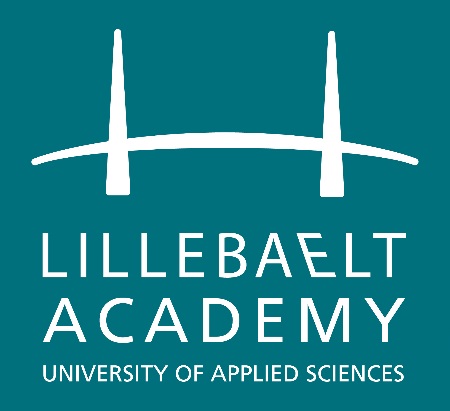
**IT Technology**

**Recovery Plan**



LILLEBAELT ACADEMY

UNIVERSITY OF APPLIED SCIENCE

Author

Group 2

Tuesday 28 November 2017

**Table of Contents**

[1 Requirements: 1](#_Toc499637211)

[1.1 vMX: 1](#_Toc499637212)

[1.2 vSRX: 1](#_Toc499637213)

[1.3 Ubuntu machines: 1](#_Toc499637214)

[2 Relations: 1](#_Toc499637215)

[3 Protocols: 2](#_Toc499637216)

[4 VLANS 3](#_Toc499637217)

[5 LibreNMS 3](#_Toc499637218)

[6 Clients setup 4](#_Toc499637219)

[7 vMXs set up: 5](#_Toc499637220)

[7.1 London FP 6](#_Toc499637221)

[7.2 London CP 7](#_Toc499637222)

[7.3 Manchester FP 7](#_Toc499637223)

[7.4 Manchester CP 8](#_Toc499637224)

[7.5 Birmingham FP 8](#_Toc499637225)

[7.6 Birmingham CP 9](#_Toc499637226)

[8 vSRX Firewall set up: 9](#_Toc499637227)

[9 Appendix A(London) 10](#_Toc499637228)

[10 Appendix B(Manchester) 11](#_Toc499637229)

[11 Appendix C(Birmingham) 13](#_Toc499637230)

[12 Appendix D(Firewall) 15](#_Toc499637231)

[13 Appendix E(LibreNMS) 15](#_Toc499637232)

# Requirements:

In our system we got three vMX 17.2R1.13 routers for each city that is our project (London, Manchester, Birmingham), one vSRX (model) for a firewall, three virtual clients (Ubuntu 16), one virtual monitoring server (Ubuntu, LibreNMS) and physical PE router. All virtual devices are running on an ESXi(64GB of RAM, 1.8T of Memory, 16 CPUs)

## vMX:

* Control Panel – 2GB of RAM, 1 vCPU
* Forwarding Plane -8GM of RAM, 4 vCPUs

## vSRX:

* 2GB of RAM, 2 vCPUs

## Ubuntu machines:

* 2GB of RAM, 60GB of Memory, 2 vCPUs

# Relations:

Every connection between the routers is point-to-point:

* London – Manchester

FD00:0:2::1:0/127 FD00:0:2::1:1/127

* London – Birmingham

FD00:0:2::2:0/127 FD00:0:2::2:1/127

* Manchester – Birmingham

FD00:0:2::5:0/127 FD00:0:2::5:1/127

* London – Firewall

FD00:0:2::3:0/127 FD00:0:2::3:1/127

* Firewall – PE

FD00:0:2::7:0/127 FD00:0:2::7:1/127

The connections between the routers, the clients and the monitoring server

* London
* Router – Client

FD00:0:2::4:0/64 FD00:0:2::4:1/127??

* Router – Monitoring Server

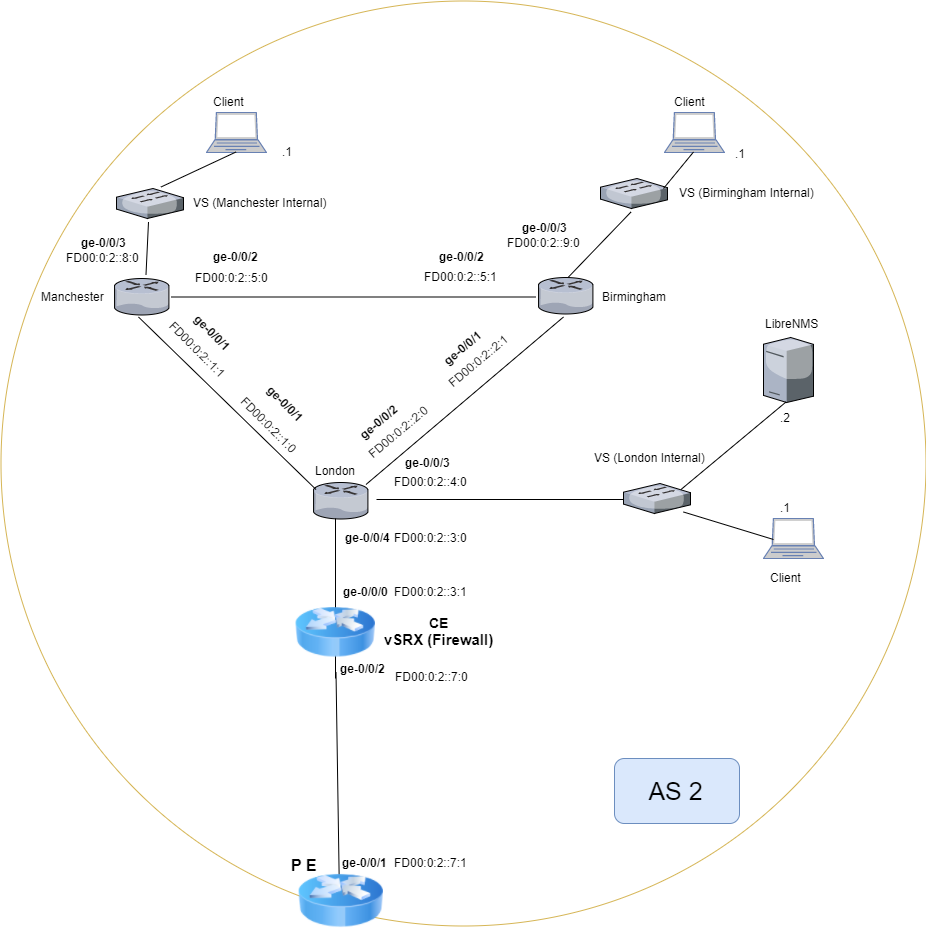
FD00:0:2::4:0/64 FD00:0:2::4:2/127??

