Alation

Instructions for Crème Brûlée

Matthias Funke

Professional Services

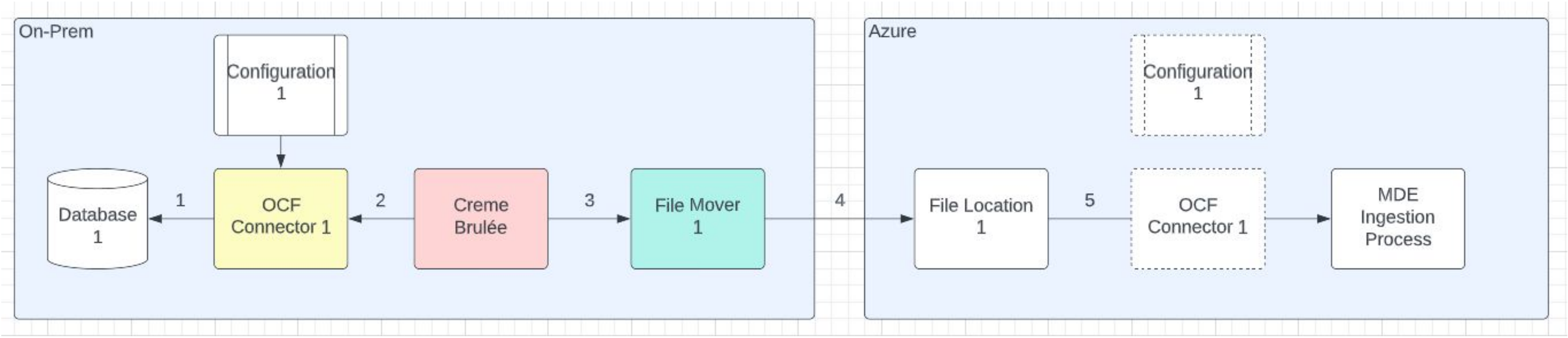
Allow MDE remotely

- Vattenfall wants to be able to do MDE remotely
- Solution described here uses existing OCF connectors (should work for any RDBMS OCF connector)
- Tested with SQL Server, Snowflake, Athena, Glue, Postgres
- Custom code "creme_brulee.py" is lightweight and low maintenance:
  - filters out blacklisted schemas
  - converts file format to make suitable for ingestion

Vattenfall Remaining Work

- Build framework for moving files
- Build framework for deploying containers automatically for each datasource
- Build framework for config management:
    - JDBC URI
    - username
    - password
    - any other params required for some

End-to-End Process



1. Yellow container uses OCF framework (built in Java)
2. Red container uses Python to transform the files into suitable format and filters out undesired schemas
3. File Mover (Vattenfall) moves the files to the Alation server
4. Ingestion on Alation server is done using shadow OCF connector (file based)

OCF Container

1. Obtain the connector ZIP file from Alation
2. Unzip it
3. Load the image
4. Verify the image

```
sh-4.2$ aws s3 cp s3://com.alationpro/creme/alation-awsglue-1.0.1.1117.zip .
download: s3://com.alationpro/creme/alation-awsglue-1.0.1.1117.zip to
./alation-awsglue-1.0.1.1117.zip
sh-4.2$ unzip ./alation-awsglue-1.0.1.1117.zip
Archive:  ./alation-awsglue-1.0.1.1117.zip
  inflating: alation-awsglue-ocf:1.0.1.1117.img
  inflating: MANIFEST.MF
sh-4.2$ sudo docker image load -i alation-awsglue-ocf:1.0.1.1117.img
Loaded image: alation-awsglue-ocf:1.0.1.1117
sh-4.2$ sudo docker image ls |grep glu
alation-awsglue-ocf              1.0.1.1117              89119871006a   4 months ago    252MB
```

Create a config.json file

1. The relevant keys are listed in the MANIFEST file

```
[
{
 "key": "jdbc_uri",
 "value": "sqlserver://sql-server.alationproserv.com:1433",
 "type": "TEXT"
},
{
 "key": "use_ntlm",
 "value": "True",
 "type": "BOOLEAN"
},
{
 "key": "domain_name",
 "value": "ALATTIONPROSERV",
 "type": "TEXT"
},
{
 "key": "username",
 "value": "alation-svc",
 "type": "TEXT"
},
{
 "key": "password",
 "value": "<REDACTED>",
 "type": "ENCRYPTEDTEXT"
}
]
```

