

Informationen über den Bericht

Der alte vServer war ein *vServer 1536* von Netcup, der neue ein *vServer Gold*. Meine Domains lagen bei Netcup und ALL-INKL (Neue Medien Münnich). Auf dem vServer war Debian Lenny installiert, die Befehle müssen bei anderen Systemen deshalb vermutlich angepasst werden! Auf dem vServer selbst lief ein LAMP System mit Syscp, einige IRC-BNC's, einige IRC-Bots, Squid, Mail-Server und einige andere – eher unbekannte oder selbst entwickelte – Programme/Dienste/Scripte. Ich versuche zwar jeden Schritt genau zu beschreiben, aber jeder vServer ist anders, dementsprechend muss man auch selbst handeln können, wenn etwas nicht wie gewünscht klappt. Der Bericht geht außerdem davon aus, dass die Daten des vServer im Recovery Modus in /vserver liegen (so wie es bei Netcup der Fall ist). Für Backups kann man bei Netcup gleich das mitgelieferte Backup-Tool im Control Panel verwenden, das im openVCP integriert ist. Alle beschriebenen Aktionen wurden einen Tag vorher mit einem Test-vServer ausprobiert, dabei habe ich auch alle Befehle und Erfahrungen dafür gesammelt. Der eigentliche Bericht stammt letztendlich von meinem Produktivsystem. bzip2 könnte man bei allen Aktionen auch einfach weglassen, wenn man etwas Zeit (dafür aber nicht Traffic) sparen möchte. Bitte dann aber nicht vergessen die Dateinamen und Befehle jeweils anzupassen. Ich habe mich für bzip2 entschieden, da ich mir so ein komplettes Backup einmal wieder für zu Hause runterladen kann, denn ansonsten speichere ich alle Backups auf Backupservern. Ohne bzip2 wäre es einfach viel zu groß zum Runterladen geworden (ebenfalls bei meiner Internetverbindung). Für das Klonen/Umziehen sollte man mehrere Stunden Zeit einplanen, vor allem bei größeren Systemen, bei denen vielleicht mehr Konfigurationsdateien angepasst werden müssen und mehr Dateien gepackt, komprimiert und kopiert werden müssen. Falls die Gefahr besteht, dass die eigene Internetverbindung während der Aktionen abbricht, sollte man „screen“ verwenden. Also eigentlich eh bei fast jedem ADSL Anschluss durch die Zwangstrennung ;-)

- » **Ich übernehme keine Haftung für Schäden, die direkt oder indirekt durch diesen Bericht entstanden sind, die Verwendung der Befehle erfolgt auf eigene Gefahr!**

Vorbereitungen

Wie bei jedem Serverumzug sollte man die User von betroffenen Projekten zuerst einmal informieren, dass durch die IP-/Domainänderung die Domain(s) bis zu 48 Stunden nicht erreichbar sein werden.

Vorbereitungen beim alten vServer

Deaktivierung aller relevanten Cron-Einträge und allgemeinen Scripten, die Dienste doppelt starten könnten (wie IRC-BNC's, Bots, Backupprozesse, usw.), wenn man den alten vServer nachher wieder starten möchte. Alle wichtigen Dateien zusätzlich sichern (sollte eigentlich klar sein bei so großen Änderungen, oder?). Der vServer wird zwar geklont (und die Daten bleiben vorerst auf dem alten vServer vorhanden), aber ein Vertipper passiert ja schnell ;-)

Zusätzlich könnte man auch gleich Cache Dateien bereinigen/löschen, das spart nachher Zeit und Traffic. Cache Dateien mitzukopieren wäre absolut unnötig und idiotisch :P

Vorbereitungen beim neuen vServer

Der neue vServer sollte bereits die benötigte Anzahl an IP's zugeordnet haben. Außerdem sollte man ihn komplett neuinstallieren (am besten dasselbe Betriebssystem wie beim anderen vServer), damit die /dev Einträge wirklich für unser System gebraucht werden können.

vServer im Recovery Modus booten

Nun muss man beide vServer (den alten und neuen) im Recovery Modus booten. Das temporäre Root-Passwort von beiden vServern bitte notieren, das brauchen wir noch öfters!

