

TAP Hands-On Workshop

Module 2: Visualizing Your Data with an App

Overview

This module uses a Java application to read a dataset from Hadoop Distributed File System (HDFS) that was created in Module 1: Performing Analytics on Your Data. The module uses Dataset-Reader to fetch data that was modified in the first module. The data is then presented to the user in a graphical form.

Workshop Steps	Time Required
Step 1	Varies based on your current environment
Steps 2 - 3	10 minutes
Step 4	20 minutes
Step 5	10 minutes
Step 6	10 minutes
Steps 7 - 9	10 minutes
Total Workshop Time	60 minutes

Setup

The app developer needs to prepare his or her local environment/computer with the tools required to perform the analytics and view the results.

Important: You need to have access to an instance or to set up an instance of Analytics Toolkit (ATK).

- The software you need includes:
 - **Git** - a widely adopted version-control system for software development. <https://git-scm.com/downloads>
 - **Apache Maven** - a software project management and comprehension tool which manages a project's build, reporting and documentation. <https://maven.apache.org/download.cgi>
 - **Java 8** - a programming language expressly designed for use in the distributed environment of the Internet. <https://java.com/en/download/>
 - **Java IDE** - Java Integrated Development Environment (IDE) software application that enables the writing and debugging of Java programs.
 - **Eclipse** - an integrated development environment. It contains a base workspace for customizing the environment. <https://eclipse.org/downloads/>

Objectives

After completing this module, you will have:

- Demonstrated how to deploy an app to Trusted Analytics Platform (TAP) and viewed the app in a web browser.
- Identified the code that allows the app to access data from HDFS.
- Exhibited the ability to assemble an app using Maven.
- Demonstrated how to use the Cloud Foundry console to create and view a service instance.
- Learned how to access an instance of ATK and deploy apps on TAP.
- Demonstrated the ability to install multiple software tools needed to build and operate domain-specific applications.

Module Steps

1. Install all necessary developer tools listed in Setup.
2. Visit the Dataset-Reader app.
3. Review the Dataset-Reader app flow, the sample folders and the app code with Java IDE. Specifically, check how the app accesses data from HDFS and creates the graphic output.
4. Clone the Dataset-Reader app using *git clone*, compile the app using *mvn compile* and create a Java package using the *mvn package*.
5. Log into the TAP and access the App Developer tool. Set up your environment and edit the manifest.yml file.
6. Deploy the app on TAP. Deploy the app to the platform using the *cf push* command and confirm the app is deployed on TAP.
7. Try to view the output of Dataset-Reader in a browser. Why are you not able to view the output?
8. Update the environment and set the link to the input dataset. Use the *cf set-env* command to set the link and use *cf restage* to ensure changes take effect.
9. View the output of Dataset-Reader.

Important: The URLs and input commands provided in the workshop will change based on the instance used and the setup of your network. Check with the workshop instructor or consult with your development operations team (DevOps) to obtain the proper information.

Results

This section provides step-by-step instructions and results. The following text colors are used to make it easier to read:

- General-instruction text is in **Black**.
- Commands you input are in **Red**.
- The output from executed commands is in **Blue**.
- Links to the input dataset that changes with every execution are in **Green**.

1. **Install all necessary developer tools listed in Setup**

1.1. No additional instructions are provided for this step.

2. **Visit the Dataset-Reader app**










2.1. The app is located here:

<https://github.com/trustedanalytics/dataset-reader-sample>

3. Review Dataset-Reader

3.1. App flow:

- The dataset is uploaded through the Data Catalog into the platform. The file is then stored on the HDFS.
- The app developer performs analysis on the data using ATK within the TAP. The results are also stored on HDFS.
- The Dataset-Reader application is uploaded into the platform and binds to the file.
- The Dataset-Reader app presents the data graphically in a set of charts.

	Branch: master ▾	dataset-reader-sample / +
Removed dependency to the hdfs instance from the manifest		
 szymonbultrowicz authored 8 days ago		
 docs	Documentation update	
 license	Netflow data preparation	
 src	Removed dependency to the hdfs instance from the manifest	
 .gitignore	Added .classpath to gitignore	
 LICENSE.txt	Initial commit	
 README.md	Update README.md	
 pom.xml	DPNG-2126 Add support for binding a folder on HDFS to dataset-reader	

3.2. Sample folders:

- **docs** - Dataset-Reader graphic.
- **license** - verifies the licenses for mvn and Apache. It formats the license text information that is displayed in the app.
- **src** - contains the manifest.yml file, a number of java files and the InputStreamProviderTest.java file.

