

Pivotal.

# Pivotal Cloud Foundry App Fundamentals

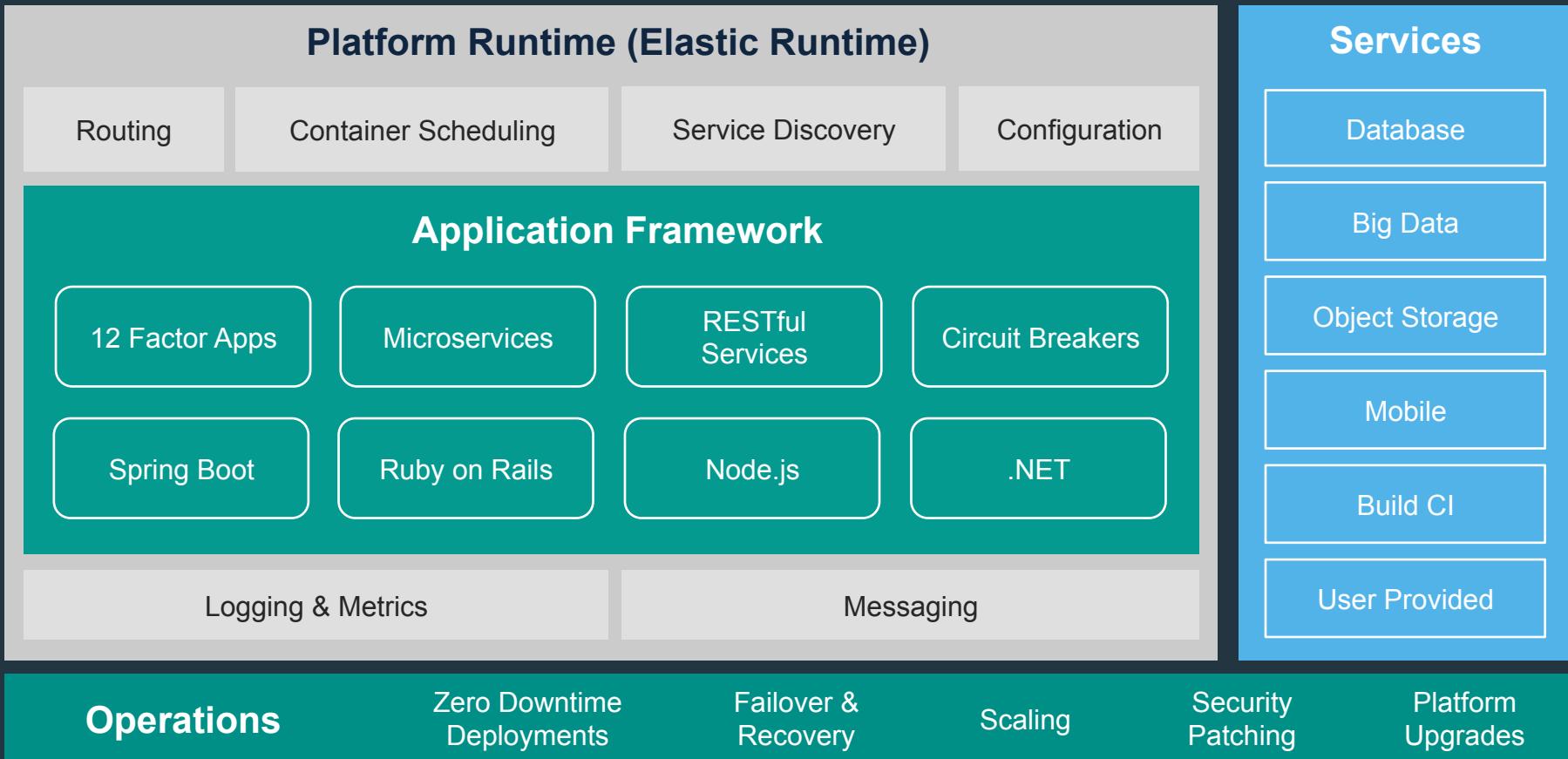


# Agenda

---

- Push / Bind / Scale
- CLI, Manifests, API & Aps Mgr
- Orgs, Spaces, Users & Quotas
- Buildpacks
- Logging
- Domains and routes
- Basic Services Overview
- Blue/Green Upgrades

