Hive Schema Tool

Metastore Schema Verification

Version

Introduced in Hive 0.12.0. See HIVE-3764.

Hive now records the schema version in the metastore database and verifies that the metastore schema version is compatible with Hive binaries that are going to accesss the metastore. Note that the Hive properties to implicitly create or alter the existing schema are disabled by default. Hive will not attempt to change the metastore schema implicitly. When you execute a Hive query against an old schema, it will fail to access the metastore:

```
$ build/dist/bin/hive -e "show tables"
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. java.lang.RuntimeException:
Unable to instantiate org.apache.hadoop.hive.metastore.HiveMetaStoreClient
```

The log will contain an error about version information not found:

```
...
Caused by: MetaException(message:Version information not found in metastore. )
...
```

By default the configuration property **hive.metastore.schema.verification** is false and metastore to implicitly write the schema version if it's not matching. To enable the strict schema verification, you need to set this property to true in `hive-site.xml`.

See Hive Metastore Administration for general information about the metastore.

The Hive Schema Tool

Version

Introduced in Hive 0.12.0. See HIVE-5301. (Also see HIVE-5449 for a bug fix.)

The Hive distribution now includes an offline tool for Hive metastore schema manipulation. This tool can be used to initialize the metastore schema for the current Hive version. It can also handle upgrading the schema from an older version to current. It tries to find the current schema from the metastore if it is available. This will be applicable to future upgrades like 0.12.0 to 0.13.0. In case of upgrades from older releases like 0.7.0 or 0.10.0, you can specify the schema version of the existing metastore as a command line option to the tool.

The `schematool` figures out the SQL scripts required to initialize or upgrade the schema and then executes those scripts against the backend database. The metastore DB connection information like JDBC URL, JDBC driver and DB credentials are extracted from the Hive configuration. You can provide alternate DB credentials if needed.

The `schematool` Command

The `schematool` command invokes the Hive schema tool with these options:

```
$ schematool -help
usage: schemaTool
 -dbType <databaseType>             Metastore database type
 -driver <driver>                   Driver name for connection
 -dryRun                            List SQL scripts (no execute)
 -help                              Print this message
 -info                              Show config and schema details
 -initSchema                        Schema initialization
 -initSchemaTo <initTo>             Schema initialization to a version
 -metaDbType <metaDatabaseType>     Used only if upgrading the system catalog for hive
 -passWord <password>               Override config file password
 -upgradeSchema                     Schema upgrade
 -upgradeSchemaFrom <upgradeFrom>   Schema upgrade from a version
 -url <url>                         Connection url to the database
 -userName <user>                   Override config file user name
 -verbose                           Only print SQL statements
(Additional catalog related options added in Hive 3.0.0 (HIVE-19135] release are below.
 -createCatalog <catalog>       Create catalog with given name
 -catalogLocation <location>        Location of new catalog, required when adding a catalog
 -catalogDescription <description>  Description of new catalog
 -ifNotExists                       If passed then it is not an error to create an existing catalog
 -moveDatabase <database>                     Move a database between catalogs.  All tables under it would
still be under it as part of new catalog. Argument is the database name. Requires --fromCatalog and --toCatalog
parameters as well
 -moveTable  <table>                Move a table to a different database.  Argument is the table name. Requires
--fromCatalog, --toCatalog, --fromDatabase, and --toDatabase
 -toCatalog  <catalog>              Catalog a moving database or table is going to.  This is required if you
are moving a database or table.
 -fromCatalog <catalog>             Catalog a moving database or table is coming from.  This is required if you
are moving a database or table.
 -toDatabase  <database>            Database a moving table is going to.  This is required if you are moving a
table.
 -fromDatabase <database>           Database a moving table is coming from.  This is required if you are moving
a table.
```

The **dbType** is required and can be one of:

```
 derby|mysql|postgres|oracle|mssql
```

Version

The dbType "`mssql`" was added in Hive 0.13.1 with HIVE-6862.

Usage Examples

- Initialize to current schema for a new Hive setup:

```
$ schematool -dbType derby -initSchema
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 0.13.0
Initialization script hive-schema-0.13.0.derby.sql
Initialization script completed
schemaTool completed
```

- Get schema information:

```
$ schematool -dbType derby -info
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Hive distribution version:       0.13.0
Metastore schema version:        0.13.0
schemaTool completed
```

- Attempt to get schema information with older metastore:

```
$ schematool -dbType derby -info
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Hive distribution version:       0.13.0
org.apache.hadoop.hive.metastore.HiveMetaException: Failed to get schema version.
*** schemaTool failed ***
```

  Since the older metastore doesn't store the version information, the tool reports an error retrieving it.

- Upgrade schema from an 0.10.0 release by specifying the 'from' version:

```
$ schematool -dbType derby -upgradeSchemaFrom 0.10.0
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting upgrade metastore schema from version 0.10.0 to 0.13.0
Upgrade script upgrade-0.10.0-to-0.11.0.derby.sql
Completed upgrade-0.10.0-to-0.11.0.derby.sql
Upgrade script upgrade-0.11.0-to-0.12.0.derby.sql
Completed upgrade-0.11.0-to-0.12.0.derby.sql
Upgrade script upgrade-0.12.0-to-0.13.0.derby.sql
Completed upgrade-0.12.0-to-0.13.0.derby.sql
schemaTool completed
```

- Upgrade dry run can be used to list the required scripts for the given upgrade.

```
$ build/dist/bin/schematool -dbType derby -upgradeSchemaFrom 0.7.0 -dryRun
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting upgrade metastore schema from version 0.7.0 to 0.13.0
Upgrade script upgrade-0.7.0-to-0.8.0.derby.sql
Upgrade script upgrade-0.8.0-to-0.9.0.derby.sql
Upgrade script upgrade-0.9.0-to-0.10.0.derby.sql
Upgrade script upgrade-0.10.0-to-0.11.0.derby.sql
Upgrade script upgrade-0.11.0-to-0.12.0.derby.sql
Upgrade script upgrade-0.12.0-to-0.13.0.derby.sql
schemaTool completed
```

  This is useful if you just want to find out all the required scripts for the schema upgrade.

- Moving a database and tables under it from default Hive catalog to a custom spark catalog

```
build/dist/bin/schematool -moveDatabase db1 -fromCatalog hive -toCatalog spark
```

- Moving a table from Hive catalog to Spark Catalog

```
# Create the desired target database in spark catalog if it doesn't already exist.
beeline ... -e "create database if not exists newdb";
schematool -moveDatabase newdb -fromCatalog hive -toCatalog spark

# Now move the table to target db under the spark catalog.
schematool -moveTable table1 -fromCatalog hive -toCatalog spark  -fromDatabase db1 -toDatabase newdb
```