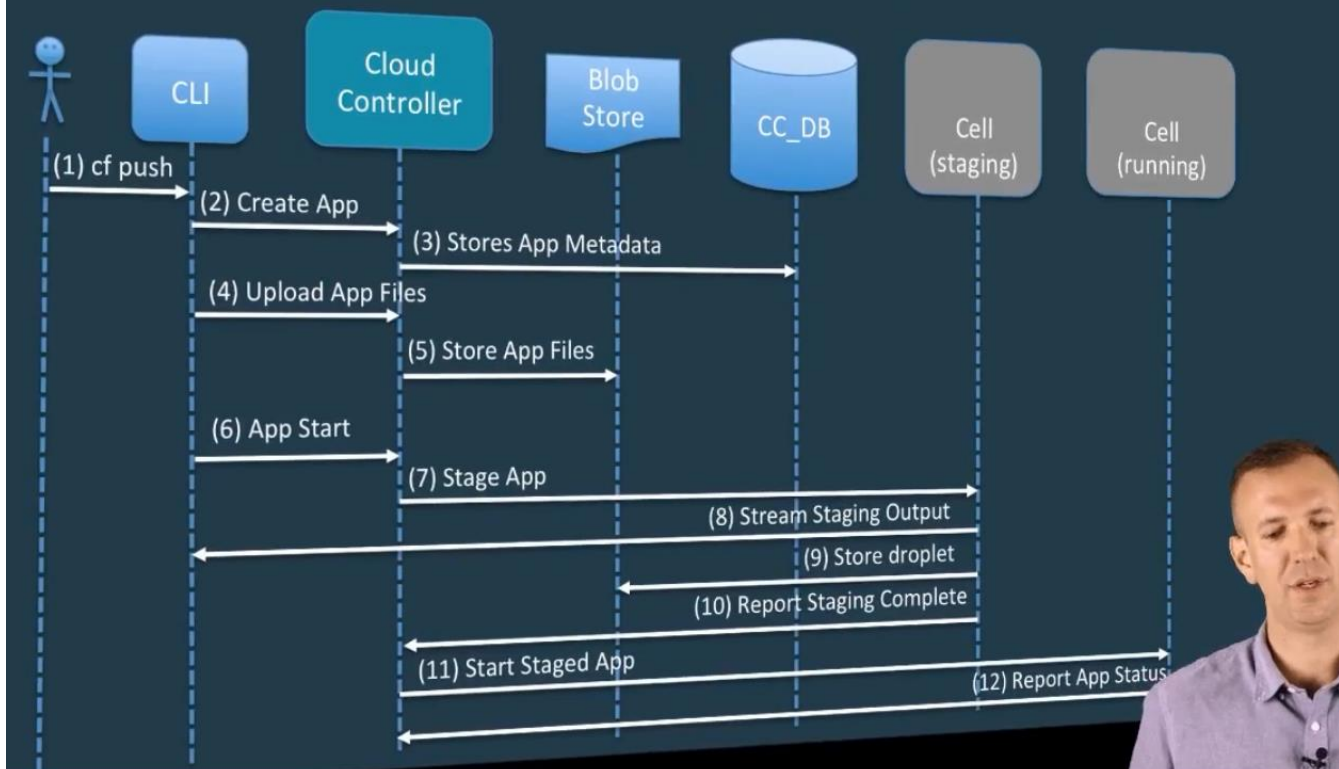


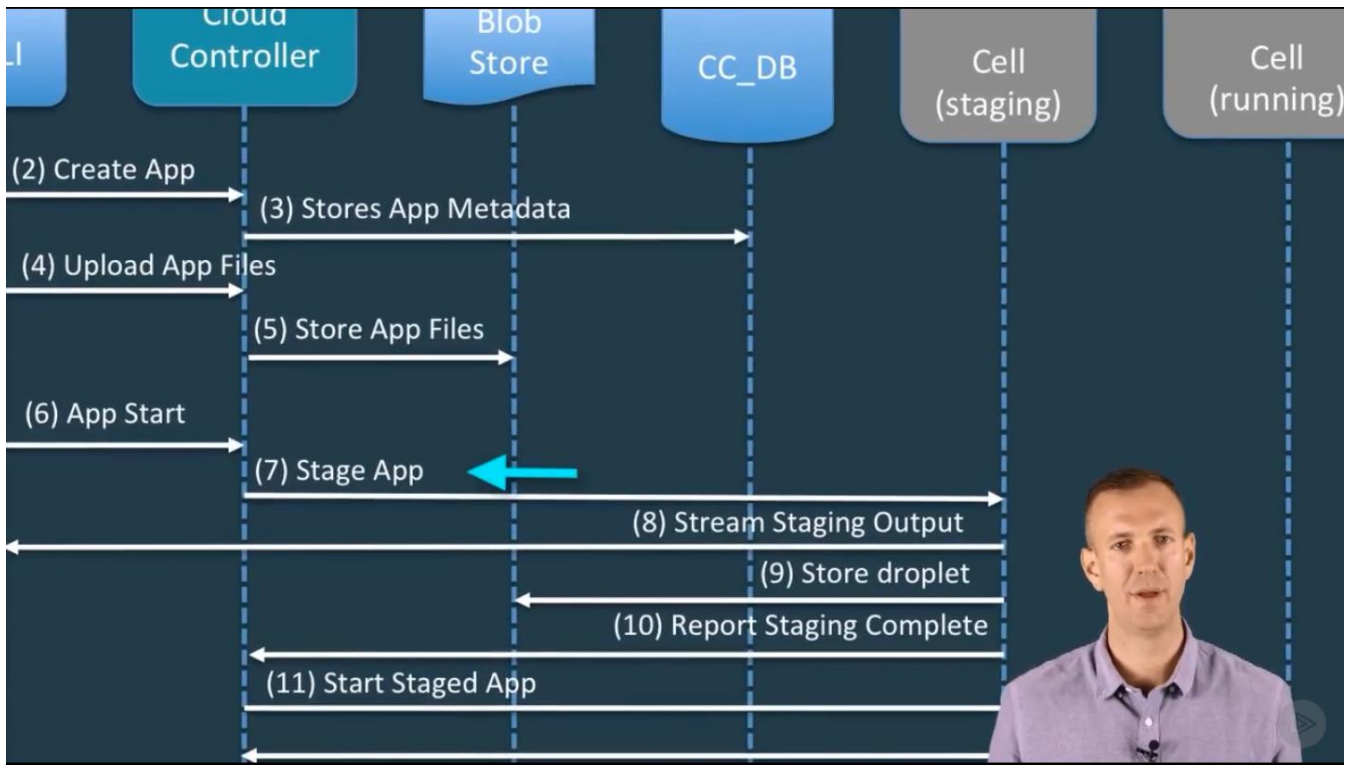
Buildpack

Agenda

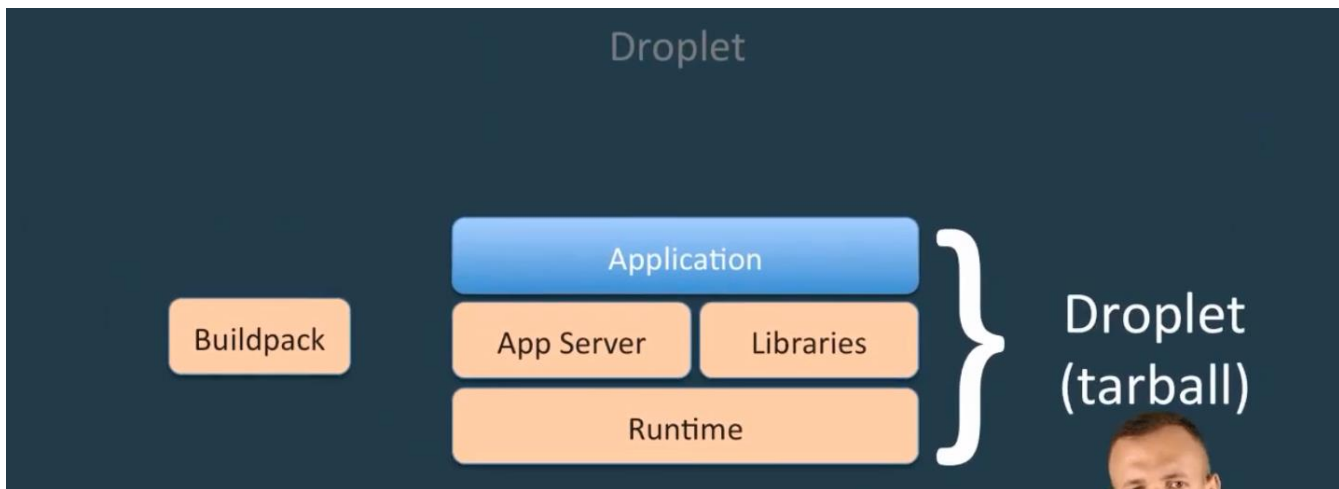
1. Review Staging
2. Buildpack API

Review Pushing An App

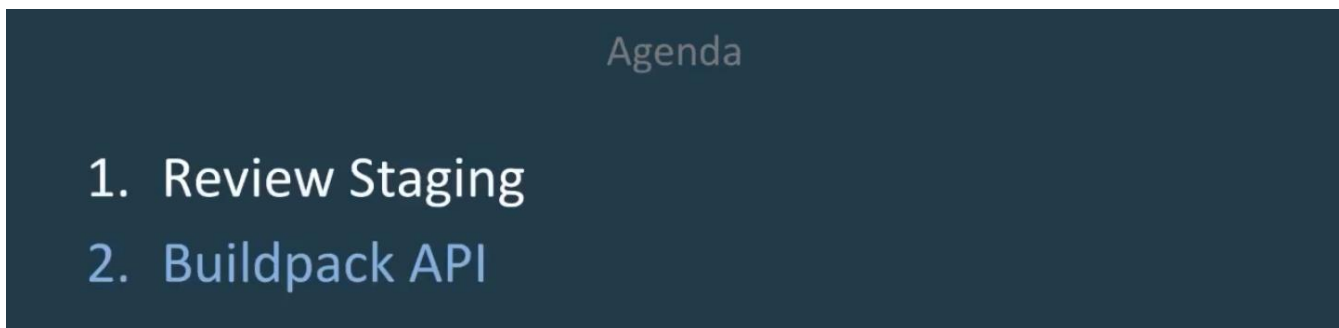




The **Stage App** step in the sequence diagram is where most of the work happens when it comes to buildpacks.



This ends up in the creation of the droplet that includes your application and all of the runtime support it needs to run.



Buildpack API

`bin/detect`

Determines whether the buildpack can stage the application

`bin/compile`

Builds the droplet

`bin/release`

Provides information on how to run the application

There are 3 scripts that get run for implementing the Buildpack API, they can be implemented in any language that you want depending on how complex your build steps are.

`bin/detect`

Detection criteria:

