

Содержание

Об авторе	11
Благодарности	12
О технических обозревателях	13
Предисловие	15
Глава 1. Развертывание Zabbix	22
Определение размера окружения.....	23
Архитектура Zabbix.....	24
Установка Zabbix.....	26
Предварительные требования	28
Настройка сервера.....	29
Настройка агента.....	30
Установка и создание пакета	31
Установка из пакетов.....	32
Настройка сервера	32
Установка базы данных.....	34
Подготовка базы данных	43
Оценка размера базы данных	45
Очистка истории.....	47
Веб-интерфейс	54
Мастер настройки – настройка веб-интерфейса.....	54
Планирование мощностей с помощью Zabbix.....	59
Эффект наблюдателя	59
Выбор параметров для мониторинга.....	59
Определение базовой оценки.....	61
Нагрузочное тестирование.....	61
Прогнозирование тенденций.....	63
В заключение	64
Глава 2. Распределенный мониторинг	66
Прокси-серверы Zabbix.....	67
Развертывание прокси-сервера Zabbix	68
Команды управления прокси-сервером Zabbix во время выполнения	71
Развертывание прокси-сервера Zabbix из RPM-пакета	72
Использование других баз данных с прокси-серверами	76
Движение данных мониторинга в системе Zabbix	77
Движение данных мониторинга через прокси-серверы	78

Мониторинг прокси-серверов Zabbix.....	80
Вопросы безопасности.....	82
Отказ от конфигурации сети.....	83
Изолирование сети.....	84
Простые туннели.....	85
Протокол SSH.....	85
Программа stunnel.....	86
Использование полноценной VPN.....	87
В заключение.....	88

Глава 3. Высокая доступность и отказоустойчивость 89

Высокая доступность.....	89
Уровни обслуживания.....	90
Некоторые вопросы высокой доступности.....	92
Автоматизация аварийного переключения с применением диспетчера ресурсов.....	93
Репликация файловой системы с помощью DRBD.....	93
Реализация высокой доступности для веб-сервера.....	94
Настройка HTTPD.....	95
Расemaker и механизм STONITH.....	97
Расemaker – так ли необходим кворум?.....	98
Расemaker – идея закрепления ресурсов.....	98
Расemaker – настройка Apache/HTTPD.....	99
Реализация высокой доступности для сервера Zabbix.....	101
Реализация высокой доступности для базы данных.....	103
Кластеризация PostgreSQL.....	105
Зеркалирование логического тома с помощью LVM и DRBD.....	106
Обязательные условия использования DRBD на LVM.....	107
Создание устройства DRBD поверх раздела LVM.....	107
Включение ресурсов в DRBD.....	108
Определение первичного устройства DRBD.....	110
Создание файловой системы на устройстве DRBD.....	110
Кластеры Расemaker – интеграция с DRBD.....	111
Настройка включения DRBD.....	112
Расemaker – настройка LVM.....	112
Расemaker – настройка PostgreSQL.....	113
Расemaker – настройка сети.....	113
Расemaker – заключительные настройки.....	114
Настройка кластера – заключительная проверка.....	114
Производительность и оптимизация DRBD.....	115
Эффективная синхронизация DRBD.....	116
Включение онлайн-верификации для DRBD.....	117
DRBD – некоторые аспекты настройки сети.....	118
В заключение.....	120

