KUBERNETES ARCHITECTURE

Kubernetes Architecture

Operators

An important concept for orchestration is the use of operators, otherwise known as controllers or watch-loops. Various operators ship with Kubernetes, and you can create your own, as well. A simplified view of an operator is an agent, or *Informer*, and a downstream store. Using a DeltaFIFO queue, the source and downstream are compared. A loop process receives an **`obj`** or object, which is an array of deltas from the FIFO queue. As long as the delta is not of the type **`Deleted`**, the logic of the operator is used to create or modify some object until it matches the specification.

The *Informer* which uses the API server as a source requests the state of an object via an API call. The data is cached to minimize API server transactions. A similar agent is the *SharedInformer*; objects are often used by multiple other objects. It creates a shared cache of the state for multiple requests.

A *Workqueue* uses a key to hand out tasks to various workers. The standard Go work queues of rate limiting, delayed, and time queue are typically used.

The **`endpoints`**, **`namespace`**, and **`serviceaccounts`** operators each manage the eponymous resources for Pods. Deployments manage replicaSets, which manage Pods running the same podSpec, or replicas.