

Atividade de Aprofundamento Ingestão de Dados

Questão 1 - Trabalhando com o Sqoop

- 1) Crie uma tabela no banco de dados testeingestao chamada "marketing_banco"
 - a) No terminal digite: `mysql -u root -p`
 - b) Caso não exista vamos criar uma base chamada "testeingestao", digite:
`create database testeingestao;`
 - c) Agora criaremos a tabela "marketing_banco". Ainda no terminal digite todo o script a seguir:

`use testeingestao;`

```
CREATE TABLE marketing_banco(  
idade int not null,  
estadoCivil varchar(30) not null,  
trabalho varchar(50) not null,  
casa int not null,  
emprestimo int not null,  
campanha int not null,  
contato varchar(50) not null);
```

2. Essa tabela deve conter os seguintes campos:

- idade inteiro não nulo,
- estado civil varchar tamanho 30 não nulo,
- trabalho varchar tamanho 50 não nulo,
- casa inteiro não nulo,
- empréstimo inteiro não nulo,
- campanha inteiro não nulo
- contato varchar tamanho 50 não nulo

Baixe os dados no seguinte link: Salve eles em
"/home/cloudera/Downloads/"

<https://drive.google.com/file/d/1s5JotiouBon3DU7urq8JfDJeTtN2IZbY>

- 1) Agora você deve inserir os dados na tabela. Você pode inserir rapidamente no MySQL usando o seguinte comando LOAD conforme visto no tutorial do Sqooq.

```
load data local infile "/home/cloudera/Downloads/makt_banco.csv"   
into table marketing_banco fields terminated by ","   
lines terminated by "\n" ignore 1 lines;
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
mysql> use testeingestao;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
mysql> CREATE TABLE marketing_banco(  
-> idade int not null,  
-> estadoCivil varchar(30) not null,  
-> trabalho varchar(50) not null,  
-> casa int not null,  
-> emprestimo int not null,  
-> campanha int not null,  
-> contato varchar(50) not null  
-> );  
Query OK, 0 rows affected (0.08 sec)  
mysql> load data local infile '/home/cloudera/Downloads/makt_banco.csv'   
-> into table marketing_banco fields terminated by ','  
-> lines terminated by '\n' ignore 1 lines;  
Query OK, 45211 rows affected, 65535 warnings (0.33 sec)  
Records: 45211 Deleted: 0 Skipped: 0 Warnings: 316477  
mysql>
```

4. Importe os dados da tabela "marketing_banco" para o HDFS usando o Sqoop.

```
sqoop import --connect jdbc:mysql://localhost:3306/testeingestao --  
username root --password cloudera --table marketing_banco -m 1
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
Total time spent by all reduces in occupied slots (ms)=0  
Total time spent by all map tasks (ms)=17195  
Total vcore-milliseconds taken by all map tasks=17195  
Total megabyte-milliseconds taken by all map tasks=17607680  
Map-Reduce Framework  
Map Input records=45211  
Map output records=45211  
Input split bytes=87  
Spilled Records=0  
Failed Shuffles=0  
Merged Map outputs=0  
GC time elapsed (ms)=493  
CPU time spent (ms)=3510  
Physical memory (bytes) snapshot=121491456  
Virtual memory (bytes) snapshot=1511235584  
Total committed heap usage (bytes)=60751872  
File Input Format Counters  
Bytes Read=0  
File Output Format Counters  
Bytes Written=542532  
21/05/21 07:28:52 INFO mapreduce.ImportJobBase: Transferred 529.8164 KB in 48.26  
54 seconds (10.9771 KB/sec)  
21/05/21 07:28:52 INFO mapreduce.ImportJobBase: Retrieved 45211 records.  
[cloudera@quickstart ~]$
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
Spilled Records=0  
Failed Shuffles=0  
Merged Map outputs=0  
GC time elapsed (ms)=493  
CPU time spent (ms)=3510  
Physical memory (bytes) snapshot=121491456  
Virtual memory (bytes) snapshot=1511235584  
Total committed heap usage (bytes)=60751872  
File Input Format Counters  
Bytes Read=0  
File Output Format Counters  
Bytes Written=542532  
21/05/21 07:28:52 INFO mapreduce.ImportJobBase: Transferred 529.8164 KB in 48.26  
54 seconds (10.9771 KB/sec)  
21/05/21 07:28:52 INFO mapreduce.ImportJobBase: Retrieved 45211 records.  
[cloudera@quickstart ~]$ hdfs dfs -ls  
Found 6 items  
drwxr-xr-x - cloudera cloudera 0 2021-05-21 07:28 marketing_banco  
drwxr-xr-x - cloudera cloudera 0 2021-05-19 16:36 pessoas  
drwxr-xr-x - cloudera cloudera 0 2021-05-20 10:02 testeTune  
drwxr-xr-x - cloudera cloudera 0 2021-05-09 15:16 testehdfs  
drwxr-xr-x - cloudera cloudera 0 2021-05-12 16:07 testemapreduce  
drwxr-xr-x - cloudera cloudera 0 2021-05-20 05:13 testesqoop  
[cloudera@quickstart ~]$
```