Глава 4. Сбор данных	121
Сбор простых данных	121
Потоки данных и элементы	123
Ловушки элементов Zabbix	126
Потоки данных	126
Мониторинг базы данных с помощью Zabbix	127
ODBC	127
Установка драйверов баз данных	128
Драйвер MySQL ODBC	128
Драйвер PostgreSQL ODBC	130
Драйвер Oracle ODBC	131
Конфигурационные файлы unixODBC	132
Компиляция Zabbix с поддержкой ODBC	133
Элементы мониторинга базы данных	134
Некоторые замечания о запросах ODBC SQL	135
Мониторинг через JMX	136
Защищенность JMX	137
Установка шлюза Zabbix Java gateway	138
Настройка JMX в Zabbix	140
Ключи JMX	140
Некоторые замечания о JMX	142
Мониторинг через SNMP	143
Запросы SNMP	146
Ловушки SNMP	148
Демон snmptrapd	148
Обработка ловушек в сценарии на Perl	149
Мониторинг через SSH	153
Настройка SSH-аутентификации с ключом	154
Мониторинг через IPMI	156
Первые шаги с IPMI	156
Настройка учетных записей IPMI	157
Настройка элементов IPMI в Zabbix	159
Мониторинг веб-страниц	161
Аутентификация для мониторинга веб-страниц	162
Завершение сеанса	166
Агрегированные и вычисляемые элементы	168
Агрегированные элементы	169
Вычисляемые элементы	171
В заключение	172
Глава 5. Визуализация данных	173
Графики	174
Простые графики	174
Ситуационные графики	177
Особенности ситуационных графиков	178

Нестандартные графики.....	179
Обзор всех параметров настройки графиков	184
Визуализация данных с применением карт.....	187
Создание первой карты в Zabbix	190
Важные замечания о макросах и адресах URL.....	193
Внутри карты.....	195
Выбор элементов.....	197
Использование макросов в картах	198
Комплексные экраны.....	200
Создание экрана	200
Динамические элементы	202
Слайд-шоу	204
Проблема управления слайдами на большом мониторе.....	205
Замечания о слайдах для больших мониторов.....	205
Автоматизация слайд-шоу.....	206
Информация об уровне обслуживания.....	207
Настройка предоставления информации об уровне обслуживания	208
В заключение	211

Глава 6. Управление оповещениями 212

Выражения триггеров.....	212
Выбор элементов и функций.....	213
Выбор между интервалом времени и количеством замеров	214
Функции определения даты и времени.....	215
Важность триггера	216
Выбор между абсолютными и относительными значениями.....	216
Операции как способ связывания.....	217
Управление зависимостями триггеров.....	220
Выполнение действий	221
Определение действия.....	222
{EVENT.DATE} и {EVENT.TIME}	223
{INVENTORY.SERIALNO.A} и подобные макросы	223
Определение условий.....	224
Выбор операций	226
Шаги и эскалация.....	226
Сообщения и способы оповещения	228
Удаленные команды	229
В заключение	230

Глава 7. Управление шаблонами 231

Создание шаблонов.....	231
Добавление сущностей в шаблон	232
Использование макросов	233
Пользовательские макросы	238
Импортирование и экспортирование шаблонов.....	239

Присоединение шаблонов к хостам.....	239
Вложенные шаблоны.....	241
Комбинирование шаблонов	242
Обнаружение хостов.....	242
Автоматическая регистрация активных агентов	245
Настройка автоматической регистрации	246
Практический пример.....	247
Низкоуровневое обнаружение	248
В заключение	254
Глава 8. Внешние сценарии	256
Внешние проверки.....	257
Местоположение сценария	257
Особенности работы внешних проверок.....	258
Реализация сценария.....	261
Основные правила создания сценариев.....	262
Дополнительные замечания о внешних проверках.....	263
Параметр UserParameter.....	263
Гибкость параметра UserParameter.....	264
Замечания по использованию параметра UserParameter	265
Отправка данных с помощью zabbix_sender	266
Новый сценарий.....	267
Сценарий-обертка для вызова check_ora_sendtrap	268
Достоинства и недостатки выделенного сервера для внешних сценариев.....	269
Протоколы Zabbix.....	270
Протокол Zabbix get.....	270
Протокол Zabbix sender	271
Интересная недокументированная особенность.....	272
Свойство clock в объектах JSON.....	273
Протокол Zabbix agent	274
Еще некоторые варианты ответов	276
Протокол низкоуровневого обнаружения.....	276
Взаимодействие с Zabbix.....	280
Реализация протокола Zabbix sender на Java.....	280
Реализация протокола Zabbix sender на Python	282
Некоторые замечания о разработке агента.....	283
В заключение	284
Глава 9. Расширение Zabbix	286
Zabbix API.....	286
Первые шаги	288
Аутентификация	289
Использование библиотеки PyZabbix.....	291
Исследование Zabbix API с помощью JQuery	294
Массовые операции.....	297

