
Subject: Re: /vz/private , /vz/root , backup und mount
Posted by [TheStig](#) on Fri, 05 Mar 2010 10:08:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hm... ich mounte meine LVMs/Partitionen/Festplatten immer zuerst in der HN ein und dann stelle sie dann per mount --bind einem VPS zur Verfügung. Auf die Art und Weise steigt auch tar in diese Verzeichnisse ab (zumindest bei mir . Allerdings benutze ich das so nicht aktiv - sondern archiviere immer einzelne (Web-)Anwendungen bzw. deren MySQL-DBs. Dazu verwende ich ein Script, dass per vzctl exec VEID mysqldump zuerst einen (in der virtuellen Maschine) lokalen Dump erzeugt und diesen dann von der HN "weitschickt".

Mein Script arbeitet da ungefähr wie folgt:

config-Part:

```
#define database source: res_mysql="[declare as mysql] [where to store] [dbase_name] [dbuser] [dbpasswd] [(if_db_in_VZ)VZID]"  
# the name of the array has to start with "res"
```

```
res_customer1database="mysql www_myapp_net /var/backup/local/ bkupuser securepasswd  
144"
```

Das Backup-Script lädt dann ungefähr folgend:

```
source $configfile
```

```
date1=`date +%Y.%m.%d`
```

```
errlog="/var/log/wherever/"
```

```
vzctl="/usr/sbin/vzctl"
```

```
#### mysqldump function
```

```
_dump() {
```

```
# check for backup dir
```

```
if [ ! -d $2/$5 ] ; then mkdir -p $2/$5 ; fi
```

```
# check if to use inside a VPS
```

```
case $# in
```

```
# dump locally
```

```
4 ) cd $2 && mysqldump -u $3 -p$4 $1 > $1.$date1.sql && \
```

```
tar Pczf $1.$date1.sql.tar.gz $1.$date1.sql && rm -rf $1.$date1.sql 2>> $errlog ;;
```

```
# dump inside VPS
```

```
5 ) $vzctl exec $5 "if [ ! -d $2 ] ; then mkdir -p $2 ; fi && cd $2 && \
```

```
mysqldump -u $3 -p$4 $1 > $1.$date1.sql && \
```

```
tar Pczf $1.$date1.sql.tar.gz $2/$1.$date1.sql && rm -rf $2/$1.$date1.sql" 2>> $errlog
```

```
# now for example move from VPS to backup device
```

```
mv /vz/root/$5/$2/$1.$date1.sql.tar.gz $2/$5
```

```
;;
```

```
esac
```

Mein Backup-Verzeichnis wird per rsync (read only) freigegeben und der Backup-Server ist für den automatischen Abgleich zuständig.

Damit der obere Code-Teil funktioniert muss der Inhalt der Config noch an die dump-Funktion weitergegeben werden. Das mach ich so:

```
#### call functions
k=${!res@}
for b in $k ; do
a=${!b}
    case ${a[0]} in
mysql ) _dump "${a[1]}" "${a[2]}" "${a[3]}" "${a[4]}" "${a[5]}" ;;
* ) _mailto "Don't know how to backup ${a[1]}" "ERROR" ;;
esac

done
```

Das ganze mach ich über case und Funktionen, weil mein Script noch andere Definitionen als "mysql" als Wert \$0 des Arrays kennt, zb. vhosts, und diese dann über eine andere Funktion archiviert und zum Backup freigibt. Kann ich bei bedarf gerne nachliefern. ebenso die Funktion mailto, welche mich per eMail über Fehler informiert.

Außerdem hab lösche ich meine mehr als eine Woche alten Archive lokal (und via rsync --delete auch am Backup) mit

```
find $2/$5/ -mtime +7 -execdir rm -f '{}' \;
```

direkt im Backup-Script.
