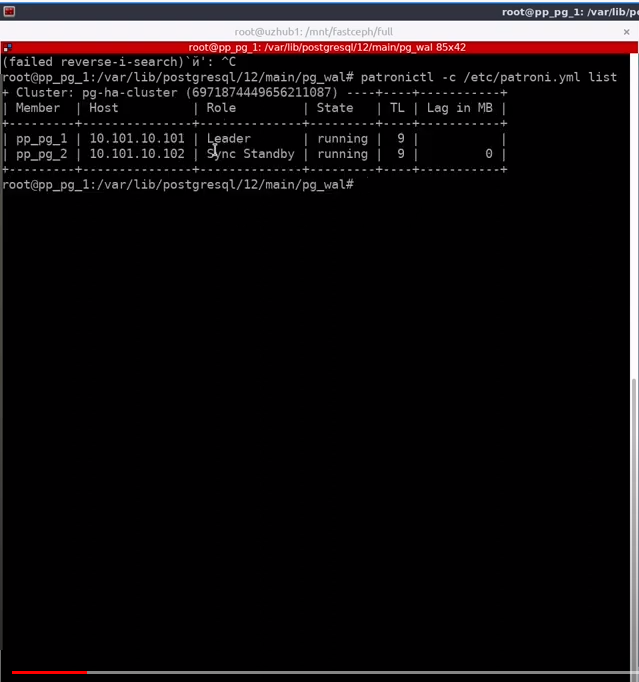
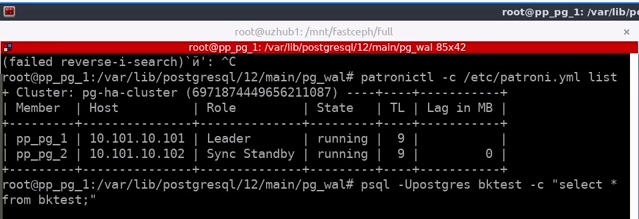
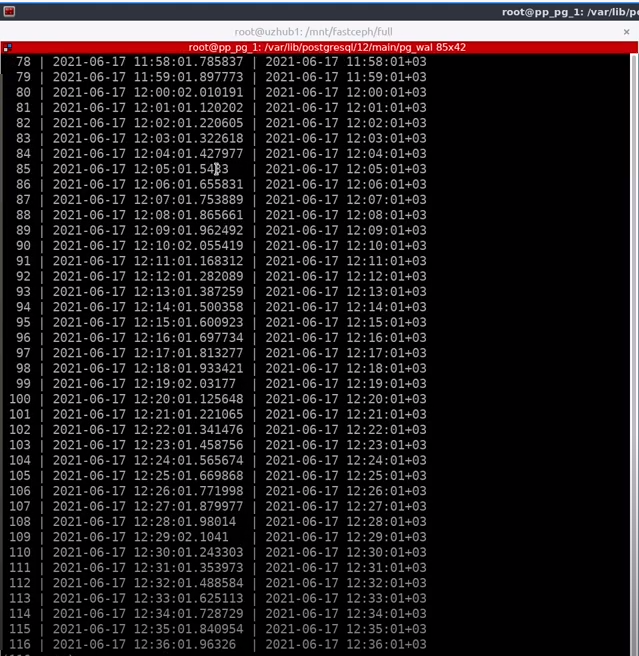
Стенд состоит из:



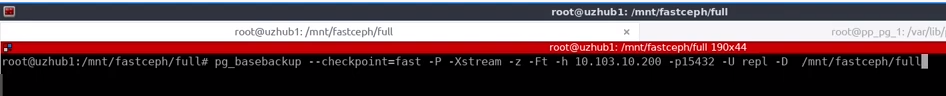
Есть в cron скрипт который пишет в таблицу данные каждую минуту.



Вот что пишет.



