CI/CD Pipelines

How CI/CD works on CompactHive

CompactHive uses Woodpecker CI for continuous integration and delivery. It runs inside your Kubernetes namespace using the Kubernetes backend — each pipeline step executes as its own pod.

- **Woodpecker Server** — receives webhooks from your Git provider (Forgejo, or GitHub in beta) and stores pipeline state.
- **Woodpecker Agent** — picks up jobs and creates pipeline pods in your namespace.
- **Kaniko** — builds your container image without a Docker daemon (runs as a regular pod).
- **Your private registry** — stores the images Kaniko builds.

Setting up your pipeline

1. **Registry secrets (automatic).** `registry_url`, `registry_user`, and `registry_password` are auto-injected into Woodpecker during provisioning — no manual setup needed. You can verify they exist in Woodpecker under **Settings → Secrets** (they appear as global secrets).
2. **Pipeline file.** Add a `.woodpecker.yaml` to your repository root. The portal's Developer Guide generates a ready-to-use template, pre-filled with your tenant's registry path.
3. **Push.** Woodpecker picks up the push via webhook, creates a pipeline pod, builds with Kaniko, and pushes to your registry.

Pipeline template

This is the general shape of a CompactHive `.woodpecker.yaml`. Your portal's Developer Guide shows the exact version with your registry project path already filled in.

```
# Registry secrets are auto-injected — no manual setup needed.
steps:
  - name: build
    image: ghcr.io/osscontainertools/kaniko:v1.28.1-debug
```

```
    commands:
      - mkdir -p /kaniko/.docker
      - |
        cat > /kaniko/.docker/config.json << DOCKEREOF
        {
          "auths": {
            "$${REGISTRY_URL}": {
              "username": "$${REGISTRY_USER}",
              "password": "$${REGISTRY_PASSWORD}"
            }
          }
        }
        DOCKEREOF
      - /kaniko/executor
          --context=$CI_WORKSPACE
          --dockerfile=Dockerfile
          --destination=$${REGISTRY_URL}/YOUR_PROJECT/$CI_REPO_NAME:$CI_COMMIT_SHA
          --destination=$${REGISTRY_URL}/YOUR_PROJECT/$CI_REPO_NAME:latest
          --cache=true
          --insecure
    environment:
      REGISTRY_URL: { from_secret: registry_url }
      REGISTRY_USER: { from_secret: registry_user }
      REGISTRY_PASSWORD: { from_secret: registry_password }
```

Notes:

- `YOUR_PROJECT` is your registry project name (shown on the portal Connect page).
- **Two tags per build** — the commit SHA (for rollback) and `:latest` (for convenience).
- `--insecure` is required because, inside the cluster, the registry is reached over plain HTTP (TLS is terminated at the edge, in front of the registry's public hostname).
- `--cache=true` enables Kaniko layer caching for faster rebuilds.

Customization

Only build on your main branch:

```
steps:
  - name: build
    image: ghcr.io/osscontainertools/kaniko:v1.28.1-debug
    when:
```

```
      - branch: main
```

Add a test step before the build:

```
steps:
  - name: test
    image: node:20-alpine
    commands:
      - npm ci
      - npm test

  - name: build
    image: ghcr.io/osscontainertools/kaniko:v1.28.1-debug
    depends_on: [test]
```

Useful Woodpecker built-in variables

Variable	Example	Description
`CI_REPO_NAME`	`my-app`	Repository name
`CI_COMMIT_SHA`	`a1b2c3d`	Full commit hash
`CI_COMMIT_BRANCH`	`main`	Branch name
`CI_WORKSPACE`	`/woodpecker/src/...`	Workspace path (source code)
`CI_PIPELINE_EVENT`	`push`	Event type (push, pull_request, tag)

See Woodpecker's own documentation for advanced features (matrix builds, cron, services, volumes) not covered here.

Troubleshooting

Problem	Solution
Pipeline not triggering	Make sure the repo is **activated** in the Woodpecker UI. For GitHub (beta), check webhook delivery in your GitHub repo settings.

Problem	Solution
Kaniko auth error	Verify the 3 registry secrets exist and are correct — check for typos if you ever set them manually.
Build timeout	Lower tiers have less CI CPU/memory headroom. Multi-stage Dockerfiles that shrink the build context usually help.
Image not found on deploy	Check that the image tag on the Apps page matches what the pipeline pushed — the Registry page in the portal shows what's actually there.

Revision #1
Created 2026-11-03 14:21:27 UTC by Docs Sync
Updated 2026-11-03 14:21:27 UTC by Docs Sync