

Blue Green Deployment Lab guide

```
cf create-service cleardb spark attendee-mysql
```

```
1) git clone https://github.com/pivotal-education/pcf-attendee-service-code.git
```

```
cd .\pcf-attendee-service-code
./mvnw package
cf push attendee-service -p .\target\attendee-1.0.jar -m 752M --random-route
cf bind-service attendee-service attendee-mysql

cf restart attendee-service

$ cf create-user-provided-service attendee-service -p uri

uri> https://attendee-service-appreciative-dingo.cfapps.io/attendees
```

- 2) \$ git clone <https://github.com/pivotal-education/pcf-articulate-code.git>
- 3) \$ cd pcf-articulate-code
- 4) \$./mvnw clean package
- 5) \$ cf push articulate -p .\target\articulate-1.0.jar -m 512M --random-route --no-start

Bind articulate to the attendee-service user provided service

```
$ cf bind-service articulate attendee-service
```

Tip: Use 'cf restage articulate' to ensure env variable changes take effect" message at this time.

Restart the application.

```
$ cf restart articulate
```

Access articulate logs

- 1) Review the documentation on [application logging](#)
- 2) Tail the logs of the articulate application

```
$ cf logs articulate
```

- 3) Open another terminal window and start the articulate application
- 4) Review the output from both terminal windows

```
$ cf start articulate
```

- 5) Open a browser and view the articulate application and read through the demo application

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.

```
graph LR; articulate["articulate  
(Spring Boot, Spring MVC, Spring Cloud Connectors, Thymeleaf, Bootstrap)"] -- "REST (http/json)" --> attendee-service["attendee-service  
(Spring Boot, Spring Data REST, Spring Data JPA)"]; attendee-service --- MySQL[(MySQL)];
```

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: 10.254.0.66:8080
Cell Address: 10.10.115.117:64642
Java Version: 1.8.0_101

Services

None

Description

The 12 Factor App

- 6) Observe the log output when the articulate web page is refreshed (More logs are added)

- 7) Stop tailing logs

1. Go to the terminal tailing the logs
2. Send an interrupt (Control + c)

Steps to Perform a Blue-Green Deployment

Step 1: To simulate a blue-green deployment, first scale articulates to multiple instances

```
$ cf scale articulate -i 2
```

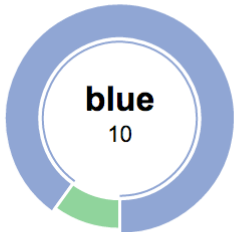
Step 2: Browse to the articulate Blue-Green page

[Articulate](#) [Scale & HA](#) [Services](#) [Blue-Green](#) [Spring Boot](#) ▾

Blue-Green Deployment

How hard it is for you to upgrade your application with minimal downtime?

This page shows the load balancing between application versions based on route mappings. [See more in the description.](#)



Start Reset Stop

Provided to you by Pivotal!

Application Environment Information

Application Name: articulate
Instance Index: 0
Container Address: 10.254.0.54:8080
Cell Address: 10.10.115.39:60617
Java Version: 1.8.0_71

Services

user-provided: attendee-service

Description

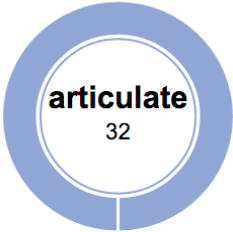
Step 3: Assume that the deployed application is version 1. Generate some traffic and press the Start button. Leave this open as a dedicated tab in your browser

Step 4: Observe the existing application handling all web requests

[Articulate](#) [Scale & HA](#) [Services](#) [Blue-Green](#) [Spring Boot](#) ▼

Blue-Green Deployment

articulate - 32



Start Reset Stop

Application Environment Information

Application Name: articulate

Instance Index: 1

Container Address: 10.254.1.2:8080

Cell Address: 10.10.114.71:60747

Java Version: 1.8.0_71

Services

user-provided: attendee-service

Description

Provided to you by Pivotal!

Step 5: Record the subdomain (host) for the articulate application

For example:

```
$ cf routes
```

Getting routes as droberts@pivotal.io ...

space	host	domain	apps
dev	articulate-heartsickening-elegance	pcfi1.fe.gopivotal.com	articulate

Step 6: Now, push the next version of articulate

However, this time specify the subdomain by appending `-temp` to the production route.

