Pivotal Cloud Foundry®

Service Brokers

Adding Managed Services

Pivotal

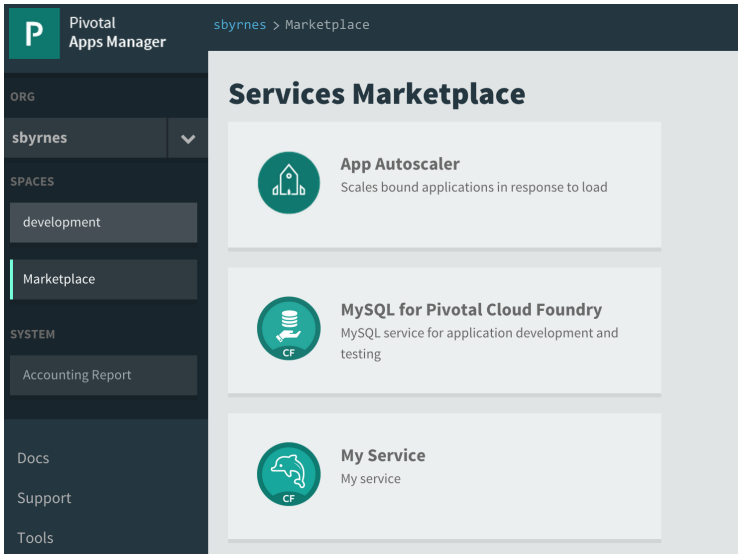
Topics

- **Managed Services Overview**
- Service Broker API
- Service and Service Broker Implementation
- MySQL Service for Pivotal Cloud Foundry

Pivotal

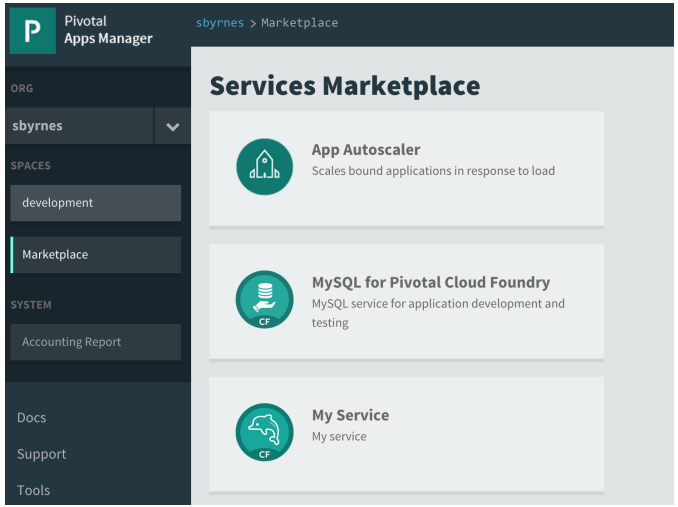
Why Managed Services?

- A managed service is a “first class” citizen on the Cloud Foundry platform
  - Available in the marketplace
  - Has service plans
  - Can provision or reserve resources
  - Can provision credentials
  - Services instances can be created
- For operators:
  - Can add a custom service to the platform and let the developers provision them
- For developers:
  - Can provision and use services immediately and without involving operations
  - Increases agility



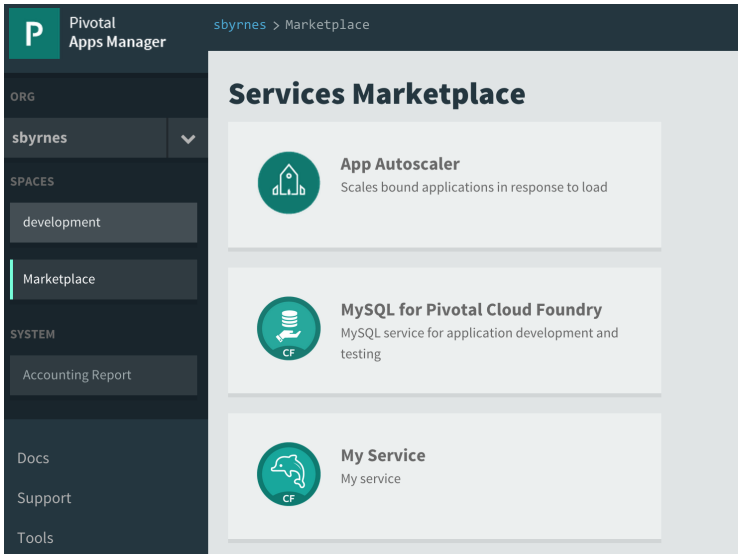
Managed Services

- Available as plans in the marketplace
- The service *may* be a Cloud Foundry application
- The service *may* be a BOSH deployment (for example, MySQL)
- Includes a service broker, which implements a service broker API
- The service broker *may* be a Cloud Foundry application

