

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

**Система управления производственным предприятием**

Версия 2.1

Руководство системного программиста

**Листов 22**

## СОДЕРЖАНИЕ

1	Архитектура системы .....	3
2	Требования к операционной среде .....	3
2.1	ОС «Astra Linux Special Edition» .....	4
2.1.1	Назначение .....	4
2.1.2	Функциональные возможности .....	4
2.1.3	Требования к техническим средствам .....	4
3	Компоненты операционной среды .....	5
3.1	Система управления базами данных .....	5
3.2	Офисный пакет .....	5
3.3	Файловый сервер .....	5
4	Установка системы .....	6
4.1	Установка серверного программного обеспечения .....	6
4.1.1	Установка операционной среды .....	6
4.1.2	Установка СУБД .....	6
4.1.3	Создание БД .....	6
4.1.4	Установка сервера отчетов .....	10
4.2	Установка клиентского программного обеспечения .....	11
4.2.1	Установка операционной среды .....	11
4.2.2	Установка офисного пакета OpenOffice .....	11
4.2.3	Установка клиентского программного обеспечения .....	11
4.2.4	Подключение сервера отчетов .....	12
5	Резервное копирование и восстановление БД .....	12
5.1	Логическое резервирование и восстановление .....	12
5.1.1	Создание логической резервной копии БД .....	12
5.1.2	Восстановление БД из логической копии .....	13
5.2	Непрерывное архивирование и физическое резервирование .....	14
5.2.1	Настройка непрерывного архивирования журналов повтора .....	14
5.2.2	Создание физической копии БД .....	16
5.2.3	Восстановление БД из физической копии .....	18
	Приложение 1 .....	21

## **1 Архитектура системы**

Система имеет классическую клиент-серверную архитектуру (сервер приложений - клиент) и представляется следующими функциональными блоками:

- серверная часть, обеспечивающая управление данными и реализацию бизнес-логики системы;
- клиентская часть, обеспечивающая визуализацию данных пользователю.

В качестве серверной компоненты применяется СУБД Синергия-БД, разработанная на основе ПО с открытым исходным кодом PostgreSQL. СУБД в данном случае выполняет как роль классической системы управления базами данных, так и сервера приложений за счет реализации всех расчетных программ в стеке PG/SQL.

Клиентские компоненты представляют собой набор программных модулей, обеспечивающих взаимодействие пользователя с серверными компонентами и визуализацию результата выполнения расчетных процедур, а также офисных программ для работы с электронными документами в формате RTF и ODS.

## **2 Требования к операционной среде**

В качестве основной операционной среды, для построения типовой системы применяется программное изделие операционная система специального назначения «Astra Linux Special Edition» (далее ОС «Astra Linux SE»).

ОС «Astra Linux SE» разработана АО «НПО РусБИТех» и соответствует требованиям к ОС (А второго класса защиты. ИТ.ОС.А2.ПЗ) Схема сертификации серийная. Сертификат ФСТЭК №2557, действителен до 27.01.2021.

Для клиентских рабочих мест допускается использование ОС из семейства Windows, при выполнении требований класса защищённости аттестуемой АС.

## 2.1 ОС «Astra Linux Special Edition»

### 2.1.1 Назначение

ОС предназначена для построения автоматизированных систем в защищенном исполнении, обрабатывающих информацию, содержащую сведения, составляющие государственную тайну с грифом не выше «совершенно секретно».

### 2.1.2 Функциональные возможности

ОС предоставляет следующие возможности:

- установку и функционирование на современных серверах и рабочих станциях на платформах с процессорной архитектурой x86-64, а также поддержку современного периферийного оборудования;
- поддержку основных сетевых протоколов (TCP/IP, DHCP, DNS, FTP, TFTP, SMTP, IMAP, HTTP, NTP, SSH, NFS, SMB);
- организацию сетевого домена с централизованным хранением учетных записей;
- работу с мультимедийными данными;
- работу с реляционными БД;
- работу с электронной почтой;
- работу с гипертекстовыми данными;
- обработку текстовых документов и электронных таблиц различных форматов.

