Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Cloud Foundry - Buildpacks

Objectives of CF - Buildpacks

- Purpose:
  - To learn pivotal cloud foundry BuildPack.
- Product:
  - Introduction to Buildpacks
    - What are Buildpacks?
    - Deploying to Cloud Foundry
    - Using Buildpacks
  - Customizing Buildpacks
    - Buildpack API
    - Java Buildpack
    - Configure / Extend Java Buildpack
- Process:
  - To learn configure buildpacks and customize buildpacks in Cloud Foundry.

Table of Contents

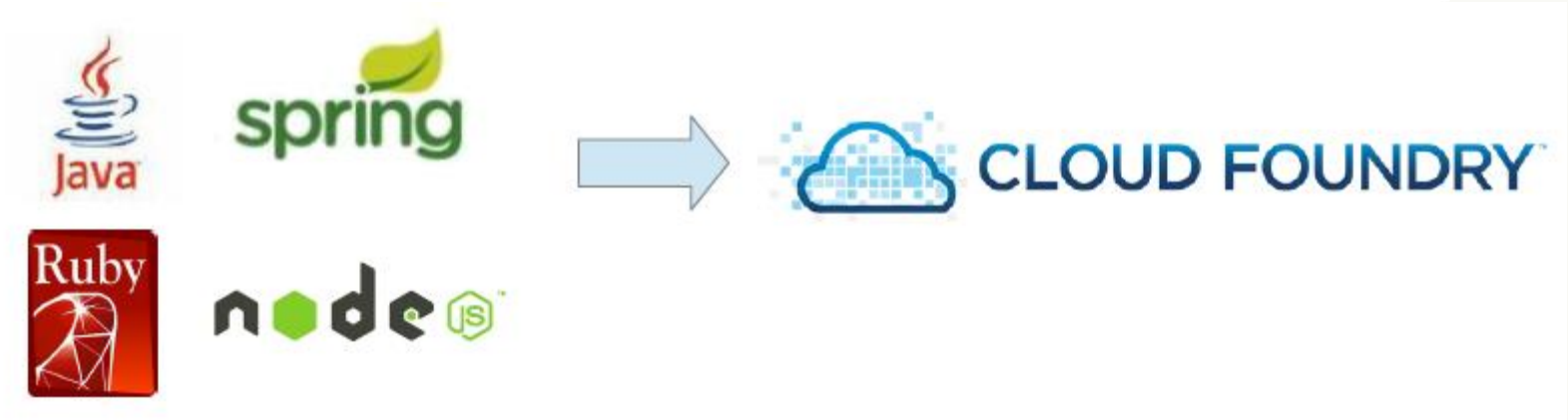
INTRODUCTION TO BUILDPACKS

What are Buildpacks?

- Applications
  - Consists of source code and application frameworks used by developers to create application
    - Java/Spring
    - Ruby/Rails
    - Java Script for Node.js
    - …

What are Buildpacks?

- The Question:
  - Applications can be written in many languages / frameworks:



  - …and yet each type can run in Cloud Foundry
  - How is this possible?

Configuring a Server from scratch

- If you were configuring a new server to run an application, what would you include / install ?

  - Operating system
  - Runtimes for your software (Java, Ruby, Python, etc .)
  - Containers as needed (e.g. Tomcat for Java, Apache HTTPD for PHP)
  - Frameworks as needed (APM tools)
  - Application binaries
- Good idea to write a script to do this.