Для проверки механизма восстановления делаем backup с помощью pg\_basebackup



pg\_basebackup --checkpoint=fast -P Xstream -z -Ft -h 10.103.10.200 -p5432 -U repl -D /mnt/fastceph/full

где: --checkpoint=fast - мы не дожидаемся пока у нас заполнится жкрнал

-P - будет отображать прогресс в действиях

-Xstream - используем stream, а не fetch

-z - используем компрессию

-Ft - формат tar

-h 10.103.10.200 -

-p15432 -

-U repl - пользователь repl. Данный пользователь должен обладать правами репликации

-D /mnt/fastceph/full - место куда будет записан backup

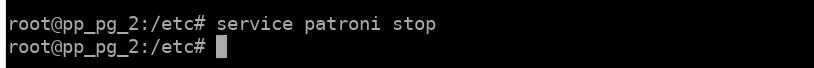
-R - Создать файл [standby.signal](https://postgrespro.ru/docs/enterprise/13/warm-standby" \l "FILE-STANDBY-SIGNAL) и добавить параметры конфигурации в файл postgresql.auto.conf в целевом каталоге (или внутри архива, если используется формат tar). Это упрощает настройку ведомого сервера при восстановлении этой копии. В файл postgresql.auto.conf будут записаны параметры соединения и слот репликации, если его использует pg\_basebackup, так что впоследствии при потоковой репликации будут использоваться те же параметры.

Backup создан. Получили 2 архива.



Для восстановления мы должны остановить кластер Patroni на всех nodes





Дополнительно проверяем статус сервиса patroni

На обоих node:

1. переименовываем папку с данными /var/lib/postgresql/12/main на /var/lib/postgresql/12/main\_old

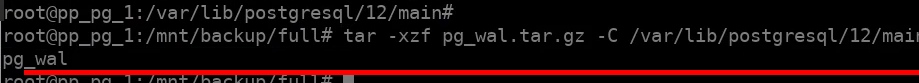
mv /var/lib/postgresql/12/main /var/lib/postgresql/12/main\_old

1. Создаем папку /var/lib/postgresql/12/main: mkdir -p /var/lib/postgresql/12/main
2. Делаем владельцем данной папки postgres: chown -R postgres:postgres /var/lib/postgresql/12/main
3. Предоставляем права на папку 700 : chmod -R 700 /var/lib/postgresql/12/main

На Leader восстанавливаем данные из файлов архива созданного backup:

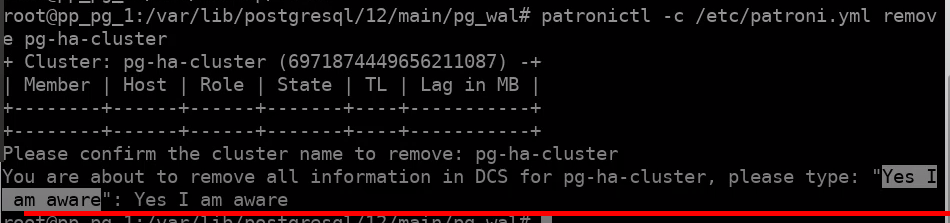


Разархивируем архив с wal в соответствующую директорию.



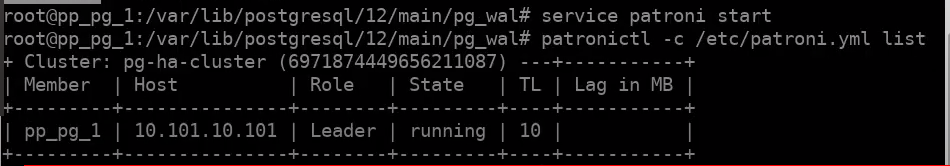
Проверяем содержание и что владельцем все разархивированного является postgres и права там 700. Если это не так делаем п 3 и 4 (см выше)

Удаляем данные кластера patroni из etcd на leader.



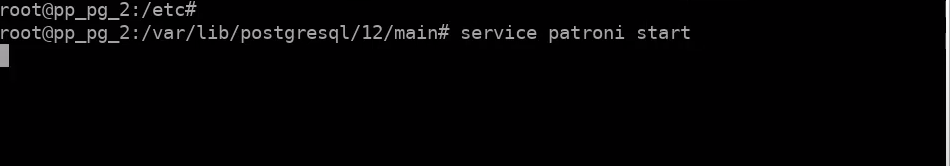
При удалении Patroni запросит ввод полной фразы.

Для того чтобы кластер patroni его нам нужно запустить.



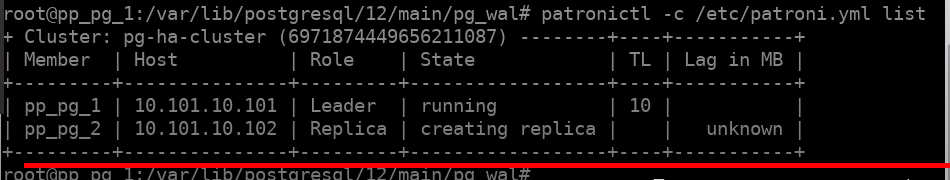
Проверяем что восстановили в БД.

