Talend cloud-deployer Maven Plugin

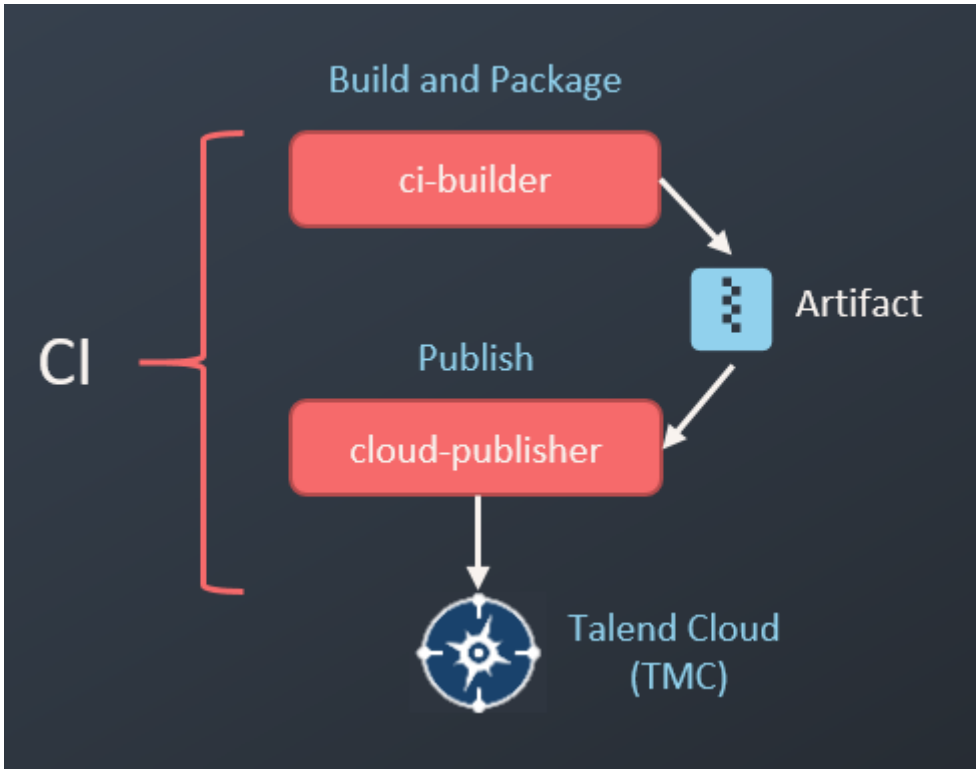
Redha Boukefoussa - (C) Talend Inc.

Version 8.5.0

Table of Contents

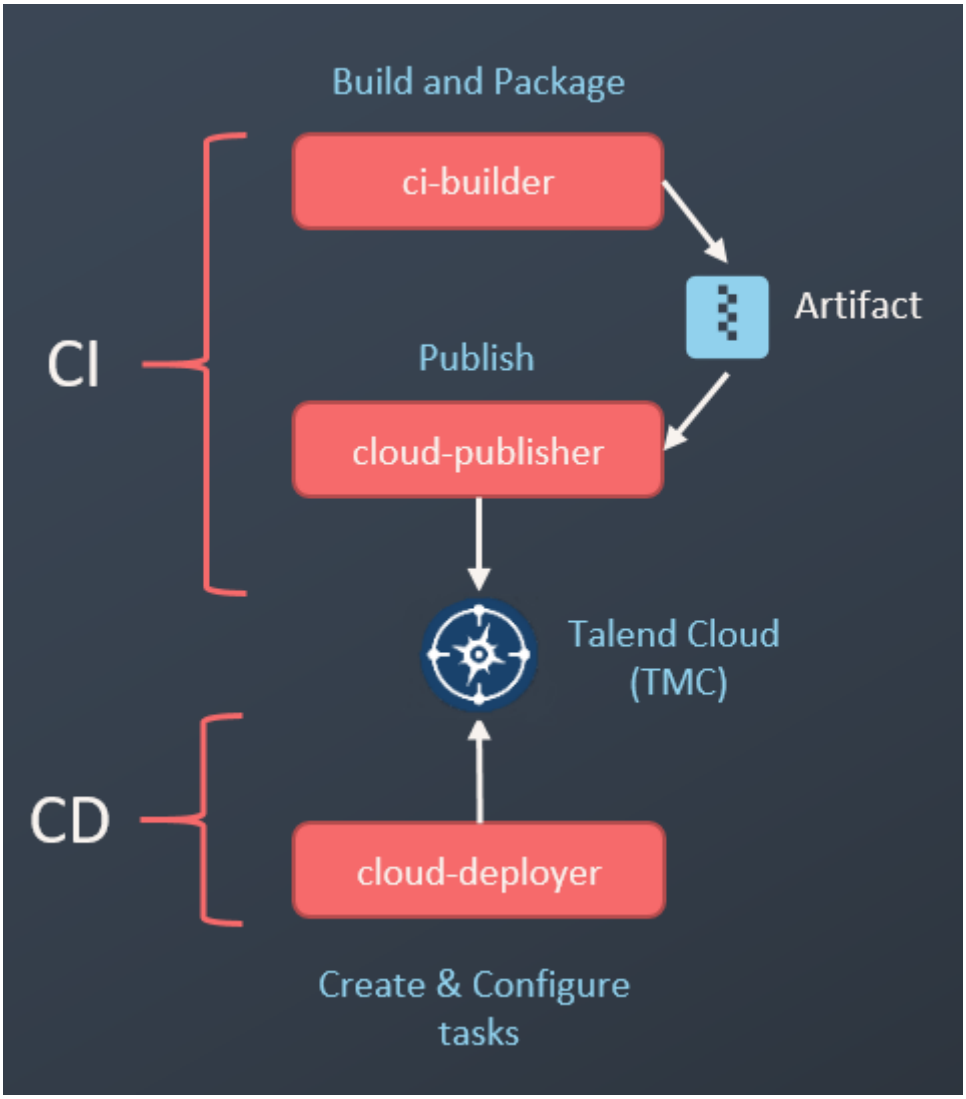
Introduction

For Talend Cloud, the SDLC workflow includes the Continuous Integration (CI) workflow implemented by the `ci-builder` Maven plugin that build and package the Talend Job artifact, followed by the `cloud-publisher` plugin that upload the generated job artifact into the internal Talend Cloud artifact repository.



The `cloud-publisher` plugins offers also the possibility to create a task for this artifact with a name based as the artifact name. This Implementation is usually sufficient for the majority of the customers, but some have reported the lack of options to handle additional configurations, such as the assignment of the Remote Engine, the list of the job connections, and all the other options available in the Talend Management Console User Interface.

To address those requirements for the automation of the Tasks management and a proper Continuous Deployment (CD) implementation, a new Maven plugin has been developed coined `cloud-deployer`. It leverages the TMC REST APIs that allow to automate the definition and configuration of a Talend Cloud Task/Flow.



The choice for a Maven plugin was deliberate to be CI/CD server agnostic hence support the ones our customers are using (Jenkins, Azure Devops, GitLab, ...) without the need to develop a dedicated solution for each of them.

IMPORTANT Starting with version `8.0.3`, `cloud-deployer` implements the artifact publication supported by `cloud-publisher`; allowing to associate the publication of an artifact with the creation/update of the associated task in the same Maven command.

The following documents the installation, configuration and usage of the `cloud-deployer` Maven plugin.

Download

The clouddeployer Maven plugin is not currently published in Maven Central repository but a package containing the jar file, the associated pom file and this documentation can be downloaded from this url:

https://talend-op2c-downloads.s3.amazonaws.com/clouddeployer-maven-plugin/clouddeployer-maven-plugin-8.5.0.zip

Requirements

Java version: 11

Maven version: `3.9.4`

Installation

The plugin must be accessible by Maven; therefore, it needs to be installed either in the local `.m2` repository or, preferably, uploaded into the Artifact Repository configured for Talend CI/CD.

The repository `talend-custom-libs-release` where the other Talend Maven plugins: `ci-builder` and `cloud-publisher` reside is a good candidate.

Local repository

Unzip the downloaded package and run the following maven command in the created folder to install the plugin into the local `.m2` repository.

```
mvn install:install-file \
-Dfile=clouddeployer-maven-plugin-8.5.0.jar \
-DpomFile=clouddeployer-maven-plugin-8.5.0.pom
```

Remote repository

The plugin (and it's pom file) can be uploaded into a Nexus or JFrog Artifactory repository using the Maven deploy goal `deploy:deploy-file`.

From the unzipped folder, run the following commands. Update the variables `REPOSITORY_URL` to match your Artifact Repository server.

```
# For Nexus 2/3
REPOSITORY_URL=http<s>://<host>:<8081 | port>/repository/talend-custom-libs-release
# For JFrog Artifactory
# REPOSITORY_URL=http<s>://<host>:<8081 | port>/artifactory/talend-custom-libs-release

mvn \
-s settings.xml \
-DrepositoryId=talend-custom-libs-release \
-Dfile=clouddeployer-maven-plugin-8.5.0.jar \
-DpomFile=clouddeployer-maven-plugin-8.5.0.pom \
-url=$REPOSITORY_URL \
deploy:deploy-file
```

The parameter `repositoryId` is the id (represented by the target repository name) associated with the `server` entry (authentication) in your `setting.xml` file.

Ex: `settings.xml`

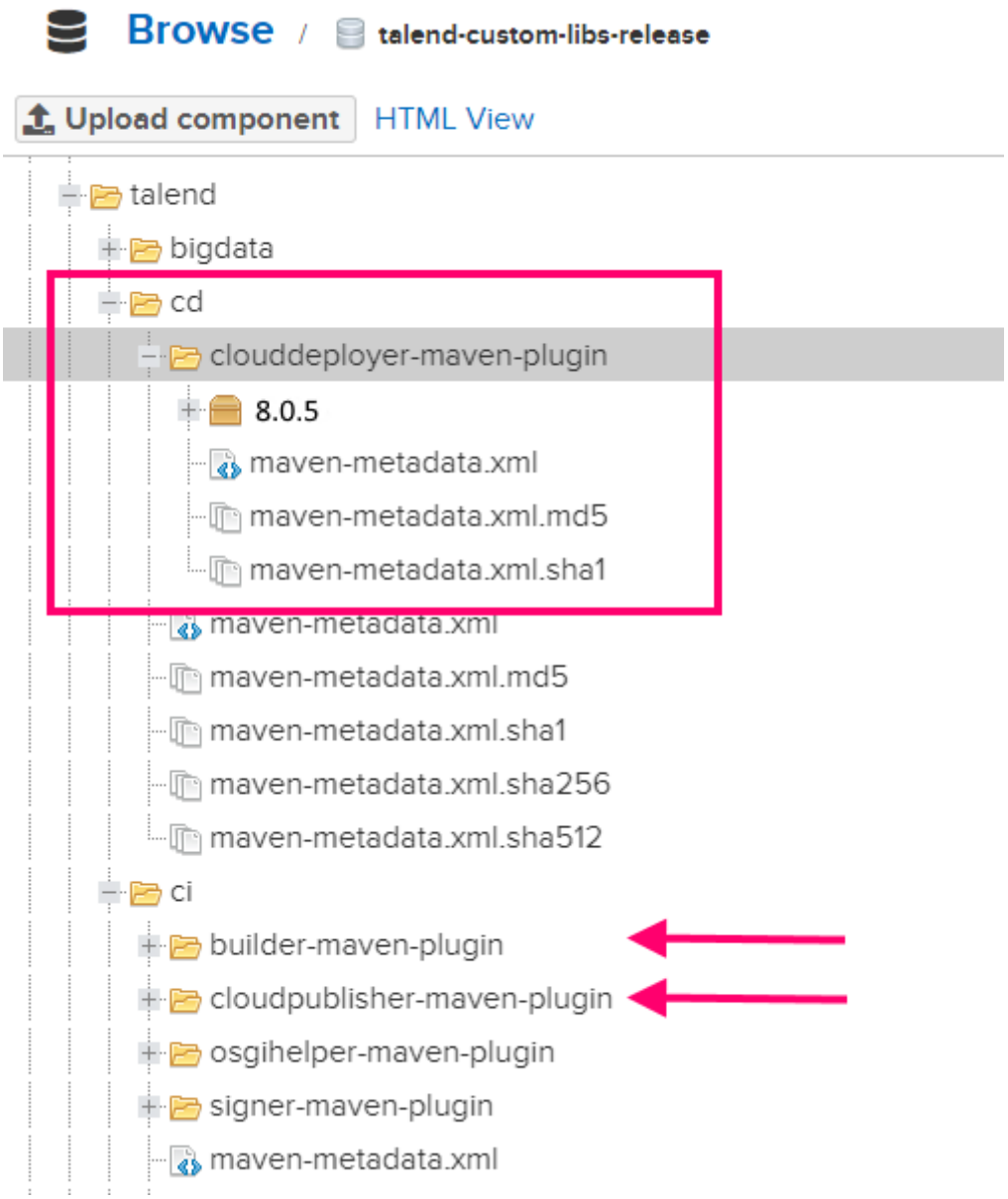
```
<?xml version="1.0" encoding="UTF-8"?>

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
```

```
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <!--  <localRepository></localRepository>-->
  <pluginGroups>
  </pluginGroups>

  <proxies>
  </proxies>

  <servers>
    <!-- Credential to access 'talend-custom-libs-release' repository -->
    <server>
      <id>talend-custom-libs-release</id>
      <username>admin</username>
      <password>Talend123</password>
    </server>

  </servers>

  <mirrors>
  </mirrors>

  <profiles>
  </profiles>

  <activeProfiles>
  </activeProfiles>
</settings>
```

clouddeployer GAV

```
<groupId>org.talend.cd</groupId>
<artifactId>clouddeployer-maven-plugin</artifactId>
<version>8.5.0</version>
```

`clouddeployer` shares similar groupId as the other Talend Maven plugins (notice the `cd` vs `ci` package).



