Master Executive di II Livello

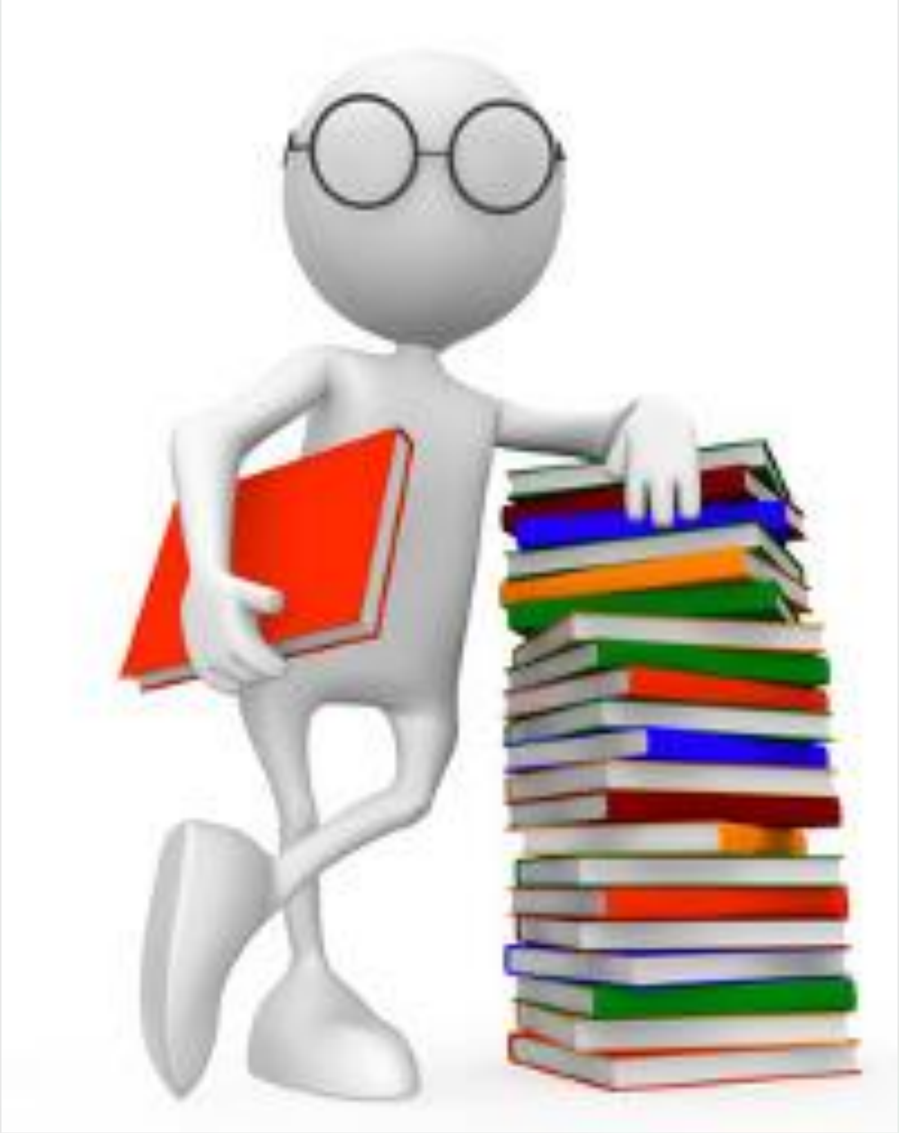
BIG DATA ANALYSIS AND BUSINESS INTELLIGENCE

Vamsi Krishna Varma Gunturi
Data science intern at ISTAT
vamsivarmagunturi@gmail.com

