Zabbix.
Практическое руководство

Андреа Далле Вакке

[PACKT] PUBLISHING

ДМК издательство

УДК 004.7: 004.451.9Zabbix
ББК 32.972.5
Д15

Далле Вакке А.
Д15 Zabbix. Практическое руководство / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2017. – 356 с.: ил.

ISBN 978-5-97060-462-5

В книге описана система Zabbix – одно из самых популярных решений мониторинга сетей и приложений.

Описана настройка Zabbix, рассмотрены сценарии мониторинга, создание собственных компонент, автоматизация с использованием Zabbix API, а также интеграция Zabbix с внешними системами.

Издание предназначено системным администраторам и архитекторам, желающим интегрировать инфраструктуру Zabbix в свое окружение.

УДК 004.7: 004.451.9Zabbix
ББК 32.972.5

ISBN 978-1-78528-926-2 (анг.)
ISBN 978-5-97060-462-5 (рус.)

Содержание

Глава 1

Развертывание Zabbix

Если вы читаете эту книгу, значит, вы почти наверняка уже устанавливали и использовали Zabbix. Почти наверняка вы пытались использовать эту систему в небольшом или среднем окружении, но с тех пор ситуация изменилась, ваше окружение разрослось, и вы столкнулись с новыми проблемами. В наши дни окружения растут или изменяются очень быстро, и порой довольно сложно оставаться в полной готовности поддерживать надежный мониторинг.

Обычно начальное развертывание системы мониторинга выполняется под руководством какого-либо самоучителя или инструкции, и это распространенная ошибка. Такой подход оправдан для небольших окружений, где время простоя не имеет большого значения, где не приходится беспокоиться о проблемах восстановления сайтов после аварий и, вообще, где все выглядит очень просто.

Почти всегда в таких случаях развертывание и настройка выполняются без учета появления новых элементов, триггеров и событий, которые должны обрабатываться сервером. Если вы уже установили Zabbix и желаете реализовать возможность дальнейшего расширения своего решения мониторинга или, напротив, решили спроектировать и создать новую инфраструктуру мониторинга, эта глава поможет вам.

Данная глава поможет также решить сложную задачу настройки/обновления Zabbix для использования в больших и очень больших окружениях. Она охватывает все аспекты такой задачи, начиная от определения большого окружения до использования Zabbix в роли ресурса с планируемой мощностью. Здесь будут представлены все возможные решения на основе Zabbix, включая практический пример установки с прицелом на обслуживание большого окружения и возможность дальнейшего совершенствования.

К концу этой главы вы узнаете, как действует Zabbix, какие таблицы требуют особого внимания, как рационализировать административные работы в большом окружении, что, как показывает опыт прошлых лет, является действительно очень сложной задачей.

В этой главе рассматриваются следующие темы:

- как определить, что перед вами действительно большое окружение, и какие окружения можно считать большими;
- настройка/обновление Zabbix в большом и очень большом окружении;

- установка Zabbix в трехуровневой системе при наличии готового решения для большого окружения;
- оценка требований к базе данных и определение общего объема хранимых данных;
- знакомство с наиболее тяжелыми таблицами и задачами базы данных;
- оптимизация работ с целью снижения нагрузки на СУБД и повышения эффективности системы в целом;
- основные идеи планирования мощности с учетом возможностей Zabbix.

Определение размера окружения

Основное внимание в этой книге уделяется большим окружениям, поэтому нужно определить, хотя бы приблизительно, какое окружение можно считать большим. Размер определяется разными характеристиками, но в самом простом случае окружение можно назвать большим, если:

- оно распределено географически;
- количество контролируемых устройств исчисляется сотнями или даже тысячами;
- количество проверок, выполняемых каждую секунду, превышает 500;
- имеется большое количество элементов, триггеров и данных для обработки (объем базы данных превышает 100 ГБ);
- доступность и производительность являются критически важными характеристиками.

Все эти пункты характеризуют большое окружение; установка и обслуживание инфраструктуры Zabbix в таком окружении играют важную роль.

Установка является четко определенной задачей, для выполнения которой специально выделяется время, и относится к разряду важнейших, потому что создает основу для мощной и надежной инфраструктуры мониторинга. Кроме того, после создания некоторого задела порой очень непросто что-то передвинуть/переместить без потери данных. Существуют и другие аспекты, которые необходимо учитывать: у нас появится множество задач, связанных с системой мониторинга, большинство из которых придется решать ежедневно, но в больших окружениях они требуют особого внимания.

В небольших окружениях с маленькими базами данных резервное копирование требует минутных усилий, но для большой базы данных решение той же задачи займет намного больше времени.

Планы по восстановлению должны регулярно пересматриваться и тестироваться, чтобы знать, сколько времени потребуется на решение этой задачи в случае аварийных ситуаций.

Помимо повседневного обслуживания, необходимо предусмотреть время на тестирование и ввод в эксплуатацию обновлений, чтобы максимально уменьшить их влияние на решение повседневных задач.