Далее запускаем patroni на второй node

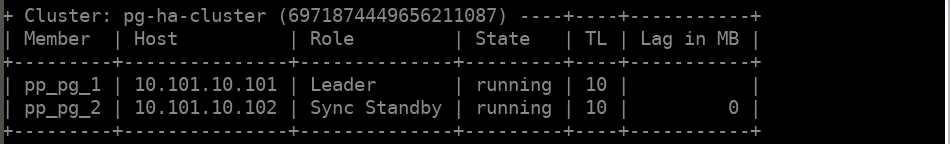


И отслеживаем ее состояние



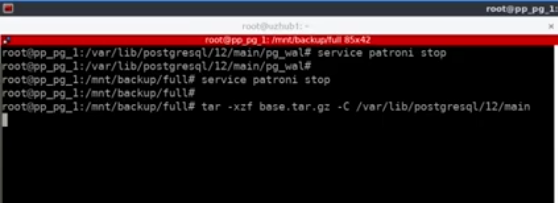




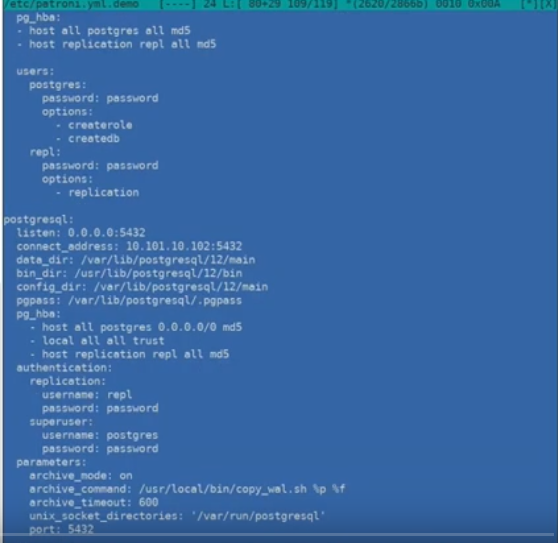


Восстановление по PIRT

1. Останавливаем сервис patroni на всех нодах
2. Очищаем каталоги с данными
3. Распаковываем архив созданный pg\_basebackup, но не распаковываем архив pg\_wal

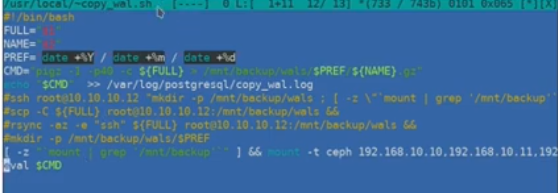


1. Что patroni умел сохранять wal файлы в архиве необходимо добавить следующие строки в конфигурационный файл

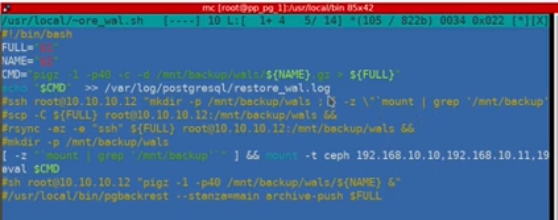


Нам нужны: archive\_mode, archive\_command, archive\_timeout

Скрипт copy\_wal

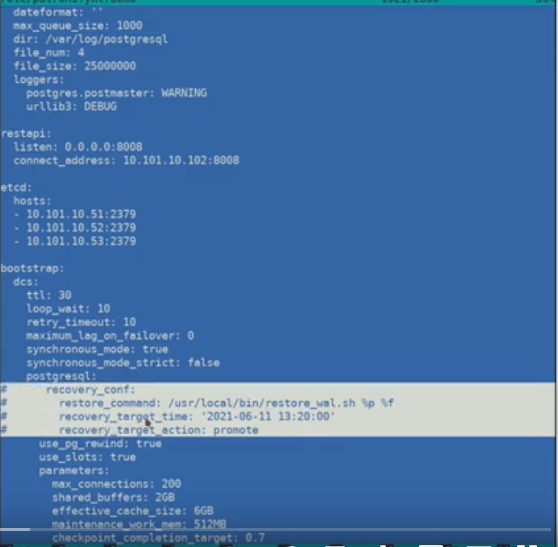


Скрипт восстановления.



