Configuration as Data - A toolchain intro
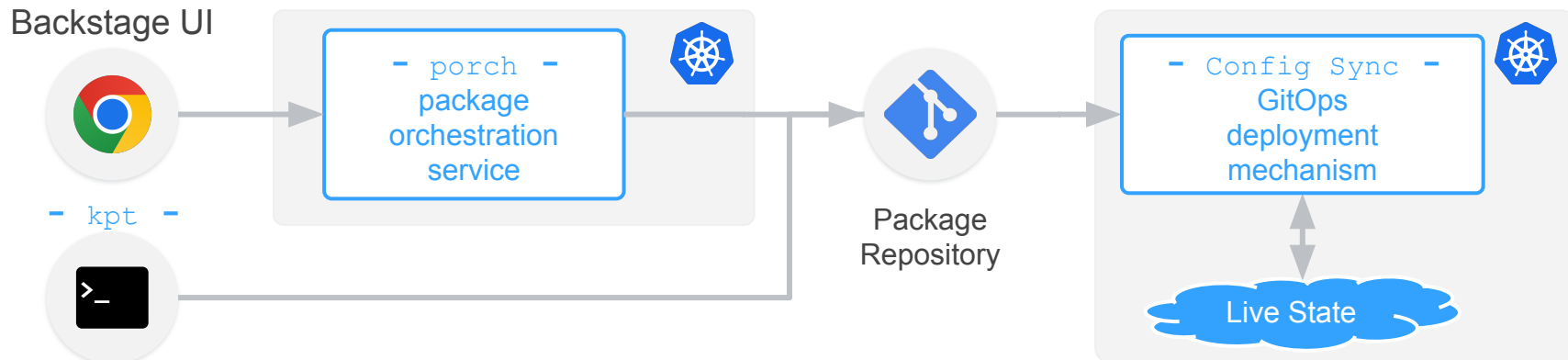
Configuration as Data - toolchain

This Learning with Nephio episode is providing an introduction to the Configuration as Data (CaD) toolchain which consist of the following tools:

- `kpt`, the Command Line Interface (CLI)
- `porch`, aka Package Orchestrator (supports various package operations):
  - Create, clone, upgrade, rollback, delete, read, write packages
  - Execute functions and function pipelines on packages
  - Change control
  - Package and function discovery
- `Config Sync`
  - A tool for continuously managing and deploying configurations across Kubernetes clusters as defined by a source of truth (Git repository, OCI image or Helm chart)

Sample code repo - intro

The sample code repo contains directories which contain examples of how to use CaD tooling with progressive sophistication:

- **Day 1** - crawl
  - Provisions a simple app using `Config Sync` and Kubernetes Manifests
- **Day 2** - walk
  - Provisions a simple app using `Config Sync` and `kpt`
- **Day 3** - run
  - Provisions a simple app using `Config Sync`, `kpt` and `porch`
- **Day 4** - fly
  - Provisions a set of Kubernetes clusters and a simple app using `Config Sync`, `kpt` and `porch`
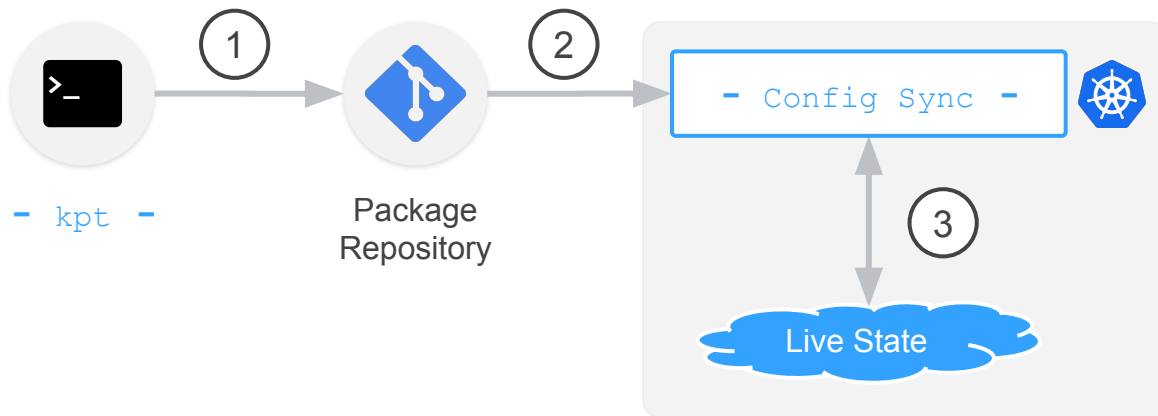
Day 1 - crawl

