

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ «JАТОВА»

Руководство по настройке. Часть 1.  
Управление режимом работы узлов кластера.  
Компонент «jaDog»

643.72410666.00067-07 98 02-01

Листов 284

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

## АННОТАЦИЯ

В документе 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».

В первой части руководства приводится информация, позволяющая ознакомиться с основными функциями компонента «jaDog» на примере построения кластера, состоящего из главного и резервного узлов.

Во второй части руководства приводятся сведения по настройке различных конфигураций кластеров, в том числе и геораспределенных, а также различные подходы к настройке репликации данных.

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

В качестве примера описаны настройки SSL соединения, с использованием тестового удостоверяющий центр (УЦ) с алгоритмом ключа ГОСТ Р 34.10-2012 на основе веб-интерфейса КриптоПро УЦ версии 2.0. Работа СУБД «Jatoba» с использованием тестовых УЦ не гарантируется производителем. Поддержка на такие конфигурации не оказывается.

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 — 3.4.

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

Степени важности примечаний, применяемые в документе:

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



**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.

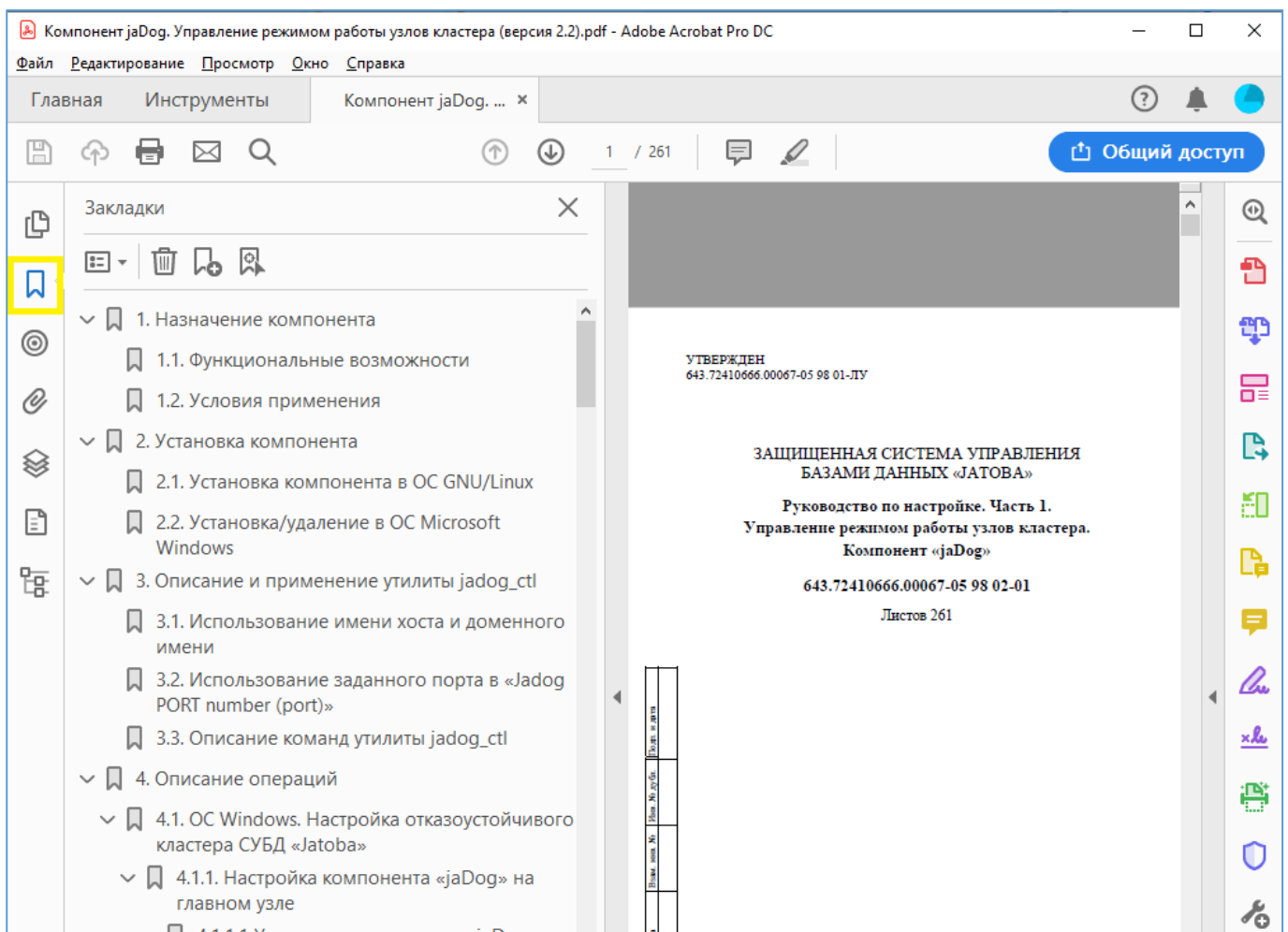


Рисунок 1.1 - Структура документа

## СОДЕРЖАНИЕ

1. Назначение компонента .....	9
1.1. Функциональные возможности .....	12
1.2. Инженерные решения компонента .....	12
1.3. Условия применения .....	14
2. Установка компонента .....	16
2.1. Установка компонента в ОС GNU/Linux .....	18
2.1.1. Установка пакетов компонента на главном узле (Master) .....	18
2.1.2. Настройка аутентификации на главном узле (Master) .....	19
2.1.3. Установка пароля системного пользователя ОС «postgres» на главном узле .....	20
2.1.4. Установка расширения «jadow» на главном узле .....	20
2.1.5. Создание пользователя «jadow_user» на главном узле .....	21
2.1.6. Установка пакетов компонента «jaDog» на резервных узлах (Slave) .....	22
2.2. Установка/удаление в ОС Microsoft Windows .....	23
3. ОС GNU/Linux Автоматическая настройка кластера СУБД «Jatoba» с помощью файлов конфигурации .....	26
3.1. Silent (тихий) режим установки компонента «jaDog» на главном узле .....	27
3.2. Использование файла ответов для автоматизированного формирования кластера .....	30
3.3. Подготовка и выпуск сертификатов SSL .....	31
3.4. Формирование файла ответов из шаблона .....	31
3.5. Формирование кластера при помощи файла ответов .....	34
3.6. Изменение состава кластера при помощи файла ответов .....	36
3.6.1. Добавление нового узла(ов) в кластер с использованием файла ответов .....	37
3.6.2. Исключение узла(ов) из кластера с использованием файла ответов .....	39
3.7. Изменение состава/параметров кластера на основе целевой структуры .....	41
3.7.1. Получение целевой структуры и настроек кластера .....	41
3.7.2. Сохранение состава и параметров кластера в файл .....	42
3.7.3. Применение кодирования параметров в Base64 .....	43
3.7.4. Ограничения на изменения параметров .....	44
3.7.5. Изменение параметров кластера с использованием файла ответов .....	45
3.8. Выполнение SQL-скриптов при автоматизированной настройке кластера .....	46
4. Проверка состояния кластера .....	48
4.1. Проверка работоспособности кластера .....	48
4.2. Подключение к узлу кластера .....	49
4.2.1. Конфигурационный файл аутентификации (jadow_hba.cfg) .....	49
4.2.2. Подключение по умолчанию .....	52
4.2.3. Подключение используя параметры (ip port 'login') .....	52
4.2.4. Подключение используя параметры соединения (connect ip port 'login') .....	53



4.2.5. Подключение по SSL используя переменные окружения .....	54
4.2.6. Подключение по SSL используя параметры по умолчанию .....	55
4.2.7. Подключение по SSL используя строку подключения .....	55
4.3. Перезагрузка компонентов «jaDog» в кластере .....	56
4.3.1. Перезагрузка компонента «jaDog» на узле кластера .....	60
4.3.2. Перезагрузка СУБД на узлах кластера .....	62
4.4. Перезагрузка конфигурации СУБД .....	63
4.4.1. Перезагрузка конфигурации СУБД на узле кластера .....	63
4.4.2. Перезагрузка конфигурации СУБД в кластере .....	66
4.5. Остановка и запуск СУБД кластера .....	67
4.5.1. Остановка СУБД кластера, главного или резервного узлов .....	67
4.5.2. Запуск СУБД кластера, главного или резервного узлов .....	68
5. Описание и применение утилиты «jadog_ctl» .....	70
5.1. Использование имени хоста и доменного имени .....	71
5.2. Использование заданного порта в «Jadog PORT number (port)» .....	73
5.3. Описание команд утилиты «jadog_ctl» .....	75
6. ОС GNU/Linux. Настройка отказоустойчивого кластера СУБД «Jatoba» в ручном режиме .....	92
6.1. Установка пакетов СУБД на главном узле (Master) .....	92
6.2. Настройки аутентификации на главном узле .....	93
6.3. Установка пароля системного пользователя ОС «postgres» на главном узле .....	94
6.4. Установка расширения «jadog» на главном узле .....	94
6.5. Вызов функции «add_jadog_user» для создания пользователя «jadog_user» на главном узле .....	95
6.6. Установка пакетов СУБД на резервном узле (Slave) .....	96
6.6.1. Установка СУБД на резервном узле инсталлятором .....	96
6.6.2. Установка СУБД на резервном узле вручную из локального репозитория .....	97
6.6.3. Действия не требующие выполнения на резервном узле при установке СУБД из локального репозитория .....	98
6.7. Настройка компонента «jaDog» на главном узле через сервис «jadog» .....	99
6.7.1. Пункт меню 1 «Database server and jadog directory settings» .....	102
6.7.2. Пункт меню 2 «Inter-jadog communication setting» .....	104
6.7.3. Пункт меню 3 «User / Admin access network setting» .....	106
6.7.4. Пункт меню 4 «Administrator account list» .....	110
6.7.5. Пункт меню 5 «Database server system account and connection settings» .....	111
6.7.6. Пункт меню 6 «Failover setting» .....	117
6.7.7. Пункт меню 7 «Replication setting» .....	119
6.7.8. Пункт меню 8 «WAL archive and restore settings» .....	124
6.7.9. Пункт меню 9 «Reporting and logging» .....	126
6.7.10. Пункт меню 10 «REST API Settings» .....	132
6.7.11. Пункт меню 11 «Recovery settings» .....	133

6.7.12. Пункт меню 12 «Reset all settings to the default values» .....	134
6.7.13. Пункт меню 13 «Check and show all settings» .....	135
6.7.14. Пункт меню 14 «Save setting and setup jadog».....	135
6.7.15. Запуск серверной части компонента «jaDog» на главном узле.....	137
6.8. Настройка и запуск компонента «jaDog» на резервном узле .....	138
6.8.1. Конфигурирование и запуск компонента «jaDog» на резервном узле .....	139
6.8.2. Запуск серверной части компонента «jaDog» на резервном узле .....	145
6.9. Настройка кластера из двух узлов (главный-резервный) .....	146
6.10. Подключение к сетевому узлу (connect ip port 'login') .....	146
6.11. Создание кластера (add cluster 'name') .....	147
6.12. Добавление узла кластера в качестве главного (cluster add master ip).....	147
6.13. Статус узлов кластера (cluster status).....	148
6.14. Присвоение публичного IP-адреса главному узлу кластера (cluster activate).....	149
6.15. Добавление резервного узла в асинхронном режиме (cluster add slave ip) .....	150
6.16. Добавление резервного узла в синхронном режиме (cluster add sync slave ip).....	151
6.17. Изменение типа репликации резервного узла .....	152
6.18. Автоматическая настройка параметров резервного узла .....	155
6.19. Файл состояния кластера jadog_state.yml.....	156
7. Дополнительные функциональные возможности .....	157
7.1. Просмотр имени кластера (clusters list) .....	157
7.2. Деактивация публичного IP-адреса главного узла кластера (cluster deactivate).....	157
7.3. Удаление узла кластера (cluster delete node) .....	159
7.4. Версия протокола jadog (protocol version).....	159
7.5. Вывод справочной информации (\h).....	159
7.6. Выход из консоли jadog_ctl (\q) .....	160
7.7. Создание дата-центра (datacenter create 'name') .....	160
7.8. Удаление дата-центра (datacenter delete 'name').....	161
7.9. Включение узла кластера в дата-центр (datacenter 'name' attach node ip).....	161
7.10. Выключение узла кластера из дата-центра (datacenter 'name' detach node ip).....	162
7.11. Принудительное переключение на текущий дата-центр (datacenter promote).....	163
7.12. Множественные подключения .....	163
7.13. Управление учетными записями .....	164
7.13.1. Управление учетными записями компонента кластера .....	164
7.13.2. Управление учетными записями компонента для СУБД .....	169
7.14. Редактирование и получение параметров кластера «jaDog».....	171
7.14.1. Редактирование параметров через интерфейс сервиса «jadog» .....	171
7.14.2. Редактирование и получение параметров через утилиту «jadog_ctl».....	172
7.14.3. Редактирование и получение параметров в терминальном режиме через утилиту «jadog_ctl» ..	176
7.15. Процедура принудительной смены ролей серверов (switchover) .....	178

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

7.16. Режим технического обслуживания (maintenance).....	179
7.16.1. Включение режима технического обслуживания на узле .....	181
7.16.2. Включение режима технического обслуживания кластера.....	181
7.16.3. Вывод кластера из режима технического обслуживания .....	182
7.16.4. Вывод отдельного узла из режима технического обслуживания .....	183
7.17. Процедура обработки отказа (failover).....	183
7.18. Восстановление после отказа (fail) .....	186
7.19. Настройка SSL соединения в ОС GNU/Linux компонента «jaDog» с СУБД.....	189
7.20. Настройка SSL соединения в ОС GNU/Linux для узлов кластера.....	190
7.20.1. Установка криптопровайдера.....	190
7.20.2. Установка библиотеки OpenSSL с поддержкой ГОСТ 2012 .....	191
7.20.3. Пакет OpenSSL .....	193
7.20.4. Настройка конфигурационного файла «openssl_conf» .....	194
7.20.5. Формирование сертификата .....	195
7.20.6. Экспорт сформированного сертификата .....	198
7.20.7. Редактирование сформированного сертификата.....	199
7.20.8. Установка корневого сертификата УЦ и списка отозванных сертификатов CRL .....	200
7.20.9. Настройка главного узла для SSL соединения .....	202
7.20.10. Настройка резервного узла для SSL соединения.....	206
7.21. Применение узла с ролью «Referee».....	207
7.21.1. Добавление узла с ролью «Referee» с помощью «jadowctl».....	208
7.21.2. Добавление узла с ролью «Referee» в процессе автоматизированной настройки кластера .....	208
7.21.3. Удаление узла с ролью «Referee» .....	209
7.22. Получение информации о статусе узлов кластера .....	209
7.23. Планировщик заданий кластера.....	213
7.23.1. Ограничения планировщика заданий кластера .....	214
7.23.2. Добавление задания в планировщик.....	217
7.23.3. Получение списка заданий .....	218
7.23.4. Удаление задания из планировщика.....	220
7.23.5. Параметры планировщика .....	221
7.24. Синхронизация файлов в кластере .....	223
7.24.1. Ограничения менеджера синхронизации файлов.....	223
7.24.2. Настройка синхронизации файлов в кластере .....	224
7.24.3. Добавление нового файла(ов) в список синхронизации.....	225
7.24.4. Добавление нового каталога для синхронизации.....	226
7.24.5. Просмотр списка файлов/каталогов для синхронизации.....	227
7.24.6. Выполнение синхронизации файлов и каталогов на узлах кластера.....	228
7.24.7. Удаление файлов/каталогов из списка синхронизации .....	231
7.24.8. Параметры синхронизации файлов в кластере .....	232

7.25. Запуск пользовательских сценариев при смене роли узла .....	233
7.25.1. Типы сценариев .....	233
7.25.2. Поддерживаемые события .....	233
7.25.3. Ограничения при запуске сценариев .....	234
7.25.4. Предварительные требования запуска сценариев .....	234
7.25.5. Алгоритм запуска и выполнения сценариев .....	234
7.25.6. Настройка конфигурации запуска сценариев .....	235
7.25.7. Проверка работоспособности сценариев.....	239
8. Обновление и удаление компонента .....	241
8.1. Обновление (миграция) компонента «jaDog».....	241
8.2. Удаление компонента «jaDog».....	246
9. Сообщения об ошибках .....	247
9.1. Ошибка добавления резервного узла.....	247
Приложение 1 .....	249
Структура конфигурационного файла jadog.yml.....	249
Структура файла состояния кластера jadog_state.yml.....	270
Перечень исполняемых скриптов в ОС Windows и GNU Linux .....	273
Приложение 2 .....	275
Пример установки СУБД «Jatoba» из локального репозитория для ОС Ubuntu .....	275
Термины и определения.....	281
Перечень сокращений .....	282

## 1. НАЗНАЧЕНИЕ КОМПОНЕНТА

СУБД «Jatoba» может применяться в составе группы серверов, образующих отказоустойчивый кластер. Кластер на базе СУБД строится на механизмах потоковой репликации, заложенных в ядро СУБД.

В СУБД «Jatoba» поддерживается простая схема репликации с использованием одного главного сервера и нескольких резервных серверов (работающих в синхронном или асинхронном режимах).

Компонент «jaDog» входит в состав СУБД «Jatoba», предназначен для контроля состояния серверов (далее — узлов) кластера СУБД и обеспечивает выполнение функций обработки отказа и восстановления узлов кластера в автоматическом режиме.

В руководстве приведено описание настройки компонента «jaDog» и настройки кластера под управлением ОС Windows Server 2016 (далее – ОС Windows) и Ubuntu 22.04 (далее – ОС Ubuntu). Для кластеров под управлением поддерживаемых ОС семейства Linux настройка выполняется аналогично настройке кластера под управлением ОС Ubuntu.

В руководстве настройка и работа компонента «jaDog» рассматривается на примере кластера из двух узлов: один главный и один резервный. Добавление дополнительных резервных серверов выполняется аналогичным набором действий. Схематично физическая структура узлов кластера показана на рисунке 1.1.

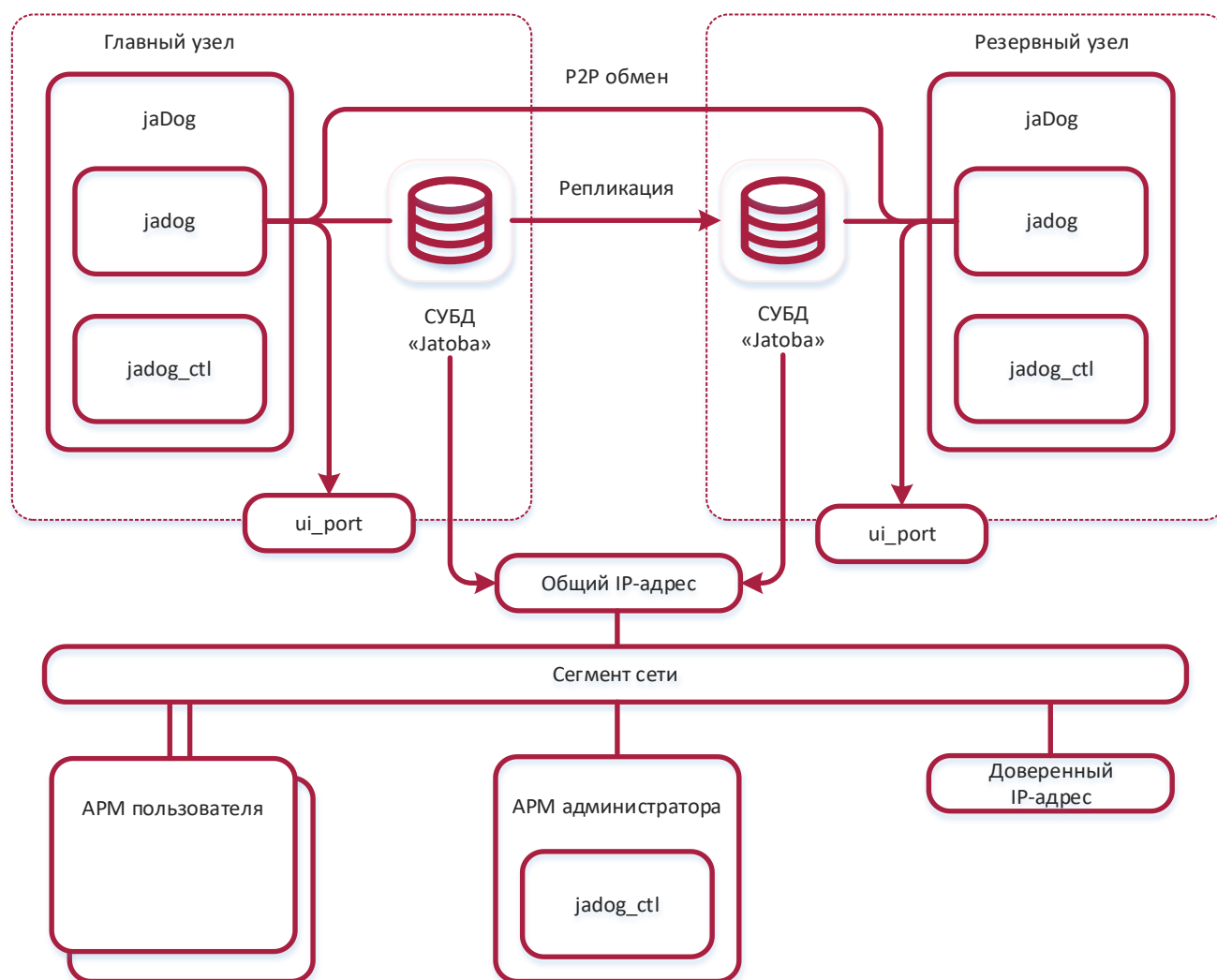


Рисунок 1.1 – Физическая структура узлов кластера. Пример для ОС GNU/Linux  
Основные понятия, используемые на рисунке 1.1:

- jaDog**
- компонент «jaDog» контролирует состояние сервера СУБД, в случае аварийной ситуации переключает роль главного сервера на один из резервных серверов (осуществляет «failover» – обработку отказа); jaDog-процессы серверной части постоянно взаимодействуют между собой, обмениваясь текущим состоянием каждого узла (peer-2-peer обмен), клиентская часть (утилита «jadog\_ctl») предоставляет администраторам интерфейс для контроля состояния узлов кластера и управления jaDog-процессами
- Jatoba**
- сервер СУБД «Jatoba» с настроенным режимом репликации журнала транзакций (или журнала опережающей записи)

- Ui\_port –  
(user\_interface\_port) – порт доступа административных подключений для управления кластером через утилиту «`jadog_ctl`» (по умолчанию используется порт 54321)
- Главный узел  
(Master) – роль сервера СУБД в реализации отказоустойчивого режима работы, при которой он принимает все запросы от пользователей на изменение данных. Главный узел осуществляет ведение основного журнала транзакций и обеспечивает условия его передачи подчиненным узлам. В настоящее время предусмотрено, что только один узел в кластере может иметь роль главного
- Доверенный адрес  
(trusted\_address) – ресурс или узел располагается в назначенном сетевым администратором надежном сетевом сегменте (настройка маршрутизации, дополнительный хост-арбитр), доступность которого можно проверить по протоколу ICMP (команда «`ping`»). Используется главным и резервным узлом для проверки доступности сетевой инфраструктуры и правильного принятия решения о произошедшем отказе
- Кластер – группа серверов СУБД (узлов), обеспечивающих заданный режим надежности функционирования СУБД для прикладных программ. В настоящее время реализован наиболее распространенный режим отказоустойчивости: режим горячей замены функций главного узла
- Общий IP-адрес  
(public IP) – IP-адрес, назначением которого управляет компонент «`jaDog`» в процессе функционирования. При нормальном состоянии работы этот адрес назначается главному узлу. На этот адрес рекомендуется настраивать пользовательские приложения, в случае отказа («`fail`») — выхода из строя главного сервера — компонент «`jaDog`» запустит процедуру обработки отказа, которая включает следующие шаги:
- изоляция текущего главного узла,
  - выбор из числа резервных узлов нового главного узла,
  - повышение роли выбранного резервного узла до роли главного узла,
  - миграция общего IP адреса на главный узел
- Резервный узел  
(Slave) – роль сервера СУБД в реализации отказоустойчивого режима работы, при которой он постоянно находится в режиме синхронизации (получение и восстановление) журнала транзакций. Также находится в режиме горячей замены — при

выходе из строя главного узла может принять на себя его роль. При этом потеря данных сводится к минимуму и может составить последние незафиксированные транзакции, сделанные на главный узел в момент его выхода из строя

Сегмент сети — логически обособленная часть сети, обеспечивающая соединение пользователей с кластером серверов и компонентов «jaDog» с доверенным адресом

### 1.1. Функциональные возможности

Компонент «jaDog» устанавливается совместно с СУБД «Jatoba» на каждый узел в кластере, включает в себя серверную и клиентскую часть, которые выполняют следующие функции:

- контроль состояния узла кластера;
- обмен данными с компонентами «jaDog», установленными на других узлах кластера (осуществляется обмен текущим состоянием каждого узла);
- в случае аварийной ситуации — автоматическое переключение роли главного сервера на один из резервных серверов (осуществляет процедуру обработки отказа «failover»);
- автоматизация действий пользователей при ручном переключении роли сервера СУБД (при выполнении процедуры принудительной смены ролей серверов «switchover»).

В ОС Microsoft Windows серверная часть устанавливается в виде служб jaDogService и jaDogRunAsService.

