

Opensource ICT Solutions

ZABBIX

Germany
conference 2025

Zabbix and Netconf



Opensource ICT Solutions

Who am I?



Nathan Liefing
Zabbix Consultant / Trainer

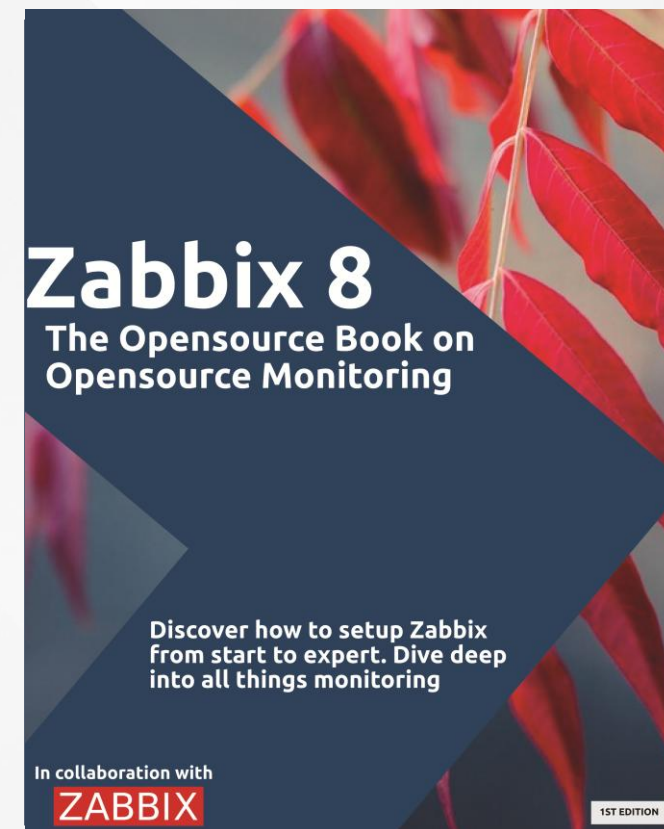
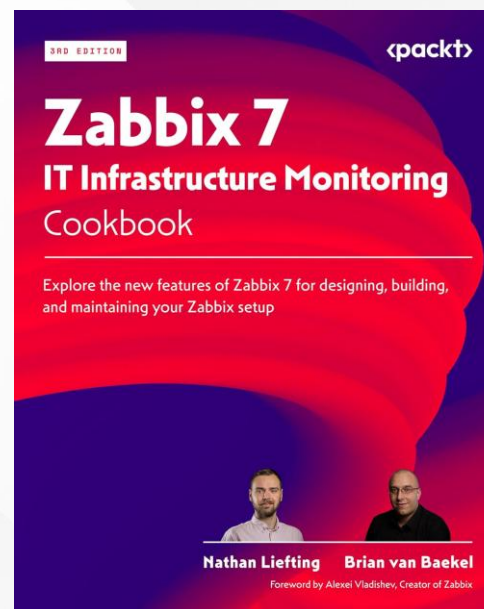
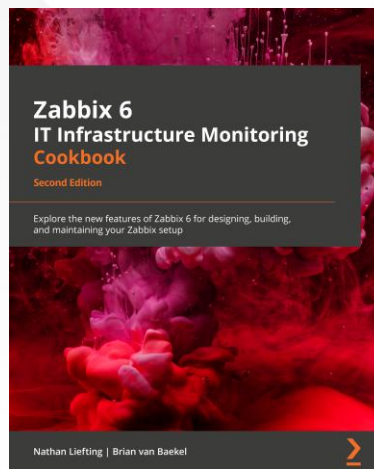
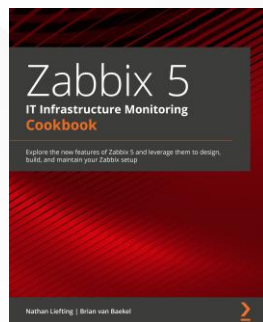


Opensource ICT Solutions

- Zabbix support
- Zabbix training
- Zabbix consultancy
- And more...



<https://www.linkedin.com/company/opensource-ict-solutions/>



How to monitor a network device

- 3 options:
 - SNMP
 - Very well known protocol (old: v1 1989)
 - Various versions / options
 - The S is a big lie
 - RFCs are too open -> complex implementations from vendors
 - Netconf
 - Introduced late 2006 (RFC 4741), revised in 2011(RFC 6241)
 - Using RPC
 - XML format (and later JSON)
 - Built on top of SSH
 - Streaming telemetry



Zabbix?

- SNMP
 - Was available for years already
 - Improvements in 6.4 (walk/get options)
 - Even better improvements in 7.0 (async polling)
- Still limitations mainly due to how SNMP is built by the vendor



Since 7.2: Netconf support!

- Template: Juniper MX by SNMP

Good morning!

I wanted to ask you for advice

While studying and searching for data on Juniper, I learned that you can receive data not only via SNMP but also netconf, how convenient do you think it is for the user to use this method.

As an engineer, I find it more convenient to use commands in netconf with their subsequent processing
for exmpl

But the Juniper Mist managed devices (so usually EX and SRX equipment) use netconf as well, so its get...

