opHA 3 Getting Started

- Introduction
- Installation
    - Installation Prerequisites
    - Getting Started
    - Installation Steps
        - Enter License and accept EULA
- opHA Set Up
    - opHA Discovering a new Peer
    - opHA Edit a Peer
    - opHA Data Synchronisation
    - opHA Roles
        - Change the role
    - opHA Centralised Configuration
    - Activity
    - Peer Status
    - Configuring the remote URLs
    - Configuring the remote URLs - opCharts
- Peer Capabilities

Upgrade to version 3.2.0 and later.
What's next

Introduction

The purpose of the guide is install and set up from scratch opHA 3.0, the Opmantek High Availability solution.

Installation

This guide provides instructions on how to manually install opHA on a server.

Installation Prerequisites

- The individual performing this installation has a small bit of Linux experience
- Root access is available
- Internet access is required for installing any missing but required software packages
- NMIS must be installed on the same server that opHA is being installed on. Here you can read the NMIS installation guide.
- You will need a license for opHA (CONTACT US for an evaluation license)
- **opHA has to be installed onto the Primary and each Poller NMIS server**
- **If using opCharts on Primary each poller will also need opCharts.**

Getting Started

Download the latest product version from opmantek.com



Prior to attempting to install opHA and undertaking configuration it is CRITICAL that clock synchronisation via ntpd or chronyd is fully operational and all systems validate as being synchronise off a similar clock source linked to the global clocking infrastructure.

chronyd is purported to be better suited to virtual environments and contains additional reporting features to assist with troubleshooting when used in a private clock distribution setup. ChronyD is the default for RedHat.

The requirement for good timekeeping is a basic logging element however, SSL crypto is time dependent and variation in time beyond a couple of minutes can cause various rejections, timeout responses and failures.

Having the time of day clock synchronised is an underlying design feature of streaming cryptography such as SSL as a component of the protection mechanism.

Installation Steps

Download or transfer the .run using wget, scp or sftp or any transfer tool. Repeat this process for each server involved.

- Start the interactive installer and follow the instructions (**Note**: Should be run as sudo):

```
/tmp# chmod 755 opHA-Linux-x86_64-3.0.6.run
/tmp# ./opHA-Linux-x86_64-3.0.6.run
Verifying archive integrity... All good.
Uncompressing opHA 3.0.6  100%

++++++++++++++++++++++++++++++++++++++++++++++++++++++++
opHA (3.0.6) Installation script
++++++++++++++++++++++++++++++++++++++++++++++++++++++++

This installer will install opHA into /usr/local/omk.
To select a different installation location please rerun the
installer with the -t option.
```

- The script can also run with in a smarter non-interactive installation with preseeding.
- The installer will interactively guide you through the steps of installing opHA. Please make sure to read the on-screen prompts carefully.
- When the installer finishes, opHA is installed into `/usr/local/omk`, and the default configuration files are in `/usr/local/omk/conf`, ready for your initial config adjustments.
- A detailed log of the installation process is saved as `/usr/local/omk/install.log`, and subsequent upgrades or installations of other Opmantek products will add to that logfile.

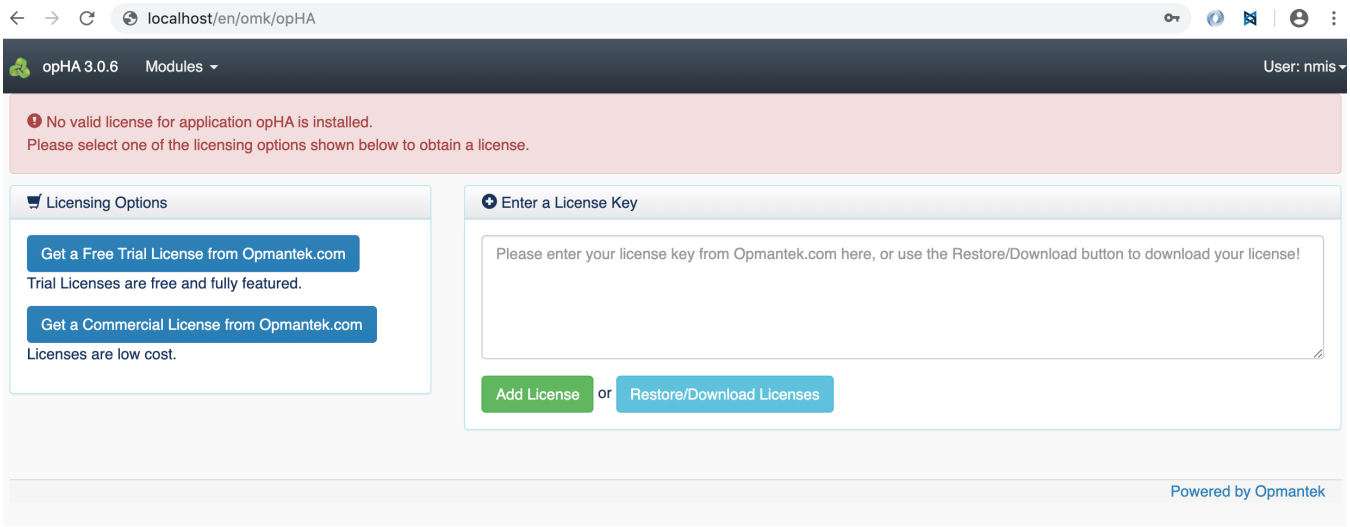
Enter License and accept EULA

If the installation was successful we will be able to see this message:

