(../..)

Purpose of this lab

- How to use the `app-autoscaler` service
- Estimated Time: 25 minutes

Setup

1. Ensure the `articulate` application has one application instance running.

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

Provision and Bind an Autoscaler Service Instance

1. Read the documentation about App Autoscaling (http://docs.pivotal.io/pivotalcf/appsman-services/autoscaler/using-autoscaler.html).
2. Review what's in the marketplace.

```
cf marketplace
```

3. Create a autoscaler service instance.

```
cf create-service app-autoscaler standard autoscaler
```

4. Bind the service to `articulate`.

```
cf bind-service articulate autoscaler
```

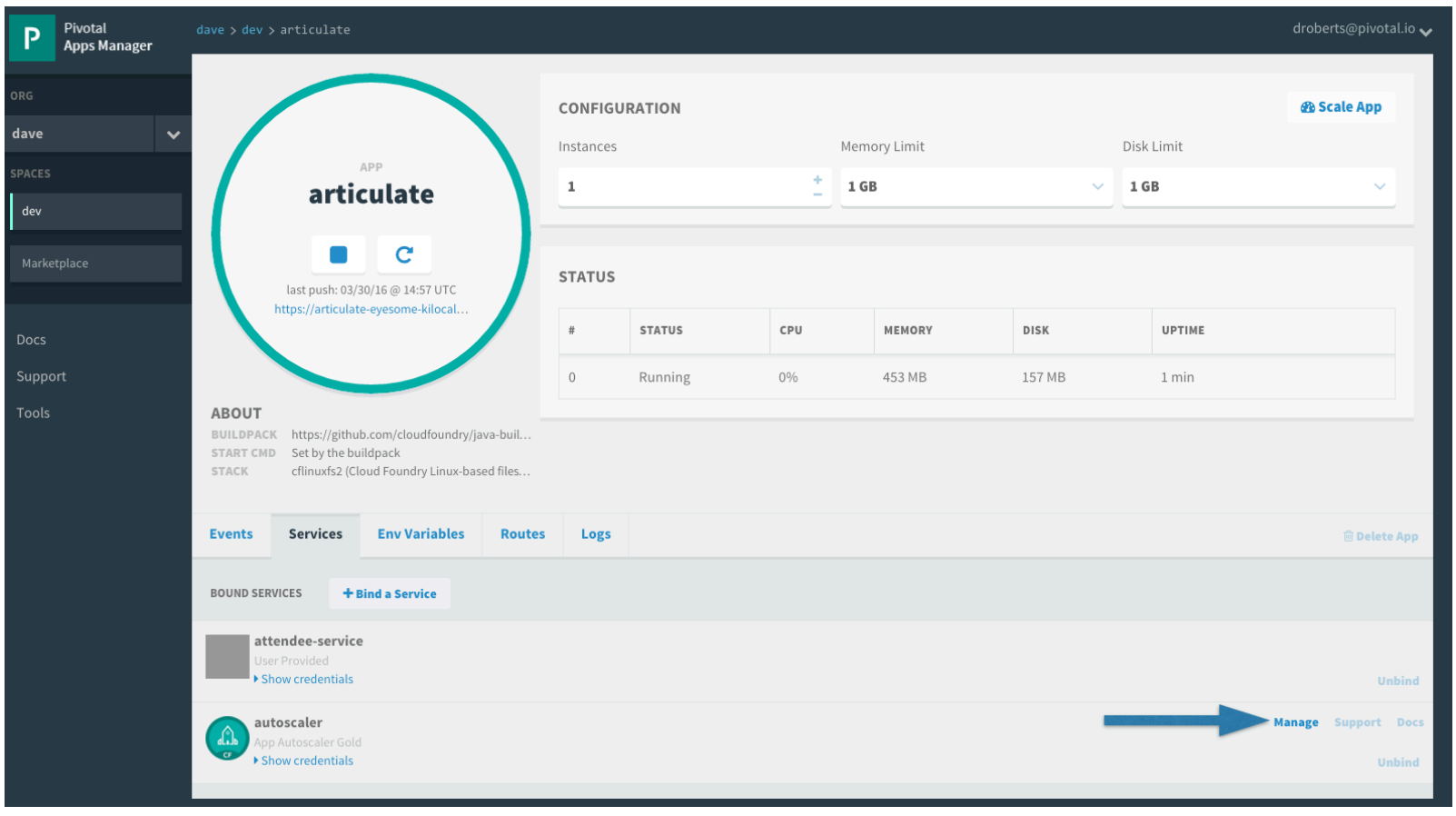
You can ignore the "TIP: Use 'cf restage articulate' to ensure your env variable changes take effect" message at this time.

