initMAX

ZABBIX PREMIUM PARTNER

ZABBIX CERTIFIED TRAINER

ISP Alliance a. s. – Conference 2024

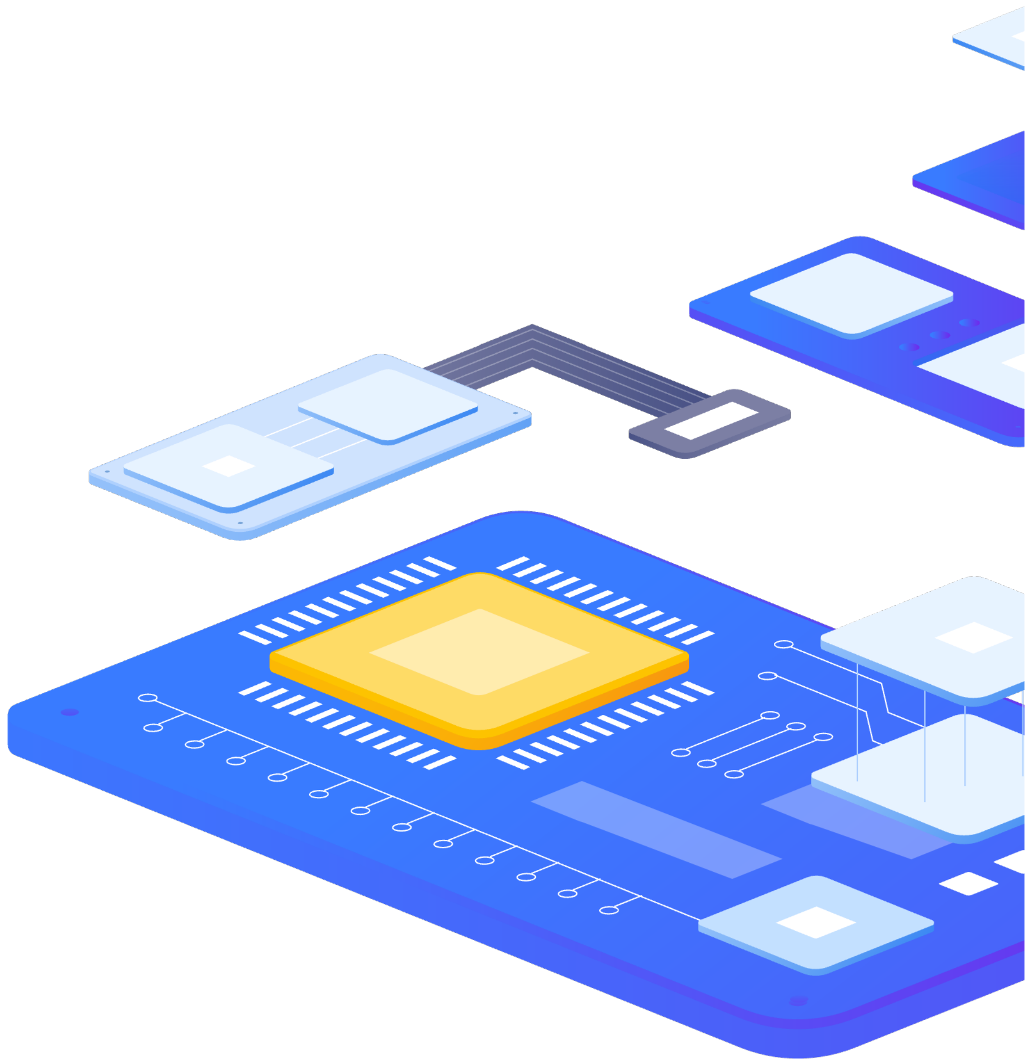
Zabbix – Tips and Tricks

ISP Alliance a. s. – Conference 2024

Zabbix – Tips and Tricks

initMAX

Tips and Tricks for effective monitoring of network devices with Zabbix.



1

Low-Level Discovery tips and tricks

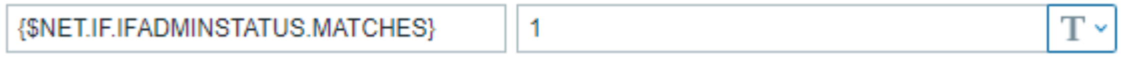
Dynamic resource monitoring

Low-level discovery provides a way to automatically create items, triggers, and graphs for different entities on a host.

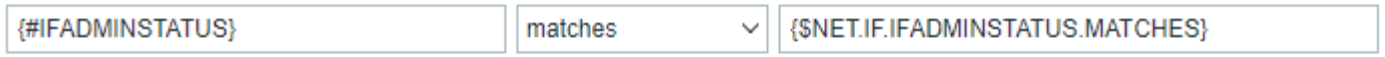
Configuring Low-level Discovery brings needs for data exceptions and conditoins for unexcepted situations. There are three tools in Discovery to provide solution:

- **Preprocessing** - allows to define transformation rules (for example fix data)
- **Filters** – defines entities that match the criteria (exclude interface for example)
- **Overrides** - allows setting rules to modify the list of item, trigger, graph .. (create items and graphs but not triggers or with different severity for unnecessary interfaces)

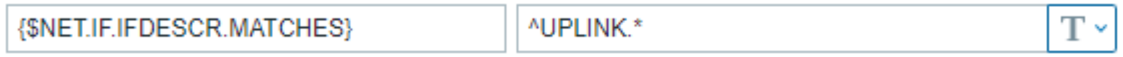
LLD Filters

- Defines whether element passes the Discovery rule. (based on regular expression)
- Examples (with user macro definition):
  - Monitor Network interfaces with ifAdminStatus = 1 (UP)
    - Macro: 

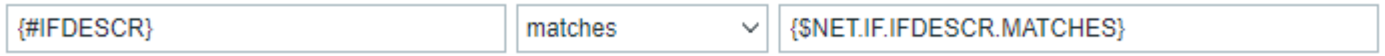
{$NET.IF.IFADMINSTATUS.MATCHES}	1

    - LLD filter: 

{#IFADMINSTATUS}	matches	{$NET.IF.IFADMINSTATUS.MATCHES}

  - Monitor Network interfaces with „UPLINK“ prefix in ifDescr
    - Macro: 

{$NET.IF.IFDESCR.MATCHES}	^UPLINK.*

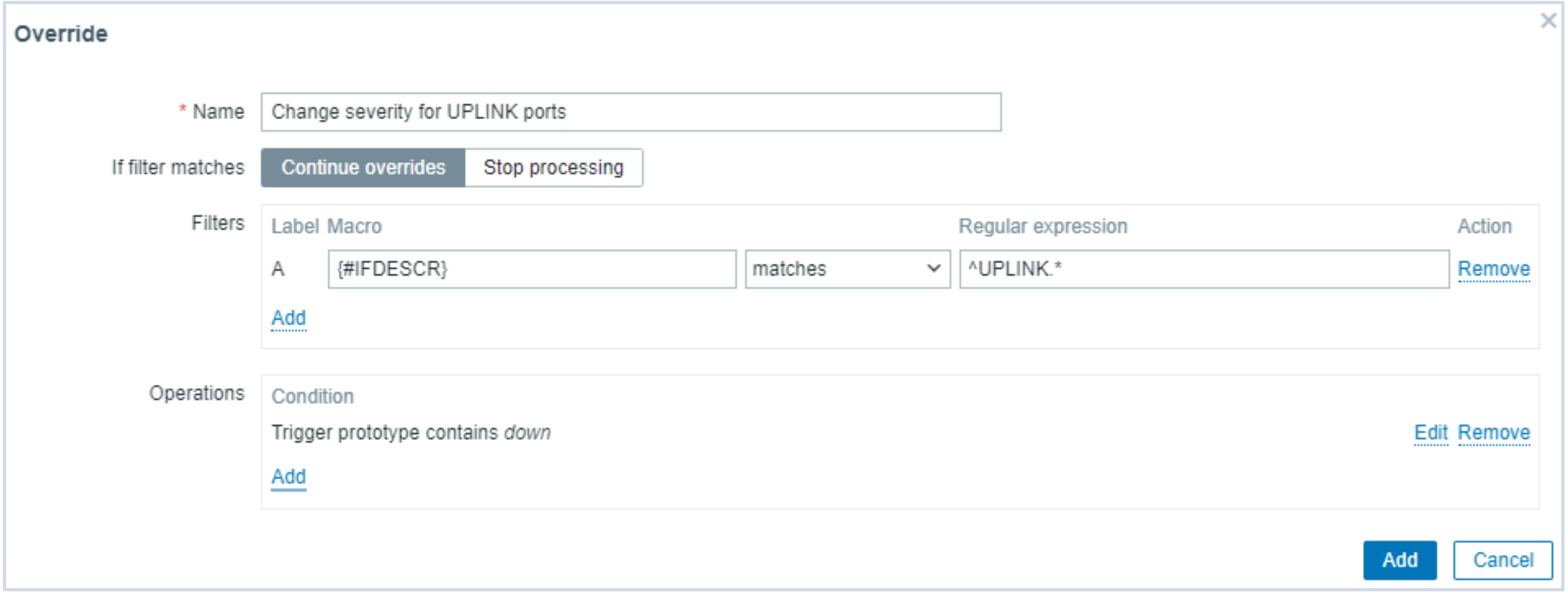
    - LLD filter: 

{#IFDESCR}	matches	{$NET.IF.IFDESCR.MATCHES}

LLD Overrides

- Defines exceptions for an element which passes the Discovery and Filter rule.