Configuration

Plugin Properties

Property	Required	Default & valid Values	Description
cloud.domain	☑ Yes	*Values:* • AWS_AP • AWS_AU • AWS_EU • AWS_US • AZURE_US • AT	
cloud.token	☑ Yes		**Notes**: Basic Authentication with username and password is not supported. See 'Generating a Personal Access Token'
The following properties are the same as the ones used in `cloud-publisher` and used for the artifact publication.			
service.accelerate			For large generated artifact (Big Data), set this parameter to true to optimize the publication.
cloud.publisher.screenshot			Add this parameter with a true value if you want to publish the screenshot of generated artifact
cloud.publisher.remove-double-quotes			Set to true, `cloud-deployer` will remove any double-quotes characters around context variables of type String during the publication of the artifact. **This parameter doesn't exist in the original `cloud-publisher`.**

Property	Required	Default & valid Values	Description
publish-version			Allows to set a specific version for the artifact instead of the version computed by Talend Cloud. The value can be set as: • Free text: `1.2.3` or use the keywords: • `MODULE_VERSION`: the version will be the version of the Maven module of the project asset: `0.1`. • `USER_VERSION`: the version will be the deployment version set in the project asset. **Default**= null (Use Talend Cloud computed version) **This parameter doesn't exist in the original** `cloud-publisher`.
The following properties are all prefixed with `cloud.deployer.` **Only the final part of the property is listed below.**			
environment	☑ Yes		The environment must be the one where the artifact has been published.
workspace	☑ Yes		The workspace must be the one where the artifact has been published.
artifactVersion			The artifact version is the one published by this plugin. However, it is possible to skip the publication and set a specific version for the creation/update of the task. If the publication is skipped and the `artifactVersion` is not set, the latest published version is used and the task wil be set to always use the latest version (auto-updatable).

Property	Required	Default & valid Values	Description
taskName			If not provided, the task name will be the name of the generated artifact. Similar to 'cloud-publisher'. The value support template resolution, see the dedicated section of the Configuration template for more details.
taskDescription			The value support template resolution, see the dedicated section of the Configuration template for more details.
tags			'|' (pipe) delimited list of tag names. See: 'Managing tags' The value support template resolution, see the dedicated section of the Configuration template for more details.
connections			'|' (pipe) delimited list of: **<Application/System>=<Connection name>** **Examples**: cloud.deployer.connections="awss3=mybucket|ftp=myftp" See the list of supported connections and 'how to create connections'
resources			'|' (pipe) delimited list of: **<resource_file_xxxx>=<resource> <resource_directory_xxxx>=<resource>** **Examples**: cloud.deployer.resources="resource_file_job_config=prod_config" See 'Managing resources' for more details.

Property	Required	Default & valid Values	Description
parameters			'|' (Pipe) delimited list of: **<parameter name>=<parameter value>**. The <parameter name> can be in the form of **'parameter_XXXX'** for cloud parameters or any job context variable name: **"Advanced parameter"** as defined on Talend cloud. **Examples**: cloud.deployer.parameters="parameter_filename=myFilename.txt|max_row=10" See 'Naming user-defined variables' for more details.
engineType		**CLOUD** *Values:* • CLOUD • REMOTE_ENGINE • REMOTE_ENGINE_CLUSTER • CLOUD_EXCLUSIVE	
engineName			Valid for engineType: • REMOTE_ENGINE • REMOTE_ENGINE_CLUSTER
runProfile			Set the runProfile name defined for the selected REMOTE_ENGINE or REMOTE_ENGINE_CLUSTER.
logLevel		**WARN** *Values:* • OFF • INFO • WARN • ERROR	

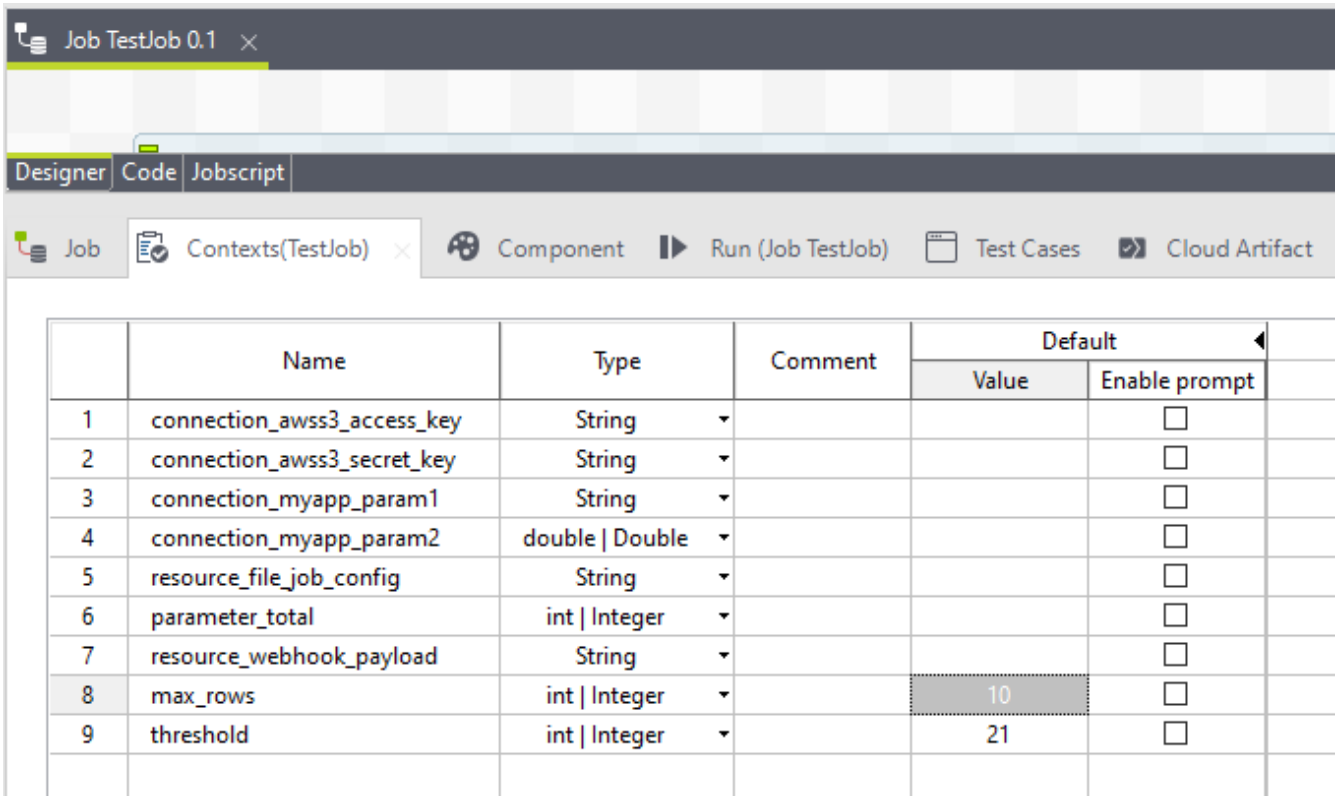
Property	Required	Default & valid Values	Description
autoUpdate		**false** *Values:* • true • false	If true, the task will be configured to always use the latest version of the associated published artifact. **This does not apply to data services or route artifacts**
parallelExecution		**false** *Values:* • true • false	If enabled, depending on the configuration and available tokens, it allows to: • Run the task simultaneously on two Remote Engines in the selected cluster. • Start two Cloud Engines to run the task on them in parallel.
impersonatedUser			Set the user executing the Job on REMOTE_ENGINE installed on Unix.
timeout			Custom Task timeout in minutes. The default value is 5 minutes and the minimum is 1.
triggers			Set a list (separated by |||) of a supported 'Trigger' type: • MANUAL • ONCE • DAILY • WEEKLY • MONTHLY • CRON • WEBHOOK See the dedicated section on how to set the value.+ If multiple triggers are set, a Schedule will be created.
scheduleId			Allows to attach an existing Schedule (list of associated Triggers) to the task. The **scheduleId takes precedence over triggers**. See documentation for more details

Property	Required	Default & valid Values	Description
configFile			All the above parameters can also be set via a configuration file. The value should be a path of the configuration path. See the dedicated section of the Configuration file for more details. The configFile parameter can be combined with the commandline maven parameters. If a parameter is defined twice, the commandline value takes precedence.
reportsDir			Directory path where the generated execution report will be saved. The generated file name for the report is based on the template: `cloud-deployer-report_<name of job/route_version>.txt`. The `.` in version is replaced with `_`. Ex: For the job `Job1_0.1` the generated report file is: `cloud-deployer-report_Job1_0_1.txt`

Usage

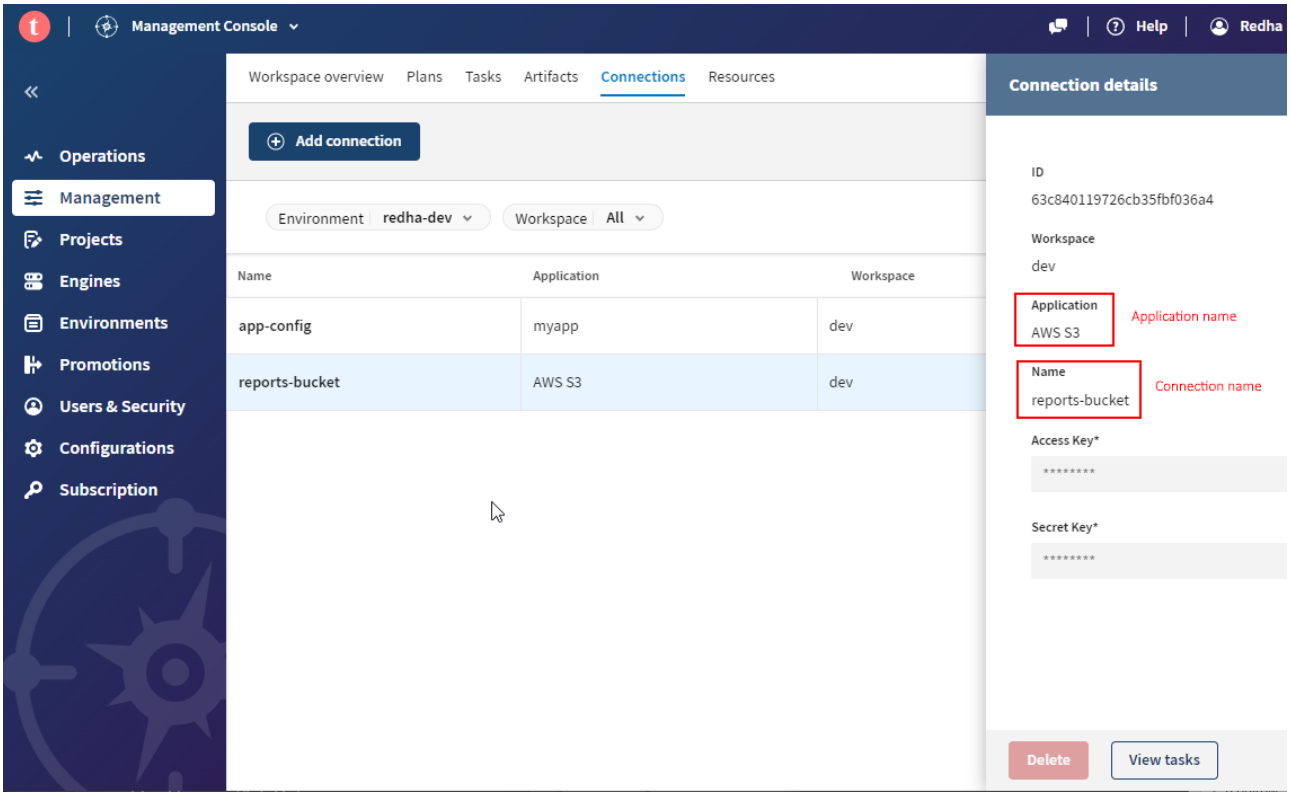
Job sample

The following section of the documentation will use a Talend project called `TALEND8TEST` with a simple DI Job `TestJob`, version `0.1` with the following context variables:

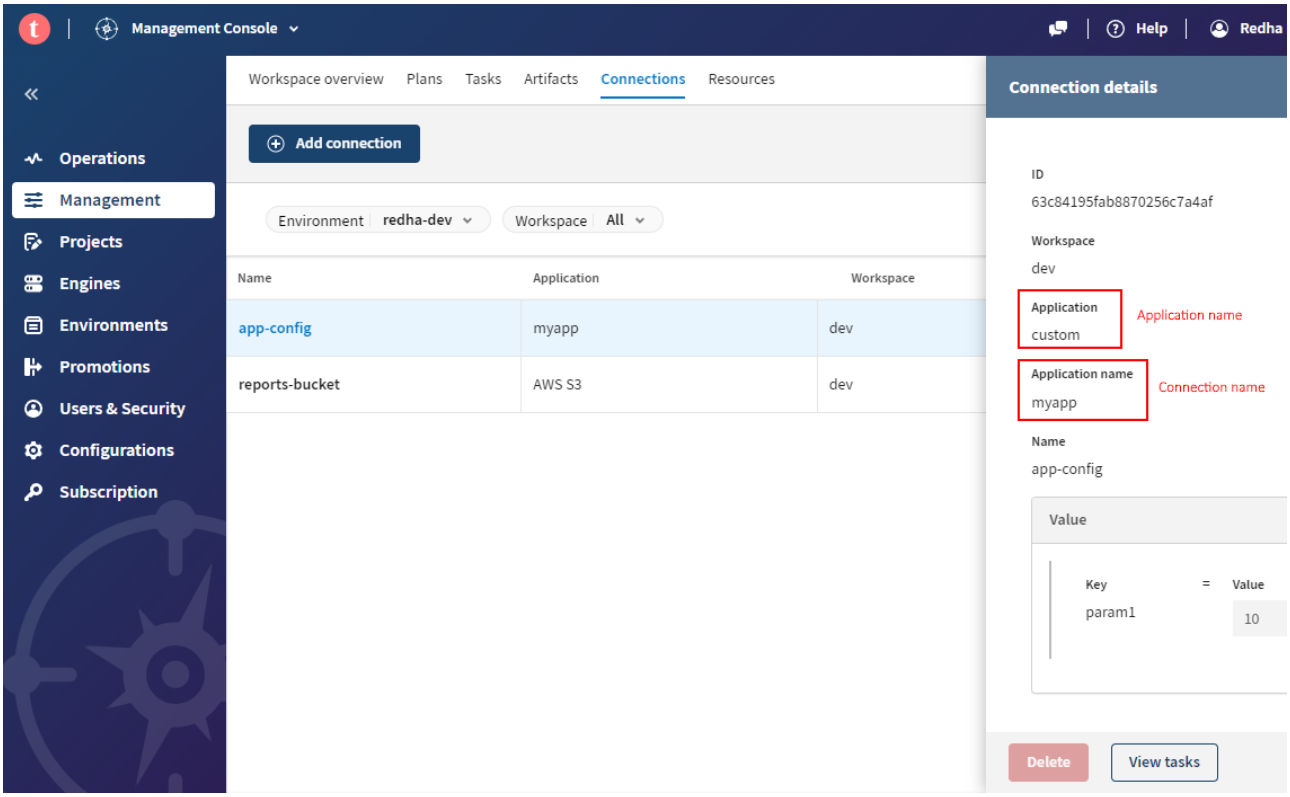


	Name	Type	Comment	Default	
				Value	Enable prompt
1	connection_awss3_access_key	String			☐
2	connection_awss3_secret_key	String			☐
3	connection_myapp_param1	String			☐
4	connection_myapp_param2	double | Double			☐
5	resource_file_job_config	String			☐
6	parameter_total	int | Integer			☐
7	resource_webhook_payload	String			☐
8	max_rows	int | Integer		10	☐
9	threshold	int | Integer		21	☐

