

ENGLISH

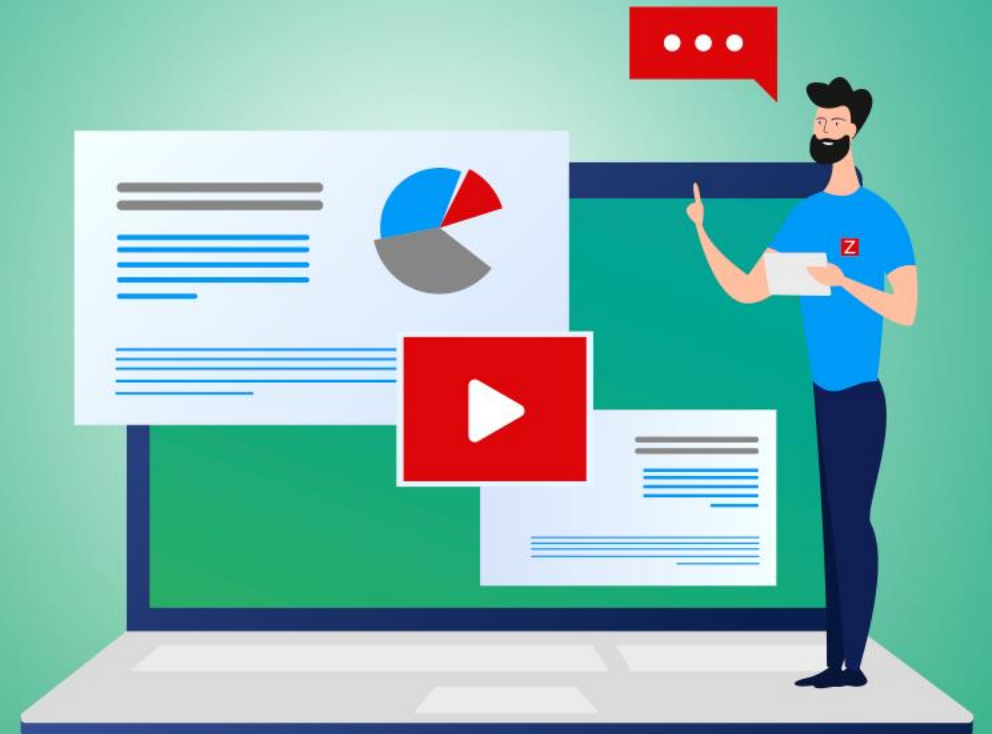


MEETUP ONLINE '21

BUILDING TEMPLATES FOR SNMP DEVICES

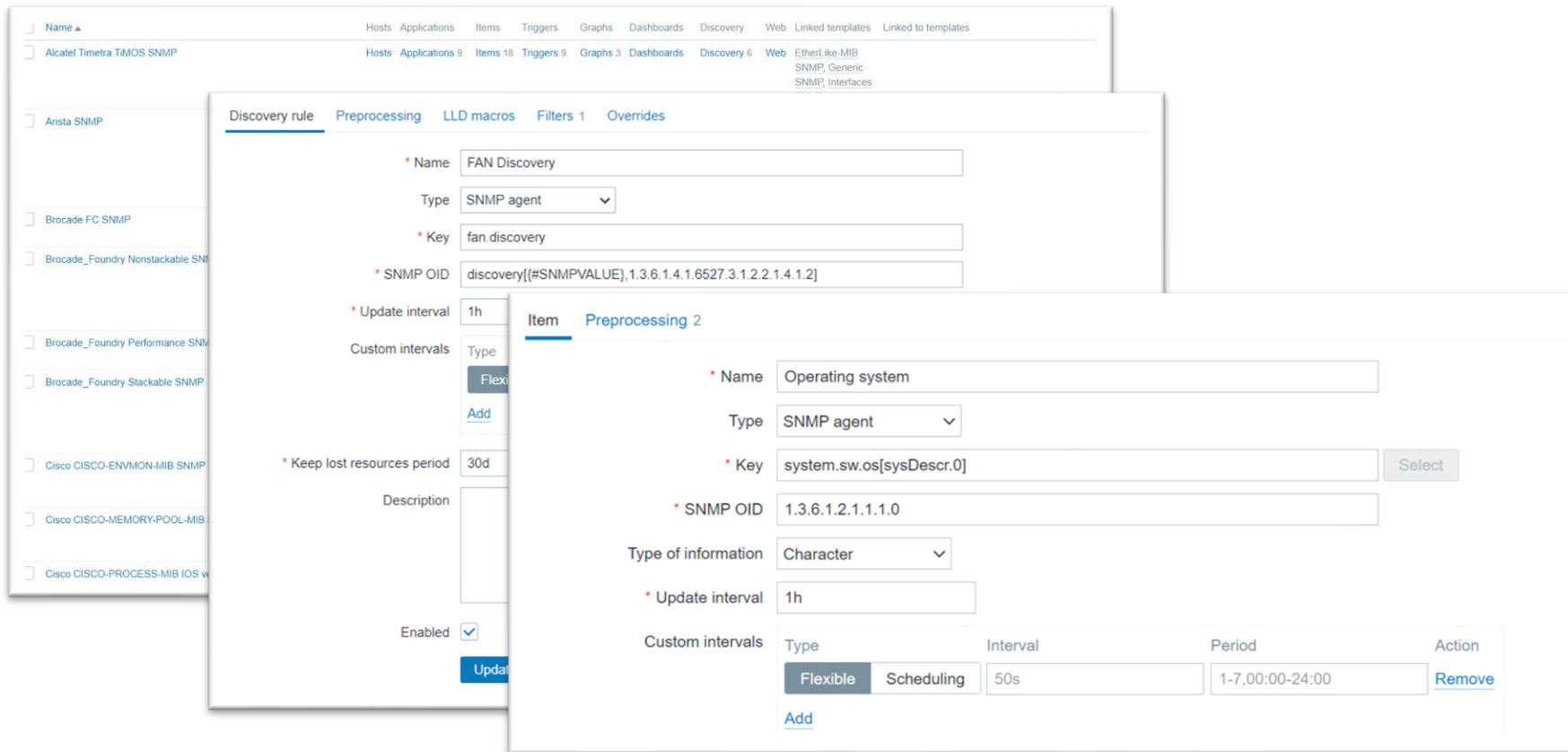
ARTŪRS LONTONS
TECHNICAL SUPPORT ENGINEER

ZABBIX



MONITORING SNMP AGENTS WITH ZABBIX

- ✓ Use built in or community templates
- ✓ Create items based on vendor documentation/snmpwalk results
- ✓ Use LLD to discover SNMP entities



