



- [Accueil](#)
- [A propos](#)
- [Nuage de Tags](#)
- [Contribuer](#)
- [Who's who](#)

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source

26 jan 2009

Un peu plus loin avec Linux vserver

Catégorie : [Administration système](#) Tags : [GLMF](#)



Retrouvez cet article dans : [Linux Magazine 92](#)

Dans le numéro 90 de janvier 2007, je vous avais présenté cet outil de virtualisation très utile et rapide à mettre en œuvre. Dans cet article, je vais vous présenter l'aspect plus système et quelques astuces sur les vservers, comme : Comment copier un vserver sur une machine distante ? Comment changer l'adresse IP du vserver sans forcément réinstaller le vserver ?...

1. Introduction

Tout ce qui sera montré ici sont des procédures expérimentées durant un projet lors de mon cursus scolaire. Je vais donc essayer de vous expliquer quelques astuces qui vous aideront sûrement. Par exemple, après avoir réalisé mon projet sous une VMWARE 5 alors qu'il fallait une version 4.5. Là, on se dit : « eh zut, faut que je refasse tout !!!! ». Eh ben, détrompez-vous, une simple copie des bons répertoires suffit. Pour les tests qui vont suivre, et du fait que cela constitue également mon projet, je me placerai dans une distribution Debian Sarge installée dans une VMWARE version 4.5. Mais avant tout cela, il y a une notion à comprendre sur le fonctionnement de vserver qui est celle du contexte. Ensuite, je vous présenterai plus en détail, et avec quelques astuces, ce qu'est un vserver.

2. La notion de contexte

Tout d'abord, maintenant que vous savez installer un vserver, un point important à dire dans un premier temps sur son fonctionnement. Vous vous demandez : mais comment

cela marche-t-il ? Je vais essayer d'être très clair. Il faut donc introduire la notion de « contexte ». Le but de ce contexte est de cacher tous les processus et de permettre à un processus externe de ne pas interagir avec un processus d'un vserver. Par défaut, un contexte sur la machine hôte est créé, c'est le contexte « 0 ». C'est lui qui va permettre de définir une séparation entre les différents autres contextes. C'est grâce à la modification du noyau, c'est-à-dire au patch appliqué lors de l'installation, que ceci est possible. Par conséquent, lorsqu'un processus est créé à l'intérieur d'un contexte, ce processus appartient à ce contexte. Faisons par exemple le test de lister tous les processus sur la machine hôte ainsi que ceux appartenant à un vserver. Voyons dans un premier temps le résultat du coté hôte.

```
vmdebian:~# ps a
  PID TTY          STAT       TIME COMMAND
  1934 tty1      Ss+        0:00 -bash
  1935 tty2      Ss+        0:00 /sbin/getty 38400 tty2
  1936 tty3      Ss+        0:00 /sbin/getty 38400 tty3
  1937 tty4      Ss+        0:00 /sbin/getty 38400 tty4
  1938 tty5      Ss+        0:00 /sbin/getty 38400 tty5
  1939 tty6      Ss+        0:00 /sbin/getty 38400 tty6
  1993 pts/0     Ss         0:00 -bash
  2131 pts/1     Ss         0:00 -bash
  2136 pts/1     R+         0:00 ps a
vmdebian:~#
```

On remarque que l'on n'aperçoit que les processus internes à la machine hôte. Pour vous convaincre que c'est vraiment le cas, regardons le résultat à l'intérieur d'un contexte.

```
Loches:/# ps a
  PID TTY          STAT       TIME COMMAND
  1999 pts/0     S+         0:00 /bin/bash -login
  2138 pts/1     S          0:00 /bin/bash -login
  2153 pts/1     R+         0:00 ps a
Loches:/#
```

