Customer :Avalara

Prepared by :Shanthan Kakulavaram

Date : 03/31/2020

Infrastructure Guide

Contents

[Talend Cloud functional architecture 3](#_Toc36566279)

[Local network 3](#_Toc36566280)

[Port Need to Be Opened to connect to studio to run remotely 4](#_Toc36566281)

[Remote Engines 4](#_Toc36566282)

[Remote Engines Backup Files 5](#_Toc36566283)

[Cloudwatch Monitoring 5](#_Toc36566284)

[Remote Engine Enablement for Studio 5](#_Toc36566285)

[Remote Engines Environment Configuration 6](#_Toc36566286)

[Remote Engines Health Check 7](#_Toc36566287)

[Production and Disaster Recovery Strategy 7](#_Toc36566288)

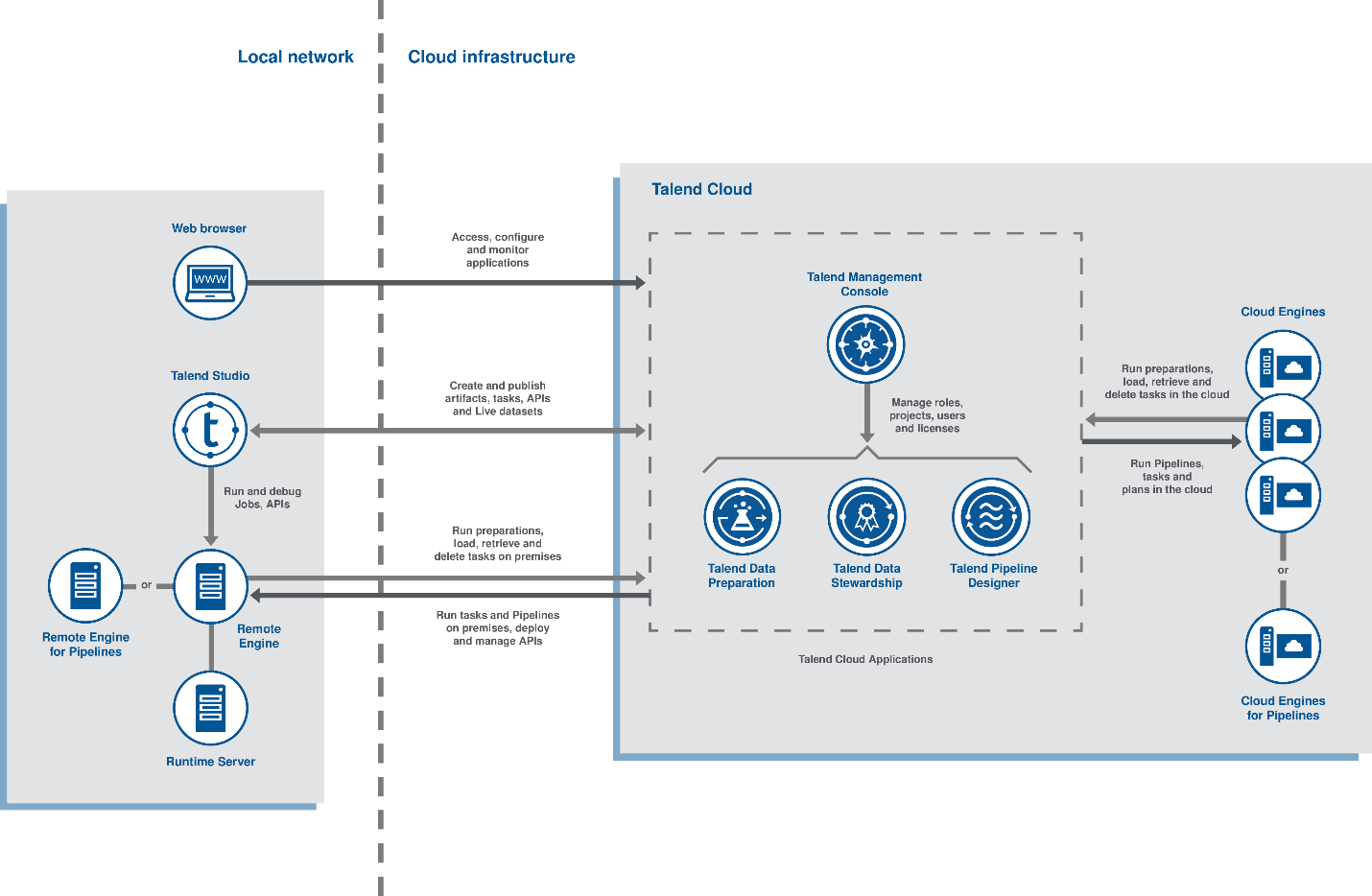
[Pairing Remote Engine with existing Paired Remote Engine config 7](#_Toc36566289)

