Pivotal Cloud Foundry®

Cloud Foundry Services

Overview

Pivotal

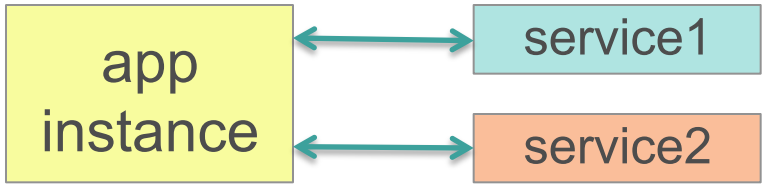
Topics

- **Introduction to services**
- Create and bind managed services
- Use a managed service in an application
- Create user-provided services

Pivotal

Cloud Foundry Services

- A service is an **external** application **dependency** or **component** such as:
  - External state
    - Database, cache, message queue
  - Tools
    - Monitoring app, external logging sink, OAuth security
  - External applications
    - Microservice, web service, legacy application

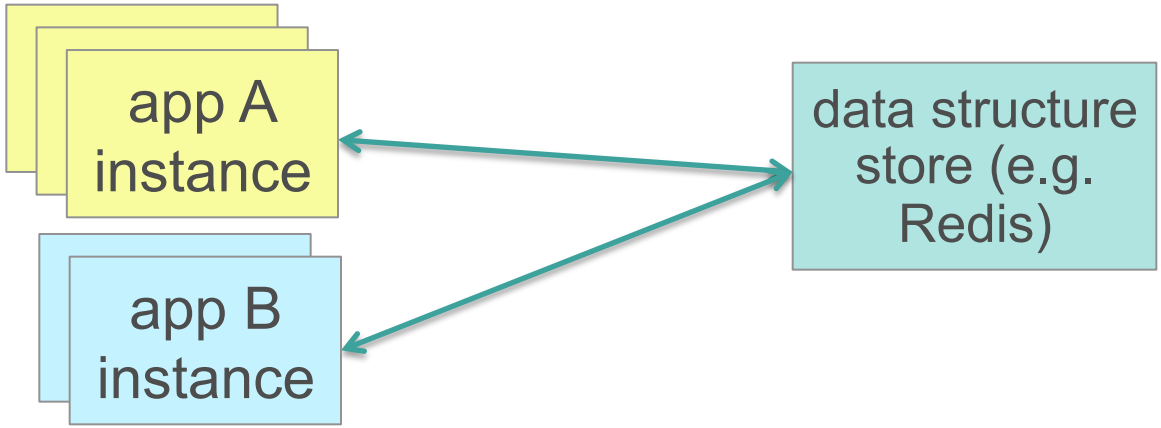


services are external to the app instance

Pivotal

Application State

- App instances are ephemeral (temporary)- state is not stored there
- Store state in services
  - May be *shared* among many applications and app instances



Two Types of Services

- Managed services (*"marketplace"* services)
  - Available from the marketplace 'catalog'
  - Instances are often **provisioned and managed** by the platform, for use by applications
  - The platform supplies connection information to applications
  - Can be custom created
- User provided services
  - The service is not in the marketplace
  - The platform only supplies connection information to applications

