AN OFFICIAL GUIDE TO MAKING AND MANAGING **GREAT TEMPLATES**

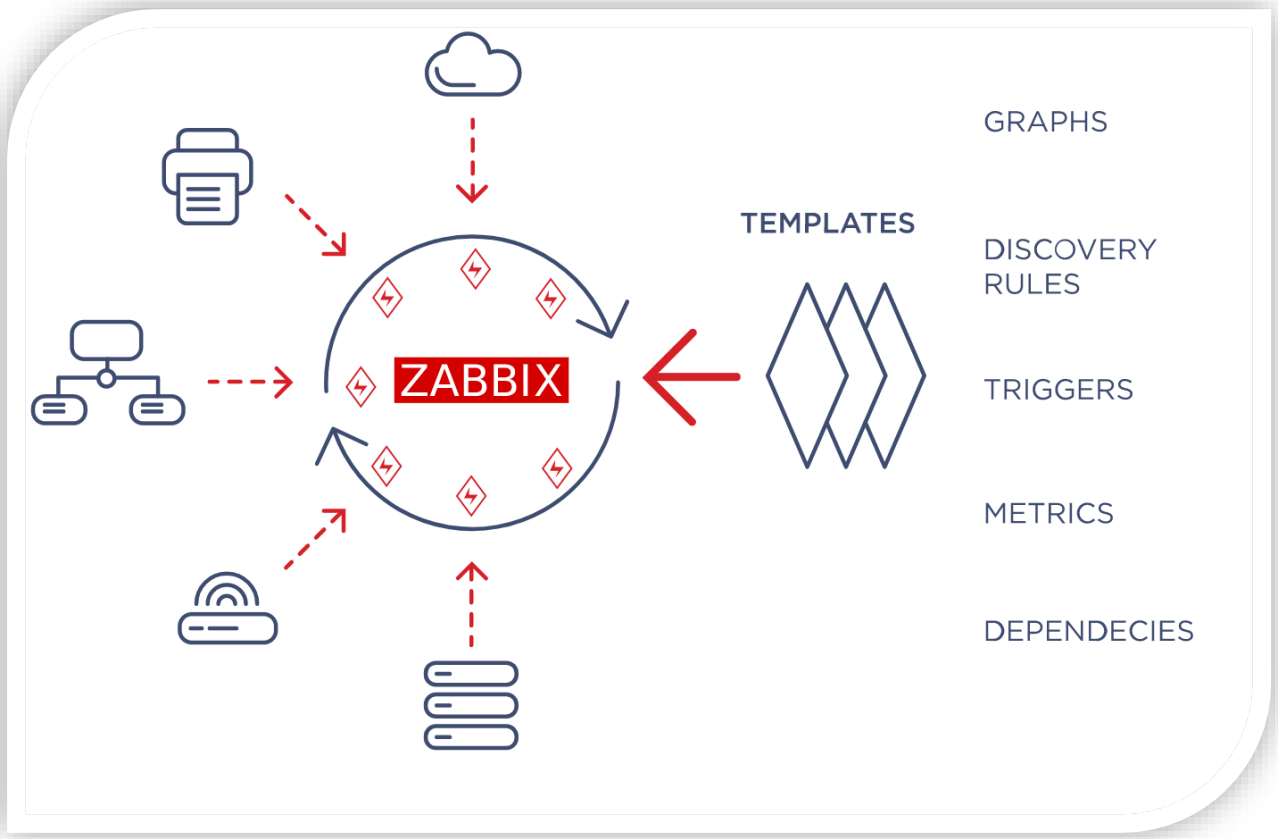
Vitaly Zhuravlev
ZABBIX Zabbix Solution Architect

ZABBIX SUMMIT '19

WHAT IS A TEMPLATE

AN OFFICIAL GUIDE TO MAKING AND MANAGING
GREAT TEMPLATES

What is a template



A template is the monitoring blueprint that can be used in order to centrally manage multiple hosts monitoring configuration.

Use templates out of the box

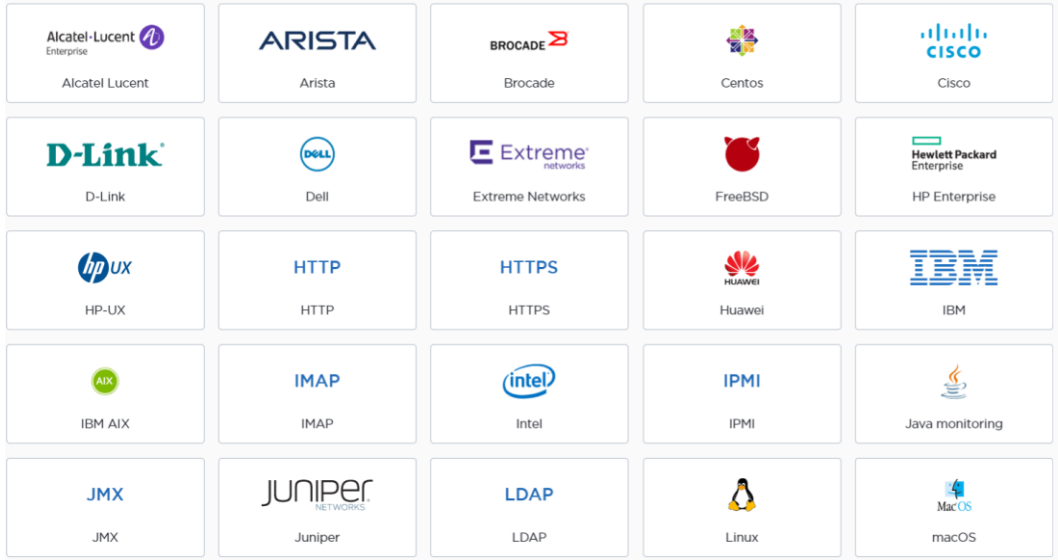
- Comes preinstalled in fresh Zabbix installations
- Available in git repo as XML files

  git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates

Use templates from the community

Share

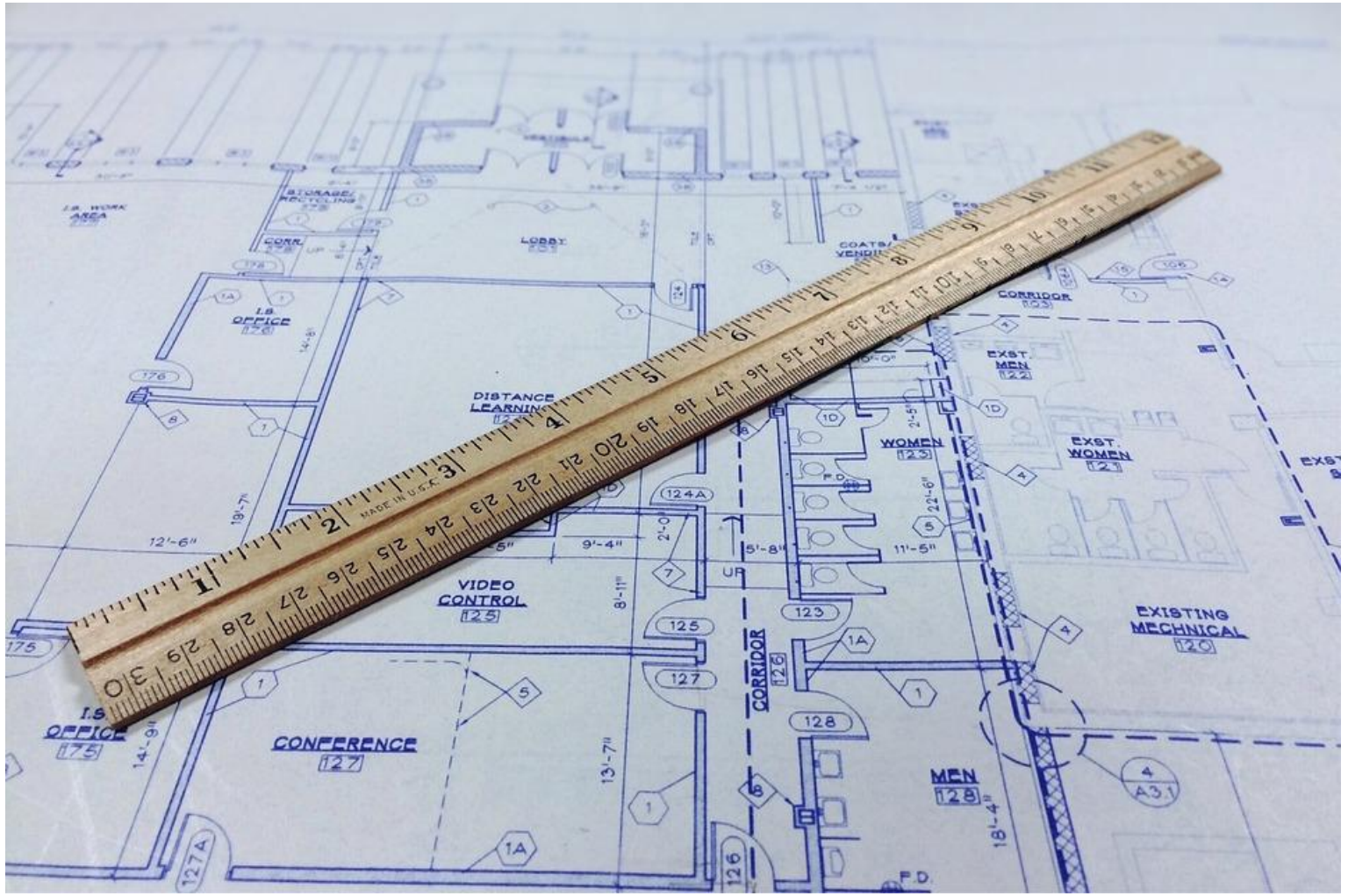
share.zabbix.com



zabbix.com/integrations/

GitHub

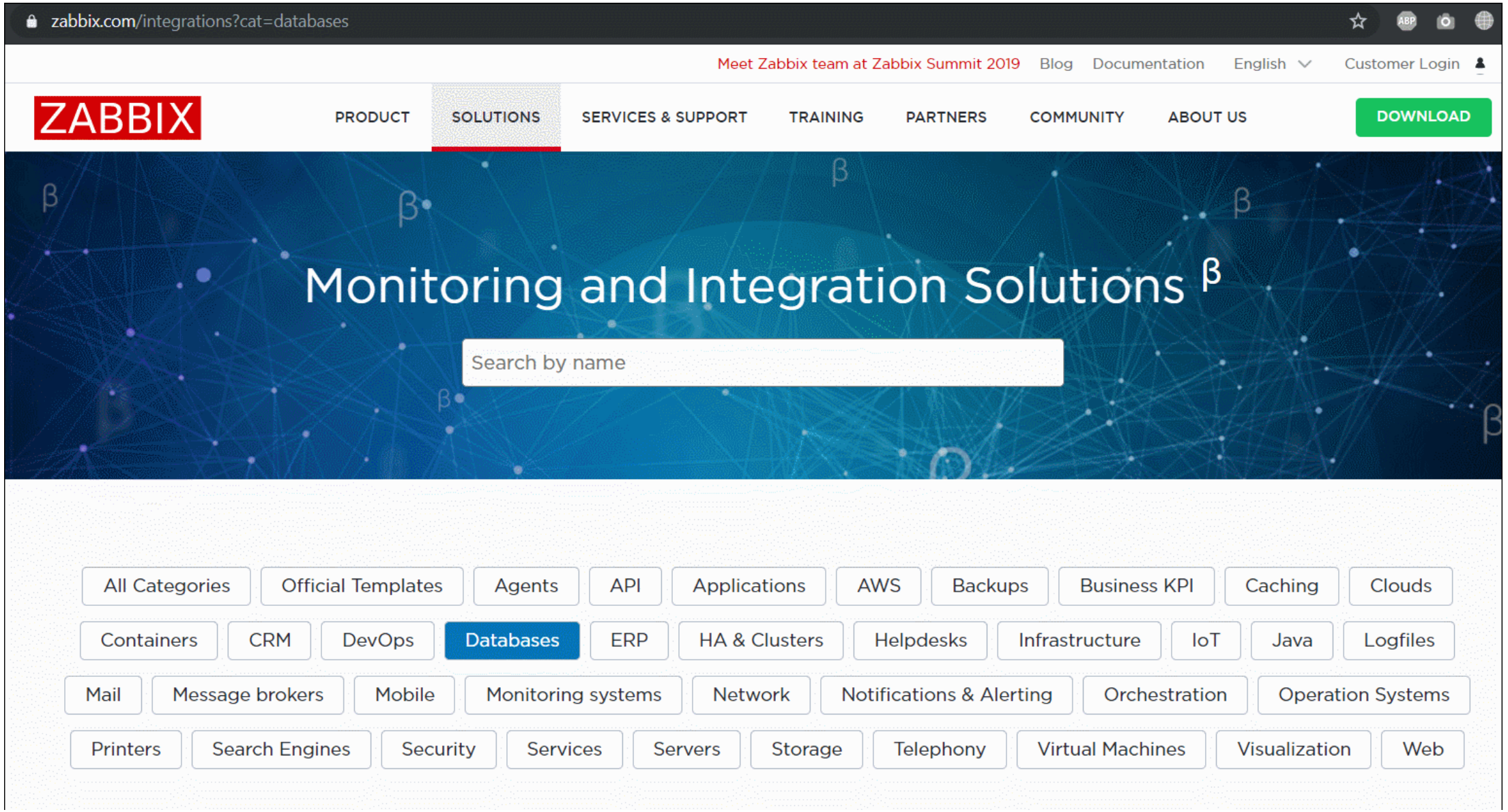
Design your own templates



ZABBIX SUMMIT '19

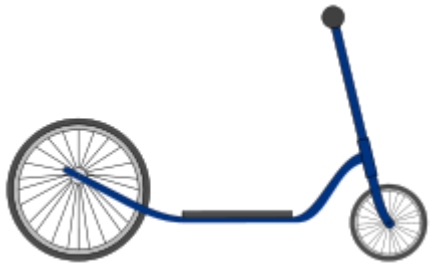
Current situation

What do we have now?