### 2.1.3 Требования к техническим средствам

Для функционирования ОС необходима следующая минимальная конфигурация оборудования:

- аппаратная платформа – процессор с архитектурой x86-64 (AMD, Intel);
- оперативная память – от 1 ГБ;
- объем свободного дискового пространства – от 4 ГБ;
- устройство чтения DVD-дисков;

- стандартный монитор SVGA 15”.

### 3 Компоненты операционной среды

#### 3.1 Система управления базами данных

В качестве сервера базы данных применяется СУБД Синергия-БД, разработанная на базе ПО с открытым исходным кодом PostgreSQL версии 11, скомпилированная с параметром FUNC\_MAX\_ARGS равным 200.

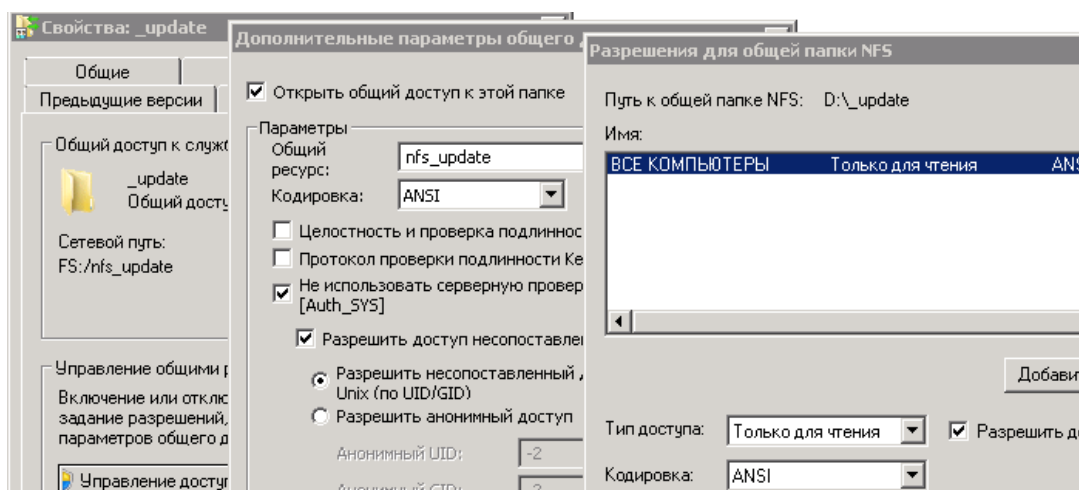
#### 3.2 Офисный пакет

Для обработки текстовых документов и электронных таблиц различных форматов в ОС «Astra Linux SE» и «Синергия» применяется OpenOffice, в ОС Windows Microsoft Office.

#### 3.3 Файловый сервер

Для организации доступа к шаблонам отчетов по сети можно использовать протокол NFS.

Для обеспечения доступа к ресурсам, размещенным на сетевой папке сервера под управлением ОС Windows, необходимо к роли «Файловый сервер» добавить службу «Службы для NFS». Пример настроек доступа по протоколу NFS показан ниже.



## **4 Установка системы**

### **4.1 Установка серверного программного обеспечения**

#### 4.1.1 Установка операционной среды

На сервере базы данных допускается установка ОС «Astra Linux SE» версии 1.5 и выше.

4.1.1.1 Установка ОС «Astra Linux SE» выполняется в соответствии с Руководством администратора, входящим в состав поставки программного продукта.

#### 4.1.2 Установка СУБД

В качестве сервера базы данных допускается использовать СУБД Синергия-БД, либо сборки СУБД PostgreSQL версии 11 и выше.

4.1.2.1 Установка и настройка СУБД Синергия-БД выполняется в соответствии с документом Руководство системного программиста 07623615.00082-01 32 01, входящим в состав поставки программного продукта.

Рекомендуется выполнить настройку политики управления паролями в соответствии с разделом 5.5 Руководства по комплексу средств защиты 07623615.00082-01 90 01, входящего в состав поставки программного продукта.

