Data Lake Quick Start from Cognizant and Talend

Out of the Box Data Lake

User Guide

October 2017

Cognizant Technology Solutions

Contents

1. Purpose

The Data Lake Quick Start from Cognizant and Talend illustrates Big Data best practices with sample Talend jobs running the Talend Quickstart for Data Lake. The jobs have been developed by Cognizant for integrating Spark, RedShift, Hadoop and S3 technologies into a Data Lake implementation. This Guide provides an overview of the application architecture, the demo workflow, and the sample jobs realizing the workflow.

1.1. About Out of Box Data Lake Quick Start

Data Lakes in the Cloud are a key driver of Digital Transformation initiatives. They enable data and operational agility by enabling access to historical and real-time data for analytics. Cognizant in partnership with AWS and Talend brings together a solution that enables customers to build and deploy a Data Lake on AWS in 50% less time.

The Quickstart applies DevOps principles to accelerate the process of building and deploying a Data Lake solution in AWS. This solution leverages AWS Cloud Formation templates to provision required resources, services and data lake integration components including S3, Talend Big Data suite, EMR, Redshift. Please refer to the Out of Box Data Lake Deployment Guide for details regarding how to automatically provision your AWS Big Data stack along with the supporting Talend Infrastructure.

1.2. Document Structure

Section 2 briefly covers the logical Talend architecture including the different Talend servers and how they interact.

Section 3 reviews the demo workflow and the sample Cognizant jobs used to realize the workflow. The sample jobs are intended to illustrate a range of different Talend connectors and components. There are multiple ways of achieving the desired integration.

Section 4 summarizes the parameters used by Talend Jobs.

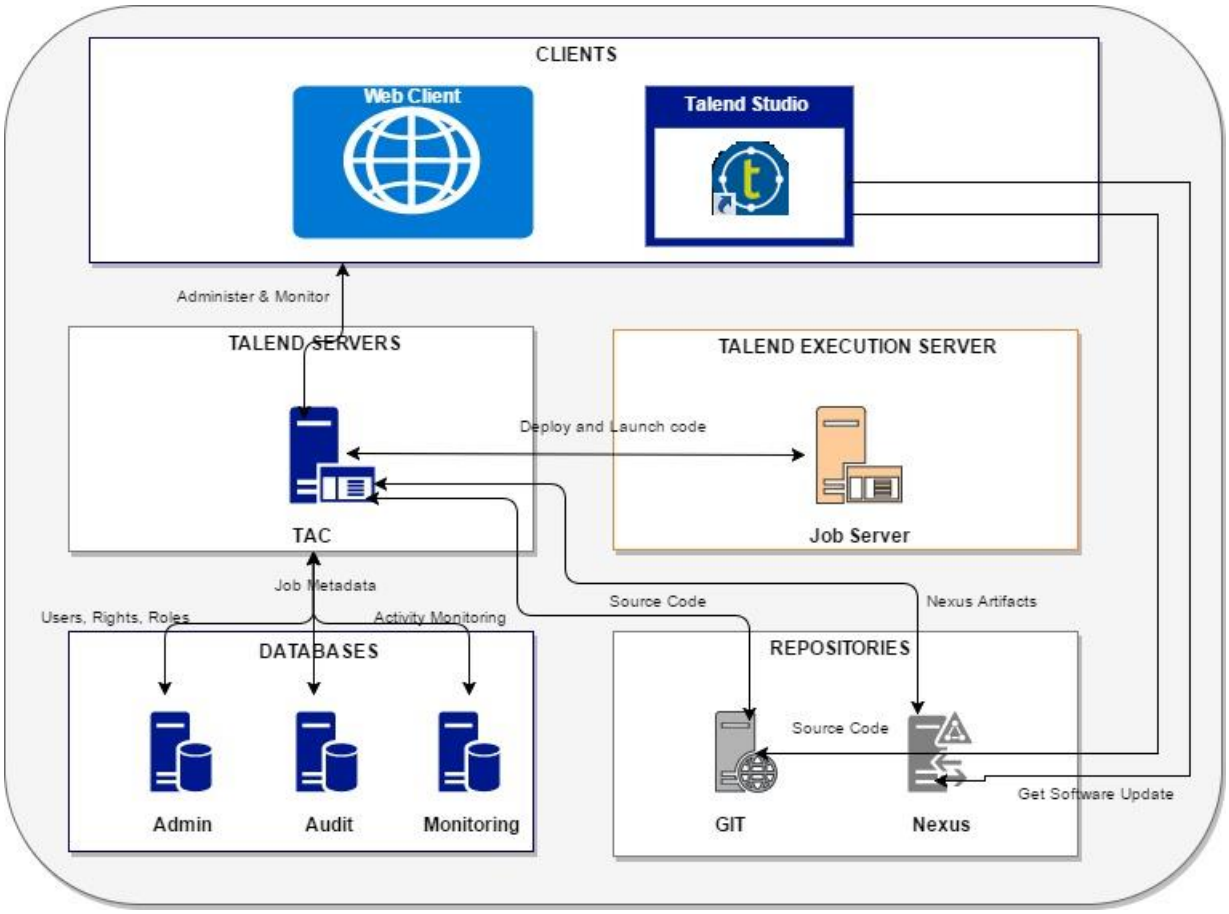
Section 5 shows you how to prepare Talend Studio. The Cloud Formation scripts provision EMR, Redshift, RDS, and S3 in AWS. The scripts also provide a Talend Studio environment on a remote Linux desktop in AWS. You can use Talend Studio on the remote desktop, or set it up on your local laptop.

Section 6 provides a step-by-step walk-through of running the sample jobs in the Quickstart environment.

2. Talend Architecture

The following diagram shows the key Talend software components:

- Talend Studio
- Talend Administration Center
- Talend JobServer
- Talend Log Server
- Talend Artifact Repository (Nexus)
- Git Projects



Talend Software Components

Please refer to the Talend Big Data Reference Architecture document on help.talend.com for more information. Descriptions of each solution component is also available in the regular Talend Reference Architecture. They are summarized below for convenience.

2.1. Talend Studio

The Talend Studio is used to visually design and build data integration jobs. The Talend Studio allows developers to focus on data-centric tasks such as Integration, Profiling, validation, transformations, and lookups.
The studio is based on Eclipse 4.4 RCP (Rich Client Platform). Only Eclipse plugins allowed by the Talend license can be used within the Talend Studio. All features are license activated.

2.2. Talend Administration Center

The Talend Administration Center (TAC) is a web application hosted on Tomcat. In general, only one Talend Administration Center is needed per Talend environment.
The Talend Administration Center maintains administration Metadata including users, projects, authorizations, job schedules, configuration, and runtime history within the database TAC database.

2.3. Talend JobServer

The Talend JobServer is a lightweight agent used for execution and monitoring of Talend tasks deployed through the TAC Job Conductor. It can also be used by Talend Studio users through the Distant Run function.

The Talend JobServer is a server component that runs as a service. There are no license restrictions on the number of JobServers that a customer can install. The Talend JobServer also monitors the server health (CPU, RAM, and Disk Usage).

2.4. Talend Log Server

The Talend Log Server is based on ElasticSearch, LogStash, and Kibana. It is used to streamline the capture and storage of logs from the Talend Administration Center, and Jobs scheduled via the Job Conductor.

2.5. Git

Git is used for source control management (SCM) of Talend Projects. It allows you to take full advantage of features including versioning, branching, tagging, and cloning repositories.

2.6. Artifact Repository

The Artifact Repository is Nexus OSS bundled with the Talend product. It is used for the following:

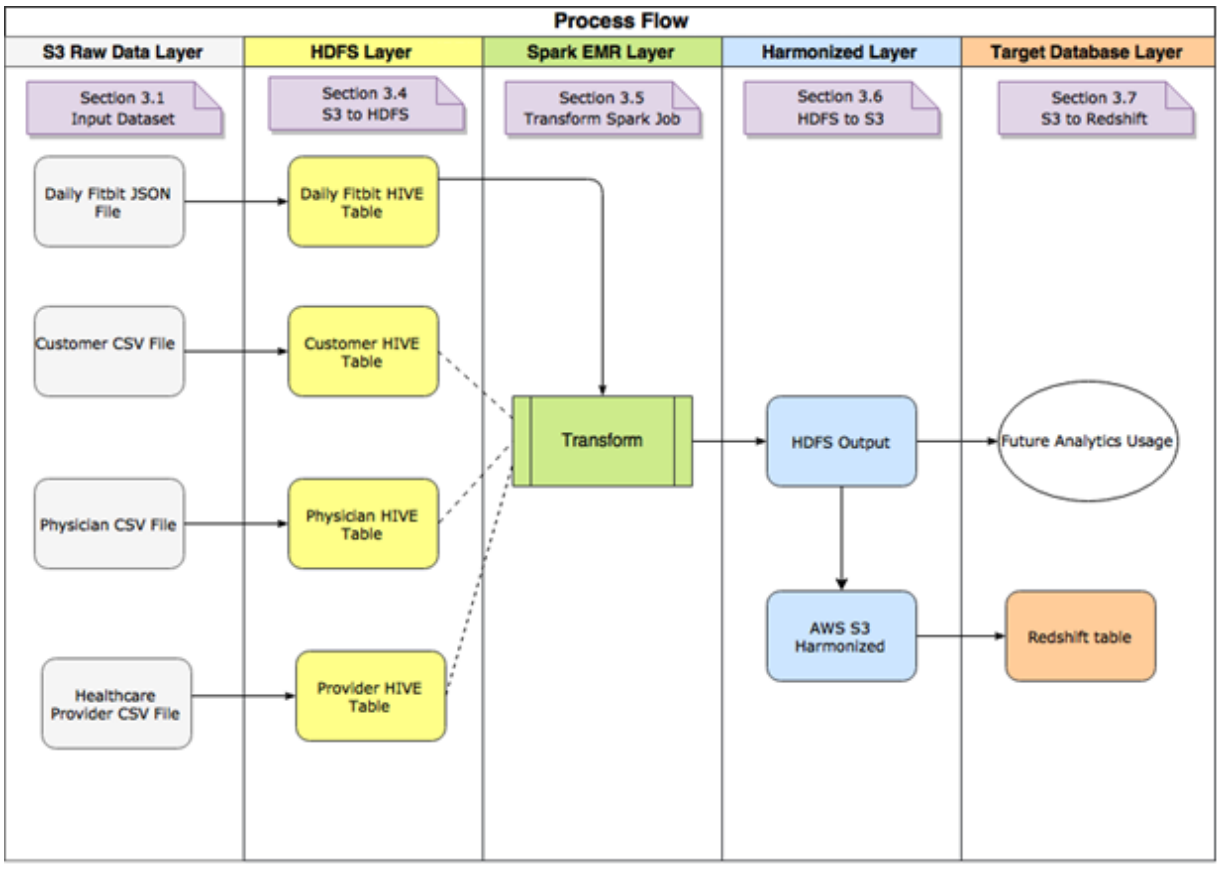
- Receive and store patches from Talend Website for deployment
- Store published artifacts (Jobs, Services, Routes) from the Talend Studio
- Store third party libraries needed by the Talend Studio and Command Line