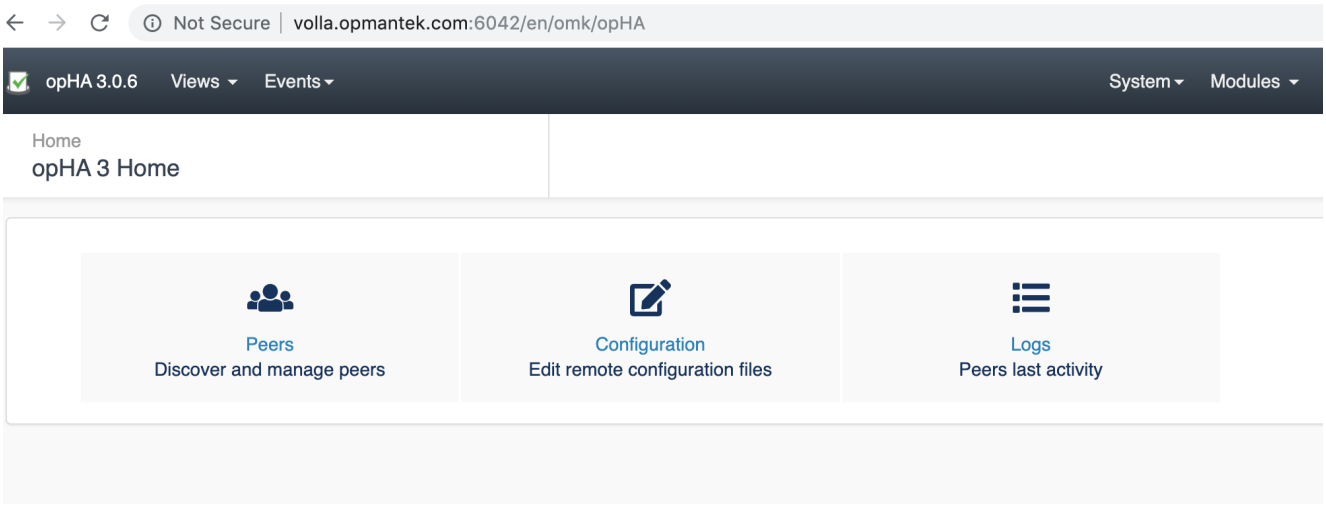
```
If your browser is running on the same machine as opHA was
installed onto, this would be

http://localhost/omk/opHA/
```

*This URL should present you with a webpage that allows you to enter a license key and accept a EULA. This step will need to be completed on **each** opHA instance!*



After successful license key and EULA acceptance you will be presented with a dashboard that looks like this:



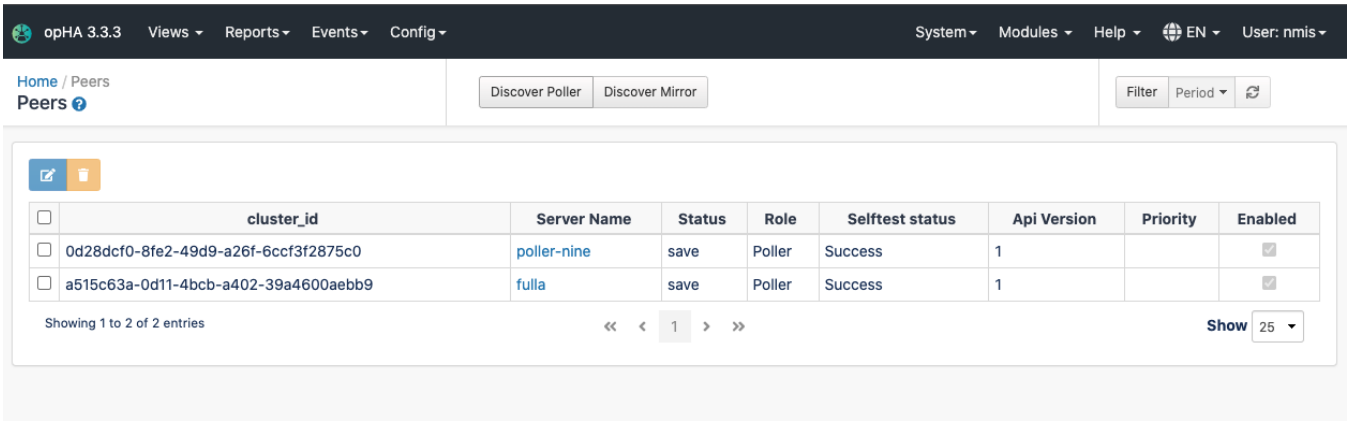
opHA needs to be installed in all the servers - Primaries and pollers - but we will be using the GUI only from the **Primary** server.

opHA Set Up

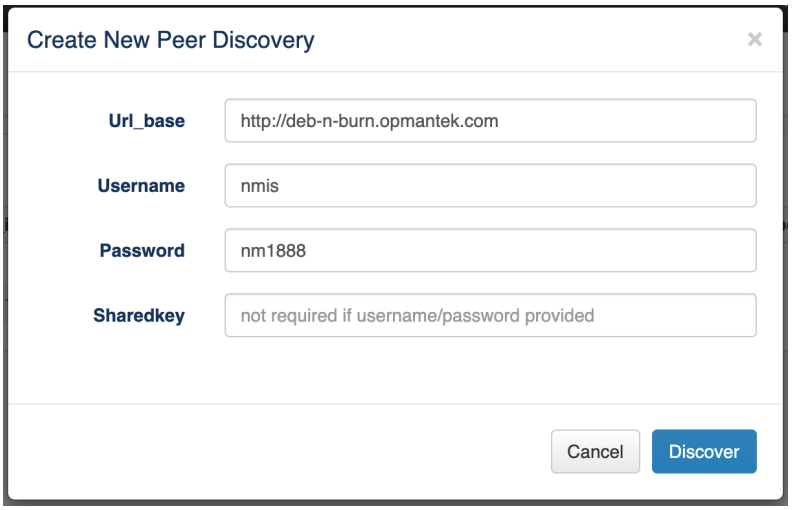
Here you can find how to perform the basis to start running opHA.

opHA Discovering a new Peer

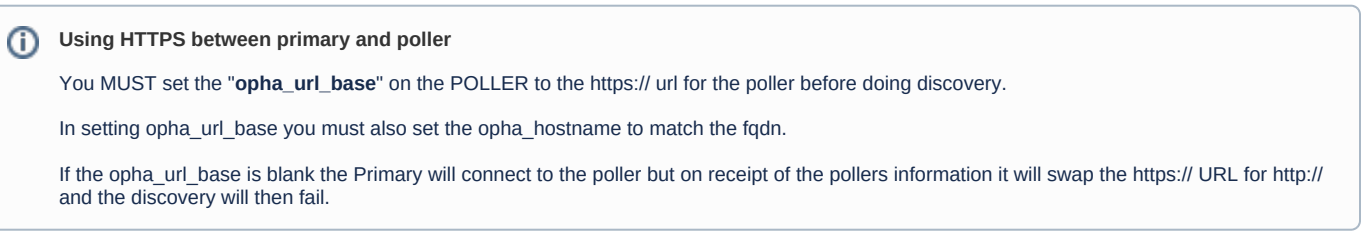
We can add pollers using the **Peers** screen, in **Discover Poller** button:



We need to introduce the follow information for the Discovery:



- URL of the peer
- Username and Password OR the SharedKey
    - For Shared Key this for the value of omkd_secrets set in the /usr/local/omk/conf/opCommon.json of the poller you are doing a discovery on.



