https://docs.google.com/document/d/1D7lRazHA74SM7VIe-e3hWMVDsjzxDn4x-qYP_Rtttgc/edit?usp=sharing

Sauvegarde base des donées et fichier Web

🖥️ Machine 1 → `debian`

Serveur source, celui qui contient les bases de données + sites web (OCS/GLPI plus tard)

🖥️ Machine 2 → `serveurBackup`

Serveur de sauvegarde avec `rsnapshot` + clés SSH

📦 Objectif final :

Depuis `serveurBackup`, tu pourras lancer automatiquement tous les jours :

- Une connexion SSH vers `debianOcs`
- Un `mysqldump` distant
- Une copie des fichiers/dumps avec `rsync` via `rsnapshot`
- Le tout géré par des **rotations (daily/weekly/monthly)**

2. On va créer un user `save` sur les deux

Sur **les deux machines** :

```
Ajout de l'utilisateur « save » au groupe « users » ...
root@deb12:~# sudo usermod -aG sudo save
```

Générer les clés SSH sur `serveurBackup`

`Tu auras une clé privée ici : ~/.ssh/id_rsa et une clé publique ~/.ssh/id_rsa.pub`

```
root@deb12:~# ssh-keygen -t rsa -b 7096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:KE9SU64QK1BZTpSkLSRUM2jOr1kEB16TFMZKcpVWZK8 root@deb12
The key's randomart image is:
+---[RSA 7096]----+
| .=X/%O..        |
|..*=O%.+         |
| B.=+ = o        |
|  + .+ =         |
|   oo E S        |
|    o=           |
|   +  .          |
|  o              |
|                 |
+----[SHA256]-----+
```

Étape 2 : Copier la clé publique sur BSD-Test

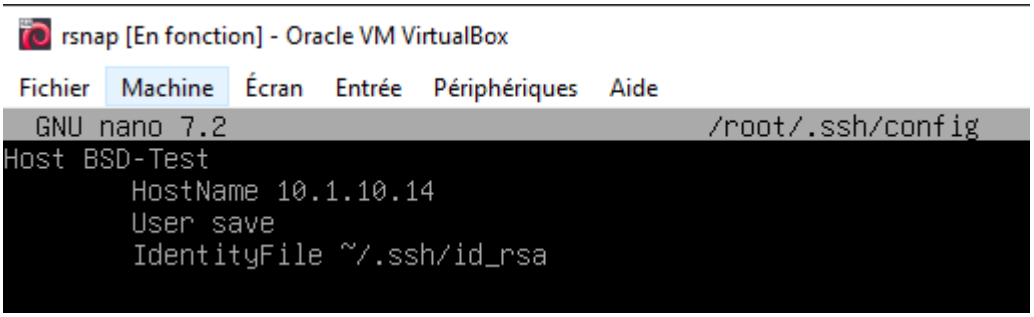
Toujours sur serveurBackup, lance :

```
root@deb12:~# ssh-copy-id save@10.1.10.14
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '10.1.10.14 (10.1.10.14)' can't be established.
ED25519 key fingerprint is SHA256:cOHqSaOxHBFgoNlMmdduT16eLHxjWVkO5QU3yh52v4U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
save@10.1.10.14's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'save@10.1.10.14'"
and check to make sure that only the key(s) you wanted were added.

root@deb12:~#
```

Machine source (avec les bases à sauvegarder) → bsd → 10.1.10.14

Machine de backup (avec rsnapshot) → rsnap → 10.1.10.13

🔥 Optionnel mais clean : fichier ~/.ssh/config sur rsnap



On rentres directement sur machine base des donnees sans mot de passe → **c'est bon** ✅

```
root@deb12:~# ssh BSD-Test
Linux deb12 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 24 09:21:38 2025 from 10.1.10.13
save@deb12:~$
```

Créer une base MySQL de test + un utilisateur backupbdd

On va faire ça sur la machine **bsd** (IP: 10.1.10.14), qui est le serveur source.

Installer MariaDB sur bsd-test (10.1.10.14)

```
root@deb12:~# sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script
ll.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
root@deb12:~# sudo systemctl start mariadb
root@deb12:~# _
```

Créer la base + l'utilisateur backupbdd

```
MariaDB [(none)]> CREATE DATABASE test_sauvegarde;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'backupbdd'@'localhost' IDENTIFIED BY 'eleve';
Query OK, 0 rows affected (0,009 sec)

MariaDB [(none)]> GRANT SELECT, LOCK TABLES ON test_sauvegarde.* TO 'backupbdd'@'localhost';
Query OK, 0 rows affected (0,006 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> EXIT;
Bye
root@deb12:~#
```

