Pivotal

Course developed by
Pivotal Academy

Pivotal Cloud Foundry Developer v1.7

BLUE-GREEN AND MICROSERVICES

Blue-Green Deployments

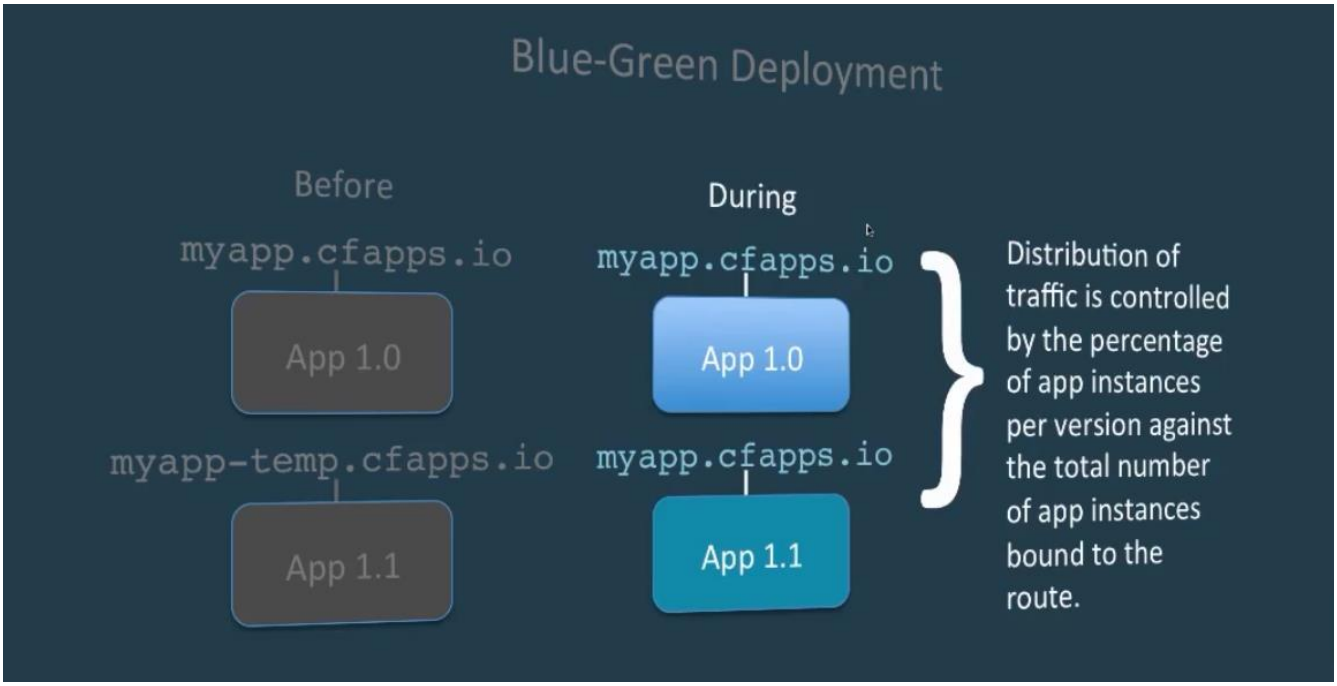
Agenda

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

Blue-Green Deployment

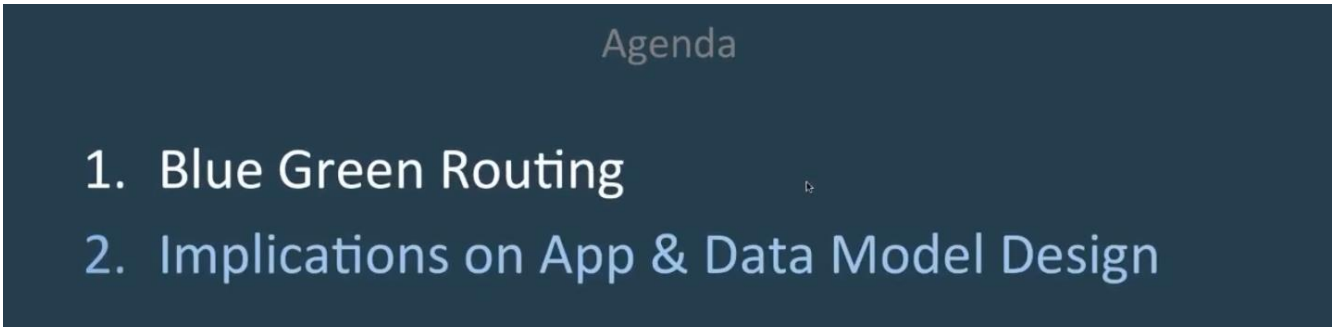
Zero downtime when upgrading can be accomplished with blue-green deployments.