Ruby	Gemfile exists
Node	Package.json exists
Python	Setup.py exists

The buildpack looks for some specific file in order to decide whether it can stage the application for you or not.

`bin/detect`

```
$ cf push
```

Cell iterates over all admin and system buildpacks calling:

```
bin/detect scripts
```

until one returns exit code 0

```
$ cf push -b <url|name>
```

```
bin/detect
```

is not called.

When you do a **`$ cf push`** command and don't specify what buildpack to use, the cell is going to iterate over all the system and admin buildpacks in PCF and run checks to see which one can build your droplet.

bin/compile

Provide any of the following if needed:

Runtime	Java VM, Ruby Interpreter, JavaScript Interpreter
App Server	Tomcat, Nginx, WEBrick
Support Libraries	Ruby gems, NPM packages, A...ents

The compile script helps to build the actual droplet.

bin/release

Output a YAML document in the following form:

```
config_vars:  
  name: value  
default_process_types:  
  web: <start command>
```

Only the **web** value is used to get the start command for the app

Let us now see a brief demo of using buildpacks

```
DROBERTS-MBPRO:articulate droberts$ cf app articulate
Showing health and status for app articulate in org dave / space dev as droberts@pivotal.io...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io
last uploaded: Tue Jun 7 21:21:03 UTC 2016
stack: cflinuxfs2
buildpack: java-buildpack=v3.6-offline-https://github.com/cloudfoundry/java-buildpack.git#5194155 java-main open-jdk-jre=1.8.0_71 open-jdk-like-memory-calculator=2.0.1_RELEASE spring-auto-reconfiguration=1.10.0_RELEASE

#0 state since cpu memory disk details
running 2016-06-08 09:25:44 AM 0.1% 408.6M of 512M 154.2M of 1G
DROBERTS-MBPRO:articulate droberts$
```

