

kubara Community Call

8th of July





First, catching up on **v0.10.0**:

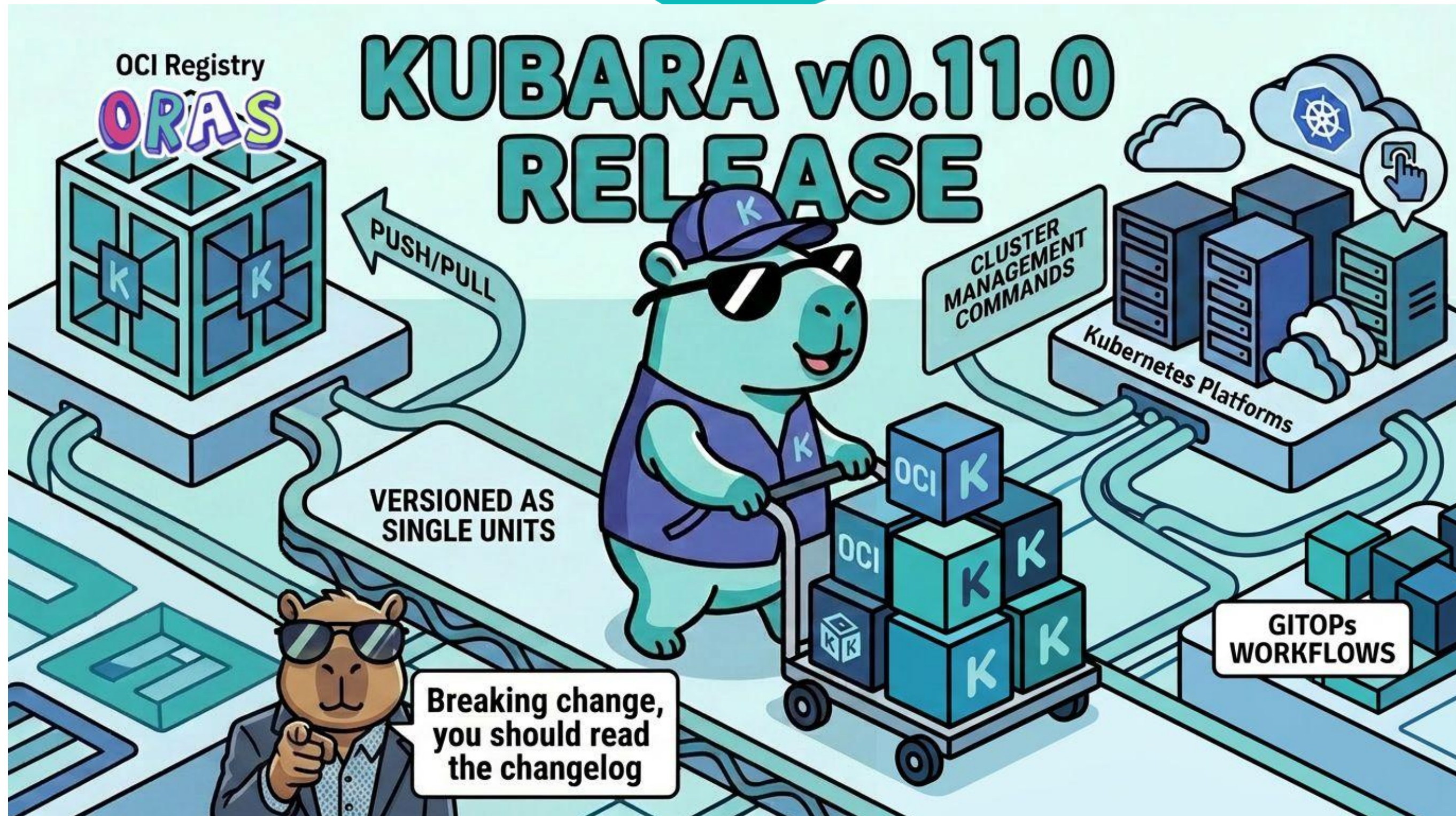
kubara now runs fully local on kind: test-drive the whole kubara workflow, complete Helm stack included – faster, and no cloud account needed. And Velero joined the built-in services for backup & recovery.

```
$ kubara init --local
$ kubara bootstrap

# spins up a kind cluster with Argo CD,
# OpenBao & Co. – no cloud account needed
```

```
# config.yaml
services:
  velero:
    status: enabled

# backup & recovery, built in
```



