talend

Talend Big Data Sandbox

Big Data Insights Cookbook

talend

Talend Big Data Sandbox

Big Data Insights Cookbook

Overview

Pre-requisites

Setup & Configuration

Hadoop Distribution Download

Demo *(Scenario)*

About this cookbook

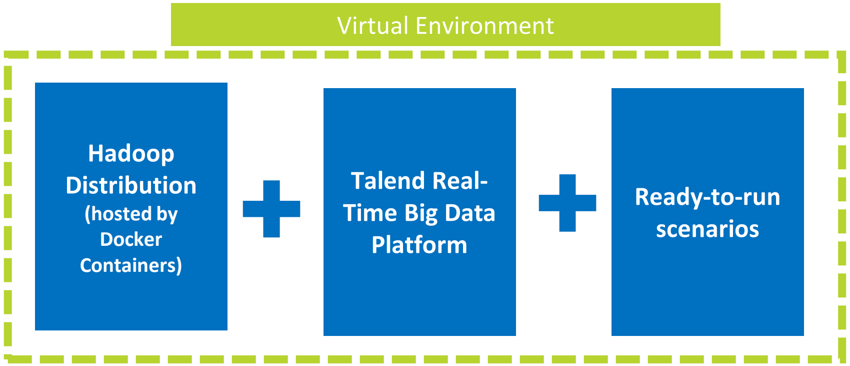
What is the Talend Cookbook?

Using the Talend Real-Time Big Data Platform, this Cookbook provides step-by-step instructions to build and run an end-to-end integration scenario.

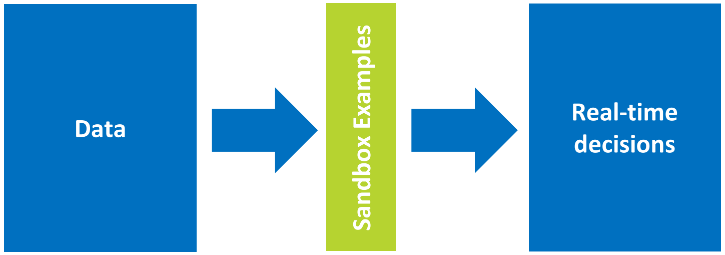
The demos are built on real world use-casees and demonstrate how Talend, Spark, NoSQL and real-time messaging can be easily integrated into your daily business.

Whether batch, streaming or real-time integration, you will begin to understand how Talend can be used to address your big data challenges and move your business into the Data-Driven Age.

What is the Big Data Sandbox?


Virtual Environment
Hadoop Distribution (hosted by Docker Containers)
+
Talend Real-Time Big Data Platform
+
Ready-to-run scenarios

The Talend Real-Time Big Data Sandbox is a virtual environment that combines the Talend Real-Time Big Data Platform with some sample scenarios pre-built and ready-to-run.



See how Talend can turn data into real-time decisions through sandbox examples that integrate Apache Kafka, Spark, Spark Streaming, Hadoop and NoSQL.

What Pre-requisites are required to run Sandbox?

Talend Platform for Big Data includes a graphical IDE (Talend Studio), teamwork management, data quality, and advanced big data features.

Internet connection required for the entire setup process

To see a full list of features please visit Talend's Website:
http://www.talend.com/products/real-time-big-data

You will need a Virtual Machine player such as VMWare or Virtualbox, which can be downloaded here:

- VMware Player Site
- Virtualbox Site

Follow the VM Player install instructions from the provider

The recommended host machine should have:

Memory
8-10GB

Disk Space
20GB

(5GB is for the image download)

Download the Sandbox Virtual Machine file
https://info.talend.com/prodevaltpbdrealtimesandbox.html

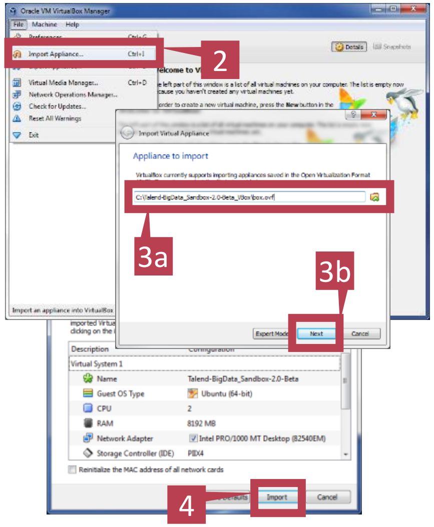
How do I set-up & configure Sandbox?

Follow the steps below to install and configure your Big Data Sandbox:

- Save the downloaded Virtual Machine file to a location on your local PC that is easy to access (e.g. C:/TalendSandbox)
- Follow the instructions below based on the Virtual Machine Player and matching Sandbox file that you are using

Virtualbox

1. Open Virtualbox.
2. From the menu bar, select **File > Import Appliance…**
3. Navigate to the **.ova** file that you downloaded. Select it and click **Next**.
4. Accept the default Appliance Settings by clicking **Import.**



The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the

VirtualBox currently supports importing appliances saved in the Open Virtualization Format

C:\Talend-BigData_Sandbox-2.0-Beta_VBox\box.ovf

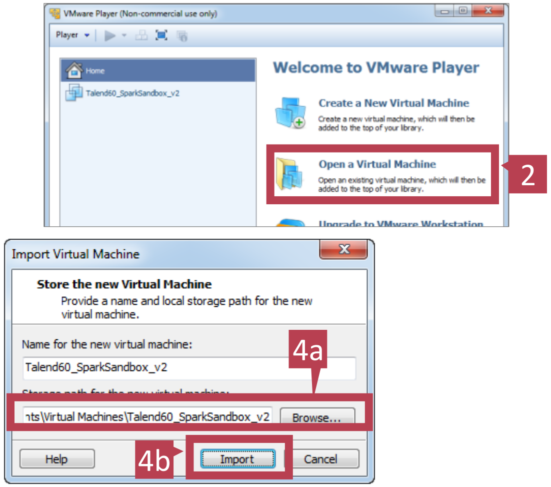
Import an appliance into VirtualBox

Description	Configuration
Virtual System 1	
Name	Talend-BigData_Sandbox-2.0-Beta
Guest OS Type	Ubuntu (64-bit)
CPU	2
RAM	8192 MB
Network Adapter	☑ Intel PRO/1000 MT Desktop (82540EM)
Storage Controller (IDE)	PIIX4

□ Reinitialize the MAC address of all network cards

VMware Player

1. Open VMware Player.
2. Click on "**Open a Virtual Machine**"
3. Navigate to the **.ova** file that you downloaded. Select it and click **Open**.
4. Select the Storage path for the new Virtual Machine (e.g. C:/TalendSandbox/vmware) and then click **Import**.



Create a new virtual machine, which will then be added to the top of your library.

Open an existing virtual machine, which will then be added to the top of your library.

Store the new Virtual Machine

Provide a name and local storage path for the new virtual machine.

Name for the new virtual machine: Talend60_SparkSandbox_v2

Storage path for the new virtual machine: nts\Virtual Machines\Talend60_SparkSandbox_v2

Note: The Talend Big Data Sandbox Virtual Machines come pre-configured to run with 8GB RAM and 2 CPU's. You may need to adjust these settings based on your PC's capabilities. While not pre-configured, it is also recommended to enable a Sound Card/Devise before starting the VM to take advantage of Tutorial Videos within the Virtual Environment.

Talend Big Data Sandbox

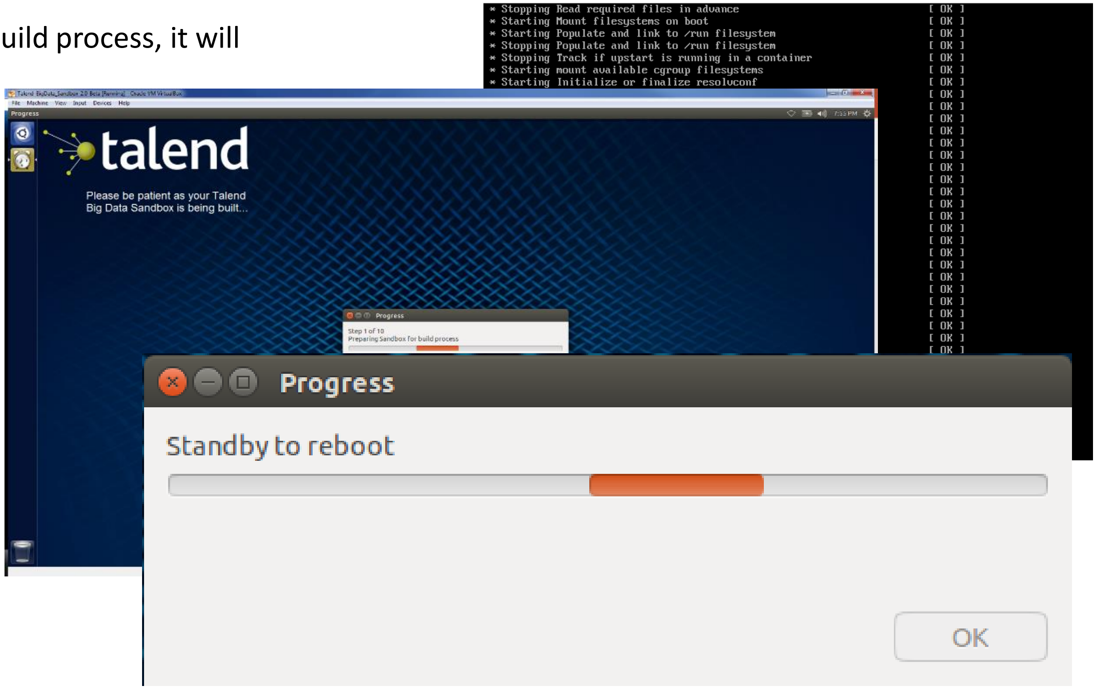
Big Data Insights Cookbook

Starting the VM for the first time…

- When you start the Talend Big Data Sandbox for the first time, the virtual machine will begin a **6-step process to build** the Virtual Environment.

- This process can take **10-20 mins depending on internet connection speeds** and network traffic. Popup messages will be present on screen to keep you informed of the progress.

- Once the Sandbox has completed it's build process, it will **automatically reboot**.



```
* Stopping Read required files in advance                    [ OK ]
* Starting Mount filesystems on boot                         [ OK ]
* Starting Populate and link to /run filesystem              [ OK ]
* Stopping Populate and link to /run filesystem              [ OK ]
* Stopping Track if upstart is running in a container        [ OK ]
* Starting mount available cgroup filesystems                [ OK ]
* Starting Initialize or finalize resolvconf                 [ OK ]
```

Talend BigData_Sandbox 2.0 Beta [Running] - Oracle VM VirtualBox

Please be patient as your Talend Big Data Sandbox is being built…

Progress

Step 1 of 10
Preparing Sandbox for build process

Progress

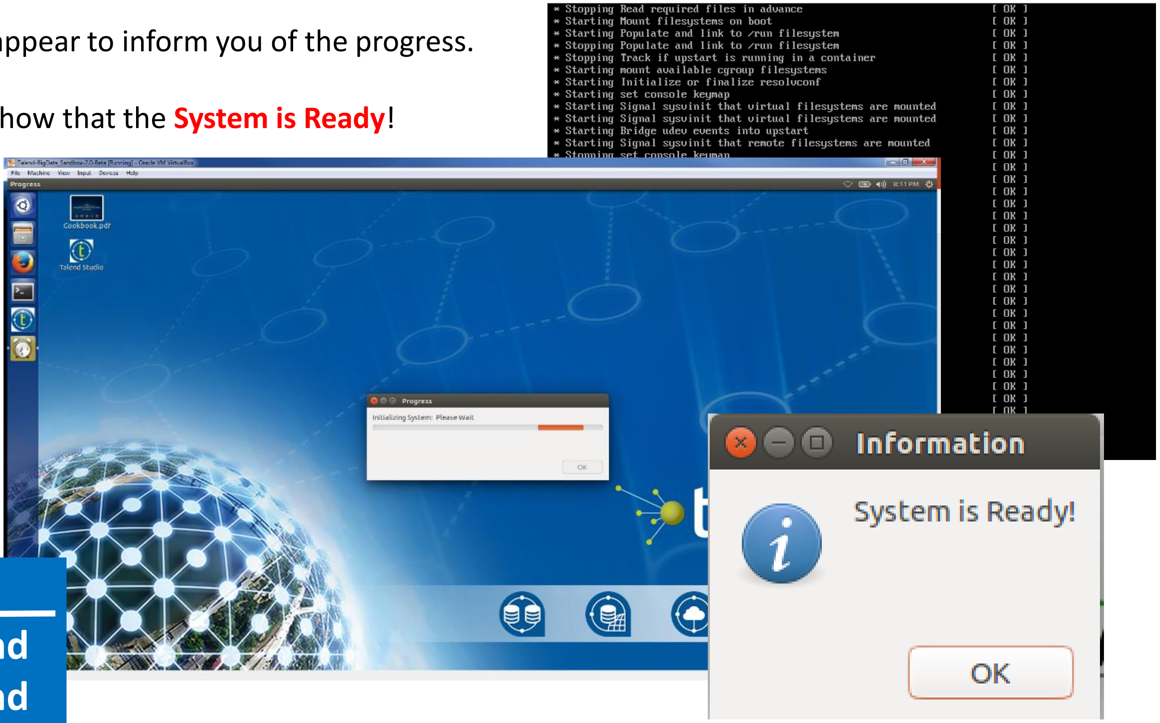
Standby to reboot

Login Info

User:	**talend**
Password:	**talend**
Sudo Password:	**talend**

Starting the VM for the first time (cont.)

- Once the Virtual Machine reboots, the **Docker Components** that were installed during the build process will need to **initialize.**
- Additional **Popup messages** will appear to inform you of the progress.
- When complete, a message will show that the **System is Ready**!



```
* Stopping Read required files in advance                        [ OK ]
* Starting Mount filesystems on boot                             [ OK ]
* Starting Populate and link to /run filesystem                  [ OK ]
* Stopping Populate and link to /run filesystem                  [ OK ]
* Stopping Track if upstart is running in a container            [ OK ]
* Starting mount available cgroup filesystems                    [ OK ]
* Starting Initialize or finalize resolvconf                     [ OK ]
* Starting set console keymap                                    [ OK ]
* Starting Signal sysvinit that virtual filesystems are mounted  [ OK ]
* Starting Signal sysvinit that virtual filesystems are mounted  [ OK ]
* Starting Bridge udev events into upstart                       [ OK ]
* Starting Signal sysvinit that remote filesystems are mounted   [ OK ]
* Stopping set console keymap                                    [ OK ]
```

Talend-BigData_Sandbox-2.0-Beta [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Progress
8:11 PM
Cookbook.pdf
Talend Studio
Progress
OK
Information
OK

Initializing System: Please Wait

System is Ready!

Login Info

User: talend

Password: talend

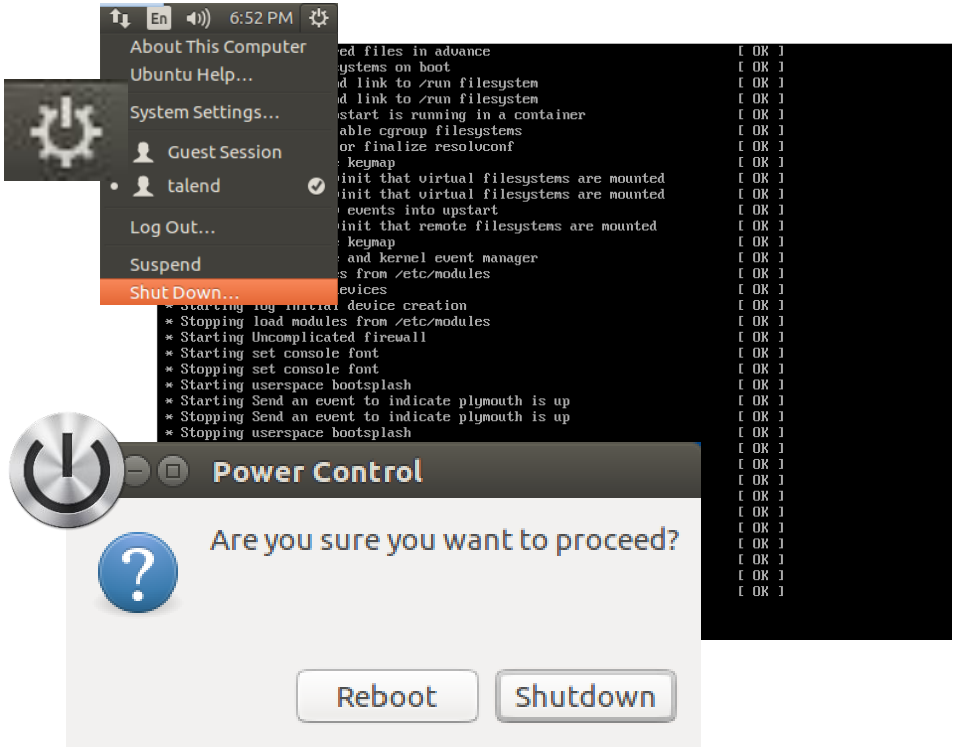
Sudo Password: talend

Shutting Down the VM

- There are 2 methods to shut down the Virtual Machine - **Standard** and **Advanced**

- **Standard Shutdown**
  - This is the standard shutdown method for Ubuntu
  - It is available from the system menu at the Top-Right of the menu bar

- **Advanced Shutdown**
  - Found on the Desktop, Double-click the Power Icon to start a custom shutdown script that cleanly exits all running Docker Containers before shutting down the VM. It will take a few minutes to execute but will make Startup quicker.
  - You can choose to either <u>Shutdown</u> or <u>Reboot</u> your VM via the Advanced Shutdown method.

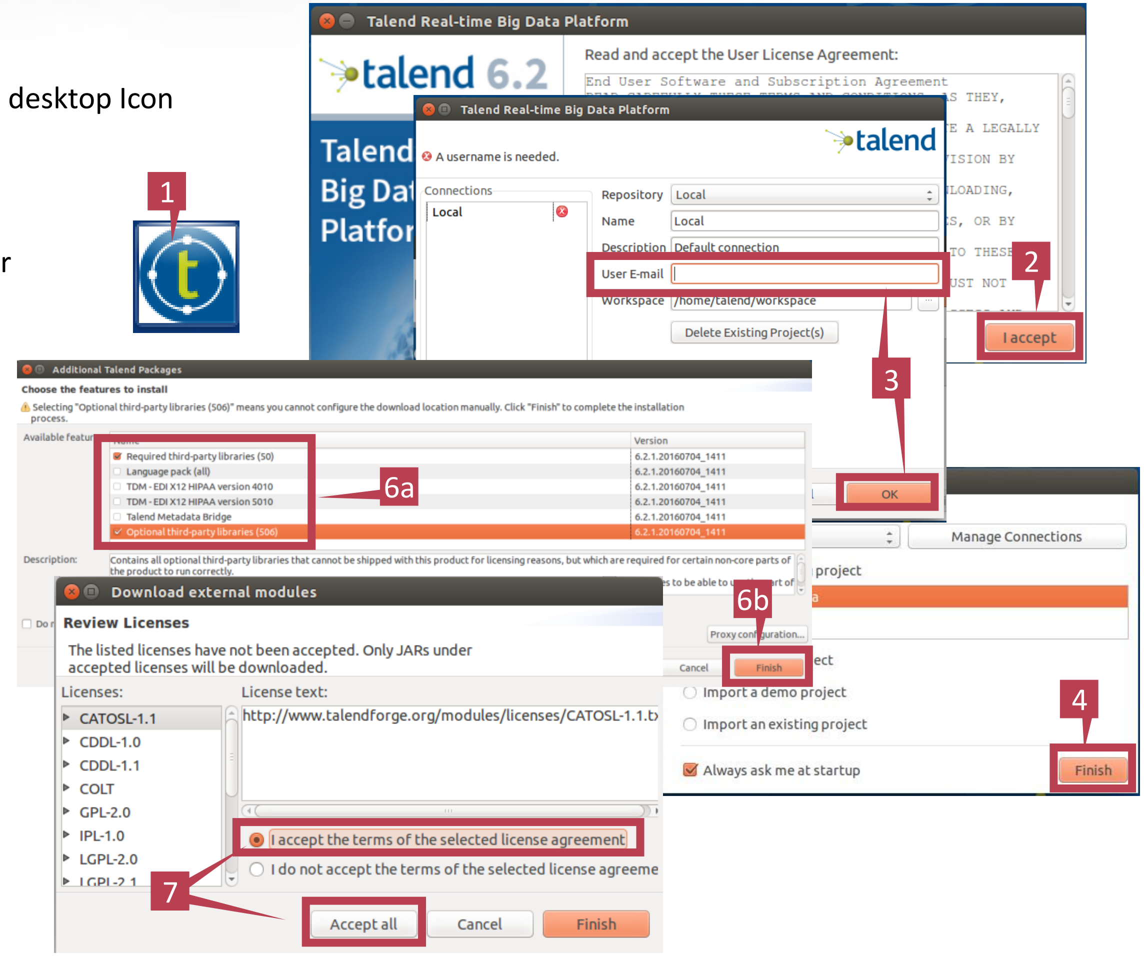


```
ed files in advance                                  [ OK ]
ystems on boot                                       [ OK ]
d link to /run filesystem                            [ OK ]
d link to /run filesystem                            [ OK ]
start is running in a container                      [ OK ]
able cgroup filesystems                              [ OK ]
or finalize resolvconf                               [ OK ]
 keymap                                              [ OK ]
init that virtual filesystems are mounted            [ OK ]
init that virtual filesystems are mounted            [ OK ]
 events into upstart                                 [ OK ]
init that remote filesystems are mounted             [ OK ]
 keymap                                              [ OK ]
 and kernel event manager                            [ OK ]
s from /etc/modules                                  [ OK ]
evices                                               [ OK ]
* Starting log initial device creation               [ OK ]
* Stopping load modules from /etc/modules            [ OK ]
* Starting Uncomplicated firewall                    [ OK ]
* Starting set console font                          [ OK ]
* Stopping set console font                          [ OK ]
* Starting userspace bootsplash                      [ OK ]
* Starting Send an event to indicate plymouth is up  [ OK ]
* Stopping Send an event to indicate plymouth is up  [ OK ]
* Stopping userspace bootsplash                      [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
                                                     [ OK ]
```

Are you sure you want to proceed?

Starting Talend Studio for the first time…

1. Start Talend Studio by double-clicking on the desktop Icon or single clicking on the Unity Bar Icon.
2. Click **I Accept** the End User License Agreement.
3. Click on **Manage Connections** and enter your email address, then Click **OK.**
4. Select the **Base_Project** – java project and click **Finish**.
5. Once Studio Loads, Close the Welcome Screen.
6. Install **Additional Talend Packages**. Select *Required third-party libraries* and click **Finish**.
7. A popup will display all 3rd party licenses that need acceptance. Click the "*I accept the terms of the selected license agreement*" radio button and click **Accept all.**
8. **Let the downloads complete before continuing.**



Talend Real-time Big Data Platform

talend 6.2

Talend Big Data Platform

Read and accept the User License Agreement:

```
End User Software and Subscription Agreement
...AS THEY,
...E A LEGALLY
...VISION BY
...LOADING,
...S, OR BY
...TO THESE
...UST NOT
```

I accept

Talend Real-time Big Data Platform

A username is needed.

Connections: Local

Repository: Local

Name: Local

Description: Default connection

User E-mail:

Workspace: /home/talend/workspace

Delete Existing Project(s)

OK

Manage Connections

project

Import a demo project

Import an existing project

☑ Always ask me at startup

Finish

Additional Talend Packages

Choose the features to install

⚠ Selecting "Optional third-party libraries (506)" means you cannot configure the download location manually. Click "Finish" to complete the installation process.

Available features:

Name	Version
☑ Required third-party libraries (50)	6.2.1.20160704_1411
□ Language pack (all)	6.2.1.20160704_1411
□ TDM - EDI X12 HIPAA version 4010	6.2.1.20160704_1411
□ TDM - EDI X12 HIPAA version 5010	6.2.1.20160704_1411
□ Talend Metadata Bridge	6.2.1.20160704_1411
✓ Optional third-party libraries (506)	6.2.1.20160704_1411

Description: Contains all optional third-party libraries that cannot be shipped with this product for licensing reasons, but which are required for certain non-core parts of the product to run correctly.

Proxy configuration...

Cancel | Finish

Download external modules

Review Licenses

The listed licenses have not been accepted. Only JARs under accepted licenses will be downloaded.