Create a my_conf.yaml file

1. Blacklist any schemas you don't want (all lower case)
2. Provide data source ID (optional)
3. Provide file name for CSV file (optional)

```
# yaml file for snowflake MDE
blacklist:
- citibike.mytestschema
- citibike.mytestschema2

~~ds_id: 20~~
~~csv_file_name: my_csv.csv~~
```

Create shared Docker volume

1. RAM disk means no passwords are permanently saved to disk

```
docker volume create --driver local --opt type=tmpfs --opt device=tmpfs --opt o=size=100m,uid=1000
foo

[root@ip-172-31-5-116 _data]# docker volume inspect foo
[
    {
        "CreatedAt": "2022-12-12T13:39:47Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/foo/_data",
        "Name": "foo",
        "Options": {
            "device": "tmpfs",
            "o": "size=100m,uid=1000",
            "type": "tmpfs"
        },
        "Scope": "local"
    }
]
```

Copy files into Docker volume

```
[root@ip-172-31-5-116 _data]# ls -l
total 28
-rw-r--r-- 1 root root 4577 Dec 12 10:18 creme_brulee.py # code for filtering

-rw-r--r-- 1 root root  156 Dec 12 10:17 my_conf.yaml # configuration for filters

-rw-r--r-- 1 root root  326 Dec  6 11:27 snow.json # configuration for data source
```

Start OCF Container

1. Start the container as a daemon
2. Verify it has started

```
[root@ip-172-31-5-116 _data]# docker run -d -e PORT=8980 -v foo:/creme --name alation-awsglue-ocf
alation-awsglue-ocf:1.0.1.1117
825d31e72941872ea76a8e1dfc75bc5d803a455134f18128eebe3a1f2a3eb8da
[root@ip-172-31-5-116 _data]# docker logs $(docker ps -f name=alation-awsglue-ocf -q)
INFO    2022-12-12 13:51:06,351 [main] alation.sdk.core.grpc.server.GrpcServer  - Attempting to start
the Server
INFO    2022-12-12 13:51:06,462 [main] alation.sdk.core.grpc.server.GrpcServer  - Server started,
listening on 8980
[root@ip-172-31-5-116 _data]#
```

Run MDE in OCF Container

The command starts MDE in the correct container with the correct configuration

```
[root@ip-172-31-5-116 _data]# docker exec $(docker ps -f name=alation-awsglue-ocf -q) java -cp
/opt/APP.jar alation.sdk.rdbms.grpc.client.AbstractRDBMSClient -p 8980 --conf /creme/glue.json -o
/creme CONF
INFO   2022-12-12 14:09:02,841 [main] alation.sdk.rdbms.grpc.client.ConfigurationVerificationClient
- Send request to verify configuration!
INFO   2022-12-12 14:09:06,436 [main] alation.sdk.core.grpc.client.AbstractClient  - success

[root@ip-172-31-5-116 _data]# docker exec $(docker ps -f name=alation-awsglue-ocf -q) java -cp
/opt/APP.jar alation.sdk.rdbms.grpc.client.AbstractRDBMSClient -p 8980 --conf /creme/glue.json -o
/creme META
INFO   2022-12-12 14:09:12,834 [main] alation.sdk.rdbms.grpc.client.MetadataExtractionClient  - Send
request to run metadata extraction!
INFO   2022-12-12 14:09:18,148 [main] alation.sdk.core.grpc.client.AbstractClient  - success
[root@ip-172-31-5-116 _data]#
```

Optional: inspect output files

Successful MDE produces .txt files

```
[root@ip-172-31-5-116 _data]# ls -lrt
total 1000
-rw-r--r-- 1 root root    326 Dec  6 11:27 snow.json
-rw-r--r-- 1 root root    296 Dec  7 12:42 analytics.json
-rw-r--r-- 1 root root    156 Dec 12 10:17 my_conf.yaml
-rw-r--r-- 1 root root   4577 Dec 12 10:18 creme_brulee.py
-rw-r--r-- 1 root root    292 Dec 12 14:08 glue.json
-rw-r--r-- 1 root root  33718 Dec 12 14:09 table.txt
-rw-r--r-- 1 root root   4002 Dec 12 14:09 schema.txt
-rw-r--r-- 1 root root   1097 Dec 12 14:09 extractionProgress.txt
-rw-r--r-- 1 root root 949877 Dec 12 14:09 column.txt
-rw-r--r-- 1 root root      8 Dec 12 14:09 Status.txt
[root@ip-172-31-5-116 _data]# cat schema.txt
{"id_":{"name_":{"original_":"default","normalized_":"","memoizedIsInitialized":-1,"unknownFields":{"
fields":{},"fieldsDescending":{}},"memoizedSize":-1,"memoizedHashCode":0},"memoizedIsInitialized":-1,
"unknownFields":{"fields":{},"fieldsDescending":{}},"memoizedSize":-1,"memoizedHashCode":0},"sourceCo
mment_":"Default Hive
database","memoizedIsInitialized":-1,"unknownFields":{"fields":{},"fieldsDescending":{}},"memoizedSiz
e":-1,"memoizedHashCode":0}
```

