

## Blue-Green Deployments

### Agenda

1. Blue-Green Routing
2. Implications on App Design

### Blue-Green Deployment

Zero downtime when upgrading can be accomplished with blue-green deployments.



## Blue-Green Deployment



We might assign a temp URL to our staging instance to allow us to test it before moving it into production

## Blue-Green Deployment



We temporarily map the production route to both application versions during blue-green deployment and distribute traffic accordingly.

## Blue-Green Deployment



We then unmap the route from the older version when we have confidence in the new version instance

## Agenda

1. Blue Green Routing
2. Implications on App & Data Model Design

## Serializing Objects

If using serializing objects, don't make destructive changes.

e.g. don't remove fields, do have a `serialVersionUID`



## The 12 Factors



### Admin Processes

Run admin/mgmt tasks as one-off processes.

e.g. migrating data

### Database

Do make changes idempotent.

e.g. copy data to a new field (don't delete data)

### Database

No destructive database changes allowed.

e.g. don't drop a column

## Database

Do have backwards compatible changes.

e.g. nullable fields

Let us now demonstrate blue-green deployment with the articulate application.

```
DROBERTS-MBPRO:articulate droberts$ cf apps
Getting apps in org dave / space dev as droberts@pivotal.io...
OK
```

name	requested state	instances	memory	disk	urls
articulate	started	1/1	512M	1G	articulate-turbosupercharged-spinneret.cfapps.haas-
attendee-service	started	1/1	512M	1G	attendee-service-monochromatic-guarantee.cfapps.haa-

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

We use the `$ cf apps` command to see the applications running in our Org and Space, but we want to make the articulate application more production-like with at least 2 instances running.

```
DROBERTS-MBPRO:articulate droberts$ cf scale articulate -i 2
Scaling app articulate in org dave / space dev as droberts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$
```

We scale out the articulate application to 2 instances using the `$ cf scale articulate -i 2` command.

```
DROBERTS-MBPRO:articulate droberts$ 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	articulate-turbosupercharged-spinneret.cfapps.haas-39.pew.pivotal.io
attendee-service	started	1/1	512M	1G	attendee-service-monochromatic-guarantee.cfapps.haas-39.pew.pivotal.io

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

We copy the URL for the articulate app

# 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 Service 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:** 10.254.0.38:8080  
**Cell Address:** 10.65.188.47:60218  
**Java Version:** 1.8.0\_71

### Services

**user-provided:** attendee-service

### Description

The 12 Factor App



# 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:** 10.254.0.38:8080  
**Cell Address:** 10.65.188.47:60218  
**Java Version:** 1.8.0\_71

### Services

**user-provided:** attendee-service

### Description





## 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: 10.254.0.38:8080

Cell Address: 10.65.188.47:60218

Java Version: 1.8.0\_71

### Services

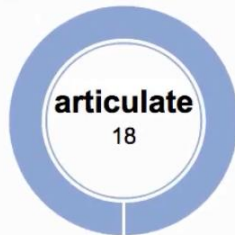
user-provided: attendee-service

### Description



## Blue-Green Deployment

articulate - 18



Start

Reset

Stop

Provided to you by Pivotal!

### Application Environment Information

Application Name: articulate

Instance Index: 0

Container Address: 10.254.0.38:8080

Cell Address: 10.65.188.47:60218

Java Version: 1.8.0\_71

### Services

user-provided: attendee-service

### Description



We then start generating multiple requests to the articulate application to see which instance is serving each request

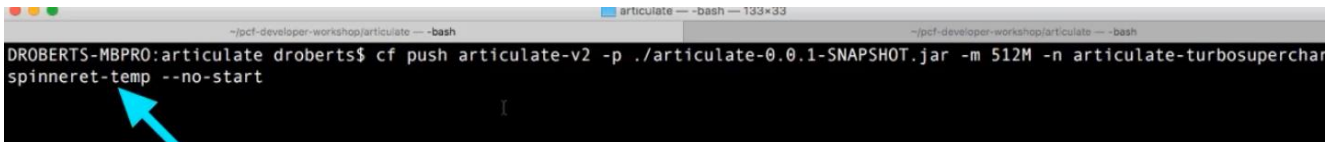
```
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...