zabbix.com/integrations?cat=databases
Meet Zabbix team at Zabbix Summit 2019
Blog
Documentation
English
Customer Login
ZABBIX
PRODUCT
SOLUTIONS
SERVICES & SUPPORT
TRAINING
PARTNERS
COMMUNITY
ABOUT US
DOWNLOAD
Monitoring and Integration Solutions β
Search by name
All Categories
Official Templates
Agents
API
Applications
AWS
Backups
Business KPI
Caching
Clouds
Containers
CRM
DevOps
Databases
ERP
HA & Clusters
Helpdesks
Infrastructure
IoT
Java
Logfiles
Mail
Message brokers
Mobile
Monitoring systems
Network
Notifications & Alerting
Orchestration
Operation Systems
Printers
Search Engines
Security
Services
Servers
Storage
Telephony
Virtual Machines
Visualization
Web

Current situation

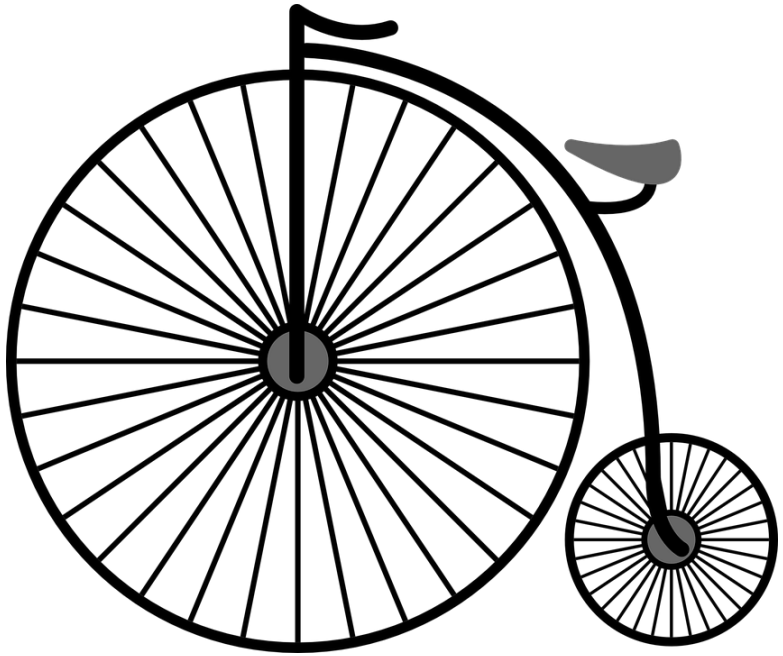
Reinventing the wheel



Percona MySQL Monitoring

Template DB MySQL

Template MySQL (800+ items)



Template Mysql/Mariadb monitoring

What else?

- No common style guidelines

- Templates could have unpredicted dependencies on literally everything




ba&sh

- Hard to contribute to others work

Let’s make it organized

1. Make it possible to avoid external dependencies
    - JavaScript preprocessing
    - HTTP agent
    - Golang zabbix-agent2 with plugins
    - More to come!

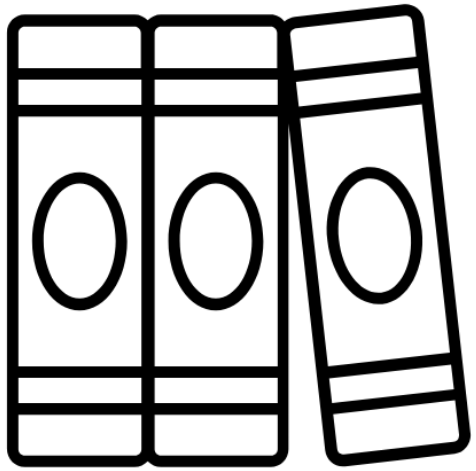
Let’s make it organized

2. Create templates for all basic components or “resources”

Let’s make it organized

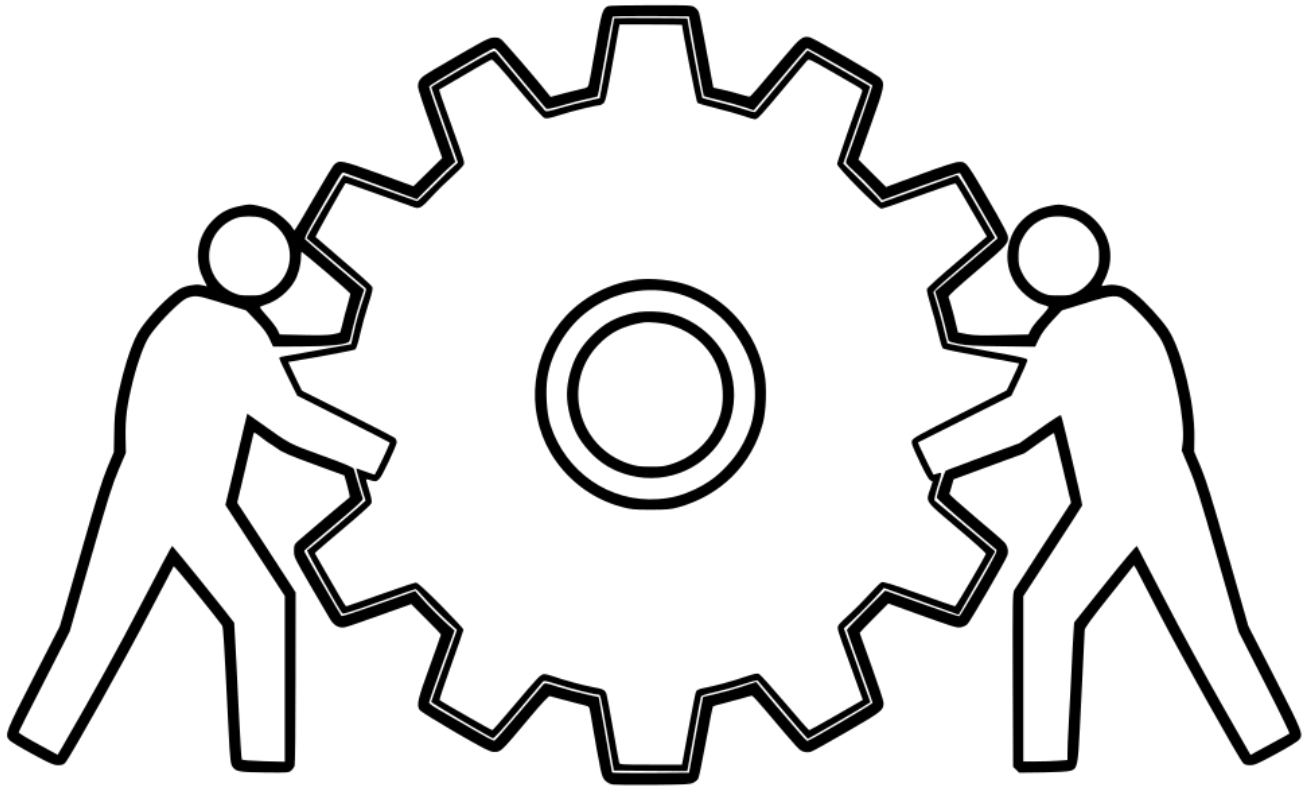
3. Template guidelines for everyone

- Naming conventions
- Best practices
- Snippets library



Let’s make it organized

4. Working together on templates should be easy



What is the “resource template” anyway?

RESOURCES AND SERVICES

AN OFFICIAL GUIDE TO MAKING AND MANAGING
GREAT TEMPLATES

Resources and services

Everything can be seen as a **service** or as a **resource**

Resources and services

Our monitoring approach

– monitor services first!

Resources and services

Resources are typical components reused in completely different IT architectures

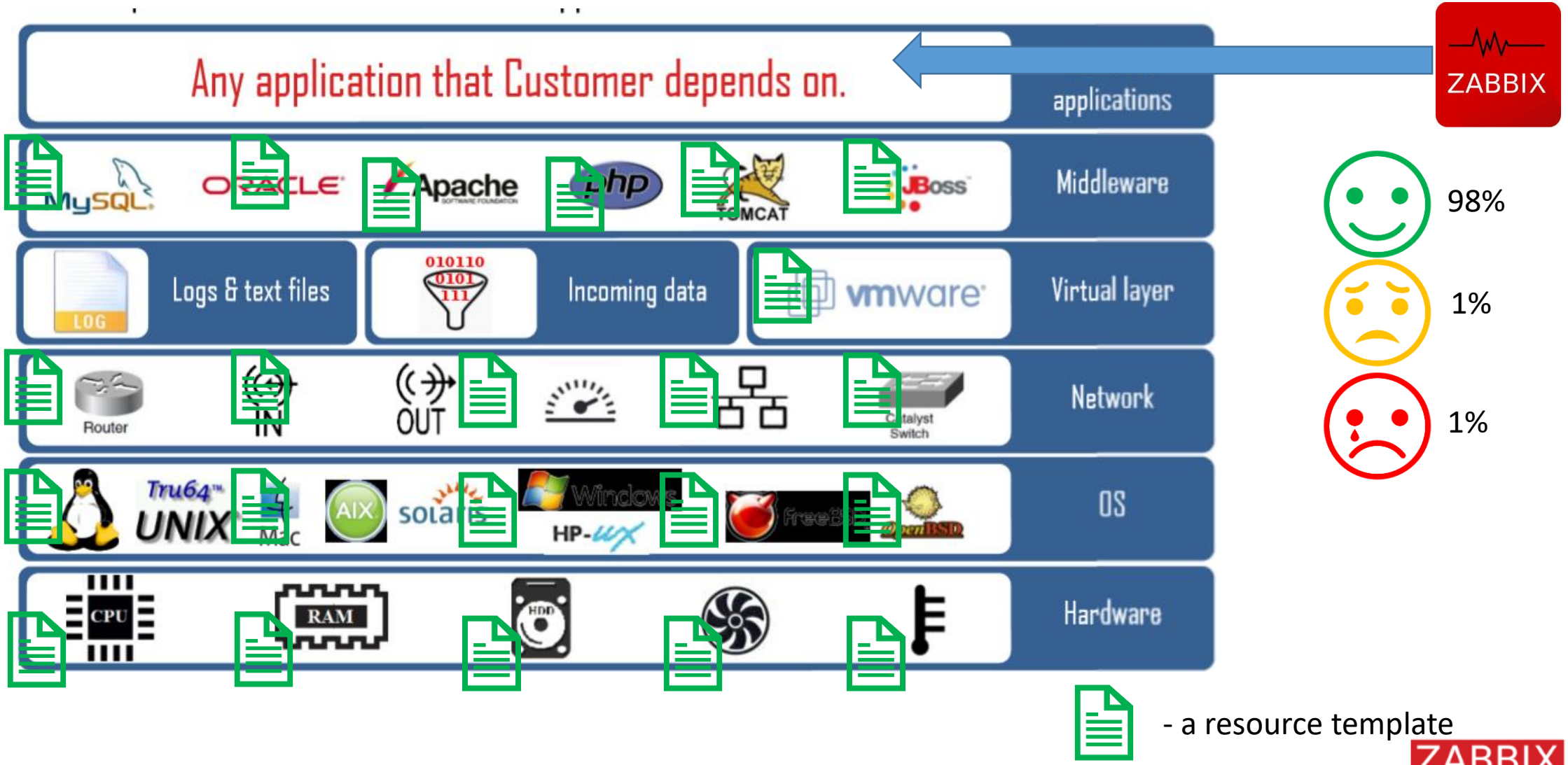
- databases: MySQL, PostgreSQL, MSSQL, Redis, Mongo.....
- middleware: RabbitMQ, Kafka, ActiveMQ
- servers: IBM, HP, Supermicro
- network devices: Cisco, Juniper, D-Link...
- OS: Linux, Windows, HP-UX, BSD, AIX
- cloud: AWS, Azure
- virtualization and containers: VMware, Hyper-V, Docker, Kubernetes, OpenShift, OpenStack