Архитектура Zabbix

Zabbix можно определить как распределенную систему мониторинга с централизованным веб-интерфейсом (с помощью которого осуществляется управление). Из основных особенностей системы хотелось бы особо выделить следующие:

- Zabbix имеет централизованный веб-интерфейс;
- сервер может выполняться под управлением практически любой Unix-подобной операционной системы;
- данная система мониторинга имеет готовых агентов для большинства операционных систем Unix, Unix-подобных и Microsoft Windows;
- система легко интегрируется с другими системами благодаря прикладному интерфейсу, реализованному для множества языков программирования;
- мониторинг может осуществляться по протоколам SNMP (v1, v2 и v3), IPMI, JMX, ODBC, SSH, HTTP(S), TCP/UDP и Telnet;
- данная система мониторинга позволяет нам создавать собственные элементы и графики и интерполировать данные;
- простота настройки.

На рис. 1.1 изображена трехуровневая архитектура Zabbix.

Рис. 1.1 ❖ Трехуровневая архитектура Zabbix

Архитектура Zabbix для большого окружения состоит из трех разных серверов/компонентов (которые также должны настраиваться с учетом высокой доступности):

- веб-сервер;
- сервер баз данных;
- сервер Zabbix.

Полная инфраструктура Zabbix в больших окружениях позволяет вводить в игру еще двух актеров, играющих важную роль. Это агенты Zabbix и прокси-серверы Zabbix. Пример такой инфраструктуры представлен на рис. 1.2.

В этой инфраструктуре имеется централизованный сервер Zabbix, к которому подключено несколько прокси-серверов, обычно по одному на ферму или подсеть.

Сервер Zabbix получает данные от **прокси-серверов Zabbix**, прокси-серверы получают данные от подключенных к ним **агентов Zabbix**, все данные сохраняются в выделенной СУБД, а доступ к этим данным осуществляется посредством веб-интерфейса. Если заглянуть в реализацию системы, можно увидеть, что веб-интерфейс написан на языке PHP, а сервер, прокси-серверы и агенты – на языке C.

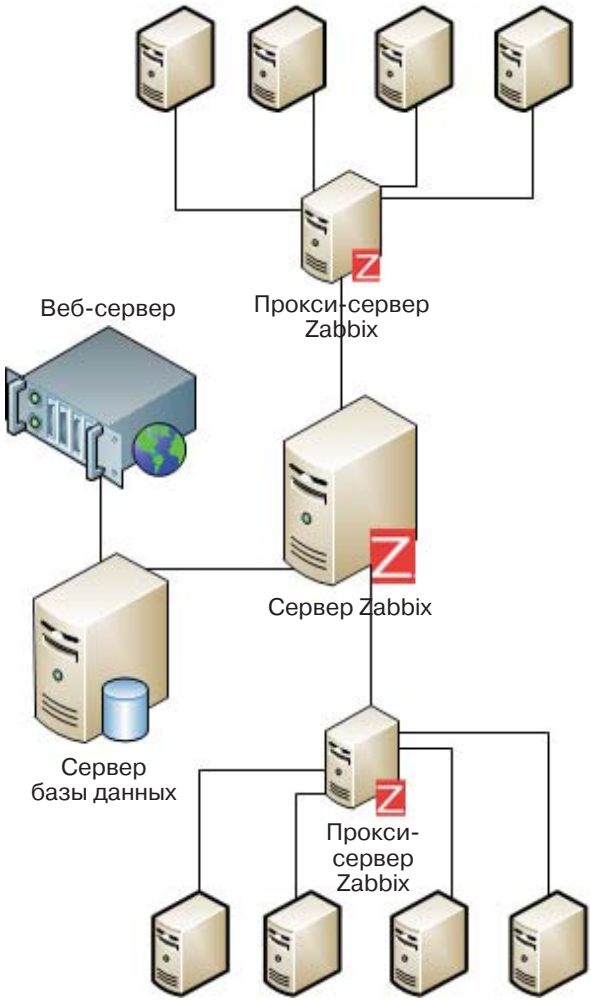

Рис. 1.2 ❖ Два дополнительных компонента инфраструктуры Zabbix

Выбор языка C для реализации сервера, прокси-серверов и агентов обусловлен стремлением обеспечить наивысшую производительность и минимальное потребление системных ресурсов. Все компоненты оптимизированы на достижение максимальной производительности.

Используя прокси-серверы, можно реализовать разные архитектуры. Ниже перечислено несколько таких архитектур в порядке возрастания сложности:

- с единственным сервером;
- с единственным сервером и множеством прокси-серверов;
- распределенная архитектура (доступна только в версиях Zabbix 2.3.0 или выше).

Архитектура с единственным сервером не предполагает использования в большом окружении. Это простейшая архитектура, где мониторинг осуществляет единственный сервер, которая может служить неплохой начальной точкой.

Вероятнее всего, у вас уже имеется установленная система Zabbix. Zabbix отличается высокой гибкостью и позволяет расширить установку с единственным сервером до следующего уровня: мониторинга с применением прокси-серверов.

