Upgrading from NMIS8 compatible applications to NMIS9 compatible applications (V2 - Detailed)

Summary

In this guide we will migrate the data from NMIS8 and OMK Applications to nmis9 and OMK Applications.

- Server: uburnto (ubuntu 18)   deb-n-burn (debian 10)
- nmis 8.7.2 nmis 9.2.2
- opCharts 2.5.1 opCharts 3
- opConfig 3.5.2 opConfig 4
- opEvents 2.6.3 opEvents 3

- opHA 2.2.2   opHA 3
- opReports 3.4.2 opReports 4

Please note that, to make use of some utility versions for the migration, at least these versions are needed:

- opCharts 4.2.5
- opConfig 4.2.4
- opEvents 4.0.2 (opEvents 3.2.1 just for the migration scripts specific for opEvents).
- opHA 3.3.1
- opReports 4.2.2

Options

1. Use a VM for the nmis9 server: We would need to run additional steps to migrate the data.
2. Install the apps once the data is migrated: The installers will upgrade the data to the new format.

Upgrade process: Using nmis9 VM (NMIS9)

Prerequisites

- NMIS 9 and OMK apps are installed on the server.

Step 1. Stop all daemons

We first need to make sure all the daemons are stopped in the destination server:

```
sudo service nmis9d stop
sudo service omkd stop
sudo service opeventsd stop
sudo service opchartsd stop
sudo service opconfigd stop
```

Step 2. Move and adapt NMIS files

There are a couple of options to move configuration files:

- (Option 1): Adapt just needed configuration files in the destination server (Cleaner option).
- (Option 2): Move all configurations from origin server and adapt in the new server.

Configuration Files: Option 1

Copy

Copy - just - the configuration files customised from the source server:

```
rsync -r root@myserver.com:/usr/local/nmis8/conf/Config.nmis /usr/local/nmis9/conf
rsync -r root@myserver.com:/usr/local/nmis8/conf/users.dat /usr/local/nmis9/conf
...
```

We can move all the files except the Table-tablename.nmis.

- Import the old nmis8 configuration and review important configuration options

Configure

We need to make some adjustments to the Config file.

- Review important parameters, as the number of workers (workers parameter in config). We can use this rule to get a starter number, but this will depend on the server resources, the amount of data collected by the node, etc. If the server is too slow we would need to decrease the number. If there are nodes with more than 3x late collects (Check the polling summary), the number should be increased:

```
(Number of nodes x AVG Collect time) / 300
```

300 _ Default polling time.

- There is a script to use in opCommon (Just from the last software versions) that compares configuration items from a configuration file missing from another.

```
/usr/local/omk/bin/opcommon-cli.pl act=compare_configs nmis8=/tmp/nmis8.nmis nmis9=/tmp/nmis9.nmis
```

This will compare the keys. But if we want to compare the values (To compare, the default configuration file and the customised, to see want has changed):

```
/usr/local/omk/bin/opcommon-cli.pl act=compare_custom default=/usr/local/nmis9/conf-default/Config.nmis custom=
/usr/local/nmis9/conf/Config.nmis
```

- Then, fix files permissions:

```
/usr/local/nmis9/bin/nmis-cli act=fixperms
```

Import NMIS 8 configuration Items

More information here

Review important configuration Items

More information here

Configuration Files: Option 2

Copy

Copy the configuration directory from the source server.

Create a conf backup first:

```
mv /usr/local/nmis9/conf /tmp/nmis9/conf
```

This can be done really easy with rsync:

```
rsync -r root@myserver.com:/usr/local/nmis8/conf /usr/local/nmis9
```

Configure

We need to make some adjustments to the file. This is:

- Replace all nmis8 occurrences by nmis9.
- Replace 'sql' section for 'database'.
- Add database section.

```
  'database' => {
    'db_name' => 'nmisng',
    'db_password' => 'op42flow42',
    'db_port' => '27017',
    'db_query_timeout' => 5000,
    'db_server' => 'localhost',
    'db_username' => 'opUserRW',
    'db_never_remove_indices' => [ 'nodes' ],
  },
...
 '<nmis_base>' => '/usr/local/nmis9',
 '<url_base>' => '/nmis9',
 '<cgi_url_base>' => '/cgi-nmis9',
..
                'nmisd_max_workers' => 10, # set to zero to disable multiprocessing
                'nmisd_worker_cycle' => 100,        # workers sleep no longer than X between cycles
                'nmisd_scheduler_cycle' => 10, # the supervisor daemon sleeps no longer than X between cycles

                'nmisd_fping_worker' => 'true', # if true, at dedicated worker for fping use used
                'postpone_clashing_schedule' => 30, # if positive, clashing jobs are retried N seconds later, 
otherwise clashing jobs are discarded
```