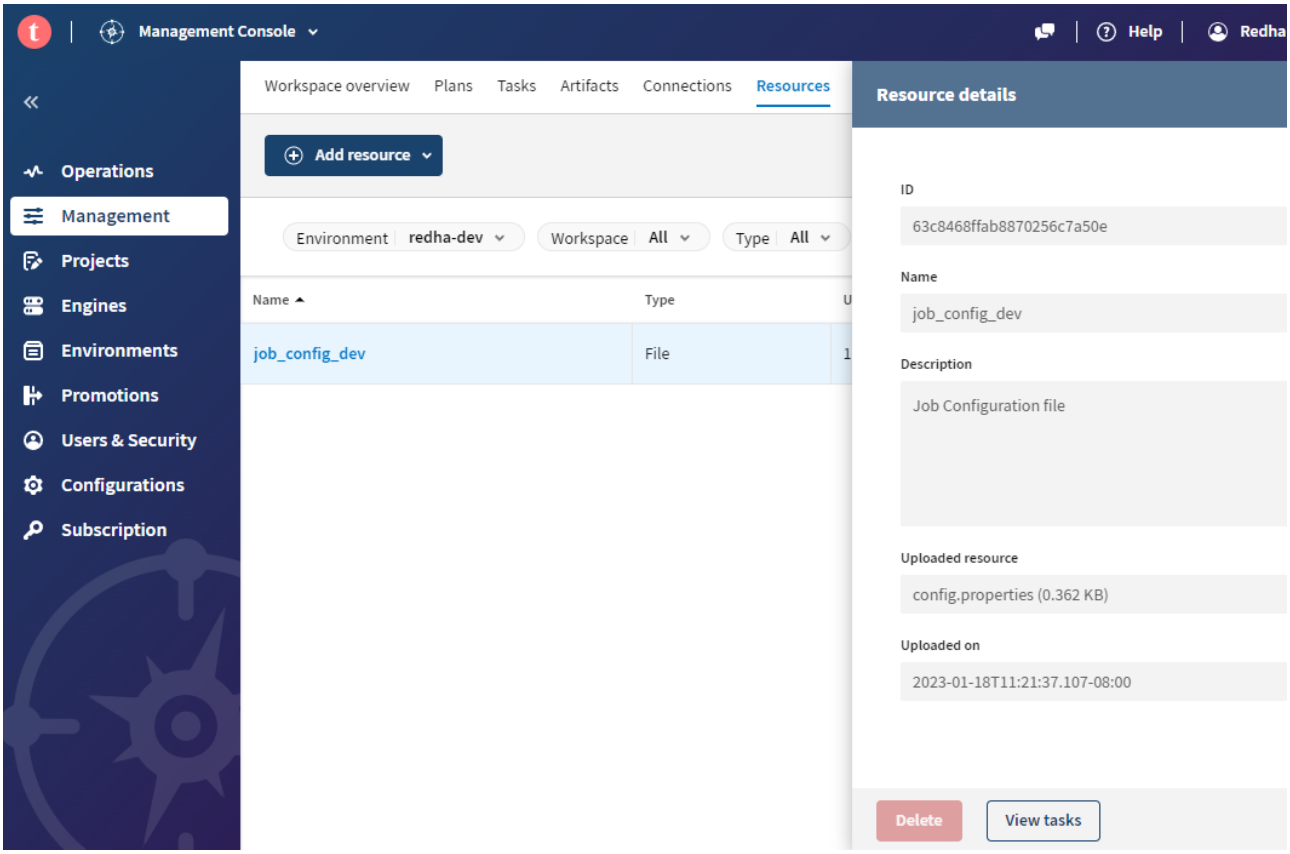
- The lines 1 & 2 define the parameters of the standard `AWS s3` connection with a TMC definition as:


Management Console
Help
Redha
Operations
Management
Projects
Engines
Environments
Promotions
Users & Security
Configurations
Subscription
Workspace overview
Plans
Tasks
Artifacts
Connections
Resources
Add connection
Environment redha-dev
Workspace All
Name
Application
Workspace
app-config
myapp
dev
reports-bucket
AWS S3
dev
Connection details
ID
63c840119726cb35fbf036a4
Workspace
dev
Application
AWS S3
Application name
Name
reports-bucket
Connection name
Access Key*
****
Secret Key*
****
Delete
View tasks

- The lines 3 & 4 define a custom connection with the application name `myapp`:



- Line 5 defines the resource file `my_config_file` with the TMC definition:



- The line 6 `parameter_total` defines a context variable exposed as Advanced parameter
- The line 7 `resource_webhook_payload` defines a context variable for the payload of a webhook. See the dedicated section of the 'Triggers' configuration for more details.

- The remaining variables are standard job context variables.

Maven commands

In this scenario, a single task will be created for the module/job passed as `-pl` parameter in the Maven command.

The associated Artifact will be determined based on the module definition.

Example: Create a task from a job/artifact (not all the following properties are required).

```
mvn -B \
 -s <path of settings.xml file> \  ①
 -f <path>/TALEND8TEST/poms/pom.xml \ ②
 -am -pl jobs/process/job1_0.1 \  ③
 -Dcloud.domain="AWS_US" \  ④
 -Dcloud.token=<cloud token> \  ⑤
 -Dcloud.deployer.environment="redha-dev" \  ⑥
 -Dcloud.deployer.workspace="dev" \  ⑦
 -Dcloud.deployer.taskName="task_TALEND8TEST_TestJob" \  ⑧
 -Dcloud.deployer.taskDescription="Task created by the cloud-deployer plugin" \  ⑨
 -Dcloud.deployer.tags="cloud-deployer" \  ⑩
 -Dcloud.deployer.connections="awss3=reports-bucket|myapp=app-config" \  ⑪
 -Dcloud.deployer.parameters="parameter_total=50|max_rows=35" \  ⑫
 -Dcloud.deployer.resources="resource_file_job_config=job_config_dev" \  ⑬
 -Dcloud.deployer.engineType="REMOTE_ENGINE_CLUSTER" \  ⑭
 -Dcloud.deployer.engineName="redha-dev-cluster" \  ⑮
 -Dcloud.deployer.runProfile="rp-redha-dev-cluster" \ ⑯
 -Dcloud.deployer.logLevel="INFO" \  ⑰
 -Dcloud.deployer.parallelExecution=true \  ⑱
 -Dcloud.deployer.impersonatedUser="talend" \
 -Dcloud.deployer.timeout=3 \  ⑲
 -Dcloud.publisher.screenshot=true \ ⑳
org.talend.cd:clouddeployer-maven-plugin:8.5.0:publish-deploy
```

① Path of the Maven settings.xml file.

② Path of the Talend Project directory root pom.

③ Maven module name to create the task for: `jobs/process/job1_0.1`.
The module will be used to determine the corresponding artifact.
The parameter **cloud.deployer.artifactVersion** is not used here. `clouddeployer` will search & use the latest version of the artifact.

④ Talend Cloud domain/region.

⑤ Talend Cloud token.

⑥ Cloud environment where the task will be created/updated.

⑦ Cloud workspace where the task will be created/updated.

⑧ Task name. It this example, it is based on the project name and job name.

⑨ Task description.

⑩ Optional task tags. They will be created if they don't exist.

⑪ List of the two Cloud connections to use in the Task.

⑫ Initialization of a list of advanced parameters and context variables.

⑬ Cloud Resource used by the task.

⑭ Cloud engine type. In this example, Remote Engines Cluster (Default: CLOUD).

⑮ Name of the Cloud engine type.

⑯ Engine Cluster runProfile name to set. The run profile must already exist.

⑰ Set task logLevel to INFO (Default: WARN).

⑱ Allows parallel executions (Default: false).

⑲ Custom task timeout of 3 minutes.

⑳ cloud-publisher parameter sued for the publication.

Cloud-deployer Maven goal to invoke.

Maven Output

```
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] TALEND8TEST Codes Master                                           [pom]
[INFO] TALEND8TEST Routines                                               [jar]
[INFO] TALEND8TEST TestJob-0.1.1 (0.1,Job Designs)                        [jar]
[INFO]
[INFO] -----------------< org.example.theonering:code.Master >-----------------
[INFO] Building TALEND8TEST Codes Master 8.5.0                            [1/3]
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] --- clouddeployer-maven-plugin:8.5.0:publish-deploy (default-cli) @ code.Master
---
[INFO] Skipping non-job artifact deployment
[INFO]
[INFO] ----------------< org.example.theonering.code:routines >----------------
[INFO] Building TALEND8TEST Routines 8.5.0                                [2/3]
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clouddeployer-maven-plugin:8.5.0:publish-deploy (default-cli) @ routines
---
[INFO] Skipping non-job artifact deployment
[INFO]
[INFO] ------------------< org.example.theonering.job:Job1 >-------------------
[INFO] Building TALEND8TEST Job1-0.1.1 (0.1,Job Designs) 0.1.1            [3/3]
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clouddeployer-maven-plugin:8.5.0:publish-deploy (default-cli) @ Job1 ---
[INFO] Properties:                              ①
[INFO]  - Cloud domain: AWS_US
[INFO]  - Auth method: Token
[INFO]  - Environment: redha-dev
[INFO]  - Workspace: dev
[INFO]  - Artifact version: Latest
[INFO]  - Task Name: task_TALEND8TEST_TestJob
[INFO]  - Task Descriptions: 'Task created by the cloud-deployer plugin'
[INFO]  - Tags:
[INFO]    - by-cloud-deployer
[INFO]  - Connections:                          ②
[INFO]    - awss3=reports-bucket
[INFO]    - myapp=app-config
[INFO]  - Parameters:
[INFO]    - parameter_total=50
[INFO]    - max_rows=35
[INFO]  - Resources:
[INFO]    - resource_file_job_config=job_config_dev
[INFO]  - engineType: REMOTE_ENGINE_CLUSTER
```

```
[INFO]  - RuntimeName: redha-dev-cluster
[INFO]  - Engine run profile: rp-redha-dev-cluster
[INFO]  - LogLevel: INFO
[INFO]  - Parallel Execution: true
[INFO]  - Impersonated User: talend
[INFO]  - Task timeout: 3
[INFO]  - Triggers:
[INFO] ----------------------------------------
[INFO] RelativePath: process
[INFO] StartingDir: /home/redha/cloud-deployer/workspace/TALEND8TEST/process
[INFO] Search Artifact: TestJob
[INFO]   - Published Artifact id: 5f7501e612837d7d45eb7da9
[INFO] Search Task: task_TALEND8TEST_TestJob
[INFO]   - Task does not exist => Create
[INFO]   - Task 'task_TALEND8TEST_TestJob' - version: 0.1
[INFO] Runtime configuration
[INFO]   - Requested task runtime type: REMOTE_ENGINE_CLUSTER
[INFO] Trigger configuration
[INFO]   - No schedule setup requested
[INFO]   - No trigger setup requested
[INFO] ----------------------------------------
[INFO] Cloud Deployer done. Report:
[INFO]  - Artifact:                        ③
[INFO]    - Name: TestJob
[INFO]    - Id: 62f83eaf7ab6e017d585d1fc
[INFO]    - Version: 0.1.40.20231801070314
[INFO]    - Url: https://tmc.us.cloud.talend.com/manage-
artifacts/6207f129ce4cda223c78f732/workspace/6207f129ce4cda223c78f733/detail/job/62f83
eaf7ab6e017d585d1fc
[INFO]  - Task:                            ④
[INFO]    - Name: task_TALEND8TEST_TestJob
[INFO]    - Id: 63c8a7e9fab8870256c7a73b
[INFO]    - Version: 0.1
[INFO]    -  https://tmc.us.cloud.talend.com/tasks-and-plans-
administration/6207f129ce4cda223c78f732/6207f129ce4cda223c78f733/tasks/job/63c8a7e9fab
8870256c7a73b/detail
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.495 s                              ⑤
[INFO] Finished at: 2023-07-03T10:00:20-07:00
[INFO] ------------------------------------------------------------------------
```

① The clouddeployer starts by listing all the input properties.

② The connections, parameters and resources parameter values are parsed and displayed as individual lists.