For example, (Subdomain is different):

```
$ cd ~/pivotal-cloud-foundry-developer-workshop/articulate/  
$ cf push articulate1 -p .\target\articulate-1.0.jar -m 512M -n articulate-heartsickening-eleg  
ance-temp --no-start
```

Note: bold color articulate host link should be replaced with your articulate host link

Step 7: Bind articulate-v2 to the attendee-service user provided service


```
$ cf bind-service articulate1 attendee-service
```

Step 8: Start the application

```
$ cf start articulate1
```

Step 9: Now there are two versions of the deployed app

Open a new tab and view version 2 of articulate in the browser. Take note of the application name.


 Articulate Scale & 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.

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-v2
Instance Index: 0
Container Address: 10.254.0.66:8080
Cell Address: 10.10.115.117:64646
Java Version: 1.8.0_71

Services

user-provided: attendee-service

Description

The 12 Factor App


At this point in the deployment process, further testing of the version can be done before releasing it to the customers.

Step 10: Let's assume we are ready to start directing production traffic to version 2. We need to map our production route to articulate-v2

For example, (your domain and subdomain will be different):

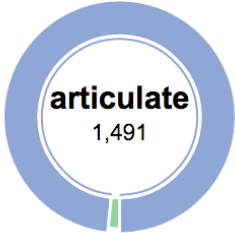
```
$ cf map-route articulate1 cfapps.io -n articulate-accountable-turtle
```

Step 11: Return to the browser tab where the load has started and is sending requests to version 2

 Articulate Scale and HA Services Blue-Green Spring Boot ▾

Blue-Green Deployment

articulate - 1491
articulate-v2 - 9



articulate
1,491

Start Reset Stop

Application Environment Information

Application Name: articulate
Instance index: 1
Container address: 10.254.0.10:8080
Cell address: 10.68.104.29:60160

Services
Using embedded H2 DB

Description

Provided to you by Pivotal

Step 12: Press the Reset button, so that we can see how the load get distributed across app instances

Check if the configuration is same as given below:

```
# cf apps
Getting apps in org dave / space dev as droberts@pivotal.io...
OK
```

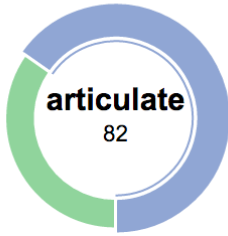
name	requested	state	instances	memory	disk	urls
articulate	started	2/2	512M	1G	...	
articulate1	started	1/1	512M	1G	...	

This will result in one-third of the requests going to version 2.



Blue-Green Deployment

articulate - 82
articulate-v2 - 43



Start Reset Stop

Application Environment Information

Application Name: articulate

Instance Index: 0

Container Address: 10.254.0.54:8080

Cell Address: 10.10.115.39:60617

Java Version: 1.8.0_71

Services

user-provided: attendee-service

Description

Provided to you by Pivotal!

Step 13: Move more traffic to version 2

```
# cf scale articulate -i 1  
# cf scale articulate-v2 -i 2
```

If the load generator is reset, you will see two-third of the traffic going to `articulate-v2`.

Step 14: Move all traffic to version 2

Remove the production route from the articulate application.

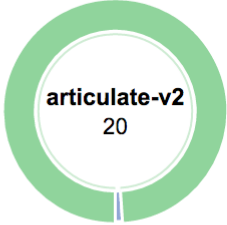
For example, (The domain and subdomain will be different):

```
# cf unmap-route articulate cfapps.io -n articulate-accountable-turtle
```

If the load generator is reshuffled, you will see all traffic going to `articulate-v2`.

Blue-Green Deployment

articulate - 0
articulate-v2 - 20



Start

Reset

Stop

Application Environment Information

Application Name: articulate

Instance Index: 0

Container Address: 10.254.0.54:8080

Cell Address: 10.10.115.39:60617

Java Version: 1.8.0_71

Services

user-provided: attendee-service

Description

Provided to you by Pivotal!

Note: Refreshing the entire page will update the application name.

Step 15: Remove the temp route from the articulate-v2 application

For example, (The domain and subdomain will be different):

```
# cf unmap-route articulate-v2 pcfi1.fe.gopivotal.com -n articulate-heartsickening-elegance-temp
```

This completes the blue-green deployment.