Ранее я рассматривал различные конфигурации для мониторинга параметров и программ в windows и linux. Сейчас я хочу рассказать, как мониторить с помощью Zabbix произвольный сервис (службу), который работает либо локально на сервере, либо на внешнем tcp порту. Это может быть что угодно — ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet или любой другой сервис.

Содержание:

- 1 Введение
- 2 Описание работы простых проверок (simple check)
- 3 Мониторинг доступности сервиса по сети
- 4 Мониторинг локальной службы в linux
- 5 Заключение
- 6 Дополнительные материалы по Zabbix

Введение

Если у вас еще нет своего сервера для мониторинга, то рекомендую материалы на эту тему. Для тех, кто предпочитает систему CentOS:

1. Установка CentOS 8.
2. Настройка CentOS 8.
3. Установка и настройка zabbix сервера.

То же самое на Debian 10, если предпочитаете его:

1. Установка Debian 10.
2. Базовая настройка Debian.
3. Установка и настройка zabbix на debian.

В заббикс существуют различные способы получать данные для мониторинга. Наиболее распространенные источники информации:

- Zabbix агент. Устанавливается на наблюдаемую машину и отправляет данные на сервер мониторинга.
- SNMP агент. Чаще всего присутствует на устройстве, либо может быть установлен на сервер.
- Простые проверки — simple check. Выполняются непосредственно на сервере zabbix с помощью встроенных инструментов, не требуют дополнительных действий со стороны хоста.
- Внешние проверки — external checks. Как и простые проверки выполняются на сервере мониторинга, но не встроенными средствами, а внешними скриптами.

Есть и другие способы получения данных. Не буду их все перечислять, ознакомиться с ними можно в соответствующем разделе официальной документации. В нашем случае мы воспользуемся первыми двумя способами для мониторинга служб и сервисов в linux.

Тут можно пойти разными путями. Меня интересует мониторинг различных линукс служб, работающих как локально (samsdaemon, postgrey) в пределах конкретного сервера, так и для публичного доступа по сети, в частности squid, smtp, imap, http. Первое, что пришло в голову, это использовать итем с ключом service_state[]. Но как оказалось, этот тип данных снимает значения только с системных служб windows. Я не сразу это понял и некоторое время повозился в консоли, не понимая, почему при тестировании значения получаю сообщение, что данный item не поддерживается:

```
# zabbix_agent -t service_state[sshd]
service_state[sshd] [m|ZBX_NOTSUPPORTED] [Unsupported item key.]
```

Дальше придумал через UserParameter запускать какой-нибудь скрипт, который будет проверять запущен ли сервис в системе или нет. Например с помощью *ps ax | grep squid*. В принципе, рабочий вариант, но мне казалось, что такую простую задачу можно решить проще и быстрее, без создания на каждом хосте скрипта и изменения файла конфигурации. И я не ошибся. Есть 2 различных способа мониторинга служб (сервисов) в linux с помощью zabbix. Рассмотрим первый из них.

Описание работы простых проверок (simple check)

Стал искать материал на эту тему и прочитал про simple check (простые проверки) в zabbix. Оказалось, это то, что нужно. Их можно использовать для безагентских проверок удаленных сервисов. При этом требуется минимум настроек и только на сервере. Можно создать шаблон и распространить на любое количество хостов.

Принцип работы простых проверок следующий. Вы создаете item, в нем указываете тип **simple check**, в качестве ключа выбираете **net.tcp.service**[сервис,<ip>,<порт>], указываете соответствующие параметры в скобках и все. Сервер сам начинает опрашивать указанный сервис и возвращать в зависимости от его доступности **0** или **1**. Устанавливать агент на хост не нужно. Мониторить можно любую сетевую службу, к которой есть доступ по tcp.

