

ТОЛЬКО НЕОБХОДИМАЯ ИНФОРМАЦИЯ!

БОЛЕЕ
3
КЛАСТЕРОВ
РАЗВЕРНУТО

QNB

Разворачиваем кластер
на Qluster за три дня с нуля

серия для

“ чайников ”

- ✓ Установка и настройка с нуля
- ✓ Репликация и отказоустойчивость
- ✓ Мониторинг и бэкапы

КРАТКИЙ СПРАВОЧНИК
1-Е ИЗДАНИЕ
Для сомневающихся

Команда QNB

Авторы всех развёрнутых кластеров



QNB Press

<https://quantom.info/>

*Команда разработчиков QNB
«Авторы всех развёрнутых кластеров»*

Руководство по развёртыванию кластера Qluster на QNB

Краткий справочник для сомневающихся
1-е издание

*Москва
2026*

УДК 004.658:004.725
ББК 32.973.2-018.2

Руководство по развёртыванию кластера Qluster на QNB: Краткий справочник для сомневающийся, 1-е издание / Команда разработчиков QNB, «Авторы всех развёрнутых кластеров» - Москва, 2026. - 32 с.

В справочнике обсуждаются вопросы настройки параметров, конфигурирования на примере развёртывания конкретного кластера СУБД «Квант-Гибрид», российского форка PostgreSQL, с помощью сравнительно нового кластерного программного обеспечения Qluster. Qluster разработан тем же производителем, представляет собой альтернативу известному пакету Patroni, но основан на модифицированном и оптимизированном алгоритме консенсуса и написан на языке программирования Rust.

УДК 004.658:004.725
ББК 32.973.2-018.2

© KVANTOM LLC, 2019-2026

Содержание

1. Общие сведения о компонентах	5
1.1. Что такое QHB Qluster	5
1.2. Компоненты и их назначение	5
1.3. Порты	5
1.4. Общая схема	6
2. Условия данного руководства	7
2.1. Пакеты QHB	7
3. Установка пакетов	8
3.1. На узлах кластера (qluster-01, qluster-02, qluster-03)	8
3.2. На сервере QCP (qluster-nfs)	8
3.3. Проверка установки	8
4. Создание пользователя и настройка NFS	9
4.1. Создание пользователя qhb	9
4.2. Настройка NFS-сервера (qluster-nfs — 192.168.90.52)	9
4.3. Настройка NFS-клиентов на узлах кластера	10
5. Подготовка директорий	12
6. Развёртывание конфигурации	13
6.1. Конфигурация QHB (qhb.conf)	13
6.2. Конфигурация аутентификации (qhb_hba.conf)	14
6.3. Конфигурация Qluster (config.yaml)	16
6.4. Файл окружения Qluster	17
6.5. Активация сервиса Qluster	18
7. Инициализация кластера	19
7.1. Остановка и очистка (при чистой установке)	19
7.2. Подготовка узлов	19
7.3. Запуск лидера (qluster-01)	19
7.4. Запуск первого фолловера (qluster-02)	20
7.5. Запуск второго фолловера (qluster-03)	20
7.6. Проверка репликации	21
8. Настройка QCP (Connection Pooler)	22
8.1. Создание конфигурации QCP	22
8.2. Запуск QCP	24
8.3. Проверка подключения через QCP	24
9. Проверка работоспособности кластера	26
9.1. Проверка узлов	26
9.2. Проверка репликации (на лидере)	26
9.3. Проверка QCP (на сервере QCP)	26
9.4. Кластерная карта (с любого узла)	26
10. Переинициализация кластера	27
10.1. Остановка всех сервисов	27
10.2. Очистка данных	27
10.3. Запуск лидера	27
10.4. Запуск фолловеров (по одному)	27
10.5. Настройка конфигурации кластера через API (на лидере)	28
10.6. Запуск QCP	28
11. Диагностика проблем QCP	29
12. Полное удаление кластера	30
12.1. Остановка сервисов	30

12.2. Отмонтирование NFS (на узлах кластера)	30
12.3. Удаление пакетов	30
12.4. Удаление директорий	31
12.5. Удаление NFS-конфигурации (на NFS-сервере)	31
12.6. Удаление пользователя (на всех серверах)	31
12.7. Проверка	31
13. Краткая шпаргалка	32

1. Общие сведения о компонентах

1.1. Что такое QNB Qluster

QNB Qluster — это кластерное решение на базе СУБД QNB, обеспечивающее отказоустойчивость и масштабирование чтения через потоковую репликацию.

Кластер строится из нескольких серверов (обычно от 3 до 5, нечётное число для корректного голосования). Один из серверов выполняет роль **лидера** (принимает запись), остальные — **фолловеры** (реплики, принимают чтение и готовы заменить лидера при его отказе).

1.2. Компоненты и их назначение

QNB — СУБД «Квант-Гибрид», совместимая с PostgreSQL. Устанавливается из пакета qnb-core на каждый узел кластера. По умолчанию принимает соединения на порту **5432**.

Qluster — менеджер кластера. Управляет выбором лидера, репликацией, добавлением и удалением узлов. Устанавливается из пакета qluster на каждый узел кластера. Предоставляет HTTP API на порту **8080** для мониторинга и управления.

QBackup — утилита для работы с WAL-архивами (журналами транзакций). Устанавливается из пакета qbackup на каждый узел кластера. Используется для архивирования и восстановления WAL-файлов.

QCP (Connection Pooler) — пулер соединений. Принимает подключения от приложений и распределяет их по узлам кластера. Устанавливается из пакета qcp. По умолчанию принимает соединения на порту **6432**. QCP может работать на отдельном сервере, на одном из узлов кластера или на NFS-сервере — размещение определяет администратор.

