Хитрости и трюки Zabbix

Кузюткина Элина Леонидовна
Инженер тех. поддержки, тренер ZABBIX

ZABBIX
РОССИЯ

1. {$USER_MACROS}

Хитрости и трюки Zabbix

ZABBIX
РОССИЯ

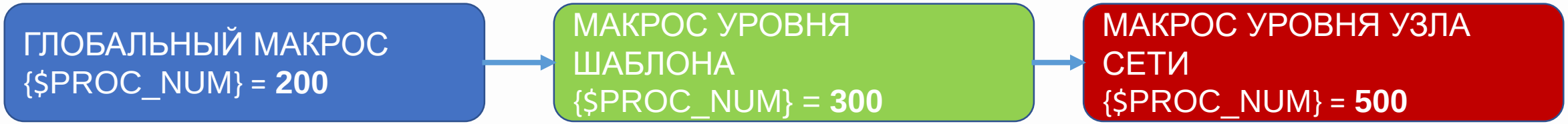
Что такое {$USER_MACROS}?

Макрос — это символьное имя в шаблонах, заменяемое при обработке препроцессором на последовательность символов

Это – **переменные**, которые способны хранить различную информацию

- пороги срабатывания для триггеров
- различные фильтры
- учетные данные
- …..

Макросы определяются на нескольких уровнях, более локальный перекрывает предыдущий



Фиксированные пороговые значения в шаблонах

ZABBIX РОССИЯ

Разные серверы могут иметь разные **пороговые значения** для таких элементов, как загрузка процессора, объем свободной памяти, количество процессов и т.д.

{Template OS: system.cpu.load[percpu,avg1].avg(5m)}>**5**
{Template OS: vm.memory.size[available].last(0)}<**20M**
{Template OS: proc.num[].avg(5m)}>**300**

Динамические пороги в шаблонах

ZABBIX РОССИЯ

Используйте **{$USER_MACROS}** в качестве **порогового значения** для таких элементов данных как загрузка CPU, размер свободной памяти, количества процессов и т.д.

{Template OS: system.cpu.load[percpu,avg1].avg(5m)}>**{$CPU_LOAD}**
{Template OS: vm.memory.size[available].last(0)}<**{$MEMORY_FREE}**
{Template OS: proc.num[].avg(5m)}>**{$PROC_NUM}**

Динамические пороги в шаблонах

МАКРОС УРОВНЯ ШАБЛОНА

МАКРОС УРОВНЯ УЗЛА СЕТИ

{$PROC_NUM} = **300**

Обычный узел сети

{$PROC_NUM} = **300**

Сильно нагруженный узел сети

{$PROC_NUM} = **500**

Менее загруженный узел сети

{$PROC_NUM} = **200**

Фиксированные номера портов в шаблонах

ZABBIX РОССИЯ

Одинаковые сервисы могут использовать различные **номера портов** tcp

{Template OS: net.tcp.service[ssh,**22**]
{Template OS: net.tcp.service[http,**80**]
{Template OS: net.tcp.service[https,**443**]

Динамические номера портов в шаблонах

ZABBIX РОССИЯ

Используйте **{$USER_MACROS}** как **номер порта** для tcp/udp соединений ssh, http, https и других служб

{Template OS: net.tcp.service[ssh,**{$SSH_PORT}**]
{Template OS: net.tcp.service[http,**{$HTTP_PORT}**]
{Template OS: net.tcp.service[https,**{$HTTPS_PORT}**]

Динамические номера портов в шаблонах

МАКРОС УРОВНЯ ШАБЛОНА

МАКРОС УРОВНЯ УЗЛА СЕТИ

{$HTTP_PORT} = **80**

Обычный узел сети

{$HTTP_PORT} = **80**

Нестандартный узел сети 1

{$HTTP_PORT} = **8080**

Нестандартный узел сети 2

{$HTTP_PORT} = **8000**

Фиксированные пороги для элементов данных LLD (Правил Низкоуровневого Обнаружения)

ZABBIX РОССИЯ

Разные точки монтирования имеют различный **размер**, и следовательно

пороговые значения для предупреждения о их заполненности отличаются

• /boot	малый размер	**100M**
• /	средний размер	**10G**
• /data	большой размер	**1TB**

Используйте контекст для макросов в пороговых значениях для элементов данных LLD

ZABBIX РОССИЯ

Разные точки монтирования имеют различный **размер**, поэтому используйте

контекст макросов в выражениях триггеров

- /boot малый размер {$LOW_SPACE:”/boot”}
- / средний размер {$LOW_SPACE:”/”}
- /data большой размер {$LOW_SPACE:”/data”}

{host:vfs.fs.size[{#FSNAME},free].last()} < {$LOW_SPACE:"{#FSNAME}"}

Может также использоваться для **имен дисков Windows**

Как работает контекст пользовательских макросов

МАКРОС УРОВНЯ ШАБЛОНА

{$LOW_SPACE} = **1G**

{$LOW_SPACE: " /boot "} = **10M**

{$LOW_SPACE: " / "} = **500M**

{$LOW_SPACE: " /var "} = **5G**

МАКРОС УРОВНЯ УЗЛА СЕТИ

{$LOW_SPACE} = **1G**

{$LOW_SPACE: " /boot "} = **10M**

{$LOW_SPACE: " / "} = **500M**

{$LOW_SPACE: " /var "} = **10G**

{$LOW_SPACE: " /data "} = **50G**

Разный набор служб на разных серверах

ZABBIX РОССИЯ

Разные серверы имеют разные сервисы, которые необходимо отслеживать

- Сервер 1 DHCP client, Windows defender
- Сервер 2 DHCP client, Windows defender, MS Exchange
- Сервер 3 DHCP client, RDP Service, RPC

Использовать только @Global regular expressions неудобно и сложно, ведь возможных **комбинаций очень много**

Используйте комбинацию {$USER_MACROS} и @global regular expressions

МАКРОС УРОВНЯ ШАБЛОНА

{$SERVICES} =**none**

ГЛОБАЛЬНОЕ РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ

@SERVICES = (**DNS Client|DHCP Client**)

МАКРОС УРОВНЯ УЗЛА СЕТИ

Обычный узел сети

{$SERVICES} = **none** OR @SERVICES

Узел сети со службами exchange

{$SERVICES}= **(exchange)** OR @SERVICES

Узел сети со службами Tomcat и WSUS

{$SERVICES}= **(Tomcat|WSUS)** OR @SERVICES

Пример фильтрации служб

ZABBIX РОССИЯ

МАКРОС УРОВНЯ УЗЛА СЕТИ

Макросы | Инвентарные данные узла сети | Шифрование

Макросы узла сети | Макросы узла сети и унаследованные

Макрос		Значение	
{$SERVICES}	⇒	(Tomcat|WSUS)	Удалить

Добавить

Обновить | Клонировать | Полное клонирование | Удалить | Отмена

ГЛОБАЛЬНОЕ РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ

Регулярные выражения

Выражения | Тест

Имя: Services

Выражения

Тип выражения	Выражение	Разделитель	Регистрозависимое	Действие
Результат ИСТИНА	^(DHCP|DNS) client$			Удалить

Добавить

Обновить | Клонировать | Удалить | Отмена

Правило обнаружения | Фильтры

Тип вычисления: Или — A or C

Фильтры

Подпись	Макрос		Регулярное выражение	Действие
A	{#SERVICE.NAME}	совпадает	@Services	Удалить
C	{#SERVICE.NAME}	совпадает	{$SERVICES}	Удалить

Добавить

Обновить | Клонировать | Удалить | Отмена

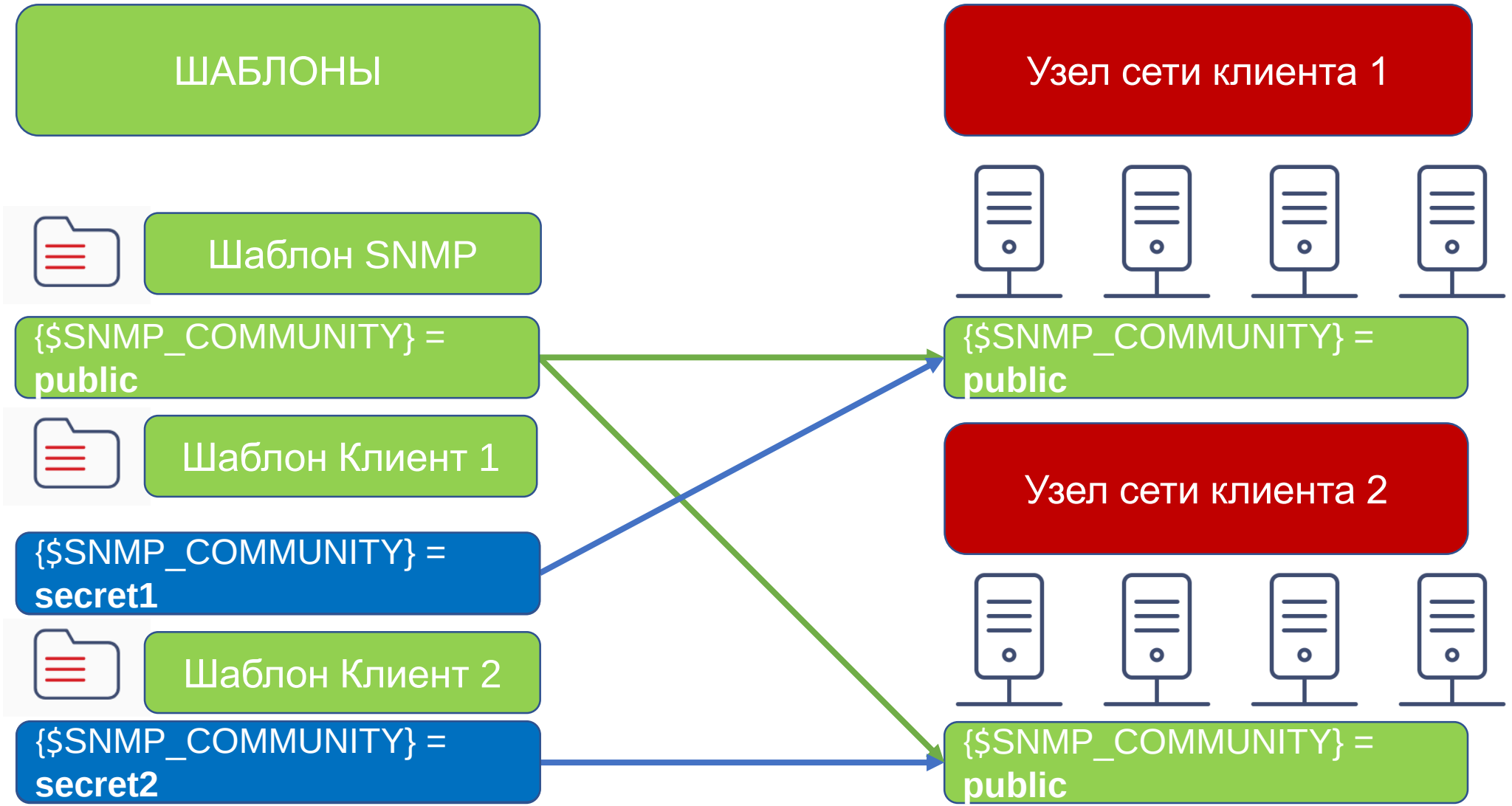
Разные клиенты используют разные учетные данные