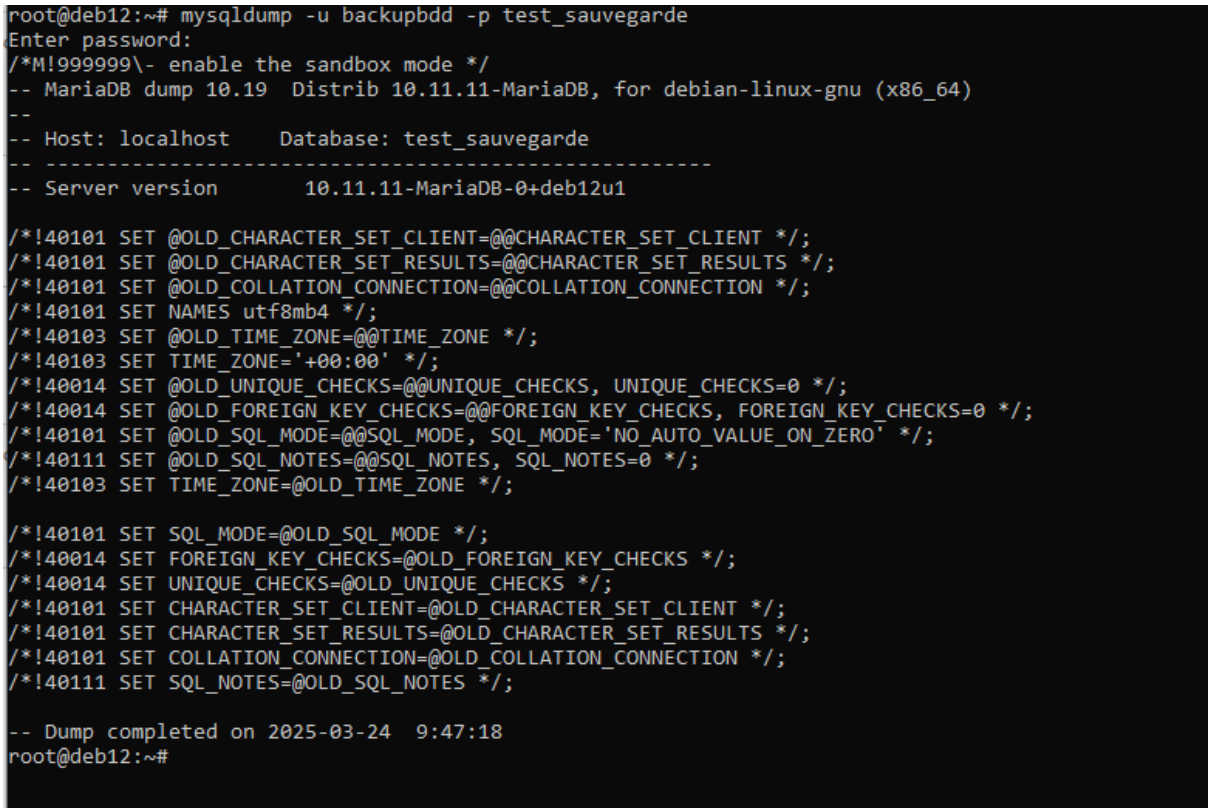
test

```
root@deb12:~# mysql -u backupbdd -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test_sauvegarde    |
+--------------------+
```

🔹 Test de dump :

Tu viens de réussir un **dump complet** de la base `test_sauvegarde` avec l'utilisateur `backupbdd` ✅

Ça veut dire que les droits sont bons, la base est prête, et le mot de passe fonctionne. On peut passer à l'automatisation.



```
root@deb12:~# mysqldump -u backupbdd -p test_sauvegarde
Enter password:
/*M!999999\- enable the sandbox mode */
-- MariaDB dump 10.19  Distrib 10.11.11-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: test_sauvegarde
-- ------------------------------------------------------
-- Server version       10.11.11-MariaDB-0+deb12u1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2025-03-24  9:47:18
root@deb12:~#
```

Script de sauvegarde automatique

On va créer un script sur `bsd` (la machine source) qui :

- Génère un dump `.sql` tous les jours
- Le stocke dans `/home/save/dumpsql/`

- Utilise l'utilisateur backupbdd

Tu dois créer le script **sur la machine bsd (10.1.10.14)**, **en tant que l'utilisateur save**, vu que c'est lui qui gère la sauvegarde sur ce serveur.

```
root@deb12:~# mkdir -p /home/save/dumpsql
root@deb12:~# nano /home/save/backup_bdd.sh
root@deb12:~# chmod +x /home/save/backup_bdd.sh
root@deb12:~#
```

```
  GNU nano 7.2                                  /home/save/backup_bdd.sh *
#!/bin/bash

# Date actuelle
DATE=$(date +%F)

# Dossier de sauvegarde
DEST="/home/save/dumpsql"

# Utilisateur BDD
USER="backupbdd"
PASS="eleve"
DB="test_sauvegarde"

# 🗄️  Dump de la base
mysqldump -u $USER -p$PASS $DB > $DEST/${DB}_$DATE.sql
```

✅ Test du script :

