Sqoop Job

...

Automating things

IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

Trendytech Insights legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.

why a Sqoop Job?

Saved jobs remember the parameters used to specify a job, so they can be re-executed by invoking the job.

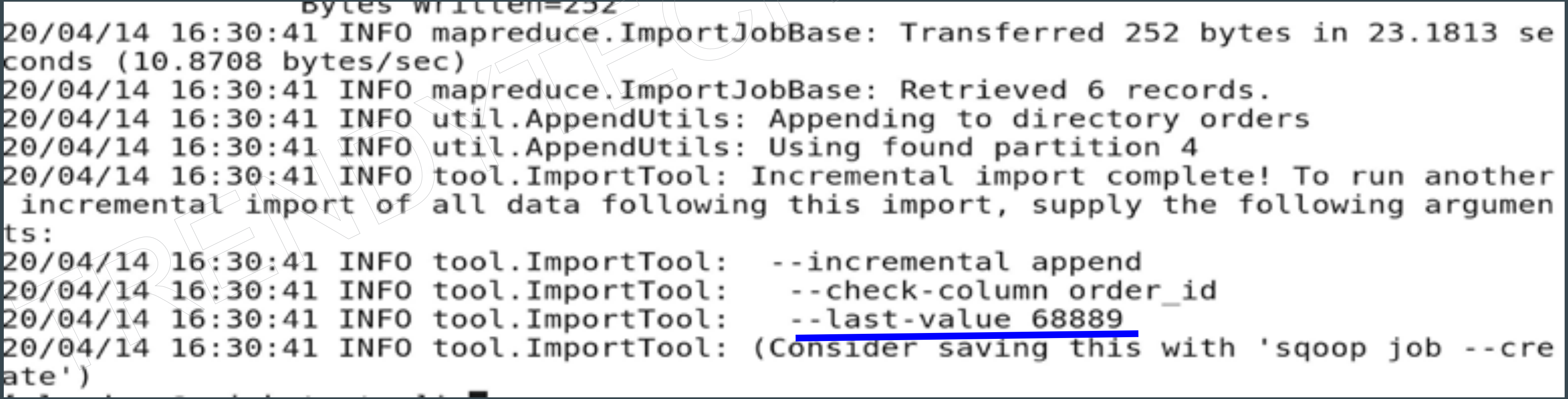
If a saved job is configured to perform an incremental import, state regarding the most recently imported rows is updated in the saved job to allow the job to continually import only the newest rows.

TRENDYTECH 9108179578

What was the pain point?

In the previous session we were doing incremental import.

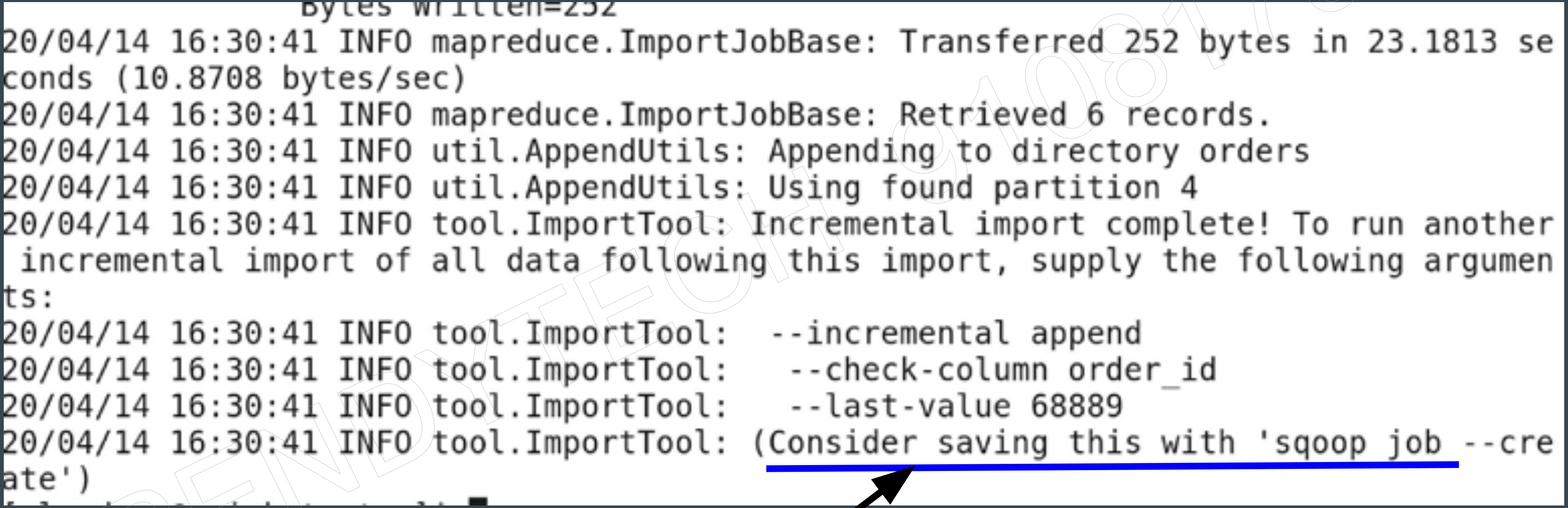
We have to manually keep a track of last value and then supply that during the next run. This was a manual process & was a big pain point.



```
                Bytes Written=252
20/04/14 16:30:41 INFO mapreduce.ImportJobBase: Transferred 252 bytes in 23.1813 se
conds (10.8708 bytes/sec)
20/04/14 16:30:41 INFO mapreduce.ImportJobBase: Retrieved 6 records.
20/04/14 16:30:41 INFO util.AppendUtils: Appending to directory orders
20/04/14 16:30:41 INFO util.AppendUtils: Using found partition 4
20/04/14 16:30:41 INFO tool.ImportTool: Incremental import complete! To run another
 incremental import of all data following this import, supply the following argumen
ts:
20/04/14 16:30:41 INFO tool.ImportTool:  --incremental append
20/04/14 16:30:41 INFO tool.ImportTool:   --check-column order_id
20/04/14 16:30:41 INFO tool.ImportTool:   --last-value 68889
20/04/14 16:30:41 INFO tool.ImportTool: (Consider saving this with 'sqoop job --cre
ate')
```