NFS — сетевая файловая система, используемая для общего хранения WAL-архивов. WAL-архивы должны быть доступны всем узлам кластера, поэтому они хранятся на NFS-шаре, которая монтируется на каждый узел. NFS-сервер может быть отдельной машиной, либо совмещён с QCP или другой ролью.

Что такое Leader и Follower?

Leader (лидер) — основной узел, на который идёт запись данных.

Follower (фолловер, реплика) — узел, который получает копии данных от лидера. Используется для чтения и обеспечения отказоустойчивости.

Если лидер выходит из строя, один из фолловеров автоматически становится новым лидером.

Что такое WAL?

Write-Ahead Log (WAL) — журнал предзаписи. Перед тем как изменить данные, СУБД записывает изменения в WAL. Эти записи используются для репликации (передачи данных на фолловеры) и для восстановления после сбоя.

1.3. Порты

Все порты можно изменить в конфигурационных файлах соответствующих компонентов. Значения по умолчанию:

Порт	Компонент	Назначение
5432	QNB	Подключения к СУБД
6432	QCP	Подключения через пулер
8080	Qluster	HTTP API управления кластером
9999/udp	Qluster	Межузловое взаимодействие (multicast)

Таблица 1. Порты

1.4. Общая схема

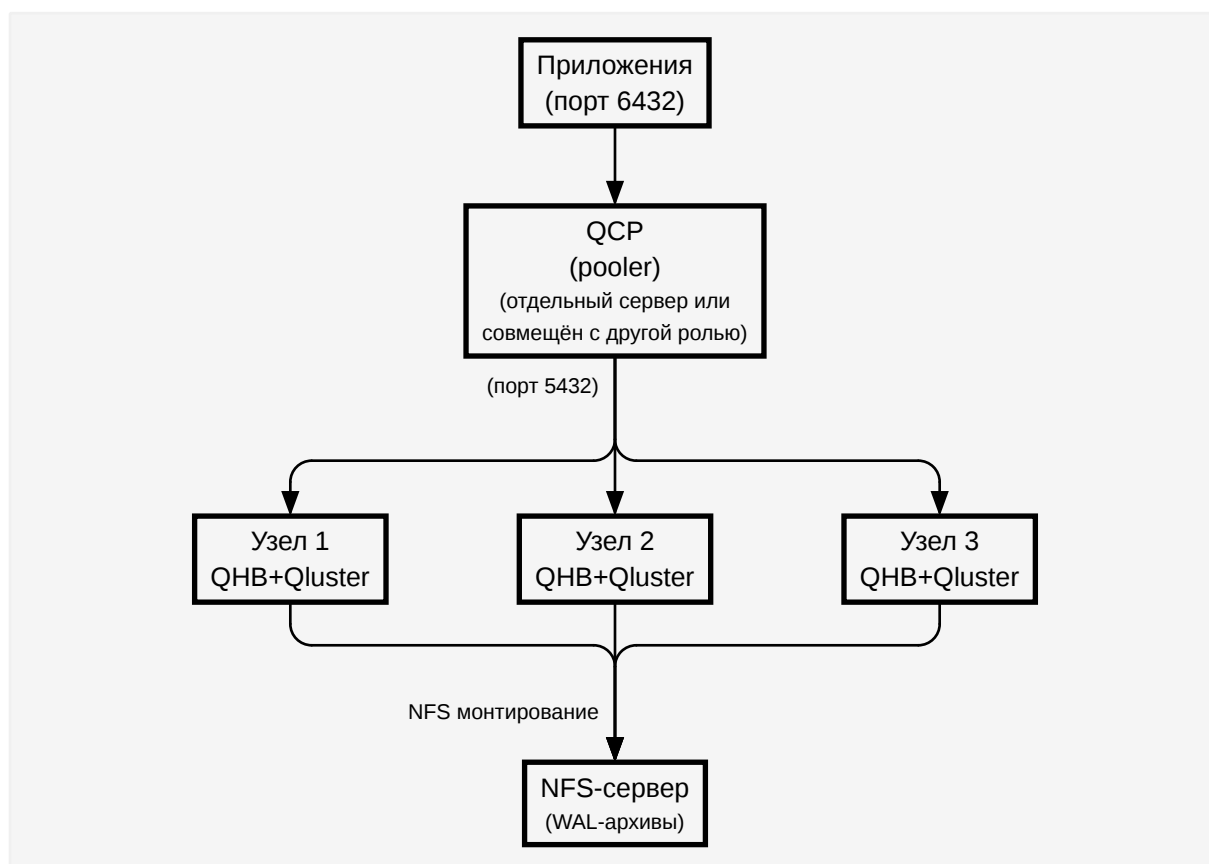


Схема 1. Схема стенда

Qluster автоматически назначает роли: один из узлов становится лидером, остальные — фолловерами. При сбое лидера происходит автоматическое переключение.

2. Условия данного руководства

В данном руководстве мы разворачиваем кластер на **4 серверах** со следующим распределением ролей. В вашей ситуации количество серверов и распределение ролей может отличаться — адаптируйте инструкции под свои условия.

