



УТВЕРЖДЕНО

Генеральный директор
ООО «Клауд Солюшнс»

Д.С. Хлебородов



Описание процессов, обеспечивающих поддержание жизненного цикла

Cloud X Greenplum

СОДЕРЖАНИЕ

1. Основные операции установки (обновлении) и поддержки кластера Greenplum	4
1.1 Установка (обновление) Greenplum	4
1.1.1 Функциональные возможности deb пакета	4
1.1.2 Добавление репозитория с deb пакетами Greenplum	4
1.1.3 Установка Greenplum из репозитория	4
1.2 Инициализация кластера Greenplum.....	4
1.2.1 Настройка беспарольного ssh доступа для gpadmin	5
1.2.2 Настройка /etc/hosts	5
1.2.3 Создание host файла инициализации	5
1.2.4 Создание конфигурационного файла инициализации	5
1.2.5 Запуск инициализации кластера	6
1.3 Инициализация нового standby хоста	6
1.3.1 Удаление дата директории на standby хосте.....	6
1.3.2 Запуск команды инициализации нового standby хоста	7
1.3.3 Проверка синхронизации нового stanby хоста	7
1.4 Конфигурирование кластера Greenplum.....	7
1.4.1 Конфигурация мастера и сегментов	7
1.4.2 Классификация параметров	7
1.4.3 Просмотр и изменение параметров.....	8
1.4.4 Просмотр параметров.....	8
1.4.5 Установка master параметров	9
1.4.6 Установка local параметров.....	9
1.5 Запуск и остановка кластера Greenplum	9
1.5.1 Утилиты для запуска и остановки	10
1.5.2 Запуск кластера Greenplum	10
1.5.3 Остановка кластера Greenplum	10
1.5.4 Перезагрузка изменений в файле конфигурации без остановки системы	10
1.6 Методы борьбы с bloat	11
1.6.1 VACUUM.....	11
1.6.2 VACUUM ANALYZE	11
1.6.3 VACUUM FULL.....	11

1.6.4 VACUUM и VACUUM FULL для append-optimized таблиц	12
1.7 Запуск и остановка кластера Greenplum	12
1.7.1 Создание пользователя в СУБД.....	12
1.7.2 Редактирование pg_hba.conf.....	12
1.8 Управление ролями и привилегиями в Greenplum	13
1.8.1 Понятие ролей в Greenplum	13
1.8.2 Атрибуты	13
1.8.3 Привилегии	14
1.8.4 Предоставление доступа к объектам БД.....	15
2 Операции по восстановлению кластера Greenplum	16
2.1 Восстановление сегментов.....	16
2.1.1 Детектирование вышедших из строя сегментов	16
2.1.2 Сценарии восстановления	16
Инкрементальное восстановление.....	16
Полное восстановление	16
Ребалансировка сегментов	17
2.2 Активация standby хоста	17
2.2.1 Запуск команды активации standby хоста.....	17
2.2.2 Ход работы утилиты	18
3 Методы решения аварийных ситуаций при администрировании кластера Greenplum	18
3.1 Аварийные ситуации и методы их решения администратором	18
4 Требования к численности и квалификации персонала для обеспечения жизненного цикла ПО19	
4.1 Требования к численности персонала	20
4.2 Требования к квалификации персонала, порядку их подготовки и контроля знаний и навыков	20

1. Основные операции установки (обновления) и поддержки кластера Greenplum

1.1 Установка (обновление) Greenplum

Установка (обновление) Greenplum осуществляется с помощью deb пакетов, собираемых командой Bigdata CloudX и размещаемых во внешнем репозитории CloudX.

1.1.1 Функциональные возможности deb пакета

Представленный в репозитории deb пакет выполняет следующие функции:

- Установка бинарных пакетов Greenplum в директории /usr/local/gpdb
- Установка зависимостей в виде apt пакетов
- Установка бинарных пакетов библиотеку gp-xerces в директорию /usr/local
- Установка pip2 и всех необходимых python библиотек
- Создание основного пользователя gpadmin и выдача ему всех необходимых

доступов

- Установка параметров подключения ssh
- Добавление en_US.UTF-8 локали

1.1.2 Добавление репозитория с deb пакетами Greenplum

Для получения доступа к репозиторию greenplum необходимо выполнить ряд следующих команд:

```
repo_host="http://deb.cloudx.group/bigdata/repo"
# Добавление gpg ключа
wget -qO - "${repo_host}/ps-bigdata-cloudx_public_key.gpg" | sudo apt-key add -
# Добавление CloudX репозитория
echo "deb [arch=amd64] ${repo_host}/ bullseye main" | sudo tee
/etc/apt/sources.list.d/cloudx_bigdata.list
# Обновление apt кэша
sudo apt update
```

1.1.3 Установка Greenplum из репозитория

Установка последней версии Greenplum:

```
sudo apt instal gpdb-6
```

С указанием конкретной версии:

```
sudo apt instal gpdb-6=6.24.1
```

1.2 Инициализация кластера Greenplum

После установки бинарных файлов Greenplum и всех его зависимостей на хосты, необходимо инициализировать кластер. Это делается через master хост Greenplum.

