

Маленькая инструкция к Zabbix

Отсюда: <http://rustedowl.livejournal.com/15791.html>

Итак, Заббикс. Система, по меньшей мере, неочевидная - но спустя некоторое время ее логика становится ясна. Самый главный плюс сервера Zabbix - то что все проверки настраиваются через веб-интерфейс. Редактирование файлов - только для агента, да и там мороки немного.

Полноценной литературы к Zabbix'у - только его вики, но разработчики не делают копию статьи если в ней по сравнению с предыдущей версией поменялось немного, предполагая что у желающих найдется время прочесть документацию ко всем выпускам и листы изменений между версиями. Но, конечно, есть и царский путь: пойти учиться. Стоимость обучения какие-то жалкие 1500\$.

Теперь - самое важное, что следует о нем знать! Перед тем как что-то делать, нужно распланировать это, в голове или на бумаге. На бумаге будет лучше - спустя некоторое время не придется вспоминать чего там как. Заббикс позволяет мониторить все что угодно, но из-за неочевидности многих моментов люди сталкиваются с множеством проблем: тормозами, несрабатыванием триггеров и приходом пятидесяти сообщений в секунду. Это все решается стандартным инструментарием в сочетании с планированием.

Сразу после установки НЕОБХОДИМО удалить все предоставленные для примеров шаблоны, элементы и триггеры. Если их просто взять и применить к машинам - тормоза гарантированы. Они не просто избыточны, они ИЗБЫТОЧНЫ!!!!!!111 Так же можно удалить предустановленные группы, все равно придется создавать свои собственные.

Для начала требуется создать группу узлов сети (Настройка - Группы узлов сети). Это важно - без группы нельзя создать ни узел, ни шаблон. Сама по себе группа никаких свойств не имеет и не назначает, это простое объединение. ВАЖНО: Шаблоны, находящиеся в группе узлов НЕ ПРИМЕНЯЮТСЯ к узлам находящейся в этой же группе. О привязке шаблонов будет позже.

Теперь создаем узел сети (Настройка - Узлы сети). По пунктам

Имя узла сети - некое внутреннее имя, которым Заббикс будет манипулировать внутри себя. Так же применяется для включения активных проверок в агенте Заббикс (о нем тоже позже). Недопустима кириллица или пробелы.

Видимое имя - применяется только в списках узлов и может применяться в оповещениях.

Группы - в каких группах узел находится. Может находиться сразу в нескольких, если это нужно по соображениям безопасности (в Zabbix пользователям можно настраивать доступ к узлам в зависимости от прав)

Интерфейсы агента - Здесь можно указать IP-адрес или DNS-имя узла. Агент устанавливается не обязательно, но что-то указать надо, это будет адрес для проверок по умолчанию. Поддерживается собственный агент Zabbix, протокол SNMP, JMX и IMPI. Если надо - то даже все одновременно.

Описание - все ясно и так.

Наблюдение через прокси - Для разгрузки сервера можно использовать прокси, которые будут

собирать информацию с агентов самостоятельно, а затем передавать ее на сервер. Прокси настраиваются отдельно, и затем их можно выбрать в этом списке.

Активировано - все ясно и так.

Вкладки в этом окне позволяют производить дополнительные настройки, которые я рассматривать не буду.

Нажимаем «Сохранить» и переходим к следующему шагу: созданию элемента данных. Опционально сначала можно создать группу элементов данных, процедура полностью аналогичная группе узлов сети.

Имя - аналогично «имени узла сети».

