

Job Scheduler

Presenters

- Yash, TME

aruba

a Hewlett Packard
Enterprise company



The background features a solid red circle in the top-left corner and a large, dark blue shape with a white dotted pattern that occupies the right and bottom portions of the frame.

Overview

Job Scheduler

10.8 - Overview

- The Job Scheduler enables you to execute batches of CLI commands on a user-configured schedule or interval.
- A Job is a list of cli commands (on both configure and enable node) which are to be executed when the job is triggered by a schedule. In short, Job - **What needs to be executed**.
- A Schedule is a config mentioning the time when a trigger must occur and the set of jobs that need to be executed when trigger occurs. In short, Schedule – **When needs to be executed**.
- Feature supported on CX platforms.
- Use cases: To Timer based ACL, Telemetry Data, toggles a port, Switch reboots, QoS policy changes, system health status checks, statistics clearing, clean-up, and saving the running configuration.

In the config context (config)#:

```
job <JOB-NAME>
no job <JOB-NAME>
```

Subcommands available In the job config context

```
(config-job):
[no] enable (default is enable)
[no] desc <DESCRIPTION>
[no] [<SEQ-NUM>] [delay <DELAY>] cli <COMMAND>
resequence <START-SEQ-NUM> <INCREMENT>
```

In the config context (config)#:

```
[no] schedule <SCHEDULE_NAME> [ transient ]
[no] enable (default is enable)
[no] [<SEQ_NO>] job <job_name>
[no] trigger every {days <1-365> | hours <1-8760> | minutes <30-525600>}
[count <1-1000> ] [from [HH:]MM [YYYY-MM-DD] ]
[no] trigger on [HH:]MM { daily | weekly <1-7> | monthly <1-31> } [count
<1-1000>] [from YYYY-MM-DD]
[no] trigger at [HH:]MM [YYYY-MM-DD]
[no] desc <DESCRIPTION>
resequence <START> <INCREMENT>
```

Job Scheduler example: Timer ACL

– Timer ACL

- Time-based ACLs enables or disables an access control list or a specific access control entry on a particular schedule. The ACLs could be enabled/disabled on a specific calendar time or a periodic schedule
- For example: enable/disable on YYYY-MM-DD HH:MM daily, weekly, weekends Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

```
Job timer_acl_enable

enable

10 cli config

20 cli access-list ip myACL

Job timer_acl_disable

enable

10 cli config

20 cli no access-list ip myACL


schedule timer_acl_start

trigger on 20:00 weekly Fri

job timer_acl_enable

schedule timer_acl_end

trigger on 06:00 weekly Mon

job timer_acl_disable
```

Job Scheduler example: Telemetry Switch data

– Telemetry data

- Telemetry data are used by Net Insight running in Aruba Central for analyzing the switch and provide meaningful insight to the administrator.
- The switch needs to send the telemetry data periodically to the Net Insight.
- Job scheduler can be used to periodically collect these telemetry data and store the output in the job history file.
- These Job history files are accessible over the REST interface and the Net Insight can use these interface to collect the data from the switch.

Job telemetry_data

```
enable

10 cli show system

20 cli show system resource-utilization

30 cli show environment

40 cli show logging -r -s warn -n 10

schedule collect_periodic_telemetry

    trigger every hour 2

    job telemetry_data
```


Know about Job

- What needs to be executed.
- A maximum of 20 commands can be used in a job.
- Maximum number of jobs configurable in a system vary from **platform to platform**, to see the maximum number of jobs and job execution output use command `show capacities job` (See below).
 - Example: 6300 support 32 job configuration and preserve 10 execution output.
- Only non-interactive CLIs/commands are supported in a job. Commands with simple (y/n) prompts (example like boot system) will be automatically confirmed with "y." and supported as part of 10.8
- Commands requiring more complex user input (such as password change) are not supported as part of 10.8.
- Deleting a job also removes it from any schedule that uses the job, preventing further attempts to execute the job.
- Jobs must complete execution within 5 minutes, else job will be force-stopped after 5 minutes.
- Jobs are always executed in sequence.
- Same job can't be executed less than 30 minutes.
- If Job is going to take more than 5 minutes, then configure multiple jobs and add them to scheduler.
- If a given date is not available on that month (say 31st or Feb 29), then it will be executed on last day of the month.
- Each job will have a corresponding job history folder (`/var/log/job_history/<job_name>_[1-10].txt.gz`)
- The job history files will be preserved in the `/fs/logs/bootX` folder during switch reboot.



Know about Schedule

- When it needs to be executed.
- By default, all schedule is persistent (stays after reboot). We support transient schedule; transient causes the schedule to be cleared upon switch reboot. By default, schedules are maintained after switch reboots.
- A maximum of 10 jobs per schedule, each job can have 20 commands, to see the maximum number of schedule and job configurable in schedule, use command `show capacities job` (see below).

