

# Sauvegarder et restaurer WordPress

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Collecter les informations requises](#)

[Télécharger et mettre à jour des scripts](#)

[Mettre à jour le profil WordPress](#)

## Introduction

Ce document décrit comment utiliser CloudCenter pour migrer une application pour sauvegarder et restaurer le contenu dans un groupement Amazon S3.

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CloudCenter
- Amazon S3

### Components Used

Les informations de ce document sont basées sur CloudCenter v4.8.1.1.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Informations générales

Ce document suppose que CloudCenter est déjà installé et fonctionne correctement. De plus, WordPress a déjà été modélisé et confirmé pour fonctionner correctement. Ce processus n'a été testé qu'avec l'utilisation d'un groupement S3 comme référentiel et migre d'un déploiement à un autre entre les clouds publics, bien qu'il doive fonctionner entre les clouds publics et privés tant

que la connectivité avec le groupement S3 est confirmée dans le nuage de destination. Cette procédure ne doit être effectuée que pour une preuve de concept. Il n'utilise pas les meilleures pratiques car les clés secrètes sont codées en dur dans le script pour faciliter leur utilisation. Les scripts spécifiés ont été écrits pour CentOS avec l'utilisation d'un serveur web Apache.

## Collecter les informations requises

CloudCenter utilise quelques scripts pour pouvoir sauvegarder et restaurer les données sur la machine virtuelle correcte, afin de remplir les scripts, certaines informations doivent être collectées au préalable.

- Nom du groupement Amazon S3
- Chemin dans le compartiment où les données de sauvegarde sont stockées
- Clé Amazon S3
- Clé secrète Amazon S3

**Note:** Le compartiment S3 doit également être ajouté en tant que référentiel dans CloudCenter.

## Télécharger et mettre à jour des scripts

- Quatre scripts sont nécessaires pour gérer la migration de WordPress. Deux scripts de sauvegarde, un pour la base de données et un autre pour le serveur Web, et deux scripts de restauration.
- Téléchargez BackupRestore.zip et extrayez-le de **RestoreServer.sh**, **RestoreDB.sh**, **BackupDB.sh** et **BackupServer.sh**.
- À partir de chaque élément, mettez à jour le **groupement**, **chemin**, **clé S3**, **secret S3**.
- Le script de sauvegarde pour le serveur Web ferme le répertoire **/var/www/** dans un fichier appelé **server.zip** qui est stocké dans le répertoire **/tmp**. Il télécharge ensuite **server.zip** dans le groupement S3 avec les informations d'identification spécifiées.
- Le script de restauration pour le serveur Web, télécharge le fichier **server.zip** et le décompresse dans le répertoire **/var/www/**. Aucun de ces scripts ne vérifie les erreurs, ni ne vérifie le système d'exploitation installé, cela peut causer des problèmes si WordPress a été installé sur un autre système d'exploitation ou avec un autre serveur Web, autre qu'Apache.
- Le script de sauvegarde de la base de données effectue un vidage de la base de données (DB) et le zip avant de le télécharger dans le compartiment S3.
- Le script de restauration de la base de données crée la base de données, puis utilise le vidage de la base de données qu'il a téléchargé à partir du compartiment S3 pour recréer la base de données.

