

УТВЕРЖДЕН
643.72410666.00067-07 98 01-ЛУ

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «JАТОВА»

Руководство по настройке. Часть 2.
Управление режимом работы узлов кластера.
Компонент «jaDog»

643.72410666.00067-07 98 02-02

Листов 129

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

АННОТАЦИЯ

Во второй части документа приведены сведения, необходимые для настройки отказоустойчивого кластера защищенной системы управления базами данных «Jatoba» (далее — СУБД «Jatoba») с использованием компонента «jaDog», входящего в состав СУБД «Jatoba». Настоящее руководство предназначено для администратора СУБД «Jatoba».

Представленные в документе снимки экрана могут отличаться для различных версий настраиваемой СУБД и предназначены для демонстрации хода настройки отказоустойчивого кластера СУБД «Jatoba».

Администратор СУБД «Jatoba» должен иметь навыки по работе с системами управления базами данных (СУБД) PostgreSQL или защищенной СУБД «Jatoba» (ООО «Газинформсервис»).



Примеры в данном документе приведены для СУБД «Jatoba» версии ядра 5, а также для СУБД «Jatoba» версии ядра 6.

Для СУБД «Jatoba» версий ядра 5 и 6 используется версия компонента — 3.2.

На приведенных в руководстве иллюстрациях версия компонента «jaDog» может отличаться от фактической.

Степени важности примечаний, применяемые в документе:



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием



Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Настоящий документ оснащён навигацией, гиперссылками и перекрестными ссылками. В приложении Adobe Acrobat навигация вызывается кнопкой «Закладки», как показано на рисунке 1.1.

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

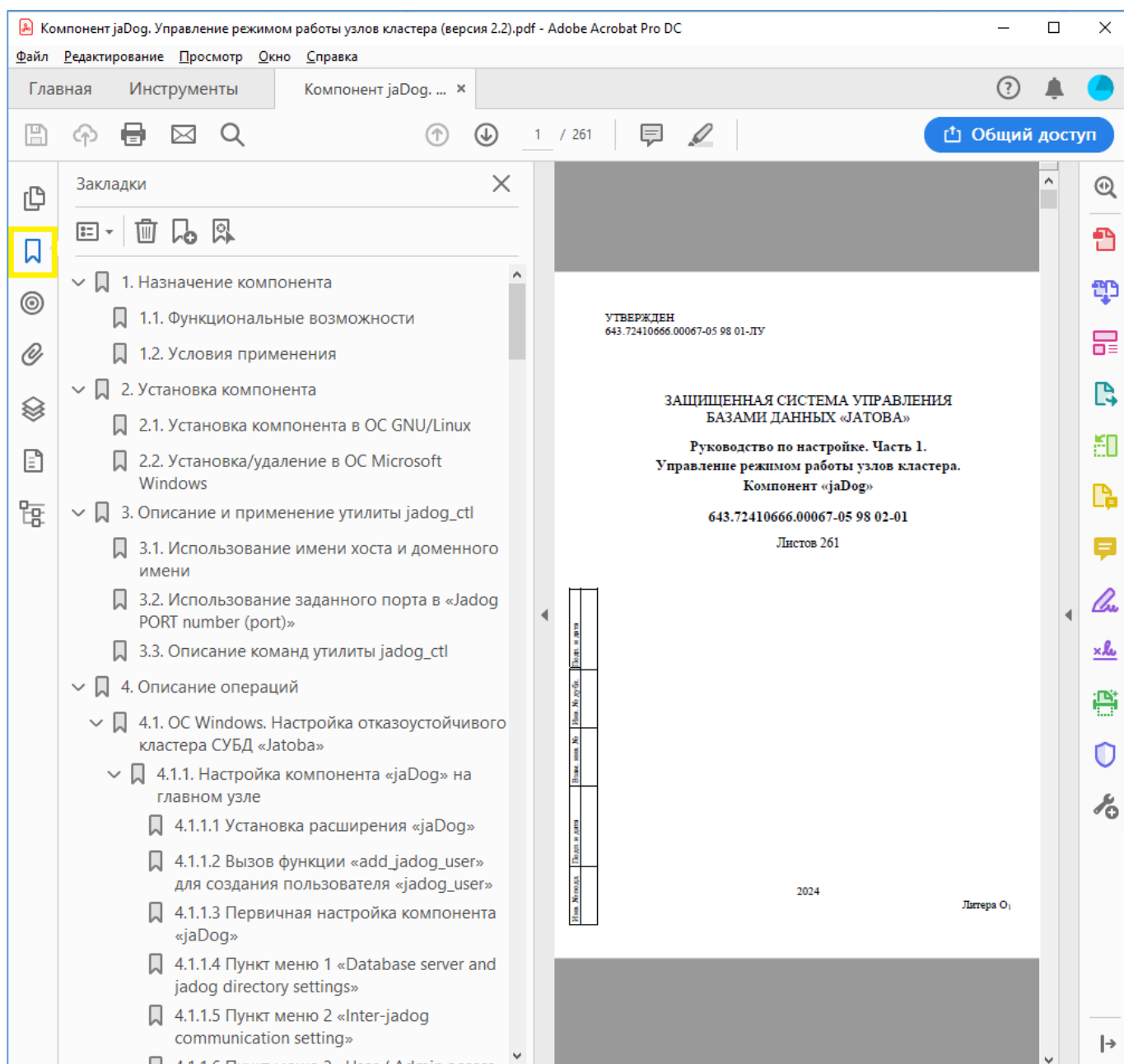


Рисунок 1.1 - Структура документа

СОДЕРЖАНИЕ

| | |
|--|----|
| 1. Общие сведения..... | 6 |
| 1.1. Виды кластеров | 6 |
| 1.1.1. Отказоустойчивый кластер | 6 |
| 1.1.2. Геораспределенный, отказоустойчивый кластер..... | 8 |
| 2. Перекрестная репликация. Использование файлов ответов..... | 10 |
| 2.1. Формирование файла ответов для кластера с перекрестной репликацией | 12 |
| 2.2. Запуск настройки кластера с перекрестной репликацией..... | 12 |
| 2.3. Настройка компонента «jaDog» для перекрестной репликации на резервном узле (втором экземпляре СУБД) с помощью файла ответов | 16 |
| 2.4. Настройка параметра dc_public_address | 18 |
| 2.5. Настройка параметров архивирования и восстановления WAL..... | 19 |
| 3. Перекрестная репликация. Настройка в ручном режиме | 22 |
| 3.1. Установка первого экземпляра СУБД..... | 22 |
| 3.2. Создание второго экземпляра СУБД..... | 23 |
| 3.3. Установка расширения «jadow» на первом экземпляре СУБД..... | 29 |
| 3.4. Создание пользователя «jadow_user» на первом экземпляре СУБД | 29 |
| 3.5. Настройка jadow для двух экземпляров СУБД | 31 |
| 3.5.1. Настройка компонента «jaDog» на главном узле (первом экземпляре СУБД) | 31 |
| 3.5.2. Настройка компонента «jaDog» на резервном узле (втором экземпляре СУБД) | 36 |
| 3.5.3. Настройка сервиса «jadow» для резервного узла второго экземпляра СУБД | 38 |
| 3.6. Создание кластера..... | 40 |
| 4. Каскадная репликация. Использование файлов ответов | 41 |
| 4.1. Формирование файла ответов для кластера с каскадной репликацией..... | 41 |
| 4.2. Запуск настройки кластера с каскадной репликацией..... | 42 |
| 4.3. Настройка компонента «jaDog» для каскадной репликации на подчиненном главном узле и резервном узле с помощью файла ответов | 42 |
| 5. Каскадная репликация. Настройка в ручном режиме | 44 |
| 5.1. Настройка компонента «jaDog» на главном узле при каскадной репликации | 44 |
| 5.2. Настройка и запуск компонента «jaDog» на резервных узлах при каскадной репликации..... | 45 |
| 5.3. Настройка кластера для каскадной репликации..... | 51 |
| 5.4. Работа кластера с каскадной репликацией в дата-центрах | 57 |
| 6. Геораспределенный, отказоустойчивый кластер. Решение JA_DTC_AS | 59 |
| 6.1. Режимы работы геораспределенного отказоустойчивого кластера..... | 59 |
| 6.1.1. Нормальный режим работы | 59 |
| 6.1.2. Выход из строя Master в DC1 | 59 |
| 6.1.3. dc_failover. Аварийный режим № 1. Последовательный выход из строя Master-серверов..... | 60 |
| 6.1.4. DTC_Network_Failure. Аварийный режим № 2 Пропадание связи между площадками..... | 61 |
| 6.1.5. DC_Promote. Смена роли DC | 62 |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| | |
|---|--|
| 6.1.6. Нештатный обратный DC_Promote..... | 62 |
| 6.1.7. Пример выполнения «DC_Promote» | 64 |
| 7. Перенаправление подключений между дата-центрами..... | 70 |
| 7.1. Настройка и запуск компонента «jaDog» на резервных узлах..... | 71 |
| 7.2. Установка и настройка компонента «jaPooler» | 77 |
| 7.3. Конфигурирование компонента «jaPooler» | 77 |
| 7.4. Запуск сервиса компонента..... | 80 |
| 8. Применение REST API | 82 |
| 8.1. Настройка функционала REST API | 82 |
| 8.1.1. Настройка параметров REST API через консольную утилиту «jadog»..... | 82 |
| 8.1.2. Настройка параметров REST API в файле ответов | 82 |
| 8.2. Проверка функционала REST API..... | 83 |
| 8.2.1. Проверка обработки запросов к REST API..... | 83 |
| 8.2.2. Коды возврата состояния jaDog REST API..... | 84 |
| 8.2.3. Результат выполнения запросов к REST API | 84 |
| 8.3. Функционал обработки и выполнения команд в утилите «jadog_ctl»..... | 85 |
| 8.3.1. Настройка асинхронной обработки команд (set async mode)..... | 85 |
| 8.3.2. Настройка синхронной обработки команд (set sync mode) | 85 |
| 8.3.3. Результат последней выполненной команды (get last response)..... | 86 |
| 8.3.4. Получение результата выполнения (get result) | 86 |
| 8.4. Перечень REST API команд и соответствие аналогам утилиты «jadog_ctl»..... | 88 |
| 9. Настройка группы кластеров (bundle) с компонентом «jaDog» в ручном режиме..... | 104 |
| 9.1. Создание группы кластеров..... | 105 |
| 9.2. Присоединение дополнительных кластеров к группе кластеров | 107 |
| 9.3. Файл состояния группы кластеров | 108 |
| 9.4. Отсоединение кластера от группы кластеров | 109 |
| 9.5. Удаление группы кластеров..... | 110 |
| 9.6. Особенности взаимодействия компонентов «jaDog» и «ja_Hipe_Cluster»..... | 110 |
| 9.6.1. Процедура принудительной смены ролей серверов (switchover, SO)..... | 111 |
| 10. Настройка группы кластеров с компонентом «jaDog» в автоматическом режиме..... | 112 |
| 10.1. Формирование файла ответов для группы кластеров | 112 |
| 10.2. Запуск автоматической настройки группы кластеров | 114 |
| Приложение 1 | 117 |
| Пример монтирования сетевого каталога для настройки архивирования WAL..... | 117 |
| Приложение 2 | 126 |
| Пример формирования сертификата для узла кластера | Ошибка! Закладка не определена. |
| Термины и определения | 127 |
| Перечень сокращений..... | 128 |

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Виды кластеров

1.1.1. Отказоустойчивый кластер

Отказоустойчивый кластер состоит из двух и более узлов (Node). В котором один узел выполняет роль Master, а остальные узлы выполняют роль Slave.

В сети должны быть зарезервированы IP-адреса для каждого из узлов и отдельно внешний IP-адрес (Public address), к которому будут подключаться пользователи СУБД с запросом на чтение и запись (RW).

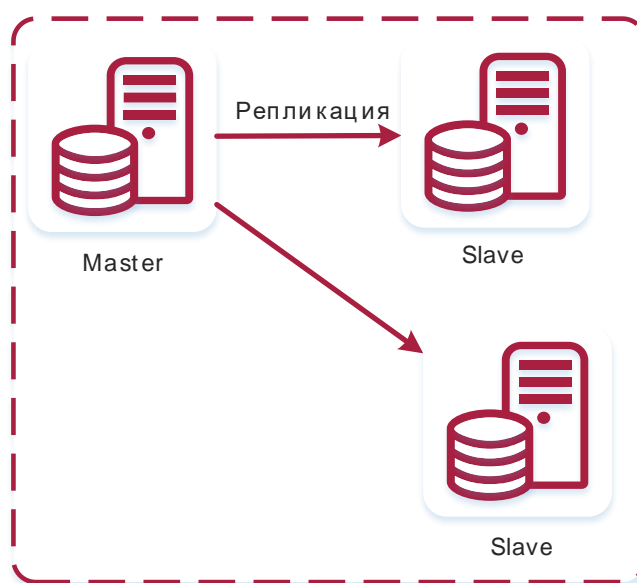


Рисунок 1.1 – Схема отказоустойчивого кластера

Для повышения консистентности данных и прекращения потерь в отказоустойчивом кластере может использоваться механизм параллельного архивирования WAL-файлов с Master узла и копированием архива WAL на Slave узлы.

WAL (Write-Ahead Log) файлы в PostgreSQL — это контрольные точки, которые используются для обеспечения целостности и согласованности данных в БД. Они пишутся на внешний сетевой диск перед тем, как изменения будут применены к таблицам, и служат для восстановления БД в случае сбоя.

В зависимости от типа репликации возможно построение кластера с каскадной или перекрёстной репликацией данных между узлами.

В зависимости от вида репликации возможно построение кластера с синхронной (по умолчанию) или асинхронной репликацией данных между узлами. Синхронная репликация устанавливается по умолчанию.



При добавлении узлов в кластер в ручном режиме (см. первую часть документа «Компонент jaDog. Управление режимом работы узлов кластера (версия 4, часть 1)» 643.72410666.00067-07 98 02-01), вид репликации узла указывается при помощи одной из команд:

```
cluster add slave [ip] [port] as ['node_name']
cluster add sync slave [ip] [port] as ['node_name']
```

При использовании файла ответов для автоматизированной настройки кластера асинхронная репликация указывается в секции nodes.[node_name]:

```
replication_type: async
```

Описание настройки кластеров с перекрестной или каскадной репликацией приводится в данном руководстве в разделах 2-3 и 4-5 соответственно.

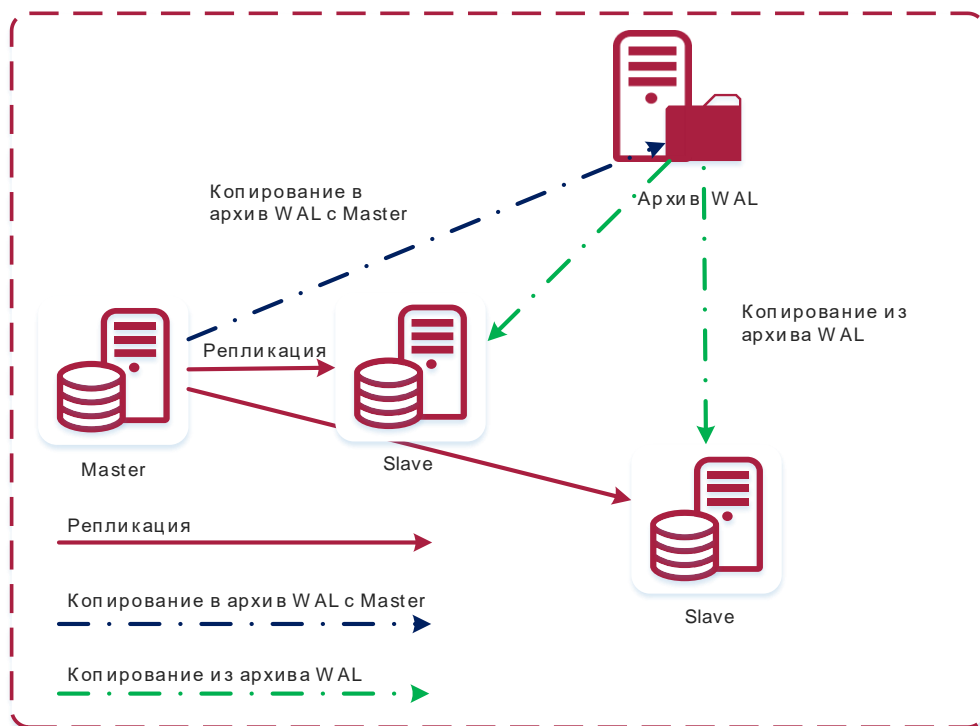


Рисунок 1.2 – Схема отказоустойчивого кластера с копированием WAL на файловый сервер

1.1.2. Геораспределенный, отказоустойчивый кластер

Геораспределенный, отказоустойчивый кластер подразумевает, что его узлы находятся на разных географически разделенных площадках – дата-центрах. В составе кластера должно быть от четырех узлов, которые могут находиться в разных подсетях.

Подразумевается, что каждая из площадок (дата-центров) находится в своей подсети и между подсетями настроены правила маршрутизации.

Описание настройки геораспределенного отказоустойчивого кластера приводится в разделе 1 данного руководства.

Параметры стека протоколов, используемых при построении геораспределенного отказоустойчивого кластера, приведены в таблице 1.1.

Таблица 1.1 – Параметры протоколов используемых СУБД

| Наименование | Протокол | Параметры |
|---|----------|-----------------|
| Database port (db_port) | Libpq | 5432 |
| Jadog TCP port (user_interface) | TCP | 54321, 54322 |
| Jadog PORT number (port) | Jadog | 12345, (Custom) |
| Jadog searching protocol port (jadog_search_port) | Jadog | 12346 |

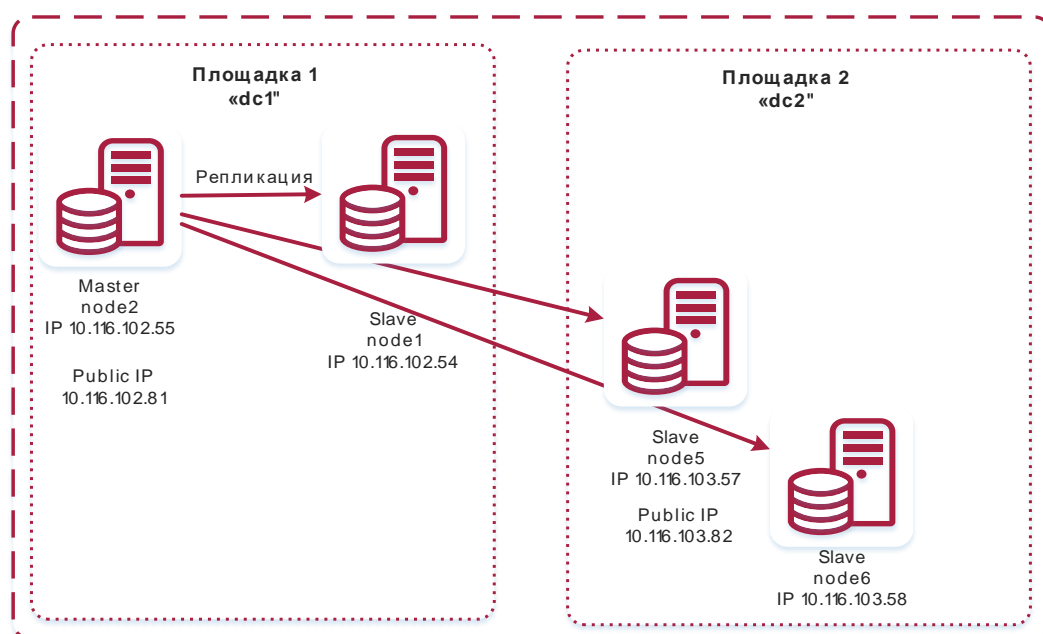


Рисунок 1.3 – Схема геораспределенного кластера

Файловый сервер для WAL – архивов может располагаться на отдельной площадке. Такое расположение позволяет минимизировать риск потери WAL-архива.

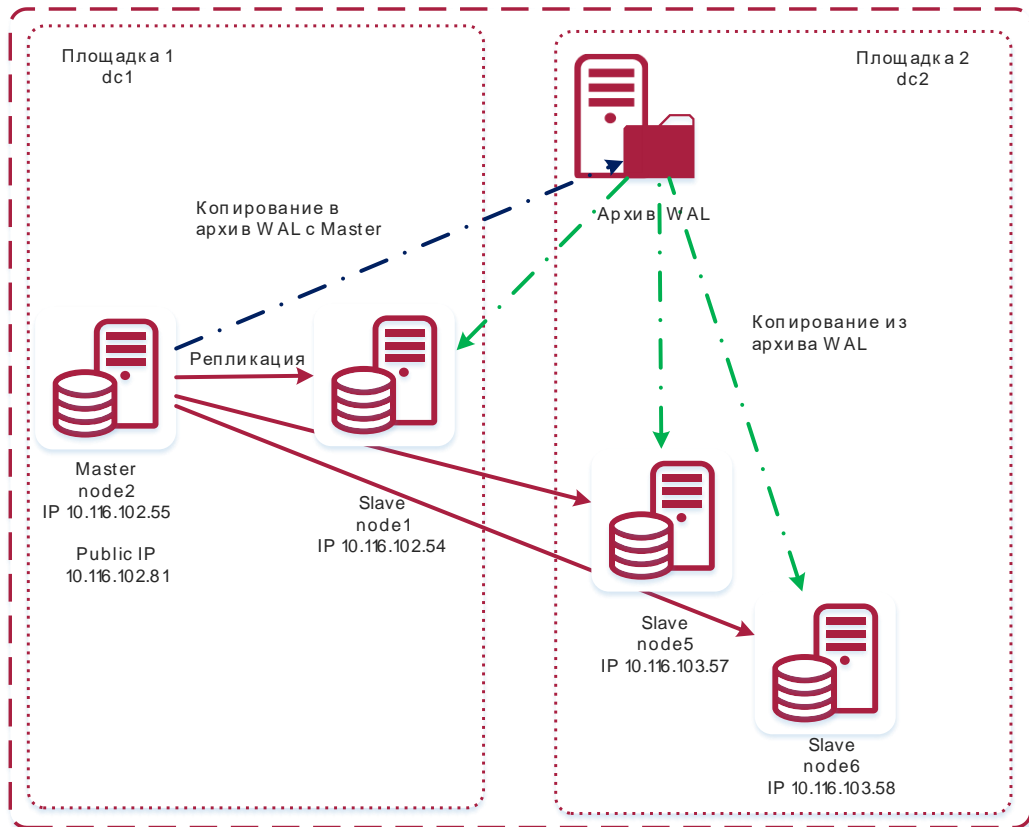


Рисунок 1.4 – Схема геораспределенного кластера с копированием WAL на файловый сервер

2. ПЕРЕКРЕСТНАЯ РЕПЛИКАЦИЯ. ИСПОЛЬЗОВАНИЕ ФАЙЛОВ ОТВЕТОВ

СУБД «Jatoba» поддерживает установку нескольких экземпляров СУБД на одном хосте (сервере), с последующим созданием кластера или кластеров с перекрестным типом репликации.

Применение такого типа репликации поможет снизить количество серверов, используемых для СУБД, и затрат на количество лицензий.

В представленной ниже схеме каждый из серверов включает в себя как минимум два экземпляра СУБД.

Второй экземпляр СУБД должен использоваться в роли «Slave» для кластера.

Таким образом два сервера поддерживают два кластера.

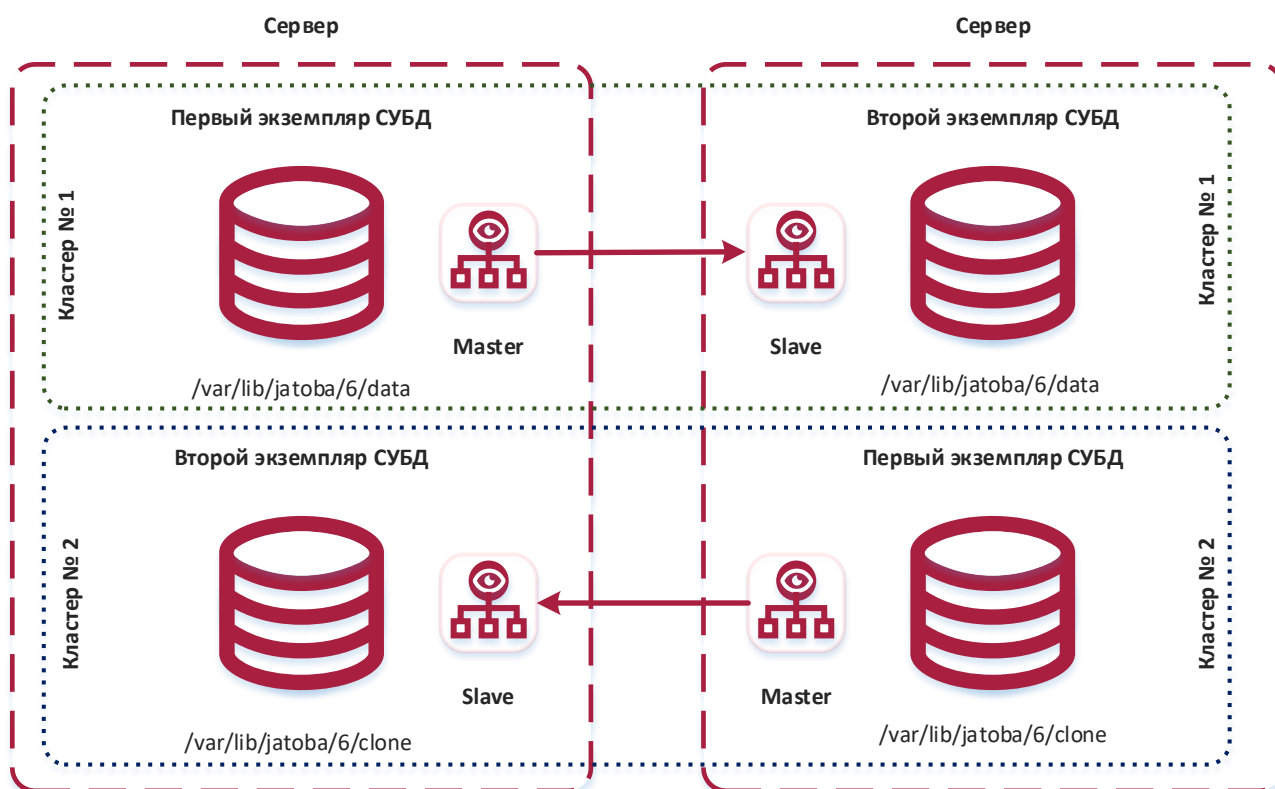


Рисунок 2.1 – Схема перекрестной репликации

В рассматриваемом примере установки и конфигурировании отказоустойчивого кластера «jaDog» с перекрестной репликацией на ОС Ubuntu 22.04 используются параметры сети кластера, приведенные в таблице 2.1.

Таблица 2.1 – Конфигурация сети узлов кластера с использованием перекрестной репликации

| № | Имя сервера | Подсеть кластера | Маска подсети | Номер сетевого порта Jatoba | Public IP | Роль | Имя кластера |
|---|-------------|------------------|---------------|-----------------------------|------------------|--------|--------------|
| | | 10.116.102.0/24 | | | | | |
| | | IP-адрес | | | | | |
| 1 | Node1 | 10.116.102.54/24 | 255.255.255.0 | 5432 | 10.116.102.81/24 | Master | cluster1 |
| 2 | Node2 | 10.116.102.55/24 | 255.255.255.0 | 5433 | 10.116.102.81/24 | Slave | |
| 3 | Node3 | 10.116.102.55/24 | 255.255.255.0 | 5432 | 10.116.102.82/24 | Master | cluster2 |
| 4 | Node4 | 10.116.102.54/24 | 255.255.255.0 | 5433 | 10.116.102.82/24 | Slave | |

Установка компонента «jaDog» на главном и резервном узле отличаются по количеству выполняемых действий.



Предварительными условиями для настройки компонента «jaDog» на главном узле кластера с перекрестной репликацией являются:

- установленная СУБД «Jatoba» (см. «Руководство по установке»);
- установленный компонент «jaDog» (см. п.п 2.1.1, часть первая «Компонент jaDog. Управление режимом работы узлов кластера»);
- настроенная аутентификация пользователей в файле pg_hba.conf (см. п.п 2.1.2, часть первая «Компонент jaDog. Управление режимом работы узлов кластера»);
- установленный пароль системного пользователя;
- установленное расширение «jadog» для СУБД (см. п.п 2.1.4, часть первая «Компонент jaDog. Управление режимом работы узлов кластера»);
- созданный и настроенный пользователь jadog_user (см. п.п 2.1.5, часть первая «Компонент jaDog. Управление режимом работы узлов кластера»);
- настроены и выпущены сертификаты SSL для всех узлов кластера, для которых будет далее выполнена настройка перекрестной репликации.


Установка и настройка узлов кластера приведена в первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» в разделах 2, 3 и 6.

2.1. Формирование файла ответов для кластера с перекрестной репликацией

Шаблон файла ответов автоматизированного построения кластеров с перекрестной репликацией содержит в себе все необходимые настройки. Далее готовый файл ответов будет называться «`jadog_cross_cluster.yml`».

-  При редактировании шаблонов файлов ответов последующие уровни параметров в иерархии отделяются от предыдущих двумя пробелами.
-  Шаблон файла ответов автоматизированного построения кластеров с перекрестной репликацией «`jadog_cross_cluster.yml`» формируется на основании шаблона, расположенного в директории `/usr/jatoba-6/share/doc/jadog/clusters_kits/jadog_referee/init_jadog_referee.yml.in`

В файле ответов «`jadog_cross_cluster.yml`» для автоматизированного построения кластеров с каскадной репликацией для каждого параметра приводятся комментарии, описывающие его назначение.

-  В случае, если для параметра `slot_name` не определено значение, то оно генерируется автоматически по формуле:

```
[rs_]+[hostname]+[_]+[datetime]
```

Где, `rs_` - префикс (от сокр. replication slot), `hostname` – короткое название узла без DNS-записи, `datetime` – только локальные дата и время в формате `ddmmyyhhmmss`.

Например:

```
rs_host123_2590925070015
```

Если в имени узла содержатся буквы в верхнем регистре, дефис (-) или точки, то:

- верхний регистр – будет переведен в нижний.
- дефисы и точки – удалены из полученного `hostname`.

Существующий слот репликации во время работы не переназначается.

2.2. Запуск настройки кластера с перекрестной репликацией

Формирование и настройка узлов кластера с перекрестной репликацией выполняется при помощи специального «нулевого» режима (`jadog0`).

Предварительно «нулевой» режим (jalog0) должен быть обязательно запущен на всех узлах перед выполнением формирования кластера из файла ответов.

Запуск специального «нулевого» режима (jalog0) в режиме упрощенной небезопасной аутентификации (подробнее описание и требования в п. 3.5. Режим упрощенной небезопасной аутентификации в первой части документа «Компонент jaDog. Управление режимом работы узлов кластер» 643.72410666.00067-07 98 02-01) и создание временного пользователя выполняется с использованием ключа --basic при помощи команды:

```
/usr/jatoba-6/bin/jalog jalog0 --basic -U [temp_user] -W  
[temp_password]
```

Где temp_user – имя создаваемого временного пользователя; temp_password – пароль временного пользователя, состоящий из не менее чем шести символов и одной цифры.

Сервис jalog0 при запуске по умолчанию задействует сетевой порт TCP 64321. В случае необходимости запуска сервиса «jalog0» на другом сетевом порту необходимо использовать ключ [-p] | [--port] [номер_порта].

Примеры запуска сервиса «jalog0» с указанием сетевого порта TCP не по умолчанию:

```
/usr/jatoba-6/bin/jalog jalog0 --basic -U [temp_user] -W  
[temp_password] --port [номер_порта]
```

или

```
/usr/jatoba-6/bin/jalog jalog0 --basic -U [temp_user] -W  
[temp_password] -p [номер_порта]
```

Пример:

```
/usr/jatoba-6/bin/jalog jalog0 --basic -U tempInstallUser -W  
paSSw0rd! -p 4444
```

При запуске сервиса «jalog0» выполняется проверка по следующим параметрам:

- Указываемый сетевой порт имеет допустимый диапазон (0-65535);
- Указываемый сетевой порт не занят другими сервисами и службами;
- Количество настраиваемых сетевых портов не более одного;
- Количество задействованных ключей не более одного.

Если указанные выше условия запуска не удовлетворяются в запуске сервиса «jalog0» будет отказано до устранения проблем.

Сетевой порт, который указывает пользователь (см. пример выше) для запуска специального «нулевого» режима jalog0, может использоваться на всех узлах кластера или только на некоторых:

- В первом случае в секции default_node_params необходимо указать следующие параметры:

```
default_node_params:  
  jalog0:  
    port:4444
```

- Во втором случае в секции nodes файла ответов для каждого узла необходимо указать следующие параметры:

```
nodes:  
  - node_name: node_11  
    jalog0:  
      port: 4444  
    ...  
  - node_name: node_12  
    jalog0:  
      port: 4445
```

Выполнение сервиса «jalog0» с указанием номера сетевого порта, отличного от значения по умолчанию применяется при использовании перекрестной репликации между узлами кластера.

После этого на узле «Master» открыть дополнительное окно терминала и выполнить чтение параметров из файла ответов «jadog_cross_cluster.yml» и развертывание узлов кластеров с перекрестной репликацией:

```
/usr/jatoba-6/bin/jadog_ctl create_cluster -q -f json -T  
999999 -c /<dir1>/jadog_cross_cluster.yml --basic -U  
[temp_user]-W [temp_password]
```

Где <dir1> - путь к каталогу, в котором расположен файл ответов в формате YML
temp_user – имя временного пользователя (см. п. 3.5.1); temp_password – пароль временного пользователя.



В случае возникновения ошибок при автоматической настройке кластера производится откат всех изменений. Событие при этом записывается в журнал компонента «jaDog» с перечнем причин остановки.

В случае возникновения ошибок при автоматической настройке нескольких кластеров откат изменений производится только для кластера, в процессе создания которого произошла ошибка.

В обоих случаях необходимо повторно запустить процедуру чтения параметров из файла конфигурации и развертывание узлов кластера(ов).

В случае если пользователю необходимо создать несколько кластеров их настройки последовательно указываются в файле YML. Общая структура файла будет иметь следующий вид:

```
cluster_settings:  
- cluster_name: myCluster1 #Параметры кластера №1  
  ...  
- cluster_name: myCluster2 #Параметры кластера №2  
  ...
```

Системные события в процессе автоматической настройки кластера записываются в журнал, который расположен по следующему пути:

```
/usr/jatoba-6/var/log/jadog/jadog0_<номер_порта>.json
```

Журнал событий автоматической настройки кластера представлен в формате JSON для удобства контроля формирования кластера администратором.

Также процесс формирования и настройки кластера возможно контролировать при помощи команды:

```
cluster status
```

После завершения формирования кластера на основании файла ответов jadog_cross_cluster.yml сервис «jadog0» завершает свою работу.

2.3. Настройка компонента «jaDog» для перекрёстной репликации на резервном узле (втором экземпляре СУБД) с помощью файла ответов

Другим способом настройки компонента «jaDog» на резервном узле (втором экземпляре СУБД) является использование настроенного файла конфигурации «jadog.yml» главного узла (первого экземпляра СУБД).

Для запуска настройки компонента «jaDog» второго экземпляра СУБД необходимо выполнить команду:

```
./jadog setup -C usr/jatoba-6/etc/jadog/
```

Далее, в пункте меню 2 «Inter-jadog communication settings» в параметре «Jadog service name (main:service_name)» указать название сервиса второго экземпляра компонента «jaDog» - [jadog2]. Остальные настройки второго экземпляра компонента «jaDog» настраиваются в соответствии с требованиями к нему.

Альтернативным вариантом выполнения установки узла кластера в тихом режиме является выполнение следующей последовательности действий:

— в шаблоне установочного файла ответов «jadog_setup_faq2.yml» внести значения параметров, например:

```
main:
  service_name: jadog2
  ip: 10.96.1.139/24
  public_address: 10.96.1.140/24
  network_interface: ens33
  interconnect_user: admin
```



```
jadog_users:
  - name: jadog_user
    password: [указать_пароль_jadog_user]
    address: 10.96.1.0/24
    method: sha-256
replication:
  slot_name: jadog_slot_2
db_connection:
  auth_method: password
db_connection_settings:
  passfile: /usr/jatoba-6/bin/.pgpass
  user_pass: [указать_пароль] # пароль пользователя СУБД
postgresql:
  db_service_name: jatoba-6
```



Шаблон для установочного файла ответов «jadog_setup_faq2.yml» располагается по пути /usr/jatoba-6/share/doc/jadog/clusters_kits/standalone/jadog_setup_faq.yml

Описание параметров приведено в п.6.7 и в Приложении 1 первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01.