Сервер	IP-адрес	Роль
qluster-01	192.168.90.48	Узел кластера (начальный лидер)
qluster-02	192.168.90.49	Узел кластера (фолловер)
qluster-03	192.168.90.50	Узел кластера (фолловер)
qluster-nfs	192.168.90.52	NFS-сервер + QCP

Таблица 2. Сервера и распределение ролей

Особенности нашей конфигурации:

- 3 узла кластера (минимум для кворумной репликации)
- QCP совмещён с NFS-сервером на отдельной машине
- Все серверы в одной подсети 192.168.90.0/24
- ОС: Ubuntu/Debian (команды для RHEL/CentOS приводятся отдельно, где есть различия)

Для вашего случая:

количество узлов кластера выбирайте нечётным (3 или 5) для корректной работы кворума. QCP и NFS можно разместить на отдельных серверах или совместить с узлами кластера. IP-адреса, подсети и имена серверов замените на свои.

2.1. Пакеты QHB

Пакеты для необходимой операционной системы доступны в репозитории Quantom (<https://repo.quantom.info/qhb/std-1/>).

Как ими правильно воспользоваться, описано на «Станице загрузки» документации QHB (<https://repo.quantom.info/qhb/std-1/doc/Appendix/download.html>).

Для развёртывания кластера потребуются следующие пакеты:

Пакет	Где устанавливать	Назначение
qhb-core	Все узлы кластера	Ядро СУБД QHB
qluster	Все узлы кластера	Менеджер кластера
qbackup	Все узлы кластера	Утилита работы с WAL-архивами
qcp	Сервер с ролью QCP	Пулер соединений

Таблица 3. Пакеты для установки

Скачайте пакеты нужной версии для вашей ОС и разместите на серверах, где они будут установлены.

3. Установка пакетов

3.1. На узлах кластера (qluster-01, qluster-02, qluster-03)

Выполните на **каждом узле** кластера. Предполагается, что пакеты уже скачаны и размещены в каталоге /tmp/qhb-packages/.

Debian/Ubuntu:

```
sudo apt install -y /tmp/qhb-packages/qhb-core_*.deb
sudo apt install -y /tmp/qhb-packages/qluster_*.deb
sudo apt install -y /tmp/qhb-packages/qbackup_*.deb
```

RHEL/CentOS:

```
sudo yum localinstall -y /tmp/qhb-packages/qhb-core-*.rpm
sudo yum localinstall -y /tmp/qhb-packages/qluster-*.rpm
sudo yum localinstall -y /tmp/qhb-packages/qbackup-*.rpm
```

Что делают эти команды?

apt install (Debian/Ubuntu) и yum localinstall (RHEL/CentOS) устанавливают программу из локального файла пакета, автоматически подтягивая зависимости — если для работы программы нужны другие пакеты, система установит их сама.

3.2. На сервере QCP (qluster-nfs)

Debian/Ubuntu:

```
sudo apt install -y /tmp/qhb-packages/qcp_*.deb
```

RHEL/CentOS:

```
sudo yum localinstall -y /tmp/qhb-packages/qcp-*.rpm
```

3.3. Проверка установки

На **всех** серверах:

```
dpkg -l | grep -E 'qhb|qluster|qbackup|qcp' # Debian/Ubuntu
# или
rpm -qa | grep -E 'qhb|qluster|qbackup|qcp' # RHEL/CentOS
```

Установленные пакеты должны отобразиться в списке.

4. Создание пользователя и настройка NFS

4.1. Создание пользователя qhb

Выполните на **каждом сервере** (все узлы кластера + NFS-сервер):

```
# Создаём группу qhb с фиксированным gid
sudo groupadd -g 1003 qhb

# Создаём пользователя qhb с фиксированными uid и gid
sudo useradd -u 1001 -g qhb -m -s /bin/bash qhb

# Проверяем
id qhb
# Ожидаемый вывод: uid=1001(qhb) gid=1003(qhb) groups=1003(qhb)
```

Зачем одинаковые uid/gid на всех серверах?

NFS идентифицирует пользователей по числовым uid/gid, а не по имени. Если uid/gid пользователя qhb будут отличаться на разных серверах, возникнут проблемы с правами доступа к файлам на NFS-шаре.

Числа 1001 и 1003 — пример, вы можете использовать другие, главное — чтобы они были **одинаковыми на всех серверах**.

4.2. Настройка NFS-сервера (qluster-nfs — 192.168.90.52)

Debian/Ubuntu:

```
sudo apt update
sudo apt install -y nfs-kernel-server nfs-common
```

RHEL/CentOS:

```
sudo yum install -y nfs-utils
sudo systemctl enable --now nfs-server
```

Создаём директорию и настраиваем экспорт:

```
# Создаём директорию для WAL-архивов
sudo mkdir -p /qhb-backup
sudo chown qhb:qhb /qhb-backup
sudo chmod 755 /qhb-backup
```

Откройте файл /etc/exports в текстовом редакторе:

```
sudo nano /etc/exports
```

Добавьте строку, перечислив IP-адреса **всех узлов кластера**:

```
/qhb-backup 192.168.90.48(rw, sync, no_subtree_check, no_root_squash) \  
            192.168.90.49(rw, sync, no_subtree_check, no_root_squash) \  
            192.168.90.50(rw, sync, no_subtree_check, no_root_squash)
```

Адаптация:

замените IP-адреса на адреса ваших узлов кластера. Если узлов больше или меньше — добавьте или уберите соответствующие записи.

