

Script de sauvegarde avec Borg

Pour faire mes sauvegardes j'utilise *BorgBackup*. Les données sont envoyées sur un serveur de sauvegarde.

Les sauvegardes ont lieu quatre fois par jour en lançant le script dans un *cronjob*. S'en suit la réception d'un e-mail à chaque sauvegarde.

Au passage, une fois par jour, le script vérifie les mises à jour disponibles.

```
#!/bin/sh

export BORG_PASSPHRASE='<BORG_PASSPHRASE>'

# some helpers and error handling:
printMessage() { printf "\n*** %s => %s ***\n\n" "$( date +"%A %d %B %T" )" "$*" >&2; }
trap 'echo $( date ) Backup interrupted >&2; exit 2' INT TERM

printMessage "Starting mysql backup"

mysqldump --defaults-file=<variables_file_path> --all-databases > <output_file_path>

printMessage "Starting backup"

/usr/bin/borg create \
    --stats \
    --filter=AME \
    --compression auto,zstd,10 \
    --exclude-caches \
    --exclude='/var/lib/{snapd,selinux}' \
    <backup_server_url>:repo::{now} \
    /var/{www,lib} \
    /root \
    /etc \
    /home
```

```
backup_exit=$?
```

```
printMessage "Pruning repository"
```

```
# Use the `prune` subcommand to maintain 7 daily, 2 weekly and 2 monthly
# archives of THIS machine. The '{hostname}-' prefix is very important to
# limit prune's operation to this machine's archives and not apply to
# other machines' archives also:
```

```
borg prune \
  --list \
  --show-rc \
  --keep-hourly 3 \
  --keep-daily 7 \
  --keep-weekly 2 \
  --keep-monthly 2 \
  --save-space \
  <backup_server_url>:repo
```

```
prune_exit=$?
```

```
# use highest exit code as global exit code
global_exit=$(( backup_exit > prune_exit ? backup_exit : prune_exit ))
```

```
if [ ${global_exit} -eq 0 ]; then
  printMessage "Backup and Prune finished successfully"
elif [ ${global_exit} -eq 1 ]; then
  printMessage "Backup and/or Prune finished with warnings"
else
  printMessage "Backup and/or Prune finished with errors"
fi
```

```
if [ $(date "+%H") -eq 6 ]; then
  printMessage "Check updates"
  dnf check-update
fi
```

```
exit ${global_exit}
```