Note: Configure jobs before configuring the trigger command.

A schedule can be of 3 types:

1. **Periodic:** trigger occurs every X minutes/hours/days
 - Example: trigger every minutes 30 start 09:26 2021-04-06
2. **Calendar:** trigger occurs at time HH:MM on days d1,d2,d3 ('of a week' OR 'of a month')
 - Example: trigger on 12:30 monthly 12,31 start 2021-04-06
3. **Oneshot:** trigger occurs only once at the specified time & date. Then it expire.
 - Example: trigger at 23:45 2021-07-03



Job Schedule Simple Example

Port bring-up @ 08:03

```
job j1
5 cli show run interface 1/1/1
8 cli conf t
10 cli interface 1/1/1
20 cli no shutdown
30 delay 1 cli show run interface 1/1/1
40 cli end

schedule s1
10 job j1
trigger at 08:03 2021-08-17

job j2
5 cli show run interface 1/1/1
8 cli conf t
10 cli interface 1/1/1
20 cli shutdown
30 delay 1 cli show run interface 1/1/1
40 cli end

schedule s2
10 job j2
trigger every minutes 30 start 08:04 2021-08-17

job j3
5 cli show run interface 1/1/1
8 cli conf t
10 cli interface 1/1/1
20 cli no shutdown
30 delay 1 cli show run interface 1/1/1
40 cli end

schedule s3
10 job j3
trigger on 08:05 monthly 28 start 2021-08-17
```

```
6300# show job j1 execution-output 1
=====
Command: show run interface 1/1/1
time: Wed Jul 28 08:03:05 2021
=====
interface 1/1/1
    shutdown
    no routing
    vlan access 1
    exit
=====
Command: conf t
time: Wed Jul 28 08:03:05 2021
=====
Command: interface 1/1/1
time: Wed Jul 28 08:03:05 2021
=====
Command: no shutdown
time: Wed Jul 28 08:03:05 2021
=====
Command: show run interface 1/1/1
time: Wed Jul 28 08:03:06 2021
=====
interface 1/1/1
    no shutdown
    no routing
    vlan access 1
    exit
=====
Command: end
time: Wed Jul 28 08:03:06 2021
=====
6300#
```


Job Scheduler Examples

Create a port toggle job and then schedules the job for execution on Monday and Friday night at 11:45 PM, starting on Aug 2, 2021, with a one-year duration.

Create a Job **PTog1**

```
CXSwitch(config)# job PTog1
CXSwitch(config-job-PTog1)# desc Toggle port 1/1/1
CXSwitch(config-job-PTog1)# 10 cli config
CXSwitch(config-job-PTog1)# 20 cli interface 1/1/1
CXSwitch(config-job-PTog1)# 30 cli shutdown
CXSwitch(config-job-PTog1)# 40 delay 10 cli no shutdown
CXSwitch(config-job-PTog1)# 50 cli end
CXSwitch(config-job-PTog1)# exit
```

Create a Schedule **PT2xW**

```
CXSwitch(config)# schedule PT2xW
CXSwitch(config-schedule-PT2xW)# desc Monday & Friday 11:45 PM port
toggles
CXSwitch(config-schedule-PT2xW)# 10 job PTog1
CXSwitch(config-schedule-PT2xW)# trigger on 23:45 weekly 2,6 count
104 start 2021-08-02
CXSwitch(config-schedule-PT2xW)# exit
```

Create a job to Reboot a switch, that saves the running configuration and then reboots the switch, on the last day of the month at 3:00 AM, starting on January 31 2022, with a two-year duration.

Create a Job **Reboot_sw1**

```
switch(config)# job Reboot_sw1
switch(config-job-Reboot_sw1)# desc Save config then
reboot switch
switch(config-job-Reboot_sw1)# 10 cli config
switch(config-job-Reboot_sw1)# 20 cli write memory
switch(config-job-Reboot_sw1)# 30 cli boot system
switch(config-job-Reboot_sw1)# exit
switch(config)#
```

Create a Schedule **RB_LDM**

```
switch(config)# schedule RB_LDM
switch(config-schedule-RB_LDM)# desc Monthly reboot 3:00 AM
switch(config-schedule-RB_LDM)# 10 job Reboot_sw1
switch(config-schedule-RB_LDM)# trigger on 3:00 monthly 31 count
24 start 2022-01-31
switch(config-schedule-RB_LDM)# exit
```

