Varmuuskopiointi: image

- Sammuta virtuaalikone ensin - elävästä koneesta otettu kopio ei ole koherentti - ja:

  cp kone1.img kone1.img.bak
- Voidaan myös siirtää toiseen alustakoneeseen (xml-tiedosto siirrettävä myös, virsh dumpxml...)

Varmuuskopiointi: image

- Tiedostojen omistaja muuttuu kun virtuaalikone käynnistetään, palautus esim.

  rm -f kone1.img; cp kone1.img.bak kone1.img

  tai

  cp **-f** kone1.img.bak kone1.img

Varmuuskopiointi: image

- ei säännöllinen varmuuskopiointitapa, vain poikkeustilanteissa, levyimagea siirrettäessä tai ennen versiopäivitystä tai jotain ”tämä saattaa rikkoa kaiken” -säätöä
- polun vaihtuessa muutettava xml-tiedostoa (virsh edit, tai virsh dumpxml...define)

Muistin kopiointi: virsh save

- Joskus halutaan kopio käynnissä olevan koneen tilasta (RAM &c), sen voi tehdä näin:

  virsh save kone tiedosto
- Kone ”hibernoi”, jatkaa komennolla

  virsh restore tiedosto

Muistin kopiointi: virsh save

- Samasta tiedostosta voi palauttaa monta kertaa ”samaan ajanhetkeen”; levyimage on tällöin talletettava ja palautettava erikseen
- Joskus kätevä ennen vaarallisia kokeiluja
- Voi vaihtaa alustakonetta, mutta käyttöjärjestelmän ja kirjastojen oltava kyllin samanlaisia kuin talletettaessa

command expansion, eval

- Komennon tulos merkkijonoon:

  year=$(date +%Y); month=$(date +%m)

  day=`date +%d` # vanha tapa, ei suositeltava

  files=$(ls -l $(grep -l kala *.txt))

command expansion, eval

- Merkkijono komennoksi:

  eval $(date '+year=%Y month=%m day=%d')
- x=ls; $x # toimii

  x='y=z'; $x # ei toimi

  x='y=z'; eval $x; echo $y # toimii

loop device

Mountataan tiedosto kuin se olisi levy:

mount /tmp/disk.img /mnt/tmp -t ext4 -o loop=/dev/loop3

Käytetään tiedostoa kuin se olisi levy:

losetup -f /dev/loop0 file.img

(yleensä tehtävä roottina tai sudo ...)

loop device 2

Partitiot näkyviin:

kpartx -a /dev/loop0

Mountataan em. tiedoston 1. partitio:

mount /dev/mapper/loop0p1 /mnt/tmp

Vapautetaan:

umount /mnt/tmp; losetup -d /dev/loop0

Virtuaalilevyn suurentaminen 1

Edellyttää yleensä boottia, ehkä useampaakin. Vaarallinen!

Vaiheet:

(1) Suurennetaan levyimage

(2) Muutetaan partitiointia

(3) Suurennetaan tiedostojärjestelmä(t) ja swap

Virtuaalilevyn suurentaminen 2

virsh shutdown kone # jatkuu alustakoneessa

mv kone.img kone.img.bak

cp kone.img.bak kone.img

(ensin mv jotta saadaan kirjoitusoikeudet kohdalleen)

qemu-img resize kone.img +2GB

Virtuaalilevyn suurentaminen 3

virsh start kone # jatkuu virtuaalikoneessa

sudo fdisk /dev/vda # tai parted

- poistetaan kaikki partitiot ja luodaan uusiksi, root-partition alun pitää säilyä täsmälleen ennallaan (swap voi siirtyä, muut jos siirtää datankin), samoin tyyppien ja boot-lipun; voi myös lisätä uusia partitioita. Huom. vaikuttaa vasta bootin jälkeen!