— Сохранить и разместить установочный файл ответов «jadog_setup_faq2.yml» в директории /usr/jatoba-6/bin/;

— Запустить установку (настройку) jadog второго экземпляра СУБД в тихом режиме выполнив команды в терминале ОС:

```
# cd /usr/jatoba-6/bin/
# ./jadog setup -c jadog_setup_faq2.yml
```

— Проверить статус сервиса:

```
# systemctl status jadog2
```

Сервис «jadog2» второго экземпляра СУБД компонента «jaDog» будет загружен, но не активирован и не запущен.

— Добавить в автозагрузку ОС и запустить сервис «jadog2» командами в терминале ОС:

```
# systemctl enable jadog2  
# systemctl start jadog2
```

Дальнейшую настройку второго экземпляра компонента «jaDog» возможно выполнить при помощи редактирования параметров кластера, описанных в п.п. 7.18 первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01.

2.4. Настройка параметра dc_public_address



Дальнейшее описание настройки параметра dc_public_address справедливо в том числе для автоматизированной настройки кластеров с другими типами репликации.

В случае автоматизированной настройки кластера при помощи файла ответов и при использовании в кластере нескольких дата-центров пользователю необходимо определить параметр dc_public_address.

В параметре dc_public_address указывают IP-адрес, через который будет происходить доступ пользователей к СУБД.

Для каждого дата-центра dc_public_address определяется в отдельности. В случае если для двух и более кластеров необходимо использовать один и тот же dc_public_address его рекомендуется определять через параметр public_address секции cluster_settings:default_node_params в файле ответов.

Параметр dc_public_address настраивается в секции datacenters:datacenter (IP-адреса приводятся в листинге в качестве примера, секции настройки нод скрыты):


```
datacenters:  
  - datacenter: DC1    # Уникальное наименование дата-центра.  
    Дата-центр может быть один или несколько.  
    dc_public_address: 10.116.102.80/32 # public_address по  
    умолчанию для всего DC. Если public_address явно не задан для  
    ноды, то этот адрес будет установлен для каждой ноды в DC.  
    [Блок описания нод кластера DC1]
```

```
- datacenter: DC2    # Уникальное наименование дата-центра.  
Дата-центр может быть один или несколько.  
    dc_public_address: 10.116.102.81/32 # public_address по  
умолчанию для всего DC. Если public_address явно не задан для  
ноды, то этот адрес будет установлен для каждой ноды в DC.  
[Блок описания нод кластера DC2]
```


При формировании файла ответов в нем необходимо учитывать приоритет установки параметров IP-адреса, через который будет происходить доступ пользователей к СУБД, в следующей последовательности:

- Параметр `public_address` в секции `datacenters:nodes:node_name` – указывается для каждого конкретного узла и имеет наивысший приоритет в файле ответов;
- Параметр `public_address` в секции `cluster_settings:default_node_params` – в этом случае применяется для всех узлов всех дата-центров, входящих в кластер;
- Параметр `dc_public_address` в секции `datacenters:datacenter` – в этом случае применяется для всех узлов дата-центра (значение `dc_public_address` заменяет собой `public_address` в настройках узлов).

Параметры могут быть установлены одновременно, но использоваться будет тот параметр, который имеет высший приоритет.

-  В случае если параметр `public` (или `dc_public_address`) не настроен в файле тихой инсталляции, то выполнение автоматизированного развертывания кластера будет прервано с отображением ошибки.

2.5. Настройка параметров архивирования и восстановления WAL

-  Дальнейшее описание настройки параметров архивирования и восстановления файлов WAL (write ahead log) справедливо в том числе для автоматизированной настройки кластеров с другими типами репликации

Секция «`wal_archive`» - настройки архивирования и восстановления WAL, предназначена для установки параметров и команд резервного копирования.

При первоначальной настройке кластера включение механизма резервного копирования WAL необязательно.

Описание и рекомендуемые значения по умолчанию параметров секции «wal_archive» приведено в п.п. 6.7.8 первой части «Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog» 643.72410666.00067-07 98 02-01.

Для настройки параметров механизма резервного копирования и восстановления WAL в файл ответов необходимо включить секции «wal_archive» и «restore» следующего содержания (здесь и далее пути к каталогам хранения архивных копий WAL приводятся в качестве примеров):

```
wal_archive:
  directory: "/nfs/archive_wal" # Системный каталог для
архивирования/восстановления WAL
  command: "cp %p /nfs/archive_wal/%f" # Команда
копирования архива WAL
  cleanup_command: "/usr/jatoba-6/bin/pg_archivecleanup
/nfs/archive_wal" # Команда очистки архива WAL
  cleanup_needed: false # Признак необходимости запуска
процесса очистки архива WAL
  cleanup_timeout: 30000 # Временной промежуток очистки
архива WAL в миллисекундах
wal_restore:
  command: "cp /nfs/arhive_wal/%f %p"
```



Необходимо указывать действительные пути к каталогам хранения архивных копий файлов WAL. В случае указания ошибок в пути к каталогам архивных копий файлов WAL возможно возникновение ошибок при проведении процедур восстановления данных СУБД.

Секции «wal_archive» и «restore» возможно расположить в двух предопределенных местах:

— Секция cluster_settings:default_node_params – настройки в данной секции распространяются на кластер и определяются для всех узлов, входящих в него;

— Секция `cluster_settings:datacenters:` - `[datacenter_name]:nodes` – настройки в данной секции распространяются только на те узлы, для которых это определено. Например, секция `nodes` может выглядеть следующим образом:

```
datacenters:
- datacenter:
  nodes:
    - node_name: node_11
      main:
        ip: 192.168.72.141
        replication:
          slot_name: rs_node_11
    - node_name: node_12
      main:
        ip: 192.168.72.142
        replication:
          slot_name: rs_node_12
      wal_archive:
        directory: "/nfs/archive_wal"
        command: "cp %p /nfs/archive_wal/%f"
        cleanup_command: "/usr/jatoba-
6/bin/pg_archivecleanup /nfs/archive_wal"
        cleanup_needed: false
        cleanup_timeout: 30000
      wal_restore:
        command: "cp /nfs/arhive_wal/%f %p"
```



В качестве значений параметров `wal_archive:command` и `restore:wal_restore_command` также могут указываться сторонние внешние программы по архивированию и восстановлению. Настройка параметров архивации и восстановления сторонних внешних программ выполняется согласно руководствам к ним.

3. ПЕРЕКРЕСТНАЯ РЕПЛИКАЦИЯ. НАСТРОЙКА В РУЧНОМ РЕЖИМЕ

Схема установки нескольких экземпляров СУБД представлена на рисунке 3.1.

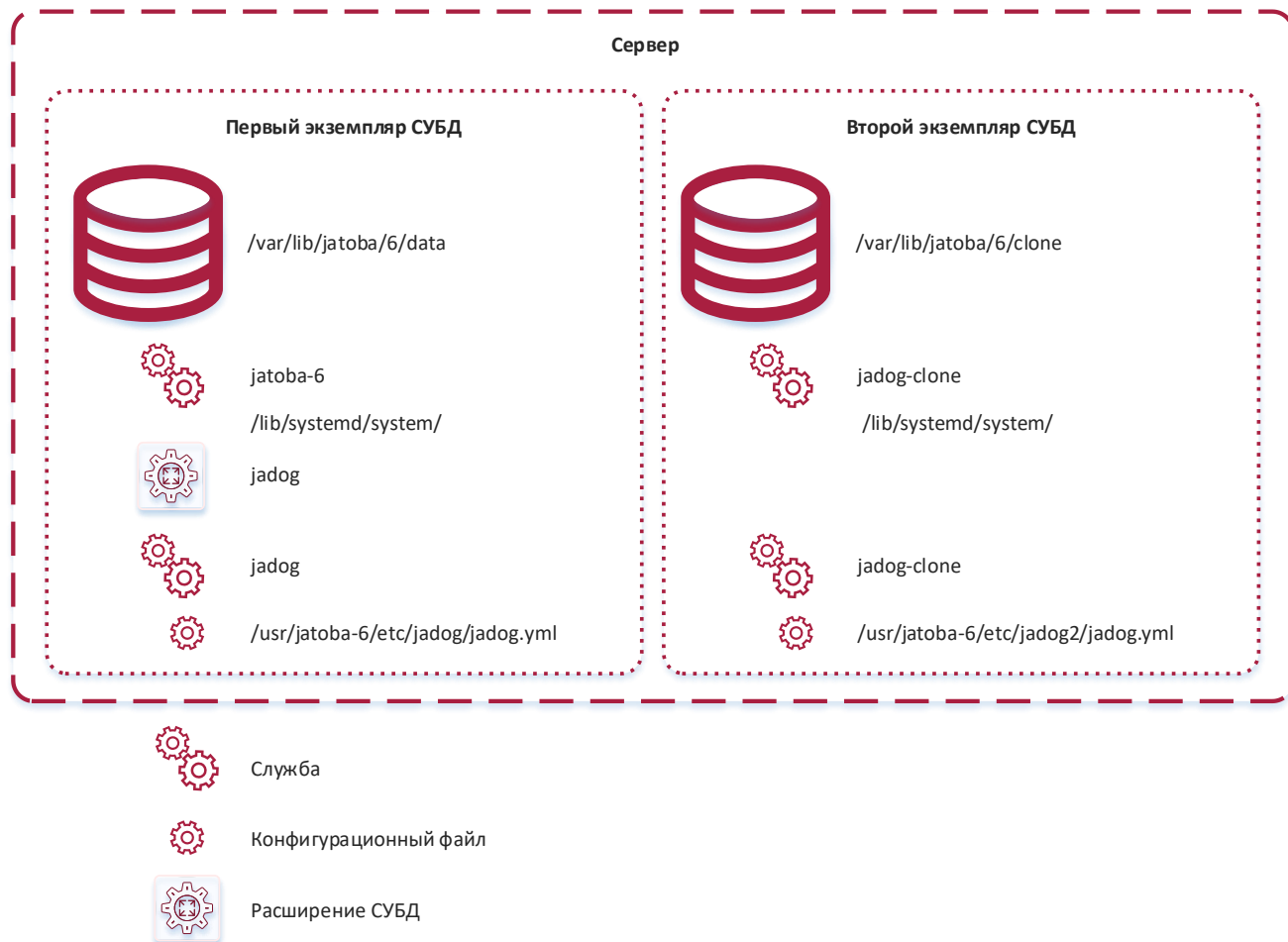


Рисунок 3.1 – Схема установки двух экземпляров СУБД на одном сервере
Ход установки описан ниже.

3.1. Установка первого экземпляра СУБД

Первый экземпляр СУБД устанавливается из локального репозитория, как описано в документе «Руководство по установке СУБД Jatoba 6» 643.72410666.00067-07 97 01.

Также для установки первого экземпляра СУБД с ролью «Master» можно воспользоваться инсталлятором, работа которого также описана в документе «Руководство по установке СУБД Jatoba 6» 643.72410666.00067-07 97 01.

После выполнения установки СУБД требуется:

– установить пакет компонента «jaDog»:

```
# apt-get install jatoba6-jadog
```

– установить пароль для пользователя СУБД «postgres»:

```
# su -l postgres
psql
\password
```

3.2. Создание второго экземпляра СУБД

Создание второго экземпляра СУБД требует выполнения следующих шагов:

– создать ссылку для запуска командой в консоли ОС:

```
# systemctl enable jatoba-6
```

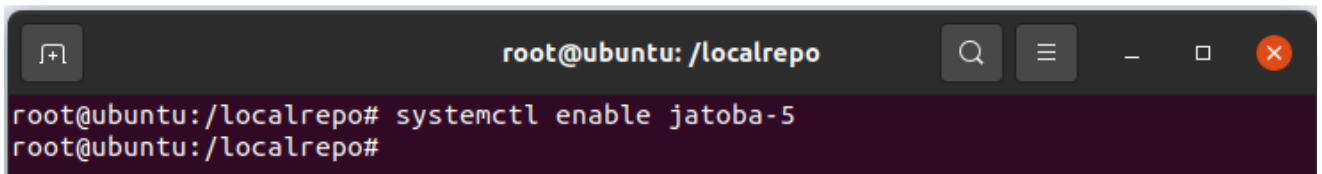


Рисунок 3.2 – Создание ссылки для запуска

– скопировать файл сервиса командой в консоли ОС:

```
# cp /lib/systemd/system/jatoba-6.service
/lib/systemd/system/jatoba-clone.service
```

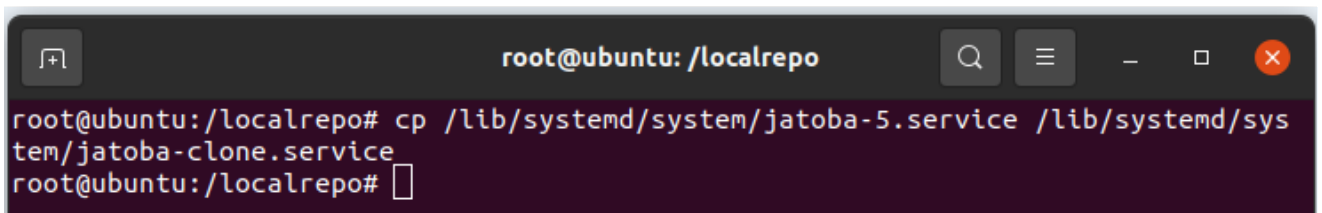


Рисунок 3.3 – Копирование файла сервиса СУБД

В результате в каталоге /lib/systemd/system/ появится файл «jatoba-clone.service» для второго экземпляра СУБД.

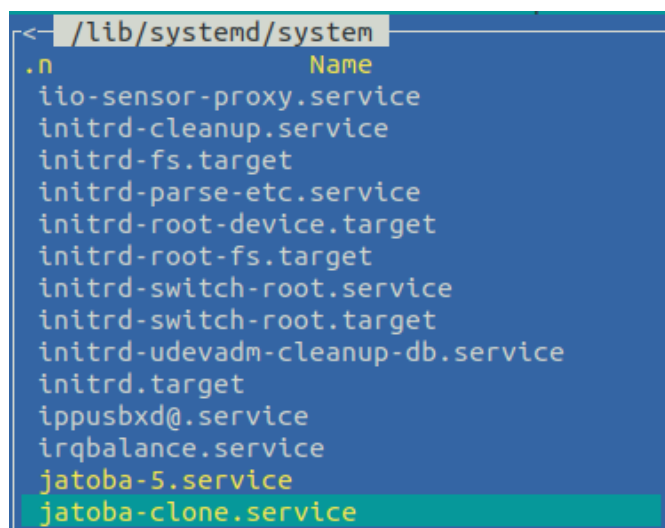


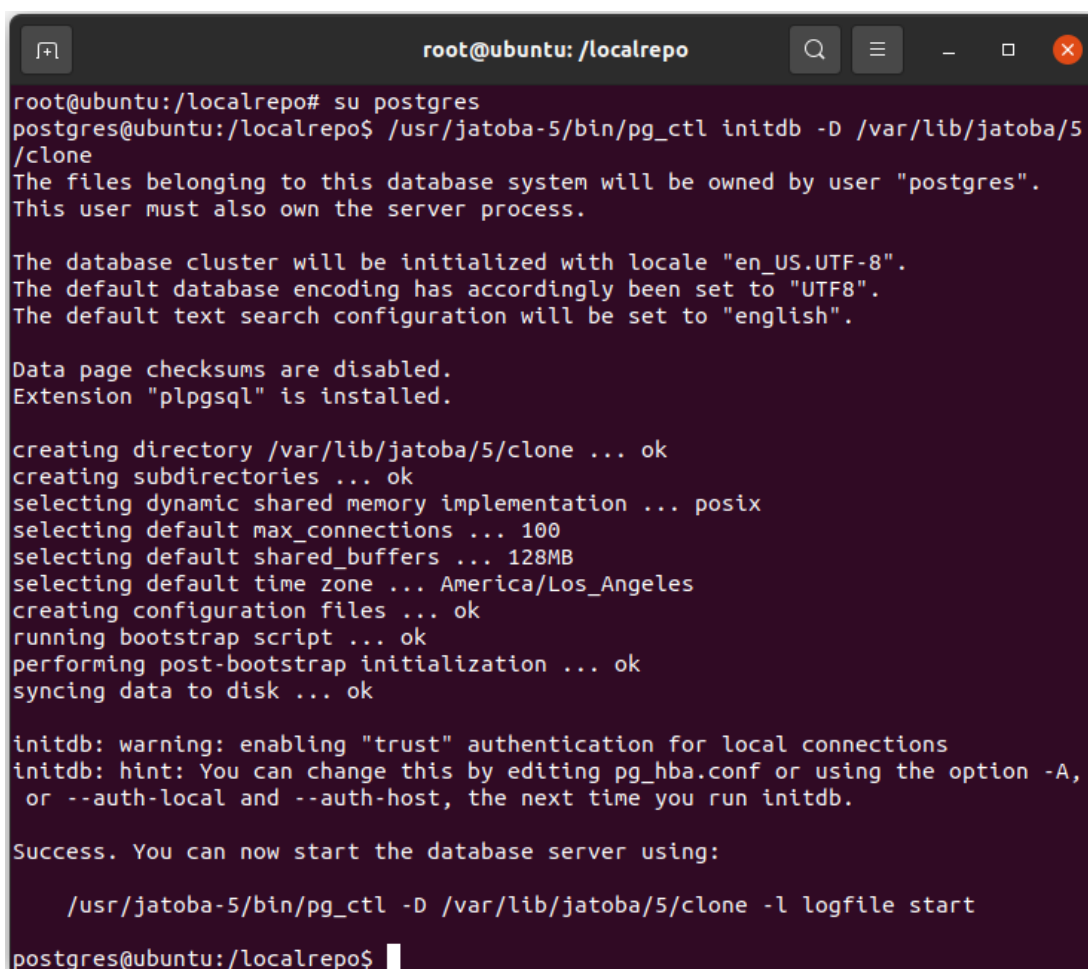
Рисунок 3.4 – Содержание каталога /lib/systemd/system/

– выполнить смену пользователя на postgres:

```
# su postgres
```

– инициализировать новый каталог данных командой в консоли ОС:

```
/usr/jatoba-6/bin/pg_ctl initdb -D /var/lib/jatoba/6/clone
```


A terminal window titled 'root@ubuntu: /localrepo' with standard Ubuntu window controls. The terminal shows the execution of 'su postgres' and then '/usr/jatoba-5/bin/pg_ctl initdb -D /var/lib/jatoba/5/clone'. It displays various initialization messages including locale settings, encoding, and configuration file creation. The process concludes with a success message and a command to start the database server.

```
root@ubuntu:/localrepo# su postgres
postgres@ubuntu:/localrepo$ /usr/jatoba-5/bin/pg_ctl initdb -D /var/lib/jatoba/5/clone
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.
Extension "plpgsql" is installed.

creating directory /var/lib/jatoba/5/clone ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... America/Los_Angeles
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A,
or --auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    /usr/jatoba-5/bin/pg_ctl -D /var/lib/jatoba/5/clone -l logfile start

postgres@ubuntu:/localrepo$
```

Рисунок 3.5 – Ход инициализации нового каталога СУБД

В результате будет создан каталог данных по пути /var/lib/jatoba/6/clone.

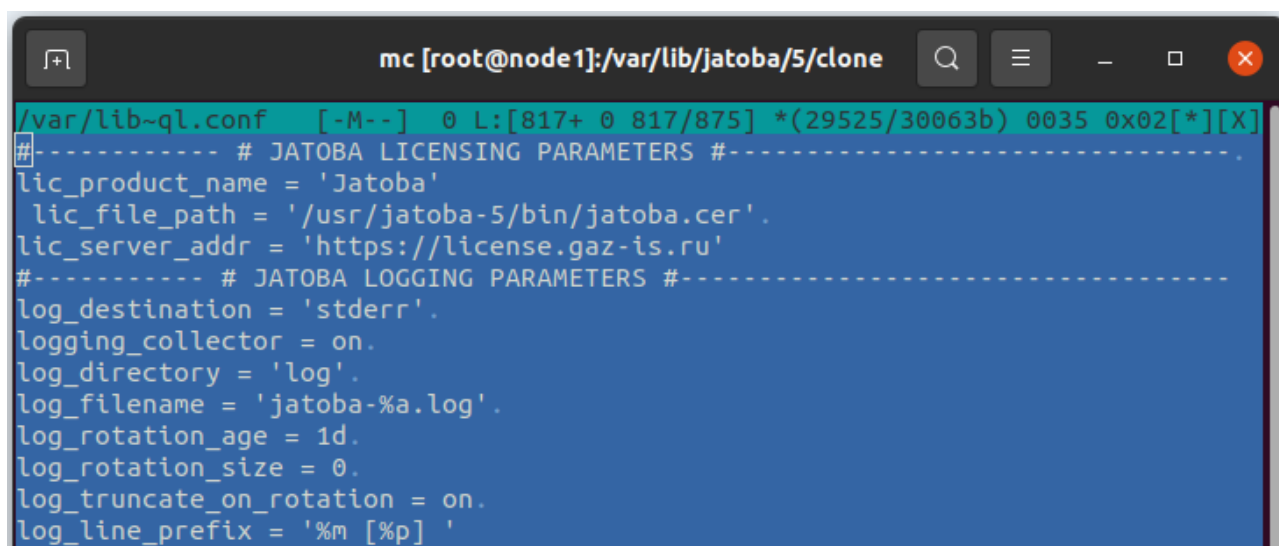
Структура каталога представлена на рисунке 3.6.



Рисунок 3.6 – Структура каталога

– в данном каталоге /var/lib/jatoba/6/clone/ дополнить параметрами конфигурационный файл «postgresql.conf»:

```
log_destination = 'stderr'
logging_collector = on
log_directory = 'log'
log_filename = 'jatoba-%a.log'
log_rotation_age = 1d
log_rotation_size = 0
log_truncate_on_rotation = on
log_line_prefix = '%m [%p] '
```



```
mc [root@node1]:/var/lib/jatoba/5/clone
/var/lib~ql.conf [-M--] 0 L:[817+ 0 817/875] *(29525/30063b) 0035 0x02[*][X]
#----- # JATOBA LICENSING PARAMETERS #-----
lic_product_name = 'Jatoba'
lic_file_path = '/usr/jatoba-5/bin/jatoba.cer'.
lic_server_addr = 'https://license.gaz-is.ru'
#----- # JATOBA LOGGING PARAMETERS #-----
log_destination = 'stderr'.
logging_collector = on.
log_directory = 'log'.
log_filename = 'jatoba-%a.log'.
log_rotation_age = 1d.
log_rotation_size = 0.
log_truncate_on_rotation = on.
log_line_prefix = '%m [%p] '
```

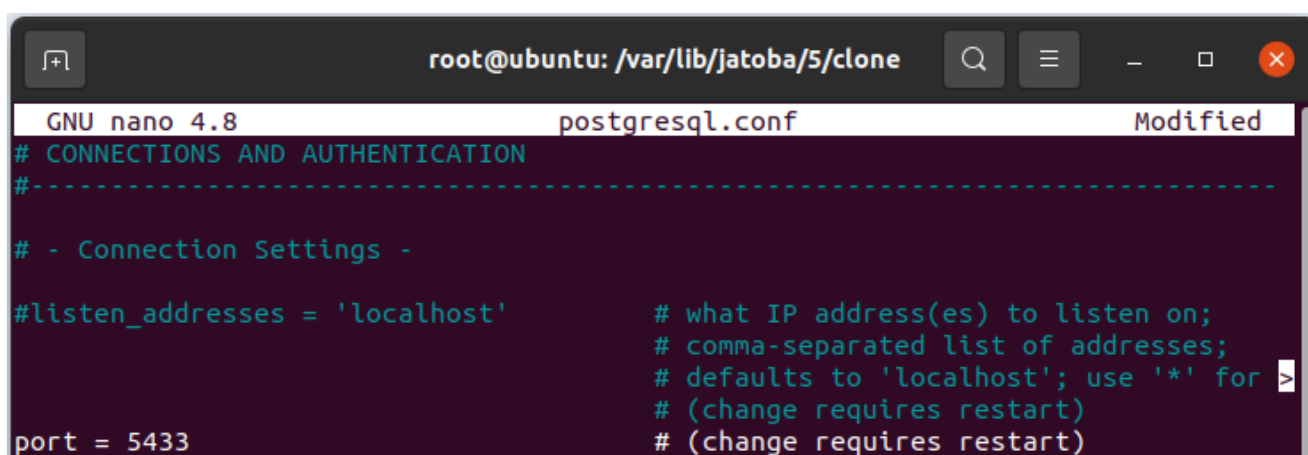
Рисунок 3.7 – Параметры конфигурационного файла «postgresql.conf» второго экземпляра СУБД



Параметры логирования приведены в качестве примера и должны отличаться от требований, предъявляемых к СУБД в зависимости от типа ИС.

– в разделе «Connection Settings» конфигурационного файл «postgresql.conf» установить параметр:

```
port = 5433
```



```
root@ubuntu: /var/lib/jatoba/5/clone
GNU nano 4.8 postgresql.conf Modified
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
                                         # comma-separated list of addresses;
                                         # defaults to 'localhost'; use '*' for
                                         # (change requires restart)
port = 5433                             # (change requires restart)
```

Рисунок 3.8 – Параметр «port» второго экземпляра СУБД

– проверить и установить метод аутентификации scram-sha-256 в конфигурационном файле «pg_hba.conf», расположенном по пути /var/lib/jatoba/6/clone/;

– изменить расположение БД в копии сервиса командой:

```
# nano /lib/systemd/system/jatoba-clone.service
```

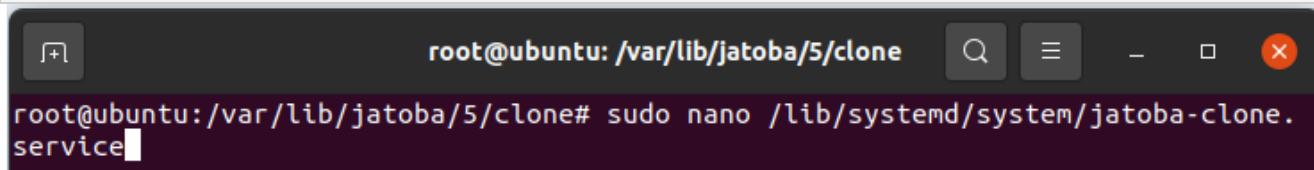


Рисунок 3.9 – Редактирование файла сервиса

– внести в разделе «Location of database directory» строку:

```
Environment=PGDATA=/var/lib/jatoba/6/clone/
```

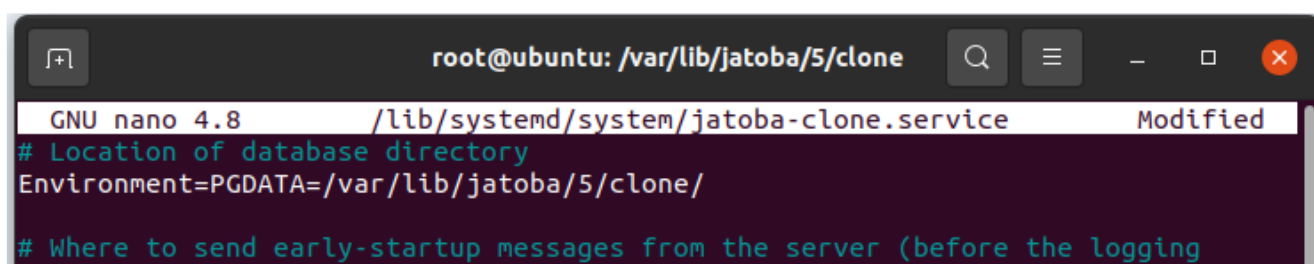


Рисунок 3.10 – Изменение строки «Environment»

– выполнить старт новой службы:

```
# systemctl start jatoba-clone
```

– проверить статус службы командой:

```
# systemctl status jatoba-clone
```

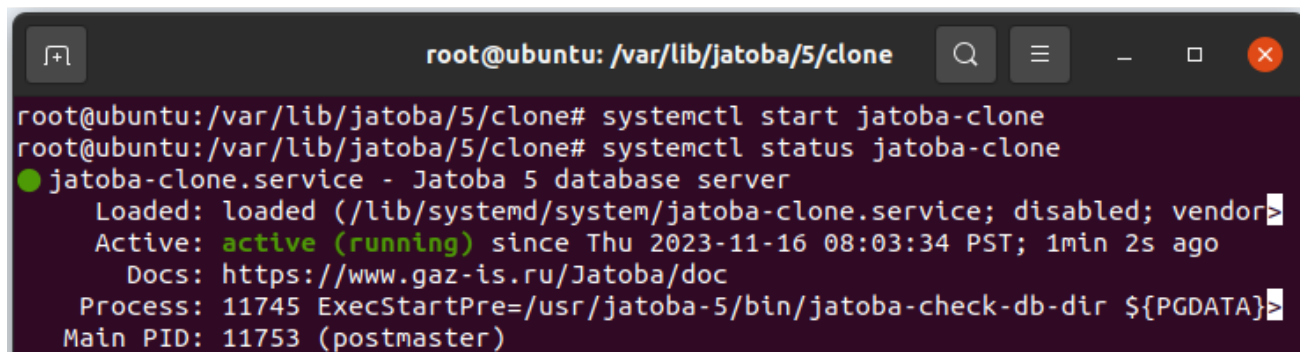
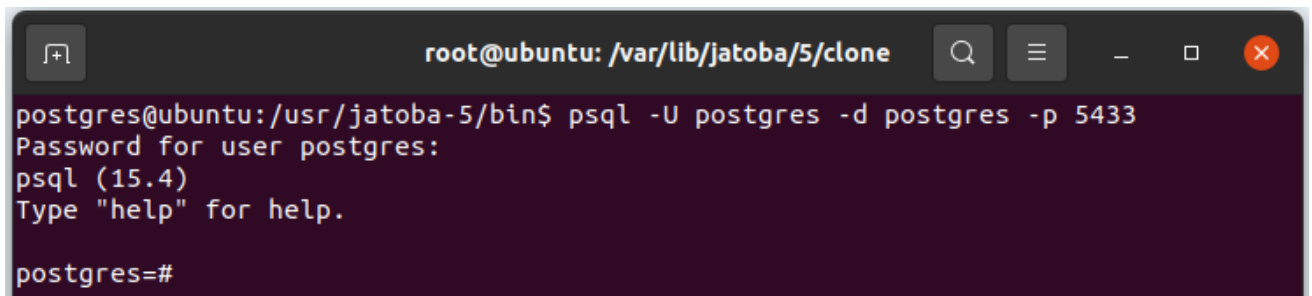


Рисунок 3.11 – Старт и проверка службы jatoba-clone

– выполнить подключение к новому экземпляру СУБД:

```
psql -U postgres -d postgres -p 5433
```



```
root@ubuntu: /var/lib/jatoba/5/clone
postgres@ubuntu:/usr/jatoba-5/bin$ psql -U postgres -d postgres -p 5433
Password for user postgres:
psql (15.4)
Type "help" for help.

postgres=#
```

Рисунок 3.12 – Вход во второй экземпляр СУБД

На данном шаге создание второго экземпляра СУБД закончено.

3.3. Установка расширения «jadow» на первом экземпляре СУБД

Установка расширения «jadow» на первой СУБД идентична описанным действиям в п. 3.3 первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01. С одной особенностью, что по значению порта надо контролировать подключение к экземпляру СУБД.

В результате потребуется:

- выполнить подключение к psql:

```
psql -U postgres -d postgres -p 5432
```

- создать расширение «jadow»:

```
CREATE EXTENSION jadow;
```

На втором экземпляре СУБД выполнять описанные действия не надо, т.к. в процессе создания кластера данные в БД синхронизируются.

3.4. Создание пользователя «jadow_user» на первом экземпляре СУБД

Установка пользователя «jadow_user» на первом экземпляре СУБД идентична описанным действиям в п. 3.4 первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01.

Пользователь создается при помощи SQL-команды с синтаксисом:

```
SELECT jadow.add_jadow_user('jadow_user');
```

Пароль для пользователя создается при помощи SQL-команды:

```
ALTER ROLE jalog_user with password '[password]';
```

3.5. Настройка jaDog для двух экземпляров СУБД

Настройка компонента «jaDog» для двух экземпляров СУБД сводится к формированию конфигурационного файла «jadog.yml» для первого экземпляра СУБД и копирования его с последующим редактированием для второго экземпляра СУБД.

Фактически компонент будет работать с двумя файлами конфигураций «jadog.yml» двух экземпляров СУБД. Конфигурационные файлы «jadog.yml» при этом будут располагаться в разных директориях.

3.5.1. Настройка компонента «jaDog» на главном узле (первом экземпляре СУБД)

Действия по настройке кластера на основной СУБД описаны в п.п 6.7 документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01.

Запуск настройки «jaDog» для первого экземпляра СУБД выполняется командами:

```
# cd /usr/jatoba-6/bin  
# ./jadog setup
```

Устанавливаемые параметры на главном узле (первом экземпляре СУБД) кластера приведены в таблице 3.1.