5. Restart the application.

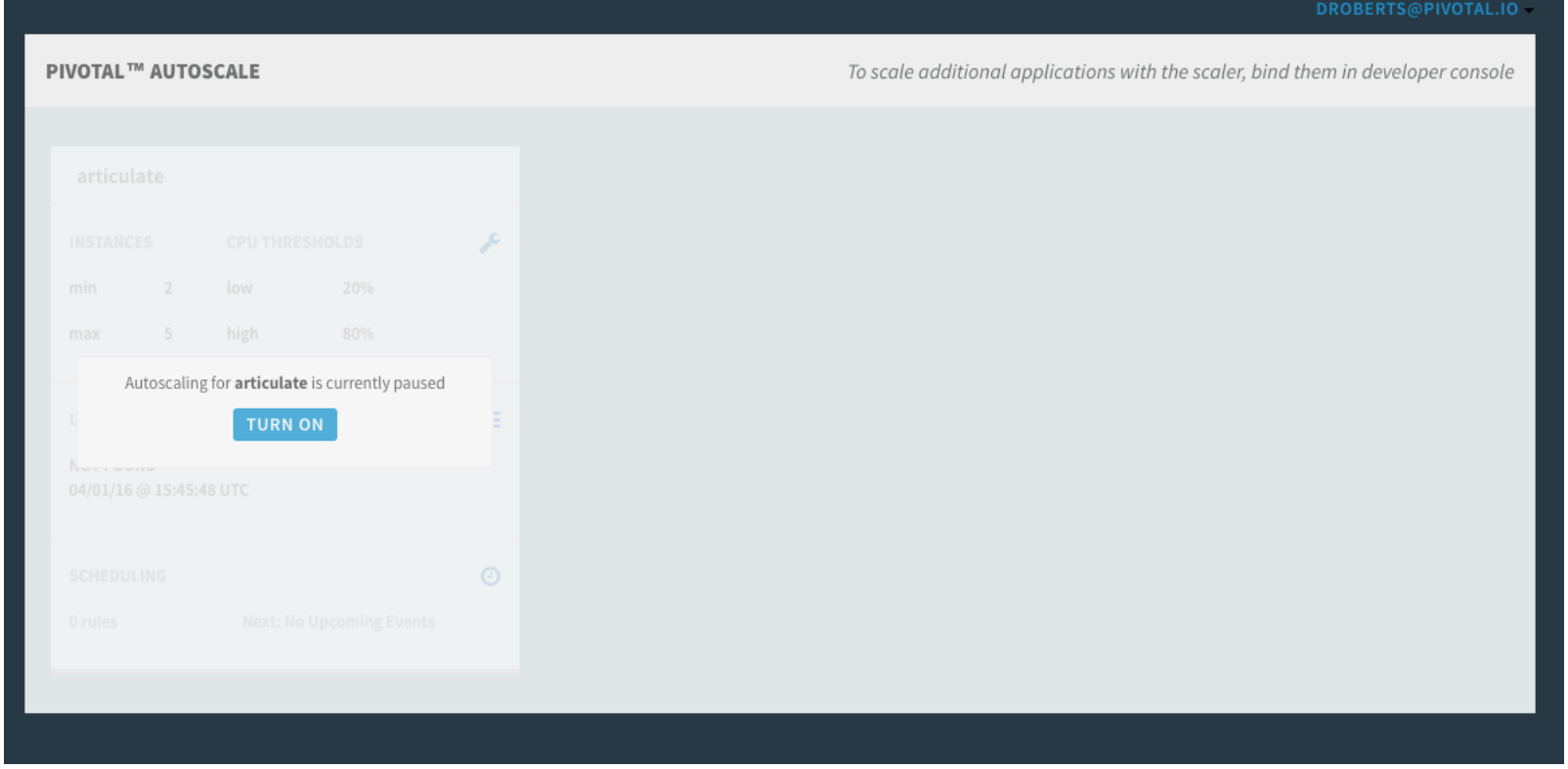
```
cf restart articulate
```

Configuring Autoscaling Service Instance

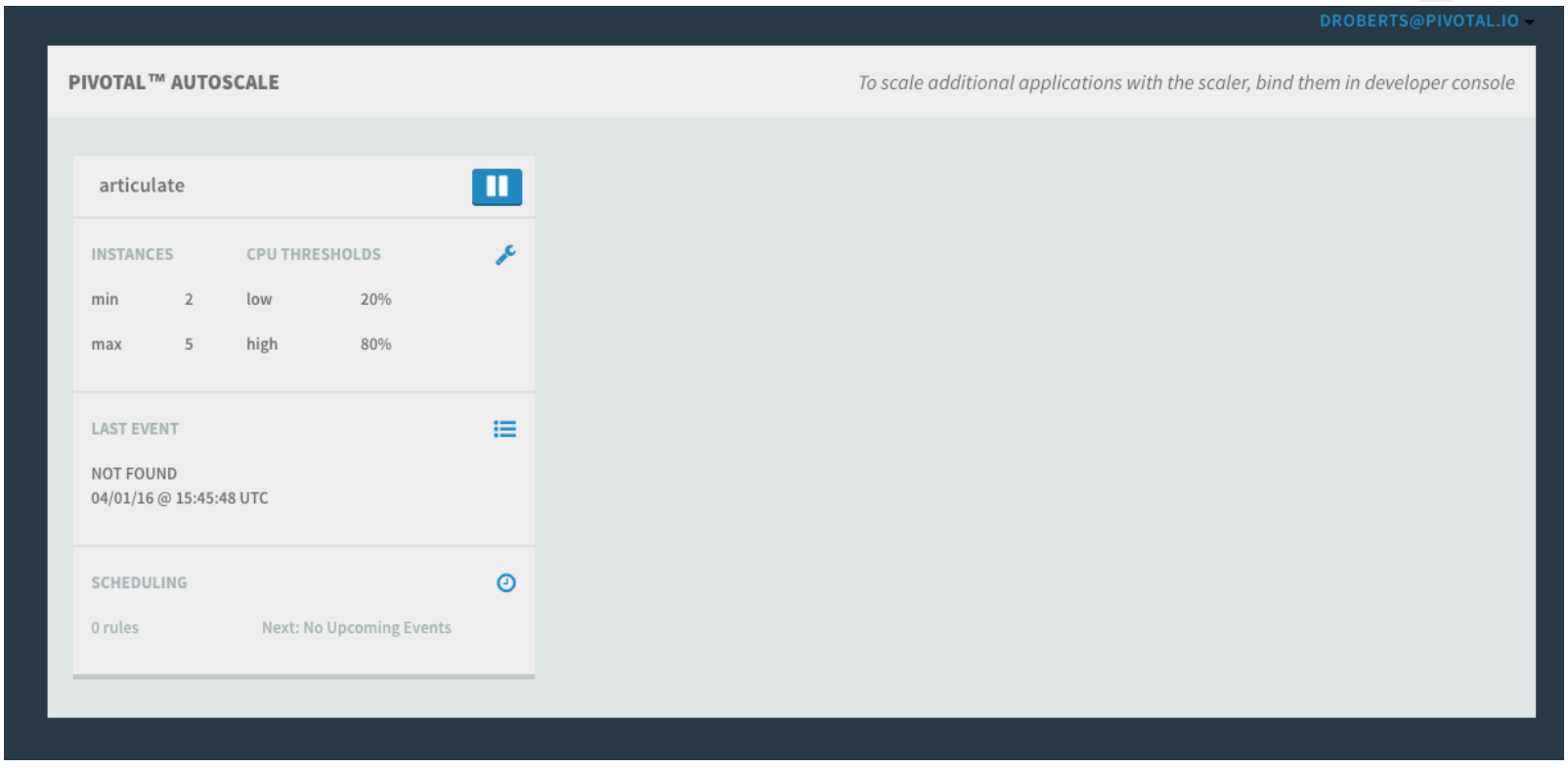
1. View `articulate` in Apps Manager.
2. Navigate to the `Services` tab.
    1. Click the `Manage` link for the autoscaler (this will open a new tab).



2. `Turn on` autoscaling for `articulate`.



We will use policy defaults. Note that the default policy for minimum instances is `2`.



3. Return to Apps Manager and observe the number of instances increase from `1` to `2`.

Generate Load and Observe the Results

1. Download Apache JMeter (http://jmeter.apache.org/download_jmeter.cgi). It will be used to generate load. Review the JMeter getting started (http://jmeter.apache.org/usermanual/get-started.html) directions.

   Helpful hints for `Linux and Mac`:

2. Add execute permissions to the jmeter script

```
chmod +x $WHERE_YOU_EXTRACTED_THE_ZIP/bin/jmeter.sh
```

3. Start Jmeter by executing

   ```
   $WHERE_YOU_EXTRACTED_THE_ZIP/bin/jmeter.sh
   ```

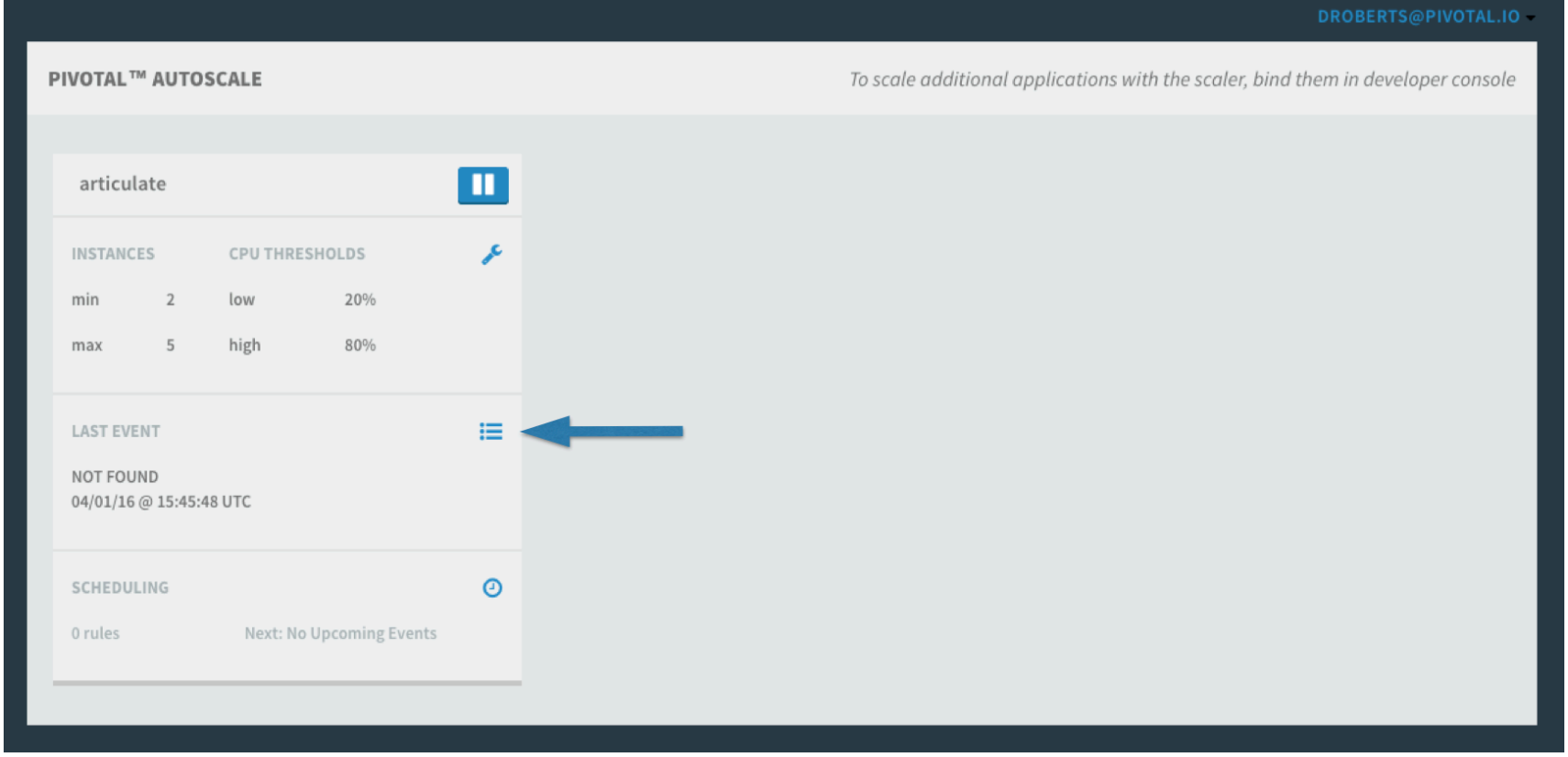
   Helpful hints for `Windows`: Start Jmeter by double clicking the following file in Windows Explorer

   ```
   $WHERE_YOU_EXTRACTED_THE_ZIP/bin/jmeter.bat
   ```

4. Download the load-gen.jmx (load-gen.jmx) file.
5. Open the load-gen.jmx file with the JMeter GUI.
6. Expand the content of the JMeter test plan by:
7. Selecting the `Load Generator` test plan on the left pane.
8. Going to the menu: `Options` → `Expand All`.
9. Edit the test plan to point to the `/service` endpoint on your application. For example, if your application was deployed to `articulate.example.com` you would set the *Server Name or IP* field on the *HTTP Request Defaults* item to `articulate.example.com/service`.
10. Save the test plan: `File` → `Save`.
11. Run the test plan: `Run` → `Start`.
12. Use Apps Manager and the `cf` CLI to observe your service scale up and back down based on load.

    NOTE: You may see `articulate` crash when placing load on it in this scenario. This is okay. We are running the application with a lower memory setting so that we can scale within quota limits.

13. Stop the test plan: `Run` → `Stop`.
14. Review Autoscale history.



Clean up

1. Unbind the `autoscaler` service instance.

```
cf unbind-service articulate autoscaler
```

2. Delete the `autoscaler` service instance.

```
cf delete-service autoscaler
```

3. Scale `articulate` back to original settings.

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

4. Restart `articulate`.

```
cf restart articulate
```

Questions

- How do you handle autoscaling today?
- What 12 factor principles are important when it comes to scaling?
- How do you handle scaling at the data layer?

(https://pivotal.io)

course version: 1.5.3