Monitoring the routers

In this exercise, we will set up LibreNMS as our network monitoring package. The software will poll your routers using SNMP and provide visibility of BGP sessions and other useful information

Throughout these instructions we will use the names and addresses related to **ISP6** on **IXP1**. You should make sure you change any names and addresses in the examples to match your assigned group of machines.

Login to your management PC

```
ssh sysadm@pc1-isp6-ixp1.ws.nsrc.org
```

The password will be given in class.

NOTE: These instructions assume you are the root user. If you are not, prepend `sudo` to the shell commands (the ones that aren't at `mysql>` prompts) or temporarily become a user with root privileges with `sudo -s`.

Install MySQL database

LibreNMS stores its data in a MySQL database. We will assume that the database is running on the same machine as your network management server (this is the most common initial deployment scenario).

```
apt-get install mysql-server mysql-client
```

You will be prompted for a root password for MySQL during the installation. Use the login password you were given in class. Then we can connect to the MySQL server:

```
mysql -uroot -p
```

Input the MySQL root password to enter the MySQL command-line interface where you will get a `mysql>` prompt.

Create the **librenms** database:

```
CREATE DATABASE librenms;
GRANT ALL PRIVILEGES ON librenms.*
  TO 'librenms'@'localhost'
  IDENTIFIED BY 'abc123'
;
FLUSH PRIVILEGES;
exit
```

Here we are using "abc123" as the password for librenms to access MySQL.

NOTE: Don't use a password like this on a production server!

Install the Linux packages

Install the required software:

```
apt-get install libapache2-mod-php5 php5-cli php5-mysql php5-gd php5-snmp
php-pear php5-curl \
        snmp graphviz php5-mcrypt php5-json apache2 fping imagemagick
whois mtr-tiny \
        nmap python-mysqldb snmpd mysql-client php-net-ipv4 php-net-ipv6
rrdtool git
```

The packages listed above are an all-inclusive list of packages that were necessary on a clean install of Ubuntu 14.04.

Install the LibreNMS software

LibreNMS is installed using git. The initial install from github.com is called a `git clone`; subsequent updates are done through `git pull`.

The initial clone can take quite a while (nearly 3 minutes on a 10Mbps connection is typical), so for this workshop we have set up a local mirror for you to use. Run the following:

```
cd /opt
wget http://www.ws.nsrc.org/git/librenms.tgz
tar xzf librenms.tgz
```

Normally the URL to clone would be `git clone https://github.com/librenms/librenms.git`

Set up the web server

To prepare the web interface (and adding devices shortly), you'll need to create and chown a directory as well as create an Apache vhost.

1) Create and chown the `rrd` directory and create the `logs` directory:

```
cd /opt/librenms
mkdir rrd logs
chown www-data:www-data logs/ rrd/
```

2) Create `/etc/apache2/sites-available/librenms.conf` containing the following:

```
<VirtualHost *:80>
  DocumentRoot /opt/librenms/html/
```

```
  ServerName  librenms.ispX-ixpY.ws.nsrc.org
  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>
```

Change **X** and **Y** in `librenms.isp**X**-ixp**Y**.ws.nsrc.org` to correspond to your management PC name. For example:

```
pc1-isp6-ixp1.ws.nsrc.org => librenms.isp6-ixp1.ws.nsrc.org
```

3) On Ubuntu 14.04, mcrypt is not enabled on install. Run the following to enable it:

```
php5enmod mcrypt
```

4) Now enable the vhost and restart Apache

```
a2ensite librenms
a2enmod rewrite
service apache2 restart
```

Configuring the LibreNMS software

We need to configure MySQL password details in the LibreNMS software:

```
  cd /opt/librenms
  cp config.php.default config.php
  editor config.php
```

Change the values to the right of the equal sign for lines beginning with `$config[db_]` to match your database information as setup above.

```
$config['db_host'] = "localhost";
$config['db_user'] = "librenms";
$config['db_pass'] = "abc123";
$config['db_name'] = "librenms";
```

Change the value of `$config['snmp']['community']` from `public` to `NetManage`:

```
$config['snmp']['community'] = "NetManage";
```

We're using ISIS rather than OSPF in our workshop:

```
$config['enable_ospf'] = 0; # Enable OSPF session collection and display
$config['enable_isis'] = 1; # Enable ISIS session collection and display
```

Save the changes!

Initialise the database

Initiate the **librenms** database with the following command:

```
php build-base.php
```

Create admin user

Create the admin user - priv should be 10

```
php adduser.php <name> <pass> 10
```

Substitute your desired username and password – and leave the angled brackets off. We suggest you use **sysadm** and the class password.

Add your routers

LibreNMS uses hostnames when it monitors devices. Normally we'd add our routers to the DNS but for this workshop we'll just add our ISP routers to the /etc/hosts file. Edit /etc/hosts and add entries for your devices. For example, if you have been allocated 100.66.12.0/23, then your entries would look like:

```
100.66.12.1  r1-isp6
100.66.13.1  r2-isp6
```

Save the changes and check that you can ping the routers using these names.

Now we can add the devices to LibreNMS:

```
php addhost.php r1-isp6 NetManage v2c
php addhost.php r2-isp6 NetManage v2c
```

This assumes you haven't made SNMP community changes – if you have, replace `NetManage` with your community. It also assumes SNMP v2c.

Test poller

Discover localhost and poll it for the first time: do this at the command line so you can see if it is working properly.

```
cd /opt/librenms
php discovery.php -h all && php poller.php -h all
```

Create cronjob

Create the cronjob which will run periodic tasks required by librenms

```
cp librenms.cron /etc/cron.d/librenms
```

Daily Updates

LibreNMS performs daily updates by default. At 00:15 system time every day, a `git pull –no-edit –quiet` is performed. You can override this default by editing your `config.php` file. Remove the comment (the `#` mark) on the line:

```
#$config['update'] = 0;
```

so that it looks like this:

```
$config['update'] = 0;
```

Install complete

That's it! You now should be able to log in to <http://librenms.ispX-ixpY.ws.nsrc.org/>. Please note that we have not covered HTTPS setup in this example, so your LibreNMS install is not secure by default. Please do not expose it to the public Internet unless you have configured HTTPS and taken appropriate web server hardening steps.

You should use your instance of LibreNMS to monitor your routers during the workshop - you'll be able to see information about your BGP sessions once they're working

---

NOTE: These instructions are based on the [official LibreNMS installation notes](http://docs.librenms.org/Installation/Installation-(Debian-Ubuntu)/) and have been tested on a fresh install of Ubuntu 14.04.

From:
https://wiki.lpnz.org/ - **Workshops**

Permanent link:
https://wiki.lpnz.org/doku.php?id=2015:pacnog17-ws:track1:librenms

Last update: **2015/07/15 00:45**