Virtuaalilevyn suurentaminen 4

```
sudo shutdown -r now
    # swapin käyttöönoton epäonnistuminen hidastaa
    # boottia; nopeampaa jos poistaa swapin fstab'ista
# bootin jälkeen, jos kaikki meni hyvin:
sudo resize2fs /dev/vda1; df
grep swap /etc/fstab
# cut'n'paste UUID= ... edeltä tai:
eval $(awk '/UUID.*swap/{print $1}' /etc/fstab)
```

Virtuaalikoneen suurentaminen 5

sudo mkswap -U $UUID /dev/vda5

sudo swapon -a; cat /proc/swaps

shutdown -r now # varmuuden vuoksi

rm -f kone.img.bak # alustakoneessa

(kun on varmistettu että kaikki ok)

Virtuaalilevyn suurentaminen 6

Jos on root-oikeudet (sudo) alustakoneessa (tai välissä siirtää levyimagen koneeseen jossa root-oikeudet on), toinen tapa:

virsh shutdown kone

losetup -f # tulos (/dev/loop2 tms) talteen, käytetään alla

losetup -f /dev/loop2 kone.img

Virtuaalilevyn suurentaminen 7

kpartx -a /devloop2 # partitiot näkyviin

eval $(blkid /dev/mapper/loop2p5 | awk '{print $2}')

UUID talteen

kpartx -d /dev/loop2 # partitiot pois, muuttuvat alla

fdisk /dev/loop2 # partitioi, root-partition alku ennallaan

Virtuaalilevyn suurentaminen 8

kpartx -a /dev/loop2 # uudet partitiot näkyviin

e2fsck -f /dev/mapper/loop2p1 # varmuuden vuoksi

resize2fs /dev/mapper/loop2p1

mkswap -u $UUID /dev/mapper/loop2p5

swap UUID joka talletettiin edellä

Virtuaalilevyn suurentaminen 9

kpartx -d /dev/loop2

losetup -d /dev/loop2

virsh start kone

helpotuksen huokaus tai ...

Virtuaalilevyn suurentaminen 10

Jos levyä joutuu suurentamaan tällä tavoin, jotain on mennyt pieleen suunnitteluvaiheessa. Vaihtoehtona kannattaa harkita uudelleenasennusta.

Jos levyjen ja osioiden ennakoidaan muuttuvan, LVM on yleensä hyvä idea.

Asennus: konsolit

- Virtuaalikonsolia käytettäessä voi asennuslokia seurata reaaliajassa neljännessä konsoli-ikkunassa (Alt-F4 tai Ctrl-Alt-F4, graafisen virtuaalikonsolin kanssa Send key/Ctrl+Alt+F4) tai tutkia taaksepäin toisessa ja kolmannessa (Alt-F2, Alt-F3), esim.

  more /var/log/syslog.log

Asennus: konsolit

- Sarjaporttikonsolin kanssa käytettävissä ei ole useampia konsoli-ikkunoita, mutta asennusvalikossa on “Execute a shell”, johon joskus pääsee...
- Lokissa on yleensä aina virheeltä näyttäviä viestejä jotka eivät oikeasti ole vakavia; kannattaa tutkia onnistuneidenkin asennusten lokeja

Asennus: konsolit

- Konsoleita voi olla useita, erityisesti sekä sarjaporttikonsoli että (virtuaalinen) graafinen konsoli; --extra-args -optiolla (tai -x) annetuista console= -asetuksista viimeinen on ensisijainen (/dev/console); esim.

  -x 'console=tty0 console=/dev/ttyS0,115200n8 serial'

- -x (--extra-args) ei toimi --cdrom -option kanssa, mutta roottina voi tehdä --location /jotain/ubuntux.iso -x ...

Asennus: konsolit

- Jos virt-install -komennolle määrittelee sarjaporttikonsolin mutta ei '--graphics none', konsoliviestit saattavat kuitenkin mennä graafiseen virtuaalikonsoliin; jos X-yhteyttä ei ole, siihen voi ottaa yhteyden virt-viewer -komennolla toisesta ikkunasta; joskus taas voi olla tarpeen avata uusi sarjaporttikonsoli komennolla virsh console *kone*

Asennus: konsolit

- Toisen konsolin voi lisätä myös jälkikäteen (virsh edit...) ja se on yleensä parempi ratkaisu, asennus kahden konsolin kanssa käyttäytyy joskus oudosti. Sarjaporttikonsoli on helpompi lisätä jälkeenpäin kuin graafinen.

