traceroute

- Reititys- ja palomuuriongelmien selvittämisessä yksi perustyökalu on traceroute, joka yrittää selvittää mitä kautta paketti kulkee:

  traceroute [options] kone

- Usein mitään optioita ei tarvita, mutta jos jokin palomuuri välissä estää normaalin toiminnan, voi kokeilla vaihtoehtoisia menetelmiä:

  -I käytä ICMP ECHOa (ping)

  -T käytä TCP SYN-paketteja

- Muitakin optioita erikoistarpeisiin on, mm.

  -n älä hae koneiden nimiä DNS:stä

  -m *max_ttl* aseta yläraja hyppymäärälle (oletus 30)

  -w *waittime* kuinka monta sekuntia vastausta odotetaan (oletus 5)

here-document

- Joskus skriptissä halutaan antaa komennolle syötteeksi monta riviä käyttämättä aputiedostoa. Kätevä tapa siihen on ns. *here-document:*

  komento <<loppusana

- Se lukee samasta skriptitiedostosta seuraavia rivejä ja syöttää ne komennolle *loppusanaan* (usein EOF) saakka. Jos loppusanaa ei suojata lainausmerkeillä tai \:llä, shell käsittelee erikoismerkkinsä (kuten $) normaaliin tapaan. Esim.

- KALA=lohi

  cat <<EOF >kalat.txt

  hauki

  ahven

  $KALA

  EOF

nginx

- ”Engine X”, toiseksi yleisin (apachen jälkeen) www-palvelin Linux-ympäristössä, yleistyy nopeasti
- Ei ihan yhtä ”full-featured” kuin apache, mutta monissa tilanteissa kevyempi (etenkin muistin tarve pienempi, joskus vertailukelpoinen jopa lighttpd:n kanssa)
- Yleinen (tehokas) edustapalvelimena (proxy)

nginx

- Ubuntu tarjoaa useita asennuspaketteja:

  nginx-core

  nginx-light

  nginx-full

  nginx-extras

  - Virtuaalipaketti "nginx" asentaa kulloisenkin oletussetin (tällä hetkellä ~ nginx-core) mutta ei vaihda sitä päivityksissä
  - ks. apt-cache search ^nginx

nginx konfiguraatiotiedostot

- Konfiguraatiohakemisto /etc/nginx
- Pääkonfiguraatiotiedosto /etc/nginx/nginx.conf
- Lisämodulit /etc/nginx/modules-enabled/*.conf (linkkejä → ../modules-available tai /usr/share/nginx/modules-available)
- Lisäkonfiguraatiotiedostoja /etc/nginx/conf.d/*.conf (automaattisia)

  ja /etc/nginx/snippets/*.conf (otettava käyttöön erikseen)
- Tiedostotyyppimäärittelyt /etc/nginx/mime.types
- Omia konfiguraatiotiedostoja voi sisällyttää muualtakin include-direktiivillä (vrt. grep include /etc/nginx/nginx.conf)

nginx virtual host -määritykset

- Virtual hostit (www-) hakemistossa /etc/nginx/sites-available, käytössä olevat /etc/nginx/sites-enabled (linkkejä edelliseen)
- Oletus /etc/nginx/sites-available/default, sijainti:

  grep root /etc/nginx/sites-available/default

  - /var/www/html, /usr/share/nginx/html tms

  grep index /etc/nginx/sites-available/default

  - Jos useita vaihtoehtoja, ensimmäistä löytynyttä käytetään
  - Yleensä ainakin index.html toimii
- Jos default-konfiguraatiota ei käytetä, poistetaan linkki:

  rm /etc/nginx/sites-enabled/default

nginx: virtual host, esimerkki

/etc/nginx/sites-available/tt2.student.it.jyu.fi:

```
server {
  listen 80; # Ipv4; voisi olla myös esim. 8080 tai 443 (https)
  listen [::]:80; # IPv6
  server_name tt2.student.it.jyu.fi;
  root /var/www/tt2.student.it.jyu.fi/html;
  index index.html index.htm;
  location / {
      try_files $uri $uri/ =404;
   }
}
```

