**ИНСТРУКЦИЯ ПО УСТАНОВКЕ**

**RT.WAREHOUSE**

2021

ОГЛАВЛЕНИЕ

[ОБЩИЕ СВЕДЕНИЯ 6](#_Toc74742066)

[1 ТРЕБОВАНИЯ К RT.WAREHOUSE 7](#_Toc74742067)

[1.1 Требования к операционной системе 7](#_Toc74742068)

[1.2 Требования к программному обеспечению 7](#_Toc74742069)

[1.3 Требования к JDK 9](#_Toc74742070)

[1.4 Требования к оборудованию и сети 9](#_Toc74742071)

[1.5 Требования к хранилищу 10](#_Toc74742072)

[1.6 Распределения Hadoop 10](#_Toc74742073)

[2 АРХИТЕКТУРА RT.WAREHOUSE 12](#_Toc74742074)

[2.1 Мастер 13](#_Toc74742075)

[2.1.1 Основной мастер 13](#_Toc74742076)

[2.1.2 Резервный мастер 13](#_Toc74742077)

[2.2 Сегменты 14](#_Toc74742078)

[2.2.1 Основные сегменты 14](#_Toc74742079)

[2.2.2 Резервные сегменты 14](#_Toc74742080)

[2.2.3 Отказоустойчивость сегментов и восстановление 15](#_Toc74742081)

[2.2.4 Пример аппаратного стека хоста сегмента 16](#_Toc74742082)

[2.2.5 Пример схемы сегментного диска 17](#_Toc74742083)

[2.3 Интерконнект 17](#_Toc74742084)

[2.3.1 Резервный интерконнект 18](#_Toc74742085)

[2.3.2 Конфигурация сетевого интерфейса 18](#_Toc74742086)

[2.3.3 Конфигурация коммутатора 19](#_Toc74742087)

[2.4 Хосты ETL для загрузки данных 20](#_Toc74742088)

[3 ОЦЕНКА ЁМКОСТИ ХРАНИЛИЩА 22](#_Toc74742089)

[3.1 Расчет полезной ёмкости диска 22](#_Toc74742090)

[3.2 Расчёт размера пользовательских данных 23](#_Toc74742091)

[3.3 Расчёт требований к пространству для метаданных и журналов 23](#_Toc74742092)

[4 НАСТРОЙКА ВАШЕЙ СИСТЕМЫ 25](#_Toc74742093)

[4.1 Отключение программного обеспечения SELinux и файрвола 25](#_Toc74742094)

[4.2 Рекомендуемые настройки параметров ОС 27](#_Toc74742095)

[4.2.1 Файл hosts 27](#_Toc74742096)

[4.2.2 Файл sysctl.conf 28](#_Toc74742097)

[4.2.3 Ограничения системных ресурсов 30](#_Toc74742098)

[4.2.4 Параметры монтирования XFS 31](#_Toc74742099)

[4.2.5 Настройки дискового ввода-вывода 31](#_Toc74742100)

[4.2.6 Transparent Huge Pages (THP) 32](#_Toc74742101)

[4.2.7 Удаление объекта IPC 33](#_Toc74742102)

[4.2.8 Лимит подключений по SSH 34](#_Toc74742103)

[4.3 Синхронизация системных часов 35](#_Toc74742104)

[4.3.1 Настройка NTP 35](#_Toc74742105)

[4.4 Создание пользователя с правами администратора RT.Warehouse 36](#_Toc74742106)

[5 УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ RT.WAREHOUSE 39](#_Toc74742107)

[5.1 Установка RT.Warehouse 39](#_Toc74742108)

[5.2 Включение SSH без пароля 40](#_Toc74742109)

[5.3 Подтверждение вашей установки 41](#_Toc74742110)

[5.4 Об установке базы данных RT.Warehouse 42](#_Toc74742111)

[6 СОЗДАНИЕ ОБЛАСТЕЙ ХРАНЕНИЯ ДАННЫХ 43](#_Toc74742112)

[6.1 Создание областей хранения данных на хостах мастера и резервного мастера 43](#_Toc74742113)

[6.1.1 Создание расположения каталога данных на мастере 43](#_Toc74742114)

[6.2 Создание областей хранения данных на хостах сегментов 43](#_Toc74742115)

[6.2.1 Создание расположения каталогов данных на всех хостах сегмента 44](#_Toc74742116)

[7 ПРОВЕРКА ВАШИХ СИСТЕМ 45](#_Toc74742117)

[7.1 Проверка производительности сети 45](#_Toc74742118)

[7.2 Проверка дискового ввода-вывода и пропускной способности памяти 46](#_Toc74742119)

[8 ИНИЦИАЛИЗАЦИЯ СИСТЕМЫ RT.WAREHOUSE 48](#_Toc74742120)

[8.1 Обзор 48](#_Toc74742121)

[8.2 Инициализация RT.Warehouse 48](#_Toc74742122)

[8.2.1 Создание файла хоста инициализации 49](#_Toc74742123)

[8.2.2 Создание файла конфигурации RT.Warehouse 50](#_Toc74742124)

[8.2.3 Запуск утилиты инициализации 51](#_Toc74742125)

[8.2.4 Использование скрипта возврата 52](#_Toc74742126)

[8.2.5 Установка часового пояса RT.Warehouse 52](#_Toc74742127)

[8.3 Установка переменных сред RT.Warehouse 53](#_Toc74742128)

[8.4 Следующие шаги 54](#_Toc74742129)

[8.4.1 Разрешение клиентских подключений 54](#_Toc74742130)

[8.4.2 Создание баз данных и загрузка данных 54](#_Toc74742131)

[9 УСТАНОВКА ДОПОЛНИТЕЛЬНЫХ ПОСТАВЛЯЕМЫХ МОДУЛЕЙ 55](#_Toc74742132)

[10 НАСТРОЙКА ЧАСОВОГО ПОЯСА И ПАРАМЕТРОВ ЛОКАЛИЗАЦИИ 56](#_Toc74742133)

[10.1 Настройка часового пояса 56](#_Toc74742134)

[10.2 Поддержка локализации в RT.Warehouse 56](#_Toc74742135)

[10.2.1 Влияние локализации 59](#_Toc74742136)

[10.2.2 Устранение неполадок в регионах 59](#_Toc74742137)

[10.3 Поддержка набора символов 59](#_Toc74742138)

[10.4 Установка набора символов 62](#_Toc74742139)

[10.5 Преобразование набора символов между сервером и клиентом 63](#_Toc74742140)

[11 ОБНОВЛЕНИЕ БОЛЕЕ РАННЕЙ ВЕРСИИ RT.WAREHOUSE 6 66](#_Toc74742141)

[11.1 Предварительные условия 66](#_Toc74742142)

[11.2 Обновление 6.x до более новой версии 6.x 67](#_Toc74742143)

[12 [ОПЦИОНАЛЬНО] ВКЛЮЧЕНИЕ IPTABLES 69](#_Toc74742144)

[12.1 Включение iptables 69](#_Toc74742145)

[12.2 Пример правил iptables 70](#_Toc74742146)

[12.2.1 Пример правил iptables для мастера и резервного мастера 71](#_Toc74742147)

[12.2.2 Пример правил iptables хоста сегмента 72](#_Toc74742148)

[13 УТИЛИТЫ УПРАВЛЕНИЯ УСТАНОВКОЙ 74](#_Toc74742149)

[14 ПЕРЕМЕННЫЕ СРЕДЫ RT.WAREHOUSE 75](#_Toc74742150)

[14.1 Обязательные переменные среды 75](#_Toc74742151)

[14.1.1 GPHOME 75](#_Toc74742152)

[14.1.2 PATH 75](#_Toc74742153)

[14.1.3 LD\_LIBRARY\_PATH 75](#_Toc74742154)

[14.1.4 MASTER\_DATA\_DIRECTORY 75](#_Toc74742155)

[14.2 Необязательные переменные среды 75](#_Toc74742156)

[14.2.1 PGAPPNAME 76](#_Toc74742157)

[14.2.2 PGDATABASE 76](#_Toc74742158)

[14.2.3 PGHOST 76](#_Toc74742159)

[14.2.4 PGHOSTADDR 76](#_Toc74742160)

[14.2.5 PGPASSWORD 76](#_Toc74742161)

[14.2.6 PGPASSFILE 76](#_Toc74742162)

[14.2.7 PGOPTIONS 76](#_Toc74742163)

[14.2.8 PGPORT 77](#_Toc74742164)

[14.2.9 PGUSER 77](#_Toc74742165)

[14.2.10 PGDATESTYLE 77](#_Toc74742166)

[14.2.11 PGTZ 77](#_Toc74742167)

[14.2.12 PGCLIENTENCODING 77](#_Toc74742168)

[15 ПРИМЕР ANSIBLE PLAYBOOK 78](#_Toc74742169)

[15.1 Ansible Playbook — установка RT.Warehouse для CentOS 7 79](#_Toc74742170)

ОБЩИЕ СВЕДЕНИЯ

Информация об установке, настройке и обновлении программного обеспечения RT.Warehouse и настройке хост-компьютеров RT.Warehouse.

ТРЕБОВАНИЯ К RT.WAREHOUSE

В этом разделе описаны требования к платформе RT.Warehouse и программному обеспечению операционной системы.

Требования к операционной системе

RT.Warehouse работает на следующих платформах операционных систем:

CentOS 64-бит 7.x.

CentOS 64-бит 6.x.

Ubuntu 18.04 LTS.

РЕД ОС 7.x

Сервер RT.Warehouse поддерживает TLS версии 1.2 в системах РЕД ОС и TLS версии 1.3 в системах Ubuntu.

Требования к программному обеспечению

RT.Warehouse требует следующих программных пакетов в системах РЕД ОС, которые устанавливаются автоматически в качестве зависимостей при установке RPM-пакета RT.Warehouse:

apr;

apr-util;

bash;

bzip2;

curl;

krb5;

libcurl;

libevent;

libxml2;

libyaml;

zlib;

openldap;

openssh;

openssl;

openssl-libs (RHEL7/Centos7);

perl;

readline;

rsync;

R;

sed (used by gpinitsystem);

tar;

zip.

В системах Ubuntu для RT.Warehouse требуются следующие программные пакеты, которые устанавливаются автоматически в качестве зависимостей при установке RT.Warehouse с установщиком пакетов Debian:

libapr1;

libaprutil1;

bash;

bzip2;

krb5-multidev;

libcurl3-gnutls;

libcurl4;

libevent-2.1-6;

libxml2;

libyaml-0-2;

zlib1g;

libldap-2.4-2;

openssh-client;

openssh-client;

openssl;

perl;

readline;

rsync;

sed;

tar;

zip;

net-tools;

less;

proute2.

RT.Warehouse использует Python 2.7.12, который включён в установку продукта (и не устанавливается как зависимость пакета).

|  |
| --- |
| **Внимание.**  SSL можно использовать только на хосте мастера системы RT.Warehouse. SSL нельзя использовать на хостах сегментов системы. |

|  |
| --- |
| **Внимание.**  Для всех хостов системы RT.Warehouse необходимо отключить SELinux. Вам также следует отключить программное обеспечение файрвола, хотя программное обеспечение файрвола можно включить, если это требуется в целях безопасности. См. Отключение программного обеспечения SELinux и файрвола. |

Требования к JDK

RT.Warehouse может использовать следующие версии Java для PL/Java и PXF:

1. Open JDK 8 или Open JDK 11, доступный на [AdoptOpenJDK](https://adoptopenjdk.net/).

Требования к оборудованию и сети

В следующей таблице перечислены минимальные рекомендуемые спецификации для аппаратных серверов, предназначенных для поддержки RT.Warehouse в системах Linux.

Все хост-серверы в вашей системе RT.Warehouse должны иметь одинаковую конфигурацию оборудования и программного обеспечения.

Таблица 1 — Минимальные требования к оборудованию

| Наименование | Значение |
| --- | --- |
| CPU | Любой CPU, совместимый с x86\_64. |
| RAM | 16 ГБ RAM на сервер. |
| Дисковое пространство | 150 МБ на каждый хост для установки RT.Warehouse;  Примерно 300 МБ на инстанс сегмента для метаданных;  Соответствующее свободное пространство для данных на дисках с ёмкостью не более 70%. |
| Сеть | 10 Gigabit Ethernet в массиве;  При наличии нескольких интерфейсов рекомендуется соединение NIC;  RT.Warehouse может использовать протоколы IPV4 или IPV6. |

Требования к хранилищу

RT.Warehouse следует запускать в файловой системе XFS.

RT.Warehouse может работать в сети или в общем хранилище, если общее хранилище представлено как блочное устройство серверам, на которых запущен RT.Warehouse, а файловая система XFS смонтирована на блочном устройстве. Использование сетевых файловых систем не рекомендуется. При использовании сетевого или общего хранилища необходимо использовать зеркальное отображение RT.Warehouse так же, как и локальное хранилище, и не следует вносить никаких изменений в схему зеркального отображения или схему восстановления сегментов.

Другие функции общего хранилища, такие как дедупликация и/или репликация, могут использоваться с RT.Warehouse, если они не мешают ожидаемой работе RT.Warehouse.

RT.Warehouse может быть развёрнут в виртуализированных системах только в том случае, если хранилище представлено как блочные устройства, а файловая система XFS смонтирована для хранения каталогов сегментов.

RT.Warehouse может работать на серверах Amazon Web Services (AWS) с использованием хранилища экземпляров Amazon (Amazon использует имена томов ephemeral[0-20]) или хранилища Amazon Elastic Block Store (Amazon EBS). При использовании хранилища Amazon EBS хранилище должно представлять собой RAID томов Amazon EBS и подключаться к файловой системе XFS.

Распределения Hadoop

RT.Warehouse обеспечивает доступ к HDFS с помощью Platform Extension Framework (PXF).

PXF v5.8.1 интегрирован с RT.Warehouse и обеспечивает доступ к Hadoop, хранилищу объектов и внешним хранилищам данных SQL.

PXF может использовать Cloudera, Hortonworks Data Platform, MapR и общие дистрибутивы Apache Hadoop.

PXF объединяет все файлы JAR, от которых он зависит, включая следующие библиотеки Hadoop:

Hadoop версии 2.9.2;

Hive версии 1.2.2;

HBase версии 1.3.2.

|  |
| --- |
| **Примечание.**  Если вы планируете получить доступ к данным в формате JSON, хранящимся в кластере Cloudera Hadoop, для PXF требуется дистрибутив Hadoop Cloudera версии 5.8 или более поздней. |

АРХИТЕКТУРА RT.WAREHOUSE

Ниже представлен общий обзор архитектуры системы RT.Warehouse.

RT.Warehouse хранит и обрабатывает большие объёмы данных, распределяя нагрузку между несколькими хостами. Логически RT.Warehouse — это массив отдельных баз данных PostgreSQL, работающих вместе для представления единого образа базы данных.

**Мастер** — это точка входа в систему RT.Warehouse. Это инстанс базы данных, к которому пользователи подключаются и отправляют команды SQL. Мастер координирует рабочую нагрузку в других инстансах базы данных в системе, называемых **Сегментами**, которые обрабатывают и хранят данные.

Сегменты связываются друг с другом и с мастером через **Интерконнект**, сетевой уровень RT.Warehouse.



RT.Warehouse — это только программное решение. Аппаратное обеспечение и программное обеспечение базы данных не связаны.

RT.Warehouse работает на различных серверных платформах от сертифицированных поставщиков оборудования. Производительность зависит от оборудования, на котором он установлен.

Поскольку база данных распределена по нескольким машинам в системе RT.Warehouse, правильный выбор и конфигурация оборудования важны для достижения максимальной производительности.

В этом разделе описываются основные компоненты системы RT.Warehouse, а также соображения и концепции оборудования, связанные с каждым компонентом: Мастер, Сегменты и Интерконнект. Кроме того, в системе могут быть дополнительные Хосты ETL для загрузки данных и #topic\_e5t\_whm\_kbb для мониторинга рабочей нагрузки и производительности запросов.

Мастер

Основной мастер

Мастер — это точка входа в систему RT.Warehouse. Это процесс сервера базы данных, который принимает клиентские соединения и обрабатывает команды SQL, которые передают пользователи системы. Пользователи подключаются к RT.Warehouse через мастера с помощью клиентской программы, совместимой с PostgreSQL, например, psql или ODBC-приложение.

Мастер ведёт системный каталог (набор системных таблиц, содержащих метаданные о самой системе RT.Warehouse), однако мастер не содержит никаких пользовательских данных. Данные хранятся только на сегментах. Мастер аутентифицирует клиентские соединения, обрабатывает входящие команды SQL, распределяет рабочую нагрузку между сегментами, координирует результаты, возвращаемые каждым сегментом, и представляет окончательные результаты клиентской программе.

Поскольку мастер не содержит никаких пользовательских данных, он имеет очень небольшую нагрузку на диск. Для мастера требуется быстрый выделенный ЦПУ для загрузки данных, обработки соединений и планирования запросов, поскольку часто требуется дополнительное пространство для загрузки файлов загрузки и файлов резервных копий, особенно в производственных средах. Клиенты могут также решить запустить ETL и инструменты отчётности на мастере, что требует большего дискового пространства и вычислительной мощности.

