opHA Installation and Configuration Guide

⚠ Please refer to opHAv2 instructions

opHA2 Installation and Configuration Guide

🛑 These instructions are deprecated and only for the old version 1.

Installation Prerequisites

- The individual performing this installation has some Linux experience.
- NMIS8 is installed on the same server where opHA will be installed
- NMIS8 is installed in /usr/local/nmis8
- opHA will be installed into /usr/local/nmis8
- Root access is available (not always needed but much easier)
- **Perl 5.10**
- **RRDtool 1.4.7**
- **NMIS 8.3.24G or later**
- **opHA will be installed onto the Primary and each Poller NMIS server**

Installation Steps

Install CPAN Libraries

NMIS optionally uses a UUID for devices, this is enabled by default with using opHA, the Perl CPAN library Data::UUID will be required, this is described in the article Using Unique Identifiers (UUID) for NMIS Nodes. You can install this using CPAN

```
cpan
install Data::UUID
```

Install opHA

This step will be repeated for each NMIS Primary and poller server

- Copy the opHA tarball to the poller or Primary NMIS server (a tarball is a GZIP'd tar file, e.g. opHA-1.1.tar.gz)
    1. You may need to use SCP or FTP to get the file onto the server.
- The file will now likely be in the users home directory.
- If the installation directory does not already exist
- Change into the directory where the tarball was copied
- Untar the file

```
ophaVersion=1.4.1
#(not required for Debian/Ubuntu)
unalias cp
cd ~
tar xvf ~/opHA-$ophaVersion.tar.gz
cd opHA-$ophaVersion/
cp admin/convert_nmis_db.pl /usr/local/nmis8/admin
cp cgi-bin/connect.pl /usr/local/nmis8/cgi-bin
cp lib/NMIS/Connect.pm /usr/local/nmis8/lib/NMIS

cp install/Access.opha.nmis /usr/local/nmis8/install
cp install/Tables.opha.nmis /usr/local/nmis8/install
cp install/Table-Nodes.opha.nmis /usr/local/nmis8/install

cp install/BusinessServices.nmis /usr/local/nmis8/install
cp install/Servers.nmis /usr/local/nmis8/install
cp install/Customers.nmis /usr/local/nmis8/install
cp install/ServiceStatus.nmis /usr/local/nmis8/install
cp install/Tenants.nmis /usr/local/nmis8/install
cp install/Table-* /usr/local/nmis8/install

# Merge your current Access configuration with the opHA access configuration
/usr/local/nmis8/admin/updateconfig.pl /usr/local/nmis8/install/Access.opha.nmis /usr/local/nmis8/conf/Access.
nmis
/usr/local/nmis8/admin/updateconfig.pl /usr/local/nmis8/install/Tables.opha.nmis /usr/local/nmis8/conf/Tables.
nmis

/usr/local/nmis8/admin/fixperms.pl
```

Optional Step - Install or Patch

If this is a fresh installation, copy the following files as samples, if this is an existing installation and you are upgrading, you do not need to do this step.

```
cp install/Table-* /usr/local/nmis8/conf
cp install/Table-Nodes.opha.nmis /usr/local/nmis8/conf/Table-Nodes.nmis
cp install/BusinessServices.nmis /usr/local/nmis8/conf
cp install/Servers.nmis /usr/local/nmis8/conf
cp install/Customers.nmis /usr/local/nmis8/conf
cp install/ServiceStatus.nmis /usr/local/nmis8/conf
cp install/Tenants.nmis /usr/local/nmis8/conf
```

Using JSON for NMIS Database

Supported in opHA is having NMIS use JSON for its database, this will require NMIS 8.4.8g or greater. This should be enabled on all servers running in an opHA cluster. The following needs to be run on every Primary and poller server in a cluster and this should be co-ordinated to run very close together.

```
/usr/local/nmis8/admin/convert_nmis_db.pl
```

This script will stop NMIS polling, convert the database files, update the NMIS configuration to use the new database format, then start the polling again.

opHA Authentication Model

opHA has a simple yet strong authentication model, to prevent unwanted access to NMIS data.

The poller is configured with:

- An NMIS user and password, by default this is an Apache htpasswd file, defined in /usr/local/nmis8/conf/users.dat
- An NMIS user, with associated privileges, defined in /usr/local/nmis8/conf/Users.nmis
- An NMIS user to use for the authentication policy enforcement, defined in /usr/local/nmis8/conf/Config.nmis
- Server Community, which the server must use to request data.

The Primary is configured with (for each poller):

- An NMIS user and password, which needs to match the poller configuration
- A poller/server community, which needs to match the poller configuration.