Using HTTPS between primary and poller

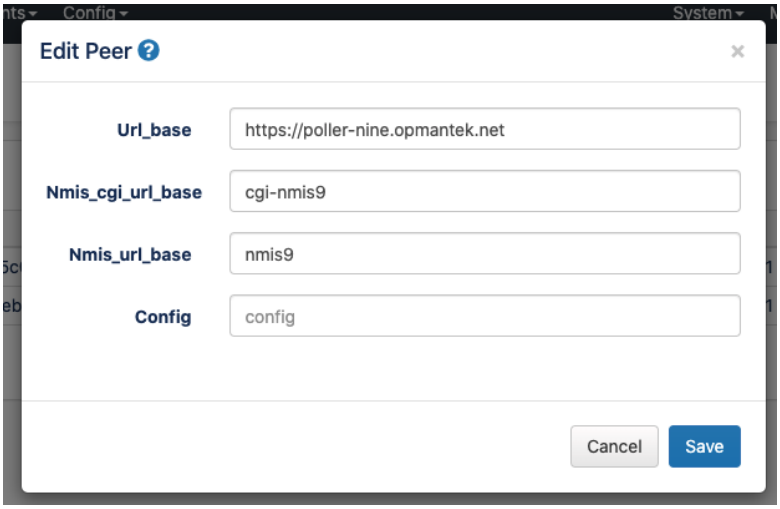
You MUST set the "**opha_url_base**" on the POLLER to the https:// url for the poller before doing discovery.

In setting opha_url_base you must also set the opha_hostname to match the fqdn.

If the opha_url_base is blank the Primary will connect to the poller but on receipt of the pollers information it will swap the https:// URL for http:// and the discovery will then fail.

opHA Edit a Peer

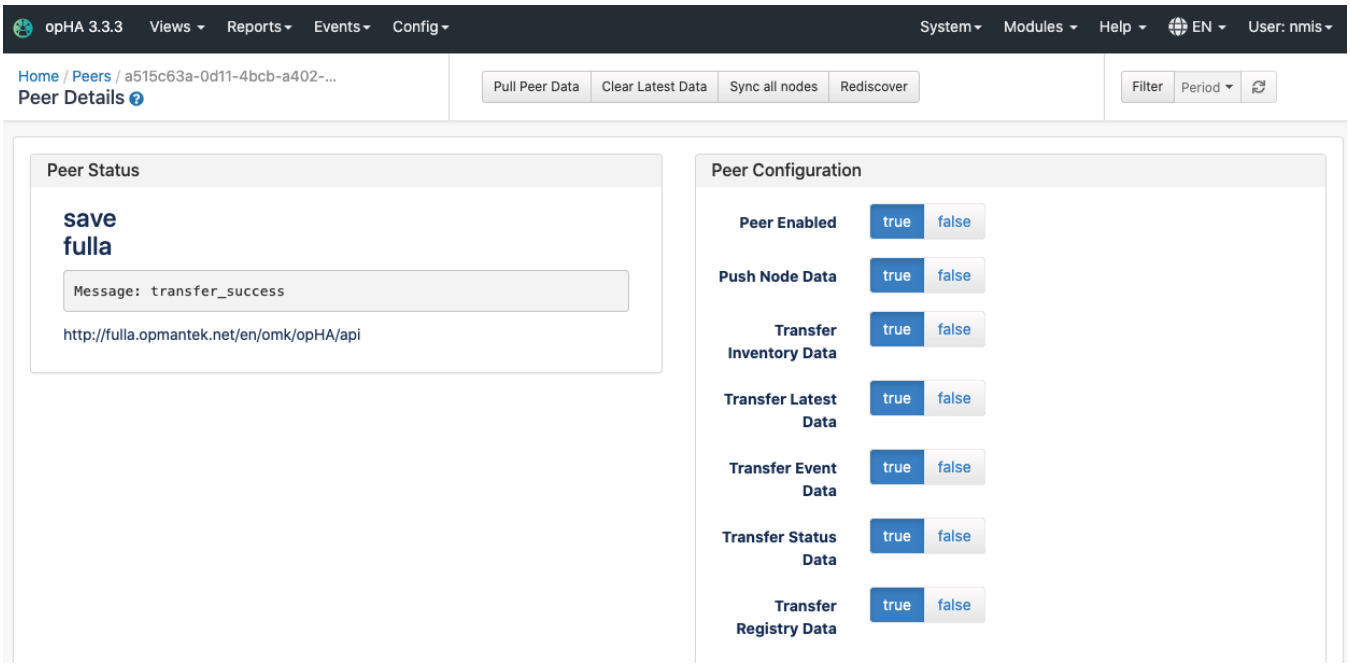
Once the peer was successfully discovered, we can edit the peer configuration - but it is not needed by default:



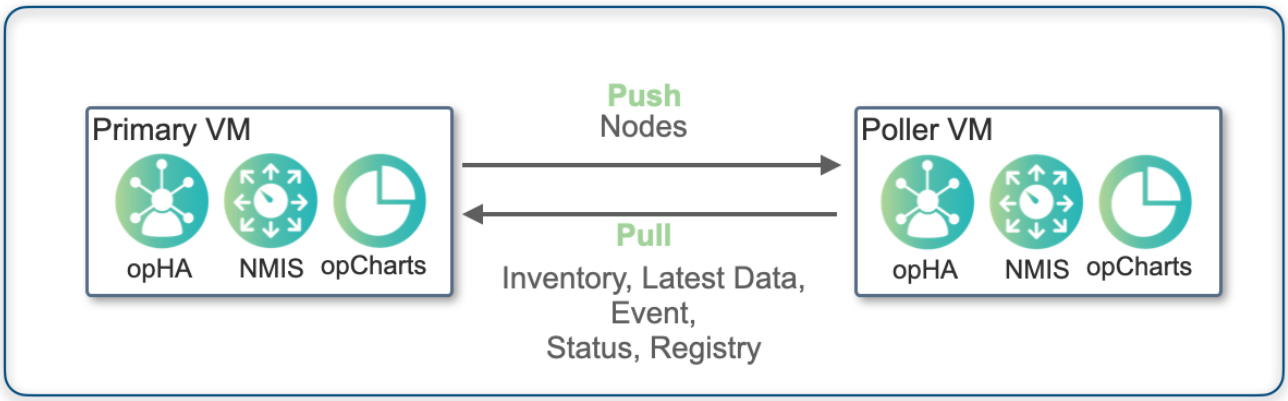
This will edit the nmis information for the poller. It will be used in NMIS to redirect a node to a poller.