Build Creme Brulee container from Dockerfile

Place Dockerfile and requirements.txt in a separate directory

```
[root@ip-172-31-5-116 creme_brulee]# docker build . -t creme-brulee:1.2
Sending build context to Docker daemon  8.192kB
Step 1/4 : FROM python:3.8
..
Removing intermediate container b3743e802c67
 ---> a4eea27eed0a
Successfully built a4eea27eed0a
Successfully tagged creme-brulee:1.2
[root@ip-172-31-5-116 creme_brulee]# ls
Dockerfile  requirements.txt
[root@ip-172-31-5-116 creme_brulee]# docker image ls
REPOSITORY                       TAG                     IMAGE ID       CREATED         SIZE
creme-brulee                     1.2                     a4eea27eed0a   2 minutes ago   1.08GB
```

Convert txt files to dump files and filter schemas

Successful conversion produces .dump files. Note file names change

```
[root@ip-172-31-5-116 _data]# docker run -v foo:/creme creme-brulee:1.2 python3
/creme/creme_brulee.py /creme/my_conf.yaml
All done.
[root@ip-172-31-5-116 _data]# ls -lrt
total 1332
-rw-r--r-- 1 root root    326 Dec  6 11:27 snow.json
-rw-r--r-- 1 root root    296 Dec  7 12:42 analytics.json
-rw-r--r-- 1 root root    156 Dec 12 10:17 my_conf.yaml
-rw-r--r-- 1 root root   4577 Dec 12 10:18 creme_brulee.py
-rw-r--r-- 1 root root    292 Dec 12 14:08 glue.json
-rw-r--r-- 1 root root  33718 Dec 12 14:09 table.txt
-rw-r--r-- 1 root root   4002 Dec 12 14:09 schema.txt
-rw-r--r-- 1 root root   1097 Dec 12 14:09 extractionProgress.txt
-rw-r--r-- 1 root root 949877 Dec 12 14:09 column.txt
-rw-r--r-- 1 root root      8 Dec 12 14:09 Status.txt
-rw-r--r-- 1 root root    672 Dec 12 14:14 Schema.dump
-rw-r--r-- 1 root root   9849 Dec 12 14:14 Table.dump
-rw-r--r-- 1 root root 281954 Dec 12 14:14 Attribute.dump
-rw-r--r-- 1 root root  38448 Dec 12 14:14 my_csv.csv
```

/creme is the mounted volume inside the creme brulee container

Move files to Alation Server (Vattenfall)