nginx: virtual host, esimerkki

```
ln -s  ../sites-available/tt2.student.it.jyu.fi /etc/nginx/sites-enabled
# huom. ei erityistä komentoa (vrt. lighty-enable-mod)
mkdir -p /var/www/tt2.student.it.jyu.fi/html
cat >/var/www/tt2.student.it.jyu.fi/html/index.html <<\EOF
<html><title>tt2</title><body><p>tt2 here</p></body></html>
EOF
systemctl reload nginx
tail /var/log/nginx/error.log # jos ei toimi...
```

nginx konfiguraatiodirektiivit

- Nginx'in konfiguraatiotiedostossa olevat määritykset (direktiivit) voivat olla globaaleja tai jossain aaltosuluilla {} rajatussa kontekstissa: http{} server{} location{} events{}
- Direktiivien loppuun tulee yleensä puolipiste (ei }:n jälkeen)
- Yleisiä globaaleja asetuksia: user (käyttäjätunnus jolla nginx pyörii), worker_processes (montako palvelinprosessia käynnistetään), pid (minne nginx-pääprosessin PID talletetaan); näitä ei yleensä tarvitse muuttaa
- Konfiguraatiotiedostoon voi lukea muita include-direktiivillä, jokerit toimivat

nginx konfiguraatiodirektiivit

- Kaikki http(s)-palvelimiin liittyvät direktiivit (erityisesti myös server{}) ovat http{}:n sisällä; kaikki .../sites-enabled/... tiedostot tulevat automaattisesti sen sisään (toteutettu include-direktiivillä html{}:n sisällä)
- Virtual host -määritykset tehdään server{} -direktiivillä, niiden sisällön ohjaukset sen sisällä location{}'illa
- events{} ohjaa yhteyksien muodostumista (esim. worker_connections: kuinka monta yhtaikaista yhteyttä per palvelinprosessi sallitaan)

nginx: server{}

listen [*osoite*][:*portti*] [*optiot*];

- jos :*portti* puuttuu, oletetaan 80
- jos *osoite* puuttuu, oletus = * = kaikki IPv4-osoitteet
- [::] = kaikki IPv6 -osoitteet
- voi olla useita samassa server{}:ssä
- yleinen optio default_server, muita mm. ssl, proxy_protocol

nginx: server{}

server_name *nimi* [*nimi...*];

- nimi tai nimet joihin vastataan; ensimmäinen ensisijainen
- jokerit ja regexp'it sallittuja: *.example.com ~^www\..*\.example\.net

root *polku*;

- sisällön juurihakemisto, johon suhteessa URL-polut tulkitaan

nginx: server{}

index *file* [*file...*];

- tiedostonimi jota käytetään URLin viitatessa hakemistoon ts. lopussa /
- jos useita, käytetään ensimmäistä joka löytyy

error_log *file* | stderr | syslog:server=*...*

- minne virheilmoitukset kirjoitetaan (paljon optioita)

location …

nginx: location{}

location [= ~ ~* ^~] *URI* {...}

location @*nimi* {...}

- URIsta riippuvia asetuksia (server{}in sisällä); voi olla useita sisäkkäin
- @*nimi* määrittää nimetyn sijainnin käytettäväksi uudelleenohjausviittauksissa
- location{}in sisällä voi olla mm.

    root, index, error_log, proxy_pass ...

nginx: location{}

- URI voi olla

  polun *alku* (esim. /); jos edessä ^~, ohittaa regexpit

  - käsitellään pituusjärjestyksessä, pisin voittaa

  =koko polku (”= /kala/hauki.html”)

  - voittaa kaikki muut; voi olla esim. yhtaikaa sekä ”/” että ”= /”

  ~regexp (”~ \.php$”), ~* case-insensitive

  - regexp'it voittavat polunalkumääritykset ilman ^~ -prefiksiä
  - regexp'it tulkitaan järjestyksessä, ensimmäinen voittaa

nginx: try_files

try_files *file ... URI;*

- Vaihtoehtoja miten tiedosto tulkitaan, ensimmäinen löytynyt voittaa

  $uri on alkuperäinen polku sellaisenaan

  =*code* palauttaa virhekoodin

  - yleisin virhekoodi 404 Not Found
  - lista koodeista ks.