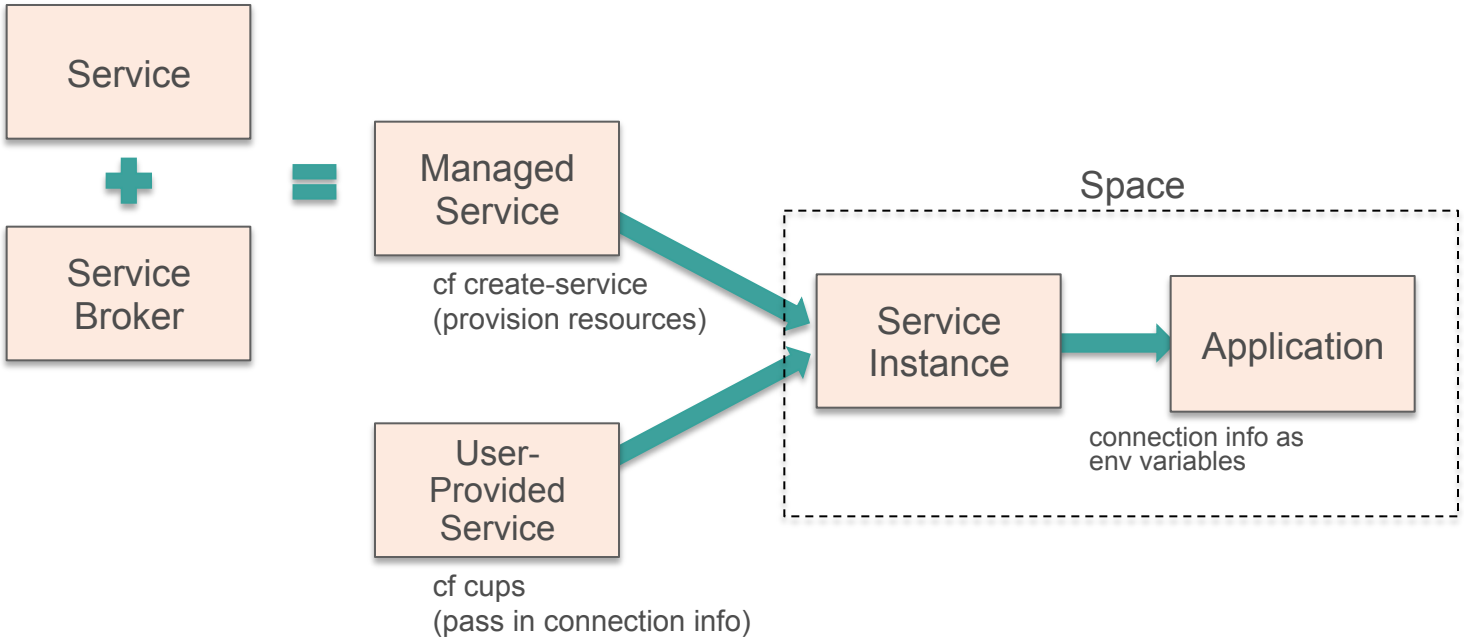


Service Broker Overview

- Service brokers provide a standardized way to add a managed service to the Cloud Foundry platform
  - Service broker API
- Create a service broker to add a custom service to the marketplace



The Services Big Picture

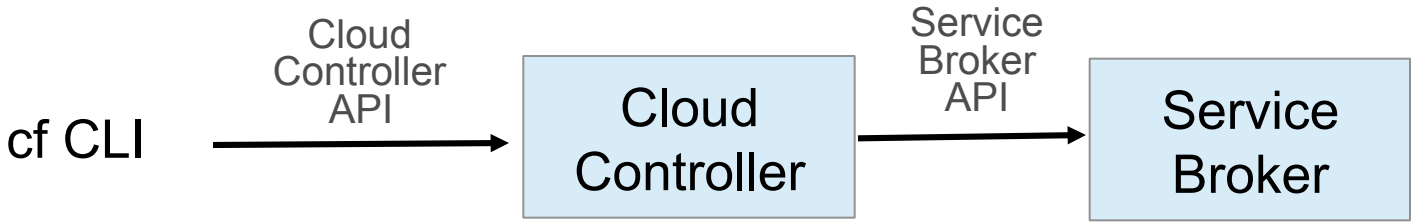


For managed services, connection info is provisioned when binding the service instance to the application