ZABBIX РОССИЯ

- Разные клиенты используют **разные учетные данные**
  - для пароля SNMPv2
  - для SNMPv3 логина и пароля
  - для учетных данных SSH
  - для учетных данных веб-сайта

Теоретически, вы можете просто клонировать шаблон для каждого клиента, но потом вам придется управлять **этим множеством** шаблонов

Разные клиенты используют разные учетные данные

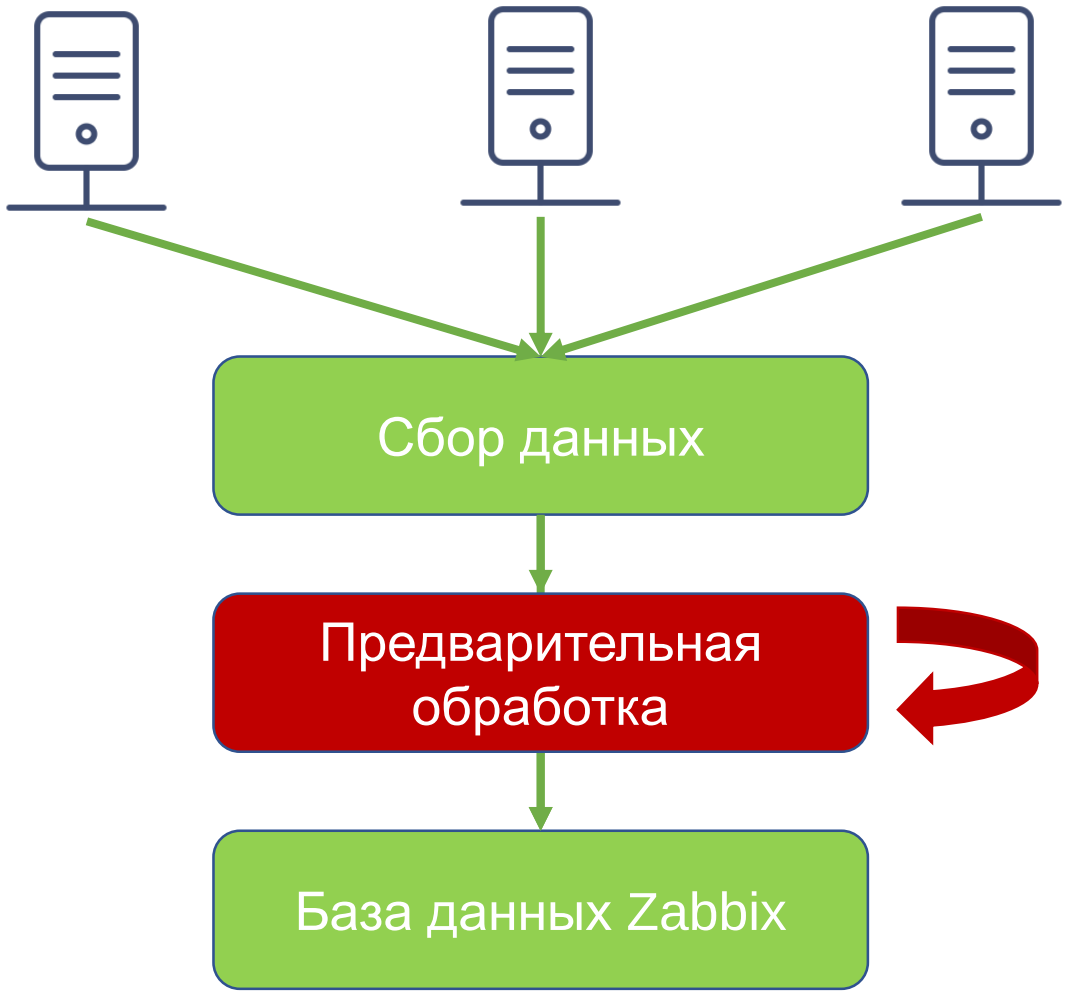


2. Предобработка

Хитрости и трюки Zabbix

ZABBIX
РОССИЯ

Когда выполняется предварительная обработка?



Нужно делить или умножать значения

ZABBIX РОССИЯ

- Конвертация Байтов в Биты
- Конвертация Битов в Байты
- Конвертация мимисекунд в секунды
- ....

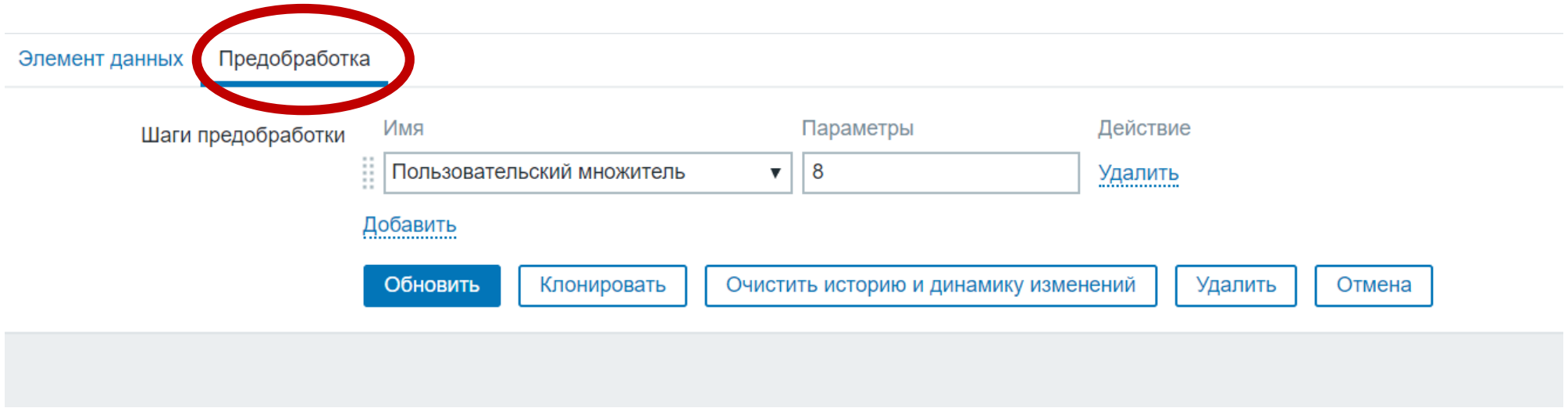
Вы можете задать единицы измерения для удобного отображения информации, но что если вы хотите хранить **преобразованные данные**?

Используйте шаг предварительной обработки – пользовательский множитель

ZABBIX РОССИЯ

Конвертируйте байты в биты используя **множитель 8**

Конвертируйте биты в байты используя **множитель 0.125**



В последних версиях Zabbix для этого используется новая **вкладка «Предобработка»**

Вам нужно извлечь числовые данные из отчёта

ZABBIX РОССИЯ

- Память в Linux
- Отчет о свободном пространстве Linux
- Любой отчет, содержащий числовые данные

```
[root@zabbix40 ~]# free -m
              total        used        free      shared  buff/cache   available
Mem:            991         357         383           7         250         477
Swap:           819           0         819
```

Извлекаем число из текста

```
[root@zabbix40 ~]# free -m
              total        used        free      shared  buff/cache   available
Mem:            991         357         383           7         250         477
Swap:           819           0         819
```

Элемент данных | Предобработка

Шаги предобработки

Имя	Параметры	Действие
Регулярное выражение	^Swap:.*(\b[0-9]+\b).*(\b[0-9	\3	Удалить
Пользовательский множитель	1048576		Удалить

Добавить

Обновить | Клонировать | Очистить историю и динамику изменений | Удалить | Отмена

Извлекайте данные при помощи **шаблонов PCRE REGEX**

Полученные данные могут быть обработаны на следующих шагах

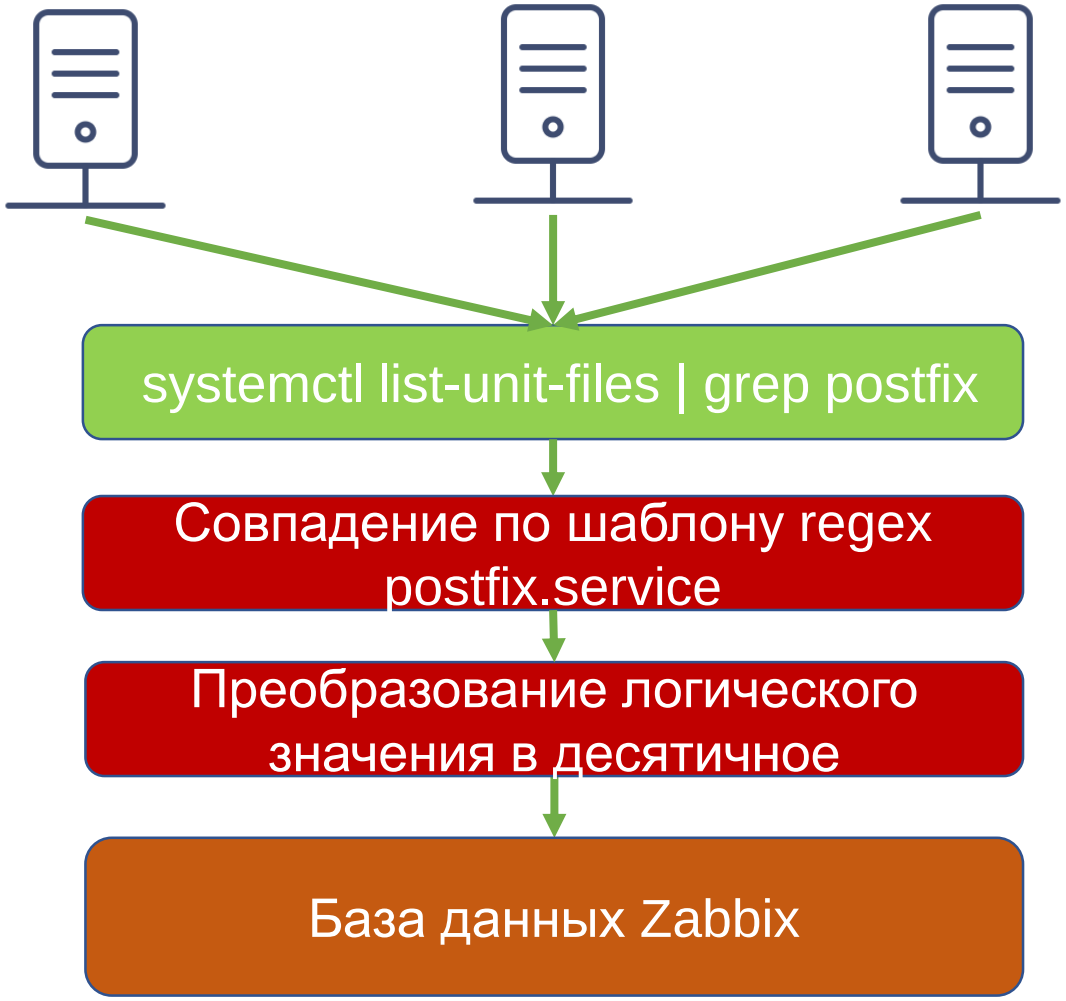
Необходимо преобразовать двоичные текстовые данные в десятичные