Alle Dateien packen und komprimieren

Nun loggen wir uns beim alten vServer mit dem temporären Root-Passwort ein. Um das Kopieren der Daten nachher zu verkürzen müssen wir bzip2 einmal nachinstallieren:

```
apt-get install bzip2
```

Nun legen wir uns ein temporäres Verzeichnis, in dem wir unsere tar.bz2 Datei erstellen:

```
mkdir ~/clone
```

Und jetzt geht's ans Packen und Komprimieren aller Dateien, ohne nice geht's vermutlich schneller, aber wir beeinträchtigen damit wenigstens keine anderen vServer Kunden:

```
tar cfv ~/clone/clone.tar /vserver  
nice -n 19 bzip2 -9 ~/clone/clone.tar
```

Das Packen und Komprimieren dauert jetzt je nach Anzahl und Größe der Dateien mindestens 10 Minuten, wenn man sehr viel Speicher belegt hat wäre 1 Stunde oder mehr aber auch keine Besonderheit. Wenn man hier etwas Zeit sparen möchte, kann man die Blockgröße beim Komprimieren mit bzip2 (-9 = beste Komprimierung) herunter setzen oder bzip2 komplett weglassen, das muss jeder für sich entscheiden, wie schnell der Prozessor und das Netzwerk des vServers ist, was also besser ist. Die gesparte Zeit geht dann beim Kopieren über das Netzwerk zum neuen vServer aber vielleicht wieder verloren. Außerdem muss man bedenken, dass der Traffic nicht bei jedem Hoster inklusive ist, je nachdem von wo man wohin wechselt. Während die Dateien noch gepackt und komprimiert werden kann man sich eigentlich bereits dem nächsten Schritt zuwenden oder einmal einen Kaffee/Tee/Limo trinken ;-)

Neuer vServer: Weitere Vorbereitung

Auch hier müssen wir bzip2 nachinstallieren:

```
apt-get install bzip2
```

Nun legen wir noch einen temporären Arbeitsordner an, den wir nachher gleich brauchen:

```
mkdir ~/clone
```

Alles kopieren...

Sofern beim alten vServer nun endlich alles gepackt und komprimiert ist, kopieren wir über SSH unsere fertige tar.bz2 Datei auf den neuen vServer. Dazu führen wir folgenden Befehl auf dem alten vServer aus, die IP muss durch die IP des neuen vServer ersetzt werden:

```
scp ~/clone/clone.tar.bz2 root@0.0.0.0:~/clone/clone.tar.bz2
```

Das Kopieren dauert jetzt wieder je nach Größe der tar.bz2 Datei einige Minuten bis Stunden. Hier heißt es wieder warten und Kaffee/Tee/Limo trinken. Sobald das Kopieren abgeschlossen ist, haben wir auf dem alten vServer alles erledigt. Ich empfehle jedoch den alten vServer jetzt trotzdem noch im Recovery Modus online zu lassen. Denn wenn man nachher auf dem neuen vServer einen schwerwiegenden Fehler macht und die tar.bz2 Datei nochmals kopieren müsste, würde man so verdammt viel Zeit sparen. Und vielleicht will man sich die tar.bz2

Bericht: vServer klonen/umziehen

Autor: killerbees19 (Christian Schrötter) – <http://www.happytec.at/>

Lizenz: Creative Commons BY-NC-ND 3.0 AT (<http://creativecommons.org/licenses/by-nc-nd/3.0/at/>)

Datei ja noch komplett auf den Computer heruntergeladen? Als Tipp: Falls man die tar.bz2 Datei später noch haben möchte, muss man sie unbedingt in ein Unterverzeichnis von /vserver verschieben, da sie ansonsten beim nächsten Boot des vServer weg ist!

Wir entpacken unsere tar.bz2 Datei

Auf dem neuen vServer können wir nun alles einfach dekomprimieren und entpacken (inklusive der Datei-Informationen wie z.B. Rechte, Besitzer, Änderungsdatum, ...).

```
bunzip2 ~/clone/clone.tar.bz2
cd /
mv vserver vserver_backup
tar xfvp ~/clone/clone.tar
```

Dabei kann es am Ende einen Fehler von „tar“ geben, diesen kann man im Normalfall aber ignorieren, da es sich dabei nur darum handeln sollte, dass /dev nicht wiederhergestellt werden konnte. Danach müssen wir das alte „echte“ /dev zurückholen:

```
rm -r /vserver/dev
mv /vserver_backup/dev /vserver/dev
```