TRENDYTECH 9108179578

The solution to the problem is create a Sqoop Job



```
                Bytes Written=252
20/04/14 16:30:41 INFO mapreduce.ImportJobBase: Transferred 252 bytes in 23.1813 se
conds (10.8708 bytes/sec)
20/04/14 16:30:41 INFO mapreduce.ImportJobBase: Retrieved 6 records.
20/04/14 16:30:41 INFO util.AppendUtils: Appending to directory orders
20/04/14 16:30:41 INFO util.AppendUtils: Using found partition 4
20/04/14 16:30:41 INFO tool.ImportTool: Incremental import complete! To run another
 incremental import of all data following this import, supply the following argumen
ts:
20/04/14 16:30:41 INFO tool.ImportTool:  --incremental append
20/04/14 16:30:41 INFO tool.ImportTool:   --check-column order_id
20/04/14 16:30:41 INFO tool.ImportTool:   --last-value 68889
20/04/14 16:30:41 INFO tool.ImportTool: (Consider saving this with 'sqoop job --cre
ate')
```

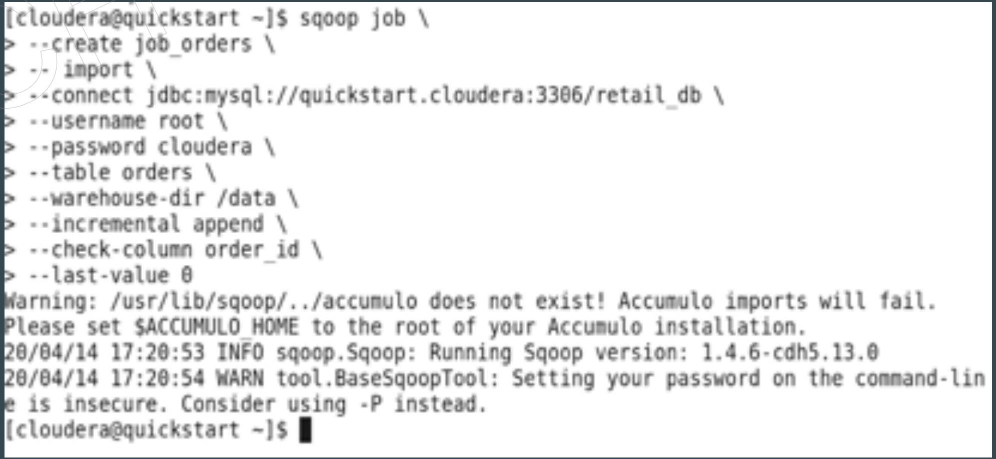
Even sqoop recommends that

TRENDYTECH 9108179578

creating a Sqoop Job (for incremental import)

```
sqoop job \
--create job_orders \
-- import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username root \
--password cloudera \
--table orders \
--warehouse-dir /data \
--incremental append \
--check-column order_id \
--last-value 0
```

Note: there is a space here between -- & import

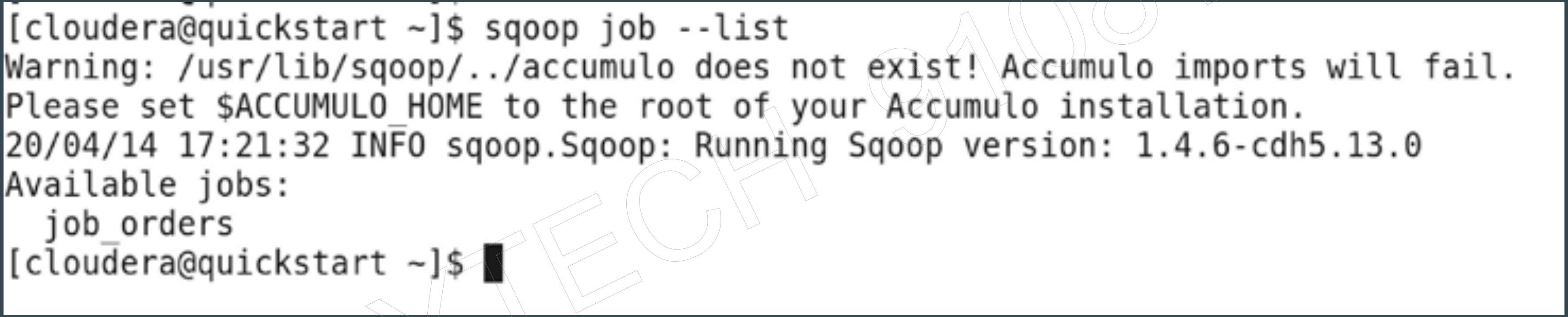


```
[cloudera@quickstart ~]$ sqoop job \
> --create job_orders \
> -- import \
> --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
> --username root \
> --password cloudera \
> --table orders \
> --warehouse-dir /data \
> --incremental append \
> --check-column order_id \
> --last-value 0
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:20:53 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
20/04/14 17:20:54 WARN tool.BaseSqoopTool: Setting your password on the command-lin
e is insecure. Consider using -P instead.
[cloudera@quickstart ~]$ 
```