3.3. InputStreamProvider App code:

- **Code lines 1 - 15:** Displays the license information produced by the 3 XML files in the license folder.
- **Code line 16:** Compiles code for org.trustedanalytics.resourceserver.data.
- **Code lines 18 - 33:** Imports the necessary Apache filesystem, path and Java utilities.
- **Code lines 35 - 44:** Defines the input stream for the dataset and builds an HDFS URL based on the configuration settings supplied.
- **Code lines 57 - 73:** The app tries to open a path to the dataset. If the dataset is not found, it tries to find the directory path to the dataset. It then attempts to read the file. It displays the error message "Cannot read file," if the file is unreadable.
- **Code lines 74 - 82:** If the app is unable to locate the file, it returns the error message "Given path XXXX is neither file nor directory."

3.4. Review the App code:

- Download the InputStreamProvider app from:
<https://github.com/trustedanalytics/dataset-reader-sample/blob/master/src/main/java/org/trustedanalytics/resourceserver/data/InputStreamProvider.java>
- Open Eclipse and view the app.
- Check how the app accesses data from HDFS and creates the graphic output.

4. Clone, compile and create a Java package

4.1. Clone the Dataset-Reader app:

- To upload your project to GitHub, you need a folder for it to live in. Clone the Dataset-Reader repository by executing the command:

```
git clone https://github.com/trustedanalytics/dataset-reader-sample.git
```

- You should see similar results when you execute the *git clone* command:

```
Cloning into 'dataset-reader-sample'...
```

```
remote: Counting objects: 135, done.
```

```
remote: Compressing objects: 100% (4/4), done.
```

```
remote: Total 135 (delta 0), reused 0 (delta 0), pack-reused 130
```

```
Receiving objects: 100% (135/135), 442.52 KiB | 0 bytes/s, done.
```

```
Resolving deltas: 100% (31/31), done.
```

```
Checking connectivity... done.
```

4.2. Compile the Dataset-Reader app using the command:

- This step takes a considerable amount of time to complete.

```
mvn compile
```

- **Note:** If you prefer to run the code on your local computer or server, you will need to provide the path to your file location. Use this command to establish the path to your file:

```
FILE=<path_to_file> mvn spring-boot:run -Dspring.profiles.active=local
```

- You should see similar results when you execute the *mvn compile* command. (Many of the log output results are omitted.)

```
user@host:~/demo/dataset-reader-sample$ mvn compile
```

```
[INFO] Scanning for projects...
```

```
Downloading: https://repository.cloudera.com/artifactory/cloudera-repos/org/springframework/boot/spring-boot-starter-parent/
```

```
1.2.5.RELEASE/spring-boot-starter-parent-1.2.5.RELEASE.pom
```

```
[INFO] -----
```

```
[INFO] Building dataset-reader-sample 0.4.1-SNAPSHOT
```

```
[INFO] -----
```

```
[INFO] Checking licenses...
```

```
[INFO]
```

```
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ dataset-reader-sample ---
```

```
[INFO] Using 'UTF-8' encoding to copy filtered resources.
```

```

[INFO] Copying 1 resource
[INFO] Copying 15 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.3:compile (default-compile) @ dataset-
reader-sample ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 4 source files to /Users/ID/target/classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 07:13 min
[INFO] Finished at: 2015-08-17T13:42:46-07:00
[INFO] Final Memory: 36M/109M
[INFO] -----