3. Talend Demo Workflow

The demo job demonstrates the end to end data lake flow from Data Ingestion through Transformation and Loading using Talend. The sample project *'oodle_demo'* includes native Spark on EMR jobs leveraging built for a specific customer fitness tracker use case.

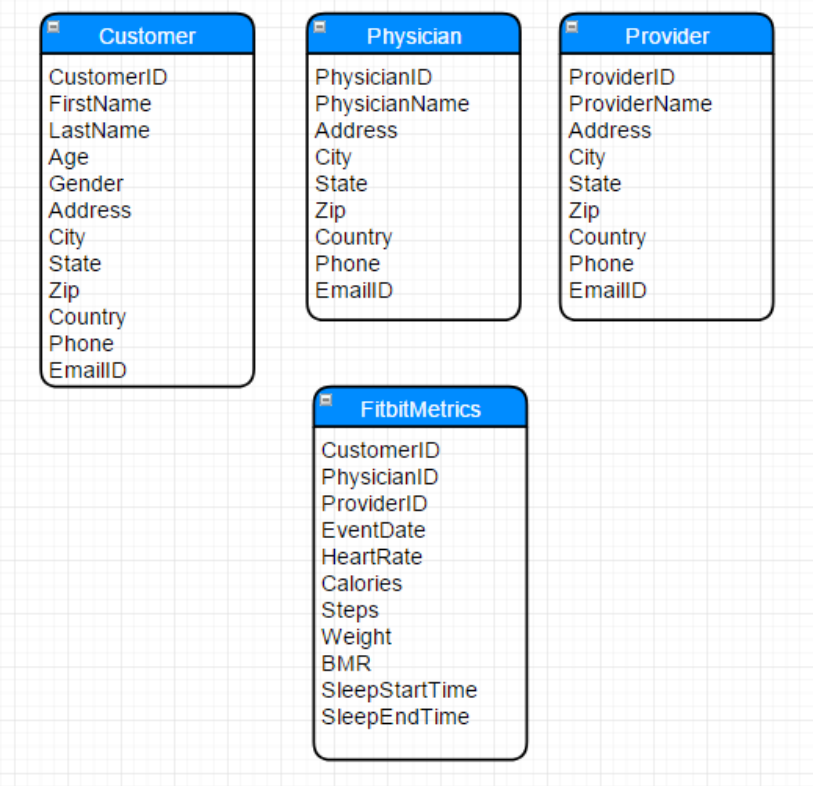
The data flow is as follows:

1	Data Ingestion: Loading CSV/JSON data from S3 to HDFS / Hive
2	Data Processing: Transformation and aggregation using various Talend's spark and Hadoop features
3	Data Repository: Load and build warehouse using Redshift



3.1. Input Dataset

The jobs process 4 datasets – Customer, Physician, Provider and Fitbit daily feed Metrics. These datasets are sourced from an AWS S3 bucket set up by the Cloud Formation scripts.



Fitbit Daily Feed – This contains information about Heartrate, Calories spent, BMR, Weight, Steps, Sleep time, etc for customers in JSON format

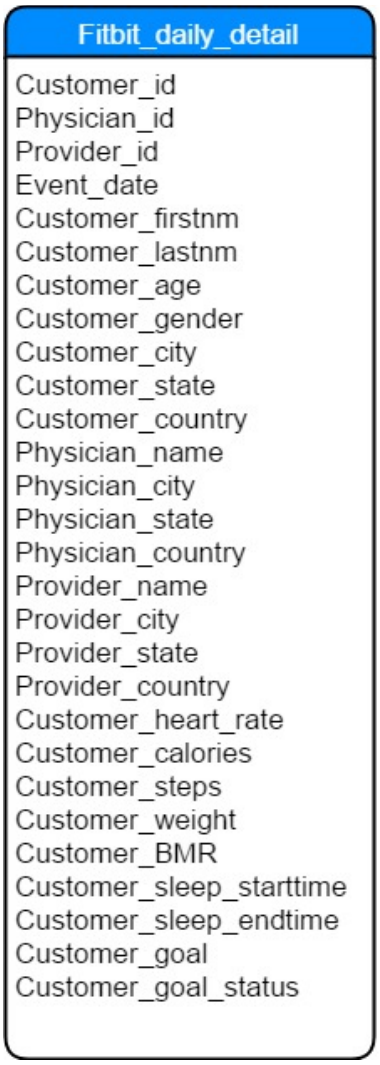
Customer–Customer related information such as Customer_id, Name, Age, Contact and Demographic details in CSV format

Physician– Physician related information with Physician_id, Name, Contact and demographic details in CSV format

Provider– Healthcare Providers such as Provider_id, Provider name, office contact details in CSV format