[Pairing Remote Engine with webservice 8](#_Toc36566290)

[Applying Talend Remote Engine Patches 9](#_Toc36566291)

# Talend Cloud functional architecture

Talend Cloud is a fully-managed cloud option. It provides the great data integration, data quality, and Big Data features from Talend in a cloud environment that is managed, monitored, maintained, and secured by Talend.



The diagram of the Talend Cloud functional architecture is divided into two main parts: the local network and the cloud infrastructure.

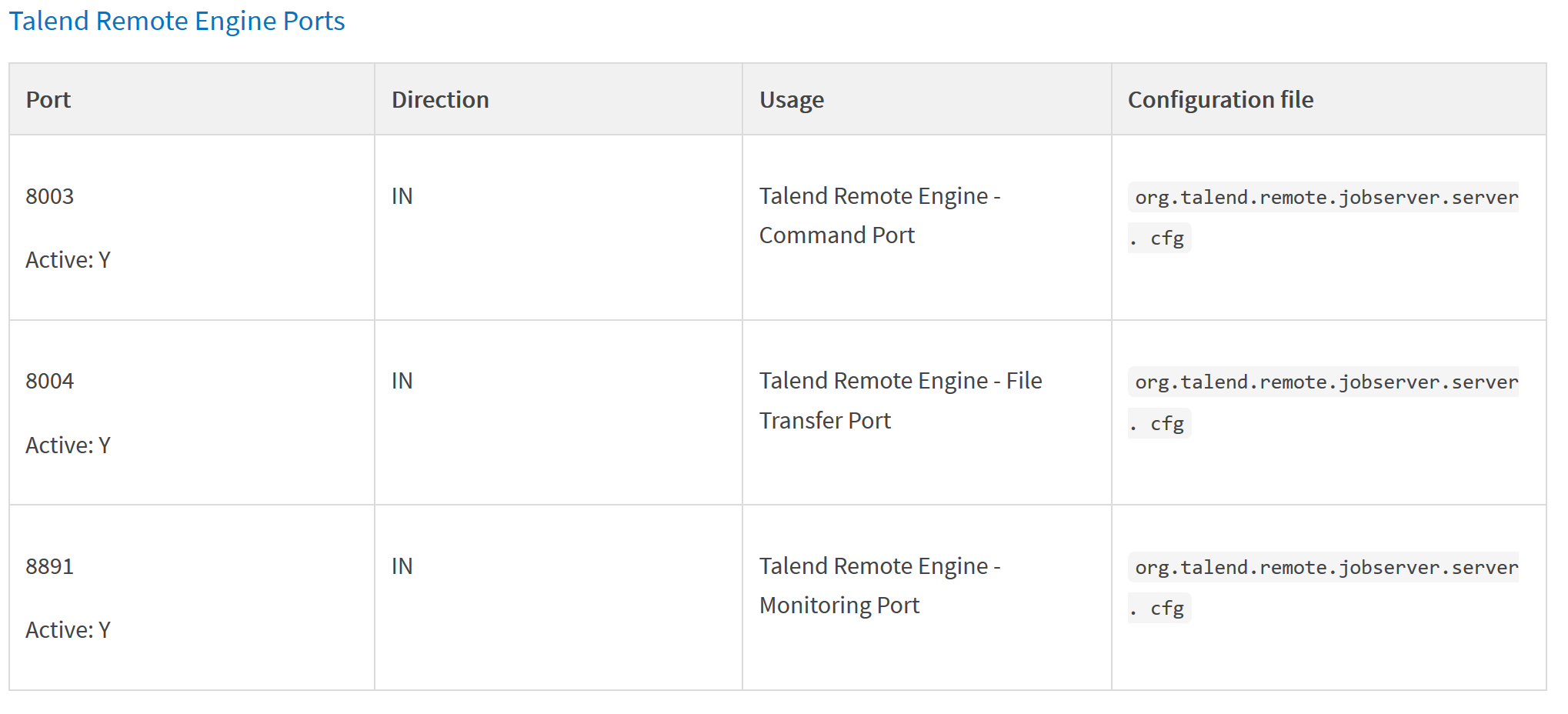
## Local network

The local network includes a web browser, Talend Studio, and a Remote Engine or a Remote Engine for Pipelines.

* From your web browser, you can access the different cloud applications.
* From Talend Studio, you can:
  + Publish data integration Jobs to Talend Cloud Management Console as Tasks, make them available to web users, and run them in the cloud.
  + Benefit from the Talend Cloud Data Preparation features through the use of the tDatasetInput, tDatasetOutput, and tDataprepRun components. You can create datasets from various databases and export them in Talend Cloud Data Preparation, or leverage a preparation directly in a data integration Job or Spark Job.
  + Use Jobs with tDataStewardshipTaskOutput, tDataStewardshipTaskInput and tDataStewardshipTaskDelete to load, retrieve, or delete tasks from the campaigns created in the Talend Cloud Data Stewardship.
* The Talend Remote Engine is used to run Jobs, Tasks, and preparations, or pipelines, on premises.

## Port Need to Be Opened to connect to studio to run remotely

PROCESS\_MESSAGE\_PORT = 8558 Direction = OUT Publishes message.



