Chapitre 3
Installer Hadoop en local

1. Introduction

L'installation de Hadoop constitue une étape fondamentale pour toute organisation ou tout étudiant qui souhaite comprendre concrètement le fonctionnement de ce Framework. Bien au-delà d'un simple exercice technique, cette étape permet de plonger au cœur de l'architecture distribuée et de découvrir comment les différents composants de Hadoop collaborent pour traiter de grands volumes de données.

Dans sa forme la plus simple, Hadoop peut être installé sur une seule machine en mode pseudo-distribué. Ce type d'installation, souvent utilisé à des fins pédagogiques ou de test, simule le comportement d'un cluster mais reste limité en termes de performances et de parallélisme. En revanche, le mode fully-distribué correspond à une configuration réelle en cluster, dans laquelle plusieurs nœuds physiques ou virtuels coopèrent. C'est ce mode qui reflète véritablement la puissance et la philosophie du Big Data, en exploitant simultanément la mémoire, le CPU et le stockage de multiples serveurs.

L'objectif de ce chapitre est de guider pas à pas le lecteur à travers ces étapes d'installation. Nous commencerons par rappeler les prérequis matériels et logiciels nécessaires pour mettre en place un environnement fonctionnel. Ensuite, nous détaillerons l'installation en mode distribué, en présentant les concepts clés du cluster, la préparation des binaires, la configuration des fichiers XML, ainsi que le démarrage des services HDFS et YARN. Une fois le cluster opérationnel, nous validerons l'installation à travers un premier job MapReduce simple, avant d'aborder les possibilités d'automatisation avec des scripts ou des outils de provisioning modernes comme Ansible.

À travers cette démarche progressive, le lecteur sera en mesure de comprendre non seulement comment installer Hadoop, mais surtout pourquoi chaque étape est nécessaire. Cette approche lui donnera les bases pour administrer un cluster plus vaste, résoudre des problèmes courants et préparer le terrain pour des usages avancés tels que Spark, Hive ou encore le machine learning distribué.

2. Prérequis techniques

Avant de se lancer dans l'installation d'Hadoop en local, il est indispensable de poser les bases d'un environnement solide. Hadoop n'est pas un logiciel ordinaire que l'on télécharge puis installe par défaut : c'est une plateforme distribuée qui simule à elle seule l'architecture d'un data center miniature. Même dans une configuration locale, l'utilisateur doit s'assurer que son système remplit un ensemble de conditions techniques, aussi bien matérielles que logicielles. Cette étape préliminaire peut sembler fastidieuse, mais elle conditionne le bon fonctionnement du cluster et évite de nombreux blocages au moment de l'exécution.

2.1 Configuration matérielle minimale et recommandée

Hadoop a été imaginé à l'origine pour tourner sur des grappes entières de serveurs connectés en réseau, chacun jouant un rôle bien précis dans le stockage ou le traitement des données. Pourtant, pour apprendre et expérimenter, il est tout à fait possible de l'installer sur une seule machine, qui va alors jouer symboliquement le rôle de l'ensemble d'un cluster. Cette flexibilité rend Hadoop accessible aux étudiants et aux ingénieurs qui souhaitent comprendre son fonctionnement sans disposer immédiatement d'une infrastructure lourde. Mais cette simplicité apparente ne doit pas masquer la réalité : même en local, Hadoop reste exigeant en termes de ressources, et négliger ce point mène rapidement à des ralentissements ou à des blocages.

Le processeur constitue le premier élément critique. Avec un simple CPU double cœur, on peut effectivement installer Hadoop et exécuter des jobs modestes, mais les limites se font sentir dès que plusieurs services tournent en parallèle. Le NameNode, le DataNode, le ResourceManager ou encore le NodeManager sont autant de processus distincts qui sollicitent le processeur. Si celui-ci est trop limité, la machine entière devient vite saturée, ce qui ralentit non seulement Hadoop mais aussi le système d'exploitation lui-même. En revanche, avec quatre cœurs, l'expérience devient nettement plus confortable, et un processeur moderne à six ou huit cœurs permet d'aborder des scénarios plus réalistes, proches d'un véritable cluster distribué.

