***Compte rendu :***

***Bd NoSQL & Big Data***

***TP1 : Initiation à Hadoop***

Réalisé par : Riahi Farah

Classe : ING2 INFO

Objectifs du TP :

L'objectif du TP est de se familiariser avec le cluster Hadoop 3 et d'acquérir des compétences pratiques dans la programmation et la gestion du système de fichiers distribué (HDFS). Le TP vise à approfondir les connaissances sur les concepts clés de HDFS ainsi que sur la gestion efficace des fichiers et dossiers à l'aide des commandes HDFS.

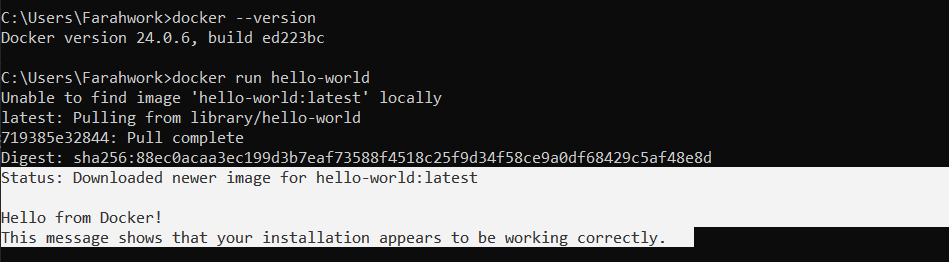
Partie 1 :

* Installation et configuration :
* Installation Docker :

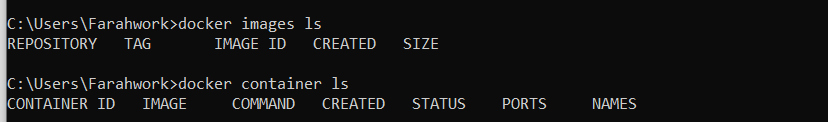


* Vérification de l'installation de Docker & Exécution d'un conteneur de test :

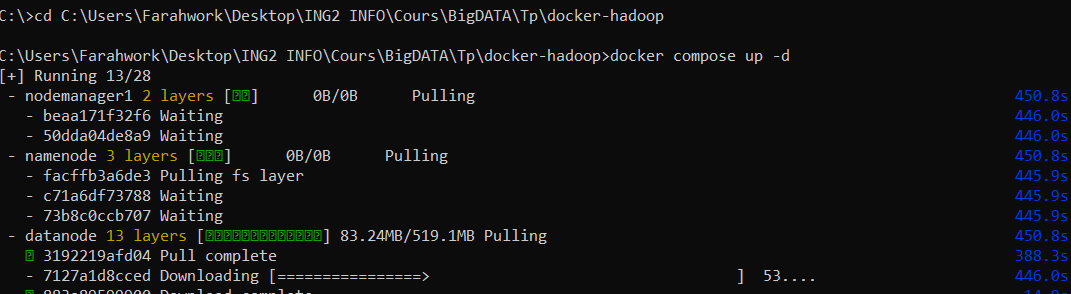
docker run hello-world

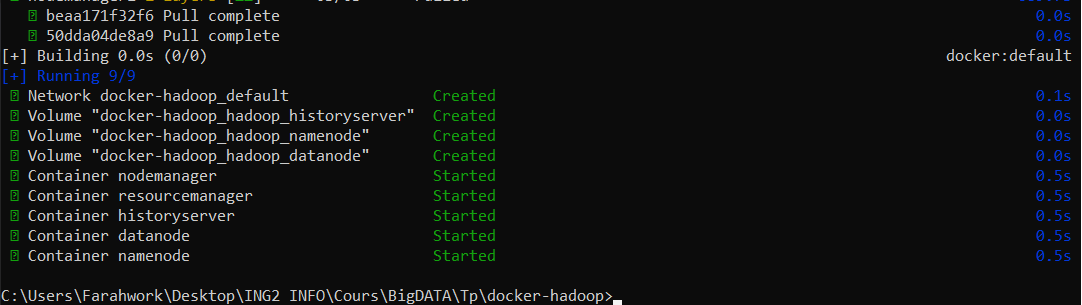


* Lister les images et les containers sur docker



* Cloner l’image du Cluster Hadoop & configuration :
* Hadoop est un écosystème open source conçu pour stocker et traiter de grandes quantités de données sur des clusters de matériel standard. Il est conçu pour être utilisé sur des clusters de serveurs, offrant une grande évolutivité et la possibilité de traiter des quantités massives de données.



