

kubara Community Call

10 June

Join my call



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

v0.9.0 Changes

The old hardcoded service setup has been replaced with dynamic **ServiceDefinition** files.

Built-in services are now defined as YAML files, and service schemas and defaults are generated from these definitions at runtime.

```
kubara / src / internal / catalog / built-in / services / argo-cd.yaml
```

tuunit fix: naming scheme for cluster types (#303)

Code Blame 12 lines (12 loc) · 218 Bytes

```
1 ---
2 apiVersion: kubara.io/v1alpha1
3 kind: ServiceDefinition
4 metadata:
5   name: argocd
6   annotations:
7     kubara.io/category: application-management
8 spec:
9   chartPath: argo-cd
10  status: disabled
11  clusterTypes:
12    - hub
```

kubara / src / internal / catalog / built-in /

kubara-renovate[bot] and la-cc chore(deps):... c374ece · last week History

Name	Last commit message	Last commit da...
..		
customer-service-catalog	chore(deps): update terraform stac...	last week
managed-service-catalog	chore(deps): update terraform stac...	last week
services	fix: naming scheme for cluster type...	3 weeks ago

defaults

Catalog handling is stricter now. Service names must be **canonical**, *apiVersion* must be **kubara.io/v1alpha1**, and every *ServiceDefinition* must define ***spec.status***.

Catalog loading errors are now reported instead of silently falling

kubara / src / internal / catalog / built-in / services / argo-cd.yaml

tuunit fix: naming scheme for cluster types (#303)

Code Blame 12 lines (12 loc) · 218 Bytes

```
1 ---
2 apiVersion: kubara.io/v1alpha1
3 kind: ServiceDefinition
4 metadata:
5   name: argocd
6   annotations:
7     kubara.io/category: application-management
8   spec:
9     chartPath: argo-cd
10    status: disabled
11    clusterTypes:
12      - hub
```

New catalog commands were added:

1

```
kubara catalog create my-catalog  
cd my-catalog  
kubara catalog add widget-dashboard
```

2

```
.  
├── Catalog.yaml  
├── customer-service-catalog  
│   ├── helm  
│   │   ├── example  
│   │   └── terraform  
│   │       └── example  
├── managed-service-catalog  
│   ├── helm  
│   └── terraform  
└── services  
    └── widget-dashboard.yaml  
  
10 directories, 2 files
```

3

```
apiVersion: kubara.io/v1alpha1  
kind: ServiceDefinition  
metadata:  
  name: widget-dashboard  
spec:  
  chartPath: widget-dashboard  
  clusterTypes:  
  - hub  
  - spoke  
  status: disabled
```

4

```
> customer-service-catalog  
> managed-service-catalog  
> my-catalog
```

https://docs.kubara.io/latest-stable/1_getting_started/catalogs/

Existing legacy configs are migrated automatically on first load.
This includes updating old unversioned configs to version:

v1alpha1, normalizing old keys, moving inline config into the new format, and renaming **controlplane / worker** to **hub / spoke**.