space  host                                domain                                port  path  type  apps
dev    articulate-turbosupercharged-spinneret cfapps.haas-39.pez.pivotal.io        8080  /     GET   articulate
dev    attendee-service-monochromatic-guarantee cfapps.haas-39.pez.pivotal.io        8080  /     GET   attendee-ser
DROBERTS-MBPRO:articulate droberts$
```

We then start creating the green application for articulate v2. The **\$ cf routes** command will give us all the available routes in this Org Space.

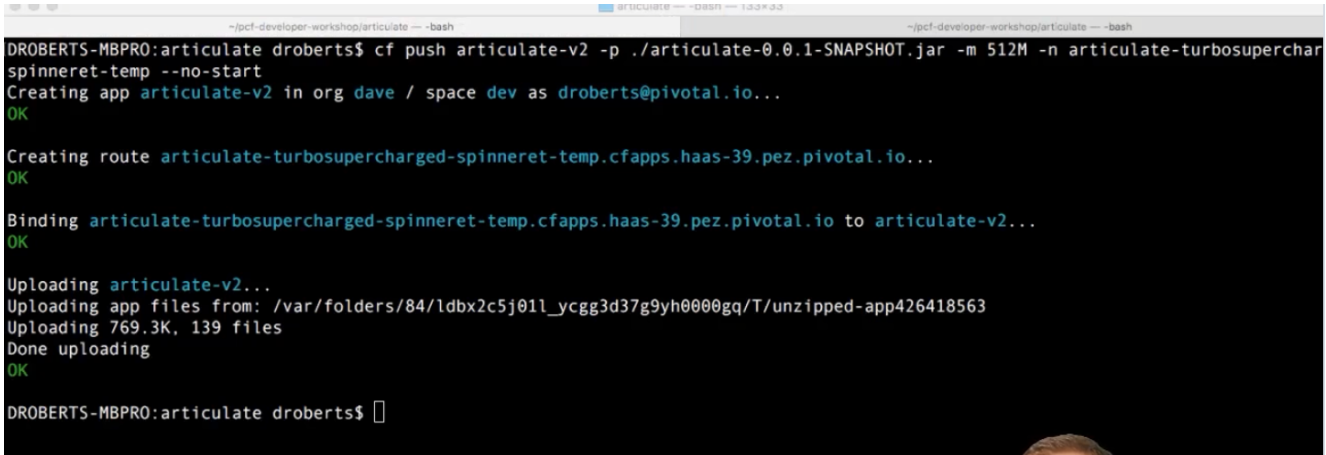
```
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...

space  host                                domain                                port  path  type  apps  service
dev    articulate-turbosupercharged-spinneret cfapps.haas-39.pez.pivotal.io        8080  /     GET   articulate
dev    attendee-service-monochromatic-guarantee cfapps.haas-39.pez.pivotal.io        8080  /     GET   attendee-service
DROBERTS-MBPRO:articulate droberts$
```

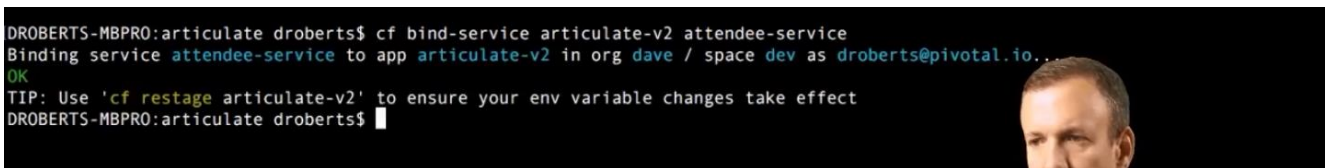


