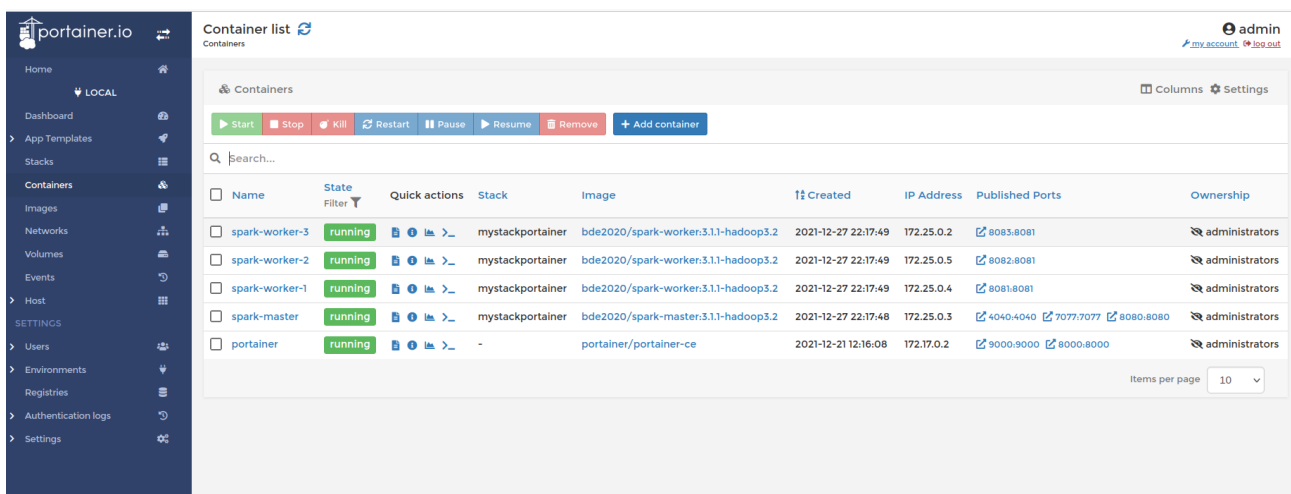
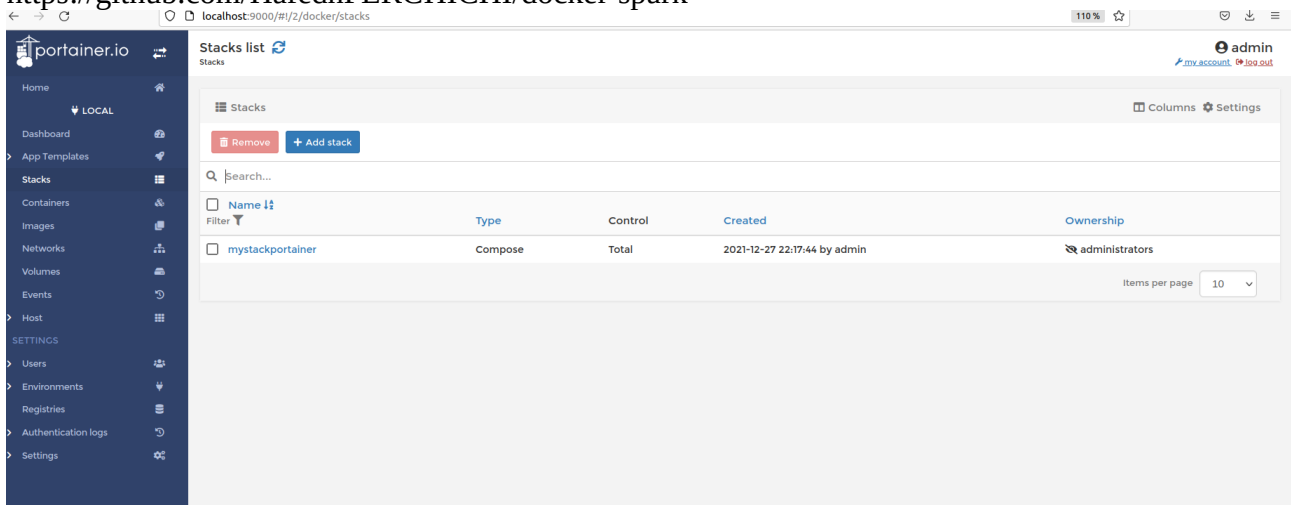