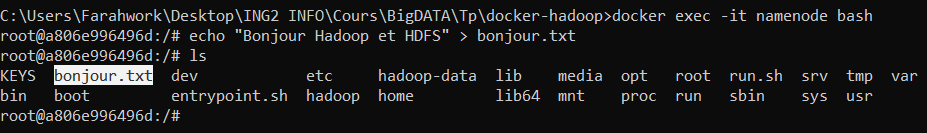


* docker-compose up -d :

1. **docker-compose :** C'est une commande pour exécuter des applications composées de plusieurs conteneurs, définies dans un fichier de configuration appelé "docker-compose.yml".
2. **up :** Ceci est l'une des commandes de base de Docker Compose. Elle construit, (re)crée, démarre et lie les conteneurs pour un service.
3. **-d :** C'est une option de la commande qui signifie "détaché". Elle indique à Docker de lancer les conteneurs en arrière-plan, ce qui signifie qu'ils s'exécuteront en tâche de fond.

En résumé, la commande "docker-compose up -d" est utilisée pour démarrer tous les services spécifiés dans le fichier "docker-compose.yml" et les exécuter en arrière-plan.

Partie 2 : Commandes HDFS

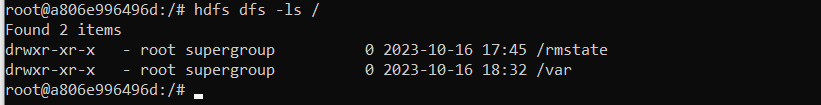


La commande ls :

* Pour le système de fichiers local du conteneur namenode.

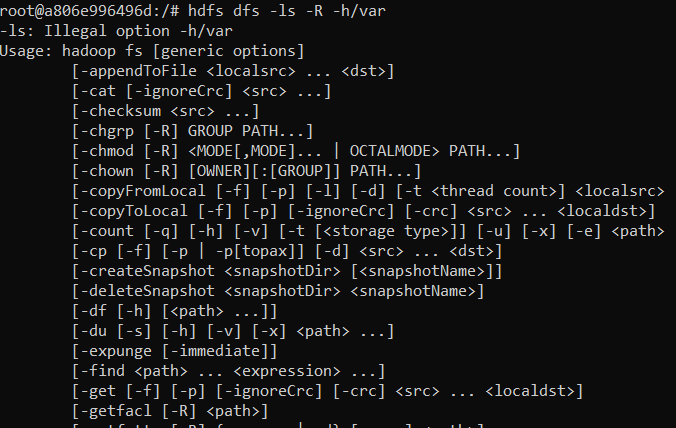
La commande hdfs dfs -ls / :

* Pour le système de fichiers HDFS.



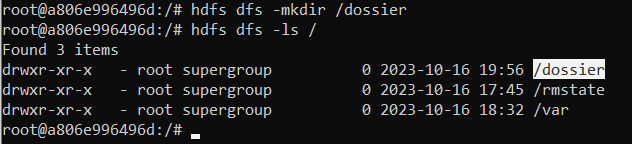
* L'option -ls dans la commande est une commande standard dans Hadoop pour lister le contenu des répertoires. En utilisant / comme argument, on spécifie qu’on souhaite lister le contenu du répertoire racine.
* Cela confirme qu'il y a deux répertoires présents à la racine de votre système de fichiers HDFS :