ZABBIX РОССИЯ

- systemctl возвращает состояние служб как enabled или disabled
- Мы хотим хранить состояние в виде 1 или 0 для простых триггерных функций или для визуализации в графиках

```
[root@zabbix42 ~]# systemctl list-unit-files | grep postfix
postfix.service                             enabled
```

Используйте регулярные выражения



1	0
true	false
y	n
yes	no
On	off
enabled	disabled
up	down

Преобразование логического значения в десятичное

ZABBIX РОССИЯ

Мастер	Имя ▲	Триггеры	Ключ	Интервал	История	Динамика изменений	Тип	Группы элементов данных	Состояние	Инфо
•••	Postfix service status		ssh.run[postfix_service_status]	30s	90d	365d	SSH агент	Services	Активировано	

Отображено 1 из 1 найденных

Элемент данных | Предобработка

Шаги предобработки

	Имя	Параметры		Custom on fail	Действие
1:	Регулярное выражение	postfix.service\s+(\S+)	\1	☐	Удалить
2:	Логический в десятичный			☐	Удалить

Добавить

Обновить | Клонировать | Проверить сейчас | Очистить историю и динамику изменений | Удалить | Отмена

☐	Postfix service status ssh.run[postfix_service_status]	30s	90d	365d	SSH агент	08.03.2019 21:04:50	1	График

3. Зависимые элементы данных

Хитрости и трюки Zabbix

ZABBIX
РОССИЯ

Требуется извлечь всю числовую информацию из текста

ZABBIX РОССИЯ

- Использование памяти в Linux, отчет о свободном пространстве или любой другой отчет, который содержит числовые данные
- Используя обычные элементы данных нам потребуется **9(!)** проверок чтобы собрать всю информацию
- Это дополнительный сетевой трафик и использование процессора на узле сети.

```
[root@zabbix40 ~]# free -m
              total        used        free      shared  buff/cache   available
Mem:            991         357         383           7         250         477
Swap:           819           0         819
```

Используйте зависимые элементы данных

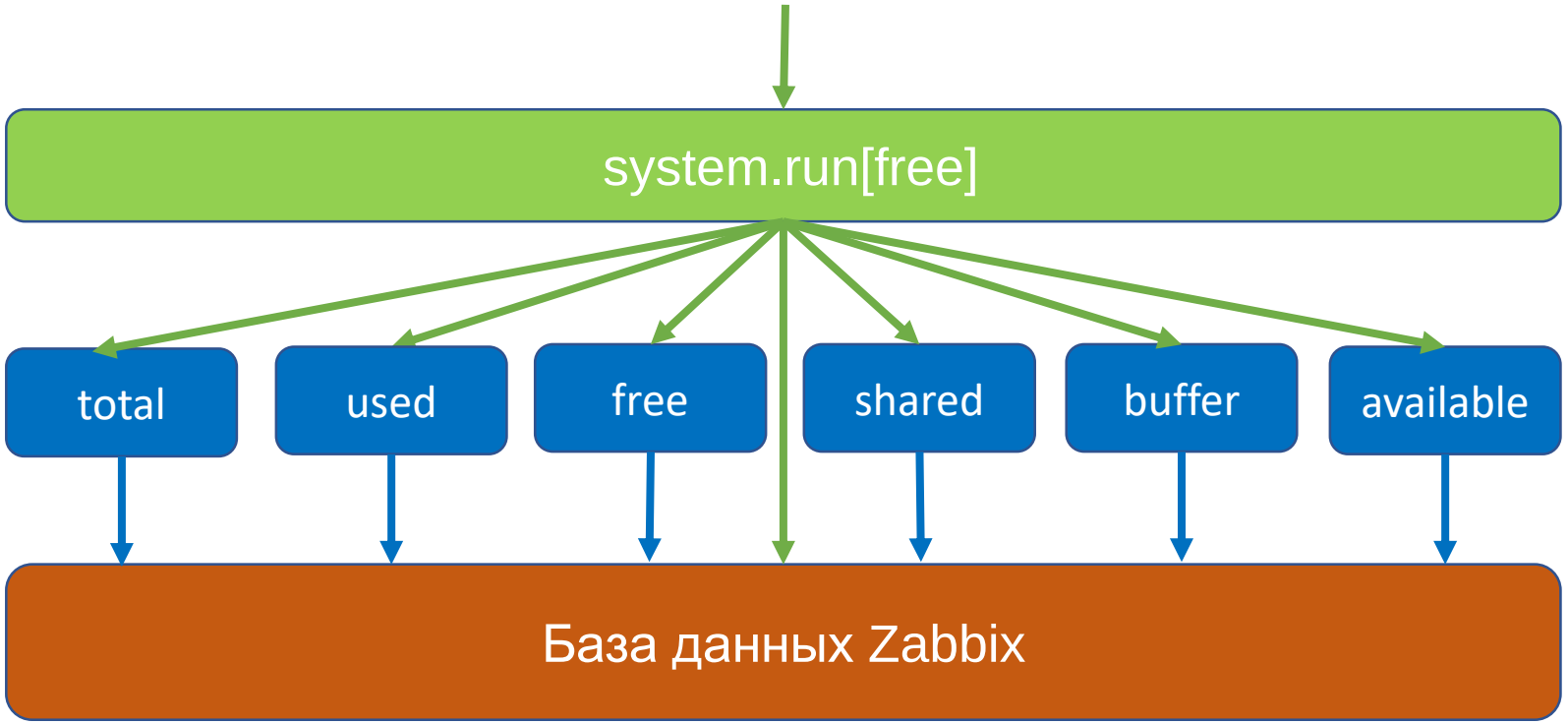
ZABBIX РОССИЯ

Исходные данные

Основной элемент данных собирающий текстовый тип информации

Зависимые элементы данных с числовым типом информации

```
[root@zabbix40 ~]# free -m
              total        used        free      shared  buff/cache   available
Mem:            991         357         383           7         250         477
Swap:           819           0         819
```



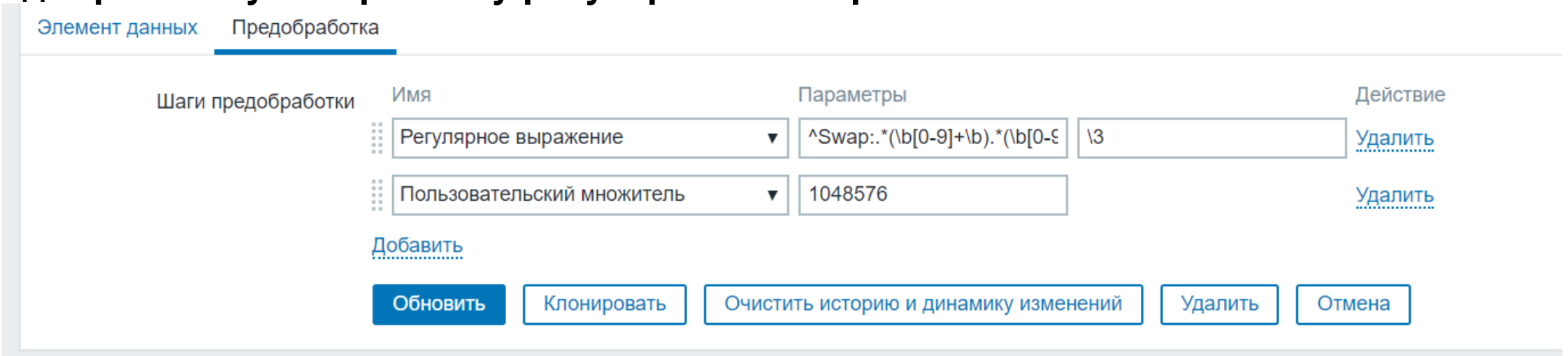
Используйте зависимые элементы данных

ZABBIX РОССИЯ

Соберите все информацию с помощью обычного текстового элемента данных:

```
2018-09-04 09:40:46                total        used        free      shared  buff/cache   available
                Mem:    67359399936 28717457408  6309957632  7909801984 32331984896 29861924864
                Swap:   25769799680           0 25769799680
```

Создайте зависимые элементы данных для каждого значения используя предварительную обработку регулярными выражениями:



Настройка зависимых элементов данных

ZABBIX РОССИЯ

☐	•••	Memory Report	ssh.run[check.memory] 1m	0	
☐	•••	Memory Report: Memory Shared Size	ssh.ram.shared	90d	720d
☐	•••	Memory Report: Memory Total Size	ssh.ram.total	90d	720d
☐	•••	Memory Report: Memory Used Size	ssh.ram.used	90d	720d
☐	•••	Memory Report: Swap Free Size	ssh.swap.free	90d	720d
☐	•••	Memory Report: Swap Total Size	ssh.swap.total	90d	720d
☐	•••	Memory Report: Swap Used Size	ssh.swap.used	90d	720d

