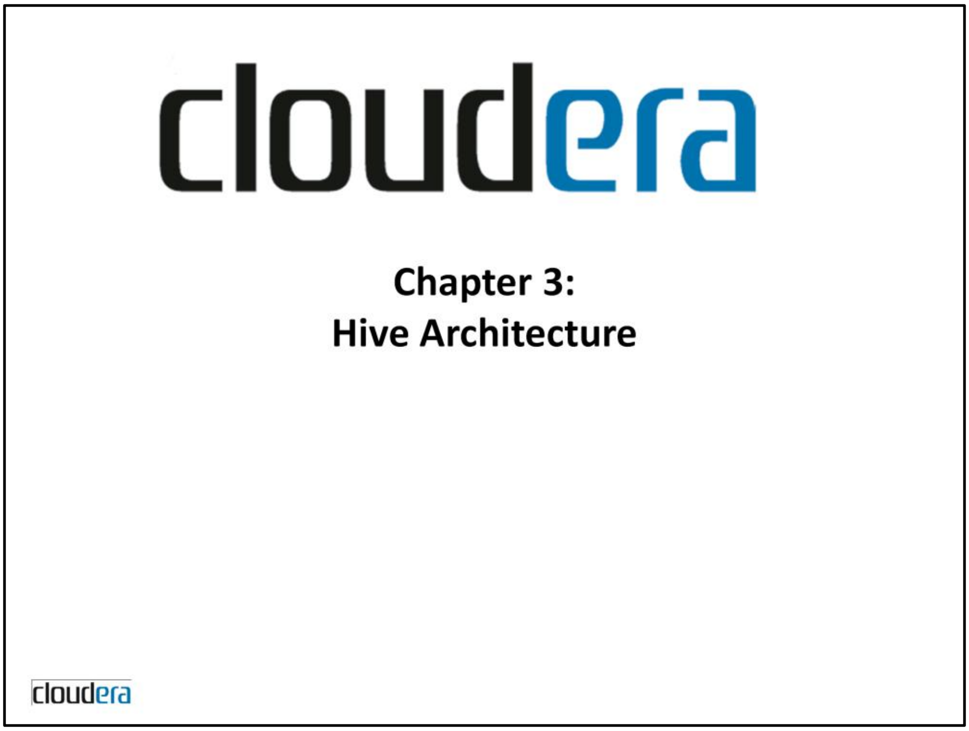
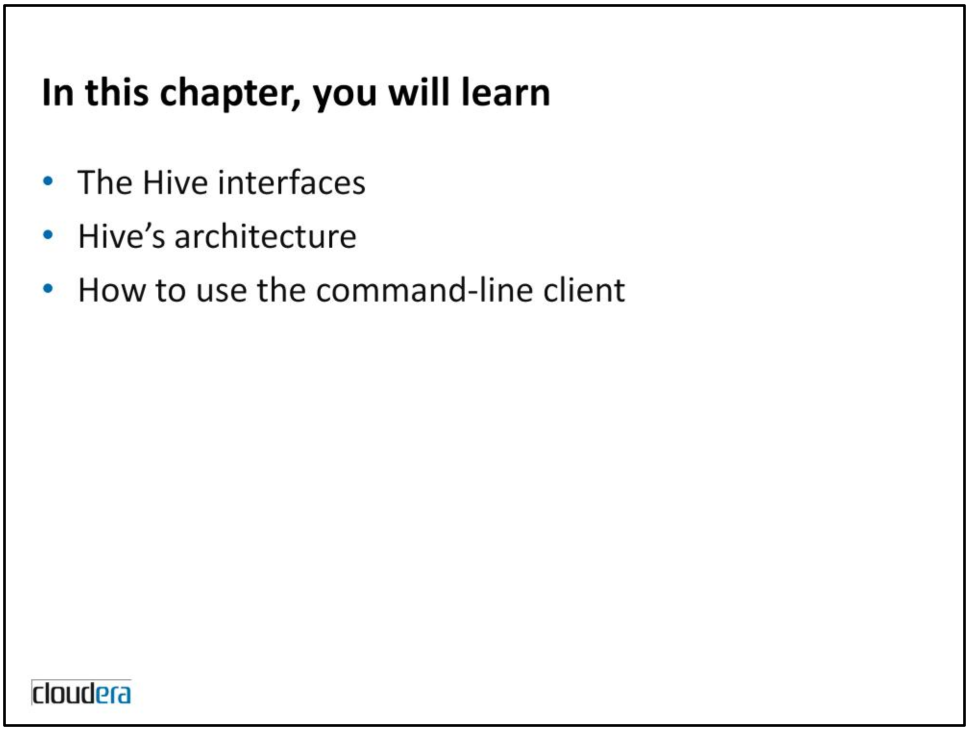
