Аэронавигатор

# RMS

## Инструкция по резервному копированию сервера

## Руководство системного администратора

Санкт-Петербург

2020

# СОДЕРЖАНИЕ

1 СОЗДАНИЕ БАЗОВОЙ РЕЗЕРВНОЙ КОПИИ ..... 3
1.1 Программа pg_basebackup ..... 4
1.2 Синтаксис ..... 4
1.3 Описание ..... 4
1.4 Параметры ..... 5
2 ВОССТАНОВЛЕНИЕ РЕЗЕРВНОЙ КОПИИ ..... 15
3 РЕЗЕРВНАЯ КОПИЯ ПОРТАЛА ..... 18

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

# 1 Создание базовой резервной копии

Проще всего получить базовую резервную копию, используя программу pg_basebackup. Эта программа сохраняет базовую копию в виде обычных файлов или в архиве tar. Если гибкости pg_basebackup не хватает, вы также можете получить базовую резервную копию, используя низкоуровневый API.

Продолжительность создания резервной копии обычно не имеет большого значения. Однако если вы эксплуатируете сервер с отключённым режимом `full_page_writes`, вы можете заметить падение производительности в процессе резервного копирования, так как режим `full_page_writes` включается принудительно на время резервного копирования.

Чтобы резервной копией можно было пользоваться, нужно сохранить все файлы сегментов WAL, сгенерированные во время и после копирования файлов. Для облегчения этой задачи, процесс создания базовой резервной копии записывает *файл истории резервного копирования*, который немедленно сохраняется в области архивации WAL. Данный файл получает имя по имени файла первого сегмента WAL, который потребуется для восстановления скопированных файлов. Например, если начальный файл WAL назывался `000000010000123400000055CD`, файл истории резервного копирования получит имя `000000010000123400000055CD.007C9330.backup`. (Вторая часть имени файла обозначает точную позицию внутри файла WAL и обычно может быть проигнорирована.) Как только вы заархивировали копии файлов данных и файлов сегментов WAL, полученных в процессе копирования (по сведениям в файле истории резервного копирования), все заархивированные сегменты WAL с именами, меньшими по номеру, становятся ненужными для восстановления файловой копии и могут быть удалены. Но всё же рассмотрите возможность хранения нескольких наборов резервных копий, чтобы быть абсолютно уверенными, что вы сможете восстановить ваши данные.

Файл истории резервного копирования — это просто небольшой текстовый файл. В него записывается метка, которая была передана pg_basebackup, а также время и текущие сегменты WAL в момент начала и завершения резервной копии. Если вы связали с данной меткой соответствующий файл дампа, то заархивированного файла истории достаточно, чтобы найти файл дампа, нужный для восстановления.

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

Поскольку необходимо хранить все заархивированные файлы WAL с момента последней базовой резервной копии, интервал базового резервного копирования обычно выбирается в зависимости от того, сколько места может быть выделено для архива файлов WAL. Также стоит отталкиваться от того, сколько вы готовы ожидать восстановления, если оно понадобится — системе придётся воспроизвести все эти сегменты WAL, а этот процесс может быть долгим, если с момента последней базовой копии прошло много времени.

## 1.1 Программа pg_basebackup

pg_basebackup — создать резервную копию кластера PostgreSQL

## 1.2 Синтаксис

`pg_basebackup` [***параметр***...]

## 1.3 Описание

pg_basebackup предназначен для создания резервных копий работающего кластера баз данных PostgreSQL. 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 ведомого сервера при реализации трансляции файлов или потоковой репликации.

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 PostgreSQL, и при этом используется протокол репликации. Подключение должно осуществляться от лица суперпользователя или пользователя с правом `REPLICATION` , а в `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.

Можно запустить одновременно несколько команд `pg_basebackup`, но с точки зрения производительности лучше делать всего одну копию одновременно, а затем копировать получаемый результат.

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

С помощью pg_basebackup можно получить базовую копию не только на ведущем, но и на ведомом сервере. Для этого на ведомом сервере необходимо разрешить соединения репликации (параметры `max_wal_senders` и hot_standby, а также настроить аутентификацию компьютера). При этом на ведущем необходимо включить full_page_writes.