The screenshot displays the Zabbix web interface configuration for SNMP agents. It is divided into two main sections: 'Discovery rule' and 'Item'.

Discovery rule configuration:

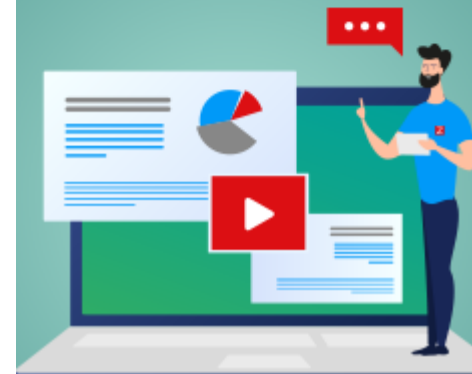
- Name: FAN Discovery
- Type: SNMP agent
- Key: fan.discovery
- SNMP OID: discovery([#SNMPVALUE],1.3.6.1.4.1.6527.3.1.2.2.1.4.1.2)
- Update interval: 1h
- Keep lost resources period: 30d
- Enabled:

Item configuration:

- Name: Operating system
- Type: SNMP agent
- Key: system.sw.os[sysDescr.0] (with a 'Select' button)
- SNMP OID: 1.3.6.1.2.1.1.1.0
- Type of information: Character
- Update interval: 1h

Custom intervals table:

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00



BUILDING CUSTOM SNMP TEMPLATES

I wish to build custom SNMP templates, but where do I start?

✓ Vendor documentation is your best friend! *(If available)*

coDeviceWirelessInterfaceStatusTable

.1.3.6.1.4.1.8744.5.25.1.2.1

not-accessible

Device wireless interface status attributes.

coDeviceWirelessInterfaceStatusEntry

.1.3.6.1.4.1.8744.5.25.1.2.1.1

not-accessible

An entry in the coDeviceWirelessInterfaceStatusTable. coDevDisIndex - Uniquely identifies a device in the MultiService Controller. coDevWirIfStaRadioIndex - Uniquely identifies a radio on the device.

coDevWirIfStaRadioIndex

.1.3.6.1.4.1.8744.5.25.1.2.1.1.1

not-accessible

Specifies the index of a radio on the device.

coDevWir

.1.3.6.1.4.1.

Link to co

coDevWir

.1.3.6.1.4.1.

The curre

coDevWir

.1.3.6.1.4.1.

Identifies

Table 2-9 scfCpuInfo(1.3.6.1.4.1.211.1.15.4.1.1.9)