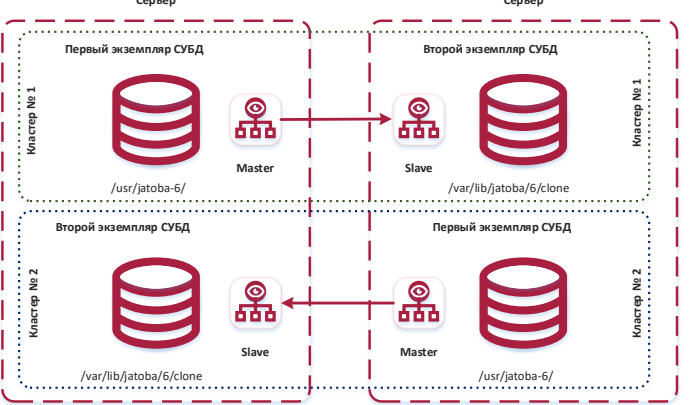
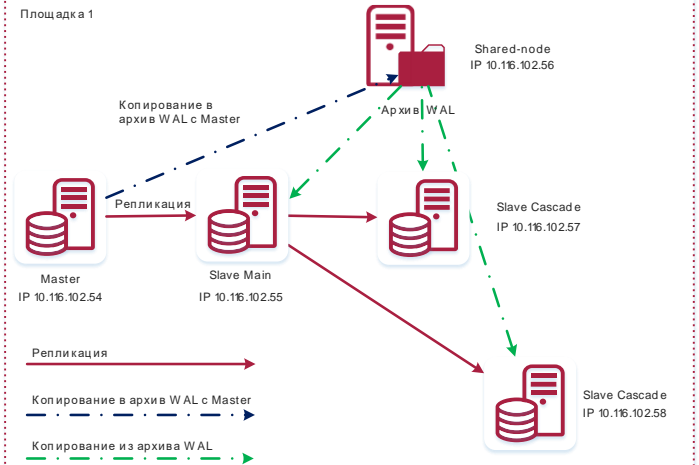
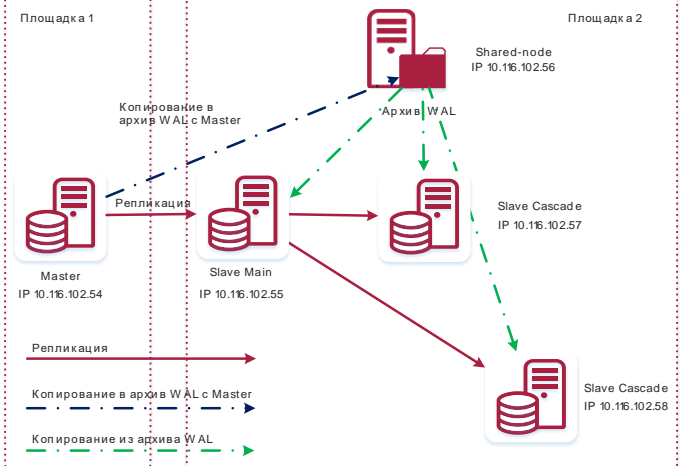
Клиентская часть представляет собой консольную утилиту «jadog\_ctl». Утилита «jadog\_ctl» предоставляет администраторам интерфейс для подключения и управления работой серверной части.

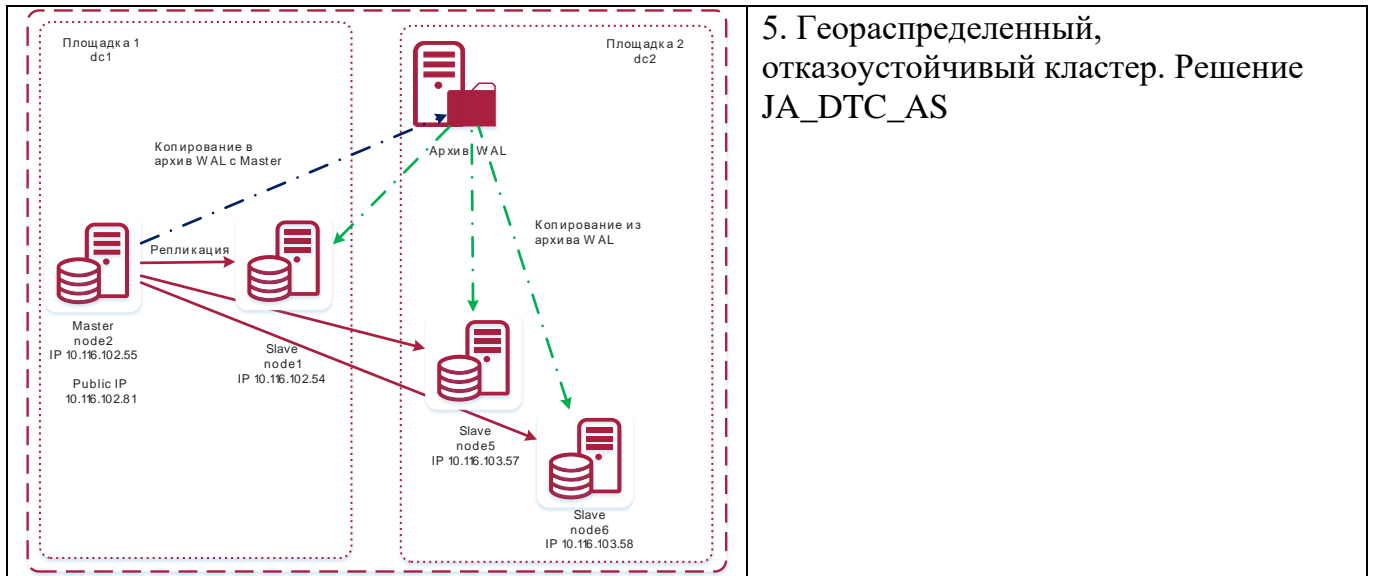
### 1.2. Инженерные решения компонента

Помимо перечисленных функциональных возможностей, компонент «jaDog» предоставляет инженерные решения работы кластера, представленные в таблице 1.1



Таблица 1.1 – Инженерные решения компонента «jaDog»

Схема работы компонента	Разделы описания инженерного решения во второй части руководства
	<p>1. ПЕРЕКРЕСТНАЯ РЕПЛИКАЦИЯ. ИСПОЛЬЗОВАНИЕ ФАЙЛОВ ОТВЕТОВ</p> <p>2. ПЕРЕКРЕСТНАЯ РЕПЛИКАЦИЯ. НАСТРОЙКА В РУЧНОМ РЕЖИМЕ</p>
	<p>3. КАСКАДНАЯ РЕПЛИКАЦИЯ. ИСПОЛЬЗОВАНИЕ ФАЙЛОВ ОТВЕТОВ</p> <p>4. КАСКАДНАЯ РЕПЛИКАЦИЯ. НАСТРОЙКА В РУЧНОМ РЕЖИМЕ</p>
	<p>4.4. Работа кластера с каскадной репликацией в дата-центрах</p>



## 5. Геораспределенный, отказоустойчивый кластер. Решение JA\_DTC\_AS

### 1.3. Условия применения

Компонент «jaDog» функционирует в рамках локальной сети и в сетях с центрами обработки данных (ЦОД).

Компонент «jaDog» может использоваться совместно с:

- СУБД «Jatoba»;
- компонентом пользовательского веб-интерфейса для администраторов «Jatoba data safe» (рисунок 1.2).

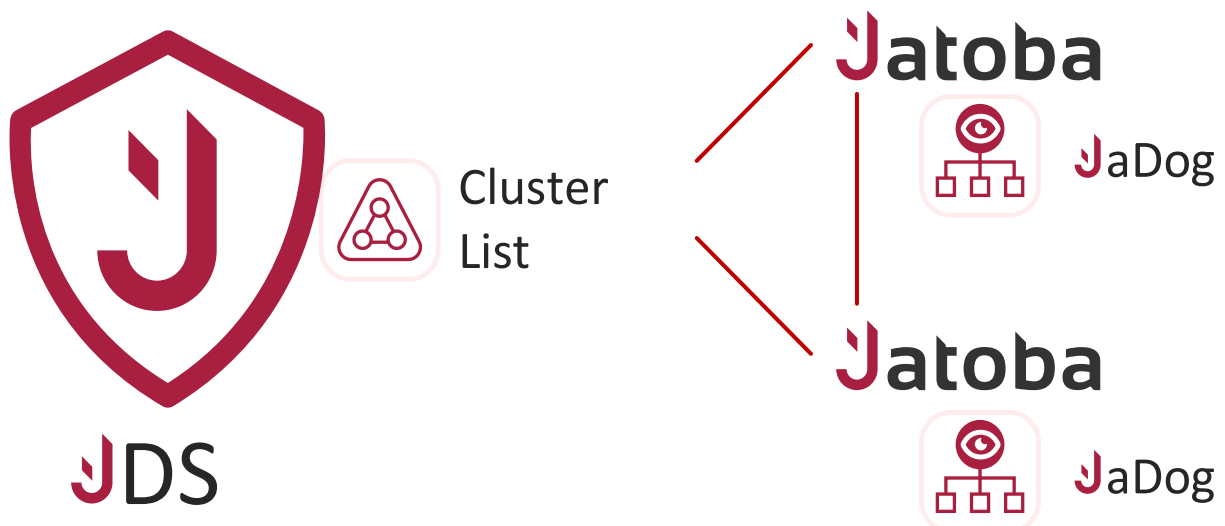


Рисунок 1.2 – Взаимодействие компонента jaDog с JDS

Узлы кластера должны работать на одном семействе операционных систем.

Компонент «jaDog» имеет следующие ограничения в эксплуатации:

- не поддерживает работу с протоколом динамической настройки узла DHCP;
- не поддерживает работу с вложенными табличными пространствами СУБД «Jatoba»;
- поддерживает только версия IPv4 сетевого протокола;
- конфигурационный файл аутентификации jadog\_hba.cfg (п. 4.2.1) регулирует доступ только к локальному экземпляру компонента.

## 2. УСТАНОВКА КОМПОНЕНТА

Компонент «jaDog» устанавливается на сервер при помощи менеджера пакетов (в случае ОС GNU Linux) или инсталлятора (в случае ОС Microsoft Windows). Правила и последовательность установки СУБД «Jatoba» приведены в документе «Защищенная система управления базами данных «Jatoba». Руководство по установке».



На всех узлах кластера должна быть установлена одна и та же версия компонента «jaDog». Совместная работа разных версий компонента «jaDog» не гарантируется и не поддерживается.

Установка компонента «jaDog» включает установку серверной части сервиса «jadog» (файл «jadog.exe» в ОС Microsoft Windows), клиентской части утилиты «jadog\_ctl» (файл «jadog\_ctl.exe» в ОС Microsoft Windows) и вспомогательных серверных скриптов для выполнения системных операций управления СУБД на узле кластера и сетевыми настройками узла в кластере.

Блок-схема установки компонента «jaDog» в ОС GNU/Linux представлена на рисунке 2.1.

В результате установки серверная часть устанавливается в виде службы и работает в фоновом режиме, клиентская часть устанавливается в виде утилиты «jadog\_ctl».

Управление поведением серверной части компонента выполняется администратором через утилиту «jadog\_ctl» посредством заданного набора команд.

Для удаленного контроля состояния узлов кластера и управления параметрами серверной части кластеров утилита «jadog\_ctl» может быть установлена на сервер или рабочее место администратора, не входящие в состав кластера, путем переноса и запуска на исполнение файла «jadog\_ctl» (файл «jadog\_ctl.exe» в ОС Microsoft Windows).

Установка компонента «jaDog» возможна следующими способами:

- установка базовых пакетов СУБД инсталлятором, установка пакета jaDog вручную;
- установка всех необходимых пакетов вручную.

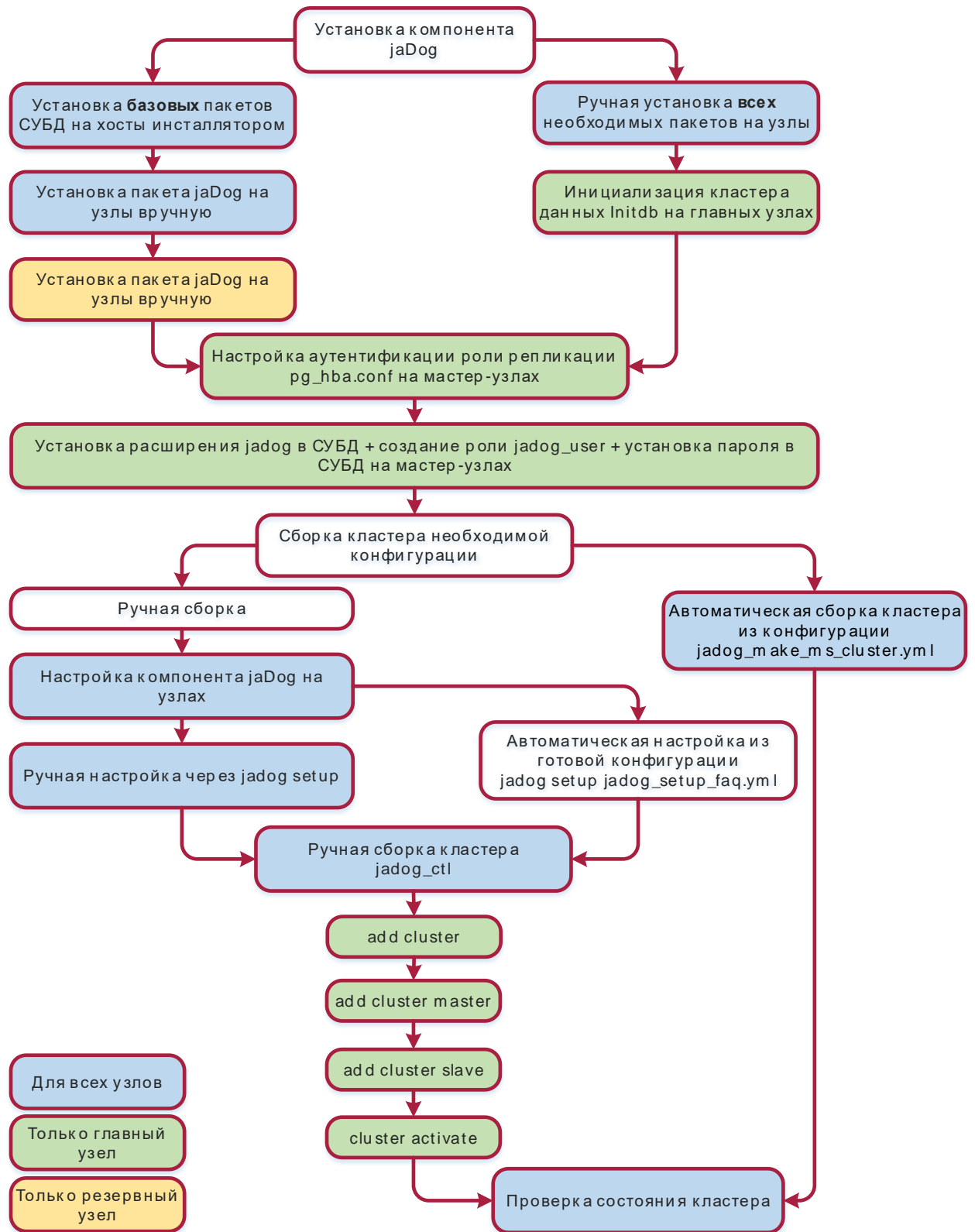


Рисунок 2.1 – Блок схема установки компонента jaDog в ОС GNU/Linux

## 2.1. Установка компонента в ОС GNU/Linux

### 2.1.1. Установка пакетов компонента на главном узле (Master)

Базовую установку пакетов компонента «jaDog» в ОС GNU/Linux на главном узле можно провести двумя способами:

- 1) Установка из локального репозитория (CDROM) – производится из файлов, записанных на компакт-диск или скопированных с него.
- 2) Установка непосредственно из deb/rpm-файлов – производится опционально, по усмотрению пользователя.

Команда установки пакетов компонента «jaDog» в разных дистрибутивах Linux может отличаться, также могут отличаться некоторые шаги установки, связанные с отдельными особенностями дистрибутивов Linux и способами распространения продуктов третьих лиц. Описание команды установки дано в виде группы команд для четырех разных дистрибутивов Linux:

– классический Debian Linux и все дистрибутивы от него производные:

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

– классический Red Hat и все дистрибутивы от него производные:

```
yum install jatoba6-jadog
```

– ALTLinux – дистрибутив, построенный на базе RPM-пакетов, но использующий apt в качестве высокоуровневого менеджера пакетов (команда установки выглядит аналогично Debian):

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

– openSUSE – дистрибутив, построенный на базе RPM-пакетов, но использующий собственный пакетный менеджер zypper:

```
zypper install jatoba6-jadog
```

Все необходимые отличия, связанные с установкой в разных видах дистрибутивов Linux, также даны с пометкой соответствующей ОС.

### 2.1.2. Настройка аутентификации на главном узле (Master)

На главном узле кластера с ролью Master установить требуемый метод аутентификации для пользователя СУБД «jadowg\_user» от имени и справками которого будет выполняться репликация на узлах кластера.

Поддерживаются следующие методы аутентификации:

- MD5;
- SSL;
- LDAP(s);
- RADIUS;
- PAM;
- SSPI;
- GSSAPI.

В качестве примера устанавливается метод аутентификации MD5 (password) в конфигурационном файле pg\_hba.conf и добавляются строки для подключения пользователя СУБД «jadowg\_user» (см. рис.2.2).

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all jadowg_user 127.0.0.1/32 md5
host all jadowg_user 10.116.102.0/24 md5
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all md5
host replication jadowg_user 10.116.102.0/24 md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
```

Рисунок 2.2 – Первоначальные настройки аутентификации

Остальные параметры аутентификации пользователей СУБД устанавливаются в соответствии с требованиями информационной системы и в данном руководстве не рассматриваются.



Дополнительная настройка конфигурационного файла postgresql.conf для настройки архивирования WAL не требуется.

### 2.1.3. Установка пароля системного пользователя ОС «postgres» на главном узле

Установить пароль для системного пользователя ОС «postgres»:

```
sudo passwd postgres
```

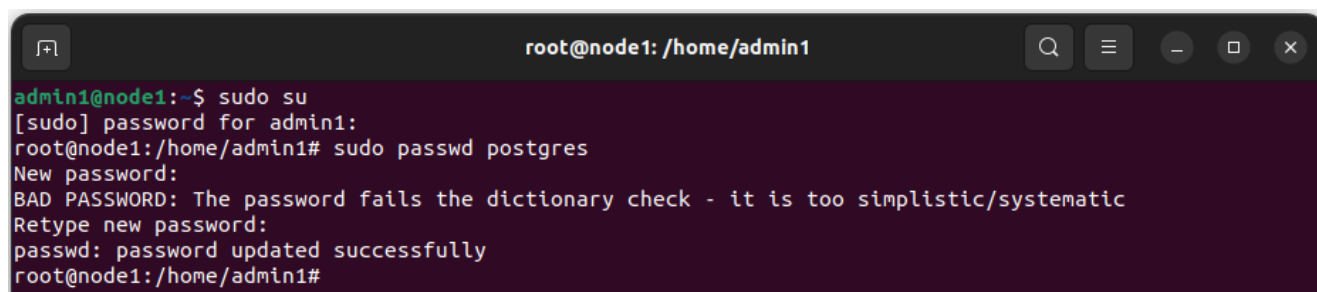


Рисунок 2.3 – Установка пароля для пользователя ОС

После чего перезапустить службу «jatoba-6» и проверить ее статус:

```
# systemctl restart jatoba-6  
# systemctl status jatoba-6
```

### 2.1.4. Установка расширения «jadow» на главном узле

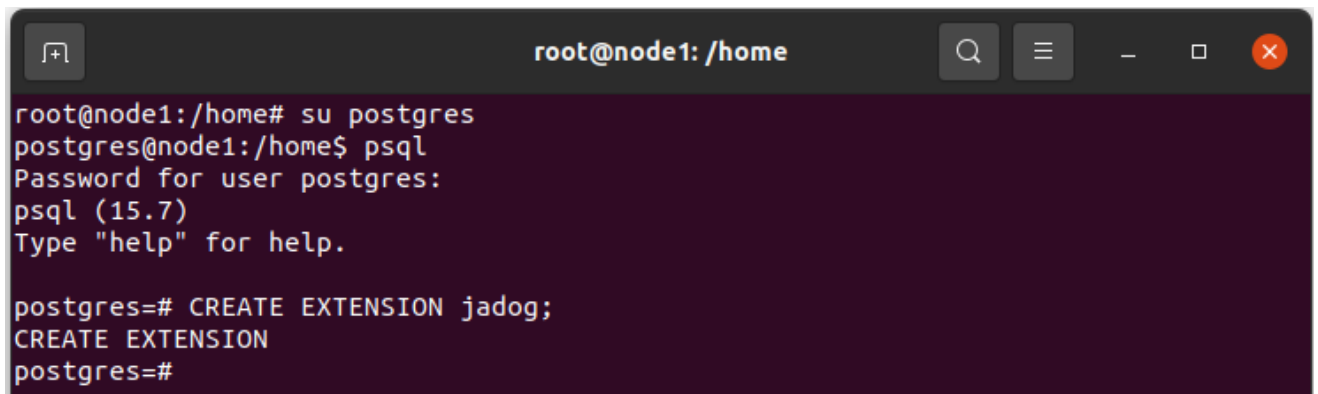
Расширение «jadow» СУБД служит, для:

- проведения репликации между узлами кластера;
- установки дополнительных функций, в том числе функции «add\_jadow\_user».

Расширение «jadow» для СУБД «Jatoba» устанавливается при помощи SQL-команды:

```
CREATE EXTENSION jadow;
```





```
root@node1: /home
root@node1:/home# su postgres
postgres@node1:/home$ psql
Password for user postgres:
psql (15.7)
Type "help" for help.

postgres=# CREATE EXTENSION jalog;
CREATE EXTENSION
postgres=#
```

Рисунок 2.4 – Установка расширения

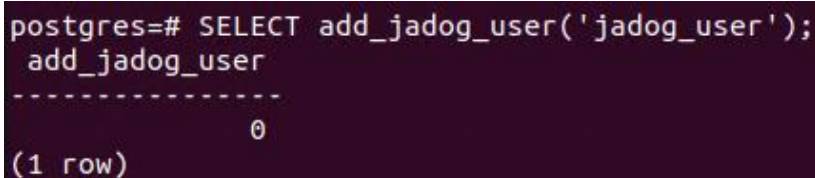
### 2.1.5. Создание пользователя «jalog\_user» на главном узле

С целью соблюдения принципа «Назначения минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование информационной системы» устанавливается специальная функция «add\_jalog\_user». Данная функция СУБД используется для создания специальной роли «jalog\_user» с атрибутами:

- login;
- replication.

Роль создается при помощи SQL-команды:

```
SELECT add_jalog_user('jalog_user');
```



```
postgres=# SELECT add_jalog_user('jalog_user');
add_jalog_user
-----
0
(1 row)
```

Рисунок 2.5 – Создание пользователя «jalog\_user»

Пароль для роли создается при помощи SQL-команды:

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

### Например

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

Таким образом создаваемому пользователю автоматически будут назначены атрибуты и привилегии.

Убедиться в успешном создании пользователя можно через SQL-команду просмотра списка пользователей СУБД:

```
\du
```

```

root@node1: /home/admin1
File Edit View Search Terminal Help
postgres-# \du
                                List of roles
Role name |                               Attributes                               | Member of
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
jadog_user | Replication                               | {pg_read_all_stats}
postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
postgres-#
  
```

Рисунок 2.6 – Просмотр списка пользователей СУБД

### 2.1.6. Установка пакетов компонента «jaDog» на резервных узлах (Slave)

Для резервного узла должен использоваться отдельный сервер с установленной СУБД «Jatoba».

Установка пакетов компонента «jaDog» на резервных узлах выполняется аналогично процедуре установки пакетов компонента на главном узле. Отличие заключается в отсутствии необходимости в каталоге «data» базы данных. В случае наличия данного каталога после установки СУБД на резервном узле его необходимо удалить вместе с содержимым при помощи команды:

```
rm -r /var/lib/jatoba/6/data/
```

Если каталог «data» базы данных не будет удален после установки пакета компонента «jaDog», то в дальнейшем, при добавлении резервного узла в кластер, будет создан новый каталог по пути /var/lib/jatoba/6/data/, а существующий каталог «data» будет переименован.

Установить на резервном узле пароль для системного пользователя ОС «postgres»:

```
sudo passwd postgres
```



**При установке компонента «jaDog» из локального репозитория на резервном узле не выполняются нижеописанные действия:**

- инициализация каталога данных СУБД при помощи команды;

Каталог СУБД должен быть пуст. При синхронизации узлов кластера с главного узла данные будут перенесены.

- проверка лицензионных параметров в конфигурационном файле «postgresql.conf»;

Конфигурационный файл «postgresql.conf» будет скопирован при синхронизации.

- добавление сервиса «jatoba-б» в список автозапуска и проверка статуса сервиса;

Запуск СУБД на резервном узле выполнит компонент «jaDog» и в автоматическом режиме установит службы.

- установка пароля для пользователя СУБД «postgres»;

Установить пароль для пользователя СУБД «postgres» невозможно, т.к. отсутствуют конфигурационные файлы.

Дальнейших проверочных действий на резервных узлах не требуется и можно переходить к действиям, описанным в п. п. 6.8.1, 6.8.2.

## **2.2. Установка/удаление в ОС Microsoft Windows**

Компонент устанавливается в составе СУБД «Jatoba» под управлением ОС Windows при первичной установке:

- в окне «Выбор типа установки» следует выбрать тип установки «Выборочная» (см. рис. 2.7);

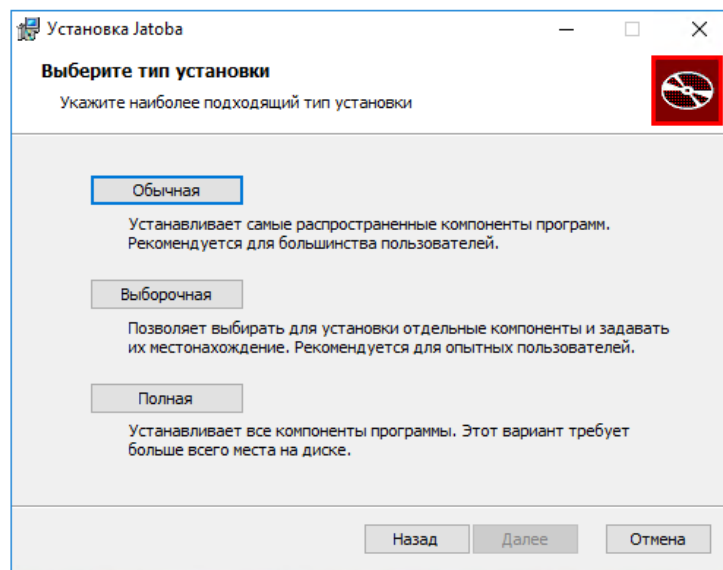


Рисунок 2.7 – Окно выбора типа установки

– в окне «Выборочная установка» выбрать «Управления кластером (jadog)» (см. рис. 2.8);

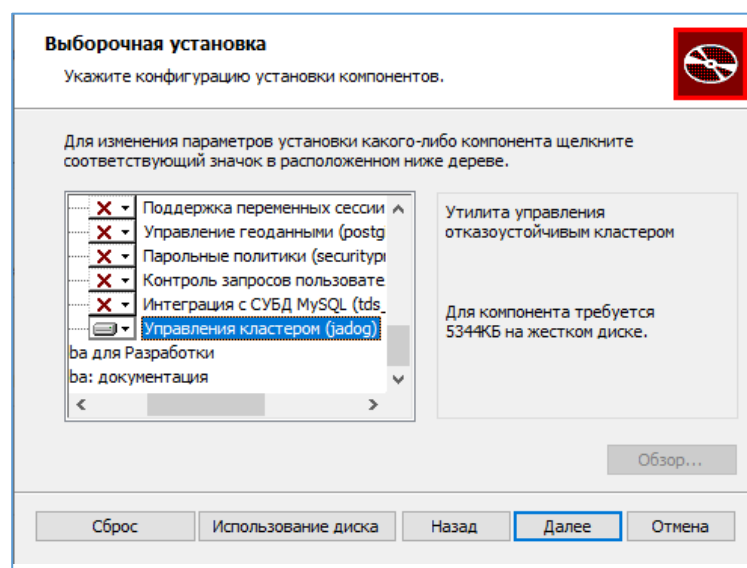


Рисунок 2.8 – Выбор устанавливаемых компонент

– в открывшемся окне «Все готово к установке Jatoba» запустить процесс установки, нажав кнопку «Установить» (см. рис. 2.9);

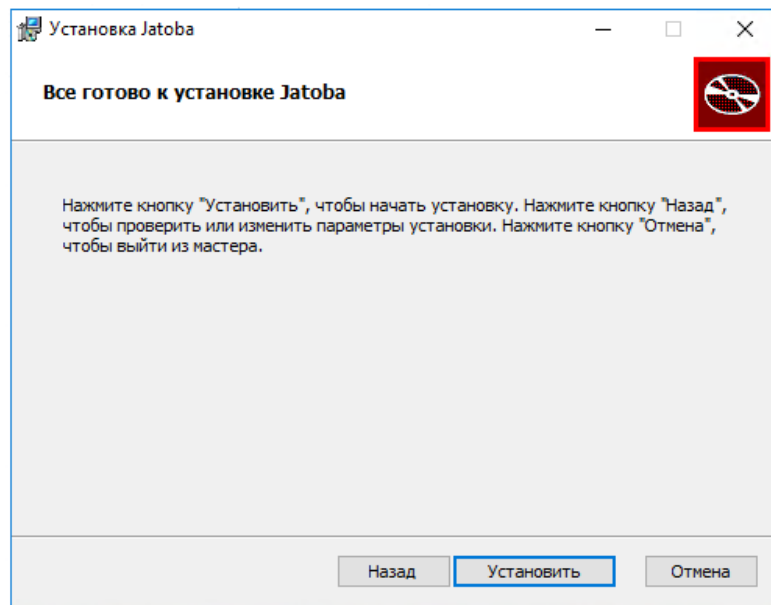


Рисунок 2.9 – Окно «Все готово к установке Jatoba»

### 3. ОС GNU/LINUX АВТОМАТИЧЕСКАЯ НАСТРОЙКА КЛАСТЕРА СУБД «JATOBA» С ПОМОЩЬЮ ФАЙЛОВ КОНФИГУРАЦИИ

СУБД «Jatoba» поддерживает автоматическую настройку как главного узла, так и кластера при помощи использования файлов ответов (Silent make cluster) в формате YML.

Автоматическая настройка кластера при помощи использования файлов конфигурации позволяет автоматизировать процедуру развертывания узлов кластера с заранее заданными параметрами.

Автоматическая настройка кластера выполняется посредством специального сервиса jadow0, который является часть компонента «jaDog».

Специальный сервис jadow0 предназначен для интерпретации и обработки параметров настраиваемого кластера, указанных в файле ответов (Silent make cluster) в формате YML.

Автоматическая настройка кластера при помощи файлов ответов выполняется при помощи следующих файлов автоматизации:

- 1) Настройка экземпляра главного узла кластера (файл ответов «jadow\_setup\_faq.yml») (см. п.п. 3.1-3.4).
- 2) Настройка простого кластера (файл ответов «jadow\_make\_ms\_cluster.yml») (см. п.п. 3.5).

В качестве примера будет использована конфигурация кластера из таблицы 3.1:

Таблица 3.1 – Конфигурация сети для автоматической настройки кластера при помощи файлов конфигурации

№	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	cluster1
2	node2	10.116.102.55/24	255.255.255.0	10.116.102.81/24	Slave	

Предварительными условиями на узле (Master) являются:

- Установленная СУБД Jatoba;
- Выполнена инициализация каталога СУБД Jatoba;
- Установлен компонент «jaDog» (см. 2.1.1);
- Выполнена настройка компонента «jaDog»:

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

- установлено расширение jalog для СУБД (см. 2.1.4);
- создан пользователь jalog\_user (см. 2.1.5);
- Определены необходимые права в pg\_hba.conf (блок replication) (см. 2.1.2);
- Подготовлены и выпущены сертификаты SSL для размещения на главном узле.

Предварительными условиями на резервном узле (Slave) являются:

- Установленная СУБД Jatoba (запуск сервиса СУБД не выполняется, инициализация не выполняется);
- Установлен компонент «jaDog» (см. 2.1.6);
- Подготовлены и выпущены сертификаты SSL для размещения на резервных узлах.

### 3.1. Silent (тихий) режим установки компонента «jaDog» на главном узле





Шаблон установочного файла ответов «jalog\_setup\_faql.yml» расположен в каталоге /usr/jatoba-<ver>/share/jalog/scripts.

Выполнение установки главного узла кластера в тихом режиме требует выполнения следующей последовательности действий:

- 1) скопировать шаблон установочного файла ответов «jalog\_setup\_faql.yml» и внести в него значения параметров главного узла:

```
param_jalog:
  ip: 10.116.102.54/32          # IP адрес. jalog IP address
  public_address: 10.116.102.81/32 # Виртуальный IP адрес
кластера. Public address
  network_interface: ens18      # Наименование сетевого
интерфейса для виртуального IP кластера
  jalog_users:                  # Список пользователей jalog
    - name: admin              # Имя пользователя
      pass: 12345678           # Пароль пользователя. Смените
пароль при переводе системы в промышленную эксплуатацию
  address: 127.0.0.1/32         # Разрешенный адрес для
подключения
  method: sha-256              # Разрешенный способ
подключения
param_replication:
  replication_slot_name: node1  # Имя слота репликации
db_connection_settings:
  user_pass: postgres          # Пароль базы данных
```

```
passfile: /usr/jatoba-6/bin/.pgpass # Файл с паролями базы данных
```

-  Приведенный выше шаблон файла ответов «jadog\_setup\_faq.yml» является примером, значения параметров в котором необходимо изменить под существующие технологические решения.
-  В silent (тихом) режиме установки компонента «jaDog» ошибочные значения в атрибутах прерывают установку. В этом случае пользователю выводятся сообщения об ошибках. После устранения ошибок производится повторный запуск silent (тихий) режим установки компонента «jaDog».

Секция `jadog_users` предназначена для указания данных авторизации пользователя «jaDog» на главном узле. Указанные в ней параметры в дальнейшем будут записаны в файл «`jadog_hba.cfg`» (см. п.п. 4.2.1). В случае если секция `jadog_users` файла ответов «`jadog_setup_faq.yml`» не заполнена применяются параметры по умолчанию (`address: local`, `method: sha-256`).

В случаях если для авторизации будут использоваться сертификаты SSL (см. п.п. 3.3) для параметра `method` секции `jadog_users` файла ответов «`jadog_setup_faq.yml`» необходимо указать значение `ssl`. После чего в файле «`jadog.yml`» необходимо активировать параметр `ssl` указав для него значение `true`.

Для определения имени сетевого интерфейса на главном узле необходимо воспользоваться командой:

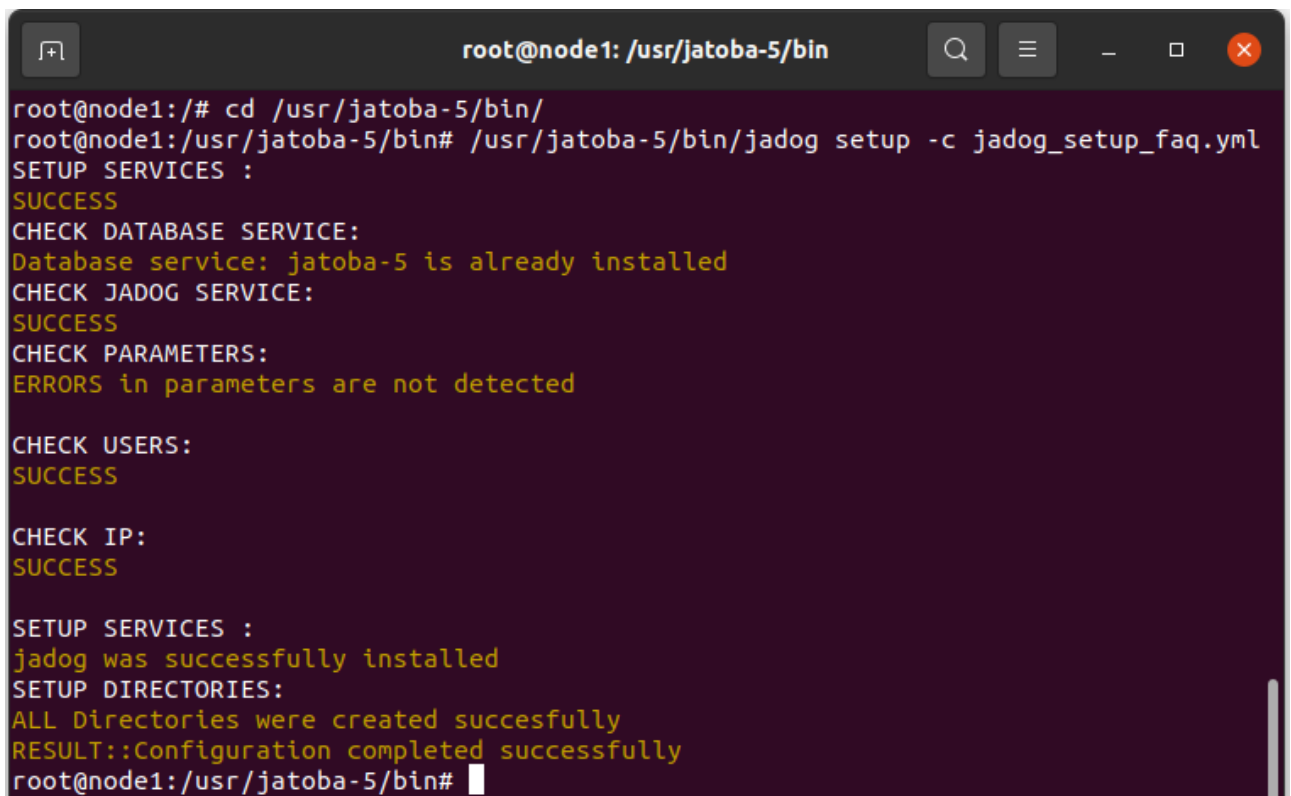
```
ip a
```

Полученное имя сетевого интерфейса необходимо внести в параметр «`network_interface`» в файл ответов «`jadog_setup_faq.yml`».

- 2) Сохранить и разместить файл ответов «`jadog_setup_faq.yml`» в директории `/usr/jatoba-6/bin/`;
- 3) Запустить установку (настройку) компонента «jaDog» на главном узле в тихом режиме выполнив команды в терминале ОС:

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





```
root@node1: /usr/jatoba-5/bin
root@node1:/# cd /usr/jatoba-5/bin/
root@node1:/usr/jatoba-5/bin# /usr/jatoba-5/bin/jadog setup -c jadog_setup_faq.yml
SETUP SERVICES :
SUCCESS
CHECK DATABASE SERVICE:
Database service: jatoba-5 is already installed
CHECK JADOG SERVICE:
SUCCESS
CHECK PARAMETERS:
ERRORS in parameters are not detected

CHECK USERS:
SUCCESS

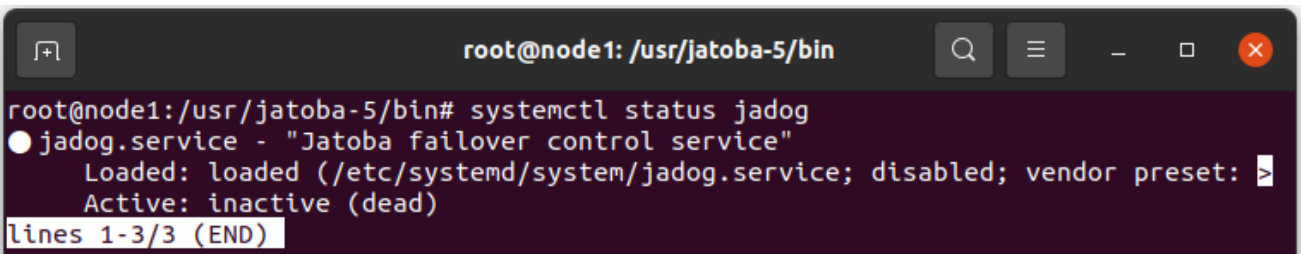
CHECK IP:
SUCCESS

SETUP SERVICES :
jadog was successfully installed
SETUP DIRECTORIES:
ALL Directories were created succesfully
RESULT::Configuration completed successfully
root@node1:/usr/jatoba-5/bin#
```

Рисунок 3.1 – Тихая установка узла кластера

- 4) Проверить статус сервиса «jadog»:

```
systemctl status jadog
```



```
root@node1: /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin# systemctl status jadog
● jadog.service - "Jatoba failover control service"
   Loaded: loaded (/etc/systemd/system/jadog.service; disabled; vendor preset: en
   Active: inactive (dead)
lines 1-3/3 (END)
```

Рисунок 3.2 – Статус службы после

Сервис «jadog» будет загружен, но не активирован и не запущен.

- 5) Добавить в автозагрузку ОС и запустить сервис «jadog» командами в терминале ОС:

```
# systemctl enable jadog
# systemctl start jadog
```

```

root@node1: /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin# systemctl enable jadog
Created symlink /etc/systemd/system/multi-user.target.wants/jadog.service → /etc/s
ystemd/system/jadog.service.
root@node1:/usr/jatoba-5/bin# systemctl start jadog
root@node1:/usr/jatoba-5/bin# systemctl status jadog
● jadog.service - "Jatoba failover control service"
   Loaded: loaded (/etc/systemd/system/jadog.service; enabled; vendor preset: e
   Active: active (running) since Fri 2024-06-21 09:51:47 MSK; 11s ago

```

Рисунок 3.3 – Запуск и проверка статуса сервиса «jadog»

Дальнейшее конфигурирование главного узла кластера возможно через редактирование параметров, описанного в п. 7.14 «Редактирование и получение параметров кластера «jaDog».

### 3.2. Использование файла ответов для автоматизированного формирования кластера

После подготовки главного узла кластера выполняется автоматическая настройка кластера при помощи использования файлов ответов (Silent make cluster) в формате YML.

В качестве примера будет использована конфигурация из таблицы 3.2:

Таблица 3.2 – Конфигурация сети кластера для автоматической настройки кластера при помощи файла конфигурации

№	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	cluster1
2	node2	10.116.102.55/24	255.255.255.0	10.116.102.81/24	Slave	

Предварительными условиями на главном узле «Master» являются:

- Установленная СУБД «Jatoba»;
- Выполнена инициализация каталога СУБД «Jatoba»;
- Установлен компонент «jaDog» (см. 2.1);
- Выполнена настройка компонента «jaDog»:
  - установлено расширение jadog для СУБД (см. 2.1.4);
  - создан пользователь jadog\_user (см. 2.1.5);
- Определены необходимые права в pg\_hba.conf (блок replication) (см. 2.1.2);
- Подготовлены и выпущены сертификаты SSL (см. 3.3).

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

Предварительными условиями на резервном узле «Slave» являются:

- Установленная СУБД «Jatoba» (запуск сервиса СУБД не выполняется, инициализация не выполняется);
- Установлен компонент «jaDog»;
- Подготовлены и выпущены сертификаты SSL.

### 3.3. Подготовка и выпуск сертификатов SSL

Для установления доверенного соединения сервиса jadog0 с узлами проектируемого кластера применяется технология TLS (SSL). При наличии удостоверяющего центра проверка выполняется по всей цепочке сертификатов.

Применение технологии TLS (SSL) при использовании сервиса jadog0 позволяет повысить безопасность и предотвратить выполнение MITM-атак (Man-In-The-Middle attack).

Подготовленные сертификаты SSL должны быть выпущены как для главного узла «Master», так и для резервного узла (-ов) «Slave». В файлах конфигурации CNF (например cert.cnf) сертификатов SSL должны содержаться IP-адреса всех узлов создаваемого кластера в поле:

```
subjectAltName=IP:10.116.102.54,IP:10.116.102.55
```

После указания значений поля subjectAltName выполняется сборка и генерация сертификатов TLS (SSL).



Если выпущенные сертификаты SSL не являются валидными или срок их доверенности истек, то при формировании кластера возникнет ошибка соединения сервиса jadog0 с узлами кластера и будет выполнен откат всех изменений. Также ошибка может возникнуть в том случае, если на каком-либо узле отсутствует требуемый сертификат SSL.

При возникновении данной ситуации необходимо проверить используемые сертификаты SSL, выпустить повторно, и установить их на всех узлах, входящих в проектируемый кластер.

### 3.4. Формирование файла ответов из шаблона



Шаблон файла ответов «jadog\_make\_ms\_cluster.yml» расположен в каталоге /usr/jatoba-<ver>/share/jadog/scripts.

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

Для определения состава и параметров кластера пользователем должен быть вручную создан файл ответов в формате YML с следующим содержимым:

```
# Файл ответов для автоматического формирования кластера jaDog.
# Значения не закомментированные - обязательные. Рекомендуется
установить собственные значения.
# Значения закомментированные - будут использоваться значения
по-умолчанию. Их можно раскомментировать и установить
собственные значения.
# Любые другие дополнительные параметры, которых нет в шаблоне,
возможно размещать в следующих группах:
# - cluster_settings.cluster_name.default_node_params (будет
применено к каждой(!) ноде кластера, включая мастер) или
# - cluster_settings.cluster_name.datacenters.nodes.node_name
(будет применено к конкретной ноде)
# Добавляя параметры необходимо соблюдать нотацию
конфигурационного файла jadog.yml (Обычно он тут: /usr/jatoba-
6/etc/jadog/jadog.yml )
apiVersion: jaDog v3.2 # Версия jaDog
kind: jadog_ms_cluster # Назначение скрипта
cluster_settings:
  - cluster_name: cluster1          # Наименование кластера.
Оbligательный параметр. Кластер может быть один или несколько.
    cluster_master_node: node1      # Нода, являющаяся Мастером.
Оbligательный параметр.
    activated: true                  # Флаг активации кластера
после формирования. true - активировать (назначать
public_address), false - не активировать.
    wal_level = replica              # Определяет, какой
объем информации записывается в WAL, replica/ logical
    jadog_users:                    # Учетные записи для
администрирования jaDog. Создаются УЗ и запись в jadog_hba.cfg.
      - name: admin                  # УЗ можно создать несколько.
Для обеспечения безопасности смените пароль УЗ после установки.
        pass: admin_pass             # Для обеспечения
безопасности смените пароль УЗ после установки.
        address: all
        method: sha-256
      - name: user                    # УЗ можно создать
несколько. Для обеспечения безопасности смените пароль УЗ после
установки.
        pass: user_pass              # Для обеспечения
безопасности смените пароль УЗ после установки.
        address: all
        method: sha-256
```

```

default_node_params:          # Параметры по-умолчанию,
которые будут использовать для каждой ноды. Записи включаются в
нотации конфигурационного файла jadog.yml.
    param_jadog:              # Параметры конфигурирования
экземпляра jaDog.
        public_address: 10.116.102.81/32 # Виртуальный IP адрес
кластера. Public address. Обязательный параметр.
        network_interface: ens18      # Наименование сетевого
интерфейса для виртуального IP кластера. Network interface
name. Обязательный параметр.
        interconnect_user: jadog_user
        database: postgres
        passfile: /usr/jatoba-6/bin/.pgpass # Jatoba passfile
        user: jadog_user              # УЗ для работы с СУБД
        user_pass: jadog_user_pass    # Пароль УЗ для работы с
СУБД. Обязательный параметр. Сменить пароль УЗ в СУБД после
формирования кластера.
    datacenters: #
        - datacenter: dc1            # Уникальное наименование дата-
центра. Датацентр может быть один или несколько.
        nodes:                      # Блок описания узлов
кластера
            - node_name: node1        # Кластерное имя узла,
включаются в нотации конфигурационного файла jadog.yml
            jadog0:                  # Настройки специального
режима jadog0
                port: 65321
                param_jadog:          # Параметры
конфигурирования экземпляра jaDog.
                ip: 10.116.102.54     # IP адрес главного узла
кластера
                network_interface: ens18 # Наименование сетевого
интерфейса
                param_replication:    # Параметры репликации
узла
                    replication_slot_name: node1 # Уникальное в
рамках кластера имя слота репликации.
            - node_name: node2        # Кластерное имя узла,
включаются в нотации конфигурационного файла jadog.yml
            param_jadog:              # Параметры
конфигурирования экземпляра jaDog.
            ip: 10.116.102.55        # IP адрес резервного
узла
            network_interface: ens18  # Наименование сетевого
интерфейса
            param_replication:        # Параметры репликации
узла

```

```
replication_slot_name: node2 # Уникальное в  
рамках кластера имя слота репликации.
```



Приведенный выше шаблон файла ответов «`jadog_make_ms_cluster.yml`» является примером, значения параметров в котором необходимо изменить под существующие технологические решения.

В файле ответов «`jadog_make_ms_cluster.yml`» указывается необходимый набор параметров для развертывания работоспособного кластера.

Если для параметра `method` секции `jadog_users` файла ответов «`jadog_make_ms_cluster.yml`» указывается значение `ssl`, то в секции `param_ssl` необходимо активировать параметр `ssl` указав для него значение `true`.

В случае если при настройке кластера необходимо добавить/удалить узлы необходимо указать их параметры в файле ответов, а также выпустить новые сертификаты SSL (см. п.п 3.3).

### 3.5. Формирование кластера при помощи файла ответов

Формирование и настройка узлов кластера выполняется при помощи специального «нулевого» режима (`jadog0`). Данный режим `jadog0` используется для передачи параметров формирования кластера из файла ответов в формате YML.

Предварительно «нулевой» режим (`jadog0`) должен быть обязательно запущен на всех узлах кластера перед его формированием из файла ответов.

Запуск «нулевого» режима (`jadog0`) производится в терминале ОС при помощи команды:

```
/usr/jatoba-6/bin/jadog jadog0 --sslcafile /<dir2>/ca.crt --  
sslcertfile /<dir2>/client.crt --sslcrlfile /<dir2>/ca.crl --  
sslkeyfile /<dir2>/client.key
```

Где `<dir2>` - путь к каталогу, в котором расположены сертификаты SSL.

После этого на главном узле с ролью «Master» необходимо открыть окно терминала ОС и выполнить чтение параметров из файла конфигурации, и развертывание узлов кластера при помощи команды:

```
/usr/jatoba-6/bin/jadog_ctl create_cluster -T 9999999 -c  
/<dir1>/ jadog_make_ms_cluster.yml --ssl --sslcafile  
/<dir2>/ca.crt --sslcertfile /<dir2>/client.crt --sslcrfile  
/<dir2>/ca.crl --sslkeyfile /<dir2>/client.key
```

Где <dir1> - путь к каталогу, в котором расположен файл ответов в формате YML;  
<dir2> - путь к каталогу, в котором расположены сертификаты SSL.



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
```

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

После завершения формирования кластера на основании файла ответов cluster\_config.yml сервис «jadow0» завершает свою работу.

### 3.6. Изменение состава кластера при помощи файла ответов

Использование в компоненте «jaDog» встроенного функционала обработки настроек файла ответов позволяет администратору в автоматизированном режиме изменять состав существующего кластера:

- Добавлять новые резервные узлы;
- Исключать резервные узлы из кластера;
- Изменять параметры узлов;
- Изменять принадлежность узлов к дата-центру;
- Изменять тип репликации узлов (синхронный, асинхронный, каскадный).

Обновление состава кластера при использовании файла ответов позволяет одновременно добавить/исключить несколько узлов без необходимости ручной настройки, либо внести групповые изменения в их параметры.

Так как изменение состава кластера на основе целевой структуры функционально может состоять из нескольких промежуточных процедур (добавление/удаление узла, изменение типа репликации и так далее), то в случае возникновения в процессе ошибок будет выполнен откат изменений до последней успешной процедуры.



При выполнении изменения состава кластера стоит обращать внимание на правильность используемых IP-адресов и номеров сетевых портов компонента «jaDog». Ошибочные значения могут привести в потере доступа к инфраструктуре кластера.

При выполнении изменения состава кластера не выполняется:

- Удаление учетных записей пользователей кластера (только добавление новых и изменение настроек существующих);
- Удаление главного узла кластера (с ролью MASTER).

Предварительными условиями изменения состава кластера на основе целевой структуры являются:

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



- Выполнение команд на только с главного узла кластера с ролью MASTER;
- Точное соответствие в файле ответов, при помощи которого выполняется изменение состава, названий кластера и его узлов фактическим;

При выполнении изменения состава кластера все выполняемые действия и операции в обязательном порядке записываются в журналы информационной безопасности.

В примере ниже рассматривается кластер, состоящий из дата-центра «dc1», в который требуется добавить новый резервный узел «node3» с IP-адресом 10.116.102.56 и слотом репликации «node3». Остальные настройки узла «node3» аналогичны другим резервным узлам кластера.