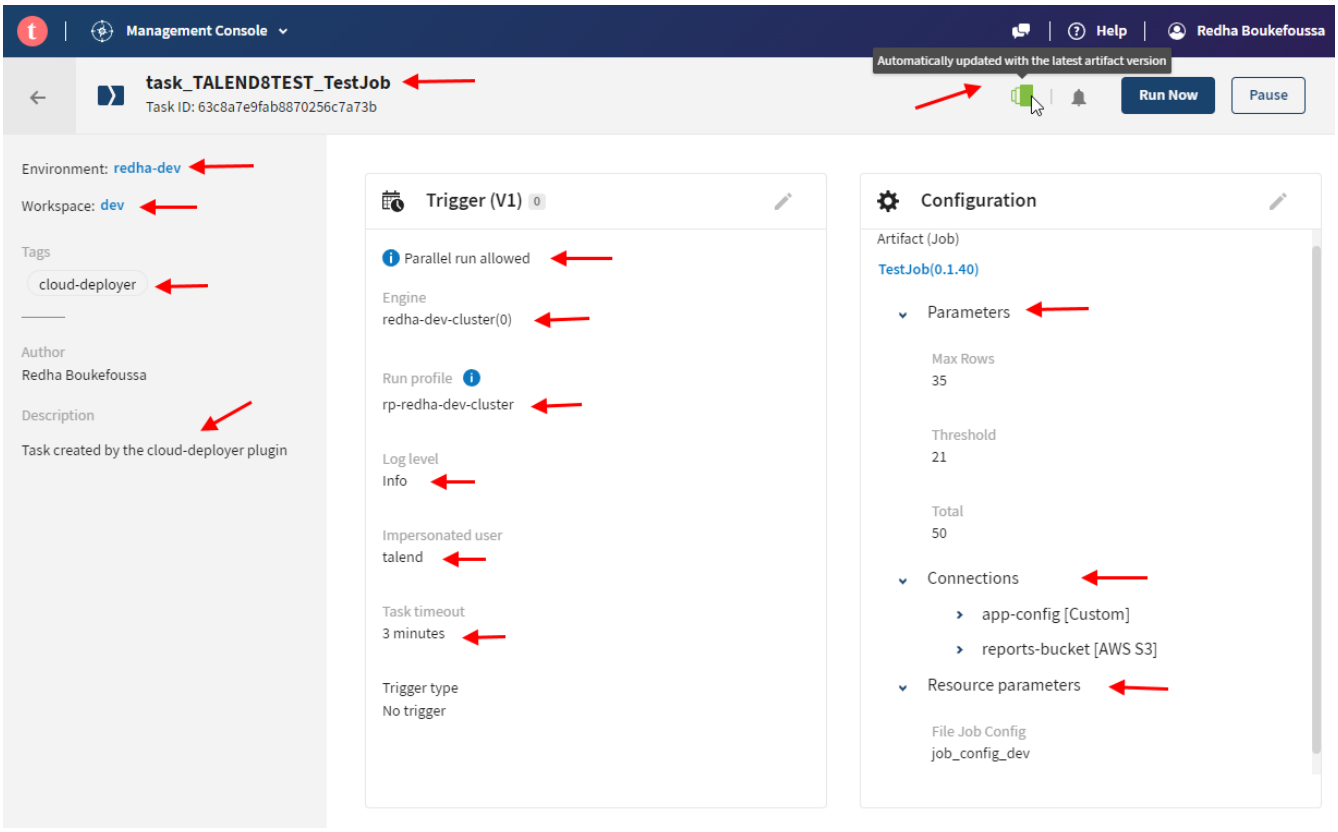
③ If successful, the execution ends with a report listing the artifact information, including a direct link to the definition in TMC.

④ As well as the task information, including a direct link to the definition in TMC.

⑤ As the processing involves primarily API calls, the processing should take a handful of seconds.

TMC Result

As we can see below, every aspect of the task can be configured with the `clouddeployer` Maven plugin.



To configure the Task schedule/trigger, see the dedicated section.

Task 'Triggers' Configuration

This following section describes the usage of the property: `cloud.deployer.triggers` and its value naming convention
to add any available Task 'triggers` (up to 15) available on TMC.

Each trigger definition is separated by `|||` (3)

Template: `cloud.deployer.triggers=<trigger #1>|||<trigger #2>`

Ex: 2 triggers. First of type ONCE and the second of type CRON

```
cloud.deployer.triggers="ONCE|once|2023-01-30 10:00|||CRON|cron trigger|2023-01-30|15 10 ? * 6L
*"
```

Each 'trigger' option is documented below.

The first 2 options `type` & `name` are mandatory for all triggers.

For 'Triggers' using a time zone; cloud-deployer sets the default value as the timezone where it is executed.

Options

Manual

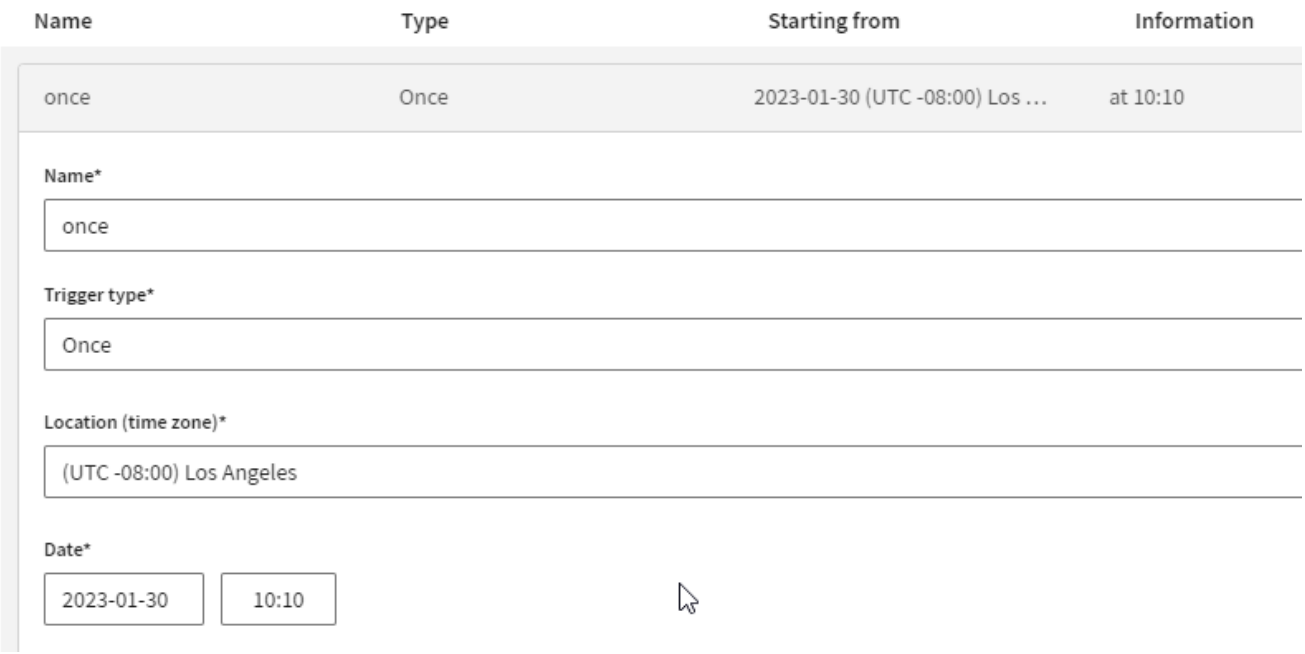
Type	**trigger**
Manual	Default value
Example: Do not set or set it to blank: cloud.deployer.triggers=""	

Once

Type	trigger
Once	*Template*: ONCE|<Name>|<Start date & time> *Format*: **ONCE|Text|yyyy-MM-dd HH:mm**

Example:
cloud.deployer.triggers="**ONCE|once|2023-01-30 10:00**"
(Once, starting on Jan 30th 2023 at 10:00 AM.)

In TMC:

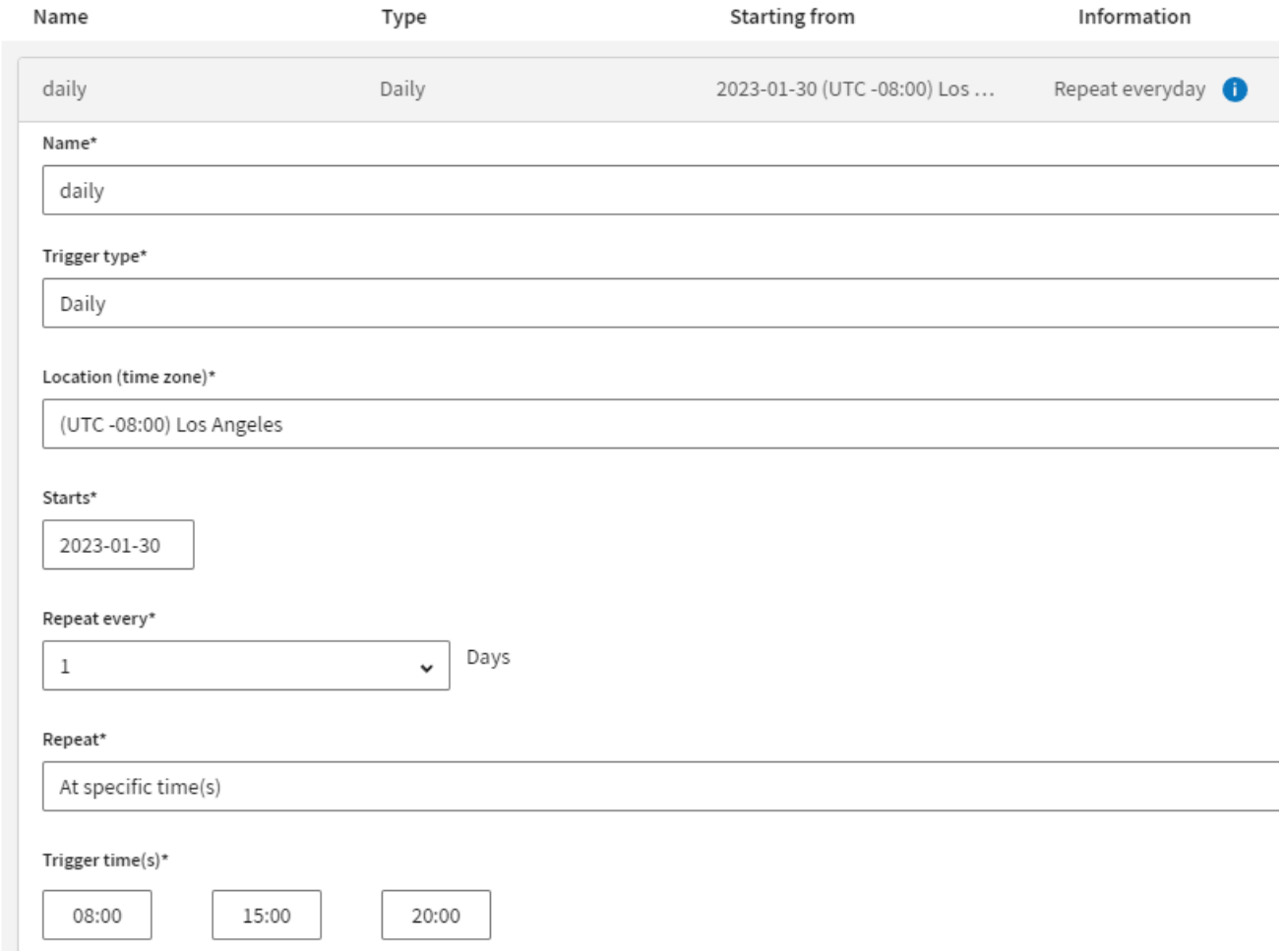


Daily

Type	trigger
Daily **(At Times)**	*Template*: DAILY|<Name>|<Start date>|<Days repeat>|TIMES|<time #1>,<time #n> *Format*: **DAILY|Text|yyyy-MM-dd|Number|TIMES|HH:mm,HH:mm, ...**

Example:
cloud.deployer.triggers="**DAILY|daily|2023-01-30|2|TIMES|08:00,15:00,20:00**"
(Every two days starting on Jan 30th 2023 at 08:00 AM, 03:00 PM & 08:00 PM)

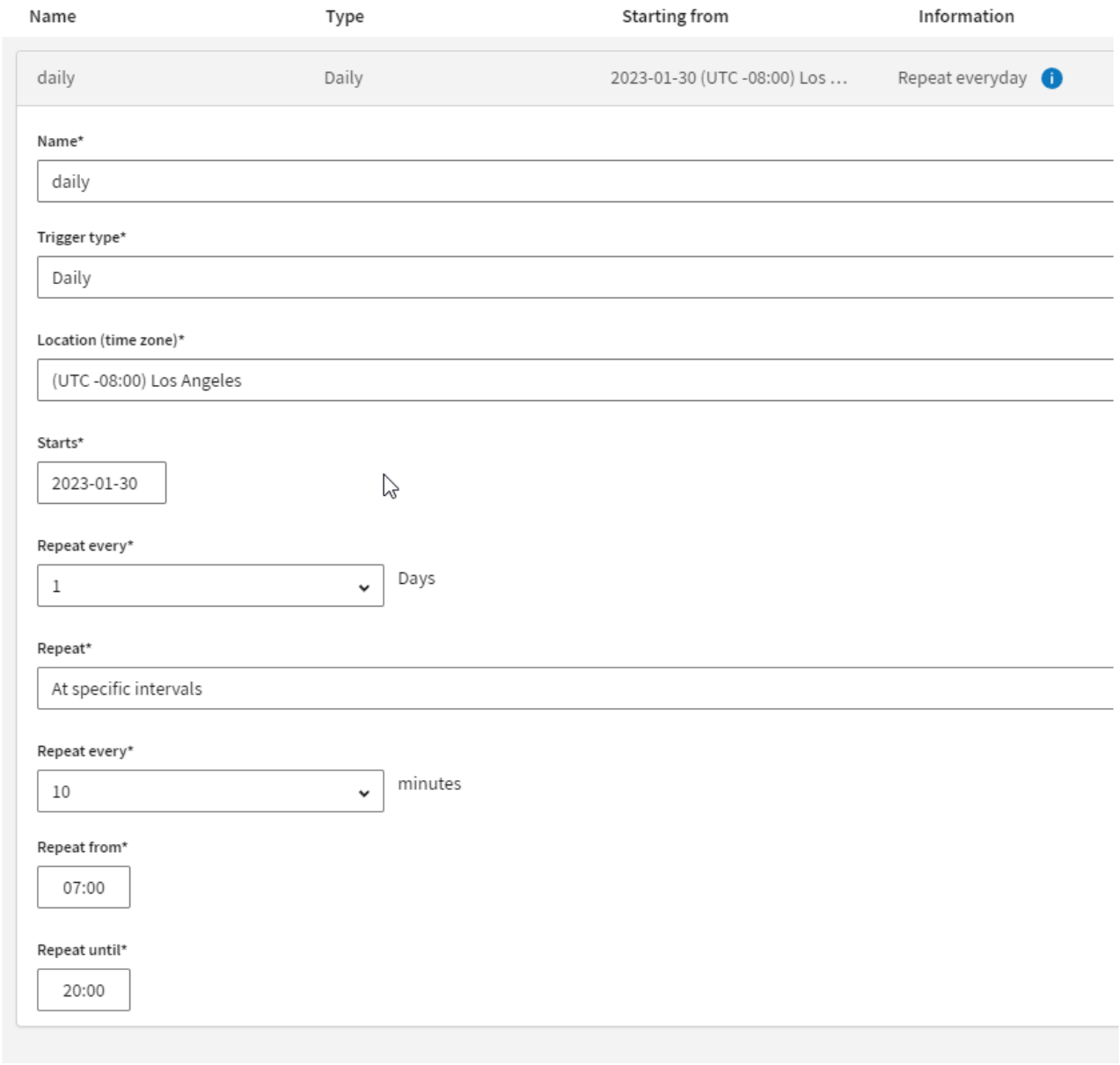
In TMC:



Type	trigger
Daily (Intervals)	*Template*: DAILY|<Name>|<Start date>|<Days repeat>|INTERVALS|<start time>|<Minutes repeat>|<end time> *Format*: **DAILY|Text|yyyy-MM-dd|Number|INTERVALS|HH:mm|Number|HH:MM**

Example:
cloud.deployer.triggers="**DAILY|daily|2023-01-30|1|INTERVALS|08:00|10|20:00**"
(Every day starting on Jan 30th 2023 at 08:00 AM until 08:00 PM every 10 minutes)

In TMC:

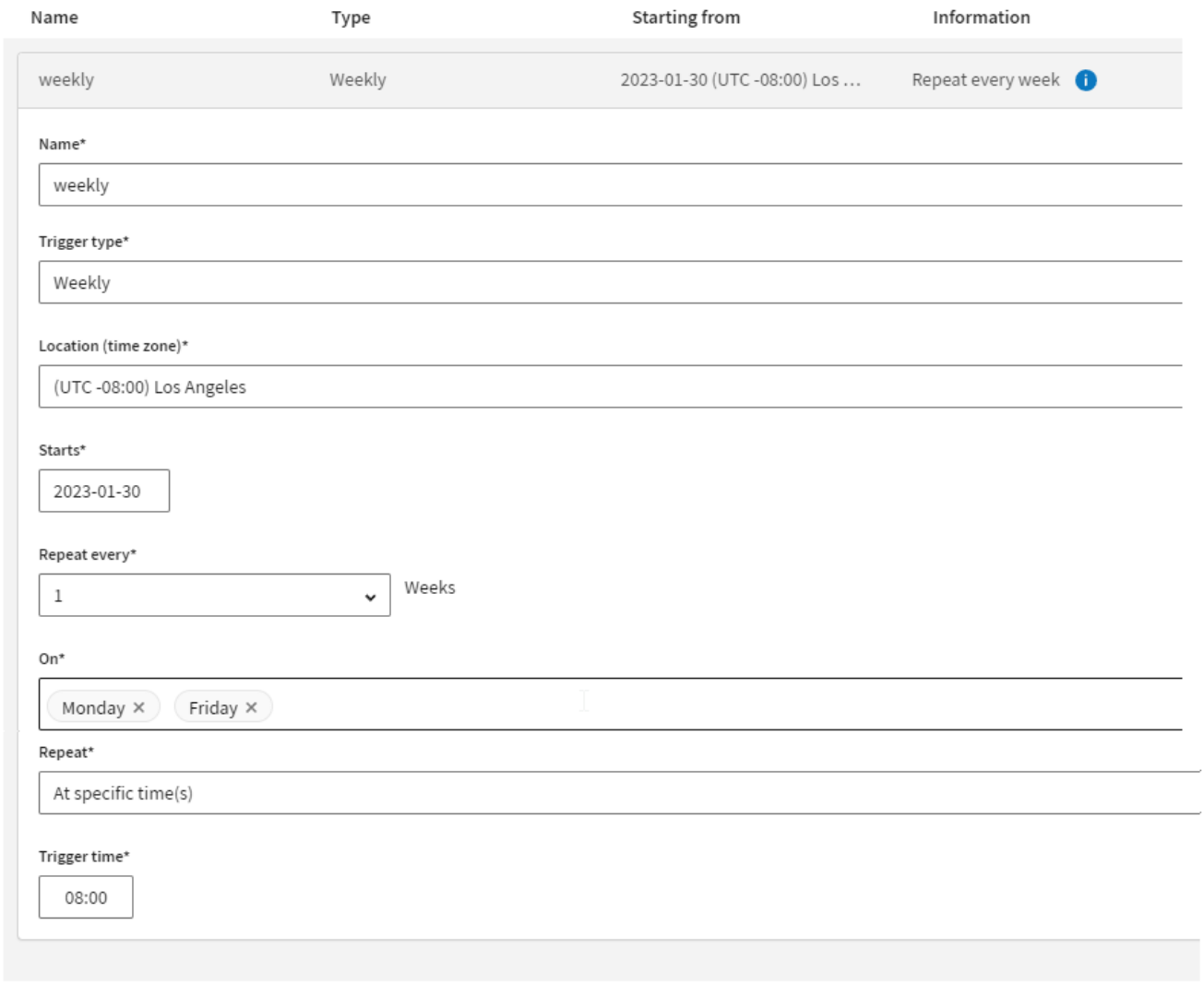


Weekly

Type	trigger
Weekly (At Times)	*Template*: WEEKLY|<Name>|<Start date>|<Week repeat>|TIMES|<start time>|<week day#1>,..., <week day#n> *Format*: **WEEKLY|Text|yyyy-MM-dd|Number|TIMES|HH:mm|<long format week day>,...**

Example:
cloud.deployer.triggers="**WEEKLY|weekly|2023-01-30|1|TIMES|08:00|MONDAY,FRIDAY**"
(Every week starting on Jan 30th 2023 at 08:00 AM on Monday and Friday)

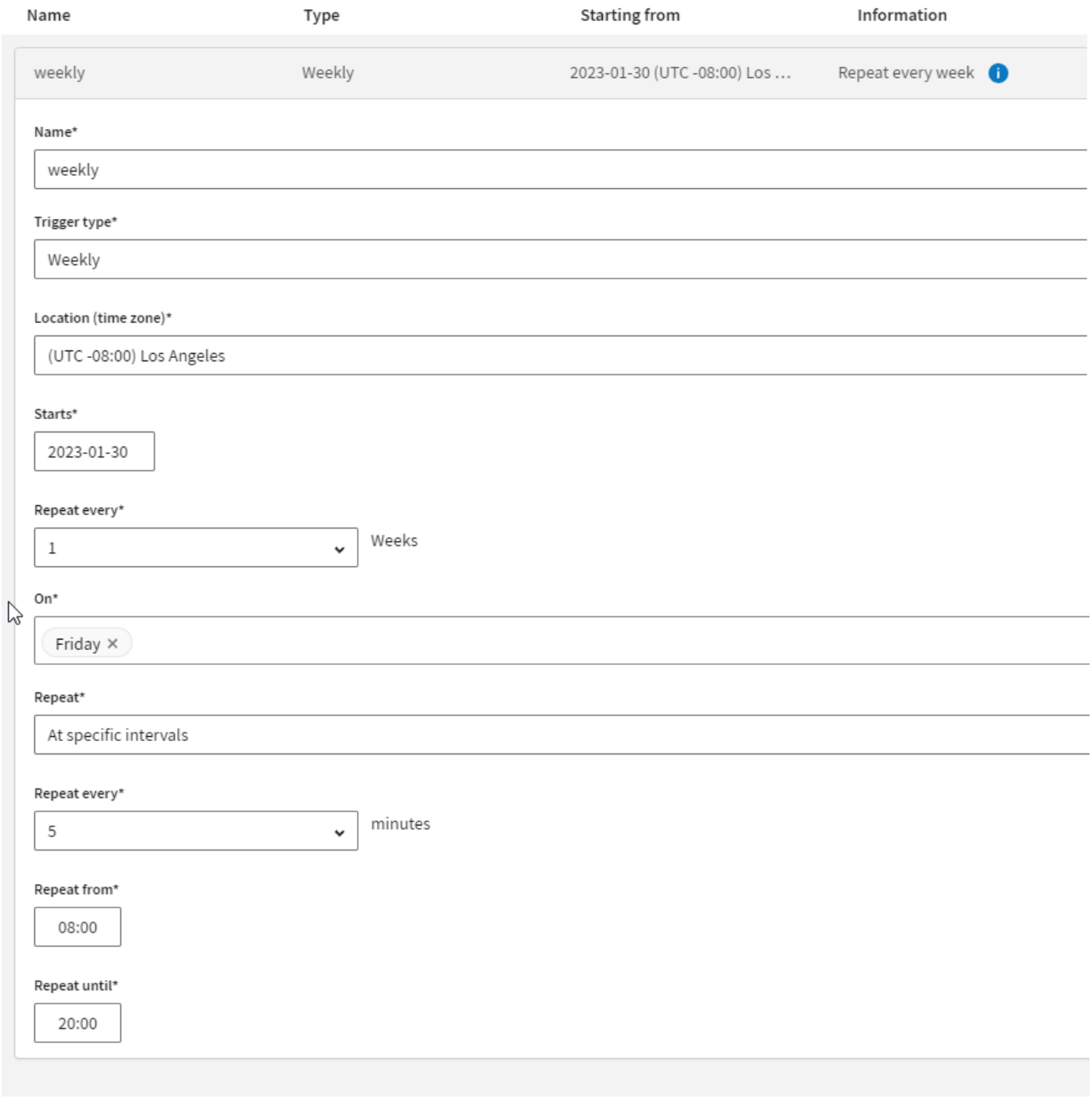
In TMC:



Type	trigger
Weekly (Intervals)	*Template*: WEEKLY|<Name>|<Start date>|<Days repeat>|INTERVALS|<start time>|<Minutes repeat>|<end time>|<week day#1>, ..., <week day#n> *Format*: **WEEKLY|Text|yyyy-MM-dd|Number|INTERVALS|HH:mm|Number|HH:MM|<long format week day>,...**

Example:
cloud.deployer.triggers="**WEEKLY|weekly|2023-01-30|1|INTERVALS|08:00|5|20:00|FRIDAY**"
(Every week starting on Jan 30th 2023 at 08:00 AM until 08:00 PM every 5 minutes on Friday)

In TMC:

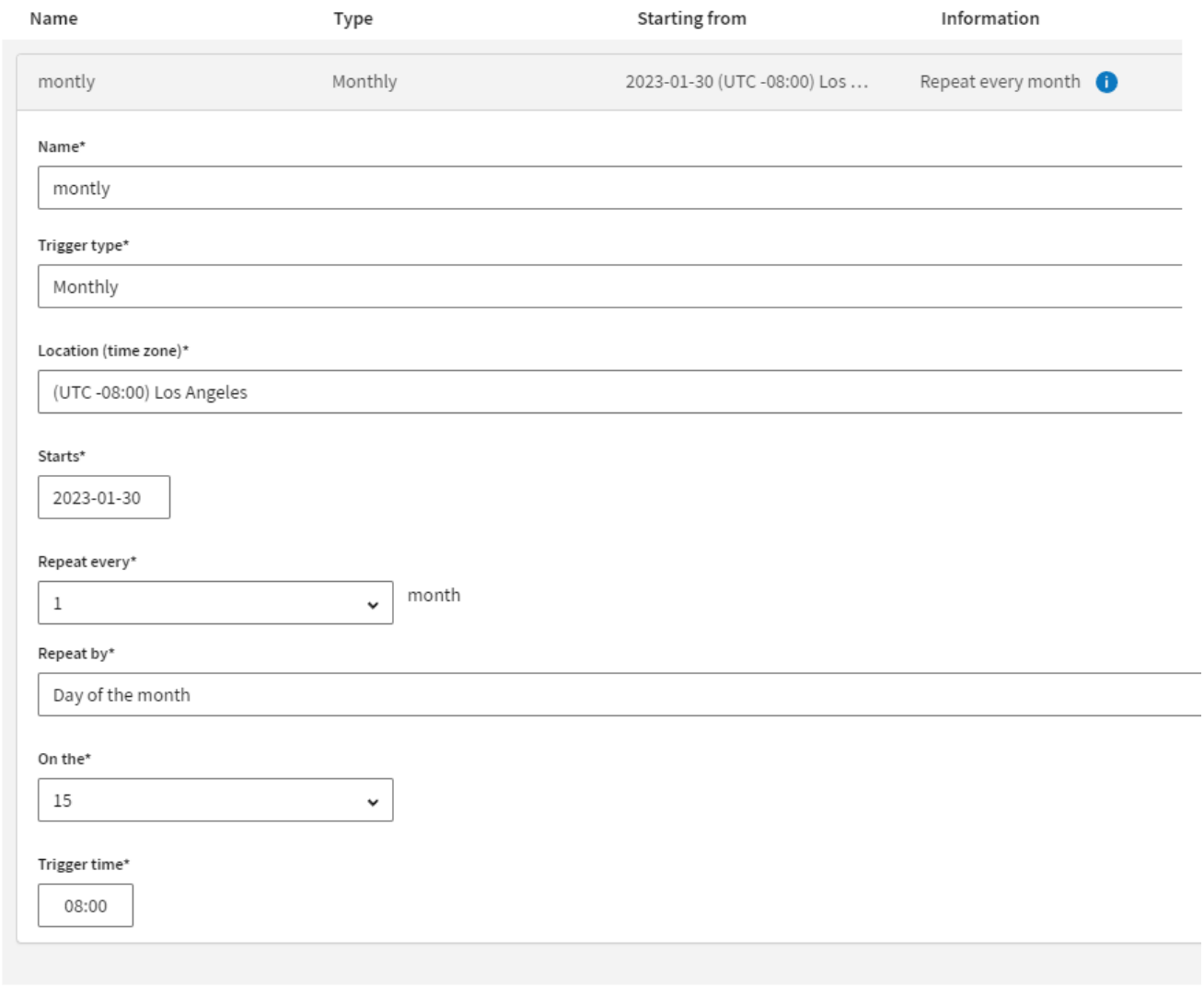


Monthly

Type	trigger
Monthly (Day of the Month)	*Template*: MONTHLY|<Name>|<Start date>|<Month repeat>|DAY_OF_MONTH|<start time>|<day of the month> *Format*: **MONTHLY|Text|yyyy-MM-dd|Number|DAY_OF_MONTH|HH:mm|Number**

Example:
cloud.deployer.triggers="**MONTHLY|monthly|2023-01-30|1|DAY_OF_MONTH|08:00|15**"
(Every month, on the 15, starting on Jan 30th 2023 at 08:00 AM)