Introduction to Sqoop

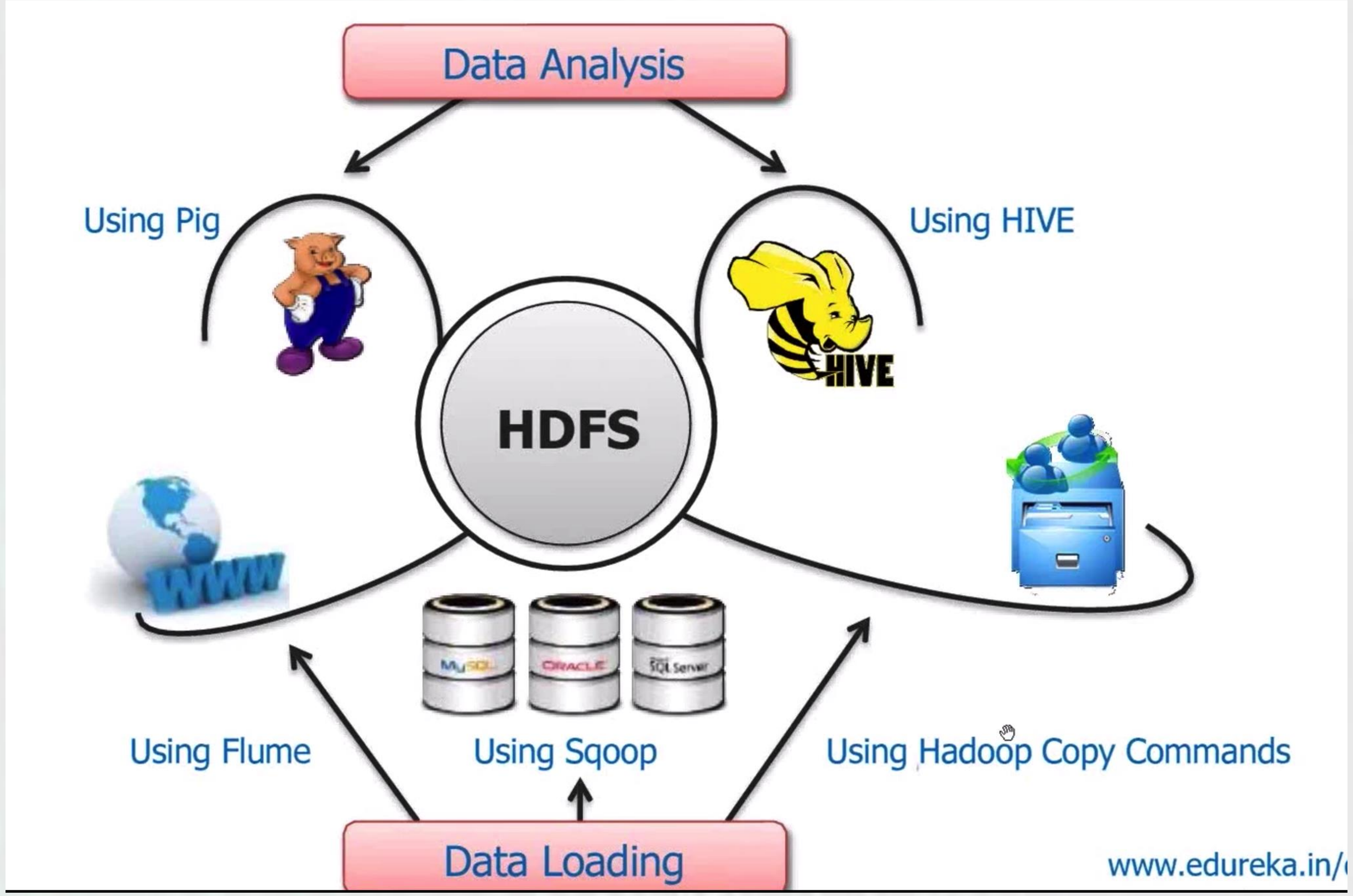
fondazione
INUIT
TOR VERGATA

Topics



- Tools for data loading
- Sqoop
- Why Sqoop ?
- How Sqoop works ?
- Sqoop architecture
- Sqoop features
- Flume vs Sqoop
- Sqoop import
- Sqoop export
- Sqoop commands

