# R5B08 - Continuité de services

## Sauvegarde/Restauration

Bruno BEAUFILS

2024/2025

# 1. Généralités

# 2. Sauvegardes : concepts

# 3. Sauvegardes : bonnes pratiques

# Continuité de services

## Objectifs

- **minimiser le temps d'arrêt des services et le coût de reprise**
  - Perte de données admissible vs Durée Maximale d'Interruption Admissible
    - *Recovery Point Objective* vs *Recovery Time Objective*
  - Plan de Reprise d'Activité vs Plan de Continuité d'Activité
    - *Disaster Recovery Plan* vs *Business Continuity Plan*

## Moyens

- description de **procédures**
- mise en place **d'outils**
- description des individus, rôles et tâches

# Plans

- Plan de Continuité d'Activités (Plan de Reprise d'Activités)
  1. **Analyse des risques**
     - identifier les menaces
     - identifier les activités critiques
  2. **Analyse des impacts**
     - identifier l'impact d'un risque réalisé
     - identifier les impacts intolérables
  3. **Stratégie de sécurisation**
     - décrire et appliquer les processus de sécurisation
- Stratégie de sécurisation
  - mesures **préventives**
    - **sauvegardes**
    - secours (*spares*)
  - mesures **curatives**
    - **restauration** des données
    - redémarrage applications et machines
  - exercice
    - tests régulier de l'application des plans
  - mise à jour régulière

# Pourquoi sauvegarder ?

## Un seul objectif

**Pouvoir récupérer les données après leur disparition**

## Pourquoi / Comment

- Comme pour les **assurances**
  - on investit dedans
  - on espère ne pas avoir à s'en servir
- Techniques
  - redondance : on ajoute de l'information pour reconstruire les données
  - **réplication** : on recopie les données

# Perte de données

- **Nature**
  - disparition partielle = modification
  - disparition complète = « *désastre* »
- **Origine**
  - panne matérielle
  - suppression involontaire
    - erreur de manipulation
    - bug dans un logiciel
  - suppression volontaire
    - malveillance
- **Inévitable**

# Besoins d'accès

## Court terme : sauvegarde

- pertes
  - erreur utilisateur
  - erreur logiciel
  - panne matériel
  - violation de sécurité
  - catastrophe naturelle
- récupération
  - fichiers
  - systèmes
  - tout (désastre)

## Long terme : archivage

- propriétés : pérennité, intégrité, confidentialité, accessibilité
- données immuables
- séparé des sauvegardes régulières, généralement hors-site
- restauration lente
- chiffrement important (gestion des clés)
- besoins légaux, patrimoniaux

# Rappel : accès aux données

## Blocs

- accès par bloc d'octets (*block device*)
  - taille dépendante du support
  - accès bas niveau
- local : disque, partition, volume (RAID, LVM, etc.)
- réseau : accès à un périphérique distant (NBD, iSCSI, Fibre Channel, etc.)

## Fichiers

- accès via un système de fichiers (dossiers, permissions, etc.)
  - norme POSIX (cf `hier(7)`)
  - transparent pour les applications
- local : stocké dans un block device (`ext4`, `btrfs`, `zfs`, etc.)
- réseau : accessible à travers le réseau
  - client/serveur : `nfs`, `cifs`
  - distribué : `afs`, `ceph`, `glusterfs`, `lustre`, etc.

# Rappel : accès aux données (suite)

## Applicatifs

- **SQL**
  - commandes de création, mise à jour de tables
  - `pg_dump` / `psql`, `mysqldump` / `mysql`
- **Objets** (*blobs*)
  - accès par des méthodes spécifiques
    - méta-données
    - identifiant unique
    - données
  - utilisé dans le *cloud*
    - dépend des besoins
  - exemples : S3, NoSQL (`redis`, `mongodb`, etc.)

1. Généralités

2. Sauvegardes : concepts

3. Sauvegardes : bonnes pratiques

# Terminologie

- technique de sauvegarde
  - **réplication** (copie)
  - **déduplication**
  - redondance *(pas vraiment une sauvegarde)*
- type de sauvegarde
  - complète
  - incrémentale
  - différentielle
- accès aux sauvegardes
  - **journalisation**
    - trace des modifications
    - construction des données
  - **snapshot**
    - image des données à un moment donné
- contenu
  - méta-données (propriétaire, droits, etc.)
  - données

# Sauvegarde complète

## Principe

**Copie pure et simple des données**

- on copie intégralement les données d'un disque sur un autre
- pour restaurer on fait l'opération inverse

## Détails

- avantages : simplicité
- inconvénients : double espace de stockage nécessaire
- outils
  - `dd(1)` : copier un fichier dans un autre
    - fonctionne avec les fichiers blocs
  - `tar(1)` : rassembler des fichiers dans un autre (archive)
    - conserve les propriétés des fichiers (droits, propriétaires)
    - archive peut être compressée/décompressée via par exemple `gzip/gunzip`

# Sauvegarde complète

## Principe

**Copie pure et simple des données**

- on copie intégralement les données d'un disque sur un autre
- pour restaurer on fait l'opération inverse

## Détails

- avantages : simplicité
- inconvénients : double espace de stockage nécessaire
- outils
  - `dd(1)` : copier un fichier dans un autre
    - fonctionne avec les fichiers blocs
  - `tar(1)` : rassembler des fichiers dans un autre (archive)
    - conserve les propriétés des fichiers (droits, propriétaires)
    - archive peut être compressée/décompressée via par exemple `gzip/gunzip`