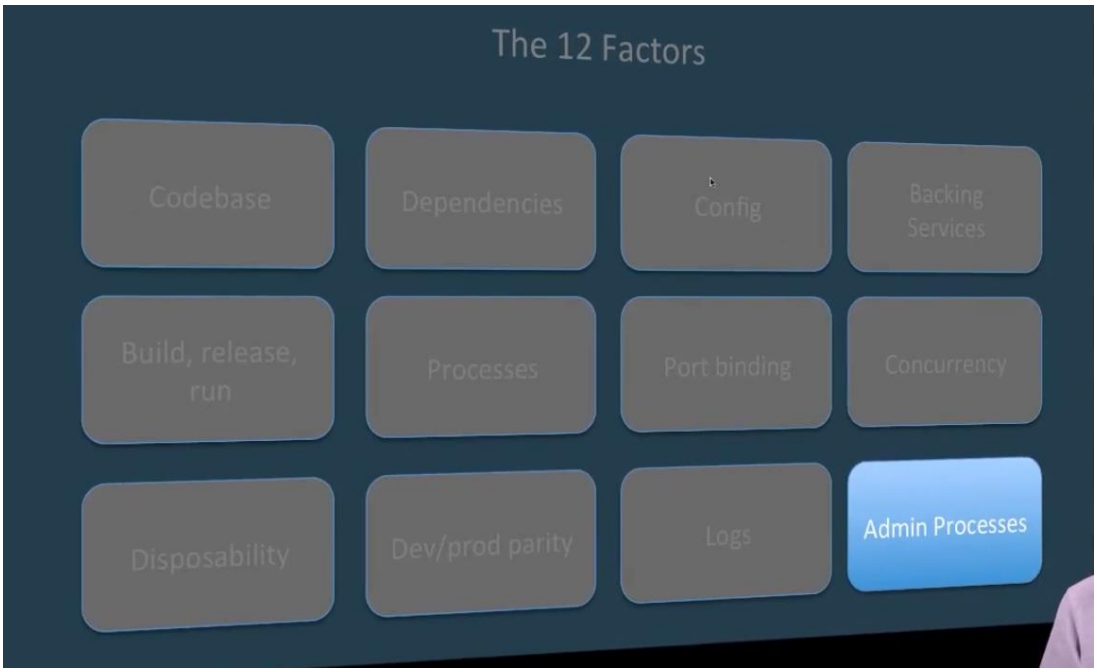
We might assign a temp URL to our staging instance to allow us to test it before moving it into production



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

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





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

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.

```
articulate — -bash — 133×33
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.pez.pivotal.i
attendee-service   started           1/1         512M     1G     attendee-service-monochromatic-guarantee.cfapps.haas-39.pez.pivotal.
io
DROBERTS-MBPRO:articulate droberts$
```