Перед началом обновления состава кластера его рекомендуется перевести его в режим технического обслуживания при помощи консольной утилиты «jadog\_ctl» и команды:

```
set maintenance
```

Это обеспечит безопасное снятие нагрузки с узлов кластера и последующую процедуру обновления его состава. После завершения всех работ вывод кластера из режима технического обслуживания производится при помощи команды:

```
reset maintenance
```

### 3.6.1. Добавление нового узла(ов) в кластер с использованием файла ответов



Перед выполнением процедуры добавления нового резервного узла в кластер необходимо убедиться в том, что на нем отсутствует каталог /var/lib/jatoba/6/data (не выполнена инициализация БД), либо для каталога data установлены права доступа drwx----- по умолчанию для системного пользователя postgres.

```
postgres@node3:~$ chmod 700 /var/lib/jatoba/6/data
```

Для того чтобы добавить новый резервный узел «node3» в существующий кластер необходимо:

- 1) Перед началом обновления состава кластера требуется снять нагрузку с БД (отключить ПО, клиентов и т.д.) переведя кластер в режим технического обслуживания.

2) Внести изменения в шаблон файла ответов «`jadog_make_ms_cluster.yml`», с помощью которого был развернут кластер (см. п.п. 3.4). Для этого в секцию «`nodes`» потребуется добавить фрагмент с настройками нового узла:

```
- node_name: node3
  param_jadog:
    ip: 10.116.102.56
    network_interface: ens18
  param_replication:
    replication_slot_name: node3
```

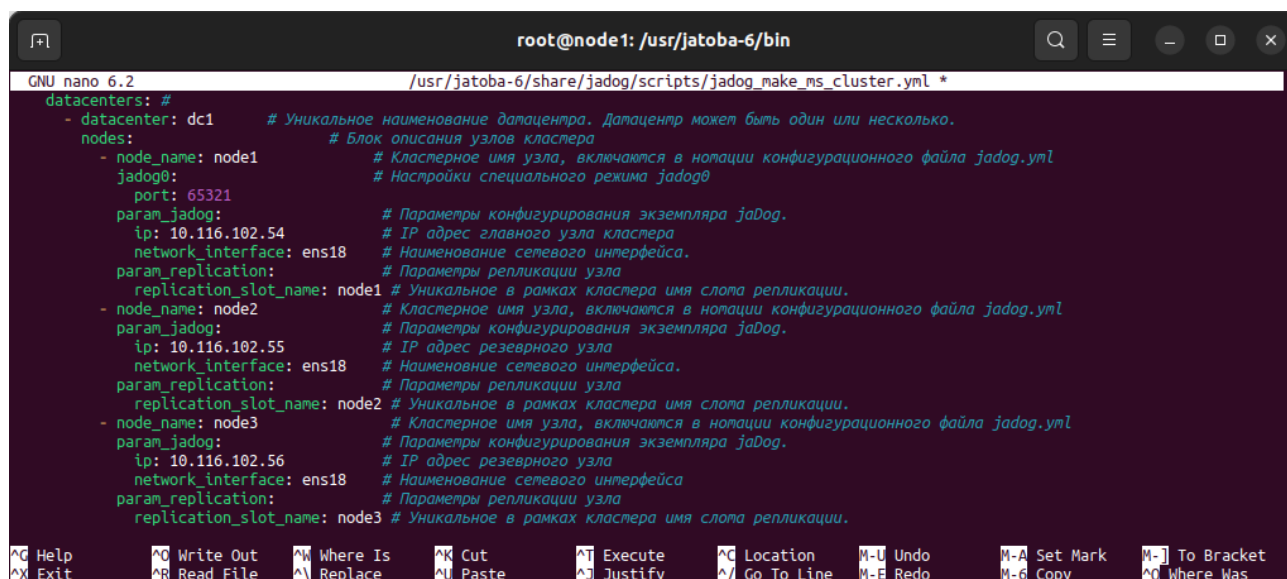


Рисунок 3.4 – Секция «`nodes`» шаблона файла ответов

При добавлении нового резервного узла важно вносить изменения именно в тот шаблон файла ответов, который был использован для автоматизированного создания кластера (см. п.п. 3.4) – это обеспечит согласованность настроек.



В случае отсутствия файла ответов, на основании которого выполнялись разворачивание и настройка кластера, его возможно получить по текущей структуре кластера (см. п.п. 3.7).

- 3) Сохранить изменения в шаблоне файла ответов «`jadog_make_ms_cluster.yml`».
- 4) На каждом узле кластера, которые требуется добавить в него в качестве резервных, в окне терминала ОС на запустить «нулевой» режим (`jadog0`). Запуск «нулевого» режима (`jadog0`) производится при помощи команды:

```
/usr/jatoba-6/bin/jadog jadog0 --sslcafile /<dir2>/ca.crt --  
sslcertfile /<dir2>/client.crt --sslcrllfile /<dir2>/ca.crl --  
sslkeyfile /<dir2>/client.key
```

Где <dir2> - путь к каталогу, в котором расположены сертификаты SSL.

5) На главном узле кластера в окне терминала ОС запустить специальный режим обновления конфигурации при помощи команды:

```
/usr/jatoba-6/bin/./jadog_ctl cluster_update -T 9999999 -q -f  
json --ssl --sslcafile /<dir2>/ca.crt --sslcertfile  
/<dir2>/client.crt --sslkeyfile /<dir2>/client.key -c  
/<dir1>/jadog_make_ms_cluster.yml
```

Где <dir1> - путь к каталогу, в котором расположен файл ответов в формате YML;  
<dir2> - путь к каталогу, в котором расположены сертификаты SSL.



При использовании файла ответов производится проверка внесенных изменений. В случае возникновения ошибок при добавлении резервного узла в кластер производится откат всех изменений, а администратор получает сообщение с текстом ошибки. Событие при этом записывается в журнал компонента «jaDog» с перечнем причин остановки процедур.

При возникновении подобной ситуации необходимо проверить корректность внесенных изменений в секции «nodes» и повторно запустить процедуру чтения параметров из файла ответов (см. пункт 4).

6) После успешного завершения процедур добавления резервного узла(ов) в кластер необходимо проверить состояние всех узлов кластера при помощи команды:

```
cluster status
```

После добавления узла(ов) на нем производится синхронизация информации с другими узлами кластера с целью обеспечения согласованности и непротиворечивости данных.

### 3.6.2. Исключение узла(ов) из кластера с использованием файла ответов

Исключение узла(ов) из кластера не подразумевает какое-либо удаление данных с узла. Узел в процессе выполнения данной процедуры только исключается из кластера.

Здесь в качестве примера будет использован резервный узел «node3», добавленный в кластер согласно п.п. 3.6.1.

Для того чтобы исключить резервный узел «node3» из существующего кластера необходимо:

1) Перед началом обновления состава кластера необходимо снять с него нагрузку с БД (отключить ПО, клиентов и т.д.).

2) Внести изменения в шаблон файла ответов «jadog\_make\_ms\_cluster.yml», с помощью которого был развернут кластер (см. п.п. 3.4). Для этого из секции «nodes» потребуется удалить фрагмент:

```
- node_name: node3
  param_jadog:
    ip: 10.116.102.56
    public_address: 10.116.102.81
    network_interface: ens18
    interconnect_user: admin
  param_replication:
    replication_slot_name: node3
  db_connection_settings:
    user_pass: admin
  param_log:
    logs_type: csv, security.csv
    logs_level: info
```

3) Сохранить изменения в шаблоне файла ответов «jadog\_make\_ms\_cluster.yml».

4) Запустить специальный режим обновления конфигурации на главном узле кластера при помощи команды:

```
/usr/jatoba-6/bin/./jadog_ctl cluster_update -T 9999999 -q -f
json --ssl --sslcafile /<dir2>/ca.crt --sslcertfile
/<dir2>/client.crt --sslkeyfile /<dir2>/client.key -c
/<dir1>/jadog_make_ms_cluster.yml
```

Где <dir1> - путь к каталогу, в котором расположен файл ответов в формате YML;  
<dir2> - путь к каталогу, в котором расположены сертификаты SSL

5) После успешного завершения процедур исключения резервного узла(ов) из кластера необходимо проверить состояние всех узлов кластера при помощи команды:

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

```
cluster status
```

В случаях, когда из кластера с каскадным типом репликации исключаются резервные узлы, при выполнении автоматизированного изменения состава кластера сначала исключаются резервные узлы с ролью «Cascade Slave», затем резервные узлы с ролью «Primary Slave».

### **3.7. Изменение состава/параметров кластера на основе целевой структуры**

Компонент «jaDog» предоставляет администратору кластера (информационной системы) функционал формирования файла ответов в формате YML на основе структуры и состава существующего кластера. Таким образом возможно сохранение структуры и состава кластера для воспроизведения и/или внесения изменений.

#### **3.7.1. Получение целевой структуры и настроек кластера**

Компонент «jaDog» позволяет получить структуру кластера в следующих формах:

– Краткий (Short) – минимальный набор параметров кластера. В этом варианте при выводе будут отсутствовать параметры кластера, которые совпадают с параметрами по умолчанию.

– Полный (Full) – полный набор параметров кластера. В этом варианте файл ответа будет содержать все параметры кластера.

Если форма не указана, то по умолчанию используется краткий вариант.

Для того чтобы отобразить в терминале ОС настройки и состав кластера в краткой форме необходимо:

1) На главном узле кластера с ролью «Master» подключиться к компоненту «jaDog» при помощи консольной утилиты «jadog\_ctl».

2) Выполнить команду получения краткой формы параметров структуры кластера:

```
cluster get structure short
```

```

root@node1: /usr/jatoba-6/bin
root@node1:usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> cluster get structure short
Reply: apiVersion: jaDog v3.3.0
kind: jadog_ms_cluster
cluster_settings:
- cluster_name: cluster1
  cluster_master_node: cluster1_node1
  activated: false
  db_init_conn_string: host=127.0.0.1 port=5432 user=postgres dbname=postgres password=change_on_install
  cluster_referee_nodes: ~
  jadog_users:
    - name: all
      pass: change_on_install
      address: all
      method: sha-256
  datacenters:
    - datacenter: DEFAULT
      nodes:
        - node_name: cluster1_node1
          db_connection_settings:
            user_pass: change_on_install
          param_replication:
            replication_slot_name: node1
          param_jadog:
            ip: 10.116.102.54/24
            public_address: 10.116.102.81/24
            network_interface: ens18
          param_archive:
            wal_archive_directory: /nfs/arhive_wal
            wal_archive_command: cp %p /nfs/arhive_wal/%f
            wal_archive_cleanup_command: /usr/jatoba-6/bin/pg_archivecleanup /nfs/arhive_wal %s
          param_restore:
            wal_restore_command: cp /nfs/arhive_wal/%f %p
          param_synchronous:
            synchronous_commit: remote_apply
            synchronous_type: ANY
            synchronous_commit_max_nodes: 1
        - node_name: cluster1_node2
          db_connection_settings:
            user_pass: change_on_install
          param_replication:
            replication_slot_name: node2

```

Рисунок 3.5 – Получение структуры кластера в краткой форме

Для того чтобы отобразить в терминале ОС параметры и состав кластера в полной форме необходимо:

- 1) На главном узле кластера с ролью «Master» подключиться к компоненту «jaDog» при помощи консольной утилиты «jadog\_ctl».
- 2) Выполнить команду получения полной формы параметров структуры кластера:

```
cluster get structure full
```

### 3.7.2. Сохранение состава и параметров кластера в файл

Для того чтобы сохранить параметры и состав кластера в внешний файл необходимо воспользоваться перенаправлением вывода консольной утилиты «jadog\_ctl». Например:

```
./jadog_ctl -q -C "cluster get structure full" >>
/home/<user>/cluster1_structure_full.yml
```

При выполнении данной команды с использованием консольной утилиты «jadog\_ctl» будет выведен запрос пароля пользователя компонента «jaDog».

```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadow_ctl -q -C "cluster get structure full" >> /home/admin1/cluster1_structure_full.yml
Password:
root@node1:/usr/jatoba-6/bin#
```

Рисунок 3.6 – Перенаправление структуры и настроек кластера в внешний файл YML

```
GNU nano 6.2 /home/admin1/cluster1_structure_full.yml
apiVersion: jadow v3.3.0
kind: jadow_ms_cluster
cluster_settings:
- cluster_name: cluster1
  cluster_master_node: cluster1_node1
  activated: true
  db_init_conn_string: host=127.0.0.1 port=5432 user=postgres dbname=postgres password=change_on_install
  cluster_referee_nodes: ~
  jadow_users:
  - name: all
    pass: change_on_install
    address: all
    method: sha-256
  datacenters:
  - datacenter: DEFAULT
    nodes:
    - node_name: cluster1_node1
      db_connection_settings:
        user_pass: change_on_install
        host: 127.0.0.1
        port: 5432
        database: postgres
        user: jadow_user
        passfile: /usr/jatoba-6/bin/db_passfile
        db_auth_method: password
        ssl_ca_file: ""
        ssl_crl_file: ""
        ssl_cert_file: ""
        ssl_key_file: ""
        ssl_mode: verify-full
      param_path:
        state_path: /usr/jatoba-6/etc/jadow
        module_path: /usr/jatoba-6/share/jadow/scripts
        log_path: /usr/jatoba-6/var/log/jadow
      param_ja_hipe:
        db_name: ""
      param_postgres:
        db_fail_connection_attempts: 5
        db_check_interval: 5000
        db_bin_path: /usr/jatoba-6/bin
        db_data_path: /var/lib/jatoba-6/data
        db_service_name: jatoba-6
      param_replication:
        replication_slot_name: node1
```

Рисунок 3.7 – Содержимое файла ответов, полученного на основе целевой структуры (текущего или другого кластера)

### 3.7.3. Применение кодирования параметров в Base64

При взаимодействии с внешними информационными системами удобно использовать представление структуры и полного набора параметров кластера в двоичном формате Base64. Для этого в формате JSON также используется атрибут message, в котором, в свою очередь, передается информация в двоичном формате Base64. Для этого применяется опция -f json. Например, для краткой формы:

```
./jadow_ctl -f json -C "cluster get structure"
```

Или для полной формы:

```
./jadow_ctl -f json -C "cluster get structure full"
```



При выполнении данной команды с использованием консольной утилиты «jadog\_ctl» будет выведен запрос пароля пользователя компонента «jaDog».

Декодирование информации о структуре и набору параметров кластера, содержащейся в атрибуте message JSON-ответа, из двоичного формата Base64 осуществляется на стороне внешней информационной системы.

[illegible]

Рисунок 3.8 – Вывод параметров и структуры кластера в атрибуте message JSON-ответа в виде кодированного текста Base64

### 3.7.4. Ограничения на изменения параметров

Полученный файл ответов, содержащий структуру и параметры кластера в дальнейшем можно изменять в соответствии с планируемой или существующей инфраструктурой.

Использование в компоненте «jaDog» встроенного функционала обработки настроек файла ответов позволяет администратору в автоматизированном режиме изменять состав существующего кластера:

- Изменять параметры узлов;
- Изменять принадлежность узлов к дата-центру;
- Изменять тип репликации резервных узлов (синхронный, асинхронный,

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



- Добавлять новые учетные записи;
- Изменять параметры учетных записей.

При внесении изменений в параметры аутентификации учетных записей доступно (см. ниже пример секции «`jadog_users`»):

- Список IP-адресов, с которых разрешен доступ «`address:`»;
- Разрешенный метод аутентификации учетной записи «`method:`».

Назначение указанных параметров приведено в п.п. 2.1.2.



Полученный таким образом файл ответов на основе текущего кластера не содержит пароль пользователей компонента «`jaDog`».

**Пример секции «`jadog_users`»:**

```
jadog_users:
- name: main_user
  pass: change_on_install
  address: all
  method: sha-256
```

Данный параметр, перед последующим использованием сформированного файла ответов, необходимо изменить.

Аналогично необходимо определить пароль пользователя СУБД.

**Пример секции «`db_connection_settings`»:**

```
db_connection_settings:
  user_pass: change_on_install
```



Удаление в файле ответов из секции «`jadog_users`» существующих учетных записей пользователей приводит к остановке процедуры обновления параметров кластера.

### 3.7.5. Изменение параметров кластера с использованием файла ответов

После внесения изменений (например IP-адресов узлов кластера, дата-центров и так далее) файл ответов можно использовать для:

- Настройки нового кластера (см. п.п. 3.5);
- Изменения параметров действующего кластера.

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

Процедура изменения параметров действующего кластера с использованием подготовленного файла ответов на основе целевой структуры ничем не отличается от процедуры создания нового (см. п.п. 3.5).

### 3.8. Выполнение SQL-скриптов при автоматизированной настройке кластера

В файле ответов возможно указать расположение SQL-скриптов, которые будут выполнены после завершения настройки общего кластера, обычного кластера, а также составляющих их узлов. SQL-скрипты позволяют автоматизировать установку расширений, а также настройку параметров БД, СУБД или кластера в целом.

Описание настройки общего кластера приведено во второй части документа «Компонент jaDog. Управление режимом работы узлов кластера (версия 3.4, часть 2)»  
643.72410666.00067-07 98 02-02

Порядок выполнения SQL-скриптов:

- 1) Инициализация БД.
- 2) Выполнение скрипта(ов) из раздела `cluster_setting.sql_scripts`, если он(и) есть.
- 3) Выполнение скрипта(ов) из раздела `default_cluster_params.sql_scripts`, если он(и) есть.
- 4) Создание кластера и выполнение его настройки (создание пользователя `jadog_user`, установка служб `jadog` и так далее).
- 5) Сборка бандла (при необходимости).
- 6) Выполнение скрипта(ов) из раздела `bundle.bundle_scripts.cluster_name.sql_scripts`, если он(и) есть.

SQL-скрипты могут выполняться:

– Для общего кластера с использованием параметра `bundle:bundle_scripts`. Например:

```
bundle:
  bundle_scripts:
    /home/user/clusters/ja_hipe_cluster/bundle_script.sql      # Раздел
пользовательских скриптов, выполняемых на уровне создаваемого
бандла. Скрипты определяются администратором кластера.
```

SQL-скрипты уровня общего кластера выполняются с правами пользователя, указанного в параметре `cluster_settings.db_init_conn_string` секции `default_cluster_params` файла ответов. SQL-скрипты уровня общего кластера устанавливаются администратором СУБД.

– Для каждого кластера в отдельности с использованием параметра `cluster_settings:sql_scripts`. Например:

```
cluster_settings:
- cluster_name: cluster1 # Наименование кластера. Обязательный
  параметр. Кластер может быть один или несколько.
  sql_scripts:
  -
    /home/user/clusters/ja_hipe_cluster/cl_coord_inside_script.sql
```

SQL-скрипты должны располагаться файловой системе главного узла

Результат выполнения SQL-скрипта регистрируется в журнале компонента «jaDog». Запись выполнения SQL-скрипта в журнале содержит следующую информацию:

- Список запросов и состав SQL-скрипта;
- Путь к каталогу, в котором располагается SQL-скрипт;
- Дата и время запуска;
- Узел кластера, на котором производится запуск;
- Результат выполнения SQL-скрипта.

## 4. ПРОВЕРКА СОСТОЯНИЯ КЛАСТЕРА

### 4.1. Проверка работоспособности кластера

Проверку работоспособности кластера возможно провести путем:

- создания тестовой таблицы на главном сервере;
- проверки созданной таблицы и пользователя «jadowg\_user».

Тестовая таблица создается при помощи SQL-команды:

```
create table table1(id integer, name varchar(40));
```

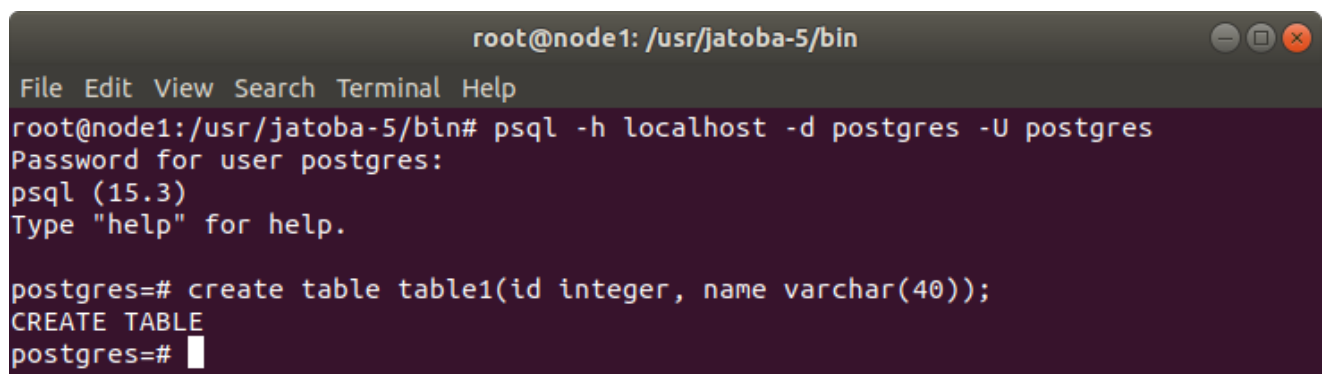


Рисунок 4.1 – Создание тестовой таблицы

И проверить созданную таблицу на резервном сервере SQL-командой:

```
SELECT * from table1;
```

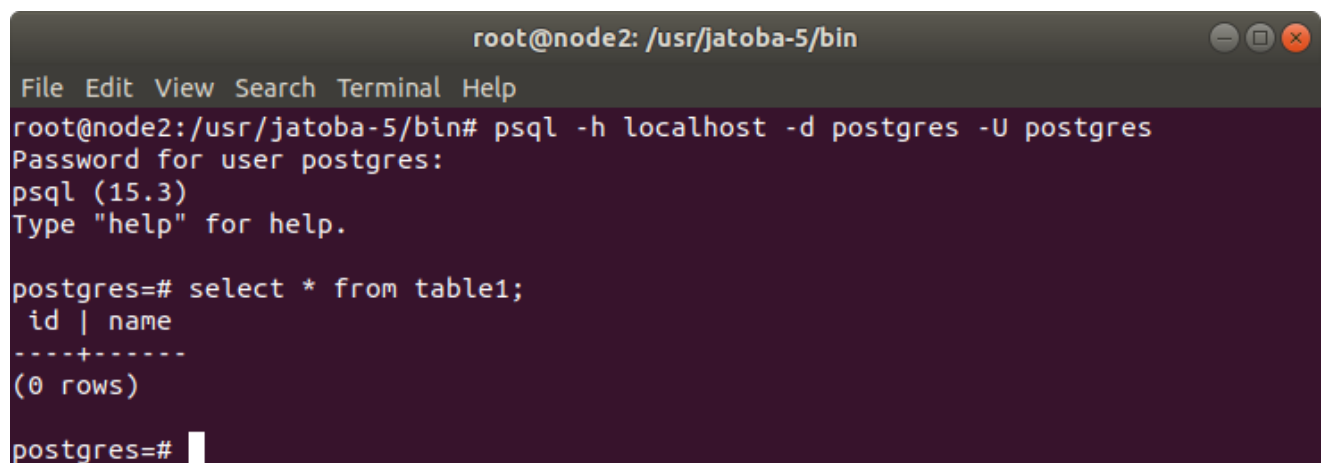


Рисунок 4.2 – Список таблиц

Проверить список пользователей на резервном сервере SQL-командой:

```
\du
```

```

root@node2: /usr/jatoba-5/bin
File Edit View Search Terminal Help
root@node2: /usr/jatoba-5/bin# psql -h localhost -d postgres -U postgres
Password for user postgres:
psql (15.3)
Type "help" for help.

postgres=# select * from table1;
 id | name
----+-----
(0 rows)

postgres=# \du
                                List of roles
Role name | Attributes | Member of
-----+-----+-----
jadog_user | Replication | {pg_read_all_stats}
postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

postgres=#

```

Рисунок 4.3 – Список пользователей

## 4.2. Подключение к узлу кластера

Подключение к узлу кластера регулируется конфигурационным файлом аутентификации `jadog_hba.cfg`

### 4.2.1. Конфигурационный файл аутентификации (`jadog_hba.cfg`)

Конфигурационный файл `jadog_hba.cfg` заполняется параметрами аутентификации после создания основного файла конфигурации `jadog.yml` (см. п. 6.7.14.1).

Конфигурационный файл аутентификации `jadog_hba.cfg` расположен по пути:

```
/usr/jatoba-6/etc/jadog/jadog_hba.cfg
```

Данный файл регулирует аутентификацию пользователей только к локальному компоненту «jaDog».

Общий принцип обеспечения информационной безопасности, реализуемый в файле аутентификации пользователей `jadog_hba.cfg`: запрещено все, что не разрешено явно.

В случае размещения на сервере нескольких экземпляров компонентов «jaDog» для каждого из них формируется отдельный файл `jadog_hba.cfg`.

Структура файла строится на примере оригинального файл `pg_hba.conf` СУБД.

# USER	ADDRESS	METHOD
user	IP-address /IP-mask	auth-method

В поле «USER» по умолчанию установлен пользователь «admin» устанавливаемый при конфигурировании узла кластера (см.п. 6.7.2.6). По требованию могут быть установлены:

- имена конкретных пользователей;
- значение «all» для всех пользователей.

В поле «ADDRESS» вносятся IP адреса подсетей и/или IP-адрес кластера в формате CIDR-маски IPv4.