4.1.2.2 За информацией по установке СУБД PostgreSQL от другого производителя ПО необходимо обратиться на соответствующий Интернет-ресурс.

#### 4.1.3 Создание БД

##### 4.1.3.1 Настройка конфигурационных параметров СУБД

После установки СУБД на сервер и перед началом работы, рекомендуется в конфигурационном файле `/var/lib/sdb/sdb-11/data/postgresql.conf` модифицировать следующие параметры:

- `listen_addresses` установить в значение '\*', задаёт адреса TCP/IP, по которым сервер будет принимать подключения от клиентских приложений;

- `max_connections` – определяет максимально допустимое количество соединений с сервером БД, устанавливается исходя из планируемых потребностей;
- `tcp_keepalives_idle` – задаёт период неактивности (в секундах), после которого по TCP клиенту должен отправляться сигнал сохранения соединения, необходимо изменять, если в процессе работы происходит обрыв соединения после простоя или при длительных расчетах, по умолчанию данный параметр установлен в значение 2 часа;
- `tcp_keepalives_interval` – задаёт интервал (в секундах), по истечении которого следует повторять сигнал сохранения соединения, если ответ от клиента не был получен, необходимо изменять, если в процессе работы происходит обрыв соединения после простоя или при длительных расчетах;
- `shared_buffers` – задаёт объём памяти, который будет использовать сервер баз данных для буферов в разделяемой памяти, рекомендуемое значение 12ГБ;
- `work_mem` – задаёт объём памяти, который будет использоваться для внутренних операций сортировки и хеш-таблиц, прежде чем будут задействованы временные файлы на диске, рекомендуемое значение 32МВ;
- `maintenance_work_mem` – задаёт максимальный объём памяти для операций обслуживания БД, в частности `VACUUM`, `CREATE INDEX` и `ALTER TABLE ADD FOREIGN KEY`, рекомендуемое значение 1GB;
- `checkpoint_timeout` – максимальное время между автоматическими контрольными точками в WAL (в секундах), рекомендуемое значение 15min.

В процессе эксплуатации системы значения параметров, возможно, потребуется изменить, исходя из условий эксплуатации и нагрузки на СУБД.

В файле `/var/lib/sdb/sdb-11/data/pg_hba.conf` изменить строку подключений по локальному адресу (127.0.0.1) следующим образом:

```
host      all      all  127.0.0.1/32      trust
local    all      all  trust
```

Добавить дополнительную строку определяющую диапазон IP-адресов с которых возможно подключение к СУБД, например:

```
host      all      all  10.10.0.0/16      md5
```

Более подробную информацию по конфигурационным параметрам СУБД можно получить из документа Руководство системного программиста, входящим в состав поставки Синергия-БД.

#### 4.1.3.2 Подготовка БД

Создать директорию, например `/pgindex`, в которой будут размещены индексы базы данных, при необходимости смонтировать в нее дисковый раздел.

С помощью утилиты `psql` выполнить скрипт `pre_install.sql`, для этого необходимо выполнить команду:

```
root@localhost:~# psql -d postgres -h localhost -U postgres
--set=USERNAME=sysadm --set=PWD='\sysadm\' --set=NEWDBNAME=erp
--set=INDEXTABLESPACE=pgindex --set=INDEXDIR=\'/pgindex\'
-o ./output.log -f ./pre_install.sql
```

Данная команда принимает следующий набор параметров:

`-d` – задает имя базы данных, к которой происходит подключение, параметр не должен меняться;

`-h` – задает имя/IP-адрес сервера СУБД;

`-U` – имя супер-пользователя, от имени которого выполняется скрипт, параметр не должен меняться;

`--set=USERNAME` – задает имя пользователя-владельца создаваемой базы данных;

`--set=PWD` – задает пароль пользователя-владельца создаваемой базы данных;

`--set=NEWDBNAME` – имя создаваемой базы данных;



`--set=INDEXTABLESPACE` – имя табличного пространства, в котором размещаются индексы, в случае изменения имени при восстановлении базы данных будут выданы предупреждения, и индексы создадутся в табличном пространстве `pg_default`;

`--set=INDEXDIR` – задает путь к директории (необходимо предварительно создать, владельцем директории должен быть назначен пользователь `postgres:postgres`), где размещаются файлы табличного пространства, заданного параметром `INDEXTABLESPACE`;

`-o` – имя файла журнала выполнения команд скрипта;

`-f` – задает путь, где находится файл скрипта `pre_install.sql`.

