ОПЕРАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ «ASTRA LINUX SPECIAL EDITION» РУСБ.10015-01

Руководство по КСЗ. Часть 3. Защищенная СУБД

Оперативное обновление 1.8.3

Бюллетень № 2025-0811SE18

Листов 14

2025

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

В настоящем документе приведены кумулятивные изменения в документ РУСБ.10015-01 97 01-3 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 3. Защищенная СУБД» из комплектности изделия РУСБ.10015-01 «Операционная система специального назначения «Astra Linux Special Edition» (далее по тексту — ОС).

При администрировании СУБД с установленным оперативным обновлением согласно бюллетеню № 2025-0811SE18 рекомендуется руководствоваться документом РУСБ.10015-01 97 01-3 совместно с настоящим документом.

Документ предназначен для администраторов и разработчиков баз данных.

2. ПЕРЕЧЕНЬ ИЗМЕНЕНИЙ

2.1. Подраздел «1.2. Состав»

В подразделе 1.2 в таблице 1 добавить описание дополнительно поставляемых модулей:

Список дополнительно поставляемых модулей приведен в таблице 1:

Таблица 1 – Дополнительно поставляемые модули

Модуль	Описание
credcheck	Позволяет задавать требования к учетным данным при создании, переименовании и смене пароля пользователя. Например, установить минимальную длину пароля
dbcopies_decoding	Предоставляет слоты логической репликации при использовании механизма копирования баз данных 1С
fasttrun	Сокращает время выполнения операций очистки таблиц
fulleq	Реализует операцию точного (побитового) сравнения значений
hypopg	Добавляет поддержку гипотетических/виртуальных индексов
mchar	Добавляет поддержку типов данных, которые совместимы с Microsoft SQL Server
online_analyze	Автоматически выполняет команду анализа ANALYZE после выполнения команд INSERT, UPDATE, DELETE, SELECT, INTO для затронутых таблиц
page_repair	Восстанавливает отдельные поврежденные страницы с использованием резервных данных, хранящихся на сервере репликации
pgaudit	Осуществляет детальный аудит сеансов и/или объектов через стандартное средство ведения журнала регистрации событий СУБД
pgauditlogtofile	Дополнение к модулю pgaudit. Перенаправляет ведение журнала аудита в отдельный файл, вместо использования стандартного средства ведения журнала регистрации событий СУБД
pg_background	Обеспечивает выполнение команд в фоновом режиме с использованием автономных транзакций, что позволяет не блокировать выполнение других команд в текущей сессии
pg_cron	Планировщик заданий на основе Cron, который работает внутри БД и использует тот же синтаксис
pg_repack	Позволяет устранять фрагментацию (BLOAT) таблиц и индексов, а также восстанавливать порядок строк по исходному индексу. В отличие от команд CLUSTER и VACUUM FULL, модуль не требует полной блокировки (ACCESS EXCLUSIVE) при выполнении действий с таблицами
pg_store_plans	Предоставляет средства анализа планов выполнения всех SQL-запросов, выполняемых сервером СУБД
pg_wait_sampling	Предоставляет информацию о событиях ожидания текущих процессов

2.2. Подраздел «5.1. Порядок применения мандатных правил управления доступом»

В подразделе 5.1 приведенный абзац изложить в следующей редакции и после него добавить новый абзац:

Для администратора БД предусмотрены PARSEC-привилегии игнорирования мандатного управления доступом, только таким образом можно производить регламентные работы с БД (например, восстановление резервной копии), т. к. это требует установки меток данных, сохраненных ранее. Подробное описание PARSEC-привилегий приведено в документе РУСБ.10015-01 97 01-1.

В условиях применения мандатного управления доступом пользователь `postgres` в СУБД обладает следующими PARSEC-привилегиями:

- `PARSEC_CAP_IGNMACLVL`;
- `PARSEC_CAP_IGNMACCAT`;
- `PARSEC_CAP_CHMAC`;
- `PARSEC_CAP_SETMAC`.

2.3. Подраздел «10.1. Настройка репликации»

В подраздел 10.1 добавить пункты 10.1.4 и 10.1.5 и изложить их в следующей редакции:

10.1.4. Настройка логической репликации

Логическая репликация основана на передаче изменений данных от ведущего (публикующего) сервера на уровне SQL-команд и их применении на подписывающем сервере (подписчике).