- Review important parameters, as the number of workers (workers parameter in config). We can use this rule to get a starter number, but this will depend on the server resources, the amount of data collected by the node, etc. If the server is too slow we would need to decrease the number. If there are nodes with more than 3x late collects (Check the polling summary), the number should be increased:

```
(Number of nodes x AVG Collect time) / 300
```

300 _ Default polling time.

- There is a script to use in opCommon (Just from the last software versions) that compares configuration items from a configuration file missing from another.

```
/usr/local/omk/bin/opcommon-cli.pl act=compare_configs nmis8=/tmp/nmis8.nmis nmis9=/tmp/nmis9.nmis
```

- Copy tables from conf-default:

```
/usr/local/nmis9 cp conf-default/Table-* conf
```

- Then, fix files permissions:

```
/usr/local/nmis9/bin/nmis-cli act=fixperms
```

Check

Even with the daemon stop, we can check if the configuration is working. We can access the GUI:

http://server/cgi-nmis9/nmiscgi.pl

If we see any 500 Error, we can review the apache logs, usually in the path /var/log/apache2/error.log

Model Files

We can move all the model customisations in the models-custom directory. We can use rsync for this again.

NOTE Some models may need to be adjusted. Specially if they are using nmis internal functions.

Plugins

We have moved the conf/plugins directory using rsync. But the custom plugins need to be adapted to NMIS9.

Here you can find further information.

Step 3. Copy RRDs

This can be done really easy with rsync:

```
rsync -r root@myserver.com:/usr/local/nmis8/database/nodes /usr/local/nmis9/database
```

Then fix permissions:

```
/usr/local/nmis9/bin/nmis-cli act=fixperms
```

Run script to adapt names

This script will adapt nmis8 rrd names to nmis9 (From nmis 9.2.1):

```
/usr/local/nmis9/admin/node_admin.pl act=move-nmis8-rrd-files {node=nodeX|ALL|uuid=nodeUUID} [remove_old=1]
[force=1]
```

Step 4. Import Nodes

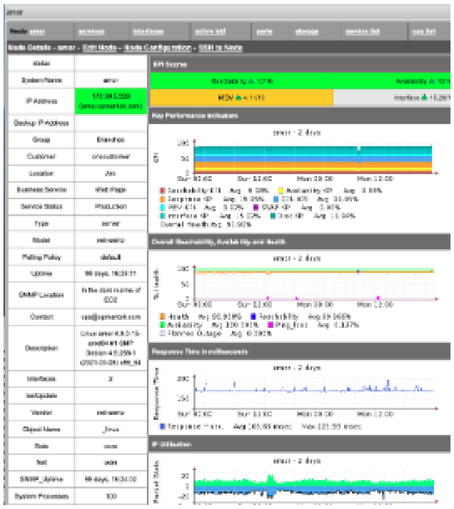
We have copied the nodes file from nmis8 using rsync. We would need to import the nodes now:

```
/usr/local/nmis9/admin/node_admin.pl act=import_bulk nodes=/usr/local/nmis9/conf/Nodes.nmis debug=true
```

Step 5. Verify

We can start nmis9d. Depending on the number of nodes, we should leave it running for more or less time.

We can open nmis9 and you should see the nodes with the rrd information:



Upgrade process: Using nmis9 VM (OMK)

Prerequisites

- NMIS 9 and OMK apps are installed on the server.
- All the services had been stoped

- **Important**: NMIS9 should have being updated, and the nodes have been migrated and being collected/updated. This is because, the migration scripts will need to use some data from the nodes to operate.

Step 1. Move and adapt OMK files

There are a couple of options to move configuration files:

- (Option 1): Adapt just needed configuration files in the destination server (Cleaner option).
- (Option 2): Move all configurations from origin server and adapt in the new server.