  In this case is a change of trigger severity:



Override

* Name: Change severity for UPLINK ports

If filter matches: [✓ Continue overrides] Stop processing

Filters

Label	Macro		Regular expression	Action
A	{#IFDESCR}	matches	^UPLINK.*	Remove

Add

Operations

Condition	
Trigger prototype contains *down*	Edit Remove

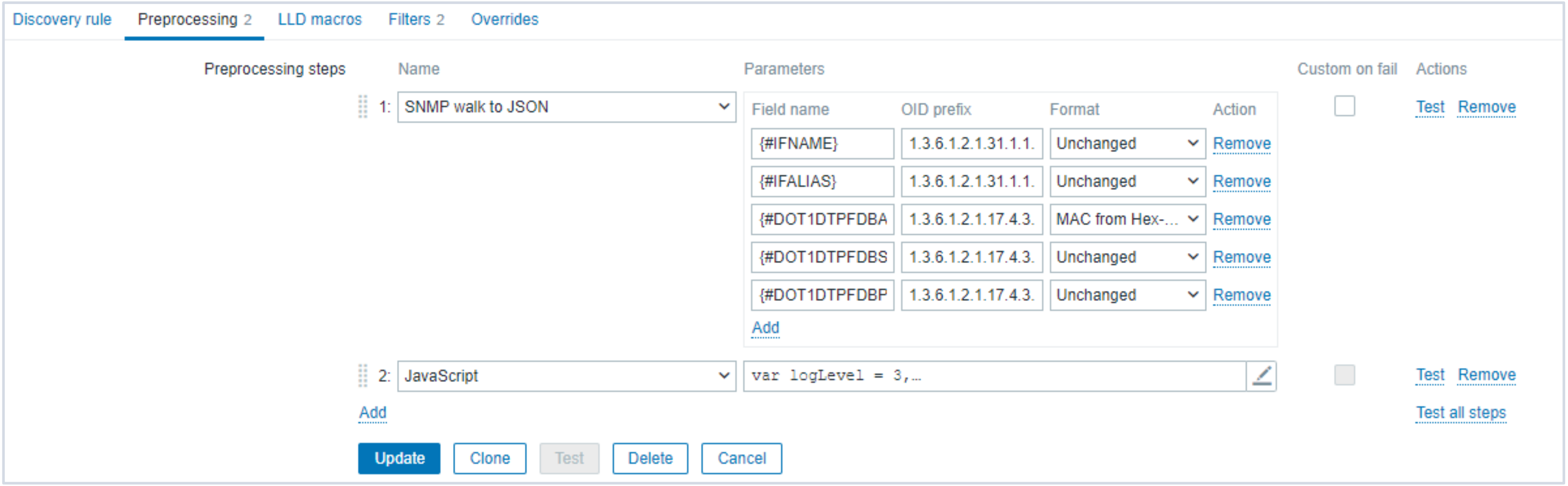
Add

Add Cancel

ISP Alliance a. s. – Conference 2024

LLD Preprocessing

- Preprocessing – transforms input data to JSON format



Discovery rule | Preprocessing 2 | LLD macros | Filters 2 | Overrides

Preprocessing steps

	Name	Parameters	Custom on fail	Actions
1:	SNMP walk to JSON	(see below)	☐	Test Remove
2:	JavaScript	`var logLevel = 3,…`	☐	Test Remove

Step 1 parameters:

Field name	OID prefix	Format	Action
{#IFNAME}	1.3.6.1.2.1.31.1.1.	Unchanged	Remove
{#IFALIAS}	1.3.6.1.2.1.31.1.1.	Unchanged	Remove
{#DOT1DTPFDBA	1.3.6.1.2.1.17.4.3.	MAC from Hex-…	Remove
{#DOT1DTPFDBS	1.3.6.1.2.1.17.4.3.	Unchanged	Remove
{#DOT1DTPFDBP	1.3.6.1.2.1.17.4.3.	Unchanged	Remove

Add

Add | Test all steps

Update | Clone | Test | Delete | Cancel

Tagging

Tags on item / trigger level– Why?

- Dynamic Tags based on LLD macros
- Filtering items in latest data
- Correlations
- Notifications

2

Event correlation

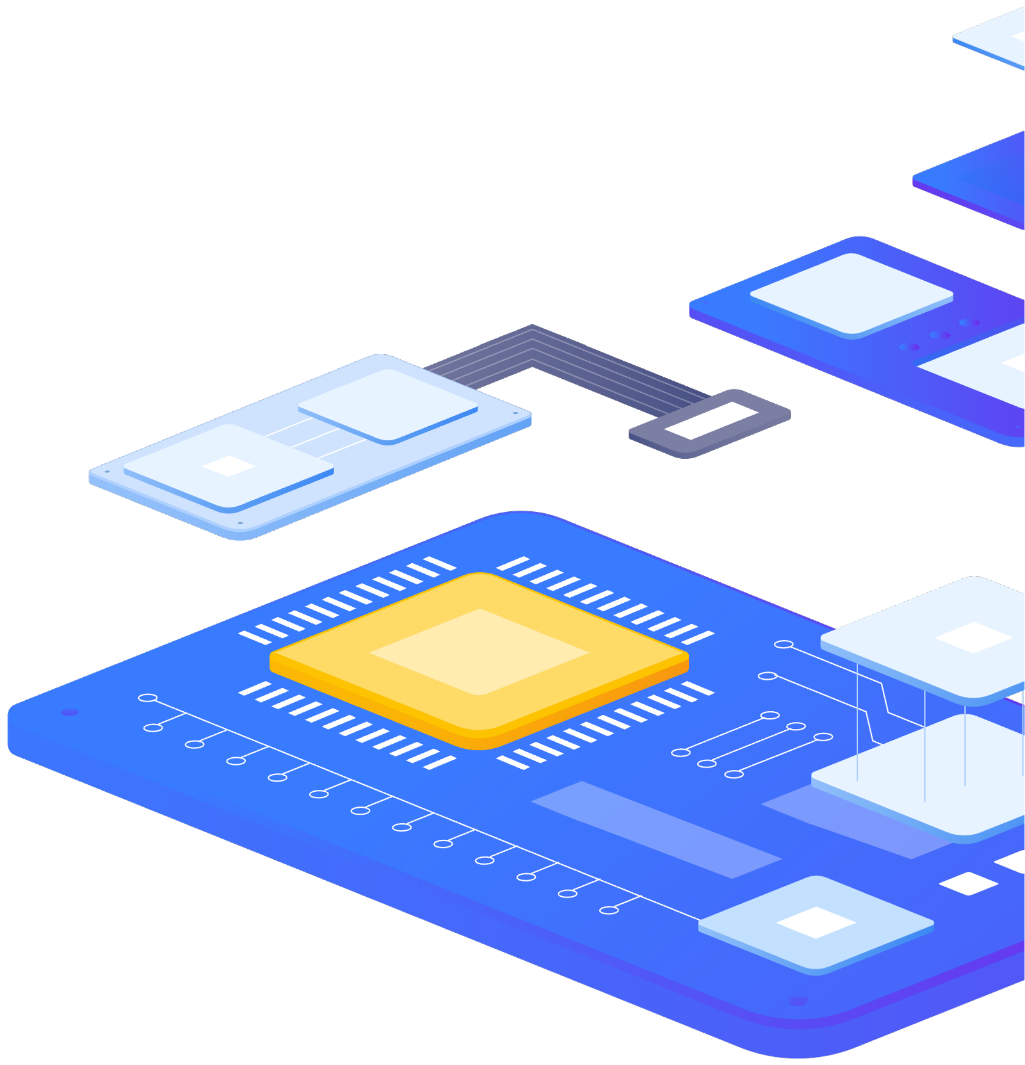
ISP Alliance a. s. – Conference 2024

EVENT CORRELATION

In Zabbix, it is possible to correlate problem events with their resolution.

- **On trigger level** - Allows to correlate separate problems reported by one trigger, need to have Multiple Problem Event Generation mode enabled for a trigger
- **Globally** - Problems reported from different triggers can be correlated using global correlation rules

Avoid using common tag names that may end up being used by different correlation configurations



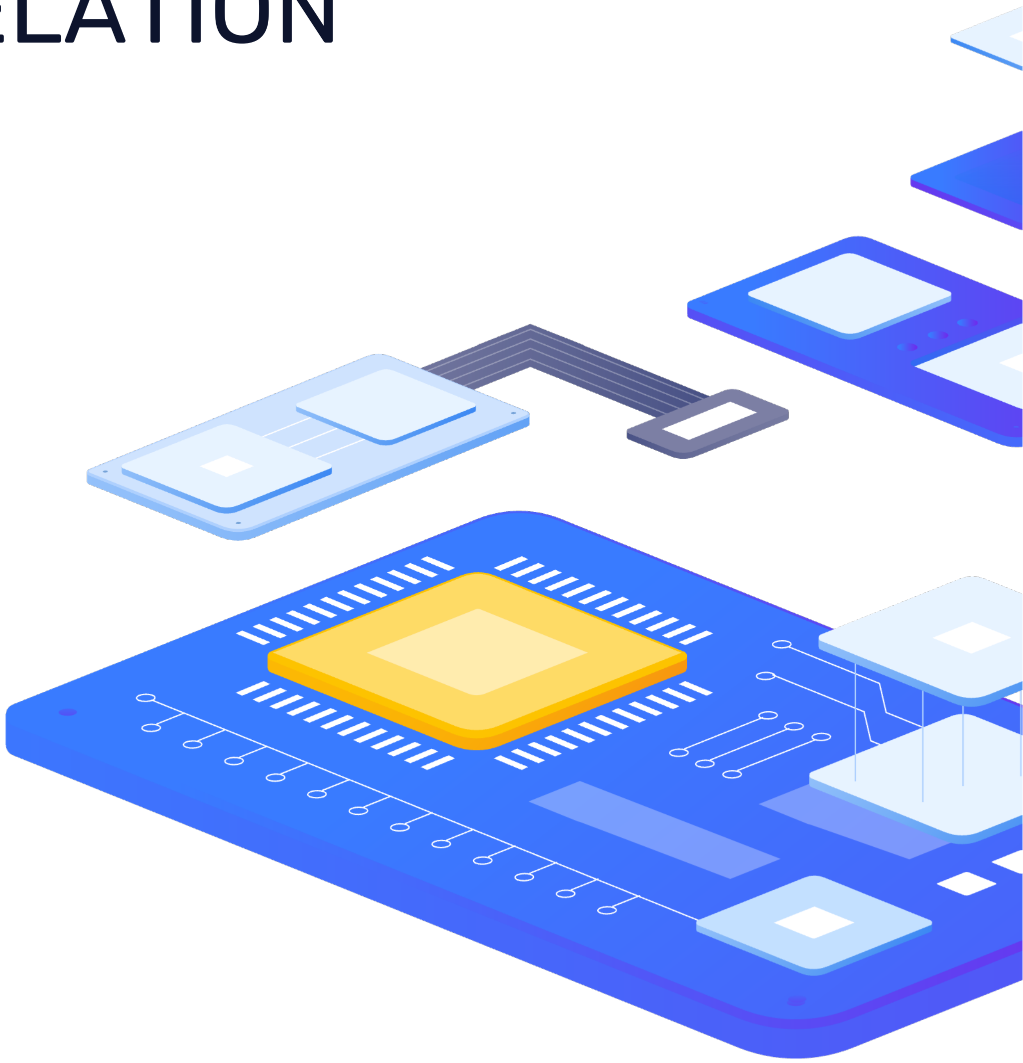
TRIGGER-BASED EVENT CORRELATION

In general, an OK event closes all problem events created by one trigger, but there are cases when we require a more detailed approach.