Особенности и ограничения технологии логической репликации:

- репликация поддерживается только для таблиц. Представления, материализованные представления и другие объекты не поддерживаются;
- в секционированных таблицах изменения реплицируются от имени конечных секций;
- команда `TRUNCATE` применяется только к тем таблицам, которые входят в подписку;
- данные последовательностей не обновляются автоматически на подписывающем сервере (подписчике). Их следует актуализировать вручную;
- схемы базы данных, команды создания и изменения таблиц не реплицируются. Схему базы данных необходимо предварительно скопировать вручную;
- данные, хранящиеся как большие объекты (Large Objects), не реплицируются.

Для настройки логической репликации необходимо установить параметры на ведущем и подписывающем серверах.

На ведущем сервере установить параметры:

- `wal_level = logical` — включает механизм логической репликации;
- `max_replication_slots` — определяет максимально допустимое количество слотов репликации. Значение должно быть не меньше суммы подписывающих серверов и резерва для временных слотов, которые создаются при синхронизации таблиц;
- `max_wal_senders` — определяет максимальное количество одновременно работающих процессов передачи WAL. Значение должно быть не меньше суммы значений параметра `max_replication_slots` и резерва для физических серверов репликации при необходимости;
- `max_worker_processes` — определяет общее количество фоновых рабочих процессов. Значение должно быть не меньше суммы значений параметра `max_wal_senders` и резерва для других ожидаемых рабочих процессов.

На подписывающем сервере установить параметры:

- `max_logical_replication_workers` — определяет количество рабочих процессов логической репликации. Значение должно быть не меньше суммы значений активных подписок на публикации и резерва для фоновой синхронизации;
- `max_worker_processes` — определяет общее количество фоновых рабочих процессов. Значение должно быть не меньше суммы значения параметра `max_logical_replication_workers` и резерва для других ожидаемых рабочих процессов.

Конкретные параметры и их значения, а также порядок действий по настройке могут отличаться в зависимости от архитектуры системы, требований безопасности и особенностей использования.

Пример

Для настройки логической репликации выполнить следующие действия:

1) на ведущем сервере в конфигурационном файле `/etc/postgresql/<версия>/<кластер>/postgresql.conf` задать следующие параметры:

```
wal_level = logical
max_wal_senders = 10
max_replication_slots = 10
max_worker_processes = 16
```

2) на ведущем сервере в конфигурационном файле /etc/postgresql/<версия_СУБД>/<имя_кластера>/pg_hba.conf добавить разрешение на подключение для пользователя, выполняющего репликацию:

```
host replication $REPL $SLAVE_IP scram-sha-256
```

3) на подписывающем сервере в конфигурационном файле /etc/postgresql/<версия>/<кластер>/postgresql.conf задать следующие параметры:

```
max_logical_replication_workers = 8
max_worker_processes = 16
```

4) перезапустить кластеры баз данных:

а) на ведущем сервере:

```
sudo pg_ctlcluster $VERSION $MASTER restart
```

б) на подписывающем сервере:

```
sudo pg_ctlcluster $VERSION $SLAVE restart
```

5) на ведущем сервере создать пользователя репликации:

```
CREATE ROLE $REPL WITH LOGIN REPLICATION PASSWORD '$REPL_PASSWORD';
```

6) на ведущем сервере в базе данных $MASTER_DB создать источник данных репликации:

```
CREATE PUBLICATION my_publication FOR ALL TABLES;
```

В случае использования мандатного управления доступом дополнительно задать уровень и категорию конфиденциальности для источника данных репликации:

```
MAC LABEL ON PUBLICATION my_publication IS
'{<уровень_конфиденциальности>,<категория_конфиденциальности>}';
```

7) на ведущем сервере создать слот для логической репликации:

```
SELECT * FROM pg_create_logical_replication_slot
('repl_slot', 'pgoutput');
```

В случае использования мандатного управления доступом дополнительно задать уровень и категорию конфиденциальности для слота:

```
MAC LABEL ON SLOT repl_slot IS
'{<уровень_конфиденциальности>,<категория_конфиденциальности>}';
```

8) на подписывающем сервере создать пользователя с правами администратора БД, от имени которого планируется создать получателя данных репликации (подписку):

```
CREATE ROLE $REPL WITH LOGIN PASSWORD '$REPL_PASSWORD';
ALTER ROLE $REPL WITH SUPERUSER;
```

