



The Enterprise class Monitoring Solution for Everyone

[WWW.ZABBIX.COM](http://WWW.ZABBIX.COM)

# Zabbix 3.0 Тренинг Сертифицированный Профессионал

День 2

# ПОВЕСТКА

Шифрование



Расширенные  
функции  
триггеров



Zabbix API



Производи-  
тельность



Высокая  
доступность  
и  
избыточность



Сертификация





ПОДДЕРЖКА  
ШИФРОВАНИЯ

# ШИФРОВАНИЕ В ZABBIX

## Зачем ?

### Шифрование

- защита деликатных данных (например, данные конфигурации от Zabbix сервера на прокси могут содержать учетные данные для доступа к наблюдаемым узлам сети)

### Аутентификация

- доверие "другой стороне"
- предотвращение отправки сфальсифицированных данных на Zabbix

# ШИФРОВАНИЕ В ZABBIX

Как ?

## Протокол безопасности транспортного уровня *TLS* 1.2

- в настоящее время не известных уязвимостей "из-за дизайна"

## Выбор библиотек

- **OpenSSL** с 1.0.1 --with-openssl
- **GnuTLS** с 3.1.18 --with-gnutls
- **mbed TLS (PolarSSL)** с 1.3.9 – 1.3.x, не 2.0.0.  
--with-mbedtls

# ШИФРОВАНИЕ В ZABBIX

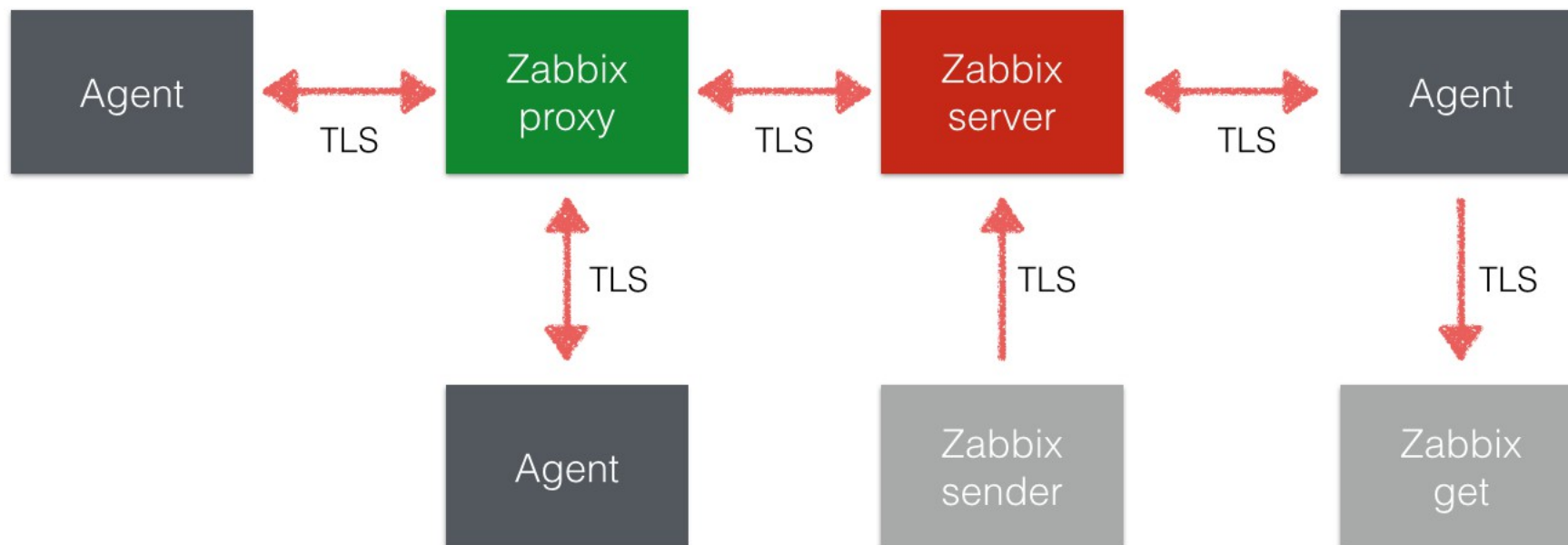
Можно использовать набор криптографических инструментов OpenSSL, GnuTLS или mbed TLS (PolarSSL)

Разные компоненты могут использовать разные наборы инструментов

Соединение можно настроить на использование сертификата или PSK

Использование тех же портов, что и при незашифрованных соединениях

Имеется две части: К узлу сети (для пассивных проверок) и ОТ узла сети (активные проверки и zabbix\_sender)



Все соединения между Zabbix сервером, прокси и агентами можно выборочно настроить на использование шифрования или оставить незашифрованными

Шифрование поддерживается утилитами командной строки

# НОВЫЕ ОПЦИИ В ZABBIX

**TLSConnect** – **исходящие** соединения (активные проверки)

**TLSAccept** – **входящие** соединения (пассивные проверки)

**TLSCAFile** – CA сертификаты

**TLSCRLFile** – отозванные сертификаты

**TLSServerCertIssuer** – издатель сертификата сервера

**TLSServerCertSubject** – субъект сертификата сервера

**TLSCertFile** – сертификат агента

**TLSKeyFile** – приватный ключ агента

**TLSPSKIdentity** – строка идентификатор pre-shared ключа

**TLSPSKFile** – файл с pre-shared ключем



# ШИФРОВАНИЕ ZABBIX

## Сертификаты

- Издатель
- Субъект

## PSK (pre-shared ключи)

- PSK идентификатор
- PSK значение

# ШИФРОВАНИЕ В ZABBIX

## Сертификаты

```
Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=C development team, CN=ZBXNEXT-1263 Signing CA
Validity
    Not Before: Dec 19 12:17:06 2014 GMT
    Not After : Dec 18 12:17:06 2016 GMT
Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=C development team, CN=Zabbix server
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
```

```
...
-----BEGIN CERTIFICATE-----
MIIECDCCAvcgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
```

## PSK (pre-shared ключи)

- **PSK identity = Berlin proxy**
- **PSK value = 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963**