3.2. Output Dataset

Aggregated fitness data is output to Redshift



3.3. Talend Job

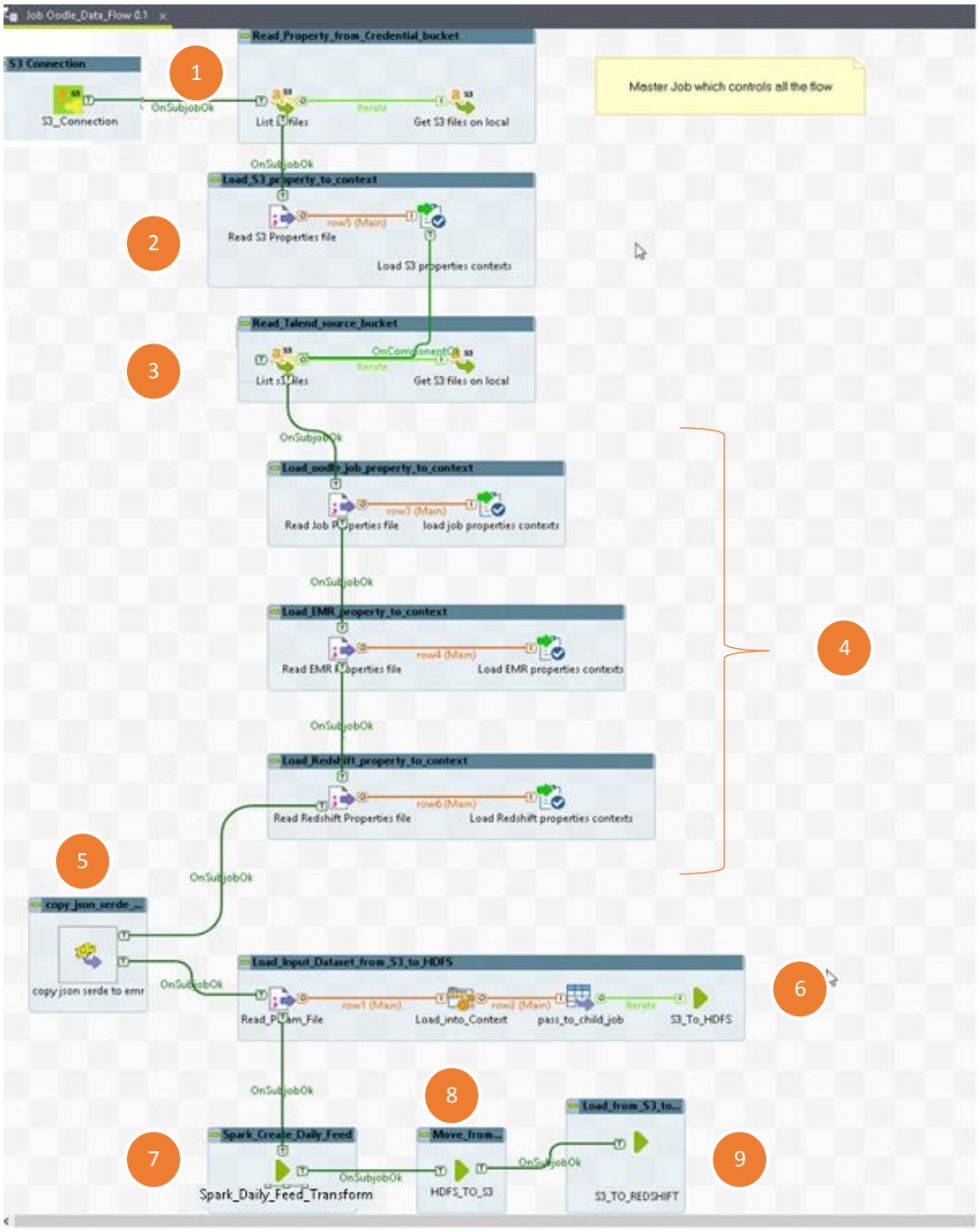


Figure 1 Parent Job Orchestrating Child Jobs

1	Copy the property files having metadata about S3, EMR and Redshift from the Credentials S3 bucket to local directory using ***tS3Get*** component. tS3Get uses the access key and secret key set in the context variables to access the *Credentials* S3 bucket.
2	Load the *TalendSource* and *TalendTarget* S3 bucket names into context variable using ***tContextLoad*** component.
3	Copy the input files and job specific parameter files and jars from the *TalendSource* S3 bucket to Talend Job server using ***tS3Get*** component. tS3Get uses the access key and secret key to access the *TalendSource* S3 bucket.
4	Load EMR, Redshift and job parameters into context variables using ***tContextLoad*** component.
5	Copy dependent libraries to EMR (json-serde-1.3.7-jar-with-dependencies.jar) ***tHDFSPut*** component
6	Load Input data set from *TalendSourcebucket* to HDFS using ts3get and ***tHDFSPut*** component. For More details please refer section 3.4
7	Perform join and transform data using Talend native Spark framework and load the data into HDFS. For more details please refer Section 3.5
8	Use Standard Talend job to copy the load ready files from HDFS to S3 *TalendTarget* bucket.
9	Load data from S3 to redshift using ***tRedshiftBulkExec*** component. For more details please refer section 3.6

3.4. S3 to HDFS

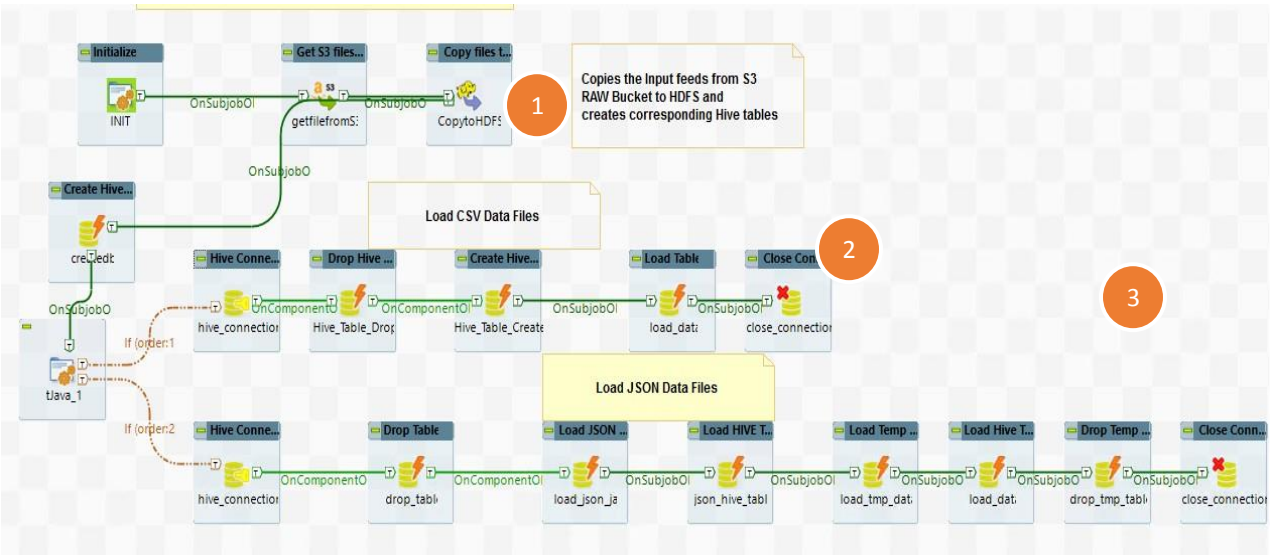


Figure 2 S3 to HDFS child job (step #4 in parent job)

1	Push the input data set which are in CSV and JSON format from S3Sourcebucket to HDFS using ***tS3Get*** and ***tHFDSPut*** component.
2	Create HIVE table and load the input CSV data set into HIVE using ***tHiveRow*** component
3	Create HIVE table and load the input JSON data set into HIVE using ***tHiveRow*** component

3.5.Spark_Daily_Feed_Transform Job

This Spark job does the lookups between the Input extracts, calculates reporting metrics, and creates the Harmonized extract on HDFS. The whole process runs on Amazon EMR cluster.

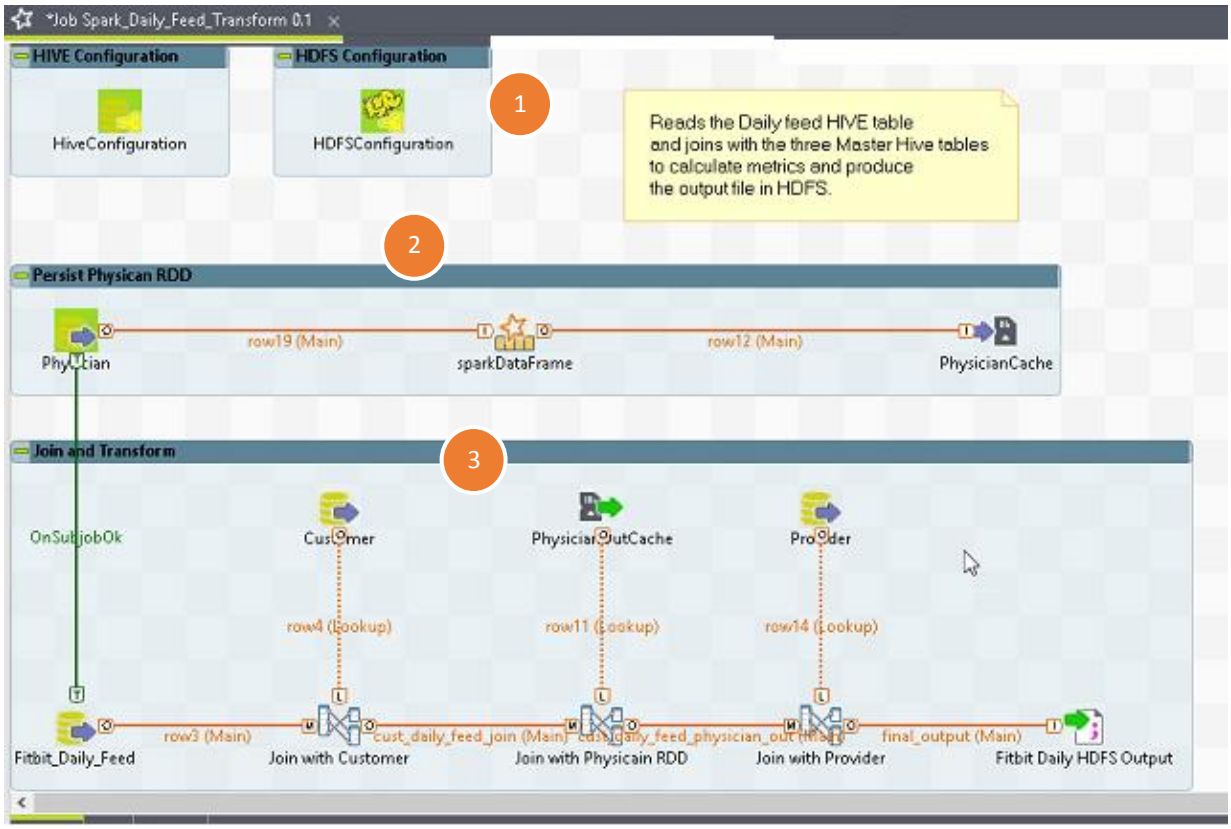


Figure 3: Spark child job (step #7 in parent job)

1	Set up HIVE and HDFS connection using ***tHIVEconfiguration*** and ***tHDFSconfiguration***
2	Create dataframe from Physician HIVE Table and persist RDD using ***tSQLRow*** and ***tCacheout*** component
3	Join Fitbit HIVE table and look up with Customer, Physician RDD and Provider HIVE tables using ***tMap*** component. Include transformation logic for derived attributes in the tMap component. Write output file into HDFS using ***tHDFSOutput*** component.

3.6. HDFS to S3 Job

This standard job copies the final output file from HDFS to S3 Harmonized layer. The Access Key and Secret Access Key details are passed from context variables, and not specified in the S3 Component.

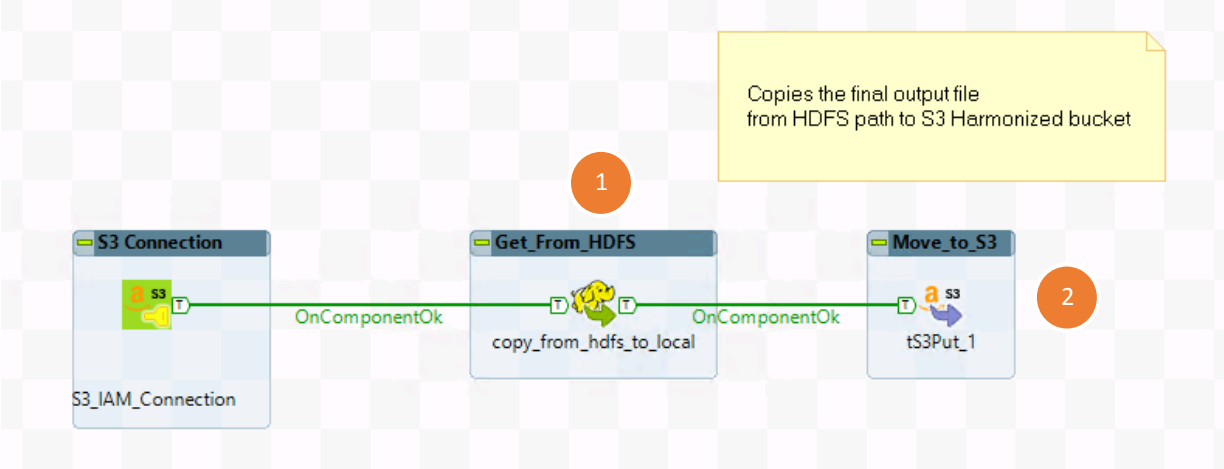


Figure 4: HDFS to S3 child job (step #8 in parent job)

1	Use ***tHDFSGet*** to copy the file from EMR to the job server directory.
2	Copy the file from job server to S3 Harmonized bucket (TalendTarget bucket) using ***tS3Put*** component.