```

4.3. Create a Java package using the command:

mvn package

- You should see similar results when you execute the *mvn package* command. (Many of the log output results are omitted.)

```

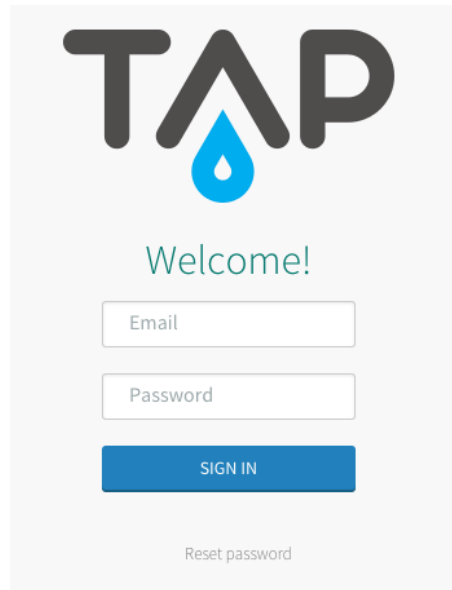
user@host:~/demo/dataset-reader-sample$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building dataset-reader-sample 0.4.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:copy-resources (copy-resources) @
dataset-reader-sample ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.075s
[INFO] Finished at: Thu Sep 10 16:18:41 CEST 2015
[INFO] Final Memory: 33M/292M
[INFO] -----

```

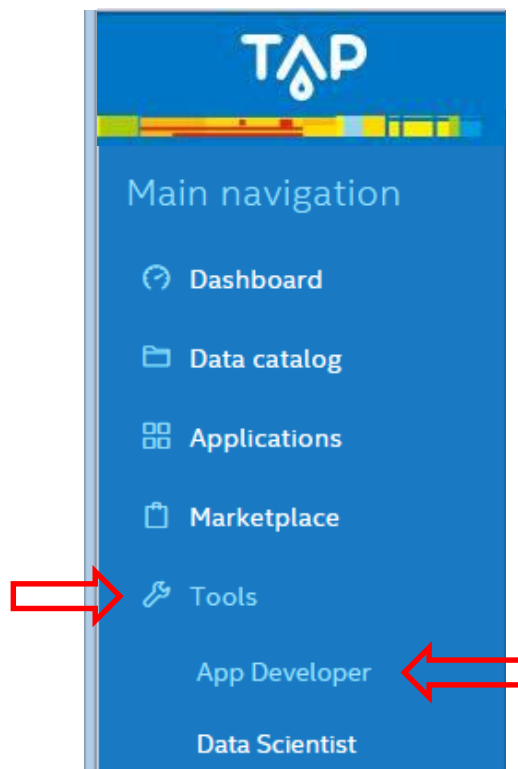
5. Deploy the app to the platform, and confirm it is deployed

5.1. Log into the platform:

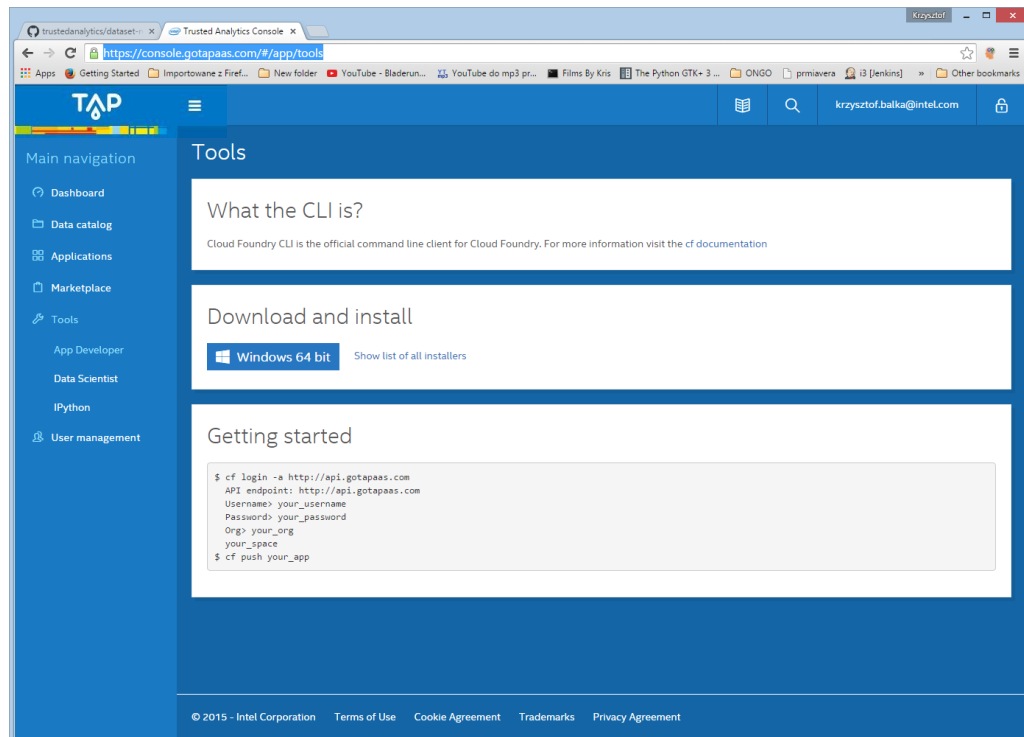
- Log into the TAP Cloud Foundry and set the proper organization and space using the URL: <https://console.gotapaas.com>
- **Note:** The URL will change, depending on the instance you are using.