Correlate separate problems reported by one trigger

- **Tags are used** to extract values and create identification for problem events
- Problems can be closed individually based on **matching tags** and their values

Useful for events, log files, SNMP traps, etc.



TRIGGER-BASED EVENT CORRELATION

Substring extraction is usually used for populating the tag name or tag value, with a specific value using a macro function, i.e. :

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

- By applying a regular expression to the value obtained by the {ITEM.VALUE}, {ITEM.LASTVALUE} macro or a low-level discovery macro

```
IF-MIB::ifOperStatus.4         type=2  value=INTEGER: 2
{{ITEM.VALUE}.regsub("IF-MIB::ifOperStatus.(\d+)",\1)}
```

initMAX

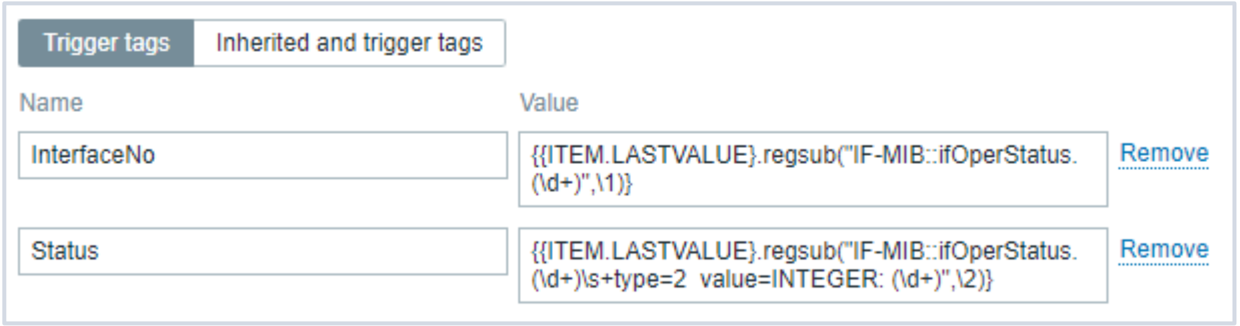
EXAMPLE – Port Operational status - SNMPtrap

```
14:19:42 2024/09/04 ZBXTRAP 10.1.1.212
PDU INFO:
  requestid                      1459624785
  notificationtype               TRAP
  receivedfrom                   UDP: [10.1.1.212]:55579->[10.1.1.91]:162
  errorindex                     0
  errorstatus                    0
  messageid                      0
  version                        1
  transactionid                  6
VARBINDS:
  DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (123552444) 14 days,
7:12:04.44
  SNMPv2-MIB::snmpTrapOID.0      type=6  value=OID: IF-MIB::linkDown
  SNMP-COMMUNITY-MIB::snmpTrapAddress type=4  value=Hex-STRING: 00 00 00 00 00 00 00 00
00 00 FF FF 0A 01 01 D4
  IF-MIB::ifIndex.4              type=2  value=INTEGER: 4
  IF-MIB::ifAdminStatus.4        type=2  value=INTEGER: 1
  IF-MIB::ifOperStatus.4         type=2  value=INTEGER: 2
```

ISP Alliance a. s. – Conference 2024

EXAMPLE – Port Operational status

So, creating a trigger with an example tag:



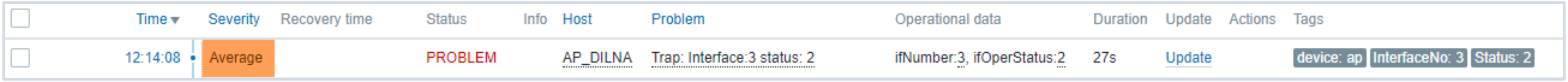
Trigger tags | Inherited and trigger tags

Name	Value	
InterfaceNo	{{ITEM.LASTVALUE}.regsub("IF-MIB::ifOperStatus.(\d+)",\1)}	Remove
Status	{{ITEM.LASTVALUE}.regsub("IF-MIB::ifOperStatus.(\d+)\s+type=2 value=INTEGER: (\d+)",\2)}	Remove

Would allow us to extract error ID from a log line:

```
InterfaceNo: 4
Status:  2
```

To create a problem that would be informative and possible to correlate:

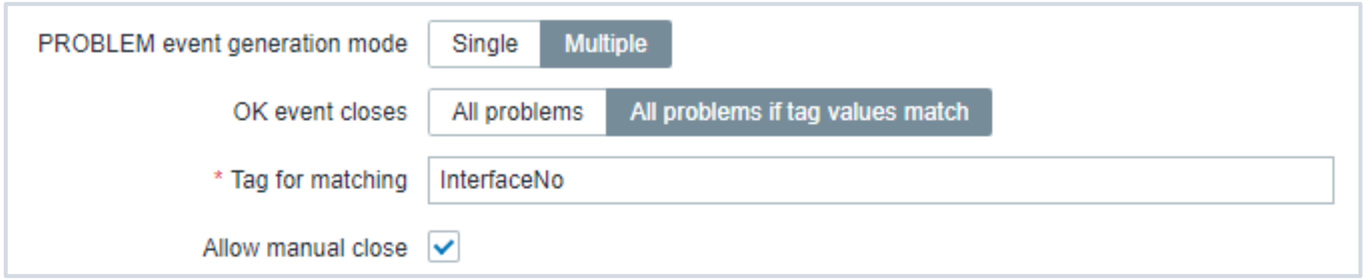


	Time ▾	Severity	Recovery time	Status	Info	Host	Problem	Operational data	Duration	Update	Actions	Tags
☐	12:14:08	Average		PROBLEM		AP_DILNA	Trap: Interface:3 status: 2	ifNumber:3, ifOperStatus:2	27s	Update		device: ap InterfaceNo: 3 Status: 2

ISP Alliance a. s. – Conference 2024

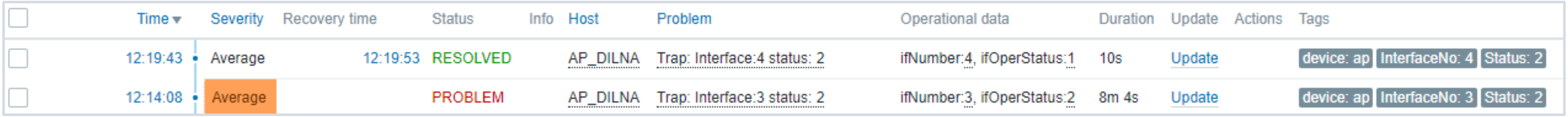
EXAMPLE – Port Operational status

Correlate triggers by Tag value:



PROBLEM event generation mode	Single [✓ Multiple]
OK event closes	All problems [✓ All problems if tag values match]
* Tag for matching	InterfaceNo
Allow manual close	✓

› Problem resolution based on Tag value:



	Time ▾	Severity	Recovery time	Status	Info	Host	Problem	Operational data	Duration	Update	Actions	Tags
☐	12:19:43	Average	12:19:53	RESOLVED		AP_DILNA	Trap: Interface:4 status: 2	ifNumber:4, ifOperStatus:1	10s	Update		device: ap InterfaceNo: 4 Status: 2
☐	12:14:08	Average		PROBLEM		AP_DILNA	Trap: Interface:3 status: 2	ifNumber:3, ifOperStatus:2	8m 4s	Update		device: ap InterfaceNo: 3 Status: 2

2

Effective throttling

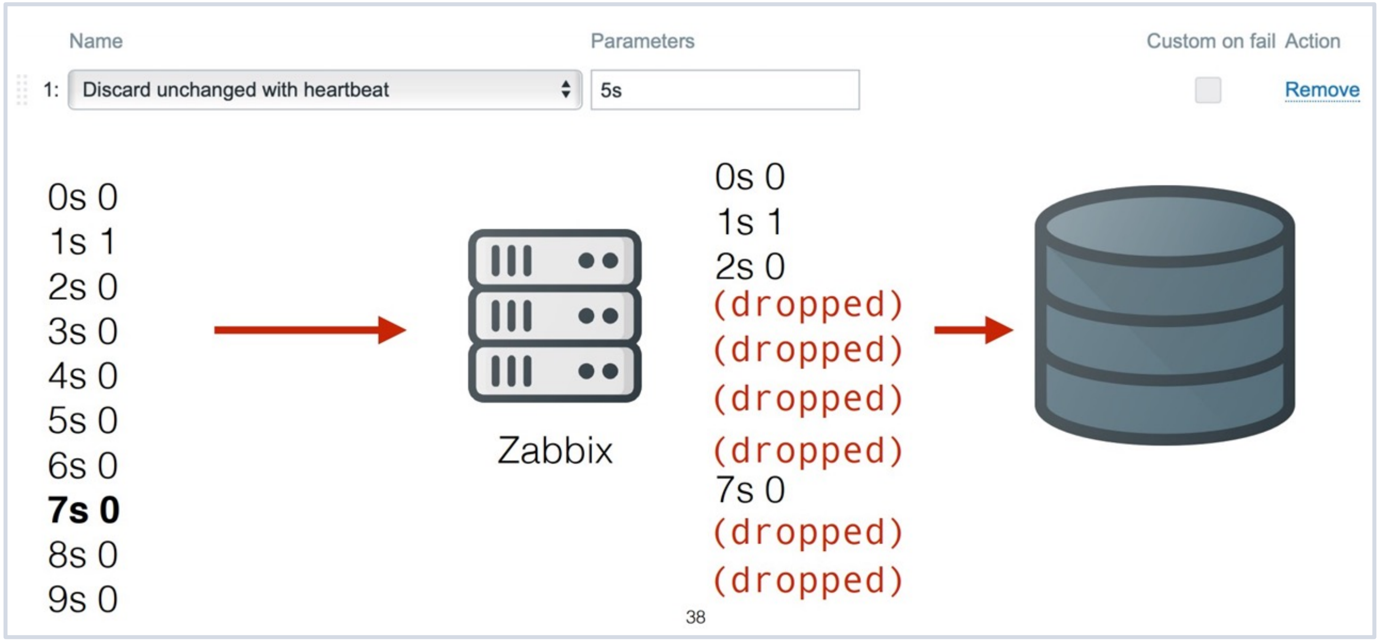
THROTTLING

For high-frequency monitoring, we need functionality to offload core components from the extensive load.