We can see information about the articulate application using the **\$ cf app articulate** command as above. We can see that the articulate app is using the **java-buildpack version 3.6**


```
DROBERTS-MBPRO:articulate droberts$ cf buildpacks
Getting buildpacks...

buildpack      position  enabled  locked  filename
staticfile_buildpack  1        true    false   staticfile_buildpack-cached-v1.3.6.zip
java_buildpack_offline 2        true    false   java_buildpack-offline-v3.6.zip
ruby_buildpack  3        true    false   ruby_buildpack-cached-v1.6.16.zip
nodejs_buildpack 4        true    false   nodejs_buildpack-cached-v1.5.11.zip
go_buildpack     5        true    false   go_buildpack-cached-v1.7.5.zip
python_buildpack 6        true    false   python_buildpack-cached-v1.5.5.zip
php_buildpack    7        true    false   php_buildpack-cached-v4.3.10.zip
binary_buildpack 8        true    false   binary_buildpack-cached-v1.0.1.zip
DROBERTS-MBPRO:articulate droberts$
```

We can run the **\$ cf buildpacks** command to see the list of available buildpacks for use, we also have the option of using custom buildpacks that suit our needs.

The screenshot shows the Articulate application interface in a web browser. The page has a green header with the Articulate logo and navigation links: Articulate, Scale & HA, Services, Blue-Green, and Spring Boot. The main content area is titled "Welcome to Articulate!" and contains several sections:

- Application Environment Information:** A sidebar on the right containing:
 - Application Name: articulate
 - Instance Index: 0
 - Container Address: 10.254.0.6:8080
 - Cell Address: 10.65.188.46:60324
 - Java Version: 1.8.0_71 (highlighted with a red arrow)
 - Services: user-provided: attendee-service
 - Description
 - The 12 Factor App
- Application Architecture:** A section with a diagram showing the application stack:
 - articulate** (Spring Boot, Spring MVC, Spring Cloud Connectors, Thymeleaf, Bootstrap) connects via **REST (http/json)** to **attendee-service** (Spring Boot, Spring Data JPA, Spring Cloud Connectors), which in turn connects to **MySQL**.
- How to use this Application:** A section with text explaining the application's purpose and how to use it, including a list of menu items and their descriptions.

A video player is visible in the bottom right corner of the screenshot, showing a man speaking.

Let us now specify a new custom buildpack that will allow us to do different things

```
DROBERTS-MBPRO:articulate droberts$ cf push articulate -p ./articulate-0.0.1-SNAPSHOT.jar -b https://github.com/cloudfoundry/java-buildpack.git
```

```
RTS-MBPRO:articulate droberts$ cf push articulate -p ./articulate-0.0.1-SNAPSHOT.jar -b https://github.com/cloudfoundry/java-buildpack.git
```