opHA Data Synchronisation

Once a peer is added, we can pull the peer to synchronise all the data:



Please note that the nodes are pushed to the pollers, but the other data is pulled:

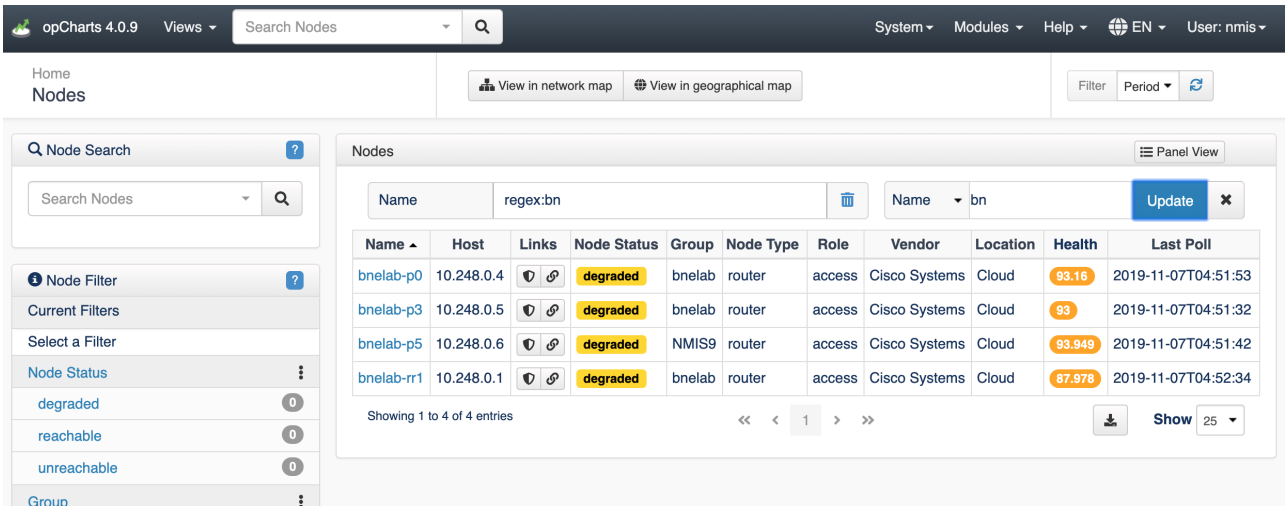


We can also use the **opHA cli tool** to synchronise the data - and use it for other actions:

```
/usr/local/omk/bin/opha-cli.pl
Usage: opha-cli.pl act=[action to take] [options...]
opha-cli.pl act=discover url_base=... username=... password=.... role=... mirror=...
opha-cli.pl act=<import_peers|export_peers|list_peers>
opha-cli.pl act=delete_peer {cluster_id=...|server_name=...}
opha-cli.pl act=pull [data_types=X...] [peers=Y] [force=t]
            pull data types except nodes
            primary <-- peers
...
```

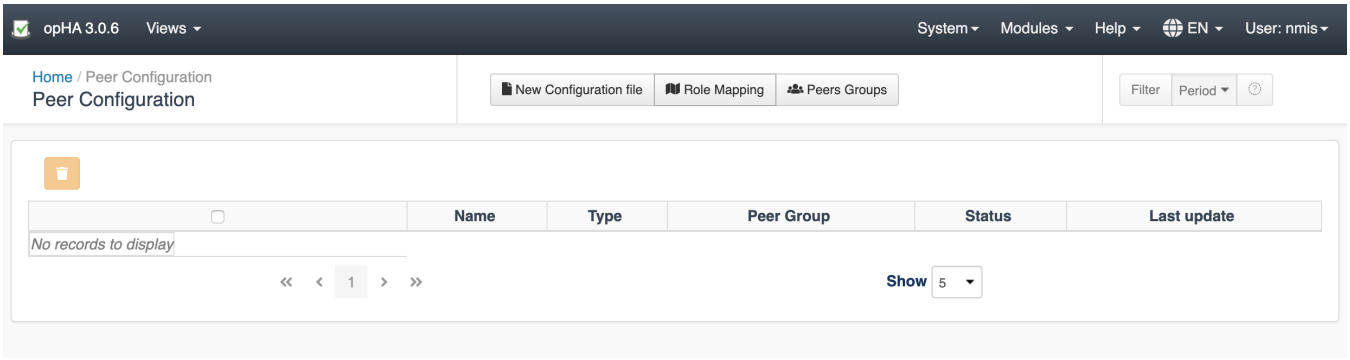
With this tool we can automate the synchronisation process with a cron job to keep all the data up to date.

Once we have synchronised a peer we will be able to see its data from NMIS or opCharts from the **Primary**.