# ТЕКУЩИЕ ОГРАНИЧЕНИЯ

- Повторное использование сеанса TLS сессии при помощи кэширования или тикетов сессий не поддерживается.
- Каждое соединение начинается с TLS handshake, которое занимает время и ресурсы CPU.
- Поддержка шифрования для соединений с Zabbix Java gateway.
- Поддерживаются только X.509 сертификаты с RSA ключами. Поддержка сертификатов с ECDSA ключами даёт лучшую производительность.
- Отзыв сертификата проверяется на соответствие CRL файлам, online проверка от издателя в настоящее время не поддерживается.
- Приватные ключи хранятся в виде простого текста в файлах, которые доступны для чтения компонентами Zabbix в процессе запуска.
- Pre-shared ключи вводятся в Zabbix веб-интерфейсе и хранятся в базе данных Zabbix в виде простого текста.
- Шифрование не защищает соединения:
  - между Zabbix веб-интерфейсом и веб-браузером пользователя
  - между Zabbix сервером (прокси) и базой данных Zabbix.

```

13120:20150907:130022.367 In send_buffer() host:'127.0.0.1' port:10053 entries:5/100
13120:20150907:130022.368 In zbx_tls_connect(): psk_identity:"test agent"
13120:20150907:130022.368 GnuTLS debug: level=3 "ASSERT: gnutls_constate.c:586"
13120:20150907:130022.368 GnuTLS debug: level=4 "HSK[0x24b0170]: Keeping ciphersuite: ECDHE_PSK_AES_128_CBC_SHA256 (C0.37)"
13120:20150907:130022.368 GnuTLS debug: level=4 "HSK[0x24b0170]: Keeping ciphersuite: ECDHE_PSK_AES_128_CBC_SHA1 (C0.35)"
13120:20150907:130022.368 GnuTLS debug: level=4 "HSK[0x24b0170]: Keeping ciphersuite: PSK_AES_128_GCM_SHA256 (00.A8)"
13120:20150907:130022.368 GnuTLS debug: level=4 "HSK[0x24b0170]: Keeping ciphersuite: PSK_AES_128_CBC_SHA256 (00.AE)"
13120:20150907:130022.368 GnuTLS debug: level=4 "HSK[0x24b0170]: Keeping ciphersuite: PSK_AES_128_CBC_SHA1 (00.8C)"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: Sending extension SAFE RENEGOTIATION (1 bytes)"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: Sending extension SUPPORTED ECC (8 bytes)"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: Sending extension SUPPORTED ECC POINT FORMATS (2 bytes)"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (4.1) RSA-SHA256"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (4.2) DSA-SHA256"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (4.3) ECDSA-SHA256"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (5.1) RSA-SHA384"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (5.3) ECDSA-SHA384"
13120:20150907:130022.368 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (6.1) RSA-SHA512"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (6.3) ECDSA-SHA512"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (3.1) RSA-SHA224"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (3.2) DSA-SHA224"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (3.3) ECDSA-SHA224"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (2.1) RSA-SHA1"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (2.2) DSA-SHA1"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: sent signature algo (2.3) ECDSA-SHA1"
13120:20150907:130022.369 GnuTLS debug: level=4 "EXT[0x24b0170]: Sending extension SIGNATURE ALGORITHMS (28 bytes)"
13120:20150907:130022.369 GnuTLS debug: level=4 "HSK[0x24b0170]: CLIENT HELLO was queued [110 bytes]"
13120:20150907:130022.369 GnuTLS debug: level=3 "ASSERT: gnutls_buffers.c:1104"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: SERVER HELLO (2) was received. Length 83[83], frag offset 0, frag length: 83, sequence: 0"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: Server's version: 3.3"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: SessionID length: 32"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: SessionID: 3ae4c90f148ae142793ee392c02a533656d4b0070f58e4c8a8630e8f8231bd3e"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: Selected cipher suite: ECDHE_PSK_AES_128_CBC_SHA256"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: Selected compression method: NULL (0)"
13120:20150907:130022.370 GnuTLS debug: level=4 "EXT[0x24b0170]: Parsing extension 'SAFE RENEGOTIATION/65281' (1 bytes)"
13120:20150907:130022.370 GnuTLS debug: level=4 "EXT[0x24b0170]: Parsing extension 'SUPPORTED ECC POINT FORMATS/11' (2 bytes)"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: Safe renegotiation succeeded"
13120:20150907:130022.370 GnuTLS debug: level=3 "ASSERT: gnutls_buffers.c:1104"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: SERVER KEY EXCHANGE (12) was received. Length 71[71], frag offset 0, frag length: 71, sequence: 0"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: Selected ECC curve SECP256R1 (2)"
13120:20150907:130022.370 GnuTLS debug: level=3 "ASSERT: gnutls_buffers.c:1104"
13120:20150907:130022.370 GnuTLS debug: level=4 "HSK[0x24b0170]: SERVER HELLO DONE (14) was received. Length 0[0], frag offset 0, frag length: 1, sequence: 0"
13120:20150907:130022.371 GnuTLS debug: level=4 "HSK[0x24b0170]: CLIENT KEY EXCHANGE was queued [82 bytes]"
13120:20150907:130022.372 GnuTLS debug: level=4 "REC[0x24b0170]: Sent ChangeCipherSpec"
13120:20150907:130022.372 GnuTLS debug: level=4 "HSK[0x24b0170]: Cipher Suite: ECDHE_PSK_AES_128_CBC_SHA256"
13120:20150907:130022.372 GnuTLS debug: level=4 "HSK[0x24b0170]: Initializing internal [write] cipher sessions"
13120:20150907:130022.372 GnuTLS debug: level=4 "HSK[0x24b0170]: recording tls-unique CB (send)"
13120:20150907:130022.372 GnuTLS debug: level=4 "HSK[0x24b0170]: FINISHED was queued [16 bytes]"
13120:20150907:130022.372 GnuTLS debug: level=3 "ASSERT: session_ticket.c:639"
13120:20150907:130022.374 GnuTLS debug: level=4 "HSK[0x24b0170]: Cipher Suite: ECDHE_PSK_AES_128_CBC_SHA256"
13120:20150907:130022.374 GnuTLS debug: level=3 "ASSERT: gnutls_buffers.c:1104"
13120:20150907:130022.375 GnuTLS debug: level=4 "HSK[0x24b0170]: FINISHED (20) was received. Length 12[12], frag offset 0, frag length: 12, sequence: 0"
13120:20150907:130022.375 End of zbx_tls_connect():SUCCEED (established TLS1.2 ECDHE-PSK-AES-128-CBC-SHA256-32-16)
13120:20150907:130022.375 JSON before sending [{"request":{"agent data":{"data":[{"host":"Zabbix server","key":"system.swap.size[free]","value":"850554880","clock":1441620017,"ns":"98.921489","clock":1441620017,"ns":"362343880"},{"host":"Zabbix server","key":"system.swap.size[total]","value":"859828224","clock":1441620017,"ns":"70082560","clock":1441620017,"ns":"362896658"},{"host":"Zabbix server","key":"vm.memory.size[total]","value":"513433600","clock":1441620017,"ns":"363176981"}]},"response":{"success":"processed: 5; failed: 0; total: 5; seconds spent: 0.000271"}}]
13120:20150907:130022.376 JSON back [{"response":{"success":"processed: 5; failed: 0; total: 5; seconds spent: 0.000271"}}]
13120:20150907:130022.376 In check_response() response:{"response":"success","info":"processed: 5; failed: 0; total: 5; seconds spent: 0.000271"}
13120:20150907:130022.376 info from server: 'processed: 5; failed: 0; total: 5; seconds spent: 0.000271'

```

# ПРАКТИЧЕСКАЯ УСТАНОВКА

Настройте PSK шифрование между "*Zabbix сервером*  
← *Zabbix прокси (активный)*"

Настройте PSK шифрование между "*Zabbix сервером*  
←→ *Zabbix агентом*"

Используйте элементы данных "*Zabbix агент*",  
"*Zabbix агент (активный)*" и "*Последние данные*",  
чтобы протестировать ваши настройки

Получите значение с элемента данных с  
использованием утилиты Zabbix Get (подключитесь с  
использованием PSK)



РАСШИРЕННЫЕ  
ФУНКЦИИ  
ТРИГГЕРОВ

# ПРОГНОЗИРОВАНИЕ ТЕНДЕНЦИЙ (ЗНАЧЕНИЕ)

Функция:

`forecast(сек|#кол-во,<сдвиг_времени>,время,<аппр.>,<режим>)`

Параметры:

*сек* – период времени

*#кол-во* – количество значений

*<сдвиг\_времени>* - период вычисления

*время* – горизонт предсказания в секундах

*<аппроксимация>* - используемая функция (линейная, полиномN, экспоненциальная, логарифмическая, степенная)

*<режим>* - требуемый вывод (значение, макс, мин, дельта, среднее)

Пример:

`{ora01_bi:vfs.fs.size[/,free].forecast(7d,,7d)}<100M`

# ПРОГНОЗИРОВАНИЕ ТЕНДЕНЦИЙ (ВРЕМЯ)

Функция:

`timeleft(сек| #кол-во,<сдвиг_времени>,порог,<аппр.>)`

Параметры:

*сек* – период времени

*#кол-во* – количество значений

*<сдвиг\_времени>* - период вычисления

*порог* – значение, которое требуется превысить

*<аппроксимация>* - используемая функция (линейная, полиномN, экспоненциальная, логарифмическая, степенная)

Пример:

`{ora01_bi:vfs.fs.size[/,free].timeleft(1d,,104857600)}<1h`



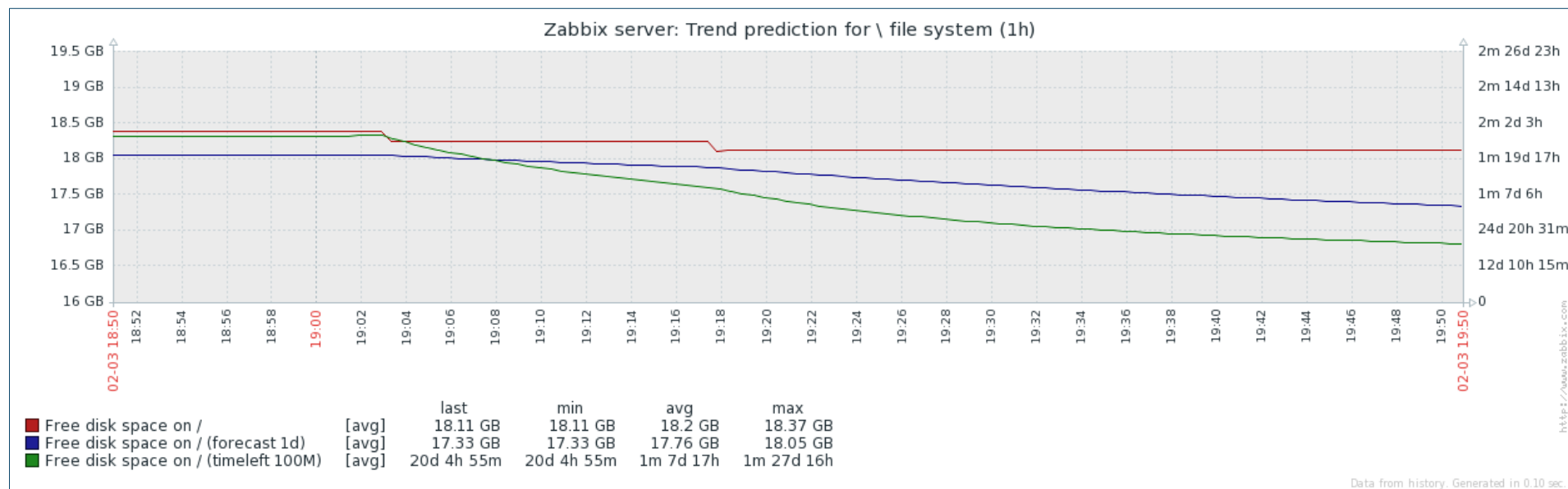
# ПРОГНОЗИРОВАНИЕ ТЕНДЕНЦИЙ

Используйте вычисляемые элементы данных для визуализации значений

Примеры:

```
forecast("vfs.fs.size[/,free]",1d,,1d)
```

```
timeleft("vfs.fs.size[/,free]",1d,,104857600)
```