Catalogs can now be **packaged, versioned and published** to any registry – just like Helm charts:

```
$ kubara catalog package
```

```
# packs your catalog into the local cache
```

```
$ kubara catalog login ghcr.io --username you
```

```
# authenticate against any OCI registry
```

```
$ kubara catalog push oci://ghcr.io/acme/platform:0.1.0
```

```
# publish it as a versioned artifact
```

```
$ kubara catalog pull oci://ghcr.io/acme/platform:0.1.0
```

```
# consume it anywhere – just like helm
```

https://docs.kubara.io/latest-stable/2_concepts/catalog_distribution/
https://docs.kubara.io/latest-stable/2_concepts/catalog_templating/

Terraform support for **T Cloud Public** is here: kubara provisions the full stack – CCE clusters, load balancers and everything around them.

T Cloud Public offers no managed secrets manager, so kubara deploys OpenBao for you – and Helm charts arrive with matching provider values, Velero included.

```
# config.yaml
clusters:
  - name: hub
    terraform:
      provider: t-cloud-public
      projectId: my-tenant
```



kubara now onboards **AI coding agents** out of the box:

kubara agents scaffolds AGENTS.md right in your platform repo: it tells the agent where docs (Markdown + llms.txt), source code and first steps live – no searching, no crawling, no wasted tokens.

AGENTS.md

Agent guide for this kubara platform repository

This repository is managed with **kubara** – an opinionated, GitOps-first CLI (single Go binary). You are likely a coding agent (Claude Code, Codex, ...): this is your token-lean entry point.

- Installed kubara version: **v0.11.0**
- Command reference: **kubara --help**
- Config reference: **kubara schema** → config.yaml
- Rendered docs: docs.kubara.io
- Full docs index: llms.txt

Core workflow (run in this order)

1. Be in a git repo with a config.yaml
2. **kubara init** – generate config.yaml
3. **kubara generate --helm** – render artifacts
4. **kubara bootstrap <cluster-name>**

5. Commit, push, and apply CD, done! (the rest is automated)

llms.txt

kubara – Docs

- [Home](https://docs.kubara.io/v0.11.0/index.md)

Getting Started

- [Quick Start](https://docs.kubara.io/v0.11.0/1_getting_started/quick_start/index.md)
- [Installation](https://docs.kubara.io/v0.11.0/1_getting_started/installation/index.md)
- [Prerequisites](https://docs.kubara.io/v0.11.0/1_getting_started/prerequisites/index.md)
- [Bootstrapping](https://docs.kubara.io/v0.11.0/1_getting_started/bootstrapping/index.md)

Concepts

- [Overview](https://docs.kubara.io/v0.11.0/2_concepts/overview_core_concept/index.md)
- [Platform Engineering](https://docs.kubara.io/v0.11.0/2_concepts/gitops_platform_engineering/index.md)
- [Catalogs](https://docs.kubara.io/v0.11.0/2_concepts/catalogs/index.md)
- [Catalog Distribution](https://docs.kubara.io/v0.11.0/2_concepts/catalog_distribution/index.md)
- [Catalog Templating](https://docs.kubara.io/v0.11.0/2_concepts/catalog_templating/index.md)