Таблица 3.1 – Перечень устанавливаемых параметров на главном узле (первом экземпляре СУБД)

| № | Меню/Пункт меню | Параметры |
|----------|--|-------------------------------------|
| 1 | Database server and jadog directory settings Menu | |
| 1 | Database server binaries (path:db_bin_path) | [/usr/jatoba-6/bin] |
| 2 | Database server data (postgresql:db_data_path) | [/var/lib/jatoba/6/data] |
| 3 | Jadog configs (path:config_path) | [/usr/jatoba-6/etc/jadog] |
| 4 | Jadog module files (path:module_path) | [/usr/jatoba-6/share/jadog/scripts] |
| 5 | Jadog state (path:state_path) | [/usr/jatoba-6/etc/jadog] |
| 2 | Inter-jadog communication settings | |
| 1 | Jadog service name (main:service_name) | [jadog] |
| 2 | Jadog IP address (main:ip) | Текущий IP (10.96.1.139/24) |
| 3 | Jadog PORT number (main:port) | [12345] |
| 4 | SSL on (tls:tls) | [false] |
| 5 | Jadog interconnection user (main:interconnect_user) | [admin] |
| 3 | User / Admin access network settings | |
| 1 | Public address (main:public_address) | Внешний IP адрес (10.96.1.140/24) |
| 2 | Public address control interval (main:public_address_control_interval) | [5000] |
| 3 | Public address control attempts (main:public_address_control_attempts) | [3] |
| 4 | DB monitor timeout (postgresql:db_check_interval) | [5000] |
| 5 | Trusted IP address (main:trusted_address) | [] |
| 6 | RDBMS trusted ip (ja_hipe_cluster:rdbms_trusted_ip) | [] |
| 7 | RDBMS trusted port (ja_hipe_cluster:rdbms_trusted_port) | [] |
| 8 | Network interface name (main:network_interface) | Имя сетевого интерфейса (ens33) |
| 9 | Jadog TCP port (main:user_interface_port) | [54321] |

| № | Меню/Пункт меню | Параметры |
|----------|--|-------------------------------|
| 4 | Administrator account list | |
| 1 | Add new account to the list | admin |
| 2 | Delete account from the list | |
| 3 | Clear all accounts | |
| 5 | Database server system account and connection settings | |
| 1 | Database host (db_connection_settings:db_host) | [127.0.0.1] |
| 2 | Database port (db_connection_settings:db_port) | [5432] |
| 3 | Database service name (db_connection_settings:db_service_name) | [jatoba-6] |
| 4 | Database name (db_connection_settings:db_name) | [postgres] |
| 5 | Jadog password file (db_connection_settings:db_passfile) | /usr/jatoba-6/bin/db_passfile |
| 6 | Database auth method (db_connection:auth_method) | [password] |
| 7 | Jadog to database CA file (db_connection_settings:ssl_ca_file) | [] |
| 8 | Jadog to database CRL file (db_connection_settings:ssl_crl_file) | [] |
| 9 | Jadog to database cert file (db_connection_settings:ssl_cert_file) | [] |
| 10 | Jadog to database key file (db_connection_settings:ssl_key_file) | [] |
| 11 | Jadog to database SSL mode (db_connection_settings:ssl_mode) | [very-full] |
| 12 | Jadog database user name (db_connection_settings:db_jadog_user) | [jadog_user] |
| 13 | Jadog database user password (db_connection_settings:db_jadog_user_pass) | |
| 14 | Database server OS user (system:system_user) | [postgres] |
| 15 | Replication slot name (replication:slot_name) | [jadog_slot] |
| 16 | Replication Slot TTL (replication:slot_ttl) | [300000] |
| 6 | «Failover setting» | |
| 1 | Auto failover mode [true/false] (cluster_behavior:autofailover_mode) | [true] |
| 2 | Auto dc promote mode [true/false] (cluster_behavior:dc_autofailover_mode) | [false] |

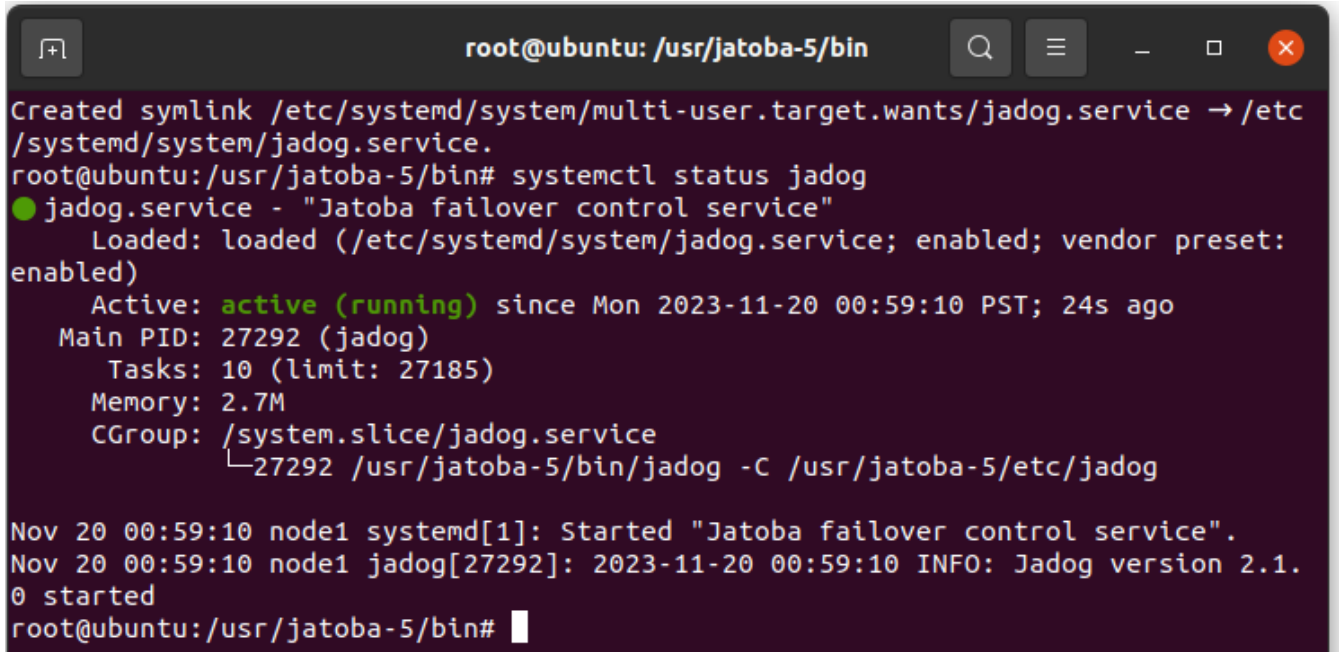
| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | Меню/Пункт меню | Параметры |
|----------|--|-------------------------------|
| 7 | «Replication setting» | |
| 1 | Replication node name (replication:slot_name) | [jadog_slot] |
| 2 | Synchronous commit setting (synchronous:synchronous_commit) | remote_apply |
| 3 | Synchronous type (synchronous:synchronous_type) | ANY |
| 4 | The number of synchronous standbys (synchronous:synchronous_commit_max_nodes) | [3] |
| 5 | The wal_level value (postgresql:wal_level) | [replica] |
| 6 | List of ignored replication slots (replication:ignore_replication_slots) | [] |
| 8 | WAL archive and restore settings | |
| 1 | System directory for WAL archive/restore (wal_archive:directory) | [] |
| 2 | WAL archive copy command (wal_archive:command) | [] |
| 3 | WAL archive restore command (wal_restore:command) | [] |
| 4 | WAL archive cleanup process on (wal_archive:cleanup_needed) | [false] |
| 5 | WAL archive cleanup command (wal_archive:cleanup_command) | [] |
| 6 | WAL archive cleanup timeout (wal_archive:cleanup_timeout) | [30000] |
| 9 | Reporting and logging | |
| 1 | Log directory (log:path) | [/usr/jatoba-6/var/log/jadog] |
| 2 | Log filename (log:file_name) | [jadog] |
| 3 | Log file mode (log:mode) | [0600] |
| 4 | Log format (log:type) | [txt] |
| 5 | Max log file usage duration (log:rotation_age) | [1d] |
| 6 | Max log file size (log:rotation_size) | [10MB] |
| 7 | Truncate, not append, logs (log:truncate_on_rotation) | [false] |
| 8 | Log verbosity (log:level) | [info] |

| № | Меню/Пункт меню | Параметры |
|-----------|--|---|
| 9 | Security log directory (security_log:path) | [/usr/jatoba-6/var/log/jadog]. |
| 10 | Security log filename (security_log:file_name) | [jadog-%a] |
| 11 | Security log file mode (security_log:file_mode) | [0600] |
| 12 | Allow to write into file (log:file) | [true] |
| 13 | Allow to write into stdout (log:screen) | [false] |
| 10 | REST API settings | |
| 1 | REST API use (rest_api:api_use) | [false] |
| 2 | REST API listen address (rest_api:listen_address) | [0.0.0.0] |
| 3 | REST API listen port (rest_api:listen_port) | [54443] |
| 4 | REST API TLS server certificate (rest_api:cert_file) | [] |
| 5 | REST API TLS server private key (rest_api:key_file) | [] |
| 6 | REST API TLS CA certificate (rest_api:ca_file) | [] |
| 7 | REST API TLS server revocation list (rest_api:crl_file) | [] |
| 11 | Recovery settings | |
| 1 | Allow backup if dir exist (after_rewind_fail_backup:backup_allow) | [true] |
| 2 | Time delay (ms) of the cluster response (recovery:cluster_timeout) | [3000000] |
| 3 | backup script file (backup:script_backup) | /usr/jatoba-6/share/jadog/scripts/backup.sh |
| 12 | Reset all settings to the default values | |
| 13 | Check and show all settings | |
| 14 | Save settings and setup jadog | |

После чего требуется запустить и добавить сервис «jaDog» в автозапуск ОС командами:

```
systemctl start jadog  
systemctl enable jadog  
systemctl status jadog
```

A terminal window titled 'root@ubuntu: /usr/jatoba-5/bin' showing the execution of systemctl commands. The output indicates that the 'jadog.service' is 'active (running)' and provides details such as its PID (27292), tasks (10), memory usage (2.7M), and CGroup. Log messages at the bottom confirm the service's start and version (2.1.0).

```
Created symlink /etc/systemd/system/multi-user.target.wants/jadog.service → /etc/systemd/system/jadog.service.  
root@ubuntu:/usr/jatoba-5/bin# systemctl status jadog  
● jadog.service - "Jatoba failover control service"  
   Loaded: loaded (/etc/systemd/system/jadog.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2023-11-20 00:59:10 PST; 24s ago  
     Main PID: 27292 (jadog)  
        Tasks: 10 (limit: 27185)  
       Memory: 2.7M  
      CGroup: /system.slice/jadog.service  
              └─27292 /usr/jatoba-5/bin/jadog -C /usr/jatoba-5/etc/jadog  
  
Nov 20 00:59:10 node1 systemd[1]: Started "Jatoba failover control service".  
Nov 20 00:59:10 node1 jadog[27292]: 2023-11-20 00:59:10 INFO: Jadog version 2.1.0 started  
root@ubuntu:/usr/jatoba-5/bin#
```

Рисунок 3.13 – Запуск и проверка статуса сервиса «jadog»

3.5.2. Настройка компонента «jaDog» на резервном узле (втором экземпляре СУБД)

После формирования параметров кластера на главном узле (первом экземпляре СУБД) в каталоге /usr/jatoba-6/etc/jadog/ находится конфигурационный файл «jadog.yml».

Для подготовки резервного узла (второго экземпляра СУБД) кластера необходимо скопировать конфигурацию главного узла кластера (первого экземпляра СУБД) «jadog.yml» в каталог /usr/jatoba-6/etc/jadog2/ и внести изменения.

Копирование каталога конфигурации с вложенным файлом «jadog.yml» выполняется командой:

```
cp -r ../etc/jadog/ ../etc/jadog2
```

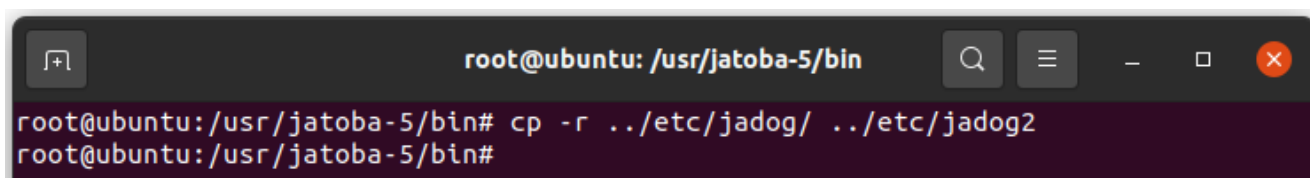


Рисунок 3.14 – Копирование каталога

Структура каталога /usr/jatoba-6/etc/ показана на рисунке 3.15. На этом этапе каталоги /usr/jatoba-6/etc/jadog/ и /usr/jatoba-6/etc/jadog2/ содержат одинаковую конфигурацию.

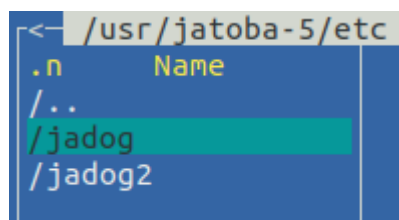


Рисунок 3.15 – Структура каталога usr/jatoba-5/etc

Файл конфигурации /usr/jatoba-6/etc/jadog2/jadog.yml для второго экземпляра СУБД требует внесения изменений. Для чего необходимо открыть файл конфигурации в редакторе командой:

```
nano /usr/jatoba-6/etc/jadog2/jadog.yml
```

Установить следующие параметры:

```
db_port: 5433
path:
  db_bin_path: /usr/jatoba-6/bin
postgresql:
  db_data_path: /var/lib/jatoba/6/clone
  db_service_name: jatoba-clone
replication:
  slot_name: jadog_slot_2
main:
  port: 2222
  user_interface_port: 54322
```

```
log:
    screen: true
    file: false
```



Значение параметра `slot_name` является уникальным для каждого экземпляра компонента «jaDog».

```
#Рабочие директории
param_path:
  state_path: /usr/jatoba-6/etc/jadog           # Путь к файлу текущего состояния
  module_path: /usr/jatoba-6/share/jadog/scripts # Путь к файлам скриптов
  log_path: /usr/jatoba-6/var/log/jadog         # Путь к каталогу логов
#Параметры подключения к базе данных
db_connection_settings:
  db_auth_method: password                     # Тип аутентификации к субд: password / ssl
#Параметры подключения jadog
param_connection:
  conn_string: host=127.0.0.1 port=5432 user=jadog_user dbname=postgres passfile=/usr/jatoba-6/bin/.pgpass
  к субд
  max_admin_connections: 10                   # Максимальное количество пользовательских подключений
#Параметры ja hipe
param_ja_hipe:
  db_name: ""                                 # Имя базы данных ja hipe
#Параметры postgres
param_postgres:
  db_fail_connection_attempts: 5               # Количество попыток подключения при ошибке
  db_check_interval: 5000                     # Интервал проверки субд
  db_port: 5433
  db_bin_path: /usr/jatoba-6/bin               # Путь до каталога с исполняемыми файлами субд
  db_data_path: /var/lib/jatoba-6/clone        # Путь до каталога данных субд
  db_config_path: /var/lib/jatoba-6/clone      # Путь к каталогу конфигурации клонированной версии
  db_service_name: jatoba-clone               # Имя сервиса субд
#Параметры репликации
param_replication:
  replication_slot_name: jadog_slot_2         # Имя слота репликации
#Пользователь системы
param_system:
  system_user: postgres                       # Имя системной учетной записи ос, владельца субд
#Параметры jadog
```

Рисунок 3.16 – Измененные параметры в конфигурационном файле `/usr/jatoba-6/etc/jadog2/jadog.yml`

3.5.3. Настройка сервиса «jadog» для резервного узла второго экземпляра СУБД

По умолчанию сервис «jadog» хранится в каталоге:

```
cd /etc/systemd/system
```

Для запуска второго сервиса «jadog» для резервного узла второго экземпляра СУБД потребуется клонировать файл сервиса «jadog» командами в ОС:

```
cp /etc/systemd/system/jadog.service
   /etc/systemd/system/jadog-clone.service
```


```
root@ubuntu: /etc/systemd/system
root@ubuntu:/etc/systemd/system# cd /etc/systemd/system
root@ubuntu:/etc/systemd/system# cp /etc/systemd/system/jadog.service /etc/systemd/system/jadog-clone.service
root@ubuntu:/etc/systemd/system#
```

Рисунок 3.17 – Выполнение клонирования файла сервиса «jadog»

В результате в каталоге /etc/systemd/system появится файл «jadog-clone.service»

<


>


 Computer

etc


systemd

system ▾







▾





▬





★ Starred

 Home

 Desktop



| Name ▾ | Size | Modified |
|---|-----------|----------|
|  jadog.service | 193 bytes | 00:11 |
|  jadog-clone.service | 193 bytes | 03:05 |

Рисунок 3.18 – Расположение файлов сервиса «jadog»

Клонированный файл сервиса «jadog-clone.service» нуждается в редактировании, чтобы при последующей автозагрузке он пользовался собственным конфигурационным файлом, расположенным в каталоге:

```
/usr/jatoba-6/etc/jadog2/jadog.yml
```

Для этого требуется открыть файл конфигурации в редакторе командой:

```
nano /etc/systemd/system/jadog-clone.service
```

Установить следующие параметры:

```
ExecStart="/usr/jatoba-6/bin/jadog" -C "/usr/jatoba-6/etc/jadog2"
```

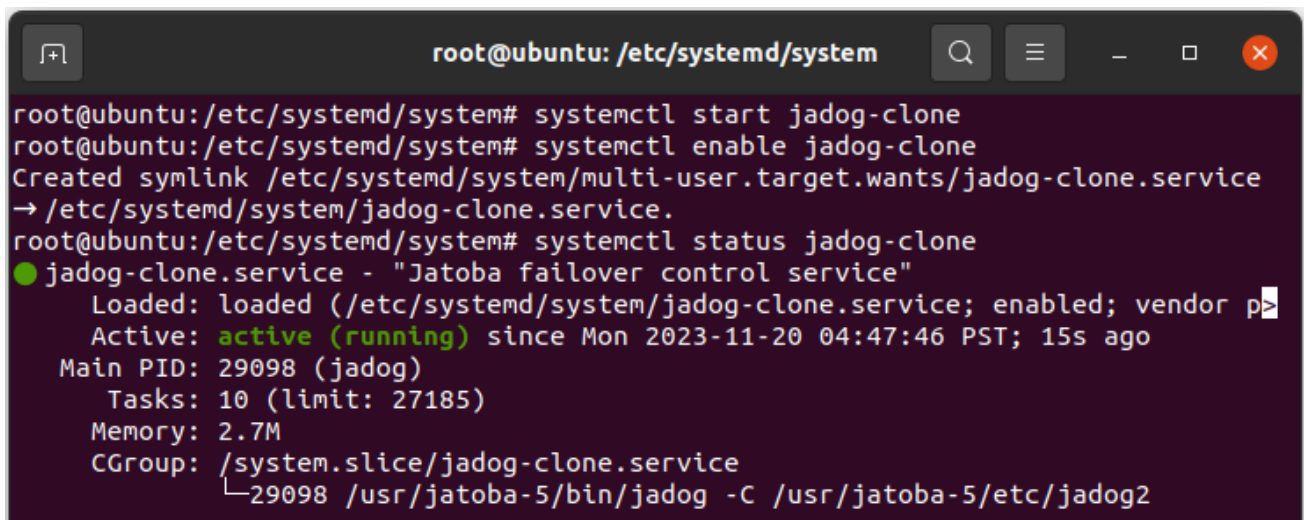
```
GNU nano 4.8 /etc/systemd/system/jadog-clone.service Modified
[Unit]
Description="Jatoba failover control service"
After=network-online.target
[Service]
ExecStart="/usr/jatoba-5/bin/jadog" -C "/usr/jatoba-5/etc/jadog2"
[Install]
WantedBy=multi-user.target
```

Рисунок 3.19 – Редактирования файла сервиса «jadog-clone.service»

Тем самым указывается директория, хранящая ранее клонированный и отредактированный файл конфигурации узла кластера.

Запуск, включение в автозагрузку ОС и проверка статуса сервиса выполняется командами:

```
systemctl start jadog-clone
systemctl enable jadog-clone
systemctl status jadog-clone
```



```
root@ubuntu: /etc/systemd/system
root@ubuntu:/etc/systemd/system# systemctl start jadog-clone
root@ubuntu:/etc/systemd/system# systemctl enable jadog-clone
Created symlink /etc/systemd/system/multi-user.target.wants/jadog-clone.service
→ /etc/systemd/system/jadog-clone.service.
root@ubuntu:/etc/systemd/system# systemctl status jadog-clone
● jadog-clone.service - "Jatoba failover control service"
   Loaded: loaded (/etc/systemd/system/jadog-clone.service; enabled; vendor p
   Active: active (running) since Mon 2023-11-20 04:47:46 PST; 15s ago
     Main PID: 29098 (jadog)
        Tasks: 10 (limit: 27185)
       Memory: 2.7M
      CGroup: /system.slice/jadog-clone.service
              └─29098 /usr/jatoba-5/bin/jadog -C /usr/jatoba-5/etc/jadog2
```

Рисунок 3.20 – Запуск и проверка статуса «jadog-clone.service»

3.6. Создание кластера

Дальнейшее конфигурирование кластера зависит от типа кластера.

Последовательная репликация описана в п. 6.9 первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01.

Перекрестная репликация описана в разделе 3 настоящего документа.

4. КАСКАДНАЯ РЕПЛИКАЦИЯ. ИСПОЛЬЗОВАНИЕ ФАЙЛОВ ОТВЕТОВ


Каскадная репликация позволяет снизить количество подключений и нагрузку на узел с ролью Master. Основную нагрузку с него снимает подчиненный резервный узел (Primary Slave) работает как получатель и отправитель. К подчиненному резервному узлу (Primary Slave) подключаются подчиненные каскадные узлы (Cascade Slave).

В данном разделе рассматривается вариант настройки кластера с каскадной репликацией в автоматизированном режиме с использованием подготовленного файла ответов. Ручная настройка кластера с каскадной репликацией приведена в разделе 5.

В рассматриваемом примере установки и конфигурировании отказоустойчивого кластера «jaDog» на ОС Ubuntu 22.04 используются параметры сети кластера, приведенные в таблице 4.1.


Таблица 4.1 – Конфигурация сети кластера для каскадной репликации

| № | Имя сервера | IP-адрес | Маска подсети | Public IP | Роль | Имя кластера |
|---|-------------|------------------|---------------|------------------|---------------|-----------------|
| 1 | node1 | 10.116.102.54/24 | 255.255.255.0 | 10.116.102.81/24 | Master | cluster_cascade |
| 2 | node2 | 10.116.102.55/24 | 255.255.255.0 | 10.116.102.81/24 | Primary Slave | |
| 4 | node3 | 10.116.102.57/24 | 255.255.255.0 | 10.116.102.81/24 | Cascade Slave | |

 Узлы кластера с каскадной репликацией должны работать в асинхронном режиме.

4.1. Формирование файла ответов для кластера с каскадной репликацией

Шаблон файла ответов автоматизированного построения кластеров с каскадной репликацией содержит в себе все необходимые настройки. Далее данный шаблон файла ответов будет называться «jadog_cascading_cluster.yml».

 Шаблон файла ответов автоматизированного построения кластеров с каскадной репликацией «jadog_cascading_cluster.yml» формируется на основании шаблона, расположенного в директории /usr/jatoba-6/share/doc/jadog/clusters_kits/jadog_referee/init_jadog_referee.yml.in

В шаблоне файла ответов «`jadog_cascading_cluster.yml`» для автоматизированного построения кластеров с перекрестной репликацией для каждого параметра приводятся комментарии, описывающие его назначение.



По умолчанию главный подчиненный узел (Primary Slave), входящий в кластер с каскадной репликацией, настраивается на асинхронный тип репликации с главным узлом с помощью параметра `replication_type: async`. Для каскадных резервных узлов (Cascade Slave) указывается как `replication_type: none`



В случае, если для параметра `slot_name` не определено значение, то оно генерируется автоматически по формуле:

```
[rs_]+[hostname]+[_]+[datetime]
```

Где, `rs_` - префикс (от сокр. replication slot), `hostname` – короткое название узла без DNS-записи, `datetime` – только локальные дата и время в формате `ddmmyyhhmmss`.

Например:

```
rs_host123_2590925070015
```

Если в имени узла содержатся буквы в верхнем регистре, дефис (-) или точки, то:

- верхний регистр – будет переведен в нижний.
- дефисы и точки – удалены из полученного `hostname`.

Существующий слот репликации во время работы не переназначается.

4.2. Запуск настройки кластера с каскадной репликацией

Формирование и настройка узлов кластера с каскадной репликацией выполняется аналогично процедуре для кластера с перекрестной репликацией (см. п. 2.2).

4.3. Настройка компонента «jaDog» для каскадной репликации на подчиненном главном узле и резервном узле с помощью файла ответов

Настройка компонента «jaDog» для каскадной репликации на подчиненном главном узле (Primary Slave) и резервном узле (Cascade Slave) с помощью файла ответов

выполняется аналогично процедуре для кластера с перекрестной репликацией (см. п.п. 2.3).

5. КАСКАДНАЯ РЕПЛИКАЦИЯ. НАСТРОЙКА В РУЧНОМ РЕЖИМЕ

В данном разделе рассматривается вариант настройки каскадной репликации с использованием сервиса «jadowg» с параметром «setup».

В рассматриваемом примере установки и конфигурировании отказоустойчивого кластера «jaDog» на ОС Ubuntu 22.04 используются параметры сети кластера, приведенные в таблице 5.1.

Таблица 5.1 – Конфигурация сети кластера для каскадной репликации

| № | Имя сервера | IP-адрес | Маска подсети | Public IP | Роль | Имя кластера |
|---|-------------|------------------|---------------|------------------|---------------|--------------|
| 1 | Node1 | 10.116.102.54/24 | 255.255.255.0 | 10.116.102.81/24 | Master | Cascade |
| 2 | Node2 | 10.116.102.55/24 | 255.255.255.0 | 10.116.102.81/24 | Slave Main | |
| 3 | Shared-node | 10.116.102.56/24 | 255.255.255.0 | | File-server | |
| 4 | Node4 | 10.116.102.57/24 | 255.255.255.0 | 10.116.102.81/24 | Slave Cascade | |
| 5 | Node5 | 10.116.102.58/24 | 255.255.255.0 | 10.116.102.81/24 | Slave Cascade | |

В работе кластера с каскадной репликацией также будет участвовать и файловый сервер с архивом WAL.



Узлы кластера с каскадной репликацией должны работать в асинхронном режиме.

5.1. Настройка компонента «jaDog» на главном узле при каскадной репликации

Настройка компонента «jaDog» на главном узле при каскадной репликации полностью описана в пунктах настоящего руководства. Должны быть выполнены шаги, описанные в пунктах с параметрами, приведенными в таблице 5.2 для Node1:

- установка пароля системного пользователя ОС «postgres»;
- установка расширения «jadowg»;
- вызов функции «add_jadowg_user» для создания пользователя «jadowg_user»;
- установка пакетов СУБД на резервном узле (Slave).

5.2. Настройка и запуск компонента «jaDog» на резервных узлах при каскадной репликации

Резервные узлы Node2, Node4 и Node5 настраиваются с параметрами, приведенными в таблице 5.2 по п. 5.6 первой части «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01.

Устанавливаемые помощи сервиса «jalog» с параметром «setup» параметры кластера с каскадной репликацией, приведены в таблице 5.2.

Таблица 5.2 – Перечень устанавливаемых параметров для каскадной репликации

| № | Меню/Пункт меню | Параметры | Node1 | Node2 | Node4 | Node5 |
|----------|--|-------------------------------------|------------------|------------------|------------------|------------------|
| 1 | Database server and jalog directory settings Menu | | | | | |
| 1 | Database server binaries (path:db_bin_path) | [/usr/jatoba-6/bin] | = | = | = | = |
| 2 | Database server data (postgresql:db_data_path) | [/var/lib/jatoba/6/data] | = | = | = | = |
| 3 | Jalog configs (path:config_path) | [/usr/jatoba-6/etc/jalog] | = | = | = | = |
| 4 | Jalog module files (path:module_path) | [/usr/jatoba-6/share/jalog/scripts] | = | = | = | = |
| 5 | Jalog state (path:state_path) | [/usr/jatoba-6/etc/jalog] | = | = | = | = |
| 2 | Inter-jalog communication settings | | | | | |
| 1 | Jalog service name (main:service_name) | | | | | |
| 2 | Jalog IP address (main:ip) | Текущий IP (10.116.102.54/24) | 10.116.102.54/24 | 10.116.102.55/24 | 10.116.102.57/24 | 10.116.102.58/24 |
| 3 | Jalog PORT number (main:port) | [12345] | = | = | = | = |
| 4 | SSL on (tls:tls) | [false] | = | = | = | = |
| 5 | Jalog interconnection user (main:interconnect_user) | [admin] | = | = | = | = |
| 3 | User / Admin access network settings | | | | | |
| 1 | Public address (main:public_address) | Внешний IP адрес (10.116.102.81/24) | 10.116.102.81/24 | 10.116.102.81/24 | 10.116.102.81/24 | 10.116.102.81/24 |
| 2 | Public address control interval (main:public_address_control_interval) | [5000] | = | = | = | = |
| 3 | Public address control attempts (main:public_address_control_attempts) | [3] | = | = | = | = |
| 4 | DB monitor timeout (postgresql:db_check_interval) | [5000] | = | = | = | = |
| 5 | Trusted IP address (main:trusted_address) | [] | [] | [] | [] | [] |
| 6 | RDBMS trusted ip (ja_hipe_cluster:rdbms_trusted_ip) | [] | [] | [] | [] | [] |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node4 | Node5 |
|----------|---|---|------------------------------------|-------|-------|-------|-------|
| | 7 | RDBMS trusted port (ja_hipe_cluster:rdbms_trusted_port) | [] | [] | [] | [] | [] |
| | 8 | Network interface name (main:network_interface) | Имя сетевого интерфейса (ens18) | = | = | = | = |
| | 9 | Jadog TCP port (main:user_interface_port) | [54321] | = | = | = | = |
| 4 | Administrator account list | | | | | | |
| | 1 | Add new account to the list | admin | admin | admin | admin | admin |
| | 2 | Delete account from the list | | | | | |
| | 3 | Clear all accounts | | | | | |
| 5 | Database server system account and connection settings | | | | | | |
| | 1 | Database host (db_connection_settings:host) | [127.0.0.1] | = | = | = | = |
| | 2 | Database port (db_connection_settings:port) | [5432] | = | = | = | = |
| | 3 | Database service name (postgresql:db_service_name) | [jatoba-6] | = | = | = | = |
| | 4 | Database name (db_connection_settings:database) | [postgres] | = | = | = | = |
| | 5 | Jadog password file (db_connection_settings:passfile) | /usr/jatoba- 6/bin/db_passfile | = | = | = | = |
| | 6 | Database auth method (db_connection:auth_method) | [password] | = | = | = | = |
| | 7 | Jadog to database CA file (db_connection_settings:ssl_ca_file) | [] | [] | [] | [] | [] |
| | 8 | Jadog to database CRL file (db_connection_settings:ssl_crl_file) | [] | [] | [] | [] | [] |
| | 9 | Jadog to database cert file (db_connection_settings:ssl_cert_file) | [] | [] | [] | [] | [] |
| | 10 | Jadog to database key file (db_connection_settings:ssl_key_file) | [] | [] | [] | [] | [] |
| | 11 | Jadog to database SSL mode (db_connection_settings:ssl_mode) | [verify-full] | = | = | = | = |
| | 12 | Jadog database user name (db_connection_settings:user) | [jadog_user] | = | = | = | = |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node4 | Node5 |
|----------|---|--|-------------------------------|--------------|--------------|--------------|--------------|
| | 13 | Jadog database user password (db_connection_settings:user_pass) | [] | = | = | = | = |
| | 14 | Database server OS user (system:system_user) | [postgres] | = | = | = | = |
| | 15 | Replication node name (replication:slot_name) | [] | node1 | node2 | node4 | node5 |
| | 16 | Replication Slot TTL (replication:slot_ttl) | [300000] | = | = | = | = |
| 6 | «Failover setting» | | | | | | |
| | 1 | Auto failover mode [true/false] (cluster_behavior:autofailover_mode) | [true] | [true] | [true] | [true] | [true] |
| | 2 | Auto dc promote mode [true/false] (cluster_behavior:dc_autofailover_mode) | [true] | [true] | [true] | [true] | [true] |
| 7 | «Replication setting» | | | | | | |
| | 1 | Replication node name (replication:slot_name) | [] | node1 | node2 | node4 | node5 |
| | 2 | Synchronous commit setting (synchronous:synchronous_commit) | remote_apply | remote_apply | remote_apply | remote_apply | remote_apply |
| | 3 | Synchronous type (synchronous:synchronous_type) | ANY | ANY | ANY | ANY | ANY |
| | 4 | The number of synchronous standbys (synchronous:synchronous_commit_max_nodes) | 4 | 4 | 4 | 4 | 4 |
| | 5 | The wal_level value (postgresql:wal_level) | [replica] | = | = | = | = |
| | 6 | List of ignored replication slots (replication:ignore_replication_slots) | [] | = | = | = | = |
| 8 | WAL archive and restore settings | | | | | | |
| | 1 | System directory for WAL archive/restore (wal_archive:directory) | [/nfs/arhive_wal] | = | = | = | = |
| | 2 | WAL archive copy command (wal_archive:command) | [cp %p /nfs/arhive_wal/%f] | = | = | = | = |
| | 3 | WAL archive restore command (wal_restore:command) | [cp /nfs/arhive_wal/%f %p] | = | = | = | = |
| | 4 | WAL archive cleanup process on (wal_archive:cleanup_needed) | [false] | = | = | = | = |

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node4 | Node5 |
|-----------|-------------------------------|--|---|-------|-------|-------|-------|
| | 5 | WAL archive cleanup command (wal_archive:cleanup_command) | [/usr/jatoba-6/bin/pg_archivecleanup /nfs/arhive_wal %s] | = | = | = | = |
| | 6 | WAL archive cleanup timeout (wal_archive:cleanup_timeout) | [30000] | = | = | = | = |
| 9 | Reporting and logging | | | | | | |
| | 1 | Log directory (log:path) | [/usr/jatoba-6/var/log/jadog] | = | = | = | = |
| | 2 | Log filename (log:file_name) | [jadog] | = | = | = | = |
| | 3 | Log file mode (log:mode) | [0600] | = | = | = | = |
| | 4 | Log format (log:type) | [txt] | = | = | = | = |
| | 5 | Max log file usage duration (log:rotation_age) | [1d] | = | = | = | = |
| | 6 | Max log file size (log:rotation_size) | [10MB] | = | = | = | = |
| | 7 | Truncate, not append, logs (log:truncate_on_rotation) | [false] | = | = | = | = |
| | 8 | Log verbosity (log:level) | [info] | = | = | = | = |
| | 9 | Security log directory (security_log:path) | [/usr/jatoba-6/var/log/jadog]. | = | = | = | = |
| | 10 | Security log filename (security_log:file_name) | [jadog-%a] | = | = | = | = |
| | 11 | Security log file mode (security_log:file_mode) | [0600] | = | = | = | = |
| | 12 | Allow to write into file (log:file) | [true] | = | = | = | = |
| | 13 | Allow to write into stdout (log:screen) | [false] | = | = | = | = |
| 10 | REST API settings Menu | | | | | | |
| | 1 | REST API use (rest_api:api_use) | [false] | = | = | = | = |
| | 2 | REST API listen address (rest_api:listen_address) | [0.0.0.0] | = | = | = | = |
| | 3 | REST API listen port (rest_api:listen_port) | [54443] | = | = | = | = |
| | 4 | REST API TLS server certificate (rest_api:cert_file) | [] | = | = | = | = |
| | 5 | REST API TLS server private key (rest_api:key_file) | [] | = | = | = | = |
| | 6 | REST API TLS CA certificate (rest_api:ca_file) | [] | = | = | = | = |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node4 | Node5 |
|-----------|---|---|-----------|-------|-------|-------|-------|
| | 7 | REST API TLS server revocation list (rest_api:crl_file) | [] | = | = | = | = |
| 11 | Recovery settings | | | | | | |
| | 1 | Allow backup if dir exist (after_rewind_fail_backup:backup_allow) | [true] | = | = | = | = |
| | 2 | Time delay (ms) of the cluster response (recovery:cluster_timeout) | [3000000] | = | = | = | = |
| | 3 | backup script file (backup:script_backup) | [] | = | = | = | = |
| 12 | Reset all settings to the default values | | | | | | |
| 13 | Check and show all settings | | | | | | |
| 14 | Save settings and setup jadog | | | | | | |

5.3. Настройка кластера для каскадной репликации

Первоначально требуется установить параметры аутентификации на сервере Master. С этой целью в конфигурационном файле `/var/lib/jatoba/6/data/pg_hba.conf` требуется внести строки с указанием подсети, в которой работает кластер.

```
host    replication    jadog_user    10.116.102.0/24    scram-sha-256
host    all            jadog_user    10.116.102.0/24    scram-sha-256
```

Тем самым, пользователю СУБД «jadog_user» разрешается подключаться к СУБД в подсети, проводить репликацию по методу аутентификации «scram-sha-256» в подсети.

The screenshot shows a terminal window with the title `root@node1: /var/lib/jatoba/18/data`. The editor is GNU nano 6.2, editing the file `pg_hba.conf`. The content of the file is as follows:

```
# TYPE      DATABASE     USER        ADDRESS            METHOD
#-----
# jaDog autogenerate parameters
#-----
local      all         postgres    scram-sha-256
local      all         jadog_user  scram-sha-256
host       all         postgres    10.116.102.0/24    trust
host       all         postgres    127.0.0.1/32       scram-sha-256
host       all         jadog_user  127.0.0.1/32       scram-sha-256
host       all         jadog_user  10.116.102.0/24    scram-sha-256
host       replication jadog_user  127.0.0.1/32       scram-sha-256
host       replication jadog_user  10.116.102.0/24    scram-sha-256
```

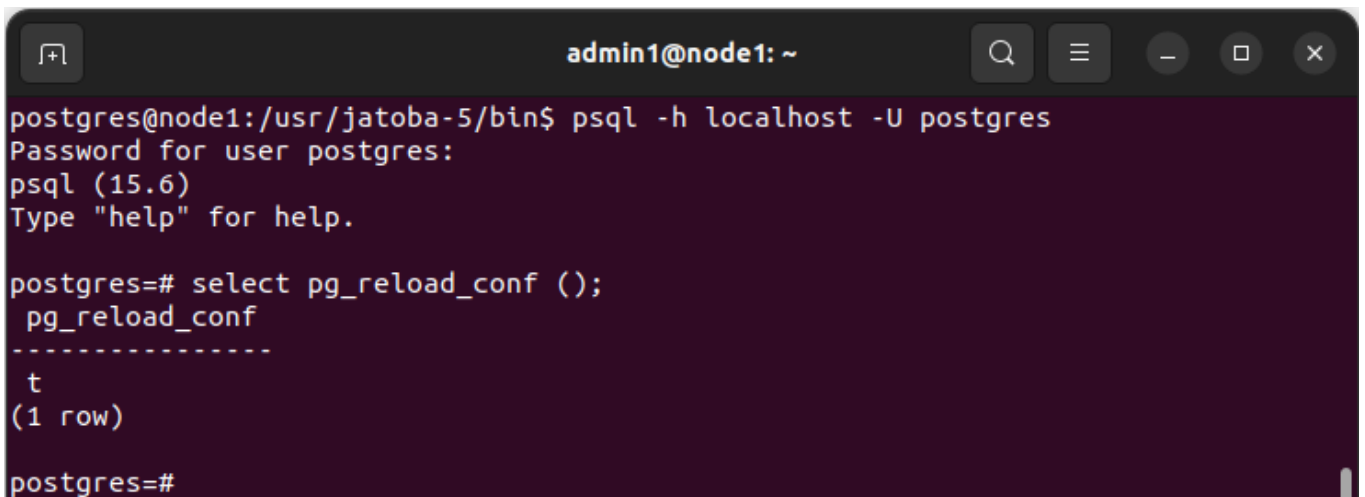
At the bottom of the terminal, there is a status bar with various keyboard shortcuts: `^G Help`, `^O Write Out`, `^W Where Is`, `^K Cut`, `^T Execute`, `^C Location`, `^X Exit`, `^R Read File`, `^_ Replace`, `^U Paste`, `^J Justify`, and `^_ Go To Line`.

