EuroSkills Test Project

IT Network Systems Administration

(39)

Module A – Linux Environment

Submitted by:

Janos Csoke HU

Svetlana Lapenko KZ

Igors Bumanis LV

Timotej Gruden SI

Ander Guerra ES

INDEX

INTRODUCTION

The competition has a fixed start and finish time. You must decide how to best divide your time.

Please carefully read the following instructions!

When the competition time ends, please leave your station in a running state. The assessment will be done in the state as it is. No reboot will be initiated as well as powered off machines will not be powered on!

Please use the information below for all the servers and clients.

LOGIN

Username:	root	localadmin
Password:	Passw0rd!	Passw0rd!

Use the string ***Passw0rd!*** as password everywhere where a password, passphrase etc. is needed. Pay attention to the zero sign in the string that replaces the capital letter O!

SYSTEM CONFIGURATION

Region/time zone:	Europe/Copenhagen
Locale:	English US (UTF-8)
Key Map:	English US

RESOURCES

You will find the topology and the IP addressing on Appendix A. The list of the users to be created can be found on Appendix A. Both are located at the end of this document.

OPERATING SYSTEMS AND SOFTWARE

Every system in this task uses **Debian 12**. Every VM you see in the topology chart on Appendix A is preinstalled on the physical host, named accordingly. The hosts use VMware ESXi as a virtualization platform, and you can connect to the resources using your PCs with vSphere Web Client or VMware Workstation also.

You can connect to the ESXi host with esxi.local URL.

The pre-installed VMs contain only the base system and a few additional packages (see in the software section), you can install additional software using virtual optical disks.

For testing purposes, all VM have been installed with the following test tools: **smbclient, curl, lynx, dnsutils, ldap-utils, ftp, lftp, wget, nfs-common, rsync, telnet, traceroute, tcptraceroute, tcpdump, net-tools, cifs-utils**.

ssh is also installed and configured in all the machines, so all of them are accessible via SSH for easy testing. At the end of the competition time, please be sure that all the machines are accessible.

You can find Debian install ISOs in the datastore.

INDRUSTY COMPLIANCE

The test project does not always give you an exact specification. In these situations, you have the chance to choose which software to use, which path to follow - you will find information on the tasks about the paths you can choose from. The more sophisticated a solution is the better mark you are going to get for it.

DESCRIPTION OF PROJECT AND TASKS

You are a new IT Engineer of the Lego APS company. The goal of your next project is to build the whole IT infrastructure of the company, which is divided into two different sites, located in Herning and Billund cities.

INET

R-INT

All the configurations in R-INT computer are already done, and you should not touch them. Configurations Include:

R-INT task1 - IP addressing/routing

- IP addressing and routing.

apt install resolvconf

nano /etc/network/interfaces

```
  GNU nano 7.2
# This file describes the network interfaces
# and how to activate them. For more informa

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens192 ens224 ens256

iface ens192 inet static
address 203.0.113.1/29

iface ens224 inet static
address 203.0.113.9/29

iface ens256 inet static
address 203.0.114.1/25

dns-nameservers 127.0.0.1
```

systemctl restart networking

Activate routing

nano /etc/sysctl.conf

#Comment out this line
net.ipv4.ip_forward=1

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

sysctl -p

R-INT task2 - NTP

- NTP server on ntp.nordictech.dk.

```
timedatectl set-timezone Europe/Copenhagen

apt install ntpsec

nano /etc/ntpsec/ntp.conf
#comment out the line
#tos minclock 4 minsane 3

# line 34 : comment out default settings
#pool 0.debian.pool.ntp.org iburst
#pool 1.debian.pool.ntp.org iburst
#pool 2.debian.pool.ntp.org iburst
#pool 3.debian.pool.ntp.org iburst

systemctl restart ntpsec

# verify status
ntpq -p
```

R-INT task3 - DNS server

- DNS server for nordictech.dk domain.

```
apt install bind9
nanano /etc/bind/med.conf.local
zone "nordictech.dk" {
        type master;
        file "/etc/bind/db.nordictech.dk";
};
zone "billund.lego.dk" {
        type forward;
        forwarders {203.0.113.2;};
};
zone "herning.lego.dk" {
        type forward;
        forwarders {203.0.113.10;};
};

nano /etc/bind/db.nordictech.dk
$TTL    86400
@       IN      SOA     ntp.nordictech.dk. root.nordictech.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ) ; Negative Cache TTL
;
@       IN      NS      ntp.nordictech.dk.
ntp     IN      A       203.0.113.1
www     IN      A       203.0.113.1
```

systemctl restart bind9

R-INT task4 - web server

- Web server for www.nordictech.dk website.

apt install nginx

nano /var/www/html/index.nginx-debian.html

<h1>Welcome to NordicTech Denmark!</h1>

HOME

HOME task1 - Remote access VPN

1. Configure a Remote Access VPN connection to R-HQ. The VPN connection builds up automatically when starting this computer.

Step 1: Install openvpn

apt install openvpn

Step 2: certificates and keys. Assure that these files are located in /etc/openvpndirectory (created in HQ-DC task 1 & HQ-SAM-1 task 2) / ta.key created in R-HQ task 4

- CHAIN-CA.crt
- HOME.key
- HOME.crt
- ta.key

Step 3: Config file:

cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/
nano /etc/openvpn/client.conf

```
client
dev tun
#proto udp
remote 203.0.113.2 1195

ca /etc/openvpn/CHAIN-CA.crt
key /etc/openvpn/HOME.key
cert /etc/openvpn/HOME.crt
tls-auth /etc/openvpn/ta.key 1

cipher AES-256-GCM

persist-key
```

persist-tun

verb 3
log /var/log/openvpn/client.log

Step 5: enable the service (auto start when boot), start the service and check

systemctl enable openvpn@client
systemctl restart openvpn@client
systemctl status openvpn@client

```
localadmin@HOME:~$ systemctl status openvpn@client.service
● openvpn@client.service - OpenVPN connection to client
     Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled-runtime; preset: enabled)
     Active: active (running) since Sat 2025-06-14 06:52:35 EDT; 1min 57s ago
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 865 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 10)
     Memory: 3.2M
        CPU: 63ms
     CGroup: /system.slice/system-openvpn.slice/openvpn@client.service
             └─865 /usr/sbin/openvpn --daemon ovpn-client --status /run/openvpn/client.status 10 --cd /etc/openvpn --config /etc/op>

Warning: some journal files were not opened due to insufficient permissions.
lines 1-15/15 (END)
```

ip a

```
localadmin@HOME:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:9c:b2:06 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 203.0.114.50/25 brd 203.0.114.127 scope global noprefixroute ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::34d4:8c6c:b6ac:d69f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.39.0.6 peer 10.39.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::2018:3447:faa1:ccea/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
localadmin@HOME:~$
```

ip r

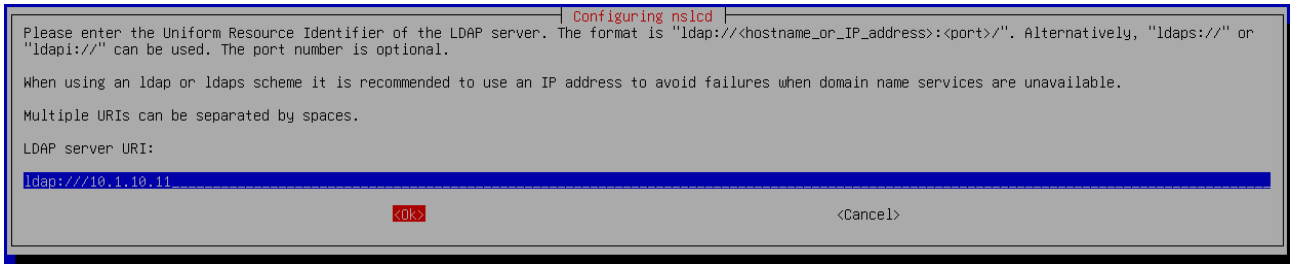
```
       valid_lft forever preferred_lft forever
localadmin@HOME:~$ ip r
default via 203.0.114.1 dev ens192 proto static metric 100
10.1.10.0/24 via 10.39.0.5 dev tun0
10.1.20.0/24 via 10.39.0.5 dev tun0
10.1.30.0/24 via 10.39.0.5 dev tun0
10.39.0.1 via 10.39.0.5 dev tun0
10.39.0.5 dev tun0 proto kernel scope link src 10.39.0.6
203.0.114.0/25 dev ens192 proto kernel scope link src 203.0.114.50 metric 100
localadmin@HOME:~$
```

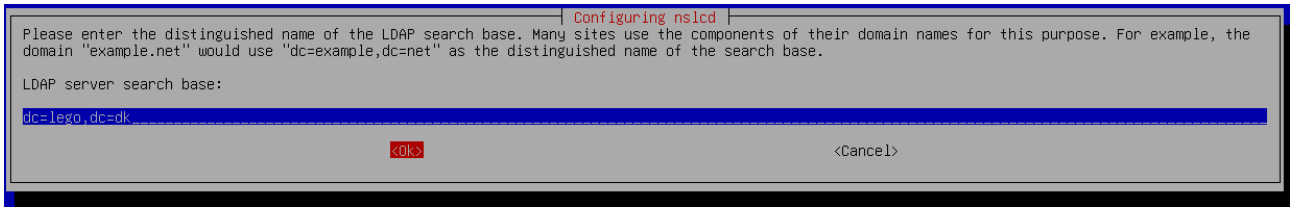
Now communication exists from HOME to HQ networks.

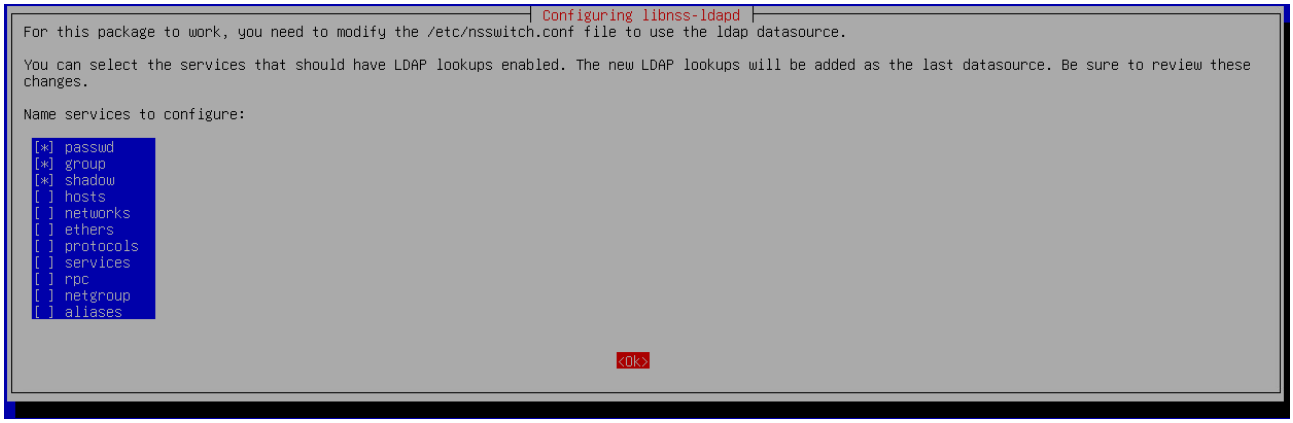
HOME task2 - LDAP auth

2. Configure LDAP authentication using the LDAP server. Login with frida. Allow fallback to local account if VPN is down.

apt -y install libnss-ldapd libpam-ldapd ldap-utils





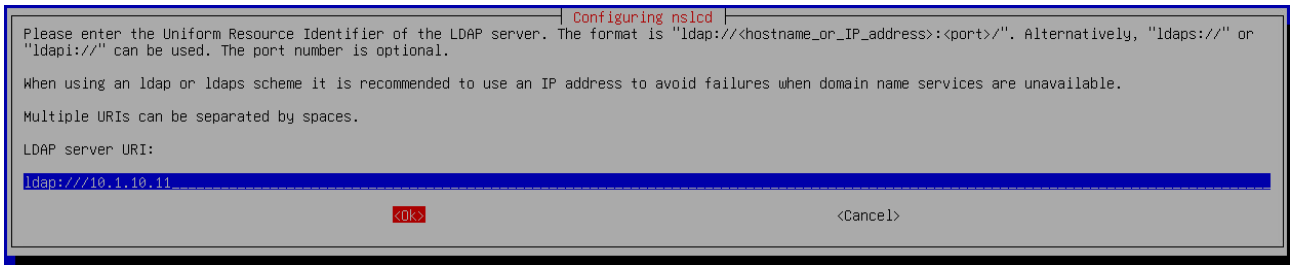


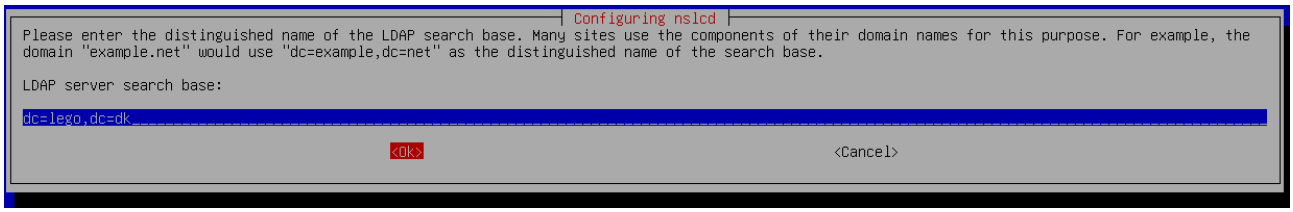
nano /etc/pam.d/common-session

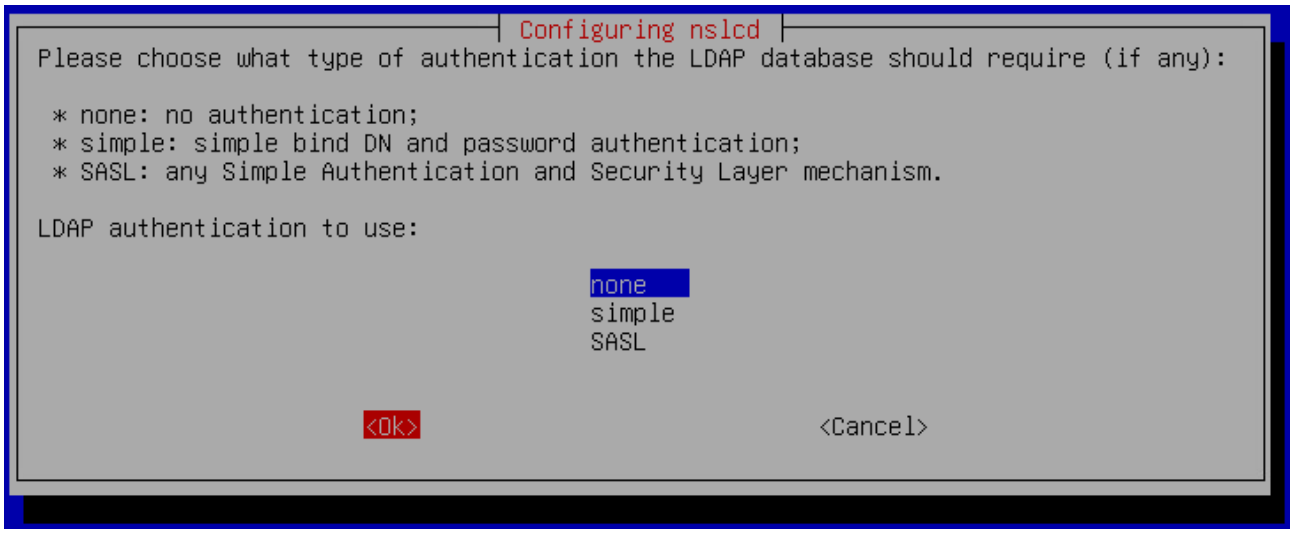
add to the end if need (create home directory automatically at initial login)

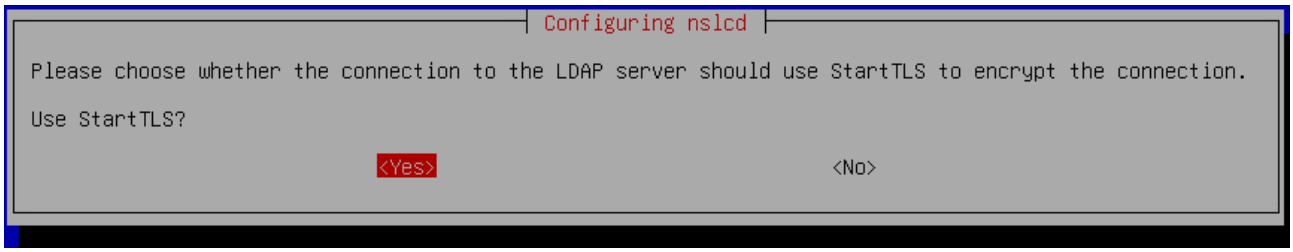
session optional pam_mkhomedir.so skel=/etc/skel umask=077

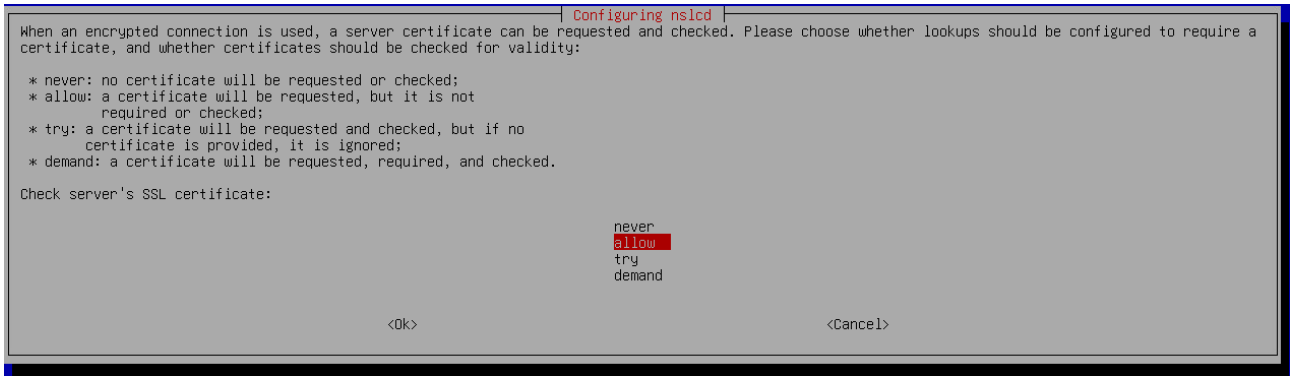
dpkg-reconfigure nslcd



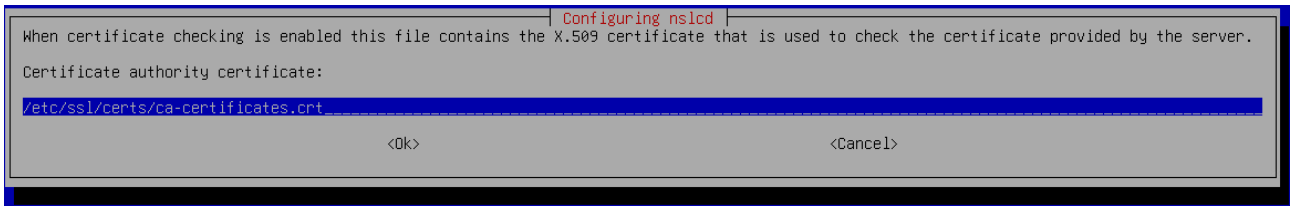








We can leave /etc/ssl/certs/ca-certificates.crt if we have added CA.crt and SUBCA.crt to system.



nano /etc/nslcd.conf

should be something like this:

```
# /etc/nslcd.conf
# nslcd configuration file. See nslcd.conf(5)
# for details.

# The user and group nslcd should run as.
uid nslcd
gid nslcd

# The location at which the LDAP server(s) should be reachable.
uri ldap:///10.1.10.11

# The search base that will be used for all queries.
base dc=lego,dc=dk

# The LDAP protocol version to use.
#ldap_version 3

# The DN to bind with for normal lookups.
#binddn cn=annonymous,dc=example,dc=net
#bindpw secret

# The DN used for password modifications by root.
#rootpwmoddn cn=admin,dc=example,dc=com

# SSL options
ssl start_tls
tls_reqcert allow
tls_cacertfile /etc/ssl/certs/ca-certificates.crt

# The search scope.
#scope sub
```

systemctl restart nscd nslcd

Check:

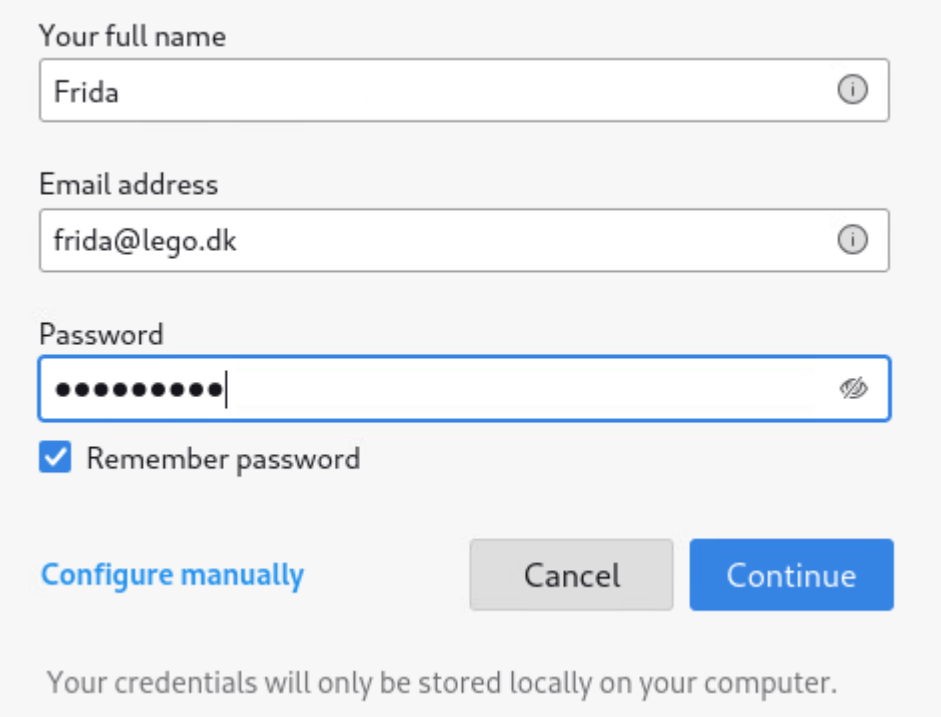
getent passwd | grep 400

```
root@HOME:~# getent passwd |grep 400
admin:x:4001:5001:admin:/home/admin:/bin/bash
frida:x:4002:5002:frida:/home/frida:/bin/bash
ella:x:4003:5002:ella:/home/ella:/bin/bash
carl:x:4004:5003:carl:/home/carl:/bin/bash
root@HOME:~#
```

HOME task3 - Thunderbird email client

3. Install Thunderbird e-mail client to use with frida@lego.dk. Send email to ella.

apt install thunderbird



HOME task4 - Filezilla

4. Install Filezilla Client and preconfigure a profile to access the SFTP server on BR-SRV.

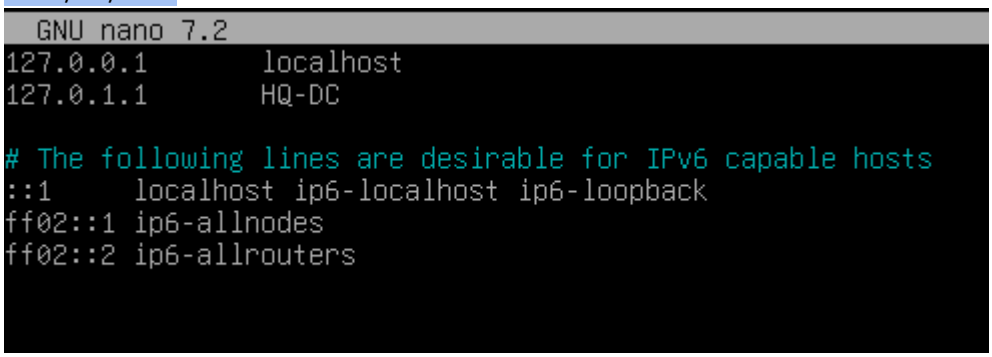
General configuration

Set up the following on all the machines in HQ site and in R-BR:

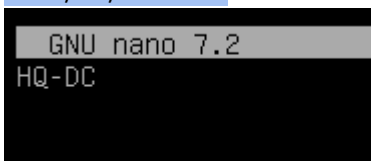
General config task1 - hostname

- hostname (example with HQ-DC)

nano /etc/hosts



nano /etc/hostname



reboot

General config task2 - NetConfig

- network configuration (example with HQ-DC)

apt install resolvconf

nano /etc/network/interfaces

auto ens192
iface ens192 inet static
address 10.1.10.11/24
gateway 10.1.10.1
dns-nameservers 10.1.20.11

```
  GNU nano 7.2
# This file describes the network interfaces available on
# and how to activate them. For more information, see inte

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens192
iface ens192 inet static
address 10.1.10.11/24
gateway 10.1.10.1
dns-nameservers 10.1.20.11
```

systemctl restart networking

- example with R-HQ

apt install resolvconf

nano /etc/network/interfaces

```
  GNU nano 7.2
# This file describes the network interfaces av
# and how to activate them. For more informatio

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens161 ens192 ens224 ens256

iface ens224 inet static
address 10.1.10.1/24

iface ens256 inet static
address 10.1.20.1/24

iface ens161 inet static
address 10.1.30.1/24

iface ens192 inet static
address 203.0.113.2/29
gateway 203.0.113.1
dns-nameservers 10.1.20.11
```

R-BR

```
  GNU nano 7.2
# This file describes the network interfaces
# and how to activate them. For more informat

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens224 ens192 ens256

iface ens224 inet static
address 10.2.10.1/24

iface ens256 inet static
address 10.2.30.1/24

iface ens192 inet static
address 203.0.113.10/29
gateway 203.0.113.9
dns-nameservers 203.0.113.9
```

General config task3 - timezone

- time zone

timedatectl set-timezone Europe/Copenhagen

General config task4 - keyboard layout

- keyboard layout

already configured

General config task5 - NTP

Install and configure NTP server service on R-HQ and R-BR, synchronized to ntp.nordictech.dk. Sync all the machines on HQ site to R-HQ and sync BR-SRV to R-BR.