* Manchester
* Router – Client

FD00:0:2::8:0/127 FD00:0:2::8:1/127

* Birmingham
* Router – Client

FD00:0:2::9:0/127 FD00:0:2::9:1/127

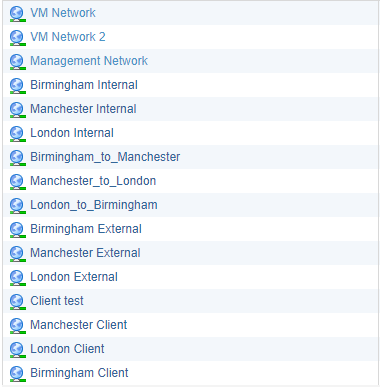


# Protocols:

To make every device in our system to be able to communicate with every other we use OSPFv3 which is configure on every router. To do this we set the OSPFv3 in protocols with the area in which they will operate and after this we added the interfaces that are going to operate there.

# VLANS

To make our machines communicate between each other we are using the following small networks:



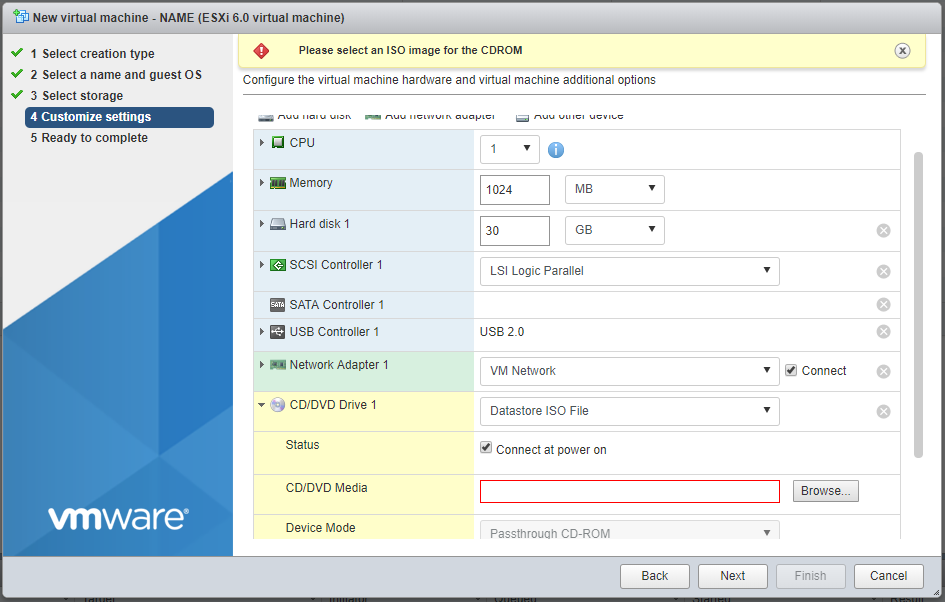
# LibreNMS

We use UbuntuOS for out LibreNMS Server. You can find the instructions for the way to set it up in Appendix E.

# Clients setup

To set up the clients you need to download the Ubuntu OS from <https://www.ubuntu.com/download/desktop>. After you finish your downloading you can add it to the ESXi data storage or you can use it from your PC as well.

1. Change Hard disk 1 to 30 GB
2. Change Network Adapter 1 to VM network(so you can have Internet to download whatever you need on you machine and later change to the network in which this machine will operate)
3. Change CD/DVD Drive 1 to Datastore ISO File and click on Browse…