Throttling is the exact thing that will allow you to drop repetitive values on a Pre-processing level and collect only changing values.

Throttling types:

- Discard unchanged
- Discard unchanged with heartbeat

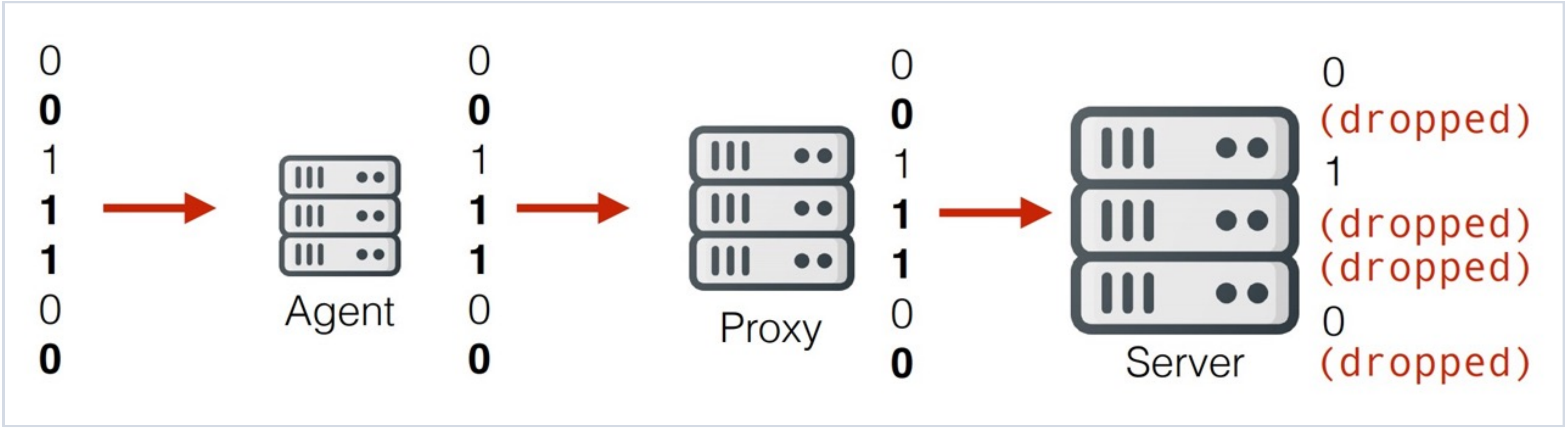


Name	Parameters	Custom on fail	Action
1: Discard unchanged with heartbeat	5s		Remove

THROTTLING

Throttling on proxy

- Proxy discards value during item preprocessing



initMAX

THROTTLING TYPES

Time	Values	Discard unchanged	Discard unchanged with heartbeat 30s	Explanation
00:00	0	0	0	
00:05	0			No value because same as previous
00:10	0			
00:15	1	1	1	Received different value
00:20	1			
00:25	1			
00:30	1			
00:35	1			
00:40	0	0	0	Received different value
00:45	0			
00:50	0			
00:55	0			
01:00	0			
01:05	0			
01:10	0		**0**	Value written because of heartbeat 30s
01:15	0			
01:20	1	1	1	Received different value
01:25	1			
01:30	0	0	0	Received different value
01:35	0			
01:40	0			
01:45	0			
01:50	0			
01:55	0			
02:00	0		**0**	Value written because of heartbeat 30s
02:05	0			

Throttling and false positives protection using history

If you use Throttling on state items, you may encounter false positive alerts.

This is because throttling does not allow you to use the min, max or avg functions to evaluate multiple values.

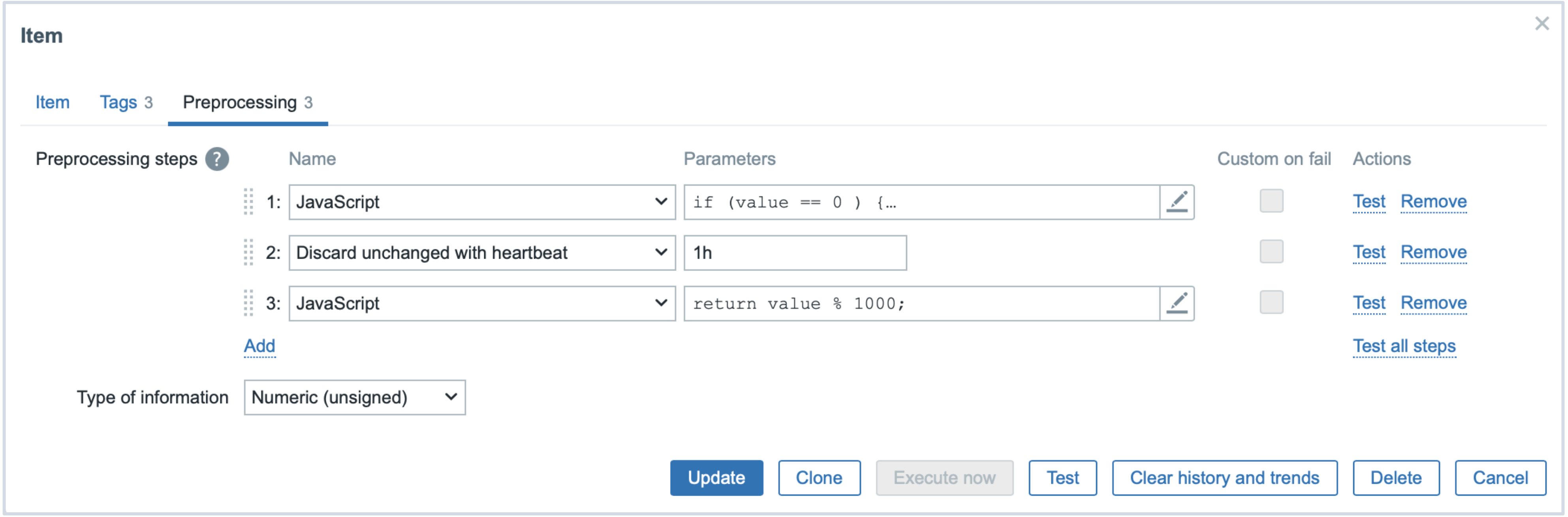
This is because Zabbix discards the same, consecutive states.

- Max, Min, Avg trigger functions
- Discard unchanged with heartbeat

Solution - Preprocessing

Change the value in error state:
(WIKI: https://www.initmax.com/wiki/throttling-and-false-positives-protection-using-min-max-avg/)

- Step 1: Add timestamp to value when is not OK
- Step 2: Discard unchanged with heartbeat
- Step 3: Remove timestamp



Item

Item | Tags 3 | Preprocessing 3

Preprocessing steps

	Name	Parameters	Custom on fail	Actions
1:	JavaScript	`if (value == 0 ) {…`	☐	Test Remove
2:	Discard unchanged with heartbeat	1h	☐	Test Remove
3:	JavaScript	`return value % 1000;`	☐	Test Remove

Add | Test all steps

Type of information: Numeric (unsigned)

Solution - Preprocessing

Add timestamp with javascript:

```
if (value == 0 ) {
  return value;
} else {
  return (Math.floor(Date.now() / 1000) - 1707000000 )*1000 + value;
}
```

- After throttling, you get back the original value using the expression:

```
return value % 1000;
```

