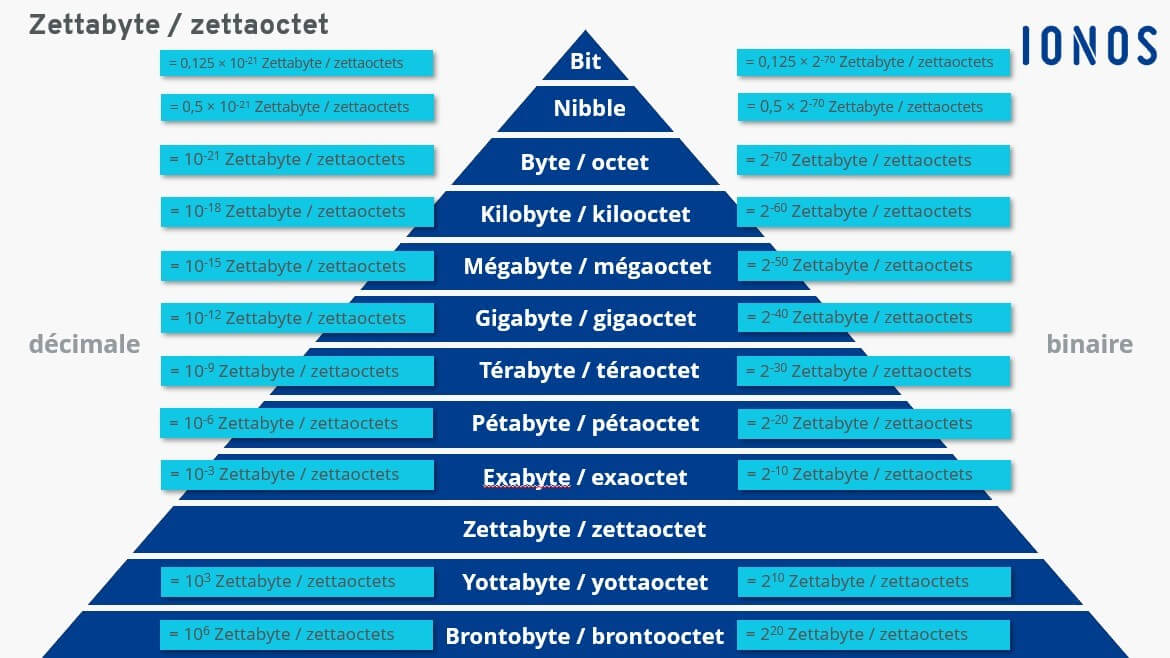
BIG DATA



Introduction :

Pour le traitement de données mise en place du cluster Hadoop avec docker

TP1 :

1. Installer Docker / Docker desktop

2. cloner le projet https://gitlab.com/f.cardenas14/hadoopspark2024

3. débuguer le projet

4. Lancer le cluster et parcourir l’interface web d’hadoop

5. Modifier docker-compose.yml pour avoir 3 datanodes

6. Option : trouver une solution pour exposer la solution puis créer

un datanode sur une machine distante et le relier à l’infrastructure

7. Ajouter des fichiers et les manipuler



TP2 : SPARK

Documentation : <https://spark.apache.org/docs/latest/api/python/index.html>

**Lancer spark sur le cluster** • Option : créer deux scripts ‘’startspark.sh’’ et ‘stopspark.sh’’ pour on/off spark **• ajouter le dossier spark-logs dans « user/root »** docker exec namenode /usr/local/spark/sbin/start-master.sh docker exec datanode1 /usr/local/spark/sbin/start-worker.sh spark://namenode:7077 docker exec datanode2 /usr/local/spark/sbin/start-worker.sh spark://namenode:7077 docker exec namenode /usr/local/spark/sbin/stop-master.sh docker exec datanode1 /usr/local/spark/sbin/stop-worker.sh docker exec datanode2 /usr/local/spark/sbin/stop-worker.sh **• Utiliser la commande « start-all.sh » sur le namenode afin de lancer le cluster complet**

1. Installer python, pip et pyspark sur les conteneurs (ajouter au docker file ?)

2. Créer ou télécharger un fichier texte à utiliser pour le comptage

3. héberger le fichier texte sur le HDFS

4. Lancer le script

Comptage de mots

From pyspark impot sparkContext

# Instansiation d’un SparkContext

Sc = SparkContect()

# Lecture d’un fichier texte : le fichier est décomposé en lignes.

Lines = sc.textFile(«.txt)