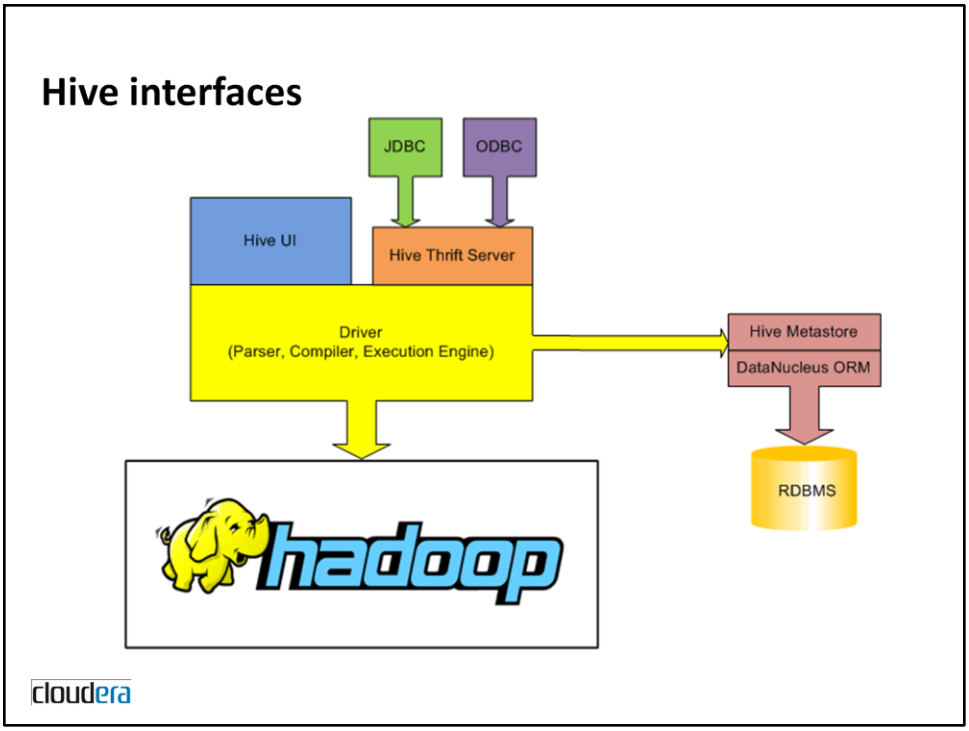