3.7. S3 to Redshift Job

This final standard job loads the transformed output file from AWS *TalendTarget* bucket to Redshift table that will be further used for reporting purpose.

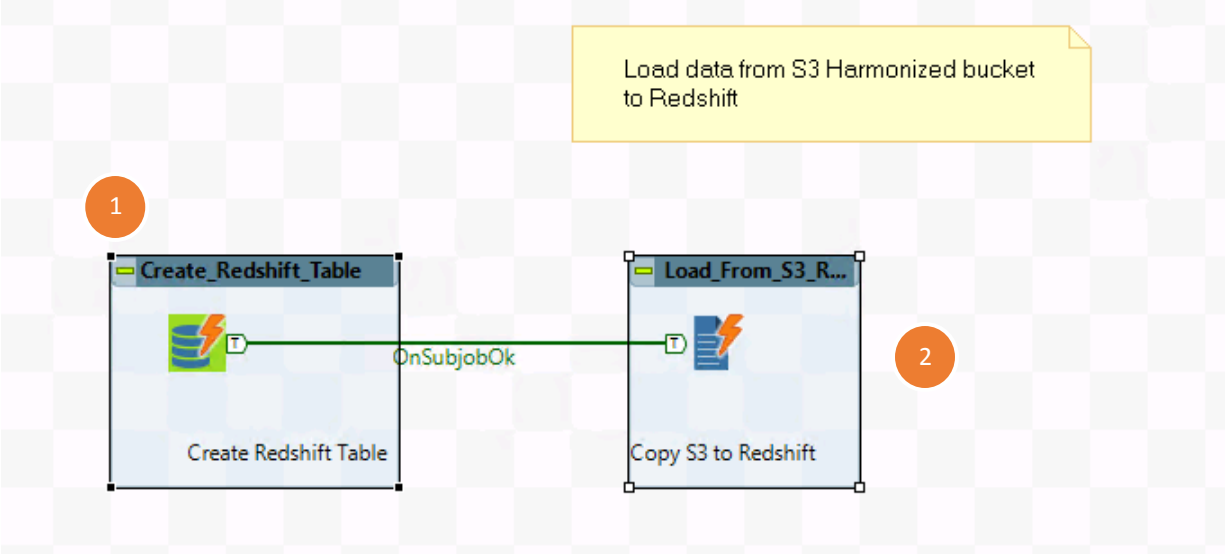


Figure 5: S3 to Redshift child job (step #9 in parent job)

1	Creates the **Fitbit_daily_tgt_tbl** Redshift table using ***tRedshiftRow*** component.
2	Load the final output file on S3 Harmonized bucket to Redshift table using ***tRedshiftBulkExec*** component. The access key and secret key to access the TalendTarget S3 bucket is passed from the context variable.

4. Job Parameters

The Talend jobs are driven by a set of parameters that are loaded from property files stored in the Credentials and Talend Source buckets. The name of the S3 credential bucket is the primary parameter provided by the user to the jobs. The Credential bucket name can be retrieved by looking at the Cloud Formation stacks output section in AWS Management Console.

The Credential bucket has the S3 property file (oodle-S3.properties). The S3 property file has the name of the Source and Target buckets.

Below 2 configuration files are available in the Talend Source Bucket.

- **oodle_input_data.txt** – Contains all the input information about Source and Target File location, Filename, File layout (that will be used to create Hive Table Schema), File Type, etc. These will used as context variables inside the Talend jobs.

- **oodle-job.properties** – This properties file contains all the additional configuration parameters that are internally used by the talend jobs

Apart from this, the following properties files are generated as part of cloudformation template and fed to Talend context variable. These are present in the Credentials bucket.

- **oodle-s3.properties** –S3 source and Target configuration parameters.
- **oodle-emr.properties** – EMR node name configuration parameters for Talend spark configuration
- **oodle-redshift.properties** – contains Redshift database details

Below are the parameters used by Talend jobs

Parameter name	Description	Usage in Jobs
S3_Folder_Name	Source File S3 Folder name	Used in *S3_to_HDFS* job
S3_File_Name	Source Filenames on S3	Used in *S3_to_HDFS* job
HDFS_Output_Path	HDFS file location for each feed	Used in *S3_to_HDFS* job
Hive_DB_Name	Hive Database name	Used in *S3_to_HDFS* job
Hive_Table_Name	Hive Table names for each feed	Used in *S3_to_HDFS* job
Hive_Table_Schema	Hive Table schema for each feed	Used in *S3_to_HDFS* job
Load_Type	File Type to identify load process	Used in *S3_to_HDFS* job
S3_Access_Key	S3 Access Key to access S3 buckets	Used in *S3_to_HDFS, HDFS_to_S3* and *S3_to_Redshift* jobs
S3_Secret_Key	S3 Secret Key to access S3 buckets	Used in *S3_to_HDFS, HDFS_to_S3* and *S3_to_Redshift* jobs
S3_Source_Bucket	S3 Source bucket for Talend jobs	Used in *S3_to_HDFS* job to fetch the Source File location details
S3_Source_Folder_Name	Source file folder location on Amazon S3	Used in *S3_to_HDFS* job to fetch the Source File location details
S3_Target_Bucket	S3 Target bucket to store output file from Talend jobs	Used in *HDFS_to_S3,* and *S3_to_Redshift* jobs to fetch the Target File location details
S3_Target_Folder_Name	Target file folder location on Amazon S3	Used in *HDFS_to_S3* and *S3_to_Redshift* jobs to fetch the Target File location details
Hive_Port	Hive Port number	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hive_Database	Hive Database name	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity

Hive_Username	Hive credential	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hive_Password	Hive credential	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hive_Server	Hive Server name on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hive_AdditionalJDBCParameters	Hive Additional JDBC Parameter	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hadoop_URI	Hadoop URI details on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hadoop_ResourceManager	Hadoop Resource Manager on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hadoop_ResourceManagerScheduler	Hive Resource Manager Scheduler on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hadoop_JobHistory	Hadoop Jobhistory on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hadoop_username	Hadoop credential	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
Hadoop_STG_DIR	Hadoop Intermediate file location on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
HDFS_Stg_Output_Path	Hadoop Staging file location on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
HDFS_Tgt_Output_Path	Hadoop Target file location on EC2	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
HDFS_OutputDailyFeedDir	Hadoop file location on EC2 for Daily Output feed	Used in both *S3_to_HDFS* and *Spark_Daily_Feed_Transform* job for Hive connectivity
InputParamFileName	Input Parameter File name with absolute Hadoop path	Used in Parent *Master_job* to load the contexts
ConfigParamFileName	Config Parameter File name with absolute Hadoop path	Used in Parent *Master_job* to load the contexts
JsonSerDeJarPath	JSON serde JAR file path on Hadoop	Used in *S3_to_HDFS* job to fetch the daily JSON file details
RedshiftHost	Redshift Host server name	Used in *S3_to_Redshift* job

RedshiftPassword	Redshift credential	Used in *S3_to_Redshift* job
RedshiftDBName	Redshift Database name	Used in *S3_to_Redshift* job
RedshiftPort	Redshift Port number	Used in *S3_to_Redshift* job
RedshiftUsername	Redshift Credential	Used in *S3_to_Redshift* job
TalendSourceBucket	Source Input Data File S3 bucket	Used in Parent *Master_job* to copy the files from S3 to Jobserver
TalendTargetBucket	Target File S3 Bucket	Used in *HDFS_to_S3* and *S3_to_Redshift* job
CredentialBucket	Config property files' S3 Bucket	Used in Parent *Master_job* to copy the files from S3 to Jobserver
Studio_jobserver_homedir	Home location of Jobserver, where the S3 files are copied	Used in in Master job and *S3_to_HDFS* job
Input_oodle_paramfile	Parameter filename	Used in Parent *Master_job* to load the contexts

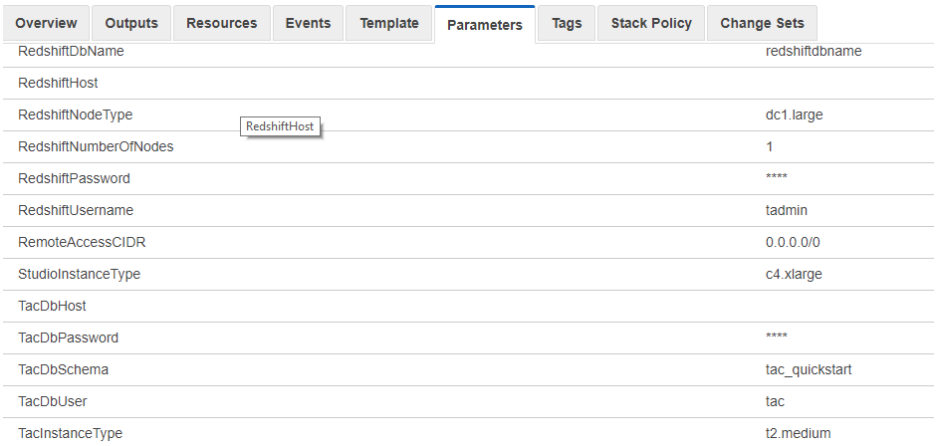
5. Preparing Talend Studio

The Talend Studio can either be installed locally or in a remote server. Follow the instructions in section 5.2 for installing Talend Studio on the local desktop. Refer to section 5.2 to use remote server installation of the Talend studio.