Перераспределение хостов между прокси-серверами	297
Добавление и изменение учетных записей	298
Экспортирование данных	301
Извлечение табличных данных	302
Создание графиков на основе данных.....	304
Пакет программ Graphviz	305
Создание графа зависимостей триггеров.....	306
Создание карт Zabbix на основе файлов с описанием	308
В заключение	314
Глава 10. Интеграция с Zabbix	315
Интеграция с WhatsApp.....	316
Подготовка к отправке сообщений.....	317
Регистрация клиента yowsup.....	318
Отправка первого сообщения в WhatsApp.....	319
Настройка безопасности клиента yowsup.....	319
Создание первой группы в Zabbix для рассылки оповещений.....	322
Интеграция yowsup с Zabbix.....	326
Обзор системы Request Tracker	331
Настройка RT для интеграции с Zabbix.....	333
Создание отдельной очереди для Zabbix.....	334
Настройка заявок – раздел «Ссылки»	335
Настройка заявок – приоритет заявки.....	335
Настройка заявок – собственные поля	336
Соединение с Request Tracker API	338
Настройка Zabbix для интеграции с Request Tracker.....	341
Создание заявок RT из событий Zabbix.....	344
В заключение	348
Предметный указатель	349

Об авторе

Андреа Далле Вакке (Andrea Dalle Vacche) – высококвалифицированный профессионал с более чем 15-летним опытом работы в ИТ-индустрии.

Закончил университет в городе Феррара, Италия (Univeristà degli Studi di Ferrara) по курсу «Информационные технологии». Это образование послужило фундаментом, на который Андреа опирается в своих изысканиях с тех пор. Имеет сертификаты многих уважаемых крупных игроков в ИТ-индустрии, в том числе Cisco, Oracle, ITIL и, конечно, Zabbix. Также имеет сертификат «Red Hat Certified Engineer». На протяжении всей своей карьеры работал со многими масштабными вычислительными окружениями, часто на ролях, требующих обширных знаний. Это еще больше повысило его квалификацию, расширило круг практических навыков и укрепило стремление к применению полученных знаний на практике.

Любовь к Zabbix проснулась в Андреа, когда он занимался администрированием баз данных Oracle и разработкой приложений, использующих их. Основное время он тратил на снижение «затрат на эксплуатацию», специализируясь на мониторинге и автоматизации. Именно тогда он столкнулся с Zabbix и по достоинству оценил техническую гибкость инструмента и простоту администрирования с его применением. Используя новые знания в качестве стартовой площадки, Андреа вдохновился идеей создать Orabbix, первый комплект открытого программного обеспечения для мониторинга Oracle, который был бы полностью совместим с Zabbix. Он опубликовал несколько статей о программном обеспечении, связанном с Zabbix, таком как DBforBIX. Ознакомиться с его проектами можно на персональном веб-сайте автора: <http://www.smartmarmot.com>.

В настоящее время Андреа работает старшим архитектором в ведущем инвестиционном банке, в весьма разнородном вычислительном окружении. Он отвечает за очень широкий круг задач, сталкивается со многими критическими аспектами платформ Unix/Linux и вынужден особое внимание уделять разнородному стороннему программному обеспечению, имеющему стратегическую важность для развития банка.

Андреа также играет важную роль в группе обеспечения безопасности банка, занимаясь такими направлениями, как защищенность, сохранение тайны, стандартизация, аудит, удовлетворение требований регулятора и решения поддержки безопасности.

Кроме этой книги, он также написал:

- «Mastering Zabbix», Packt Publishing;
- «Zabbix Network Monitoring Essentials», Packt Publishing.