R-HQ & R-BR

apt install ntpsec

nano /etc/ntpsec/ntp.conf

#DO NOT comment out the line
tos minclock 4 minsane 3

line 34 : comment out default settings and add R-INT as NTP server
#pool 0.debian.pool.ntp.org iburst
#pool 1.debian.pool.ntp.org iburst
#pool 2.debian.pool.ntp.org iburst
#pool 3.debian.pool.ntp.org iburst
pool ntp.nordictech.dk iburst

#if DNS is not working use this line instead
pool 203.0.113.1 iburst

systemctl restart ntpsec

verify status
ntpq -p

```
root@R-HQ:~# ntpq -p
     remote                                refid      st t when poll reach   delay   offset   jitter
==============================================================================================================
 203.0.113.1                          .POOL.          16 p    -  256    0   0.0000   0.0000   0.0001
 203.0.113.1                          .INIT.          16 u    - 1024    0   0.0000   0.0000   0.0001
root@R-HQ:~# _
```

HQ-DC and the rest of the NTP clients

apt install ntpsec

nano /etc/ntpsec/ntp.conf
#DO NOT comment out the line
tos minclock 4 minsane 3

line 34 : comment out default settings and add R-HQ as NTP server
#pool 0.debian.pool.ntp.org iburst
#pool 1.debian.pool.ntp.org iburst
#pool 2.debian.pool.ntp.org iburst
#pool 3.debian.pool.ntp.org iburst
pool 10.1.10.1 iburst

systemctl restart ntpsec

verify status
ntpq -p

```
root@HQ-DC:~# ntpq -p
     remote                                refid      st t when poll reach   delay   offset   jitter
==============================================================================================================
 10.1.10.1                            .POOL.          16 p    -  256    0   0.0000   0.0000   0.0001
 10.1.10.1                            .INIT.          16 u    -   64    0   0.0000   0.0000   0.0001
root@HQ-DC:~#
```

HQ site (Billund)

This is the company's headquarter site, located in Billund, with limited server services and clients.

R-HQ

This is the edge router and firewall of the HQ site. For this reason, it should allow devices to reach each other between network segments and the Internet.

You must configure the services of this server according to the following requirements.

R-HQ task1 - FW masquerade

1. Traffic from networks on HQ site cannot go to the Internet using private addressing.

Activate routing

nano /etc/sysctl.conf

#Comment out this line
net.ipv4.ip_forward=1

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

sysctl -p

Enable nftables

systemctl enable nftables

```
root@R-HQ:~# systemctl enable nftables.service
Created symlink /etc/systemd/system/sysinit.target.wants/nftables.service → /lib/systemd/system/nftables.service.
root@R-HQ:~# systemctl status nftables.service
● nftables.service - nftables
     Loaded: loaded (/lib/systemd/system/nftables.service; enabled; preset: enabled)
     Active: inactive (dead)
       Docs: man:nft(8)
             http://wiki.nftables.org
```

Prepare nftables configuration file. Add NAT table to /etc/nftables.conf file

cat /etc/nftables.conf

```
root@R-HQ:~# cat /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority filter;
        }
        chain forward {
                type filter hook forward priority filter;
        }
        chain output {
                type filter hook output priority filter;
        }
}
root@R-HQ:~# _
```

cat /usr/share/doc/nftables/examples/nat.nft >> /etc/nftables.conf
cat /etc/nftables.conf

```
root@R-HQ:~# cat /usr/share/doc/nftables/examples/nat.nft >> /etc/nftables.conf
root@R-HQ:~# cat /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority filter;
        }
        chain forward {
                type filter hook forward priority filter;
        }
        chain output {
                type filter hook output priority filter;
        }
}
#!/usr/sbin/nft -f

table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0;

                #Thanks to nftables maps, if you have a previous iptables NAT (destination NAT) ruleset like this:
                # % iptables -t nat -A PREROUTING -p tcp --dport 1000 -j DNAT --to-destination 1.1.1.1:1234
                # % iptables -t nat -A PREROUTING -p udp --dport 2000 -j DNAT --to-destination 2.2.2.2:2345
                # % iptables -t nat -A PREROUTING -p tcp --dport 3000 -j DNAT --to-destination 3.3.3.3:3456

                # It can be easily translated to nftables in a single line:

                dnat tcp dport map { 1000 : 1.1.1.1, 2000 : 2.2.2.2, 3000 : 3.3.3.3} \
                                : tcp dport map { 1000 : 1234, 2000 : 2345, 3000 : 3456 }
        }

        chain postrouting {
                type nat hook postrouting priority 0;

                #Likewise, in iptables NAT (source NAT):
                # % iptables -t nat -A POSTROUTING -s 192.168.1.1 -j SNAT --to-source 1.1.1.1
                # % iptables -t nat -A POSTROUTING -s 192.168.2.2 -j SNAT --to-source 2.2.2.2
                # % iptables -t nat -A POSTROUTING -s 192.168.3.3 -j SNAT --to-source 3.3.3.3

                # Translated to a nftables one-liner:

                snat ip saddr map { 192.168.1.1 : 1.1.1.1, 192.168.2.2 : 2.2.2.2, 192.168.3.3 : 3.3.3.3 }
        }
}
root@R-HQ:~#
```

nano /etc/nftables.conf

```
....
table ip nat {
        chain postrouting {
                type nat hook postrouting priority 0;

                oif ens192 ip saddr {10.1.10.0/24, 10.1.20.0/24, 10.1.30.0/24} masquerade;
        }
}
```

systemctl restart nftables

R-HQ task2 - FW port forwarding + traffic

2. Make sure the public services (DNS, mail, web) of the HQ site are accessible from the internet. Configure firewall with nftables. Incoming packets should be dropped by default. Allow minimal traffic for the services to work.

nano /etc/nftables.conf

```
#!/usr/sbin/nft -f

flush ruleset

table ip filter{
        chain input {
                type filter hook input priority filter;
                policy drop;
                ct state established,related accept;
                udp dport {123} accept;
                tcp dport {22} accept;
        }
                chain forward {
                type filter hook forward priority filter;
                policy drop;
                ct state established,related accept;
                udp dport 53 accept;
                tcp dport { 53, 80, 443, 143, 587 } accept;
        }
        chain output {
                type filter hook output priority filter;
                policy accept;
        }

table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0;

                iif ens192 dnat udp dport map {53 : 10.1.20.11};
```

```
                iif ens192 dnat tcp dport map {53 : 10.1.20.11};
                iif ens192 dnat tcp dport map {80 : 10.1.20.11, 443 : 10.1.20.11};
                iif ens192 dnat tcp dport map {143 : 10.1.20.11, 587: 10.1.20.11};
        }

        chain postrouting {
                type nat hook postrouting priority 0;

                oif ens192 ip saddr {10.1.10.0/24, 10.1.20.0/24, 10.1.30.0/24} masquerade;
        }
}
```

systemctl restart nftables

R-HQ task3 - Site-to-site VPN

3. Ensure a secure channel between HQ and BRANCH sites. If this channel breaks, the clients of the HQ site can access the public services of BRANCH.

Step 1: Install openvpn

apt install openvpn

Step 2: certificates and keys. Assure that these files are located in /etc/openvpndirectory (created in HQ-DC task 1 & HQ-SAM-1 task 2)

- CHAIN-CA.crt
- R-HQ.key
- R-HQ.crt

Step 3: create DH and ta.key and place them in /etc/openvpn directory:

openssl dhparam -out dh2048.pem 2048
openvpn --genkey secret ta.key

Transfer ta.key to R-BR via scp:

scp /etc/openvpn/ta.key root@203.0.113.10:/etc/openvpn/

Step 4: Config file:

nano /etc/openvpn/s2s.conf

```
dev tun
#proto udp
#local 203.0.113.2
#port 1194
```

ifconfig 10.8.0.1 10.8.0.2
route 10.2.10.0 255.255.255.0
route 10.2.30.0 255.255.255.0
topology p2p

ca /etc/openvpn/CHAIN-CA.crt
key /etc/openvpn/R-HQ.key
cert /etc/openvpn/R-HQ.crt
dh /etc/openvpn/dh2048.pem
tls-crypt /etc/openvpn/ta.key
tls-server

cipher AES-256-GCM

keepalive 10 60
verb 3
log /var/log/openvpn/s2s.log

Step 5: enable the service (auto start when boot), start the service and check

systemctl enable openvpn@s2s
systemctl restart openvpn@s2s
systemctl status openvpn@s2s

```
root@R-HQ:/etc/openvpn# systemctl status openvpn@s2s
● openvpn@s2s.service - OpenVPN connection to s2s
     Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-06-14 11:25:46 CEST; 1min 42s ago
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 128152 (openvpn)
     Status: "Pre-connection initialization successful"
      Tasks: 1 (limit: 10)
     Memory: 1.5M
        CPU: 82ms
     CGroup: /system.slice/system-openvpn.slice/openvpn@s2s.service
             └─128152 /usr/sbin/openvpn --daemon ovpn-s2s --status /run/openvpn/s2s.status 10 --cd /etc/openvpn --config /etc/openvpn/s2s.conf --writepid /run/

Jun 14 11:25:46 R-HQ systemd[1]: Starting openvpn@s2s.service - OpenVPN connection to s2s...
Jun 14 11:25:46 R-HQ systemd[1]: Started openvpn@s2s.service - OpenVPN connection to s2s.
lines 1-16/16 (END)
```

ip a

```
root@R-HQ:/etc/openvpn# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens161: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:25 brd ff:ff:ff:ff:ff:ff
    altname enp4s0
    inet 10.1.30.1/24 brd 10.1.30.255 scope global ens161
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd25/64 scope link
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:07 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 203.0.113.2/29 brd 203.0.113.7 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd07/64 scope link
       valid_lft forever preferred_lft forever
4: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:11 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 10.1.10.1/24 brd 10.1.10.255 scope global ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd11/64 scope link
       valid_lft forever preferred_lft forever
5: ens256: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:1b brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    inet 10.1.20.1/24 brd 10.1.20.255 scope global ens256
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd1b/64 scope link
       valid_lft forever preferred_lft forever
68: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::4b01:2258:6fd3:3c5b/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
root@R-HQ:/etc/openvpn# _
```

ip r

```
root@R-HQ:/etc/openvpn# ip r
default via 203.0.113.1 dev ens192 onlink
10.1.10.0/24 dev ens224 proto kernel scope link src 10.1.10.1
10.1.20.0/24 dev ens256 proto kernel scope link src 10.1.20.1
10.1.30.0/24 dev ens161 proto kernel scope link src 10.1.30.1
10.2.10.0/24 via 10.8.0.2 dev tun0
10.2.30.0/24 via 10.8.0.2 dev tun0
10.8.0.2 dev tun0 proto kernel scope link src 10.8.0.1
203.0.113.0/29 dev ens192 proto kernel scope link src 203.0.113.2
root@R-HQ:/etc/openvpn#
```

Go to R-BR task 3

R-HQ task4 - Remote access VPN

4. Configure a remote access VPN service for remote workers. Make sure that VPN clients can access the same resources as the clients of the HQ site.

Step 1: Install openvpn (maybe already installed in task 3)

apt install openvpn

Step 2: certificates and keys. Assure that these files are located in /etc/openvpndirectory (created in HQ-DC task 1 & HQ-SAM-1 task 2)

- CHAIN-CA.crt
- R-HQ.key
- R-HQ.crt

Step 3: create DH and ta.key and place them in /etc/openvpn directory (maybe already created in task 3):

openssl dhparam -out dh2048.pem 2048
openvpn --genkey secret ta.key

Transfer ta.key to HOME via scp:

scp /etc/openvpn/ta.key root@203.0.113.50:/etc/openvpn/

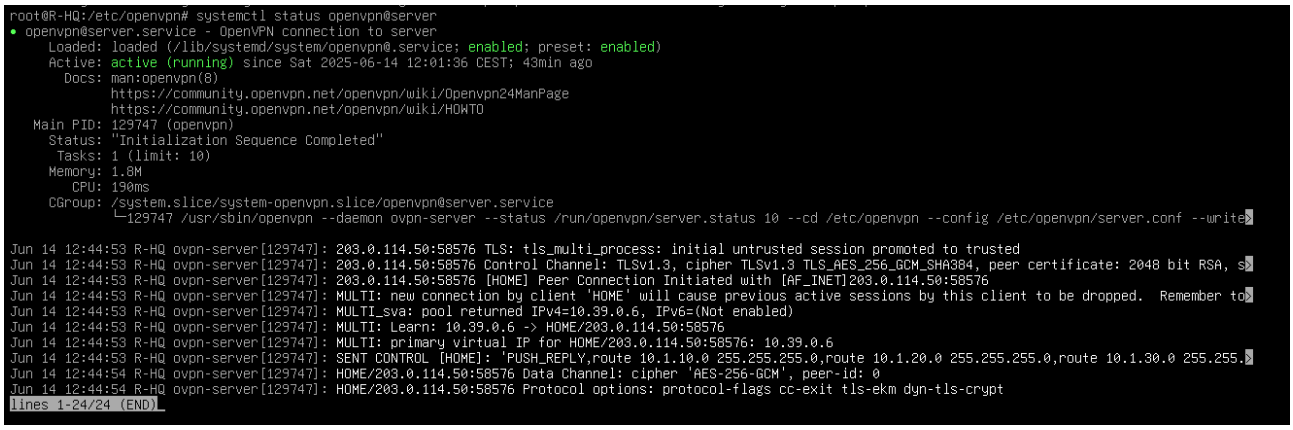
Step 4: Config file:

cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf /etc/openvpn/
nano /etc/openvpn/s2s.conf

```
dev tun
#proto udp
local 203.0.113.2
port 1195

ca /etc/openvpn/CHAIN-CA.crt
key /etc/openvpn/R-HQ.key
cert /etc/openvpn/R-HQ.crt
dh /etc/openvpn/dh2048.pem
tls-auth /etc/openvpn/ta.key 0

server 10.39.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt

push “route 10.1.10.0 255.255.255.0”
push “route 10.2.10.0 255.255.255.0”
push “route 10.3.10.0 255.255.255.0”

cipher AES-256-GCM

persist-key
persist-tun

status /var/log/openvpn/openvpn-status.log

verb 3
log /var/log/openvpn/server.log
```

Step 5: enable the service (auto start when boot), start the service and check

systemctl enable openvpn@server
systemctl restart openvpn@server
systemctl status openvpn@server



ip a

```
root@R-HQ:/etc/openvpn# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens161: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:25 brd ff:ff:ff:ff:ff:ff
    altname enp4s0
    inet 10.1.30.1/24 brd 10.1.30.255 scope global ens161
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd25/64 scope link
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:07 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 203.0.113.2/29 brd 203.0.113.7 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd07/64 scope link
       valid_lft forever preferred_lft forever
4: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:11 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 10.1.10.1/24 brd 10.1.10.255 scope global ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd11/64 scope link
       valid_lft forever preferred_lft forever
5: ens256: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:06:bd:1b brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    inet 10.1.20.1/24 brd 10.1.20.255 scope global ens256
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe06:bd1b/64 scope link
       valid_lft forever preferred_lft forever
77: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::2afc:716b:1264:b3a2/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
81: tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.39.0.1 peer 10.39.0.2/32 scope global tun1
       valid_lft forever preferred_lft forever
    inet6 fe80::dde:3f65:4da8:7f12/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
root@R-HQ:/etc/openvpn#
```

ip r

```
root@R-HQ:/etc/openvpn# ip r
default via 203.0.113.1 dev ens192 onlink
10.1.10.0/24 dev ens224 proto kernel scope link src 10.1.10.1
10.1.20.0/24 dev ens256 proto kernel scope link src 10.1.20.1
10.1.30.0/24 dev ens161 proto kernel scope link src 10.1.30.1
10.2.10.0/24 via 10.8.0.2 dev tun0
10.2.30.0/24 via 10.8.0.2 dev tun0
10.8.0.2 dev tun0 proto kernel scope link src 10.8.0.1
10.39.0.0/24 via 10.39.0.2 dev tun1
10.39.0.2 dev tun1 proto kernel scope link src 10.39.0.1
203.0.113.0/29 dev ens192 proto kernel scope link src 203.0.113.2
root@R-HQ:/etc/openvpn#
```

Go to HOME task 1

HQ-DC

HQ-DC task1 - Root CA

1. Create a root certificate authority. Use the next parameters:
   C=DK, O=Lego APS, CN=Lego APS Root CA
   Place CA.crt CA certificate file in /ca folder.
   This Root CA must only be used to sign the Subordinate CA certificate located on HQ-SAM-1

Step 1: Create a private key for the Root CA

```
mkdir -p /ca && cd /ca
######openssl genrsa -out CA.key 4096
openssl genrsa -out CA.key
```

Step 2: Create Certificate of the Root CA

```
####openssl req -x509 -new -nodes -key CA.key -sha256 -days 3650 -subj "/C=DK/O=Lego APS/CN=Lego APS Root CA"
-out CA.crt
openssl req -x509 -new -nodes -key CA.key -subj "/C=DK/O=Lego APS/CN=Lego APS Root CA" -out CA.crt
```

Step 3: Create a private key for the Intermediate CA

```
####openssl genrsa -out SUBCA.key 4096
openssl genrsa -out SUBCA.key
```

Step 2: Create a CSR (Certificate Signing Request) for the Intermediate CA

```
openssl req -new -key SUBCA.key -subj "/C=DK/O=Lego APS/CN=Lego APS Intermediate CA" -out SUBCA.csr
```

Step 5: Sign CSR of the Intermediate CA using Root CA and get Certificate of the Intermediate CA

```
####openssl x509 -req -in SUBCA.csr -CA CA.crt -CAkey CA.key -CAcreateserial -out SUBCA.crt -days 3650 -extfile
<(printf “basicConstraints=CA:true”)
openssl x509 -req -in SUBCA.csr -CA CA.crt -CAkey CA.key -CAcreateserial -out SUBCA.crt -extfile <(echo
“basicConstraints=CA:true”)
```

note: basic constraints CA=true is for trusting all the chain

Step 6: In some configuration we will need a certificate chain joining CA.crt and SUBCA.crt. Create it:

```
cat SUBCA.crt CA.crt > CHAIN-CA.crt
```

Step 6: Copy Intermediate CA files (SUBCA.csr, SUBCA.key, SUBCA.crt) and CA.crt

to HQ-SAM-1

***NOTE: Previously, /ca directory should be created on HQ-SAM-1*

```
scp SUBCA* root@10.1.10.21:/ca/

scp CA.crt root@10.1.10.21:/ca/
```

Step 7: Add the Root CA and the Intermediate CA certificate to the trusted root certificates

```
apt install -y ca-certificates
cp *.crt /usr/local/share/ca-certificates
update-ca-certificates
```

Repeat steps 6 (copy certs to another machine) and 7 (trust root and intermediate certificates) in each of these machines.

Go to HQ-SAM-1 to create server certificates.

HQ-DC task2 -LDAP server

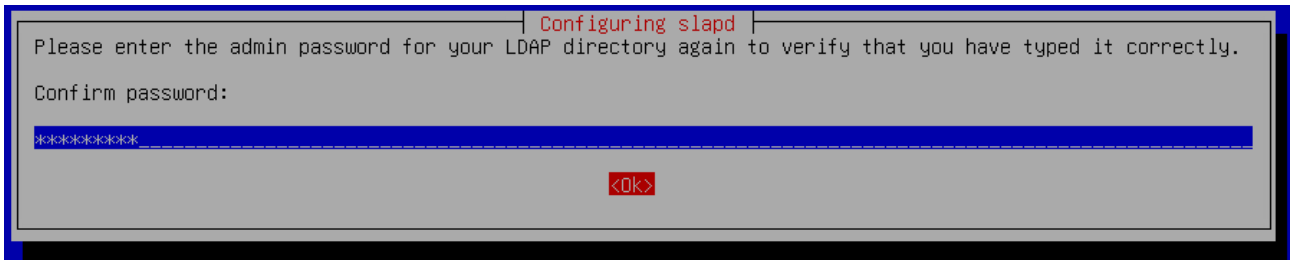
2. Deploy a directory service with LDAP protocol and SSL/TLS security. Create all objects listed in Appendix B.