5.1. Get Metadata details of Datalake services

1. Go to AWS management console and open CloudFormation services
2. Click parent stack and go to outputs tab
3. All host name and below details would be available in outputs tab of parent stack. These details will be required for step by step execution of job.
   - Talend Credential Bucket
   - Talend Source Bucket
   - Talend Target Bucket
   - TAC URL
   - GIT URL
   - Nexus URL
   - xWindows Studio DNS
   - EMR Master node DNS
   - Redshift host

4. Click Parameters tab. User name of TAC, job server, Redshift, GIT and Nexus will be available in this section. Password provided while creating the stack should be known to user. These details will be required for step by step execution of job.



5.2. Install Talend Studio Locally

Install JRE

IMPORTANT: If you opt to use Talend Studio locally, you will need to install the Oracle JRE in order to be able to run the Studio.

Download and install JRE from here based on the local configuration

Set up JAVA_HOME

Set up JAVA HOME environment variable for Talend studio to use the Java environment installed on your machine

To do so, proceed as follows:

1. Find the folder where Java is installed, usually C:\Program Files\Java\JREx.x.x (sample).
2. Open the Start menu and type Environment variable in the search bar to open the Environment variable properties (or) Right click My computer or This PC icon in desktop → Properties -> Advanced system setting → click Environment variables
3. Under System Variables, click New to create a variable
4. Enter variable name as JAVA_HOME, enter the path of the Java 8 JRE, and click OK.
5. Under System Variables, select the Path variable, click Edit... and add the following variable at the end of the Path variable value:

   ;%JAVA_HOME%\bin

IMPORTANT: If you opt to use Talend Studio on your laptop, you must ensure that your IP address is included in the REMOTE_CIDR network range configured in the Quickstart Parameters.

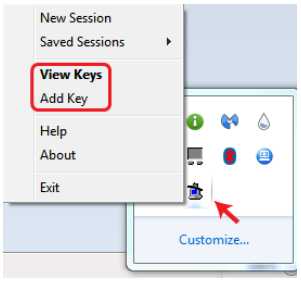
Unzip the Talend Studio archive

1. Copy the archive *Talend-Studio-YYYYYYYY_YYYY-VA.B.C.zip* to a directory of your choice.
2. Unzip it.

5.3. Connecting to Remote Studio using X2Go

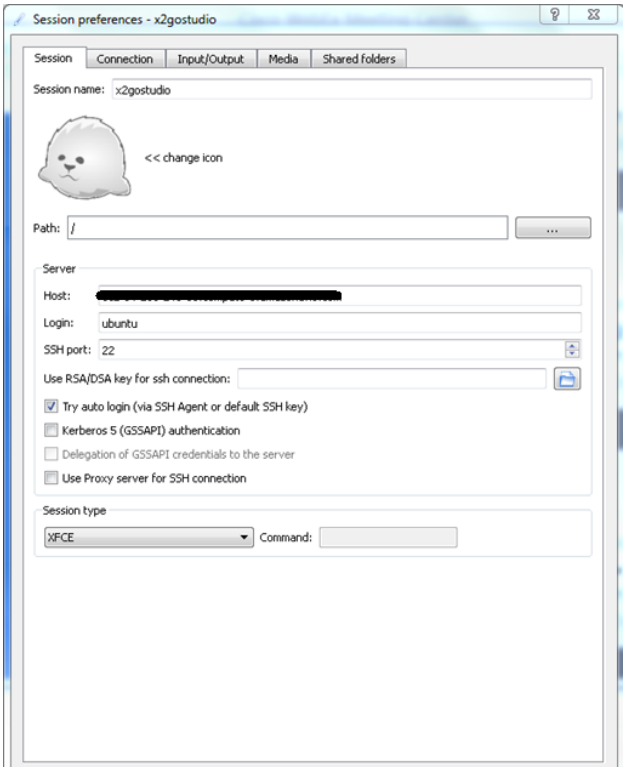
5.3.1. X2Go on Windows

1. Download puttygen from here
2. Use puttygen to convert .PEM file that will be received while creating to .PPK file. Please refer section *'Converting Your Private Key Using PuTTYgen'* in this link for more details
3. Download Pageant from here
4. Add PPK key (from step 2) to pageant

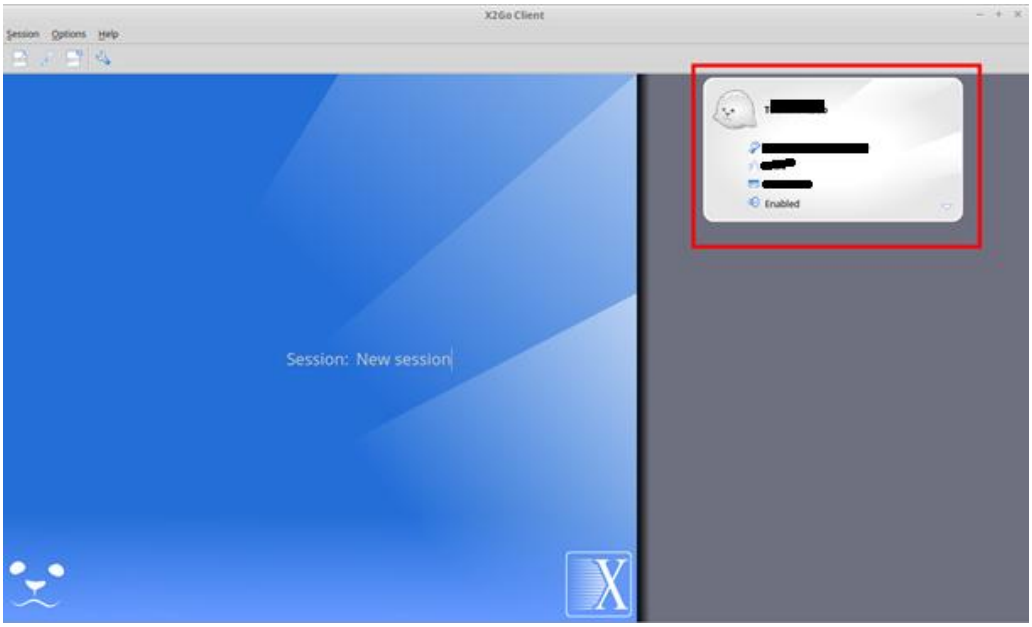


5. Download X2Go client from https://wiki.x2go.org/doku.php/download:start
6. Follow the installation instructions in https://wiki.x2go.org/doku.php/doc:installation:x2goclient
7. Launch X2Go client

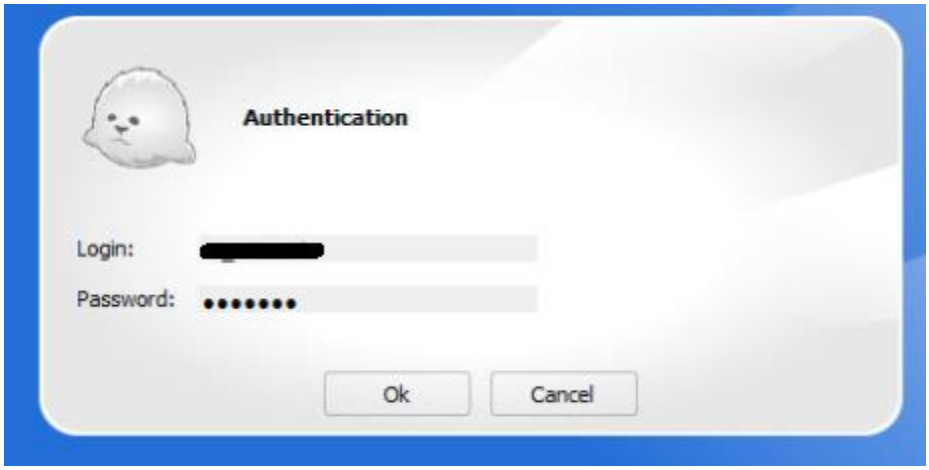
8. Enter host name details and check "Try auto login Agent or default"
9. Select session type as XCFE



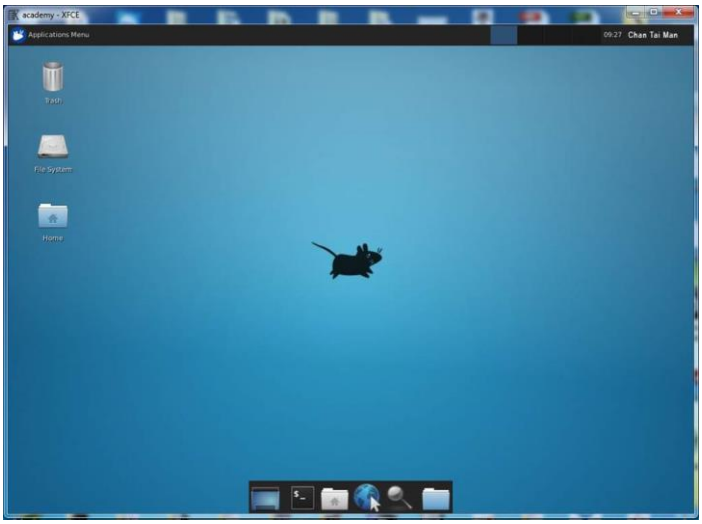
10. Click and activate the newly created session