Что означают параметры:

- `rw` — разрешена запись и чтение
- `sync` — синхронная запись на диск (надёжнее, но чуть медленнее)
- `no_subtree_check` — отключает проверку поддерева (улучшает производительность)
- `no_root_squash` — root-пользователь на клиенте сохраняет root-права на NFS

Применяем настройки и запускаем NFS:

```
sudo exportfs -ra  
  
sudo systemctl restart nfs-kernel-server # Debian/Ubuntu  
# или  
sudo systemctl restart nfs-server # RHEL/CentOS  
  
sudo systemctl enable nfs-kernel-server # Debian/Ubuntu  
# или  
sudo systemctl enable nfs-server # RHEL/CentOS  
  
# Проверяем, что экспорт виден  
showmount -e localhost
```

Вывод должен показать что-то вроде:

```
Export list for localhost:  
/qhb-backup 192.168.90.48,192.168.90.49,192.168.90.50
```

4.3. Настройка NFS-клиентов на узлах кластера

Выполните на **каждом узле кластера** (qluster-01, qluster-02, qluster-03):

Debian/Ubuntu:

```
sudo apt install -y nfs-common
```

RHEL/CentOS:

```
sudo yum install -y nfs-utils
```

Создаём директории и монтируем NFS:

```
# Создаём локальные директории
sudo mkdir -p /opt/qhb-backup
sudo chown qhb:qhb /opt/qhb-backup
sudo chmod 750 /opt/qhb-backup

sudo mkdir -p /opt/qhb-backup/wal
sudo chown qhb:qhb /opt/qhb-backup/wal
sudo chmod 755 /opt/qhb-backup/wal

# Монтируем NFS-шару
sudo mount -t nfs -o rw,sync,hard,intr,_netdev \
    192.168.90.52:/qhb-backup /opt/qhb-backup/wal
```

Адаптация:

замените 192.168.90.52 на IP-адрес вашего NFS-сервера.

Проверяем:

```
# Должна показать NFS-шару
df -h /opt/qhb-backup/wal

# Тестируем запись от имени пользователя qhb
sudo -u qhb touch /opt/qhb-backup/wal/test_write.txt
sudo -u qhb rm -f /opt/qhb-backup/wal/test_write.txt
```

Чтобы NFS монтировалась автоматически при загрузке, добавьте строку в /etc/fstab:

```
sudo nano /etc/fstab
```

Добавьте:

```
192.168.90.52:/qhb-backup /opt/qhb-backup/wal nfs rw,sync,hard,intr,_netdev 0 0
```

Что такое /etc/fstab?

Файл, описывающий файловые системы, которые монтируются автоматически при загрузке ОС. Каждая строка — одна файловая система.

5. Подготовка директорий

Выполните на **каждом узле кластера** (qluster-01, qluster-02, qluster-03):

```
# Основная директория QHB
sudo mkdir -p /qhb
sudo chown qhb:qhb /qhb
sudo chmod 755 /qhb

# Директория данных (ВАЖНО: режим 0700!)
sudo mkdir -p /qhb/data
sudo chown qhb:qhb /qhb/data
sudo chmod 700 /qhb/data

# Удаляем lost+found, если есть (появляется на отдельных разделах)
sudo rm -rf /qhb/data/lost+found

# Директория для конфигурационных файлов-шаблонов
sudo mkdir -p /usr/local/qhb/data
sudo chown qhb:qhb /usr/local/qhb/data
sudo chmod 755 /usr/local/qhb/data

# Исправляем владельца директории Qluster (создаётся при установке пакета)
sudo chown -R qhb:qhb /opt/qluster
sudo chmod -R 775 /opt/qluster
```

Почему /qhb/data имеет режим 0700?

СУБД QHB (как и PostgreSQL) требует, чтобы директория данных была доступна **только** владельцу. Это мера безопасности — никто, кроме пользователя qhb, не должен иметь прямого доступа к файлам базы данных. Если права будут другими, СУБД откажется запускаться.

Проверяем результат:

```
ls -lad /qhb /qhb/data /usr/local/qhb/data /opt/qluster /opt/qhb-backup \
/opt/qhb-backup/wal
```

6. Развёртывание конфигурации

6.1. Конфигурация QHB (qhb.conf)

Создайте файл `/usr/local/qhb/data/qhb.conf` на **каждом узле** кластера со следующим содержимым:

```
# -----  
# QHB configuration file  
# -----  
  
listen_addresses = '*'  
max_connections = 100  
shared_buffers = 128MB  
dynamic_shared_memory_type = posix  
max_wal_size = 1GB  
min_wal_size = 80MB  
  
log_destination = 'stderr'  
logging_collector = on  
log_directory = 'log'  
log_filename = 'qhb-%Y-%m-%d.log'  
log_rotation_age = 1d  
log_timezone = 'Europe/Moscow'  
  
datestyle = 'iso, mdy'  
timezone = 'Europe/Moscow'  
lc_messages = 'ru_RU.UTF-8'  
lc_monetary = 'ru_RU.UTF-8'  
lc_numeric = 'ru_RU.UTF-8'  
lc_time = 'ru_RU.UTF-8'  
  
default_text_search_config = 'pg_catalog.english'  
  
wal_log_hints = on  
full_page_writes = on  
wal_level = replica  
archive_mode = on  
archive_command = '/usr/bin/qbackup backup-wal -B /opt/qhb-backup -f %f -p %p'  
restore_command = '/usr/bin/qbackup restore-wal -f /opt/qhb-backup/wal/%f -p %p'  
  
wal_receiver_create_temp_slot = on  
cluster_name = 'qluster1-qhb'
```

Параметры, которые следует адаптировать под свои условия:

Параметр	Описание	Рекомендации
max_connections	Максимум одновременных подключений	Увеличьте при большой нагрузке (200, 500...)
shared_buffers	Оперативная память для кеша данных	Рекомендуется 25% от общей RAM сервера (напр. 2GB для сервера с 8 GB)
max_wal_size	Максимальный размер WAL между контрольными точками	Увеличьте при интенсивной записи
log_timezone, timezone	Часовой пояс	Замените на свой (напр. UTC, Asia/Novosibirsk)
lc_messages, lc_monetary, lc_numeric, lc_time	Локаль	Замените на свою (напр. en_US.UTF-8)
cluster_name	Имя кластера для идентификации	Задайте произвольное имя

