Installasjon av klientserver

Teknisk dokumentasjon
10.05.2011

Oversikt

Klientserveren trenger i hovedsak litt ekstra ram, for en serie på f.eks. 10 maskiner kanskje 4GB, to raske internettkort (Gigabit) og helst tokjernes prosessor. Tre tjenester danner hovedgrunnlaget i tynn/halvtykk-klientboot:
tftpboot, dhcpd og nfs.

Krav

Klientserveren trenger min 1GB RAM og helst to gigabits nettverkskort. Ellers kan en gammel boks brukes til noen få maskiner. Nettverket er uansett flaskehalsen.

Installering

Klientserver installeres med Ubuntu Server (p.t. v12.04 LTS) og openSSH. Kjør sudo apt-get update && sudo apt-get upgrade før noe annet, slik at maskinen er oppdatert. Resten gjøres via ssh.
ekstra pakker: nfsd, tftpd-hpa & dhcpd

(Alternativt kan det installeres desktop-versjon med vino for remote desktop)

```
sudo apt-get install nfs-kernel-server tftpd-hpa isc-dhcp-server subversion syslinux
```

Oppgradering

NB. ved oppgradering fra tidligere server til Ubuntu server 12.04 er det et problem som går igjen: nfs-kernel-server blir ikke fullstendig oppgradert, mangler rpcbind.

dette kan fikses ved å reinstallere nfsd:

```
sudo apt-get install --reinstall nfs-kernel-server
```

Nettoppsett

wlan : får dhcp fra eksternt nett (i vårt tilfelle 10.172.x.x)
lan : statisk nett 192.168.0.1/255.255.255.0
NB: Merk at resten av manualen forutsetter at eth0 er koblet mot eksternt nett (WLAN) mens eth1 er koblet mot svitsj for klientene (LAN)!

settes opp i /etc/network/interfaces:

```
auto lo
iface lo inet loopback

# wlan
auto eth0
iface eth0 inet dhcp

# lan
auto eth1
iface eth1 inet static
    address 192.168.0.1
    netmask 255.255.255.0
```

Gateway

Serveren fungerer også som gateway. Trenger bare to modifikasjoner, NAT masquerade, slik at all trafikk maskeres som gatewayens ip:

MASQ på hele nettverket, output på nic koblet mot internett (=gateway)

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

IP forward, slik at all trafikk mot internt nic (eth1 med dhcp-server) også går til eksternt nic (gateway)

```
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

For at det skal gjelde etter reboot, må følgende settes i sysctl.conf:

```
net.ipv4.conf.default.forwarding=1
# eller
net.ipv4.ip_forward=1
```

Som sagt antar vi her at eth1 med ip 192.168.0.1 er koblet mot switsj for internt lan. Alle klienter kobler direkte på denne.

Brannmur

Siden klientserveren står på personalnettet (10.172.x.x) må det legges noen føringer på bruken. Det er viktig at bare de nødvendige tjenestene får tilgang, samt at bare den faktiske skriveren som publikum skal ha tilgang til kan brukes. Resten underlegges en streng policy (-P INPUT -j DENY). Vi bruker iptables:

```
# iptables for klientserver
WAN_NIC="eth0"
```

```
LAN_NIC="eth1"
PRINTER_IP="10.172.14.2"

# allow known ports
# start with opening input policy before flushing table, otherwise you deny yourself access ;)
iptables -P INPUT ACCEPT
iptables -F

iptables -A INPUT -i lo -j ACCEPT   # accept local traffic of course...