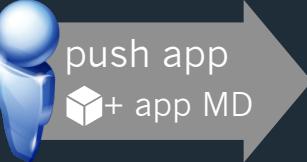
# Pivotal Cloud Foundry Architecture

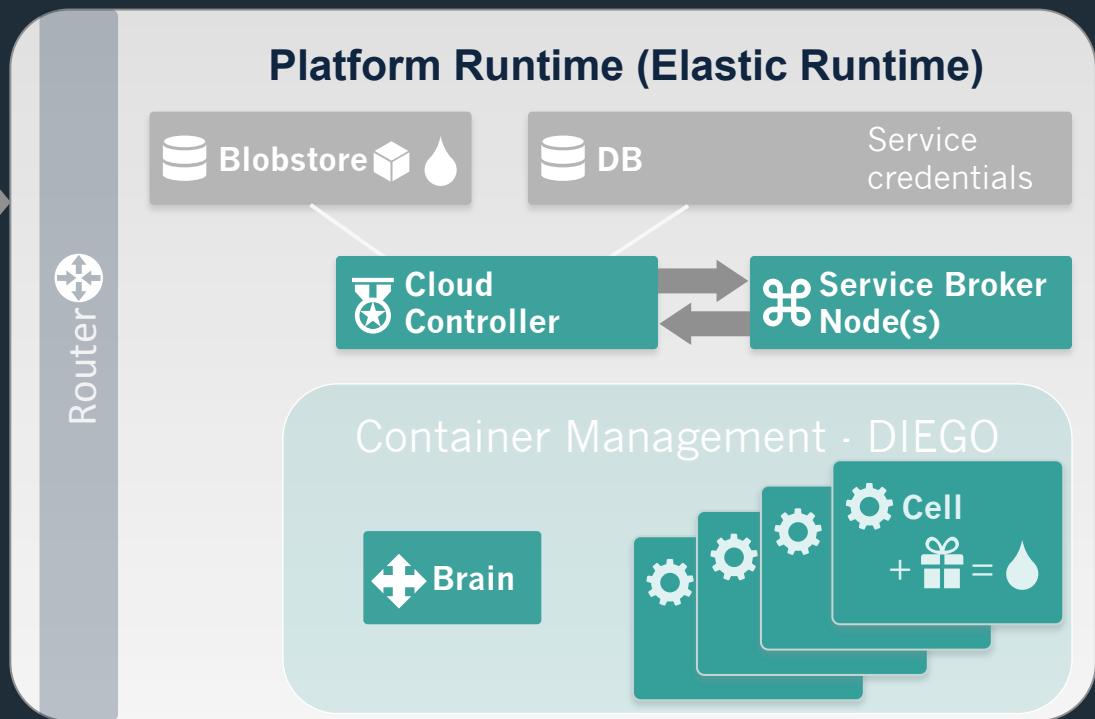




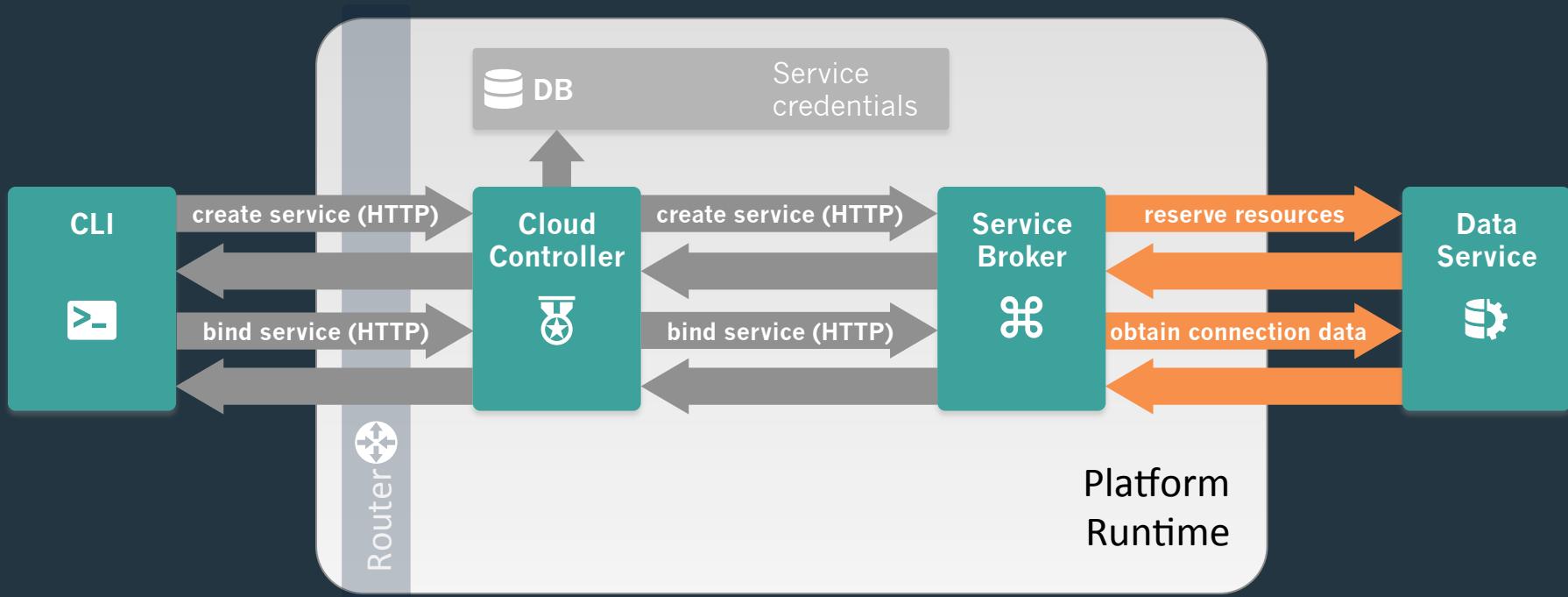
# CF PUSH, BIND AND SCALE

# Deploying an Application

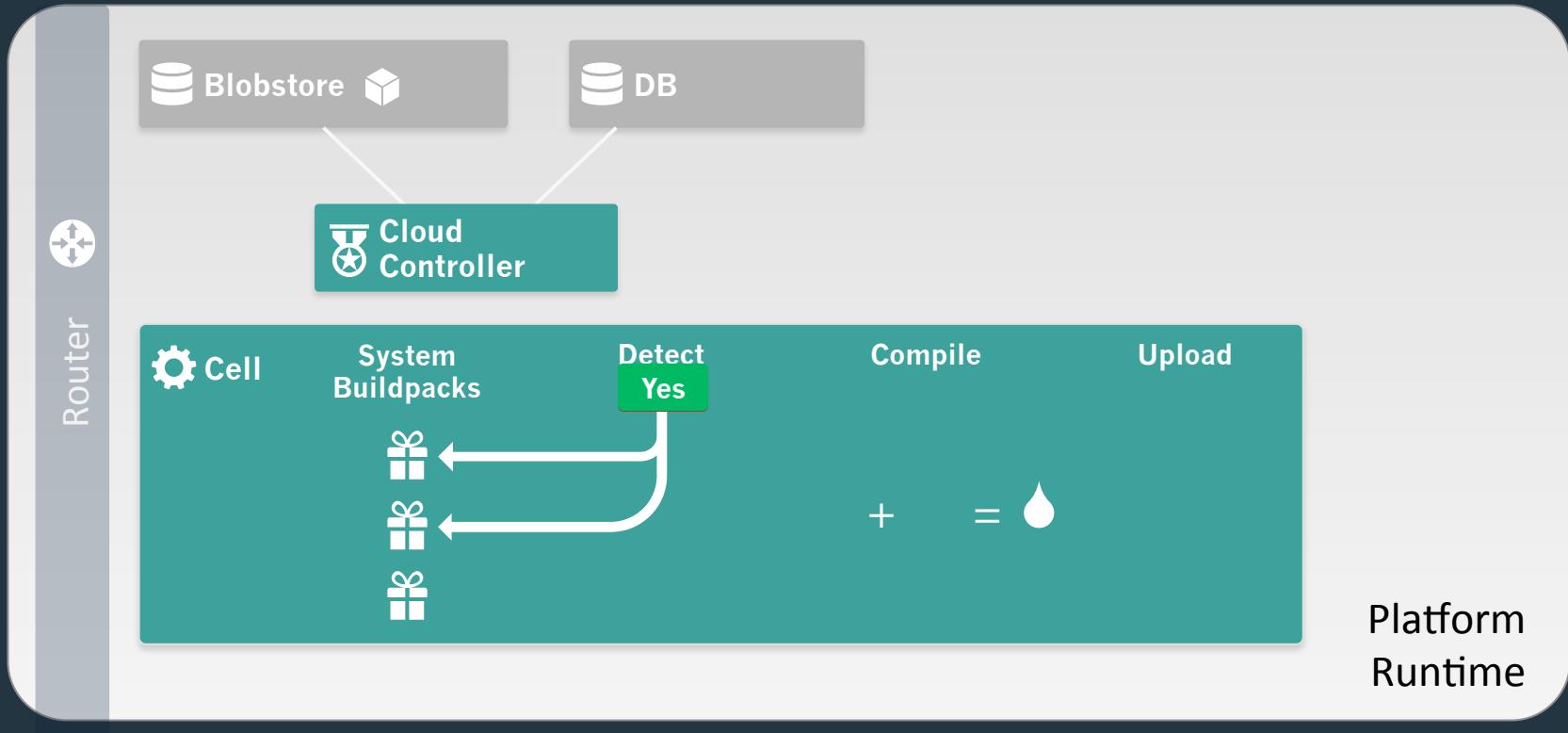
- ① Upload app bits and metadata 
- ② Create and bind services
- ③ Stage application
- ④ Deploy application