Мониторинг с применением прокси-серверов реализуется путем настройки одного сервера Zabbix и нескольких прокси-серверов, по одному на ветвь или вычислительный центр. Такая конфигурация проста в обслуживании и обладает преимуществом решений централизованного мониторинга. Она обеспечивает хороший баланс между сложностью реализации и мониторингом большого окружения. Архитектуру с единственным сервером и множеством прокси-серверов, изображенную на рис. 1.2, можно (приложив немало усилий) расширить до распределенной архитектуры.

Начиная с версии 2.4.0 система Zabbix не поддерживает сценариев распределенного мониторинга узлов. И действительно, если загрузить исходный код версии Zabbix, обсуждаемой в книге, а затем код версии Zabbix 2.4.3, можно заметить, что ветвь кода, управлявшая узлами, была удалена.

Все возможные архитектуры Zabbix подробно обсуждаются в *главе 2 «Распределенный мониторинг»*.

Установка Zabbix

Конфигурация, обсуждаемая в этой главе, предусматривает создание отдельного сервера для каждого из следующих основных компонентов:

- веб-интерфейс;
- сервер Zabbix;
- база данных Zabbix.

Мы опишем эту конфигурацию потому, что:

- она может служить основой для дальнейшего расширения за счет добавления прокси-серверов и узлов;
- каждый компонент выполняется на выделенном сервере;
- подобные конфигурации являются отправной точкой для организации мониторинга больших окружений;
- она имеет большое распространение;
- она почти наверняка станет для вас отправной точкой, пригодной для дальнейшего расширения и совершенствования инфраструктуры мониторинга.

Фактически эта первая установка пригодится всем, кто собирается расширить существующую инфраструктуру. Если имеющееся у вас решение для мониторинга реализовано как-то иначе, вам стоит запланировать переход на архитектуру с тремя выделенными серверами.

Если после организации трехуровневой системы производительность все еще оставляет желать лучшего, можно запланировать и обдумать конфигурацию, лучше подходящую для ваших условий.

Осуществляя мониторинг большого окружения, следует:

- использовать выделенные серверы, чтобы упростить дальнейшее расширение;
- реализовать конфигурацию с высокой доступностью;
- реализовать конфигурацию с высокой отказоустойчивостью.

В трехуровневой системе нагрузка на центральный процессор серверного компонента не имеет большого значения, по крайней мере для сервера Zabbix. Потребляемая вычислительная мощность напрямую зависит от количества хранимых элементов и частоты обновления (количества проверок в минуту), а не от объема памяти.

В действительности сервер Zabbix не особенно требователен к производительности центрального процессора, но весьма взыскателен к объему памяти. По опыту, четырехъядерного процессора с 8 ГБ памяти вполне достаточно для мониторинга более чем 1000 хостов.

Существуют два основных способа установки Zabbix:

- загрузить последнюю версию исходного кода и скомпилировать его;
- установить из предварительно скомпилированного пакета.

Существует еще один путь: загрузить образ виртуальной машины с сервером Zabbix, настроить его и запустить, но мы не будем рассматривать этот способ, потому что полный контроль и знание необходимых шагов намного лучше. Кроме того, главный недостаток такого решения – в том, что эти образы (доступны по адресу: http://www.zabbix.com/ru/download.php), по утверждению самой компании Zabbix, непригодны для промышленной эксплуатации.

Установка из предварительно скомпилированных пакетов имеет следующие преимущества:

- упрощает процедуру обновления;
- автоматически удовлетворяет зависимости.

Установка из исходных кодов также имеет свои преимущества:

- возможность компиляции только необходимых составляющих;
- возможность статической сборки агента и установки в разных версиях Linux;
- возможность полного контроля над обновлениями.

Большие окружения обычно включают компьютеры, работающие под управлением разных версий Linux, Unix и Microsoft Windows. Такие сценарии весьма типичны в гетерогенных инфраструктурах, и если необходимо установить агента Zabbix на каждый Linux-сервер, придется иметь дело с разными версиями агентов и конфигурационными файлами, размещенными в разных местах.

Чем выше уровень стандартизации на серверах, тем проще обслуживать и обновлять инфраструктуру. Флаг компиляции `--enable-static` дает возможность стандартизовать агента для разных версий и выпусков Linux, и это является большим преимуществом. Агента, скомпилированного статически, легко можно развернуть где угодно и использовать один и тот же конфигурационный файл, хранящийся в одном месте. Процедуру развертывания также можно стандарти-

зовать; единственное, что может отличаться, – это сценарий запуска/остановки и порядок регистрации на требуемом уровне запуска.

Та же идея применима к коммерческим версиям Unix – того же самого агента можно устанавливать на разные версии Unix, выпускаемые одним поставщиком.

Предварительные требования

Перед компиляцией Zabbix необходимо ознакомиться с предварительными требованиями. Для поддержки веб-интерфейса нужно установить следующее программное обеспечение:

- Apache (1.3.12 или выше);
- PHP (5.3.0 или выше).