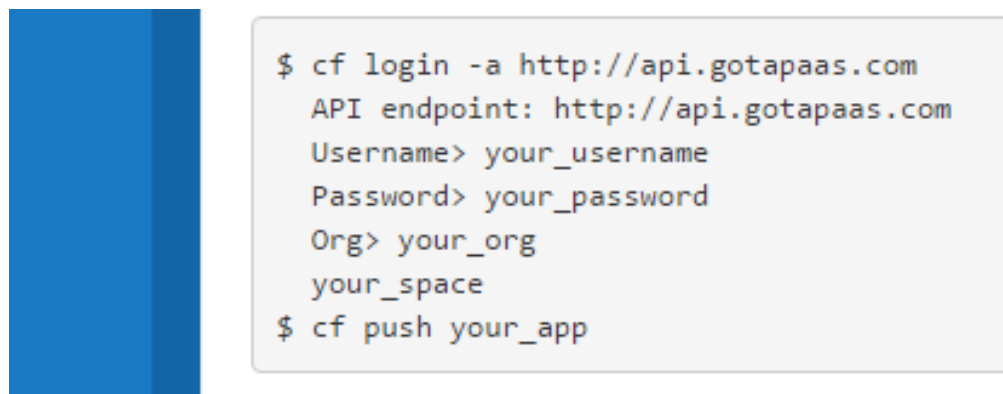
- In the Main navigation window, click Tools and then click App Developer.



- Check the platform api endpoint in the console view. (In this example, the api endpoint is `http://api.gotapass.com`):



- **Note:** The Cloud Foundry login command and CF Push command screen have been enlarged for easier viewing.



5.2. Set up the environment within the TAP Cloud Foundry App Developer window, by executing the commands:

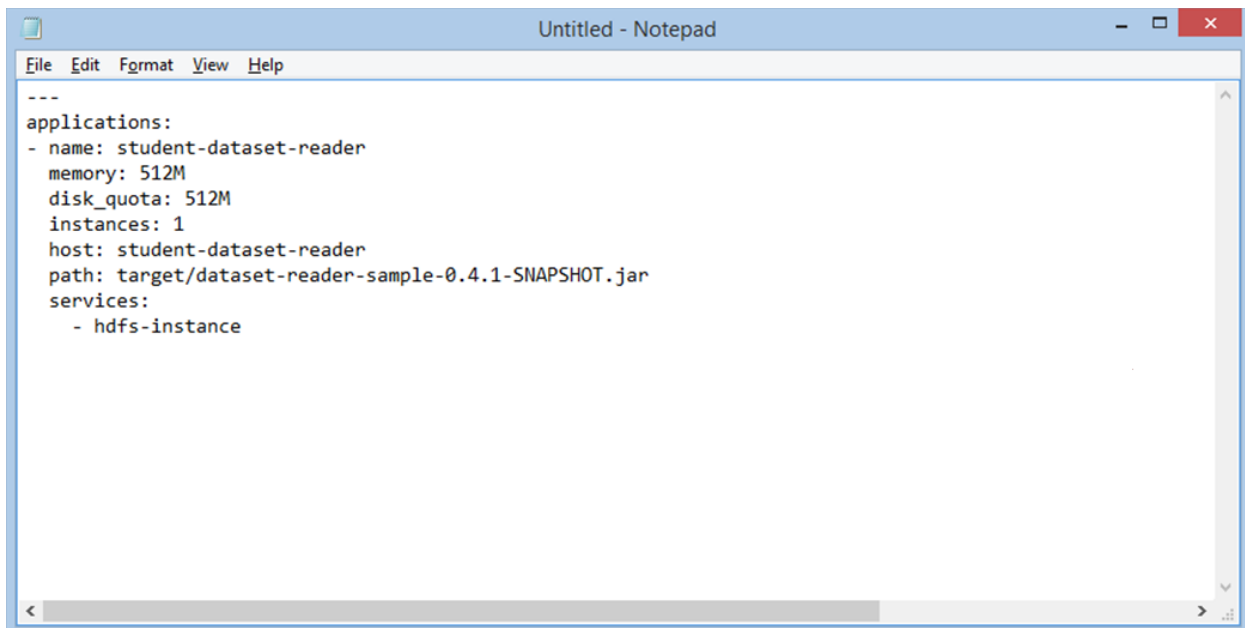
- `cf api <platform API address>`
- `cf login`
- `cf target -o <organization name> -s <space name>`
 - Select an org (enter **1** for the IDF org)
 - **Note:** Your responses are based on the instance you use.
 - 1. IDF
 - 2. workshop
 - Org> **1**
 - Targeted org IDF
 - Select a space (enter **2** for the dev space):
 - **Note:** Your responses are based on the instance you use.
 - 1. IDF-demo
 - 2. dev
 - Space> **2**
 - Targeted space dev
 - API endpoint: <http://api.gotapaas.com> (API version: 2.28.0)
 - User: **Your_ID_Here**
 - Org: IDF
 - Space: dev