We copy the URL for the articulate app

Articulate

articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io

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

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

Articulate

articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io/bluegreen

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: 10.254.0.38:8080

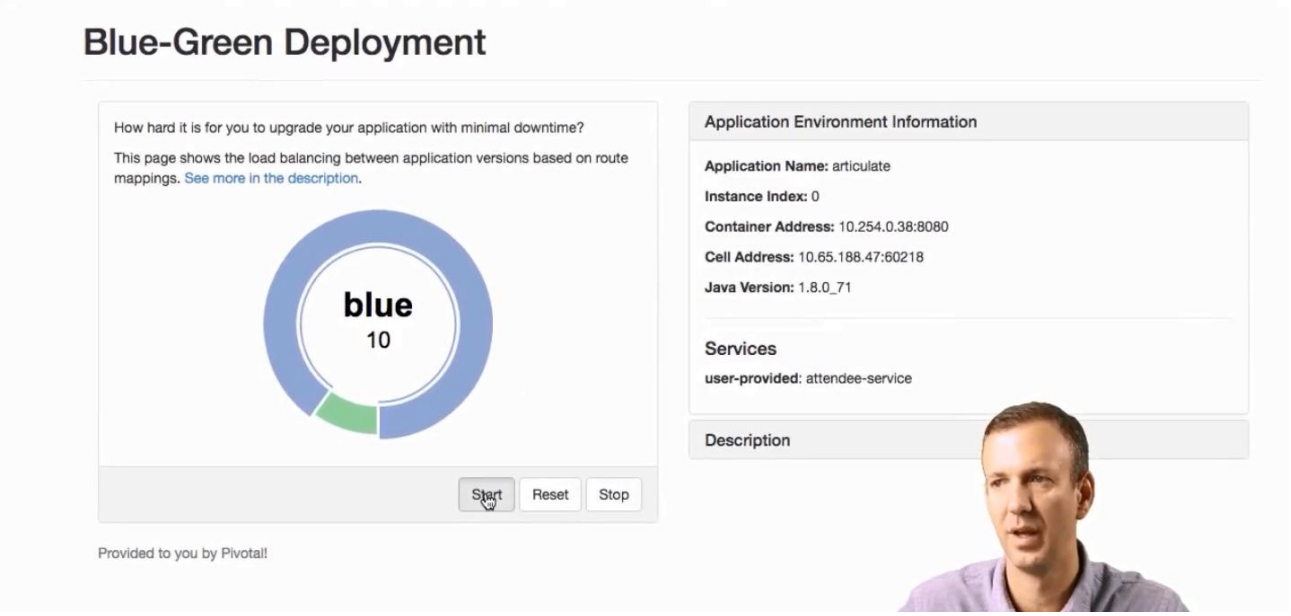
Cell Address: 10.65.188.47:60218

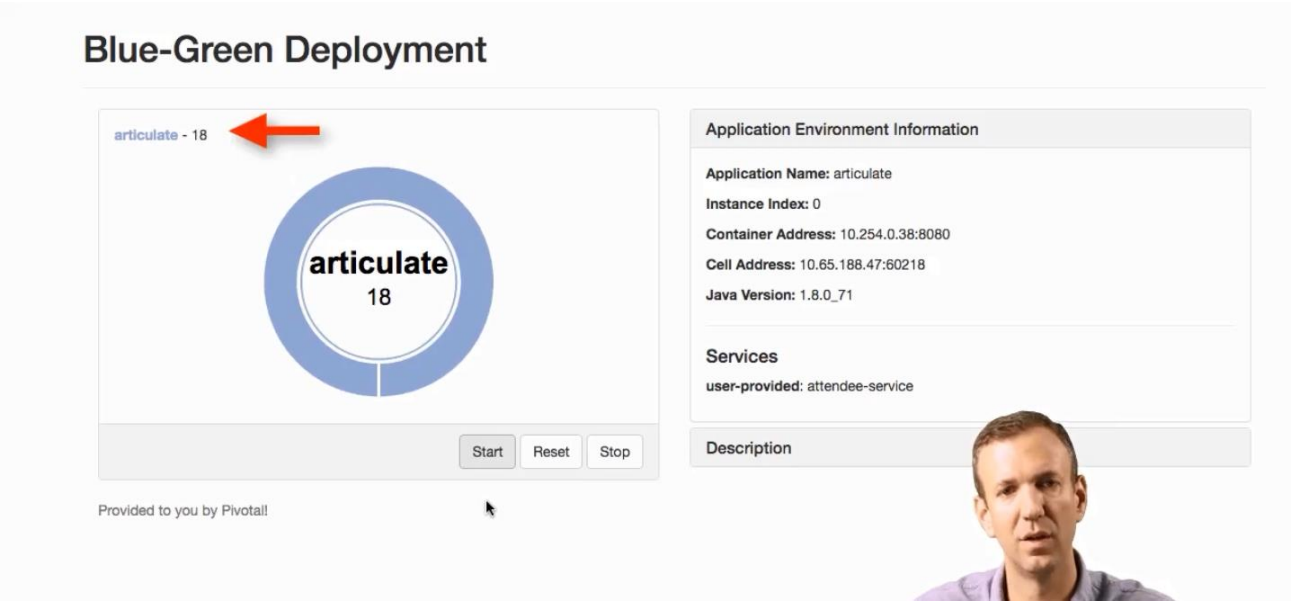
Java Version: 1.8.0_71

Services

user-provided: attendee-service

Description





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

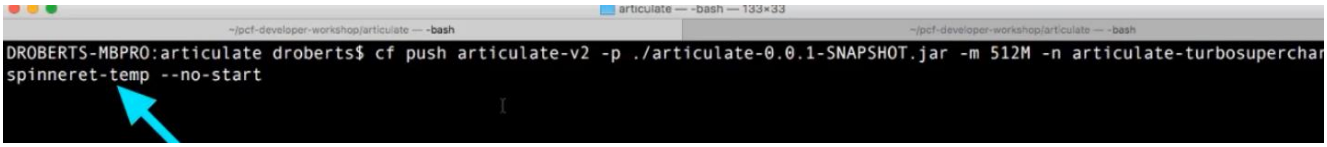
```
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...

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

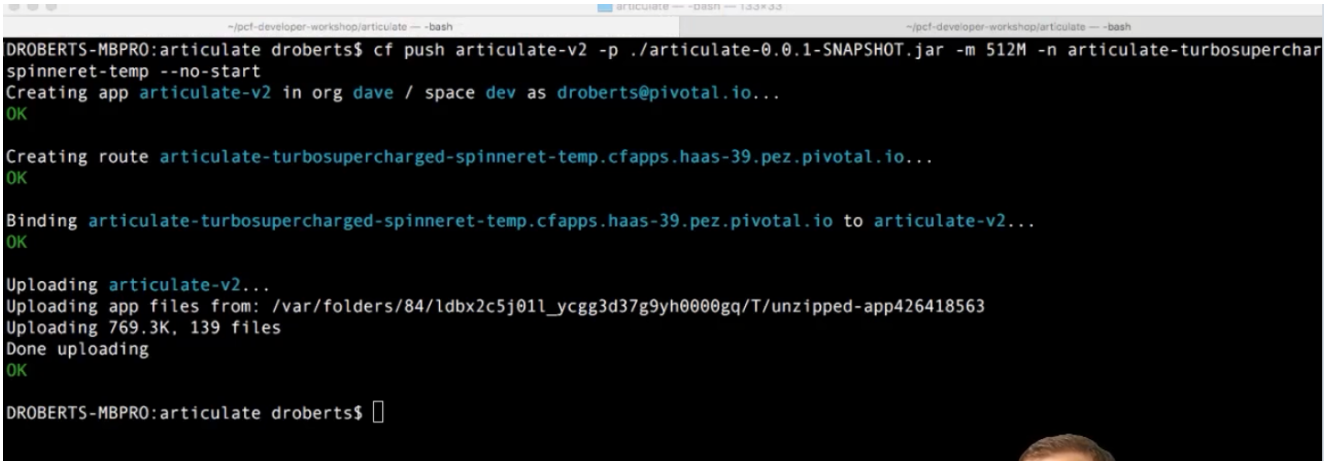
We then start creating the green application for articulate v2. The ***$ cf routes*** command will give us all the available routes in this Org Space.

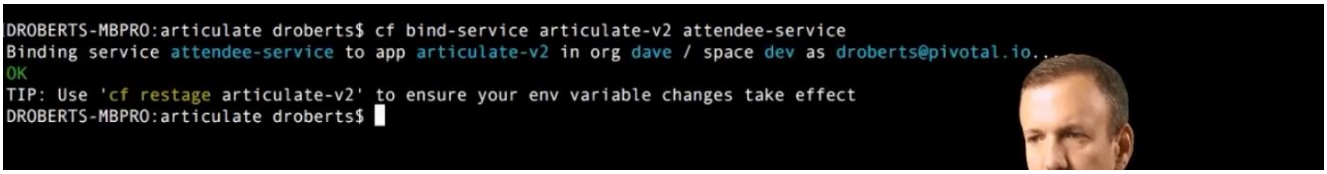
```
articulate — -bash — 133×33
~/pcf-developer-workshop/articulate — -bash
~/pcf-developer-workshop/articulate — -bash
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...