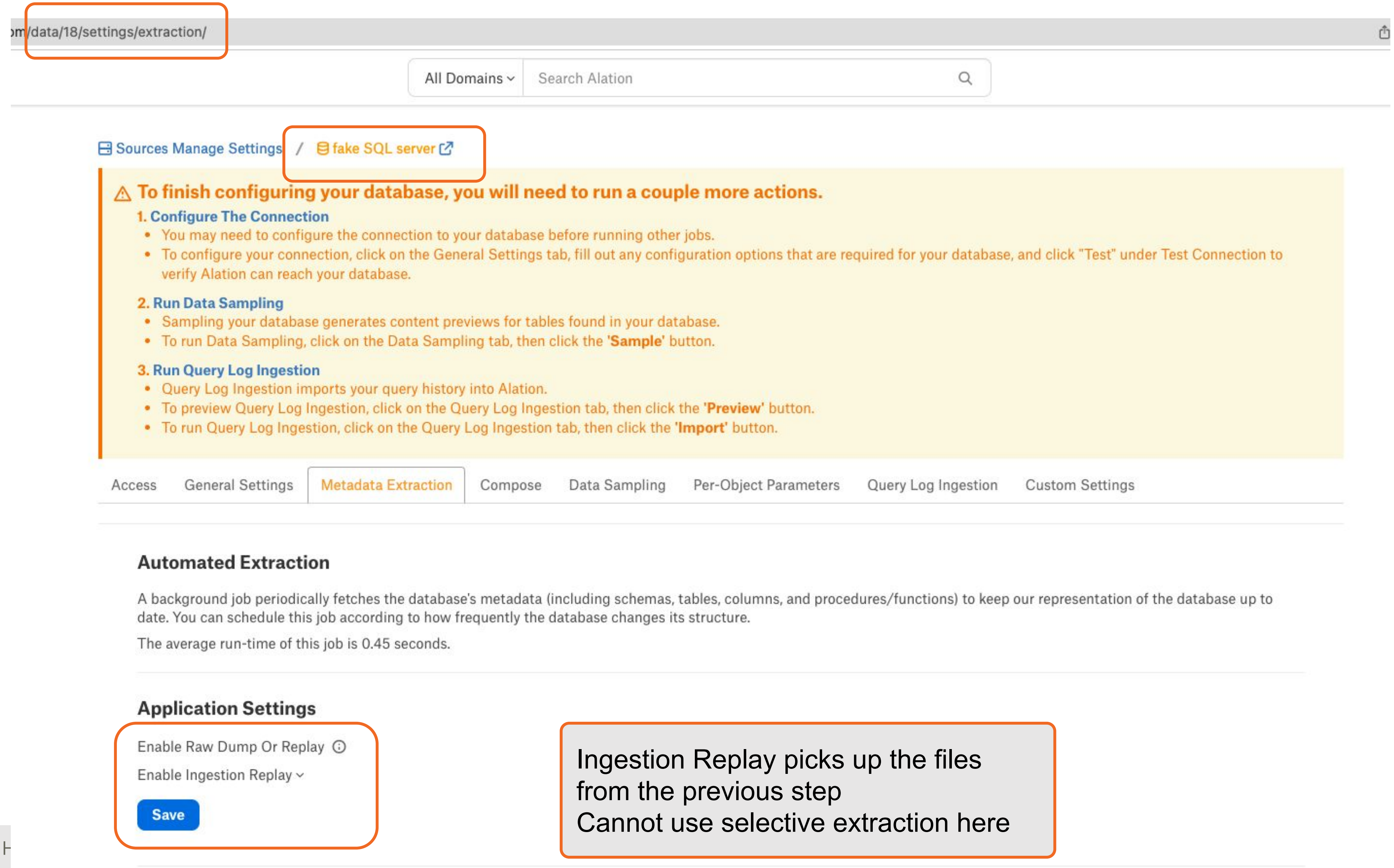
```
ls -l /data/site_data/tmp/rosemeta/
total 28
drwxrwxr-x 4 alation alation 4096 Dec  5 14:36 18
drwxrwxr-x 4 alation alation 4096 Dec  6 13:48 20
drwxrwxr-x 3 alation alation 4096 Dec  6 13:04 21
drwxr-xr-x 3 root    root    4096 Dec 12 14:30 28
drwxr-xr-x 3 alation alation 4096 Nov 29 12:49 4
drwxr-xr-x 5 alation alation 4096 Dec  6 12:51 6
drwxr-xr-x 2 alation alation 4096 Oct 24 09:17 7

sh-4.2$ find .
.
./6
./6/1969
./6/1969/Schema.dump
./6/1969/Catalog.dump
./6/1969/Table.dump
./6/1969/Attribute.dump
./6/2013
./6/2013/Schema.dump
./6/2013/Table.dump
```

These are data source IDs
Use the correct OCF connector
No other config required or possible

Directory name can be any number
MDE will use the highest number
Recommended: 4-digit

Example: 2013 is used, not 1969

Alation Side: "Fake" OCF Connector



om/data/18/settings/extraction/

Sources Manage Settings / fake SQL server

To finish configuring your database, you will need to run a couple more actions.

1. **Configure The Connection**
   - You may need to configure the connection to your database before running other jobs.
   - To configure your connection, click on the General Settings tab, fill out any configuration options that are required for your database, and click "Test" under Test Connection to verify Alation can reach your database.
2. **Run Data Sampling**
   - Sampling your database generates content previews for tables found in your database.
   - To run Data Sampling, click on the Data Sampling tab, then click the '**Sample**' button.
3. **Run Query Log Ingestion**
   - Query Log Ingestion imports your query history into Alation.
   - To preview Query Log Ingestion, click on the Query Log Ingestion tab, then click the '**Preview**' button.
   - To run Query Log Ingestion, click on the Query Log Ingestion tab, then click the '**Import**' button.

