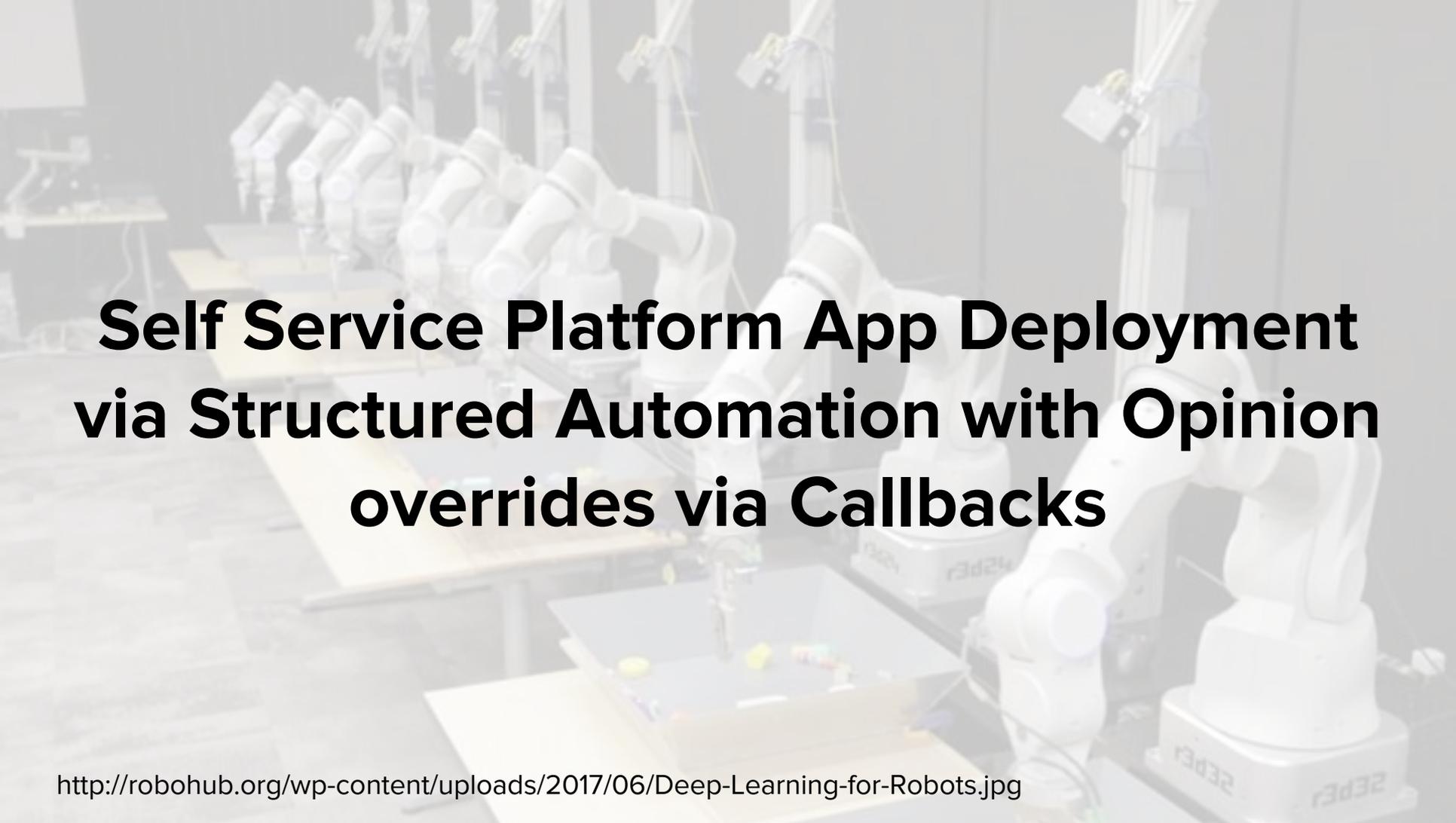


# Structure and Opinions Software Deployment with Cloud Foundry

Image credit - <http://www.rmeoc.org/wp-content/uploads/2016/07/foundry.jpg>

A row of white robotic arms, likely Universal Robots, is shown in a laboratory or industrial setting. The arms are arranged in a line, and the background is slightly blurred. The text is overlaid on the image in a large, bold, black font.