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

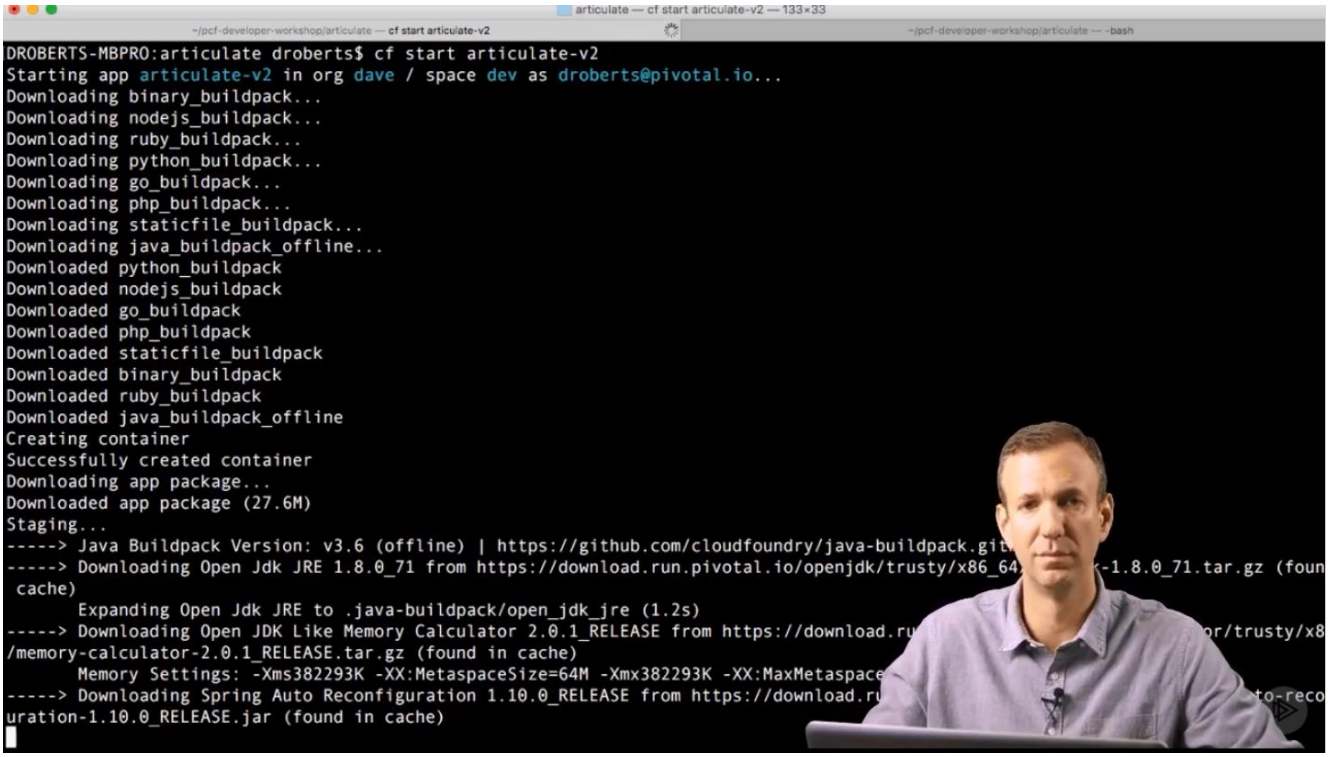


We now push the new version of the articulate application using the ***$ cf push articulate-v2 -p ./articulate-0.0.1-SNAPSHOT.jar -m 512M -n articulate-turbosupercharged-spinneret-temp - -no-start*** command, we simply push the new app to the same sub-domain with a ***-temp*** suffix attached to the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

