Pivotal

PCF Dev Enablement

- cf push

Derrick Chua
Senior Platform Architect
tchua@pivotal.io
July 2019

Pre-requisites (DBS)

- PAS account in DBS environment
- Download and install cf cli
  - According to DBS protocol
  - Desired state:

```
Derrick-Chuas-MacBook-Pro:employees-api tmchua$ cf
cf version 6.41.0+dd4c76cdd.2018-11-28, Cloud Foundry command line tool
Usage: cf [global options] command [arguments...] [command options]

Before getting started:
  config    login,l      target,t
  help,h    logout,lo

Application lifecycle:
  apps,a        run-task,rt    events
  push,p        logs           set-env,se
  start,st      ssh            create-app-manifest
  stop,sp       app            delete,d
  restart,rs    env,e
  restage,rg    scale

Services integration:
  marketplace,m        create-user-provided-service,cups
  services,s           update-user-provided-service,uups
  create-service,cs    create-service-key,csk
```

Onsi Fakhouri
@onsijoe

cf push haiku

here is my source code
run it on the cloud for me
i do not care how

3:18pm · 12 May 2015 · Twitter for iPhone

59 RETWEETS **87** LIKES

Pivotal

Experience cf push

Pivotal

Hands-On: Step 1

- Create hands on directory “~/workshop”
- Download employees-api microservice from https://github.com/derrick81/cna-dev-training
- Unzip download package to “~/workshop”
- Desired state:

```
Derrick-Chuas-MacBook-Pro:cna-dev-training-master tmchua$ pwd
/Users/tmchua/workshop/cna-dev-training-master
Derrick-Chuas-MacBook-Pro:cna-dev-training-master tmchua$ ls
Icon?                generate-load.sh     mvnw                 src
README.md            manifest-service.yml mvnw.cmd             target
employees-api.iml    manifest.yml         pom.xml
```

You can clear the contents of ~/workshop on your training machine if it already exists

Hands-On: Step 2

- Using cf cli, log in to your PWS account using the following command
  - cf login -a api.run.pivotal.io

```
Derrick-Chuas-MacBook-Pro:cna-dev-training-master tmchua$ cf login -a api.run.pivotal.io
API endpoint: api.run.pivotal.io

Email> tchua@pivotal.io

Password>
Authenticating...
OK

Select an org (or press enter to skip):
1. APJ
2. tchua

Org> 2
Targeted org tchua

Targeted space development



API endpoint:   https://api.run.pivotal.io (API version: 2.137.0)
User:           tchua@pivotal.io
Org:            tchua
Space:          development
```