La mémoire vive joue un rôle tout aussi crucial. Hadoop repose sur de nombreuses structures internes, notamment pour gérer les métadonnées ou pour faire tourner les conteneurs YARN. Avec seulement 4 Go de RAM, l'installation devient presque impraticable, car le système d'exploitation et la machine virtuelle Java consomment déjà l'essentiel de la mémoire disponible. En disposant de 8 Go, il est possible de faire tourner Hadoop et d'exécuter des traitements de base, mais la marge reste faible et les autres applications tournant en parallèle risquent de ralentir fortement. Pour un usage fluide, il est conseillé de travailler avec au moins 16 Go, ce qui permet de répartir convenablement la mémoire entre les services sans mettre la machine sous tension. Ceux qui souhaitent travailler sur des jeux de données plus volumineux ou lancer plusieurs jobs simultanément trouveront un vrai confort à disposer de 32 Go ou davantage.

L'espace disque constitue un autre facteur déterminant. HDFS, le système de fichiers distribué d'Hadoop, repose sur la réplication des données, avec trois copies par défaut. Ainsi, un simple fichier de 1 Go en occupera en réalité 3 sur le disque. À cela s'ajoutent les journaux, les fichiers temporaires et les caches intermédiaires nécessaires aux traitements. Avec 100 Go libres, on peut réaliser des expérimentations de base, mais dès que les volumes augmentent, cette capacité devient insuffisante. Il est préférable de disposer d'au moins 500 Go pour travailler sereinement, et l'usage d'un SSD, bien que non obligatoire, apporte un gain considérable. Les phases intensives comme le tri ou le shuffle, qui manipulent énormément de données intermédiaires, bénéficient fortement de la rapidité d'accès qu'offre ce type de disque.

Enfin, même si l'installation en local ne requiert pas un réseau externe, Hadoop repose entièrement sur la communication via TCP/IP. Tous les services, y compris sur une seule machine, échangent par la pile réseau. Une configuration réseau stable et une carte correctement paramétrée sont donc essentielles, et dès que l'on passe à une configuration multi-nœuds, un réseau Gigabit devient absolument indispensable pour éviter les congestions et garantir de bonnes performances.

Pour résumer ces recommandations, le tableau ci-dessous présente une comparaison entre une configuration minimale et une configuration conseillée pour une installation locale d'Hadoop :

Critère	Configuration minimale	Configuration recommandée
Système d'exploitation	Linux (Ubuntu/Debian/ CentOS)	Linux 64-bit (Ubuntu LTS ou CentOS)
Processeur (CPU)	2 cœurs	4 cœurs ou plus
Mémoire (RAM)	4 Go	8 Go ou plus (16 Go idéalement)
Disque dur (stockage)	20 Go	100 Go ou plus, idéalement avec SSD
Java	JDK 8 ou JDK 11	JDK 11 (version stable)
Python	Optionnel (≥ 3.6)	Python 3.8 ou plus

Critère	Configuration minimale	Configuration recommandée
Réseau	SSH activé, ports Hadoop libres	SSH sans mot de passe, noms d'hôte fixés

2.2 Système d'exploitation et compatibilité

Hadoop est historiquement né dans l'écosystème Linux et reste aujourd'hui encore intimement lié à ce système. La quasi-totalité des clusters déployés en production tournent sur des distributions Linux reconnues pour leur stabilité, comme CentOS, Red Hat Enterprise Linux ou encore Ubuntu Server. Pour un apprentissage ou une installation locale, Ubuntu s'impose souvent comme le choix naturel : sa communauté est particulièrement active et de nombreux tutoriels y font référence. Debian et Rocky Linux représentent également d'excellentes alternatives, chacune offrant une robustesse et un cycle de mises à jour adapté aux environnements professionnels.