- ⑤ Manage application health



# Creating and binding services



# Staging an Application

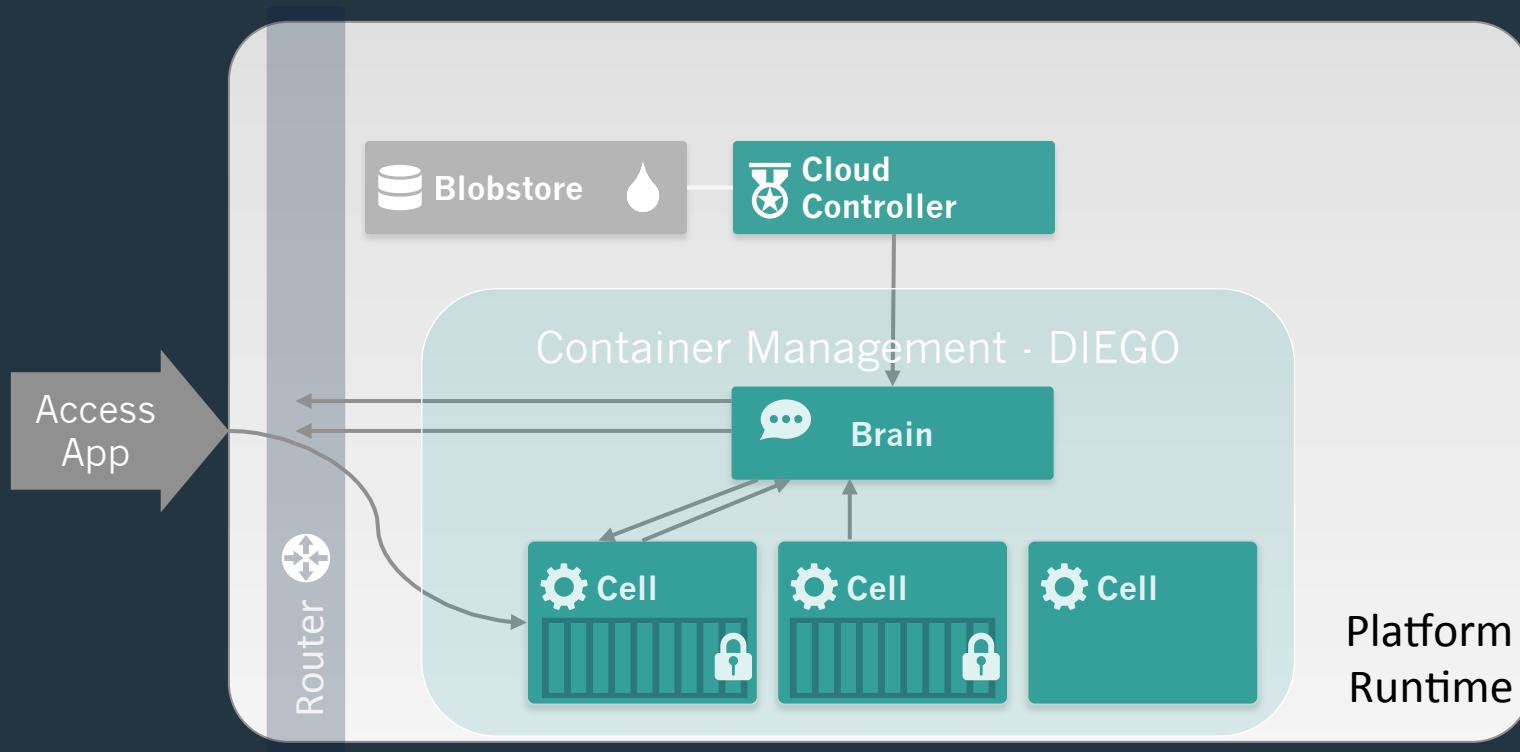


# Getting your application to scale

- Can be done via CLI:
  - At deployment time (via manifest.yml or as a modifier to cf push)
  - During run time without interrupting operations (via cf scale --instances 10)
- Can also be done via Apps Manager