Резервный мастер

При желании вы можете развернуть резервную копию или зеркало инстанса мастера. Резервный мастер служит в качестве горячего резерва, если основной мастер становится неработоспособным. Вы можете развернуть резервный мастер на назначенном хосте резервного мастера или на одном из хостов сегментов.

Резервный мастер поддерживается в актуальном состоянии с помощью процесса репликации журнала транзакций, который выполняется на резервном мастере и синхронизирует данные между основным мастером и резервным мастером. Если основной мастер выходит из строя, процесс репликации журнала завершается, и администратор может активировать резервный мастер вместо него. Когда резервный мастер активен, реплицированные журналы используются для восстановления состояния основного мастера на момент последней успешно зафиксированной транзакции.

Поскольку мастер не содержит никаких пользовательских данных, необходимо синхронизировать только таблицы системного каталога между основной и резервной копиями. Когда эти таблицы обновятся, изменения автоматически будут скопированы на резервный мастер, поэтому он всегда синхронизирован с основным.



Сегменты

Основные сегменты

В RT.Warehouse в сегментах хранятся данные и выполняется большая часть обработки запросов. Пользовательские таблицы и их индексы распределяются по доступным сегментам в системе RT.Warehouse. Каждый сегмент содержит отдельную часть данных. Инcтансы сегментов — это процессы сервера базы данных, которые обслуживают сегменты. Пользователи не взаимодействуют напрямую с сегментами в системе RT.Warehouse, а делают это через мастера.

В эталонных конфигурациях оборудования RT.Warehouse количество инстансов сегментов на сегмент-хост определяется количеством эффективных CPU или ядер CPU. Например, если сегмент-хост имеет два двухъядерных процессора, вы можете расположить два или четыре основных сегмента на хосте. Если сегмент-хост имеет три четырехъядерных процессора, вы можете расположить три, шесть или двенадцать сегментов на хосте. Тестирование производительности поможет выбрать лучшее количество сегментов для выбранной аппаратной платформы.

Резервные сегменты

При развёртывании системы RT.Warehouse у вас есть возможность настроить зеркальные сегменты. Зеркальные сегменты позволяют запросам базы данных переключаться на резервный сегмент, если основной сегмент становится недоступным.

Зеркальный сегмент всегда должен находиться на другом хосте, нежели его основной сегмент. Зеркальные сегменты могут быть расположены по хостам в системе в одной из двух стандартных конфигураций или в пользовательской конфигурации, которую вы создаёте. Конфигурация по умолчанию, называемая групповым зеркалированием, размещает зеркальные сегменты для всех основных сегментов одного хоста на другом хосте. Другой вариант, называемый распределённым зеркалированием, распределяет зеркала для основных сегментов каждого хоста по остальным хостам. Распределённое зеркалирование требует, чтобы в системе было больше хостов, чем основных сегментов на хосте. На хостах с несколькими сетевыми интерфейсами основной и зеркальный сегменты равномерно распределяются между интерфейсами.

На рисунке ниже показано, как таблицы данных распределяются по сегментам при настройке параметра группового зеркалирования по умолчанию.



Отказоустойчивость сегментов и восстановление

Если в системе RT.Warehouse включено зеркалирование, система автоматически переключается на зеркальную копию, если основная копия становится недоступной. Система RT.Warehouse может оставаться работоспособной в случае отказа инстанса сегмента или хоста, только если все части данных доступны на оставшихся активных сегментах.

Если мастеу не удаётся подключиться к инстансу сегмента, он отмечает этот инстанс сегмента как недействительный в системном каталоге RT.Warehouse. Инстанс сегмента остаётся недействительным и не работает, пока администратор не вернёт этот сегмент обратно онлайн. Администратор может восстановить отказавший сегмент, пока система работает. В процессе восстановления копируются только те изменения, которые были пропущены, пока сегмент не работал.

Если у вас не включено зеркалирование и сегмент становится недействительным, система автоматически выключается. Перед продолжением операций администратор должен восстановить все отказавшие сегменты.

Пример аппаратного стека хоста сегмента

Независимо от выбранной вами аппаратной платформы рабочая нода обработки RT.Warehouse (хост сегмента) настраивается типично, как описано в этом разделе.

Хосты сегментов выполняют большую часть обработки базы данных, поэтому серверы хоста сегмента настроены таким образом, чтобы обеспечить максимальную производительность системы RT.Warehouse. Производительность RT.Warehouse будет такой же высокой, как и у самого медленного сервера сегмента в массиве. Поэтому важно убедиться, что базовое оборудование и операционные системы, на которых работает RT.Warehouse, работают на оптимальном уровне производительности. Также рекомендуется, чтобы все хосты сегмента в массиве RT.Warehouse имели идентичные аппаратные ресурсы и конфигурации.

Узлы сегмента также должны быть предназначены только для операций с RT.Warehouse. Чтобы получить наилучшую производительность запросов, вы не хотите, чтобы RT.Warehouse конкурировал с другими приложениями за ресурсы компьютера или сети.

На следующей схеме показан пример аппаратного стека хоста сегмента RT.Warehouse. Количество эффективных процессоров на хосте является основой для определения того, сколько инстансов основных сегментов RT.Warehouse развернуть на каждом хосте сегмента. В этом примере показан хост с двумя эффективными CPU (один двухъядерный CPU). Обратите внимание, что на каждое ядро CPU приходится один инстанс основного сегмента (или пара основной/зеркальный при использовании зеркалирования).



Пример схемы сегментного диска

Каждый CPU обычно отображается на логический диск. Логический диск состоит из одной основной файловой системы (и, возможно, зеркальной файловой системы), получающей доступ к пулу физических дисков через канал ввода-вывода или контроллер диска. Логический диск и файловая система предоставляются операционной системой. Большинство операционных систем предоставляют возможность логическому диску использовать группы физических дисков, организованных в RAID-массивы.



Интерконнект

Интерконнект — это сетевой уровень RT.Warehouse. Когда пользователь подключается к базе данных и отправляет запрос, в каждом из сегментов создаются процессы для обработки этого запроса. Интерконнект относится к межпроцессному взаимодействию между сегментами, а также к сетевой инфраструктуре, от которой зависит это взаимодействие. Интерконнект использует стандартную коммутационную матрицу 10 Gigabit Ethernet.

По умолчанию соединение с RT.Warehouse использует UDP (User Datagram Protocol) с управлением потоком для межсетевого трафика для отправки сообщений по сети. Программное обеспечение RT.Warehouse выполняет дополнительную проверку пакетов, не выполняемую UDP, поэтому надежность эквивалентна TCP (Transmission Control Protocol), а производительность и масштабируемость превосходят TCP.

Резервный интерконнект

Резервность интерконнекта может быть достигнуто путём развёртывания в вашей сети двух коммутаторов 10 Gigabit Ethernet и резервных подключений 10 Gigabit к серверу мастера RT.Warehouse и хост-серверам сегментов.

Конфигурация сетевого интерфейса

Хост сегмента обычно имеет несколько сетевых интерфейсов, предназначенных для межсетевого трафика RT.Warehouse. Хост мастера обычно имеет дополнительные внешние сетевые интерфейсы в дополнение к интерфейсам, используемым для межсетевого трафика.

В зависимости от количества доступных интерфейсов вы должны распределить сетевой трафик интерконнекта по количеству доступных интерфейсов. Это делается путем присвоения интансов сегментов конкретному сетевому интерфейсу и обеспечения равномерного распределения основных сегментов по количеству доступных интерфейсов.

Это делается путем создания отдельных имён адресов хостов для каждого сетевого интерфейса. Например, если у хоста четыре сетевых интерфейса, то у него будет четыре соответствующих адреса хоста, каждый из которых соответствует одному или нескольким инстансам основного сегмента. Файл /etc/hosts следует настроить так, чтобы он содержал не только имя хоста каждой машины, но также все адреса хостов интерфейса для всех хостов RT.Warehouse (мастера, резервного мастера, сегментов и хостов ETL).

При такой конфигурации операционная система автоматически выбирает лучший путь к месту назначения. RT.Warehouse автоматически балансирует сетевые назначения для максимального параллелизма.



Конфигурация коммутатора

При использовании нескольких коммутаторов 10 Gigabit Ethernet в массиве RT.Warehouse равномерно разделите количество подсетей между каждым коммутатором.

В этом примере конфигурации, если бы у нас было два коммутатора, NIC 1 и NIC 2 на каждом хосте использовали бы коммутатор 1, а NIC 3 и NIC 4 на каждом хосте использовали бы коммутатор 2. Для хоста мастера имя хоста, привязанное к NIC 1 (и, следовательно, с помощью коммутатора 1) — это эффективное имя хоста мастера для массива.

Таким образом, при развёртывании сервера резервного мастера для целей резервирования он должен отображаться на NIC, который использует другой коммутатор, чем основной мастер.



Хосты ETL для загрузки данных

RT.Warehouse поддерживает быструю параллельную загрузку данных с помощью функции внешних таблиц. Используя внешние таблицы в сочетании с параллельным файловым сервером RT.Warehouse (gpfdist), администраторы могут добиться максимального распараллеливания и загрузки полосы пропускания из своей системы RT.Warehouse. Многие производственные системы развёртывают выделенные серверы ETL для загрузки данных. На этих машинах работает параллельный файловый сервер RT.Warehouse (gpfdist), но не инстансы RT.Warehouse.

Одним из преимуществ использования программы файлового сервера gpfdist является то, что она гарантирует, что все сегменты в вашей системе RT.Warehouse полностью используются при чтении из файлов данных внешней таблицы.

Программа gpfdist может передавать данные инстансам сегмента со средней скоростью около 350 МБ/с для файлов с разделителями в текстовом формате и 200 МБ/с для файлов в формате CSV. Поэтому при запуске gpfdist вам следует учитывать следующие параметры, чтобы максимизировать пропускную способность сети ваших систем ETL:

1. Если ваш ETL-сервер сконфигурирован с несколькими сетевыми интерфейсными картами (NIC), как описано в разделе Конфигурация сетевого интерфейса, запустите один инстанс gpfdist на вашем ETL-хосте, а затем определите определение вашей внешней таблицы, чтобы имя хоста каждого NIC было объявлено в параметре LOCATION. Это позволяет сетевому трафику между хостами сегмента RT.Warehouse и ETL-хостом использовать все NIC одновременно.



Запустите несколько инстансов gpfdist на своем ETL-хосте и разделите файлы внешних данных поровну между каждым инстансом. Например, если у вас есть система ETL с двумя сетевыми картами (NIC), вы можете запустить два инстанса gpfdist на этом сервере, чтобы максимизировать производительность нагрузки. Затем вы должны равномерно разделить файлы данных внешней таблицы между двумя программами gpfdist.



ОЦЕНКА ЁМКОСТИ ХРАНИЛИЩА

Чтобы оценить, сколько данных может вместить ваша система RT.Warehouse, используйте данные измерения в качестве рекомендаций.

Также имейте в виду, что вам может потребоваться дополнительное пространство для размещения файлов резервных копий и файлов загрузки данных на каждом хосте сегмента.

Расчет полезной ёмкости диска

Чтобы вычислить, сколько данных может вместить система RT.Warehouse, вы должны рассчитать полезную ёмкость диска для каждого хоста сегмента, а затем умножить полученное число на количество хостов сегмента массива RT.Warehouse. Начните с необработанной ёмкости физических дисков на хосте сегмента, которые доступны для хранения данных (raw\_capacity), а именно:

|  |
| --- |
| disk\_size \* number\_of\_disks |

Учитывайте накладные расходы на форматирование файловой системы (примерно 10%) и уровень RAID, который вы используете. Например, при использовании RAID-10 расчёт будет следующим:

|  |
| --- |
| (raw\_capacity \* 0.9) / 2 = formatted\_disk\_space |

Для оптимальной производительности не заполняйте диски полностью, а используйте 70% или ниже. Помня об этом, рассчитайте доступное дисковое пространство следующим образом:

|  |
| --- |
| formatted\_disk\_space \* 0.7 = usable\_disk\_space |

После того, как вы отформатировали дисковые массивы RAID и учли максимальную рекомендуемую ёмкость (usable\_disk\_space), вам нужно будет рассчитать, сколько пространства фактически доступно для пользовательских данных (U). Если для избыточности данных используются зеркала RT.Warehouse, размер ваших пользовательских данных увеличится вдвое (2\*U). RT.Warehouse также требует, чтобы некоторое пространство было зарезервировано в качестве рабочей области для активных запросов. Рабочее пространство должно составлять примерно одну треть размера ваших пользовательских данных (рабочее пространство = U/3):

|  |
| --- |
| With mirrors: (2 \* U) + U/3 = usable\_disk\_space  Without mirrors: U + U/3 = usable\_disk\_space |

Рекомендации по временному файловому пространству и пространству пользовательских данных предполагают типичную аналитическую рабочую нагрузку. Рабочие нагрузки с высокой степенью одновременного выполнения или рабочие нагрузки с запросами, требующими очень большого количества временного пространства, могут выиграть от резервирования большей рабочей области. Как правило, общая пропускная способность системы может быть увеличена при уменьшении использования рабочей области за счёт надлежащего управления рабочей нагрузкой. Кроме того, временное пространство и пользовательское пространство можно изолировать друг от друга, указав, что они находятся в разных табличных пространствах.

Расчёт размера пользовательских данных

Как и во всех базах данных, размер необработанных данных будет немного больше после их загрузки в базу данных. В среднем необработанные данные будут занимать примерно в 1,4 раза больше на диске после загрузки в базу данных, но могут быть меньше или больше в зависимости от используемых типов данных, типа хранения таблиц, сжатия в базе данных и т.д.

1. Page Overhead — когда ваши данные загружаются в RT.Warehouse, они делятся на страницы по 32 КБ каждая. Каждая страница имеет 20 байт служебных данных.

Row Overhead — в обычной таблице хранения 'heap' каждая строка данных имеет 24 байта служебных данных строки. Таблица хранения 'append-optimized' имеет только 4 байта служебных строк.

Attribute Overhead — для самих значений данных размер, связанный с каждым значением атрибута, зависит от выбранного типа данных. Как правило, можно использовать минимально возможный тип данных для хранения (при условии, что известны возможные значения, которые будет иметь столбец).

Indexes — в RT.Warehouse индексы распределяются по хостам сегментов, как и данные таблицы. Тип индекса по умолчанию в RT.Warehouse — B-tree. Поскольку размер индекса зависит от количества уникальных значений в индексе и данных, которые нужно вставить, предварительный расчёт точного размера индекса невозможен. Однако с помощью этих формул можно приблизительно оценить размер индекса.

|  |
| --- |
| B-tree: unique\_values \* (data\_type\_size + 24 bytes)  Bitmap: (unique\_values \* number\_of\_rows \* 1 bit \* compression\_ratio / 8) + (unique\_values \* 32) |

Расчёт требований к пространству для метаданных и журналов

На каждом хосте сегмента необходимо учесть пространство для файлов журналов и метаданных RT.Warehouse:

1. Системные метаданные — для каждого инстанса сегмента RT.Warehouse (основного или зеркального) или инстанса мастера, работающего на хосте, оставьте приблизительно 20 МБ для системных каталогов и метаданных.

Write Ahead Log — для каждого сегмента RT.Warehouse (основного или зеркального) или инстанса мастера, работающего на хосте, выделите место для Write Ahead Log (WAL). WAL разделён на файлы сегментов по 64 МБ каждый. Максимальное количество файлов WAL вычисляются так:

|  |
| --- |
| 2 \* checkpoint\_segments + 1 |

Вы можете использовать эту формулу для оценки потребности в пространстве для WAL. По умолчанию параметр checkpoint\_segments для инстанса RT.Warehouse равен 8, что означает 1088 МБ пространства WAL, выделенного для каждого сегмента или инстанса мастера на хосте.

Файлы журналов RT.Warehouse — каждый инстанс сегмента и инстанс мастера создают файлы журнала базы данных, которые со временем будут расти. Для этих файлов журналов следует выделить достаточно места, и следует использовать какой-либо тип ротации журналов, чтобы файлы журналов не становились слишком большими.

Данные командного центра — агенты сбора данных, используемые командным центром, работают на том же наборе хостов, что и инстанс RT.Warehouse, и используют системные ресурсы этих хостов. Потребление ресурсов процессами агента сбора данных на этих хостах минимально и не должно существенно влиять на производительность базы данных. Исторические данные, собранные агентами сбора, хранятся в его собственной базе данных Command Center (с именем gpperfmon) в вашей системе RT.Warehouse. Собранные данные распределяются так же, как и обычные данные базы данных, поэтому вам нужно будет учитывать дисковое пространство в расположениях каталогов данных ваших инстансах сегментов RT.Warehouse. Требуемый объём места зависит от количества исторических данных, которые вы хотите сохранить. Исторические данные не сокращаются автоматически. Администраторы баз данных должны настроить политику сокращения, чтобы поддерживать размер базы данных Command Center.

НАСТРОЙКА ВАШЕЙ СИСТЕМЫ

Раздел описывает, как подготовить среду операционной системы для установки программного обеспечения RT.Warehouse.