TRENDYTECH 9108179578

See the list of sqoop jobs

sqoop job --list



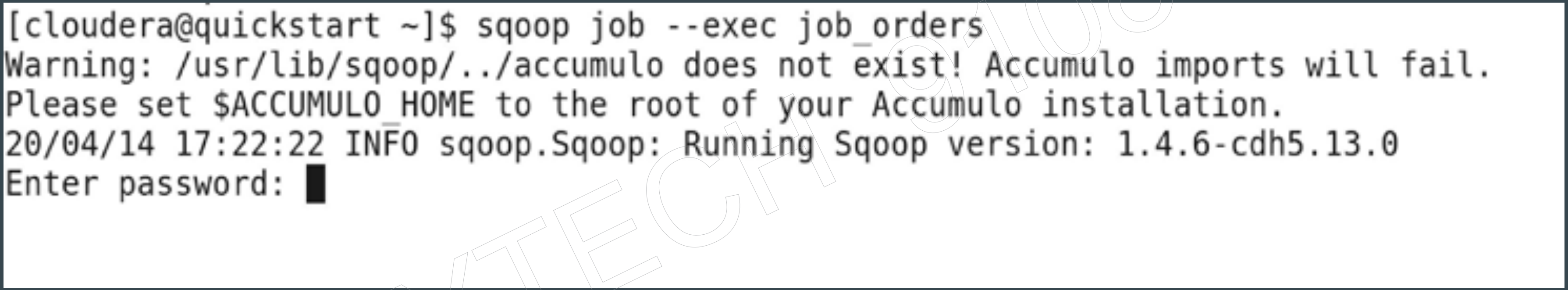
```
[cloudera@quickstart ~]$ sqoop job --list
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:21:32 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Available jobs:
  job_orders
[cloudera@quickstart ~]$ 
```

This command will show us all the created sqoop jobs.

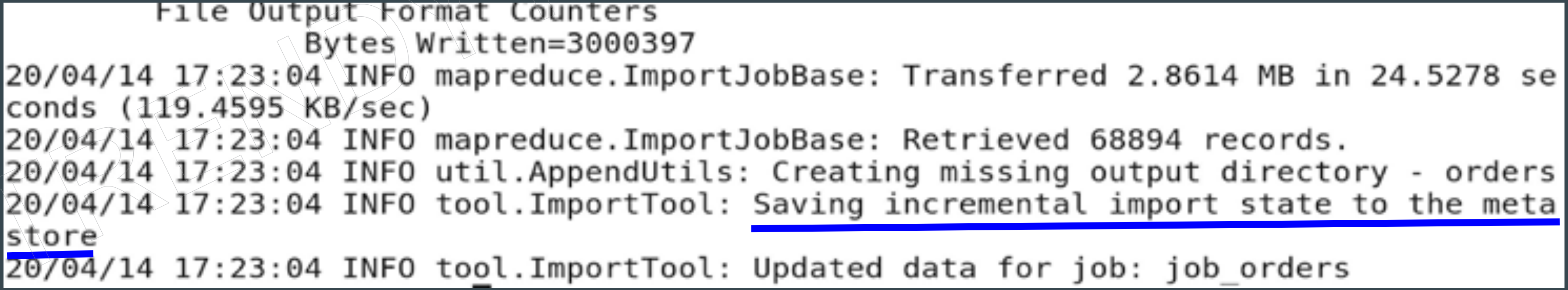
TRENDYTECH 9108179578

Executing the sqoop job

sqoop job --exec job_orders



```
[cloudera@quickstart ~]$ sqoop job --exec job_orders
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:22:22 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
```

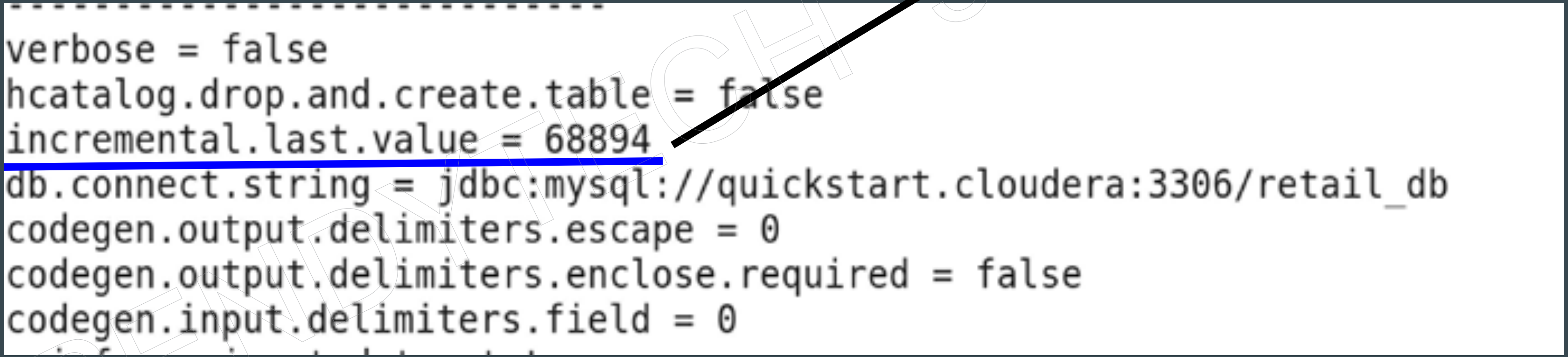


```
        File Output Format Counters
                Bytes Written=3000397
20/04/14 17:23:04 INFO mapreduce.ImportJobBase: Transferred 2.8614 MB in 24.5278 se
conds (119.4595 KB/sec)
20/04/14 17:23:04 INFO mapreduce.ImportJobBase: Retrieved 68894 records.
20/04/14 17:23:04 INFO util.AppendUtils: Creating missing output directory - orders
20/04/14 17:23:04 INFO tool.ImportTool: Saving incremental import state to the meta
store
20/04/14 17:23:04 INFO tool.ImportTool: Updated data for job: job_orders
```

