

Kako pripremiti identičnu kopiju Linux poslužitelja



Iako su serveri kvalitetniji od običnih, korisničkih računala, i oni se mogu pokvariti. Sistemci koji su navikli svoje korisnike na 24/7 uslugu razmišljaju unaprijed, pa se nastoje pripremiti za situaciju kad server zbog hardverskog kvara ispadne iz produkcije. Ovdje ćemo predstaviti jedno jeftino i praktično rješenje. Naime, moguće je napraviti identičnu kopiju softvera i podataka sa produkcijskog Linux servera kako bi bila spremna za prijenos na zamjensko računalo.

Ideja je da na drugi Linux sistem ili na sam produkcijski poslužitelj spojimo, npr. preko USB sučelja, disk koji se sinkronizira sa sistemskim diskom poslužitelja i spreman je da ga, u slučaju potrebe, ugradimo u rezervni poslužitelj. Na taj ćemo način u kratkom roku podići funkcionalan zamjenski poslužitelj, u stanju posljednje sinkronizacije.

Postupak se može najzornije prikazati primjerom.

Predpostavimo da je na izvoru sistemski disk `/dev/sda`, a ciljni (sistemski) disk budućeg rezervnog poslužitelja `/dev/sdc`. U tekstu ćemo se držati ove konvencije, a vi ćete, naravno, sve prilagoditi stanju na vašem sustavu. Važno je samo da smo uvijek svjesni koje naredbe izvršavamo na izvornom, a koje na ciljnom računalu!

Ciljni disk u pravilu treba biti jednakog ili većeg kapaciteta od izvornog.

Prvo treba pripremiti ciljni disk da odgovara izvornom, tj. prekopirati master boot record i partcijsku tabelu te formatirati particije.

Stvaranje mbr datoteke **na izvornom računalu**.

```
dd if=/dev/sda of=sda.mbr bs=512 count=1
```

Zatim spremimo partcijsku tabelu kao tekstualnu datoteku:

```
sfdisk -d /dev/sda > sda.pt
```

Provjera (naredbe koje utipkavate su istaknute podebljanim slovima):

file sda.mbr

```
sda.mbr: x86 boot sector, Linux i386 boot LOader; partition 1: ID=0x82, starthead 1, startsector 63, 2104452 sectors; partition 2: ID=0x83, active, starthead 0, startsector 2104515, 771120 sectors; partition 3: ID=0x83, starthead 0, startsector 2875635, 16787925 sectors; partition 4: ID=0xf, starthead 254, startsector 19663560, 121885155 sectors
```

file sda.pt

```
sda.pt: ASCII textcat sda.pt
# partition table of /dev/sda
```

```
unit: sectors
/dev/sda1 : start=      63, size=  2104452, Id=83
/dev/sda2 : start=  2104515, size=   771120, Id=83, bootable
/dev/sda3 : start=  2875635, size= 16787925, Id=83
/dev/sda4 : start= 19663560, size=121885155, Id= f
/dev/sda5 : start= 19663623, size= 10490382, Id=83
/dev/sda6 : start= 30154068, size=  2088387, Id=83
/dev/sda7 : start= 32242518, size=109306197, Id=83
```

df

| Filesystem | 1K-blocks | Used | Available | Use% | Mounted on |
|------------|-----------|----------|-----------|------|--------------|
| /dev/sda2 | 373343 | 270608 | 83457 | 77% | / |
| tmpfs | 1038068 | 0 | 1038068 | 0% | /lib/init/rw |
| udev | 10240 | 628 | 9612 | 7% | /dev |
| tmpfs | 1038068 | 4 | 1038064 | 1% | /dev/shm |
| /dev/sda3 | 8262068 | 5834976 | 2007396 | 75% | /var |
| /dev/sda5 | 5162796 | 1576628 | 3323912 | 33% | /usr |
| /dev/sda6 | 1027768 | 16540 | 959020 | 2% | /tmp |
| /dev/sda7 | 53795484 | 13393200 | 40402284 | 25% | /home |

Prekopiramo MBR datoteku **na ciljni disk**:

```
dd if=sda.mbr of=/dev/sdc bs=512 count=1
```

