Pivotal Cloud Foundry®

Buildpacks

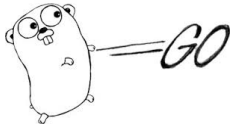
Pivotal

Buildpack Topics

- **Why Buildpacks?**
- Using
- Developer Configuration
- Administering
- API
- Customizing and Creating

Platform Flexibility

- Buildpacks provide an API to allow for the adoption of new languages and runtimes into the platform
  - Same basic operational and developer workflows
- The diverse landscape of languages and runtimes will continue to evolve



Developer Perspective- The cf push Philosophy

Onsi Fakhouri (Pivotal engineering):

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

Haiku

Buildpacks are a key part of making this possible

Buildpacks Make Operations Manageable

- Controls what frameworks/runtimes are used on the platform
- Provides consistent deployments across environments
  - Stops deployments from piling up at operation's doorstep
  - Enables a self-service platform
- Eases ongoing operations burdens:
  - Security vulnerability is identified
  - Subsequently fixed with a new buildpack release
  - Restage applications

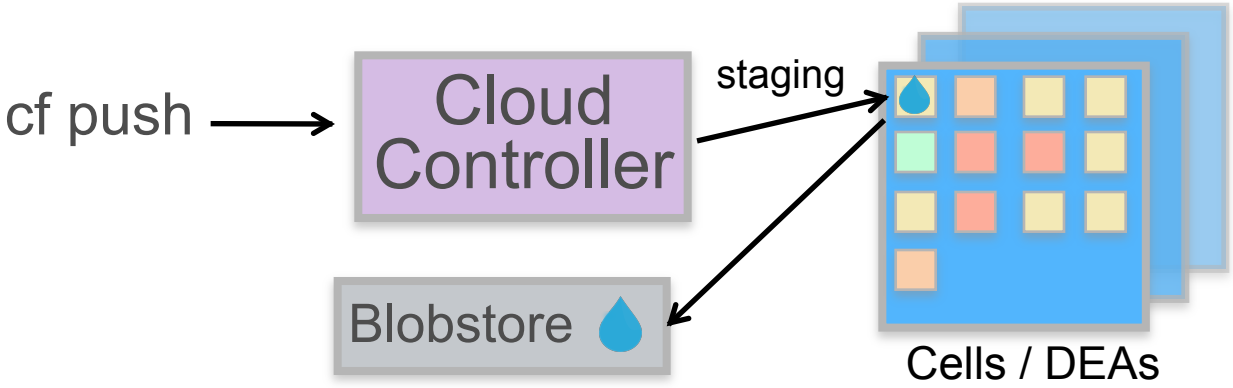
Pivotal

Buildpack Topics

- Why Buildpacks?
- **Using**
- Developer Configuration
- Administering
- API
- Customizing and Creating

What is a Buildpack?

- Enables Cloud Foundry to be language agnostic
  - Based on Heroku buildpacks
- Three **staging** scripts and their dependencies
  - Run inside of a staging container on Elastic Runtime
- Produces a droplet- a compressed archive for running an app instance in a container



The detect Script

- Inspects the pushed application to determine if the buildpack can handle the application

Ruby A Programmer's Best Friend	`Gemfile` exists?
node.js	`package.json` exists?
php	`.php` file exists?

Buildpack Detection

- Use **`cf buildpacks`** to view the available buildpacks
- The position column defines the order in which the detect scripts are run
- The first detect script that returns a `0` stops the detection process

my-php-app88 — bash — 115×14

```
greylag:my-php-app88 sbyrnes$ cf buildpacks
Getting buildpacks...

buildpack              position   enabled   locked   filename
ruby_buildpack         1          true      false    ruby-including-unsupported_buildpack-cached-v1.4.1.zip
nodejs_buildpack       2          true      false    nodejs-including-unsupported_buildpack-cached-v1.3.1.zip
java_buildpack         3          true      false    java-buildpack-v3.0.zip
go_buildpack           4          true      false    go-including-unsupported_buildpack-cached-v1.3.1.zip
liberty_buildpack      5          true      false    liberty_buildpack.zip
python_buildpack       6          true      false    python-including-unsupported_buildpack-cached-v1.3.1.zip
php_buildpack          7          true      false    php-including-unsupported_buildpack-cached-v3.2.1.zip
staticfile_buildpack   8          true      false    staticfile_buildpack-cached-v1.0.0.zip
binary_buildpack       9          true      false    binary_buildpack-cached-v1.0.0.zip
greylag:my-php-app88 sbyrnes$
```