TRENDYTECH 9108179578

Checking the saved state of the job

sqoop job --show job_orders



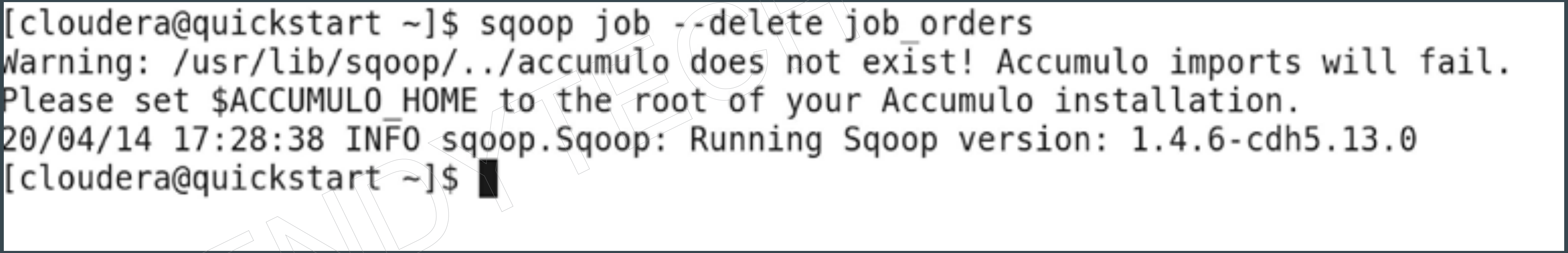
```
verbose = false
hcatalog.drop.and.create.table = false
incremental.last.value = 68894
db.connect.string = jdbc:mysql://quickstart.cloudera:3306/retail_db
codegen.output.delimiters.escape = 0
codegen.output.delimiters.enclose.required = false
codegen.input.delimiters.field = 0
```

The current job state is saved for the next run.

TRENDYTECH 9108179578

Deleting a Sqoop Job

sqoop job --delete job_orders

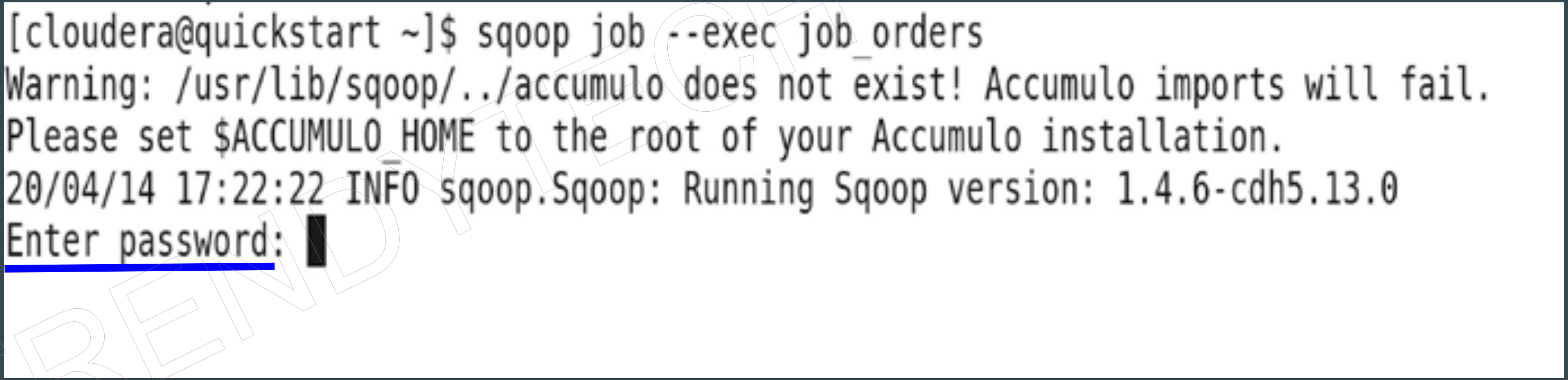


```
[cloudera@quickstart ~]$ sqoop job --delete job_orders
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:28:38 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
[cloudera@quickstart ~]$ 
```

TRENDYTECH 9108179578

But the job is not completely automated. why?

Because it asks for password when we run the sqoop job.



```
[cloudera@quickstart ~]$ sqoop job --exec job_orders
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:22:22 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
```

TRENDYTECH 9108179578

What is the solution to the previous problem?

A **password file** can help in this case. This means that our password (to connect to Database) will be stored in a file. We need to specify the file path so that we do not have to provide the password manually at runtime.

How to create password file

```
echo -n "cloudera" >> .password-file
```

If you try creating file using normal ways then some special characters gets appended at the end of password. Which will lead to incorrect password error.

Note: In the above command the name of file is password-file. Also "." before the filename indicates its a hidden file. So the output of echo (cloudera) is stored in a hidden file named as password-file.