Для сборки сервера Zabbix необходимы:

- СУБД: можно использовать открытые альтернативы, такие как PostgreSQL и MySQL;
- пакет `zlib-devel`;
- пакет `mysql-devel`: реализует поддержку MySQL (не требуется в нашем примере);
- пакет `postgresql-devel`: реализует поддержку PostgreSQL;
- пакет `glibc-devel`;
- пакет `curl-devel`: используется для поддержки веб-мониторинга;
- пакет `libidn-devel`: требуется для пакета `curl-devel`;
- пакет `openssl-devel`: требуется для пакета `curl-devel`;
- пакет `net-snmp-devel`: реализует поддержку SNMP;
- пакет `popt-devel`: может требоваться пакету `net-snmp-devel`;
- пакет `rpm-devel`: может требоваться пакету `net-snmp-devel`;
- пакет `OpenIPMI-devel`: реализует поддержку IPMI;
- пакет `iksemel-devel`: реализует поддержку протокола Jabber;
- пакет `Libssh2-devel`;
- пакет `sqlite3`: необходим для поддержки SQLite, используется как промежуточная база данных Zabbix (обычно на прокси-серверах).

Установить все зависимости в Red Hat Enterprise Linux можно с помощью `yum` (с привилегиями root), но сначала нужно подключить репозиторий EPEL следующей командой:

```
# yum install epel-release
```

Выполните команду `yum install`, чтобы установить все необходимые пакеты:

```
# yum install zlib-devel postgresql-devel glibc-devel curl-devel gcc automake postgresql
libidn-devel openssl-devel net-snmp-devel rpm-devel OpenIPMI-devel iksemel-devel libssh2-
devel openldap-devel
```

Пакет `iksemel-devel` используется для отправки Jabber-сообщений. Это действительно очень полезная возможность, так как позволяет серверу Zabbix посылать сообщения в чат. Кроме того, Jabber поддерживается в Zabbix как один из типов средств оповещения, для которых есть возможность настроить рабочие часы, чтобы избежать получения сообщений, когда вы находитесь не на работе.

Настройка сервера

Для нормальной работы сервера Zabbix требуется создать и настроить непривилегированную учетную запись. В любом случае, если демон запускается с привилегиями root, он автоматически переключается на работу с правами учетной записи Zabbix, если она имеется:

```
# groupadd zabbix
# useradd -m -s /bin/bash -g zabbix zabbix
# useradd -m -s /bin/bash -g zabbix zabbixsvr
```

Примечание. Никогда не запускайте сервер с привилегиями пользователя root, потому что это увеличивает риск плачевных последствий в случае взлома.

Предыдущие строки позволяют гарантировать повышенный уровень безопасности. Сервер и агент должны выполняться с разными учетными записями; в противном случае агент получит возможность доступа к конфигурации сервера Zabbix. Теперь, используя учетную запись Zabbix, можно загрузить и извлечь исходные тексты из файла `tar.gz`:

```
# wget http://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/2.4.4/zabbix-
2.4.4.tar.gz/download -O zabbix-2.4.4.tar.gz

# tar -zxvf zabbix-2.4.4.tar.gz
```

Теперь займемся подготовкой исходных текстов к компиляции. Кстати, утилита настройки имеет справочный раздел:

```
# cd zabbix-2.4.3

# ./configure --help
```

В данном примере исходные тексты сервера мы настроим, указав следующие параметры:

```
# ./configure --enable-server --enable-agent --with-postgresql -with-libcurl --with-jabber
--with-net-snmp --enable-ipv6 --with-openipmi --with-ssh2 --with-ldap
```

Команды `zabbix_get` и `zabbix_send` генерируются, только если при настройке исходных текстов был указан флаг `--enable-agent`.

Если подготовка к компиляции завершилась без ошибок, на экране должны появиться примерно такие строки:

```
config.status: executing depfiles commands

Configuration:

  Detected OS:           linux-gnu
  Install path:          /usr/local
  Compilation arch:      linux

  Compiler:               gcc
  Compiler flags:        -g -O2 -I/usr/include -I/usr/include/rpm
-I/usr/local/include -I/usr/lib64/perl5/CORE -I. -I/usr/include -I/usr/
```

```
include -I/usr/include -I/usr/include

  Enable server:          yes
  Server details:
    With database:          PostgreSQL
    WEB Monitoring:         cURL
    Native Jabber:          yes
    SNMP:                   yes
    IPMI:                   yes
    SSH:                    yes
    ODBC:                   no
    Linker flags:           -rdynamic -L/usr/lib64 -L/usr/lib64
-L/usr/lib -L/usr/lib -L/usr/lib
    Libraries: -lm -ldl -lrt -lresolv -lpq -liksemel
-lnetsnmp -lssh2 -lOpenIPMI -lOpenIPMIposix -lldap -llber -lcurl

  Enable proxy:           no

  Enable agent:           yes

  Agent details:
    Linker flags:           -rdynamic -L/usr/lib
    Libraries:              -lm -ldl -lrt -lresolv -lldap -llber -lcurl

  Enable Java gateway:    no
  LDAP support:           yes
  IPv6 support:           yes

************************************************************
*                  Now run 'make install'                  *
*                                                          *
*                Thank you for using Zabbix!               *
*                  <http://www.zabbix.com>                 *
************************************************************
```

Мы пока запустим только команду `make`, без команды `make install`. Чтобы определить другой каталог для установки сервера Zabbix, используйте флаг `--prefix` команды `configure`, например: `--prefix=/opt/zabbix`. Теперь следуйте инструкциям в разделе «Установка и создание пакета».