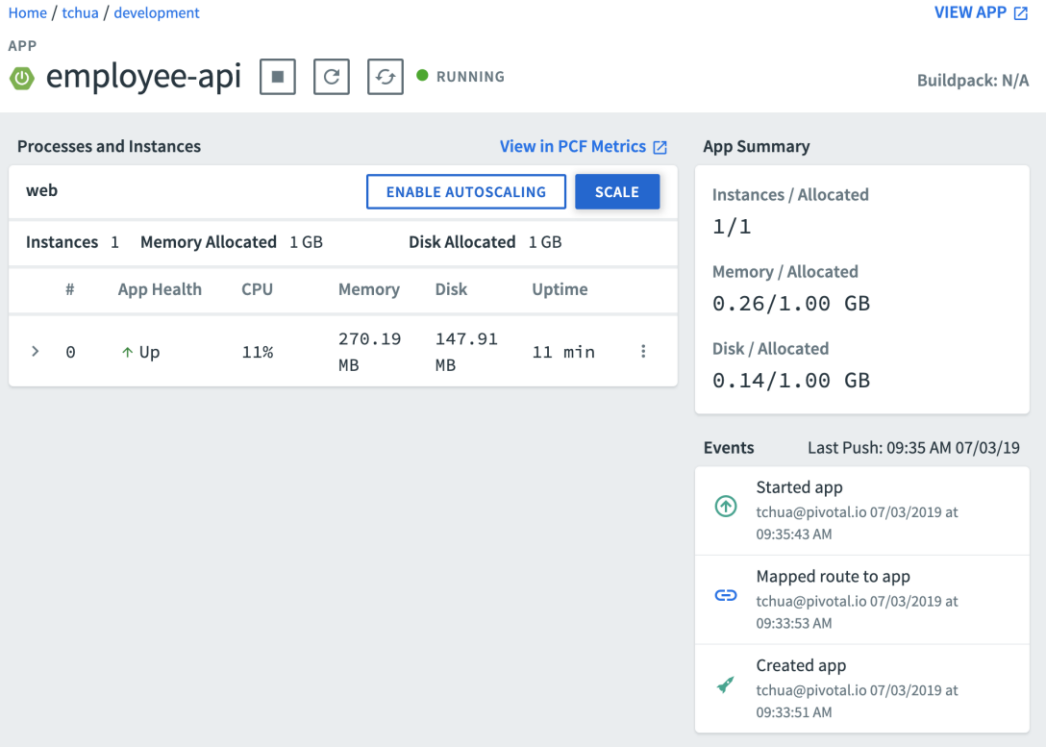
Hands-On: Step 3

- cf push

```
Derrick-Chuas-MacBook-Pro:cna-dev-training-master tmchua$ cf push
Pushing from manifest to org tchua / space development as tchua@pivotal.io...
Using manifest file /Users/tmchua/workshop/cna-dev-training-master/manifest.yml
Getting app info...
Creating app with these attributes...
+ name:        employee-api
  path:        /Users/tmchua/workshop/cna-dev-training-master/target/employeesapi-0.0.1-SNAPSHOT.jar
+ instances:   1
+ memory:      1G
  routes:
+   employee-api-bright-zebra.cfapps.io

Creating app employee-api...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
 400.58 KiB / 400.58 KiB [==========================================================================] 100.00% 1
m28s

Waiting for API to complete processing files...
timeout connecting to log server, no log will be shown


Staging app and tracing logs...
   Downloading dotnet_core_buildpack_beta...
   Downloading staticfile_buildpack...
```

Hands-On: Step 4

- Log in to your PWS account and view your application
  - https://run.pivotal.io/



Pivotal

Hands-On: Step 5

- Query the employees API GET methods

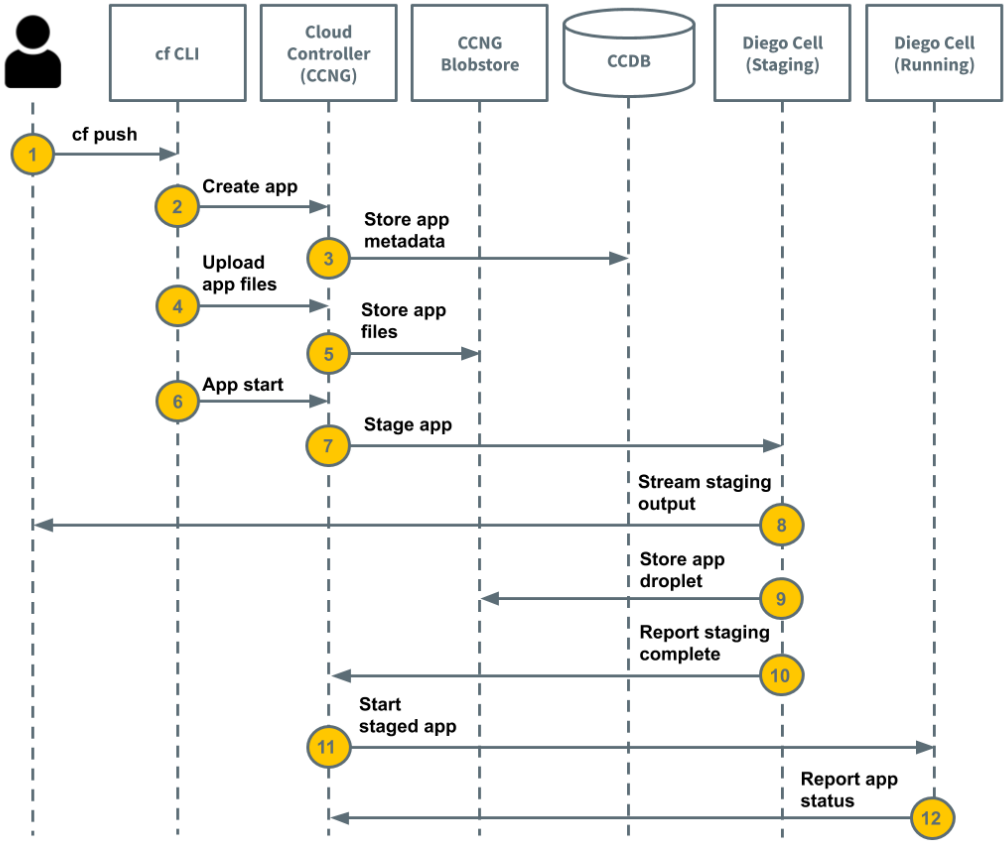
https://employee-api-bright-zebra.cfapps.io/employees

