Утвержден

ДВНБ.20002-02-УД

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СУБД «ЛИРА-Р»

Руководство администратора

ДВНБ.20002-02 94 01

Листов 33

2021

АННОТАЦИЯ

Настоящий документ является руководством администратора реляционной системы управления базами данных «Лира-Р» ДВНБ.20002-02 (далее по тексту — СУБД).

В документе приведено описание защищенной СУБД, ее настройка и установка.

Описаны серверные службы СУБД и утилиты командной строки, включая инструменты резервного копирования и восстановления данных.

Описано тестирование СУБД.

Документ предназначен для системных администраторов и разработчиков баз данных.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

СУБД предназначена для создания информационных и управляющих систем в составе автоматизированных систем, обрабатывающих информацию ограниченного доступа.

СУБД представлена в двух вариантах исполнения для функционирования под управлением следующих операционных систем (далее по тексту — ОС):

1) Операционная система специального назначения «Astra Linux Special Edition» РУСБ.10015-01.

2) Операционная система общего назначения «Стрелец» (ОС ОН «Стрелец») ФЛИР.90001-01.

СУБД по своим функциональным возможностям соответствует объектно-реляционной системе управления базами данных с открытыми исходными текстами PostgreSQL.

PostgreSQL — это свободно распространяемая объектно-реляционная система управления базами данных, поддерживающая стандарты языка SQL ANSI SQL-92, SQL-99, SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016.

2. СТРУКТУРА ПРОГРАММЫ

2.1. Состав СУБД

В состав СУБД входят следующие компоненты:

- средство управления кластерами (2.1.1);
- сервер СУБД (2.1.2);
- процедурные языки программирования (2.1.3);
- клиентские утилиты СУБД (2.1.4);
- прикладной программный интерфейс СУБД (2.1.5);
- графическое средство администрирования СУБД (2.1.6);
- средства поддержки геоданных (2.1.7);
- дополнительно поставляемые расширения сервера СУБД (2.1.8);
- средства тестирования сервера СУБД (2.1.9).

В настоящем дистрибутиве используется СУБД PostgreSQL версии 12.

2.1.1. Средство управления кластерами

Средство управления кластеров используется для создания и управления ими.

В состав данного средства входят следующие пакеты:

- `postgresql-common` — содержит набор скриптов для управления размещением и управлением кластеров;
- `postgresql` — метапакет, устанавливающий сервер СУБД;
- `postgresql-client` — метапакет, устанавливающий клиентов СУБД;
- `postgresql-doc` — метапакет для установки документации по СУБД.

2.1.2. Сервер СУБД

Сервер СУБД представлен пакетом `postgresql-xx` (здесь и далее xx - текущая версия СУБД, например 12), в состав которого входят сервисная служба `postgres`, утилиты для функционирования сервера СУБД и поставляемые с сервером расширения его функциональности:

- `postgres` — основная сервисная служба СУБД, реализующая сервер СУБД, и обеспечивающая обработку запросов, управление форматом данных и выполняющая фоновые операции обслуживания БД;
- `initdb` — утилита для создания кластера СУБД;

- pg_archivecleanup — утилита для удаления устаревших WAL сервера СУБД;
- pg_checksums — утилита для управления подсчетом контрольных сумм данных в кластере СУБД;
- pg_controldata — утилита для вывода сведений о сервере СУБД;
- pg_ctl — утилита управления кластером СУБД;
- pg_resetwal — утилита WAL и другой управляющей информации СУБД;
- pg_rewind — утилита синхронизации кластеров СУБД;
- pg_test_fsync — утилита определения лучшего варианта синхронизации WAL;
- pg_test_timing — утилита определения издержек замера времени;
- pg_upgrade — утилита обновления кластера СУБД;
- pg_waldump — утилита для вывода WAL в читаемом для человека виде.

Список расширений СУБД приведен в таблице 1.

Таблица 1

Расширение	Описание
adminpack	Функции администрирования.
amcheck	Функции проверки логической целостности стуктуры индексов.
auth_delay	Функции задержки в процессе проверки подлинности паролей, что усложняет их перебор.
auto_explain	Функции автоматического протоколирования планов выполнения медленных запросов.
bloom	Метод доступа для индексов, основанный на фильтре Блума.
btree_gin	Индексный класс GIN для btree.
btree_gist	Индексный класс GiST для btree.
checkpass	Тип данных для хранения зашифрованных паролей и функции для работы с ним.
citext	Тип данных для строк, нечувствительных к регистру.
cube	Тип данных многомерных кубов.
dblink	Функции доступа к другим базам данных из текущей базы данных.
dict_int	Дополнительный словарь полнотекстового поиска для целых чисел.
dict_xsyn	Дополнительный словарь полнотекстового поиска для синонимов.
earthdistance	Функции вычисления расстояния между точками на поверхности Земли.