9) на подписывающем сервере подключиться к базе данных $SLAVE_DB пользователем $REPL и создать получателя данных репликации (подписку) в отключенном состоянии, без создания слота:

```
CREATE SUBSCRIPTION my_subscription
CONNECTION 'host=$MASTER_IP port=$MASTER_PORT dbname=$MASTER_DB
user=$REPL password=$REPL_PASSWORD'
PUBLICATION my_publication WITH
(create_slot = false, enabled = false);
```

В случае использования мандатного управления доступом дополнительно задать уровень и категорию конфиденциальности для получателя данных репликации:

```
MAC LABEL ON SUBSCRIPTION my_subscription IS
'{<уровень_конфиденциальности>,<категория_конфиденциальности>}';
```

10) назначить слот для получателя данных репликации (подписки):

```
ALTER SUBSCRIPTION my_subscription SET (slot_name = 'repl_slot');
```

11) включить подписку:

```
ALTER SUBSCRIPTION my_subscription ENABLE;
```

12) если используется мандатное управление доступом, то на ведущем сервере в базе данных $MASTER_DB задать уровень и категорию конфиденциальности на слоте логической репликации:

```
MAC LABEL ON SLOT repl_slot IS
'{<уровень_конфиденциальности>,<категория_конфиденциальности>}';
```

13) проверить состояние репликации:

а) на ведущем сервере:

```
SELECT * FROM pg_stat_replication;
```

б) на подписывающем сервере:

```
SELECT * FROM pg_stat_subscription;
```

Примечания:

1. Классификационная метка подписки (SUBSCRIPTION) задает метку сессии процесса, который получает и применяет данные репликации (logical replication worker). Эта метка должна входить в допустимый диапазон классификационных меток пользователя СУБД, от имени которого создается подписка и запускается процесс logical replication worker. Если метка

выходит за пределы диапазона, то процесс репликации не запустится. Допустимый диапазон меток пользователя СУБД формируется установкой минимальной и максимальной классификационных меток соответствующего пользователя в ОС.

2. Классификационная метка публикации (PUBLICATION) не связана с меткой процесса, который на стороне публикации считывает и отправляет данные. Данный процесс запускается с максимальной меткой из допустимого диапазона пользователя с именем $REPL, указанного в параметре CONNECTION. Допустимый диапазон меток пользователя СУБД формируется установкой минимальной и максимальной классификационных меток соответствующего пользователя в ОС.

3. При использовании мандатного управления доступом изменение классификационной метки для получателя данных репликации (подписки) возможно только при остановленном процессе логической репликации. При попытке изменить метку при активной подписке будет выдано сообщение об ошибке. Для изменения метки необходимо предварительно отключить подписку командой:

```
ALTER SUBSCRIPTION my_subscription DISABLE;
```

10.1.5. Особенности логической репликации при использовании мандатного управления доступом

Если для логической репликации применяется пользователь postgres, то она выполняется без ограничений и охватывает все таблицы, включая защищенные (см. 5.1). Если для репликации применяются пользователи, которым в ОС предоставлены привилегии игнорирования уровня и категорий конфиденциальности, то логическая репликация также выполняется без ограничений по доступу ко всем таблицам. Поэтому для репликации не рекомендуется применять пользователя postgres или пользователей, имеющих привилегии игнорирования уровня и категорий конфиденциальности.

Для логической репликации таблиц, на которых выключен строковый контроль доступа, допускается использование пользователей без PARSEC-привилегий при соблюдении следующих условий:

- на стороне источника данных репликации (публикующем сервере) уровень конфиденциальности пользователя должен быть не ниже уровня конфиденциальности таблицы, а категории конфиденциальности пользователя должны включать все категории конфиденциальности таблицы;
- на стороне получателя данных репликации (подписки) уровень и категории конфиденциальности пользователя должны совпадать с уровнем и категориями конфиденциальности таблицы.

Для логической репликации таблиц, на которых включен строковый контроль доступа, пользователь на стороне получателя данных дополнительно должен:

- обладать привилегией PARSEC_CAP_CHMAC;

- иметь диапазон классификационных меток, позволяющий устанавливать метку сессии на любые уровни и категории конфиденциальности, встречающиеся в строках реплицируемой таблицы. Допустимый диапазон меток пользователя СУБД формируется установкой минимальной и максимальной классификационных меток соответствующего пользователя в ОС.