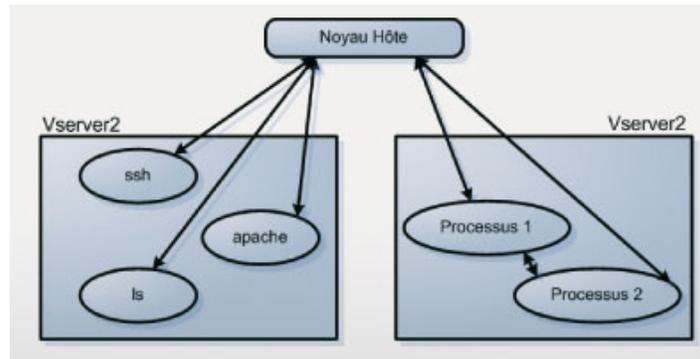
Le résultat reste quand même différent de ce que l'on a eu dans le premier exemple. En effet, ici nous n'avons que ce qui est interne au contexte, donc bien séparé des autres contextes et donc visible que par le contexte « propriétaire ». Toutefois, une commande existe pour afficher l'ensemble des processus lancés à partir de la machine hôte dont le résultat se trouve sur le troisième exemple. Il s'agit de la commande **vps**.

```
vmdebian:~# vps a
  PID CONTEXT          TTY     STAT  TIME COMMAND
  1934    0 MAIN              tty1    Ss+   0:00 -bash
  1935    0 MAIN              tty2    Ss+   0:00 /sbin/getty 38400 tty2
  1936    0 MAIN              tty3    Ss+   0:00 /sbin/getty 38400 tty3
  1937    0 MAIN              tty4    Ss+   0:00 /sbin/getty 38400 tty4
  1938    0 MAIN              tty5    Ss+   0:00 /sbin/getty 38400 tty5
  1939    0 MAIN              tty6    Ss+   0:00 /sbin/getty 38400 tty6
  1993    0 MAIN              pts/0   Ss    0:00 -bash
  1999 49152 Loches            pts/0   S+    0:00 /bin/bash -login
  2131    0 MAIN              pts/1   Ss    0:00 -bash
  2161    1 ALL_PROC          pts/1   S+    0:00 vps a
  2162    1 ALL_PROC          pts/1   R+    0:00 ps a
vmdebian:~#
```

Pour finir cette explication, sur le schéma de la figure 1, j'ai essayé de faire un résumé de ce que je viens de vous expliquer.

(Voir figure 1, page 49)

J'ai donc essayé de vous montrer ici les différentes interactions possibles entre la machine hôte et le contexte. Dans le vserver 1, j'ai intitulé des processus avec des noms connus. Les flèches montrent que la machine hôte peut voir ces processus. Dans le vserver 2, j'ai nommé deux processus sans nom, juste pour vous montrer que ces deux processus peuvent dialoguer entre eux.



2.1 Création d'un contexte

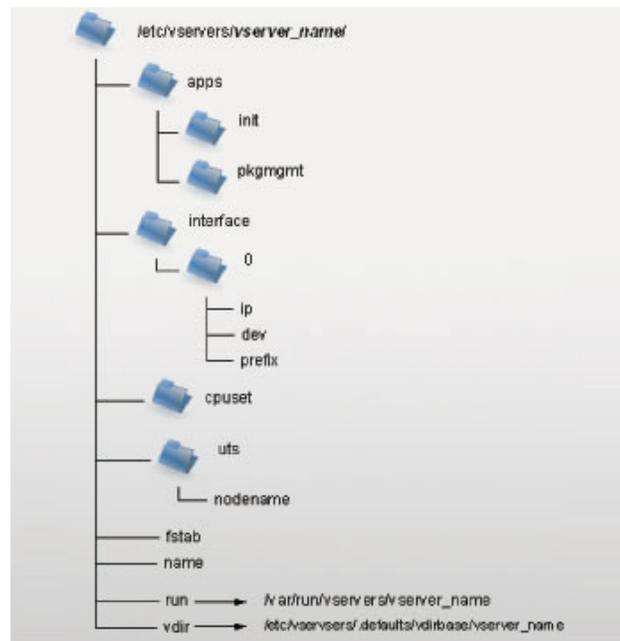
Lors de la création d'un nouveau contexte, ce dernier va être associé à un sous-répertoire de la machine hôte. Ce répertoire va ensuite correspondre à la « racine » du vserver ou plutôt du contexte. Il contiendra tous les services, répertoires ou encore fichiers nécessaires pour le bon fonctionnement du vserver. Au final, nous avons un ensemble de processus tournant dans un endroit « clos » disposant d'une configuration personnelle : nom de machine, adresse IP, utilisateur... et cela constitue un serveur virtuel.

2.2 Le réseau dans tout cela...

Chaque contexte dispose d'une adresse IP. Le noyau fera un routage vers le processus adéquat contenu dans le bon contexte. Il est donc impossible que deux vservers disposent de la même adresse IP. Cela constituerait un conflit d'adressage IP, comme ce serait le cas dans un LAN avec deux adresses identiques.

3. Qu'est-ce qu'un vserver ?

Vous avez sans doute déjà installé votre vserver sans comprendre le rôle de chaque répertoire. Je vais vous expliquer l'utilité de certains d'entre eux. Tout d'abord, après avoir suivi le processus d'installation et créé votre premier vserver, vous aurez une architecture comme représenté sur la figure 2.