Bien qu'Hadoop puisse fonctionner sur d'autres systèmes, des limitations apparaissent rapidement. Sous macOS, il est techniquement possible d'installer Hadoop, mais certains scripts et utilitaires ne sont pas toujours compatibles. L'installation demande alors des ajustements, ce qui peut compliquer inutilement l'expérience pour un débutant. Quant à Windows, la situation est encore plus contraignante : une installation directe y est rarement utilisée, même pour des tests, car l'écosystème Hadoop n'a jamais été pensé pour cet environnement. Toutefois, plusieurs solutions permettent de contourner cette difficulté. La plus répandue est l'utilisation de WSL (*Windows Subsystem for Linux*), qui installe une distribution Linux (souvent Ubuntu) à l'intérieur de Windows, offrant ainsi un environnement parfaitement compatible avec Hadoop. Une autre approche consiste à recourir à Docker, qui encapsule Hadoop dans des conteneurs préconfigurés. Cette méthode simplifie considérablement la mise en place et permet de disposer d'un cluster prêt à l'emploi, tout en restant fidèle à la logique d'architecture distribuée.

Le choix du système d'exploitation ne doit pas être pris à la légère. En phase d'apprentissage, l'important est de disposer d'un environnement stable et documenté, afin de se concentrer sur la compréhension des concepts Hadoop sans être ralenti par des problèmes techniques liés à l'OS. En production, le critère principal devient la fiabilité et la compatibilité avec les autres briques logicielles de l'écosystème Big Data. Dans tous les cas, il est recommandé de choisir une distribution éprouvée et de s'y tenir : changer de système en cours de route complique inutilement l'installation et rend la maintenance plus difficile.

2.3 Préparation logicielle

Si l'infrastructure matérielle constitue la base indispensable à toute installation Hadoop, la préparation logicielle est tout aussi cruciale. Hadoop n'est pas une simple application « clé en main » : il repose sur une combinaison d'outils et de dépendances logicielles, dont certains doivent être configurés avec soin avant même de lancer la première commande.

2.3.1 Java, pilier incontournable

Hadoop est écrit en Java, et à ce titre, il nécessite la présence d'une machine virtuelle Java (JVM). Sans elle, aucun des services du cluster NameNode, DataNode, ResourceManager, NodeManager ne peut démarrer. Les versions les plus couramment utilisées sont Java 8 et Java 11, qui offrent à la fois stabilité et compatibilité avec la majorité des distributions Hadoop disponibles.

Il est fortement recommandé d'installer la version la plus récente supportée par la distribution Hadoop choisie, puis de configurer correctement la variable d'environnement JAVA_HOME. Cette variable indique à Hadoop l'emplacement de l'installation de Java. Une mauvaise configuration conduit systématiquement à des erreurs lors du démarrage des services, erreurs parfois difficiles à diagnostiquer pour un utilisateur débutant.

2.3.2 SSH, un prérequis pour la communication interne

Un aspect souvent méconnu est que, dans certaines configurations, Hadoop utilise le protocole SSH pour lancer des services sur différents nœuds. Toutefois, en mode pseudo-distribué avec Hadoop 3.x, SSH n'est pas indispensable et devient surtout nécessaire dans les déploiements multi-nœuds. Les différents processus, simulant un cluster distribué, utilisent SSH pour exécuter des commandes à distance, même si elles restent confinées à la même machine.

Il est donc indispensable de configurer un accès SSH sans mot de passe vers localhost. Cela implique la génération de clés publiques et privées (via `ssh-keygen`) puis leur enregistrement dans le fichier `~/.ssh/authorized_keys`. Une fois cette étape réalisée, la commande :

```
$ ssh localhost
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Last login: Tue Oct 1 14:18:55 2025 from 127.0.0.1
user@hadoop-master:~$
```

ne doit plus demander de mot de passe. Ce détail, qui peut sembler anodin, est pourtant la clé du bon fonctionnement des scripts automatiques de Hadoop.