Так как Greenplum Database является распределенной системой, процесс инициализации системы управления базой данных (СУБД) Greenplum включает инициализацию

нескольких индивидуальных экземпляров базы данных PostgreSQL (в Greenplum они называются экземплярами сегмента).

Каждый экземпляр базы данных (главный и все сегменты) должен быть инициализирован на всех хостах в системе таким образом, чтобы все они могли совместно работать как единая СУБД. Greenplum предоставляет свою собственную версию initdb, называемую gpinitssystem, которая отвечает за инициализацию базы данных на главном сервере и на каждом сегменте, а также за запуск каждого экземпляра в правильном порядке.

1.2.1 Настройка беспарольного ssh доступа для gpadmin

В первую очередь необходимо предоставить беспарольный доступ между всеми хостами кластера Greenplum пользователю gpadmin. Делается это добавлением публичного ssh ключа в файл ~/.ssh/authorized_keys для каждого хоста. Таким образом, если кластер состоит из 4-ех хостов, то каждый из них должен содержать публичные ssh ключи других хостов в файле ~/.ssh/authorized_keys.

1.2.2 Настройка /etc/hosts

Для корректной работы Greenplum Database, важно правильно настроить файл /etc/hosts на всех узлах системы. В этом файле должны быть указаны IP-адреса и соответствующие им полные доменные имена и короткие имена всех хостов кластера. Пример такого блока в /etc/hosts для кластера из 4-ех хостов:

```
172.20.226.223 vs-dwh-gpm1.st.tech vs-dwh-gpm1
172.20.226.224 vs-dwh-gpm2.st.tech vs-dwh-gpm2
172.20.226.221 vs-dwh-gps1.st.tech vs-dwh-gps1
172.20.226.222 vs-dwh-gps2.st.tech vs-dwh-gps2
```

1.2.3 Создание host файла инициализации

Инструмент gpinitssystem требует файл с хостами, содержащий список адресов для каждого сегментного хоста. В этот файл необходимо добавить полные доменные имена всех сегментных хостов в кластере. Пример такого хост файла /home/gpadmin/gpdb_configs/gpdb_segment_hosts:

```
vs-dwh-gps1.st.tech
vs-dwh-gps2.st.tech
```

1.2.4 Создание конфигурационного файла инициализации

Файл конфигурации Greenplum Database определяет параметры для утилиты gpinitssystem, указывая, как следует настроить систему Greenplum. Пример такого файла конфигурации /home/gpadmin/gpdb_configs/gpinitssystem.conf:

```
# Параметры конфигурации для инициализации кластера
ARRAY_NAME="CloudX GPDB cluster"
MACHINE_LIST_FILE="/home/gpadmin/gpdb_configs/gpdb_segment_hosts"
MASTER_HOSTNAME=vs-dwh-gpm1
MASTER_DIRECTORY=/data1/master
MASTER_PORT=5432
SEG_PREFIX=gpseg
PORT_BASE=10000
MIRROR_PORT_BASE=10500
declare -a DATA_DIRECTORY=("/data1/primary" "/data1/primary"
"/data1/primary" "/data1/primary")
declare -a MIRROR_DATA_DIRECTORY=("/data1/mirror" "/data1/mirror"
"/data1/mirror" "/data1/mirror")
```

```
ENCODING=UNICODE
DATABASE_NAME=prod_dwh
CHECK_POINT_SEGMENTS=8
```

Ключевые параметры в данном файле, которые можно изменить:

- MACHINE_LIST_FILE - путь до файла с сегмент хостами для инициализации
- MASTER_HOSTNAME - полное доменное имя мастер хоста
- MASTER_DIRECTORY - директория установки Greenplum мастера
- SEG_PREFIX - префикс сегмента
- declare -a DATA_DIRECTORY - определяет директории для сегментов

Greenplum. Количество блоков в этом списке определяет количество primary сегментов на каждом сегмент хосте

- declare -a MIRROR_DATA_DIRECTORY - аналогично declare -a DATA_DIRECTORY, но для зеркальных сегментов
- DATABASE_NAME - название базы данных, которая будет создана при инициализации кластера

1.2.5 Запуск инициализации кластера

Для инициализации кластера используется команда gpinitssystem. Так, команда для инициализации кластера с конфигурацией 1 мастер, 1 стэндбай и 2 сегмент хоста будет выглядеть следующим образом:

```
su - gpadmin
source /usr/local/greenplum-db/greenplum_path.sh
gpinitssystem -c /home/gpadmin/gpdb_configs/gpinitssystem.conf -h
/home/gpadmin/gpdb_configs/gpdb_segment_hosts -s vs-dwh-gpm2.st.tech
```

1.3 Инициализация нового standby хоста

Утилита gpinitstandby добавляет резервный standby мастер в ваш кластер Greenplum. Если в вашей системе уже настроен резервный координатор, используйте опцию -r для его удаления перед добавлением нового standby координатора.