This model enables you to use separate credentials for each poller or the same credentials for each poller, providing for simple configuration, and more secure configuration if required.

All communications between Primary and poller can be done over SSL if required, this is supported by configuring your server HTTPD to support SSL and then configuring the Primary, poller communications to use HTTPS.

opHA Poller Configuration

This configuration will be done on each NMIS Poller Server. By default, the shared community for a poller is "secret" if you want to change this to something specific you can edit the NMIS Configuration item "poller_community" using your favourite text editor, edit this line and change secret to your desired opHA community string.

```
'server_community' => 'secret',
```

Verify that the Apache user has been configured for Primary functions. The default userid is "nmismst" and the file /usr/local/nmis8/conf/users.dat should include an entry like

```
nmismst:vnnFthCKoHsps
```

opHA Primary Configuration

Server Name for opHA

Server names need to be lower case with no spaces, e.g. NMIS_Server24 is bad, nmis_server24 is good.

Adding Pollers to Servers.nmis

Once the pollers have been setup, you can configure the Primary with each of its pollerss. This is done by editing the file /usr/local/nmis8/conf/Servers. nmis, and adding a section for each server.

The default entries look like this:

```
'nmis1' => {
   'community' => 'secret',
   'name' => 'nmis1',
   'config' => 'Config.nmis',
   'protocol' => 'https',
   'port' => '443',
   'host' => 'nmis1.domain.com',
   'portal_protocol' => 'http',
   'portal_port' => '80',
   'portal_host' => 'nmis1.alternate.com',
   'cgi_url_base' => '/cgi-nmis8',
   'url_base' => '/nmis8',
   'user' => 'nmismst',
   'passwd' => 'C00kb00k'
},
'nmis2' => {
  'community' => 'secret',
  'name' => 'nmis2',
  'config' => 'Config.nmis',
  'protocol' => 'http',
  'port' => '80',
  'host' => '192.168.1.42',
  'portal_protocol' => 'http',
  'portal_port' => '80',
  'portal_host' => 'nmis2',
  'cgi_url_base' => '/cgi-nmis8',
  'url_base' => '/nmis8',
  'user' => 'nmismst',
  'passwd' => 'C00kb00k'
}
```

Edit the entry to look like this, in this example the hostname of the poller is "vali":

```
'vali' => {
  'community' => 'YOURNAMEHERE',
  'name' => 'vali',
  'config' => 'Config.nmis',
  'protocol' => 'http',
  'port' => '80',
  'host' => 'vali',
  'portal_protocol' => 'http',
  'portal_port' => '80',
  'portal_host' => 'vali',
  'cgi_url_base' => '/cgi-nmis8',
  'url_base' => '/nmis8',
  'user' => 'nmismst',
  'passwd' => 'C00kb00k'
}
```

There are many options in this configuration but unless you are wanting to change the defaults considerably most of them will not matter. If you wanted to use HTTPS to connect between the Primary and the poller, you could use https as the protocol and update the port accordingly. You can use different user and passwd permissions here.

If you were presenting the Poller and needed to use an alternate connection, e.g. through a reverse proxy for presenting a portal, you would modify the portal_protocol, portal_port and portal_host accordingly.

Promoting NMIS to be a Primary

By default, an NMIS server operates in standalone mode (which is also poller mode), to have NMIS behave in a Primary fashion, you will need to modify the configuration, so you can edit the NMIS Configuration item "server_Primary" using your favourite text editor, edit this line and change from "false" to "true".

```
'server_master' => 'true',
```

Adding Poller Groups to Primary

On each poller you will need to determine which groups are currently in use.

```
[root@vali conf]# grep group_list /usr/local/nmis8/conf/Config.nmis
  'group_list' => 'HQ,HQDev',
```

This will result in a list of groups which need to be added to the NMIS Primary, edit /usr/local/nmis8/conf/Config.nmis and add these groups to that list, this is a comma separated list.

```
'group_list' => 'NMIS8,DataCenter,Branches,Sales,Campus,HeadOffice,HQ,HQDev',
```

You can also use the admin script /usr/local/nmis8/admin/grouplist.pl on the Primary to find and patch all groups used by all devices collected from the pollers.

