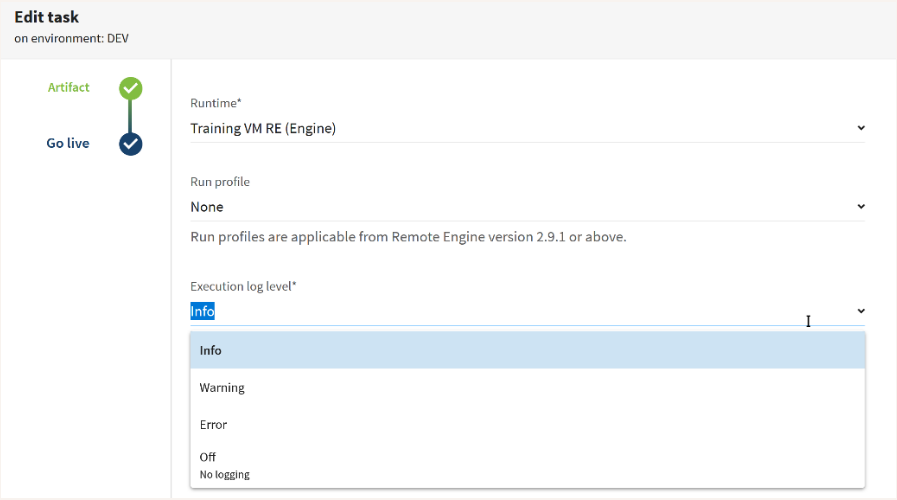
Logging and monitoring in Talend Management Console

OVERVIEW

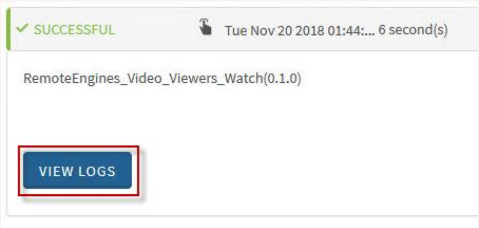
Talend Cloud allows you to run tasks on distant engines while monitoring the task executions and the resources they consume. In Talend Management Console, you can run tasks on a Cloud Engine or a Remote Engine using various levels of execution logs. When a task runs on a Remote Engine, you can monitor the engine resources using the Observability Metrics Collector (OMC).

EXECUTION LOGS

Follow these tips when using the Job execution logs in Talend Management Console.

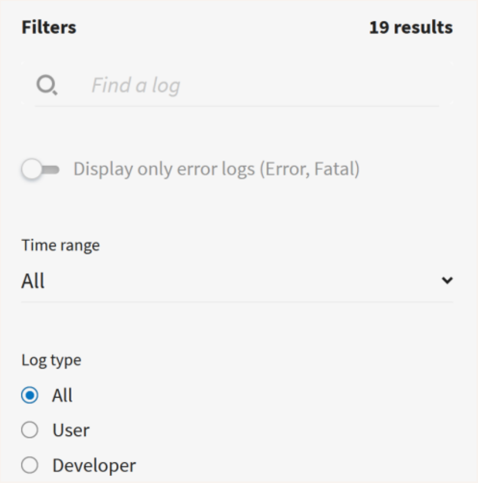
Configuring a task	
 Edit task on environment: DEV Artifact Go live Runtime*: Training VM RE (Engine) Run profile: None Run profiles are applicable from Remote Engine version 2.9.1 or above. Execution log level*: Info Info Warning Error Off No logging	Select the Execution log level when you configure a task execution in Talend Management Console: • **Error** for problematic events where the execution process may or may not continue • **Warning** for potentially harmful events where the execution process continues • **Info** for coarse-grained informational events that highlight execution processes

Accessing the logs


✓ SUCCESSFUL
Tue Nov 20 2018 01:44:... 6 second(s)
RemoteEngines_Video_Viewers_Watch(0.1.0)
VIEW LOGS

When the Job ends, click the **VIEW LOGS** button to access the logs.