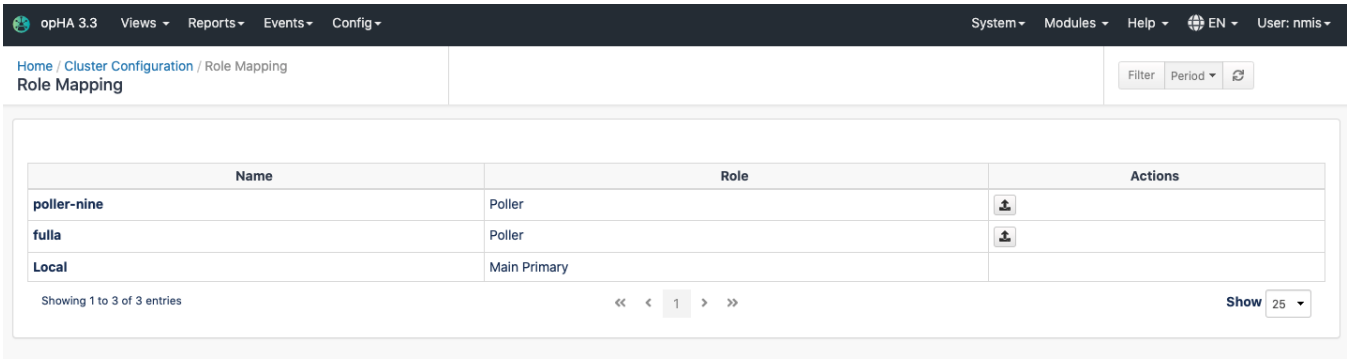
opHA Roles

We can see the peer roles using the **configuration menu > Role Mapping**:



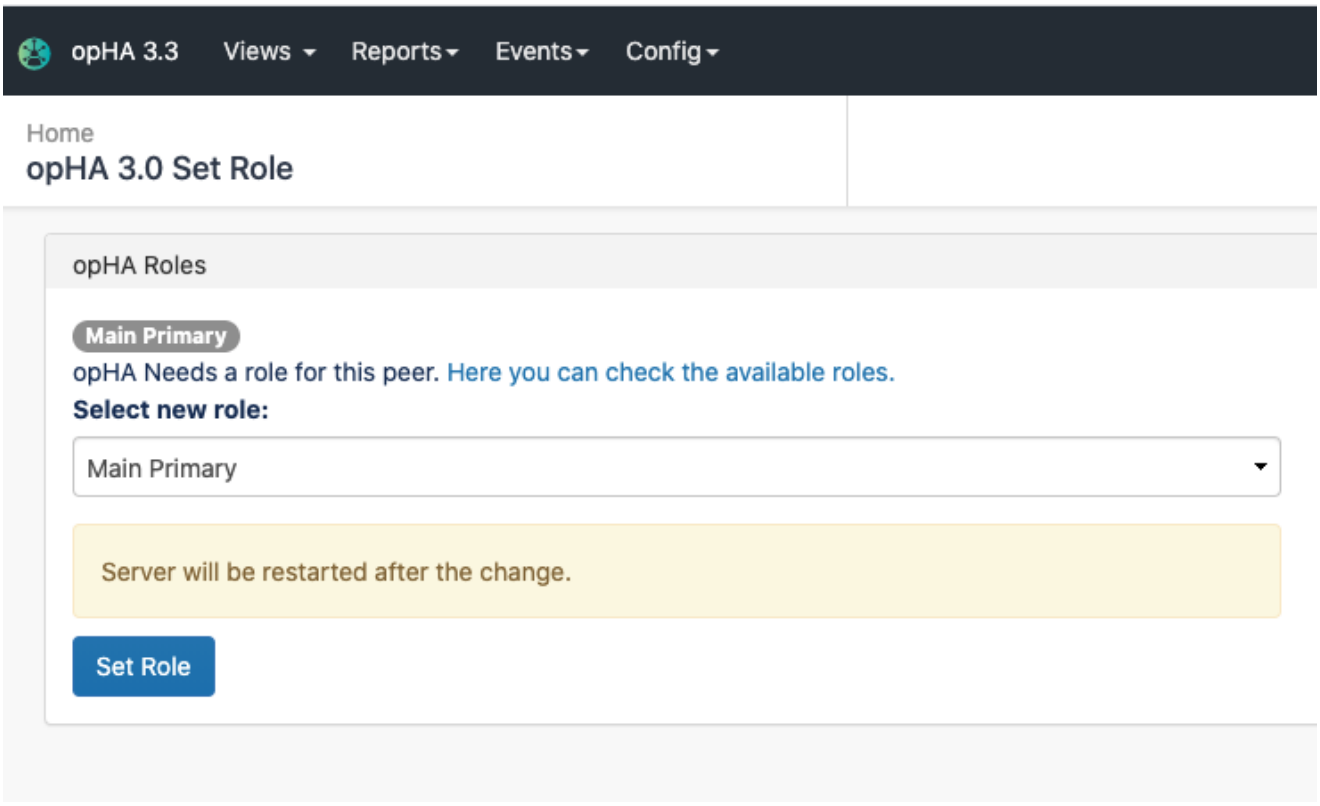
All the roles and capabilities are documented in the following wiki page.

When we create a poller, we should choose which role will be a peer, a poller or a mirror. Once we discover a peer, opHA primary will send the role to the peer. From the role mapping menu, we would be able to see a peer server and the role assigned, and also click in the button to resend the role.



Change the role

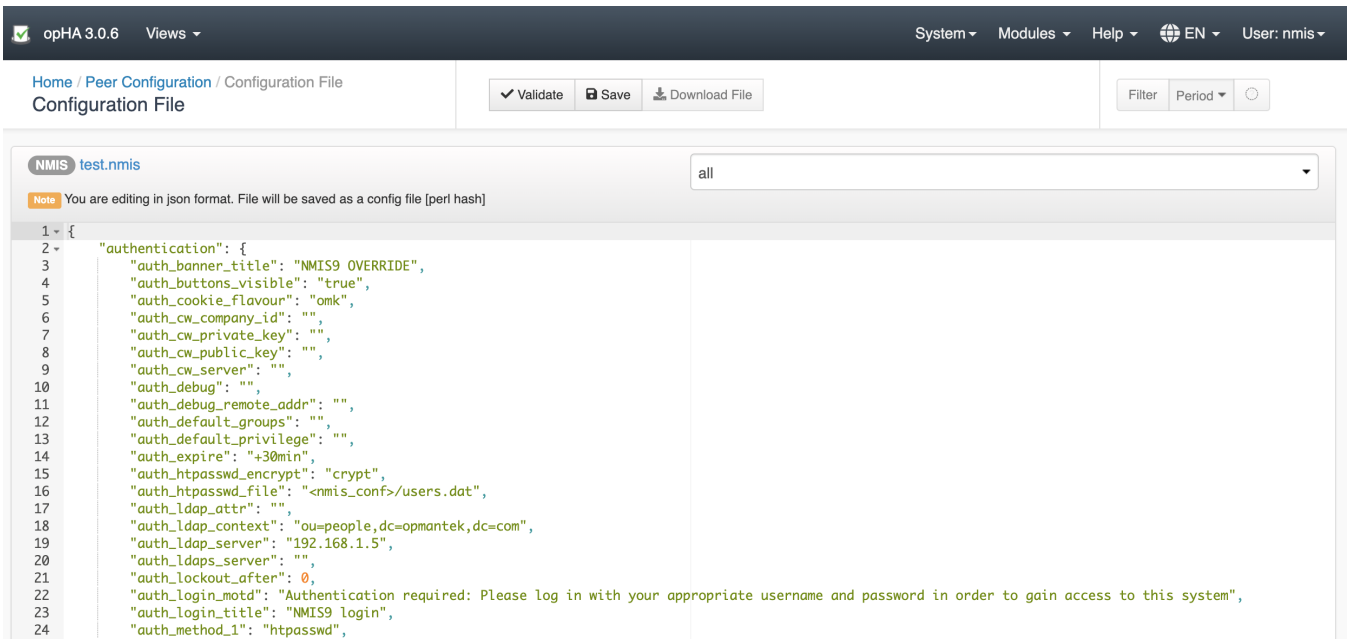
To set up the local role, we can go to the menu *Views* > *Set Role*:



Please note, the server will be restarted after the change, because all the server functionality will change.

opHA Centralised Configuration

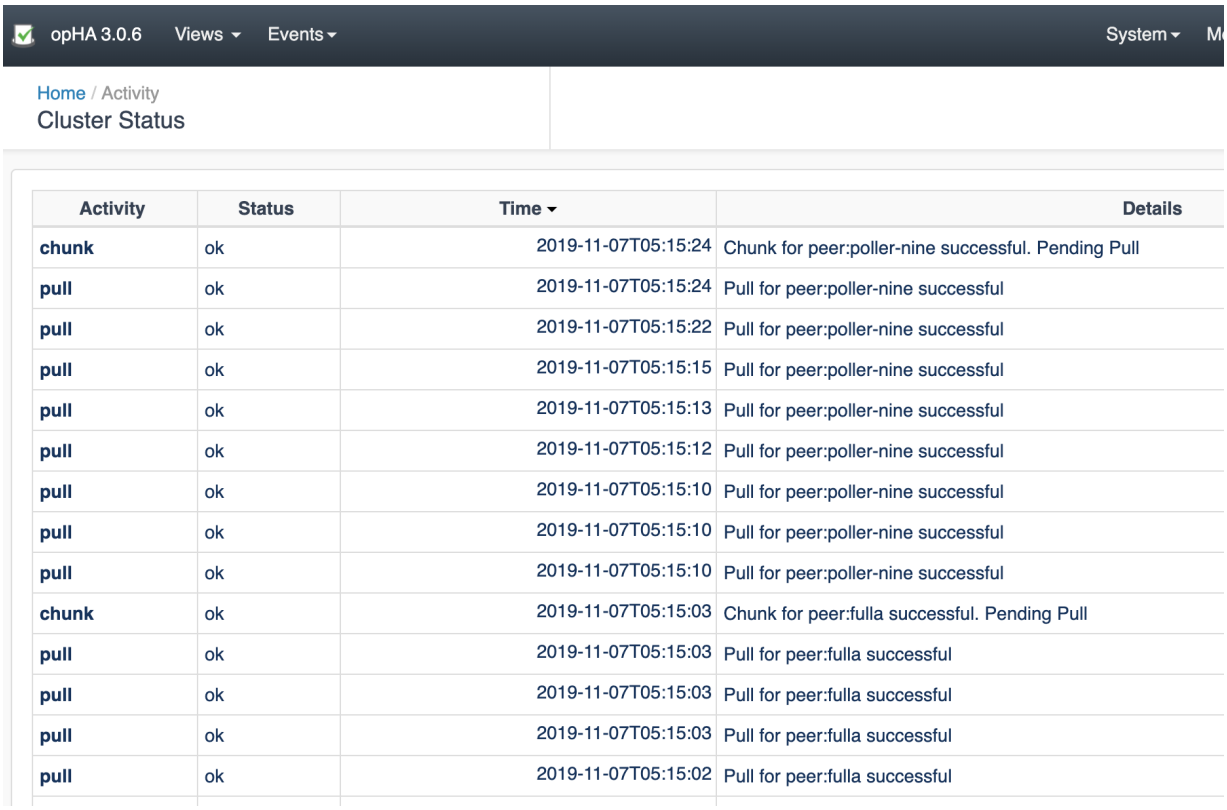
opHA version 3.0.5 brings a new feature to modify the **NMIS** and **omk configuration** from the **Primary** server. The centralised configuration feature allows you to easily and quickly manage a large multi-server network.



It is important to note that once NMIS is updated from a Primary, it cannot be edited from the poller.

Activity

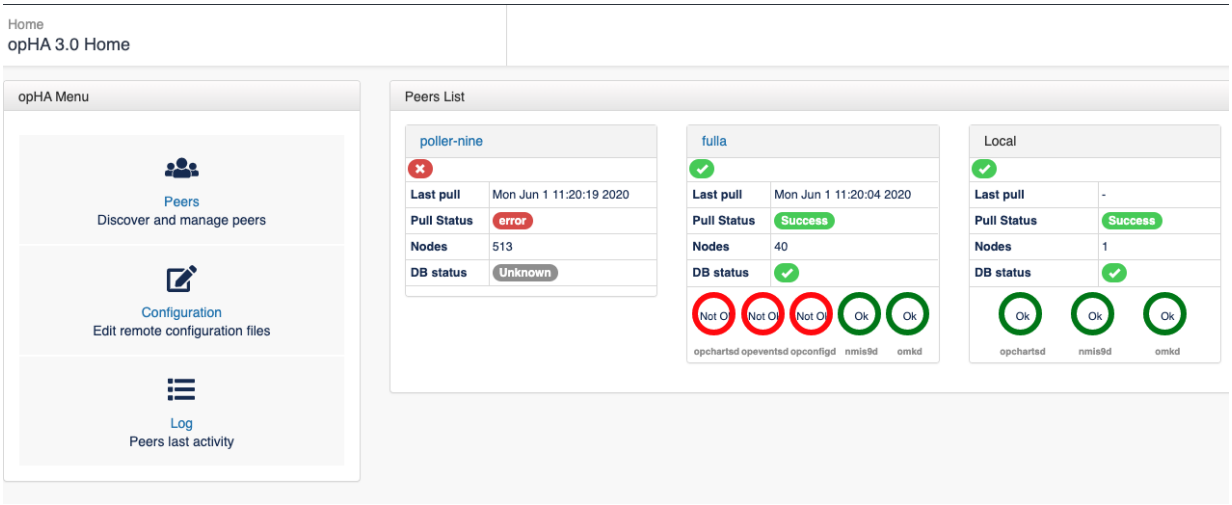
We can check the opHA activity from the activity menu:



Activity	Status	Time	Details
chunk	ok	2019-11-07T05:15:24	Chunk for peer:poller-nine successful. Pending Pull
pull	ok	2019-11-07T05:15:24	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:22	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:15	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:13	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:12	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:10	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:10	Pull for peer:poller-nine successful
pull	ok	2019-11-07T05:15:10	Pull for peer:poller-nine successful
chunk	ok	2019-11-07T05:15:03	Chunk for peer:fulla successful. Pending Pull
pull	ok	2019-11-07T05:15:03	Pull for peer:fulla successful
pull	ok	2019-11-07T05:15:03	Pull for peer:fulla successful
pull	ok	2019-11-07T05:15:03	Pull for peer:fulla successful
pull	ok	2019-11-07T05:15:02	Pull for peer:fulla successful

Peer Status

We can check the peer status in the opHA Home page:

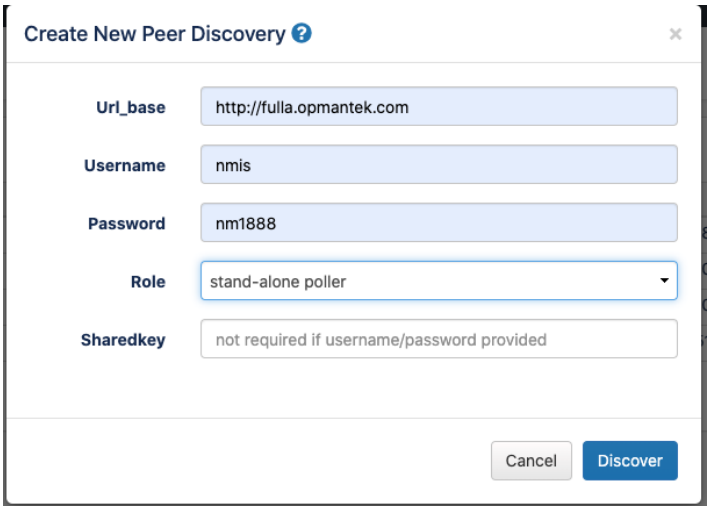


opHA will check in every pull the status of the peer using the endpoint:

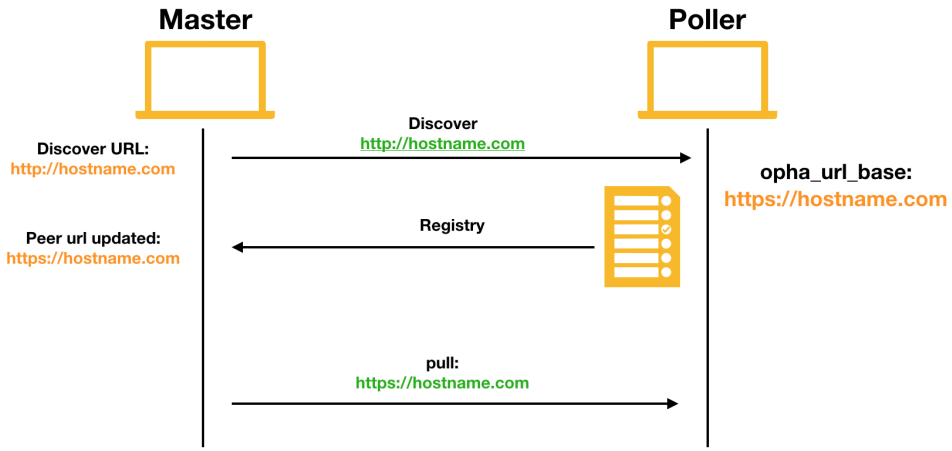
```
http://host/en/omk/opHA/api/v1/selftest
```

Configuring the remote URLs

When we discover a peer, we use the URL_base setting to access to the remote server:



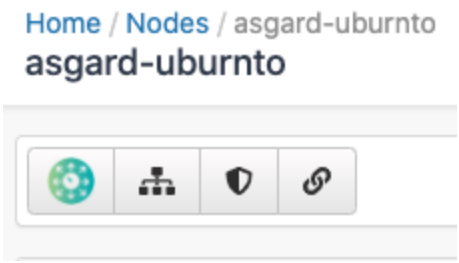
These are the details that opHA is going to use to negotiate the data to be saved from the poller for the data synchronisation.



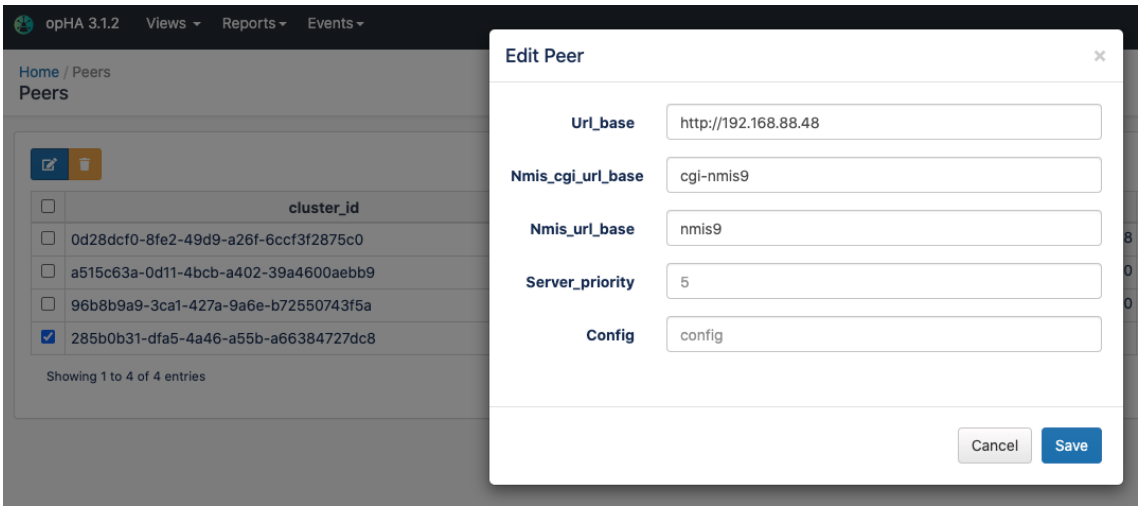
- When we set opha_url_base in the poller, during the discover, that url is going to be sent in the discovery information, that will later be used for the pull operation.
- If we don't set opha_url_base, the discovery url will be used.
- That url will be used by default for view the nodes in NMIS. This includes:
  - Click in a remote node in nmis: We will be redirected to the poller:



  - Click in the NMIS button in opCharts: We will be redirected to the poller - We are able to use this remote nodes NMIS button from opCharts version 4.0.7.



- The NMIS URL can be modified in opHA, when we edit a peer:



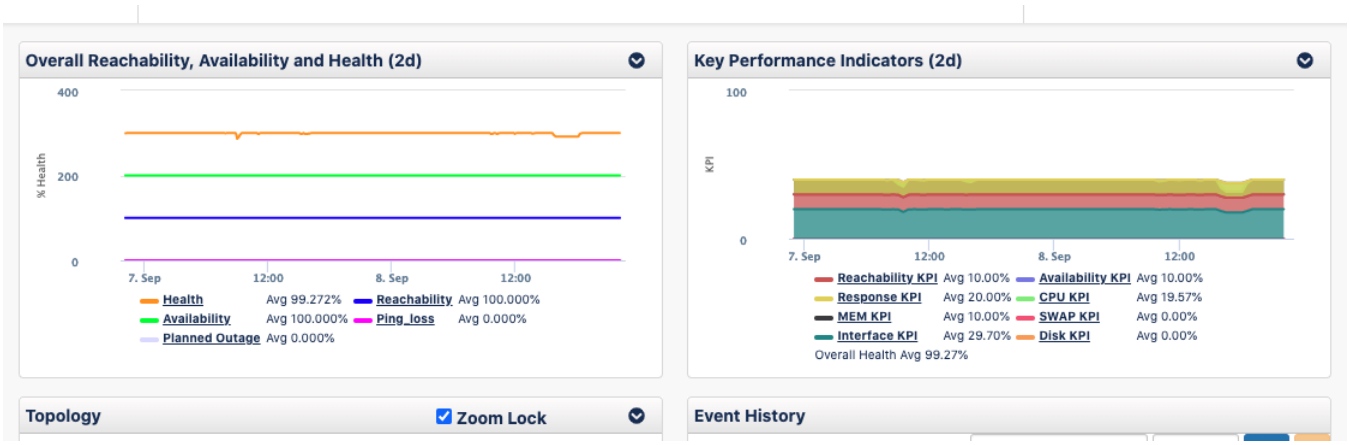
Configuring the remote URLs - opCharts

In the opHA synchronisation, the registry data for each product configured in the server will be sent to the Primary.

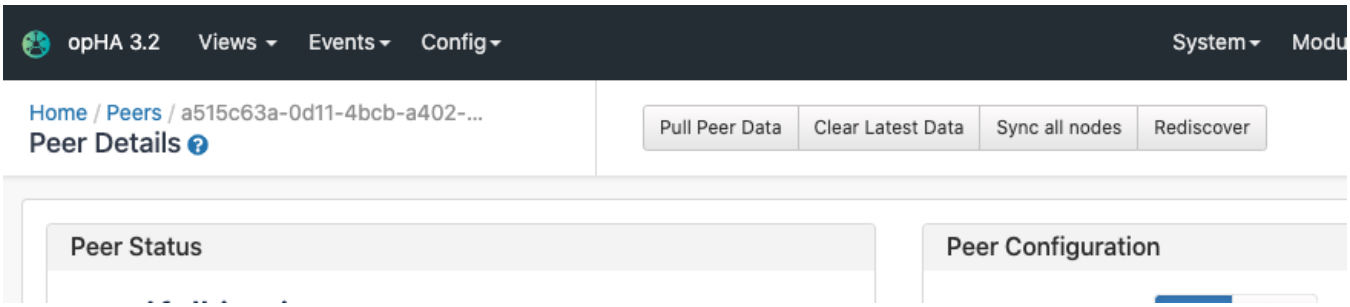
Each registry has a url property. In the opCharts case, it will be used to see the graphs from the poller.

If the registry of the poller is not being generated or the registry pull fails or is not performed, the graphs will not be loaded. So, to make sure everything works:

- Check that omkd_url_base is set in the poller.
- opcharts_url_base and opcharts_hostname can be blank, but the key needs to be defined.
- Check the registry for opCharts in
  `http://host/en/omk/opHA/api/v1/registry`
- Check that the pull is working for that server.



Peer Capabilities



In the following url, http://server/en/omk/opHA/peers/CLUSTERID you can perform several actions about the following peer capabilities:

- **Pull Peer Data**: The Primary will request all peer data enabled - except for nodes from version 3.2.1. Since the last synchronisation date. This action is performed periodically (cron job) for all peers.
- **Clear Latest Data**: Will remove the last synchronisation date, so the next pull will be complete.
- **Sync all nodes**: From 3.2.1. The Primary will send all the nodes to the peer. This means, if the peer doesn't have a node, It will be created. If the Primary does not have a node and the peer does, it will be removed from the peer. You can read more in Centralised node management page.
- **Rediscover**: From 3.2.1. The Primary will update the peer data, this is the registry url access, the user and the server name, based on the peer information:
  - **opha_api_user:** From opCommon.json config file
  - **opha_url_base:** From opCommon.json config file
  - **server_name**: From nmis config file, Config.nmis

Upgrade to version 3.2.0 and later.

Starting from version 3.2.0, we have made significant changes on our internal shared code for all our applications to work on Opmantek's latest and fastest platform, however, previously installed product are not compatible with this version.

To find out more about this upgrade please read:

- Upgrading Opmantek Applications
- Product Compatibility

What's next

- opHA 3 - Centralised configuration
- opHA 3 cli tool
- opHA 3 Redundant Node Polling and Centralised Node Management