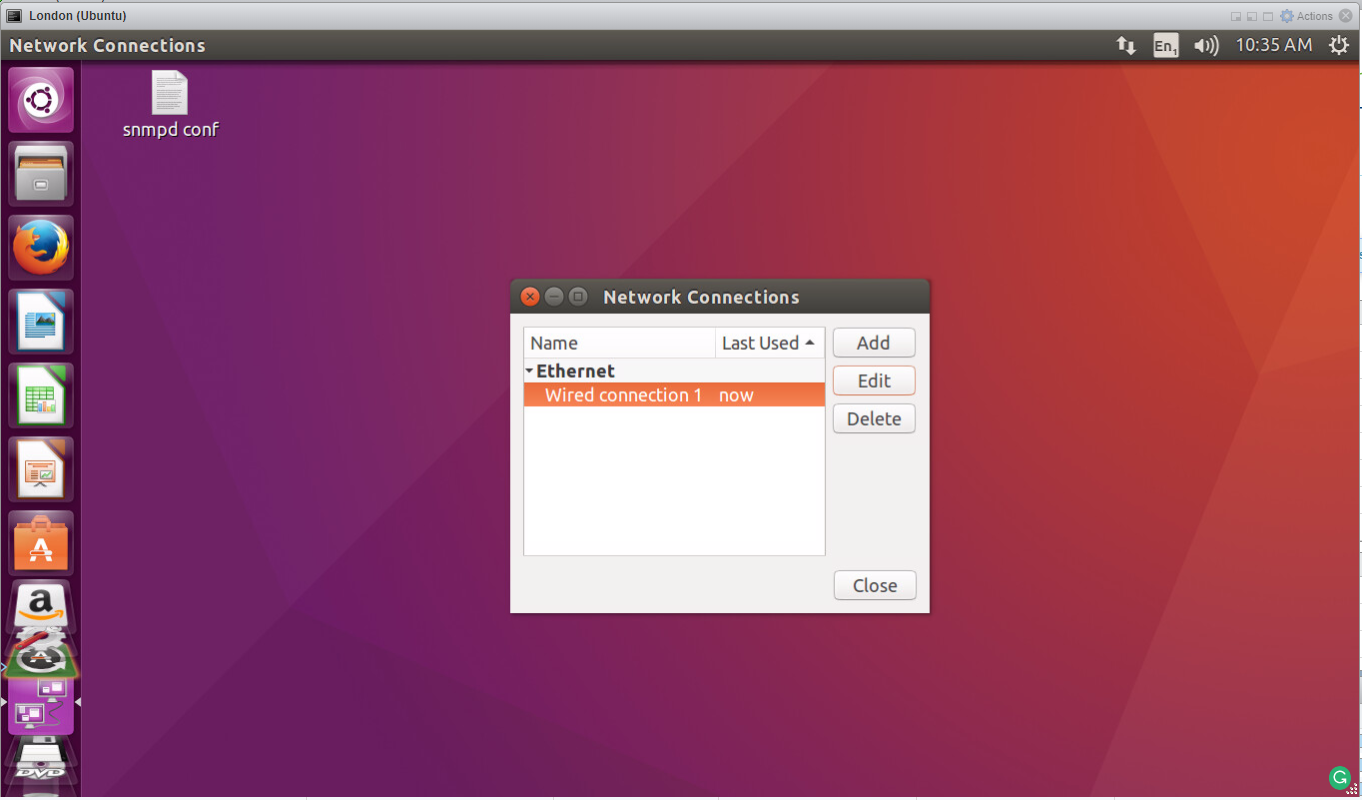


After this you should start you’re the machine and set up user, pass, etc. After your this your machine should be ready to be used. When you download everything, you need change the network to the one that the machine is going to operate in.

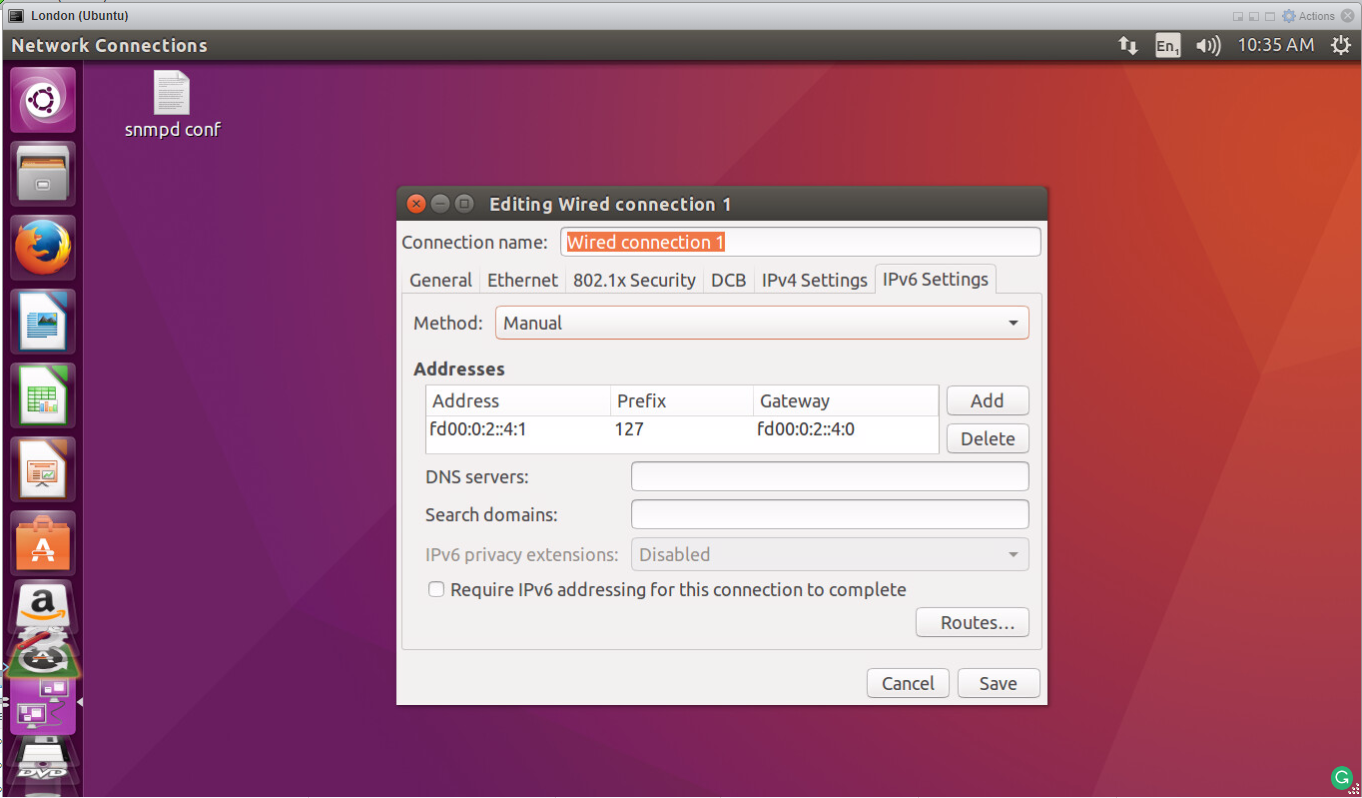
1. Press on the icon with the up and down arrows to open the connection and go in Edit Connections…



1. Select the connection and press Edit



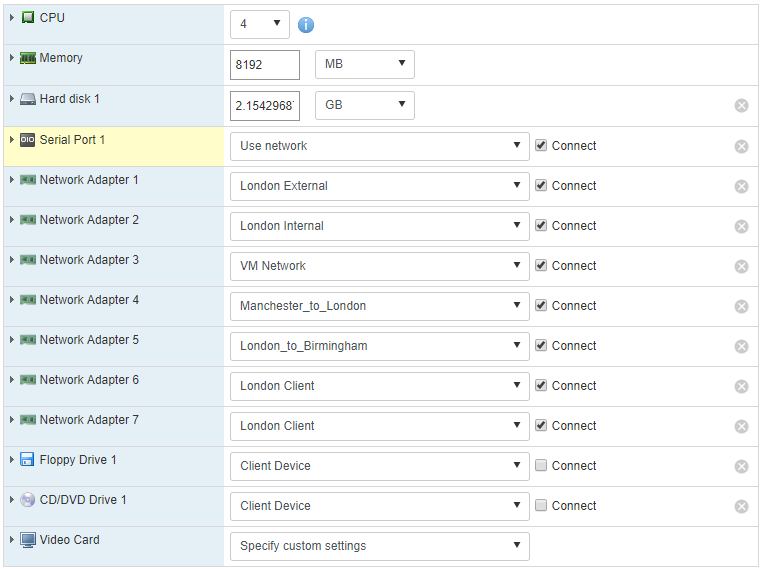
1. Go to IPv6 Settings and give the setting you need(on the picture is shown the connection settings for the London Clients)