Заметьте, что при копировании с ведомого сервера есть некоторые ограничения:

- Файл истории резервного копирования в целевом кластере баз данных не создаётся.
- Нет гарантии, что все необходимые файлы журналов WAL будут доступны на момент завершения создания резервной копии. Если планируется использовать резервную копию для восстановления из журнала, то для обеспечения целостности необходимо самостоятельно добавлять их с помощью параметра `-x`.
- Если ведомый сервер переключается в роль ведущего в процессе копирования, копирование прерывается.
- Все необходимые для резервной копии WAL-записи должны содержать полные страницы, для чего нужно включить режим `full_page_writes` на ведущем и не использовать в `archive_command` такие утилиты, как pg_compresslog, которые могут удалить записанные полные страницы из WAL.

## 1.4 Параметры

Описанные далее аргументы командной строки влияют на размещение и формат вывода.

`-D` ***каталог***
`--pgdata=`***каталог***

Целевой каталог для записи данных. pg_basebackup создаст его и родительские, если необходимо. Каталог может быть создан заранее, но должен быть пустым, иначе возникнет ошибка.

Если резервирование работает в режиме `tar`, а имя каталога имеет значение `-` (тире), то tar-файл будет писаться в `stdout`.

Этот флаг является обязательным.

`-F` ***формат***

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

`--format=`***формат***

Устанавливает формат вывода. ***формат*** может принимать следующие значения:

`p`
`plain`

Записывает выводимые данные в обычные файлы, сохраняя структуру текущих каталогов данных и табличных пространств. Если в кластере нет дополнительных табличных пространств, 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 располагаются на исходном сервере. (Чтобы изменить эти пути, воспользуйтесь параметром `--tablespace-mapping`).

Это формат по умолчанию.

`t`
`tar`

Записывает в целевой каталог файлы в формате tar. Основной каталог хранения данных будет писаться в файл `base.tar`, а табличные пространства — в файлы, именованные в соответствии с их OID.

Если имя целевого каталога задано как `-` (тире), то данные будут писаться в стандартный вывод, что позволяет, например, использовать gzip. Это возможно лишь в случае без использования дополнительных табличных пространств.

`-r` ***скорость_передачи***
`--max-rate=`***скорость_передачи***

Максимальная скорость передачи данных с сервера. Значение задаётся в Кб/с. Для установки значения в мегабайтах, можно использовать суффикс `M`. Также допустим суффикс `k`, но он не принципиален. Допустимые значения лежат в рамках между 32 Кб/с и 1024 Мб/с.

Служит для снижения влияния на производительность сервера со стороны работающего pg_basebackup.

Этот параметр всегда оказывает влияние на передачу каталога данных, а

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

на передачу файлов WAL он влияет, только если выбран метод передачи `fetch`.

```
-R
--write-recovery-conf
```

Записать минимальный файл `recovery.conf` в каталог вывода (или базовый архивный файл при использовании формата tar) для упрощения настройки ведомого сервера. В файл `recovery.conf` будут записаны параметры соединения и, если указан, слот репликации, который использует pg_basebackup, так что впоследствии при потоковой репликации будут использоваться те же параметры.

```
-S имя_слота
--slot=имя_слота
```

Этот параметр может применяться только вместе с `-X stream`. Он устанавливает использование заданного слота репликации при потоковой передаче WAL. Если базовая копия предназначена для использования на ведомом сервере с потоковой репликацией, затем должен использоваться слот с тем же именем в `recovery.conf`. Тем самым гарантируется, что сервер не удалит никакие необходимые данные WAL после того, как базовая копия будет получена, и до того, как начнётся потоковая репликация.

```
-T старый_каталог=новый_каталог
--tablespace-mapping=старый_каталог=новый_каталог
```

Переместить табличное пространство
из *старого_каталога* в *новый_каталог* в процессе копирования. Чтобы перемещение произошло, в параметре *старый_каталог* должен задаваться в точности путь табличного пространства, как он определён. (Но не будет ошибкой, если табличного пространства, на которое указывает *старый_каталог*, в архиве не окажется.) И *старый_каталог*, и *новый_каталог* должны задаваться абсолютными путями. Если в пути встречается символ `=`, его необходимо экранировать обратной косой чертой. Этот параметр можно добавить несколько раз для нескольких табличных пространств. См. примеры ниже.

Если табличное пространство перемещается таким способом, символические ссылки внутри основного каталога хранения данных также приводятся в соответствие с новым местоположением. Таким образом, для экземпля-

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

ра сервера подготавливается новый каталог данных, в котором все табличные пространства оказываются в новом расположении.

В настоящее время этот параметр работает только с обычным форматом вывода; если выбран формат tar, параметр игнорируется.