Na ciljni disk možemo prenijeti partijsku tabelu, u kojoj je moguće i ručno preinačiti po potrebi veličine particija:

```
sfdisk /dev/sdc < sda.pt
```

Možemo formatirati nove particije u grafičkom alatu *gparted* ili iz komandne linije:

```
mkswap /dev/sdc1
mkfs.ext3 /dev/sdc2
mkfs.ext3 /dev/sdc3
mkfs.ext3 /dev/sdc5
mkfs.ext3 /dev/sdc6
mkfs.ext3 /dev/sdc7
```

S alatom *gparted* mogu se eventualno proširiti pojedine particije ako je ciljni disk većeg kapaciteta od izvornog.

Time je priprema zamjenskog diska gotova. Preostaje izvršiti kopiranje sadržaja izvornog na ciljni disk, najbolje s programom **rsync**.

Ako je ciljni disk spojen lokalno na izvorno računalo, sinkronizacija se može izvršiti priloženom skriptom **rsync_lokal.sh**, koja sinkronizira sadržaj particija ciljnog diska **/dev/sdc** sa onima izvornog diska **/dev/sda** (koje diskove imate priključene na sustav, najbolje se vidi (uz *root* ovlasti !)
naredbom:

```
fdisk -l
```

pa odgovarajuće prilagodite priložene skripte za lokalno odnosno udaljeno sinkroniziranje). *)

Ako ciljni disk nije spojen izravno na izvorno računalo, nego na neko drugo, onda treba napraviti

pripremu, da se s računala na kojem je spojen ciljni disk može pristupiti izvornom računalu bez prijave s kor. imenom i lozinkom (ovo je pogotovo važno ako na izvorno računalo nije dopušteno spajati se kao root korisnik s nekog drugog računala).

Na izvornom i ciljnom računalu (na koji je spojen ciljni disk) korisnik treba imati isto korisničko ime i sudo ovlasti za pokretanje programa *rsync*, drugim riječima u */etc/sudoers* na oba računala treba imati unos poput ovoga

```
korisn_ime ALL=(root) NOPASSWD:/usr/bin/rsync
```

Na ciljnom dodatno još i

```
korisn_ime ALL=(root) NOPASSWD:/bin/mount
korisn_ime ALL=(root) NOPASSWD:/bin/umount
```

Gornji unosi */usr/bin/rsync*, */bin/mount*, */bin/umount* ovise o tome koju putanju imaju dotični programi, to treba pogledati naredbom *'which'* i na izvornom i na ciljnom računalu, npr:

```
$ which rsync
/usr/bin/rsync
```

Prekontrolirajmo da li korisnik na oba računala ima bash ljusku kao podrazumijevanu, tj da na oba računala ima u */etc/passwd* datoteki stavku poput ove:

```
korisn_ime:x:1004:1004::/home/rsync:/bin/bash
```

Korisnik na ciljnom računalu treba načiniti **rsa ključ** naredbom:

```
ssh-keygen
```

(u ovom primjeru je na sva pitanja kod izvršenja naredbe odgovarano s Enter)

Kopirati stvoreni ključ **na izvorno računalo** (tražit će se lozinka korisnika na tom (=izvornom) računalu):

```
ssh-copy-id -i ~/.ssh/id_rsa.pub host.ustanova.hr
```

(umjesto *host.ustanova.hr* unijeti pravo ime **izvornog** računala)

I sada korisnik sa sudo ovlastima može na ciljnom računalu pokrenuti skriptu ***rsync_remote.sh*** za sinkronizaciju ciljnog diska s izvornim:

```
./rsync_remote.sh
```

(Ukoliko pri izvršavanju skripte *rsync_remote.sh* sistem ipak traži ukucavanje lozinke, rješenje koje funkcionira je da se skriveni **.ssh** direktorij **root** korisnika **na ciljnom računalu** zamijeni tj. nadomjesti onim od korisnika koji izvršava skriptu na tom istom računalu, možda je dovoljno

postaviti i *softlink* - ključne datoteke su *known_hosts* i *id_rsa*)

Ovaj način sinkronizacije čini mi se prilično sigurnim, jer je na izvornom računalu korisniku dovoljno da kao superkorisnik može pokrenuti samo naredbu ***rsync***, a na ciljnom samo naredbe ***rsync***, ***mount*** i ***umount***.