Infrastructure Presets

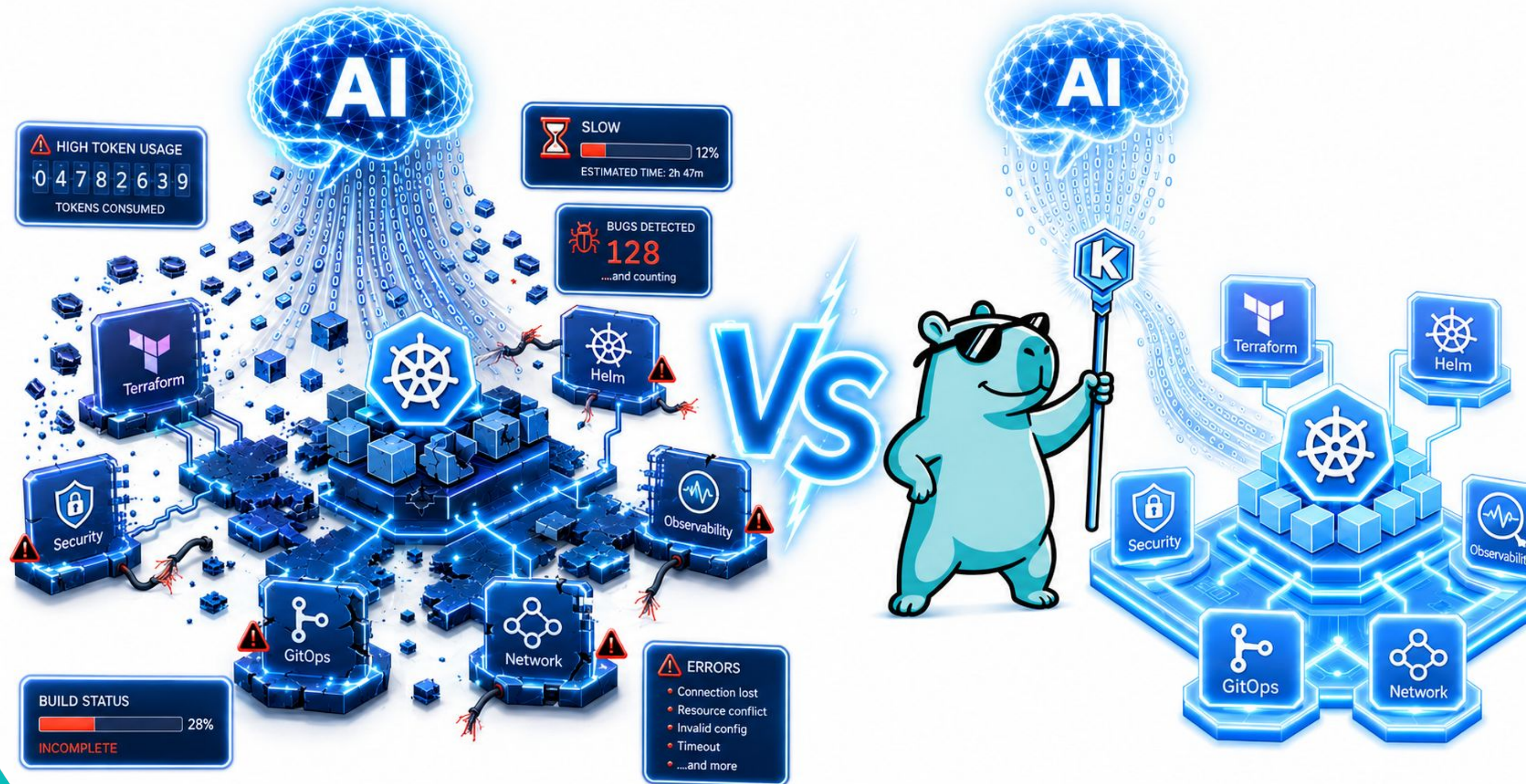
Export all infrastructure as a single Helm chart or as a set of Terraform modules.