Продолжение таблицы 1

Расширение	Описание
file_fdw	Обертка внешних данных (FDW) для файлов.
fuzzymatch	Функции вычисления схожести строк.
hstore	Тип данных ключ-значение.
intagg	Агрегатор и нумератор целых чисел.
intarray	Функции работы с массивами целых чисел.
isn	Тип данных для нумерации товаров.
lo	Функции управления большими объектами LARGE OBJECT.
ltree	Тип данных для представления меток данных в иерархической структуре.
pageinspect	Функции просмотра страниц баз данных.
passwordcheck	Функции проверки на простоту пользовательских паролей при их назначении.
pg_buffercache	Функции просмотра текущей информации о буферах.
pgcrypto	Криптографичечкие функции.
pg_freespacemap	Функции исследования карты свободного пространства (FSM).
pg_prewarm	Функции загрузки данных отношений в буферы СУБД
pgrowlocks	Функции просмотра информации о блокировках заданного отношения.
pg_stat_statements	Функции отслеживания выполнения запросов сервером СУБД.
pgstattuple	Функции для получения статистики на уровне кортежей.
pg_trgm	Функции и операторы для определения схожести алфавитно-цифровых строк на основе триграмм.
pg_visibility	Функции для просмотра информации о видимости кортежей таблицы.
postgres_fdw	Обертка внешних данных, используя которую можно получить доступ к данным, находящимся на других серверах PostgreSQL.
seg	Тип данных для определения длины отрезков с нечеткой длиной.
spi	Примеры реализации SPI и тригеров.
sslinfo	Функции вывода информации о сертификате клиента.
tablefunc	Примеры реализаций функций, возвращающие строки.
tcn	Триггерная функция, уведомляющая приёмники уведомлений об изменениях в любой таблице, к которой привязан триггер.
test_decoding	Пример модуля логического декодирования.
tsm_system_rows	Метод извлечения SYSTEM_ROWS, который можно использовать вместе с TABLESAMPLE для генерации тестовых данных.
tsm_system_time	Метод извлечения SYSTEM_TIME, который можно использовать вместе с TABLESAMPLE для генерации тестовых данных.

Окончание таблицы 1

Расширение	Описание
unaccent	Словарь полнотекстового поиска, убирающий все диакритические знаки лексем.
uuid-ossp	Функции генерации UUID.
xml2	Функции для выполнения запросов XPath и преобразований XSLT.
fasttrun	Транзакционно-небезопасная функция для усечения временных таблиц (поддержка 1С).
fulleq	Дополнительный оператор равенства для совместимости с Microsoft SQL Server (поддержка 1С).
mchar	Дополнительные типы данных для совместимости с Microsoft SQL Server (поддержка 1С).
online_analyze	Набор функций, которые немедленно обновляют статистику после операций INSERT, UPDATE, DELETE или SELECT INTO в целевых таблицах (поддержка 1С).
plantuner	Поддержка указаний для планировщика, позволяющих отключать или подключать определённые индексы при выполнении запроса (поддержка 1С).

2.1.3. Процедурные языки

Компонент «Процедурные языки» содержит набор расширений для написания хранимых процедур на разных языках программирования:

- postgresql-plperl-xx — процедурный язык Perl;
- postgresql-plpython3-xx — процедурный язык Python версии 3;
- postgresql-pltcl-xx — процедурный язык Tcl.

2.1.4. Клиентские утилиты СУБД

Клиентские утилиты СУБД содержатся в пакете postgresql-client-xx:

- clusterdb — утилита кластеризации базы данных;
- createdb — утилита создания базы данных;
- createuser — утилита для создания новых ролей в СУБД;
- dropdb — утилита для удаления существующей базы данных;
- dropuser — утилита для удаления существующей роли из СУБД;
- ecpg — встроенный C-препроцессор SQL;
- pg_basebackup — утилита создания резервной копии сервера СУБД;

- pg_config — утилита для вывода параметров конфигурации текущей установленной версии СУБД;
- pg_dump — утилита для создания резервной копии базы данных;
- pg_dumpall — утилита для создания резервной копии кластера в виде SQL;
- pg_isready — проверка соединения с сервером СУБД;
- pg_receivewal — утилита для просмотра WAL сервера;
- pg_recvlogical — утилита управления потоками логического декодирования;
- pg_restore — утилита восстановления бинарных дампов баз данных, сделанных с помощью pg_dump;
- psql — интерактивный терминал;
- reindexdb — утилита переиндексирования базы данных;
- vacuumdb — утилита для проведения очистки и анализа сервера баз данных.

2.1.5. Прикладной программный интерфейс СУБД

Компонент «Прикладные программные интерфейсы СУБД» содержит библиотеки прикладного программного интерфейса доступа к СУБД:

- libpq5 — библиотека C драйвера для доступа к СУБД;
- libecpg6 — библиотека встраивания SQL в C.

2.1.6. Графический инструмент администрирования

Компонент «Графическое средство администрирования» содержит графическую утилиту pgadmin4, предназначенное для управления объектами баз данных и ролями и состоит из следующих пакетов:

- pgadmin4-common — серверная компонента для запуска pgadmin4;
- pgadmin4 — графическое приложение для запуска pgadmin4;
- pgadmin4-doc — документация для pgadmin4;
- pgadmin4-apache2 — WSGI приложение для развертывания серверной компоненты на apache2;
- flask-* — web-компоненты, необходимые для работы pgadmin4.