Asennus: konsolit

- Sarjaporttikonsolin määritys xml:ssä:

```
<serial type='pty'>
    <target type='serial' port='0'>
</serial>
<console type='pty'>
    <target type='serial' port='0'>
</console>
```

Asennus: konsolit

- Ubuntu 16.04:ssä (ainakin) sarjaporttikonsoli tulee xml:ään aina, se vain pitää aktivoida VM:ssä:

  systemctl enable serial-getty@ttyS0.service

  systemctl start serial-getty@ttyS0.service

- Em. säädöt saattaa joutua tekemään (ssh-yhteyden yli) vaikka asennusvaiheessa olisi jo käytetty sarjakonsolia

Asennus: konsolit

- Grub sarjaporttikonsolin kanssa: /etc/default/grub

  GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0"

  GRUB_TERMINAL=serial

  GRUB_SERIAL_COMMAND="serial --unit=0 --speed=115200 --word=8 --parity=no –stop=1"

  Kaikki “HIDDEN_TIMEOUT” -rivit kannattaa myös kommentoida pois ja GRUB_TIMEOUT -aikaa ehkä pidentää

- Editoinnin jälkeen update-grub

Asennus: konsolit

- Graafisen konsolin määritys xml:ssä (pci slot-osoitteita voi joutua vaihtamaan):

```
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0' state='disconnected'/>
  <alias name='channel0'/>
  <address type='virtio-serial' controller='0' bus='0' port='1'/>
</channel>
<input type='mouse' bus='ps2'/>
<input type='keyboard' bus='ps2'/>
<graphics type='spice' port='5905' autoport='yes' listen='127.0.0.1'>
  <listen type='address' address='127.0.0.1'/>
  <image compression='off'/>
</graphics>
```

Asennus: konsolit

```
<video>
  <model type='qxl' ram='65536' vram='65536' vgamem='16384' heads='1'/>
  <alias name='video0'/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
</video>
<redirdev bus='usb' type='spicevmc'>
  <alias name='redir0'/>
</redirdev>
<redirdev bus='usb' type='spicevmc'>
  <alias name='redir1'/>
</redirdev>
```

Asennus: konsolit

- Sarjaporttikonsoliin voi kytkeytyä uudesta ikkunasta:

  virsh console *kone*

  tai etänä alustakoneeseen *host* lisäämällä optio

  --connect qemu+ssh://*host*/system
- Yhteyden katkaisu ^] (suominäppiksellä Ctrl+AltGr+])

ssh

- ssh (Secure SHell) suorittaa komennon tai avaa pääteistunnon etäkoneessa:

  `ssh [options] [user@]kone [komento]`

- Paljon optioita erikoistilanteisiin, mm.

  -X salli etäkoneen käyttää paikallista näyttöä

  -p *port* käytä porttia *port* oletuksen 22 asemesta

  -i *id_file* käytä epästandardia avaintiedostoa

  (oletus ~/.ssh/id_{dsa,ecdsa,ed25519,rsa})

  -v verbose, debug-tulostus

scp

- scp (Secure CoPy) kopioi tiedostoja etäkoneeseen:

  `scp [`*`optiot`*`] `*`tiedosto`*` [`*`user`*`]@kone:[kohde]`

  tai etäkoneesta:

  `scp [`*`optiot`*`] [`*`user`*`]@kone:tiedosto kohde`

  `missä kohde` voi olla hakemisto tai tiedostonimi.