Pivotal

Specifying a Buildpack

- Use the –b parameter when running **cf push** to avoid unnecessary buildpack file copying and detection
  - Can also specify the buildpack in the application manifest
- The buildpack parameter can name an installed buildpack or point to a custom buildpack in a git repository

```
$ cf push simplephpapp -b "php_buildpack"
Uploading simplephpapp...
Creating container
Downloading buildpacks (php_buildpack)...
Staging...
```

```
---
applications:
- name: myapp
  buildpack: php_buildpack
```

Pivotal

Custom Buildpacks

- You can specify custom buildpacks located in git repositories
  - Custom buildpacks can be disabled by the administrator (Ops Manager > Pivotal Elastic Runtime > Cloud Controller > Disable Custom Buildpacks)
- Here a custom fork of the php-buildpack is used…

```
$ cf push myphpapp -b
https://github.com/mygitaccount/php-buildpack

Downloading buildpacks (
https://github.com/mygitaccount/php-buildpack)...
Staging...
```

Pivotal

Buildpack Topics

- Why Buildpacks?
- Using
- **Developer Configuration**
- Administering
- API
- Customizing and Creating

Developer Configuration

- Implementations of buildpacks vary
- View the github repository for a specific buildpack to view configuration options

cloudfoundry-community / **cf-docs-contrib**

Code | Pull requests 0 | Wiki | Pulse | Graphs

Buildpacks

JT Archie edited this page on Oct 14 · 57 revisions

Buildpack information for Cloud Foundry v2

Custom buildpacks

List of available buildpacks

Built-in:

- https://github.com/cloudfoundry/java-buildpack
- https://github.com/cloudfoundry/ruby-buildpack
- https://github.com/cloudfoundry/nodejs-buildpack
- https://github.com/cloudfoundry/php-buildpack
- https://github.com/cloudfoundry/go-buildpack
- https://github.com/cloudfoundry/python-buildpack
- https://github.com/cloudfoundry/staticfile-buildpack
- https://github.com/cloudfoundry/binary-buildpack

Community created:

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

Pivotal

Example Developer Configuration- PHP

- Configure PHP applications with a `.bp-config/options.json` file in the application directory
- For example, you could enable mysqli extensions

.bp-config/options.json

```
{
    "PHP_EXTENSIONS": [ "mysqli"]
}
```

https://github.com/cloudfoundry/php-buildpack/blob/master/docs/config.md

Pivotal

Example Developer Configuration- Java

- Java buildpack configuration can be overridden with an environment variable matching the configuration file
  - Prefix the environment variable with JBP_CONFIG_ and drop the `.yml`
  - The variable value is inline YAML
- For example, you could change the default version of Java to 7:

https://github.com/cloudfoundry/java-buildpack/tree/master/config

open_jdk_jre.yml

```
---
applications:
- name: myapp
  buildpack: java_buildpack
  env:
    JBP_CONFIG_OPEN_JDK_JRE: '{jre: { version: 1.7.0_+ }}'
```

Java Buildpack Configuration

- Supports a variety of JVM languages, containers, and frameworks
- The buildpack's GitHub home page has links to configuration information

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

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

- Standard Containers
  - Dist ZIP
  - Groovy (Configuration)
  - Java Main (Configuration)
  - Play Framework
  - Ratpack
  - Spring Boot
  - Spring Boot CLI (Configuration)
  - Tomcat (Configuration)
- Standard Frameworks
  - AppDynamics Agent (Configuration)
  - Debug (Configuration)
  - DynaTrace Agent (Configuration)
  - Introscope Agent (Configuration)
  - Java Options (Configuration)
  - JRebel Agent (Configuration)
  - JMX (Configuration)
  - Luna Security Provider (Configuration)
  - MariaDB JDBC (Configuration)
  - New Relic Agent (Configuration)
  - Play Framework Auto Reconfiguration (Configuration)
  - Play Framework JPA Plugin (Configuration)
  - PostgreSQL JDBC (Configuration)
  - Spring Auto Reconfiguration (Configuration)

