
- [Your first 15 minutes](#)
- [Included services](#)
- [Deploying your app](#)
- [Hive Code \(beta\)](#)
- [Trial, plans & support](#)

What is CompactHive

CompactHive is a self-hosted, managed infrastructure platform built for small technical teams — the 2 to 8 person indie and bootstrapped product teams who need real infrastructure but don't have a dedicated DevOps engineer. Instead of stitching together a dozen separate SaaS tools and cloud accounts, you get one integrated stack: Git hosting, CI/CD pipelines, managed databases, monitoring and status dashboards, an in-portal coding agent, and documentation — provisioned automatically the moment you sign up, and managed for you from then on.

The idea is simple: you should be able to push code and have it built, deployed, and monitored without becoming an infrastructure specialist first. CompactHive handles the plumbing — pipelines, registries, TLS, backups, dashboards — so your team can spend its time on the product instead of the platform underneath it.

Every tenant gets their own isolated namespace with its own set of services, sized to the tier you're on. Basic tier starts as a free 14-day trial with no card required, so you can kick the tyres before committing to anything.

Where to go next

- compacthive.io — the marketing site, pricing, and sign-up flow.
- status.compacthive.io — live platform status: is the platform up, can new tenants sign up, are deploys succeeding.
- The rest of this book walks through your first 15 minutes on the platform, what services are included, how deployment works, and what to expect from the trial and support.

If you're reading this because a link in the site footer brought you here — welcome. The next page picks up right where sign-up leaves off.

Your first 15 minutes

Here's what happens between clicking "sign up" and having a working environment.

1. Sign up for the trial

Sign up starts you on the **Basic tier as a 14-day free trial** — no card required. You just need an email and a password (or an SSO provider, if one is offered on the sign-up page) to get going.

2. Auto-provisioning

As soon as your account is created, CompactHive automatically provisions your tenant in the background: a dedicated namespace, your included services, a Git integration, and a starting registry. You don't file a ticket or wait for a human — provisioning kicks off immediately after signup.

While this is happening, the portal and the in-browser coding page will show a "setting up" state and poll until everything is ready. This usually takes a short while; you don't need to refresh manually.

3. What appears in the portal

Once provisioning finishes, your portal dashboard becomes your home base. From there you can:

- Connect a Git repository (GitHub, GitLab, or the built-in Gitea instance) so pushes trigger builds.
- Open **Hive Code**, the in-portal coding agent, to start building directly in the browser.
- See your included services (monitoring, database, CI, registry, and more depending on tier).
- Track deploys through a live timeline: Pushed → Built → Deployed → Live.

4. The onboarding checklist

A short checklist is pinned to the top of your portal sidebar to guide the first steps:

1. Start building your app (via Hive Code or your own repo).
2. Get your first successful build.
3. Confirm your app is reachable (live).
4. Invite a teammate or add a custom domain.

You can dismiss the checklist at any point — it's a guide, not a gate. The next page covers exactly which services come with your tenant.

Included services

Every CompactHive tenant gets a set of managed services provisioned into their namespace automatically. Which ones you have depends on your tier — higher tiers add more.

Service	What it is	Available from
Monitoring	Metrics, logs, and dashboards for your namespace (built on Prometheus, Loki, and Grafana)	All tiers
Uptime Kuma	Uptime monitoring and status checks for your apps	All tiers
Woodpecker CI	Continuous integration that builds your app on every push	Basic and up
PostgreSQL	A managed database for your application	Basic and up
Container Registry	A private registry that stores the images your CI pipeline builds	Basic and up
Gitea	A built-in, self-hosted Git server — an alternative to bringing your own GitHub/GitLab	All tiers (opt-in)
MinIO (S3-compatible storage)	Object storage for files, backups, and uploads	Team and up
KeyDB (Redis-compatible cache)	An in-memory cache/data store for your app	Team and up
FileBrowser	A web UI for browsing and managing files in your namespace	Business and up
Headscale VPN	Private network access into your namespace's services	Business and up

A few notes on how to read this:

- "All tiers" includes the Basic trial — you don't need to be on a paid tier to get monitoring, uptime checks, or Gitea.
- Services scale with your tier's resource allocation (CPU/RAM/storage), not just availability — a Team-tier database has more headroom than a Basic-tier one.
- You don't have to configure any of this by hand. Services appear in your portal already wired up; you connect your app to them.