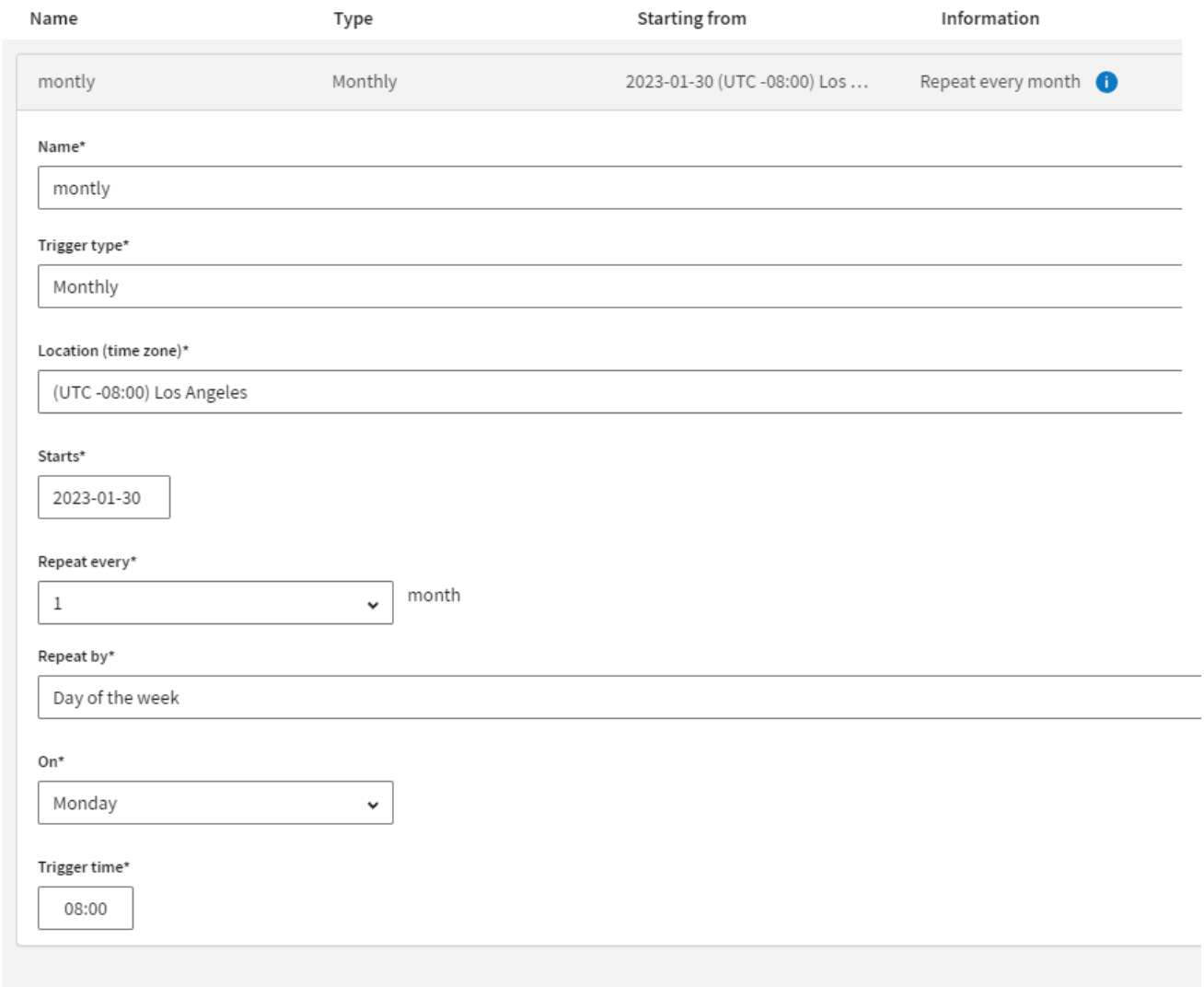
In TMC:



Type	trigger
Monthly **(Day of the week)**	*Template*: MONTHLY|<Name>|<Start date>|<Month repeat>|DAY_OF_WEEK|<start time>|<week day> *Format*: **MONTHLY|Text|yyyy-MM-dd|Number|DAY_OF_WEEK|HH:mm|<long format week day>,...**

Example:
cloud.deployer.triggers="**MONTHLY|monthly|2023-01-30|1|DAY_OF_WEEK|08:00|MONDAY**"
(Every month on Monday starting on Jan 30th 2023 at 08:00 AM)

In TMC:

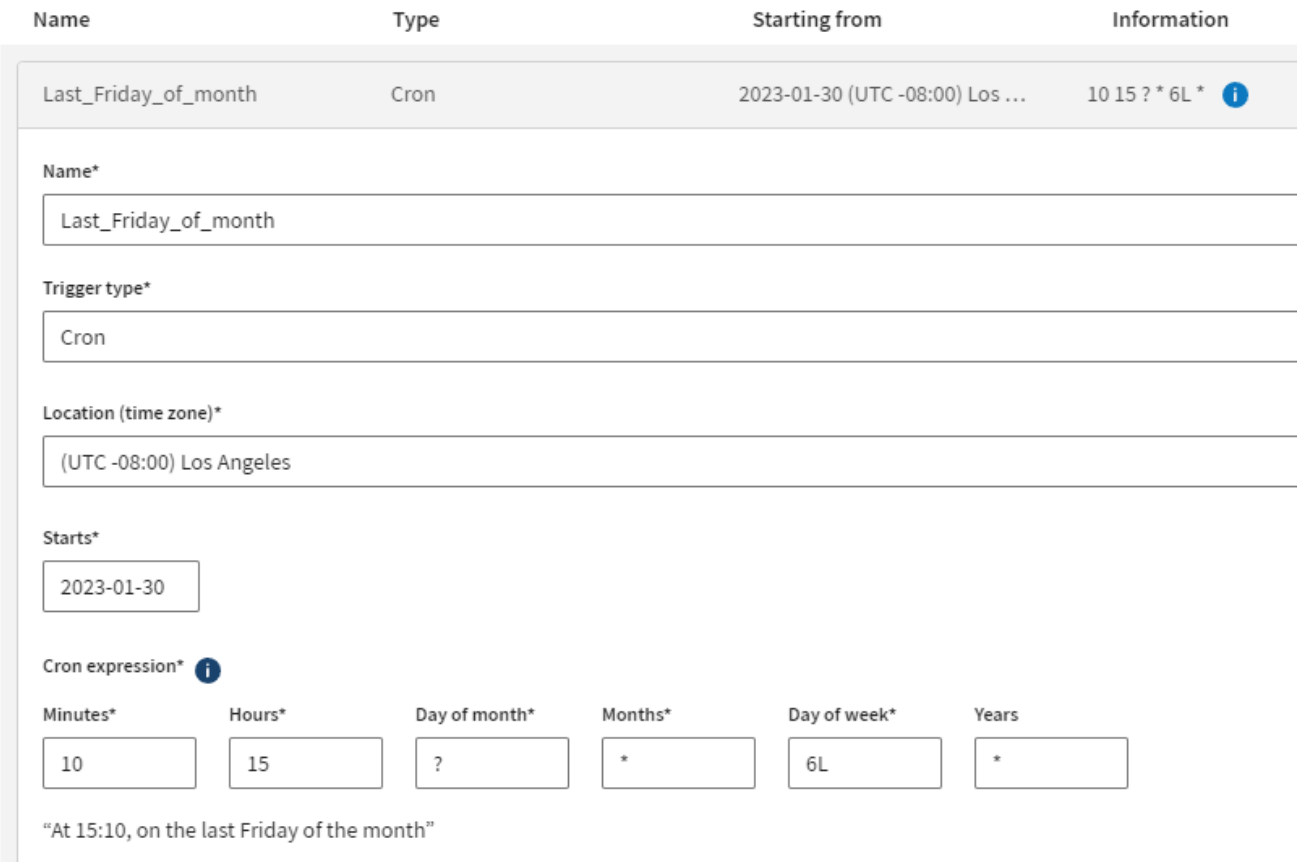


CRON

Type	trigger
CRON	*Template*: CRON|<Name>|<Start date>|<CRON expression> *Format*: **CRON|Text|yyyy-MM-dd|Text** See 'Configuring a Cron trigger' for more details.

Example:
cloud.deployer.triggers="**CRON|Last_Friday_of_month|2023-01-30|15 10 ? * 6L**"
Execute the task at `10:15 am, on the last Friday of every month`

In TMC:



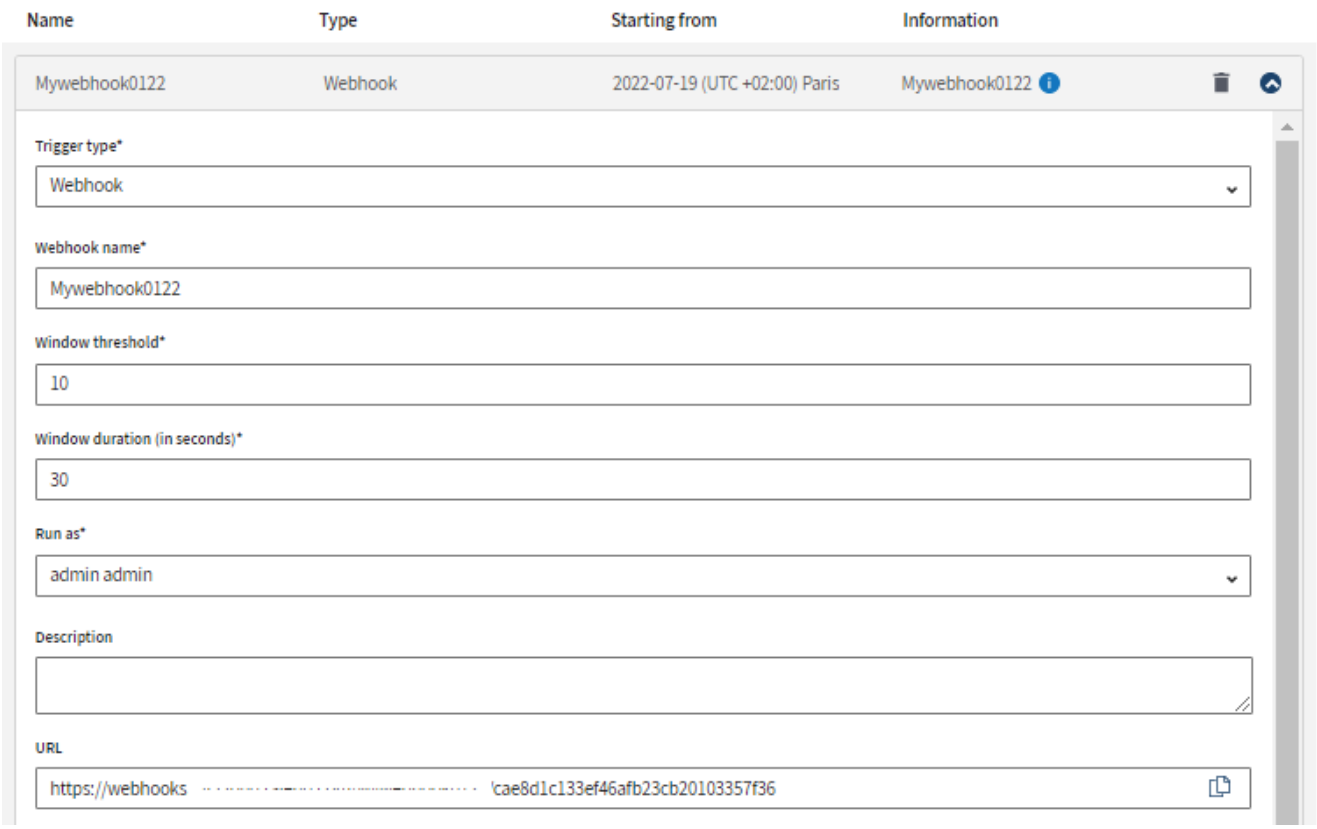
IMPORTANT: The **CRON expression is parsed**. If valid, it is translated in plain English and displayed in the log.
If invalid, the task is not be created/updated.

Webhook

Type	triggers
Webhook	*Template*: WEBHOOK|<Name>|WINDOW|<Window threshold>|<Window duration>|RUN_AS|<user>|<description> *Format*: **WEBHOOK|Text|WINDOW|Number|Number|RUN_AS|Text|Text** See 'Defining a Webhook parameter in Talend Studio' and 'Webhooks in Talend Cloud Management Console' for more details.

Example:
cloud.deployer.triggers="**WEBHOOK|MyWebHook0122|WINDOW|10|30|RUN_AS|admin admin|Task WebHook**"

In TMC:



IMPORTANT: WebHooks and any other trigger type described above are exclusive and cannot be combined.

Usage

Maven Command

In this scenario we will be updating the triggers of the previously created task, hence only the required parameters are provided, but the trigger parameter can be set during the creation as well.