Chapter 3: Hive Architecture



In this chapter, you will learn

- The Hive interfaces
- Hive's architecture
- How to use the command-line client



Hive interfaces

JDBC

ODBC

Hive UI

Hive Thrift Server

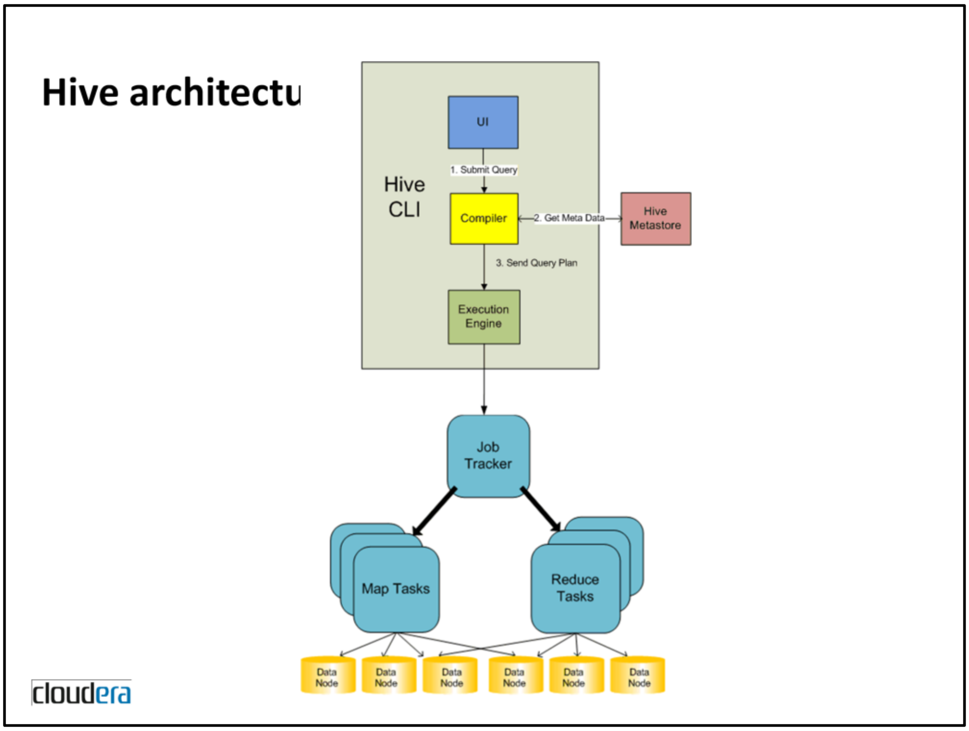
Driver
(Parser, Compiler, Execution Engine)

Hive Metastore

DataNucleus ORM

RDBMS

Hive runs on the user’s machine. The Hive UI is a command-line interface (CLI) for querying Hadoop. To use Hive, nothing needs to be installed on the Hadoop cluster. To use a JDBC/ODBC client, a thrift server must be installed and run to expose the Hive program as a service (it is common to run more than 1 Thrift Server if multiple clients will be using JDBC/ODBC).



Hive architectu

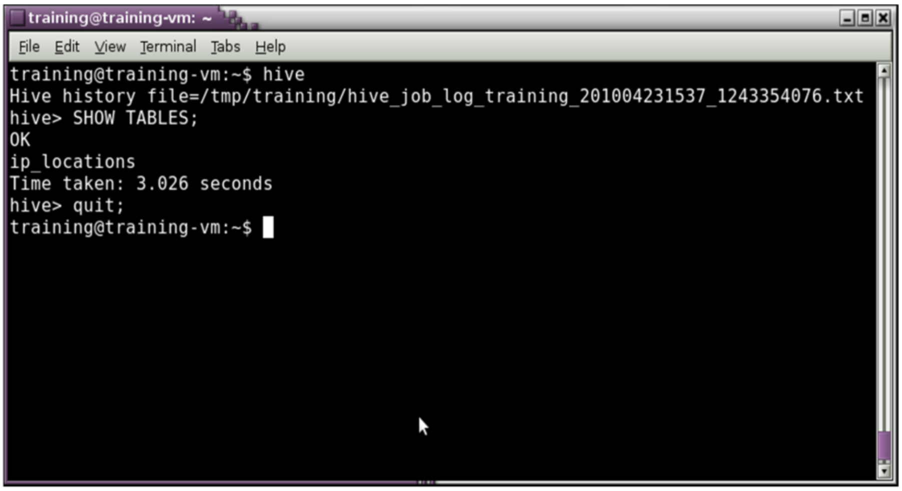
When a command is submitted to Hive, several things typically happen:

1. Parse the query
2. Get metadata from the metastore (this may be the embedded Derby database or a centralized RDBMS).
3. Create a logical plan (abstract syntax tree)
4. Optimize the plan
5. Create a physical plan. This is a directed acyclic graph of map reduce jobs.