- Provision a simple app using `Config Sync` and Kubernetes Manifests
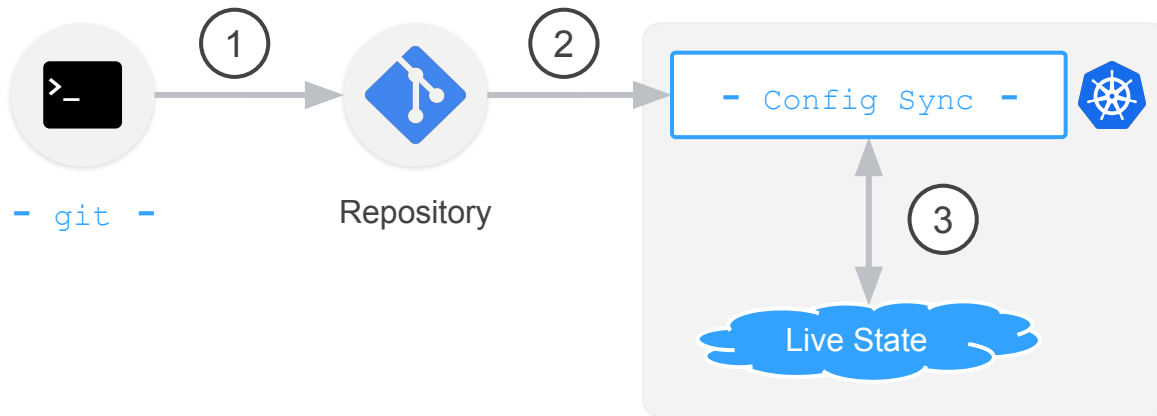
1. `git` commits Kubernetes `yaml` manifests to repository
2. `Config Sync` continuously syncs/pulls manifests into Kubernetes cluster
3. Operators reconcile live state

Day 2 - walk

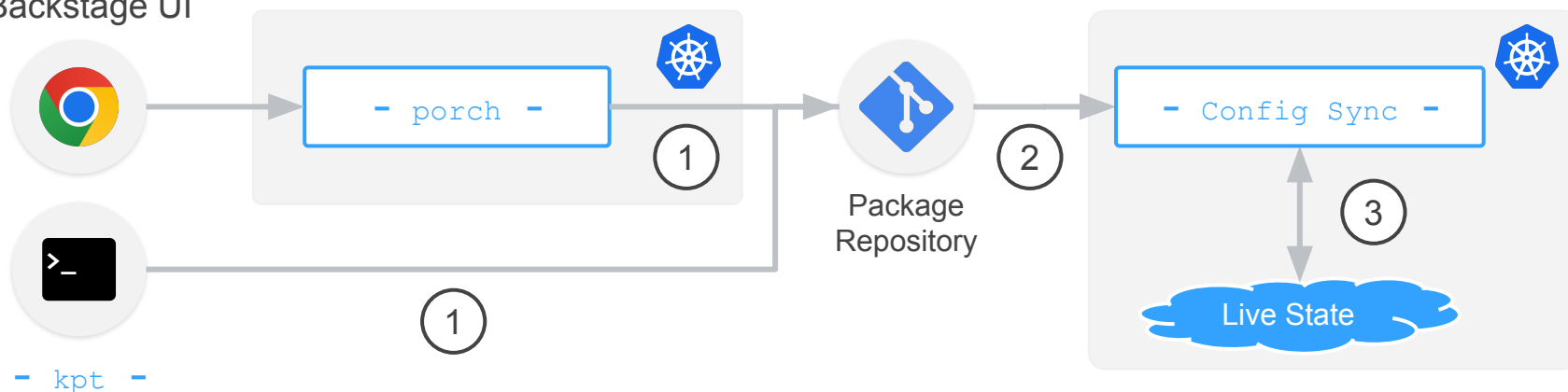- Provision a simple app using `Config Sync` and `kpt`

1. `kpt` commits packages to repository
2. `Config Sync` continuously syncs/pulls manifests into Kubernetes cluster
3. Operators reconcile live state

Day 3 - run

- Provision a simple app using `Config Sync`, `kpt` and `porch`

Backstage UI

1. `kpt` or `porch` commits packages to repository
2. `Config Sync` continuously syncs/pulls manifests into Kubernetes cluster
3. Operators reconcile live state

Day 4 - fly

- Provision a set of Kubernetes clusters and a simple app using `Config Sync`, `kpt` and `porch`

1. `kpt` or `porch` commits packages to repo
2. `Config Sync` syncs manifests into cluster
3. Reconcile live state
4. Operator provisions cluster
5. `kpt` or `porch` commits packages to repo
6. `Config Sync` syncs manifests into cluster
7. Reconcile live state -> provision app