TRENDYTECH 9108179578

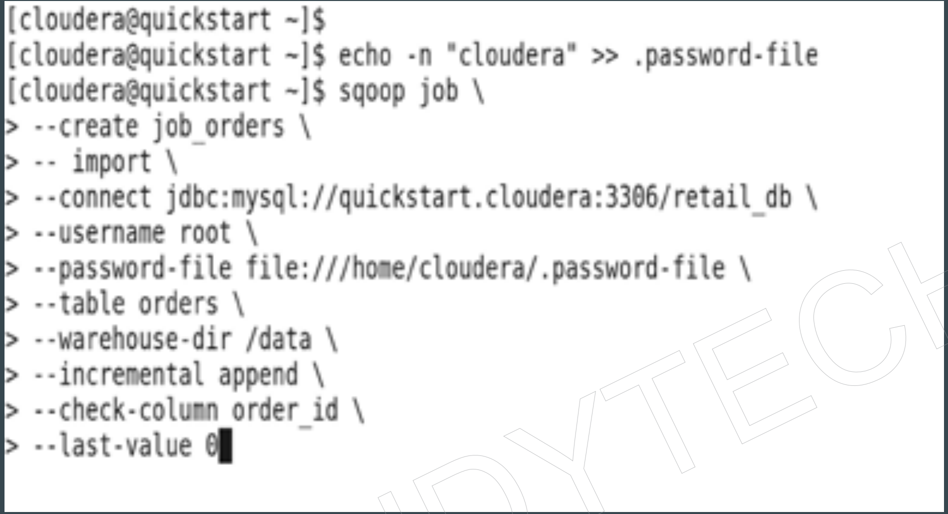
re-create the sqoop job which uses the password file.

```
sqoop job \
--create job_orders \
-- import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username root \
--password-file file:///home/cloudera/.password-file \
--table orders \
--warehouse-dir /data \
--incremental append \
--check-column order_id \
--last-value 0
```

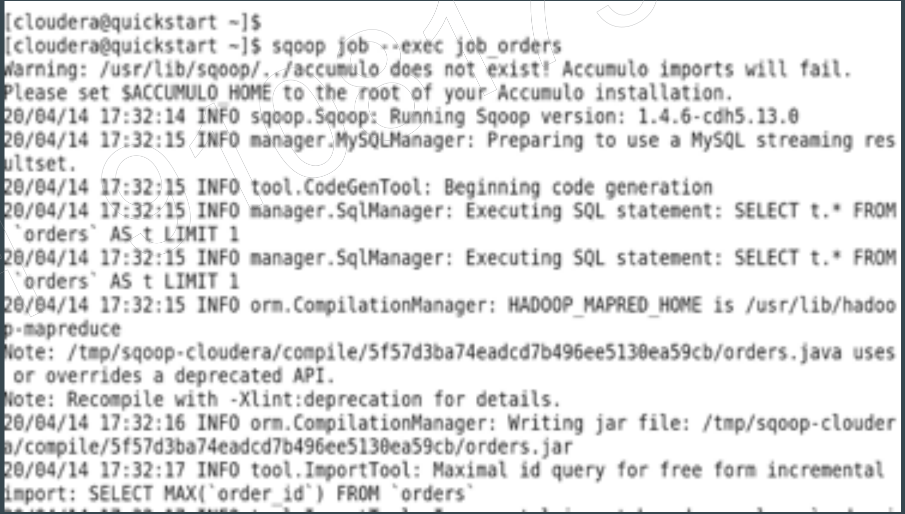
file:// indicates that the password file is in local, and not in HDFS. if you do not mention file:// then it will expect the file in HDFS.

TRENDYTECH 9108179578

Run the job & this time it wont ask for password.



```
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ echo -n "cloudera" >> .password-file
[cloudera@quickstart ~]$ sqoop job \
> --create job_orders \
> -- import \
> --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
> --username root \
> --password-file file:///home/cloudera/.password-file \
> --table orders \
> --warehouse-dir /data \
> --incremental append \
> --check-column order_id \
> --last-value 0
```



```
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ sqoop job --exec job_orders
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:32:14 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
20/04/14 17:32:15 INFO manager.MySQLManager: Preparing to use a MySQL streaming res
ultset.
20/04/14 17:32:15 INFO tool.CodeGenTool: Beginning code generation
20/04/14 17:32:15 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM
 `orders` AS t LIMIT 1
20/04/14 17:32:15 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM
 `orders` AS t LIMIT 1
20/04/14 17:32:15 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoo
p-mapreduce
Note: /tmp/sqoop-cloudera/compile/5f57d3ba74eadcd7b496ee5130ea59cb/orders.java uses
 or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/04/14 17:32:16 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-clouder
a/compile/5f57d3ba74eadcd7b496ee5130ea59cb/orders.jar
20/04/14 17:32:17 INFO tool.ImportTool: Maximal id query for free form incremental
import: SELECT MAX(`order_id`) FROM `orders`
```

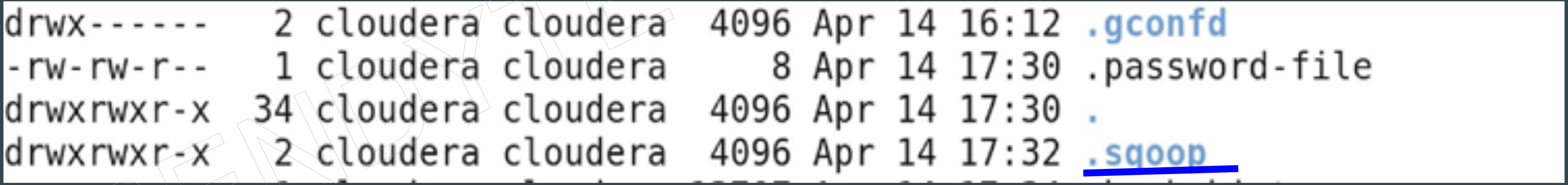
Run the job again using: sqoop job --exec job_orders
We can see that during execution it does not ask for password anymore.

TRENDYTECH 9108179578

Where is the state of job stored?