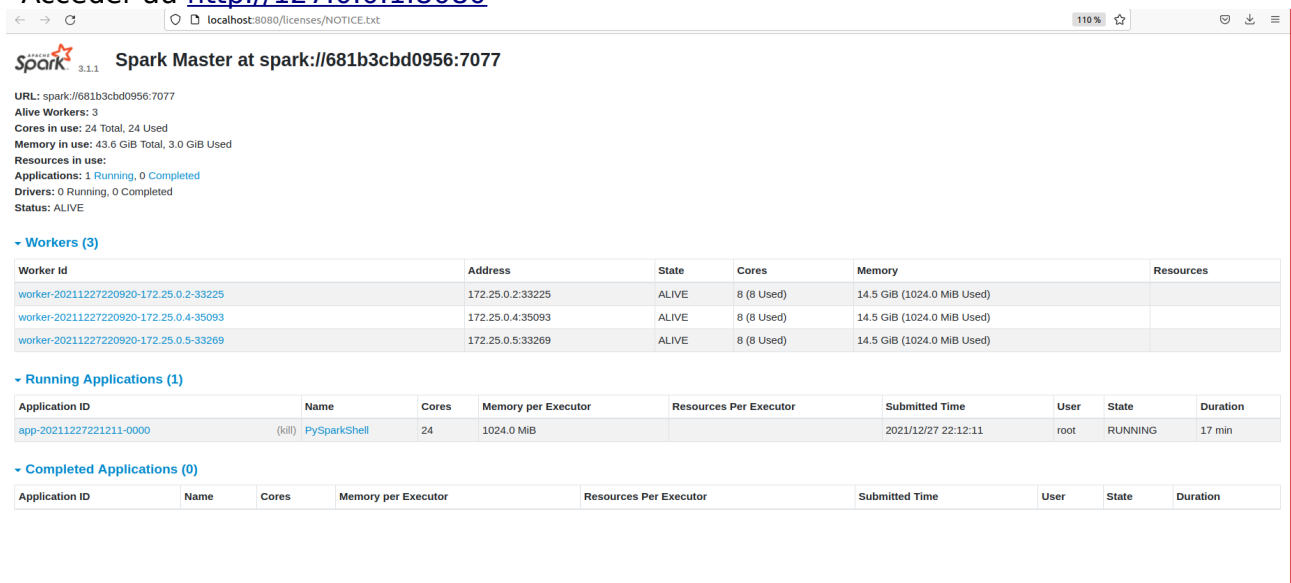
Projet évaluation: WordCount use cluster Spark with docker & portainer

*Accéder à portainer et add stack(deploy stack) via le lien GitHub cidessous :
<https://github.com/HafedhFERCHICHI/docker-spark>



*Visualiser le cluster(un master et trois nœuds esclaves)

*Accéder au <http://127.0.0.1:8080>



Worker id	Address	State	Cores	Memory	Resources
worker-20211227220920-172.25.0.2-33225	172.25.0.2:33225	ALIVE	8 (8 Used)	14.5 GiB (1024.0 MiB Used)	
worker-20211227220920-172.25.0.4-35093	172.25.0.4:35093	ALIVE	8 (8 Used)	14.5 GiB (1024.0 MiB Used)	
worker-20211227220920-172.25.0.5-33269	172.25.0.5:33269	ALIVE	8 (8 Used)	14.5 GiB (1024.0 MiB Used)	

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20211227221211-0000	(kill) PySparkShell	24	1024.0 MiB		2021/12/27 22:12:11	root	RUNNING	17 min

Accéder à l'interface utilisateur Web du workers <http://127.0.0.1:8081>

localhost:8081

110%

3.1.1

Spark Worker at 172.25.0.4:35093

ID: worker-20211227220920-172.25.0.4-35093

Master URL: spark://681b3cbd0956:7077

Cores: 8 (8 Used)

Memory: 14.5 GiB (1024.0 MiB Used)

Resources:

[Back to Master](#)

↳ Running Executors (1)

ExecutorID	State	Cores	Memory	Resources	Job Details	Logs
0	RUNNING	8	1024.0 MiB		ID: app-20211227221211-0000 Name: PySparkShell User: root	stdout stderr

*Créer un fichier inputFile.txt

*Lancer Spark-shell sous le conteneur spark-master

/spark/bin/pyspark --master spark://spark-master:7077: python

Importation du fichier contenant les mots

Decoupage des mots

Indexation des mots en dictionnaire

Reduction du dictionnaire de mots en comptant le nombre de mots

```

>>> text_file = sc.textFile("/data/inputFile.txt")
>>> text_file.collect()
[('On ne change pas', "On met juste les costumes d'autres sur soi", 'On ne change pas', "Une veste ne cache qu'un peu de ce qu'on voit", '')]
>>> WordCount = text_file.flatMap(lambda x : str(x).split(' '))
>>> WordCount = WordCount.map( lambda x : (x,1))
>>> WordCount = WordCount.reduceByKey(lambda x,y: x+y)
>>> WordCount.collect()
[(('change', 2), ('pas', 2), ('juste', 1), ('les', 1), ('d'autres', 1), ('sur', 1), ('soi', 1), ('veste', 1), ('cache', 1), ('peu', 1), ('ce', 1), ('voit', 1), ('', 1), ('Une', 1), ('ne', 3), ('qu'un', 1), ('de', 1), ('qu'on', 1), ('On', 3), ('met', 1), ('costumes', 1))
>>>

```

Sauvegarde du résultat

```

>>> WordCount.collect()
[(('change', 2), ('pas', 2), ('juste', 1), ('les', 1), ('d'autres', 1), ('sur', 1), ('soi', 1), ('veste', 1), ('cache', 1), ('peu', 1), ('ce', 1), ('voit', 1), ('', 1), ('Une', 1), ('ne', 3), ('qu'un', 1), ('de', 1), ('qu'on', 1), ('On', 3), ('met', 1), ('costumes', 1))
>>> WordCount.saveAsTextFile("/data/outputdata")
>>>

```

*Résultat WordCount dans le volume

```

bash-5.0# cd data
bash-5.0# ls
inputFile.txt  outputdata
bash-5.0#

```