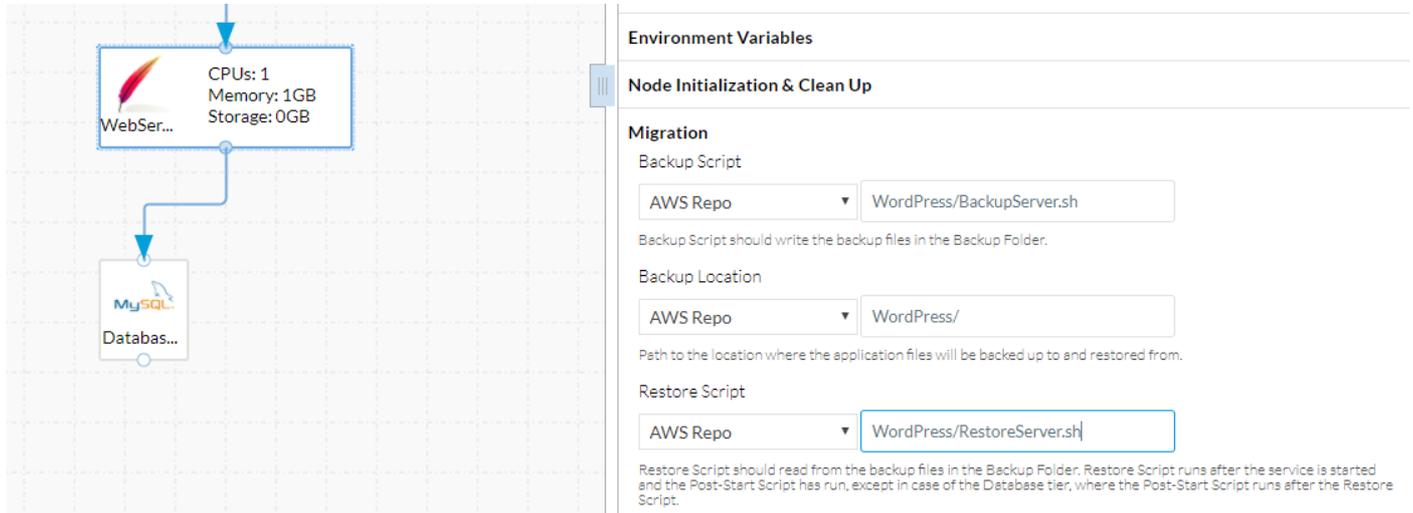
**Note:** Ces scripts ont la clé et le secret S3 stockés en texte brut, ce qui n'est pas recommandé et ne doit être utilisé que comme preuve de concept, ou au moment du test initial.

Une fois tous les champs mis à jour, téléchargez les scripts dans un référentiel CloudCenter afin qu'ils puissent être référencés dans un profil d'application.

# Mettre à jour le profil WordPress

Quelques mises à jour du profil sont nécessaires pour utiliser ces nouveaux scripts.

Sous WebServer, sélectionnez Migration et ajoutez un chemin d'accès à BackupServer.sh dans Backup Script, référez également l'emplacement de sauvegarde dans Backup Location, et enfin ajoutez le chemin d'accès à **RestoreServer.sh** dans **Restore Script** comme indiqué dans l'image.



The screenshot displays the configuration for the Migration section of a WordPress profile. The left pane shows a diagram with a WebServer tier (1 CPU, 1GB Memory, 0GB Storage) and a MySQL Database tier. The right pane shows the configuration for the Migration section, including Backup Script, Backup Location, and Restore Script.

**Environment Variables**

**Node Initialization & Clean Up**

**Migration**

Backup Script  
AWS Repo: WordPress/BackupServer.sh

Backup Script should write the backup files in the Backup Folder.