В процессе выполнения скрипта создается новый пользователь-владелец, ему назначаются необходимые привилегии, создается база данных, в которой будут зарегистрированы необходимые дополнительные расширения PostgreSQL.

#### 4.1.3.3 Восстановление БД

Эталонный экземпляр системы поставляется в виде двоичного дампа-файла (или директории, содержащей набор файлов) базы данных.

Восстановление дампа выполняется в следующей последовательности:

Скопировать дамп-файл или директорию логической копии на сервер, где установлена СУБД (например, в папку `/tmp`).

Для корректного восстановления объектов БД из дампа-файла, необходимо наделить встроенную системную роль `dbsa` полномочиями суперпользователя, выполнив команду:

```
alter role dbsa superuser;
```

В дальнейшем восстановление БД будет выполняться в контексте данной роли. После завершения процедуры восстановления полномочия суперпользователя у роли `dbsa` рекомендуется изъять.

С помощью утилиты `pg_restore` выполнить восстановление базы данных и дампа-файла, для этого необходимо выполнить команду:

```
root@localhost:~# pg_restore --host=localhost
--username=sysadm --dbname=erp --no-password --role=dbsa --verbose
/tmp/erp.backup
```

или, в случае поставки логической копии в формате директории:

```
root@localhost:~# pg_restore --host=localhost
--username=sysadm --dbname=erp --no-password --role=dbsa --jobs=12
--verbose /tmp/erp_dump_dir
```

В приведенном примере команда `pg_restore` выполняет подключение к серверу СУБД с именем `localhost` (параметр `--host`) от имени пользователя-владельца БД `sysadm` (параметр `--username`) и выполняет восстановление базы данных `erp` (параметр `--dbname`) из дамп-файла `/tmp/erp.backup` или из директории `/tmp/erp_dump_dir`.

#### 4.1.4 Установка сервера отчетов

Для организации централизованного доступа к шаблону отчетов используем сервер NFS.

Сервер службы находится в пакете `nfs-kernel-server`, для установки необходимо выполнить следующую команду:

```
apt-get install nfs-kernel-server
```

Настройте сервер, отредактировав конфигурационный файл `/etc/exports`:

```
/opt/report *(rw, sync, no_subtree_check)
```

Запустите команду `exportfs` с ключом «-a» и перезапустите службы NFS:

```
exportfs -a
run_init /etc/init.d/nfs-kernel-server restart
run_init /etc/init.d/nfs-common restart
```

## 4.2 Установка клиентского программного обеспечения

### 4.2.1 Установка операционной среды

На клиентских рабочих местах ОС «Astra Linux SE» версии 1.5 и выше, а также ОС из семейства Windows.

4.2.1.1 Установка ОС «Astra Linux SE» выполняется в соответствии с документом Руководство администратора, входящим в состав поставки программного продукта.

4.2.1.2 Установка операционной среды Windows выполняется в соответствии с эксплуатационной документацией.

### 4.2.2 Установка офисного пакета OpenOffice

За информацией по установке офисного пакета следует обратиться к Интернет-ресурсу по адресу <https://www.openoffice.org/download/common/instructions.html>.

### 4.2.3 Установка клиентского программного обеспечения

Клиентское программное обеспечение для ОС семейства Windows поставляется в виде архива (zip- или rar-файла). Для установки необходимо извлечь содержимое архива на рабочее место пользователя.

