TP 6

Administration système

Dardor Rochdi

Stratégie de sauvegarde

EXERCICE 6.1 :

La sauvegarde incrémentale est une méthode de sauvegarde qui consiste à enregistrer uniquement les fichiers qui ont été modifiés ou créés depuis la dernière sauvegarde, qu'elle soit complète ou incrémentale. Cette approche permet une optimisation significative des ressources en termes d'espace de stockage et de temps nécessaire pour effectuer les sauvegardes, notamment dans les environnements où les données évoluent fréquemment.

Lors de la première opération, une sauvegarde complète est réalisée afin de disposer d'une base initiale contenant l'intégralité des données. Par la suite, chaque sauvegarde incrémentale ne prend en compte que les changements apportés depuis la sauvegarde précédente, qu'il s'agisse de modifications, de suppressions ou d'ajouts. Par exemple, une sauvegarde complète effectuée le lundi servira de référence, tandis que les sauvegardes incrémentales des jours suivants enregistreront uniquement les modifications survenues depuis la veille.

Cependant, bien que cette méthode soit avantageuse pour réduire les temps de traitement et la consommation d'espace disque, elle présente également des inconvénients. La restauration des données nécessite la disponibilité de la dernière sauvegarde complète ainsi que de toutes les sauvegardes incrémentales jusqu'à la date souhaitée, ce qui peut rendre le processus plus long et plus complexe. De plus, si une sauvegarde incrémentale est corrompue ou perdue, cela peut compromettre l'intégrité de l'ensemble des données à restaurer.

En pratique, des outils tels que **rsync**, **tar**, ou des solutions avancées comme Bacula et Veeam, sont souvent utilisés pour automatiser et optimiser les sauvegardes incrémentales, garantissant ainsi une gestion efficace des données dans un cadre sécurisé.

EXERCICE 6.2 :

Planning de sauvegarde :

1. **Sauvegarde complète** :
    - Fréquence : Une fois par semaine (par exemple, dimanche à 2h du matin).
    - Justification : Les sauvegardes complètes nécessitent plus de temps et d'espace, mais elles servent de base pour les sauvegardes incrémentales.

Dimanche est idéal, car c'est souvent une période de faible activité pour les utilisateurs.

2. **Sauvegardes incrémentales** :
    - Fréquence : Quotidienne (du lundi au samedi à 2h du matin).
    - Justification : Les sauvegardes incrémentales sont rapides et consomment peu d'espace, ce qui permet de capturer efficacement les modifications journalières.
3. **Sauvegarde des bases critiques (MySQL, LDAP)** :
    - Fréquence : Deux fois par jour (par exemple, à 12h et 23h).
    - Justification : Les bases de données subissent souvent des modifications fréquentes, justifiant une sauvegarde plus régulière pour éviter des pertes importantes.

Justification des horaires :

- **Heure de nuit (2h du matin)** : La majorité des utilisateurs ne sont pas actifs à ces heures, ce qui réduit l'impact des sauvegardes sur les performances du serveur.
- **Milieu de journée (12h)** : Permet de capturer les données critiques après une matinée de travail.
- **Fin de soirée (23h)** : Garantit une sauvegarde avant les activités nocturnes ou les éventuels traitements automatiques.

EXERCICE 6.3 :

• /home (Répertoires des utilisateurs) :

Volume : Ce répertoire contient les fichiers personnels des utilisateurs ayant accès au serveur. En supposant qu'un utilisateur utilise en moyenne 10 Go d'espace de stockage, le volume total sera d'environ $n \times 10\ Go$, où n représente le nombre d'utilisateurs.

Méthode de sauvegarde : Des sauvegardes incrémentales quotidiennes (par exemple, à 2h du matin) pour capturer les modifications fréquentes, complétées par une sauvegarde complète hebdomadaire le dimanche à 2h du matin.

• Base LDAP :

Volume : Cette base est modeste, avec une taille estimée à environ 1 Go. Elle contient des informations sur les identités et les accès des utilisateurs.

Méthode de sauvegarde : Une sauvegarde complète mensuelle (par exemple, le premier dimanche du mois à 2h30 du matin), avec des sauvegardes incrémentales toutes les

semaines, le dimanche à 2h45 du matin, pour prendre en compte les modifications occasionnelles.

• **Base de données MySQL :**

Volume : La base de données MySQL contient les données principales du site WordPress et d'autres applications, avec une taille estimée à 10 Go, susceptible d'évoluer avec le temps.

Méthode de sauvegarde : Une sauvegarde complète hebdomadaire (dimanche à 3h00 du matin) pour garantir une base fiable, accompagnée de sauvegardes incrémentales quotidiennes à 1h00 du matin pour capturer les modifications journalières. L'outil `mysqldump` sera utilisé pour ces sauvegardes.

• **Service web WordPress :**

Volume : Les fichiers du service web (fichiers de configuration, logs, thèmes, plugins) représentent environ 1 Go de données.

Méthode de sauvegarde : Utilisation de l'outil `tar` pour archiver et sauvegarder les fichiers de manière organisée, en suivant la même fréquence que les sauvegardes des bases MySQL.