2.1.7. Средство поддержки геоданных

Компонент «Средство поддержки геоданных» состоит из библиотек для работы с географическими типами данных, а также средства хранения и работы с геоданными в СУБД:

- `libcgal*`, `libcoin*`, `libgeos*`, `openscenegraph*` — библиотеки для работы с географическими типами данных;
- `postgis`, `postgresql-xx-postgis-3-scripts`, `postgresql-xx-postgis-3` — средства для организации хранения и функции обработки геоданных в СУБД.

2.1.8. Дополнительно поставляемые расширения сервера СУБД

Компонент «Дополнительно поставляемые расширения сервера СУБД» содержит расширения сервера СУБД для решения различных прикладных задач:

- `postgresql-xx-orafce` — средство, облегчающее миграцию БД Oracle на СУБД PostgreSQL;
- `postgresql-xx-jsquery` — расширение, добавляющий язык запросов к типу данных `jsonb`;
- `postgresql-xx-mysql-fdw` — FDW для доступа к СУБД MySQL;
- `postgresql-xx-rum` — метод доступа `rum`, предназначенный для ускорения полнотекстового поиска;
- `postgresql-xx-pghintplan` — хинты планировщика СУБД для тестирования запросов.
- `postgresql-xx-pldebugger` — отладчик кода Pl/pgSQL.

2.1.9. Средства тестирования сервера СУБД

Компонент «Средства тестирования сервера СУБД» содержит регрессионные тесты СУБД, библиотеки для проведения регрессионного тестирования, а также средство запуска тестов и содержится в пакете `postgresql-setests-xx`.

2.2. Состав дистрибутива

Состав дистрибутива представлен в таблице 2.

Таблица 2

Название пакета	Описание
libaccompat-*-se	Библиотека платформозависимых функций СЗИ (* — strelets, smolensk).
libecpg6	Библиотека встраивания SQL в C.
libecpg-compat3	Разделяемая библиотека встраивания C в SQL (старая версия).
libcgal12 libcgal-qt5-12	Вспомогательная C++ библиотека геометрических вычислений для PostGIS.
libcoin*	Вспомогательная библиотека 3D геометрии для PostGIS.
libgeos*	Вспомогательная библиотека для GIS, реализующая геометрические объекты.
libopenscenegraph*	Библиотека 3D графов.
libpq5	Клиентская библиотека доступа к PostgreSQL, реализованная на языке C.
lira-r-licences	Тексты лицензий компонентов СУБД.
pgadmin4	Графическая утилита администрирования PostgreSQL.
pgadmin4-common	Локализация и документация для pgadmin4.
pgadmin4-apache2	WSGI приложение для развертывания серверной компоненты на apache2.
flask-*	WEB-компоненты, необходимые для pgadmin4.
postgis	Метапакет для расширения postgis.
postgis-doc	Документация для расширения postgis.
postgresql-12	Серверный пакет PostgreSQL версии 12.
postgresql	Серверный метапакет PostgreSQL.
postgresql-12-orafce	Расширение PostgreSQL, упрощающее миграцию баз Oracle на PostgreSQL.
postgresql-12-pghintplan	Расширение PostgreSQL хинтов планировщика.
postgresql-12-pldebugger	Расширение PostgreSQL для отладки кода на PL/pgSQL.
postgresql-12-jsquery	Расширение, добавляющий язык запросов к типу данных jsonb.
postgresql-12-mysql-fdw	Расширение FDW для доступа к СУБД MySQL.
postgresql-12-rum	Расширение, добавляющее метод доступа rum.
postgresql-12-postgis-3	Библиотека расширения postgis для PostgreSQL.

Окончание таблицы 2

Название пакета	Описание
`postgresql-12-postgis-3-scripts`	Скрипты для функционирования расширения `postgis`.
`postgresql-client-12`	Клиентские утилиты командной строки для PostgreSQL.
`postgresql-client-common`	Вспомогательные компоненты `postgresql-common` для клиентских утилит.
`postgresql-common`	Компоненты для реализации инфраструктуры серверов СУБД PostgreSQL для Debian-подобных ОС.
`postgresql-doc-12`	Документация для PostgreSQL.
`postgresql-doc`	Метапакет, устанавливающий документацию для PostgreSQL.
`postgresql-plperl-12`	Процедурный язык Pl/perl для PostgreSQL.
`postgresql-plpython3-12`	Процедурный язык Pl/python версии 3 для PostgreSQL.
`postgresql-pltcl-12`	Процедурный язык Pl/tcl для PostgreSQL.
`postgresql-setests-12`	Регрессионные тесты и средство их запуска для PostgreSQL.

3. УСТАНОВКА ПРОГРАММЫ

Текст исполняемого модуля СУБД поставляется в виде репозитория на оптическом носителе.

Для установки необходимо вставить диск репозитория СУБД в устройство чтения дисков DVD-ROM, после чего выполнить в терминале от имени администратора следующие команды по подключению репозитория и обновлению сведений о составе пакетной базы:

```
$ sudo apt-cdrom add
$ sudo apt-get update
```

При установке компонентов СУБД может потребоваться установка дополнительных пакетов из репозитория ОС.