В процессе установки клиентских модулей на рабочее место пользователя в файле `connection.ini`, расположенный в корне папки клиентских модулей, записываются настройки соединения с сервером БД, например:

```
[ERP]
DatabaseType=PostgreSQL
Server=10.184.13.23
Port=5432
Database=erp
Schema=
```

где `Server` – имя или IP-адрес сервера БД, `Port` – номер порта посредством которого устанавливается соединение с СУБД, `Database` – имя базы данных на сервере.

При необходимости создать дополнительные ссылки или ярлыки на «рабочем столе» пользователя.

#### 4.2.4 Подключение сервера отчетов

NFS-клиент находится в пакете `nfs-common`, для установки необходимо выполнить следующую команду:

```
apt-get install nfs-common
```

В файле `/etc/fstab` добавить строку:

```
<адрес_сервера>:/opt/report /opt/report nfs default 0 0
```

## 5 Резервное копирование и восстановление БД

### 5.1 Логическое резервирование и восстановление

#### 5.1.1 Создание логической резервной копии БД

Создание логической копии выполняется с использованием утилиты `pg_dump`, входящей в поставку СУБД.

Пример использования команды `pg_dump` приведен ниже.

```
pg_dump -d testdb -h localhost -p 5432 -U postgres -v -F d -Z  
9 -j 8 -f /dump/testdb 2>/dump/log.txt
```

Данная команда принимает следующий набор параметров:

- d – задает имя базы данных для резервирования;
- h – имя/IP-адрес сервера СУБД;
- p – номер порта сервера СУБД;
- U – имя пользователя для подключения к БД;
- v – вывод подробных сообщений в процессе создания копии;
- F – задает формат выводимых данных резервной копии (возможны следующие значения: d – директория, c – пользовательский, t – формат tar), если параметр не задан, то используется текстовый формат;
- z – задает степень сжатия файлов резервной копии;
- j – задает количество параллельно выполняемых потоков при выгрузке данных;
- f – задает имя выходного файла, директории.

В результате выполнения команды в директории `/dump/testdb` будет создана резервная копия базы данных `testdb`.

Важное преимущество `pg_dump` в сравнении с другим методом резервного копирования, описанным далее, состоит в том, что вывод `pg_dump` обычно можно загрузить в более новые версии PostgreSQL, в то время как резервная копия на уровне файловой системы и непрерывное архивирование жёстко зависят от версии сервера. Также, только метод с применением `pg_dump` можно использовать при переносе базы данных на другую аппаратную архитектуру, например, при переносе с 32-битной на 64-битную версию сервера.

Логическая копия, создаваемая `pg_dump`, является внутренне согласованной, то есть, представляет собой снимок базы данных на момент начала запуска `pg_dump`.

Более подробную информацию по использованию утилиты `pg_dump` можно получить из документа Руководство системного программиста 07623615.00082-01 32 01, входящим в состав поставки СУБД Синергия-БД.

### 5.1.2 Восстановление БД из логической копии

Восстановление базы данных и логической копии выполняется утилитой `pg_restore`, входящей в поставку СУБД.

Для восстановления из логической копии необходимо предварительно создать пустую базу данных в соответствии с инструкциями, приведенными в разделе 4.1.3.2 данного руководства.

Далее необходимо выполнить команду:

```
pg_restore -d testdb -h localhost -p 5432 -U sysadm --  
role=dbsa -j 8 -v /dump/testdb 2>/dump/log.txt
```

Данная команда принимает следующий набор параметров:

- d – задает имя базы данных для резервирования;
- h – имя/IP-адрес сервера СУБД;
- p – номер порта сервера СУБД;
- U – имя пользователя для подключения к БД;

-v – вывод подробных сообщений в процессе восстановления копии;  
-j – задает количество параллельно выполняемых потоков при восстановлении данных.

В результате выполнения команды из директории /dump/testdb, в которой расположена логическая копия, будет восстановлена база данных testdb.

Более подробную информацию по использованию утилиты pg\_restore можно получить из документа Руководство системного программиста 07623615.00082-01 32 01, входящим в состав поставки СУБД Синергия-БД.