Благодарности

В первую очередь я хочу поблагодарить мою жену Анну (Anna) за ее поддержку и понимание. Она не раз оказывала мне помощь и давала ценные советы. Большое спасибо Фифи (Fifi) за умиротворяющее мурлыканье и пушистый покой.

Особое спасибо я хочу сказать всему коллективу издательства Packt Publishing и Адриану (Adrian) в частности. Их советы, поправки и предложения были по-настоящему ценными для меня. Весь коллектив проявил высокий профессионализм.

О технических обозревателях

Григорий Чернышев (Grigory Chernyshev) – старший инженер по выпуску/инженер по организации взаимодействий (senior release manager/DevOps engineer) в отделе онлайн-игр компании Mail.Ru Group. Специализируется на управлении конфигурациями, автоматизации процесса сборки, мониторинге, выпуске версий и создании сценариев на языке Python. Имеет опыт работы в таких проектах, как Allods Online и Skyforge – массовых ролевых игр AAA-класса, получивших известность по всему миру. В своей повседневной работе он использует Zabbix для мониторинга внутренних игровых серверов, гетерогенных агентов сборки и множества инфраструктурных серверов.

Помимо этого, он пишет плагины для систем Atlassian Jira и JetBrains Teamcity – в случае с последней даже победил на конкурсе WordPress Plugins в 2015 году!

Я хочу сказать спасибо моей жене за терпение, моим родителям за счастливое детство и координатору проекта, Санчите (Sanchita), за ее неиссякаемый энтузиазм и поддержку.

Нитиш Кумар (Nitish Kumar) – ведущий специалист по платформе Wintel в компании HT Media Ltd. и независимый технический блогер, занимающийся популяризацией самых разных технологий. Вот уже восемь лет занимается разными технологиями от Microsoft и открытыми решениями (включая, но не ограничиваясь: Spiceworks, продукты ManageEngine, Zabbix, MS Active Directory, MS Exchange Servers и др.), из которых два последних года посвятил рентабельным решениям корпоративного уровня с целью уменьшить сложность их требований и обеспечить более рациональное использование рабочего времени обслуживающего их персонала. Нитиш с большим энтузиазмом участвует в различных корпоративных событиях и общественных вебинарах. Особый интерес он испытывает к мобильным технологиям и часто пишет статьи о различных устройствах и технологиях. Имеет степень магистра информационных технологий, полученную в институте прикладной физики и технологий в Аллахабаде (Индия), и в область его интересов входят технологии от Microsoft, открытое программное обеспечение и мобильные устройства. Его блог находится по адресу: <http://nitishkumar.net>, желающие могут написать ему на электронный почтовый ящик: nitish@nitishkumar.net.

Нитиш является соавтором книги «Getting Started with Spiceworks», Packt Publishing. Также участвовал как технический обозреватель в подготовке других книг о Zabbix и Spiceworks.

Николас Пьер (Nicholas Pier) – сетевой инженер. Занимается веб-разработкой, проектированием сетевой инфраструктуры вычислительных центров на основе виртуализации и решений SAN. Пишет промежуточное программное обеспечение для бизнес-приложений. На данный момент Николас имеет множество промышленных сертификатов, включая Cisco CCNP, VMware VCP-DCV и множество других сертификатов от компаний Cisco и CompTIA. В свободное время увлекается пивоварением, бегом на длинные дистанции и чтением книг.

Тимоти Скоппетта (Timothy Scoppetta) – системный администратор. Специализируется на автоматизации, непрерывной интеграции и создании отказоустойчивых инфраструктур. Работал в Google и множестве начинающих компаний. В настоящее время занимается преподаванием ультрасовременных инструментов и эффективных приемов в институте.