# **Self Service Platform App Deployment via Structured Automation with Opinion overrides via Callbacks**

<http://robohub.org/wp-content/uploads/2017/06/Deep-Learning-for-Robots.jpg>

# A little about Andrew Ripka



@rippmn



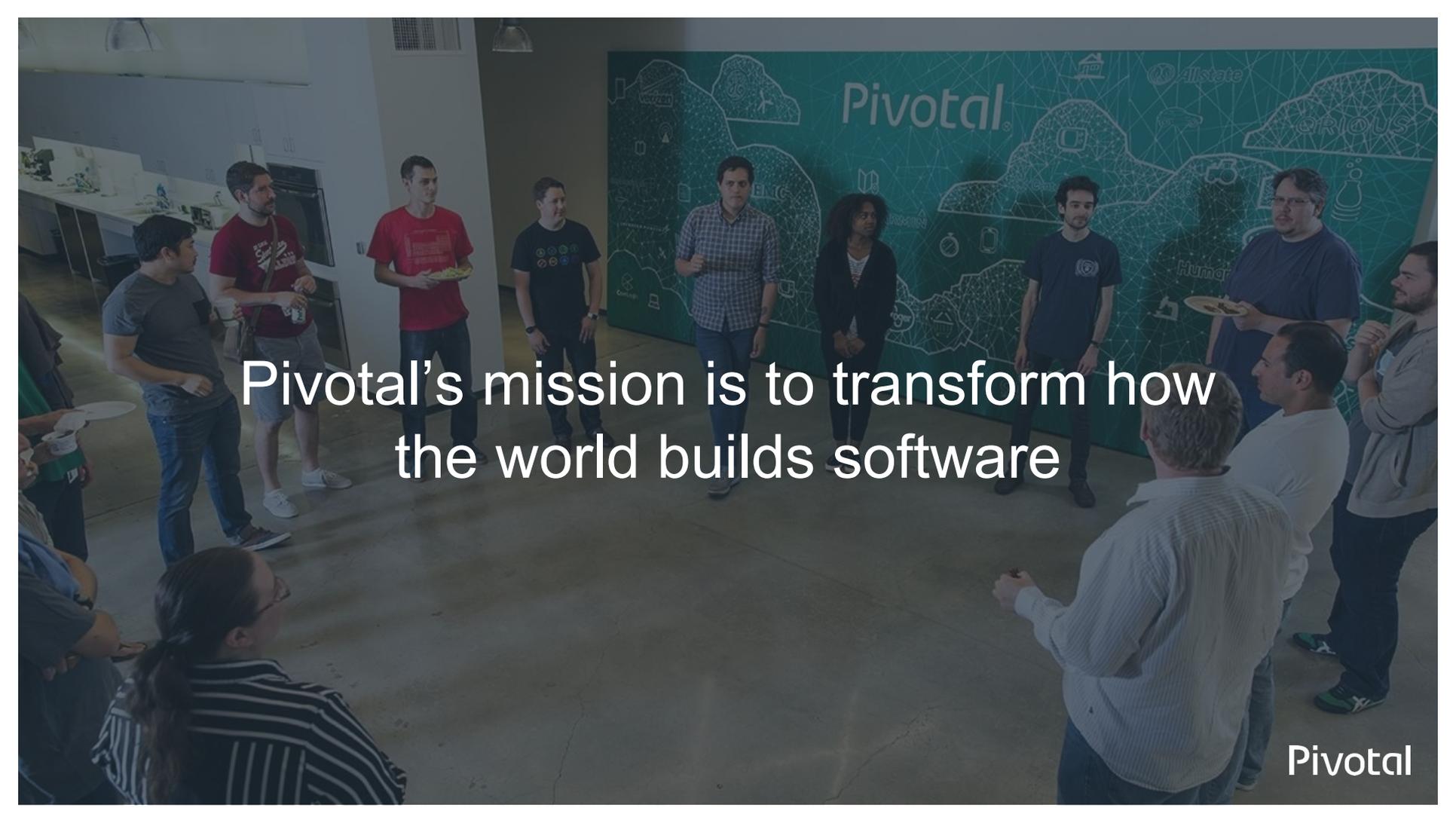
CLOUDFOUNDRY

UNITEDHEALTH GROUP\*



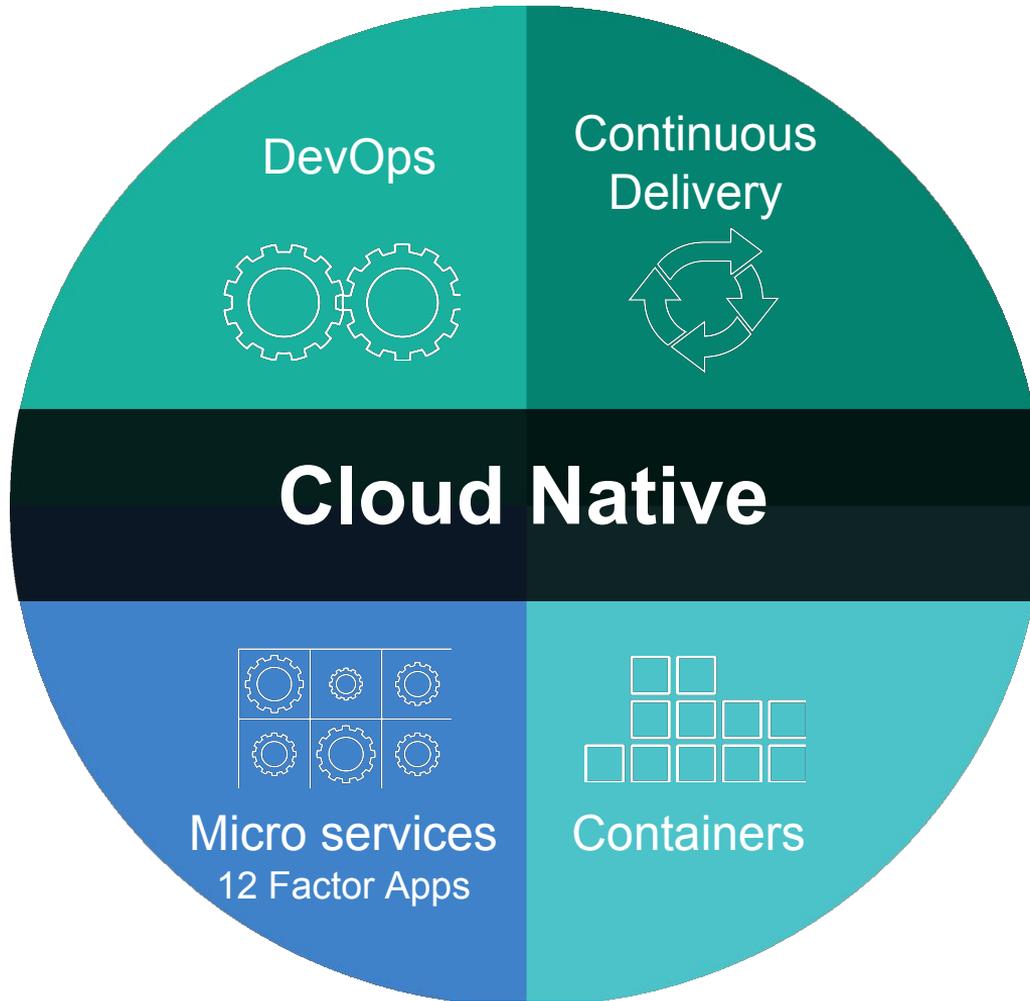
Pivotal®





Pivotal's mission is to transform how  
the world builds software

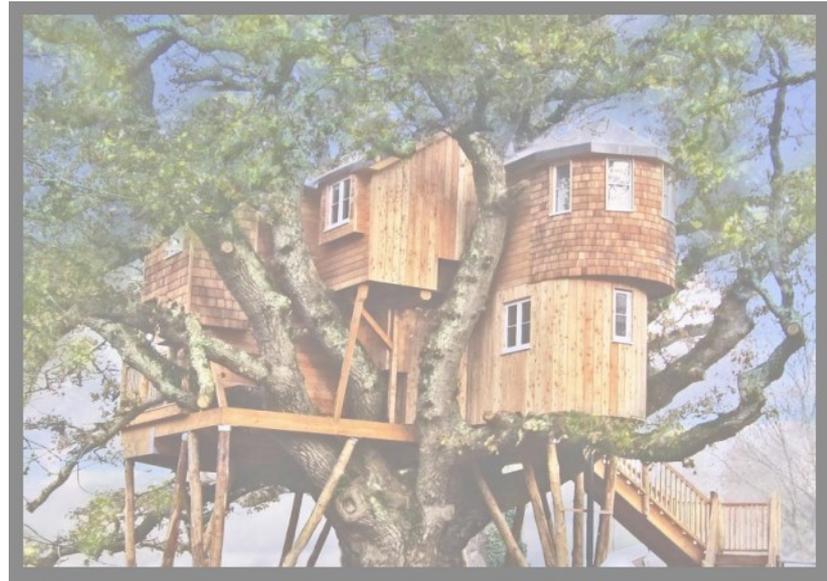
Pivotal



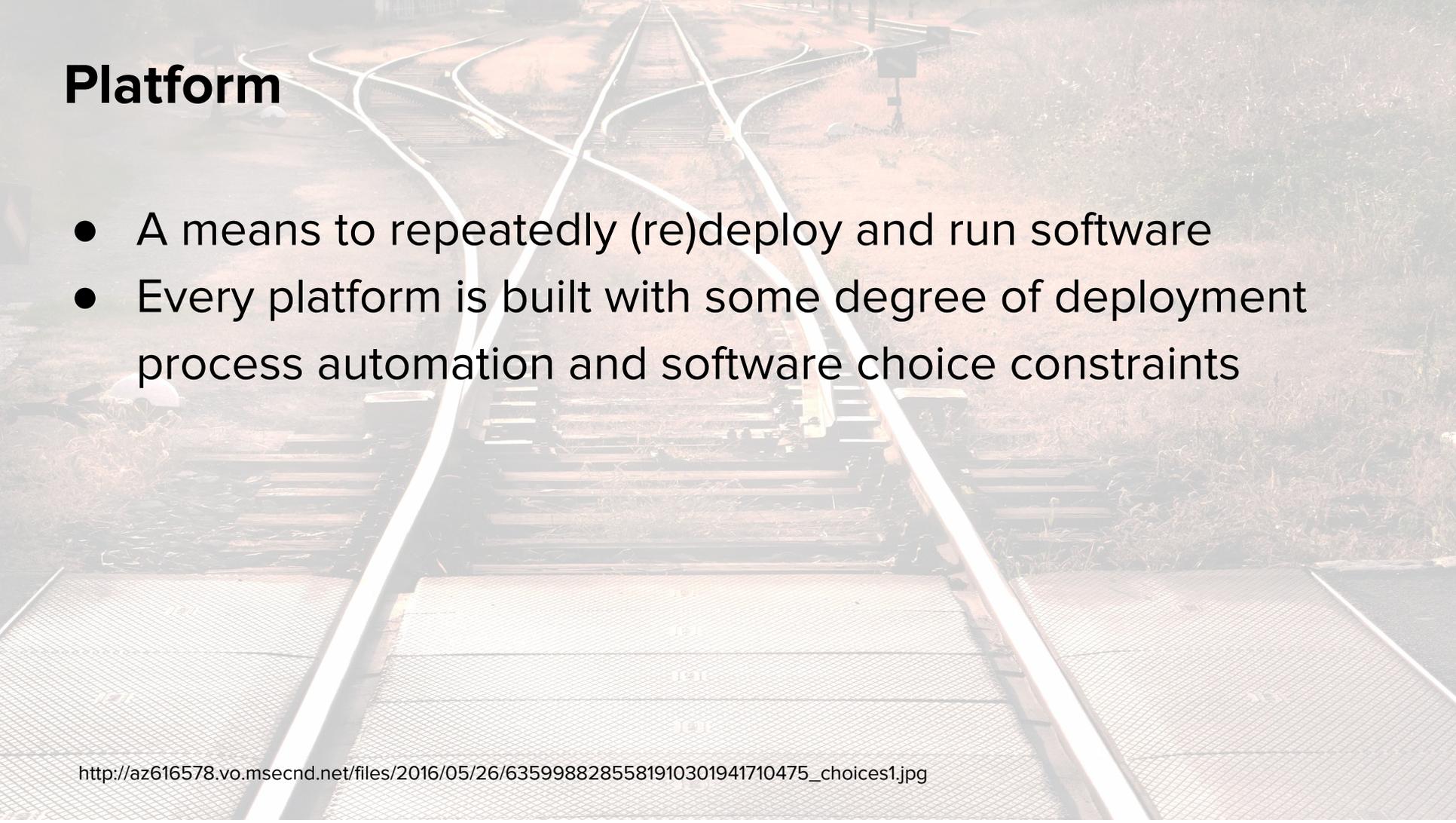
# Context Setting

**Structure** - framework for producing outcomes

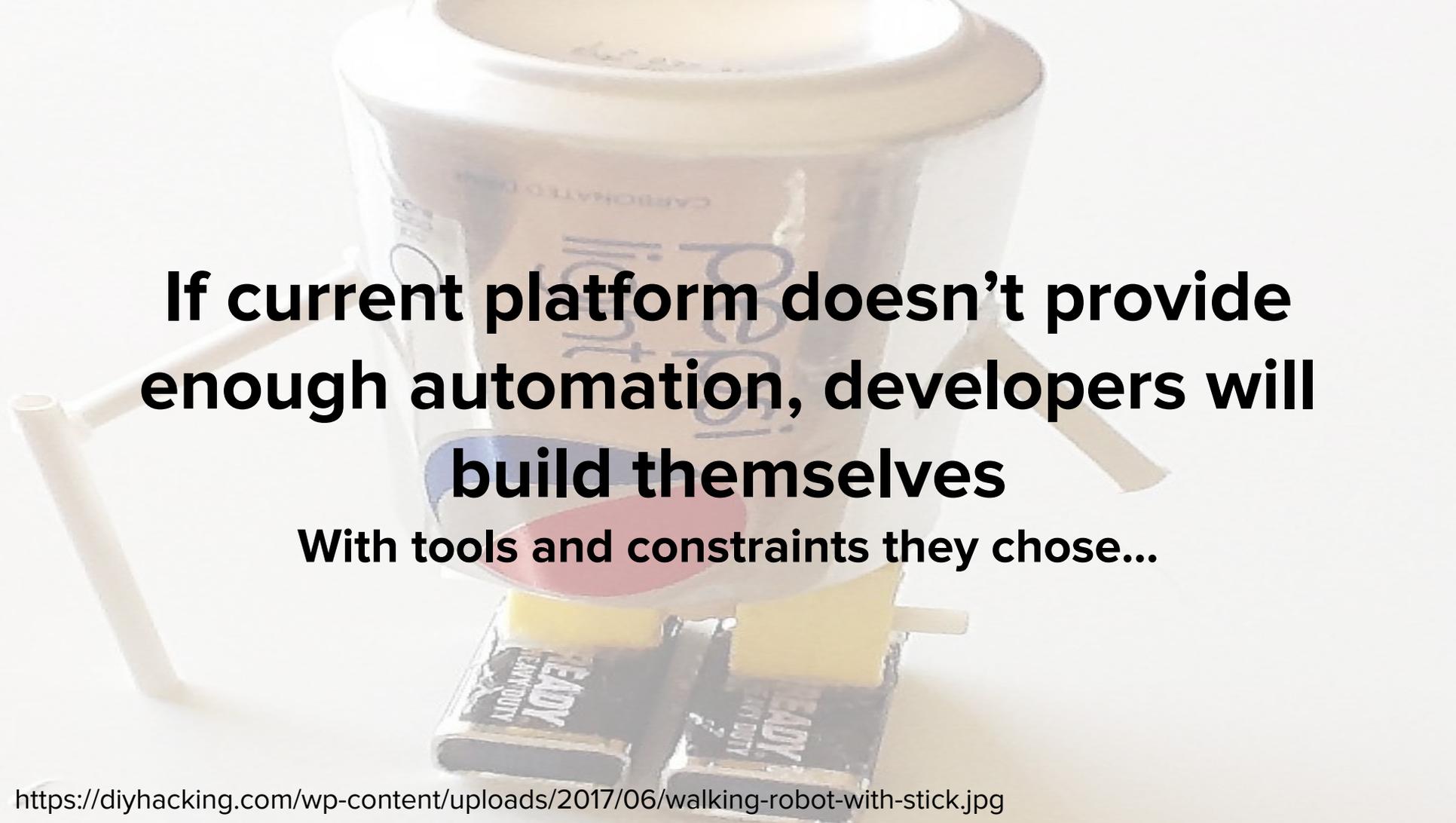
**Opinions** - specific implementations reaching desired outcomes



# Platform



- A means to repeatedly (re)deploy and run software
- Every platform is built with some degree of deployment process automation and software choice constraints



**If current platform doesn't provide  
enough automation, developers will  
build themselves**

**With tools and constraints they chose...**

visiplex

**Ad hoc automation is awesome, until  
someone who wasn't involved gets  
paged.**

**(Like the operations team)**



**A platform's overall goal should be to  
make the right thing to do the easiest  
thing to do**

**(thus providing automation with desired constraints)**



Self Service

**Enable this with a self service API to trigger the software deployment lifecycle (and other features)**

A photograph of a highway with a guardrail, overlaid with a semi-transparent grey box containing text. The guardrail is in the foreground, and the road curves into the distance. There are signs on the side of the road. The text is centered in the grey box.

**And ensure provided constraints are applied**

A photograph showing construction workers in high-visibility vests and hard hats installing a metal beam guardrail on a road shoulder. The guardrail is being placed on concrete blocks. In the background, there are several pickup trucks and a large utility pole. The scene is outdoors on a sunny day.