The screenshot shows the Pivotal Apps Manager web interface. On the left, a sidebar navigation includes 'ORG' (mborges-org), 'SPACES' (development, production), 'Marketplace', 'Docs', 'Support', and 'Tools'. The main content area displays the 'pcf-scale-prod' application under 'mborges-org > development > pcf-scale-prod'. A large teal circle highlights the app name 'pcf-scale-prod' and its status indicators (blue square and green circle). Below the app name, it says 'last push: 12/22/15 @ 23:56 UTC' and provides the URL 'https://pcf-scale-v1\_2.south.fe.piv...'. To the right, there's a 'CONFIGURATION' section with fields for 'Instances' (1), 'Memory Limit' (128 MB), and 'Disk Limit' (1 GB). A 'Scale App' button is located at the top right of this section. Below configuration is a 'STATUS' table with columns for #, STATUS, CPU, MEMORY, DISK, and UPTIME. It shows 1 instance running with 0% CPU, 100 MB memory, 84.2 MB disk, and an uptime of 13 d 23 hr 58 min. At the bottom, tabs for Events, Services, Env Variables, Routes, and Logs are visible, along with a 'Delete App' link. A 'RECENT EVENTS' section shows a single event: 'updated app' by mborges@pivot.alio 12/24/2015 at 02:32 AM UTC.

# The App-Container Approach - Scaling



A dark, grainy photograph of a modern building's interior, possibly an airport or a large terminal. The space is dominated by large glass windows and a complex steel and glass structural framework. In the foreground and middle ground, numerous people are seen walking across a polished floor, their silhouettes reflected in the light. The overall atmosphere is one of a busy, modern public space.

**CLI, MANIFEST, API, APPS MANAGER**

# Platform Runtime - Pivotal Apps Manager

- Manage Organizations, users, applications and Spaces
- Monitor applications logs, services and routes
- Access Service Marketplace, create services and bind to applications

The screenshot shows the Pivotal Apps Manager interface for the organization 'pivotalcf-demo'. The left sidebar lists the organization and its spaces: 1 - Development, 2 - Testing, 3 - Staging, 4 - Production, and Marketplace. The main dashboard displays the organization's quota (2.63 GB of 100 GB Limit) and member count (14 Members). Below this, four spaces are listed with their respective app and service counts and percentage of org quota used.

Space	Apps	Services	Percentage of Org Quota
1 - Development	1	0	0% of Org Quota
2 - Testing	1	0	0% of Org Quota
3 - Staging	1	0	0% of Org Quota
4 - Production	2	0	1% of Org Quota

Below the spaces, there is a button to 'Add A Space'.

# Pivotal Apps Mgr. - App View

The image shows two screenshots of the Pivotal Apps Manager interface. On the left, the 'development' space overview is displayed, showing one running application ('pcf-scale-prod') and one service instance ('auto-scaler'). On the right, the detailed view for the 'pcf-scale-prod' application is shown, including configuration (1 instance, 128 MB memory), status (1 running instance), and recent events ('started app'). A large green arrow points from the left screenshot to the right one.

- Drill into a space to see all application and services instances
- Then drill into an application to see configuration, status, event, logging, routes, environment variables and service instances bound to the application

# CLI - Command Line Interface

- Command line utility providing easy access to the Pivotal CF commands.
- Scriptable
- Fully documented (cf –help)

```
-> cf
NAME:
  cf - A command line tool to interact with Cloud Foundry

USAGE:
  [environment variables] cf [global options] command [arguments...] [command options]

VERSION:
  6.13.0-e68ce0f-2015-10-15T22:53:29+00:00

BUILD TIME:
  2016-01-05 12:14:41.736991003 -0600 EST

GETTING STARTED:
  help                               Show help
  login                             Log user in
  logout                            Log user out
  passwd                           Change user password
  target                            Set or view the targeted org or space

  api                                Set or view target api url
  auth                               Authenticate user non-interactively

APPS:
  apps                             List all apps in the target space
  app                               Display health and status for app

  push                             Push a new app or sync changes to an existing app
  scale                            Change or view the instance count, disk space limit, and memory limit for an app
  delete                           Delete an app
  rename                           Rename an app

  start                            Start an app
  stop                             Stop an app
  restart                          Restart an app
  restage                          Restage an app
```

# Manifest Files

- Application manifests tell *cf push* what to do with applications
- From how many instances to create and how much memory to use.
- Help automate deployment, specially of multiple apps at once
- Can list services to be bound to the application
- YAML format – <http://yaml.org>

```
1 ---|  
2 # all applications use these settings and services  
3 domain: shared-domain.com  
4 memory: 1G  
5 instances: 1  
6 services:  
7 - clockwork-mysql  
8 applications:  
9 - name: springtock  
10 host: tock09876  
11 path: ./spring-music/build/libs/spring-music.war  
12 - name: springtick  
13 host: tick09875  
14 path: ./spring-music/build/libs/spring-music.war
```

# Cloud Controller API

- Cloud Controller component of Elastic Runtime manages all Cloud Foundry APIs
- CF CLI and other clients like Apps Manager directly call this API
- Before accessing the CC API, you must get an access token from the ERS component called User Account and Authentication (UAA) server
- <http://apidocs.cloudfoundry.org>

Cloud Foundry API

App Routes (Experimental)

- List routes
- Map a Route
- Unmap a Route

App Usage Events

- List all App Usage Events
- Purge and reseed App Usage Events
- Retrieve a Particular App Usage Event