Configuration Files: Option 1

Copy

Copy - just - the configuration files customised from the source server:

```
rsync -r root@myserver.com:/usr/local/omk/conf/opCommon /usr/local/omk/conf
...
```

And all the customised files/folders (If needed).

opConfig:

```
command_sets.d
compliance_policies
config_parsers
config_plugins
config_sets.d
phrasebooks
schedule
```

opEvents:

```
EventActions.nmis
EventDB.nmis
EventEmails.nmis
EventListRules.nmis
EventNmisRules.nmis
EventParserRules.nmis
EventRules.nmis
EventSyslogRules.nmis
EventTivoliRules.nmis
EventTrapRules.nmis
parser_plugins
```

And users.dat file:

```
ln -s ../nmis9/conf/users.dat conf/users.dat
```

Convert to JSON

It can be done with the following command:

```
/usr/local/omk# /usr/local/omk/bin/opcommon-cli.exe act=convert_json_dir dir=conf
```

Configure

Update opCommon.json checking the customised item. You can use an useful script for that:

```
/usr/local/omk/bin/opcommon-cli.pl act=compare_custom default=/usr/local/omk/install/opCommon.json custom=/usr
/local/omk/conf/opCommon.json
```

Configuration Files: Option 2

Copy

Copy the configuration directory from the source server.

Create a conf backup first:

```
mv /usr/local/omk/conf /tmp/omk/conf
```

This can be done really easy with rsync:

```
rsync -r root@myserver.com:/usr/local/omk/conf /usr/local/omk
```

Convert to JSON

It can be done with the following command:

```
/usr/local/omk# /usr/local/omk/bin/opcommon-cli.exe act=convert_json_dir dir=conf
```

Configure

Update opCommon.json:

```
"database": {
   "db_name": "omk_shared",
   ...
    "nmis": {
        "<nmis9_dir>": "/usr/local/nmis9",
        "<nmis9_logs>": "/usr/local/nmis9/logs",
        "nmis9_cgi_url_base": "/cgi-nmis9",
        "nmis9_dir": "/usr/local/nmis9",
```

Create symbolic link to nmis9 users file:

```
ln -s ../nmis9/conf/users.dat conf/users.dat
```

opEvents:

- Some node properties have been updated. This mean the event action rules should be updated regarding this document.
- Parser plugins also needs to be update.

Step 2. Move OMK Database

- Stop services in origin server
- Generate mongodump:

```
mkdir /data/mongodump
mongodump -u=opUserRW -p=op42flow42 --out=/data/mongodump
```

- copy the data with rsync:

```
rsync -r root@myserver.com:/data/mongodump /data
```

- Restore:

```
mongorestore -u opUserRW -p op42flow42 /data/mongodump/
```

Step 3. OMK Applications considerations

opCharts ✓

opCharts uses the same database.

Run the cli option to restore indexes:

```
/usr/local/omk/bin/opcharts-cli.exe act=setup-db debug=1
```

In order to upgrade the information about monitored services, we would need to also run the following:

```
rsync -r root@myserver.com:/usr/local/nmis8/var/service_status /tmp
/usr/local/omk/bin/opcharts-cli.exe act=migrate-monitored-services dir=/tmp/service_status debug=1
```

NOTE In order for this to work, nmis9 should have run and created the inventory with the monitored services there.

opEvents ✓

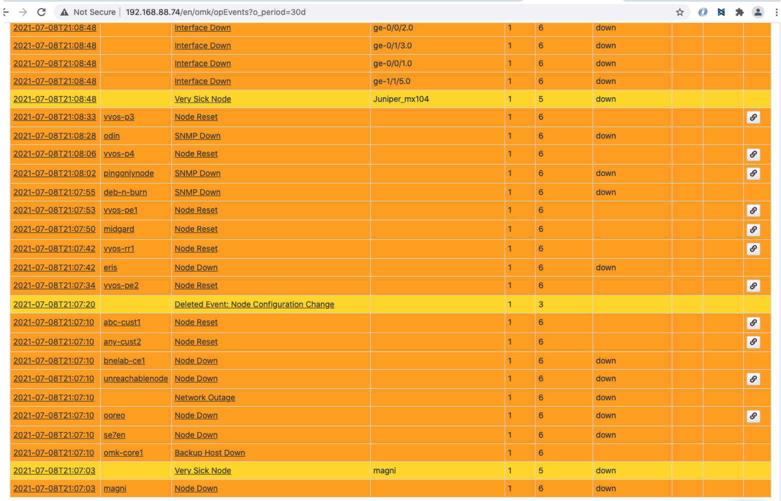
Update database name in opCommon.json:

```
"opevents_db_name": "nmis",
```