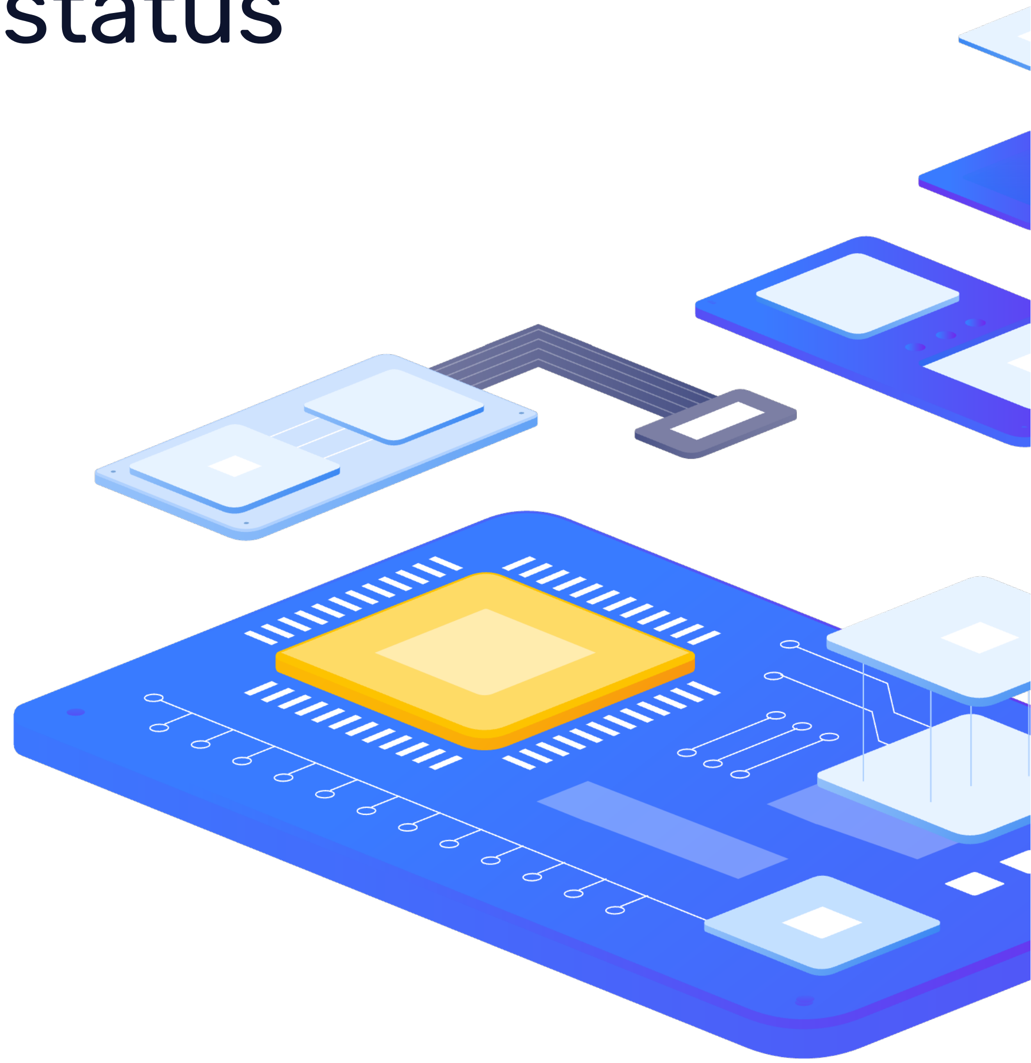
- Trigger condition:

```
min(//service.info["{#SERVICE.NAME}",state],#3)<>0
```

Example: Interface Operational status

Usage:

- Useless for traffic counter and nonostable values
- Usefull for status items:
  - Interface Opreational status
  - Service status



3

Switch port neighbours

More complex monitoring scenario

Task Definition:

- Collect MAC address of switch port connected device
- Compare this MAC address with inventory address of zabbix hosts – if found, store hostname as item value.

initMAX

Example: Switch port Neighbours

Technology stack:

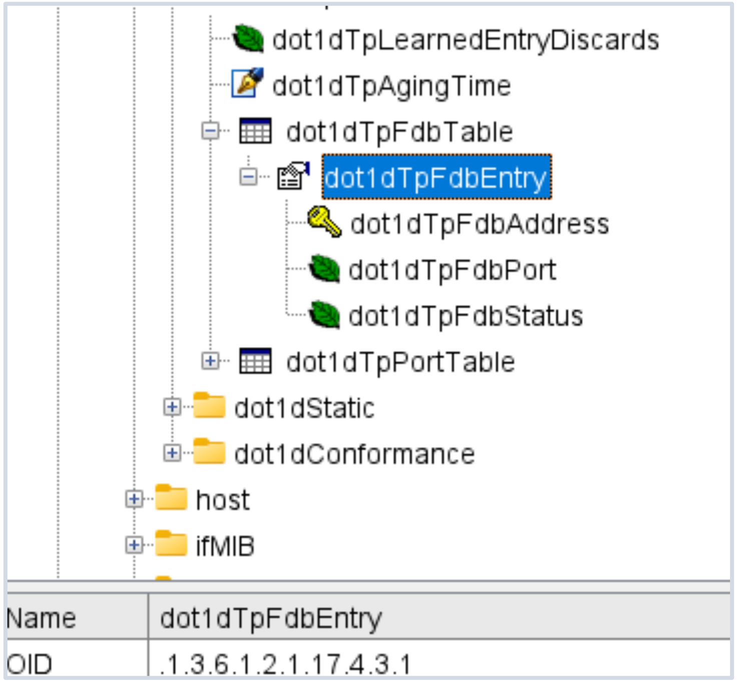
- SNMP table combination
- walk[] item
- Javascript preprocessing
- LLD processing
- API access

ISP Alliance a. s. – Conference 2024

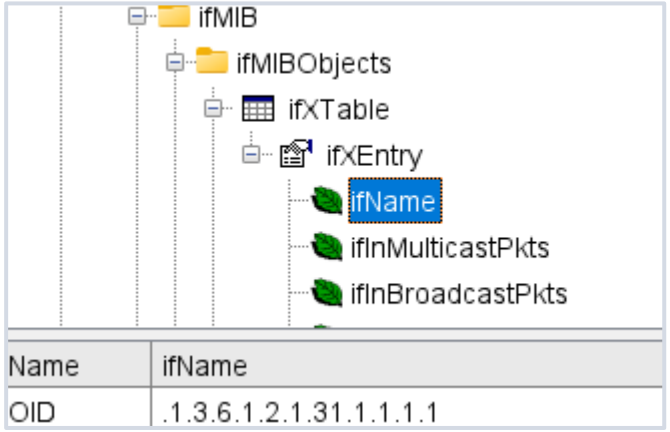
Example: Switch port Neighbours

More than one SNMP table combination

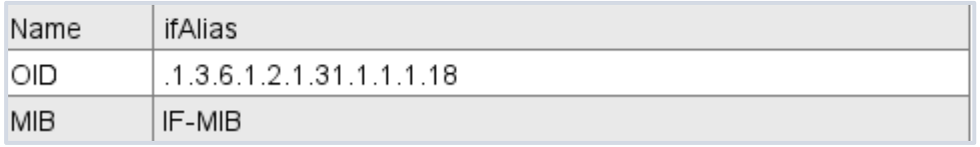
- The Bridge MIB module for managing devices that support IEEE 802.1D - .1.3.6.1.2.1.17 (**MAC table** of connected devices)
- Interface MIB table - .1.3.6.1.2.1.31 (**Interface table** containing additional information about interfaces)



Name	dot1dTpFdbEntry
OID	.1.3.6.1.2.1.17.4.3.1



Name	ifName
OID	.1.3.6.1.2.1.31.1.1.1.1



Name	ifAlias
OID	.1.3.6.1.2.1.31.1.1.1.18
MIB	IF-MIB

Example: Switch port Neighbours

- walk[] item

```
.1.3.6.1.2.1.17.4.3.1.1.0.8.155.194.18.194 = Hex-STRING: 00 08 9B C2 12 C2
.1.3.6.1.2.1.17.4.3.1.1.0.12.41.1.79.10 = Hex-STRING: 00 0C 29 01 4F 0A
.1.3.6.1.2.1.17.4.3.1.1.0.12.41.1.79.236 = Hex-STRING: 00 0C 29 01 4F EC
.1.3.6.1.2.1.17.4.3.1.2.0.8.155.194.18.194 = INTEGER: 46
.1.3.6.1.2.1.17.4.3.1.2.0.12.41.1.79.10 = INTEGER: 51
.1.3.6.1.2.1.17.4.3.1.2.0.12.41.1.79.236 = INTEGER: 51
.1.3.6.1.2.1.17.4.3.1.3.0.8.155.194.18.194 = INTEGER: 3
.1.3.6.1.2.1.17.4.3.1.3.0.12.41.1.79.10 = INTEGER: 3
.1.3.6.1.2.1.17.4.3.1.3.0.12.41.1.79.236 = INTEGER: 3
.1.3.6.1.2.1.31.1.1.1.1.46 = STRING: "gi46"
.1.3.6.1.2.1.31.1.1.1.1.51 = STRING: "gi51"
.1.3.6.1.2.1.31.1.1.1.18.46 = STRING: "SERVER-vmWare"
.1.3.6.1.2.1.31.1.1.1.18.51 = STRING: "UPLINK-pristavba"
```

initMAX

Example: Switch port Neighbours

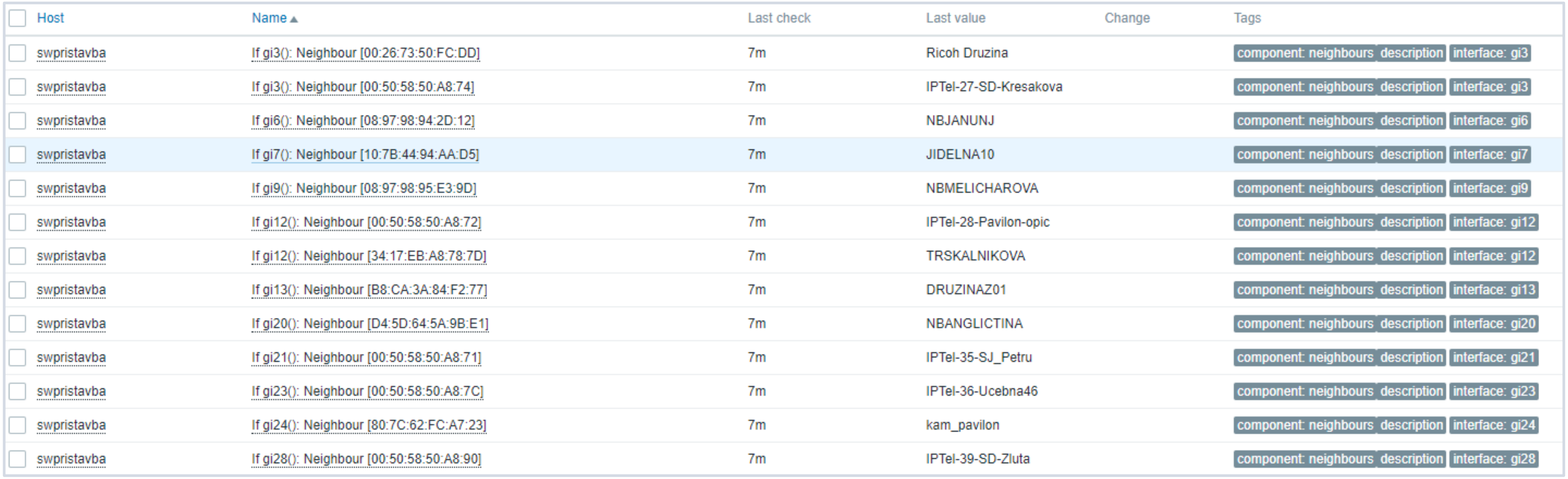
- Javascript preprocessing

```
arr = JSON.parse(value);
macArr = [];
for (x in arr) {
  if (arr[x]['{#DOT1DTPFDBPORT}']) {
    ifindex = arr.map(function (e) { return e['{#SNMPINDEX}'];
}).indexOf(arr[x]['{#DOT1DTPFDBPORT}'])
      if (ifindex >0) {
            arr[x]['{#IFNAME}']=arr[ifindex]['{#IFNAME}'];
            arr[x]['{#IFALIAS}']=arr[ifindex]['{#IFALIAS}'];
            macArr.push(arr[x]);
        }
    }
}
return JSON.stringify(macArr);
```

