Laboration – Nätverk

OBS det kan förekomma små skillnader vad gäller sökvägar samt befintliga kommandon mellan olika distributioner. T.ex. mellan CentOS och Ubuntu,

Material: För att genomföra laborationen behöver man ha tillgång till en dator (vm) med Linux installerat. Windows Subsystem for Linux går att använda också. För vissa övningar kan lokal administratörsbehörighet behövas. I labben utgår det från att man är inloggad med en användare som heter **sysadmin**

Mål: I denna laboration kommer du att utföra följande uppgifter:

- Undersök nätverkskonfigurationsinformation

OBS IP-adresser, DNS- och dator-namn kan variera beroende på hur aktuell labbmiljö ser ut.

Undersök nätverkskonfiguration

I denna uppgift kommer du att köra flera kommandon och undersöka olika filer för att visa din nätverkskonfiguration.

För att ta reda på din Internet Protocol (IP)-adress, kör kommandot `ifconfig`:

```
sysadmin@localhost:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 192.168.1.2  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::42:c0ff:fea8:102  prefixlen 64  scopeid 0x20<link>
        ether 02:42:c0:a8:01:02  txqueuelen 0  (Ethernet)
        RX packets 11  bytes 878 (878.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 1  bytes 49 (49.0 B)
```

OBS i vissa Linux distributioner så är inte `ifconfig` installerat längre. I stället använder man det nyare kommandot `ip address show`

Utdatan visar två huvudblock med information. Det första blocket, indenterat med `eth0`, visar information om ditt första Ethernet-nätverkskort. Det andra blocket, indenterat med `lo`, visar information om *loopback* eller den interna nätverksgränssnittet.

Den andra raden i varje block innehåller relevant information för version 4 av Internet Protocol (kallas IPv4) medan den tredje raden har information för version 6 av Internet Protocol (IPv6). IPv4 är en äldre metod för att identifiera datorer med en serie siffror. Den används fortfarande i stor utsträckning idag trots att den förbättrade IPv6-metoden har funnits i många år.

IPv4-adresser visas som fyra decimaltal mellan 0 och 255, separerade med punkter.

IPv6-adresser är *128-bitars* tal som visas som hexadecimala siffror mellan 0 och f. De hexadecimala siffrorna organiseras oftast i grupper om fyra siffror, separerade med kolon. Om flera på varandra följande hexadecimala siffror har värdet noll ersätts de med två kolon.

Att ha en IP-adress gör att ditt system kan kommunicera med andra system på samma nätverk. Med *routning*-enheter kan du även kommunicera med system på andra nätverk. För att visa tabellen med routningsinformation använder du kommandot `route`:

```
route
```

```
sysadmin@localhost:~$ route
Kernel IP routing table
Destination  Gateway         Genmask    Flags Metric Ref  Use Iface
default      192.168.1.1     0.0.0.0     UG    0      0      0 eth0
192.168.1.0  0.0.0.0     255.255.255.0   U     0      0      0 eth0
```

När du ansluter till andra datorer kan du använda antingen en IP-adress eller ett värdnamn. Värdnamn kan användas om de är inlagda i filen `/etc/hosts` tillsammans med tillhörande IP-adress, eller om en Domain Name Server (DNS) tillhandahåller översättning mellan IP-adress och värdnamn.

Några namn som ofta finns med i filen `/etc/hosts` är `localhost` och `localhost.localdomain`, vilka båda används för att referera till den aktuella maskinen.

Verifiera att IP-adressen `127.0.0.1` har en post i filen `/etc/hosts`:

```
grep 127.0.0.1 /etc/hosts
```

Utdatan ska se ut ungefär så här och definiera `localhost`-namnen:

Kommandot `ping` kan användas för att ta reda på om ett system för närvarande är anslutet till ett nätverk.

```
sysadmin@localhost:~$ grep 127.0.0.1 /etc/hosts
127.0.0.1       localhost
```

Ibland kan ett system vara konfigurerat så att det inte svarar på ping-förfrågningar. Därför betyder avsaknad av svar på ett `ping`-kommando inte nödvändigtvis att systemet inte är anslutet till ett nätverk. Ett snabbt svar på `ping`-kommando visar däremot att systemet är anslutet till ett nätverk.