```
DROBERTS-MBPRO:articulate droberts$ cf push articulate-v2 -p ./articulate-0.0.1-SNAPSHOT.jar -m 512M -n articulate-turbosupercharged-spinneret-temp --no-start
```

We now push the new version of the articulate application using the **`$ cf push articulate-v2 -p ./articulate-0.0.1-SNAPSHOT.jar -m 512M -n articulate-turbosupercharged-spinneret-temp --no-start`** command, we simply push the new app to the same sub-domain with a **`-temp`** suffix attached to the name as above.

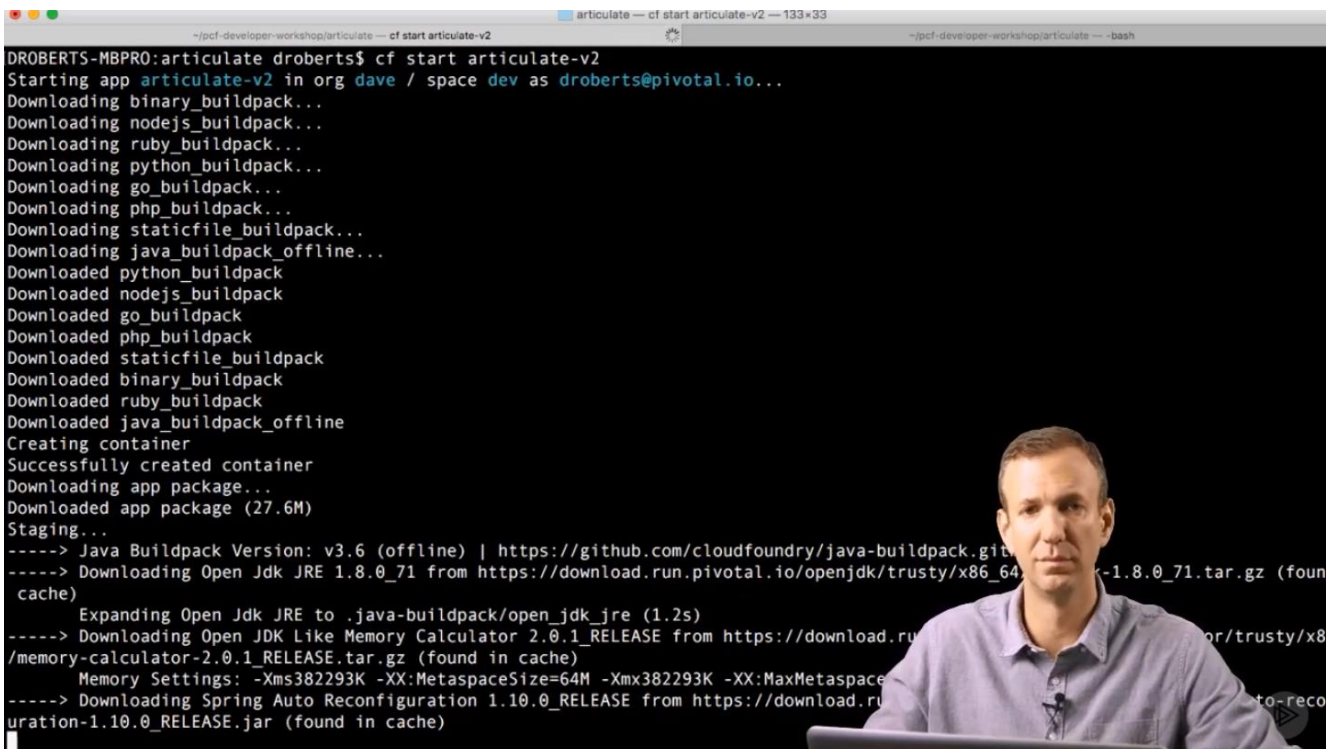


```
DROBERTS-MBPRO:articulate droberts$ cf push articulate-v2 -p ./articulate-0.0.1-SNAPSHOT.jar -m 512M -n articulate-turbosupercharged-spinneret-temp --no-start
Creating app articulate-v2 in org dave / space dev as droberts@pivotal.io...
OK
Creating route articulate-turbosupercharged-spinneret-temp.cfapps.haas-39.pez.pivotal.io...
OK
Binding articulate-turbosupercharged-spinneret-temp.cfapps.haas-39.pez.pivotal.io to articulate-v2...
OK
Uploading articulate-v2...
Uploading app files from: /var/folders/84/ldb2c5j01l_ycgg3d37g9yh0000gq/T/unzipped-app426418563
Uploading 769.3K, 139 files
Done uploading
OK
DROBERTS-MBPRO:articulate droberts$
```



```
DROBERTS-MBPRO:articulate droberts$ cf bind-service articulate-v2 attendee-service
Binding service attendee-service to app articulate-v2 in org dave / space dev as droberts@pivotal.io...
OK
TIP: Use 'cf restage articulate-v2' to ensure your env variable changes take effect
DROBERTS-MBPRO:articulate droberts$
```

We then bind the new version of the articulate application so that it is wired/bound up to the attendee service using the **`$ cf bind-service articulate-v2 attendee-service`** command.