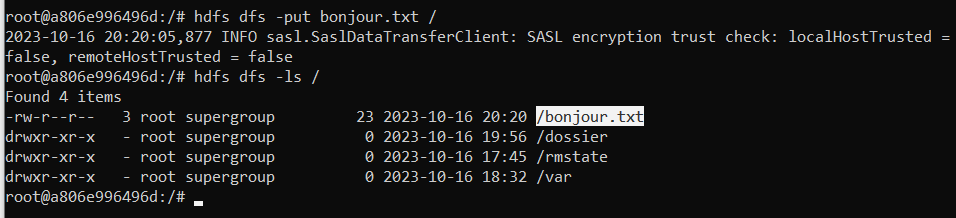
1. /rmstate : Ce répertoire est répertorié avec des droits d'accès et un horodatage de modification. Le répertoire est actuellement vide, comme indiqué par la valeur "0".
2. /var : Ce répertoire est également répertorié avec des droits d'accès et un horodatage de modification. Comme pour le répertoire précédent, il est également actuellement vide.



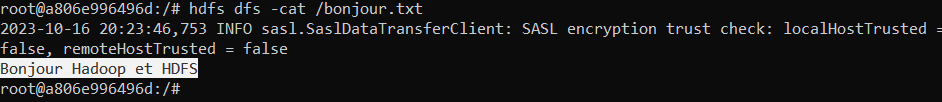


* n'a rien affiché, cela pourrait indiquer qu'il n'y a pas de fichiers ou de sous-dossiers dans le répertoire /var
* Est utilisée pour lister les fichiers et sous-dossiers récursivement dans le répertoire "/var" de HDFS, tout en affichant la taille des fichiers dans des unités lisibles par l'homme (Ko, Mo, Go).
* L'option -ls est utilisée pour lister le contenu du répertoire, -R pour lister récursivement les sous-répertoires, et l'option -h est utilisée pour afficher les tailles de fichiers dans des unités lisibles par l'homme.

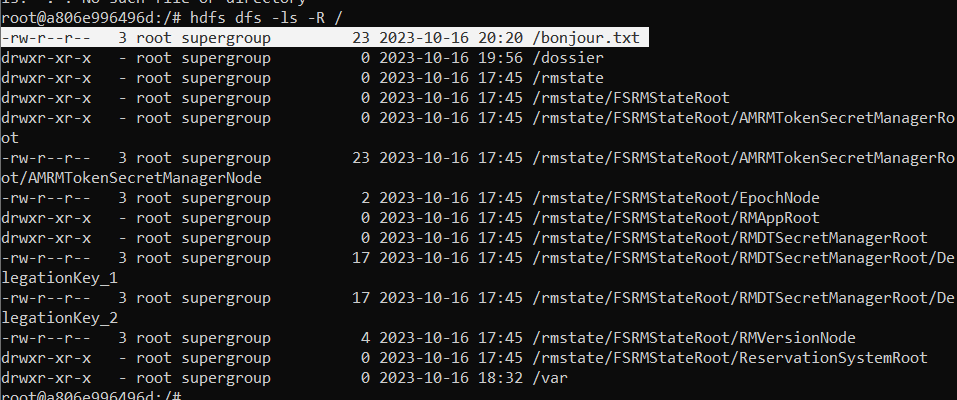




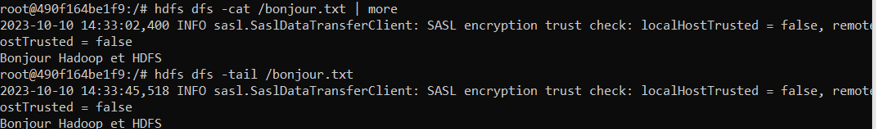
* recharger le fichier "bonjour.txt" dans le système de fichiers HDFS en utilisant la commande hdfs dfs -put bonjour.txt/ . Maintenant ça existe



* voici le contenu du fichier bonjour.txt.



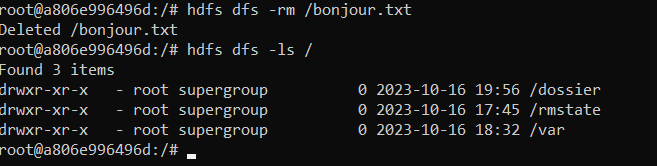
* comme c’est déjà expliqué dans des étapes précédentes, cette commande permet de lister récursivement le contenu de tous les répertoires et sous-répertoires à partir de la racine du système de fichiers HDFS. Cela inclura également les informations sur les fichiers et les répertoires présents dans la racine ainsi que dans les sous-répertoires.