# ПРАКТИЧЕСКАЯ УСТАНОВКА

Создайте следующие вычисляемые элементы данных в *"Template Basic"*:

```
forecast(system.cpu.load,30m,,30m)  
last(<forecast.item.key>,,30m)
```

Создайте пользовательский график с тремя элементами данных (два упомянуты выше + оригинальный элемент данных загрузки CPU)

Добавьте триггер "Будущее значение загрузки CPU через 30 минут будет больше 1"

Проверьте последние данные

# ЗАМЕТКИ ПО ПРОГНОИЗИРОВАНИЮ ТЕНДЕНЦИЙ

Данные из таблиц "trends\*" не используются

Прогнозирование отображает сейчас какое значение элемента данных ожидается через некоторое время

Некоторые метрики, к сожалению, не прогнозируемы (например, CPU)

Forecast() и timeleft() с линейной аппроксимацией по умолчанию и полиномом 2–3 степени с точки зрения производительности

# ПОДСКАЗКИ ПО ПРОГНОЗИРОВАНИЮ ТЕНДЕНЦИЙ

Если у вас нет идей о том, какое поведение вашей системы мониторинга, начните с линейной (аппроксимация по умолчанию)

Если ваши данные меняются не по прямой, а по кривой, вам возможно необходимо попробовать полином

Степенная аппроксимация может быть полезной, если ваши данные содержат "взлеты" и "падения"

Экспоненциальную аппроксимацию можно использовать для определения пиков

Используйте длительные интервалы с большим количеством точек данных для получения более точных долгосрочных прогнозов

Прогнозы на основе более длительных интервалов могут быть очень медленными в реакции на быстрое изменение в тенденциях

# ПРЕДСКАЗАНИЕ ТЕНДЕНЦИЙ

Дополнительная литература:

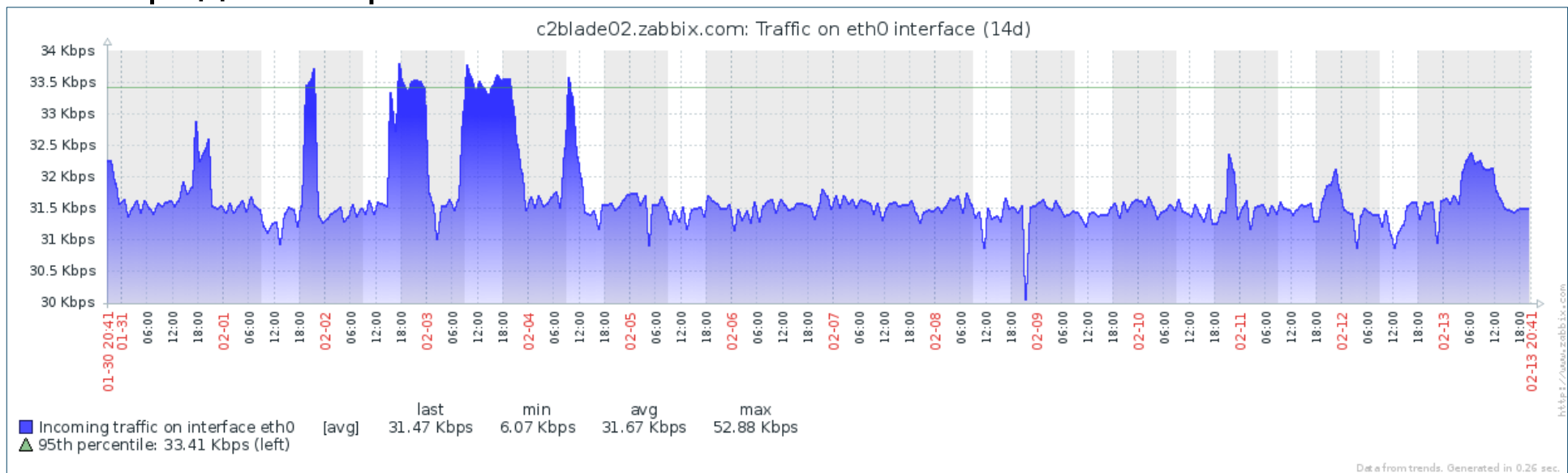
<https://www.zabbix.com/documentation/3.0/manual/config/triggers/prediction>

[http://zabbix.org/mw/images/1/18/Prediction\\_docs.pdf](http://zabbix.org/mw/images/1/18/Prediction_docs.pdf)

# ПРОЦЕНТИЛЬ

Эта функция используется для определения процента доступности. 95-ый процентиль является значением, которое больше 95% наблюдаемых значений:

- ✓ определять уровень пропускной способности без случайных пиков
- ✓ не принимать во внимание пиковый трафик
- ✓ определять различные аномалии



# ПРОЦЕНТИЛЬ

Функция:

`percentile(период/#кол-во, сдвиг_времени, процент)`

Параметры:

*период* – период времени

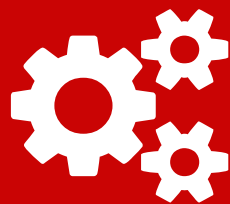
*#кол-во* – количество значений

*сдвиг\_времени* – период сдвига времени

*процент* - диапазон от 0 до 100

Пример:

`{crtr05_rlx:net.if.in[eth0,bytes].percentile(10m,,95)}>10M`



ZABBIX API



# ОБЗОР API

Интерфейс основан на веб-сервере

Безопасность: SSL, аутентификации по паролю, аудит

Основан на JSON-RPC v2.0 спецификации

С учётом прав доступа

# СТРУКТУРА API

## **232** различных методов

Каждый из методов выполняет одну конкретную задачу

Примеры:

**host.create** - создает новый узел сети

**history.get** - получает данные истории

**item.update** – обновляет существующие элементы  
данных

# ФУНКЦИОНАЛ API

## Интеграция Zabbix со сторонними приложениями

- Интеграция с Puppet/CFEngine/Chef/bcfg2 и другими системами управления конфигурацией
- Интеграция с системами задач
- Интеграция с системами инвентаризации для заполнения инвентарных данных

## Создание сервисов поверх Zabbix

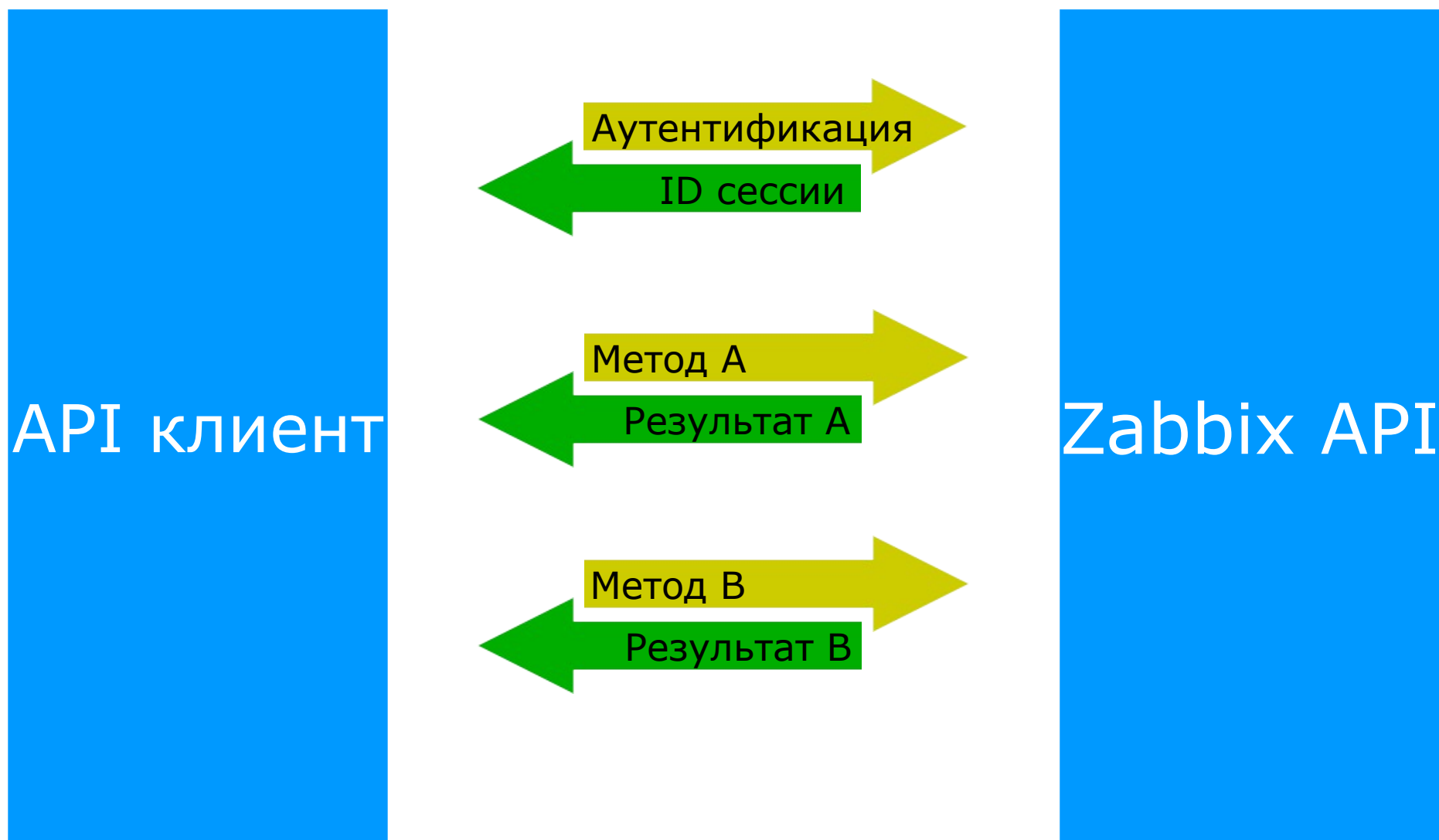
Масштабные изменения конфигурации или развертывания

