

4 Прокси

Обзор

Zabbix прокси - это процесс, способный собирать данные мониторинга с одного или нескольких наблюдаемых устройств и отправлять эту информацию Zabbix серверу, таким образом прокси работает от имени сервера. Все собранные данные локально буферизуются и затем отправляются Zabbix серверу, которому принадлежит этот прокси.

Развертывание прокси не обязательно, но может быть очень полезно для распределения нагрузки одиночного Zabbix сервера. Если только прокси собирают данные, то обработка этих данных на сервере не так сильно нагружает CPU и I/O диска.

Zabbix прокси - идеальное решение для централизованного мониторинга удаленных объектов, филиалов и сетей, где отсутствуют локальные администраторы.

Для Zabbix прокси требуется отдельная база данных.

Zabbix прокси поддерживает следующие базы данных SQLite, MySQL и PostgreSQL. Вы можете использовать Oracle или IBM DB2 на свой собственный риск, при этом возможны определенные ограничения, например в [значениях](#), возвращаемых правилами низкоуровневого обнаружения.

Смотрите также: [Использование прокси в распределенной среде](#)

Запуск прокси

Если установлен из пакета

Zabbix прокси работает как демон. Для запуска прокси выполните:

```
shell> service zabbix-proxy start
```

Эта команда будет работать на большинстве GNU/Linux системах. На других системах вам, возможно, потребуется выполнить:

```
shell> /etc/init.d/zabbix-proxy start
```

Аналогично, для остановки/перезапуска/просмотра состояния, используйте следующие команды:

```
shell> service zabbix-proxy stop
shell> service zabbix-proxy restart
shell> service zabbix-proxy status
```

Запуск вручную

Если выше приведенные команды не работают, вам необходимо запустить прокси вручную. Найдите путь к бинарному файлу `zabbix_proxy` и выполните:

```
shell> zabbix_proxy
```

Следующие параметры командной строки могут быть использованы с Zabbix прокси:

```
-c --config <файл>          путь к файлу конфигурации (по умолчанию
/etc/zabbix/zabbix_proxy.conf)
-R --runtime-control <опция> выполнение административных функций
-h --help                    вывод этого сообщения помощи
-V --version                  вывод номера версии
```

Выполнение административных функций не поддерживается в OpenBSD и NetBSD.

Примеры запуска Zabbix прокси с параметрами командой строки:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf
shell> zabbix_proxy --help
shell> zabbix_proxy -V
```

Управление работой

Опции управления работой:

Опция	Описание	Цель
<code>config_cache_reload</code>	Перезагрузка кэша конфигурации. Игнорируется, если кэш уже загружается в текущий момент времени. Активный Zabbix прокси подключится к Zabbix серверу и запросит данные конфигурации.	
<code>housekeeper_execute</code>	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент.	

Опция	Описание	Цель
log_level_increase[=<цель>]	Увеличение уровня журналирования, действует на все процессы, если цель не указана.	Цель можно указать с помощью: тип процесса - все процессы указанного типа (например, poller) Смотрите весь список типов процессов прокси .
log_level_decrease[=<цель>]	Уменьшение уровня журналирования, действует на все процессы, если цель не указана.	тип процесса,N - тип процесса и номер (например, poller,3) pid - идентификатора процесса (от 1 до 65535). Для больших значений указывайте цель в виде 'тип-процесса,N'.

Пример использования административных функций для перезагрузки кэша конфигурации прокси:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
config_cache_reload
```

Пример использования административных функций для вызова выполнения очистки базы данных:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
housekeeper_execute
```

Примеры использования административных функций по изменению уровня журналирования:

Увеличение уровня журналирования по всем процессам:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_increase
```

Увеличение уровня журналирования у второго процесса поллера:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_increase=poller,2
```

Увеличение уровня журналирования у процесса с PID 1234:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_increase=1234
```

Уменьшение уровня журналирования по всем http поллер процессам:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_decrease="http poller"
```

Пользователь процесса

Zabbix прокси спроектирован для запуска от не привилегированного пользователя (non-root). Он будет работать от любого не привилегированного пользователя от которого был запущен. Таким образом, вы можете запускать прокси от имени любого не привилегированного пользователя, без каких-либо последствий.

Если вы попытаетесь запустить прокси от 'root', он сразу переключится на пользователя 'zabbix', который должен присутствовать в вашей системе. Единственный способ запустить прокси от пользователя 'root' - соответствующим образом отредактировать параметр 'AllowRoot' в конфигурационном файле прокси.

Файл конфигурации

Для получения подробной информации смотрите опции [файла конфигурации](#) по настройке zabbix_proxu.

Типы процессов прокси

- configuration syncer - процесс управления кэшем данных конфигурации в оперативной памяти
- data sender - процесс отправки данных с прокси
- discoverer - процесс обнаружения устройств
- heartbeat sender - процесс уведомления сервера прокси сервером о состоянии прокси
- history syncer - процесс, который записывает историю в БД
- housekeeper - процесс удаления старых данных истории
- http poller - поллер веб-мониторинга
- icmp pinger - поллер проверок icmping
- ipmi manager - менеджер IPMI поллеров
- ipmi poller - поллер для проверок по IPMI
- java poller - поллер для Java проверок
- poller - обычный поллер для пассивных проверок
- self-monitoring - процесс сбора внутренней статистики сервера
- snmp trapper - траппер сбора/обработки SNMP трапов
- task manager - процесс для удаленного выполнения задач, которые запрашиваются другими компонентами (например, возможности закрытия проблемы, подтверждения проблемы, принудительной проверки значения элемента данных, удаленной команды)
- trapper - процесс-улавливатель для активных проверок, трапов, и прокси
- unreachable poller - поллер недоступных устройств
- vmware collector - коллектор данных VMware, ответственный за сбор данных со служб VMware

Можно использовать файл журнала прокси для наблюдения за этими типами процессов.

Различные типы процессов Zabbix прокси можно мониторить используя внутренний [элемент данных zabbix\[процесс,<тип>,<режим>,<состояние>\]](#).

Поддерживаемые платформы

Zabbix прокси запускается на тех же [поддерживаемых платформах](#), что и Zabbix сервер.

Региональные настройки (локаль)

Обратите внимание что сервер требует UTF-8 локаль чтобы некоторые текстовые элементы данных могли быть интерпретированы корректно. Большинство современных Unix-подобных систем имеют локаль UTF-8 установленной по умолчанию, тем не менее, есть некоторые системы где это необходимо задать вручную.

From:

<https://www.zabbix.com/documentation/current/> - **Zabbix Documentation 4.4**

Permanent link:

<https://www.zabbix.com/documentation/current/ru/manual/concepts/proxy>

Last update: **2019/12/27 08:34**

