

Pivotal®

Cloud Foundry

101 Overview

SECTION 1 PRE-REQUISITES & INTRO

DevNexus 2107 - Cloud, Containers, Serverless Computing, Docker : Oh my!

Pre-Requisites

- Windows or Mac OS

Java Dependencies

- Java JDK 1.7+
- Maven / Gradle
- PCF CLI (<https://console.run.pivotal.io/2/tools>)
- PCF account (Pivotal Web Services “Trial account” will be best) Go to run.pivotal.io and click the “Sign up for free” button.
- Your preferred git client or <https://git-scm.com/downloads>
- Chrome or Firefox....(not IE)

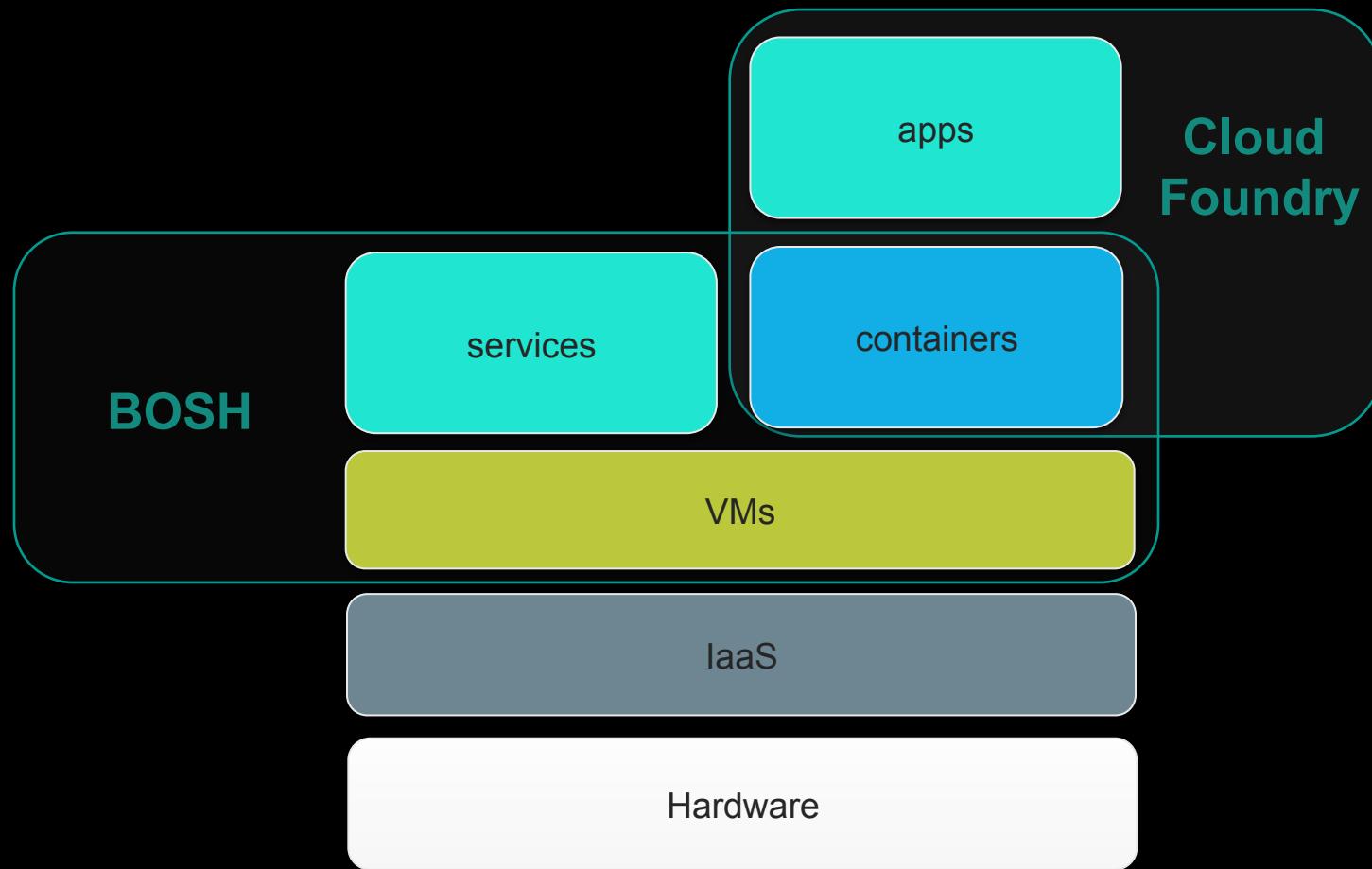
Agenda

- Morning Sessions
 - Getting Started with Cloud Foundry
 - Language, Docker, and Container Support
- Afternoon Sessions
 - “Day 2” Capabilities
 - Monitoring
 - Patching
 - Upgrading via Blue/Green
 - Getting Started with SpringBoot
 - Cloud-Native Micro-Services & Beyond

Cloud Foundry....

- Open Source Cloud Computing Platform
- Developed at VMware
 - Now owned by Pivotal Software
 - Pivotal is a Joint Venture of EMC, Vmware, GE, Msft, and Ford
- Designed for
 - Fast application development and deployment.
 - Highly scalable and available architecture.
 - DevOps-friendly workflows.
 - Reduced chance of human error.
 - Multi-tenant compute efficiencies.
 - High degree of Operational support via autonomous resource mgmt.

How does it work?





SECTION 2 CF LOGIN & CF TARGET



← Apps Manager

Welcome!

mgunter@pivotal.io

.....

SIGN IN

Create account

Reset password

CLI
cf login....

MGunter-MB-Pro:crashcourse mgunter\$ cf login
API endpoint: <https://api.run.pivotal.io>

Email> mgunter@pivotal.io

Password>

Authenticating...