EXERCICE 6.4 :

1. Disque dur local (HDD ou SSD)

- **Avantages** :
    - Accès rapide aux données.
    - Facilité d'utilisation et de mise en œuvre.
    - Coût abordable pour des volumes de stockage importants.
- **Inconvénients** :
    - Vulnérable aux pannes matérielles.
    - Exposé aux risques locaux (incendie, vol, etc.).
    - Limité à un seul emplacement physique.

2. Disque dur externe

- **Avantages** :
    - Mobile, permet de déplacer les sauvegardes hors site.
    - Facilité de connexion et de configuration.
- **Inconvénients** :

    - Risques de perte ou d'endommagement physique.
    - Peut-être moins performant que le stockage interne.

3. Support optique (CD/DVD/Blu-ray)

- **Avantages** :
    - Coût faible par unité.
    - Résistant aux pannes électriques.
- **Inconvénients** :
    - Faible capacité de stockage (surtout pour CD/DVD).
    - Temps d'écriture plus long.
    - Susceptible aux rayures et à la détérioration avec le temps.

4. Serveur NAS (Network Attached Storage)

- **Avantages** :
    - Accès partagé et centralisé sur un réseau.
    - Options de redondance comme RAID pour protéger les données.
- **Inconvénients** :
    - Plus coûteux qu'un disque dur simple.
    - Dépendance à la disponibilité du réseau.

5. Cloud (Stockage en ligne)

- **Avantages** :
    - Accessible depuis n'importe où.
    - Résilient aux pannes locales grâce à la redondance des centres de données.
    - Évolutivité en fonction des besoins.
- **Inconvénients** :
    - Dépendance à une connexion Internet fiable.
    - Coût récurrent en fonction du volume de données.
    - Préoccupations de confidentialité et de sécurité.

6. Bande magnétique

- **Avantages** :
    - Très grande capacité de stockage.
    - Durable et économique pour des sauvegardes à long terme.
- **Inconvénients** :
    - Temps d'accès plus long (lecture séquentielle).
    - Nécessite du matériel spécifique souvent coûteux.

EXERCICE 6.5 :

La gestion du stockage des supports de sauvegarde est essentielle pour garantir l'intégrité des données et assurer une restauration efficace en cas de besoin. Voici les aspects à considérer :

Environnement physique :

Température et humidité :
Il est crucial d'éviter les conditions extrêmes qui pourraient détériorer les supports de sauvegarde. Un environnement frais et sec est recommandé pour préserver leur longévité.

Protection contre les chocs :
Les supports doivent être protégés des vibrations et des impacts physiques. Les disques durs et autres supports sensibles doivent être entreposés dans des boîtiers robustes pour limiter les risques de dommages.

Sécurité :
Les sauvegardes doivent être protégées contre le vol et les accès non autorisés. Cela implique un stockage dans des lieux sécurisés, idéalement distants du site principal des données. Pour renforcer la sécurité, les données sensibles devraient être chiffrées afin d'empêcher toute exploitation en cas de perte ou de vol.

Compatibilité matérielle :
Il est important de conserver les équipements nécessaires à la lecture et à l'écriture des supports. Certains formats de stockage peuvent devenir obsolètes au fil du temps, nécessitant un matériel spécifique qui pourrait ne plus être disponible. Une vérification régulière de la compatibilité matérielle est donc essentielle.

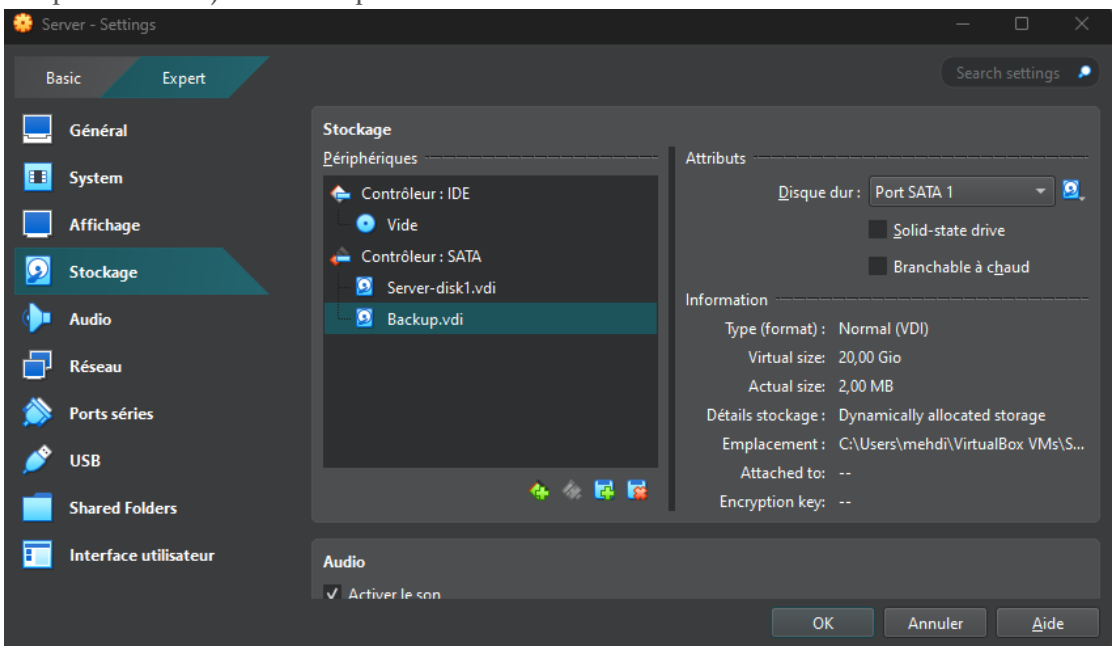
Planification des rotations :
La mise en place d'un plan de rotation des supports permet de limiter les risques liés à l'usure ou à la corruption des données. Cela consiste à utiliser plusieurs supports en alternance pour éviter de conserver les données sur un seul support pendant une période prolongée, tout en facilitant la récupération en cas de problème.