Topics

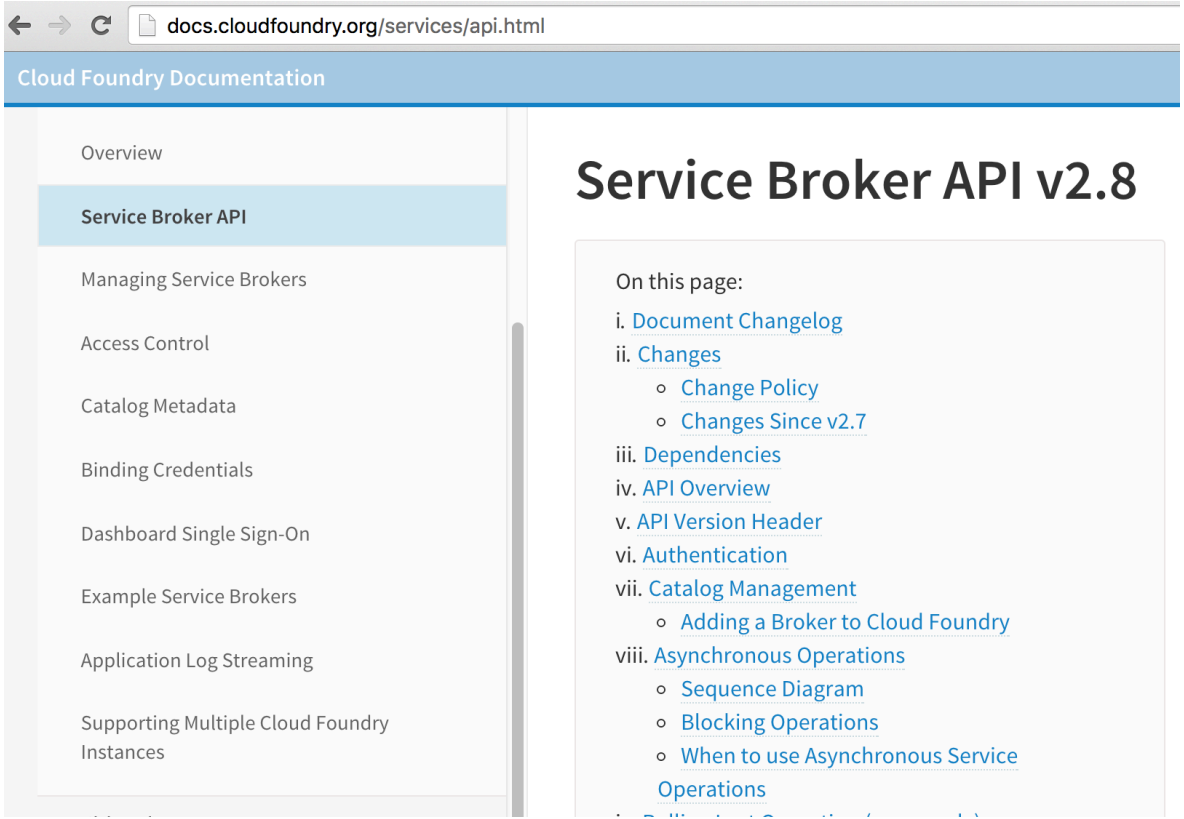
- Managed Services Overview
- **Service Broker API**
- Service and Service Broker Implementation
- MySQL Service for Pivotal Cloud Foundry

The Cloud Controller vs. Service Broker APIs



- Requests from developers or operators related to services start with the cf CLI or other client
- The cf CLI actually uses the cloud controller API
- The cloud controller calls the service broker API

Service Broker API



Service Broker API: /catalog

Request

Route

```
GET /v2/catalog
```

cURL

```
curl -H "X-Broker-API-Version: 2.7" http://username:password@broker-url/v2/catalog
```

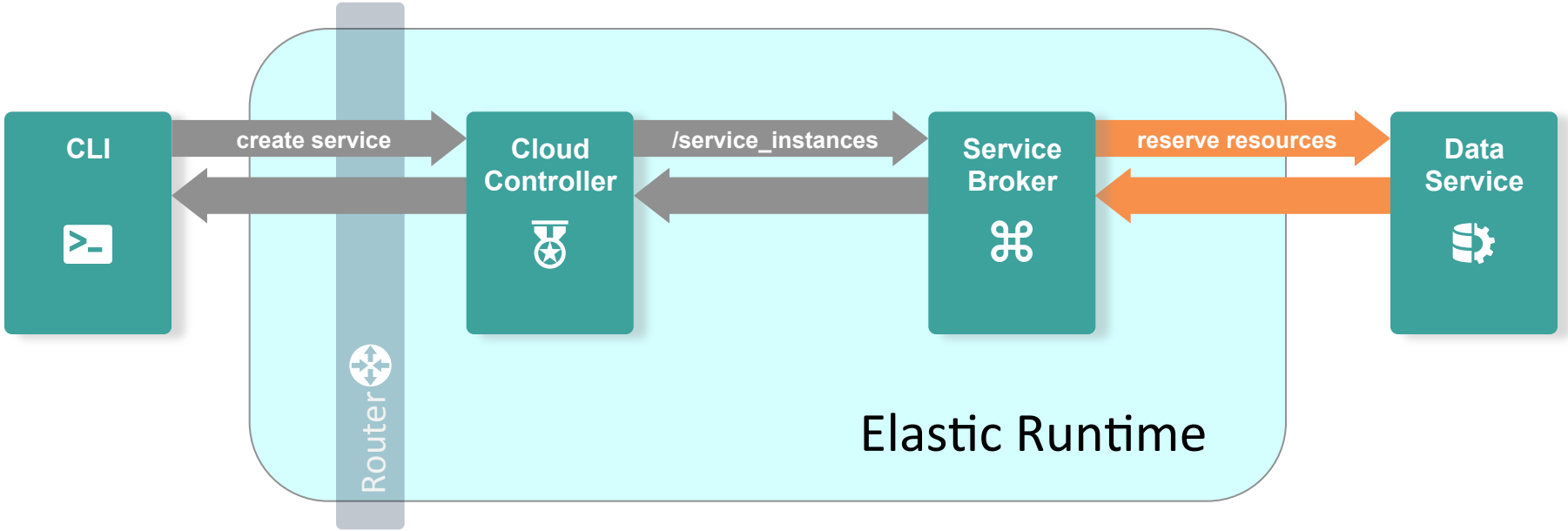
- The /catalog endpoint returns JSON with information like the service's name, description and plans
- The catalog is visible in the marketplace

