Printout

Wednesday, August 16, 2017 1:35 PM

Cumulus Education Lab

Blue Green

(../..)

Purpose of this lab

In this lab, you will deploy a new version of an application with zero downtime, using a blue-green deployment.

Exercises

Setup

If you do not already have the articulate application cloned to your machine, clone it and push it to Cloud Foundry.

```
git clone https://github.com/pivotal-education/pcf-articulate-code.git
cd pcf-articulate-code
./mvnw package
cf push articulate -p target/articulate-0.0.1-SNAPSHOT.jar -m 512M --rand
om-route
```

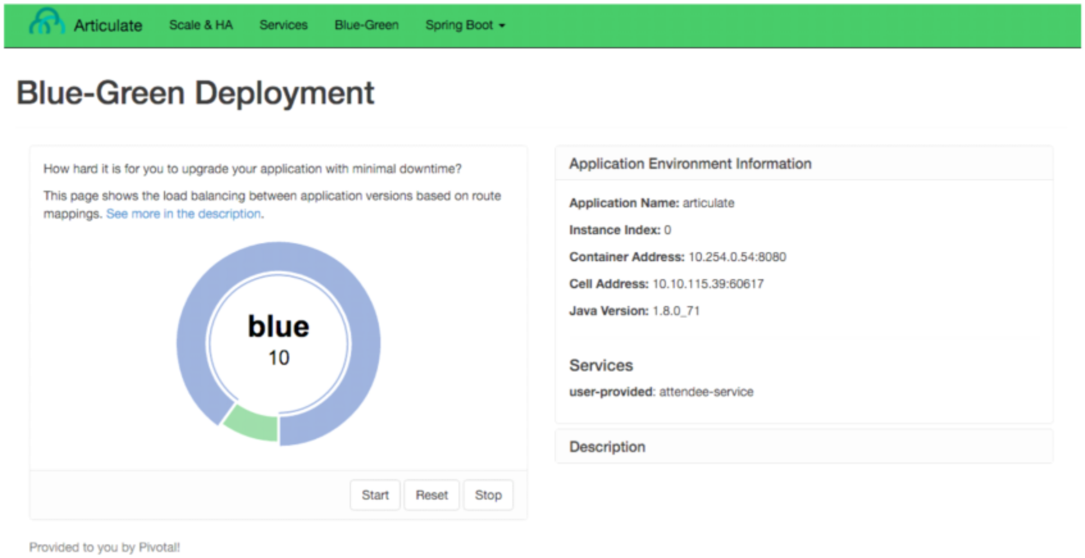
To simulate a blue-green deployment, scale `articulate` to multiple instances.

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

Perform a Blue-Green Deployment

1. Read about using Blue-Green Deployments to reduce downtime and risk (https://docs.pivotal.io/pivotalcf/devguide/deploy-apps/blue-green.html).

2. In a browser, navigate to the `articulate` `Blue-Green` page (e.g. https://articulate-heartsickening-elegance.pcfi1.fe.gopivotal.com/bluegreen; your route will be different).



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.