Ton script fonctionne parfaitement et le fichier .sql est bien généré dans /home/save/dumpsql/ → t'es officiellement prêt pour l'automatisation !

```
root@deb12:~# /home/save/backup_bdd.sh
root@deb12:~# ls /home/save/dumpsql/
test_sauvegarde_2025-03-24.sql
root@deb12:~#
```

Étape suivante : Planifier le script avec cron (à 3h du matin)

On va dire à Linux de lancer /home/save/backup_bdd.sh **tous les jours à 03h00 du matin**, sans que tu fasses rien

```
root@deb12:~# crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
crontab: installing new crontab
root@deb12:~#
```

```
# m h  dom mon dow   command
0 3 * * * /home/save/backup_bdd.sh
```

🎯 Objectif :

Depuis le machine **rsnap**, on va :

- Utiliser **rsnapshot** pour se connecter en SSH à bsd
- Récupérer les dumps SQL situés dans /home/save/dumpsql
- Sauvegarder ça localement dans /media/nas/sauvegarde
- Gérer la rotation : **7 jours / 4 semaines / 12 mois**

✅ Étape 1 : Installer rsnapshot sur rsnap

```
root@deb12:~# apt install rsnapshot -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  liblchown-perl rsync
```

Créer le dossier de sauvegarde local

```
root@deb12:~# sudo mkdir -p /media/nas/sauvegarde/
root@deb12:~# sudo chmod -R 755 /media/nas/sauvegarde/
root@deb12:~# _
```

Modifier la config de rsnapshot

```
root@deb12:~# nano /etc/rsnapshot.conf _
```

```
#############################################
#     BACKUP LEVELS / INTERVALS             #
# Must be unique and in ascending order     #
# e.g. alpha, beta, gamma, etc.             #
#############################################

retain  daily   7
retain  weekly  4
retain  mounthly        12
#retain delta   3
```

```
############################
# SNAPSHOT ROOT DIRECTORY #
############################

# All snapshots will be stored under this root directory.
#
snapshot_root   /media/nas/sauvegarde
```

```
# RSYNC.SAMBA.ORG
#backup rsync://rsync.samba.org/rsyncftp/       rsync.samba.org/rsyncftp/
backup save@BSD-Test:/home/save/dumpsql         BSD-Test_
```

```
# rsync must be enabled for anything to work. This is the only command that
# must be enabled.
#
cmd_rsync       /usr/bin/rsync
```

```
root@deb12:~# sudo rsnapshot daily
root@deb12:~#
```

Toujour sur machine rsnapshot

```
root@deb12:~# sudo rsnapshot daily
root@deb12:~# ls /media/nas/sauvegarde/
daily.0  daily.1
root@deb12:~# _
```

Planifier rsnapshot avec cron

```
# m h  dom mon dow   command
* * * * * /usr/bin/rsnapshot daily
```

```
root@deb12:~# sudo /usr/bin/rsnapshot daily
root@deb12:~# _
```

```
root@deb12:~# ls /media/nas/sauvegarde/
daily.0  daily.1  daily.2  daily.3  daily.4
root@deb12:~#
```

Ça veut dire que **rsnapshot a bien exécuté 7 sauvegardes quotidiennes**, en rotation automatique

Chaque dossier correspond à un jour précédent (daily.0 = aujourd'hui, daily.1 = hier, etc.)

```
root@deb12:~# ls /media/nas/sauvegarde/daily.5/BSD-Test/home/save/dumpsql/test_sauvegarde_2025-03-24
.sql
/media/nas/sauvegarde/daily.5/BSD-Test/home/save/dumpsql/test_sauvegarde_2025-03-24.sql
root@deb12:~#
```

c'est **parfaitement en place** :

Ton fichier test_sauvegarde_2025-03-24.sql est bien sauvegardé
Il est versionné (daily.0 à daily.6)
Tout est fonctionnel à 100% 🔐📦

Ça veut dire :

- save@bsd = le user + alias SSH (qu'on a mis dans ~/.ssh/config)
- :/home/save/dumpsql/ = le chemin distant à sauvegarder
- bsd/ = le nom du dossier dans /media/nas/sauvegarde/

Ton rsnapshot daily fonctionne à 100% ✅

Resumé configue

🗒️ 🔐 CONFIG EXPLIQUÉE SIMPLEMENT :

🖥️ Machine bsd (10.1.10.14)

📍 **Ce qu'elle fait** :
→ Elle contient la base de données (MySQL/MariaDB)

🔁 **Elle exécute chaque jour un script qui :**

- Fait un mysqldump de la base test_sauvegarde

- Enregistre le fichier .sql dans : /home/save/dumpsql/

⏰ **C'est déclenché automatiquement par cron tous les jours à 3h**

🖥️ Machine rsnap (10.1.10.13)

📍 **Ce qu'elle fait** :
→ Elle récupère automatiquement les fichiers .sql de bsd

📦 **Elle utilise rsnapshot pour :**

- Se connecter à bsd via SSH
- Copier le dossier /home/save/dumpsql/
- Le stocker dans /media/nas/sauvegarde/ avec des dossiers tournants :

📂 OÙ SONT LES BACKUPS ?