v0.8.0

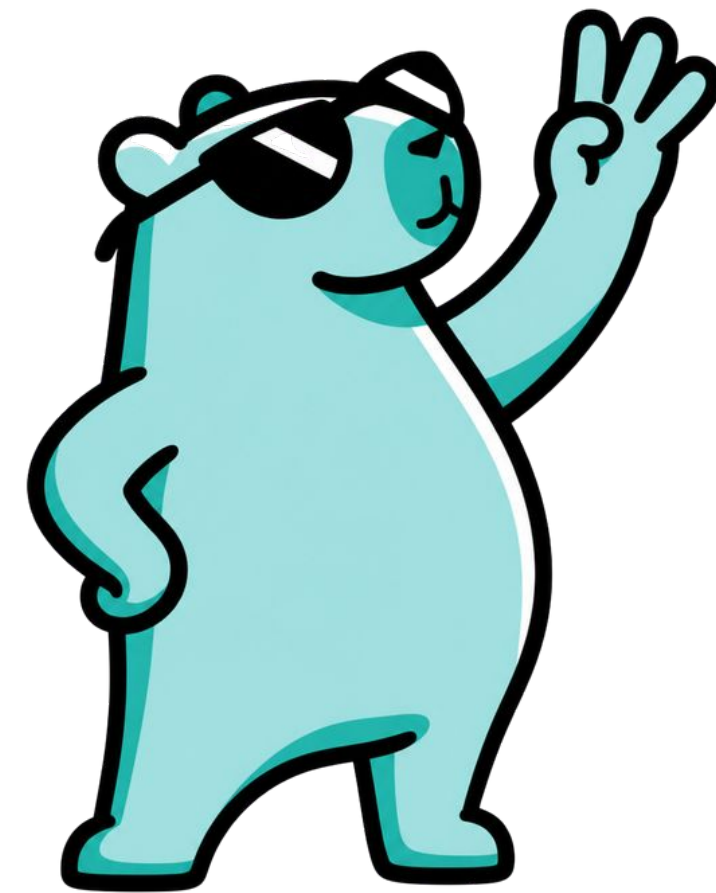
```
clusters:
- name: hub
  stage: dev
  type: controlplane
  dnsName: hub-dev.stackit.run
  ssoOrg: xy
  ssoTeam: xy
  ingressClassName: traefik
  terraform:
    provider: stackit
    projectId: xy
    kubernetesType: ske
    kubernetesVersion: "1.34"
  dns:
    name: hub-dev.stackit.run
    email: my-test@nowhere.com
  argocd:
    repo:
      https:
        customer:
          url: https://github.com/la-cc/demos.git
          targetRevision: main
        managed:
          url: https://github.com/la-cc/demos.git
          targetRevision: main
    helmRepo:
      url: '*'
  services:
    argocd:
      status: disabled
    certManager:
      status: enabled
      clusterIssuer:
        name: letsencrypt-staging
        email: yourname@your-domain.de
        server: https://acme-staging-v02.api.letsencrypt.org/directory
  externalDns:
```