<https://docs.kubara.io/>

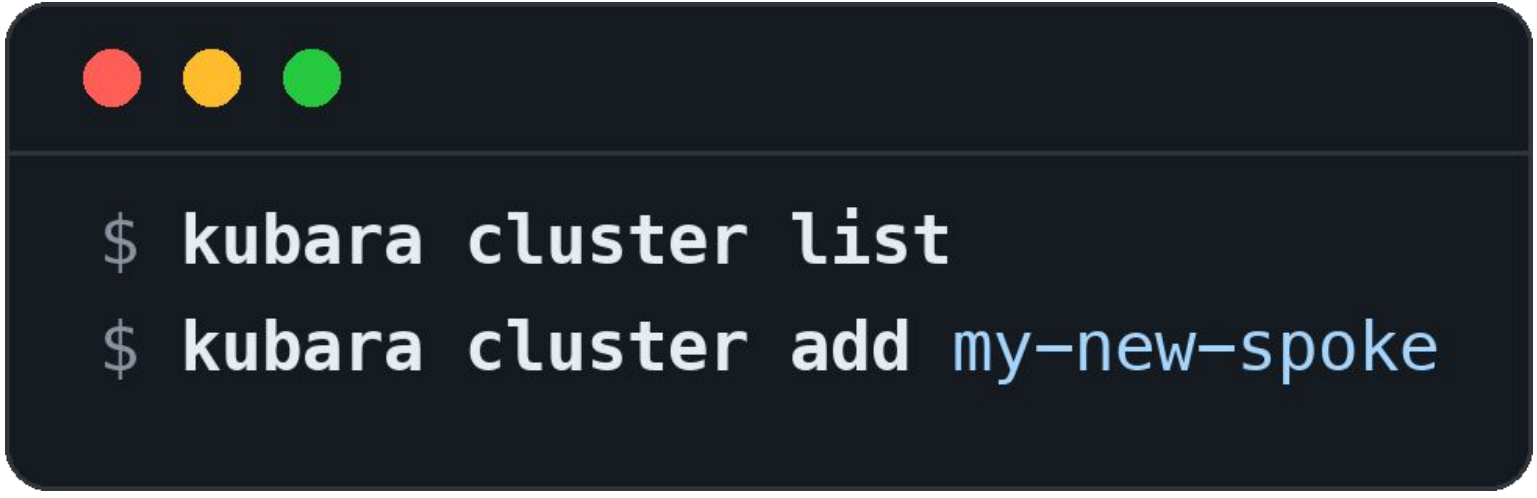
We put it to the test.

Claude Code & Codex building a Kubernetes platform,
from scratch vs. with kubara – **full story coming to the blog soon.**