**System should allow for constraints to be added to or changed**

# Providing a Platform

- Provide a framework that defines deployment lifecycle structured with API callbacks
- Use the framework to implement existing deployment models with it as base opinions
- Provide a means for devs to use lifecycle to sandbox new deployment opinions for reuse by others (once operationalized)



CLOUDFOUNDRY

# Foundation Driven

CLOUDFOUNDRY  
FOUNDATION

We see a **world of computing** that is...

## Ubiquitous and Flexible

Supporting public, private, and hybrid application environments

## Portable and Interoperable

Enabling users to move their applications wherever they need to go

## Vibrant and Growing

Underlying a massive ecosystem of applications and developers based on an efficient marketplace

<https://www.cloudfoundry.org/foundation>

# CLOUD FOUNDRY

The logo for Cloud Foundry, featuring a blue lightbulb with a blue liquid-like shape inside, dripping down into a grey gear.

Cloud Foundry gives companies the **speed, simplicity and control** they need to develop and deploy applications faster and easier.

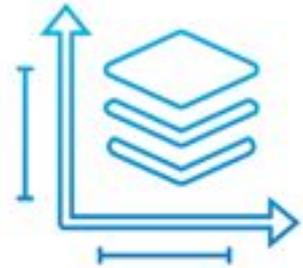
# CLOUD **FO**UNDRY



**Open Source**



**Faster to Iterate**



**Scalable Platform**

# CLOUD FOUNDRY

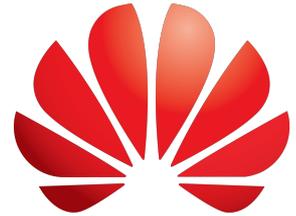
- Platform As a Service
  - Polyglot Runtime supporting multiple languages and frameworks
  - Choice of IaaS Providers
  - Choice of Application Services



CLOUDFOUNDRY

# Certified Distros

**Atos**



**HUAWEI**



**swisscom**

**IBM**

**FUJITSU**



**Pivotal  
Cloud Foundry®**

<https://cloudfoundry.org/certified-platforms/>

# CLOUD FOUNDRY

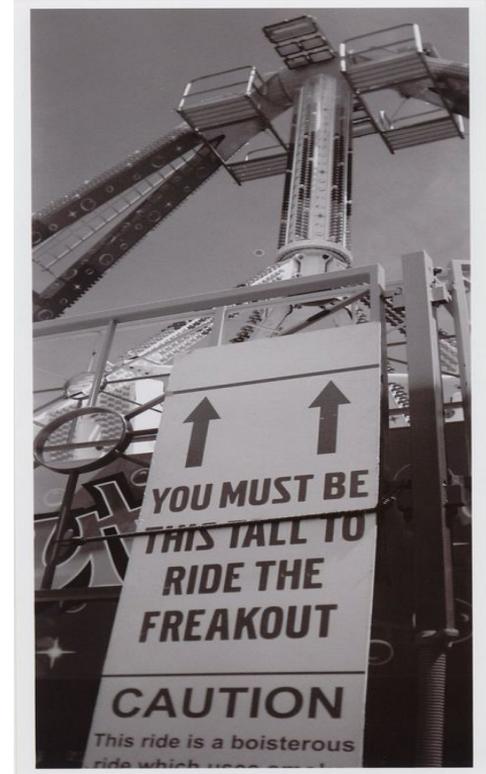


CLOUD FOUNDRY



# Essential Elements of Enterprise Platform

- Rapid Provisioning
- Rapid Application Deployment & Management
- Scale Horizontally
- Basic Monitoring, Logs & Metrics
- No-downtime platform upgrades
- Dynamic load balancing and routing
- Manage supporting services (data, messaging)
- Failure detection and health remediation



# The developer haiku



<https://www.youtube.com/watch?v=SSxI9eonBVs>

# The operator haiku



`bosh deploy`  
(operator)