Data Loading tools



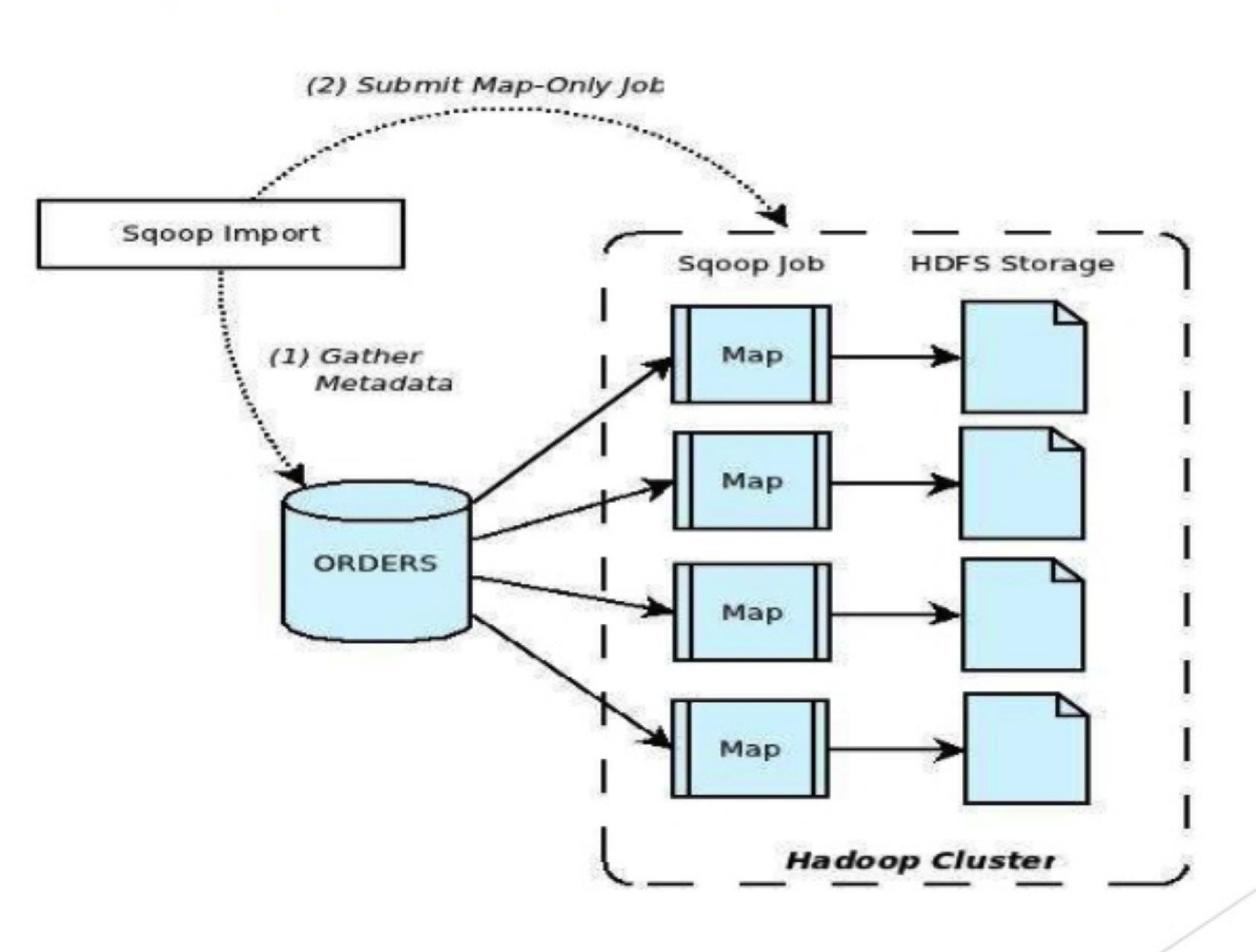
Sqoop

- **SQ**l for had**OOP.** Sqoop is data ingestion tool. Initially, Sqoop was developed and maintained by Cloudera. Later, on 23 July 2011, it was incubated by Apache. In April 2012, the Sqoop project was promoted as Apache’s top-level project.
- SQOOP is a tool designed for transfer data between HDFS and RDBMS such as MySQL, Oracle etc..,
- Export data back to RDBMS.
- Simple as user specifies the “what” and leave the “how” to underlying processing engine.
- Rapid development.
- No programming experience is required.
- Sqoop can easily integrate with Hadoop and dump structured data from relational databases on HDFS, complimenting the power of Hadoop.

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
- Allows data imports from external datastores and enterprise data warehouses into Hadoop
- Parallelizes data transfer for fast performance and optimal system utilization
- Copies data quickly from external systems to Hadoop
- Makes data analysis more efficient

How Sqoop works ?