Resources and services

Once the service is monitored, move on to infrastructure or resource level, look for answers:

- WHY my service is not working at this moment?
- WHY my service is so slow?
- WHY my service didn’t perform well in the past
- My service is about to blow out let’s take some countermeasures!
- etc ☺

Resources and services



Resources and services

Let’s stop reinventing the wheel and start using the same **set of resource templates**

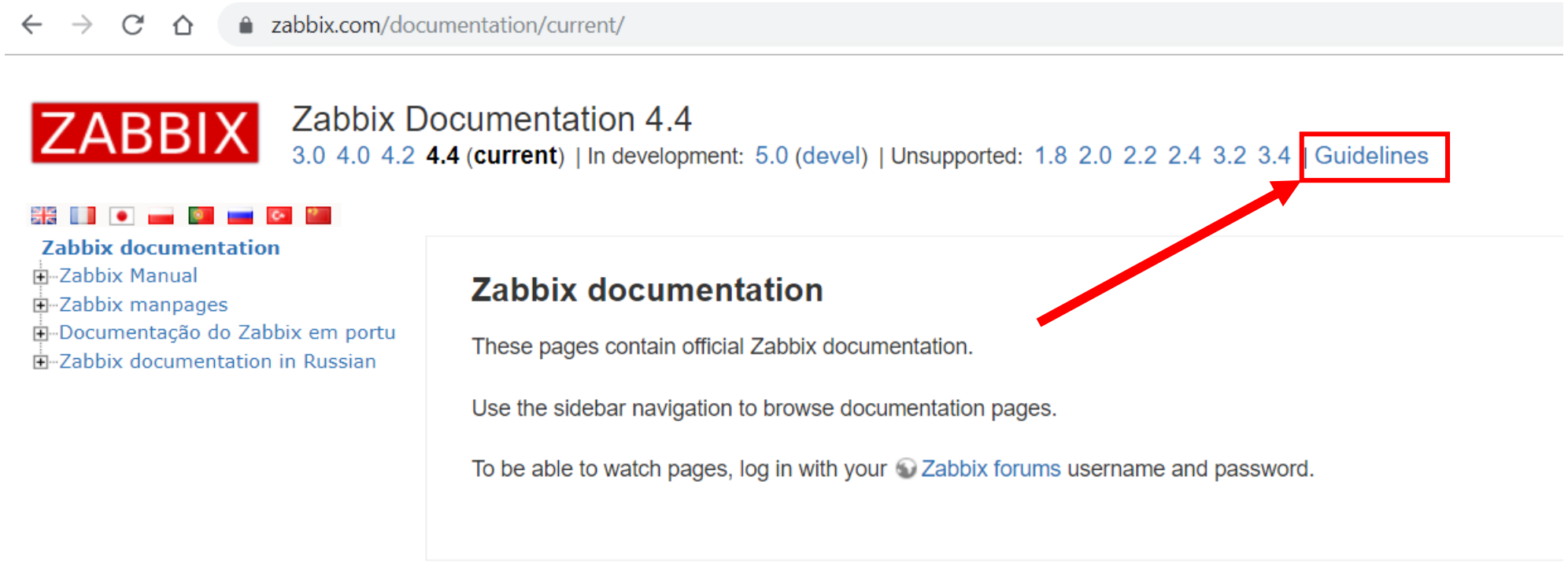
Then users can concentrate on monitoring of **business services –** things that are important in the first place

TEMPLATE GUIDELINES

AN OFFICIAL GUIDE TO MAKING AND MANAGING
GREAT TEMPLATES

Template guidelines

- Published in our documentation



zabbix.com/documentation/guidelines/templates

- Applies to all templates but concentrates on resource templates

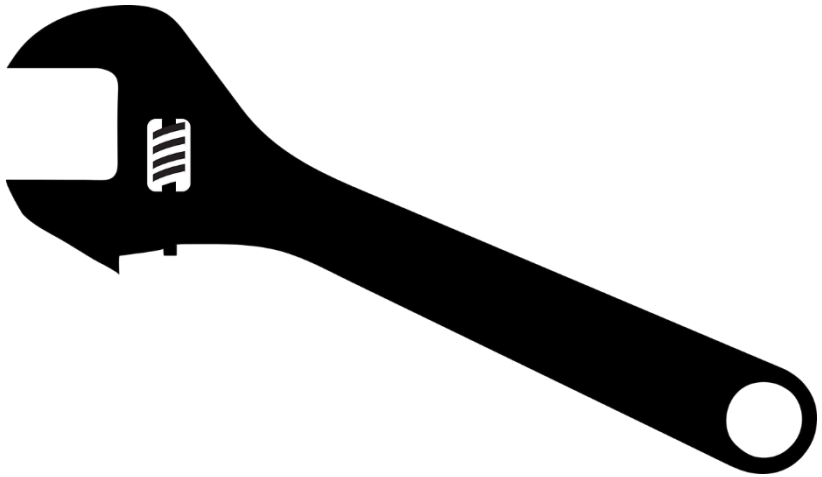
Template guidelines

Let’s make it clear. Two important questions.

- What makes a good resource template?
- What should be in the resource template?

What makes a good resource template?

1. Universal, one-size-fits-all



User macros

LLD (low-level discovery)

~~**Rare metrics**~~

What makes a good resource template?

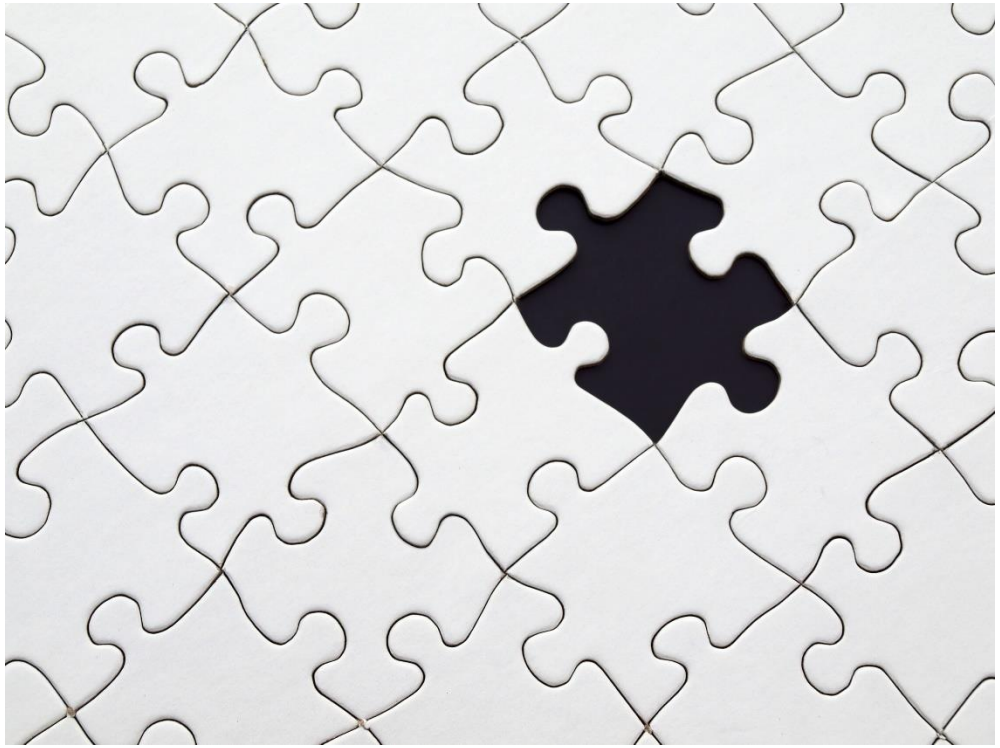
2. Expertise applied

- Key metrics picked, non-relevant filtered out
- Triggers for the most critical problems
    - But no problems “noise”

What makes a good resource template?

3. Modularity

Template is kept within a single resource



What should be in the resource template?

1. Availability monitoring ✓ and fault monitoring 
2. Performance monitoring 
3. State and inventory monitorir 
4. Security monitoring 

Template guidelines. Key tips

- Don’t overengineer it, just a template
- Use defaults when in doubt
  - for delay
  - for history and trends
- Instead spend your time on what's more important, e.g. what metrics to collect first

Template guidelines. Key tips

- English first (share this one)
  - Then localized copy of the template if necessary
- Avoid global regex and global macros
- Avoid passwords in user macros
- Include visualization
  - Screens and graphs are the must

Template guidelines. Key tips

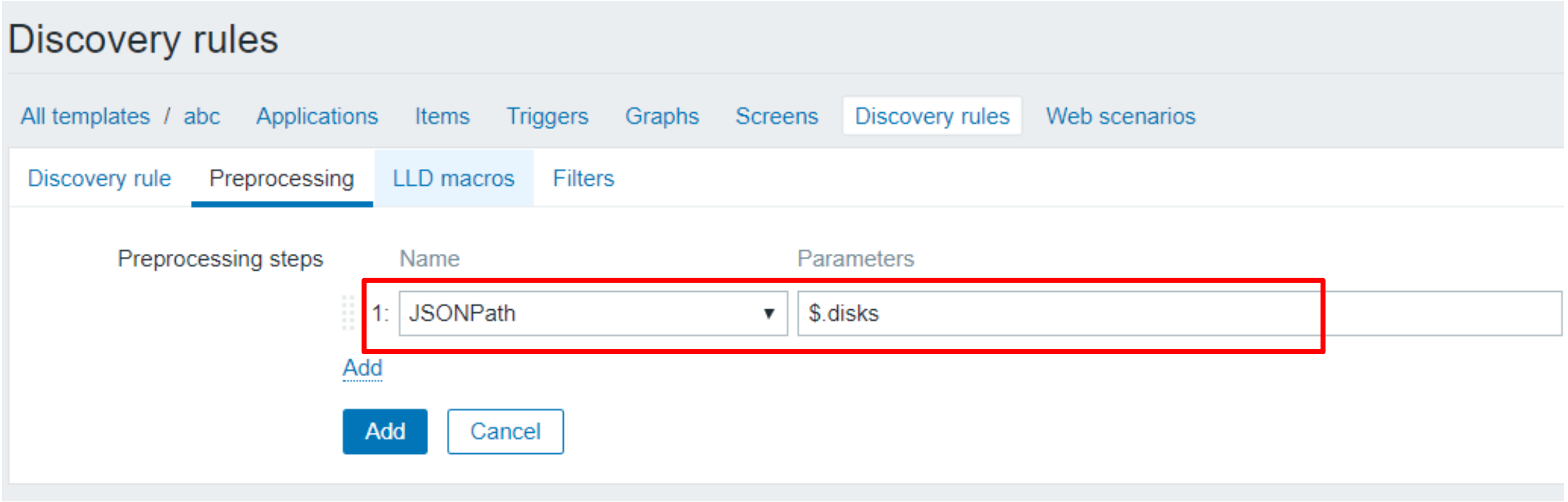
- KISS - keep it simple, stupid
  - KISS for the end template users
  - even if under the hood it is more complex 😐
- Documentation!
  - README.md near the XML file,
  - fill an item or a trigger description
- Avoid external dependencies as much as you can
  Use preprocessing, JS..., build new zabbix-agent2 plugins ☺