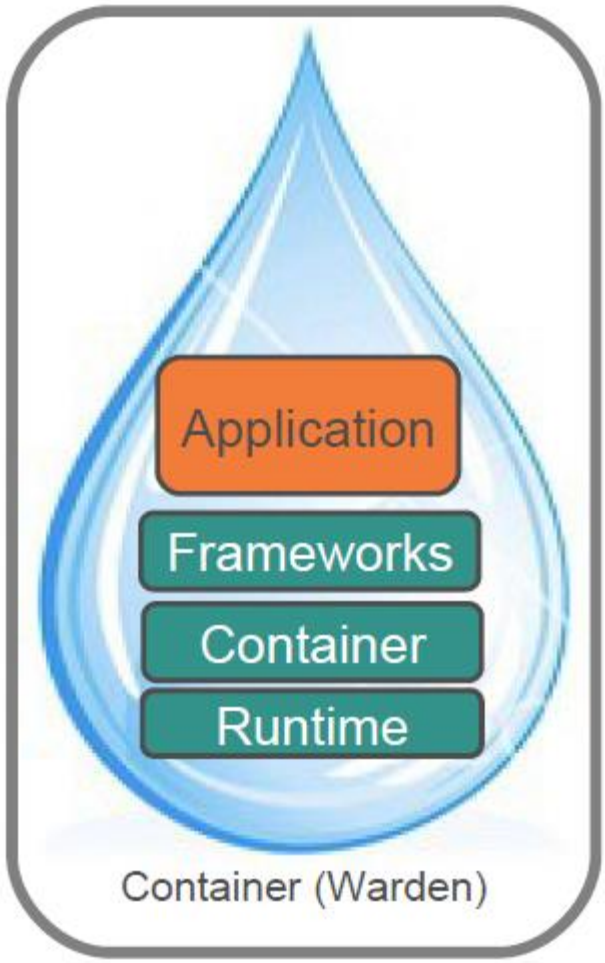
A Buildpack Does the Same Thing

Except the goal is run on Cloud Foundry

Buildpack – a combination of scripts that assembles runtimes, containers, frameworks, and your application into a droplet

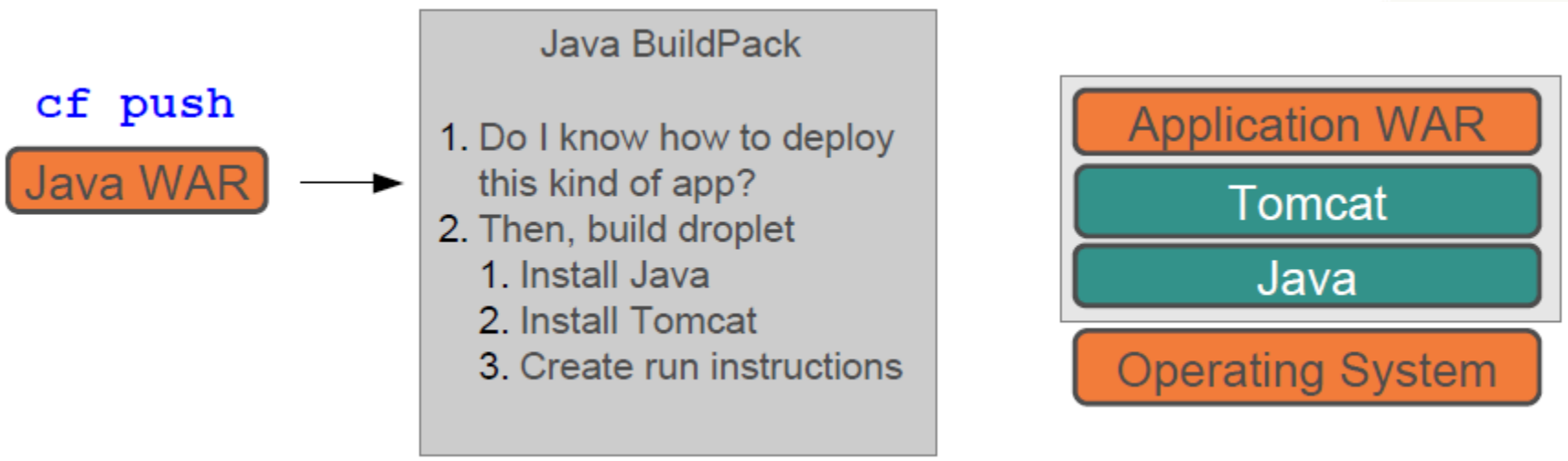
Droplets run inside Warden Containers, which run inside DEAs

- Fully explained in the architecture session.



The Answer : Buildpacks

- **Buildpacks** define how assemble a droplet to run a specific kind of application
- Example:



- The Buildpack “builds” the “droplet” to run an app.
  - Called staging the application

Buildpacks are Not…

- Buildpacks …
  - Are not a special build process for our application
    - Buildpacks build droplets
  - Do not run on our local machine
    - Buildpacks run on CF during the staging process