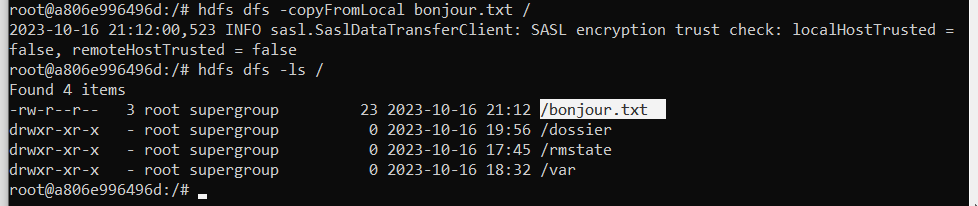


* La commande hdfs dfs -cat bonjour.txt | more est utilisée pour afficher le contenu du fichier "bonjour.txt" dans le terminal, où le résultat est envoyé à la commande more. La commande more permet de visualiser le contenu du fichier page par page dans le terminal, ce qui est utile pour afficher de grands fichiers texte.
* hdfs dfs -tail bonjour.txt est utilisée pour afficher les derniers kilooctets du fichier "bonjour.txt" dans HDFS. Elle est utile lorsque vous voulez inspecter les dernières lignes ou le contenu récent d'un fichier sans avoir à afficher l'intégralité du fichier.

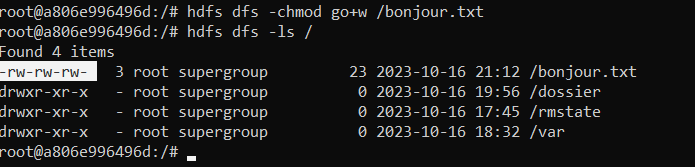


* maintenant on a supprimé bonjour.txt, voila le fichier n’existe pas :

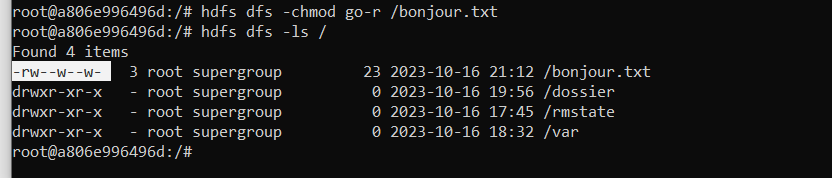




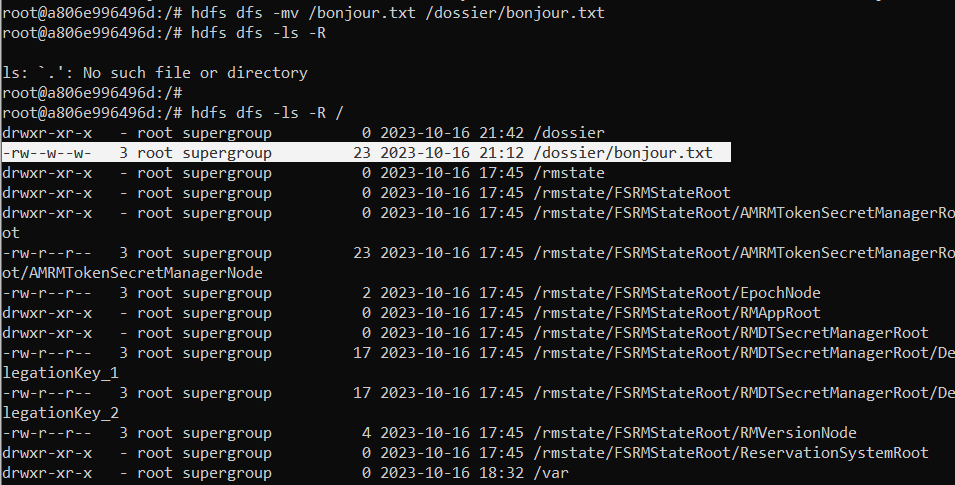
* permet de copier des fichiers depuis le système de fichiers local vers HDFS. On peut remarquer l’existence de bonjour.txt à nouveau.