Возвращаемые значения net.tcp.service

0 сервис недоступен

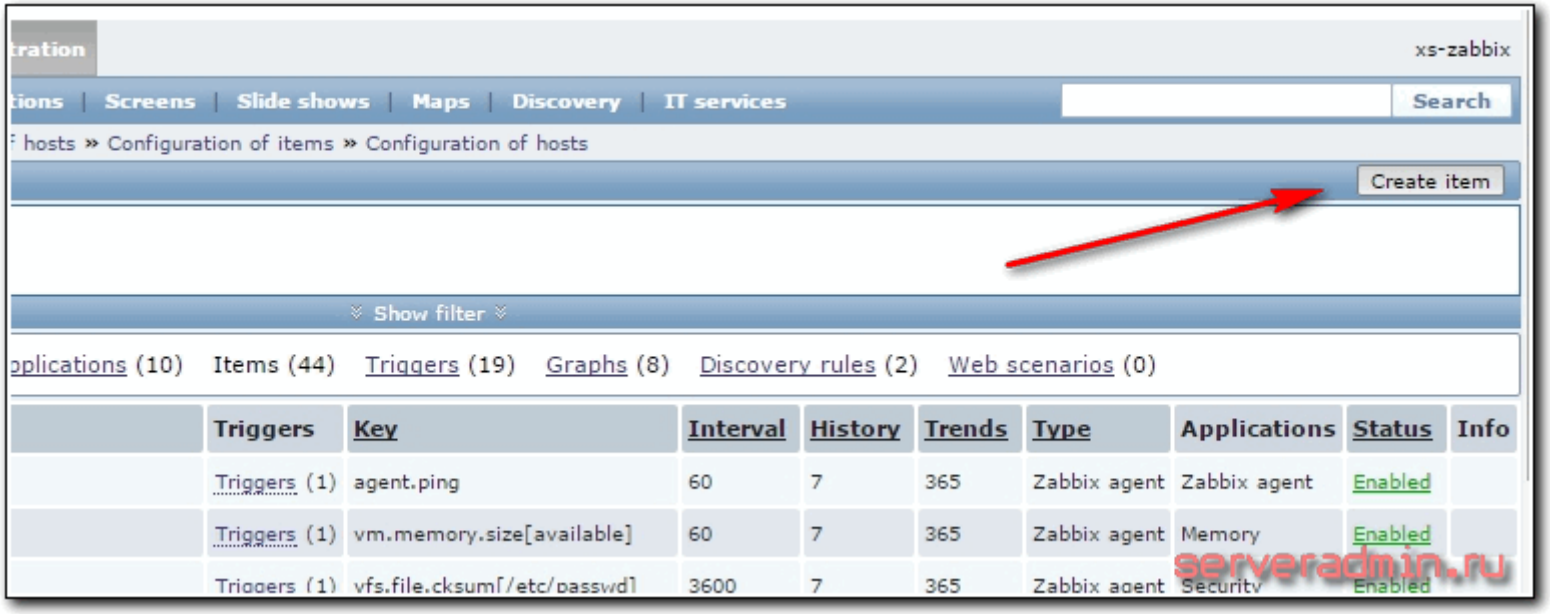
1 сервис работает

Всего в простых проверках доступны 5 ключей. Подробнее о них читайте в документации. В данном случае меня будет интересовать только ключ net.tcp.service. В нем предопределены алгоритмы проверок следующих служб: ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, https, telnet. Детали реализации проверки каждой службы описаны тут. Если вы мониторите службу, которая не входит в указанный выше список, то происходит просто проверка возможности подключения, без отправки и получения каких-то данных.