Выполните следующие задачи по порядку:

1. Убедитесь, что ваши хост-системы соответствуют требованиям, описанным в разделе 1.

Отключите SELinux и файрвол.

Задайте необходимые параметры операционной системы.

Синхронизируйте системные часы.

Создайте учётную запись gpadmin.

Если не указано иное, эти задачи должны выполняться для всех хостов в массиве RT.Warehouse (мастера, резервного мастера и хостов сегментов).

Соглашение об именах хоста RT.Warehouse для хоста мастера — mdw, а для хоста резервного мастера — smdw.

Соглашение об именах хостов сегментов — sdwN, где sdw — префикс, а N — целое число. Например, имена хостов сегментов будут sdw1, sdw2 и так далее. Связывание NIC рекомендуется для хостов с несколькими интерфейсами, но когда интерфейсы не связаны, по соглашению к имени хоста добавляется тире (-) и номер. Например, sdw1-1 и sdw1-2 — это два имени интерфейса для хоста sdw1.

|  |
| --- |
| **Примечание.**  Автоматизация шагов настройки, описанных в этом разделе, и установка программного обеспечения RT.Warehouse с помощью инструмента подготовки системы, такого как Ansible, Chef или Puppet, могут сэкономить время и обеспечить надежную и повторяемую установку RT.Warehouse. |

Отключение программного обеспечения SELinux и файрвола

Для всех хост-систем RT.Warehouse под управлением РЕД ОС необходимо отключить SELinux. Выполните следующие действия:

1. Как пользователь root проверьте статус SELinux:

|  |
| --- |
| # sestatus  SELinuxstatus: disabled |

Если SELinux не отключён, отключите его, отредактировав файл /etc/selinux/config. От имени пользователя root измените значение параметра SELINUX в файле конфигурации следующим образом:

|  |
| --- |
| SELINUX=disabled |

Если в ваших системах установлен System Security Services Daemon (SSSD), отредактируйте файл конфигурации SSSD и установите для параметра selinux\_provider значение none, чтобы предотвратить связанные с SELinux отказы аутентификации SSH, которые могут возникать даже при отключённом SELinux. Отредактируйте /etc/sssd/sssd.conf как root и добавьте следующий параметр:

|  |
| --- |
| selinux\_provider=none |

Перезагрузите систему, чтобы применить все внесённые изменения, и убедитесь, что SELinux отключён.

Вам также следует отключить программное обеспечение файрвола, такое как iptables (в таких системах, как РЕД ОС), firewalld (в таких системах, как РЕД ОС) или ufw (в системах Ubuntu отключено по умолчанию).

Если вы решили включить iptables в RT.Warehouse в целях безопасности, см. [ОПЦИОНАЛЬНО] ВКЛЮЧЕНИЕ IPTABLES.

Чтобы отключить iptables, выполните следующие действия:

1. Как пользователь root проверьте состояние iptables:

|  |
| --- |
| # /sbin/chkconfig --list iptables |

Если iptables отключён, вывод команды будет следующим:

|  |
| --- |
| iptables 0:off 1:off 2:off 3:off 4:off 5:off 6:off |

При необходимости выполните эту команду от имени пользователя root, чтобы отключить iptables.

|  |
| --- |
| /sbin/chkconfig iptables off |

После применения изменений вам потребуется перезагрузить систему.

Для систем с firewalld проверьте состояние firewalld с помощью команды:

|  |
| --- |
| # systemctl status firewalld |

Если firewalld отключён, вывод команды будет следующим:

|  |
| --- |
| \* firewalld.service - firewalld - dynamic firewall daemon  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)  Active: inactive (dead) |

При необходимости выполните следующие команды от имени пользователя root, чтобы отключить firewalld:

|  |
| --- |
| # systemctl stop firewalld.service  # systemctl disable firewalld.service |

Дополнительную информацию о настройке программного обеспечения файрвола смотрите в документации по файрволу или вашей операционной системе.

Рекомендуемые настройки параметров ОС

RT.Warehouse требует, чтобы определённые параметры операционной системы (ОС) Linux были установлены на всех хостах в вашей RT.Warehouse (мастера и сегментов).

Как правило, необходимо изменить следующие категории параметров системы:

1. Shared Memory — инстанс RT.Warehouse не будет работать, если сегмент разделяемой памяти для вашего ядра не имеет надлежащего размера. Большинство установок ОС по умолчанию имеют слишком низкие значения общей памяти для RT.Warehouse. В системах Linux вы также должны отключить убийцу OOM (out of memory).

Сеть — в системах RT.Warehouse большого объёма необходимо установить определённые параметры настройки, связанные с сетью, чтобы оптимизировать сетевые соединения, выполняемые интерконнектом RT.Warehouse.

Пользовательские ограничения — пользовательские ограничения управляют ресурсами, доступными процессам, запускаемым оболочкой пользователя. RT.Warehouse требует более высокого ограничения на допустимое количество файловых дескрипторов, которые может открыть один процесс. Настройки по умолчанию могут привести к сбою некоторых запросов к RT.Warehouse, поскольку в них закончились файловые дескрипторы, необходимые для обработки запроса.

В частности, вам необходимо отредактировать следующие параметры конфигурации Linux:

1. Файл hosts.

Файл sysctl.conf.

Ограничения системных ресурсов.

Параметры монтирования XFS.

Настройки дискового ввода-вывода.

Опережающее чтение;

Планировщик дискового ввода-вывода.

Transparent Huge Pages (THP).

Удаление объекта IPC.

Лимит подключений по SSH.

Файл hosts

Отредактируйте файл /etc/hosts и убедитесь, что он включает имена хостов и все имена адресов интерфейсов для каждой машины, участвующей в вашей системе RT.Warehouse.

Файл sysctl.conf

Параметры sysctl.conf, перечисленные в этом разделе, предназначены для обеспечения производительности, оптимизации и согласованности в самых разных средах. Измените эти настройки в соответствии с вашей конкретной ситуацией и настройками.

Задайте параметры в файле /etc/sysctl.conf и перезагрузите его с помощью sysctl -p:

|  |
| --- |
| # kernel.shmall = \_PHYS\_PAGES / 2 # См. Страницы общей памяти  kernel.shmall = 197951838  # kernel.shmmax = kernel.shmall \* PAGE\_SIZE  kernel.shmmax = 810810728448  kernel.shmmni = 4096  vm.overcommit\_memory = 2 # См. Память хоста сегмента  vm.overcommit\_ratio = 95 # См. Память хоста сегмента  net.ipv4.ip\_local\_port\_range = 10000 65535 # См. Настройки порта  kernel.sem = 500 2048000 200 4096  kernel.sysrq = 1  kernel.core\_uses\_pid = 1  kernel.msgmnb = 65536  kernel.msgmax = 65536  kernel.msgmni = 2048  net.ipv4.tcp\_syncookies = 1  net.ipv4.conf.default.accept\_source\_route = 0  net.ipv4.tcp\_max\_syn\_backlog = 4096  net.ipv4.conf.all.arp\_filter = 1  net.core.netdev\_max\_backlog = 10000  net.core.rmem\_max = 2097152  net.core.wmem\_max = 2097152  vm.swappiness = 10  vm.zone\_reclaim\_mode = 0  vm.dirty\_expire\_centisecs = 500  vm.dirty\_writeback\_centisecs = 100  vm.dirty\_background\_ratio = 0 # См. Системная память  vm.dirty\_ratio = 0  vm.dirty\_background\_bytes = 1610612736  vm.dirty\_bytes = 4294967296 |

Страницы общей памяти

RT.Warehouse использует разделяемую память для связи между процессами postgres, которые являются частью одного и того же инстанса postgres.kernel.shmall устанавливает общий объём разделяемой памяти в страницах, который может использоваться в масштабе всей системы. kernel.shmmax устанавливает максимальный размер одного сегмента разделяемой памяти в байтах.

Установите значения kernel.shmall и kernel.shmax в зависимости от физической памяти вашей системы и размера страницы. Как правило, значение обоих параметров должно составлять половину физической памяти системы.

Используйте переменные операционной системы \_PHYS\_PAGES и PAGE\_SIZE, чтобы установить параметры.

|  |
| --- |
| kernel.shmall = ( \_PHYS\_PAGES / 2)  kernel.shmmax = ( \_PHYS\_PAGES / 2) \* PAGE\_SIZE |

Чтобы вычислить значения для kernel.shmall и kernel.shmax, выполните следующие команды, используя команду getconf, которая возвращает значение переменной операционной системы.

|  |
| --- |
| $ echo $(expr $(getconf \_PHYS\_PAGES) / 2)  $ echo $(expr $(getconf \_PHYS\_PAGES) / 2 \\* $(getconf PAGE\_SIZE)) |

В качестве наилучшей практики рекомендуется установить следующие значения в файле /etc/sysctl.conf, используя вычисленные значения. Например, в хост-системе установлено 1583 ГБ памяти, и она возвращает следующие значения: \_PHYS\_PAGES = 395903676 и PAGE\_SIZE = 4096. Это будут значения kernel.shmall и kernel.shmmax:

|  |
| --- |
| kernel.shmall = 197951838  kernel.shmmax = 810810728448 |

Если у мастера RT.Warehouse конфигурация разделяемой памяти отличается от конфигурации хостов сегмента, значения \_PHYS\_PAGES и PAGE\_SIZE могут отличаться, а значения kernel.shmall и kernel.shmax на хосте мастера будут отличаться от значений на хостах сегмента.

Память хоста сегмента

Параметр ядра Linux vm.overcommit\_memory используется ОС для определения объёма памяти, который можно выделить для процессов. Для RT.Warehouse этот параметр всегда должен иметь значение 2.

vm.overcommit\_ratio — это процент RAM, который используется для процессов приложений, а остальная часть зарезервирована для операционной системы. По умолчанию в Red Hat Enterprise Linux установлено значение 50.

Настройки порта

Чтобы избежать конфликтов портов между RT.Warehouse и другими приложениями во время инициализации RT.Warehouse, обратите внимание на диапазон портов, указанный параметром операционной системы net.ipv4.ip\_local\_port\_range.

При инициализации RT.Warehouse с использованием файла конфигурации кластера gpinitsystem не указывайте порты RT.Warehouse в этом диапазоне.

Например, если net.ipv4.ip\_local\_port\_range = 10000 65535, установите эти значения для номеров базовых портов RT.Warehouse.

|  |
| --- |
| PORT\_BASE = 6000  MIRROR\_PORT\_BASE = 7000 |

Для получения информации о файле конфигурации кластера gpinitsystem см. раздел 8.

Для развёртываний Azure с RT.Warehouse избегайте использования порта 65330; добавьте в sysctl.conf следующую строку:

|  |
| --- |
| net.ipv4.ip\_local\_reserved\_ports=65330 |

Системная память

Для хост-систем с объёмом памяти более 64 ГБ рекомендуются следующие настройки:

|  |
| --- |
| vm.dirty\_background\_ratio = 0  vm.dirty\_ratio = 0  vm.dirty\_background\_bytes = 1610612736 # 1.5GB  vm.dirty\_bytes = 4294967296 # 4GB |

Для хост-систем с 64 ГБ памяти или меньше удалите vm.dirty\_background\_bytes и vm.dirty\_bytes и установите для двух параметров отношения следующие значения:

|  |
| --- |
| vm.dirty\_background\_ratio = 3  vm.dirty\_ratio = 10 |

Увеличьте vm.min\_free\_kbytes, чтобы запросы PF\_MEMALLOC от сетевых драйверов и драйверов хранилища были легко удовлетворены. Это особенно важно для систем с большим объёмом системной памяти. В этих системах значение по умолчанию часто слишком мало. Используйте эту команду awk, чтобы установить для vm.min\_free\_kbytes рекомендуемые 3% системной физической памяти:

|  |
| --- |
| awk 'BEGIN {OFMT = "%.0f";} /MemTotal/ {print "vm.min\_free\_kbytes =", $2 \* .03;}'  /proc/meminfo >> /etc/sysctl.conf |

Не устанавливайте для vm.min\_free\_kbytes значение выше 5% системной памяти, так как это может привести к нехватке памяти.

Ограничения системных ресурсов

Задайте следующие параметры в файле /etc/security/limits.conf:

|  |
| --- |
| \* soft nofile 524288  \* hard nofile 524288  \* soft nproc 131072  \* hard nproc 131072 |

Для систем РЕД ОС значения параметров в файле /etc/security/limits.d/90-nproc.conf (РЕД ОС) или /etc/security/limits.d/20-nproc.conf (РЕД ОС) переопределяет значения в файле limits.conf. Убедитесь, что для всех параметров в файле переопределения установлено необходимое значение. Модуль Linux pam\_limits устанавливает ограничения для пользователей, считывая значения из файла limits.conf, а затем из файла переопределения. Для получения информации о PAM и пользовательских лимитах смотрите документацию по PAM и pam\_limits.

Выполните команду ulimit -u на каждом хосте сегмента, чтобы отобразить максимальное количество процессов, доступных каждому пользователю. Убедитесь, что возвращаемое значение — 131072.

Параметры монтирования XFS

XFS является предпочтительной файловой системой хранения данных на платформах Linux. Используйте команду mount со следующими рекомендуемыми параметрами монтирования XFS:

|  |
| --- |
| rw,nodev,noatime,nobarrier,inode64 |

Параметры XFS также можно установить в файле /etc/fstab. В этом примере записи из файла fstab указаны параметры XFS.

|  |
| --- |
| /dev/data /data xfs nodev,noatime,nobarrier,inode64 0 0 |

Настройки дискового ввода-вывода

1. Значение опережающего чтения.

Каждый файл дискового устройства должен иметь значение упреждающего чтения (blockdev) 16384. Чтобы проверить значение упреждающего чтения дискового устройства:

|  |
| --- |
| # /sbin/blockdev --getra devname |

Например:

|  |
| --- |
| # /sbin/blockdev --getra /dev/sdb |

Чтобы установить blockdev (упреждающее чтение) на устройстве:

|  |
| --- |
| # /sbin/blockdev --setra bytes devname |

Например:

|  |
| --- |
| # /sbin/blockdev --setra 16384 /dev/sdb |

|  |
| --- |
| **Примечание.**  Команда blockdev --setra не является постоянной, её нужно запускать каждый раз при перезагрузке системы. Как запустить команду, зависит от вашей системы, но вы должны убедиться, что параметр упреждающего чтения устанавливается каждый раз при перезагрузке системы. |

Планировщик дискового ввода-вывода

Планировщик дискового ввода-вывода Linux для доступа к диску поддерживает различные политики, такие как CFQ, AS и deadline.

Рекомендуется использовать планировщик deadline. Чтобы указать планировщик до следующей перезагрузки системы, выполните следующее:

|  |
| --- |
| # echo schedulername > /sys/block/devname/queue/scheduler |

Например:

|  |
| --- |
| # echo deadline > /sys/block/sbd/queue/scheduler |

|  |
| --- |
| **Примечание.**  Использование команды echo для установки политики планировщика дискового ввода-вывода не является постоянным, поэтому вы должны убедиться, что команда запускается при каждой перезагрузке системы. Как запустить команду, зависит от вашей системы. |

Один из способов установить политику планировщика ввода-вывода во время загрузки — использовать параметр ядра elevator. Добавьте параметр elevator=deadline в команду ядра в файле /boot/grub/grub.conf, файле конфигурации загрузчика GRUB. Это пример команды ядра из файла grub.conf в РЕД ОС. Команда состоит из нескольких строк для удобства чтения.

|  |
| --- |
| kernel /vmlinuz-2.6.18-274.3.1.el5 ro root=LABEL=/  elevator=deadline crashkernel=128M@16M quiet console=tty1  console=ttyS1,115200 panic=30 transparent\_hugepage=never  initrd /initrd-2.6.18-274.3.1.el5.img |

Чтобы указать планировщик ввода-вывода во время загрузки в системах, использующих grub2, таких как РЕД ОС, используйте системную утилиту grubby. Эта команда добавляет параметр при запуске от имени пользователя root.

|  |
| --- |
| # grubby --update-kernel=ALL --args="elevator=deadline" |

После добавления параметра перезагрузите систему.

Команда grubby отображает настройки параметров ядра.

|  |
| --- |
| # grubby --info=ALL |

Дополнительные сведения об утилите grubby смотрите в документации к операционной системе. Если команда grubby не обновляет ядра, см. примечание в Лимит подключений по SSH.

Transparent Huge Pages (THP)

В системах, использующих grub2, таких как РЕД ОС 7.x, используйте системную утилиту grubby. Эта команда добавляет параметр при запуске от имени пользователя root.

|  |
| --- |
| # grubby --update-kernel=ALL --args="transparent\_hugepage=never" |

После добавления параметра перезагрузите систему.

Для систем Ubuntu установите пакет hugepages и выполните эту команду от имени пользователя root:

|  |
| --- |
| # hugeadm --thp-never |

Эта команда cat проверяет состояние THP. Вывод показывает, что THP отключён.

|  |
| --- |
| $ cat /sys/kernel/mm/\*transparent\_hugepage/enabled  always [never] |

Дополнительную информацию о THP или утилите grubby смотрите в документации по операционной системе. Если команда grubby не обновляет ядра, см. примечание в Лимит подключений по SSH.