`--xlogdir=`***`каталог_xlog`***

Указывает размещение каталога хранения журналов транзакций. Задаваемый в параметре ***`каталог_xlog`*** путь должен быть абсолютным. Каталог с журналом транзакций можно задать только при создании копии в простом режиме.

`-x`
`--xlog`

Использование параметра эквивалентно `-X` в режиме `fetch`.

`-X` ***`метод`***
`--xlog-method=`***`метод`***

Включает все необходимые файлы журналов транзакций (файлы WAL) в резервную копию. В том числе включаются все журналы транзакций, сгенерированные в процессе создания резервной копии. Если параметр указан, то главный процесс БД может быть запущен непосредственно с восстановленным каталогом, без дополнительного архива журналов; таким образом полученная резервная копия будет вполне самодостаточной.

Для сбора журналов транзакций поддерживаются следующие методы:

`f`
`fetch`

Файлы журналов транзакций собираются в конце процесса копирования. Таким образом необходимо установить достаточно большое значение параметра wal_keep_segments, чтобы избежать преждевременного удаления файлов журнала. В случае удаления файлов до завершения процесса копирования возникнет ошибка, а копия будет непригодной к использованию.

`s`
`stream`

Передавать журнал транзакций в процессе создания резервной копии. При

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

этом открывается второе соединение к серверу, по которому будет передаваться журнал транзакций, одновременно с созданием резервной копии. Таким образом будут использоваться два подключения из разрешённых параметром max_wal_senders. И если клиент будет успевать получать журнал транзакций, ведущему серверу не потребуется хранить дополнительные журналы транзакций.

```
-z
--gzip
```

Включает gzip-сжатие для выводимого tar-файла с уровнем компрессии по умолчанию. Сжатие поддерживается лишь в режиме упаковки.

```
-Z уровень
--compress=уровень
```

Включает gzip-сжатие для выводимого tar-файла и задаёт уровень сжатия от 0 (без сжатия) до 9 (максимальное сжатие). Сжатие поддерживается только с форматом tar.

Описанные далее аргументы командной строки влияют на генерацию резервной копии и ход выполнения приложения.

```
-c fast|spread
--checkpoint=fast|spread
```

Устанавливает режим контрольных точек: fast (быстрый) или spread (протяжённый, по умолчанию). Подробнее см. Подраздел 25.3.3.

```
-l метка
--label=метка
```

Устанавливает метку для созданной резервной копии. Если не указана, то по умолчанию будет использовано значение «`pg_basebackup base backup`».

```
-P
--progress
```

Включает отчёт о прогрессе. Если этот режим включён, то во время создания копии будет передаваться примерный процент выполнения. Так как данные в базе могут меняться во время копирования, это значение будет лишь приближённым и может достигать не точно `100%`. В частности, когда в копию включается журнал WAL, конечный размер невозможно предсказать

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

заранее, и в этом случае ожидаемый конечный размер будет увеличиваться, превысив ориентировочный полный размер без WAL.

Если режим включён, то процесс копирования начнется с перечисления размеров всей базы, а затем продолжится отправкой непосредственно данных. Это может немного увеличить время операции, в частности, пройдёт больше времени до начала передачи данных.

```
-v
--verbose
```

Включает режим подробного вывода. Будет выводится некоторая дополнительная информация при начале и завершении, а также имена обрабатываемых файлов, если включён отчёт о прогрессе.

Далее описаны параметры управления подключением.

```
-d строка_подключения
--dbname=строка_подключения
```

Указывает параметры подключения к серверу в формате строки подключения; они будут переопределять любые одноимённые параметры, заданные в командной строке.

Параметр называется `--dbname` для согласованности с другими клиентскими приложениями, но так как pg_basebackup не подключается к какой-либо конкретной базе, это имя в строке подключения игнорируется.

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

Указывает имя компьютера, на котором работает сервер. Если значение начинается с косой черты, оно определяет каталог Unix-сокета. Значение по умолчанию берётся из переменной окружения `PGHOST`, если она установлена. В противном случае выполняется подключение к Unix-сокету.

`-- p` ***порт***
`--port=`***порт***

Указывает TCP-порт или расширение файла локального Unix-сокета, через который сервер принимает подключения. Значение по умолчанию определяется переменной окружения `PGPORT`, если она установлена, либо числом, заданным при компиляции

`-s` ***interval***
`--status-interval=`***interval***

Указывает интервал в секундах между отправкой пакетов статуса, отправляемых на сервер. Это позволяет упростить мониторинг прогресса. Чтобы выключить периодическое обновление статуса, необходимо установить значение в ноль. При этом обновление будет отправляться по запросу сервера для избежания отсоединения по истечению времени. Значение по умолчанию составляет 10 секунд.