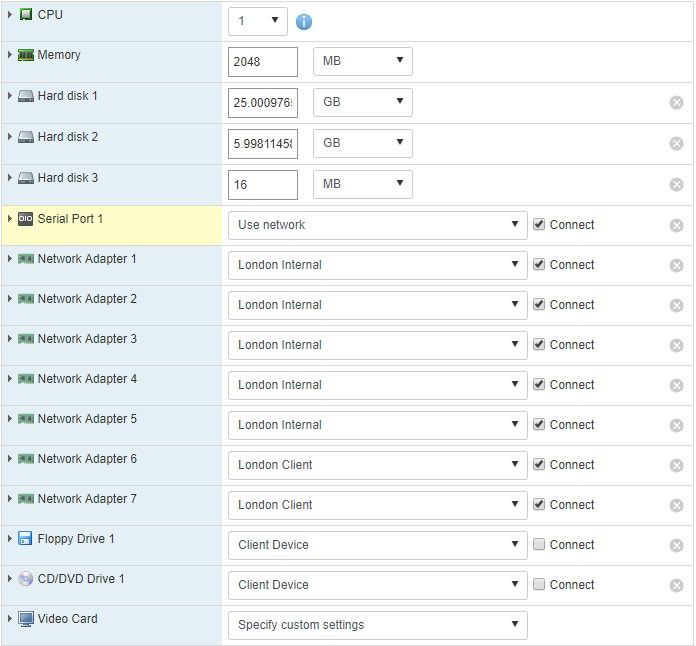
# vMXs set up:

As soon as you have install both Forwarding Plane and Control Plane make sure you have set the configurations as on the following pictures.