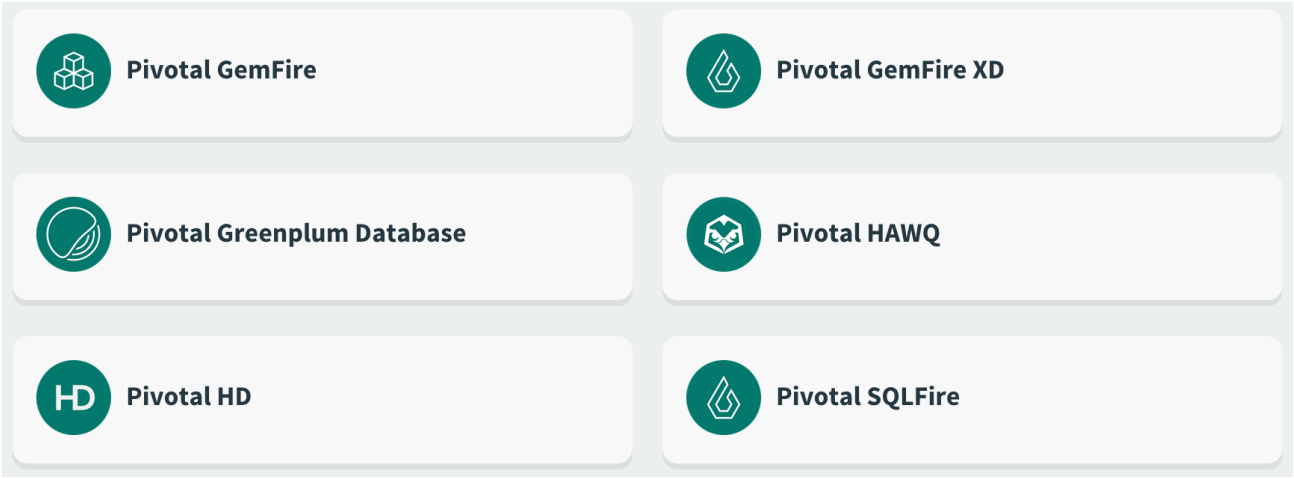
Topics

- Introduction to services
- **Create and bind managed services**
- Use a managed service in an application
- Create user-provided services

Pivotal

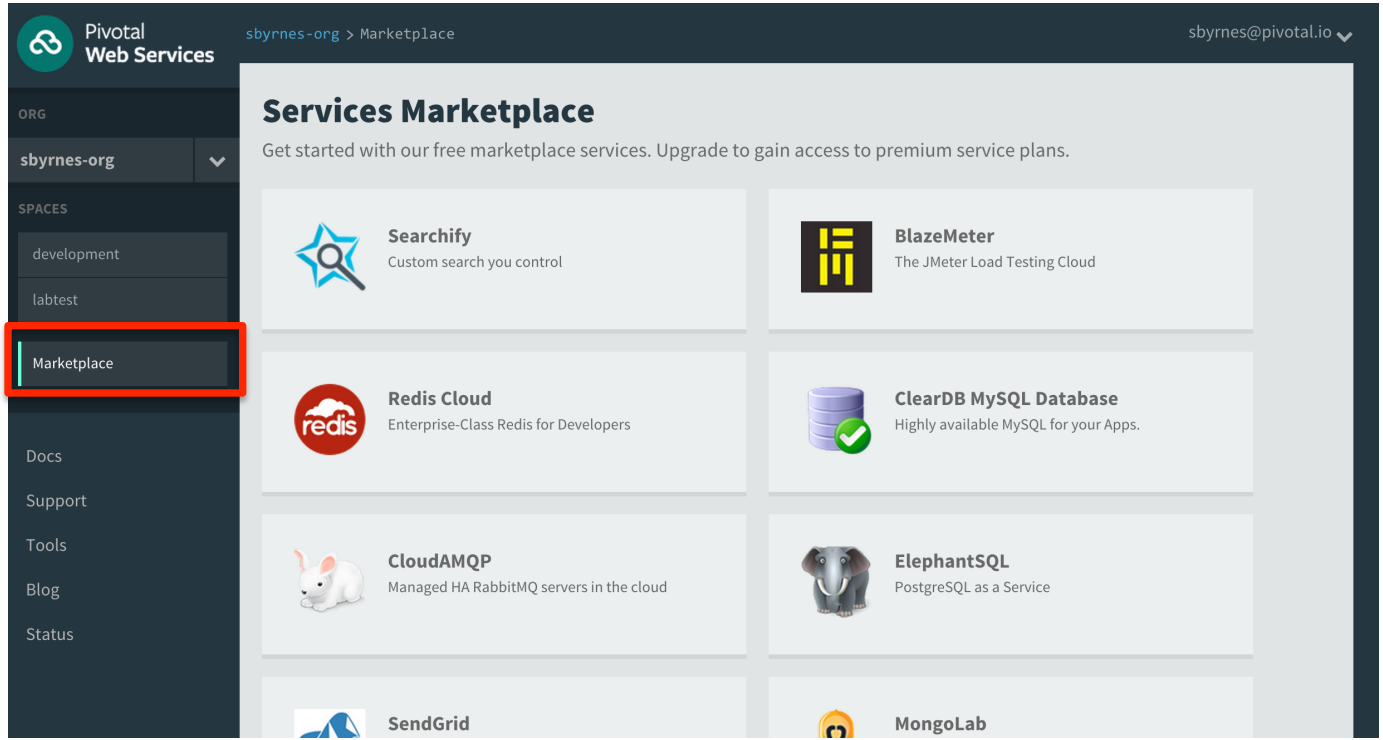
Installing Managed Services

- Many pre-packaged services for Pivotal Cloud Foundry
  - See https://network.pivotal.io
- Install using Ops Manager (an admin/operator tool)



Pivotal Web Services (PWS)