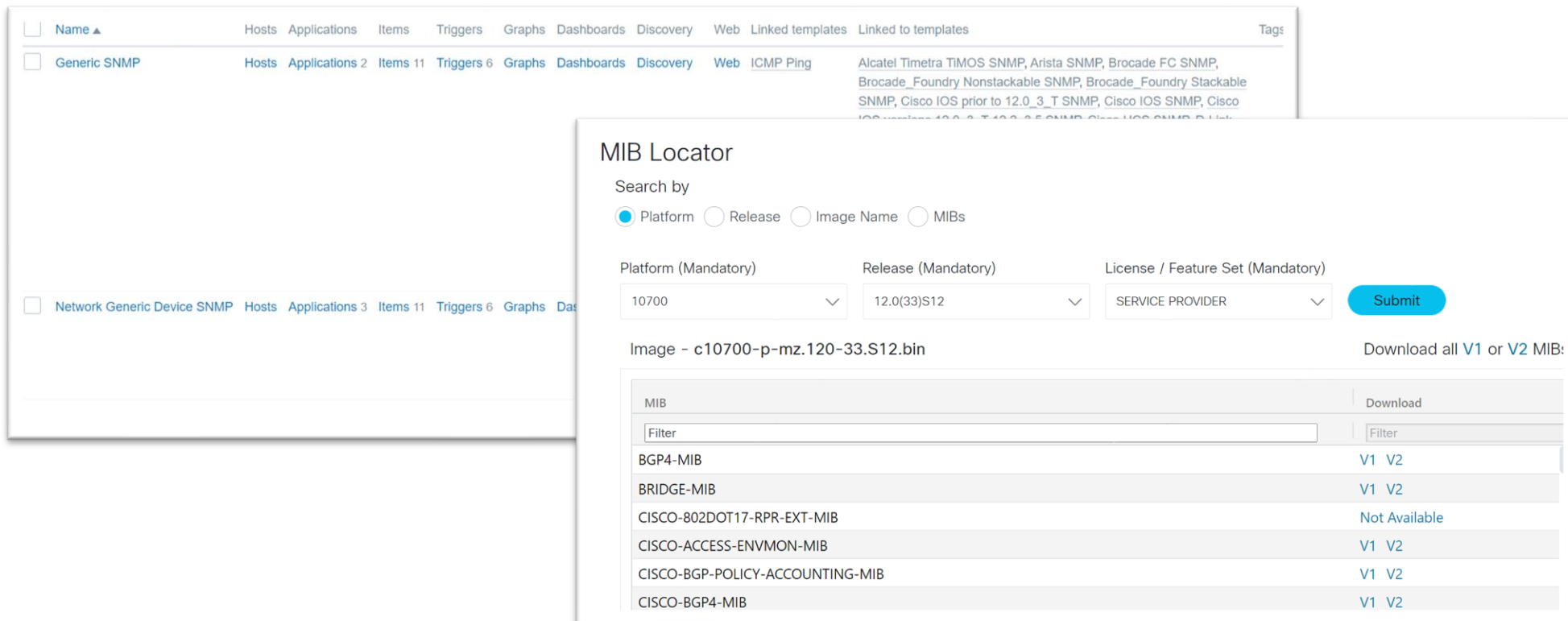
OBJECT-NAME	SUMMARY	OID	INDEX	SYNTAX	MAX-ACCESS
scfCpuNumber	CPU information count (table)	scfCpuInfo.1	.0	Integer32	ro
scfCpuTable	CPU information (table)	scfCpuInfo.2	-	SEQUENCE OF ScfMemoryEntry	na
scfCpuEntry	-	scfCpuTable.1	-	ScfMemoryEntry	na
scfCpuBoardType	Board identity	scfCpuEntry.1	.Parts identifier (*1)	ScfComponentType	ro
scfCpuBoardId	Board number	scfCpuEntry.2	.Parts identifier (*1)	ScfIndex	ro
scfCpuModuleType	Part identity	scfCpuEntry.3	.Parts identifier (*1)	ScfComponentType	ro
scfCpuModuleId	Part number	scfCpuEntry.4	.Parts identifier (*1)	ScfIndex	ro
scfCpuSubType	Part (sub) identity	scfCpuEntry.5	.Parts identifier (*1)	ScfComponentType	ro
scfCpuSubId	Part (sbu) number	scfCpuEntry.6	.Parts identifier (*1)	ScfIndex	ro
scfCpuType	CPU type name	scfCpuEntry.7	.Parts identifier (*1)	DisplayString	ro
scfCpuFrequency	CPU frequency	scfCpuEntry.8	.Parts identifier (*1)	Integer32	ro
scfCpuAdditionalInfo	CPU additional information	scfCpuEntry.9	.Parts identifier (*1)	DisplayString	ro
scfCpuMemoryMode	Memory mirror mode	scfCpuEntry.10	.Parts identifier (*1)	ScfMemoryMirroMode	ro
scfCpuState	CPU operating state	scfCpuEntry.11	.Parts identifier (*1)	ScfStateTC	ro

*1; For "Parts identifier" in INDEX, see "4.1 Parts identifier (OID)".



BUILDING CUSTOM SNMP TEMPLATES

- ✓ Check for vendor provided, vendor specific MIB files!
- ✓ Out of the box SNMP templates can also be used with general purpose MIBs!



The screenshot shows the Zabbix MIB Locator interface. The background displays a list of SNMP templates, including 'Generic SNMP' and 'Network Generic Device SNMP'. The foreground shows the 'MIB Locator' dialog box with the following details:

MIB Locator

Search by: Platform Release Image Name MIBs

Platform (Mandatory): 10700
Release (Mandatory): 12.0(33)S12
License / Feature Set (Mandatory): SERVICE PROVIDER

Image - c10700-p-mz.120-33.S12.bin

Download all V1 or V2 MIBs:

MIB	Download
BGP4-MIB	V1 V2
BRIDGE-MIB	V1 V2
CISCO-802DOT17-RPR-EXT-MIB	Not Available
CISCO-ACCESS-ENVMON-MIB	V1 V2
CISCO-BGP-POLICY-ACCOUNTING-MIB	V1 V2
CISCO-BGP4-MIB	V1 V2



BUILDING CUSTOM SNMP TEMPLATES

Okay, but how would I create a test environment for my templates?

- ✓ Misconfigured templates could cause a spike in SNMP requests on your device
- ✓ Sometimes the device is not directly accessible during the development

github.com/etingof/snmpsim

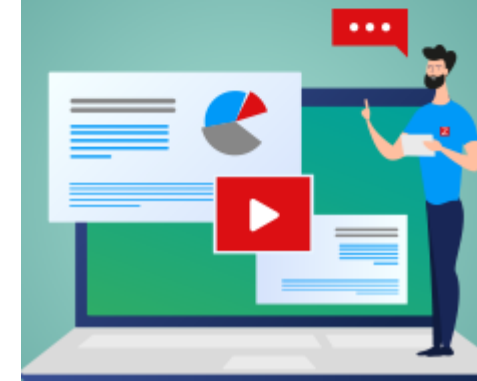
SNMP Simulator

pypi v0.4.7 | python 2.4 | 2.5 | 2.6 | 2.7 | 3.2 | 3.3 | 3.4 | 3.5 | 3.6 | 3.7 | build passing | license BSD

This is a pure-Python, open source and free implementation of SNMP agents simulator distributed under 2-clause BSD license.

Features

- SNMPv1/v2c/v3 support
- SNMPv3 USM supports MD5/SHA/SHA224/SHA256/SHA384/SHA512 auth and DES/3DES/AES128/AES192/AES256 privacy crypto algorithms
- Runs over IPv4 and/or IPv6 transports
- Simulates many EngineID's, each with its own set of simulated objects
- Varies response based on SNMP Community, Context, source/destination addresses and ports
- Can gather and store snapshots of SNMP Agents for later simulation
- Can run simulation based on MIB files, snmpwalk and sapwalk output
- Can gather simulation data from network traffic or tcpdump snoops
- Can gather simulation data from external program invocation or a SQL database
- Can trigger SNMP TRAP/INFORMs on SET operations
- Capable to simultaneously simulate tens of thousands of Agents
- Offers REST API based [control plane](#)
- Gathers and reports extensive activity metrics
- Pure-Python, easy to deploy and highly portable
- Can be extended by loadable Python engine



PREPARING THE PROPER TOOLS

Let's simulate an SNMP device!