Cette organisation garantit la fiabilité des sauvegardes sur le long terme et minimise les risques de perte de données critiques.

Espace de stockage

EXERCICE 6.6 :

On procède à l'ajout du disque sur la vm :



EXERCICE 6.7 :

On liste les disques pour vérifier :

```
vboxuser@Server:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   10G  0 disk
├─sda1   8:1    0    9G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sdb      8:16   0   20G  0 disk
sr0     11:0    1 1024M  0 rom
vboxuser@Server:~$
```

On voit bien le disque de 20gb qu'on a ajouté.

On crée une nouvelle partition en étendu qui fait 20gb :

```
vboxuser@Server:~$ sudo fdisk /dev/sdb
[sudo] password for vboxuser:

Welcome to fdisk (util-linux 2.38.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x7ed79de8.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): e
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039):

Created a new partition 1 of type 'Extended' and of size 20 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

vboxuser@Server:~$
```

On formate la nouvelle partition :

```
vboxuser@Server:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   10G  0 disk
├─sda1   8:1    0    9G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sdb      8:16   0   20G  0 disk
└─sdb1   8:17   0   20G  0 part
sr0     11:0    1 1024M  0 rom
vboxuser@Server:~$
```

```
vboxuser@Server:~$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 5242624 4k blocks and 1310720 inodes
Filesystem UUID: dc93f038-9d8f-4910-885d-03bbbd57db68
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

vboxuser@Server:~$
```

On crée un point de montage, on monte puis on vérifie :

```
vboxuser@Server:~$ sudo mkdir /mnt/backup
vboxuser@Server:~$ sudo mount /dev/sdb1 /mnt/backup
vboxuser@Server:~$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
udev                       953M     0  953M   0% /dev
tmpfs                      197M  1.3M  196M   1% /run
/dev/sda1                  8.9G  7.8G  626M  93% /
tmpfs                      984M     0  984M   0% /dev/shm
tmpfs                      5.0M  8.0K  5.0M   1% /run/lock
192.168.0.1:/srv/nfs/home  8.9G  5.6G  2.8G  67% /mnt/nfs_home
tmpfs                      197M   88K  197M   1% /run/user/1000
/dev/sdb1                   20G   24K   19G   1% /mnt/backup
vboxuser@Server:~$
```

On ajoute une ligne au fichier /etc/fstab pour monter automatiquement le disque au démarrage:

```
 GNU nano 7.2                                                    /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=f315e985-b485-4741-b3ef-c6f313200116 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=a6a5c161-fdeb-41f7-8063-e0bd5300ba41 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
192.168.0.1:/srv/nfs/home /mnt/nfs_home nfs defaults 0 0
/dev/sdb1   /mnt/backup   ext4   defaults   0   2
```

EXERCICE 6.8 :

La méthode de stockage utilisée ici, consistant à monter un second disque dur local pour effectuer les sauvegardes, présente à la fois des avantages et des inconvénients.

Avantages :

1. **Performance élevée :**
   Étant un disque local, les opérations de lecture et d'écriture se font rapidement, sans dépendance à un réseau externe.

2. **Facilité de mise en place :**
   Ajouter un disque et configurer son montage est relativement simple et rapide à réaliser, ce qui permet une mise en œuvre immédiate.

3. **Coût faible :**
   Comparé à d'autres solutions comme le NAS ou le cloud, un disque dur local présente un coût d'acquisition et de maintenance relativement faible.

Inconvénients :

1. **Absence de redondance physique :**
   Si le serveur subit un problème matériel grave (incendie, panne critique, vol), les sauvegardes locales risquent d'être perdues en même temps que les données principales.

2. **Faible résilience :**
   En cas de panne du disque de sauvegarde, les données sauvegardées risquent d'être définitivement perdues si aucune copie supplémentaire n'est conservée ailleurs.

3. **Maintenance nécessaire :**
   Ce type de solution nécessite une surveillance régulière de l'état du disque (avec des outils comme smartctl pour vérifier la santé du disque).

4. **Pas de sauvegarde hors site :**
   Les sauvegardes étant stockées au même endroit que le serveur, il n'y a aucune protection en cas de sinistre majeur. Une bonne pratique serait de compléter cette méthode avec une solution de sauvegarde hors site, comme le cloud ou un disque dur externe régulièrement déplacé.

Proposition d'amélioration :

Pour renforcer cette méthode de stockage, je propose de mettre en place une stratégie de rotation des sauvegardes, avec :

1. **Un second disque externe :**
   Pour réaliser des copies régulières des sauvegardes et les stocker hors site.

2. **Une sauvegarde périodique dans le cloud :**
   Cette solution peut être envisagée pour des sauvegardes critiques (bases de données et fichiers essentiels), bien que cela représente un coût supplémentaire.