I want our developers to implement the ability to use ssh with the -s netconf parameter



- Which led to <https://support.zabbix.com/browse/ZBXNEXT-9588>

Title: SSH item subsystem support

Summary

Currently there is no SSH subsystem support (SSH flag "-s") for SSH items.

We need to introduce SSH subsystem support to allow to use netconf over ssh – possibility to monitor devices which support only netconf.

Use case

As a user I want SSH subsystem support for SSH items (identical to SSH flag "-s")

- to connect to devices which supports netconf or similar
- limit access to one command only (configured in SSH subsystem)
- streamline SSH command execution across various monitoring targets
(e.g. backup would resolve to /opt/backup.sh on one system or c:\scripts\backup.cmd on another)



What is Netconf?

- Basically: open a connection via SSH and use one of the 'base operations' to do something
- Device replies with requested information
- Not in Zabbix (yet?): Event notifications



Netconf base operations

- ***<get>***
 - Retrieves all or part of the information about the running configuration and device state.
- ***<get-config>***
 - Retrieves all or part of the configuration information available from a specified configuration datastore.
- ***<edit-config>***
 - Submits all or part of a configuration to a target configuration datastore.



Netconf base operations 2

- **<copy-config>**
 - Creates or replaces a target configuration datastore with the information from another configuration datastore.
- **<delete-config>**
 - Deletes a target configuration datastore, but only if it's not running.
- **<lock>**
 - Locks a target configuration datastore, unless a lock already exists on any part of that datastore.
- **<unlock>**
 - Releases a lock on a configuration datastore that was previously locked through a <lock> operation.



Netconf base operations 3

- **<close-session>**
 - Requests the NETCONF server to gracefully terminate an open session.
- **<kill-session>**
 - Forces a session's termination, causing current operations to be aborted.



How to build this in Zabbix

Item

Item

Tags

Preprocessing 3

* Name

Software information

Type

SSH agent

* Key

ssh.run[software-information,169.254.1.2,830,,,netconf]

Select

Type of information

Text

Host interface

None

Authentication method

Password

* User name

{NETCONF.USERNAME}

Password

{NETCONF.PASSWORD}

* Executed script

```
<rpc>
  <get-software-information/>
</rpc>
<rpc>
  <close-session/>
</rpc>
]]>]]>
```

* Update interval

1h



Result

```
<!-- No zombies were killed during the creation of this user interface -->
<!-- user root, class super-user -->
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
  </capabilities>
  <session-id>22979</session-id>
</hello>
]]>]]>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:junos="http://xml.juniper.net/junos/24.2R1-S1.10/junos">
  <software-information>
    <host-name></host-name>
    <product-model>mx204</product-model>
    <product-name>JNP204 [MX204]</product-name>
    <junos-version>24.2R1-S1.10</junos-version>
    <package-information>
      <name>os-kernel</name>
      <package-name>os-kernel-prd-x86-64-20240711.b8a9aee_builder_bsd15_242</package-name>
      <comment>JUNOS OS Kernel 64-bit [20240711.b8a9aee_builder_bsd15_242]</comment>
    </package-information>
```



New MX template by Netconf

- 4 SSH items
- That's it.

≡

Items

All hosts / MX - netconf template

Enabled

ZBX

Items 918

Triggers 129

Graphs 7

Discovery rules 12

Web scenarios

<input type="checkbox"/>	Name ▲	Triggers	Key
<input type="checkbox"/>	... Juniper MX by NETCONF: DOM: Get data		ssh.run[JuniperMxDom,{JUNIPER.NETCONF.IP},{JUNIPER.NETCONF.PORT},,,,netconf]
<input type="checkbox"/>	... Juniper MX by NETCONF: Interface information: Get data		ssh.run[JuniperMxInterface,{JUNIPER.NETCONF.IP},{JUNIPER.NETCONF.PORT},,,,netconf]
<input type="checkbox"/>	... Juniper MX by NETCONF: Resource: Get data		ssh.run[JuniperMxResource,{JUNIPER.NETCONF.IP},{JUNIPER.NETCONF.PORT},,,,netconf]
<input type="checkbox"/>	... Juniper MX by NETCONF: Routing protocols: Get data		ssh.run[JuniperMxBgpOspf,{JUNIPER.NETCONF.IP},{JUNIPER.NETCONF.PORT},,,,netconf]



1: Master item

<input type="checkbox"/>	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
<input type="checkbox"/>	*** Juniper MX by NETCONF: Resource: Get data		ssh.run[JuniperMxResource,{JUNIPER.NETCONF.IP},{JUNIPER.NETCONF.PORT},,,,netconf]	0	0		SSH agent	Enable

```
{
  "results": [
    {
      "hello": {
        "capabilities": {
          "capability": [
            "urn:ietf:params:netconf:base:1.0",
            .....
          ]
        },
        "session-id": "11089"
      }
    },
    {
      "rpc-reply": {
        "fpc-information": {
          "@style": "brief",
          "fpc": {
            "slot": "0",
            "state": "Online",
            "temperature": {
              "@celsius": "0",
              "#text": "N/A"
            }
          }
        }
      }
    }
  ]
}
```