First, we will need to make sure we have the proper tools:

- ✓ CentOS 8
- ✓ Zabbix 5.2
- ✓ Zabbix documentation (SNMP discovery)
- ✓ SNMPSIM software
- ✓ snmpwalk command results from our device
- ✓ Vendor documentation
- ✓ Device MIB files

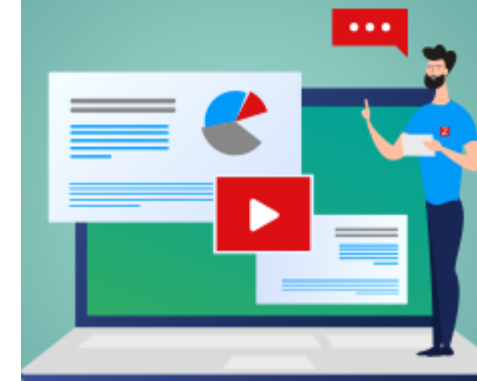


PERFORM THE SNMPWALK

- ✓ Perform snmpwalk on your device

```
[root@localhost ~]# snmpwalk -v2c -On -c Meetup
192.168.1.126
.1.3.6.1.2.1.1.1.0 = STRING: 1148VXP
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.664.1.1416
.1.3.6.1.2.1.1.3.0 = Timeticks: (813572029) 94 days,
3:55:20.29
.1.3.6.1.2.1.1.4.0 = STRING: www.adtran.com
.1.3.6.1.2.1.1.7.0 = INTEGER: 4
.1.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
.1.3.6.1.2.1.2.1.0 = INTEGER: 6160
.1.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
.1.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
..1.3.6.1.2.1.2.2.1.1.10001 = INTEGER: 10001
...
```

- ✓ Store the result in a separate file



INSTALLING SNMPSIM

- ✓ Install python

```
yum install python3
```

- ✓ Use pip (*package installer for Python*) to install snmpsim

```
pip3 install snmpsim
```

- ✓ Snmpsim won't run under elevated user permissions
- ✓ Create new user group and user accounts

```
groupadd snmpd  
useradd -g snmpd snmpd
```

- ✓ Create a directory for snmpwalk output and MIB file storage

```
mkdir -p /usr/share/snmpsim/data
```



RUN SNMPSIM

- ✓ Run snmpsim and pass the listen IP/Port

```
snmpsimd.py --agent-udp4-endpoint=192.168.1.126:1024
```

- ✓ The snmpwalk file name becomes the community name

```
Configuring /usr/share/snmpsim/data/192.168.1.126.raw.snmpwalk controller  
SNMPv1/2c community name: 192.168.1.126.raw  
SNMPv3 Context Name: 6bdad8c3906f65190f7c5f4674434a6c or 192.168.1.126.raw
```



TESTING SNMP SIM

- ✓ Let's see if we can snmpwalk the simulated device

```
[root@localhost ~]# snmpwalk -v2c -c '192.168.1.126.raw' 192.168.1.126:1024
SNMPv2-MIB::sysDescr.0 = STRING: 1148VXP
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.664.1.1416
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (813572029) 94 days,
3:55:20.29
SNMPv2-MIB::sysContact.0 = STRING: www.adtran.com
SNMPv2-MIB::sysName.0 = STRING: WINF-OKHR
SNMPv2-MIB::sysLocation.0 = STRING: FM 946 SOUTH @ WINFREY RD
SNMPv2-MIB::sysServices.0 = INTEGER: 4
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
IF-MIB::ifNumber.0 = INTEGER: 6160
IF-MIB::ifIndex.1 = INTEGER: 1
```

- ✓ Don't forget to specify the proper community, IP and port!



TESTING SNMPSIM FROM ZABBIX

ENGLISH



MEETUP ONLINE '21

✓ Now let's try and create a host in Zabbix

The screenshot shows the Zabbix web interface for adding a new host. The 'Host' tab is selected. The form contains the following fields and options:

- Host name:** SNMP device
- Visible name:** SNMP device
- Groups:** SNMP/Meetup (with a 'Select' button and a search input 'type here to search')
- Interfaces:** A table with columns: Type, IP address, DNS name, Connect to, Port, Default.
 - Row 1: Type: SNMP, IP address: 192.168.1.126, DNS name: (empty), Connect to: IP, DNS, Port: 1024, Default: Remove (selected).
- SNMP version:** SNMPv2 (dropdown menu)
- SNMP community:** 192.168.1.126.raw
- Use bulk requests:**
- Description:** SNMP Host for Zabbix Meetup

✓ Once again – mind the IP, Port and community! (*Community = SNMPWalk file name!*)



TESTING SNMPSIM FROM ZABBIX

✓ Let's create an item and test if our SNMP device

Item Preprocessing 1

* Name	<input type="text" value="Port 7 incoming traffic"/>
Type	<input type="text" value="SNMP agent"/>
* Key	<input type="text" value="ifHCInOctets.103007"/> <input type="button" value="Select"/>
* Host interface	<input type="text" value="192.168.1.126 : 1024"/>
* SNMP OID	<input type="text" value=".1.3.6.1.2.1.31.1.1.1.6.103007"/>
Type of information	<input type="text" value="Numeric (unsigned)"/>
Units	<input type="text" value="Bps"/>
* Update interval	<input type="text" value="1m"/>

✓ But how did I obtain the numeric OID form if the SNMPWalk output was textual?



