Borgmatic

Borgmatic est un *wrapper* autour de Borg qui en simplifie infiniment l'utilisation.

Installation

Il nous faut d'abord Borg (utilisez la version des backports Debian si vous êtes encore en *stretch*) :

```
apt install borgbackup
```

D'habitude, je préfère utiliser les paquets Debian, mais la version pip de Borgmatic apporte une option en plus que j'apprécie particulièrement.

```
apt install python3-pip
pip3 install borgmatic
```

Configuration

C'est là que Borgmatic est pratique : il permet une configuration très simple de Borg.

Générez un fichier de configuration :

```
generate-borgmatic-config
```

Cela créera le fichier `/etc/borgmatic/config.yaml`. Si vous souhaitez que la configuration soit dans un autre fichier :

```
generate-borgmatic-config -d /etc/borgmatic/autre_config.yaml
```

La configuration générée (exemple) est très simple à comprendre et auto-documentée, je ne vais l'expliquer, je vais me contenter d'en mettre certains points en valeur

Le dépôt

On le verra plus bas, j'utilise un serveur distant pour faire les sauvegardes. Donc :

```
location:
    repositories:
        - borg@server:/var/lib/borg/depot/
```

La *passphrase*

Choisissez quelque chose de costaud et sauvegardez-là quelque part !

```
storage:
    encryption_passphrase: "foo-bar-baz"
```

Utilisation d'une clé SSH particulière

Je crée plus bas une clé SSH dédiée pour Borg pour faire les sauvegardes sur le serveur distant. Il faut donc que j'indique à Borgmatic que je souhaite utiliser cette clé.

```
storage:
    ssh_command: ssh -i /root/.ssh/id_borgmatic
```

Les hooks

Situés à la fin du fichier de configuration, il s'agit d'actions qui seront effectuées avant et après la sauvegarde, ou en cas de problème lors de l'exécution.

Personnellement, j'aime bien avoir la liste de mes sauvegardes après qu'une sauvegarde soit effectuée. Je vais donc écrire :

```
hooks:
    after_backup:
        - /usr/local/bin/borgmatic list
```

Comme il est conseillé d'exporter la clé du dépôt borg, je mets aussi ceci dans les `hooks` :

```
    before_backup:
        - borg key export borg@server:/var/lib/borg/depot/ /etc/ssl/private/borg.key
```

Rétention et vérification du dépôt et des archives

Je fais ça sur le serveur, j'y reviendrai plus loin.

Cron

On crée un petit cron (avec l'utilisateur `root`) pour sauvegarder régulièrement l'ordinateur :

```
35 0 * * * borgmatic create -c /etc/borgmatic/mon_pc.yaml --stats 2>&1
```

Si ce n'est pas un serveur allumé 24h/24, il est préférable de mettre ça dans `/etc/anacrontab` :

```
@daily  15      backup borgmatic create -c /etc/borgmatic/mon_pc.yaml --stats 2>&1
```

Cela lancera le backup tous les jours, 15 minutes après le démarrage du pc. Ou plus tard.

Attention, sur Debian, les tâches anacron ne sont lancées que si vous êtes sur secteur ! Pour changer ça, reportez-vous au fichier `/usr/share/doc/anacron/README.Debian`.

Préparation du serveur du dépôt distant

Je préfère faire mes sauvegardes sur un disque distant : si mon disque lâche, j'aurai toujours mes sauvegardes.

Pour ce faire, je crée une clé dédiée sur l'ordinateur à sauvegarder, sans mot de passe vu que borgmatic va tourner automatiquement :

```
ssh-keygen -o -a 100 -t ed25519 -f /root/.ssh/id_borgmatic -N ''
```

Sur le serveur de sauvegarde, j'installe Borg et Borgmatic comme précédemment puis je crée un utilisateur `borg` :

```
adduser --system --home /var/lib/borg --shell /bin/bash --disabled-password borg
mkdir /var/lib/borg/.ssh
chmod 700 /var/lib/borg/.ssh
```

```
touch /var/lib/borg/.ssh/authorized_keys
chmod 600 /var/lib/borg/.ssh/authorized_keys
chown -R borg: /var/lib/borg/.ssh
```

Et je mets la clé publique créée précédemment dans `/var/lib/borg/.ssh/authorized_keys`, avec quelques restrictions :

```
command="/usr/bin/borg --umask=077 --info serve --append-only --restrict-to-repository
/var/lib/borg/becky2/",restrict ssh-ed25519
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX root@mon_pc
```

On génère une configuration :

```
generate-borgmatic-config -d /etc/borgmatic/mon_pc.yaml
```

Et c'est dans cette configuration qu'on va configurer les durées de rétention et les vérifications du dépôt et des archives.