Buildpack Structure

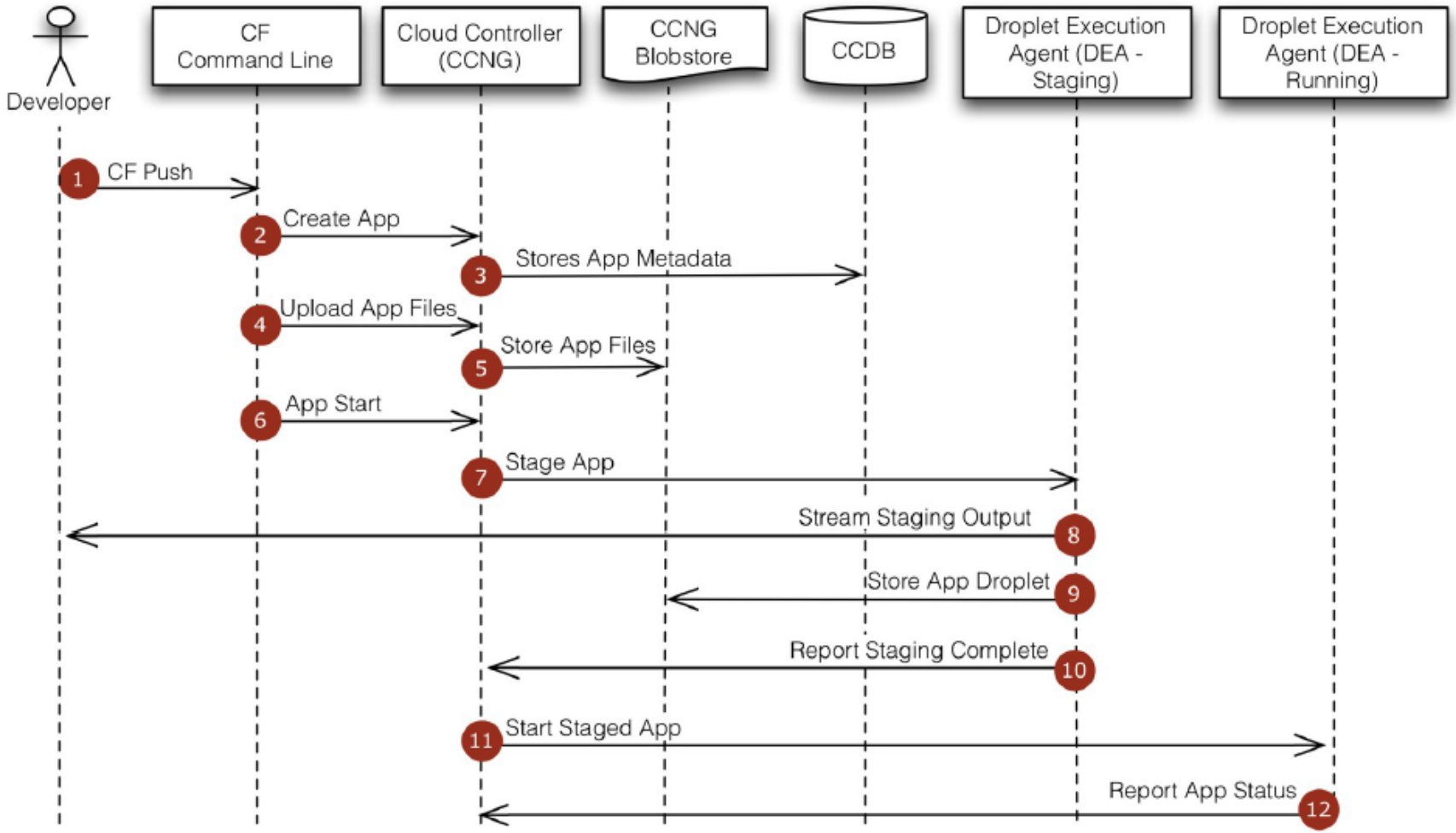
- Often written as Ruby script with three parts:
  - **Detect** if the buildpack should be applied
  - **Compile** (pack) the Droplet by combining the application code with runtimes, frameworks, plug-ins etc. necessary for the application
  - **Release** the app to be deployed to an assigned DEA

