THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
JAPAN

AUTOMOTIVE LINUX SUMMIT

Consuming cloud services with the Kubernetes Service Catalog

Neil Peterson
@nepeters

THE LINUX FOUNDATION

Agenda

- Current trends in application architecture
- Kubernetes Service Catalog
- Service Brokers
- Hello world demo
- In depth look at Kubernetes object
- End to end demo

Technology spanning applications

A single application can be built / deployed across a diverse technical stack (Kubernetes, container instance, function, hosted data store).

Deployment and management challenges

- Multiple deployment routines
- Multiple management tools
- Secrets management
- Instance management

We need a common deployment and management platform

Kubernetes

Kubernetes: a platform for running applications (not just containers)

- Yes, container management is front and center
- Extensible API allows for cloud native integrations
- Operating system / kernel for managing applications
- Manage application lifecycle, availability, resourcing, scale, and security
- So, how can Kubernetes integrate with Azure?

Kubernetes Service Catalog

Provision cloud services (and others) from Kubernetes

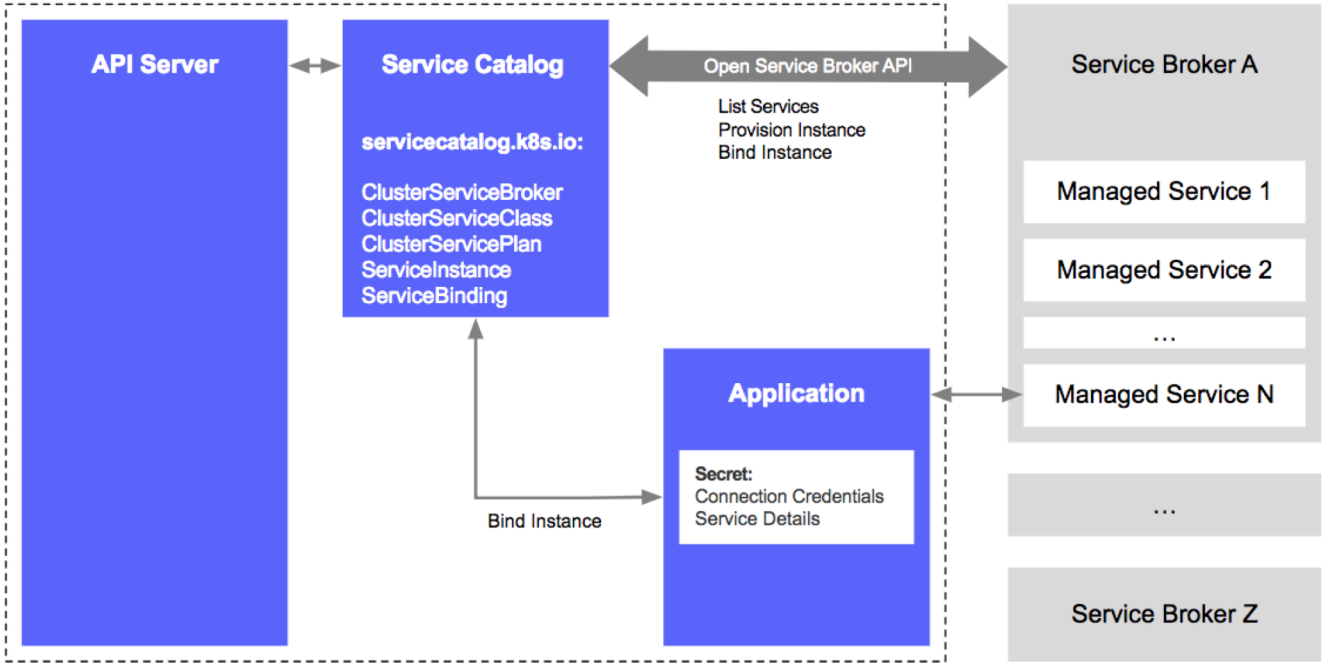
- Extends Kubernetes so that it ”speaks” open service broker
- Aggregated API extension
- Installed with Helm
- Two components API Server and Controller
- Adds five new types to Kubernetes
  - ClusterServiceBroker
  - ClusterServiceClass
  - ClusterServicePlan
  - ServiceInstance
  - ServiceBinding