Dans ces répertoires, il va vous être possible de réaliser plusieurs choses. Je vais vous expliquer comment faire en sorte qu'un vserver démarre automatiquement au démarrage du poste principal ou encore comment faire pour que, lorsque l'on lance la commande `ifconfig`, nous arrivions à voir apparaître l'adresse IP de la machine.

3.1 Démarrer un vserver au démarrage de la machine hôte

Lors de la machine hôte, il est possible de choisir de démarrer un vserver automatiquement. Dans la plupart des documentations vues sur le net, dans le fameux fichier `vserver_name.conf`, il suffisait de rajouter `ONBOOT=yes`. Or, depuis le changement, il faut pour cela faire la chose suivante :

```
# mkdir -p /etc/vservers/vserver1/apps/init
# echo «default» > /etc/vservers/vserver1/apps/init/mark
```

Dans le répertoire `apps/init` correspondant à notre vserver, nous allons créer un fichier contenant juste `default`. Au démarrage, un service va donc détecter ce fichier et lancer tous les `vservers` qui auront ce fichier.

3.2 Connaître l'adresse IP d'un vserver en étant dans le vserver

Il est très difficile de vous présenter toutes les options possibles des Linux vserver. Dans l'ancienne version, il suffisait d'éditer un fichier du genre `vserver_name.conf` dans le répertoire `/etc/vservers`. Depuis, les choses ont pas mal évolué. Dans le répertoire d'un Linux vserver, il est possible de créer des fichiers de configuration bien spécifique. Par exemple, lorsque vous êtes dans un vserver, vous allez sûrement trouver dommage de ne pas connaître en tapant `ifconfig` l'adresse IP du serveur. Pour ce faire, arrêter le vserver,

éditer le fichier `/etc/vserver/vserver_name/interface/0/name` et mettez un nom, par exemple `dummy`. Relancez votre vserver, tapez `ifconfig` et là vous aurez l'adresse IP du serveur.

```
Loches:/# ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:0C:29:E4:97:59
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2190 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1912 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:289857 (283.0 KiB)  TX bytes:678615 (662.7 KiB)
          Interruption:18 Adresse de base:0x1080

eth0:dumm Lien encap:Ethernet  HWaddr 00:0C:29:E4:97:59
          inet adr:192.168.0.21 Bcast:192.168.0.255 Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interruption:18 Adresse de base:0x1080

Loches:/#
```

3.3 Exemple d'un fichier pris sur le net et réalisation avec la nouvelle méthode

Dans cette partie, nous allons prendre exemple sur un fichier pris sur le net et voir comment faire la même chose, mais avec la nouvelle façon de faire. Nous avons bien essayé de faire avec un fichier de même type mais, hélas, cela n'a pas donné le résultat tant espéré.

```
# Most easy thing is to have an own ip-address for each vserver
IPROOT=10.95.81.17
IPROOTMASK=255.255.255.0
# How shall the networkdevice be named from the view of the vserver
IPROOTDEV=eth0
# shown hostname
S_HOSTNAME=mysql5
# lock = you are not allowed to create a new context in your context
# nproc = let ULIMIT-value be global for this context
S_FLAGS=>>lock nproc>>
# What capabilities shall the server have? Look for explanation
S_CAPS=>>CAP_NET_RAW CAP_NET_BIND_SERVICE>>
# Start this server on booting? This will be checked in /etc/init.d/vservers
ONBOOT=yes
```

Dans la première partie de ce fichier, nous trouvons l'adresse IP, le masque du vserver et le nom vu par le vserver pour l'interface réseau. Allez dans le répertoire `/etc/vservers/vserver_name/interface/0/`. Vous trouverez 3 fichiers.

- `ip`: contenant l'adresse IP;
- `prefix`: le masque de sous-réseau;
- `dev`: contenant le nom du périphérique réseau.

Il vous suffit donc de remplir ces trois fichiers pour configurer l'interface réseau. `S_HOSTNAME` correspond au nom de la machine. Ce dernier se trouve dans `/etc/vservers/vserver_name/uts/nodename`. Inscrivez dans ce fichier le nom que vous désirez, relancez et constatez le changement. Le fichier `name` se trouvant dans le

répertoire du vserver contient le nom du vserver. La ligne `S_FLAGS` sera désormais dans le fichier `flags` directement dans le répertoire du vserver. Au même endroit que pour les `flags`, mais dans le fichier `capabilities`, vous mettez la ligne `S_CAPS` → `CAP_NET_RAW` `CAP_NET_BIND_SERVICE`. La dernière ligne a déjà été traitée dans le paragraphe « 3.1 Démarrer un vserver au démarrage de la machine hôte ».