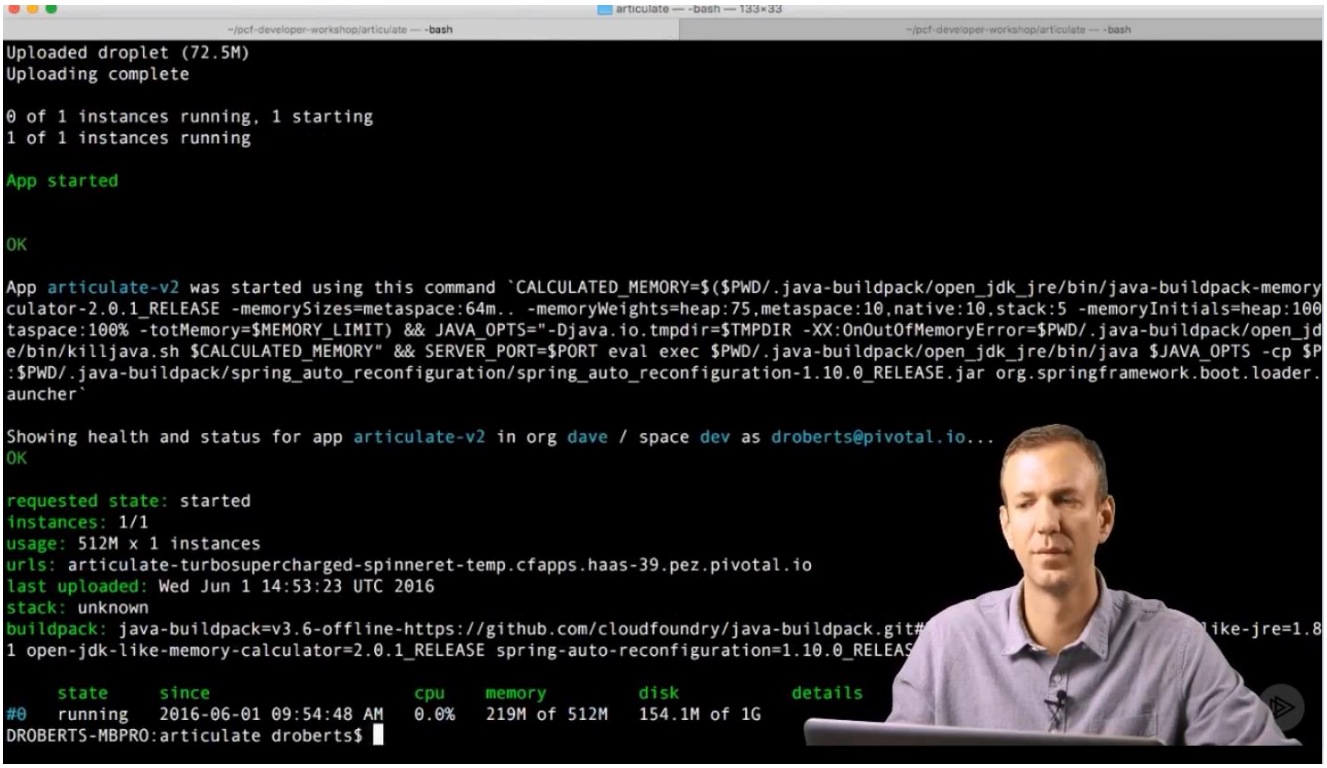




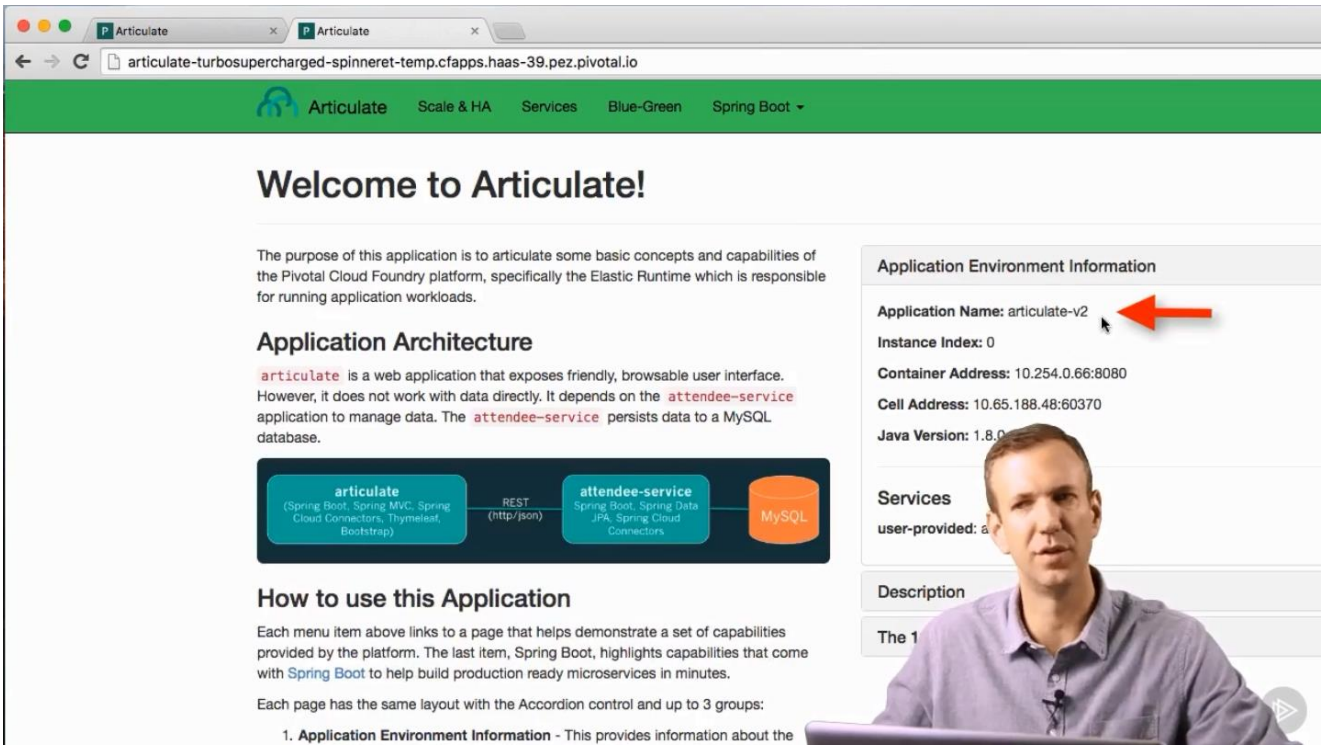
We then bind the new version of the articulate application so that it is wired/bound up to the attendee service using the ***$ cf bind-service articulate-v2 attendee-service*** command.



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



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.