Run migration scripts:

```
/usr/local/omk/bin/opevents-cli.exe act=migrate-nodes debug=8
/usr/local/omk/bin/opevents-cli.exe act=migrate debug=8
```

Then, you should be able to see all the events:



2021-07-08T21:08:48		Interface Down	ge-0/0/2.0	1	6	down			
2021-07-08T21:08:48		Interface Down	ge-0/1/3.0	1	6	down			
2021-07-08T21:08:48		Interface Down	ge-0/0/1.0	1	6	down			
2021-07-08T21:08:48		Interface Down	ge-1/1/5.0	1	6	down			
2021-07-08T21:08:48		Very Sick Node	Juniper_mx104	1	5	down			
2021-07-08T21:08:33	vyos-p3	Node Reset		1	6				🔗
2021-07-08T21:08:28	odin	SNMP Down		1	6	down			
2021-07-08T21:08:06	vyos-p4	Node Reset		1	6				🔗
2021-07-08T21:08:02	pingonlynode	SNMP Down		1	6	down			🔗
2021-07-08T21:07:55	deb-n-burn	SNMP Down		1	6	down			
2021-07-08T21:07:53	vyos-pe1	Node Reset		1	6				🔗
2021-07-08T21:07:50	midgard	Node Reset		1	6				🔗
2021-07-08T21:07:42	vyos-rr1	Node Reset		1	6				🔗
2021-07-08T21:07:42	eris	Node Down		1	6	down			
2021-07-08T21:07:34	vyos-pe2	Node Reset		1	6				🔗
2021-07-08T21:07:20		Deleted Event: Node Configuration Change		1	3				
2021-07-08T21:07:10	abc-cust1	Node Reset		1	6				🔗
2021-07-08T21:07:10	any-cust2	Node Reset		1	6				🔗
2021-07-08T21:07:10	bnelab-ce1	Node Down		1	6	down			
2021-07-08T21:07:10	unreachablenode	Node Down		1	6	down			🔗
2021-07-08T21:07:10		Network Outage		1	6	down			
2021-07-08T21:07:10	ooreo	Node Down		1	6	down			🔗
2021-07-08T21:07:10	se7en	Node Down		1	6	down			
2021-07-08T21:07:10	omk-core1	Backup Host Down		1	6				
2021-07-08T21:07:03		Very Sick Node	magni	1	5	down			
2021-07-08T21:07:03	magni	Node Down		1	6	down			

Run ensure indexes:

```
/usr/local/omk/bin/opevents-cli.pl act=setup-db debug=1
```

opConfig ✓

Update database name in opCommon.json:

```
"opconfig_db_name": "nmis",
```

Also, to migrate activation, os_info and connection info information, we would need to run an additional script:

```
/usr/local/omk/bin/opconfig-cli.pl act=migrate-nodes debug=9
```

Run ensure indexes in the db:

```
/usr/local/omk/bin/opconfig-cli.pl  act=setup-db debug=1
```

opReports ✓

NMIS8 opReports and NMIS9 opReports work the same way:

- Report Schedules are stored in /usr/local/conf/schedule
- Reports caching is stored in /usr/local/omk/var/opreports
- Reports are stored in /data/omk/var/reports (unless otherwise specified by the user)

If you followed the instructions above, your Report Schedules are back in place, as you've copied over the entire omk-old/conf directory in step five. The /data/omk/var reports directory should still be where it was at the beginning of the upgrade, as it was never changed.

We can copy the opreports and reports directories into the new application to move all the reports:

```
rsync -r root@myserver.com:/usr/local/omk/var/reports var/
rsync -r root@myserver.com:/usr/local/omk/var/opreports var/
```

opHA ✓

opHA 3 works different from opHA 2. The peers would need to be imported manually using the GUI or the cli, using nmis8/conf/Servers.nmis file.

Further information can be found here.