Тонкий момент.

В patroni можно прописать выделенные опции, но опция recovery\_target\_time не работает. Почему пока не понятно.

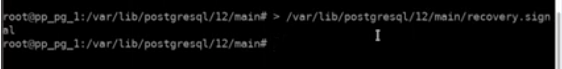


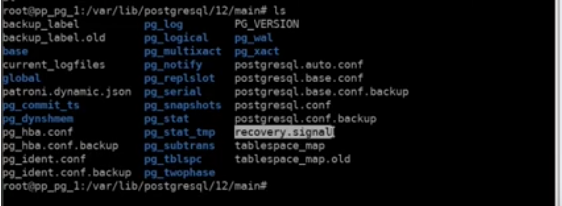
Переходим к восстановлению.

1. восстанавливаем полный бэкап

До 11 версии включительно за восстановление отвечал файл recovery.conf и его наличие приводило кластер в режим восстановления.

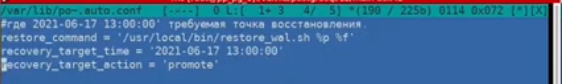
В 12 и выше версиях для восстановления используется создание файла recovery.signal в каталоге данных



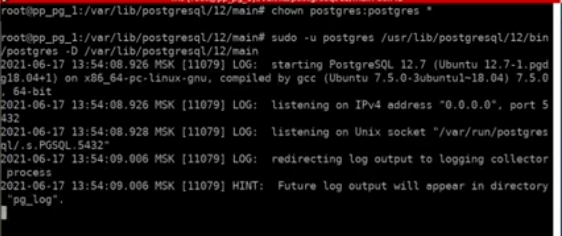


теперь надо прописать параметры восстановления в файд postgresql.auto.conf

В данный файл добавляем следующие строки ручками

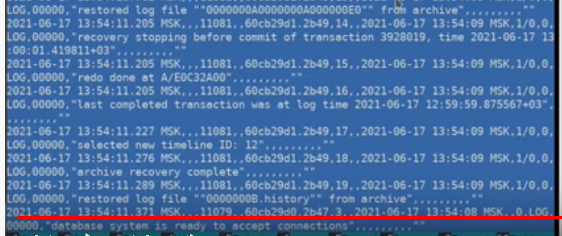


ВНИМАНИЕ: Стартовать нужно не patroni, а postgresql - в данном случае



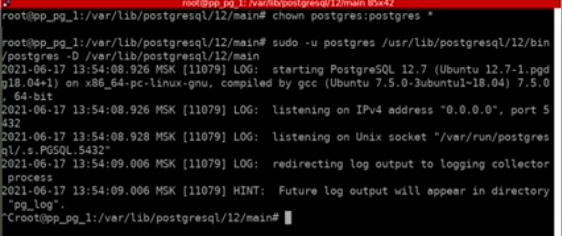
После смотрим log

Последней записью должно быть database system is ready to accept connections

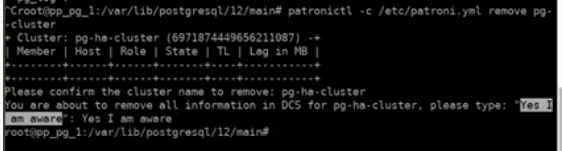


Это решено с postgresql, но не rс патрони.

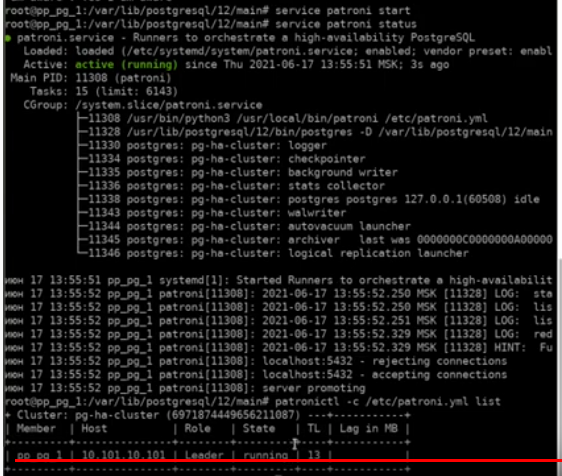
После того как postgresql восстановлен, его останавливаем.



далее удаляем кластер патрони.



запускаем патрони.



Проверяем.