The execution engine submits the jobs sequentially in dependency order, running jobs in parallel whenever possible. There are certain queries that do not require MapReduce. Metadata-only commands, for example, only need to communicate with the Hive metastore to evaluate results. Some queries cause Hive to accessing HDFS, but do not require any map/reduce phases. For example, "`SELECT * FROM table`" would not require MapReduce.

Note: do not open more than one hive shell at a time when using the default Derby database for the metastore

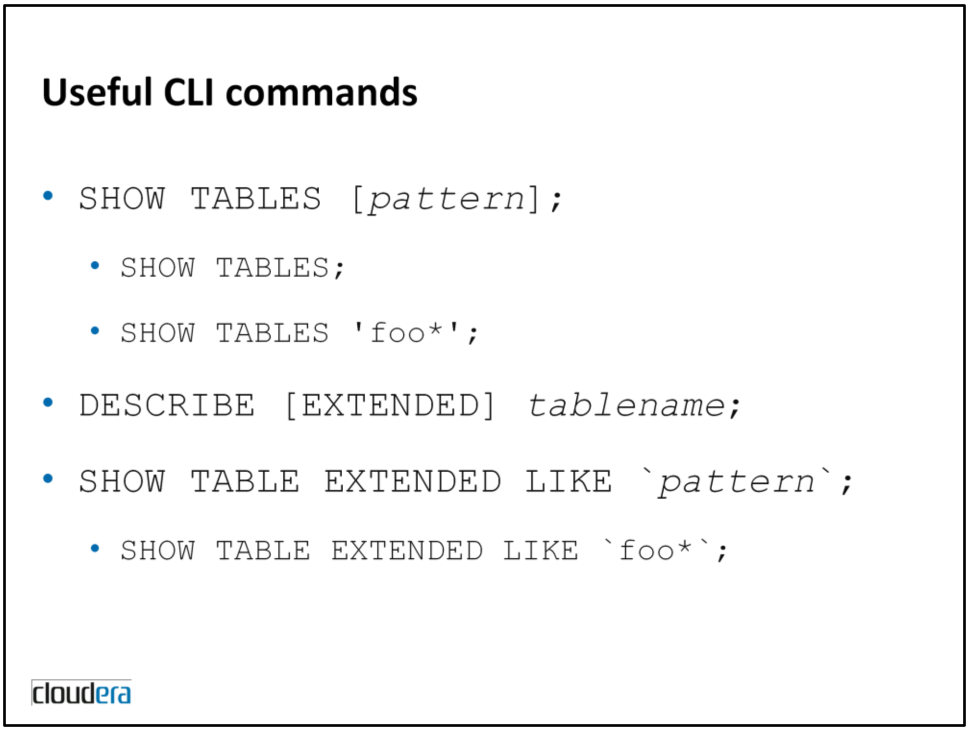
Invoke the Hive CLI



```
training@training-vm:~$ hive
Hive history file=/tmp/training/hive_job_log_training_201004231537_1243354076.txt
hive> SHOW TABLES;
OK
ip_locations
Time taken: 3.026 seconds
hive> quit;
training@training-vm:~$
```

To invoke the Hive command-line interface, execute the `hive` binary at a terminal. This requires that $HIVE_HOME/bin be in the path. Once started, Hive will print a message about the history file location and give a prompt:

```
hive>
```