Рисунок 5.1 – Добавление параметров репликации в конфигурационном файле «pg_hba.conf»

Применение установленных параметров в конфигурационном файле «pg_hba.conf» выполняется от имени и справами привилегированного пользователя СУБД SQL-командой:

```
select pg_reload_conf ();
```

Параметры применяются без перезагрузки СУБД.



```
admin1@node1: ~  
postgres@node1:/usr/jatoba-5/bin$ psql -h localhost -U postgres  
Password for user postgres:  
psql (15.6)  
Type "help" for help.  
  
postgres=# select pg_reload_conf ();  
pg_reload_conf  
-----  
t  
(1 row)  
  
postgres=#
```

Рисунок 5.2 – SQL-команда применения параметров конфигурационных файлов

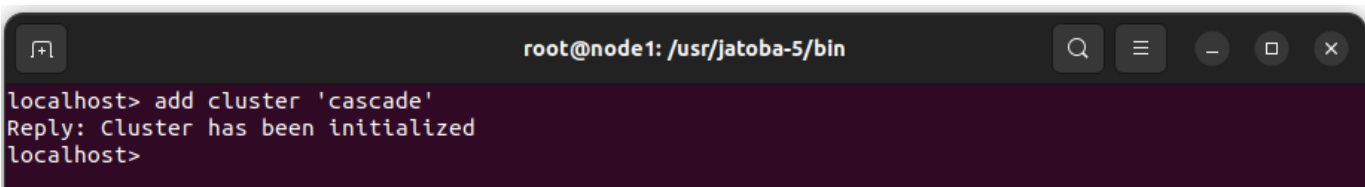
Следующим шагом является непосредственное конфигурирование каскадного кластера.

На сервере, который будет выполнять роль Master, подключиться к утилите «jadowg_ctl» в терминале ОС:

```
./jadowg_ctl
```

– создать кластер:

```
add cluster 'cascade'
```



```
root@node1: /usr/jatoba-5/bin  
localhost> add cluster 'cascade'  
Reply: Cluster has been initialized  
localhost>
```

Рисунок 5.3 – Команда создания кластера

– присвоить узлу роль Master:

```
cluster add master 10.116.102.54 12345
```

```
root@node1: /usr/jatoba-5/bin
localhost> add cluster 'cascade'
Reply: Cluster has been initialized
localhost> cluster add master 10.116.102.54 12345
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Node has been successfully initialized
localhost>
```

Рисунок 5.4 – Присвоение узлу роли Master

На данном этапе обозначен узел, который будет выполнять роль Master в кластере. Схема исходного состояния кластера представлена на 5.5.

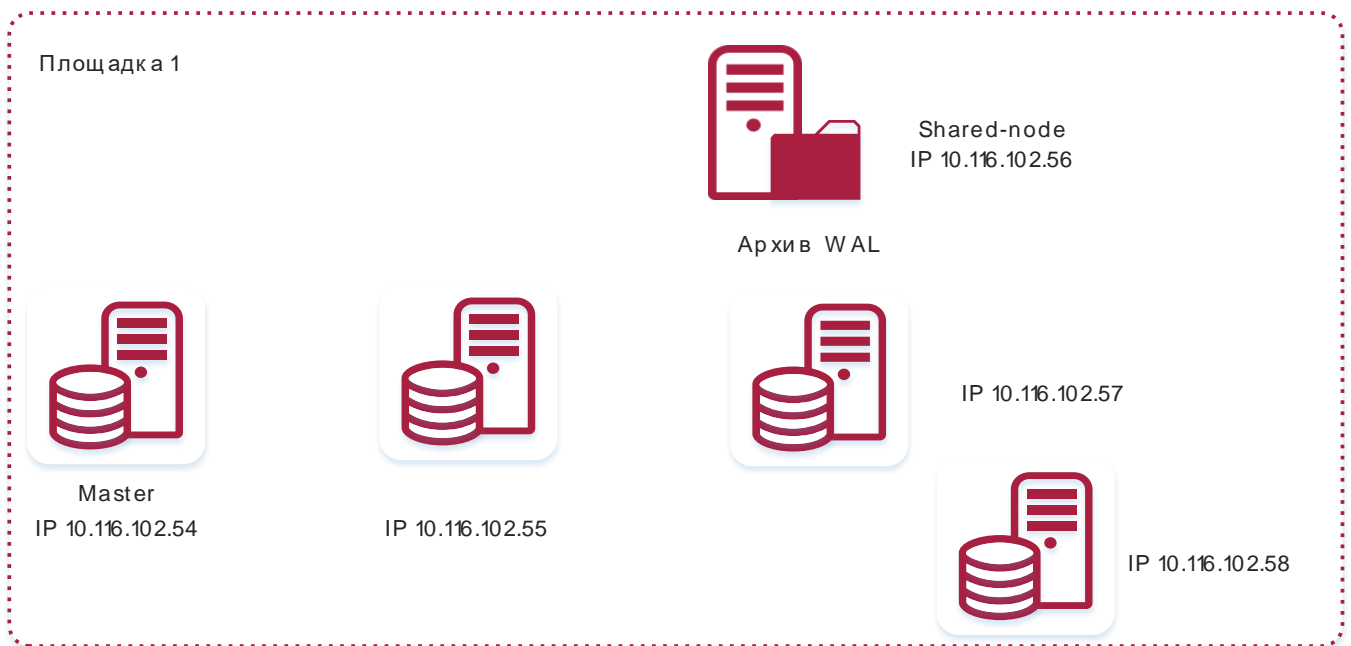


Рисунок 5.5 – Исходное состояние узлов кластера с каскадной репликацией

После определения главного узла с ролью Master, к нему добавляется подчиненный узел (Slave Main) при помощи команды:

```
cluster add slave 10.116.102.55 12345
```

Для того чтобы проверить статус добавленного узла необходимо выполнить следующую команду:

```
cluster status
```

В выводе состояния кластера будет отображаться информация, что кластер создан и состоит из двух узлов.

```

root@node1: /usr/jatoba-5/bin

localhost> cluster status

+-----+-----+
|Jadog version|2.2.0|
+-----+-----+

+-----+-----+-----+-----+
|Cluster name|AutoFailover|AutoDCFailover|Public IP|
+-----+-----+-----+-----+
|cascade     |t             |t             |10.116.102.81/24|
+-----+-----+-----+-----+

+-----+-----+
|Datacenter|STATUS|
+-----+-----+
|DEFAULT   |ACTIVE|
+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Datacenter|Node           |State           |Connection state|Replication type|Write LSN|Replication slot name|Slot status|Public IP|Node status|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT   |10.116.102.54:12345(node1)|Master(ACTIVE)|t              |-              |[0/3000198] -      |f          |f        |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT   |10.116.102.55:12345()    |Slave(ACTIVE)|t              |async          |[0/3000198] node2  |t          |f        |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

localhost>

```

Рисунок 5.6 – Вывод состояние кластера

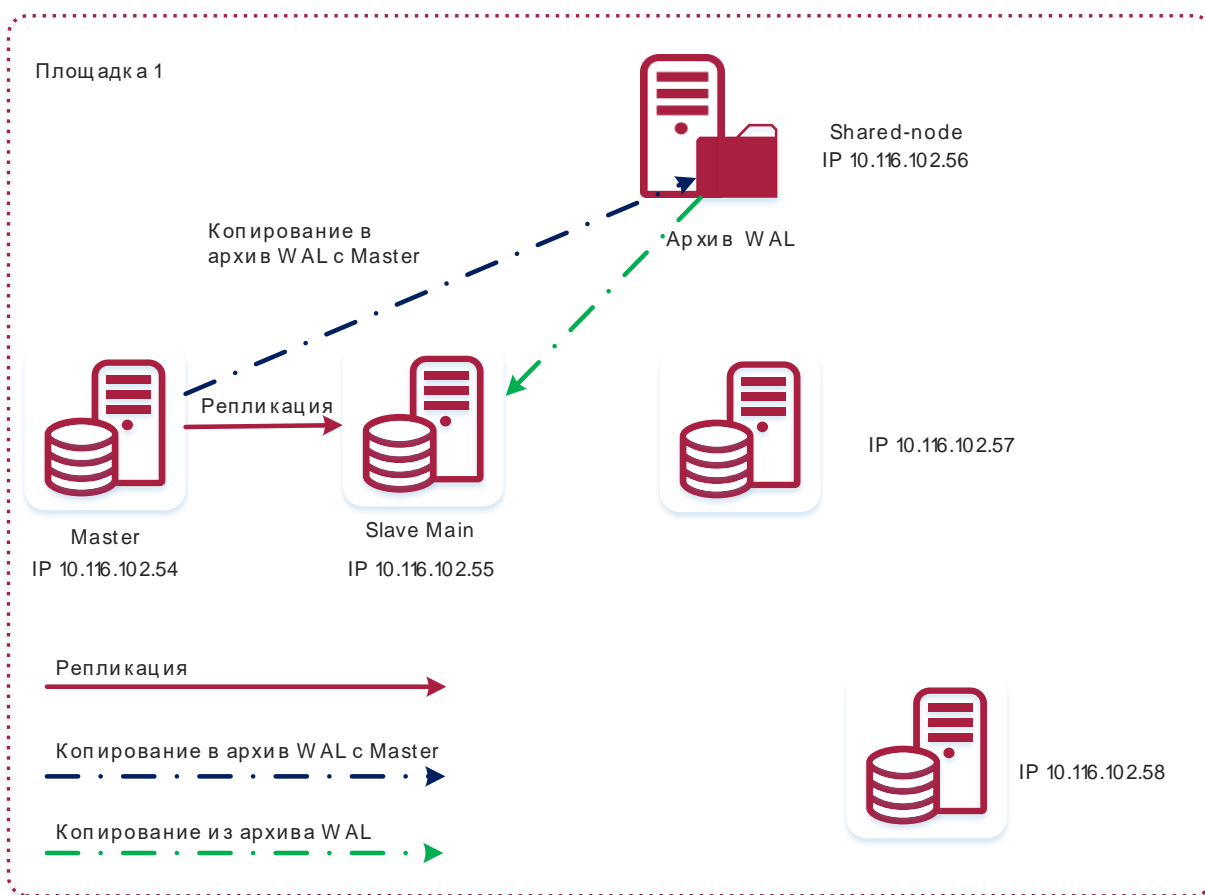
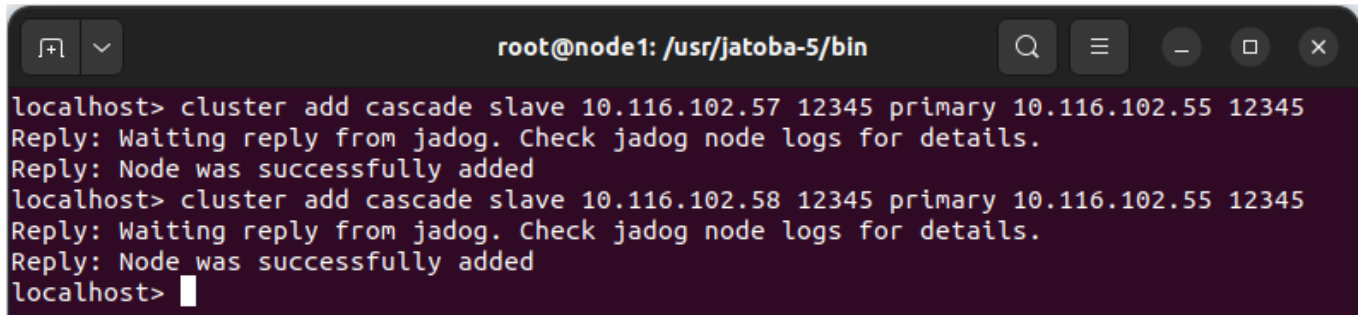


Рисунок 5.7 – Структура кластера с подключенным вторым подчиненным главным узлом (Slave Main)

Ко второму подчиненному главному узлу (Slave Main) подключаются узлы с ролью подчиненных каскадных узлов (Slave Cascade):

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

```
cluster add cascade slave 10.116.102.57 12345 primary  
10.116.102.55 12345  
cluster add cascade slave 10.116.102.58 12345 primary  
10.116.102.55 12345
```

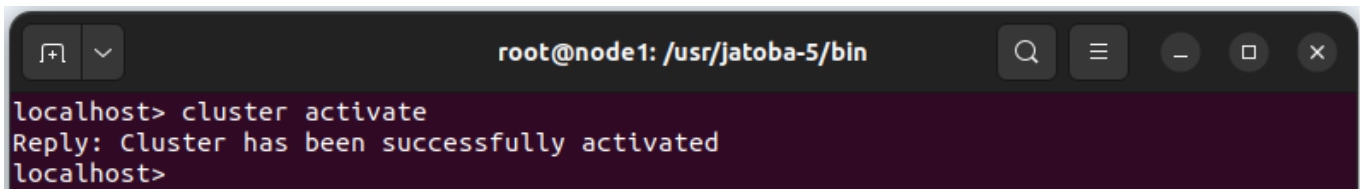


```
root@node1: /usr/jatoba-5/bin  
localhost> cluster add cascade slave 10.116.102.57 12345 primary 10.116.102.55 12345  
Reply: Waiting reply from jadog. Check jadog node logs for details.  
Reply: Node was successfully added  
localhost> cluster add cascade slave 10.116.102.58 12345 primary 10.116.102.55 12345  
Reply: Waiting reply from jadog. Check jadog node logs for details.  
Reply: Node was successfully added  
localhost>
```

Рисунок 5.8 - подключение узлов Slave Cascade

Активировать кластер командой, т.е. активировать использование публичного IP-адреса (public ip):

```
cluster activate
```



```
root@node1: /usr/jatoba-5/bin  
localhost> cluster activate  
Reply: Cluster has been successfully activated  
localhost>
```

Рисунок 5.9 – Команда активации кластера

Вывести статус кластера командой:

```
cluster status
```

В выводе статуса кластера видно, что в кластере:

- узел Node1 IP-10.116.102.54 выполняет роль Master;
- узел Node2 IP-10.116.102.55 выполняет роль Slave или Slave Main и подключен к узлу Node1 IP-10.116.102.54;
- узлы Node4 IP-10.116.102.57 и Node5 IP-10.116.102.58 выполняют роль Slave или Slave Cascade и подключены к узлу Node2 IP-10.116.102.55.

```

root@node1: /usr/jatoba-5/bin

localhost> cluster status

-----+-----+
|Jadog version|2.2.0|
+-----+-----+

+-----+-----+-----+-----+
|Cluster name|AutoFailover|AutoDCFailover|Public IP|
+-----+-----+-----+-----+
|cascade|t|t|10.116.102.81/24|
+-----+-----+-----+-----+

+-----+-----+
|Datacenter|STATUS|
+-----+-----+
|DEFAULT|ACTIVE|
+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Datacenter|Node|State|Connection state|Replication type|Write LSN|Replication slot name|Slot status|Public IP|Node status|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT|10.116.102.54:12345(node1)|Master(ACTIVE)|t|-|0/3000198|-|f|t|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT|10.116.102.55:12345()|Slave(ACTIVE)|t|async|0/3000198|node2|t|f|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+
|Datacenter|Cascade node|State|Primary node|Node status|
+-----+-----+-----+-----+
|DEFAULT|10.116.102.57:12345()|Slave(ACTIVE)|10.116.102.55:12345()|
+-----+-----+-----+-----+
|DEFAULT|10.116.102.58:12345()|Slave(ACTIVE)|10.116.102.55:12345()|
+-----+-----+-----+-----+

localhost>

```

Рисунок 5.10 – Вывод статуса кластера с каскадной репликацией

Таким образом сформирован кластер с каскадной репликацией.

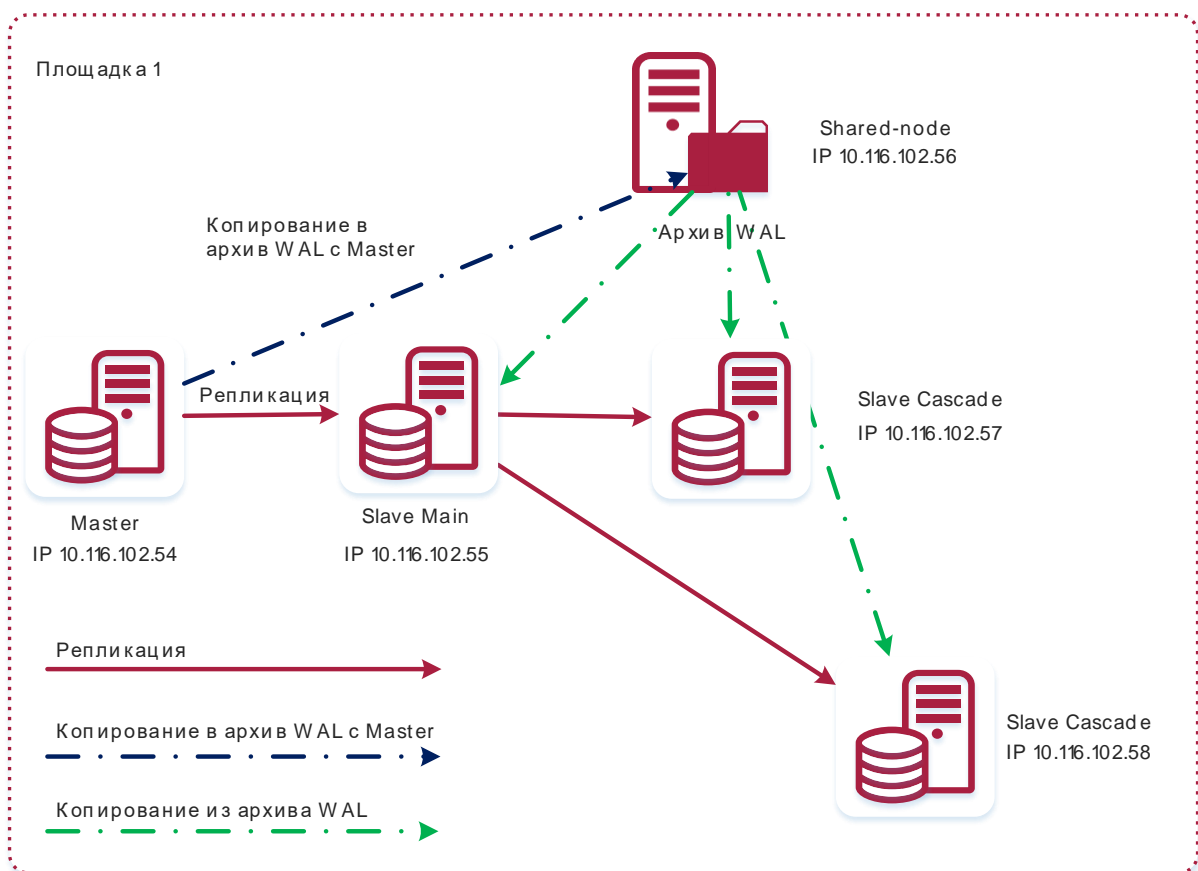


Рисунок 5.11 – Схема сформированного кластера с каскадной репликацией

5.4. Работа кластера с каскадной репликацией в дата-центрах

Кластер с каскадной репликацией имеет функциональную возможность работы в дата-центрах и поддерживаются следующие конфигурации:

- размещение кластера в одном дата-центре;

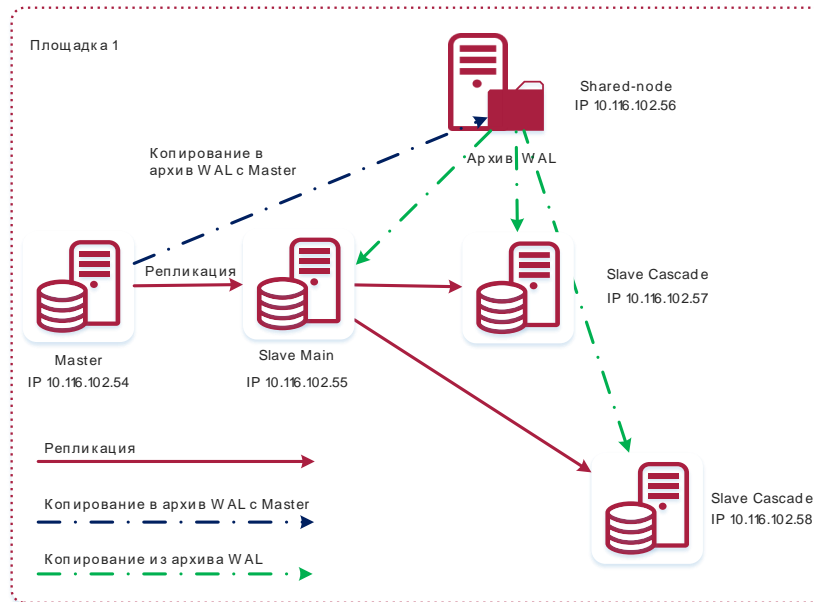


Рисунок 5.12 – Схема размещения кластера в одном дата-центре

- размещение узла с ролью Master в одном дата-центре и резервных узлов Slave Main, Slave Cascade в другом дата-центре.

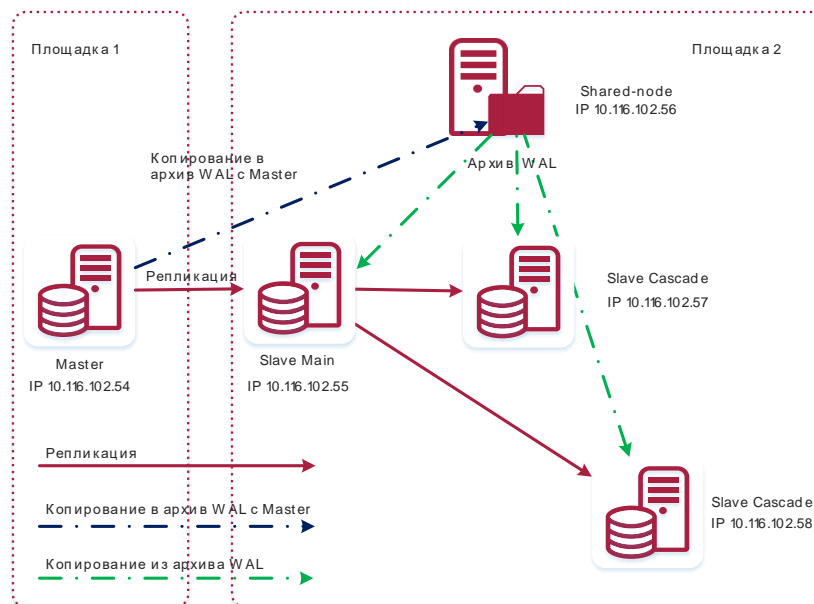


Рисунок 5.13 – Схема размещения узла с ролью Master в одном дата-центре и резервных узлов Slave Main, Slave Cascade в другом дата-центре



При выполнении операции смены главного узла (Switchover) выполняется смена роли главного узла (Master) кластера. При этом подчиненные резервные узлы (Slave Cascade) переключаются на «новый» главный узел.

Основные правила определения нового Slave Main в кластере с каскадной репликацией:

- Предыдущий узел Slave Main при переключении не может быть новым Slave Main узлом для Slave Cascade;
- Узел не может быть Slave Main для себя;
- Slave Main узел не может быть выбран из числа узлов-арбитров (Referee);
- Slave Main узел не может быть выбран из числа Slave Cascade;
- В случае, если не удалось определить новый Slave Main узел для подчиненных каскадных узлов Slave Cascade, то Slave Main узлом станет новый мастер. В этом случае, кластер перестает быть каскадным.

6. ГЕОРАСПРЕДЕЛЕННЫЙ, ОТКАЗОУСТОЙЧИВЫЙ КЛАСТЕР. РЕШЕНИЕ JA_DTC_AS

ja_DTC_AS — это инженерное решение применения компонента «jaDog», позволяющее объединять узлы кластера в дата-центры.

Дата-центр — это логическая сущность, позволяющая сопоставить узлы кластера по подсетям в дата-центрах.

6.1. Режимы работы геораспределенного отказоустойчивого кластера

6.1.1. Нормальный режим работы

При нормальном режиме работы кластера используется единственный публичный IP-адрес (Public IP) который зарезервирован и используется на основной площадке (DC1).

Узел Master:

- принимает подключения от пользователей;
- записывает изменения в БД;
- реплицирует данные на узлы Slave;
- записывает WAL-архив на сетевой диск.

Узлы Slave:

- при необходимости копируют WAL-архив с сетевого диска;
- принимают SQL-запросы на чтение данных.

6.1.2. Выход из строя Master в DC1

В случае выхода из строя узла Master на первой площадке (DC1) ближайший узел на площадке примет на себя роль ведущего узла. Запустится «Процедура обработки отказа (failover)», описанная в п.п 7.21 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01. Время переходных операций равно 10 сек.

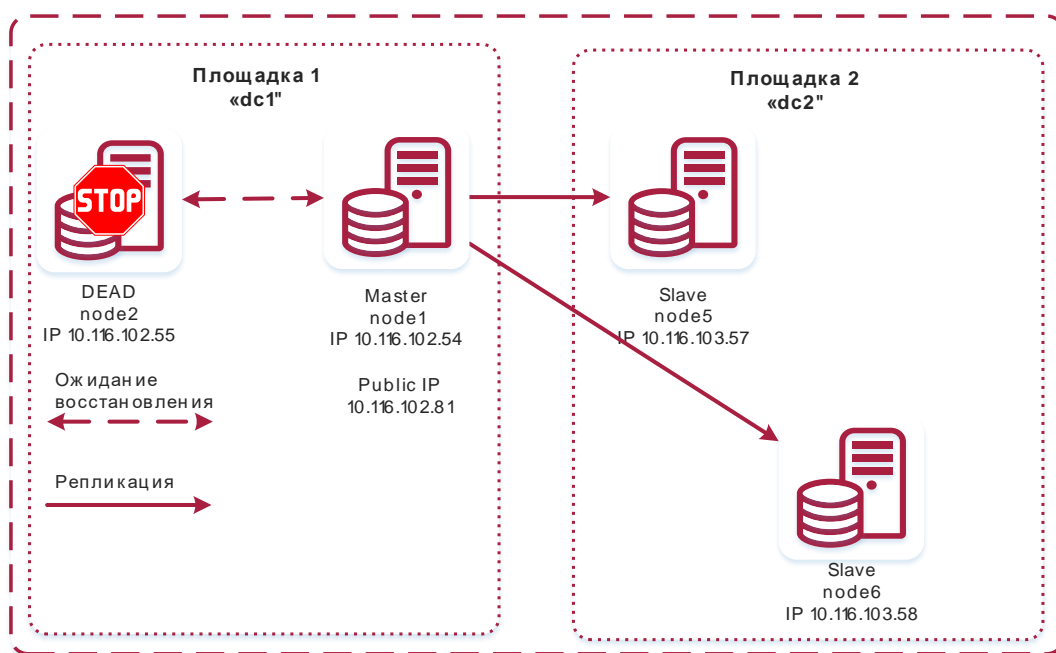


Рисунок 6.1 – Схема выхода из строя узла Master на первой площадке (DC1)

В целом кластер не потеряет работоспособности, но потребует внимания администратора СУБД.

6.1.3. dc_failover. Аварийный режим № 1. Последовательный выход из строя Master-серверов

В случае последовательного выхода из строя Master-серверов в сегменте кластера, расположенного в дата-центре «DC1», кластером будет обрабатываться режим смены дата-центров (dc_failover).

В режим «dc_failover» входят следующие шаги:

- определение нового сервера с ролью «Master» в работоспособном сегменте кластера на площадке «DC2»;
- переход на сервер с ролью «Master» на площадке «DC2» в течение 10 секунд;
- переход на резервную площадку «DC2», которая становится основной.

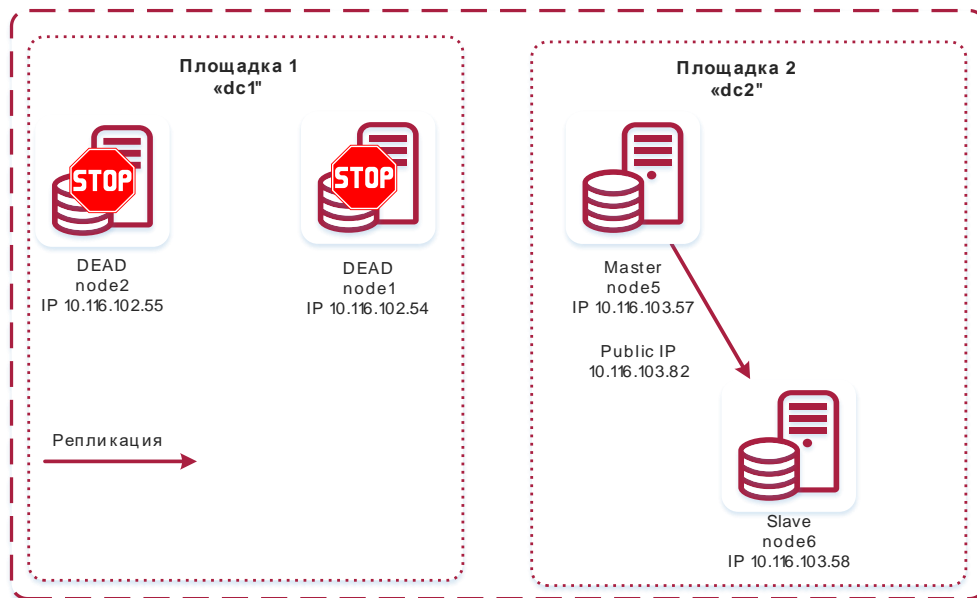


Рисунок 6.2 – Режим «dc_failover»

В режим «dc_failover» может выполняться как в ручном, так и в автоматическом режиме.

При установленном параметре `auto_DC_Promote = true` процедура обработки отказа выполнится в автоматическом режиме, т.е. автоматически выполнит вышеописанные действия.

При установленном параметре `auto_DC_Promote = false` кластер перейдет в режим ожидания. В режиме ожидания кластер ожидает от администратора команды «datacenter promote», описанной в п. 7.11 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01, которая служит подтверждением использования другой площадки (дата-центра). Команда может поступить с любого узла «jaDog».

6.1.4. DTC_Network_Failure. Аварийный режим № 2 Пропадание связи между площадками

В случае пропадания связи между площадками (дата-центрами, ЦОД) основной сегмент кластера будет фиксировать отсутствие репликации с резервной площадкой и продолжит работу в нормальном режиме.

Резервный сегмент кластера перейдет в режим ожидания. В этом режиме будет ожидать восстановления связи между площадками и возобновления репликации или команды «datacenter promote», описанной в п. 7.11 части первой «Компонент jaDog».

Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01, от администратора или Global Server Load Balancing (GLSB).

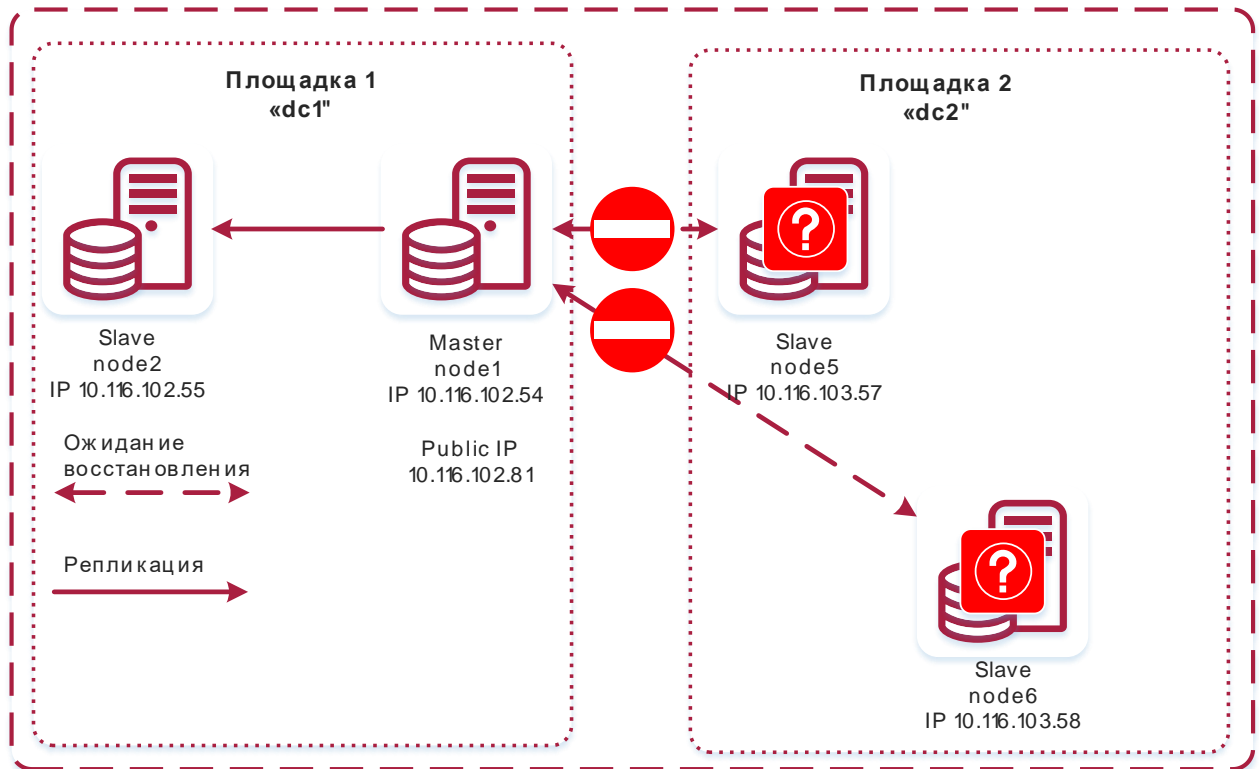


Рисунок 6.3 – Аварийный режим DTC_Network_Failure

6.1.5. DC_Promote. Смена роли DC

В штатном режиме перевода кластера GLSB или администратор:

- переводит весь трафик на резервную площадку «DC2»;
- DC_Health_Script понимает, что трафик перенаправлен в резервную площадку и дает команду «DC_Promote» jaDog на смену площадки;
- jaDog должен инициировать процесс «DC_Promote».

После назначения нового мастера будет поднят Public IP в «DC2» и все системы могут работать с СУБД.

Пример выполнения команды datacenter promote (DC_Promote) приведен в п. 6.1.7 данного руководства.

6.1.6. Нештатный обратный DC_Promote

В ситуации, когда кластер переключается на резервную площадку «DC2» с основной площадки «DC1», он находится в промежуточном состоянии. Прерывание мероприятия недопустимо.

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

Если во время мероприятия по смене площадок (дата-центров, ЦОД) поступит обратная команда, то кластер приостановит ее выполнение. Это достигается механизмом асинхронного выполнения команд таких как:

- cluster add slave ip (см. п. 6.16 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01);
- cluster delete node (см. п. 7.3 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01);
- switchover (см. п. 5.3.3 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01);
- datacenter promote (dcPromote) (см. п. 7.11 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01).

