

Table of Contents

YCE jobs 3

Creating a command job 3

Select Nodes 3

Stored jobs 4

Commands 5

Scenario 8

Scheduling 8

YCE jobs

Command jobs are used to configure modelled nodes with the possibility to use templates, variables, conditionals etc.

Creating a command job

To create a command job select the node(s) you wish to send a job to. You may enter the nodes directly using their hostname, or use a [node group](#).

Push Command jobs

Node list

Enter nodes, sites or clients (with wildcards).
Items should be separated (comma, whitespace, semicolon, newline).

Add list

Group tag: Client

Refresh

Node groups

BT93812
campus
LAB
ServerCompany
sp

>>

<<

Selected Devices

it002-s50
it002-s51
it002-s52
itand-s01
itand-s50

(5)

Next

Select Nodes

Using the initial node group and node list selection, a summary of these nodes is given. Use the checkboxes to individually include the nodes for which a command job will be scheduled.

Technical documentation - <https://labs-wiki.netyce.com/>

Push Command jobs

Selected nodes:

BT93812 / IT002

[SA] Via Chieri 68, Andezeno (10020)

<input type="checkbox"/>	IT002-S50	active / Acc2960	Cisco_IOS
<input type="checkbox"/>	IT002-S51	active / Acc2960	Cisco_IOS
<input type="checkbox"/>	IT002-S52	active / Acc2960	Cisco_IOS
<input checked="" type="checkbox"/>	ITAND-S01	active / SA2960	Cisco_IOS
<input type="checkbox"/>	ITAND-S50	active / Acc2960	Cisco_IOS

Select all

Stored jobs

A stored job may be selected from the dropdown menu. You may also save your own jobs, either privately or available for your colleagues. They can be restricted to only be available for a certain Client-type (tentant) if desired.

In the example below the stored job “ACL_Edit” was loaded. This retrieved the command and scenario data stored under that name which was saved for re-use by one of the NetYCE users.

A stored job can be modified by members of the same user group only.

Load jobfilter x

Filter

name v

Load

Delete

Save jobas

Save

for client-type share as ☒ publicdescr

loaded 'ACL_Edit' by group member 'sanjay'

Commands:

```
1 ip access-list resequence <acl_name> 10 10
2 ip access-list extended <acl_name>
3 |deny_ip| 1 deny ip any host <deny_ip>
4 |permit_ip| 2 permit ip any host <permit_ip>
5 ip access-list resequence <acl_name> 10 10
```

Evaluate

Scenario:

```
1 [parameters]
2 acl_name = test
3 permit_ip = 1.2.3.4
4 deny_ip = 5.6.7.8
5
6 [scenario]
7 Description <node> Command job...
8 task = Command_job
9
10 end
```

Commands

The *Commands* input field, can be used to apply static configuration lines or [templates](#), with all capabilities a template could ([variables](#), [conditionals](#), [relations](#), [functions](#), ...) or it could be empty if you wish to apply the configurations differently using scenarios.

If you wish to store this information in the job, don't forget to save it.

Commands:

```
1 |<node_position> == NA| {tpl_primary_device}  
2  
3 ! remaining configuration for all devices  
4 ntp server 192.168.13.26
```

Evaluate

Below the commands input field, an Evaluate button can be used to test/verify your configuration. This will test its outcome against the selected node.

If multiple devices are selected the first one will be used for the evaluation only.