Und einige andere Dateien kopieren wir uns noch als Backup:

```
mkdir /vserver/backup
mv /vserver_backup /vserver/backup
mv ~/clone /vserver/backup
```

Nun geht's an anpassen der IP's in den Konfigurationsdateien. Jetzt werden nur die wichtigsten Dateien geändert, die meisten anderen Änderungen kann man dann sowieso erst machen, wenn der vServer wieder läuft. Folgende Dateien müssen angepasst werden:

```
/vserver/etc/hosts
/vserver/etc/bind/default.zone (sofern installiert)
/vserver/etc/issue.net (sofern verändert und relevante Informationen drinnen stehen)
```

Welche Dateien geändert werden müssen, kann man durch eine kurze Suche feststellen, einfach alle IP's (und Hostnamen) des alten vServers suchen. Ich habe mich dabei auf das Verzeichnis /etc beschränkt, andere kann man ja auf Wunsch auch noch durchsuchen:

```
cd /vserver/etc && find -type f|xargs grep "0\.\0\.\0\.\0"
```

Je nach installierter Software muss nun mehr oder weniger anpassen. Bitte bedenke, dass man sehr viele Konfigurationsdateien nicht direkt so anpassen sollte und erst später mit den wirklichen Tools (ob Kommandozeilenprogramme oder z.B. Syscp|Confixx|Plesk für die Configs vom Apache) alles ändern sollte. Wenn man sich halbwegs sicher ist, dass die wichtigsten Änderungen durchgeführt wurden (bei mir waren es vorerst nicht mehr), kann man den vServer wieder im normalen Modus booten lassen. Dabei auftretende Fehler bitte unbedingt notieren, die gehören alle der Reihe nach behoben!

Beim Boot gab es bei mir keine Fehler, die mit dem Umzug zusammenhängen. Die zwei erschienenen Fehler wurden durch Konfigurationsfehler bereits Wochen vorher verursacht und nun auch gleich behoben. Zuerst einmal musste ich meine Apache Config ändern, um überhaupt an mein Syscp ranzukommen (da es bei mir nur über eine bestimmte Subdomain aufrufbar ist). In Syscp habe ich dann einmal alle IP's geändert. Danach änderte ich noch alle DNS Einstellungen (was bei den vielen Subdomains etwas dauerte), je nach Hoster muss man dazu den Support anschreiben, wenn man die Änderungen nicht selbst vornehmen kann. Im

Bericht: vServer klonen/umziehen

Autor: killerbees19 (Christian Schrötter) – <http://www.happytec.at/>

Lizenz: Creative Commons BY-NC-ND 3.0 AT (<http://creativecommons.org/licenses/by-nc-nd/3.0/at/>)

openVCP Control Panel habe ich dann den Hostnamen noch angepasst und die rDNS Einträge für die neuen IP's von meinem Hoster setzen lassen. Die Konfigurationen vom Apache und Mailserver muss man je nach installierten Erweiterungen dann auch noch anpassen. In MySQL (auch auf anderen Servern) musste ich dann noch die erlaubten Hosts der Benutzer ändern. Außerdem sollte man nicht vergessen die Firewallregeln neu anzulegen, wenn man z.B. die im openVCP integrierte Firewall benutzt.

Falls man auf dem neuen vServer nicht mehr dieselbe Größe an Arbeitsspeicher zur Verfügung hat, egal ob jetzt mehr oder weniger Ram ist, empfiehlt es sich auch, die Apache und MySQL Konfigurationen im Bezug auf die Ressourcen anzupassen.

Falls man die Ausfallzeit durch die DNS-Änderung verkürzen möchte, kann man auf dem alten vServer auch rinetd installieren und die wichtigsten Ports/IP's auf den neuen vServer weiterleiten lassen. Das kann recht nützlich sein und erspart den Usern die bis zu 48 Stunden dauernde Ausfallzeit.

Insgesamt war mein vServer 4 Stunden nicht erreichbar plus noch eine Stunde, bis wirklich alle Konfigurationen angepasst waren. Wenn man bzip2 weglässt und sich beeilt (nicht wie ich dazwischen diesen Bericht tippt ; -) schafft man den Umzug aber sicher auch innerhalb einer Stunde. Natürlich nur, wenn nicht zu viele Dateien kopiert und angepasst werden müssen.

Nach einigen Tagen kann ich jedenfalls nur noch sagen: Alles funktioniert perfekt :-)