Rétention

C'est très simple. Pour garder un backup par mois sur les 6 derniers mois, un par semaine sur les 8 dernières semaines et un par jour sur les 14 derniers jours, il suffit de :

```
retention:
    keep_daily: 14
    keep_weekly: 8
    keep_monthly: 6
```

Comme Borg déduplique comme un monstre, cela ne prendra pas énormément de place (sauf si vous sauvegardez des trucs qui changent tout le temps)

Vérification du dépôt et des archives

Cela permet de s'assurer que votre dépôt et vos archives sont utilisables et non corrompus. Je me contente de décommenter la configuration proposée :

```
consistency:
   checks:
       - repository
       - archives
   check_last: 3
```

J’ai laissé `check_last` à 3 car vérifier toutes les archives peut être fort long. Donc une vérification des 3 dernières devrait faire l’affaire, surtout si on vérifie quotidiennement.

Dépôt

Attention, comme on est là sur le serveur distant, il faut lui donner un dossier local !

```
location:
    repositories:
        - /var/lib/borg/depot/
```

Script de suppression d’anciennes sauvegardes / vérification des archives

Il y a des éléments dans ce script qui prendront tout leur sens plus tard.

NOTA BENE : ce script est tout à fait adaptable à un usage à Borg sans Borgmatic.

On prépare le terrain :

```
apt install libcpanel-json-xs-perl
mkdir -P /opt/borgmatic-stats/tmp/ /opt/borgmatic-stats/json/
cd /opt/borgmatic-stats/
wget https://framagit.org/snippets/3716/raw -O verify-archives.pl
chmod 700 -R /opt/borgmatic-stats/
chown borg: -R /opt/borgmatic-stats/
```

Puis on met ceci dans `/opt/borgmatic_prune_and_check.sh` :

```
#!/bin/bash
cat <<EOF
============================================
== mon_pc
============================================
EOF

/usr/local/bin/borgmatic list -c /etc/borgmatic/mon_pc.yaml --json > /opt/borgmatic-
stats/tmp/mon_pc.json.tmp
CONTINUE=0
```

```
if [[ -e /opt/borgmatic-stats/json/mon_pc.json ]]
then
    echo "Checking repository consistency."
    CONTINUE=$(/opt/borgmatic-stats/verify-archives.pl --old /opt/borgmatic-
stats/json/mon_pc.json --new /opt/borgmatic-stats/tmp/mon_pc.json.tmp)
    echo "Repository consistency checked."
else
    CONTINUE=1
fi

if [[ $CONTINUE == '1' ]]
then
    ## Check
    /usr/local/bin/borgmatic check -c /etc/borgmatic/mon_pc.yaml 2>&1 && \
    echo "Repository checked." && \
    ## Allow pruning
    borg config /var/lib/borg/depot/ append_only 0 && \
    ## Prune
    /usr/local/bin/borgmatic prune -c /etc/borgmatic/mon_pc.yaml --stats 2>&1 && \
    echo "Repository pruned."
    ## List
    echo "Borg archives of mon_pc:"
    /usr/local/bin/borgmatic list -c /etc/borgmatic/mon_pc.yaml 2>&1

    /usr/local/bin/borgmatic list -c /etc/borgmatic/mon_pc.yaml --json > /opt/borgmatic-
stats/json/mon_pc.json

    ## Disallow pruning
    borg config /var/lib/borg/depot/ append_only 1

    echo ''
else
    cat <<EOF

****************************************
* ALERT ON mon_pc! *
****************************************
All of the old archives can’t be retrieved from the repository (/var/lib/borg/depot/)!

Someone may have deleted an archive from mon_pc.
```

```
Pruning has been aborted. Please manually review the repository.

$CONTINUE
EOF
fi
```

On n’oublie pas de le rendre exécutable :

```
chmod +x /opt/borgmatic_prune_and_check.sh
```

Cron

Attention ! Il faut éditer la *crontab* de l’utilisateur `borg` :

```
crontab -e -u borg
```

Et son contenu :

```
15 3 * * * /opt/borgmatic_prune_and_check.sh
```

Initialisation du dépôt de sauvegarde

Sur l’ordinateur à sauvegarder :

```
borgmatic init -c /etc/borgmatic/mon_pc.yaml --append-only -e repokey
```

Le dépôt est initialisé en **append-only** (mais de toute façon, on le force dans le `.ssh/authorized_keys` du serveur distant) et avec un chiffrement par mot de passe mais avec la clé dans le dossier du dépôt distant.

Lancement d’une sauvegarde

Sur l’ordinateur à sauvegarder :

```
borgmatic create -c /etc/borgmatic/mon_pc.yaml --stats
```

Le `--stats` est l'option que j'affectionne et dont je parlais au début : cela affiche des statistiques sur la sauvegarde qui a été faite : sa taille, sa taille compressée, sa taille dédupliquée, le temps que ça a pris...