Step 1: Following certificates and keys should be in HQ-DC machine:

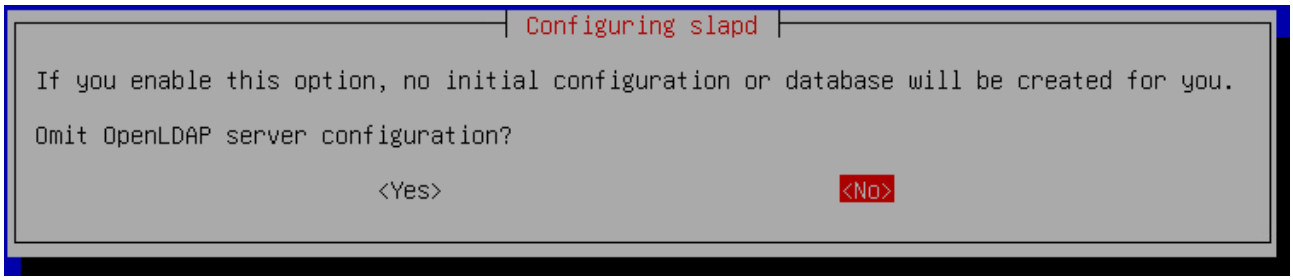
- SUBCA.crt / CHAIN-CA.crt
- HQ-DC.key
- HQ-DC.crt

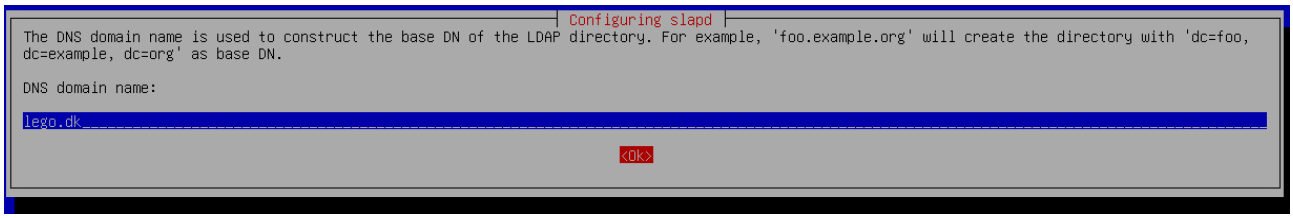
Step 1: Install LDAP service and first configuration

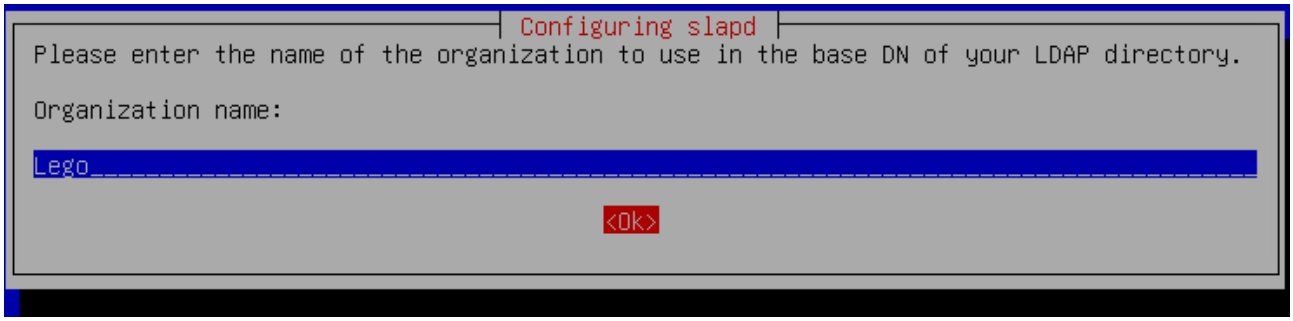
```
apt install slapd
```

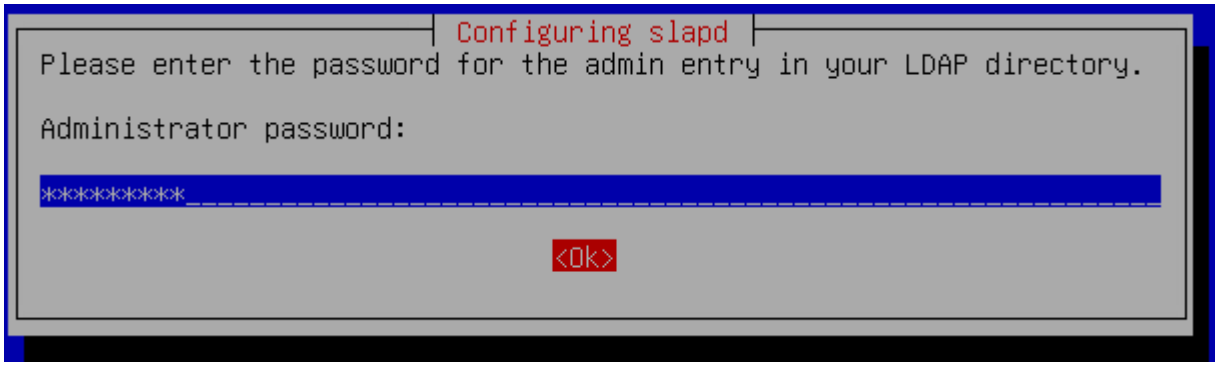


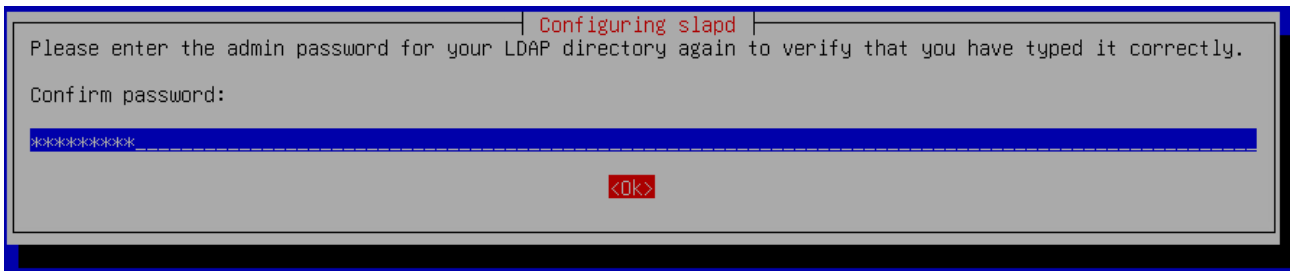
dpkg-reconfigure slapd

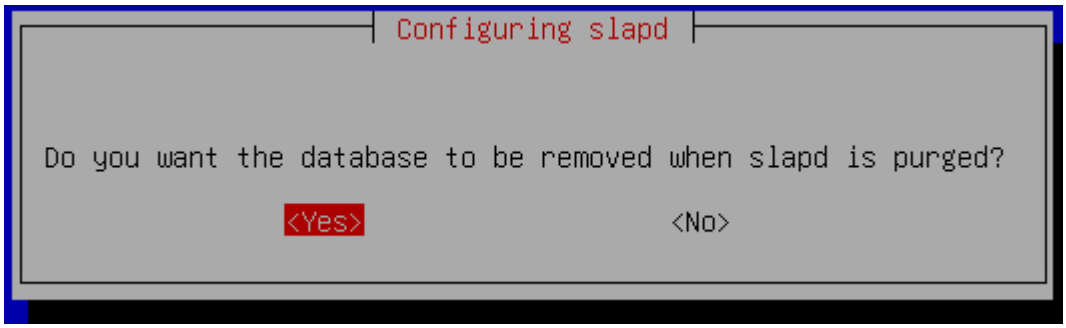


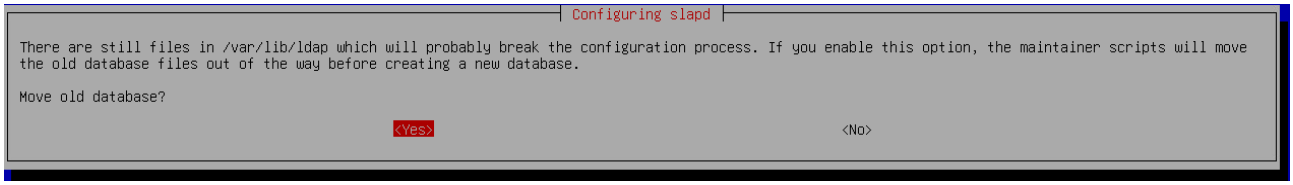












Step 2: Add OUs

nano ous.ldif

```
#ous.ldif
dn: ou=Billund,dc=lego,dc=dk
objectClass: organizationalUnit
ou: Billund

dn: ou=Herning,dc=lego,dc=dk
objectClass: organizationalUnit
ou: Herning
```

ldapadd -x -D cn=admin,dc=lego,dc=dk -W -f ous.ldif

Check

ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=organizationalUnit)"

```
root@HQ-DC:~/ldap# ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=organizationalUnit)"
# extended LDIF
#
# LDAPv3
# base <dc=lego,dc=dk> with scope subtree
# filter: (objectClass=organizationalUnit)
# requesting: ALL
#

# Billund, lego.dk
dn: ou=Billund,dc=lego,dc=dk
objectClass: organizationalUnit
ou: Billund

# Herning, lego.dk
dn: ou=Herning,dc=lego,dc=dk
objectClass: organizationalUnit
ou: Herning

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
root@HQ-DC:~/ldap# _
```

ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=organizationalUnit)" | grep dn

```
root@HQ-DC:~/ldap# ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=organizationalUnit)" | grep dn
dn: ou=Billund,dc=lego,dc=dk
dn: ou=Herning,dc=lego,dc=dk
```

Step 3: Add users

touch ldapusers.ldif
slappasswd > ldapusers.ldif

nano ldapusers.ldif

#ldapusers.ldif
dn: uid=admin,ou=Billund,dc=lego,dc=dk
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: admin
sn: admin
userPassword: {SSHA}cUp80QP8sSTJBPoYBhOE2da7zOS69JhT
loginShell: /bin/bash
homeDirectory: /home/admin
uidNumber: 4001
gidNumber: 5001

dn: uid=frida,ou=Billund,dc=lego,dc=dk

```
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: frida
sn: frida
userPassword: {SSHA}cUp80QP8sSTJBPoYBhOE2da7zOS69JhT
loginShell: /bin/bash
homeDirectory: /home/frida
uidNumber: 4002
gidNumber: 5002

dn: uid=ella,ou=Billund,dc=lego,dc=dk
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: ella
sn: ella
userPassword: {SSHA}cUp80QP8sSTJBPoYBhOE2da7zOS69JhT
loginShell: /bin/bash
homeDirectory: /home/ella
uidNumber: 4003
gidNumber: 5002

dn: uid=carl,ou=Herning,dc=lego,dc=dk
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: carl
sn: carl
userPassword: {SSHA}cUp80QP8sSTJBPoYBhOE2da7zOS69JhT
loginShell: /bin/bash
homeDirectory: /home/carl
uidNumber: 4004
gidNumber: 5003
```

```
ldapadd -x -D cn=admin,dc=lego,dc=dk -W -f ldapusers.ldif
```

Check

```
ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=inetOrgPerson)"
```

```
uidNumber: 4001
gidNumber: 5001
uid: admin

# frida, Billund, lego.dk
dn: uid=frida,ou=Billund,dc=lego,dc=dk
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: frida
sn: frida
loginShell: /bin/bash
homeDirectory: /home/frida
uidNumber: 4002
gidNumber: 5002
uid: frida

# ella, Billund, lego.dk
dn: uid=ella,ou=Billund,dc=lego,dc=dk
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: ella
sn: ella
loginShell: /bin/bash
homeDirectory: /home/ella
uidNumber: 4003
gidNumber: 5002
uid: ella

# carl, Herning, lego.dk
dn: uid=carl,ou=Herning,dc=lego,dc=dk
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: carl
sn: carl
loginShell: /bin/bash
homeDirectory: /home/carl
uidNumber: 4004
gidNumber: 5003
uid: carl

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 4
root@HQ-DC:~/ldap# _
```

ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=inetOrgPerson)" | grep dn

```
root@HQ-DC:~/ldap# ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=inetOrgPerson)" |grep dn
dn: uid=admin,ou=Billund,dc=lego,dc=dk
dn: uid=frida,ou=Billund,dc=lego,dc=dk
dn: uid=ella,ou=Billund,dc=lego,dc=dk
dn: uid=carl,ou=Herning,dc=lego,dc=dk
root@HQ-DC:~/ldap# _
```

Step 4: Add groups

```
nano ldapgroups.ldif
```

```
#ldapgroups.ldif
dn: cn=admins,ou=Billund,dc=lego,dc=dk
objectClass: posixGroup
cn: admins
gidNumber: 5001
memberUid: admin

dn: cn=billund,ou=Billund,dc=lego,dc=dk
objectClass: posixGroup
cn: billund
gidNumber: 5002
memberUid: frida,ella

dn: cn=herning,ou=Herning,dc=lego,dc=dk
objectClass: posixGroup
cn: herning
gidNumber: 5003
memberUid: carl
```

```
ldapadd -x -D cn=admin,dc=lego,dc=dk -W -f ldapgroups.ldif
```

Check

```
ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=posixGroup)"
```

```
root@HQ-DC:~/ldap# ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=posixGroup)"
# extended LDIF
#
# LDAPv3
# base <dc=lego,dc=dk> with scope subtree
# filter: (objectClass=posixGroup)
# requesting: ALL
#

# admins, Billund, lego.dk
dn: cn=admins,ou=Billund,dc=lego,dc=dk
objectClass: posixGroup
cn: admins
gidNumber: 5001
memberUid: admin

# billund, Billund, lego.dk
dn: cn=billund,ou=Billund,dc=lego,dc=dk
objectClass: posixGroup
cn: billund
gidNumber: 5002
memberUid: frida,ella

# herning, Herning, lego.dk
dn: cn=herning,ou=Herning,dc=lego,dc=dk
objectClass: posixGroup
cn: herning
gidNumber: 5003
memberUid: carl

# search result
search: 2
result: 0 Success

# numResponses: 4
# numEntries: 3
root@HQ-DC:~/ldap# _
```

ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=posixGroup)" | grep dn

```
root@HQ-DC:~/ldap# ldapsearch -x -b "dc=lego,dc=dk" -H ldap://localhost "(objectClass=posixGroup)" | grep dn
dn: cn=admins,ou=Billund,dc=lego,dc=dk
dn: cn=billund,ou=Billund,dc=lego,dc=dk
dn: cn=herning,ou=Herning,dc=lego,dc=dk
root@HQ-DC:~/ldap# _
```

Step 5: configure LDAP over SSL/TLS

copy HQ-DC.key and HQ-DC.crt to /etc/ldap/sasl2/

chown openldap:openldap /etc/ldap/sasl2/*

ls -la /etc/ldap/sasl2/

```
root@HQ-DC:~/ldap# ls -la /etc/ldap/sasl2/
total 20
drwxr-xr-x 2 root     root     4096 Jun 14 20:08 .
drwxr-xr-x 5 root     root     4096 Jun 14 19:06 ..
-rw-r--r-- 1 openldap openldap 1094 Jun 14 20:08 HQ-DC.crt
-rw-r--r-- 1 openldap openldap  928 Jun 14 20:08 HQ-DC.csr
-rw------- 1 openldap openldap 1704 Jun 14 20:08 HQ-DC.key
root@HQ-DC:~/ldap# _
```

nano mod_ssl.ldif

```
# create new
dn: cn=config
changetype: modify
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/sasl2/HQ-DC.crt
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/sasl2/HQ-DC.key
```

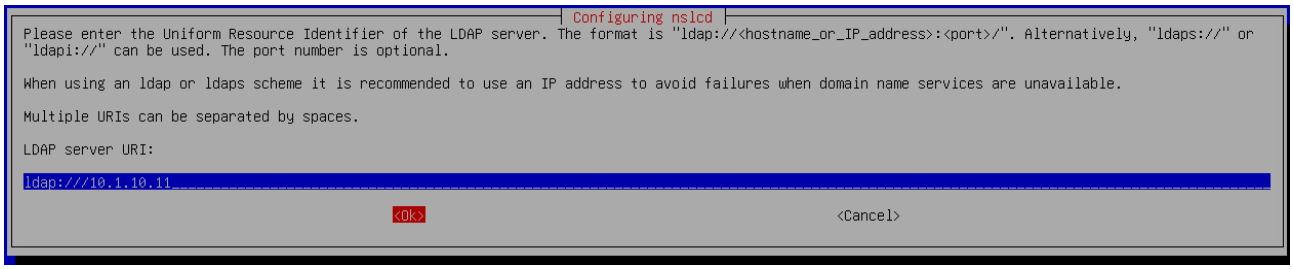
ldapmodify -Y EXTERNAL -H ldapi:/// -f mod_ssl.ldif

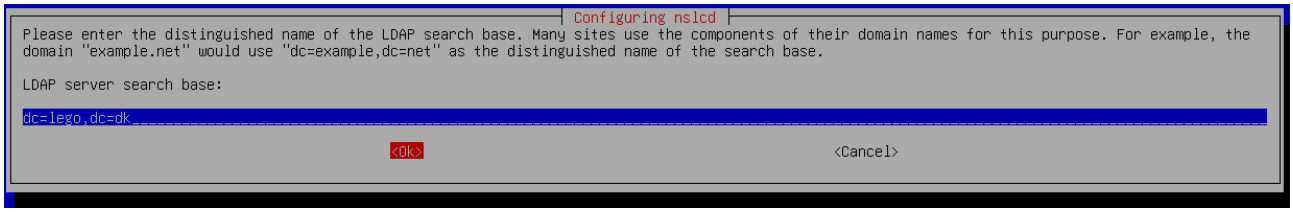
```
root@HQ-DC:~/ldap# ldapmodify -Y EXTERNAL -H ldapi:/// -f mod_ssl.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"

root@HQ-DC:~/ldap# _
```

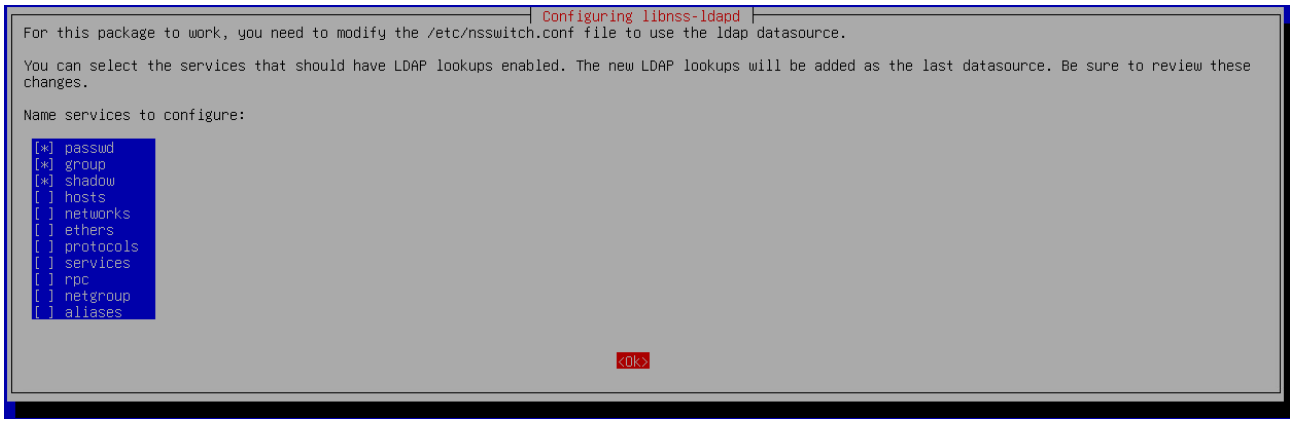
systemctl restart slapd

Step 6: Add a client

apt -y install libnss-ldapd libpam-ldapd ldap-utils



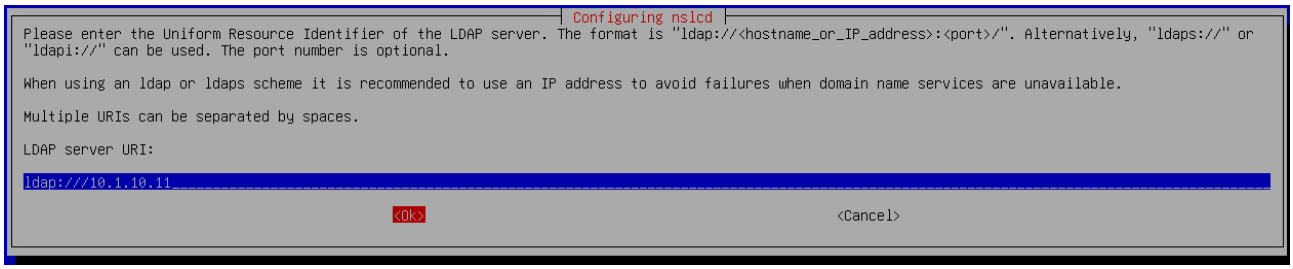


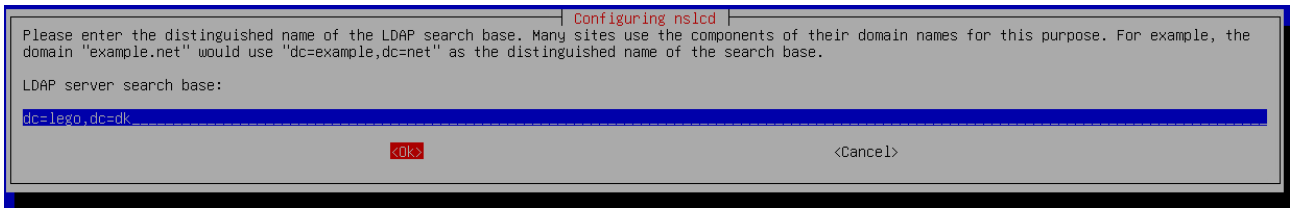


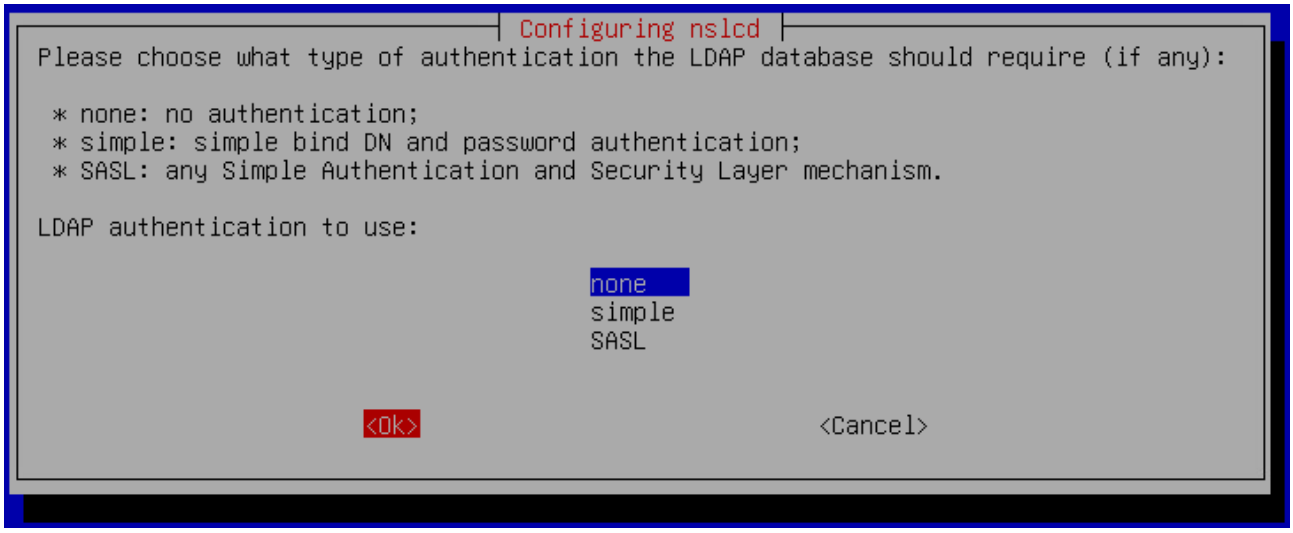
nano /etc/pam.d/common-session

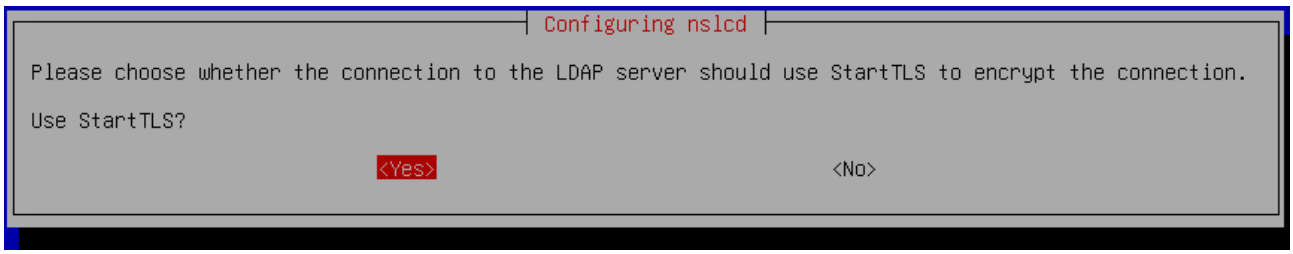
add to the end if need (create home directory automatically at initial login)
session optional pam_mkhomedir.so skel=/etc/skel umask=077