GETTING THE NUMERIC OID FORM

✓ Textual output

```
IF-MIB::ifHCInOctets.103007 = Counter64: 7566464822  
IF-MIB::ifHCInOctets.103008 = Counter64: 48097542881  
IF-MIB::ifHCInOctets.103009 = Counter64: 75748849150  
IF-MIB::ifHCInOctets.103010 = Counter64: 25963616931
```

✓ Let's use snmptranslate

```
[root@localhost ~]# snmptranslate -On -IR ifHCInOctets  
.1.3.6.1.2.1.31.1.1.1.6
```

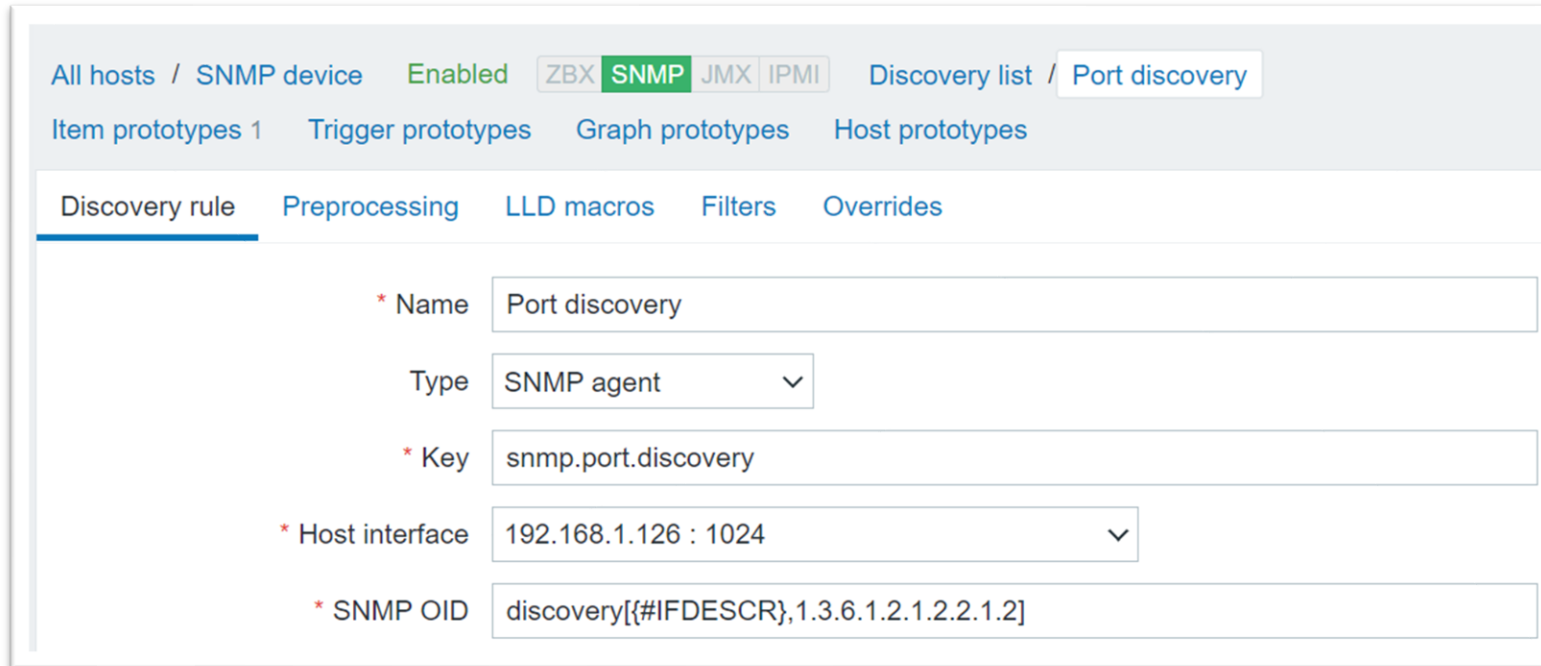
✓ No we just need to add the index – 103007 at the end of the OID

* Key	<input type="text" value="ifHCInOctets.103007"/>	<input type="button" value="Select"/>
* Host interface	<input type="text" value="192.168.1.126 : 1024"/>	▼
* SNMP OID	<input type="text" value=".1.3.6.1.2.1.31.1.1.1.6.103007"/>	



CREATING SNMP AGENT LLD RULE

✓ Create an LLD rule



The screenshot shows the Zabbix configuration interface for creating an LLD rule. The breadcrumb navigation is "All hosts / SNMP device Enabled ZBX SNMP JMX IPMI Discovery list / Port discovery". Below this are links for "Item prototypes 1", "Trigger prototypes", "Graph prototypes", and "Host prototypes". The "Discovery rule" tab is selected, with sub-tabs for "Preprocessing", "LLD macros", "Filters", and "Overrides". The configuration fields are as follows:

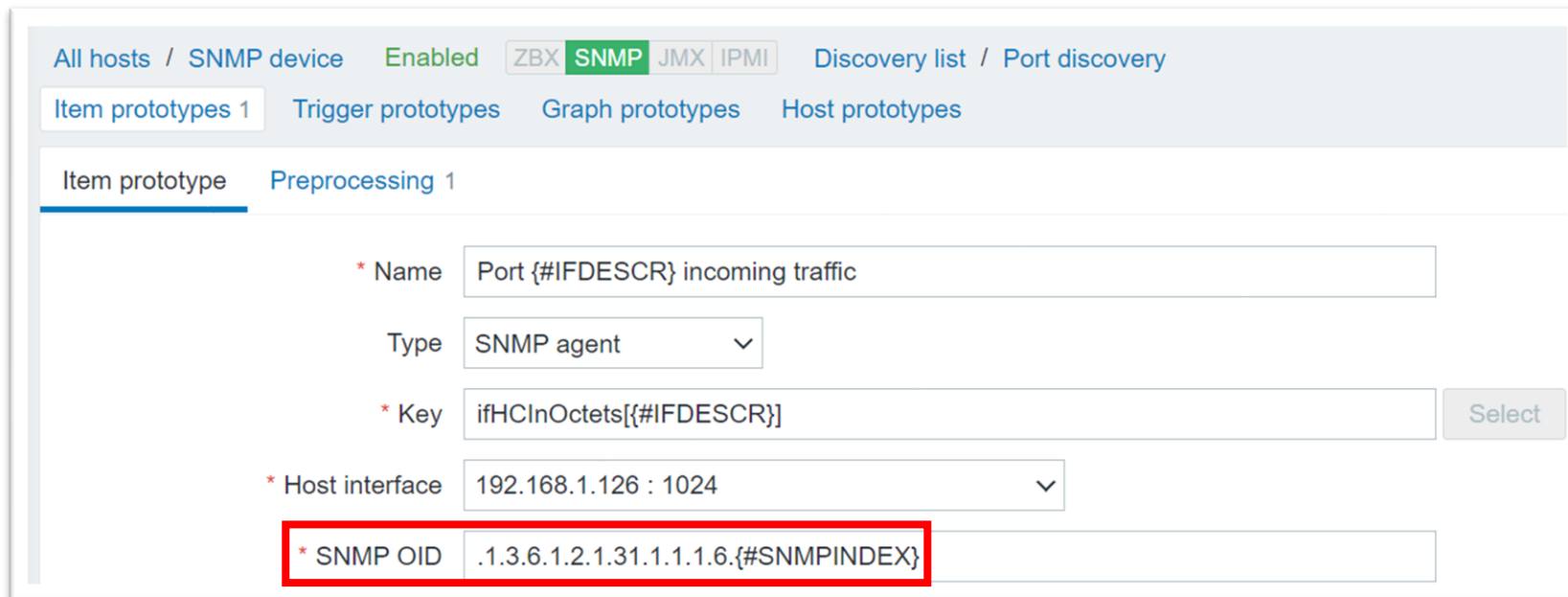
- * Name: Port discovery
- Type: SNMP agent
- * Key: snmp.port.discovery
- * Host interface: 192.168.1.126 : 1024
- * SNMP OID: discovery[#{IFDESCR},1.3.6.1.2.1.2.2.1.2]

- ✓ We will be discovering all of the indexes on 1.3.6.1.2.1.2.2.1.2 (*IFDescr*)
- ✓ We will also discover all of the available port descriptions



CREATING SNMP AGENT LLD ITEM PROTOTYPE

- ✓ Create an item prototype for incoming traffic




The screenshot shows the Zabbix web interface for creating an item prototype. The breadcrumb navigation is "All hosts / SNMP device Enabled ZBX SNMP JMX IPMI Discovery list / Port discovery". The "Item prototypes 1" tab is active. The "Item prototype" section is titled "Preprocessing 1". The configuration fields are:

- * Name: Port {#IFDESCR} incoming traffic
- Type: SNMP agent
- * Key: ifHCInOctets[{#IFDESCR}] (with a "Select" button)
- * Host interface: 192.168.1.126 : 1024
- * SNMP OID: .1.3.6.1.2.1.31.1.1.1.6.{#SNMPINDEX} (highlighted with a red box)

- ✓ Discovered index {#SNMPINDEX} will be added at the end of the OID (*ifHCInOctets*)



ERROR – NO SUCH INSTANCE CURRENTLY EXISTS

Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type	Applications	Status	Info
	...		ifHCInOctets.103007	1m	90d	365d	SNMP agent		Enabled	
	Port discovery: Port AltInband1 incoming traffic		ifHCInOctets[AltInband1]	1m	90d	365d	SNMP agent		Not supported	
	Port discovery: Port AltInband2 incoming traffic		ifHCInOctets[AltInband2]	1m	90d	365d	SNMP agent			
	Port discovery: Port AltInband3 incoming traffic		ifHCInOctets[AltInband3]	1m	90d	365d	SNMP agent			

No Such Instance currently exists at this
OID

- ✓ Caused by having extra indexes in IfDescr compared to ifHCInOctets
- ✓ Fixed by filtering out unnecessary Indexes by IfDescr:

All hosts / SNMP device Enabled ZBX SNMP JMX IPMI Discovery list / Port discovery

Item prototypes 1 Trigger prototypes Graph prototypes Host prototypes

Discovery rule Preprocessing LLD macros **Filters 1** Overrides

Filters	Label	Macro	Regular expression	Action
	A	{#IFDESCR}	does not match <input type="checkbox"/> (^AltInband ^backplane FXS ^vds ^lpbk ^eth0\$ ^eth1\$	Remove

[Add](#)

[Update](#) [Clone](#) [Execute now](#) [Test](#) [Delete](#) [Cancel](#)

LLD ENTITY FILTERING

✓ To make filtering easier, try discovering additional OID's (*IFTYPE*)

All templates / [Template Adtran TA1148](#) / [Discovery list](#) / [ethernet-like Network interfaces dis...](#)

[Item prototypes 9](#) / [Trigger prototypes 4](#) / [Graph prototypes 1](#) / [Host prototypes](#)

[Discovery rule](#) / [Preprocessing](#) / [LLD macros](#) / [Filters 3](#) / [Overrides](#)

Parent discovery rules [Template Module Interfaces SNMPv2 ethernet-csmacd](#)

* Name

Type

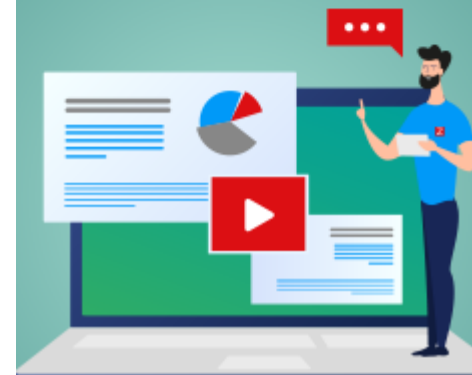
* Key

* SNMP OID

[Discovery rule](#) / [Preprocessing](#) / [LLD macros](#) / [Filters 3](#) / [Overrides](#)