3.1. Установка сервера

Для установки сервера СУБД требуется выполнить команду:

```
$ sudo apt-get install postgresql
```

Данная команда установит пакет серверных компонентов СУБД, создаст кластер по умолчанию `main`, функционирующий на порту 5432 с суперпользователем `postgres` и одноименной базой данных `postgres`.

Далее проводится настройка параметров сервера СУБД, создание необходимых ролей и баз данных согласно разделам 4 и 6.

3.2. Установка процедурных языков

Для установки процедурных языков программирования необходимо установить один или несколько пакетов, указанных в 2.1.3 с помощью команды:

```
$ sudo apt-get install postgresql-<язык>-12
```

3.3. Установка программного интерфейса СУБД

Установка пакетов компонента «Программный интерфейс СУБД» производится с помощью команды:

```
$ sudo apt-get install <пакет>
```

В качестве пакета необходимо указать пакет, указанный в 2.1.5

3.4. Установка клиента

Для установки клиентских утилит СУБД требуется выполнить команду:

```
$ sudo apt-get install postgresql-client
```

3.5. Установка дополнительно поставляемых расширений

Установка дополнительно поставляемых расширений, указанных в 2.1.8 производится с помощью команды

```
$ sudo apt-get install <пакет>
```

4. НАСТРОЙКА

Настройка сервера СУБД заключается в редактировании параметров конфигурационных файлов.

Для организации управления нескольких кластеров СУБД используются утилиты пакета `postgresql-common`. Принцип организации нескольких кластеров СУБД на одной операционной системе состоит в физическом разнесении каталогов данных СУБД, конфигурационных файлов и прочих файлов.

Каталог конфигурационных файлов располагается по пути `/etc/postgresql/<версия>/<имя_кластера>`.

Помимо основного конфигурационного файла `postgresql.conf` дополнительно используются следующие:

- `pg_hba.conf` — файл аутентификации клиентов;
- `pg_ident.conf` — файл сопоставления имен пользователей.

Расположение указанных файлов, каталога данных и прочих файлов определяется в `postgresql.conf` следующими параметрами:

- `data_directory` — каталог для хранения данных;
- `config_file` — определяет путь конфигурационного файла (`postgresql.conf`);
- `hba_file` — путь к файлу настройки аутентификации `pg_hba.conf`;
- `indent_file` — путь к файлу настройки сопоставления имен пользователей;
- `external_pid_file` — путь к файлу с идентификатором запущенного процесса сервиса `postgres`.

По умолчанию каталог данных расположен по пути `/var/lib/postgresql/<версия>/<имя_кластера>`, а разделяемые файлы сервера `/usr/share/postgresql/<версия>`.

За установку соединения отвечают следующие параметры файла `postgresql.conf`:

- `listen_addresses` — определяет TCP/IP-адреса, по которым сервер ожидает подключения от клиентов. Список адресов задается через запятую. При указании символа `*` в качестве значения данного параметра, сервер при-

нимает подключения с любого IP-адреса;

- `port` — определяет TCP-порт для подключения;

- `max_connections` — определяет максимальное количество одновременных подключений;

- `superuser_reserved_connections` — определяет максимальное число одновременно открытых подключений от суперпользователей;

- `unix_socket_group` — определяет группу, владеющую доменным UNIX-сокетом;

- `unix_socket_permissions` — определяет права доступа на сокет.

Описание других параметров конфигурации приведены в официальной документации PostgreSQL.

4.1. Аутентификация

При попытке соединения с сервером СУБД клиентское приложение указывает пользователя СУБД, от имени которого осуществляется подключение. В пределах окружения SQL активное имя пользователя СУБД определяет права на объекты БД.

СУБД предлагает несколько различных методов аутентификации клиента. Метод, используемый для аутентификации конкретного клиентского соединения, может быть выбран на основе адреса узла сети клиента, БД и пользователя.

Несмотря на то, что имена пользователей СУБД PostgreSQL логически отделены от имен пользователей ОС, в которой запущен сервер, в соответствии с требованиями по защите информации от НСД требуется сопоставление пользователей СУБД пользователям ОС. Таким образом, при настройке аутентификации в СУБД следует использовать только методы аутентификации, в которых осуществляется подобное сопоставление.

5. НАСТРОЙКА АУТЕНТИФИКАЦИИ

При попытке соединения с сервером СУБД клиентское приложение указывает пользователя СУБД, от имени которого осуществляется подключение. В пределах окружения SQL активное имя пользователя СУБД определяет права на объекты БД.

СУБД предлагает несколько различных методов аутентификации клиента. Метод, используемый для аутентификации конкретного клиентского соединения, может быть выбран на основе адреса узла сети клиента, БД и пользователя.

В дополнение к стандартным методам аутентификации добавлен метод аутентификации по хешу пароля ГОСТ Р 34.11-2012.