Если в таблице установлен флаг CCR=ON, то необходимо явно задать уровень и категорию конфиденциальности для получателя данных репликации (подписки) следующей командой:

```
MAC LABEL ON SUBSCRIPTION <имя_подписки> IS
'{<уровень_конфиденциальности>,<категория_конфиденциальности>}';
```

При этом уровень и категория конфиденциальности получателя данных должны быть не ниже уровня и категории конфиденциальности таблицы. Это необходимо, потому что процесс репликации запускается с классификационной меткой сессии, совпадающей с классификационной меткой получателя данных репликации.

Примеры:

1. Репликация таблицы с классификационной меткой {0,0} может выполняться пользователем с классификационной меткой {0,0}. Для репликации таблицы с классификационной меткой {2,0} возможно использовать пользователя с классификационной меткой {3,0} на стороне источника данных и классификационной меткой {2,0} на стороне получателя данных.

2. Осуществляется репликация таблицы с классификационной меткой {5,0}, содержащей строки с классификационными метками {2,0}, {3,0} и {4,0}. В данном случае на стороне получателя данных минимальная классификационная метка допустимого диапазона пользователя в ОС не должна быть выше {2,0}, а максимальная классификационная метка в ОС не должна быть ниже {4,0} при CCR=OFF или {5,0} при CCR=ON. Пользователь на стороне получателя данных должен обладать привилегией PARSEC_CAP_CHMAC. На стороне источника данных пользователь должен иметь максимальную классификационную метку допустимого диапазона не ниже классификационной метки таблицы {5,0} при CCR=ON или не ниже самой секретной строки {4,0} при CCR=OFF.

2.4. Раздел «13. Регистрация событий в СУБД»

Изложить раздел 13 в следующей редакции:

13. Регистрация событий в СУБД

13.1. Регистрация событий средствами ОС

В СУБД регистрация событий безопасности выполняется с учетом требований ГОСТ Р 59548-2022. Регистрация событий безопасности, настройка реагирования системы на события и информирование администратора осуществляется подсистемой регистрации событий из состава ОС. Описание подсистемы регистрации событий и журнала событий приведено в документе РУСБ.10015-01 97 01-1.

Подсистема регистрации событий собирает информацию о событиях безопасности, связанных с функционированием СУБД и действиями пользователей СУБД, и обеспечивает их хранение в журнале событий безопасности, а также предоставляет инструменты для просмотра собранных данных и реагирования на события.

13.2. Встроенные средства регистрации событий в СУБД

13.2.1. Режимы регистрации событий

В СУБД доработаны встроенные средства регистрации событий. Для установки режима регистрации событий используется параметр `ac_audit_mode` конфигурационного файла `/etc/postgresql/<версия>/<кластер>/postgresql.conf`. Изменение данного параметра применяется только после перезапуска кластера СУБД. Параметр может принимать значения:

- `default` — настройки регистрации событий заданы по умолчанию. Для администратора СУБД, от имени которого создается кластер, регистрируются все события. Для всех остальных пользователей регистрируются только подключения и завершения сессий, а также все неуспешные события;
- `external` — настройки регистрации событий задаются в конфигурационном файле `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf` с помощью маски регистрации событий;
- `none` — регистрация событий отключена.

После обновления предыдущих версий СУБД следует проверить и при необходимости отредактировать файлы `/etc/postgresql/<версия>/<кластер>/postgresql.conf` и `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf`:

1) в файле `/etc/postgresql/<версия>/<кластер>/postgresql.conf`:
   а) проверить значение параметра `ac_audit_mode` и при необходимости установить одно из допустимых значений: `default`, `external` или `none`;
   б) убедиться, что параметр `ac_audit_destination` имеет одно из допустимых значений: `logfile` или `parsec`;
2) в файле `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf`:
   а) проверить, что маски регистрации событий заданы с использованием символьных значений;

б) исключить использование числовых значений масок регистрации событий, так как они не поддерживаются.

13.2.2. Установка маски регистрации событий

В СУБД маска регистрации событий устанавливается при инициализации сессии пользователя на основании текущего режима регистрации событий (см. 13.2.1). Маска регистрации событий может быть изменена администратором только в режиме регистрации событий `external` в конфигурационном файле `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf` (см. 13.2.3).

Маска регистрации событий задается с помощью символов (см. таблицу 11).