here are my servers  
go make them a cloud foundry  
I do not care how

2:37 / 26:56

# Culture



Dev



Dev



IT Ops

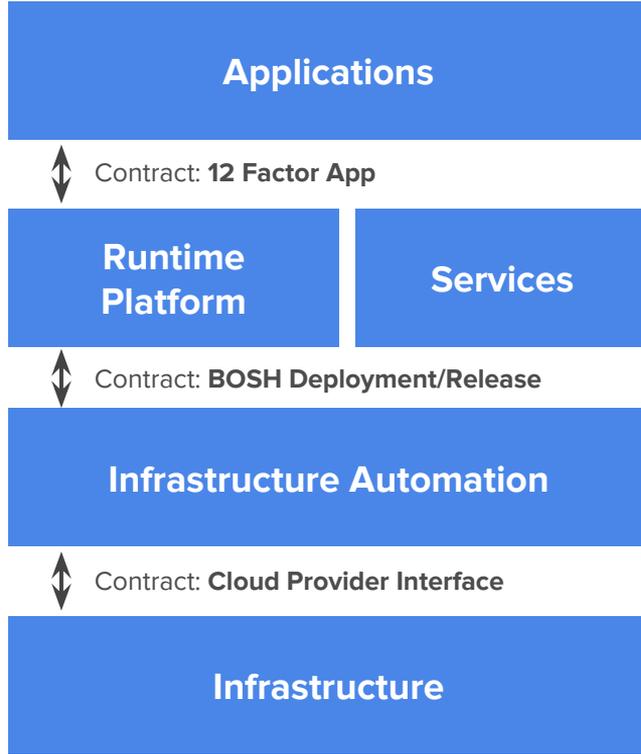


IT Ops

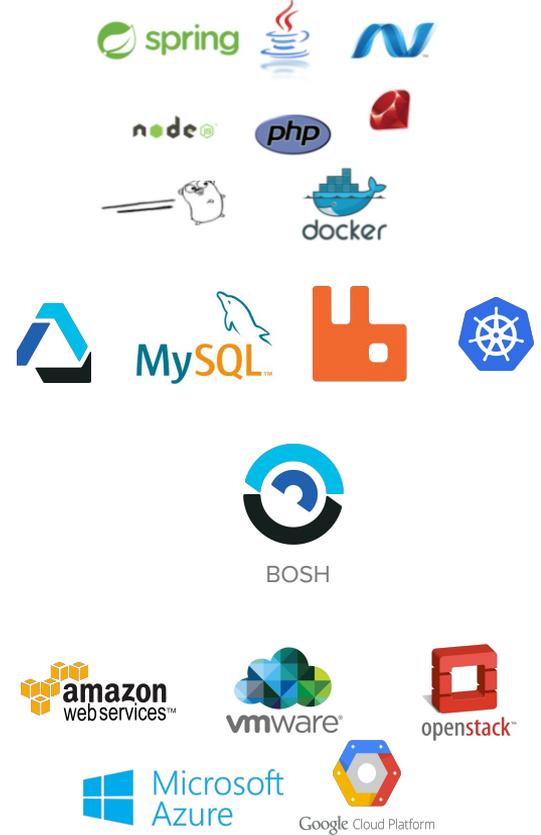


IT Ops

# Cloud Native Platform



# Tools



# Culture



Dev



Dev



IT Ops

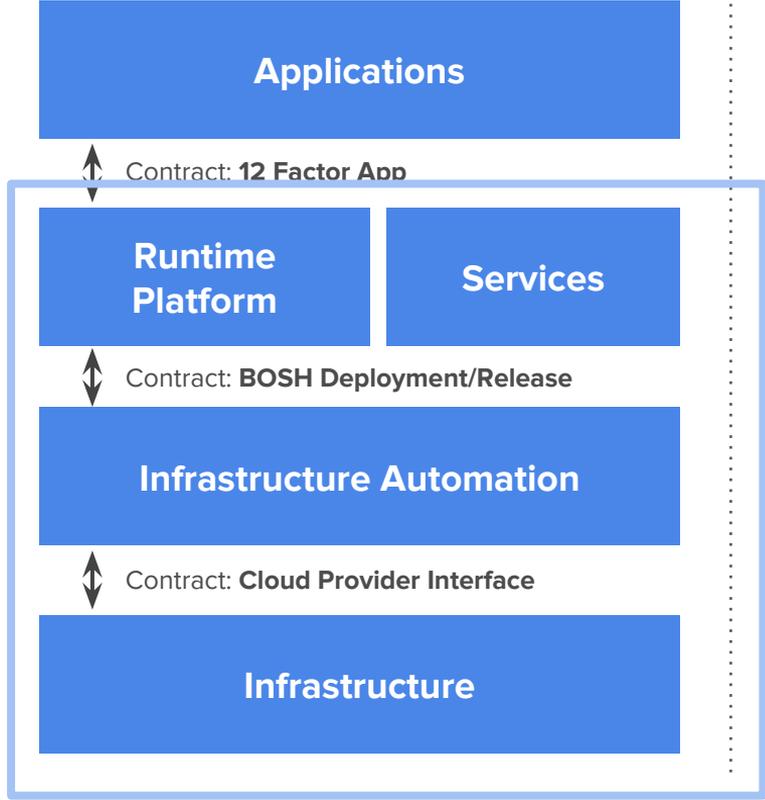


IT Ops

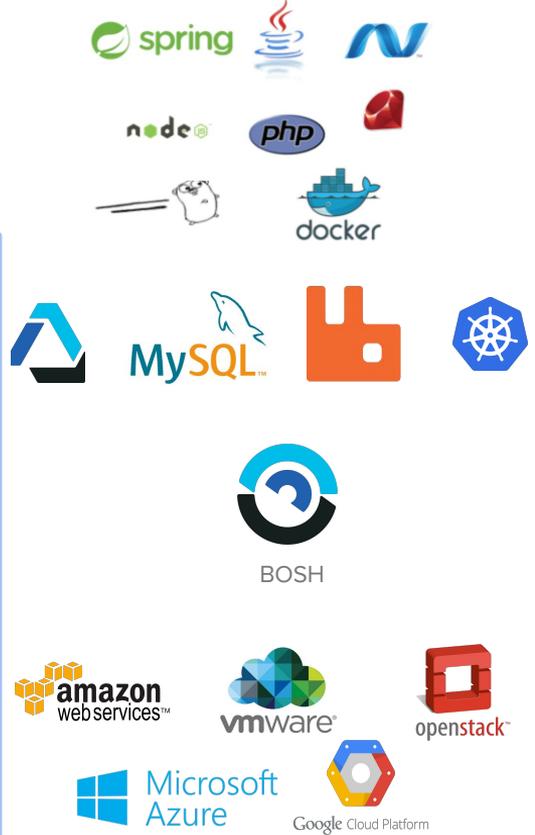


IT Ops

# Cloud Native Platform



# Tools





CLOUDFOUNDRY

# The Structures (and Default Opinions)

# Culture



Dev



Dev



IT Ops

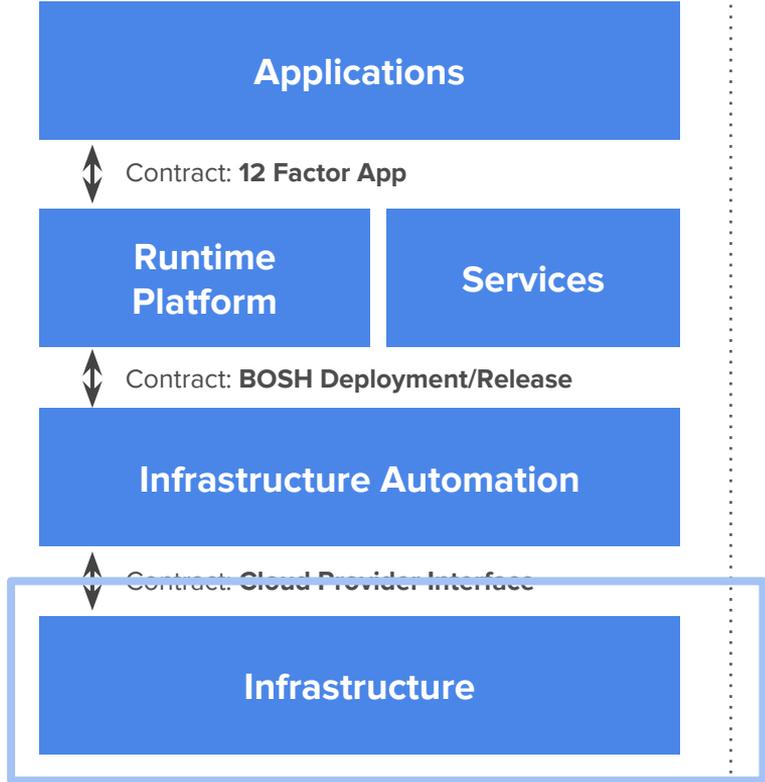


IT Ops

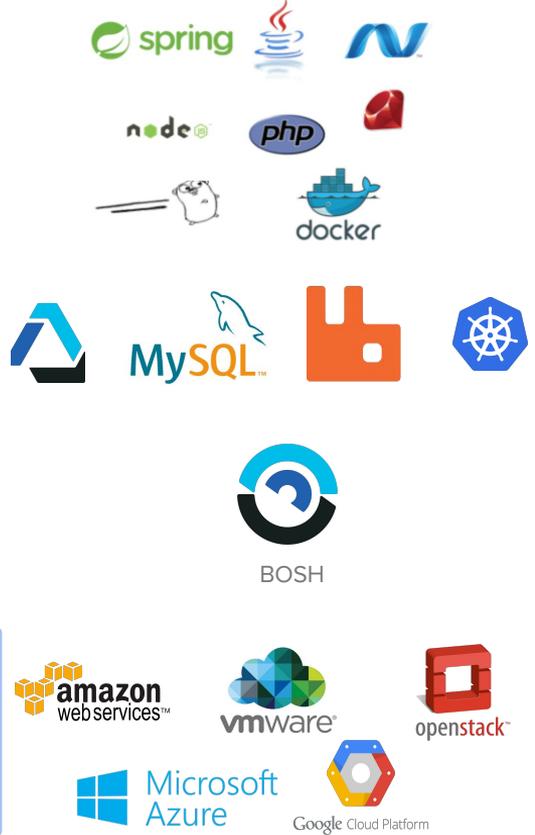


IT Ops

# Cloud Native Platform



# Tools



# Infrastructure (as a Service)

Infrastructure Resource Pool  
that can be provisioned via  
API calls



# Culture



Dev



Dev



IT Ops

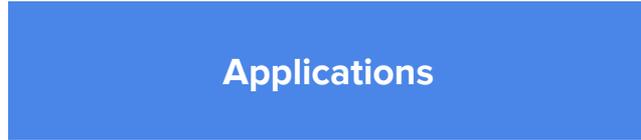


IT Ops



IT Ops

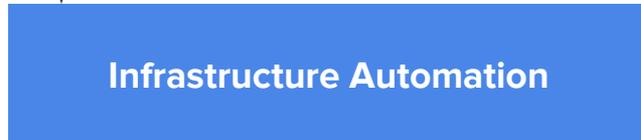
# Cloud Native Platform



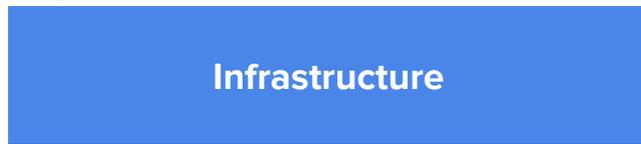
Contract: 12 Factor App



Contract: BOSH Deployment/Release



Contract: Cloud Provider Interface



# Tools





CLOUDFOUNDRY

# Infrastructure Automation



# What is BOSH?

- BOSH is an open source tool chain for release engineering, deployment, and lifecycle management of large-scale distributed services.
- BOSH was specifically developed to facilitate the deployment of Cloud Foundry
- BOSH Interacts with IaaS via Cloud Provider Interface (CPI)