6. АДМИНИСТРИРОВАНИЕ

В задачу администрирования баз данных входят:

- управление кластером БД;
- настройка параметров конфигурации кластера;
- резервирование и восстановление резервных копий;
- управление объектами баз данных и ролями;
- управление разграничением доступа.

Управление кластером БД приведено в 6.1.

Описание настройки параметров конфигурации кластера приведено в 4.

Описание процесса резервирования и восстановления резервных копий приведено в разделе 6.2.

Управление объектами баз данных и ролями приведено в разделе 6.3.

6.1. Управление кластером

Для решения задачи управления кластерами рекомендуется использовать скрипты из пакета `postgresql-common`. Они используют серверные утилиты командной строки, но учитывают особенности ОС, под управлением которых работают кластера СУБД.

К описываемым утилитами относятся:

- `pg_conftool` — утилита просмотра и настройки параметров конфигурации (см. 6.1.1);
- `pg_createcluster` — утилита создания новых кластеров (см. 6.1.2);
- `pg_ctlcluster` — утилита по управлению кластером (см. 6.1.3);
- `pg_dropcluster` — утилита удаления кластера (см. 6.1.4);
- `pg_lscluster` — утилита для просмотра состояния кластеров (см. 6.1.5);
- `pg_renamecluster` — утилита для переименования кластера (см. 6.1.6);
- `pg_upgradecluster` — утилита обновления кластера (см. 6.1.7).

Данные утилиты используют единый формат вызова:

```
$ sudo pg_<имя_утилиты> [опции] <версия> <имя_кластера>
 ↪  <действие>
```

В зависимости от назначения утилиты некоторые компоненты командной строки могут отсутствовать.

Далее приведено описание указанных утилит. Детальное описание можно найти в страницах помощи `man` каждой утилиты.

6.1.1. Настройка параметров конфигурации — `pg_conftool`

Утилита `pg_conftool` позволяет посмотреть значение активных параметров конфигурации из файла `postgresql.conf`, установить или изменить значение параметра.

Для просмотра всех активных настроек используется следующая команда:

```
$ sudo pg_conftool <версия> <имя_кластера> show all
```

Для просмотра значения заданного параметра используется следующая команда:

```
$ sudo pg_conftool <версия> <имя_кластера> show
 ↪  <имя_параметра>
```

Для установки значения параметра конфигурации используется следующая команда:

```
$ sudo pg_conftool <версия>  <имя_кластера> set
 ↪  <имя_параметра> <значение>
```

Для удаления параметра конфигурации используется следующая строка:

```
$ sudo pg_conftool <версия> <имя_кластера> reset
 ↪  <имя_параметра>
```

6.1.2. Создание кластера — `pg_createcluster`

Для создания кластера СУБД используется утилита `pg_createcluster`. Она принимает версию кластера и его имя и создает кластер с настройками по умолчанию. Работа утилиты завершается вызовом утилиты `initdb` с определенными параметрами размещения каталогов конфигурационных файлов и каталога данных, а также локали, используемой в ОС. Также существует возможность указать параметры конфигурации кластера, которые будут переданы команде `initdb`.

Формат вызова:

```
$ sudo pg_createcluster [опции] <версия>  <имя_кластера> [--
 ↪  опции_initdb ]
```

6.1.3. Управление кластером — `pg_ctlcluster`

Для управления кластером СУБД используется утилита `pg_ctlcluster`. Утилита принимает версию кластера, его имя и одно из следующих действий:

- `start` — запуск кластера;
- `stop` — остановка кластера;
- `restart` — перезапуск кластера;
- `reload` — перечитать конфигурации без остановки кластера;
- `status` — вывод статуса кластера;
- `promote` — переводит резервный сервер в режим основного.

Работа утилиты завершается вызовом утилиты `pg_ctl`, которой переданы корректные пути расположения каталогов кластера.

Формат вызова:

```
$ sudo pg_ctlcluster [опции] <версия>  <имя_кластера>
 ↪  <действие> [ -- опции_pg_ctl ]
```

6.1.4. Удаление кластера — `pg_dropcluster`

Для удаления кластера используется утилита `pg_dropcluster`. Она принимает версию кластера и его имя, после чего выполняет удаление каталогов указанного кластера. Дополнительно может быть указана опция `--stop`. В таком случае удаление кластера произойдет после его остановки.

Формат вызова:

```
$ sudo pg_dropcluster <версия> <имя_кластера> [ --stop ]
```

6.1.5. Просмотр состояние кластеров — `pg_lscluster`

Для просмотра состояния о существующих кластерах используется утилита `pg_lscluster`.

Формат вызова:

```
$ pg_lscluster [опции] [ версия [ имя_кластера ] ]
```

6.1.6. Переименование кластера — `pg_renamecluster`

Для переименования кластера используется утилита `pg_renamecluster`. Она принимает версию кластера, его текущее имя и новое имя. Утилита переименовывает каталоги кластера, а также пути конфигурационных файлов в `postgresql.conf`.

