
Getting Started with
Cloud Foundry
Ian Huston, Data Scientist
Pivotal
PyData London 2015

Who am I?

- Ian Huston
- @ianhuston
- www.ianhuston.net
- Talk resources: https://github.com/ihuston/python-cf-examples

- Data Scientist
- Use PyData stack for predictive analytics and machine learning
- Previously a theoretical physicist using Python for numerical simulations & HPC

Pivotal

Who are Pivotal?

Apache Tomcat

CLOUD FOUNDRY™

OPEN DATA PLATFORM

Pivotal
Big Data Suite

Pivotal™

Aims for this tutorial

Goals:

- Understand how to use CF as a developer/data scientist
- Understand how to push PyData apps and bind to services

Anti-goals:

- Understand CF from an operations viewpoint
- Install & run a full Cloud Foundry installation

Pivotal

Plan

- What is Cloud Foundry?
- Getting started with a hosted CF instance
- Pushing your first app
- What are buildpacks?
- Using PyData packages
- Using data services like Redis
- Putting it all together

Pivotal

Important Resources

Clone this repo:

https://github.com/ihuston/python-cf-examples

OR download and unpack this file:

http://tinyurl.com/cf-pydata

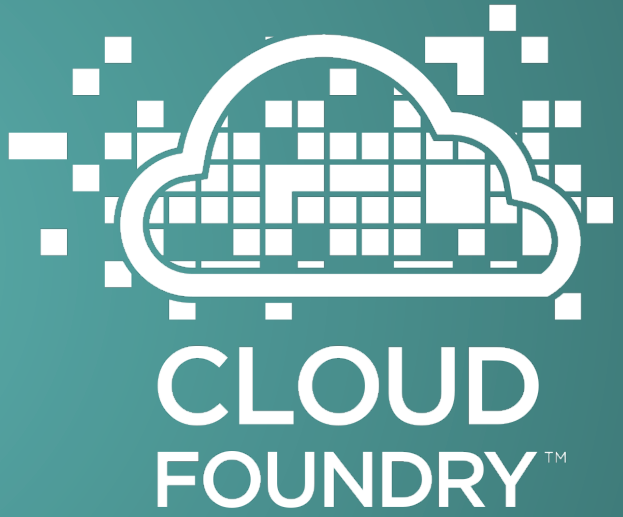
Pivotal

What is Cloud Foundry?

http://cloudfoundry.org

Open Source
Multi-Cloud Platform

Simple App Deployment,
Scaling & Availability



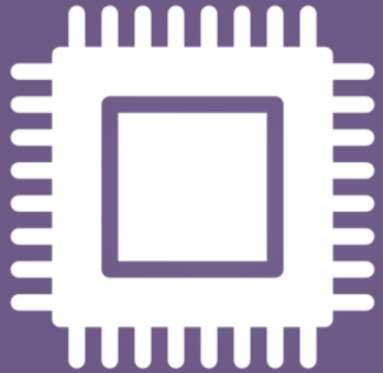
Cloud Applications Haiku

Here is my source code
Run it on the cloud for me
I do not care how.

- Onsi Fakhouri
@onsijoe

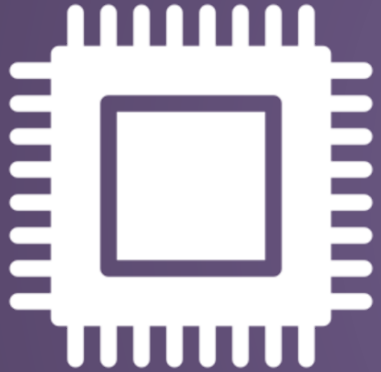
Pivotal

How can data scientists use CF?