Предисловие

С самого первого выпуска, состоявшегося в 2001 году, система Zabbix зарекомендовала себя как очень мощное и эффективное решение для мониторинга. Это открытый продукт, поэтому его легко получить и развернуть, а уникальный подход к мониторингу и отправке предупреждений позволяет на равных конкурировать с другими решениями, как открытыми, так и коммерческими. Это очень мощный и компактный пакет с очень низкими требованиями к аппаратуре и поддерживаемому программному обеспечению. Если к перечисленному добавить еще простоту в использовании, становится очевидно, что Zabbix отлично подходит для мониторинга небольших окружений с ограниченным бюджетом. Но когда дело доходит до управления мониторингом большого количества объектов со сложными настройками и зависимостями, масштабируемость и распределенная архитектура Zabbix предстают в полном своем блеске. Как никакой другой продукт, Zabbix идеально подходит для больших и сложных распределенных окружений, позволяя эффективно управлять и извлекать полезную информацию из объектов мониторинга и событий, что особенно важно, если не важнее, чем решение обычных проблем доступности и простоты использования.

Это – второе издание книги, первое было написано в соавторстве с Андреа Далле Вакке (Andrea Dalle Vacche) и Стефано Кеван Ли (Stefano Kewan Lee).

Цель этой книги – помочь вам получить максимум от системы Zabbix и наладить эффективный мониторинг больших и сложных окружений.

О чем рассказывается в книге

Глава 1 «Развертывание Zabbix» описывает оптимальный выбор аппаратного и программного обеспечения для сервера Zabbix и базы данных с учетом текущей вычислительной инфраструктуры, целей мониторинга и возможного расширения в будущем. Эта глава включает также раздел с интереснейшим обсуждением размеров базы данных, который может пригодиться для оценки окончательного ее объема для стандартного окружения. Здесь также охватываются вопросы правильного определения размеров окружения и кратко обсуждаются измеряемые показатели, что также может пригодиться для планирования мощностей. Глава содержит практические примеры и теоретические расчеты, чтобы читатель мог получить навыки, необходимые для развертывания в действующем окружении.

Глава 2 «Распределенный мониторинг» исследует различные компоненты Zabbix, действующие на стороне сервера и клиента (агента). На одних и тех же примерах сетей будут даны различные распределенные решения, а также описаны их достоинства и недостатки. В дополнение к развертыванию и настройке аген-

тов здесь описываются настройки прокси-серверов, а также рассматриваются вопросы обслуживания и управления изменениями. В этом разделе охватываются все возможные архитектурные реализации Zabbix, а также положительные и отрицательные стороны.

Глава 3 «Высокая доступность и отказоустойчивость» охватывает вопросы высокой доступности и отказоустойчивости. Здесь вы научитесь выбирать параметры высокой доступности для каждого из трех основных уровней Zabbix. Обсуждение основывается на информации, представленной в двух предыдущих главах. Первая часть книги завершается несколькими сценариями развертывания, включающими высокодоступные серверы и базы данных, организованные в иерархические уровни и распределенные архитектуры, пригодные для мониторинга тысяч географически распределенных объектов. Эта глава включает практический пример и описание нескольких возможных сценариев.

Глава 4 «Сбор данных» выходит за рамки использования простых агентов и SNMP-запросов и затрагивает некоторые более сложные источники данных. В ней исследуются мощные встроенные функции Zabbix, порядок их использования и выбор параметров для мониторинга, чтобы обеспечить полный контроль без чрезмерного увеличения нагрузки на систему. Здесь также исследуются вопросы агрегирования значений и их использование в мониторинге сложных окружений с кластерами или еще более сложными грид-архитектурами (grid architectures).

Глава 5 «Визуализация данных» рассказывает о мощных возможностях визуализации данных в Zabbix. Она будет особенно полезна тем, кому требуется выяснить или обосновать необходимость расширения/обновления аппаратных средств. Здесь вы узнаете, как на основе данных мониторинга создавать динамические карты, организовать коллекции графиков для визуализации на больших экранах в центрах управления и реализовать общее качественное представление. Эта глава охватывает вопросы качественной визуализации результатов мониторинга, которая поможет своевременно выявлять проблемы и предупреждать их. Здесь также исследуются некоторые эффективные приемы использования отчетов о качестве обслуживания (Service Level Agreement, SLA), поддерживаемые системой Zabbix.