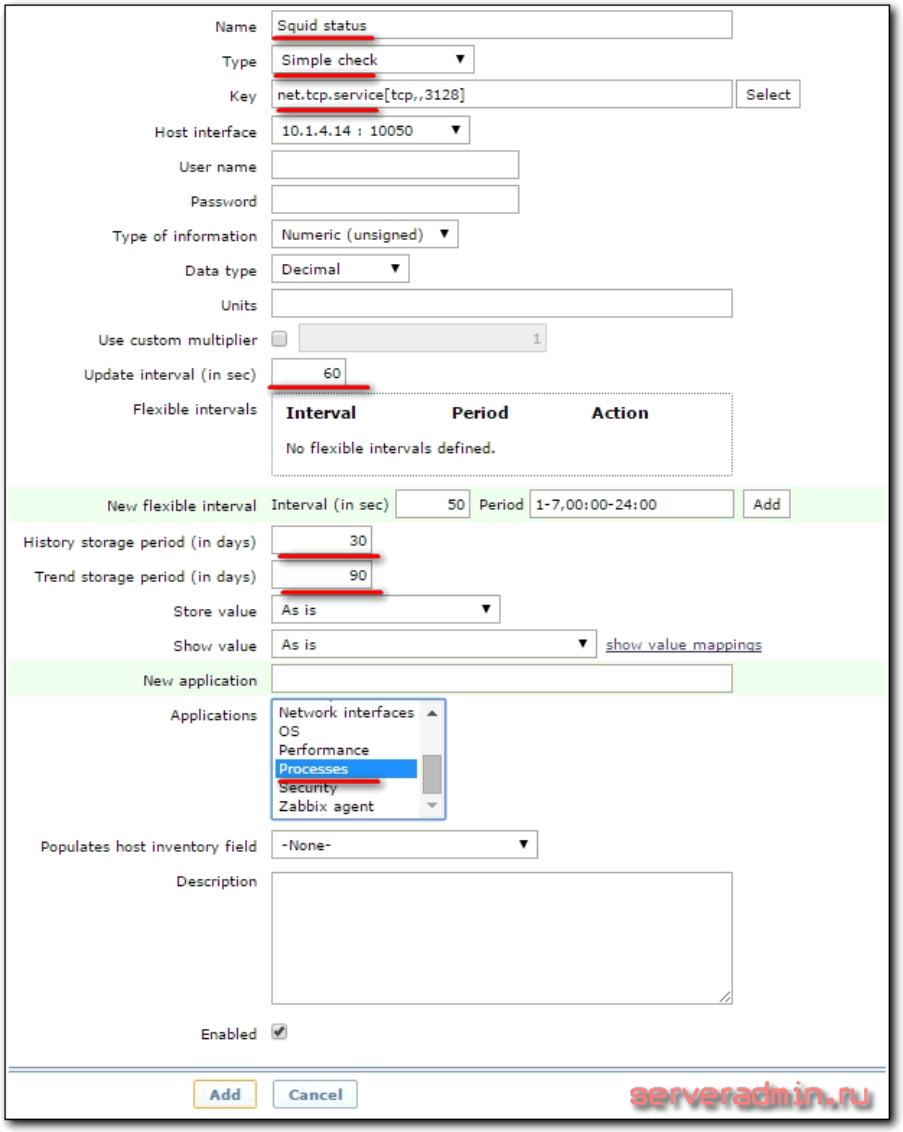
Мониторинг доступности сервиса по сети

В качестве примера настроим мониторинг доступности прокси сервера **squid**. Он запущен на linux сервере и этот хост уже добавлен на сервер мониторинга. Данные поступают с помощью агента, но мы не будет его использовать. Просто создадим одиночный **item** для проверки доступности squid и **trigger** для отправки уведомления, если сервис не работает. В данном примере я рассмотрю настройку на примере конкретного хоста. Если у вас несколько серверов с squid, которые вы хотите мониторить, то все элементы лучше создать не отдельно на каждом хосте, а сразу сделать template и назначить его нужным хостам.

Итак, идем в **Configuration** -> **Hosts** и выбираем там хост, на котором установлен squid. Переходим в раздел **Items** и нажимаем **Create item**.