OK

```
> cf help
```

NAME:

cf - A command line tool to interact with Cloud Foundry

USAGE:

```
cf [global options] command [arguments...] [command options]
```

VERSION:

6.20.0+25b1961-2016-06-29

GETTING STARTED:

help
version
login
logout
passwd
target

Show help
Print the version
Log user in
Log user out
Change user password
Set or view the targeted org or space

```
> cf apps -h
```

NAME:

apps - List all apps in the target space

USAGE:

cf apps

ALIAS:

a



List Commands



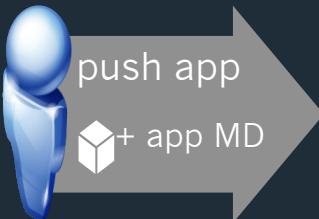
Show help for a single command

A large crowd of people walking through a modern glass and steel building, likely an airport terminal, with their reflections visible on the polished floor.

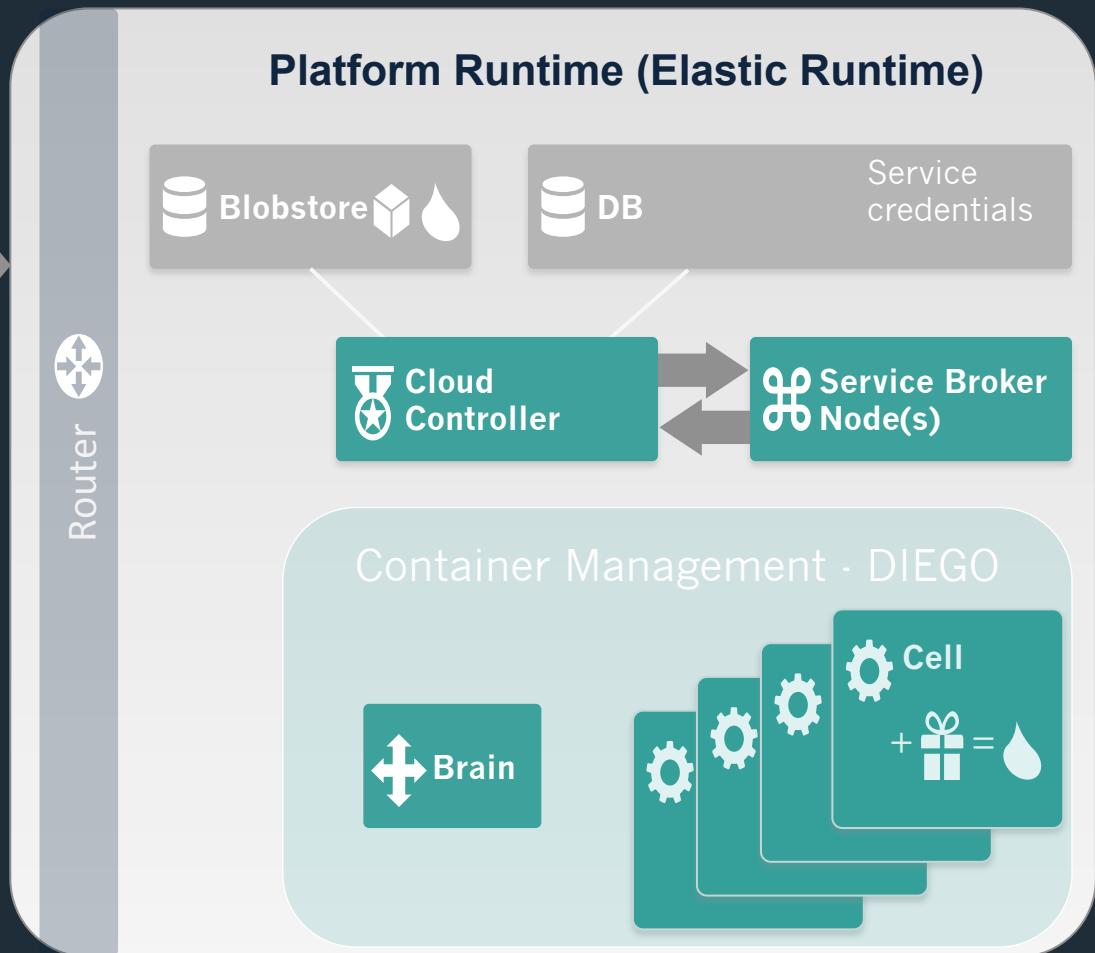
SECTION 3 CF PUSH & RUN

CF PUSH

- ① Upload app bits and metadata



- ② Create and bind services
- ③ Stage application
- ④ Deploy application
- ⑤ Manage application health



Pivotal™

Spring Music ♪



Albums

[view as: | sort by: [title](#) [artist](#) [year](#) [genre](#) | [+add an album](#)]