Licenses: CATOSL-1.1, CDDL-1.0, CDDL-1.1, COLT, GPL-2.0, IPL-1.0, LGPL-2.0, LGPL-2.1

License text: http://www.talendforge.org/modules/licenses/CATOSL-1.1.tx

● I accept the terms of the selected license agreement

○ I do not accept the terms of the selected license agreeme

Accept all | Cancel | Finish

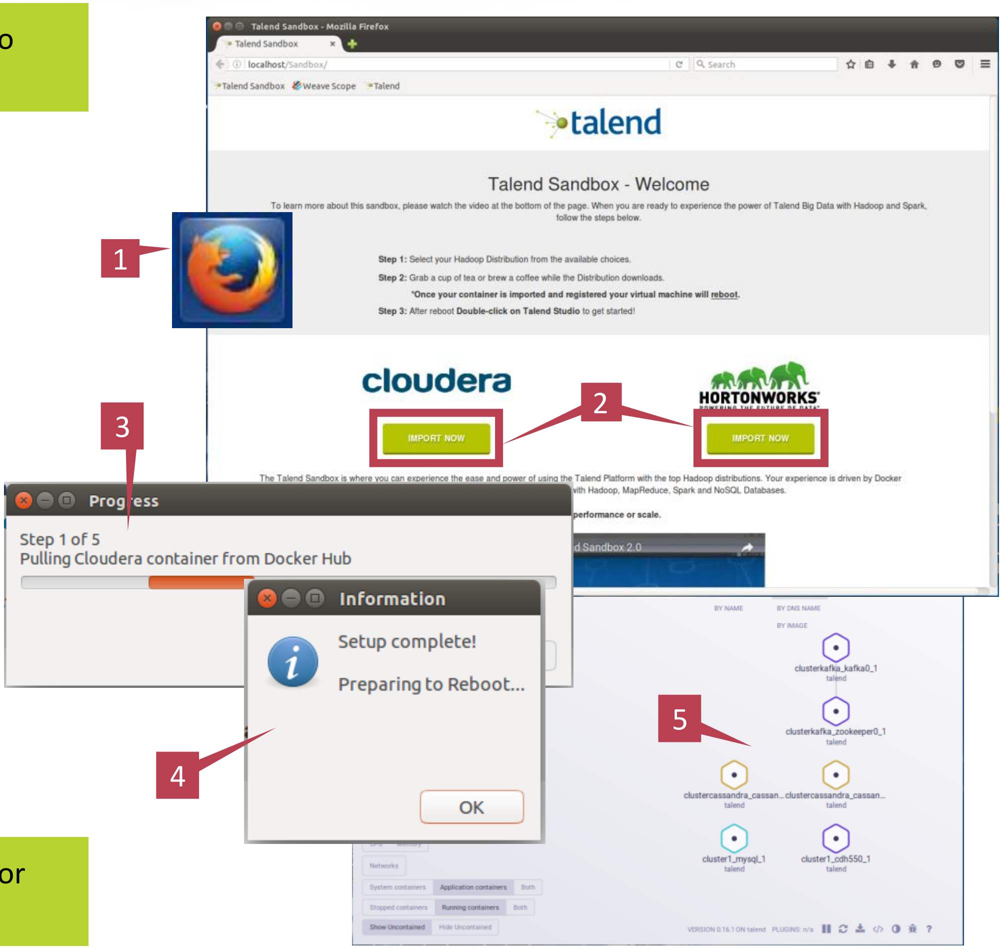
Choosing a Distribution…

Note: It is not necessary to download a Distribution to evaluate the Sandbox. Click Here to begin now!

Follow the steps below to install a Hadoop Distribution in the Talend Big Data Sandbox:

1. Start **Firefox**
2. **Choose the Distribution** you would like to evaluate with the Talend Platform.
3. **Be patient** as the Virtual Machine downloads and installs the selected Distributions. Each distribution is approx. 2.5GB *(Notifications will indicate progress.)*
4. The **Virtual Machine will reboot** when the installation is complete.
5. Upon reboot, the system will initialize the installed Docker containers. Once complete, you will see a popup stating "System is Ready!!"

Note: Be sure to watch the available Tutorial Videos for more information on the Sandbox



Talend Sandbox - Mozilla Firefox

localhost/Sandbox/

Talend Sandbox - Welcome

To learn more about this sandbox, please watch the video at the bottom of the page. When you are ready to experience the power of Talend Big Data with Hadoop and Spark, follow the steps below.

Step 1: Select your Hadoop Distribution from the available choices.

Step 2: Grab a cup of tea or brew a coffee while the Distribution downloads.

***Once your container is imported and registered your virtual machine will reboot.**

Step 3: After reboot **Double-click on Talend Studio** to get started!

The Talend Sandbox is where you can experience the ease and power of using the Talend Platform with the top Hadoop distributions. Your experience is driven by Docker … with Hadoop, MapReduce, Spark and NoSQL Databases.

… **performance or scale.**

Progress

Step 1 of 5
Pulling Cloudera container from Docker Hub

Information

Setup complete!

Preparing to Reboot...

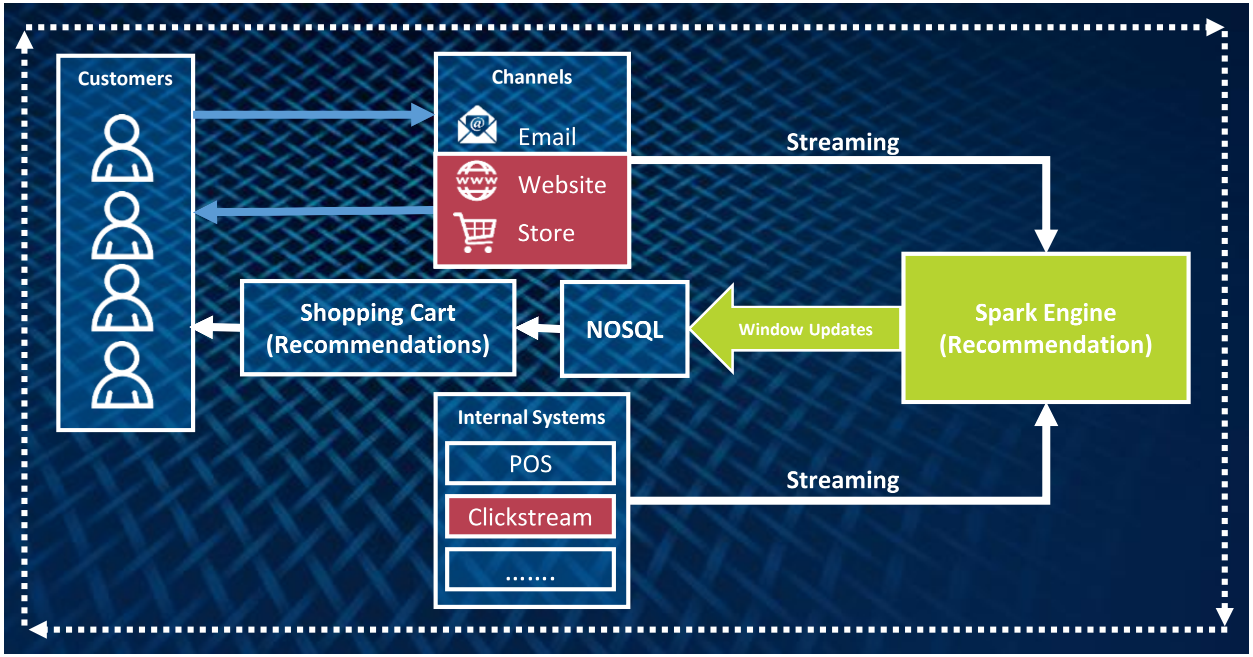
Note: This demo is available in Local Mode and Distribution Mode. In Local Mode it utilizes Talend's Local Spark Engine and Local File System. In Distribution Mode, it utilizes the selected Distro's Yarn Resource Manager and HDFS File System.

Overview:

In this Demo you will see a simple version of making your website an Intelligent Application.

You will experience:

- Building a Spark Recommendation Model
- Setting up a new Kafka topic to help simulate live web traffic coming from Live web users browsing a retail web store.
- Most important you will see first-hand with Talend how you can take streaming data and turn it into real-time recommendations to help improve shopping cart sales.



The following Demo will help you see the value that using Talend can bring to your big data projects:
The Retail Recommendation Demo is designed to illustrate the simplicity and flexibility Talend brings to using Spark in your Big Data Architecture.

This Demo highlights:

Kafka

Create a Kafka Topic to Produce and Consume real-time streaming data

Machine Learning

Create a Spark recommendation model based on specific user actions

Spark Streaming

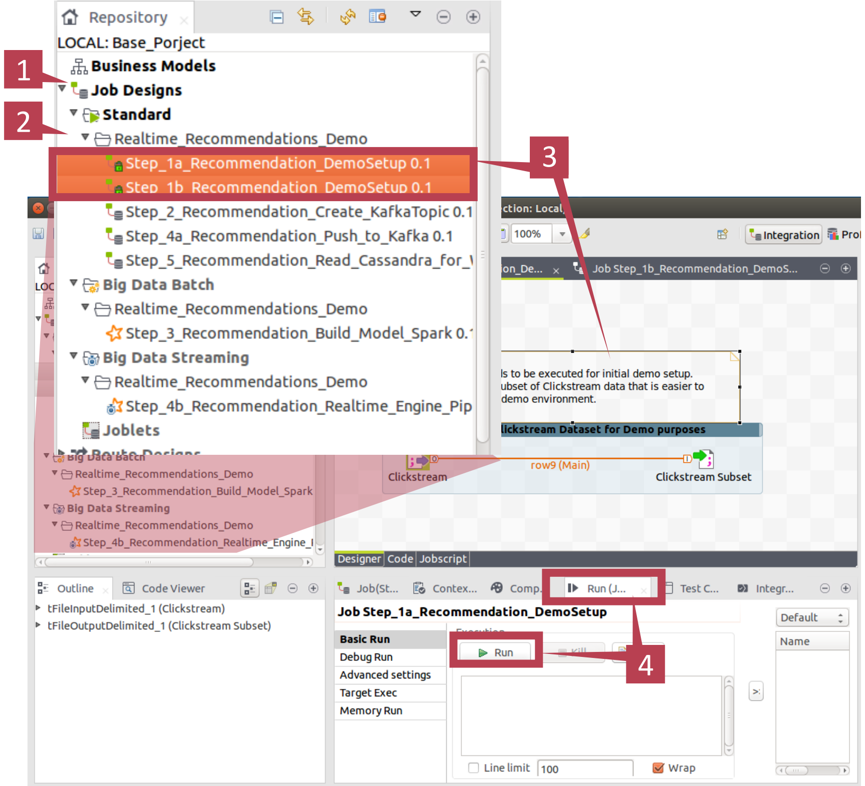
Stream live recommendations to a Cassandra NoSQL database for "Fast Data" access for a WebUI

If you are familiar with the ALS model, you can update the ALS parameters to enhance the model or just leave the default values.

Demo Setup:

Before you can execute the Retail Recommendation Demo, you will need to generate the source data and pre-populate the Cassandra Lookup Tables.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard Jobs > Realtime_Recommendation_Demo**
3. Double click on **Step_1a_Recommendation_DemoSetup 0.1** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.
5. When the job is finished, repeat steps 1-4 for **Step_1b_Recommendation_DemoSetup 0.1**

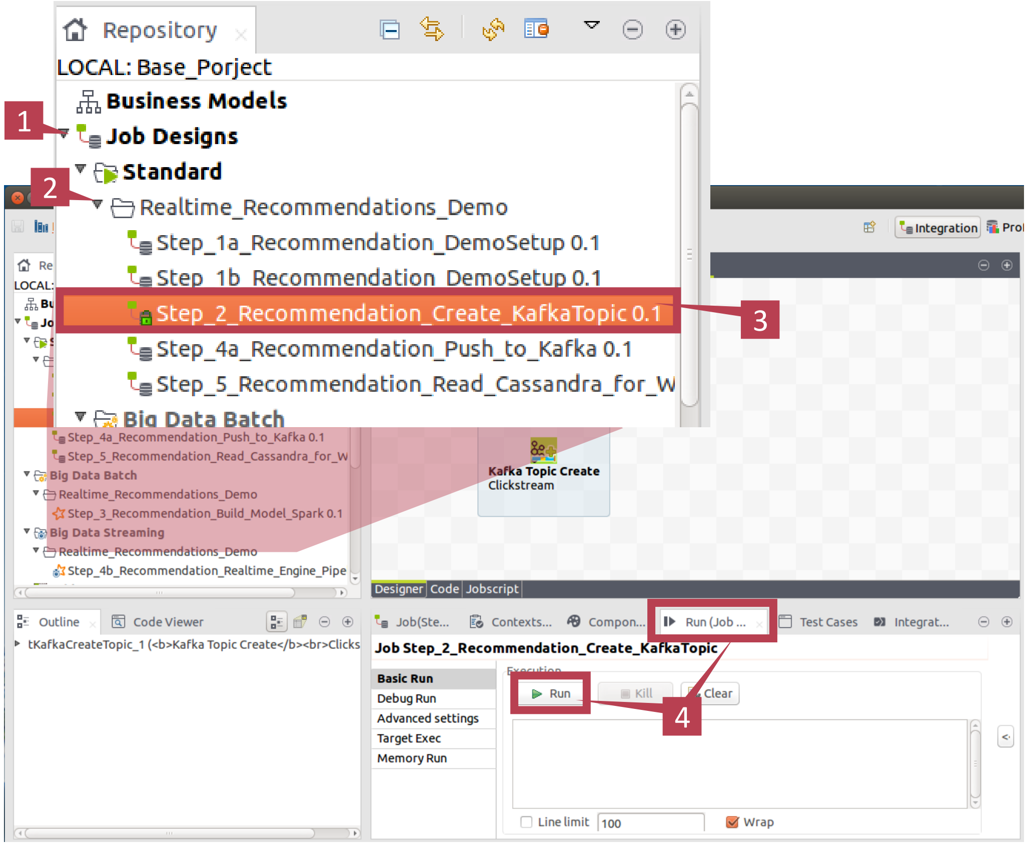

Repository
LOCAL: Base_Porject
Business Models
Job Designs
Standard
Realtime_Recommendations_Demo
Step_1a_Recommendation_DemoSetup 0.1
Step_1b_Recommendation_DemoSetup 0.1
Step_2_Recommendation_Create_KafkaTopic 0.1
Step_4a_Recommendation_Push_to_Kafka 0.1
Step_5_Recommendation_Read_Cassandra_for_\
Big Data Batch
Realtime_Recommendations_Demo
Step_3_Recommendation_Build_Model_Spark 0.1
Big Data Streaming
Realtime_Recommendations_Demo
Step_4b_Recommendation_Realtime_Engine_Pip
Joblets
1
2
3
4
Job Step_1b_Recommendation_DemoS...
ls to be executed for initial demo setup.
ubset of Clickstream data that is easier to
demo environment.
lickstream Dataset for Demo purposes
Clickstream
row9 (Main)
Clickstream Subset
Designer
Code
Jobscript
Outline
Code Viewer
tFileInputDelimited_1 (Clickstream)
tFileOutputDelimited_1 (Clickstream Subset)
Job(St...
Contex...
Comp...
Run (J...
Test C...
Integr...
Job Step_1a_Recommendation_DemoSetup
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Kill
Default
Name
Line limit
100
Wrap

Execute the Retail Recommendation Demo:

Create a Kafka Topic:

1. Navigate to the **Job Designs** folder:
2. Click on **Standard Jobs > Realtime_Recommendation_Demo**
3. Double click on **Step_2_Recommendation_Create_KafkaTopic 0.1** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

Now you can generate the recommendation model by loading the product ratings data into the Alternating Least Squares (ALS) Algorithm. Rather than coding a complex algorithm with Scala, a single Spark component available in Talend Studio simplifies the model creation process. **The resultant model can be stored in HDFS or in this case, locally.**


Repository
LOCAL: Base_Porject
Business Models
1
Job Designs
2
Standard
Realtime_Recommendations_Demo
Step_1a_Recommendation_DemoSetup 0.1
Step_1b_Recommendation_DemoSetup 0.1
Step_2_Recommendation_Create_KafkaTopic 0.1
3
Step_4a_Recommendation_Push_to_Kafka 0.1
Step_5_Recommendation_Read_Cassandra_for_W
Big Data Batch
Realtime_Recommendations_Demo
Step_3_Recommendation_Build_Model_Spark 0.1
Big Data Streaming
Realtime_Recommendations_Demo
Step_4b_Recommendation_Realtime_Engine_Pipe
Integration
Kafka Topic Create
Clickstream
Designer
Code
Jobscript
Outline
Code Viewer
tKafkaCreateTopic_1 (<b>Kafka Topic Create</b>
Clicks
Job(Ste...
Contexts...
Compon...
Run (Job ...
Test Cases
Integrat...
Job Step_2_Recommendation_Create_KafkaTopic
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Execution
Run
Kill
Clear
4
Line limit
100
Wrap

talend

Talend Big Data Sandbox

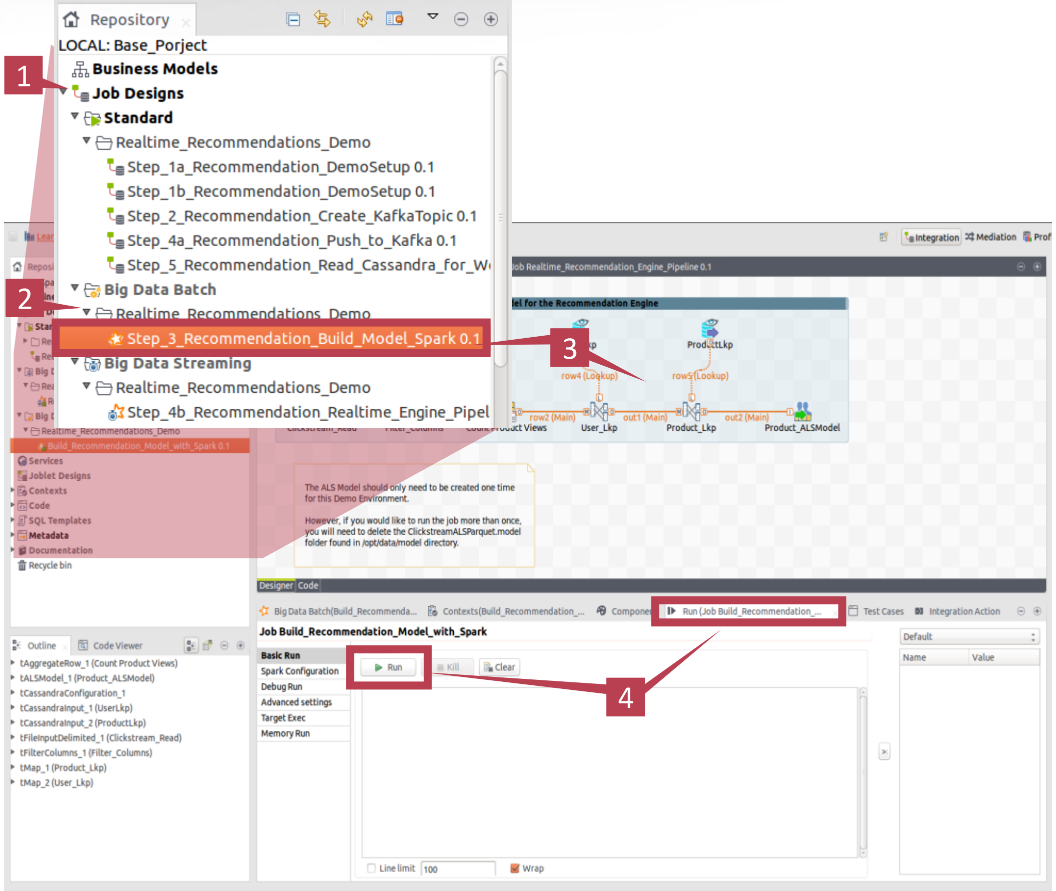
Big Data Insights Cookbook

Execute the Retail Recommendation Demo:

Generate a Recommendation Model using Spark.

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Batch** > **Realtime_Recommendations_Demo**
3. Double click on **Step_3_Recommendation_Build_Model_Spark 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

With the Recommendation model created, your lookup tables populated and your Kafka topic ready to consume data, you can now stream your Clickstream data into your Recommendation model and put the results into your Cassandra tables for reference from a WebUI.


Repository
LOCAL: Base_Porject
1
Business Models
Job Designs
Standard
Realtime_Recommendations_Demo
Step_1a_Recommendation_DemoSetup 0.1
Step_1b_Recommendation_DemoSetup 0.1
Step_2_Recommendation_Create_KafkaTopic 0.1
Step_4a_Recommendation_Push_to_Kafka 0.1
Step_5_Recommendation_Read_Cassandra_for_W
2
Big Data Batch
Realtime_Recommendations_Demo
Step_3_Recommendation_Build_Model_Spark 0.1
3
Big Data Streaming
Realtime_Recommendations_Demo
Step_4b_Recommendation_Realtime_Engine_Pipel
Integration
Mediation
Job Realtime_Recommendation_Engine_Pipeline 0.1
row2 (Main)
row4 (Lookup)
row5 (Lookup)
out1 (Main)
out2 (Main)
User_Lkp
ProductLkp
Product_Lkp
Product_ALSModel
Count Product Views
Services
Joblet Designs
Contexts
Code
SQL Templates
Metadata
Documentation
Recycle bin
Designer
Code
Big Data Batch(Build_Recommenda...
Contexts(Build_Recommendation_...
Run (Job Build_Recommendation_...
Test Cases
Integration Action
Job Build_Recommendation_Model_with_Spark
Basic Run
Spark Configuration
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Kill
Clear
4
Default
Name
Value
Line limit
100
Wrap
Outline
Code Viewer
tAggregateRow_1 (Count Product Views)
tALSModel_1 (Product_ALSModel)
tCassandraConfiguration_1
tCassandraInput_1 (UserLkp)
tCassandraInput_2 (ProductLkp)
tFileInputDelimited_1 (Clickstream_Read)
tFilterColumns_1 (Filter_Columns)
tMap_1 (Product_Lkp)
tMap_2 (User_Lkp)

The ALS Model should only need to be created one time for this Demo Environment.

However, if you would like to run the job more than once, you will need to delete the ClickstreamALSParquet.model folder found in /opt/data/model directory.

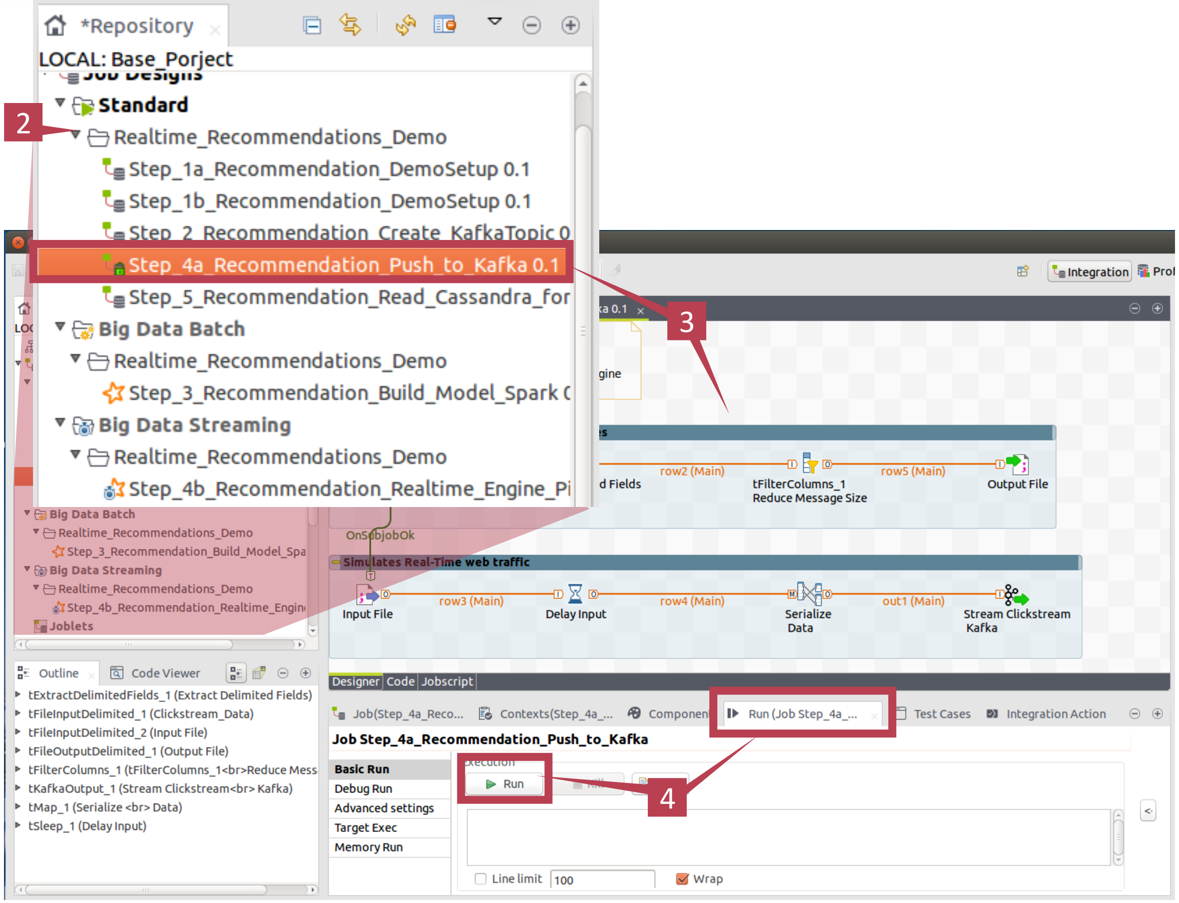
If you are familiar with the ALS model, you can update the ALS parameters to enhance the model or just leave the default values.

Execute the Retail Recommendation Demo:

1. Navigate to the **Job Designs** folder.
2. Click on **Standard Jobs > Realtime_Recommendations_Demo**
3. Double click on **Step_4a_Recommendation_Push_to_Kafka 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

This job is setup to simulate real-time streaming of web traffic and clickstream data into a Kafka topic that will then be consumed by our recommendation engine to produce our recommendations.

After starting the Push to Kafka, continue to the next steps of the demo.