# Cloud Provider Interface

## Stemcell

- `create_stemcell(image, cloud_properties)`
- `delete_stemcell(stemcell_id)`

## VM

- `create_vm(agent_id, stemcell_id, resource_pool, networks, disk_locality, env)`
- `delete_vm(vm_id)`
- `reboot_vm(vm_id)`
- `has_vm(vm_id)`
- `set_vm_metadata(vn_id, metadata)`
- `configure_networks(vm_id, networks)`



# Cloud Provider Interface

## Disk

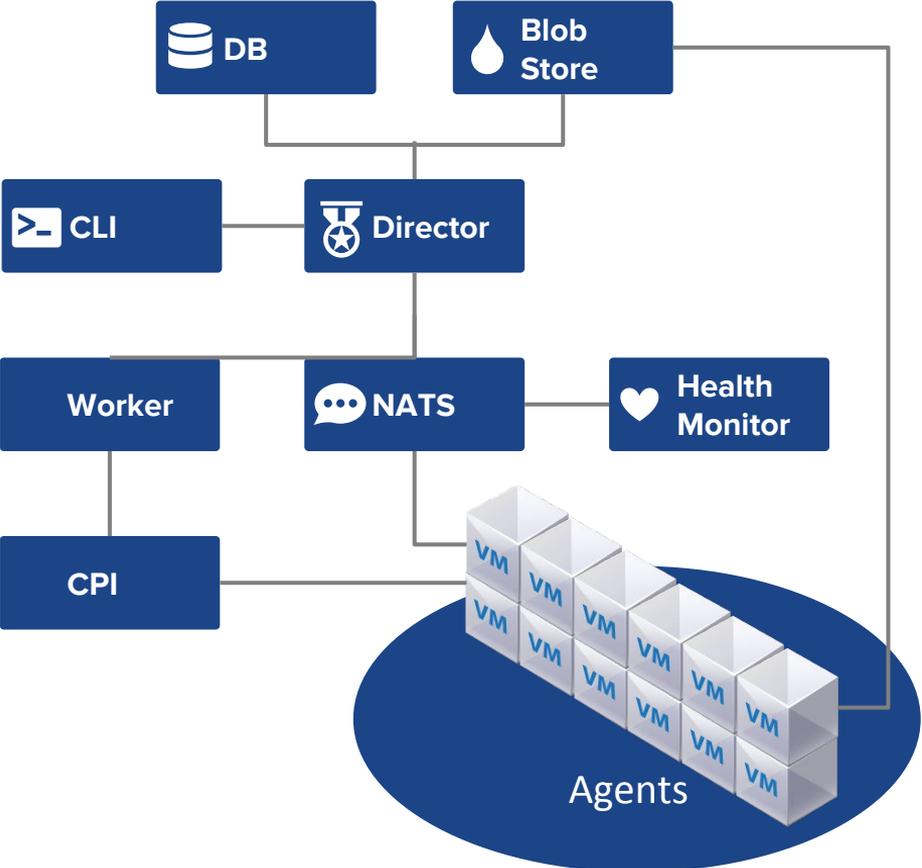
- `create_disk(size, vm_locality)`
- `delete_disk(disk_id)`
- `attach_disk(vm_id, disk_id)`
- `detach_disk(vm_id, disk_id)`
- `has_disk(disk_id)`
- `get_disks(disk_id, metadata)`

## Disk Snapshots

- `snapshot_disk(disk_id, metadata)`
- `delete_snapshot (disk_id)`
- `current_vm_id(vm_id)`

