



INF8200 : Systèmes et infrastructures pour les données massives

HIVER 2024

Nom : NOME GNE MANUELA ESTHER

Code permanent : NOME15269503

1. Tâche no. 1 : Modification de votre cluster

`helm upgrade --set worker.replicaCount=3 spark-release oci://registry-1.docker.io/bitnamicharts/spark`

```
PS D:\tp1> helm upgrade spark-release oci://registry-1.docker.io/bitnamicharts/spark --reuse-values --set worker.replicaCount=3
Pulled: registry-1.docker.io/bitnamicharts/spark:8.7.3
Digest: sha256:de3e91b0649ab97afeb1de18963161fa74701815fd93e84a6e135953021fff4fe
Release "spark-release" has been upgraded. Happy Helming!
NAME: spark-release
LAST DEPLOYED: Sat Mar  2 21:54:57 2024
NAMESPACE: default
STATUS: deployed
REVISION: 3
TEST SUITE: None
NOTES:
CHART NAME: spark
CHART VERSION: 8.7.3
APP VERSION: 3.5.1
```

On affiche les pods lancés, à l'aide de la commande `kubectl get pods` pour vérifier si un autre worker est ajouté.

```
PS D:\tp1> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
spark-release-master-0             1/1     Running   0           22h
spark-release-worker-0             1/1     Running   0           22h
spark-release-worker-1             1/1     Running   0           22h
spark-release-worker-2             1/1     Running   0           56s
```

2. Tâche no. 1.5 (point bonus): réparer pyspark-shell

✓ Question : Quel est le bug ?

```
PS D:\tp1> docker exec -it 81ba952c1e46 bash
I have no name!@spark-release-master-0:/opt/bitnami/spark$ pyspark
Error: pyspark does not support any application options.
```

L'erreur indique que PySpark ne prend pas en charge les options d'application. Pour exécuter PySpark à l'intérieur du conteneur Docker, nous devons d'abord nous assurer que PySpark est correctement configuré dans l'environnement de notre conteneur.

✓ La solution dans le [liens](#)

J'ai importé le fichier en local, modifier la ligne `"${SPARK_HOME}"/bin/spark-submit pyspark-shell-main --name "PySparkShell" "$@"` par `"${SPARK_HOME}"/bin/spark-submit pyspark-shell-main "$@"` pour corriger l'erreur puis copier le répertoire `PySpark` de notre système local vers le conteneur Docker en cours d'exécution

```
D:\tp1>docker cp 81ba952c1e46:/opt/bitnami/spark/bin/pyspark D:\tp1
Successfully copied 4.61kB to D:\tp1

D:\tp1>docker cp D:\tp1\pyspark 81ba952c1e46:/opt/bitnami/spark/bin/
Successfully copied 4.61kB to 81ba952c1e46:/opt/bitnami/spark/bin/
```

- ✓ Lancer notre conteneur et démarrer notre session PySpark, en exécutant les commandes suivantes:

```
PS C:\Users\manue> docker exec -it 81ba952c1e46 bash pyspark
```

```
PS C:\Users\manue> docker exec -it 81ba952c1e46 bash pyspark
Python 3.11.8 (main, Feb 22 2024, 03:37:54) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/03/04 04:40:59 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to

  ____
 /  __ \
/   /  \
/_____/    version 3.5.1
```

3. Tâche no. 2 : Script PySpark

Pour lancer ce script sur kubernetes, j'ai procédé sur PowerShell comme suis :

- Création du dossier TP1 dans mon cluster

```
C:\Users\manue> docker exec 81ba952c1e46 mkdir /opt/bitnami/spark/TP1
```

- Je vérifie qu'il est bien crée en listant les dossiers

```
C:\Users\manue> docker exec 81ba952c1e46 ls /opt/bitnami/spark
```