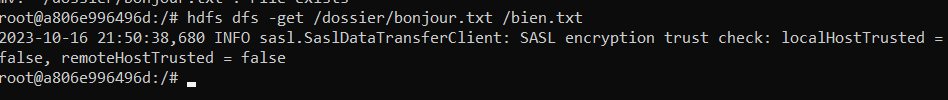
* La commande hdfs dfs -chmod go+w bonjour.txt est utilisée pour modifier les permissions sur le fichier "bonjour.txt" dans HDFS.
* go+w : Cela ajoute des droits d'écriture (+w) pour le groupe (g) et les autres utilisateurs (o). Cela signifie que les membres du groupe et les autres utilisateurs auront désormais la permission d'écrire dans le fichier "bonjour.txt" s'ils en avaient été empêchés auparavant.



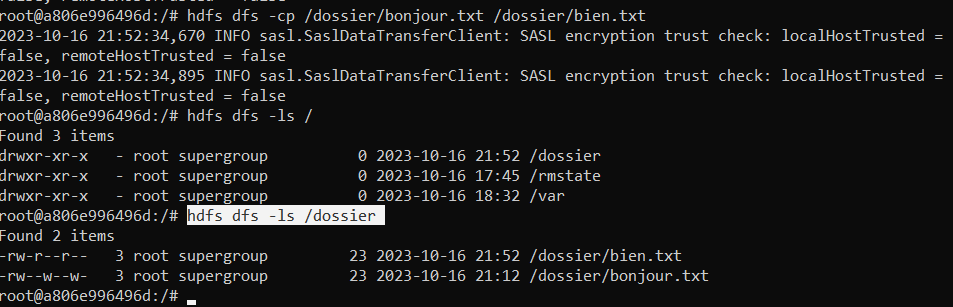
* utilisée pour supprimer les permissions de lecture (r) pour le groupe et les autres utilisateurs sur le fichier "bonjour.txt" dans HDFS. Cela signifie que ni le groupe ni les autres utilisateurs ne pourront lire ce fichier.

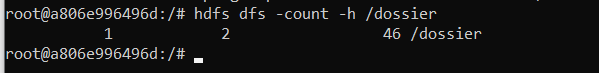


* utilisée pour déplacer le fichier "bonjour.txt" vers le répertoire "dossier" dans HDFS.



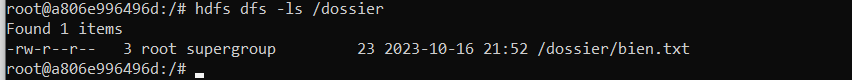
* utilisée pour transférer le fichier "bonjour.txt" du système de fichiers HDFS vers le système de fichiers local tout en le renommant en "bien.txt" dans le système de fichiers local.



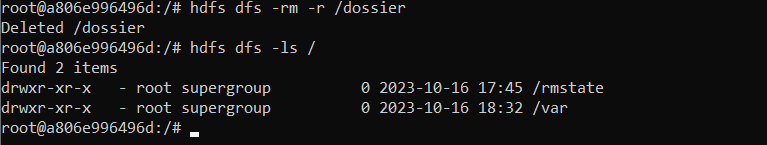


* utilisée pour afficher le nombre de sous-dossiers, de fichiers et la taille totale en octets occupée dans le répertoire spécifié "/dossier" sur HDFS.





