

# Обновление ПГ

## Инструкция по обновлению PostgreSQL с 11 до 15 в кластере Patroni

### Введение

Эта инструкция описывает безопасный процесс обновления PostgreSQL с версии 11 до 15 в высокодоступном кластере, управляемом Patroni.

#### Внимание:

- Все операции следует выполнять последовательно.
- Перед началом убедитесь, что есть резервная копия данных.
- Рекомендуется провести полную проверку на staging-окружении.

### Часть 1: Подготовка к обновлению

#### Шаг 1: Настройка конфигурации для старой версии (11)

Убедитесь, что в Ansible-ветке заданы корректные параметры для текущей версии:

```
---  
# move to postgresql_ha later  
mountpoint: /data/postgres  
mount_opts: "noatime"  
postgresql_data_dir: "/data/postgres/{{ postgresql_version }}/data"  
patroni_pg_version: "{{ postgresql_version }}"  
patroni_pg_data: "{{ postgresql_data_dir }}"  
# end of temporary postgresql_ha block  
  
# leave here at cluster level  
postgresql_version: "11"  
patroni_version: 3.3.0  
patroni_scope: "pg{{ postgresql_version }}"  
patroni_namespace: patroni
```

#### Шаг 2: Очистка /etc/fstab

Удалите старые записи, связанные с PostgreSQL:

```
sed -i '/postgres/d' /etc/fstab
```

## Шаг 3: Установка пакетов PostgreSQL 15 и зависимостей

Выполните установку новой версии PostgreSQL **до остановки кластера**, чтобы избежать простоев:

```
ansible-playbook playbook_site.yaml --diff -l pgsql_ha_dwh --tag  
postgresql_install
```

Это установит `postgresql15`, `postgresql15-server`, `pg_repack_15` и другие необходимые пакеты.

## Шаг 4: Проверка конфигурации через Ansible (в режиме `--check`)

Убедитесь, что изменения в LVM и каталогах данных не вызовут конфликтов:

```
ansible-playbook playbook_site.yaml --diff -l pgsql_ha_dwh --tag  
lvm,patroni,postgresql_manage --check
```

---

## Часть 2: Остановка кластера

### Шаг 5: Пауза кластера Patroni

Переведите кластер в режим обслуживания:

```
/opt/patroni/venv/bin/patronictl -c /opt/patroni/patroni.yaml pause --wait
```

### Шаг 6: Остановка Patroni на всех узлах

Останавливайте в следующем порядке:

1. Все реплики
2. Мастер

```
systemctl stop patroni
```

## Шаг 7: Принудительная остановка PostgreSQL (если процессы остались)

На каждом узле убедитесь, что PostgreSQL полностью остановлен:

```
ps aux | grep postgres
# Если процессы есть – завершите их вручную через pg_ctl
/usr/pgsql-11/bin/pg_ctl -D /data/postgres/11/data/ stop -m fast
```

## Часть 3: Подготовка файловой системы и данных

### Шаг 8: Перемонтирование каталога данных (если требуется)

Если данные хранятся в подкаталоге версии, убедитесь, что структура корректна:

```
umount /data/postgres/11 && \
mount -o noatime /dev/mapper/data-postgres /data/postgres/ && \
install -v -d -o postgres -g postgres -m 0755 /data/postgres/11 && \
mv -v /data/postgres/data/ /data/postgres/11/
```

Этот шаг может быть пропущен, если структура каталогов уже соответствует ожидаемой.

### Шаг 9: Фиксация локалей

Проверьте текущие настройки локалей — они должны совпадать с теми, что будут использованы при инициализации нового кластера:

```
SELECT name, setting FROM pg_settings WHERE name LIKE 'lc_%';
```

**Ожидаемые значения:**

- lc\_collate и lc\_ctype → ru\_RU.UTF-8
- остальные → C

Запишите эти значения — они понадобятся на следующем шаге.