Sur la machine rsnap :

```
root@deb12:~# ls /media/nas/sauvegarde/daily.5/BSD-Test/home/save/dumpsql/
test_sauvegarde_2025-03-24.sql
root@deb12:~# _
```

💥 EN CAS DE PROBLÈME (ex: base supprimée)

1. Rends-toi sur rsnap
2. Choisis une version du dump :

```
root@deb12:~# cd /media/nas/sauvegarde/daily.5/BSD-Test/home/save/dumpsql/
root@deb12:/media/nas/sauvegarde/daily.5/BSD-Test/home/save/dumpsql#
```

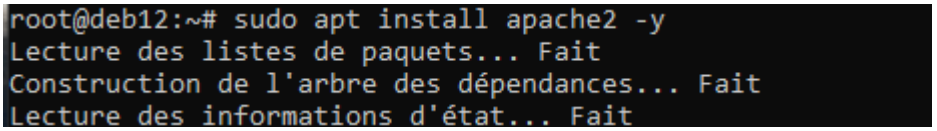
Copie le fichier .sql vers bsd ou un autre serveur

Restaure la base avec :

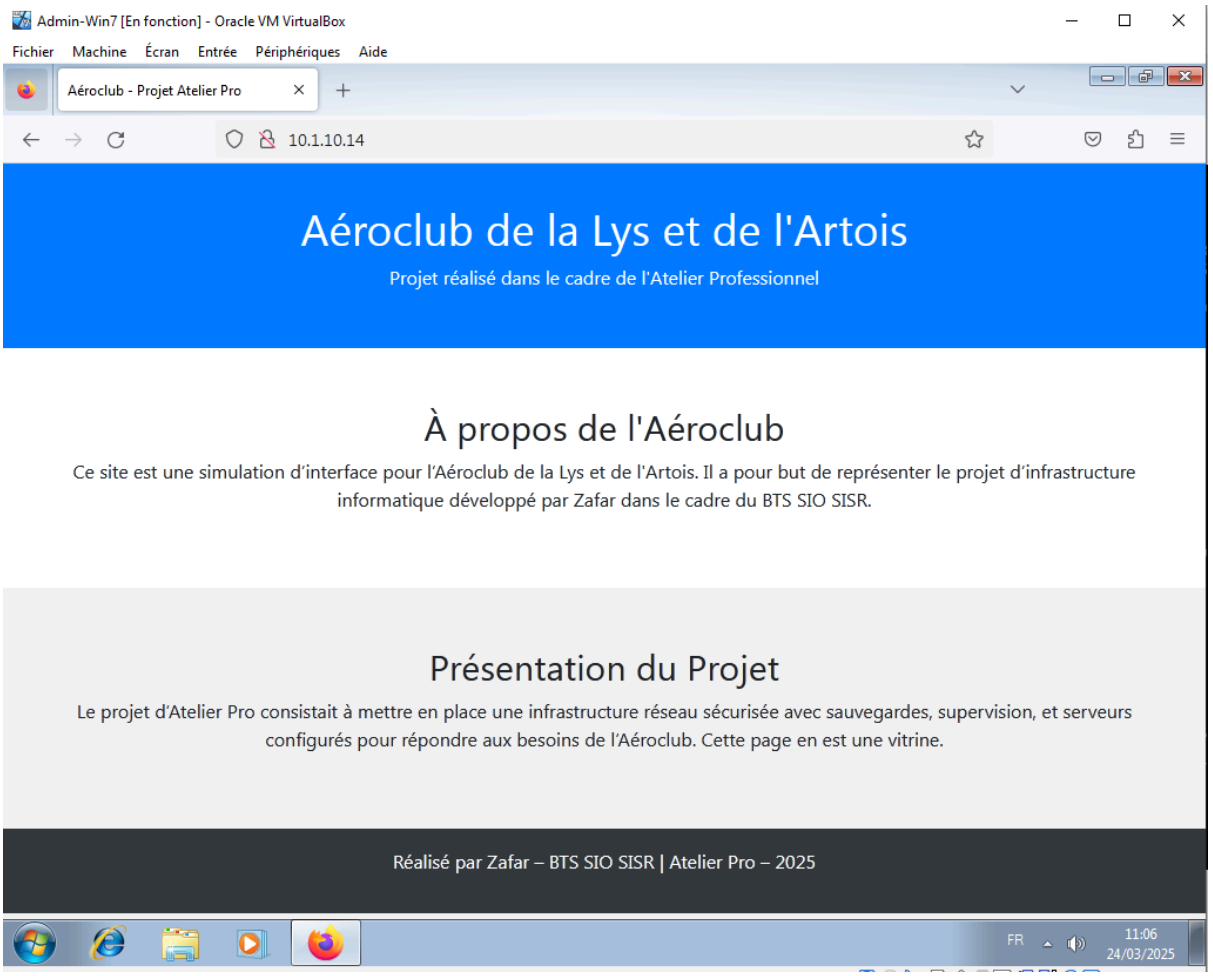
backup des fichiers web (Apache)