11. Once this session is selected, it will prompt for the user on the remote machine's credentials.



12. Below screen ensures successful login



5.3.2. X2Go on Linux

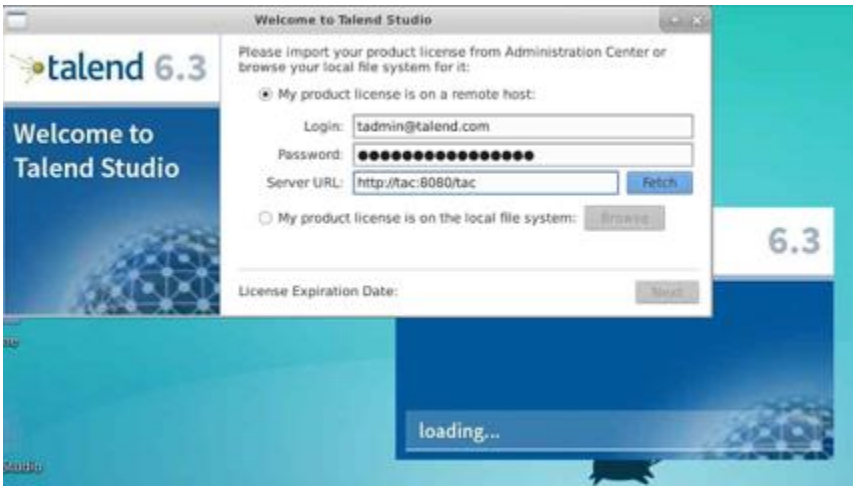
1. Follow the installation instructions in

   https://wiki.x2go.org/doku.php/doc:installation:x2goclient

2. Pls follow instruction in this link to configure and connect to X2GO instance

5.4. Setup Studio to connect with TAC

1. Launch *Talend Studio*.
2. Choose “My Product license is on remote host”
3. Enter Login, password and server URL details and click Fetch. Once you see a green highlighted message “Your license for Talend Real-Time Big Data Platform is valid” click Next.

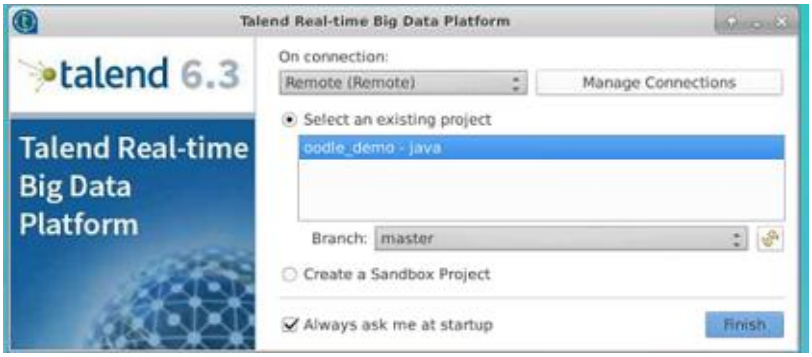


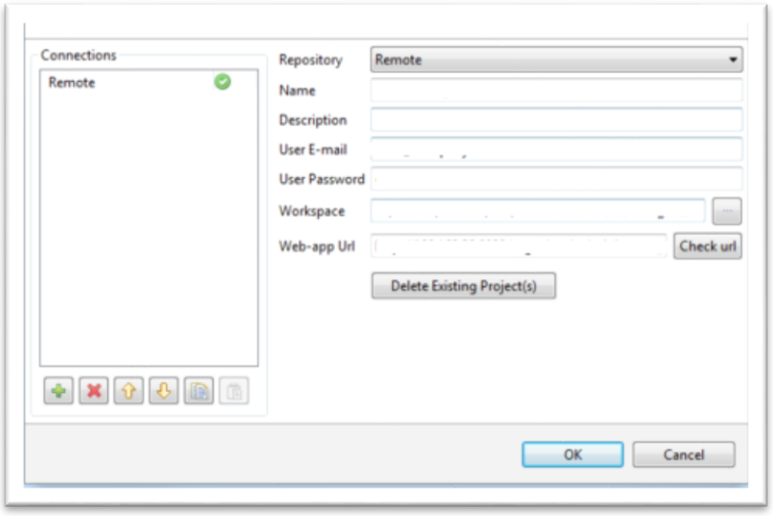
4. Select “Remote” on connection and click Manage connections. In the pop up window, pls enter following details
5. Set the **Repository** type as *Remote* and enter a **Name** and **Description** for the connection, the **E-mail** and **Password** for the user you created in *Talend Administration Center. Pls use tadmin@talend.com for email. You can get these values from the Outputs section of the parent Stack.*

5.1. *If you are running Studio on the remote desktop, then you can use this*. URL for the **Web-app Url** field *http://tac:8080/tac* . The tac hostname has been defined for you in the /etc/hosts file on the remote machine.

5.2. *If you are running* Talend Studio on your own laptop then the TAC hosts name will not be defined, and you will need to specify the TAC hostname URL based on the values provided to you in the Outputs section of the parent Stack. Pls refer section 5.1 for steps to navigate to output section of parent stack.

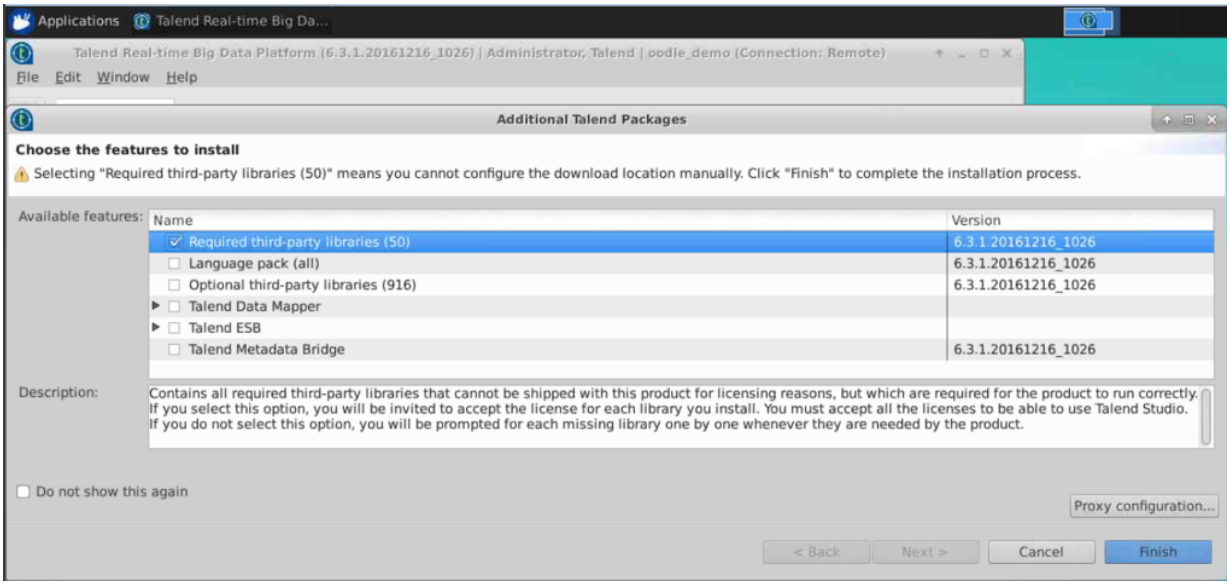
6. If installing on your own laptop, be careful not to use an existing local workspace. If needed, you can create another folder in the Talend Studio alongside the default workspace folder.
7. Click **OK**.
8. Project ‘oodle_demo – java’ will be listed on successful completion.



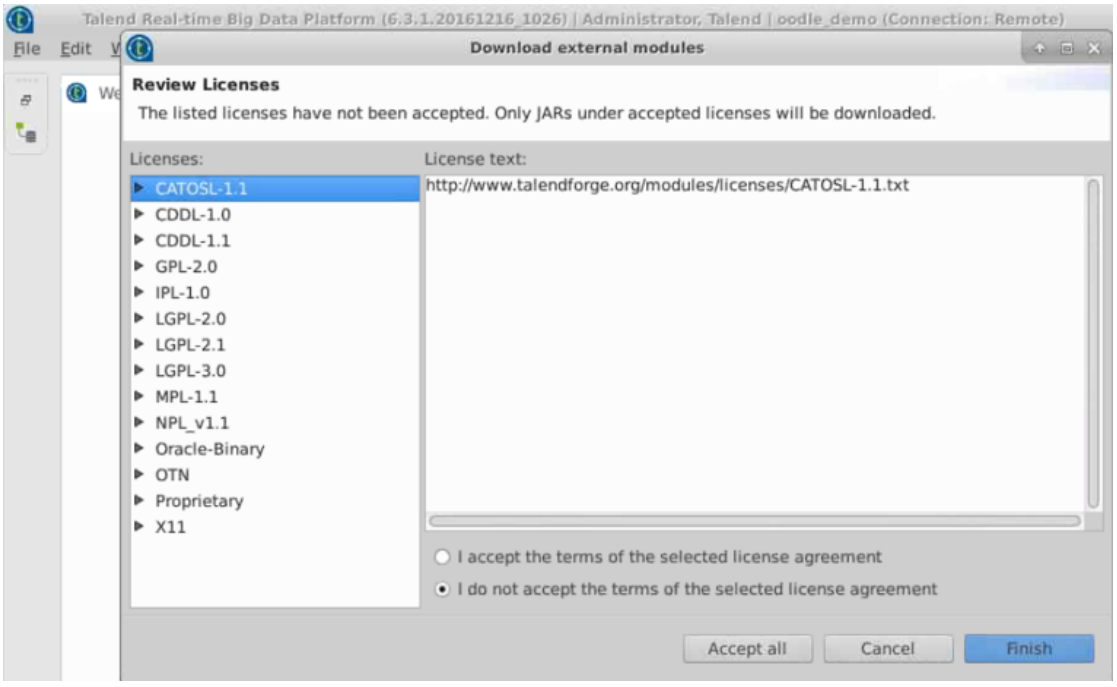


5.5. Loading Libraries to Studio

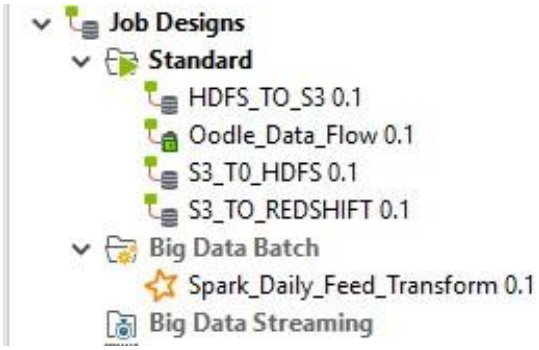
1. Some external third-party libraries are required by some components or connection wizards to work properly and must also be installed. The [Additional Talend Packages] wizard lets you install the additional packages.