Настройка агента

Чтобы подготовить исходные тексты агента к компиляции, выполните следующую команду:

```
# ./configure --enable-agent

# make
```

Если передать утилите `configure` ключ `--enable-static`, команда `make` выполнит статическую компоновку агента с библиотеками, благодаря чему выполняемому файлу для запуска не нужны будут внешние библиотеки; этот прием может пригодиться для подготовки агента, способного выполняться в разных версиях Linux.

Установка и создание пакета

В двух предыдущих разделах мы закончили компиляцию исходных текстов и теперь готовы выполнить установку; для чего достаточно запустить следующую команду:

```
# make install
```

Но я рекомендую не торопиться с установкой, а воспользоваться программой `checkinstall`. Она создаст пакет Zabbix и установит его.

Загрузить программу можно по адресу ftp://ftp.pbone.net/mirror/ftp5.gwdg.de/pub/opensuse/repositories/home:/ikoinoba/CentOS_CentOS-6/x86_64/checkinstall-1.6.2-3.el6.1.x86_64.rpm.

Обратите внимание, что `checkinstall` – лишь один из возможных способов создания пакета, пригодного к распространению.

Можно также воспользоваться предварительно собранной программой `checkinstall`. Текущей, на момент написания этих строк, была версия `checkinstall-1.6.2-20.4.i686.rpm` (в Red Hat/CentOS); пакет также требует установки `rpm-build`, которую можно выполнить командой (с привилегиями root):

```
# yum install rpm-build rpmdevtools
```

После этого требуется создать необходимые каталоги:

```
# mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
```

Наличие пакета многое упрощает; его легче распространять и обновлять, плюс можно создавать пакеты для разных видов диспетчеров пакетов: `rpm`, `deb` и `tgz`.

Программа `checkinstall` способна создавать пакеты для Debian (ключ `-D`), Slackware (ключ `-S`) и Red Hat (ключ `-R`). Это особенно удобно для создания пакета с агентом Zabbix (статически скомпонованным) и установки его на серверы в окружении.

Теперь необходимо приобрести привилегии пользователя root или воспользоваться командой `sudo`, чтобы создать пакет:

```
# checkinstall --nodoc -R --install=no -y
```

Если процедура создания прошла без проблем, должно появиться примерно такое сообщение:

```
**********************************************************************
Done. The new package has been saved to
/root/rpmbuild/RPMS/x86_64/zabbix-2.4.4-1.x86_64.rpm
You can install it in your system anytime using:
     rpm -i zabbix-2.4.4-1.x86_64.rpm
**********************************************************************
```

После этого выполните (с привилегиями root) следующую команду, чтобы установить пакет:

```
# rpm -i zabbix-2.4.4-1.x86_64.rpm
```

Теперь система Zabbix установлена. Двоичные файлы сервера установлены в каталог <prefix>/sbin, утилиты – в каталог <prefix>/bin, а man-страницы (страницы справочного руководства) – в каталог <prefix>/share.

Установка из пакетов

Для полноты обзора всех возможных методов установки рассмотрим шаги, которые требуется выполнить, чтобы установить Zabbix из предварительно собранных rpm-пакетов.

Сначала необходимо подключить репозиторий:

```
# rpm -ivh http://repo.zabbix.com/zabbix/2.4/rhel/6/x86_64/zabbix-2.4.4-1.el6.x86_64.rpm
```

Эта команда создаст файл /etc/yum.repos.d/zabbix.repo с параметрами репозитория и подключит его.

Заглянув в репозиторий Zabbix, можно увидеть, что внутри «неподдерживаемого» дерева http://repo.zabbix.com/non-supported/rhel/6/x86_64/ доступны следующие пакеты: iksemel, fping, libssh2 и snmptt.

Теперь, чтобы установить сервер Zabbix и веб-интерфейс, достаточно выполнить команду:

```
# yum install zabbix-server-pgsql
```

А на веб-сервере (не забыв подключить репозиторий):

```
# yum install zabbix-web-pgsql
```

Чтобы установить агента, требуется выполнить только одну команду:

```
# yum install zabbix-agent
```

Если вы решите использовать пакеты RPM, имейте в виду, что конфигурационные файлы в этом случае сохраняются в каталог /etc/zabbix/. Но на протяжении всей книги будет предполагаться, что эти файлы находятся в каталоге /usr/local/etc/.