*Repository
LOCAL: Base_Porject
Job Designs
Standard
Realtime_Recommendations_Demo
Step_1a_Recommendation_DemoSetup 0.1
Step_1b_Recommendation_DemoSetup 0.1
Step_2_Recommendation_Create_KafkaTopic 0
Step_4a_Recommendation_Push_to_Kafka 0.1
Step_5_Recommendation_Read_Cassandra_for
Big Data Batch
Realtime_Recommendations_Demo
Step_3_Recommendation_Build_Model_Spark
Big Data Streaming
Realtime_Recommendations_Demo
Step_4b_Recommendation_Realtime_Engine_Pi
Joblets
2
3
4
Integration
row2 (Main)
tFilterColumns_1
Reduce Message Size
row5 (Main)
Output File
OnSubjobOk
Simulates Real-Time web traffic
Input File
row3 (Main)
Delay Input
row4 (Main)
Serialize Data
out1 (Main)
Stream Clickstream Kafka
Designer
Code
Jobscript
Outline
Code Viewer
tExtractDelimitedFields_1 (Extract Delimited Fields)
tFileInputDelimited_1 (Clickstream_Data)
tFileInputDelimited_2 (Input File)
tFileOutputDelimited_1 (Output File)
tFilterColumns_1 (tFilterColumns_1
Reduce Mess
tKafkaOutput_1 (Stream Clickstream
 Kafka)
tMap_1 (Serialize
 Data)
tSleep_1 (Delay Input)
Job(Step_4a_Reco...
Contexts(Step_4a_...
Component
Run (Job Step_4a_...
Test Cases
Integration Action
Job Step_4a_Recommendation_Push_to_Kafka
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Execution
Run
Line limit
100
Wrap

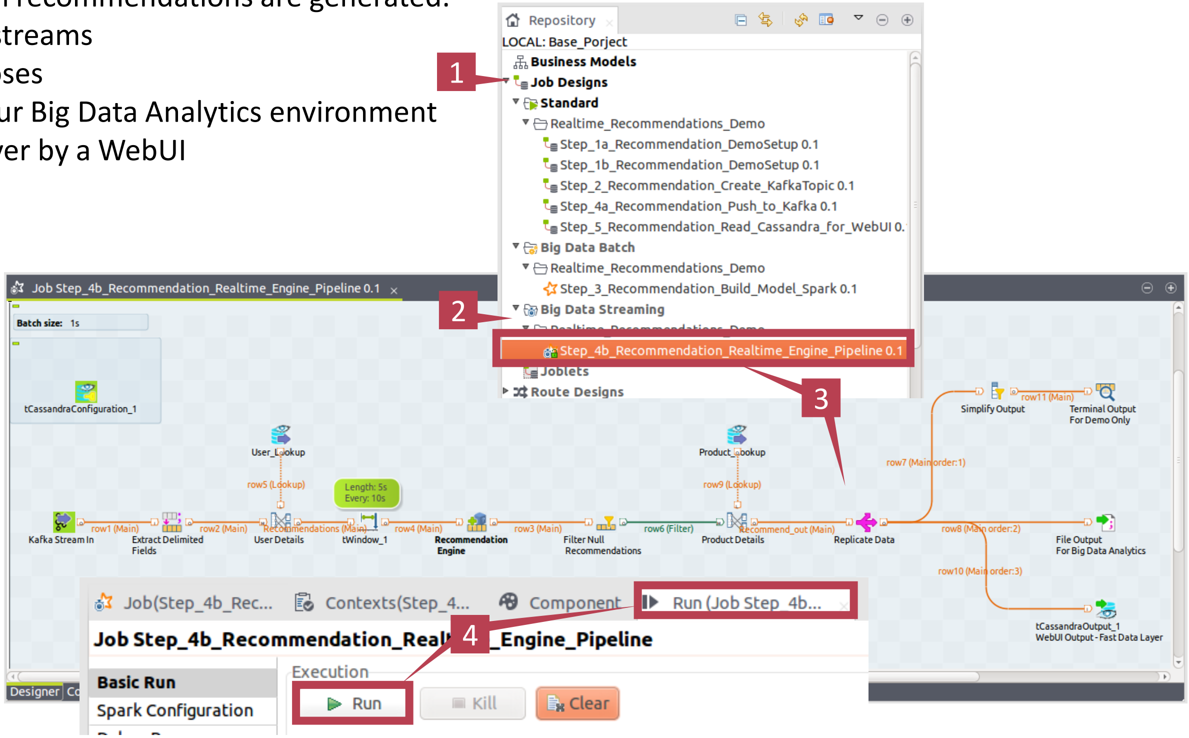
Execute the Retail Recommendation Demo:

In this job:

- A Kafka Consumer reads in Clickstream Data.
- The data is fed into the Recommendation Engine, producing Real-time “offers” based on the current user’s activity.
- The tWindow component controls how often recommendations are generated.
- The recommendations are sent to 3 output streams
  - ✓ **Execution window** for viewing purposes
  - ✓ **File System** for later processing in your Big Data Analytics environment
  - ✓ **Cassandra** for use in a “Fast Data” layer by a WebUI

With data streaming to the Kafka Topic…
Start the recommendation pipeline.

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Streaming > Realtime_Recommendation_Demo**
3. Double click on **Step_4b_Recommendation_Realtime_Engine_Pipeline 0.1.** This opens the job in the designer window.
4. Click on **Run** to Start Recommendation Engine

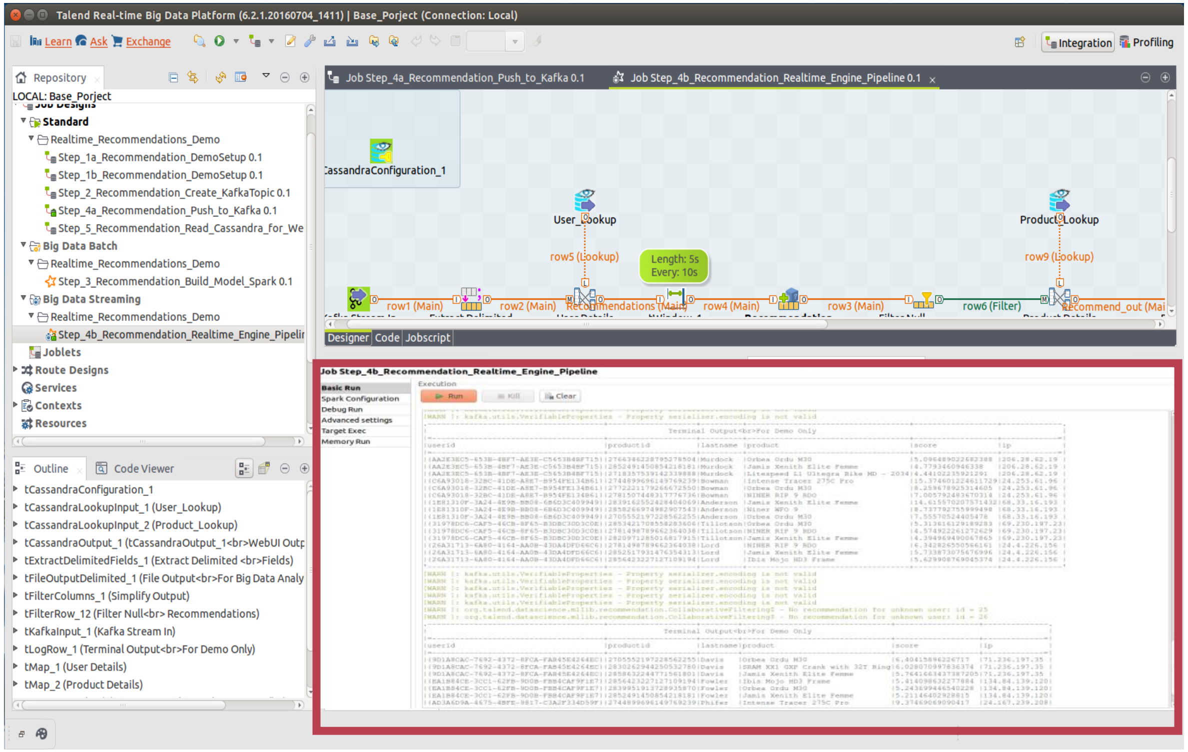

Repository
LOCAL: Base_Porject
Business Models
Job Designs
Standard
Realtime_Recommendations_Demo
Step_1a_Recommendation_DemoSetup 0.1
Step_1b_Recommendation_DemoSetup 0.1
Step_2_Recommendation_Create_KafkaTopic 0.1
Step_4a_Recommendation_Push_to_Kafka 0.1
Step_5_Recommendation_Read_Cassandra_for_WebUI 0.
Big Data Batch
Realtime_Recommendations_Demo
Step_3_Recommendation_Build_Model_Spark 0.1
Big Data Streaming
Step_4b_Recommendation_Realtime_Engine_Pipeline 0.1
Joblets
Route Designs
1
2
3
4
Job Step_4b_Recommendation_Realtime_Engine_Pipeline 0.1
Batch size: 1s
tCassandraConfiguration_1
User_Lookup
Product_Lookup
Length: 5s
Every: 10s
Kafka Stream In
Extract Delimited Fields
User Details
tWindow_1
Recommendation Engine
Filter Null Recommendations
Product Details
Replicate Data
Simplify Output
Terminal Output For Demo Only
File Output For Big Data Analytics
tCassandraOutput_1 WebUI Output - Fast Data Layer
row1 (Main)
row2 (Main)
row5 (Lookup)
Recommendations (Main)
row4 (Main)
row3 (Main)
row6 (Filter)
row9 (Lookup)
Recommend_out (Main)
row7 (Main order:1)
row11 (Main)
row8 (Main order:2)
row10 (Main order:3)
Designer
Job(Step_4b_Rec...
Contexts(Step_4...
Component
Run (Job Step_4b...
Job Step_4b_Recommendation_Realtime_Engine_Pipeline
Basic Run
Spark Configuration
Execution
Run
Kill
Clear

Execute the Retail Recommendation Demo:

Watch the execution output window. You will now see your real-time data coming through with recommended products based on your Recommendation Model.

Recommendations are also written to a Cassandra database so they can be referenced by a WebUI to offer, for instance, last minute product suggestions when a customer is about to check-out.

- Once you have seen the results, you can **Kill** the Recommendation Engine and the Push to Kafka jobs to stop the streaming recommendations.


Talend Real-time Big Data Platform (6.2.1.20160704_1411) | Base_Porject (Connection: Local)
Learn
Ask
Exchange
Integration
Profiling
Repository
LOCAL: Base_Porject
Job Designs
Standard
Realtime_Recommendations_Demo
Step_1a_Recommendation_DemoSetup 0.1
Step_1b_Recommendation_DemoSetup 0.1
Step_2_Recommendation_Create_KafkaTopic 0.1
Step_4a_Recommendation_Push_to_Kafka 0.1
Step_5_Recommendation_Read_Cassandra_for_We
Big Data Batch
Realtime_Recommendations_Demo
Step_3_Recommendation_Build_Model_Spark 0.1
Big Data Streaming
Realtime_Recommendations_Demo
Step_4b_Recommendation_Realtime_Engine_Pipelir
Joblets
Route Designs
Services
Contexts
Resources
Outline
Code Viewer
tCassandraConfiguration_1
tCassandraLookupInput_1 (User_Lookup)
tCassandraLookupInput_2 (Product_Lookup)
tCassandraOutput_1 (tCassandraOutput_1
WebUI Outp
tExtractDelimitedFields_1 (Extract Delimited
Fields)
tFileOutputDelimited_1 (File Output
For Big Data Analy
tFilterColumns_1 (Simplify Output)
tFilterRow_12 (Filter Null
 Recommendations)
tKafkaInput_1 (Kafka Stream In)
tLogRow_1 (Terminal Output
For Demo Only)
tMap_1 (User Details)
tMap_2 (Product Details)
Job Step_4a_Recommendation_Push_to_Kafka 0.1
Job Step_4b_Recommendation_Realtime_Engine_Pipeline 0.1
CassandraConfiguration_1
User_Lookup
Product_Lookup
row5 (Lookup)
row9 (Lookup)
Length: 5s
Every: 10s
row1 (Main)
row2 (Main)
Recommendations (Main)
row4 (Main)
row3 (Main)
row6 (Filter)
Recommend_out (Mai
Designer
Code
Jobscript
Job Step_4b_Recommendation_Realtime_Engine_Pipeline
Basic Run
Spark Configuration
Debug Run
Advanced settings
Target Exec
Memory Run
Execution
Run
Kill
Clear

```
[WARN ]: kafka.utils.VerifiableProperties - Property serializer.encoding is not valid
|                                        Terminal Output<br>For Demo Only                                        |
|userid                                 |productid          |lastname |product                              |score              |ip            |
|(AA2E3EC5-653B-4BF7-AE3E-C5653B4BF715)|2766346228795278504|Murdock  |Orbea Ordu M30                       |5.09648902268238 8 |206.28.62.19  |
|(AA2E3EC5-653B-4BF7-AE3E-C5653B4BF715)|2852491450854218181|Murdock  |Jamis Xenith Elite Femme             |4.7793460946338    |206.28.62.19  |
|(AA2E3EC5-653B-4BF7-AE3E-C5653B4BF715)|2718357539142339888|Murdock  |Litespeed L1 Ultegra Bike MD - 2034|4.44102235921291   |206.28.62.19  |
|(C6A93018-32BC-41DE-A8E7-B954FE134B61)|2744899696149769239|Bowman   |Intense Tracer 275C Pro              |15.37460122461172 9|24.253.61.96  |
|(C6A93018-32BC-41DE-A8E7-B954FE134B61)|2772221179266672550|Bowman   |Orbea Ordu M30                       |8.2596789253146 05 |24.253.61.96  |
|(C6A93018-32BC-41DE-A8E7-B954FE134B61)|2781507448317776736|Bowman   |NINER RIP 9 RDO                      |7.00579248367031 4 |24.253.61.96  |
|(1E81310F-3A24-4E9B-BB08-6B6D3C409949)|2839162552428404069|Anderson |Jamis Xenith Elite Femme             |14.61557020757143 2|68.33.16.193  |
|(1E81310F-3A24-4E9B-BB08-6B6D3C409949)|2858266974982907543|Anderson |Niner WFO 9                          |8.73779275599949 8 |68.33.16.193  |
|(1E81310F-3A24-4E9B-BB08-6B6D3C409949)|2705552197228562255|Anderson |Orbea Ordu M30                       |7.5557052440547 8  |68.33.16.193  |
|(31978DC6-CAF5-46CB-8F65-B3DBC30D3C0E)|2853421708558283606|Tillotson|Orbea Ordu M30                       |5.31361612918928 3 |69.230.197.23|
|(31978DC6-CAF5-46CB-8F65-B3DBC30D3C0E)|2781498789662364038|Tillotson|NINER RIP 9 RDO                      |4.57492226127262 9 |69.230.197.23|
|(31978DC6-CAF5-46CB-8F65-B3DBC30D3C0E)|2820971285016817915|Tillotson|Jamis Xenith Elite Femme             |4.39496949006786 5 |69.230.197.23|
|(26A31713-6A80-4164-AA0B-43DA4DFD66C6)|2781498789662364038|Lord     |NINER RIP 9 RDO                      |6.34282655056616 1 |24.4.226.156  |
|(26A31713-6A80-4164-AA0B-43DA4DFD66C6)|2852517931476354313|Lord     |Jamis Xenith Elite Femme             |5.73387307567699 6 |24.4.226.156  |
|(26A31713-6A80-4164-AA0B-43DA4DFD66C6)|2856423227127109194|Lord     |Ibis Mojo HD3 Frame                  |5.62990876904537 4 |24.4.226.156  |
[WARN ]: kafka.utils.VerifiableProperties - Property serializer.encoding is not valid
[WARN ]: kafka.utils.VerifiableProperties - Property serializer.encoding is not valid
[WARN ]: kafka.utils.VerifiableProperties - Property serializer.encoding is not valid
[WARN ]: kafka.utils.VerifiableProperties - Property serializer.encoding is not valid
[WARN ]: kafka.utils.VerifiableProperties - Property serializer.encoding is not valid
[WARN ]: org.talend.datascience.mllib.recommendation.CollaborativeFiltering$ - No recommendation for unknown user: id = 25
[WARN ]: org.talend.datascience.mllib.recommendation.CollaborativeFiltering$ - No recommendation for unknown user: id = 26
|                                        Terminal Output<br>For Demo Only                                        |
|userid                                 |productid          |lastname|product                          |score              |ip             |
|(9D1A8CAC-7692-4372-8FCA-FAB45E4264EC)|2705552197228562255|Davis   |Orbea Ordu M30                   |6.4041589622671 7  |71.236.197.35  |
|(9D1A8CAC-7692-4372-8FCA-FAB45E4264EC)|2830262944250532780|Davis   |SRAM XX1 GXP Crank with 32T Ring|6.02807099783637 4 |71.236.197.35  |
|(9D1A8CAC-7692-4372-8FCA-FAB45E4264EC)|2858632244771561801|Davis   |Jamis Xenith Elite Femme         |5.76416634373872 05|71.236.197.35  |
|(EA1B84CE-3CC1-62FB-9D0B-FB84CAF9F1E7)|2856423227127109194|Fowler  |Ibis Mojo HD3 Frame              |5.41409863227788 4 |134.84.139.120|
|(EA1B84CE-3CC1-62FB-9D0B-FB84CAF9F1E7)|2839951913728935870|Fowler  |Orbea Ordu M30                   |5.24369944654022 8 |134.84.139.120|
|(EA1B84CE-3CC1-62FB-9D0B-FB84CAF9F1E7)|2852491450854218181|Fowler  |Jamis Xenith Elite Femme         |5.2114640292881 5  |134.84.139.120|
|(AD3A6D9A-4675-4BFE-9817-C3A2F334D59F)|2744899696149769239|Phifer  |Intense Tracer 275C Pro          |9.3746906909041 7  |24.167.239.208|
```

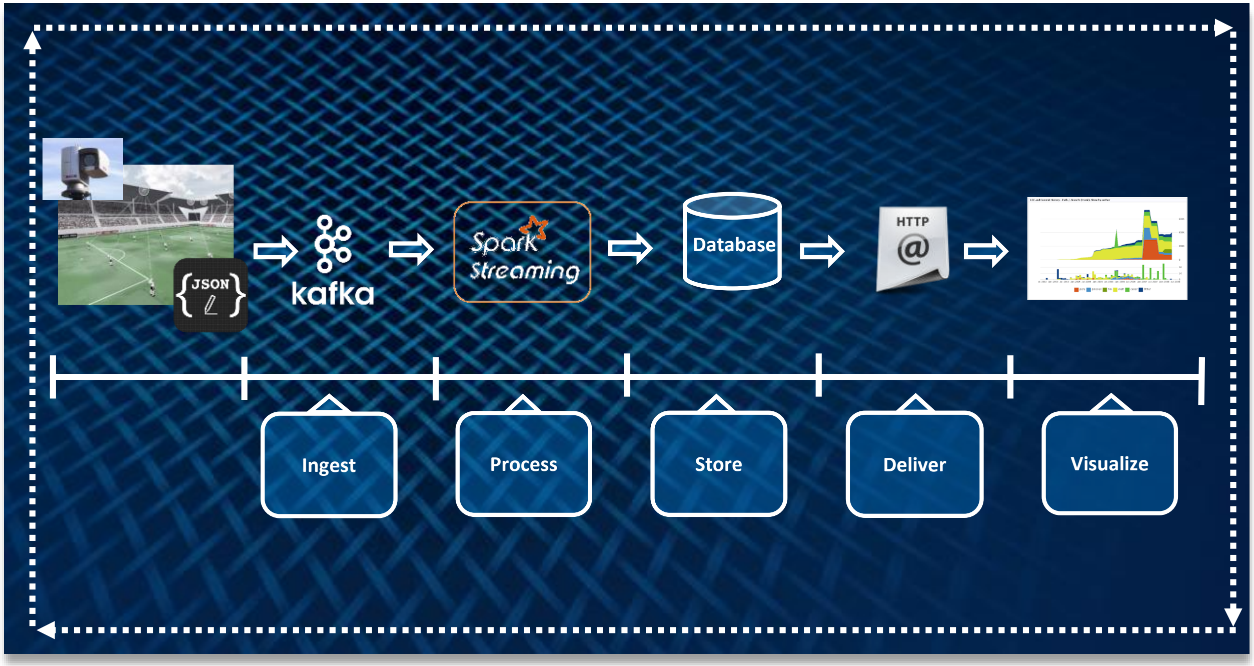
Note: Execution of this demo requires a Hadoop distribution. If a distro hasn't been selected, click here.

Overview:

In this example we will utilize real-time streaming of data through a Kafka queue to track on-field player movements at a sporting event.

You will experience:

- Creating and populating a Kafka queue with real-time streaming data from an IoT device (i.e. field camera sensors).
- Using Spark Streaming technology to calculate speed and distance traveled by individual players.
- Charting player speed and distance in a real-time web-based dashboard.



This Demo highlights:

IoT data to Kafka

Capture IoT data in XML files, then load that data to a Kafka Queue for real-time processing.

Spark Streaming

Use Spark Streaming Technology to quickly calculate player distance and speed as their positions change on the playing field.

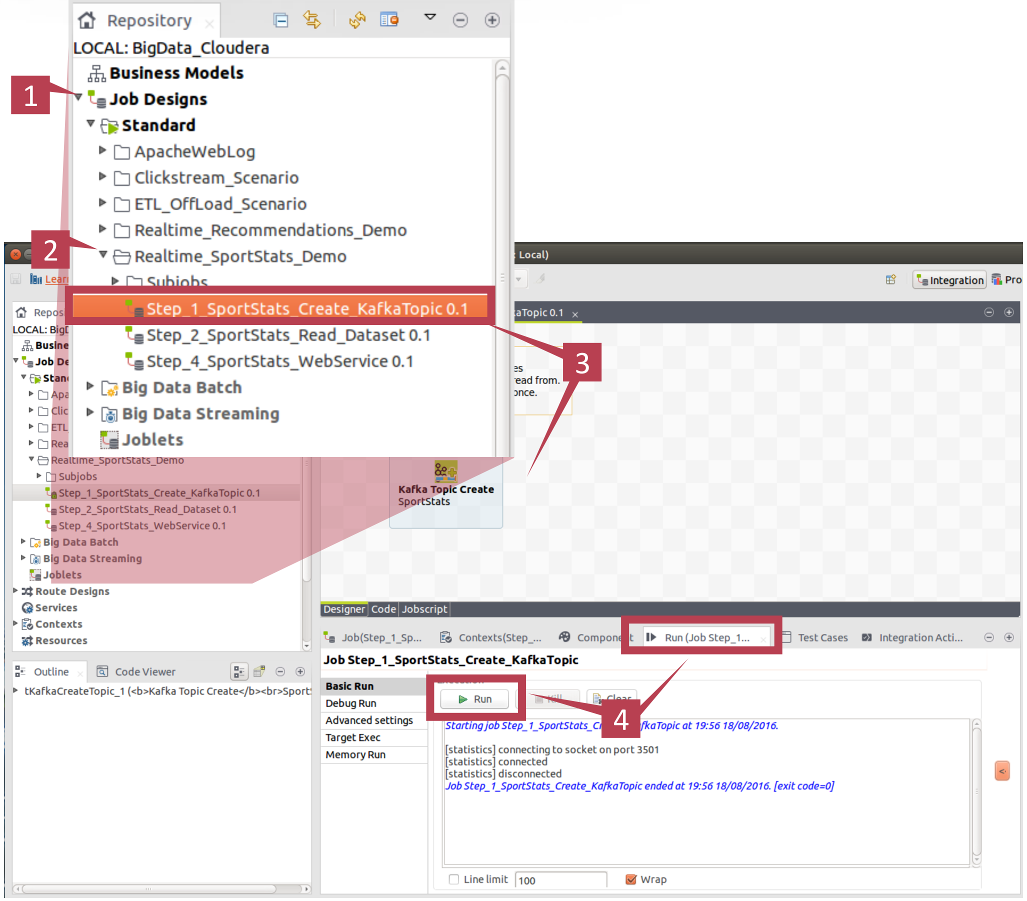
REST Service to Live Dashboard

Use a restful web service to track player movements in a web-based dashboard.