5.3. Create an HDFS Instance

- On the TAP console, go to marketplace and create a new instance of HDFS, name it hdfs-instance

5.4. Edit the manifest.yml file:

- It is **critical** that you edit the manifest.yml file:
 - Change the application and host names, e.g., host name = dataset-reader; platform URL = example.com; and application domain = dataset-reader.example.com.
 - name: **<your application name>**
 - host: **<application host name>**
 - **Note:** A copy of this file is located at:
<https://github.com/trustedanalytics/dataset-reader-sample/tree/master/src/cloudfoundry>
- Add hdfs-instance to the manifest (see screenshot below):
 - We are binding the hdfs-instance service to our app. Otherwise, the HdfsConfig object from the InputStreamProvider.java will not be created.
- To edit the manifest file, enter: user@host:~/demo/dataset-reader-sample\$ **vim manifest.yml**



```
---
applications:
- name: student-dataset-reader
  memory: 512M
  disk_quota: 512M
  instances: 1
  host: student-dataset-reader
  path: target/dataset-reader-sample-0.4.1-SNAPSHOT.jar
  services:
    - hdfs-instance
```

6. Deploy the app in TAP

6.1. Locate the URL of the file created by ATK:

- If necessary, sign into TAP.
- Click Applications under Main navigation to view the status of the running instance in the Applications window.
- Copy the URL of the instance that was created in Module 1.

The screenshot displays the TAP interface. On the left, the 'Main navigation' menu includes 'Dashboard', 'Data catalog', 'Applications' (highlighted with a red arrow), 'Marketplace', and 'Tools'. The main area is titled 'Applications' and contains a table with the following data:

Name	Status	Instances	URLs
platform-context	✓	1	platform-context.gotapaas.com
roadrunner	✓	1	atk-3f07b63a-99ae-422f-8010.gotapaas.com
router-metrics-provider	✓	1	router-metrics-provider.gotapaas.com
service-catalog	✓	1	service-catalog.gotapaas.com
smtp-broker	✓	1	smtp-broker.gotapaas.com
user-management	✓	1	user-management.gotapaas.com
Web-Workshop	✓	0	atk-d572dd00-be78-438a-960d.gotapaas.com
zookeeper-wssb-broker	✓	1	zookeeper-wssb-broker.gotapaas.com