Useful CLI commands

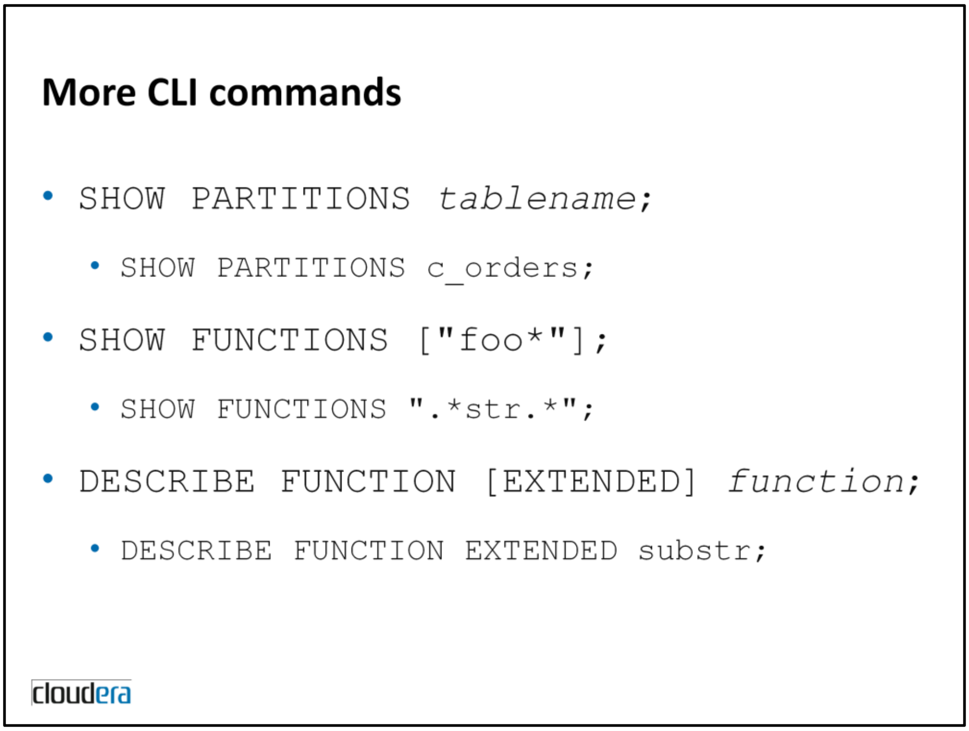
- SHOW TABLES [*pattern*];
  - SHOW TABLES;
  - SHOW TABLES 'foo*';
- DESCRIBE [EXTENDED] *tablename*;
- SHOW TABLE EXTENDED LIKE `` `*pattern*` ``;
  - SHOW TABLE EXTENDED LIKE `` `foo*` ``;

cloudera

The Hive command-line interface (CLI) is used to execute queries, but it can do other things too. To list all the tables in Hive, type SHOW TABLES. In version 0.5 is also possible to list tables that match a regular expression: `SHOW TABLES 'foo*'` (this will list all tables that start with "foo"). The regular expression can contain '*' for any character(s) or "|" for a choice.

To see the structure of a particular table, use `DESCRIBE` *`tablename`*. `DESCRIBE EXTENDED` *`tablename`* will add additional information about the table definition, such as the owner of the table (Carl, is this used for anything?) and what input format is used to read the data.

`SHOW TABLE EXTENDED` gives information about the matching tables such as the number of files for that table, total file size, and partition information. Note that the pattern should use backticks, not quotes. This command is available in 0.5.