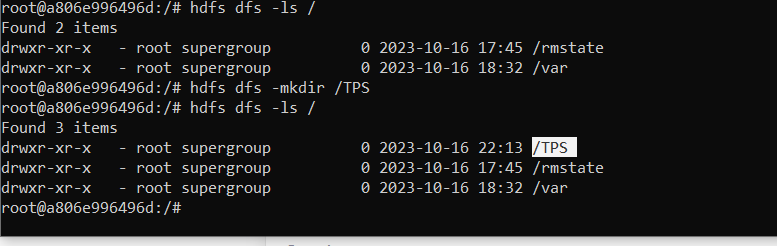
* le dossier a été supprimé avec succès .



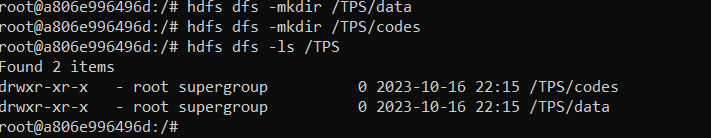
* utilisée pour supprimer récursivement le répertoire "dossier" dans HDFS, y compris tous ses fichiers et sous-répertoires.

Partie 3 : Création d’une arborescence et téléchargement de fichier

1-

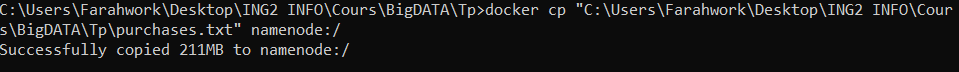


* tout d’abord la création du dossier parent TPS

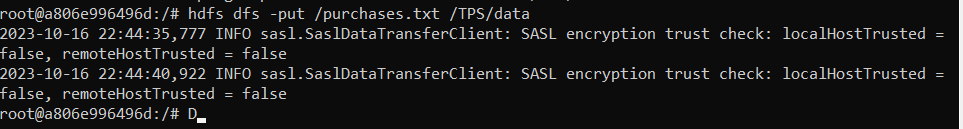


* Ensuite, on crée les sous-répertoires "data" et "codes" à l'intérieur du répertoire parent "TPS" .

2-



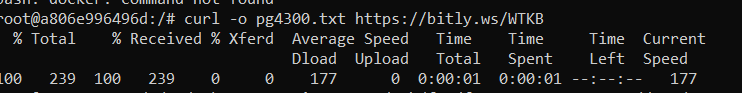
* Cette commande a réussi à copier le fichier "purchases.txt" depuis notre machine hôte vers le container namenode.



* déplacer le fichier dans le dossier "data" sur HDFS.

3-

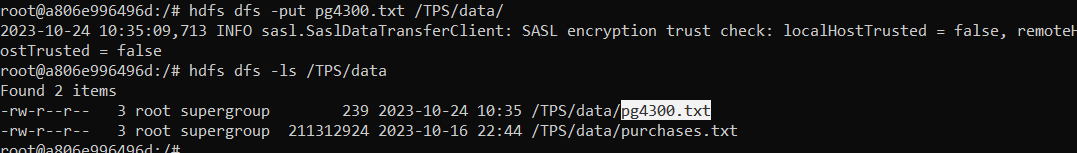
Pour télécharger le fichier "pg4300.txt" directement sur le noeud namenode à partir de notre machine hôte, on peut utiliser la commande wget depuis le container namenode en utilisant docker exec :



Pour vérifier si le fichier a déjà été téléchargé dans le conteneur avant de le copier vers HDFS, vous pouvez exécuter les commandes suivantes pour vérifier la présence du fichier :

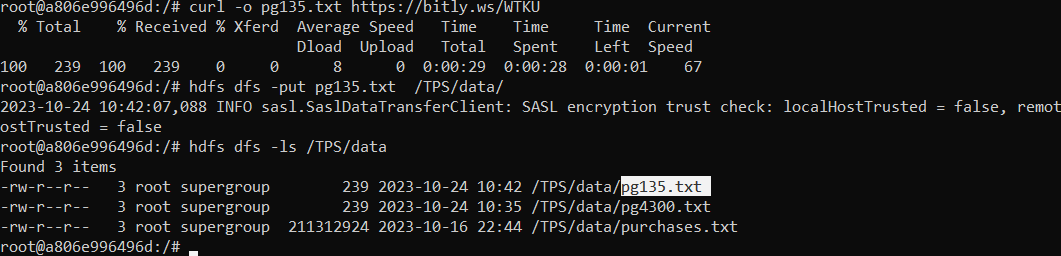


Deplacer sur data :