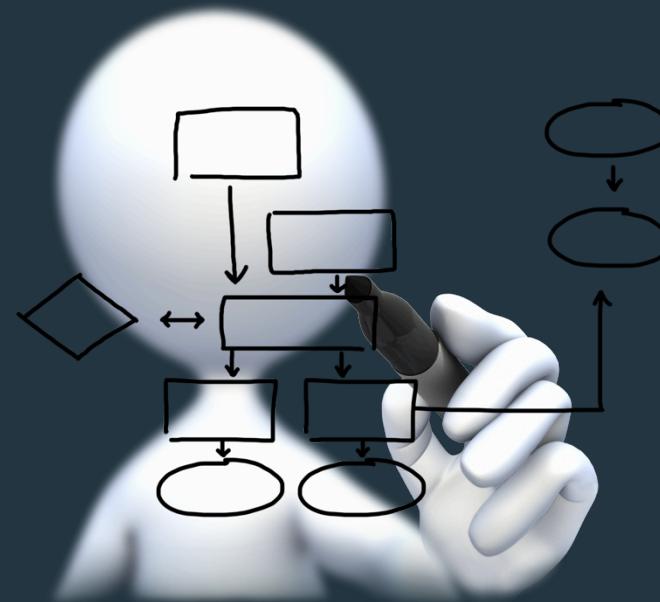
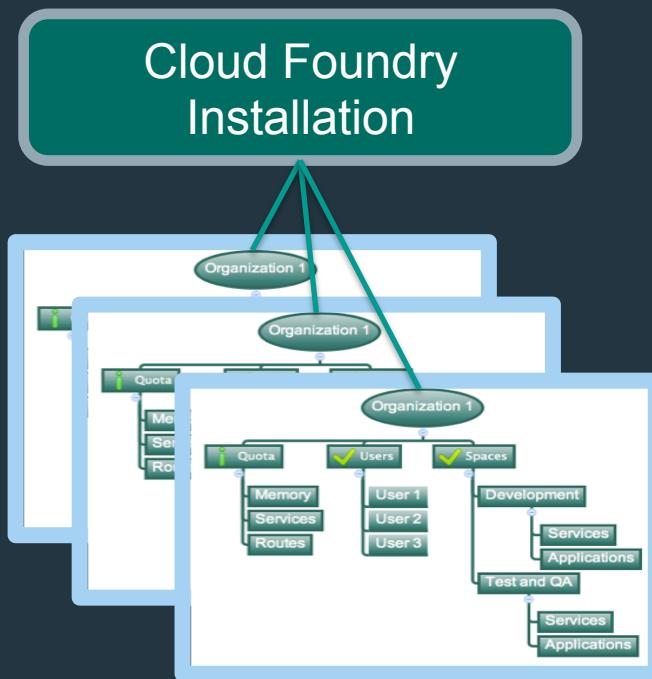
Apps

- Associate Route with the App
- Copy the app bits for an App
- Creating a Docker App (experimental)
- Creating an App
- Delete a Particular App
- Downloads the bits for an App
- Downloads the staged droplet for an App
- Get App summary
- Get detailed stats for a STARTED App
- Get the env for an App
- Get the instance information for a STARTED App
- List all Apps
- List all Routes for the App
- List all Service Bindings for the App
- Remove Route from the App
- Remove Service Binding from the App
- Restart an App
- Retrieve a Particular App
- Terminate the running App Instance at the given index
- Updating an App
- Uploads the bits for an App



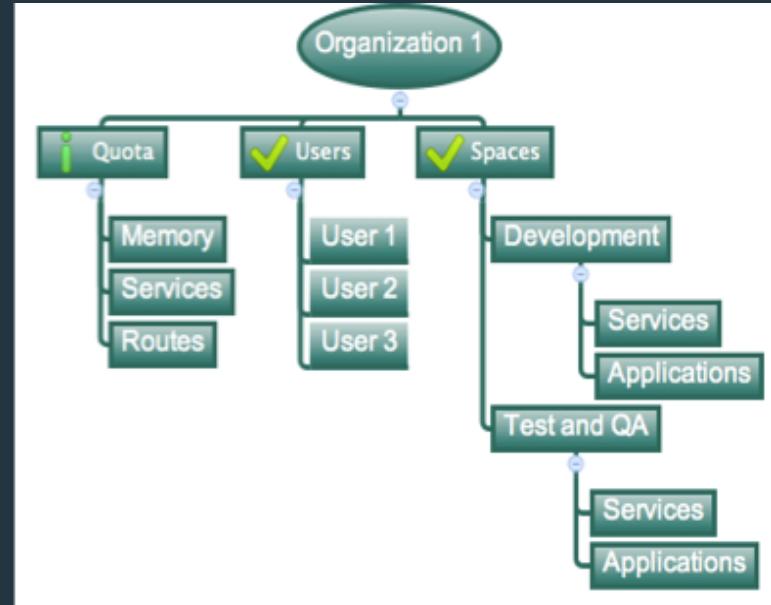
# ORGs, SPACES, USERS AND QUOTA

# Orgs, Spaces, Users and Quotas



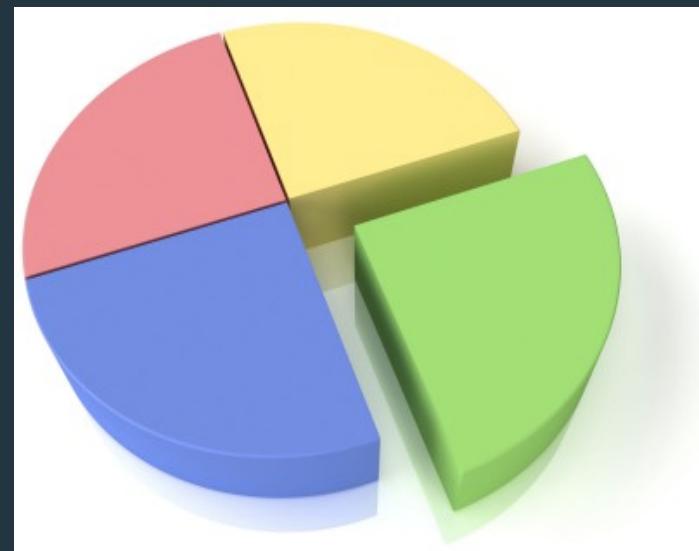
# Organizations

- Top-most administrative unit
- Logical division division within a Pivotal Cloud Foundry Install / Foundation
  - Typically a company, department, application suite or large project
- Each organization has its own users and assigned quota
- User permissions / Roles are specified per space within an organization
- Sub-divided into spaces