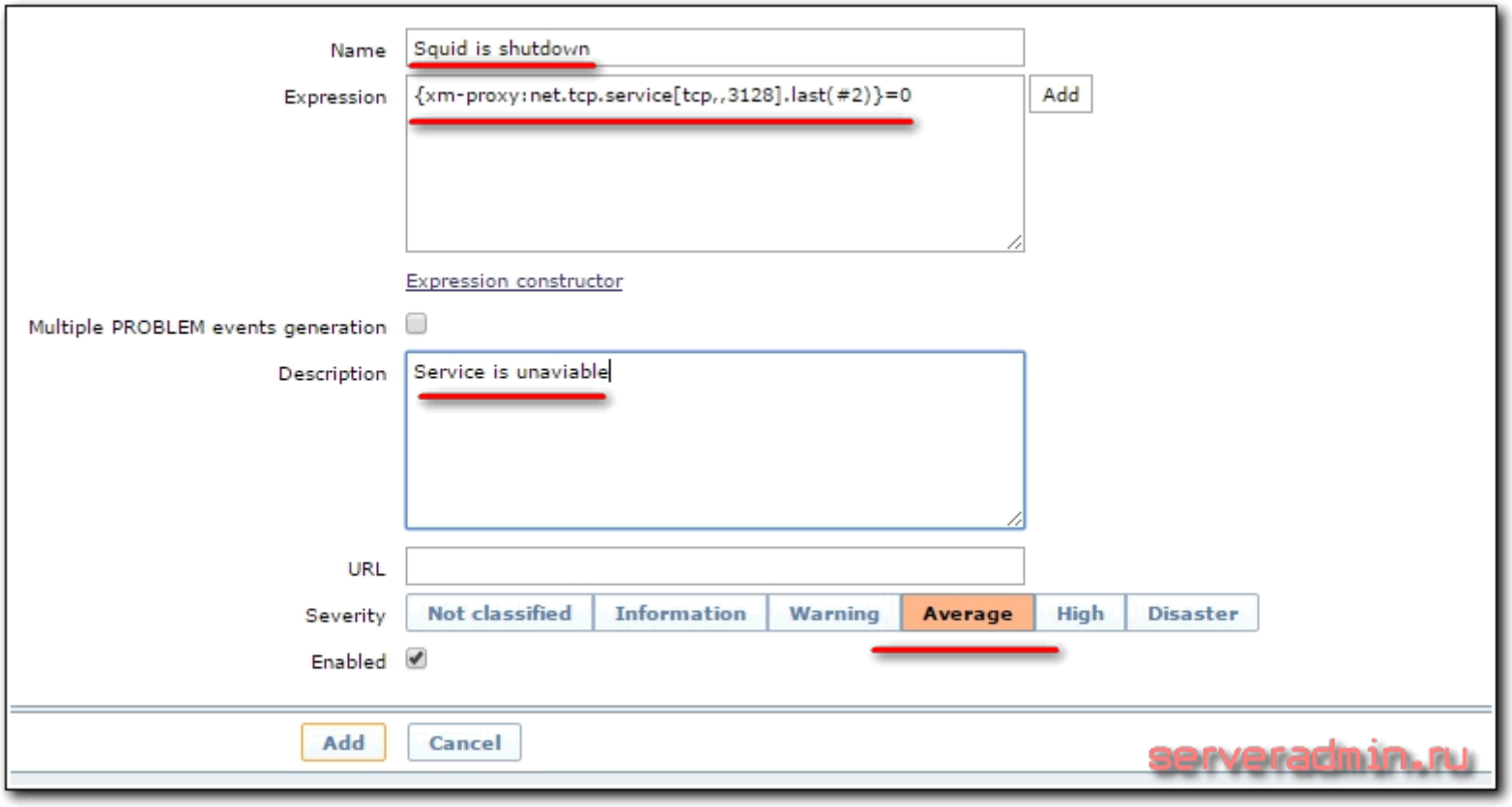
Заполняем необходимые параметры элемента.


Name
Squid status
Type
Simple check
Key
net.tcp.service[tcp,,3128]
Select
Host interface
10.1.4.14 : 10050
User name
Password
Type of information
Numeric (unsigned)
Data type
Decimal
Units
Use custom multiplier
1
Update interval (in sec)
60
Flexible intervals
Interval
Period
Action
No flexible intervals defined.
New flexible interval
Interval (in sec)
50
Period
1-7,00:00-24:00
Add
History storage period (in days)
30
Trend storage period (in days)
90
Store value
As is
Show value
As is
show value mappings
New application
Applications
Network interfaces
OS
Performance
Processes
Security
Zabbix agent
Populates host inventory field
-None-
Description
Enabled
Add
Cancel
serveradmin.ru

Обязательно заполнить первые 3, остальные на ваше усмотрение. Я считаю, что проверять каждые 30 секунд и хранить 90 дней информацию излишне, поэтому изменяю эти параметры в сторону увеличения.