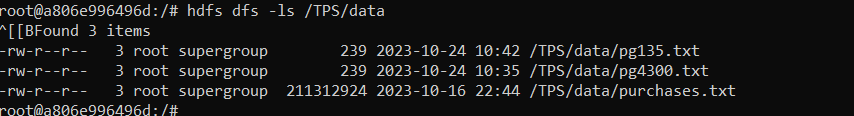
4-

Pour télécharger le fichier pg135.txt depuis l'URL donnée et le placer dans le dossier HDFS :

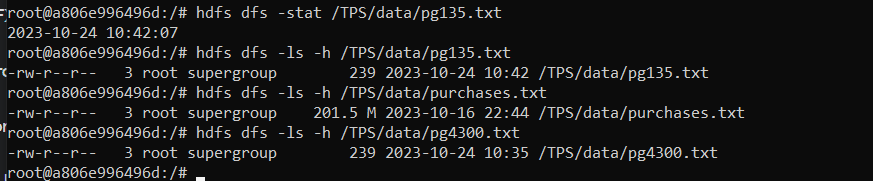


5-

Lister le contenu du répertoire "data" sur HDFS :

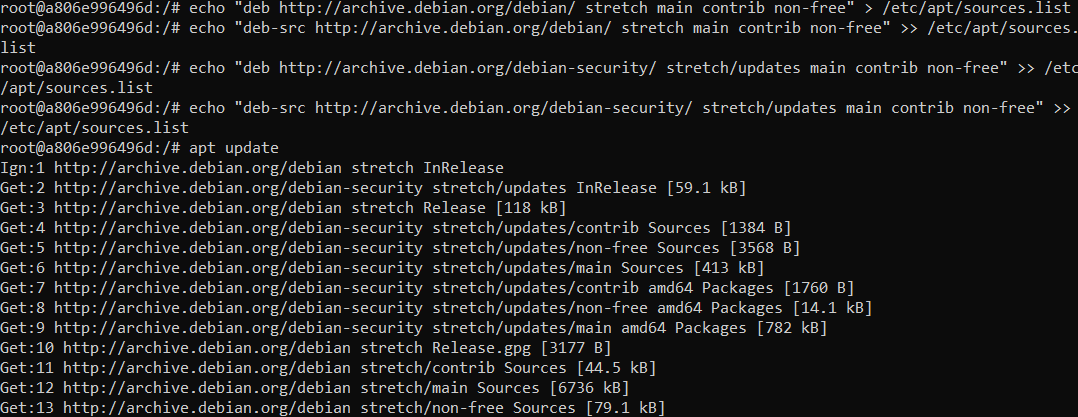


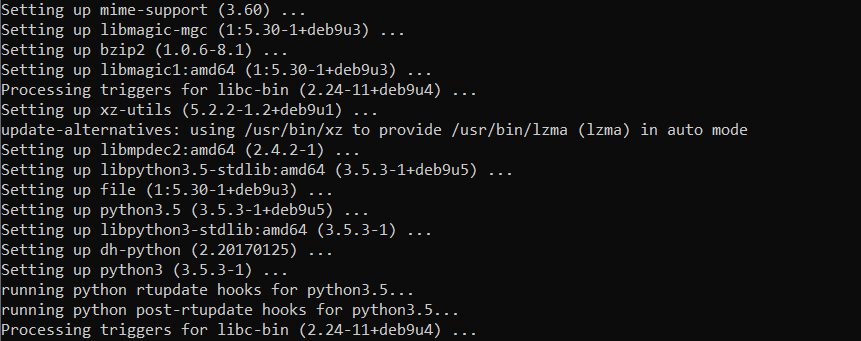
Afficher des informations sur les fichiers téléchargés, comme leur taille, leur propriétaire et les autorisations :



Partie 4 :

Etapes d’installation de python sur les centenaires :







* Python bien installé .