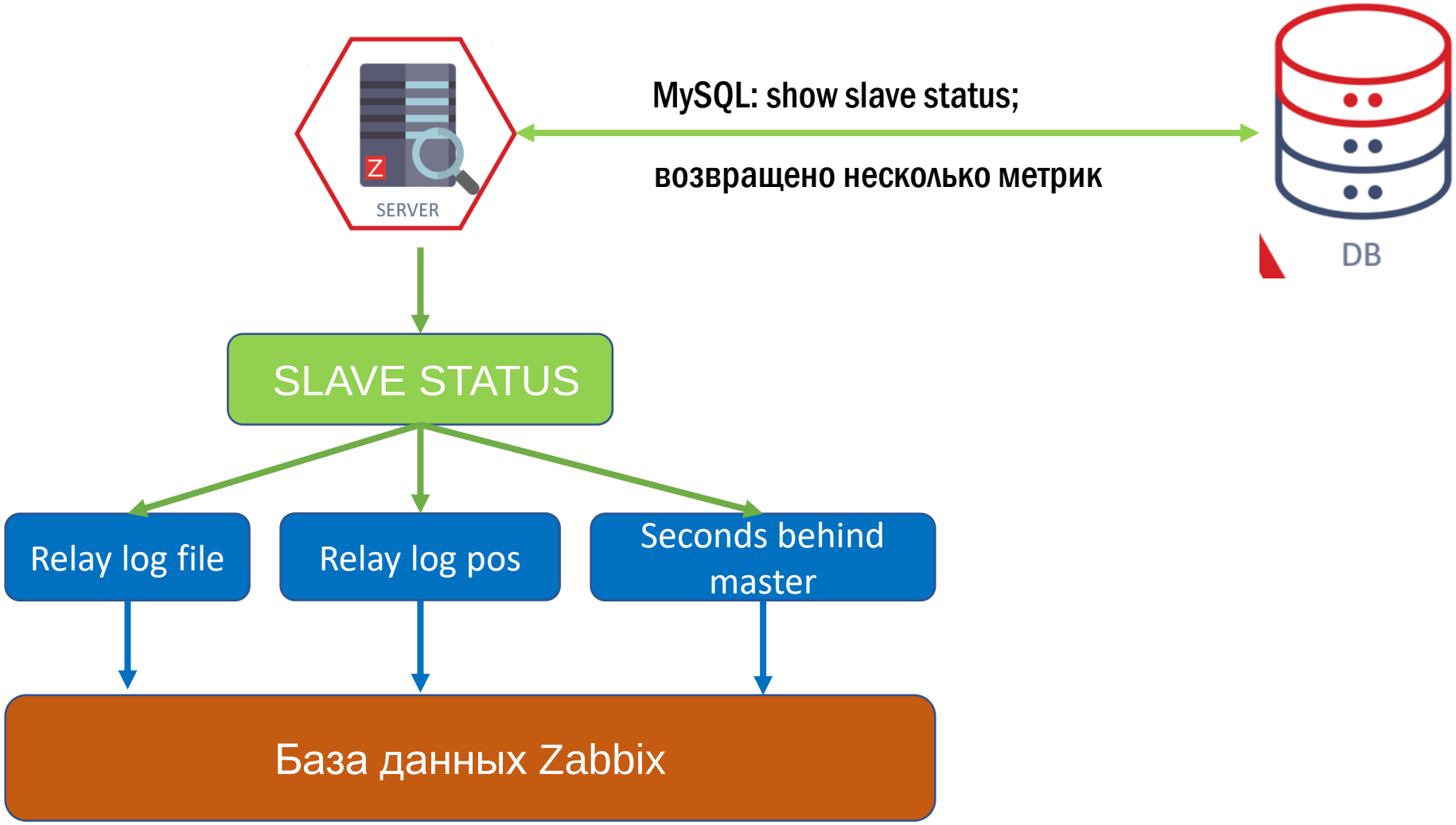
Мониторинг репликации MySQL базы данных

ZABBIX РОССИЯ

```
MariaDB [(none)]> show slave status \G
*************************** 1. row ***************************
Slave_IO_State: Waiting for master to send event
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: master2-bin.000001
Read_Master_Log_Pos: 926945751
Relay_Log_File: master1-relay-bin.000002
Relay_Log_Pos: 207526
Relay_Master_Log_File: master2-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
```

Использование ODBC элементов данных означает сбор данных в **несколько подходов,** а это значит **дополнительный** трафик, использование CPU и несколько подключений к базе данных.

Используйте запросы SQL и зависимые элементы данных



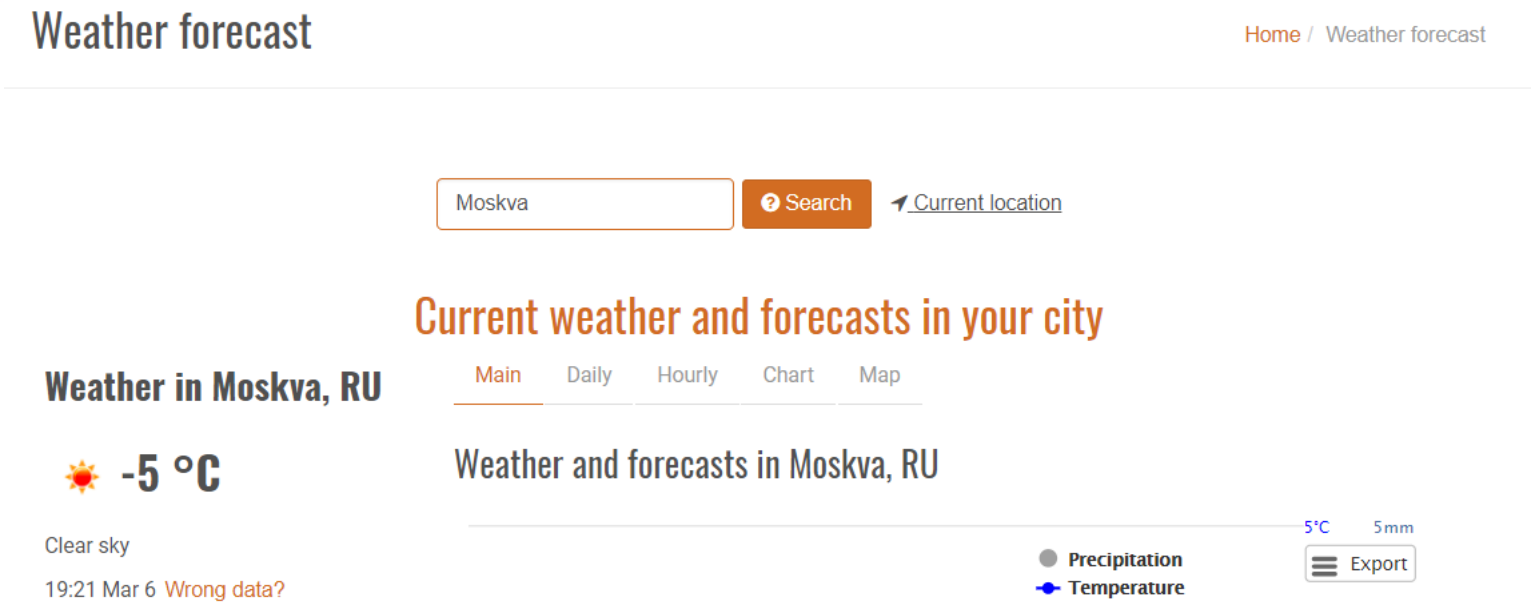
Настройка зависимых элементов данных

ZABBIX РОССИЯ

Мастер	Имя ▲	Триггеры	Ключ	Интервал	История	Динамика
•••	MysqL DB replication status		mysql.slavestatus	30s	0	
•••	MysqL DB replication status: Read_Master_Log_Pos		Read_Master_Log_Pos		90d	365d
•••	MysqL DB replication status: Relay_Log_File		Relay_Log_File		90d	
•••	MysqL DB replication status: Relay_Log_Pos		Relay_Log_Pos		90d	365d
•••	MysqL DB replication status: Seconds_Behind_Master	Triggers 1	Seconds_Behind_Master		90d	365d
•••	MysqL DB replication status: Slave_IO_Running	Triggers 1	Slave_IO_Running		90d	365d
•••	MysqL DB replication status: Slave_IO_State		Slave_IO_State		90d	
•••	MysqL DB replication status: Slave_SQL_Running	Triggers 1	Slave_SQL_Running		90d	365d

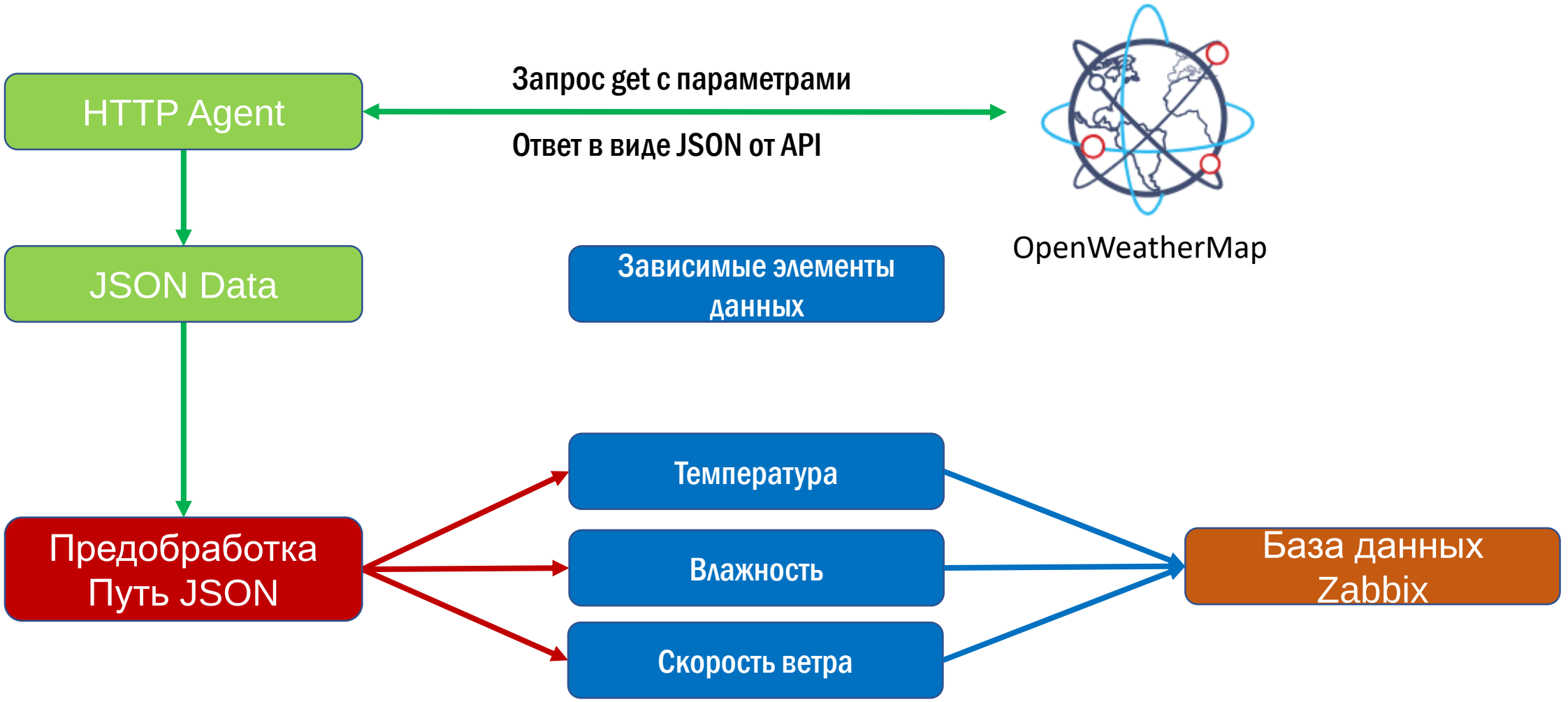
▼	**MySQL Replication** (8 Items)		
	MysqL DB replication status		
	Read_Master_Log_Pos	2018-10-04 11:30:30	926945751
	Relay_Log_File	2018-10-04 11:30:30	master1-relay-bin.000002
	Relay_Log_Pos	2018-10-04 11:30:30	207526
	Seconds_Behind_Master	2018-10-04 11:30:30	0
	Slave_IO_Running	2018-10-04 11:30:30	1
	Slave_IO_State	2018-10-04 11:30:30	Waiting for master to send event
	Slave_SQL_Running	2018-10-04 11:30:30	1