Удаление объекта IPC

Отключите удаление объекта IPC для РЕД ОС или Ubuntu. Параметр systemd по умолчанию RemoveIPC=yes удаляет IPC-соединения при выходе из системы несистемных учётных записей пользователей. Это приводит к сбою утилиты gpinitsystem RT.Warehouse с ошибками семафоров. Чтобы избежать этой проблемы, выполните одно из следующих действий.

1. Когда вы добавляете учётную запись пользователя операционной системы gpadmin в мастере в п. 4.4, создайте пользователя как системную учётную запись.

Отключите RemoveIPC. Установите этот параметр в /etc/systemd/logind.conf в хост-системах RT.Warehouse.

|  |
| --- |
| RemoveIPC=no |

Настройка вступает в силу после перезапуска службы systemd-login или перезагрузки системы. Чтобы перезапустить службу, запустите эту команду от имени пользователя root.

|  |
| --- |
| service systemd-logind restart |

Лимит подключений по SSH

Некоторые утилиты управления RT.Warehouse, включая gpexpand, gpinitsystem и gpaddmirrors, для выполнения своих задач используют соединения secure shell (SSH) между системами. В развёртываниях больших RT.Warehouse, облачных развёртываниях или развёртываниях с большим количеством сегментов на хост эти утилиты могут превышать максимальный лимит для неаутентифицированных соединений на хостах. Когда это происходит, вы получаете такие ошибки, как: ssh\_exchange\_identification: Connection closed by remote host.

Чтобы увеличить этот порог подключения для вашей системы RT.Warehouse, обновите параметр конфигурации SSH MaxStartups в одном из файлов конфигурации демона SSH /etc/ssh/sshd\_config или /etc/sshd\_config.

Если вы укажете MaxStartups с использованием единственного целочисленного значения, вы определите максимальное количество одновременных неаутентифицированных подключений. Например:

|  |
| --- |
| MaxStartups 200 |

Если вы укажете MaxStartups с использованием синтаксиса "start:rate:full", вы включите случайное раннее прерывание соединения с помощью демона SSH. start определяет максимальное количество разрешённых попыток SSH-соединения без аутентификации. По достижении start числа попыток подключения без аутентификации демон SSH отклоняет процент последующих попыток подключения rate. full определяет максимальное количество попыток подключения без аутентификации, после которых все попытки отклоняются. Например:

|  |
| --- |
| Max Startups 10:30:200 |

Подробную информацию о параметрах конфигурации SSH смотрите в документации по SSH для вашего дистрибутива Linux.

|  |  |  |
| --- | --- | --- |
| **Примечание.**  Если команда grubby не обновляет ядра системы РЕД ОС 7.x, вы можете вручную обновить все ядра в системе. Например, чтобы добавить параметр transparent\_hugepage=never ко всем ядрам в системе.   1. Добавьте параметр в строку GRUB\_CMDLINE\_LINUX в параметре файла в /etc/default/grub.  |  | | --- | | GRUB\_TIMEOUT=5  GRUB\_DISTRIBUTOR="$(sed 's, release .\*$,,g' /etc/system-release)"  GRUB\_DEFAULT=saved  GRUB\_DISABLE\_SUBMENU=true  GRUB\_TERMINAL\_OUTPUT="console"  GRUB\_CMDLINE\_LINUX="crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet transparent\_hugepage=never"  GRUB\_DISABLE\_RECOVERY="true" |   От имени пользователя root запустите команду grub2-mkconfig, чтобы обновить ядра.   |  | | --- | | # grub2-mkconfig -o /boot/grub2/grub.cfg |   Перезагрузите систему. |

Синхронизация системных часов

Вы должны использовать NTP (Network Time Protocol) для синхронизации системных часов на всех хостах, составляющих вашу систему базы данных RT.Warehouse. См. [www.ntp.org](http://www.ntp.org/) для получения дополнительной информации о NTP.

NTP на хостах сегмента должен быть настроен на использование хоста мастера в качестве основного источника времени и резервного мастера в качестве вторичного источника времени. На хостах мастера и реервного мастера настройте NTP так, чтобы он указывал на выбранный вами сервер времени.

Настройка NTP

Чтобы настроить NTP, выполните следующие действия:

1. На хосте мастера войдите в систему как root и отредактируйте файл /etc/ntp.conf. Установите параметр server, чтобы он указывал на сервер времени NTP вашего центра обработки данных. Например (если IP-адрес NTP-сервера вашего центра обработки данных — 10.6.220.20):

|  |
| --- |
| server 10.6.220.20 |

На каждом хосте сегмента войдите в систему как root и отредактируйте файл /etc/ntp.conf. Установите первый параметр server так, чтобы он указывал на хост мастера, а второй параметр server, чтобы указывать на хост резервного мастера. Например:

|  |
| --- |
| server mdw prefer  server smdw |

На хосте резервного мастера войдите в систему как root и отредактируйте файл /etc/ntp.conf. Установите первый параметр server так, чтобы он указывал на хост основого мастера, а второй параметр server — на сервер времени NTP вашего центра обработки данных. Например:

|  |
| --- |
| server mdw prefer  server 10.6.220.20 |

На хосте мастера используйте демон NTP для синхронизации системных часов на всех хостах RT.Warehouse. Например, используя gpssh:

|  |
| --- |
| # gpssh -f hostfile\_gpssh\_allhosts -v -e 'ntpd' |

Создание пользователя с правами администратора RT.Warehouse

Создайте специальную учётную запись пользователя операционной системы на каждом хосте для запуска и администрирования RT.Warehouse. Эта учётная запись пользователя по соглашению называется gpadmin.

|  |
| --- |
| **Внимание.**  Вы не можете запустить сервер RT.Warehouse от имени пользователя root. |

Пользователь gpadmin должен иметь разрешение на доступ к службам и каталогам, необходимым для установки и запуска RT.Warehouse.

Пользователь gpadmin на каждом хосте RT.Warehouse должен иметь установленную пару ключей SSH и иметь возможность подключаться по SSH с любого хоста в кластере к любому другому хосту в кластере без ввода пароля или ключевой фразы (так называемый "passwordless SSH"). Если вы включите SSH без пароля с хоста мастера на все остальные хосты в кластере ("1-n passwordless SSH"), вы можете позже использовать утилиту командной строки RT.Warehouse gpssh-exkeys, чтобы включить беспарольный SSH с каждого хоста на любой другой хост ("n-n passwordless SSH").

При желании вы можете предоставить пользователю gpadmin привилегию sudo, чтобы вы могли легко администрировать все хосты в кластере RT.Warehouse как gpadmin с помощью команд sudo, ssh/scp и gpssh/gpscp.

Следующие шаги показывают, как настроить пользователя gpadmin на хосте, установить пароль, создать пару ключей SSH и [опционально] включить возможность sudo. Эти действия должны выполняться с правами root на каждом хосте кластера RT.Warehouse. (Для большого кластера RT.Warehouse вам нужно автоматизировать эти шаги, используя инструменты подготовки вашей системы.)

|  |
| --- |
| **Примечание.**  См. ПРИМЕР ANSIBLE PLAYBOOK для примера, который показывает, как автоматизировать задачи по созданию пользователя gpadmin и установке программного обеспечения RT.Warehouse на всех хостах в кластере. |

1. Создайте группу и пользователя gpadmin.

|  |
| --- |
| **Примечание.**  Если вы устанавливаете RT.Warehouse на РЕД ОС и хотите отключить удаление объекта IPC, создав пользователя gpadmin в качестве системной учётной записи, укажите как параметр -r (создание пользователя как системной учётной записи), так и параметр -m (создание домашнего каталога) для команды useradd. В системах Ubuntu вы должны использовать параметр -m с командой useradd, чтобы создать домашний каталог для пользователя. |

В этом примере создаётся группа gpadmin, создаётся пользователь gpadmin как системная учётная запись с домашним каталогом и как член группы gpadmin, а также создаётся пароль для пользователя.

|  |
| --- |
| # groupadd gpadmin  # useradd gpadmin -r -m -g gpadmin  # passwd gpadmin  New password: <changeme>  Retype new password: <changeme> |

|  |
| --- |
| **Примечание.**  Убедитесь, что у пользователя gpadmin одинаковые номера идентификатора пользователя (uid) и идентификатора группы (gid) на каждом хосте, чтобы предотвратить проблемы со скриптами или службами, которые используют их для идентификации или разрешений.  Например, резервное копирование RT.Warehouse на некоторые сетевые файловые системы или устройства хранения может завершиться ошибкой, если у пользователя gpadmin разные номера uid или gid на разных хостах сегмента.  Когда вы создаете группу и пользователя gpadmin, вы можете использовать параметр groupadd -g, чтобы указать номер gid, и параметр useradd -u, чтобы указать номер uid. Используйте команду id gpadmin, чтобы увидеть uid и gid для пользователя gpadmin на текущем хосте. |

Переключитесь на пользователя gpadmin и сгенерируйте пару ключей SSH для пользователя gpadmin.

|  |
| --- |
| $ su gpadmin  $ ssh-keygen -t rsa -b 4096  Generating public/private rsa key pair.  Enter file in which to save the key (/home/gpadmin/.ssh/id\_rsa):  Created directory '/home/gpadmin/.ssh'.  Enter passphrase (empty for no passphrase):  Enter same passphrase again: |

При появлении запроса на парольную фразу нажмите Enter, чтобы для соединений SSH не требовалось вводить парольную фразу.

[Опционально] Предоставьте sudo доступ пользователю gpadmin.

В РЕД ОС запустите visudo и раскомментируйте запись группы %wheel.

|  |
| --- |
| %wheel ALL=(ALL) NOPASSWD: ALL |

Убедитесь, что вы раскомментируете строку с ключевым словом NOPASSWD.

Добавьте пользователя gpadmin в группу wheel с помощью команды.

|  |
| --- |
| # usermod -aG wheel gpadmin |

УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ RT.WAREHOUSE

Раздел описывает, как установить бинарные файлы программного обеспечения RT.Warehouse на все хосты, которые будут составлять вашу систему RT.Warehouse, как включить SSH без пароля для пользователя gpadmin и как проверить установку.

Установка RT.Warehouse

Вы должны установить RT.Warehouse на каждом хост-компьютере системы RT.Warehouse. Вы можете настроить минимальную систему RT.Warehouse для работы в одной хост-системе с инстансом мастера и инстансами сегмента, работающими в одной хост-системе.

Выпуски RT.Warehouse доступны в виде tar-архивов исходного кода, установщиков RPM для РЕД ОС и пакетов DEB для Debian и Ubuntu. Для операционной системы Ubuntu Greenplum также предлагает бинарный файл, который можно установить с помощью команды apt-get в системе Ubuntu Personal Package Archive.

Перед тем как приступить к установке RT.Warehouse, убедитесь, что вы выполнили действия, описанные в разделе 4, для настройки каждой из хост-машин мастера, резервного мастера и сегментов для RT.Warehouse.

|  |
| --- |
| **Внимание.**  После установки RT.Warehouse необходимо установить переменные среды RT.Warehouse. См. п. 8.3. |

См. раздел 15, в котором показан пример сценария, показывающий, как можно автоматизировать создание пользователя gpadmin и установку RT.Warehouse.

Следуйте этим инструкциям, чтобы установить RT.Warehouse из предварительно созданного бинарного файла.

|  |
| --- |
| **Внимание.**  Вам потребуется доступ sudo или root для установки из предварительно созданного бинарного файла. |

1. Загрузите и скопируйте пакет RT.Warehouse в домашний каталог пользователя gpadmin на хост-машины мастера, резервного мастера и сегментов. Имя файла дистрибутива имеет формат greenplum-db-<version>-<platform>.rpm для систем РЕД ОС или greenplum-db-<version>-<platform>.deb для систем Ubuntu, где <platform> похож на rhel7-x86\_64 (Red Hat 7 64-bit).

С помощью sudo (или от имени пользователя root) установите пакет RT.Warehouse на каждый хост-компьютер с помощью программного обеспечения управления пакетами вашей системы.

Для систем РЕД ОС выполните команду yum:

|  |
| --- |
| $ sudo yum install ./greenplum-db-<version>-<platform>.rpm |

Для систем Ubuntu выполните команду apt:

|  |
| --- |
| $ sudo apt install ./greenplum-db-<version>-<platform>.deb |

Команда yum или apt устанавливает зависимости программного обеспечения, копирует файлы программного обеспечения RT.Warehouse в каталог для конкретной версии /usr/local/greenplum-db-<version> и создаёт символическую ссылку /usr/local/greenplum-db на каталог установки.

Измените владельца и группу установленных файлов на gpadmin:

|  |
| --- |
| $ sudo chown -R gpadmin:gpadmin /usr/local/greenplum\* |

Включение SSH без пароля

Пользователь gpadmin на каждом хосте RT.Warehouse должен иметь возможность подключаться по SSH с любого хоста в кластере к любому другому хосту в кластере без ввода пароля или ключевой фразы (так называемый "passwordless SSH"). Если вы включите SSH без пароля с хоста мастера на все остальные хосты в кластере ("1-n passwordless SSH"), вы можете использовать утилиту командной строки RT.Warehouse gpssh-exkeys, чтобы включить беспарольный SSH с каждого хоста на любой другой хост ("n-n passwordless SSH").

1. Войдите на хост мастера как пользователь gpadmin.

Найдите файл пути в каталоге установки RT.Warehouse.

|  |
| --- |
| $ source /usr/local/greenplum-db-<version>/greenplum\_path.sh |

|  |
| --- |
| **Примечание.**  Добавьте указанную выше исходную команду в .bashrc пользователя gpadmin или другой файл запуска оболочки, чтобы путь к RT.Warehouse и переменные среды устанавливались при каждом входе в систему как gpadmin. |

С помощью команды ssh-copy-id добавьте открытый ключ пользователя gpadmin в SSH-файл authorized\_hosts на всех остальных хостах кластера.

|  |
| --- |
| $ ssh-copy-id smdw  $ ssh-copy-id sdw1  $ ssh-copy-id sdw2  $ ssh-copy-id sdw3  . . . |

Команда включает "1-n passwordless SSH". Вам будет предложено ввести пароль пользователя gpadmin для каждого хоста. Если у вас есть команда sshpass в вашей системе, вы можете использовать следующую команду, чтобы избежать запроса:

|  |
| --- |
| $ SSHPASS=<password> sshpass -e ssh-copy-id smdw |

В домашнем каталоге gpadmin создайте файл с именем hostfile\_exkeys, который содержит настроенные на компьютере имена хостов и адреса хостов (имена интерфейсов) для каждого хоста в вашей системе RT.Warehouse (хостов мастера, резервого мастера и сегментов). Убедитесь, что нет пустых строк или лишних пробелов. Проверьте файл /etc/hosts в вашей системе на предмет правильных имён хостов для использования в вашей среде. Например, если у вас есть мастер, резервный мастер и три хоста сегмента с двумя несвязанными сетевыми интерфейсами на хост, ваш файл будет выглядеть примерно так:

|  |
| --- |
| mdw  mdw-1  mdw-2  smdw  smdw-1  smdw-2  sdw1  sdw1-1  sdw1-2  sdw2  sdw2-1  sdw2-2  sdw3  sdw3-1  sdw3-2 |

Запустите утилиту gpssh-exkeys с файлом hostfile\_exkeys, чтобы включить "n-n passwordless SSH" для пользователя gpadmin.

|  |
| --- |
| $ gpssh-exkeys -f hostfile\_exkeys |

Подтверждение вашей установки

Чтобы убедиться, что программное обеспечение RT.Warehouse было установлено и настроено правильно, выполните следующие шаги подтверждения с хоста мастера RT.Warehouse. При необходимости исправьте любые проблемы, прежде чем переходить к следующей задаче.

1. Войдите на хост мастера как gpadmin:

|  |
| --- |
| $ su - gpadmin |

С помощью утилиты gpssh проверьте, можете ли вы войти на все хосты без запроса пароля, и убедитесь, что программа RT.Warehouse была установлена ​​на всех хостах. Используйте файл hostfile\_exkeys, который вы использовали для настройки SSH без пароля. Например:

|  |
| --- |
| $ gpssh -f hostfile\_exkeys -e 'ls -l /usr/local/greenplum-db-<version>' |

Если установка прошла успешно, вы сможете войти на все хосты без запроса пароля. Все хосты должны показать, что у них одинаковое содержимое в их каталогах установки, и что каталоги принадлежат пользователю gpadmin.

Если вам будет предложено ввести пароль, выполните следующую команду, чтобы повторить обмен ключами ssh:

|  |
| --- |
| $ gpssh-exkeys -f hostfile\_exkeys |

Об установке базы данных RT.Warehouse

greenplum\_path.sh — файл, содержащий переменные среды для RT.Warehouse. См. п. 8.3;

bin — каталог, который содержит утилиты управления RT.Warehouse. Этот каталог также содержит клиентские и серверные программы PostgreSQL, большинство из которых также используются в RT.Warehouse;

docs/cli\_help — каталог, содержащий файлы справки для утилит командной строки RT.Warehouse;

docs/cli\_help/gpconfigs — каталог, содержащий образцы файлов конфигурации gpinitsystem и файлы хоста, которые можно изменять и использовать при установке и инициализации системы RT.Warehouse;

ext — связанные программы (например, Python), используемые некоторыми утилитами RT.Warehouse;

include — файлы заголовков C для RT.Warehouse;

lib — файлы RT.Warehouse и библиотеки PostgreSQL;

sbin — вспомогательные/внутренние скрипты и программы;

share — общие файлы для RT.Warehouse.