Data Services

Easy control of incoming data

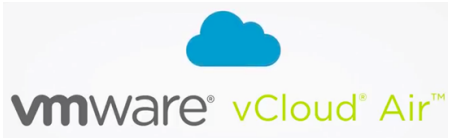


Distributed computation

Data Driven Applications

Multi-cloud

- Same applications running across different cloud providers:



Private

Public

Hosted

Pivotal

Cloud Foundry Foundation

See relative contributions at http://dashboard.cloudfoundry.org/

Pivotal

Hosted options

Pivotal
Web Services

- IBM Bluemix
- Anynines
- HP Helion Development Platform
- Activestate Stackato
- CenturyLink AppFog

Pivotal

Getting Started on Pivotal Web Services

- Go to http://run.pivotal.io
- **Trial lasts for 60 days, no credit card required**
- Click ‘Sign Up’ and enter your email
- Open confirmation email and verify registration
- Complete SMS verification check
- Choose an organisation name (your name/handle is fine)

Pivotal

Downloading the Command Line Interface

- Register at http://run.pivotal.io
- In the CF management website click ‘Tools’ and download appropriate package
- Or go to https://github.com/cloudfoundry/cli
- Or use homebrew on OSX:

```
$ brew tap pivotal/tap
$ brew install cloudfoundry-cli
```

- Provides the `cf` command, your interface to Cloud Foundry

Pivotal

cf push

Challenge 1: Pushing your first app

Pivotal

Logging in to PWS

- Choose PWS API endpoint:

```
$ cf api https://api.run.pivotal.io
```

- Login:

```
$ cf login
```

- Enter your email address and password
- You should see output like:

```
API endpoint: https://api.run.pivotal.io (API version:2.28.0)
User:           YOUR_USER_NAME
Org:            YOUR_ORG_NAME
Space:          development
```

Pivotal

Challenge 1: Pushing your first application

- Choose PWS API endpoint: `$ cf api https://api.run.pivotal.io`
- Login: `$ cf login`
- Code directory: `01-simple-python-app`
- Deploy with

```
$ cf push
```

- Check it worked at http://myapp-RANDOM-WORDS.cfapps.io
- Turn off your app when finished with `cf stop myapp` (but not yet!)