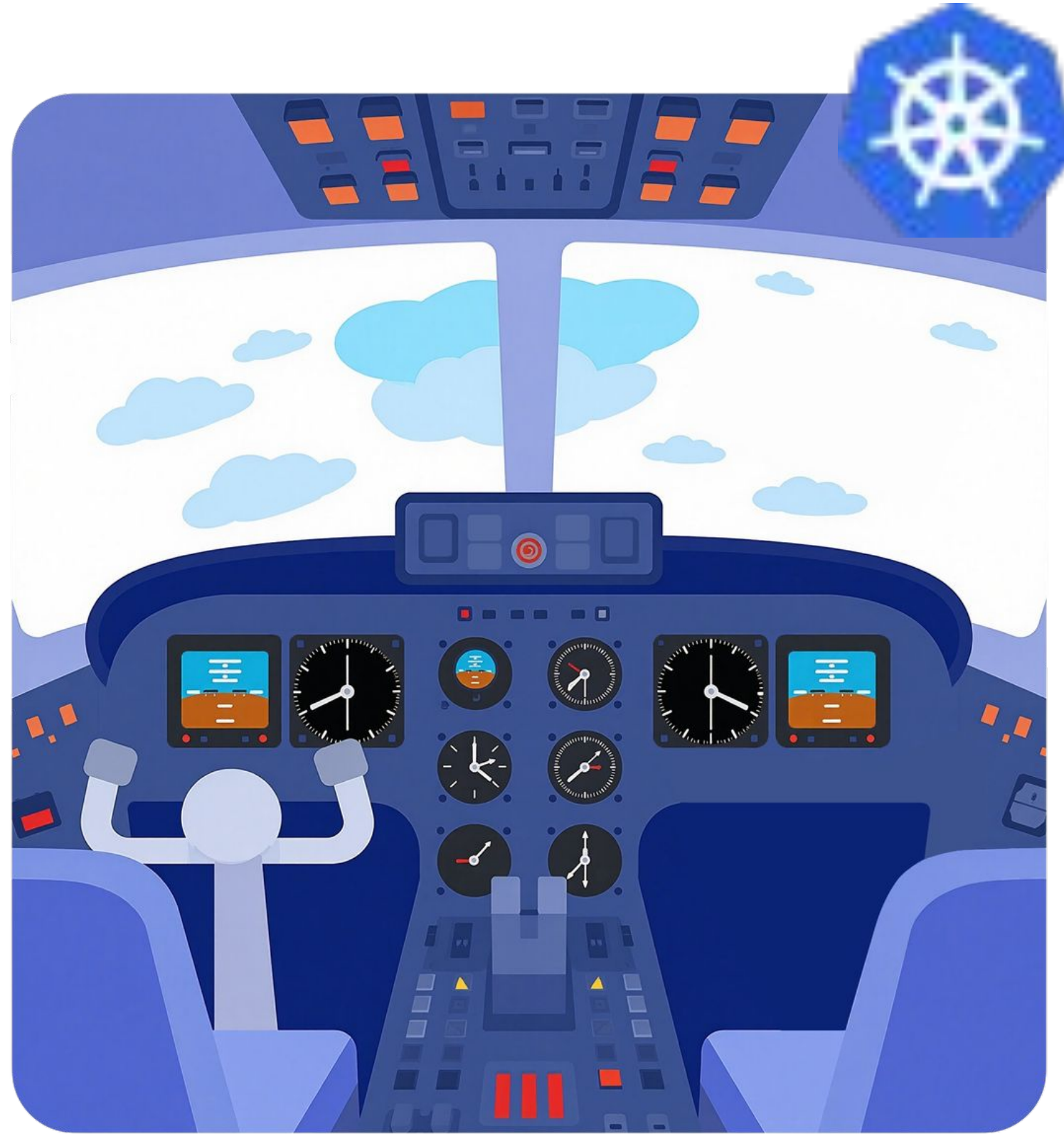
v0.9.0

```
clusters:
- name: hub
  stage: dev
  type: hub
  dnsName: hub-dev.stackit.run
  ssoOrg: xy
  ssoTeam: xy
  ingressClassName: traefik
  terraform:
    provider: stackit
    projectId: xy
    kubernetesType: ske
    kubernetesVersion: "1.34"
  dns:
    name: hub-dev.stackit.run
    email: my-test@nowhere.com
  argocd:
    repo:
      https:
        customer:
          url: https://github.com/la-cc/demos.git
          targetRevision: main
        managed:
          url: https://github.com/la-cc/demos.git
          targetRevision: main
    helmRepo:
      url: '*'
  services:
    argocd:
      status: disabled
    cert-manager:
      status: enabled
      config:
        clusterIssuer:
          email: yourname@your-domain.de
          name: letsencrypt-staging
          server: https://acme-staging-v02.api.letsencrypt.org/directory
  external-dns:
```