Attention au droiot www data

On va utiliser BSD-Test comme **serveur web:**



```
root@deb12:/var/www/html# nano index.html
root@deb12:/var/www/html# sudo chown www-data:www-data /var/www/html/index.html
root@deb12:/var/www/html# sudo chmod 644 /var/www/html/index.html
root@deb12:/var/www/html# sudo systemctl restart apache2
root@deb12:/var/www/html#
```



Plan propre :

- 📦 **Sauvegardes SQL : `/media/nas/sauvegarde/`**
- 🌐 **Sauvegardes site web : `/media/nas/sauvegarde-web/`**

Objectif :

Depuis rsnap, rsnapshot va :

- **Se connecter à bsd (alias : BSD-Test)**
- **Récupérer tous les fichiers du site web : /var/www/html/**
- **Les stocker dans : /media/nas/sauvegarde-web/daily.X/**

si tu veux deux dossiers de destination différents comme :

- **/media/nas/sauvegarde/ → pour SQL**
- **/media/nas/sauvegarde-web/ → pour le site web**

Créer le dossier dédié pour le site web :

```
root@deb12:/media/nas/sauvegarde# mkdir -p /media/nas/sauvegarde-web
root@deb12:/media/nas/sauvegarde# cd ..
root@deb12:/media/nas# ls
sauvegarde  sauvegarde-web
root@deb12:/media/nas#
```

Faire une copie de la conf rsnapshot :

```
root@deb12:~# cp /etc/rsnapshot.conf /etc/rsnapshot-web.conf
root@deb12:~# nano /etc/rsnapshot-web.conf _
```

Modifier ce nouveau fichier

```
# All snapshots will be stored under this root directory.
#
snapshot_root   /media/nas/sauvegarde-web_
```

```
#backup rsync://rsync.samba.org/rsyncftp/       rsync.samba.org/rsyncftp/
backup  save@BSD-Test:/var/www/html     site/
```

Résultat : les fichiers du site Apache seront copiés dans :

/media/nas/sauvegarde-web/daily.0/site/index.html

Test de cette nouvelle confet Lance une sauvegarde test

```
root@deb12:~# rsnapshot -c /etc/rsnapshot-web.conf configtest
Syntax OK
root@deb12:~# rsnapshot -c /etc/rsnapshot-web.conf daily
root@deb12:~#
```

Le fichier index.html de ton site Apache est bien sauvegardé ici :

Ajouter la tâche cron pour les fichiers web

```
root@deb12:~# rsnapshot -c /etc/rsnapshot-web.conf configtest
Syntax OK
root@deb12:~# rsnapshot -c /etc/rsnapshot-web.conf daily
root@deb12:~# ls /media/nas/sauvegarde-web/daily.0/site/
var
root@deb12:~# ls /media/nas/sauvegarde-web/daily.0/site/var/www/html/index.html
/media/nas/sauvegarde-web/daily.0/site/var/www/html/index.html
```

Où ?

Sur la machine rsnap (la machine de sauvegarde)

```
# m h  dom mon dow   command
* * * * * /usr/bin/rsnapshot daily
* * * * * /usr/bin/rsnapshot -c /etc/rsnapshot-web.conf daily
```

```
root@deb12:/media/nas/sauvegarde-web# tree -L 6
.
├── daily.0
│   └── site
│       └── var
│           └── www
│               └── html
│                   └── index.html
├── daily.1
│   └── site
│       └── var
│           └── www
│               └── html
│                   └── index.html
└── daily.2
    └── site
        └── var
            └── www
                └── html
                    └── index.html
```

```
root@deb12:/media/nas/sauvegarde# tree -L 2
.
├── daily.0
│   ├── BSD-Test
│   └── BSD-Test-site
├── daily.1
│   ├── BSD-Test
│   └── BSD-Test-site
├── daily.2
│   ├── BSD-Test
│   └── BSD-Test-site
├── daily.3
│   ├── BSD-Test
│   └── BSD-Test-site
├── daily.4
│   ├── BSD-Test
│   └── BSD-Test-site
├── daily.5
│   ├── BSD-Test
```

🗒️ Mémo Sauvegarde – Projet Atelier Pro – Amin (Zafar)

🔹 Structure de tes machines

Rôle	Nom machine	IP	Fonction principale
Source (BDD + Web)	BSD-Test	10.1.10.14	Contient : base MariaDB + site Apache
Backup	rsnap	10.1.10.13	Stocke les dumps SQL + fichiers web via rsnapshot

---

📁 Organisation des sauvegardes

Type	Dossier de sauvegarde sur rsnap	Fichier de conf utilisé
📦 Bases SQL	/media/nas/sauvegarde/	/etc/rsnapshot.conf
🌐 Site Web	/media/nas/sauvegarde-web/	/etc/rsnapshot-web.conf