## **5.2 Непрерывное архивирование и физическое резервирование**

### **5.2.1 Настройка непрерывного архивирования журналов повтора**

В процессе работы СУБД ведёт журнал упреждающей записи (WAL), который расположен в подкаталоге pg\_wal директории с данными кластера баз данных. В этот журнал записываются все изменения, вносимые в файлы данных. Прежде всего, журнал существует для безопасного восстановления после краха сервера: если происходит аварийное завершение работы сервера СУБД, целостность БД может быть восстановлена в результате «воспроизведения» записей, зафиксированных после последней контрольной точки.

Для успешного восстановления с применением непрерывного архивирования (также называемого «оперативным резервным копированием»), необходима непрерывная последовательность заархивированных файлов WAL, начинающаяся не позже, чем с момента начала копирования. Так что для начала необходимо настроить и протестировать процедуру архивирования файлов WAL до того, как создать первую базовую копию.

В абстрактном смысле, запущенная СУБД производит неограниченно длинную последовательность записей WAL. СУБД физически делит эту последовательность на файлы-сегменты WAL, которые обычно имеют размер в 16 МБ. Файлы-сегменты получают цифровые имена, которые отражают их

позицию в абстрактной последовательности WAL. Когда архивирование WAL не применяется, система обычно создаёт только несколько файлов-сегментов и затем перезаписывает их, меняя номер в имени более не нужного файла-сегмента на больший. Предполагается, что файлы-сегменты, чье содержимое предшествует последней контрольной точке, уже не представляют интереса и могут быть перезаписаны.

При архивировании данных WAL необходимо считывать содержимое каждого файла-сегмента, как только он заполняется, и сохранять эти данные куда-то, прежде чем файл-сегмент будет переработан и использован повторно. В зависимости от применения и доступного аппаратного обеспечения, возможны разные способы хранения журналов повтора: можно скопировать файлы-сегменты в смонтированный по NFS каталог на другую машину, записать их на ленту или собрать их в пакет и записать на CD, либо какие-то совсем другие варианты. Чтобы у администратора баз данных была гибкость в этом плане, СУБД пытается не делать каких-либо предположений о том, как будет выполняться архивация. Вместо этого, СУБД позволяет администратору указать команду оболочки, которая будет запускаться для копирования завершенного файла-сегмента в нужное место.

Чтобы включить архивирование WAL, установите в параметре конфигурации `wal_level` значение `replica`, в `archive_mode` – значение `on`, и задайте необходимую команду оболочки в параметре `archive_command`. На практике эти параметры всегда задаются в файле `postgresql.conf`.

После изменения параметров необходимо перезапустить сервер СУБД командой:

```
service postgresql-sdb-11 restart
```

В `archive_command` символы `%p` заменяются полным путём к файлу, подлежащему архивации, а `%f` заменяются только именем файла.

Пример команды архивации:

```
archive_command = 'cp %p /mnt/server/archivedir/%f'
```

Данная команда копирует архивируемые сегменты WAL в каталог /mnt/server/archivedir.

После замены параметров %p и %f фактически запускаемая команда будет выглядеть так:

```
cp pg_wal/000000010000000A9000000065  
/mnt/server/archivedir/000000010000000A9000000065
```

Подобная команда будет генерироваться для каждого следующего архивируемого файла.

Важно, чтобы команда архивирования возвращала нулевой код завершения, если и только если она завершилась успешно. Получив нулевой результат, СУБД будет полагать, что файл успешно заархивирован и удалит его или переработает. Однако, ненулевой код состояния скажет СУБД, что файл не заархивирован; попытки заархивировать его будут периодически повторяться, пока это не удастся.

### 5.2.2 Создание физической копии БД

Перед созданием физической копии БД необходимо убедиться, что включен режим непрерывной архивации журналов повтора.

Создание физической копии выполняется утилитой pg\_basebackup, входящей в поставку СУБД.