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***

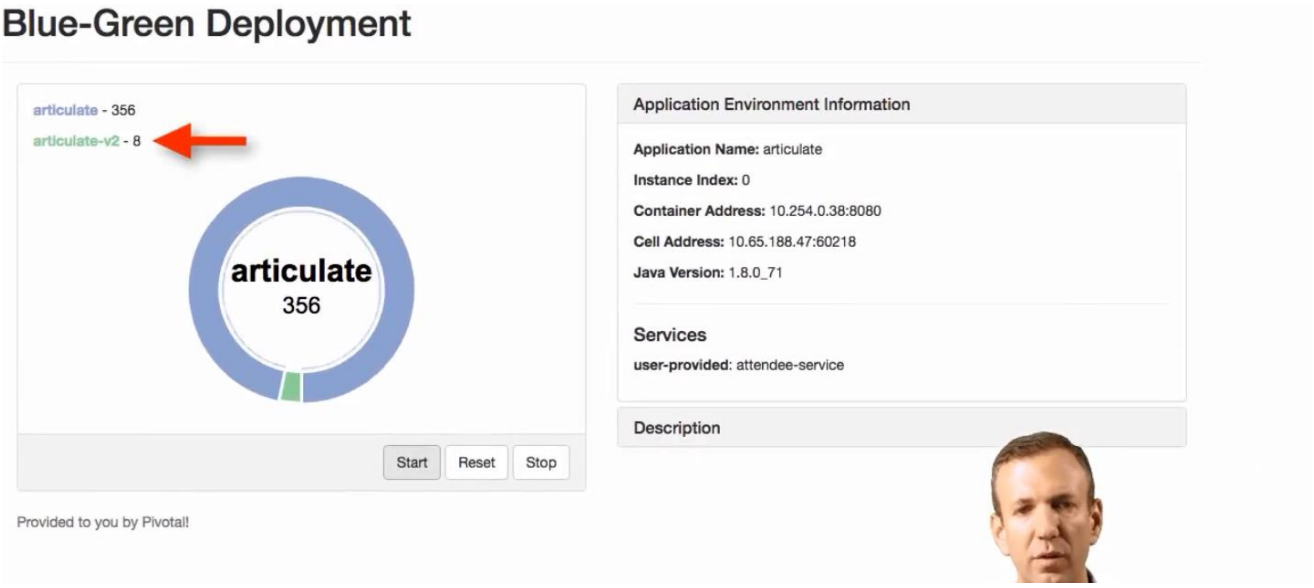
```
DROBERTS-MBPRO:articulate droberts$ cf routes
Getting routes for org dave / space dev as droberts@pivotal.io ...

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$
```

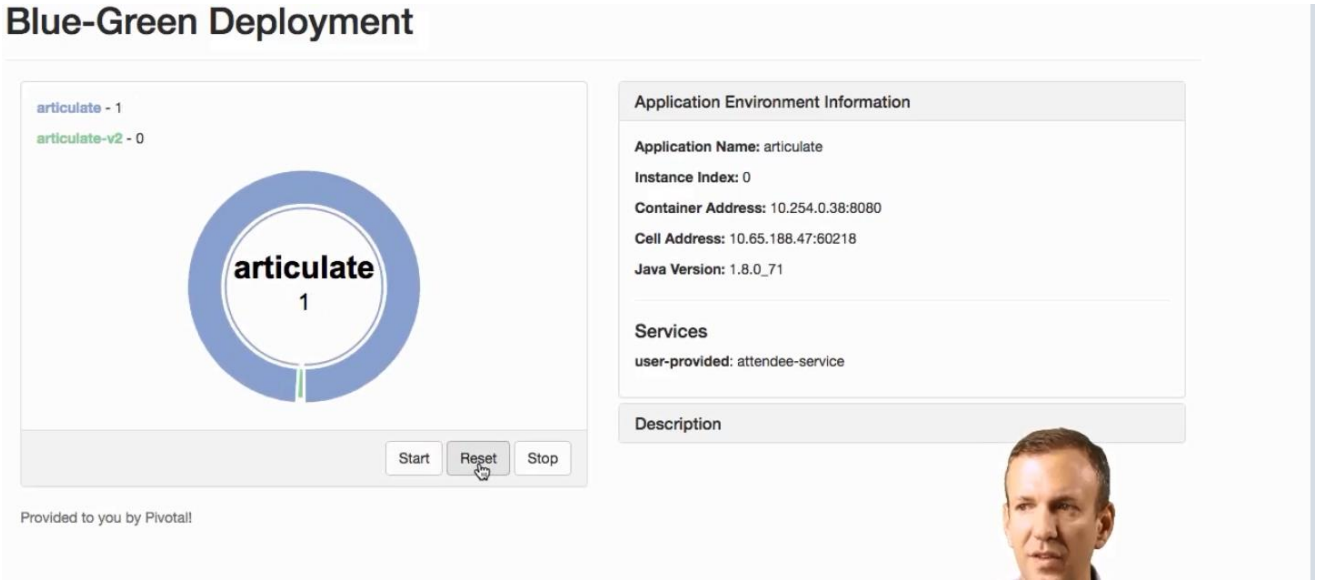
```
ticulate droberts$ cf map-route articulate-v2 cfapps.haas-39.pez.pivotal.io -n articulate-turbosupercharged-spinnere
```

