LVM-esim: systeemilevyn vaihto

Vaihdettava systeemilevy, jolla on sekä LVM:n ulkopuolinen partitio 1 (/dev/sda1, mountattuna /boot) että PV (/dev/sda2), boot loader (grub) master boot recordissa

- Partitioidaan uusi levy (sdb) käsin niin, että ensimmäinen (sdb1) on ainakin yhtä iso kuin ennenkin, lopusta sdb2, sitten partx/kpartx/partprobe tai boot
- Mountataan /boot varmuuden vuoksi readonly-tilaan:

```
mount -o remount,ro /boot
```

LVM-esim: systeemilevyn vaihto

- Kopioidaan /boot dd:llä:

  dd if=/dev/sda1 of=/dev/sdb1

  umount /boot; mount /dev/sdb1 /boot # tarkista että OK

  resize2fs /dev/sdb1 # jos kokoa suurennettiin

- Asennetaan boot loader uudelle levylle:

  grub-install /dev/sdb

LVM-esim: systeemilevyn vaihto

- LVM-osan vaihto pvmove'lla kuten edellä mutta laitteina PV-partitiot /dev/sda2 ja /dev/sdb2
- Poistetaan vanha levy
- Bootataan mahdollisimman pian varmuuden vuoksi - saattaa vaatia BIOSin tai VM:n määritysten muuttamista, mahdollisesti update-initramfs -u, /etc/fstab (jos levykirjainviittauksia tai UUID vaihtui)

LVM: asennus

- Olemassaolevan koneen muuttaminen LVM:ää käyttämään ei oikein onnistu kuin osittain, yleensä helpompaa asentaa uudestaan.
- Asennettaessa kannattaa jättää levyä vapaaksi koska kaiken suurentaminen on helpompaa kuin pienentäminen.
- Tarjolla olevat automaatti-LVM -vaihtoehdot harvoin toimivat sellaisinaan, osiointia kannattaa tehdä jo asennusvaiheessa siltä osin kuin tietää tarvitsevansa, vaikka muutoksia onkin helpohkoa tehdä myöhemminkin.

date

- date # tämänhetkinen kellonaika oletusformaatissa
- date '+%d.%m.%Y' # päivä.kuukausi.vuosi
- date '+%H:%M:%S' # kellonaika hh:mm:ss
- date '+%a' # viikonpäivä 3-kirjaimisena lyhenteenä (vrt locale)
- date '+%u' # viikonpäivä numerona 1-7 (1=ma)
- date -**I**[hours|minutes|seconds|ns|date] # ISO-formaatti
- date -d 'yesterday - 1 day' # toissapäivä
- TZ='America/Los_Angeles' date # aika LA:ssa; vrt tzselect
- date -u # UTC-aika
- date -r *file* # tiedoston muutosaika
- paljon muitakin, ks. man date, info date

cron: /etc/crontab

#min hour day month weekday user command(s)

- joka päivä kello 03:17:

  17 3 * * * root komento1 && komento2

- joka sunnuntai kello 7:15 ja 17:15:

  15 7,17 * * 0 root komento

  15 7,17 * * 7 root komento

cron: /etc/crontab

- joka kolmas tasatunti:

  0 */3 * * * root komento
- 15 minuuttia yli parillisten tuntien 8-16 arkisin:

  15 8-16/2 * * 1-5 root komento
- Helmikuussa joka päivä kerran minuutissa 08:00-08:59:

  * 8 * 2 * root komento
- joka sunnuntai **ja** joka kuukauden 15. päivä kello 10:

  0 10 15 * 7 root komento

cron: /etc/crontab

- Joka kuukauden ensimmäinen päivä kello 7:

  0-7 1 * * * root komento
- Joka kuukauden ensimmäinen sunnuntai, eri tapoja:

  0 7 1-7 * * root [$(date '+%u') = 7] && komento

  0 7 * * 7 root if [$(date '+%d') -lt 8]; then komento; fi

  0 7 * * 7 root [$(date '+%d') -ge 8] || komento

  Tämä siis **ei** toimi oikein:

  0 7 1-7 * 7 root komento

cron: /etc/crontab

- joka kuukauden viimeinen päivä, erimittaisille kuukausille erikseen:

  0 7 31 * * root komento

  0 7 30 4,6,9,11 * root komento

  0 7 29 2 * root komento

  0 7 28 2 * root date -d $(date '+%Y')-02-29 2>/dev/null || komento

  viimeisessä date -d aiheuttaa virheen ellei ole karkausvuosi, virheilmoitus hävitetään ohjaamalla stderr /dev/null'iin
- joka kuukauden viimeinen päivä, toinen tapa:

  0 7 28-31 * * root [$(date -d tomorrow '+%m') = $(date '+%m')] || komento

/etc/cron.d, user crontabs

- /etc/cron.d/*: syntaksi kuten /etc/crontab (ja yleensä suoraan /etc/crontab'iin ei laiteta juuri mitään)
- käyttäjän crontab: muuten kuten /etc/crontab mutta käyttäjäkenttä puuttuu

  asetetaan komennolla crontab t*iedosto*

  tiedosto konventionaalisesti $HOME/.crontab
- sisällön voi tarkistaa komennolla crontab -*l*
- /var/spool/cron/crontabs/… (ei pidä muokata suoraan)
- huom. ympäristö (PATH, SHELL) eri kuin yleensä

/etc/cron.*, crontab.{allow,deny}

- /etc/cron.{hourly,daily,weekly,monthly}

  suoritettavat komennot (skriptit) sellaisinaan, tarkka suoritusajankohta voi vaihdella (ks. /etc/crontab)

- voidaan rajoittaa keille crontab sallittu:

  /etc/cron.{allow,deny}

  - jos molemmat olemassa cron.allow voittaa

anacron

- koneissa jotka eivät ole aina päällä (läppärit…)
- komentojen suoritusajankohtaa ei määritellä tarkasti, vain kuinka monen päivän välein ne pitäisi suorittaa, tai @monthly "kerran kuussa"
- muistaa milloin komento on viimeksi ajettu ja ajaa uudestaan jos annettu määrä päiviä on mennyt
- jos asennettuna, hoitaa yleensä hakemistoissa /etc/cron.{daily,weekly,monthly} olevat komennot
- /etc/anacrontab
- Ei yleensä kaipaa säätämistä

at, batch

- echo komento | at aika

  at -f file aika

  - aika voidaan antaa monella tavalla, esim. ”08:15”, ”midnight”, ”noon tomorrow”, ”now +3 weeks”...
- atq, atrm
- /etc/at.{allow,deny} # kuten cron.{allow,deny}
- echo komento | batch # aja kun kuorma sallii (load<1.5)

/home'n pienentäminen

/home-osion pienentäminen ei onnistu kun admin-tunnus (esim. tt0) käyttää sitä:

```
sudo lvresize -r -L500M /dev/tt1-vg/lvhome
Do you want to unmount "/home"? [Y|n] y
umount: /home: target is busy
```

Tarkistus:

```
lsof /home
ps -fu tt0
```

Ongelma on kierrettävissä monella tavalla.

/home'n pienentäminen

- Vaihdetaan tt0:n kotihakemisto:

  cp -a /home/tt0 /; usermod -d /tt0 tt0

  # kirjaudutaan ulos ja takaisin

  sudo lvresize -r -L 500M /dev/tt1-vg/lvhome

- Asetetaan rootille salasana:

  sudo passwd root # vaikea!

  # kirjaudutaan konsolilta roottina ja lvresize …

/home'n pienentäminen

- Avataan ssh suoraan rootille:

  mkdir /root/.ssh

  cat /jossain/id_rsa.pub>>/root/.ssh/authorized_keys

  exit # ja takaisin ssh root@kone

  grep RootLogin /etc/ssh/sshd_config # muuta tarvittaessa

  lvresize …

- Tehdään koonmuutos ajastetusti:

  echo 'lvresize …' | sudo at 'now + 1 minute'

  # kirjaudutaan ulos ja odotetaan minuutti

/jotain pienentäminen

- Tiedostojärjestelmän pienentäminen edellyttää siis sen poistamista käytöstä (umount), ja se taas ettei mikään prosessi käytä sitä
- /usr, /var pienennettävissä single-user tilassa
- /:n pienentäminen ei yleensä onnistu koneen sisältä lainkaan; onnistuu pysäyttämällä kone ja mounttaamalla levy toiseen koneeseen, tai boottamalla kone cd:ltä tms

/jotain pienentäminen

- Jos -r -optio lvresize:stä pienennettäessä unohtuu, tiedostojärjestelmä todennäköisesti tuhoutuu saman tien. Vaihtoehto on tehdä resize2fs *ensin,* mutta silloin pitää osata laskea käsin oikea koko (ja muistaa fsck).
- Kaikkia tiedostojärjestelmätyyppejä ei voi pienentää yhtä helposti. Esim. xfs:ää tai jfs:ää ei voi pienentää lainkaan: ainoa mahdollisuus on luoda kokonaan uusi tiedostojärjestelmä ja kopioida data sinne.