ISP Alliance a. s. – Conference 2024

initMAX

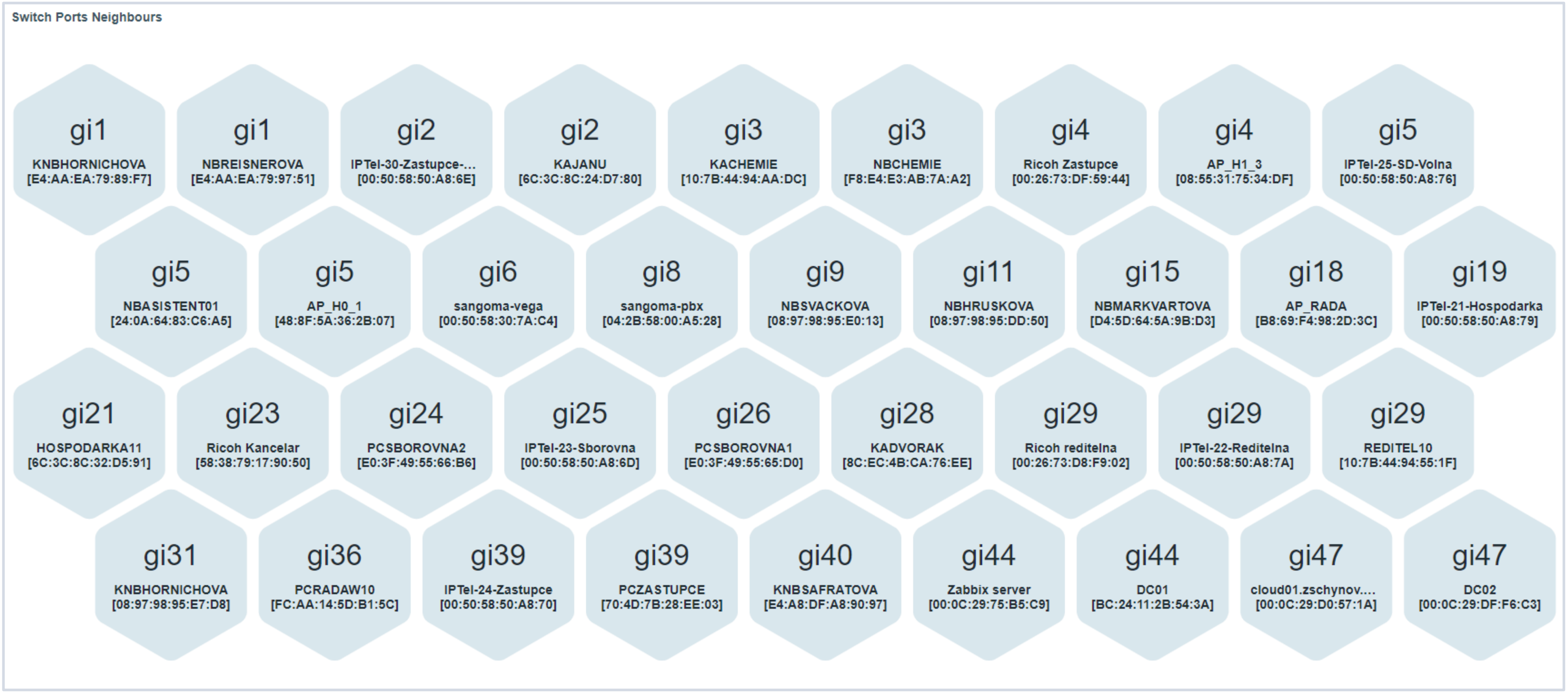
Example: Switch port Neighbours – latest data



☐	Host	Name ▲	Last check	Last value	Change	Tags
☐	swpristavba	If gi3(): Neighbour [00:26:73:50:FC:DD]	7m	Ricoh Druzina		component: neighbours description interface: gi3
☐	swpristavba	If gi3(): Neighbour [00:50:58:50:A8:74]	7m	IPTel-27-SD-Kresakova		component: neighbours description interface: gi3
☐	swpristavba	If gi6(): Neighbour [08:97:98:94:2D:12]	7m	NBJANUNJ		component: neighbours description interface: gi6
☐	swpristavba	If gi7(): Neighbour [10:7B:44:94:AA:D5]	7m	JIDELNA10		component: neighbours description interface: gi7
☐	swpristavba	If gi9(): Neighbour [08:97:98:95:E3:9D]	7m	NBMELICHAROVA		component: neighbours description interface: gi9
☐	swpristavba	If gi12(): Neighbour [00:50:58:50:A8:72]	7m	IPTel-28-Pavilon-opic		component: neighbours description interface: gi12
☐	swpristavba	If gi12(): Neighbour [34:17:EB:A8:78:7D]	7m	TRSKALNIKOVA		component: neighbours description interface: gi12
☐	swpristavba	If gi13(): Neighbour [B8:CA:3A:84:F2:77]	7m	DRUZINAZ01		component: neighbours description interface: gi13
☐	swpristavba	If gi20(): Neighbour [D4:5D:64:5A:9B:E1]	7m	NBANGLICTINA		component: neighbours description interface: gi20
☐	swpristavba	If gi21(): Neighbour [00:50:58:50:A8:71]	7m	IPTel-35-SJ_Petru		component: neighbours description interface: gi21
☐	swpristavba	If gi23(): Neighbour [00:50:58:50:A8:7C]	7m	IPTel-36-Ucebna46		component: neighbours description interface: gi23
☐	swpristavba	If gi24(): Neighbour [80:7C:62:FC:A7:23]	7m	kam_pavilon		component: neighbours description interface: gi24
☐	swpristavba	If gi28(): Neighbour [00:50:58:50:A8:90]	7m	IPTel-39-SD-Zluta		component: neighbours description interface: gi28

ISP Alliance a. s. – Conference 2024

Example: Switch port Neighbours - visualization



Switch Ports Neighbours

gi1
KNBHORNICHOVA
[E4:AA:EA:79:89:F7]

gi1
NBREISNEROVA
[E4:AA:EA:79:97:51]

gi2
IPTel-30-Zastupce-...
[00:50:58:50:A8:6E]

gi2
KAJANU
[6C:3C:8C:24:D7:80]

gi3
KACHEMIE
[10:7B:44:94:AA:DC]

gi3
NBCHEMIE
[F8:E4:E3:AB:7A:A2]

gi4
Ricoh Zastupce
[00:26:73:DF:59:44]

gi4
AP_H1_3
[08:55:31:75:34:DF]

gi5
IPTel-25-SD-Volna
[00:50:58:50:A8:76]

gi5
NBASISTENT01
[24:0A:64:83:C6:A5]

gi5
AP_H0_1
[48:8F:5A:36:2B:07]

gi6
sangoma-vega
[00:50:58:30:7A:C4]

gi8
sangoma-pbx
[04:2B:58:00:A5:28]

gi9
NBSVACKOVA
[08:97:98:95:E0:13]

gi11
NBHRUSKOVA
[08:97:98:95:DD:50]

gi15
NBMARKVARTOVA
[D4:5D:64:5A:9B:D3]

gi18
AP_RADA
[B8:69:F4:98:2D:3C]

gi19
IPTel-21-Hospodarka
[00:50:58:50:A8:79]

gi21
HOSPODARKA11
[6C:3C:8C:32:D5:91]

gi23
Ricoh Kancelar
[58:38:79:17:90:50]

gi24
PCSBOROVNA2
[E0:3F:49:55:66:B6]

gi25
IPTel-23-Sborovna
[00:50:58:50:A8:6D]

gi26
PCSBOROVNA1
[E0:3F:49:55:65:D0]

gi28
KADVORAK
[8C:EC:4B:CA:76:EE]

gi29
Ricoh reditelna
[00:26:73:D8:F9:02]

gi29
IPTel-22-Reditelna
[00:50:58:50:A8:7A]

gi29
REDITEL10
[10:7B:44:94:55:1F]

gi31
KNBHORNICHOVA
[08:97:98:95:E7:D8]

gi36
PCRADAW10
[FC:AA:14:5D:B1:5C]

gi39
IPTel-24-Zastupce
[00:50:58:50:A8:70]

gi39
PCZASTUPCE
[70:4D:7B:28:EE:03]

gi40
KNBSAFRATOVA
[E4:A8:DF:A8:90:97]

gi44
Zabbix server
[00:0C:29:75:B5:C9]

gi44
DC01
[BC:24:11:2B:54:3A]

gi47
cloud01.zschynov....
[00:0C:29:D0:57:1A]

gi47
DC02
[00:0C:29:DF:F6:C3]