    http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml
    https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

nginx: try_files

- Esim.

```
location / {
  try_files /huoltokatko.html $uri $uri/ $uri.html =404;
}
location /pictures/ {
  try_files $uri /pictures/oletus.jpg;
}
```

nginx: try_files

```
location / {
  try_files $uri @varakone;
}
location @varakone {
  proxy_pass http://192.168.6.7:8080;
}
```

nginx: php

- sudo apt-get install php-common php-cli php-fpm
- Halutun virtual hostin konfiguraatiotiedostoon:

```
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
}
```

- Joskus tarvitaan lisäksi (polku voi vaihdella)

```
fastcgi_pass unix:/var/run/php-fpm.sock;
```

- Yleensä halutaan lisätä index-määritykseen index.php, että /-loppuiset URIt toimisivat myös php:n kanssa
- sudo systemctl restart nginx

nginx: rewrite

rewrite *regexp korvaus* [*lippu*]

- Muuntaa URIn toiseksi regexp-säännöllä
- Jos useita, suoritetaan järjestyksessä
- Liput:

  break: lopeta rewrite-sääntöjen suoritus tähän

  last: lopeta tähän ja ala käsitellä muutettua URIa alusta

  - voi johtaa silmukkaan...

  redirect: palauta ”temporary redirect” -koodi

  permanent: palauta ”permanent redirect” -koodi

nginx: rewrite

- Esim.

```
location /download/ {
   rewrite ^(/download/.*)/media/(.*)\..*$ $1/mp3/$2.mp3 break;
   rewrite ^(/download/.*)/audio/(.*)\..*$ $1/mp3/$2.ra break;
   return  403;
}
```

nginx: proxy_pass

proxy_pass *URL*;

- Ohjaa (yleensä location{}in määräämän) URLin toiseen
- Esim.

```
location / {
  proxy_pass http://192.168.122.4:8080/;
}
location /images/ {
  proxy_pass http://192.168.122.5/pics/;
}
```

nginx: proxy_pass

- Yhdistettävissä esim. rewrite-direktiivin kanssa:

```
location /user/ {
    rewrite    /user/([^/]+) /users?name=$1 break;
    proxy_pass http://192.168.7.8/;
}
```

nginx: reverse proxy

- Esimerkki: halutaan erottaa virtual hosteja sisäisesti eri virtuaalikoneisiin.
- Ohjataan kaikki ulkoa tulevat pyynnöt edustapalvelimena toimivalle nginx'ille, jossa on kullekin virtual hostille tämäntapainen määritys:

```
server {
  listen 80;
  server_name 1.example.org;
  location / { proxy_pass http://kone1:8080/; }
}
```

nginx: reverse proxy

- Virtual hosteja voidaan jakaa eri virtuaalikoneisiin kuten halutaan, jossain voi pyöriä nginx, jossain apache, jossain lighttpd... jopa useita samassa koneessa eri porteissa, edustakoneen huolehtiessa ohjauksesta oikeaan paikkaan.
- Edustakone voi myös vaihtaa protokollaa, erityisesti https→ http (jolloin taustakoneiden ei tarvitse huolehtia sertifikaateista), tai ftp → http(s) tms

lynx, elinks

- Joskus on hyödyllistä testata www-sivujen toimintaa komentoriviltä interaktiivisesti, ei-graafisella selaimella. Tunnetuimpia ovat:
- lynx: vanhin, pieni ja kevyt, ei javascriptiä eikä css:ää
- elinks: tukee (jotenkin) javascriptiä ja css:ää
- Molemmissa paljon optioita, erityisesti automatisoituun testaukseen sopiva -dump
- Vrt. wget ja curl

Skriptien suorituksesta

- Skriptin ensimmäisellä rivillä voidaan määrätä käytettävä tulkkiohjelma (shell) mahdollisine optioineen:

  ```
  #! /bin/bash -e
  ```