We push the articulate service again but now use the `-b` flag that allows us to deploy an application using a custom buildpack that we specify, we can specify a GitHub repo where the custom buildpack exists.

```
DROBERTS-MBPRO:articulate droberts$ cf push articulate -p ./articulate-0.0.1-SNAPSHOT.jar -b https://github.com/cloudfoundry/java-buildpack.git
Updating app articulate in org dave / space dev as droberts@pivotal.io...
OK
Uploading articulate...
Uploading app files from: /var/folders/84/ldbx2c5j01l_ycgg3d37g9yh0000gq/T/unzipped-app419671338
Uploading 769.5K, 139 files
Done uploading
```

```
Starting app articulate in org dave / space dev as droberts@pivotal.io...
Creating container
Successfully created container
Downloading app package...
Downloaded app package (27.6M)
Downloading build artifacts cache...
Downloaded build artifacts cache (88M)
Staging...
-----> Java Buildpack Version: 29c79f2 | https://github.com/cloudfoundry/java-buildpack.git#29c79f2
-----> Downloading Open Jdk JRE 1.8.0_91-unlimited-crypto from https://download.run.pivotal.io/openjdk/trusty/x86_64
jdk-1.8.0_91-unlimited-crypto.tar.gz (found in cache)
Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.2s)
-----> Downloading Open JDK Like Memory Calculator 2.0.2_RELEASE from https://download.run.pivotal.io/memory-calculator-2.0.2_RELEASE.tar.gz (found in cache)
Memory Settings: -Xms317161K -XX:MaxMetaspaceSize=64M -Xss228K -Xmx317161K -XX:MetaspaceSize=64M
-----> Downloading Spring Auto Reconfiguration 1.10.0_RELEASE from https://download.run.pivotal.io/auto-reconfiguration-1.10.0_RELEASE.jar (found in cache)
Exit status 0
Staging complete
Uploading droplet, build artifacts cache...
Uploading build artifacts cache...
Uploading droplet...
Uploaded build artifacts cache (88M)
```

```
App articulate was started using this command `CALCULATED_MEMORY=$(($PWD/.java-buildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-2.0.2_RELEASE -memorySizes=metaspace:64m.,stack:228k.. -memoryWeights=heap:65,metaspace:10,native:stack:10 -memoryInitials=heap:100%,metaspace:100% -stackThreads=300 -totMemory=$MEMORY_LIMIT) && JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh $CALCULATED_MEMORY" && SERVER_PORT=$(eval exec $PWD/.java-buildpack/open_jdk_jre/bin/java $JAVA_OPTS -cp $PWD/. org.springframework.boot.loader.JarLauncher))`
```

```
Showing health and status for app articulate in org dave / space dev as droberts@pivotal.io...
OK
```

```
requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: articulate-turbosupercharged-spinneret.cfapps.haas-39.piez.pivotal.io
last uploaded: Wed Jun 8 14:44:30 UTC 2016
stack: unknown
buildpack: https://github.com/cloudfoundry/java-buildpack.git
```

	state	since	cpu	memory	disk	details
#0	running	2016-06-08 09:45:32 AM	0.1%	409.4M of 512M	154.2M of 1G	

```
DROBERTS-MBPRO:articulate droberts$
```

The articulate app is now deployed

```

DROBERTS-MBPRO:articulate droberts$ cf app articulate
Showing health and status for app articulate in org dave / space dev as droberts@pivotal.io...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io
last uploaded: Wed Jun 8 14:44:30 UTC 2016
stack: cflinuxfs2
buildpack: https://github.com/cloudfoundry/java-buildpack.git

state since      cpu  memory      disk      details
#0  running  2016-06-08 09:45:32 AM  0.0%  309.3M of 512M  154.6M of 1G
DROBERTS-MBPRO:articulate droberts$

```

We are now using the latest version of the Java buildpack that we got from the GitHub URL we specified with the -b flag.

The purpose of this application is to articulate some basic concepts and capabilities of the Pivotal Cloud Foundry platform, specifically the Elastic Runtime which is responsible for running application workloads.

Application Architecture

articulate is a web application that exposes friendly, browsable user interface. However, it does not work with data directly. It depends on the **attendee-service** application to manage data. The **attendee-service** persists data to a MySQL database.



How to use this Application

Each menu item above links to a page that helps demonstrate a set of capabilities provided by the platform. The last item, Spring Boot, highlights capabilities that come with Spring Boot to help build production ready microservices in minutes.

Each page has the same layout with the Accordion control and up to 3 groups:

1. **Application Environment Information** - This provides information about the application environment when running inside PCF. You can see the Application Name, Container and Services information. This is useful to show things like load balancing, self healing, service binding among other things.
2. **Description** - additional context for the given page.
3. **The Twelve-Factor App** - a methodology for building modern, scalable applications. Links to applicable factors will be provided.

Provided to you by Pivotal

Application Environment Information

Application Name: articulate
Instance Index: 0
Container Address: 10.254.0.6:8080
Cell Address: 10.65.188.48:60592
Java Version: 1.8.0_91

Services

user-provided: attendee-service

Description

The 12 Factor App



The articulate service is now using an updated JRE version it got from the custom buildpack we specified.

```

DROBERTS-MBPRO:articulate droberts$ cf set-env articulate JBP_CONFIG_OPEN_JDK_JRE "{jre: { version: 1.8.0_45 }}"
Setting env variable 'JBP_CONFIG_OPEN_JDK_JRE' to '{jre: { version: 1.8.0_45 }}' for app articulate in org dave / sp
ev as droberts@pivotal.io...
OK
TIP: Use 'cf restage' to ensure your env variable changes take effect
DROBERTS-MBPRO:articulate droberts$