1.3.1 Удаление дата директории на standby хосте

При инициализации нового standby хоста необходимо предварительно удалить все данные из дата директории standby хоста. Иначе процесс инициализации standby завершится со следующей ошибкой:

```
20231120:10:44:31:4176330 gpinitstandby:vs-dwh-gpm2:gpadmin-[ERROR]:-
Data directory already exists on host vs-dwh-gpm1.st.tech
20231120:10:44:31:4176330 gpinitstandby:vs-dwh-gpm2:gpadmin-[ERROR]:-
Failed to create standby
20231120:10:44:31:4176330 gpinitstandby:vs-dwh-gpm2:gpadmin-[ERROR]:-
Error initializing standby master: master data directory exists
```

Так, на standby хосте необходимо выполнить:

```
sudo rm -rf /data1/master/gpseg-1
```

Сделать это необходимо для того, чтобы все данные с master хоста были успешно реплицированы на standby хост.

1.3.2 Запуск команды инициализации нового standby хоста

Для инициализации standby хоста необходимо выполнить на активном master хосте из-под пользователя gpadmin, указав в -s адрес нового standby хоста:

```
gpinitstandby -as vs-dwh-gpm2.st.tech
```

1.3.3 Проверка синхронизации нового standby хоста

Для проверки, что новый standby хост синхронизирован с текущим мастером, запустите gpstate -f. При синхронизации, в выводе gpstate -f вы найдете такую строку:

```
20230607:06:50:06:004205 gpstate:test1-m:gpadmin-[INFO]:--Sync state: sync
```

Или с помощью psql:

```
psql -d prod_dwh -c 'SELECT client_addr, state FROM pg_stat_replication';
```

```
client_addr | state
-----+-----
172.20.226.224 | streaming
(1 row)
```

1.4 Конфигурирование кластера Greenplum

Параметры конфигурации сервера влияют на поведение СУБД Greenplum. Они являются частью системы конфигурации "Grand Unified Configuration" ("GUCs") из PostgreSQL. Большинство из них идентичны параметрам конфигурации PostgreSQL, однако некоторые специфичны именно для Greenplum.

1.4.1 Конфигурация мастера и сегментов

Файлы конфигурации сервера содержат параметры, определяющие поведение сервера. Файл конфигурации СУБД Greenplum **postgresql.conf** находится в каталоге данных экземпляра базы данных. Мастер и каждый сегмент имеют свой собственный файл **postgresql.conf**.

1.4.2 Классификация параметров

Каждый конфигурационный параметр Greenplum имеет свою классификацию, которая объясняет, как изменяется и применяется данный параметр.

Так, например, для параметра **max_connections**, определяющего максимальное количество подключений к серверу базы данных:

max_connections		
Диапазон значений	Стандартное значение	Классификация параметра
10 - 8388607	250 on master 750 on segments	local, system, restart

Проверить классификацию каждого параметра и определить условия его применения можно в [документации](#). Здесь также можно увидеть стандартные значения и тип данных значения для каждого параметра конфигурации.

Описание параметров классификаторов:

Классификация параметра	Описание параметра
master or local	master – устанавливается только в postgresql.conf для master host. Значение передается (или в некоторых случаях игнорируется) экземплярам сегмента во время выполнения запроса. local – устанавливается в postgresql.conf для master host И для каждого segment host . Каждый экземпляр сегмента проверяет свой postgresql.conf, чтобы получить значение этого параметра. Для применения всегда требует перезапуска сервера.
session or system	session – может быть изменен в рамках базы данных, роли, роли в базе данных, а также на уровне сессии. Если параметр задан на нескольких уровнях, то приоритет имеет наиболее детализированная настройка (то есть параметр, установленный на уровне сессии, перезаписывает все остальные параметры). system – может быть изменен только на уровне postgresql.conf.
restart or reload	restart – для применения параметра требуется перезапуск кластера Greenplum с помощью gpstop -r. reload – для применения параметра требуется только перечитать файлы конфигурации с помощью gpstop -u.
superuser	superuser – параметр может быть изменен только ролью, обладающей правами суперпользователя.
read only	read only – параметр не может быть изменен и доступен только для просмотра.

1.4.3 Просмотр и изменение параметров

Для просмотра и изменения параметров в файле конфигурации **postgresql.conf** используется утилита **gpconfig**, которая позволяет администраторам устанавливать, отменять настройки или просматривать параметры конфигурации из файлов **postgresql.conf** всех экземпляров (master, segments и mirrors) в СУБД Greenplum. При установке параметра администратор также имеет возможность указать другое значение для master, если это необходимо.

Дополнительная информация об утилите gpconfig представлена в [документации](#).

