

Buildpack

Table of Contents

Requirements

What you will learn

Exercises

Use a Custom Buildpack

Change the Java version

Questions

Estimated Time: 25 minutes

Requirements

This lab is part of a workshop available at <https://github.com/Pivotal-Field-Engineering/DevNexus2017>
General Pre-Requisites are available in the Readme.md

In general the Labs should be done in numerical order since they are interdependent.


What you will learn

- How to use a custom buildpack
- How to configure the Java Buildpack

Exercises

Use a Custom Buildpack

- 1) Review the documentation on deploying with custom buildpacks (<http://docs.pivotal.io/pivotalcf/buildpacks/custom.html#deploying-with-custom-buildpacks>) and how dependencies are handled with the Java Buildpack online package (<https://github.com/cloudfoundry/java-buildpack#online-package>) and the offline package (<https://github.com/cloudfoundry/java-buildpack#offline-package>).
- 2) Review the Java Version reported by `articulate`.

 Articulate Scale and HA Services Blue-Green Spring Boot ▾

Welcome to Articulate!

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.

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: 10.254.0.10:8080

Cell address: 10.68.104.27:60298

Java Version: 1.8.0_65

Services

Using embedded H2 DB

Description

The 12 Factor App

- 3) Review which buildpack is in use.

```
$ cf app articulate
```

4) Push `articulate` again, but this time specify a custom buildpack. In this case, we will use the latest version of the Java Buildpack (<https://github.com/cloudfoundry/java-buildpack>) on GitHub.

```
$ cd sample-apps/articulate  
$ cf push articulate -p ./articulate-0.0.1-SNAPSHOT.jar -b https://github.com/cloudfoundry/java-buildpack.git
```

5) Using your browser, refresh the `articulate` application.

It's likely (but not required) that the `Java Version` changed.

What Just Happened?

We instructed our application to use a custom buildpack (as opposed to a system provided one).

In this case, we used the Java Buildpack source on Github as our custom buildpack. The Java Buildpack source is continuously updated and it is an online (<https://github.com/cloudfoundry/java-buildpack#online-package>) package of the buildpack. Meaning it has access to all dependencies via the network (it has access to all JRE versions, etc). Whereas, the system provided Java buildpack is offline (<https://github.com/cloudfoundry/java-buildpack#offline-package>), with a limited set of dependencies. For both the online and offline packages, unless the Java version is specified the application is run with the latest version of Java available to the buildpack.

Change the Java version

1) Review the Java Buildpack configuration and extension documentation (<https://github.com/cloudfoundry/java-buildpack#configuration-and-extension>).

2) Let's assume that we want to run `articulate` on a specific version Java.

```
$ cf set-env articulate JBP_CONFIG_OPEN_JDK_JRE "{jre: { version: 1.8.0_45 }}"
```

3) Using your browser, refresh the articulate application.

QUESTION: Is the articulate running with 1.8.0_45 ? Why not?

4) Restage articulate .

```
$ cf restage articulate
```

QUESTION: Would cf restart be sufficient instead of cf restage ? Why not?

5) Using your browser, refresh the articulate application.

Questions

- What other items are easily customized with the Java Buildpack?
- If you use Java, what items do you think would need customization in your environment?

[Back to TOP](#)

© Copyright Pivotal. All rights reserved.