## Remote Engines

All the components of Talend Cloud is managed in Talend Cloud Web UI except the Remote Engines. The infrastructure team must install and manage the Remote Engines.

* Configuring the remote Engine Manually – (These steps can be automated through ansible)
  + Update the file “Talend-Remote-Engine-wrapper.conf” to set the Java-Home
  + Update the preauthorized.key.cfg file  with the correct remote engine  configuration
  + Restart the remote engine (Systemctl start Talend-Remote-Engine)

Configuration of Remote Engine:

1. Set JAVA\_HOME path in <RemoteEngineInstallationDirectory>/etc/Talend-Remote-Engine-wrapper.conf.
2. Set Remote Engine SSH Keys, Remote Engine Name and Description in <RemoteEngineInstallationDirectory>/etc/preauthorized.key.cfg

**remote.engine.pre.authorized.key=**{*Paste the preauthorized key*}

**remote.engine.name=**{*Enter the name of the Remote Engine*}

**remote.engine.description=**{Enter the description of the Remote Engine}

1. Remove concurrent task limit – set max.deployed.flows=0 in <RemoteEngineInstallationDirectory>/etc/org.talend.ipaas.rt.deployment.agent.cfg

Note: Start or Restart the Remote Engine – systemctl start Talend Remote Engine-service.

## Remote Engines Backup Files

The Snapshot of Remote Engine is backed up daily end of the day. Snapshot of last 7 days are stored.

Talend Home directory will be ‘<REMOTE\_Egnine\_Installation\_Path>’. This folder will be backed up daily and last 7 days back up will stored.

Logs are stored in Splunk:

Remote Engine server Log can be found at ‘<REMOTE\_Egnine\_Installation\_Path>data/log/wrapper.log’.