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

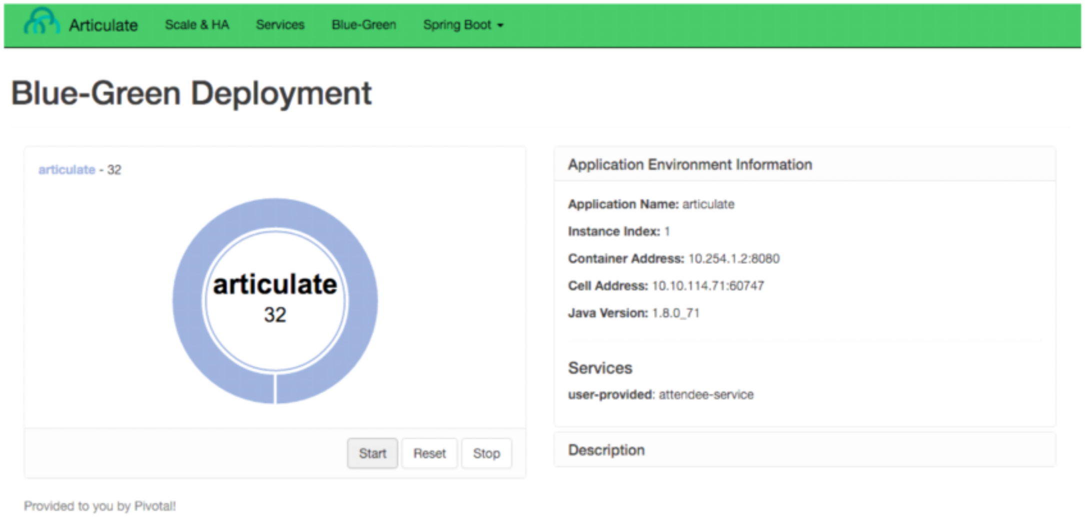
Description

Provided to you by Pivotal!

3. Assume that the deployed application is version 1. To simulate traffic, press the `Start` button.

   Leave this open as a dedicated tab in your browser.

4. Observe the existing application handling all the web requests.



Blue-Green Deployment

articulate - 32

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!

5. Record the subdomain (`host`) for the `articulate` application. This is the production route. You will use this in the next step. For example:

   ```
   cf routes
   ```

   ```
   Getting routes as droberts@pivotal.io ...

   space   host                                       domain
             apps
   dev     articulate-heartsickening-elegance         pcfi1.fe.gopivo
   tal.com   articulate
   ```

6. Now let's `push` the next version of `articulate` .

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

   For example (your subdomain will be different):

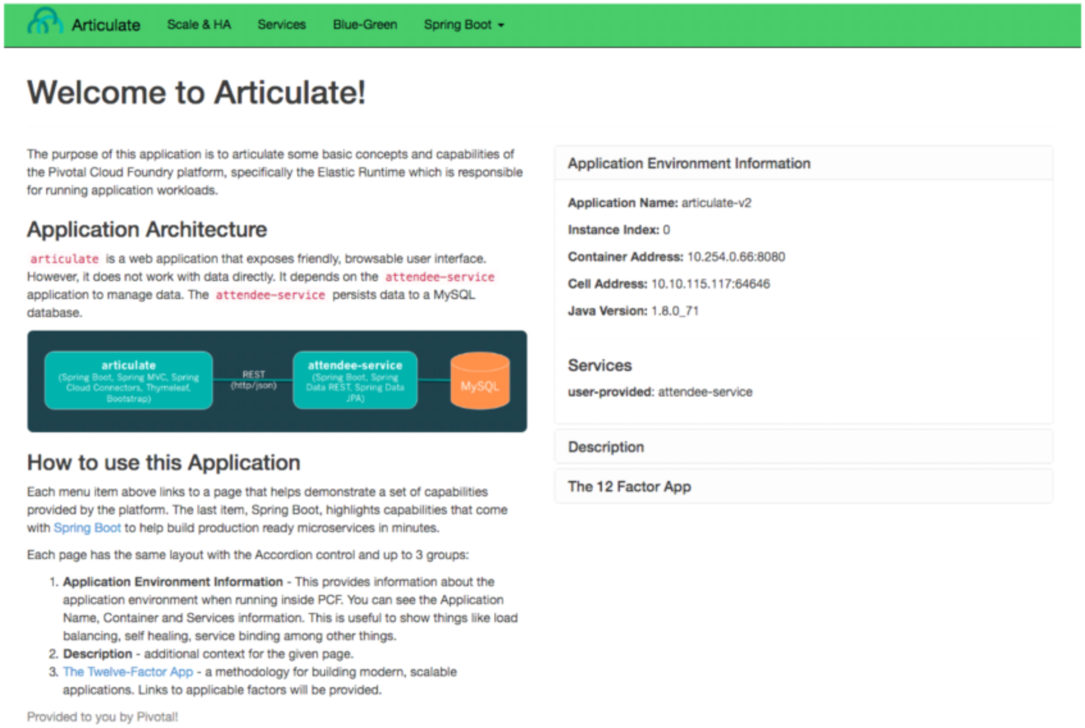
   ```
   cd ~/workspace/articulate/
   cf push articulate-v2 -p target/articulate-0.0.1-SNAPSHOT.jar -m 512
   M -n articulate-heartsickening-elegance-temp --no-start
   ```