Более мощные операции

Скрипты командной строки

... и многое другое

# ПОТОК СООБЩЕНИЙ API



# ПРИМЕР - АУТЕНТИФИКАЦИЯ

## Аутентификация через curl:

```
$ curl -i -X POST -H 'Content-  
    Type:application/json' -d'  
{ "jsonrpc": "2.0",  
  "method": "user.login",  
  "params": {  
    "user": "Admin",  
    "password": "zabbix"},  
  "auth": null, "id": 0 }  
' http://195.13.231.163/zabbix/api_jsonrpc.php
```

## Ответ:

```
HTTP/1.1 200 OK  
Date: Wed, 11 Nov 2015 09:32:41 GMT  
Server: Apache/2.2.15 (CentOS)  
X-Powered-By: PHP/5.3.3  
Access-Control-Allow-Origin: *  
Access-Control-Allow-Headers: Content-Type  
Access-Control-Allow-Methods: POST  
Access-Control-Max-Age: 1000  
Content-Length: 68  
Connection: close  
Content-Type: application/json
```

```
{ "jsonrpc": "2.0",  
  "result": "2f2ec4720863281c34cdd3c4c8a5de46", "id": 0 }
```

# ПРИМЕР – ПОЛУЧЕНИЕ УЗЛА СЕТИ

## Получение узла сети через curl:

```
$ curl -i -X POST -H 'Content-Type: application/json' -d '{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "filter": {
      "host": "Zabbix server"
    }
  },
  "auth": "2f2ec4720863281c34cdd3c4c8a5de46",
  "id": 1
}' http://195.13.231.163/zabbix/api_jsonrpc.php
```

## Ответ:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10126",
      "proxy_hostid": "0",
      "host": "Zabbix server",
      "status": "0",
      "disable_until": "0",
      ...
      "name": "Zabbix server",
      "flags": "0",
      "templateid": "0",
      "description": ""
    }
  ],
  "id": 1
}
```

# ПРИМЕР – ПОЛУЧЕНИЕ УЗЛА СЕТИ ПРИ ПОМОЩИ ДРУГИХ ЯЗЫКОВ

Установите библиотеку PyZabbix Python  
при помощи pip:

```
# yum install python-pip
# pip install pyzabbix
```

Ауторизуйтесь & получите узел сети  
через скрипт:

```
#!/usr/bin/env python
from pyzabbix import ZabbixAPI

zapi = ZabbixAPI("http://195.13.231.163/zabbix")
zapi.login("Admin", "zabbix")

result = zapi.host.get(filter={"host" : "Zabbix server"})
for h in result:
    for key in sorted(h):
        print "%s: %s " % (key, h[key])
```

Ответ:

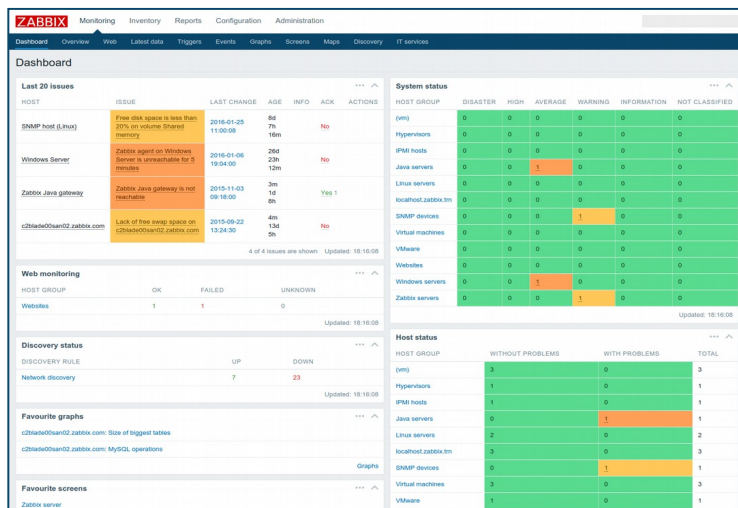
```
$ ./host_get.py
available: 1
description:
disable_until: 0
error:
errors_from: 0
flags: 0
host: Zabbix server
hostid: 10126
...
snmp_available: 0
snmp_disable_until: 0
snmp_error:
snmp_errors_from: 0
status: 0
templateid: 0
```



ПРОИЗВОДИТЕЛЬНОСТЬ



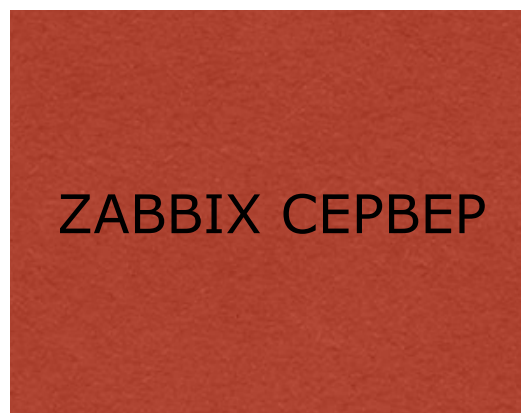
# ПРОСТОЙ ПОТОК ДАННЫХ



Визуализация



Оповещения



Анализ



Сбор данных

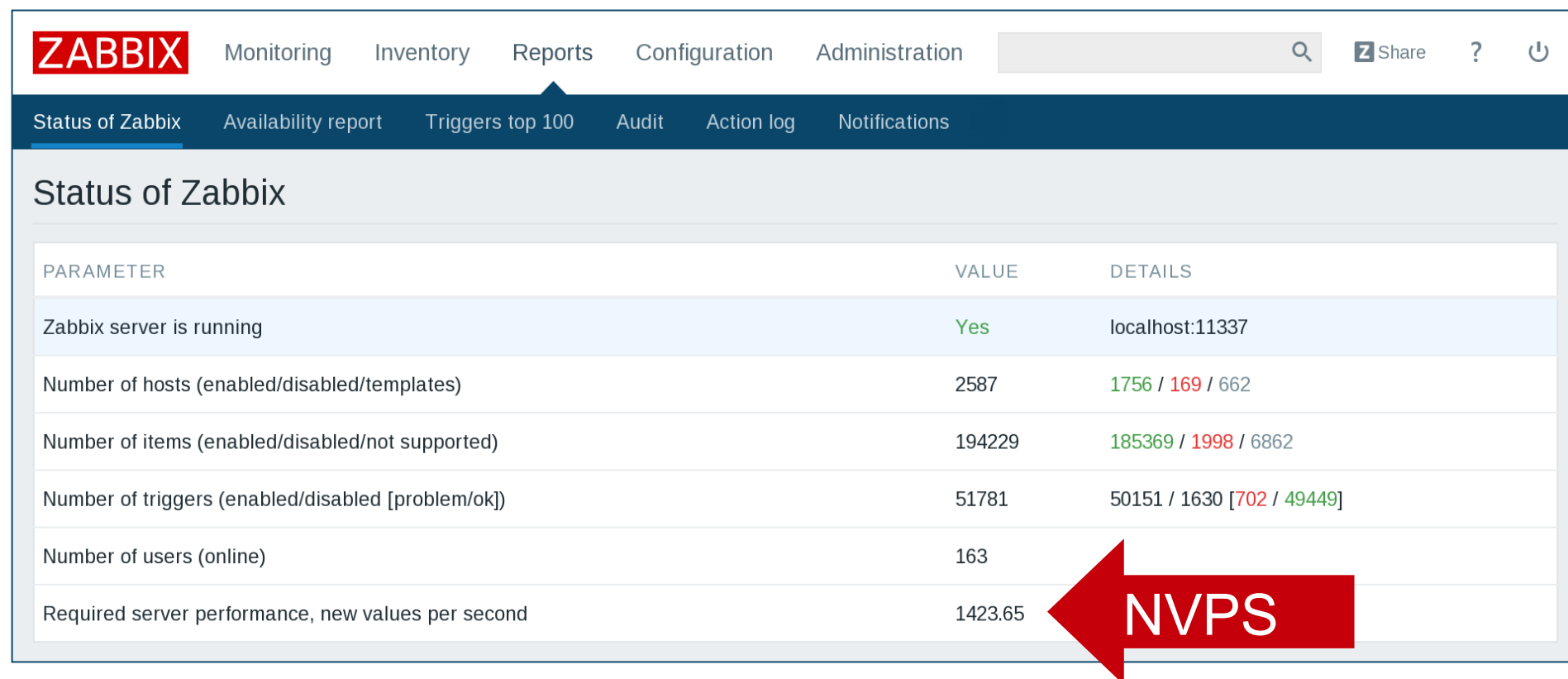


История

# МЕТРИКИ ПРОИЗВОДИТЕЛЬНОСТИ ZABBIX

Количество значений обрабатываемых в секунду (NVPS)

Грубая оценка NVPS видна на ПАНЕЛЕ Zabbix



ZABBIX			
Monitoring Inventory Reports Configuration Administration			
Status of Zabbix Availability report Triggers top 100 Audit Action log Notifications			
Status of Zabbix			
