Introduction to the Open Service Broker API

Doug Davis | IBM

dug@us.ibm.com | @duginabox

A Brief History...

- PaaS with a mission to make managing Cloud apps simple

```
$ cf push myapp
$ cf scale myapp -i 5
```

- CF manages deployment, orchestration, routing ...
- Let's developers focus on coding, not infrastructure

Applications are not islands

- Often applications leverage ancillary "Services"
  - E.g. Application stores data in database
- Critical to application's success
  - But developers shouldn't spend their time managing them

Managing Services Can Be A Challenge

- Creating and managing services is non-trivial
  - Duplication of effort across teams, or
  - Ops team manages it for you on **their** schedule
  - Managing credentials could be problematic
    - Sent via email, sticky-notes, etc...
    - Where are they stored? Plain text in config files?

- CF shifts the burden to the Platform via self-service model
  - "Tell us what you need and we'll manage it for you"
  - Service Credentials are protected and provided at runtime

User's Perspective

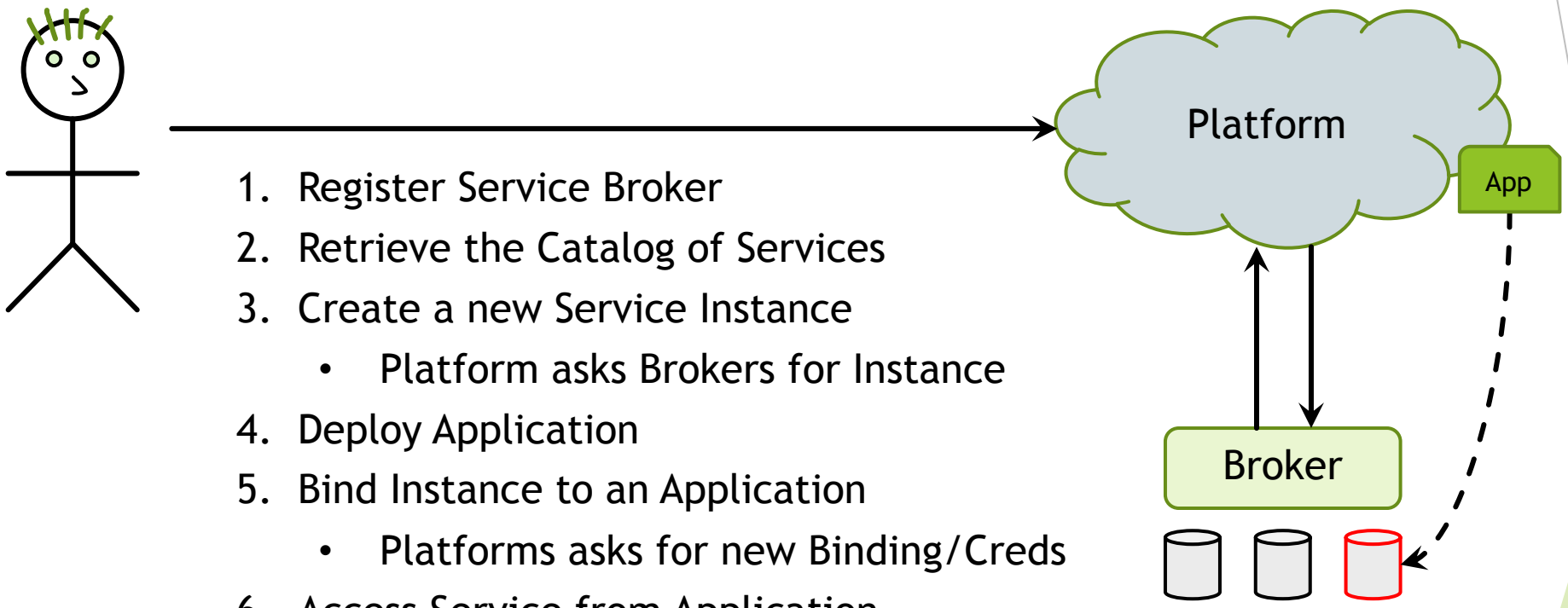
- Easy user experience

```
$ cf create-service mysql free myDB
$ cf bind-service myApp myDB
```

- Credentials are made available to "myApp" via an env var

```
VCAP_SERVICE = {
"mysql":[{
  "credentials":
    "username":"fd7d1b58", "password":"c07750d55",
    "host":"fd7d1b58.db-svc.com", "port":443, ...
  }]
}
```