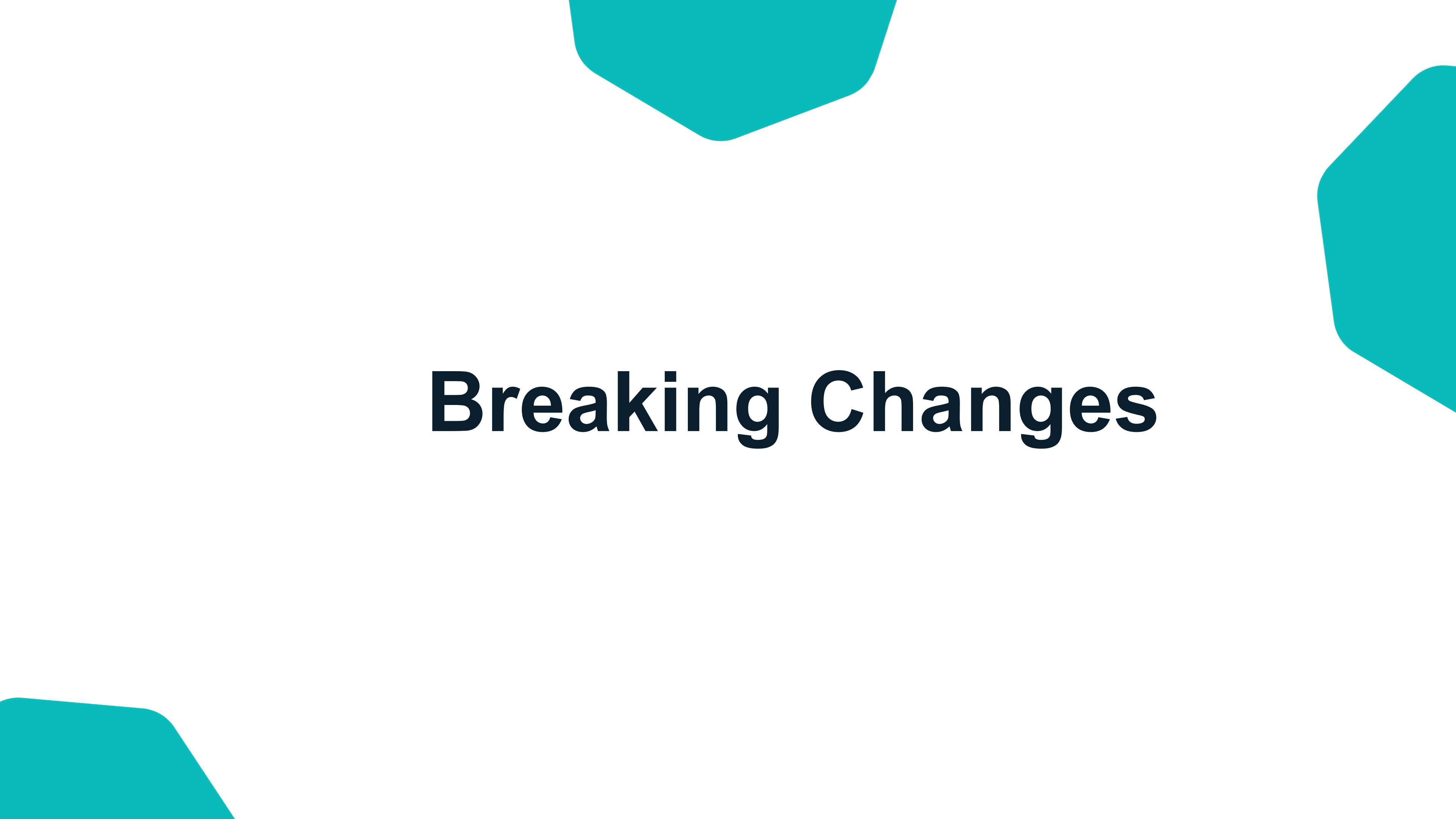


Spoke clusters now join with **one command** – straight from the CLI.

And rendering got smarter: only charts you actually use are generated, and activated spoke services are added to Argo CD on the hub automatically.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains two lines of text: a prompt followed by 'kubara cluster list' and another prompt followed by 'kubara cluster add my-new-spoke'.

```
$ kubara cluster list  
$ kubara cluster add my-new-spoke
```

The background features three teal-colored geometric shapes: a hexagon at the top center, a pentagon on the right side, and a trapezoid at the bottom left.

Breaking Changes

Cluster Secret Stores are now a **map**, and secrets moved to scalable per-cluster paths.

Using our STACKIT SKE Terraform? Secrets are migrated automatically – run **terraform plan & apply** before pushing a newly rendered state.

before

```
# additional-values.yaml
clusterSecretStores:
  - name: my-cluster
    provider: ...

# secrets at the root of
# your secret manager
<secret>
```

v0.11.0

```
# additional-values.yaml
clusterSecretStores:
  my-cluster:
    provider: ...

# scalable secret paths
<cluster>/<stage>/<namespace>/<secret>
<cluster>/<stage>/cluster_secrets/<secret>
```

Already built a catalog? Add **spec.version** once – that's the whole migration.

Versions are what make catalogs packageable and distributable as registry artifacts.

```
# Catalog.yaml
apiVersion: kubara.io/v1alpha1
kind: Catalog
metadata:
  name: your-catalog
spec:
  version: 0.1.0
```

Questions? Feedback? Stories?

The floor is yours – we want to hear from you.

Next community call: Wednesday, August 12, 2 PM CEST.



Join us!
Build with us!

<https://github.com/kubara-io/kubara>
<https://kubernetes.slack.com/archives/C0B2X0DLPR6>

Star us!

