DETERMINISTIC DELIVERY

Keep control of your config

A controllable foundation for Helm, GitOps, and AI-generated config — so what you review is exactly what deploys.

ConfigHub · cub installer · helm-expt · cub-scout

For telco platform & delivery teams who need carrier-grade, repeatable rollouts

Building ConfigHub

THE SHIFT

New workloads

- Kubernetes and GitOps are the home for cloud-native apps.
- AI inference now consolidates onto that same Kubernetes — GPUs, model serving, agents.
- The next platform shift runs AI on the same substrate as K8s and this needs a new system of record for configuration.

WHO IS BUILDING IT

BG **Brian Grant**
Co-creator of Kubernetes · ex-Google

AR **Alexis Richardson**
Founder of GitOps · ex-Weaveworks (CEO)

JJ **Jesper Joergensen**
Co-founder · enterprise platforms

Every era of operations has a system of record

1990s–2000s

IT Ops

Mainframes, servers

SYSTEM OF RECORD

ServiceNow

2010s

Cloud Ops

AWS · Azure · GCP

SYSTEM OF RECORD

Atlassian / Jira

2015–2024

DIY DevOps

Kubernetes · Git · OCI · CNCF

SYSTEM OF RECORD

GitHub · GitLab · JFrog

2025 +

Agentic Ops

AI + Kubernetes + CNCF

SYSTEM OF RECORD

ConfigHub & …

Use AI to analyse and operate K8s applications – Add AI to enhance K8s apps – Run agentic dev/test sandbox on K8s – Run K8s on GPU stacks

A problem you already know

1. **Carrier-grade means deterministic** Repeatable, auditable, identical every rollout — across regions, clusters, and maintenance windows.

2. **Helm is a package tool not a safe operational API** What you reviewed often isn't what actually deployed — hidden hooks, lookups, and version drift. Surprises!

3. **AI now generates config at scale** More variants, more overlays, less human review. Non-determinism compounds, fast.

THE RULE

If you add non-determinism on top,
you need a fully controllable foundation underneath.

That foundation is a source of desired-state truth.

Sylva friends please help me today by commenting on my Generative GitOps doc

THE PAIN

When Helm bites: five surprises

Hidden hooks & lifecycle

Secrets and certs are generated at install time, not in your manifest. A config-only apply leaves you “created but not working.”

Quirky, drifting rendering

helm template ≠ what deploys. Lookups, capabilities and ordering shift across chart versions and value sets.

Image & registry surprises

Charts pin images that aren’t pullable on your target — ImagePullBackOff on an otherwise “valid” release.

CRDs, webhooks & ordering

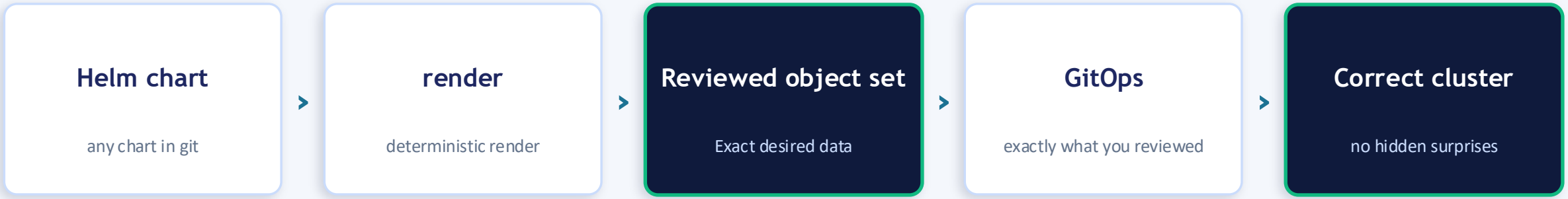
Admission certs and CRDs must land in sequence. One bulk apply fails; partial state is hard to reason about.

Day-2 drift

Upgrades and rollbacks are risky with no source of desired-state truth to diff against. Surprises in production.

THE FOUNDATION

How we could solve this



Complements git, OCI & GitOps

It doesn’t replace them — it makes them deterministic. Git stays source, OCI stays artifacts, GitOps stays delivery.

A graph of variants

We can store the rendered, reviewed object set as flat YAML: the exact manifest that will apply plus the up/down dependencies and chain of provenance

Helm behaves like plain manifests

No hidden surprises at apply time. Governed, auditable, repeatable — the same every rollout. And not just for Helm – for all config and ops.

WHAT CONFIGHUB IS

Configuration as data, not text

One source of truth for operational configuration — what **GitHub became for source code**. Install once, customize with typed "Variants," and every change carries who changed it, why, and a signed proof.