Job Scheduler

Showing the port toggle job and schedule information

```
CXSwitch# show job PTog1
Job Name : PTog1 Enabled : Yes
Description : Toggle port 1/1/1
Status : waiting
Number of commands : 5
Total execution count : 1
Failed execution count : 0

Job execution history
-----

Instance number : 1
Execution status : success
Execution start time : Mon Aug 2 23:45:00 2021
Execution duration : 10s

Job CLI commands
-----
10 cli config
20 cli interface 1/1/1
30 cli shutdown
40 delay 10 cli no shutdown
50 cli end
```

```
CXSwitch# show schedule PT2xW
Schedule Name: PT2xW
Schedule config
-----
Description : Monday & Friday 11:45 PM port toggles
Enabled : Yes
Trigger type : calendar
Transient : No
Max trigger count : 104
Trigger start date : 2021-08-02 23:45
Schedule Status
-----
Trigger status : active
Next trigger time : Fri Aug 6 23:45:00 2021
Triggered count : 1
Scheduled Jobs
-----
10 : PTog1
```

```
CXSwitch# show job PTog1 execution-output 1
```

```
6300M-7-VSF# show events -r -d schedulerd
```

Switch Reboot Simple Job Scheduler Example

Create a Job and Schedule as below

```
job 1
  10 cli boot system
schedule 1 transient
  10 job 1
  trigger at 08:05 2021-07-28
```

```
CXSwitch# show boot-history
Management module
=====
```

```
Index : 1
Boot ID : 4454f1d214724bf6bcde59f1645d6d64
Current Boot, up for 7 mins 26 secs
```

```
Index : 0
Boot ID : 292846f21c744e4aa4d17f2c00defb9a
28 Jul 21 08:05:19 : Reboot requested by user
CXSwitch#
```

```
CXSwitch# show events -a -d schedulerd
```

```
-----
Event logs from previous boots
-----
```

```
2021-07-28T08:03:27.795858+00:00 8320 schedulerd[2054]: Event|12201|LOG_INFO|AMM|1/1|Creating schedule 1, trigger time(s): 2021-07-28 08:05.
2021-07-28T08:05:14.405135+00:00 8320 schedulerd[2054]: Event|12202|LOG_INFO|AMM|1/1|Schedule 1 triggered, trigger_count: 1
-----
```

```
Event logs from current boot
-----
```

Note: Transient schedule (Schedule is not persistent across the reboot). By default, a schedule is persistent across the reboot

Schedule Trigger examples

Examples	CLI Syntax
1pm on every monday and thursday	trigger on 13:00 weekly 2,4 start 2021-08-17
On 22th Aug 2021 10:00am	trigger at 10:00 2021-08-22
Daily at 10:00am	trigge on 10:00 daily start 2021-08-17
Last day of every month at 6:45am	trigger on 06:45 monthly 31
30th of every month at 7.30am	trigger on 07:30 monthly 30
29th of every month at 3.15am	trigger on 03:45 monthly 29
28th of every month at 8pm	trigger on 20:00 monthly 28
Once in every 5 hours	every hour 5
Every 45 minutes from now	every minutes 45
Every alternative day from now	every day 2

Scale: Job and Schedule

- The capacities command provides the information about the maximum number of jobs and schedules that can be configured in the system. It also provides information about the maximum number of jobs that can be configured under one schedule and the maximum number of job execution output that are preserved per job

```
8320# show capacities job
System Capacities: Filter Job
Capacities Name                                     Value
-----
Maximum number of job execution output preserved per job 10
Maximum number of jobs configurable in a system          32
```

```
8320# show capacities schedule
System Capacities: Filter Schedule
Capacities Name                                     Value
-----
Maximum number of jobs configurable in a schedule        10
Maximum number of schedules configurable in a system     32
```


Event Logs

Daemon	Event ID	Severity	Message	Description
schedulerd	12201	Info	'Creating schedule {name}, trigger time(s): {start_datetime}{details}'	This event log appears when the trigger command is configured for a schedule.
schedulerd	12202	Info	'Schedule {name} triggered, trigger_count: {trigger_count}'	This event log appears when a schedule triggers at the configured time.
schedulerd	12203	Info	'Timezone changed. Re-creating schedule {name}, trigger time(s): {start_datetime}{details}'	This event log appears when timezone is updated on the switch.

10.8 Job Scheduler Feature Support

Supported as part of 10.8	Not Supported as part of 10.8
<ul style="list-style-type: none">• CLI Config commands• CLI Enable node commands (no password prompt)• Switch/Module/Member reboot• Redundancy switchover• Diagnostic data collection• Packet capture	<ul style="list-style-type: none">• “copy” commands• show tech• show job• show boot-history• show core-dump• sleep• terminal-monitor• repeat• start-shell• User defined syslog/event can be trigger.• Any CLI command that involves user interaction or that could potentially take more than 5 minutes of time should not be added as part of job.• Nested Jobs

The background features a solid dark blue field. In the top-left corner, a large red circle is partially visible. A large, irregular shape in the center-right is filled with a pattern of small, light blue dots, creating a halftone effect. The word "Demo" is written in white, bold, sans-serif font, positioned over the red circle.

Demo



a Hewlett Packard
Enterprise company

Thank you

yashavanth.n.n@hpe.com