Глава 6 «Управление оповещениями» приводит примеры сложных триггеров и условий срабатывания, а также рекомендации по выбору правильного количества триггеров и оповещений. Ее цель – помочь выдержать баланс, чтобы не оставить незамеченными возможные проблемы и не вызвать появления большого числа ложных срабатываний. В этой главе вы также узнаете, как использовать действия для автоматического исправления простых проблем, активировать действия без участия человека с целью согласования разных триггеров и событий и внедрить их процесс управления. Кроме того, здесь вы узнаете, какие операции можно автоматизировать, чтобы уменьшить нагрузку на администраторов и оптимизировать процесс администрирования, дополнив его возможностью опережения событий.

Глава 7 «Управление шаблонами» содержит рекомендации по эффективному управлению шаблонами: конструирование сложных шаблонов из простых ком-

понентов, управление эффектами, вызванными изменениями в шаблонах, поддержка существующих объектов мониторинга и связывание шаблонов с вновь обнаруженными узлами сети. Эта глава завершает вторую часть книги, посвященную различным средствам мониторинга и управления данными, имеющимися в Zabbix. В третьей, заключительной части книги обсуждаются возможности интеграции Zabbix со сторонними продуктами и мощные возможности расширения системы.

Глава 8 «Внешние сценарии» рассказывает, как писать сценарии для мониторинга объектов, которые не поддерживаются ядром Zabbix. Описывает преимущества и недостатки хранения сценариев на стороне сервера или агента, как запускать или планировать их выполнение, и подробно анализирует протокол агентов Zabbix. Здесь вы узнаете обо всех возможных побочных эффектах, задержках и нагрузке, вызванных сценариями; научитесь реализовывать все необходимые проверки, зная все, что связано с ними. Эта глава включает примеры различных сценариев на Bash, Java и Python, опираясь на которые, вы легко сможете написать свои сценарии, расширяющие возможности мониторинга Zabbix.

Глава 9 «Расширение Zabbix» углубляется в прикладной интерфейс Zabbix и особенности его использования для создания специализированных интерфейсных элементов и сложных расширений. Она охватывает также вопросы выборки результатов мониторинга для дальнейшего исследования и составления отчетов. Включает простые примеры на Python реализации экспортирования и дальнейшей обработки данных, выполнения массовых и сложных операций, связанных с мониторингом объектов, и, наконец, автоматизации различных аспектов управления, таких как создание и настройка учетных записей пользователей, их активация и т. п.

Глава 10 «Интеграция с Zabbix» завершает книгу обсуждением вопросов интеграции Zabbix с другими системами. Интеграция имеет большое значение для успешного управления любыми большими и сложными окружениями. Здесь вы узнаете, как использовать встроенные особенности Zabbix, обращаться к прикладному интерфейсу или напрямую к базе данных для обмена информацией с различными выше- и нижестоящими системами и приложениями. Познакомитесь с конкретными примерами организации взаимодействий с приложениями инвентаризации, системами паспортизации отказов и системами хранения данных.

Кому адресована эта книга

Как следует из названия книги – «Zabbix. Практическое руководство. Второе издание», вы не найдете здесь подробных, пошаговых инструкций (за исключением, может быть, описания процедуры установки, которая будет описана с самого начала) по основам использования Zabbix. Несмотря на то что в книге приводится масса подробной информации по установке сервера или настройке элементов, триггеров и экранов, в ней предполагается, что вы уже имеете некоторое знакомство с особенностями работы системы и готовы сосредоточить внимание на более про-

двинутых аспектах. Даже если прежде вам не приходилось использовать Zabbix, вы все равно сможете почерпнуть немало ценного из этой книги, но в этом случае я настоятельно рекомендую обратиться к официальной документации Zabbix, доступной по адресу: <https://www.zabbix.com/documentation/2.4/ru/manual>, чтобы восполнить любые пробелы в ваших знаниях.