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-turbosupe
t
Creating route articulate-turbosupercharged-spinneret.cfapps.haas-39.pez.pivotal.io for org dave / space dev as drob
.
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 /
berts@pivotal.io...
OK
DROBERTS-MBPRO:articulate droberts$
```



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

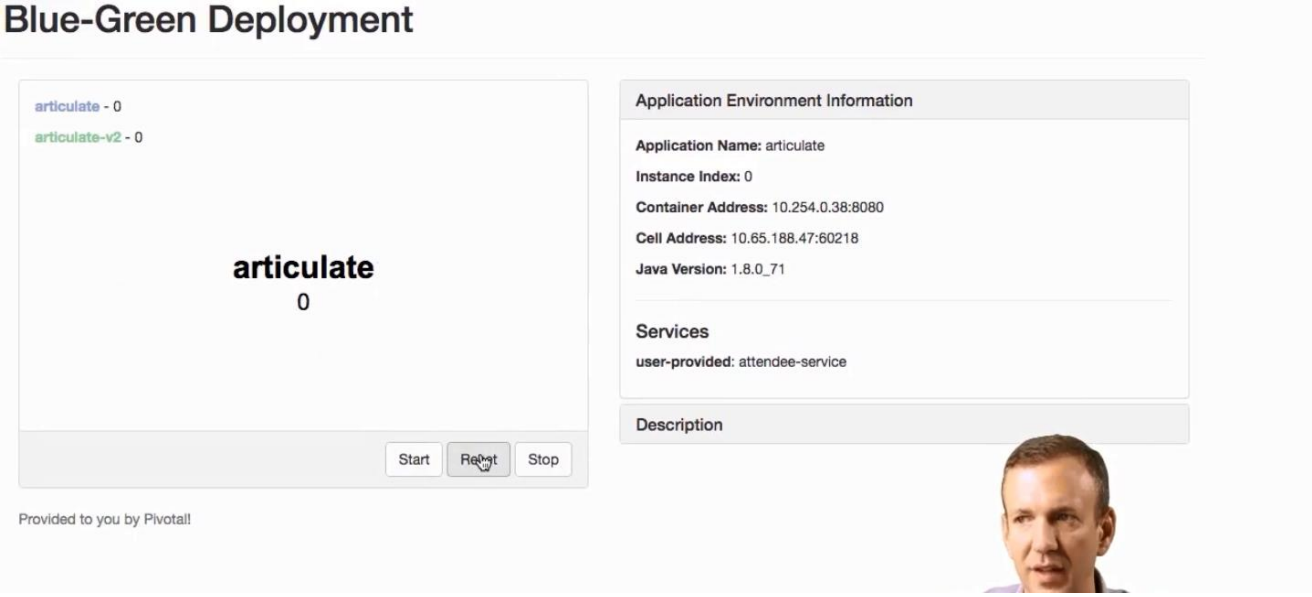


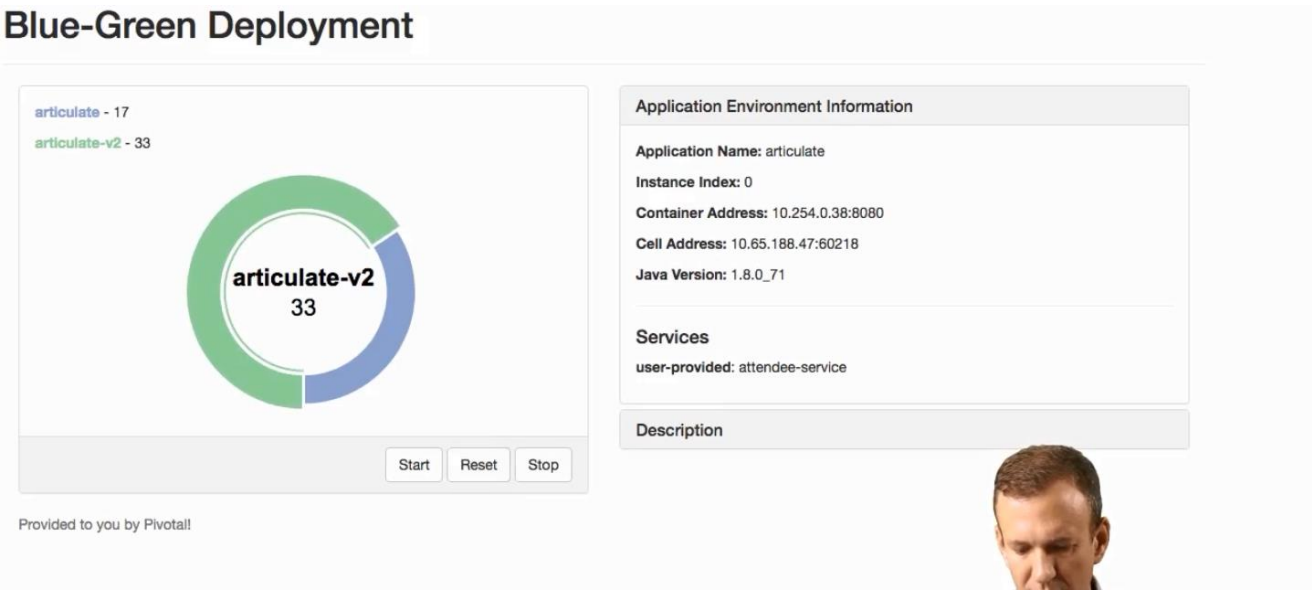
Let us rest this and see how the traffic is distributed

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