Формат вызова:

```
$ sudo pg_renamecluster <версия> <имя_кластера>
 ↪  <новое_имя_кластера>
```

6.1.7. Обновление кластера — `pg_upgradecluster`

Утилита `pg_upgradecluster` выполняет обновление существующего кластера на последнюю версию сервера по умолчанию. Она принимает версию кластера и его имя. Утилита создает кластер с таким же именем, как и существующий, выполяет выгрузку дампа с помощью `pg_dumpall` и восстанавливает его на новый кластер.

Формат вызова:

```
$ sudo pg_upgradecluster [опции] <версия>  <имя_кластера> [
 ↪  <новый_каталог_данных> ]
```

6.2. Резервирование и восстановление

Резервирование и восстановление резервных копий может быть выполнено для всего кластера или отдельно выбранной базы данных.

Для создания резервной копии требуется подключиться администратором СУБД и выполнить операцию создания резервной копии кластера:

```
$ pg_dumpall -h <хост> -U <пользователь> -p <порт> > dump.sql
```

Данная утилита создаст дамп кластера в формате SQL.

Для создания резервной копии одной базы данных используется утилита `pg_dump`. Данная утилита позволяет сделать резервную копию в следующих форматах:

- `c` — бинарный формат (по умолчанию);
- `p` — формат текстового файла на языке SQL;
- `d` — формат директории;
- `t` — формат архива (tar).

Указать формат, в котором требуется выполнить создание резервной копии можно с помощью ключа `-F` (или `--format`):

```
$ pg_dump --format c  -h <хост> -U <пользователь>  -p <порт>
 ↪  --format c > dump
```

Восстановление резервной копии, сделанной в текстовом формате производится с помощью команды:

```
$ psql -h <хост> -U <пользователь> -p <порт> -f <дамп.sql>
```

Восстановление резервной копии, сделанной в других форматах (бинарном, директории или архивном) осуществляется с помощью утилиты `pg_restore`:

```
$ pg_restore -h <хост>  -U <пользователь> -p <порт> -d
 ↪  <целевая_бд> <дамп>
```

Подробное описание описанных в данном разделе утилит приведено в `man`.

6.3. Управление объектами баз данных и ролями

Для управления объектами баз данных и ролями могут быть использована графический инструмент управления `pgadmin3` и консольная утилита `psql`.

Утилита `psql` — интерактивный терминал, позволяющий выполнять управление объектами баз данных с помощью языка SQL.

Для подключения к базе данных требуется выполнить команду:

```
$ psql -h <хост> -p <порт> -U <пользователь> -d <БД>
```

После успешного подключения к БД, вводятся команды языка SQL. Подробно о синтаксисе команд SQL описано в официальной документации по PostgreSQL.

Для создания нового пользователя в базе данных в интеративном терминале `psql` вводится следующая команда:

```
CREATE USER <имя_пользователя>;
```

Для создания новой базы данных в интерактивный терминал `psql` вводится следующая команда:

```
CREATE DATABASE <имя_бд>;
```

6.4. Управление разграничением доступа

Дискреционное управление доступа осуществляется посредством назначения списка разрешенных действий для роли на объект БД с помощью конструкций языка SQL.

Для делегирования дискреционных прав доступа к объектам используется команда SQL `GRANT`, а для отмены — команда `REVOKE`. Например, если в системе существует пользователь `user1`, то ему может быть предоставлено право на изменение данных в таблице `Данные` с помощью следующей команды:

```
GRANT UPDATE ON "Данные" TO user1;
```

Для предоставления прав доступа к объекту сразу всем пользователям системы существует специальное «имя пользователя» PUBLIC, а для предоставления всех прав — специальное «право» ALL. Например, чтобы дать всем пользователям полный доступ к таблице Данные, следует использовать следующую команду:

```
GRANT ALL ON "Данные" TO user1;
```

Для различных объектов БД определены разные наборы из доступных прав доступа:

- CONNECT — установка соединения с БД;
- CREATE — создание объекта в БД, схеме или табличном пространстве;
- USAGE — использование объектов, как правило не содержащих непосредственно данные (БД, схемы, языки и т.п.);
- SELECT — чтение данных из таблицы или представления;
- INSERT — вставка новых данных;
- DELETE — удаление некоторых/всех данных в таблице;
- UPDATE — изменение данных;
- REFERENCES — использование данных таблицы для внешних ключей;
- TRIGGER — создание и назначение для таблицы триггеров;
- TRUNCATE — очистка таблицы (удаление всех данных);
- EXECUTE — исполнение хранимой процедуры или функции.

Более подробно синтаксис управления дискреционным разграничением доступа и применяемые к объектам БД права подступа описаны в документации на язык SQL PostgreSQL.

6.5. Мандатное управление доступом к объектам БД

Мандатное управление доступа осуществляется на основе контекста сессии пользователя СУБД, который определяется по входящему соединению. Все создаваемые данные пользователя, включая метаданные, наследуют мандатную метку сессии пользователя.

При обработке запроса пользователя выполняется проверка возможности выполнения заданной операции согласно сопоставлению мандатного уровня доступа

пользователя (мандатной метки его сессии) с мандатной меткой объекта БД. В случае допустимости операции и наличии меток на записях в таблице БД, аналогичным образом производиться проверка возможности проведения операции над конкретной записью в таблице.