Тип - какая проверка будет использоваться. Их очень много, но наиболее интересными на начальном этапе являются Zabbix агент, Zabbix агент (активный) и «Простая проверка». Zabbix-агент может быть пассивным и активным. Основное отличие - в способе взаимодействия с сервером. В пассивном режиме (в агенте указан только адрес сервера Zabbix) агент выполняет проверку только после того как сервер его пнет. Схема: сервер → Агент → Сервер → Агент. На это тратятся вычислительные ресурсы сервера (Zabbix создаст специальный процесс, который пинает некоторое количество агентов и собирает полученные с них данные). Агент при этом слушает на порту 10500 (по-умолчанию) и этот порт ему надо открыть. В активном режиме (в агенте указан адрес ServerActive= адрес сервера Zabbix Hostname=ИМЯ УЗЛА СЕТИ как он настроен в Zabbix-сервере) агент сам стучится на сервер, собирает проверки назначенные его узлу, выполняет эти проверки и отправляет на сервер результаты, иногда опрашивая сервер не появились ли новые проверки. Нагрузка на сервер при этом минимальна, так что стоит использовать именно этот режим везде где возможно.

Ключ - собственно, проверка которую может выполнить Zabbix, с целой кучей настроек. Все это можно посмотреть в официальной документации Zabbix, разбирать каждую в отдельности - можно целую книгу написать. Опишу лишь некоторые неочевидные особенности. Параметры заключенные в <> указать можно, но это не обязательно. Бывает полезно в некоторых случаях. В случае с agent.ping он возвращает 1 если проверка успешна и не возвращает ничего если проверка провалена. Триггеры надо писать с учетом этого момента и вместо параметра last использовать nodata.

Интерфейс узла сети - все ясно и так.

Тип информации - Какой тип данных будет храниться в базе. Указывать его обязательно, поскольку в случае неправильного типа элемент данных работать не будет. Узнать об этом получится не сразу, а только спустя некоторое время, которое требуется веб-интерфейсу Zabbix на передачу команд движку, выполнения команды движком и возвращением ошибки веб-интерфейсу. Это может занять изрядное время.

Тип данных - если тип информации «Целое положительное» то предлагает выбрать в какой системе счисления это целое записывать.

Единица измерения - можно оставить пустым, но лучше заполнить. Дело в том, что данные которые Zabbix получает можно посмотреть в разделе Мониторинг-Последние данные, там список из «Имя элемента данных» - «Значение элемента данных». Если указать единицу измерения, она будет отображаться и там. Если измеряется размер, то стоит указывать В (байты), Zabbix сам подставит правильный множитель.

Пользовательский множитель - все ясно и так.

Интервал обновления (в сек) - Частота с которой обновляется содержимое элемента данных. **ОЧЕНЬ КОВАРНАЯ ШТУКА!** Дело в том, что у ключей простых проверок может быть некая длительность запроса, и если она больше интервала обновления (к примеру, обычный ping в случае с Zabbix посылает пять пакетов, раз в секунду. Это настраивается, надо лишь слегка почитать встроенную документацию) то полученные данные будут непредсказуемыми, поскольку заббикс будет слать серии пингов с нескольких дочерних процессов. Так же имеет смысл не опрашивать слишком часто то, что слишком часто не меняется. Скажем, раз в 30 секунд узнавать размер используемого SWAP, свободного места на диске или состояния SMART не обязательно, а в базу эти значения записываться будут.

Переменные интервалы - здесь задается расписание, по которому будет изменяться интервал, если это необходимо.

Период хранения истории (в днях) - Сколько времени в базе будет храниться история изменения этого элемента. Можно ставить 1 день, но нельзя ставить 0 (в этом случае триггеры работать не будут). Слишком долго историю хранить тоже не стоит, база данных все же не резиновая. Этот параметр, в отличие от следующего, хранит **КАЖДОЕ** значение, записанное в базу с частотой интервала обновления.

Период хранения динамики изменений (в днях) - Сколько времени в базе будет храниться **УСРЕДНЕННОЕ ЗА ДЕНЬ** значение элемента. На это требуется гораздо меньше места в базе, а для статистики подходит ничуть не хуже. Впрочем, все зависит от того что мониторится.