Prva sinkronizacija traje nekoliko sati, ali sve kasnije u pravilu traju samo nekoliko minuta.***) Radi toga se skripta može pozivati iz *crona* npr. svaki sat, a može i samo svaki dan, što je stvar osobne procjene. Za pokretanje iz *crona*, kao korisnik na ciljnom računalu upišete:

```
crontab -e
```

i u vašem podrazumjevanom editoru otvorit će se cron datoteka u koju upišete nešto poput:

```
15 3 * * * ~/rsync_remote.sh &
```

za skriptu *rsync_remote.sh*, ako se nalazi u vašem matičnom direktoriju, pa će se sinkronizacija odvijati jednom dnevno noću u tri sata i petnaest minuta, pod uvjetom da su i izvorno i ciljno računalo uključeni i spojeni na mrežu.

Nakon prve sinkronizacije, potrebno je mountati **root (/)** particiju ciljnog diska ako nije već mountana npr:

```
mount /dev/sdc2 /mnt/sdc2
```

i napraviti na njoj dodatne nužno potrebne direktorije *proc*, *sys*, *tmp*, *var*, *dev*, *usr*, *mnt*, *home* ...

```
mkdir /mnt/sdc2/proc /mnt/sdc2/sys /mnt/sdc2/tmp /mnt/sdc2/var /mnt/sdc2/dev /mnt/sdc2/usr /mnt/sdc2/mnt /mnt/sdc2/home
```

tj sve direktorije koji se naredbom ***--exclude*** u *rsync_skripti* isključuju iz sinkronizacije.

Preostaje još učiniti ciljni disk 'bootabilnim'. To ovisi koja je verzija *grub boot loader-a*

Instalacija stare verziju grub-a, naredbe koje utipkavate istaknute su podebljanim slovima:

```
grub
grub> find /boot/grub/stage1
(hd0,0)
(hd2,1)
grub> root (hd2,1)
grub> setup (hd2)
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/e2fs_stage1_5" exists... yes
Running "embed /boot/grub/e2fs_stage1_5 (hd2)"... 17 sectors are embedded.
succeeded
Running "install /boot/grub/stage1 (hd2) (hd2)1+17 p (hd2,1)/boot/grub/stage2 /boot/grub/menu.lst"... succeeded
Done.
grub> quit
```

Inače, što se tiče novije verzije (grub2) podešavanja, moguće ga je podesiti i iz 'chroot' okoliša, ako se sistem pokrene s nekog butabilnog CD-a.

Pokrene se terminalni prozor i otipkaju se primjerice ovakve naredbe:

```
sudo fdisk -l
```

ili

```
fdisk -l
```

(tj. bez 'sudo' uz 'root' ovlasti.) da se vidi koje su particije u igri, pa onda teba ući u "chroot" okoliš naredbama, npr:

```
sudo mount /dev/sda2 /mnt #ako je na ciljnom disku '/' (root) particija /dev/sda2
sudo mount --bind /dev /mnt/dev
sudo mount --bind /proc /mnt/proc
```

```
sudo chroot /mnt
```

Treba montirati sada /usr particiju na ciljnom disku u *chroot* okolištu da bi imali pristup izvršnim datotekama, kao što je *grub-install* na putanji /usr/sbin/grub-install. (ovisno što želite raditi, montirati trebate i druge particije u *chroot* okolištu, kao što su /var, /home ...), za instalaciju *gruba* /var particija je svakako nužna:

```
mount /dev/sda5 /usr
```

```
grub-install --recheck /dev/sda
grub-mkconfig
update-grub
```

Možete testirati da li je *grub* instaliran ako pokrenete slijedeću naredbu **prije** i **poslije** instalacije *grub-a*:

```
dd if=/dev/sda bs=512 count=1 | xxd
```

pa ako negdje u ispisu možete pročitati riječ "GRUB" nakon instalacije, valjda je *grub* instaliran. Može se testirati i s **qemu** emulatorom, naredbom:

```
qemu /dev/sda
```

tj. podići linux s ciljnog diska kao virtualni OS, no to uradite bez utaknutog mrežnog kabla ili unutar lokalne mreže.