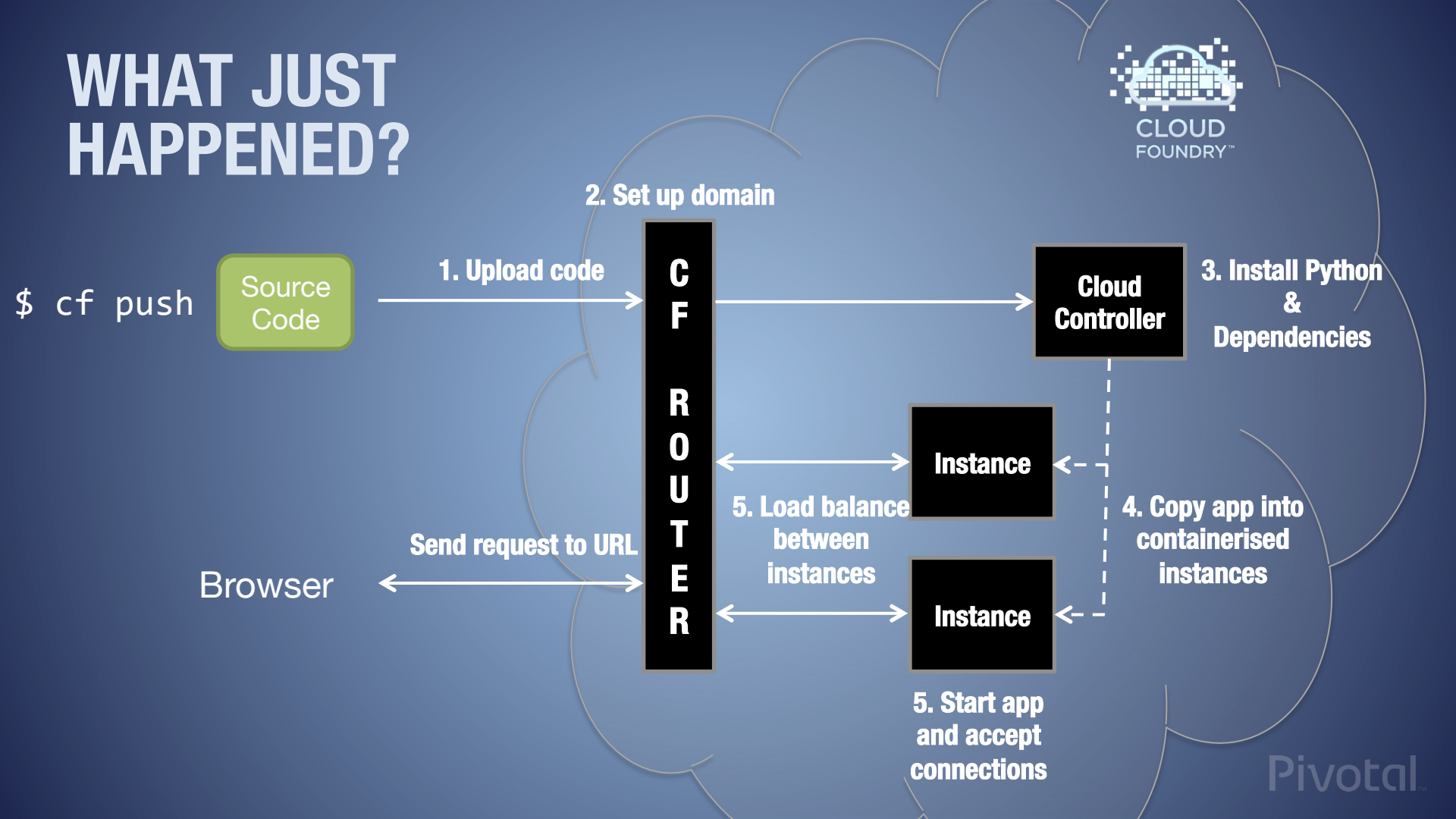
Pivotal

Simple Flask App Demo

- Simple one page “Hello World” web app
- Video: https://www.youtube.com/watch?v=QOfD6tnoAB8
- Demonstrates:
  - Installation of requirements
  - Scaling properties
- Need to Provide:
  - App files
  - Dependencies listed in requirements.txt file
  - Optional manifest.yml file with configuration for deployment

Pivotal

WHAT JUST HAPPENED?



What just happened?

1. Application code is uploaded to CF
2. Domain URL is set up ready for routing
3. Cloud controller builds application in container:
   - Python interpreter selected
   - Dependencies installed with pip
4. Container is replicated to provide instances
5. App starts and Router load balances requests

- See what's happening using logs: `$ cf logs myapp --recent`

Pivotal

Python on Cloud Foundry

- First class language (with Go, Java, Ruby, Node.js, PHP)
- Automatic app type detection
  - Looks for requirements.txt or setup.py
- **Buildpack** takes care of
  - Detecting that a Python app is being pushed
  - Installing Python interpreter
  - Installing packages in requirements.txt using pip
  - Starting web app as requested (e.g. python myapp.py)

Pivotal

Scaling memory and instances

- We can scale our application as needed in terms of memory and number of instances:

```
$ cf scale myapp -i 5
$ cf scale myapp -m 256M
```

- Check app in browser to see different ports being used.
- Scale back down with `$ cf scale myapp -i 1`

Pivotal

How does this work?

- Containerised application is cached and ready to be deployed.
- Scaling number of instances replicates container and load balances requests across all instances.
- Scaling memory requires restarting app.
- Auto-scaling is also possible.

