

Tungsten Replicator Master Class

Advanced: Working with Data Warehouse Targets

Chris Parker

VP of Customer Success, EMEA

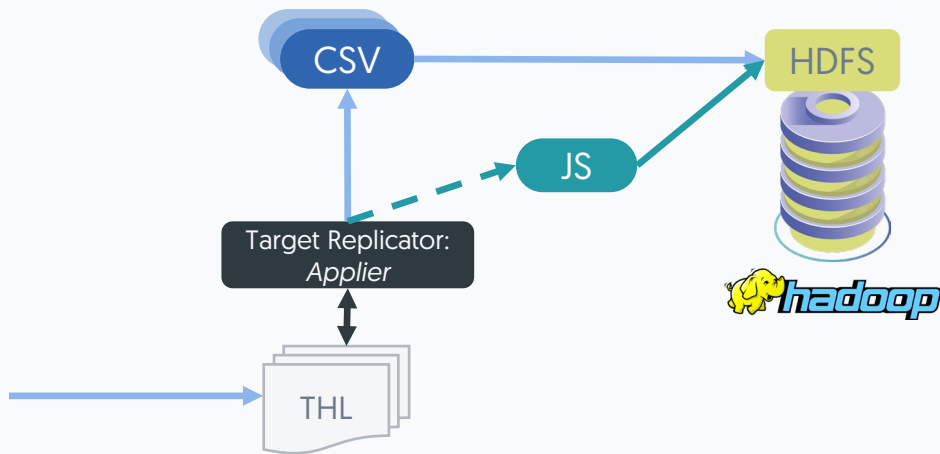
Topics

In this short course, we will:

- Review replicator flow
- Explore Hadoop, Redshift and Vertica specific pre-requisites
- Review configurations

Replicator Flow

How Hadoop Replication Works



How the Hadoop Materialisation Works

```
insert into t1
  values (1,"Hello World!");

insert into t1
  values (2,"Meet Continuent");

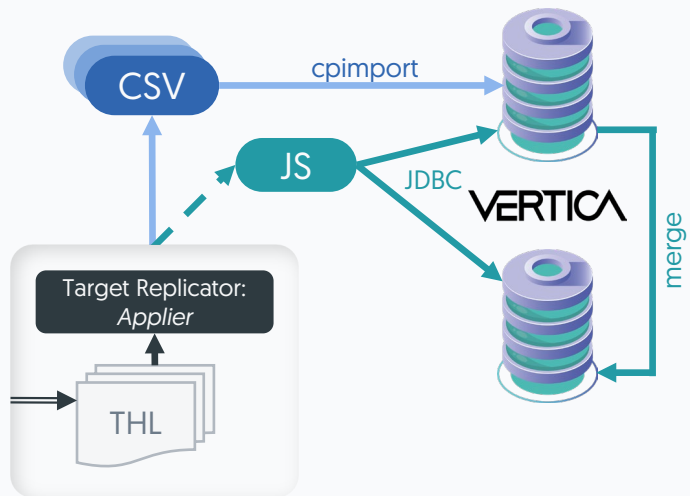
update t1
  set msg="Goodbye World"
  where ID = 1;
```

Op	Seqno	ID	Msg
I	1	1	Hello World!
I	2	2	Meet Continuent
D	3	1	
I	3	1	Goodbye World

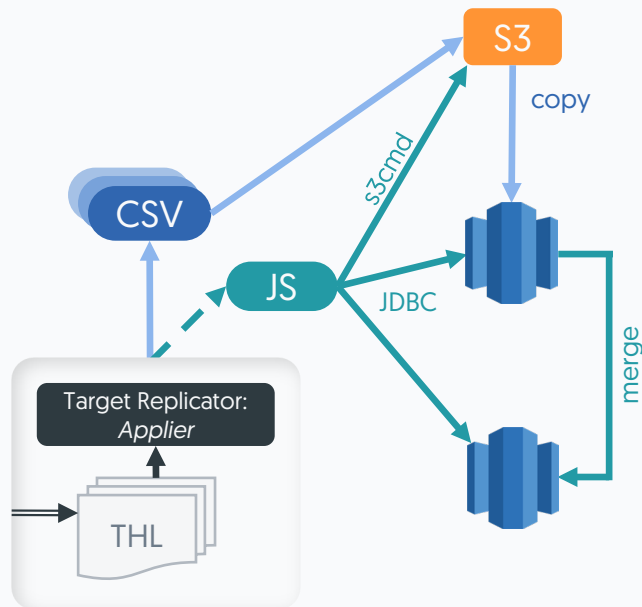


Op	Seqno	ID	Msg
I	2	2	Meet Continuent
I	3	1	Goodbye World

How Vertica Replication Works



How Redshift Replication Works



Prerequisites

- Review online documentation
 - <https://docs.continuent.com>
 - Download the Prerequisite Checklist
- Extractor/applier hosts
 - OS user
 - /etc/hosts
 - sudoers
 - Ruby
 - Java
- Network
 - Review port requirements
- MySQL
 - my.cnf settings
 - User accounts
 - ROW based binary logging

