Table of Contents

CMDB jobs

Basic command jobs issues CLI commands to any (supported) device. It allows the use of simple CLI commands and templates, variable substitution from the CMDB node, the 'CMDB' Region, or the Domain and scenarios.

It requires some attributes to be set:

- Management IP (cmdb node)
- CMDB node linked to a Domain
- Username (domain > User > Rme_user)
- Password (domain > User > Rme_passwd)

Selecting the nodes

The nodes can be added directly using the text field. Multiple hostnames, ip addresses or FQDNs can added be at once. Just make sure they are separated by a comma, whitespace, semicolon or newline. If these nodes already exist in the database, that specific node is used. If the node doesn't exist yet, you are still allowed to execute jobs on that node, but you will have to enter the required Domain and vendor each time.

Alternatively, nodes can be selected from the `Node groups` as well. The `Group tag` filter can be used to filter the type of node groups shown.

Selected nodes

Once the [Next] button is clicked, the selected nodes are listed at the top of the page. If a nodes' information is incomplete, dropdown menus are enabled to complete the information. The checkbox left to each node name marks it for scheduling a job for this node. Nodes are automatically de-selected if information was lacking.

The `Address` field will accept IPv4 or IPv6 addresses or the full-qualified-domain-name (fqdn) for the node. This will be used to establish a management session with the node. In case a fqdn is used, ensure the DNS can resolve it.

When the nodes' source is labeled as a 'CMDB' node and the user had to provide additional information before it could be scheduled, this information will be saved. This will only happen is the node is actually scheduled.

Push Basic Commands

Selected nodes:

	Node	Address / FQDN	Domain (service-code)	Source Vendor_type
☑	172.17.0.24	172.17.0.24	LAB ⌄	
				Cisco_Nexus ⌄
☑	ls1-lab	192.168.0.201	YCE (YCE)	YCE
				Cisco_Nexus
☐	ls2-lab	192.168.0.202	YCE (YCE)	YCE
				Cisco_Nexus
☐	ss1-lab	192.168.0.200	YCE (YCE)	YCE
				Cisco_Nexus

Select all

Jobs, Commands, Scenario

The Commands and Scenario text fields can be modified for single use or can be stored to re-use.

Load job

| filter | filter stored jobs by name | x | Filter |
| name | some_vars_example ▾ | Load | Delete |

Save job

as	some_vars_example	Save
for	client-type CMDB ▾	share as ☑ public
descr	Issue commands to the selected nodes

loaded '**some_vars_example**' by group member 'NetYCE support'

Commands:

```
 1  Node: <node>
 2  fqdn: <node_fqdn>
 3  vendor_type: <Vendor_type>
 4  Admin_domain: <Admin_domain>
 5  # Once 1 is done, all others below can be found as well. Interesting.
 6  CMDB tag: <Cmdb_tag>
 7  CMDB attr1: <Cmdb_attr1>
 8  CMDB attr2: <Cmdb_attr2>
 9  CMDB attr3: <Cmdb_attr3>
10  CMDB attr4: <Cmdb_attr4>
11
```

Evaluate

Scenario:

```
1  [parameters]
2
3
4  [scenario]
5  Description <node> Basic command job...
6  task = Basic_cmd_job
7
8  end
```

Load job

Modifications made to `Commands` and `Scenario` can be stored. This can be use full for recurring changes, like password updates.

To save a job, make sure to provide a unique name and press save. This will make sure the client-type, description, Commands and Scenario values are saved.

Next time this job is available from the drop-down menu. By marking the stored-job `public`, other users will have access to this job definition too.

Commands

The Commands field can be used to provide the configuration lines to be sent to the node. These line

can be a combination of:

- plain configuration lines
- configuration lines using variables (`<variable>`)
- template name (`{template}`)

Variables are available as attributes within the node (either ad-hoc, CMDB or YCE). Ad-hoc nodes (unknown to CMDB or YCE) don't have any variables but will defer to shared CMDB variables. Since all nodes refer to a Domain for the login credentials, all variables of the referred Domain are available to the job. Likewise, the Region 'CMDB' offers its set of (customizable) variables.

Besides the database variables, runtime variables are available as well and can be added to the `[[parameters]]` section of the `Scenario`.

Templates can be included using `{template}}` syntax. This allows to have vendor agnostic jobs, since templates with the same name can exist for every vendor (and subtype). See templates for more information.

A **plain configuration** line could be:
`snmp-server host 1.2.3.4`

The same could be achieve using a **domain variable**:
`snmp-server host <Snmp_traphost1>`

Calling the **snmp template** would look like:
`{snmp}`

Evaluate

The `[Evaluate]` button allows you to test the configuration and the variable substitution for a specific node.

The first node with its checkbox set will be used for the 'evaluate' function.

Scenario

With the scenario you are in control of the order of communication and execution of commands and functions.

The scenario allow you to specify runtime variables in the `[[parameters]]` section and allows tasks and functions to be entered within the `[[scenario]]` section.

Below an example is shown on the parameter variables for runtime use.

Scenario:

```
1  [parameters]
2  user1 = netyce
3  passwd1 = difficultpassword
4
5  [scenario]
6  Description <node> Basic command job...
7  task = Basic_cmd_job
8
9  end
```

The `task = Basic_cmd_job` is a shortcut to include the stored scenario of that name at this point in the scenario. The (default) contents of this scenario/task simply imports the commands one by one on the node CLI and aborts on the first command failure.

On error, the action is logged a failure and the scenario **stop**s and is listed as FAILED, otherwise the log lists completed and the job **end**s as SUCCESSFUL.

```
Description <node> Basic command job

Cmd_exec_basic -n <node> -a <node_addr> -v <vendor_type> -d <domain> -f
<node>.cmd <verbose>
if <error>
    Log_action -n <node> -a Basic_cmd_job -m "Failed executing commands"
    stop
endif
Log_action -n <node> -a Basic_cmd_job -m "Completed executing commands"
```

Here you can dive deeper into scenarios and its functions.

Scheduler

Once you are happy with the configuration you wish to push, the job can be scheduled using the scheduler. Select an execution moment in the future or select Now and press schedule.

Schedule:	tomorrow ▼ at 5 ▼ 05 ▼ or ☑ now	change-id: C000xxxxxx
Schedule	-auto- ▼ server	☑ verbose log
Cancel		

If you wish to read more on the scheduler, more information can be found here: Scheduler