СОЗДАНИЕ ОБЛАСТЕЙ ХРАНЕНИЯ ДАННЫХ

Раздел описывает, как создать расположение каталогов, где хранятся данные RT.Warehouse для каждого инстанса мастера, резервного мастера и сегментов.

Создание областей хранения данных на хостах мастера и резервного мастера

На хостах мастера и резервного мастера RT.Warehouse требуется область хранения данных для хранения системных данных RT.Warehouse, таких как данные каталога и другие метаданные системы.

Создание расположения каталога данных на мастере

Расположение каталога данных на мастере отличается от местоположения в сегментах. Мастер не хранит никаких пользовательских данных, только таблицы системного каталога и системные метаданные хранятся на мастере, поэтому вам не нужно назначать столько места для хранения, сколько в сегментах.

1. Создайте или выберите каталог, который будет служить вашей областью хранения основных данных. Этот каталог должен иметь достаточно места на диске для ваших данных и принадлежать пользователю и группе gpadmin. Например, выполните следующие команды от имени пользователя root:

|  |
| --- |
| # mkdir -p /data/master |

Измените право собственности на этот каталог на пользователя gpadmin. Например:

|  |
| --- |
| # chown gpadmin:gpadmin /data/master |

С помощью gpssh также создайте местоположение каталога основных данных на сервере резервного мастера. Например:

|  |
| --- |
| # source /usr/local/greenplum-db/greenplum\_path.sh  # gpssh -h smdw -e 'mkdir -p /data/master'  # gpssh -h smdw -e 'chown gpadmin:gpadmin /data/master' |

Создание областей хранения данных на хостах сегментов

На хостах сегмента RT.Warehouse для основных сегментов требуются области хранения данных. Для зеркальных сегментов требуются отдельные места для хранения.

Создание расположения каталогов данных на всех хостах сегмента

1. На хосте мастера войдите как root:

|  |
| --- |
| # su |

Создайте файл с именем hostfile\_gpssh\_segonly. В этом файле должно быть указано только одно имя хоста, настроенное для каждого хоста сегмента. Например, если у вас есть три хоста сегмента:

|  |
| --- |
| sdw1  sdw2  sdw3 |

С помощью gpssh создайте основные и зеркальные каталоги данных на всех хостах сегмента одновременно, используя только что созданный файл hostfile\_gpssh\_segonly. Например:

|  |
| --- |
| # source /usr/local/greenplum-db/greenplum\_path.sh  # gpssh -f hostfile\_gpssh\_segonly -e 'mkdir -p /data/primary'  # gpssh -f hostfile\_gpssh\_segonly -e 'mkdir -p /data/mirror'  # gpssh -f hostfile\_gpssh\_segonly -e 'chown -R gpadmin /data/\*' |

ПРОВЕРКА ВАШИХ СИСТЕМ

Проверьте свое оборудование и производительность сети. RT.Warehouse предоставляет утилиту управления под названием gpcheckperf, которую можно использовать для выявления проблем аппаратного и системного уровня на машинах в массиве RT.Warehouse. gpcheckperf запускает сеанс на указанных хостах и выполняет следующие тесты производительности:

Производительность сети (gpnetbench\*);

Производительность дискового ввода-вывода (dd test);

Пропускная способность памяти (stream test).

Перед использованием gpcheckperf настройте доверенность хостов между хостами, участвующими в тесте производительности. Вы можете использовать утилиту gpssh-exkeys для обновления известных файлов хоста и обмена открытыми ключами между хостами, если вы этого ещё не сделали. Обратите внимание, что gpcheckperf вызывает gpssh и gpscp, поэтому эти утилиты RT.Warehouse должны быть в вашем $PATH.

1. Проверка производительности сети.

Проверка дискового ввода-вывода и пропускной способности памяти.

Проверка производительности сети

Чтобы проверить производительность сети, запустите gpcheckperf с одним из вариантов запуска теста сети: тест параллельной пары (-r N), тест последовательной пары (-r n) или полный матричный тест (-r M). Утилита запускает программу тестирования производительности сети, которая передает 5-секундный поток данных с текущего хоста на каждый удалённый хост, включённый в тест. По умолчанию данные передаются параллельно каждому удалённому хосту, а минимальная, максимальная, средняя и медианная скорость передачи данных по сети указывается в мегабайтах в секунду (МБ/с). Если итоговая скорость передачи ниже ожидаемой (менее 100 МБ/с), вы можете запустить тест сети последовательно, используя параметр -r n, чтобы получить результаты для каждого хоста. Чтобы запустить полноматричный тест пропускной способности, вы можете указать -r M, который заставит каждый хост отправлять и получать данные с любого другого указанного хоста. Этот тест лучше всего использовать для проверки того, может ли коммутационная матрица выдерживать полноматричную рабочую нагрузку.

Большинство систем в массиве RT.Warehouse сконфигурировано с несколькими сетевыми интерфейсными картами (NIC), каждая из которых находится в своей собственной подсети. При тестировании производительности сети важно тестировать каждую подсеть индивидуально. Например, учитывая следующую конфигурацию сети с двумя NIC на хост:

Таблица 2 — Пример конфигурации сетевого интерфейса

| Хост RT.Warehouse | NIC подсети 1 | NIC подсети 2 |
| --- | --- | --- |
| Сегмент 1 | sdw1-1 | sdw1-2 |
| Сегмент 2 | sdw2-1 | sdw2-2 |
| Сегмент 3 | sdw3-1 | sdw3-2 |

Вы должны создать четыре отдельных файла хоста для использования с сетевым тестом gpcheckperf:

Таблица 3 — Пример содержимого файла хоста для проверки сети

| hostfile\_gpchecknet\_ic1 | hostfile\_gpchecknet\_ic2 |
| --- | --- |
| sdw1-1 | sdw1-2 |
| sdw2-1 | sdw2-2 |
| sdw3-1 | sdw3-2 |

Затем вы должны запустить gpcheckperf один раз для каждой подсети. Например (при тестировании чётного числа хостов запустите в режиме тестирования параллельных пар):

|  |
| --- |
| $ gpcheckperf -f hostfile\_gpchecknet\_ic1 -r N -d /tmp > subnet1.out  $ gpcheckperf -f hostfile\_gpchecknet\_ic2 -r N -d /tmp > subnet2.out |

Если у вас нечётное количество хостов для тестирования, вы можете запустить его в режиме последовательного тестирования (-r n).

Проверка дискового ввода-вывода и пропускной способности памяти

Чтобы проверить производительность диска и пропускной способности памяти, запустите gpcheckperf с параметрами тестового запуска диска и потока (-r ds). При тестировании диска используется команда dd (стандартная утилита UNIX) для проверки последовательной пропускной способности логического диска или файловой системы. Тест памяти использует программу тестирования STREAM для измерения устойчивой пропускной способности памяти. Результаты сообщаются в МБ в секунду (МБ/с).

Чтобы запустить дисковый и потоковый тесты:

1. Войдите в систему на хосте мастера как пользователь gpadmin.

Загрузите файл пути greenplum\_path.sh из вашей установки RT.Warehouse. Например:

|  |
| --- |
| $ source /usr/local/greenplum-db/greenplum\_path.sh |

Создайте файл хоста с именем hostfile\_gpcheckperf с одним именем хоста для каждого хоста сегмента. Не включайте хост мастера. Например:

|  |
| --- |
| sdw1  sdw2  sdw3  sdw4 |

Запустите утилиту gpcheckperf, используя только что созданный файл hostfile\_gpcheckperf. Используйте параметр -d, чтобы указать файловые системы, которые вы хотите протестировать на каждом хосте (у вас должен быть доступ для записи в эти каталоги). Вам нужно будет протестировать все расположения каталогов данных основного и зеркального сегментов. Например:

|  |
| --- |
| $ gpcheckperf -f hostfile\_gpcheckperf -r ds -D \  -d /data1/primary -d /data2/primary \  -d /data1/mirror -d /data2/mirror |

Утилите может потребоваться время для выполнения тестов, поскольку она копирует очень большие файлы между хостами. Когда он будет завершён, вы увидите сводные результаты тестов Disk Write, Disk Read и Stream.

ИНИЦИАЛИЗАЦИЯ СИСТЕМЫ RT.WAREHOUSE

Раздел описывает, как инициализировать систему RT.Warehouse.

Инструкции в этом разделе предполагают, что вы уже подготовили свои хосты, как описано в разделе 4, и установили программное обеспечение RT.Warehouse на всех хостах в системе в соответствии с инструкциями в разделе 5.

Обзор

Поскольку RT.Warehouse является распределённой, процесс инициализации системы управления базой данных (СУБД) RT.Warehouse включает в себя инициализацию нескольких отдельных инстансов базы данных PostgreSQL (называемых инстансами сегментов в RT.Warehouse).

Каждый инстанс базы данных (мастер и все сегменты) должен быть инициализирован на всех хостах в системе таким образом, чтобы все они могли работать вместе как единая СУБД. RT.Warehouse предоставляет свою собственную версию initdb под названием gpinitsystem, которая заботится об инициализации базы данных на инстансах мастера и всех сегментов и запуске каждого инстанса в правильном порядке.

После инициализации и запуска системы баз данных RT.Warehouse вы можете создавать базы данных и управлять ими, как в обычной СУБД PostgreSQL, подключившись к мастеру RT.Warehouse.

Инициализация RT.Warehouse

Для инициализации RT.Warehouse необходимо выполнить высокоуровневые задачи:

1. Убедитесь, что вы выполнили все задачи по установке, описанные в разделах 4 и 5.

Создайте файл хоста, содержащий адреса хостов ваших сегментов. См. п. 8.2.1.

Создайте файл конфигурации системы RT.Warehouse. См. п. 8.2.2.

По умолчанию RT.Warehouse инициализируется с использованием локали хоста мастера системы. Убедитесь, что это правильный языковой стандарт, который вы хотите использовать, поскольку некоторые параметры локали не могут быть изменены после инициализации. Дополнительные сведения см. в разделе 10.

Запустите утилиту инициализации RT.Warehouse на хосте мастера. См. п. 8.2.3.

Установите часовой пояс RT.Warehouse. См. п. 8.2.5.

Создание файла хоста инициализации

Утилите gpinitsystem требуется файл хоста, содержащий список адресов для каждого хоста сегмента. Утилита инициализации определяет количество инстансов сегмента на хост по количеству адресов хостов, перечисленных для каждого хоста, умноженному на количество местоположений каталогов данных, указанных в файле gpinitsystem\_config.

Этот файл должен содержать только адреса хостов сегмента (не мастера или резервного мастера). Для сегментных машин с несколькими несвязанными сетевыми интерфейсами в этом файле должны быть перечислены имена адресов хостов для каждого интерфейса — по одному в каждой строке.

|  |
| --- |
| **Примечание.**  Соглашение об именах хостов сегмента RT.Warehouse — sdwN, где sdw — префикс, а N — целое число. Например, sdw2 и т.д. Если хосты имеют несколько несвязанных сетевых адаптеров (NIC), по соглашению к имени хоста добавляется тире (-) и номер. Например, sdw1-1 и sdw1-2 — это два имени интерфейса для хоста sdw1. Однако для создания отказоустойчивой сети с балансировкой нагрузки рекомендуется использовать NIC. |

Чтобы создать файл хоста инициализации:

1. Войдите в систему как gpadmin.

|  |
| --- |
| $ su - gpadmin |

Создайте файл с именем hostfile\_gpinitsystem. В этом файле добавьте имя/имена адреса хоста интерфейсов хоста вашего сегмента, по одному имени в строке, без лишних строк или пробелов. Например, если у вас есть четыре хоста сегмента с двумя несвязанными сетевыми интерфейсами каждый:

|  |
| --- |
| sdw1-1  sdw1-2  sdw2-1  sdw2-2  sdw3-1  sdw3-2  sdw4-1  sdw4-2 |

Сохраните и закройте файл.

|  |
| --- |
| **Примечание.**  Если вы не уверены в именах хостов и/или именах адресов интерфейсов, используемых вашими машинами, загляните в файл /etc/hosts. |

Создание файла конфигурации RT.Warehouse

Файл конфигурации RT.Warehouse сообщает утилите gpinitsystem, как вы хотите настроить систему RT.Warehouse. Пример файла конфигурации можно найти в $GPHOME/docs/cli\_help/gpconfigs/gpinitsystem\_config.

Чтобы создать файл gpinitsystem\_config:

1. Войдите как gpadmin.

|  |
| --- |
| $ su - gpadmin |

Сделайте копию файла gpinitsystem\_config для использования в качестве отправной точки. Например:

|  |
| --- |
| $ cp $GPHOME/docs/cli\_help/gpconfigs/gpinitsystem\_config \  /home/gpadmin/gpconfigs/gpinitsystem\_config |

Откройте только что скопированный файл в текстовом редакторе.

Установите все необходимые параметры в соответствии с вашей средой. Система RT.Warehouse должна содержать инстанс мастера и как минимум два инстанса сегмента (даже при настройке системы с одним хостом).

Параметр DATA\_DIRECTORY определяет, сколько сегментов будет создано на каждом хосте. Если хосты ваших сегментов имеют несколько сетевых интерфейсов, и вы использовали их имена адресов интерфейсов в своём файле хоста, количество сегментов будет равномерно распределено по количеству доступных интерфейсов.

Чтобы указать PORT\_BASE, просмотрите диапазон портов, указанный в параметре net.ipv4.ip\_local\_port\_range в файле /etc/sysctl.conf.

Вот пример обязательных параметров в файле gpinitsystem\_config:

|  |
| --- |
| ARRAY\_NAME="Greenplum Data Platform"  SEG\_PREFIX=gpseg  PORT\_BASE=6000  declare -a DATA\_DIRECTORY=(/data1/primary /data1/primary /data1/primary /data2/primary /data2/primary /data2/primary)  MASTER\_HOSTNAME=mdw  MASTER\_DIRECTORY=/data/master  MASTER\_PORT=5432  TRUSTED SHELL=ssh  CHECK\_POINT\_SEGMENTS=8  ENCODING=UNICODE |

[Опционально] Если вы хотите развернуть зеркальные сегменты, раскомментируйте и установите параметры зеркалирования в соответствии с вашей средой. Чтобы указать MIRROR\_PORT\_BASE, просмотрите диапазон портов, указанный в параметре net.ipv4.ip\_local\_port\_range в файле /etc/sysctl.conf. Вот пример дополнительных параметров зеркала в файле gpinitsystem\_config:

|  |
| --- |
| MIRROR\_PORT\_BASE=7000  declare -a MIRROR\_DATA\_DIRECTORY=(/data1/mirror /data1/mirror /data1/mirror /data2/mirror /data2/mirror /data2/mirror) |

|  |
| --- |
| **Примечание.**  Вы можете инициализировать систему RT.Warehouse только с основными сегментами, а позже развернуть зеркала с помощью утилиты gpaddmirrors. |

Сохраните и закройте файл.

Запуск утилиты инициализации

Утилита gpinitsystem создаст систему RT.Warehouse, используя значения, определённые в файле конфигурации.

Запуск утилиты

Для запуска утилиты инициализации:

1. Выполните следующую команду, указав путь и имя файла конфигурации инициализации (gpinitsystem\_config) и файла хоста (hostfile\_gpinitsystem). Например:

|  |
| --- |
| $ cd ~  $ gpinitsystem -c gpconfigs/gpinitsystem\_config -h gpconfigs/hostfile\_gpinitsystem |

Для полностью резервированной системы (с резервным мастером и конфигурацией расширенного зеркала) укажите параметры -s и ‑S. Например:

|  |
| --- |
| $ gpinitsystem -c gpconfigs/gpinitsystem\_config -h gpconfigs/hostfile\_gpinitsystem \  -s standby\_master\_hostname -S |

Утилита проверит вашу информацию о настройке и убедится, что она может подключиться к каждому хосту и получить доступ к каталогам данных, указанным в вашей конфигурации. Если все предварительные проверки прошли успешно, утилита предложит вам подтвердить вашу конфигурацию. Например:

|  |
| --- |
| => Continue with Greenplum creation? Yy/Nn |

Нажмите y, чтобы начать инициализацию.

Затем утилита начнёт настройку и инициализацию инстанса мастера и каждого инстанса сегмента в системе. Каждый инстанс сегмента настраивается параллельно. В зависимости от количества сегментов этот процесс может занять некоторое время.

По завершении успешной установки утилита запустит вашу систему RT.Warehouse. Вы должны увидеть:

|  |
| --- |
| => RT.Warehouse instance successfully created. |

Устранение неполадок при инициализации

Если утилита обнаружит какие-либо ошибки при настройке инстанса, весь процесс завершится ошибкой и может оставить вас с частично созданной системой. Обратитесь к сообщениям об ошибках и журналам, чтобы определить причину сбоя и место в процессе сбоя. Файлы журнала создаются в ~/gpAdminLogs.

