

opHA 3 cli tool

- [Introduction](#)
- [Core functionality](#)
 - [Discover Peer](#)
 - [Import and Export Peers](#)
 - [Delete Peer](#)
 - [Pull](#)
 - [Mirror and poller pulls](#)
 - [Nodes Synchronisation](#)
 - [sync-processed-nodes](#)
 - [Import Initial data](#)
- [Cleanup Functions](#)
 - [cleanup](#)
 - [clean orphan nodes](#)
 - [resync nodes](#)
 - [Clean data](#)
 - [Cleanup poller](#)
- [Diagnosis information](#)
 - [get status](#)
 - [Show roles](#)
 - [Data Verify](#)
 - [Check Duplicates](#)
 - [Lock Peer](#)
 - [Setup DB](#)
- [Encryption](#)

Introduction

Version: 3.3.3

The opHA 3 has a cli tool to perform the same operations than the CLI, but with some additional debugging information and it also allows the task automation.

```
/usr/local/omk/bin/oph-cli.pl
```

```
Usage: opha-cli.pl act=[action to take] [options...]
oph-cli.pl act=discover url_base=... username=... password=.... role=... mirror=...
oph-cli.pl act=<import_peers|export_peers|list_peers>
oph-cli.pl act=delete_peer {cluster_id=...|server_name=...}
oph-cli.pl act=pull [data_types=X...] [peers=Y] [force=t]
    pull data types except nodes
    primary <-- peers
oph-cli.pl act=sync-all-nodes [peers=Y]
    sync all node data from the primary to the pollers
    primary --> peers
oph-cli.pl act=sync-processed-nodes
    sync node data based on changes done by NMIS9 node_admin.pl
    primary --> peers
oph-cli.pl act=import_config_data
    for firsts installation, provide initial data (groups)

oph-cli.pl act=cleanup simulate=f/t
    clean metadata and files
oph-cli.pl act=clean_orphan_nodes simulate=f/t
    remove nodes with unknown cluster_id
oph-cli.pl act=resync_nodes peer=server_name
    remove the nodes from the poller in the primary and pull the nodes from the poller
    primary <-- peers
oph-cli.pl act=clean_data peer=server_name [all=true]
    Like resync data but with all the data types
    primary <-- peers
    By default, cleanup just pull data
    all=true includes nodes
oph-cli.pl act=cleanup_poller simulate=f/t
    from the pollers, clean duplicate configuration items and files
oph-cli.pl act=check_duplicates
    check for duplicate nodes
oph-cli.pl act=get_status
oph-cli.pl act=setup-db
oph-cli.pl act=show_roles
oph-cli.pl act=data_verify
oph-cli.pl act=lock_peer {cluster_id=...|server_name=...}
oph-cli.pl act=unlock_peer {cluster_id=...|server_name=...}
oph-cli.pl act=peer_islocked {cluster_id=...|server_name=...}
    Encryption key
oph-cli.pl act=push_encryption_key
```



To get debug information in any command, please run with the following argument:

debug=1..9

E.g. [oph-cli.pl](#) act=resync_nodes peer=server_name debug=8

Core functionality

Discover Peer

You can discover a new peer with the following command:

```
oph-cli.pl act=discover url_base=... username=... password=.... role=... mirror=...
```

Where:

- role: Poller or Mirror

- Mirror: Should we specify the mirror of which poller when role=mirror

Import and Export Peers

We can import, export and list all the peers information with:

```
opha-cli.pl act=<import_peers|export_peers|list_peers>
```

Delete Peer

It is possible to delete a peer with the command:

```
opha-cli.pl act=delete_peer {cluster_id=...|server_name=...}
```

This command will remove the peer and all the associated data: Nodes, inventory, latest data, etc.

We need to specify OR the cluster_id OR the server name.

