



CCA175^{Q&As}

CCA Spark and Hadoop Developer Exam

Pass Cloudera CCA175 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/cca175.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cloudera
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

Problem Scenario 59 : You have been given below code snippet.

```
val x = sc.parallelize(1 to 20)
```

```
val y = sc.parallelize(10 to 30) operationl
```

```
z.collect
```

Write a correct code snippet for operationl which will produce desired output, shown below.

```
Array[Int] = Array(16,12, 20,13,17,14,18,10,19,15,11)
```

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

```
val z = x.intersection(y)
```

intersection : Returns the elements in the two RDDs which are the same.

QUESTION 2

Problem Scenario 25 : You have been given below comma separated employee information. That needs to be added in /home/cloudera/flumetest/in.txt file (to do tail source) sex,name,city 1,alok,mumbai 1,jatin,chennai 1,yogesh,kolkata 2,ragini,delhi 2,jyotsana,pune 1,valmiki,banglore Create a flume conf file using fastest non-durable channel, which write data in hive warehouse directory, in two separate tables called flumemaleemployee1 and flumefemaleemployee1 (Create hive table as well for given data). Please use tail source with /home/cloudera/flumetest/in.txt file.

Flumemaleemployee1 : will contain only male employees data flumefemaleemployee1 : Will contain only woman employees data

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Create hive table for flumemaleemployee1 and .\`

```
CREATE TABLE flumemaleemployee1
```

```
(
```

```
sex_type int, name string, city string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\`,`; CREATE TABLE flumefemaleemployee1 ( sex_type int, name string, city string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\`,`;
```

Step 2 : Create below directory and file mkdir /home/cloudera/flumetest/ cd /home/cloudera/flumetest/ Step 3 :

Create flume configuration file, with below configuration for source, sink and channel and save it in flume5.conf.

```
agent.sources = tailsrc agent.channels = mem1 mem2 agent.sinks = stdl std2 agent.sources.tailsrc.type = exec
```

```
agent.sources.tailsrc.command = tail -F /home/cloudera/flumetest/in.txt agent.sources.tailsrc.batchSize = 1
```

```
agent.sources.tailsrc.interceptors = i1 agent.sources.tailsrc.interceptors.i1.type = regex_extractor
```

```
agent.sources.tailsrc.interceptors.il.regex = A(\d} agent.sources.tailsrc.interceptors.M.serializers = t1
```

```
agent.sources.tailsrc.interceptors.i1.serializers.t1.name = type agent.sources.tailsrc.selector.type = multiplexing
```

```
agent.sources.tailsrc.selector.header = type agent.sources.tailsrc.selector.mapping.1 = mem1
```

```
agent.sources.tailsrc.selector.mapping.2 = mem2 agent.sinks.std1.type = hdfs agent.sinks.std1.channel = mem1
```



```
agent.sinks.std1.batchSize = 1 agent.sinks.std1.hdfs.path = /user/hive/warehouse/flumemaleemployeei
agent.sinks.std1.rollInterval = 0 agent.sinks.std1.hdfs.tileType = Data Stream agent.sinks.std2.type = hdfs
agent.sinks.std2.channel = mem2 agent.sinks.std2.batchSize = 1 agent.sinks.std2.hdfs.path = /user/hive/warehouse/flumefemaleemployee1 agent.sinks.std2.rollInterval = 0 agent.sinks.std2.hdfs.tileType = Data Stream
agent.channels.mem1.type = memory agent.channels.mem1.capacity = 100 agent.channels.mem2.type = memory
agent.channels.mem2.capacity = 100 agent.sources.tailsrc.channels = mem1 mem2 Step 4 : Run below command
which will use this configuration file and append data in hdfs. Start flume service: flume-ng agent -conf
/home/cloudera/flumeconf -conf-file /home/cloudera/flumeconf/flume5.conf --name agent Step 5 : Open another terminal
create a file at /home/cloudera/flumetest/in.txt. Step 6 : Enter below data in file and save it. l.alok.mumbai 1 jatin.chennai
1,yogesh,kolkata 2,ragini,delhi 2,jyotsana,pune 1,valmiki,banglore Step 7 : Open hue and check the data is available in
hive table or not. Step 8 : Stop flume service by pressing ctrl+c
```