Execute the Sport Stats Demo:

Create a Kafka topic from which live data will stream

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Realtime_SportStats_Demo**
3. Double click on **Step_1_SportStats_Create_KafkaTopic 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.



tKafkaCreateTopic_1 (<b>Kafka Topic Create</b>
Sport

Job Step_1_SportStats_Create_KafkaTopic

```
Starting job Step_1_SportStats_Cr    fkaTopic at 19:56 18/08/2016.

[statistics] connecting to socket on port 3501
[statistics] connected
[statistics] disconnected
Job Step_1_SportStats_Create_KafkaTopic ended at 19:56 18/08/2016. [exit code=0]
```

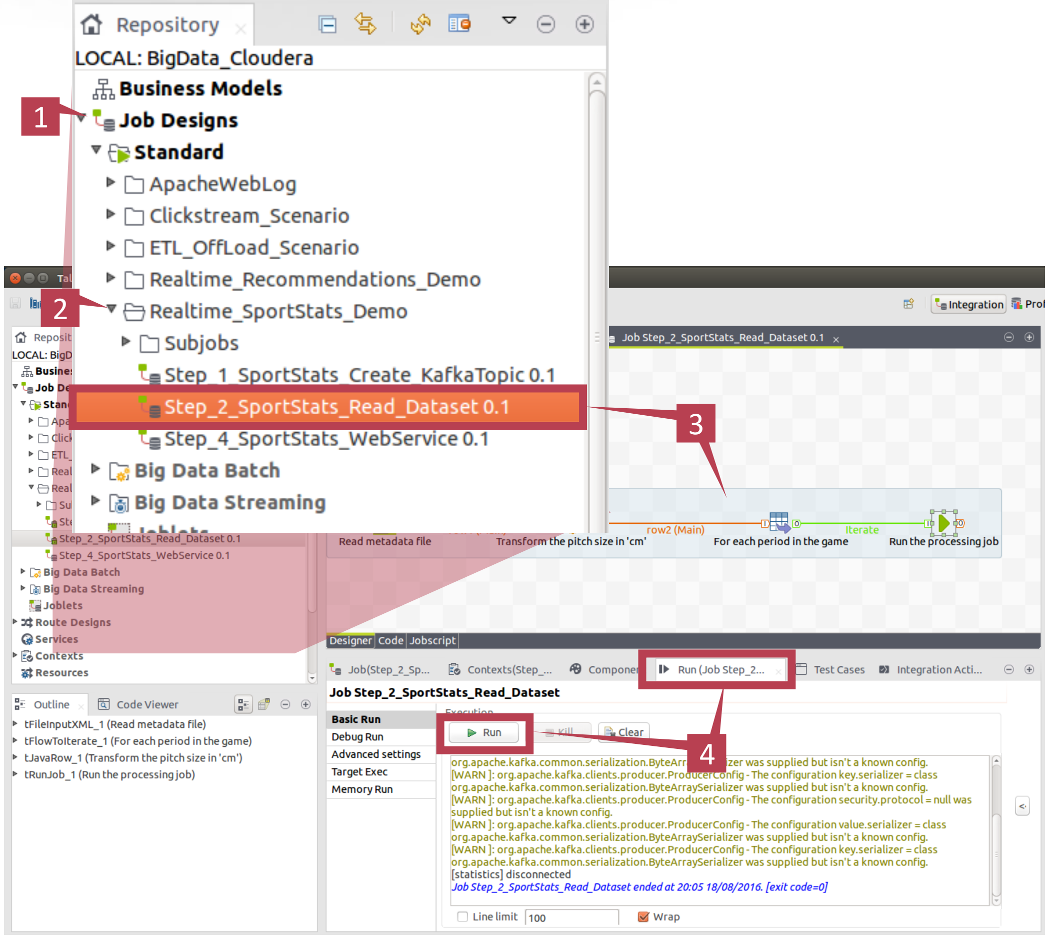
□ Line limit 100 ☑ Wrap

Execute the Sport Stats Demo:

Read data from an XML file (generated by sensor readings, for example) and populate the Kafka topic

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Realtime_SportStats_Demo**
3. Double click on **Step_2_SportStats_Read_Dataset 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

This step simulates live player-tracking data being fed to a Kafka topic.



- tFileInputXML_1 (Read metadata file)
- tFlowToIterate_1 (For each period in the game)
- tJavaRow_1 (Transform the pitch size in 'cm')
- tRunJob_1 (Run the processing job)

Job Step_2_SportStats_Read_Dataset

```
org.apache.kafka.common.serialization.ByteArr    izer was supplied but isn't a known config.
[WARN ]: org.apache.kafka.clients.producer.ProducerConfig - The configuration key.serializer = class
org.apache.kafka.common.serialization.ByteArraySerializer was supplied but isn't a known config.
[WARN ]: org.apache.kafka.clients.producer.ProducerConfig - The configuration security.protocol = null was
supplied but isn't a known config.
[WARN ]: org.apache.kafka.clients.producer.ProducerConfig - The configuration value.serializer = class
org.apache.kafka.common.serialization.ByteArraySerializer was supplied but isn't a known config.
[WARN ]: org.apache.kafka.clients.producer.ProducerConfig - The configuration key.serializer = class
org.apache.kafka.common.serialization.ByteArraySerializer was supplied but isn't a known config.
[statistics] disconnected
Job Step_2_SportStats_Read_Dataset ended at 20:05 18/08/2016. [exit code=0]
```

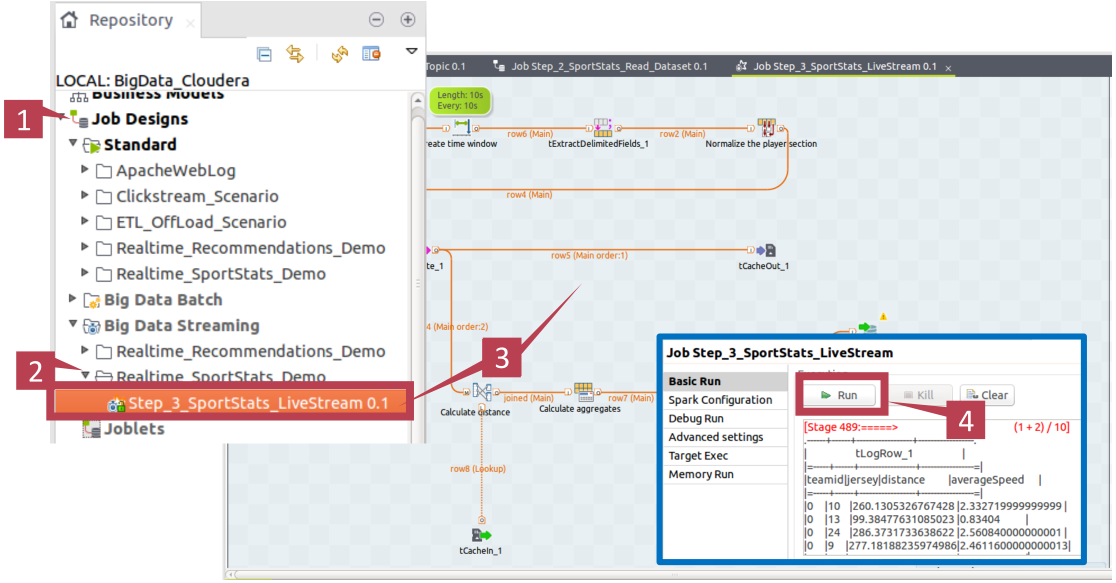
Execute the Sport Stats Demo:

In this job:

- A Kafka Consumer reads the sensor data.
- A tWindow component controls how often data is read from the Kafka topic – in this case, 10 seconds worth of data is read every 10 seconds.
- The data is normalized for easier processing.
- Using the tCache components the process calculates distance and speed based on current and previous player positions.
- The resultant data is sent to 2 output streams
  - ✓ **Execution window** for viewing purposes
  - ✓ **MySQL Database** where it will be read by a web service to generate dashboard graphics. (MySQL is running on a Docker container)

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Streaming > Realtime_SportStats_Demo**
3. Double click on **Step_3_SportStats_LiveStream 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

After starting the SportStats Live Stream, continue to the next steps of the demo.



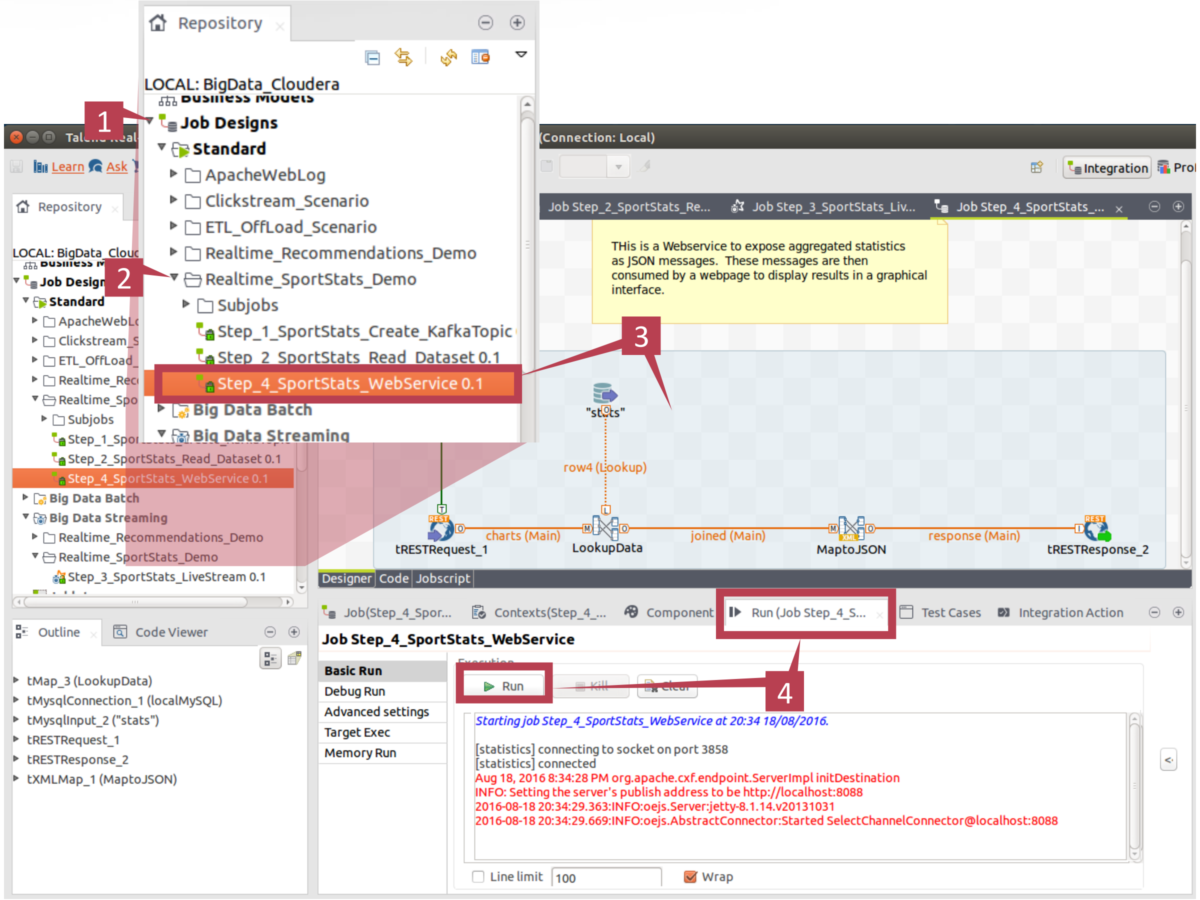
```
[Stage 489:=====>                                    (1 + 2) / 10]
.------+------+-----------------+-----------------.
|               tLogRow_1                |
|=-----+------+-----------------+----------------=|
|teamid|jersey|distance         |averageSpeed     |
|=-----+------+-----------------+----------------=|
|0     |10    |260.1305326767428 |2.332719999999999 |
|0     |13    |99.38477631085023 |0.83404           |
|0     |24    |286.3731733638622 |2.560840000000001 |
|0     |9     |277.18188235974986|2.461160000000013|
```

Execute the Sport Stats Demo:

Start the Web Service to populate the Sport Stats web-based dashboard

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Realtime_SportStats_Demo**
3. Double click on **Step_4_SportStats_WebService 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

With the Web Service running, continue to the next step in this demo.



THis is a Webservice to expose aggregated statistics as JSON messages. These messages are then consumed by a webpage to display results in a graphical interface.

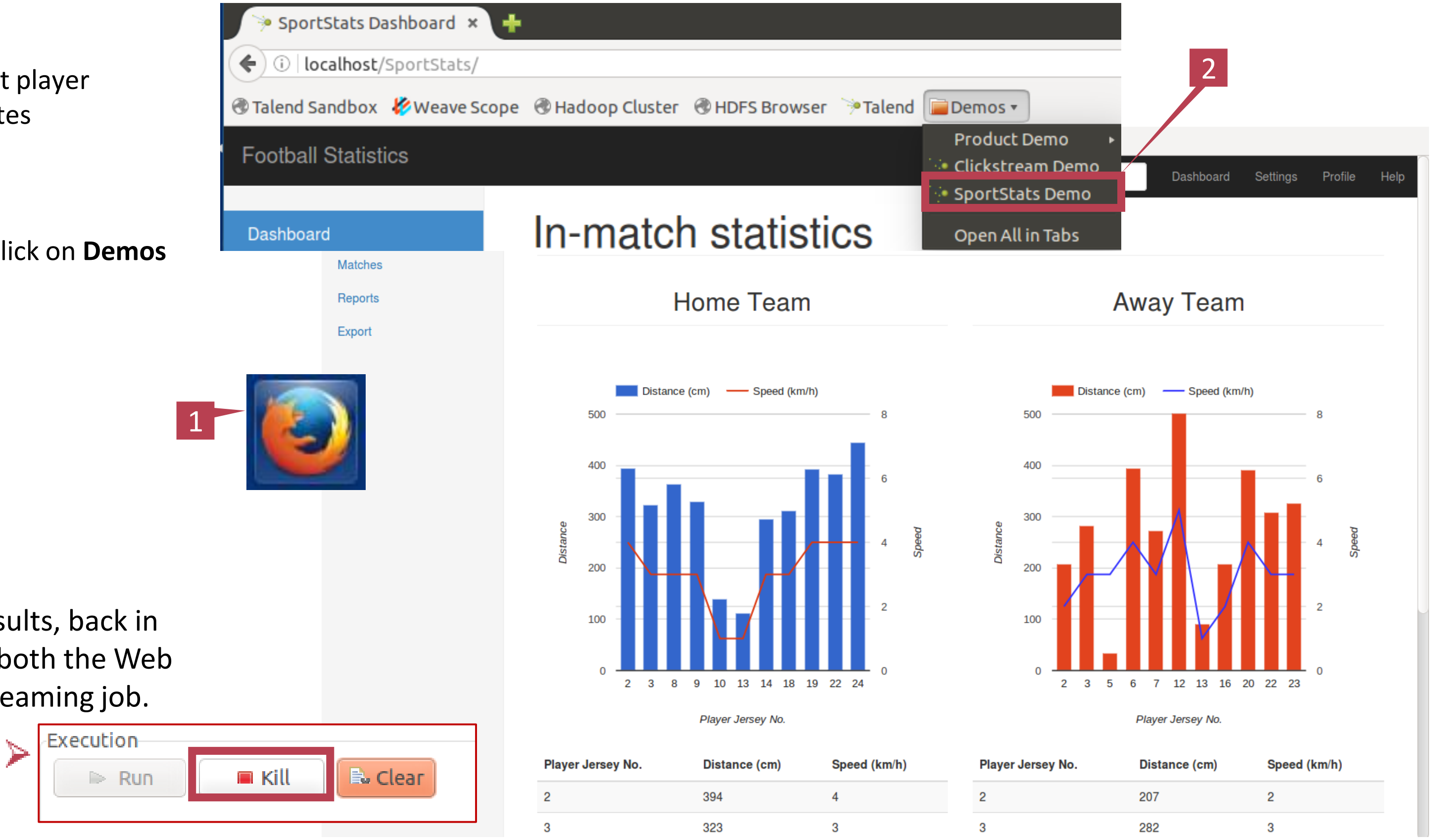
```
Starting job Step_4_SportStats_WebService at 20:34 18/08/2016.

[statistics] connecting to socket on port 3858
[statistics] connected
Aug 18, 2016 8:34:28 PM org.apache.cxf.endpoint.ServerImpl initDestination
INFO: Setting the server's publish address to be http://localhost:8088
2016-08-18 20:34:29.363:INFO:oejs.Server:jetty-8.1.14.v20131031
2016-08-18 20:34:29.669:INFO:oejs.AbstractConnector:Started SelectChannelConnector@localhost:8088
```

Execute the Sport Stats Demo:

Watch the Live Dashboard reflect player movements with real-time updates

1. Open Firefox Web Browser.
2. On the Bookmarks toolbar, click on **Demos > SportStats Demo**

- Once you have seen the results, back in Talend Studio, you can **Kill** both the Web Service job and the Live Streaming job.



Player Jersey No.	Distance (cm)	Speed (km/h)
2	394	4
3	323	3

Player Jersey No.	Distance (cm)	Speed (km/h)
2	207	2
3	282	3

Talend Big Data Sandbox

Big Data Insights Cookbook

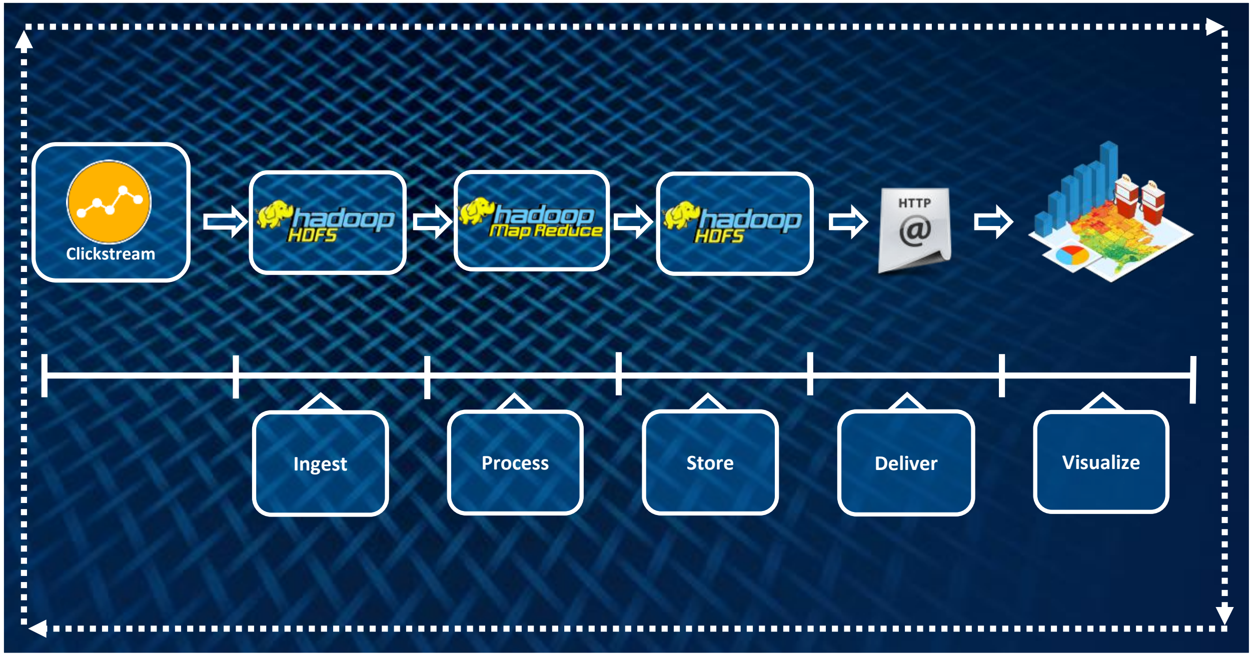
Note: Execution of this demo requires a Hadoop distribution. If a distro hasn't been selected, click here.

Overview:

In this example we demonstrate using native Map Reduce to enrich a dataset and aggregate the results for different web-based dashboards.

You will experience:

- Data loading to HDFS.
- Using MapReduce to enrich and aggregate data within the Hadoop Environment.
- Use of 3rd party graphing tools to generate a web-based dashboard of the calculated results.



This Demo will highlight:

HDFS

Read and Write data to HDFS with simple components from Talend

Native MapReduce

Use Talend's MapReduce components to enrich and analyze data, natively, in Hadoop

Insights

Feed your analysis data to a graphing tool such as Microsoft Excel or Tableau for stunning displays of the results.

talend

Talend Big Data Sandbox

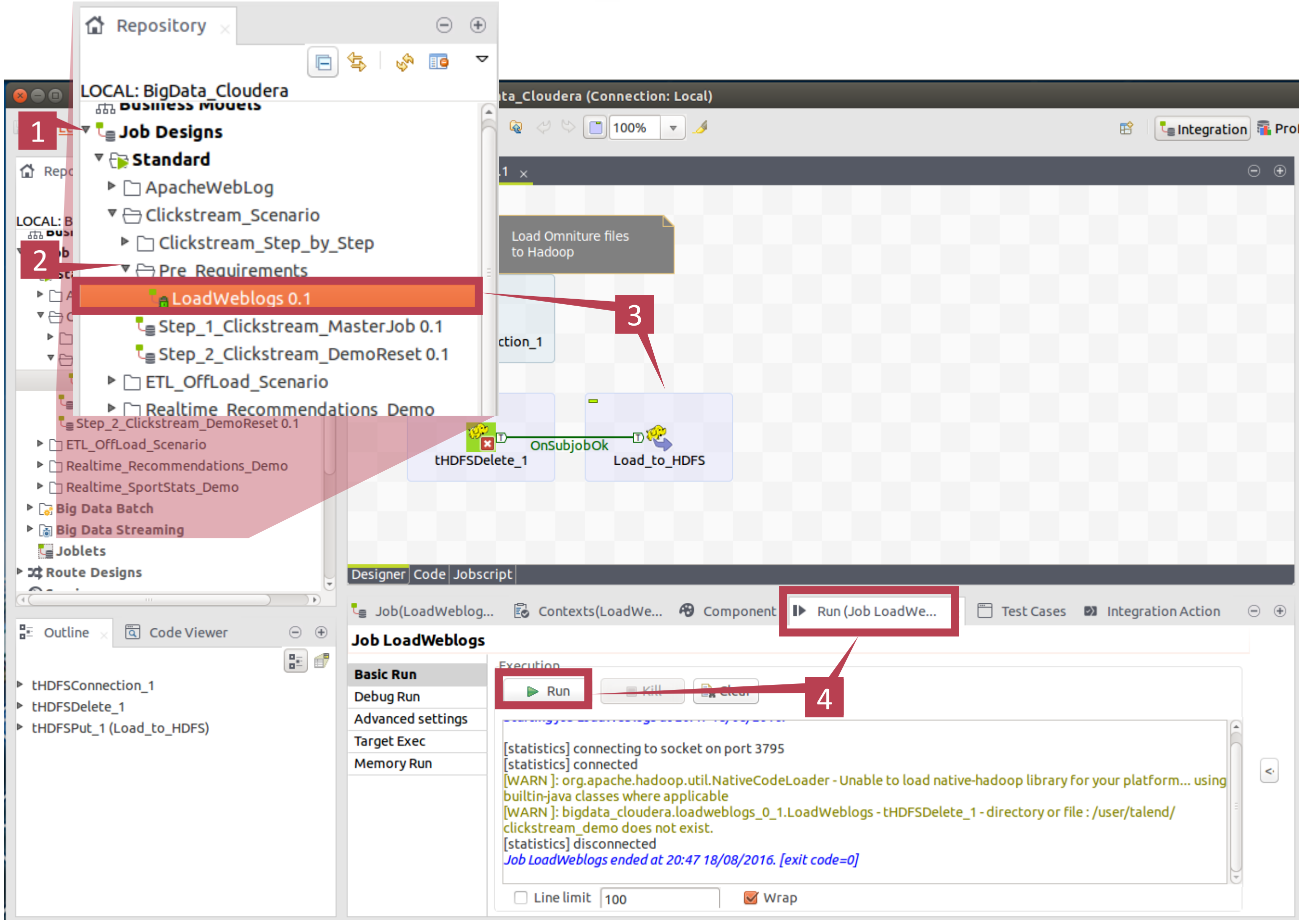
Big Data Insights Cookbook

Demo Setup:

Load data to HDFS.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Clickstream_Scenario > Pre_Requirements**
3. Double click on **LoadWeblogs 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

When this job completes, look-up files will be uploaded to HDFS for use by the MapReduce jobs.



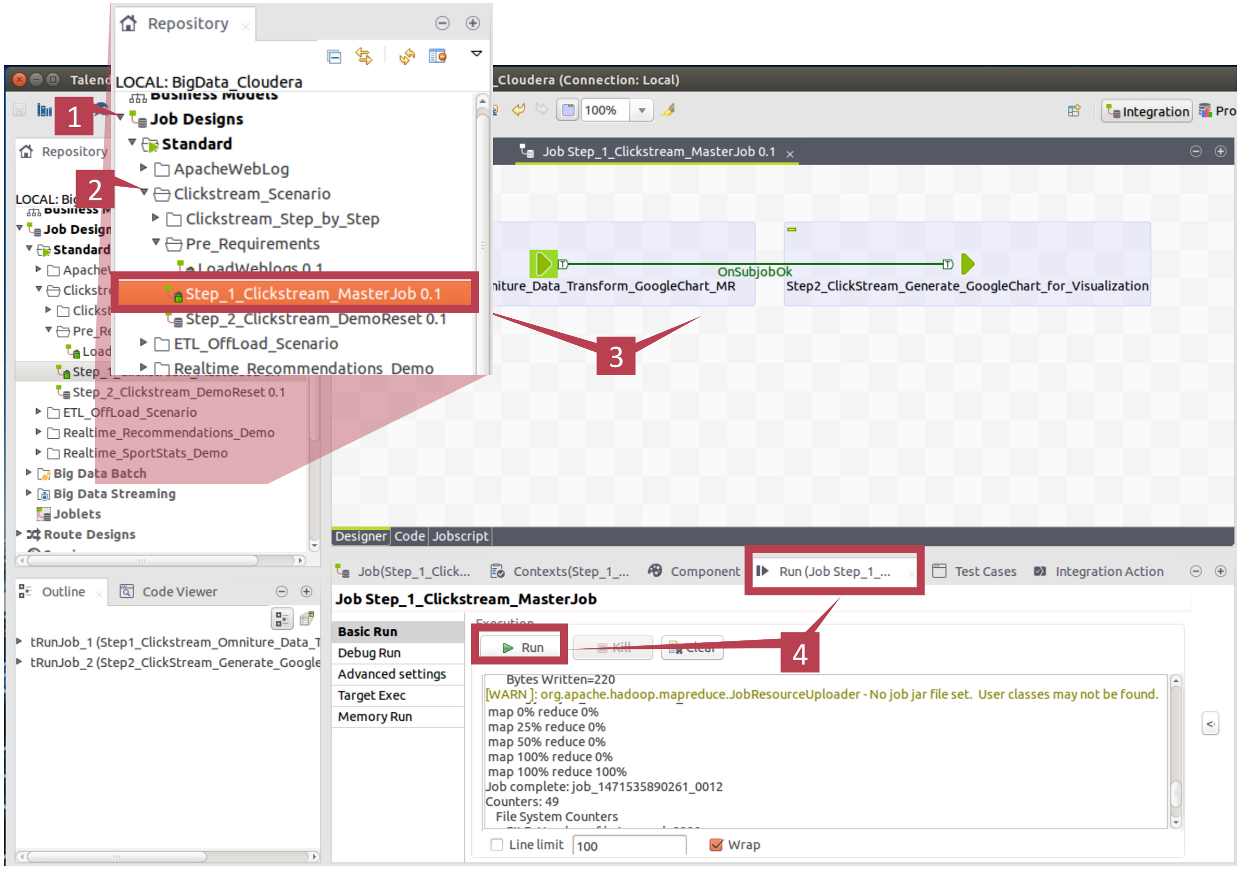
```
[statistics] connecting to socket on port 3795
[statistics] connected
[WARN ]: org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
[WARN ]: bigdata_cloudera.loadweblogs_0_1.LoadWeblogs - tHDFSDelete_1 - directory or file : /user/talend/
clickstream_demo does not exist.
[statistics] disconnected
Job LoadWeblogs ended at 20:47 18/08/2016. [exit code=0]
```

Execute the Clickstream Demo:

The result of this process is aggregated data indicating the product interests of different areas across the United States for visualization within a Google Chart.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Clickstream_Scenario**
3. Double click on **Step_1_Clickstream_MasterJob 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

Note: If asked to Download and Install additional Jar files, Click on **Download and Install**.