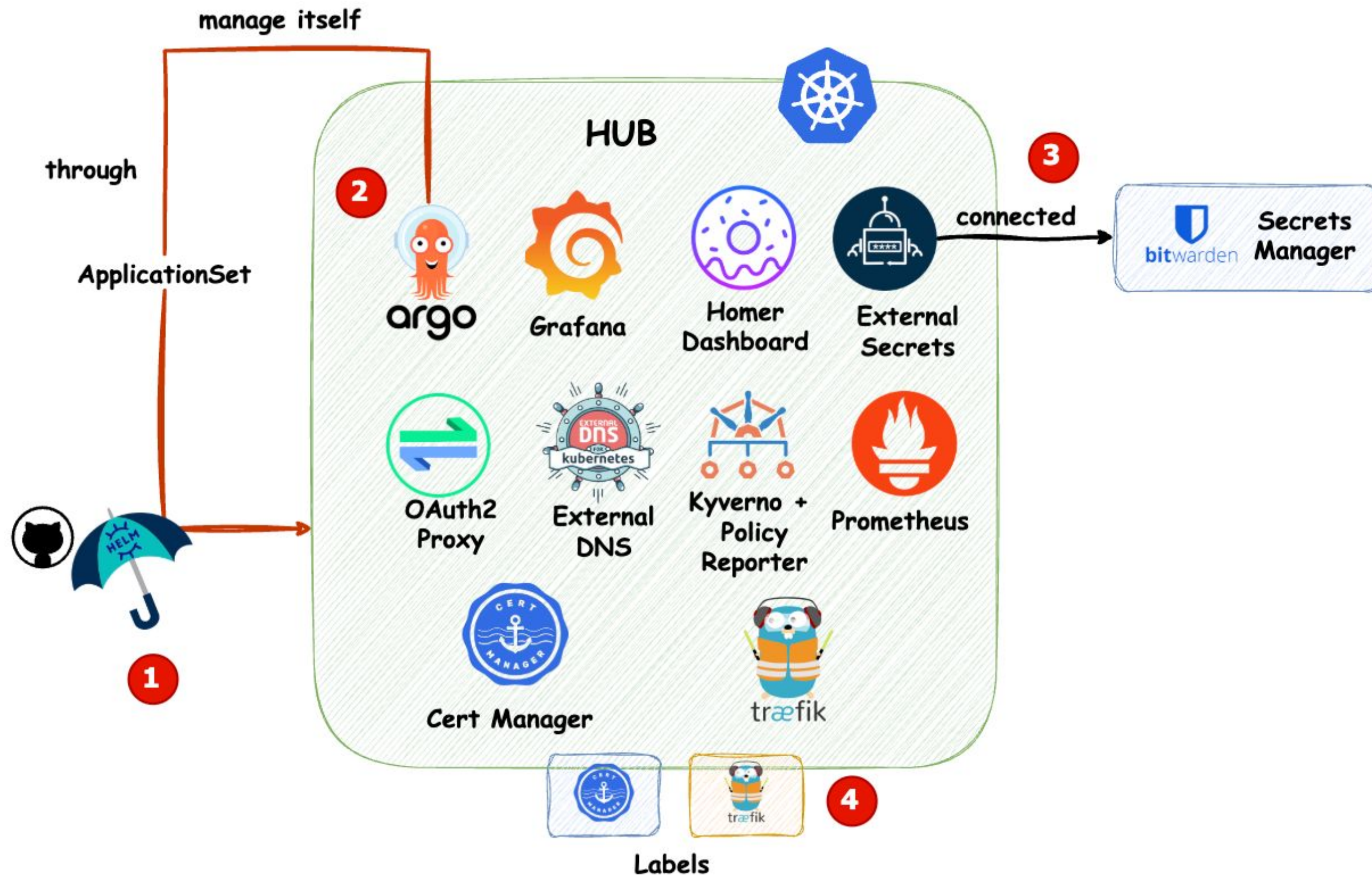
Understand kubara in 3 simple use cases