Job logs can be found at ‘<REMOTE\_Egnine\_Installation\_Path>/TalendJobServersFiles/jobexecutions/logs’

Note/Action Item: Lifetime of Karaf.log and Wrapper.log

## Cloudwatch Monitoring

Cloudwatch alerts are to be configured to send us the notifications on the below metrics

* CPU Utilization -   >  given threshold  (Threshold can be set to 85-95%)
* Memory utilization (Currently Member metrics are not collected on the AWS)  > the  threshold (Threshold can be set to 95%)

## Remote Engine Enablement for Studio

Open file path - <REMOTE\_Egnine\_Installation\_Path>etc/org.talend.remote.jobserver.server file and edit the below lines comment them with #:

### Define ports

# the host/port used to launch talend commands

org.talend.remote.jobserver.server.TalendJobServer.COMMAND\_SERVER\_PORT=8003

**#org.talend.remote.jobserver.server.TalendJobServer.COMMAND\_SERVER\_HOST=localhost**

# the host/port for the file transfert

org.talend.remote.jobserver.server.TalendJobServer.FILE\_SERVER\_PORT=8004

**#org.talend.remote.jobserver.server.TalendJobServer.FILE\_SERVER\_HOST=localhost**

# The host/port used for monitoring the servers state

org.talend.remote.jobserver.server.TalendJobServer.MONITORING\_PORT=8891

**#org.talend.remote.jobserver.server.TalendJobServer.MONITORING\_HOST=localhost**

# The host/port used for the execution of process messages publisher, 8558 by default

org.talend.remote.jobserver.server.TalendJobServer.PROCESS\_MESSAGE\_PORT=8558

**#org.talend.remote.jobserver.server.TalendJobServer.PROCESS\_MESSAGE\_HOST=localhost**

**# The local hostname of this server (default value is "localhost")**

**#org.talend.remote.jobserver.server.TalendJobServer.LOCAL\_HOST=localhost**

## Remote Engines Environment Configuration

The EC2 instance with Remote Engine installed will have configuration for all the jobs that being executed in respective Remote Engine. Some of these configurations will be controlled by developers and remaining by infrastructure team.

**Job Exececution configs** - The parameters (like JVM parametes) that needs to be set by Infrastructure Admin will be set as Environment variable. If a parameter needs to be edited or added, Infrastructure Admin will add/edit the environment variable.

**Unix command to configure Remote Engines and additional RE configs:**

* + 1. Set Maximum number of tasks that can be executed –

Filepath - <REMOTE\_Egnine\_Installation\_Path>etc/org.talend.eventlogging.agent.cfg

Set max deployed flows=0

* + 1. Set Maximum Heap size in MB –

Filepath - <REMOTE\_Egnine\_Installation\_Path>etc/Talend Remote Engine-wrapper.conf

Prod & Performance - Set wrapper.java.maxmemory=2048.

Non Prod - Set wrapper.java.maxmemory=512.

* + 1. Set Runtime log level –

Filepath - <REMOTE\_Egnine\_Installation\_Path>/org.ops4j.pax.logging.cfg

log4j2.rootLogger.level = DEBUG

* + 1. Lifetime of Job logs(Optional do not change unless needed) –

Filepath - <REMOTE\_Egnine\_Installation\_Path>etc/org.talend.remote.jobserver.server.cfg

org.talend.remote.jobserver.commons.config.JobServerConfiguration.MAX\_DURATION\_BEFORE\_CLEANING\_OLD\_EXECUTIONS\_LOGS=7776000

*Also validate other ‘# Temporary data cleaning parameters’*

* + 1. Set count for incoming requests or deployed jobs in Queue(Optional do not change unless needed)–

Filepath - <REMOTE\_Egnine\_Installation\_Path>etc/ org.talend.eventlogging.agent.cfg queue.input.consumers.count=5

## Remote Engines Health Check

The Remote Engine service can be verified with response of curl to configuration API provided by Remote Engine. This API is used to pair Remote Engine and it also confirms that the Remote Engine service is available.

Health Check curl command - curl -x http://ip-addaress:8043/configuration

(Alternate way is to verify if the Remote Engine listening ports 8043, 8003, 8004 and 8891,8558 are listed in netstat command.)

Note: A new API for RE’s will be released in a month or so what should provide the needed functionality.

