developer.ibm.com/**code**

Kubernetes Service Catalog

Morgan Bauer - mbauer@us.ibm.com

December, 2017

Today's Agenda

- Open Service Broker API (OSBAPI)
- Kubernetes
- Service-Catalog
- Demo

Open Service Broker API (OSBAPI)

- Originated with Cloud Foundry as Services API
- API with five resources
  - List Catalog
  - Provision Instance
  - Bind Instance
  - Unbind Instance
  - Deprovision Instance

- Spec describes how a Platform will interact with a broker
- Several Platforms, with 2 main ones
  - CloudFoundry (CF)
  - Kubernetes

https://www.openservicebrokerapi.org/

- Many Companies working together
  - Pivotal/CloudFoundry Foundation
  - IBM
  - Fujitsu
  - Red Hat
  - SAP
  - Deis now Microsoft
  - Google
- Work done to remove many CF specific concepts
  - Example: Organization and Space become part of a new Context-Profile
- v2.13 released, continuing a line of versions starting at CF

Why OSBAPI?

- CF style: separate stateful persistent storage from stateless apps
- Stateful persistent storage provided by services
- Services managed and provided by experts
  - backed up
  - secure
  - run with best practices
  - Let the experts do what they know best how to do
- How do I get a service?
  - Service Brokers!

Kubernetes

- Container Orchestrator
- Aggregated APIServers for Extension of Kubernetes API
- kubectl as CLI for all operations
- Has `Service` as a reference point
  - no explicit backing
  - need pods behind it to provide an API
- Still could use a way to connect stateless app to stateful service

Service Catalog

- Manages service brokers using OSBAPI
- Service Catalog provides the Kubernetes Platform of OSBAPI
- Native Kubernetes objects as interaction
  - YAML/JSON over HTTP with Group/Kind/Version as Type
  - Generated go Client
  - kubectl is instantly compatible with no source code changes to kubectl
  - APIServer for persistence
  - Controllers for desired-state reconciliation
- When installed, appears as native as any core part of K8s

- Companies working together
  - Red Hat
  - IBM
  - Deis now Microsoft
  - Google
- Extensive reuse of Kubernetes Core Source Code
  - API Machinery
  - Code Generation for server and Client
- Use of experimental Kuberenetes features
  - Aggregated API
  - Delegated Auth
  - Reuse
  - Kubectl Plugins

Demo