Achtung Baby

U2

1991

Rock



Nevermind

Nirvana

1991

Rock



Abbey Road

The Beatles

1969

Rock



Rumours

Fleetwood Mac

1977

Rock



Sun Sessions

Elvis Presley

1976

Rock



Thriller

Michael Jackson

1982

Pop



Exile on Main Street

The Rolling Stones

1972

Rock



Born to Run

Bruce Springsteen

1975

Rock



London Calling

The Clash

1980

Rock



Hotel California

The Eagles

1976

Rock



Led Zeppelin

Led Zeppelin

1969

Rock



IV

Led Zeppelin

1971

Rock



THE TWELVE FACTORS

I. Codebase

One codebase tracked in revision control, many deploys

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

Store config in the environment

IV. Backing services

Treat backing services as attached resources

V. Build, release, run

Strictly separate build and run stages

VI. Processes

Execute the app as one or more stateless processes

VII. Port binding

Export services via port binding

VIII. Concurrency

Scale out via the process model

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity

Keep development, staging, and production as similar as possible

XI. Logs

Treat logs as event streams

XII. Admin processes

Run admin/management tasks as one-off processes

Manifest

```
1  ---
2  instances: 1
3  memory: 1024M
4  applications:
5  - name: fortune-service
6    host: fortunes-mg
7    path: fortune-teller-fortune-service/target/fortune-teller-fortune-service-0.0.1-SNAPSHOT.jar
8    services:
9      - fortunes-db
10     - config-server
11     - service-registry
12   - name: fortune-ui
13     host: fortunes-ui-mg
14     path: fortune-teller-ui/target/fortune-teller-ui-0.0.1-SNAPSHOT.jar
15     services:
16       - config-server
17       - service-registry
18       - circuit-breaker-dashboard
19 env:
20   SPRING_PROFILES_ACTIVE: pcf
21   CF_TARGET: https://api.run.pivotal.io
```

manifest.yml

- ✓ Instances
- ✓ Memory
- ✓ Multiple apps
- ✓ Names, hosts, path
- ✓ Services
- ✓ Environment vars

- iii. [Always Provide an Application Name to cf push](#)
- iv. [How cf push Finds the Application](#)
- v. [Precedence Between Manifests, Command Line Options, and Most Recent Values](#)
- vi. [Optional Attributes](#)
 - o [The buildpack attribute](#)
 - o [The command attribute](#)
 - o [The disk quota attribute](#)
 - o [The domain attribute](#)
 - o [The domains attribute](#)
 - o [The stack attribute](#)
 - o [The instances attribute](#)
 - o [The memory attribute](#)
 - o [The health-check-type attribute](#)
 - o [The host attribute](#)
 - o [The hosts attribute](#)
 - o [The no-hostname attribute](#)
 - o [The routes attribute](#)
 - o [The random-route attribute](#)
 - o [The path attribute](#)
 - o [The timeout attribute](#)
 - o [The no-route attribute](#)
 - o [Environment Variables](#)
 - o [Services](#)
- vii. [Describing Multiple Applications with One Manifest](#)
- viii. [Minimizing Duplication](#)
- ix. [Multiple Manifests with Inheritance](#)

Documentation on
Manifest.yml

Domains

- Each Cloud Foundry installation has a default *app domain*
- Domains provide a namespace from which to create routes
- Requests for any routes created from the domain will be routed to Elastic Runtime.
- Domains can be shared or private in regards to PCF organizations

The screenshot shows the Pivotal Apps Manager interface for the organization 'mborges-org'. The left sidebar lists 'ORG' (mborges-org), 'SPACES' (development, production, Marketplace), and links to 'Docs', 'Support', and 'Tools'. The main content area displays organization details: 'ORG: mborges-org', 'QUOTA: 2% 128 MB of 5 GB Limit', '2 Spaces', '1 Domain', '2 Members', and a 'DOMAINS' section with a button to 'Add a Domain'. A table lists domains under 'NAME': 'south.fe.pivot.alio' (SHARED). The 'south.fe.pivot.alio' entry is highlighted with a red border.

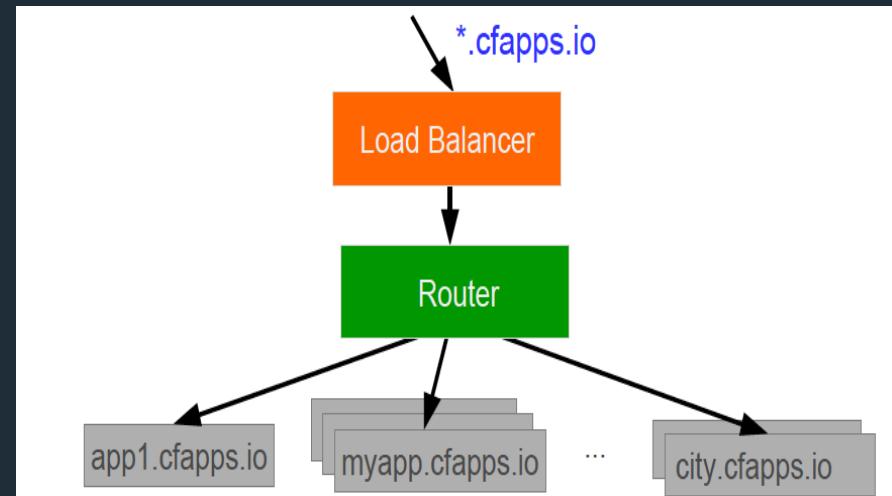
Routes

- HTTP requests are routed to apps pushed by a “ROUTE”
- Many Appsto a Route
- Many Routes.... to an App
- Routes belong to a space

The screenshot shows the Pivotal Apps Manager interface for the 'mborges-org > development > pcf-scale-prod' space. The main area displays the app 'pcf-scale-prod' with its configuration (1 instance, 128 MB memory, 1 GB disk) and status (Running, 0% CPU, 88.3 MB memory, 84.2 MB disk, 1 d 2 hr 31 min uptime). A large green circle highlights the 'APP' section. Below it, the 'ABOUT' section provides details like buildpack (Ruby), start cmd (Set by the buildpack), and stack (clinuxfs2). The 'Routes' tab is selected, showing two mapped routes:
https://pcf-scale-prod.south.fe.pivotal.io
https://pcf-scale-v1_2.south.fe.pivotal.io

Routes and Domains

- A wildcard entry (*) is added to the DNS for the app domain
- The Router uses the subdomain to map to application instance(s)
- Host and Domain are in a shared namespace.
- No duplicates allowed!





