

Scheduling jobs through events

Starting from version 8.2.2 there will be the option available to trigger a job after a syslog event. There are a few settings you need to take into account, but after setting it up it should work automatically.

Set-up

First of all we need to add the following triggers to `etc/yce_events.conf`:

```
#
type=SingleWithSub
ptype=RegExp
pattern=syslog_received_for_(\S+)
sub=create_job
arg=$1
desc=$0
action=write - normal save OR save main node $1 matches.
#
```

The key is the `sub=create_job`: this calls the `create_job` function of the `yce_events` daemon, and this schedules a job for the node. The node in question is put in the `(\S+)` of the pattern, which gets pushed to the variable `$1`. You can customize the pattern to the syslog message you need, as long as the hostname corresponds to `(\S+)`.

This triggers a job that contains a task with scenario "Syslog_task". This task is automatically provided by `netyce`. You can edit this scenario as you see fit for your needs.

To trigger the job, you need a user. The name of this user is 'xch'. This user is automatically generated by `netYCE`, but you need to change the user group to a user group you see fit. This user group at least needs operator rights.

As soon as you have updated your `yce_events.netyce.conf` and set the permissions for the `xch` user, this should work.

Using interfaces as a second argument

If you want to send along a second argument, the way to do it is as follows:

```
# type=SingleWithSub ptype=RegExp pattern=syslog_received_for_(\S+)_(\S+) sub=create_job
arg=$1 arg2=$2 desc=$0 action=write - normal save OR save main node $1 matches. #
```

Arg1 contains the hostname, Arg2 contains the interface name. This will end up in the job as a parameter called "interface", which can then be used in your scenario as you please.

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

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:reference:system_events:jobs

Last update: **2024/07/03 12:31**