The state of job is stored locally in a hidden folder named (.sqoop) which resides in home directory (/home/cloudera)

ls -altr /home/cloudera (here **a** is to list all files including the hidden files)



```
drwx------   2 cloudera cloudera  4096 Apr 14 16:12 .gconfd
-rw-rw-r--   1 cloudera cloudera     8 Apr 14 17:30 .password-file
drwxrwxr-x  34 cloudera cloudera  4096 Apr 14 17:30 .
drwxrwxr-x   2 cloudera cloudera  4096 Apr 14 17:32 .sqoop
```

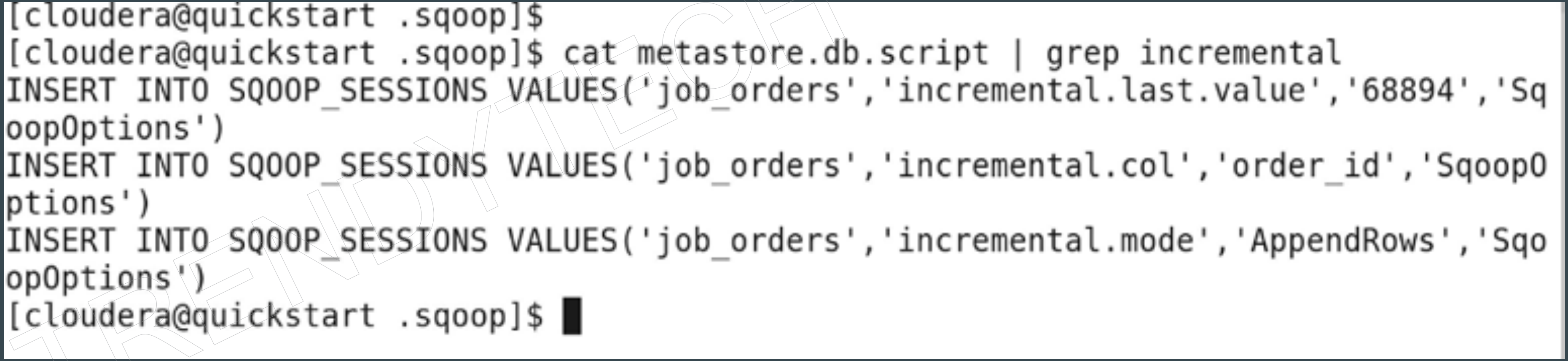
TRENDYTECH 9108179578

Where is the state of job stored?

cd .sqoop

cat metastore.db.script | grep incremental

We are piping the results to grep and searching for incremental keyword



```
[cloudera@quickstart .sqoop]$
[cloudera@quickstart .sqoop]$ cat metastore.db.script | grep incremental
INSERT INTO SQOOP_SESSIONS VALUES('job_orders','incremental.last.value','68894','Sq
oopOptions')
INSERT INTO SQOOP_SESSIONS VALUES('job_orders','incremental.col','order_id','SqoopO
ptions')
INSERT INTO SQOOP_SESSIONS VALUES('job_orders','incremental.mode','AppendRows','Sqo
opOptions')
[cloudera@quickstart .sqoop]$ █
```

TRENDYTECH 9108179578

What if someone gets hold of your password-file?

This can lead to password being leaked out & it’s a serious concern.

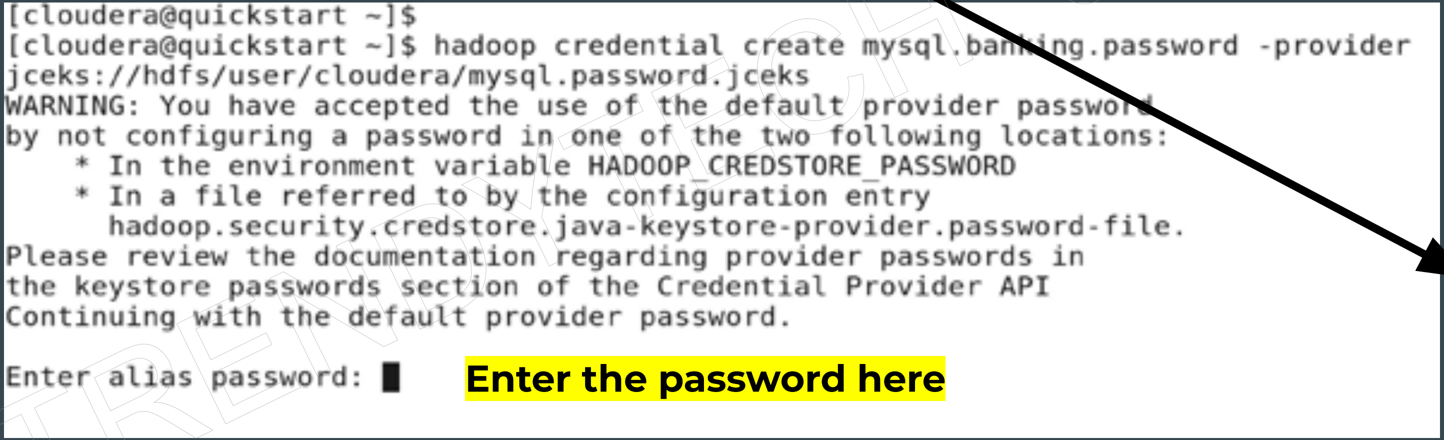
So what is the solution to this?

Here comes, the concept of **password alias using encrypted password file with jceks (java cryptography encryption key store)**

TRENDYTECH 9108179578

How to create a password alias

hadoop credential create **mysql.banking.password** -provider **jceks://hdfs/user/cloudera**/mysql.password.jceks



```
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ hadoop credential create mysql.banking.password -provider
jceks://hdfs/user/cloudera/mysql.password.jceks
WARNING: You have accepted the use of the default provider password
by not configuring a password in one of the two following locations:
    * In the environment variable HADOOP_CREDSTORE_PASSWORD
    * In a file referred to by the configuration entry
      hadoop.security.credstore.java-keystore-provider.password-file.
Please review the documentation regarding provider passwords in
the keystore passwords section of the Credential Provider API
Continuing with the default provider password.

Enter alias password: █
```