Buildpack Topics

- Why Buildpacks?
- Using
- Developer Configuration
- **Administering**
- API
- Customizing and Creating

Administering System Buildpacks

```
BUILDPACKS:
   buildpacks                          List all buildpacks
   create-buildpack                    Create a buildpack
   update-buildpack                    Update a buildpack
   rename-buildpack                    Rename a buildpack
   delete-buildpack                    Delete a buildpack
```

- **`cf buildpacks`** - lists all installed/system buildpacks
- **`cf create-buildpack <name> <path> <position>`**
  - **`<path>`** – local directory / zip file / URL / URL to zip file
  - **`<position>`** – relative order in buildpack list
  - **`--enable`** / **`--disable`**
- Administrator permissions required

Changing Buildpack Position

- Use cf update-buildpack to change a buildpack's detect position
- For example, if node.js apps are mostly pushed, an administrator can move it to position 1 with -i 1

```
greylag:simplephpapp sbyrnes$ cf update-buildpack -h
NAME:
   update-buildpack - Update a buildpack

USAGE:
   cf update-buildpack BUILDPACK [-p PATH] [-i POSITION] [--enable|--disable] [--lock|--unlock]

TIP:
   Path should be a zip file, a url to a zip file, or a local directory. Position is a positive integer,
sets priority, and is sorted from lowest to highest.

OPTIONS:
   -i             The order in which the buildpacks are checked during buildpack auto-detection
   -p             Path to directory or zip file
   --lock         Lock the buildpack to prevent updates
   --unlock       Unlock the buildpack to enable updates
   --enable       Enable the buildpack to be used for staging
   --disable      Disable the buildpack from being used for staging
```

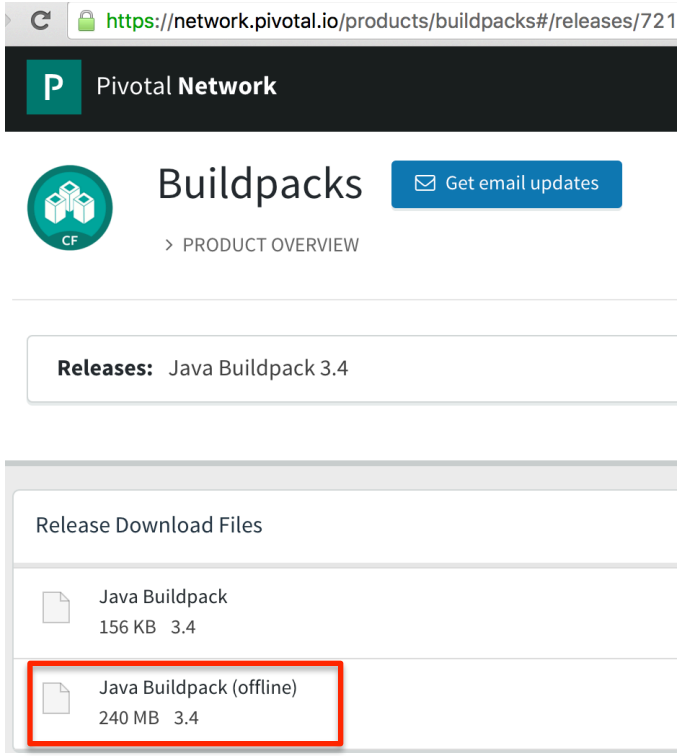
Locking Buildpacks

- Use cf update-buildpack --lock to prevent buildpack version updates
  - A way to control the production environment

```
greylag:simplephpapp sbyrnes$ cf update-buildpack -h
NAME:
   update-buildpack - Update a buildpack

USAGE:
   cf update-buildpack BUILDPACK [-p PATH] [-i POSITION] [--enable|--disable] [--lock|--unlock]

TIP:
   Path should be a zip file, a url to a zip file, or a local directory. Position is a positive integer,
sets priority, and is sorted from lowest to highest.

OPTIONS:
   -i             The order in which the buildpacks are checked during buildpack auto-detection
   -p             Path to directory or zip file
   --lock         Lock the buildpack to prevent updates
   --unlock       Unlock the buildpack to enable updates
   --enable       Enable the buildpack to be used for staging
   --disable      Disable the buildpack from being used for staging
```