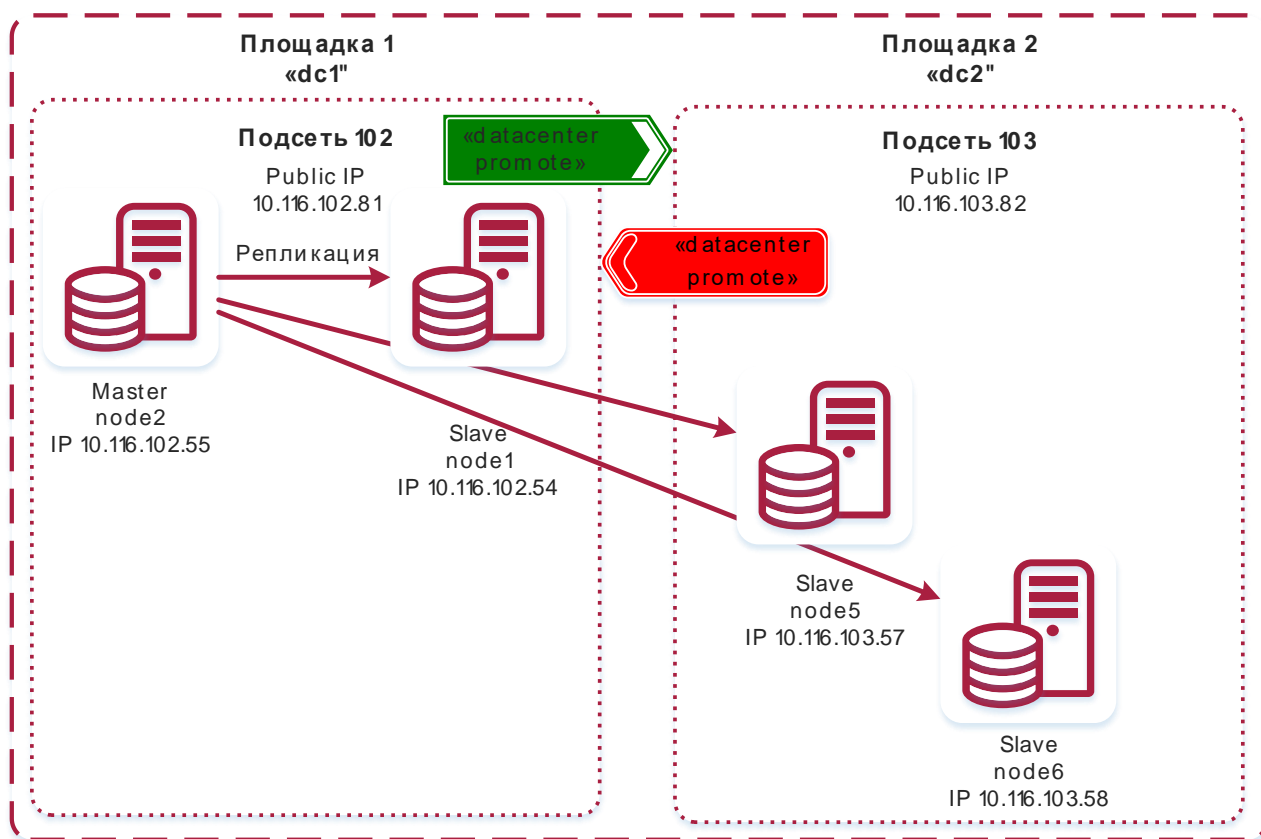


Рисунок 6.4 – Нештатный обратный «DC_Promote»

Выполнение команды по смене площадок (дата-центров, ЦОД) начнется только после выполнения первой команды.

6.1.7. Пример выполнения «DC_Promote»

Работа кластера в разных подсетях, и соответственно в разных дата-центрах, требует подготовительных действий, т.к. конфигурационный файл правил аутентификации в СУБД «pg_hba.conf» не настраивается автоматически на разные подсети при конфигурировании узла кластера.

Схема первоначального состояния узлов кластера представлена на рисунке 6.5

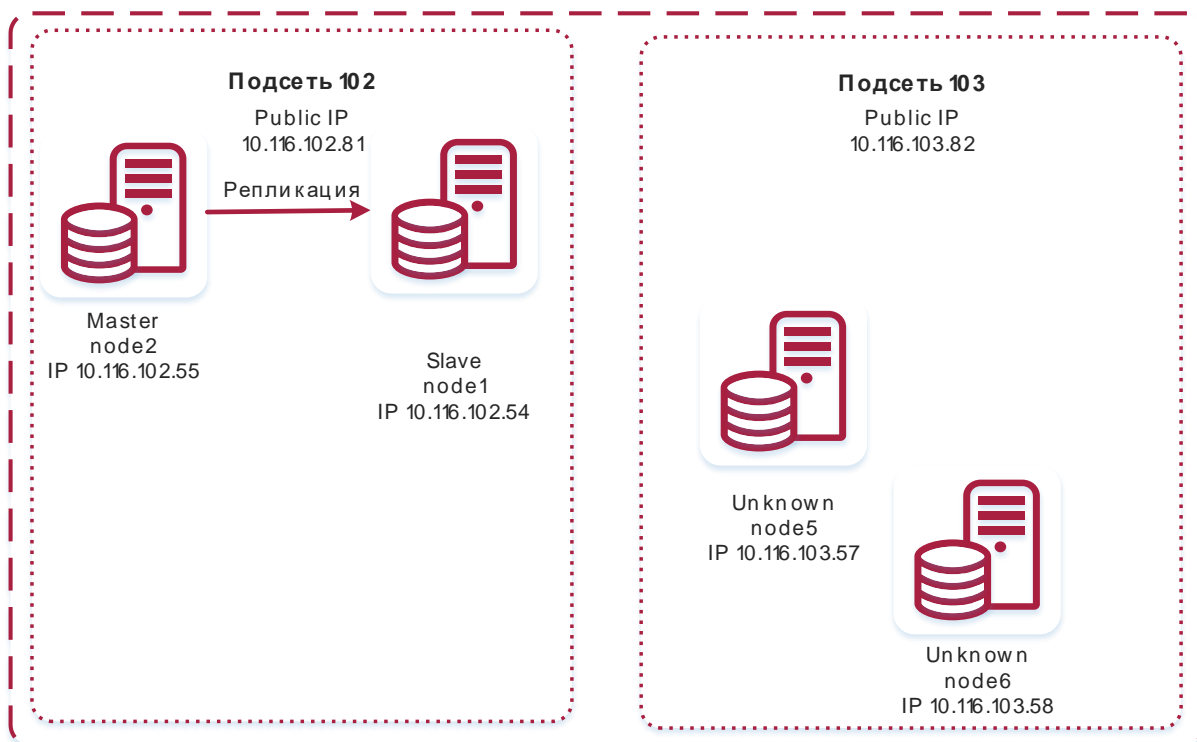


Рисунок 6.5 – Начальное состояние узлов кластера

В рассматриваемом примере узлы кластера имеют сетевую адресацию, представленную в таблице 6.1.

Таблица 6.1 – Сетевая адресация серверов стенда кластера «jaDog» для работы в дата-центре

| № | Имя сервера | IP-адрес | Маска подсети | Public IP | Роль/состояние | Дата-центр |
|--------------------|-------------|------------------|---------------|------------------|----------------|------------|
| 1 | JDS | 10.116.102.40 | 255.255.255.0 | | | |
| Подсеть 102 | | | | | | |
| 2 | Node1 | 10.116.102.54/24 | 255.255.255.0 | 10.116.102.81/24 | Slave | dc1 |
| 3 | Node2 | 10.116.102.55/24 | 255.255.255.0 | 10.116.102.81/24 | Master | dc1 |
| 4 | Node3 | 10.116.102.57/24 | 255.255.255.0 | | | |
| 5 | Node4 | 10.116.102.58/24 | 255.255.255.0 | | | |
| Подсеть 103 | | | | | | |
| 6 | Node5 | 10.116.103.57/24 | 255.255.255.0 | 10.116.103.82/24 | Unknown | dc2 |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| | | | | | | |
|---|-------|------------------|---------------|------------------|---------|-----|
| 7 | Node6 | 10.116.103.58/24 | 255.255.255.0 | 10.116.103.82/24 | Unknown | dc2 |
|---|-------|------------------|---------------|------------------|---------|-----|

Дата-центры еще не сформированы.

Узел Node2 выполняет роль «Master», Узел Node1 выполняет роль «Slave» и сконфигурированы на публичный IP-адрес 10.116.102.81/24.

Узлы Node5 и Node6 являются свободными и сконфигурированы на публичный IP-адрес 10.116.103.82/24.

6.1.7.1 Подготовительные действия для создания кластера

На узле Node2 с ролью «Master» потребуется внести дополнительные параметры для подключения узлов из другой подсети, внося следующие строки:

```
host replication jalog_user 10.116.102.0/24 scram-sha-256
host replication jalog_user 10.116.103.0/24 scram-sha-256
```

```

root@node1: /var/lib/jatoba/18/data
GNU nano 6.2 pg_hba.conf *
# TYPE DATABASE USER ADDRESS METHOD
#-----
# jaDog autogenerate parameters
#-----
local all postgres scram-sha-256
local all jalog_user scram-sha-256
host all postgres 10.116.102.0/24 trust
host all postgres 127.0.0.1/32 scram-sha-256
host all jalog_user 127.0.0.1/32 scram-sha-256
host all jalog_user 10.116.102.0/24 scram-sha-256
host all jalog_user 10.116.103.0/24 scram-sha-256
host replication jalog_user 127.0.0.1/32 scram-sha-256
host replication jalog_user 10.116.102.0/24 scram-sha-256
host replication jalog_user 10.116.102.0/24 scram-sha-256
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

```

Рисунок 6.6 – Параметры конфигурационного файла pg_hba.conf на узле «Master»

Сохранить внесенные изменения.

Применение параметров целесообразнее выполнить не через перезагрузку служб, а через SQL-команду. Для чего от имени и с правами пользователя «postgres» войти в СУБД и выполнить SQL-команду:

```
SELECT pg_reload_conf();
```

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

```

root@node2: /home/admin1
postgres@node2:/home$ cd /usr/jatoba-5/bin/
postgres@node2:/usr/jatoba-5/bin$ psql
Password for user postgres:
psql (15.5)
Type "help" for help.

postgres=# SELECT pg_reload_conf();
 pg_reload_conf 
-----
 t
(1 row)

postgres=#
  
```

Рисунок 6.7 – Выполнение перезагрузки параметров СУБД

Далее можно переходить к конфигурированию кластера в дата-центрах в разделе JDS «Список кластеров».

6.1.7.2 Добавление узлов в кластер из другой подсети. Меню «Узел» (Node)

Авторизовавшись на узле кластера с ролью «Master» необходимо добавить резервные узлы из другой подсети при помощи команд в консольной утилите «jadowg_ctl»:

```

cluster add slave 10.116.103.57 12345
cluster add slave 10.116.103.58 12345
  
```

Промежуточное состояние стенда представлено в таблице 6.2.

Таблица 6.2 – Сетевая адресация серверов стенда кластера «jaDog» для работы в дата-центре

| № | Имя сервера | IP-адрес | Маска подсети | Public IP | Роль/состояние | Дата-центр |
|-------------|-------------|------------------|---------------|------------------|----------------|------------|
| 1 | JDS | 10.116.102.40 | 255.255.255.0 | | | |
| Подсеть 102 | | | | | | |
| 2 | Node1 | 10.116.102.54/24 | 255.255.255.0 | 10.116.102.81/24 | Slave | |
| 3 | Node2 | 10.116.102.55/24 | 255.255.255.0 | 10.116.102.81/24 | Master | |
| 4 | Node3 | 10.116.102.57/24 | 255.255.255.0 | | | |
| 5 | Node4 | 10.116.102.58/24 | 255.255.255.0 | | | |
| Подсеть 103 | | | | | | |
| 6 | Node5 | 10.116.103.57/24 | 255.255.255.0 | 10.116.103.82/24 | Slave | |
| 7 | Node6 | 10.116.103.58/24 | 255.255.255.0 | 10.116.103.82/24 | Slave | |

6.1.7.3 Создание Дата-центров

Каждому из дата-центров требуется дать название. Для рассматриваемого примера создаются дата-центры:

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

- «dc1»;
- «dc2».

Это соответствует команде в утилите «jadog_ctl»:

```
datacenter create '[name]'
```

6.1.7.4 Присоединение узлов кластера к Дата-центру

К созданным дата-центрам присоединяются узлы кластера:

- Node1 IP-10.116.102.54/24 и Node2 IP- 10.116.102.55/24 к «dc1»;
- Node5 IP-10.116.103.57/24 и Node6 IP-10.116.103.58/24 к «dc2».

Это соответствует команде в утилите «jadog_ctl»:

```
datacenter 'dc1' attach node 10.116.102.54 12345
datacenter 'dc1' attach node 10.116.102.55 12345
datacenter 'dc2' attach node 10.116.103.57 12345
datacenter 'dc2' attach node 10.116.103.58 12345
```

В результате узлы кластера распределены по дата-центрам. Один узел выполняет роль «Master», остальные узлы работают с ролью «Slave».

В подсети «102» кластер работает с Public IP – 10.116.102.81/24. Второй публичный адрес 10.116.103.82/24 в подсети «103» не используется. Полученная конфигурация кластера представлена в таблице 6.3 и показана на рисунке 6.8.

Таблица 6.3 – Сетевая адресация серверов стенда кластера «jaDog», работающих в дата-центрах

| № | Имя сервера | IP-адрес | Маска подсети | Public IP | Роль/состояние | Дата-центр |
|-------------|-------------|------------------|---------------|------------------|----------------|------------|
| 1 | JDS | 10.116.102.40 | 255.255.255.0 | | | |
| Подсеть 102 | | | | | | |
| 2 | Node1 | 10.116.102.54/24 | 255.255.255.0 | 10.116.102.81/24 | Slave | dc1 |
| 3 | Node2 | 10.116.102.55/24 | 255.255.255.0 | 10.116.102.81/24 | Master | dc1 |
| 4 | Node3 | 10.116.102.57/24 | 255.255.255.0 | | | |
| 5 | Node4 | 10.116.102.58/24 | 255.255.255.0 | | | |
| Подсеть 103 | | | | | | |
| 6 | Node5 | 10.116.103.57/24 | 255.255.255.0 | 10.116.103.82/24 | Slave | dc2 |
| 7 | Node6 | 10.116.103.58/24 | 255.255.255.0 | 10.116.103.82/24 | Slave | dc2 |

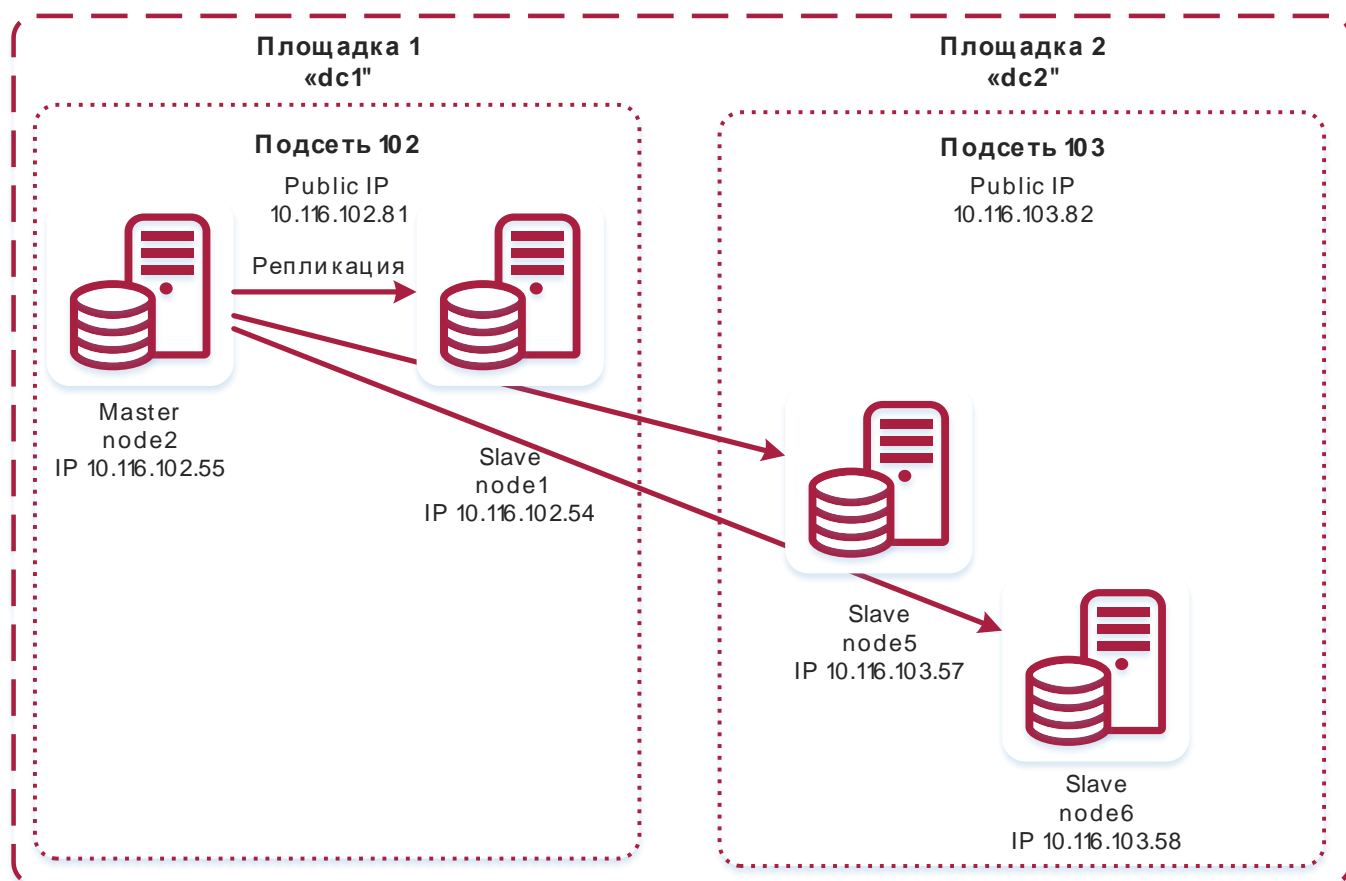


Рисунок 6.8 – Схема кластера в дата-центрах

6.1.7.5 Выполнение «DC Promote»

Data Centre Promote (англ.) означает продвижение центра обработки данных (дата-центра) до уровня ведущего. Функциональная возможность «DC Promote» состоит в том, что «ведущий» дата-центр, в котором был узел кластера с ролью «Master», сменит свою роль на «ведомого». А резервный, «ведомый» дата-центр, принудительно продвинется до роли «ведущего» и в его подсети выбранный узел кластера возьмет на себя роль «Master».

Функциональная возможность «DC Promote» может использоваться при:

- аварийных ситуациях;
- плановых переключениях.

Выполнить «DC Promote» возможно подключившись к узлу, который планируется на роль «Master». В данном случае это узел кластера Node5 с IP-10.116.103.57 находящийся в «dc2».

После подтверждения действия начнется процесс принудительного продвижения дата-центра «dc2». Узел кластера Node5 возьмет на себя роль «Master» и автоматически активируется Public IP-10.116.103.82/24.

Схема репликации кластера изменится на представленную.

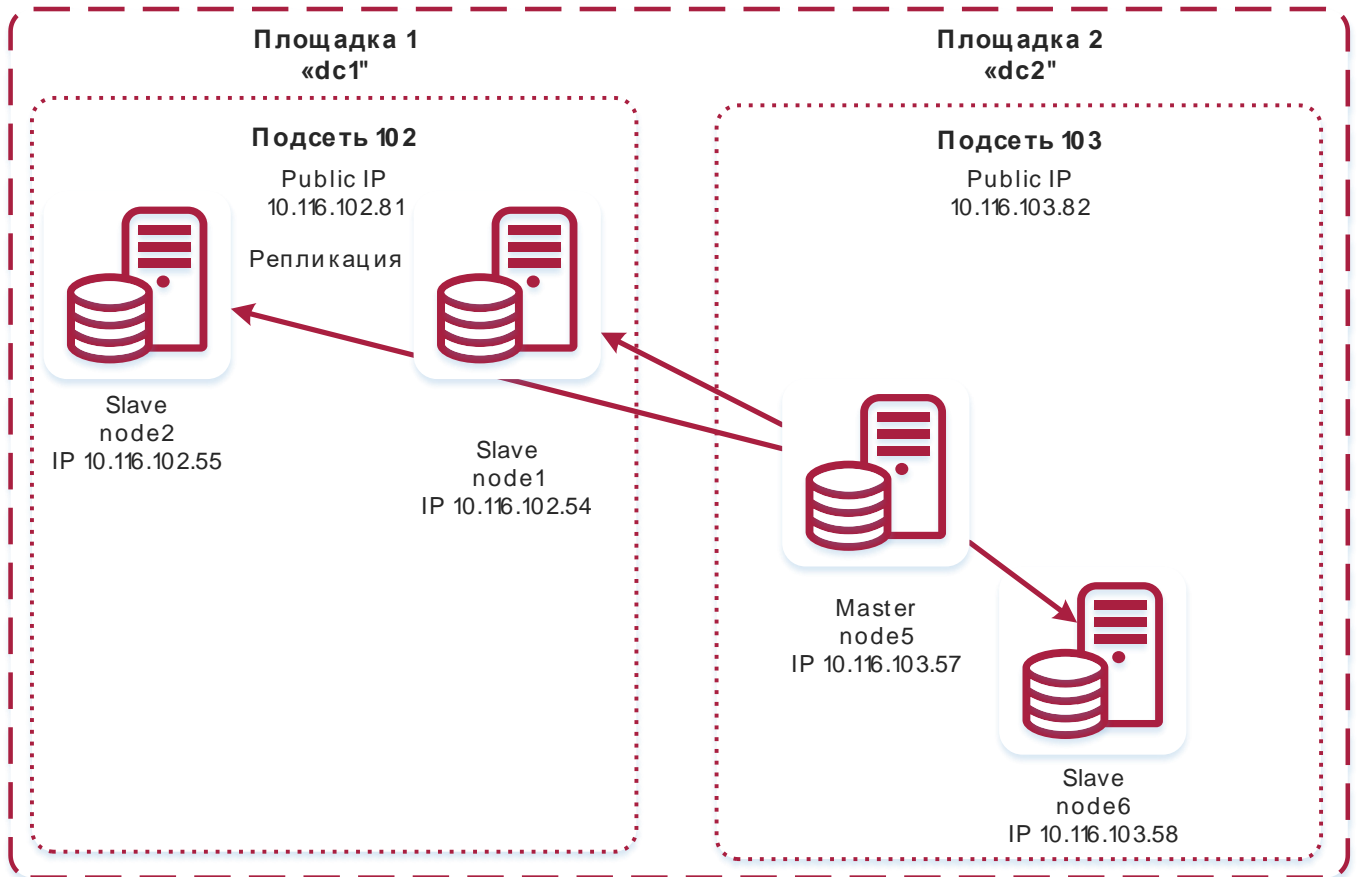


Рисунок 6.9 – Схема кластера после выполнения «DC Promote»

7. ПЕРЕНАПРАВЛЕНИЕ ПОДКЛЮЧЕНИЙ МЕЖДУ ДАТА-ЦЕНТРАМИ

Концепция перенаправления подключений состоит в том, что компонент балансировки подключений пользователей к СУБД «jaPooler» установлен в устойчивой подсети, принимает подключения от приложения (пользователей) и, опрашивая Public_IP, направляет запрос в тот узел кластера, в тот дата-центр, который откликается на запрос. В описываемой конфигурации компонент «jaPooler» работает в отдельном «прокси» режиме.

Принципиальная схема сети и взаимодействия компонентов СУБД «Jatoba» в ней, представлена на рисунке 7.1.

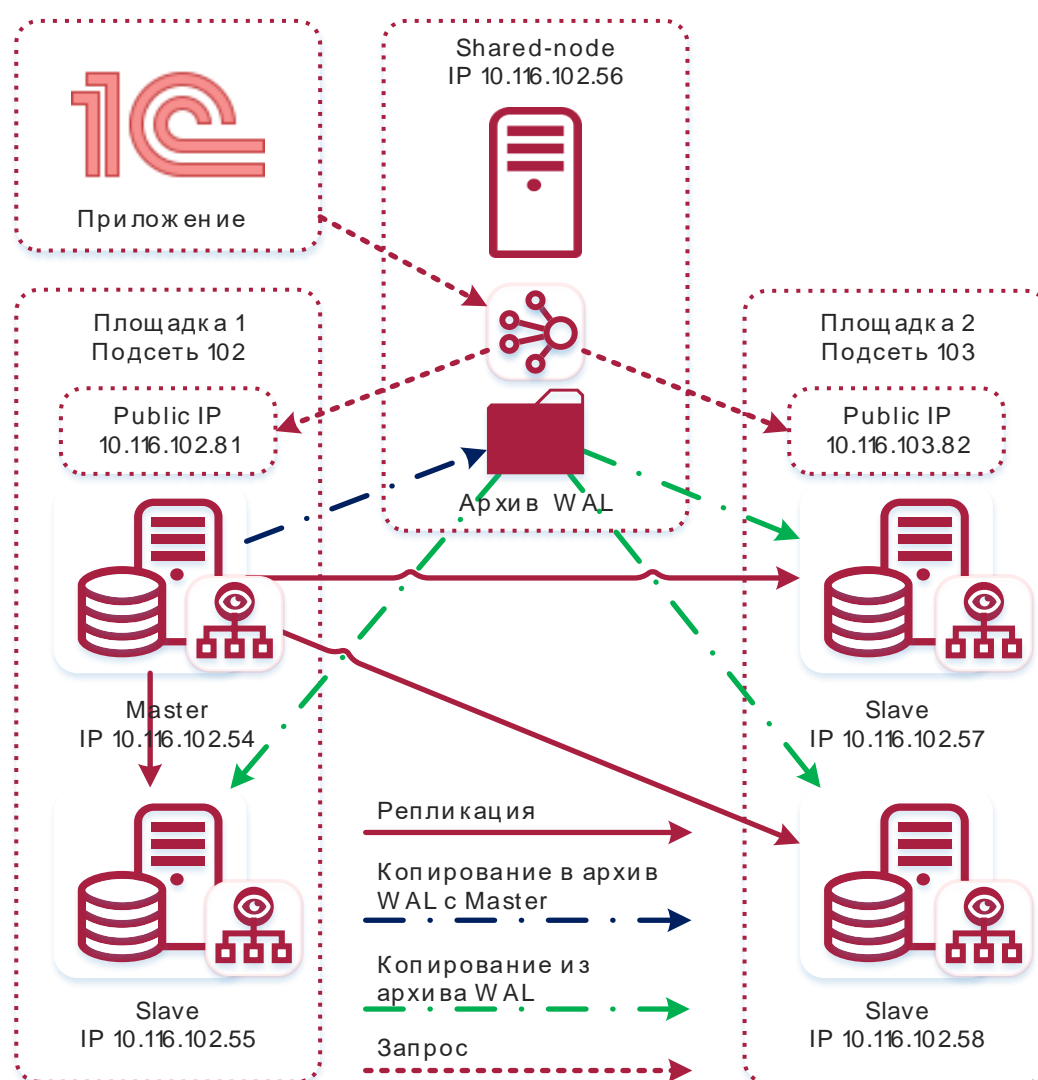


Рисунок 7.1 – Принципиальная схема взаимодействия компонент СУБД

На указанной схеме 4 основных блока:

- 1) Клиентское приложение, которое настраивается на подключение к СУБД с установленным компонентом «jaPooler».

2) Сервер СУБД «Jatoba» с установленным компонентом «jaPooler», который параллельно выполняет роль файлового сервера с архивом WAL.

3) Узлы кластера в основной подсети и в первом дата-центре.

4) Узлы кластера в резервной подсети и во втором дата-центре.

Узлы кластера настраиваются по классической схеме асинхронной репликацией, как описано в разделе 6 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01. Дополнительно может быть настроено архивирование WAL в сетевой каталог.

В рассматриваемом примере установки и конфигурирования отказоустойчивого кластера «jaDog» на ОС Ubuntu 22.04 используются параметры сети кластера, приведенные в таблице 7.1.

Таблица 7.1 – Конфигурация сети кластера при перенаправлении подключений

| № | Имя сервера | IP-адрес | Маска подсети | Public IP | Роль | Имя кластера | Дата-центр |
|-------------|-------------|------------------|---------------|------------------|-----------------------|--------------|------------|
| Подсеть 102 | | | | | | | |
| 1 | Node1 | 10.116.102.54/24 | 255.255.255.0 | 10.116.102.81/24 | Master | test | dc1 |
| 2 | Node2 | 10.116.102.55/24 | 255.255.255.0 | 10.116.102.81/24 | Slave | | dc1 |
| Подсеть 102 | | | | | | | |
| 3 | Shared-node | 10.116.102.56/24 | 255.255.255.0 | | File-server, Balancer | | |
| Подсеть 103 | | | | | | | |
| 4 | Node6 | 10.116.103.57/24 | 255.255.255.0 | 10.116.103.82/24 | Slave | test | dc2 |
| 5 | Node7 | 10.116.103.58/24 | 255.255.255.0 | 10.116.103.82/24 | Slave | | dc2 |

Использование компонента «jaPooler» для перенаправления подключений между дата-центрами дополняет инженерное решение, описанное в разделе 6 «Геораспределенный, отказоустойчивый кластер. Решение JA_DTC_AS» настоящего руководства.

7.1. Настройка и запуск компонента «jaDog» на резервных узлах

Резервные узлы Node2, Node6 и Node7 настраиваются с параметрами, приведенными в таблице 7.2 по п. 6.8 части первой «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01 и в разделе 4 «Геораспределенный, отказоустойчивый кластер. Решение JA_DTC_AS» настоящего руководства.

Устанавливаемые параметры кластера, приведены в таблице 7.2.

Таблица 7.2 – Перечень устанавливаемых параметров для каскадной репликации

| № | Меню/Пункт меню | Параметры | Node1 | Node2 | Node6 | Node7 |
|----------|---|--|------------------|------------------|------------------|------------------|
| 1 | Database server and jalog directory settings Menu | | | | | |
| 1 | Database server binaries (path:db_bin_path) | [/usr/jatoba-6/bin] | = | = | = | = |
| 2 | Database server data (postgresql:db_data_path) | [/var/lib/jatoba/6/data] | = | = | = | = |
| 3 | Jalog configs (path:config_path) | [/usr/jatoba-6/etc/jalog] | = | = | = | = |
| 4 | Jalog module files (path:module_path) | [/usr/jatoba-6/share/jalog/scripts] | = | = | = | = |
| 5 | Jalog state (path:state_path) | [/usr/jatoba-6/etc/jalog] | = | = | = | = |
| 2 | Inter-jalog communication settings | | | | | |
| 1 | Jalog service name (main:service_name) | [jalog] | = | = | = | = |
| 2 | Jalog IP address (main:ip) | Текущий IP (10.116.102.54/24) | 10.116.102.54/24 | 10.116.102.55/24 | 10.116.103.57/24 | 10.116.103.58/24 |
| 3 | Jalog PORT number (main:port) | [12345] | = | = | = | = |
| 4 | SSL on (tls:tls) | [false] | = | = | = | = |
| 5 | Jalog interconnection user (main:interconnect_user) | [admin] | = | = | = | = |
| 3 | User / Admin access network settings | | | | | |
| 1 | Public address (main:public_address) | Внешний IP адрес (10.116.102.81/24) | 10.116.102.81/24 | 10.116.102.81/24 | 10.116.103.82/24 | 10.116.103.82/24 |
| 2 | Public address control interval (main:public_address_control_interval) | [5000] | = | = | = | = |
| 3 | Public address control attempts (main:public_address_control_attempts) | [3] | = | = | = | = |
| 4 | DB monitor timeout (postgresql:db_check_interval) | [5000] | = | = | = | = |
| 5 | Trusted IP address (main:trusted_address) | [] | [] | [] | [] | [] |
| 6 | RDBMS trusted ip (ja_hipe_cluster:rdbms_trusted_ip) | [] | [] | [] | [] | [] |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node6 | Node7 |
|----------|---|---|---------------------------------|-------|-------|-------|-------|
| | 7 | RDBMS trusted port (ja_hipe_cluster:rdbms_trusted_port) | [] | [] | [] | [] | [] |
| | 8 | Network interface name (main:network_interface) | Имя сетевого интерфейса (ens18) | = | = | = | = |
| | 9 | Jadog TCP port (main:user_interface_port) | [54321] | = | = | = | = |
| 4 | Administrator account list | | | | | | |
| | 1 | Add new account to the list | admin | admin | admin | admin | admin |
| | 2 | Delete account from the list | | | | | |
| | 3 | Clear all accounts | | | | | |
| 5 | Database server system account and connection settings | | | | | | |
| | 1 | Database host (db_connection_settings:db_host) | [127.0.0.1] | = | = | = | = |
| | 2 | Database port (db_connection_settings:db_port) | [5432] | = | = | = | = |
| | 3 | Database service name (db_connection_settings:db_service_name) | [jatoba-6] | = | = | = | = |
| | 4 | Database name (db_connection_settings:db_name) | [postgres] | = | = | = | = |
| | 5 | Jadog password file (db_connection_settings:db_passfile) | /usr/jatoba-6/bin/db_passfile | = | = | = | = |
| | 6 | Database auth method (db_connection_settings:db_auth_method) | [password] | = | = | = | = |
| | 7 | Jadog to database CA file (db_connection_settings:jadog_to_db_ca_file) | [] | [] | [] | [] | [] |
| | 8 | Jadog to database CRL file (db_connection_settings:jadog_to_db_crl_file) | [] | [] | [] | [] | [] |
| | 9 | Jadog to database cert file (db_connection_settings:jadog_to_db_cert_file) | [] | [] | [] | [] | [] |
| | 10 | Jadog to database key file (db_connection_settings:jadog_to_db_key_file) | [] | [] | [] | [] | [] |
| | 11 | Jadog to database SSL mode (db_connection_settings:ssl_mode) | [very-full] | = | = | = | = |
| | 12 | Jadog database user name (db_connection_settings:db_jadog_user) | [jadog_user] | = | = | = | = |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node6 | Node7 |
|----------|---|--|-------------------------------|--------------|--------------|--------------|--------------|
| | 1 3 | Jadog database user password (db_jadog_user_pass) | [] | = | = | = | = |
| | 1 4 | Database server OS user (system:system_user) | [postgres] | = | = | = | = |
| | 1 5 | Replication slot name (replication:slot_name) | [] | node1 | node2 | node6 | node7 |
| | 1 6 | Replication Slot TTL (replication:slot_ttl) | [300000] | = | = | = | = |
| 6 | «Failover setting» | | | | | | |
| | 1 | Auto failover mode [true/false] (cluster_behavior:autofailover_mode) | [true] | [true] | [true] | [true] | [true] |
| | 2 | Auto dc promote mode [true/false] (cluster_behavior:dc_autofailover_mode) | [true] | [true] | [true] | [true] | [true] |
| 7 | «Replication setting» | | | | | | |
| | 1 | Replication node name (replication:slot_name) | [] | node1 | node2 | node6 | node7 |
| | 2 | Synchronous commit setting (synchronous:synchronous_commit) | remote_apply | remote_apply | remote_apply | remote_apply | remote_apply |
| | 3 | Synchronous type (synchronous:synchronous_type) | ANY | ANY | ANY | ANY | ANY |
| | 4 | The number of synchronous standbys (synchronous:synchronous_commit_max_nodes) | 4 | 4 | 4 | 4 | 4 |
| | 5 | The wal_level value (postgresql:wal_level) | [replica] | = | = | = | = |
| | 6 | List of ignored replication slots (replication:ignore_replication_slots) | [] | = | = | = | = |
| 8 | WAL archive and restore settings | | | | | | |
| | 1 | System directory for WAL archive/restore (wal_archive:directory) | [/nfs/arhive_wal] | = | = | = | = |
| | 2 | WAL archive copy command (wal_archive:command) | [cp %p /nfs/arhive_wal/%f] | = | = | = | = |
| | 3 | WAL archive restore command (wal_restore:command) | [cp /nfs/arhive_wal/%f %p] | = | = | = | = |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node6 | Node7 |
|-----------|------------------------------|---|---|-------|-------|-------|-------|
| | 4 | WAL archive cleanup process on (wal_archive:cleanup_needed) | [false] | = | = | = | = |
| | 5 | WAL archive cleanup command (wal_archive:cleanup_command) | [/usr/jatoba-6/bin/pg_archivecleanu p /nfs/arhive_wal %s] | = | = | = | = |
| | 6 | WAL archive cleanup timeout (wal_archive:cleanup_timeout) | [30000] | = | = | = | = |
| 9 | Reporting and logging | | | | | | |
| | 1 | Log directory (log:path) | [/usr/jatoba-6/var/log/jadog] | = | = | = | = |
| | 2 | Log filename (log:file_name) | [jadog] | = | = | = | = |
| | 3 | Log file mode (log:mode) | [0600] | = | = | = | = |
| | 4 | Log format (log:type) | [csv] | = | = | = | = |
| | 5 | Max log file usage duration (log:rotation_age) | [1d] | = | = | = | = |
| | 6 | Max log file size (log:rotation_size) | [10MB] | = | = | = | = |
| | 7 | Truncate, not append, logs (log:truncate_on_rotation) | [false] | = | = | = | = |
| | 8 | Log verbosity (log:level) | [info] | = | = | = | = |
| | 9 | Security log directory (security_log:path) | [/usr/jatoba-6/var/log/jadog]. | = | = | = | = |
| | 10 | Security log filename (security_log:file_name) | [jadog-%a] | = | = | = | = |
| | 11 | Security log file mode (security_log:file_mode) | [0600] | = | = | = | = |
| | 12 | Allow to write into file (log:file) | [true] | = | = | = | = |
| | 13 | Allow to write into stdout (log:screen) | [false] | = | = | = | = |
| 10 | REST API Settings | | | | | | |
| | 1 | REST API use (rest_api:api_use) | [false] | = | = | = | = |
| | 2 | REST API listen address (rest_api:listen_address) | [0.0.0.0] | = | = | = | = |
| | 3 | REST API listen port (rest_api:listen_port) | [54443] | = | = | = | = |
| | 4 | REST API TLS server certificate (rest_api:cert_file) | [] | = | = | = | = |
| | 5 | REST API TLS server private key (rest_api:key_file) | [] | = | = | = | = |
| | 6 | REST API TLS CA certificate (rest_api:ca_file) | [] | = | = | = | = |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| № | | Меню/Пункт меню | Параметры | Node1 | Node2 | Node6 | Node7 |
|----|---|--|-----------|-------|-------|-------|-------|
| | 7 | REST API TLS server revocation list (rest_api:crl_file) | [] | = | = | = | = |
| 11 | Recovery settings | | | | | | |
| | 1 | Allow backup if dir exist (after_rewind_fail_backup:after_rewind_fail_backup_allow) | [true] | = | = | = | = |
| | 2 | Time delay (ms) of the cluster response (recovery:cluster_timeout) | [3000000] | = | = | = | = |
| | 3 | backup script file (backup:script_backup) | [] | = | = | = | = |
| 12 | Reset all settings to the default values | | | | | | |
| 13 | Check and show all settings | | | | | | |
| 14 | Save settings and setup jadog | | | | | | |

7.2. Установка и настройка компонента «jaPooler»

Компонент «jaPooler» для перенаправления подключений используется как утилита со своей службой в ОС.

