

# Installation

## Campus Network Design Workshop

---

You are going to run LibreNMS in docker containers on your “srv2” virtual machine. Once this is complete, it will be reachable under a separate URL for each campus:

- <http://librenms.campusY.ws.nsrc.org/>

## Installing LibreNMS on srv2

*One of the people* in your group should do this task - the others watch.

Open an SSH connection to srv2 in your campus.

The configuration files you need to run the docker containers are already preset. Run the following commands to select the correct directory and then update them:

```
cd /usr/local/src/nmm-docker-compose
sudo git pull
```

Next, you are going to start “traefik” which is a HTTP proxy that allows inbound web connections to be routed to the correct container.

```
cd traefik
docker compose up -d
cd ..
```

You should see a series of files being downloaded, ending up something like this:

```
[+] Running 5/5
  :: reverse-proxy Pulled
3.2s
  :: 9621f1afde84 Pull complete
0.4s
  :: f62b4fe8af5e Pull complete
0.7s
  :: ba654a766834 Pull complete
2.1s
  :: 55d4785f2007 Pull complete
2.2s
[+] Running 2/2
```

```
:: Network traefik_proxy          Created
0.1s
:: Container traefik-reverse-proxy-1 Started
0.5s
```

(Don't forget the `cd ..` step after this!)

Now repeat for LibreNMS:

```
cd librenms
docker compose up -d
cd ..
```

You should see something similar, although with a larger number of downloads and a larger set of containers running.

If you have any problem, talk to your instructor.

## About docker compose

“docker compose” (which was known as “docker-compose” in older versions of docker) reads in a configuration file called `docker-compose.yml`. This tells it how to start up one or more software containers, and connect them together.

The `-d` flag which you gave to `docker compose up` means “detach”: that is, once the containers are started, they will continue to run in the background and you are returned to the shell.

Try the following commands to monitor your docker setup:

```
docker compose ls      # shows all the "docker compose" projects
                        running
docker ps              # shows all the individual docker
                        containers running
docker stats           # shows the memory and CPU use of each
                        container
                        # (continuous stream; use ctrl-C to exit)
```

## Create LibreNMS admin account

Now go to the LibreNMS web interface, which is reachable via the virtual training platform web interface under Campus X > Web > librenms, or as <http://librenms.campusY.ws.nsrc.org> (<http://librenms.campusY.ws.nsrc.org>) for in-person training.

The first time you connect, it will ask you to create an admin account. Enter the following:

- Username: `sysadm`

- Password: <CLASS\_PASSWORD>
- Email: `sysadm@srv1.campusY.ws.nsrc.org` (replace `campusY` with your campus)

Then click the tick, then “Finish install”, then “Dashboard”, which will bring you back to a login screen.

Congratulations, LibreNMS is ready to use! Move onto the next exercise.

## Reference: Installing LibreNMS at home

To install LibreNMS in your own network, then you should follow the instructions in the documentation site (<https://docs.librenms.org/>) .

There are three options:

- Install from source (<https://docs.librenms.org/Installation/Install-LibreNMS/>) . Make sure you don't deploy on any OS other than the versions they show, as you may find problems with incompatible version of PHP and other dependencies.
- Run the docker image (<https://docs.librenms.org/Installation/Docker/>) using `docker compose` . This is what we did in the lab for simplicity.
- Run the VM image (<https://docs.librenms.org/Installation/Images/>) under a hypervisor such as `libvirt/kvm` or `virtualbox`. The image is only updated intermittently, so check the release date, but you should be able to run an update within the VM.

## Tuning LibreNMS

There are some tuning steps (<https://docs.librenms.org/Support/Performance/>) to improve the performance of LibreNMS. These have already been done in the lab.

### Use rrdcached

`rrdcached` allows LibreNMS to write information about devices to memory instead of directly to disk. LibreNMS is a resource-intensive tool and adding memory cache support to LibreNMS will lessen the load on your server.

See <https://docs.librenms.org/Extensions/RRDCached/> (<https://docs.librenms.org/Extensions/RRDCached/>) .

### Tuning Mysql / MariaDB

The following settings can go in `/etc/mysql/mariadb.conf.d/99-librenms.cnf` to override defaults:

```
[mysqld]
innodb_file_per_table=1
lower_case_table_names=0
innodb_flush_log_at_trx_commit = 2
```

Some systems and/or older versions of LibreNMS may also require `sql-mode=""` .

See:

- <https://docs.librenms.org/Installation/Install-LibreNMS/#configure-mariadb>  
(<https://docs.librenms.org/Installation/Install-LibreNMS/#configure-mariadb>)
- <https://docs.librenms.org/Support/Performance/#mysql-optimisation>  
(<https://docs.librenms.org/Support/Performance/#mysql-optimisation>)
- [https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar\\_innodb\\_flush\\_log\\_at\\_trx\\_commit](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_flush_log_at_trx_commit)  
([https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar\\_innodb\\_flush\\_log\\_at\\_trx\\_commit](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_flush_log_at_trx_commit))
- <https://docs.librenms.org/Installation/Installation-CentOS-6-Apache-Nginx/#general>  
(<https://docs.librenms.org/Installation/Installation-CentOS-6-Apache-Nginx/#general>)