Sauvegarde

EXERCICE 6.9 :

Pour Pour organiser efficacement les fichiers de sauvegarde, je vais suivre une approche méthodique en définissant une structure de répertoires claire et hiérarchisée. Chaque sauvegarde sera stockée dans un dossier distinct, et ces dossiers seront classés selon le type de sauvegarde (complète ou incrémentale) et la date à laquelle la sauvegarde a été réalisée.

---

1. Nom des Dossiers de Sauvegarde

- Chaque sauvegarde sera placée dans un dossier nommé avec la date de la sauvegarde au format YYYY-MM-DD pour une identification rapide.
- Deux répertoires principaux seront créés :
    - full/ : Contiendra les sauvegardes complètes.
    - incremental/ : Contiendra les sauvegardes incrémentales.

Exemples de noms de dossiers :

- backup/full/2025-01-05/
- backup/incremental/2025-01-06/

---

2. Contenus des Dossiers de Sauvegarde

Chaque dossier de sauvegarde contiendra les fichiers et répertoires correspondants aux différents services sauvegardés, selon la structure suivante :

backup/

├── full/

│ ├── 2025-01-05/

│ │ ├── user_data/

│ │ │ └── user_full_backup_2025-01-05.tar.gz

│ │ ├── mysql/

│ │ │ └── mysql_full_backup_2025-01-05.sql

│ │ └── ldap/

```
│   │       └── ldap_full_backup_2025-01-05.ldif
│   └── ...
└── incremental/
    ├── 2025-01-06/
    │   ├── user_data/
    │   │   └── user_backup_incremental_2025-01-06.tar.gz
    │   ├── mysql/
    │   │   └── mysql_backup_incremental_2025-01-06.sql
    │   └── ldap/
    │       └── ldap_backup_incremental_2025-01-06.ldif
    └── ...
```

Description de la structure :

- Le répertoire principal backup/ contient deux sous-répertoires : full/ pour les sauvegardes complètes et incremental/ pour les sauvegardes incrémentales.
- Chaque sous-répertoire (full/ et incremental/) est subdivisé en dossiers nommés selon la date de la sauvegarde.
- À l'intérieur de chaque dossier daté, les données sont organisées par type :
    - user_data/ : Contient les fichiers compressés des répertoires /home.
    - mysql/ : Contient les dumps des bases de données MySQL.
    - ldap/ : Contient les exports de la base LDAP.

---

3. Rotation des Sauvegardes

- Afin de gérer efficacement l'espace de stockage, je mettrai en place une **politique de rotation** des sauvegardes :
    - Je conserverai les **3 dernières sauvegardes complètes** et les **7 dernières sauvegardes incrémentales**.
    - Les sauvegardes plus anciennes seront supprimées ou archivées sur un support de stockage externe.

Justifications

1. **Facilité de Gestion** :
   Cette organisation me permettra de retrouver rapidement une sauvegarde spécifique en fonction de sa date et de son type.

2. **Récupération Sélective** :
   En organisant les sauvegardes par type de données (utilisateurs, bases MySQL, LDAP), je pourrai récupérer uniquement les données nécessaires sans devoir restaurer l'ensemble du système.

3. **Historique Clair** :
   L'inclusion de la date dans le nom des dossiers permet de conserver un historique détaillé des sauvegardes, facilitant ainsi la traçabilité des opérations de sauvegarde.

EXERCICE 6.10 :

Pour sauvegarder les fichiers des utilisateurs tout en préservant leurs permissions et leurs propriétés (droits, dates, etc.), j'ai créé un script nommé backup_users.sh :

```
  GNU nano 7.2                                                backup_users.sh
#!/bin/bash
# Script de sauvegarde des fichiers utilisateurs

# Définir les variables
BACKUP_DIR="/mnt/backup/full/$(date +'%Y-%m-%d')/user_data"
SOURCE_DIR="/home"

# Créer le répertoire de sauvegarde
mkdir -p "$BACKUP_DIR"

# Effectuer la sauvegarde avec tar
tar -czvf "$BACKUP_DIR/user_full_backup_$(date +'%Y-%m-%d').tar.gz" "$SOURCE_DIR"

echo "Sauvegarde des fichiers utilisateurs terminée avec succès."
```

- **Explication** :
  - BACKUP_DIR : Répertoire de destination de la sauvegarde.
  - SOURCE_DIR : Répertoire contenant les fichiers des utilisateurs.
  - La commande tar est utilisée pour compresser les fichiers dans une archive .tar.gz, tout en préservant leurs propriétés.

Ensuite, j'ai attribué les permissions d'exécution au script :

```
vboxuser@Server:~$ nano backup_users.sh
vboxuser@Server:~$ sudo chmod +x backup_users.sh
vboxuser@Server:~$
```

EXERCICE 6.11 :

Lorsqu'il s'agit de copier des fichiers de sauvegarde sur un CD ou un DVD, plusieurs précautions doivent être prises pour garantir l'intégrité et la sécurité des données. Tout d'abord, il est essentiel de s'assurer que le support est correctement formaté et compatible avec le système utilisé. Le format ISO 9660 est souvent recommandé car il est largement pris en charge par les lecteurs et systèmes d'exploitation. De plus, la taille des fichiers doit être vérifiée pour ne pas dépasser la capacité du support, qui est généralement d'environ 700 Mo pour un CD et 4,7 Go pour un DVD. Si les fichiers sont trop volumineux, ils peuvent être divisés en plusieurs parties à l'aide d'outils comme split.