1.4.4 Просмотр параметров

Для просмотра текущих конфигурационных параметров можно воспользоваться утилитой **gpconfig** с флагом **-s**. Например:

```
gpconfig -s max_connections
```

Так, как утилита поддерживает просмотр и изменение не всех параметров конфигурации Greenplum, можно воспользоваться SQL оператором **SHOW**:

```
psql -d prod_dwh -c 'SHOW ALL;'
```

```
psql -d prod_dwh -c 'SHOW max_connections;'
```

1.4.5 Установка master параметров

Для изменения параметра конфигурации мастера необходимо обновить этот параметр в файле **postgresql.conf** непосредственно на мастере рекомендуется использовать утилиту **gpconfig** с флагом **--masteronly**:

```
gpconfig -c max_connections -v 100 --masteronly
```

Для сессионных параметров, которые не требуют перезапуска сервера, загрузите изменения **postgresql.conf** следующим образом:

```
gpstop -u
```

Для изменений параметров, которые требуют перезапуска сервера, перезапустите СУБД Greenplum следующим образом:

```
gpstop -r
```

1.4.6 Установка local параметров

Для изменения параметра локальной конфигурации в нескольких сегментах необходимо обновить этот параметр в файле **postgresql.conf** каждого соответствующего сегмента, как для основных, так и для зеркальных сегментов. Рекомендуется использовать утилиту **gpconfig** для установки параметра во всех файлах **postgresql.conf** в системе Greenplum. В данном примере значение **max_connections** для сегментов устанавливается 750, а для мастера 150 (если не указывать ключ **-m**, то на мастере также будет выставлено значение из ключа **-v**):

```
gpconfig -c max_connections -v 750 -m 150
```

Для сессионных параметров, которые не требуют перезапуска сервера, загрузите изменения **postgresql.conf** следующим образом:

```
gpstop -u
```

Для изменений параметров, которые требуют перезапуска сервера, перезапустите СУБД Greenplum следующим образом:

```
gpstop -r
```

1.5 Запуск и остановка кластера Greenplum

Greenplum — это распределенная система баз данных, состоящая из множества серверных экземпляров (мастер и сегменты), которые запускаются или останавливаются на всех хостах системы таким образом, чтобы они могли совместно работать как единая СУБД.

Процедура запуска и остановки Greenplum отличается от стандартных процедур в PostgreSQL.

1.5.1 Утилиты для запуска и остановки

- **gpstart**: Используется для запуска Greenplum Database.
- **gpstop**: Используется для остановки Greenplum Database.

Обе утилиты находятся в директории **\$GPHOME/bin** на мастер-хосте Greenplum. Не рекомендуется использовать команды **kill -9** или **kill -11**, так как они могут привести к повреждению базы данных и затруднить анализ причин проблем.

Все нижеперечисленные операции должны выполняться пользователем **gpadmin**.

1.5.2 Запуск кластера Greenplum

Для запуска инициализированного кластера Greenplum используется утилита **gpstart**, запускаемая на master host. Данная утилита запускает базу данных Greenplum, запуская все экземпляры базы данных Postgres в кластере. **gpstart** организует этот процесс и выполняет его параллельно. Для запуска без подтверждения пользователя (ключ **-a**):

```
gpstart -a
```

Более подробная информация об утилите представлена в [документации](#) ([gpstart \(vmware.com\)](#))

1.5.3 Остановка кластера Greenplum

Утилита **gpstop** предназначена для остановки или перезапуска кластера Greenplum и всегда запускается на мастер-хосте. При активации **gpstop** останавливает все процессы postgres в системе, включая мастер и все экземпляры сегмента.

Система ожидает завершения всех активных транзакций перед остановкой. Если спустя две минуты все еще есть активные соединения, **gpstop** предложит вам либо продолжить ожидание в умном режиме, выбрать быстрый режим или остановиться в немедленном режиме.

Чтобы остановить базу данных Greenplum без подтверждения пользователя (ключ **-a**):

```
gpstop -a
```

Чтобы остановить базу данных Greenplum в быстром режиме (ключ **-M fast**) и без подтверждения пользователя (ключ **-a**):

```
gpstop -aM fast
```

Для перезапуска Greenplum можно воспользоваться утилитой с флагом **-r** (сюда также можно указать ключ **-aM fast**):

```
gpstop -raM fast
```

Более подробная информация об утилите представлена в [документации](#) ([gpstop \(vmware.com\)](#))

1.5.4 Перезагрузка изменений в файле конфигурации без остановки системы

Вы можете применить изменения в файлах конфигурации Greenplum Database без полной остановки системы.

Утилита **gpstop** позволяет перезагрузить изменения в файле конфигурации **pg_hba.conf** и *runtime* параметры в файле **postgresql.conf** мастера без прерывания работы сервиса. Активные сессии применяют изменения при повторном подключении к базе данных. Однако многие параметры конфигурации сервера требуют полного перезапуска системы для их активации.