Marketplace Home Page in Apps Manager Console



Finding Available Managed Services

cf CLI

- Check the marketplace for available managed services
  - **`cf marketplace`**

files2 — bash — 157×28

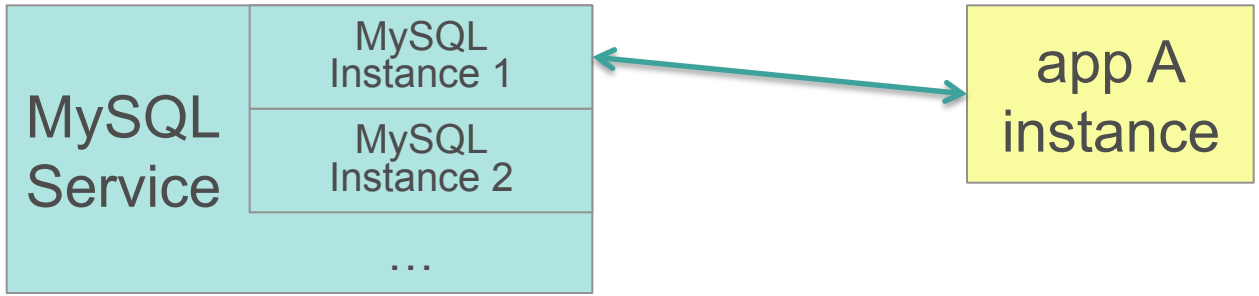
```
greylag:files2 sbyrnes$ cf marketplace
Getting services from marketplace in org sbyrnes-org / space development as sbyrnes@pivotal.io...
OK

service          plans                                                                                  description
3scale           free_appdirect, basic_appdirect*, pro_appdirect*                                       API Management Platform
blazemeter       free-tier, basic1kmr*, pro5kmr*, pp10kmr*                                              The JMeter Load Testing Cloud
cedexisopenmix   opx_global*, openmix-gslb-with-fusion-feeds*                                           Openmix Global Cloud &amp;amp; Data Center Load Balancer
cedexisradar     free-community-edition                                                                 Free Website&amp;amp; Mobile App Performance Reports
cleardb          spark, boost*, amp*, shock*                                                            Highly available MySQL for your Apps.
cloudamqp        lemur, tiger*, bunny*, rabbit*, panda*                                                 Managed HA RabbitMQ servers in the cloud
cloudforge       free, standard*, pro*                                                                  Development Tools In The Cloud
elephantsql      turtle, panda*, hippo*, elephant*                                                      PostgreSQL as a Service
flashreport      trial, basic*, silver*, gold*, platinum*                                               Generate PDF from your data
ironmq           pro_platinum*, free, pro_gold*, large*, medium*, small*, pro_silver*                   Powerful Durable Message Queueing Service
ironworker       large*, pro_gold*, pro_platinum*, pro_silver*, small*, medium*, free                   Scalable Background and Async Processing
loadimpact       lifree, li100*, li500*, li1000*                                                        Automated and on-demand performance testing
memcachedcloud   100mb*, 250mb*, 500mb*, 1gb*, 2-5gb*, 5gb*, 30mb                                      Enterprise-Class Memcached for Developers
memcachier       dev, 100*, 250*, 500*, 1000*, 2000*, 5000*, 7500*, 10000*, 20000*, 50000*, 100000*    The easiest, most advanced memcache.
mongolab         sandbox                                                                                Fully-managed MongoDB-as-a-Service
newrelic         standard                                                                               Manage and monitor your apps
pubnub           free                                                                                   Build Realtime Apps that Scale
rediscloud       100mb*, 250mb*, 500mb*, 1gb*, 2-5gb*, 5gb*, 10gb*, 50gb*, 30mb                         Enterprise-Class Redis for Developers
searchify        small*, plus*, pro*                                                                    Custom search you control
searchly         small*, micro*, professional*, advanced*, starter, business*, enterprise*              Search Made Simple. Powered-by Elasticsearch
sendgrid         free, bronze*, silver*                                                                 Email Delivery. Simplified.
stamplay         plus*, premium*, core, starter*                                                        API-first development platform
statica          starter, spike*, micro*, medium*, large*, enterprise*, premium*                        Enterprise Static IP Addresses
```