## Часть 4: Инициализация нового кластера и обновление

### Шаг 10: Инициализация кластера PostgreSQL 15

Используйте те же локали, что и в старом кластере:

```
/usr/pgsql-15/bin/initdb -D /data/postgres/15/data/ \
--lc-collate 'en_US.UTF-8' \
--lc-ctype 'en_US.UTF-8' \
--data-checksums
```

### Шаг 11: Сверка чекпоинтов (мастер и реплики)

На всех узлах выполните:

```
/usr/pgsql-11/bin/pg_controldata -D /data/postgres/11/data/ | grep "Latest
checkpoint"
```

Убедитесь, что позиции совпадают — это гарантирует согласованность данных при копировании.

### Шаг 12: Проверка возможности обновления

Запустите pg\_upgrade в режиме проверки:

```
/usr/pgsql-15/bin/pg_upgrade \
-b /usr/pgsql-11/bin \
-B /usr/pgsql-15/bin \
-d /data/postgres/11/data \
-D /data/postgres/15/data \
--link --jobs 10 --check
```

Если проверка завершилась успешно — переходите к обновлению.

### Шаг 13: Выполнение обновления

```
/usr/pgsql-15/bin/pg_upgrade \
-b /usr/pgsql-11/bin \
-B /usr/pgsql-15/bin \
-d /data/postgres/11/data \
```

```
-D /data/postgres/15/data \
--link --jobs 10
```

После завершения будут сгенерированы скрипты: `analyze_new_cluster.sh`, `delete_old_cluster.sh`, `update_extensions.sql`.

## Шаг 14: Копирование конфигурации

Перенесите конфиги из старого кластера и адаптируйте под PostgreSQL 15:

```
cp -av /data/postgres/11/data/*.conf /data/postgres/15/data/
# Удалите устаревшие параметры
sed -i '/wal_keep_segments/d' /data/postgres/15/data/postgresql.conf
```

## Часть 5: Запуск обновлённого кластера

### Шаг 15: Ручной запуск PostgreSQL 15 (тест)

Проверьте, что новый кластер запускается:

```
/usr/pgsql-15/bin/pg_ctl -D /data/postgres/15/data/ -l /data/postgres/15/log
start
```

Если всё в порядке — остановите:

```
/usr/pgsql-15/bin/pg_ctl -D /data/postgres/15/data/ stop
```

### Шаг 16: Обновление конфигурации Patroni под PostgreSQL 15

Измените переменные в Ansible-ветке:

```
---
postgresql_version: "15"
patroni_scope: "pg{{ postgresql_version }}"
# остальные параметры без изменений
```

### Шаг 17: Применение конфигурации Patroni на мастере

```
ansible-playbook playbook_site.yaml --diff -l ycpb-dwh-db01s.digital.kfc.ru  
--tag patroni
```

## Шаг 18: Запуск Patroni на мастере

```
systemctl start patroni
```

## Шаг 19: Выполнение пост-апгрейд скриптов

На мастере выполните:

```
psql -f /data/postgres/15/update_extensions.sql  
/usr/pgsql-15/bin/vacuumdb --all --analyze-in-stages
```

## Шаг 20: Проверка архивации WAL

Убедитесь, что архивация работает:

- Проверьте логи PostgreSQL
- Убедитесь, что новые WAL-файлы появляются в архиве

## Шаг 21: Синхронизация реплик

Для каждой реплики:

1. Выполните rsync (если используется физическая репликация через копирование):

```
rsync --archive --delete --hard-links --size-only --no-inc-recursive \  
/data/postgres <replica_host>:/data --progress --verbose
```

2. Примените конфигурацию Patroni:

```
ansible-playbook playbook_site.yaml --diff -l <replica_host> --tag  
patroni
```

3. Запустите Patroni:

```
systemctl start patroni
```

## Шаг 22: Удаление старого кластера

После полной проверки работоспособности:

```
./delete_old_cluster.sh
```

---

## Часть 6: Финальная проверка

- Убедитесь, что кластер в состоянии `running` и `synced`
- Проверьте репликацию: `patronictl list`
- Выполните тестовые запросы от приложений
- Убедитесь, что мониторинг не фиксирует ошибок

| **Обновление завершено. Кластер работает на PostgreSQL 15.**