Afficher la liste des sauvegardes

```
borgmatic list -c /etc/borgmatic/mon_pc.yaml
```

Restauration de sauvegardes

Voir la section kivabien de l'article sur Borg.

Explication de l'option `--append-only`

Contrairement à ce qu'on pourrait croire, `--append-only` n'empêche pas de supprimer des sauvegardes (Ce ticket Github m'en a fait prendre conscience). Ainsi, un attaquant ayant pris le contrôle de l'ordinateur à sauvegarder pourra supprimer des sauvegardes. Sauf qu'elle ne seront pas vraiment supprimées : l'attaquant aura créé des transactions qui indiquent la suppression des sauvegardes mais ces transactions ne seront réellement appliquées que lors d'une action comme `prune` depuis un ordinateur qui aura le droit de faire sauter le **append-only**. C'est comme un `BEGIN TRANSACTION;` sans `COMMIT;` en SQL □□

C'est le cas du script `/opt/borgmatic_prune_and_check.sh` qui fait effectivement sauter (et le remet après) le verrou avec :

```
borg config /var/lib/borg/depot/ append_only 0
```

Comme on automatise la suppression des anciennes sauvegardes avec `cron`, il nous faut un moyen de repérer de tels changements. C'est tout le but de `/opt/borgmatic-stats/verify-archives.pl` qui compare la liste des archives du dépôt avec la liste créée la dernière fois que `/opt/borgmatic_prune_and_check.sh` a supprimé les anciennes archives. Comme le serveur distant est le seul à normalement pouvoir supprimer des archives, il est logique de retrouver toutes les anciennes archives au lancement suivant.

Ainsi, en cas d'incohérence, la suppression des anciennes archives ne s'effectue pas.

NOTA BENE : ceci n'empêchera pas un attaquant de changer la configuration de borgmatic pour lui faire sauvegarder des trucs inutiles plutôt que ce qui vous intéresse. Ou de couper les sauvegardes. C'est un inconvénient des sauvegardes en *push* plutôt qu'en *pull*. Je n'ai pas encore trouvé de solution à ça. Peut-être un script qui analyserait la sortie de `borgmatic create --stats --jon` qui me permettrait de repérer des changements importants dans la taille de la sauvegarde, le temps de sauvegarde ou le ratio de déduplication ?

En cas d'incohérence : restaurer des sauvegardes supprimées par un attaquant

Sur le serveur distant :

```
su - borg
cd /var/lib/borg/depot/
```

Regarder le log des transactions pour trouver les n° des transactions suspectes : `cat transactions`

Ce qui nous donne un truc du genre :

```
transaction 23, UTC time 2019-08-01T13:40:34.435019
transaction 25, UTC time 2019-08-01T13:42:05.662188
transaction 27, UTC time 2019-08-01T13:43:18.403771
transaction 29, UTC time 2019-08-01T13:43:53.306636
transaction 31, UTC time 2019-08-01T13:44:51.831937
transaction 33, UTC time 2019-08-01T13:45:17.786886
transaction 35, UTC time 2019-08-01T22:17:12.044179
transaction 37, UTC time 2019-08-02T08:02:55.005430
transaction 43, UTC time 2019-08-02T22:17:10.068843
transaction 45, UTC time 2019-08-03T22:17:09.625745
transaction 47, UTC time 2019-08-04T22:17:09.673447
transaction 49, UTC time 2019-08-05T22:17:13.709208
transaction 51, UTC time 2019-08-06T10:11:26.301496
transaction 53, UTC time 2019-08-06T10:20:46.178014
transaction 55, UTC time 2019-08-06T10:26:47.940003
```

Là, il faut savoir quand a eu lieu le problème. Ici, c'était le 6 août à partir de 10h. Donc les transactions 51 à 55. Mais **attention** ! Il s'agit en fait des transactions **50** à 55 : le numéro indiqué dans le fichier est celui du dernier fichier modifié par la transaction. Il faut donc partir du n+1 de la dernière transaction correcte.

Après, c'est facile :

```
rm hints.* index.* integrity.*
rm data/**/{50..55}
```

On finit par rafraîchir le cache du dépôt :

```
borg delete --cache-only /var/lib/borg/depot/
```

Sur l’ordinateur sauvegardé, par contre, je n’arrivais pas à supprimer le cache, j’avais ce message :

```
Cache, or information obtained from the security directory is newer than repository - this is
either an attack or unsafe (multiple repos with same ID)
```

Ceci m'a réglé le problème :

```
rm -rf ~/.cache/borg/le_dossier_avec_un_nom_monstrueux
rm -rf ~/.config/borg/security/un_autre_dossier_avec_un_nom_monstrueux
```

Révision #4
Créé 2020-01-23 16:58:36 CET par Luc
Mis à jour 2023-08-21 09:10:56 CEST par Luc