PARAMETER	VALUE	DETAILS	
Zabbix server is running	Yes	localhost:11337	
Number of hosts (enabled/disabled/templates)	2587	1756 / 169 / 662	
Number of items (enabled/disabled/not supported)	194229	185369 / 1998 / 6862	
Number of triggers (enabled/disabled [problem/ok])	51781	50151 / 1630 [702 / 49449]	
Number of users (online)	163		
Required server performance, new values per second	1423.65		

# ИМЕЮЩАЯСЯ ПРОИЗВОДИТЕЛЬНОСТЬ ZABBIX



Hardware: 10 Core CPU, 32ГБ, RAID10 BBWC

Бюджет: около 4K EUR

Zabbix в состоянии собирать **2 миллиона** значений в минуту или около **30.000** значений в секунду

В реальной жизни производительность куда хуже.

**Почему?!**

# ФАКТОРЫ УМЕНЬШАЮЩИЕ ПРОИЗВОДИТЕЛЬНОСТЬ

Тип элементов данных, типы значений, SNMPv3,  
количество триггеров и **сложность** триггеров

Настройки очистки истории и, таким образом, **размер  
базы данных**

Количество пользователей работающих с веб-  
интерфейсом

# ПРОИЗВОДИТЕЛЬНОСТЬ VS КОЛИЧЕСТВО УЗЛОВ СЕТИ

60 элементов данных по узлу сети, частота обновления раз в минуту

Количество узлов сети
100
1 000
10 000

Производительность (значений в секунду)
100
1 000
10 000

300 элементов данных по узлу сети, частота обновления раз в минуту

Количество узлов сети
100
1 000
10 000

Производительность (значений в секунды)
500
5 000
<b>50 000</b>

# МЕДЛЕННО VS БЫСТРО

Что	Медленно	Быстро
Размер базы данных	Большой	Умещается в памяти
Выражения триггеров	<code>min()</code> , <code>max()</code> , <code>avg()</code>	<code>last()</code> , <code>nodata()</code>
Сбор данных	Polling (SNMP, безагентные, пассивный агент)	Trapping (активные агенты)
Типы данных	Текстовый, строковый	Числовой

Анализ истории влияет на производительность Zabbix.  
Но не так сильно. Особенно начиная с Zabbix 2.2

# ВИДИМЫЕ ПРИЗНАКИ ПЛОХОЙ ПРОИЗВОДИТЕЛЬНОСТИ

Очередь Zabbix содержит много задерживающихся элементов данных

Администрирование->Очередь

Частые пропуски данных на графиках, отсутствие данных по некоторым элементам данных

Ошибочные сработки триггеров с функцией `nodata()`

Не отвечающий веб-интерфейс

Отсутствие оповещений

# ПРИЯТНЫЙ ВИД ОЧЕРЕДИ

Queue of items to be updated							Overview ▼
ITEMS	5 SECONDS	10 SECONDS	30 SECONDS	1 MINUTE	5 MINUTES	MORE THAN 10 MINUTES	
Zabbix agent	0	0	0	0	0	0	
Zabbix agent (active)	0	13	0	0	0	0	
Simple check	0	0	0	0	0	0	
SNMPv1 agent	0	0	0	0	0	0	
SNMPv2 agent	0	0	0	0	0	0	
SNMPv3 agent	0	0	0	0	0	0	
Zabbix internal	0	0	0	0	0	0	
Zabbix aggregate	0	0	0	0	0	0	
External check	0	0	0	0	0	0	
Database monitor	0	0	0	0	0	0	
IPMI agent	0	0	0	0	0	0	
SSH agent	0	0	0	0	0	0	
TELNET agent	0	0	0	0	0	0	
JMX agent	0	0	0	0	0	0	
Calculated	0	0	0	0	0	0	





ИДЕНТИФИКАЦИЯ  
И ИСПРАВЛЕНИЕ  
ОБЩИХ ПРОБЛЕМ

# ОСНОВНЫЕ УТИЛИТЫ

top, ntop...

iostat, vmstat, sar

Сам Zabbix

Статистика базы данных, innotop

ps x|grep zabbix\_server