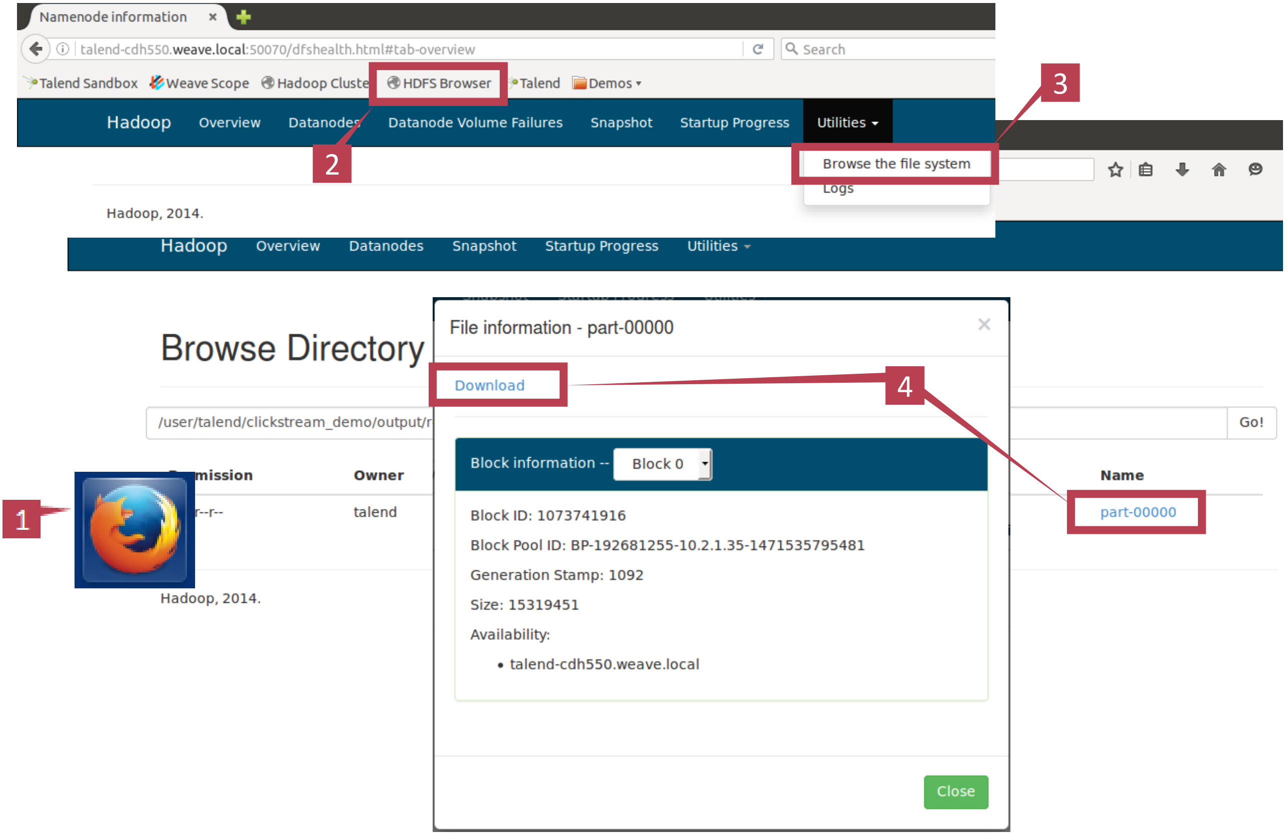


```
    Bytes Written=220
[WARN ]: org.apache.hadoop.mapreduce.JobResourceUploader - No job jar file set.  User classes may not be found.
 map 0% reduce 0%
 map 25% reduce 0%
 map 50% reduce 0%
 map 100% reduce 0%
 map 100% reduce 100%
Job complete: job_1471535890261_0012
Counters: 49
  File System Counters
```

Execute the Apache Weblog Demo:

View the data in HDFS.

1. Open Firefox
2. Click on the bookmarked link titled **HDFS Browser**
3. In the **Utilities** Dropdown, select **Browse the File System** and navigate to */user/talend/clickstream_demo/output/results*
4. To view the data file, you must download it from HDFS. This can be done right within the web browser by clicking on *part-00000* and choosing **download**



/user/talend/clickstream_demo/output/r

-rw-r--r-- talend part-00000

File information - part-00000

Download

Block information -- Block 0

Block ID: 1073741916

Block Pool ID: BP-192681255-10.2.1.35-1471535795481

Generation Stamp: 1092

Size: 15319451

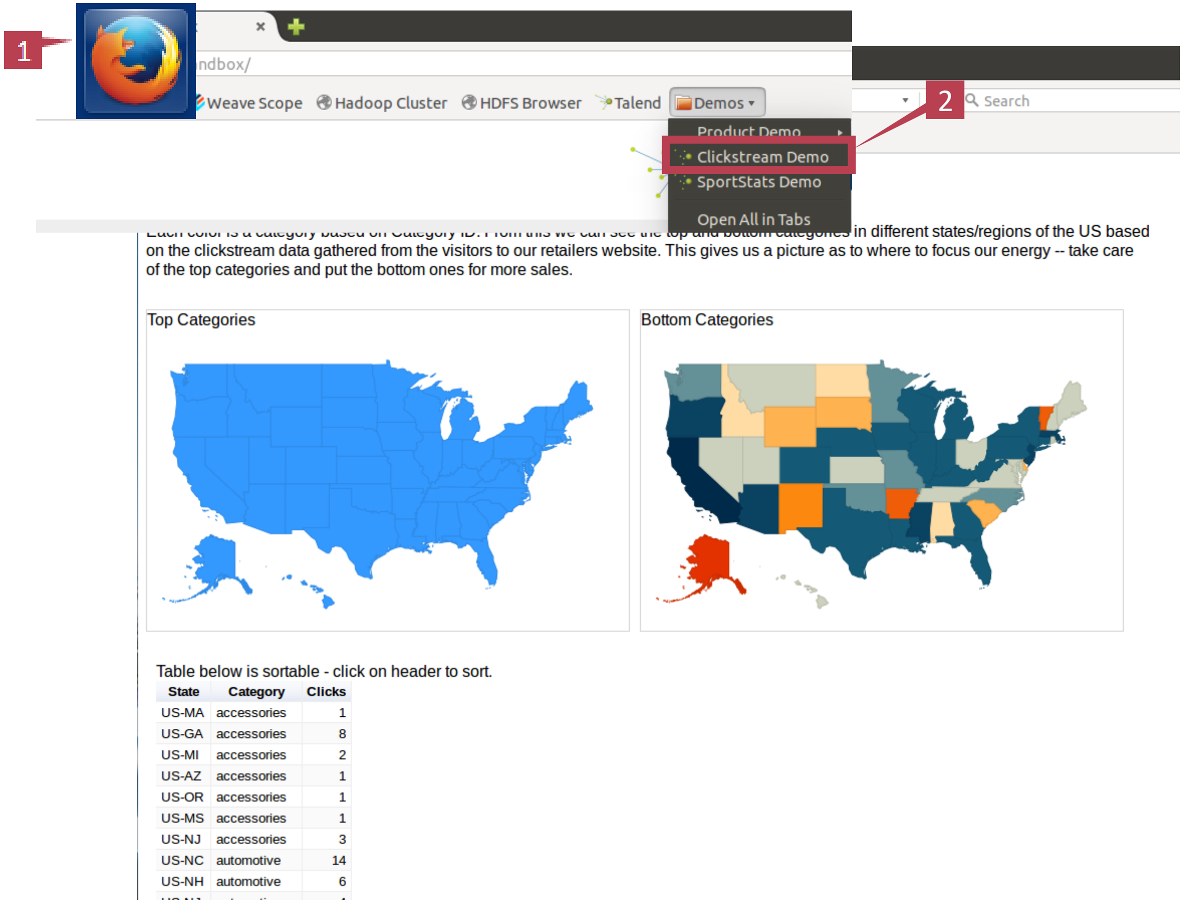
Availability:

- talend-cdh550.weave.local

Execute the Clickstream Demo:

View the analytical analysis dashboard.

1. Open **Firefox** Web Browser.
2. On the Bookmarks toolbar, click on **Demos > Clickstream Demo**
3. **Mouse over** the states to see the counts.



Each color is a category based on Category ID. From this we can see the top and bottom categories in different states/regions of the US based on the clickstream data gathered from the visitors to our retailers website. This gives us a picture as to where to focus our energy -- take care of the top categories and put the bottom ones for more sales.

Table below is sortable - click on header to sort.

State	Category	Clicks
US-MA	accessories	1
US-GA	accessories	8
US-MI	accessories	2
US-AZ	accessories	1
US-OR	accessories	1
US-MS	accessories	1
US-NJ	accessories	3
US-NC	automotive	14
US-NH	automotive	6

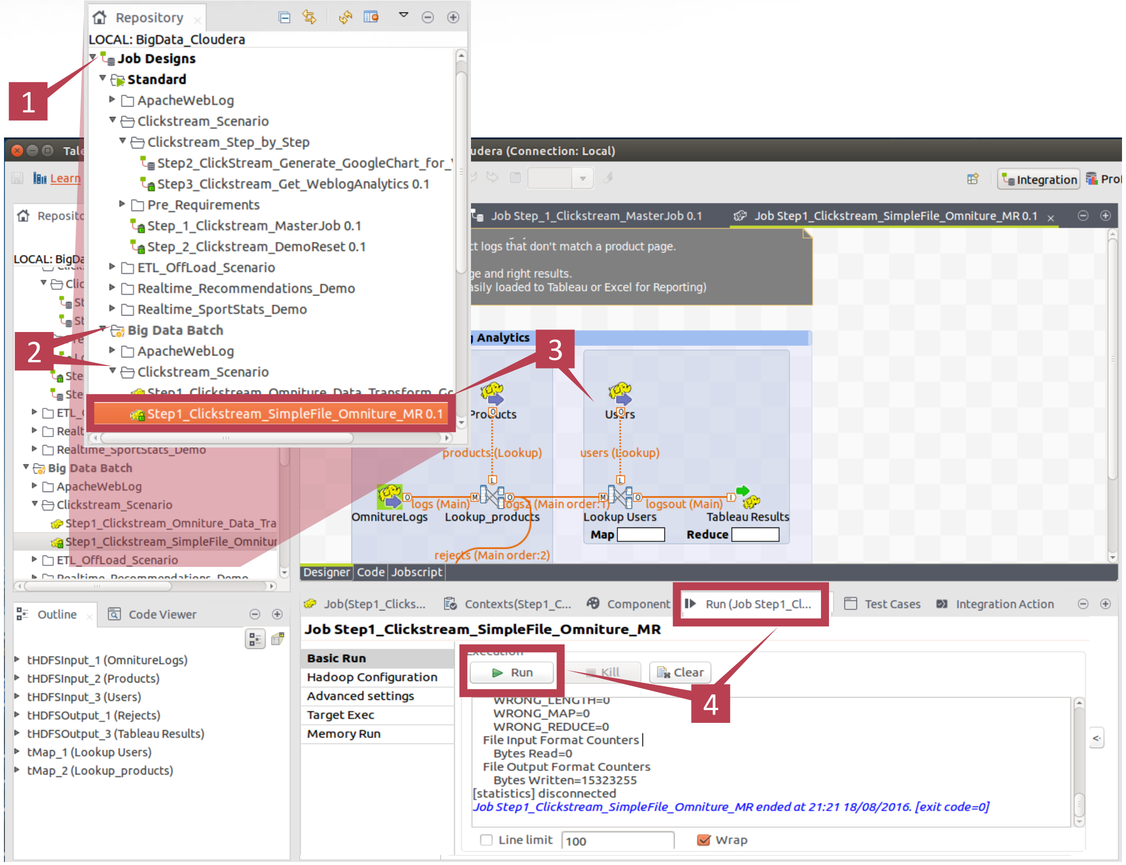
Execute the Clickstream Demo:

Additional analysis can be done to calculate the age and gender of users accessing specific links.

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Batch > Clickstream_Scenario**
3. Double click on **Step_1_Clickstream_SimpleFile_Omniture_MR 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

The results of this job can be found in the HDFS File Browser: */user/talend/clickstream_demo/output/results*

With our analysis data in HDFS, we can load into a Hive Table for further querying or import to a visualization tool. Continue to the next steps of the demo to see how this can be done.



ct logs that don't match a product page.

ge and right results.
asily loaded to Tableau or Excel for Reporting)

Job Step1_Clickstream_SimpleFile_Omniture_MR

Line limit: 100

```
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=15323255
[statistics] disconnected
Job Step1_Clickstream_SimpleFile_Omniture_MR ended at 21:21 18/08/2016. [exit code=0]
```

Talend Big Data Sandbox

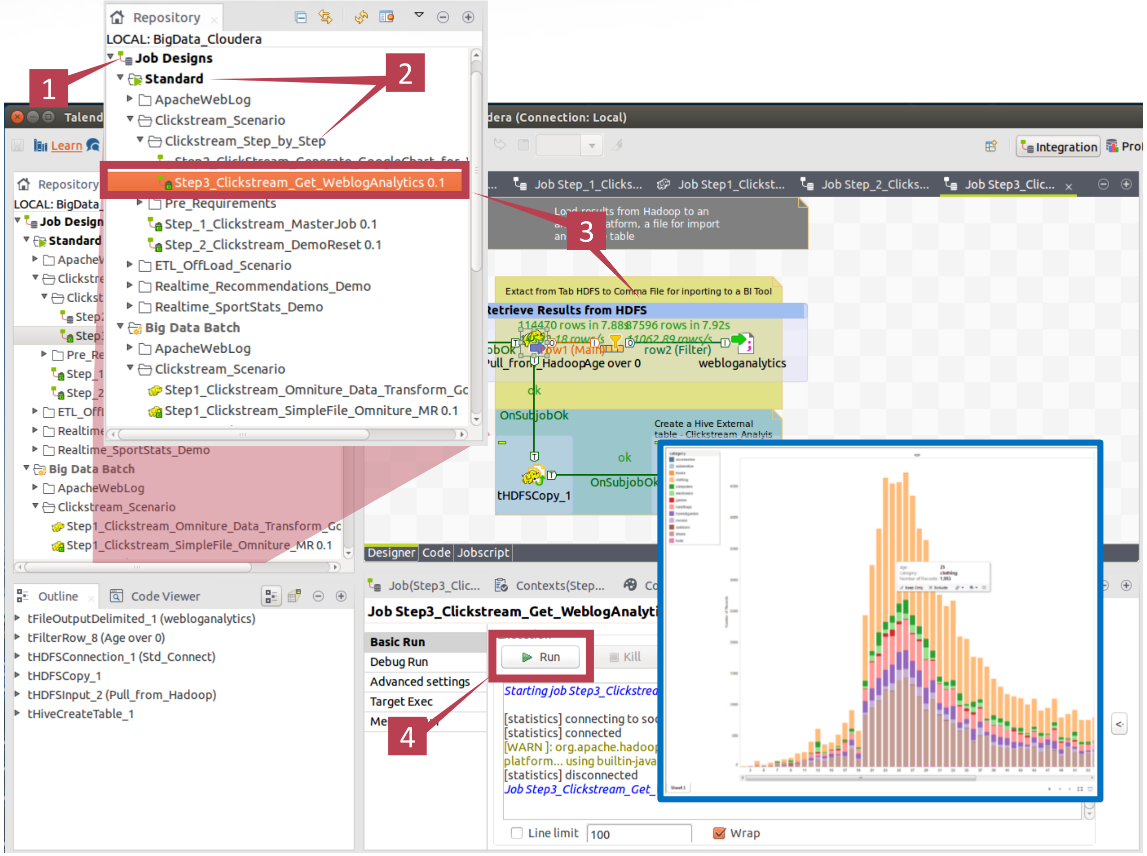
Big Data Insights Cookbook

Execute the Clickstream Demo:

With our analysis complete, we can pull the raw file from HDFS or even put it into a Hive table for further querying.

1. Navigate to the **Job Designs** folder.
2. Navigate to **Standard > Clickstream_Scenario > Clickstream_Step_by_Step**
3. Double click on **Step3_Clickstream_Get_WeblogAnalytics 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

The results of this job can be found on the local VM file system: */home/talend/Documents/Clickstream/webloganalytics.csv*



```
Starting job Step3_Clickstrea
[statistics] connecting to so
[statistics] connected
[WARN ]: org.apache.hadoo
platform... using builtin-java
[statistics] disconnected
Job Step3_Clickstream_Get_
```

➢ This file could be imported to MS Excel or other BI tools like Tableau (not included in the Big Data Sandbox) to generate additional dashboards.

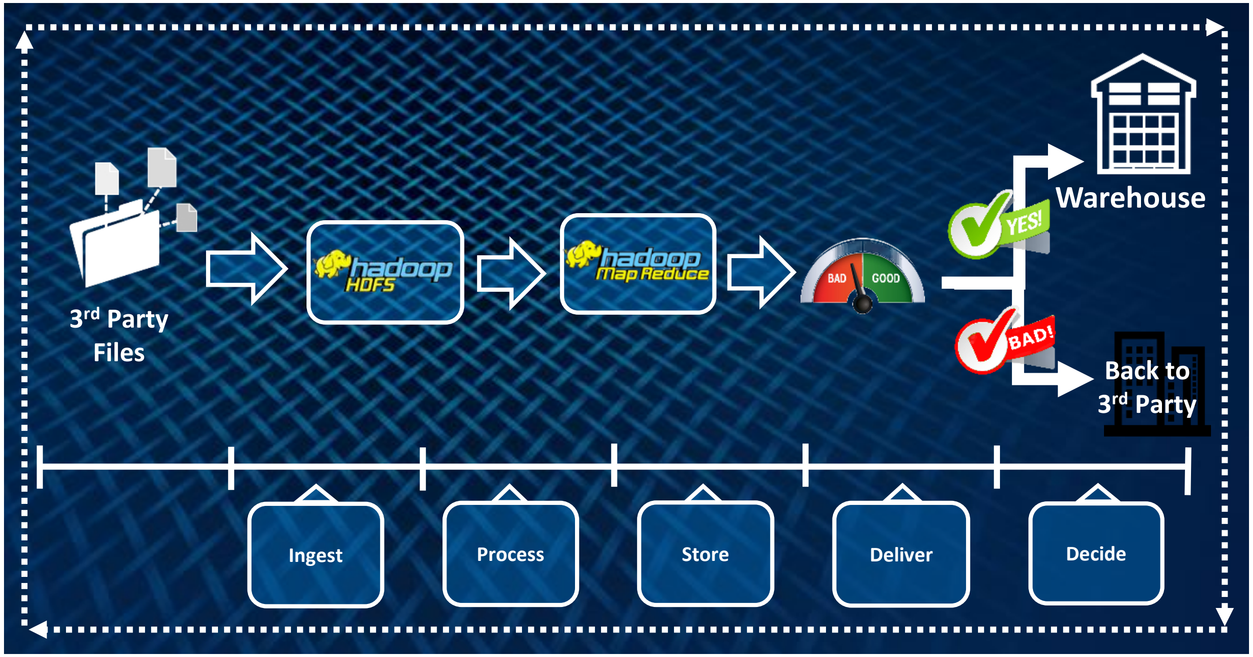
Note: Execution of this demo requires a Hadoop distribution. If a distro hasn't been selected, click here.

Overview:

In this example we demonstrate how using Talend with Hadoop can speed up and simplify processing large volumes of 3rd Party Data. The sample data is simulating a Life Sciences Prescribing habits data file from a 3rd Party vendor.

You will experience:

- optimizing your data warehouse by off-loading the ETL overhead to Hadoop and HDFS.
- Fast, Pre-load analytics on large volume datasets.
- Multiple Reports from same datasets to make informed and intelligent business decision that could decrease spend or increase revenue.



This Demo will highlight:

Large volume processing

With Talend and Hadoop, you can process Gigabytes and Terabytes of data in a fraction of the time.

Pre-load Analytics

By analyzing large volumes of data BEFORE loading it to your Data Warehouse, you eliminate the overhead of costly data anomalies in the Data Warehouse.

ETL Off-loading

Utilizing Talend with a Hadoop Cluster, you can optimize your Data Warehouse by removing the costly overhead of data processing.

Talend Big Data Sandbox

Big Data Insights Cookbook

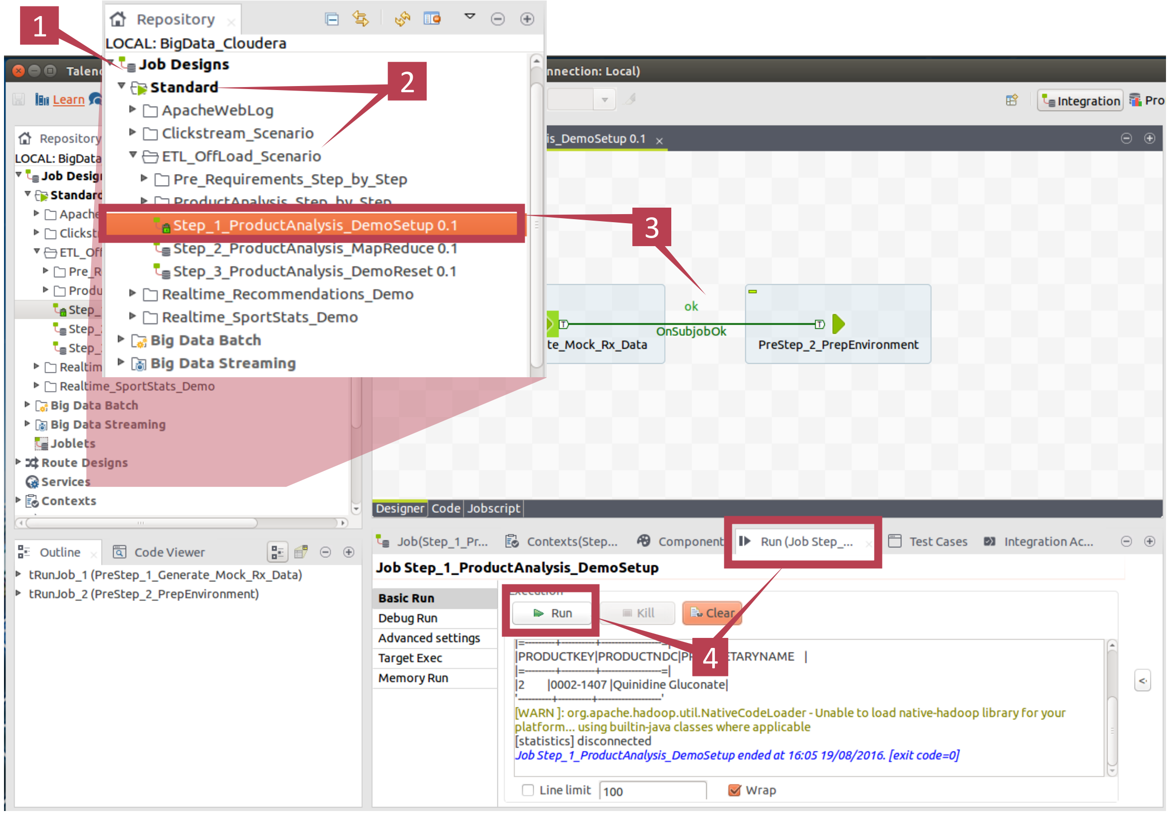
Note: To quickly and easily see the value of the ETL Off-Load Demo, proceed with the below steps. If you would like a more in-depth experience and more control over the source data, Click here…

Demo Setup:

To Execute this demo, you must first generate the source files for processing within Hadoop

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario**
3. Double click on **Step_1_ProductAnalysis_DemoSetup 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

When this job completes, Source files will reside on the Virtual Machine to be processed by the demo. Additionally, within HDFS an initial report will have been generated by which the demo will compare for analysis.