# make sure -i points to correct network interfaces
# allow ssh from outside (WAN)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# allow established connections to continue
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# start allow rules from LAN
iptables -A INPUT -i $LAN_NIC -p tcp --dport 37 -j ACCEPT   # time
iptables -A INPUT -i $LAN_NIC -p udp --dport 37 -j ACCEPT
iptables -A INPUT -i $LAN_NIC -p udp --dport 67:69 -j ACCEPT   # dhcpd+tftpd
iptables -A INPUT -i $LAN_NIC -p tcp --dport 80 -j ACCEPT   # http
iptables -A INPUT -i $LAN_NIC -p tcp --dport 111 -j ACCEPT   # rpc
iptables -A INPUT -i $LAN_NIC -p udp --dport 111 -j ACCEPT
iptables -A INPUT -i $LAN_NIC -p tcp --dport 113 -j ACCEPT  # bootp
iptables -A INPUT -i $LAN_NIC -p udp --dport 113 -j ACCEPT
iptables -A INPUT -i $LAN_NIC -p udp --dport 123 -j ACCEPT # ntp
iptables -A INPUT -i $LAN_NIC -p udp --dport 512 -j ACCEPT # exec
iptables -A INPUT -i $LAN_NIC -p tcp --dport 631 -j ACCEPT  # cupsd
iptables -A INPUT -i $LAN_NIC -p udp --dport 631 -j ACCEPT
iptables -A INPUT -i $LAN_NIC -p tcp --dport 873 -j ACCEPT # rsync
iptables -A INPUT -i $LAN_NIC -p udp --dport 873 -j ACCEPT
iptables -A INPUT -i $LAN_NIC -p udp --dport 881 -j ACCEPT  # rpc.statd
iptables -A INPUT -i $LAN_NIC -p tcp --dport 2049 -j ACCEPT   # nfs
iptables -A INPUT -i $LAN_NIC -p udp --dport 2049 -j ACCEPT

iptables -A INPUT -i $LAN_NIC -p tcp --dport 35923 -j ACCEPT   # pxeboot!

# important: masquerade, to allow internet from/to clients
iptables -t nat -A POSTROUTING -o $WAN_NIC -j MASQUERADE

# add the ip of the printer

# iptables -A PREROUTING -t nat -p tcp -d $EXTERNAL_IP -i $EXTERNAL_INTERFACE --dport 9100 -j SNAT --to $PRINTER_IP

iptables -A FORWARD -d $PRINTER_IP -s 192.168.0.0/24 -j ACCEPT
iptables -P FORWARD DROP

# deny the rest
iptables -P INPUT DROP
```

Deretter må reglene lagres:

```
iptables-save > /etc/iptables.up.rules
```

og følgende legges inn i /etc/network/interfaces for $WAN_NIC slik at reglene hentes ved oppstart

```
pre-up iptables-restore < /etc/iptables.up.rules
```

lav bredbåndshastighet - limit

Et verktøy som enkelt begrenser bredbåndshastigheten slik at ikke trafikk stopper nettet helt:

```
sudo apt-get install wondershaper
```

på iface ut på nett (eth0) sett 512k både inn/ut:

```
sudo wondershaper eth0 512 512
```

for å gjøre innstillingen permanent må det legges til interfacet i /etc/network/interfaces:

```
auto eth0
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 512 512
    down /sbin/wondershaper clear eth0
```

DHCP-server

DHCP-serveren styrer all oppstartskommunikasjon mellom server og klienter. Den deler ut IP-adresser (faste eller dynamiske), informasjon om nettverket (gateway, routes, etc.) samt info om oppstartsbildet.

Velg kort som skal fungere som dhcpd-server, skill med mellomrom i:

/etc/default/isc-dhcp-server

```
INTERFACES="eth1"
```

Sett deretter opp subnettet for hvert enkelt nettverkskort, gjort restriktivt på host-basis mot mac-adresse på nettverkskort, slik at ingen skal kunne koble til tilfeldig.

dhcpd.conf (enkel versjon)

dhcpd er hovedkonfigurasjonen som i prinsippet styrer alt av nettverk:

- utdeling av ip-adresser
- ruting mot tftp-boot (pxeboot fra klientene)
- ruting mot imageserver/nfsserver
- oppsett av dns, domener, etc. for klientene
- gruppering av klienter etter macadresser fra nettverkskort
- ruting mot oppstartskonfigurasjon oppdelt etter grupper

Hvis oppsettet er enkelt, en server som betjener like bilder kan det gjøres enkelt:

```
cat <<EOF | sudo tee /etc/dhcp/dhcpd.conf
# dhcpd.conf
#
# Configuration file for ISC dhcpd (see 'man dhcpd.conf')
#
#
authoritative;
ddns-update-style ad-hoc;
option log-servers 192.168.0.1;

allow bootp;
allow booting;
default-lease-time 86400;
max-lease-time 86400;