  Yleisin optio on -r, joka kopioi koko hakemistopuun. Optio -i toimii kuten ssh:n kanssa, mutta portin vaihto-optio on -P *port*, kun taas -p säilyttää aikaleimat ja oikeudet.

ssh & scp: autentikointi

- Autentikointitapoja on useita, yleisimmät:
  - (etäkoneen) salasana
  - avaintiedostopari (identity file)

    luodaan komennolla `ssh-keygen -t rsa`

    salainen avain, yleensä id_rsa, sijaitsee lokaalissa koneessa, yleensä hakemistossa $HOME/.ssh

    julkinen avain, yleensä id_rsa.pub, sijaitsee etäkoneessa, yleensä tiedostossa $HOME/.ssh/authorized_keys (jossa niitä voi olla monta, jokainen omalla rivillään)

    - avaintyyppi yleensä rsa, muita dsa*, ecdsa, ed25519

      *) deprecated, ei suositella

ssh & scp: autentikointi

- Avaimella voi olla (ja yleensä on syytä olla) oma salasanansa (passphrase)
- Avain + passphrase -yhdistelmällä saadaan aikaan ”two-factor authentication”: pitää kaapata sekä avain(tiedosto) että salasana ennen kuin voi autentikoitua toisena
- Salasanan voi vaihtaa komennolla ssh-keygen -p
- rootin ssh-avaimella ei yleensä ole salasanaa (koska sitä käytetään automatisoiduissa skripteissä ym)

ssh & scp: autentikointi

- Esimerkki:

  ssh-keygen -t rsa

  ssh kone mkdir -p .ssh

  ssh kone 'cat >>.ssh/authorized_keys' <~/.ssh/id_rsa.pub

  tai, *jos etäkoneessa ei varmasti ole muita avaimia ennestään:*

  scp ~/.ssh/id_rsa.pub kone:.ssh/authorized_keys

ssh-agent

- Jos ssh-avaimella on salasana, sen toistuvalta kirjoittamiselta välttyy apuohjelmalla ssh-agent, joka pitää sitä muistissa session ajan. Se voi käynnistyä automaattisesti, ellei, käsin esim.

  exec ssh-agent bash

- Avaimen tallennus voi myös olla automaattista tai käsin komennolla

  ssh-add

.ssh/known_hosts

- ssh tallettaa tiedostoon ~/.ssh/known_hosts tunnettujen (aiemmin käytettyjen) koneiden avaimen tunnisteen (fingerprint). Jos avain on muuttunut, se valittaa mahdollisesta man-in-the -middle -hyökkäyksestä.
- Virheestä pääsee eroon poistamalla ao. rivin known_hosts -tiedostosta – tai rivit, niitä on yleensä kaksi (koneen nimellä ja IP:llä).
- Voi tehdä myös globaalin /etc/ssh/ssh_known_hosts

.ssh/known_hosts

- Rivin poistaminen known_hosts -tiedostosta onnistuu koneen nimelle komennolla

  ssh-keygen -R *hostname*

  tai millä tahansa editorilla (sekä nimelle että IP:lle), myös sed käy:

  sed -i 40d ~/.ssh/known_hosts

- Jos samassa koneessa pyörii useita ssh-demoneja (eri porteissa), known_hosts -tiedostoa joutuu editoimaan käsin (samalle IP:lle monta riviä)

.ssh/config, ssh_config

- ssh:n toimintaan vaikuttavia asetuksia voi tehdä globaalisti tiedostossa /etc/ssh/ssh_config tai käyttäjäkohtaisesti tiedostossa ~/.ssh/config (ks. man ssh_config). Esim.

```
AddKeysToAgent yes
CheckHostIP no
Host tt1
        Hostname tt1.student.it.jyu.fi
        User tt0
        Port 50022
        IdentityFile ~/.ssh/backup_id
```

.ssh/config, ssh_config

- Asetuksia voi eriyttää mm. käyttäjätunnuksen tai koneen mukaan match-säännöllä, esim.