# Quotas and Plans

- Different quota limits (e.g. small, enterprise, default, runaway) can be assigned per Organization
- Quotas defines
  - Total Memory
  - Total # of services
  - Total # of Routes
- Sub-divided into spaces



# Spaces

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



# Users and Roles

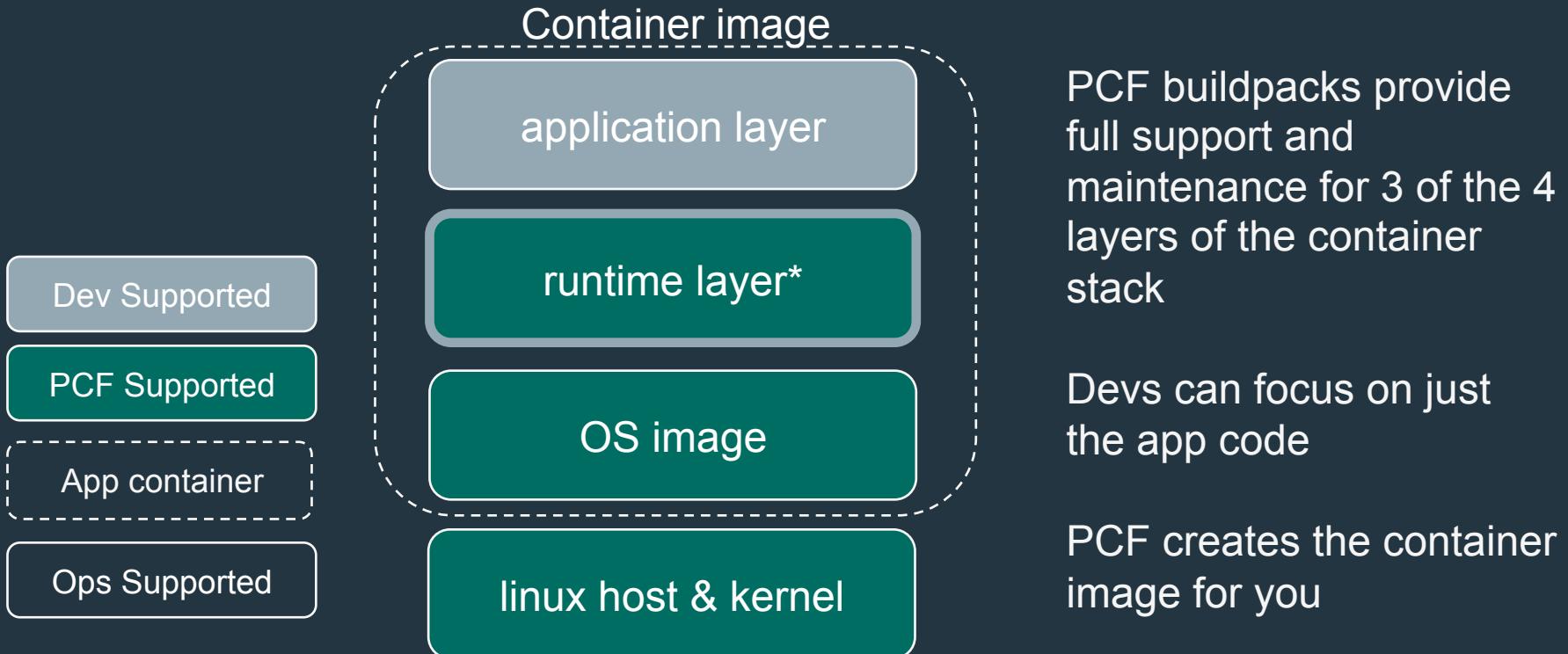


- Users are members of an organization
  - Usually they are operators or developers (not application end users)
  - Users are sent an email invite and asked to create an account
- Users have specific organization and space roles
  - Organization roles grant permissions in an organization
  - Space roles grant permissions in a particular space
  - A combination defines the user's overall permissions

A dark, atmospheric photograph showing the silhouettes of many people walking through a modern building with large glass windows. The floor reflects the light from the windows, creating a bright path for the silhouettes.

# BUILDPACKS

# Platform Runtime – Application Frameworks



\* Devs may bring a custom buildpack

# Runtime Benefits With PCF

runtime layer



## 6 Language Runtimes Maintained with Buildpacks

- Java, Node, Python, PHP, Ruby, Go

## Enterprise Support with Buildpacks or Docker Images

### Enterprise Middleware Support

- Apache Tomcat Support from #1 Committer

### Enterprise Framework Support

- Spring is the #1 Enterprise Java Framework

# Custom Buildpacks



- Cloud Foundry Community provides buildpacks for other languages
- Or write your own
  - Usually by forking / adapting an existing buildpack
- For list of CF Community Buildpacks
  - <https://github.com/cloudfoundry-community/cf-docs-contrib/wiki/Buildpacks>