Filtering the logs



Filters **19 results**

Find a log

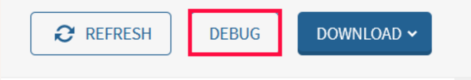
Display only error logs (Error, Fatal)

Time range
All

Log type
● All
○ User
○ Developer

For a clearer view, filter the logs using keywords, time ranges, and log types.

Possible actions



You can download the logs and run the task in debug mode.

BEST PRACTICES

Talend recommends that you:

- Set the log level to **Info** during testing phases only.
- Set the log level to **Warning** or **Error** in a production environment.
- Use **debug mode** only in actual debugging scenarios and rarely or never in a production environment, because running a task in debug mode can generate a huge amount of detail.

OBSERVABILITY METRICS COLLECTOR (OMC)

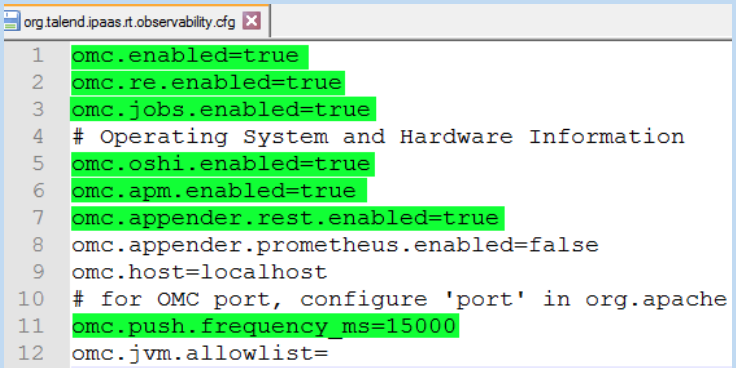
Follow these steps to enable and query the Observability Metrics Collector (OMC).

1

Use the Remote Engine configuration files located in **<remote engine installation directory>\etc** to enable and configure the OMC.

From there you can:

- Enable metrics generation
- Configure the metrics format (JSON or Prometheus)
- Configure the metrics push frequency
- Configure the metrics lifetime
- Add more contextual information to the metrics (optional)



```
omc.enabled=true
omc.re.enabled=true
omc.jobs.enabled=true
# Operating System and Hardware Information
omc.oshi.enabled=true
omc.apm.enabled=true
omc.appender.rest.enabled=true
omc.appender.prometheus.enabled=false
omc.host=localhost
# for OMC port, configure 'port' in org.apache
omc.push.frequency_ms=15000
omc.jvm.allowlist=
```

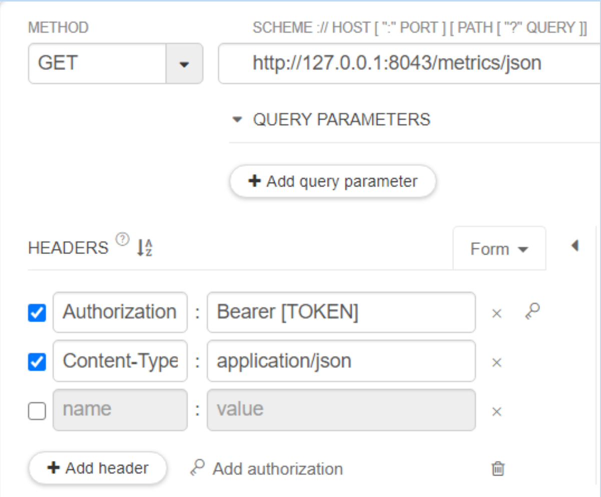
Main configuration files:

- **org.talend.ipaas.rt.observability.cfg**
  for enabling OMC and configuring the metrics generation frequency
- **org.talend.observability.omc.cache.apm.cfg**
  for setting the metrics lifetime
- **org.talend.ipaas.rt.observability.jobContextConfig.cfg**
  for adding contextual details

2

Copy the token in the file **org.talend.observability.http.security.cfg** to access the Remote Engine monitoring API.