```

Now let us see a case where our application needs to run with a specific version of the Java JRE, we can simply set an environment variable and the Java buildpack to specify the exact JRE version using the **\$ cf set-env articulate JBP_CONFIG_OPEN_JDK_JRE "{jre: { version: 1.8.0_45 }}"** command we need as above. We can also set many other parameters like the tomcat version we want to be used also.


```

Restaging app articulate in org dave / space dev as droberts@pivotal.io...
Creating container
Successfully created container
Downloading app package...
Downloaded app package (27.6M)
Downloading build artifacts cache...
Downloaded build artifacts cache (88M)
Staging...
----> Java Buildpack Version: 29c79f2 | https://github.com/cloudfoundry/java-buildpack.git#29c79f2
----> Downloading Open Jdk JRE 1.8.0_45 from https://download.run.pivotal.io/openjdk/trusty/x86_64/openjdk-1.8.0_45
gz (found in cache)
    Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.2s)
----> Downloading Open JDK Like Memory Calculator 2.0.2_RELEASE from https://download.run.pivotal.io/memory-calculator-2.0.2_RELEASE.tar.gz (found in cache)
    Memory Settings: -Xms317161K -XX:MetaspaceSize=64M -Xss228K -Xmx317161K -XX:MaxMetaspaceSize=64M
----> Downloading Spring Auto Reconfiguration 1.10.0_RELEASE from https://download.run.pivotal.io/auto-reconfiguration-1.10.0_RELEASE.jar (found in cache)
Exit status 0
Staging complete
Uploading droplet, build artifacts cache...
Uploading build artifacts cache...
Uploading droplet...
Uploaded build artifacts cache (88M)

```

Then we restage the articulate service

```

App articulate was started using this command `CALCULATED_MEMORY=$(($PWD/.java-buildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-2.0.2_RELEASE -memorySizes=metaspace:64m...stack:228k.. -memoryWeights=heap:65,metaspace:10,native:10 -memoryInitials=heap:100%,metaspace:100% -stackThreads=300 -totMemory=$MEMORY_LIMIT) && JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh $CALCULATED_MEMORY" && SERVER_PORT=$(eval exec $PWD/.java-buildpack/open_jdk_jre/bin/java $JAVA_OPTS -cp $PWD/. org.springframework.boot.loader.JarLauncher)
Showing health and status for app articulate in org dave / space dev as droberts@pivotal.io...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io
last uploaded: Wed Jun 8 14:44:30 UTC 2016
stack: unknown
buildpack: https://github.com/cloudfoundry/java-buildpack.git

#0 state since cpu memory disk details
running 2016-06-08 09:49:07 AM 0.1% 316.6M of 512M 154.6M of 1G
DROBERTS-MBPRO:articulate droberts$

```

The purpose of this application is to articulate some basic concepts and capabilities of the Pivotal Cloud Foundry platform, specifically the Elastic Runtime which is responsible for running application workloads.

Application Architecture

articulate is a web application that exposes friendly, browsable user interface. However, it does not work with data directly. It depends on the **attendee-service** application to manage data. The **attendee-service** persists data to a MySQL database.



How to use this Application

Each menu item above links to a page that helps demonstrate a set of capabilities provided by the platform. The last item, Spring Boot, highlights capabilities that come with Spring Boot to help build production ready microservices in minutes.

Each page has the same layout with the Accordion control and up to 3 groups:

1. **Application Environment Information** - This provides information about the application environment when running inside PCF. You can see the Application Name, Container and Services information. This is useful to show things like load balancing, self healing, service binding among other things.
2. **Description** - additional context for the given page.
3. **The Twelve-Factor App** - a methodology for building modern, scalable applications. Links to applicable factors will be provided.

Application Environment Information

Application Name: articulate
Instance Index: 0
Container Address: 10.254.0.6:8080
Cell Address: 10.65.188.48:60594
Java Version: 1.8.0_45

Services

user-provided: attendee-service

Description

The 12 Factor App

We are now running with the specific JRE version we specified using the environment variables approach

Buildpack

Recap