## Production and Disaster Recovery Strategy

Maximus will have 5 Remote Engines for production and divided as following:

3 Primary Remote Engines: This remote engine will be created in same region. They will be used as primary option to execute Talend jobs.

2 Secondary Remote Engines: These remote engines are host in different region than the primary remote engines. Out of 2 Secondary Remote Engines, one engine will be active and other one will be passive. Both the remote engines will be paired but the passive remote engine will be stopped.

When an active remote engine goes down, it will be detected by Load Balancer that is set to monitor the status of remote engine with Health Check. Then passive remote engine is started to execute talend jobs. Once Primary is fixed or started, the secondary passive will be stopped.

Before stopping the passive remote engine, we should check if any task is being executed in passive remote engine and no task is being deployed to execute in passive remote engine.

Note/ Action Item: Endpoint to verify jobs executed in RE. So that we can confirm no jobs executed in Passive before shutting it down.

If primary tasks goes down, the scheduled jobs will be terminated. Is there way to restart them with Secondary RE?

## Pairing Remote Engine with existing Paired Remote Engine config

The Remote Engine in a EC2 instance is paired with Talend cloud with preauthorized key to receive execution request to execute the job. In order to automate the process of Disaster Recovery (DR) of Remote Engine, we are going to perform health check to identify the status Remote Engine. If the remote engine is down, we must have Remote Engine installed in DR EC2 instance with no preauthorized key set.

When a remote engine is paired, then below files are modified with cloud encrypted keys and credentials in folder <RemoteEngineInstallationDirectory>/etc/:

* org.ops4j.pax.url.mvn.cfg
* org.talend.eventlogging.agent.cfg
* org.talend.eventlogging.sender.jms.cfg
* org.talend.ipaas.rt.deployment.agent.cfg
* org.talend.ipaas.rt.dispatcher.client.cfg
* org.talend.ipaas.rt.dts.client.cfg
* org.talend.ipaas.rt.eventsource.amq.cfg
* org.talend.ipaas.rt.lts.client.cfg
* org.talend.ipaas.rt.pairing.agent.cfg

Follow below steps after installing the Remote Engines in DR instances.

1. The above listed 9 files must be copied from original remote engines to DR remote engines.

2. Modify the config files to configure the DR remote engines. Set concurrent task limit to 0 – set max.deployed.flows=0 in <RemoteEngineInstallationDirectory>/etc/org.talend.ipaas.rt.deployment.agent.cfg

3. Make sure that the original Remote Engine service is not running. Start or restart the service in DR remote engines.

4. Verify that Remote Engines paired in Talend cloud.

## Pairing Remote Engine with webservice

Run below curl command to pair the Remote Engine. Remote Engine preauthorized key must be copied from Talend cloud for API call.

Note: Replace ‘XXXXX…..’ with preauthorized key from Talend cloud.

curl -X POST \

  http://localhost:8043/configuration \

  -H 'Accept: \*/\*' \

  -H 'Accept-Encoding: gzip, deflate' \

  -H 'Cache-Control: no-cache' \

  -H 'Connection: keep-alive' \

  -H 'Content-Length: 201' \

  -H 'Content-Type: application/x-www-form-urlencoded' \

  -H 'Host: localhost:8043' \

  -H 'cache-control: no-cache' \

  -d 'pairingServiceUrlSelect=https%3A%2F%2Fpair.us.cloud.talend.com&pairingServiceUrl=https%3A%2F%2Fpair.us.cloud.talend.com&preAuthorizedKey=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

To re-pair Remote Engine with same pre-authorized key:

* + 1. Delete the value for ‘remote.engine.id’ in <RemoteEngineInstallationDirectory>/etc/org.talend.ipaas.rt.pairing.agent.cfg



* + 1. Pair the remote engine with the above API curl command with pre authorized key.

## Applying Talend Remote Engine Patches

Usually new Remote Engine installer released.

1. Remote Engine Installer with update and fixes released in Talend Cloud Downlowds.
2. On Demand fix will be released as installers in Talend cloud downloads or shared with client with cloud storages and URL access will be granted.

Further research required by consultant – Talend provide Read Me document with Patches.