1

Stored as data

Every value lives literally in the config — no templates, no hidden variables. Humans read it; machines query, validate, and edit it through one API.

2

Customized by Variants

Install the canonical package once; layer thin typed deltas on top. Vendor updates merge cleanly — no per-customer forks of the whole stack.

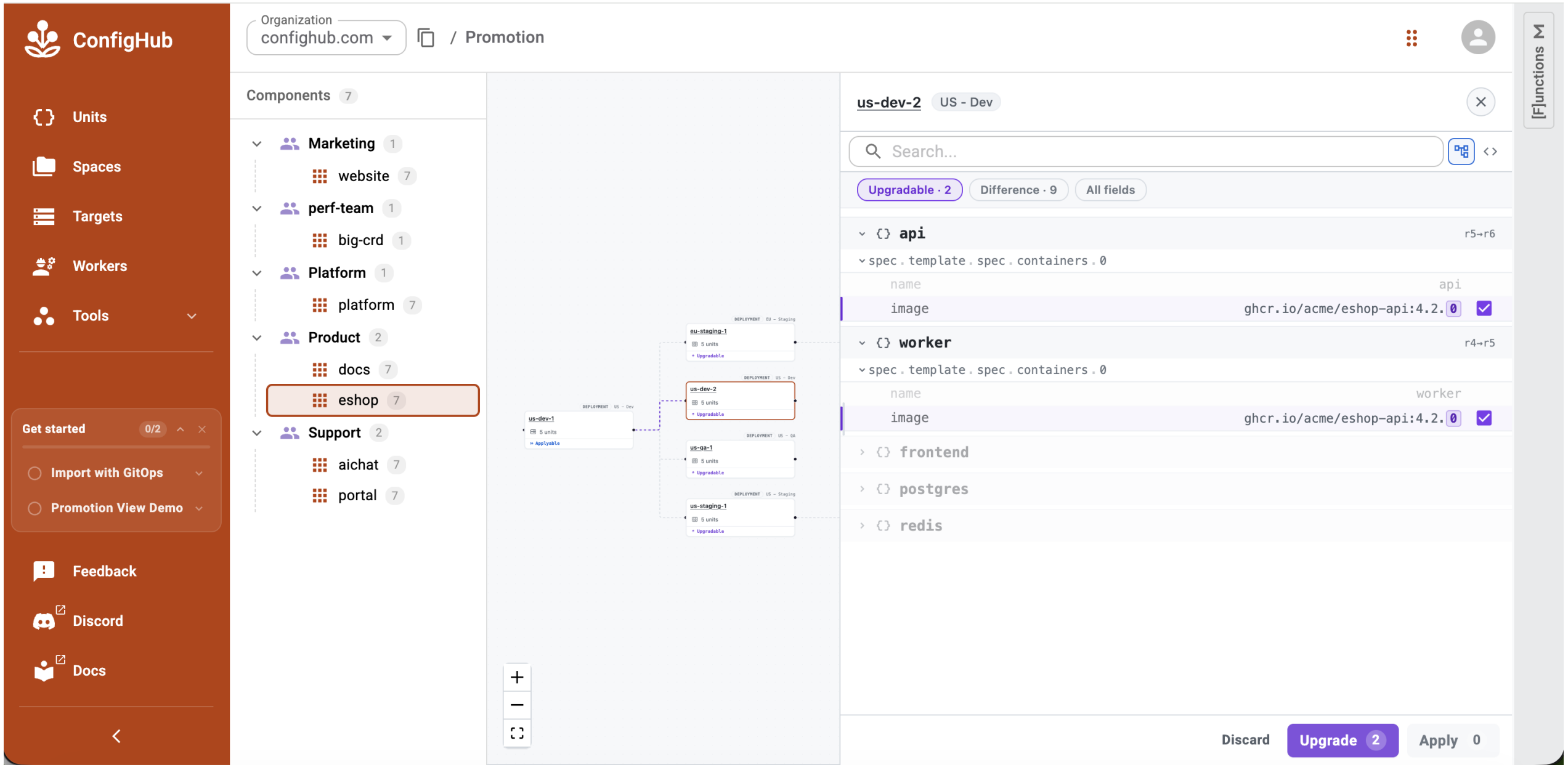
3

Governed by construction

Validation, policy and a signed proof chain run before anything is applied. The operation is the audit record — not a log written afterward.

AI agents produce Variants, not Helm patches. Mass customization that stays governed — safe agentic by construction.