For exact tier resource sizes and current pricing, see the [pricing page](#). The next page covers how your code actually gets from a `git push` to a live URL.

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.

Hive Code (beta)

Hive Code is currently in beta.

Hive Code is CompactHive's in-portal coding agent, available at `/portal/code`. It lets you build an app by describing what you want in plain language, directly in your browser — no local dev environment required.

How it works

1. **Bring your own model credentials.** Before starting a session, connect a model: either an API key (OpenAI or Anthropic are supported as a fallback option), or, where available, a connected subscription. CompactHive doesn't include model access as part of your plan — you're using your own credentials.
2. **Pick or create a repo.** You can point Hive Code at an existing repository in your built-in Gitea instance, or have it create a brand-new one for you.
3. **Describe what you want.** The agent writes code against your repo, running server-side in your own tenant namespace.
4. **It ships itself.** When the agent finishes a piece of work, it commits and pushes the change back to your repo. From there, the normal CI/CD pipeline takes over exactly as if you'd pushed the code yourself: Woodpecker builds the image with Kaniko, and — if auto-deploy is on — it goes live. The same Pushed → Built → Deployed → Live timeline applies.

There are no extra buttons to click to "ship" what the agent builds — pushing is shipping, same as it is for humans on the platform.

Current limitations

- Hive Code works against repositories in the built-in Gitea instance. If your tenant is connected to GitHub instead, use the standard "Connect GitHub" flow in Settings to build and push manually; direct GitHub-backed Hive Code sessions are on the roadmap but not available yet.
- As a beta feature, expect the experience to keep evolving — if something feels rough around the edges, that's expected right now, not a sign you're doing it wrong.

Hive Code is included on all tiers; you only need to supply your own model credentials to use it.

Trial, plans & support

Trial lifecycle

Signing up puts you on the **Basic tier as a 14-day free trial**, no card required. Here's what happens if you don't add a payment method:

1. **Day 0-14: Trial.** Full access to your Basic-tier tenant.
2. **Day 14-21: Grace period (7 days).** Your trial has ended, but your tenant keeps running. You can add a card at any point during this window to continue without interruption.
3. **Day 21-51: Sleeping (30 days).** If no payment method is added, your tenant is paused — its workloads are scaled down, but your data (databases, files, repos) is preserved untouched.
4. **Day 51: Deletion.** If the tenant is still unpaid after the sleep period, it and all its data are permanently deleted.

That's 51 days total from signup to deletion if nothing is done — but **adding a payment method at any point in that window immediately reactivates your tenant**, no matter which stage it's in.

Plans & pricing

CompactHive offers a small number of tiers beyond Basic, each with more CPU, memory, storage, and additional services (see the previous page for what's included at each tier). Current plans and pricing are kept up to date on the pricing page rather than duplicated here:

- compacthive.io/pricing

Support

- **Email:** support@compacthive.io — the primary channel for bug reports, questions, and feature requests.

- **Status page:** status.compacthive.io — live status for the platform, sign-ups, tenant deploys, and custom apps, independent of the main platform infrastructure.

A note on uptime

CompactHive's shared tiers (which is what you're on unless you've specifically arranged private infrastructure) are run on a **best-effort basis** — there's no percentage uptime commitment or service-credit guarantee attached to them. We work to keep things running well and respond to incidents promptly, but if strict contractual uptime guarantees are a hard requirement for you, that's a conversation to have with us directly rather than an assumption to make from a shared tier.