---

🔐 Connexions SSH

- Utilisateur dédié : save
- Connexion sans mot de passe entre rsnap ➜ BSD-Test

Contenu des fichiers de conf rsnapshot

/etc/rsnapshot.conf (sauvegarde SQL)

```
snapshot_root   /media/nas/sauvegarde/
retain  daily   7
retain  weekly  4
retain  monthly 12

backup  save@BSD-Test:/home/save/dumpsql/   bsd-sql/
```

/etc/rsnapshot-web.conf (sauvegarde site web)

```
snapshot_root   /media/nas/sauvegarde-web/
retain  daily   7
retain  weekly  4
retain  monthly 12

backup  save@BSD-Test:/var/www/html/   site/
```

📂 Emplacements importants

Contenu	Emplacement
Dump SQL	/home/save/dumpsql/ sur BSD-Test
Fichiers web Apache	/var/www/html/ sur BSD-Test

Backup SQL (rsnap)	**/media/nas/sauvegarde/daily.0/bsd-sql/**
Backup site web (rsnap)	**/media/nas/sauvegarde-web/daily.0/site/var/www/html/**

PLAN D'ACTION : Sauvegarde vers le Cloud

Ton vrai objectif :

Tu veux que le serveur distant (WAN) – donc la machine 192.168.62.133 – vienne se connecter à rsnap (en LAN 10.x.x.x) pour venir récupérer les fichiers de sauvegarde, et les stocker chez lui.

👉En gros : c'est le serveur distant qui "vient chercher" les backups (pas rsnap qui envoie)

Rsnap → 10.1.10.13 (LAN) → Stocke les sauvegardes

Bsd → 10.1.10.14 (LAN) → Source des données (web + SQL)

Serveur-backup-wan → 192.168.62.133 VIENT → chercher les fichiers depuis rsnap via SSH

🔒 Ce qu'on doit faire :

Ouvrir le port **22** du serveur rsnap sur son routeur (NAT)
Depuis 192.168.62.133, se connecter en **SSH** à rsnap (via l'IP publique du routeur)

Faire un rsync vers /media/nas/sauvegarde/ et /media/nas/sauvegarde-web/

Automatiser avec cron ou un script

1.🔧 Sur le **routeur côté rsnap (réseau 10.x)**

Fais une redirection NAT : ip nat inside source static tcp 10.1.10.13 22 interface GigabitEthernet 8 2222

2. 🗝️ Sur la machine **WAN (192.168.62.133)**

✅ ÉTAPE 1 : Installer les outils sur le serveur distant

```
root@P232502-Backup:~# sudo apt install rsync openssh-client -y
Reading package lists... Done
Building dependency tree... Done
Reading st
The follow
```

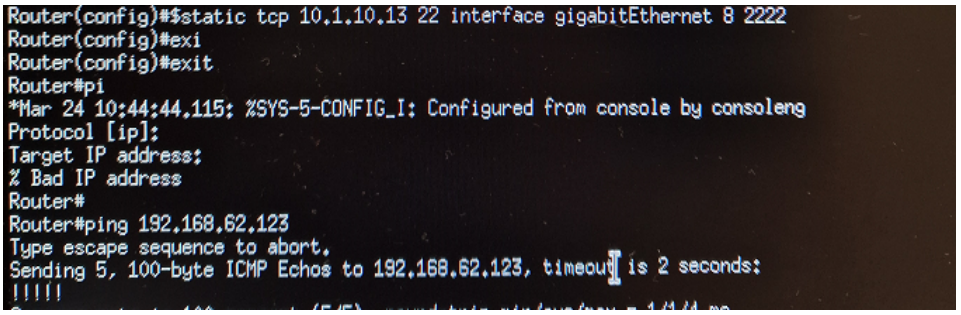
✅ ÉTAPE 2 : Générer une clé SSH

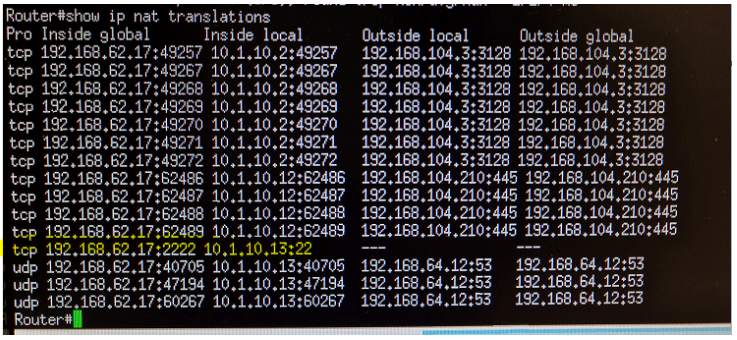
```
root@P232502-Backup:~# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:JVLR3iwdkt3VbuibFNycShE02EzieOAMjL5yuvUkct0 root@P232502-Backup
The key's randomart image is:
+---[RSA 4096]----+
|      o.o+ +**..o|
|     . o+ B.++o .|
|    . . .=.B o.=.|
|     . . o+ +.+.=|
|      . S  ....o |
|   . o. .    .o  |
|   .++ o E   . o |
|   .+ +       o  |
|   ..  .         |
+----[SHA256]-----+
root@P232502-Backup:~#
```