  - optioita: -e = keskeytä virheeseen, -x = jäljitä suoritusta
- Jos skriptitiedosto on suoritettava (`chmod +x`), sen voi ajaa suoraan tyyliin `./skripti.sh`, muuten `bash skripti.sh` (tai `bash -ex skripti.sh` jne) tai `sh skripti.sh`
- Shellille voi antaa skriptin komentorivillä tyyliin `sh -c '…'`, hyödyllistä mm. sudo tai ssh:n kanssa rajaamaan mikä shell tulkitsee minkäkin osan komentorivistä

Tilapäiset tiedostot

- Kun skriptissä tarvitaan tilapäistä tiedostoa, sopiva paikka on usein /tmp tai /var/tmp - mutta tiedostonimeä ei yleensä kannata kiinnittää, useammastakin syystä:
  - Jos skriptiä voidaan ajaa monta kertaa samaan aikaan, saman tiedostonimen käyttö aiheuttaa ongelmia
  - Jos aiottu nimi on jo jonkin muun käytössä, sen käyttö ei yleensä onnistu
  - Kiinteä tiedostonimi saattaa muodostaa tietoturvaongelman (hyökkäysvektori mustahatuille)
- Yksi keino on prosessin id, ympäristömuuttujassa $$:

  MYFILE=/tmp/.jotain$$ # ei kovin turvallinen

mktemp

- Yleinen ratkaisu tilapäistiedostotarpeisiin on

  mktemp [-du] [-p DIR] *template*

  template on nimimalli, jossa pitää olla vähintään 3 X:ää (oletus tmp.XXXXXXXXXX)

  -p (--tmpdir) hakemisto jonka alle tiedosto luodaan

  -d (--directory) luo hakemisto tiedoston asemesta

  -u (--dry-run) älä luo tiedostoa, palauta vain nimi
- Vastaavat kirjastokutsut mktemp(3), mkstemp(3) ja mkdtemp(3)
- Luotu tiedosto pitää erikseen hävittää kun sitä ei enää tarvita.

trap

- Skriptissä voi varautua sen keskeytymiseen virheeseen:

  trap '*komento*' *signal[s]*

  Signaalina 0 tarkoittaa skriptin loppumista mistä tahansa syystä, ja se on hyödyllinen erityisesti "loppusiivoukseen".

  Esim.

  ```
  MYTEMPFILE=$(mktemp /tmp/.jotainXXX)
  trap "rm $MYTEMPFILE" 0
  ```

  (Huom. lainausmerkkien ' ' ja " " ero)

telnet

- Telnet on alunperin tehty pääteyhteyden muodostamiseksi verkon yli, mutta koska siinä ei ole minkäänlaista salausta, siitä on luovuttu eikä telnet-demonia enää juuri missään käytetä. Telnet-client on kuitenkin yhä hyödyllinen avointen tcp-porttien testaamiseen ja tekstipohjaisten protokollien kokeilemiseen käsin:

  telnet *kone portti*

telnet

- Esim. kokeillaan postipalvelimen toimintaa:

```
$ telnet smtp.example.org 25
Trying 10.23.4.25...
Connected to smtp.example.org.
Escape character is '^]'.
220 mail1.example.org ESMTP; Wed, 6 May 2015 07:51:39 +0300
quit
221 2.0.0 mail1.example.org closing connection
Connection closed by foreign host.
```

Vanhojen kernelien siivous

- Usein kernel-päivitykset jättävät vanhatkin kernelit paikalleen. Ennen pitkää ne täyttävät /boot-osion (tai /:n).
- Yleensä apt (tai apt-get) autoremove poistaa tarpeettomat (säilyttää kaksi uusinta), mutta aina se ei toimi.
- Joissakin (vanhemmissa) Ubuntu-versioissa auttaa

  purge-old-kernels [--keep *n*] # säilyttää *n* uusinta, oletus 2

  - Ei aina valmiiksi asennettuna, tarvittaessa apt-get install byobu
  - Kutsuu apt-get -komentoa eikä toimi jos se on rikki (erityisesti jos päivitys tai asennus on jo kaatunut levyn täyttymiseen…)
- Jos em. eivät toimi, vanhoja kerneleitä voi poistaa yksitellen käsin (dpkg -l | grep linux image; apt --purge tai dpkg -r ...)