Testa om localhost-maskinen svarar på fyra `ping`-förfrågningar:

```
ping -c4 localhost
```

```
sysadmin@localhost:~$ ping -c4 localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.032 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.029 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.035 ms

--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.029/0.032/0.035/0.007 ms
```

Till skillnad från ping-kommandot som finns i Microsoft Windows-operativsystemet, slutar Linux ping-kommandot inte att skicka *förfrågningar* automatiskt. Om du glömmer att ange `-c`-alternativet måste du själv avbryta kommandot genom att hålla ner **Control** och trycka på **C** (**CTRL** + **C**).

Värdnamn kan också användas om de är registrerade hos en Domain Name System (DNS)-server. Om ditt system är anslutet till ett nätverk med DNS-servrar, konfigurerar `nameserver`-posten i filen `/etc/resolv.conf` ditt system för att använda dessa servrar för att översätta värdnamn till IP-adresser.

Visa filen `/etc/resolv.conf` för att se om det finns någon `nameserver`-post:

```
cat /etc/resolv.conf
```

Utdatan ska visa en `nameserver`-post:

```
sysadmin@localhost:~$ cat /etc/resolv.conf
search vixen.lb
nameserver 127.0.0.11
options ndots:0
```

Använd kommandot `dig` för att slå upp namnet `localhost.localdomain` och få dess IP-adress:

```
dig localhost.localdomain
```

```
sysadmin@localhost:~$ dig localhost.localdomain

; <<>> DiG 9.11.3-1ubuntu1.5-Ubuntu <<>> localhost.localdomain
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56461
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 0708514fa6562be38505ea9d6000c50b4aea1805343782f8 (good)
;; QUESTION SECTION:
;localhost.localdomain.         IN      A

;; ANSWER SECTION:
localhost.localdomain.  86400   IN      A       127.0.0.1

;; AUTHORITY SECTION:
localdomain.            86400   IN      NS      localhost.localdomain.

;; Query time: 1 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Thu Feb 22 02:22:45 UTC 2024
;; MSG SIZE  rcvd: 108                        1
```

Observera att utdatan visar att det är den första nameserver som anges i filen `/etc/resolv.conf` som ger svaret i utdatan (`;; SERVER: 127.0.0.11#53(127.0.0.11)`).

Observera
Om kommandot ovan inte fungerar, kör följande kommando innan du försöker igen med `dig`-kommandot ovan. När du ombeds ange ett lösenord ange lösenordet för din användare:

```
sudo /etc/init.d/bind9 restart
```

Du kan använda kommandot `dig` för att slå upp andra *fullständigt kvalificerade domännamn*.
Använd `dig`-kommandot för att slå upp värdnamnet `cserver.example.com` och få dess IP-adress:

```
dig cserver.example.com
```

```
sysadmin@localhost:~$ dig cserver.example.com

; <<>> DiG 9.11.3-1ubuntu1.1-Ubuntu <<>> cserver.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25402
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c33d9913b915efc1e055c1575c006fe19ca3975ce1951a1c (good)
;; QUESTION SECTION:
;cserver.example.com.           IN      A

;; ANSWER SECTION:
cserver.example.com.    86400   IN      A       192.168.1.2

;; AUTHORITY SECTION:
example.com.            86400   IN      NS      example.com.

;; ADDITIONAL SECTION:
example.com.            86400   IN      A       192.168.1.2

;; Query time: 0 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Thu Feb 22 02:24:57 UTC 2024
;; MSG SIZE  rcvd: 122
```

Ett fullständigt kvalificerat domännamn (FQDN) innehåller inte bara värdnamnet, utan också den domän som värdnamnet tillhör. För FQDN `cserver.example.com` är `cserver` värdnamnet (hostname) och `example.com` är domänen.