# LAN dhcp setup
subnet 192.168.0.0 netmask 255.255.255.0 {

# kjente klienter
group {
        option routers 192.168.0.1; # use as gateway
        option domain-name "deichman.no";
        option domain-name-servers 10.172.2.1; # deicman.no dns
        option broadcast-address 192.168.0.255;

        next-server 192.168.0.1;
        filename "/tftpboot/pxelinux.0";   # kompilert pxelinux image med nfs support
        use-host-decl-names on;
        deny unknown-clients;

# eksempel på kjent enkelt klient
host klient1 { hardware ethernet 00:08:74:15:c0:71; fixed-address 192.168.0.11; }
host klient2 { hardware ethernet 00:11:85:f1:bb:19; fixed-address 192.168.0.12; }

} #end group

# ukjente klienter får IP uten tilgang for feilsøking
pool   {
        range 192.168.0.100 192.168.0.200;
        allow unknown-clients;

} #end pool
} #end subnet
EOF
```

Det er lett å gjøre feil i dhcpd.conf-oppsettet. Ved oppstart får du ikke nødvendigvis mye info hvis det er feil, men du kan teste:

```
/usr/sbin/dhcpd -t -cf /etc/dhcp/dhcpd.conf
```

Ukjente leaser havner i /var/lib/dhcp/dhcpd.leases, så det kan være et fint sted å fange opp ukjente og uønskede tilkoblinger.

dhcpd.conf (avansert)

hvis oppsettet er mer komplisert: flere nettverk, ulike bilder og subnett som skal serves fra samme server, blir det litt mer komplisert.
Alt styres fortsatt fra dhcpd.conf, men det trengs litt mer hacking. Tips: http://www.novell.com/coolsolutions/tip/16951.html

Eksempel:

```
# dhcpd.conf
#
# Configuration file for ISC dhcpd (see 'man dhcpd.conf')
#
#
authoritative;
ddns-update-style none;

option routers 192.168.200.126;

# PXE options space -- necessary for giving options to pxe clients
       option space pxelinux;
       option pxelinux.magic       code 208 = string;
       option pxelinux.configfile   code 209 = text;
       option pxelinux.pathprefix  code 210 = text;
       option pxelinux.reboottime code 211 = unsigned integer 32;
```

```
# extra options
  # RFC3442 routes (destination_ip,route_ip)
  option rfc3442-classless-static-routes code 121 = array of integer 8;
  # MS routes
  option ms-classless-static-routes code 249 = array of integer 8;

allow bootp;
allow booting;

subnet 192.168.200.0 netmask 255.255.255.128 {
    option domain-name "deichman.no";

    # Searchstations PXE
    group {
                site-option-space "pxelinux";
        option pxelinux.magic "f1:00:74:7e";

                # Force send the PXELINUX options (specified in hexadecimal)
                if exists dhcp-parameter-request-list {
                option dhcp-parameter-request-list = concat(option dhcp-parameter-request-list,d0,d1,d2,d3,79,f9);
                }
        option pxelinux.configfile "/pxelinux.cfg/searchstations";

        deny unknown-clients;
        next-server 192.168.200.125;
        filename "pxelinux.0";
        # Musikkavdelingen

        host musikksok1 {
            hardware ethernet 00:01:2e:bc:c8:cf;
            fixed-address 192.168.200.111;
            }
        host musikksok2 {
            hardware ethernet 00:12:79:66:f8:e2;
            fixed-address 192.168.200.112;
            }
                }
}
```

Det sentrale her er at klientene må tvinges til å spørre etter diverse valg som de ellers ikke ville spurt etter.

Grub2 (utgått)

Har testet ut grub v2 som oppstartsystem, da det gir muligheten for å boote iso-bilder direkte. Men grunnet betatilstanden på grub2 og det faktum at den er svært selektiv på de ulike nettverkskortene (hele gx-serien støttes bare delvis i pxe-boot), så ble denne valgt bort.

TFTP

Thin File Transfer Protocol er den som serverer oppstartfilene, dvs. kjernen (linux kernel) og root-systemet (initrd.gz), ofte også kalt initramsys. TFTP kjører som en selvstendig tjener, tftpd, eller under initd.
NB! Merk at etter ubuntu v10.10 er default ipv6, så du må presisere ipv4 i options, ellers vil det neppe virke.
Installering i ubuntu: (rediger /etc/default/tftpd-hpa)