4

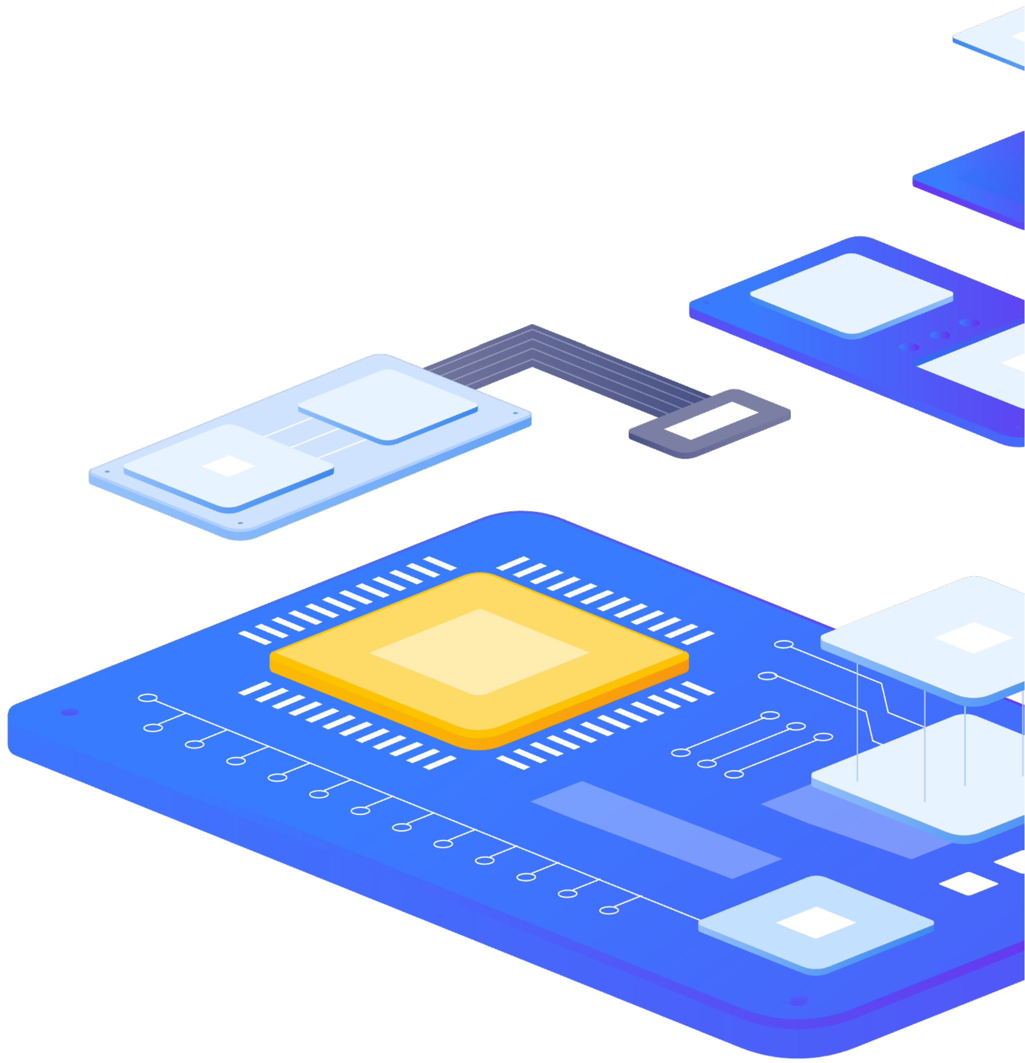
Template Dashboards

ISP Alliance a. s. – Conference 2024

Template Dashboards

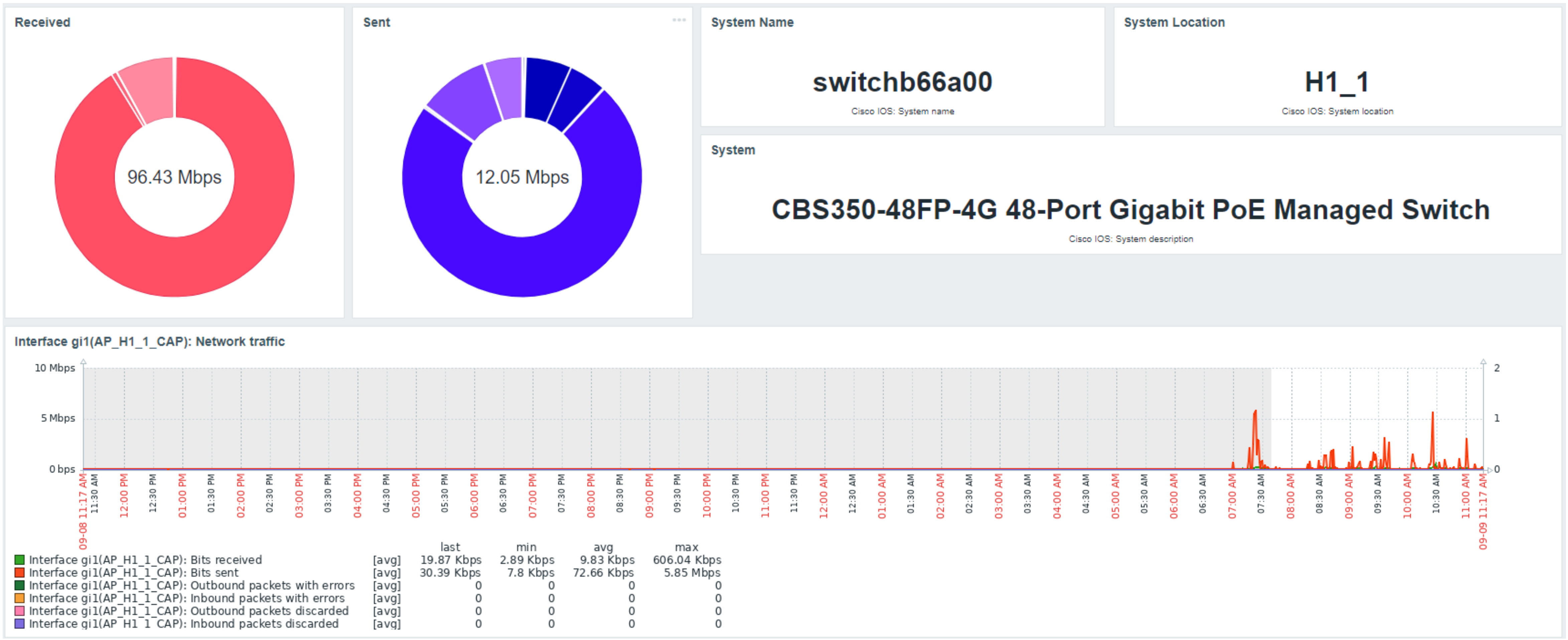
Big changes in zabbix version 7.0

- Possibility to use **any widgets** inside Template Dashboard
- Communication between widgets



ISP Alliance a. s. – Conference 2024

Example: Template Dashboards



Received

96.43 Mbps

Sent

12.05 Mbps

System Name

switchb66a00

Cisco IOS: System name

System Location

H1_1

Cisco IOS: System location

System

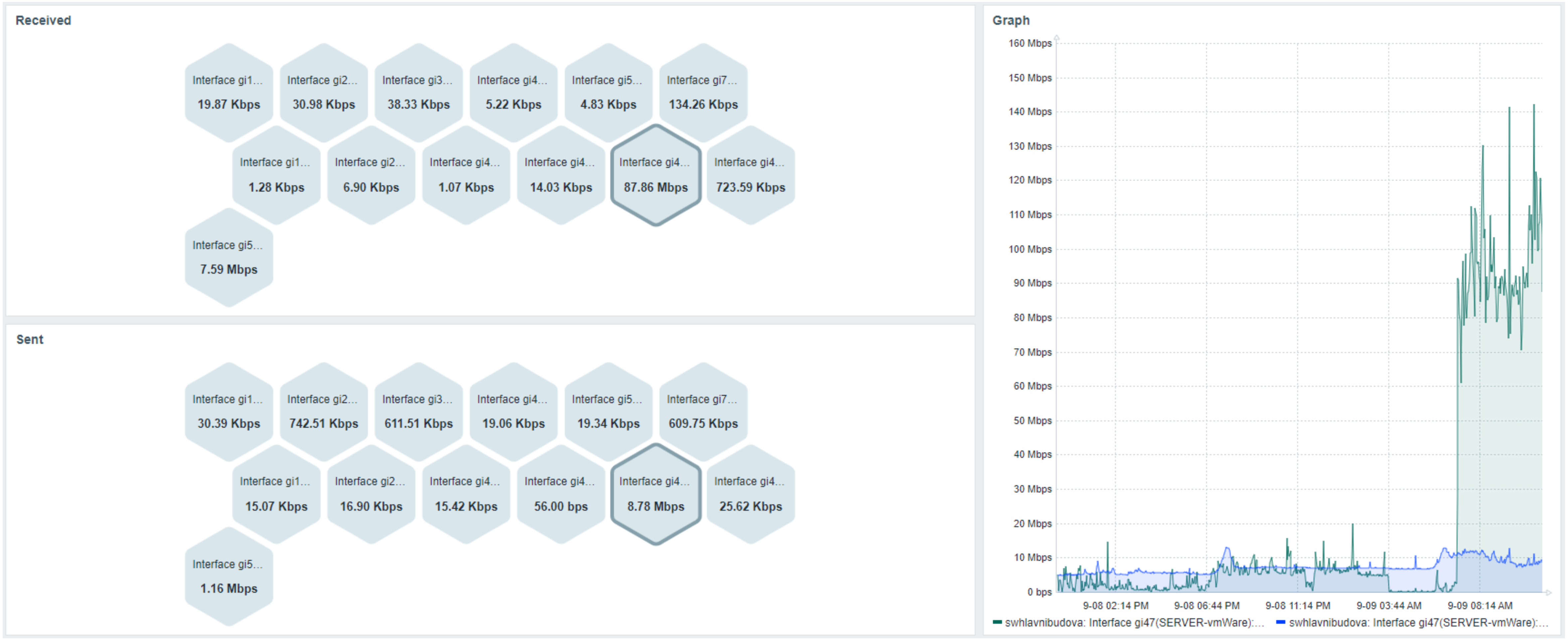
CBS350-48FP-4G 48-Port Gigabit PoE Managed Switch

Cisco IOS: System description

Interface gi1(AP_H1_1_CAP): Network traffic

		last	min	avg	max
🟩 Interface gi1(AP_H1_1_CAP): Bits received	[avg]	19.87 Kbps	2.89 Kbps	9.83 Kbps	606.04 Kbps
🟥 Interface gi1(AP_H1_1_CAP): Bits sent	[avg]	30.39 Kbps	7.8 Kbps	72.66 Kbps	5.85 Mbps
🟩 Interface gi1(AP_H1_1_CAP): Outbound packets with errors	[avg]	0	0	0	0
🟧 Interface gi1(AP_H1_1_CAP): Inbound packets with errors	[avg]	0	0	0	0
🟪 Interface gi1(AP_H1_1_CAP): Outbound packets discarded	[avg]	0	0	0	0
🟪 Interface gi1(AP_H1_1_CAP): Inbound packets discarded	[avg]	0	0	0	0