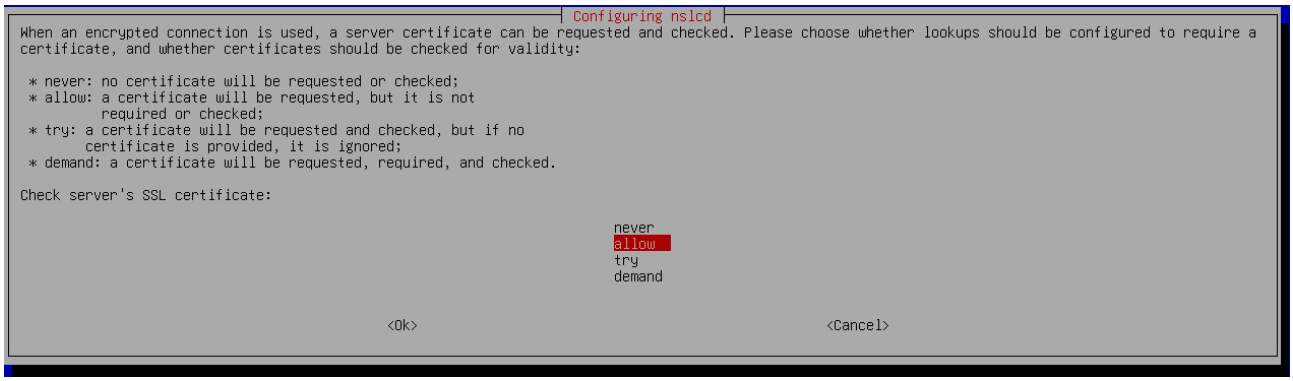
dpkg-reconfigure nslcd



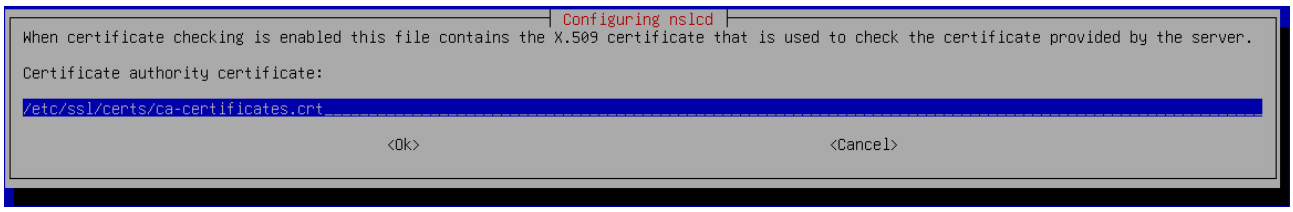








We can leave /etc/ssl/certs/ca-certificates.crt if we have added CA.crt and SUBCA.crt to system.



nano /etc/nslcd.conf

should be something like this:

```
# /etc/nslcd.conf
# nslcd configuration file. See nslcd.conf(5)
# for details.

# The user and group nslcd should run as.
uid nslcd
gid nslcd

# The location at which the LDAP server(s) should be reachable.
uri ldap:///10.1.10.11

# The search base that will be used for all queries.
base dc=lego,dc=dk

# The LDAP protocol version to use.
#ldap_version 3

# The DN to bind with for normal lookups.
#binddn cn=annonymous,dc=example,dc=net
#bindpw secret

# The DN used for password modifications by root.
#rootpwmoddn cn=admin,dc=example,dc=com

# SSL options
ssl start_tls
tls_reqcert allow
tls_cacertfile /etc/ssl/certs/ca-certificates.crt

# The search scope.
#scope sub
```

systemctl restart nscd nslcd

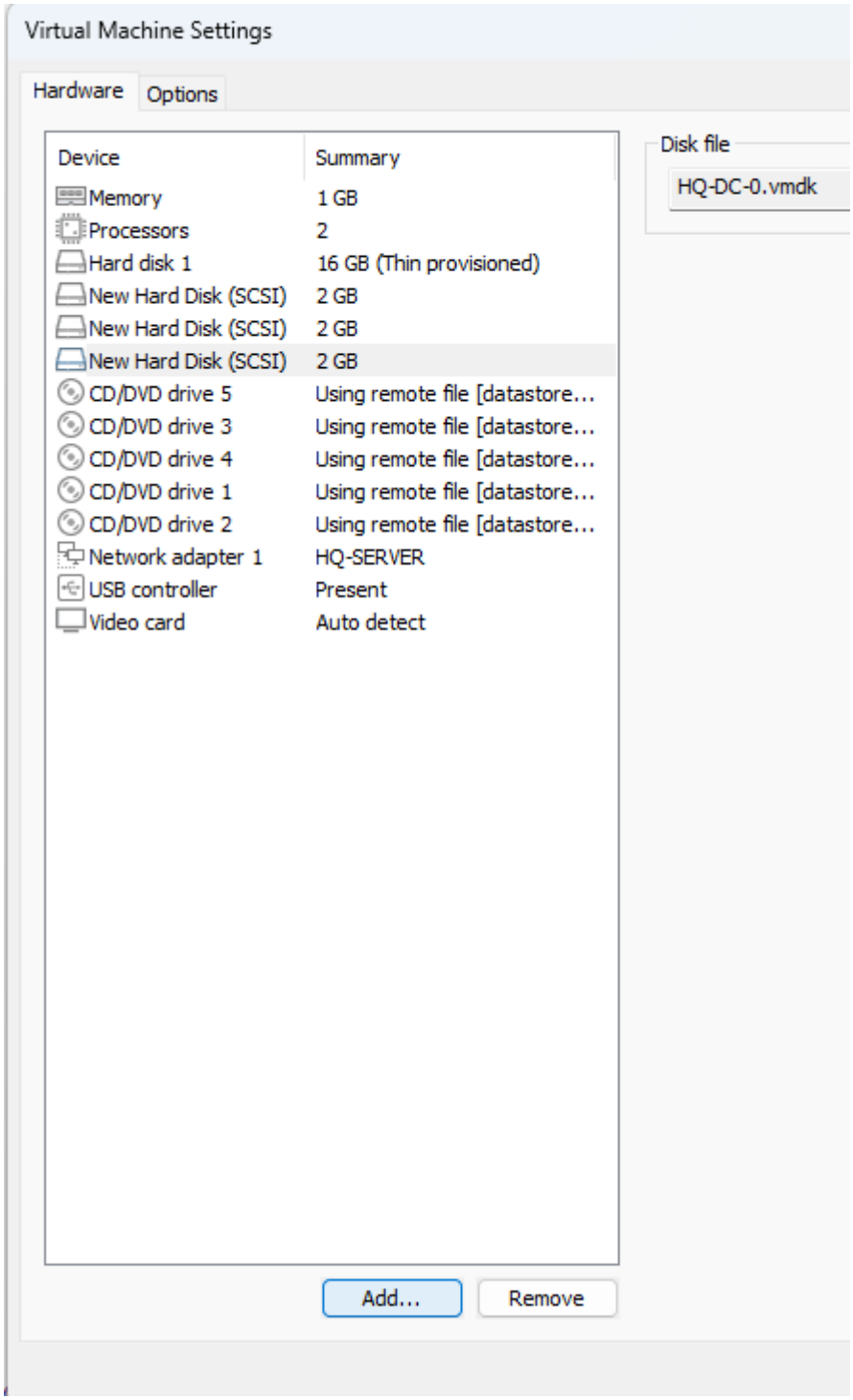
Check:

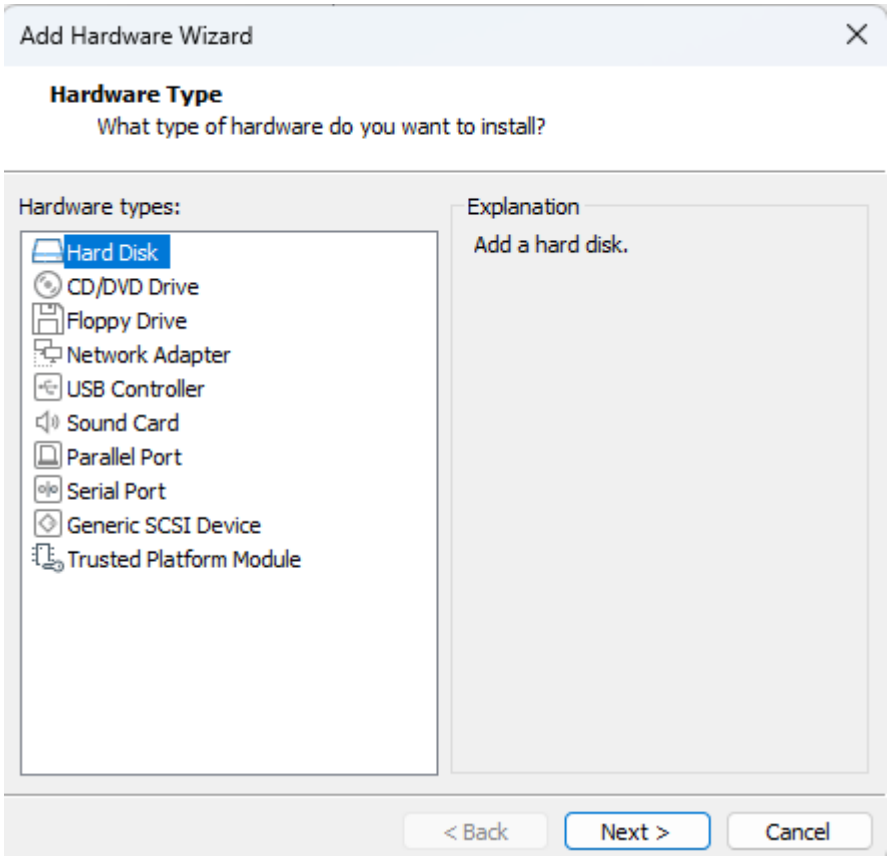
getent passwd

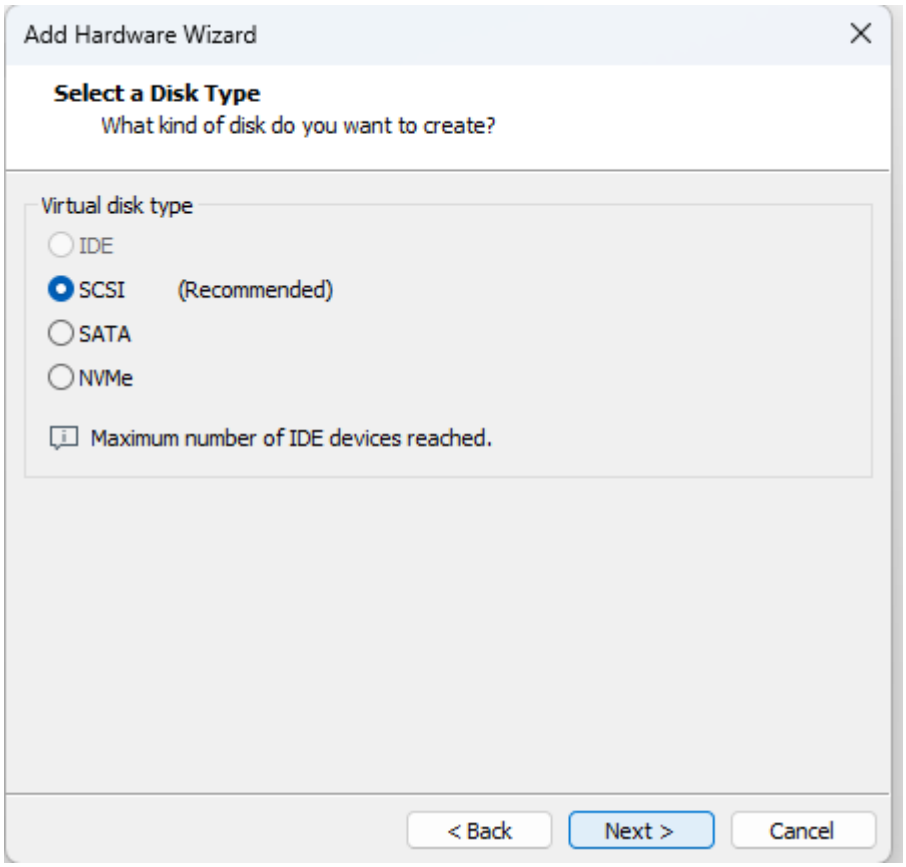
```
root@HQ-DC:~/ldap# getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
_rpc:x:102:65534::/run/rpcbind:/usr/sbin/nologin
tcpdump:x:103:109::/nonexistent:/usr/sbin/nologin
statd:x:104:65534::/var/lib/nfs:/usr/sbin/nologin
ntpsec:x:105:110::/nonexistent:/usr/sbin/nologin
openldap:x:106:111:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false
nslcd:x:107:112:nslcd name service LDAP connection daemon,,,:/run/nslcd:/usr/sbin/nologin
admin:x:4001:5001:admin:/home/admin:/bin/bash
frida:x:4002:5002:frida:/home/frida:/bin/bash
ella:x:4003:5002:ella:/home/ella:/bin/bash
carl:x:4004:5003:carl:/home/carl:/bin/bash
root@HQ-DC:~/ldap#
```

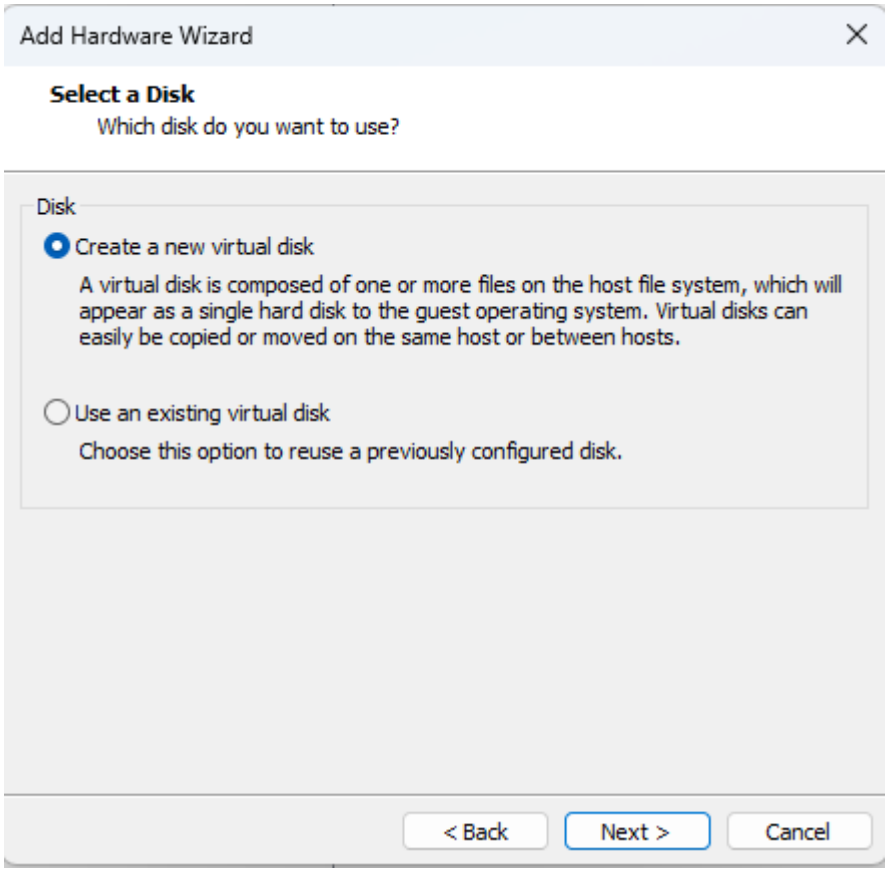
HQ-DC task3 - ZFS disk

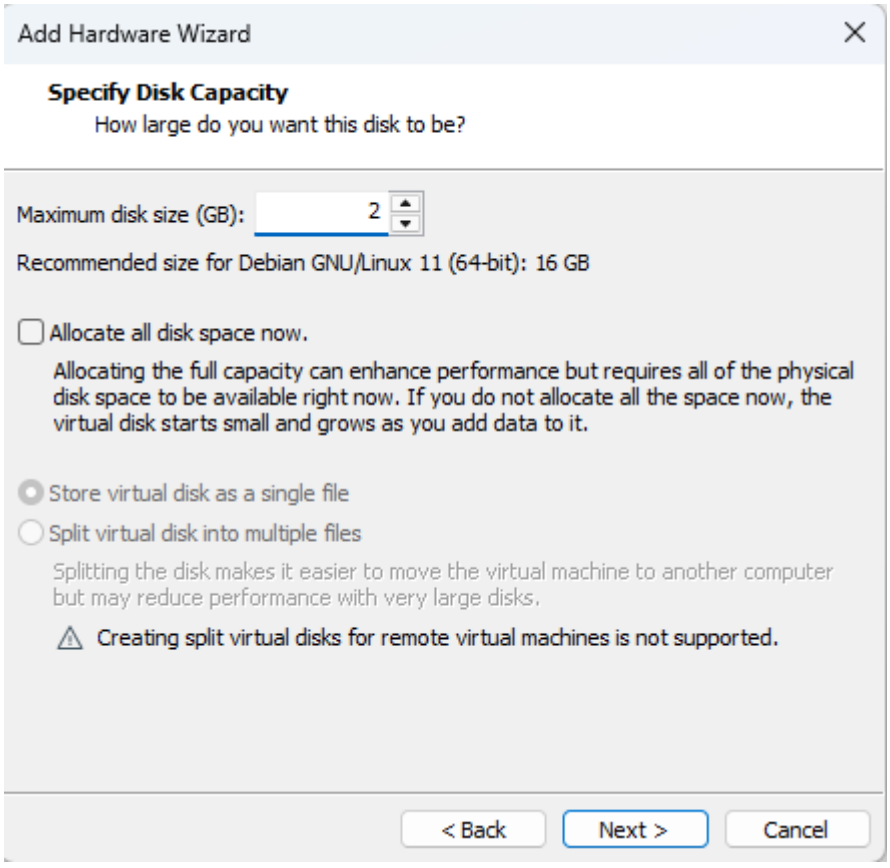
3. Add 4 new 2GB HDD to a machine. Configure software-based encrypted ZFS array and mount this to /share path.

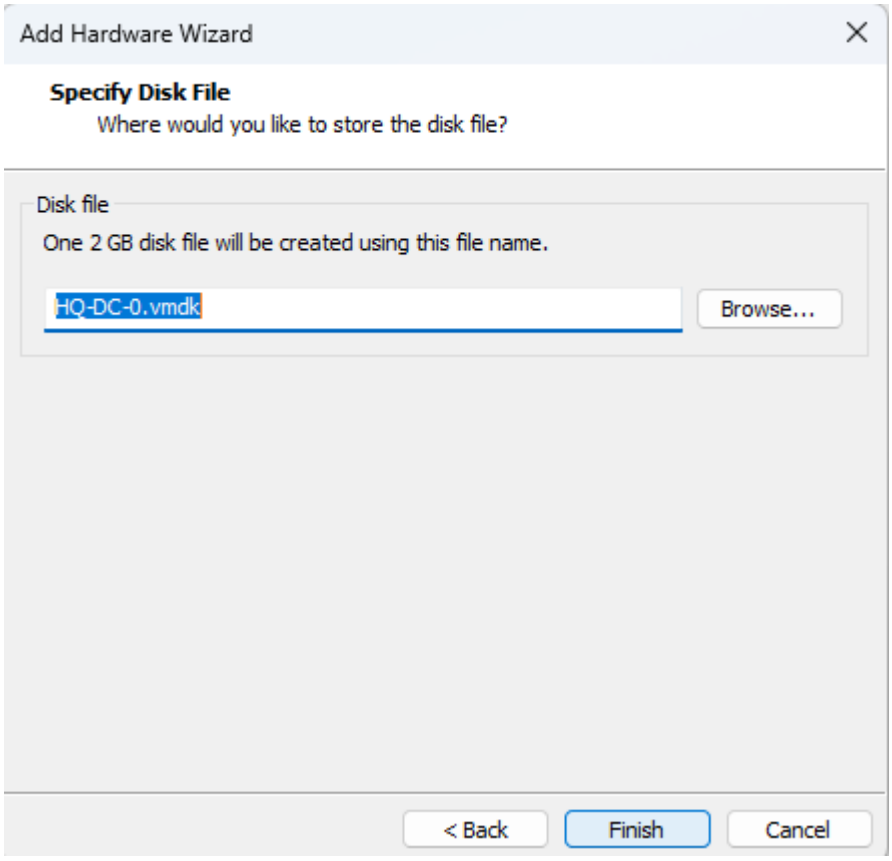

Virtual Machine Settings
Hardware
Options
Device
Summary
Memory
1 GB
Processors
2
Hard disk 1
16 GB (Thin provisioned)
New Hard Disk (SCSI)
2 GB
New Hard Disk (SCSI)
2 GB
New Hard Disk (SCSI)
2 GB
CD/DVD drive 5
Using remote file [datastore...
CD/DVD drive 3
Using remote file [datastore...
CD/DVD drive 4
Using remote file [datastore...
CD/DVD drive 1
Using remote file [datastore...
CD/DVD drive 2
Using remote file [datastore...
Network adapter 1
HQ-SERVER
USB controller
Present
Video card
Auto detect
Disk file
HQ-DC-0.vmdk
Add...
Remove


Add Hardware Wizard
Hardware Type
What type of hardware do you want to install?
Hardware types:
Hard Disk
CD/DVD Drive
Floppy Drive
Network Adapter
USB Controller
Sound Card
Parallel Port
Serial Port
Generic SCSI Device
Trusted Platform Module
Explanation
Add a hard disk.
< Back
Next >
Cancel


Add Hardware Wizard
Select a Disk Type
What kind of disk do you want to create?
Virtual disk type
IDE
SCSI (Recommended)
SATA
NVMe
Maximum number of IDE devices reached.
< Back
Next >
Cancel


Add Hardware Wizard
Select a Disk
Which disk do you want to use?
Disk
Create a new virtual disk
A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.
Use an existing virtual disk
Choose this option to reuse a previously configured disk.
< Back
Next >
Cancel


Add Hardware Wizard
Specify Disk Capacity
How large do you want this disk to be?
Maximum disk size (GB):
2
Recommended size for Debian GNU/Linux 11 (64-bit): 16 GB
Allocate all disk space now.
Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.
Store virtual disk as a single file
Split virtual disk into multiple files
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.
Creating split virtual disks for remote virtual machines is not supported.
< Back
Next >
Cancel



Power on the machine.

Check the HDDs:

```
fdisk -l
```

```
root@HQ-DC:~# fdisk -l
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdd: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdc: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x21c5dc1b

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 31553535 31551488   15G 83 Linux
/dev/sda2       31555582 33552383  1996802  975M  5 Extended
/dev/sda5       31555584 33552383  1996800  975M 82 Linux swap / Solaris


Disk /dev/sde: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@HQ-DC:~#
```

Or with

lsblk -o NAME,SIZE,TYPE,MOUNTPOINT

```
root@HQ-DC:~# lsblk -O NAME,SIZE,TYPE,MOUNTPOINT
lsblk: NAME,SIZE,TYPE,MOUNTPOINT: not a block device
root@HQ-DC:~# lsblk -o NAME,SIZE,TYPE,MOUNTPOINT
NAME    SIZE TYPE MOUNTPOINT
sda      16G disk
├─sda1   15G part /
├─sda2    1K part
└─sda5  975M part [SWAP]
sdb       2G disk
sdc       2G disk
sdd       2G disk
sde       2G disk
sr0    22.2G rom
sr1    22.3G rom
sr2    21.5G rom
sr3    22.3G rom
sr4     1.6G rom
root@HQ-DC:~#
```

In this example, HDDs are:
/dev/sdb
/dev/sdc
/dev/sdd
/dev/sde

Create partition for each disk:

`fdisk /dev/sdb`
type `n`
press enter until you return to the main prompt
type `w`

```
zpool create RAID mirror /dev/sdb1 /dev/sdc1 mirror /dev/sdd1 /dev/sde1

zfs create RAID/smb -o mountpoint=/share -o encryption=on -o
keyformat=passphrase -o keylocation=prompt
```

HQ-DC task4 - CIFS directory share

4. Configure CIFS service for the profile directory of the corporate users. Use the /share/users/<username> as the path of the profile directories. Make sure the users can only access their own shared home folder.