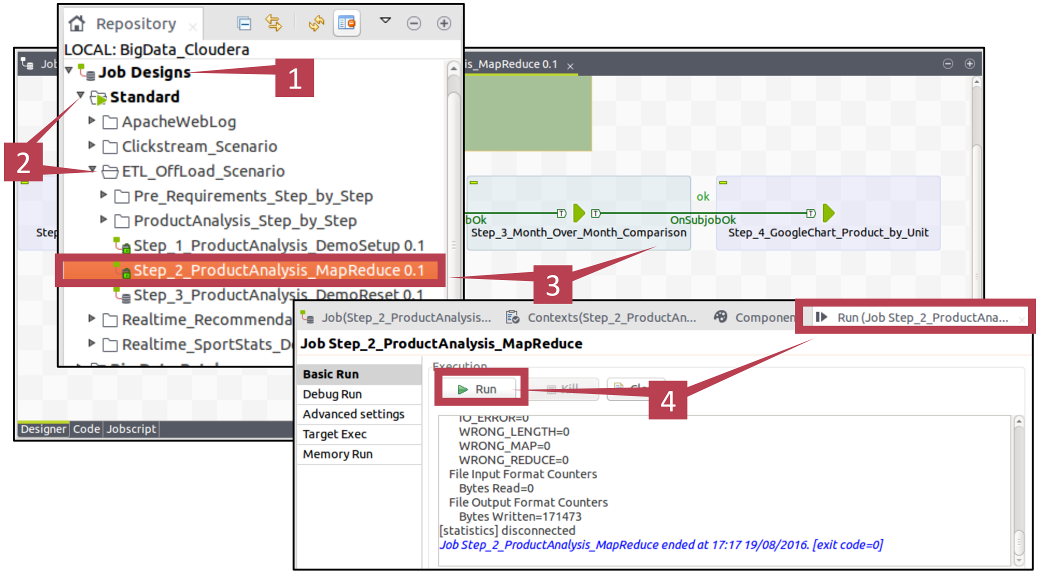
```
|=---------+----------+----------------=|
|PRODUCTKEY|PRODUCTNDC|PR    ETARYNAME  |
|=---------+----------+----------------=|
|2         |0002-1407 |Quinidine Gluconate|
'----------+----------+-----------------'
[WARN ]: org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[statistics] disconnected
Job Step_1_ProductAnalysis_DemoSetup ended at 16:05 19/08/2016. [exit code=0]
```

Execute the ETL Off-Load "One-Click" Demo:

In this "One-Click" version of the demo:

- Source files are placed in HDFS.
- MapReduce is used to collectively analyze all the compressed files.
- The resultant analysis is then compared to the previous months results and reports are generated.
- The generated reports are then sent to the Google Charts API for a graphical representation of the data.
- The resultant reports can be viewed in a web browser:
  - ✓ **Product by Physician** shows the number of prescriptions a physician has written for a particular drug
  - ✓ **Net Change** shows the total number of prescriptions for a particular drug across all physicians

1. Navigate to the **Job Designs** folder.
2. Click on **Standard >ETL_OffLoad_Scenario**
3. Double click on **Step_2_ProductAnalysis_MapReduce 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.


Repository
LOCAL: BigData_Cloudera
Job Designs
Standard
ApacheWebLog
Clickstream_Scenario
ETL_OffLoad_Scenario
Pre_Requirements_Step_by_Step
ProductAnalysis_Step_by_Step
Step_1_ProductAnalysis_DemoSetup 0.1
Step_2_ProductAnalysis_MapReduce 0.1
Step_3_ProductAnalysis_DemoReset 0.1
Realtime_Recommenda
Realtime_SportStats_D
1
2
3
4
is_MapReduce 0.1
ok
OnSubjobOk
Step_3_Month_Over_Month_Comparison
Step_4_GoogleChart_Product_by_Unit
Designer
Code
Jobscript
Job(Step_2_ProductAnalysis...
Contexts(Step_2_ProductAn...
Componen
Run (Job Step_2_ProductAna...
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Execution
Run
Kill

Job Step_2_ProductAnalysis_MapReduce

```
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=171473
[statistics] disconnected
Job Step_2_ProductAnalysis_MapReduce ended at 17:17 19/08/2016. [exit code=0]
```

Click Here to finish this demo…

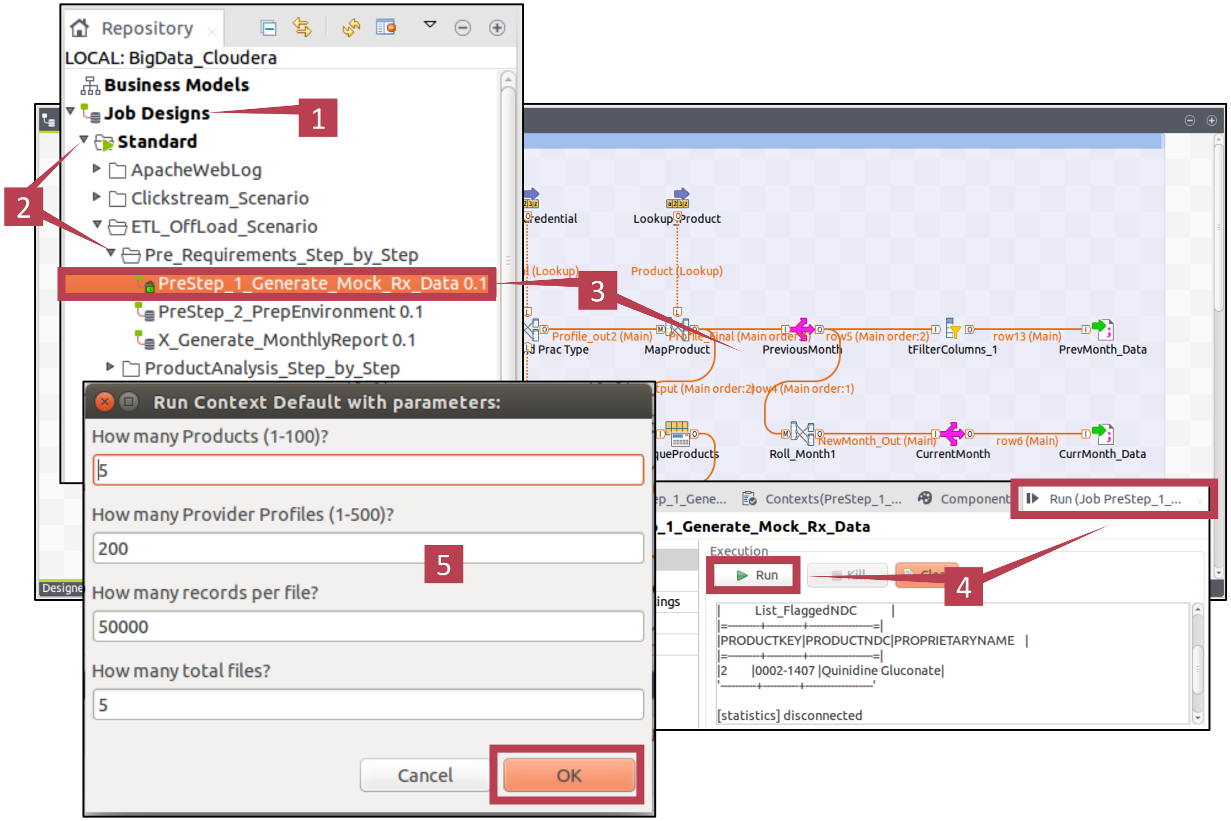
Talend Big Data Sandbox

Big Data Insights Cookbook

Demo Setup:

In this step-by-step version of the demo, you will see just how simple it is to work with Talend and Hadoop. You will also have more control over the source data used within the demo for a more personalized experience.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario > Pre_Requirements_Step_by_Step**
3. Double click on **PreStep_1_Generate_Mock_Rx_Data 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.
5. When the job starts, **edit the values** as you wish *(staying within the suggested parameters and keeping in mind you are working in a virtual environment with limited space)* or leave the default values. Click **OK** when done.



Repository

LOCAL: BigData_Cloudera

- Business Models
- Job Designs
  - Standard
    - ApacheWebLog
    - Clickstream_Scenario
    - ETL_OffLoad_Scenario
      - Pre_Requirements_Step_by_Step
        - PreStep_1_Generate_Mock_Rx_Data 0.1
        - PreStep_2_PrepEnvironment 0.1
        - X_Generate_MonthlyReport 0.1
      - ProductAnalysis_Step_by_Step

Run Context Default with parameters:

How many Products (1-100)?

5

How many Provider Profiles (1-500)?

200

How many records per file?

50000

How many total files?

5

Cancel | OK

_1_Generate_Mock_Rx_Data

Execution

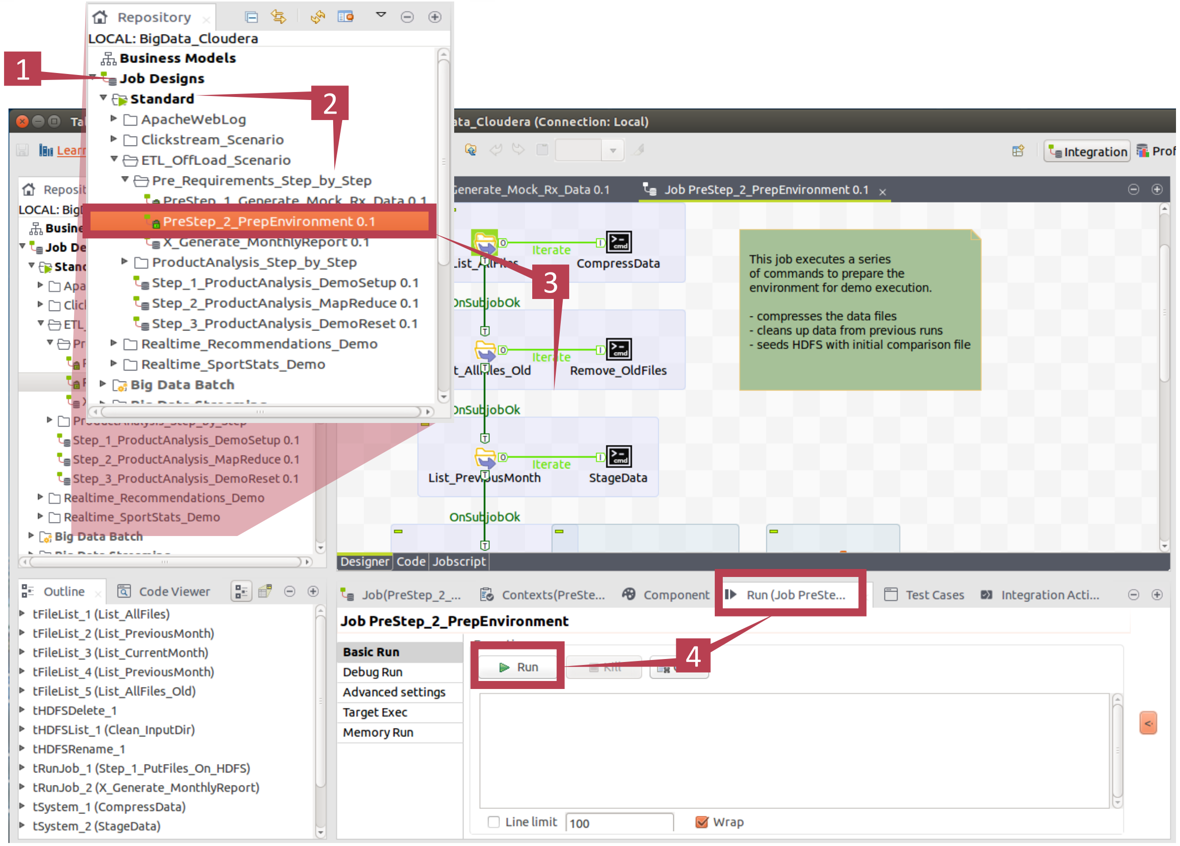
```
|          List_FlaggedNDC        |
|=---------+---------+---------------=|
|PRODUCTKEY|PRODUCTNDC|PROPRIETARYNAME   |
|=---------+---------+---------------=|
|2        |0002-1407 |Quinidine Gluconate|
'---------+---------+---------------'

[statistics] disconnected
```

Demo Setup (cont.):

Once you have generated your Mock Rx data, you will need to initialize the Hadoop environment with comparison data – in this case, it would be the ”Previous Month” analysis.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario > Pre_Requirements_Step_by_Step**
3. Double click on **PreStep_2_PrepEnvironment 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

Note: Once this job completes, you are ready to execute the step-by-step ETL Off-Load Demo.


1
2
3
4
Repository
LOCAL: BigData_Cloudera
Business Models
Job Designs
Standard
ApacheWebLog
Clickstream_Scenario
ETL_OffLoad_Scenario
Pre_Requirements_Step_by_Step
PreStep_1_Generate_Mock_Rx_Data 0.1
PreStep_2_PrepEnvironment 0.1
X_Generate_MonthlyReport 0.1
ProductAnalysis_Step_by_Step
Step_1_ProductAnalysis_DemoSetup 0.1
Step_2_ProductAnalysis_MapReduce 0.1
Step_3_ProductAnalysis_DemoReset 0.1
Realtime_Recommendations_Demo
Realtime_SportStats_Demo
Big Data Batch
ata_Cloudera (Connection: Local)
Integration
Generate_Mock_Rx_Data 0.1
Job PreStep_2_PrepEnvironment 0.1
List_AllFiles
Iterate
CompressData
OnSubjobOk
t_AllFiles_Old
Remove_OldFiles
List_PreviousMonth
StageData
Designer
Code
Jobscript
Outline
Code Viewer
tFileList_1 (List_AllFiles)
tFileList_2 (List_PreviousMonth)
tFileList_3 (List_CurrentMonth)
tFileList_4 (List_PreviousMonth)
tFileList_5 (List_AllFiles_Old)
tHDFSDelete_1
tHDFSList_1 (Clean_InputDir)
tHDFSRename_1
tRunJob_1 (Step_1_PutFiles_On_HDFS)
tRunJob_2 (X_Generate_MonthlyReport)
tSystem_1 (CompressData)
tSystem_2 (StageData)
Job(PreStep_2_...
Contexts(PreSte...
Component
Run (Job PreSte...
Test Cases
Integration Acti...
Job PreStep_2_PrepEnvironment
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Kill
□ Line limit 100
☑ Wrap

This job executes a series of commands to prepare the environment for demo execution.

- compresses the data files
- cleans up data from previous runs
- seeds HDFS with initial comparison file

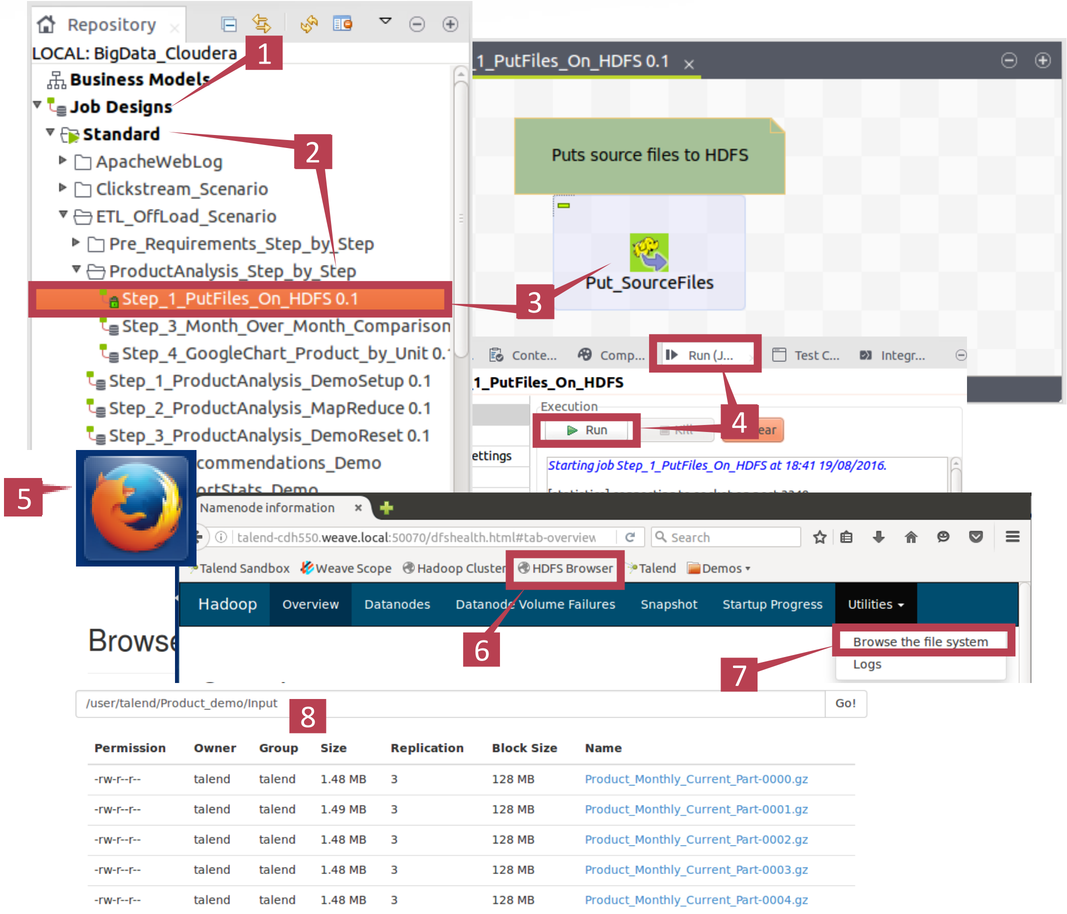
Execute the ETL Off-Load "Step-by-Step" Demo:

With the demo environment setup complete, we can begin examining the ETL Off-load Process.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario > ProductAnalysis_Step_by_Step**
3. Double click on **Step_ 1_PutFiles_on_HDFS 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

When this job is complete, you will have your custom-generated source files on HDFS. To view the files:

5. Open **Firefox**
6. Click on the **HDFS Browser** link on the Bookmarks Toolbar
7. Select **Browse the file system** from the Utilities dropdown.
8. Navigate to */user/talend/ Product_demo/Input*



Starting job Step_1_PutFiles_On_HDFS at 18:41 19/08/2016.

/user/talend/Product_demo/Input

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	talend	talend	1.48 MB	3	128 MB	Product_Monthly_Current_Part-0000.gz
-rw-r--r--	talend	talend	1.49 MB	3	128 MB	Product_Monthly_Current_Part-0001.gz
-rw-r--r--	talend	talend	1.48 MB	3	128 MB	Product_Monthly_Current_Part-0002.gz
-rw-r--r--	talend	talend	1.48 MB	3	128 MB	Product_Monthly_Current_Part-0003.gz
-rw-r--r--	talend	talend	1.48 MB	3	128 MB	Product_Monthly_Current_Part-0004.gz

Talend Big Data Sandbox

Big Data Insights Cookbook

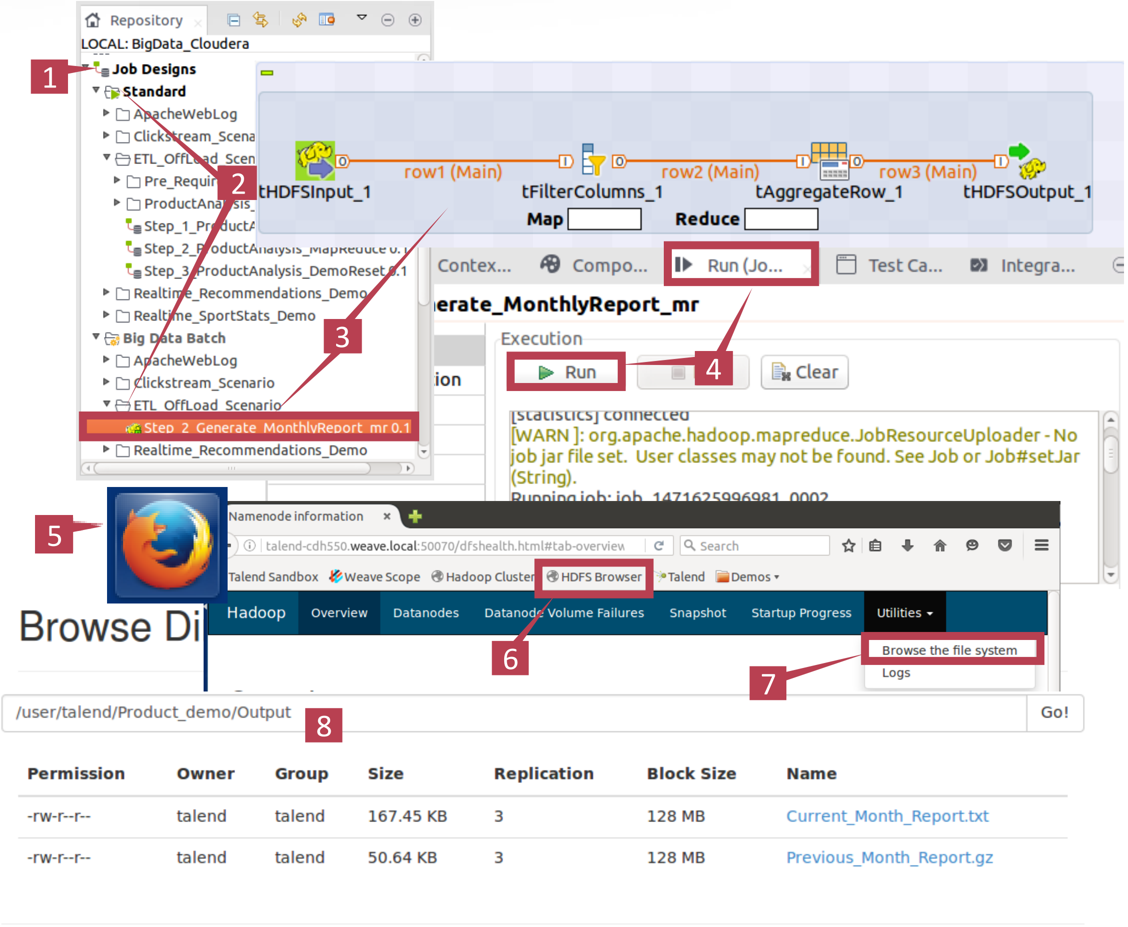
Execute the ETL Off-Load "Step-by-Step" Demo:

Now that your source data is in HDFS, we can use the power of Hadoop and MapReduce to analyze the large dataset.

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Batch > ETL_OffLoad_Scenario**
3. Double click on **Step_2_Generate_MonthlyReport_mr 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

When this job is complete, you can again navigate to the Hadoop file system to view the generated file:

5. Open **Firefox**
6. Click on the **HDFS Browser** link on the Bookmarks Toolbar
7. Select **Browse the file system** from the Utilities dropdown.
8. Navigate to */user/talend/Product_demo/Output*



```
[statistics] connected
[WARN ]: org.apache.hadoop.mapreduce.JobResourceUploader - No job jar file set.  User classes may not be found. See Job or Job#setJar(String).
Running job: job_1471625996981_0002
```

/user/talend/Product_demo/Output

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	talend	talend	167.45 KB	3	128 MB	Current_Month_Report.txt
-rw-r--r--	talend	talend	50.64 KB	3	128 MB	Previous_Month_Report.gz

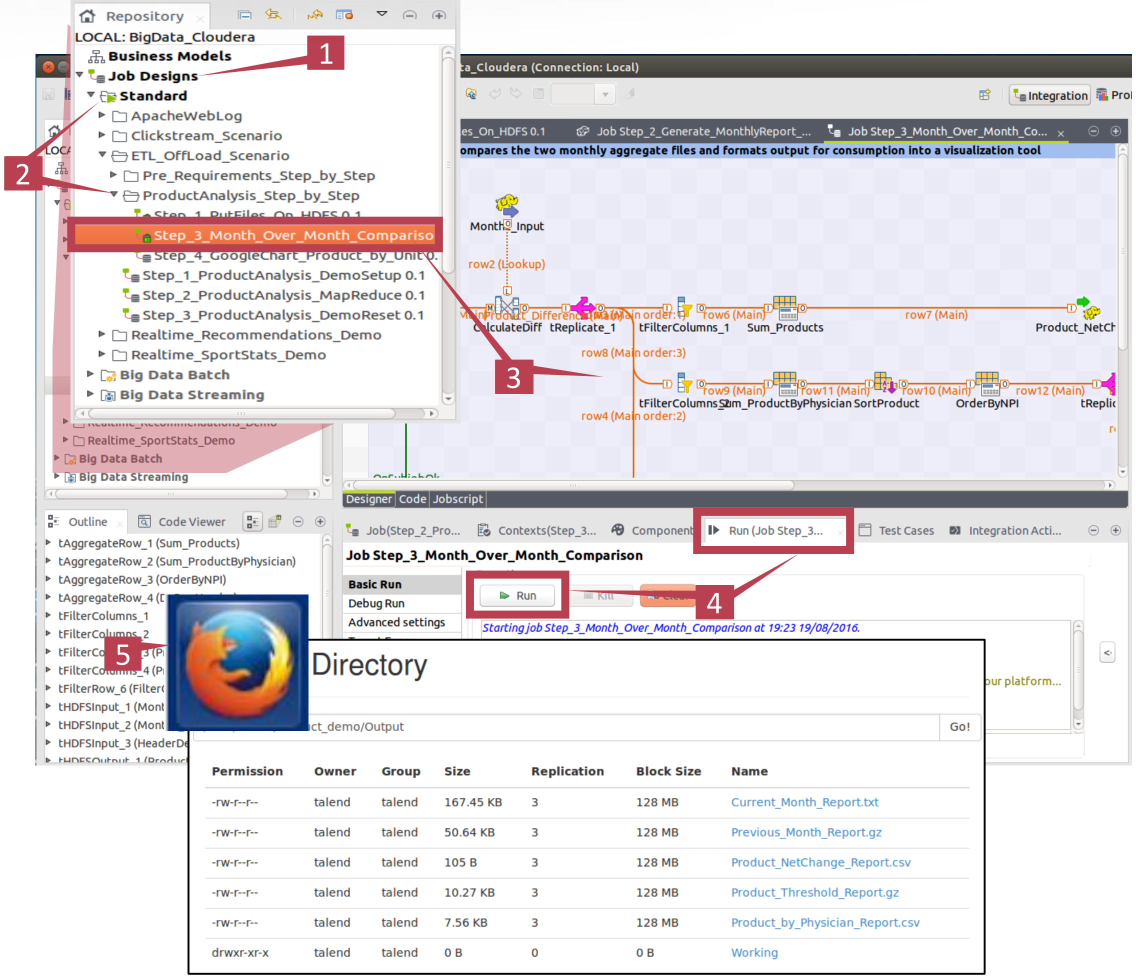
Execute the ETL Off-Load "Step-by-Step" Demo:

With the Previous Month analysis as our baseline, we can now compare our Current Month analysis and track any anomalies.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario > ProductAnalysis_Step_by_Step**
3. Double click on **Step_3_Month_Over_Month_Comparison 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

When this job is complete, you can again navigate to the Hadoop file system to view the generated files:

5. Open **Firefox**
6. Click on the **HDFS Browser** link on the Bookmarks Toolbar
7. Select **Browse the file system** from the Utilities dropdown.
8. Navigate to */user/talend/Product_demo/Output*



Compares the two monthly aggregate files and formats output for consumption into a visualization tool

Job Step_3_Month_Over_Month_Comparison

Starting job Step_3_Month_Over_Month_Comparison at 19:23 19/08/2016.

ct_demo/Output

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	talend	talend	167.45 KB	3	128 MB	Current_Month_Report.txt
-rw-r--r--	talend	talend	50.64 KB	3	128 MB	Previous_Month_Report.gz
-rw-r--r--	talend	talend	105 B	3	128 MB	Product_NetChange_Report.csv
-rw-r--r--	talend	talend	10.27 KB	3	128 MB	Product_Threshold_Report.gz
-rw-r--r--	talend	talend	7.56 KB	3	128 MB	Product_by_Physician_Report.csv
drwxr-xr-x	talend	talend	0 B	0	0 B	Working

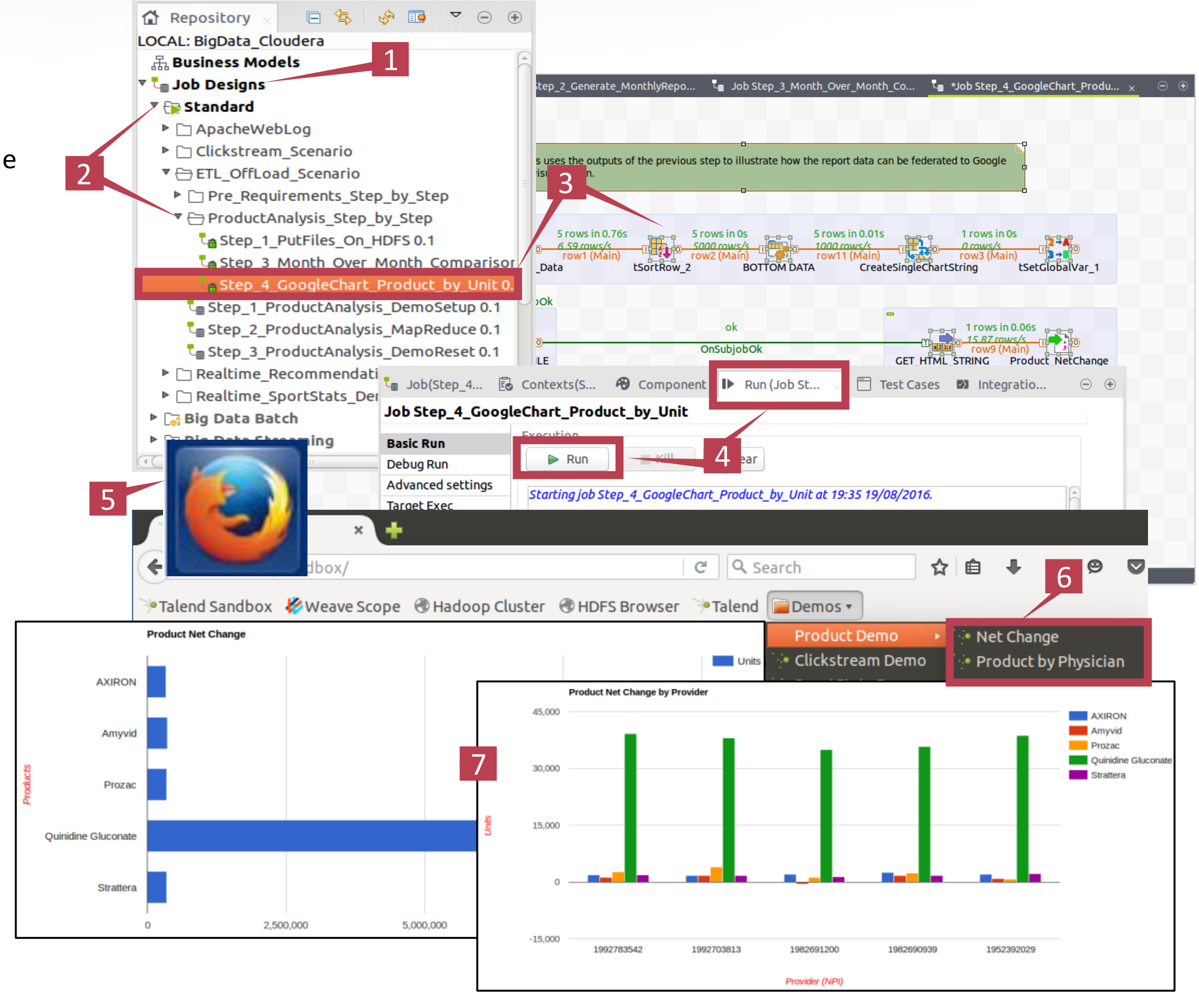
Execute the ETL Off-Load “Step-by-Step” Demo:

The final step is to generate the charts using Google Charts API

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario > ProductAnalysis_Step_by_Step**
3. Double click on **Step_4_GoogleChart_Product_by_Unit 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.

When this job is complete, you can view the generated reports from the web browser:

5. Open **Firefox**
6. From a new tab, Click on **Demos > Product Demo > Net Change** to view the report.
7. Repeat Step 2 above to open the **Product by Physician** Report.

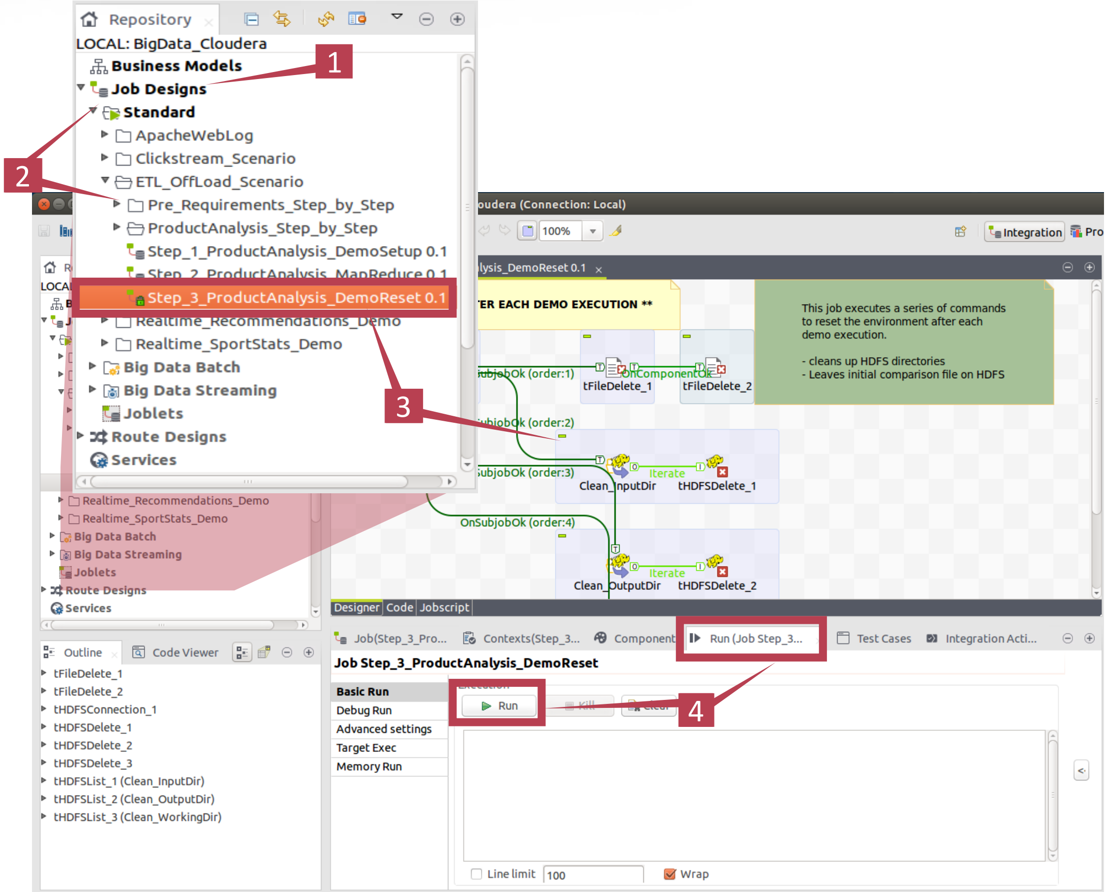


Starting job Step_4_GoogleChart_Product_by_Unit at 19:35 19/08/2016.

Execute the ETL Off-Load "Step-by-Step" Demo:

Reset the demo and run it again! You can run this demo over and over and get different results by changing the Source Files.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ETL_OffLoad_Scenario**
3. Double click on **Step_3_ProductAnalysis_DemoReset 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute.



This job executes a series of commands to reset the environment after each demo execution.

- cleans up HDFS directories
- Leaves initial comparison file on HDFS

Run through this demo again!

One Click

Step-by-step

talend

Talend Big Data Sandbox

Big Data Insights Cookbook

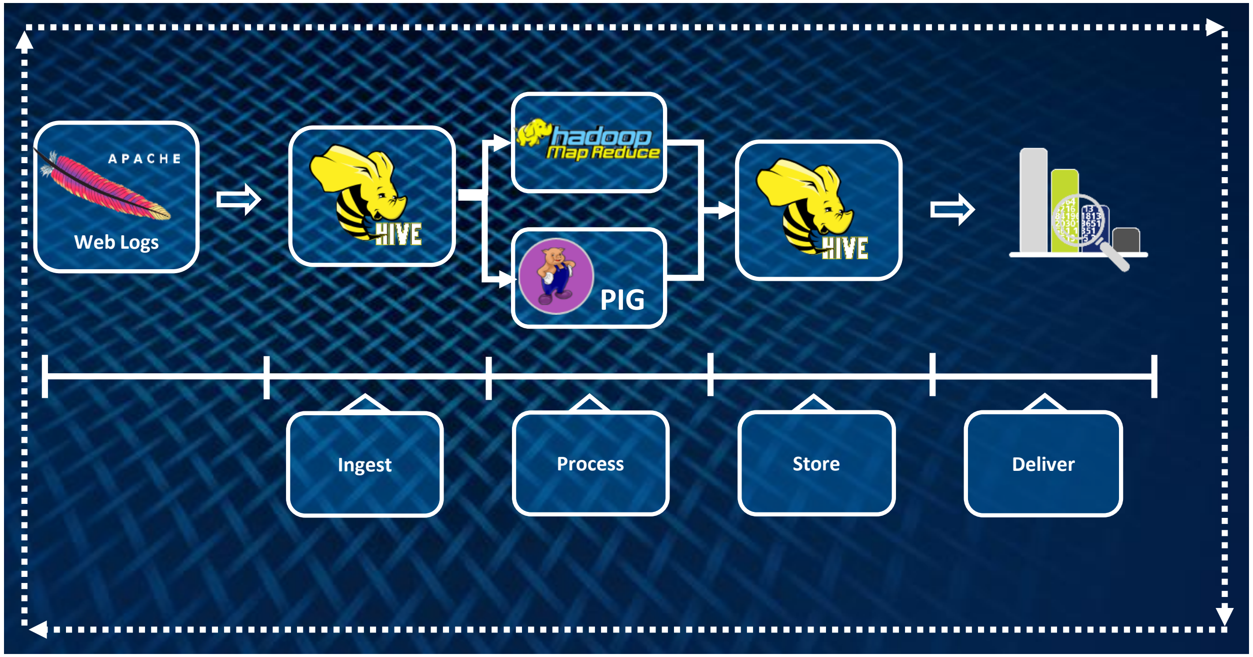
Note: Execution of this demo requires a Hadoop distribution. If a distro hasn't been selected, click here.

Overview:

In this example we demonstrate using different Big Data Methods to aggregate and analyze large volumes of weblog data.

You will experience:

- Using Hive to store and access Data in a Hadoop Distributed File System
- Using standard MapReduce to analyze and count IP addresses in an Apache log file.
- Performing the same Analysis (count of IP addresses in an Apache log file) using Pig.



This Demo will highlight:

Hive Components

Connect, Create, Read and Write with Hive Components to access data in HDFS

Native MapReduce

Use Talend's MapReduce components to access and analyze data, natively, from HDFS

Pig Components

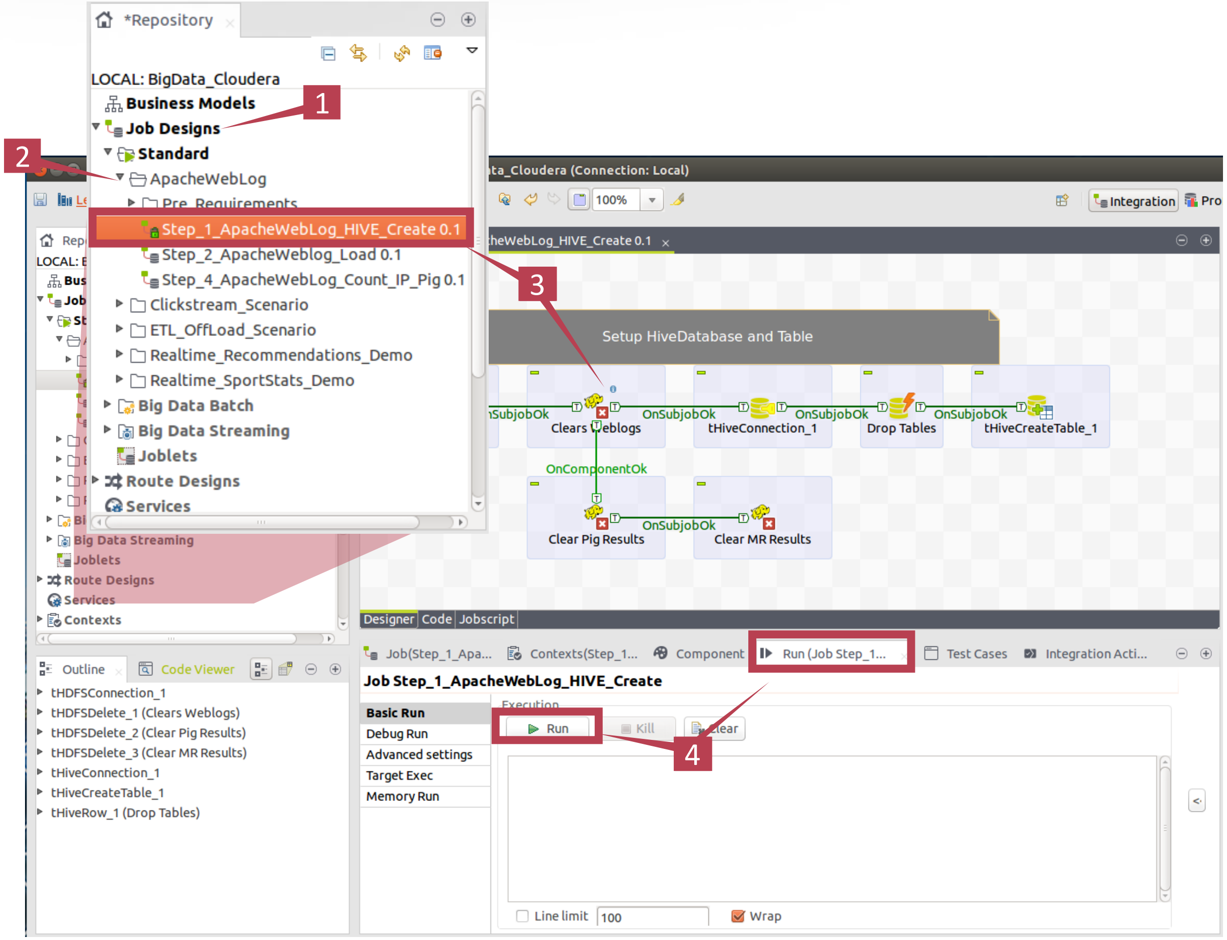
Understand the flexibility of Talend's capabilities to perform the same operations with multiple technologies

Execute the Apache Weblog Demo:

Create the Hive Tables in HDFS and clear out old datasets.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ApacheWebLog**
3. Double click on **Step_1_ApacheWebLog_HIVE_Create 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

When this job completes, old datasets from previous executions will have been cleaned up and a fresh Hive Table will be generated in HDFS.



Talend Big Data Sandbox

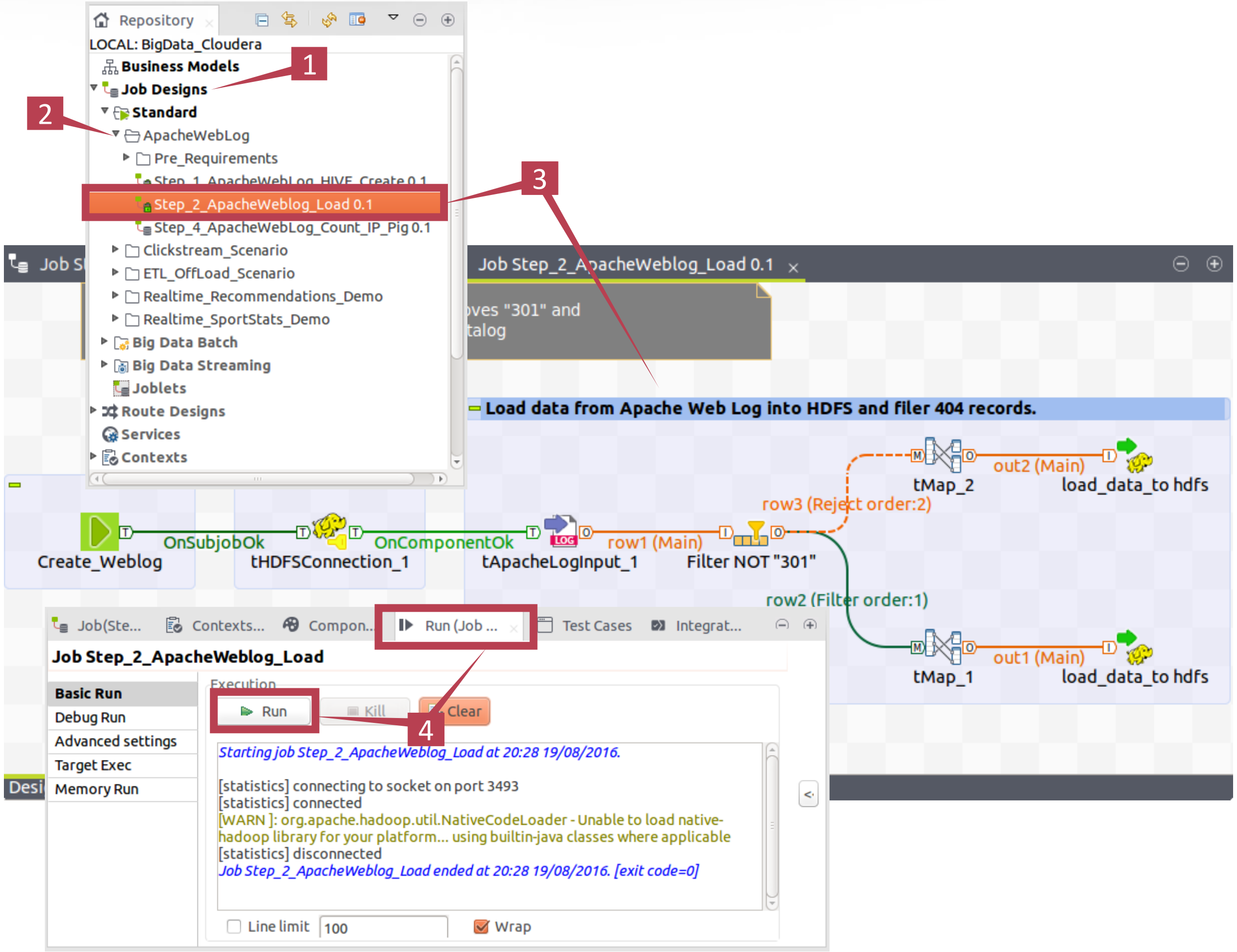
Big Data Insights Cookbook

Execute the Apache Weblog Demo:

Filter the Apache Weblog files and load them to HDFS.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ApacheWebLog**
3. Double click on **Step_2_ApacheWeblog_Load 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

This job filters "301" codes from the Weblog and loads the data to HDFS where it can be viewed by both the HDFS file browser or a Hive Query.

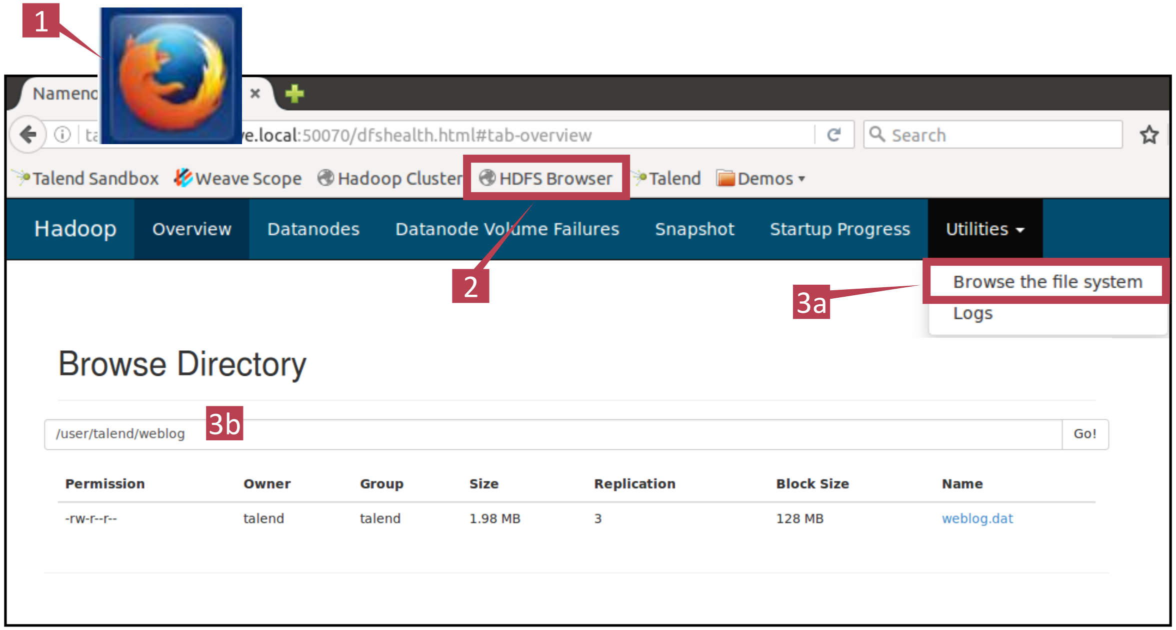


```
Starting job Step_2_ApacheWeblog_Load at 20:28 19/08/2016.

[statistics] connecting to socket on port 3493
[statistics] connected
[WARN ]: org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[statistics] disconnected
Job Step_2_ApacheWeblog_Load ended at 20:28 19/08/2016. [exit code=0]
```

Execute the Apache Weblog Demo:

View the data in HDFS.

1. Open **Firefox**
2. Click on the bookmarked link titled **HDFS Browser**
3. In the **Utilities** Dropdown, select **Browse the File System** and navigate to */user/talend/weblog*



Browse Directory

/user/talend/weblog

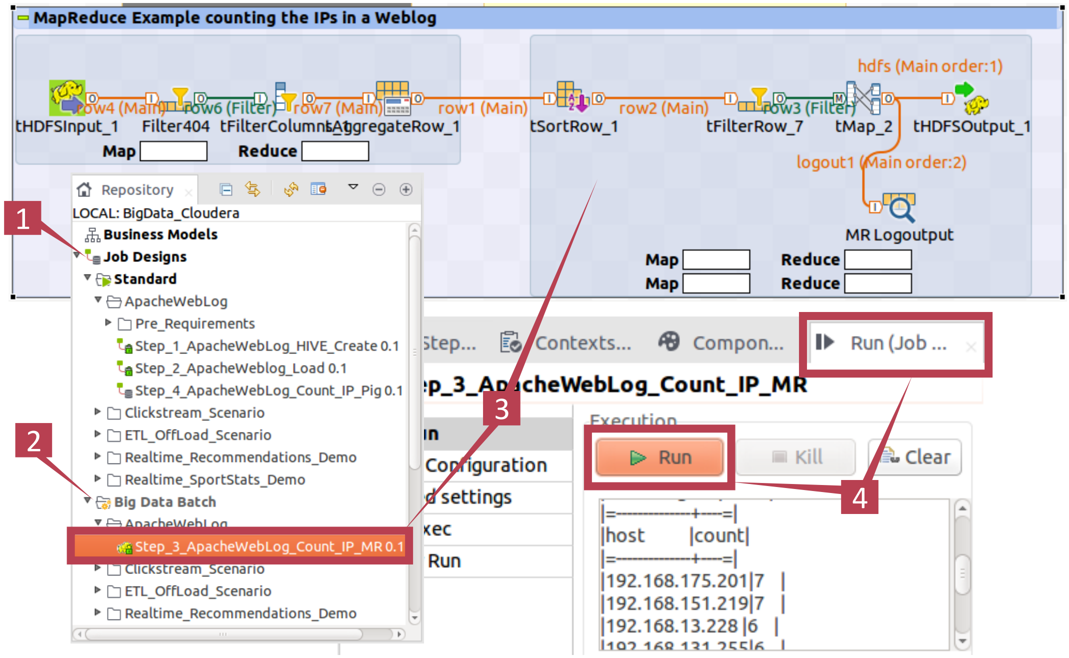
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	talend	talend	1.98 MB	3	128 MB	weblog.dat

Note: While the data is loaded into HDFS, it is also saved in a location where the created Hive table is expecting data. So now you can view the data both through a Hive query or HDFS file browsing.