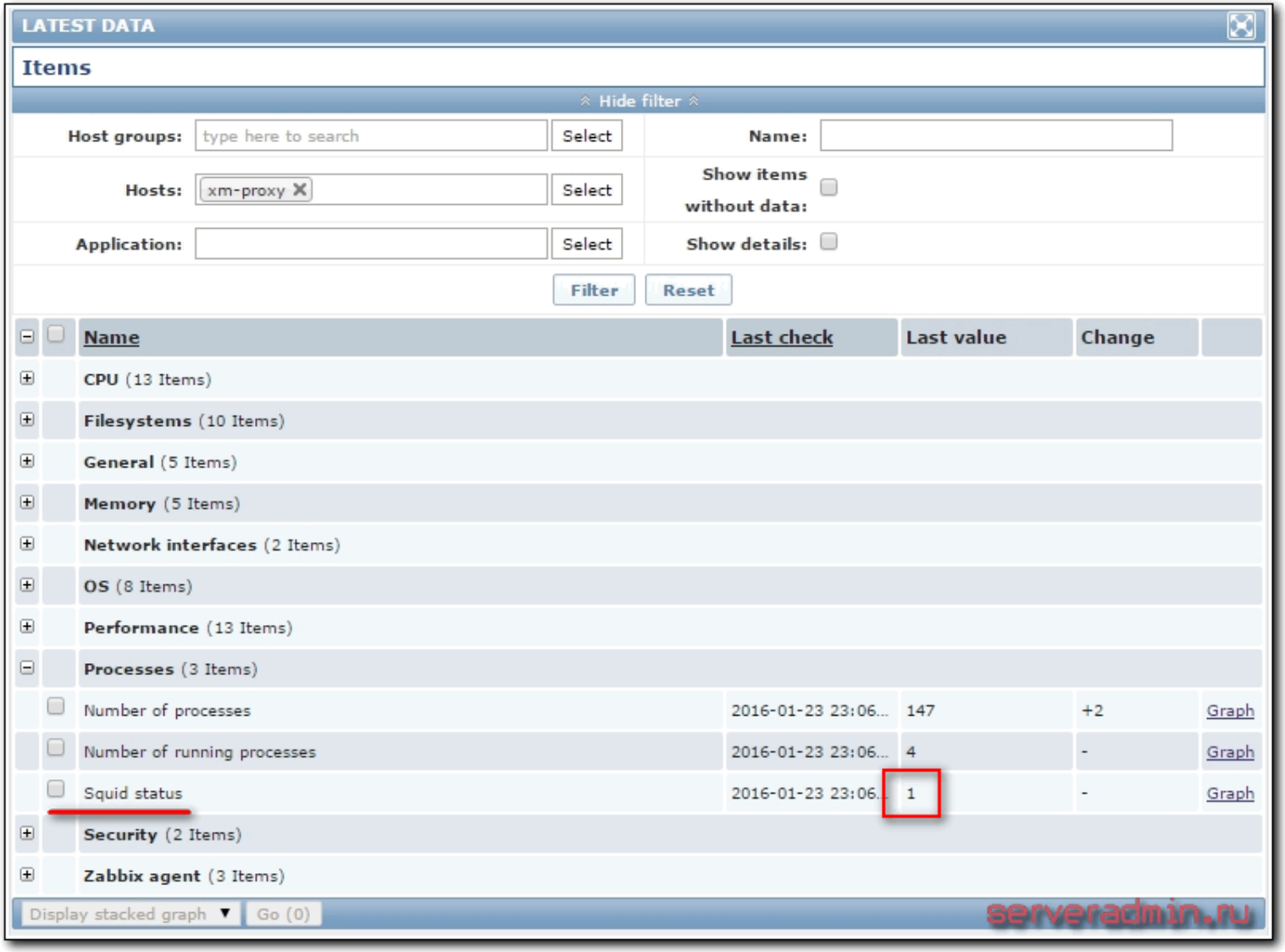
Squid status	Имя итема.
Simple check	Тип итема.
net.tcp.service[tcp,,3128]	Проверять tcp порт 3128 на указанном хосте. Если вы проверяете статус службы, расположенной не на том же хосте, к которому прикрепляете item, то после первой запятой можно указать необходимый адрес.

Сразу создадим триггер, который в случае возврата в последних двух проверках значения итемом 0, будет отправлять уведомление о том, что служба недоступна. Для этого идем в раздел triggers и жмем Create trigger. Заполняем параметры элемента.



Выражение *{xm-proxy:net.tcp.service[tcp,,3128].last(#2)}=0* означает, что триггер срабатывает, если 2 последних значения были равны 0.

Ждем пару минут и идем в Latest data проверять поступаемые значения.


LATEST DATA
Items
Hide filter
Host groups: type here to search Select
Name:
Hosts: xm-proxy Select
Show items without data:
Application: Select
Show details:
Filter Reset
Name
Last check
Last value
Change
CPU (13 Items)
Filesystems (10 Items)
General (5 Items)
Memory (5 Items)
Network interfaces (2 Items)
OS (8 Items)
Performance (13 Items)
Processes (3 Items)
Number of processes 2016-01-23 23:06... 147 +2 Graph
Number of running processes 2016-01-23 23:06... 4 - Graph
Squid status 2016-01-23 23:06... 1 - Graph
Security (2 Items)
Zabbix agent (3 Items)
Display stacked graph
Go (0)
serveradmin.ru

Чтобы проверить работу триггера, достаточно зайти на сервер и остановить squid. Если вы все сделали правильно, то после второй проверки, которая определит, что squid не отвечает по заданному адресу, вы получите уведомление на почту об этом. Если у вас не настроены или не работают уведомления на почту в zabbix, то читайте мою статью на эту тему.