Service Broker API: /service_instances

- **PUT** /v2/**service_instances**/:instance_id
  - Create (provision) a new service instance
  - For longer operations, use asynchronous provisioning by appending accepts_incomplete=true
  - The cloud controller creates and passes a unique :instance_id to the service broker
  - The :instance_id is used for future calls, such as the DELETE
- **DELETE** /v2/**service_instances**/:instance_id
  - Delete (deprovision) a service instance

Creating a Service Instance

- The **`cf create-service`** command is executed, making the service instance available to the current space
  - Calls the service broker's **`PUT`** `/v2/` **`service_instances`**`/:instance_id` endpoint

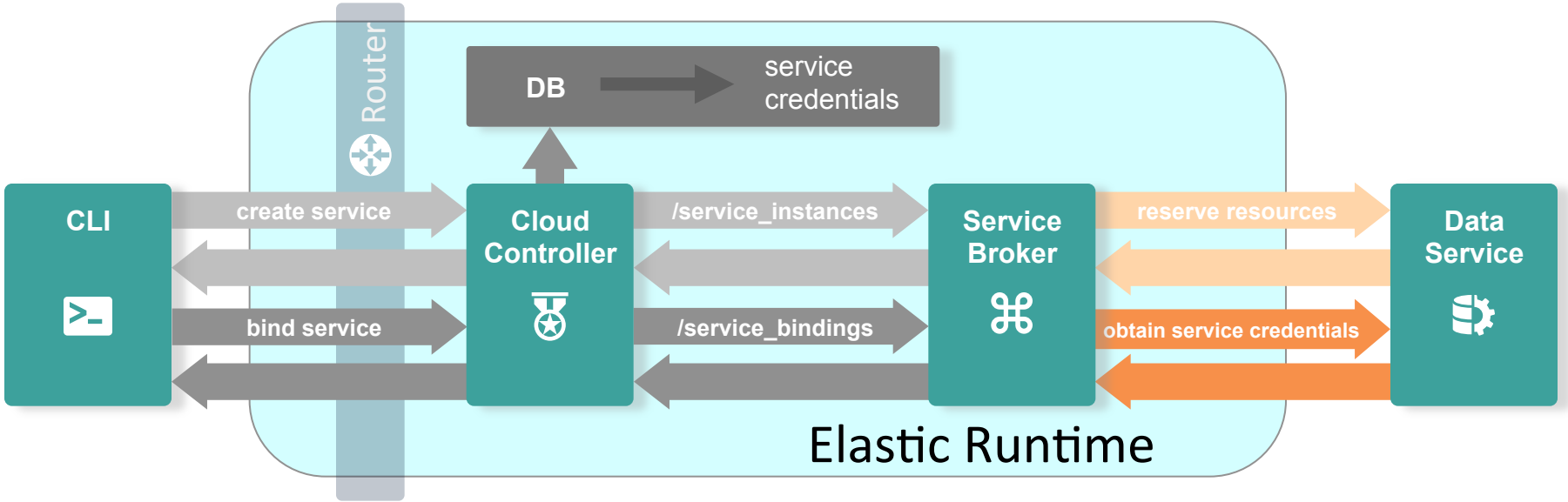


Service Broker API: /service_bindings

- **PUT** /v2/service_instances/:instance_id/**service_bindings**/:id
  - Create a new binding to a service instance
  - Results in connection information passed from the service broker to the cloud controller and its database
    - The service broker *should* provision but may also reuse credentials
  - The :instance_id is from the /service_instances endpoint
  - The :id represents the service binding and is created and passed from the cloud controller
    - The cloud controller associates the service instance, service binding and the application
- **DELETE** /v2/service_instances/:instance_id/**service_bindings**/:id
  - Unbind from a service instance

Binding a Service

- The **`cf bind-service`** command is executed
  - Calls the service broker's endpoint- **`PUT`** `/v2/service_instances/:instance_id/service_bindings/:id`
  - Adds service credentials for the bound application to the cloud controller database
  - Those credentials are passed to application instances via environment variables



Service Broker API- Update Service Plan

- **PATCH** /v2/**service_instances**/:instance_id
  - Update a service instance to a different plan
  - The /catalog endpoint must include **`plan_updateable: true`**

Pivotal

Service Broker API- Summary

- GET /v2/catalog
  - List services and plans available from this broker
- **PUT** /v2/**service_instances**/:instance_id
  - Create (provision) a new service instance
- **DELETE** /v2/**service_instances**/:instance_id
  - Delete (deprovision) a service instance
- **PATCH** /v2/**service_instances**/:instance_id
  - Update a service instance to a different plan
- **PUT** /v2/service_instances/:instance_id/**service_bindings**/:id
  - Create a new binding to a service instance
- **DELETE** /v2/service_instances/:instance_id/**service_bindings**/:id
  - Unbind from a service instance

