



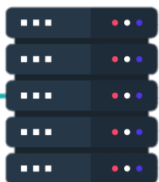
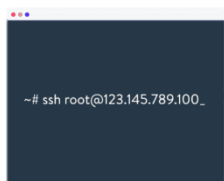
Backup WordPress

:~\$ SSH/CLI # ligne de commande



Le SSH (Secure Shell)

- ❑ Le **SSH** est un protocole de **connexion sécurisée**
- ❑ Permet d'établir une connexion entre un **serveur** et le **client**
- ❑ Le shell (coque) est le logiciel (**terminal** ou **console**) qui permet l'accès à distance
- ❑ Le client peut interagir avec le serveur via des **lignes de commandes (CLI)**



Secure Shell

===

Terminal sécurisé

Prérequis

- ❑ Un **server** **SSH** (ex.: OpenSSH)
- ❑ Un **client** **SSH** (Terminal sur Mac, CMD sur Windows 10)
- ❑ Installer [WP-CLI](#) sur le serveur
- ❑ Installer les paquets **ZIP** et **CRON** sur le serveur
- ❑ Installer le script BASH [Dropbox Uploader](#)



Objectifs



Sauvegarder

- Exporter la base de données SQL [1]
- Générer une archive de WordPress [2]
- Téléverser l'archive sur Dropbox [3]



Restaurer

- [3] Importer la base de données SQL
- [2] Décompresser l'archive de WordPress
- [1] Télécharger l'archive depuis Dropbox





Sauvegarder

```
# On se connecte au serveur en SSH
ssh utilisateur@serveur

# On se positionne dans le dossier de WP
cd /var/www/html/

# On exporte la base de données (WP-CLI)
wp db export database.sql

# On archive tous les fichiers et dossiers de WP
zip -r backup.zip *.[^.]*

# On se repositionne dans le dossier utilisateur
cd

# On téléverse le zip sur Dropbox
./dropbox_uploader.sh upload /var/www/html/backup.zip /

# On supprime les fichiers précédemment générés
rm /var/www/html/{database.sql,backup.zip}
```



Restaurer

```
# On se connecte au serveur en SSH
ssh utilisateur@serveur

# On crée un dossier temporaire
mkdir tmp
WP_DIR=/var/www/html/

# On télécharge la sauvegarde dans le dossier temporaire
./dropbox_uploader.sh download backup.zip ./tmp

# On supprime le contenu dans le dossier de WP
rm -rf $WP_DIR{*,[^.]*}

# On extrait le .zip dans le dossier de WP
unzip ./tmp/backup.zip -d $WP_DIR
rm -rf ./tmp

# On importe la base de données (WP-CLI)
cd $WP_DIR && wp db import database.sql
rm database.sql
```



Automatiser - Script Bash

- ❑ Eviter de taper de nombreuses lignes de commandes
- ❑ Ajouter de la **logique** et des vérifications
- ❑ Utilisation de **scripts Bash** (.sh)

```
# Sauvegarder - Exécuter le fichier wp-backup.sh  
./wp-backup.sh  
  
# Restaurer - Exécuter le fichier wp-restore.sh  
./wp-restore.sh
```



Automatiser - Cron

- ❑ Utiliser **CRON** et **CRONTAB** pour planifier des sauvegardes récurrentes
- ❑ Générateur de crontab: http://www.desmoulins.fr/index.php?pg=informatique!unix!crontab_generator

```
# Exemples de crons (* * * * * => mm hh jj MMM JJJ)
```

```
1 * * * * root cmd # Exécutée toutes les heures passées d'une minute
30 8 * * * root cmd # Exécutée tous les jours à 08h30
00 4 * * 0 root cmd # Exécutée tous les dimanches à 4h00
40 4 1 * * root cmd # Exécutée tous les 1er du mois à 4h40
01 * 19 07 * root cmd # Exécutée toutes les heures passées d'une minute tous les 19/07
```

```
# Exécuter le fichier wp-backup.sh toutes les 24H (à 00h00) pour sauvegarder
0 0 * * * ./wp-backup.sh
```




Amélioration

- ☐ Utiliser un meilleur format de compression: **GZIP** (.gz)
- ☐ Utiliser un **serveur dédié** aux sauvegardes (ex.: [Digital Ocean Spaces](#))
- ☐ Utiliser **rsync** pour faire des sauvegardes différentielles
- ☐ Protéger les sauvegardes par **chiffrement** (ex.: **GPG**, **gnupg**)



Des questions?

:~\$ merci de votre attention

