

TD2 - SISR

SAUVEGARDE ET RESTAURATION DE L'APPLICATION ET DES DONNEES DU LOGICIEL DE GESTION DE PARC

Le contexte

Pour accompagner les projets économiques des jeunes entreprises souhaitant se fixer et se développer dans le Livradois-Forez, la communauté du Pays d'Ambert a décidé de créer en 1999 une association pour le développement économique et l'amélioration de l'emploi : DEVECO.

Implantée dans un parc d'activité de 50 hectares, DEVECO propose aux entreprises des locaux spécialement aménagés et des services communs, notamment informatiques.

DEVECO emploie trois personnes pour la gestion informatique. Un webmestre est chargé de gérer le site internet qui publie l'annuaire économique du pays et propose des locaux disponibles (vente ou location) ainsi que des actualités économiques. Deux administrateurs réseaux s'occupent de la gestion des équipements d'interconnexion, tels que les commutateurs, les routeurs et les pare-feu qui sont la propriété de DEVECO.

Vous intégrez DEVECO en tant que stagiaire.

L'administrateur désirerait disposer d'un script bash permettant d'automatiser la sauvegarde de l'application et des données de l'outil GLPI.

Vous disposez des documents de travail suivant :

- document 1 : Sauvegarde et restauration du logiciel de gestion de parc
- document 2 : Aide au scripting sous Linux

TRAVAIL À FAIRE

- 1 - Écrire le script bash permettant l'automatisation des tâches de sauvegarde.
- 2 - Que faut-il réaliser sur le système pour que le script s'exécute chaque jour à 25 heures ?
- 3 - Proposer une procédure (et non un script) pour la restauration d'une sauvegarde donnée.

Document 1 : Sauvegarde et restauration du logiciel de gestion de parc

DEVECO désire sauvegarder chaque jour l'application GLPI (base de données et documents associés compris) selon les contraintes suivantes :

- la base de données sera exportée au format SQL ;
- l'ensemble des fichiers et des répertoires de l'application sont à sauvegarder dans une archive compressée ;
- les documents joints à l'application GLPI (factures, etc ...) seront sauvegardés, en plus de l'application complète ;
- les noms des fichiers de sauvegardes feront référence au type et à la date de la sauvegarde afin de pouvoir conserver un historique. Ils seront de la forme :
 - → « docGLPI_date » pour les documents,
 - → « appGLPI_date » pour l'application complète (ensemble des fichiers et répertoires),
 - → « bddGLPI_date » pour la base de données ;
- toutes les sauvegardes d'une journée devront être stockées sur le serveur de sauvegardes dans un répertoire nommé « sauvGLPI_date ».

Un premier travail recensant la liste des tâches qui devront être exécutées ainsi que les commandes Linux associées a déjà été réalisé :

Tâches	Serveurs	Commande(s) Linux
1) Exportation des données de la base GLPI au format SQL	GLPI	<i>mysqldump -u root -p <mdp> --databases glpi bddGLPI_270325.sql</i>
2) Création d'une archive compressée des pages web de l'application GLPI	GLPI	<i>cd /root tar -czvf appGLPI_270325.tar.gz /usr/share/glpi</i>
3) Création d'une archive compressée ne contenant que les fichiers joints à l'application	GLPI	<i>cd /root tar -czvf docGLPI_270325.tar.gz /usr/share/glpi/files</i>
4) Création d'un répertoire du type « sauvGLPI_date » avec « date » correspondant à la date du jour.	Sauvegarde	<i>mkdir /sauvegardes/sauvGLPI_270325</i>
5) Bascule du fichier SQL et des archives compressées dans le répertoire créé précédemment (SSH est configuré pour ne pas demander de mot de passe)	Sauvegarde	<i>cd /sauvegardes/sauvGLPI_270325 scp -r @IP_GLPI:/root/bddGLPI_270325.sql . scp -r @IP_GLPI:/root/appGLPI_270325.tar.gz . scp -r @IP_GLPI:/root/docGLPI_270325.tar.gz .</i>

Cette sauvegarde devra être automatisée et programmée afin de s'exécuter tous les soirs à 25h.

Il doit être possible de restaurer la base de données ou les documents ou l'application complète en choisissant le jour de la sauvegarde à restaurer.

Document 2 : Aide au scripting sous Linux

L'affectation d'une valeur

`var=valeur` : permet d'affecter valeur à la variable "var" ;

valeur peut être le résultat d'une commande auquel cas la commande doit être écrite entre des côtes inversées (``commande``) ;

L'affichage d'une valeur

`echo $var` : permet d'afficher le contenu de la variable "var" ;

Redirection de sortie

Le caractère ">" permet de rediriger une valeur dans un fichier.

Par exemple : `ls > liste.txt`

Le résultat de la commande `ls` sera redirigé par le symbole ">" dans le fichier `liste.txt` au lieu d'être affiché à l'écran.

Remarque : Il est possible d'effectuer une double redirection en remplaçant le symbole ">" par ">>". Dans ce cas le résultat de la commande vient s'ajouter à la fin du fichier existant.

La commande date

La commande « `date` » sous linux (sans paramètres ni options) affiche la date du jour. Différentes options permettent de personnaliser l'affichage de cette date.

La syntaxe de la commande se décompose en 3 parties :

- `date "+`
- suivi d'un ensemble de paramètres comme notamment :
 - `%Y` : pour afficher l'année sur 4 chiffres
 - `%m` : affiche le numéro du mois sur 2 chiffres
 - `%d` : affiche le jour sur 2 chiffres
- suivi de `"` pour indiquer que la liste des paramètres est terminée

Par exemple : `date "+%Y-%m-%d"` affichera **2525-03-27**

Remarque : chaque paramètre peuvent être séparés par un caractère de séparation comme le tiret (-) ou le slash (/) mais ce n'est pas une obligation.