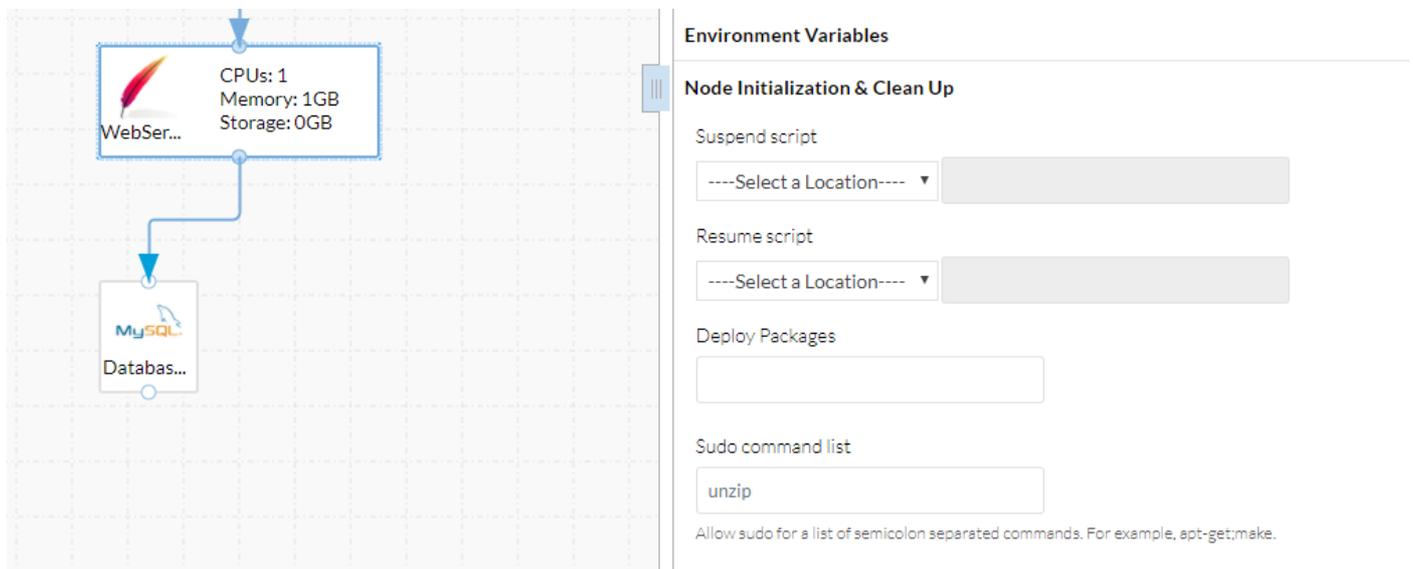
Backup Location  
AWS Repo: WordPress/

Path to the location where the application files will be backed up to and restored from.

Restore Script  
AWS Repo: WordPress/RestoreServer.sh

Restore Script should read from the backup files in the Backup Folder. Restore Script runs after the service is started and the Post-Start Script has run, except in case of the Database tier, where the Post-Start Script runs after the Restore Script.

RestoreServer.sh a besoin d'une autorisation pour pouvoir décompresser les fichiers dans **/var/www/ce** que cliquer n'a pas l'autorisation de faire. Sous **Node Initialization & Clean Up** ajoutez **unzip** à la liste de commandes de Sudo. Cela donne au script l'autorisation d'exécuter **unzip** en tant que root comme indiqué dans l'image.



The screenshot displays the configuration for the Node Initialization & Clean Up section of a WordPress profile. The left pane shows a diagram with a WebServer tier (1 CPU, 1GB Memory, 0GB Storage) and a MySQL Database tier. The right pane shows the configuration for the Node Initialization & Clean Up section, including Suspend script, Resume script, Deploy Packages, and Sudo command list.

**Environment Variables**

**Node Initialization & Clean Up**

Suspend script  
----Select a Location----

Resume script  
----Select a Location----

Deploy Packages

Sudo command list  
unzip

Allow sudo for a list of semicolon separated commands. For example, apt-get;make.

Le niveau de base de données nécessite des modifications similaires à celles du serveur Web, à savoir le **script de sauvegarde**, l'**emplacement de sauvegarde** et le **script de restauration** comme indiqué dans l'image.

The image shows a configuration interface for application migration. On the left, a diagram illustrates a migration path from a 'WebSer...' node to a 'MySQL Databas...' node. The MySQL node is configured with 1 CPU, 1GB of memory, and 0GB of storage. On the right, the 'Migration' settings are detailed:

- Script executed before the backup is taken:** (Empty field)
- Backup Script:** Set to 'WordPress/BackupDB.sh' (AWS Repo dropdown)
- Backup Script should write the backup files in the Backup Folder:** (Empty field)
- Backup Location:** Set to 'WordPress' (AWS Repo dropdown)
- Path to the location where the application files will be backed up to and restored from:** (Empty field)
- Restore Script:** Set to 'WordPress/RestoreDB.sh' (AWS Repo dropdown)
- Restore Script should read from the backup files in the Backup Folder. Restore Script runs after the service is started and the Post-Start Script has run, except in case of the Database tier, where the Post-Start Script runs after the Restore Script.** (Empty field)
- Post-Migrate Script:** (Empty field)

Une fois ces modifications effectuées, il vous suffit **d'enregistrer** le profil d'application.

Désormais, un nouveau déploiement doit pouvoir être migré d'un noeud à un autre.