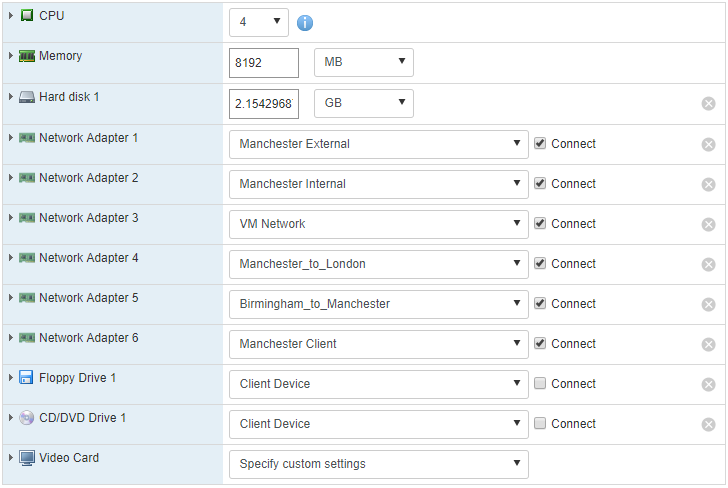
## London FP



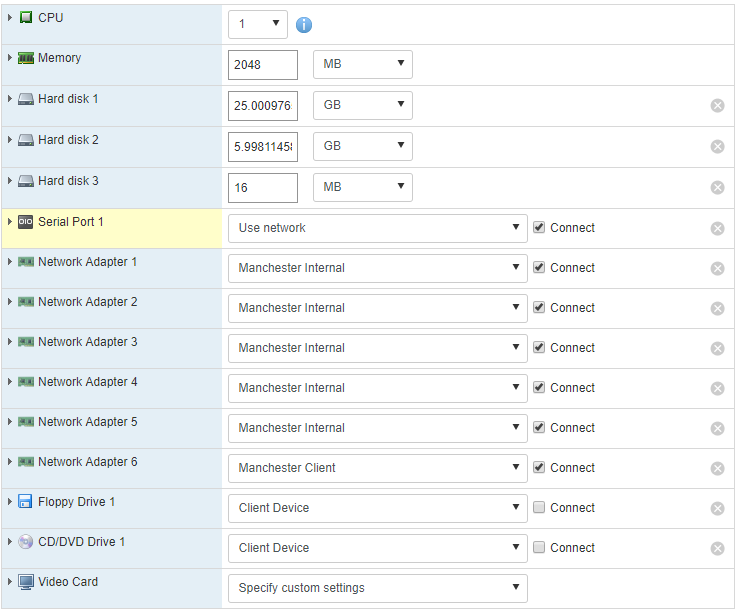
## London CP



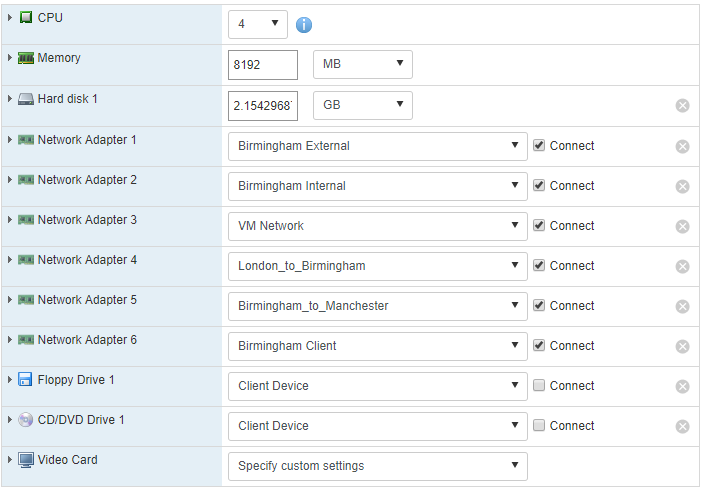
## Manchester FP



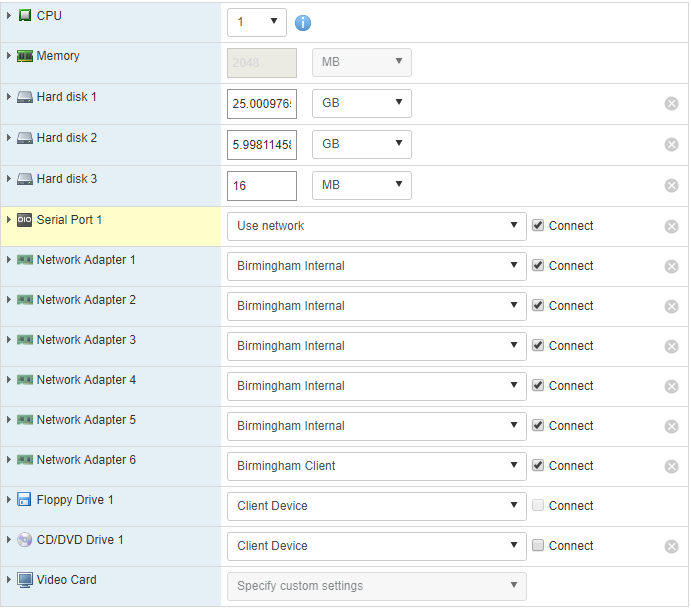
## Manchester CP



## Birmingham FP



## Birmingham CP

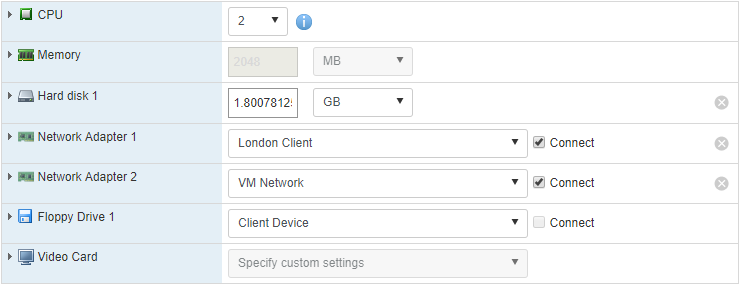


After you are done with the installation of all machines run them and open the CPs.

* For London CP we use the configurations form Appendix A
* For Manchester CP we use the configurations form Appendix B
* For Birmingham CP we use the configurations form Appendix C

# vSRX Firewall set up:

As soon as you have install the vSRX make sure you have set the configurations as on the following picture.



After you are done with the installation of the machine, run it and use the configurations from Appendix D

# Appendix A(London)

version 17.2R1.13;

system {

host-name London;

root-authentication {

encrypted-password "$6$Q6DwjOPi$CfaWyzaaaH2.W/Rvi7PPyN.MjhQsuT.P5zgCdgj3NIlpg.lGN0gMh7E3HrO3F1pIVAoA/tofw4JRMH/qb2Z/e."; ## SECRET-DATA

}

syslog {

user \* {

any emergency;

}

file messages {

any notice;

authorization info;

}

file interactive-commands {

interactive-commands any;

}

}

processes {

dhcp-service {

traceoptions {

file dhcp\_logfile size 10m;

level all;

flag all;

}

}

}

}

interfaces {

ge-0/0/1 {

unit 0 {

family inet {

address 10.217.16.110/24;

}

family inet6 {

address fd00:0:2::1:0/127;

}

}

}

ge-0/0/2 {

unit 0 {

family inet {

address 10.217.16.120/24;

}

family inet6 {

address fd00:0:2::2:0/127;

}

}

}

ge-0/0/3 {

unit 0 {

family inet {

address 10.217.16.130/24;

}

family inet6 {

address fd00:0:2::4:0/64;

}

}

}

ge-0/0/4 {

unit 0 {

family inet6 {

address fd00:0:2::3:0/127;

}

}

}

fxp0 {

unit 0 {

family inet {

dhcp {

vendor-id Juniper-vmx;

}

}

}

}

lo0 {

unit 0 {

family inet6 {

address fd00:0:2::1/128;

}

}

}

}

snmp {

name "London CP";

description "London CP";

location London;

client-list list0 {

10.217.19.237/24;

}

community public {

authorization read-write;

}

}

routing-options {

router-id 192.168.2.10;

autonomous-system 65210;

}

protocols {

ospf3 {

area 0.0.0.2 {

interface ge-0/0/1.0;

interface ge-0/0/2.0;

interface ge-0/0/3.0;

interface ge-0/0/4.0;

interface lo0.0;

}

}

}

policy-options {

policy-statement bgp {

term bgp {

from protocol bgp;

then accept;

}

}

policy-statement ospf3 {

term ospf3 {

then accept;

}

}

policy-statement send-v6 {

from {

family inet6;

protocol direct;

}

then accept;

}

}

# Appendix B(Manchester)

version 17.2R1.13;

system {

host-name Manchester;

root-authentication {

encrypted-password "$6$13VJ2t/5$/M3nPPtKZFmn3AeKJw9cx58q6/lOOSF9sYeQQ9FzMTX0f7rVZlnSBZZeMQPAJAna3CNPO5HT358uKPZF

//y530"; ## SECRET-DATA

}

syslog {

user \* {

any emergency;

}

file messages {

any notice;

authorization info;

}

file interactive-commands {

interactive-commands any;

}

}

processes {

dhcp-service {

traceoptions {

file dhcp\_logfile size 10m;

level all;

flag all;

}

}

}

}

chassis {

fpc 0 {

pic 0 {

interface-type ge;

number-of-ports 4;

}

lite-mode;

}

}

interfaces {

ge-0/0/1 {

unit 0 {

family inet {

address 10.217.16.111/24;

}

family inet6 {

address fd00:0:2::1:1/127;

}

}

}

ge-0/0/2 {

unit 0 {

family inet {

address 10.217.16.121/24;

}

family inet6 {

address fd00:0:2::5:0/127;

}

}

}

ge-0/0/3 {

unit 0 {

family inet {

address 10.217.16.131/24;

}

family inet6 {

address fd00:0:2::8:0/127;

}

}

}

fxp0 {

unit 0 {

family inet {

dhcp {

vendor-id Juniper-vmx;

}

}

}

}

lo0 {

unit 0 {

family inet6 {

address fd00:0:2::2/128;

}

}

}

}

snmp {

name "Manchester CP";

description "Manchester CP";

location Manchester;

client-list list0 {

10.217.19.237/24;

}

community public {

authorization read-write;

}

}

routing-options {

router-id 192.168.2.20;

}

protocols {

ospf3 {

area 0.0.0.2 {

interface ge-0/0/1.0;

interface ge-0/0/2.0;

interface ge-0/0/3.0;

interface lo0.0;