2. Accept the license terms, click Accept all and click Finish to install all the modules.



3. Close the welcome screen. Below jobs would be listed by default in the job designs.

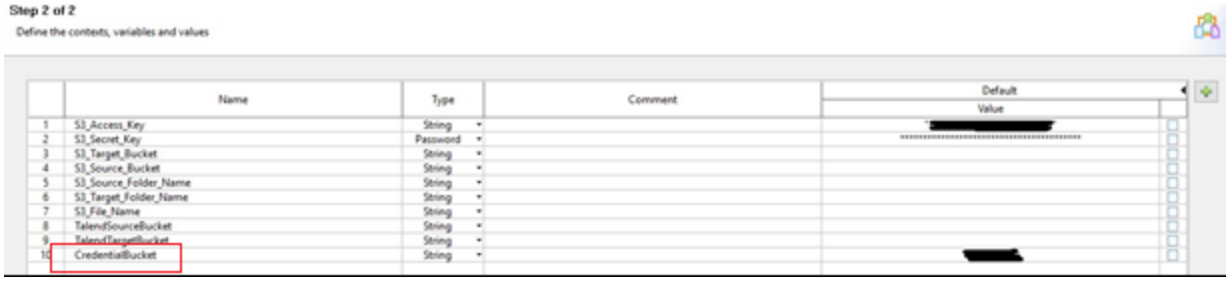


6. Step by Step Execution of demo job

6.1. Configure the Job

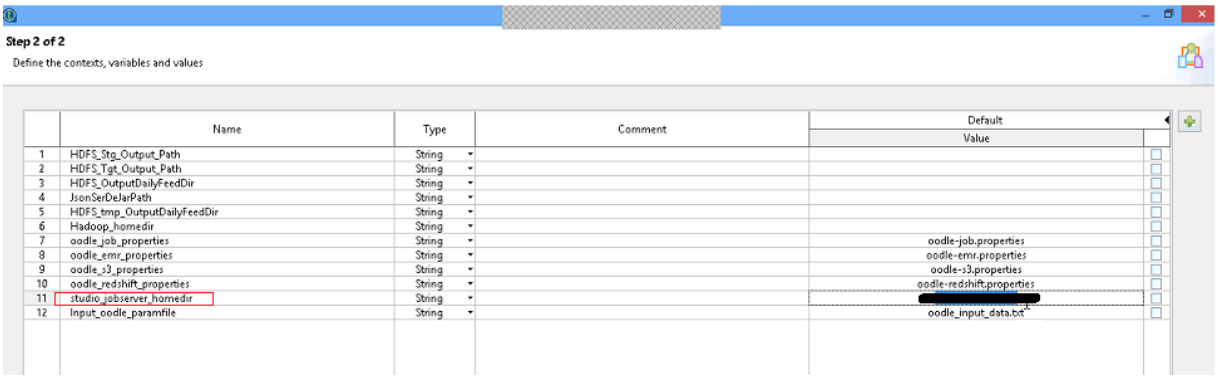
Go to contexts and open *AWS_S3_Context,* specify the following

1) *CredentialBucket* variable. This details will be available in output section of parent stack. Pls refer section 5.1 for steps to navigate to output section of parent stack.
2) *S3_Access_Key* and *S3_Secret_Key* variables.



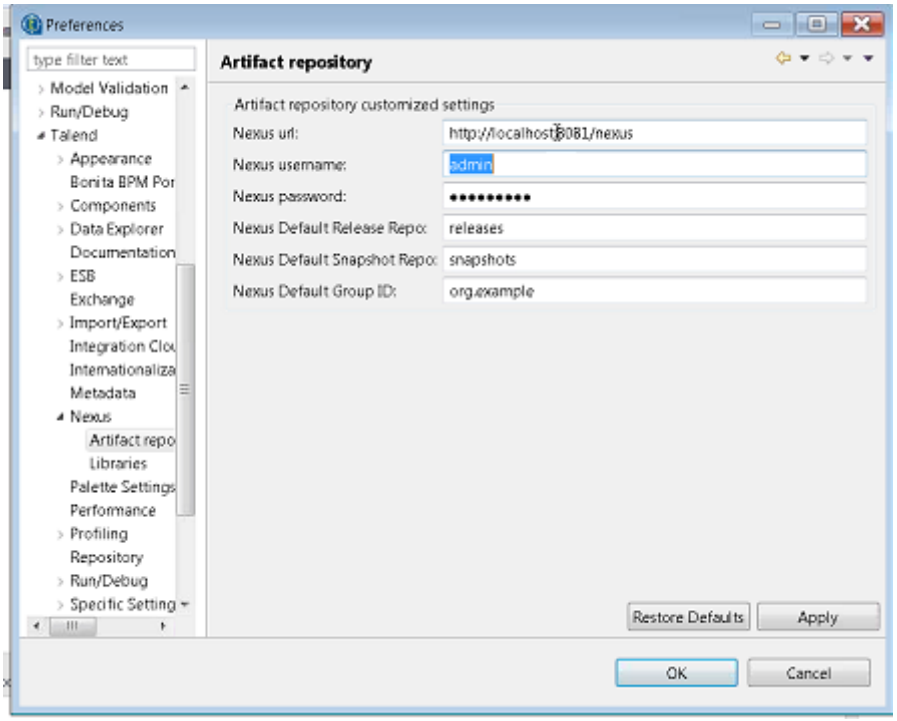
Open Oodle_Default_Context context, specify the path under studio_jobserver_homedir i.e. landing directory where the S3 files should be copied to in the job server.

Important: Make sure that directory has write permission to 'Talend' user.



6.2. Configure Nexus in Talend Studio

1. In Talend Studio, navigate to Window -> Preferences
2. In the Preferences dialogue, select Nexus -> Artifact Repository
3. Provide the Nexus URL, Username and password.

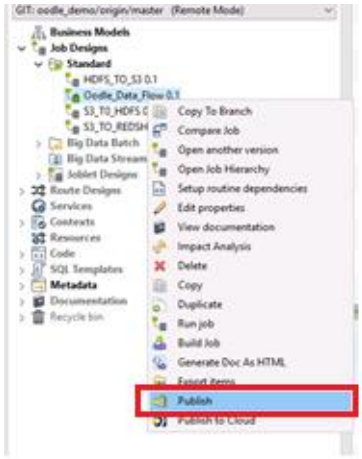


- Nexus URL can be retrieved from the output of cloud formation parent stack. Pls refer section 5.1 for steps to navigate to output section of parent stack.
- User name and password can be retrieved from oodle-nexus.properties present in S3 credential Bucket. Details of credential bucket will be available in output tab of parent stack.
- No changes to Nexus Default release Repo, Nexus Default Snapshots Repo and Nexus Default Group ID

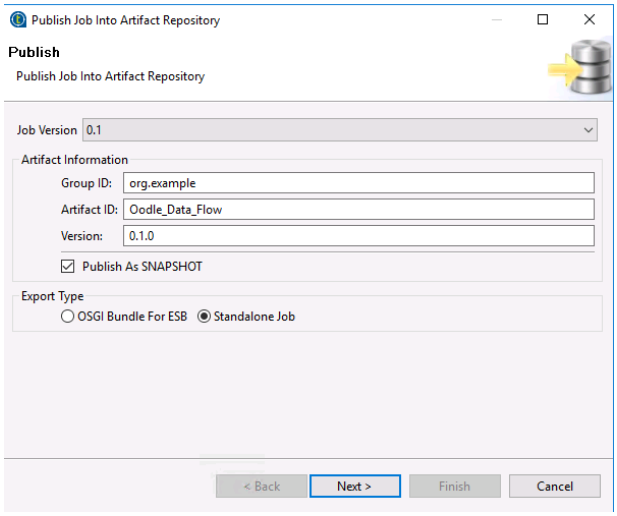
- Click OK

6.3. Publish the Job to Nexus and run from TAC

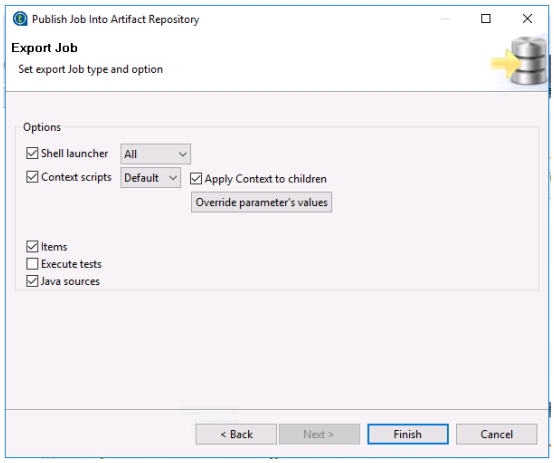
1. Right click OODLE_DATA_FLOW job → Select Publish