```
# /etc/default/tftpd-hpa
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS=""
```

evt. installer xinetd (sudo apt-get install xinetd) og lag en daemon for tftp:

```
cat <<EOF | sudo tee /etc/xinetd.d/tftp
service tftp
{
protocol        = udp
port            = 69
socket_type     = dgram
wait            = yes
user            = root
server          = /usr/sbin/in.tftpd
server_args     = -u root -s /tftpboot
disable         = no
}
EOF
```

NB! Pass på at TFTP_DIRECTORY i /etc/default/tftp-hpa stemmer overens med rotmappen som brukes til tftp, f.eks. som /tftpboot som vil bli brukt videre her. Det viktige her er å lage en mappe som skal danne rotfilsystemet for klientbildene, /tftpboot og at denne legges inn med chroot-jail, slik at alle som booter fra pxe bare får tilgang til denne mappen og undermapper.

Foreslått struktur:

```
/tftpboot/ (pxelinux.0 og andre pxeboot-filer)
  pxelinux.cfg/ (bootfiler for syslinux)
  statistics/ (innloggingsstatistikk)
  boot/
    newimages/ nye iso-bilder
    mounts/ (stier til mounting av bilder)
        clientimage
        libkiimage
```

```
        ...
```

Det kan gjøres med linux-magi slik:

```
sudo mkdir -p /tftpboot/{pxelinux.cfg,statistics,boot/{newimages,mounts/{libkiimage,clientimage,searchstation}}}
```

PXEboot

PXEboot er selve oppstartsystemet som støttes av de fleste nettverkskort, dvs. evnen til å starte fra nettverk. Dette må aktiveres i bios på maskinen som skal brukes som klient. PXE-boot bruker tjenerens dhcpd til å skaffe info om nett og oppstartsbilde.

PXElinux ligger i Syslinux-pakken som ble installert over.
må kopiere filene til /tftpboot:

```
sudo cp /usr/lib/syslinux/pxelinux.0 /tftpboot/
```

I roten av /tftpboot legges 'pxelinux.0', som er et syslinux bootbilde, som igjen peker til

/tftpboot/pxelinux.cfg/

hvor klienten først prøver å finne en menyliste som svarer til egen mac-adresse slik xx-xx-xx-xx-xx-xx, og som ender opp med å laste 'default' hvis den ikke finner noe annet. Denne inneholder så info om de forskjellige bildene som skal lastes, i syslinux-format:

```
default mycel
prompt 1
timeout 100
menu title Deichmans publikumsklienter
display message.txt
F1 message.txt
F2 f2.txt

LABEL client
KERNEL /boot/mounts/clientimage/casper/vmlinuz
APPEND initrd=/boot/mounts/clientimage/casper/initrd.gz boot=casper netboot=nfs nfsroot=192.168.0.1:/tftpboot/boot/mounts/c

LABEL libki
KERNEL /boot/mounts/mycelimage/casper/vmlinuz
APPEND initrd=/boot/mounts/mycelimage/casper/initrd.gz boot=casper netboot=nfs nfsroot=192.168.0.1:/tftpboot/boot/mounts/my

LABEL searchstation
KERNEL /boot/mounts/searchstation/casper/vmlinuz
APPEND initrd=/boot/mounts/searchstation/casper/initrd.gz boot=casper netboot=nfs nfsroot=192.168.0.1:/tftpboot/boot/mounts
```

det er også her spesifikke modul-opsjoner legges inn (i APPEND).

f.eks. ved problemer med HDMI og rekkefølge av lydkort (se klientinstall og Issue #~~166~~)

```
snd-hda-intel.index=1 snd-hda-intel.id=SD
```

message.txt angir oppstartsteksten for boot.

NFS

NFS er network file system, og brukes til alt som skal deles. I første rekke er det live-CD-klonen som klientene skal bruke, men det kan også benyttes for å lage felles 'skratsj'-område for klientene, hvis ønskelig.
NFS er enkelt å bruke, det er tre hovedlinjer:

- bildet som blir laget på klonemaskinen må lastes til tjener og legges i /tftpboot/boot/newimages/ (se klientinstall).
- .iso-bildet må mountes som loop (ro), helst i en undermappe
- undermappen må eksporteres i nfs

Først må .iso-bildet mountes

