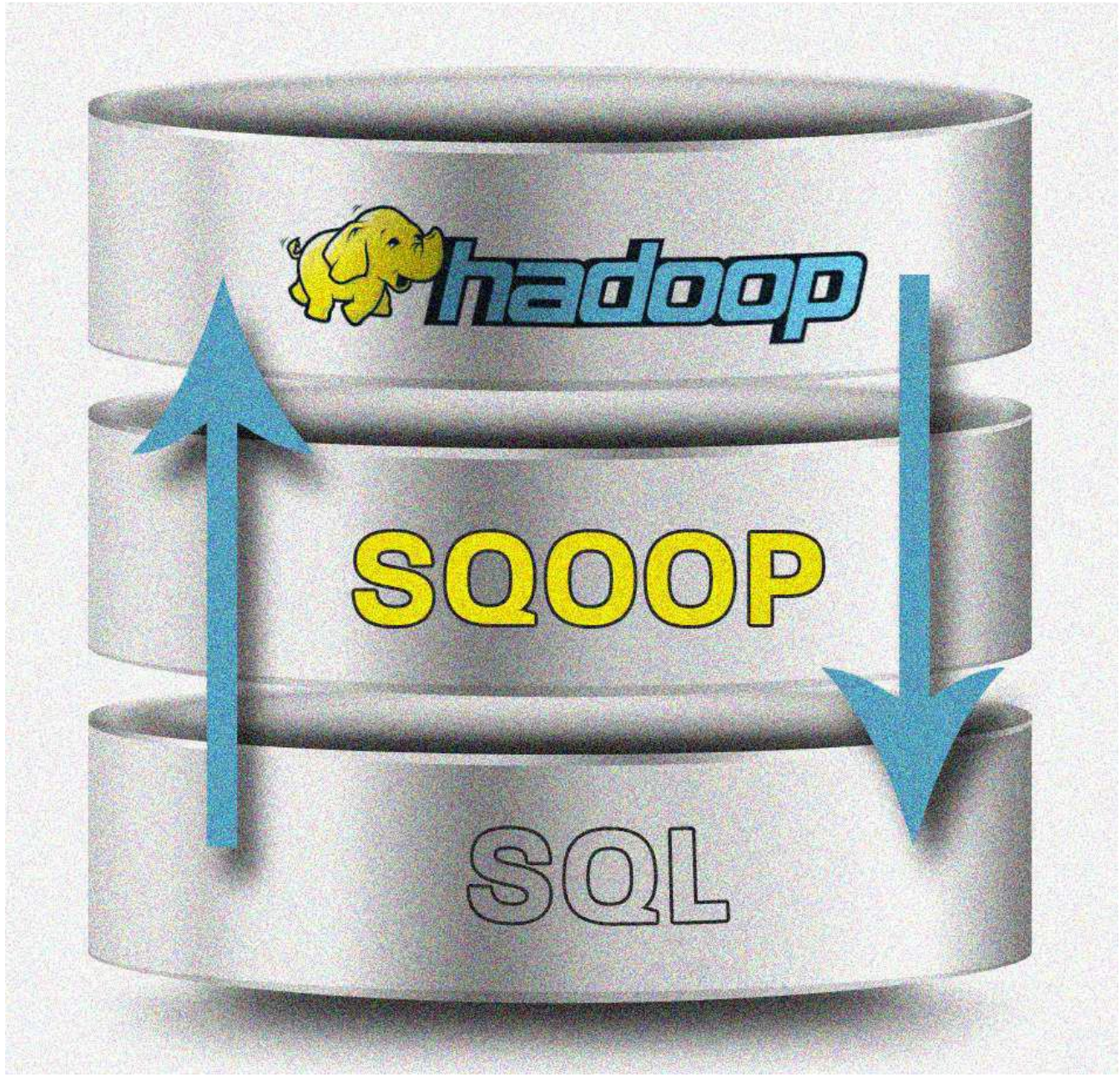
Apache Sqoop


hadoop
SQOOP
SQL

What is Sqoop?

- Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
- Sqoop imports data from external structured datastores into HDFS or related systems like Hive and HBase.
- Sqoop can also be used to export data from Hadoop and export it to external structured datastores such as relational databases and enterprise data warehouses.
- Sqoop works with relational databases such as: Teradata, Netezza,Oracle, MySQL, Postgres, and HSQLDB.

Why Sqoop?

- As more organizations deploy Hadoop to analyse vast streams of information, they may find they need to transfer large amount of data between Hadoop and their existing databases, data warehouses and other data sources

- Loading bulk data into Hadoop from production systems or accessing it from map-reduce applications running on a large cluster is a challenging task since transferring data using scripts is a inefficient and time-consuming task

How Sqoop Works?

- Sqoop provides a pluggable connector mechanism for optimal connectivity to external systems.
- The Sqoop extension API provides a convenient framework for building new connectors which can be dropped into Sqoop installations to provide connectivity to various systems.
- Sqoop itself comes bundled with various connectors that can be used for popular database and data warehousing systems.

Who Uses Sqoop?

- Online Marketer Coupons.com uses sqoop to exchange data between Hadoop and the IBM Netezza data warehouse appliance,The organization can query its structres databases and pipe the results into Hadoop using sqoop.

- Education company The Apollo group also uses the software not only to extract data from databases but to inject the results from Hadoop jobs back into relational databases

- And countless other hadoop users use sqoop to efficiently move their data

Importing Data - Lists databases in your mysql database

```
$ sqoop list-databases --connect jdbc:mysql://<mysql-
server>/employees --
username airawat --password myPassword
.
.
.
14/08/31 16:45:58 INFO manager.MySQLManager: Preparing to use a
MySQL
streaming resultset
information_schema
employees
test
```

Lists tables in your mysql database

```
sqoop list-tables --connect jdbc:mysql://<<mysql-
server>>/employees --
username airawat --password myPassword
.
.
.
14/08/31 16:45:58 INFO manager.MySQLManager: Preparing to use a
MySQL
streaming resultset.
departments
dept_emp
dept_manager
employees
employees_exp_stg
employees_export
salaries
titles
```

Importing data in MySql into HDFS

```
$ sqoop import \
--connect jdbc:mysql://mySqlServer-node/employees \
--username myUID \
--password myPWD \
--table employees \
-m 1 \
--target-dir /user/training/sqoop-mysql/employees
.
.
.
.9139 KB/sec)
14/08/31 22:32:25 INFO mapreduce.ImportJobBase: Retrieved 300024
records
```

Executing imports with an options file for static information

- Rather than repeat the import command along with connection related input required, each time, you can pass an options file as an argument to sqoop.

- Create a text file, as follows, and save it someplace, locally on the node you are running the sqoop client on.

Sample Options file:

```
$ vi SqoopImportOptions.txt
#
#Options file for sqoop import
#
Import  –connect jdbc:mysql://airawat-mySqlServer-node/employees  --username myUID  --password myPwd
#
#All other commands should be specified in the command line
```

Options File – Command

- The command

```
$ sqoop --options-file SqoopImportOptions.txt \
--table       departments \
-m 1 \
--target-dir /user/airawat/sqoop-mysql/departments
.
.
14/08/31 22:48:55 INFO mapreduce.ImportJobBase:
Transferred 153 bytes in 26.2453 seconds (5.8296 bytes/sec)
14/08/31 22:48:55 INFO mapreduce.ImportJobBase: Retrieved 9
records.
```

-m argument is to specify number of mappers. The department table has a handful of records, so I am setting it to 1.