Pivotal

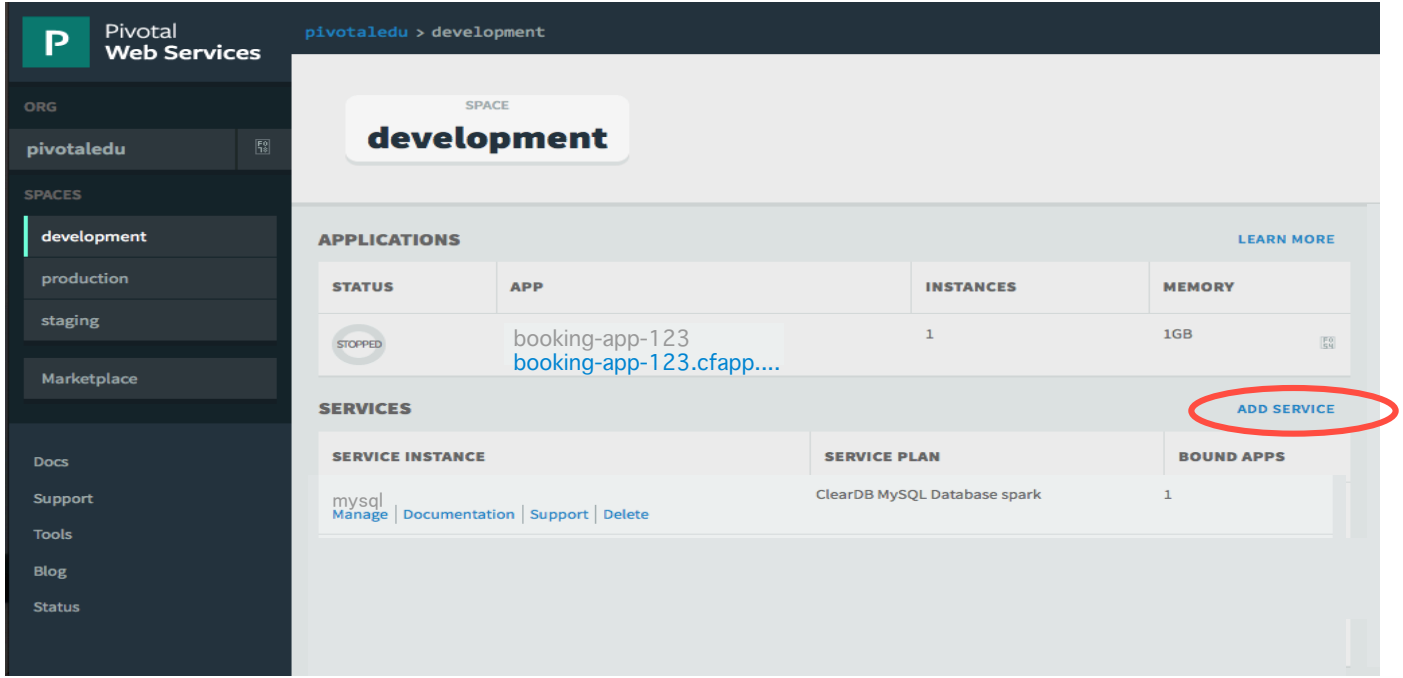
Service vs. Service Instance

- **Services** can create **services instances**
- Think of the service as a template for the service instance
  - For example, a MySQL service creates “tenants” of isolated database instances
  - A service instance is one of those tenants
- App instances only see the service instances they are bound to



Finding or Adding Service Instances- Apps Manager

Service instances are available to the **space**



Finding Existing Service Instances

cf CLI

- List existing service instances with **`cf services`**
  - In *current space*
  - Lists both managed and user-provided service instances
- In this example: one service instance called mydb

```
example$ cf services
Getting services in org pivotaledu / space development as user@domain...
OK

name        service     plan    bound-apps
mydb        cleardb     spark   booking-app-123
```

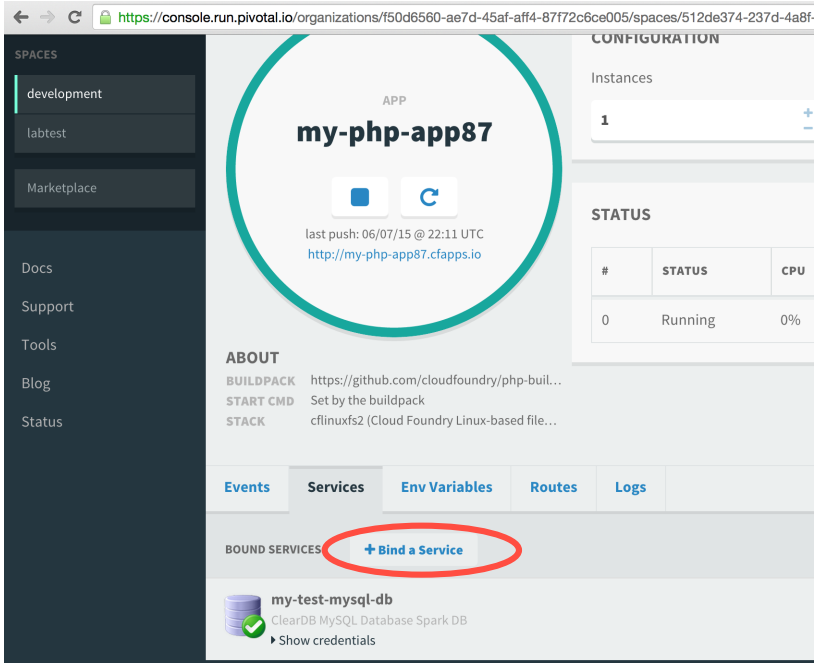
Adding a Managed Service Instance to a Space (cf CLI)