✅ CHECKLIST DE TA CONF ROUTEUR

Tu vas envoyer la clé sur l'IP publique de ton routeur (qui redirige vers rsnap) :

Redirection NAT :

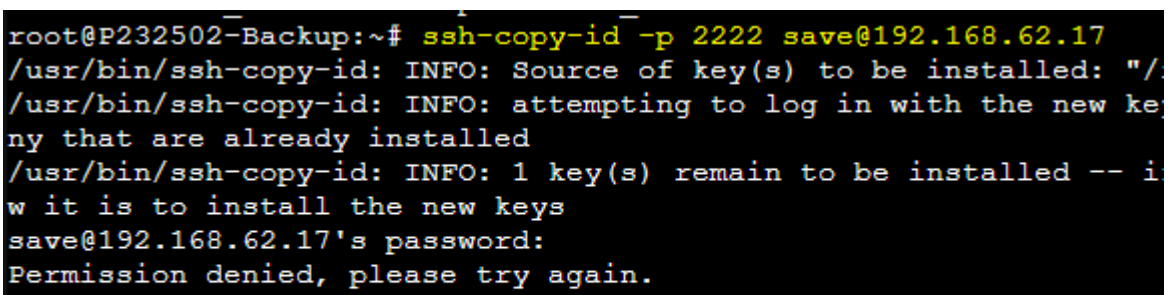




NAT redirige le port 2222 → rsnap

Connexion SSH depuis WAN

📤 Ensuite, **tu dois envoyer la clé publique depuis Backup(Proxmox) vers rsnap** :



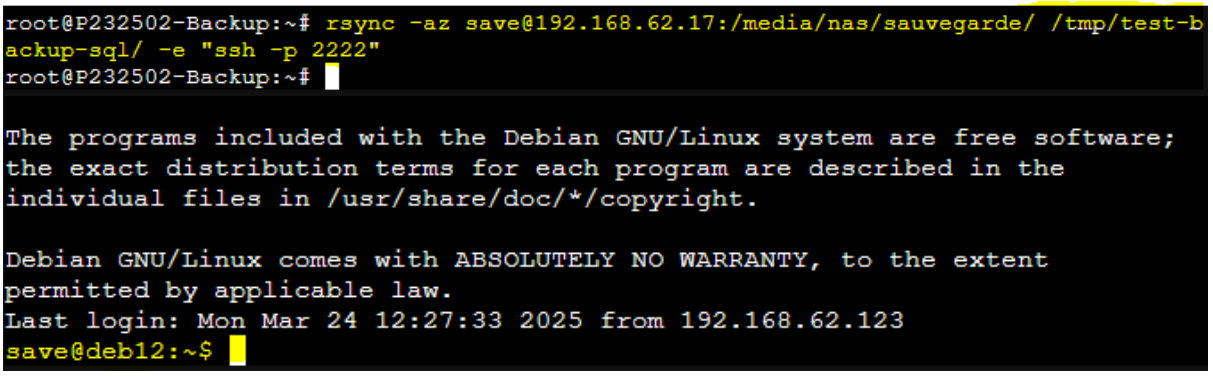
✅ Résultat final :

🔁 Qui se connecte à qui ?

Machine WAN Baclup (192.168.62.123) Se connecte via ssh -p 2222 Pour accéder à rsnap (10.1.10.13) et récupérer les backups

Machine rsnap (LAN) Reçoit la connexion Elle doit **reconnaître la clé** du client WAN

Teste juste **l'accès à la sauvegarde SQL** (par exemple) :



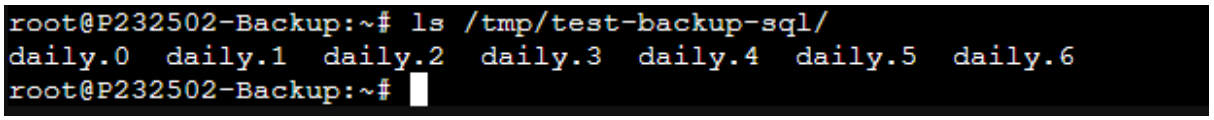
Le rsync s'est exécuté **sans erreur** 💯 et t'as eu **aucun message** → ce qui veut dire :