Avoid external dependencies

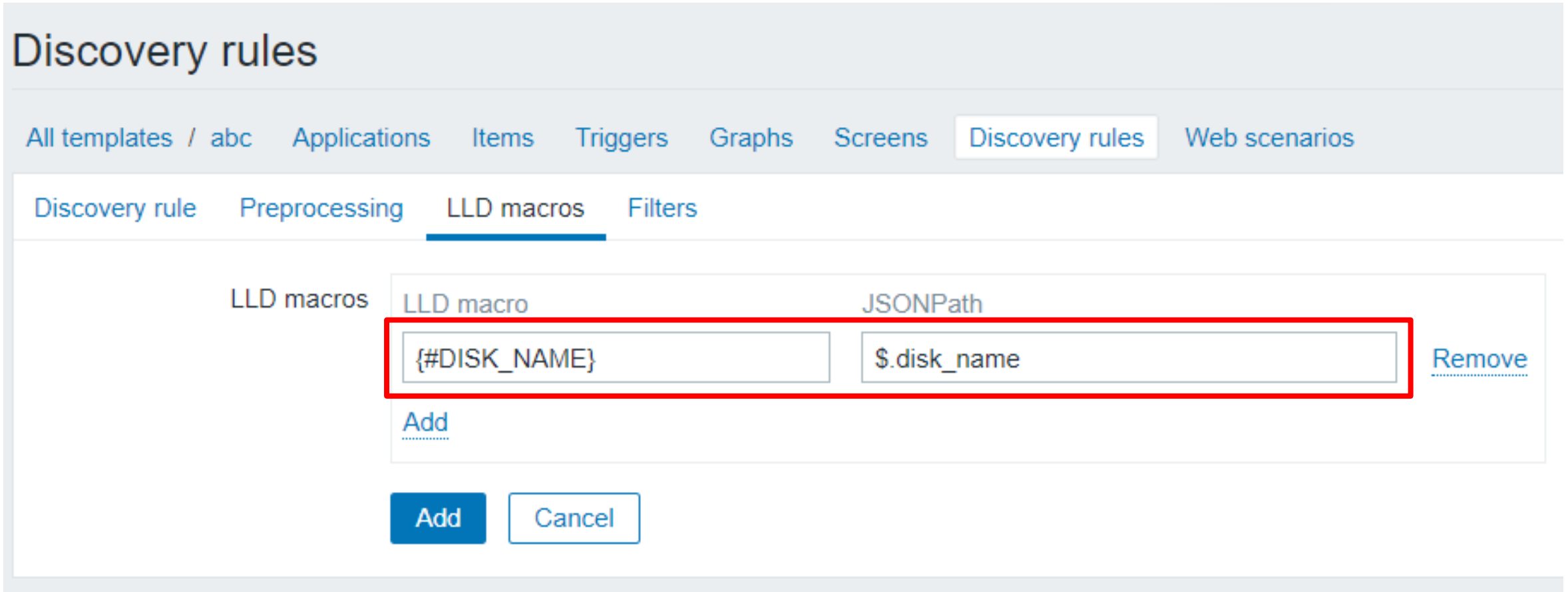
Now (4.2) you can do discovery out of any JSON array

```
{
 "disks":[
  {
   "disk_name": "disk1",
   "disk_status": 1,
  },
  {
   "disk_name": "disk2",
   "disk_status": 0
  }
 ]
}
```

LLD preprocessing



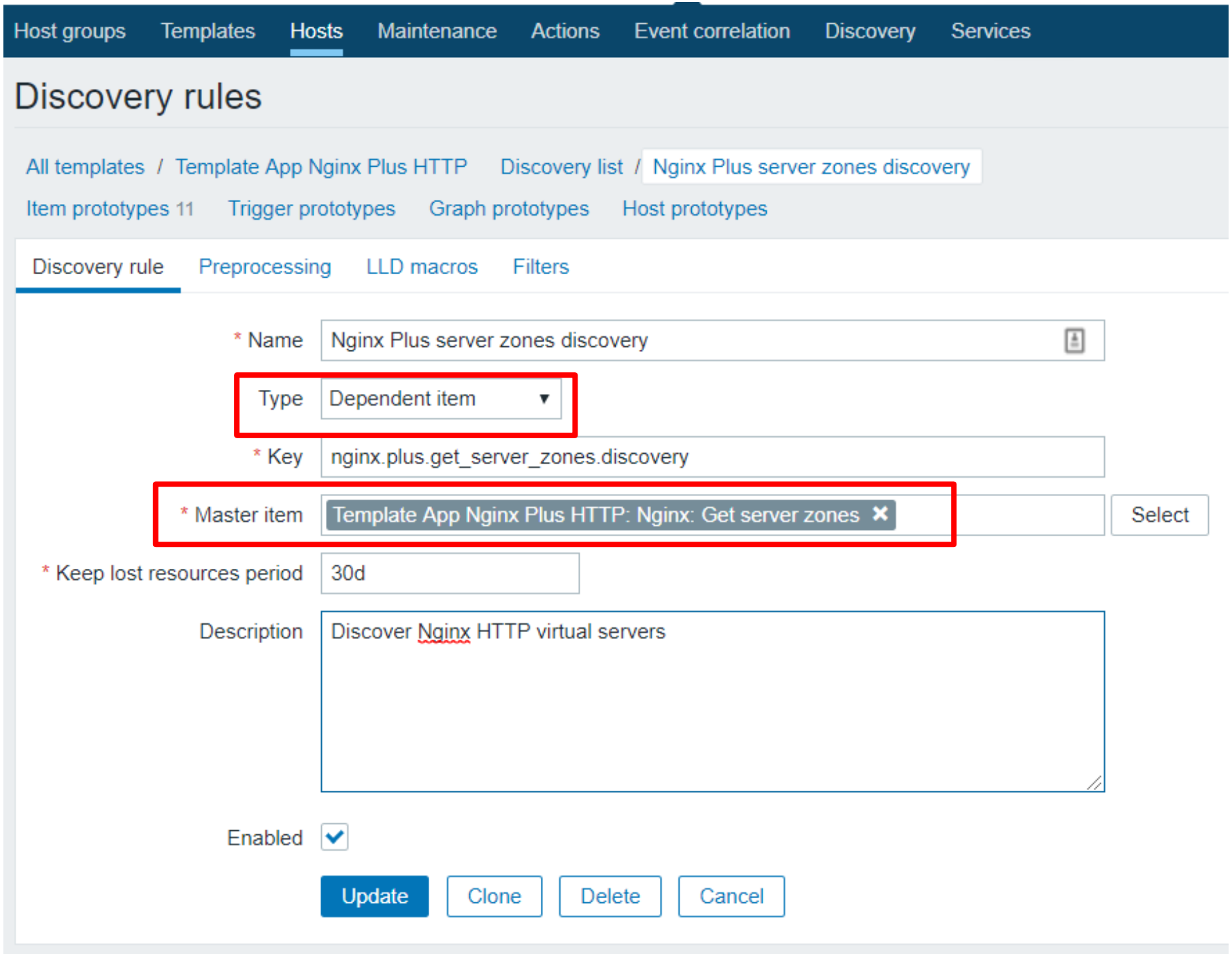
LLD macros



Convert JSON object to JSON array

http://demo.nginx.com/api/3/http/server_zones


```
{
  "hg.nginx.org": {
    "processing": 0,
    "requests": 175276,
    "responses": {
      "1xx": 0, "2xx": 162948, "3xx": 10117, "4xx": 2125, "5xx": 8, "total": 175198
    },
    "discarded": 78,
    "received": 50484208,
    "sent": 7356417338
  },
 "trac.nginx.org": {
    "processing": 7,
    "requests": 448613,
    "responses": {
      "1xx": 0, "2xx": 305562, "3xx": 87065, "4xx": 23136, "5xx": 5127, "total": 420890
    },
    "discarded": 27716,
    "received": 137307886,
    "sent": 3989556941
  }
}
```



Discovery rules

All templates / Template App Nginx Plus HTTP Discovery list / Nginx Plus server zones discovery

Item prototypes 11 Trigger prototypes Graph prototypes Host prototypes

Discovery rule Preprocessing LLD macros Filters

Preprocessing steps Name Parameters Custom on fail Actions

JavaScript

```
function (value) {
    //parsing NGINX plus output like in footer:
    output = Object.keys(JSON.parse(value)).map(function(zone){
        return {"{#NGINX_ZONE}": zone}
    })
    return JSON.stringify({"data": output})
}
```

65354 symbols remaining

```
//parsing NGINX plus output:
output = Object.keys(JSON.parse(value)).map(function(zone){
    return {"{#NGINX_ZONE}": zone}
})
return JSON.stringify({"data": output})
```

Item prototypes

Create item prototype

All templates / Template App Nginx Plus HTTP Discovery list / Nginx Plus server zones discovery

Item prototypes 11 Trigger prototypes Graph prototypes Host prototypes

Wizard	Name ▲	Key	Interval	History	Trends	Type	Applications	Create enabled
•••	Nginx: Get server zones: {#NGINX_ZONE}: Discarded	nginx.plus.discarded[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Processing	nginx.plus.processing[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Received	nginx.plus.received[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Requests	nginx.plus.requests[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Responses 1xx	nginx.plus.responses.1xx[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Responses 2xx	nginx.plus.responses.2xx[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Responses 3xx	nginx.plus.responses.3xx[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Responses 4xx	nginx.plus.responses.4xx[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Responses 5xx	nginx.plus.responses.5xx[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Responses total	nginx.plus.responses.total[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes
•••	Nginx: Get server zones: {#NGINX_ZONE}: Sent	nginx.plus.sent[{#NGINX_ZONE}]		90d	365d	Dependent item		Yes

Displaying 11 of 11 found

Structured text can be used as well: vfs.file.contents[/proc/diskstats]

```
7       0 loop0 2 0 10 0 0 0 0 0 0 0 0
7       1 loop1 0 0 0 0 0 0 0 0 0 0 0
7       2 loop2 0 0 0 0 0 0 0 0 0 0 0
7       3 loop3 0 0 0 0 0 0 0 0 0 0 0
7       4 loop4 0 0 0 0 0 0 0 0 0 0 0
7       5 loop5 0 0 0 0 0 0 0 0 0 0 0
7       6 loop6 0 0 0 0 0 0 0 0 0 0 0
7       7 loop7 0 0 0 0 0 0 0 0 0 0 0
8       0 sda 192218 21315 11221888 13020540 28630719 8482221 801446972 388811708 0 265066852
401774948
8       1 sda1 252 59 11294 5424 6 0 12 464 0 4160 5888
8       2 sda2 4 0 8 72 0 0 0 0 0 72 72
8       5 sda5 191918 21256 11208378 13014352 22872982 8482221 801446960 215739516 0 99497600
228699704
252       0 dm-0 186763 0 10985130 22979168 31930494 0 799946248 396490524 0 265080476 419505356
252       1 dm-1 26897 0 220608 688352 187589 0 1500712 23501956 0 212608 24190464
```

eps Name Parameters Custom on fail Actions

1

Add

U

JavaScript

```
function (value) {
  1  var parsed = value.split("\n").reduce(function(acc, x, i) {
  2    parts = x.trim().split(/ +/)
  3    acc["values"][parts[2]] = parts
  4    acc["lld"].push({"{#DEVNAME}":parts[2]})
  5    return acc;
  6  }, {"values":{}, "lld": []});
  7
  8  return JSON.stringify(p
}
```

65292 symbols remaining

```
var parsed = value.split("\n").reduce(function(acc, x, i) {
  parts = x.trim().split(/ +/)
  acc["values"][parts[2]] = parts
  acc["lld"].push({"{#DEVNAME}":parts[2]})
  return acc;
}, {"values":{}, "lld": []});

return JSON.stringify(parsed);
```