- Use **`cf create-service`**
  - Allows selection of service and plan

- The service instance becomes available to applications in the *current space*
  - For multiple spaces, run **`cf create-service`** in *each* space

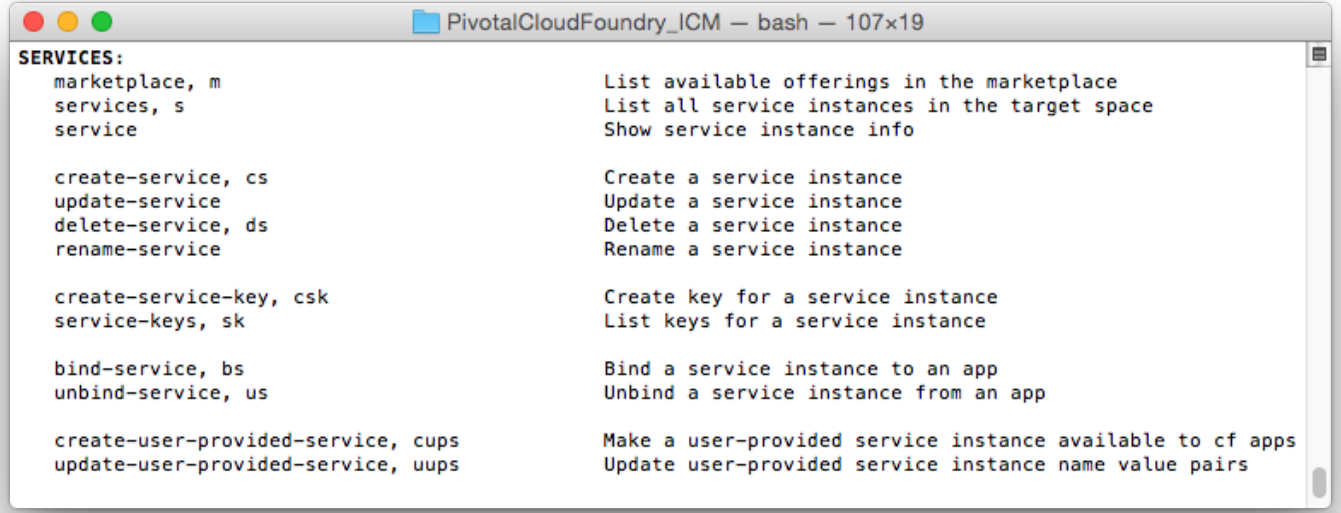
Binding Service Instances to an Application
Apps Manager