```
DROBERTS-MBPRO:articulate droberts$ cf start articulate-v2
Starting app articulate-v2 in org dave / space dev as droberts@pivotal.io...
Downloading binary_buildpack...
Downloading nodejs_buildpack...
Downloading ruby_buildpack...
Downloading python_buildpack...
Downloading go_buildpack...
Downloading php_buildpack...
Downloading staticfile_buildpack...
Downloading java_buildpack_offline...
Downloaded python_buildpack
Downloaded nodejs_buildpack
Downloaded go_buildpack
Downloaded php_buildpack
Downloaded staticfile_buildpack
Downloaded binary_buildpack
Downloaded ruby_buildpack
Downloaded java_buildpack_offline
Creating container
Successfully created container
Downloading app package...
Downloaded app package (27.6M)
Staging...
-----> Java Buildpack Version: v3.6 (offline) | https://github.com/cloudfoundry/java-buildpack.git
-----> Downloading Open Jdk JRE 1.8.0_71 from https://download.run.pivotal.io/openjdk/trusty/x86_64/1.8.0_71.tar.gz (found in cache)
Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.2s)
-----> Downloading Open JDK Like Memory Calculator 2.0.1_RELEASE from https://download.run.pivotal.io/memory-calculator-2.0.1_RELEASE.tar.gz (found in cache)
Memory Settings: -Xms382293K -XX:MetaspaceSize=64M -Xmx382293K -XX:MaxMetaspaceSize=64M
-----> Downloading Spring Auto Reconfiguration 1.10.0_RELEASE from https://download.run.pivotal.io/spring-auto-reconfiguration-1.10.0_RELEASE.jar (found in cache)
```

Then we start the version 2 of the articulate application using the **`$ cf start articulate-v2`** command. This will provision staging, have a container to run the version 2 instance in



```
~/pcf-developer-workshop/articulate --- -bash
Uploaded droplet (72.5M)
Uploading complete

0 of 1 instances running, 1 starting
1 of 1 instances running

App started

OK

App articulate-v2 was started using this command `CALCULATED_MEMORY=$(PWD/.java-buildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-2.0.1_RELEASE -memorySizes=metaspace:64m... -memoryWeights=heap:75,metaspace:10,native:10,stack:5 -memoryInitials=heap:100,metaspace:100% -totMemory=$MEMORY_LIMIT) && JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh $CALCULATED_MEMORY" && SERVER_PORT=$PORT eval exec $PWD/.java-buildpack/open_jdk_jre/bin/java $JAVA_OPTS -cp $PWD/.java-buildpack/spring_auto_reconfiguration/spring_auto_reconfiguration-1.10.0_RELEASE.jar org.springframework.boot.loader.launcher`

Showing health and status for app articulate-v2 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-temp.cfapps.haas-39.pez.pivotal.io
last uploaded: Wed Jun 1 14:53:23 UTC 2016
stack: unknown
buildpack: java-buildpack=v3.6-offline-https://github.com/cloudfoundry/java-buildpack.git#... like-jre=1.8
1 open-jdk-like-memory-calculator=2.0.1_RELEASE spring-auto-reconfiguration=1.10.0_RELEASE

#0 state since cpu memory disk details
running 2016-06-01 09:54:48 AM 0.0% 219M of 512M 154.1M of 1G
DROBERTS-MBPRO:articulate droberts$
```

We now have v2 running but we have not yet bound it to the route where we have v1 running in production for blue-green deployment to take place.

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.

articulate  
(Spring Boot, Spring MVC, Spring Cloud Connectors, Thymeleaf, Bootstrap)

REST (http/json)

attendee-service  
(Spring Boot, Spring Data JPA, Spring Cloud Connectors)

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 Information

Application Name: articulate-v2

Instance Index: 0

Container Address: 10.254.0.66:8080

Cell Address: 10.65.188.48:60370

Java Version: 1.8.0\_66

### Services

user-provided: a

### Description

The 1

The articulate-v2 is now running in its temp environment where we can test it out before moving it to production.