Offline Buildpacks

- Builds droplets *without* internet connection
- http://network.pivotal.io contains offline buildpacks



Buildpack Topics

- Why Buildpacks?
- Using
- Developer Configuration
- Administering
- **API**
- Customizing and Creating

Scripting Languages

- Buildpacks are written in a scripting language
  - This is why the Java buildpack is not written in Java
- Bash- for simple buildpacks, such as Node.js
  - Can also call to other scripting languages- the PHP buildpack scripts call Python
- Ruby- for more involved buildpacks, such as Java

23

Script 1- bin/detect

- The detect script determines if the buildpack applies to the application being pushed
- Returns `0` and language information if the buildpack applies
- Returns `1` if the buildpack doesn't apply

Branch: **master** ▾ **nodejs-buildpack** / bin / **detect**

aemengo Output buildpack information in detect script [#100757

4 contributors

Executable File | 11 lines (8 sloc) | 161 Bytes

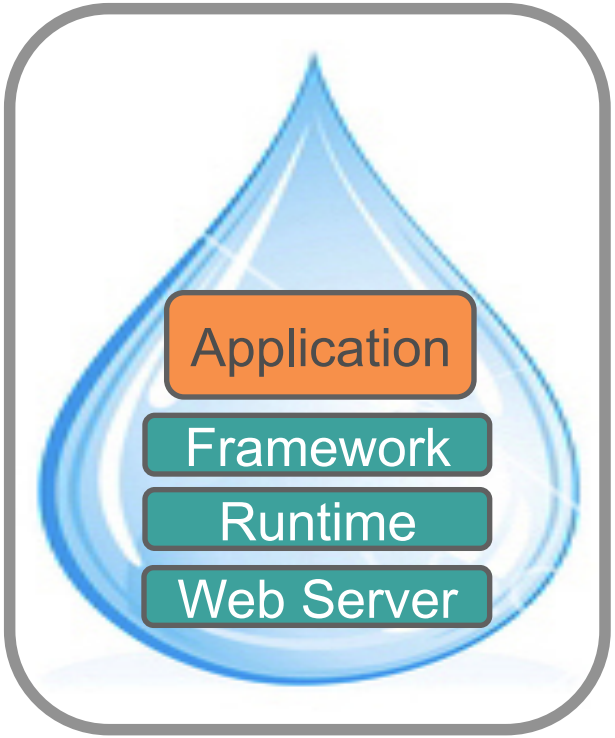
```
1  #!/usr/bin/env bash
2  # bin/detect <build-dir>
3
4  BP=$(dirname $(dirname $0))
5  if [ -f $1/package.json ]; then
6    echo "node.js `cat $BP/VERSION`"
7    exit 0
8  fi
9
10 exit 1
```

The node.js buildpack looks for a pushed file named `package.json`

24

Script 2- bin/compile

- The compile script is run after a detect script was successful
- Assembles an application with all of its runtime dependencies
  - Downloads, installs and configures dependencies such as a web server and a programming runtime
- Produces a droplet
- A non-zero return indicates the compile script failed



OS container

25

Example Compile Output- PHP Buildpack

- During **`cf push`**, you will see staging output
- You can also view this information with **`cf logs`**
- During PHP compile, you can see that the web server (httpd) and the PHP runtime are included

```
Staging...
-------> Buildpack version 4.3.0
Installing HTTPD
Downloaded [file:///tmp/buildpacks/
d5171dc06ed338008c7fdcb4eee474f0/dependencies/https___pivotal-
buildpacks.s3.amazonaws.com_concourse-
binaries_httpd_httpd-2.4.17-linux-x64.tgz] to [/tmp]
Installing PHP
…
Staging complete
```