Använd kommandot `dig` för att slå upp värdnamnet till IP-adressen `192.168.1.2`:

```
dig -x 192.168.1.2
```

```
sysadmin@localhost:~$ dig -x 192.168.1.2

; <<>> DiG 9.11.3-1ubuntu1.1-Ubuntu <<>> -x 192.168.1.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59190
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;2.1.168.192.in-addr.arpa.      IN      PTR
```

Värdnamnet hittar du i ANSWER SECTION i utdatan från `dig`-kommandot. Värdnamnet i exemplet ovan ser ut så här:

```
;; ANSWER SECTION:
2.1.168.192.in-addr.arpa. 600   IN      PTR     84a6444e-f044-4754-98e6-
64cc9f7dadea.1.84a6444e-f044-4754-98e6-64cc9f7dadea.LAN.
```

OBS resultatet variera beroende på labbmiljö och nätverksinställningar

Kommandot `netstat` utför en mängd olika uppgifter relaterade till nätverk. För att få en uppfattning om några av dess funktioner, kör kommandot med alternativet `--help`:

```
netstat --help
```

```
sysadmin@localhost:~$ netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -i | [-cnNe] -M | -s [-6tuw] }

        -r, --route              display routing table
        -i, --interfaces         display interface table
        -g, --groups             display multicast group memberships
        -s, --statistics         display networking statistics (like SNMP)
        -M, --masquerade         display masqueraded connections

        -v, --verbose            be verbose
        -W, --wide               don't truncate IP addresses
        -n, --numeric            don't resolve names
        --numeric-hosts          don't resolve host names
        --numeric-ports          don't resolve port names
        --numeric-users          don't resolve user names
        -N, --symbolic           resolve hardware names
        -e, --extend             display other/more information
        -p, --programs           display PID/Program name for sockets
        -o, --timers             display timers
        -c, --continuous         continuous listing
```

Ett vanligt användningsområde för `netstat` är att ta reda på vilka tjänster som *lyssnar* på eller väntar på en inkommande anslutning. Till exempel är en tjänst som används för att tillåta fjärr- eller nätverksinloggningar Secure SHell eller SSH. SSH lyssnar normalt på TCP-port 22.

Välkända portar är portnummer i intervallet 0–1023, som vanligtvis används av systemprocesser för att tillhandahålla nätverkstjänster. En lista över tjänstenamn och tillhörande portnummer finns i filen `/etc/services`.

Använd kommandot `netstat` för att kontrollera om det finns en process som lyssnar på TCP-port 22, för `ssh`.

```
netstat -tl

netstat -tln
```

```
sysadmin@localhost:~$ netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:38453        0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:domain             [::]:*                  LISTEN
sysadmin@localhost:~$ netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.2:53          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:38453        0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
```

Flaggan `-t` till kommandot `netstat` begränsar listningen till TCP-portar; `-l` visar bara portar där tjänster lyssnar; `-n` visar nätverksadresser numeriskt:

```
sysadmin@localhost:~$ netstat -ltn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.2:53          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:44441        0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
```

Ett mer modernt sätt är att använda kommandot `ss` för att se vilka anslutningar som för närvarande är etablerade mellan den lokala maskinen och fjärrmaskiner, samt statistik om dessa anslutningar och liknande.

Precis som med kommandot `netstat` kan du få mycket användbar information direkt från `ss`, vilket visas i exemplet nedan där vi använder kommandot *ss* för att visa nätverksstatistik:

```
ss
```

```
sysadmin@localhost:~$ ss
Netid State       Recv-Q  Send-Q    Local Address:Port      Peer Address:Port
tcp   CLOSE-WAIT  116     0             127.0.0.1:58006        127.0.0.1:8000
tcp   CLOSE-WAIT  116     0             127.0.0.1:58010        127.0.0.1:8000
tcp   FIN-WAIT-2  0       0             127.0.0.1:8000         127.0.0.1:58014
tcp   CLOSE-WAIT  116     0             127.0.0.1:58014        127.0.0.1:8000
tcp   FIN-WAIT-2  0       0             127.0.0.1:8000         127.0.0.1:58010
tcp   FIN-WAIT-2  0       0             127.0.0.1:8000         127.0.0.1:58006
```

OBS för att kunna se något resultat med `ss` så måste vi kanske generera lite trafik med ssh eller liknande verktyg. T.ex. ansluta med `ssh` till localhost.

Extra övningar

För att enklare kunna konfigurera nätverksinställningar är det oftast bäst att lägga till ett extra virtuellt nätverkskort till den virtuella maskinen och ansluta detta till ett privat lokalt nätverk. Så att vi inte konfigurera det befintliga nätverkskortet som används.