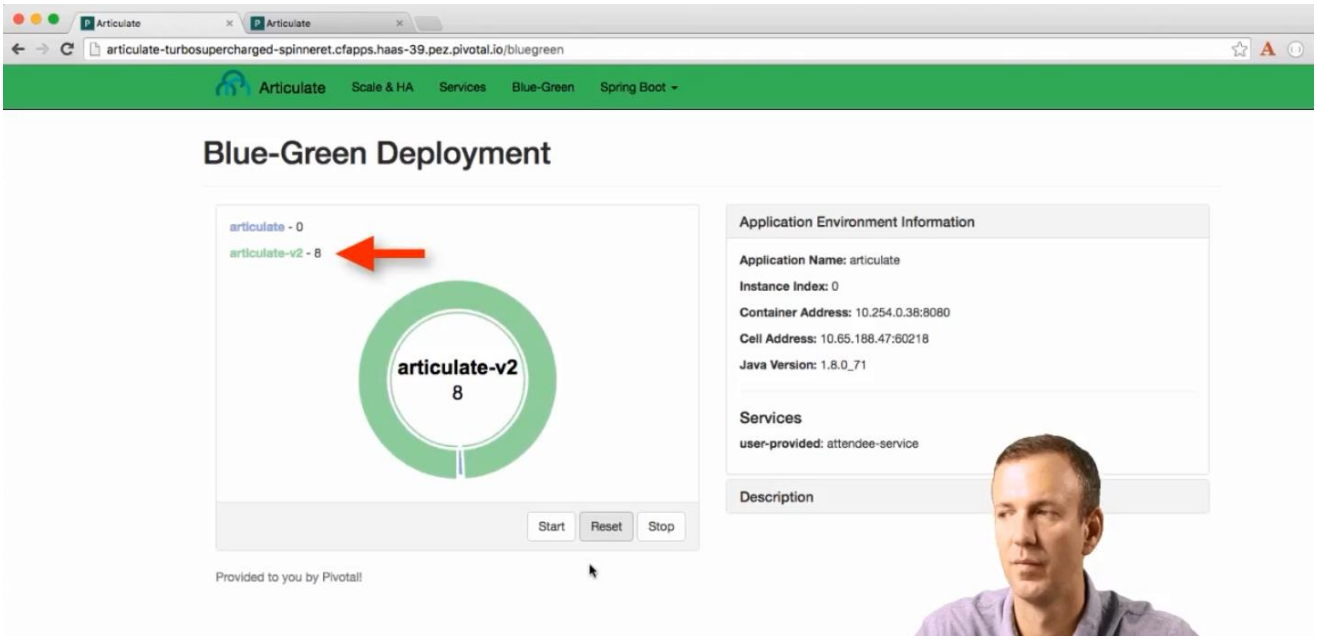




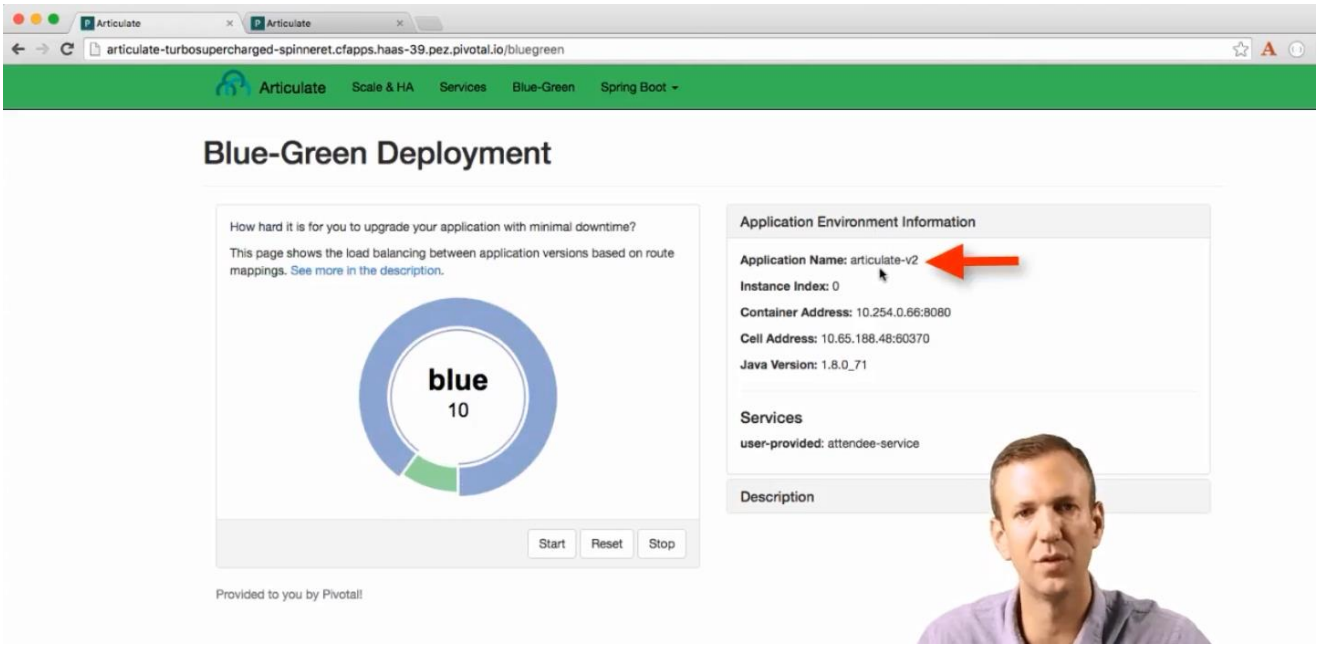
The v2 version is now getting more traffic.

```
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.



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



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-spinne
ret-temp
Removing route articulate-turbosupercharged-spinneret-temp.cfapps.haas-39.pez.pivotal.io from app articulate-v2 in org dave / space d
ev 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.