В зависимости от того, когда в процессе возникла ошибка, вам может потребоваться очистить её, а затем снова попробовать утилиту gpinitsystem. Например, если некоторые инстансы сегментов были созданы, а некоторые из них вышли из строя, вам может потребоваться остановить процессы postgres и удалить все каталоги данных, созданные утилитой, из ваших областей хранения данных. Скрипт возврата создаётся, чтобы при необходимости помочь с этой очисткой.

Использование скрипта возврата

Если утилита gpinitsystem выйдет из строя, она создаст следующий скрипт возврата, если она оставила вашу систему в частично установленном состоянии:

|  |
| --- |
| ~/gpAdminLogs/backout\_gpinitsystem\_<user>\_<timestamp> |

Вы можете использовать этот скрипт для очистки частично созданной системы RT.Warehouse. Этот скрипт возврата удалит все каталоги данных, созданные утилитой, процессы postgres и файлы журналов. После исправления ошибки, которая привела к сбою gpinitsystem, и запуска скрипта возврата, вы должны быть готовы повторить попытку инициализации массива RT.Warehouse.

В следующем примере показано, как запустить скрипта возврата:

|  |
| --- |
| $ sh backout\_gpinitsystem\_gpadmin\_20071031\_121053 |

Установка часового пояса RT.Warehouse

Рекомендуется настроить RT.Warehouse и хост-системы на использование известного поддерживаемого часового пояса. RT.Warehouse использует часовой пояс из набора внутренних часовых поясов PostgreSQL. Установка часового пояса RT.Warehouse не позволяет RT.Warehouse выбирать часовой пояс при каждом перезапуске кластера и устанавливает часовой пояс для инстансов мастера и сегментов.

Используйте утилиту gpconfig, чтобы показать и установить часовой пояс RT.Warehouse. Например, эти команды показывают часовой пояс RT.Warehouse и устанавливают часовой пояс US/Pacific.

|  |
| --- |
| # gpconfig -s TimeZone  # gpconfig -c TimeZone -v 'US/Pacific' |

После изменения часового пояса необходимо перезапустить RT.Warehouse. Команда gpstop -ra перезапускает RT.Warehouse. Представление каталога pg\_timezone\_names предоставляет информацию о часовом поясе RT.Warehouse.

Дополнительные сведения о часовом поясе RT.Warehouse см. в разделе 10.

Установка переменных сред RT.Warehouse

Вы должны настроить свою среду на мастере RT.Warehouse (и резервном мастере). В вашем каталоге $GPHOME находится файл greenplum\_path.sh с настройками переменных среды для RT.Warehouse. Вы можете получить этот файл в профиле оболочки запуска пользователя gpadmin (например, .bashrc).

Утилиты управления RT.Warehouse также требуют, чтобы была установлена ​​переменная среда MASTER\_DATA\_DIRECTORY. Это должно указывать на каталог, созданный утилитой gpinitsystem в расположении каталога основных данных.

|  |
| --- |
| **Примечание.**  Скрипт greenplum\_path.sh изменяет операционную среду для поддержки работы утилит, специфичных для RT.Warehouse. Эти же изменения в среде могут негативно повлиять на работу других утилит системного уровня, таких как ps или yum. Используйте отдельные учётные записи для выполнения системного администрирования и администрирования базы данных, вместо того, чтобы пытаться выполнять обе функции как gpadmin. |

Чтобы настроить пользовательскую среду для RT.Warehouse:

1. Убедитесь, что вы вошли как gpadmin:

|  |
| --- |
| $ su - gpadmin |

Откройте файл профиля (например, .bashrc) в текстовом редакторе. Например:

|  |
| --- |
| $ vi ~/.bashrc |

Добавьте в этот файл строки для исходного файла greenplum\_path.sh и установите переменную среды MASTER\_DATA\_DIRECTORY. Например:

|  |
| --- |
| source /usr/local/greenplum-db/greenplum\_path.sh  export MASTER\_DATA\_DIRECTORY=/data/master/gpseg-1 |

[Опционально] Вы также можете установить некоторые переменные среды клиентского сеанса, такие как PGPORT, PGUSER и PGDATABASE для удобства. Например:

|  |
| --- |
| export PGPORT=5432  export PGUSER=gpadmin export PGDATABASE=default\_login\_database\_name |

[Опционально] Если вы используете РЕД ОС 7, добавьте следующую строку в конец файла .bashrc, чтобы включить использование команды ps в среде greenplum\_path.sh:

|  |
| --- |
| export LD\_PRELOAD=/lib64/libz.so.1 ps |

Сохраните и закройте файл.

После редактирования файла профиля загрузите его, чтобы изменения вступили в силу. Например:

|  |
| --- |
| $ source ~/.bashrc |

Если у вас есть хост резервного мастера, скопируйте файл среды также на резервный мастер. Например:

|  |
| --- |
| $ cd ~  $ scp .bashrc standby\_hostname:`pwd` |

|  |
| --- |
| **Примечание.**  Файл .bashrc не должен выводить никаких сообщений. Если вы хотите, чтобы сообщения отображались для пользователей при входе в систему, используйте вместо этого файл .profile. |

Следующие шаги

После того, как ваша система настроена и работает, выполните шаги, представленные ниже.

Разрешение клиентских подключений

После первой инициализации RT.Warehouse она будет разрешать только локальные подключения к базе данных из роли gpadmin (или любого другого системного пользователя, запускающего gpinitsystem). Если вы хотите, чтобы другие пользователи или клиентские машины могли подключаться к RT.Warehouse.

Создание баз данных и загрузка данных

После проверки вашей установки вы можете начать создавать базы данных и загружать данные.

УСТАНОВКА ДОПОЛНИТЕЛЬНЫХ ПОСТАВЛЯЕМЫХ МОДУЛЕЙ

Дистрибутив RT.Warehouse включает несколько модулей contrib, полученных из PostgreSQL и RT.Warehouse, которые вы можете установить.

Каждый модуль обычно упаковывается как расширение RT.Warehouse. Вы должны зарегистрировать эти модули в каждой базе данных, в которой хотите их использовать. Например, чтобы зарегистрировать модуль dblink в базе данных с именем testdb, используйте команду:

|  |
| --- |
| $ psql -d testdb -c 'CREATE EXTENSION dblink;' |

Чтобы удалить модуль из базы данных, сбросьте соответствующее расширение. Например, чтобы удалить модуль dblink из базы данных testdb:

|  |
| --- |
| $ psql -d testdb -c 'DROP EXTENSION dblink;' |

|  |
| --- |
| **Примечание.**  Когда вы удаляете расширение модуля из базы данных, любая пользовательская функция, созданная вами в базе данных, которая ссылается на функции, определенные в модуле, больше не будет работать. Если вы создали какие-либо объекты базы данных, использующие типы данных, определённые в модуле, RT.Warehouse уведомит вас об этих зависимостях, когда вы попытаетесь удалить расширение модуля. |

Таким образом вы можете зарегистрировать следующие модули:

citext;

dblink;

fuzzystrmatch;

gp\_sparse\_vector;

hstore;

pageinspect;

pgcrypto;

sslinfo.

НАСТРОЙКА ЧАСОВОГО ПОЯСА И ПАРАМЕТРОВ ЛОКАЛИЗАЦИИ

Раздел описывает доступные часовые пояса и функции локализации RT.Warehouse.

Настройка часового пояса

RT.Warehouse выбирает для использования часовой пояс из набора внутренних часовых поясов PostgreSQL. Доступные часовые пояса PostgreSQL берутся из базы данных часовых поясов Internet Assigned Numbers Authority (IANA), а RT.Warehouse при необходимости обновляет свой список доступных часовых поясов, когда база данных IANA изменяется на PostgreSQL.

RT.Warehouse выбирает часовой пояс, сопоставляя часовой пояс PostgreSQL со значением параметра конфигурации сервера TimeZone или часового пояса хост-системы, если TimeZone не установлен. Например, при выборе часового пояса по умолчанию из часового пояса хост-системы RT.Warehouse использует алгоритм для выбора часового пояса PostgreSQL на основе файлов часовых поясов хост-системы. Если системный часовой пояс включает информацию о секундах координации, RT.Warehouse не может сопоставить системный часовой пояс с часовым поясом PostgreSQL. В этом случае база данных RT.Warehouse вычисляет «наилучшее совпадение» с часовым поясом PostgreSQL на основе информации от хост-системы.

Рекомендуется настроить RT.Warehouse и хост-системы на использование известного поддерживаемого часового пояса. Это устанавливает часовой пояс для инстансов мастера и сегментов RT.Warehouse и предотвращает выбор RT.Warehouse наиболее подходящего часового пояса при каждом перезапуске кластера с использованием текущего часового пояса системы и файлов часовых поясов RT.Warehouse (которые могли быть обновлены из базы данных IANA. с момента последнего перезапуска). Используйте утилиту gpconfig, чтобы показать и установить часовой пояс RT.Warehouse. Например, эти команды показывают часовой пояс RT.Warehouse и устанавливают часовой пояс US/Pacific.

|  |
| --- |
| # gpconfig -s TimeZone  # gpconfig -c TimeZone -v 'US/Pacific' |

После изменения часового пояса необходимо перезапустить RT.Warehouse. Команда gpstop -ra перезапускает RT.Warehouse. Представление каталога pg\_timezone\_names предоставляет информацию о часовом поясе RT.Warehouse.

Поддержка локализации в RT.Warehouse

RT.Warehouse поддерживает локализацию двумя способами:

1. Использование функций локализации операционной системы для обеспечения порядка сортировки, форматирования чисел и т. д.

Предоставление ряда различных наборов символов, определённых на сервере RT.Warehouse, включая многобайтовые наборы символов, для поддержки хранения текста на всех типах языков и обеспечения перевода набора символов между клиентом и сервером.

Поддержка локализации относится к приложению, уважающему культурные предпочтения в отношении алфавитов, сортировки, форматирования чисел и т. д. RT.Warehouse использует стандартные средства локализации ISO C и POSIX, предоставляемые серверной операционной системой. За дополнительной информацией обращайтесь к документации вашей операционной системы.

Поддержка локализации автоматически инициализируется при инициализации системы RT.Warehouse. Утилита инициализации gpinitsystem по умолчанию инициализирует массив RT.Warehouse с настройкой локализации среды выполнения, поэтому, если ваша система уже настроена на использование языкового стандарта, который вы хотите в своей системе RT.Warehouse, вам больше ничего не нужно делать.

Когда вы готовы запустить RT.Warehouse и хотите использовать другой языковой стандарт (или вы не уверены, какой языковой стандарт установлен в вашей системе), вы можете указать gpinitsystem, какой именно языковой стандарт использовать, указав параметр -n *locale*. Например:

|  |
| --- |
| $ gpinitsystem -c gp\_init\_config -n sv\_SE |

См. Инициализация RT.Warehouse для получения информации о процессе инициализации базы данных.

В приведённом выше примере в качестве языкового стандарта устанавливается шведский (sv), используемый в Швеции (SE). Другими возможными вариантами могут быть en\_US (американский английский) и fr\_CA (французско-канадский). Если для языкового стандарта может быть полезно более одного набора символов, то спецификации выглядят следующим образом: cs\_CZ.ISO8859-2. Какие локали и под какими именами доступны в вашей системе, зависит от того, что было предоставлено поставщиком операционной системы и что было установлено. В большинстве систем команда locale -a предоставит список доступных локалей.

Иногда полезно смешивать правила из нескольких локалей, например, использовать английские правила сортировки, но испанские сообщения. Для поддержки этого существует набор подкатегорий локали, которые контролируют только определённый аспект правил локализации:

LC\_COLLATE — порядок сортировки строк;

LC\_CTYPE — классификация символов;

LC\_MESSAGES — язык сообщений;

LC\_MONETARY — форматирование денежных сумм;

LC\_NUMERIC — форматирование чисел;

LC\_TIME — форматирование даты и времени.

Если вы хотите, чтобы система вела себя так, как если бы она не поддерживала языковой стандарт, используйте специальный языковой стандарт C или POSIX.

Природа некоторых категорий локали заключается в том, что их значение должно быть фиксированным на протяжении всего срока службы системы RT.Warehouse. То есть, как только gpinitsystem запустится, вы больше не сможете их изменить. LC\_COLLATE и LC\_CTYPE — это те категории. Они влияют на порядок сортировки индексов, поэтому они должны оставаться фиксированными, иначе индексы в текстовых столбцах будут повреждены. RT.Warehouse обеспечивает это путём записи значений LC\_COLLATE и LC\_CTYPE, которые видит gpinitsystem. Сервер автоматически принимает эти два значения в зависимости от языкового стандарта, который был выбран во время инициализации.

Другие категории языковых стандартов могут быть изменены по желанию всякий раз, когда сервер запущен, путём установки параметров конфигурации сервера, которые имеют то же имя, что и категории языковых стандартов. Значения по умолчанию, выбранные gpinitsystem, записываются в файлы конфигурации postgresql.conf мастера и сегментов и используются в качестве значений по умолчанию при запуске системы RT.Warehouse. Если вы удалите эти назначения из файлов postgresql.conf мастера и каждого сегмента, сервер унаследует настройки из своей среды выполнения.

Обратите внимание, что поведение сервера в локали определяется переменными средами, которые видит сервер, а не средой какого-либо клиента. Поэтому будьте осторожны, чтобы настроить правильные параметры локали на каждом хосте RT.Warehouse (мастере и сегментах) перед запуском системы. Следствием этого является то, что если клиент и сервер настроены в разных регионах, сообщения могут отображаться на разных языках в зависимости от того, откуда они были отправлены.

Наследование языкового стандарта из среды выполнения означает в большинстве ОС следующее: для данной категории языкового стандарта, скажем, сопоставления, следующие переменные среды проверяются в этом порядке, пока не будет обнаружена одна из установленных: LC\_ALL, LC\_COLLATE (переменная, соответствующая соответствующая категория), LANG. Если ни одна из этой переменной среды не задана, по умолчанию используется языковой стандарт C.

Некоторые библиотеки локализации сообщений также обращаются к переменной среде LANGUAGE, которая переопределяет все другие настройки локали с целью установки языка сообщений. В случае сомнений обратитесь к документации по вашей операционной системе, в частности к документации по gettext, для получения дополнительной информации.

Native language support (NLS), которая позволяет переводить сообщения на предпочтительный язык пользователя, не включена в RT.Warehouse для языков, отличных от английского. Это не зависит от поддержки других локалей.

Влияние локализации

Настройки локали влияют на следующие функции SQL:

Порядок сортировки в запросах с использованием ORDER BY для текстовых данных.

Возможность использовать индексы с условием LIKE.

Функции upper, lower и initcap.

Семейство функций to\_char.

Недостатком использования других локалей, кроме C или POSIX, в RT.Warehouse является их влияние на производительность. Это замедляет обработку символов и предотвращает использование LIKE обычных индексов. По этой причине используйте локали только в том случае, если они вам действительно нужны.

Устранение неполадок в регионах

Если поддержка языкового стандарта не работает должным образом, убедитесь, что поддержка языкового стандарта в вашей операционной системе настроена правильно. Чтобы проверить, какие языковые стандарты установлены в вашей системе, вы можете использовать команду locale -a, если она предусмотрена вашей операционной системой.

Убедитесь, что RT.Warehouse действительно использует тот языковой стандарт, который, по вашему мнению, используется. Параметры LC\_COLLATE и LC\_CTYPE определяются во время инициализации и не могут быть изменены без перенастройки gpinitsystem. Другие настройки локали, включая LC\_MESSAGES и LC\_MONETARY, изначально определяются средой операционной системы хостов мастера и/или сегмента, но могут быть изменены после инициализации путём редактирования файла postgresql.conf каждого инстанса мастера RT.Warehouse и инстанса сегмента. Вы можете проверить активные настройки локали хоста мастера с помощью команды SHOW. Обратите внимание, что каждый хост в вашем массиве RT.Warehouse должен использовать одинаковые настройки локали.

Поддержка набора символов

Поддержка набора символов в RT.Warehouse позволяет хранить текст в различных наборах символов, включая однобайтовые наборы символов, такие как серия ISO 8859, и многобайтовые наборы символов, такие как EUC (Extended Unix Code), UTF-8 и внутренний код Mule. Все поддерживаемые наборы символов могут прозрачно использоваться клиентами, но некоторые из них не поддерживаются для использования на сервере (то есть как кодирование на стороне сервера). Набор символов по умолчанию выбирается при инициализации массива RT.Warehouse с помощью gpinitsystem. Его можно переопределить при создании базы данных, поэтому у вас может быть несколько баз данных, каждая с различным набором символов.