Правила мандатного разграничения доступа распространяются и применяются не только к данным, но и к метаданным (описанию структуры БД).

ВНИМАНИЕ! Применение мандатного разграничения доступа к метаданным ведет к различному представлению структуры БД при обращении к СУБД с разными уровнями доступа. Прикладное ПО должно быть разработано с учетом этого и обеспечивать корректную обработку ошибок подобного рода.

Согласно применяемой в ОС модели мандатного разграничения доступа дополнительно к мандатной метке конфиденциальности вводится понятие объектов-контейнеров (объектов, которые могут содержать другие объекты). Для задания способа доступа к объектам внутри контейнеров используется мандатный признак CCR (Container Clearance Required). В случае когда он установлен, доступ к контейнеру и его содержимому определяется мандатной меткой конфиденциальности контейнера, в противном случае доступ к содержимому разрешен без учета уровня конфиденциальности контейнера. Метка объекта не может превышать метку контейнера, в котором он содержится.

В качестве главного контейнера выбрано табличное пространство `pg_global`.

Применение мандатного разграничения доступа к объектам осуществляется одновременно с проверкой дискреционных прав доступа к ним во время разбора запроса. Для записей мандатное разграничение доступа выполняется непосредственно при работе методов доступа к ним (последовательных или индексных). При этом все множество операций с данными и метаданными рассматривается в рамках мандатного разграничения следующим образом:

- `INSERT` — доступ на запись;
- `UPDATE`, `DELETE` — последовательное выполнение доступа на чтение и запись информации;
- `SELECT` — доступ на чтение.

- CREATE, ADD — доступ на запись;
- ALTER, DROP — последовательное выполнение доступа на чтение и запись информации;
- использование или обращение к объекту в других SQL-командах — доступ на чтение.

В случае добавления данных, записи, помещаемые в таблицы, для которых установлена защита на уровне записей, наследуют текущую метку пользователя. При модификации данных записи сохраняют свою метку.

ВНИМАНИЕ! Предусмотрены системные привилегии игнорирования мандатного разграничения доступа для администраторов и ролей резервного копирования, поскольку только таким образом можно производить регламентные работы с БД (например, восстановление резервной копии), т. к. это требует установки меток данных, сохраненных ранее.

При создании объекта БД мандатная метка наследуется с метки текущей сессии пользователя, при этом признак CCR контейнера выставляется в значение ON.

Существует набор санкционированных команд изменения правил мандатного разграничения доступа с помощью конструкций языка SQL, позволяющий администратору изменять и назначать мандатные метки объектам БД.

Если текущая сессия обладает привилегией ac_capable_chmac, то существует возможность задать мандатную метку с помощью следующей команды:

```
MAC LABEL ON <тип объекта> <имя> IS <мандатная метка>;
```

Мандатный признак CCR для объектов-контейнеров может быть изменен командой:

```
MAC CCR ON <тип объекта> <имя> IS { ON | OFF };
```

Мандатная метка и признак CCR кластера задается следующими запросами:

```
MAC LABEL ON CLUSTER IS <мандатная метка>;
MAC CCR ON CLUSTER IS { ON | OFF };
```

или

```
MAC LABEL ON TABLESPACE pg_global IS <мандатная метка>;
MAC CCR ON TABLESPACE pg_global IS { ON | OFF };
```

7. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

В соответствии с нормативными документами СУБД «Лира-Р» содержит следующие доработки по отношению к стандартной версии PostgreSQL:

- добавлен метод аутентификации по алгоритму ГОСТ Р 34.11-2012;
- добавлена подсистема регистрации событий пользователей СУБД;
- добавлена подсистема очистки памяти СУБД.

Настройка метода аутентификации по алгоритму ГОСТ Р 34.11-2012 приведена в 7.1.

Описание и настройка подсистемы регистрации событий приведена в 7.2.

Описание и настройка подсистемы очистки памяти приведена в 7.3.

7.1. Аутентификация по хешу пароля ГОСТ Р 34.11-2012

В дополнение к уже существующим методам аутентификации парольной аутентификации добавлен метод `gost`, шифрующий пароль по алгоритму ГОСТ Р 34.11-2012 с длиной 256 бит при его передаче серверу. Сервер сверяет хеш пароля пользователя, хранящегося в таблице `pg_authid`, с паролем, присланным от клиента. При совпадении хешей доступ к СУБД разрешается.

Для настройки аутентификации по алгоритму `gost` необходимо:

1) В конфигурационном файле сервера `postgresql.conf` установить значение параметра шифрования паролей `password_encryption` значение `gost`;

2) В конфигурационном файле аутентификации сервера `pg_hba.conf` установить метод аутентификации `gost`;

3) Перезагрузить сервер СУБД;

4) Администратором СУБД сменить или назначить пароли пользователей.

7.2. Регистрация событий пользователей

В СУБД используется специально разработанная подсистема регистрации событий пользователей, используемая для протоколирования действий пользователей СУБД согласно правилам регистрации событий.