`-U` ***имя_пользователя***
`--username=`***имя_пользователя***

Имя пользователя, под которым производится подключение.

`-w`
`--no-password`

Не выдавать запрос на ввод пароля. Если сервер требует аутентификацию по паролю и пароль не доступен с помощью других средств, таких как файл `.pgpass`, попытка соединения не удастся. Этот параметр может быть полезен в пакетных заданиях и скриптах, где нет пользователя, который

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

вводит пароль.

`-W`
`--password`

Принудительно запрашивать пароль перед подключением к базе данных.

Это несущественный параметр, так как pg_basebackup запрашивает пароль автоматически, если сервер проверяет подлинность по паролю. Однако чтобы понять это, pg_basebackup лишний раз подключается к серверу. Поэтому иногда имеет смысл ввести `-W`, чтобы исключить эту ненужную попытку подключения.

Другие флаги:

`-V`
`--version`

Вывести версию pg_basebackup.

`-?`
`--help`

Вывести справку по команде pg_basebackup.

## Переменные окружения

Как и большинство других утилит PostgreSQL, приложение также использует переменные окружения, поддерживаемые libpq .

## Замечания

Прежде чем начнётся копирование, на сервере с копируемой базой необходимо выполнить контрольную точку. И если копирование запускается без ключа `--checkpoint=fast`, это может занять некоторое время, в течение которого pg_basebackup не будет проявлять никакой активности.

Резервная копия будет включать в себя все файлы каталога хранения данных и табличных пространств, а также конфигурационные файлы и прочие файлы, размещённые в каталоге данных. Однако копируются лишь

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

простые файлы и каталоги. Символьные ссылки (кроме тех, что указывают на табличные пространства) и файлы спецустройств игнорируются.

Если не указан параметр `--tablespace-mapping`, табличные пространства в простом формате будут копироваться в тот же путь, который они имеют на сервере. Поэтому при наличии табличных пространств создать базовую копию в простом формате на том же сервере не удастся, так как копия будет направлена в те же каталоги, где располагаются исходные табличные пространства.

Когда применяется режим формата tar, пользователь должен позаботиться о том, чтобы все архивы tar были распакованы до запуска сервера PostgreSQL. Если имеются дополнительные табличные пространства, архивы tar для них должны быть распакованы в правильные каталоги. В таком случае для этих табличных пространств сервером будут созданы символические ссылки, согласно содержимому файла `tablespace_map`, включённого в архив `base.tar`.

pg_basebackup совместим с серверами той же или более младших версий, но не ниже 9.1. Однако режим трансляции WAL (`-X stream`) поддерживается с версиями сервера не ниже 9.3, а режим формата tar (`--format=tar`) текущей версии совместим только с версиями сервера не ниже 9.5.

## Примеры

Создание резервной копии сервера `mydbserver` и сохранение её в локальном каталоге `/usr/local/pgsql/data`:

```
$ pg_basebackup -h mydbserver -D /usr/local/pgsql/data
```

Создание резервной копии локального сервера в отдельных сжатых файлах tar для каждого табличного пространства и сохранение их в каталоге `backup` с индикатором прогресса в процессе выполнения:

```
$ pg_basebackup -D backup -Ft -z -P
```

Чтобы создать резервную копию локальной базы с единым табличным пространством и сжатием через bzip2:

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

```
$ pg_basebackup -D - -Ft | bzip2 > backup.tar.bz2
```

Команда завершится ошибкой, если в базе будет более одного табличного пространства.

Создание резервной копии локальной базы данных с перемещением табличного пространства `/opt/ts` в `./backup/ts`:

```
$ pg_basebackup -D backup/data -T /opt/ts=$(pwd)/backup/ts
```

На случай возможного краха сервера рекомендуется организовать регулярное копирование файлов, содержащих резервные копии, на внешние носители (сетевые устройства, ленты и т.п.).