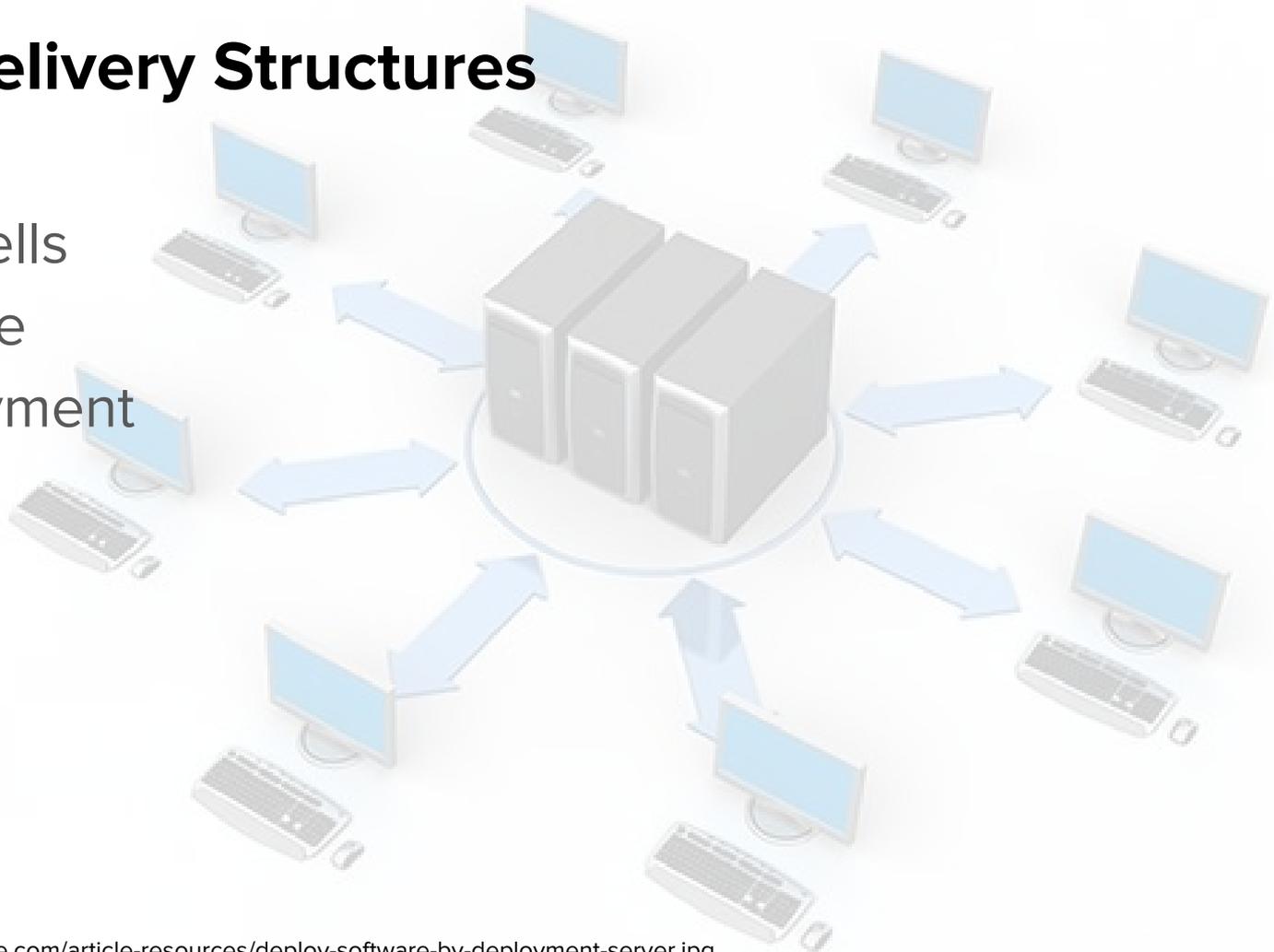


# BOSH Architecture



# BOSH Delivery Structures

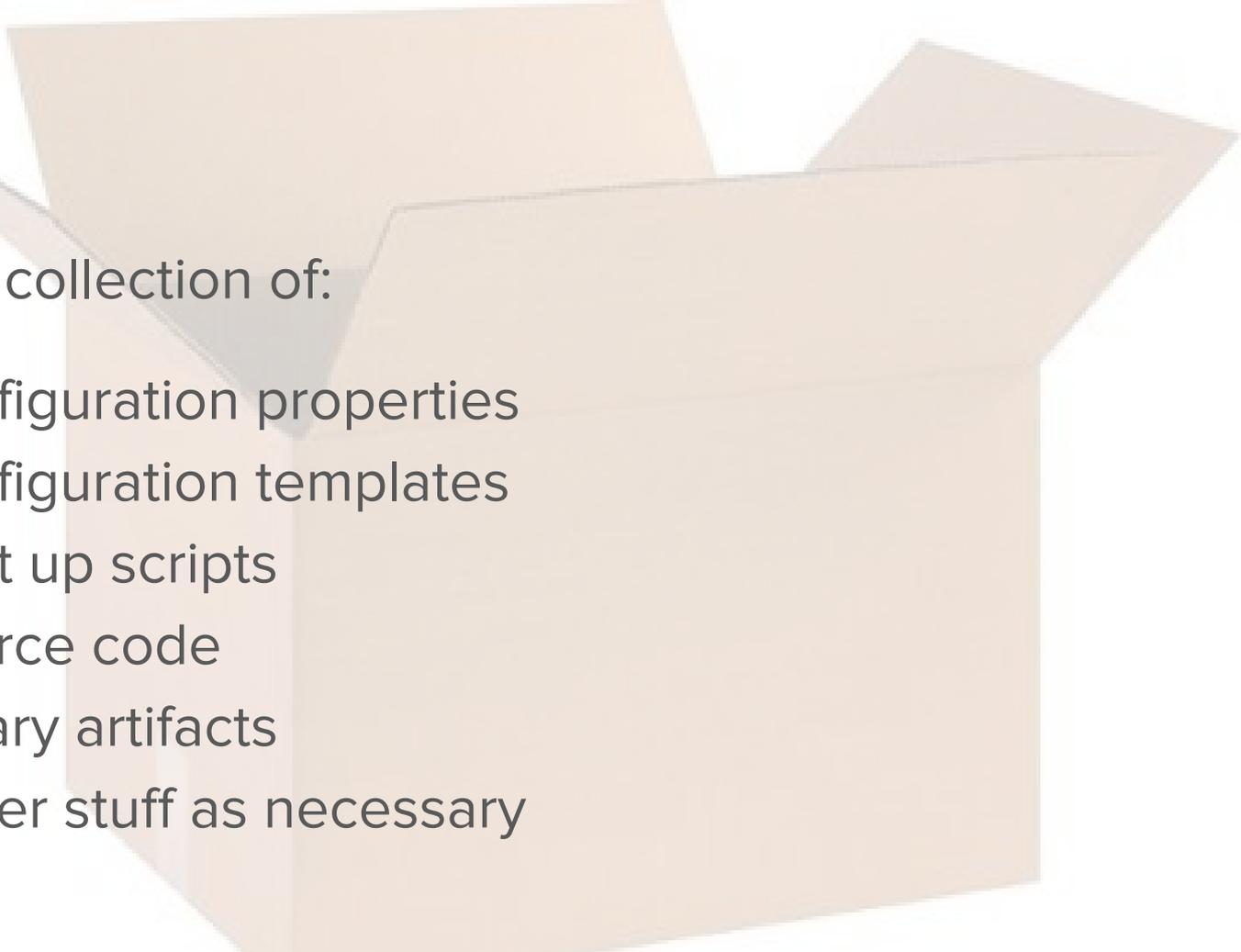
- Stemcells
- Release
- Deployment



# Stemcell

- Base OS System
- BOSH Agent
- Same image for all Infrastructures
- No specific info about software to be installed
- Wrapped in IaaS packaging that allows fast cloning
  - vSphere - VMDK
  - AWS – AMI
  - Google Compute and Azure - Image

# Release



Versioned collection of:

- configuration properties
- configuration templates
- start up scripts
- source code
- binary artifacts
- Other stuff as necessary

# Deployment

- Collection of VMs (1 or more)
- Built from stemcells
- Populated with Releases and Persistent Disk

# Deployment Example

```
---
name: my-redis-deployment
director_uuid: 1234abcd-5678-efab-9012-3456cdef7890

releases:
- {name: redis, version: 12}

resource_pools:
- name: redis-servers
  network: default
  stemcell:
    name: bosh-aws-xen-ubuntu-trusty-go_agent
    version: 2708
  cloud_properties:
    instance_type: m1.small
    availability_zone: us-east-1c

networks:
- name: default
  type: manual
  subnets:
  - range: 10.10.0.0/24
    gateway: 10.10.0.1
    static:
    - 10.10.0.16 - 10.10.0.18
```

Full Example - <https://bosh.io/docs/sample-manifest.html>

# Culture



Dev



Dev



IT Ops

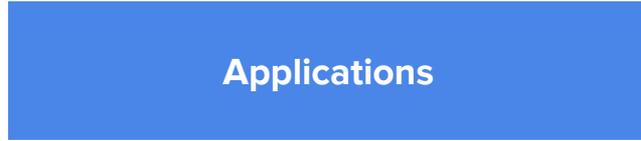


IT Ops



IT Ops

# Cloud Native Platform



Contract: **12 Factor App**



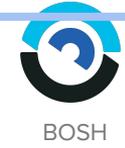
Contract: **BOSH Deployment/Release**



Contract: **Cloud Provider Interface**



# Tools

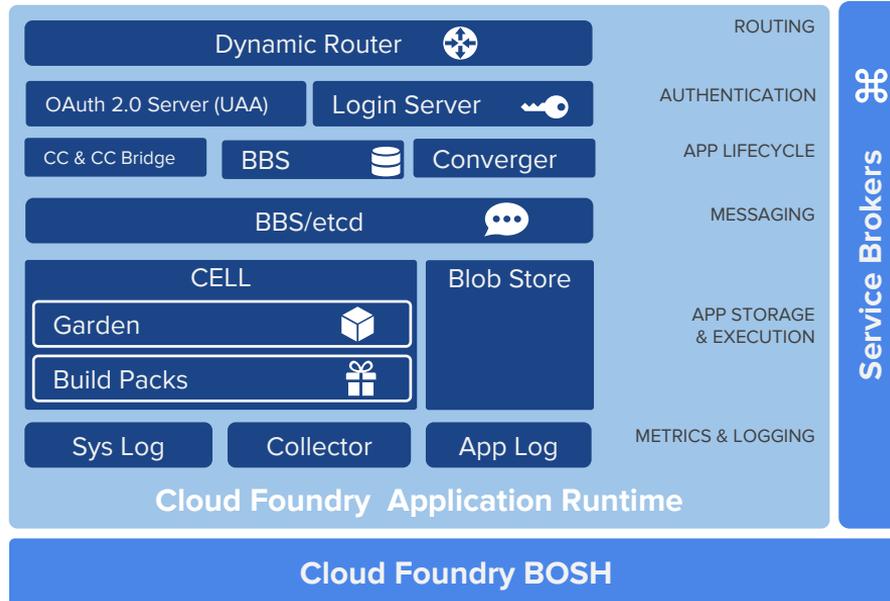




CLOUD **FOUNDRY**

**APPLICATION**  
**RUNTIME™**

# Application Runtime Architecture



Microsoft Azure



Google Cloud Platform



openstack

vmware

# Self Service API - Cloud Controller

- Fine grained REST API to the platform app lifecycle
- Coarse grained CLI to call REST API

```
>cf push
```

<https://apidocs.cloudfoundry.org/268/>

Home

Version 268 - CC API VERSION 2.89.0 ▾

Version 3

## Cloud Foundry API

### App Usage Events

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

### Apps

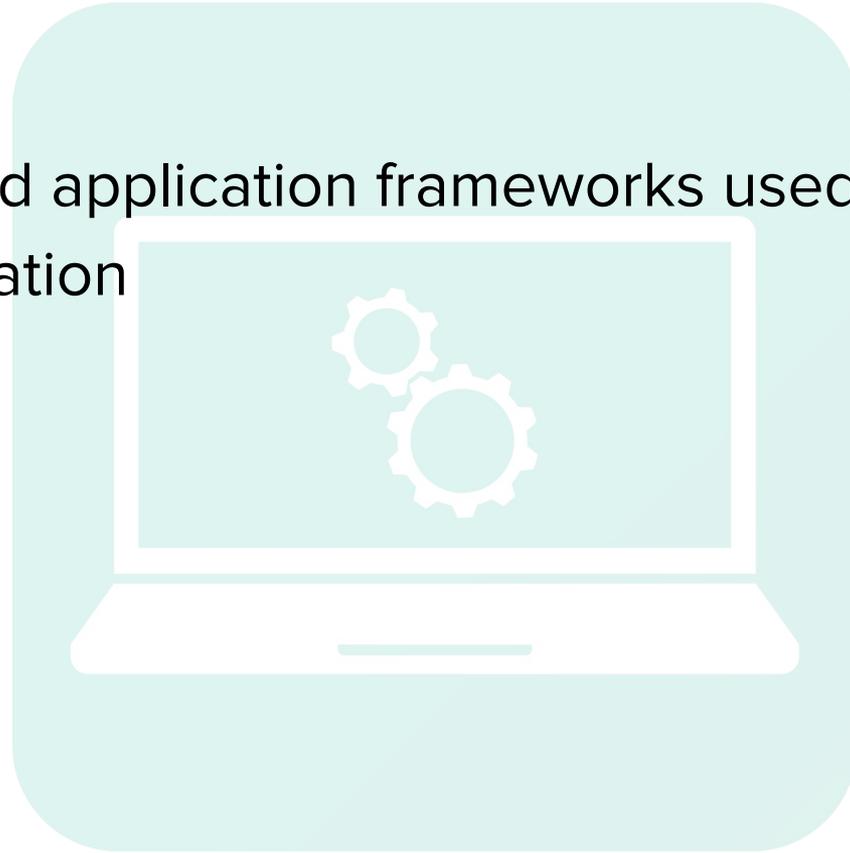
- Associate Route with the App
- Copy the app bits for an App
- Creating a Docker App
- Creating an App
- Delete a Particular App
- Downloads the bits for an App
- Downloads the staged droplet for an App
- Get App summary
- Get detailed stats for a STARTED App
- Get the env for an App
- Get the instance 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
- Restage an App
- Retrieve a Particular App
- Terminate the running App Instance at the given index
- Updating an App
- Uploads the bits for an App