NOTE : *Assemble or pack would be a better name than Compile*

No code compilation is happening

DEPLOYING TO CLOUD FOUNDRY

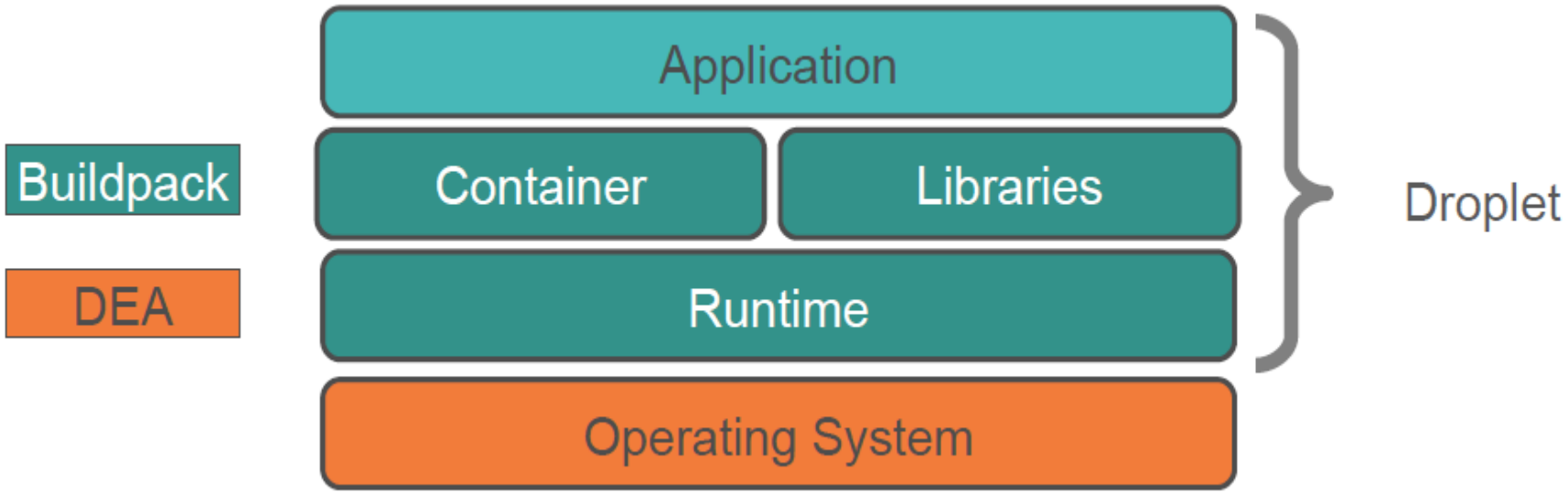
Deploying to CF



All work described was performed by Capgemini or a Capgemini affiliate

Staging and Buildpacks

- Build packs are responsible for preparing the machine image for an application



Available Buildpacks

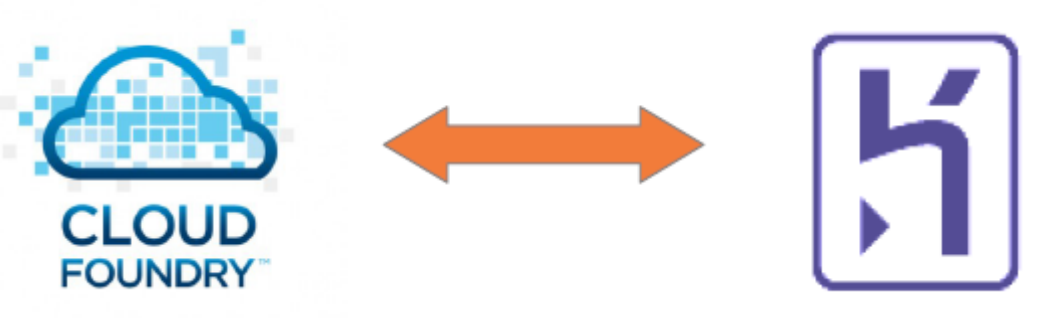
- Buildpacks are either
  - Installed into a cloud foundry instance or
  - Loaded from an external location at push time
- Buildpacks provided by public Cloud Foundry
  - Note: This list expands over time!