Hadoop

- HDFS writeable by replicator user

Vertica

- User accounts
- JDBC drivers

Redshift

- User accounts
- S3 Bucket
- S3 Tools for uploading
- AWS JSON config

All tables need primary keys

AWS JSON Config for Redshift

Key	Description
awsS3Path	The location within your S3 storage where files should be loaded.
awsAccessKey / awsSecretKey	The S3 access & secret keys to access your S3 storage. Not required if awsIAMRole is used.
awsIAMRole	The IAM role configured to allow Redshift to interact with S3. Not required if awsAccessKey and awsSecretKey are in use.
multiServiceTarget	A boolean value used to indicate if there are multiple appliers writing into the single Redshift Target. (Default: false)
singleLockTable	To indicate the table lock behaviour when multiServiceTarget is true. Will be ignored if multiServiceTarget set to false (Default: true)
lockTablePrefix	The prefix for the lock tables when singleLockTable is false. (Default: lock_XXX_)
s3Binary	The binary to use for loading csv file up to S3. [Valid Values: s3cmd, s4cmd, aws] (Default: s3cmd)
redshiftCopyOptions	Allows the passing of additional valid syntax to be added to the Redshift COPY command during csv loading from S3 into Redshift Staging Tables
cleanUpS3Files	A boolean value used to identify whether the CSV files loaded into S3 should be deleted after they have been imported and merged. (Default: true)
gzipS3Files	Setting to true will result in the csv files being gzipped prior to loading into S3 (Default: false)
storeCDCIn	A definition table that stores the change data from the load, in addition to importing to staging and base tables.

Provisioning Options

- Traditional CSV export and import
- Dump and load through Blackhole engine
- If target support standard SQL, extract data as INSERTS
- For Hadoop, use Sqoop

Object Mapping

Hadoop

- MySQL database → HDFS directory
- Table → Hive compatible CSV File
- Row → line in the file

Redshift & Vertica

[PostgreSQL Interface and Syntax]

- MySQL instance → database
- MySQL database → schema

DDLScan

- Reverse engineers MySQL objects
- Creates target specific DDL
- Can be used for single objects or entire databases
- Must be configured prior to starting replicators
- Must be run twice, once for base tables, once for staging tables

```
ddlscan -service alpha -template ddl-mysql-redshift.vm -db test >ddl.sql
```

```
ddlscan -service alpha -template ddl-mysql-redshift-staging.vm -db test >ddl-staging.sql
```

```
ddlscan -service alpha -template ddl-mysql-redshift-lock.vm -db test >ddl-lock.sql
```

Extractor Config

```
[defaults]
user=tungsten
install-directory=/opt/continuent
mysql-allow-intensive-checks=true
profile-script=~/.bash_profile
disable-security-controls=true
rest-api-admin-user=tungsten
rest-api-admin-password=secret
replicator-rest-api-authentication=true
replicator-rest-api-ssl=true
replicator-rest-api-port=8097
replicator-rest-api-address=0.0.0.0

[alpha]
master=tr-ext-2
members=tr-ext-2
replication-user=tungsten
replication-password=secret
replication-port=3306
enable-heterogeneous-service=true
svc-applier-filters=dropstatementdata,replicate
property=replicator.filter.replicate.ignore=mysql
#or
property=replicator.filter.replicate.do=<schemanamehere>
```

Applier Configs

Vertica

```
[defaults]
user=tungsten
install-directory=/opt/continuent
profile-script=~/.bash_profile
disable-security-controls=true
rest-api-admin-user=tungsten
rest-api-admin-password=secret
replicator-rest-api-authentication=true
replicator-rest-api-ssl=true
replicator-rest-api-port=8097
replicator-rest-api-address=0.0.0.0
```

```
[alpha]
master=tr-ext-2
members=verticahost
datasource-type=vertica
replication-user=dbadmin
replication-password=Secret123
replication-port=5433
vertica-dbname=demo
batch-enabled=true
batch-load-template=vertica6
batch-load-language=js
svc-applier-block-commit-interval=30s
svc-applier-block-commit-size=250000
```

continuent

Redshift

```
[defaults]
user=tungsten
install-directory=/opt/continuent
profile-script=~/.bash_profile
disable-security-controls=true
rest-api-admin-user=tungsten
rest-api-admin-password=secret
replicator-rest-api-authentication=true
replicator-rest-api-ssl=true
replicator-rest-api-port=8097
replicator-rest-api-address=0.0.0.0
```

```
[alpha]
master=tr-ext-2
members=tr-app-1
datasource-type=redshift
replication-user=dbadmin
replication-password=Secret123
replication-port=5439
replication-host=redshift-endpoint
redshift-dbname=demo
batch-enabled=true
batch-load-template=redshift
svc-applier-block-commit-interval=30s
svc-applier-block-commit-size=250000
```

Applier Configs

Hadoop

```
[defaults]
user=tungsten
install-directory=/opt/continuent
profile-script=~/.bash_profile
disable-security-controls=true
rest-api-admin-user=tungsten
rest-api-admin-password=secret
replicator-rest-api-authentication=true
replicator-rest-api-ssl=true
replicator-rest-api-port=8097
replicator-rest-api-address=0.0.0.0

[alpha]
master=tr-ext-2
members=hadoopapplier
datasource-type=file
property=replicator.datasource.global.csvType=hive
replication-user=tungsten
replication-password=secret
batch-enabled=true
batch-load-template=hadoop
batch-load-language=js
svc-applier-block-commit-interval=30s
svc-applier-block-commit-size=250000
```

Understanding Block Commit

- Block commit groups multiple transactions into a single commit for higher throughput.
- Ideal for batch-load targets where latency is less critical.
- Configure `svc-applier-block-commit-size` to commit after large batches of events.
- Optionally use `svc-applier-block-commit-interval` to time-based flush long-running loads.
- Results: reduced commit overhead, faster bulk apply, and higher replication efficiency.

Summary

What we have learnt today:

- Reviewed replicator flow
- Explored Hadoop, Redshift and Vertica specific pre-requisites
- Reviewed configurations

Thank you for listening

continuent.com

Chris Parker

VP of Customer Success, EMEA