Необходимо собрать погодные данные для вашего местоположения



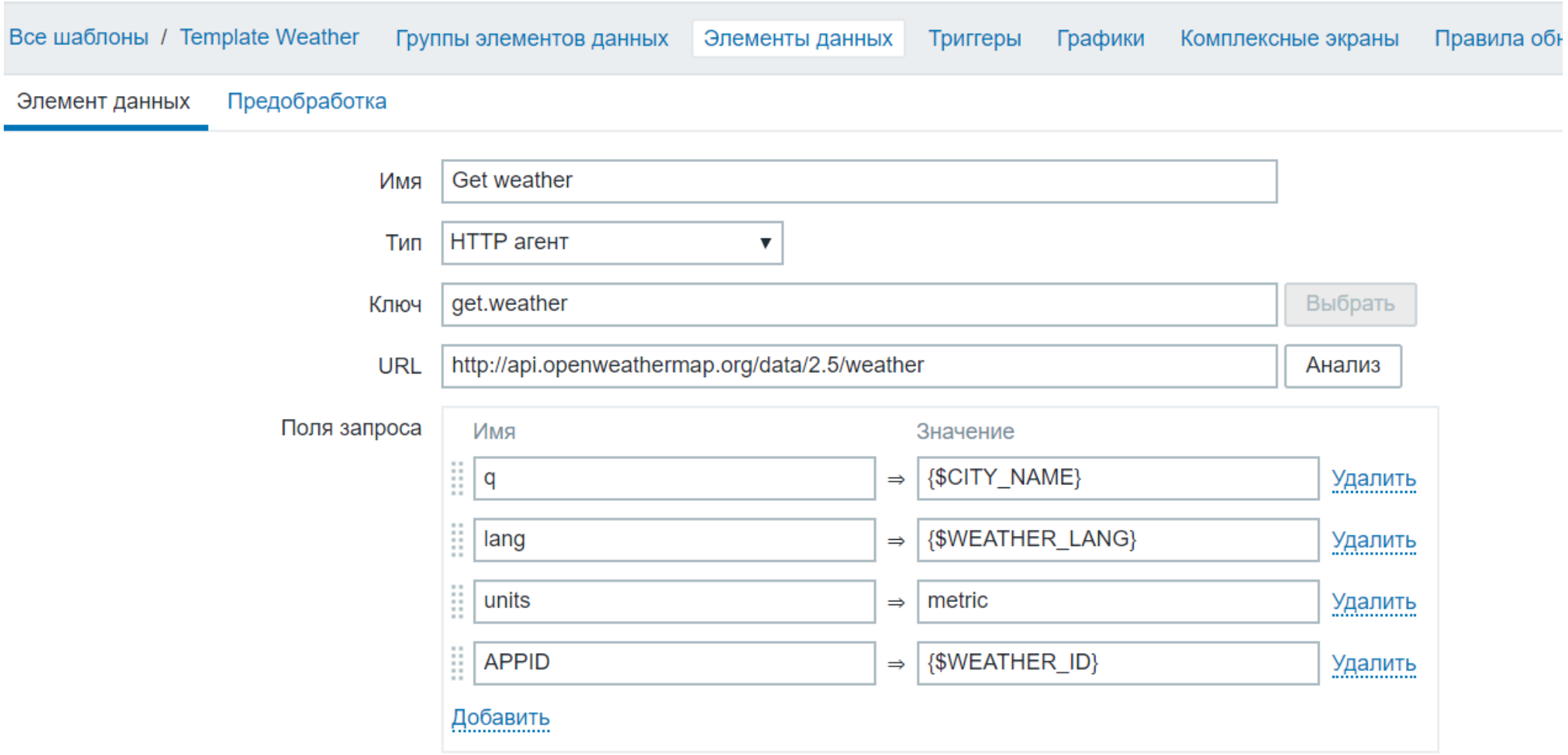
- Хотите получить данные о температуре, влажности, скорости ветра
- Можно использовать собственный curl скрипт, но в итоге его логика может заметно **усложниться**

Используйте OpenWeatherMap API и предобработку JSON



Используйте HTTP agent с {$USER_MACROS}

ZABBIX РОССИЯ



Ответом вы получите JSON объект

06.03.2019 21:01:34 {"coord":{"lon":69.69,"lat":37.84},"weather":[{"id":801,"main":"Clouds","description":"облачно","icon":"02n"}],"base":"stations","main":{"temp":11,"pressure":1015,"humidity":62,"temp_min":11,"temp_max":11},"visibility":10000,"wind":{"speed":1},"clouds":{"all":24},"dt":1551886200,"sys":{"type":1,"id":8989,"message":0.0153,"country":"TJ","sunrise":1551836738,"sunset":1551878388},"id":1221194,"name":"Moskva","cod":200}

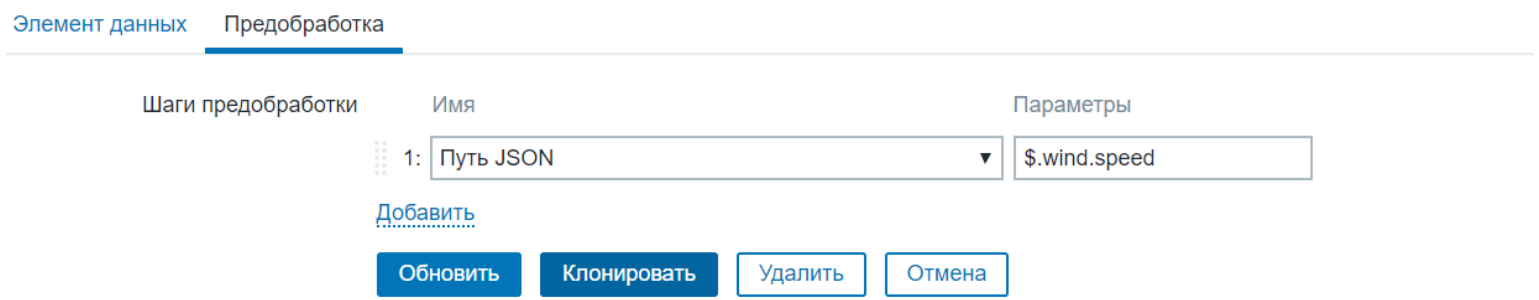

```
{
   "coord":{ },
   "weather":[ ],
   "base":"stations",
   "main":{
      "temp":11,
      "pressure":1015,
      "humidity":62,
      "temp_min":11,
      "temp_max":11
   },
   "visibility":10000,
   "wind":{
      "speed":1
   },
   "clouds":{ },
   "dt":1551886200,
   "sys":{ },
   "id":1221194,
   "name":"Moskva",
   "cod":200
}
```


$.main.temp

$.main.humidity

$.wind.speed

Используйте предобработку для JSON



Предобработка JSON Path

Имя ▲	Интер…	История	Дина…	Тип	Последняя проверка	Последнее значение	Изменение	Инфо
weather (5 элементов данных)								
Get weather get.weather	30s	90d		HTTP агент	07.03.2019 06:45:34	{"coord":{"lon":69.69,"lat":…		История
Get weather HTTP response code get.weather.http		90d		Зависимый …	07.03.2019 06:45:34	200		История
Humidity get.weather.humidity		90d	365d	Зависимый …	07.03.2019 06:45:34	54 %		График
Temperature get.weather.temperature		90d	365d	Зависимый …	07.03.2019 06:45:34	14 C		График
Wind speed get.weather.wind.speed		90d	365d	Зависимый …	07.03.2019 06:45:34	3		График

Зависимые элементы данных

4. Низкоуровневое обнаружение

Хитрости и трюки Zabbix

ZABBIX
РОССИЯ

Необходимо настроить обнаружение пользовательских SNMP метрик

ZABBIX
РОССИЯ

- Многообразие моделей принтеров
- Хотите узнать все параметры принтера
- Расходники
  - Лотки для бумаги
  - Количество отпечатанных страниц
  - Уровень чернил

Используйте обнаружение SNMP правилах LLD

ПРАВИЛО ОБНАРУЖЕНИЯ

Картриджи discovery[{#SNMPVALUE},.1.3.6.1.2.1.43.11.1.1.6]

Объем картриджа .1.3.6.1.2.1.43.8.2.1.8

Уровень картриджа .1.3.6.1.2.1.43.8.2.1.9

ПРОТОТИП ЭЛЕМЕНТА ДАННЫХ

Описание .1.3.6.1.2.1.43.11.1.1.6.{#SNMPINDEX}

Объем картриджа .1.3.6.1.2.1.43.11.1.1.8.{#SNMPINDEX}

Уровень картриджа .1.3.6.1.2.1.43.11.1.1.9.{#SNMPINDEX}

Правило LLD и прототипы элементов данных

ZABBIX РОССИЯ

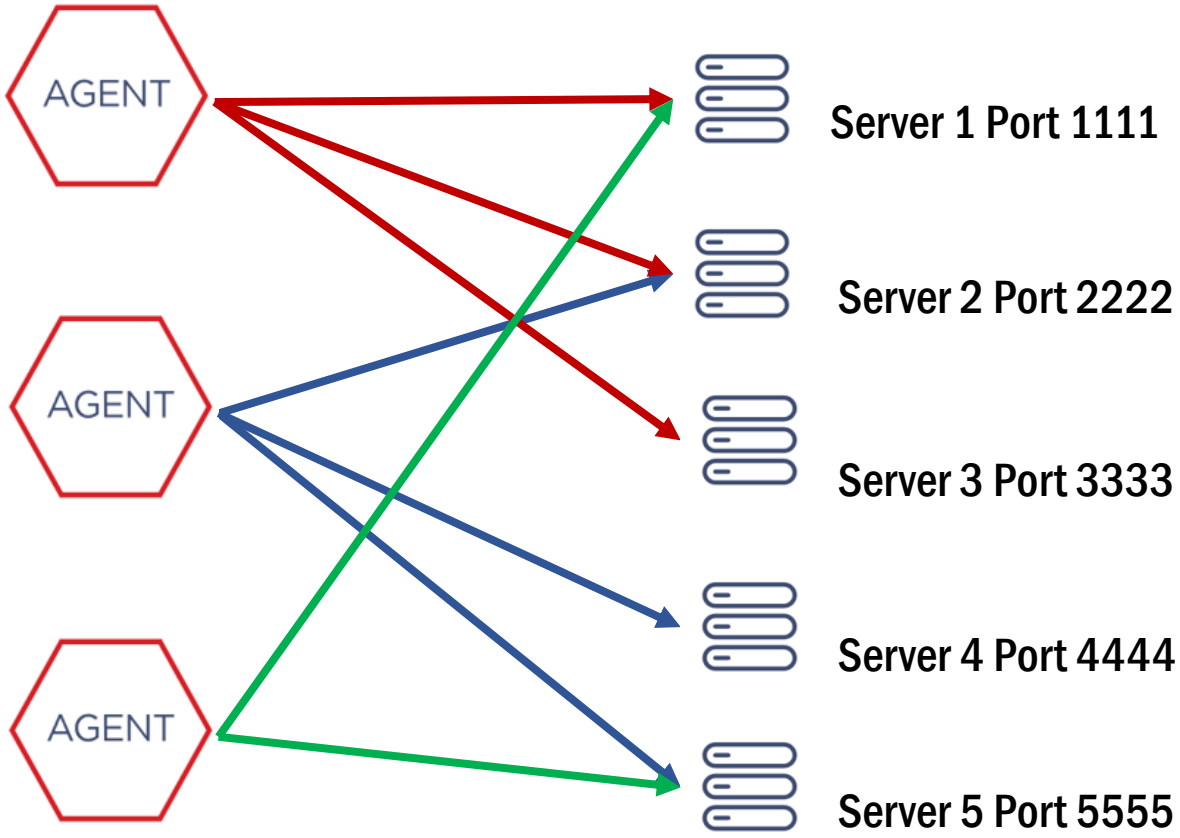
Правило обнаружения | LLD macros | Фильтры