Таблица 4 — Наборы символов RT.Warehouse

| Наименование | Описание | Язык | Сервер? | Байт/  символ | Алиасы |
| --- | --- | --- | --- | --- | --- |
| BIG5 | Big Five | Traditional Chinese | Нет | 1-2 | WIN950, Windows950 |
| EUC\_CN | Extended UNIX Code-CN | Simplified Chinese | Да | 1-3 |  |
| EUC\_JP | Extended UNIX Code-JP | Japanese | Да | 1-3 |  |
| EUC\_KR | Extended UNIX Code-KR | Korean | Да | 1-3 |  |
| EUC\_TW | Extended UNIX Code-TW | Traditional Chinese, Taiwanese | Да | 1-3 |  |
| GB18030 | National Standard | Chinese | Нет | 1-2 |  |
| GBK | Extended National Standard | Simplified Chinese | Нет | 1-2 | WIN936, Windows936 |
| ISO\_8859\_5 | ISO 8859-5, ECMA 113 | Latin/Cyrillic | Да | 1 |  |
| ISO\_8859\_6 | ISO 8859-6, ECMA 114 | Latin/Arabic | Да | 1 |  |
| ISO\_8859\_7 | ISO 8859-7, ECMA 118 | Latin/Greek | Да | 1 |  |
| ISO\_8859\_8 | ISO 8859-8, ECMA 121 | Latin/Hebrew | Да | 1 |  |
| JOHAB | JOHA | Korean (Hangul) | Да | 1-3 |  |
| KOI8 | KOI8-R(U) | Cyrillic | Да | 1 | KOI8R |
| LATIN1 | ISO 8859-1, ECMA 94 | Western European | Да | 1 | ISO88591 |
| LATIN2 | ISO 8859-2, ECMA 94 | Central European | Да | 1 | ISO88592 |
| LATIN3 | ISO 8859-3, ECMA 94 | South European | Да | 1 | ISO88593 |
| LATIN4 | ISO 8859-4, ECMA 94 | North European | Да | 1 | ISO88594 |
| LATIN5 | ISO 8859-9, ECMA 128 | Turkish | Да | 1 | ISO88599 |
| LATIN6 | ISO 8859-10, ECMA 144 | Nordic | Да | 1 | ISO885910 |
| LATIN7 | ISO 8859-13 | Baltic | Да | 1 | ISO885913 |
| LATIN8 | ISO 8859-14 | Celtic | Да | 1 | ISO885914 |
| LATIN9 | ISO 8859-15 | LATIN1 with Euro and accents | Да | 1 | ISO885915 |
| LATIN10 | ISO 8859-16, ASRO SR 14111 | Romanian | Да | 1 | ISO885916 |
| MULE\_INTERNAL | Mule internal code | Multilingual Emacs | Да | 1-4 |  |
| SJIS | Shift JIS | Japanese | Нет | 1-2 | Mskanji, ShiftJIS, WIN932, Windows932 |
| SQL\_ASCII | Unspecified[[1]](#footnote-1) | Any | Нет | 1 |  |
| UHC | Unified Hangul Code | Korean | Нет | 1-2 | WIN949, Windows949 |
| UTF8 | Unicode, 8-bit | All | Да | 1-4 | Unicode |
| WIN866 | Windows CP866 | Cyrillic | Да | 1 | ALT |
| WIN874 | Windows CP874 | Thai | Да | 1 |  |
| WIN1250 | Windows CP1250 | Central European | Да | 1 |  |
| WIN1251 | Windows CP1251 | Cyrillic | Да | 1 | WIN |
| WIN1252 | Windows CP1252 | Western European | Да | 1 |  |
| WIN1253 | Windows CP1253 | Greek | Да | 1 |  |
| WIN1254 | Windows CP1254 | Turkish | Да | 1 |  |
| WIN1255 | Windows CP1255 | Hebrew | Да | 1 |  |
| WIN1256 | Windows CP1256 | Arabic | Да | 1 |  |
| WIN1257 | Windows CP1257 | Baltic | Да | 1 |  |
| WIN1258 | Windows CP1258 | Vietnamese | Да | 1 | ABC, TCVN, TCVN5712, VSCII |

Установка набора символов

gpinitsystem определяет набор символов по умолчанию для системы RT.Warehouse, считывая значение параметра ENCODING в файле gp\_init\_config во время инициализации. Набор символов по умолчанию — UNICODE или UTF8.

Вы можете создать базу данных с другим набором символов, помимо того, который используется по умолчанию для всей системы. Например:

|  |
| --- |
| => CREATE DATABASE korean WITH ENCODING 'EUC\_KR'; |

|  |
| --- |
| **Внимание.**  Хотя вы можете указать любую кодировку для базы данных, неразумно выбирать кодировку, которая не соответствует выбранной вами локали. Параметры LC\_COLLATE и LC\_CTYPE подразумевают определённую кодировку, а операции, зависящие от языкового стандарта (такие как сортировка), могут неправильно интерпретировать данные, которые находятся в несовместимой кодировке. |

Поскольку эти настройки локали заморожены gpinitsystem, очевидная гибкость использования различных кодировок в разных базах данных является скорее теоретической, чем реальной.

Один из способов безопасного использования нескольких кодировок — это установить языковой стандарт на C или POSIX во время инициализации, тем самым отключив любую реальную осведомлённость о языковых стандартах.

Преобразование набора символов между сервером и клиентом

RT.Warehouse поддерживает автоматическое преобразование набора символов между сервером и клиентом для определённых комбинаций набора символов. Информация о преобразовании хранится в таблице системного каталога мастера pg\_conversion. RT.Warehouse поставляется с некоторыми предопределёнными преобразованиями, или вы можете создать новое преобразование с помощью команды SQL CREATE CONVERSION.

Таблица 5 — Преобразование набора символов клиент/сервер

| Набор символов сервера | Доступные наборы символов клиента |
| --- | --- |
| BIG5 | не поддерживается как серверная кодировка |
| EUC\_CN | EUC\_CN, MULE\_INTERNAL, UTF8 |
| EUC\_JP | EUC\_JP, MULE\_INTERNAL, SJIS, UTF8 |
| EUC\_KR | EUC\_KR, MULE\_INTERNAL, UTF8 |
| EUC\_TW | EUC\_TW, BIG5, MULE\_INTERNAL, UTF8 |
| GB18030 | не поддерживается как серверная кодировка |
| GBK | не поддерживается как серверная кодировка |
| ISO\_8859\_5 | ISO\_8859\_5, KOI8, MULE\_INTERNAL, UTF8, WIN866, WIN1251 |
| ISO\_8859\_6 | ISO\_8859\_6, UTF8 |
| ISO\_8859\_7 | ISO\_8859\_7, UTF8 |
| ISO\_8859\_8 | ISO\_8859\_8, UTF8 |
| JOHAB | JOHAB, UTF8 |
| KOI8 | KOI8, ISO\_8859\_5, MULE\_INTERNAL, UTF8, WIN866, WIN1251 |
| LATIN1 | LATIN1, MULE\_INTERNAL, UTF8 |
| LATIN2 | LATIN2, MULE\_INTERNAL, UTF8, WIN1250 |
| LATIN3 | LATIN3, MULE\_INTERNAL, UTF8 |
| LATIN4 | LATIN4, MULE\_INTERNAL, UTF8 |
| LATIN5 | LATIN5, UTF8 |
| LATIN6 | LATIN6, UTF8 |
| LATIN7 | LATIN7, UTF8 |
| LATIN8 | LATIN8, UTF8 |
| LATIN9 | LATIN9, UTF8 |
| LATIN10 | LATIN10, UTF8 |
| MULE\_INTERNAL | MULE\_INTERNAL, BIG5, EUC\_CN, EUC\_JP, EUC\_KR, EUC\_TW, ISO\_8859\_5, KOI8, LATIN1 to LATIN4, SJIS, WIN866, WIN1250, WIN1251 |
| SJIS | не поддерживается как серверная кодировка |
| SQL\_ASCII | не поддерживается как серверная кодировка |
| UHC | не поддерживается как серверная кодировка |
| UTF8 | все поддерживаемые кодировки |
| WIN866 | WIN866 |
| ISO\_8859\_5 | KOI8, MULE\_INTERNAL, UTF8, WIN1251 |
| WIN874 | WIN874, UTF8 |
| WIN1250 | WIN1250, LATIN2, MULE\_INTERNAL, UTF8 |
| WIN1251 | WIN1251, ISO\_8859\_5, KOI8, MULE\_INTERNAL, UTF8, WIN866 |
| WIN1252 | WIN1252, UTF8 |
| WIN1253 | WIN1253, UTF8 |
| WIN1254 | WIN1254, UTF8 |
| WIN1255 | WIN1255, UTF8 |
| WIN1256 | WIN1256, UTF8 |
| WIN1257 | WIN1257, UTF8 |
| WIN1258 | WIN1258, UTF8 |

Чтобы включить автоматическое преобразование набора символов, вы должны указать RT.Warehouse набор символов (кодировку), который вы хотите использовать в клиенте. Для этого есть несколько способов:

1. Использование команды \encoding в psql, которая позволяет вам на лету изменять клиентскую кодировку.

Использование SET client\_encoding TO. Установить клиентскую кодировку можно с помощью этой команды SQL:

|  |
| --- |
| => SET CLIENT\_ENCODING TO 'value'; |

Чтобы запросить текущую клиентскую кодировку:

|  |
| --- |
| => SHOW client\_encoding; |

Чтобы вернуться к кодировке по умолчанию:

|  |
| --- |
| => RESET client\_encoding; |

Использование переменной среды PGCLIENTENCODING. Когда PGCLIENTENCODING определён в клиентской среде, эта клиентская кодировка автоматически выбирается при подключении к серверу. (Впоследствии это можно изменить, используя любой из других методов, упомянутых выше.)

Установка параметра конфигурации client\_encoding. Если client\_encoding установлен в файле мастера postgresql.conf, эта клиентская кодировка автоматически выбирается при подключении к RT.Warehouse. (Впоследствии это можно изменить, используя любой из других методов, упомянутых выше.)

Если преобразование определённого символа невозможно — предположим, вы выбрали EUC\_JP для сервера и LATIN1 для клиента, тогда некоторые японские символы не имеют представления в LATIN1 — тогда выдаётся сообщение об ошибке.

Если набор символов клиента определён как SQL\_ASCII, преобразование кодировки отключено, независимо от набора символов сервера. Использование SQL\_ASCII неразумно, если вы не работаете с данными, состоящими только из ASCII. SQL\_ASCII не поддерживается в качестве кодировки сервера.

ОБНОВЛЕНИЕ БОЛЕЕ РАННЕЙ ВЕРСИИ RT.WAREHOUSE 6

Путь обновления, поддерживаемый для данного релиза, — RT.Warehouse 6.x до более новой версии RT.Warehouse 6.x.

|  |
| --- |
| **Внимание.**  Установите часовой пояс RT.Warehouse на значение, совместимое с вашими хост-системами. Установка часового пояса RT.Warehouse не позволяет RT.Warehouse выбирать часовой пояс при каждом перезапуске кластера и устанавливает часовой пояс для инстансов мастера и сегмента. После обновления до данного релиза, если вы не установили значение часового пояса RT.Warehouse, убедитесь, что выбранный часовой пояс RT.Warehouse приемлем для вашего развёртывания. Дополнительные сведения см. в разделе 10. |

Предварительные условия

Перед запуском процесса обновления выполните следующие проверки:

1. Проверьте работоспособность оборудования хоста RT.Warehouse и убедитесь, что хосты соответствуют требованиям для работы RT.Warehouse. Утилита RT.Warehouse gpcheckperf может помочь вам подтвердить требования к хосту.

|  |
| --- |
| **Примечание.**  Если вам нужно запустить утилиту gpcheckcat, запустите её за несколько недель до обновления во время периода обслуживания. При необходимости вы можете решить любые проблемы, обнаруженные утилитой, до запланированного обновления. |

Утилита находится в $GPHOME/bin. Переведите RT.Warehouse в ограниченный режим при запуске утилиты gpcheckcat.

Если gpcheckcat сообщает о несоответствиях в каталоге, вы можете запустить gpcheckcat с параметром -g, чтобы сгенерировать скрипты SQL для исправления несоответствий.

После запуска скриптов SQL снова запустите gpcheckcat. Возможно вам придётся повторить процесс запуска gpcheckcat и создания скриптов SQL, чтобы убедиться в отсутствии несоответствий. Запустите скрипты SQL, созданные gpcheckcat в неактивной системе. Утилита может сообщать ложные предупреждения, если в системе есть активность.

В процессе миграции с RT.Warehouse 6 создается резервная копия некоторых файлов и каталогов в $MASTER\_DATA\_DIRECTORY. Вручную сделайте резервную копию и удалите все оставшиеся файлы и каталоги из каталога $MASTER\_DATA\_DIRECTORY перед переносом.

Если вы настроили Platform Extension Framework (PXF) в предыдущей установке RT.Warehouse, вы должны остановить службу PXF, и вам может потребоваться создать резервную копию файлов конфигурации PXF перед обновлением до новой версии RT.Warehouse.

Если вы еще не настроили PXF, никаких действий не требуется.

Обновление 6.x до более новой версии 6.x

Обновление RT.Warehouse 6.x до более новой версии 6.x предполагает остановку RT.Warehouse, обновление бинарных файлов программного обеспечения RT.Warehouse и перезапуск RT.Warehouse. Если вы используете пакеты расширения RT.Warehouse, существуют дополнительные требования. См. Предварительные условия.

1. Войдите на хост мастера RT.Warehouse как администратор RT.Warehouse:

|  |
| --- |
| => RESET client\_encoding; |

Выполните интеллектуальное выключение вашей системы RT.Warehouse 6.x (активные подключения к базе данных могут отсутствовать). В этом примере для отключения запросов на подтверждение используется параметр –a:

|  |
| --- |
| $ gpstop -a |

Скопируйте новый установочный пакет программного обеспечения RT.Warehouse в домашний каталог пользователя gpadmin на хосты мастера, резервного мастера и сегментов.

От имени пользователя root установите новую версию ПО RT.Warehouse на каждом хосте.

Например, в системах РЕД ОС:

|  |
| --- |
| $ sudo yum install greenplum-db-<version>-<platform>.rpm |

В системах Ubuntu:

|  |
| --- |
| # apt install ./greenplum-db-<version>-<platform>.deb |

Обновите разрешения для новой установки. Например, запустите эту команду от имени пользователя root, чтобы изменить пользователя и группу установленных файлов на gpadmin.

|  |
| --- |
| $ sudo chown -R gpadmin:gpadmin /usr/local/greenplum\* |

При необходимости обновите файл greenplum\_path.sh на хостах мастера и резервного мастера для использования с вашей конкретной установкой. Вот несколько примеров.

Если RT.Warehouse использует аутентификацию LDAP, отредактируйте файл greenplum\_path.sh, добавив строку:

|  |
| --- |
| export LDAPCONF=/etc/openldap/ldap.conf |

Если RT.Warehouse использует PL/Java, вам может потребоваться установить или обновить переменные среды JAVA\_HOME и LD\_LIBRARY\_PATH в greenplum\_path.sh.

|  |
| --- |
| **Примечание.**  При сравнении предыдущего и нового файлов greenplum\_path.sh имейте в виду, что при установке некоторых расширений RT.Warehouse также обновляется файл greenplum\_path.sh. greenplum\_path.sh из предыдущего выпуска может содержать обновления, которые были результатом установки этих расширений. |

Измените среду суперпользователя RT.Warehouse (gpadmin) и убедитесь, что вы используете файл greenplum\_path.sh для новой установки. Например, измените следующую строку в .bashrc или выбранном вами файле профиля:

|  |
| --- |
| source /usr/local/greenplum-db-<current\_version>/greenplum\_path.sh |

на:

|  |
| --- |
| source /usr/local/greenplum-db-<new\_version>/greenplum\_path.sh |

Или, если вы используете символическую ссылку (/usr/local/greenplum-db) в файлах вашего профиля, обновите ссылку, чтобы она указывала на недавно установленную версию. Например:

|  |
| --- |
| $ rm /usr/local/greenplum-db  $ ln -s /usr/local/greenplum-db-<new\_version> /usr/local/greenplum-db |

Загрузите файл среды, который вы только что отредактировали. Например:

|  |
| --- |
| $ source ~/.bashrc |

После обновления всех хостов сегмента войдите в систему как пользователь gpadmin и перезапустите систему RT.Warehouse:

|  |
| --- |
| # su - gpadmin  $ gpstart |

Если вы настроили PXF в предыдущей установке RT.Warehouse, необходимо повторно инициализировать службу PXF после обновления RT.Warehouse.

После обновления RT.Warehouse убедитесь, что все функции работают должным образом. Например, проверьте, что резервное копирование и восстановление выполняются должным образом, а функции RT.Warehouse, такие как определяемые пользователем функции, и расширения, такие как MADlib и PostGIS, работают должным образом.

[ОПЦИОНАЛЬНО] ВКЛЮЧЕНИЕ IPTABLES

В системах Linux вы можете настроить и включить файрвол iptables для работы с RT.Warehouse.

|  |
| --- |
| **Примечание.**  Если включён iptables, производительность RT.Warehouse может снизиться. Вы должны протестировать производительность вашего приложения с включенным iptables, чтобы убедиться, что производительность приемлема. |

Для получения дополнительной информации о iptables см. Отключение программного обеспечения SELinux и файрвола.

Включение iptables

1. Как gpadmin запустите эту команду на хосте мастера RT.Warehouse, чтобы остановить RT.Warehouse:

|  |
| --- |
| $ gpstop -a |

На хостах RT.Warehouse:

Обновите файл /etc/sysconfig/iptables в соответствии с Пример правил iptables.