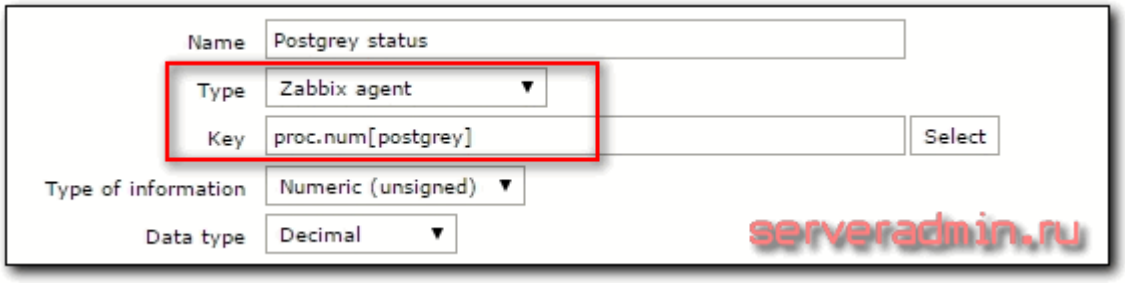
Мониторинг локальной службы в linux

С мониторингом удаленного tcp сервиса разобрались, а что делать, если служба работает локально и к ней невозможно подключиться из вне. Тут уже не обойтись без установки zabbix агента. Если он установлен на хосте, то можно воспользоваться итемом с ключом **proc.num**. Этот ключ возвращает в качестве значения количество запущенных процессов. И если таких процессов больше одного, можно считать, что служба запущена.

Рассмотрим на примере мониторинга службы postgrey, реализующей greylist для борьбы со спамом. Она работает локально на почтовом сервере linux и является критическим сервисом, так как без него почтовый сервер postfix не будет принимать почту, выдавая временную ошибку почтовой системы. Проверим работу ключа proc.num:

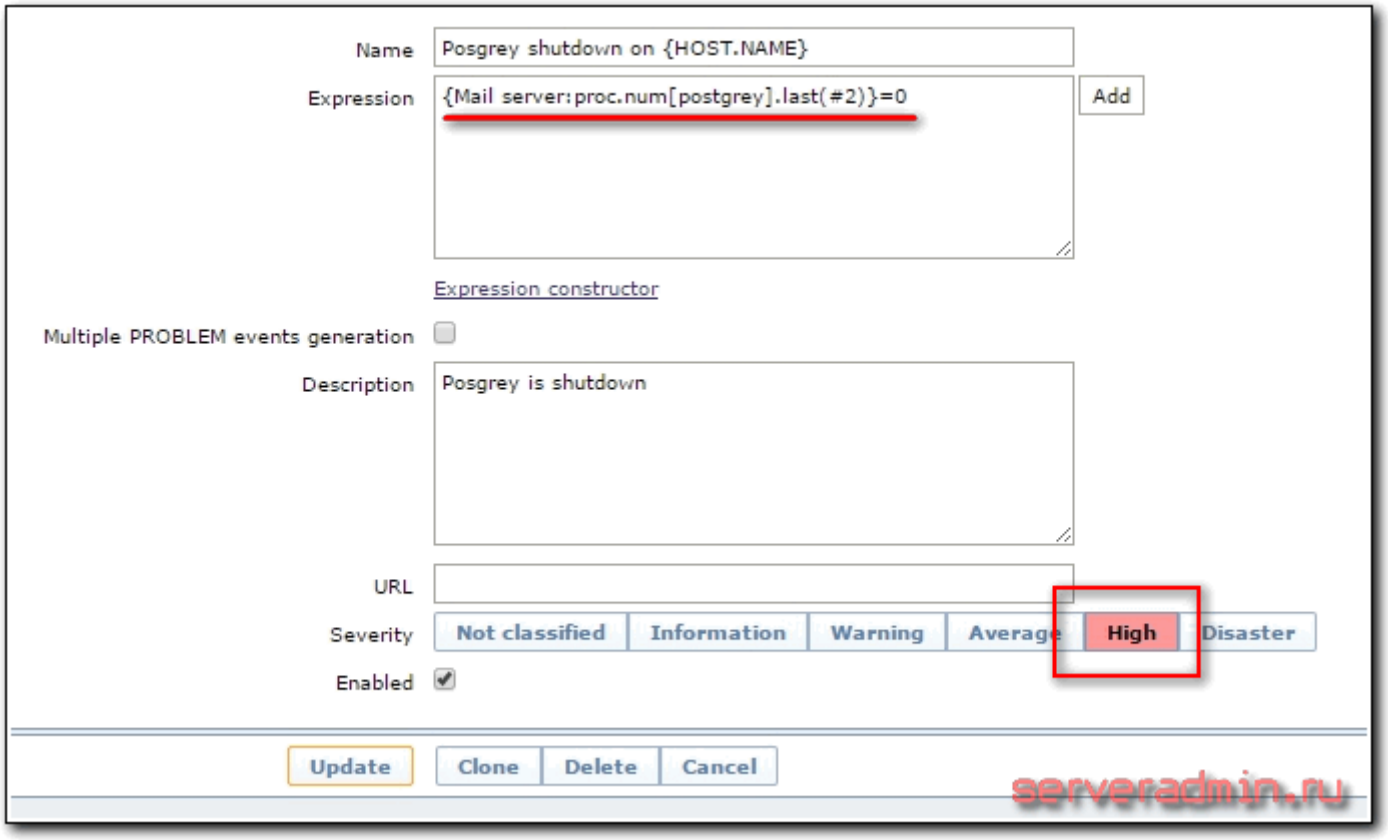
```
# zabbix_agentd -t proc.num[postgrey]
proc.num[postgrey] [u|1]
```