* Имя	Supplies
Тип	SNMPv2 агент ▾
* Ключ	suppliesDescription
* SNMP OID	discovery[{#SNMPVALUE},.1.3.6.1.2.1.43.11.1.1.6]
* SNMP community	{$SNMP_COMMUNITY}

Прототип элемента данных | Предобработка

* Имя	Level of supplies $1
Тип	SNMPv2 агент ▾
* Ключ	suppliesLevel[{#SNMPVALUE}] Выбрать
* SNMP OID	.1.3.6.1.2.1.43.11.1.1.9.{#SNMPINDEX}

Проверка доступности удаленного порта на других узлах сети



- Используется ключ net.tcp.port
- Множество серверов и портов
- Каждый агент имеет свой список серверов для проверки

Используйте собственный скрипт для LLD

ZABBIX РОССИЯ

```
#/bin/bash
IFS=':, ' read -r -a array <<< "$1"
idx=0
echo {\"data\":[
while [ -n "${array[$idx]}" ]; do
        echo -n \{\"{#R_IP}\":\""${array[$idx]}"\",\"{#R_PORT}\":\""${array[$idx+1]}"\"\}
        let idx=$idx+2
        [ -n "${array[idx]}" ] && echo "," || echo
done
echo ]}
exit
```

- Простой **bash скрипт** будет работать из коробки на большинстве платформ
- Использует **{$USER_MACRO}** во входных данных, возвращает **JSON объект**

Создание правила низкоуровневого обнаружения

ZABBIX РОССИЯ

Правило обнаружения | LLD macros | Фильтры

* Имя: Remote TCP connection discovery

Тип: Внешняя проверка

* Ключ: check_remote_port_lld.sh[{$LLD_REMOTE_CHECK}]

* Интервал обновления: 8h

Шаблон | Присоединенные шаблоны | Макросы

Шаблонные макросы | Унаследованные и макросы из шаблонов

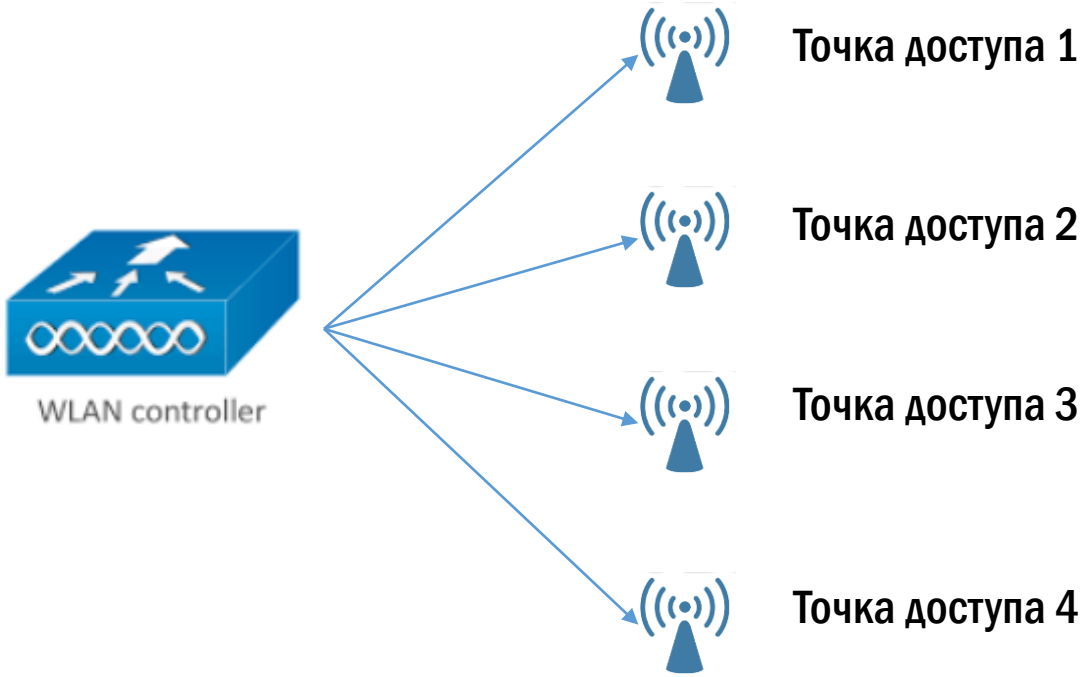
Макрос		Значение	
{$LLD_REMOTE_CHECK}	⇒	10.47.181.89:8080,10.47.181.80:8080,10.47	Удалить

Добавить

Обновить | Клонировать | Полное клонирование | Удалить | Удалить и очистить | Отмена

Remote TCP connection discovery: Remote TCP service to 10.74.181.79:8080]	Triggers 3	net.tcp.service.perf[tcp,10.74.181.79,8080]
Remote TCP connection discovery: Remote TCP service to 10.74.181.80:8080]	Triggers 3	net.tcp.service.perf[tcp,10.74.181.80,8080]
Remote TCP connection discovery: Remote TCP service to 10.74.181.81:8080]	Triggers 3	net.tcp.service.perf[tcp,10.74.181.81,8080]
Remote TCP connection discovery: Remote TCP service to 10.74.181.93:8080]	Triggers 3	net.tcp.service.perf[tcp,10.74.181.93,8080]

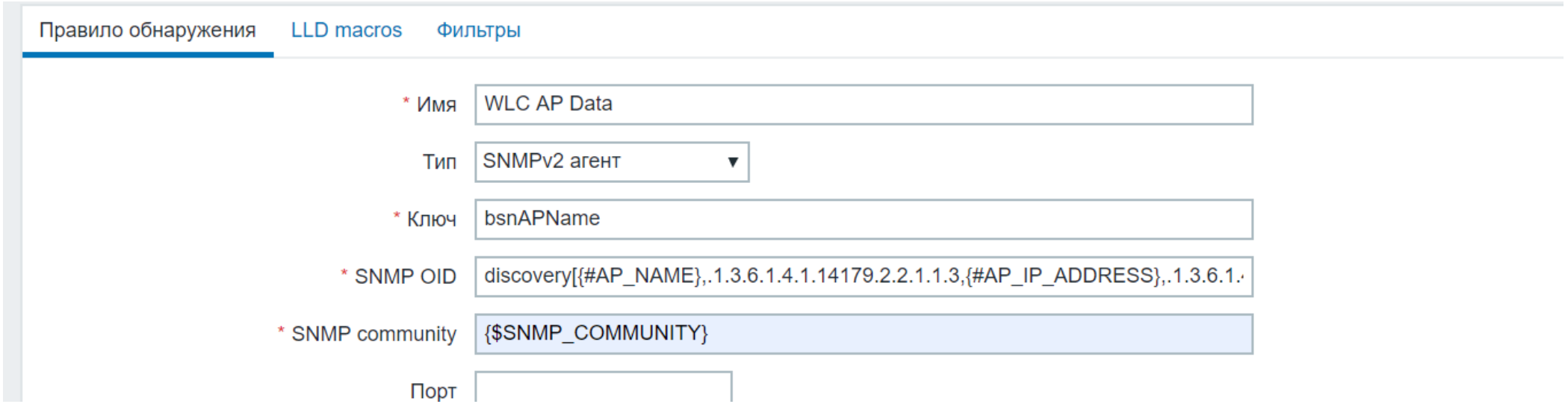
Все данные точек доступа собираются на контроллере

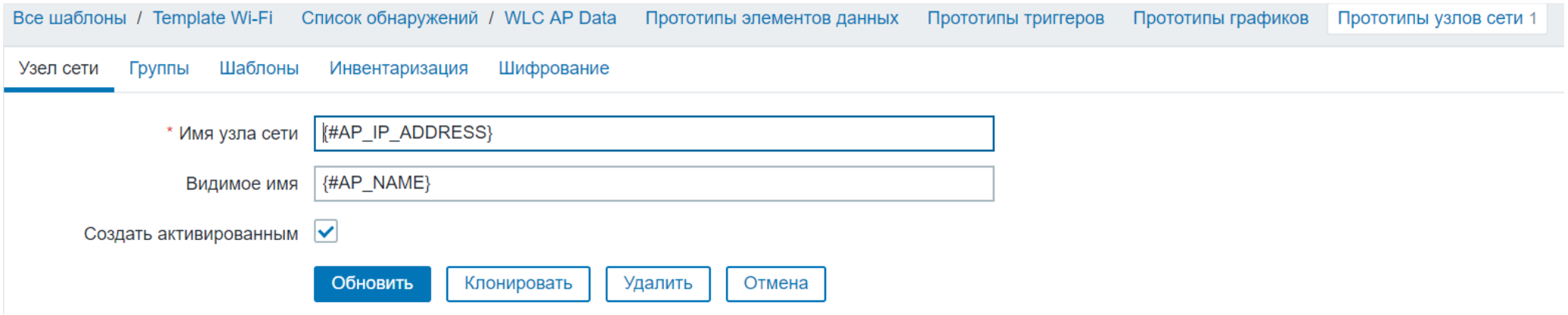


- WLAN контроллер возвращает все SNMP данные по каждой точке доступа
- Точки доступа используются разными клиентами

Создайте правило обнаружения и прототипы узлов сети для сбора метрик по точкам доступа

ZABBIX РОССИЯ





Для каждого узла сети присоедините шаблон

ZABBIX РОССИЯ

Все шаблоны / Template Wi-Fi | Список обнаружений / WLC AP Data | Прототипы элементов данных | Прототипы триггеров | Прототипы графиков | Прототипы узлов сети 1