Таблица 4. Параметры конфигурации для адаптации

Параметры, критически важные для работы кластера (менять не рекомендуется без понимания последствий):

- wal_level = replica — необходим для потоковой репликации
- archive_mode = on — необходим для архивирования WAL
- archive_command и restore_command — пути к утилите qbackup; менять только если она установлена в другое место
- wal_log_hints = on — необходим для корректной работы с WAL

Устанавливаем владельца и права:

```
sudo chown qhb:qhb /usr/local/qhb/data/qhb.conf
sudo chmod 644 /usr/local/qhb/data/qhb.conf
```

6.2. Конфигурация аутентификации (qhb_hba.conf)

Создайте файл /usr/local/qhb/data/qhb_hba.conf на **каждом узле** кластера:

```

# QHB Client Authentication Configuration
# TYPE DATABASE USER ADDRESS METHOD

# Локальные подключения через Unix-сокеты
local all all trust

# Подключения с localhost
host all all 127.0.0.1/32 trust

# Подключения с узлов кластера
host all all 192.168.90.0/24 trust

# IPv6 localhost
host all all ::1/128 trust

# Репликация – локальная
local replication all trust
host replication all 127.0.0.1/32 trust

# Репликация – с узлов кластера
host replication all 192.168.90.0/24 trust

# Репликация – IPv6 localhost
host replication all ::1/128 trust

```

Что обязательно нужно адаптировать:

Строка	Что менять
host all all 192.168.90.0/24 trust	Замените 192.168.90.0/24 на подсеть ваших серверов. Если серверы в разных подсетях — добавьте отдельную строку для каждой
host replication all 192.168.90.0/24 trust	Аналогично — подсеть, из которой разрешена репликация

Таблица 5. Адаптация qhb_hba.conf

Что такое trust?

Метод аутентификации, означающий «доверять без пароля». Это удобно для начальной настройки и в защищённых сетях. Для продуктивных окружений рекомендуется использовать md5 или scram-sha-256 с настройкой паролей.

Что такое hba?

Host-Based Authentication — правила, определяющие кто, откуда и каким способом может подключаться к СУБД. Строки обрабатываются сверху вниз, применяется первое совпадение.

Устанавливаем права:

```
sudo chown qhb:qhb /usr/local/qhb/data/qhb_hba.conf
sudo chmod 644 /usr/local/qhb/data/qhb_hba.conf
```

6.3. Конфигурация Qluster (config.yaml)

Создайте файл `/opt/qluster/config.yaml` на **каждом узле** кластера. **Содержимое одинаковое для всех узлов**, за исключением двух параметров — `ip` и `interface`, которые зависят от конкретного сервера.

Пример для qluster-01 (192.168.90.48):

```
ip: "192.168.90.48"
api-port: 8080

udp:
  socket:
    group: "239.0.0.1:9999"
    interface: "ens18"
    pacing-rate: 268435456
    loopback: false
    send-buffer-size: 1048576
    recv-buffer-size: 1048576
  max-log-size: 64000
  send-buffer-capacity: 256
  recv-buffer-capacity: 256

cluster:
  node-timeout: 30s
  heartbeat-interval: 10s
  pre-voting-timeout: 10s
  leader-voting-timeout: 10s
  quorum-replication: true
  replication-factor: 0
  replication-retry-interval: 1s
  autoremove-enabled: false
  autoremove-minimum: 3
  remove-interval: 1m
  set-leader-timeout: 30s

hello-retry-interval: 1s
```

Параметры, которые нужно менять на каждом узле:

Параметр	Описание	Что менять
<code>ip</code>	IP-адрес этого узла	Обязательно замените на IP текущего сервера. На <code>qluster-02</code> будет <code>192.168.90.49</code> , на <code>qluster-03</code> — <code>192.168.90.50</code>
<code>interface</code>	Имя сетевого интерфейса	Замените <code>ens18</code> на имя вашего интерфейса (см. ниже как узнать)

Таблица 6. Параметры конфигурации Qluster для замены на каждом узле

Параметры, общие для всего кластера (одинаковые на всех узлах):

Параметр	Описание	Когда менять
group	Multicast-группа для обмена данными	Только если 239.0.0.1:9999 конфликтует с другими сервисами. Должен быть одинаковым на всех узлах
api-port	Порт HTTP API Qluster	По умолчанию 8080, измените если занят
autoremove-minimum	Минимум узлов для автоудаления	Задайте число, равное количеству узлов кластера

Таблица 7. Параметры конфигурации Qluster, общие для всего кластера

Как узнать имя сетевого интерфейса?

Выполните команду:

```
ip addr show
```

Вы увидите список интерфейсов с их IP-адресами. Нужен тот, на котором назначен IP из вашей рабочей подсети. Типичные имена: eth0, ens18, ens192, enp0s3.

Скопируйте конфигурацию в /etc/qluster/:

```
sudo mkdir -p /etc/qluster
sudo cp /opt/qluster/config.yaml /etc/qluster/config.yaml
sudo chown qhb:qhb /opt/qluster/config.yaml /etc/qluster/config.yaml
sudo chmod 644 /opt/qluster/config.yaml /etc/qluster/config.yaml
```

6.4. Файл окружения Qluster

Создайте файл /etc/systemd/system/qluster.env на **каждом узле** кластера:

```
sudo tee /etc/systemd/system/qluster.env > /dev/null << 'EOF'
# Qluster environment variables
CONFIG_PATH=/etc/qluster/config.yaml
PGDATA=/qhb/data
PGPORT=5432
PGUSER=qhb
PGHOME=/usr/local/qhb
BACKUP_PATH=/opt/qhb-backup
EOF
```

Что делает эта команда?

sudo tee записывает текст в файл с правами root.