Pivotal

Topics

- Managed Services Overview
- Service Broker API
- **Service and Service Broker Implementation**
- MySQL Service for Pivotal Cloud Foundry

Service Broker Implementation

- Service broker implementation is up to the service provider/developer
- Cloud Foundry only requires that the service provider implement the service broker API
- Deploy using **`cf create-service-broker`**
  - Requires admin privileges
  - The service is then available in the marketplace

Developing a Custom Service Broker

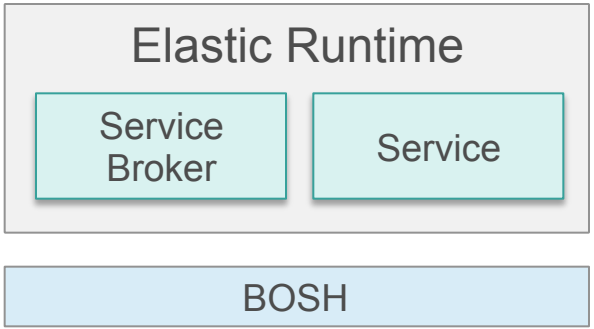
- Several GitHub projects exist
  - Use as starting points for defining a service broker
- Download generic project at
  - https://github.com/cloudfoundry-community/spring-boot-cf-service-broker
- Or rework Mongo DB example project
  - https://github.com/spring-cloud-samples/cloudfoundry-service-broker

Service Instance Provisioning Examples

- For non-data services, provisioning could just mean getting an account on an existing system
- For the MySQL for Pivotal Cloud Foundry service, provisioning results in a database being created on the installed MySQL server
- The Cloud Foundry platform is flexible and can support many MySQL provisioning examples
  - Creating new MySQL server instances running on its own VM or in a container on a VM
  - A database with business schema already there
  - A copy of a full database, for example a QA database that is a copy of the production database

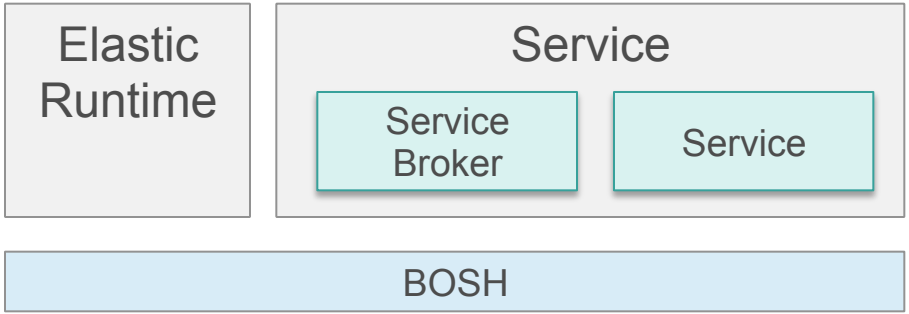
Service and Service Broker Deployment Models (1/3)

- Service and service broker pushed as applications in an Elastic Runtime space
- Service broker endpoints can be added to an existing service



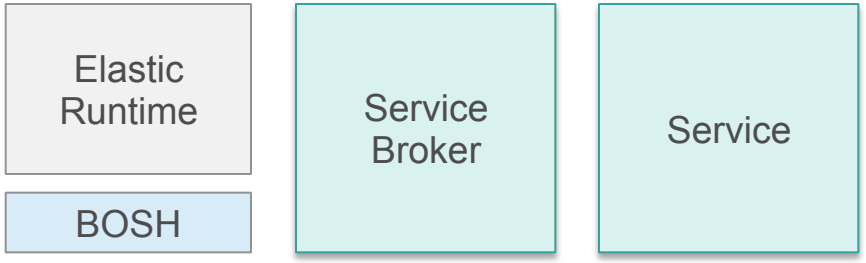
Service and Service Broker Deployment Models (2/3)

- Entire service (and optionally service broker) packaged and deployed by BOSH alongside Cloud Foundry
- This is the MySQL for Pivotal Cloud Foundry model



Service and Service Broker Deployment Models (3/3)

- Entire service, including service broker, deployed and maintained outside of BOSH and the Elastic Runtime



Any combination of the deployment models will also work

Pivotal

Topics

- Managed Services Overview
- Service Broker API
- Service and Service Broker Implementation
- **MySQL Service for Pivotal Cloud Foundry**

Example: MySQL Service

- Pivotal Web Services
  - The MySQL Server and service broker have already been installed
  - MySQL services are available in the marketplace
- For custom installations
  - Use Ops Manager to install the MySQL for Pivotal Cloud Foundry tile
    - The MySQL Server and service broker are installed using BOSH, based on settings in Ops Manager
  - MySQL services are then available in the marketplace