Хранение значения - будет ли в базе храниться само значение, или его изменение относительно предыдущего значения. Отображение значения - преобразование значения в другой формат, более приемлемый для чтения человеком. В частности, есть режим отображения как «Zabbix agent ping status». Влияет только на то, как значение будет выглядеть в Мониторинг - Последние данные.

Далее можно добавить узел в группу элементов данных и настроить заполнение инвентарного поля (Zabbix может получать тип ОС, Hostname и IP-адрес, и сразу добавлять это в собственную инвентарную базу).

Описание - все ясно и так.

Нажимаем Добавить и переходим к триггерам.

Имя триггера - аналогично «Видимому имени узла сети». Будет отображаться в уведомлении (если настроить) так что стоит делать его информативным.

Выражение - собственно, условие которое Zabbix должен проверять. Выражение можно создавать непосредственно из интерфейса. В итоге получится что-то вроде:
{внутреннее_имя_хоста:icmpping.max(#10)}= 0 Триггер сработает, если на последние 10 пингов нет ответа.

Многократная генерация событий «ПРОБЛЕМА» - все ясно и так, но я пока ни разу это не использовал.

Описание - все ясно и так.

URL - можно указать адрес с решением этой проблемы в базе знаний.

Важность - В основном нужно для определения какой тип уведомления использовать. Например, Предупреждения слать на e-mail, а Чрезвычайные - как SMS.

Зависимости - та самая вещь, про которую очень многие забывают. Зависимые триггеры не выполняют действий и не шлют оповещений, если триггер от которого они зависят сработал. В итоге вместо 30 писем о недоступности рабочих станций вы получите только одно письмо о том что выключили свет. С зависимостями тем не менее есть один важный момент - зависимые триггеры должны срабатывать МЕДЛЕННЕЕ, чем тот от которого они зависят. Если зависимые триггеры работают быстрее то толку от зависимостей не будет.

Итак, триггеры есть, теперь самое важное - действия. Действия настраиваются в Настройка - Действия

ОБЯЗАТЕЛЬНО удалите действие по умолчанию. Оно шлет набитое малополезной информацией сообщение всем пользователям-админам, всеми доступными способами. Уведомления должны быть краткими и информативными (хотя бы те, которые будут отправляться СМСкой)

Создаем новое действие. Источник события - Триггеры.

Имя - информативное имя, по которому сходу можно понять за что отвечает действие.

Тема по умолчанию - то, что будет прислано в поле «Тема» электронного письма или первой строкой для Jabber и СМСки. Сообщение по умолчанию: здесь можно писать ВСЕ ЧТО УГОДНО! По умолчанию шлется такое количество технической информации, что не сразу понятно что куда и как.

Сообщение о восстановлении - Если триггер вернулся в нормальный режим, прислать сообщение, указанное в полях ниже (они появятся, если поставить галочку).

Вкладка «Условия»

Тип вычисления - здесь можно выбрать логическую формулу, по которой будут вычисляться условия.

Условия - Условия, по которым действие будет определять что выполняться должно именно оно. По умолчанию используется «Состояние обслуживания не в обслуживании» и «Значение триггера = ПРОБЛЕМА». Под это условие попадают ВСЕ триггеры, что чаще всего не нужно. Стоит добавить другое условие, для того чтобы конкретизировать условия.

Вкладка «Операции»

Длительность шага операции по умолчанию - сколько времени будет выполняться между шагами, если это значение в шаге не указано.

Далее добавляется сама операция. Их всего две - «Отправлять сообщение» и «Удаленная команда». В обоих случаях все интуитивно понятно. Разве что при «Отправить сообщение» маленькая строка «Добавить» добавляет шаг операции, а кнопка «Добавить» ниже создает действие.

[zabbix](#), мониторинг

From:

<https://wiki.rtzra.ru/> - RTzRa's hive

Permanent link:

<https://wiki.rtzra.ru/devops/zabbix/zabbix-main-ideas>

Last update: **2021/10/28 15:12**