7. Start the application.

   ```
   cf start articulate-v2
   ```

8. Now we have two versions of our app deployed.

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


Articulate
Scale & HA
Services
Blue-Green
Spring Boot
articulate
(Spring Boot, Spring MVC, Spring Cloud Connectors, Thymeleaf, Bootstrap)
REST
(http/json)
attendee-service
(Spring Boot, Spring Data REST, Spring Data JPA)
MySQL

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

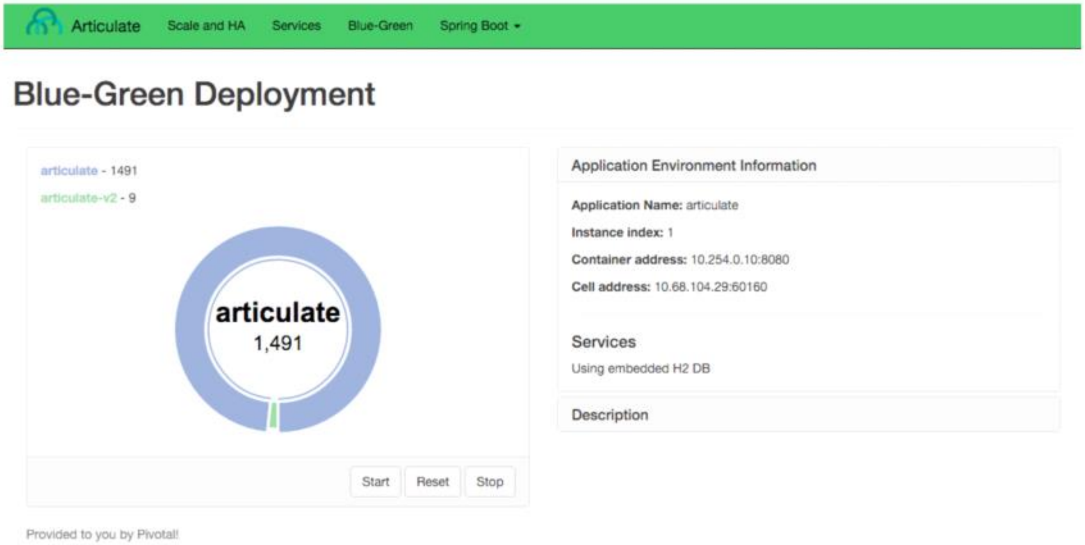
At this point in the deployment process, you could do further testing of the version you are about to release before exposing customers to it. This is also a good time to run tests against this new version of the application to make sure it works as expected.

9. Assume you are ready to start directing production traffic to version 2. You need to map the production route to `articulate-v2`.

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

   ```
   cf map-route articulate-v2 pcfi1.fe.gopivotal.com -n articulate-heartsickening-elegance
   ```

10. Return to the browser tab where you started the load. You should see requests are now being sent to version 2.



Blue-Green Deployment

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!

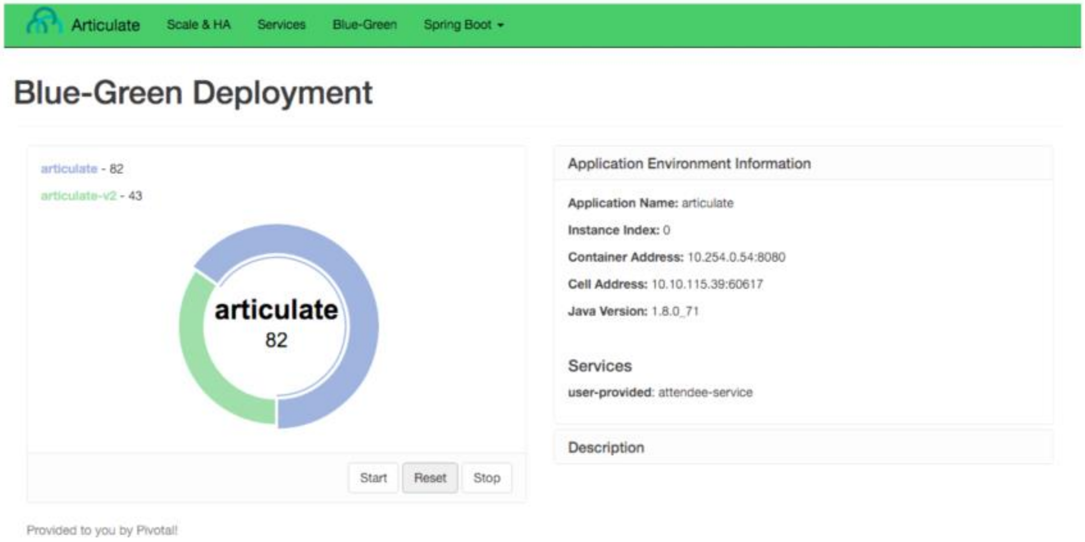
11. Press the `Reset` button to see how the load gets distributed across app instances.