Утилита pg\_basebackup предназначена для создания резервных копий работающего кластера баз данных СУБД. Процедура создания копии не влияет на работу других клиентов. Полученные копии могут использоваться для обеих стратегий восстановления – на заданный момент в прошлом и в качестве отправной точки для резервного сервера при реализации трансляции файлов или потоковой репликации.

Утилита pg\_basebackup создаёт бинарную копию файлов кластера, контролируя режим создания копии автоматически. Резервные копии всегда создаются для кластера целиком и невозможно создать копию для какой-либо сущности базы отдельно. Для этой цели можно использовать, например, утилиту pg\_dump.



Копия создаётся через обычное подключение к СУБД, и при этом используется протокол репликации. Подключение должно осуществляться от лица суперпользователя или пользователя с правом `REPLICATION`, а в файле `pg_hba.conf` должно быть прописано подключение для репликации. Значение `max_wal_senders` на сервере должно быть достаточно большим, чтобы допускать минимум ещё одно подключение для копирования.

Пример создания физической копии с использованием команды `pg_basebackup` приведен ниже.

```
pg_basebackup -h localhost -U postgres -D /mnt/server/backup  
-F t -P -v -z -Z 9
```

Данная команда принимает следующий набор параметров:

- h – имя/IP-адрес сервера СУБД, копия которого создается;
- U – имя пользователя для подключения к СУБД;
- D – директория, в которую будет записана копия;
- F – определяет формат вывода. формат может принимать следующие значения: `p` (`plain`) – записывает выводимые данные в обычные файлы, сохраняя текущую схему размещения каталогов данных и табличных пространств. Если кластер не имеет дополнительных табличных пространств, то вся база будет помещена в заданный каталог. Иначе основной каталог хранения данных будет помещён в целевой каталог, а все остальные табличные пространства – в те же абсолютные пути, в которых они находятся на сервере. Это формат по умолчанию. `t` (`tar`) – записывает в целевой каталог файлы в формате `tar`. Основной каталог хранения данных будет записан в файл `base.tar`, а табличные пространства – в файлы, именованные в соответствии с их `OID`;
- P – включает отчёт о прогрессе;
- v – вывод подробных сообщений в процессе создания копии;
- z – включает `gzip`-сжатие для выводимого `tar`-файла с уровнем сжатия по умолчанию;
- Z – задает степень сжатия файлов резервной копии.

В результате выполнения команды в директории `/mnt/server/backup` будет создана резервная копия сервера БД.

Более подробную информацию по использованию утилиты `pg_basebackup` можно получить из документа Руководство системного программиста 07623615.00082-01 32 01, входящим в состав поставки СУБД Синергия-БД.

### 5.2.3 Восстановление БД из физической копии

При необходимости восстановления кластера баз данных из резервной копии порядок действий следующий:

1. Остановите сервер баз данных, если он запущен.
2. Если есть доступное дисковое пространство, скопируйте весь текущий каталог кластера баз данных и все табличные пространства во временный каталог на случай, если они вам понадобятся. Если дискового пространства недостаточно, необходимо сохранить как минимум содержимое подкаталога `pg_wal` каталога кластера, так как он может содержать журналы, не попавшие в архив перед остановкой системы.
3. Удалите все существующие файлы и подкаталоги из каталога кластера и из корневых каталогов, используемых табличных пространств.
4. Восстановите файлы базы данных из архивной копии файлов. Важно, чтобы у восстановленных файлов были правильные разрешения и правильный владелец (пользователь, от имени которого запущен сервер БД, а не `root`). Если вы используете табличные пространства, убедитесь также, что символьные ссылки в `pg_tblspc` восстановились корректно.
5. Удалите все файлы из `pg_wal`, они восстановились из резервной копии файлов и поэтому, скорее всего, будут старше текущих. Если вы вовсе не архивировали `pg_wal`, создайте этот каталог с правильными правами доступа, но если это была символьная ссылка, восстановите ее.
6. Если на шаге 2 вы сохранили не заархивированные файлы с сегментами WAL, скопируйте их в `pg_wal`. (Лучше всего именно копировать, а не

перемещать их, чтобы у вас остались неизменённые файлы на случай, если возникнет проблема и всё придётся начинать сначала.)