The Magic

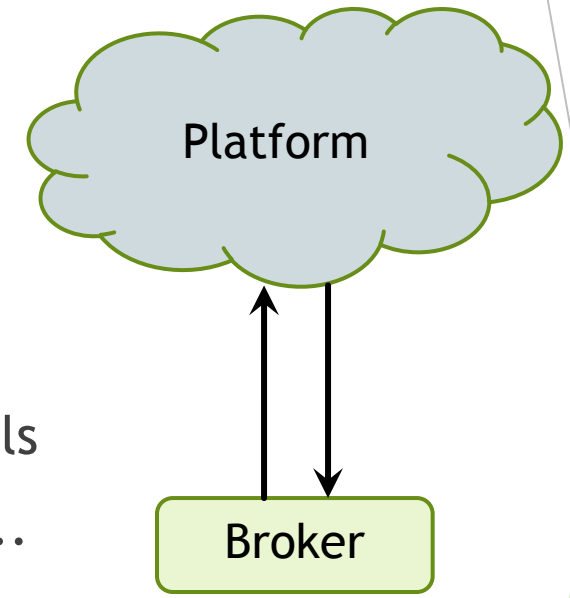


1. Register Service Broker
2. Retrieve the Catalog of Services
3. Create a new Service Instance
   - Platform asks Brokers for Instance
4. Deploy Application
5. Bind Instance to an Application
   - Platforms asks for new Binding/Creds
6. Access Service from Application
   - Using Creds from Binding
   - VCAP_SERVICES env var

Cloud Foundry Service Broker API

- API between the Platform and a Service Broker
- Abstracts the Service Lifecycle APIs

- Service Brokers
  - Manage all aspects of Service's lifecycle
  - User Initiated: Create, Delete, Provide Credentials
  - Automatic: Auto-Scale, Backup, Recovery, QoS, ...
  - Hosted anywhere – in or out of the Platform
    - Application is usually unaware



What is a Service?

- A service can be just about anything

- Data & Analytics – e.g. DBs, ElasticSearch
- Integration – e.g. Box, Twitter, SendGrid
- Utilities – e.g conversions, speech to text
- Infrastructure – networks, volumes, routing
- DevOps – monitoring, metrics, auto-scaling

Why?

- Application Developers / Managers
  - Can focus on their business logic
  - Services managed by the experts
  - Self-service model speeds up CI/CD timelines
- Service Providers
  - Low barrier or entry for new Service Providers
  - Interop: easily integrated into environments that supports the API
  - With ease of access to services, an increase in their usage ($)

Open Service Broker API

- CFF donated SB API to the Open Service Broker API Project

- OSB API Project
  - Evolve API into a community specification
  - To promote **interoperability** across Cloud Platforms (beyond CF)
    - Cloud Foundry, Kubernetes, OpenShift
  - Support of key Cloud leaders:
    - Fujitsu, Google, IBM, Pivotal, RedHat and SAP

OSB API – Looking Forward

- Removing CF specifics in the spec
  - Org, space to be replaced with "context"
  - Define a Context Profile for each Platform
- Parameter Schemas
  - To define the shape of the "parameters"
  - Enables more advanced UI/presentation
- Enable additional Auth mechanisms
  - Beyond Basic Auth

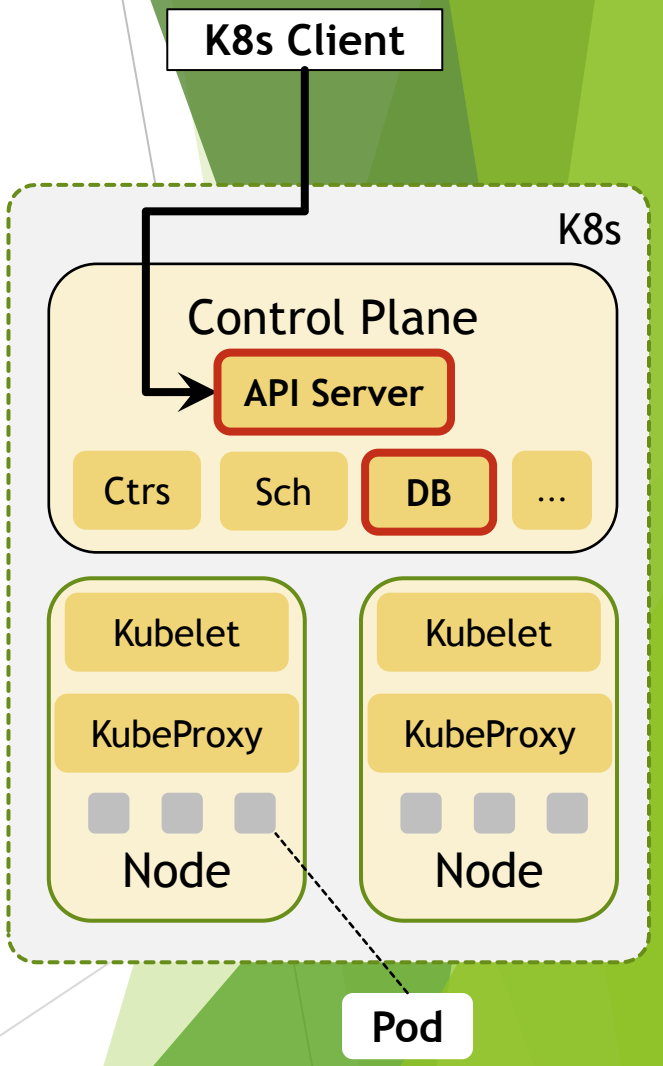
OSB API – Looking Forward – Part Deux

- Define additional Service Lifecycle Actions
  - E.g. Backup/restore for DBs
  - Allow for Service specific extensions
- Allow for a more RESTful model – e.g. GET
- Allow all operations to be asynchronous
- Originating Identity

Kubernetes in a minute!