Узел сети | Группы | Шаблоны | Инвентарные данные узла сети | Шифрование

Присоединенные шаблоны

Имя	Действие
Template Cisco WLC Discovery	Отсоединить

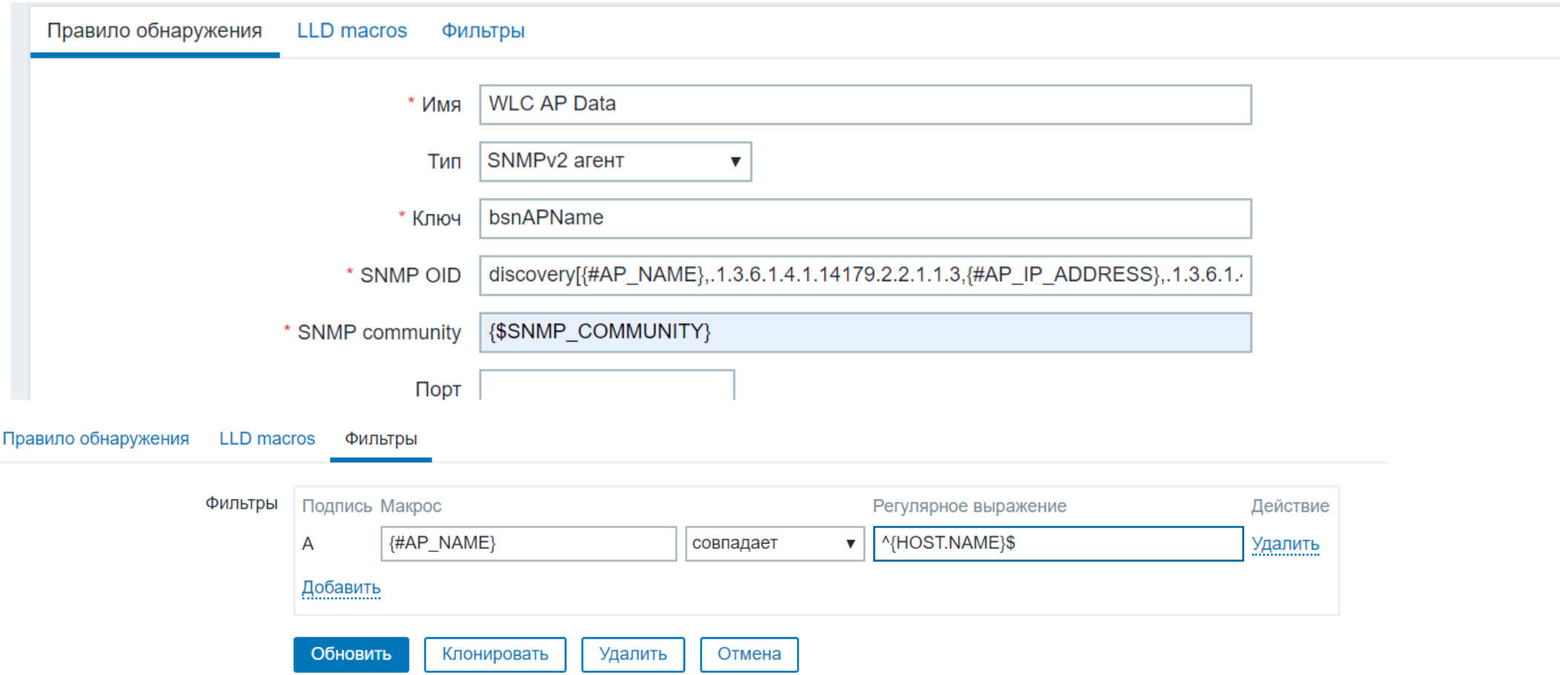
Соединить с новыми шаблонами

начните печатать для поиска | Выбрать

Добавить

Обновить | Клонировать | Удалить | Отмена

Данные все еще собираются из WLC и фильтруются по актуальной точке



Используйте **фильтры LLD** чтобы отфильтровать данные только по конкретной точке доступа

Результат – узлы сети, автоматически созданные для каждой точки доступа

WLC AP data: APc47d.4f3a.8181	Applications 2	Items 8	Triggers 1	Graphs 1	Discovery 1	Web	172.16.0.4: 161
WLC AP data: APc84c.75ee.212d	Applications 2	Items 8	Triggers 1	Graphs 1	Discovery 1	Web	172.16.0.4: 161
WLC AP data: APc84c.757a.11a7	Applications 2	Items 8	Triggers 1	Graphs 1	Discovery 1	Web	172.16.0.4: 161
WLC AP data: APd0d0.fd2e.17ce	Applications 2	Items 8	Triggers 1	Graphs 1	Discovery 1	Web	172.16.0.4: 161
WLC AP data: PMO	Applications 2	Items 8	Triggers 1	Graphs 1	Discovery 1	Web	172.16.0.4: 161

AP Data (6 Items)

APc47d.4f3a.8181 Ap If No Of Users 2.4 GHz	2018-09-28 01:12:12	3 users
APc47d.4f3a.8181 Ap If No Of Users 5 GHz	2018-09-28 01:12:12	0 users
APc47d.4f3a.8181 AP If Phy Channel Number 2.4 GHz	2018-09-28 01:10:12	1
APc47d.4f3a.8181 AP If Phy Channel Number 5 GHz	2018-09-28 01:10:12	36
APc47d.4f3a.8181 AP Ip Address	2018-09-28 00:35:12	172.16.0.49

Преимущества использования прототипов узла сети

ZABBIX РОССИЯ

- Управляются Zabbix **внутренним механизмом LLD**
  - Добавляются по необходимости
  - Автоматически удаляются или изменяются
- Шаблоны и группы назначаются **автоматически**
- Группы узлов сети могут генерироваться **динамически** из макросов LLD
- Можно **ограничить доступ** только к части данных с использованием групп узлов и групп пользователей.

Работа объемных LLD правил в нестабильных средах

ZABBIX РОССИЯ

- Мы хотим обнаруживать и мониторить **лог файлы**
- Набор логов изменяется **каждые сутки**, но **момент** времени изменения нам **не известен**
- **Список логов большой** и обработка полученного JSON накладна по ресурсам – мы не можем запускать правило часто

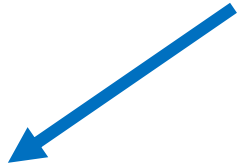
Можно написать скрипт, определяющий изменения конфигурации и если таковые есть он отправит готовый JSON при помощи zabbix_sender. Запускать скрипт по расписанию.

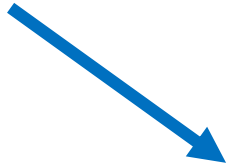
Но что, если у нас нет возможности использовать локальный планировщик?

Троттлинг

ZABBIX РОССИЯ

Мы будем обрабатывать значение только в случае его изменения





Основные элементы данных

Это уменьшит объем входных данных, если они в основном статичны

Элемент данных правила LLD

Это позволит уменьшить интервал опроса для **LLD** правила и позволит быстрее реагировать на изменения в конфигурации не нагружая при этом сервер

Троттлинг в LLD

ZABBIX РОССИЯ

Правило обнаружения | Предобработка | LLD macros | Фильтры

* Имя: Log discovery rule

Тип: Zabbix агент

* Ключ: discovery.log

* Интерфейс узла сети: 127.0.0.1 : 10050

* Интервал обновления: 1h

Правило обнаружения | Предобработка | LLD macros | Фильтры

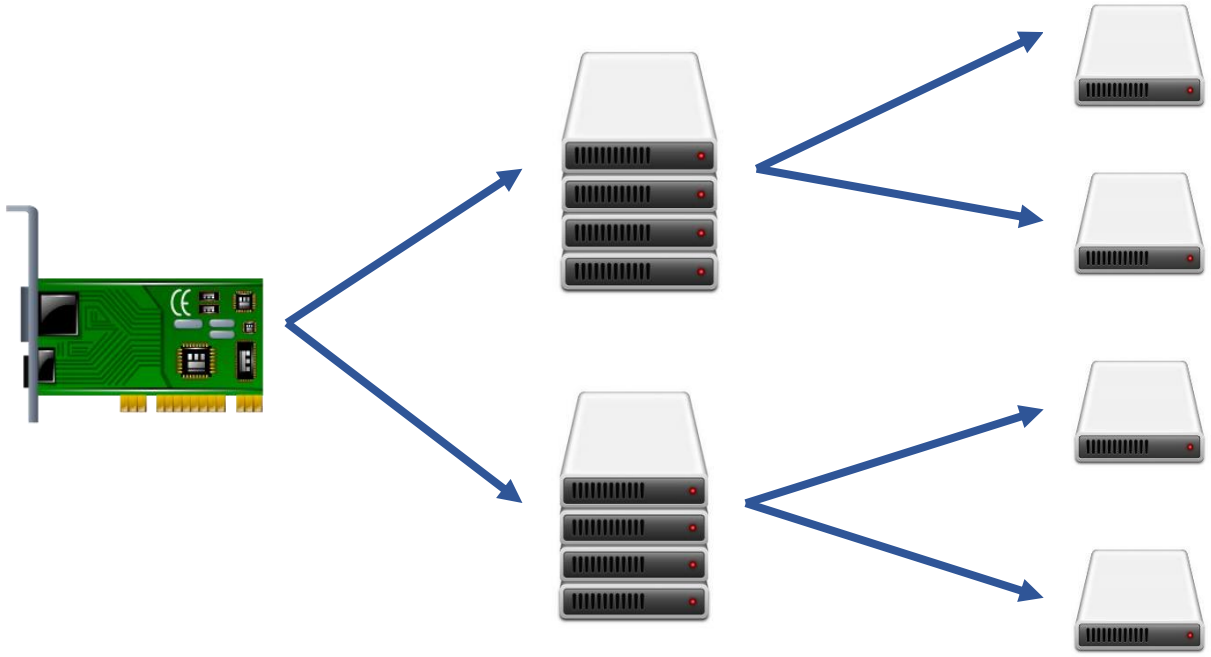
Шаги предобработки

	Имя	Параметры	Custom on fail	Действие
1:	Discard unchanged with heartbeat	24h		Удалить

Добавить

Добавить | Отмена

JSON в LLD правилах



- RAID контролер возвращает данные обо всех массивах и дисках
- На разных контролерах используется разное количество дисков
- Мы хотим отслеживать состояние и производительность всех дисков

Пример возвращаемых контроллером данных


```
{
  "Controllers":{
    "Command Satus":{ },
    "Response Data":[
      {
        "Drive":"/c0/e62/s0",
        "EID:Slt":"62:0",
        "DID":"9",
        "State":"Onln",
        "DG":"0",
        "Size":"930.391 GB",
        "Intf":"SATA",
        "Med":"HDD",
        "SED":"N",
        "PI":"N",
        "Sesz":"512B",
        "Model":"ST1000NX0423  00AJ145LEN",
        "Sp":"U",
        "Drive /c0/e62/s0 - Detailed Information":{
          "Drive /c0/e62/s0 State":{ }
        }
      },
      {
        "Drive":"/c0/e62/s1",
        "EID:Slt":"62:1",
        "DID":9,
        "State":"Onln",
        "DG":0,
        "Size":"930.391 GB",
        "Intf":"SATA",
        "Med":"HDD",
```


JSON для правила LLD

{#DRIVE} -> $.Drive
{#SIZE} -> $.Size

Предобработка в LLD

ZABBIX РОССИЯ

Прототипы элементов данных 1 | Прототипы триггеров | Прототипы графиков | Прототипы узлов сети

Правило обнаружения | Предобработка | LLD macros | Фильтры

Шаги предобработки

	Имя	Параметры	Custom on fail	Действие
1:	JSONPath	$['Controllers']['Response Data']	☐	Удалить

Добавить

Обновить | Клонировать | Проверить сейчас | Удалить | Отмена

Прототипы элементов данных 4 | Прототипы триггеров | Прототипы графиков | Прототипы узлов сети

Правило обнаружения | Предобработка | LLD macros | Фильтры

LLD macros

LLD macro	JSONPath	
{#DRIVE}	$.Drive	Удалить
{#MODEL}	$.Model	Удалить
{#SIZE}	$.Size	Удалить
{#STATE}	$.State	Удалить