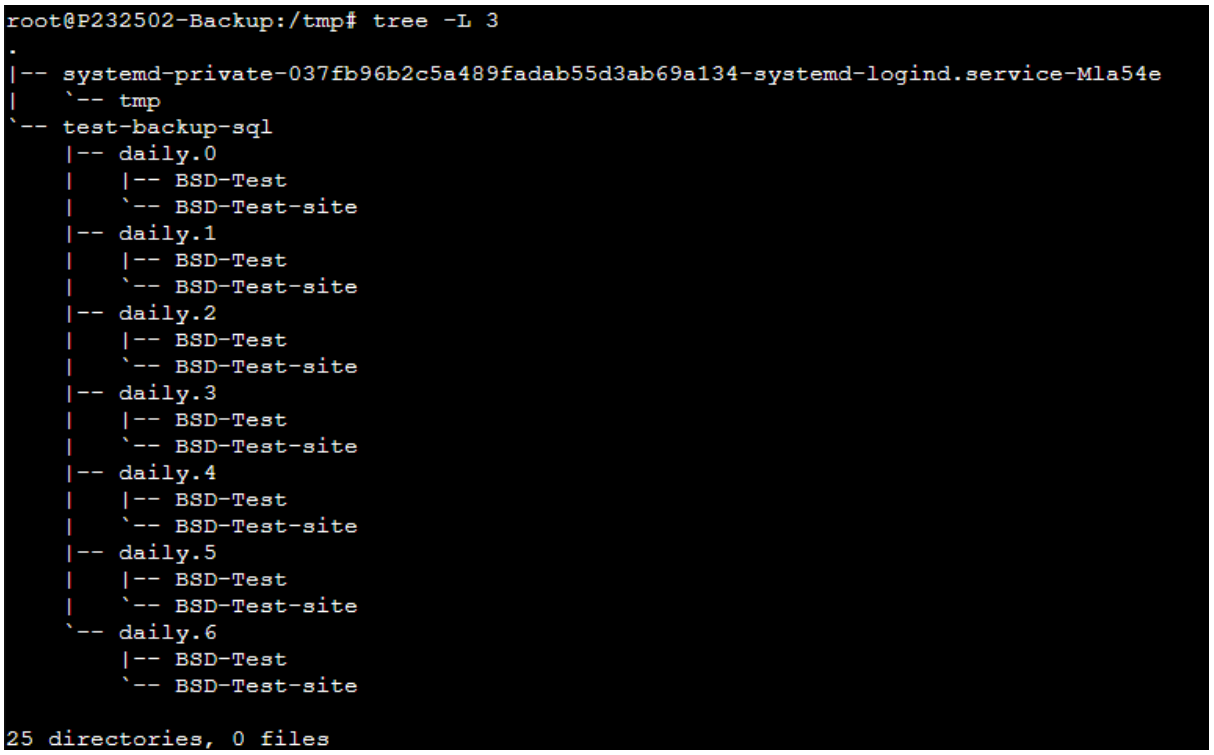
✅Il a réussi à se connecter ET transférer les fichiers
✅Et il n'y avait **rien de nouveau** à synchroniser (tout est déjà à jour)

🔍 Pour vérifier que les fichiers ont bien été récupérés sur serveur Backup :

Tu vois les daily.0, daily.1, ..., daily.6 = **tes sauvegardes SQL sont bien là** !

Ton rsync a **récupéré toutes les rotations de rsnapshot**, exactement comme prévu.





✅ Maintenant : on passe en PROD 💼

On va créer le **script final + automatisation avec cron** sur ta machine WAN.

```
root@P232502-Backup:~# mkdir -p /home/backup
```

```
root@P232502-Backup:~# nano /home/backup/pull_backup.sh
```

pve-elev1 - Proxmox Console — Mozilla Firefox

```
root@P232502-Backup:/home/backup# /home/backup/pull_backup.sh
root@P232502-Backup:/home/backup# ls
amin-backups  pull_backup.sh
root@P232502-Backup:/home/backup#

mkdir -p /home/backup/amin-backups/web

#  Synchronisation SQL
rsync -az save@192.168.62.17:/media/nas/sauvegarde/ /home/backup/amin-backups/sql/ -e "ssh -p 2222"

#  Synchronisation site web
rsync -az save@192.168.62.17:/media/nas/sauvegarde-web/ /home/backup/amin-backups/web/ -e "ssh -p 2222"
```

```
root@P232502-Backup:~# chmod +x /home/backup/pull_backup.sh
root@P232502-Backup:~# /home/backup/pull_backup.sh
```

```
root@P232502-Backup:/home/backup# chmod +x /home/backup/pull_backup.sh
root@P232502-Backup:/home/backup# ls /home/backup/amin-backups/sql/daily.0/
BSD-Test  BSD-Test-site
root@P232502-Backup:/home/backup# ls /home/backup/amin-backups/web/daily.0/
site
root@P232502-Backup:/home/backup#
```

⏰ Active l'automatisation avec cron (si pas encore fait)

Automatiser le script

➡️ Tous les jours à 5h du mat, le serveur WAN viendra chercher les sauvegardes automatiquement
📥📥

```
# m h  dom mon dow   command
0 5 * * * /home/backup/pull_backup.sh



crontab: installing new crontab
root@P232502-Backup:/home/backup# crontab -e
```