Варианты указания IP-маски:

```
all
localhost
0.0.0.0/0
192.168.0.0/16
192.168.11.201/24
```

В поле «METHOD» указывается один из возможных методов аутентификации к компоненту «jaDog»:

- sha-256;
- ssl.

```

root@node1: /home/admin1
GNU nano 6.2 /usr/jatoba-6/etc/jadog/jadog_hba.cfg *
# synopsis follows.
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which jaDog user names they can use, which
# jaDog they can access. Records take one of these forms:
#
# USER ADDRESS METHOD
# (The uppercase items must be replaced by actual values.)
#
# The first field is the connection type:
#
# USER can be "all", a user name.
#
# ADDRESS specifies the set of hosts the record matches. It can be a
# host name, or it is made up of an IP address and a CIDR mask that is
# an integer (between 0 and 32 (IPv4) that
# specifies the number of significant bits in the mask.
# Instead of a CIDR-address, you can write "all" to match any IP address,
# "samehost" to match any of the server's own IP addresses,
# or "samenet" to match any address in any subnet that the server is
# directly connected to.
#
# METHOD can be "sha-256", "ssl".
#
# This file is read on server startup and when the server receives a
# SIGHUP signal. If you edit the file on a running system, you have to
# SIGHUP the server for the changes to take effect, run reload in jadog_ctl or
# "jadog_ctl -c reload"
#
# Put your actual configuration here
# -----
# USER ADDRESS METHOD
all all sha-256

```

Рисунок 4.4- Конфигурационный файл аутентификации jadog\_hba.cfg



При возникновении дублирования параметров или пересечения свойств доступа к использованию принимается последний установленный параметр при прочих равных условиях. Таким образом если первой строкой заданы определенные правила для аутентификации отдельного или отдельных учетных записей, а в столбце #USER последней строки конфигурационного файла вместо прямого имени учетной записи указано значение 'all', то для всех пользователей (и персонифицированных и технологических) будет применено данное правило.

Если весь компонент «jaDog» настроен на аутентификацию с использованием сертификатов SSL, то администратор может установить метод аутентификации "password" в файле jadog\_hba.cfg.

После внесения изменений в файл jadog\_hba.cfg необходимо выполнить перезагрузку конфигурации компонента «jaDog» при помощи команды:

```

cd /usr/jatoba-6/bin
./jadog_ctl -C " reload jadog"

```



Команда применима для случая если узел не входит в кластер или является главным узлом (с ролью «Master») кластера.

Допускается использование гибридной аутентификации.

Конфигурационный файл аутентификации `jadog_hba.cfg` также может формироваться на основании установочного файла ответов (см. п.п 3.1) в режиме `silent` (тихой) установки через сервис «`jadog`».

Если файл конфигурации не задан, то для установленных учетных записей в компоненте «`jaDog`» доступ запрещен (принцип «запрещено все, что не разрешено»).

#### 4.2.2. Подключение по умолчанию

Подключение по умолчанию выполняется вызовом утилиты «`jadog_ctl`», без дополнительных параметров командой:

```
./jadog_ctl
```

Вне зависимости от текущего сеанса пользователя, утилита по умолчанию выберет имя пользователя «`admin`» и предложит ввести только пароль.

```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.2.0
Terminate parser with \q command
Use \h for help

Password:
Recv response for response id: 0
Reply: You're authenticated (id = 0)
10.116.102.54>
```

Рисунок 4.5 – Подключение по умолчанию

#### 4.2.3. Подключение используя параметры (`ip port 'login'`)

Подключение к узлу кластера через утилиту «`jadog_ctl`», возможно выполнить через указание параметров в строке подключения. Синтаксис строки подключения аналогичен синтаксису подключения к СУБД:



```
-h [host_ip/host_name/hosts] -p [Jadog PORT number (port)] -U [user]
```

### Например

```
./jadog_ctl -h localhost -p 54321 -U admin
```

```
root@node1:/usr/jatoba-5/bin# ./jadog_ctl -h localhost -p 54321 -U admin
Supported Jadog version 3.1.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
localhost> █
```

Рисунок 4.6 – Подключение к узлу кластера с параметрами «ip port 'login'»

#### 4.2.4. Подключение используя параметры соединения (connect ip port 'login')

Подключение к утилите «jadog\_ctl» производится командами:

```
# cd /usr/jatoba-6/bin
# ./jadog_ctl
```

Если имеется УЗ администратора кластера «admin», то утилита не потребует ввода имени пользователя, а выведет запрос установленного пароля.

```
root@node1:/usr/jatoba-5/bin# ./jadog_ctl
Supported Jadog version 3.1.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> █
```

Рисунок 4.7 – Запрос пароля администратора кластера

В этом случае соединение уже будет установлено и для подключения к другому узлу потребуется отключиться от текущего узла командой:

```
disconnect
```

После чего ввести команду соединения с узлом создаваемого кластера:

```
connect [host_ip/host_name/hosts] [Jadog PORT number (port)] [user]
```

### Например

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

Находясь на хосте 10.116.102.54 подключимся к текущему узлу кластера по умолчанию:

```
# cd /usr/jatoba-6/bin  
# ./jadog_ctl
```

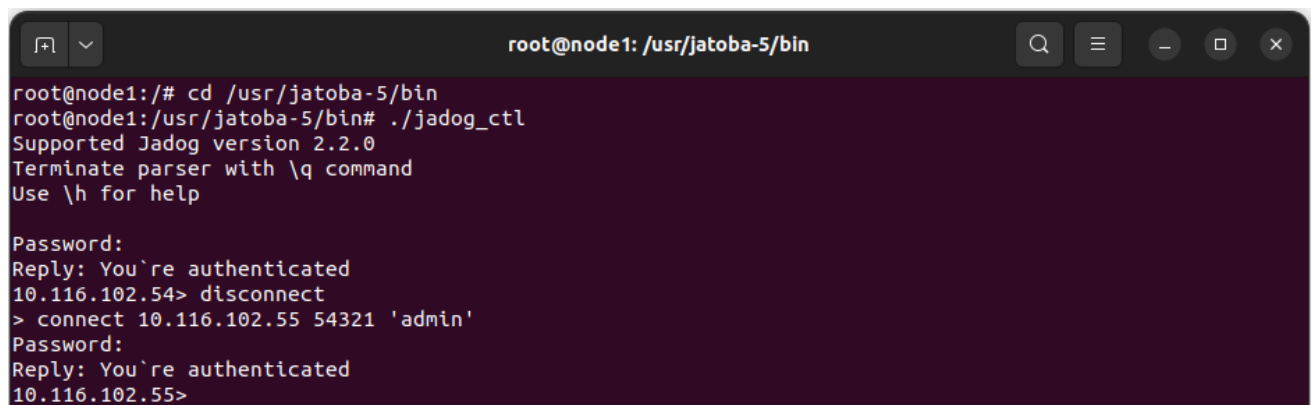
Отключимся от текущего узла:

```
disconnect
```

Находясь в утилите «jadog\_ctl», проведем подключение к узлу кластера на хосте 10.116.102.55:

```
connect 10.116.102.55 54321 'admin'
```

После успешного подключения к узлу кластера, в командной строке будет выведен IP-адрес подключения (рисунок 4.8).



```
root@node1: /usr/jatoba-5/bin  
root@node1:/# cd /usr/jatoba-5/bin  
root@node1:/usr/jatoba-5/bin# ./jadog_ctl  
Supported Jadog version 2.2.0  
Terminate parser with \q command  
Use \h for help  
Password:  
Reply: You're authenticated  
10.116.102.54> disconnect  
> connect 10.116.102.55 54321 'admin'  
Password:  
Reply: You're authenticated  
10.116.102.55>
```

Рисунок 4.8 – Подключение к выбранному узлу кластера

#### 4.2.5. Подключение по SSL используя переменные окружения

Подключение по SSL используя переменные окружения может использоваться в том случае, когда компонент настроен на хранение использование сертификатов для SSL соединения.

Например, ключи и сертификаты в директории:

```
/localrepo
```

Потребуется задать переменные окружения в консоли ОС командами:

```
# export JADOG_CTL_CAFILE='/localrepo/ca.crt'
```

```
# export JADOG_CTL_CERTFILE='/localrepo/server.crt'  
# export JADOG_CTL_CRLFILE='/localrepo/ca.crl'  
# export JADOG_CTL_KEYFILE='/localrepo/server.key'
```

После чего подключение к утилите «jadog\_ctl» по SSL станет доступно по команде в терминале ОС:

```
./jadog_ctl --ssl true
```

#### 4.2.6. Подключение по SSL используя параметры по умолчанию

Подключение по SSL с параметрами по умолчанию может использоваться в том случае, когда компонент настроен на хранение сертификатов в директории:

```
~/jadog/
```

Настройка компонента на SSL соединение описана в подразделе 7.20 «Настройка SSL соединения в ОС GNU/Linux для узлов кластера». Компонент «jaDog» поддерживает SSL соединение между узлами кластера.

#### 4.2.7. Подключение по SSL используя строку подключения

Подключение по SSL с помощью строки подключения может использоваться в том случае, когда компонент настроен на хранение использование сертификатов для SSL соединения.

Настройка компонента на SSL соединение описана в подразделе 7.20 «Перезагрузка компонентов кластера».

В случае необходимости изменения конфигурационных параметров компонента «jaDog» и/или СУБД «Jatoba» выполняется перезагрузка компонентов кластера или его узлов.

Для перезагрузки компонентов кластера используется утилита «jadog\_ctl».

Количество подключений утилиты «jadog\_ctl» к любому доступному узлу кластера не ограничено.

В журнале сообщений компонента «jaDog» фиксируются все попытки подключения/отключения удаленной утилиты «jadog\_ctl». Также в журнале сообщений

компонента «jaDog» фиксируются все процедуры по перезагрузке сервисов компонента «jaDog» при выполнении команд в утилите «jadog\_ctl».

Использование для перезагрузки служб узлов кластера утилиты «jadog\_ctl» является предпочтительным и наиболее безопасным.



Перезагрузка кластера всегда производится при подключении только к главному узлу

С целью обеспечения безопасной и стабильной работы кластера применяется ограничение перезагрузки сервиса компонента «jaDog» и/или его частей во время выполнения команд смены состояний узлов кластера или нестабильном состоянии этих узлов. О невозможности выполнения перезагрузки сервисов компонента «jaDog» и/или его частей пользователю отображаются сообщения.

#### 4.3. Перезагрузка компонентов «jaDog» в кластере

Для того чтобы выполнить перезагрузку всего кластера необходимо выполнить следующие действия:

1) Убедиться в отсутствии подключений со стороны пользователей на общий адрес кластера. При необходимости отключить все в ручном режиме (с ожиданием завершения или аварийно с прерыванием). Также можно удалить адрес с интерфейса во избежание новых подключений.

2) На главном узле подключиться к компоненту «jaDog» при помощи утилиты «jadog\_ctl»;

3) Далее выполнить команду:

```
reload jadog on cluster
```

4) При выводе запроса «Try to reload jadog on cluster. Are you sure [Y/N]?>» нажать клавишу Y и клавишу Enter. Для отмены перезагрузки кластера нажать клавишу N.

```
root@node1:/usr/jatoba-5/bin# ./jadog_ctl
Supported Jadog version 3.1.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> reload jadog on cluster
Try to restart clusterAre you sure [Y/N]?>
```

Рисунок 4.9 – Подтверждение перезагрузки кластера

- 5) Результатом успешной перезагрузки кластера будет вывод сообщения «Reply: Command has been successfully completed».
- 6) После перезагрузки проверить статус узлов кластера:

```
cluster status
```



Важно отметить, что перезагрузка компонентов «jaDog» в кластере, выполняемая с его главного узла, относится только к его кластеру. Для перезагрузки компонентов других кластеров необходимо выполнить подключение к их главным узлам и выполнить аналогичные команды.

Также перезагрузка кластера возможна в режиме прямого выполнения команд утилитой «jadog\_ctl». В этом случае команда будет выглядеть следующим образом:

```
./jadog_ctl -h localhost -p 54321 -U admin -W [пароль] -C
"reload jadog on cluster"
```



При выполнении перезагрузки кластера в режиме прямого выполнения команд утилитой «jadog\_ctl» подтверждение «Try to reload jadog on cluster. Are you sure [Y/N]?>» не выводится

Другой способ перезагрузки кластера заключается в использовании средств ОС и строгой последовательности выполняемых команд как на главном, так и на резервных узлах кластера. Данный способ следует использовать в случае невозможности перезагрузки кластера средствами утилиты «jadog\_ctl».



При отключении узлов кластера сначала отключается компонент «jaDog», затем СУБД «Jatoba». Компонент «jaDog» поддерживает служебное соединение с

СУБД «Jatoba» и при отключении в обратном порядке возникает предпосылка отказа.

Кроме того, при отключении сначала отключаются службы на резервном узле, затем на главном.

Порядок перезагрузки кластера при использовании средств ОС следующий:

1) Убедиться в отсутствии подключений со стороны пользователей на общий адрес кластера. При необходимости отключить все в ручном режиме (с ожиданием завершения или аварийно с прерыванием). Также можно удалить адрес с интерфейса во избежание новых подключений.

2) Остановить службу компонента «jaDog» на резервном узле:

– для ОС Windows:

```
sc stop jadogservice
```

– для ОС Ubuntu:

```
systemctl stop jadog
```

3) Остановить службу компонента «jaDog» на главном узле:

– для ОС Windows:

```
sc stop jadogservice
```

– для ОС Ubuntu:

```
systemctl stop jadog
```

4) Остановить службу СУБД «Jatoba» на главном узле:

– для ОС Windows:

```
sc stop JatobaServer
```

– для ОС Ubuntu:

```
systemctl stop jatoba-6
```

5) Остановить службу СУБД «Jatoba» на резервном узле:

- для ОС Windows:

```
sc stop jatobaserver
```

- для ОС Ubuntu:

```
systemctl stop jatoba-6
```

Порядок включения служб узлов кластера после внесения требуемых изменений в параметры компонента «jaDog» и/или СУБД «Jatoba» выполняется в обратном порядке:

1) Включение службы СУБД «Jatoba» на главном узле:

- для ОС Windows:

```
sc start JatobaServer
```

- для ОС Ubuntu:

```
systemctl start jatoba-6
```

2) Включение служб СУБД «Jatoba» на резервном узле:

- для ОС Windows:

```
sc start JatobaServer
```

- для ОС Ubuntu:

```
systemctl start jatoba-6
```

3) Включение службы компонента «jaDog» на главном узле:

- для ОС Windows:

```
sc start jadogservice
```

- для ОС Ubuntu:

```
systemctl start jadog
```

4) Включение службы компонента «jaDog» на резервном узле:

- для ОС Windows:

```
sc start jadogservice
```

- для ОС Ubuntu:

```
systemctl start jadog
```

После включения необходимо дождаться завершения процедуры RECOVER в компоненте «jaDog» по настройке состояния «Master» (главный) и «Slave» (резервный) соответственно. В журнале сообщений будут сообщения следующего содержания:

- на главном узле:

```
2021-02-04 17:51:43 DEBUG: RECOVERY: local node state: Master
2021-02-04 17:51:43 DEBUG: RECOVERY: send state to
10.116.102.54
2021-02-04 17:51:43 DEBUG: RECOVERY: remote node state: Slave
2021-02-04 17:51:43 DEBUG: RECOVERY: over
```

- на резервном узле:

```
2021-02-04 17:53:33 DEBUG: RECOVERY: local node state: Slave
2021-02-04 17:53:33 DEBUG: RECOVERY: send state to
10.116.102.55
2021-02-04 17:53:33 DEBUG: RECOVERY: remote node state: Master
2021-02-04 17:53:33 DEBUG: RECOVERY: over
```

#### 4.3.1. Перезагрузка компонента «jaDog» на узле кластера

Утилита «jadog\_ctl» предоставляет администратору возможности перезагрузки компонентов кластера на узлах.

Для того чтобы выполнить перезагрузку компонента кластера «jaDog» на узле необходимо выполнить следующие действия:

- 1) На главном узле подключиться к компоненту «jaDog» при помощи утилиты «jadog\_ctl»;
- 2) Далее выполнить команду:

```
reload jadog '[name_node]'
```



3) При выводе запроса «Try to reload jalog on node "test\_node1". Are you sure [Y/N]?>» нажать клавишу Y и клавишу Enter. Для отмены перезагрузки кластера нажать клавишу N.

```
root@node1:/usr/jatoba-5/bin# ./jalog_ctl
Supported Jalog version 3.1.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> reload jalog 'test_node1'
Try to reload jalog on node "test_node1". Are you sure [Y/N]?> █
```

Рисунок 4.10 – Подтверждение перезагрузки узла кластера

4) Результатом успешной перезагрузки узла кластера будет вывод сообщения «Reply: Command has been successfully completed».

5) После перезагрузки проверить статус узла кластера:

```
cluster status
```

Также перезагрузка узла кластера возможна в режиме прямого выполнения команд утилитой «jalog\_ctl». В этом случае команда будет выглядеть следующим образом:

```
./jalog_ctl -h localhost -p 54321 -U admin -W [пароль] -C
"reload jalog 'test_node1'"
```

Альтернативной командой в режиме прямого выполнения команд утилитой «jalog\_ctl» является указанием вместо названия узла в кластере его IP-адреса и сетевого порта. В этом случае команда будет выглядеть следующим образом:

```
./jalog_ctl -h localhost -p 54321 -U admin -W [пароль] -C
"reload jalog [IP-адрес] [порт]"
```



При выполнении перезагрузки кластера в режиме прямого выполнения команд утилитой «jalog\_ctl» подтверждение «Try to reload jalog on cluster. Are you sure [Y/N]?>» не выводится

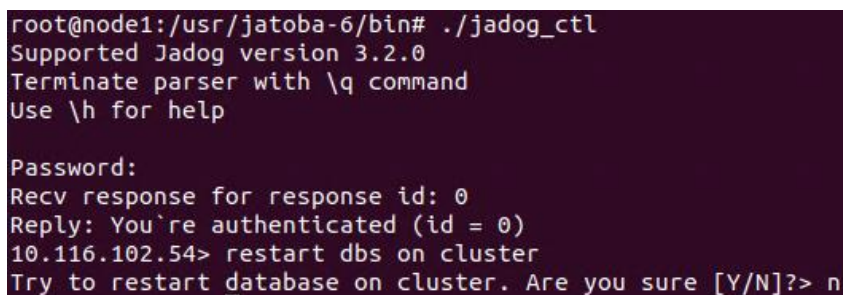
### 4.3.2. Перезагрузка СУБД на узлах кластера

Перезагрузка СУБД на узлах кластера при помощи утилиты «jadow\_ctl» выполняется в следующей последовательности:

- 1) На главном узле подключиться к компоненту «jaDog» при помощи утилиты «jadow\_ctl»;
- 2) Далее выполнить команду:

```
restart dbs on cluster
```

При выводе запроса «Try to restart database on cluster. Are you sure [Y/N]?>» нажать клавишу Y и клавишу Enter. Для отмены перезагрузки СУБД на узлах кластера нажать клавишу N.



```
root@node1:/usr/jatoba-6/bin# ./jadow_ctl
Supported Jadow version 3.2.0
Terminate parser with \q command
Use \h for help

Password:
Recv response for response id: 0
Reply: You're authenticated (id = 0)
10.116.102.54> restart dbs on cluster
Try to restart database on cluster. Are you sure [Y/N]?> n
```

Рисунок 4.11 – Подтверждение перезагрузки СУБД на узлах кластера

После завершения перезагрузки СУБД на узлах кластера необходимо проверить статус узлов кластера при помощи команды:

```
cluster status
```

Также перезагрузка СУБД на узлах кластера возможна в режиме прямого выполнения команд утилитой «jadow\_ctl». В этом случае команда будет выглядеть следующим образом:

```
./jadow_ctl -h localhost -p 54321 -U admin -W [пароль] -C  
"restart dbs on cluster"
```



При выполнении перезагрузки СУБД на узлах кластера в режиме прямого выполнения команд утилитой «jadow\_ctl» подтверждение «Try to restart database on cluster. Are you sure [Y/N]?>» не выводится

#### 4.4. Перегрузка конфигурации СУБД

Перегрузка конфигурации СУБД требуется в случае внесения изменений в конфигурационные файлы СУБД «Jatoba».

При использовании внешних информационных систем доступна возможность перегрузки конфигурации СУБД в кластере или на отдельном узле при помощи REST API. Примеры запросов REST API представлены в таблице 7.1 во второй части документа 643.72410666.00067-07 98 02-02.

##### 4.4.1. Перегрузка конфигурации СУБД на узле кластера

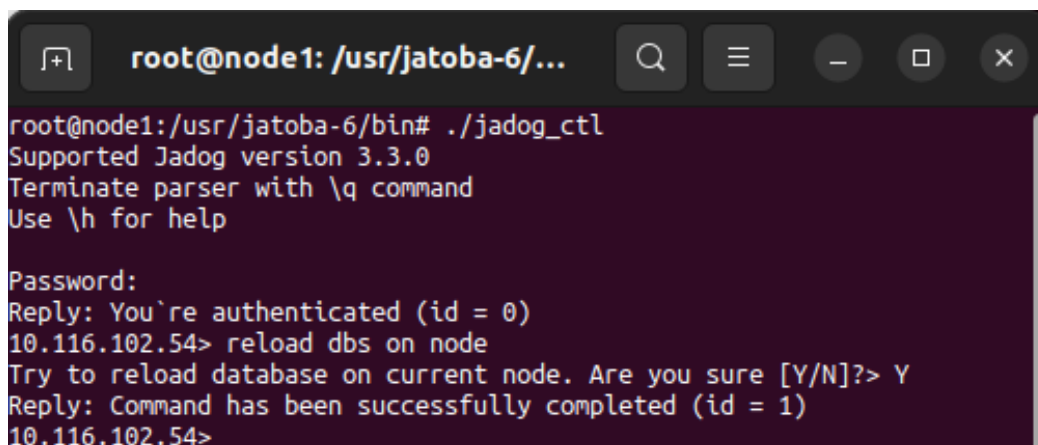
Перегрузка конфигурации СУБД на отдельном узле кластера возможна как на главном узле кластера, так и на резервных, с указанием их имени или IP-адреса и номера сетевого порта.

Перегрузка конфигурации СУБД на главном узле кластера при помощи утилиты «jadog\_ctl» выполняется в следующей последовательности:

- 1) На главном узле кластера подключиться к компоненту «jaDog» при помощи утилиты «jadog\_ctl»;
- 2) Далее выполнить команду:

```
reload dbs on node
```

При выводе запроса «Try to reload database on current node. Are you sure [Y/N]?> » нажать клавишу Y и клавишу Enter. Для отмены перегрузки конфигурации СУБД на узле кластера необходимо нажать клавишу N.



```
root@node1: /usr/jatoba-6/...
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> reload dbs on node
Try to reload database on current node. Are you sure [Y/N]?> Y
Reply: Command has been successfully completed (id = 1)
10.116.102.54>
```

Рисунок 4.12 – Подтверждение перегрузки конфигурации СУБД на отдельном узле кластера

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

После завершения перезагрузки конфигурации СУБД на главном узле кластера необходимо проверить статус узлов кластера при помощи команды:

```
cluster status
```

Также перезагрузка конфигурации СУБД на главном узле кластера возможна в режиме прямого выполнения команд утилитой «jalog\_ctl». В этом случае команда будет выглядеть следующим образом:

```
./jalog_ctl -h localhost -p 54321 -U admin -W [пароль] -C  
"reload dbs on node"
```



При выполнении перезагрузки конфигурации СУБД на главном узле кластера в режиме прямого выполнения команд утилитой «jalog\_ctl» подтверждение «Try to reload database on current node. Are you sure [Y/N]?>» не выводится

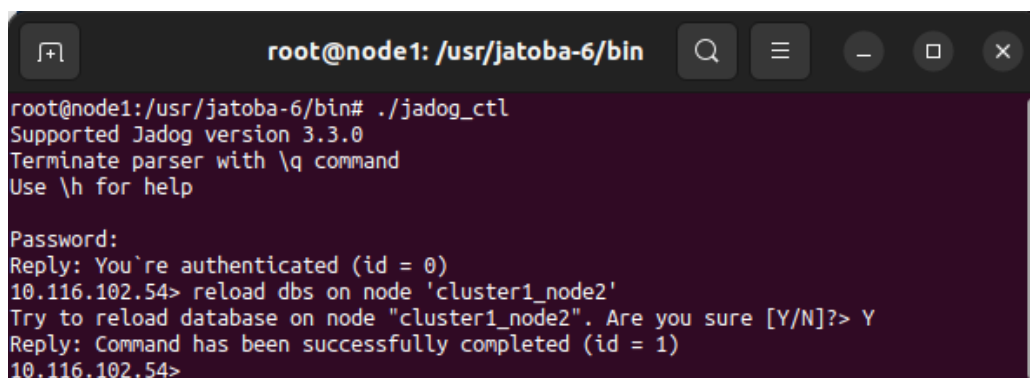
Перезагрузка конфигурации СУБД на отдельном узле с указанием его названия или IP-адреса и номера сетевого порта возможна с использованием следующих команд утилиты «jalog\_ctl»:

1) Перезагрузка конфигурации СУБД на отдельном узле с указанием его названия в кластере

```
reload dbs on node 'node_name'
```

### Пример:


```
reload dbs on node 'cluster1_node2'
```



```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> reload dbs on node 'cluster1_node2'
Try to reload database on node "cluster1_node2". Are you sure [Y/N]?> Y
Reply: Command has been successfully completed (id = 1)
10.116.102.54>
```

Рисунок 4.13 – Подтверждение перезагрузки конфигурации СУБД на отдельном узле кластера с указанием его названия

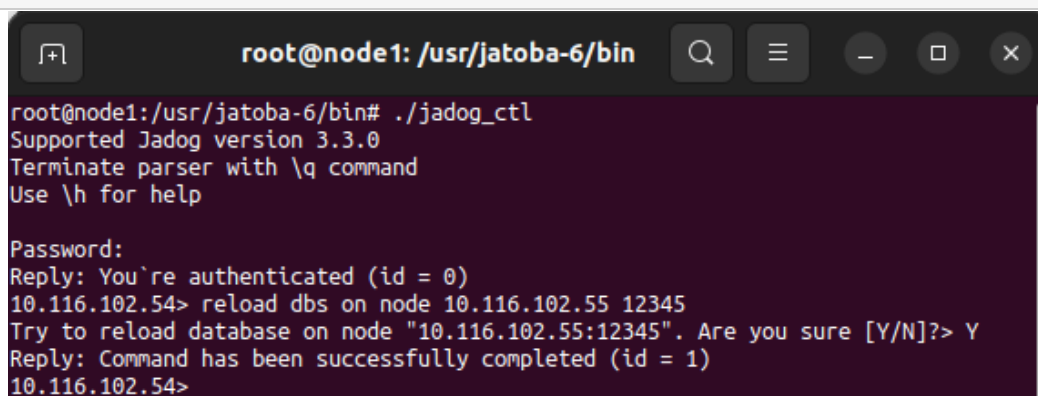
 При выполнении перезагрузки конфигурации СУБД на отдельном узле кластера с указанием его названия данный параметр представляет собой название кластера + название узла в кластере. Например «cluster1\_node2»

2) Перезагрузка конфигурации СУБД на отдельном узле с указанием IP-адреса и номера сетевого порта

```
reload dbs on node [ip] [port]
```

**Пример:**

```
reload dbs on node 10.116.102.55 12345
```



```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> reload dbs on node 10.116.102.55 12345
Try to reload database on node "10.116.102.55:12345". Are you sure [Y/N]?> Y
Reply: Command has been successfully completed (id = 1)
10.116.102.54>
```

Рисунок 4.14 – Подтверждение перезагрузки конфигурации СУБД на отдельном узле кластера с указанием его IP-адреса и номера сетевого порта

В режиме прямого выполнения команд утилитой «jadog\_ctl» перезагрузка конфигурации СУБД на отдельном узле с использованием названия узла выглядит следующим образом:

```
./jadog_ctl -h localhost -p 54321 -U admin -W [пароль] -C "  
reload dbs on node 'node10' "
```

Или для случая указания IP-адреса и номера сетевого порта узла:

```
./jadog_ctl -h localhost -p 54321 -U admin -W [пароль] -C "  
reload dbs on node 10.116.102.55 12345"
```



При выполнении перезагрузки конфигурации СУБД на главном узле кластера в режиме прямого выполнения команд утилитой «jadog\_ctl» подтверждение не выводится

После завершения перезагрузки конфигурации СУБД на узлах кластера необходимо проверить статус этих узлов кластера при помощи команды:

```
cluster status
```

#### 4.4.2. Перезагрузка конфигурации СУБД в кластере

Перезагрузка конфигурации СУБД кластера требуется для случаев, когда необходимо выполнить чтение конфигураций СУБД на всех узлах кластера.

