|  |  |
| --- | --- |
| **cloudenabled** | |
| **Author** | **Sushil Kumar** |
| **Documentation Version** | **V1.0** |

**Pivotal Cloud Foundry-Developer**

**Pivotal Cloud Foundry Developer – Services**

#### Table of Contents

* [What you will learn](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#what-you-will-learn:d680e8a854a7cbad6d490c445cba2eba)
* [Exercises](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#exercises:d680e8a854a7cbad6d490c445cba2eba)
  + [Review articulate dependencies](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#review-articulate-dependencies:d680e8a854a7cbad6d490c445cba2eba)
  + [Push the attendee-service application](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#push-the-attendee-service-application:d680e8a854a7cbad6d490c445cba2eba)
  + [Add a Managed Service](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#add-a-managed-service:d680e8a854a7cbad6d490c445cba2eba)
    - [How does this work?](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#how-does-this-work:d680e8a854a7cbad6d490c445cba2eba)
    - [Questions](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#questions:d680e8a854a7cbad6d490c445cba2eba)
  + [Add a User Provided Service Instance](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#add-a-user-provided-service-instance:d680e8a854a7cbad6d490c445cba2eba)
    - [Questions](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#questions-1:d680e8a854a7cbad6d490c445cba2eba)
* [Beyond the class](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/services/index.html#beyond-the-class:d680e8a854a7cbad6d490c445cba2eba)

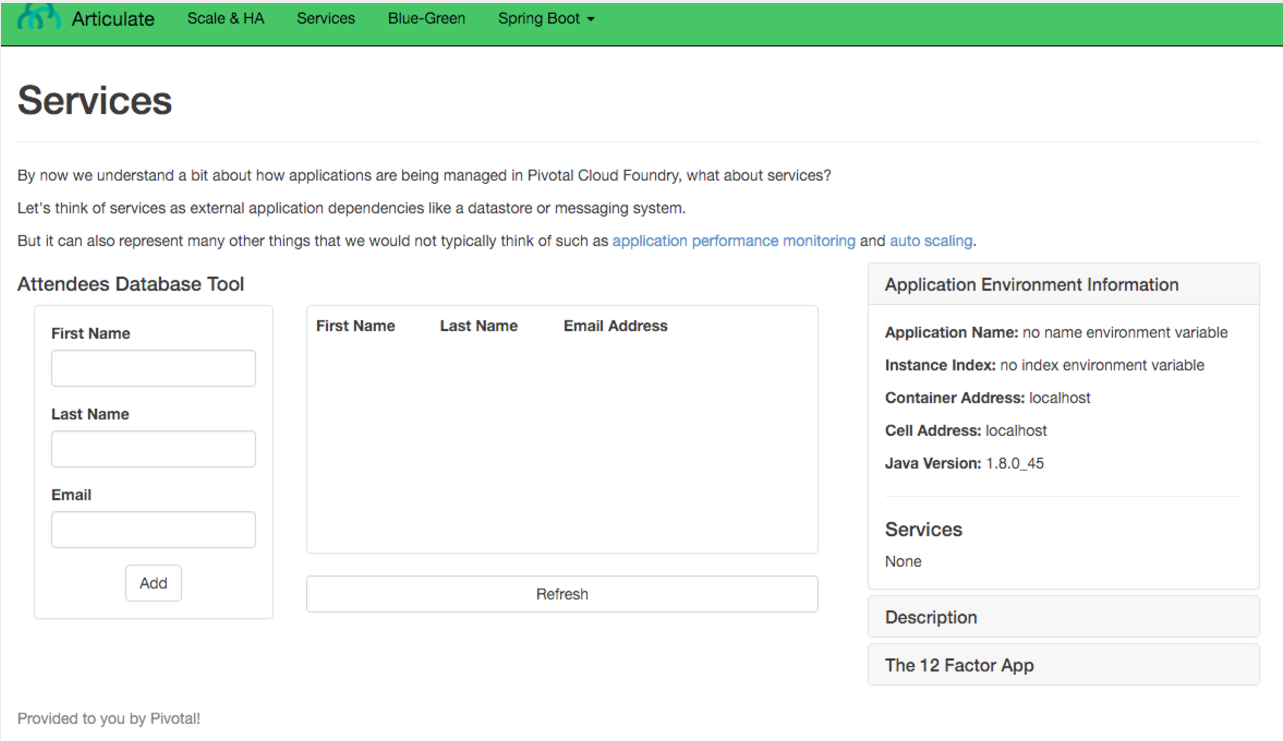
# What you will learn

* How to create a managed service instance
* How to create a user provided service instance
* How to bind an application to a service instance

# Exercises

## Review articulate dependencies

articulate exposes functionality to add attendees on the Services page. However, articulate doesn’t do this alone. It makes REST calls to the attendee-service application. To learn more about services, let’s provision the attendee-serviceapplication.



## Push the attendee-service application

1) [Download](https://content.enablement.pivotal.io/pivotal-cloud-foundry-developer/resources/demo-apps/attendee-service-0.0.1-SNAPSHOT.jar) the attendee-service application. Copy the file to folder: ~/pivotal-cloud-foundry-developer-workshop/attendee-service/ (~ is shorthand for the home directory in Linux, Mac and Unix based operating systems). You will need to create this directory in your [home](https://en.wikipedia.org/wiki/Home_directory) directory.

**NOTE:** If your browser warns you about downloading this file please proceed to download it.

[Source](https://github.com/pivotal-enablement/attendee-service) is not required, but you may be curious how it works as you move through the labs.

2) Push the attendee-service application.

$ cd ~/pivotal-cloud-foundry-developer-workshop/attendee-service/

$ cf push attendee-service -p ./attendee-service-0.0.1-SNAPSHOT.jar -m 512M --random-route

Does attendee-service start up correctly? Why not?