Pivotal

Buildpacks

Challenge 2: Pushing PyData apps

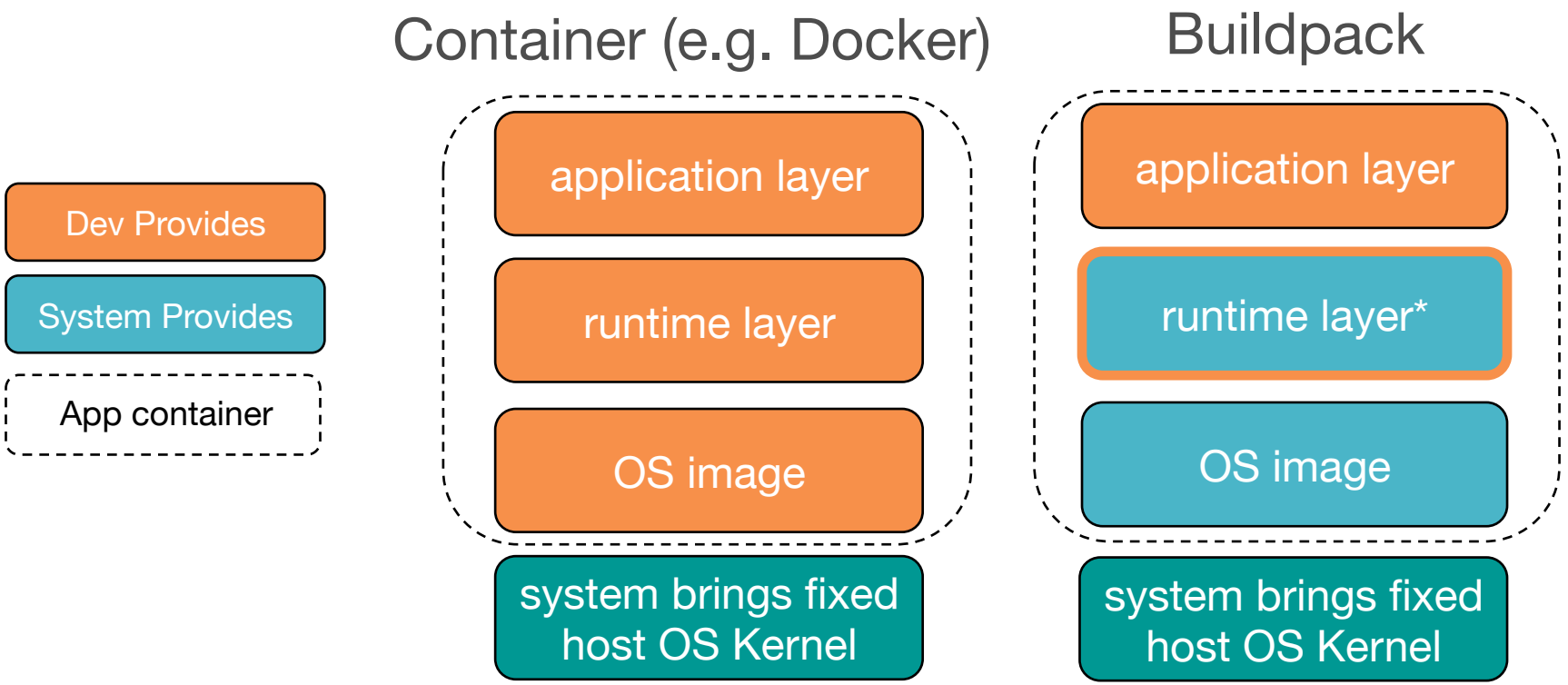
Pivotal

What are buildpacks?

- Idea and format from Heroku
- Responsible for doing whatever is necessary to get your app running.
- Buildpacks take care of
  - Detecting which type of application is being pushed
  - Installing the appropriate run-time
  - Installing required dependencies or other artifacts
  - Starting the application as requested
- Official buildpacks for Python, Java, Node.js, Go, Ruby, PHP & for static websites and running binaries

Pivotal

Containers vs Buildpacks



* Devs may bring a custom buildpack

Pivotal

Custom buildpacks

- Instead of using an official buildpack you can use any custom buildpack installed on your CF or available on Github.
- Only 3 shell scripts needed:
  - detect
  - compile
  - release
- Specify buildpack with `-b` or in manifest.yml

Pivotal

Community Buildpacks

https://github.com/cloudfoundry-community/cf-docs-contrib/wiki/Buildpacks

- Lots of languages:
  Clojure, Haskell, .NET, Erlang, Elixir, etc.
- RShiny app buildpack:
  https://github.com/alexkago/cf-buildpack-r
- Can also use some Heroku buildpacks without modification.

Pivotal

Official Python buildpack

- ✓ Great for simple pip based requirements
- ✓ Well tested and officially maintained
- ✓ Covers both Python 2 and 3
- ✗ Suffers from the Python Packaging Problem:
  - Hard to build packages with C, C++ or Fortran extensions
  - Complicated local configuration of libraries and paths needed
  - Takes a long time to build main PyData packages from source

Pivotal

Using conda for package management

CONTINUUM
ANALYTICS

- http://conda.pydata.org
- Benefits:
  - Uses precompiled binary packages
  - No fiddling with Fortran or C compilers and library paths
  - Known good combinations of main package versions
  - Really simple environment management (better than virtualenv)
  - Easy to run Python 2 and 3 side-by-side

Go try it out if you haven't already!