```
zabbix 5010 0.0 0.4 262148 4132 ? S 11:21 0:00 zabbix_server -c /etc/zabbix/zabbix_server.conf
zabbix 5016 0.0 0.3 262156 3196 ? S 11:21 0:00 zabbix_server: configuration syncer [synced configuration in 0.032329 sec, idle 60 sec]
zabbix 5017 0.0 0.2 262148 2256 ? S 11:21 0:00 zabbix_server: db watchdog [synced alerts config in 0.001830 sec, idle 60 sec]
zabbix 5018 0.2 0.9 378996 9192 ? S 11:21 0:01 zabbix_server: poller #1 [got 0 values in 0.000020 sec, idle 1 sec]
zabbix 5020 0.2 0.8 380832 9136 ? S 11:21 0:01 zabbix_server: poller #2 [got 0 values in 0.000018 sec, idle 1 sec]
zabbix 5021 0.2 0.8 378760 8972 ? S 11:21 0:01 zabbix_server: poller #3 [got 9 values in 0.005274 sec, idle 1 sec]
zabbix 5022 0.2 0.8 378732 9016 ? S 11:21 0:01 zabbix_server: poller #4 [got 0 values in 0.000019 sec, idle 1 sec]
zabbix 5023 0.1 0.8 378728 9008 ? S 11:21 0:00 zabbix_server: poller #5 [got 0 values in 0.000016 sec, idle 1 sec]
zabbix 5024 0.0 0.5 364468 5376 ? S 11:21 0:00 zabbix_server: unreachable poller #1 [got 1 values in 2.928658 sec, getting values]
zabbix 5025 0.0 0.5 364468 5360 ? S 11:21 0:00 zabbix_server: unreachable poller #2 [got 0 values in 0.000013 sec, idle 3 sec]
zabbix 5026 0.0 0.4 362396 4836 ? S 11:21 0:00 zabbix_server: unreachable poller #3 [got 0 values in 0.000017 sec, idle 3 sec]
zabbix 5027 0.0 0.5 364468 5396 ? S 11:21 0:00 zabbix_server: unreachable poller #4 [got 0 values in 0.000020 sec, idle 3 sec]
zabbix 5028 0.0 0.4 362336 4832 ? S 11:21 0:00 zabbix_server: unreachable poller #5 [got 0 values in 0.000042 sec, idle 3 sec]
zabbix 5029 0.0 0.2 262148 2992 ? S 11:21 0:00 zabbix_server: trapper #1 [processed data in 0.000000 sec, waiting for connection]
zabbix 5030 0.0 0.3 262184 3792 ? S 11:21 0:00 zabbix_server: trapper #2 [processed data in 0.000658 sec, waiting for connection]
zabbix 5031 0.0 0.3 262292 3804 ? S 11:21 0:00 zabbix_server: trapper #3 [processed data in 0.000540 sec, waiting for connection]
zabbix 5032 0.0 0.2 262148 2996 ? S 11:21 0:00 zabbix_server: trapper #4 [processed data in 0.000000 sec, waiting for connection]
zabbix 5033 0.0 0.3 262292 3804 ? S 11:21 0:00 zabbix_server: trapper #5 [processed data in 0.000575 sec, waiting for connection]
```

# РАЗЛИЧНЫЕ ВИДЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ

*"Я добавил лишь 5 новых узлов сети, Zabbix воспринял их очень вял" :-)*

*"Zabbix ооочень медленный, а у меня всего 48 узлов сети" :-)*

*однако:*

*"Zabbix Milestone достигнут - 1000 узлов сети и все еще добавляем" :-)*

*"Обновление состояния: 8500 узлов сети, 950400 элементов данных, 670340 триггеров, 9550 vps" :-)*

**Так в чем разница?**

# ОСНОВНЫЕ ПРОБЛЕМЫ ПРИ НАЧАЛЬНОЙ УСТАНОВКЕ

Использование шаблонов по умолчанию

Создавайте свои умные шаблоны

Настройки базы данных по умолчанию

Оптимизируйте настройки базы данных для лучшей производительности

Не оптимальная конфигурация Zabbix сервера

Оптимизируйте конфигурацию Zabbix сервера

Настройки очистки истории не подходят по аппаратным спецификациям

Использование устаревших релиз

Всегда используйте последний стабильный релиз!

# КАК УЗНАТЬ ЧТО ПРОИЗВОДИТЕЛЬНОСТЬ БД ПЛОХАЯ?

Файл конфигурации Zabbix сервера, [zabbix\\_server.conf](#)

LogSlowQueries=3000



ОПТИМИЗИРУЙТЕ  
НАСТРОЙКИ  
ZABBIX

# ПОЛУЧАЙТЕ ВНУТРЕННЮЮ СТАТИСТИКУ

Реальное значение VPS

`zabbix[ocache, values, all]`

`zabbix[queue, 1m]` количество элементов данных с задержкой более чем в 1 минуту

Компоненты Zabbix сервера

Alerter, Configuration syncer, DB watchdog, discoverer, escalator, history syncer, http poller, housekeeper, icmp pinger, ipmi poller, poller, trapper, etc.

Кэш Zabbix сервера

history write cache, value cache, trend write cache, vmware cache, etc.

# ПОЛУЧАЙТЕ ВНУТРЕННЮЮ СТАТИСТИКУ

Готовые к использованию шаблоны:

Template App Zabbix Server

Template App Zabbix Proxy

Template App Zabbix Agent



# ВНУТРЕННЯЯ СТАТИСТИКА: ОБЗОР

Кэш истории: свободно в числовом/текстовом кэше (в %)

`zabbix[wcache,history,pfree]`

Кэш истории: свободно в кэше динамики изменений (в %)

`zabbix[wcache,trend,pfree]`

Кэш истории: количество ожидаемых/обрабатываемых значений Zabbix

`zabbix[requiredperformance]`

`zabbix[wcache,values]`

Кэш конфигурации

`zabbix[rcache,buffer,pfree]`

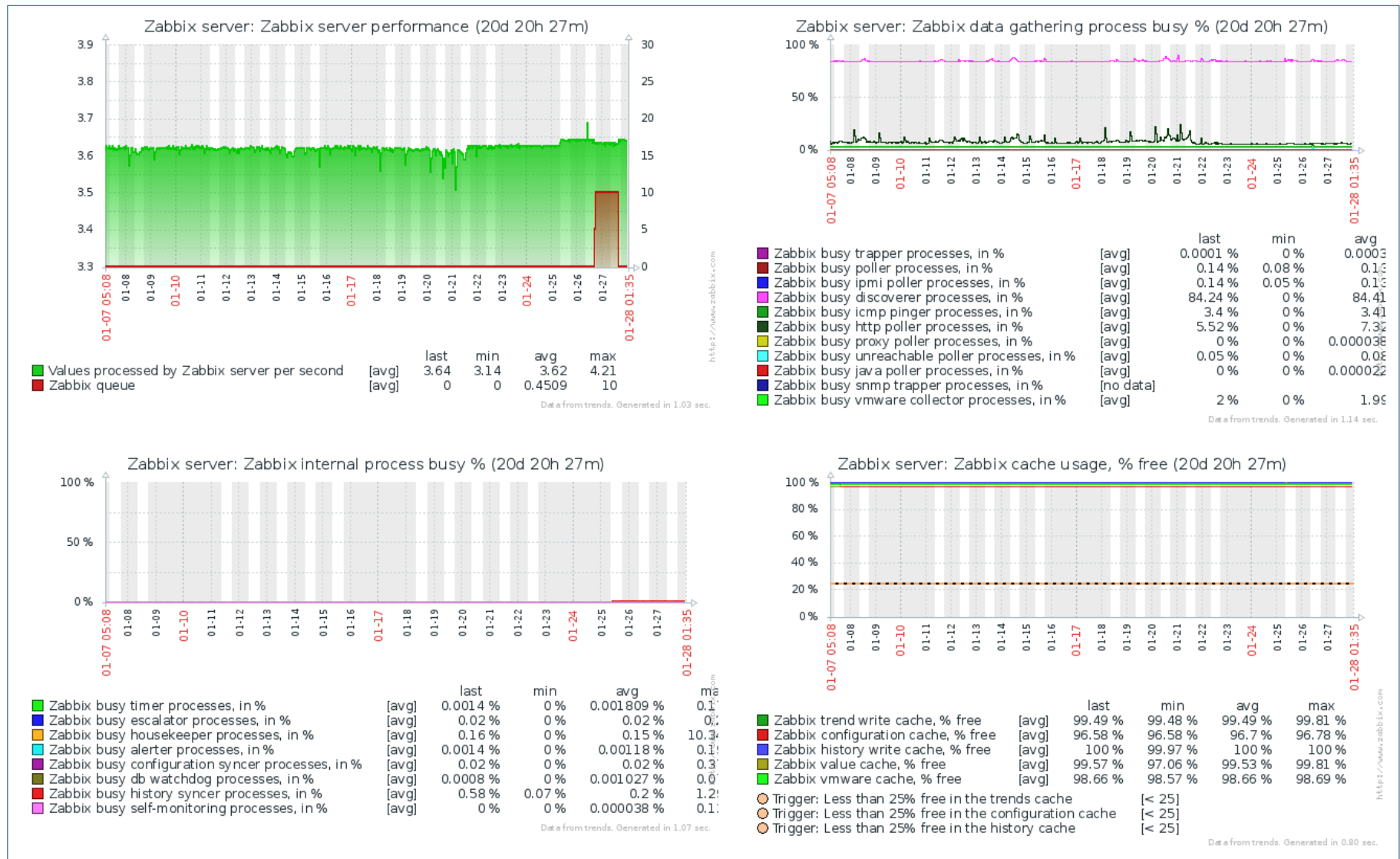
Кэш значений

`zabbix[vcache,cache,mode]`

Количество элементов данных в очереди ожидания

`zabbix[queue,<от>,<до>]`

# КАК ОНО ВЫГЛЯДИТ



# НАСТРОЙТЕ КОЛИЧЕСТВО ПРОЦЕССОВ (ПРИМЕР)

Файл конфигурации Zabbix сервера, [zabbix\\_server.conf](#):