Example: Template Dashboards



Received

Interface gi1… 19.87 Kbps
Interface gi2… 30.98 Kbps
Interface gi3… 38.33 Kbps
Interface gi4… 5.22 Kbps
Interface gi5… 4.83 Kbps
Interface gi7… 134.26 Kbps
Interface gi1… 1.28 Kbps
Interface gi2… 6.90 Kbps
Interface gi4… 1.07 Kbps
Interface gi4… 14.03 Kbps
Interface gi4… 87.86 Mbps
Interface gi4… 723.59 Kbps
Interface gi5… 7.59 Mbps

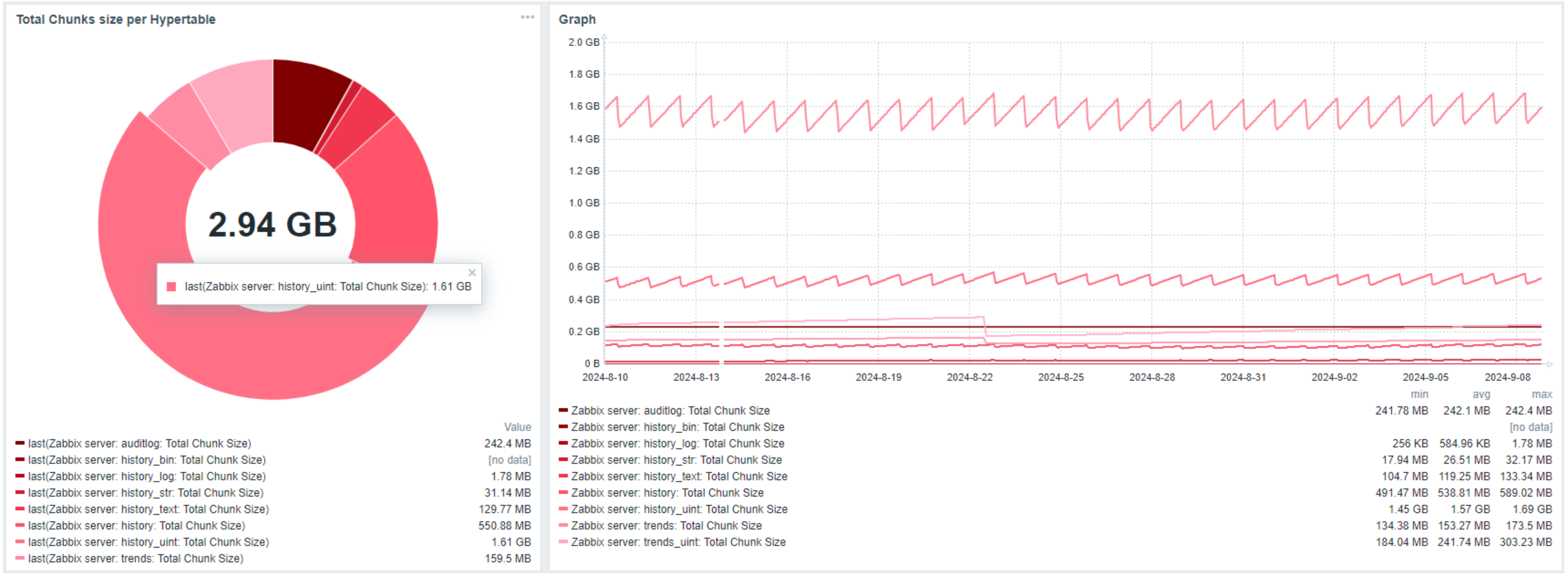
Sent

Interface gi1… 30.39 Kbps
Interface gi2… 742.51 Kbps
Interface gi3… 611.51 Kbps
Interface gi4… 19.06 Kbps
Interface gi5… 19.34 Kbps
Interface gi7… 609.75 Kbps
Interface gi1… 15.07 Kbps
Interface gi2… 16.90 Kbps
Interface gi4… 15.42 Kbps
Interface gi4… 56.00 bps
Interface gi4… 8.78 Mbps
Interface gi4… 25.62 Kbps
Interface gi5… 1.16 Mbps

Graph

initMAX

Example: Template Dashboards



Total Chunks size per Hypertable

2.94 GB

last(Zabbix server: history_uint: Total Chunk Size): 1.61 GB

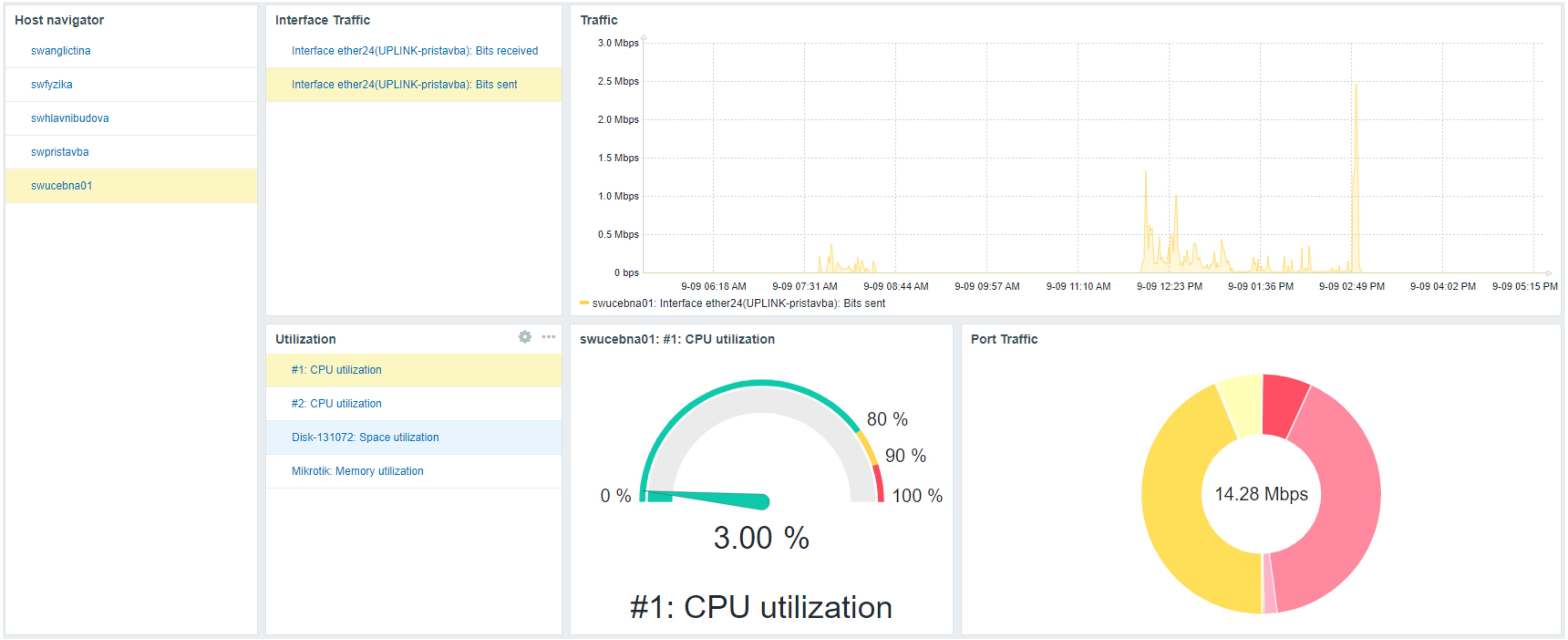
	Value
last(Zabbix server: auditlog: Total Chunk Size)	242.4 MB
last(Zabbix server: history_bin: Total Chunk Size)	[no data]
last(Zabbix server: history_log: Total Chunk Size)	1.78 MB
last(Zabbix server: history_str: Total Chunk Size)	31.14 MB
last(Zabbix server: history_text: Total Chunk Size)	129.77 MB
last(Zabbix server: history: Total Chunk Size)	550.88 MB
last(Zabbix server: history_uint: Total Chunk Size)	1.61 GB
last(Zabbix server: trends: Total Chunk Size)	159.5 MB

Graph

	min	avg	max
Zabbix server: auditlog: Total Chunk Size	241.78 MB	242.1 MB	242.4 MB
Zabbix server: history_bin: Total Chunk Size			[no data]
Zabbix server: history_log: Total Chunk Size	256 KB	584.96 KB	1.78 MB
Zabbix server: history_str: Total Chunk Size	17.94 MB	26.51 MB	32.17 MB
Zabbix server: history_text: Total Chunk Size	104.7 MB	119.25 MB	133.34 MB
Zabbix server: history: Total Chunk Size	491.47 MB	538.81 MB	589.02 MB
Zabbix server: history_uint: Total Chunk Size	1.45 GB	1.57 GB	1.69 GB
Zabbix server: trends: Total Chunk Size	134.38 MB	153.27 MB	173.5 MB
Zabbix server: trends_uint: Total Chunk Size	184.04 MB	241.74 MB	303.23 MB

ISP Alliance a. s. – Conference 2024

Last tip: Zabbix Dashboard with navigators



Host navigator
- swanglictina
- swfyzika
- swhlavnibudova
- swpristavba
- swucebna01

Interface Traffic
- Interface ether24(UPLINK-pristavba): Bits received
- Interface ether24(UPLINK-pristavba): Bits sent

Traffic

🟨 swucebna01: Interface ether24(UPLINK-pristavba): Bits sent

Utilization
- #1: CPU utilization
- #2: CPU utilization
- Disk-131072: Space utilization
- Mikrotik: Memory utilization

swucebna01: #1: CPU utilization

3.00 %

#1: CPU utilization

Port Traffic

Questions?

ISP Alliance a. s. – Conference 2024

Contact us:

Phone:	+420 800 244 442
Web:	https://www.initmax.cz
Email:	tomas.hermanek@initmax.cz
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184