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

# 2 Восстановление резервной копии

Если случилось худшее, и вам необходимо восстановить базу данных из резервной копии. Порядок действий таков:

1. Остановите сервер баз данных, если он запущен.
2. Если у вас есть место для этого, скопируйте весь текущий каталог кластера баз данных и все табличные пространства во временный каталог на случай, если они вам понадобятся. Учтите, что эта мера предосторожности требует, чтобы свободного места на диске было достаточно для размещения двух копий существующих данных. Если места недостаточно, необходимо сохранить как минимум содержимое подкаталога `pg_xlog` 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 резервной копии файлов. Важно, чтобы у восстановленных файлов были правильные разрешения и правильный владелец (пользователь, запускающий сервер, а не `root`!). Если вы используете табличные пространства, убедитесь также, что символьные ссылки в `pg_tblspc/` восстановились корректно.
5. Удалите все файлы из `pg_xlog/`; они восстановились из резервной копии файлов и поэтому, скорее всего, будут старее текущих. Если вы вовсе не архивировали `pg_xlog/`, создайте этот каталог с правильными правами доступа, но если это была символьная ссылка, восстановите её.
6. Если на шаге 2 вы сохранили незаархивированные файлы с сегментами WAL, скопируйте их в `pg_xlog/`. (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 обычные 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

© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.
Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону

предотвратить повторный запуск режима восстановления), а затем перейдёт к обычной работе с базой данных.

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

Ключевой момент этой процедуры заключается в создании файла конфигурации восстановления, описывающего, как будет выполняться восстановление и до какой точки. В качестве прототипа вы можете использовать файл `recovery.conf.sample` (он обычно помещается в каталог `share/` после установки). Единственное, что совершенно необходимо указать в `recovery.conf` — это команду `restore_command`, которая говорит PostgreSQL, как получать из архива файл-сегменты WAL. Как и `archive_command`, это командная строка для оболочки. Она может содержать символы `%f`, которые заменятся именем требующегося файла журнала, и `%p`, которые заменятся целевым путём для копирования этого файла. (Путь задаётся относительно текущего рабочего каталога, т. е. каталога кластера данных.) Если вам нужно включить в команду сам символ `%`, напишите `%%`. Простейшая команда, которая может быть полезна, такая:

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

Эта команда копирует заархивированные ранее сегменты WAL из каталога `/mnt/server/archivedir`. Разумеется, вы можете использовать что-то более сложное, возможно, даже скрипт оболочки, который укажет оператору установить соответствующую ленту.

Важно, чтобы данная команда возвращала ненулевой код возврата в случае ошибки. Эта команда ***будет*** вызываться и с запросом файлов, отсутствующих в архиве; в этом случае она должна вернуть ненулевое значение и это считается штатной ситуацией. В исключительной ситуации, когда команда была прервана сигналом (кроме SIGTERM, который применяется в процессе остановки сервера базы данных) или произошла ошибка оболочки (например, команда не найдена), восстановление будет прервано и сервер не запустится.

Не все запрашиваемые файлы будут сегментами WAL; следует также ожидать запросов файлов с суффиксом `.history`. Также учтите, что базовое имя пути `%p` будет отличаться от `%f`; не думайте, что они взаимозаменяемы.

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

Сегменты WAL, которые не найдутся в архиве, система будет искать в `pg_xlog/`; благодаря этому можно использовать последние незаархивированные сегменты. Однако файлы в `pg_xlog/` будут менее предпочтительными, если такие сегменты окажутся в архиве.

Обычно при восстановлении обрабатываются все доступные сегменты WAL и, таким образом, база данных восстанавливается до последнего момента времени (или максимально близкого к нему, в зависимости от наличия сегментов WAL). Таким образом, восстановление обычно завершается с сообщением «файл не найден»; точный текст сообщения об ошибке зависит от того, что делает `restore_command`. Вы также можете увидеть сообщение об ошибке в начале восстановления для файла с именем типа `00000001.history`. Это также нормально и обычно не говорит о какой-либо проблеме при восстановлении в простых ситуациях.

Если вы хотите восстановить базу на какой-то момент времени (скажем, до момента, когда неопытный администратор базы данных удалил основную таблицу транзакций), просто укажите требуемую точку остановки в `recovery.conf`. Вы можете задать эту точку, иначе называемую «целью восстановления», по дате/времени, именованной точке восстановления или определённому идентификатору транзакции. На момент написания этой документации полезными могут быть только указания даты/времени или имени точки восстановления, пока нет никаких средств, позволяющих точно определить, какой идентификатор транзакции нужно выбрать.

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*

# 3 Резервная копия Портала

Необходимо организовать регулярное резервное копирование папки *C:\Program Files\Apache Software Foundation\Tomcat 7.0* на внешние носители.

*© 2020 Copyright ООО «Аэронавигатор» - ВСЕ ПРАВА ЗАЩИЩЕНЫ.*
*Данная информация является собственностью ООО «Аэронавигатор», любое ее использование преследуется по закону*