Для перезагрузки изменений в файле конфигурации без остановки системы Greenplum используйте команду:

```
gpstop -u
```

1.6 Методы борьбы с bloat

В системах управления базами данных, таких как Greenplum, "bloat" описывает избыточное использование дискового пространства, вызванное удаленными или обновленными строками, которые еще не были освобождены. Это может замедлить производительность и привести к неэффективному использованию дискового пространства. В этой статье рассмотрим методы борьбы с "bloat" в Greenplum. Данный эффект наследуется Greenplum от PostgreSQL из-за парадигмы MVCC.

VACUUM – это процесс в Greenplum Database, который позволяет восстановить место, занятое удаленными кортежами (или строками). Когда строки удаляются или становятся устаревшими из-за обновления, они физически не удаляются из таблицы. Вместо этого они остаются на диске до тех пор, пока не будет выполнена операция VACUUM.

После выполнения VACUUM рекомендуется собрать статистику по таблицам для оптимизатора с помощью команды ANALYZE (в Greenplum лучше использовать analyzedb).

Есть несколько вариантов использования этой операции:

1.6.1 VACUUM

- Отмечает удаленные и устаревшие данные в таблицах и индексах для будущего повторного использования;
- Дополнительное пространство не возвращается операционной системе. Оно остается доступным для повторного использования в рамках одной и той же таблицы;
- Может запускаться без указания конкретной таблицы. В таком случае под процесс VACUUM попадают все таблицы в базе;

1.6.2 VACUUM ANALYZE

- Сначала выполняет VACUUM, а затем ANALYZE для каждой выбранной таблицы. Это сочетание команд удобно для ежедневного обслуживания.

1.6.3 VACUUM FULL

- Перезаписывает всю таблицу в новый дисковый файл без лишнего пространства, позволяя освободить неиспользуемое пространство и вернуть его операционной системе;
- Значительно медленнее обычного VACUUM и требует эксклюзивной

блокировки на каждой обрабатываемой таблице;

- Является очень дорогостоящей операцией и может занять много времени для завершения на больших, распределенных таблицах базы данных Greenplum. Выполняйте операции VACUUM FULL во время периодов обслуживания базы данных.

1.6.4 VACUUM и VACUUM FULL для append-optimized таблиц

- Требуется достаточно доступного дискового пространства для размещения нового сегментного файла во время процесса VACUUM;
- Позволяет выполнять сканирование, вставки, удаления и обновления таблицы во время перезаписи сегментного файла, однако на короткое время устанавливается эксклюзивная блокировка для замены текущего сегментного файла на новый;
- Если при выполнении VACUUM отношение скрытых строк к общему числу строк в сегментном файле меньше порогового значения (10 по умолчанию), то сегментный файл не перезаписывается. Пороговое значение можно настроить с помощью параметра конфигурации сервера `gp_appendonly_compaction_threshold`;
- VACUUM FULL игнорирует порог и перезаписывает сегментный файл независимо от отношения.

1.7 Запуск и остановка кластера Greenplum

Аутентификация клиентов управляется конфигурационным файлом, который традиционно называется **pg_hba.conf** и расположен в каталоге с данными кластера базы данных. Так, для предоставления пользователю доступа к базе данных Greenplum необходимо создать пользователя с соответствующими правами и атрибутами в самой СУБД, а также настроить способы его подключения в конфигурационном файле **pg_hba.conf**.

1.7.1 Создание пользователя в СУБД

Создание пользователей в Greenplum осуществляется аналогично PostgreSQL с помощью SQL выражения **CREATE ROLE** с указанием необходимых атрибутов для авторизации (LOGIN, PASSWORD). Например:

```
CREATE ROLE bob WITH LOGIN PASSWORD 'password';
```

1.7.2 Редактирование pg_hba.conf

Обычный формат файла **pg_hba.conf** представляет собой набор записей, по одной в строке и интерпретируется следующим образом:

- пустые строки игнорируются, как и любой текст комментария после знака # ;
- записи не продолжаются на следующей строке;
- записи состоят из некоторого количества полей, разделённых между собой пробелом и/или tabs. В полях могут быть использованы пробелы, если они взяты в кавычки;

– если в кавычки берётся какое-либо зарезервированное слово в поле базы данных, пользователя или адресации (например, `all` или `replication`), то слово теряет своё особое значение и просто обозначает базу данных, пользователя или сервер с данным именем.

Для предоставления пользователям доступа к базе данных необходимо добавить соответствующую строку в конфигурационный файл `pg_hba.conf` по следующему формату:

```
local      <database> <user> <auth-method> [<auth-options>]
host       <database> <user> <address> <auth-method> [<auth-options>]
hostssl    <database> <user> <address> <auth-method> [<auth-options>]
hostnossl  <database> <user> <address> <auth-method> [<auth-options>]
host       <database> <user> <IP-address> <IP-mask> <auth-method>
[<auth-options>]
hostssl    <database> <user> <IP-address> <IP-mask> <auth-method>
[<auth-options>]
hostnossl  <database> <user> <IP-address> <IP-mask> <auth-method>
[<auth-options>]
```

Так, для предоставления пользователю `bob` доступа к базе данных `prod_dwh` с методом аутентификации `md5` с определенного хоста, запись может выглядеть так:

```
host      prod_dwh  bob  10.21.232.36/32  md5
```

Для применения внесенных изменений, необходимо загрузить новый файл конфигурации с помощью утилиты **gpstop -u**:

```
gpstop -u
```

1.8 Управление ролями и привилегиями в Greenplum

1.8.1 Понятие ролей в Greenplum

Greenplum управляет правами доступа к базе данных с помощью ролей. Понятие ролей включает в себя понятия групповых ролей (роли, которые входят в другие роли) и пользователей (роли с возможностью авторизации в базе данных). Создание, изменение и наследование ролей в Greenplum осуществляется с помощью SQL-команд от имени пользователя, имеющего соответствующие доступы.