## Add a Managed Service

1) Review the [documentation](http://docs.pivotal.io/pivotalcf/devguide/services/managing-services.html) on managing service instances

2) Review what services are available in the marketplace.

$ cf marketplace

3) Provision a MySql service instance.

**Pivotal Cloud Foundry:**

$ cf create-service p-mysql 100mb-dev attendee-mysql

**Pivotal Web Services:**

$ cf create-service cleardb spark attendee-mysql

4) Now we need to bind the application with the service instance.

$ cf **bind**-service attendee-service attendee-mysql

You can ignore the “TIP: Use ‘cf restage attendee-service’ to ensure your env variable changes take effect” message at this time.

5) Restart the application.

$ cf restart attendee-service

6) View the attendee-service in a browser.

You should see response similar to the following (pic is using the [JSON Formatter for Chrome](https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa?hl=en)):



### How does this work?

1) Read about how twelve-factor apps handle [backing services](http://12factor.net/backing-services) and [configuration](http://12factor.net/config).

2) Read about [VCAP\_SERVICES](https://docs.pivotal.io/pivotalcf/devguide/deploy-apps/environment-variable.html#VCAP-SERVICES).

3) View the environment for attendee-service.

$ cf env attendee-service

4) Different languages/frameworks will have various ways to read environment variables. attendee-service takes advantage of a [Java Buildpack](https://github.com/cloudfoundry/java-buildpack) feature called [Auto-Reconfiguration](https://github.com/cloudfoundry/java-buildpack-auto-reconfiguration) that will automatically re-write bean definitions to connect with services bound to an application.

### Questions

* After binding a service to an application why is the application restarted/restaged?
* In this case, why could we restart vs restage?

## Add a User Provided Service Instance

In the enterprise, not all services will be provisioned by Pivotal Cloud Foundry.

For example, consider your Oracle RAC cluster. How can we connect our applications running on Pivotal Cloud Foundry to these external systems?

Additionally, how can we easily connect applications together running on the platform?

articulate's default configuration for the attendee-service uri is http://localhost:8181/attendees. The subsequent steps will allow you to override the default configuration with your own.

1) Read about [user provided service instances](http://docs.pivotal.io/pivotalcf/devguide/services/user-provided.html).

2) Create a user provided service instance. This will create an interactive prompt. Don’t use the literal below for the value of uri, use your attendee-service uri. Also make sure to specify http and include /attendees in the path.

**NOTE:** This will not work with https.

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

uri> http://attendee-service-surfy-glt.pcfi1.fe.gopivotal.com/attendees

3) Bind articulate to the attendee-service user provided service.

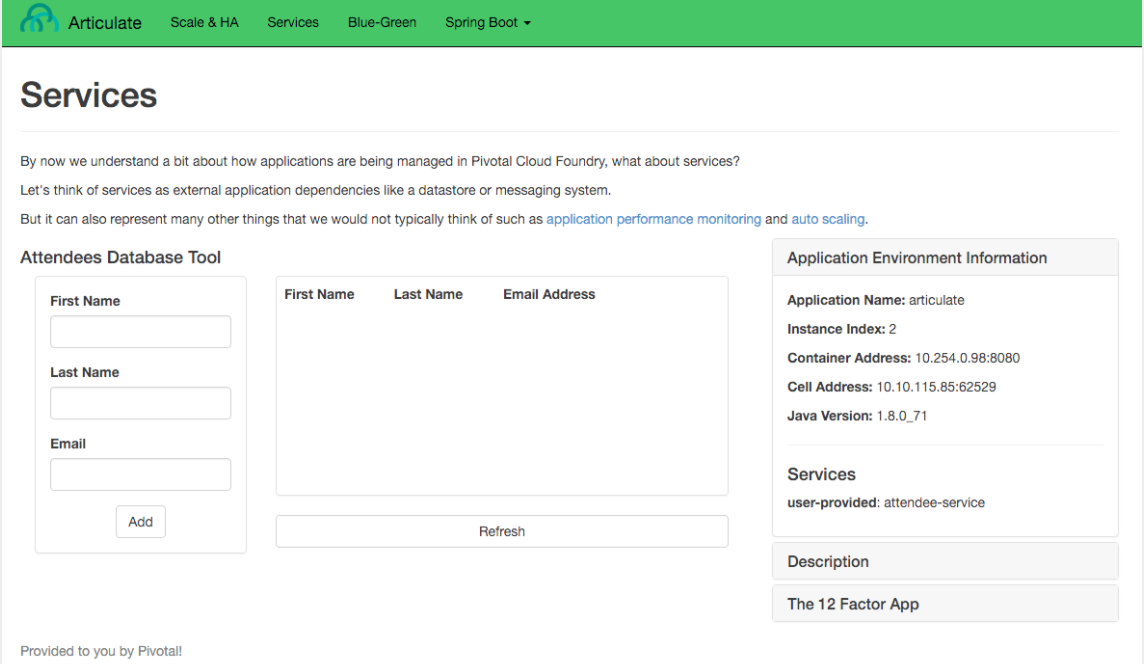
$ cf **bind**-service articulate attendee-service

You can ignore the “TIP: Use ‘cf restage articulate’ to ensure your env variable changes take effect” message at this time.

4) Restart the application.

$ cf restart articulate

5) Refresh the articulate Services page. You can now see the attendee-service listed under Services.



6) Review the environment.

$ cf env articulate

7) Add some attendees.

**NOTE:** If you can’t add attendees review the articulate logs and the user provided service instance configuration.

### Questions

* From an application perspective, are managed services instances different from user provided service instances?