Open Service Broker

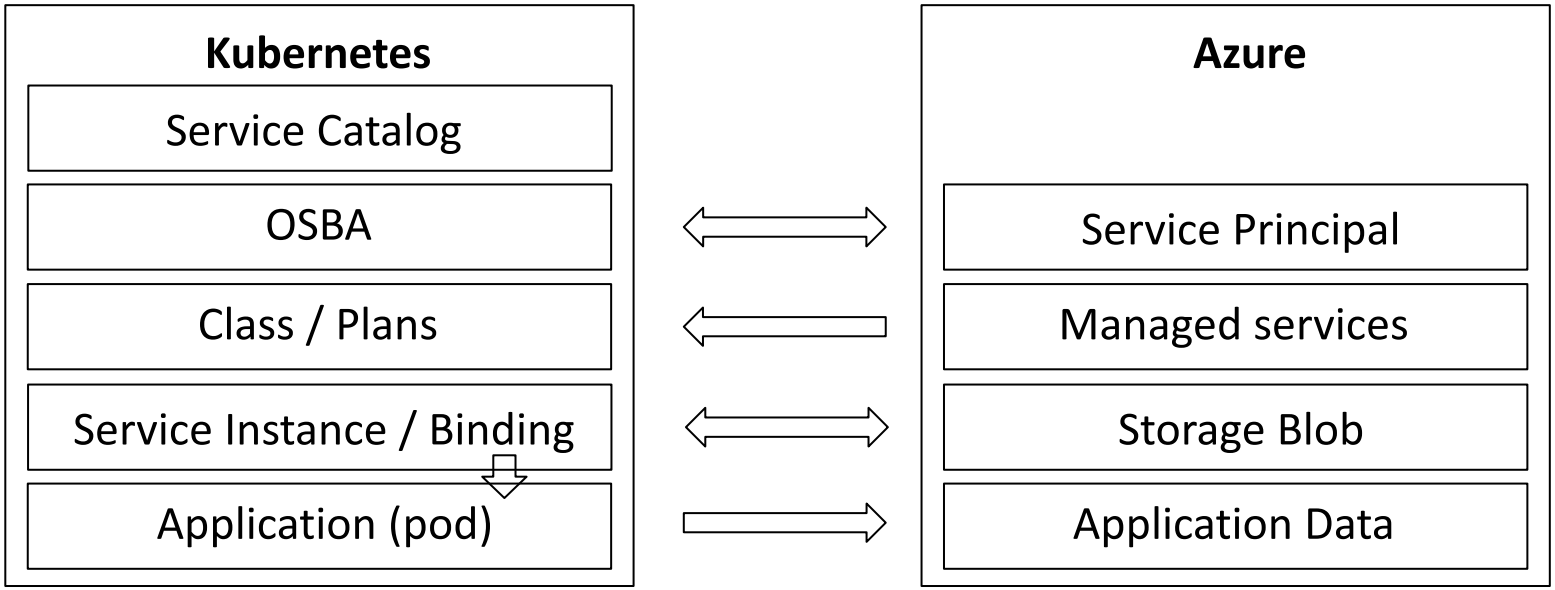
API specification for a standard cloud prodder interface.

- Specifies five operations (provision, bind, unbind, deprovision, update)
- Broker API available for all major cloud operators

API Diagram



Sample workflow (OSBA)



Demo – Hello World

- https://docs.microsoft.com/en-us/azure/aks/integrate-azure
- https://github.com/neilpeterson/open-service-broker-azure-samples

Service Instance

Represents an instance of a managed cloud service.

- Intent to provision cloud service
- Watched by service catalog controller
- Service is created by service broker

```
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceInstance
metadata:
  name: azure-storage-demo
  namespace: default
spec:
  clusterServiceClassExternalName: azure-storage
  clusterServicePlanExternalName: blob-container
  parameters:
    location: eastus
    resourceGroup: osba-azure-storag-demo
```

Service Binding

Request for credentials and / or connection strings for a service instance.

- Intent to use a cloud service
- Watched by service catalog controller
- Secrets are stored in a Kubernetes secret

```yaml
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceBinding
metadata:
  name: azure-storage-binding
  namespace: default
spec:
  instanceRef:
    name: azure-storage-demo
  secretName: osba-azure-storag-demo
```

Application Consumption

How are service bindings used?

