

opConfig adding nodes and credentials

This guide covers basic configuration for adding new nodes, providing access credentials and checking operation.

There is an advanced configuration guide which then covers compliance management, creating support for new operating systems.

Steps

- [Prerequisites](#)
- [opConfig Concepts](#)
- [Configure access: Adding Credential Sets, Managing Credential Sets](#)
- [Adding or Modifying Nodes](#)
- [Automating node import](#)
- [Checking operation](#)
- [Extend and customise with advanced configuration](#)

Prerequisites

- opConfig installed and setup
- Understanding of opConfig terms and operation refer to

opConfig User Manual

opConfig Concepts

The main concepts to bear in mind are *nodes*, *credential sets*, *commands*, *changes* and *revisions*:

- **NODES:**
 - Nodes are devices/computers that opConfig knows about, and which it is configured to run commands for.
 - As opConfig needs to connect to the node in question and execute commands on that node, the node needs to be configured with access credentials. In opConfig these are stored independent from the nodes in what opConfig calls credential sets.
- **CREDENTIAL SETS:**
 - Credential sets are a combination of usernames, passwords, privileged passwords etc. allowing access to the devices CLI.
 - Once the credential set has been used to create a working CLI access then "commands" can be issued and the results recorded.
- **COMMAND SETS**
 - Commands are normally command line constructs which will be executed on the node in question.
 - (Some are "passive commands" like "audit-import" which are not actually run on the node but the result is associated with node.
 - Commands can be grouped and collected into what opConfig calls a "command set". Command sets are configured to apply only to particular OS and maybe versions or platforms.
 - The command output is captured and stored by opConfig.
 - Command outputs are compared against the previous revision, and if different it's saved as a new revision in opConfig. It could also be a one-shot command which is not analyzed in great detail (e.g. a process listing or some other diagnostic command)
 - A command can be marked for change detection in which case more detailed analysis occurs for changes.
- **CHANGES / REVISIONS:**
 - Revisions are the command outputs as collected over time.
 - opConfig lets you distinguish between "very dynamic" and "mostly static" commands in an efficient flexible fashion:
 - Static commands should be marked for detailed change detection. In this case, a new revision is created if and only if there are (relevant) differences between the most recent state and the new command output.
 - Dynamic commands should not be marked for change detection. Every time such a command is run, the output will be saved as a new revision - regardless of whether there were any differences between historic and current state.

Configure access: Adding Credential Sets, Managing Credential Sets

Credentials for all connections made by opConfig are configurable from the opConfig GUI ONLY. Before anything else you need to create sets of credentials to access you devices. At this point in time, opConfig supports only Telnet and SSH, and for SSH only password-based authentication is supported.

Select the menu "System", then "Edit Credential Sets". Credential sets can be shared by any number of nodes.

Each credential set has to have a unique name, by which it is referenced in the nodes' connection settings. The description field is self-explanatory and optional.

A credential set has to specify a User Name property, which is used when logging in to the nodes the set applies to. At this time, opConfig supports only password-based authentication at the node, and the Password property of the credential set establishes the primary password for this user name.

SSH Key-based authentication

SSH Key-based authentication is supported from version 3.0.2. Considerations:

- A key with no passphrase is needed.
- The key needs to be added to the device.
- The key needs to be set in RFC4716 format.

As a key example configuration:

[Home](#) / [Credential Sets](#) / [Edit Credential Sets](#)

Edit Credential Sets

Name	<input type="text" value="root access key"/>	?
Description	<input type="text" value="Enter New Description"/>	?
User Name	<input type="text" value="root"/>	?
Password	<input type="text" value="Enter New Password"/>	?
	Current State: Not Set!	
Password (Superuser/Privileged/Enable)	<input type="text" value="Enter New Password (Superuser/Privileged/Enable)"/>	?
	Current State: Not Set!	
Automatically Privileged	<input type="text" value="Yes"/>	?
SSH Key	<pre>-----BEGIN RSA PRIVATE KEY----- AAA... -----END RSA PRIVATE KEY-----</pre>	?
	Current State: Set	

An SSH Private Key without passphrase, in RFC4716 format. The key is never shown after being set. To (re)set, copy and paste the key here.

Unprivileged user

Some commands cannot be performed by an unprivileged user, which is why opConfig also supports elevating the privileges on demand. To control this, a credential set can optionally include a Superuser/Privileged/Enable Password. Depending on the node's platform and personality, different mechanisms will be used to gain increased privileges:

- On Cisco IOS devices, this password is used with the `enable` command.
- With personality `bash` (the default for Unix-like systems), the command `sudo` is used to become the superuser. Sudo therefore needs to be installed and configured on such nodes, and the User Name in question needs to be authorized for sudo.

Naturally not all commands require elevated privileges; see the section on Command Sets for how to determine and configure those.

Please note that the Credential Set editing dialogs *never show existing passwords* (or their length or existence); You can only overwrite password entries. All credential sets are stored in the database in encrypted form.

Adding or Modifying Nodes

To tell opConfig to run commands for a node it needs to be told about the node's existence and what properties the node has (e.g. what platform, what OS, what credential set, what protocol to use to contact the node). Adding a node for opConfig can be done using the GUI or the command line tools `opconfig-cli.pl` and `opnode_admin.pl`. You can add node information manually to opConfig, or you can import node's info from NMIS or OpenAudit.

opConfig can connect to any node (and run commands for it) as long as it has valid connection settings for it (and as long as it is not disabled for opConfig).