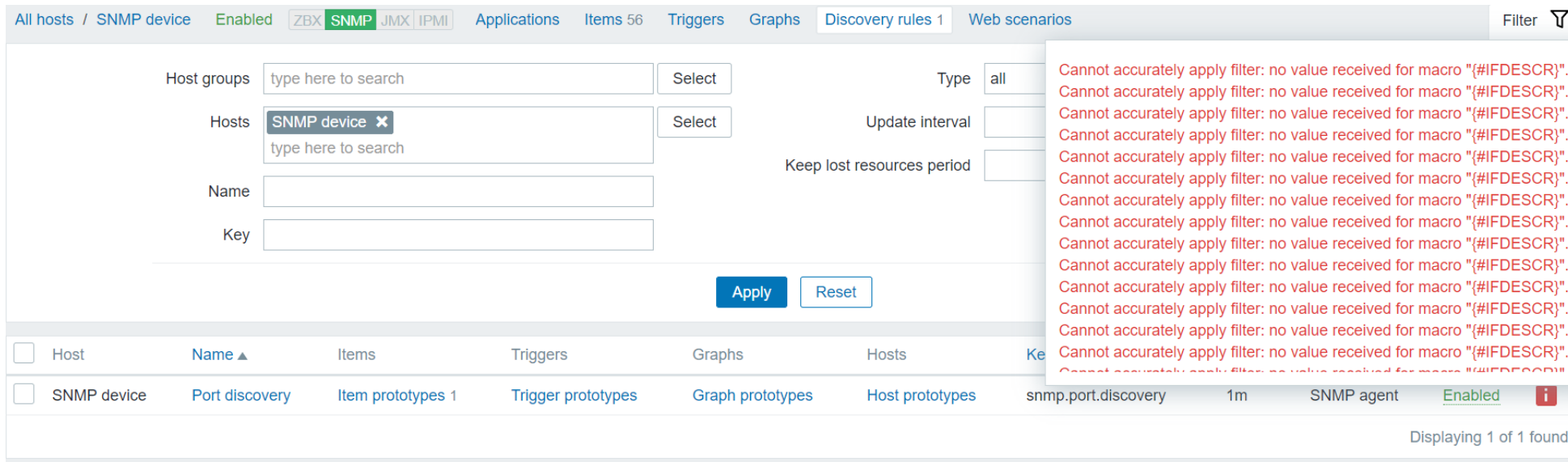
Type of calculation (A and B) and C

Filters	Label	Macro		Regular expression	Action
	A	<input type="text" value="{#IFTYPE}"/>	matches	<input type="text" value="{#IFTYPE}"/>	Remove
	B	<input type="text" value="{#IFTYPE}"/>	does not match	<input type="text" value="{#IFTYPE}"/>	Remove
	C	<input type="text" value="{#SNMPINDEX}"/>	does not match	<input type="text" value="{#SNMPINDEX}"/>	Remove



ERROR – NO VALUE RECEIVED FOR MACRO

☑ When trying to discover multiple attributes – {#IFNAME}, {#IFTYPE}, {#IFDESCR}



The screenshot shows the Zabbix Discovery Rules configuration page for the 'SNMP device' host group. The configuration includes fields for Host groups, Hosts, Name, Key, Type, Update interval, and Keep lost resources period. A table below the configuration shows the discovery rule details. An error message is displayed in a red box, stating: "Cannot accurately apply filter: no value received for macro {#IFDESCR}".

Host	Name	Items	Triggers	Graphs	Hosts	Key	Update interval	Keep lost resources period	SNMP agent	Status
<input type="checkbox"/>	SNMP device	Port discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	snmp.port.discovery	1m	SNMP agent	Enabled

Displaying 1 of 1 found



ERROR – NO VALUE RECEIVED FOR MACRO

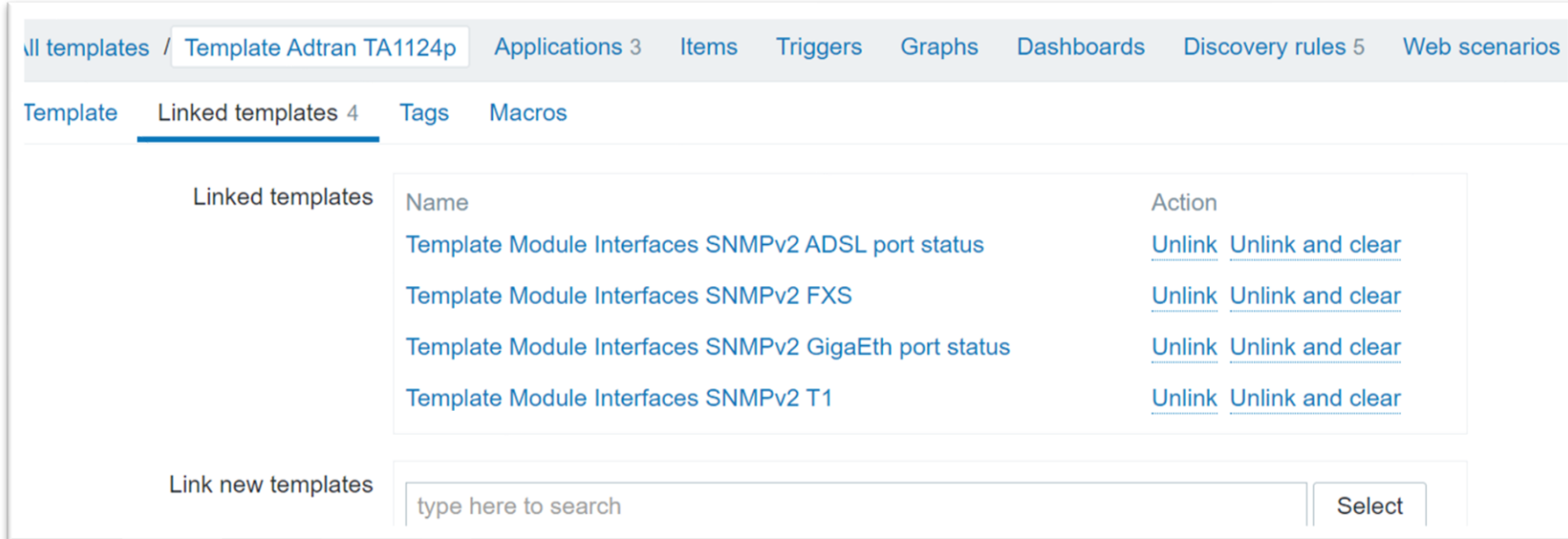
- ❑ Zabbix tries to apply filters for all of the OID's in the discovery rule