{"values":{"loop0":["7","0","loop0","2","0","10","0","0","0","0","0","0","0","0"],"loop1":["7","1","loop1","0","0","0","0","0","0","0","0","0","0","0"],"loop2":["7","2","loop2","0","0","0","0","0","0","0","0","0","0","0"],"loop3":["7","3","loop3","0","0","0","0","0","0","0","0","0","0","0"],"loop4":["7","4","loop4","0","0","0","0","0","0","0","0","0","0","0"],"loop5":["7","5","loop5","0","0","0","0","0","0","0","0","0","0","0"],"loop6":["7","6","loop6","0","0","0","0","0","0","0","0","0","0","0"],"loop7":["7","7","loop7","0","0","0","0","0","0","0","0","0","0","0"],"sda":["8","0","sda","192319","21316","11225512","13022788","28634853","8483355","801573324","388890252","0","265106688","401855724"],"sda1":["8","1","sda1","252","59","11294","5424","6","0","12","464","0","4160","5888"],"sda2":["8","2","sda2","4","0","8","72","0","0","0","0","0","72","72"],"sda5":["8","5","sda5","192019","21257","11212002","13016600","22876273","8483355","801573312","215792244","0","99514184","228754668"],"dm-0":["252","0","dm-0","186863","0","10988738","22981380","31935086","0","800072600","396577656","0","265120280","419594700"],"dm-1":["252","1","dm-1","26899","0","220624","688400","187589","0","1500712","23501956","0","212632","24190512"]},

"lld":[{"{#DEVNAME}":"loop0"},{"{#DEVNAME}":"loop1"},{"{#DEVNAME}":"loop2"},{"{#DEVNAME}":"loop3"},{"{#DEVNAME}":"loop4"},{"{#DEVNAME}":"loop5"},{"{#DEVNAME}":"loop6"},{"{#DEVNAME}":"loop7"},{"{#DEVNAME}":"sda"},{"{#DEVNAME}":"sda1"},{"{#DEVNAME}":"sda2"},{"{#DEVNAME}":"sda5"},{"{#DEVNAME}":"dm-0"},{"{#DEVNAME}":"dm-1"}]}

Discovery rules

All templates / Template OS Linux Zabbix agent Discovery list / Block devices discovery

Item prototypes 8 Trigger prototypes 1 Graph prototypes 2 Host prototypes

Discovery rule Preprocessing LLD macros Filters

* Name: Block devices discovery

Type: Dependent item

* Key: vfs.dev.discovery

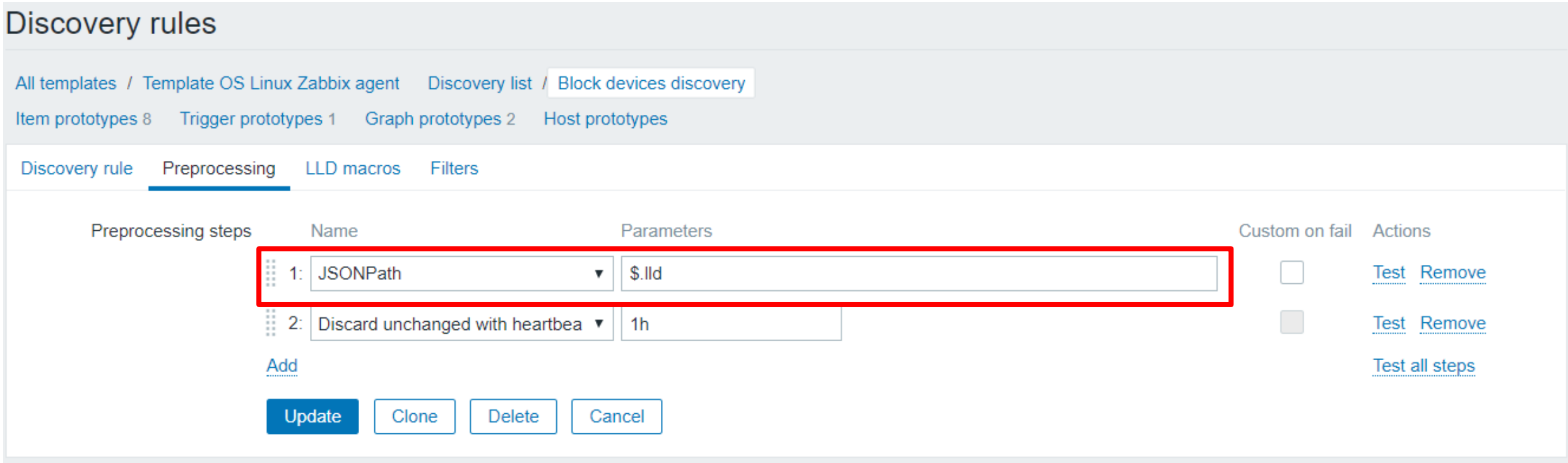
* Master item: Template OS Linux Zabbix agent: Get /proc/diskstats Select

* Keep lost resources period: 30d

Description:

Enabled

Update Clone Delete Cancel



Item prototypes

Create item prototype

All templates / Template OS Linux Zabbix agent Discovery list / Block devices discovery

Item prototypes 8 Trigger prototypes 1 Graph prototypes 2 Host prototypes

Wizard	Name ▲	Key	Interval	History	Trends	Type	Applications	Create enabled
•••	Get /proc/diskstats: {#DEVNAME}: Disk average queue size (avgqu-sz)	vfs.dev.queue_size[{#DEVNAME}]		90d	365d	Dependent item		Yes
•••	Get /proc/diskstats: {#DEVNAME}: Disk read rate	vfs.dev.read.rate[{#DEVNAME}]		90d	365d	Dependent item		Yes
•••	{#DEVNAME}: Disk read request avg waiting time (r_await)	vfs.dev.read.await[{#DEVNAME}]	1m	90d	365d	Calculated		Yes
•••	Get /proc/diskstats: {#DEVNAME}: Disk read time (rate)	vfs.dev.read.time.rate[{#DEVNAME}]		0	365d	Dependent item	Zabbix raw items	Yes
•••	Get /proc/diskstats: {#DEVNAME}: Disk utilization	vfs.dev.util[{#DEVNAME}]		90d	365d	Dependent item		Yes
•••	Get /proc/diskstats: {#DEVNAME}: Disk write rate	vfs.dev.write.rate[{#DEVNAME}]		90d	365d	Dependent item		Yes
•••	{#DEVNAME}: Disk write request avg waiting time (w_await)	vfs.dev.write.await[{#DEVNAME}]	1m	90d	365d	Calculated		Yes
•••	Get /proc/diskstats: {#DEVNAME}: Disk write time (rate)	vfs.dev.write.time.rate[{#DEVNAME}]		0	365d	Dependent item	Zabbix raw items	Yes

Displaying 8 of 8 found

0 selected Create enabled Create disabled Mass update Delete

Disk sda (6 Items)				
sda: Disk average queue size (avgqu-sz)	2019-08-19 14:04:16	0.31	-0.3	Graph
sda: Disk read rate	2019-08-19 14:04:16	8.51 r/s	-1.42 r/s	Graph
sda: Disk read request avg waiting time (r_await)	2019-08-19 14:04:29	0.59 ms	+0.08 ms	Graph
sda: Disk utilization	2019-08-19 14:04:16	24.04 %	-5.22 %	Graph
sda: Disk write rate	2019-08-19 14:04:16	19.02 w/s	-6.61 w/s	Graph
sda: Disk write request avg waiting time (w_await)	2019-08-19 14:04:32	15.62 ms	+4.03 ms	Graph

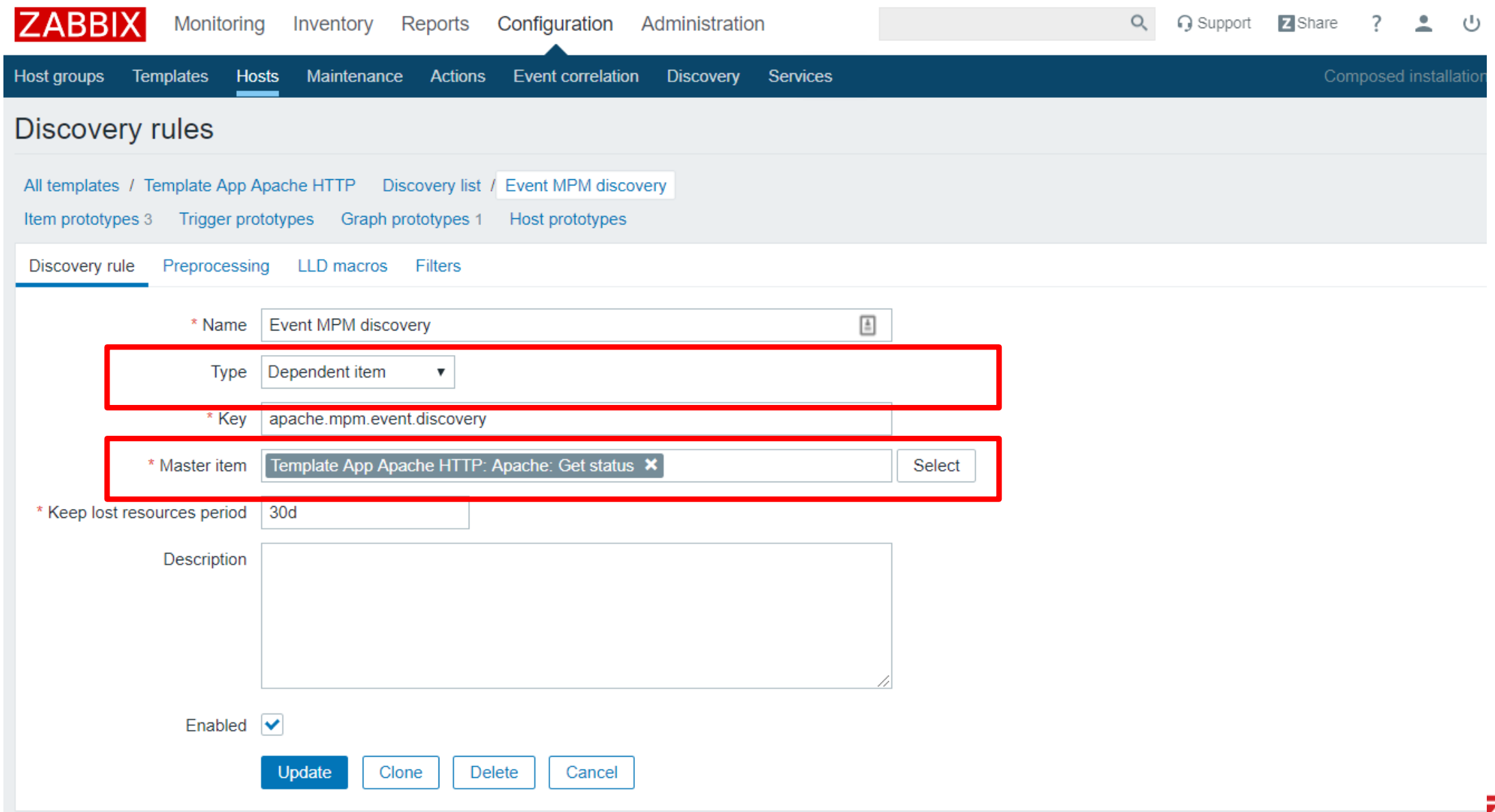
And more guidelines

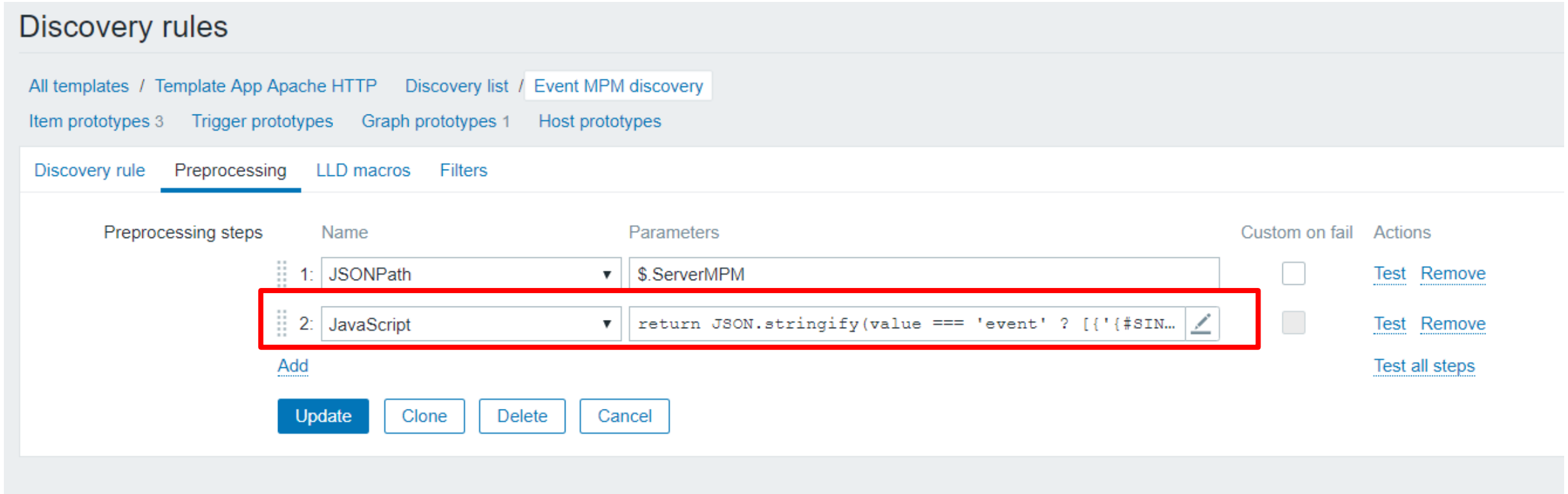
- All items and triggers should be enabled
- Avoid unsupported items
  - For that use LLD (low level discovey)

Singleton LLD

Apache HTTP as an example

127.0.0.1
ServerVersion: Apache/2.4.41 (Unix)
ServerMPM: event
Server Built: Aug 14 2019 00:35:10
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 189613
……..
Processes: 4
Stopping: 0
BusyWorkers: 7
IdleWorkers: 93
ConnsTotal: 13
ConnsAsyncWriting: 0
ConnsAsyncKeepAlive: 5
ConnsAsyncClosing: 0
Scoreboard:
_____W_____W_____
_____LW____W_____W_W_____.....
.....
.....