2.3.3 Outils utilitaires indispensables

Au-delà de Java et de la configuration d'un accès SSH fonctionnel, un certain nombre d'outils Unix/Linux sont incontournables pour préparer et administrer un environnement Hadoop. Ces utilitaires, souvent considérés comme secondaires, jouent pourtant un rôle déterminant dans le bon déroulement de l'installation et dans la maintenance quotidienne du cluster.

Les commandes `wget` ou `curl` sont par exemple nécessaires pour télécharger les binaires officiels depuis le site d'Apache ou depuis des dépôts miroirs. Elles permettent également de récupérer rapidement des scripts ou des dépendances complémentaires sans avoir à passer par une interface graphique. Une fois les fichiers obtenus, l'étape suivante consiste à les décompresser : pour cela, tar et gzip constituent les outils classiques. Hadoop étant distribué sous forme d'archives compressées, leur maîtrise est indispensable dès la première manipulation.

D'autres commandes, plus orientées diagnostic, facilitent la gestion du cluster au quotidien. L'instruction `ps` permet de vérifier que les processus Hadoop (NameNode, DataNode, ResourceManager, etc.) sont correctement lancés, tandis que netstat ou ss offrent une visibilité sur les ports ouverts et les connexions réseau, points cruciaux pour détecter des conflits ou des blocages. Ces utilitaires, bien que souvent installés par défaut sur les distributions Linux modernes, ne doivent pas être considérés comme acquis. Une vérification préalable évite bien des surprises lors du démarrage des services Hadoop.

En résumé, ces outils constituent la « boîte à outils de base » de l'administrateur Hadoop. Ils ne se limitent pas à l'installation : ils deviennent rapidement indispensables pour diagnostiquer un problème, contrôler l'activité du cluster ou automatiser certaines opérations. Leur présence et leur bon fonctionnement sont donc à valider avant toute tentative de mise en place d'un environnement Hadoop, même en mode local.

2.3.4 Python et autres langages de script

Même si Hadoop peut fonctionner sans la présence de Python, ce langage occupe une place particulière dans l'écosystème Big Data. En pratique, une grande partie des tutoriels, des exercices pédagogiques et des scripts utilitaires s'appuie sur lui. Dès l'installation, il n'est pas rare que l'utilisateur souhaite exécuter le programme WordCount, exemple emblématique qui illustre le fonctionnement d'un job MapReduce. Ce programme existe bien sûr en Java, mais sa déclinaison en Python permet de tester le système plus rapidement, sans avoir à écrire ni compiler un code Java complet.

Disposer d'une version récente de Python 3 est donc vivement recommandé. La compatibilité minimale se situe autour de la version 3.6, mais pour bénéficier des optimisations et du support des bibliothèques récentes, il est préférable d'installer Python 3.8 ou une version ultérieure. Cela permet non seulement de lancer les exemples fournis avec Hadoop, mais aussi d'assurer une cohérence avec les Frameworks modernes de l'écosystème, tels que PySpark, qui constitue aujourd'hui l'une des interfaces les plus utilisées pour manipuler les données à grande échelle.

En allant plus loin, l'intégration de Python ouvre la voie à une utilisation hybride d'Hadoop : au-delà de MapReduce, il devient possible de développer des pipelines de traitement plus souples, d'interagir avec des bibliothèques de data science et même de connecter Hadoop avec des environnements analytiques comme Jupyter. Ainsi, Python ne se contente pas d'être un outil pédagogique : il s'impose comme un compagnon naturel pour tout administrateur ou analyste souhaitant exploiter Hadoop dans des scénarios modernes et variés.

2.4 Réseau et configuration de base

Hadoop est une plateforme distribuée qui repose entièrement sur la communication réseau, même lorsqu'il est installé sur une seule machine. En effet, chacun de ses services échange des informations via la pile TCP/IP, comme s'ils étaient déployés sur plusieurs serveurs distincts. Comprendre la configuration réseau de base est donc essentiel pour assurer un fonctionnement correct du cluster et éviter les erreurs au démarrage.