Execute the Apache Weblog Demo:

Use MapReduce to analyze and calculate distinct IP count.

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Batch > ApacheWebLog**
3. Double click on **Step_3_ApacheWeblog_Count_IP_MR 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute
5. When the job completes, You can view the results that are output to the Job Execution Window.



```
|=-----------+----=|
|host       |count|
|=-----------+----=|
|192.168.175.201|7   |
|192.168.151.219|7   |
|192.168.13.228 |6   |
|192.168.131.255|6   |
```

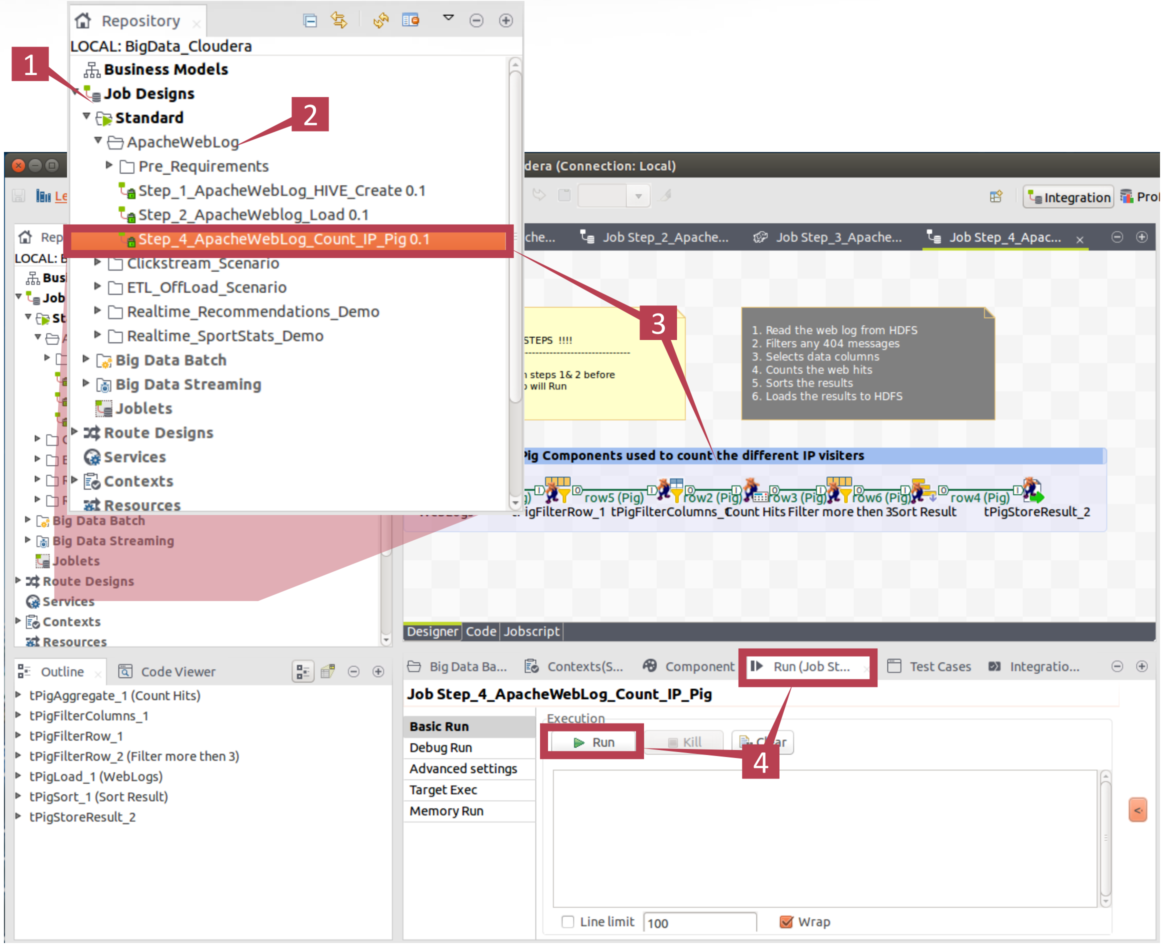
➢ The data from this job is also saved to HDFS. In the HDFS File Browser navigate to */user/talend/weblogMR/mr_apache_ip_out* to see the new files.

Execute the Apache Weblog Demo:

Use MapReduce to analyze and calculate distinct IP count.

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > ApacheWebLog**
3. Double click on **Step_4_ApacheWeblog_Count_IP_Pig 0.1.** This opens the job in the designer window.
4. From the Run tab, click on **Run** to execute

- ➢ The data from this job is also saved to HDFS. In the HDFS File Browser navigate to */user/talend/weblogPIG/apache_ip_cnt* to see the new files.



1. Read the web log from HDFS
2. Filters any 404 messages
3. Selects data columns
4. Counts the web hits
5. Sorts the results
6. Loads the results to HDFS

This Bonus Demo will highlight:

Talend Flexibility

Switch between MapReduce and Spark Frameworks with just a few clicks of the mouse.

Simple Configuration

Once a job has been converted to a new Framework, configuration is quick and simple in Talend Studio.

Future-Proof Technology

Talend's Code-generation Architecture makes it easy to abreast of the latest Technology Trends.

Talend Big Data Sandbox

Big Data Insights Cookbook

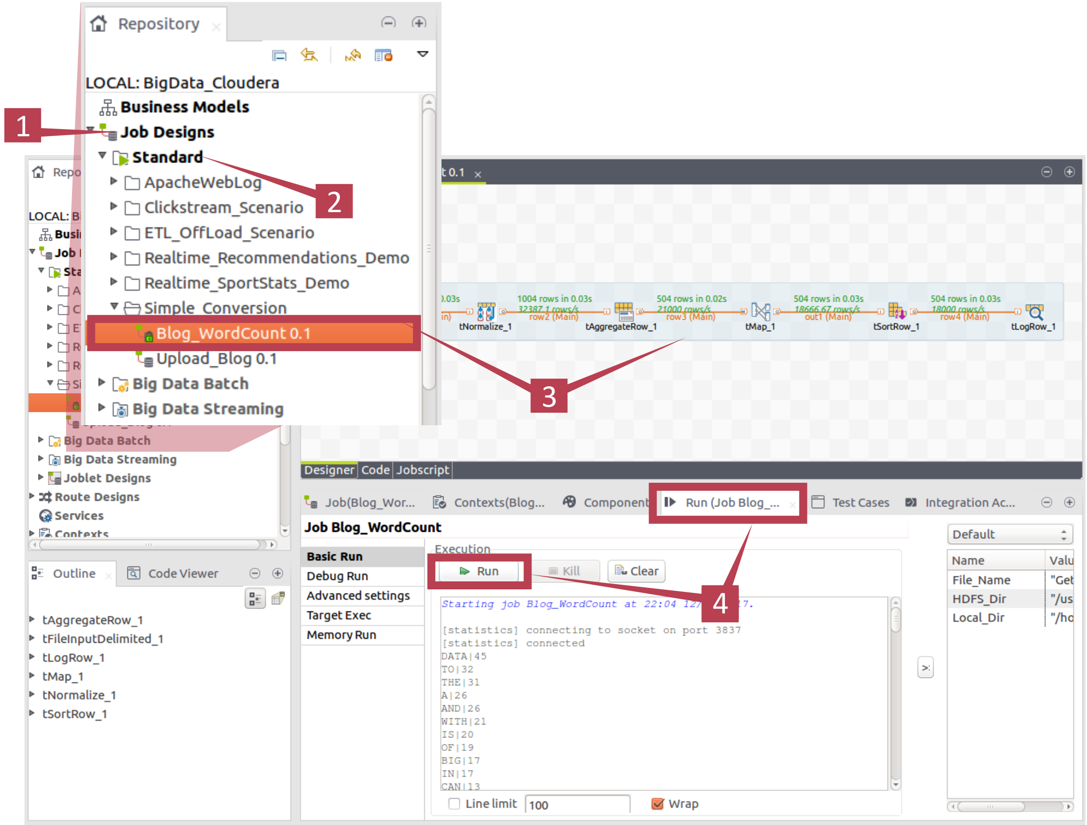
Simple Conversion

This Demo takes a text version of a Talend Blog Post – Getting Started With Big Data – and does a simple word count. The Word Count example is a basic teaching tool for understanding how Map Reduce Technology works.

Execute the Simple Conversion Demo:

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Simple_Conversion**
3. Double click on **Blog_WordCount 0.1.** This opens the job in the designer window.
4. From the **Run** tab, click on **Run** to execute

The output appears in the execution window, displaying the number of occurrences of each word within the blog post.



```
Starting job Blog_WordCount at 22:04 12/ 7.

[statistics] connecting to socket on port 3837
[statistics] connected
DATA|45
TO|32
THE|31
A|26
AND|26
WITH|21
IS|20
OF|19
BIG|17
IN|17
CAN|13
```

Name	Valu
File_Name	"Get
HDFS_Dir	"/us
Local_Dir	"/ho

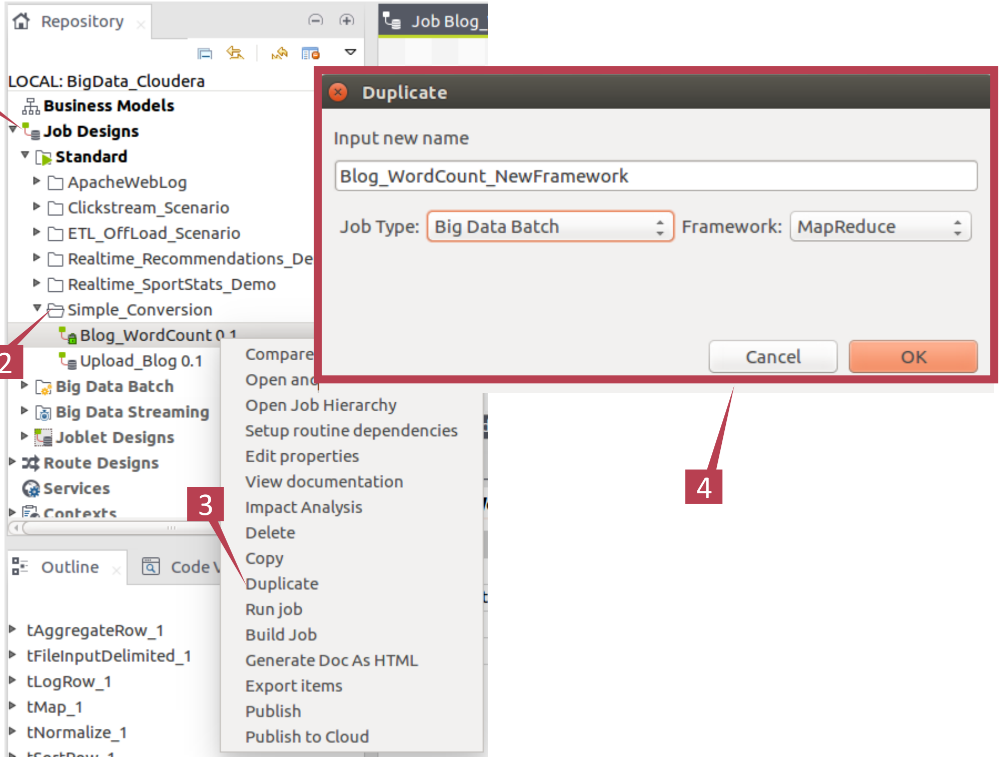
- ➢ Now lets see how quickly we can convert this job to a Map Reduce Job and then a Spark Job.

Converting to a Map Reduce Batch Job:

1. Navigate to the **Job Designs** folder.
2. Click on **Standard > Simple_Conversion**
3. Right-click on **Blog_WordCount 0.1.** From the drop-down menu, select **Duplicate**.
4. Rename the job **Blog_WordCount_NewFramework** then in the Job Type dropdown, choose **Big Data Batch**. Finally, in the Framework dropdown, choose **MapReduce** and click **OK**.

   Just that quickly, the standard job has been converted to a Big Data Batch job using the MapReduce Framework

➢ Before you can run the job in MapReduce or Spark, you need to execute the **Upload_Blog** job in **Job Designs > Standard > Simple_Conversion.** Right-click on the job and select **Run Job** from the dropdown window.



Duplicate

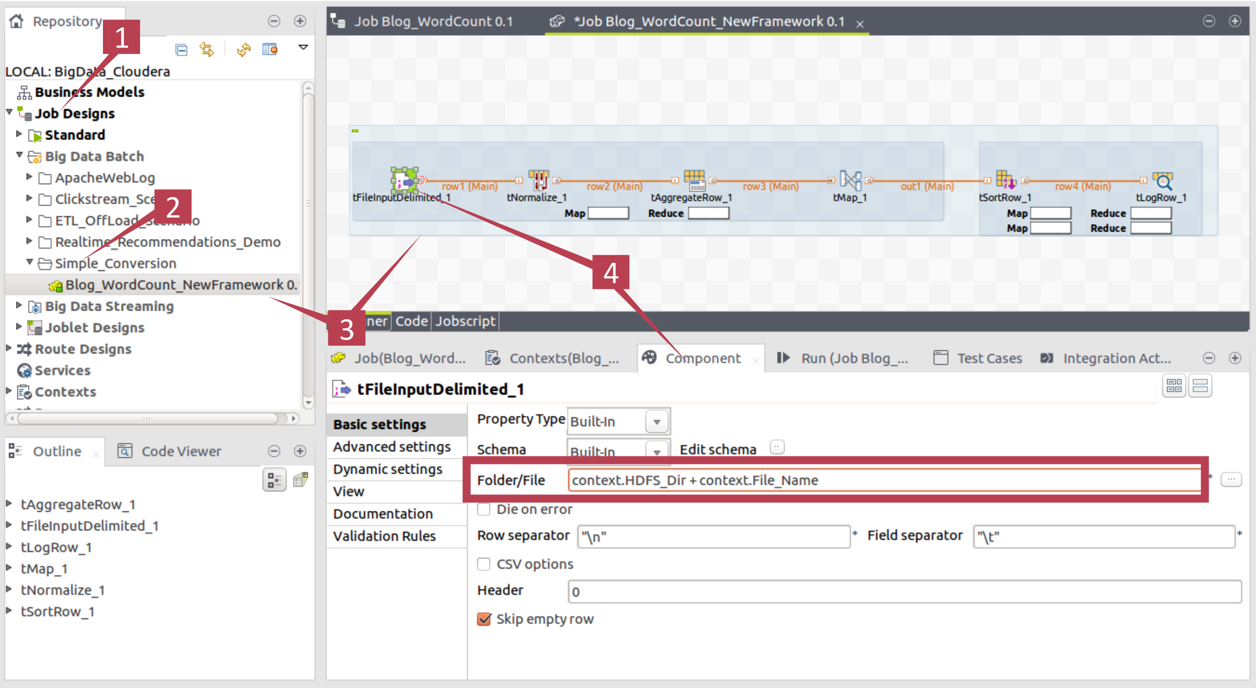
Input new name: Blog_WordCount_NewFramework

Job Type: Big Data Batch

Framework: MapReduce

Configure the new job and Execute:

1. Navigate to the **Job Designs** folder.
2. Click on **Big Data Batch > Simple_Conversion**
3. Your newly converted job exists here. Double click on **Blog_WordCount_NewFramework 0.1** to open in the Design Window.
4. Next, In the Designer Window, click on the **tFileInputDelimited** component and then click on the **Component Tab**. Here, change the **Folder/File** property to *context.**HDFS**_Dir + context.File_Name*



tFileInputDelimited_1

Basic settings | Advanced settings | Dynamic settings | View | Documentation | Validation Rules

Property Type: Built-In

Schema: Built-In   Edit schema

Folder/File: context.HDFS_Dir + context.File_Name

□ Die on error

Row separator: "\n"   Field separator: "\t"

□ CSV options

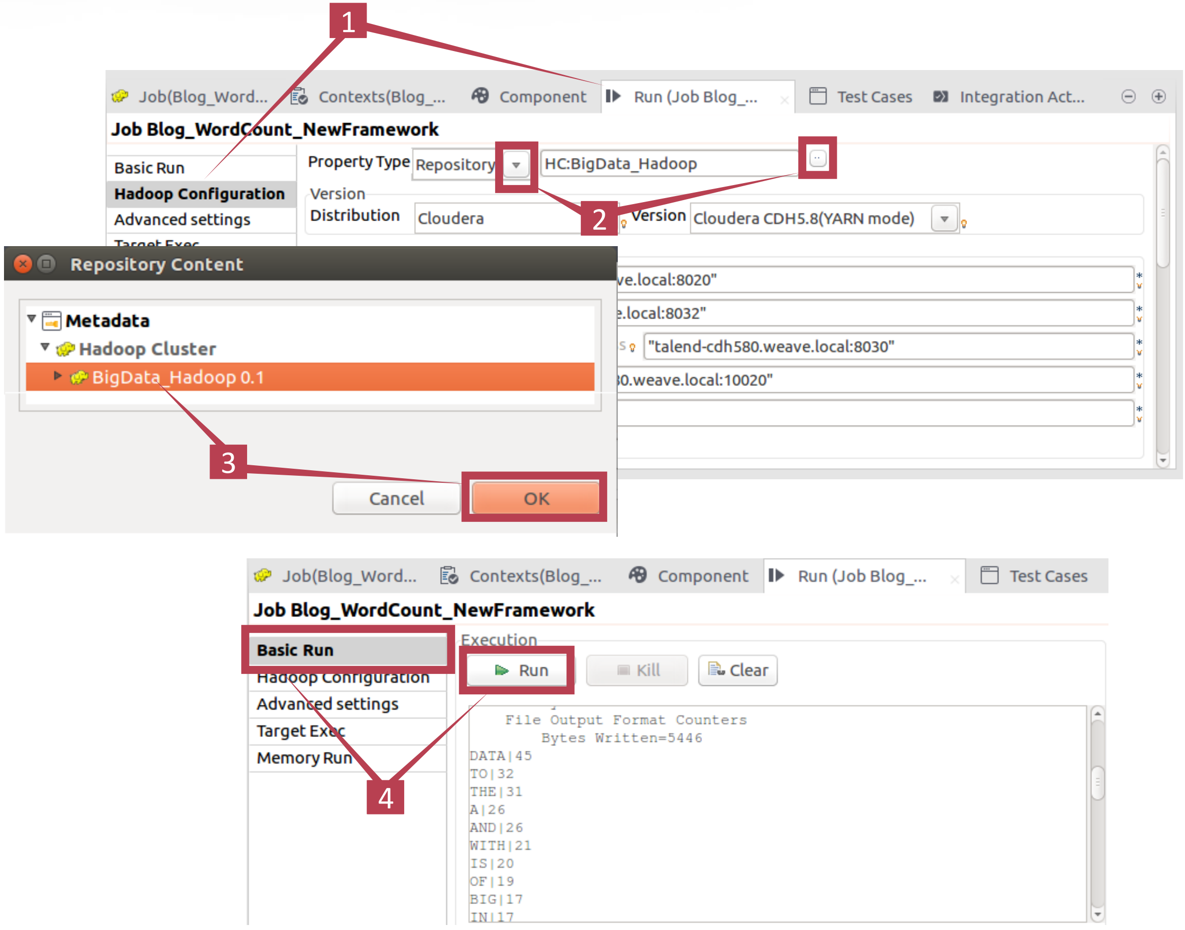
Header: 0

☑ Skip empty row

Simple Conversion

Configure the new job and Execute:

1. Now click on the **Run** tab and choose **Hadoop Configuration**.
2. Set the Property Type to **Repository** then click the **ellipsis** to pull up the Repository Content.
3. Click on the Hadoop Cluster Dropdown and then select **BigData_Hadoop** and click **OK**. This will import the preset Hadoop configuration into the job configuration.
4. Now Click on **Basic Run** and then **Run**. The job now executes on the Hadoop Cluster as a MapReduce Job.



Job Blog_WordCount_NewFramework

Property Type: Repository — HC:BigData_Hadoop

Version — Distribution: Cloudera — Version: Cloudera CDH5.8(YARN mode)

...ve.local:8020"
...e.local:8032"
"talend-cdh580.weave.local:8030"
...0.weave.local:10020"

Metadata
- Hadoop Cluster
  - BigData_Hadoop 0.1

```
    File Output Format Counters
        Bytes Written=5446
DATA|45
TO|32
THE|31
A|26
AND|26
WITH|21
IS|20
OF|19
BIG|17
IN|17
```

Simple Conversion

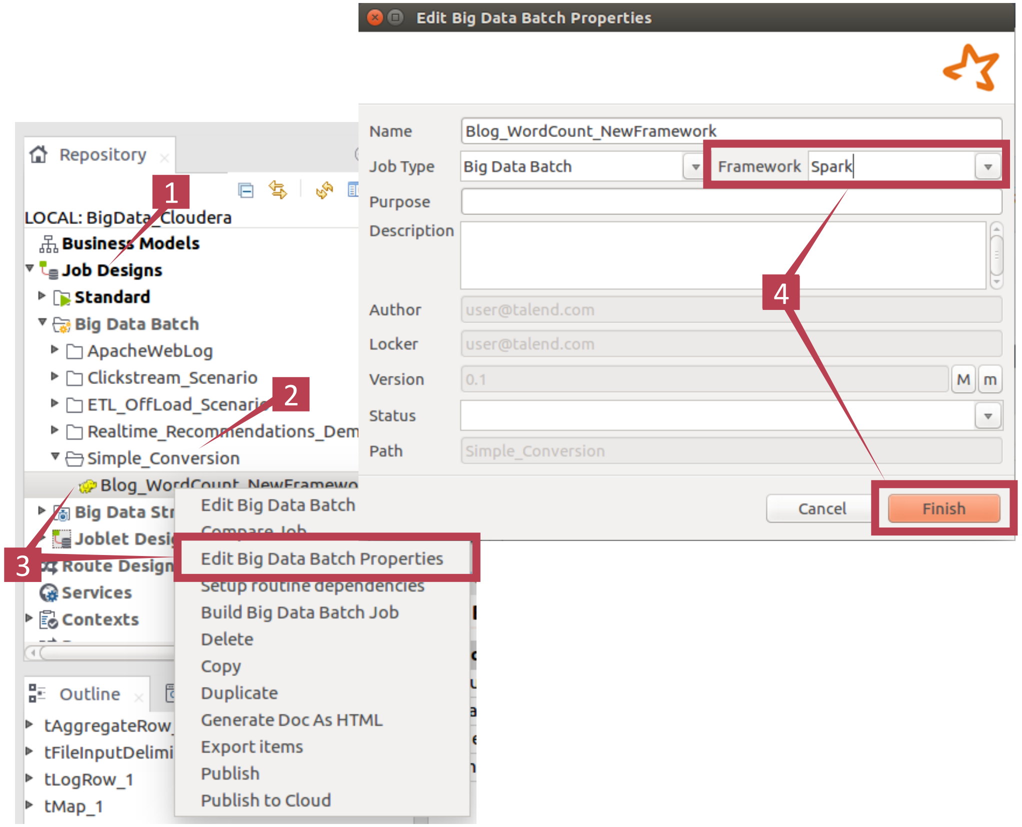
➢ We can now take this same MapReduce job and convert it to execute on the Spark Framework.

Convert to Spark Job:

1. In the **Job Designs** folder.
2. Click on **Big Data Batch > Simple_Conversion**
3. Rather than duplicating the job again, this time we are going to Right-click on **Blog_WordCount_NewFramework 0.1** and select **Edit Big Data Batch Properties** from the dropdown menu.
4. In the Popup window, all you need to do is choose **Spark** from the Framework dropdown. Now click **OK**.

Just that quickly, the MapReduce job has now been converted to run on the Spark Framework.

No further configuration needs to be done, you can open the job, click on the Run Tab and execute the job on Spark!



Edit Big Data Batch Properties

Name: Blog_WordCount_NewFramework

Job Type: Big Data Batch

Framework: Spark

Purpose:

Description:

Author: user@talend.com

Locker: user@talend.com

Version: 0.1

Status:

Path: Simple_Conversion

Conclusion

Simplify Big Data Integration

Talend vastly simplifies big data integration, allowing you to leverage in-house resources to use Talend's rich graphical tools that generate big data code (Spark, MapReduce, PIG, Java) for you.

Talend is based on standards such as Eclipse, Java, and SQL, and is backed by a large collaborative community.

So you can up skill existing resources instead of finding new resources.

Built for Batch and Real-time Big Data

Talend is built for batch and real-time big data. Unlike other solutions that “map” to big data or support a few components, Talend is the first data integration platform built on Spark with over 100 Spark components.

Whether integrating batch (MapReduce, Spark), streaming (Spark), NoSQL, or in real-time, Talend provides a single tool for all your integration needs.

Talend’s native Hadoop data quality solution delivers clean and consistent data at infinite scale.

Lower Operating Costs

Talend lowers operations costs.

Talend’s zero footprint solution takes the complexity out of…

✓Integration Deployment
✓Management
✓Maintenance

A usage based subscription model provides a fast return on investment without large upfront costs.