QUESTION 3

Problem Scenario 73 : You have been given data in json format as below.

```
{"first_name":"Ankit", "last_name":"Jain"}
{"first_name":"Amir", "last_name":"Khan"}
{"first_name":"Rajesh", "last_name":"Khanna"}
{"first_name":"Priynka", "last_name":"Chopra"}
{"first_name":"Kareena", "last_name":"Kapoor"}
{"first_name":"Lokesh", "last_name":"Yadav"}
```

Do the following activity

1.
create employee.json file locally.
2.
Load this file on hdfs
3.
Register this data as a temp table in Spark using Python.
4.
Write select query and print this data.
5.
Now save back this selected data in json format.

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : create employee.json file locally.



vi employee.json (press insert) past the content.

Step 2 : Upload this tile to hdfs, default location hadoop fs -put employee.json

Step 3 : Write spark script

```
#Import SQLContext
```

```
from pyspark import SQLContext
```

```
#Create instance of SQLContext sqlContext = SQLContext(sc)
```

```
#Load json file
```

```
employee = sqlContext.jsonFile("employee.json")
```

```
#Register RDD as a temp table employee.registerTempTablef\\"EmployeeTab"}
```

```
#Select data from Employee table
```

```
employeeInfo = sqlContext.sql("select * from EmployeeTab")
```

```
#Iterate data and print
```

```
for row in employeeInfo.collect():
```

```
print(row)
```

Step 4 : Write dataas a Text file employeeInfo.toJSON().saveAsTextFile("employeeJson1") Step 5: Check whether data has been created or not hadoop fs -cat employeeJson/part"

QUESTION 4

Problem Scenario 94 : You have to run your Spark application on yarn with each executor

20GB and number of executors should be 50. Please replace XXX, YYY, ZZZ

```
export HADOOP_CONF_DIR=XXX
```

```
./bin/spark-submit \
```

```
-class com.hadoopexam.MyTask \
```

```
xxx\
```

```
-deploy-mode cluster \ # can be client for client mode
```

```
YYY\
```

```
222 \
```

```
/path/to/hadoopexam.jar \
```

```
1000
```



Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution

XXX: -master yarn YYY : -executor-memory 20G ZZZ: -num-executors 50

QUESTION 5

Problem Scenario 13 : You have been given following mysql database details as well as other info. user=retail_dba password=cloudera database=retail_db jdbc URL = jdbc:mysql://quickstart:3306/retail_db Please accomplish following.

1.

Create a table in retaildb with following definition.

```
CREATE table departments_export (department_id int(11), department_name varchar(45),  
created_date T1MESTAMP DEFAULT NOWQ);
```

2.

Now import the data from following directory into departments_export table,

/user/cloudera/departments new

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Login to musql db

```
mysql --user=retail_dba -password=cloudera
```

```
show databases; use retail_db; show tables;
```

step 2 : Create a table as given in problem statement.

```
CREATE table departments_export (departmentid int(11), department_name varchar(45),  
created_date T1MESTAMP DEFAULT NOW());
```

```
show tables;
```

Step 3 : Export data from /user/cloudera/departmentsnew to new table departments_export

```
sqoop export -connect jdbc:mysql://quickstart:3306/retail_db \
```

```
-username retaildba \
```

```
--password cloudera \
```

```
--table departments_export \
```

```
-export-dir /user/cloudera/departments_new \
```



-batch

Step 4 : Now check the export is correctly done or not. mysql -user*retail_dba password=cloudera

show databases;

use retail_db;

show tables;

select\\' from departments_export;

[CCA175 PDF Dumps](#)

[CCA175 VCE Dumps](#)

[CCA175 Practice Test](#)