# Sauvegarde incrémentale

## Principe

Ne copier que les **données modifiées depuis la dernière fois**

- nécessite une sauvegarde complète
- notion de **niveau de sauvegarde** : modifications considérées depuis quelle sauvegarde ?
  - sauvegarde intégrale — niveau **0**
  - depuis la dernière — niveau **1**
  - depuis l'avant-dernière — niveau **2**
  - depuis *n* sauvegardes — niveau **n**

## Détails

- avantages : réduit l'espace de stockage nécessaire
- inconvénients : restauration étape par étape
- outils
  - `dump(8)` : sauvegarder un système de fichiers ou un répertoire
  - `restore(8)` : restaurer l'état à partir de dumps
    - il faut appliquer les sauvegardes les unes après les autres

# Sauvegarde incrémentale

## Principe

Ne copier que les **données modifiées depuis la dernière fois**

- nécessite une sauvegarde complète
- notion de **niveau de sauvegarde** : modifications considérées depuis quelle sauvegarde ?
  - sauvegarde intégrale — niveau **0**
  - depuis la dernière — niveau **1**
  - depuis l'avant-dernière — niveau **2**
  - depuis *n* sauvegardes — niveau **n**

## Détails

- avantages : réduit l'espace de stockage nécessaire
- inconvénients : restauration étape par étape
- outils
  - `dump(8)` : sauvegarder un système de fichiers ou un répertoire
  - `restore(8)` : restaurer l'état à partir de dumps
    - il faut appliquer les sauvegardes les unes après les autres

# Sauvegarde différentielle

## Principe

Ne copier que les **données modifiées depuis la dernière copie complète**

- nécessite une sauvegarde complète

## Détails

- avantages : restauration plus rapide qu'avec de l'incrémental
- inconvénients : nécessite plus d'espace qu'avec de l'incrémental
- outils
  - `rsync` : copie différentielle de fichiers
    - permet de ne copier que les fichiers modifiés
    - permet de supprimer les fichiers disparus
    - nécessite une **référence**

# Sauvegarde différentielle

## Principe

Ne copier que les **données modifiées depuis la dernière copie complète**

- nécessite une sauvegarde complète

## Détails

- avantages : restauration plus rapide qu'avec de l'incrémental
- inconvénients : nécessite plus d'espace qu'avec de l'incrémental
- outils
  - `rsync` : copie différentielle de fichiers
    - permet de ne copier que les fichiers modifiés
    - permet de supprimer les fichiers disparus
    - nécessite une **référence**

# Déduplication

## Principe

**Factoriser le contenu binaire via un index**

- découper les données en blocs identifiés
- stocker une seule fois les blocs identifiés
- garder une donnée via une liste des index de ses blocs

## Détails

- avantages : réduction de l'espace
- inconvénients : risque de perte des données
- outils :
  - pas d'outil Unix standard
  - BorgBackup, restic, KOPIA

# Déduplication

## Principe

**Factoriser le contenu binaire via un index**

- découper les données en blocs identifiés
- stocker une seule fois les blocs identifiés
- garder une donnée via une liste des index de ses blocs

## Détails

- avantages : réduction de l'espace
- inconvénients : risque de perte des données
- outils :
  - pas d'outil Unix standard
  - BorgBackup, restic, KOPIA

1. Généralités

2. Sauvegardes : concepts

# 3. Sauvegardes : bonnes pratiques

# Quelques règles

- règle mnémotechnique : **3-2-1**
  - **3 copies des données**
    - chaque données doit exister en 3 exemplaires
    - (original + 2 copies)
  - **2 supports différents**
    - chaque données doit exister sur 2 supports différents
  - **1 copie hors-site**
    - chaque donnée doit exister sur une copie « hors site »
    - ailleurs que là où les données sont utilisées
- Sauvegarde de Schrödinger
  *« The condition of any backup is unknown until a restore is attempted »*

# Quelques règles

- règle mnémotechnique : **3-2-1**
  - **3 copies des données**
    - chaque données doit exister en 3 exemplaires
    - (original + 2 copies)
  - **2 supports différents**
    - chaque données doit exister sur 2 supports différents
  - **1 copie hors-site**
    - chaque donnée doit exister sur une copie « hors site »
    - ailleurs que là où les données sont utilisées
- Sauvegarde de Schrödinger
  *« The condition of any backup is unknown until a restore is attempted »*

# Considérations

- **confiance**
  - sauvegarde = copie
    - si les données initiales étaient corrompues, la sauvegarde aussi
  - n'utiliser que des outils de confiance
  - vérifier l'**intégrité** des copies
- Sauvegarde : propriétés à imposer
  - **régulière**
  - **fréquente**
  - **automatique**
  - **vérifiée** régulièrement
  - **chiffrée**

# Outils

## Fonctionnalités à considérer

- compression
- planification
- accès à travers le réseau
- accès aux fichiers en cours d'utilisation (*lock*)

## Panorama succint

- libre
  - rsync, rdiff-backup, rsnapshot
  - BorgBackup, restic, KOPIA
    - Comparaison des 3
  - Duplicity (utilise des archives tar)
  - Burp
- privatrice
  - Veeam : solution privatrice intégrée

## Critères de choix

- Pérennité de l'outil
- Adaptation au contexte (capacité, vitesse de transfert, fiabilité, etc.)

# Outils

## Fonctionnalités à considérer

- compression
- planification
- accès à travers le réseau
- accès aux fichiers en cours d'utilisation (*lock*)

## Panorama succint

- libre
  - rsync, rdiff-backup, rsnapshot
  - BorgBackup, restic, KOPIA
    - Comparaison des 3
  - Duplicity (utilise des archives tar)
  - Burp
- privatrice
  - Veeam : solution privatrice intégrée

## Critères de choix

- Pérennité de l'outil
- Adaptation au contexte (capacité, vitesse de transfert, fiabilité, etc.)