HQ-DC task5 - Share automatic mount

5. Enable automatic mount of network shares on clients.

HQ-SAM-1

HQ-SAM-1 task1 - HA DHCP

1. Create a HA DHCP cluster with HQ-SAM-2 for the HQ-CLIENT network.

```
apt install isc-dhcp-server
```

```
nano /etc/dhcp/dhcpd.conf
```

```
authoritative;
failover peer "dhcp-failover" {
        primary;
        address 10.1.10.21;
        port 647;
        peer address 10.1.10.22;
        peer port 647;
        max-response-delay 30;
        max-unacked-updates 10;
        load balance max seconds 3;
        mclt 1800;
        split 128;
}

subnet 10.1.10.0 netmask 255.255.255.0 {
}
subnet 10.1.30.0 netmask 255.255.255.0 {
        #interface ens192;
        option domain-name-servers 10.1.20.11;
        #option domain-name "billund.lego.dk";
        option routers 10.1.30.1;
        #option broadcast-address 10.1.30.255;
        option ntp-servers 10.1.30.1;
        #option time-offset -10800;
        default-lease-time 86400;
        pool{
            failover peer "dhcp-failover";
            range 10.1.30.21 10.1.30.120;
            max-lease-time 172800;
        }
}
```

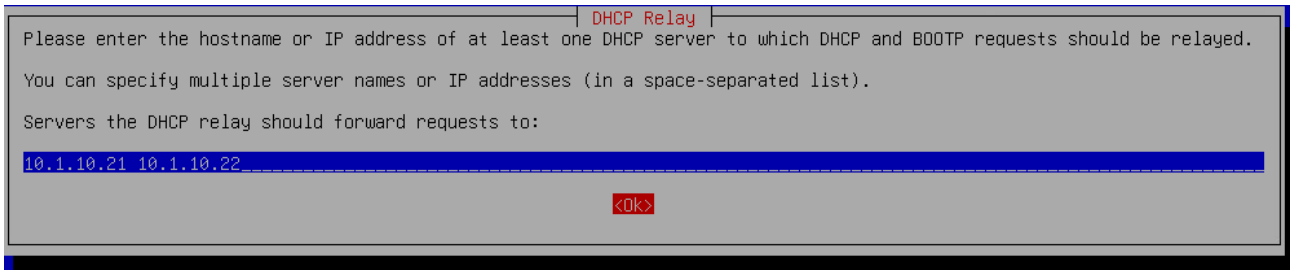
```
nano /etc/default/isc-dhcp-server
```

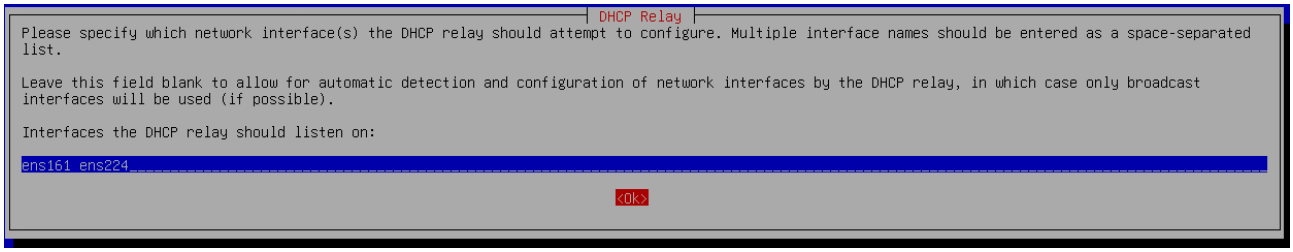
```
INTERFACES=”ens192”
```

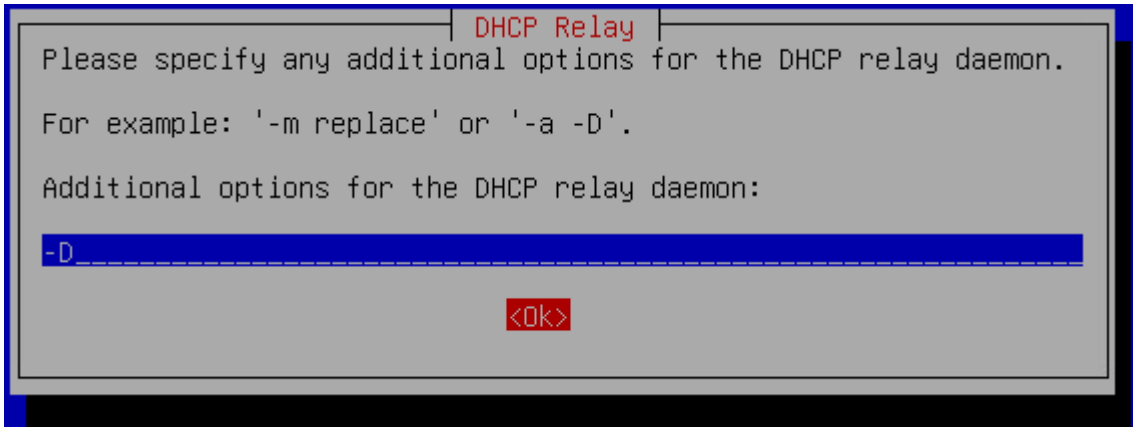
```
systemctl restart isc-dhcp-server
```

GO TO R-HQ

apt install isc-dhcp-relay







via config file

nano /etc/default/isc-dhcp-relay

SERVERS=”10.1.10.21 10.1.10.22”
INTERFACES=”ens161 ens224”
OPTIONS=”-D”

systemctl restart isc-dhcp-relay

HQ-SAM-1 task2 - Intermediate CA

2. Create a subordinate certificate authority. Use the next parameters:

C=DK, O=Lego APS, CN=Lego APS Intermediate CA

Place SUBCA.crt CA certificate file in /ca folder.

The Sub CA should issue all the necessary certificates, which should contain (only and exactly) the following fields:

C=DK, O=Lego APS, CN=<FQDN>

Make sure all servers and the client applications accept the certs issued by this Sub CA.

Step 1: Create /ca directory

```
mkdir -p /ca && cd /ca
```

Go to HQ-DC, create all the necessary files and come back here

When you come back, in /ca directory should be:

```
root@HQ-SAM-1:~# cd /ca/
root@HQ-SAM-1:/ca# ls -la
total 24
drwxr-xr-x  2 root root 4096 Jun 11 21:02 .
drwxr-xr-x 19 root root 4096 Jun 11 21:01 ..
-rw-r--r--  1 root root 1911 Jun 11 21:02 CA.crt
-rw-r--r--  1 root root 1801 Jun 11 21:02 SUBCA.crt
-rw-r--r--  1 root root 1647 Jun 11 21:02 SUBCA.csr
-rw-------  1 root root 3272 Jun 11 21:02 SUBCA.key
root@HQ-SAM-1:/ca# _
```

Step 2: Add the Root CA and the Intermediate CA certificate to the trusted root certificates

```
sudo apt install -y ca-certificates
sudo cp *.crt /usr/local/share/ca-certificates
sudo update-ca-certificates
```

For server certificates

Step 1: Create the server key

```
####openssl genrsa -out xxx.key 2048
openssl genrsa -out xxx.key
```

Step 2: Create the CSR (Certificate Signing Request)

```
openssl req -new -key xxx.key -subj "/C=DK/O=Lego APS/CN=xxx.xxx.dk" -out xxx.csr
```

Step 3: Sign the CSR with the Intermediate cert and generate the server crt

```
####openssl x509 -req -in xxx.csr -CA SUBCA.crt -CAkey SUBCA.key -CAcreateserial -out xxx.crt -days 825
openssl x509 -req -in xxx.csr -CA SUBCA.crt -CAkey SUBCA.key -CAcreateserial -out xxx.crt
```

Note: for some situations, maybe we must set the cert as “server” cert or a “client” cert, using one of these parameters in the command:

openssl x509 -req -in xxx.csr -CA SUBCA.crt -CAkey SUBCA.key -CAcreateserial -out xxx.crt -days 825 -extfile <(echo “extendedKeyUsage=serverAuth”)

openssl x509 -req -in xxx.csr -CA SUBCA.crt -CAkey SUBCA.key -CAcreateserial -out xxx.crt -days 825 -extfile <(echo “extendedKeyUsage=clientAuth”)

certificates to create

- www.billund.lego.dk
- www.herning.lego.dk
- mail.billund.lego.dk
- LDAP
- R-BR (for Site-to-site VPN) clientAuth
- R-HQ (for Site-to-site VPN and client-to-site VPN) serverAuth
- HOME (for client-to-site VPN) clientAuth

Last step: Move certificates to the correspondent machines using scp

HQ-SAM-2

HQ-SAM-2 task1 - HA DHCP

1. Create a HA DHCP cluster with HQ-SAM-1 for the HQ-CLIENT network.

apt install isc-dhcp-server

nano /etc/dhcp/dhcpd.conf

```
failover peer "dhcp-failover" {
        secondary;
        address 10.1.10.22;
        port 647;
        peer address 10.1.10.21;
        peer port 647;
        max-response-delay 30;
        max-unacked-updates 10;
        load balance max seconds 3;
}

subnet 10.1.10.0 netmask 255.255.255.0 {
}
subnet 10.1.30.0 netmask 255.255.255.0 {
        #interface ens192;
        option domain-name-servers 10.1.20.11;
        #option domain-name "billund.lego.dk";
        option routers 10.1.30.1;
        #option broadcast-address 10.1.30.255;
        option ntp-servers 10.1.30.1;
```

```
        #option time-offset -10800;
        default-lease-time 86400;
        pool{
            failover peer "dhcp-failover";
            range 10.1.30.21 10.1.30.120;
            max-lease-time 172800;
        }
}
```

nano /etc/default/isc-dhcp-server

```
INTERFACES=”ens192”
```

systemctl restart isc-dhcp-server

HQ-SAM-2 task2 - scheduled backup task

2. Configure a scheduled backup of /share/users directory of HQ-DC to /backup/users directory on HQ-SAM-2 every 5 minutes.

HQ-DMZ-1

HQ-DMZ-1 task1 - CRL

1. Configure CRL (Certificate Revocation List Distribution point) available in http://crl.lego.dk.

HQ-DMZ-2 task1 - DNS server

2. Install and configure a DNS server for both forward and reverse lookups of billund.lego.dk domain. Configure SPF, DKIM and DMARC records for email domain. It must serve requests from the Internet as well but cannot use private addressing for that purpose. Configure this DNS server as a secondary DNS server for herning.lego.dk domain. DNS requests for other domains must be forwarded to the correct DNS server.

apt install bind9

nano /etc/bind/med.conf

```
#comment out default zones line

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
//include "/etc/bind/named.conf.default-zones";
```

nano /etc/bind/named.conf.local

```
acl internal {10.1.10.0/24; 10.1.20.0/24; 10.1.30.0/24; 10.2.10.0/24; 10.2.30.0/24; localhost; };
```

```
acl external {any; };

view int {
        match-clients { internal; };
        allow-recursion { any; };

        zone "billund.lego.dk" {
                type master;
                file "/etc/bind/db.billund.lego.dk.int";
                allow-transfer { 10.2.10.11; };
                notify yes;
        };

        zone "20.1.10.in-addr.arpa" {
                type master;
                file "/etc/bind/db.10.1.20";
                allow-transfer { 10.2.10.11; };
                notify yes;
        };

        zone "herning.lego.dk" {
                type slave;
                masters { 10.2.10.11; };
                file "db.herning.lego.dk.int";
        };

        zone “10.2.10.in-addr.arpa" {
                type slave;
                masters { 10.2.10.11; };
                file "db.10.2.10";
        };

        include "/etc/bind/zones.rfc1918";
        include "/etc/bind/named.conf.default-zones";
};

view ext{
        match-clients { external; };
        allow-recursion { any; };

        zone "billund.lego.dk" {
                type master;
                file "/etc/bind/db.billund.lego.dk.ext";
                allow-transfer { 203.0.113.10; };
                notify yes;
        };
        zone "113.0.203.in-addr.arpa" {
                type master;
                file "/etc/bind/db.203.0.113";
                allow-transfer { 203.0.113.10; };
                notify yes;
        };
```

```
        zone "herning.lego.dk" {
                type slave;
                masters { 203.0.113.10; };
                file "db.herning.lego.dk.ext";
        };

        include "/etc/bind/zones.rfc1918";
        include "/etc/bind/named.conf.default-zones";
};
```

nano /etc/bind/db.billund.lego.dk.int

```
$TTL    86400
@       IN      SOA     www.billund.lego.dk. root.billund.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ) ; Negative Cache TTL
;
@       IN      NS      www.billund.lego.dk.
@       IN      MX 10   mail.billund.lego.dk.
www     IN      A       10.1.20.11
mail    IN      A       10.1.20.12
```

nano /etc/bind/db10.1.20

```
$TTL    86400
@       IN      SOA     www.billund.lego.dk. root.billund.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ) ; Negative Cache TTL
;
@       IN      NS      www.billund.lego.dk.
11      IN      PTR     www.billund.lego.dk.
12      IN      PTR     mail.billund.lego.dk.
```

nano /etc/bind/db.billund.lego.dk.ext

```
$TTL    86400
@       IN      SOA     www.billund.lego.dk. root.billund.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
```

```
                        2419200         ; Expire
                         86400 ); Negative Cache TTL
;
@       IN      NS      www.billund.lego.dk.
@       IN      MX 10   mail.billund.lego.dk.
www     IN      A       203.0.113.2
mail    IN      A       203.0.113.2
```

nano /etc/bind/db.203.0.113

```
$TTL    86400
@       IN      SOA     www.billund.lego.dk. root.billund.lego.dk. (
                            1           ; Serial
                        604800          ; Refresh
                         86400          ; Retry
                        2419200         ; Expire
                         86400 ); Negative Cache TTL
;
@       IN      NS      www.billund.lego.dk.
2       IN      PTR     www.billund.lego.dk.
2       IN      PTR     mail.billund.lego.dk.
10      IN      PTR     www.herning.lego.dk.
```

REGISTROS SPF, DKIM, DMARC y SRV para SMTP e IMAP:

```
_smtps._tcp     IN      SRV 0 0 465 hq-dmz-1
_imaps._tcp     IN      SRV 0 0 993 hq-dmz-1
_dmarc          IN      TXT "v=DMARC1; p=none"
@               IN      TXT "v=spf1 +a +mx +a:hq-dmz-1.billund.lego.dk -all"
default._domainkey      IN      TXT "v=DKIM1; p=CLAVEPUBLICA"
~
~
```

Queda pendiente desarrollar cómo obtener la clave DKIM.

systemctl restart bind9

Secondary transferred zone files will appear in /var/cache/bind/.

```
root@HQ-DMZ-1:~# ls -la /var/cache/bind/
total 40
drwxrwxr-x  2 root bind 4096 Jun 15 10:48 .
drwxr-xr-x 12 root root 4096 Jun 10 17:02 ..
-rw-r--r--  1 bind bind  264 Jun 15 10:42 db.10.2.10
-rw-r--r--  1 bind bind  332 Jun 15 10:42 db.herning.lego.dk.int
-rw-r--r--  1 bind bind  221 Jun 11 10:30 ext.mkeys
-rw-r--r--  1 bind bind 2913 Jun 11 10:30 ext.mkeys.jnl
-rw-r--r--  1 bind bind  221 Jun 11 10:30 int.mkeys
-rw-r--r--  1 bind bind 2913 Jun 11 10:30 int.mkeys.jnl
-rw-r--r--  1 bind bind  297 Jun 10 12:19 managed-keys.bind
-rw-r--r--  1 bind bind  785 Jun 10 12:19 managed-keys.bind.jnl
root@HQ-DMZ-1:~#
```

If we want to update a zone the SERIAL NUMBER MUST BE CHANGED.
To force updates we can use the following command:

```
rndc reload
```

HQ-DMZ-1 task3 - HA web server

3. Install and configure a HA web service with HQ-DMZ-2. Serve the https://www.billund.lego.dk site using a valid certificate signed by the company's Sub CA. HTTP requests must be redirected to HTTPS. Website should show the text "Welcome to Billund". For HA you must use HAproxy.

Assure that the following certs and keys are in HQ-DMZ-1 and HQ-DMZ-2:

- www.billund.lego.dk.key
- www.billund.lego.dk.crt
- CA.crt
- SUBCA.crt
- CHAIN-CA.crt

Install apache2

```
apt install apache2
```

Edit the message on the main webpage:

```
echo “Welcome to Billund. Site served by HQ-DMZ-1” > /var/www/html/index.html
```

Copy cert and key to /etc/ssl/certs and /etc/ssl/private directories:

```
cp www.billund.lego.dk.crt  /etc/ssl/certs
cp www.billund.lego.dk.key  /etc/ssl/private
```

Trust CA.crt and SUBCA.crt:

```
cp *CA*crt /usr/local/share/ca-certificates/
update-ca-certificates
```

Enable SSL module:

```
a2enmod ssl
```

Enable HTTPS site:

```
a2ensite default-ssl.conf
```

Configure certificate paths:

```
nano /etc/apache2/sites-enabled/default-ssl.conf
```

<VirtualHost *:80>

```
        DocumenRoot /var/www/html
        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/www.billund.lego.dk.crt
        SSLCertificateKeyFile       /etc/ssl/private/www.billund.lego.dk.key
</VirtualHost>
```

```
systemctl restart apache2
```

Install HAproxy

```
apt install haproxy
```

```
nano /etc/haproxy/haproxy.cfg
```

Add to the end

```
frontend apache_front
    bind *:80
    default_backend    apache_backend_servers
    option          forwardfor
backend apache_backend_servers
    balance         roundrobin
    server          HQ-DMZ-1 10.1.20.11:80 check
    server          HQ-DMZ-2 10.1.20.12:80 check
```

HQ-DMZ-2

HQ-DMZ-2 task1 - HA web server

1. Install and configure a HA web service with HQ-DMZ-1. See details there.

HQ-DMZ-2 task2 - Email server

2. Configure e-mail service accessible on mail.billund.lego.dk to send and receive email for the lego.dk domain. LDAP users access their mailboxes using SSL/TLS-secured IMAP (port 143) and send emails using STARTTLS-secured SMTP (port 587).

Assure that the following certs and keys are in HQ-DMZ-2:

- mail.billund.lego.dk.key
- mail.billund.lego.dk.crt
- CA.crt
- SUBCA.crt
- CHAIN-CA.crt

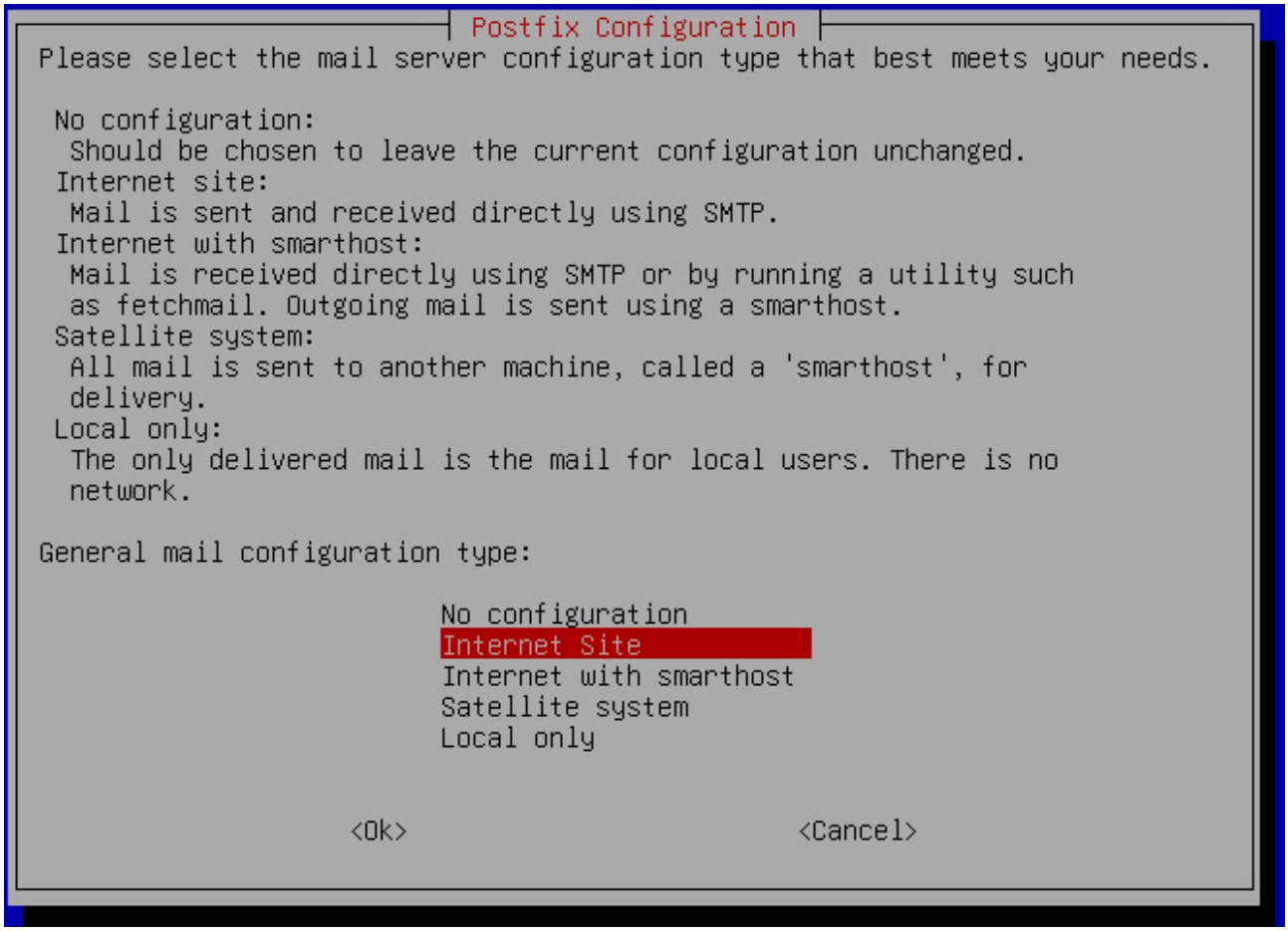
Copy cert and key to /etc/ssl/certs and /etc/ssl/private directories:

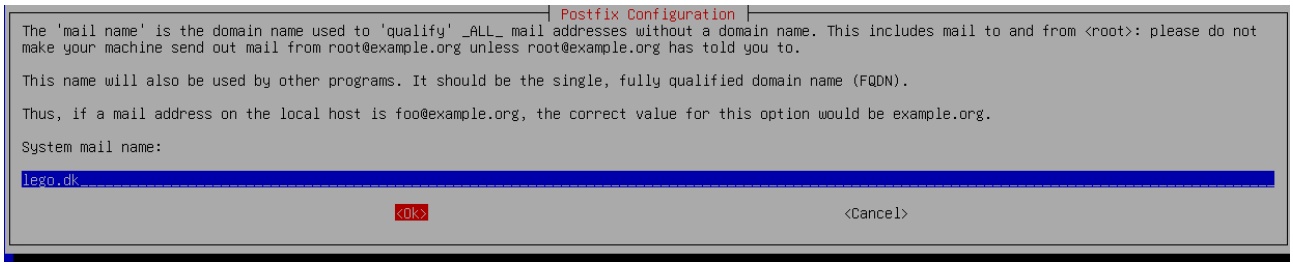
```
cp mail.billund.lego.dk.crt  /etc/ssl/certs
cp mail.billund.lego.dk.key  /etc/ssl/private
```

Install postfix

```
apt install postfix
```

Postfix configuration while installing:





nano /etc/postfix/master.cf

Comment out and modify these lines:

```
#/etc/postfix/master.cf
submission inet n - y - - smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_tls_wrappermode=no
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
  -o smtpd_sasl_type=dovecot
  -o smtpd_sasl_path=private/auth
```

nano /etc/postfix/main.cf

Comment out and modify these lines:

#/etc/postfix/main.cf
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 0.0.0.0/0
#Enable TLS Encryption when Postfix receives incoming emails
smtpd_tls_cert_file=/etc/ssl/certs/mail.billund.lego.dk.crt
smtpd_tls_key_file=/etc/ssl/private/mail.billund.lego.dk.key
smtpd_tls_security_level=may
smtpd_tls_loglevel = 1
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache

#Enable TLS Encryption when Postfix sends outgoing emails
smtp_tls_security_level = may
smtp_tls_loglevel = 1
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

#Enforce TLSv1.3 or TLSv1.2
smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtpd_tls_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtp_tls_mandatory_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtp_tls_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1

**Nota: los parámetros de /etc/postfix/main.cf se pueden configurar con el comando “postconf -e…”, y te permite usar el tabulador para ver las opciones, por ejemplo:
postconf -e ‘smtpd_tls_cert_file = /etc/ssl/certs/mail.billund.lego.dk.crt’

Más ejemplos en https://www.linuxparty.es/10-comunicaciones/9387-instalar-y-configurar-postfix-y-dovecot-con-autenticacion-y-tls

systemctl restart postfix

Check that port 587 (postfix-SMTP) is listening:

ss -lnpt | grep master

```
root@HQ-DMZ-2:~# ss -lntp | grep master
LISTEN 0      100          0.0.0.0:587       0.0.0.0:*    users:(("master",pid=14529,fd=18))
LISTEN 0      100          0.0.0.0:25        0.0.0.0:*    users:(("master",pid=14529,fd=13))
LISTEN 0      100             [::]:587          [::]:*    users:(("master",pid=14529,fd=19))
LISTEN 0      100             [::]:25           [::]:*    users:(("master",pid=14529,fd=14))
root@HQ-DMZ-2:~# _
```

According to 30% changes, IMAP will be one of these options:

- ...using SSL/TLS-secured IMAP (port 993)...
- ...using STARTTLS-secured IMAP (port 143)...