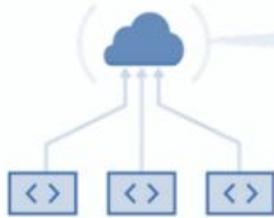
# **Application Deployment**

# Application

Source code and application frameworks used by developers to create application

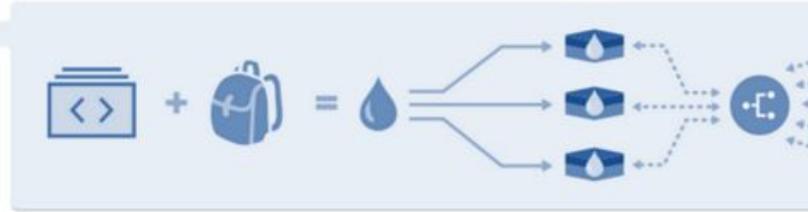


# Automated App Deployment



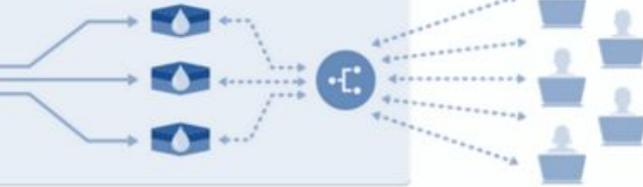
## Step 1 - Upload

Upload application files via the Platform Controller API



## Step 2 - Stage

Take application files and package with other dependencies into the container file



## Step 3 - Distribute

Platform controller schedules (assigns) the container to cluster cell

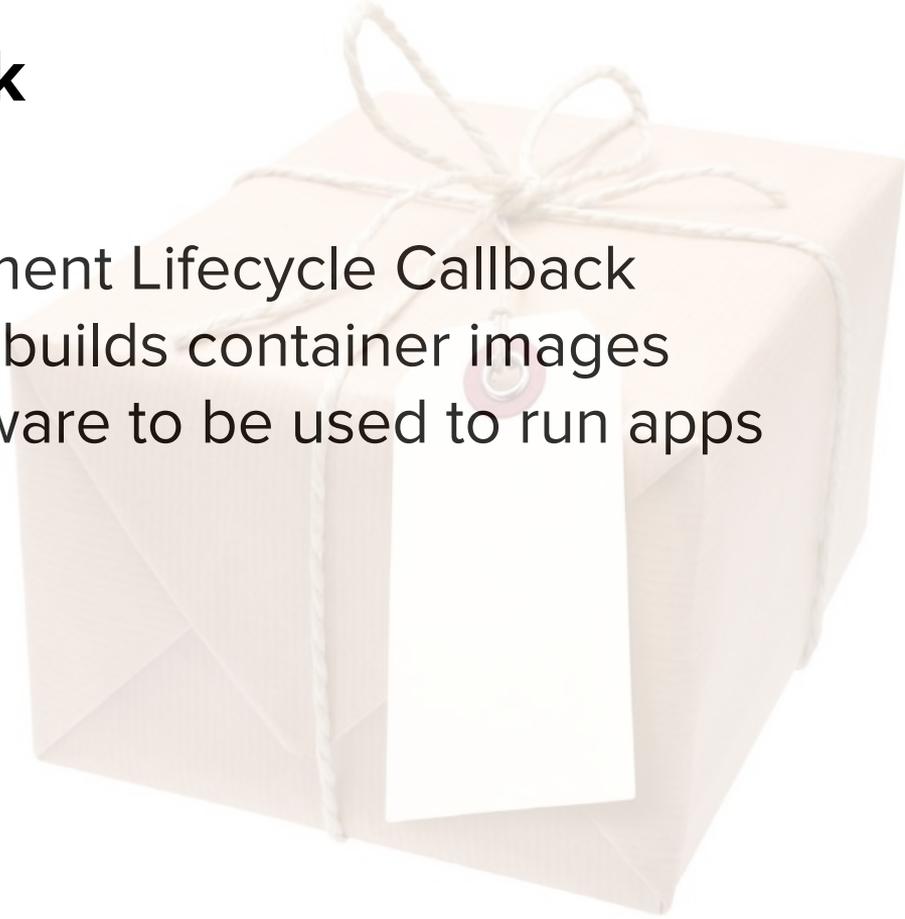


## Step 4 - Run

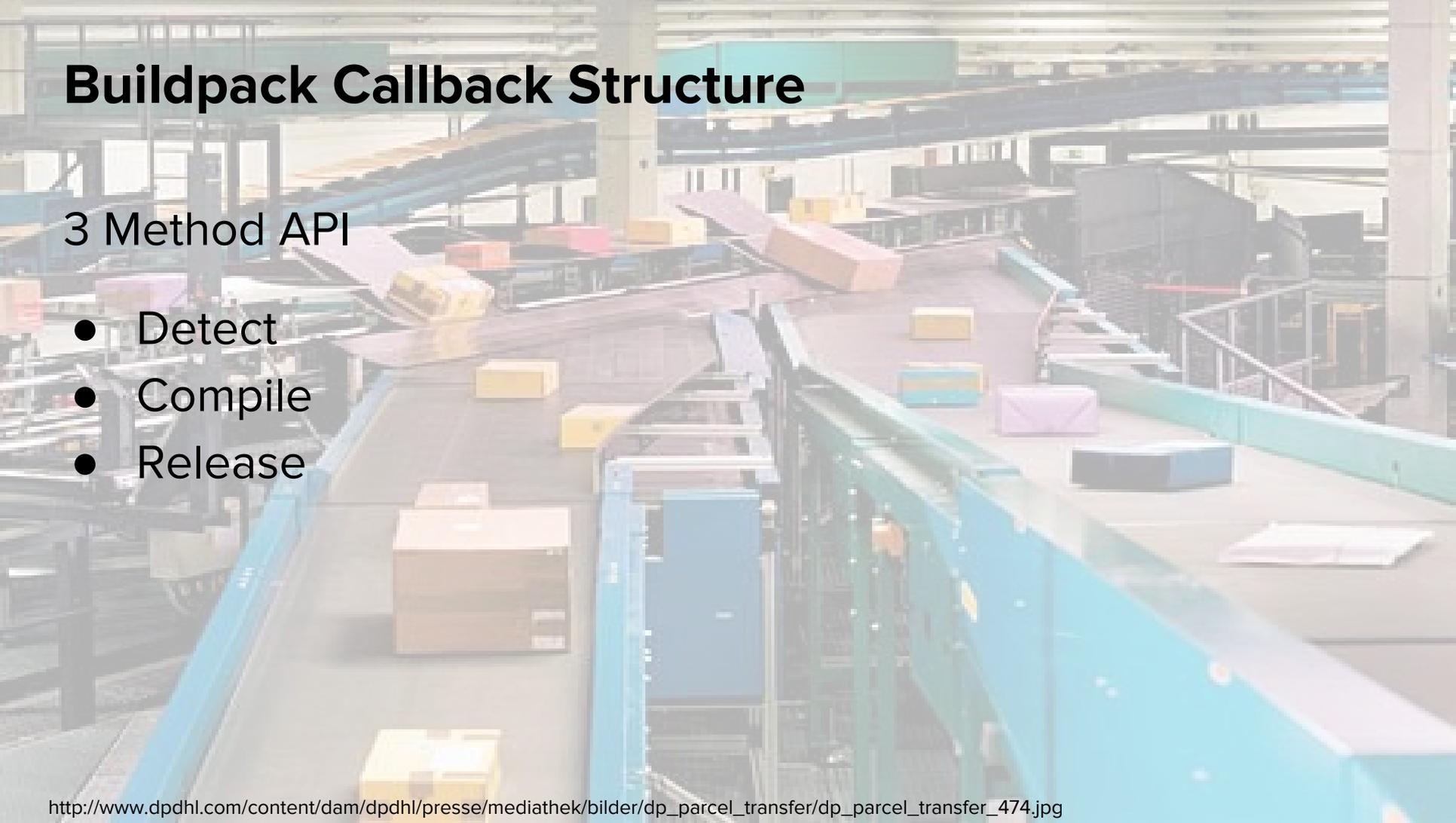
Assigned cluster cell starts application using provided metadata and informs the platform load balancer/router that application traffic can be routed to container endpoint

# The Buildpack

- App Deployment Lifecycle Callback
- Process that builds container images
- Defines software to be used to run apps



# Buildpack Callback Structure



## 3 Method API

- Detect
- Compile
- Release

# Detect

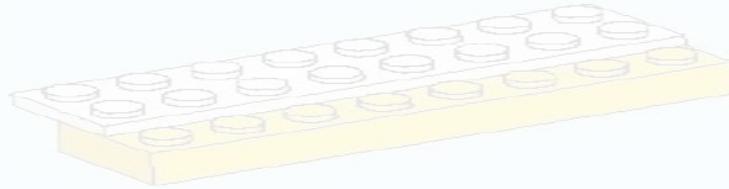
Determine if attributes of the app type(s) that can be handled exist in provided artifacts

# Compile

Collect and assemble necessary software to run the application type



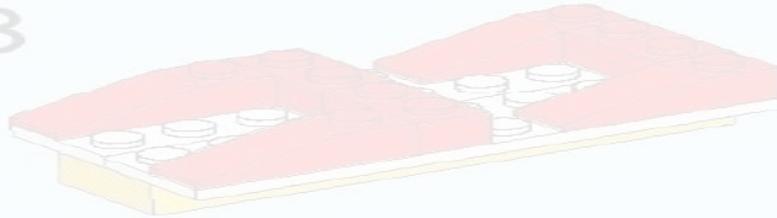
1



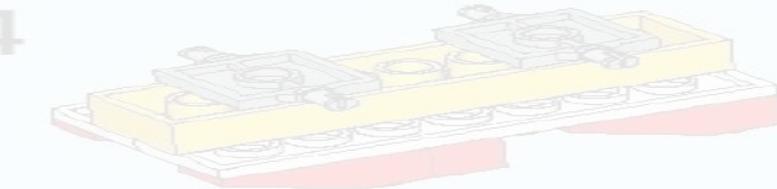
2



3



4



# Release

A person wearing a light-colored shirt and dark shorts is holding a large, fluffy, brown and white dog. The dog is being held in front of the person, and its head is turned towards the left. The background is a bright, grassy field with some trees in the distance. The overall scene is outdoors and appears to be a release or a walk in a park.