Установка и настройка компонента «jaPooler» полностью описана в документе «Руководство по настройке. Часть 15. Балансировка подключений пользователей к СУБД. Компонент «jaPooler».

7.3. Конфигурирование компонента «jaPooler»

Основные настройки компонента «jaPooler» проводятся в файле /usr/jatoba-<ver>/etc/pgbouncer.ini. Неактивные параметры маркируются символами «;» «#» в начале строки.

В конфигурационном файле указываются публичные IP – адреса (Public IP) сегментов кластера, распределенных по дата-центрам. Строка имеет синтаксис:

```
postgres = host=<public ip node1>,<public ip node2>  
port=5432 host_balancer=off strategy=always_rw auth_user=postgres
```

В рассматриваемом примере строка в конфигурационном файле pgbouncer.ini будет иметь вид:

```
postgres = host=10.116.102.81,10.116.103.82 port=5432  
host_balancer=off strategy=always_rw auth_user=postgres
```

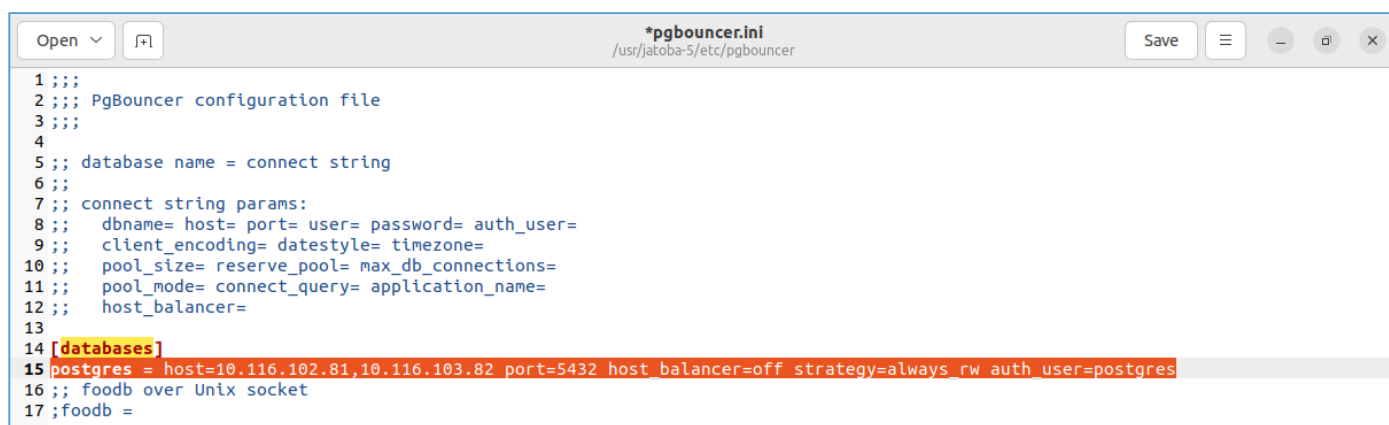


Рисунок 7.2 – Строка переподключений по Public IP

Публичные IP – адреса (Public IP) используются для отправки запросов от клиентских приложений к сегментам кластера. В случае выхода из строя основного дата-центра, запросы перенаправляются на Public IP резервного дата-центра.

В строке указывается стратегия балансировки «strategy=always_rw». В таком режиме все запросы на чтение и на запись отправляются на узел с ролью Master.

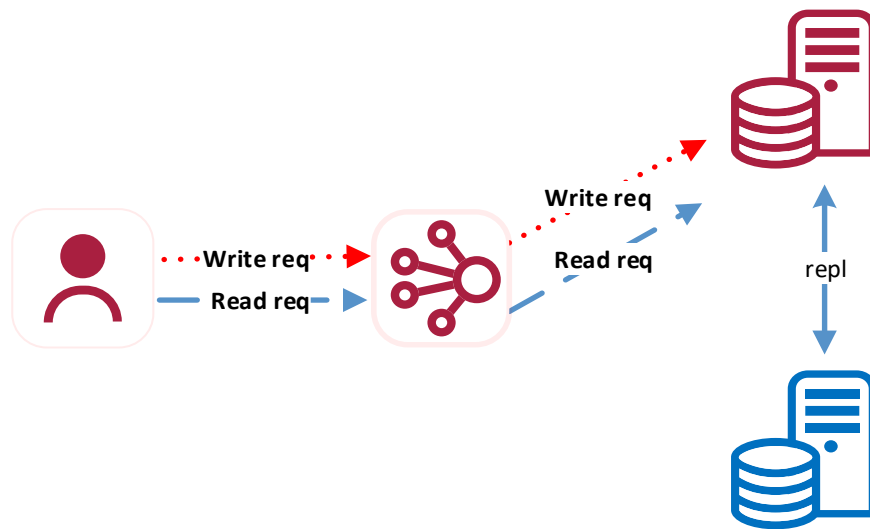


Рисунок 7.3 – Схема работы при установленной стратегии «always_rw»

Активировать строку запроса для извлечения пароля пользователя из базы данных:

```
auth_query = SELECT username, passwd FROM pg_shadow WHERE
username=$1
```

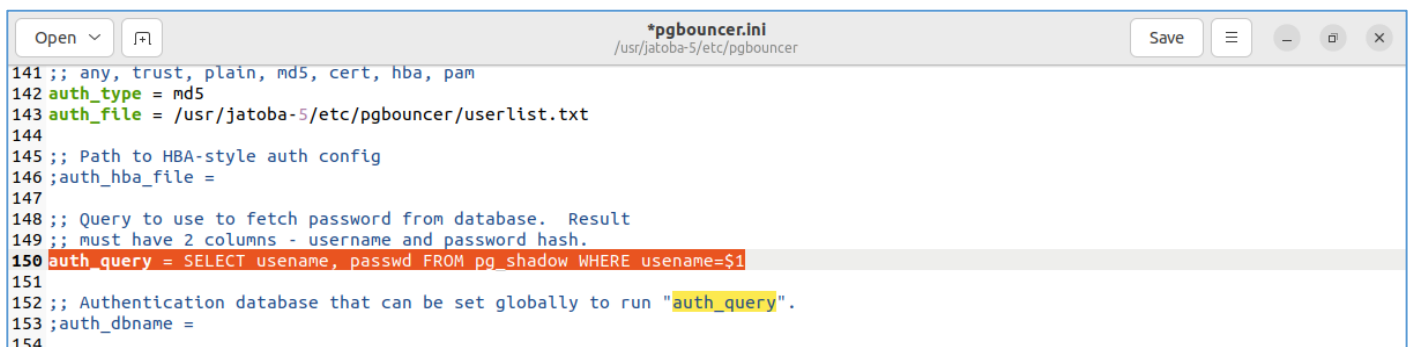


Рисунок 7.4 – Строка запроса для извлечения пароля пользователя из БД

Активировать строку с указанием базы данных для аутентификации:

```
auth_dbname = postgres
```

```

141 ;; any, trust, plain, md5, cert, hba, pam
142 auth_type = md5
143 auth_file = /usr/jatoba-5/etc/pgbouncer/userlist.txt
144
145 ;; Path to HBA-style auth config
146 auth_hba_file =
147
148 ;; Query to use to fetch password from database. Result
149 ;; must have 2 columns - username and password hash.
150 auth_query = SELECT username, passwd FROM pg_shadow WHERE username=$1
151
152 ;; Authentication database that can be set globally to run "auth_query".
153 auth_dbname = postgres
154

```

Рисунок 7.5 – Строка с указанием БД для аутентификации

На данном шаге настройка конфигурационного файла закончена. В конфигурационном файле компонента jaPooler /usr/jatoba-<ver>/etc/userlist.txt указываются пользователи, имеющие право доступа к СУБД в дата-центрах.

В строке указывается:

- имя пользователя;
- метод аутентификации слитно с хешем пароля по 128-битному алгоритму хеширования MD5.

Хеш пароля вычисляется в специализированном калькуляторе. И в строке кодировки потребуется указать пароль и имя пользователя.

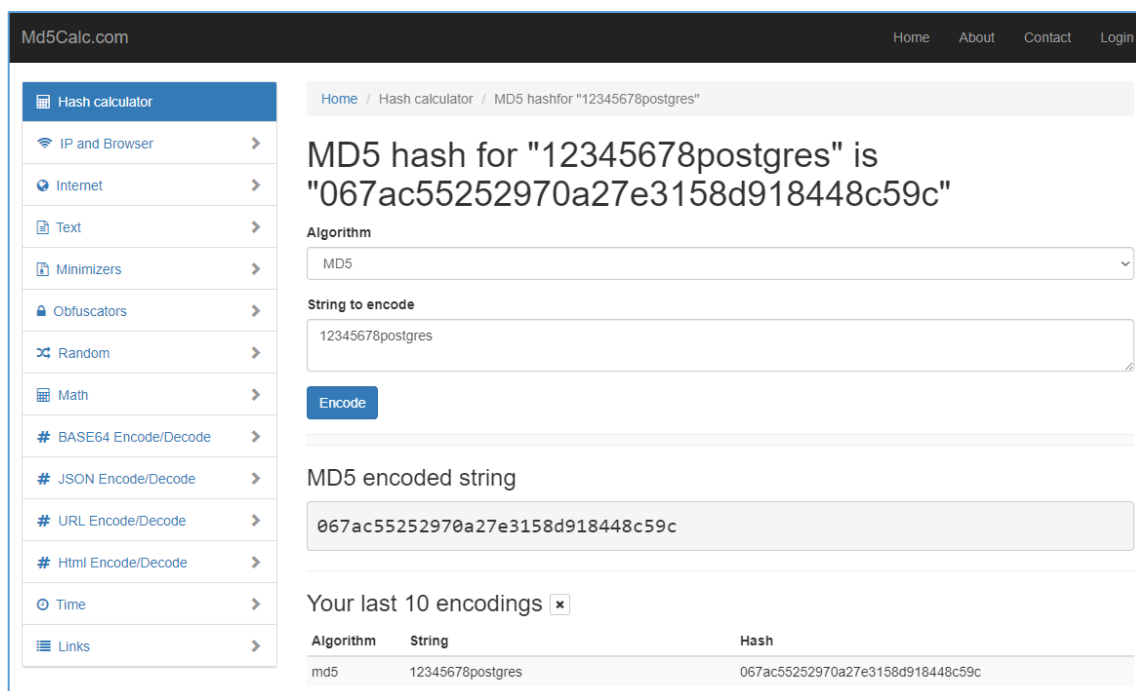


Рисунок 7.6 – Калькулятор хеша паролей

В представленном примере строка имеет следующий вид:

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

```
"postgres" "md5067ac55252970a27e3158d918448c59c"
```

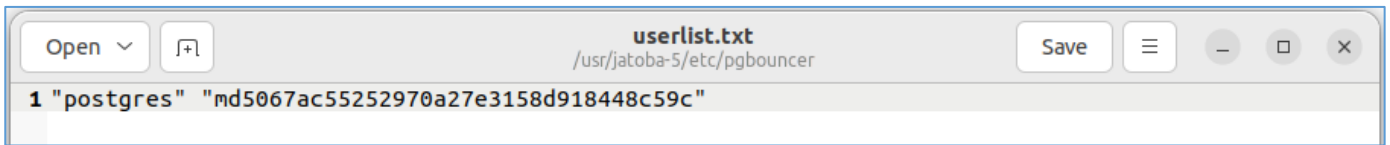


Рисунок 7.7 – Строка пароля пользователя

7.4. Запуск сервиса компонента

Постоянная работа компонента обеспечивается установкой и загрузкой службы в автозагрузку ОС.

Установка и загрузка службы в автозагрузку ОС выполняется в терминале от имени и с правами привилегированного пользователя командами:

```
systemctl start pgbouncer
systemctl status pgbouncer
systemctl enable pgbouncer
```

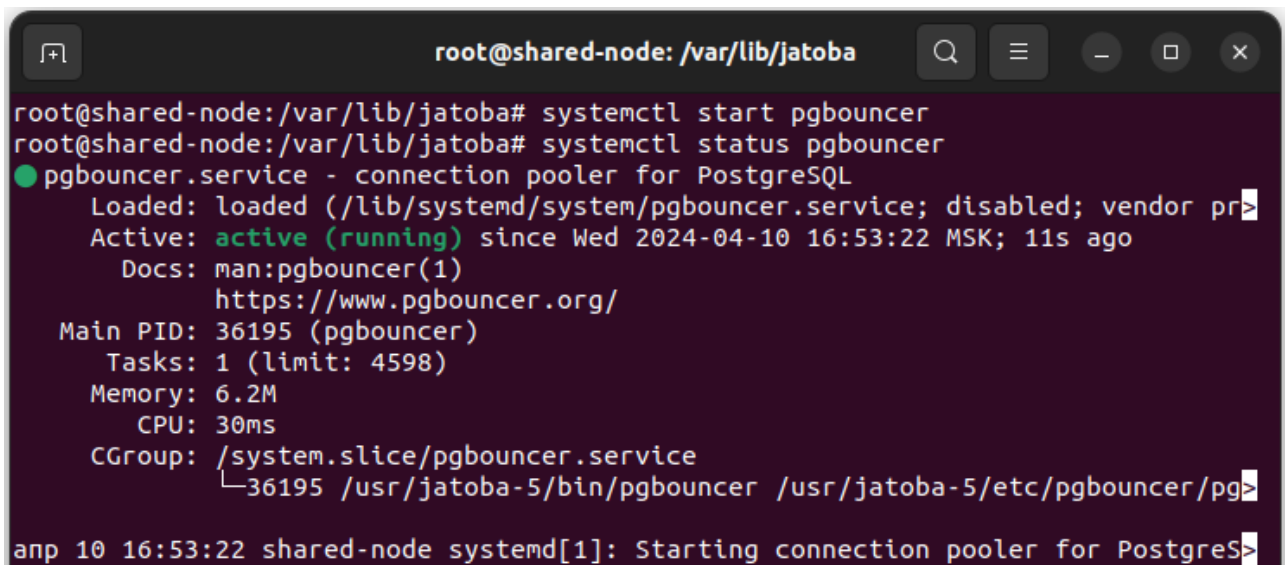


Рисунок 7.8 – Запуск службы компонента

На данном этапе настройка геораспределенного, отказоустойчивого кластера (решение JA_DTC_AS) закончена.

Схема переадресации запросов при отказе одного из дата-центров представлена на рисунке 7.9.

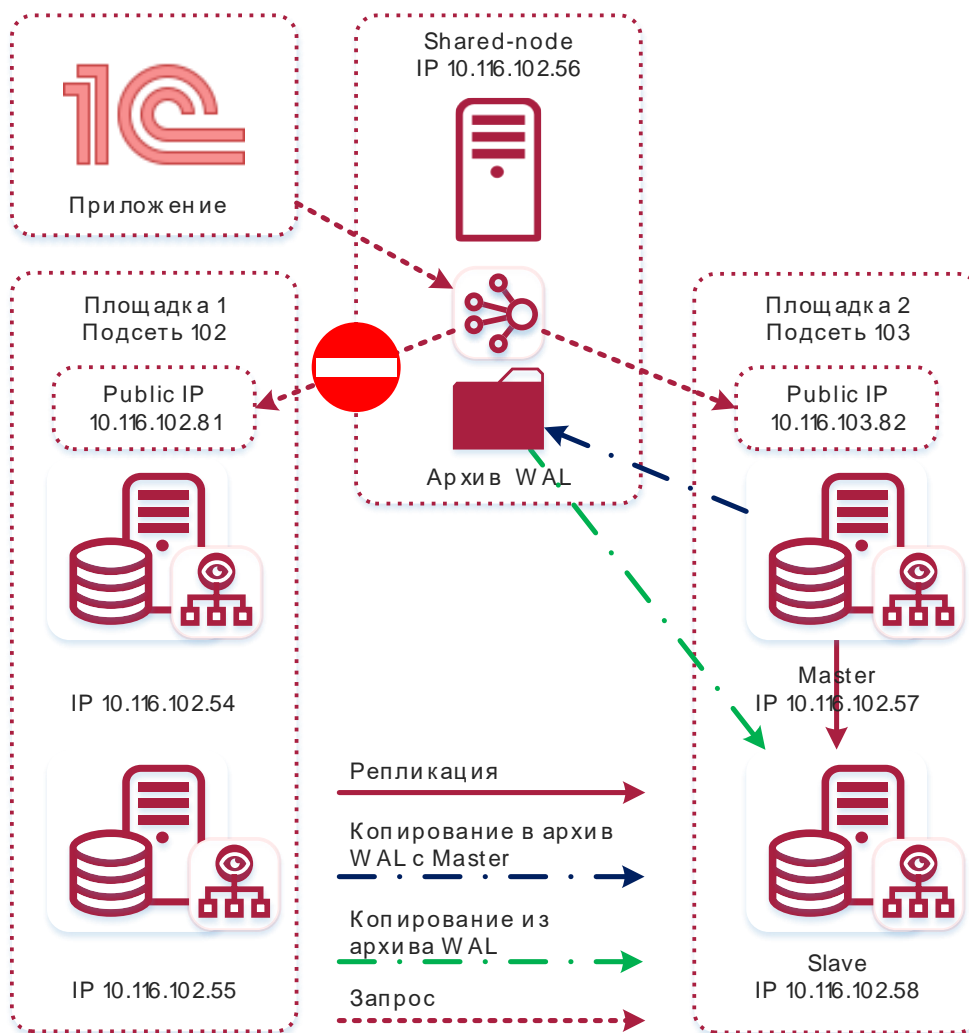


Рисунок 7.9 – Схема переадресации запросов при отказе одного из дата-центров

8. ПРИМЕНЕНИЕ REST API

Компонент «jaDog» предоставляет программный интерфейс REST API для получения информации о состоянии кластера узлов, а также для управления этими узлами.

С целью документирования функционала интерфейса REST API компонента «jaDog» в каталоге /usr/jatoba-6/share/docs размещается файл api.yml. В данном файле перечислены все реализуемые методы для взаимодействия компонента «jaDog» с внешними информационными системами.

Перечень методов REST API, их назначение, а также соответствие им команд консольной утилиты «jadowctl» перечислены в таблице 8.2.

8.1. Настройка функционала REST API

Включение функционала REST API в компоненте «jaDog» возможно осуществить при помощи одного из способов:

- редактирование файла «jadow.yml»;
- настройка параметров через использование консольной утилиты «jadow» с параметром setup;
- использование файла ответов при автоматизированной настройке компонентов и узлов кластера.

8.1.1. Настройка параметров REST API через консольную утилиту «jadow»

Подробная настройка параметров управления REST API приводится в первой части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-02 в подразделе 6.7.10. Пункт меню 10 «REST API Settings».

8.1.2. Настройка параметров REST API в файле ответов

Настройка параметров REST API в файле ответов необходима при выполнении автоматизированной настройки компонентов и узлов кластера.

Для активации и определения параметров REST API в файле ответов необходимо добавить в секцию default_node_params следующий раздел:

```
rest_api:
  api_use: true
  cert_file: /var/lib/jatoba/ssl_jadow/jadow_user.crt
```

```
key_file: /var/lib/jatoba/ssl_jadog/jadog_user.key  
ca_file: /var/lib/jatoba/ssl_jadog/root.crt  
listen_address: 10.116.102.54/24  
listen_port: 54443
```

Здесь <dir> - полный путь к каталогу с сертификатами SSL, 10.116.102.54/24 – пример IP-адреса, на котором будут приниматься подключения.

8.2. Проверка функционала REST API

Для того чтобы внесенные в файл конфигурации «jadog.yml» изменения вступили в силу необходимо перезагрузить сервис компонента «jaDog» при помощи команды:

```
systemctl restart jadog
```

После перезагрузки сервиса «jadog» компонента «jaDog» необходимо проверить работоспособность REST API.

Для этого в консоли выполняется команда вывода сетевой статистики:

```
netstat -ln
```

В результате выполнения команды отобразится таблица:

```
Active Internet connections (only servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State  
tcp        0      0 10.116.102.54:54443     0.0.0.0:*               LISTEN
```

Где «10.116.102.54:54443» - IP-адрес и сетевой порт, на котором будут приниматься соединения по REST API.

8.2.1. Проверка обработки запросов к REST API

После чего можно проверить обработку запросов к REST API через утилиту curl при помощи команды:

```
curl -X GET --cacert /var/lib/jatoba/ssl_jadog/root.crt --cert  
/var/lib/jatoba/ssl_jadog/jadog_service.crt --key  
/var/lib/jatoba/ssl_jadog/jadog_service.key -k  
https://host_name(ip):54443/cluster/[cluster_name]
```

В результате выполнения команды в терминале будет отображена информация о составе и состоянии кластера, и входящих в него узлов, аналогично выполнению команды cluster status.

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

После этого настройку сервера REST API в компоненте «jaDog» можно считать завершенной.

8.2.2. Коды возврата состояния jaDog REST API

Коды состояний REST API предназначены для стандартизированного информирования клиентских систем (ИС) о результате выполнения запросов к компоненту jaDog.

Коды состояний REST API обеспечивает однозначную интерпретацию исхода выполнения команд компонента «jaDog».

При отправке запросов через REST API на выполнение команд возвращаются стандартные коды состояния, указанные в таблице 8.1.

Таблица 8.1– Коды состояний REST API компонента «jaDog»

| Код состояния REST API | Описание |
|------------------------|---|
| 200 | ОК – запрос компонента выполнен успешно. Код «200» возвращается только в том случае, если операция выполнена полностью, без каких-либо отклонений от ожидаемого результата. Все требуемые действия завершены, состояние системы соответствует описанному в спецификации API для данной операции |
| 201 | Created – в результате успешного выполнения запроса был создан пользователь, задача планировщика, узел, дата-центр или кластер. Код «201» возвращается только в том случае, если операция выполнена полностью, без каких-либо отклонений от ожидаемого результата |
| 202 | Accepted – запрос к компоненту принят на обработку. |
| 500 | Internal Server Error – запрос компонента выполнен с ошибками (неверные значения параметров). Код «500» возвращается в ситуациях, когда выполнение операции было прервано из-за исключительной ситуации, сбоя в логике работы компонента «jaDog», невозможности доступа к необходимым внутренним ресурсам или иной непредусмотренной ошибки на стороне сервера |

8.2.3. Результат выполнения запросов к REST API

При выполнении запроса атрибутов через REST API компонент может содержать следующую информацию:

– err_code – код возвращаемой ошибки;

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

- id – идентификатор запроса;
- jadog_version – версия компонента «jaDog»;
- message – текст сообщения, содержит результат выполнения запроса;
- protocol_version – версия протокола компонента «jaDog»;
- result_state_id – идентификатор результата запроса;
- state – статус;
- non_changed_params – флаг измененного/неизмененного параметра.

8.3. Функционал обработки и выполнения команд в утилите «jadog_ctl»

Команды, приведенные в данном подразделе, используются для автоматизации вывода результатов выполнения команд в утилите «jadog_ctl».

8.3.1. Настройка асинхронной обработки команд (set async mode)

Настройка асинхронной обработки команд выполняется командой:

```
set async mode
```

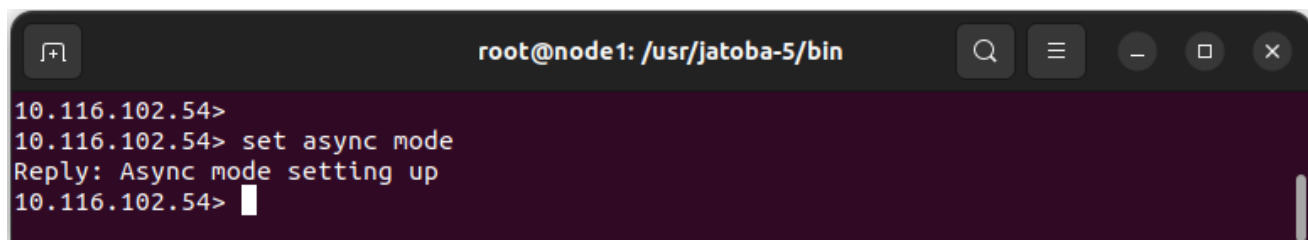


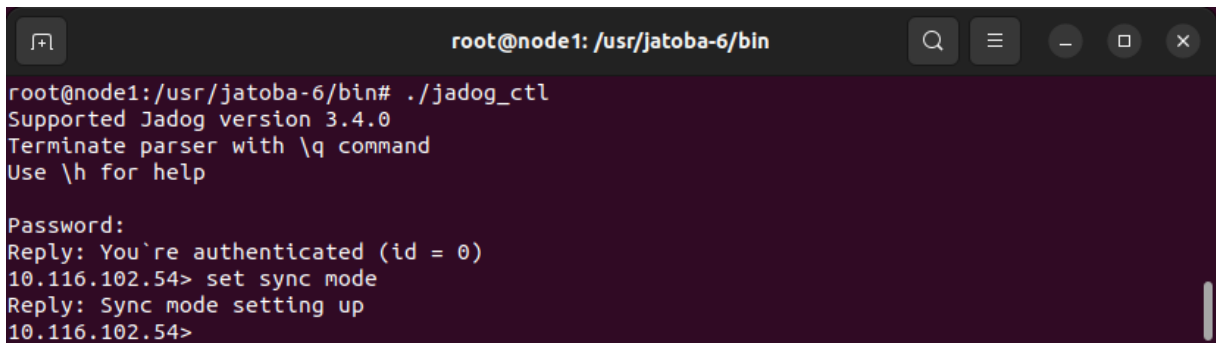
Рисунок 8.1 – Вывод результата выполнения команды «set async mode»

Применение команды описано в п. 6.1.6 «Нештатный обратный DC_Promote» второй части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-02 и применяется на узле кластера с ролью «Master».

8.3.2. Настройка синхронной обработки команд (set sync mode)

Настройка синхронной обработки команд выполняется командой:

```
set sync mode
```



```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.4.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> set sync mode
Reply: Sync mode setting up
10.116.102.54>
```

Рисунок 8.2 – Вывод результата выполнения команды «set sync mode»

8.3.3. Результат последней выполненной команды (get last response)

Команда «get last response» выдает результат последней выполненной команды, исключая команды статуса:

```
get last response
```

Команда применяется для вывода результатов команд, описанных в пунктах документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01:

- «Создание кластера (add cluster 'name')»;
- «Добавление узла кластера в качестве главного (cluster add master ip)»;
- «Удаление узла кластера (cluster delete node)»;
- «Принудительная смена ролей серверов (switchover)»;
- «Принудительное переключение на текущий дата-центр (datacenter promote)».

8.3.4. Получение результата выполнения (get result)

В консольном и терминальном режиме возможно вывести результат выполнения команды по ее идентификатору (id).

Команда имеет синтаксис:

```
get result [id]
```

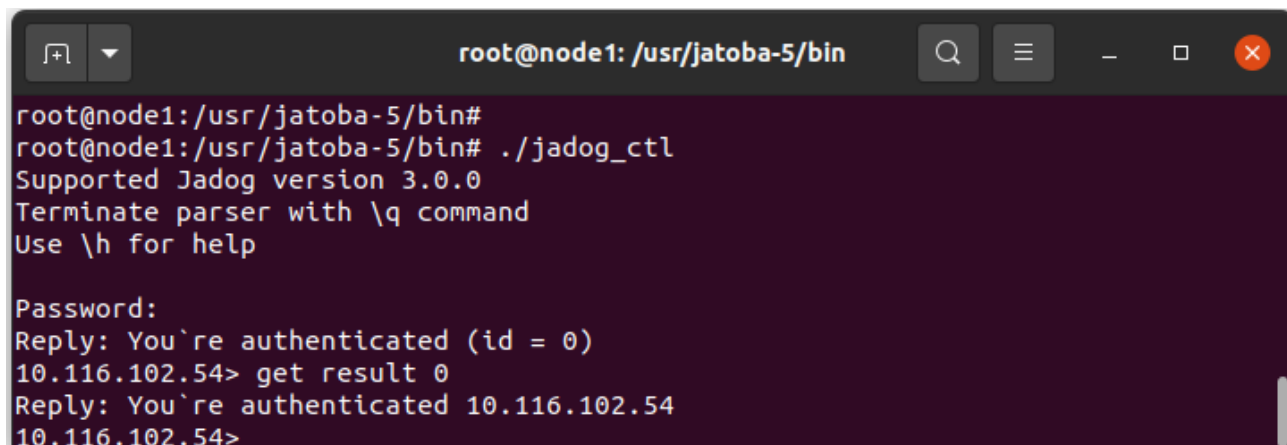
Например

Аутентифицируемся в утилите «jadog_ctl».

Утилита «jadog_ctl» выведет идентификатор операции, например id = 0. Используя значения id = 0 в команде:

```
get result 0
```

будет выведен результат указанной команды без ее выполнения.



```
root@node1: /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin# ./jadog_ctl
Supported Jadog version 3.0.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You`re authenticated (id = 0)
10.116.102.54> get result 0
Reply: You`re authenticated 10.116.102.54
10.116.102.54>
```

Рисунок 8.3 – Получение результата выполнения команды

8.4. Перечень REST API команд и соответствие аналогам утилиты «jadog_ctl»

Таблица 8.2 – Описание соответствия REST API команд с аналогами утилиты «jadog_ctl»