- Je copie mon fichier script_python dans le dossier

```
C:\Users\manue> docker cp D:\tp1\script_python.py 81ba952c1e46:/opt/bitnami/spark/TP1
```

```
C:\Users\manue> docker cp D:\tp1\script_python.py 81ba952c1e46:/opt/bitnami/spark/TP1
Successfully copied 4.61kB to 81ba952c1e46:/opt/bitnami/spark/TP1
```

- Lancement de spark-submit

```
PS C:\Users\manue> docker exec 81ba952c1e46 /opt/bitnami/spark/bin/spark-submit --master
local/opt/bitnami/spark/TP1/script_python.py 10
```

- a) Ajouter une colonne Totale contenant le total des dépenses et Montrer les 20 premières lignes du Dataframe

```
# Ajout d'une colonne 'Total' contenant la somme des depenses par ligne
df = df.withColumn('Total', sum(df[col] for col in df.columns[1:]))
# Affichage des 20 premieres lignes du DataFrame avec Pandas
df.show(n=20)
```

24/03/04 06:51:04 INFO CodeGenerator: Code generated in 30.011145 ms

| userID | Compute | Storage | Networking | Database | Analytics | Total |
|--------------|---------|---------|------------|----------|-----------|-------|
| BAKA67300004 | 856 | 637 | 301 | 218 | 867 | 2879 |
| BOUT79360000 | 999 | 678 | 787 | 391 | 804 | 3659 |
| CONV09089808 | 537 | 768 | 891 | 128 | 706 | 3030 |
| DIAS03299509 | 343 | 700 | 689 | 750 | 499 | 2981 |
| DICH19079502 | 567 | 349 | 681 | 865 | 812 | 3274 |
| FOFM64270305 | 239 | 439 | 152 | 635 | 534 | 1999 |
| GBEH24279505 | 608 | 422 | 941 | 736 | 230 | 2937 |
| JEAE20118602 | 628 | 411 | 410 | 459 | 577 | 2485 |
| LAFG13039809 | 876 | 432 | 841 | 620 | 148 | 2917 |
| LOXS25369509 | 517 | 300 | 409 | 936 | 137 | 2299 |
| MEDY29339203 | 324 | 209 | 499 | 125 | 348 | 1505 |
| NDIA68270100 | 917 | 506 | 199 | 870 | 547 | 3039 |
| NIAK12339405 | 641 | 997 | 134 | 711 | 807 | 3290 |
| NOME15269503 | 843 | 888 | 236 | 752 | 943 | 3662 |
| SONJ86350009 | 395 | 869 | 988 | 322 | 729 | 3303 |
| SONJ86350009 | 381 | 226 | 852 | 480 | 678 | 2617 |
| SOWM19289605 | 455 | 820 | 891 | 900 | 747 | 3813 |
| TOHD13369601 | 309 | 377 | 954 | 806 | 679 | 3125 |
| SQRL81297538 | 930 | 157 | 337 | 819 | 770 | 3013 |
| DFVN52952345 | 664 | 349 | 429 | 683 | 145 | 2270 |