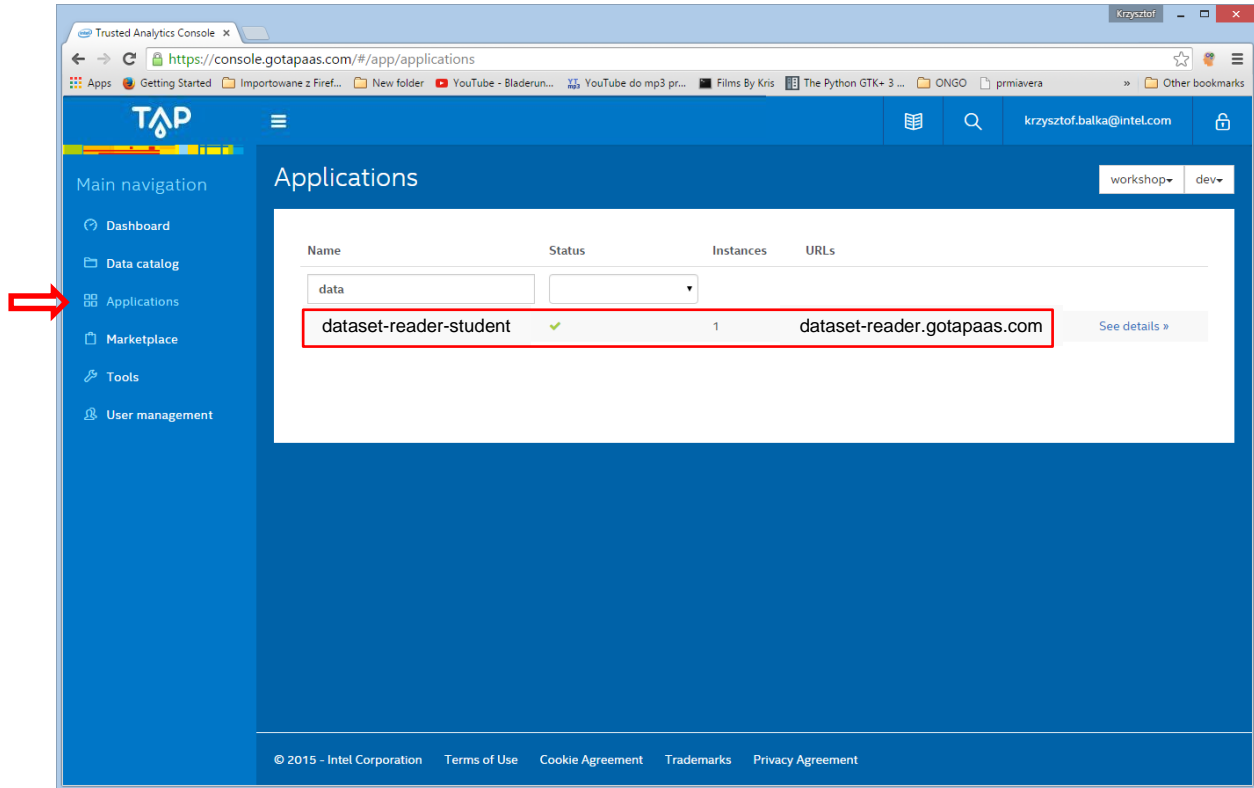
A red arrow points to the 'Web-Workshop' application, and a red box highlights its URL. At the bottom of the table, there is a pagination control showing '« 1 2 3 »'.

6.2. Deploy the app to the platform by executing the commands:

- user@host:~/demo/dataset-reader-sample\$ **cf push**
 - Using manifest file /home/local/user/demo/dataset-reader-sample/manifest.yml
 - Creating app student-dataset-reader in org IDF / space dev as ...
 - OK
 - Creating route student-dataset-reader.gotapaas.com...
 - OK
 - Binding student-dataset-reader.gotapaas.com to student-dataset-reader...
 - OK
 - Uploading student-dataset-reader...
 - Uploading app files from: /home/local/user/demo/dataset-reader-sample/target/dataset-reader-sample-0.4.1-SNAPSHOT.jar
 - Uploading 1.4M, 130 files
 - ✓ Done uploading
 - OK
 - ✓ Binding service hdfs-instance to app student-dataset-reader in org IDF / space dev as ...
 - ✓
 - ✓ OK
 - Starting app student-dataset-reader in org IDF / space dev as USER ...
 - -----> Downloaded app package (40M)
 - -----> Java Buildpack Version: v3.0 ...
 - -----> Downloading Open Jdk JRE 1.8.0_40 ...
 - ✓ 1 of 1 instances running
 - ✓ App started
 - ✓ OK
 - App student-dataset-reader was started using this command ...
 - OK
 - ✓ **requested state: started** (*Important: This indicates the app started.*)
 - ✓ **instances: 1/1**
 - usage: 512M x 1 instances
 - urls: student-dataset-reader.gotapaas.com
 - last uploaded: Thu Sep 10 14:35:06 UTC 2015
 - stack: cflinuxfs2
 - state since cpu memory disk details
 - #0 running 2015-09-10 04:35:51 PM 0.0% 376.7M of 512M 162.5M of 512M