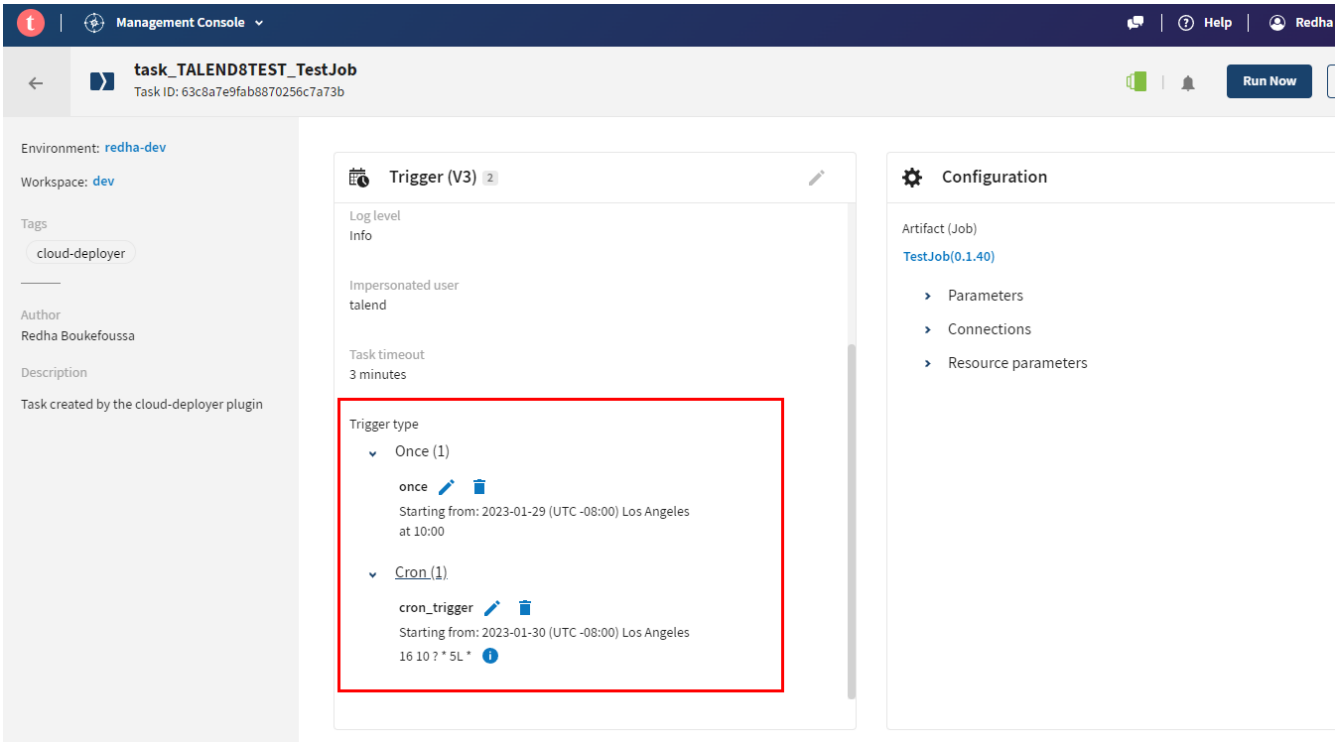
Example: triggers usage sample.

```
mvn -B \
 -s <path of settings.xml file> \
 -f <path>/TALEND8TEST/poms/pom.xml \
 -am -pl jobs/process/TestJob_0.1 \
 -Dcloud.domain="AWS_US" \
 -Dcloud.token=<cloud token> \
 -Dcloud.deployer.environment="redha-dev" \
 -Dcloud.deployer.workspace="dev" \
 -Dcloud.deployer.taskName="task_TALEND8TEST_TestJob" \
 -Dcloud.deployer.triggers="ONCE|once|2023-01-30 10:00|||CRON|cron trigger|2023-01-
30|15 10 ? * 6L *" \
org.talend.cd:clouddeployer-maven-plugin:8.5.0:publish-deploy
```

```
[INFO] -----------------< org.example.talend8test.job:TestJob >-----------------
[INFO] Building TALEND8TEST TestJob-0.1.0 (0.1,Job Designs) 0.1.0
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clouddeployer-maven-plugin:8.0.2:publish-deploy (default-cli) @ TestJob ---
[INFO] Properties:
[INFO]  - Cloud domain: AWS_US
[INFO]  - Auth method: Token
[INFO]  - Environment: redha-dev
[INFO]  - Workspace: dev
[INFO]  - Artifact version: Latest
[INFO]  - Task Name: task_TALEND8TEST_TestJob
[INFO]  - Task Descriptions: ''
[INFO]  - Tags:
[INFO]  - Connections:
[INFO]  - Parameters:
[INFO]  - Resources:
[INFO]  - engineType: not set. Default to CLOUD if create task
[INFO]  - RuntimeName:
[INFO]  - Engine run profile:
[INFO]  - LogLevel:
[INFO]  - Parallel Execution:
[INFO]  - Impersonated User:
[INFO]  - Task timeout:
[INFO]  - Triggers: ONCE|once|2023-01-29 10:00|||CRON|cron trigger|2023-01-30|16 10 ?
* 5L *
[INFO] ------------------------------------------------------------
```

```
[INFO] RelativePath: process
[INFO] StartingDir: /home/redha/cloud-deployer/workspace/TALEND8TEST/process
[INFO] Search Artifact: TestJob
[INFO]   - Published Artifact id: 62f83eaf7ab6e017d585d1fc
[INFO] Search Task: task_TALEND8TEST_TestJob
[INFO]   - Task already exist ==> Update
[INFO]   - Task 'task_TALEND8TEST_TestJob' - version: 0.1
[INFO] Runtime configuration
[INFO] Triggers and Schedule configuration
[INFO]   - No schedule setup requested
[INFO]   - 2 triggers setup requested
[INFO]   - Trigger #1
[INFO]     - Name: once - Type: ONCE
[INFO]     - Value: ONCE|once|2023-01-29 10:00
[INFO]   - Trigger #2
[INFO]     - Name: cron trigger - Type: CRON
[INFO]     - Value: CRON|cron trigger|2023-01-30|16 10 ? * 5L *
[INFO]     - Expression='16 10 ? * 5L *' => at 10:16 on the last Thursday of every
month
[INFO] ----------------------------------------
[INFO] Cloud Deployer done. Report:
[INFO]  - Artifact:
[INFO]    - Name: TestJob
[INFO]    - Id: 62f83eaf7ab6e017d585d1fc
[INFO]    - Version: 0.1.40.20231801070314
[INFO]    - Url: https://tmc.us.cloud.talend.com/manage-
artifacts/6207f129ce4cda223c78f732/workspace/6207f129ce4cda223c78f733/detail/job/62f83
eaf7ab6e017d585d1fc
[INFO]  - Task:
[INFO]    - Name: task_TALEND8TEST_TestJob
[INFO]    - Id: 63c8a7e9fab8870256c7a73b
[INFO]    - Version: 3.5
[INFO]    - Url: https://tmc.us.cloud.talend.com/tasks-and-plans-
administration/6207f129ce4cda223c78f732/6207f129ce4cda223c78f733/tasks/job/63c8a7e9fab
8870256c7a73b/detail
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.474 s
[INFO] Finished at: 2023-07-03T10:02:39-08:00
[INFO] ------------------------------------------------------------------------
```

TMC Result


Management Console
Help
Redha
task_TALEND8TEST_TestJob
Task ID: 63c8a7e9fab8870256c7a73b
Run Now
Environment: redha-dev
Workspace: dev
Tags
cloud-deployer
Author
Redha Boukefoussa
Description
Task created by the cloud-deployer plugin
Trigger (V3) 2
Log level
Info
Impersonated user
talend
Task timeout
3 minutes
Trigger type
Once (1)
once
Starting from: 2023-01-29 (UTC -08:00) Los Angeles
at 10:00
Cron (1)
cron_trigger
Starting from: 2023-01-30 (UTC -08:00) Los Angeles
16 10 ? * 5L *
Configuration
Artifact (Job)
TestJob(0.1.40)
Parameters
Connections
Resource parameters

Configuration template

The majority of the parameter values can be set as a template that will be resolved at runtime.

The template format is a set variables surrounded by `@`.

Ex `@var1@`.

There are 3 type of variables.

- Project related variables. Those are:
  - `@project.name@` : Current project name
  - `@project.name.lowercase@` : Current project name in lower-case
  - `@project.version@` : Current project/product version
  - `@job.name@` : job/route name currently processed by cloud-deployer
  - `@job.folder@` : job/route folder (based on Studio definition)
  - `@job.version@` : job/route version (in Studio)
  - `@job.context@` : job/route context group name
  - `@job.maven.gav@` : job/route Maven GAV (groupId:artifactId:version)
  - `@job.maven.group.id@` : job/route Maven group Id
  - `@job.maven.artifact.id@` : job/route Maven artifact Id
  - `@job.maven.version@` : job/route Maven version
  - `@git.branch@` : Git branch of the current project
  - `@git.commit.id@` : Git commit id
  - `@git.commit.author@` : Git commit author
  - `@git.commit.date@` : Git commit date
- Environment variables
  - Format: `@env.<env var name>@`. If available, the value of the environment variable will replace the placeholder template.
- Date &Time variables
  - `@now.date@`. Current date in local format.
  - `@now.time@`. Current time in local format.

Example:

```
export CLOUD_TOKEN=<token value>

...

mvn -B \
 -s <path of settings.xml file> \
 -f <path>/TALEND8TEST/poms/pom.xml \
 -am -pl jobs/process/TestJob_0.1 \
 -Dcloud.domain="AWS_US" \
 -Dcloud.token="@env.CLOUD_TOKEN@" \
 -Dcloud.deployer.environment="redha-dev" \
 -Dcloud.deployer.workspace="dev" \
 -Dcloud.deployer.taskName="task_@project.name.lowercase@_@job.name@" \
 -Dcloud.deployer.taskDescription="Created @now.data@ at @now.time@" \
org.talend.cd:clouddeployer-maven-plugin:8.5.0:publish-deploy
```

will be resolved as if it was

```
mvn -B \
 -s <path of settings.xml file> \
 -f <path>/TALEND8TEST/poms/pom.xml \
 -am -pl jobs/process/TestJob_0.1 \
 -Dcloud.domain="AWS_US" \
 -Dcloud.token="<token value>" \
 -Dcloud.deployer.environment="redha-dev" \
 -Dcloud.deployer.workspace="dev" \
 -Dcloud.deployer.taskName="task_talend8test_TestJob" \
 -Dcloud.deployer.taskDescription="Created 2023-07-03 at 03:14:56" \
org.talend.cd:clouddeployer-maven-plugin:8.5.0:publish-deploy
```

This template functionality can be useful, for instance, to set a naming convention for the `taskName` by leveraging the current project and job information.

Configuration File

The Cloud-deployer Maven plugin parameters can also be set via a configuration file with the parameter `cloud.deployer.configFile`.

The format is based on a simplified version of INI files.

The file name and extension are not validated. Ex: `cloud-deployer.config`, `config.ini`, `configuration.properties` are all valid names.

The plugin parameters are defined as a list of key/value pair. The difference with their commandline counterpart is that they don't use the prefix 'cloud.deployer.' and' 'cloud.' Hence, the commandline parameter `cloud.deployer.taskName` becomes `taskName` in the configuration file.

The values of any type, including String, should not be quoted (double or single quote).

Comments are supported with the character `#` starting the line.

Example: cloud-deployer config file: cloud-deployer.config

```
# -------------------
# Cloud deployer configuration
# -------------------

# Double quotes arround value are INVALID
domain="AWS_US"

token=@env.CLOUD_TOKEN@

# spaces at the start of the line and around = are VALID
  environment = redha-dev

workspace=dev # Comments at the end of the line are INVALID

taskName=task_@project.name.lowercase@_@job.name@
taskDescription=Created @now.data@ at @now.time@
tags=cloud-deployer
engineType=REMOTE_ENGINE_CLUSTER
engineName=redha-dev-cluster
logLevel=INFO
connections=awss3=reports-bucket|myapp=app-config
parameters=parameter_total=50|max_rows=35
resources=resource_file_job_config=job_config_dev
```

The additional feature the configuration file provides is the ability to overwrite the parameters based on the current processed Talend Job/Route.

Like any other Maven plugin, cloud-deployer Maven plugin is invoked for each Maven module set with the `-pl` parameter.

Example of multi modules Maven command

```
mvn -B \
 -s <path of settings.xml file> \
 -f <path>/TALEND8TEST/poms/pom.xml \
 -am \
 -pl jobs/process/TestJob_0.1,jobs/process/AWS/S3/ListBucket_0.1 \
 -Dcloud.deployer.configFile=<path>/cloud-deployer.config \
org.talend.cd:clouddeployer-maven-plugin:8.5.0:publish-deploy
```

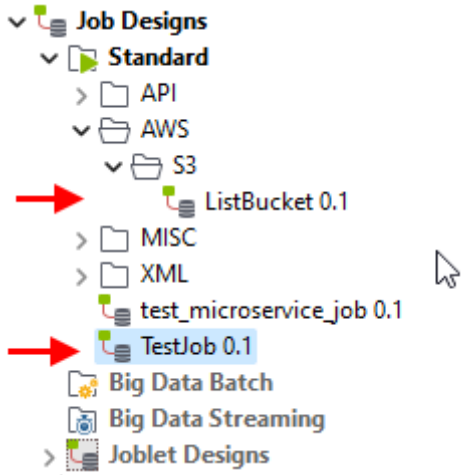
In the example above, Maven will invoke cloud-deployer twice. The first time for `jobs/process/TestJob_0.1` module and the second time for `jobs/process/AWS/S3/ListBucket_0.1` module.

The same property file will be used for both modules. With the support of template, describe above, this won't be a problem for the parameter `taskName` and `taskDescription`. However, for some parameters such as `connections`, `parameters` ,... this will generate a deployment error if the modules/jobs are not using the same values.

To support this scenario, the configuration file introduces the concept of `sections` to overwrite the default parameters for a specific Job/route.

A `section` is defined by the name/path of a Job/Route surrounded by brackets: Ex: `[<job/route name>]`.

The name/path is based on the location in Studio.



Example: cloud-deployer config file with a Job section

```
# -------------------
# Cloud deployer configuration
# -------------------
# Common configuration
# -------------------
domain=AWS_US
token=@env.CLOUD_TOKEN@

environment=redha-dev
```

```
workspace=dev

taskName=task_@project.name.lowercase@_@job.name@
taskDescription=Created @now.data@ at @now.time@

tags=cloud-deployer

engineType=REMOTE_ENGINE_CLUSTER
engineName=redha-dev-cluster

logLevel=INFO

# A section allows cloud-deployer to add and overwrite the common parameters defined
above
# with the parameters defined in this section, when it is exceuted for the module
'jobs/process/TestJob_0.1'
[TestJob_0.1]
connections=awss3=reports-bucket|myapp=app-config
parameters=parameter_total=10
resources=resource_file_job_config=job_config_dev
triggers=ONCE|once|2023-01-29 10:00|||CRON|cron trigger|2023-01-30|16 10 ? * 5L *

# This section will be ignored when executed for module 'jobs/process/TestJob_0.1'
# but used with module 'jobs/process/AWS/S3/ListBucket_0.1'
[AWS/S3/ListBucket_0.1]
taskDescription=List the entries of a S3 bucket
connections=awss3=reports-bucket
parameters=parameter_bucketName
```

In Maven command above, each module will be executed with a different set of parameters:

For module `jobs/process/TestJob_0.1`

```
domain=AWS_US
token=<token val>

environment=redha-dev
workspace=dev

taskName=task_talend8test_TestJob
taskDescription=Created 2023-07-03 at 03:14:56

tags=cloud-deployer

engineType=REMOTE_ENGINE_CLUSTER
engineName=redha-dev-cluster

logLevel=INFO

connections=awss3=reports-bucket|myapp=app-config
parameters=parameter_total=10
```

```
resources=resource_file_job_config=job_config_dev
triggers=ONCE|once|2024-01-29 10:00|||CRON|cron trigger|2024-01-30|16 10 ? * 5L *
```

For module `jobs/process/AWS/S3/ListBucket_0.1`

```
domain=AWS_US
token=<token val>

environment=redha-dev
workspace=dev

taskName=task_talend8test_ListBucket
taskDescription=List the entries of a S3 bucket

tags=cloud-deployer

engineType=REMOTE_ENGINE_CLUSTER
engineName=redha-dev-cluster

logLevel=INFO

connections=awss3=reports-bucket
parameters=parameter_bucketName
```

In the second module, note the default `taskDescription` value being overwritten by the corresponding section parameter, after all the template resolution:

Usage : GitOps

The configuration file can also be hosted in Git alongside the Talend project and referenced in the Maven command.