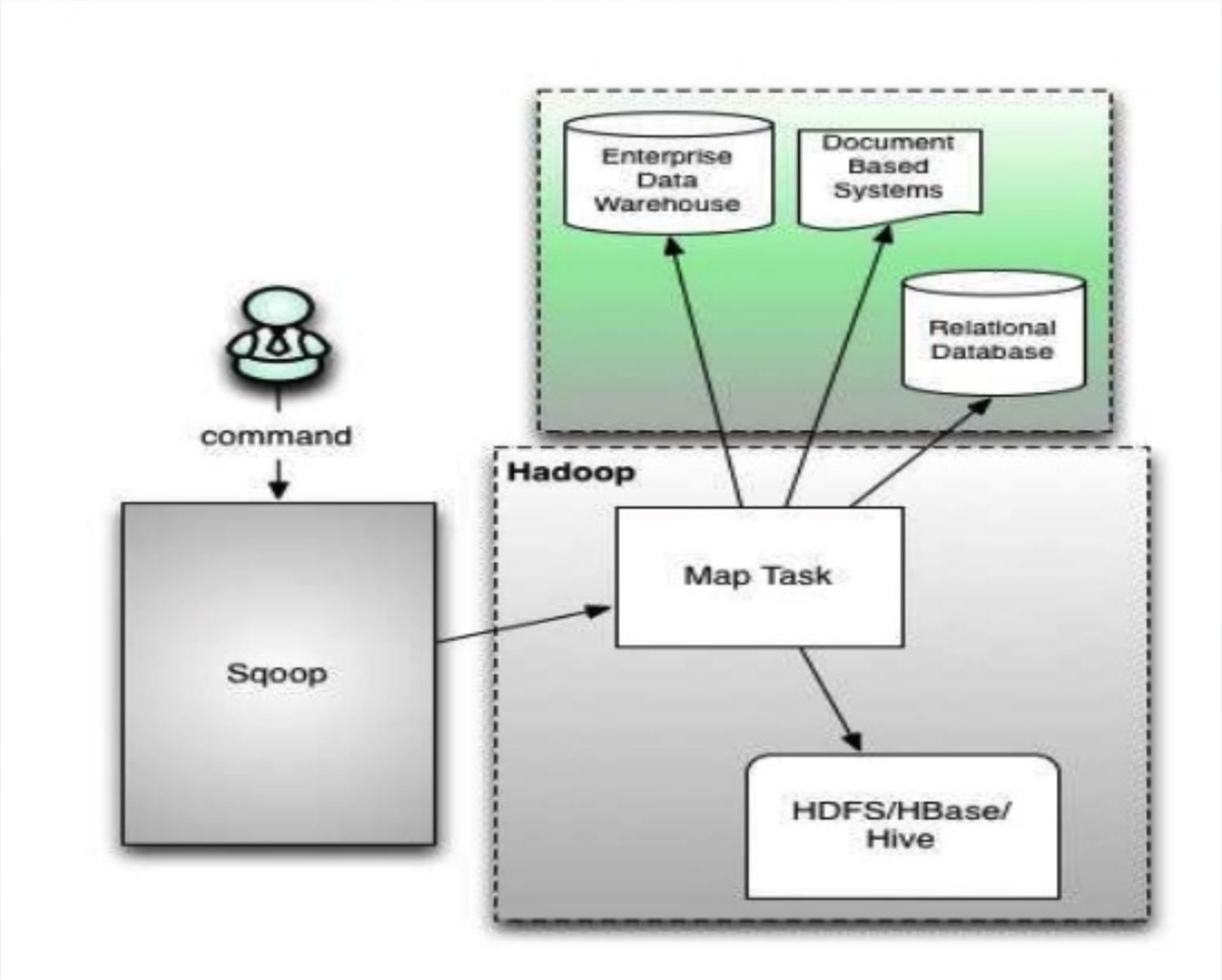


How Sqoop works ?

- Sqoop makes the life of developers easy by providing CLI for importing and exporting data. They just have to provide basic information like database authentication, source, destination, operations etc. It takes care of the remaining part.