Run ensure indexes in the db:

```
/usr/local/omk/bin/opconfig-cli.pl  act=setup-db debug=1
```

Step 8. Start daemons

```
sudo service nmis9d start
sudo service omkd start
sudo service opeventsd start
sudo service opchartsd start
sudo service opconfigd start
```

Upgrade process (II): Installing apps (NMIS9)

Prerequisites

- The destination server is a brand new server.
- We would need the last installers from all the products in the /tmp directory.

Step 1. sync nmis8 folder

We can use rsync for this:

```
rsync -r root@myserver.com:/usr/local/nmis8 /usr/local
```

Step 2. run the nmis 9 installer

We can follow the guide for this.

The installer will import the nodes from nmis8 automatically.

Step 3. stop nmis9d

When done, stop the nmis9 service. Let's do other changes before keeps running.

```
sudo service nmis9d stop
```

Step 4. Check imported nodes

Some of the nodes may not be imported, if you see that message in the logs:

```
OUTPUT: Error saving node teide: given roleType 'BGP' is not a known type: 'default,core,distribution,access'
```

Please make sure to copy all this configuration items from nmis8 to nmis9:

```
"roletype_list"
"nettype_list"
"nodetype_list"
```

And run the script again:

```
/usr/local/nmis9/admin/node_admin.pl act=import_bulk nodes=/usr/local/nmis8/conf/Nodes.nmis info=true
```

Step 5. Configuration

Configuration Files

Configure

We need to make some adjustments to the Config file.

- Review important parameters, as the number of workers (workers parameter in config). We can use this rule to get a starter number, but this will depend on the server resources, the amount of data collected by the node, etc. If the server is too slow we would need to decrease the number. If there are nodes with more than 3x late collects (Check the polling summary), the number should be increased:

```
(Number of nodes x AVG Collect time) / 300
```

300 _ Default polling time.

- There is a script to use in opCommon (Just from the last software versions) that compares configuration items from a configuration file missing from another.

```
/usr/local/omk/bin/opcommon-cli.pl act=compare_configs nmis8=/tmp/nmis8.nmis nmis9=/tmp/nmis9.nmis
```

This will compare the keys. But if we want to compare the values (To compare, the default configuration file and the customised, to see want has changed):

```
/usr/local/omk/bin/opcommon-cli.pl act=compare_custom default=/usr/local/nmis9/conf-default/Config.nmis custom=
/usr/local/nmis9/conf/Config.nmis
```

- Then, fix file permissions:

```
/usr/local/nmis9/bin/nmis-cli act=fixperms
```

Model Files

We can move all the model customisations in the models-custom directory. We can use rsync for this again.

NOTE Some models may need to be adjusted. Specially if they are using nmis internal functions.

Plugins

We have moved the conf/plugins directory using rsync. But the custom plugins need to be adapted to NMIS9.

Here you can find further information.

Step 6. Copy RRDs

Link to Step 3 Copy RRDs above

Step 7: Verify

Link to Step 5 Verify above

Upgrade process (II): Installing apps (OMK)

Prerequisites

- The destination server is a brand new server with nmis9 installed.
- nmis9 has been running for a couple of cycles to create the inventory. Then, it should be stopped.
- We would need the last installers from all the products in the /tmp directory.

Step 1. Installers running

Run the installers for each application. All the data should be automatically migrated.

Step 2. Customise configuration

Same steps as the other procedure.

Step 3. opConfig data

Run the migration script:

```
/usr/local/omk/bin/opconfig-cli.pl act=migrate-nodes debug=9
```

Step 4. opReports data

Same steps as the other procedure.

Step 5. opCharts data

Run the migration script:

```
/usr/local/omk/bin/opcharts-cli.exe act=migrate-monitored-services dir=/usr/local/nmis8/var/service_status
debug=1
```

Notes

- It is important to update nmis 9 first and leave running for a couple of poll cycles so inventory is created. nmis9 will create its own inventory, as this is now saved in the db and has a specific format. These data will be used for other scripts from the apps to migrate some data.
- opconfig and opevents share the same database in the old version. This is because the products database name should be changed to "nmis", the old database name.
- nmis8 didn't use mongo. This is because opConfig has its own nodes database. We don't want to replicate information, so now opConfig will use nmisng database to get the nodes information. We would need to use the opConfig migration script to port some specific configurations from the internal database to nmis.