Все в порядке, zabbix агент возвращает значение 1 при запущенном сервисе. Идем на сервер мониторинга, выбираем хост или шаблон и создаем новый item.

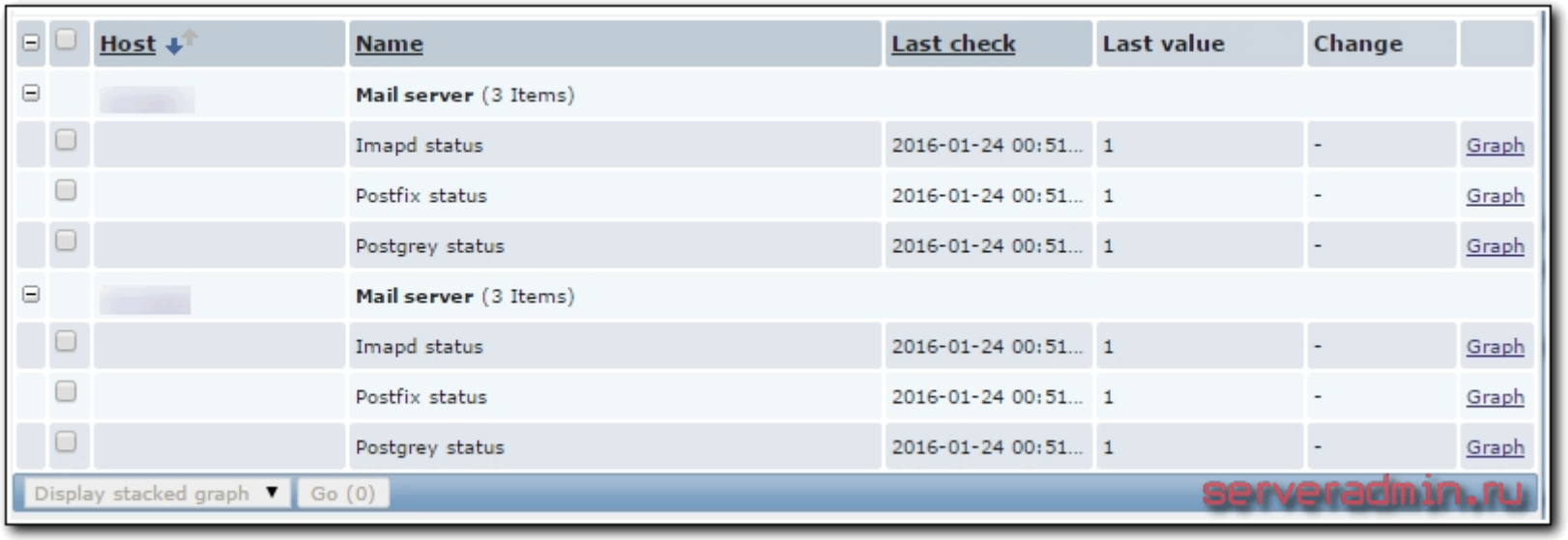


Показываю только основные параметры, остальные устанавливайте на свой вкус. Я лишь рекомендую не делать слишком частые проверки. В большинстве случаев в этом нет необходимости, а нагрузка на сервер постоянно растет при добавлении новых итемов.