SECTION 4 CF MARKETPLACE & CF BIND

“Marketplace” of Managed Services

Managed Services are integrated with PCF by implementing a documented API.

A **self-service** catalog of databases, analytics and middleware technologies:

Services ▾

-  **App Autoscaler**
Scales bound applications in response to load (beta) >
-  **Circuit Breaker**
Circuit Breaker Dashboard for Spring Cloud Applications >
-  **Config Server**
Config Server for Spring Cloud Applications >
-  **MongoDB**
NoSQL Database >
-  **MongoDB**
NoSQL Database >
-  **MySQL**
MySQL databases on demand >
-  **RabbitMQ**
RabbitMQ is a robust and scalable high-performance multi-prot... >
-  **Service Registry**
Service Registry for Spring Cloud Applications >

Push PCFDemoand Bind a RabbitMQ service

BEST RETAILER INC.

ORDERS US HEAT MAP

START DATA STREAM

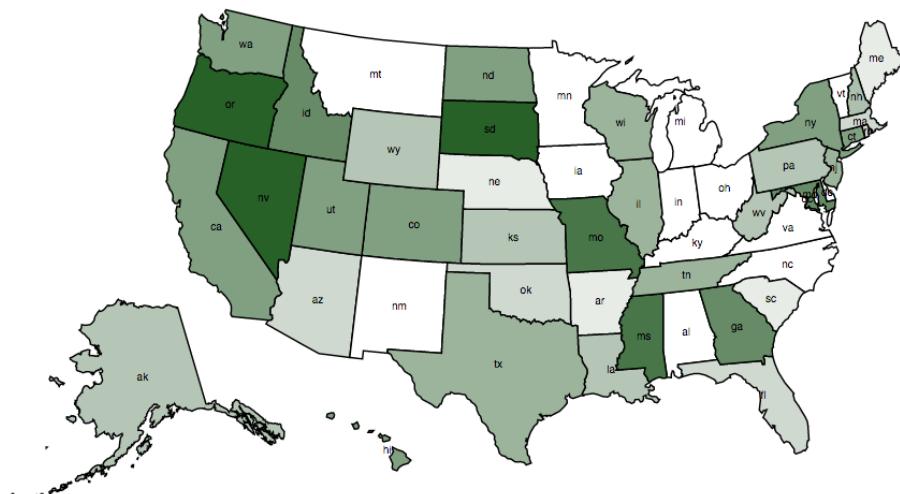
KILL APP

Instance hosted at 10.254.0.30:8080

Instance Index 0

No RabbitMQ service bound - streaming is not active

Orders density per US State (tip: click on a state for details)



© 2014 Pivotal Software, Inc.