only showing top 20 rows

b) Créer une table dépenses pour pouvoir utiliser les script SQL de vos collaborateurs

```
# Creation d'une table temporaire pour pouvoir utiliser des requetes SQL
resultat = df.createOrReplaceTempView("depenses")
# Exécution d'une requête SQL pour sélectionner les 20 premières lignes
resultat = spark.sql("SELECT * FROM depenses LIMIT 20")
# Affichage des résultats
resultat.show()
```

```

24/03/04 06:51:05 INFO DAGScheduler: Job 1 finished: showString at Na
+-----+-----+-----+-----+-----+-----+
|   userID|Compute|Storage|Networking|Database|Analytics|Total|
+-----+-----+-----+-----+-----+-----+
|BAKA67300004|    856|    637|    301|    218|    867|  2879|
|BOUT79360000|    999|    678|    787|    391|    804|  3659|
|CONV09089808|    537|    768|    891|    128|    706|  3030|
|DIAS03299509|    343|    700|    689|    750|    499|  2981|
|DICH19079502|    567|    349|    681|    865|    812|  3274|
|FOFM64270305|    239|    439|    152|    635|    534|  1999|
|GBEH24279505|    608|    422|    941|    736|    230|  2937|
|JEAE20118602|    628|    411|    410|    459|    577|  2485|
|LAFG13039809|    876|    432|    841|    620|    148|  2917|
|LOXS25369509|    517|    300|    409|    936|    137|  2299|
|MEDY29339203|    324|    209|    499|    125|    348|  1505|
|NDIA68270100|    917|    506|    199|    870|    547|  3039|
|NIAK12339405|    641|    997|    134|    711|    807|  3290|
|NOME15269503|    843|    888|    236|    752|    943|  3662|
|SONJ86350009|    395|    869|    988|    322|    729|  3303|
|SONJ86350009|    381|    226|    852|    480|    678|  2617|
|SOWM19289605|    455|    820|    891|    900|    747|  3813|
|TOHD13369601|    309|    377|    954|    806|    679|  3125|
|SQRL81297538|    930|    157|    337|    819|    770|  3013|
|DFVN52952345|    664|    349|    429|    683|    145|  2270|
+-----+-----+-----+-----+-----+-----+
24/03/04 06:51:05 INFO CodeGenerator: Code generated in 28.470175 ms

```

c) Envoyez les commandes suivantes de vos collaborateurs créant un rapport sur les données

- Le nombre d'entrée de table résultante de l'ETL

```

# Afficher le nombre d'entrees de votre table resultante de votre ETL
Nombre_entrees = spark.sql("SELECT COUNT(*) FROM depenses")
Nombre_entrees.show()

```

```

+-----+
|count(1)|
+-----+
|      38|
+-----+

```

- La moyenne de la somme totale des dépenses

```

# Afficher la moyenne de la somme totale des dépenses
Moyenne_totale = spark.sql("SELECT AVG(Total) FROM depenses")
Moyenne_totale.show()

```

```
+-----+
|avg(Total)|
+-----+
|    2818.5|
+-----+
```

- Liste des dépenses incluant la somme de l'utilisateur dont le user ID est mon code Permanent

```
#La liste des dépenses (incluant la sommes) de l'utilisateur dont le userID est mon code permanent
user_id = 'NOME15269503'
# Afficher la liste des dépenses (incluant la somme) de l'utilisateur dont le userID est mon code permanent
user_depenses = spark.sql(f"SELECT * FROM depenses WHERE userID = '{user_id}'")
user_depenses.show()
```

```
24/03/04 06:51:08 INFO SparkContext: SparkContext is stopping with exitCode 0.
+-----+-----+-----+-----+-----+-----+
|      userID|Compute|Storage|Networking|Database|Analytics|Total|
+-----+-----+-----+-----+-----+-----+
|NOME15269503|    843|    888|        236|        752|        943| 3662|
+-----+-----+-----+-----+-----+-----+
```

4. Réduire votre cluster à sa taille initiale

[helm upgrade --set worker.replicaCount=2 spark-release oci://registry-1.docker.io/bitnamicharts/spark](#)

```
C:\Users\manue>helm upgrade --set worker.replicaCount=2 spark-release oci://registry-1.docker.io/bitnamicharts/spark
Pulled: registry-1.docker.io/bitnamicharts/spark:8.7.3
Digest: sha256:de3e91b0649ab97afeb1de18963161fa74701815fd93e84a6e135953021ff4fe
Release "spark-release" has been upgraded. Happy Helming!
NAME: spark-release
LAST DEPLOYED: Mon Mar  4 02:49:39 2024
NAMESPACE: default
STATUS: deployed
REVISION: 5
TEST SUITE: None
NOTES:
CHART NAME: spark
CHART VERSION: 8.7.3
APP VERSION: 3.5.1
```

```
PS C:\Users\manue> cd D:\tp1
PS D:\tp1> kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
spark-release-master-0              1/1      Running   0            2d3h
spark-release-worker-0              1/1      Running   0            2d3h
spark-release-worker-1              1/1      Running   0            2d3h
PS D:\tp1>
```