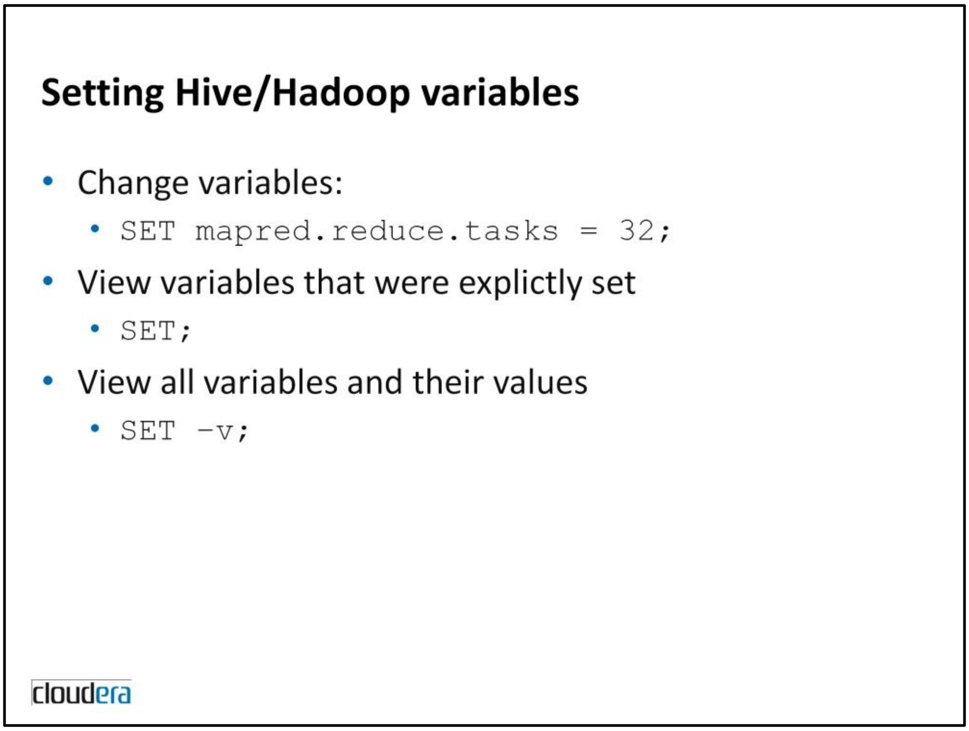
More CLI commands

- SHOW PARTITIONS *tablename*;
  - SHOW PARTITIONS c_orders;
- SHOW FUNCTIONS ["foo*"];
  - SHOW FUNCTIONS ".*str.*";
- DESCRIBE FUNCTION [EXTENDED] *function*;
  - DESCRIBE FUNCTION EXTENDED substr;

SHOW PARTITIONS returns a list of partitions for a table.

SHOW FUNCTIONS will list the functions available. It is also possible to list all functions that match a regular expression. For example, SHOW FUNCTIONS ".*str.*" will find all functions that contain the string "str".

For information about a specific function, use DESCRIBE FUNCTION. To see more verbose information (such as examples), include the keyword EXTENDED.