```
StartPollers=80  
StartPingers=10  
StartPollersUnreachable=80  
StartIPMIPollers=10  
StartTrappers=20  
StartDBSyncers=8
```

# НАСТРОЙКА РАЗМЕРА КЭШЕЙ В ПАМЯТИ (ПРИМЕР)

Файл конфигурации Zabbix сервера, [zabbix\\_server.conf](#):

```
VMwareCacheSize=64M  
CacheSize=64M  
HistoryCacheSize=128M  
TrendCacheSize=64M  
HistoryIndexCacheSize = 64M  
ValueCacheSize=64MM
```



ДОПОЛНИТЕЛЬНЫЕ  
ЗАДАЧИ

# ИСПОЛЬЗУЙТЕ ПРОКСИ

Весь сбор данных выполняется  
при помощи прокси



# ПОЛЬЗА ОТ ПРОКСИ

Zabbix прокси "конвертирует" пассивные проверки в активные

Загрузка распределена между прокси

Если один прокси перегружен, узлы сети можно переместить на другой прокси (более активное участие активных проверок)

Простое обслуживание

Кэширование данных при недоступном Zabbix сервере

# ПАРТИЦИОНИРОВАНИЕ ТАБЛИЦ

Способ разделения больших таблиц на маленькие секции.

Имеет смысл для таблиц истории:

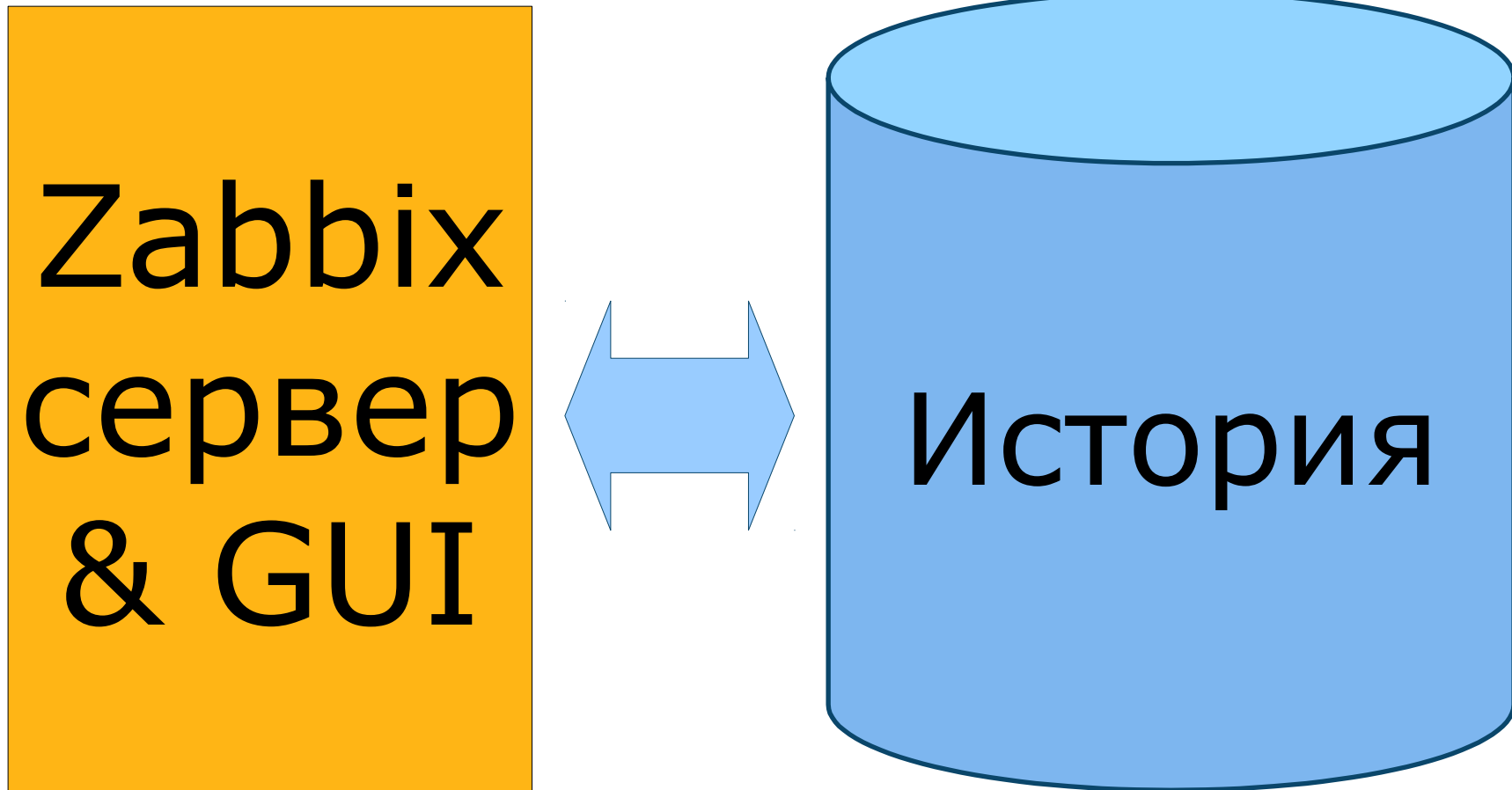
`history_*`, `trends*`, `events`

Польза:

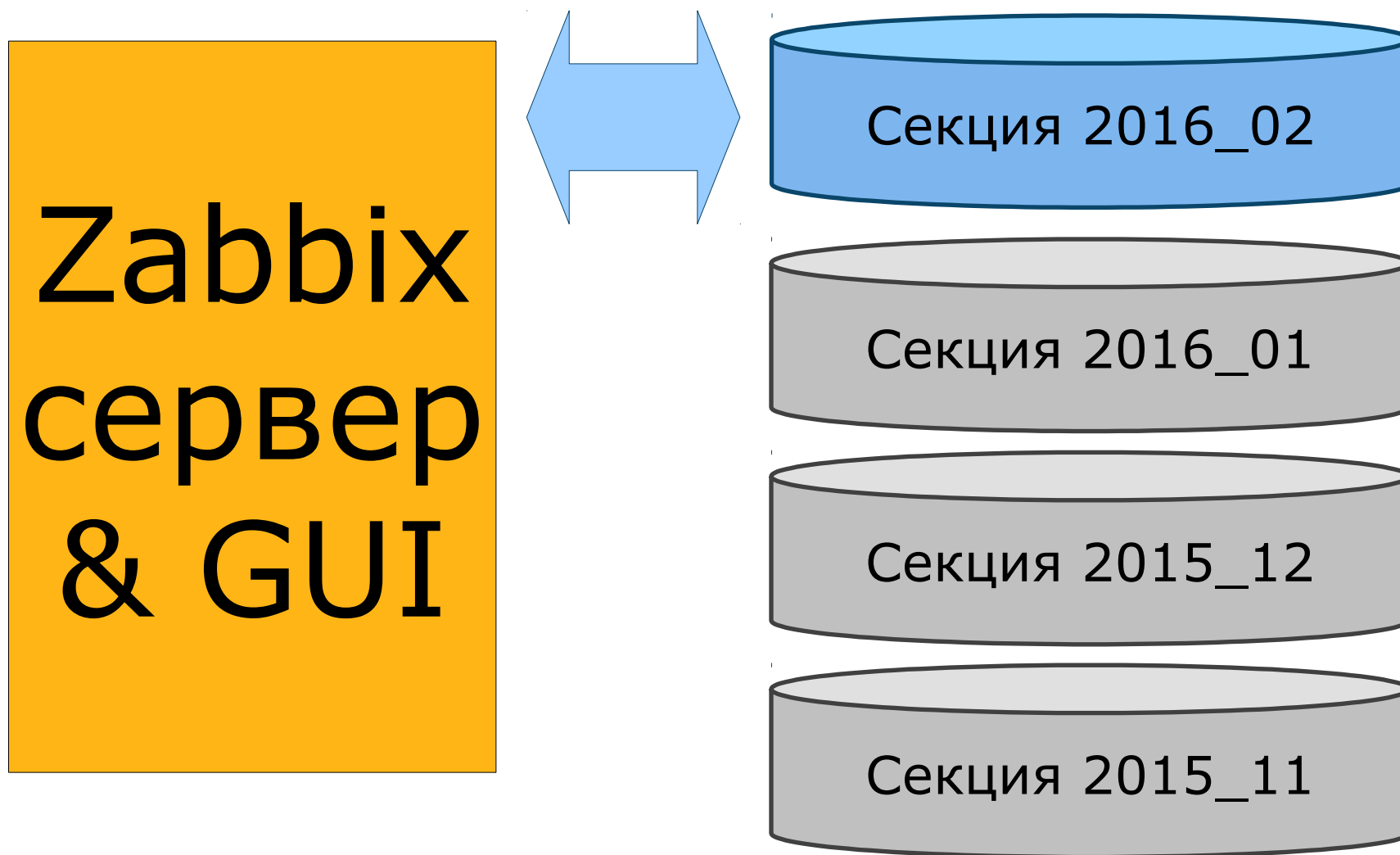
- ✓ Простое **удаление устаревших данных**
- ✓ Значительно более высокая производительность



# БЕЗ ПАРТИЦИОНИРОВАНИЯ ТАБЛИЦ



# С ПАРТИЦИОНИРОВАНИЕМ ТАБЛИЦ



# СПЕЦИФИКА MYSQL

InnoDB лучше чем MyISAM

Файловая система tmpfs в памяти для tmpdir

Поглядывайте за данными

`mysqladmin status`

`mysqladmin variables`

InnoDB

`innodb_file_per_table = 1`

`innodb_buffer_pool_size=<large> (~75% of total RAM)`

`innodb_buffer_pool_instances = 4 (MySQL 5.6 - 8)`

`innodb_flush_log_at_trx_commit = 2`

`innodb_flush_method = O_DIRECT`

`innodb_log_file_size = 256M`

Не используйте

журнал запросов

Бинарные логи, если не используется репликация

`(sync_binlog = 0)`

# МНЕ ВСЁ ЕЩЕ НУЖНА ЛУЧШАЯ ПРОИЗВОДИТЕЛЬНОСТЬ

Запускайте все компоненты Zabbix на  
раздельных серверах!

Zabbix сервер  
8 core CPU  
4ГБ RAM

База данных  
16 core CPU  
64ГБ RAM  
Быстрое  
хранилище

Веб-интерфейс  
Быстрый CPU  
4ГБ RAM



# Сводка



Определите и исправьте  
основные проблемы

Шаг 1



Оптимизируйте  
конфигурацию Zabbix

Шаг 2



Проверьте  
всё ещё раз

Шаг 3



# ПАМЯТКА

- ✓ Собирается внутренняя статистика Zabbix!  
В противном случае вы не узнаете ничего о состоянии Zabbix
- ✓ Конфигурация Zabbix оптимизирована
- ✓ Производительность базы данных оптимизирована
- ✓ Очистка истории не используется для таблиц истории

# ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Руководства по оптимизации производительности  
MySQL & PostgreSQL

Партиционирование таблиц в Zabbix

MySQL:

[http://zabbix.org/wiki/Docs/howto/mysql\\_partitioning](http://zabbix.org/wiki/Docs/howto/mysql_partitioning)

PostgreSQL:

[https://www.zabbix.org/wiki/Docs/howto/zabbix2\\_postgresql\\_partitioning](https://www.zabbix.org/wiki/Docs/howto/zabbix2_postgresql_partitioning)

Внутренние проверки Zabbix

<http://blog.zabbix.com/monitoring-how-busy-zabbix-processes-are>

[http://www.zabbix.com/documentation/3.0/ru/manual/config/items#внутренние\\_проверки](http://www.zabbix.com/documentation/3.0/ru/manual/config/items#внутренние_проверки)



ПРОБЛЕМЫ И  
РЕШЕНИЯ



# ВАЖНЫЕ ЗАМЕЧАНИЯ 1

## **Производительность**

Производительность базы данных критична  