```
sudo mount -o loop,ro /tftpboot/boot/newimages/clientimage-newest.iso /tftpboot/boot/mounts/clientimage
```

Deretter må det legges en linje inn i /etc/exports:

```
/tftpboot/boot/mounts/clientimage     192.168.0.1/24(ro,sync,no_root_squash,insecure,no_subtree_check)
```

og nfsd restartes

```
/etc/init.d/nfs-kernel-server restart
```

For å sjekke at den er eksportert riktig, bruk exportfs. Merk at hvis det er flere bilder som brukes må hver mappe må eksporteres enkeltvis.

For å automatisk mounte .iso-bildet ved oppstart kan det legges inn i fstab:

```
sudo echo '/tftpboot/boot/newimages/clientimage-newest.iso /tftpboot/boot/mounts/clientimage iso9660 auto,loop,ro 0 0' >> /
```

Statistikk (utdatert, styres nå med mycel)

Vi viderebruker nfs til å lagre statistikk. Det må gjøres noen endringer i klienten for å kunne lagre på nfs. På serveren oppretter vi en mappe /tftpboot/statistics som vi bruker til å lagre all statistikk.

Denne eksporteres deretter som skrivbar for klientene (i /etc/exports):

```
/tftpboot/statistics/     192.168.0.1/24(rw,sync,no_root_squash,insecure,no_subtree_check)
```

Her vil etter hvert de ulike klientene lagre statistikk for hver dag etter hvert som det kommer inn.
I tillegg bruker vi et statistikkverktøy som heter vnstat

```
sudo apt-get install vnstat
sudo vnstat -u -i eth1
```

endre /etc/vnstats.conf for å fjerne begrensning på båndbredde:

```
MaxBandwidth 0
```

Hvis flere servere er involvert, kan en bruke rsync for å samkjøre loggfilene til en av serverne, f.eks. master-klientserveren.

master-klientserverens ip må legges inn i /etc/hosts

```
[ip]     master-klientserver
```

Lag ssh-nøkkel

```
ssh-keygen
```

og kopier denne til master-klientserveren.

```
scp /home/katalog/.ssh/id_rsa.pub master-klientserver:/home/katalog/
```

fra master-klientserveren, legg så til nøkkelen i authorized_keys

```
cat id_rsa.pub >> ~/.ssh/authorized_keys
```

Dermed kan den synke uten passord
Legg så inn i /etc/crontab en linje for å synke hver dag:

```
20 00 * * * katalog /usr/bin/rsync -av /tftpboot/statistics/ katalog@master-klientserver:/tftpboot/statistics/
```

NB: Pass på at eier av mappen /tftpboot/statistics er samme på alle servere.

Skript

Noen skript for automatisering, kan hentes via subversion.

```
svn co http://svn.deichman.no/halvtykke-klienter/server/ /usr/local/bin/
```

For å få tilgang til statistikken eksternt har jeg laget et php-skript som spytter ut rss/xml, statistics.php. I denne filen må alle klienter legges inn. Denne legges eller lenkes så sammen med statistikkfilene i /tftpboot/statistics/ og serves med en webserver, f.eks. lighttpd på master-klientserveren:

```
sudo apt-get install lighttpd php5-cgi
sudo lighty-enable-mod fastcgi
sudo lighty-enable-mod fastcgi-php
```

deretter må webroot endres i /etc/lighttpd/lighttpd.conf til /tftpboot/statistics/

og webserverent restartes:

```
sudo service lighttpd force-reload
```

nå kan statistikken hentes ut som feed via http://serverip/statistics.php

Kloning

For å klone klientservere er det kanskje best å installere ubuntu manuelt og derette bruke rsync for å kopiere tftpboot-mappen med innhold.
For å aktivere rsync over ssh på ubuntu en kjøre visudo og legge inn

```
Defaults visiblepw
```

Dette fordi ubuntu ikke som standard godtar sudo-sesjoner over ssh som krever passord.
Deretter kan en kjøre en stty-sesjon med sudo for å holde linja åpen og en rsync mellom servere. Kjør helst en test først med -n eller --dry-run for ikke å skrive over feil!