Custom Buildpacks

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

Compatibility

- Buildpacks can be compatible with multiple PaaS offerings
- CF buildpacks follow the Heroku buildpack design
  - CF and Heroku buildpacks are compatible (if we care to make them compatible)
  - Other PaaS offerings adopting the buildpack design



USING BUILDPACKS

Built-In Buildpacks

- Use **cf buildpacks** to determine installed buildpacks

```
> cf buildpacks
Getting buildpacks...

buildpack          position  enabled  locked  filename
ruby_buildpack     1         true     false   ruby_buildpack-offline-v1.0.1.zip
nodejs_buildpack   2         true     false   nodejs-buildpack-offline-b29.zip
java_buildpack     3         true     false   java-buildpack-v2.4.zip
go_buildpack       4         true     false   go_buildpack-offline-v1.0.1.zip
liberty_buildpack  5         true     false   liberty_buildpack.zip
python_buildpack   6         true     false   python_buildpack-offline-v1.0.1.zip
php_buildpack      7         true     false   php_buildpack-offline-v1.0.1.zip
```

Managing Built-In Buildpacks

- **$> cf create-buildpack <name> <path> <order>**
  - **<path>** - local directory/ zip file /URL /URL to zip file
  - **<order>** - relative order in buildpack list
  - **--enable** / **--disable**
- Commands for update, delete, rename available
- Administrator permissions required

Automatic Detection / Explicit Reference

- **$> cf push**
  - Application checked against pre-defined buildpacks
  - Matching buildpack invoked automatically
- **$> cf push –b <buildpack-name>**
  - Desired buildpack specified (installed buildpack)
- **$> cf push –b <url>**
  - The desired buildpack is referenced by a Git URL
    - Note: “disable custom buildpacks” disables this option

Specify within manifest

- Use buildpack element
  - Specify name or URL

```
---
applications:
- name: cf-my-app
  host: cf-my-app
  domain: cfapps.io
  path: target/my-war.war
  buildpack: https://github.com/cloudfoundry/java-buildpack
```

- Remember precedence
  - Options specified in push command override manifest

Pushing an Executable

- Suppose I have a binary executable?
  - Such as a script or statically compiled C,C++ application
  - *Must* be compiled for x86 Linux
- Then I can push and run it using the “null” buildpack

```
$> ./bin/hello
Hello World
$> cf push hello --no-route -p bin -m 256m -c "./bin/hello"
       -b http://github.com/ryandotsmith/null-buildpack.git
... usual push logging ...
Hello World
$>
```

CUSTOMIZING BUILDPACKS

Buildpack API

Buildpack API

- /bin/detect app_directory
  - Inspect application bits to determine builbpack applicability
- /bin/compile app_directory cache_directory
  - Download and install runtime, container, packages, libraries; install application bits as necessary
- /bin/release app_directory
  - Build application start command

/bin/detect

	Gemfile **exists?**
	package.json **exists?**
	setup.py **exists?**

/bin/compile

- ‘Builds’ the Droplet
- Downloads and installs any necessary runtime
  - Java VM, Ruby interpreter, JavaScript interpreter …
  - Container or web server
  - Support libraries, packages, modules
    - Java jars, Ruby gems, NPM packages
- Then install the app bits into the runtime or container

/bin/compile caching

- Runtime, container, and support packages are often downloaded from sources external to Cloud Foundry
  - Depending on the buildpack
- DEA provides a location for storing downloaded artifacts to speed subsequent staging operations

/bin/release

- Builds a YAML-formatted hash with three possible keys

- On Cloud Foundry (currently) only the *web:* value is used to get the start command for the app