3.4. Le déplacement d'un vserver sur une machine distante

Dans cette partie, je vais vous expliquer comment copier un vserver sur une autre machine. Mais avant tout, il est bon de connaître l'architecture d'un vserver. Un vserver est une sorte de « chroot amélioré ». On peut changer pas mal de choses sur un vserver, si on connaît bien les répertoires qui le composent ainsi que les fichiers dans ces répertoires. Si vous avez bien suivi la méthode d'installation de l'article précédent, vous trouverez donc dans votre système deux répertoires importants. Dans le premier, `/var/lib/vservers/vserver_name`, vous trouverez l'architecture du vserver. C'est-à-dire tous les répertoires de votre système, ce que vous voyez lorsque vous entrez dans votre vserver. Le deuxième, `/etc/vservers/vserver_name`, présenté sur la figure 2, contient les fichiers de configuration de votre vserver ainsi qu'un lien vers le premier répertoire présenté. La commande reste simple, car elle consiste en une synchronisation entre les deux machines,

```
#rsync -e ssh -avHL /vservers/vserver_name      adresse_ip_machine_distante:/etc/vservers/vserver_name
#rsync -e ssh -avHL /var/lib/vservers/vserver_name  adresse_ip_machine_distante:/var/lib/vservers/vserver_name
```

puis à tester en lançant sur la machine distante si le vserver a bien été synchronisé.

3.5 Listes des outils

La liste des commandes qui vont vous être présentées ne sont utilisables qu'à partir du contexte 0, c'est-à-dire à partir de la machine hôte.

- `vtop`: `top` global sur tous serveurs ;
- `vps`: `ps` global sur tous serveurs ;
- `vpstree`: `pstree` global sur tous serveurs ;
- `vdu`: `du` qui ne se mélange pas avec les liens ;
- `vkill`: `kill` global sur tous serveurs ;
- `vserver-copy`: copie de vservers ;
- `vfiles`: extraire la liste des fichiers spécifiques (les liens durs) à un vserver (gain de place) ;
- `vunify`: inclure les fichiers manquants (liens symboliques) après un `vfiles`.

4. Caractéristiques d'un vserver

Pour ceux qui n'auraient pas lu l'article du mois dernier, je vais vous réexpliquer certaines caractéristiques d'un vserver. Tout d'abord, la première chose, c'est que comparé à d'autres méthodes comme VMWARE qui, lui, utilise la totalité des ressources de la machine, vserver, n'utilise que ce dont il a besoin. Pour faire plus simple, ce qui est utilisé en ressource par un vserver correspond aux processus tournant à l'intérieur. Ce système reste un système assez sécurisé. Comme présenté dans la figure 1, on

remarque que les discussions entre vservers sont totalement impossibles. De plus, même un super-utilisateur ne peut voir ce qui se passe à l'extérieur de son contexte. Dans le cas d'une intrusion, la seule partie visible sera le contenu du contexte correspondant. La maintenabilité d'un tel système reste simple. Il suffit de faire régulièrement des sauvegardes des répertoires du vserver dans, par exemple, un fichier `tar.gz` et de le décompresser quand cela est nécessaire, puis ensuite de relancer le vserver et tout marche à nouveau.

L'un des seuls inconvénients que l'on puisse trouver, c'est la place occupée par chaque vserver. Comme chacun dispose de son propre système de fichiers, que chaque programme se situe dans un vserver, au final, la place disque sera de plus en plus petite. En moyenne, les systèmes que j'ai mis en place pèsent environ 400 Mo. Imaginez la place prise lorsque, sur une machine en production, il y a 10 vservers qui tournent. Une technique d'unification existe pour permettre à plusieurs vservers de se partager des applications, ce qui permet un gain considérable. Comptez environ 600 voire 700 Mo au lieu de 4 Go correspondant à la place occupée par les 10 vservers sans unification.

Conclusion

Linux vserver est donc un système de virtualisation très performant et sécurisé à mon sens. La sauvegarde d'un tel système est très simple à faire et la maintenabilité simplifiée. Si une partie est affectée, la restauration se fait rapidement. Le système peut être gourmand en espace disque, mais non en ressource. L'une des seules limites que l'on peut constater, c'est que chaque vserver se partage le même noyau. Dans la plupart des cas, cela ne constitue pas une gêne importante, sauf dans le cas où une application nécessite un patch du kernel.