/usr:n pienennys kopioimalla

- /usr:n pienentäminen kopioimalla (jos se on xfs tms) onnistuu helpoiten single-user -tilassa, mutta sen voi tehdä muutenkin bootin kautta:

```
lcvreate -n lv2usr -L 1G kone-vg
mkfs … /dev/kone-vg/lv2usr  # optiot tilanteen mukaan
mkdir /newusr; mount /dev/kone-vg/lv2usr /newusr
mount -o remount,ro /usr # varmuuden vuoksi
cp -a /usr/* /newusr
sed 's-lvusr-lv2usr-g' /etc/fstab
shutdown -r now
lvremove /dev/kone-vg/lvusr; rmdir /newusr
```

Swapin pienentäminen

- Heittovaihtotiedoston eli swapin pienentäminen on helppoa, ellei muisti ole niin vähissä että se on käytössä:

```
swapoff /dev/tt1-vg/lvswap1
lvresize -L200M /dev/tt1-vg/lvswap1
mkswap -U ... /dev/tt1-vg/lvswap1 # UUID /etc/fstab'ista
swapon -a
```

- Swappia ei useinkaan tarvita lainkaan ja sen voi poistaa kokonaankin, jos koneessa on muuten tarpeeksi muistia

Admin-tunnuksesta

- Sudo-oikeudelliselle admin-tunnukselle (*tunnus*0) kannattaa tehdä erikoissäätöjä:
  - Kotihakemisto pois /home'n alta (/adminhome/tunnus0 tai /home0/tunnus0 tai vain /tunnus0)
  - UID pienemmäksi ettei Kerberos puutu siihen:

    usermod -u 999 tt0 # jotain < 1000

    - vaihtoehtoisesti kerberos-raja 1001:een tms, ks. /etc/pam.d/common-password
  - Riippuvuuksia lisäpaketeista syytä välttää (shelliä ei kannata vaihtaa zsh:ksi tms)

Asennusongelmia: lokit

Asennuksen keskeytyessä outoon virheeseen:

- grep -i error /var/log/syslog |more
- grep -i “no space” /var/log/syslog
- grep -i failed /var/log/syslog
- dmesg | more

Asennusongelmia: lokit

Kannattaa etsiä lisätietoa grepillä löydetyn virheen ympäriltä; jos (vanhan) busyboxin grep ei tunne -A ja -B optioita eikä sen more /-hakua, usein joutuu greppaamaan aikaleimaa tms

Asennusongelmia: levytila

- Levytilan täyttyminen voi aiheuttaa hyvin monenlaisia virheitä ja aiheutua satunnaisista tekijöistä (käytetty repository, vaihtelu verkon nopeudessa, jopa koneen nimi – pitempi nimi tuottaa enemmän tekstiä lokiin!)

Asennusongelmia: levytila

- Asennuksen keskeytyessä täyttynyt levy ei ehkä enää olekaan täynnä (erityisesti /var), virheilmoitusta kannattaa etsiä lokista
- Jos jokin levyosio täyttyy, asennus on yleensä aloitettava alusta tai ainakin levyosioinnista; joskus loogisen volumen laajennus “lennosta” riittää

Asennusongelmia: RAM

- Keskusmuistin loppuminen kesken asennuksen saattaa aiheuttaa outoja virheitä.
- Jos muistia ei voi lisätä, joskus auttaa kun asentaa vähemmän paketteja (ja toivoo niiden asentamisen onnistuvan myöhemmin).
- Graafinen konsoli syö enemmän muistia kuin sarjaportti (--graphics none säästää hieman).

Disaster recovery: /usr lost

Rikotaan /usr (oletetaan ettei se ole oma tiedostojärjestelmänsä):

sudo mv /usr /oldusr

sudo ei enää toimi!