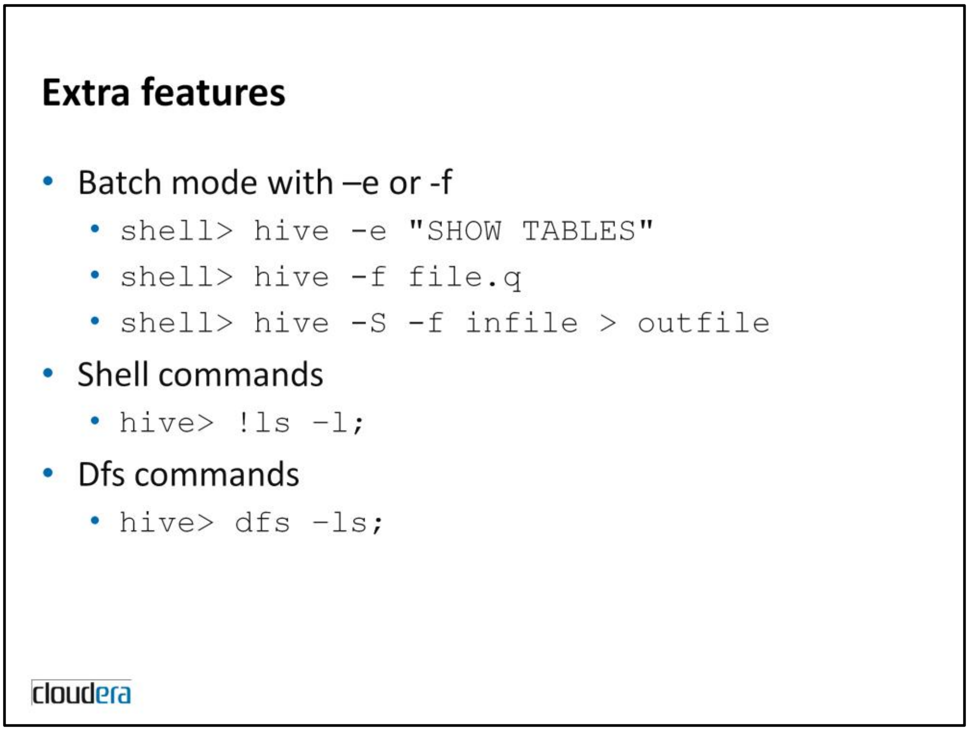
Setting Hive/Hadoop variables

- Change variables:
  - `SET mapred.reduce.tasks = 32;`
- View variables that were explictly set
  - `SET;`
- View all variables and their values
  - `SET -v;`

cloudera

The SET command can be used to set configuration variables. These variables control a variety of things, such as the location of the cluster and the location of the metastore. For example, "`SET mapred.reduce.tasks = 32`" will set the number of reducers in Hadoop's MapReduce to 32.

The "`SET;`" command will list only variables that have been explicitly set. Use "`SET -v;`" to list all properties and their values.



Extra features

- Batch mode with –e or -f
  - `shell> hive -e "SHOW TABLES"`
  - `shell> hive -f file.q`
  - `shell> hive -S -f infile > outfile`
- Shell commands
  - `hive> !ls -l;`
- Dfs commands
  - `hive> dfs -ls;`

cloudera

The CLI can also execute in batch mode. The –e option will start the CLI, execute a command, print the result and exit:

```
shell> hive -e "SHOW TABLES"
```

The –f option will read and execute commands in a file. This is also useful in combination with redirecting the output to a file using a ">" symbol. To suppress extra Hive messages, use the silent option (-S).

The Hive CLI also has the ability to pass commands to the OS shell. This is useful for executing an OS command without exiting the Hive CLI.

Furthermore, Hive can pass commands to HDFS. The Hadoop shell gives you this functionality, for example:

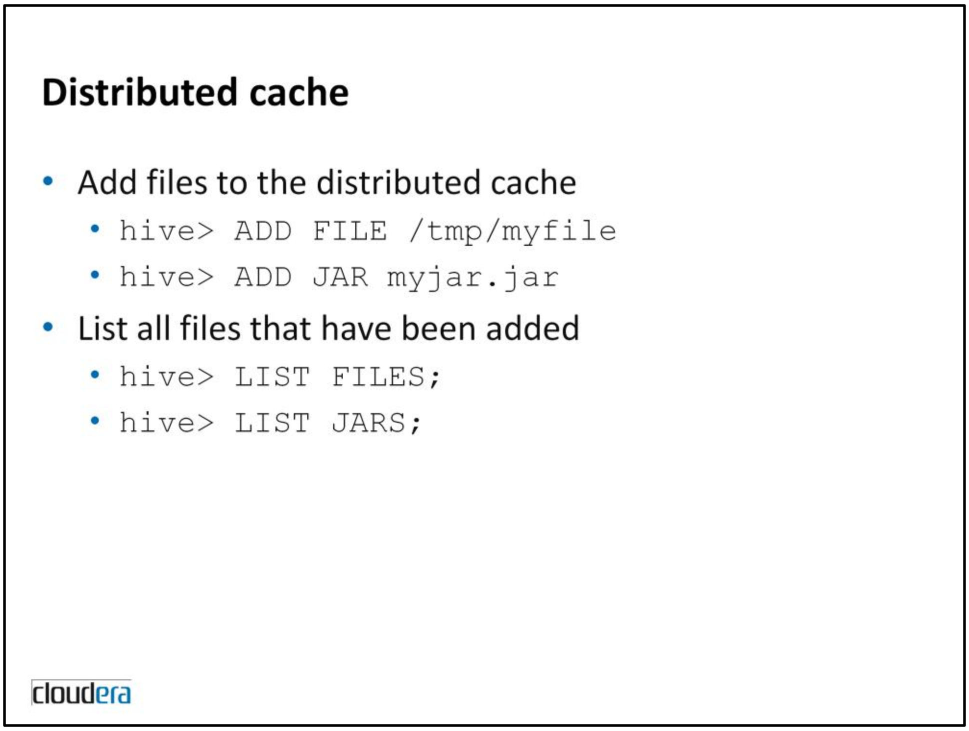
```
shell> hadoop fs -ls;
```

Therefore, it is possible to use:

```
hive> !hadoop fs -ls;
```

A shortcut is

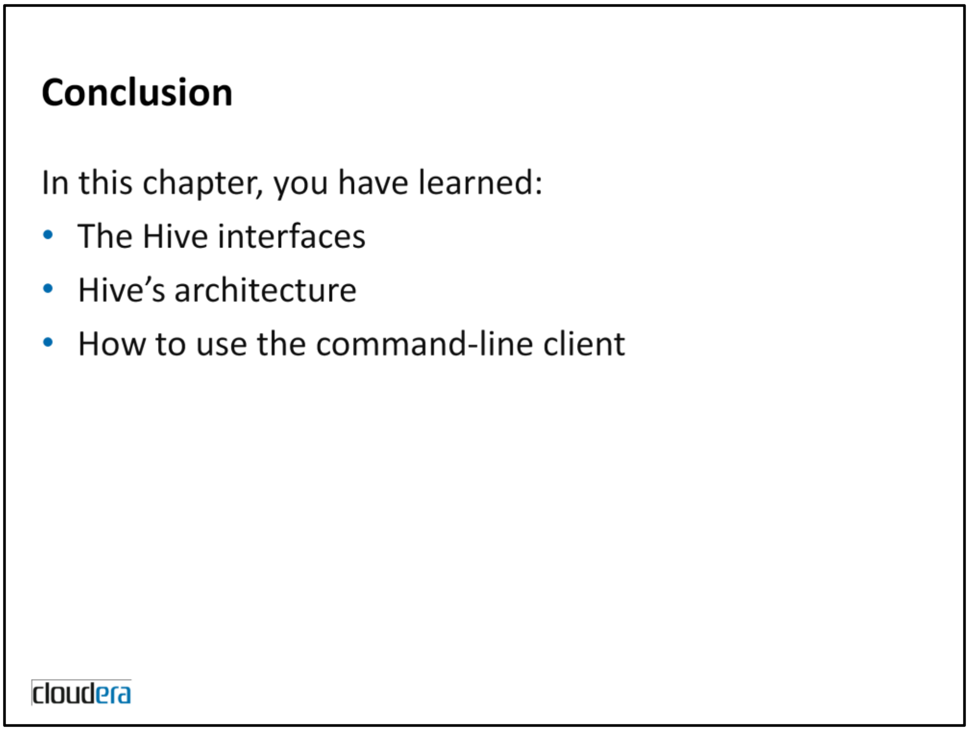
```
hive> dfs -ls;
```



Distributed cache

- Add files to the distributed cache
  - `hive> ADD FILE /tmp/myfile`
  - `hive> ADD JAR myjar.jar`
- List all files that have been added
  - `hive> LIST FILES;`
  - `hive> LIST JARS;`

The distributed cache is a Hadoop feature that allows a job to distribute a file or jar to each node in the cluster. In order to add files (or jars) to the distributed cache, use `ADD FILE|JAR`. `LIST FILES|JARS` will return all files or jars that have already been added to the distributed cache in this Hive session. All files or jars will be available for the current session only.



Conclusion

In this chapter, you have learned:

- The Hive interfaces
- Hive’s architecture
- How to use the command-line client