Applies to managed and user-provided service instances



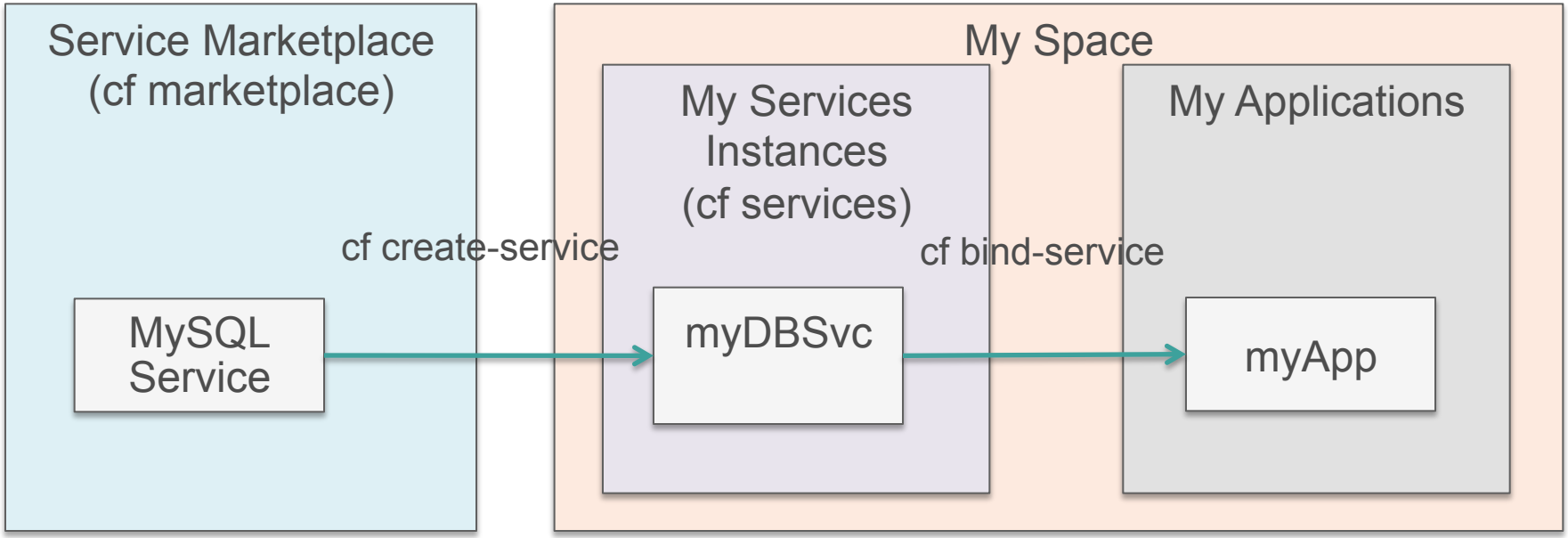
Binding Service Instances to an Application
cf CLI

- Use **`cf bind-service`**
  - Applies to managed and user-provided service instances



Pivotal

Managed Services, Service Instances and Applications



Binding Service Instances to an Application

Application Manifest

- You can use the **`services`** section of the application manifest to bind a service instance to an application with **`cf push`**
  - Applies to managed and user-provided service instances

```
---
applications:
- name: myapp
  memory: 128M
  services:
  - myservice
```

Example: Creating and Binding a Postgres Service

```
example$ cf services
Getting services in org pivotaledu / space development as user@domain...
OK
name       service     plan    bound-apps
(no services)

example$ cf create-service elephantsql turtle mypg
Creating service mypg in org pivotaledu / space development as user@domain...
OK

example$ cf services
Getting services in org pivotaledu / space development as user@domain...
OK
name       service     plan    bound-apps
mypg       elephantsql turtle

example$ cf bind-service booking-app-456 mypg
Binding service mypg to booking-app-456 in org pivotaledu / space development as
user@domain...
OK
TIP Use 'cf restage' to ensure your env variable changes take effect
```

Pivotal

Topics

- Introduction to services
- Create and bind managed services
- **Use a managed service in an application**
- Create user-provided services

Using a Managed Service – Developer View

- Create a service instance for the space (if necessary)
- Bind it to your application
- Modify your application to read configuration information via environment variables (if necessary)

DEVELOPER

cf create-service <myService>

cf bind-service <myApp> <myService>

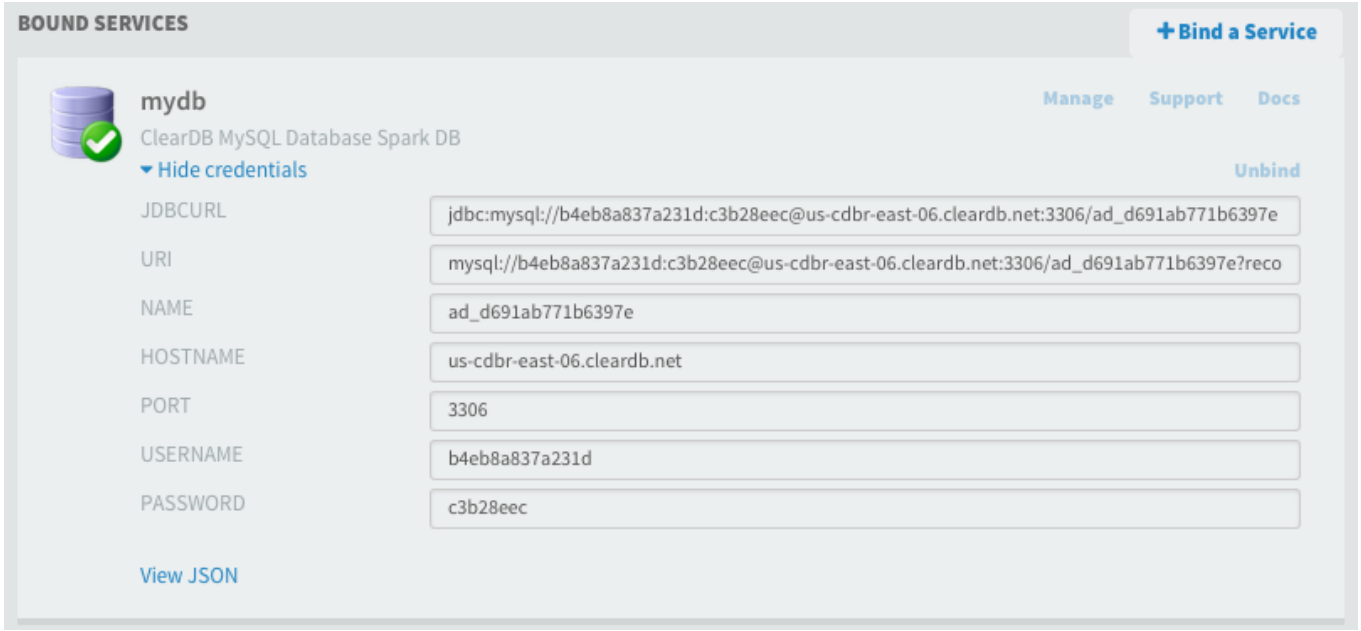
Accessing Connection Information (1 of 2)