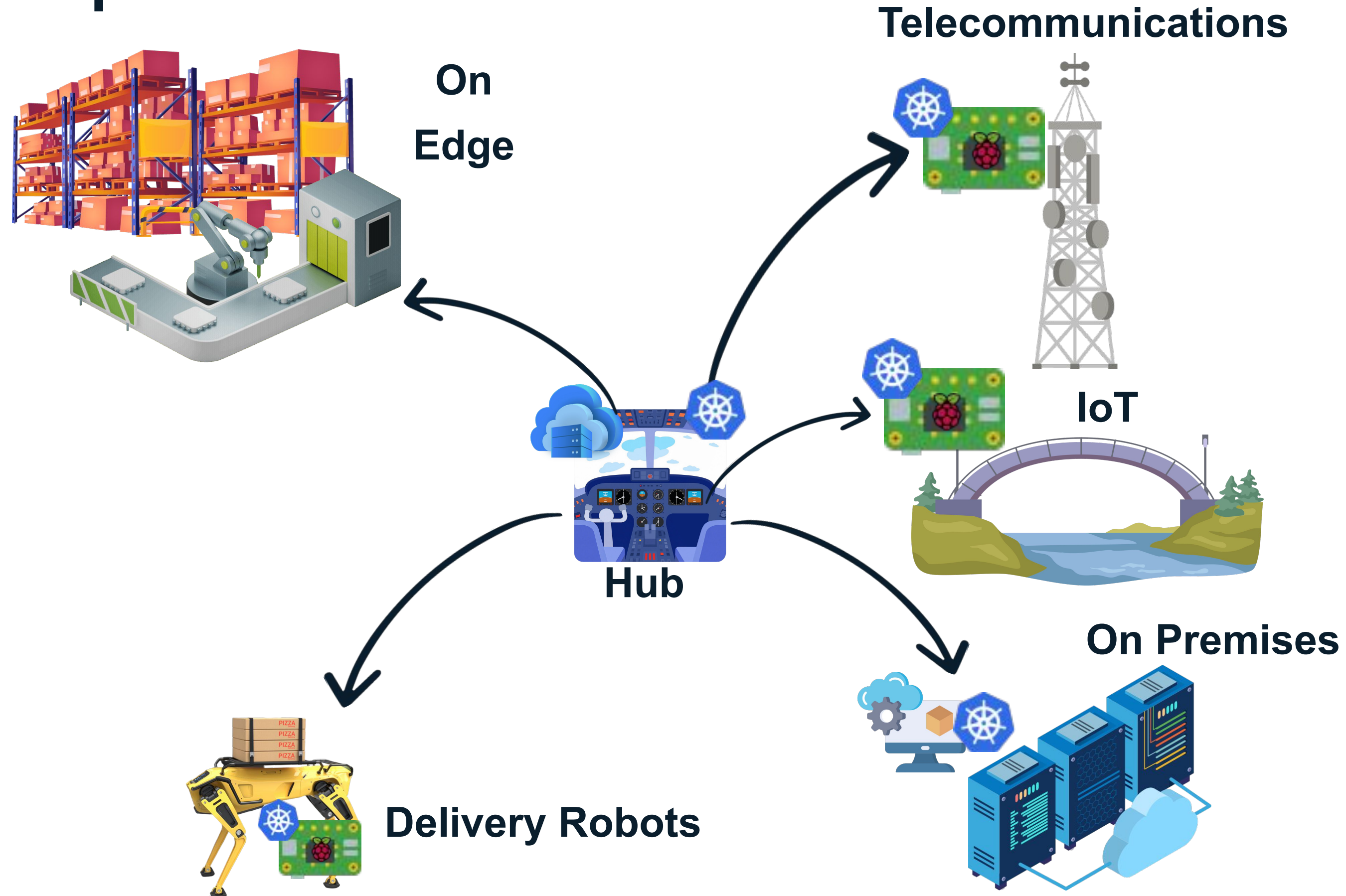
1. Bootstrap the Hub.