Disaster recovery: /usr lost

- Pelastus jos on sudo alustakoneessa:

  virsh destroy tt1 # shutdown ei toimine

  losetup -f # palauttaa (esim.) /dev/loop1

  losetup /dev/loop1 ~tt/tt1.img

Disaster recovery: /usr lost

kpartx -a /dev/loop1

mkdir -p /mnt/tmp

mount /dev/mapper/loop1p1 /mnt/tmp

mv /mnt/tmp/oldusr /mnt/tmp/usr

Disaster recovery: /usr lost

umount /mnt/tmp

kpartx -d /dev/loop1

losetup -d /dev/loop1

virsh start tt1

/usr lost, ratkaisu 2

- Ei roottia alustakoneessa, käytetään toista virtuaalikonetta apuna:

  virsh destroy tt1

  virsh attach-disk tt2 ~/tt1.img vdb

    --driver qemu --subdriver qcow2 # jos qcow2

/usr lost, ratkaisu 2

- tt2:ssa (tarvittaessa ensin reboot, jolloin myös --persistent yllä):

  mkdir -p /mnt/tmp

  fsck /dev/vbd1

  mount /dev/vdb1 /mnt/tmp

  mv /mnt/tmp/oldusr /mnt/tmp/usr

  umount /mnt/tmp

/usr lost, ratkaisu 2

- Lopuksi taas alustakoneessa:

  virsh detach-disk tt2 vdb

  virsh start tt1

/usr lost, ratkaisu 3

Käynnistetään virtuaalikone recovery-tilaan:

- virsh destroy tt1

  virt-viewer --wait tt1 &

  virsh start tt1
- nopeasti grub-menu kiinni, recovery mode

/usr lost, ratkaisu 3

- recovery-boot valikosta ”root shell prompt”

  mv /oldusr /usr; exit
- resume normal boot
- Edellyttää graafista konsolia ja nopeaa yhteyttä ja nopeita refleksejä, ellei grubia ole säädetty hitaammalle

/usr lost, ratkaisu 3b

- Recovery-tila sarjaporttikonsolilla, grub on määritetty käyttämään sarjaporttia (/etc/default/grub):

  virsh destroy tt1

  eri ikkunoissa nopeasti peräkkäin:

  virsh start tt1 virsh console tt1

  konsoli-ikkunassa recovery boot kuten edellä &c

/usr lost, ratkaisu 4

Bootataan virtuaalikone CD-imagelta (tai oikealta CD:ltä) root-oikeuksin alustakoneessa:

virsh destroy tt1

sudo kvm -name tt1 -m 256 -hda ~tt/tt1.img -cdrom /srv/ftp/iso/ubuntu...iso -boot d

valitaan ”rescue broken system”…

/usr lost, ratkaisu 4

- CD-imagen pitää tässä olla sopiva - erityisesti jos VM:ssä ei ole graafista konsolia, normaalit asennus-CD:t eivät toimi.
- Huom. kvm-komennon optiot erilaisia kuin virt-install jne (ks. man qemu-system)

/usr lost, ratkaisu 5

- Bootataan CD-imagelta ilman root-oikeuksia alustakoneessa:

  virsh destroy tt1

  virsh dumpxml tt1 >tt1.xml

/usr lost, ratkaisu 5

virsh edit tt1

- lisätään cd-rom (image):

```
<disk type='file' device='cdrom'>
  <driver name='qemu' type='raw'>
  <source file='/srv/kvm/images/ubuntuX.img'>
  <target dev='hda' bus='ide'>
  <readonly>
  <address type='drive' controller=...>
</disk>
```

/usr lost, ratkaisu 5

- vaihdetaan boottilaitteeksi cd:

```
<os>
...
<boot dev='cdrom'/>
</os>
```

/usr lost, ratkaisu 5

virsh start tt1 # rescue broken system...

virsh shutdown tt1

virsh undefine tt1

virsh define tt1.xml

inodes

- Jos kone väittää levyn olevan täynnä vaikka siellä df:n mielestä on tilaa, syy voi olla että inodet ovat lopussa. Erityisesti ext* -tiedostojärjestelmissä niiden suhde levytilaan kiinnitetään tiedostojärjestelmää luotaessa eikä sitä voi kasvattaa. Tilanteen voi tarkistaa **df -i**:llä (tai **df --inodes**).
- Inode on olennaisesti osoitin tiedostoon, niiden (käytössä olevien) määrä on suunnilleen sama kuin tiedostojen määrä.