}

}

}

# Appendix C(Birmingham)

version 17.2R1.13;

system {

host-name Birmingham;

root-authentication {

encrypted-password "$6$NDfkuEAx$TF952oe1f/IZa7e.asbD/mLBfn3Rh4lYFqbE.UZRSoNPCg8/sPGVdVroc1L4pBWQi/6/yEsCMrhr4KkRRaY3J1"; ## SECRET-DATA

}

syslog {

user \* {

any emergency;

}

file messages {

any notice;

authorization info;

}

file interactive-commands {

interactive-commands any;

}

}

processes {

dhcp-service {

traceoptions {

file dhcp\_logfile size 10m;

level all;

flag all;

}

}

}

}

interfaces {

ge-0/0/1 {

unit 0 {

family inet {

address 10.217.19.12/24;

}

family inet6 {

address fd00:0:2::2:1/127;

}

}

}

ge-0/0/2 {

unit 0 {

family inet {

address 10.217.19.22/24;

}

family inet6 {

address fd00:0:2::5:1/127;

}

}

}

ge-0/0/3 {

unit 0 {

family inet {

address 10.217.19.32/24;

}

family inet6 {

address fd00:0:2::9:0/127;

}

}

}

fxp0 {

unit 0 {

family inet {

dhcp {

vendor-id Juniper-vmx;

}

}

}

}

lo0 {

unit 0 {

family inet6 {

address fd00:0:2::3/128;

}

}

}

}

snmp {

name "Birmingham CP";

description "Birmingham CP";

location Birmingham;

client-list list0 {

19.217.19.237/24;

}

community public {

authorization read-write;

}

}

routing-options {

router-id 192.168.2.30;

}

protocols {

ospf3 {

area 0.0.0.2 {

interface ge-0/0/1.0;

interface ge-0/0/2.0;

interface ge-0/0/3.0;

interface lo0.0;

}

}

}

# Appendix D(Firewall)

version 12.1X47-D15.4;

system {

host-name vSRX\_Firewall;

root-authentication {

encrypted-password "$1$/U/jcNkw$3XHgSIQO6n23YB8megbAO0"; ## SECRET-DATA

}

services {

ssh;

web-management {

http {

interface ge-0/0/0.0;

}

}

}

license {

autoupdate {

url https://ae1.juniper.net/junos/key\_retrieval;

}

}

}

interfaces {

ge-0/0/0 {

unit 0 {

family inet6 {

address fd00:0:2::3:1/127;

}

}

}

ge-0/0/1 {

unit 0 {

family inet6 {

address 2001:0:2::1:1/127;

}

}

}

lo0 {

unit 0 {

family inet6 {

address fd00:0:2::4/128;

}

}

}

}

snmp {

name Firewall;

description Firewall;

location Odense;

client-list list0 {

10.217.19.237/24;

}

community public {

authorization read-write;

}

}

routing-options {

router-id 192.168.2.2;

autonomous-system 65200;

}

protocols {

bgp {

group to\_PE2 {

type external;

local-address 2001:0:2::1:1;

family inet6 {

unicast;

}

export [ send-v6 BGP-to-OSPF ];

peer-as 65000;

neighbor 2001:0:2::1:0;

}

}

ospf3 {

export BGP-to-OSPF;

area 0.0.0.2 {

interface ge-0/0/0.0;

interface ge-0/0/1.0;

interface lo0.0;

}

}

}

policy-options {

policy-statement BGP-to-OSPF {

from {

family inet6;

protocol [ ospf ospf3 bgp ];

}

then accept;

}

policy-statement send-v6 {

from {

family inet6;

protocol direct;

}

then accept;

}

}

security {

forwarding-options {

family {

inet6 {

mode packet-based;

}

mpls {

mode packet-based;

}

}

}

}

# Appendix E(LibreNMS)

LibreNMS is a open-source monitoring tool, great thing to monitor our devices on our network.

With great graphs, a truck-load of notification options, including PushBullet, it’s everything you want from a network tool.

**The following document contains the instructions, how to install Librenms on Ubuntu 16.04 Desktop.**

First of all check the inet address of the local machine:

ifconfig

Then let's install MySQL. (I have done in root.)

apt-get install mariadb-server mariadb-client

systemctl restart mysql

mysql -uroot -p

Then create a database:

CREATE DATABASE librenms CHARACTER SET utf8 COLLATE utf8\_unicode\_ci;

(Hit Enter)

CREATE USER 'librenms'@'localhost' IDENTIFIED BY 'YOURPASSWORD';

(Hit Enter)

GRANT ALL PRIVILEGES ON librenms.\* TO 'librenms'@'localhost';

(Hit Enter)

FLUSH PRIVILEGES;

(Hit Enter)

exit

Open with one text editor /etc/mysql/mariadb.conf.d/50-server.cnf file:

nano /etc/mysql/mariadb.conf.d/50-server.cnf

Within the [mysqld] section add:

innodb\_file\_per\_table=1

sql-mode=""

We are going to configure the web server. (Apache2)

Install and Configure Apache:

apt-get install libapache2-mod-php7.0 php7.0-cli php7.0-mysql php7.0-gd php7.0-snmp php-pear php7.0-curl snmp graphviz php7.0-mcrypt php7.0-json apache2 fping imagemagick whois mtr-tiny nmap python-mysqldb snmpd php-net-ipv4 php-net-ipv6 rrdtool git

Note that we installed php.mod 7.0 because it is one Ubuntu 16.04, The 14.05 can handle php5-version.

Open /etc/php/7.0/apache2/php.ini and search for 'date.timezone' (If you opened with nano can search with "Ctrl + W")

nano /etc/php/7.0/apache2/php.ini

;date.timezone (delete ; and set your time zone. Valid example is: Europe/Copenhagen)

date.timezone = Europe/Copenhagen

Do the same in /etc/php/7.0/cli/php.ini:

nano /etc/php/7.0/cli/php.ini

date.timezone = Europe/Copenhagen

(Important to insert the timezone what your client is configured for.)