Конструкция

```
<< 'EOF' ... EOF
```

(heredoc) позволяет вставить многострочный текст прямо в команде.

Адаптация:

если вы изменили порт QNB (по умолчанию 5432) или пути к данным — отразите это здесь.

6.5. Активация сервиса Qluster

На **каждом узле** кластера:

```
# Перезагружаем systemd, чтобы он увидел новые файлы
sudo systemctl daemon-reload

# Включаем автозапуск (сам сервис пока НЕ запускается)
sudo systemctl enable qluster
```

Что делает `systemctl enable`?

Регистрирует сервис для автоматического запуска при загрузке системы. Сам сервис при этом не запускается — мы сделаем это позже в правильном порядке.

7. Инициализация кластера

Порядок действий критически важен: сначала запускается лидер, затем фолловеры строго по одному, с паузами между ними.

7.1. Остановка и очистка (при чистой установке)

На **всех узлах** кластера:

```
# Останавливаем qluster (если он был запущен ранее)
sudo systemctl stop qluster

# Очищаем директорию данных
sudo rm -rf /qhb/data/*
sudo rm -rf /qhb/data/lost+found
```

Очистка WAL-архива — выполнить **один раз** с любого узла, где смонтирована NFS-шара:

```
sudo rm -rf /opt/qhb-backup/wal/*
```

7.2. Подготовка узлов

На **лидере (qluster-01)** — удаляем маркер add_node:

```
sudo rm -f /opt/qluster/add_node
```

На **каждом фолловере (qluster-02, qluster-03)** — создаём маркер add_node:

```
sudo -u qhb touch /opt/qluster/add_node
```

Что такое файл add_node?

Это пустой файл-маркер. Его наличие сообщает Qluster: «Этот узел — новый фолловер, он должен подключиться к лидеру и получить полную копию данных (base backup)». На лидере этого файла быть **не должно**.

7.3. Запуск лидера (qluster-01)

```
sudo systemctl start qluster

# Ждём ~15 секунд для инициализации
sleep 15

# Проверяем статус сервиса
sudo systemctl status qluster

# Смотрим логи (последние 50 строк)
sudo journalctl -u qluster -n 50 --no-pager
```

Ждём, когда СУБД QHB запустится и начнёт принимать соединения:

```
# Проверяем, слушает ли порт 5432
ss -tlnp | grep 5432

# Подключаемся к базе
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c "SELECT version();"
```

Если порт 5432 не появляется — смотрите логи:
sudo journalctl -u qluster -n 100 --no-pager
и логи QHB:
sudo -u qhb cat /qhb/data/log/qhb-\$(date +%Y-%m-%d).log.

7.4. Запуск первого фолловера (qluster-02)

Подождите 10 секунд после успешного запуска лидера. На **qluster-02**:

```
sudo systemctl start qluster

# Ждём – фолловер копирует данные с лидера, это может занять время
sleep 15

# Проверяем статус
sudo systemctl status qluster
sudo journalctl -u qluster -n 30 --no-pager

# Ждём появления порта 5432
ss -tlnp | grep 5432

# Проверяем, что узел в режиме реплики (должно вернуть "t" – true)
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c "SELECT
pg_is_in_recovery();"
```

Что такое recovery mode?

Фолловер работает в «режиме восстановления» — он получает данные от лидера и не принимает команды записи. Функция `pg_is_in_recovery()` возвращает `t` (true) на фолловере и `f` (false) на лидере.

7.5. Запуск второго фолловера (qluster-03)

Подождите 10 секунд после запуска qluster-02. На **qluster-03** выполните те же команды:

```
sudo systemctl start qluster
sleep 15
sudo systemctl status qluster
ss -tlnp | grep 5432
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c "SELECT
pg_is_in_recovery();"
```

7.6. Проверка репликации

На **лидере** (qluster-01) создаём тестовые данные:

```
# Создаём тестовую таблицу
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c \
  "CREATE TABLE IF NOT EXISTS qluster_test (id serial, created_at timestamp
  default now());"

# Вставляем строку
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c \
  "INSERT INTO qluster_test DEFAULT VALUES;"

# Ждём синхронизации
sleep 5
```

На **каждом фолловере** проверяем, что данные появились:

```
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c \
  "SELECT COUNT(*) FROM qluster_test;"
# Должно вернуть 1
```

Проверка статуса репликации на **лидере**:

```
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c \
  "SELECT * FROM pg_stat_replication;"
```

Вы должны увидеть строки по количеству фолловеров (в нашем случае — 2).

8. Настройка QCP (Connection Pooler)

QCP устанавливается на сервере, выделенном для роли пулера. В нашем случае — **qluster-nfs (192.168.90.52)**.