Таблица 11

Событие	Символ	Описание
SUBJECT	S	Добавление/изменение/удаление пользователей и групп
CONFIGURATION	s	Изменение конфигурации, влияющей на доступ к данным (запрос на изменение значения переменной `ac_session_maclabel`)
RIGHTS	R	Запрос на модификацию прав доступа к объектам БД
CHECK_RIGHTS	V	Проверка прав доступа
SELECT	r	Выборка информации из БД
INSERT	a	Добавление информации в БД
UPDATE	w	Изменение информации в БД
DELETE	d	Удаление информации из БД
TRUNCATE	D	Очистка данных
REFERENCES	x	Задание столбца таблицы в качестве внешнего ключа
TRIGGER	t	Добавление триггера к таблице
EXECUTE	X	Запуск хранимой процедуры или триггера
USAGE	U	Использование объекта БД
CREATE	C	Создание объектов в БД
CREATE TEMP	T	Создание временного объекта
DROP	E	Удаление объектов БД
ALTER	M	Изменение объекта БД
CHMAC	m	Изменение мандатных атрибутов защищаемых объектов в БД
CONNECT	c	Запрос на начало сессии
DISCONNECT	e	Запрос на окончание сессии
Зарезервированный символ	*	Полная маска регистрации событий

Маска регистрации событий имеет вид `{УСПЕХ:ОТКАЗ}`, где `УСПЕХ` — список символов для успешных событий, `ОТКАЗ` — список символов для неуспешных событий.

Пример

```
{SsRawdCTEMce:ce}
```

Пользователю для просмотра маски регистрации событий сессии необходимо в консоли СУБД выполнить команду:

```
SHOW ac_session_audit;
```

Вывод маски регистрации событий также осуществляется в символьном виде.

13.2.3. Назначение списков регистрации событий

В режиме регистрации событий `external` возможно изменять маски регистрации событий. Это позволяет самостоятельно задавать списки событий, подлежащих регистрации. Для назначения списков регистрируемых событий используется конфигурационный файл `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf` конкретного кластера СУБД. В данном файле задаются списки успешных и неуспешных запросов на доступ, которые будут регистрироваться в журнале СУБД и журнале аудита ОС для отдельных пользователей и по умолчанию. Информация о соединении пользователей с БД и разъединении с ней регистрируется всегда. Изменения в конфигурационном файле будут применены только после перезапуска кластера СУБД.

В файле `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf` значения параметров регистрации событий задаются в следующем формате:

```
success events mask = <значение> failure events mask = <значение>
user = <имя_пользователя> database = <имя_базы_данных>

success events mask = <значение> failure events mask = <значение>
user = <имя_пользователя>

success events mask = <значение> failure events mask = <значение>
database = <имя_базы_данных>

success events mask = <значение> failure events mask = <значение>
```

где `success events mask` — символьные значения для успешных запросов на доступ;
`failure events mask` — символьные значения для неуспешных запросов на доступ.

Примеры:

1. Аудит всех действий администратора СУБД:

```
success events mask = * failure events mask = * user = postgres
```

2. Для всех пользователей, подключающихся к БД `data`, выполняется регистрация всех неуспешных действий, а также событий подключения и отключения:

```
success events mask = ce failure events mask = * database = data
```

3. Для остальных пользователей выполняется регистрация всех неуспешных действий и всех успешных действий, кроме доступа к данным:

```
success events mask = SsRDxCTEMmce failure events mask = *
```

Настройки регистрации событий в конфигурационном файле `/var/lib/postgresql/<версия>/<кластер>/pg_audit.conf` рекомендуется задавать в следующем порядке:

1) настройки для конкретной роли и конкретной базы данных;
2) настройки для конкретной роли;
3) настройки для конкретной базы данных;
4) для всех остальных.

2.5. Подраздел «14.1. Создание и восстановление резервных копий баз данных с мандатными атрибутами»

В подразделе 14.1 после второго абзаца добавить абзац в следующей редакции:

Для резервной копии уровень конфиденциальности устанавливается равным максимально допустимому для текущей сессии. При этом если уровень конфиденциальности СУБД меньше уровня сессии или пользователь СУБД имеет PARSEC-привилегии игнорирования уровня конфиденциальности, то уровень конфиденциальности резервной копии устанавливается равным уровню конфиденциальности кластера СУБД. Подробное описание PARSEC-привилегий приведено в документе РУСБ.10015-01 97 01-1. Для резервной копии устанавливаются только те категории конфиденциальности, которые разрешены и в сессии, и в кластере. Это гарантирует, что копия будет доступна только субъектам, у которых есть доступ как по категориям конфиденциальности кластера, так и по категориям конфиденциальности сессии.