Что потребуется для работы с книгой

Прежде чем углубиться в настройки Zabbix, хочется отметить, что конфигурация, предлагаемая и обсуждаемая здесь, протестирована в крупном промышленном окружении (начитывающем более 1800 сетевых узлов, 89 500 элементов мониторинга и 30 000 триггеров), достаточно представительном для многих больших и очень больших окружений. Решения по поддержке высокой доступности, демонстрирующиеся в этой книге, являются не просто умозрительными рекомендациями, но были проверены на практике в ходе случившейся аварии (когда сетевые кабели были повреждены во время земляных работ).

Надо понимать, что большинство вариантов выбора из представленных в этой книге было сделано и проверено на практике. Одним из важнейших примеров может служить выбор PostgreSQL в качестве официальной СУБД для Zabbix. Система управления базами данных PostgreSQL – достаточно зрелая для промышленного использования и обладает очень богатыми функциональными возможностями:

- горячее резервирование поддерживается изначально;
- полноценная поддержка требований ACID (Atomicity, Consistency, Isolation, Durability – Атомарность, Согласованность, Изолированность, Долговечность) к транзакционной системе;
- множество различных встроенных способов организации резервных баз данных (горячее резервирование, синхронная репликация и т. д.);
- эффективное секционирование.

База данных является критически важным компонентом для Zabbix, особенно когда требуется хранить исторические данные и гарантировать постоянную производительность с ростом объема базы данных.

В этой книге мы будем исходить из нескольких предположений: в качестве системы управления пакетами используется `yum`, а операционная система установлена из дистрибутива Red Hat Enterprise Linux. Но в любом случае, кроме конкретных названий пакетов и диспетчера управления пакетами, информация в книге остается действительной для любых дистрибутивов Linux. Кроме того, обсуждаемые архитектуры и их реализации не связаны с каким-то определенным дистрибутивом. Мы не будем использовать оригинальную поддержку кластеров в Red Hat, точно так же мы не будем принимать решения, которые нельзя было бы воплотить в любом другом дистрибутиве Linux.

В книге часто встречаются упоминания различных открытых программных продуктов, но из всех них вам желательно иметь знакомство со следующими:

- Apache: <http://www.apache.org/>;
- Pacemaker: <http://clusterlabs.org/>;
- PostgreSQL: <http://www.postgresql.org/>;
- DRBD: <http://www.drbd.org>.

В этой книге также предполагается, что вы имеете некоторые навыки системного администрирования и программирования. Мы будем время от времени давать задания для самостоятельной реализации программного кода. Имея перед глазами предлагаемые примеры с подробным описанием, вы наверняка справитесь с созданием собственных плагинов или внешних программ, хорошо интегрирующихся с Zabbix. Примеры программного кода в книге написаны на двух широко распространенных языках: Java и Python. Они знакомы большинству современных программистов, а после знакомства с особенностями реализации протокола Zabbix вы без труда сможете переключаться между ними.

Zabbix – это не просто программный продукт для мониторинга; это открытое решение мониторинга, удовлетворяющее самые широкие потребности, и эта книга познакомит вас со всеми достоинствами и недостатками возможных решений.

А теперь пришло время вступить в мир Zabbix!

Соглашения

В этой книге вы обнаружите несколько стилей оформления текста, которые разделяют различные виды информации. Ниже приводятся примеры этих стилей и поясняется их значение.

Элементы программного кода в тексте, имена таблиц в базах данных, имена папок и файлов, расширения файлов, пути к каталогам в файловой системе, фиктивные адреса URL, ввод пользователя и учетные записи в Twitter оформляются так: «Большинство из этих параметров определяется в файле `php.ini`».