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|--|----------|--|--|
| Команды для работы с параметрами компонента «jaDog» | | | | |
| protocol version | Получение информации о версии компонента «jaDog» | GET | Отсутствуют | /version |
| reload jadog | Выполнить перезагрузку файлов конфигурации компонента «jaDog» (на главном узле или на узле, если он не в составе кластера) | POST | Отсутствуют | /reload |
| Отсутствует | Получение списка параметров кластера | GET | { "cluster_name": "" } | /cluster/{cluster_name}/parameters |
| show parameter [param:name] | Получение значения параметра | GET | { "cluster_name": "", "param:name": "" } | /cluster/{cluster_name}/parameters?parameters=["param:name"] |
| set parameters | Изменение набора параметров узла кластера | PATCH | { "parameters": [{"param1": "value1"}, {"param2": "value2"}] } | /cluster/{cluster_name}/set/parameters |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|--|----------|--|---|
| Команды для работы с группой кластеров (bundle) | | | | |
| bundle create ['bundle_name'] | Создание новой группы кластеров | PUT | { "bundle_name": "" } | /bundle?bundle_name= {bundle_name} |
| bundle delete ['bundle_name'] | Удаление существующей группы кластеров | DELETE | { "bundle_name": "" } | /bundle/{bundle_name} |
| bundle attach cluster [ip] [port] | Присоединение к группе кластеров существующего кластера по его IP-адресу, сетевому порту | POST | { "bundle_name": "", "ip": "", "port": "" } | /bundle/{bundle_name}/cluster/{ip} /{port}/interconnect_user/{inter connect_user} |
| bundle attach cluster {ip} {port} [interconnect_us er] | Присоединение к группе кластеров существующего кластера по его IP-адресу, сетевому порту и технологической записи администратора | POST | { "bundle_name": "", "ip": "", "port": "", "interconnect_user": "" } | /bundle/{bundle_name}/cluster/{ip} /{port}/interconnect_user/{inter connect_user} |
| bundle detach cluster {ip} {port} | Отсоединение от группы кластеров существующего кластера по его IP- | DELETE | { "bundle_name": "", "ip": "", "port": "" } | /bundle/{bundle_name}/cluster/{ip} /{port} |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|---|----------|---|--|
| | адресу, сетевому порту | | | |
| bundle detach cluster {'cluster_name'} | Отсоединение от группы кластеров существующего кластера по его названию | DELETE | { "bundle_name": "", "cluster_name": "" } | /bundle/{bundle_name}/cluster/{cluster_name} |
| bundle status | Отображение сведений о группе кластеров | GET | | /bundle/status |
| node show ['node_name'] | Отображение текущего название узла (ноды) | GET | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name} |
| Команды для работы с кластером и его узлами | | | | |
| add cluster | Создание нового кластера | PUT | { "cluster_name": "" } | /cluster?cluster_name={cluster_name} |
| cluster status | Получение описания кластера с текущими статусами узлов | GET | { "cluster_name": "" } | /cluster/{cluster_name} |
| cluster delete | Удаление кластера | DELETE | { "cluster_name": "" } | /cluster/{cluster_name}/delete |
| cluster activate | Включение кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/activate |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|---|---|----------|---|--|
| cluster deactivate | Выключение кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/deactivate |
| reload jadog on cluster | Перезагрузка файлов конфигурации всех компонентов «jaDog» в кластере | POST | { "cluster_name": "" } | /cluster/{cluster_name}/jadog/reload |
| set maintenance | Включение режима технического обслуживания кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/maintenance |
| reset maintenance | Выключение режима технического обслуживания кластера | DELETE | { "cluster_name": "" } | /cluster/{cluster_name}/maintenance |
| cluster add master [ip] [port] | Добавление в кластер главного узла по IP-адресу и сетевому порту | PUT | { "cluster_name": "", "ip": "", "port": "", "role": "" } | /cluster/{cluster_name}/node/{ip}/{ip}/port/{port}?role=master |
| cluster add master [ip] [port] as [node_name] | Добавление в кластер главного узла по IP-адресу и сетевому порту с указанием названия | PUT | { "cluster_name": "", "node_name": "", "ip": "", "port": "", "role": "" } | /cluster/{cluster_name}/node/{ip}/{ip}/port/{port}?role=master&node_name={node_name} |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|---|----------|---|--|
| cluster add [role] [sync] [ip] [port] | Добавление в кластер резервного узла по IP-адресу и сетевому порту с указанием типа репликации и названия | PUT | { "cluster_name": "", "node_name": "", "ip": "", "port": "", "role": "", "sync": "", "primary_ip": "", "primary_port": "", "primary_name": "" } | /cluster/{cluster_name}/node/{ip}/{ip}/port/{port}?role=slave&node_name={node_name}&sync={sync asyn c} |
| cluster add [role] [ip] [port] as [node_name] | Добавление в кластер узла арбитра по IP-адресу и сетевому порту с указанием и названия | PUT | { "cluster_name": "", "node_name": "", "ip": "", "port": "", "role": "" } | /cluster/{cluster_name}/node/{ip}/{ip}/port/{port}?role=referee&node_name={node_name} |
| cluster delete node [ip] [port] | Исключение узла из кластера по его IP-адресу и сетевому порту | DELETE | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip}/{port} |
| reload jadog [ip] [port] | Перезагрузка компонента «jaDog» на узле по его IP-адресу и номеру сетевого порта | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip}/{port}/jadog/reload |
| set maintenance on node [ip] [port] | Установка режима технического обслуживания на узле кластера по его IP- | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip}/{port}/maintenance |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|---|--|----------|--|--|
| | адресу и сетевому порту | | | |
| reset maintenance on node [ip] [port] | Снятие режима технического обслуживания на узле кластера по его IP- адресу и сетевому порту | DELETE | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip} /{port}/maintenance |
| reload jadog ['node_name'] | Перезагрузка компонента «jaDog» на узле по его названию | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name}/jadog/reload |
| set maintenance on node ['node_name'] | Установка режима технического обслуживания на узле кластера по его названию | POST | { "cluster_name": "", " node_name ": "" } | /cluster/{cluster_name}/node/{node_name}/maintenance |
| reset maintenance on node [ip] [port] | Снятие режима технического обслуживания на узле кластера по его названию | DELETE | { "cluster_name": "", " node_name ": "" } | /cluster/{cluster_name}/node/{node_name}/maintenance |
| reload jadog | Выполнение перезагрузки файлов конфигурации компонента «jaDog» (на главном узле или | POST | Отсутствуют | /reload |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|--|----------|---|---|
| | на узле, если он не в составе кластера) | | | |
| cluster get structure [short full] | Получение структуры кластера в кратком/полном формате в файле YML | GET | { "cluster_name": "", "format": "" } | /cluster/{cluster_name}/structure / |
| cluster reinit node ['node_name'] | Повторная инициализация узла в кластере | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name}/reinit |
| Команды для работы с дата-центрами | | | | |
| datacenter create ['dc_name'] | Создание нового дата-центра | PUT | { "dc_name": "" } | /datacenter?dc_name={dc_name} |
| datacenter delete | Удаление существующего дата-центра | DELETE | { "dc_name": "" } | /datacenter/{dc_name} |
| datacenter ['dc_name'] attach node [ip] [port] | Присоединение узла по его IP-адресу и сетевому порту к дата-центру | POST | { "dc_name": "", "ip": "", "port": "" } | /datacenter/{dc_name}/node/{ip}/{port}/attach |
| datacenter ['dc_name'] | Отсоединение узла по его IP-адресу и сетевому порту от дата-центра | POST | { "dc_name": "", "ip": "", "port": "" } | /datacenter/{dc_name}/node/{ip}/{port}/detach |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|---|----------|--|---|
| detach node [ip] [port] | | | | |
| datacenter ['name'] attach node | Присоединение узла по его названию к дата-центру | POST | { "dc_name": "", "node_name": "", } | /datacenter/{dc_name}/node/{node_ name}/attach |
| datacenter ['name'] detach node | Отсоединение узла по его названию от дата- центра | POST | { "dc_name": "", "node_name": "", } | /datacenter/{dc_name}/node/{node_ name}/detach |
| datacenter promote | Выполнение операции «switchover» для текущего дата-центра | POST | { "dc_name": "" } | /datacenter/{dc_name}/promote |
| Команды для работы с учетными записями компонента «jaDog» | | | | |
| create user или create user ['name'] with password ['password'] | Создать учетную запись для компонента «jaDog» | PUT | { "user_name": "", "password": "" } | /user?user_name=name&password= password |
| alter user ['user_name'] rename to ['new name'] или alter user ['user_name'] with password ['password'] | Изменить название учетной записи или пароль | PATCH | { "user_name": "", "password": "" } | /user/{user_name} |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|---|--|----------|---|--|
| drop user ['user_name'] | Удалить существующую учётную запись | DELETE | { "user_name": "" } | /user/{user_name} |
| alter user ['user_name'] account lock | Заблокировать учетную запись | POST | { "user_name": "" } | /user/{username}/lock |
| alter user ['user_name'] account unlock | Разблокировать учетную запись | POST | { "user_name": "" } | /user/{username}/unlock |
| show users | Получение списка учетных записей компонента «jaDog» | GET | Отсутствуют | /users |
| Команды для работы с узлами кластера | | | | |
| node show ['cluster_name'] | Получение списка узлов, входящих в кластер | GET | { "cluster_name": "" } | /cluster/{cluster_name}/node |
| node show [ip] [port] | Получение информации по IP-адресу и порту об узле, входящего в кластер | GET | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/ip/{ip}/port/{port} |
| alter node 'old_node_name' rename to 'new_node_name' | Переименование узла кластера | PATCH | { "cluster_name": "", "ip": "", "new_node_name": "" } | /cluster/{cluster_name}/node/{ip}/{port} |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|--|----------|--|--|
| alter node ip port rename to 'new_node_name' | Назначение названия узлу кластера | PATCH | { "cluster_name": "", "ip": "", "port": "", "new_node_name": "" } | /cluster/{cluster_name}/node/ip/{ ip}/port/{port} |
| switchover [ip] [port] | Выполнение переключения роли главного узла на указанный узел по его IP-адресу и номеру порта | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip} /{port}/switchover |
| node show ['node_name'] | Получение информации о статусе узла кластера | GET | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{nod e_name} |
| alter node [ip] [port] set replication type = ['replication_ty pe'] | Изменение типа репликации резервного узла кластера | PATCH | { "cluster_name": "", "ip": "", "port": "", "replication_type": "" } | /cluster/{cluster_name}/node/{nod e_name}?replication_type=[sync as ync] |
| switchover ['node_name'] | Переключение роли главного узла на указанный узел по названию | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{nod e_name}/switchover |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|--|--|----------|---|---|
| node is [(m)aster (s)lave (p)rimary (r)eferee] | Получение информации о роли узла в виде значений true/false | GET | master, slave, primary, referee | /node_is?checked_role=master slave primary referee |
| node in [maintenance cascade sync async] | Получение информации о режиме работы узла кластера и состоянии технического обслуживания | GET | maintenance, cascade, sync, async | /node_in?checked_status=maintenance cascade sync async |
| node reinit | Повторная инициализация резервного узла | POST | | /reinitialize |
| node reinit nodata | Повторная инициализация резервного узла в режиме nodata | POST | | /reinitialize?nodata=true |
| cluster add slave nodata [ip] [port] as ['node_name'] | Добавление резервного узла в кластер в режиме nodata | POST | { "cluster_name": "", "role": "", "node_name": "" } | /cluster/{cluster_name}/node/{ip}/{port}?role=slave&nodata=true&node_name={node_name} |
| Команды для работы с СУБД | | | | |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|------------------------------|--|----------|------------------------|--|
| reload dbs on cluster | Перезагрузка файлов конфигурации «jaDog» на узле кластера по его имени | POST | { "cluster_name": "" } | /cluster/{cluster_name}/dbs/reload |
| restart dbs on cluster | Перезагрузка всех СУБД в кластере | POST | { "cluster_name": "" } | /cluster/{cluster_name}/dbs/restart |
| start dbs on cluster | Запуск СУБД во всем кластере | POST | { "cluster_name": "" } | /cluster/{cluster_name}/dbs/start |
| stop dbs on cluster | Остановка СУБД во всем кластере | POST | { "cluster_name": "" } | /cluster/{cluster_name}/dbs/stop |
| reload dbs on node | Перезагрузка файлов конфигурации «jaDog» на главном узле кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/node/dbs/reload |
| restart dbs on node | Перезагрузка СУБД на главном узле кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/node/dbs/restart |
| start dbs on node | Запуск СУБД на главном узле кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/node/dbs/start |
| stop dbs on node | Остановка СУБД на главном узле кластера | POST | { "cluster_name": "" } | /cluster/{cluster_name}/node/dbs/stop |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|---|--|----------|--|---|
| reload dbs on node [ip] [port] | Перезагрузка файлов конфигурации СУБД на узле кластера по его IP адресу и сетевому порту | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip} /{port}/dbs/reload |
| restart dbs on node [ip port] | Перезагрузка СУБД на узле кластера по его IP адресу и сетевому порту | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip} /{port}/dbs/restart |
| start dbs on node [ip] [port] | Запуск СУБД на узле кластера по его IP-адресу и номеру сетевого порта | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip} /{port}/dbs/start |
| stop dbs on node [ip] [port] | Остановка СУБД на узле кластера по его IP-адресу и номеру сетевого порта | POST | { "cluster_name": "", "ip": "", "port": "" } | /cluster/{cluster_name}/node/{ip} /{port}/dbs/stop |
| reload dbs on node ['node_name'] | Перезагрузка файлов конфигурации СУБД на узле кластера по его названию | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name}/jadog/dbs/reload |
| restart dbs on node ['node_name'] | Перезагрузка СУБД на узле кластера по его названию | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name}/dbs/restart |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|---|--|----------|---|--|
| start dbs on node ['node_name'] | Запуск СУБД на узле кластера по его названию | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name}/dbs/start |
| stop dbs on node ['node_name'] | Остановка СУБД на узле кластера по его названию | POST | { "cluster_name": "", "node_name": "" } | /cluster/{cluster_name}/node/{node_name}/dbs/stop |
| Команды для работы с планировщиком заданий | | | | |
| schedule | Отображение списка текущих заданий планировщика | GET | Отсутствуют | /schedule |
| schedule completed | Отображение списка завершённых заданий планировщика | GET | Отсутствуют | /schedule/completed |
| schedule reset [job_name] | Удаление задания из планировщика по его названию | DELETE | { "job_name": "" } | /schedule/reset/{job_name} |
| schedule set [job_name] [job_delay_interval] [job_command] | Добавление задания в планировщик с указанием задержки на выполнение | POST | { "job_name": "", "job_delay_interval": "", "job_command": "" } | /schedule/set/job_delay_interval |
| schedule set [job_name] | Добавление задания в планировщик с указанием даты и времени выполнения | POST | { "job_name": "", "job_time": "", "job_command": "" } | /schedule/set/job_time |

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|---|---|----------|--|--------------------------------|
| {job_time} {job_command} | | | | |
| Команды для работы с синхронизатором файлов кластера | | | | |
| fm | Вывод списка файлов синхронизации | GET | { "cluster_name": "" } | /cluster{cluster_name}/fm |
| fm copy | Синхронизация всех файлов на узлах кластера (кроме Referee) | POST | { "cluster_name": "" } | /cluster{cluster_name}/fm |
| fm copy -n [nodes] | Синхронизация всех файлов на определенных узлах кластера (кроме Referee) | POST | { "cluster_name": "", "nodes": "" } | /cluster{cluster_name}/fm |
| fm copy --referee | Синхронизация всех файлов на узлах кластера, включая узлы Referee | POST | { "cluster_name": "", "referee": "" } | /cluster{cluster_name}/fm |
| fm copy [file id] | Синхронизация указанного файла на всех узлах кластера (кроме Referee) | POST | { "cluster_name": "", "numfile": "", "nodes": "" } | /cluster{cluster_name}/fm |
| fm copy [file id] --referee | Синхронизация указанного файла на всех узлах кластера, включая узлы Referee | POST | { "cluster_name": "", "referee": "", | /cluster{cluster_name}/fm |

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

| Аналог в утилите «jadog_ctl» | Описание команды | Операция | Параметры | Шаблон полной команды REST API |
|------------------------------|--|----------|--|--------------------------------|
| | | | "numfile": "", "nodes": "" } | |
| fm set [file path] | Добавление файла или каталога в список для синхронизации по кластеру | PATCH | { "cluster_name": "", "paths": "" } | /cluster{cluster_name}/fm |
| fm delete | Очистка списка файлов для синхронизации по узлам кластера | DELETE | { "cluster_name": "" } | /cluster{cluster_name}/fm |
| fm delete | Удаление определенного файла из списка для синхронизации по узлам кластера | DELETE | { "cluster_name": "", "numfile": "" } | /cluster{cluster_name}/fm |

9. НАСТРОЙКА ГРУППЫ КЛАСТЕРОВ (BUNDLE) С КОМПОНЕНТОМ «JADOG» В РУЧНОМ РЕЖИМЕ

Под группой (bundle) кластеров понимается совокупность кластеров, управляемых при помощи компонента «jaDog».

Структура инженерного решения в этом случае будет иметь вид:

Группа кластеров:

- Кластер 1;
- Кластер 2;
- Кластер 3 и так далее.

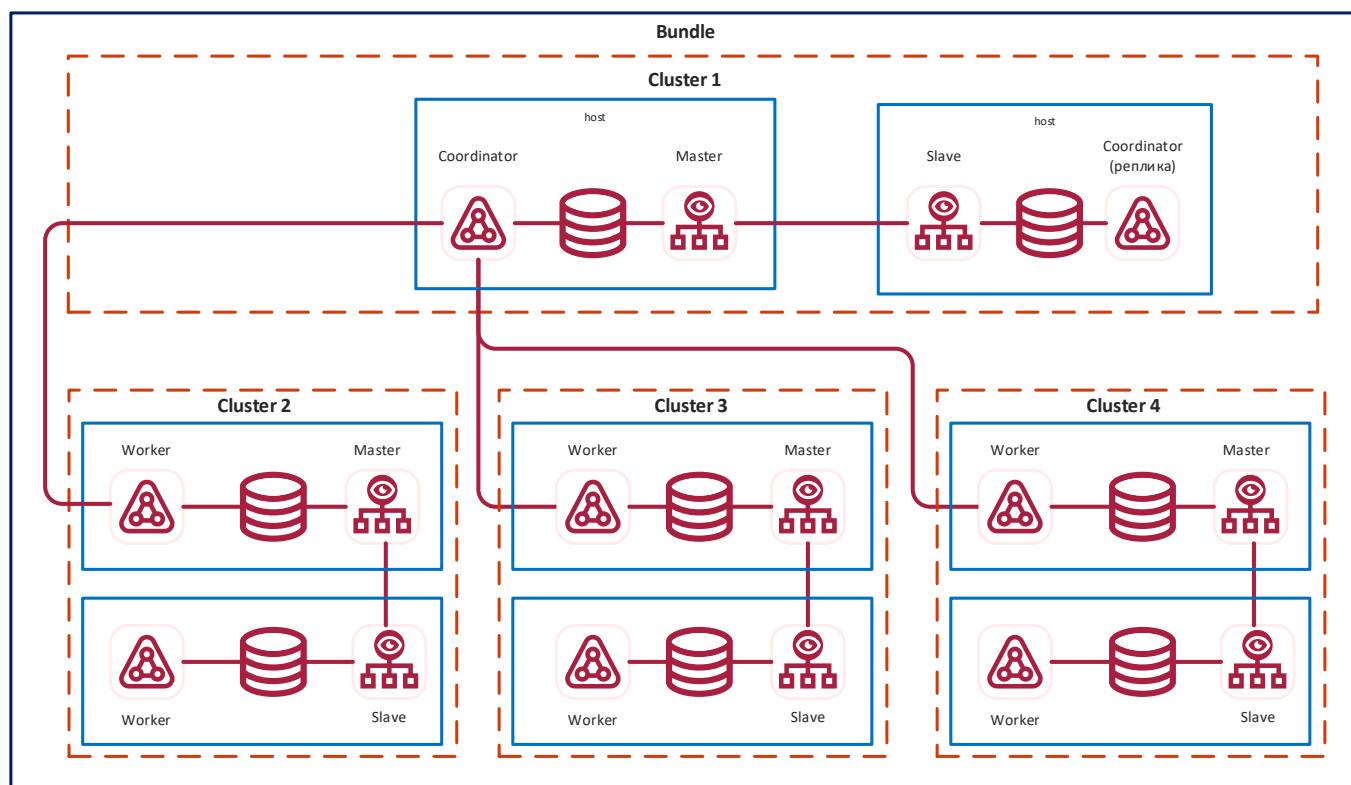


Рисунок 9.1 – Схема формирования и состава группы кластеров

Группа кластеров обеспечивает отказоустойчивость всего высокопроизводительного кластера (например, компонента «ja_Hipe_Cluster» или инженерного решения, построенного на его основе) с учетом состояния каждого подкластера.

Под подкластером (кластером второго уровня) понимается кластер компонента «jaDog», обеспечивающий отказоустойчивость отдельного узла группы кластеров.

Состояние каждого кластера в группе кластеров обеспечивается возможностью регулирования отдельно от группы, но с учетом его состояния. Поведение и работоспособность группы кластеров определяются с учетом состояния каждого кластера, входящего в нее.

В целях обеспечения информационной безопасности все действия администратора кластера или действия ИС через API в обязательном порядке фиксируются в журналах событий.

9.1. Создание группы кластеров

Группа кластеров формируется на основе того кластера (вместе с входящими в него узлами), с которого выполняется его создание.

Для того чтобы создать группу кластеров необходимо выполнить следующие команды:

1) На главном узле подкластера подключиться к консольной утилите «`jadog_ctl`» (здесь и далее команды выполняются в «`jadog_ctl`») и выполнить команду по созданию новой группы кластеров:

```
bundle create ['bundle_name']
```

Пример:

```
bundle create 'bundle0'
```

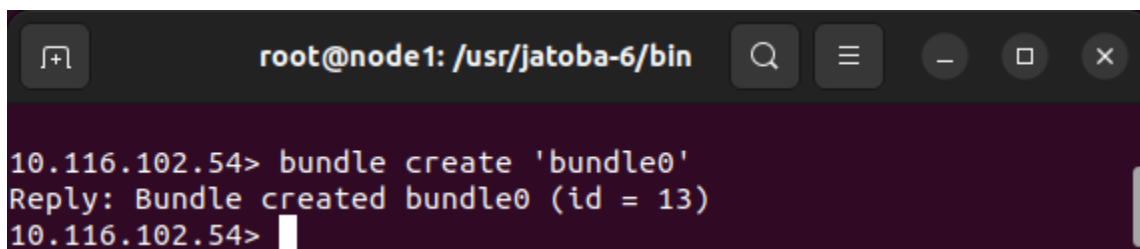
A screenshot of a terminal window with a dark background. The title bar shows 'root@node1: /usr/jatoba-6/bin'. The terminal content shows a prompt '10.116.102.54>' followed by the command 'bundle create 'bundle0''. The response is 'Reply: Bundle created bundle0 (id = 13)'. The prompt '10.116.102.54>' is shown again with a cursor.

Рисунок 9.2 – Вывод результата выполнения команды «`bundle create`»



Название группы кластеров должно соответствовать следующим требованиям:

- длина: от 2 (включительно) до 20 (включительно) символов;
- начинается и заканчивается строчной буквой или цифрой;
- разрешены строчные буквы, цифры, а также "-" и "_";
- нельзя использовать 2 символа "_" и "-" подряд в любой комбинации.

Вывод сообщения «Reply: Bundle created bundle0» соответствует успешному созданию новой группы кластеров.

При создании группы кластеров генерируется уникальный идентификатор – bundle_ulid.

2) После создания группы кластеров необходимо выполнить проверку ее состояния при помощи команды:

```
bundle status
```

```

root@node1: /usr/jatoba-6/bin
10.116.102.54> bundle create 'bundle0'
Reply: Bundle created bundle0 (id = 2)
10.116.102.54> bundle status
Reply: (id = 3)
-----+-----+-----+-----+
|JaDog version|4.1.0|
+-----+-----+-----+-----+

+-----+-----+-----+-----+
|Bundle ulid|Bundle name|Clusters|Nodes|
+-----+-----+-----+-----+
|01KAZND17ZBZDPV8F33FEHBE2|bundle0|1|3/3|
+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Cluster name|Group|Role|Node name|Node|State|Connection state|Replication type|Write LSN|Replication slot name|Slot status|Public IP|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Datacenter: DC_Piter|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|cluster1|-1|UNKNOWN|node1|10.116.102.54():12345|Master(ACTIVE)|true|-|0/70003A0|-|false|10.116.102.81|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|cluster1|-1|UNKNOWN|node2|10.116.102.55():12345|Slave(ACTIVE)|true|async|0/70003A0|rs_node2|true|10.116.102.81|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|cluster1|-1|UNKNOWN|node_referee|10.116.102.57():12345|Referee(ACTIVE)|true|-|-|-|-|~|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10.116.102.54>

```

Рисунок 9.3 – Вывод результата выполнения команды «bundle status»

Команда «bundle status» предоставляет следующую информацию:

- версия компонента «jaDog»;
- значение идентификатора Bundle ulid;
- количество кластеров присоединенных к группе;
- количество узлов в группе кластеров;
- название группы кластеров;
- название дата-центра;
- таблица с характеристиками и статусами узлов кластеров, которая дополнительно содержит идентификатор группы и роль узла в кластере ja_Hipe Citus.

На иллюстрации 9.3 видно, что группа кластеров с названием «bundle0» состоит из кластера «cluster1» и его узлов «node1», «node2» и «node_referee».

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

Структуру группы кластера в формате JSON возможно получить при помощи команды:

```
usr/jatoba-6/bin# ./jadog_ctl -h localhost -p 54321 -U [] -W  
[password_user] -C "bundle status" -q -f json
```

```
root@node1: /usr/jatoba-6/bin
root@node1:usr/jatoba-6/bin# ./jadog_ctl -h localhost -p 54321 -U admin -W 12345678 -C "bundle status" -q -f json
{
  "bundle_name": "bundle0",
  "bundle_ulid": "01KAZND17ZBZDPV8F33FEXHBE2",
  "clusters": [
    {
      "activated": true,
      "citrus_group_id": -1,
      "cluster_name": "cluster1",
      "datacenters": [
        {
          "datacenter": "DC_Piter",
          "nodes": [
            {
              "connection_state": true,
              "db_connection_settings": {
                "port": "5432"
              },
              "full_node_state": "Master(ACTIVE)",
              "is_slot_active": false,
              "main": {
                "ip": "10.116.102.54",
                "port": "12345",
                "public_address": "10.116.102.81"
              },
              "node": "10.116.102.54():12345",
              "node_name": "node1",
              "node_state": "ACTIVE",
              "pending_reload": false,
              "pending_restart": false,
              "primary_ip": "",
              "primary_port": "",
              "replication": {
                "replication_slot_name": "rs_node1",
```

Рисунок 9.4 – Получение структуры группы узлов

9.2. Присоединение дополнительных кластеров к группе кластеров

Для того чтобы присоединить группе кластеров дополнительные кластера необходимо выполнить следующее:

- 1) На главном узле кластера, который уже присоединен к группе кластеров, подключиться к консольной утилите «jadog_ctl» и выполнить команду:

```
bundle attach cluster [ip] [port]
```

Пример:

```
bundle attach cluster 10.116.102.56 12345
```

В случае если название технологической учетной записи «Jadog interconnection user» (по умолчанию «admin») в присоединяемом кластере отличается, команда будет иметь следующий синтаксис:

```
bundle attach cluster [ip] [port] ['interconnect_user']
```

Где «interconnect_user» - название технологической учетной записи для взаимодействия между сервисами компонентов «jaDog», а «ip» и «port» - IP-адрес и сетевой порт главного узла присоединяемого кластера.

Пример:

```
bundle attach cluster 10.116.102.56 12345 'admin_cluster2'
```

Название технологической учетной записи задается при выполнении настройки компонентов «jaDog» (см. первую часть документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-01).

2) После присоединения дополнительного кластера необходимо проверить статус группы кластеров и входящих в него кластеров при помощи команды:

```
bundle status
```

9.3. Файл состояния группы кластеров

После создания группы кластеров информация о его структуре и параметрах сохраняется в файл «bundle_state.yml», расположенный в каталоге /usr/jatoba-6/etc/jadog.

Файл состояния группы кластеров «bundle_state.yml» содержит иерархическую структуру, отражающую подчиненность кластеров и узлов (см. рисунок 9.5).

Дополнительно информация о принадлежности кластера к конкретной группе кластеров после успешного присоединения вносится в файл «jadog_state.yml», расположенный в каталоге /usr/jatoba-6/etc/jadog.

```

GNU nano 6.2 /usr/jatoba-6/etc/jadog/bundle_state.yml
bundle:
  bundle_ulid: 01KC3NYBP11H4B6P6K2VRPFH44
  bundle_name: bundle0
  clusters:
    - cluster_name: cluster1
      epoch: 2
      activated: true
      interconnect_user: interdog
      hipec_node_id: -1
      citus_group_id: -1
      datacenters:
        - datacenter: DC_Piter
          sync_count: 0
          nodes:
            - main:
                ip: 10.116.102.54
                port: 12345
                public_address: 10.116.102.81
                state_in_cluster: Master
                node_state: ACTIVE
                replication:
                  replication_slot_name: rs_node1
                node_name: node1
                db_connection_settings:
                  port: 5432
                sync_state: None
                primary_ip: ""
                primary_port: ""
            - main:
                ip: 10.116.102.55
                port: 12345
                public_address: 10.116.102.81
                state_in_cluster: Slave
                node_state: ACTIVE
                replication:
                  replication_slot_name: rs_node2
                node_name: node2
                db_connection_settings:
                  port: 5432
                sync_state: Async
                primary_ip: 10.116.102.54
                primary_port: 12345
            - main:
                ip: 10.116.102.57
                port: 12345

```

Рисунок 9.5 – Содержимое файла состояния группы кластеров «bundle_state.yml»

При изменении состава группы кластеров содержимое файла состояния «bundle_state.yml» обновляется на всех узлах кластеров, входящих эту группу. Таким образом все узлы содержат информацию о составе и состоянии всех узлов, что обеспечивает отказоустойчивость и не противоречивость. В случае обработки отказов (failover), а также при изменении ролей узлов (switchover), информация об этом также обновляется в файлах состояния «bundle_state.yml» на всех узлах группы кластеров.

9.4. Отсоединение кластера от группы кластеров

В случае если присоединенный ранее кластер необходимо вывести из группы это выполняется в следующей последовательности:

- 1) Подключиться к любому из главных узлов кластеров, кроме того, который требуется отсоединить, и при помощи консольной утилиты «jadog_ctl» выполнить команду:

```
bundle detach cluster {ip} {port}
```

Где «ip» и «port» - IP-адрес и сетевой порт главного узла отсоединяемого кластера.

Пример:

```
bundle detach cluster 10.116.102.56 12345
```

2) Другим вариантом отсоединения кластера является использование его названия. В этом случае команда будет иметь следующий вид:

```
bundle detach cluster [cluster_name]
```

Название отсоединяемого кластера возможно узнать при помощи команды «bundle status».

Пример:

```
bundle detach cluster cluster2
```

3) После отсоединения кластера от группы кластеров необходимо проверить ее состояние при помощи команды:

```
bundle status
```

9.5. Удаление группы кластеров

В случае необходимости удаления группы кластеров необходимо предварительно отсоединить от нее все кластеры (см. п.п. 9.4) кроме последнего.

Далее необходимо подключиться к главному узлу последнего кластера, входящего в группу, и в консольной утилите «jadowctl» выполнить следующую команду:

```
bundle delete ['bundle_name']
```

Пример:

```
bundle delete 'bundle0'
```

После выполнения данной команды группа кластеров считается удаленной.

9.6. Особенности взаимодействия компонентов «jaDog» и «ja_Hipe_Cluster»

С целью обеспечения отказоустойчивости высокопроизводительного кластера компонента «ja_Hipe_Cluster» при настройке с помощью компонента «jaDog»

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

администратор должен выполнить настройку узлов (кроме узлов с ролью «Координатор») без использования параметра Public address. В этом случае при возникновении аварийных случаев обеспечивается взаимодействие составных частей кластера непосредственно по IP-адресам.

Предварительным условием выполнения процедур обработки отказа или переключения главного узла(ов) является принадлежность всех узлов кластера компонента «ja_Hipe_Cluster» к одной группе кластеров компонента «jaDog».

Компонент «jaDog» в этом случае использует внутренние механизмы компонента «ja_Hipe_Cluster» в части реализации функций обработки отказа (failover, FO) и процедур принудительной смены ролей серверов (switchover, SO).

9.6.1. Процедура принудительной смены ролей серверов (switchover, SO)

В случае возникновения внештатной ситуации, затрагивающей главный узел кластера с ролью «Master», компонент «jaDog» выполняет переключение этой роли на новый узел (или реплику в терминологии компонента «ja_Hipe_Cluster») с использованием функции «citus_update_node». Таким образом обеспечивается резервирование как отдельной реплики, так и работоспособность репликасета (набора реплик) в рамках одной группы кластеров с использованием встроенного инструментария компонентов «ja_Hipe_Cluster» и «jaDog».

10. НАСТРОЙКА ГРУППЫ КЛАСТЕРОВ С КОМПОНЕНТОМ «JADOG» В АВТОМАТИЧЕСКОМ РЕЖИМЕ

Описание назначения и структура группы кластеров приведены в разделе 9.

Настройка группы кластеров в автоматическом режиме подразумевает использование специального файла ответов в формате YML.

Файл ответов содержит в себе подготовленный пользователем перечень параметров группы кластеров, а также дополнительные атрибуты для других компонентов, например «ja_Hipe_Cluster».

Параметры, указываемые в файле ответов, соответствует проектируемой инфраструктуре группы кластеров.

Предварительные условия:

- Подготовленная инфраструктура (проект и описание структуры группы кластеров);
- На серверах установлена ОС и обеспечена сетевая доступность;
- На серверах при помощи менеджера пакетов АРТ выполнена установка;
 - СУБД «Jatoba». Процедура установки пакетов СУБД приводится в документе «Руководство по установке СУБД Jatoba 6» 643.72410666.00067-07 97 01;
 - Компонента «jaDog»;
 - Компонента «ja_Hipe_Cluster». Процедура установки пакетов компонента приводится в документе «Компонент ja_Hipe_Cluster. Горизонтальное масштабирование» 643.72410666.00067-07 98 01-11
- На серверах запущен специальный «нулевой» режим (jadog0).

10.1. Формирование файла ответов для группы кластеров

Шаблон файла ответов автоматизированного построения группы кластеров содержит в себе все необходимые настройки. Далее готовый файл ответов будет называться «jadog_init_hipe.yml».



Шаблон файла ответов автоматизированного построения группы кластеров «jadog_init_hipe.yml» формируется на основании шаблона, расположенного в

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

директории

/usr/jatoba-

6/share/doc/jadog/clusters_kits/ja_hipe_cluster/init_ja_hipe_cluster.yml

В файле ответов «jadog_init_hipe.yml» каждого параметра приводятся комментарии, описывающие его назначение.

Параметры, заданные в файле ответов на уровне кластера (секция `cluster_settings`), имеют приоритет по сравнению с параметрами на уровне группы кластеров. Если параметр не определён в секции `cluster_settings`, то используются параметры из секции `default_cluster_params`. Пример секции:

```
default_cluster_params:                # Блок атрибутов
используемый для каждого кластера в бандле.
  db_init_conn_string: host=127.0.0.1 port=5432 user=postgres
dbname=postgres password='change_on_install' # Строка подключения к СУБД
для установки расширений и выполнения скриптов. Требуется SU прав.
  initdb:                               # Блок инициализации СУБД.
Содержит параметры инициализации СУБД
  initdb_options: "--locale=ru_RU.utf8 --encoding=UTF-8" # Строка
параметров инициализации СУБД Jatoba.
  postgresql.conf:                     # Параметры будут
установлены при формировании кластера в конец файла postgresql.conf.
  listen_addresses: "*"
  log_destination: "stderr"
  logging_collector: on
  log_directory: "log"
  log_filename: "jatoba-%Y-%m-%d_%H%M%S.log"
  log_rotation_age: 1d
  log_rotation_size: 0
  log_truncate_on_rotation: off
  log_line_prefix: "%m [%p] "
  log_statement: all
  shared_preload_libraries: "citus, pg_stat_statements"
  max_connections: 128
  max_prepared_transactions: 256
  data_sync_retry: off
  pg_hba.conf:                          # Параметры будут
установлены при формировании кластера в файл pg_hba.conf
  - local all postgres scram-sha-256
  - local ja_hipe_cluster jadog_user scram-sha-256
  - host all postgres 10.116.102.0/24 trust
  - host all postgres 127.0.0.1/32 scram-sha-256
  - host ja_hipe_cluster jadog_user 127.0.0.1/32 scram-sha-256
  - host replication jadog_user 127.0.0.1/32 scram-sha-256
  - host replication jadog_user 10.116.102.0/24 scram-sha-256
```

Узел, который является узлом-координатором, определяется через параметр `cordinator_cluster` в файле ответов. Данный параметр расположен в подсекции `ja_hipe_cluster` и является обязательным только при развертывании высоконагруженного кластера

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

компонента «ja_Hipe_Cluster». Координатор высоконагруженного кластера может быть только один. В случае если параметр `cordinator_cluster` в файле ответов не указан, то автоматическое развертывание кластера завершится с ошибкой.

Правила авторизации пользователей, описанные в файле ответов в подсекции `pg_hba.conf`, полностью заменяют собой весь блок правил авторизации в файле `/var/lib/jatoba/6/data/pg_hba.conf`.

```
pg_hba.conf: # Параметры будут установлены при формировании
кластера в файл pg_hba.conf
- local all postgres scram-sha-256
- local ja_hipe_cluster jalog_user scram-sha-256
- host all postgres 10.116.102.0/24 trust
- host all postgres 127.0.0.1/32 scram-sha-256
- host ja_hipe_cluster jalog_user 127.0.0.1/32 scram-sha-
256
- host replication jalog_user 127.0.0.1/32 scram-sha-256
- host replication jalog_user 10.116.102.0/24 scram-sha-256
```

10.2. Запуск автоматической настройки группы кластеров

Предварительно «нулевой» режим (`jalog0`) должен быть обязательно запущен на всех узлах перед выполнением формирования группы кластеров из файла ответов.

Запуск «нулевого» режима (`jalog0`) в с упрощенной небезопасной аутентификацией производится при помощи команды:

```
/usr/jatoba-6/bin/jalog jalog0 --basic -U [temp_user] -W
[temp_password]
```

Где `temp_user` – имя создаваемого временного пользователя; `temp_password` – пароль временного пользователя, состоящий из не менее чем шести символов и одной цифры.

После этого на главном узле с ролью «Master» открыть дополнительное окно/вкладку терминала и выполнить чтение параметров из файла ответов «`jalog_init_hipe.yml`» и развертывание узлов кластеров с перекрестной репликацией:

```
/usr/jatoba-6/bin/jalog_ctl create_cluster -q -f json -T 999999
-c /<dir1>/jalog_init_hipe.yml --basic -U [temp_user] -W
[temp_password]
```

Где <dir1> - путь к каталогу, в котором расположен файл ответов в формате YML
temp_user – имя временного пользователя (см. п. 3.5.1); temp_password – пароль временного пользователя.

Процесс формирования группы кластеров с использованием файла ответов сводится к последовательному выполнению следующих шагов:

- 1) Чтение и разбор файла ответов, содержащего настройки группы кластеров.
- 2) Инициализация СУБД на узлах с использованием значений из параметра initdb_options;
- 3) Установка параметров файлов postgresql.conf и pg_hba.conf;
- 4) Установка в БД расширения компонента «jaDog»
- 5) Установка необходимых расширений с использованием значений из параметра shared_preload_libraries;



В случае если для параметра shared_preload_libraries не указана установка расширения citus группа кластеров будет сформирована без компонента «ja_Hipe_Cluster»

- 6) Запуск БД на узлах кластера;
- 7) Настройка суперпользователя с использованием значений из параметра db_init_conn_string;
- 8) Назначение узла-координатора. Данный узел также должен являться главным узлом (с ролью Master) кластера компонента «jaDog».
- 9) Выполнение настройки узлов компонента «jaDog» в группе кластеров.
- 10) (Опционально) Выполнение настройки узлов компонента «ja_Hipe_Cluster» в группе кластеров – определение узла-координатора на рабочих узлах, добавление рабочих узлов в высокопроизводительный кластер.
- 11) По завершению формирования группы кластеров возможно выполнить SQL-скрипты уровня группы кластеров или входящих в нее кластеров.



В случае возникновения ошибок при автоматической настройке кластера производится откат всех изменений. Событие при этом записывается в журнал компонента «jaDog» с перечнем причин остановки.

В случае возникновения ошибок при автоматической настройке нескольких кластеров откат изменений производится только для кластера, в процессе создания которого произошла ошибка.

В обоих случаях необходимо повторно запустить процедуру чтения параметров из файла конфигурации и развертывание узлов кластера(ов).

Остановка формирования группы кластеров может прерываться в следующих случаях:

- не назначен узел-координатор в параметре `cordinator_cluster`;
- параметр `bundle.citus.cordinator_cluster` установлен более чем для одного узла;

ПРИЛОЖЕНИЕ 1

Пример монтирования сетевого каталога для настройки архивирования WAL

В рассматриваемом примере монтирования сетевого каталога для конфигурирования отказоустойчивого кластера «jaDog» на ОС Ubuntu 20.04, 22.04 используются параметры сети кластера, приведенные в таблице 1.1.

