KPT packages & KRM functions

What are kpt Packages?

kpt Packages are directories of resource configuration files. These directories include Kubernetes resource manifests (YAML files) along with additional metadata and templates.

A kpt package typically contains a Kptfile, which provides metadata and configuration options for the package, as well as subdirectories and files that define the Kubernetes resources and their structure.

These packages are designed to be reusable and composable, making it easier to manage and distribute Kubernetes configurations.

Key Components of kpt Packages

1. **Kpt File:**

   The Kptfile is a YAML file that provides metadata about the package, including its name, version, upstream source, and any configuration options.

   It also defines functions (kpt functions) that can be used to modify or validate the resources in the package.

2. **Resource Manifests:**

   These are the YAML files that define Kubernetes resources like Deployments, Services, ConfigMaps, etc.

   The resources are typically organized in a structured directory layout within the package.

3. **Functions:**

   kpt functions are containerized programs that can transform, validate, or generate configuration data.

   Functions can be declarative (defined in the Kptfile or as function configs) and can be run using the kpt fn command.

Create a KPT package

Doc link: https://kpt.dev/book/03-packages/06-creating-a-package

KRM/KPT functions

kpt functions are containerized programs that can transform, validate, or generate configuration data.

They are executed using the kpt fn run command and can be chained together to form a pipeline.

Let’s create one:
https://kpt.dev/book/05-developing-functions/02-developing-in-Go

Kpt pipeline

A sequence of functions that are applied one after another to the configuration files in a package.

The order of functions in the pipeline matters, as each function operates on the output of the previous function.

Thankyou