  Match User ”!root,*”

  SendEnv LANG LC_*

  lähettäisi kieliympäristön paitsi jos tunnus (etäkoneessa) on root

- Muita ehtoja localuser, host, originalhost ja exec (mielivaltainen komento, tulkitaan todeksi jos palautuskoodi on nolla)

.ssh/config, ssh_config

- Asetusten prioriteettijärjestys alimmasta ylimpään on

  sisäänrakennetut oletusarvot

  /etc/ssh/ssh_config

  ~/.ssh/config

  komentorivioptiot
- /etc/ssh/ssh_config siis vain oletusasetukset, ei rajoita mitä käyttäjät voivat tehdä

sshd_config

- ssh-*demonin* asetuksia (globaaleja) säädetään tiedostolla /etc/ssh/sshd_config, esim.

  PermitRootLogin prohibit-password

  DenyUsers evilguy

  AllowGroup sudo

  ForceCommand ...

- Match kuten ssh_config'issa, ehtoina user, group, host, address*, localaddress*, localport

  * myös CIDR, esim. 130.234.208.0/23

- Tällä **voi** rajoittaa mitä käyttäjät voivat tehdä

virt-install/virsh etäkäyttö

- Kone, jossa virt-install ajetaan voi olla eri kuin se, johon virtuaalikone luodaan:

  virt-install --connect qemu+ssh://lonkaX/system -n *tunnus*1 --memory …
- Myös virsh ja virt-viewer toimivat etänä samalla syntaksilla
- ssh-yhteydet on valmisteltava etukäteen (authorized_keys)
- Levyimage luotava etukäteen (qemu-img create...)

Ubuntun päivitys

- sudo apt-get update; sudo apt-get dist-upgrade
  - säännöllisesti, ja aina kun tietoturva-aukkoja tiedossa
  - apt-get upgrade jättää pois boottia vaativat päivitykset
  - automatisointi:

    /etc/apt/apt.config.d/50unattended-upgrades

    dpkg-reconfigure unattended-upgrades
  - automatiikassa vaaransa: levy voi täyttyä, jokin päivitys voi muuten vain rikkoa paikkoja

Ubuntun päivitys

- Käyttöjärjestelmäversion päivitys:

  sudo do-release-upgrade

  - muistia pitää olla riittävästi, samoin levytilaa
  - jos do-release-upgrade -komentoa ei löydy:

    sudo apt-get install update-manager-core

Ubuntun päivitys

- Vain LTS-versioiden välillä vai kaikki:

  /etc/update-manager/release-upgrades:

  ”Prompt=” lts, normal

- Päivityksen jälkeen apt-get autoremove (muutenkin aina kun kone sitä suosittaa)

for-loop, parameter expansion

```
for i in 1 3 5 7 ; do touch koe$i.txt ; done
for x in a b c ; do echo  $x > ${x}koe.txt ; done
for f in *.txt ; do cp $f $f.bak ; done
```

for-loop, parameter expansion

```
for f in *.txt ; do cp $f ${f%.txt}.bak ; done
for f in *.txt ; do cp $f ${f/txt/bak} ; done # bash
for f in a* ; do mv $f b${f#a} ; done
```

for-loop, parameter expansion

```
for n in {0..7} ; do mkdir d$n ; done # bash
for d in d{0..7} ; do mkdir $d ; done # bash
touch koe{1..7..2}.txt # bash
```

Uudelleensuuntaus, redirection

- Komento >tiedosto 2>virheet
- Komento 1>&2
- 0=stdin, 1=stdout, 2=stderr
- 3...9 vapaasti käytettävissä

redirection, read

```
while read suku etu login demo ; do
  echo $login $etu $suku; id $login
done < kurssilaiset.txt
```

redirection, read

```
while read suku etu login demo ; do # ei toimi!
  echo ${login}1; ssh ${login}1 'dpkg -l | grep acpid'
done < kurssilaiset.txt
```

redirection, read

```
while read <&3 suku etu login demo ; do # toimii
  echo ${login}1; ssh ${login}1 'dpkg -l | grep acpid'
done 3< kurssilaiset.txt
```