Для более подробного описания управления ролями и привилегиями рекомендуется обратиться к [официальной документации Greenplum](#).

Каждая роль имеет ряд атрибутов и привилегий.

1.8.2 Атрибуты

Управление атрибутами осуществляется с помощью команды `ALTER`.

Атрибуты	Описание
SUPERUSER или NOSUPERUSER	Определяет, является ли роль суперпользователем. Чтобы выдать роле атрибут суперпользователя необходимо также обладать данным атрибутом. По

Атрибуты	Описание
	умолчанию роль создается с атрибутом NOSUPERUSER.
CREATEDB или NOCREATEDB	Разрешение или запрет на создание базы данных. По умолчанию — NOCREATEDB.
CREATEROLE или NOCREATEROLE	Разрешение или запрет на создание других ролей и управление ими. По умолчанию — NOCREATEROLE.
INHERIT или NOINHERIT	Наследует ли роль привилегии ролей, в составе которых она находится. По умолчанию — INHERIT.
LOGIN или NOLOGIN	Может ли роль авторизоваться в системе, т. е. является ли роль пользователем. По умолчанию — NOLOGIN.
CONNECTION LIMIT <значение>	Количество одновременных подключений для роли с атрибутом LOGIN. Значение по умолчанию — -1 (количество не ограничено).
CREATEEXTTABLE или NOCREATEEXTTABLE	Разрешение или запрет на создание внешних таблиц. По умолчанию — NOCREATEEXTTABLE.
PASSWORD '<пароль>'	Задание пароля для роли. Если аутентификация для роли не требуется, этот атрибут можно не указывать.
ENCRYPTED или UNENCRYPTED	Сохранение пароля в виде хеш-строки или в открытом текстовом виде. По умолчанию — ENCRYPTED.

1.8.3 Привилегии

Управление привилегиями осуществляется с помощью команда **GRANT/REVOKE**.

Тип объекта	Привилегии
Таблицы, внешние таблицы, представления	SELECT INSERT UPDATE DELETE REFERENCES TRIGGER TRUNCATE ALL
Столбцы	SELECT INSERT UPDATE REFERENCES ALL

Последовательности	USAGE SELECT UPDATE ALL
Базы данных	CREATE CONNECT TEMPORARY TEMP ALL
Домены	USAGE ALL
Внешние оболочки данных	USAGE ALL
Внешние серверы	USAGE ALL
Функции	EXECUTE ALL
Процедурные языки	USAGE ALL
Схемы	CREATE USAGE ALL
Табличные пространства	CREATE ALL
Типы	USAGE ALL
Протоколы	SELECT INSERT ALL

1.8.4 Предоставление доступа к объектам БД

Для предоставления доступа к таблицам и другим "отношениям" Greenplum, используется SQL выражение **GRANT**, а для отзыва прав - SQL выражение **REVOKE**. СУБД Greenplum объединяет понятия пользователей и групп в единый вид сущности, называемый ролью. Поэтому нет необходимости использовать ключевое слово **GROUP** для определения того, является ли получатель гранта пользователем или группой. **ГРУППА** по-прежнему разрешена в команде, но это шумовое слово. Команда **GRANT** имеет два основных варианта: один, который предоставляет привилегии объекту базы данных (таблице, столбцу, представлению, внешней таблице, последовательности, базе данных, оболочке внешних данных, внешнему серверу, функции, процедурному языку, схеме или табличному пространству), и другой, который предоставляет членство в роли. Для предоставления прав на таблицу и другие сущности БД:

```
GRANT ALL ON table1 TO role1;
```


Для предоставления ролевых доступов:

```
GRANT role1 TO bob;
```

Для подробного описания использования **GRANT/REVOKE** обращайтесь по ссылкам [GRANT](#) и [REVOKE](#).

2 Операции по восстановлению кластера Greenplum

2.1 Восстановление сегментов

Экземпляры сегментов может выйти из строя по различным причинам, таким как сбой хоста, сетевой сбой или сбой диска. За обнаружение вышедших из строя сегментов и переключение на их зеркальные сегменты отвечает **FTS** (fault tolerance server).

Для восстановления сегментов Greenplum используется утилита **gprecovreseg**, которая позволяет восстановить вышедшие из строя сегменты, а также провести ребалансировку кластера.