Une attention particulière doit être portée à l'intégrité des données. Avant de graver, il est recommandé de générer un checksum (MD5 ou SHA256) pour les fichiers de sauvegarde. Après la gravure, ce checksum doit être vérifié pour s'assurer que les données n'ont pas été altérées durant le processus. Par ailleurs, il faut tenir compte de la durabilité physique des supports, car les CD/DVD sont sensibles aux rayures, à l'humidité et à la lumière directe. Un stockage dans un environnement sec, à l'abri de la lumière, et dans des boîtiers de protection est essentiel pour prolonger leur durée de vie.

Pour les données sensibles, le chiffrement est crucial avant la gravure. Cela empêche tout accès non autorisé en cas de perte ou de vol du support. Enfin, il est judicieux de graver plusieurs copies des sauvegardes et de les conserver dans des lieux différents afin de réduire les risques de perte totale en cas de détérioration ou de sinistre. Ces précautions garantissent que les sauvegardes sur CD/DVD restent accessibles, sécurisées, et utilisables en cas de besoin.

EXERCICE 6.12 :

J'ai créé un script nommé backup_mysql.sh pour sauvegarder toutes les bases de données MySQL dans un fichier dump.

```
 GNU nano 7.2                                                backup_mysql.sh *
#!/bin/bash
# Script de sauvegarde des bases MySQL

# Définir les variables
BACKUP_DIR="/mnt/backup/full/$(date +'%Y-%m-%d')/mysql"
MYSQL_USER="root"
MYSQL_PASS="changeme"

# Créer le répertoire de sauvegarde
mkdir -p "$BACKUP_DIR"

# Effectuer la sauvegarde avec mysqldump
mysqldump -u $MYSQL_USER -p$MYSQL_PASS --all-databases > "$BACKUP_DIR/mysql_full_backup_$(date +'%Y-%m-%d').sql"

echo "Sauvegarde des bases MySQL terminée avec succès."
```

Explication :

- **BACKUP_DIR** : Répertoire où le fichier de sauvegarde sera stocké.
- **MYSQL_USER** et **MYSQL_PASS** : Identifiants de l'utilisateur MySQL ayant les droits de sauvegarde.
- **mysqldump --all-databases** : Sauvegarde l'ensemble des bases de données dans un seul fichier .sql.
- La sauvegarde est horodatée avec la date du jour pour une meilleure gestion des versions.

J'ai attribué les droits d'exécution au script avec la commande suivante :

```
vboxuser@Server:~$ nano backup_mysql.sh
vboxuser@Server:~$ sudo chmod +x backup_mysql.sh
vboxuser@Server:~$
```

EXERCICE 6.13 :

La cohérence des backups de bases de données est un enjeu crucial, en particulier dans des environnements où les données sont fréquemment modifiées. Lorsqu'une sauvegarde est effectuée pendant que la base de données est active, des incohérences peuvent survenir. Cela peut se manifester par des transactions partiellement enregistrées ou des dépendances entre tables rompues, rendant les données sauvegardées inutilisables ou corrompues.

Pour garantir la cohérence des sauvegardes, plusieurs approches existent. L'utilisation de verrouillage (locking) assure que la base reste stable pendant la sauvegarde en bloquant les modifications, mais cela peut affecter l'accès des utilisateurs. Une alternative est d'effectuer les sauvegardes en mode transactionnel, ce qui garantit que

toutes les données sauvegardées appartiennent au même état logique sans bloquer l'utilisation de la base.

Les snapshots au niveau du système de fichiers offrent également une solution robuste en capturant une image complète et cohérente du système à un instant donné. Pour les bases volumineuses, les sauvegardes binaires sont souvent préférées, car elles sont rapides et préservent l'intégrité physique des fichiers. Enfin, planifier les sauvegardes pendant des périodes de faible activité réduit les risques d'incohérence tout en minimisant l'impact sur les performances.

Ces solutions doivent être choisies en fonction des besoins spécifiques, comme le niveau d'activité de la base, les contraintes opérationnelles, et la taille des données. Une combinaison de ces techniques peut souvent offrir une sauvegarde fiable et cohérente.

EXERCICE 6.14 :

Problèmes liés à une base MySQL volumineuse :

1. **Temps de sauvegarde long** :
    - Les bases de données volumineuses nécessitent un temps considérable pour être sauvegardées, ce qui peut affecter les performances du système, surtout si la base reste active pendant la sauvegarde.
2. **Impact sur les performances** :
    - Pendant la sauvegarde, la charge du serveur peut augmenter, entraînant une dégradation des performances pour les utilisateurs et les applications en cours d'utilisation.
3. **Espace de stockage insuffisant** :
    - Les fichiers de sauvegarde peuvent occuper un espace important, ce qui peut poser des problèmes si le disque destiné aux sauvegardes est limité.
4. **Problèmes de cohérence** :
    - Pour des bases très volumineuses, les risques d'incohérence augmentent, surtout si des modifications interviennent pendant le processus de sauvegarde.
5. **Difficultés de restauration** :
    - Restaurer une base volumineuse peut être un processus long et complexe, en particulier si les fichiers de sauvegarde sont corrompus ou s'il manque des sauvegardes incrémentales.