Pivotal

How to use the conda buildpack

https://github.com/ihuston/python-conda-buildpack

- Specify as a custom buildpack when pushing app with manifest or `-b` command line option.
- Export your current environment to a environment.yml file
- Or write requirements.txt (pip) and conda_requirements.txt
- Send me feedback & pull requests!

Pivotal

Challenge 2: Pushing a PyData app

- Code directory: `02-pydata-spyre-app`
- Spyre – Adam Hajari https://github.com/adamhajari/spyre
- This app uses the Pydata buildpack to install Matplotlib, NumPy and more.
- Spyre provides a simple way to build interactive web based visualisations similar to Rshiny.

Pivotal

Using Services

Challenge 3: Using Redis in an app

Pivotal

Services

These are available on Pivotal CF (Pivotal's packaged Cloud Foundry offering).

See http://run.pivotal.io for the services available on Pivotal Web Services.

Pivotal

Cloud Native Applications

- Suitable for deployment on cloud platforms
- Can scale up easily without changes
- Follow these rules: http://12factor.net
- In a nutshell:
  - Apps are stateless processes
  - Easy to create and destroy apps with no side-effects
  - Persistent state handled through backing services
  - Interact with services through port binding

Pivotal

Challenge 3: Binding a service to an app

- Create a free Redis service:

```
$ cf cs rediscloud 30mb myredis
```

- Bind to our earlier app:

```
$ cf bind-service myapp myredis
```

- See how apps find credentials in environmental variables:

```
$ cf env myapp
```

Pivotal

Challenge 3: Using Services in an app

- Code directory: `03-services-redis`
- This app binds to a Redis service if available and counts the number of hits on the app homepage.
- Add keys and values by adding /keyname/value to URL.
- Data is persisted in Redis, will survive deleting and restarting the app.
- Multiple instances all access the same service.

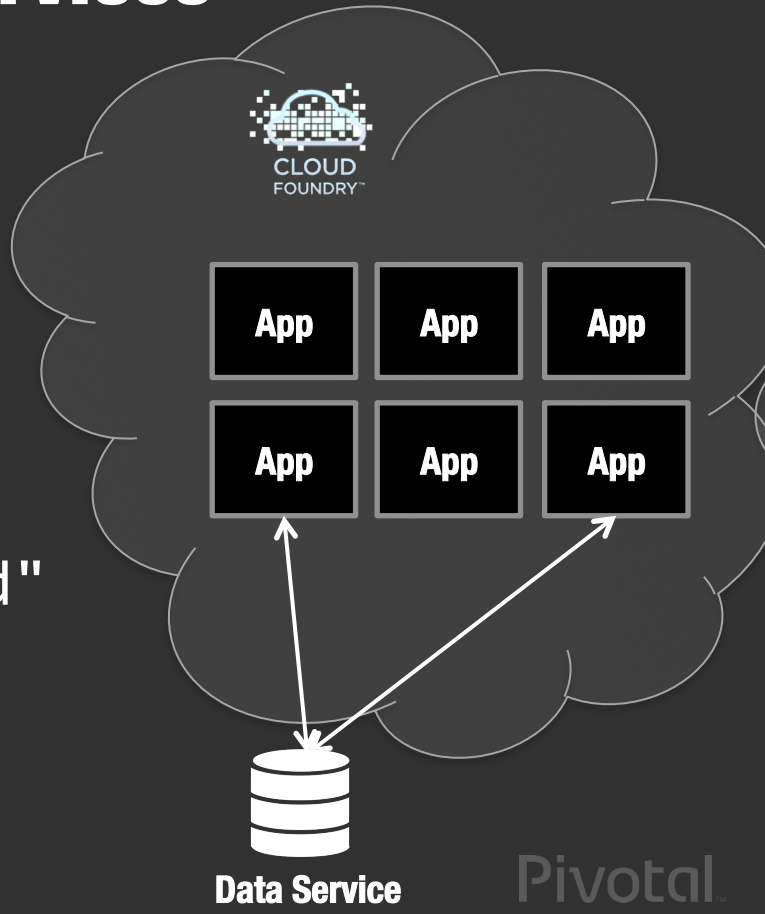
Pivotal

User Provided Services

How to add User Provided Services:

Standalone Hadoop or Apache Spark cluster, Big Data System, RDBMS etc.

```
$ cf cups SERVICE_INSTANCE -p
"host, port, username, password"
```



Pivotal

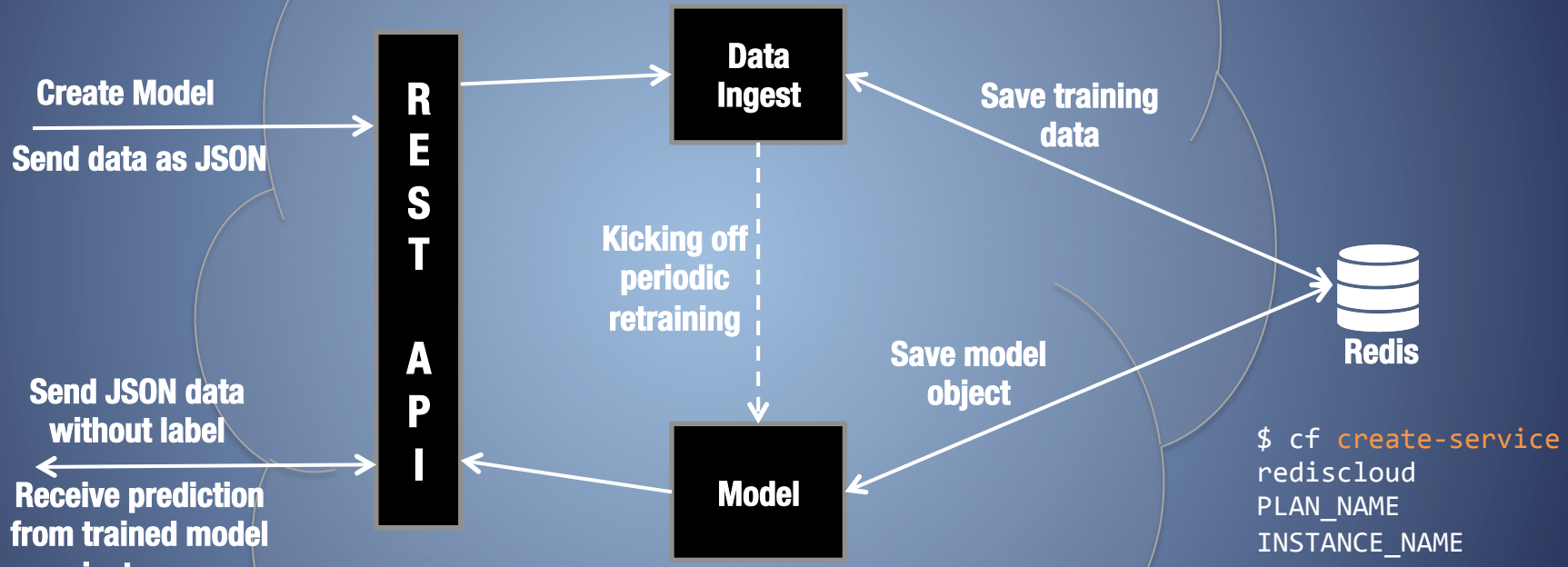
Putting it all together

Challenge 4: Build your own prediction API

Pivotal

PREDICTION API ARCHITECTURE

Deployed at:
http://dsoncf.cfapps.io



Code:
https://github.com/alexkago/ds-cfpylearning

Pivotal

Final challenge: Build your own predictive API

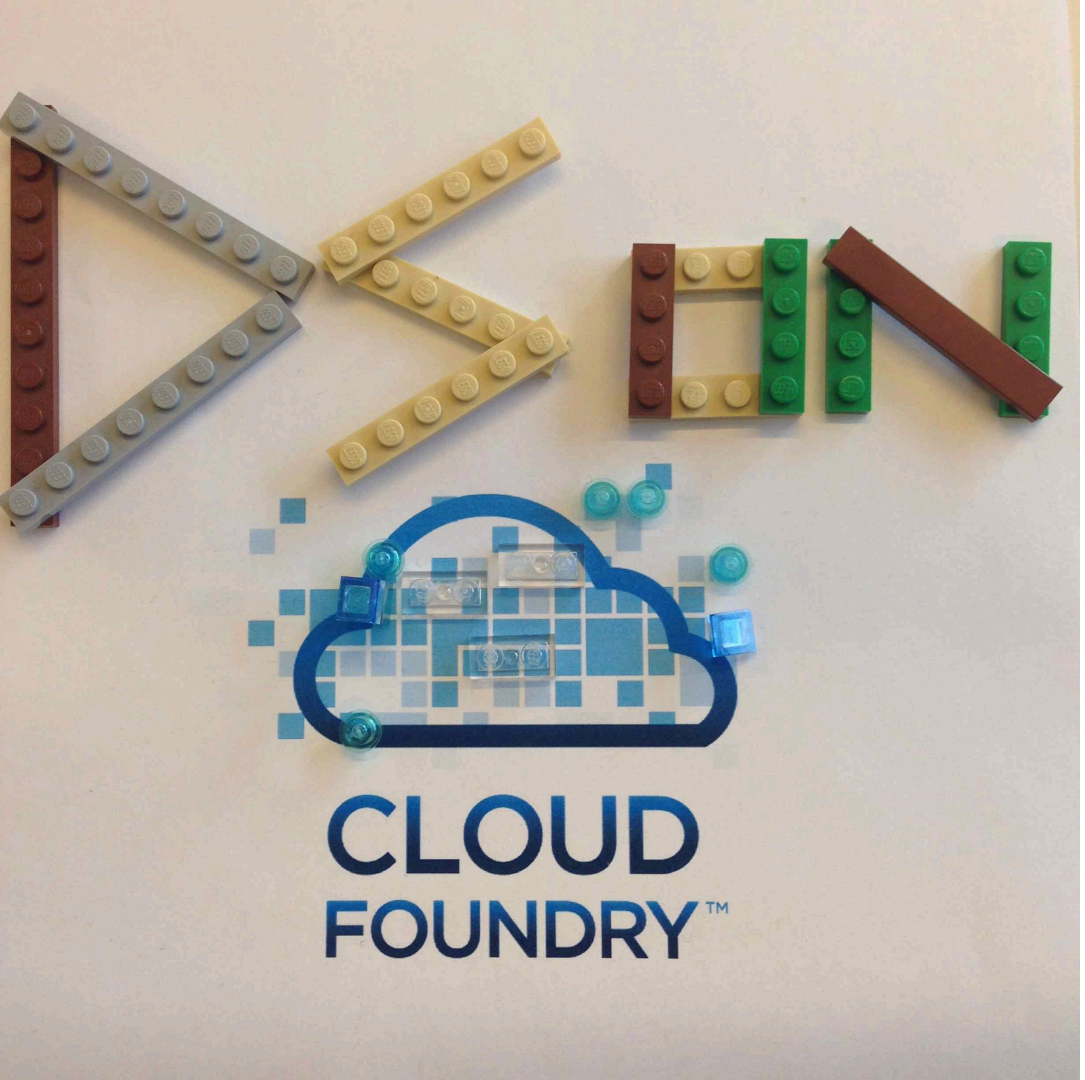
- Code directory: `04-learning-api`
- Simple Flask + scikit-learn based machine learning API
- Push the application and go to the app URL to see instructions on how to use.
- Model is built in scikit-learn and persisted in Redis
- Simplified version of this project by Alex Kagoshima: https://github.com/alexkago/ds-cfpylearning

Pivotal

Resources

- Docs: http://docs.cloudfoundry.org
- CF Summit videos: http://cfsummit.com
- Join the CF community: http://cloudfoundry.org
- CF meetups: http://cloud-foundry.meetup.com
- Don’t forget to stop your apps with `cf stop myapp`.

Pivotal



Show off your data science related Cloud Foundry apps:

Twitter: @dsoncf

http://dsoncf.com

Pivotal

@ianhuston

Pivotal