2.1.1 Детектирование вышедших из строя сегментов

Для получения информации о текущем статусе сегментов можно обратиться к утилите **gpstate**. Все нижеперечисленные операции должны выполняться пользователем **gpadmin**.

```
gpstate -e
```

Еще одним вариантом является использование системного представления **gp_segment_configuration**, где можно посмотреть как статус всех сегментов, так и их текущую и предпочитаемые роли (**primary** или **mirror**):

```
SELECT * FROM gp_segment_configuration WHERE status='d' or role!=  
preferred_role;
```

2.1.2 Сценарии восстановления

Инкрементальное восстановление

В случае, если с момента выхода сегмента из строя прошло не слишком много времени и восстановление возможно с помощью журналов транзакций (**WAL**), утилита **gprecovreseg** восстановит кластер с использованием **pg_rewind**. Утилиту необходимо запустить на мастере:

```
gprecovreseg
```

Полное восстановление

В случае если вышедшие из строя сегменты не удастся восстановить инкрементально, утилита **gprecovreseg** уведомит об этом и предложит запустить полное восстановление (ключ **-F**) с использованием **pg_basebackup**. Прежде чем запускать полное восстановление необходимо посмотреть журналы попытки инкрементального восстановления на сегменте, путь до которых будет показан после неудачной попытки

выполнения **gprecoverseg**. Для запуска полного восстановления на мастере необходимо запустить:

```
gprecoverseg -F
```

Ребалансировка сегментов

После процесса восстановления сегментов, некоторые из них могут не вернуться к своим изначальным ролям, определенным во время инициализации системы. Эта ситуация создаст несбалансированное состояние, при котором на некоторых узлах может находиться большее число активных сегментов, чем на других, что влияет на производительность всей системы.

Чтобы вернуть сегменты к их первоначальным ролям (primary или mirror), необходимо использовать утилиту **gprecoverseg** с флагом **-r**. Для этого на мастере запустите:

```
gprecoverseg -r
```

При выполнении **gprecoverseg -r** необходимо учитывать следующие моменты:

- Все сегменты должны находиться в рабочем состоянии;
- Все сегменты успешно должны быть синхронизированы;
- Любые запросы в базе будут автоматически отменены.

2.2 Активация standby хоста

При выходе из строя основного master хоста необходимо провести процедуру активации резервного standby хоста.

В Greenplum Database, утилита **gpactivatestandby** позволяет активировать резервный, standby мастер-хост и превратить его в активный мастер-экземпляр системы. После активации, standby мастер принимает на себя роль главного мастера, принимая клиентские соединения на порту мастера.

ПРИМЕЧАНИЕ

Перед запуском **gpactivatestandby** запустите **gpstate -f** для проверки синхронизации standby мастера с текущим мастером. При синхронизации, в выводе **gpstate -f** вы найдете такую строку:

```
20230607:06:50:06:004205 gpstate:test1-m:gpadmin-[INFO]:--Sync state: sync
```

При инициализации standby мастера, по умолчанию используется тот же порт, что и для активного мастера. Эту утилиту нужно запускать на мастер-хосте, который вы активируете, а не на вышедшем из строя мастер-хосте. Запуск этой утилиты подразумевает наличие настроенного standby мастера для системы (см. **gpinitstandby**).

2.2.1 Запуск команды активации standby хоста

Для активации standby хоста необходимо выполнить на standby хосте из под пользователя **gpadmin**:

```
EXPORT PGPORT=5432
```

```
gpactivatestandby -d /data1/master/gpseg-1
```

После активации standby мастера, запустите ANALYZE для обновления статистики запросов базы данных:

```
psql -d prod_dwh -c 'ANALYZE;'
```

Или с помощью analyzedb:

```
analyzedb -d prod_dwh -a
```

2.2.2 Ход работы утилиты

Утилита выполняет следующие шаги:

1. Останавливает процесс синхронизации (walreceiver) на standby мастере.
2. Обновляет системные таблицы каталогов standby мастера с использованием WAL журналов.
3. Активирует standby мастер как новый активный мастер системы.
4. Перезапускает систему Greenplum Database с новым мастер-хостом.

После активации standby мастера в роли основного мастера, в системе Greenplum Database больше нет настроенного standby мастера. Для его повторной настройки используйте утилиту gpinitstandby.

3 Методы решения аварийных ситуаций при администрировании кластера Greenplum

3.1 Аварийные ситуации и методы их решения администратором

В процессе работы с DWH Greenplum администратор может столкнуться с рядом проблем. В данной таблице представлены возможные ошибки, а также варианты их исправления:

Класс ошибки	Описание ошибки	Требуемые действия администратора при возникновении ошибки
Недоступность сегментов	Ошибки в кластере Greenplum, повлекшие вывод из строя одного или нескольких сегментов	Получить статус кластера и выяснить, какие сегменты недоступны. Просмотреть логи на наличие ошибок в master директории: /data1/master/gpseg-1/pg_log/*.csv. Просмотреть логи на хосте с недоступным сегментом: /data1/master/gpseg-<SEG_NUM>/pg_log/*.csv. Выявить проблему и локализовать и устранить ее. Запустить процесс восстановления кластера с помощью команды gprecoverseg (или gprecoverseg -F). При необходимости запустить ребаланс сегментов с помощью команды gprecoverseg -r.
Недоступность хоста мастера	Полная недоступность хоста мастера Greenplum	Передать информацию системному администратору, отвечающему за работоспособность хостов.