Automated Extraction

A background job periodically fetches the database's metadata (including schemas, tables, columns, and procedures/functions) to keep our representation of the database up to date. You can schedule this job according to how frequently the database changes its structure.

The average run-time of this job is 0.45 seconds.

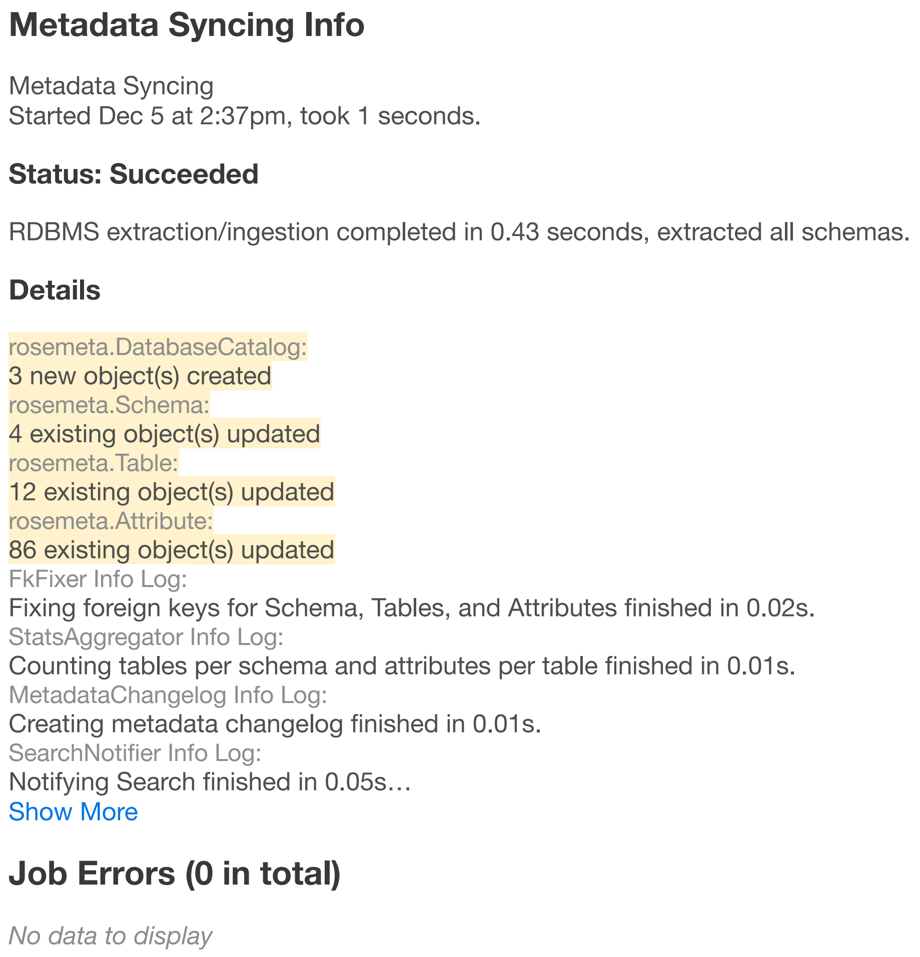
Application Settings

Enable Raw Dump Or Replay

Enable Ingestion Replay

Ingestion Replay picks up the files from the previous step
Cannot use selective extraction here

Success Page



Metadata Syncing Info

Metadata Syncing
Started Dec 5 at 2:37pm, took 1 seconds.

Status: Succeeded

RDBMS extraction/ingestion completed in 0.43 seconds, extracted all schemas.

Details

```
rosemeta.DatabaseCatalog:
3 new object(s) created
rosemeta.Schema:
4 existing object(s) updated
rosemeta.Table:
12 existing object(s) updated
rosemeta.Attribute:
86 existing object(s) updated
FkFixer Info Log:
Fixing foreign keys for Schema, Tables, and Attributes finished in 0.02s.
StatsAggregator Info Log:
Counting tables per schema and attributes per table finished in 0.01s.
MetadataChangelog Info Log:
Creating metadata changelog finished in 0.01s.
SearchNotifier Info Log:
Notifying Search finished in 0.05s…
```

Show More

Job Errors (0 in total)

No data to display

You can schedule MDE to run every hour / every day

No need for config changes