    Run the `cf apps` in the terminal.

```
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-v2           started           1/1         512M       1G
   ...
```

If your app instances and state are similar to the above, you should see about a third of the requests going to version 2.



Blue-Green Deployment

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!

12. Move more traffic to version 2.

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

If you `Reset` the load generator, you will see 2/3 of the traffic go to `articulate-v2`.

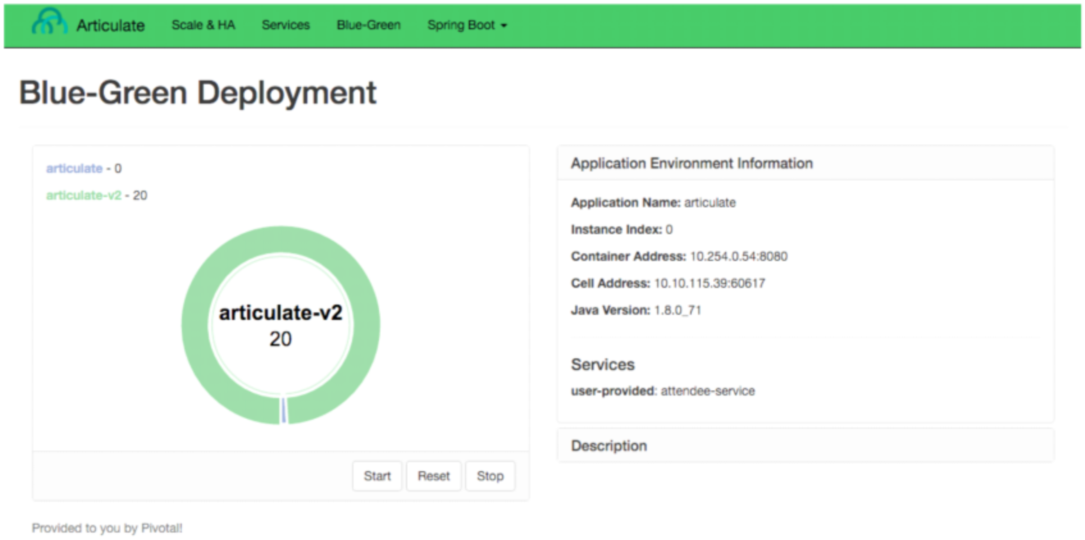
13. Move all traffic to version 2.

Remove the production route from the `articulate` application.

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

```
cf unmap-route articulate pcfi1.fe.gopivotal.com -n articulate-heart
sickening-elegance
```

If you `Reset` the load generator, you will see all the traffic goes to `articulate-v2`.



Blue-Green Deployment

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.

14. Remove the temp route from the `articulate-v2` application.

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

    ```
    cf unmap-route articulate-v2 pcfi1.fe.gopivotal.com -n articulate-he
    artsickening-elegance-temp
    ```

Congratulations! You performed a blue-green deployment.

Questions

- How would a rollback situation be handled using a blue-green deployment?
- What other design implications does running at least two versions at the same time have on your applications?
- Do you do blue-green deployments today? How is the process you just went through different?

Cleanup

Reset your environment.

1. Delete the `articulate` application.

```
cf delete articulate
```

2. Rename `articulate-v2` to `articulate`.

```
cf rename articulate-v2 articulate
```

3. Restart `articulate`.

```
cf restart articulate
```

4. Scale down.

```
cf scale articulate -i 1
```