

Configure Batch Job Runner

Contents:

- *Configure Timeout*
- *Configure Job Threads*
- *Configure BJR for EMR*
 - *Multiple BJR instances*
 - *YARN logs from EMR*
- *Configure Database*
- *Logging*

The Trifacta® platform utilizes the batch job runner service to orchestrate jobs that are executed on the selected backend running environment. This service passes jobs to the backend and tracks their progress until success or failure. This service is enabled by default.

Configure Timeout

Setting	Default Value	Description
<code>batchserver.spark.requestTimeoutMillis</code>	120000 (2 minutes)	Maximum number of milliseconds that the Batch Job Runner service should wait for a response from the Spark Job service during job execution. Default is 2 minutes.

Configure Job Threads

As needed, you can configure the number of worker threads assigned to each process that is managed by the batch job runner. Depending of the volume and complexity of jobs that you run of each type, you may choose to modify these settings to improve performance for key job types.

Tip: These settings can be configured through the Admin Settings page in the Trifacta application. See [Admin Settings Page](#).

By running environment:

Setting	Default Value	Description
<code>batchserver.workers.photon.max</code>	2	Number of worker threads for running Trifacta Photon jobs. This value corresponds to the maximum number of photon jobs that can be queued at the same time. For more information, see Configure Photon Running Environment .
<code>batchserver.workers.spark.max</code>	16	Number of worker threads for running Spark jobs. For more information, see Configure Spark Running Environment .
<code>batchserver.workers.wrangle.max</code>	16	Number of worker threads for running transformation jobs.

By job type:

Setting	Default Value	Description
batchserver.workers.ingest.max	16	Maximum number of worker threads for running ingest jobs, which are used for loading large-scale relational data into the Transformer page. After this maximum number has been reached, subsequent requests are queued.
batchserver.workers.profile.max	16	Maximum number of worker threads for running profile jobs, which provide summary and detail statistics on job results.
batchserver.workers.publish.max	16	Maximum number of worker threads for running publish jobs, which deliver pre-generated job results to other datastores.
batchserver.workers.fileconverter.max	16	Maximum number of worker threads for running fileconverter jobs, which are used to convert source formats into output formats.
batchserver.workers.filewriter.max	16	Maximum number of worker threads for running filewriter jobs, which are used for writing file-based outputs to a specified storage location.

Depending on your running environment, there may be additional parameters that you can configure to affect Batch Job Runner for that specific environment:

- *Configure for Spark*
- *Configure Photon Running Environment*

Configure BJR for EMR

Multiple BJR instances

If the Trifacta platform is connected to an EMR cluster, multiple instances of the batch job runner are deployed to manage jobs across the cluster so that if one fails, YARN jobs are still tracked. No configuration is required.

YARN logs from EMR

The following properties below can be modified for batch job runner:

Setting	Default Value	Description
aws.emr.getLogsOnFailure	false	When set to <code>true</code> , YARN logs from all nodes in the EMR cluster are collected from S3 and stored on the Trifacta node in the following location: <pre style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">/opt/trifacta/logs/jobs/<jobId>/container</pre> where: <code><jobId></code> is the Trifacta platform internal identifier for the job that failed.
aws.emr.getLogsForAllJobs	false	When set to <code>true</code> , YARN logs from nodes in the EMR cluster are collected and stored in the above location for all jobs, whether they succeed or fail. <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0; text-align: center;"> NOTE: This parameter is intended for debugging purposes only. </div>

Configure Database

The Batch Job Runner utilizes its own Jobs database. For more information, see *Configure the Databases*.

Logging

For more information on logging for the service, see *Configure Logging for Services*.