8.1. Создание конфигурации QCP

Создайте файл `/etc/qcp/config.yaml`:

```
sudo mkdir -p /etc/qcp
sudo nano /etc/qcp/config.yaml
```

Содержимое:

```
---
# QCP Configuration for QHB Cluster

listening_address: "0.0.0.0:6432"

log_level:
  default: Info
  qcp: Info

log_output: stderr
log_timestamps: true
log_labelling: auto

servers_retry_back_off:
  min_wait: "5 sec"
  max_wait: "15 sec"
  max_wait_at: 30

server:
  address:
    Tcp: "192.168.90.48:5432"

  inactive_timeout: "1 min"
  lifetime: "15 min"
  heart_beat_interval: "1 sec"
  host_resolution_timeout: "5 sec"

  connections:
    - username: qhb
      database: qhb
      backends:
        initial: 2
        min: 2
        max: 20
      client_auth:
        method: trust

  arena:
    chunk_size: 65 KB
    chunks_count: 1024

  relay_mode: Smart
  client_wait_limit: "1 min 30 sec"
  accept_clients_no_backends: false
  worker_threads: 4
```

Параметры, которые нужно адаптировать:

Параметр	Описание	Рекомендации
<code>listening_address</code>	Адрес и порт, на котором QCP принимает подключения	0.0.0.0:6432 — слушает на всех интерфейсах. Порт можно изменить
<code>address: Tcp</code>	IP и порт текущего лидера кластера	Укажите IP:порт начального лидера. QCP Leader Monitor (шаг 9) будет обновлять этот адрес автоматически при смене лидера
<code>username</code>	Пользователь для подключения к СУБД	Замените, если используете другого пользователя
<code>database</code>	База данных	Замените на имя вашей рабочей БД. Можно добавить несколько блоков в <code>connections</code> для разных баз
<code>max</code>	Максимум соединений к бэкенду	Увеличьте при высокой нагрузке
<code>worker_threads</code>	Число рабочих потоков	Рекомендуется равным числу ядер CPU
<code>client_auth: method</code>	Метод аутентификации клиентов	<code>trust</code> — без пароля; для продуктивной среды используйте другой метод

Таблица 8. Параметры конфигурации QCP, которые нужно адаптировать

Устанавливаем права:

```
sudo chown root:root /etc/qcp/config.yaml
sudo chmod 644 /etc/qcp/config.yaml
```

8.2. Запуск QCP

```
sudo systemctl daemon-reload
sudo systemctl enable qcp
sudo systemctl start qcp

# Ждём запуска
sleep 5

# Проверяем статус
sudo systemctl status qcp

# Проверяем порт
ss -tlnp | grep 6432

# Проверяем логи
sudo journalctl -u qcp -n 20 --no-pager
```

8.3. Проверка подключения через QCP

С сервера QCP (если установлен `psql`):

```
/usr/local/qhb/bin/psql -h 127.0.0.1 -p 6432 -U qhb -d qhb -c "SELECT version();"
```

Или с любого узла кластера:

```
sudo -u qhb /usr/local/qhb/bin/psql -h 192.168.90.52 -p 6432 \  
-U qhb -d qhb -c "SELECT version();"
```

Адаптация:

замените 192.168.90.52 на IP вашего сервера QCP.

Приложения подключаются к QCP по адресу <IP_сервера_QCP>:6432.

9. Проверка работоспособности кластера

9.1. Проверка узлов

На каждом узле кластера:

```
# Процесс QHB (должен быть найден)
ps aux | grep '[c]heckpointer'

# Сервис Qluster (должен быть active)
sudo systemctl is-active qluster

# Порт 5432 (должен слушать)
ss -tlnp | grep 5432

# Роль узла (PRIMARY или REPLICHA)
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -t -c \
  "SELECT CASE WHEN pg_is_in_recovery() THEN 'REPLICA' ELSE 'PRIMARY' END;"
```

9.2. Проверка репликации (на лидере)

```
# Подключённые реплики
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c \
  "SELECT application_name, client_addr, state, sync_state FROM
pg_stat_replication;"

# Текущая позиция WAL
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -t -c \
  "SELECT pg_current_wal_lsn();"
```

9.3. Проверка QCP (на сервере QCP)

```
sudo systemctl is-active qcp
ss -tlnp | grep 6432
```

9.4. Кластерная карта (с любого узла)

```
curl -s http://localhost:8080/map
```

> Для более читаемого вывода JSON можно использовать утилиту jq: `curl -s http://localhost:8080/map | jq '.'`

10. Переинициализация кластера

ВНИМАНИЕ:

Эта процедура **УДАЛИТ ВСЕ ДАННЫЕ** в кластере!

Используйте только если необходимо начать с нуля.

10.1. Остановка всех сервисов

На **каждом узле** кластера (по одному):

```
sudo systemctl stop qluster
```

На сервере QCP:

```
sudo systemctl stop qcp
```

10.2. Очистка данных

На лидере:

```
sudo rm -rf /opt/qhb-backup/wal/*      # Очищаем WAL-архив
sudo rm -f /opt/qluster/add_node     # Убираем маркер
sudo -u qhb rm -rf /qhb/data/*       # Очищаем данные БД
```

10.3. Запуск лидера

```
sudo systemctl start qluster
sleep 20
sudo systemctl status qluster
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d postgres -c "SELECT version();"
```

10.4. Запуск фолловеров (по одному)

На **каждом фолловере** (с паузой 10 секунд между ними):

```
sudo rm -rf /qhb/data/*              # Очищаем данные
sudo -u qhb touch /opt/qluster/add_node # Создаём маркер
sleep 10
sudo systemctl start qluster
sleep 20
sudo systemctl status qluster
sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d postgres -c "SELECT
pg_is_in_recovery();"
```