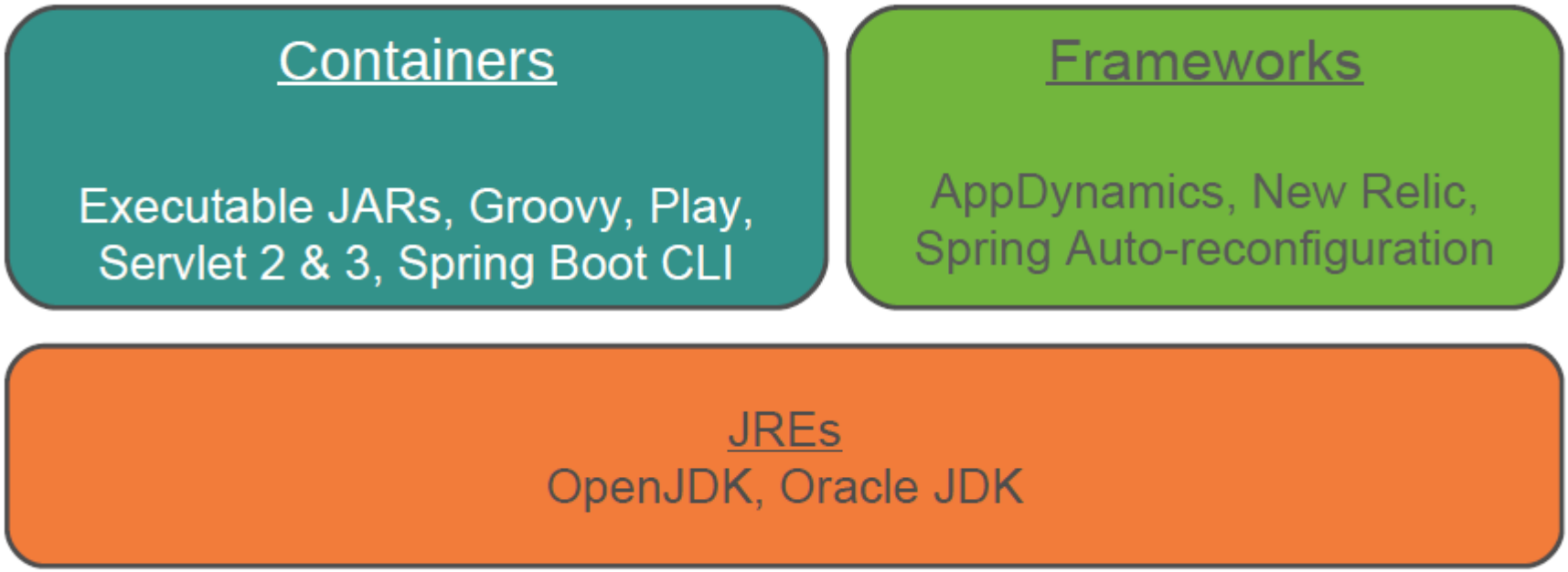
```
addons:        [ ]
config_vars:      { }
Default_process_types :
        web: <start command>
```

Java Buildpack

- Supports variety of JVM languages, containers, and frameworks with a modular, configurable, and extensible design

Java Buildpack Concepts

Java Buildpack Concepts



Container Detection Criteria

Java *main()*	META-INF/MANIFEST.MF **exists with** Main-Class **attribute set**
Tomcat	WEB-INF **directory exists**
Groovy	.groovy **file with a** main() **method, or** .groovy **file with no classes, or** .groovy **file with a shebang** (#!) **declaration**
Spring Boot CLI	**One or more POGO** .groovy **files with no** main() **method, and no** WEB-INF **directory**
Play	start **and** lib/play.play_*.jar **exist**

Framework detection criteria

App Dynamics	App Dynamics service bound to app
New Relic	New Relic service bound to app
Spring AutoConfiguration	spring-core*.jar in the application directory

/bin/compile Output Example

- Output example:

```
-----> Downloaded app package (11M)

-----> Downloading Open Jdk JRE 1.8.0_20 from
http://download.run.pivotal.io/openjdk/lucid/x86_64/open
jdk-1.8.0_20.tar.gz (1.0s)

       Expanding Open Jdk JRE to .java-
buildpack/open_jdk_jre (1.1s)

-----> Downloading Tomcat Instance 7.0.53 from
http://download.run.pivotal.io/tomcat/tomcat-
7.0.53.tar.gz (0.5s)

       Expanding Tomcat to .java-buildpack/tomcat (0.1s)

-----> Uploading droplet (49M)
```

See what's Going On

- **$> cf files <app-name> app**

  - java-buildpack.log
  - java-buildpack

    open_jdk_jre

    spring_auto_reconfiguration

    tomcat

    …
  - META-INF/
  - WEB-INF/

Log output from buildpack

Sandboxes for each component used during staging

Configure / Extend Java Buildpack

Customization

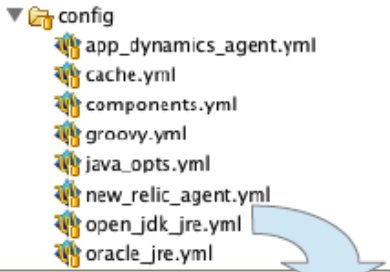
- You may alter Java buildpack
  - Configure artifacts used by standard JREs, Containers, and Frameworks
  - Extend the buildpack with your own JREs, Containers, and Frameworks

- Customization is done by forking the buildpack

- … or Simply downloading, modifying and zipping.