- Sqoop internally converts the command into MapReduce tasks, which are then executed over HDFS. It uses YARN framework to import and export the data, which provides fault tolerance on top of parallelism.

Sqoop architecture



Sqoop Features

Sqoop provides many salient features like:

- **Full Load**: Apache Sqoop can load the whole table by a single command. You can also load all the tables from a database using a single command.
- **Incremental Load**: Apache Sqoop also provides the facility of incremental load where you can load parts of table whenever it is updated.
- **Parallel import/export**: Sqoop uses YARN framework to import and export the data, which provides fault tolerance on top of parallelism.
- **Import results of SQL query**: You can also import the result returned from an SQL query in HDFS.

Sqoop Features(2)

- **Compression**: You can compress your data by using deflate(gzip) algorithm with –compress argument, or by specifying –compression-codec argument. You can also load compressed table in Apache Hive.

- **Connectors for all major RDBMS Databases**: Apache Sqoop provides connectors for multiple RDBMS databases, covering almost the entire circumference.

- **Kerberos Security Integration**: Kerberos is a computer network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Sqoop supports Kerberos authentication.

- **Load data directly into HIVE/HBase**: You can load data directly into Apache Hive for analysis and also dump your data in HBase, which is a NoSQL database.

Flume vs Sqoop

The major difference between Flume and Sqoop is that:

Flume only ingests unstructured data or semi-structured data into HDFS.

While Sqoop can import as well as export structured data from RDBMS or Enterprise data warehouses to HDFS or vice versa.

Sqoop Import

- The import tool imports individual tables from RDBMS to HDFS. Each row in a table is treated as a record in HDFS.

- When we submit Sqoop command, our main task gets divided into subtasks which is handled by individual Map Task internally.

- Map Task is the subtask, which imports part of data to the Hadoop Ecosystem. Collectively, all Map tasks import the whole data.

Example:

```
sqoop import --connect
jdbc:mysql://localhost/employees
--username root
--password root --table cities
```