Option A: ...using SSL/TLS-secured IMAP (port 993)...

HAPROXY POR DELANTE CON SSL

Option B: ...using STARTTLS-secured IMAP (port 143)...

Source: https://www.linuxbabe.com/mail-server/install-dovecot-imap-server-debian

Install and dovecot-imapd

apt install dovecot-imapd dovecot-lmtp

nano /etc/dovecot/dovecot.conf

protocols = imap lmtp

nano /etc/dovecot/conf.d/10-mail.conf

mail_location = maildir:~/Maildir
#mail_location = mbox:~/mail:INBOX=/var/mail/%u

nano /etc/dovecot/conf.d/10-master.conf

service lmtp {

```
 unix_listener /var/spool/postfix/private/dovecot-lmtp {
   mode = 0600
   user = postfix
   group = postfix
  }
}
```

nano /etc/postfix/main.cf

```
mailbox_transport = lmtp:unix:private/dovecot-lmtp
smtputf8_enable = no
```

nano /etc/dovecot/conf.d/10-auth.conf

```
disable_plaintext_auth = yes
#auth_username_format = %Lu
auth_username_format = %n
auth_mechanisms = plain
```

nano /etc/dovecot/conf.d/10-ssl.conf

```
ssl = required
ssl_cert = </etc/ssl/certs/mail.billund.lego.dk.crt
ssl_key = </etc/ssl/private/mail.billund.lego.dk.key
#ssl_prefer_server_ciphers = no
ssl_prefer_server_ciphers = yes
ssl_min_protocol = TLSv1.2
```

nano /etc/dovecot/conf.d/10-master.conf

```
service auth {
    unix_listener /var/spool/postfix/private/auth {
      mode = 0660
      user = postfix
      group = postfix
    }
}
```

systemctl restart postfix dovecot

sudo ss -lnpt | grep dovecot

HQ-DMZ-2 task3 - Zabbix monitoring (web)

3. Install and configure Zabbix for monitoring the availability of https://www.billund.lego.dk webserver. When the server is down, write an alert to /monitor/webalert.log and send an email to Frida.

apt install sudo
apt install postgresql postgresql-contrib

sudo systemctl is-enabled postgresql
sudo systemctl status postgresql

```
root@HQ-DMZ-2:~# systemctl is-enabled postgresql
enabled
root@HQ-DMZ-2:~# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Mon 2025-06-16 18:17:31 CEST; 1min 34s ago
   Main PID: 17616 (code=exited, status=0/SUCCESS)
        CPU: 4ms

Jun 16 18:17:31 HQ-DMZ-2 systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jun 16 18:17:31 HQ-DMZ-2 systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
root@HQ-DMZ-2:~# _
```

create a new PostgreSQL user 'zabbix'

sudo -u postgres createuser --pwprompt zabbix

create a new database 'zabbix' with the default owner 'zabbix'

sudo -u postgres createdb -O zabbix zabbix

```
root@HQ-DMZ-2:~# sudo -u postgres createuser --pwprompt zabbix
could not change directory to "/root": Permission denied
Enter password for new role:
Enter it again:
root@HQ-DMZ-2:~# sudo -u postgres createdb -O zabbix zabbix
could not change directory to "/root": Permission denied
root@HQ-DMZ-2:~#
```

Install from ISO image:

dpkg -i zabbix-release_latest_7.4+debian12_all.deb

apt install zabbix-server-pgsql zabbix-frontend-php php8.2-pgsql zabbix-agent

apt install nginx

dpkg -i zabbix-sql-scripts_7.0.13-1+debian12_all.deb

dpkg -i zabbix-nginx-conf_7.0.13-1+debian12_all.deb

Import the database schema for Zabbix to the 'zabbix' database with the user 'zabbix'. Enter your 'zabbix' password when prompted.

zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | sudo -u zabbix psql zabbix

open Zabbix configuration '/etc/zabbix/zabbix_server.conf' and Uncomment the default 'DBHost', 'DBName', 'DBUser', and 'DBPassword' with your PostgreSQL credentials.

nano /etc/zabbix/zabbix_server.conf

DBHost = localhost
DBName = zabbix
DBUser = zabbix
DBPassword = Passw0rd!

nano /etc/zabbix/nginx.conf

listen 8080;
server_name zabbix.billund.lego.dk;

systemctl enable zabbix-server zabbix-agent nginx php8.2-fpm

systemctl restart zabbix-server zabbix-agent nginx php8.2-fpm

systemctl status zabbix-server zabbix-agent nginx php8.2-fpm

ERROR

```
root@HQ-DMZ-2:~# systemctl status zabbix-server.service
× zabbix-server.service - Zabbix Server (PostgreSQL)
     Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; preset: enabled)
     Active: failed (Result: exit-code) since Mon 2025-06-16 19:23:11 CEST; 17min ago
   Duration: 277ms
       Docs: man:zabbix_server
    Process: 4258 ExecStart=/usr/sbin/zabbix_server --foreground (code=exited, status=1/FAILURE)
   Main PID: 4258 (code=exited, status=1/FAILURE)
        CPU: 138ms

Jun 16 19:23:11 HQ-DMZ-2 systemd[1]: Started zabbix-server.service - Zabbix Server (PostgreSQL).
Jun 16 19:23:11 HQ-DMZ-2 zabbix_server[4258]: Starting Zabbix Server. Zabbix 6.0.14 (revision 3f184b456c7).
Jun 16 19:23:11 HQ-DMZ-2 zabbix_server[4258]: Press Ctrl+C to exit.
Jun 16 19:23:11 HQ-DMZ-2 systemd[1]: zabbix-server.service: Main process exited, code=exited, status=1/FAILURE
Jun 16 19:23:11 HQ-DMZ-2 systemd[1]: zabbix-server.service: Failed with result 'exit-code'.
root@HQ-DMZ-2:~# _
```

```
on: 06000000.
   4258:20250616:192311.460 Starting Zabbix Server. Zabbix 6.0.14 (revision 3f184b456c7).
   4258:20250616:192311.461 ****** Enabled features ******
   4258:20250616:192311.461 SNMP monitoring:           YES
   4258:20250616:192311.461 IPMI monitoring:           YES
   4258:20250616:192311.461 Web monitoring:            YES
   4258:20250616:192311.461 VMware monitoring:         YES
   4258:20250616:192311.461 SMTP authentication:       YES
   4258:20250616:192311.461 ODBC:                      YES
   4258:20250616:192311.461 SSH support:               YES
   4258:20250616:192311.461 IPv6 support:              YES
   4258:20250616:192311.461 TLS support:               YES
   4258:20250616:192311.461 ******************************
   4258:20250616:192311.461 using configuration file: /etc/zabbix/zabbix_server.conf
   4258:20250616:192311.642 The server does not match Zabbix database. Current database version (mandatory/optional): 07000000/07000019. Required mandatory versi
on: 06000000.
root@HQ-DMZ-2:~# _
```

```
root@HQ-DMZ-2:~# apt list | grep zabbix

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

pcp-export-pcp2zabbix/unknown 6.0.3-1.1 amd64
pcp-export-zabbix-agent/unknown 6.0.3-1.1 amd64
python3-pyzabbix/unknown 0.8.2-1 all
zabbix-agent2/unknown 1:6.0.14+dfsg-1+b1 amd64
zabbix-agent/unknown,now 1:6.0.14+dfsg-1+b1 amd64 [installed]
zabbix-frontend-php/unknown,now 1:6.0.14+dfsg-1 all [installed]
zabbix-java-gateway/unknown 1:6.0.14+dfsg-1 all
zabbix-nginx-conf/now 1:7.0.13-1+debian12 all [installed,local]
zabbix-proxy-mysql/unknown 1:6.0.14+dfsg-1+b1 amd64
zabbix-proxy-pgsql/unknown 1:6.0.14+dfsg-1+b1 amd64
zabbix-proxy-sqlite3/unknown 1:6.0.14+dfsg-1+b1 amd64
zabbix-release/now 1:7.4-0.2+debian12 all [installed,local]
zabbix-server-mysql/unknown 1:6.0.14+dfsg-1+b1 amd64
zabbix-server-pgsql/unknown,now 1:6.0.14+dfsg-1+b1 amd64 [installed]
zabbix-sql-scripts/now 1:7.0.13-1+debian12 all [installed,local]
zabbix-web-service/unknown 1:6.0.14+dfsg-1+b1 amd64
root@HQ-DMZ-2:~#
```

Maybe we should use zabbix-release_latest_**6.0**+debian12_all.deb version and all the correspondent 6.o versions. The unique 7.x versions have been downloaded and installed manually

- zabbix-release_latest_7.4+debian12_all.deb
- zabbix-sql-scripts_7.0.13-1+debian12_all.deb
- zabbix-nginx-conf_7.0.13-1+debian12_all.deb

HQ-DMZ-2 task4 - Zabbix monitoring (CPU)

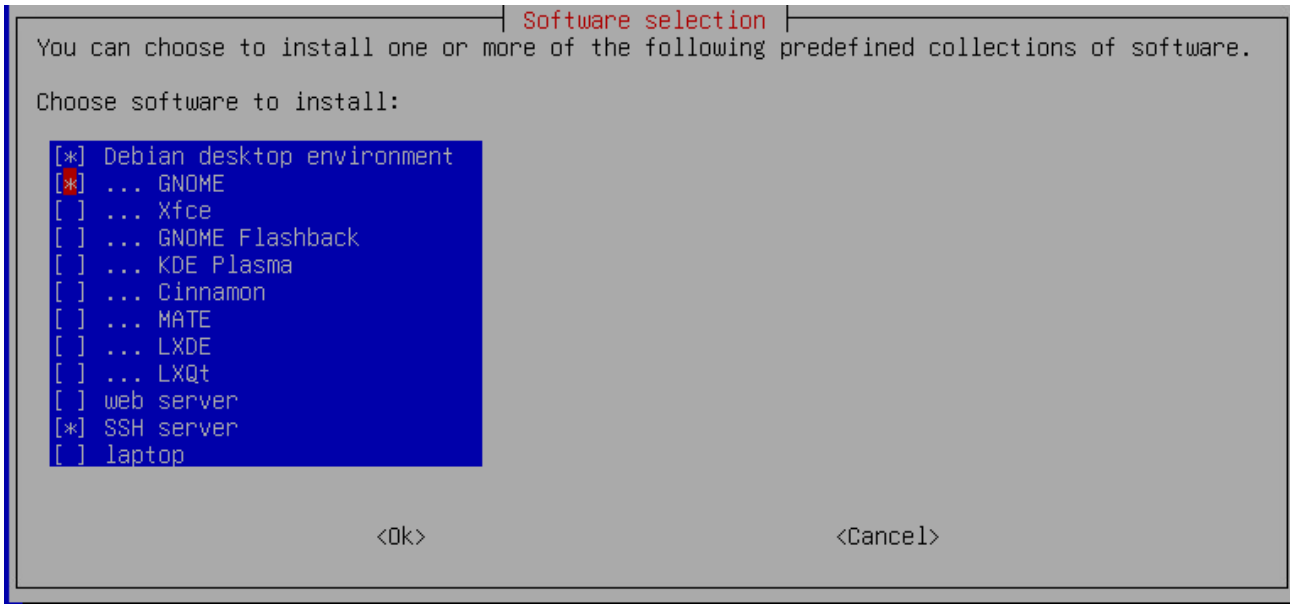
4. Configure Zabbix to monitor CPU usage of computers on BR-CLIENT network via SNMP.

HQ-CL

HQ-CL task1 - GUI

1. Install a graphic environment of your choice.

tasksel

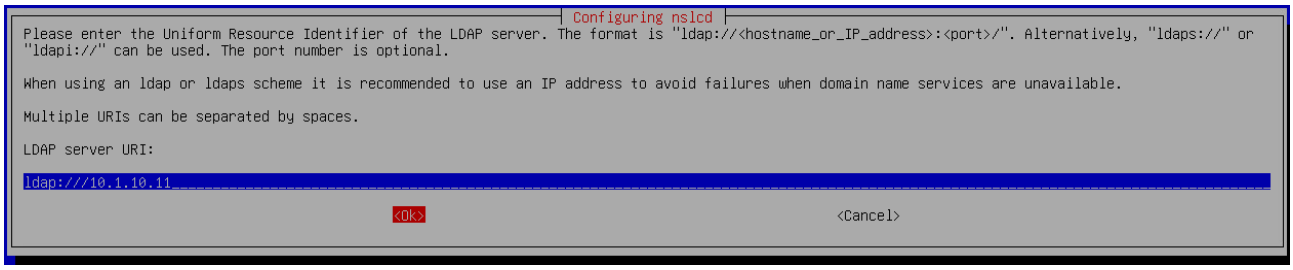


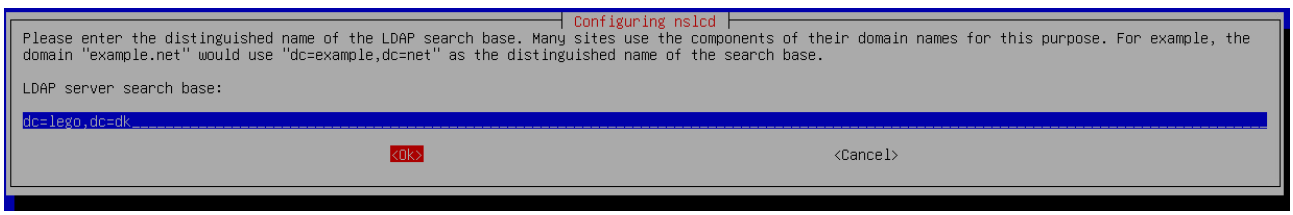
reboot

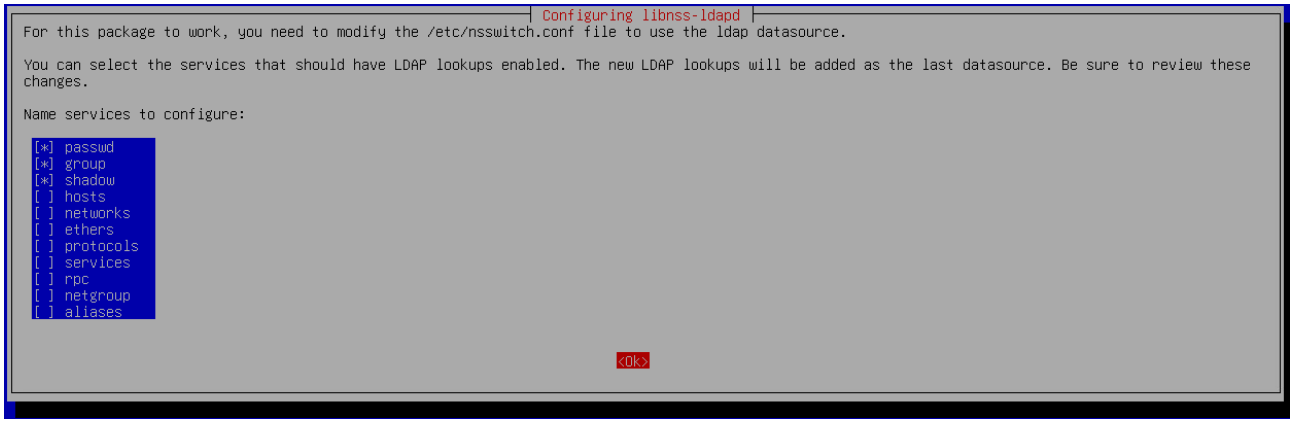
HQ-CL task1 - LDAP client

2. Configure LDAP authentication using the LDAP server. Login with ella.

apt -y install libnss-ldapd libpam-ldapd ldap-utils



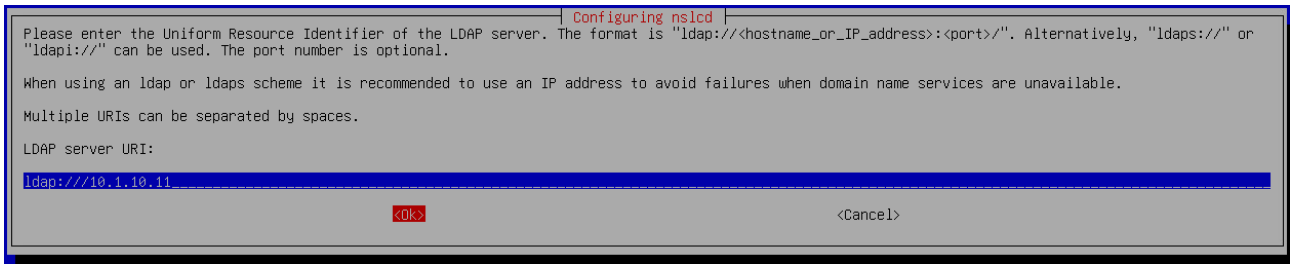


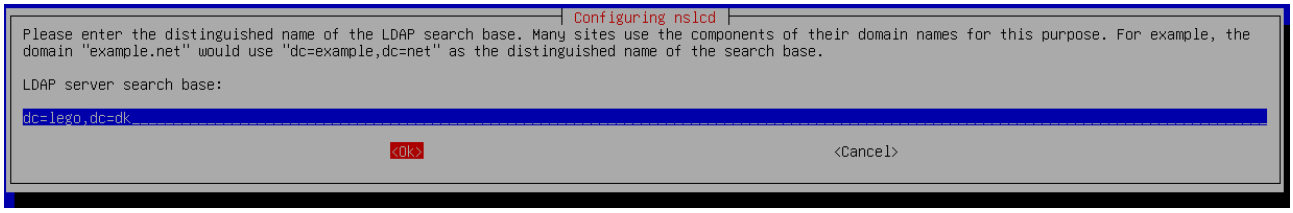


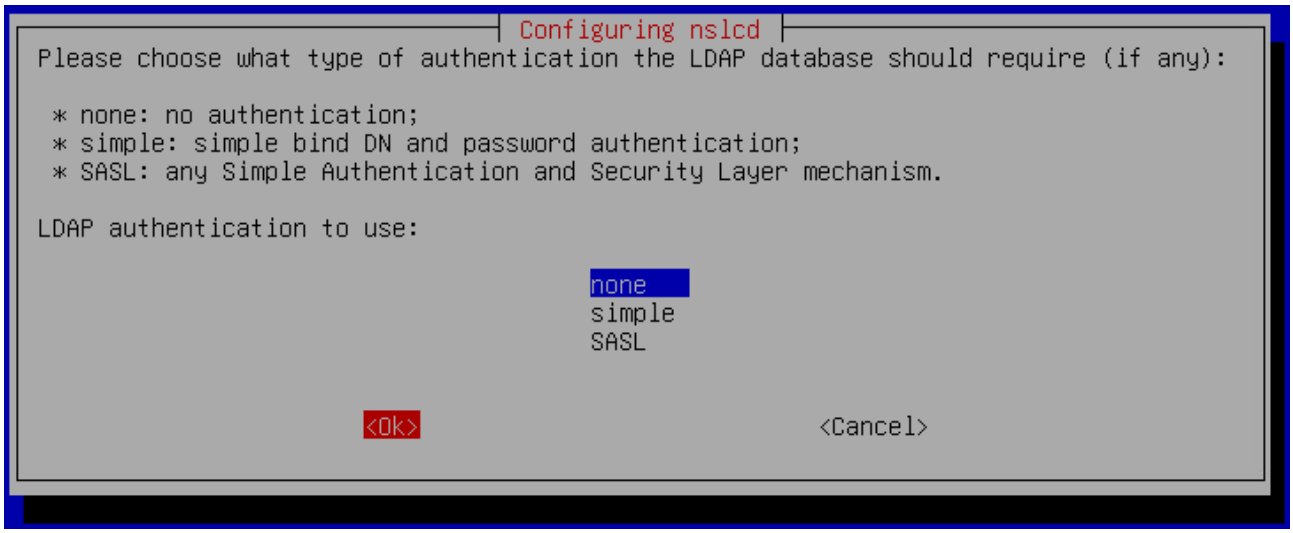
nano /etc/pam.d/common-session

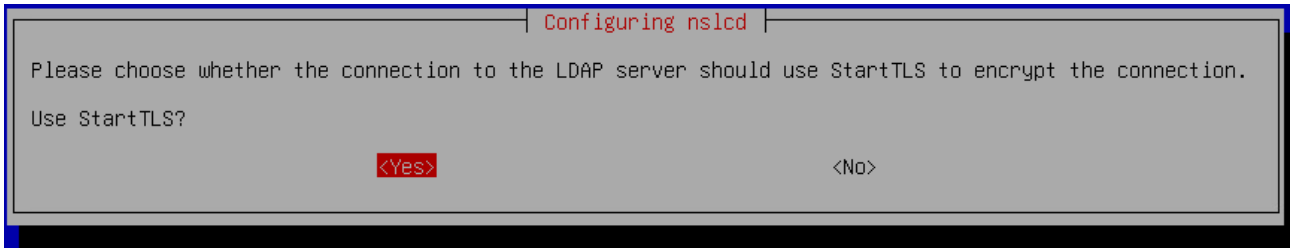
add to the end if need (create home directory automatically at initial login)
session optional pam_mkhomedir.so skel=/etc/skel umask=077