Перезагрузка конфигурации СУБД на главном узле кластера при помощи утилиты «jadog\_ctl» выполняется в следующей последовательности:

- 1) На главном узле кластера подключиться к компоненту «jaDog» при помощи утилиты «jadog\_ctl»;
- 2) Далее выполнить команду:

```
reload dbs on cluster
```

При выводе запроса «Try to reload database on cluster. Are you sure [Y/N]?>» нажать клавишу Y и клавишу Enter. Для отмены перезагрузки конфигурации СУБД на узле кластера необходимо нажать клавишу N.

После завершения перезагрузки конфигурации СУБД кластера необходимо проверить статус узлов кластера при помощи команды:

```
cluster status
```

Также перезагрузка конфигурации СУБД кластера возможна в режиме прямого выполнения команд утилитой «jalog\_ctl». В этом случае команда будет выглядеть следующим образом:

```
./jalog_ctl -h localhost -p 54321 -U admin -W [пароль] -C  
"reload dbs on cluster "
```



При выполнении перезагрузки конфигурации СУБД на главном узле кластера в режиме прямого выполнения команд утилитой «jalog\_ctl» подтверждение «Try to reload database cluster. Are you sure [Y/N]?>» не выводится

#### 4.5. Остановка и запуск СУБД кластера

При сложной структуре кластера, корректная, последовательная остановка СУБД кластера для проведения регламентных работ или правильной остановки для перезагрузки серверов является необходимым условием корректной работы.

Для выполнения данных задач применяются специальные команды остановки и запуска СУБД кластера.

##### 4.5.1. Остановка СУБД кластера, главного или резервного узлов

Остановка СУБД кластера, главного или резервного узлов выполняется через консольную утилиту «jalog\_ctl» при помощи следующих команд:

- 1) Остановка СУБД во всем кластере выполняется при помощи команды:

```
stop dbs on cluster
```

В случае остановки СУБД на всем кластере сначала выполняется перевод кластера в режим технического обслуживания «MAINTENANCE». После чего по порядку останавливаются СУБД на каждом узле кластера.

- 2) Остановка СУБД на главном узле кластера выполняется при помощи команды:

```
stop dbs on node
```

- 3) Остановка СУБД на отдельном узле кластера с указанием его IP-адреса и номера сетевого порта выполняется при помощи команды:

```
stop dbs on node [ip] [port]
```

**Пример команды:**

```
stop dbs on node 10.116.102.55 12345
```

4) Остановка СУБД на отдельном узле кластера с указанием имени узла выполняется при помощи команды:

```
stop dbs on node ['node_name']
```

**Пример команды:**

```
stop dbs on node 'node12'
```



При выполнении остановки СУБД кластера, главного или резервного узлов выполняется запись в журнал работы компонента «jaDog» (расположение /usr/lib/jatoba/6/data/log/), а также в журнал событий информационной безопасности.

#### 4.5.2. Запуск СУБД кластера, главного или резервного узлов

Запуск СУБД кластера, главного или резервного узлов выполняется через консольную утилиту «jadog\_ctl» при помощи следующих команд:

1) Запуск СУБД во всем кластере выполняется при помощи команды:

```
start dbs on cluster
```

В случае запуска СУБД на всем кластере сначала выполняется запуск СУБД на каждом узле кластера. После успешного запуска СУБД кластер выводится из режима технического обслуживания «MAINTENANCE» в штатный.

2) Запуск СУБД на главном узле кластера выполняется при помощи команды:

```
start dbs on node
```

3) Запуск СУБД на отдельном узле кластера с указанием его IP-адреса и номера сетевого порта выполняется при помощи команды:

```
start dbs on node [ip] [port]
```

**Пример команды:**

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



```
start dbs on node 10.116.102.55 12345
```

4) Запуск СУБД на отдельном узле кластера с указанием имени узла выполняется при помощи команды:

```
start dbs on node ['node_name']
```

**Пример команды:**

```
stop dbs on node 'node12'
```



При выполнении запуска СУБД кластера, главного или резервного узлов выполняется запись в журнал работы компонента «jaDog» (расположение /usr/lib/jatoba/6/data/log/), а также в журнал событий информационной безопасности.

## 5. ОПИСАНИЕ И ПРИМЕНЕНИЕ УТИЛИТЫ «JADOG\_CTL»

«jadowg\_ctl» – утилита управления кластером.

Режимы работы утилиты:

- 1) Обычный режим – подключение и ввод команд осуществляет пользователь в консоли.
- 2) Режим прямого выполнения команд – пользователь передает параметры подключения и команду в качестве аргументов при запуске. После выполнения команды утилита возвращает ответ и завершает работу.

Параметры, которые используются при запуске утилиты, указаны в таблице 5.1:

Таблица 5.1 – Параметры запуска утилиты «jadowg\_ctl»

Наименование	Описание
-h	Хост для подключения
-p	Порт для подключения
-U	Пользователь для подключения
-W	Пароль для подключения
-C	Команда для выполнения и выхода
-T	Время на выполнение команды -C (мс)



Утилита запускается как с параметрами, описанными выше, так и без них.

Работа утилиты при различных параметрах запуска представлена в таблице 5.2:

Таблица 5.2 – Работа утилиты «jadowg\_ctl» при различных параметрах

Параметры	Описание
-h -p -U	Запрос пароля, далее последует подключение Параметры не могут использоваться по отдельности
-h -p -U -W	Автоматическое подключение и вход в консоль
-h -p -U -W -C	Подключение, выполнение команды и выход со стандартным временем – 10 секунд
-h -p -U -W -C -T	Подключение, выполнение команды и выход с заданным временем
	Если не задать параметр -W в сценариях: «-h -p -U -W -C» и «-h -p -U -W -C -T», то будет запрашиваться пароль из консоли.

Параметры	Описание
--help	Справка по всем параметрам

Если не были использованы параметры запуска «-h -p -U», тогда требуется соединение через консольную команду «connect»:

```
connect [host_ip/host_name] [Jadog PORT number (port)] [user]
```

Пример:

```
connect 127.0.0.1 54321 'root'
```

После этого будет запрошен пароль из консоли и последует подключение. Если пароль был введен неверно, то он будет запрошен повторно. Дается не более трех попыток ввода пароля. После этого команда «connect» завершается с ошибкой.

Таблица 5.3 – Параметры переменных окружения «jadog\_ctl»

Наименование	Описание
JADOG_CTL_HOST	Хост для подключения по умолчанию
JADOG_CTL_PORT	Порт для подключения по умолчанию
JADOG_CTL_USER_NAME	Пользователь для подключения по умолчанию
JADOG_CTL_CERTFILE	Путь до файла сертификата, который Jadog использует для подключения к СУБД
JADOG_CTL_KEYFILE	Путь до файла с ключом, который jadog_ctl использует для подключения к СУБД
JADOG_CTL_CAFILE	Путь до сертификата центра сертификации, который jadog_ctl использует для подключения к СУБД
JADOG_CTL_CRLFILE	Путь до CRL файла, который Jadog использует для подключения к СУБД

### 5.1. Использование имени хоста и доменного имени

В операциях с кластером могут использоваться:

- IP-адрес;
- Имя хоста;
- DNS имя хоста.

#### Пример 1

Подключение по IP-адресу:

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

```
connect 10.116.102.54 54321 'admin'
```



```
root@node1: /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin# ./jadog_ctl
Supported Jadog version 2.1.0
Terminate parser with \q command
Use \h for help

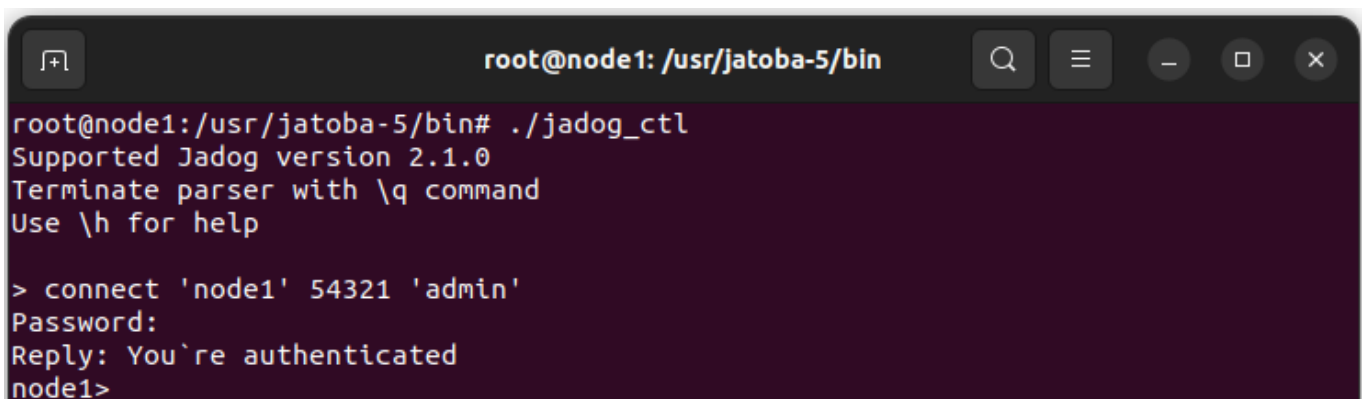
> connect 10.116.102.54 54321 'admin'
Password:
Reply: You're authenticated
10.116.102.54>
```

Рисунок 5.1 – Подключение к выбранному узлу кластера

## Пример 2

Подключение по имени хоста:

```
connect 'node1' 54321 'admin'
```

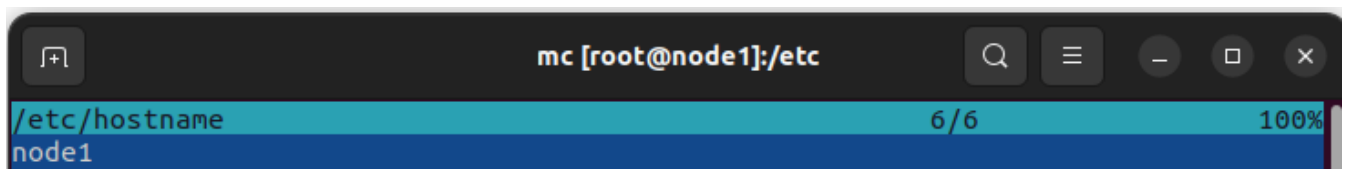


```
root@node1: /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin# ./jadog_ctl
Supported Jadog version 2.1.0
Terminate parser with \q command
Use \h for help

> connect 'node1' 54321 'admin'
Password:
Reply: You're authenticated
node1>
```

Рисунок 5.2 – Подключение по имени хоста

В GNU/Linux имя хоста хранится в файле /etc/hostname:



```
mc [root@node1]:/etc
/etc/hostname 6/6 100%
node1
```

Рисунок 5.3 – Содержание файла /etc/hostname

В ОС Windows вывести имя хоста возможно через консоль командой:

```
hostname
```

## Пример 3

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

Подключение по DNS имени хоста:

```
connect 'admin1-Standard-PC-i440FX-PIIX-1996' 54321 'admin'
```

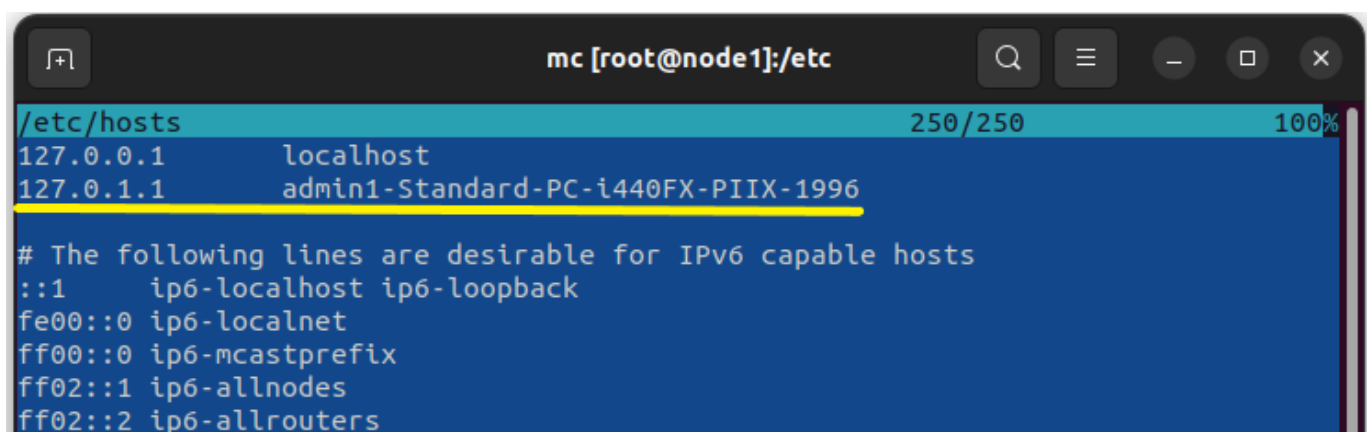


```
root@node1: /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin# ./jadog_ctl
Supported Jadog version 2.1.0
Terminate parser with \q command
Use \h for help

> connect 'admin1-Standard-PC-i440FX-PIIX-1996' 54321 'admin'
Password:
Reply: You're authenticated
admin1-Standard-PC-i440FX-PIIX-1996>
```

Рисунок 5.4 – Подключение по DNS имени хоста

DNS имя хоста хранится в файле /etc/hosts:



```
mc [root@node1]:/etc
/etc/hosts 250/250 100%
127.0.0.1    localhost
127.0.1.1    admin1-Standard-PC-i440FX-PIIX-1996

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Рисунок 5.5 – Содержание файла /etc/hosts

В ОС Windows имя хоста хранится в файле:

```
C:\Windows\System32\drivers\etc\hosts
```

Использование имени хоста приведено в подразделе 5.2 настоящего документа.

В синтаксисе команды подключения по имени хоста и по DNS имени хоста, указанные имена указываются в одинарных кавычках.

## 5.2. Использование заданного порта в «Jadog PORT number (port)»

Параметр «Jadog PORT number (port)» устанавливается в пункте меню 2 «Inter-jadog communication setting» (см. п. 6.7.2) в «Подменю 2) Jadog PORT number (port)».

По умолчанию установлен параметр – [12345], но параметр является редактируемым и допускается установка требуемого параметра.

При настройке кластера параметр «Jadog PORT number (port)» целесообразно использовать идентичным на всех узлах кластера.

### 5.3. Описание команд утилиты «jadog\_ctl»

Описание команд утилиты «jadog\_ctl» представлено в таблице 5.4.

Таблица 5.4 – Описание команд утилиты «jadog\_ctl»

Наименование	Синтаксис команды	Описание
<b>Общие команды</b>		
disconnect		Отключение от jadog
clear		Очистить консоль jadog_ctl от данных
\password		Изменение пароля текущего пользователя
\q		Выход из утилиты «jadog_ctl»
\h (п. 7.5)		Подсказка по консольным командам
connect [ip] [port] ['login'] (п. 4.2.4)	connect [host_ip/host_name/hosts] [Jadog PORT number (port)] [user]	Подключение к jadog с заданным ip от имени пользователя. Пример: <pre>connect 10.116.102.54 54321 'admin' connect 127.0.0.1 54321 'admin'</pre>
<b>Команды управления кластером</b>		
add cluster ['cluster_name'] (п. 6.11)	add cluster 'cluster_name'	Создание абстрактного кластера и привязка к нему имени. Пример: <pre>add cluster 'test'</pre>

Наименование	Синтаксис команды	Описание
delete cluster	delete cluster	Удаление кластера без добавленных узлов. Выполняется на главном узле.
cluster add master [ip] [port] (п. 6.12)	cluster add master [host_ip/host_name/hosts] [Jadog PORT number (port)]	Добавление узла с IP-адресом и сетевым портом к кластеру в роли главного. Пример: <pre>cluster add master 10.116.102.54 12345</pre>
cluster add master [ip] [port] as ['node_name'] (п. 6.12)	cluster add master [ip] [port] as ['node_name']	Добавление узла с именем, IP-адресом и сетевым портом к кластеру в роли главного. Пример: <pre>cluster add master 10.116.102.54 12345 as 'masternode'</pre>
cluster add cascade slave [ip] [port] primary [primary_ip] [primary_port] as ['node_name']	cluster add cascade slave [ip] [port] primary [primary_ip] [primary_port] as ['node_name']	Добавление в кластер узла с присвоением имени и с указанными IP-адресом и сетевым портом в подчиненного каскадного узла (Slave Cascade) по отношению к главному узлу (с указанными IP-адресом и сетевым портом). Пример: <pre>cluster add cascade slave 10.116.102.55 12345 primary 10.116.102.54 12345 as 'node_slave1'</pre>



Наименование	Синтаксис команды	Описание
cluster add cascade slave [ip] [port] primary ['primary_node_name'] as ['node_name']	cluster add cascade slave [ip] [port] primary ['primary_node_name'] as ['node_name']	Добавление в кластер узла с присвоением имени и с указанными IP-адресом и сетевым портом в подчиненного каскадного узла (Slave Cascade) по отношению к главному узлу (с указанным именем главного узла). Пример: <pre>cluster add cascade slave 10.116.102.55 12345 primary 'node_master' as 'node2'</pre>
cluster add cascade slave [ip] [port] primary ['primary_node_name']	cluster add cascade slave [ip] [port] primary ['primary_node_name']	Добавление в кластер узла с указанными IP-адресом и сетевым портом в подчиненного каскадного узла (Slave Cascade) по отношению к главному узлу (с указанным именем главного узла). Пример: <pre>cluster add cascade slave 10.116.102.55 12345 primary 'node_master'</pre>
cluster add referee [ip] [port] as ['node_name']	cluster add referee [ip] [port] as ['node_name']	Добавление узла с присвоением имени и с IP-адресом и сетевым портом к кластеру в роли рефери (referee). Пример: <pre>cluster add referee 10.116.102.55 12345 as 'node2'</pre>

Наименование	Синтаксис команды	Описание
cluster add referee [ip] [port]	cluster add referee [ip] [port]	<p>Добавление узла с IP-адресом и сетевым портом к кластеру в роли рефери (referee).</p> <p>Пример:</p> <pre>cluster add referee 10.116.102.55 12345</pre>
cluster add slave 'ip' 'port' (п. 6.15)	cluster add slave [host_ip/host_name/hosts] [Jadog PORT number (port)]	<p>Добавление узла с IP-адресом и сетевым портом к кластеру в роли резервного (slave). Выполняется в асинхронном режиме.</p> <p>Пример:</p> <pre>cluster add slave 10.116.102.55 12345</pre>
cluster add slave [ip] [port] as ['node_name']	cluster add slave [ip] [port] as ['node_name']	<p>Добавление узла с IP-адресом и сетевым портом к кластеру в роли резервного (slave) с присвоением имени узла. Выполняется в асинхронном режиме.</p> <p>Пример:</p> <pre>cluster add slave 10.116.102.55 12345 as 'node2'</pre>
cluster delete node 'ip' (п. 7.3)	cluster delete node [host_ip/host_name/hosts] [Jadog PORT number (port)]	<p>Удаление узла из кластера с заданным IP-адресом. Выполняется в асинхронном режиме.</p> <p>Пример:</p>

Наименование	Синтаксис команды	Описание
		cluster delete node 10.116.102.55 12345
cluster delete node [node_name] (п. 7.3)	cluster delete node [node_name]	Удаление узла из кластера с заданным именем. Выполняется в асинхронном режиме. Пример: cluster delete node 'node2'
cluster activate (п. 6.14)	cluster activate	Присвоение публичного IP-адреса главному узлу кластера. Пример: cluster activate
cluster deactivate (п. 7.2)	cluster deactivate	Деактивация публичного IP-адреса главного узла кластера. Пример: cluster deactivate
cluster status (п. 6.13)	cluster status	Просмотр списка всех узлов в кластере и их текущего состояния. Пример: cluster status
clusters list (п. 7.1)	clusters list	Просмотр список имен кластеров. Пример:

Наименование	Синтаксис команды	Описание
		<code>clusters list</code>
reset maintenance (п. 7.16.3)	<code>reset maintenance</code>	Деактивация для кластера режима технического обслуживания. Узлы кластера перейдут в режим восстановления.
reset maintenance on node ['node_name'] (п. 7.16.4)	<code>reset maintenance on node ['node_name']</code>	Деактивация для узла кластера режима технического обслуживания с использованием имени узла. Узел кластера перейдет в режим восстановления. Запускается на главном узле. Пример: <code>reset maintenance on node 'node1'</code>
reset maintenance on node [ip] [port] (п. 7.16.4)	<code>reset maintenance on node [ip] [port]</code>	Деактивация для узла кластера режима технического обслуживания с использованием IP-адреса и номера порта узла. Узел кластера перейдет в режим восстановления. Запускается на главном узле. Пример: <code>reset maintenance on node 10.116.102.55 12345</code>
set maintenance (п. 7.16.2)	<code>set maintenance</code>	Активация для кластера режима технического обслуживания.
set maintenance on node ['node_name'] (п. 7.16.1)	<code>set maintenance on node ['node_name']</code>	Активация для узла кластера режима технического обслуживания с использованием имени узла. Запускается на главном узле. Пример:

Наименование	Синтаксис команды	Описание
		set maintenance on node 'node1'
set maintenance on node [ip] [port] (п. 7.16.1)	set maintenance on node [ip] [port]	Активация для узла кластера режима технического обслуживания с использованием IP-адреса и номера порта узла. Запускается на главном узле. Пример: set maintenance on node 10.116.102.55 12345
<b>Команды управления узлами кластера</b>		
alter node ['old_node_name'] rename to ['new_node_name']	alter node ['old_node_name'] rename to ['new_node_name']	Изменение имени узла. Пример: alter node 'node2' rename to 'node_slave2'
alter node [ip] [port] rename to ['new_node_name']	alter node [ip] [port] rename to ['new_node_name']	Изменение имени узла по указанным IP-адресу и номеру сетевого порта. Пример: alter node 10.116.102.55 12345 rename to 'node_slave2'
<b>Команды управления пользователями</b>		
alter user ['user_name'] with password ['password'] (п.7.13)	alter user ['user_name'] with password ['password']	Изменение пароля для указанного пользователя. Пример:

Наименование	Синтаксис команды	Описание
		<pre>alter user 'user_cluster' with password 'PassWord'</pre>
alter user ['user_name'] rename to ['new name'] (п.7.13)	<pre>alter user ['user_name'] rename to ['new name']</pre>	<p>Изменение имени указанного пользователя. Пример:</p> <pre>alter user 'user_name' rename to 'user_cluster'</pre>
alter user ['user_name'] account lock (п.7.13)	<pre>alter user ['user_name'] account lock</pre>	<p>Блокировка указанного пользователя. Пример:</p> <pre>alter user 'user_cluster' account lock</pre>
alter user ['user_name'] account unlock (п.7.13)	<pre>alter user ['user_name'] account unlock</pre>	<p>Снятие блокировки с указанного пользователя. Пример:</p> <pre>alter user 'user_cluster' account unlock</pre>
create user ['user_name'] (п.7.13)	<pre>create user ['user_name']</pre>	<p>Создание пользователя с указанным именем. Ввод пароля пользователя вводится непосредственно в консоли. Пример:</p>

Наименование	Синтаксис команды	Описание
		create user 'user_cluster'
create user ['user_name'] with password ['password'] (п.7.13)	create user ['user_name'] with password ['password']	Создание пользователя с указанным именем и паролем. Пример:  create user 'user_cluster' with password 'paSSworD'
<b>Команды по управлению общим кластером (bundle)</b>		
bundle create ['bundle_name']	bundle create ['bundle_name']	Создание бандла кластера с допустимым именем. Пример:  bundle create 'bundle1'
bundle delete ['bundle_name']	bundle delete ['bundle_name']	Удаление ранее созданного бандла кластера. Пример:  bundle delete 'bundle1'
bundle attach cluster [ip] [port]	bundle attach cluster [ip] [port]	Добавление кластера в бандл по его IP-адресу и сетевому порту. Пример:  bundle attach cluster 10.116.102.55 12345
bundle attach cluster [ip] [port] ['interconnect_user']	bundle attach cluster [ip] [port] ['interconnect_user']	Добавление кластера в бандл по его IP-адресу, сетевому порту и имени технологического пользователя.

Наименование	Синтаксис команды	Описание
		<p>Пример:</p> <pre>bundle attach cluster 10.116.102.55 12345 'admin'</pre>
bundle detach cluster [ip] [port]	<pre>bundle detach cluster [ip] [port]</pre>	<p>Удаление кластера из бандла с указанным IP-адресом и сетевым портом кластера.</p> <p>Пример:</p> <pre>bundle detach cluster 10.116.102.55 12345</pre>
bundle detach cluster [cluster_name]	<pre>bundle detach cluster [cluster_name]</pre>	<p>Удаление кластера из бандла с указанным именем кластера.</p> <p>Пример:</p> <pre>bundle detach cluster clstr1</pre>
bundle status	<pre>bundle status</pre>	Вывод списка кластеров, входящих в бандл.
cancel		Отмена изменения параметров в рамках транзакционного редактирования конфигурации кластера. Запускается на главном узле.
<b>Команды управления дата-центрами кластера</b>		
datacenter create ['name'] (п. 7.7)	<pre>datacenter create 'name'</pre>	<p>Создание дата-центра</p> <pre>datacenter create 'dc1'</pre>



Наименование	Синтаксис команды	Описание
datacenter delete ['name'] (п. 7.8)	datacenter delete 'name'	Удаление дата-центра  datacenter delete 'dc1'
datacenter ['name'] attach node [ip] [port] (п. 7.9)	datacenter 'name' attach node [host_ip/host_name/hosts] [Jadog PORT number (port)]	Включение узла кластера в дата-центр  datacenter 'dc1' attach node 10.116.102.54 12345
datacenter ['name'] detach node [ip] [port] (п. 7.10)	datacenter 'name' detach node [host_ip/host_name/hosts] [Jadog PORT number (port)]	Выключение узла кластера из дата-центра  datacenter 'dc1' detach node 10.116.102.54 12345
datacenter promote (п. 7.11)	datacenter promote	Принудительное переключение на текущий центр обработки данных. Выполняется в асинхронном режиме.  datacenter promote
drop user ['user_name'] (п.7.13)	drop user ['user_name']	Удаление пользователя с указанным именем. Пример:  drop user 'user_cluster'
get last response	get last response	Отображение результата последней выполненной команды, исключая команды статуса. Запускается на главном узле.
get result id	get result 'id'	Отображение результата последней выполненной команды по указанному идентификатору, исключая команды статуса. Запускается на главном узле.

Наименование	Синтаксис команды	Описание
node show name	node show name	Отображение имени узла кластера с текущим IP-адресом. Запускается на главном узле, на резервных узлах.
node [ip] [port]	node [ip] [port]	Отображение имени узла кластера с указанным IP-адресом и номером порта. Пример: node 10.116.102.55 12345
node ['cluster_name'] [ip] [port]	node ['cluster_name'] [ip] [port]	Отображение имени узла указанного кластера с указанным IP-адресом и номером порта. Пример: node 'cluster1' 10.116.102.55 12345
node ls	node ls	Отображение списка имен узлов кластера. Запускается на главном узле, резервных узлах.
node ['cluster_name'] ls	node ['cluster_name'] ls	Отображение списка имен узлов указанного кластера. Запускается на главном узле, резервных узлах. Пример: node 'cluster1' ls
node set node_name = ['node_name']	node set node_name = ['node_name']	Переименование имени главного узла кластера. Запускается на главном узле. Пример: node set node_name = 'node_master'

Наименование	Синтаксис команды	Описание
node set node_name ['cluster_name'] [ip] [port] = ['node_name']	node set node_name ['cluster_name'] [ip] [port] = ['node_name']	Переименование имени главного узла указанного кластера. Запускается на главном узле. Пример:  node set node_name 'cluster1' 10.116.102.54 12345 = 'node_master'
reload jadog on cluster (п. 0)	reload jadog on cluster	Перезагрузка компонентов «jaDog» на всех узлах кластера. Запускается на главном узле.
reload jadog ['node_name'] (п. 0)	reload jadog 'node_name'	Перезагрузка узла кластера «jaDog». Запускается на главном узле, резервных узлах. Пример:  reload jadog 'node1'
restart dbs	restart dbs	Перезагрузка СУБД на главном узле. Запускается на главном узле.
restart dbs on cluster	restart dbs on cluster	Перезагрузка сервиса СУБД на всех узлах кластера. Запускается на главном узле.
restart dbs on node	restart dbs on node	Перезагрузка сервиса СУБД на узле. Запускается на главном узле, на резервных узлах.
restart dbs on node [ip] [port]	restart dbs on node [ip] [port]	Перезагрузка сервиса СУБД на узле с указанием IP-адреса и номера порта. Запускается на главном узле. Пример:  restart dbs on node 10.116.102.55 12345

Наименование	Синтаксис команды	Описание
restart dbs on node [ 'node_name' ]	restart dbs on node 'node_name'	Перезагрузка сервиса СУБД на узле с указанием имени узла в кластере. Запускается на главном узле. Пример: <pre>restart dbs on node 'node1'</pre>
reload jadog	reload jadog	Перезагрузка конфигурации кластера главном узле если он не входит в кластер.
save		Сохранение параметров в рамках транзакционного редактирования конфигурации кластера. Запускается на главном узле.
set parameter 'param_name' = 'param_value'	set parameter 'param_name' = 'param_value'	Установка параметра конфигурации кластера. Запускается на главном узле. В случае если изменение параметра требует перезагрузки узла или всего кластера выводится соответствующее уведомление. Пример: <pre>set parameter param_jadog:trusted_address = '10.116.102.56'</pre>
set parameters	set parameters	Вход в режим транзакционного редактирования параметров конфигурации кластера. Запускается на главном узле.
set async mode	set async mode	Включение асинхронного режима обработки команд транзакционного редактирования конфигурации кластера. Запускается на главном узле.

Наименование	Синтаксис команды	Описание
set sync mode	set sync mode	Включение синхронного режима обработки команд транзакционного редактирования конфигурации кластера. Запускается на главном узле.
show users	show users	Вывод списка пользователей кластера.
start dbs on cluster (п.п. 4.4)	start dbs on cluster	Запуск БД на всех узлах кластера. Выполняется на главном узле кластера.
start dbs on node (п.п. 4.4)	start dbs on node	Запуск БД на главном узле кластера. Выполняется на главном узле кластера.
start dbs on node [ip] [port] (п.п. 4.4)	start dbs on node [ip] [port]	Запуск БД на узле кластера по его IP-адресу и сетевому порту. Выполняется на главном узле кластера. Пример: <pre>start dbs on node 10.116.102.55 12345</pre>
start dbs on node ['node_name'] (п.п. 4.4)	start dbs on node ['node_name']	Запуск БД на узле кластера по его названию. Выполняется на главном узле кластера. Пример: <pre>start dbs on node node12</pre>
stop dbs on cluster (п.п. 4.4)	stop dbs on cluster	Остановка БД на всех узлах кластера. Выполняется на главном узле кластера.

Наименование	Синтаксис команды	Описание
stop dbs on node (п.п. 4.4)	stop dbs on node	Остановка БД на главном узле кластера. Выполняется на главном узле кластера.
stop dbs on node [ip] [port] (п.п. 4.4)	stop dbs on node [ip] [port]	Остановка БД на узле кластера по его IP-адресу и сетевому порту. Выполняется на главном узле кластера. Пример: <pre>stop dbs on node 10.116.102.55 12345</pre>
stop dbs on node ['node_name'] (п.п. 4.4)	stop dbs on node ['node_name']	Остановка БД на узле кластера по его названию. Выполняется на главном узле кластера. Пример: <pre>stop dbs on node node12</pre>
switchover [ip] [port] (п. 7.15)	switchover [host_ip/host_name/hosts] [Jadog PORT number (port)]	Переключение роли главного узла на узел с указанным IP-адресом. Выполняется в асинхронном режиме. Пример: <pre>switchover 10.116.102.55 12345</pre>
switchover ['node_name']	switchover ['node_name']	Переключение роли главного узла на узел с указанным именем. Выполняется в асинхронном режиме. Пример:

Наименование	Синтаксис команды	Описание
		switchover 'node2'
reload jalog on cluster (п.п. 4.3)	reload jalog on cluster	Перезагрузка компонентов «jaDog» на всех узлах кластера. Выполняется на главном узле.
reload jalog 'node_name' (п.п. 4.3.1)	reload jalog 'node_name'	Перезагрузка узла кластера «jaDog». Выполняется на главном узле, резервных узлах.

## 6. ОС GNU/LINUX. НАСТРОЙКА ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА СУБД «ЯТОВА» В РУЧНОМ РЕЖИМЕ

В рассматриваемом примере установки и конфигурировании отказоустойчивого кластера «jaDog» на ОС Ubuntu 22.04 в ручном режиме с помощью утилиты «jadog\_ctl» используются параметры сети кластера, приведенные в таблице 6.1.

Таблица 6.1 – Конфигурация сети кластера

№	Имя сервера	Подсеть кластера	Маска подсети	Public IP	Роль	Имя кластера
		10.116.102.0/24				
		IP-адрес				
1	Node1	10.116.102.54/24	255.255.255.0	10.116.102.81/24	Master	TEST
2	Node2	10.116.102.55/24	255.255.255.0	10.116.102.81/24	Slave	
3	Shared-node	10.116.103.56/24	255.255.255.0		File-server	

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

### 6.1. Установка пакетов СУБД на главном узле (Master)

Установка резервного узла из локального репозитория для Ubuntu описана в п.п. 6.6.2.

Установка компонента проводится одним из доступных способов:

— вручную из локального репозитория, как описано в Приложении 2, настоящего документа;

— инсталлятором СУБД, как описано в пункте «Инсталлятор СУБД «Jatoba» для ОС семейства GNU/Linux» в Руководстве администратора 643.72410666.00067-07 97 01.

После установки СУБД инсталлятором требуется установить пакет компонента командой:

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

Дальнейших проверочных действий не требуется.

После ручной установки из локального репозитория на главном узле, необходимо проверить, что в ходе установки СУБД «Jatoba» создана директория с данными «data».



## 6.2. Настройки аутентификации на главном узле

На узле с ролью Master установить требуемый метод аутентификации для пользователя СУБД «jalog\_user» от имени и справками которого будет выполняться репликация на узлах кластера.

Поддерживаются следующие методы аутентификации:

- MD5;
- SSL;
- LDAP(s);
- RADIUS;
- PAM;
- SSPI;
- GSSAPI.

В качестве примера устанавливается метод аутентификации MD5 (password) в конфигурационном файле pg\_hba.conf и добавляются строки для подключения пользователя СУБД «jalog\_user» (см. рис. 6.1).

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all jalog_user 127.0.0.1/32 md5
host all jalog_user 10.116.102.0/24 md5
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all md5
host replication jalog_user 10.116.102.0/24 md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
```