```
DROBERTS-MBPRO:articulate droberts$ cf map-route articulate-v2
```

We can now map the route for the articulate application to the v2 version, we will need to use the `$ cf map-route articulate-v2 <domain> -n <sub-domain>` command along with the domain we want to map to

```
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...
```

space	host	domain	port	path	type	apps
dev	articulate-turbosupercharged-spinneret	cfapps.haas-39.pez.pivotal.io				articulate
dev	attendee-service-monochromatic-guarantee	cfapps.haas-39.pez.pivotal.io				attendee-ser

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

```
DROBERTS-MBPRO:articulate droberts$ cf map-route articulate-v2 cfapps.haas-39.pez.pivotal.io
```

The domain is ***cfapps.haas-39.pez.pivotal.io***

```
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...
```

space	host	domain	port	path	type	apps
dev	articulate-turbosupercharged-spinneret	cfapps.haas-39.pez.pivotal.io				articulate
dev	attendee-service-monochromatic-guarantee	cfapps.haas-39.pez.pivotal.io				attendee-ser

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

```
articulate droberts$ cf map-route articulate-v2 cfapps.haas-39.pez.pivotal.io -n articulate-turbosupercharged-spinneret
```

The sub-domain or host is ***articulate-turbosupercharged-spinneret***

```
DROBERTS-MBPRO:articulate droberts$ cf map-route articulate-v2 cfapps.haas-39.pez.pivotal.io -n articulate-turbosupercharged-spinneret
Creating route articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io for org dave / space dev as droberts@pivotal.io...
OK
Route articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io already exists
Adding route articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io to app articulate-v2 in org dave / space dev as droberts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$
```

## Blue-Green Deployment

```
articulate - 356
```

```
articulate-v2 - 8
```

**articulate**  
356

Start Reset Stop

Provided to you by Pivotal!

Application Environment Information

Application Name: articulate

Instance Index: 0

Container Address: 10.254.0.38:8080

Cell Address: 10.65.188.47:60218

Java Version: 1.8.0\_71

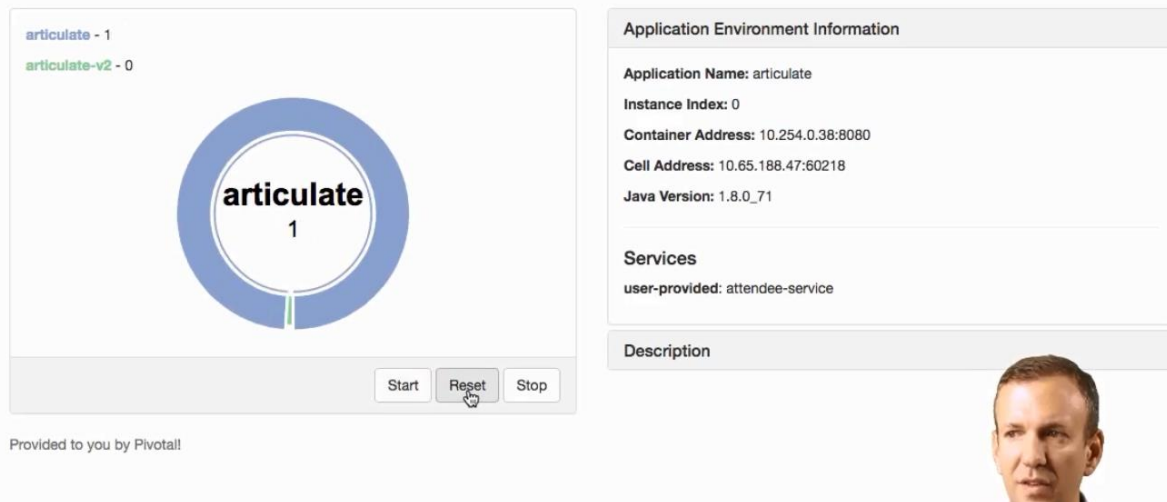
Services

user-provided: attendee-service

Description

The web requests are now going to both v1 and v2 versions of the articulate application running in production.

## Blue-Green Deployment



articulate - 1  
articulate-v2 - 0

articulate  
1

Start Reset Stop

Provided to you by Pivotal!

Application Environment Information

Application Name: articulate  
Instance Index: 0  
Container Address: 10.254.0.38:8080  
Cell Address: 10.65.188.47:60218  
Java Version: 1.8.0\_71

Services  
user-provided: attendee-service

Description

Let us rest this and see how the traffic is distributed

## Blue-Green Deployment



articulate - 26  
articulate-v2 - 13

articulate  
26

Start Reset Stop

Provided to you by Pivotal!

Application Environment Information

Application Name: articulate  
Instance Index: 0  
Container Address: 10.254.0.38:8080  
Cell Address: 10.65.188.47:60218  
Java Version: 1.8.0\_71

Services  
user-provided: attendee-service

Description

The traffic is divided by the number of instances running, we currently have two v1 versions and one v2 version running. We can now slowly start to move traffic off the v1 version of the articulate application.