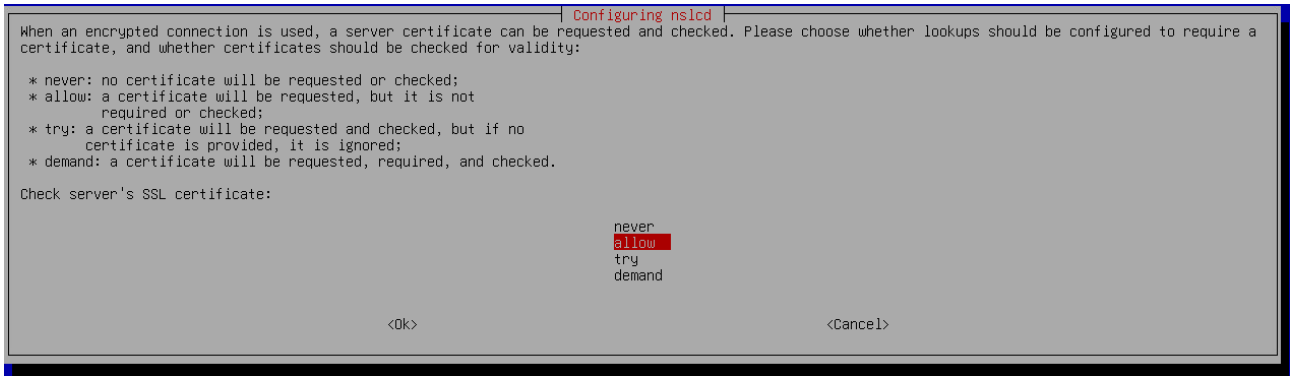
dpkg-reconfigure nslcd



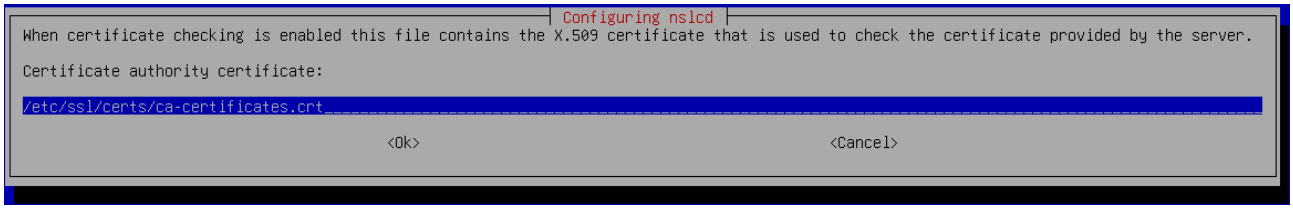








We can leave /etc/ssl/certs/ca-certificates.crt if we have added CA.crt and SUBCA.crt to system.



nano /etc/nslcd.conf

should be something like this:

```
# /etc/nslcd.conf
# nslcd configuration file. See nslcd.conf(5)
# for details.

# The user and group nslcd should run as.
uid nslcd
gid nslcd

# The location at which the LDAP server(s) should be reachable.
uri ldap:///10.1.10.11

# The search base that will be used for all queries.
base dc=lego,dc=dk

# The LDAP protocol version to use.
#ldap_version 3

# The DN to bind with for normal lookups.
#binddn cn=annonymous,dc=example,dc=net
#bindpw secret

# The DN used for password modifications by root.
#rootpwmoddn cn=admin,dc=example,dc=com

# SSL options
ssl start_tls
tls_reqcert allow
tls_cacertfile /etc/ssl/certs/ca-certificates.crt

# The search scope.
#scope sub
```

systemctl restart nscd nslcd

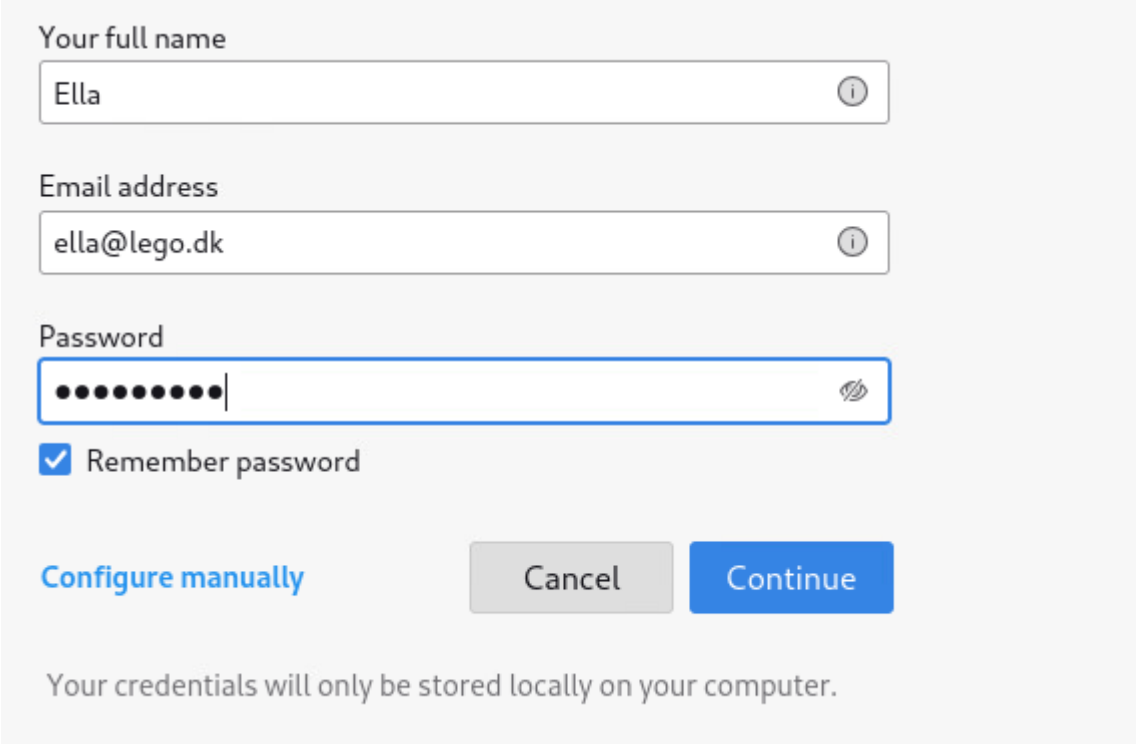
Check:

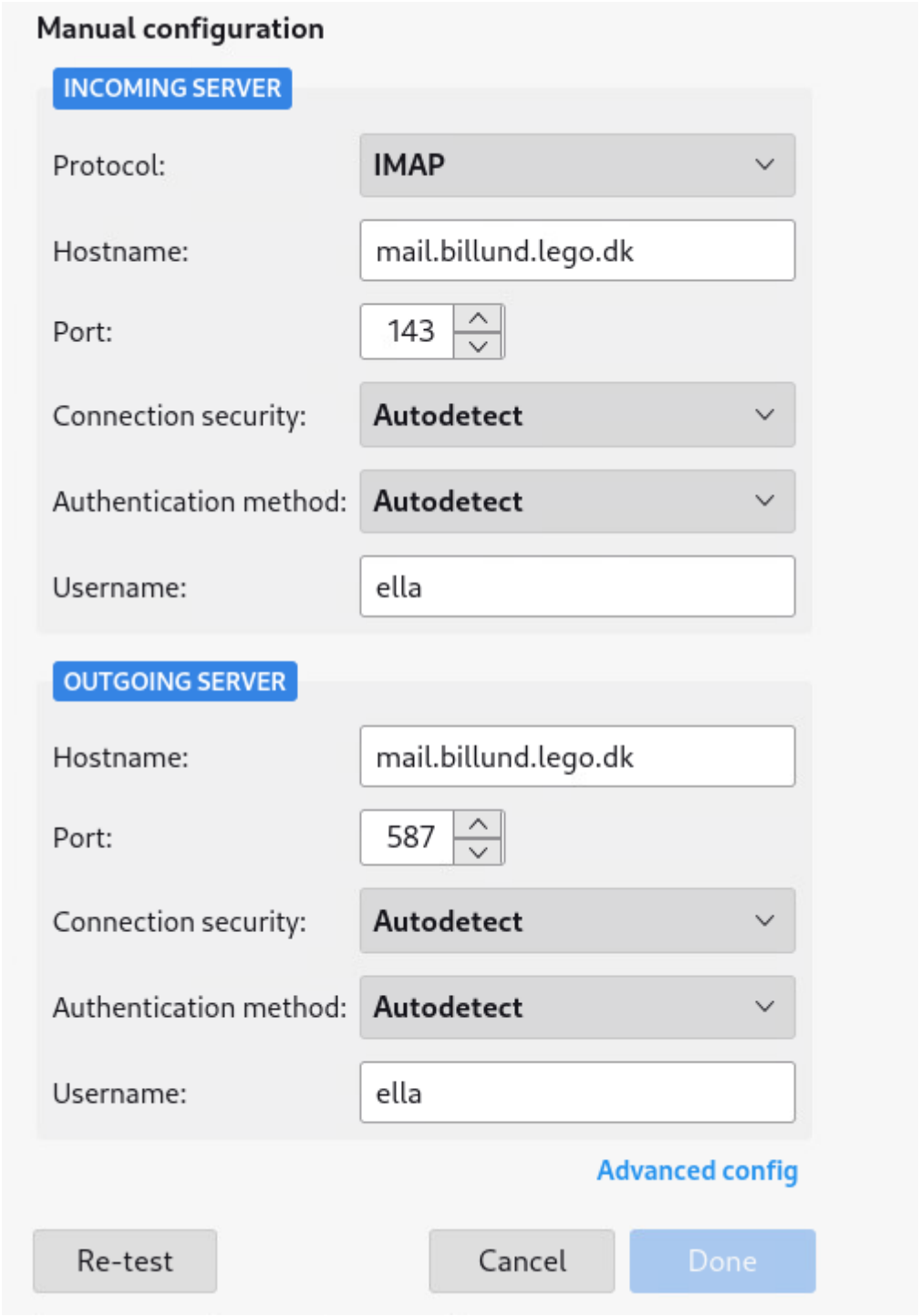
getent passwd

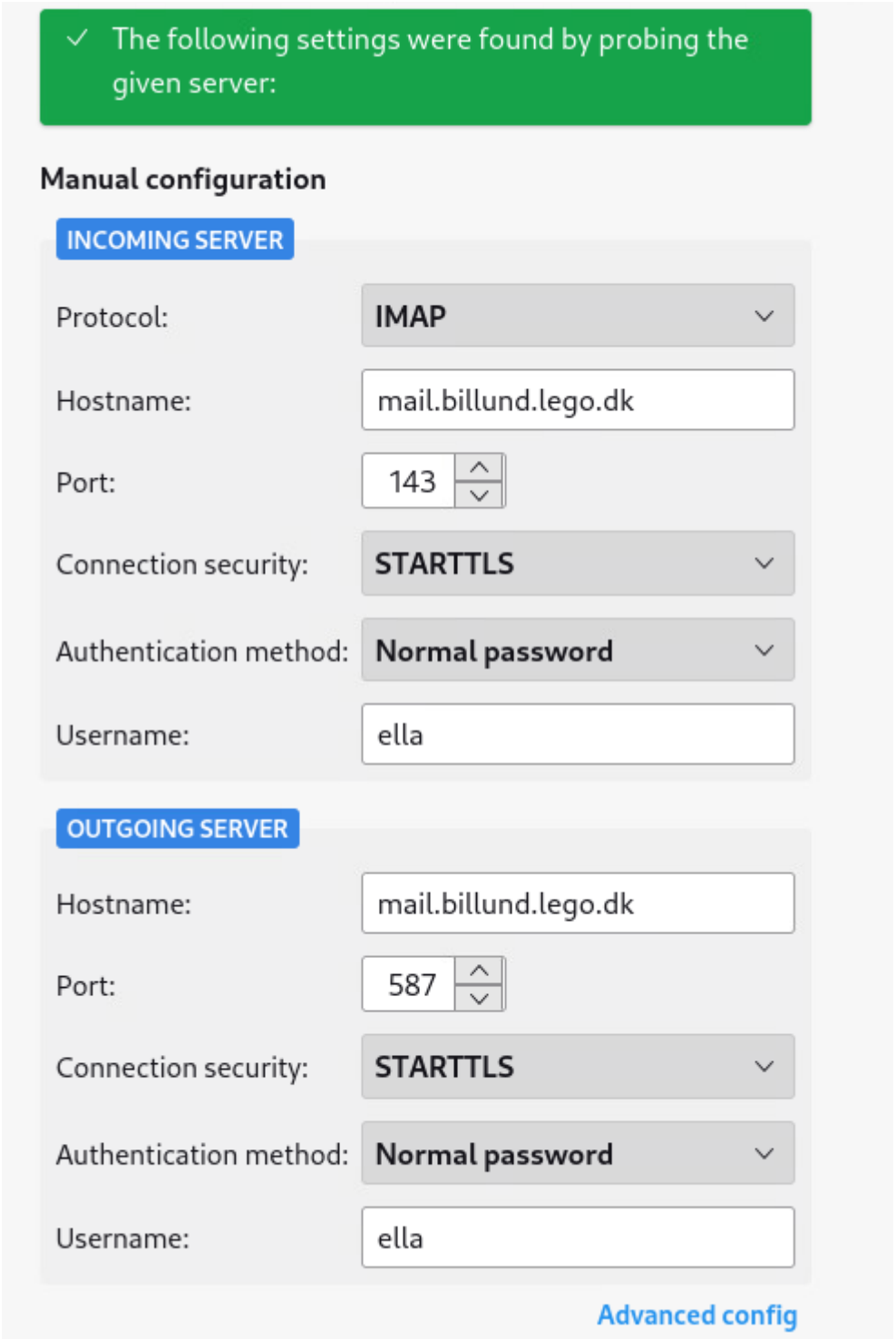
HQ-CL task3 - Thunderbird email client

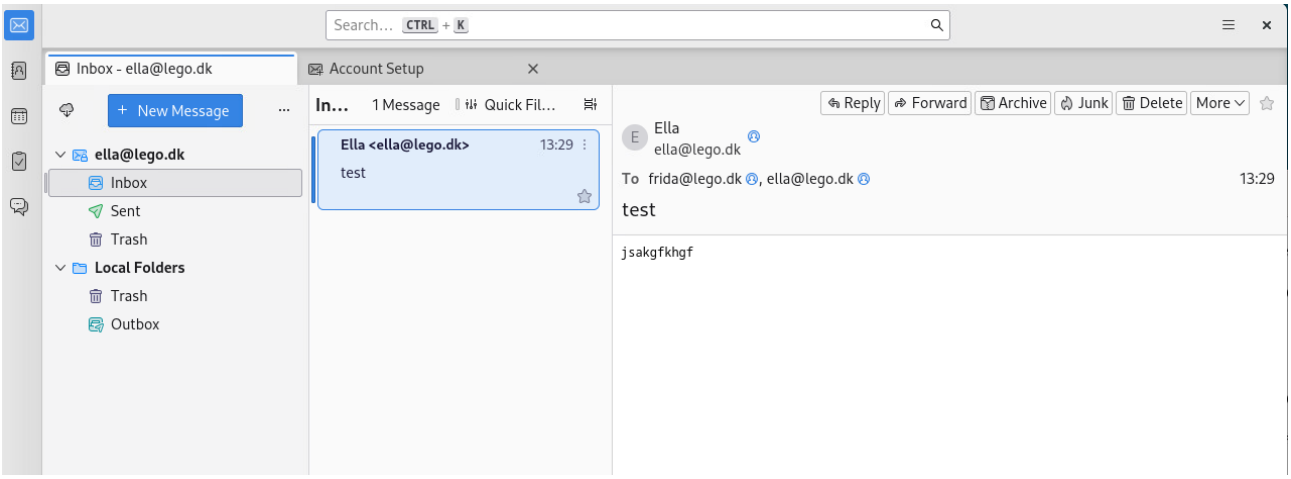
3. Install Thunderbird e-mail client to use with ella@lego.dk. Send email to frida.

Option B: ...using STARTTLS-secured IMAP (port 143)...


Your full name
Ella
Email address
ella@lego.dk
Password
Remember password
Configure manually
Cancel
Continue
Your credentials will only be stored locally on your computer.


Manual configuration
INCOMING SERVER
Protocol:
IMAP
Hostname:
mail.billund.lego.dk
Port:
143
Connection security:
Autodetect
Authentication method:
Autodetect
Username:
ella
OUTGOING SERVER
Hostname:
mail.billund.lego.dk
Port:
587
Connection security:
Autodetect
Authentication method:
Autodetect
Username:
ella
Advanced config
Re-test
Cancel
Done


✓ The following settings were found by probing the given server:
Manual configuration
INCOMING SERVER
Protocol: IMAP
Hostname: mail.billund.lego.dk
Port: 143
Connection security: STARTTLS
Authentication method: Normal password
Username: ella
OUTGOING SERVER
Hostname: mail.billund.lego.dk
Port: 587
Connection security: STARTTLS
Authentication method: Normal password
Username: ella
Advanced config



BRANCH site (Herning)

This is the company's branch site, located in Herning, with limited server services and clients.

R-BR

This is the edge router and firewall of the BRANCH site. For this reason, it should allow devices to reach each other between network segments and the Internet.

You must configure the services of this server according to the following requirements:

R-BR task1 - FW masquerade

1. Traffic from networks on BRANCH site cannot go to the Internet using private addressing.

Activate routing

nano /etc/sysctl.conf

#Comment out this line
net.ipv4.ip_forward=1

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

sysctl -p

Enable nftables

systemctl enable nftables

```
root@R-BR:~# systemctl status nftables.service
● nftables.service - nftables
     Loaded: loaded (/lib/systemd/system/nftables.service; disabled; preset: enabled)
     Active: inactive (dead)
       Docs: man:nft(8)
             http://wiki.nftables.org
root@R-BR:~# systemctl enable nftables.service
Created symlink /etc/systemd/system/sysinit.target.wants/nftables.service → /lib/systemd/system/nftables.service.
root@R-BR:~# systemctl status nftables.service
● nftables.service - nftables
     Loaded: loaded (/lib/systemd/system/nftables.service; enabled; preset: enabled)
     Active: inactive (dead)
       Docs: man:nft(8)
             http://wiki.nftables.org
root@R-BR:~#
```

Prepare nftables configuration file. Add NAT table to /etc/nftables.conf file

cat /etc/nftables.conf

```
root@R-BR:~# cat /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority filter;
        }
        chain forward {
                type filter hook forward priority filter;
        }
        chain output {
                type filter hook output priority filter;
        }
}
root@R-BR:~#
```

cat /usr/share/doc/nftables/examples/nat.nft >> /etc/nftables.conf
cat /etc/nftables.conf

```
root@R-BR:~# cat /usr/share/doc/nftables/examples/nat.nft >> /etc/nftables.conf
root@R-BR:~# cat /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority filter;
        }
        chain forward {
                type filter hook forward priority filter;
        }
        chain output {
                type filter hook output priority filter;
        }
}
#!/usr/sbin/nft -f

table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0;

                #Thanks to nftables maps, if you have a previous iptables NAT (destination NAT) ruleset like this:
                # % iptables -t nat -A PREROUTING -p tcp --dport 1000 -j DNAT --to-destination 1.1.1.1:1234
                # % iptables -t nat -A PREROUTING -p udp --dport 2000 -j DNAT --to-destination 2.2.2.2:2345
                # % iptables -t nat -A PREROUTING -p tcp --dport 3000 -j DNAT --to-destination 3.3.3.3:3456

                # It can be easily translated to nftables in a single line:

                dnat tcp dport map { 1000 : 1.1.1.1, 2000 : 2.2.2.2, 3000 : 3.3.3.3} \
                                : tcp dport map { 1000 : 1234, 2000 : 2345, 3000 : 3456 }
        }

        chain postrouting {
                type nat hook postrouting priority 0;

                #Likewise, in iptables NAT (source NAT):
                # % iptables -t nat -A POSTROUTING -s 192.168.1.1 -j SNAT --to-source 1.1.1.1
                # % iptables -t nat -A POSTROUTING -s 192.168.2.2 -j SNAT --to-source 2.2.2.2
                # % iptables -t nat -A POSTROUTING -s 192.168.3.3 -j SNAT --to-source 3.3.3.3

                # Translated to a nftables one-liner:

                snat ip saddr map { 192.168.1.1 : 1.1.1.1, 192.168.2.2 : 2.2.2.2, 192.168.3.3 : 3.3.3.3 }
        }
}
root@R-BR:~# _
```

nano /etc/nftables.conf

```
....
table ip nat {
        chain postrouting {
                type nat hook postrouting priority 0;

                oif ens192 ip saddr {10.1.10.0/24, 10.1.20.0/24, 10.1.30.0/24} masquerade;
        }
}
```

systemctl restart nftables

R-BR task2 - port forwarding and traffic filter

2. Make sure the public services (DNS, web) of the HQ site are accessible from the internet. Configure firewall with nftables. Incoming packets should be dropped by default. Allow minimal traffic for the services to work.

nano /etc/nftables.conf

```
#!/usr/sbin/nft -f

flush ruleset

table ip filter{
        chain input {
                type filter hook input priority filter;
                policy drop;
                ct state established,related accept;
                udp dport {123} accept;
                tcp dport {22} accept;
        }
                chain forward {
                type filter hook forward priority filter;
                policy drop;
                ct state established,related accept;
                udp dport 53 accept;
                tcp dport { 53, 80, 443 } accept;
        }
        chain output {
                type filter hook output priority filter;
                policy accept;
        }

table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0;
                iif ens192 dnat udp dport map {53 : 10.2.10.11};
                iif ens192 dnat tcp dport map {53 : 10.2.10.11};
                iif ens192 dnat tcp dport map {80 : 10.2.10.11, 443 : 10.2.10.11};
        }

        chain postrouting {
                type nat hook postrouting priority 0;
                oif ens192 ip saddr {10.2.10.0/24, 10.2.30.0/24} masquerade;
        }
}
```

systemctl restart nftables

R-BR task3 - Site-to-site VPN

3. Ensure a secure channel between HQ and BRANCH sites. If this channel breaks, the clients of the BRANCH site can access the public services of HQ.

Step 1: Install openvpn

apt install openvpn

Step 2: certificates and keys. Assure that these files are located in /etc/openvpndirectory (created in HQ-DC task 1 & HQ-SAM-1 task 2). ta.key created in R-HQ task 3.

- CHAIN-CA.crt
- R-BR.key
- R-BR.crt
- ta.key

Step 3: Config file:

nano /etc/openvpn/s2s.conf

```
dev tun
#proto udp
remote 203.0.113.2 1194
ifconfig 10.8.0.2 10.8.0.1
route 10.1.10.0 255.255.255.0
route 10.1.20.0 255.255.255.0
route 10.1.30.0 255.255.255.0
topology p2p

ca /etc/openvpn/CHAIN-CA.crt
key /etc/openvpn/R-BR.key
cert /etc/openvpn/R-BR.crt
tls-crypt /etc/openvpn/ta.key
tls-client

cipher AES-256-GCM

keepalive 10 60
verb 3
log /var/log/openvpn/s2s.log
```

Step 5: enable the service (auto start when boot), start the service and check

systemctl enable openvpn@s2s
systemctl restart openvpn@s2s
systemctl status openvpn@s2s

```
root@R-BR:/etc/openvpn# systemctl status openvpn@s2s
● openvpn@s2s.service - OpenVPN connection to s2s
     Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-06-14 11:38:43 CEST; 2s ago
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 30910 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 10)
     Memory: 1.6M
        CPU: 57ms
     CGroup: /system.slice/system-openvpn.slice/openvpn@s2s.service
             └─30910 /usr/sbin/openvpn --daemon ovpn-s2s --status /run/openvpn/s2s.status 10 --cd /etc/openvpn --config /etc/openvpn/s2s.conf --writepid /run/o

Jun 14 11:38:43 R-BR systemd[1]: Starting openvpn@s2s.service - OpenVPN connection to s2s...
Jun 14 11:38:43 R-BR systemd[1]: Started openvpn@s2s.service - OpenVPN connection to s2s.
lines 1-16/16 (END)
```

ip a

```
root@R-BR:/etc/openvpn# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:91:d3:3c brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 203.0.113.10/29 brd 203.0.113.15 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe91:d33c/64 scope link
       valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:91:d3:46 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 10.2.10.1/24 brd 10.2.10.255 scope global ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe91:d346/64 scope link
       valid_lft forever preferred_lft forever
4: ens256: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:91:d3:50 brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    inet 10.2.30.1/24 brd 10.2.30.255 scope global ens256
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe91:d350/64 scope link
       valid_lft forever preferred_lft forever
87: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.2 peer 10.8.0.1/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::4080:433e:4bfb:74a2/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
root@R-BR:/etc/openvpn#
```

ip r

```
root@R-BR:/etc/openvpn# ip r
default via 203.0.113.9 dev ens192 onlink
10.1.10.0/24 via 10.8.0.1 dev tun0
10.1.20.0/24 via 10.8.0.1 dev tun0
10.1.30.0/24 via 10.8.0.1 dev tun0
10.2.10.0/24 dev ens224 proto kernel scope link src 10.2.10.1
10.2.30.0/24 dev ens256 proto kernel scope link src 10.2.30.1
10.8.0.1 dev tun0 proto kernel scope link src 10.8.0.2
203.0.113.8/29 dev ens192 proto kernel scope link src 203.0.113.10
root@R-BR:/etc/openvpn# _
```

Now communication between sites is possible using private addressing.