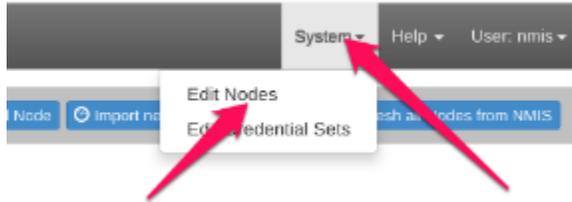
Add a node Using the GUI

Add or Import

Note

Import is only available for opConfig version prior to 4.0.0. opConfig versions \geq 4.0.0 share that information in the database.

- System menu
 - Edit Nodes.



- "Import new Nodes from NMIS" or "Add Node" - These let you create new node records either automatically or manually.

If you successfully import the node from NMIS you should only need to add the credential set and the transport protocol (which are in the connection tab). Import generally works for "Linux" like devices and for Cisco devices. For all other device types you simply need to add some details by hand. You will see what configuration you MUST still add displayed as part of the "Edit Node" screen.

Configuration Problems

- OS Info missing or incomplete
- Personality is not set
- Transport is not set
- No Credential Set selected

The problem reports are fairly self-explanatory (and clickable).

• The following is a breakdown on the information opConfig uses about the device.

1. General TAB - This is generic information about the device and is the information imported from NMIS / OpenAudit. Only the host entry needs to be correct here, and it must be a usable FQDN or IP address the rest is informational only.
2. Connection TAB - To connect to a node, opConfig needs to know some information about it
 - a. Personality this is the CLI Parsing to use to enable the issuing of commands e.g. line endings, prompts etc. The Personality includes information about the prompts, line-ending conventions etc. a node is subject to; for example, the 'ios' personality handles understanding the > prompt and "enable" command and "bash" understands shell prompts. The personalities supported are available in the drop down.
 - b. CredentialSet - *NOT automatic* and needs to be set - authentication and authorization in the form of the access credential set created earlier.
 - c. Transport (Telnet or SSH) - *NOT automatic* and needs to be set Also note this cannot get flagged as not being changed in the Configuration Problems window so do check it.
3. OS info TAB - Once connected to a node we need to know the OS and maybe version, subversion, platform in use to select the right commands to issue and how to parse the command results. This where COMMAND SETS ("command_sets.nmis" file) that opConfig uses, makes association between the OS and maybe a version and maybe a major release or train and the command to issue and how to parse it.
 - a. These fields should be automatically populated if your device was discovered by NMIS or OpenAudit and if they are Cisco IOS or Linux devices
 - b. The OS field and potentially the version and other fields must match the 'os' => and any 'version' => fields in the command_sets.nmis file.
 - c. See the command sets section later and have a look in the file if you want to know what os and version fields will work. If the import did not get results you can try the following: for Cisco IOS typically if you put OS as "IOS" and version as "12.2" you will get results and Linux OSs use just OS as "Linux"

Once you have added the device you will either need to wait for the polling cycles to complete per your cron Schedule or use command line tools below to determine results.

Import (and discovery) from the Command Line

opConfig CLI tools are found in `/usr/local/omk/bin`

Simply run `opconfig-cli.pl` without options to see a brief usage of help.

`opconfig-cli.pl` can import nodes from NMIS, to import you'd run

opconfig import from NMIS

```
opconfig-cli.pl act=import_from_nmis
##optionally you can limit it to the names of known nodes with an argument of nodes=
opconfig-cli.pl act=import_from_nmis nodes=nodeX,nodeY
```



If you have already setup credential sets, then you can let opConfig guess which to use for your node using

```
opconfig-cli.pl act=discover node=TheNewNodeName
```

If none of the Transport+Credential Set combinations work for the node, opconfig-cli.pl will print an error message.

Automating node import

One can use a cron entry to schedule the opconfig-cli.pl processes as described above, this would import new nodes and update old node information from NMIS on a regular basis.

There are two things to consider when automating this. Firstly the "discover" process which tests for a working combination of the transport (ssh/telnet) and credentials is not recommended for larger numbers of nodes or if have more than about 3 or 4 credential sets to try for obvious reasons. Secondly you may want to apply a filter on which nodes should be considered as Active for opConfig.

One way to automate some of these is to create custom entries in your OS_Rules.nmis. You can use OS_Rules filter and set policies to apply the transport type, the credential set to use and even the enabled or disabled flag.

Here is an example which extends the Cisco IOS section to automatically set the transport and credential sets for all Cisco devices.

OS_Rules

```
10 => {
    'IF' => {
        'sysDescr' => qr/IOS XR/,
    },
    'SET' => {
        'connection_info.personality' => 'ios',
        'os_info.os' => 'IOS-XR',
        ### These two lines then SET SSH as the transport and the credential_set to use for
        username and password or SSH keys.
        'connection_info.transport' => 'SSH',
        'connection_info.credential_set' => 'default_username_password',
    },
    'CAPTURE' => [
        .....
    ],
    ### One could also set ALL devices to use SSH etc with the following -
    ## Note: this is rule 1 and does not Break so continues to apply the OS_Rules
1 => {
    'IF' => {
        'sysDescr' => qr/\\S/,    ## Matches any sysDescr that is not blank or just a space
    },
    'SET' => {
        'connection_info.transport' => 'SSH',
        'connection_info.credential_set' => 'default_username_password',
    },
    BREAK => 'false',
},
```

Checking operation

[opConfig 4 Troubleshooting](#)

Extend and customise with advanced configuration

[opConfig 4 User Manual](#)



Related articles

- [opConfig adding nodes and credentials](#)
- [opConfig Setup Guide](#)