Nakon umountanja /usr i drugih particija koje ste montirali u *chroot* okolišu izlaz je:

exit

I to je sve. Ciljni disk ćemo sada, u slučaju havarije produkcijskog servera, ugraditi kao primarni disk u zamjenski poslužitelj. Za probu možemo pokušati podići zamjenski poslužitelj (ali bez mrežnog kabela!!!) i ako je hardverska konfiguracija zamjenskog poslužitelja slična konfiguraciji originalnog poslužitelja, odnosno ako se uspiju učitati kernel moduli za hardver zamjenskog poslužitelja, onda se ciljno računalo može upotrijebiti kao nadomjestak za originalni poslužitelj. Zamjenski sistem će biti u onom stanju u kojem je bio originalni sistem u vrijeme posljednje uspješne sinkronizacije.***)

Treba također obratiti pažnju na to da prilikom aktiviranja zamjenskog poslužitelja ključne servise u direktoriju `/etc/init.d` kao što su `postfix`, `ldap`, `squirrelmail` itd treba inicijalno onemogućiti PRIJE NEGO ŠTO SE UTAKNE MREŽNI KABEL, te ih JEDAN PO JEDAN uključivati u rad, te se preko logova i prikladnim testovima osvjedočiti da rade ispravno.






Update 16. 09. 2013:

Članku priložene tri pdf datoteke (engl. tekst) s uputama za migraciju linux sistema s računala na računalo.

*) Priložene skripte neće pravilno raditi, ako u njima predhodno nistu prilagođene "točke montiranja" i na sistemskom disku načinjeni odgovarajući direktoriji (`mkdir /mnt/sdc2 /mnt/sdc3 ...` ili sl.) shodno ispisu particija, koji se za ciljni disk pokaže nakon izvršenja naredbe: `sudo fdisk -l`)

) Trajanje sinkroniziranja **log datoteka može se znatno skratiti, ako se podesi da završeci log datoteka nose datume umjesto brojeva (ključna riječ `'dateext'`), isto tako sinkroniziranje je moguće raditi inkrementalno, tako da u arhivi ostaju valjane kopije stanja poslužitelja prema proizvoljno odabranim datumima (ključna riječ `'rsnapshot'`). Ove teme su predmet možebitnih idućih članaka.

***) Članak pruža načelnu orijentaciju kako načiniti vjernu kopiju Linux sistema u produkciji i ne obrađuje situacije u kojima bi eventualno trebalo raditi dodatne prilagodbe, kao što su podešavanje raid sustava, ugrađivanje raznih modula za poneki hardver (ako ih prekopirani kernel ne uspije učitati na zamjenskom računalu), možebitno dodatno podešavanje mrežnih postavki (ključna riječ `'ifconfig'`) i sl.

| Prilog | Veličina |
|--|----------|
|  rsync_lokal.sh_.txt [1] | 1.65 KB |
|  rsync_remote.sh_.txt [2] | 1.75 KB |
|  Linux_migration_preparing_server.pdf [3] | 52.23 KB |
|  Linux_migration_running_sync.pdf [4] | 65.34 KB |
|  Linux_migration_tips_and_tricks.pdf [5] | 81.15 KB |

pon, 2012-06-18 11:36 - Luka ČavaraKuharice: [Linux](#) [6]

Kategorije: [Sigurnost](#) [7]

Vote: 5

Vaša ocjena: Nema Average: 5 (4 votes)

Source URL: <https://sysportal.carnet.hr/node/1037>

Links

- [1] https://sysportal.carnet.hr/system/files/rsync_lokal.sh_.txt
- [2] https://sysportal.carnet.hr/system/files/rsync_remote.sh_.txt
- [3] https://sysportal.carnet.hr/system/files/Linux_migration_preparing_server.pdf
- [4] https://sysportal.carnet.hr/system/files/Linux_migration_running_sync.pdf
- [5] https://sysportal.carnet.hr/system/files/Linux_migration_tips_and_tricks.pdf
- [6] <https://sysportal.carnet.hr/taxonomy/term/17>
- [7] <https://sysportal.carnet.hr/taxonomy/term/30>