		<p>Проверить доступность standby хоста.</p> <p>Активировать standby с помощью утилиты <code>gpactivatestandby</code>.</p> <p>Переключение etl нагрузки, а также пользовательской нагрузки на standby.</p> <p>После возвращения в строй основного master хоста, инициализировать его, как standby хост с помощью утилиты <code>gpinitstandby</code>.</p> <p>Опционально – можно выполнить обратное переключение и сделать первоначальный master хост снова основным хостом. Делается это с помощью утилиты <code>gpactivatestandby</code>, однако после этого нужно вновь инициализировать standby хост.</p>
Недоступность хостов сегментов	Полная недоступность некоторых хостов сегментов Greenplum	<p>Передать информацию системному администратору, отвечающему за работоспособность хостов.</p> <p>После устранения проблемы и возвращения хостов в строй, запустить процесс восстановления кластера с помощью команды <code>gprecoverseg</code> (или <code>gprecoverseg -F</code>).</p> <p>Запустить ребаланс сегментов с помощью команды <code>gprecoverseg -r</code>.</p>
Нагрузка на базу	Даже простые запросы в базе выполняются слишком долго.	<p>Подключиться к базе, как <code>gadmin</code>, и с помощью <code>pg_stat_activity</code> выявить, какие запросы являются причиной повышенной нагрузки на базу.</p> <p>С помощью <code>pg_terminate_backend()</code> завершить найденные запросы.</p> <p>Если проблемный запрос - etl запрос, то отключить данный etl процесс и оптимизировать запрос. Если проблемный запрос - пользовательский запрос, то обратиться к пользователю с требованием прекратить выполнения запроса и оптимизировать его.</p>

4 Требования к численности и квалификации персонала для обеспечения жизненного цикла ПО

Поддержка осуществляется разработчиком программного обеспечения по месту нахождения персонала разработчика: город Москва, ул. Рочдельская, 15/56, этаж 2. Вопросы технической поддержки принимаются по электронной почте: Greenplum@support.cloudx.group в рабочие дни с 11.00 до 19.00.

Для эффективного администрирования и поддержки экосистемы Nadoor используется команда специалистов с различными навыками и полномочиями. Ниже представлена информация о минимальном составе команды, ролях и их полномочиях, уровне доступа и временных диапазонах для обеспечения поддержки.

Работа организована в команде с применением подходов гибкой разработки. Спринты (периоды разработки) организованы в двухнедельные циклы и поставляют инкремент обновления или исправления ошибок при необходимости.

Составление, согласование и фиксация результатов разработки документации и задач, связанных с обновлением или исправлением ошибок в ПО, осуществляется в системах JIRA и Confluence.

Разработка, доработка и отладка ПО осуществляется на вычислительных средствах, не участвующих в промышленной эксплуатации приложений.

Репозиторий исходных кодов ПО располагается на локальных ресурсах ООО «Клауд Солюшенс» в облачном сервисе Minio по адресу: 123376, город Москва, улица Рочдельская, 15, строение 56.

4.1 Требования к численности персонала

Численность персонала, обслуживающего Cloud X Greenplum, зависит от масштаба внедряемой системы и её специфических требований. Это определяется на этапе установки и настройки системы, учитывая количество управляемых кластеров, объемы данных, и требуемые уровни производительности и надежности.

4.2 Требования к квалификации персонала, порядку их подготовки и контроля знаний и навыков

Квалификация персонала должна соответствовать требованиям, обеспечивая эффективное управление и поддержание всех технических и программных компонентов Cloud X Greenplum.

Предусмотрены следующие ключевые роли в обслуживании системы:

1. **Системный администратор** (5 человек) - обладает глубокими знаниями в области управления базами данных, настройки кластеров и обеспечения безопасности системы. Должен иметь права для выполнения административных и конфигурационных задач на уровне системы.

2. **Базовый администратор баз данных** (1 человек) - специализируется на управлении и оптимизации работы баз данных в рамках Cloud X Greenplum, включая настройку производительности, мониторинг и устранение неполадок.

3. **Аналитик данных / Разработчик** (1 человек) - занимается разработкой и реализацией запросов, аналитических отчетов и интеграции с другими системами.

Требования к уровню компетенции и квалификации каждой из ролей должны быть четко определены в соответствующих должностных инструкциях и регулирующих документах. Также предусматривается регулярный контроль знаний и навыков персонала, включая тренинги и сертификации для поддержания высокого уровня квалификации.