

Meet the community



COME
JOIN US!



SAVE THE DATE

14th of October 2026

Agenda

What happened? 🙄

Event Recap
What's new?

Future 🔮

What's planned in the near future?
Which Events will we attend.

Community 💡

We want to hear from you!
Do you have a story to tell?
Do you have questions?

What's new?

Release v0.17.0

- <https://github.com/kubara-io/kubara/releases/tag/v0.17.0>
- This release introduces two breaking changes:
 - Only one hub per config file
 - Env variable parsing in the manifest templating flow was removed for the sake of security.

Versioning of Catalogs

- Releases are done **automatically twice per week** and starting tomorrow (15th of October) releases will be using temporal Versioning (CalVer) with the following pattern:
<year>.<week>.<releaseNumber>
- Tomorrow's release will therefore be **2026.42.2**

Catalog highlights

- Crossplane as an optional component 🎉 @Brilly-Bohyun
- Improved monitoring (json logging, fixed labels and scrapping paths) 🤖 @Matthiator
- Traefik middleware for http -> https redirect

Community Highlights

❑ 🐙 feat: add Gateway API networking configuration

#630 · Brilly-Bohyun opened last month Contributor · 8 of 13 tasks

· Changes requested · 9/9 · 1.0.0

💬 22

❑ 🐙 feat(general): add Crossplane as an optional component

#141 · by Brilly-Bohyun was merged last month Contributor

· 3 of 14 tasks · Approved · 12/12

🕒 1 💬 6



Thank you for the amazing contributions. Be it code, reviews, feedback or great ideas!



schnodelbotz 10:30 AM

While preparing an internal kubara demo, I've set up the same cluster several times from scratch - to test setup reproducibility, maybe even for disaster recovery purposes. It obviously works perfectly - the only mistake I made was that I forgot to run my `kubara generate`d infrastructure setup scripts before actual `kubara bootstrap`. Given the diversity of infrastructure scripts or terraform needs and personal preferences (terraform -auto-approve anybody? :) ... I thought it would be cool if kubara supported hooks, much like `.git/hooks`. For example, if I run `kubara bootstrap` and a `platform-configs/<my-cluster>/hooks/pre-bootstrap` exists, kubara would run it in that particular deployment stage, automatically. Like in git, the hooks directory could be populated with `.sample` scripts, showing users what they can be useful for. A failing, non-zero return code hook should make kubara abort the process in question.

I believe this could free co-workers from referring to documentation ("what scripts do I need to run when?") and make the process less error prone -- but would love to hear your spontaneous thoughts on this before spamming you further 🙏



4 replies

What's next for kubara? What gets us to 1.0.0?

- [ADR-0004 Integrate Libkubara](#)
- [ADR-0005 Restricting the env file to a minimum \(secrets\)](#)
- [ADR-0006 Generic GitOps Engine](#)
- [ADR-0007 Future Proofing the cluster object](#)
- [ADR-0008 Improve Multi Hub Support](#)

What is Libkubara?

<https://github.com/kubara-io/libkubara>

The core kubara functionality for config processing and templating as a library:

In-process Kubernetes CRD validation, structural defaulting, and manifest handling without a running cluster.

```
// Compile schema and CEL rules directly from CRD YAML
validator, _ := crdvalidate.Compile(
    bytes.NewReader(issuerCRD)
)

// Decode the input manifest
rawObj, _ := manifest.DecodeOneBytes(issuerYAML)

// Run structural schema validation, defaulting
result := validator.ValidateCreate(ctx,
    rawObj, crdvalidateRejectUnknown)

// Unmarshal the defaulted and validated object
// into a typed struct
var issuer certmanagerv1.Issuer
if err := result.Into(&issuer); err != nil {
    log.Fatalf("validation failed:\n%v", err)
}

fmt.Printf("Validated Issuer: %s/%s\n",
    issuerNamespace, issuer.Name)
fmt.Printf("ACME Server: %s\n", issuer.Spec.ACME.Server)
```

Restricting the env file to a minimum

- Currently we have two source of truth `.env` and `config.yaml` to define platform parameters.
- This is first of all confusing and requires the dual command flow `kubara init --prep` and `kubara init`
- We want to get rid of the forced `.env` file usage and instead only rely on a couple environment variables be it as part of an `.env` file or via `export X`
- Further at the moment all commands even generate which doesn't use any contents of `.env` requires the file to exist
- `config.yaml` becomes the single point of truth for everything except secrets that are required during bootstrap

GitOps config and beyond

We will introduce a major change to the config structure.

- A new gitops agnostic root level block with future support for other engines like flux
- Moving the hub cluster into it's own root level **hub** block
- **clusters** gets renamed to **spokes**

```
gitops:
  engine: argocd
  selfManaged: enabled
  repo:
    https:
      configs:
        url: https://github.com/my-org/platform-configs.git
        targetRevision: main
      components:
        url: https://github.com/my-org/platform-components.git
        targetRevision: main
  helmRepo:
    url: https://charts.example.com
```

```
hub:
  name: management-hub
  stage: prod
  dnsName: hub.example.com
  catalogs:
    - oci://ghcr.io/kubara-io/catalogs/general:2.3.0
  services:
    cert-manager:
      status: enabled
    traefik:
      status: enabled
```

```
spokes:
- name: workload-spoke-01
  type: spoke
  stage: prod
  dnsName: spoke01.example.com
  services:
    cert-manager:
      status: enabled
    traefik:
      status: enabled
```

Multi-Hub repository support

<https://github.com/kubara-io/kubara/pull/682>

<https://github.com/kubara-io/kubara/pull/673>

We will introduce support for repository subpaths and kubara commands will work on the root level for all at once or single environments.

Allowing for running `kubara generate` for multiple environments or using the `--hub` flag will allow for selecting single or multiple environments like so:

```
kubara generate --hub dev,test
```

```
your-gitops-repo
├── dev
│   ├── kubara.yaml
│   ├── platform-components
│   └── platform-configs
├── prod
│   ├── kubara.yaml
│   ├── platform-components
│   └── platform-configs
└── test
    ├── kubara.yaml
    ├── platform-components
    └── platform-configs
```

Meet us!

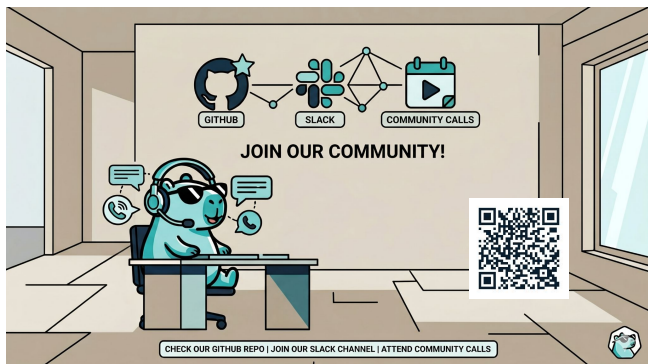
OpenInfraDays

GER - Saarbrücken - 24th - 25th October 2026

KubeCon Europe

ESP - Barcelona - 15th - 18th March 2027

Further Infos



#kubara Slack Channel in official K8s Slack Space

Next Community Call

Nov 11, 2pm CET

Q&A - Questions and Story Time

Lets chat!

Do you have a story to tell?
Do you have questions?