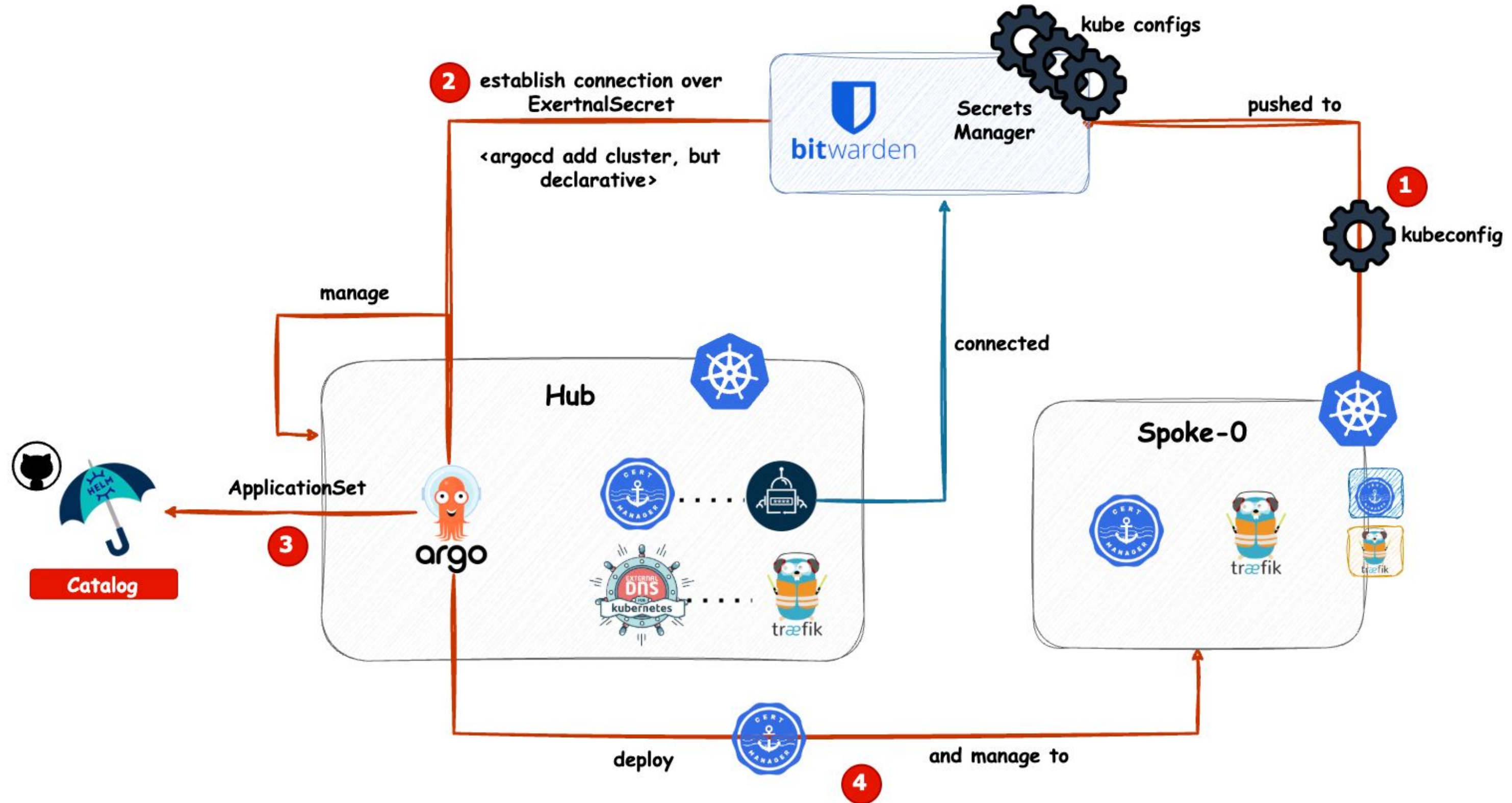
1. Bootstrap the Hub. Central Management.