Retrouvez cet article dans : [Linux Magazine 92](#)

Posté par ([La rédaction](#)) | Signature : Cyril Meusnier | Article paru dans



Laissez une réponse

Vous devez avoir ouvert une [session](#) pour écrire un commentaire.

« [Précédent](#) [Aller au contenu](#) »

[Identifiez-vous](#)

[Inscription](#)

[S'abonner à UNIX Garden](#)

• Articles de 1ère page

- [Git, les mains dans le cambouis](#)
- [PostgreSQL 8.3 : quoi de neuf ?](#)
- [Introduction à Ruby on Rails](#)
- [Linux Pratique HS N°17 - Mars/Avril 2009 - Chez votre marchand de journaux](#)

- [88 miles à l'heure!](#)
- [Développement et mise en place d'un démon Unix](#)
- [Calculer ses rendus Blender en cluster ou comment faire sa propre « render farm » avec DrQueue](#)
- [Technologie rootkit sous Linux/Unix](#)
- [CMake : la relève dans la construction de projets](#)
- [Des petits sondages pour améliorer nos magazines](#)



[Actuellement en kiosque :](#)

- **Catégories**
 - [Administration réseau](#)
 - [Administration système](#)
 - [Agenda-Interview](#)
 - [Audio-vidéo](#)
 - [Bureautique](#)
 - [Comprendre](#)

- [Distribution](#)
- [Embarqué](#)
- [Environnement de bureau](#)
- [Graphisme](#)
- [Jeux](#)
- [Matériel](#)
- [News](#)
- [Programmation](#)
- [Réfléchir](#)
- [Sécurité](#)
- [Utilitaires](#)
- [Web](#)

• Articles secondaires

- 30/10/2008

[Google Gears : les services de Google offline](#)

Lancé à l'occasion du Google Developer Day 2007 (le 31 mai dernier), Google Gears est une extension open source pour Firefox et Internet Explorer permettant de continuer à accéder à des services et applications Google, même si l'on est déconnecté....

[Voir l'article...](#)

7/8/2008

[Trois questions à...](#)

Alexis Nikichine, développeur chez IDM, la société qui a conçu l'interface et le moteur de recherche de l'EHM....

[Voir l'article...](#)

11/7/2008

[Protéger une page avec un mot de passe](#)

En général, le problème n'est pas de protéger une page, mais de protéger le répertoire qui la contient. Avec Apache, vous pouvez mettre un fichier `.htaccess` dans le répertoire à protéger....

[Voir l'article...](#)

6/7/2008

[hypermail : Conversion mbox vers HTML](#)

Comment conserver tous vos échanges de mails, ou du moins, tous vos mails reçus depuis des années ? mbox, maildir, texte... les formats ne manquent pas. ...

[Voir l'article...](#)

6/7/2008

[iozone3 : Benchmark de disque](#)

En fonction de l'utilisation de votre système, et dans bien des cas, les performances

des disques et des systèmes de fichiers sont très importantes....

[Voir l'article...](#)

1/7/2008

[Augmentez le trafic sur votre blog !](#)

Google Blog Search (<http://blogsearch.google.fr/>) est un moteur de recherche consacré aux blogs, l'un des nombreux services proposés par la célèbre firme californienne....

[Voir l'article...](#)

• [GNU/Linux Magazine](#)

- - [GNU/Linux Magazine N°113 - Février 2009 - Chez votre marchand de journaux](#)
 - [Édito : GNU/Linux Magazine 113](#)
 - [Un petit sondage pour améliorer nos magazines](#)
 - [GNU/Linux Magazine HS N°40 - Janvier/Février 2009 - Chez votre marchand de journaux](#)
 - [Edito : GNU/Linux Magazine HS 40](#)

• [GNU/Linux Pratique](#)

- - [Linux Pratique HS N°17 - Mars/Avril 2009 - Chez votre marchand de journaux](#)
 - [Édito : Linux Pratique HS N°17](#)
 - [Linux Pratique HS 17 - Communiqué de presse](#)
 - [Linux Pratique Essentiel N°6 - Février/Mars 2009 - Chez votre marchand de journaux](#)
 - [Édito : Linux Pratique Essentiel N°6](#)

• [MISC Magazine](#)

- - [Un petit sondage pour améliorer nos magazines](#)
 - [MISC N°41 : La cybercriminalité ...ou quand le net se met au crime organisé - Janvier/Février 2009 - Chez votre marchand de journaux](#)
 - [Édito : Misc 41](#)
 - [MISC 41 - Communiqué de presse](#)
 - [Les Éditions Diamond adhèrent à l'APRIL !](#)

© 2007 - 2009 [UNIX Garden](#). Tous droits réservés .