- The platform creates user accounts and passwords for service instances
  - No need to hard-code any connection information
- After binding a service instance, connection information is available to applications in environment variables: **`VCAP_SERVICES`**
- Once the application is *staged*, view connection information using
  - *`cf env [app-name]`*
  - Look for **`VCAP_SERVICES`** in the output

Pivotal

Accessing Connection Information (2 of 2)

Connection information also available via Apps Manager:



VCAP_SERVICES Property

```
VCAP_SERVICES=
{
  cleardb-n/a: [
    {
      name: "cleardb-1",
      label: "cleardb-n/a",
      plan: "spark",
      credentials: {
        name: "ad_c6f4446532610ab",
        hostname: "us-cdbr-east-03.cleardb.com",
        port: "3306",
        username: "b5d435f40dd2b2",
        password: "ebfc00ac",
        uri: "mysql://b5d435f40dd2b2:ebfc00ac@us-cdbr-east-
                          03.cleardb.com:3306/ad_c6f4446532610ab",
        jdbcUrl: "jdbc:mysql://b5d435f40dd2b2:ebfc00ac@us-
                  cdbr-east-03.cleardb.com:3306/ad_c6f4446532610ab"
      }
  ...
```

Just a very long string in JSON format

Parse to extract credentials

Topics

- Introduction to services
- Create and bind managed services
- Use a managed service in an application
- **Create user-provided services**

User Provided Service Instances

- Typically represent external assets like legacy databases
- Behave like other service instances once created
- When bound, they provide service instance credentials and other configuration information to applications
  - Avoids hard coding service instance endpoints

http://docs.pivotal.io/pivotalcf/devguide/services/user-provided.html

Pivotal

Creating User Provided Service Instances

- Use **`cf create-user-provided-service`** or **`cf cups`**
  - Provide name and parameters/credentials
  - All applications bound to *same* instance in *same* way
- The service will be listed in **`cf services`** for the target space and is available for binding to applications

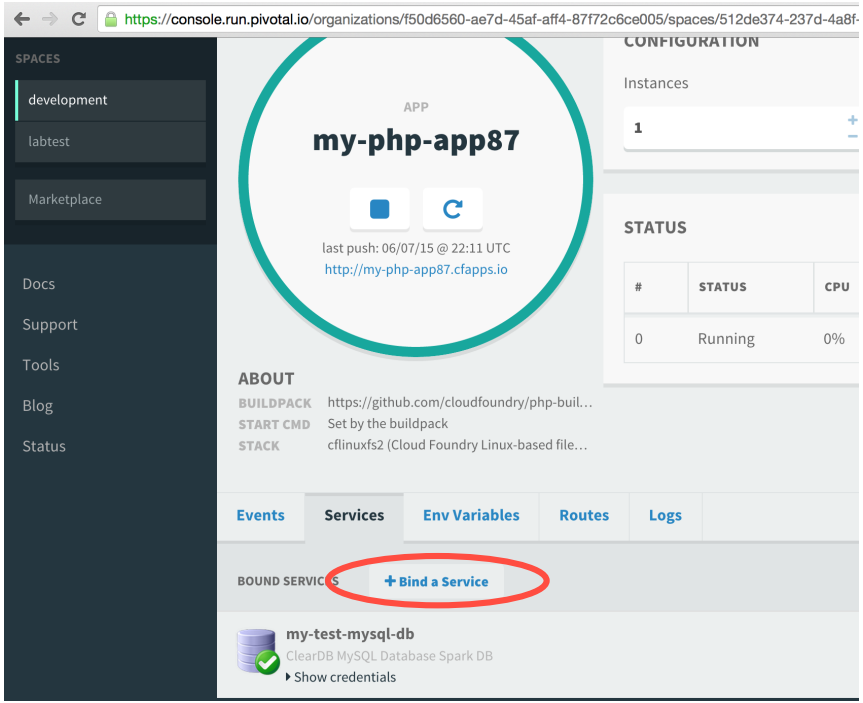
```
$ cf cups mydb -p "hostname, port, username, password, name"
hostname> db.example.com
port> 1234
username> dbuser
password> dbpasswd
name> mydb
Creating user provided service mydb ... OK
```