Define how to run the packaged application

# Provided Buildpacks (Opinions)

<http://docs.cloudfoundry.org/buildpacks/index.html>

- Binary
- GO
- Java
- .Net Core
- Node JS
- PHP
- Python
- Ruby
- Staticfile

# Services

- Application add-on that provisioned to provide an functionality leveraged by application
  - Databases (MySQL, Mongo, Redis) and Rabbit MQ
- Marketplace service instances are created and bound using a Service Broker
- Non Platform services connected via User Provided Service
- Bound services connection information made available to application at runtime

# Service Broker

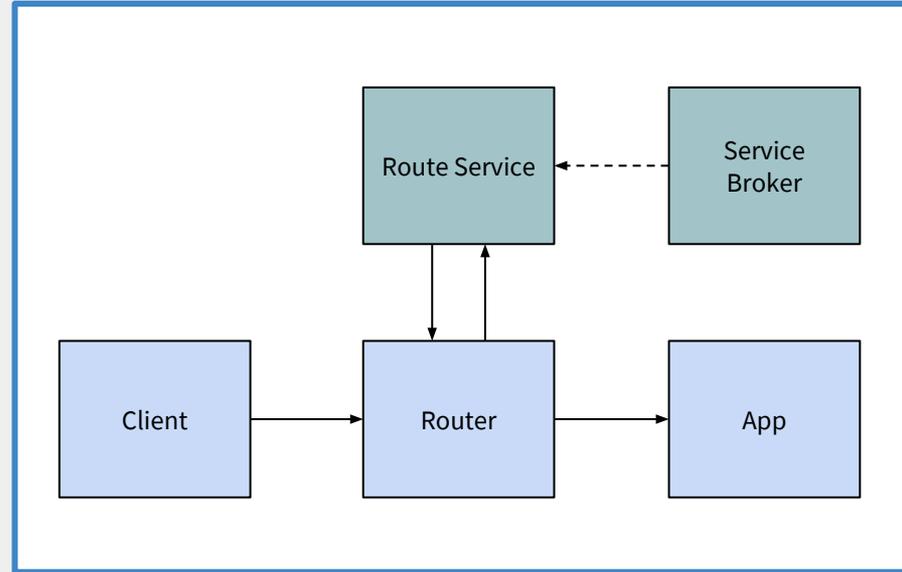
- REST endpoint implementing contract for self service creation and binding to apps by authorized users
  - create
  - delete
  - bind
  - unbind
  - catalog

# Loggregator

- System to collate and expose log event stream via API
  - Applications logs to STDOUT and STDERR
  - Cloud Foundry component events relevant to app
- Application users and operators to:
  - Tail their application logs
  - Dump a recent set of application logs
  - Continually drain their application logs to 3rd party log archive and analysis service

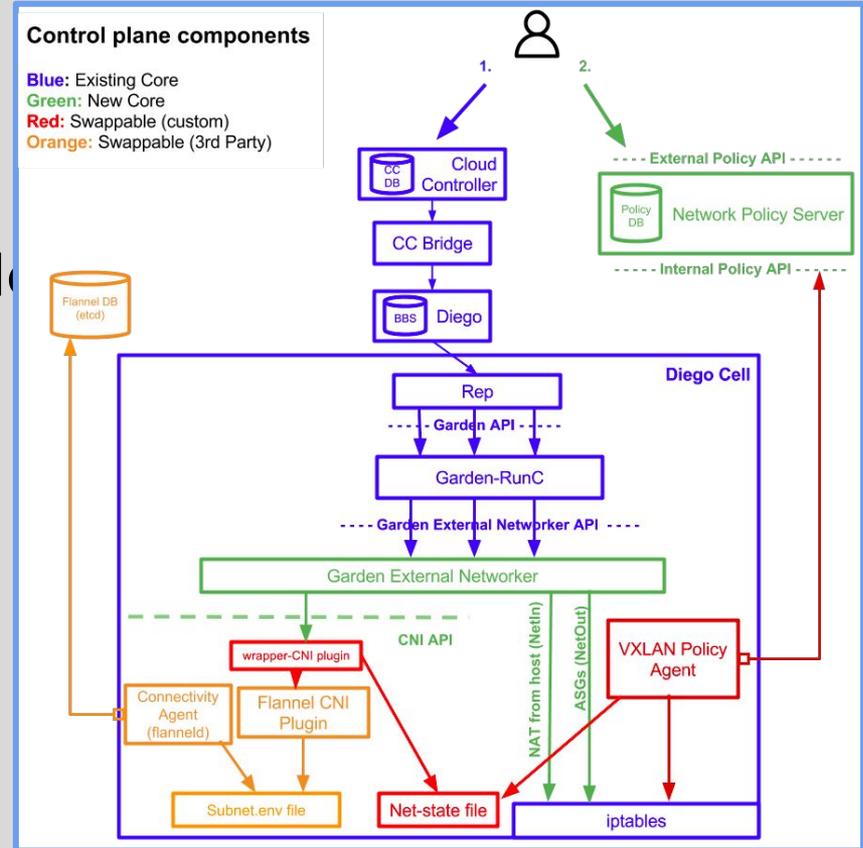
# Route Services

- Meant to insert component functionality into the application request path.
  - API gateway
  - Security gateways
  - Analytics gathering



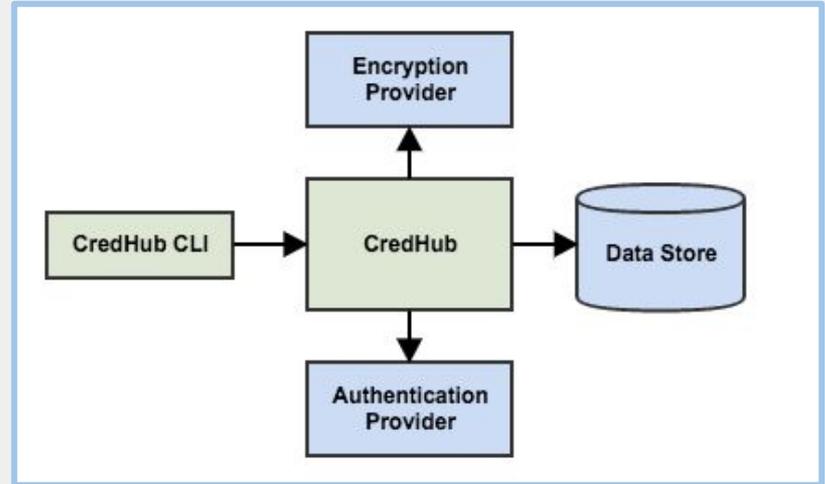
# Container Networking Control

- Means to secure communication between containers via a SDN provider
  - Defined as app policies
- CNI API used to provide common interaction across SDN Providers



# Credential Management

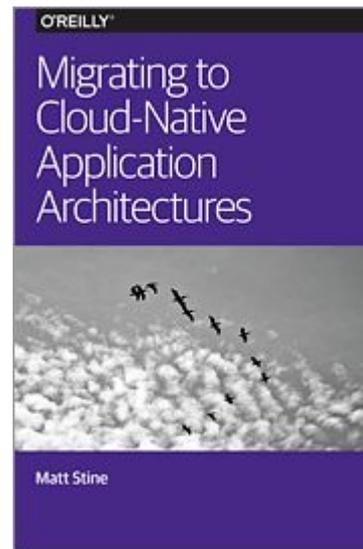
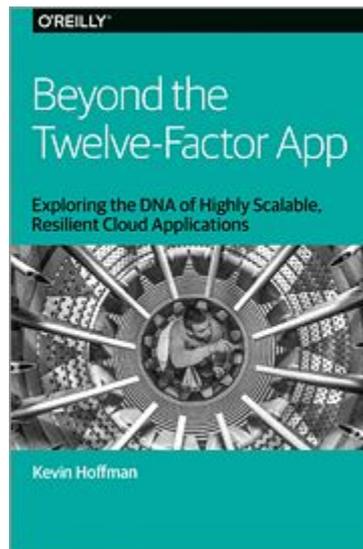
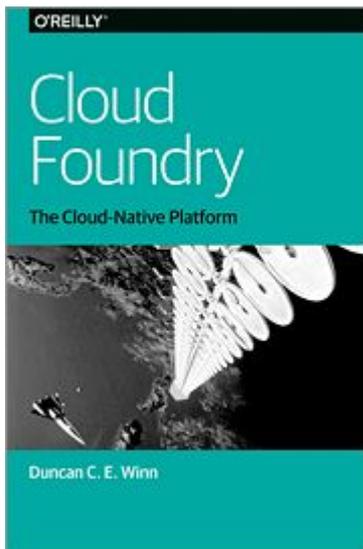
- CredHub
- Central point of control for platform credential:
  - Generation
  - Storage
  - lifecycle management
  - access control
  - access event logging



A photograph of three people. On the left and right are individuals in skeleton costumes, including a white skull mask with black eye sockets and a white ribcage. They have white hair and are wearing black headbands. The person in the center is a man with dark hair, wearing a black shirt with a repeating orange and red skull pattern. He has a surprised or excited expression with his mouth open and hands raised. The background is dark and out of focus.

**Any Questions?**

# Some Further Reading



<https://content.pivotal.io/ebooks>