7. Создайте командный файл восстановления `recovery.conf` в каталоге кластера баз данных. Вы можете также временно изменить `pg_hba.conf`, чтобы пользователи не могли подключаться, пока вы не будете уверены, что восстановление завершилось успешно.

8. Запустите сервер. Сервер запустится в режиме восстановления и начнёт считывать необходимые ему архивные файлы WAL. Если восстановление будет прервано из-за внешней ошибки, сервер можно просто перезапустить и он продолжит восстановление. По завершении процесса восстановления сервер переименует файл `recovery.conf` в `recovery.done` (чтобы предотвратить повторный запуск режима восстановления), а затем перейдёт в штатный режим.

9. Просмотрите содержимое базы данных, чтобы убедиться, что вы вернули ее к желаемому состоянию. Если это не так, вернитесь к шагу 1. Если все в порядке, разрешите пользователям подключаться к серверу, восстановив обычный файл `pg_hba.conf`.

Ключевой момент процедуры восстановления заключается в создании файла конфигурации `recovery.conf`, определяющего, как будет выполняться восстановление и до какой точки. В качестве прототипа можно использовать файл `recovery.conf.sample` (он обычно помещается в каталог `/usr/share/sdb-11/11` после установки).

Минимально, что необходимо указать в `recovery.conf` – это команду `restore_command`, которая говорит СУБД, как получать из архива файл-сегменты WAL. Как и `archive_command`, это командная строка для оболочки. Она может содержать символы `%f`, которые заменятся именем требуемого файла журнала, и `%p`, которые заменятся целевым путем для копирования этого файла.

Пример команды для параметра `restore_command`:

```
restore_command = 'cp /mnt/server/archivedir/%f %p'
```

Более подробную информацию по процедуре восстановления из физической копии можно получить из документа Руководство системного программиста 07623615.00082-01 32 01, входящим в состав поставки СУБД Синергия-БД.

### **Требования к ИТ-инфраструктуре**

Серверное ПО должно функционировать под управлением сертифицированной операционной системы Astra Linux Special Edition версии 1.5. В качестве СУБД применяется сертифицированная СУБД с открытым исходным кодом (Синергия-БД), обеспечивающая механизмы дискреционного контроля доступа. Допускается функционирование серверного ПО в виртуальной машине (выполнено тестирование в среде виртуализации VMware vSphere 5.5).

### **Требования к серверной платформе**

<b>Параметр</b>	<b>Значение</b>		
Количество сотрудников предприятия, обрабатываемых в системе	до 2000	2000-7000	7000-12000
Количество пользователей	до 200	200-400	400-600
Количество вычислительных ядер	40	80	120
Тактовая частота процессора, ГГц	2,5	2,5	2,5
Объем оперативной памяти, ГБ	256	512	1024
Объем дисковой подсистемы, ГБ	500	2000	4000
Производительность дисковой подсистемы, операций ввода/вывода в секунду (IOPS)	3000	5000	10000
Пропускная способность сетевых интерфейсов, Гбит/с	1	1	10

### Требования к клиентским рабочим местам

Параметр	Значение
Количество вычислительных ядер	2
Тактовая частота процессора, ГГц	1,8
Объем оперативной памяти, ГБ	2
Объем дисковой подсистемы	100 ГБ
Пропускная способность сетевых интерфейсов	100 Мбит/с
Операционная система	Синергия-ОС/Astra Linux SE 1.5/ Windows 7
Дополнительное программное обеспечение	WINE версии 2.19 (в среде Linux), Microsoft Office (версия 14 и выше)/Open Office (версия 4.0.1 и выше) / Libre Office (версия 5.4.6 и выше)

Указанные требования могут варьироваться в зависимости от технологии обработки информации в автоматизированной системе и должны быть уточнены на этапе внедрения.