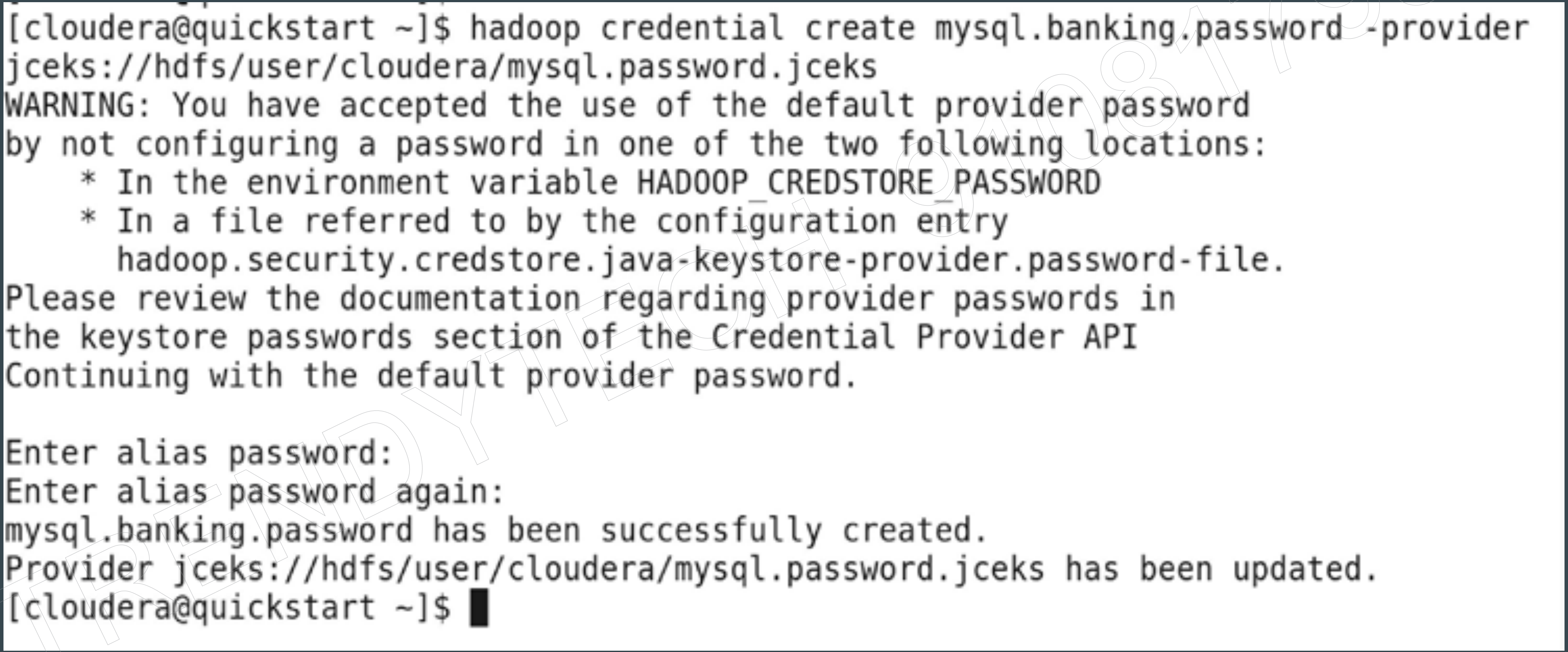
Enter the password here

Name of password alias

Location where password is stored in encrypted form.

TRENDYTECH 9108179578

Password alias created successfully



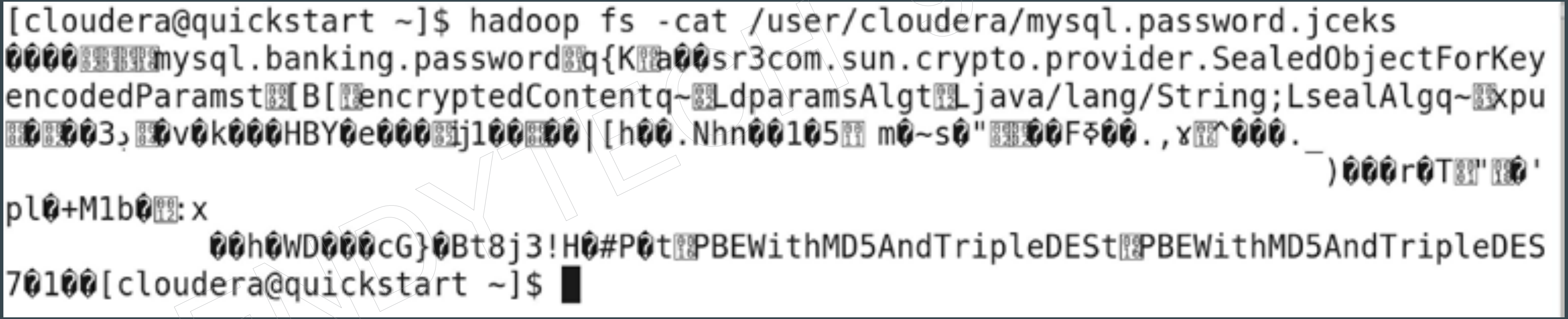
```
[cloudera@quickstart ~]$ hadoop credential create mysql.banking.password -provider
jceks://hdfs/user/cloudera/mysql.password.jceks
WARNING: You have accepted the use of the default provider password
by not configuring a password in one of the two following locations:
    * In the environment variable HADOOP_CREDSTORE_PASSWORD
    * In a file referred to by the configuration entry
      hadoop.security.credstore.java-keystore-provider.password-file.
Please review the documentation regarding provider passwords in
the keystore passwords section of the Credential Provider API
Continuing with the default provider password.

Enter alias password:
Enter alias password again:
mysql.banking.password has been successfully created.
Provider jceks://hdfs/user/cloudera/mysql.password.jceks has been updated.
[cloudera@quickstart ~]$
```