6.3. Confirm the app is deployed on TAP:

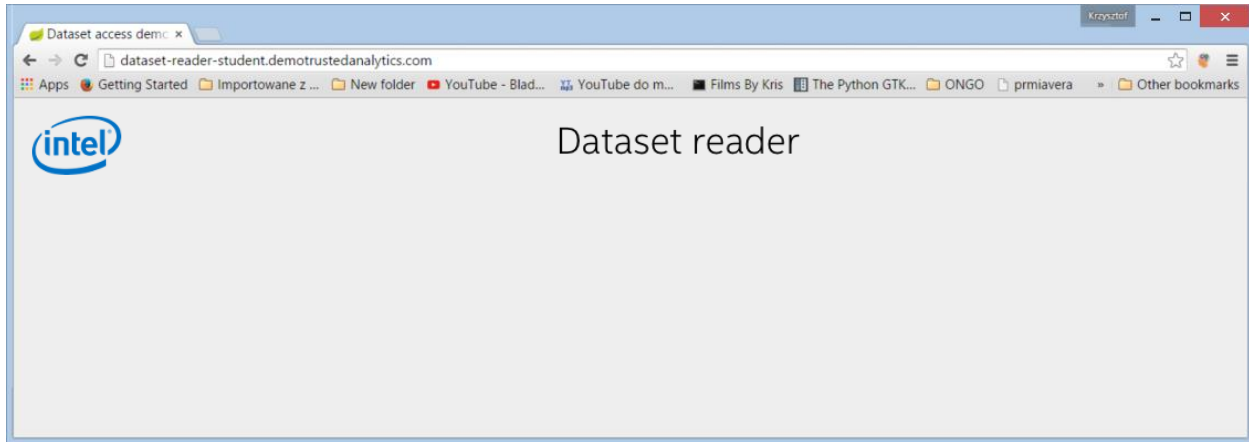
- If necessary, sign into TAP.
- Click Applications under Main navigation to confirm your app was deployed in Cloud Foundry.



7. Try to view the output of Dataset-Reader in a browser

7.1. Why are you not able to view the output?

- We see nothing on the screen, because we haven't provided the link to dataset.



8. Update the environment and set the dataset link

8.1. Update the application dataset:

- Follow the instructions in Module 1 to upload the dataset provided at <https://github.com/trustedanalytics/dataset-reader-sample/blob/master/data/nf-data-application.csv>
- Browse to your dataset from the TAP console and copy the path to the dataset in HDFS found in the targetUri field. The path is indicated in **GREEN** below and will be different for your dataset.