ConfigHub Enterprise and Cloud editions



eu-staging-1 EU - Staging

Search...

Upgradable · 2 | Difference · 22 | All fields

{} **api**	r5→r6
metadata	
namespace	eu-staging-1-eshop
spec	
replicas	2
template . spec . containers . 0	
name	api
image	ghcr.io/acme/eshop-api:4.2.1
env	
0	
name	ENVIRONMENT
value	staging
1	
name	REGION
value	eu
2	
name	LOG_LEVEL
value	info
tls . 0 . hosts	
0	eu-staging-1-eshop.demo.confighub.io

Upgrade 0 | Apply 0

image	2.11 → 2.14
resources.memory	128Mi → 256Mi

1 change to Service/api

port	8080 → 9090

worker

ghcr.io/acme/eshop-api:4.2.0

eshop-ap 6m ago

↑ INCOMING

ghcr.io/acme/eshop-api:4…

from us-dev-1 r5
4m ago
Functions: set-image-reference

OTHER VARIANTS

us-dev-1

ghcr.io/acme/eshop-api:4…

Discard | Upgrade 2 | Apply 0

Helm in ConfigHub: the theory!
“Install once. Customize your own Variants”

TODAY

The mystery bag

- Helm chart + values overlays
- Kustomize patches stacked on patches
- Per-customer forks duplicate the whole stack
- AI agents producing text munging

→ *Misconfiguration is structural*

CONFIGHUB

Install + customize

- Installer brings canonical package as base Variants
- Customizations are based on deltas
- Vendor updates merge cleanly across customizations
- ConfigHub users produce Variants, not Helm patches

→ *Customization is governed by construction*

Usage:

Attach to existing OCI infrastructure (Harbor, JFrog, ECR, ACR, GHCR + ConfigHub embeds an OCI server).
The installer brings apps in; Users & agents customize via governed Variants.
No new stack to deploy.

LIVE

Demo: governed promotion, deterministically

Each step writes a receipt

Promote, don’t patch

A reviewed variant moves through a ConfigHub ChangeSet — not an ad-hoc helm upgrade.

The bytes you reviewed land

Downstream gets exactly what was approved, with a receipt that proves it.

Convergence is witnessed

cub-scout confirms the workloads actually reached the desired state after promotion.

Also live in the GUI: ConfigHub Units, diffs, and the promotion ChangeSet.

FUTURE: how enterprise can manage configuration

Git, Helm and YAML treat configuration as text — fine when humans edit occasionally. Agents change config at machine speed and machine scale. Text can't keep up. There is just too much friction and things break.

DIGITAL COMPANY

“Git is not enough.”

Git breaks at scale. Their pattern: agentic cloud is single-tenant, high-volume, massively customized — config can't live in templates.

CLOUD APP PLATFORM

“Git has file ops, not data ops.”

The AI platform needs a data API. Agents can wrangle files — but a structured, queryable config layer is fundamentally better.

Duct-tape YAML + Helm worked early on. **At AI pace it could be a 5–10× liability.** **Let’s get ready.**

“Goal” Governance, security, compliance baked in (paid edition)

FOR OPERATORS

We stop misconfiguration.

- Typed schemas catch errors at edit time, not in production.
- Every Variant passes validation + policy eval before apply.
- Proof of evaluation is attached, not asserted out-of-band.

STOP production incidents by analysing desired state

FOR COMPLIANCE

AI audit + regulatory ready.

- Every change traceable to agent + model + session + prompt.
- Policy enforcement agents can't bypass — wired into API
- SOC 2, financial services rules, EU AI Act — by construction.

Safe agentic, not retrofitted agentic.

Achieve compliance by forcing correct constructions with policy gates – no need to “add integrated products”

OSS tools @ http://github.com/confighub

`cub installer`

Deliver the exact object set

- Pull a chart → render → review → apply.
- Plain kubectl at apply — no plugin, no surprises.
- The bytes you reviewed are the bytes that land.

`helm-expt`

Proof, not promises

- Public catalog + proof harness over 100 charts.
- Every chart run through render, live & two-cluster lanes.
- Each result is a committed, re-checkable receipt.

`cub-scout`

Witness the live truth

- object-set-matches · prerequisites-met
- workloads-converged
- Proves it’s working — not just created.

Add AI. Keep determinism.

- Desired-state truth under git, OCI & GitOps.
- Helm behaves like fully-rendered manifests.
- Safer promotions and rollouts with compliance

Try it today at ConfigHub.com or email alexis@confighub.com

confighub.com → sign up · github.com/confighub/