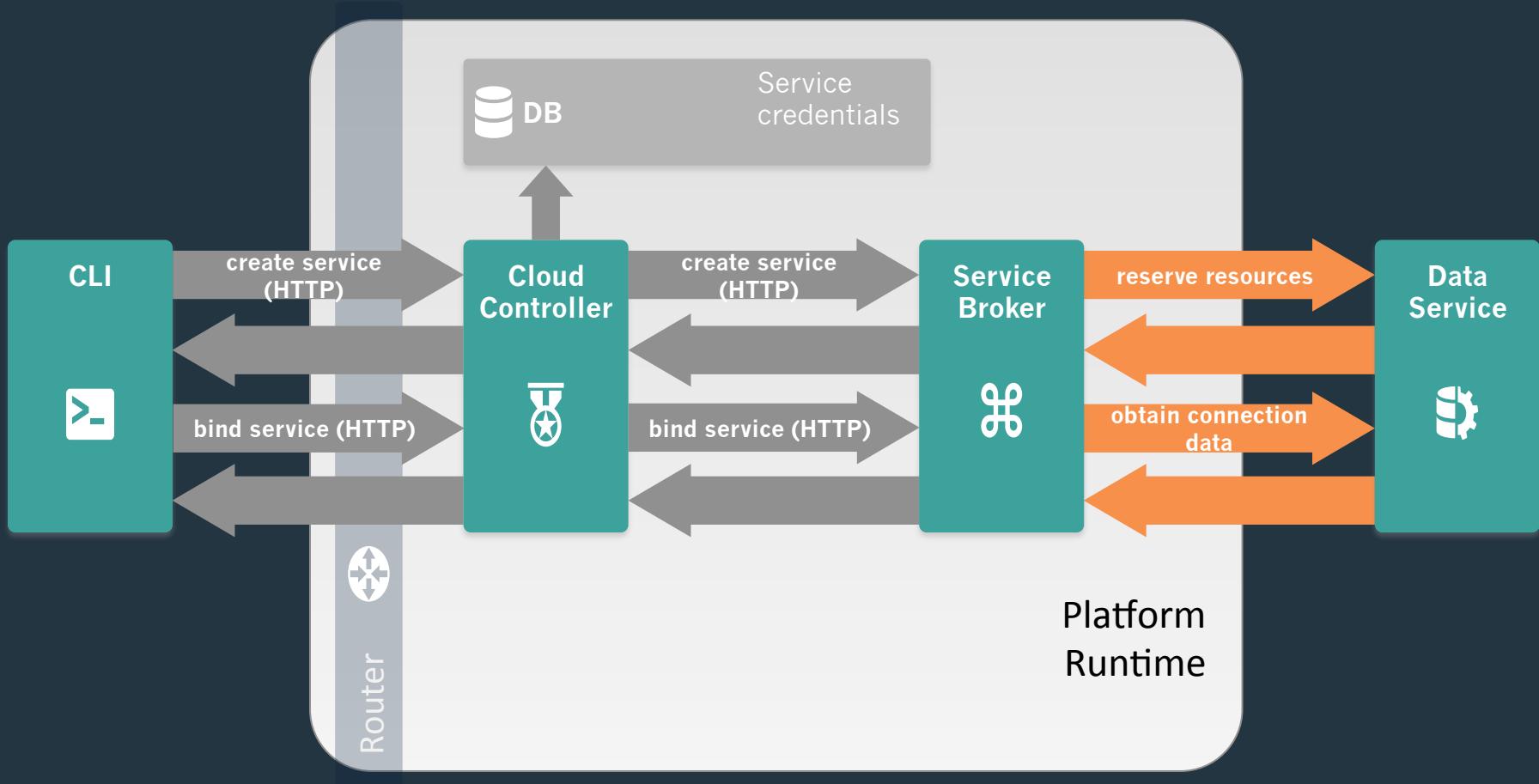
Powered by

Pivotal



RabbitMQ
RabbitMQ is

Creating and binding services



What about Services NOT in the marketplace?

- Hard-code Connection Info
- Environment Variables
- User-Provided Services:

```
MGunter-MB-Pro:demos mgunter$ cf cups my-secret-db -p "username, password, url"
username> mgunter
password> asdfasdfa
url> http://whereever
Creating user provided service my-secret-db in org Southeast / space mgunter-sp
OK
```

User-Provided Services

EXAMPLES:

```
cf create-user-provided-service my-db-mine -p "username, password"  
cf create-user-provided-service my-db-mine -p /path/to/credentials.json  
cf create-user-provided-service my-drain-service -l syslog://example.com  
cf create-user-provided-service my-route-service -r https://example.com
```

Linux/Mac:

```
cf create-user-provided-service my-db-mine -p '{"username": "admin", "password": "pa55woRD"}'
```

Windows Command Line:

```
cf create-user-provided-service my-db-mine -p "{\"username\": \"admin\", \"password\": \"pa55woRD\"}"
```

Binding Services to Apps

```
$ cf bind-service my-app mydb
Binding service mydb to my-app in org my-org / space test as me@example.com...
OK
TIP: Use 'cf push' to ensure your env variable changes take effect

$ cf restart my-app
```



Note: You must restart or in some cases re-push your application for changes to be applied to the [VCAP_SERVICES](#) environment variable and for the application to recognize these changes.



SECTION 5 CF ENV

System-Provided:

Env Vars

```
{  
  "VCAP_APPLICATION": {  
    "application_id": "e6bc1d1e-4383-4311-972c-3dd33e278ddb",  
    "application_name": "circuit-breaker-dashboard",  
    "application_uris": [  
      "hystrix-dashboard-meristematic-nytril.cfapps.io"  
    ],  
    "application_version": "aeab9221-4c79-4b7c-b514-346b7723ab91",  
    "limits": {  
      "disk": 1024,  
      "fds": 16384,  
      "mem": 384  
    },  
    "name": "circuit-breaker-dashboard",  
    "space_id": "d20566e3-67c4-43e3-8c27-19f8951424be",  
    "space_name": "experimental",  
    "uris": [  
      "hystrix-dashboard-meristematic-nytril.cfapps.io"  
    ],  
    "users": null,  
    "version": "aeab9221-4c79-4b7c-b514-346b7723ab91"  
  }  
}
```

User-Provided:

MY_ENV_VAR: some-random-string

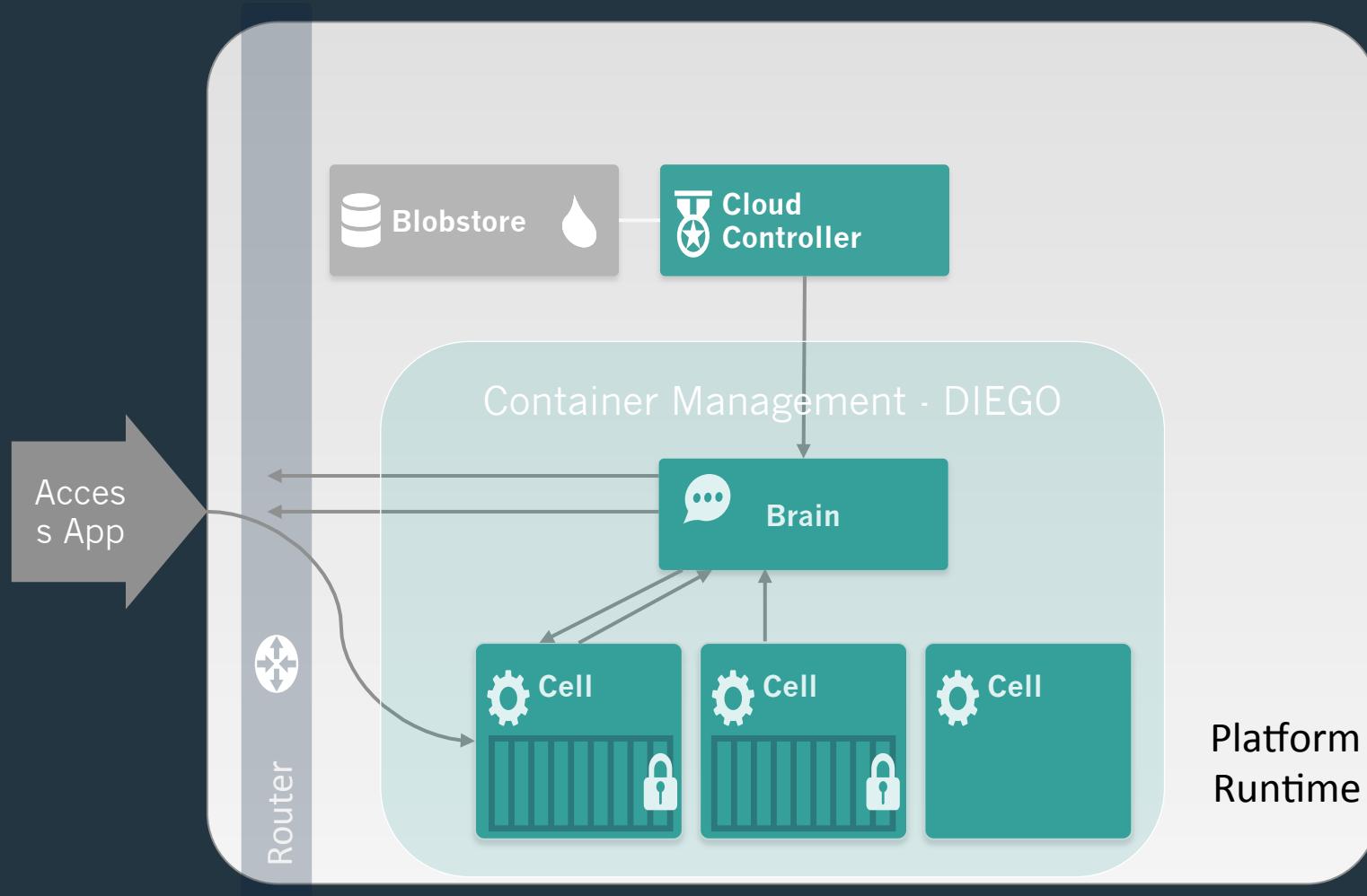
Other Environment Variables

ii. Application-Specific System Variables

- CF_INSTANCE_ADDR
- CF_INSTANCE_GUID
- CF_INSTANCE_INDEX
- CF_INSTANCE_IP
- CF_INSTANCE_PORT
- CF_INSTANCE_PORTS
- HOME
- MEMORY_LIMIT
- PORT
- PWD
- TMPDIR
- USER
- VCAP_APP_HOST
- VCAP_APP_PORT
- VCAP_APPLICATION
- VCAP_SERVICES

SECTION 6 CF SCALE & QUOTA/ROLES

The App-Container Approach - Scaling



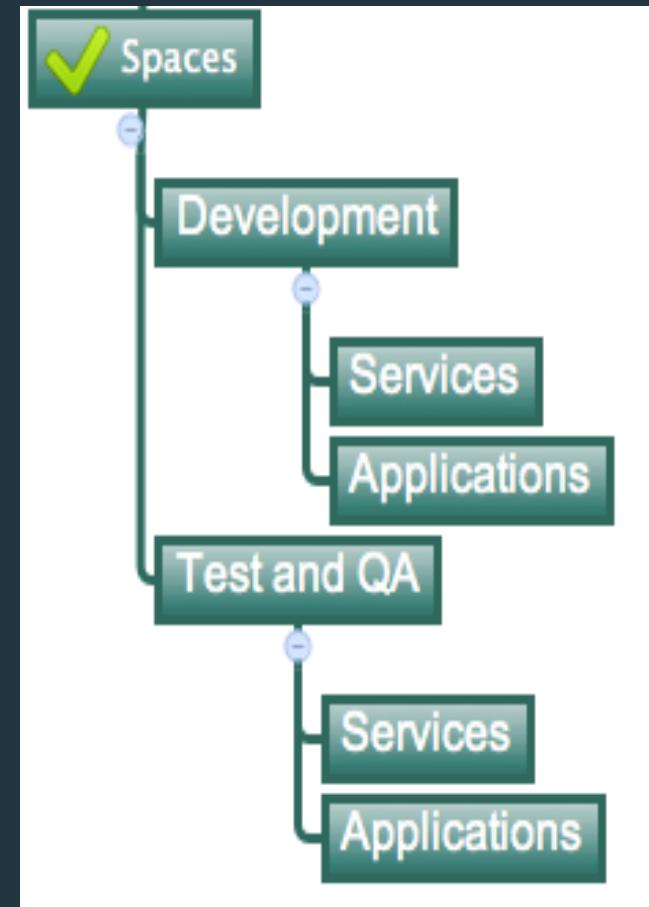
Organizations

- Top-most administrative unit
- Logical division division
- Each organization has its own users and assigned quota
- User permissions / Roles are specified per space within an organization
- Sub-divided into spaces



Spaces

- Logical sub-division within an organization
- Users authorized at an organization level can have different roles per space
- Services, Routes, and Applications are created / target per Space

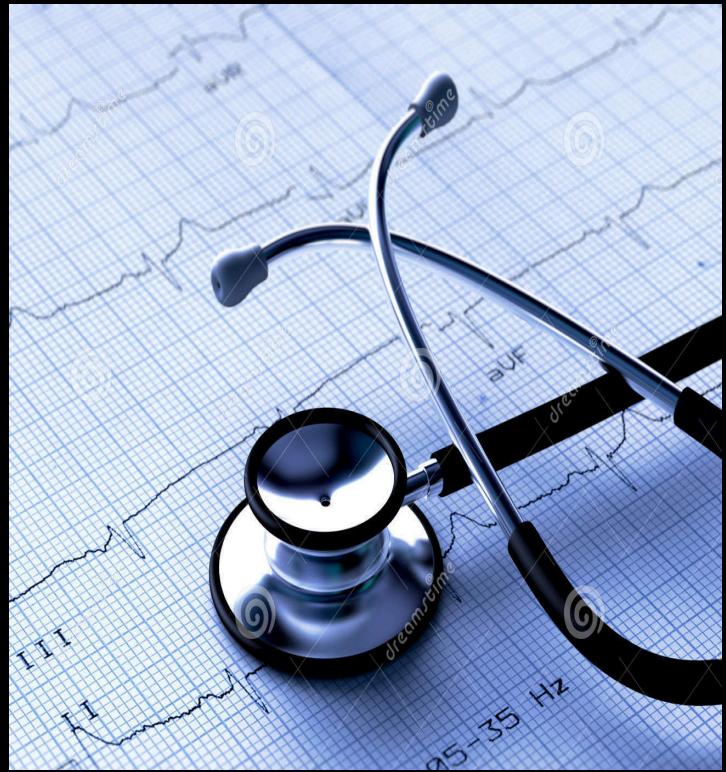


A dark, atmospheric photograph showing the silhouettes of many people walking through a modern building. The building features a large glass wall with a grid pattern, a curved ceiling, and several overhead lights. The scene is dimly lit, with the subjects appearing as dark shapes against a lighter background.