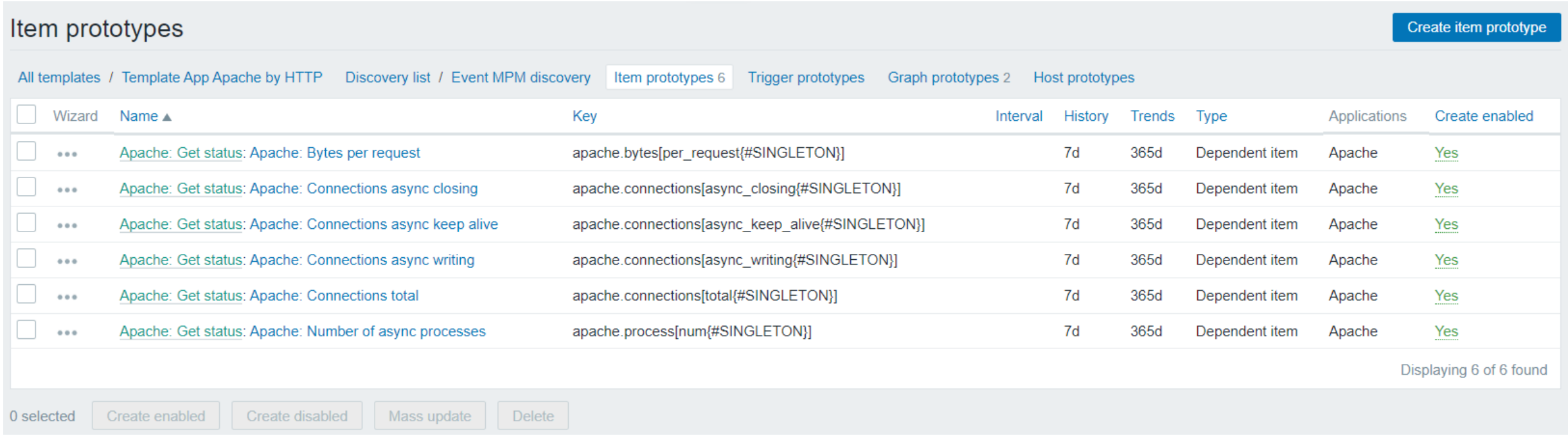
JavaScript

```
function (value) {
    1  return JSON.stringify(value === 'event' ? [{'{#SINGLETON}': ''}] : []);

}
```

65464 symbols remaining

Apache HTTP server. Use Singleton LLD for items



Item prototypes

Create item prototype

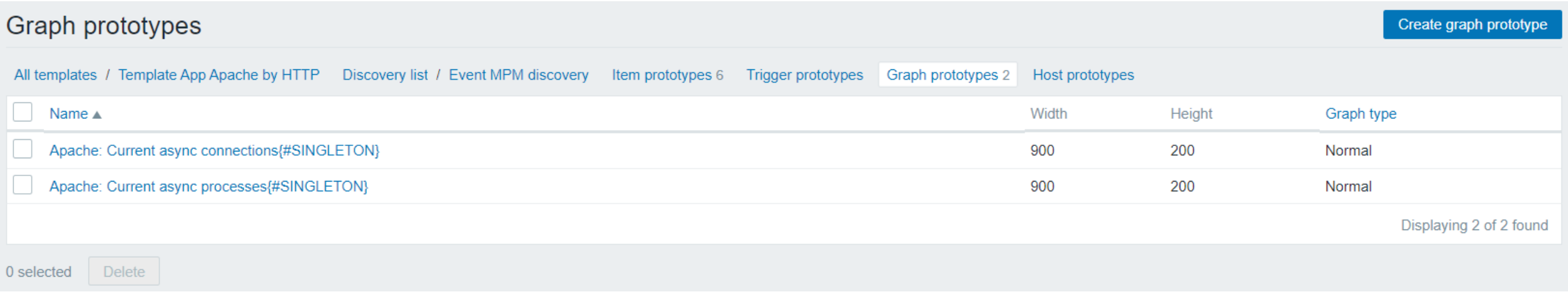
All templates / Template App Apache by HTTP | Discovery list / Event MPM discovery | Item prototypes 6 | Trigger prototypes | Graph prototypes 2 | Host prototypes

Wizard	Name ▲	Key	Interval	History	Trends	Type	Applications	Create enabled
•••	Apache: Get status: Apache: Bytes per request	apache.bytes[per_request{#SINGLETON}]		7d	365d	Dependent item	Apache	Yes
•••	Apache: Get status: Apache: Connections async closing	apache.connections[async_closing{#SINGLETON}]		7d	365d	Dependent item	Apache	Yes
•••	Apache: Get status: Apache: Connections async keep alive	apache.connections[async_keep_alive{#SINGLETON}]		7d	365d	Dependent item	Apache	Yes
•••	Apache: Get status: Apache: Connections async writing	apache.connections[async_writing{#SINGLETON}]		7d	365d	Dependent item	Apache	Yes
•••	Apache: Get status: Apache: Connections total	apache.connections[total{#SINGLETON}]		7d	365d	Dependent item	Apache	Yes
•••	Apache: Get status: Apache: Number of async processes	apache.process[num{#SINGLETON}]		7d	365d	Dependent item	Apache	Yes

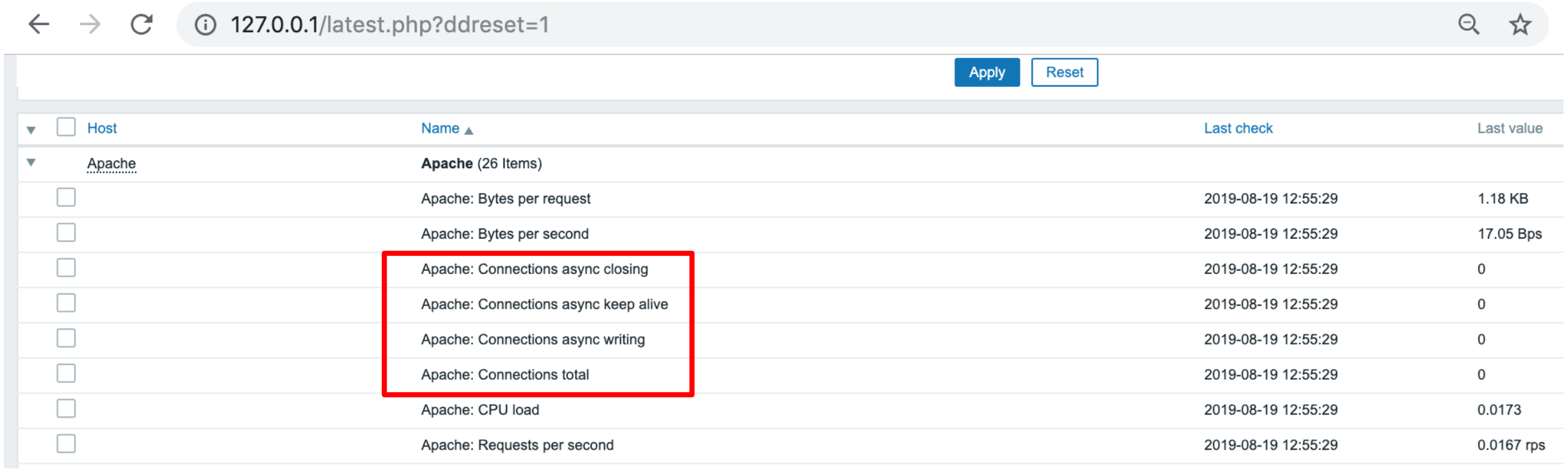
Displaying 6 of 6 found

0 selected | Create enabled | Create disabled | Mass update | Delete

Apache HTTP server. Use Singleton LLD for items and graphs



Apache HTTP server. Use Singleton LLD for items and graphs



Singleton LLD. Results

- Gives regular items, triggers, graphs
- They are created ONLY if JavaScript preprocessing filter would allow this. (in our case = ServerMPM: event)
- Universal template without any unsupported items

Moving on…. More guidelines

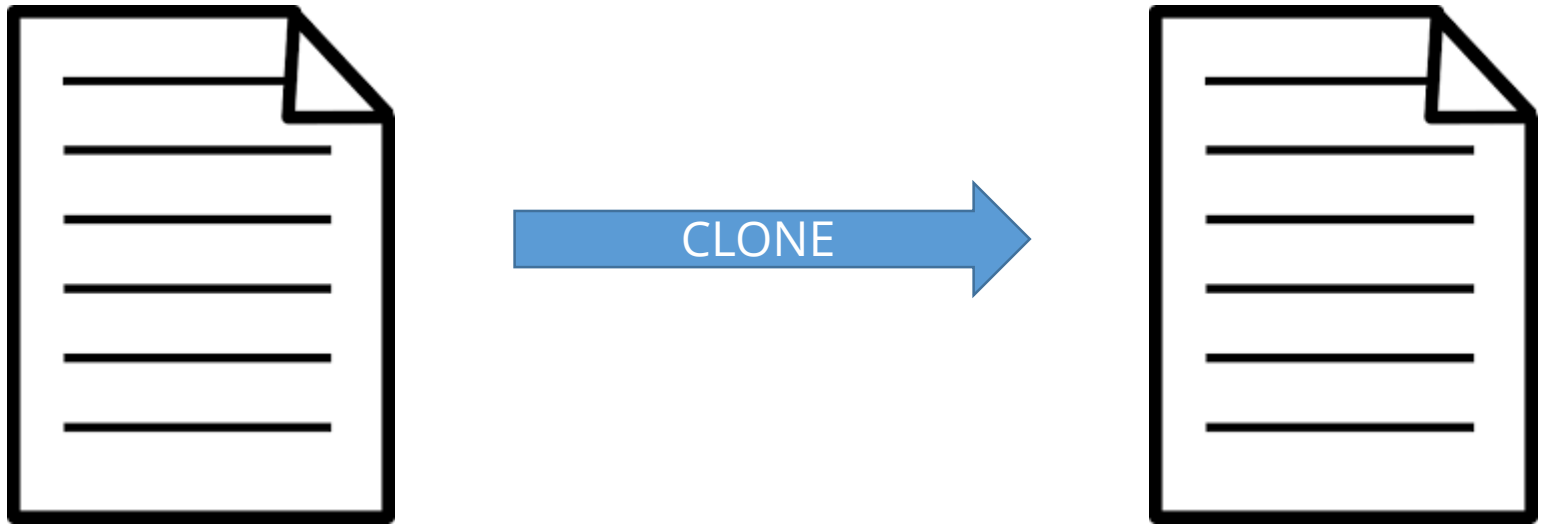
- User macros. Should be everywhere:
  - in trigger expressions
  - in item keys for flexible connections
  - LLD filters

Templates are truly universal with macros

Quick example.