Добавить

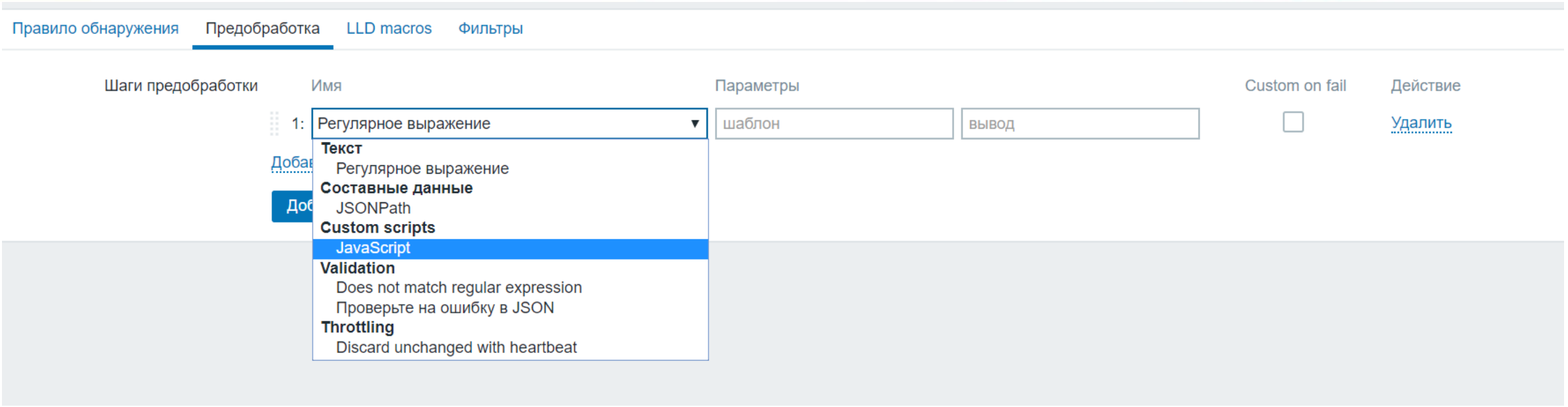
Обновить | Клонировать | Проверить сейчас | Удалить | Отмена

«Последние данные»

☐	Disk /c0/e62/s0 Temperature raid.disk.check.temp[/c0/e62/s0]	90d	365d	Зависимый …	09.03.2019 02:26:04	32 C		График
☐	Disk /c0/e62/s1 Temperature raid.disk.check.temp[/c0/e62/s1]	90d	365d	Зависимый …	09.03.2019 02:25:58	30 C	-2 C	График
☐	Error count on /c0/e62/s0 raid.disk.check.error[/c0/e62/s0]	90d	365d	Зависимый …	09.03.2019 02:26:04	0		График
☐	Error count on /c0/e62/s1 raid.disk.check.error[/c0/e62/s1]	90d	365d	Зависимый …	09.03.2019 02:25:58	0		График
☐	S.M.A.R.T alert flagged by drive /c0/e62/s0 raid.disk.check.smart[/c0/e62/s0]	90d		Зависимый …	09.03.2019 02:26:04	No		История
☐	S.M.A.R.T alert flagged by drive /c0/e62/s1 raid.disk.check.smart[/c0/e62/s1]	90d		Зависимый …	09.03.2019 02:25:58	No		История

Мы получили работающее правило без скриптов

Предобработка в правилах LLD



- Используйте Путь Json или регулярное выражение чтобы без пользовательских скриптов привести правило к JSON для LLD. LLD макросы завершат эту задачу
- Если нужна логика сложнее, например условные проверки или циклы – вы можете написать JavaScript для предобработки правила
- Проверьте на ошибки – совпадает ли с шаблоном регулярного выражения и нет ли ошибок в JSON объекте
- Троттлинг

5. Глобальная корреляция событий

Хитрости и трюки Zabbix

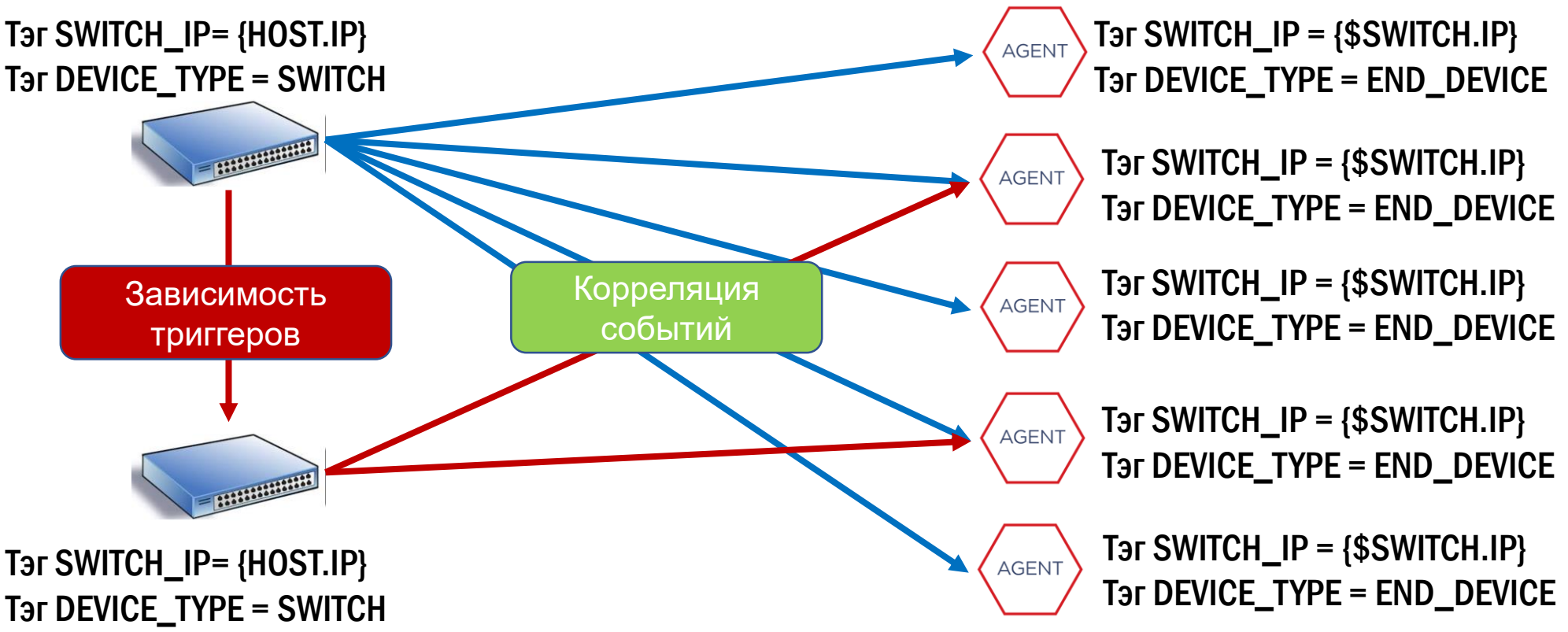
ZABBIX
РОССИЯ

Подавление оповещений

ZABBIX
РОССИЯ

- Конечные устройства подключены через **разные коммутаторы**
- Можно использовать Зависимость триггеров, но это может оказаться что устройств за каждым коммутатором **множество**
- Если устройство будет **перемещено за другой коммутатор**, придется изменить и связи триггеров

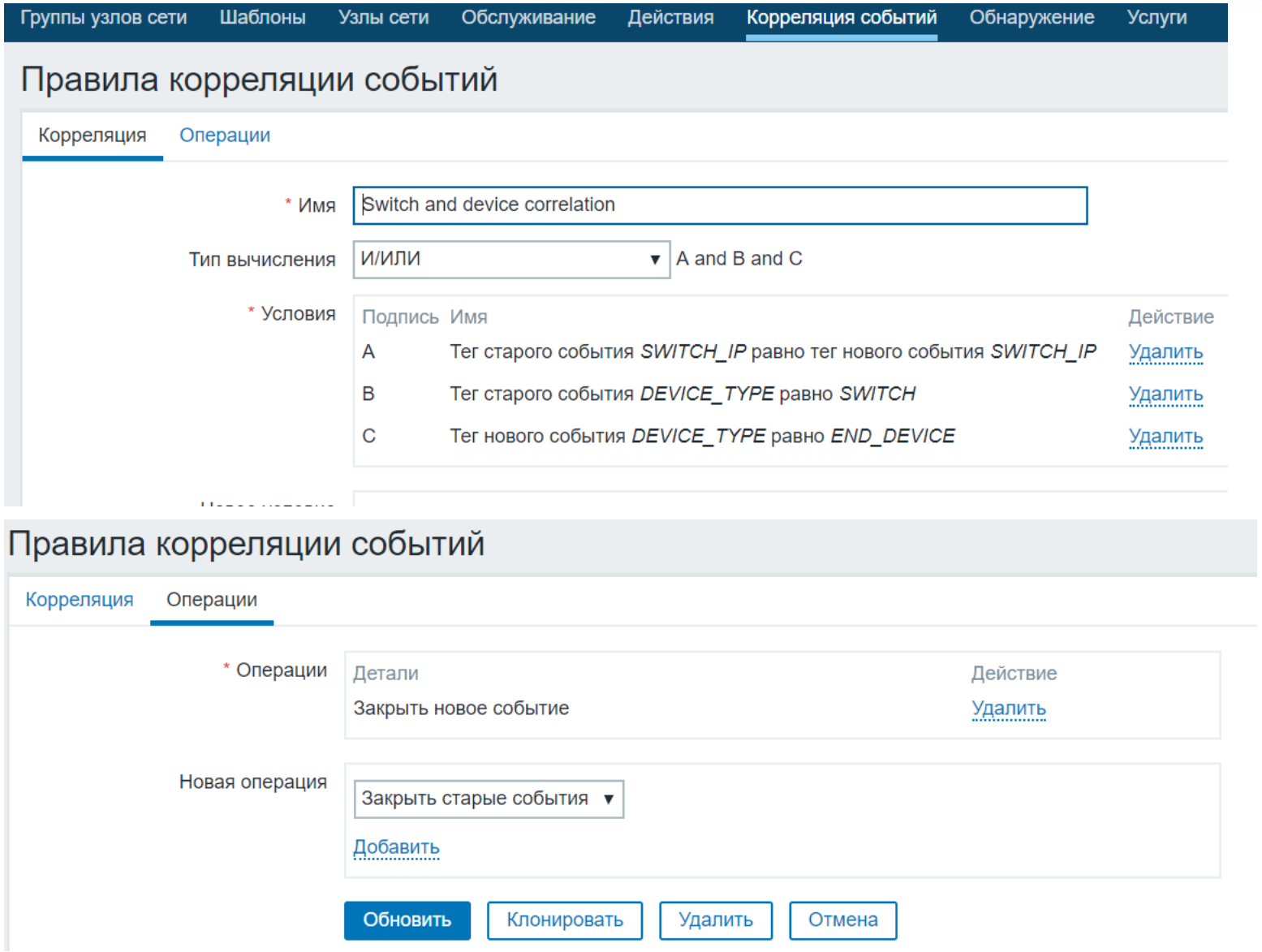
Используйте глобальную корреляцию событий на основе тегов



Макрос {$SWITCH.IP} генерируется API скриптом на основе ARP таблиц, которые собираются Zabbix с коммутатора

Создайте правило глобальной корреляции событий

ZABBIX РОССИЯ



Спасибо!

Кузюткина Элина Леонидовна
Инженер тех. поддержки, тренер ZABBIX

ZABBIX
РОССИЯ