- **Container Orchestration**
- **A DB with an asynchronous HTTP/REST front-end**
  - **User is exposed to all of the resources in the model**
- **A set of watchers that act as resources change**
  - Controllers react to CRUD operations to manage the resources
  - Schedulers watch to make sure desired state == action state
  - Workers/Kubelet watch for new Pods to be deployed on their Nodes

- Pod: set of containers that must be co-located on same Node
- Labels: filtering/searching mechanism
- KubeProxy to manage intra-cluster communications

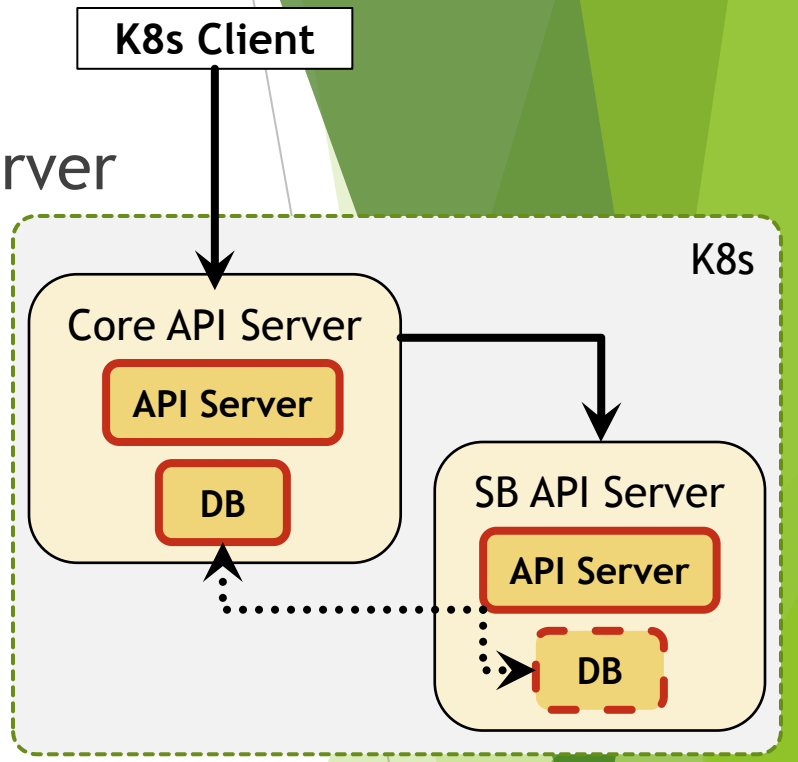


Kubernetes: Extend the Resource Model

- New OSB API specific resources
  - `Broker`
  - `ServiceClass` — "`service`" was already taken / "Plan" is nested
  - `Instance`
  - `Binding` — Credentials stored in `Secrets` in "core"
- New "core" resource
  - `PodPreSet` — Auto-injects new Pods with credentials. Defined as part of the `Binding` resource

Kubernetes: New API Server

- Since not part of "core" we needed a separate API Server
  - Basically a mini-Kubernetes
  - Accessed via API aggregation
  - Can use the "core" API Server's DB or its own
- Our work has been on the bleeding-edge
  - API Server: first to actually create a new one
  - API Aggregation: to share a single endpoint across API Servers
  - PodPreset



Kubernetes: User Experience

- Can work with resources directly

```
$ kubectl create -f broker.yaml        # Admin action
$ kubectl create -f instance.yaml      # Dev action
$ kubectl create -f binding.yaml       # Dev action
```

- Or, `kubectl` plugin for better UX

```
$ kubectl create-service-instance SERVICE_CLASS_NAME \
      PLAN_NAME NAMESPACE INSTANCE_NAME
$ kubectl bind-service-instance INSTANCE_NAME \
      BINDING_NAME NAMESPACE
```

Kubernetes: Support for OSB API - Status

- Incubator project/SIG: Service Catalog
- Will be fully OSB API specification compliant
- Very involved in the OSB API WG
  - Via IBM, RedHat and Google's participation
- Currently in 'alpha' but 'beta' will be soon
  - Beta implies backwards compatible from then on
  - So its relatively safe for enterprises to pick-up and play with

Get Involved

- Web Site: https://www.openservicebrokerapi.org/
- Github: https://github.com/openservicebrokerapi/servicebroker/
- Google Group: https://groups.google.com/forum/#!forum/open-service-broker-api
- Slack: http://slack.openservicebrokerapi.org/
- Weekly Calls (Tuesdays 12:30pm ET): https://github.com/openservicebrokerapi/servicebroker/wiki/Weekly-Call
- Kubernetes: https://github.com/kubernetes-incubator/service-catalog

Questions?

Thank You!