Под событием понимается текстовое сообщение, содержащее следующую информацию:

- метка времени;

– тип события;

– идентификатор пользователя (текущее имя пользователя, имя пользователя сессии);

– результат события (успех или отказ);

– выполняемый пользователем SQL запрос.

При авторизации пользователя определяется маска аудита сессии на основании заданных правил и устанавливается в атрибуте сессии `ac_session_audit`.

Загрузка маски аудита сессии производится в следующем порядке:

– Настройка для указанного роли и указанной СУБД;

– Настройка для указанной роли;

– Настройка для указанной БД;

– Настройка для всех остальных.

Маска аудита представляет собой текстовую строку вида `{УСПЕХ:ОТКАЗ}`, где `УСПЕХ` — список событий для регистрации успешных действий, а `ОТКАЗ` — список неуспешных событий.

Каждое событие в списках регистрации задается с помощью буквенного кода. Сопоставление между буквенным кодом и типом событий приведено в таблице 3.

Таблица 3

Событие	Символ	Описание
SUBJECT	S	Модификация состава субъектов кластера
CONFIGURATION	s	Модификация конфигурации кластера
RIGHTS	r	Модификация прав доступа к объектам БД
SELECT	r	Выборка данных
INSERT	a	Вставка данных
UPDATE	w	Модификация данных
DELETE	d	Удаление данных
TRUNCATE	D	Очистка таблицы
CREATE	C	Создание объектов БД
CREATE_TEMP	T	Создание временных таблиц
DROP	E	Удаление объектов БД
ALTER	M	Модификация объектов БД
CHMAC	m	Изменение мандатных свойств объектов БД

Окончание таблицы 3

Событие	Символ	Описание
CONNECT	c	Подключение к БД
DISCONNECT	e	Отключение от БД
Зарезервированный символ	*	Полная маска
Зарезервированный символ	*	Нулевая маска

Для назначения маски событий используется команда ALTER ROLE:

```
ALTER ROLE { ALL | имя_роли } [ IN DATABASE имя_базы_данных ]
SET ac_session_audit TO новое_значение;
```

Для удаления списка регистраций событий используется следующая команда:

```
ALTER ROLE { ALL | имя_роли }  [ IN DATABASE имя_базы_данных ]
RESET ac_session_audit;
```

П р и м е ч а н и е. Для выполнения приведенных команд требуются права администратора.

П р и м е ч а н и е. При инициализации кластера баз данных автоматически добавляется следующее правило:

```
ALTER ROLE ALL SET ac_session_audit TO '{ce:*}';
```

Для просмотра значения маски аудита используется команда:

```
SELECT current_audit;
```

Параметры функционирования подсистемы регистрации событий приведены в таблице 4.

Т а б л и ц а 4

Параметр	Описание
audit_enable	Подсистема регистрации события включена? По умолчанию включена.
audit_output_level	Режим записи событий безопасности: - statement — запись событий происходит после каждого запроса; - transaction — запись событий происходит после завершения транзакции. По умолчанию используется значение statement.

Окончание таблицы 4

Параметр	Описание
audit_destination	Определяет журналы записи событий безопасности СУБД: - internal — запись событий в журнал СУБД; - external — запись событий в централизованный журнал безопасности ОС; - all — запись событий безопасности как в журнал СУБД, так и в журнал безопасности ОС. По умолчанию используется значение internal.
audit_log_only_failures	Если активен, то регистрируются только отказы. По умолчанию фиксируются все события.
audit_stmt_length	Длина запроса, который фиксируется в событии. Если запрос превышает эту величину, то строка запроса усекается до этого значения. Значение 0 снимает ограничение на длину запроса. По умолчанию используется значение 100.

7.3. Очистка памяти

В СУБД реализована настраиваемая подсистема очистки памяти пользовательских процессов.

Режим работы очистки памяти настраивается параметром wipe_memory конфигурационного файла postgresql.conf и изменяется только после перезапуска сервера СУБД.

Он может принимать одно из нескольких значений:

- buffers — очистка буферов отношений, сброшенных на жесткий диск;
- pages — очистка страниц отношений, открытых процессом СУБД;
- files — очистка файлов отношений после применения команд VACUUM или TRUNCATE.

8. СООБЩЕНИЯ АДМИНИСТРАТОРУ

Работа сервера СУБД протоколируется в лог-файл. Кроме этого, существует возможность настройки протоколирования сообщений от сервера СУБД в системный лог, например `syslog`.

За расположение каталога с лог-файлами сервера отвечает параметр `log_directory`, за вид файла лога — `log_filename` файла `postgresql.conf`. Дополнительно могут быть настроены фиксация выполняемых SQL запросов, уровень детализации сообщений и пр.

В лог файл фиксируются моменты запуска и остановки сервера СУБД, ошибки и прочие сообщения согласно настроенному уровню детализации.

Дополнительным источником сообщений администратору могут являться стандартные потоки ввода и вывода информации клиентских утилит командной строки.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных

ОС — операционная система

СУБД — система управления базами данных

CCR — Container Clearance Required

WAL — Write Ahead Log (журнал упреждающей записи)

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					