```
DROBERTS-MBPRO:articulate droberts$ cf scale articulate -i 1
Scaling app articulate in org dave / space dev as droberts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$ cf scale articulate-v2 -i 2
Scaling app articulate-v2 in org dave / space dev as droberts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$
```

We use the `$ cf scale articulate -i 1` command to scale each version as we want

## Blue-Green Deployment



Provided to you by Pivotal!

### Application Environment Information

Application Name: articulate  
Instance Index: 0  
Container Address: 10.254.0.38:8080  
Cell Address: 10.65.188.47:60218  
Java Version: 1.8.0\_71

### Services

user-provided: attendee-service

### Description



## Blue-Green Deployment



Provided to you by Pivotal!

### Application Environment Information

Application Name: articulate  
Instance Index: 0  
Container Address: 10.254.0.38:8080  
Cell Address: 10.65.188.47:60218  
Java Version: 1.8.0\_71

### Services

user-provided: attendee-service

### Description



The v2 version is now getting more traffic.

```
DROBERTS-MBPRO:articulate droberts$ cf unmap-route articulate cfapps.haas-39.pez.pivotal.io -n articulate-turbosupercharged-spinneret
Removing route articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io from app articulate in org dave / space dev as dr
oberts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$
```

To complete the blue-green deployment and remove the v1 version completely, we need to unmap the route for the v1 version using the **\$ cf unmap-route articulate <domain> -n <sub-domain>** command.



Blue-Green Deployment

articulate - 0  
articulate-v2 - 8

articulate-v2  
8

Start Reset Stop

Provided to you by Pivotal

Application Environment Information

Application Name: articulate  
Instance Index: 0  
Container Address: 10.254.0.38:8080  
Cell Address: 10.65.188.47:60218  
Java Version: 1.8.0\_71

Services  
user-provided: attendee-service

Description

All traffic should now be going to the articulate v2 version alone.

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.

blue  
10

Start Reset Stop

Provided to you by Pivotal

Application Environment Information

Application Name: articulate-v2  
Instance Index: 0  
Container Address: 10.254.0.66:8080  
Cell Address: 10.65.188.48:60370  
Java Version: 1.8.0\_71

Services  
user-provided: attendee-service

Description

Our web page requests are also now going to the v2 version only.

```
DROBERTS-MBPRO:articulate droberts$ cf apps
Getting apps in org dave / space dev as droberts@pivotal.io...
OK
name           requested state   instances  memory  disk  urls
articulate      started           1/1        512M    1G
articulate-v2   started           2/2        512M    1G    articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io
, articulate-turbosupercharged-spinneret-temp.cfapps.haas-39.pez.pivotal.io
attendee-service started           1/1        512M    1G    attendee-service-monochromatic-guarantee.cfapps.haas-39.pez.pivotal.io
DROBERTS-MBPRO:articulate droberts$
```

If we use the **\$ cf apps** command to see our running applications, we can see that there are no URLs bound to the articulate v1 application anymore. We can also see that the articulate v2 version has both a temporary and a production route available.



```
DROBERTS-MBPRO:articulate droberts$ cf unmap-route articulate-v2 cfapps.haas-39.pez.pivotal.io -n articulate-turbosupercharged-spinneret-temp
Removing route articulate-turbosupercharged-spinneret-temp.cfapps.haas-39.pez.pivotal.io from app articulate-v2 in org dave / space dev as droberts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$
```

We can also unmap the temporary route for the articulate v2 version as above.

```
DROBERTS-MBPRO:articulate droberts$ cf apps
Getting apps in org dave / space dev as droberts@pivotal.io...
OK

name           requested state  instances  memory  disk  urls
articulate      started          1/1        512M    1G
articulate-v2   started          2/2        512M    1G    articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io
attendee-service started          1/1        512M    1G    attendee-service-monochromatic-guarantee.cfapps.haas-39.pez.pivotal.io
DROBERTS-MBPRO:articulate droberts$
```

We now have only the production URL for the articulate v2 version. This completes our blue-green deployment for the articulate application.

# Blue-Green Deployments

Recap