Solutions proposées :

1. **Sauvegardes incrémentales** :
    - Limiter la taille des sauvegardes en sauvegardant uniquement les modifications depuis la dernière sauvegarde complète. Cela réduit le temps et l'espace nécessaires pour chaque opération.
2. **Compression des sauvegardes** :
    - Utiliser des outils pour compresser les fichiers de sauvegarde, réduisant ainsi l'espace disque requis.
3. **Utilisation de snapshots** :
    - Créer des snapshots du système de fichiers ou du disque pour effectuer des sauvegardes rapides et cohérentes, indépendamment de la taille de la base.
4. **Outils spécialisés pour bases volumineuses** :
    - Utiliser des solutions comme Percona XtraBackup, qui sont conçues pour gérer efficacement les sauvegardes de grandes bases de données sans perturber les opérations en cours.
5. **Partitionnement de la base** :
    - Organiser la base en partitions pour gérer les données de manière modulaire et faciliter les sauvegardes partielles.
6. **Planification stratégique** :
    - Effectuer les sauvegardes pendant les périodes de faible activité pour minimiser l'impact sur les performances et le système.
7. **Stockage supplémentaire** :
    - Ajouter un espace de stockage dédié aux sauvegardes afin d'assurer qu'aucune limitation ne perturbe le processus.

EXERCICE 6.15 :

On crée le fichier script qui contient le code de backup de la base LDAP:

```
 GNU nano 7.2                                                backup_ldap.sh *
#!/bin/bash
# Script de sauvegarde de la base LDAP

# Définir les variables
BACKUP_DIR="/mnt/backup/full/$(date +'%Y-%m-%d')/ldap"

# Créer le répertoire de sauvegarde
mkdir -p "$BACKUP_DIR"

# Effectuer le backup de la base LDAP avec slapcat
slapcat -v -l "$BACKUP_DIR/ldap_full_backup_$(date +'%Y-%m-%d').ldif"

echo "Sauvegarde de la base LDAP terminée avec succès."
```

On lui donne lui donne les permissions d'exécution :

```
vboxuser@Server:~$ chmod +x backup_ldap.sh
vboxuser@Server:~$
```

EXERCICE 6.16 :

On crée un checksum MD5 des fichiers de sauvegarde pour permettre de vérifier leurs intégrités en cas de nécessité :

```
 GNU nano 7.2                                          backup_checksum.sh *
# Chemin vers le répertoire de sauvegarde
backup_directory="/mnt/backup"

# Fonction pour générer le checksum MD5
generate_md5_checksum() {
    local target_directory="$1"
    local checksum_file="$2"

    echo "Génération du checksum pour $target_directory"
    find "$target_directory" -type f -exec md5sum {} \; > "$checksum_file"
    echo "Checksum généré : $checksum_file"
}

# Génération du checksum pour les sauvegardes complètes
full_backup_directory="$backup_directory/full/$(date +%F)"
full_checksum_file="$full_backup_directory/checksum.txt"
generate_md5_checksum "$full_backup_directory" "$full_checksum_file"

# Génération du checksum pour les sauvegardes incrémentales
incremental_backup_directory="$backup_directory/incremental/$(date +%F)"
incremental_checksum_file="$incremental_backup_directory/checksum.txt"
generate_md5_checksum "$incremental_backup_directory" "$incremental_checksum_file"

^G Help        ^O Write Out   ^W Where Is   ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace    ^U Paste       ^J Justify     ^/ Go To Line
```

Explication :

Chemin vers les répertoires de sauvegarde :

full_backup_directory et incremental_backup_directory contiennent les chemins complets des sauvegardes en fonction de la date du jour.

Fonction generate_md5_checksum : Cette fonction parcourt tous les fichiers dans le répertoire de sauvegarde et génère un fichier checksum.txt contenant les sommes de contrôle MD5 de chaque fichier.

Création des checksums :

Le script génère un checksum pour les sauvegardes complètes et un autre pour les sauvegardes incrémentales.

Sortie : Chaque checksum est enregistré dans un fichier checksum.txt dans le répertoire correspondant.

On Exécute les commandes suivantes pour donner les permissions d'exécution au script :

```
vboxuser@Server:~$ chmod +x backup_checksum.sh
vboxuser@Server:~$
```

EXERCICE 6.17 :

Objectif : Configurer les règles cron pour exécuter automatiquement les scripts de backup selon le planning établi dans Exercice 6.2 :

Sauvegarde complète : Une fois par semaine, le dimanche à 2h du matin.

Sauvegardes incrémentales : Quotidiennes du lundi au samedi à 2h du matin.

Sauvegarde des bases critiques (MySQL, LDAP) : Deux fois par jour, à 12h et 23h.