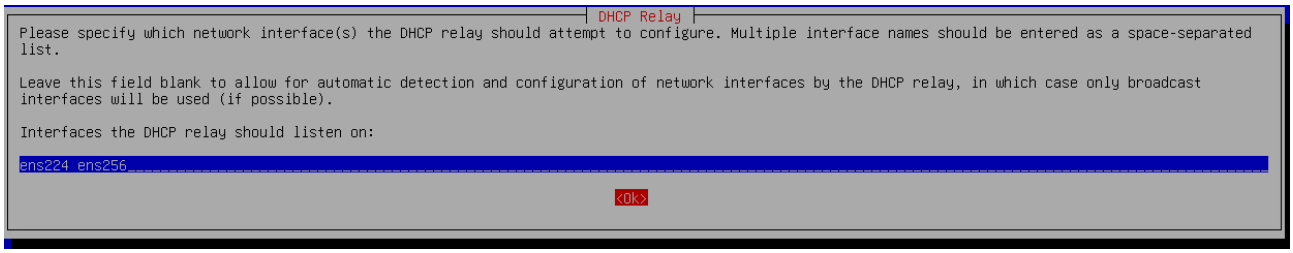
BR-SRV

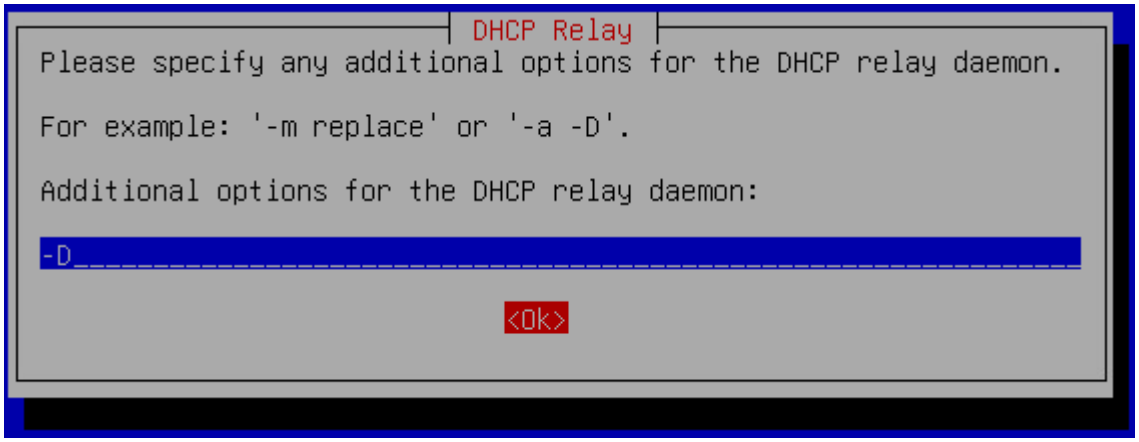
This device has GUI, VSCode preinstalled and Zeal documentation available for Ansible development.

BR-SRV task1 - DHCP

1. Create a DHCP server for the BR-CLIENT network. Assure that BR-CL device always obtains 10.2.30.50/24 IP configuration for automation tasks.

```
apt install isc-dhcp-server

nano /etc/dhcp/dhcpd.conf

authoritative;

subnet 10.2.10.0 netmask 255.255.255.0 {
}
subnet 10.2.30.0 netmask 255.255.255.0 {
        #interface ens192;
        option domain-name-servers 10.2.10.11;
        #option domain-name "herning.lego.dk";
        option routers 10.2.30.1;
        #option broadcast-address 10.2.30.255;
        option ntp-servers 10.2.30.1;
        #option time-offset -10800;
        default-lease-time 86400;
        max-lease-time 172800;
        range 10.2.30.51 10.2.30.150;

}
host BR-CL {
        hardware Ethernet 00:0C:29:FC:01:90;
        fixed-address 10.2.30.50;
}

nano /etc/default/isc-dhcp-server

INTERFACES=”ens192”


systemctl restart isc-dhcp-server


GO TO R-BR

apt install isc-dhcp-relay
```





via config file

nano /etc/default/isc-dhcp-relay

SERVERS=”10.2.10.11”
INTERFACES=”ens224 ens256”
OPTIONS=”-D”

systemctl restart isc-dhcp-relay

BR-SRV task2 - DNS

2. Install and configure a DNS server for both forward and reverse lookups of herning.lego.dk domain. It must serve requests from the Internet as well but cannot use private addressing for that purpose. Configure this DNS server as a secondary DNS server for billund.lego.dk domain. DNS requests for other domains must be forwarded to the correct DNS server.

apt install bind9

nano /etc/bind/med.conf

```
#comment out default zones line

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
//include "/etc/bind/named.conf.default-zones";
```

nano /etc/bind/med.conf.local

```
acl internal {10.1.10.0/24; 10.1.20.0/24; 10.1.30.0/24; 10.2.10.0/24; 10.2.30.0/24; localhost; };
acl external {any; };

view int {
        match-clients { internal; };
        allow-recursion { any; };

        zone "herning.lego.dk" {
                type master;
                file "/etc/bind/db.herning.lego.dk.int";
                allow-transfer { 10.1.20.11; };
                notify yes;
        };

        zone "10.2.10.in-addr.arpa" {
                type master;
                file "/etc/bind/db.10.2.10";
                allow-transfer { 10.1.20.11; };
                notify yes;

        };

        zone "billund.lego.dk" {
                type slave;
                masters { 10.1.20.11; };
                file "db.billund.lego.dk.int";
        };

        zone “20.1.10.in-addr.arpa" {
                type slave;
                masters { 10.1.20.11; };
                file "db.10.1.20";
        };

        include "/etc/bind/zones.rfc1918";
        include "/etc/bind/named.conf.default-zones";
};

view ext{
        match-clients { external; };
        allow-recursion { any; };
```

```
        zone "herning.lego.dk" {
                type master;
                file "/etc/bind/db.billund.lego.dk.ext";
                allow-transfer { 203.0.113.2; };
                notify yes;
        };

        zone "113.0.203.in-addr.arpa" {
                type master;
                file "/etc/bind/db.203.0.113";
                allow-transfer { 203.0.113.2; };
                notify yes;
        };
        zone "billund.lego.dk" {
                type slave;
                masters { 203.0.113.2; };
                file "db.billund.lego.dk.ext";
        };

        include "/etc/bind/zones.rfc1918";
        include "/etc/bind/named.conf.default-zones";
};
```

nano /etc/bind/db.billund.lego.dk.int

```
$TTL    86400
@       IN      SOA     www.herning.lego.dk. root.herning.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ) ; Negative Cache TTL
;
@       IN      NS      www.herning.lego.dk.
www     IN      A       10.2.10.11
```

nano /etc/bind/db10.2.10

```
$TTL    86400
@       IN      SOA     www.herning.lego.dk. root.herning.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ) ; Negative Cache TTL
;
@       IN      NS      www.herning.lego.dk.
```

```
11      IN      PTR     www.herning.lego.dk.
```

nano /etc/bind/db.billund.lego.dk.ext

```
$TTL    86400
@       IN      SOA     www.herning.lego.dk. root.herning.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ); Negative Cache TTL
;
@       IN      NS      www.herning.lego.dk.
www     IN      A       203.0.113.10
```

nano /etc/bind/db.203.0.113

```
$TTL    86400
@       IN      SOA     www.billund.lego.dk. root.billund.lego.dk. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 ); Negative Cache TTL
;
@       IN      NS      www.billund.lego.dk.
2       IN      PTR     www.billund.lego.dk.
2       IN      PTR     mail.billund.lego.dk.
10      IN      PTR     www.herning.lego.dk.
```

systemctl restart bind9

Secondary transferred zone files will appear in /var/cache/bind/.

```
root@BR-SRV:~# ls -la /var/cache/bind/
total 24
drwxrwxr-x  2 root bind 4096 Jun 15 10:59 .
drwxr-xr-x 21 root root 4096 Jun 11 15:19 ..
-rw-r--r--  1 bind bind  333 Jun 15 10:59 db.10.1.20
-rw-r--r--  1 bind bind  344 Jun 15 10:59 db.billund.lego.dk.int
-rw-r--r--  1 bind bind  297 Jun 11 15:20 managed-keys.bind
-rw-r--r--  1 bind bind  785 Jun 11 15:19 managed-keys.bind.jnl
root@BR-SRV:~#
```

If we want to update a zone the SERIAL NUMBER MUST BE CHANGED.
To force updates we can use the following command:

rndc reload

BR-SRV task3 - Web server

3. Install and configure a web service. Serve the https://www.herning.lego.dk site using a valid certificate signed by the company's Sub CA. Website should show the text "Welcome to Herning". Remove server signatures and header tokens from web server.

Assure that the following certs and keys are in BR-SRV:

- www.herning.lego.dk.key
- www.herning.lego.dk.crt
- CA.crt
- SUBCA.crt
- CHAIN-CA.crt

Install apache2

```
apt install apache2
```

Edit the message on the main webpage:

```
echo "Welcome to Herning" > /var/www/html/index.html
```

Copy cert and key to /etc/ssl/certs and /etc/ssl/private directories:

```
cp www.herning.lego.dk.crt  /etc/ssl/certs
cp www.herning.lego.dk.key  /etc/ssl/private
```

Trust CA.crt and SUBCA.crt:

```
cp *CA*crt /usr/local/share/ca-certificates/
update-ca-certificates
```

Enable SSL module:

```
a2enmod ssl
```

Enable HTTPS site:

```
a2ensite default-ssl.conf
```

Configure certificate paths:

```
nano /etc/apache2/sites-enabled/default-ssl.conf
```

```
<VirtualHost *:80>
        DocumenRoot /var/www/html
        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/www.herning.lego.dk.crt
        SSLCertificateKeyFile       /etc/ssl/private/www.herning.lego.dk.key
</VirtualHost>
```

Configure HTTP → HTTPS redirection:

nano /etc/apache2/sites-enabled/000-default.conf

Add Redirect line:

<VirtualHost *:443>
DocumenRoot /var/www/html
Redirect / https://www.herning.lego.dk/
</VirtualHost>

Find ServerSignature and ServerTokens configuration:

grep -nr ServerSignature /etc/apache2/

```
root@BR-SRV:~# grep -nr ServerSignature /etc/apache2/
/etc/apache2/conf-available/localized-error-pages.conf:31:# ServerAdmin email address regardless of the setting of ServerSignature.
/etc/apache2/conf-available/security.conf:22:ServerSignature Off
/etc/apache2/conf-available/security.conf:23:#ServerSignature On
root@BR-SRV:~#
```

grep -nr ServerTokens /etc/apache2/

```
root@BR-SRV:~# grep -nr ServerTokens /etc/apache2/
/etc/apache2/conf-available/security.conf:5:# ServerTokens
/etc/apache2/conf-available/security.conf:11:#ServerTokens Minimal
/etc/apache2/conf-available/security.conf:12:ServerTokens OS
/etc/apache2/conf-available/security.conf:13:#ServerTokens Full
root@BR-SRV:~#
```

Both options are in /etc/apache2/conf-available/security.conf

nano /etc/apache2/conf-available/security.conf

Comment out “ServerTokens OS” and write “ServerTokens Prod”. Comment out “ServerSignature On” and uncomment “ServerSignature Off”:

#ServerTokens Minimal
#ServerTokens OS
#ServerTokens Full
ServerTokens Prod

ServerSignature Off
#ServerSignature On

systemctl restart apache2

Nginx

```
apt install nginx nginx-extras
echo “Welcome to Herning” > /var/www/html/index.html
vim /etc/nginx/nginx.conf
# Añadir lo siguiente
server_tokens off;
more_clear_headers ‘Server’;
vim /etc/nginx/sites-enabled/default
# Añadir lo pertinente al certificado y la redirección al puerto 443.
# NO HACE FALTA CONFIGURAR SERVERNAME, ES UN ÚNICO SERVIDOR QUE PUEDE RESPONDER TODO
```

BR-SRV task4 - SFTP

4. Configure an SFTP server for LDAP user "carl" to be able to access and edit files in the web root directory.

BR-SRV task5 - syslog server

5. Deploy a centralized syslog server with TLS to collect logfiles from clients on BR-CLIENT.
   (a) Incoming logs related to DHCP should be written to /log/dhcp.log.
   (b) All other incoming logs should be written to /log/dump.log.

[redacted]

apt install rsyslog

Descomentar estas líneas en `/etc/rsyslog.conf`:

```
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
```

Crear un fichero `/etc/rsyslog.d/10-remote.conf` que tenga esto:

```
# Templates
$template DHCPLog,"/log/dhcp.log"
$template DumpLog,"/log/dump.log"

# Match logs that contain 'dhcp' in the message (case-insensitive)
if ($msg contains_i "dhcp") then ?DHCPLog
& stop

# Everything else
```

```
*.* ?DumpLog
```

Crear el directorio `/log` que sea de `syslog:adm`.

BR-CLIENT network

Clients on BR-CLIENT network will be configured automatically using Ansible playbooks from BR-SRV. For testing purposes, only BR-CL client will be used, which is already preconfigured (SSH service and SSH key authentication).

There is a preconfigured hosts file located under /etc/ansible/hosts in BR-SRV, with just one entry, the one related to BR-CL. Do not change this file. Assure that BR-CL is configured with 10.2.30.50/24 IP configuration, via DHCP or using static configuration, so the SSH connection works perfectly. You can connect with a user called 'ansible' from BR-SRV to BR-CL with a preinstalled private key.

For marking, BR-CL machine will be restored to "Start" snapshot and all playbooks will be run in order using the command “ansible-playbook playbookname.yml” in the /ansible directory of BR-SRV.

Create the following playbooks:

Ansible task1 - Basic config

1. 1-basic.yml for configuring the hostname, time zone and NTP. All hosts should receive the hostname based on the "hostname" variable in /etc/ansible/hosts file and NTP should sync to R-BR.

Instead of learning by heart that messy regexps, it’ll be much easier if we manually comment “pool …” and enter “server 10.2.30.1 iburst” in /etc/ntpsec/ntp.conf server file, and then copy this file in ansible/files folder.

```
---
- name: Configure hostname, timezone, and NTP
  hosts: all
  become: yes

  vars:
    timezone: "Europe/Copenhagen"
    files_src_dir: files
    files_dst_dir: /etc/ntpsec

  tasks:
    - name: Set hostname
      ansible.builtin.hostname:
        name: "{{ hostname }}"

    - name: Set timezone
      ansible.builtin.timezone:
        name: "{{ timezone }}"

    - name: Install ntpsec for NTP
      ansible.builtin.package:
        name: ntpsec
        state: present

    - name: Copy ntp.conf to BR-CLIENT
      copy:
```

```
    src: "{{ files_src_dir }}/ntp.conf"
    dest: "{{ files_dst_dir }}"
    owner: root
    group: root
    mode: '0600'

- name: Comment out default pool NTP servers
  ansible.builtin.replace:
    path: /etc/ntpsec/ntp.conf
    regexp: '^pool (.*)$'
    replace: '# pool \1'

- name: Configure NTP to use R-BR
  ansible.builtin.lineinfile:
    path: /etc/ntpsec/ntp.conf
    regexp: '^server {{ ntp_server }}'
    line: "server {{ ntp_server }} iburst"
    insertafter: '^#.*pool.*'
    state: present

- name: Restart and enable ntpsec
  ansible.builtin.service:
    name: ntpsec
    state: restarted
    enabled: true
```

Ansible task2 - syslog

2. 2-syslog.yml for sending log information to BR-SRV.

Ansible task3 - SNMP

3. 3-snmp.yml for configuring SNMP - Zabbix should be able to read the CPU usage of this machine.

Ansible task4 - LDAP client

4. 4-ldap.yml for configuring LDAP authentication using the LDAP server.

Ansible task5 - share auto mount

5. 5-share.yml for configuring share directory auto mount.

Ansible task6 - web server

6. 6-webserver.yml for installing and configuring a web server that will show the message "Website of <hostname>", based on the "hostname" variable in /etc/ansible/hosts file.

```
---

- name: Install and configure web server
```

```
 hosts: all
 become: yes
 vars:
  webpage_message: "Website of {{ inventory_hostname }}"

 tasks:
  - name: Install Apache web server
    ansible.builtin.package:
      name: apache2
      state: present

  - name: Ensure Apache is running and enabled
    ansible.builtin.service:
      name: apache2
      state: started
      enabled: yes

  - name: Create custom index.html
    ansible.builtin.copy:
      dest: /var/www/html/index.html
      content: "{{ webpage_message }}"
      owner: root
      group: root
      mode: '0644'
```

Change proposed by Hungary's team: The webserver must also be accessible via the HTTPS protocol.

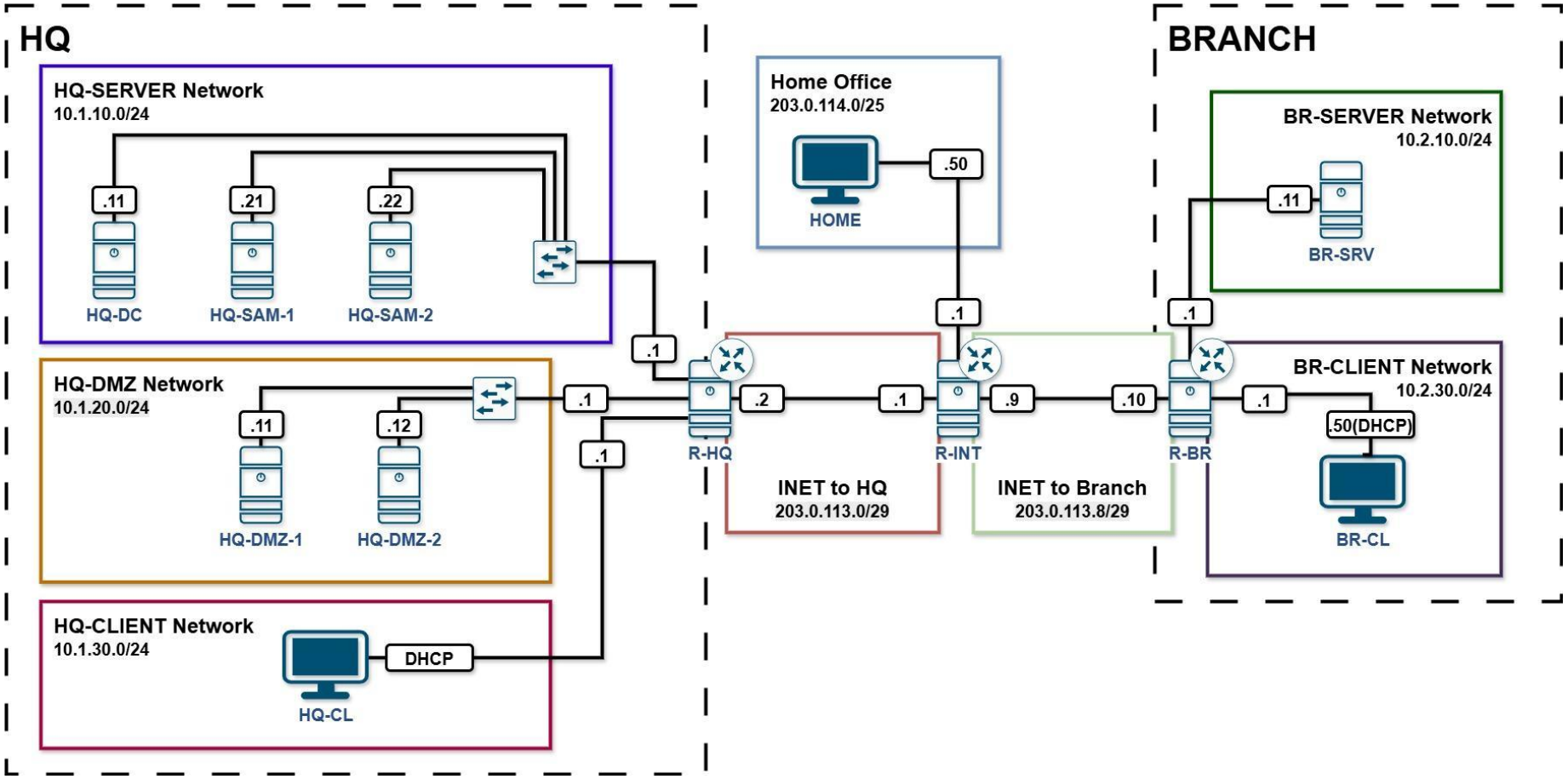
```
---
- name: Install and Configure HTTP HTTPS Apache Server on BR-CLIENT
  hosts: br-client
  become: yes

  vars:
    webpage_message: "Website of {{ inventory_hostname }}"
    certs_src_dir: files
    ssl_cert: /etc/ssl/certs/www.crt
    ssl_key: /etc/ssl/private/www.key
```

```
  ssl_ca_cert: /etc/ssl/certs/ca.crt

tasks:
  - name: Install Apache web server
    ansible.builtin.package:
      name: apache2
      state: present

  - name: Enable SSL module
    command: a2enmod ssl
    args:
      creates: /etc/apache2/mods-enabled/ssl.load

  - name: Copy www.crt to BR-CLIENT
    copy:
      src: "{{ certs_src_dir }}/www.crt"
      dest: "{{ ssl_cert }}"
      owner: root
      group: root
      mode: '0644'

  - name: Copy www.key to BR-CLIENT
    copy:
      src: "{{ certs_src_dir }}/www.key"
      dest: "{{ ssl_key }}"
      owner: root
      group: root
      mode: '0600'

  - name: Copy ca.crt to BR-CLIENT
    copy:
      src: "{{ certs_src_dir }}/ca.crt"
      dest: "{{ ssl_ca_cert }}"
      owner: root
      group: root
      mode: '0644'
```

```
- name: Copy ca.key to BR-CLIENT
  copy:
    src: "{{ certs_src_dir }}/ca.key"
    dest: /etc/ssl/private/ca.key
    owner: root
    group: root
    mode: '0600'

- name: Create SSL Virtual Host file
  copy:
    dest: /etc/apache2/sites-available/default-ssl.conf
    content: >
      <VirtualHost *:443>
        DocumentRoot /var/www/html

         SSLEngine on
        SSLCertificateFile      {{ ssl_cert }}
        SSLCertificateKeyFile   {{ ssl_key }}
        SSLCACertificateFile    {{ ssl_ca_cert }}
      </VirtualHost>

- name: Enable default SSL site
  command: a2ensite default-ssl
  args:
    creates: /etc/apache2/sites-enabled/default-ssl.conf

- name: Create custom index.html
  ansible.builtin.copy:
    dest: /var/www/html/index.html
    content: "{{ webpage_message }}"
    owner: root
    group: root
    mode: '0644'

- name: Enable and restart Apache
```

```
ansible.builtin.service:
  name: apache2
  state: restarted
  enabled: true
```

Appendix A: Network topology



Appendix B: Containers, Objects and Users

LDAP OUs

OU name
Billund
Herning

LDAP Groups

Groups	DN
admins	CN=admins,OU=Billund,DC=lego,DC=dk
billund	CN=billund,OU=Billund,DC=lego,DC=dk
herning	CN=herning,OU=Billund,DC=lego,DC=dk

LDAP Users

Username	E-mail Address	Home Directory Location on HQ-DC	Membership	Group Membership
admin	admin@lego.dk	/share/users/admin	Billund	admins
frida	frida@lego.dk	/share/users/frida	Billund	billund
ella	ella@lego.dk	/share/users/ella	Billund	billund
carl	carl@lego.dk	/share/users/carl	Herning	herning