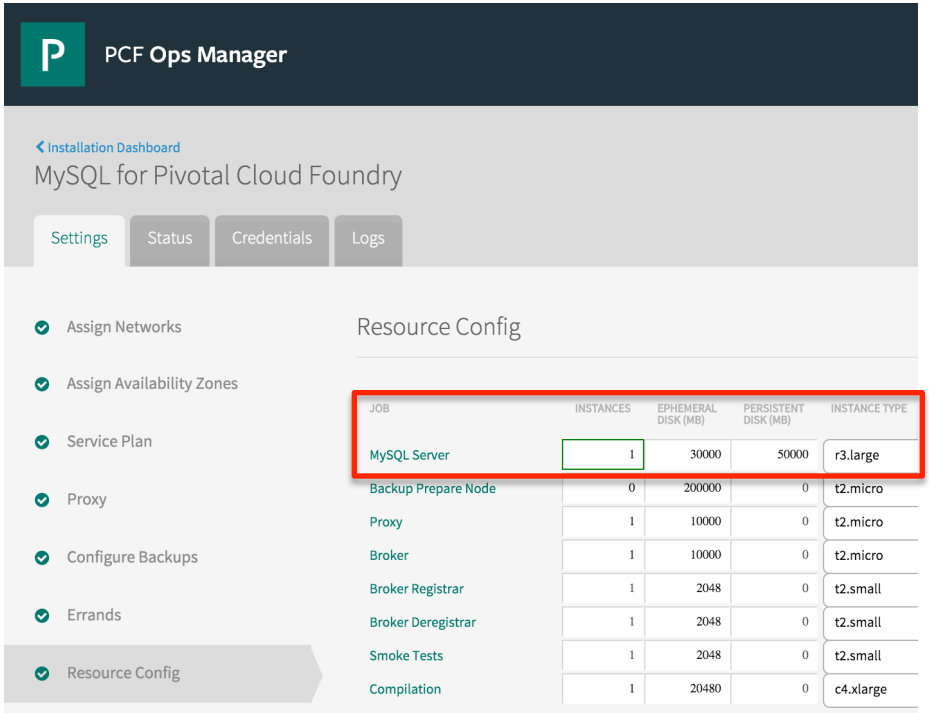
MySQL for Pivotal Cloud Foundry Service Provisioning and Binding

- MySQL database provisioning (**`cf create-service`**)
  - The service is pre-provisioned by BOSH when the MySQL tile is installed or updated
  - When creating a service instance, a “create database” statement issued on the installed MySQL Server
- MySQL binding (**`cf bind-service`**)
  - MySQL user and password are created for the application
  - Credentials are passed to the application instances as environment variables
  - The user is granted access to the schema