Customizing Configuration

- Most configuration options found in / **config**
  - determine behavior of a JRE, Container or Framework




```
---
repository_root: "{default.repository.root}/openjdk/{platform}/{architecture}"
version: 1.8.0_+
memory_sizes:
  metaspace: 64m..
memory_heuristics:
  heap: 75
  metaspace: 10
  stack: 5
  native: 10
```

repository_root and *version* typically at the top of each file.

Locating Downloads

- URLs derived from repository root
  - { default.repository.root } /openjdk/ { platform} / { architecture}
  - download.pivotal.io.s3.amazonaws.com / openjdk / lucid /x86_64
  - **index.yml** holds location of each version

```
# http://download.pivotal.io.s3.amazonaws.com/openjdk/lucid/x86_64/index.yml
---
1.8.0_25: https://download.run.pivotal.io/.../x86_64/openjdk-1.8.0_25.tar.gz
1.7.0_71: https://download.run.pivotal.io/.../x86_64/openjdk-1.7.0_71.tar.gz
1.8.0_31: https://download.run.pivotal.io/.../x86_64/openjdk-1.8.0_31.tar.gz
1.7.0_75: https://download.run.pivotal.io/.../x86_64/openjdk-1.7.0_75.tar.gz
1.8.0_40: https://download.run.pivotal.io/.../x86_64/openjdk-1.8.0_40.tar.gz
...
```

Customization by Configuration : Tomcat

- Example: customizing the Tomcat artifact for download

```
# cloudfoundry/java-buildpack/config/tomcat.yml
---
tomcat:
  version: 8.0.+
  repository_root: "{default.repository.root}/tomcat"
...
```

```
# http://files.example.com/tomcat-custom/index.yml
---
8.0.18: https://download.run.pivotal.io/tomcat/tomcat-8.0.18.tar.gz
8.0.17: https://download.run.pivotal.io/tomcat/tomcat-8.0.17.tar.gz
7.0.59: https://download.run.pivotal.io/tomcat/tomcat-7.0.59.tar.gz
8.0.20: https://download.run.pivotal.io/tomcat/tomcat-8.0.20.tar.gz
```

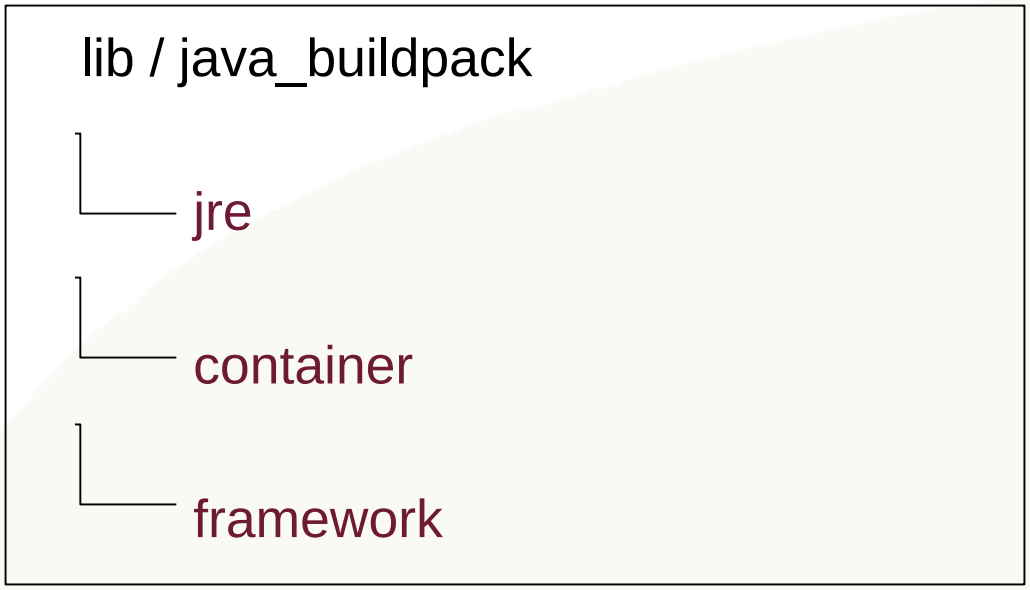
Resource Configuration

- Tomcat container supports simple customization of context.xml and server.xml
  - Files will overlay sandbox provided values.

- Not just for Tomcat
  - JDK, New Relic, etc.