Кроме того, если в месте развертывания агента Zabbix имеется действующий локальный брандмауэр, вам потребуется настроить правила iptables, чтобы обеспечить беспрепятственное прохождение трафика через порт агента Zabbix:

```
# iptables -I INPUT 1 -p tcp --dport 10050 -j ACCEPT
# iptables-save
```

Настройка сервера

Для настройки сервера нам потребуется проверить и отредактировать единственный файл:

```
/usr/local/etc/zabbix_server.conf
```

Конфигурационные файлы находятся в каталоге:

```
/usr/local/etc
```

В файле /usr/local/etc/zabbix_server.conf нужно изменить имя пользователя, пароль и имя базы данных; обратите внимание, что настройка базы данных будет выполнена ниже в этой главе, а пока просто укажите планируемые имя пользователя/пароль/имя базы данных. Итак, действуя с привилегиями учетной записи zabbix, откройте файл для редактирования:

```
# vi /usr/local/etc/zabbix_server.conf
```

и измените следующие параметры:

```
DBHost=localhost

DBName=zabbix

DBUser=zabbix

DBPassword=<укажите-здесь-свой-пароль>
```

Теперь сервер Zabbix настроен и практически готов к запуску. Местоположение файла zabbix_server.conf определяется переменной времени компиляции sysconfdir. Не забудьте ограничить доступ к конфигурационному файлу, выполнив следующую команду:

```
chmod 600/usr/local/etc/zabbix_server.conf
```

Для внешних сценариев отводится каталог:

```
/usr/local/share/zabbix/externalscripts
```

Имя этого каталога определяется переменной времени компиляции datadir.

Для сценариев оповещений отводится каталог:

```
/usr/local/share/zabbix/alertscripts
```

Его так же можно настроить во время компиляции, определив значение переменной времени компиляции datadir.

Теперь перейдем к настройке агента. В конфигурационном файле агента нужно указать IP-адрес сервера Zabbix. После этого добавьте две службы в соответствующие уровни запуска, чтобы обеспечить их активизацию в требуемый момент времени.

Для выполнения этой задачи нужно установить сценарии запуска/остановки:

- /etc/init.d/zabbix-agent;
- /etc/init.d/zabbix-proxy;
- etc/init.d/zabbix-server.

Эти сценарии находятся в каталоге misc:

```
zabbix-2.4.4/misc/init.d
```

В их числе – сценарии запуска для разных дистрибутивов Linux. Но это дерево каталогов не поддерживается и не тестируется, и в нем могут находиться устаревшие сценарии, не подходящие для современных версий Linux, поэтому желательно просмотреть их и протестировать перед установкой.

После добавления сценариев запуска/остановки в каталог /etc/init.d нужно включить их в список служб:

```
# chkconfig --add zabbix-server
# chkconfig --add zabbix-agentd
```

Теперь осталось лишь сообщить системе, в какие уровни запуска они должны быть помещены; в этой книге мы будем использовать уровни 3 и 5:

```
# chkconfig --level 35 zabbix-server on
# chkconfig --level 35 zabbix-agentd on
```

Также, если на компьютере с сервером Zabbix имеется действующий брандмауэр, необходимо настроить iptables, чтобы обеспечить беспрепятственное прохождение трафика через порт сервера Zabbix, для чего нужно выполнить следующую команду (с привилегиями root):

```
# iptables -I INPUT 1 -p tcp --dport 10051 -j ACCEPT
# iptables-save
```

Прямо сейчас мы не можем запустить сервер, потому что еще не настроена база данных.

Установка базы данных

После установки сервера Zabbix можно заняться установкой и настройкой сервера баз данных. Все шаги, описываемые ниже, выполняются на выделенном сервере. Прежде всего нужно установить сервер PostgreSQL. Проще всего для этой цели использовать пакет, входящий в состав дистрибутива, но я рекомендую использовать последнюю стабильную версию 9.x.

В Red Hat (RHEL 6.4) все еще используется версия PostgreSQL 8.x. Эта же версия по наследству применяется в CentOS и ScientificLinux. Версия PosgreSQL 9.x имеет множество преимуществ; на момент написания этих строк последней стабильной и готовой к промышленной эксплуатации была версия 9.2.

Чтобы установить версию PostgreSQL 9.4, выполните следующие шаги:

1. Найдите файлы .repo:
   - **Red Hat**: /etc/yum/pluginconf.d/rhnplugin.conf [main];
   - **CentOS**: /etc/yum.repos.d/CentOS-Base.repo, [base] и [updates].
2. Добавьте следующую строку в конец разделов, обозначенных выше:

   ```
   exclude=postgresql*
   ```

3. Откройте в браузере страницу http://yum.postgresql.org и найдите соответствующую версию пакета RPM. Например, чтобы установить PostgreSQL 9.4 в RHEL 6, выберите пакет http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-redhat94-9.4-1.noarch.rpm.
4. Подключите репозиторий командой: yum localinstall http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-centos94-9.4-1.noarch.rpm.
5. Выведите список пакетов в репозитории postgresql командой:

   ```
   # yum list postgres*
   ```

6. Отыщите требуемый пакет и установите его командой:

```
# yum install postgresql94 postgresql94-server postgresql94-contrib
```

7. После установки пакета необходимо инициализировать базу данных:

```
# service postgresql-9.4 initdb
```

Инициализацию можно выполнить также командой:

```
# /etc/init.d/postgresql-9.4 initdb
```

8. Теперь займемся настройкой параметров в файле `/var/lib/pgsql/9.4/data/postgresql.conf`. Нужно определить адрес и номер порта для приема запросов:

```
listen_addresses = '*'
port = 5432
```

Также нужно добавить пару строк для настройки `zabbix_db`, сразу за следующими строками:

```
# TYPE DATABASE     USER     ADDRESS          METHOD

# "local" is for Unix domain socket connections only
local   all         all                      trust
in /var/lib/pgsql/9.4/data/pg_hba.conf

# настройки для Zabbix
local   zabbix_db   zabbix                   md5
host    zabbix_db   zabbix   <CIDR-address>   md5
```

Ключевое слово `local` соответствует всем соединениям, выполняющимся с помощью сокетов из домена Unix (Unix-domain sockets). За ним следуют имя базы данных (`zabbix_db`), имя пользователя (`zabbix`) и метод аутентификации (в данном случае `md5`).
Ключевое слово `host` соответствует всем соединениям, выполняющимся по протоколу TCP/IP (включая SSL). За ним следуют имя базы данных (`zabbix_db`), имя пользователя (`zabbix`), маска сети (определяет хосты, которым разрешено подключаться к базе данных) и метод аутентификации (в данном случае `md5`).

9. Маска сети, определяющая хосты, которым разрешено подключаться к базе данных, в данном случае должна быть обычной маской, потому что нам требуется открыть доступ к базе данных для веб-интерфейса (то есть веб-сервера) и сервера Zabbix, которые выполняются на других компьютерах. Примером такой маски может служить `10.6.0.0/24` (небольшая подсеть). Вероятнее всего, веб-интерфейс и сервер Zabbix будут находиться в другой сети, поэтому перечислите здесь все необходимые сети и маски.
10. В заключение запустите сервер PosgreSQL командой:

```
# service postgresql-9.4 start
```

или:

```
# /etc/init.d/postgresql-9.4 start
```

Чтобы создать базу данных, нужно зарегистрироваться с учетной записью пользователя `postgres` (или пользователя в вашем дистрибутиве, который наделен полномочиями управления базами данных PostgreSQL). Создайте пользователя в базе данных (это наш пользователь `zabbix`) и подключитесь под именем этого пользователя к серверу баз данных, чтобы импортировать схему данных.

Ниже следуют команды, необходимые для импортирования:

```
# su - postgres
```

После регистрации с учетной записью пользователя `postgres` можно создать базу данных (в данном случае базу данных с именем `zabbix_db`):

```
-bash-4.1$ psql

postgres=# CREATE USER zabbix WITH PASSWORD '<ПАРОЛЬ-ПОЛЬЗОВАТЕЛЯ-БАЗЫ-ДАННЫХ>';

CREATE ROLE

postgres=# CREATE DATABASE zabbix_db WITH OWNER zabbix ENCODING='UTF8';

CREATE DATABASE

postgres=# \q
```

Сценарии создания базы данных находятся в каталоге `/database/postgresql`, куда извлекались файлы с исходным кодом. Их нужно установить в точности в следующем порядке:

```
# cat schema.sql | psql -h <IP-АДРЕС-СЕРВЕРА-БАЗ-ДАННЫХ> -W -U zabbix zabbix_db
# cat images.sql | psql -h <IP-АДРЕС-СЕРВЕРА-БАЗ-ДАННЫХ> -W -U zabbix zabbix_db
# cat data.sql | psql -h <IP-АДРЕС-СЕРВЕРА-БАЗ-ДАННЫХ> -W -U zabbix zabbix_db
```

Параметр командной строки `-h <IP-АДРЕС-СЕРВЕРА-БАЗ-ДАННЫХ>` команды `psql` помогает обойти параметр `local` в стандартном конфигурационном файле `/var/lib/pgsql/9.4/data/pg_hba.conf`.