10.5. Настройка конфигурации кластера через API (на лидере)

```
curl -X POST http://localhost:8080/config \  
-H 'Content-Type: application/json' \  
-d '{  
  "autoremove-enabled": true,  
  "autoremove-minimum": 0,  
  "heartbeat-interval": "5s",  
  "leader-voting-timeout": "5s",  
  "node-timeout": "5s",  
  "pre-voting-timeout": "5s",  
  "quorum-replication": true,  
  "replication-factor": 0,  
  "replication-retry-interval": "5s",  
  "set-leader-timeout": "30s"  
}'
```

Ключевые параметры:

- `heartbeat-interval` — как часто узлы проверяют доступность друг друга
- `node-timeout` — через сколько считать узел недоступным
- `set-leader-timeout` — таймаут на выбор нового лидера
- `quorum-replication: true` — данные подтверждаются большинством узлов (повышает надёжность)

10.6. Запуск QCP

На сервере QCP:

```
sudo systemctl start qcp  
sleep 5  
ss -tlnp | grep 6432
```

11. Диагностика проблем QCP

Если QCP не работает, выполните пошаговую проверку на сервере QCP.

Шаг 1 — Сервис:

```
sudo systemctl status qcp --no-pager -l
sudo systemctl is-active qcp
```

Шаг 2 — Процесс:

```
ps aux | grep -v grep | grep qcp
pgrep -f qcp
```

Шаг 3 — Порт:

```
ss -tlnp | grep 6432
```

Шаг 4 — Конфигурация:

```
cat /etc/qcp/config.yaml
```

Шаг 5 — Логи:

```
sudo journalctl -u qcp -n 50 --no-pager
sudo journalctl -u qcp -n 50 --no-pager | grep -i "error\|fail\|fatal"
```

Шаг 6 — Прямое подключение к узлам (минуя QCP):

```
PGPASSWORD='qhb' psql -h 192.168.90.48 -p 5432 -U qhb -d qhb -c "SELECT 1;"
PGPASSWORD='qhb' psql -h 192.168.90.49 -p 5432 -U qhb -d qhb -c "SELECT 1;"
PGPASSWORD='qhb' psql -h 192.168.90.50 -p 5432 -U qhb -d qhb -c "SELECT 1;"
```

Адаптация:

замените IP-адреса на ваши.

Шаг 7 — Сетевая доступность:

```
for node in 192.168.90.48 192.168.90.49 192.168.90.50; do
  echo "=== $node ==="
  ping -c 2 $node
  nc -zv $node 5432 2>&1
  echo ""
done
```

12. Полное удаление кластера

ВНИМАНИЕ:

Эта процедура полностью и необратимо удалит кластер, все данные и конфигурацию!

12.1. Остановка сервисов

На **каждом узле** кластера:

```
sudo systemctl stop qluster
sudo systemctl disable qluster
sudo pkill -9 qhb || true
sleep 5
```

На **сервере QCP**:

```
sudo systemctl stop qcp
sudo systemctl disable qcp
```

12.2. Отмонтирование NFS (на узлах кластера)

```
sudo umount /opt/qhb-backup/wal

# Удаляем запись из fstab
sudo sed -i '/qhb-backup/d' /etc/fstab
```

12.3. Удаление пакетов

На **узлах кластера**:

Debian/Ubuntu:

```
sudo apt purge -y qluster qbackup qhb-core
```

RHEL/CentOS:

```
sudo yum remove -y qluster qbackup qhb-core
```

На **сервере QCP**:

Debian/Ubuntu:

```
sudo apt purge -y qcp
```

RHEL/CentOS:

```
sudo yum remove -y qcp
```

12.4. Удаление директорий

На узлах кластера:

```
sudo rm -rf /qhb
sudo rm -rf /opt/qhb-backup
sudo rm -rf /opt/qluster
sudo rm -rf /usr/local/qhb
sudo rm -rf /etc/qluster
sudo rm -f /etc/systemd/system/qluster.env
sudo rm -f /lib/systemd/system/qluster.service
sudo systemctl daemon-reload
```

На сервере QCP:

```
sudo rm -rf /etc/qcp
```

12.5. Удаление NFS-конфигурации (на NFS-сервере)

```
sudo sed -i '/qhb-backup/d' /etc/exports
sudo exportfs -ra
sudo rm -rf /qhb-backup
```

12.6. Удаление пользователя (на всех серверах)

```
sudo userdel -r qhb
sudo groupdel qhb
```

12.7. Проверка

```
id qhb 2>&1 # Должно вернуть ошибку «no such user»
ps aux | grep -E 'qhb|qluster' # Не должно быть процессов
ls -d /qhb /opt/qhb-backup /opt/qluster /usr/local/qhb /etc/qluster 2>&1
# Все директории должны отсутствовать
```

13. Краткая шпаргалка

Действие	Команда
Статус Qluster	<code>sudo systemctl status qluster</code>
Логи Qluster	<code>sudo journalctl -u qluster -f</code>
Статус QCP	<code>sudo systemctl status qcp</code>
Логи QCP	<code>sudo journalctl -u qcp -f</code>
Роль узла	<code>sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c "SELECT pg_is_in_recovery();"</code>
Статус репликации	<code>sudo -u qhb /usr/local/qhb/bin/psql -U qhb -d qhb -c "SELECT * FROM pg_stat_replication;"</code>
Карта кластера	<code>curl -s http://localhost:8080/map</code>
Конфигурация кластера	<code>curl -s http://localhost:8080/config</code>
Подключение через QCP	<code>psql -h <IP_QCP> -p 6432 -U qhb -d qhb</code>
Логи монитора лидера	<code>tail -f /var/log/qcp-leader-monitor.log</code>
Логи БД	<code>tail -f /qhb/data/log/qhb-\$(date +%Y-%m-%d).log</code>

Таблица 9. Наиболее востребованные действия