Use your own mapped domain name


```
{
  "_embedded" : {
    "employees" : [ {
      "name" : "pas",
      "office" : "AU",
      "deskNum" : 123,
      "_links" : {
        "self" : {
          "href" : "https://employee-api-bright-zebra.cfapps.io/employees/1"
        },
        "employee" : {
          "href" : "https://employee-api-bright-zebra.cfapps.io/employees/1"
        }
      }
    }, {
      "name" : "lucia",
      "office" : "SG",
      "deskNum" : 456,
      "_links" : {
        "self" : {
          "href" : "https://employee-api-bright-zebra.cfapps.io/employees/2"
        },
        "employee" : {
          "href" : "https://employee-api-bright-zebra.cfapps.io/employees/2"
        }
```

Hands-On: Explore

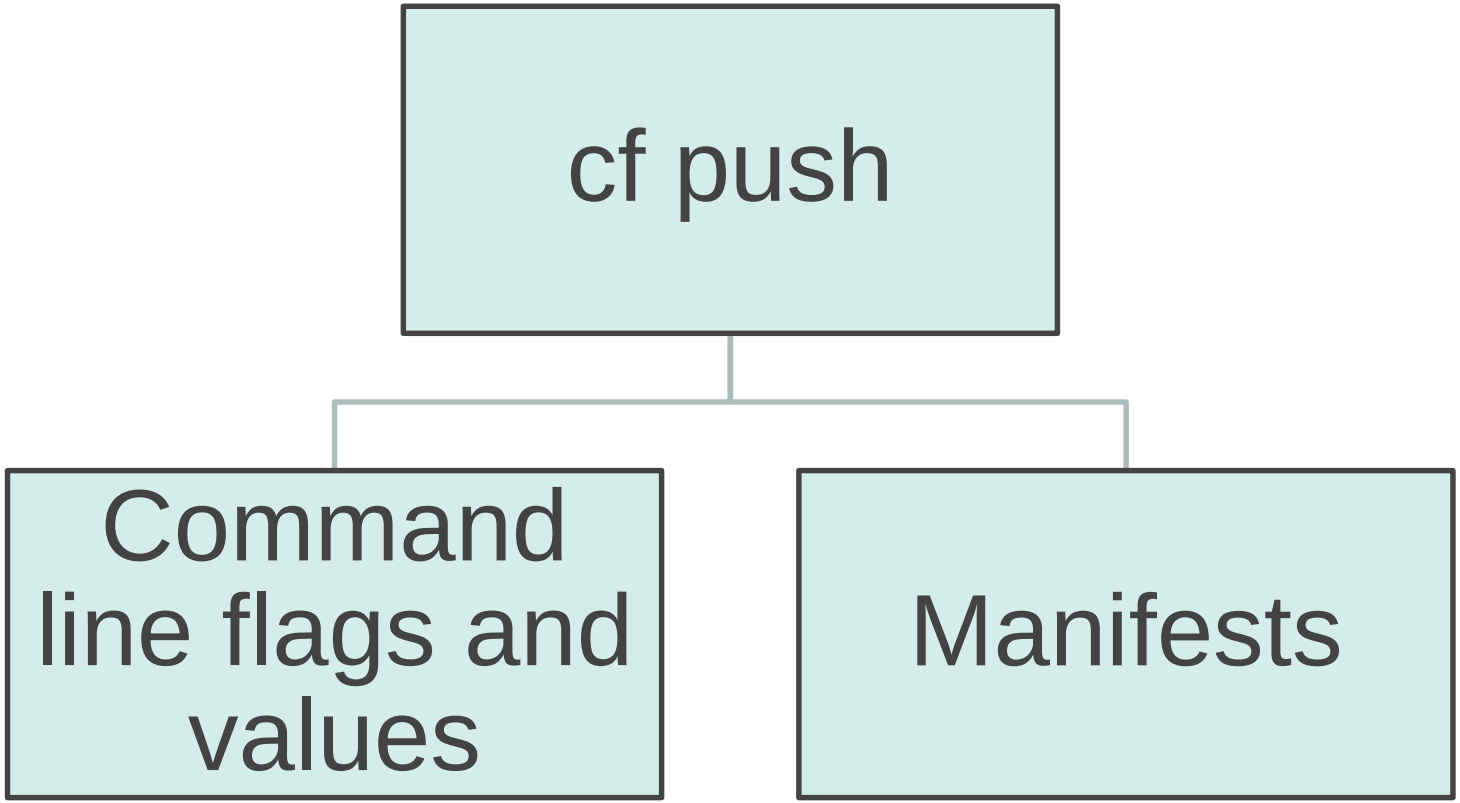
- Apps Manager UI
- cf push console output