Производительность сети

## **Интеграция с существующими системами**

Helpdesk  
Управление конфигурациями  
LDAP или решения единого входа

## **Вопросы обслуживания**

Изменения в конфигурации  
Управление пользователями и оповещениями

# ВАЖНЫЕ ЗАМЕЧАНИЯ 2

## **Обновления программного обеспечения**

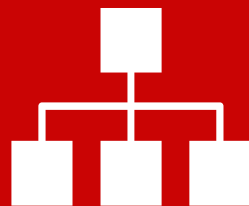
Обратная совместимость

## **Установка и настройка агентов**

Использование централизованного репозитория

Использование правильных пакетов (DEB, RPM)

Использование системы управления конфигурацией (puppet, и т.п.)



ВЫСОКАЯ

ДОСТУПНОСТЬ И  
ИЗБЫТОЧНОСТЬ

# КЛАСТЕРИЗАЦИЯ И ИЗБЫТОЧНОСТЬ

## **Основное**

Разделяемое хранилище  
Виртуальный IP

## **Решения кластеризации**

Linux HA (OpenAIS/Corosync, Pacemaker)

**Репликация базы данных, если не используется  
разделяемое хранилище**

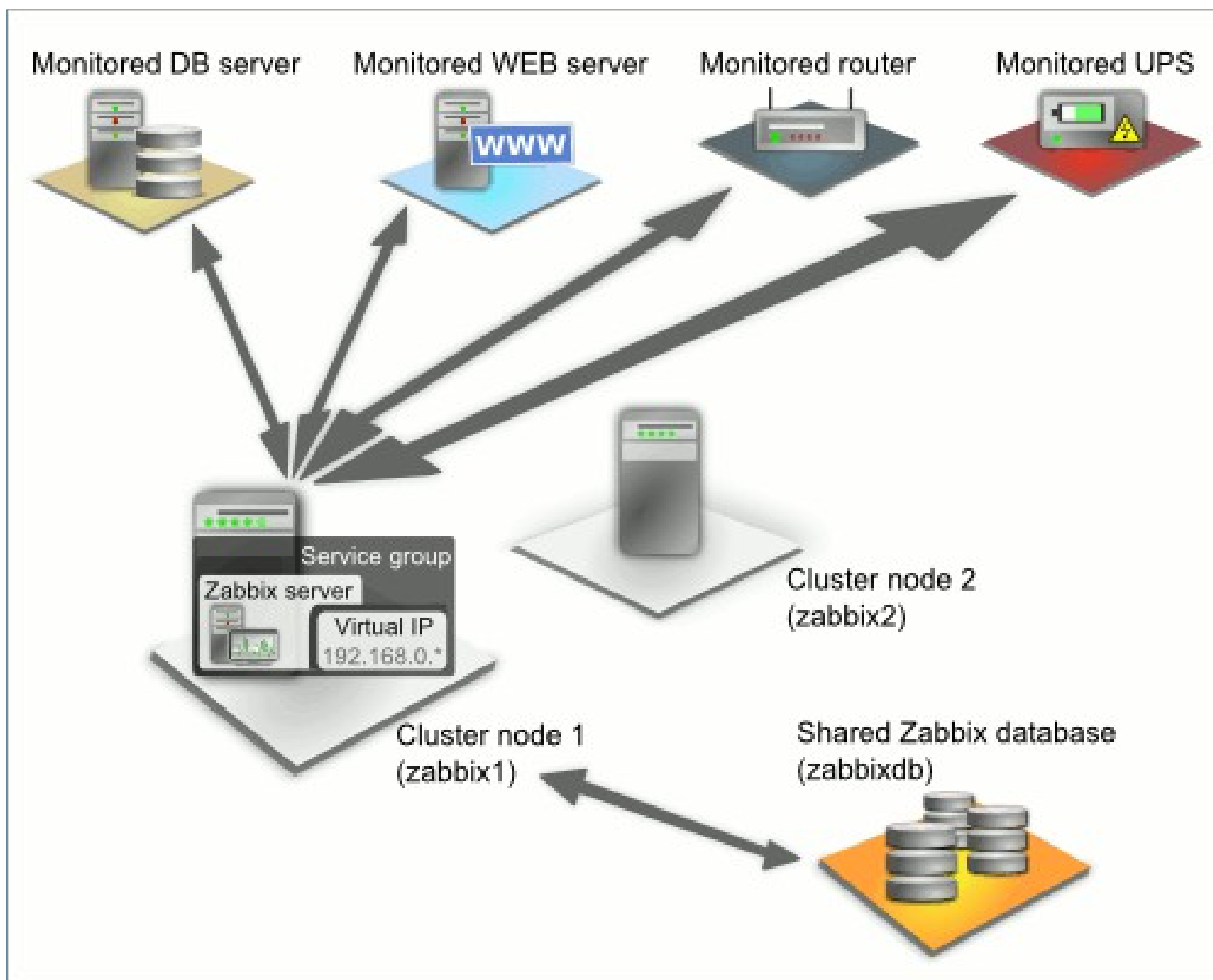
**Мастер-мастер репликация для избыточности**

**Galera Cluster для MySQL**

## **HOWTO**

[https://www.zabbix.org/wiki/Docs/howto/high\\_availability](https://www.zabbix.org/wiki/Docs/howto/high_availability)

# УСТАНОВКА РЕЗЕРВА



# ПРАКТИЧЕСКАЯ УСТАНОВКА

Настройте два Zabbix сервера с разделяемой базой данных

Используйте виртуальный IP

Группа сервисов отказоустойчивости Zabbix сервер + IP



СЕРВИСЫ  
ZABBIX

# КОММЕРЧЕСКИЕ СЕРВИСЫ





# ПРИСОЕДИНЯЙТЕСЬ К МЕРОПРИЯТИЯМ

<http://www.zabbix.com/events.php>



The banner is divided into two horizontal sections. The top section has a white background with faint, stylized mountain peaks. It contains the text 'Zabbix Conference LatAm' in a large, dark blue font, followed by the tagline 'Share your passion and search for new knowledge' in a smaller, orange font, and the dates 'April 15 - 16, 2016 | Porto Alegre, Brazil' in a small, dark blue font. The bottom section features a photograph of the Porto Alegre skyline at sunset, with buildings and a body of water. Overlaid on this image is the text 'Join the Largest Zabbix Event in Latin America!' in a large, dark blue font. To the right, there is a red box with 'ZABBIX' in white, followed by '2016' in a large, dark blue font, and 'Conference' in a large, dark blue font. Below this, a dark orange box contains the text 'LATIN AMERICA' in white. At the bottom of the image, the dates 'April 15 - 16, 2016 | Porto Alegre, Brazil' are repeated in a small, dark blue font. In the bottom right corner of the image, there are four small, colored dots (blue, green, yellow, red) arranged horizontally.

Zabbix Conference LatAm

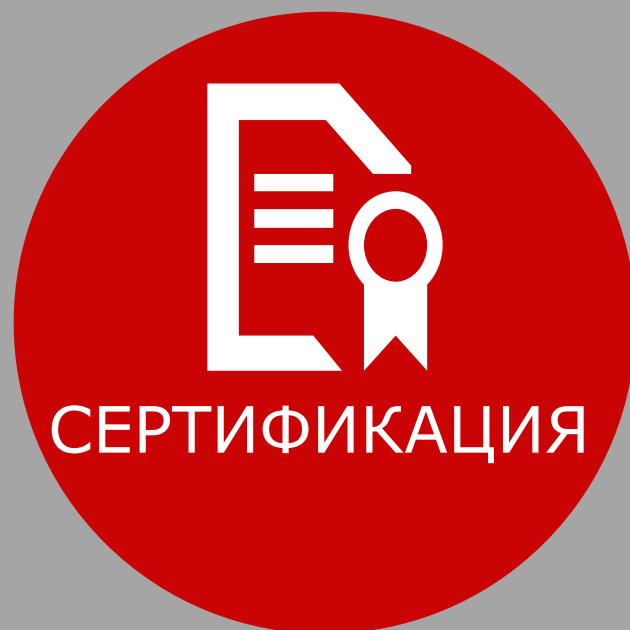
Share your passion and search for new knowledge

April 15 - 16, 2016 | Porto Alegre, Brazil

Join the Largest Zabbix Event  
in Latin America!

**ZABBIX** 2016  
Conference  
LATIN AMERICA

April 15 - 16, 2016 | Porto Alegre, Brazil



# ОБРАТНАЯ СВЯЗЬ

Форма обратной связи:

[http://www.zabbix.com/training\\_application\\_form.php](http://www.zabbix.com/training_application_form.php)

# ВОПРОСЫ?

**ZABBIX**

The Enterprise class Monitoring Solution for Everyone

[WWW.ZABBIX.COM](http://WWW.ZABBIX.COM)



ВРЕМЯ ДЛЯ ОТДЫХА :)