Reuse it for authentication when querying the API.



METHOD: GET

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]: http://127.0.0.1:8043/metrics/json

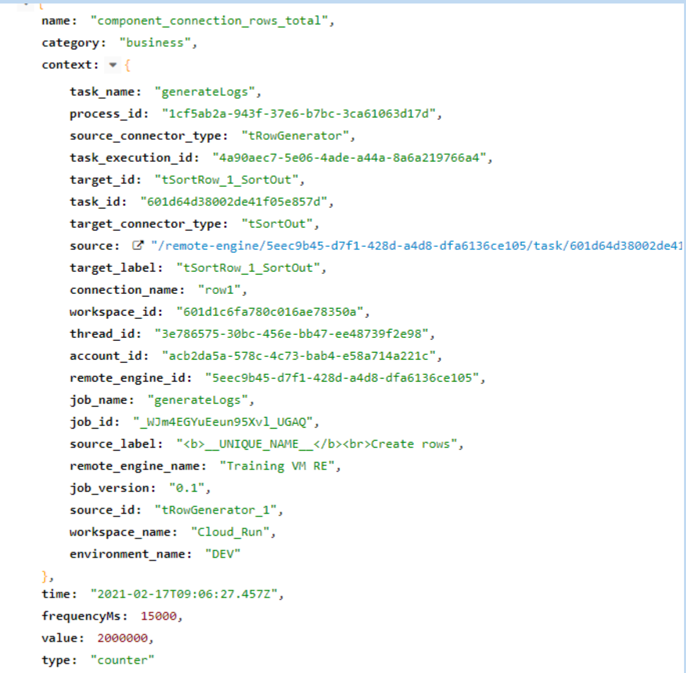
☑	Authorization	Bearer [TOKEN]
☑	Content-Type	application/json
□	name	value

3

Query the API and channel the metrics to your monitoring and alerting console.

Various kinds of metrics are available:

- Operating system resource consumption
- Remote Engine resource consumption and workload
- Job run resource consumption (CPU, memory)
- Job run history (including component information, status, and duration)
- Job run performance (for example, records processed or rejected)



```
name:  "component_connection_rows_total",
category:  "business",
context:  {
    task_name:  "generateLogs",
    process_id:  "1cf5ab2a-943f-37e6-b7bc-3ca61063d17d",
    source_connector_type:  "tRowGenerator",
    task_execution_id:  "4a90aec7-5e06-4ade-a44a-8a6a219766a4",
    target_id:  "tSortRow_1_SortOut",
    task_id:  "601d64d38002de41f05e857d",
    target_connector_type:  "tSortOut",
    source:  "/remote-engine/5eec9b45-d7f1-428d-a4d8-dfa6136ce105/task/601d64d38002de41
    target_label:  "tSortRow_1_SortOut",
    connection_name:  "row1",
    workspace_id:  "601d1c6fa780c016ae78350a",
    thread_id:  "3e786575-30bc-456e-bb47-ee48739f2e98",
    account_id:  "acb2da5a-578c-4c73-bab4-e58a714a221c",
    remote_engine_id:  "5eec9b45-d7f1-428d-a4d8-dfa6136ce105",
    job_name:  "generateLogs",
    job_id:  "_WJm4EGYuEeun95Xvl_UGAQ",
    source_label:  "<b>__UNIQUE_NAME__</b><br>Create rows",
    remote_engine_name:  "Training VM RE",
    job_version:  "0.1",
    source_id:  "tRowGenerator_1",
    workspace_name:  "Cloud_Run",
    environment_name:  "DEV"
},
time:  "2021-02-17T09:06:27.457Z",
frequencyMs:  15000,
value:  2000000,
type:  "counter"
```