Extending the Buildpack - 1

- You can extend the Java Buildpack
  - To add different JRE, Container or Framework
- Implement support class (Ruby) in the appropriate directory
  - With additional support classes as necessary



Extending the Buildpack - 2

- Support class types have similar interfaces, following the buildpack scripts naming conventions

```
# Return String or an Array<String> that identifies the component to be
# used in staging, or nil.
def detect

# Modifies the application's file system.  Component is expected to
# transform the application's file system in whatever way is necessary
(e.g. downloading files or creating symbolic links) to support the function
of the component.  Status output written to STDOUT is expected.
def compile

# Modifies the application's runtime configuration to support the function
# of the component.  Create the command required to run the application,
# taking context values into account when creating the command.  Container
# components are expected to return the command required to run the application.
def release
```

Extending the Buildpack - 3

- Add new support class to **config/components.yml**

```
# Configuration for components to use in the buildpack
---
containers:
  - "JavaBuildpack::Container::DistZip"
  - "JavaBuildpack::Container::Groovy"
  - "JavaBuildpack::Container::JavaMain"
  - "JavaBuildpack::Container::PlayFramework"
  - "JavaBuildpack::Container::Ratpack"
  - "JavaBuildpack::Container::SpringBoot"
  - "JavaBuildpack::Container::SpringBootCLI"
  - "JavaBuildpack::Container::Tomcat"
  - "JavaBuildpack::Container::YOUR-CONTAINER-HERE"

jres:
  - "JavaBuildpack::jre::OpenJdkJRE"
# - "JavaBuildpack::jre::OracleJRE"

frameworks:
  - "JavaBuildpack::Framework::AppDynamicsAgent"
...
```

...or here if JRE...

...or here if framework.

More on Customization

- Much more information and documentation included in the GitHub repository

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

Customization without Forking

- Simple customization of properties can be done without forking the buildpack
  - Set environment variables instead
    - Either using **cf set-env** or in the **env:** section of manifest
- Three options:
  - JAVA_OPTS variable
  - JBP_CONFIG variable

Change JVM Runtime Options - I

- The JAVA_OPTS variable is recognized when app runs:

```
$> cf set-env spring-music JAVA_OPTS -showversion
Setting env variable JAVA_OPTS -showversion spring-music myorg
development jlee@pivotal.io
OK
TIP: Use 'cf restage' to ensure your env variable changes take effect

$> cf restage spring-music
... usual push output ...
2015-04-10T16:45:11.88 [App/0] ERR openjdk version "1.8.0_40-"
2015-04-10T16:45:11.88 [App/0] ERR OpenJDK Runtime Environment (build
1.8.0_40--vagrant_2015_03_26_09_03-b25)
2015-04-10T16:45:11.88 [App/0] ERR OpenJDK 64-Bit Server VM (build
25.40-b25, mixed mode)
...
$>
```

Change JVM Runtime Options - II

- Most JVM options can be specified this way
  - Except some that govern memory sizing
    - Such as –Xms, -Xmx, - Xss, -XX:maxPermSize,
      XX: MarketaspaceSize, -XX:MetaspaceSize,
      -XX:PermSize
  - Most other –XX options can be used
  - For full details see:

  - https://github.com/cloudfoundry/java-buildpack/blob/master/docs/framework-java_opts.md

JBP_CONFIG variables

- Use environment variable to override a buildpack configuration file
  - Naming convention used:
    - **my_file.yml→JBP_CONFIG_MY_FILE**
  - Variable must be set to valid inline YAML syntax

- To change default version of Java to 7
  - Override **open_jdk_jre.yml**

```
>$ cf set-env my-application JBP_CONFIG_OPEN_JDK_JRE '[version:
1.7.0_+, memory_heuristics: {heap: 85, stack: 10}]'
```

Offline Buildpacks

- Wish to avoid download buildpacks from Internet
- Java buildpack can be packages as *offline* Buildpack
  - Builds droplets *without* internet connection
- One – time build process
  - Internally packages latest version of each dependency within the buildpack
  - Disables remote downloads
  - About 180M in size
  - Install using cf create-buildpack/update-buildpack
  - https://github/cloudfoundry/java-buildpack/blob/master/docs/buildpack-modes.md
- **Note**: Pivotal CF ships with offline buildpacks!

Recap

detect

/bin/compile

java buildpacks

cf set-env

forking

offline

Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

www.capgemini.com