Execute the following commands: (Enables php7.0 module, Disable mpm\_event, Considering conflict mpm\_event and mpm preford)

a2enmod php7.0

a2dismod mpm\_event

a2enmod mpm\_prefork

phpenmod mcrypt

Add librenms user:

useradd librenms -d /opt/librenms -M -r

usermod -a -G librenms www-data

Go back to /opt and clone the repository:

cd /opt

git clone https://github.com/librenms/librenms.git librenms

Configure Web interface:

cd /opt/librenms

mkdir rrd logs

chmod 775 rrd

Open /etc/apache2/sites-available/librenms.conf:

nano /etc/apache2/sites-available/librenms.conf

And add the following lines:

<VirtualHost \*:80>

DocumentRoot /opt/librenms/html/

ServerName YOURSERVERNAME

CustomLog /opt/librenms/logs/access\_log combined

ErrorLog /opt/librenms/logs/error\_log

AllowEncodedSlashes NoDecode

<Directory "/opt/librenms/html/">

Require all granted

AllowOverride All

Options FollowSymLinks MultiViews

</Directory>

</VirtualHost>

Execute the following commands: (Enabling site librenms, Enabling module rewrite, Restarting apache2)

a2ensite librenms.conf

a2enmod rewrite

systemctl restart apache2

Disable the default site if this is the only site, what you are planning to host:

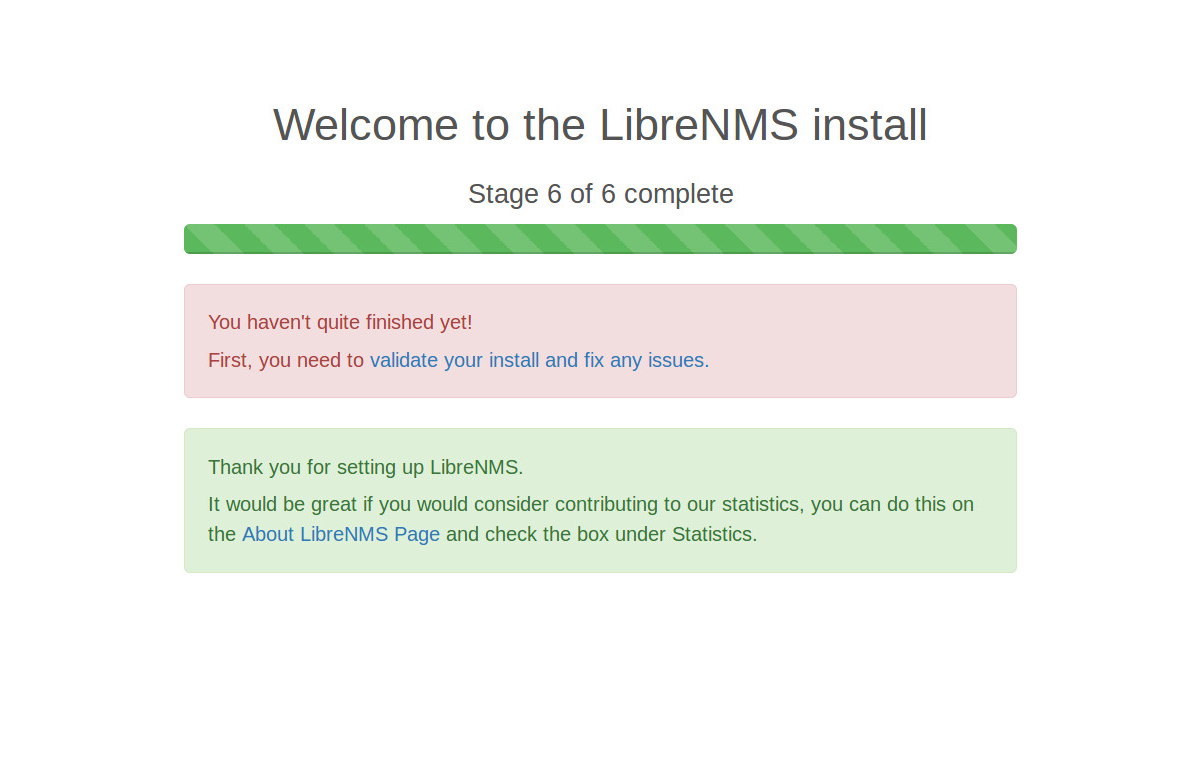
a2dissite 000-default

Web Installer

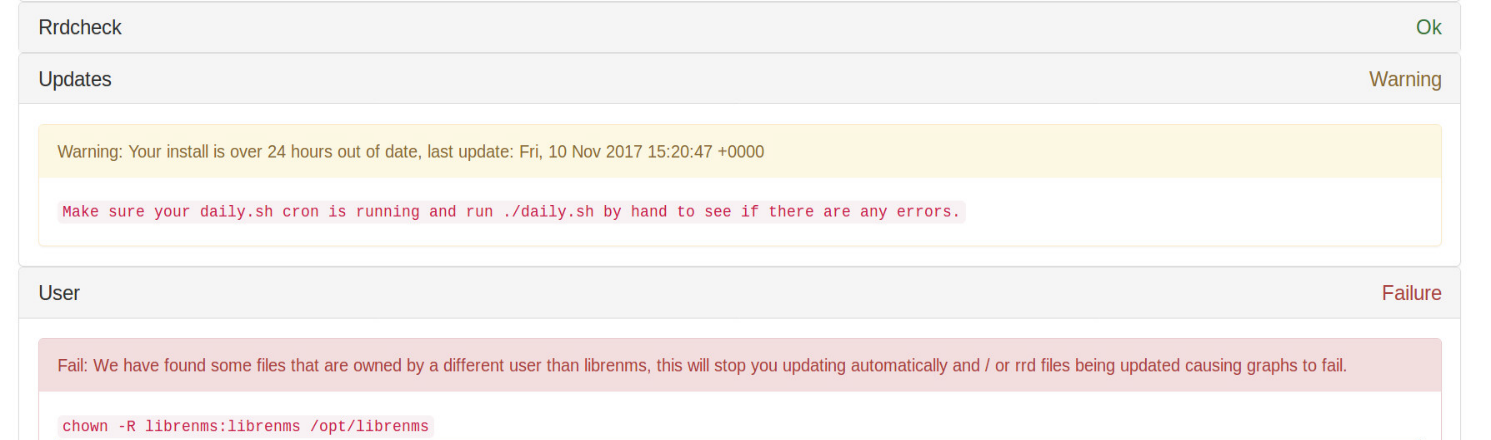
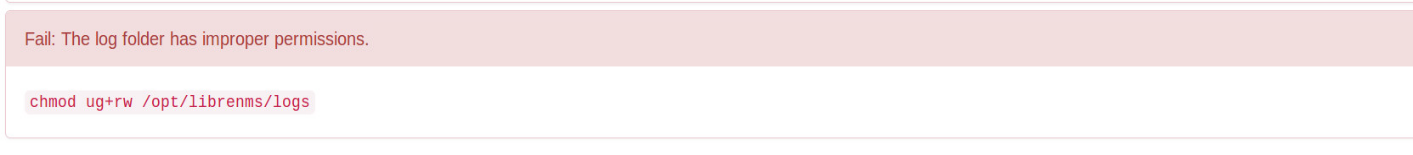
Open: <http://localhost/install.php>

And follow the instructions.