- I have a template to monitor network interfaces.
- Most of the time, for my **core network devices,** I want to discover and monitor all network interfaces
- However, for **access layer switches**, I am only interested in interfaces with description (IFDESCR) that says “UPLINK”

How we used to do it:



Template Module Interfaces

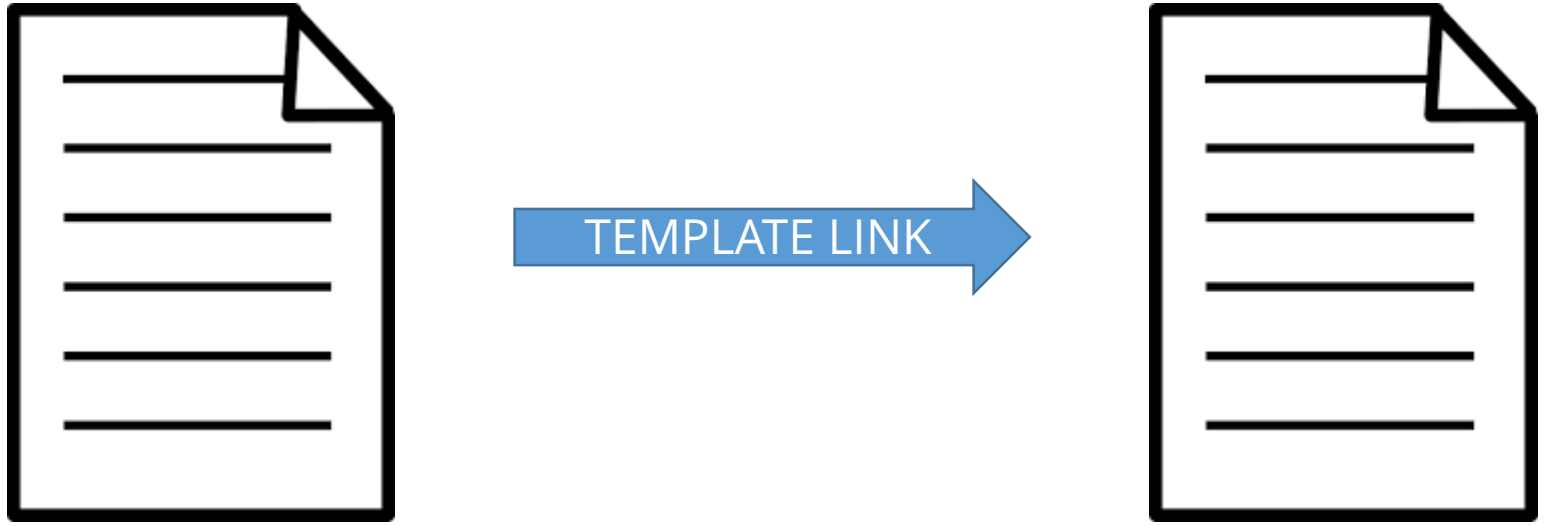
Template Module Interfaces for access layer

Templates are truly universal with macros

Quick example.

- I have a template to monitor network interfaces.
- Most of the time, for my **core network devices,** I want to discover and monitor all network interfaces
- However, for **access layer switches**, I am only interested in interfaces with description (IFDESCR) that says “UPLINK”

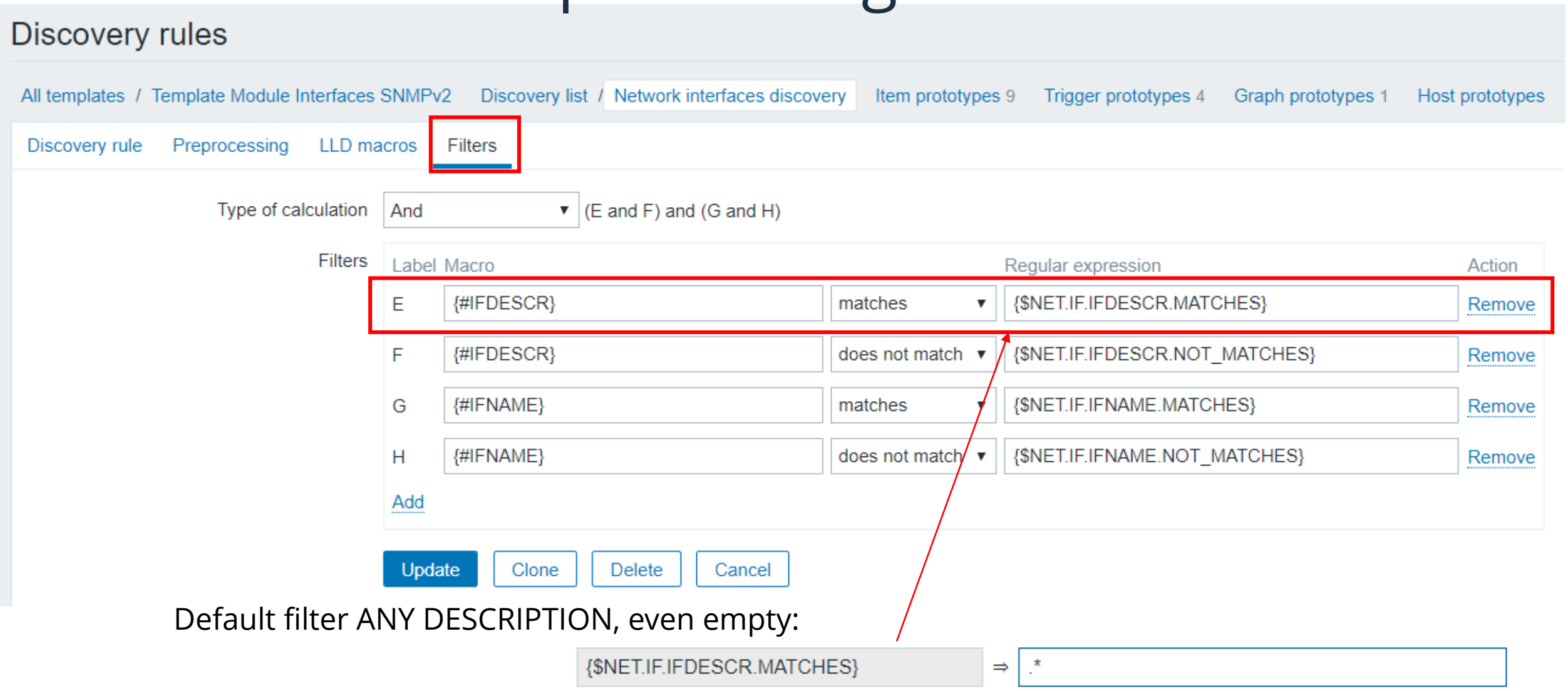
Another way to do it using macros:



Template Module Interfaces

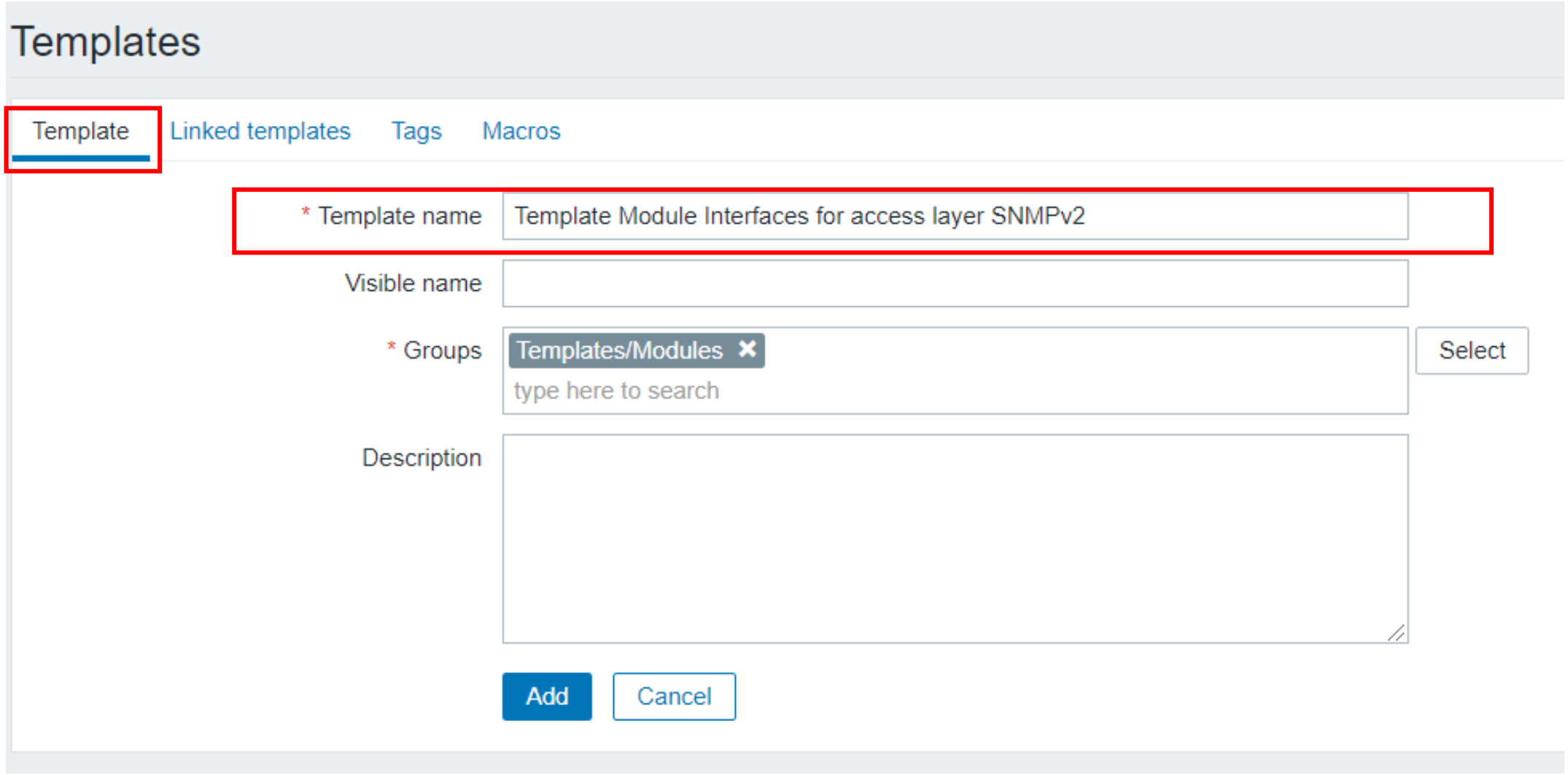
Template Module Interfaces for access layer

Filter of core template using macros

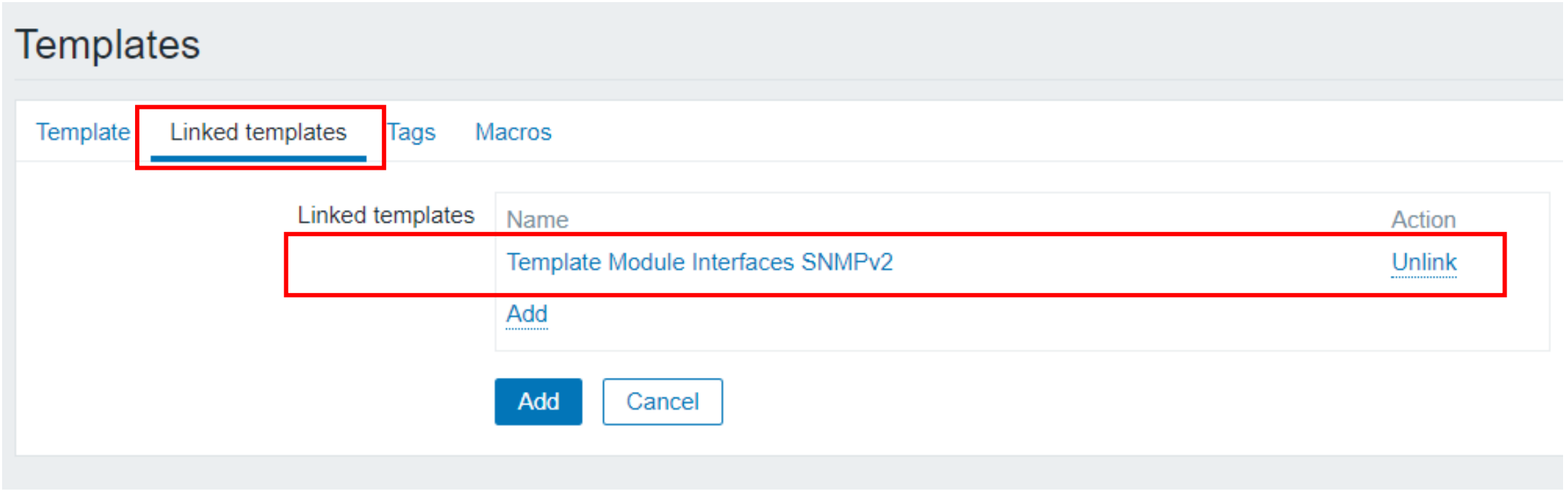


Default filter ANY DESCRIPTION, even empty:

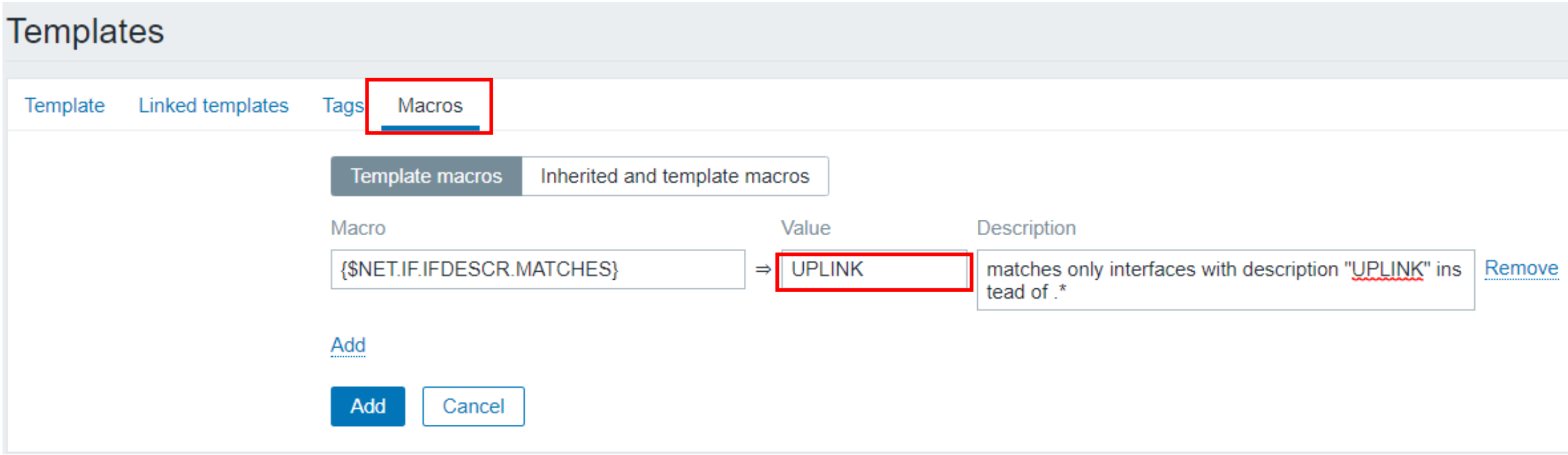
Create new empty template



Create new empty template with link



Create new empty template with link and new value for macro



Triggers

- Do not use {ITEM.LASTVALUE} in trigger names
  - Fill this in **operational data** field (Zabbix 4.4)
- Use priority scale suggested (check guidelines doc)
- Explain the threshold
- Don’t forget about macros context

Triggers. Explain the threshold

Good	**Bad**
Temperature is too high **(over 35 C for 5m)** CPU load is too high **(over 1.5)** MySQL: Refused connections **(max_connections limit reached)**	Temperature is too high (now: 40) CPU load is too high MySQL: Refused connections

Triggers. Use macros context

Good	**Bad**
{$IF.ERRORS.WARN:"{#IFNAME}"} {$TEMP_WARN:"{#SENSORNAME}"}	{$IF.ERRORS.WARN} {$TEMP_WARN}

Data collection

Use preprocessing and built-in agents (like HTTP agent):

- instead of Zabbix sender(trapper)
- instead of UserParameters/External scripts

Why avoid Zabbix sender and scripts in and use preprocessing instead?

- minimize the «observer effect»
- processing is described in the template itself no hidden magic
- platform independent
- better control of how data is collected (compared to sender)

When to resort to Zabbix sender?

- Zabbix sender protocol is implemented inside your code/app
- irregular data (events, i.e. backup completed)
- sync history data with shifted timestamp
- takes a lot of time to complete (> 3 seconds)
  - you also can use zabbix-agent2 plugins for such case now

Throttling (Discard with heartbeat). When to use?

- discrete states that rarely changes
  - i.e. disk health, interface status
- inventory items
- do not use for floats

TEMPLATES COLLABORATION

AN OFFICIAL GUIDE TO MAKING AND MANAGING
GREAT TEMPLATES

Item in 4.2 XML

```
<item>
  <name>ICMP loss</name>
  <type>3</type>
  <snmp_community/>
  <snmp_oid/>
  <key>icmppingloss</key>
  <delay>1m</delay>
  <history>1w</history>
  <trends>365d</trends>
  <status>0</status>
  <value_type>0</value_type>
  <allowed_hosts/>
  <units>%</units>
  <snmpv3_contextname/>
  <snmpv3_securityname/>
  <snmpv3_securitylevel>0</snmpv3_securitylevel>
  <snmpv3_authprotocol>0</snmpv3_authprotocol>
  <snmpv3_authpassphrase/>
  <snmpv3_privprotocol>0</snmpv3_privprotocol>
  <snmpv3_privpassphrase/>
  <params/>
  <ipmi_sensor/>
  <authtype>0</authtype>
  <username/>
  <password/>
  <publickey/>
  <privatekey/>
  <port/>
  <description/>
  <inventory_link>0</inventory_link>
  <applications>
    <application>
      <name>Status</name>
    </application>
  </applications>
  <valuemap/>
  <logtimefmt/>
  <preprocessing/>
  <jmx_endpoint/>
  <timeout>3s</timeout>
  <url/>
  <query_fields/>
  <posts/>
  <status_codes>200</status_codes>
  <follow_redirects>1</follow_redirects>
  <post_type>0</post_type>
  <http_proxy/.....
</item>
.....
```

Trigger in 4.2 XML

```
<trigger>
        <expression>{Template Module ICMP Ping:icmppingloss.min(5m)}&gt;{$ICMP_LOSS_WARN}
and {Template Module ICMP Ping:icmppingloss.min(5m)}&lt;100</expression>
        <recovery_mode>0</recovery_mode>
        <recovery_expression/>
        <name>High ICMP ping loss</name>
        <correlation_mode>0</correlation_mode>
        <correlation_tag/>
        <url/>
        <status>0</status>
        <priority>2</priority>
        <description>Loss: {ITEM.LASTVALUE1}</description>
        <type>0</type>
        <manual_close>0</manual_close>
        <dependencies>
           <dependency>
              <name>Unavailable by ICMP ping</name>
              <expression>{Template Module ICMP Ping:icmpping.max(#3)}=0</expression>
              <recovery_expression/>
           </dependency>
        </dependencies>
        <tags/>
</trigger>
```

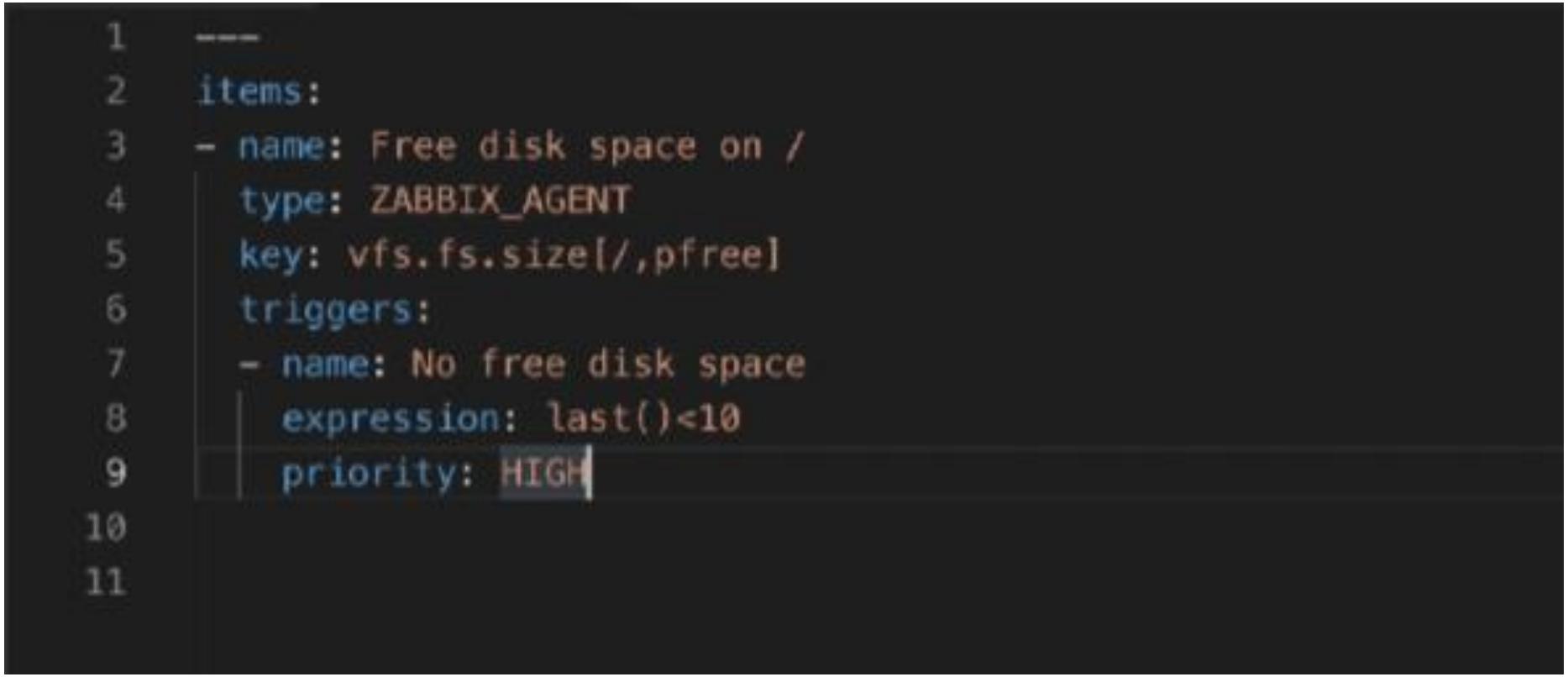
ITEM + SIMPLE TRIGGER in 4.4 XML

```
<item>
        <name>ICMP loss</name>
        <type>SIMPLE</type>
        <key>icmppingloss</key>
        <history>1w</history>
        <value_type>FLOAT</value_type>
        <units>%</units>
        <applications>
          <application>
            <name>Status</name>
          </application>
        </applications>
        <triggers>
          <trigger>
            <expression>{min(5m)}&gt;{$ICMP_LOSS_WARN} and {min(5m)}&lt;100</expression>
            <name>High ICMP ping loss</name>
            <opdata>Loss: {ITEM.LASTVALUE1}</opdata>
            <priority>WARNING</priority>
            <dependencies>
              <dependency>
                <name>Unavailable by ICMP ping</name>
                  <expression>{Template Module ICMP Ping:icmpping.max(#3)}=0</expression>
              </dependency>
            </dependencies>
          </trigger>
        </triggers>
</item>
```

- All defaults or non-relevant item parameters are **removed** from the file export
- All constants are now easy to understand STRINGs
- Simple triggers are defined inside the item node with expressions not mentioning template or host name or even metric

New XML format

- Easier to control changes in templates with Git
- Easier to write templates in text editor or with some script
- Next stop: YAML or JSON by default?



How to help

- Templates officially supported
- Create new issues or feature requests https://support.zabbix.com
- Leave feedback zabbix.com/forum/ - special topic for each new template is available

THANK YOU!

Vitaly Zhuravlev
ZABBIX Zabbix Solution Architect

ZABBIX SUMMIT '19

zabbix