Как пользователь root выполните следующие команды, чтобы включить iptables:

|  |
| --- |
| # chkconfig iptables on  # service iptables start |

Как gpadmin выполните эту команду на хотсе мастера RT.Warehouse, чтобы запустить RT.Warehouse:

|  |
| --- |
| $ gpstart -a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Внимание.**  После включения iptables эта ошибка в файле /var/log/messages указывает, что параметр для таблицы iptables слишком низкий и его необходимо увеличить.   |  | | --- | | ip\_conntrack: table full, dropping packet. |   От имени пользователя root запустите эту команду, чтобы просмотреть значение таблицы iptables:   |  | | --- | | # sysctl net.ipv4.netfilter.ip\_conntrack\_max |   Чтобы рабочая нагрузка RT.Warehouse не переполняла таблицу iptables, установите для неё следующее значение:   |  | | --- | | # sysctl net.ipv4.netfilter.ip\_conntrack\_max=6553600 |   Значение может потребоваться скорректировать для ваших хостов. Чтобы сохранить значение после перезагрузки, вы можете обновить файл /etc/sysctl.conf, как описано в Рекомендуемые настройки параметров ОС. |

Пример правил iptables

Когда iptables включён, iptables управляет IP-коммуникацией в хост-системе на основе настроек конфигурации (правил). Примеры правил используются для настройки iptables для хостов мастера RT.Warehouse, резервного мастера и хостов сегментов.

1. Пример правил iptables мастера и резервного мастера.

Пример правил iptables хоста сегмента.

Два набора правил учитывают различные типы обмена данными, которые RT.Warehouse ожидает от мастера (основного и резервного) и хостов сегментов. Правила следует добавить в файл /etc/sysconfig/iptables хостов RT.Warehouse. Для RT.Warehouse правила iptables должны разрешать следующий обмен данными:

1. Для связи клиента с мастер-сервером RT.Warehouse разрешите как минимум postgres и 28080 (интерфейс eth1 в примере).

Для системного интерконнекта с RT.Warehouse разрешите обмен данными с использованием протоколов tcp, udp и icmp (интерфейсы eth4 и eth5 в примере).

Сетевые интерфейсы, которые вы указываете в настройках iptables, являются интерфейсами для хостов RT.Warehouse, которые вы указываете в файле hostfile\_gpinitsystem. Этот файл указывается при запуске команды gpinitsystem для инициализации системы RT.Warehouse. См. раздел 8 для получения информации о файле hostfile\_gpinitsystem и команде gpinitsystem.

Для административной сети на RT.Warehouse DCA разрешите обмен данными с использованием протоколов ssh, ntp и icmp (интерфейс eth0 в примере).

В файле iptables каждая команда правила добавления (строки, начинающиеся с -A) представляют собой одну строку.

Примеры правил следует скорректировать в соответствии с вашей конфигурацией. Например:

1. Команда добавления, строки -A и параметр соединения -i должны соответствовать коннекторам для ваших хостов.

Информация о сетевой маске CIDR для параметра источника –s должна соответствовать IP-адресам вашей сети.

Пример правил iptables для мастера и резервного мастера

Пример правил iptables с комментариями для файла /etc/sysconfig/iptables на хосте мастера RT.Warehouse и хосте резервного мастера.

|  |
| --- |
| \*filter  # Following 3 are default rules. If the packet passes through  # the rule set it gets these rule.  # Drop all inbound packets by default.  # Drop all forwarded (routed) packets.  # Let anything outbound go through.  :INPUT DROP [0:0]  :FORWARD DROP [0:0]  :OUTPUT ACCEPT [0:0]  # Accept anything on the loopback interface.  -A INPUT -i lo -j ACCEPT  # If a connection has already been established allow the  # remote host packets for the connection to pass through.  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  # These rules let all tcp and udp through on the standard  # interconnect IP addresses and on the interconnect interfaces.  # NOTE: gpsyncmaster uses random tcp ports in the range 1025 to 65535  # and RT.Warehouse uses random udp ports in the range 1025 to 65535.  -A INPUT -i eth4 -p udp -s 192.0.2.0/22 -j ACCEPT  -A INPUT -i eth5 -p udp -s 198.51.100.0/22 -j ACCEPT  -A INPUT -i eth4 -p tcp -s 192.0.2.0/22 -j ACCEPT --syn -m state --state NEW  -A INPUT -i eth5 -p tcp -s 198.51.100.0/22 -j ACCEPT --syn -m state --state NEW  # Allow udp/tcp ntp connections on the admin network on Greenplum DCA.  -A INPUT -i eth0 -p udp --dport ntp -s 203.0.113.0/21 -j ACCEPT  -A INPUT -i eth0 -p tcp --dport ntp -s 203.0.113.0/21 -j ACCEPT --syn -m state --state NEW  # Allow ssh on all networks (This rule can be more strict).  -A INPUT -p tcp --dport ssh -j ACCEPT --syn -m state --state NEW  # Allow RT.Warehouse on all networks.  -A INPUT -p tcp --dport postgres -j ACCEPT --syn -m state --state NEW  # Allow Greenplum Command Center on the customer facing network.  -A INPUT -i eth1 -p tcp --dport 28080 -j ACCEPT --syn -m state --state NEW  # Allow ping and any other icmp traffic on the interconnect networks.  -A INPUT -i eth4 -p icmp -s 192.0.2.0/22 -j ACCEPT  -A INPUT -i eth5 -p icmp -s 198.51.100.0/22 -j ACCEPT  # Allow ping only on the admin network on Greenplum DCA.  -A INPUT -i eth0 -p icmp --icmp-type echo-request -s 203.0.113.0/21 -j ACCEPT  # Log an error if a packet passes through the rules to the default  # INPUT rule (a DROP).  -A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7  COMMIT |

Пример правил iptables хоста сегмента

Пример правил iptables для файла /etc/sysconfig/iptables на хостах сегмента RT.Warehouse. Правила для хостов сегмента аналогичны основным правилам с меньшим количеством интерфейсов и меньшим количеством служб udp и tcp.

|  |
| --- |
| \*filter  :INPUT DROP  :FORWARD DROP  :OUTPUT ACCEPT  -A INPUT -i lo -j ACCEPT  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  -A INPUT -i eth2 -p udp -s 192.0.2.0/22 -j ACCEPT  -A INPUT -i eth3 -p udp -s 198.51.100.0/22 -j ACCEPT  -A INPUT -i eth2 -p tcp -s 192.0.2.0/22 -j ACCEPT --syn -m state --state NEW  -A INPUT -i eth3 -p tcp -s 198.51.100.0/22 -j ACCEPT --syn -m state --state NEW  -A INPUT -p tcp --dport ssh -j ACCEPT --syn -m state --state NEW  -A INPUT -i eth2 -p icmp -s 192.0.2.0/22 -j ACCEPT  -A INPUT -i eth3 -p icmp -s 198.51.100.0/22 -j ACCEPT  -A INPUT -i eth0 -p icmp --icmp-type echo-request -s 203.0.113.0/21 -j ACCEPT  -A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7  COMMIT |

УТИЛИТЫ УПРАВЛЕНИЯ УСТАНОВКОЙ

В разделе представлены ссылки на служебные программы командной строки, используемые для установки и инициализации системы RT.Warehouse.

Полный справочник по всем утилитам RT.Warehouse см. в Справочнике по утилитам Инструкции администратора.

Следующие утилиты управления RT.Warehouse находятся в $GPHOME/bin:

gpactivatestandby;

gpaddmirrors;

gpcheckperf;

gpdeletesystem;

gpinitstandby;

gpinitsystem;

gppkg;

gpscp;

gpssh;

gpssh-exkeys;

gpstart;

gpstop.

ПЕРЕМЕННЫЕ СРЕДЫ RT.WAREHOUSE

В разделе представлена ссылка на переменные среды, которые необходимо установить для RT.Warehouse.

Установите их в профиле начальной оболочки вашего пользователя (например, ~/.bashrc или ~/.bash\_profile) или в /etc/profile, если вы хотите установить их для всех пользователей.

Обязательные переменные среды

|  |
| --- |
| **Примечание.**  GPHOME, PATH и LD\_LIBRARY\_PATH можно установить, установив источник файла greenplum\_path.sh из каталога установки RT.Warehouse. |

GPHOME

Переменная среда GPHOME должна указывать место установки программного обеспечения RT.Warehouse. Например:

|  |
| --- |
| GPHOME=/usr/local/greenplum-db-6.x.x  export GPHOME |

PATH

Переменная среда PATH должна указывать на расположение каталога bin RT.Warehouse. Например:

|  |
| --- |
| PATH=$GPHOME/bin:$PATH  export PATH |

LD\_LIBRARY\_PATH

Переменная среда LD\_LIBRARY\_PATH должна указывать на расположение файлов библиотеки RT.Warehouse/PostgreSQL. Например:

|  |
| --- |
| LD\_LIBRARY\_PATH=$GPHOME/lib  export LD\_LIBRARY\_PATH |

MASTER\_DATA\_DIRECTORY

Переменная среда MASTER\_DATA\_DIRECTORY должна указывать на каталог, созданный утилитой gpinitsystem в расположении каталога основных данных. Например:

|  |
| --- |
| MASTER\_DATA\_DIRECTORY=/data/master/gpseg-1  export MASTER\_DATA\_DIRECTORY |

Необязательные переменные среды

Ниже приведены стандартные переменные среды PostgreSQL, которые также распознаются в RT.Warehouse. Возможно, вы захотите добавить в свой профиль переменные среды, связанные с подключением, для удобства, чтобы вам не нужно было вводить так много параметров в командной строке для клиентских подключений. Обратите внимание, что эти переменные среды следует устанавливать только на хосте мастера RT.Warehouse.

PGAPPNAME

Переменная среда PGAPPNAME указывает на имя приложения, которое обычно задаётся приложением при подключении к серверу. Это имя отображается в обзоре действий и в записях журнала. Переменная среда PGAPPNAME ведёт себя так же, как параметр соединения application\_name. Значение по умолчанию для application\_name — psql. Имя не может быть длиннее 63 символов.

PGDATABASE

Переменная среда PGDATABASE содержит имя базы данных по умолчанию для использования при подключении.

PGHOST

Переменная среда PGHOST содержит имя хоста мастера RT.Warehouse.

PGHOSTADDR

Переменная среда PGHOSTADDR содержит числовой IP-адрес хоста мастера. Его можно установить вместо или в дополнение к PGHOST, чтобы избежать накладных расходов на поиск DNS.

PGPASSWORD

Переменная среда PGPASSWORD содержит пароль, используемый в том случае, если сервер требует аутентификации по паролю. Использование этой переменной среды не рекомендуется из соображений безопасности (некоторые операционные системы позволяют пользователям без прав доступа root видеть переменные среды процесса через ps). Вместо этого рассмотрите возможность использования файла ~/.pgpass.

PGPASSFILE

Переменная среда PGPASSFILE содержит имя файла паролей, используемого для поиска. Если среда не установлена, по умолчанию используется ~/.pgpass. Дополнительную информацию смотрите в теме [о файле паролей](https://www.postgresql.org/docs/9.4/libpq-pgpass.html) в документации PostgreSQL.

PGOPTIONS

Переменная среда PGOPTIONS устанавливает дополнительные параметры конфигурации для мастер-сервера RT.Warehouse.

PGPORT

Переменная среда PGPORT содержит номер порта сервера RT.Warehouse на хосте мастера. Порт по умолчанию — 5432.

PGUSER

Переменная среда PGUSER содержит имя пользователя RT.Warehouse, используемое для подключения.

PGDATESTYLE

Переменная среда PGDATESTYLE устанавливает стиль представления даты/времени по умолчанию для сеанса (эквивалентно SET datestyle TO ...).

PGTZ

Переменная среда PGTZ устанавливает часовой пояс по умолчанию для сеанса (эквивалентно SET timezone TO ...).

PGCLIENTENCODING

Переменная среда PGCLIENTENCODING устанавливает кодировку набора символов клиента по умолчанию для сеанса (эквивалентно SET client\_encoding TO ...).

ПРИМЕР ANSIBLE PLAYBOOK

Пример использования Ansible для установки версии программного обеспечения RT.Warehouse на хосты, которые будут составлять систему RT.Warehouse.

В этом playbook Ansible показано, как задачи, описанные в разделе 5, могут быть автоматизированы с помощью [Ansible](https://docs.ansible.com/).

|  |
| --- |
| **Внимание.**  Данный playbook предоставляется только в качестве примера, чтобы проиллюстрировать, как задачи настройки кластера RT.Warehouse и установки программного обеспечения могут быть автоматизированы с помощью таких инструментов подготовки, как Ansible, Chef или Puppet. |

Пример playbook разработан для использования с РЕД ОС 7. Он создаёт пользователя gpadmin, устанавливает версию программного обеспечения RT.Warehouse, устанавливает владельца и группу установленного программного обеспечения на gpadmin и устанавливает ограничения безопасности Pam для пользователя gpadmin.

Вы можете изменить скрипт для работы с платформой вашей операционной системы и для выполнения дополнительных задач настройки хоста.

Ниже приведены шаги по использованию этого скрипта Ansible.

1. Установите Ansible на управляющую ноду с помощью диспетчера пакетов. См. [документацию по Ansible](https://docs.ansible.com/) для получения помощи по установке.

Настройте SSH без пароля от управляющей ноды ко всем хостам, которые будут частью кластера RT.Warehouse. Вы можете использовать команду ssh-copy-id для установки публичного ключа SSH на каждом хосте в кластере. В качестве альтернативы, ваше программное обеспечение для подготовки может предоставить более удобные способы безопасной установки открытых ключей на нескольких хостах.

Создайте инвентарь Ansible, создав файл с именем hosts со списком хостов, из которых будет состоять кластер RT.Warehouse. Например:

|  |
| --- |
| mdw  sdw1  sdw2  ... |

Этот файл можно отредактировать и позже использовать с утилитами gpssh-exkeys и gpinitsystem RT.Warehouse.

Скопируйте приведённый ниже код playbook в файл ansible-playbook.yml на своём узле управления Ansible.

Измените переменные playbook в верхней части playbook, такие как административный пользователь gpadmin и пароль для создания, а также версию устанавливаемой RT.Warehouse.

Запустите playbook, передав пакет для установки в параметр package\_path.

|  |
| --- |
| ansible-playbook ansible-playbook.yml -i hosts -e package\_path=./greenplum-db-6.0.0-rhel7-x86\_64.rpm |

Ansible Playbook — установка RT.Warehouse для РЕД ОС 7

|  |
| --- |
| ---  - hosts: all  vars:  - version: "6.0.0"  - greenplum\_admin\_user: "gpadmin"  - greenplum\_admin\_password: "changeme"  # - package\_path: passed via the command line with: -e package\_path=./greenplum-db-6.0.0-rhel7-x86\_64.rpm  remote\_user: root  become: yes  become\_method: sudo  connection: ssh  gather\_facts: yes  tasks:  - name: create greenplum admin user  user:  name: "{{ greenplum\_admin\_user }}"  password: "{{ greenplum\_admin\_password | password\_hash('sha512', 'DvkPtCtNH+UdbePZfm9muQ9pU') }}"  - name: copy package to host  copy:  src: "{{ package\_path }}"  dest: /tmp  - name: install package  yum:  name: "/tmp/{{ package\_path | basename }}"  state: present  - name: cleanup package file from host  file:  path: "/tmp/{{ package\_path | basename }}"  state: absent  - name: find install directory  find:  paths: /usr/local  patterns: 'greenplum\*'  file\_type: directory  register: installed\_dir  - name: change install directory ownership  file:  path: '{{ item.path }}'  owner: "{{ greenplum\_admin\_user }}"  group: "{{ greenplum\_admin\_user }}"  recurse: yes  with\_items: "{{ installed\_dir.files }}"  - name: update pam\_limits  pam\_limits:  domain: "{{ greenplum\_admin\_user }}"  limit\_type: '-'  limit\_item: "{{ item.key }}"  value: "{{ item.value }}"  with\_dict:  nofile: 524288  nproc: 131072  - name: find installed greenplum version  shell: . /usr/local/greenplum-db/greenplum\_path.sh && /usr/local/greenplum-db/bin/postgres --gp-version  register: postgres\_gp\_version  - name: fail if the correct greenplum version is not installed  fail:  msg: "Expected greenplum version {{ version }}, but found '{{ postgres\_gp\_version.stdout }}'"  when: "version is not defined or version not in postgres\_gp\_version.stdout" |

После успешного выполнения скрипта можно переходить к разделам 6 и 8.

1. Параметр SQL\_ASCII ведёт себя значительно иначе, чем другие параметры. Байтовые значения 0–127 интерпретируются в соответствии со стандартом ASCII, а байтовые значения 128–255 принимаются как неинтерпретированные символы. Если вы работаете с любыми данными, отличными от ASCII, неразумно использовать параметр SQL\_ASCII в качестве клиентской кодировки. SQL\_ASCII не поддерживается в качестве кодировки сервера. [↑](#footnote-ref-1)