specify *any* list of parameters here

prompts for parameter values

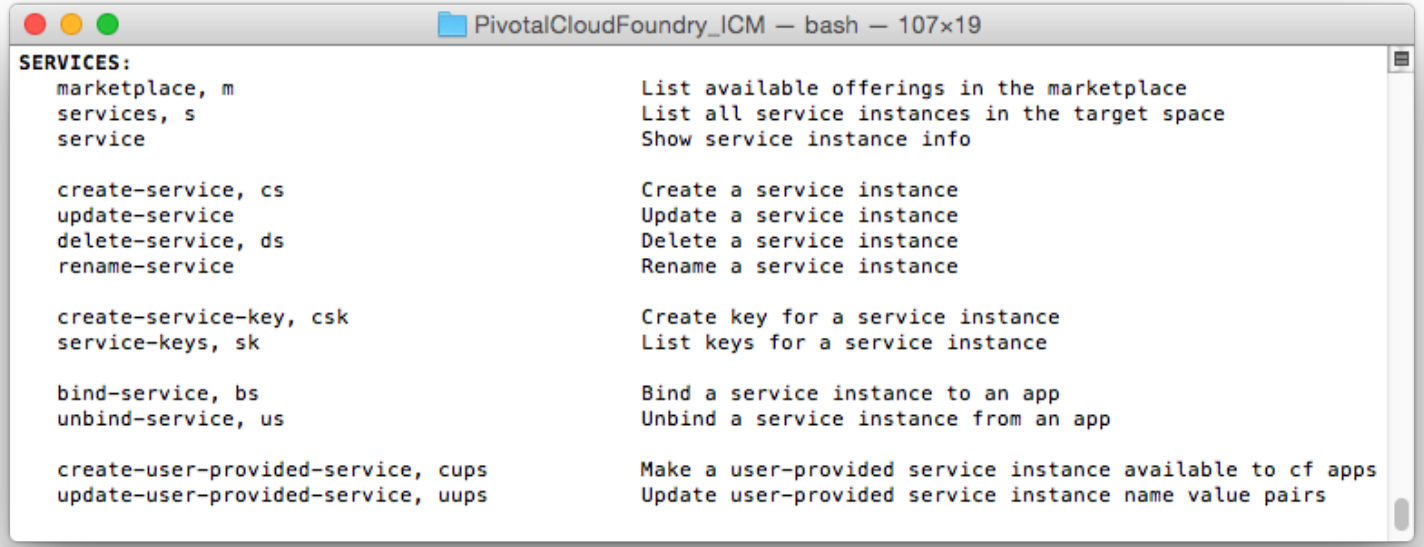
Binding User-Provided Service Instances to an Application
Apps Manager

Applies to managed and user-provided service instances



Binding User-Provided Service Instances to an Application
cf CLI

- Use *cf bind-service*
  - Applies to managed and user-provided service instances



Pivotal

User Provided Services - Accessing

- Bound service properties available in **VCAP_SERVICES** environment variable
- In your code
  - Access variable
  - Parse JSON
  - Use to connect

```
{
  user-provided: [
    {
      name: "mydb",
      label: "user-provided",
      tags: [ ],
      credentials: {
        hostname: "db.example.com",
        port: "1234",
        username: "dbuser",
        password: "dbpasswd",
        name: "mydb"
      }
    }
  ]
}
```

Pivotal

Topics

- Introduction to services
- Create and bind managed services
- Use a managed service in an application
- Create user-provided services

Pivotal

Lab

Create a managed service instance and bind to it