Questão 2 - Trabalhando com o Flume

Vamos trabalhar com as informações fornecidas pela Agência de Informações de Energia dos Estados Unidos sobre o preço da energia por quilowatt-hora, por estado e por tipo de provedor. Para tanto, baixe o Excel com os dados do site:

<https://drive.google.com/open?id=14BvVk6LjKAKfYMF8aXviFI9ARfWT2Z7K>

2. Crie uma pasta local chamada "precos_us_energia":

```
mkdir precos_us_energia
```

Vamos criar duas pastas dentro da pasta local precos_us_energia: "dados" e "conf", usaremos a pasta conf para salvar nosso agente Flume e a pasta dados para construir o canal com o HDFS:

```
cd precos_us_energia
```

```
mkdir dados
```

```
mkdir conf
```

3. Crie uma pasta no HDFS chamada "dados_energia":

```
hdfs dfs -mkdir dados_energia
```

4. Agora você deve criar um agente Flume para enviar os dados para dados_energia do HDFS. Vamos por partes:

Source: crie a source com o tipo spooldir que aponta para a pasta "precos_us_energia". O spooldir observará o diretório especificado em busca de novos arquivos e fará o envio à medida que arquivos surgirem. Após um arquivo ter sido lido pelo canal ele será renomeado indicando que a tarefa foi concluída.

Vamos passo por passo:

✓ Abra o gedit e crie um arquivo chamado agente2.conf e salve na pasta precos_us_energia/conf

```
gedit agente2.conf
```

Em nosso agente2.conf vamos:

- ✓ Primeiro criar o agente e seu respectivo source, sink e channel.

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1
```
- ✓ Source: Em nosso exemplo vamos usar o tipo spooldir e indicar qual a pasta em nosso diretório local vamos "ouvir".

```
a1.sources.r1.type = spooldir
a1.sources.r1.spoolDir = precos_us_energia/dados
```

- ✓ Channel: Crie um canal do tipo memory.

```
a1.channels.c1.type = memory  
a1.channels.c1.capacity = 1000
```

- ✓ Sink: Crie um sink que deve armazenar os dados no HDFS apontando para a pasta dados_energia.

```
a1.sinks.k1.type = hdfs  
a1.sinks.k1.hdfs.path = dados_energia
```

- ✓ Por último vamos ligar o source e sink ao channel.

```
a1.sources.r1.channels = c1  
a1.sinks.k1.channel = c1
```

5. Apresente o comando para execução do agente Flume

```
flume-ng agent --conf precos_us_energia /conf --conf-file precos_us_energia /conf/agente2.conf --name  
a1
```