Lägg till nätverkskort (eth1) i VM-miljö

- **Hyper-V (Generell beskrivning)**
  1. Stäng av VM
  2. Öppna **Hyper-V Manager**
  3. VM → **Settings**
  4. **Add Hardware** → **Network Adapter**
  5. Välj switch:
     - *Default Switch* (NAT)
     - eller *Internal / Private*
  6. Starta VM

Verifiera i Linux:

```
ip link show
```

- **VirtualBox**
  1. Stäng av VM
  2. **Settings → Network**
  3. Adapter 2:
     - Enable
     - Type: *NAT* eller *Host-only*
  4. Starta VM

Verifiera:

```
ip a
```

Övning 1 – Identifiera flera nätverkskort

Uppgift

Verifiera att systemet har fått ett extra nätverkskort (`eth1`).

```
ip link show
```

Alternativt:

```
ip address show
```

Förväntat resultat

- Minst två Ethernet-interface, t.ex.:
    - eth0
    - eth1

Notera:

- MAC-adress
- Om interfacet är UP eller DOWN
- Om IP-adress finns tilldelad

Övning 2 – Aktivera ett nätverksinterface manuellt

Om eth1 är nedstängt:

```
sudo ip link set eth1 up
```

Verifiera:

```
ip link show eth1
```

Övning 3 – Tilldela statisk IP-adress temporärt

Denna konfiguration gäller **endast tills omstart**.

```
sudo ip address add 192.168.50.10/24 dev eth1
```

Verifiera:

```
ip address show eth1
```

Testa lokal anslutning:

```
ping -c3 192.168.50.10
```

Övning 4 – Permanent nätverkskonfiguration

- **Ubuntu (Netplan)**

Visa befintliga netplan-filer:

```
ls /etc/netplan/
```

Exempel: `/etc/netplan/01-netcfg.yaml`

Redigera:

```
sudo nano /etc/netplan/01-netcfg.yaml
```

Exempelkonfiguration:

```
network:
  version: 2
  ethernets:
    eth1:
      addresses:
        - 192.168.50.10/24
      dhcp4: no
```

Applicera:

```
sudo netplan apply
```

Verifiera:

```
ip a show eth1
```

- **CentOS / Rocky / Alma (NetworkManager)**

Lista anslutningar:

```
nmcli con show
```

Skapa ny anslutning:

```
sudo nmcli con add type ethernet ifname eth1 con-name eth1-static
ipv4.method manual ipv4.addresses 192.168.50.10/24
```

Aktivera:

```
sudo nmcli con up eth1-static
```

Verifiera:

```
ip a show eth1
```

Övning 5 – Standard-gateway och routing

Visa routingtabellen:

```
ip route
```

``

Identifiera:

- Default route (default via … dev eth0)

- Lokala nät (192.168.50.0/24 dev eth1)

Lägg till temporär route:

```
sudo ip route add 192.168.60.0/24 via 192.168.50.1 dev eth1
```

Ta bort:

```
sudo ip route del 192.168.60.0/24
```

Övning 6 – DNS-test och namnuppslagning

Testa DNS:

```
dig google.com
```

Testa specifik DNS:

```
dig @8.8.8.8 example.com
```

Reverse lookup:

```
dig -x 8.8.8.8
```

Övning 7 – Traceroute och nätverksväg

Installera vid behov:

```
sudo apt install traceroute        # Ubuntu

sudo dnf install traceroute        # CentOS
```

Kör:

```
traceroute google.com
```

Undersök:

- Antal hopp
- Fördröjning
- Lokala vs externa nät

Övning 8 – Nätverkstrafik med tcpdump (felsökning)

Installera:

```
sudo apt install tcpdump

sudo dnf install tcpdump
```

Lyssna på eth1:

```
sudo tcpdump -i eth1
```

ICMP-filtrering:

```
sudo tcpdump -i eth1 icmp
```

Avsluta med CTRL + C

Övning 9 – Brandvägg och nätverk

Ubuntu (UFW)

```
sudo ufw status
sudo ufw allow ssh
sudo ufw enable
```

``

Testa:

```
ss -tln
```

CentOS (firewalld)

```
sudo firewall-cmd --list-all
sudo firewall-cmd --add-service=ssh --permanent
sudo firewall-cmd --reload
```

Övning 10 – Nätverksfelsökning (checklista)

När nätverk inte fungerar, kontrollera:

```
ip a
ip route
ping 127.0.0.1
ping <gateway>
ping 8.8.8.8
dig google.com
ss -tuln
```