Push Command jobs

[Back](#)

Evaluating commands and scenario for ITAND-S01

...(wait)...
completed

Commands for ITAND-S01:

```
ip access-list resequence test 10 10
ip access-list extended test
 1 deny ip any host 5.6.7.8
 2 permit ip any host 1.2.3.4
ip access-list resequence test 10 10
```

Scenario for ITAND-S01:

Scenario: /var/opt/yce/configs/ITAND-S01.scn
Included task: Command_job
Included task: checks

Parsing scenario:

```
1 [parameters]
2 acl_name = test
3 permit_ip = 1.2.3.4
4 deny_ip = 5.6.7.8
5 client_type = ZDE
6 change_id =
7 node = ITAND-S01
8 node_name = ITAND-S01
9 verbose = -v
10 [scenario]
11 Description ITAND-S01 Command job...
12 Description ITAND-S01 Command job
13 <commands> := "show ip int brief"
14 <commands> += "show cdp neigh | b Device"
15 <commands> += "show access-list"
16 if <a> != 1
17     foreach <command> in <commands>
18         <%cmd> := Parse_cmd -n ITAND-S01 -r "<command>" -t all_output
19         <all_output> = <all_output>%cmd>
20         config_create -n ITAND-S01 -f pre_checks.cfg -x <<EOT
            <all_output>
            EOT
21     endeach
```

Scenario

In the *Scenario* input field you may specify parameters and scenario syntax.

[Parameters]

The parameters section can only hold variables “variable = value”. Where value may be surrounded by quotes. It will hold anything till the end of the line (see the screenshot).

It may refer to another variable as well, though if not found it won't be substituted.

The *parameter* section isn't mandatory.

Scenario: Evaluate

```

1  [parameters]
2  var = value
3  var1 = val ue
4  var2 = <var>
5
6
7  [scenario]
8  Description <node> Command job...
9  task = Command_job
10
11 end

```

Scenario The scenario section can hold any syntax specified by the [Scenario syntax and commands](#). This could either be specified here directly or may be chained using the *task := command*.

The default, shown in the screenshot (for the scenario part), sets a description for the job and uses the scenario called “Command job”. The contents can be seen under [Operate > Scenarios](#)

If the scenario reaches an “end”, the job is considered successful. If it reaches a “stop”, the job result is “aborted”.

Scheduling

Once you are ready to schedule the job, you may choose to do it right away or plan it for the future. For more details on the [distributed scheduler](#).

Once scheduled, each job will be given a Job ID. The details of the job can be found in [Jobs](#) and [Job logs](#).

If you are using a multi-server architecture, you may choose the specific server or use your scheduler rules to choose it automatically. See the chapter [Distributed Scheduler](#).

To set up a dependency on Change ID's or approvals, the [Job Configuration](#) chapter will assist.

Schedule:

tomorrow

▼

at

5

▼

05

▼

or

☒ now

change-id:

C000xxxxxx

▼

Schedule

-auto-

▼

server

☒ verbose log

Cancel

For each of the nodes selected (use the checkboxes at the top of the form), a separate job will be scheduled.

Push Command jobs

Please wait until scheduling is completed

JobID	State	Server	Queue	Scheduled time	Job description
IT002-S50					
0502_0003	SCHEDULED	nyeve	yce	Tue 03-May-2022 05:05:00	IT002-S50 Command job... auto scheduler(s): nyeve default rule - Vendor_type: 'Cisco_IOS'
IT002-S51					
0502_0005	SCHEDULED	nyeve	yce	Tue 03-May-2022 05:05:02	IT002-S51 Command job... auto scheduler(s): nyeve default rule - Vendor_type: 'Cisco_IOS'
IT002-S52					
0502_0007	SCHEDULED	nyeve	yce	Tue 03-May-2022 05:05:04	IT002-S52 Command job... auto scheduler(s): nyeve default rule - Vendor_type: 'Cisco_IOS'
ITAND-S01					
0502_0009	SCHEDULED	nyeve	yce	Tue 03-May-2022 05:05:06	ITAND-S01 Command job... auto scheduler(s): nyeve default rule - Vendor_type: 'Cisco_IOS'

Back

From:
<https://labs-wiki.netyce.com/> - Technical documentation

Permanent link:
https://labs-wiki.netyce.com/doku.php/menu:operate:new_jobs:yce_jobs

Last update: 2022/05/02 08:09