- Define environment variables
- Get values from binding secret

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: osba-storage-demo
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: osba-storage-demo
    spec:
      containers:
      - name: osba-storage-demo
        image: neilpeterson/osba-storage-demo
        ports:
        - containerPort: 80
        env:
        - name: AZURE_STORAGE_ACCT
          valueFrom:
            secretKeyRef:
              key: storageAccountName
              name: osba-azure-storag-demo
        - name: AZURE_KEY
```

SVCAT

Command line tool for managing service catalog objects.

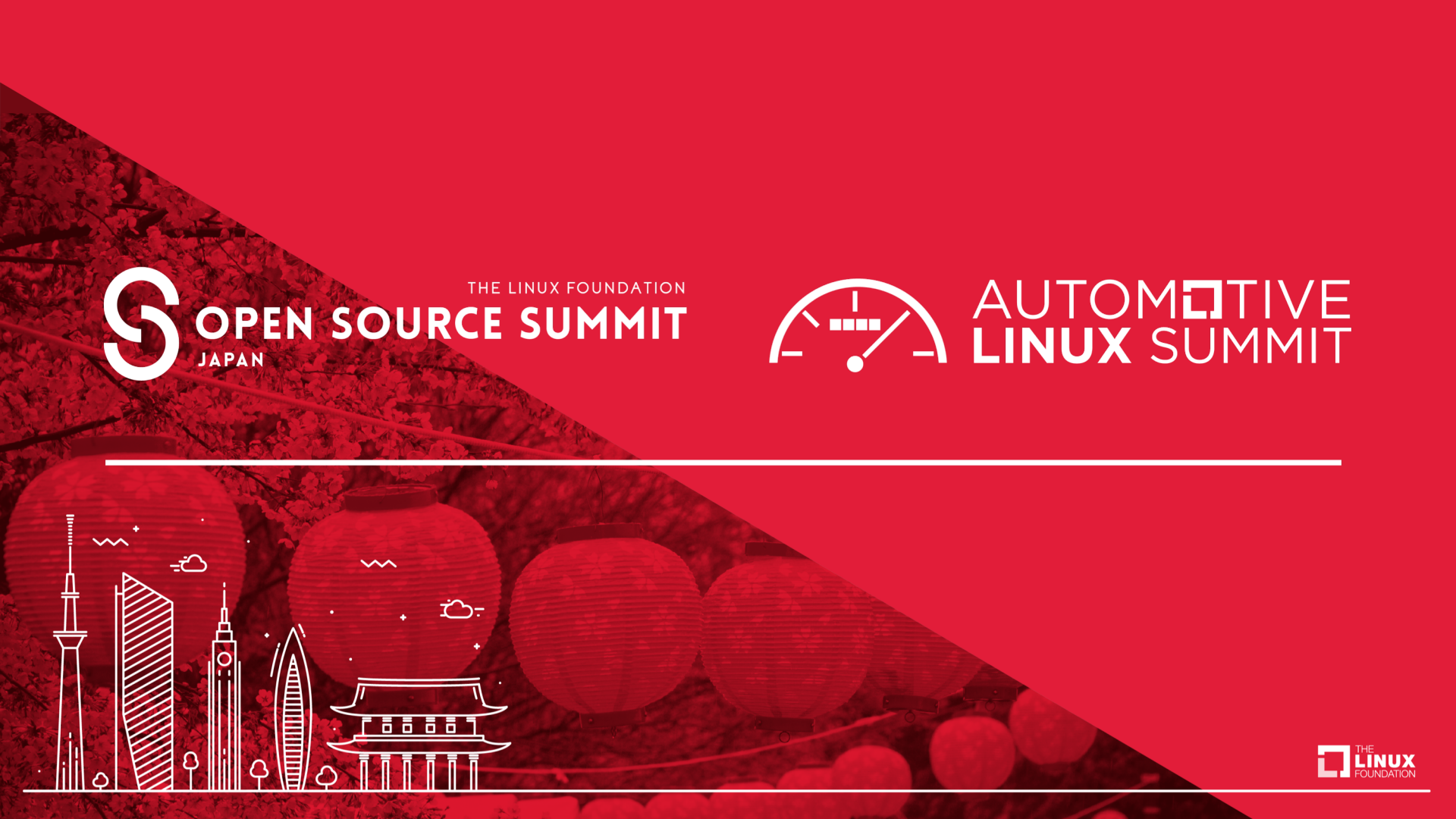
- Not a requirement, simple a convenience tool
- Installed separately (instructions)

Demo – Closer look

Some notes

Some notes based on my experience.

- Service catalog is still considered beta
- Service brokers release on own schedule
- Not all cloud provider service may be present
- Cloud service take time to provision
- Rough edges on some operations


THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
JAPAN
AUTOMOTIVE
LINUX SUMMIT
THE LINUX FOUNDATION