SSH sur plusieurs machines simultanément



Je me retrouve souvent à exécuter la même commande sur plusieurs serveurs en même temps.
Par exemple, je vais parfois avoir besoin de vérifier l'utilisation de la mémoire d'une flotte de serveurs.
D'autres fois, je veux `grep` rapidement les journaux de plusieurs machines, à des fins de dépannage.
Une fois que vous êtes au-delà de la gestion d'un ou deux serveurs, vous vous surprendrez à perdre 3 heures, si vous ne disposez pas d'un moyen d'exécuter des commandes sur plusieurs machines Linux simultanément.
Je vais vous expliquer comment exécuter rapidement des commandes sur un grand nombre de machines Linux à la fois.
Fait sur Debian Jessie

Outil dsh

Tout d'abord, nous aurons besoin d'installer dsh, qui est l'outil que nous allons utiliser pour faire plusieurs sessions ssh sur plusieurs machines à la fois.
Sur les distributions basées sur Debian exécutez:

```
aptitude install dsh
```

Une fois installé, il ya deux étapes de configuration supplémentaires dont vous aurez besoin avant de pouvoir vraiment à utiliser dsh.
La première consiste à la configuration de votre configuration du serveur ssh de telle sorte que vous pouvez vous connecter en ssh sur des machines distantes sans un mot de passe.
Il suffit de mettre ça clés SSH sur le système Linux
Une fois cela fait, vous devez construire vos fichiers de liste de machine.
Il y a deux classifications primaires de fichiers:

- fichier machines : liste qui peut contenir toutes les machines que vous possédez.
- les listes de groupes : qui vous permettent de spécifier des sous-ensembles de machines via les groupes.

Vous pouvez vous en sortir en spécifiant un fichier de machines.
une liste unique dans `~/.dsh/machines.list`, au format suivant:

```
username@hostname1
username@hostname2
username@hostname3
```

Si, toutefois, vous avez un grand nombre de machines que vous souhaitez classer par groupes, alors vous aurez probablement envie de le faire en utilisant le même format de fichier, mais en les plaçant dans un répertoire group `~/.dsh/groupe/groupname`.

Par exemple, vous pouvez avoir deux fichiers, l'un appelé `~/.dsh/groupe/serveurs_web` et un autre appelé `~/.dsh/groupe/dbservers`.



Notez que le nom de fichier que vous utilisez ici déterminera le nom du groupe comme spécifié sur la ligne de commande lorsque vous exécutez DSH.\\

Exemple d'utilisation

Une fois vos fichiers de configuration mis en place, vous êtes prêt à commencer à utiliser DSH.
Pour cet exemple, supposons que vous avez créé un groupe nommé “serveurs_web” avec deux machines (192.168.1.134 et 192.168.1.101), et que vous voulez trouver le temps de fonctionnement de ces deux machines.
Vous pouvez exécuter une commande comme:

```
dsh -r ssh -g serveurs_web -M -c "uptime"
192.168.1.134: 21:12:29 up 69 days, 47 min, 0 users, load average: 0.21,
0.31, 0.49
192.168.1.101: 21:13:18 up 69 days, 47 min, 1 user, load average: 0.01,
0.01, 0.00
```

Ici, nous précisons que nous voulons utiliser le shell distant ssh, que nous voulons utiliser le groupe `serveurs_Web`, l'option `-M` permet de préciser la machine qui a exécuté la commande et l'option `-c` que nous voulons exécuter la commande sur toutes les machines en même temps.

Typiquement, l'option -c est incroyablement utile, surtout lorsque vous exécutez une commande sur un très grand nombre de machines.
Par exemple, imaginez que vous avez une commande qui prend 5 secondes d'exécution avec l'option -c elle prendra environ 5 secondes, alors que sans l'option -c, il faudra 5 secondes multiplié par le nombre de machines.
Cependant, un inconvénient de l'option -c est que des machines de temps en temps écrivent sur la sortie d'un autre, ce qui peut rendre la lecture d'un grep un peu gênant.

Voici, le man de la commande.

Et vous comment gérez-vous votre flotte de serveurs?

Source : techthrob.com

From:
https://www.ksh-linux.info/ - **Know Sharing**

Permanent link:
https://www.ksh-linux.info/systeme/ssh-sur-plusieurs-machines-simultanement

Last update: **16/05/2022 16:04**