On utilise **crontab**

```
vboxuser@Server:~$ ls -l /usr/local/bin/backup_scripts/
total 16
-rwxr-xr-x 1 vboxuser vboxuser 941 Jan  5 23:26 backup_checksum.sh
-rwxr-xr-x 1 vboxuser vboxuser 375 Jan  5 23:22 backup_ldap.sh
-rwxr-xr-x 1 vboxuser vboxuser 449 Jan  5 23:11 backup_mysql.sh
-rwxr-xr-x 1 vboxuser vboxuser 415 Jan  5 23:03 backup_users.sh
vboxuser@Server:~$ sudo crontab -e
```

```
  GNU nano 7.2                                                /tmp/crontab.wdEjpD/crontab *
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 2 * * 0 /usr/local/bin/backup_scripts/backup_users.sh
0 2 * * 0 /usr/local/bin/backup_scripts/backup_mysql.sh
0 2 * * 0 /usr/local/bin/backup_scripts/backup_ldap.sh

0 2 * * 1-6 /usr/local/bin/backup_scripts/backup_users.sh --incremental
0 2 * * 1-6 /usr/local/bin/backup_scripts/backup_mysql.sh --incremental
0 2 * * 1-6 /usr/local/bin/backup_scripts/backup_ldap.sh --incremental

0 12 * * * /usr/local/bin/backup_scripts/backup_mysql.sh --incremental
0 23 * * * /usr/local/bin/backup_scripts/backup_mysql.sh --incremental

0 12 * * * /usr/local/bin/backup_scripts/backup_ldap.sh --incremental
0 23 * * * /usr/local/bin/backup_scripts/backup_ldap.sh --incremental
```

Justification :

Heure de nuit (2h du matin) : Cette plage horaire est choisie pour minimiser l'impact sur les performances du serveur pendant les heures de faible activité.

Heures de journée (12h et 23h) : Ces plages horaires permettent de capturer les modifications critiques des bases de données à des moments stratégiques de la journée.

Fréquence accrue pour les bases de données critiques : Justifiée par leur taux de modification plus élevé par rapport aux autres types de données.

Restauration

EXERCICE 6.18:

Étape 1 : Générez un backup entier (complet)

Pour générer une sauvegarde complète, j'ai utilisé le script backup_users.sh, backup_mysql.sh et backup_ldap.sh, que j'avais déjà créés et déplacés dans le répertoire /usr/local/bin/backup_scripts/.

Les commandes exécutées sont les suivantes :

sudo /usr/local/bin/backup_scripts/backup_users.sh

sudo /usr/local/bin/backup_scripts/backup_mysql.sh

sudo /usr/local/bin/backup_scripts/backup_ldap.sh

Les fichiers de sauvegarde générés se trouvent dans le répertoire /mnt/backup/full/YYYY-MM-DD/

```
vboxuser@Server:~$ sudo ls /mnt/backup/full
2025-01-05
vboxuser@Server:~$ sudo ls /mnt/backup/full/2025-01-05/
ldap  mysql  user_data
vboxuser@Server:~$
```

selon la structure définie précédemment, avec les sauvegardes des répertoires utilisateurs, des bases de données MySQL et de la base LDAP.

```
/home/vboxuser/.local/share/flatpak/db/
/home/vboxuser/.local/share/keyrings/
/home/vboxuser/.local/share/keyrings/user.keystore
/home/vboxuser/.local/share/keyrings/login.keyring
/home/vboxuser/.local/state/
/home/vboxuser/.local/state/wireplumber/
/home/vboxuser/.local/state/wireplumber/restore-stream
/home/vboxuser/.local/state/wireplumber/default-routes
/home/vboxuser/.sudo_as_admin_successful
/home/vboxuser/hosts.ldif
/home/vboxuser/.vboxclient-seamless-tty2-control.pid
Sauvegarde des fichiers utilisateurs terminée avec succès.
vboxuser@Server:~$ sudo /usr/local/bin/backup_scripts/backup_mysql.sh
Sauvegarde des bases MySQL terminée avec succès.
vboxuser@Server:~$ sudo /usr/local/bin/backup_scripts/backup_ldap.sh
# id=00000001
# id=00000002
# id=00000003
# id=00000004
# id=00000005
# id=00000006
# id=00000007
# id=00000008
Sauvegarde de la base LDAP terminée avec succès.
vboxuser@Server:~$
```

Étape 2 : Générez un backup incrémental

Pour la sauvegarde incrémentale, j'ai utilisé le même répertoire de scripts et exécuté les commandes suivantes :

sudo /usr/local/bin/backup_scripts/backup_users.sh --incremental

sudo /usr/local/bin/backup_scripts/backup_mysql.sh --incremental

sudo /usr/local/bin/backup_scripts/backup_ldap.sh --incremental

Les sauvegardes incrémentales ont été générées dans le répertoire /mnt/backup/incremental/YYYY-MM-DD/.

```
/home/vboxuser/.local/share/keyrings/user.keystore
/home/vboxuser/.local/share/keyrings/login.keyring
/home/vboxuser/.local/state/
/home/vboxuser/.local/state/wireplumber/
/home/vboxuser/.local/state/wireplumber/restore-stream
/home/vboxuser/.local/state/wireplumber/default-routes
/home/vboxuser/.sudo_as_admin_successful
/home/vboxuser/hosts.ldif
/home/vboxuser/.vboxclient-seamless-tty2-control.pid
Sauvegarde des fichiers utilisateurs terminée avec succès.
vboxuser@Server:~$ sudo /usr/local/bin/backup_scripts/backup_mysql.sh --incremental
Sauvegarde des bases MySQL terminée avec succès.
vboxuser@Server:~$ sudo /usr/local/bin/backup_scripts/backup_ldap.sh --incremental
# id=00000001
# id=00000002
# id=00000003
# id=00000004
# id=00000005
# id=00000006
# id=00000007
# id=00000008
Sauvegarde de la base LDAP terminée avec succès.
vboxuser@Server:~$
```

Elles ne contiennent que les fichiers modifiés ou ajoutés depuis la dernière sauvegarde complète ou incrémentale, conformément à la stratégie de sauvegarde mise en place.