1. Add Spoke Clusters.



1. Add Spoke Clusters. Distribute Workload.

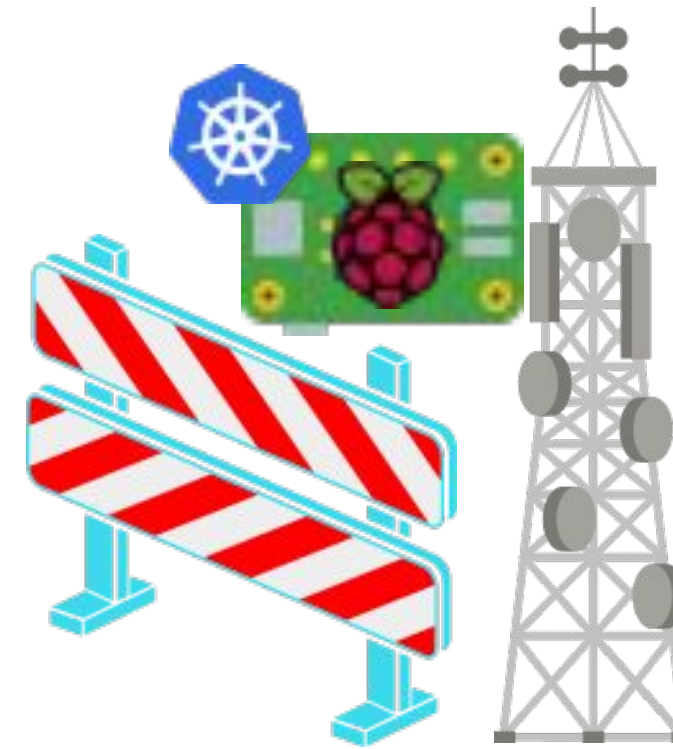


1. Deploy Business Applications.



Hub

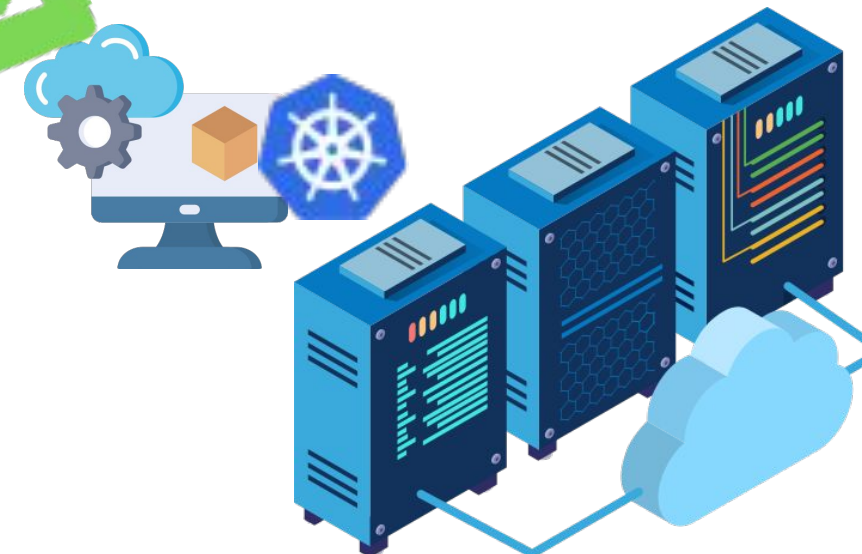
Telecommunications



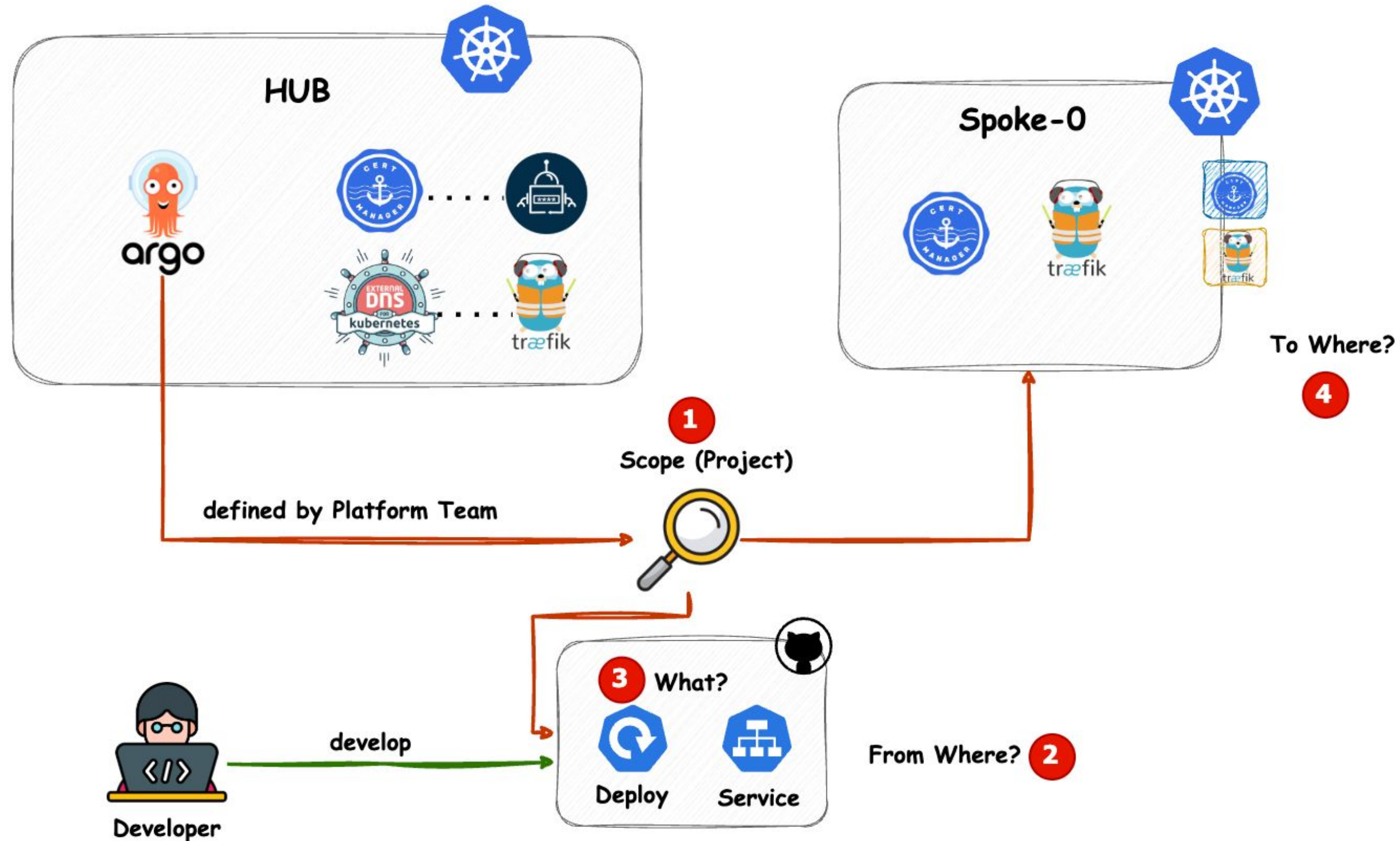
On-Premises



Scope



1. Deploy Business Applications. Multi-Tenancy.



Join us!
Build with us!

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

<https://kubernetes.slack.com/archives/C0B2X0DLPR6>

Star us!