inodes

- Inode-määrään voi vaikuttaa luontivaiheessa mkfs:n optiolla -i, bytes-per-inode (~ tiedostojen keskimääräinen koko). Oletus ext4:ssä on 16384; jos tiedossa on, että pieniä tiedostoja tulee paljon, voi käyttää pienempää arvoa, ja päinvastoin. Jälkikäteen inode-suhdetta ei ext*-tiedostojärjestelmissä voi muuttaa; tiedostojärjestelmää kasvatettaessa niitä tulee lisää alkuperäisessä suhteessa.
- Asennettaessa inodeja saa lisää valinnalla ”Typical use: news”.

inodes

- Etenkin /usr:ään tulee usein paljon pieniä tiedostoja, jolloin inodet saattavat loppua. Koska niiden suhteellista määrää ei voi lisätä vaikka tiedostojärjestelmää kasvattaa, ratkaisuksi jää joko koko tiedostojärjestelmän uudelleenluonti ja tiedostojen siirtäminen (vaikeaa /usr:lle) tai tiedostojärjestelmän jakaminen osiin (tyypillisesti /usr/src ja/tai /usr/share erilleen).

inodes

- Esimerkki: selvitetään missä /usr:n alihakemistoissa inodeja kuluu eli tiedostoja on paljon:

  ```
  for d in /usr/*; do; echo -n ”$d: ”; find $d | wc -l; done | sort -nr -k2

  # tai, jos kyllin uusi versio du:sta:

  du -s --inodes /usr/*  | sort -nr
  ```

  Suurimmat inode-syöpöt voi sitten siirtää omiksi tiedostojärjestelmikseen (ja varata niihin enemmän inodeja, esim. mkfs -i 2048 ...).

inodes

- Eri tiedostojärjestelmät käsittelevät inodeja eri tavoin:
  - xfs luo inodeja dynaamisesti eivätkä ne yleensä lopu. Jos niille varattu tila kuitenkin loppuu, sitä voi lisätä komennolla xfs_growfs -m *p,* missä *p* on inode-maksiprosenttiosuus koko levytilasta.
  - jfs ja zfs luovat myös inodeja dynaamisesti tarpeen mukaan, eivätkä ne lopu kesken ellei koko levy täyty.

fsck

- ”File System Check”: tarkistaa tiedostojärjestelmän sisäisen rakenteen (rikkinäiset osoittimet jne)
- Tehdään yleensä automaattisesti bootissa /etc/fstab'in määräämässä järjestyksessä, mutta joskus tehtävä käsin
- Eri tiedostojärjestelmätyypeille omansa (fsck.ext4, fsck.xfs jne, kokeile ls -li /sbin/*fsck*), ”fsck” on wrapperi, joka valitsee mikä niistä ajetaan. Single user tilassa wrapper saattaa puuttua ja pitää käyttää suoraan e2fsck tms.
- Kerää orvot (nimettömät) tiedostot hakemistoon lost+found

fsck

- Yleisiä optioita:
  - -A noudata /etc/fstab'in fsck-kenttää (viimeinen)
  - -R yhdessä -A:n kanssa: jätä root (/) tarkistamatta
  - -p yritä korjata viat automaattisesti (ext*)
  - -f ”force”, testaa vaikka ei näytä tarpeelliselta (ext*)
  - -y vastaa kaikkiin kysymyksiin ”yes”
- Joillekin tiedostojärjestelmille omia testaus- ja korjauskomentojaan, esim. xfs_check, xfs_repair

debugfs &c

- Jos tiedostojärjestelmä on niin sekaisin, ettei fsck (tai xfs_repair jne) saa sitä korjattua, sitä voi yrittää korjata ”käsin” tiedostojärjestelmäkohtaisella editorilla (debuggerilla), mm:
  - debugfs ext* -järjestelmille
  - xfs_db xfs:lle
  - jfs_debugfs jfs:lle
- Hyvin työlästä ja virhealtista, edellyttää tiedostojärjestelmän sisäisen rakenteen tuntemista - ei yleensä kannata kuin äärimmäisessä hädässä