Unsuccessful Buildpack Compilation

- When specifying a buildpack, the detect script is not run, but the compile script must successfully run
- Here is what happens when you push a PHP application and specify the go buildpack…

```
$ cf push myphpapp -b "go_buildpack”
Downloading buildpacks (go_buildpack)...
Staging...
 !      Godeps are required. For instructions:
 !      https://devcenter.heroku.com/articles/go-support
Failed to compile droplet
Staging failed: Exited with status 223

FAILED
BuildpackCompileFailed
```

Script 3- bin/release

- Simple script that provides the application's start command to the Cloud Controller database
  - For example, start a web server or execute a script
- The script writes YAML-formatted metadata to STDOUT
- On Cloud Foundry only the *web:* value is used- it specifies the start command for the app

```
---
  addons: []
  default_process_types:
    web: <start command>
```

Pivotal

Staging Container Lifecycle- Before Detect

- During `cf push` or `cf restage`, a staging container is created on a Cell
- Environment variables related to the app are included (e.g. by using `cf set-env`, specifying in the manifest, or by binding services)
- `<app_directory>` is created by Cloud Foundry- sometimes called the build directory
- The files from the pushed application are placed in `<app_directory>`
- System buildpacks are added (if no buildpack was specified)

Staging Container

- <app_directory>/pushedfiles
- buildpacks
- environment variables

Pivotal

29

Staging Container Lifecycle- Detect

- A buildpack's bin/detect script is executed
  - If the –b flag is used to specify a buildpack, the detect script is not run
- Cloud Foundry passes the **`<app_directory>`** as an argument to the script
- The script uses that argument to help determine if the application matches
- If so, the script returns successful (0)

Staging Container

- <app_directory>/pushedfiles
- buildpacks
- environment variables

```
bin/detect app_directory
```

Pivotal

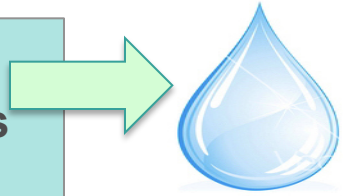
30

Staging Container Lifecycle- Compile

- Cloud Foundry passes **`<app_directory>`** as an argument to the script
- The compile script adds any dependencies to **`<app_directory>`**
  - The script can use environment variables- e.g. if New Relic is a bound service, add the agent to **`<app_directory>`**
- The contents of **`<app_directory>`** are packaged as a droplet tarball
- Cloud Foundry creates a **`<cache_directory>`** and passes it as the second argument to the script
  - The buildpack can cache staging files used for the life of the application- speeds subsequent stages

Staging Container

- <app_directory>/pushedfiles
  /dependencies
- <cache_directory>
- environment variables



```
bin/compile app_directory cache_directory
```


31

Summary: Buildpack API

- `/bin/detect app_directory`
  - Inspects the application to determine buildpack applicability
- `/bin/compile app_directory cache_directory`
  - Download and install runtime, web server, packages, libraries
  - The final app_directory is packed as a droplet
- `/bin/release app_directory`
  - Contains the application's start command- passed to the Cloud Controller database

The buildpack API is open-ended. If you can script it, you can do it.

Pivotal

Buildpack Topics

- Why Buildpacks?
- Using
- Developer Configuration
- Administering
- API
- **Customizing and Creating**

Configuration, Customization, Extension

- Most buildpacks support configuration
  - We have seen examples for staticfile, PHP and Java
  - This is recommended because it does not involve forking
- You can **customize** a buildpack, which involves forking
- Some buildpacks support **extension**, which is a form of customization where the core buildpack is not altered
  - Examples include Java and PHP

For more information on configuring, customizing or extending a particular buildpack, check its GitHub repository.

Pivotal

Custom Buildpacks

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

Microsoft .net

jetty://

IBM Liberty

Review- Buildpack Topics

- Why Buildpacks?
- Using
- Developer Configuration
- Administering
- API
- Customizing and Creating

Lab- Explore, configure and update a buildpack