```
stty -echo; ssh user@ip "sudo -v"; stty echo
sudo rsync -n -avze ssh --rsync-path="sudo rsync"
user@ip:/tftpboot/ /tftpboot/
```

merk at hele mappestrukturen, inkludert symlinks kopieres, så det kan være lurt å unmounte klientbildene først.

Dynamisk IP

Tilslutt, en generell serversak. Skulle serverne bruke dynamisk ip, er det greit å lage en rutine for å få varsel om ipaddressen er endret, slik at du ikke mister tilgang til boksen.

Vi bruker en simpel mailagent:

```
sudo apt-get install nullmailer
```

sett lokalt hostname og velg foretrukket mail relay agent.
Deretter trenger vi et skript som kjøres i crontab, som sjekker ip mot eksisterende ip. Vi kaller det checkserverip.sh.

```
Legg inn en linje i /etc/crontab som ovenfor
00 11 * * * root /usr/local/bin/checkserverip.sh
```

Webmin

Når alt er på plass, er det greit å installere et grafisk admingrensesnitt. Webmin er det mest brukte og mest konfigurerbart. Først perl ssl:

```
sudo apt-get install libnet-ssleay-perl
sudo apt-get install libcrypt-ssleay-perl
```

Last så ned siste minimal-versjon fra http://prdownloads.sourceforge.net/webadmin/ og pakk ut:

```
wget http://prdownloads.sourceforge.net/webadmin/webmin-1.620-minimal.tar.gz
tar zxvf webmin-1.620-minimal.tar.gz
```

Installer til et ok sted med superbruker

```
cd webmin-1.620
sudo ./setup.sh /usr/local/src/webmin
```

Velg port, bruker og passord og logg inn. Merk: En webmin kan også administrere flere servere. Innlogging til webmin skjer så ved https://servernavn:port. Standardtema er litt stygt, men det kan lett repareres ved å installere Stressless theme fra webmin. Bare legg inn denne urlen:

```
http://www.stress-free.co.nz/documents/theme-stressfree.tar.gz
```

Installer og aktiver.

Kjekke webmin-moduler:
net dhcpd mount exports

Blir du lei av webmin og savner terminal, er det like lett å avinstallere som å installere:

```
sudo /etc/webmin/uninstall.sh
```

Deploy av nye bilder

For å gjøre det enkelt å deploye nye klientbilder kan webmin brukes til å kopiere bilder via "cluster copy" hvor en kan kjøre skript i forkant og etterkant for å automatisere.

Lag et enkelt start/stopp-skript for klientserverne som stopper/starter nfs-server og unmounter/mounter bildene:

```
cat <<EOF | sudo tee /usr/local/bin/clientserver.sh
#!/bin/bash
# /usr/local/bin/clientserver.sh
# start -- stop clientserver
# stops/starts nfs-server and unmounts/mount client iso images

start_clientserver() {
    echo "Starter klientserver"
    echo "============="
    mount -o loop,ro /tftpboot/boot/newimages/clientimage-newest.iso /tftpboot/boot/mounts/clientimage/
        mount -o loop,ro /tftpboot/boot/newimages/mycelimage-newest.iso /tftpboot/boot/mounts/mycelimage/
    mount -o loop,ro /tftpboot/boot/newimages/libkiimage-newest.iso /tftpboot/boot/mounts/libkiimage/
    mount -o loop,ro /tftpboot/boot/newimages/searchstation-newest.iso /tftpboot/boot/mounts/searchstation/
    /etc/init.d/nfs-kernel-server start
}

stop_clientserver() {
    echo "Stopper klientserver"
    echo "============="
    /etc/init.d/nfs-kernel-server stop
    umount /dev/loop*
}

case \$1 in
    start)
        start_clientserver
    ;;
    stop)
        stop_clientserver
    ;;
```

```
    restart)
        stop_clientserver
        start_clientserver
    ;;
    *)
    echo "Usage: clientserver {start|stop|restart}"
    exit 1
esac

exit 0
EOF
```

dette må gjøres kjørbart:

```
sudo chmod +x /usr/local/bin/clientserver.sh
```

Det kan så legges inn i "cluster copy" til å kjøre før og etter deploy av nytt/nye bilder.

Filer

Ny fil