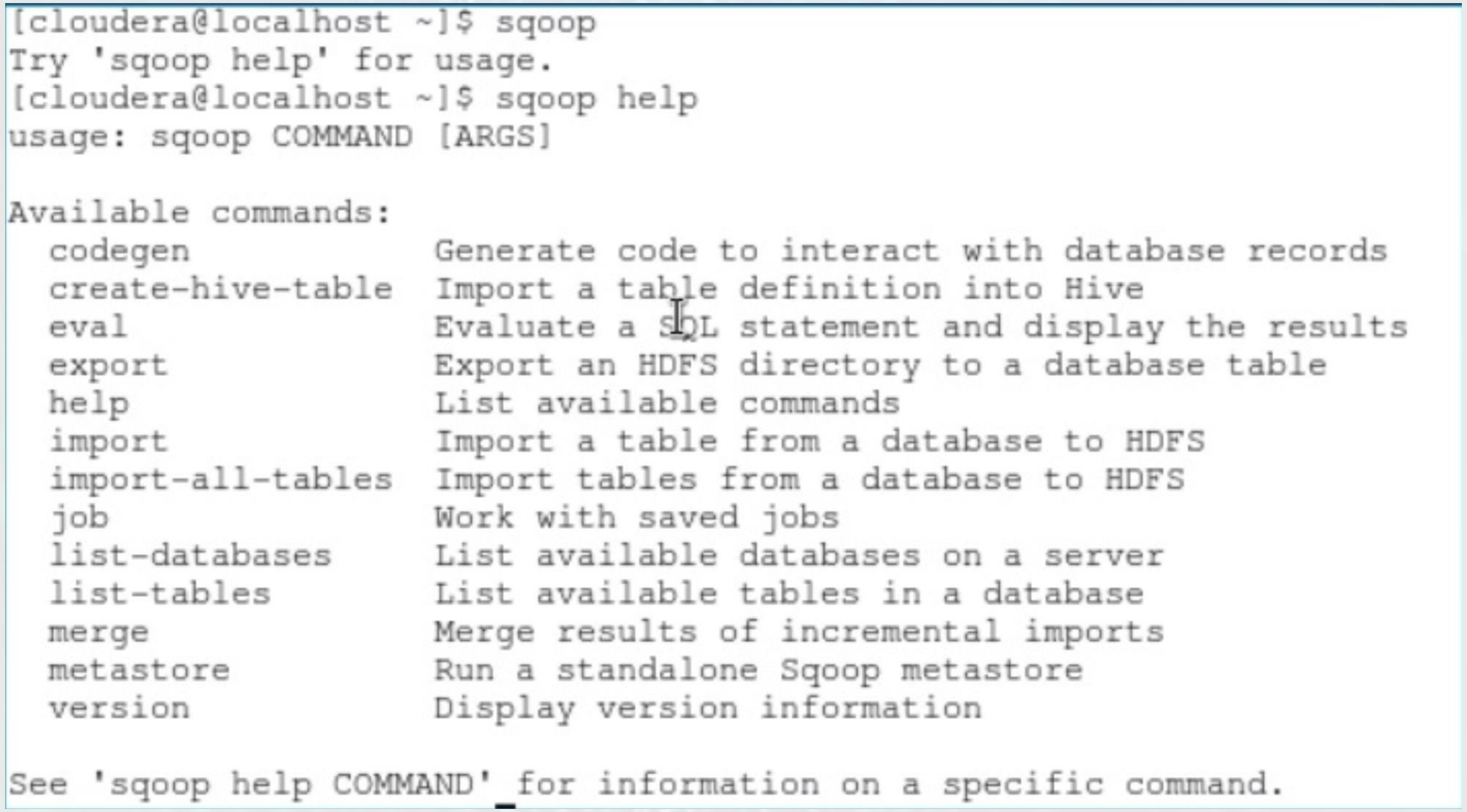
Sqoop Export

- The export tool exports a set of files from HDFS back to an RDBMS. The files given as input to Sqoop contain records, which are called as rows in the table.
- When we submit our Job, it is mapped into Map Tasks which brings the chunk of data from HDFS. These chunks are exported to a structured data destination.
- Combining all these exported chunks of data, we receive the whole data at the destination, which in most of the cases is an RDBMS (MYSQL/Oracle/SQL Server).

Example:

```
sqoop export --connect
  jdbc:mysql://localhost/employees
  --username root
  --password root --table cities --export-dir
  cities
```

Sqoop commands



```
[cloudera@localhost ~]$ sqoop
Try 'sqoop help' for usage.
[cloudera@localhost ~]$ sqoop help
usage: sqoop COMMAND [ARGS]

Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information

See 'sqoop help COMMAND' for information on a specific command.
```

Sqoop commands

Sqoop – IMPORT Command:

```
sqoop import --connect jdbc:mysql://localhost/employees
          --username root
          --table employees
```

Sqoop – IMPORT Command with target directory:

```
sqoop import --connect jdbc:mysql://localhost/employees
          --username root
          --table employees --m 1
          --target-dir /employees
```

Sqoop commands

Sqoop – IMPORT Command with Where Clause:

```
sqoop import --connect jdbc:mysql://localhost/employees
          --username root
          --table employees --m 3
          --where "emp_no > 49000"
          --target-dir /Latest_Employees
```

Sqoop – Incremental Import:

```
sqoop import --connect jdbc:mysql://localhost/employees
          --username root --table employees
          --target-dir /Latest_Employees
          --incremental append --check-column emp_no
          --last-value 499999
```

Sqoop commands

Sqoop – Import All Tables:

```
sqoop import-all-tables
--connect  jdbc:mysql://localhost/employees
--username root
```

Sqoop – List Databases:

```
sqoop list-databases
--connect jdbc:mysql://localhost/
--username root
```

Sqoop – List Tables:

```
sqoop list-tables
--connect jdbc:mysql://localhost/employees
--username root
```

Sqoop commands

Sqoop – Export:

```
sqoop export
--connect jdbc:mysql://localhost/employees
--username root
--table emp
--export-dir /user/edureka/employees
```

Sqoop – Codegen:

```
sqoop codegen
--connect jdbc:mysql://localhost/employees
--username root
--table employees
```

Master Executive di II Livello

BIG DATA ANALYSIS AND BUSINESS INTELLIGENCE

Grazie

Vamsi Krishna Varma Gunturi
Data science intern at ISTAT
vamsivarmagunturi@gmail.com

fondazione INUIT TOR VERGATA