SECTION 7 HEALTH MONITORING & TROUBLESHOOTING

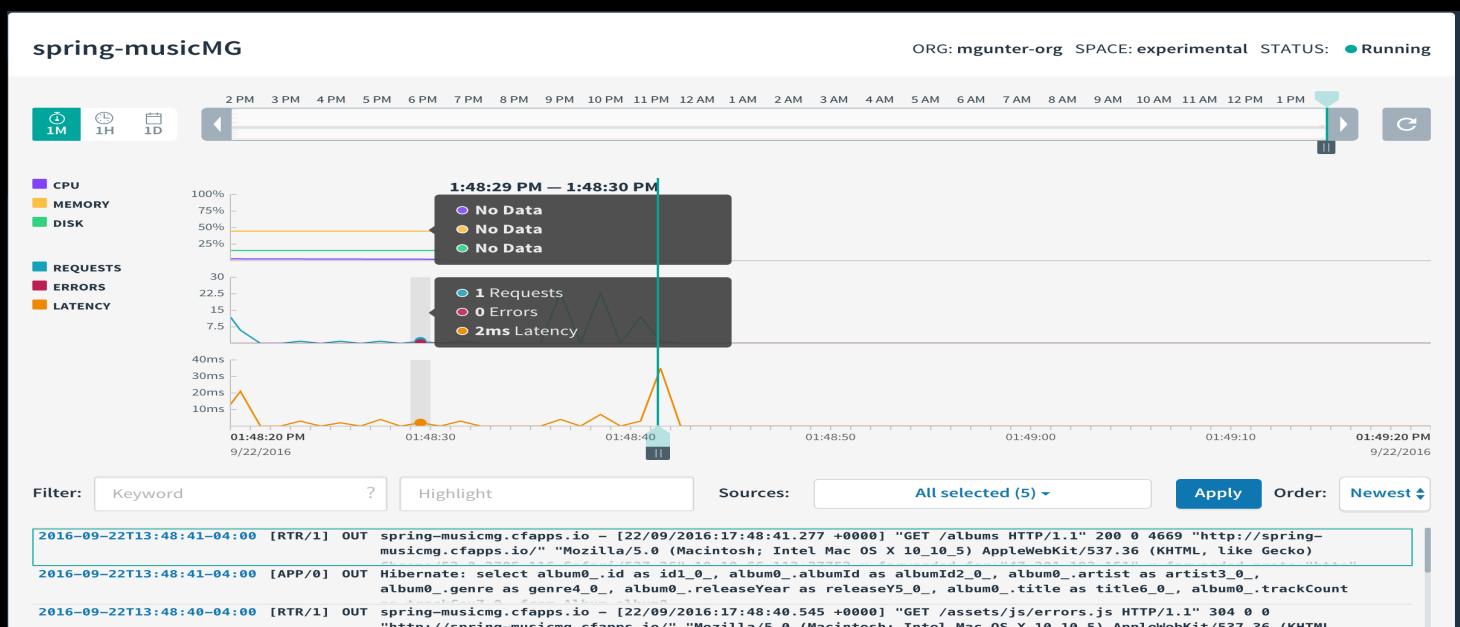
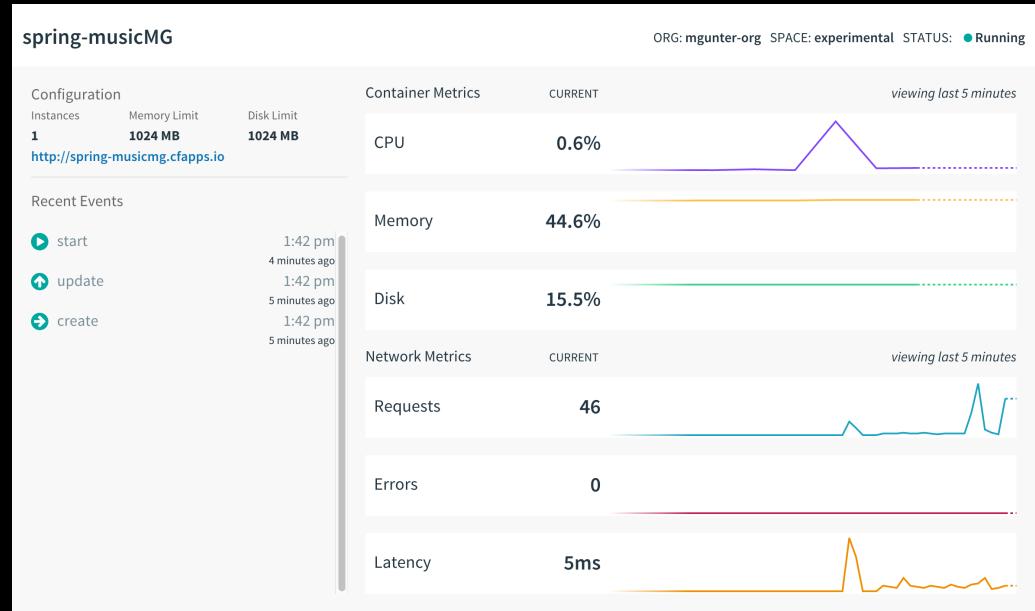
Viewing your app “vitals”