Manifest

Pivotal

Two ways to cf push



Manifest format

Manifests are written in YAML. The manifest below illustrates some YAML conventions, as follows:

- The manifest begins with three dashes.
- The `applications` block begins with a heading followed by a colon.
- The app `name` is preceded by a single dash and one space.
- Subsequent lines in the block are indented two spaces to align with `name`.

```
---
applications:
- name: my-app
  memory: 512M
  instances: 2
```

Details at https://docs.pivotal.io/pivotalcf/2-6/devguide/deploy-apps/manifest-attributes.html

Think about...

- Why use manifest?
- What's the typical length of a manifest?
- How to deploy multiple apps with a single manifest?
- How is this different from a Kubernetes deployment yaml?

Buildpacks

Pivotal

What are buildpacks?

- Provide framework and runtime support for apps.

- Examine your apps to determine
  - what dependencies to download
  - how to configure the apps to communicate with bound services

- When you push an app, Cloud Foundry automatically detects an appropriate buildpack for it.
  - This buildpack is used to compile or prepare your app for launch.

How buildpacks work?

A buildpack repository may contain the following five scripts in the `bin` directory:

- `bin/detect` determines whether or not to apply the buildpack to an app.
- `bin/supply` provides dependencies for an app.
- `bin/finalize` prepares the app for launch.
- `bin/release` provides feedback metadata to Cloud Foundry indicating how the app should be executed.
- `bin/compile` is a deprecated alternative to `bin/supply` and `bin/finalize`.

The `bin/supply` and `bin/finalize` scripts replace the deprecated `bin/compile` script. Older buildpacks may still use `bin/compile` with the latest version of Cloud Foundry. In this case, applying multiple buildpacks to a single app is not supported. Similarly, newer buildpacks may still provide `bin/compile` for compatibility with Heroku and older versions of Cloud Foundry.

Details at https://docs.pivotal.io/pivotalcf/2-6/buildpacks/understand-buildpacks.html