Рисунок 6.1 – Первоначальные настройки аутентификации

Остальные параметры аутентификации пользователей СУБД устанавливаются в соответствии с требованиями информационной системы и в данном руководстве не рассматриваются.



Дополнительная настройка конфигурационного файла postgresql.conf для настройки архивирования WAL не требуется.

### 6.3. Установка пароля системного пользователя ОС «postgres» на главном узле

Установить пароль для системного пользователя ОС «postgres»:

```
sudo passwd postgres
```

```
root@node1: /home/admin1
admin1@node1:~$ sudo su
[sudo] password for admin1:
root@node1:/home/admin1# sudo passwd postgres
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: password updated successfully
root@node1:/home/admin1#
```

Рисунок 6.2 – Установка пароля для пользователя ОС

После чего перезапустить службу «jatoba-6» и проверить ее статус:

```
# systemctl restart jatoba-6
# systemctl status jatoba-6
```

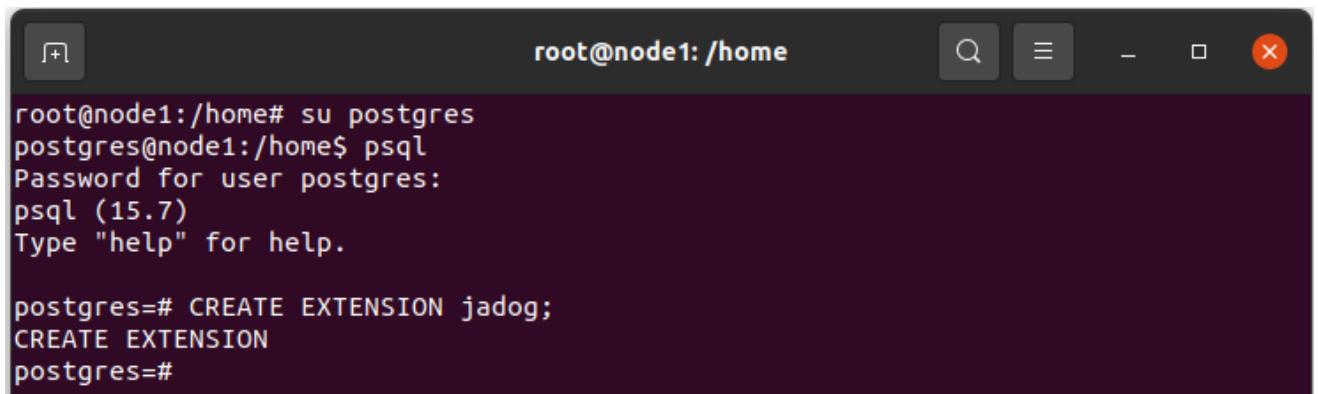
### 6.4. Установка расширения «jadow» на главном узле

Расширение СУБД «jadow» служит, для:

- проведения репликации между узлами кластера;
- установки дополнительных функций, в том числе функции «add\_jadow\_user».

Расширение «jaDog» устанавливается при помощи SQL-команды:

```
CREATE EXTENSION jadow;
```



```
root@node1: /home
root@node1:/home# su postgres
postgres@node1:/home$ psql
Password for user postgres:
psql (15.7)
Type "help" for help.

postgres=# CREATE EXTENSION jadow;
CREATE EXTENSION
postgres=#
```

Рисунок 6.3 – Установка расширения

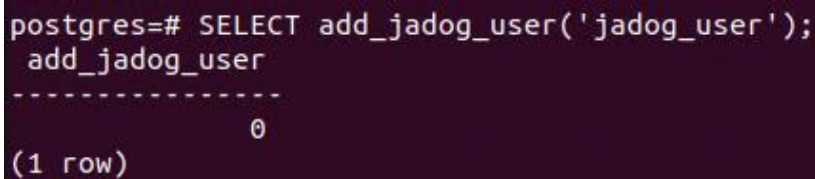
### 6.5. Вызов функции «add\_jadow\_user» для создания пользователя «jadow\_user» на главном узле

С целью соблюдения принципа «Назначения минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование информационной системы» устанавливается специальная функция «add\_jadow\_user». Данная функция СУБД используется для создания специальной роли «jadow\_user» с атрибутами:

- login;
- replication.

Роль создается при помощи SQL-команды:

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



```
postgres=# SELECT add_jadow_user('jadow_user');
add_jadow_user
-----
0
(1 row)
```

Рисунок 6.4 – Создание пользователя «jadow\_user»

Пароль для роли создается при помощи SQL-команды:

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

**Например**

```
ALTER ROLE jadow_user with password 'password';
```

Таким образом создаваемому пользователю автоматически будут назначены атрибуты и привилегии.

Убедиться в успешном создании пользователя можно через SQL-команду просмотра списка пользователей СУБД:

```
\du
```

```

root@node1: /home/admin1
File Edit View Search Terminal Help
postgres-# \du
                                List of roles
Role name |                               Attributes                               | Member of
-----+-----+-----
jadog_user | Replication                               | {pg_read_all_stats}
postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
postgres-#
  
```

Рисунок 6.5 – Просмотр списка пользователей СУБД

## 6.6. Установка пакетов СУБД на резервном узле (Slave)

Для резервного узла должен использоваться отдельный сервер с установленной СУБД «Jatoba» с пустым каталогом базы данных.

Установка компонента «jaDog» на резервном узле проводится одним из доступных способов:

— вручную из локального репозитория, как описано в п. 6.6.2, настоящего документа;

— инсталлятором СУБД, как описано в п. «Инсталлятор СУБД «Jatoba» для ОС семейства GNU/Linux» в Руководстве администратора 643.72410666.00067-07 97 01.

### 6.6.1. Установка СУБД на резервном узле инсталлятором

После установки СУБД инсталлятором требуется установить пакет компонента командой:

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

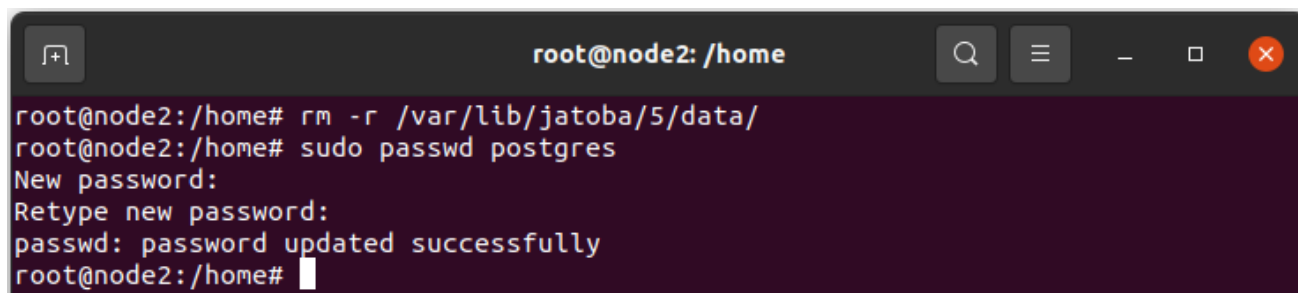
Удалить каталог «data» командой:

```
rm -r /var/lib/jatoba/6/data/
```

Установить пароль для системного пользователя ОС «postgres»:

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

```
sudo passwd postgres
```



```
root@node2: /home
root@node2:/home# rm -r /var/lib/jatoba/5/data/
root@node2:/home# sudo passwd postgres
New password:
Retype new password:
passwd: password updated successfully
root@node2:/home#
```

Рисунок 6.6 – Команды удаления каталога БД и задание пароля пользователю ОС

Дальнейших проверочных действий не требуется и можно переходить к действиям, описанным в п. п. 6.8.1, 6.8.2.

### 6.6.2. Установка СУБД на резервном узле вручную из локального репозитория

Установка СУБД описаны в документе 643.72410666.00067-07 97 01 «Защищенная система управления базами данных «Jatoba». Руководство по установке» и в Приложении 2 настоящего документа.

По примеру установки из локального репозитория выполните нижеописанные действия:

- 1) В терминале войти в режим суперпользователя, выполнив команду:

```
sudo su
```

- 2) Если команды sudo не существует – установить:

```
su -l
apt-get install sudo -y
```

- 3) Выполнить обновление системы:

```
sudo apt update && sudo apt upgrade -y
sudo apt -s dist-upgrade
sudo apt dist-upgrade
```

- 4) Создать папку localrepo в корневом каталоге:

```
mkdir /localrepo
```

- 5) В созданную папку скопировать:

- каталог <pool>;
- каталог <dist>;
- файл <DEB-GPG-KEY-Jatoba>.

6) Установить открытый ключ репозитория:

```
apt-key add /localrepo/DEB-GPG-KEY-Jatoba
```

7) Добавить описание локального репозитория в систему:

```
nano /etc/apt/sources.list.d/jatoba-6.list
```

8) Вставить в файл следующее содержимое и сохранить:

```
deb file:///localrepo stable non-free
```

9) Проиндексировать обновленное состояние репозитория:

```
apt-get update
```

10) Установить СУБД Jatoba при помощи команды:

```
apt-get install jatoba6-client jatoba6-contrib jatoba6-libs  
jatoba6-jadog jatoba6-server
```

11) Установить пароль для системного пользователя ОС «postgres»:

```
sudo passwd postgres
```

### 6.6.3. Действия не требующие выполнения на резервном узле при установке СУБД из локального репозитория



**Не выполняются нижеописанные действия:**

1) инициализация каталога данных СУБД при помощи команды;

Каталог СУБД должен быть пуст. При синхронизации узлов кластера с главного узла данные будут перенесены.

2) проверка лицензионных параметров в конфигурационном файле «postgresql.conf»;

Конфигурационный файл «postgres.conf» будет скопирован при синхронизации.

3) добавление сервиса jatoeba-6 в список автозапуска и проверка статуса сервиса;

Запуск СУБД на резервном узле выполнит компонент «jaDog» и в автоматическом режиме установит службы.

4) установка пароля для пользователя СУБД «postgres»;

Установить пароль для пользователя СУБД «postgres» невозможно, т.к. отсутствуют конфигурационные файлы.

### 6.7. Настройка компонента «jaDog» на главном узле через сервис «jadog»

Для настройки компонента «jaDog» на главном узле кластера необходимо выполнить следующие действия:

1) Открыть терминал ОС Ubuntu.

2) Авторизоваться от имени привилегированного пользователя ОС, используя команду:

```
sudo su
```

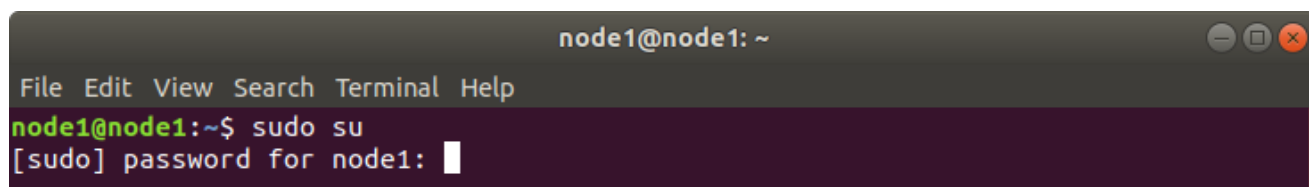


Рисунок 6.7 – Авторизация от имени привилегированного пользователя ОС

3) Перейти в директорию СУБД, используя команду:

```
cd /usr/jatoba-6/bin
```

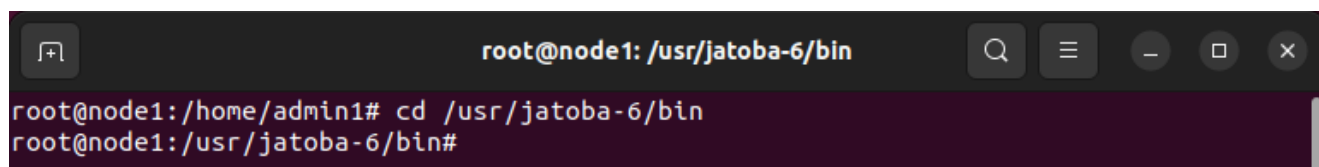


Рисунок 6.8 – Команда перехода в директорию

4) Запустить исполняемый файл сервиса «jadog» с параметром «setip» для перехода в режим конфигурирования компонента командой:

```
./jadog setup
```

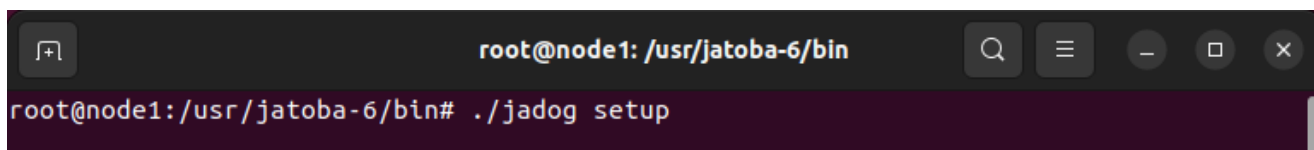


Рисунок 6.9 – Запуск исполняемого файла сервиса «jadog»

После чего откроется главное меню конфигурирования компонента.

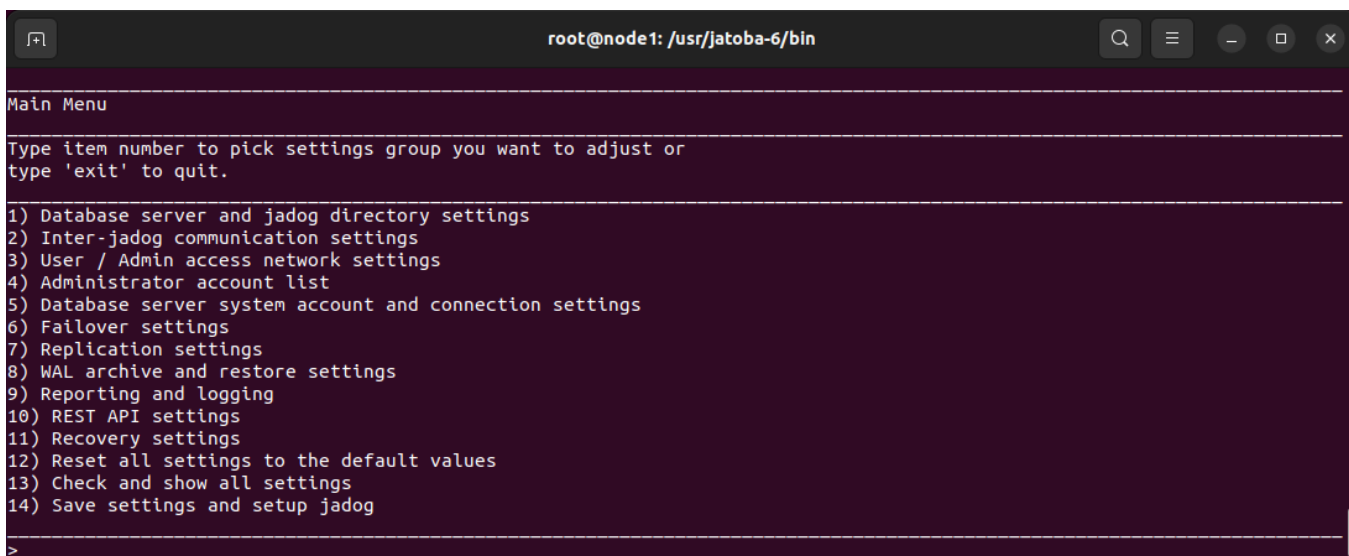


Рисунок 6.10 – Главное меню компонента

Главное меню содержит пункты:

- [\*1\) Database Server and jaDog Directory Settings\*](#) – для перехода в подменю настройки директорий размещения компонента «jaDog» и СУБД «Jatoba».
- [\*2\) Inter-jadog communication settings\*](#) – для перехода в подменю настройки параметров сетевых соединений между службами jaDog на узлах кластера СУБД.
- [\*3\) User / Admin access network settings\*](#) – для перехода в подменю настройки сетевой конфигурации доступа пользователей и администраторов к СУБД и «jaDog».
- [\*4\) Administrator account list\*](#) – для перехода в подменю настройки административных учетных записей jaDog.
- [\*5\) Database server system account and connection settings\*](#) – для перехода в подменю настройки административных учетных записей СУБД.