#IFNAME	#IFTYPE	#IFDESCR
1.3.6.1.2.1.31.1.1.1.1	1.3.6.1.2.1.2.2.1.3.1	1.3.6.1.2.1.2.2.1.2.1
1.3.6.1.2.1.31.1.1.1.2	1.3.6.1.2.1.2.2.1.3.2	1.3.6.1.2.1.2.2.1.2.2
1.3.6.1.2.1.31.1.1.1.3	1.3.6.1.2.1.2.2.1.3.3	1.3.6.1.2.1.2.2.1.2.3

- ❑ Index 3 is missing for {#IFTYPE} and {#IFDESCR}
- ❑ Filtering is still attempted by {#IFTYPE} or {#IFDESCR}

MODULAR TEMPLATES AND LLD RULES

✓ Try creating modular discovery rules or templates



The screenshot shows the Zabbix web interface for a template named 'Template Adtran TA1124p'. The breadcrumb navigation includes 'All templates / Template Adtran TA1124p' and other categories like 'Applications 3', 'Items', 'Triggers', 'Graphs', 'Dashboards', 'Discovery rules 5', and 'Web scenarios'. Below the breadcrumb, there are tabs for 'Template', 'Linked templates 4', 'Tags', and 'Macros'. The 'Linked templates' tab is active, displaying a table with the following content:

Name	Action
Template Module Interfaces SNMPv2 ADSL port status	Unlink Unlink and clear
Template Module Interfaces SNMPv2 FXS	Unlink Unlink and clear
Template Module Interfaces SNMPv2 GigaEth port status	Unlink Unlink and clear
Template Module Interfaces SNMPv2 T1	Unlink Unlink and clear

Below the table, there is a section for 'Link new templates' with a search input field containing the placeholder text 'type here to search' and a 'Select' button.

- ✓ Each discovery rule is created for specific interface type
- ✓ Allows you to link/unlink templates for specific entity types to a parent template
- ✓ You can then enable/disable discovery rules on the host level

ENGLISH

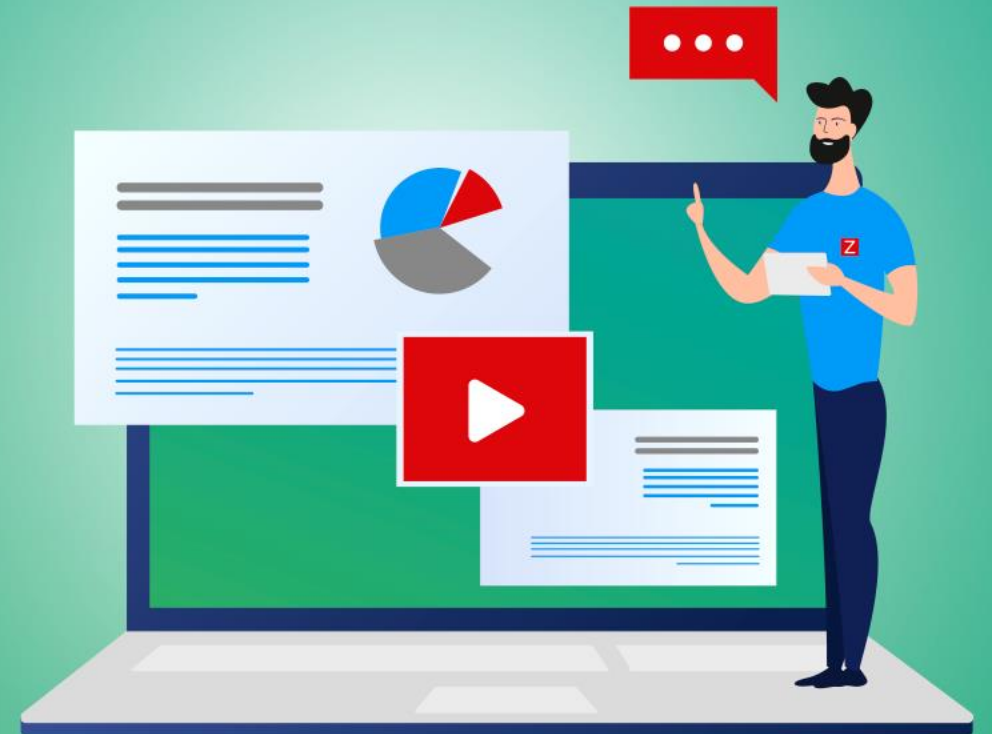


MEETUP ONLINE '21

QUESTIONS?

ARTŪRS LONTONS
TECHNICAL SUPPORT ENGINEER

ZABBIX



ENGLISH



MEETUP ONLINE '21

THANK YOU!

ARTŪRS LONTONS
TECHNICAL SUPPORT ENGINEER

ZABBIX