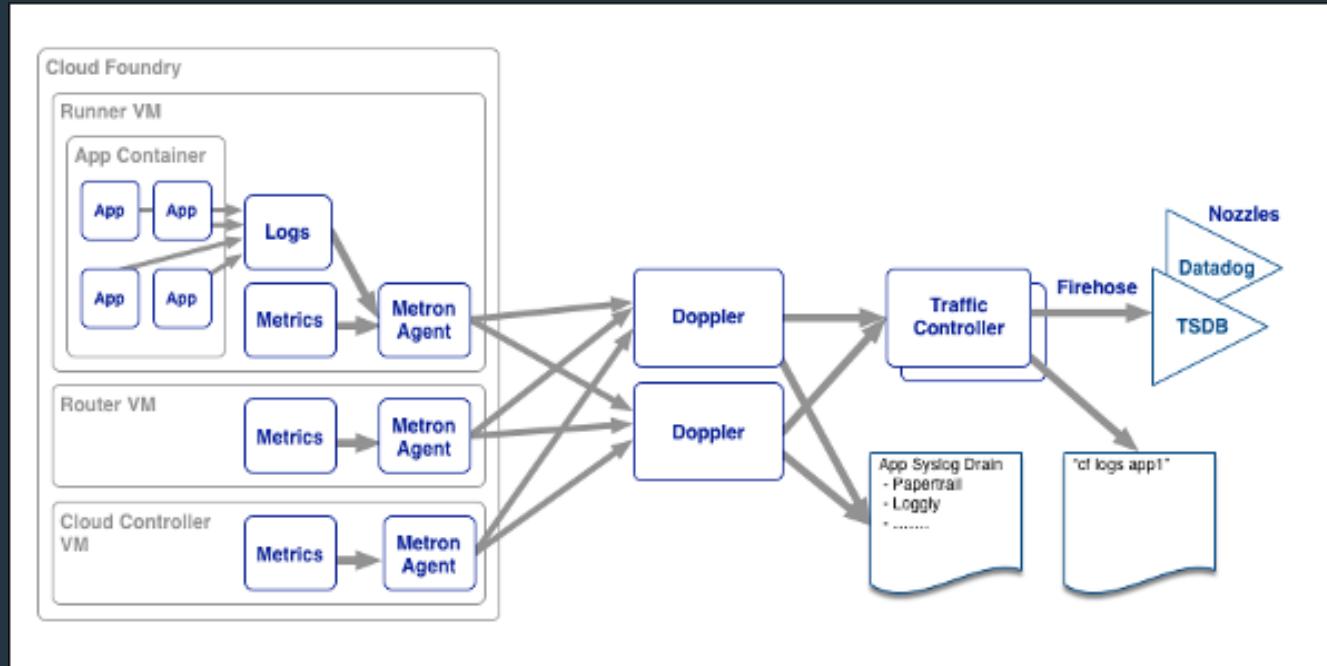


**LOGGING**

# Log Streaming and Aggregation

- **App Developers** can tail their application logs or dump the recent logs from the CF CLI, or stream these to a third party log archive and analysis services
- **Operators and administrators** can access the Loggregator firehose, the combined stream of logs from all apps, plus metrics data from CF components
- **Operators** can deploy ‘nozzles’ to the firehose. A nozzle listens to the firehose for specified events and metrics. E.g syslog or datadog nozzle

# Loggregator Components



A dark, atmospheric photograph of an airport terminal. In the foreground, several silhouetted figures of people are walking across a polished floor. A luggage cart is visible in the center. The background features large, floor-to-ceiling windows that look out onto a bright sky and some airport infrastructure.

# AUTO SCALING

# Available via the Marketplace



## App Autoscaler

Scales bound applications in response to load

# Service Management Dashboard

### spring-music

II

	INSTANCES	CPU THRESHOLDS	
min	2	low	5%
max	5	high	10%

---

### LAST EVENT

≡

Scaled app from 2 to 3 instances  
12/14/15 @ 15:49:29 UTC

---

### SCHEDULING

⌚

0 rules      Next: No Upcoming Events

### pcf demo

II

	INSTANCES	CPU THRESHOLDS	
min	2	low	20%
max	5	high	80%

---

### LAST EVENT

≡

Minimum instance limit of 2 reached  
12/14/15 @ 15:49:22 UTC

---

### SCHEDULING

⌚

2 rules      Next: 12/16/15 @ 09:00:00 UTC



# DOMAINS AND ROUTES

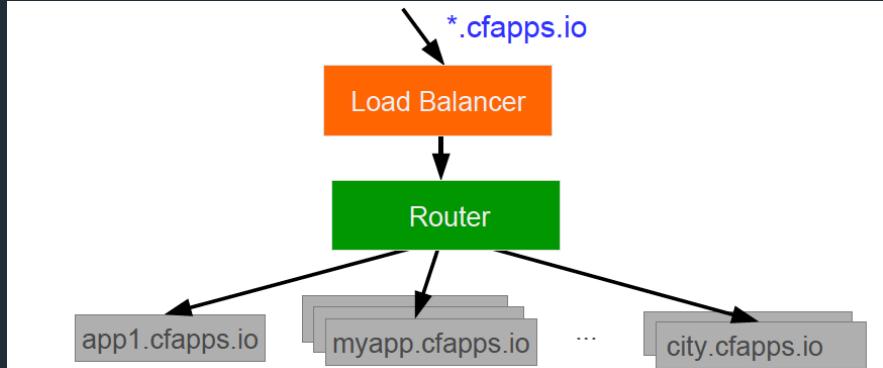
# Domains

- 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 includes sections for 'ORG', 'SPACES' (development, production, Marketplace), 'Docs', 'Support', and 'Tools'. The main content area displays organization details: 'ORG mborges-org', 'QUOTA 2%', '128 MB of 5 GB Limit', and metrics for '2 Spaces', '1 Domain', and '2 Members'. The 'DOMAINS' section contains a table with one row: 'NAME' (south.fe.pivotl.io) and 'SHARED' status. A red box highlights the 'south.fe.pivotl.io' entry.