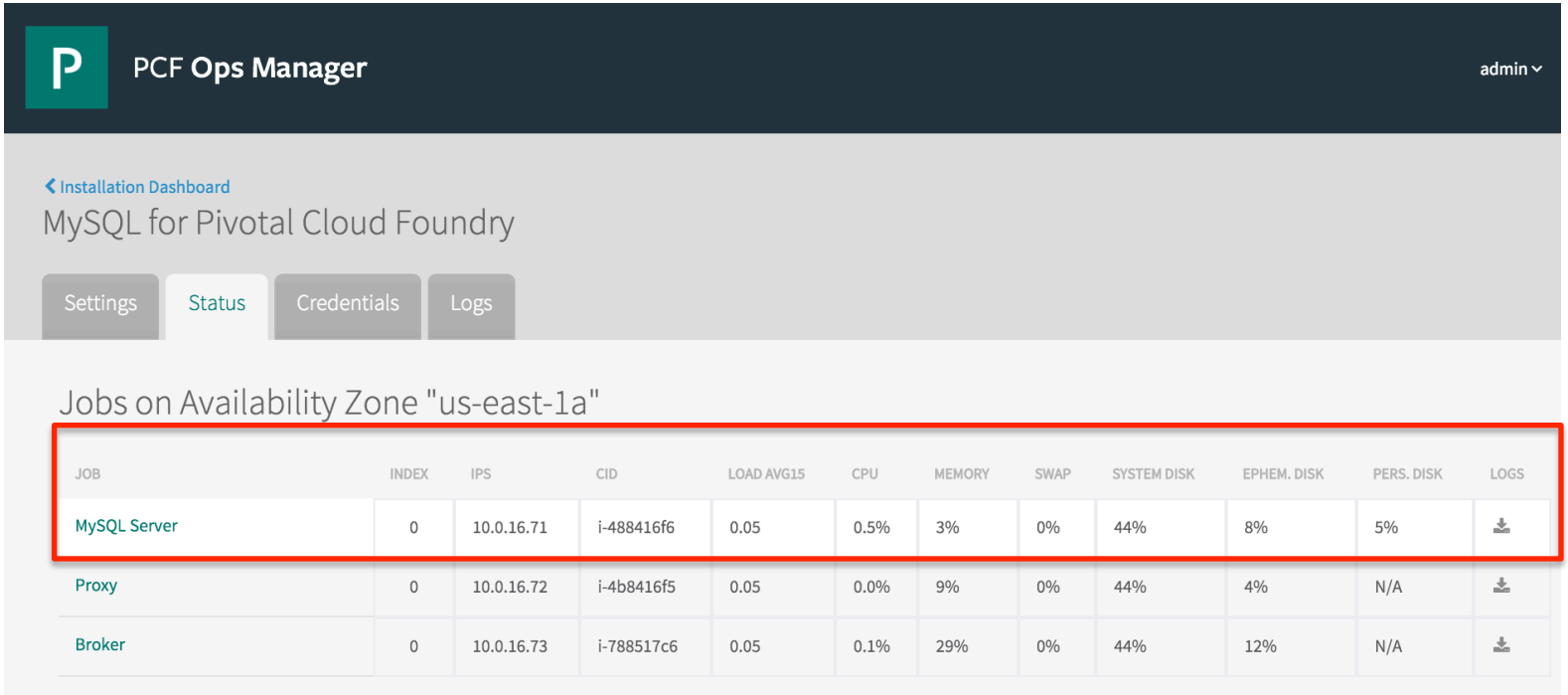
Pivotal

MySQL Server Provisioning

- Set the MySQL Server persistent disk size

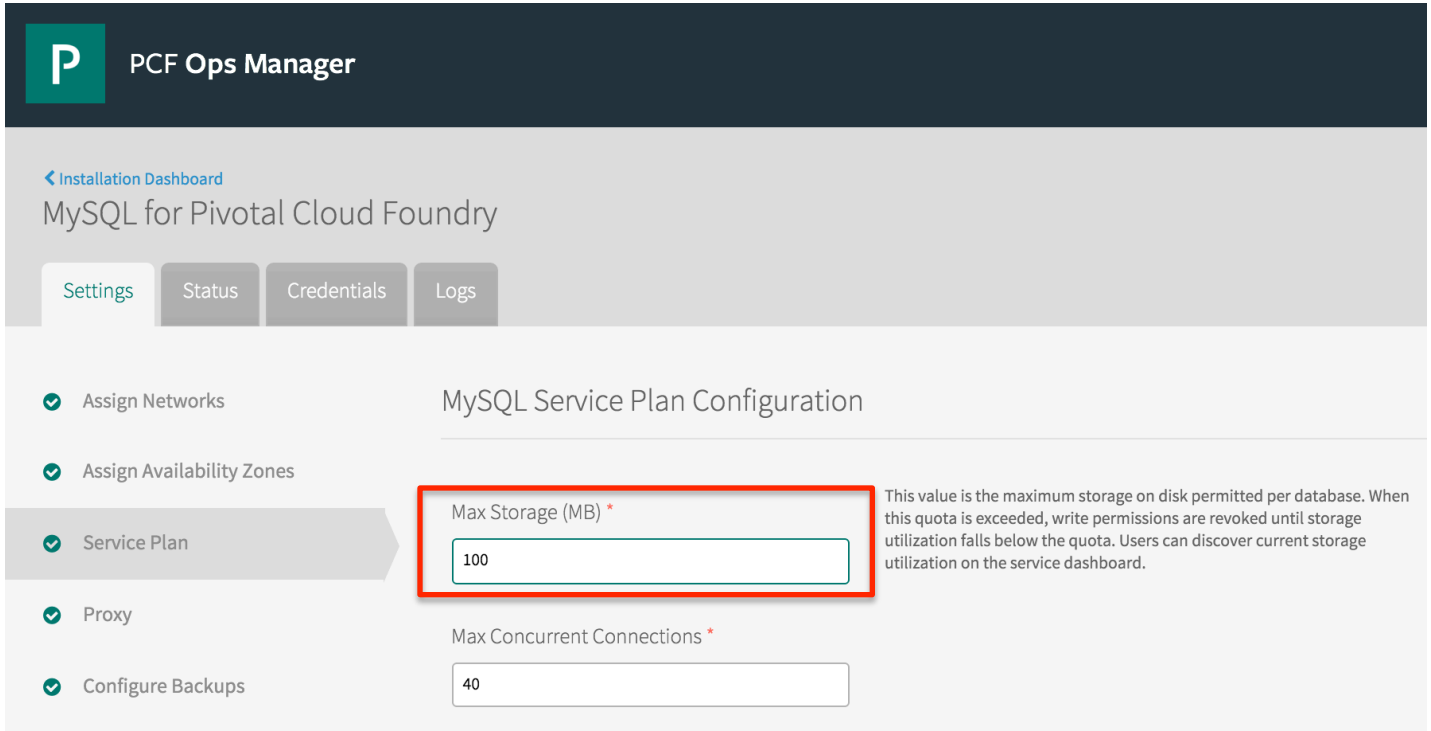


View the Current MySQL Server Disk Usage



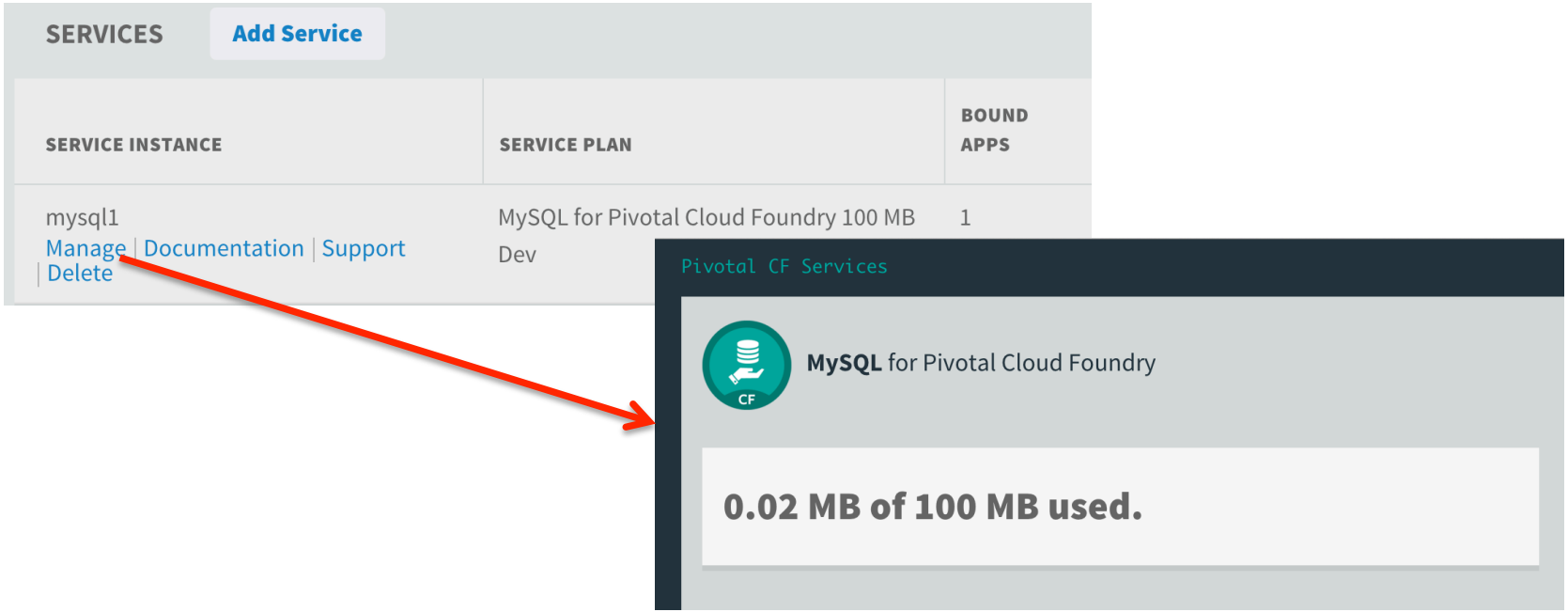
Service Instance Plan Size