Блоки кода оформляются следующим образом:

```
zabbixsrv=zabbixsvr
[ -e /etc/sysconfig/$sysconf ] && . /etc/sysconfig/$sysconf
start()
{
    echo -n $"Starting Zabbix server: "
```

Когда потребуется привлечь ваше внимание к определенному фрагменту в блоке программного кода, он будет выделяться жирным:

```
; Максимальный объем данных в запросах POST,
; которые может обрабатывать PHP.
; http://www.php.net/manual/en/ini.core.php#ini.post-max-size
post_max_size = 16M
```

Ввод или вывод в командной строке будет оформляться так:

```
# yum list postgres*
```

Новые термины и **важные** слова будут выделены жирным. Текст, отображаемый на экране, например в меню или в диалогах, будет оформляться так: «Закончив заполнение формы, щелкните на кнопке **Next**».



Так оформляются предупреждения и важные примечания.



Так оформляются советы и рекомендации.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф в разделе «Читателям – Файлы к книгам».

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии данной книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты автор-

ских прав и лицензирования. Если вы столкнетесь в Интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли принять меры.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Вопросы

Вы можете присылать любые вопросы, касающиеся данной книги, по адресу dmkpress@gmail.com или questions@packtpub.com. Мы постараемся разрешить возникшие проблемы.

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.

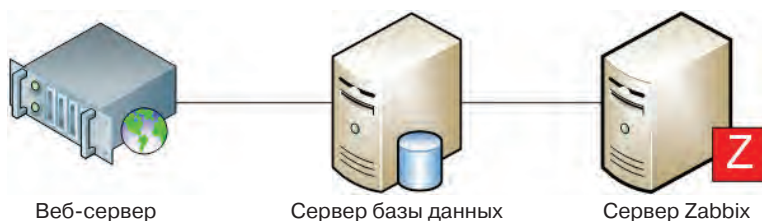


Рис. 1.1 ❖ Трехуровневая архитектура Zabbix

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 С.

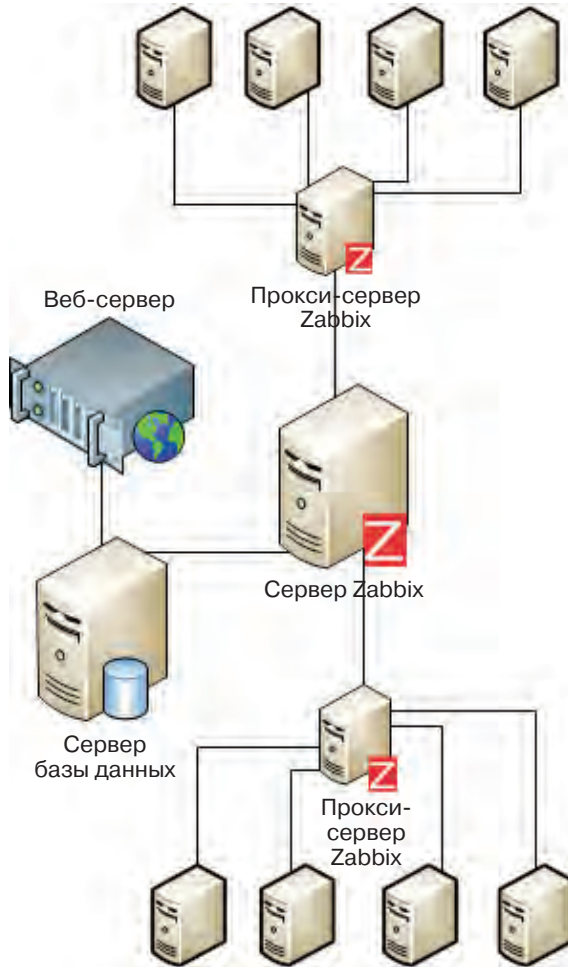


Рис. 1.2 ❖ Два дополнительных компонента инфраструктуры Zabbix



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 `popc-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
-I/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`, `ping`, `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 PostgreSQL 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 PostgreSQL командой:

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

или:

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