Once opHA has successfully pulled/pushed the devices from poller to Primary you can analyse and patch the groups list by using the following.

```
# Simply list all found groups so you can add them to 'group_list' => '...'   as above
[root@opmantek ~]# /usr/local/nmis8/admin/grouplist.pl
Branches
DataCenter
IOSXR
NMIS8
The following is the list of groups for the NMIS Config file Config.nmis
'group_list' => 'Branches,DataCenter,NMIS8,IOSXR',
## You can then simply copy this last line to replace the curren line in Config.nmis

### Alternatively the script can automatically update the Config.nmis file's 'group_list' entry for you using
the patch=true argument as follows:
/usr/local/nmis8/admin/grouplist.pl patch=true
```

Limiting Primary Group Collection

opHA supports Multi-Primary, that means you can have several Primarys collecting information from the same pollers if required. This could be especially useful if you wanted to have one Primary with all groups on a poller, and another Primary with different groups from different pollers, effectively sharing some information between groups.

To do this you use the group property in the Servers.nmis file. Edit the file and add the group property in and a regular expression in for the groups, this will take the form

```
'group' => 'Brisbane|Boston|Saratoga',
```

This will match all groups contain the sub-strings, Brisbane, Boston or Saratoga. A complete server entry would look like this.

```
'demo' => {
  'community' => 'secret',
  'name' => 'demo',
  'config' => 'Config',
  'protocol' => 'http',
  'port' => '80',
  'host' => '192.168.1.42',
  'group' => 'Brisbane|Boston|Saratoga',
  'portal_protocol' => 'http',
  'portal_port' => '80',
  'portal_host' => 'demo.dev.opmantek.com',
  'cgi_url_base' => '/cgi-nmis8',
  'url_base' => '/nmis8',
  'user' => 'nmismst',
  'passwd' => 'C00kb00k'
},
```

Test Primary Collection

You can verify if the Primary is collecting data from the pollers by running this command

```
[root@thor conf]# /usr/local/nmis8/bin/nmis.pl type=master debug=true sleep=1
NMIS Copyright (C) 1999-2011 Opmantek Limited (www.opmantek.com)
This program comes with ABSOLUTELY NO WARRANTY;
This is free software licensed under GNU GPL, and you are welcome to
redistribute it under certain conditions; see www.opmantek.com or email
contact@opmantek.com
NMIS version 8.3.4G

14:00:33 nmisMaster, Running NMIS Master Functions
14:00:33 nmisMaster, Master, processing Poller Server vali
14:00:33 nmisMaster, Get loadnodedetails from vali
14:00:33 nmisMaster, Get sumnodetable from vali
14:00:34 nmisMaster, get summary8 from vali
14:00:34 nmisMaster, get summary16 from vali
[root@thor conf]#
```

Server Priority

To handle devices being managed by more than one server with some determinism, there is a new feature in opHA 1.4 for server priority. By default a Primary server is priority 10 and a poller is priority 5, if you have two pollers managing the same nodes and you want poller1 to be used as the primary source of information, set the server priority in the Servers.nmis file to be higher than on poller2, or conversely lower the priority on poller2.

```
'nmis1' => {
 'name' => 'nmis1',
 'config' => 'Config',
 'protocol' => 'http',
 'port' => '3000',
 'host' => 'nmis1.domain.com',
 'portal_protocol' => 'http',
 'portal_port' => '80',
 'portal_host' => 'nmis1.alternate.com',
 'server_priority' => '6',
 'cgi_url_base' => '/cgi-nmis8',
 'url_base' => '/nmis8',
 'user' => 'nmismst',
 'passwd' => 'C00kb00k'
},
'nmis2' => {
 'name' => 'nmis2',
 'config' => 'Config',
 'protocol' => 'http',
 'port' => '3000',
 'host' => '192.168.1.42',
 'portal_protocol' => 'http',
 'portal_port' => '80',
 'portal_host' => 'nmis2',
 'server_priority' => '4',
 'cgi_url_base' => '/cgi-nmis8',
 'url_base' => '/nmis8',
 'user' => 'nmismst',
 'passwd' => 'C00kb00k'
}
```

This works with the Primary as well, with the Primary server being a higher priority by default. The Primary priority is set with the NMIS configuration option, master_server_priority and is 10 by default.

```
 'master_server_priority' => 10,
```

Running a Primary Collection

You can optionally have the NMIS polling cycle do the Primary collection, or you can run it separately from Cron. If you want to have it separate which is a good option, change the following NMIS configuration item nmis_master_poll_cycle to be false in the file /usr/local/nmis8/conf/Config.nmis:

```
'nmis_master_poll_cycle' => 'false',
```

Then add this line to the crontab which runs your nmis collections.

```
*/2 * * * * /usr/local/nmis8/bin/nmis.pl type=master
```

This will get your collections running every 2 minutes regardless of other polling. There is also an option called master_sleep which is so that your type=update and type=master can run every 1 minute and still have data, the default offset is 15.

Conclusion

After refreshing the web pages on the NMIS Primary server you will see the data from the pollers.