- Each service instance reserves storage on the MySQL Server

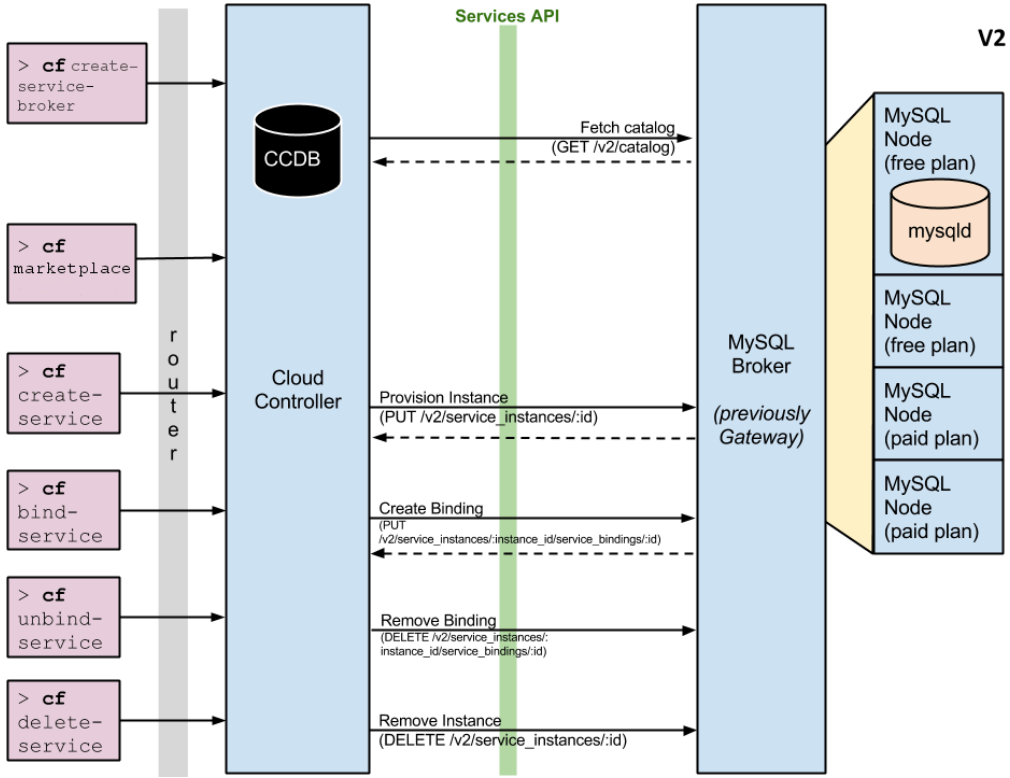


Viewing Current Service Instance Usage

- Navigate to the service instance in a space and click the Manage link



MySQL Service Summary



Pivotal

Lab

Create, register and use a new service broker