- cf events < app name >
- cf app < app name >
- cf logs < app name > --recent
- PCF Metrix



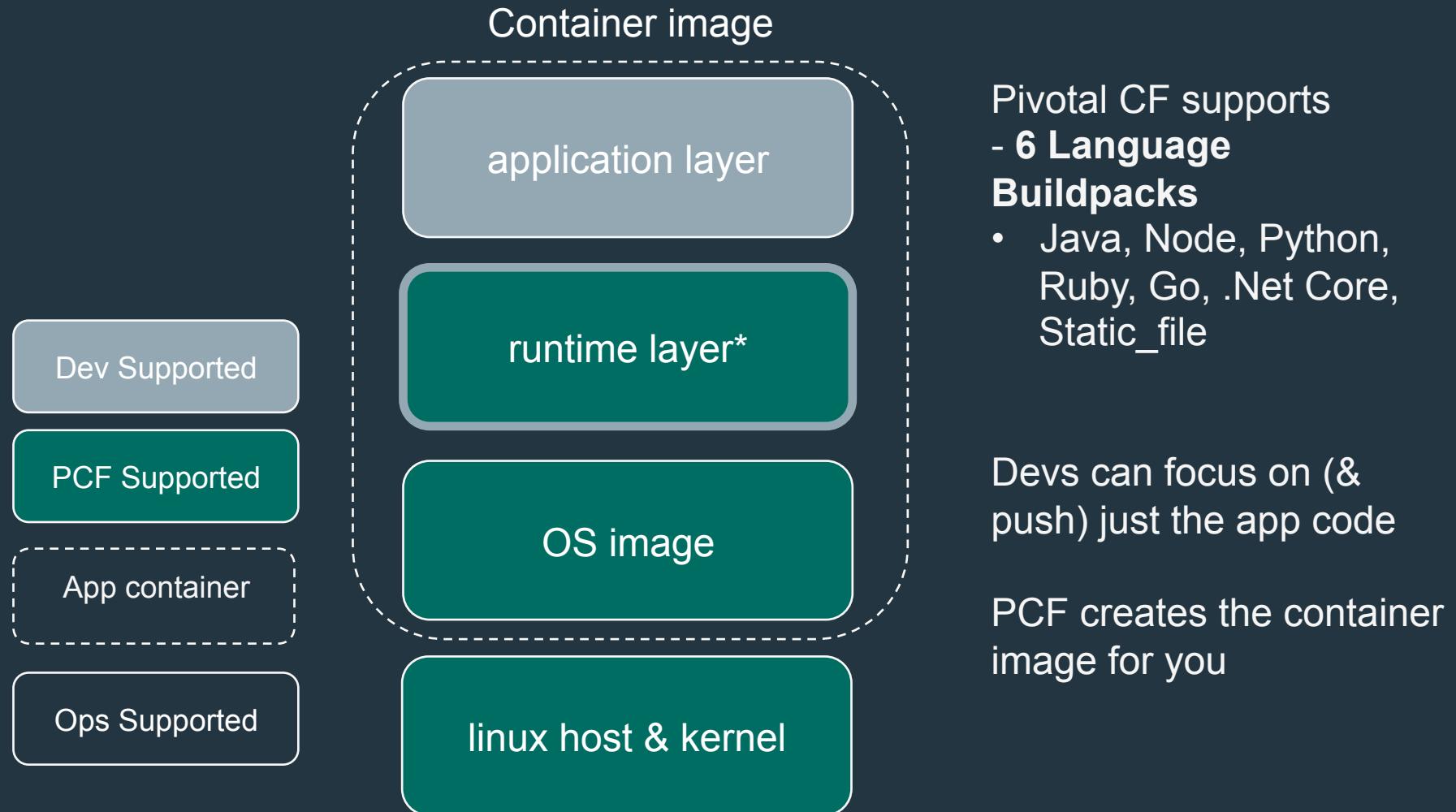
PCF Metrics....

Detailed performance info about your app.



SECTION 8 LANGUAGES & BUILDPACKS

Platform Runtime – Buildpacks



* Devs may bring a custom buildpack

Push your own Example App

Use one of the examples:

```
MGunter-MB-Pro:sample-apps mgunter$ tree -L
.
├── dotnetcore-app
├── go-sample-app
├── node-app
├── python-app
├── rails-sample-app
├── spring-app
└── staticfile-app
```

Make a change and re-push.

A dark, grainy photograph of a modern building's interior, possibly an airport or a large terminal. The space is characterized by a grid of vertical and horizontal steel beams supporting a glass roof. The floor is polished and reflects the overhead lights. In the foreground and middle ground, numerous people are walking in various directions, their figures appearing as dark silhouettes against the lighter background. The overall atmosphere is one of a busy, modern public space.

SECTION 9 WHERE TO GO FOR HELP?

Informational Resources:

CF Docs: <https://docs.cloudfoundry.org/>

PCF Docs: <https://docs.pivotal.io/pivotalcf>

This Workshop: <https://github.com/Pivotal-Field-Engineering/DevNexus2017>

Support for PCF Buildpacks, Spring, Docker and Tomcat

runtime layer



Buildpacks support

- Java, Node, Python, Ruby, Go, .Net Core(beta), Static_file

Enterprise Middleware Support

- Apache Tomcat Support from #1 Committer

Enterprise Framework Support

- Spring is the #1 Enterprise Java Framework