cf buildpacks

```
Derrick-Chuas-MacBook-Pro:cna-dev-training-master tmchua$ cf buildpacks
Getting buildpacks...

buildpack                  position   enabled   locked   filename                                          stack
staticfile_buildpack       1          true      false    staticfile_buildpack-cached-cflinuxfs3-v1.4.43.zip   cflinuxfs3
java_buildpack             2          true      false    java-buildpack-offline-cflinuxfs3-v4.19.zip          cflinuxfs3
ruby_buildpack             3          true      false    ruby_buildpack-cached-cflinuxfs3-v1.7.41.zip         cflinuxfs3
dotnet_core_buildpack      4          true      false    dotnet-core_buildpack-cached-cflinuxfs3-v2.2.12.zip  cflinuxfs3
nodejs_buildpack           5          true      false    nodejs_buildpack-cached-cflinuxfs3-v1.6.51.zip       cflinuxfs3
go_buildpack               6          true      false    go_buildpack-cached-cflinuxfs3-v1.8.41.zip           cflinuxfs3
python_buildpack           7          true      false    python_buildpack-cached-cflinuxfs3-v1.6.34.zip       cflinuxfs3
php_buildpack              8          true      false    php_buildpack-cached-cflinuxfs3-v4.3.78.zip          cflinuxfs3
binary_buildpack           9          true      false    binary_buildpack-cached-cflinuxfs3-v1.0.32.zip       cflinuxfs3
staticfile_buildpack       10         true      false    staticfile_buildpack-cached-cflinuxfs2-v1.4.43.zip   cflinuxfs2
java_buildpack             11         true      false    java-buildpack-offline-cflinuxfs2-v4.19.zip          cflinuxfs2
ruby_buildpack             12         true                                                                    cflinuxfs2
dotnet_core_buildpack      13         true                                                                    cflinuxfs2
nodejs_buildpack           14         true                                                                    cflinuxfs2
go_buildpack               15         true                                                                    cflinuxfs2
python_buildpack           16         true                                                                    cflinuxfs2
php_buildpack              17         true                                                                    cflinuxfs2
binary_buildpack           18         true                                                                    cflinuxfs2
dotnet_core_buildpack_beta 19         true
hwc_buildpack              20         true      false    hwc_buildpack-cached-windows2016-v3.1.9.zip          windows2016
binary_buildpack           21         true      false    binary_buildpack-cached-windows2016-v1.0.32.zip      windows2016
hwc_buildpack              22         true      false    hwc_buildpack-cached-windows-v3.1.9.zip              windows
binary_buildpack           23         true      false    binary_buildpack-cached-windows-v1.0.32.zip          windows
```

A stack is a prebuilt root file system (rootfs) that supports a specific operating system. For example, Linux-based systems need /usr and /bin directories at their root. The stack works in tandem with a buildpack to support apps running in compartments.

Let’s look at the Java buildpack

https://github.com/cloudfoundry/java-buildpack

Think about...

- How does buildpack help you in your deployment process?
- Is there a buildpack that you want that is not in the default list?
- Can you build your own buildpack?

Making sense of cf push

Pivotal

Let’s revisit the cf push console output

```
Derrick-Chuas-MacBook-Pro:cna-dev-training-master tmchua$ cf push
Pushing from manifest to org tchua / space development as tchua@pivotal.io...
Using manifest file /Users/tmchua/workshop/cna-dev-training-master/manifest.yml
Getting app info...
Creating app with these attributes...
+ name:        employee-api
  path:        /Users/tmchua/workshop/cna-dev-training-master/target/employeesapi-0.0.1-SNAPSHOT.jar
+ instances:   1
+ memory:      1G
  routes:
+   employee-api-bright-zebra.cfapps.io

Creating app employee-api...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
 400.58 KiB / 400.58 KiB [==========================================================================] 100.00% 1
m28s

Waiting for API to complete processing files...
timeout connecting to log server, no log will be shown


Staging app and tracing logs...
   Downloading dotnet_core_buildpack_beta...
   Downloading staticfile_buildpack...
```

Pivotal

No More Tickets, Just Self-Service

cf push

Speed & Consistency

~45 seconds

Code Complete & Tested	
Find available hosts	2 Days
Install & configure runtime	1 Day
Install & configure middleware	1 Day
Pull application source code	¼ Day
Retrieve dependent libraries	¼ Day
Create application package	¼ Day
Install, configure dependent service(s)	2 Days
Deploy container to host(s)	½ Day
Load environment variables	¼ Day
Configure load balancer	2 Days
Configure firewalls	2 Days
Update service monitoring tools	3 Days
Configure log collector	1 Day
•••	•••
Application in Production	**~15+ Days**

Pivotal®

Transforming How The World Builds Software