Создаем триггер с оповещением о недоступности сервиса. При последних двух значениях равных **0** срабатываем.



Я настраиваю триггер в шаблоне, поэтому сразу для удобства в названии триггера указываю маску для имени, чтобы было понятно в оповещении, на каком хосте сработал триггер. Как обычно, проверить поступаемые значения можно в **Latest data**.



Вот и все. Мы настроили мониторинг локальных служб linux в заббиксе.

Заключение

В своем материале я рассмотрел два различных способа, с помощью которых можно мониторить любой удаленный сервис по протоколу tcp, либо локальную службу на сервере linux. Конкретно в моих примерах можно было воспользоваться вторым способом в обоих случаях. Я этого не сделал, потому что первым способом я не просто проверяю, что служба запущена, я еще и обращаюсь к ней по сети и проверяю ее корректную работу для удаленного пользователя.

Разница тут получается вот в чем. Допустим, сервер squid у вас запущен и работает на сервере. Проверка работы локальной службы показывает, что сервис работает и возвращает значение 1. Но к примеру, вы настраивали firewall и где-то ошиблись. Сервис стал недоступен по сети, пользователи не могут им пользоваться. При этом мониторинг будет показывать, что все в порядке, служба запущена, хотя реально она не может обслужить запросы пользователей. В таком случай только удаленная проверка покажет, что с доступностью сервиса проблемы и надо что-то делать.

Из этого можно сделать вывод, что система мониторинга zabbix предоставляет огромные возможности по мониторингу. Какой тип наблюдения и сбора данных подойдет в конкретном случае нужно решать на месте, исходя из сути сервиса, за которым вы наблюдаете.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.

Дополнительные материалы по Zabbix

Рекомендую полезные материалы по Zabbix:

Настройки системы

• Установка 4.0
• Обновление 3.0 -> 3.2
• Обновление 3.4 -> 4.0
• Установка Zabbix Proxy
• Работа на NGINX

Видео и подробное описание установки и настройки Zabbix 4.0, а также установка агентов на linux и windows и подключение их к мониторингу.
Подробное описание обновления системы мониторинга zabbix версии 3.4 до новой версии 4.0.
Пошаговая процедура обновления сервера мониторинга zabbix 2.4 до 3.0. Подробное описание каждого шага с пояснениями и рекомендациями.
Подробное описание установки и настройки zabbix proxy для организации распределенной системы мониторинга. Все показано на примерах.
Подробное описание установки системы мониторинга Zabbix на веб сервер на базе nginx + php-fpm.

Мониторинг служб и сервисов

• Температура процессора
• Nginx и php-fpm
• Mysql репликация
• Службы Linux
• Рейд mdadm
• Транки Asterisk
• Synology

Мониторинг температуры процессора с помощью zabbix на Windows сервере с использованием пользовательских скриптов.
Настройка полноценного мониторинга web сервера nginx и php-fpm в zabbix с помощью скриптов и пользовательских параметров.
Мониторинг репликации mysql с помощью Zabbix. Подробный разбор методики и тестирование работы.
Описание настройки мониторинга tcp служб с помощью zabbix и его инструмента простых проверок (simple checks)
Настройка мониторинга рейда mdadm с помощью zabbix. Подробное пояснение принципа работы и пошаговая инструкция.
Подробное описание мониторинга регистраций транков (trunk) в asterisk с помощью сервера мониторинга zabbix.
Подробная инструкция со скриншотами по настройке мониторинга по snmp дискового хранилища synology с помощью сервера мониторинга zabbix.

Мониторинг различных значений

- Мониторинг сайта
- Мониторинг бэкапов
- Размер бэкапа
- Делегирование домена
- Значения из текстового файла
- Мониторинг логов

Настройка мониторинга web сайта в zabbix. Параметры для наблюдения - доступность сайта, время отклика, скорость доступа к сайту.
Один из способов мониторинга бэкапов с помощью zabbix через проверку даты последнего изменения файла из архивной копии с помощью vfs.file.time.
Подробное описание настройки мониторинга размера бэкапов в Zabbix с помощью внешних скриптов.
Пример настройки мониторинга за временем делегирования домена с помощью Zabbix и внешнего скрипта. Все скрипты и готовый шаблон представлены.
Пример распознавания и мониторинга за изменением значений в обычных текстовых файлах с помощью zabbix.
Описание мониторинга лог файлов в zabbix на примере анализа лога программы apcupsd. Отправка оповещений по событиям из лога.