TRENDYTECH 9108179578

Check the content of your file

hadoop fs -cat /user/cloudera/mysql.password.jceks



```
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/mysql.password.jceks
����mysql.banking.password q{K a��sr3com.sun.crypto.provider.SealedObjectForKey
encodedParamst [B[ encryptedContentq~ LdparamsAlgt Ljava/lang/String;LsealAlgq~ xpu
 � ��3, �v�k���HBY�e��� j1�� ��|[h��.Nhn��1�5  m�~s�" ��F���.,ɤ ^���._
                                                                    )���r�T " �'
pl�+M1b� :x
          ��h�WD���cG}�Bt8j3!H�#P�t PBEWithMD5AndTripleDESt PBEWithMD5AndTripleDES
7�1��[cloudera@quickstart ~]$ █
```

TRENDYTECH 9108179578

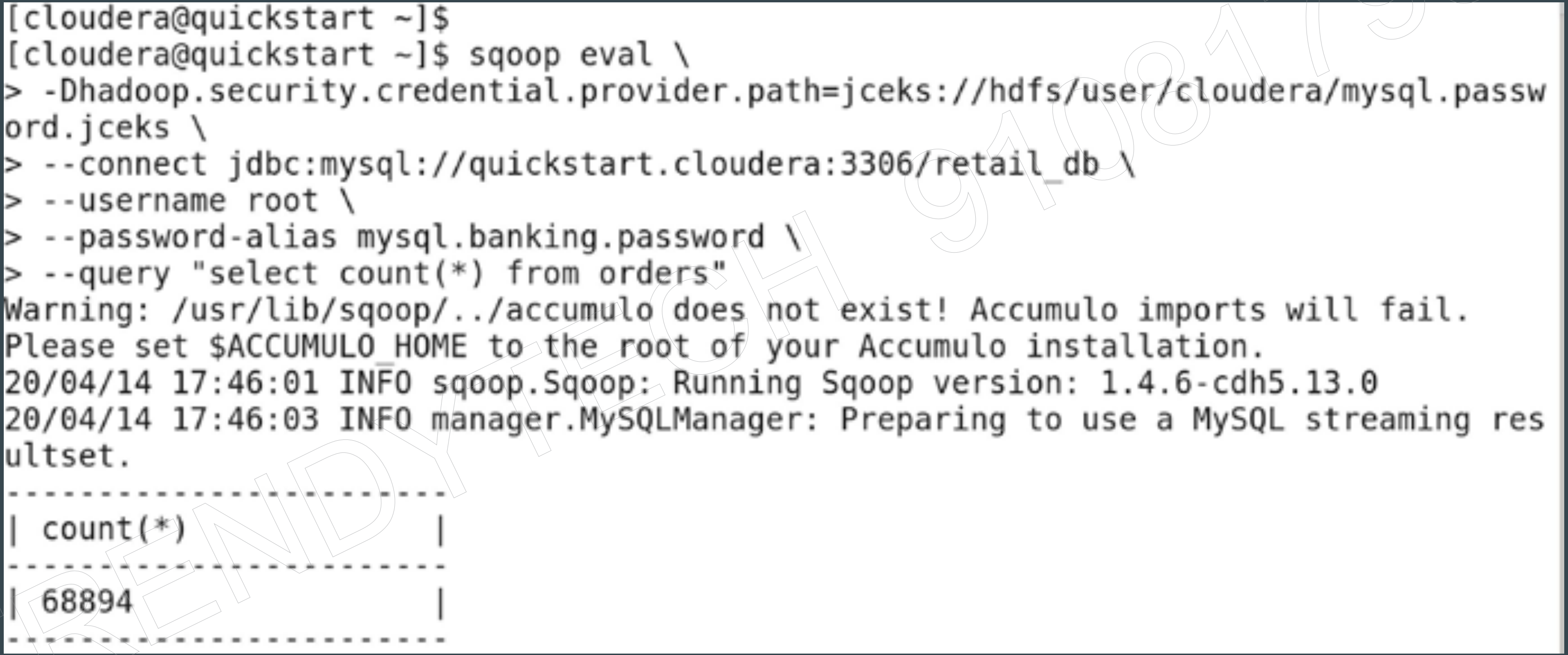
Use the password-alias in your sqoop command

A sqoop eval job to count the number of rows in orders table.

```
sqoop eval \
-Dhadoop.security.credential.provider.path=jceks://hdfs/user/cloudera/
mysql.password.jceks \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username root \
--password-alias mysql.banking.password \
--query "select count(*) from orders"
```

TRENDYTECH 9108179578

Output of Sqoop eval command



```
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ sqoop eval \
> -Dhadoop.security.credential.provider.path=jceks://hdfs/user/cloudera/mysql.passw
ord.jceks \
> --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
> --username root \
> --password-alias mysql.banking.password \
> --query "select count(*) from orders"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/14 17:46:01 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
20/04/14 17:46:03 INFO manager.MySQLManager: Preparing to use a MySQL streaming res
ultset.
-----------------------
| count(*)            |
-----------------------
| 68894               |
-----------------------
```

TRENDY TECH 9108179578

4 ways to pass the password

1. In the command using --password
2. Passing the parameter at runtime using -P
3. Storing the password in password file
4. Password alias with encrypted password

TRENDYTECH 9108179578

We learnt to Create Sqoop job & how to handle password

Happy Learning!!!

TRENDYTECH 9108179578

Google REVIEWS

5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT

9108179578

Call for more details

Follow US

Trainer	Mr. Sumit Mittal
Phone	9108179578
Email	trendytech.sumit@gmail.com
Website	https://trendytech.in/courses/big-data-online-training/
LinkedIn	https://www.linkedin.com/in/bigdatabysumit/
Twitter	@BigdataBySumit
Instagram	bigdatabysumit
Facebook	https://www.facebook.com/trendytech.in/
Youtube	TrendyTech