Теперь можно запустить сервер Zabbix и проверить связку сервер/база данных:

```
# /etc/init.d/zabbix-server start

Starting Zabbix server:                                    [ OK ]
```

Заглянув в файл журнала, можно получить более полную информацию о происходящем на сервере. Там должны находиться следующие строки (по умолчанию файл журнала размещается в `/tmp/zabbix_server.log`):

```
26284:20150114:034537.722 Starting Zabbix Server. Zabbix 2.4.4 (revision 51175).
26284:20150114:034537.722 ****** Enabled features ******
26284:20150114:034537.722 SNMP monitoring: YES
26284:20150114:034537.722 IPMI monitoring: YES
26284:20150114:034537.722 WEB monitoring: YES
26284:20150114:034537.722 VMware monitoring: YES
26284:20150114:034537.722 Jabber notifications: YES
26284:20150114:034537.722 Ez Texting notifications: YES
```

```
26284:20150114:034537.722 ODBC: YES
26284:20150114:034537.722 SSH2 support: YES
26284:20150114:034537.722 IPv6 support: YES
26284:20150114:034537.725 ******************************
26284:20150114:034537.725 using configuration file: /usr/local/etc/zabbix/zabbix_server.conf
26284:20150114:034537.745 current database version (mandatory/optional): 02040000/02040000
26284:20150114:034537.745 required mandatory version: 02040000
26284:20150114:034537.763 server #0 started [main process]
26289:20150114:034537.763 server #1 started [configuration syncer #1]
26290:20150114:034537.764 server #2 started [db watchdog #1]
26291:20150114:034537.764 server #3 started [poller #1]
26293:20150114:034537.765 server #4 started [poller #2]
26294:20150114:034537.766 server #5 started [poller #3]
26296:20150114:034537.770 server #7 started [poller #5]
26295:20150114:034537.773 server #6 started [poller #4]
26297:20150114:034537.773 server #8 started [unreachable poller #1]
26298:20150114:034537.779 server #9 started [trapper #1]
26300:20150114:034537.782 server #11 started [trapper #3]
26302:20150114:034537.784 server #13 started [trapper #5]
26301:20150114:034537.786 server #12 started [trapper #4]
26299:20150114:034537.786 server #10 started [trapper #2]
26303:20150114:034537.794 server #14 started [icmp pinger #1]
26305:20150114:034537.790 server #15 started [alerter #1]
26312:20150114:034537.822 server #18 started [http poller #1]
26311:20150114:034537.811 server #17 started [timer #1]
26310:20150114:034537.812 server #16 started [housekeeper #1]
26315:20150114:034537.829 server #20 started [history syncer #1]
26316:20150114:034537.844 server #21 started [history syncer #2]
26319:20150114:034537.847 server #22 started [history syncer #3]
26321:20150114:034537.852 server #24 started [escalator #1]
26320:20150114:034537.849 server #23 started [history syncer #4]
26326:20150114:034537.868 server #26 started [self-monitoring #1]
26325:20150114:034537.866 server #25 started [proxy poller #1]
26314:20150114:034537.997 server #19 started [discoverer #1]
```

Вообще говоря, место хранения по умолчанию файла журнала выбрано не самое лучшее, потому что каталог `/tmp` автоматически очищается при каждой перезагрузке, а сами файлы не архивируются.

Место по умолчанию можно изменить в конфигурационном файле `/etc/zabbix_server.conf`. Достаточно просто изменить параметр:

```
### Option: LogFile
LogFile=/var/log/zabbix/zabbix_server.log
```

Не забудьте после этого создать структуру каталогов, выполнив следующую команду с привилегиями root:

```
# mkdir -p /var/log/zabbix
# chown zabbixsvr:zabbixsvr /var/log/zabbix
```

Еще один важный аспект – настройка автоматической ротации журнала в logrotate. Это легко реализуется простым добавлением конфигурационного файла в каталог /etc/logrotate.d/.

Создайте файл, выполнив следующую команду с привилегиями root:

```
# vim /etc/logrotate.d/zabbix-server
```

и добавьте в него следующие строки:

```
/var/log/zabbix/zabbix_server.log {
        missingok
        monthly
        notifempty
        compress
        create 0664 zabbix zabbix
}
```

После этого перезапустите сервер Zabbix следующей командой (с привилегиями root):

```
# /etc/init.d/zabbix-server restart
Shutting down Zabbix server:                    [ OK ]
Starting Zabbix server:                         [ OK ]
```

Кроме того, проверьте, запущен ли сервер с привилегиями соответствующего пользователя:

```
# ps aux | grep "[z]abbix_server"
502 28742 1  0 13:39 ?    00:00:00 /usr/local/sbin/zabbix_server
502 28744 28742 0 13:39 ? 00:00:00 /usr/local/sbin/zabbix_server
502 28745 28742 0 13:39 ? 00:00:00 /usr/local/sbin/zabbix_server
...
```

Вывод команды показывает, что zabbix_server выполняется с привилегиями пользователя 502. Сделаем еще шаг и проверим, соответствует ли идентификатор 502 пользователю, созданному нами выше:

```
# getent passwd 502
zabbixsvr:x:502:501::/home/zabbixsvr:/bin/bash
```

Как следует из полученной строки, все замечательно.

Иногда в журнале можно увидеть строку с ошибкой:

```
28487:20130609:133341.529 Database is down. Reconnecting in 10 seconds.
```

Эта проблема может быть вызвана несколькими причинами:

- брандмауэр (локальный, на нашем сервере, или находящийся в сетевой инфраструктуре) не пропускает сетевых пакетов;
- допущены ошибки в настройках сервера PostgreSQL;
- допущены ошибки в файле zabbix_server.conf.