4. Select “Standalone Job”, click Next



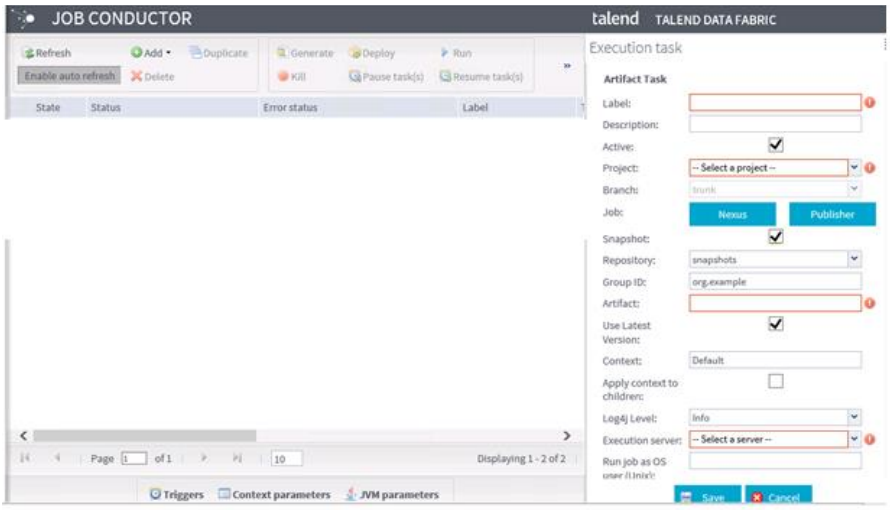
5. Select “Apply Context to children”, click finish to publish the job to Nexus



6.4. Run the Job from TAC

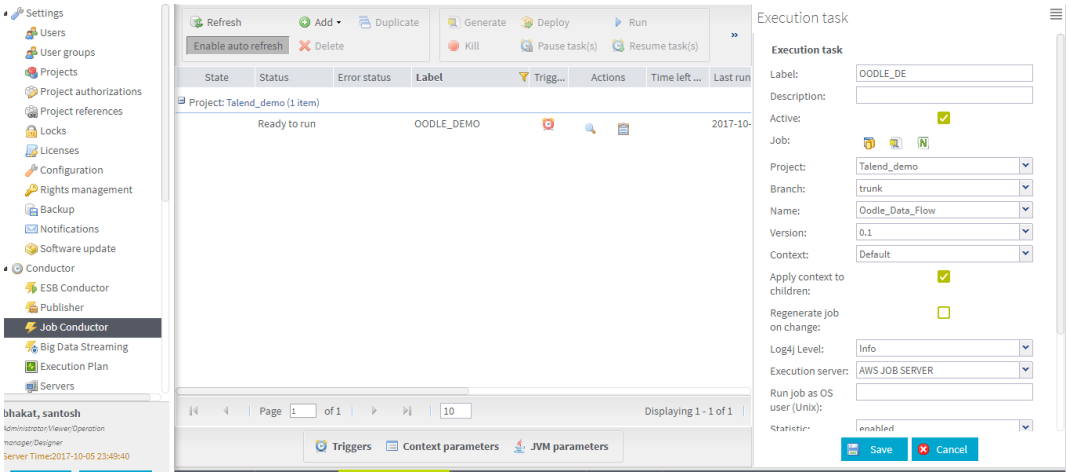
1. Connect to Talend administration console in a Browser *http://TACDNS:8080/org.talend.administrator. TACDNS will be available in output of parent stack.*

2. Go to Job Conductor in TAC and click "Add Artifact Task"



3. Click Nexus button
    a. Select the Repository
    b. Select the artifact
    c. Specify the job server against which to run
    d. Click OK

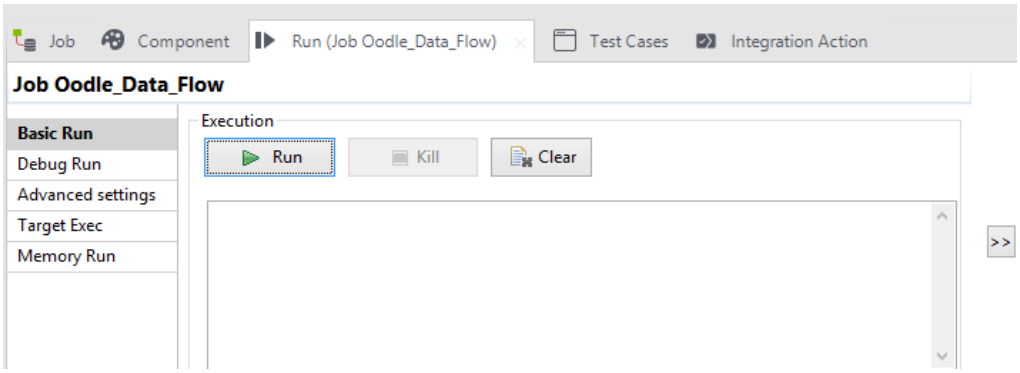
4. Click deploy and run



5. Once the execution is OK, follow verification steps

6.3 Run the Job in Studio

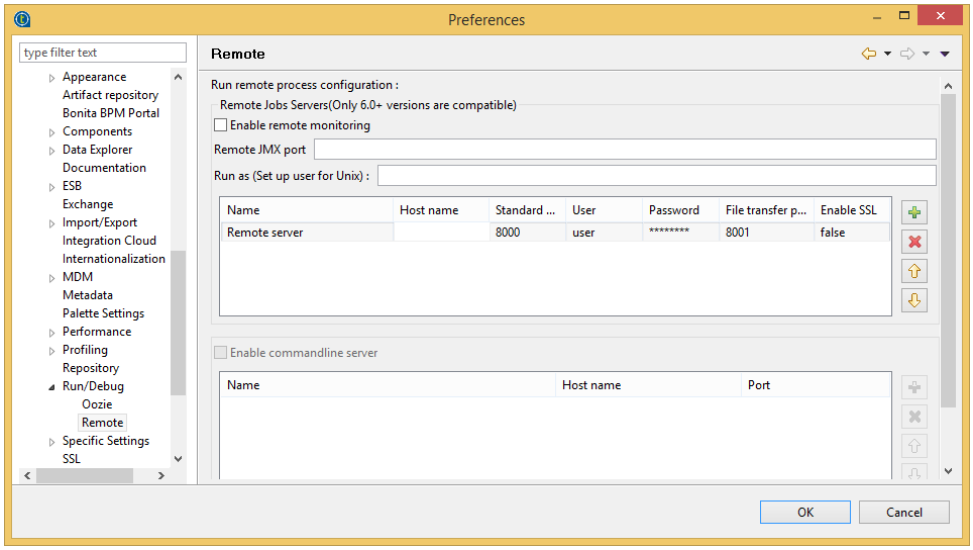
1. Open *Oodle_Data_Flow* job and go to run tab. Click Run button in Basic Run.



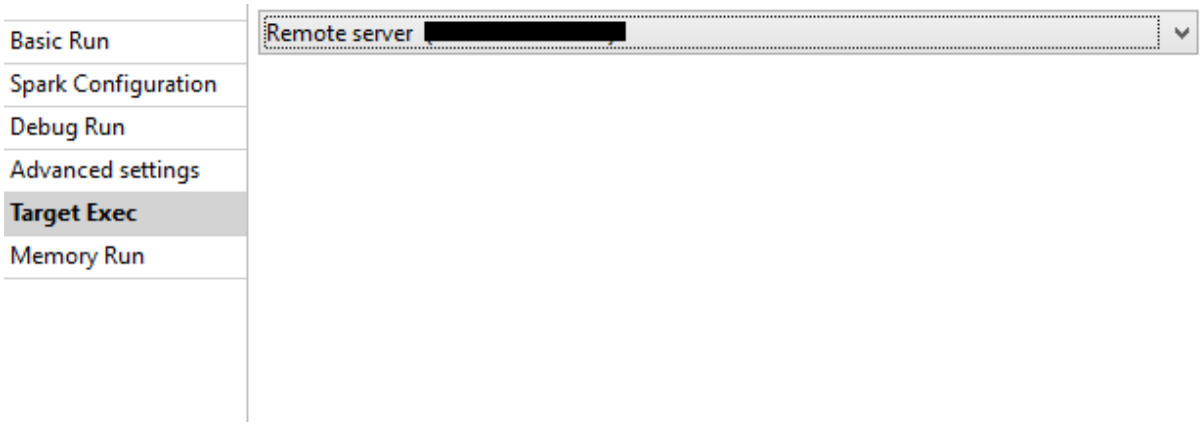
2. Once the execution is completed successfully, follow the verification steps.

6.5. Run the Job through Distant Run server

1. In Talend Studio, Go to windows → preferences → Run/Debug → Remote and update the job server host name details from the Outputs section of the parent stack. .



2. Open the OODLE_DATA_FLOW job
3. In the Run tab/view, click on the Target Exec tab, select the Distant Run server.



4. click on the Run button located in the Basic Run tab.
5. Once the execution is completed successfully, follow the verification steps.

6.6. Verification

1. Connect to HUE http://<master *node hostname>:8888* and check if Customer, Physician, Provider in the browser and fitbit_daily_feed HIVE table are created and loaded with data. Refer Section 5.1 to get master node DNS details.

2. Check for FinalDailyFeed.txt in TalendTargetBucket. Target bucket details will be available in parameter tab of parent stack. Pls refer section 5.1 for steps to navigate to parameter section of parent stack.

3. Connect to Redshift database and query Fitbit_daily_tgt_tbl table to verify if data is properly loaded from FinalDailyFeed.txt file. Refer connect to redshift using workbench to connect to redshift cluster. Refer Section 5.1 on how get the Redshift details.