2: Dependent item

<input type="checkbox"/>	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status	Tag
<input type="checkbox"/>	... Juniper MX by NETCONF: Resource: Get data Alarm: Get data		juniper.mx.alarm.data		31d		Dependent item	Enabled	co

```
{
  "results":[
    ...
    {
      "rpc-reply":{
        "alarm-information":{
          "alarm-summary":{
            "active-alarm-count":"1"
          },
          "alarm-detail":{
            "alarm-time":{
              "@seconds":"1738561515",
              "#text":"2025-02-03 05:45:15 UTC"
            },
            "alarm-class":"Major",
            "alarm-description":"PEM 0 Not Powered",
            "alarm-short-description":"PEM 0 No Power",
            "alarm-type":"Chassis"
          }
        }
      }
    }
  ],
}
```



```
[
  {
    "alarm-detail":{
      "alarm-description":"PEM 0 Not Powered",
      "alarm-time":{
        "@seconds":"1738561515",
        "#text":"2025-02-03 05:45:15 UTC"
      },
      "alarm-type":"Chassis",
      "alarm-short-description":"PEM 0 No Power",
      "alarm-class":"Major"
    },
    "alarm-summary":{
      "active-alarm-count":"1"
    }
  }
]
```



3: LLD rule

<input type="checkbox"/>	Host	Name ▲	Items	Triggers	Graphs	Hosts	Key	Interval	Type
<input type="checkbox"/>	MX - netconf template	Juniper MX by NETCONF: Alarm: Get data: Alarms discovery	Item prototypes 2	Trigger prototypes 2	Graph prototypes	Host prototypes	juniper.mx.alarm.discovery		Depe

```
[
{
  "alarmDescription":"PEM 0 Not Powered",
  "alarmClass":"Major",
  "alarmType":"Chassis",
  "alarmName":"PEM 0 No Power"
}
```

LLD macros

LLD macro	JSONPath	
<input data-bbox="1411 605 1852 648" type="text" value="{#ALARM_CLASS}"/>	<input data-bbox="1862 605 2390 648" type="text" value="\$alarmClass"/>	Remove
<input data-bbox="1411 658 1852 701" type="text" value="{#ALARM_DESCR}"/>	<input data-bbox="1862 658 2390 701" type="text" value="\$alarmDescription"/>	Remove
<input data-bbox="1411 711 1852 753" type="text" value="{#ALARM_NAME}"/>	<input data-bbox="1862 711 2390 753" type="text" value="\$alarmName"/>	Remove
<input data-bbox="1411 763 1852 806" type="text" value="{#ALARM_TYPE}"/>	<input data-bbox="1862 763 2390 806" type="text" value="\$alarmType"/>	Remove
Add		
<div><div>Update</div><div>Clone</div><div>Test</div><div>Delete</div><div>Cancel</div></div>		



4: Items as per prototypes

<input type="checkbox"/>	Name ▲	Key	Interval	History	Trends	Type	Create enabled	Discover	Tags
<input type="checkbox"/>	... Juniper MX by NETCONF: Alarm: Get data: Alarm [{#ALARM_NAME}]: Get data	juniper.mx.alarm.get.data["{#ALARM_NAME}"]		0		Dependent item	Yes	Yes	component: ra
<input type="checkbox"/>	... Juniper MX by NETCONF: Alarm [{#ALARM_NAME}]: Get data: Alarm [{#ALARM_NAME}]: Severity	juniper.mx.alarm.severity["{#ALARM_NAME}"]		31d	365d	Dependent item	Yes	Yes	class: {#ALAR



<input type="checkbox"/>	Name ▲	Triggers	Key
<input type="checkbox"/>	... Alarms discovery: Alarm: Get data: Alarm [PEM 0 No Power]: Get data		juniper.mx.alarm.get.data["PEM 0 No Power"]
<input type="checkbox"/>	... Alarms discovery: Alarm [PEM 0 No Power]: Get data: Alarm [PEM 0 No Power]: Severity	Triggers 2	juniper.mx.alarm.severity["PEM 0 No Power"]



Why use Netconf over SNMP?

- SSH so secure by default
- Way, waaaaay more efficient than SNMP ever will be
- More info is exposed
- Easier to correlate different entities
- Structured data; easier to parse



Contact



Opensource ICT Solutions B.V.

Tabakvest 87/4490
2000 Antwerpen
Belgium
T. +32 (0)3 377 69 36
E. info@oicts.be
W. <https://oicts.be>

Tappersweg 14-024
2031EV Haarlem
The Netherlands
T. +31 (0) 72 743 65 83
E. info@oicts.nl
W. <https://oicts.nl>

Opensource ICT Solutions LTD

2 Leman Street
London E1W 9US
United Kingdom
T. +44 (0) 20 4551 1827
E. info@oicts.co.uk
W. <https://oicts.co.uk>

Opensource ICT Solutions LLC

251 Little Falls Drive
Wilmington, DE 19808
United States
T. +1 (929) 377 1253
E. info@oicts.com
W. <https://oicts.com>



Opensource ICT Solutions