With this implementation, CI/CD tasks deployment can be configured, by the developer, during the development phase and automated at each Git push/pull-request, if necessary.

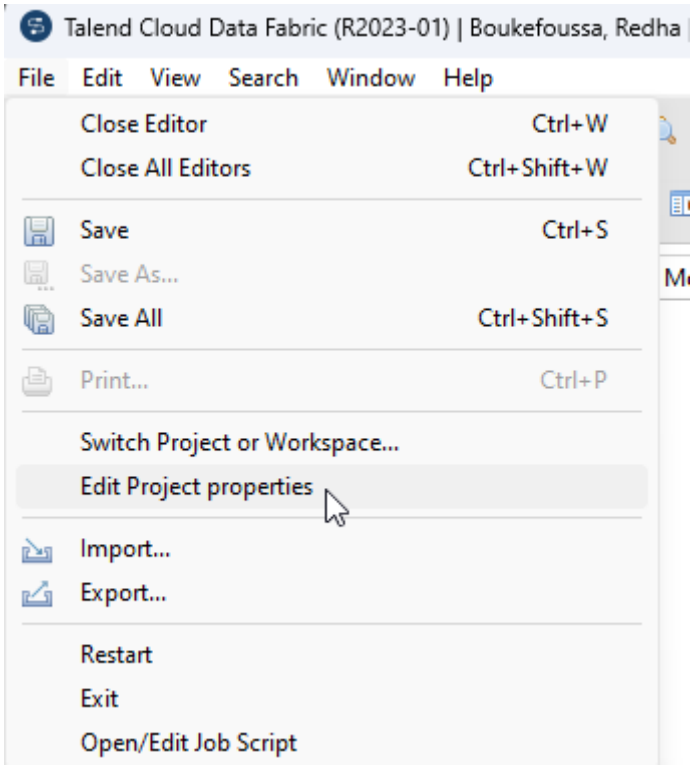


Figure 1. Project properties

Go to the panel of the Maven root pom template

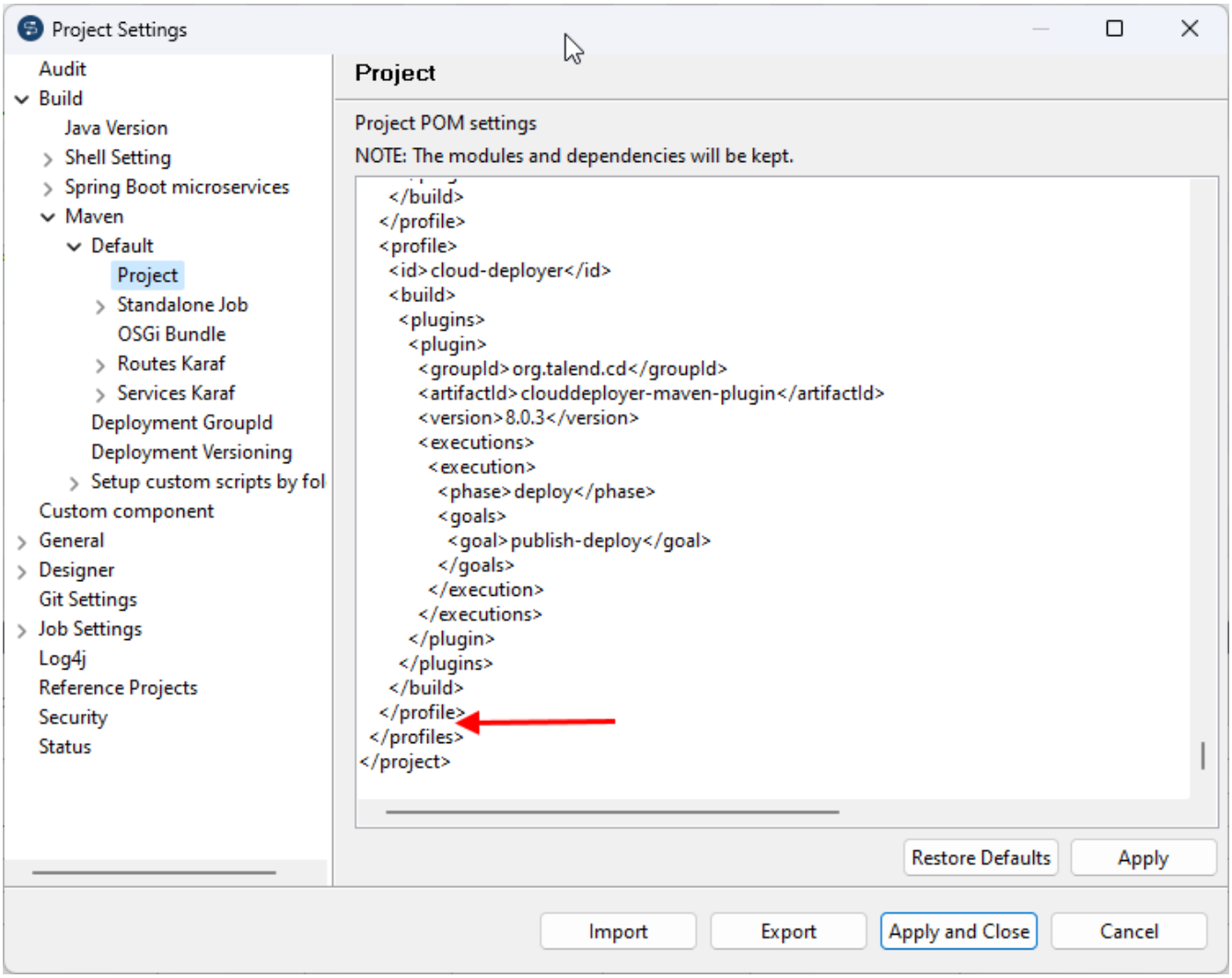


Figure 2. Project Root pom template

Jut before the closing tags: `</profiles>`, add the `cloud-deployer` dedicated profile:

```
    <profile>
      <id>cloud-deployer</id>
      <build>
        <plugins>
          <plugin>
            <groupId>org.talend.cd</groupId>
            <artifactId>clouddeployer-maven-plugin</artifactId>
            <version>8.5.0</version>
            <executions>
              <execution>
                <phase>deploy</phase>
                <goals>
                  <goal>publish-deploy</goal>
                </goals>
              </execution>
            </executions>
          </plugin>
        </plugins>
      </build>
```

```
    </profile>
```

The `cloud-deployer` profile is associated with the Maven `deploy` phase, as `cloud-publisher`.

Publication & Deployment

The main use case for `cloud-deployer` is the execution during the Continuous Deployment (CD) phase when all the artifacts have been published to Talend Cloud (TMC).

With support of the artifact publication, however, `cloud-deployer` can publish the current generated artifact, retrieve its published Id and version and create/update the task for it in the same invocation.

To automate this process that includes also the generation (`ci-builder`), a new profile can be added to the template project root pom.

Go to Studio and update the current root pom from the menu `File/Edit Project properties`

Example: `cloud-deployer` *profile usage.*

```
# Generate, package, publish and deploy (task management) in one Maven command

# This example uses the a cloud-deployer config file,
# but individual parameters are supported as well.

mvn -B \
 -s <path of settings.xml file> \
 -f <path>/TALEND8TEST/poms/pom.xml \
 -amd -pl jobs/process/TestJob_0.1 \
 -Dgeneration.type=local \              ①
 -Pcloud-deployer \                     ②
 -Dcloud.deployer.configFile=<path>/cloud-deployer.config \
 -Dcloud.deployer.reportDir=<dir path> \
deploy  ③
```

① In the first Maven command, we recognize the list of properties used by the `ci-builder` plugin.

② The profile `cloud-deployer` is set, followed by all the required properties.

③ Default Maven goal `deploy` to invoke `cloud-deployer`.

Generated Report

For each Maven module executed by the plugin, a detailed report is generated and saved into the `reportsDir` directory parameter.

The report file name is based on the Job/Route module processed. Template: `cloud-deployer-report<job/route name>< version>.txt`

Ex: Generated Report for the execution of the module: `jobs/process/AWS/s3_0.1`

```
---------------
- Started      : 2023-07-03 at 14:47:35
---------------

- PROJECT
  - Name       : TALEND8TEST
  - Branch     : master_R2023-01_cloud-deployer
  - Commit
    - Id       : 4651641aad138263e057c17e3dc986aac0bc1703
    - Date     : 2023-07-03T10:03:23-08:00
    - Author   : rboukefoussa@talend.com

- MODULE
  - Type       : process
  - Name       : S3
  - Version    : 0.1
  - folder     : aws/

- TALEND CLOUD
  - Environment: redha-dev
    - Id       : 6207f129ce4cda223c78f732
  - Workspace  : dev
    - Id       : 6207f129ce4cda223c78f733
  - Artifact
    - Name     : S3
    - Id       : 631d5fe16f0f1b0d19c0d8e4
    - Version  : 0.1.37.20230502104747
    - Url      : https://tmc.us.cloud.talend.com/manage-
artifacts/6207f129ce4cda223c78f732/workspace/6207f129ce4cda223c78f733/detail/job/631d5
fe16f0f1b0d19c0d8e4
  - Task
    - Name     : TALEND8TEST: S3 task
    - Id       : 63e02ab49dccfc51424317ea
    - Version  : 5.5
    - Engine   : redha-dev-cluster (REMOTE_ENGINE_CLUSTER)
    - Url      : https://tmc.us.cloud.talend.com/tasks-and-plans-
administration/6207f129ce4cda223c78f732/6207f129ce4cda223c78f733/tasks/job/63e02ab49dc
cfc51424317ea/detail

---------------
- Finished     : 2023-07-03 at 14:47:55
```

```
---------------
- Durations
  - Publish    : 00:11.771 min
  - Deploy     : 00:06.897 min
  - Total      : 00:19.824 min
```

How to ...

Shorten the maven goal command

To invoke a goal of a non-Standard Maven plugin such as `clouddeployer`, Maven needs to resolve the exact dependency coordinates (GAV: groupId, ArtifactI, Version) of the plugin.

Hence, the usage of the following in all previous Maven commands:

```
          # groupId    : ArtifactId               :Version:Goal
mvn ...   org.talend.cd:clouddeployer-maven-plugin:8.5.0:publish-deploy
```

However, Maven provides a way to register the prefix of plugins following a specific naming convention

`<prefix>-maven-plugin`

and hence shorten the command. See maven-prefix for more details.

As indicated in the Maven documentation, the `groupId` of `clouddeployer` can be added to the Maven setting.xml, as follows:

```
<settings>
  ...
<pluginGroups>
  <pluginGroup>org.talend.cd</pluginGroup>
</pluginGroups>
 ...
</settings>
```

We can, then, use the shorter syntax in our Maven command:

```
         # Prefix     : Goal
mvn ...  clouddeployer:publish-deploy
```

Release notes

- 8.5.0 (01/2024)

- Bug
  - Reports: Set artifact version to the longer version when cloud-deployer task management is skipped: -Dcloud.deployer.skip=true.
- New features
  - Add support of data plane artifact publication et task management.
  - Add option to set the publication version of the artifact.
- Dependency Upgrade
  - tc-apis-client to 2.0.9 ⇒ Add support of Data Plane environments.

- 8.4.0 (11/2023)

- New features
  - Addition of the execution report in json.
- Dependency Upgrade
  - tc-apis-client to 2.0.7 ⇒ Talend Cloud APIs R2023-11

- 8.3.0 (10/2023)

- Dependency Upgrade
  - tc-apis-client to 2.0.3 ⇒ Talend Cloud APIs R2023-10 - CVE reports update

- 8.2.0 (10/2023)

Increase default timeout for artifact upload to 10 min to support large artifacts.

- Dependency Upgrade
  - tc-apis-client to 2.0.2 ⇒ Talend Cloud APIs R2023-08

- 8.1.0 (10/2023)

- Dependency Upgrade
  - tc-apis-client to 2.0.1 ⇒ Talend Cloud APIs R2023-08
  - Maven to 3.9.4

- 8.0.5 (07/2023)

- Dependency Upgrade
  - tc-apis-client to 1.12.0 ⇒Talend Cloud APIs R2023-06
  - Maven to 3.9.3

- 8.0.4 (04/2023)

- Bugs
  - Fix update task run-config. Default Manual Trigger information was missing when no triggers were set/requested.
- New features
  - Additional validation when setting tasks with CLOUD engine if artifact is of type Data Services or Route: CLOUD engine is not supported.

- 8.0.3 (02/2023)

- New features
  - Integrate the `publish` goal of the `cloud-publisher` into `cloud-deployer` allowing to publish an artifact and create/configure the associated task in a single Maven call.
  - Add support for Route & Data services tasks deployment.
  - Generation of an execution report.

- 8.0.2 (01/2023)

- New Usage
  - The creation of `cloud-deployer` profile in the Talend Project root pom is not required anymore. If the plugin is available (in the local .m2 or in Nexus/Artifactory repository,) it can be used directly in the Maven command.
- New features
  - Support for template for some parameter values. See Configuration template
  - Support of configuration file that adapts to the Talend module being executed. See Configuration file
- Task Configuration
  - addition of new parameters `runProfile`, `impersonatedUser`, `timeout`
- Task Triggers:
  - Rename plugin parameter `runType` to `triggers`
  - Add support for CRON trigger
  - Triggers parameter supports definition of up to 15 triggers (TMC limit)

- If one trigger is defined, the trigger is added as part of the Task RunConfig. If more than one trigger is specified, a Schedule with the list of triggers is created and attached to the Task.
- The parameter `scheduleId` allows to attach an existing schedule to the Task. If both `triggers` & `scheduleId` parameters are set, `scheduleId` takes precedence.
- A combination of Webhook and any other trigger type are not possible (mutually exclusive)