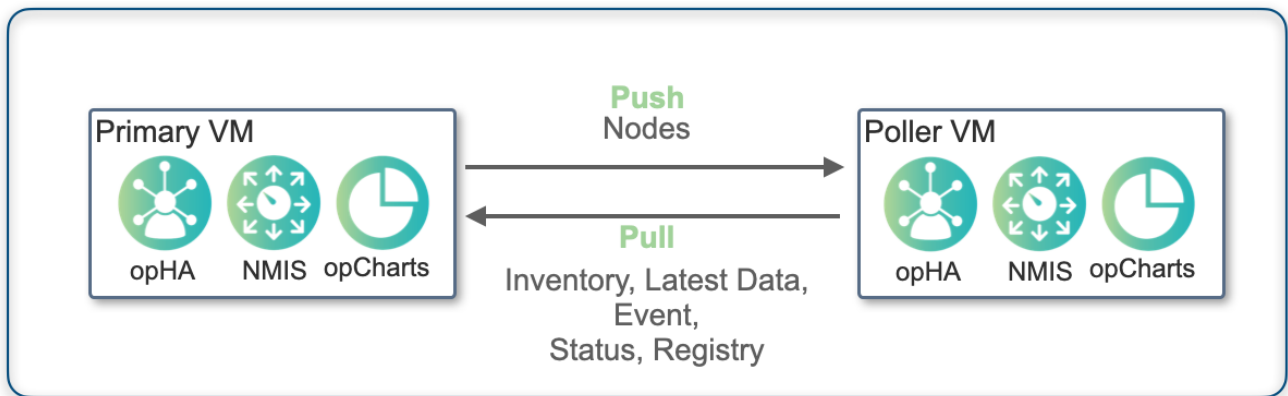
Pull

With pull, we will sync the inventory, latest data, events, status and registry data.

```
opha-cli.pl act=pull [data_types=X...] [peers=Y] [force=t]
```

Where:

- **data_types**: We can specify specific data types, if nothing is specified, all data types will be pulled.
- **peers**: List of peers to pull. If no peers are specified, all the peers will be pulled.
- **force**: By default, just the data changed from the last pull is synchronised. With this parameter, all the data will be moved.



The pull is running in the opha cron job.

Mirror and poller pulls

When we pull from a mirror, if its poller is active, just the registry and status data will be pulled.

It will happen the opposite: If the mirror is active, the poller data won't be pulled.

Nodes Synchronisation

To synchronise the nodes, we can run the polling:

```
opha-cli.pl act=sync-all-nodes [peers=Y]
```

Where:

- **peers**: List of peers to sync. If no peers are specified, all the peers will be pulled.

The sync-all-nodes is running in the opha cron job.

[More information.](#)

sync-processed-nodes

Will sync the nodes processed by NMIS9 node_admin.pl:

```
opha-cli.pl act=sync-processed-nodes
```

[More information.](#)

Import Initial data

for firsts installations, provide initial data, basically setup the groups for pollers and primary and add the peers.

```
opha-cli.pl act=import_config_data
```

Cleanup Functions

cleanup

Function to clean metadata for files and files with no metadata information. This is mainly for configuration files:

```
opha-cli.pl act=cleanup
```

By default, it will run in simulation mode.

Use **simulate=f** to perform the cleanup function.

clean orphan nodes

It is possible to check with nodes are not associated with any cluster id with the command:

```
opha-cli.pl act=clean_orphan_nodes simulate=f/t
```

By default, it will run in simulation mode.

Use **simulate=f** to remove the nodes (And associated data).

resync nodes

By default, the Primary pushes the nodes to the pollers. Running this command, it is possible to update the nodes from the pollers:

```
opha-cli.pl act=resync_nodes peer=server_name
```

Where:

- **peer**: Specify the server name.

Clean data

Will remove all the data from the peer and pull the data again.

By default, it is not removing/resync the nodes. It is possible to do it with:

- `all=true`

Cleanup poller

This operation should be run on a poller. And will clean duplicate configuration items and files:

```
opha-cli.pl act=cleanup_poller simulate=f/t
```

By default, it will run in simulation mode.

Use **simulate=f** to remove the nodes (And associated data).

Diagnosis information

get status

Get all the peer status information as an array of perl hashes:

```
opha-cli.pl act=get_status
```

This is the same information that we see in the opHA front page.

Show roles

Show the roles defined in the system:

```
opha-cli.pl act=show_roles
```

Data Verify

Show how many data do we have for each peer:

```
opha-cli.pl act=data_verify
```

How many inventory records, roles, which peer is active or enabled, also duplicate nodes and catchall inventory records duplicated.

Check Duplicates

```
opha-cli.pl act=check_duplicates
```

Similar to `data_verify`, but will report just the duplicate data.

Lock Peer

(V. >= 3.3.3) When a peer is doing a critical operation, it will be locked. We can see the lock status of a peer:

```
opha-cli.pl act=peer_islocked {cluster_id=...|server_name=...}
```

We can change the lock status of a peer with:

```
opha-cli.pl act=lock_peer {cluster_id=...|server_name=...}  
opha-cli.pl act=unlock_peer {cluster_id=...|server_name=...}
```

Setup DB

Setup DB indexed. This is run by the installer during installation or upgrade:

```
opha-cli.pl act=setup-db
```

Encryption

The primary can push the encryption key to all the pollers by running the following command:

```
opha-cli.pl act=push_encryption_key
```

It will run just if the server has the role primary and the key was not modified since the last time it was sent.

To force send it anyway, you can run it with the force=true argument:

```
opha-cli.pl act=push_encryption_key force=1
```