8.2. Set the environment variable by executing the command:

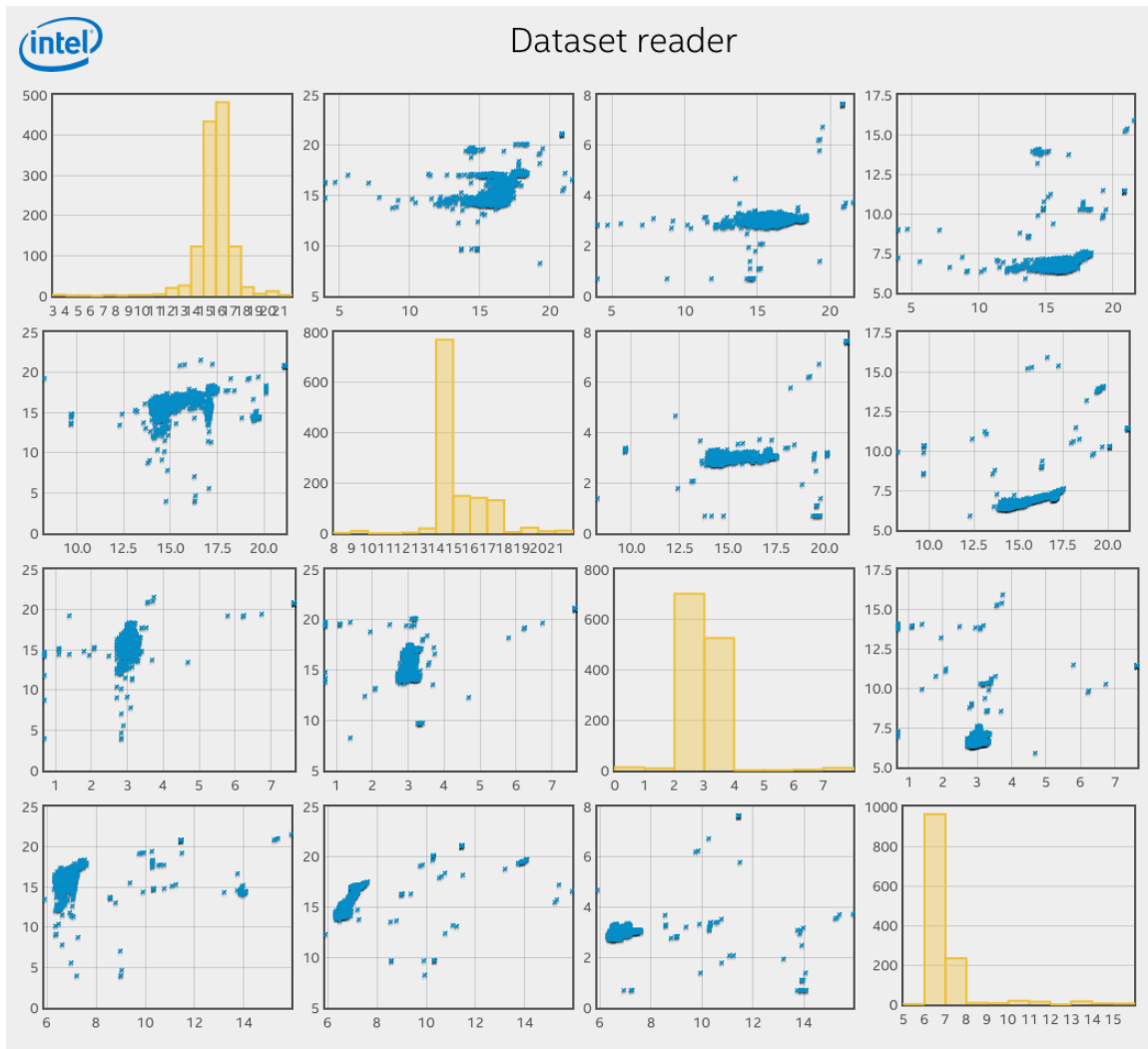
- student@host:~/demo/dataset-reader-sample\$ **cf set-env dataset-reader-student file** **hdfs://nameservice1/org/intel/hdfsbroker/userspace/2451c314-f042-435e-a9be-328029f4337c/1610dc47-cafd-418d-97ca-fd2b8674666c/000000_1**

8.3. Use *cf restage* to ensure your *env* variable changes take effect:

- student@host:~/demo/dataset-reader-sample\$ **cf restage dataset-reader-student**
 - Restaging app dataset-reader-student in org seedorg / space seedspace as admin...
 - ----> Downloaded app package (40M)
 - 1.8.0_60.tar.gz (found in cache)
 - Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.2s)
 - ----> Downloading Spring Auto Reconfiguration 1.10.0_RELEASE ...
 - ----> Uploading droplet (84M)
 - 0 of 1 instances running, 1 starting
 - 0 of 1 instances running, 1 starting
 - ✓ 1 of 1 instances running
 - ✓ App started
 - OK
 - Showing health and status for app dataset-reader-student in org ...
 - OK
 - ✓ requested state: started (**Important:** This indicates the app started.)
 - ✓ instances: 1/1
 - ✓ usage: 512M x 1 instances
 - ✓ urls: dataset-reader-student.demotrustedanalytics.com
 - last uploaded: Thu Sep 17 14:12:23 UTC 2015
 - stack: cflinuxfs2
 - state since cpu memory disk details
 - #0 running 2015-09-17 04:42 PM 0.0% 357.1M of 512M 162.5M of 512M

9. View the output of Dataset-Reader

9.1. Upon successful restage of the Dataset-Reader app, you will see the data presented in the following manner:



CONGRATULATIONS!
You have successfully completed the workshop.