# Domains – Behind the Scenes

- A wildcard entry (\*) is added to the DNS for the app domain
- That DNS entry points to a load balancer (or Cloud Foundry's HA Proxy), which points to the Cloud Foundry's Router
- The Router uses the subdomain to map to application instance(s)



# Routes

- HTTP requests are routed to apps pushed by associating an application with a URL, known as route
- Many apps can be mapped to a single route resulting in load balanced requests
- Routes belong to a space
- Application can have multiple routes

The screenshot shows the Pivotal Apps Manager interface for the 'mborges-org > development > pcf-scale-prod' space. The main view displays the application 'pcf-scale-prod' with its configuration (1 instance, 128 MB memory limit, 1 GB disk limit), status (0 running, 0% CPU, 88.3 MB memory, 84.2 MB disk, 1 d 2 hr 31 min uptime), and build details (Ruby, set by buildpack cflinuxfs2). The 'Routes' tab is selected, showing two mapped routes: <https://pcf-scale-prod.south.fe.pivot.io> and [https://pcf-scale-v1\\_2.south.fe.pivot.io](https://pcf-scale-v1_2.south.fe.pivot.io). A red box highlights these routes.



# BASIC SERVICES OVERVIEW

# What is a Service?

- Allows resources to be easily provisioned on-demand
- Typically middleware and other “components” necessary for applications
- Can be a persistent, stateful layer



# Two Types of Services



- **Managed** - Fully integrated, with fully lifecycle management
- **User-Provided** – Created and managed external to the platform

# Managed Services

**Managed Services** are integrated with Cloud Foundry by implementing a documented API for which the cloud controller is the client

**Service Broker** is a component which implements the required API.

- Service brokers advertise a catalog of service offerings and service plans to Cloud Foundry, and receive calls from the Cloud Controller for five functions: fetch catalog, create, bind, unbind, and delete.

# User Provided Service Instances

**User-provided service instances** are service instances which have been provisioned outside of Cloud Foundry

- Behave like other service instances once created
- Familiar commands ('create-service') provide service instance configuration (including credentials) to applications, eliminating the need to hard code service instance endpoints

EXAMPLE: AN ORACLE DATABASE MANAGED OUTSIDE OF, AND UNKNOWN TO CLOUD FOUNDRY

A dark, atmospheric photograph showing the silhouettes of many people walking through a modern building with large glass walls and a reflective floor. The scene is dimly lit, with light coming from the windows and some overhead fixtures.

# BLUE/GREEN DEPLOYMENT

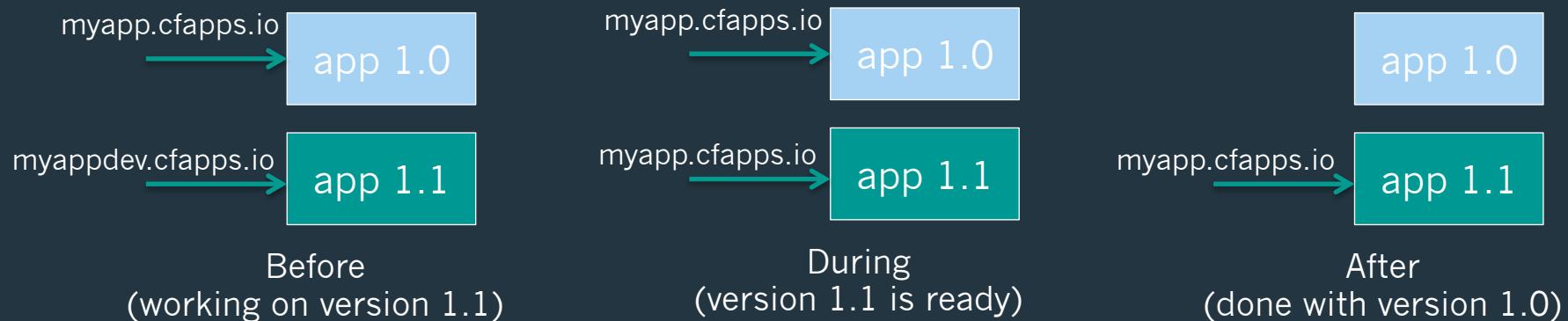
# Blue-Green Deployments

- Blue-Green deployment is an approach to upgrading application with minimal downtime.
- It also enables easy rollback to the old version if the new version experiences problems
- It works by keeping both versions online, and switching from the old version to the new version



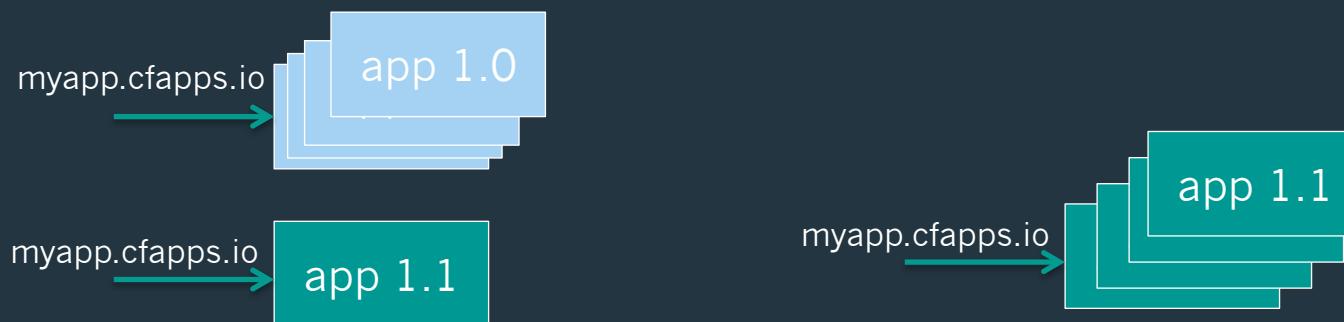
# Blue-Green Deployments using Routes

- Main idea: you always want *myapp.cfapps.io* to be good
- Use routes to achieve this
- Because you are in a single space, multiple application can use the same routes



# Slowly Turning on Traffic using Instances

- Start with four instances of blue, one instance of green
  - 80% of traffic will use version 1.0
- Decrease blue instances, increase green instances as confidence builds



# Open. Agile. Cloud-Ready.

# Pivotal.