At Stage 5 it’s possible that LibreNMS won’t have access to create a config.php file, therefore we need to do it manually.



After Stage 6 it is possible that there can be some more steps, that Libre require for us, but it’s a smart program and know what it needs. Follow the instructions.



Configure snmpd

Execute the following two commands:

php addhost.php localhost public v2c

php discovery.php -h all

By default, the LibreNMS cronjob runs poller-wrapper.py with 16 threads. The current LibreNMS recommendation is to use 4 threads per core. The default if no thread count is 16 threads.  
If the thread count needs to be changed, edit the cron file. Just add a number after poller-wrapper.py:

/opt/librenms/poller-wrapper.py 12 >> /dev/null 2>&1

cp librenms.nonroot.cron /etc/cron.d/librenms

cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf

Open the configuration file:

nano /etc/snmp/snmpd.conf

And delete RANDOMSTRINGGOESHERE from the second line.

Then execute the following commands: (If curl isn’t installed yet, do it: apt-get install curl)

curl -o /usr/bin/distro <https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp/distro>

chmod +x /usr/bin/distro

systemctl restart snmpd

Make the Cron job:

cp librenms.nonroot.cron /etc/cron.d/librenms

Copy logrotate config.

LibreNMS keeps logs in /opt/librenms/logs. To rotate out the old logs with using the provided logrotate config file:

cp misc/librenms.logrotate /etc/logrotate.d/librenms

Change the ownership on this directory:

chown -R librenms:librenms /opt/librenms

And for last Run validate.php as in the librenms directory:

cd /opt/librenms

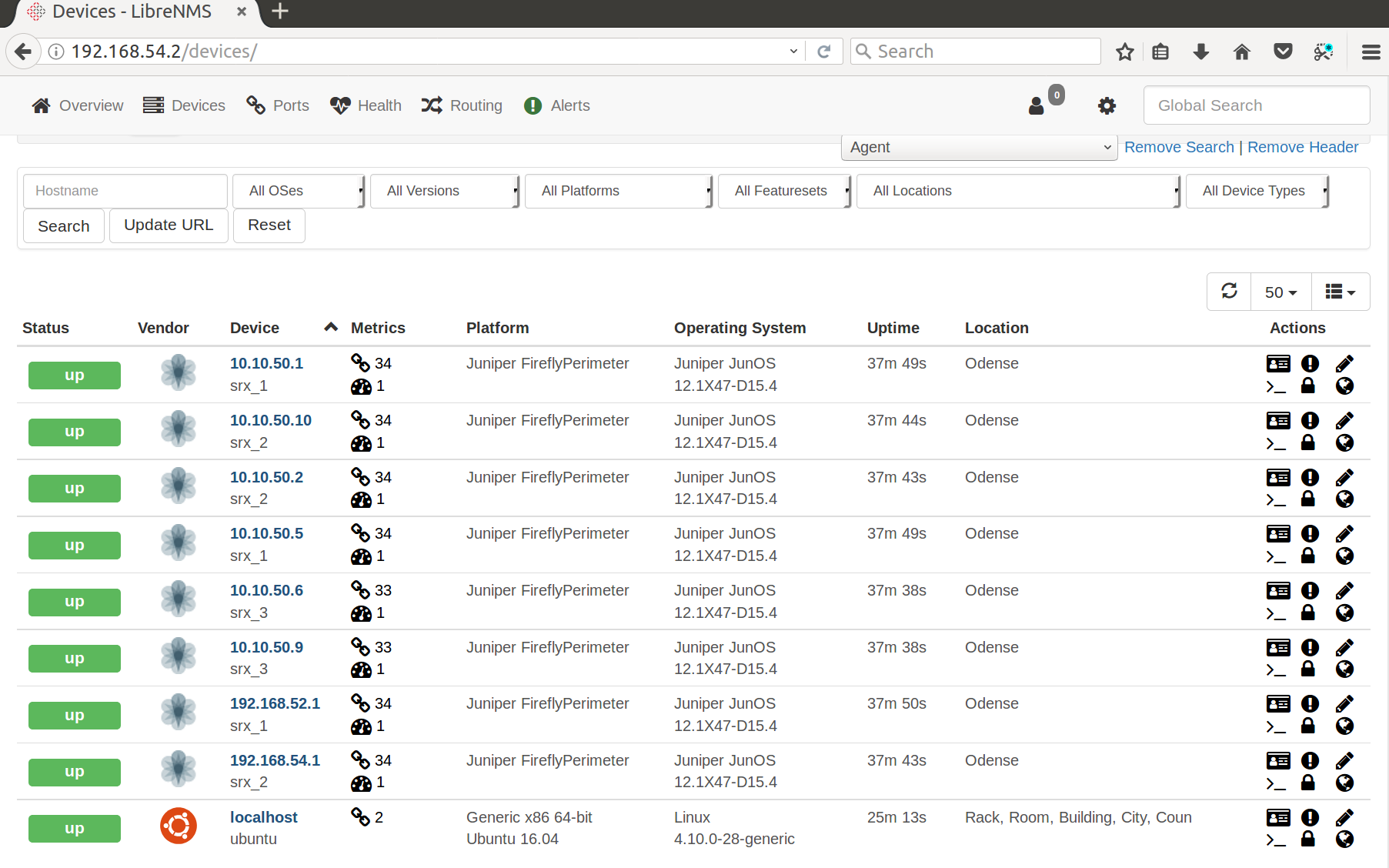
./validate.php

To make sure it works, and saved all changes let's restart some of the programs:

systemctl restart snmpd

systemctl restart apache2

systemctl restart mysql

Final result should be:

*It’s beautiful!*