EXERCICE 6.19:

On Crée une archive de la sauvegarde complète :

```
vboxuser@Server:~$ sudo tar -czvf /tmp/full_backup.tar.gz -C /mnt/backup/full .
./
./2025-01-05/
./2025-01-05/mysql/
./2025-01-05/mysql/mysql_full_backup_2025-01-05.sql
./2025-01-05/ldap/
./2025-01-05/ldap/ldap_full_backup_2025-01-05.ldif
./2025-01-05/user_data/
./2025-01-05/user_data/user_full_backup_2025-01-05.tar.gz
```

Puis , On Transfére l'archive vers la machine client1 (192.168.0.3) :

```
vboxuser@Server:~$ scp /tmp/full_backup.tar.gz vboxuser@192.168.0.3:/tmp
The authenticity of host '192.168.0.3 (192.168.0.3)' can't be established.
ED25519 key fingerprint is SHA256:TaJHDru8DYW5aRGZ6BhWLdDx4b4ylGmdix07j2TBMYU.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.0.3' (ED25519) to the list of known hosts.
vboxuser@192.168.0.3's password:
full_backup.tar.gz                                                                    100%   53MB  11.1MB/s   00:04
vboxuser@Server:~$
```

On Se connecte à la machine client1 ,Ensuite on Crée un répertoire pour les backups sur client1 :

```
vboxuser@Client1:~$ tar -xzvf /tmp/full_backup.tar.gz -C /mnt/backup/full
./
./2025-01-05/
./2025-01-05/mysql/
./2025-01-05/mysql/mysql_full_backup_2025-01-05.sql
./2025-01-05/ldap/
./2025-01-05/ldap/ldap_full_backup_2025-01-05.ldif
./2025-01-05/user_data/
./2025-01-05/user_data/user_full_backup_2025-01-05.tar.gz
vboxuser@Client1:~$
```

Décompresse l'archive contenant les données utilisateur dans le répertoire /home :

```
root@Client1:~# tar -xzvf /mnt/backup/full/2025-01-05/user_data/user_full_backup
_2025-01-05.tar.gz -C /home
home/
home/rochdi/
home/rochdi/.config/
home/rochdi/.config/goa-1.0/
home/rochdi/.face.icon
home/rochdi/nouveau_fichier.txt
home/rochdi/.bashrc
home/rochdi/.face
home/rochdi/.bash_logout
home/rochdi/.bash_history
home/rochdi/.cache/
home/rochdi/.cache/gstreamer-1.0/
home/rochdi/.cache/gstreamer-1.0/registry.x86_64.bin
home/rochdi/.cache/tracker3/
home/rochdi/.cache/tracker3/files/
home/rochdi/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%
2Fv3%2Ftracker%23Documents.db
home/rochdi/.cache/tracker3/files/ontologies.gvdb
home/rochdi/.cache/tracker3/files/meta.db
home/rochdi/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%
2Fv3%2Ftracker%23Audio.db
home/rochdi/.cache/tracker3/files/first-index.txt
```

2. Restauration de la base de données MySQL :

On Assure que le service MySQL est en cours d'exécution :

- sudo systemctl start mysql

Ensuite, On restaure la base de données avec la commande suivante :

- mysql -u root -p < /mnt/backup/full/2025-01-05/mysql/mysql_full_backup_2025-01-05.sql

On Saisi le mot de passe root de MySQL lorsque demandé.

3. Restauration de la base LDAP :

On Assure que le service LDAP (slapd) est en cours d'exécution :

- sudo systemctl start slapd

Puis, on importe les données LDAP avec la commande suivante :

- ldapadd -x -D "cn=admin,dc=istycorp,dc=fr" -W -f /mnt/backup/full/2025-01-05/ldap/ldap_full_backup_2025-01-05.ldif

EXERCICE 6.20:

Lorsque l'utilisateur "**raj**" a supprimé par erreur son dossier **htop-dev**, la restauration a été réalisée en ciblant uniquement ce dossier spécifique, sans affecter les fichiers des autres utilisateurs. Tout d'abord, les sauvegardes ont été examinées pour identifier l'archive contenant la dernière version valide du dossier. À l'aide de la commande tar --list, le contenu de l'archive a été vérifié pour s'assurer que le dossier **htop-dev** était bien présent.

La restauration a ensuite été effectuée avec la commande tar --extract, en spécifiant uniquement le dossier **htop-dev** comme cible. Cette opération a permis d'extraire le dossier directement dans le répertoire personnel de "**raj**" sans altérer les autres fichiers ou répertoires. Si des modifications ultérieures avaient été enregistrées dans une sauvegarde incrémentale, une extraction complémentaire aurait été réalisée pour intégrer les changements récents.

Enfin, les permissions et droits d'accès du dossier restauré ont été vérifiés et corrigés pour garantir que "**raj**" pouvait reprendre son travail immédiatement. Cette approche ciblée garantit une restauration précise, minimisant les risques pour les données des autres utilisateurs et respectant les principes de gestion des sauvegardes.