Таблица 1.1 – Конфигурация сети кластера с файловым сервером архива WAL

| № | Имя сервера | Подсеть кластера | Точка монтирования | Каталог архивов WAL |
|---|-------------|------------------|--------------------|---------------------|
| | | 10.116.102.0/24 | | |
| | | IP-адрес | | |
| 1 | Node1 | 10.116.102.54/24 | /nfs/arhive_wal | |
| 2 | Node2 | 10.116.102.55/24 | /nfs/arhive_wal | |
| 3 | Shared-node | 10.116.102.56/24 | | /var/nfs/arhive_wal |

Использование функциональной возможности архивирования WAL может применяться в таких инженерных решениях, как катастрофоустойчивый кластер и геораспределенный кластер, основанный на компоненте «jaDog».

Инфраструктура кластера, помимо узлов кластера, требует выделенный сетевой ресурс (файловый сервер) с каталогом, доступным для записи и чтения. Имя каталога и IP-адрес не должны изменяться.

Каждый из узлов кластера должен иметь смонтированный сетевой диск, автозагружаемый при перезагрузке операционной системы.

Структурная схема кластера с файловым сервером представлена на рисунке 1.1.

Такая структура позволяет узлу кластера с функцией Master записывать архив WAL на сетевой ресурс, а другим узлам «подхватывать» недостающие файлы из архива.

При смене ролей узлов кластера внешний архив и идентичные настройки работы с ним позволяют оперативно и с консистентными данными обеспечивать бесперебойную работу пользователей.

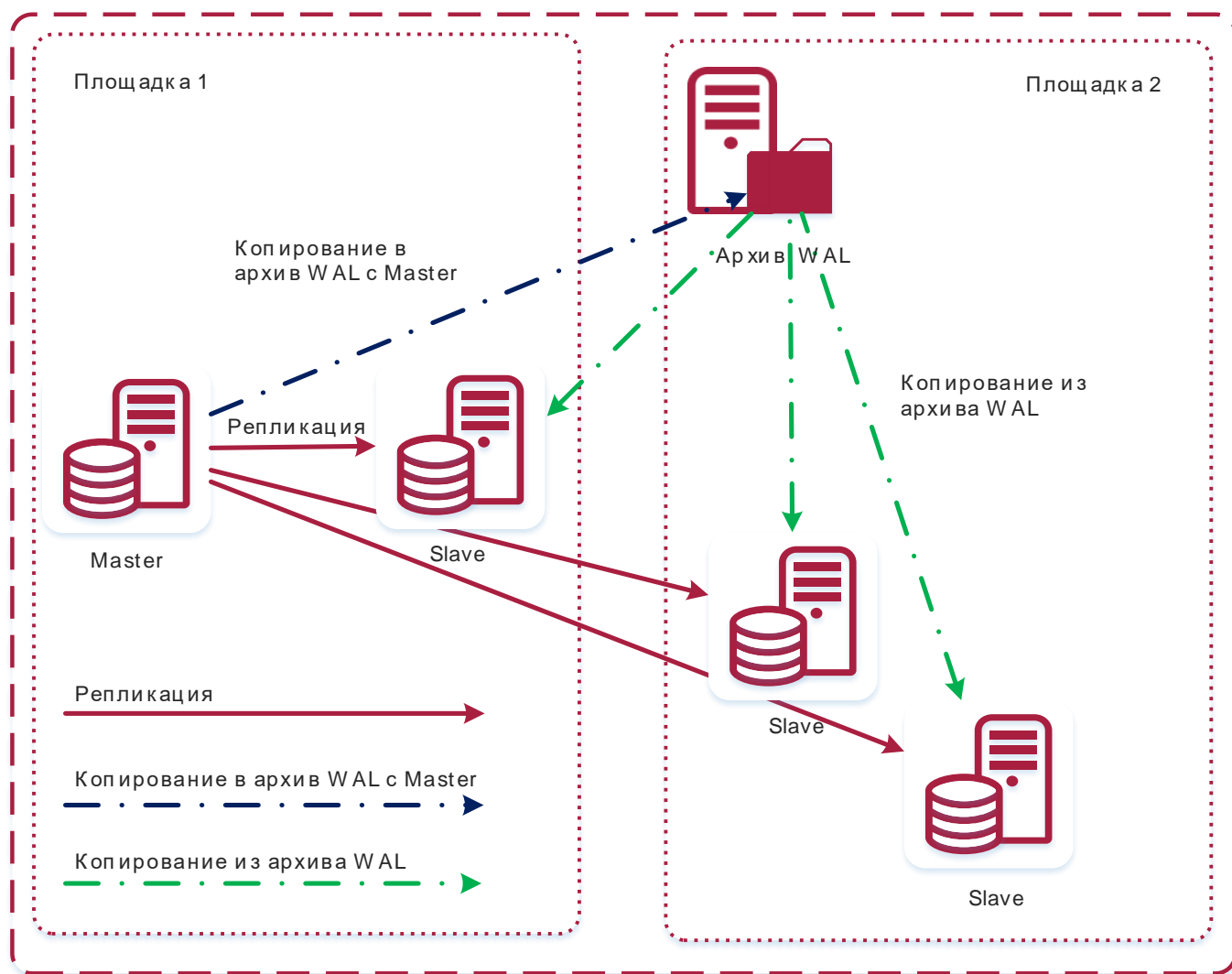


Рисунок 1.1 – Структурная схема кластера с файловым сервером

Настройка монтирования NFS

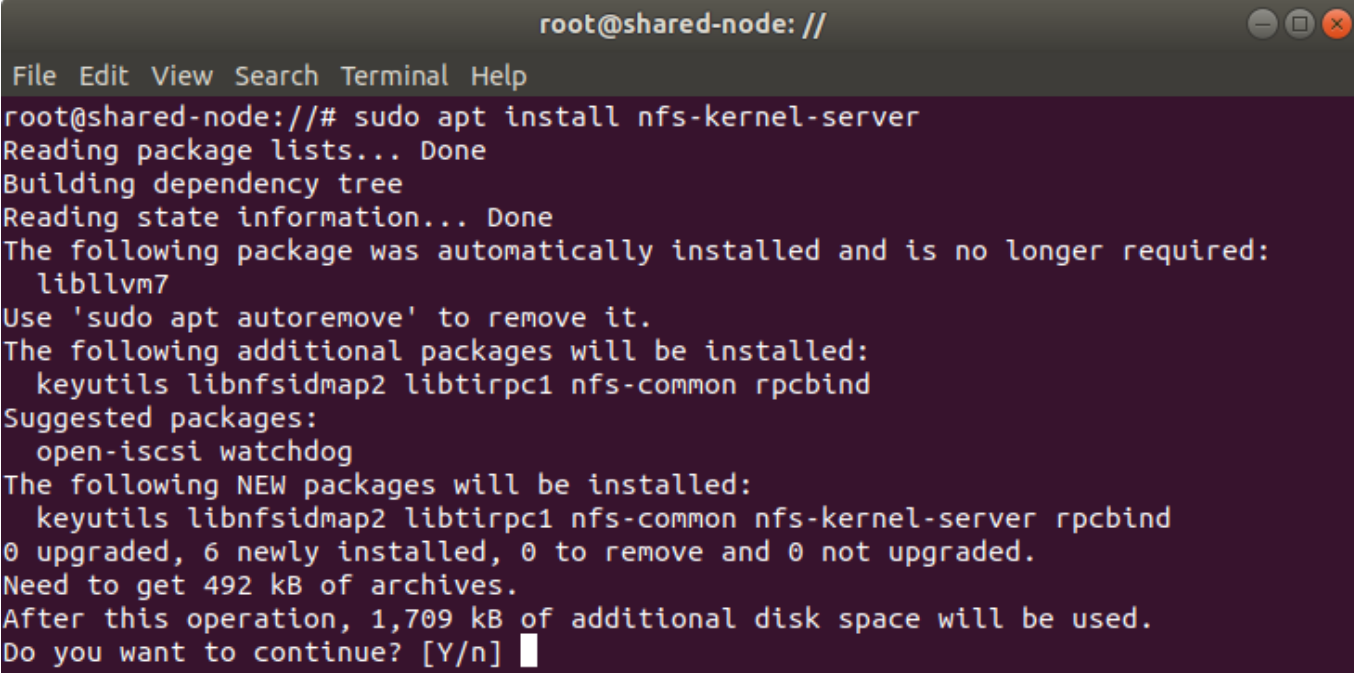
NFS (сетевая файловая система) — это распределенный протокол файловой системы, позволяющий монтировать на сервер удаленные каталоги. Это позволяет управлять пространством хранения в другом месте и выполнять запись в это пространство с нескольких клиентов. NFS обеспечивает относительно стандартизированный и производительный способ доступа к удаленным системам через сеть и хорошо работает в ситуациях, когда требуется регулярный доступ к общим ресурсам.

Загрузка и установка компонентов

На хосте Shared-node IP-10.116.102.56/24 (файловом сервере) установить пакет `nfs-kernel-server`, который позволит предоставлять доступ к каталогам.

Поскольку это первая операция, которая выполняется с помощью apt в этом сеансе, обновить индекс локальных пакетов перед установкой:

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

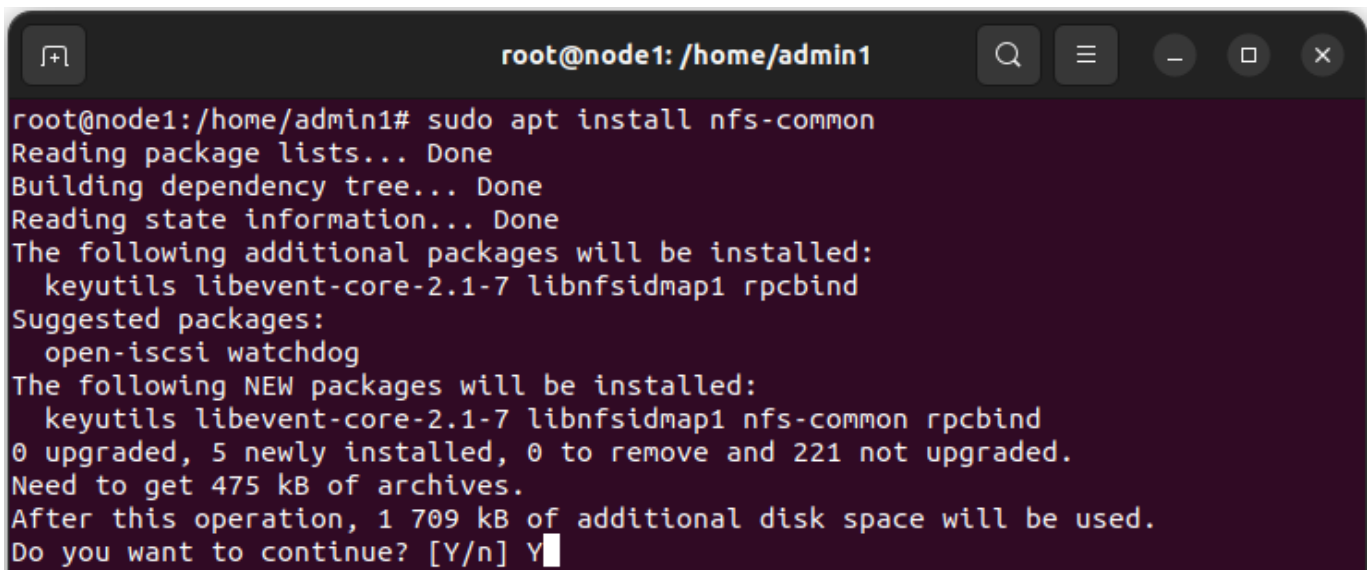
A screenshot of a terminal window titled 'root@shared-node: //'. The terminal shows the command 'sudo apt install nfs-kernel-server' being executed. The output indicates that the package is being installed along with several dependencies: keyutils, libnfsidmap2, libtirpc1, nfs-common, and rpcbind. It also shows that libllvm7 was automatically installed and is no longer required. The terminal prompts for confirmation to continue, with '[Y/n]' and a cursor visible.

```
root@shared-node: //  
File Edit View Search Terminal Help  
root@shared-node://# sudo apt install nfs-kernel-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libllvm7  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind  
Suggested packages:  
  open-iscsi watchdog  
The following NEW packages will be installed:  
  keyutils libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbind  
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.  
Need to get 492 kB of archives.  
After this operation, 1,709 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Рисунок 1.2 – Установка пакета nfs-kernel-server

На клиентах Node1 IP - 10.116.102.54/24 и Node2 IP - 10.116.102.55/24 необходимо установить пакет nfs-common, обеспечивающий функции NFS без добавления каких-либо серверных компонентов. Обновить индекс локальных пакетов перед установкой, чтобы гарантированно использовать актуальную информацию:

```
sudo apt update  
sudo apt install nfs-common
```



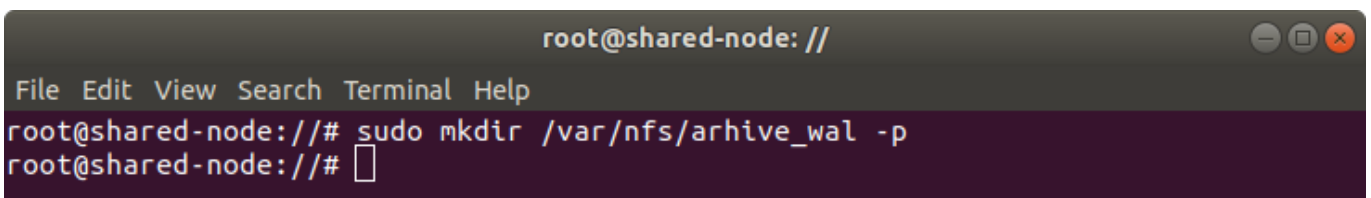
```
root@node1: /home/admin1
root@node1:/home/admin1# sudo apt install nfs-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 nfs-common rpcbind
0 upgraded, 5 newly installed, 0 to remove and 221 not upgraded.
Need to get 475 kB of archives.
After this operation, 1 709 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Рисунок 1.3 - Установка пакета nfs-common

Создание каталога на хосте (файловом сервере)

На хосте Shared-node IP-10.116.102.56/24 (файловом сервере) создать каталог командой:

```
sudo mkdir /var/nfs/arhive_wal -p
```

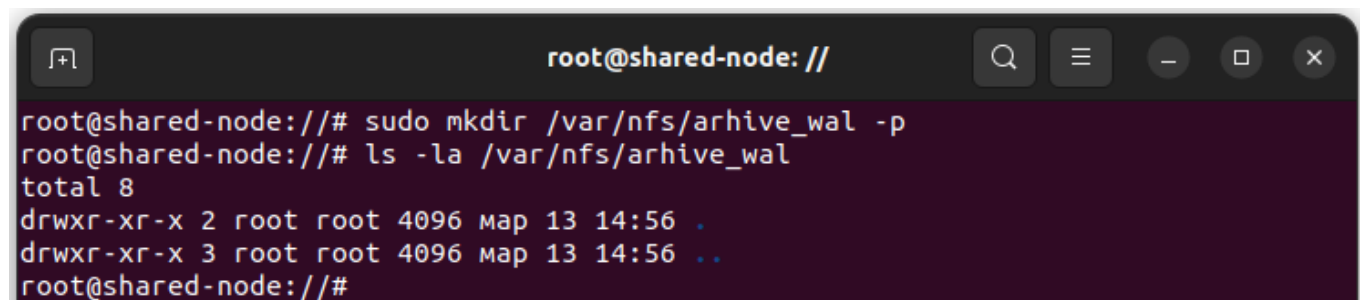


```
root@shared-node: //
File Edit View Search Terminal Help
root@shared-node://# sudo mkdir /var/nfs/arhive_wal -p
root@shared-node://#
```

Рисунок 1.4 – Создание каталога на хосте

Каталог создается с правами sudo, владельцем каталога будет пользователь root на хосте:

```
ls -la /var/nfs/arhive_wal
```



```
root@shared-node: //
root@shared-node://# sudo mkdir /var/nfs/arhive_wal -p
root@shared-node://# ls -la /var/nfs/arhive_wal
total 8
drwxr-xr-x 2 root root 4096 мар 13 14:56 .
drwxr-xr-x 3 root root 4096 мар 13 14:56 ..
root@shared-node://#
```

Рисунок 1.5 – Проверка установленных прав на каталог

Для безопасности NFS преобразует любые операции `root` на клиенте в операции с учетными данными `nobody:nogroup`. В связи с этим, необходимо изменить владельца каталога для соответствия этим учетным данным.

```
# sudo chown nobody:nogroup /var/nfs/arhive_wal
# ls -la /var/nfs/arhive_wal
```

```
root@shared-node: //
File Edit View Search Terminal Help
root@shared-node: // # sudo chown nobody:nogroup /var/nfs/arhive_wal
root@shared-node: // # ls -la /var/nfs/arhive_wal
total 8
drwxr-xr-x 2 nobody nogroup 4096 Jun 23 05:45 .
drwxr-xr-x 3 root    root    4096 Jun 23 05:45 ..
root@shared-node: // #
```

Рисунок 1.6 – Установка и проверка прав на каталог

Экспорт каталога

Открыть на хосте Shared-node IP-10.116.102.56/24 (файловом сервере) файл `/etc/exports` в текстовом редакторе с привилегиями `root`:

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

Файл содержит комментарии, показывающие общую структуру каждой строки конфигурации. Синтаксис выглядит следующим образом:

```
directory_to_share    client(share_option1,...,share_optionN)
(rw, sync, no_root_squash, no_subtree_check)
```

Возможно применение следующих опций:

- `rw`: эта опция дает клиенту доступ к чтению и записи на соответствующем томе;
- `sync`: эта опция принудительно заставляет NFS записывать изменения на диске, прежде чем отправлять ответ. В результате получаем более стабильную и согласованную среду, поскольку в ответе отражается фактическое состояние удаленного тома, при этом снижается скорость операций с файлами;
- `no_subtree_check`: данная опция предотвращает проверку вложенного дерева, когда хост проверяет фактическую доступность файла в экспортированном дереве при

каждом запросе. Это может вызвать трудности в случае переименования файла, если он открыт на клиентской системе. Проверку вложенного дерева в большинстве случаев лучше отключить;

– `no_root_squash`: по умолчанию NFS преобразует запросы удаленного пользователя `root` в запросы пользователя без привилегий на сервере. Это предназначено для обеспечения безопасности, чтобы пользователь `root` клиентской системы не мог использовать файловую систему хоста с правами `root`. Опция `no_root_squash` отключает такое поведение для определенных общих ресурсов.

Установить параметры подключения клиентов добавив строки:

```
/var/nfs/arhive_wal 10.116.102.54
(rw,sync,no_root_squash,no_subtree_check)
/var/nfs/arhive_wal 10.116.102.55
(rw,sync,no_root_squash,no_subtree_check)
```

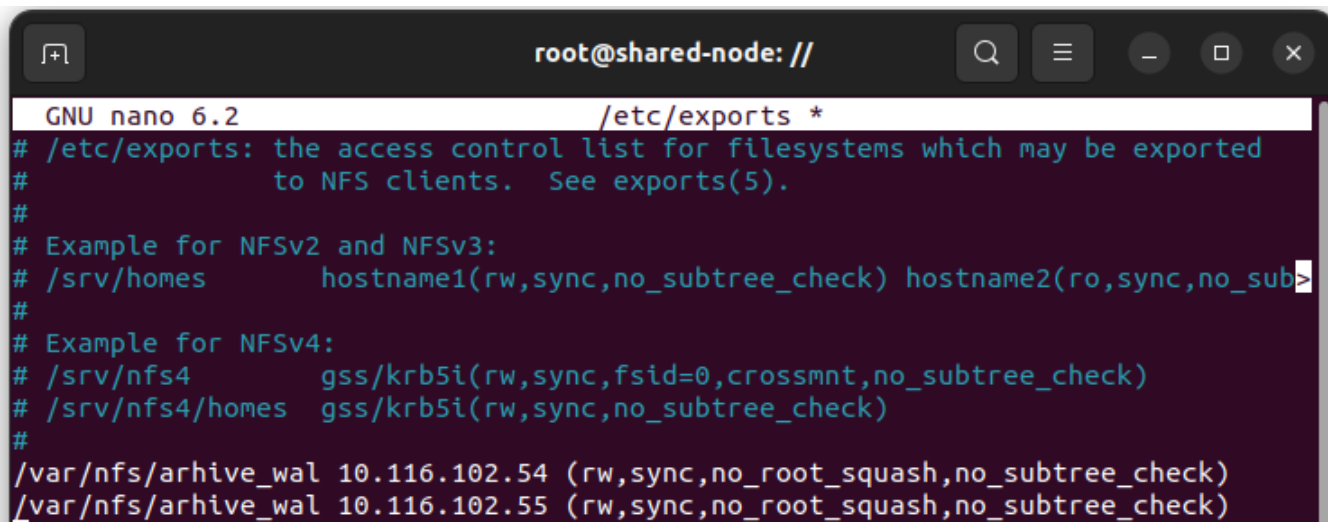


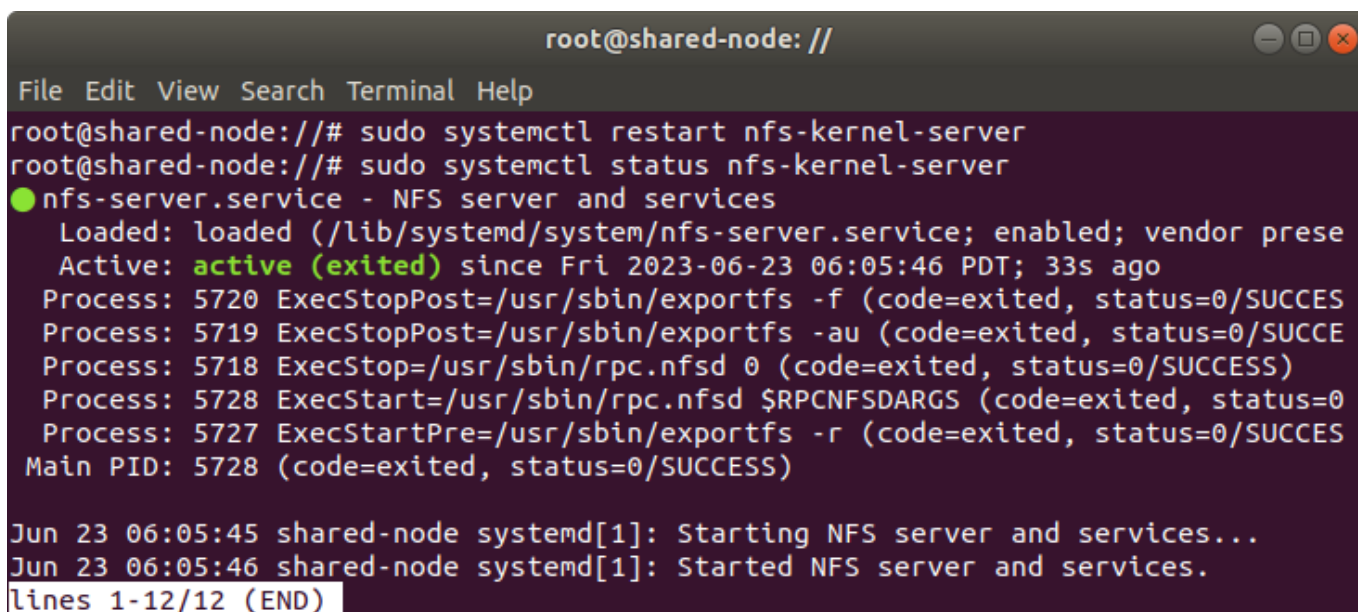
Рисунок 1.7 – Строки подключений клиентов в конфигурационном файле `/etc/exports`

Таким образом предоставляется доступ узлам кластера Node1 IP-110.116.102.54/24 и Node2 IP 10.116.102.55/24 к каталогу на внешнем файловом сервере.

После редактирования файла, его необходимо сохранить и закрыть.

Чтобы сделать общий ресурс доступным для настроенных клиентов, перезапустить сервер NFS с помощью следующей команды:

```
sudo systemctl restart nfs-kernel-server
sudo systemctl status nfs-kernel-server
```



```
root@shared-node: //
File Edit View Search Terminal Help
root@shared-node: // # sudo systemctl restart nfs-kernel-server
root@shared-node: // # sudo systemctl status nfs-kernel-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor prese
   Active: active (exited) since Fri 2023-06-23 06:05:46 PDT; 33s ago
     Process: 5720 ExecStopPost=/usr/sbin/exportfs -f (code=exited, status=0/SUCCE
     Process: 5719 ExecStopPost=/usr/sbin/exportfs -au (code=exited, status=0/SUCCE
     Process: 5718 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited, status=0/SUCCESS)
     Process: 5728 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0
     Process: 5727 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCE
    Main PID: 5728 (code=exited, status=0/SUCCESS)

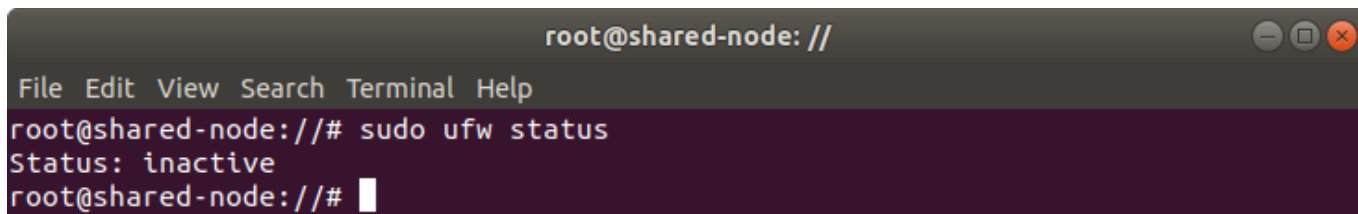
Jun 23 06:05:45 shared-node systemd[1]: Starting NFS server and services...
Jun 23 06:05:46 shared-node systemd[1]: Started NFS server and services.
lines 1-12/12 (END)
```

Рисунок 1.8 – Перезагрузка и проверка статуса службы nfs-kernel-server

Проверка брандмауэра на хосте

Состояние брандмауэра проверяется командой:

```
sudo ufw status
```



```
root@shared-node: //
File Edit View Search Terminal Help
root@shared-node: // # sudo ufw status
Status: inactive
root@shared-node: // #
```

Рисунок 1.9 – Проверка статуса брандмауэра

В представленном примере брандмауэр отключен. Если он будет активен, то необходимо выполнить команду с синтаксисом:

```
sudo ufw allow from [client_ip] to any port nfs
```

Чтобы открыть порт 2049 на хосте, необходимо указать IP-адрес клиентской системы.

На данном этапе подготовка хоста для хранения архива WAL закончена.

Создание точек монтирования и монтирование каталогов на клиентской системе

Операции монтирования проводятся на узлах кластера Node1 IP-10.116.102.54/24 и Node2 IP 10.116.102.55/24. Действия будут аналогичными.

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|

Создать каталог для монтируемого ресурса:

```
sudo mkdir -p /nfs/arhive_wal
```

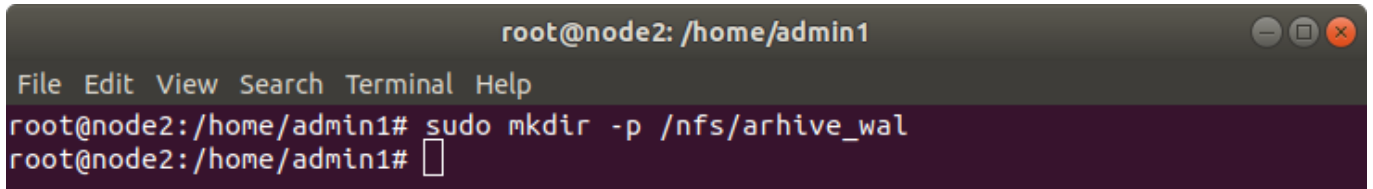


Рисунок 1.10 - Создайте каталога для монтируемого ресурса

Теперь есть место для размещения удаленных общих ресурсов и открыт брандмауэр, поэтому можно монтировать общие ресурсы, используя IP-адрес сервера хоста:

```
sudo mount 10.116.102.56:/var/nfs/arhive_wal /nfs/arhive_wal
```

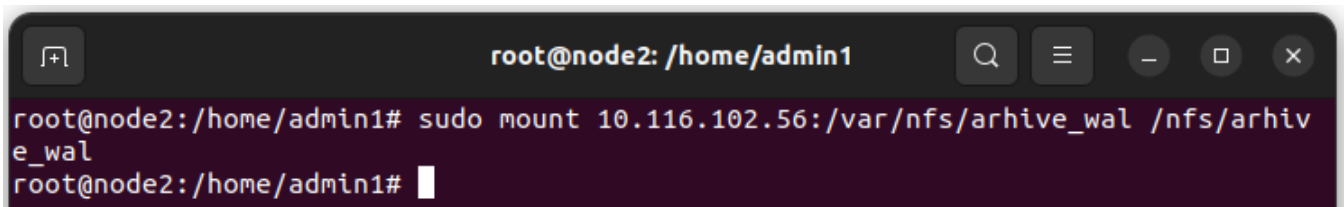


Рисунок 1.11 – Команда монтирования общего ресурса

Повторить действия на Node1:

```
# sudo mkdir -p /nfs/arhive_wal  
# sudo mount 10.116.102.56:/var/nfs/arhive_wal /nfs/arhive_wal
```

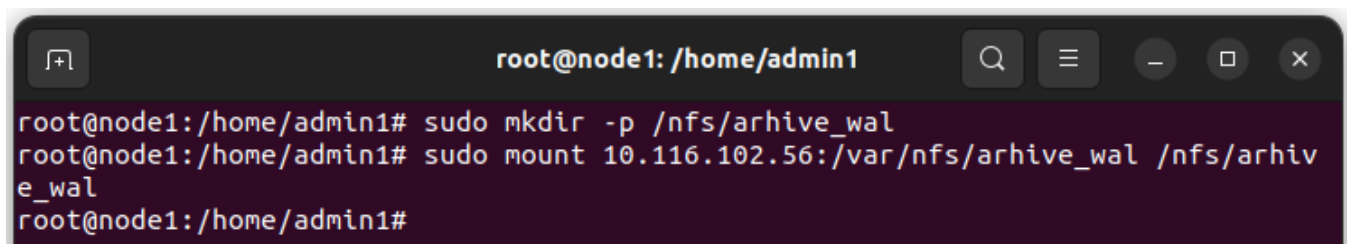


Рисунок 1.12 – Создание точки монтирования и монтирование сетевого каталога

Сетевые каталоги на узлах кластера смонтированы.

Проверить смонтированные каталоги возможно командой:

```
df -h
```

```

root@node1: /home/admin1
root@node1:/home/admin1# df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     391M        1,5M   390M    1% /run
/dev/sda3                  49G        16G    31G   34% /
tmpfs                     2,0G         0    2,0G    0% /dev/shm
tmpfs                     5,0M         0    5,0M    0% /run/lock
/dev/sda2                  512M        6,1M   506M    2% /boot/efi
tmpfs                     391M       104K    391M    1% /run/user/1000
10.116.102.56:/var/nfs/arhive_wal 49G        16G    31G   35% /nfs/arhive_wal
root@node1:/home/admin1#

```

Рисунок 1.13 – Вывод смонтированных каталогов

Монтирование удаленных каталогов NFS при загрузке

Удаленные общие ресурсы NFS можно автоматически монтировать при загрузке, для чего их нужно добавить в файл `/etc/fstab` на всех узлах кластера.

Открыть данный файл в текстовом редакторе с привилегиями `root`:

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

Добавьте в конец файла строки параметры монтируемого каталога на «Shared-node» IP-10.116.102.56:

```
10.116.102.56:/var/nfs/arhive_wal /nfs/arhive_wal nfs
auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

```

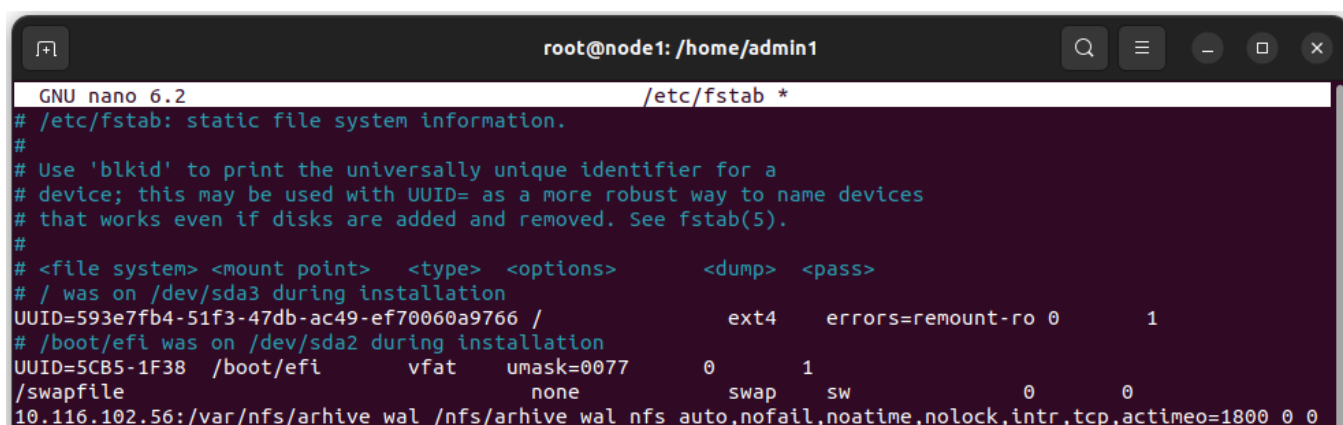
GNU nano 6.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>      <dump> <pass>
# / was on /dev/sda3 during installation
UUID=593e7fb4-51f3-47db-ac49-ef70060a9766 /      ext4      errors=remount-ro 0      1
# /boot/efi was on /dev/sda2 during installation
UUID=5CB5-1F38 /boot/efi vfat      umask=0077      0      1
/swapfile none swap      sw            0      0
10.116.102.56:/var/nfs/arhive_wal /nfs/arhive_wal nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0

```

Рисунок 1.14– Редактирование файла `/etc/fstab` на Node2

После редактирования файла его необходимо сохранить и закрыть.

Повторить действие на всех узлах кластера.



```
GNU nano 6.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda3 during installation
UUID=593e7fb4-51f3-47db-ac49-ef70060a9766 / ext4 errors=remount-ro 0 1
# /boot/efi was on /dev/sda2 during installation
UUID=5CB5-1F38 /boot/efi vfat umask=0077 0 1
/swapfile none swap sw 0 0
10.116.102.56:/var/nfs/arhive_wal /nfs/arhive_wal nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

Рисунок 1.15 – Редактирование файла /etc/fstab на Node1

На этом этапе монтирование сетевого каталога закончено. Узлы кластера будут автоматически монтировать сетевой каталог.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Smallint – тип значения, представляет числа от -32768 до 32767, используя 2 байта.

Integer – тип значения, представляет обычное целое число, которое обычно используется для хранения автоинкрементированных идентификаторов и других целочисленных значений. Он использует 4 байта и может представлять числа от -2147483648 до 2147483647.

Bigint – тип значения, как правило это «большое» целое число, способное представлять числа в диапазоне от -9223372036854775808 до 9223372036854775807, используя 8 байтов.

Boolean – это логический тип данных, который может принимать значения True (верно) или False (неверно).

CRL (Certificate Revocation List) – это список сертификатов, которые были отозваны выдающим их центром сертификации. CRL содержит информацию о сертификатах, которые больше не являются действительными, и используется центрами сертификации для проверки того, что сертификаты все еще являются действительными.

IPv4 (Internet Protocol version 4) — это четвёртая версия интернет-протокола (IP). Он использует 32-битные (четырёхбайтные) адреса, что ограничивает адресное пространство 4 294 967 296 возможными уникальными адресами. Протокол описан в RFC 791 (сентябрь 1981 года) и заменил RFC 760 (январь 1980 года).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

| | | |
|------|---|--|
| DHCP | – | Dynamic Host Configuration Protocol – протокол динамической настройки узла |
| GLSB | – | Global Server Load Balancing – распределение информации между серверами для повышения удобства работы пользователей и обеспечения более быстрой передачи информации и данных |
| SQL | – | Structured Query Language – язык структурированных запросов |
| SSL | – | Secure Sockets Layer – уровень защищенных сокетов. Криптографический протокол, который подразумевает безопасную связь |
| БД | – | База данных |
| ДСЧ | – | Датчик случайных чисел |
| ОС | – | Операционная система |
| СУБД | – | Система управления базами данных |
| УЗ | – | Учетная запись |
| УЦ | – | Удостоверяющий центр |
| ЭВМ | – | Электронно-вычислительная машина |

[illegible]

| | | |
|--------------------|--------------------------|--------------------------|
| № изменения: _____ | Подпись отв. лица: _____ | Дата внесения изм: _____ |
|--------------------|--------------------------|--------------------------|