- [6\) Failover setting](#) – для перехода в меню настройки ручного или автоматического режима отказоустойчивости (Failover).
- [7\) Replication setting](#) – для перехода в меню настройки синхронной и асинхронной репликации.
- [8\) WAL archive and restore settings](#) – для перехода в меню конфигурирование архивирования/восстановления WAL в хранилище/ из хранилища
- [9\) Reporting and logging](#) – для перехода в меню настройки логирования работы кластера.
- [10\) Reset all settings to the default values](#) – для выполнения команды сброса введенных ранее значений параметров на значения по умолчанию.
- [11\) Check and show all settings](#) – для вывода и проверки значений всех параметров.
- [12\) Save settings and setup jadog](#) – для сохранения введенных настроек, инициализации сервисов компонента «jaDog» и завершения работы jadog setup.

Далее необходимо оставить заданные по умолчанию или, последовательно выбирая подпункты меню, задать новые параметры конфигурирования компонента «jaDog» с учетом следующих правил:

- для перехода в подменю (раздел jadog setup нижнего уровня) – ввести номер подменю и нажать ENTER (далее – выбрать пункт меню);
- для ввода нового значения параметра – выбрать подпункт меню текущего раздела jadog setup, ввести новое значение параметра, нажать ENTER;
- для отмены всех введенных в разделе меню новых значений параметров и возврата к значениям по умолчанию – выбрать номер подпункта текущего раздела jadog setup «Reset all settings to the default values» и нажать ENTER;
- для перехода на предыдущий уровень меню – ввести «#» и нажать ENTER;
- для выхода из режима конфигурирования – ввести «exit» и нажать ENTER.

### 6.7.1. Пункт меню 1 «Database server and jadog directory settings»

Проконтролировать заданные по умолчанию или указать новые значения для всех директорий размещения компонента «jaDog» и СУБД «Jatoba» (рисунок 6.11):

- в главном меню выбрать пункт 1 «Database server and jadog directory settings»;
- указать требуемые директории, если они отличаются от установленных по умолчанию.

```
Database server and jadog directory settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
1) Database server binaries (param_postgres:db_bin_path)          [/usr/jatoba-6/bin]
2) Database server data (param_postgres:db_data_path)             [/var/lib/jatoba/6/data]
3) Jadog configs (param_path:config_path)                         [/usr/jatoba-6/etc/jadog]
4) Jadog module files (param_path:module_path)                   [/usr/jatoba-6/share/jadog/scripts]
5) Jadog state (param_path:state_path)                           [/usr/jatoba-6/etc/jadog]
-----
6) Reset all settings to the default values
-----
>
```

Рисунок 6.11 – Пункт меню «Database server and jadog directory settings»

#### 6.7.1.1 Подменю 1) Database server binaries (param\_postgres: db\_bin\_path)

Database server binaries (db\_bin\_path) – полный путь к системным утилитам СУБД.

По умолчанию установлен параметр:

- в GNU Linux – [/usr/jatoba-6/bin];
- в ОС Window - [C:\Program Files\GIS\Jatoba\5\bin].

#### 6.7.1.2 Подменю 2) Database server data (param\_postgres: db\_data\_path)

Database server data (db\_data\_path) – полный путь к директории данных узла кластера.

По умолчанию установлен параметр:

- в GNU Linux – [/usr/jatoba-6/data];
- в ОС Window - [C:\Program Files\GIS\Jatoba\6\data].

#### 6.7.1.3 Подменю 3) Jadog configs (param\_path: config\_path)

Jadog configs (config\_path) – полный путь размещения конфигурационного файла. Имя конфигурационного файла всегда должно быть «jadog.yml».

По умолчанию установлен параметр:

- в GNU Linux – [/usr/jatoba-6/etc/jadog];

– в ОС Window - [C:\Program Files\GIS\Jatoba\6\etc\jadog].

#### **6.7.1.4 Подменю 4) Jadow module files (param\_path: module\_path)**

Jadow module files (module\_path) – полный путь к дополнительным скриптам, необходимым для проведения системных действий.

По умолчанию установлен параметр – [/usr/jatoba-6/share/jadow/scripts]

#### **6.7.1.5 Подменю 5) Jadow state (param\_path: state\_cfg\_path)**

Jadow state (state\_cfg\_path) – полный путь до отдельного файла jadow\_state.yml с состоянием и конфигурацией кластера.

По умолчанию установлен параметр:

– в GNU Linux – [/usr/jatoba-6/etc/jadow];

– в ОС Window - [C:\Program Files\GIS\Jatoba\6\etc\jadog].

Структура файла jadow\_state.yml приведена в Приложении 1 настоящего документа.

#### **Подменю 6) Reset all settings to the default values**

Reset all settings to the default values – перезагрузка всех установленных параметров в данном меню.

### 6.7.2. Пункт меню 2 «Inter-jadog communication setting»

Проконтролировать заданные по умолчанию и указать новые значения адреса и порта привязки настраиваемого компонента «jaDog» для взаимодействия с процессами компонента «jaDog» другого узла, для чего:

- в главном меню выбрать пункт 2 «Inter-jadog communication setting»;
- установить требуемые параметры;
- для отмены внесенных изменений и возврата к значениям по умолчанию – выбрать пункт подменю 12 «Reset all settings to the default values»;
- для сохранения изменений – вернуться в главное меню.

```
Inter-jadog communication settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).
IP and PORT are appropriate parameters that specify which network will be used to communicate
with another jadog-nodes inside the RDBMS cluster. Both parameters are mandatory.
1) Jadog service name (param_jadog:service_name) [jadog]
2) Jadog IP address (param_jadog:ip) []
3) Jadog PORT number (param_jadog:port) [12345]
4) Jadog searching protocol port (param_jadog:jadog_search_port) [12346]
5) SSL on (param_ssl:ssl) [false]
6) Jadog interconnection user (param_jadog:interconnect_user) [admin]
7) Reset all settings to the default values
```

Рисунок 6.12 – Подменю «Inter-jadog communication setting»

#### 6.7.2.1 Подменю 1) Jadog service name (param\_jadog: service\_name)

Jadog service name (service\_name) – имя службы компонента «jaDog».

По умолчанию установлен параметр – [jadog].

#### 6.7.2.2 Подменю 2) Jadog IP address (param\_jadog: ip)

Jadog IP address (ip) – IP–адрес узла кластера. Указывается установленный IP–адрес сетевой карты сервера СУБД.

Значения по умолчанию не имеет.

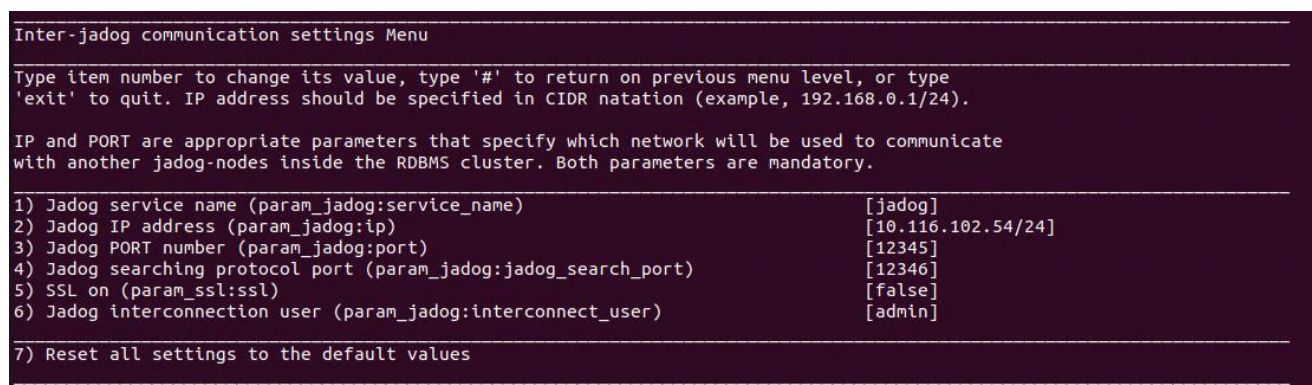


Рисунок 6.13 – Подменю «Jadog IP address (ip)»

В рассматриваемом примере устанавливается значение – 10.116.102.54/24.

### 6.7.2.3 Подменю 3) Jadog PORT number (param\_jadog: port)

Jadog PORT number (port) – порт обмена служебными данными процессов jaDog.

По умолчанию установлен параметр – [12345].

### 6.7.2.4 Подменю 4) Jadog searching protocol port (param\_jadog: jadog\_search\_port)

Jadog\_search\_port (jadog\_search\_port) – порт для поиска «свободных» узлов в подсети.

Параметр обязательный.

По умолчанию установлен параметр – [12346].

### 6.7.2.5 Подменю 5) SSL on (param\_ssl: ssl)

SSL on (ssl) – активация/деактивация режима соединения по протоколу SSL между узлами кластера.

По умолчанию установлен параметр – [false], т.е. отключен.

### 6.7.2.6 Подменю 6) Jadog interconnection user (param\_jadog: interconnect\_user)

Jadog interconnection user (interconnect\_user) – имя технологической учетной записи для взаимодействия с другими jaDog сервисами.

По умолчанию установлен параметр – [admin].

Создание технологической учетной записи приведено в п.п. 7.13.

### Подменю 7) Reset all settings to the default values

Reset all settings to the default values – перезагрузка всех установленных параметров в данном меню.



### 6.7.3. Пункт меню 3 «User / Admin access network setting»

Проконтролировать заданные по умолчанию и указать новые значения настроек сетевой конфигурации доступа пользователей и администраторов к СУБД и «jaDog»:

- в главном меню выбрать пункт 3 «User / Admin access network setting»;

```
User / Admin access network settings Menu

Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).

Public Ip address is the address that will be migrated between cluster nodes during failover
for reliable user access. Trusted address is the address to check user network accessibility.
Network interface is the network device name to which public address will be binded.

1) Public address (param_jadog:public_address) [ ]
2) Public address control interval (param_jadog:public_address_control_interval)[5000]
3) Public address control attempts (param_jadog:public_address_control_attempts)[3]
4) DB monitor timeout (param_postgres:db_check_interval) [5000]
5) Trusted IP address (param_jadog:trusted_address) [ ]
6) RDBMS trusted ip (param_jadog:rdbms_trusted_ip) [ ]
7) RDBMS trusted port (param_jadog:rdbms_trusted_port) [ ]
8) Network interface name (param_jadog:network_interface) [ ]
9) Jadog TCP port (param_jadog:user_interface_port) [54321]

10) Reset all settings to the default values
```

Рисунок 6.14 – Вид меню «User / Admin access network setting»

- указать новые значения параметров.

#### 6.7.3.1 Подменю 1) Public IP address (param\_jadog: public\_address)

Public IP address (public\_address) – IP-адрес/маска в CIDR-нотации, через который будет происходить доступ пользователей к СУБД.

```
User / Admin access network settings Menu

Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).

Public Ip address is the address that will be migrated between cluster nodes during failover
for reliable user access. Trusted address is the address to check user network accessibility.
Network interface is the network device name to which public address will be binded.

1) Public address (param_jadog:public_address) [10.116.102.81/24]
2) Public address control interval (param_jadog:public_address_control_interval)[5000]
3) Public address control attempts (param_jadog:public_address_control_attempts)[3]
4) DB monitor timeout (param_postgres:db_check_interval) [5000]
5) Trusted IP address (param_jadog:trusted_address) [ ]
6) RDBMS trusted ip (param_jadog:rdbms_trusted_ip) [ ]
7) RDBMS trusted port (param_jadog:rdbms_trusted_port) [ ]
8) Network interface name (param_jadog:network_interface) [ ]
9) Jadog TCP port (param_jadog:user_interface_port) [54321]

10) Reset all settings to the default values
```

Рисунок 6.15 – Подменю 1) Public IP address (public\_address)

Значение по умолчанию не установлено – [ ]

В рассматриваемом примере устанавливается значение – 10.116.102.81/24.

### **6.7.3.2 Подменю 2) Public address control interval (param\_jadog: public\_address\_control\_interval)**

Public address control interval (public\_address\_control\_interval) - контрольный интервал проверки доступности публичного IP-адреса.

Значение по умолчанию - [5000] в «ms».

Функциональная возможность контроля публичного IP-адреса позволяет контролировать:

- дублирование или пропадание «Public IP» на узлах кластера;

При необходимости дубли исключаются, а пропавший «Public IP» переназначается.

- слоты репликации и выводить узел с ролью «Slave» режим «Fensing».

Смена роли узла из «Slave» в роль «Fensing» происходит в случае непредвиденных ошибок репликации в активном режиме или при неактуальности каталога данных в режиме восстановления.

Такое состояние узла кластера требует перезагрузки для нормального восстановления кластера.

### **6.7.3.3 Подменю 3) Public address control attempts (param\_jadog: public\_address\_control\_attempts)**

Public address control attempts (public\_address\_control\_attempts) – количество контрольных вызовов проверок публичного IP-адреса.

Значение по умолчанию - [3].

### **6.7.3.4 Подменю 4) DB monitor timeout (param\_postgres: db\_check\_interval)**

DB monitor timeout (param\_postgres:db\_check\_interval) – период проверки доступности СУБД

Значение по умолчанию - [5000].

### **6.7.3.5 Подменю 5) Trusted IP address (trusted\_address)**

Trusted IP address (trusted\_address) – доверенный адрес служит дополнительным индикатором принятия решения при обработке отказа «failover».

Значение по умолчанию не установлено – [].

Параметр не обязательный.

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

#### **6.7.3.6 Подменю 6) RDBMS trusted ip (param\_jadog: rdbms\_trusted\_ip)**

RDBMS trusted ip (rdbms\_trusted\_ip) (relational database management system) – доверенный IP адрес системы управления реляционными базами данных (используется для компонента «ja\_Hipe\_Cluster»).

Значение по умолчанию не установлено – [].

#### **6.7.3.7 Подменю 7) RDBMS trusted port (param\_jadog: rdbms\_trusted\_port)**

RDBMS trusted port (rdbms\_trusted\_port) – доверенный порт системы управления реляционными базами данных (используется для компонента «ja\_Hipe\_Cluster»).

Значение по умолчанию не установлено – [].

Параметры «RDBMS trusted ip» и «RDBMS trusted port» используются для кластерных инженерных решений с компонентом «ja\_Hipe\_Cluster» для предотвращения синхронизации метаданных. В параметрах указывается IP-адрес и порт сервера с ролью «Координатор».

Формируется запрос проверки статуса сервера. При остановке/недоступности сервера с ролью «Координатор» блокируются операции повышения роли сервера «Slave» до момента пока координатор не будет стабилизирован.

#### **6.7.3.8 Подменю 8) Network interface name (param\_jadog: network\_interface)**

Network interface name (network\_interface) – имя сетевого интерфейса на данном узле кластера, на который осуществляется привязка общего адреса.

Значение по умолчанию не установлено – [].

В рассматриваемом примере устанавливается значение – ens33.

Имя сетевого интерфейса в ОС GNU/Linux возможно узнать в терминале ОС, используя команду:

```
ifconfig
```



```

admin1@node1: ~
admin1@node1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.116.102.54 netmask 255.255.255.0 broadcast 10.116.102.255
    inet6 fe80::20c:29ff:fe23:c3f0 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:23:c3:f0 txqueuelen 1000 (Ethernet)
    RX packets 168182 bytes 224959865 (224.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24032 bytes 2998618 (2.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1683 bytes 185911 (185.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1683 bytes 185911 (185.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

admin1@node1:~$

```

Рисунок 6.16 – Выполнение команды «ifconfig»

Если пакет не установлен, то ОС предложит его установить.

По умолчанию в ОС Windows используется имя «Ethernet».

#### 6.7.3.9 Подменю 9) Jadog TCP port (param\_jadog: user\_interface\_port)

Jadog TCP port (user\_interface\_port) – порт доступа административных подключений для управления кластером через утилиту «jadog\_ctl».

Значение по умолчанию – [54321].

```

User / Admin access network settings Menu

Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).

Public Ip address is the address that will be migrated between cluster nodes during failover
for reliable user access. Trusted address is the address to check user network accessibility.
Network interface is the network device name to which public address will be binded.

1) Public address (param_jadog:public_address) [10.116.102.81/24]
2) Public address control interval (param_jadog:public_address_control_interval)[5000]
3) Public address control attempts (param_jadog:public_address_control_attempts)[3]
4) DB monitor timeout (param_postgres:db_check_interval) [5000]
5) Trusted IP address (param_jadog:trusted_address) [ ]
6) RDBMS trusted ip (param_jadog:rdbms_trusted_ip) [ ]
7) RDBMS trusted port (param_jadog:rdbms_trusted_port) [ ]
8) Network interface name (param_jadog:network_interface) [ens18]
9) Jadog TCP port (param_jadog:user_interface_port) [54321]

10) Reset all settings to the default values

```

Рисунок 6.17 – Подменю 4) Jadog TCP port (user\_interface\_port)

#### 6.7.4. Пункт меню 4 «Administrator account list»

В меню 4 «Administrator account list» указываются учетные записи администратора либо администраторов компонента «jaDog». Предусмотренные УЗ в компоненте отсутствуют.

Для создания списка УЗ необходимо:

- в главном меню выбрать пункт 4 «Administrator account list»;
- для создания одной УЗ выбрать пункт 1 «Add new account to the list», указать логин и пароль УЗ, повторить ввод пароля, нажать ENTER;
- повторить действия для создания всех требуемых УЗ;
- Добавление новой УЗ происходит в конец списка. Нумеруются УЗ в порядке их добавления, в этом же порядке записываются в конфигурационный файл.
- для удаления одной УЗ выбрать пункт 2 «Delete account from the list», ввести номер УЗ, нажать ENTER;
- для удаления всех УЗ выбрать пункт 3 «Clear all accounts» подменю;
- для сохранения изменений – вернуться в главное меню.

```
Administrator account list Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Accounts is empty
1) Add new account to the list
2) Delete account from the list
3) Clear all accounts
```

Рисунок 6.18 – Пункт меню 4 «Administrator account list»

Предусмотренные учетные записи в компоненте отсутствуют. В данном разделе требуется ввести учетную запись административной учетной записи компонента «jaDog», от имени и с правами которой будет осуществляется подключение к узлам кластера и его администрирование.

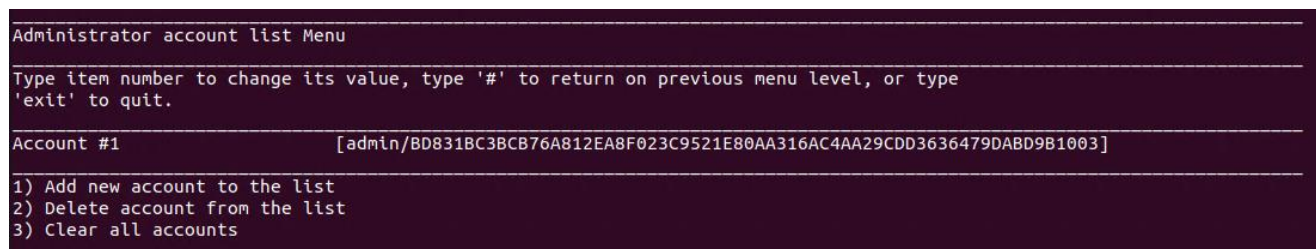


Рисунок 6.19 – Список УЗ кластера

Учетная запись должна быть идентичной на всех узлах кластера.

Созданные пользователи с хешированными паролями в формате sha-256 будут храниться в файле:

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

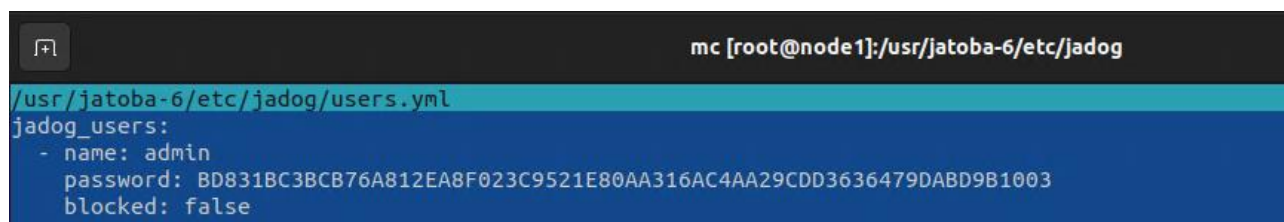


Рисунок 6.20 – Файл пользователей

Учетные данные используются при настройке доступа к экземплярам компонента «JaDog» через конфигурационный файл jadog\_hba.cfg (см. п. 4.2.1).

#### 6.7.5. Пункт меню 5 «Database server system account and connection settings»

Проконтролировать заданные по умолчанию или указать новые значения административных учетных записей СУБД «Jatoba»:

- в главном меню выбрать пункт 5 «Database server system account and connection settings»;
- указать новые значения параметров, если они отличаются от установленных по умолчанию, выбирая пункты 1 – 9 подменю (правила ввода новых значений см. выше), либо оставить заданные по умолчанию;
- для отмены внесенных изменений и возврата к значениям по умолчанию – выбрать пункт 12 подменю «Reset all settings to the default values»;
- для сохранения изменений – вернуться в главное меню.

### 6.7.5.1 Подменю 1) Database host (param\_postgres: db\_host)

Database host (db\_host) – адрес подконтрольного СУБД сервера. В текущей версии компонента «jaDog» принято, что подконтрольные процессы СУБД находятся на локальном узле.

```
Database server system account and connection settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Database auth method can take the following values: password, ssl

1) Database host (db_connection_settings:host) [127.0.0.1]
2) Database port (db_connection_settings:port) [5432]
3) Database service name (param_postgres: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_settings:db_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]
13) Jadog database user password (db_connection_settings:user_pass) HIDDEN VALUE
14) Database server OS user (param_system:system_user) [postgres]
15) Replication node name (param_replication:replication_slot_name) []

16) Reset all settings to the default values
>
```

Рисунок 6.21 – Подменю 1) Database host (db\_host)

### 6.7.5.2 Подменю 2) Database port (param\_postgres:db\_port)

Database port (param\_postgres:db\_port) – порт подключения СУБД «Jatoba». По умолчанию используется значение – 5432.

### 6.7.5.3 Подменю 3) Database service name (param\_postgres: db\_service\_name)

Database service name (param\_postgres: db\_service\_name) – имя сервиса (демона) СУБД в ОС.

По умолчанию используется значение:

- В GNU Linux – [jatoba-6];
- В ОС Windows - [JatobaServer].

### 6.7.5.4 Подменю 4) Database name (param\_postgres: db\_name)

Database name (db\_name) – имя базы данных.

По умолчанию используется имя БД – [postgres].

### 6.7.5.5 Подменю 5) Jadog password file (param\_postgres: db\_passfile)

Jadog password file (db\_passfile) – путь к файлу хранения пароля.

В указанной директории по окончании конфигурирования компонента будет создан файл «db\_passfile», в котором будет храниться аутентификационная информация служебной УЗ (роли) СУБД «jadowg\_user».

В дальнейшем от имени и с правами роли «jadowg\_user» будет проводиться репликация данных между узлами кластера.

Путь к файлу может быть назначен произвольно.



Для хранения файла «db\_passfile», рекомендуется использовать защищенную директорию

В приводимом примере устанавливается значение для:

– GNU Linux:

```
/usr/jatoba-6/bin/db_passfile
```

– ОС Windows:

```
C:\Program Files\GIS\Jatoba\6\bin\
```

При установке СУБД в ОС Windows полный доступ к каталогу «bin» назначается:

- группе пользователей «Администраторы»;
- создателю-владельцу;
- локальному пользователю ОС «postgres».

```
Database server system account and connection settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Database auth method can take the following values: password, ssl
-----
1) Database host (db_connection_settings:host) [127.0.0.1]
2) Database port (db_connection_settings:port) [5432]
3) Database service name (param_postgres:db_service_name) [jatoba-6]
4) Database name (db_connection_settings:database) [postgres]
5) Jadowg password file (db_connection_settings:passfile) [/usr/jatoba-6/bin/db_passfile]
6) Database auth method (db_connection_settings:db_auth_method) [password]
7) Jadowg to database CA file (db_connection_settings:ssl_ca_file) []
8) Jadowg to database CRL file (db_connection_settings:ssl_crl_file) []
9) Jadowg to database cert file (db_connection_settings:ssl_cert_file) []
10) Jadowg to database key file (db_connection_settings:ssl_key_file) []
11) Jadowg to database SSL mode (db_connection_settings:ssl_mode) [verify-full]
12) Jadowg database user name (db_connection_settings:user) [jadowg_user]
13) Jadowg database user password (db_connection_settings:user_pass) HIDDEN VALUE
14) Database server OS user (param_system:system_user) [postgres]
15) Replication node name (param_replication:replication_slot_name) []
-----
16) Reset all settings to the default values
>
```

Рисунок 6.22 – Подменю 5) Jadowg password file (db\_passfile)

#### **6.7.5.6 Подменю 6) Database auth method (param\_postgres: db\_auth\_method)**

Database auth method (db\_auth\_method) - тип аутентификации утилиты «jadog\_ctl» к СУБД.

Значение по умолчанию – [password]

Возможные значения password и ssl. Может быть установлено только одно значение.

Параметр аутентификации «password» применим, для:

- RADIUS;
- PAM.

Параметр аутентификации «ssl» применим, для:

- SSL;
- LDAP(s);
- SSPI;
- GSSAPI.

#### **6.7.5.7 Подменю 7) Jadog to database CA file (param\_jadog\_to\_postgres: jadog\_to\_db\_ca\_file)**

Jadog to database CA file (jadog\_to\_db\_ca\_file) - путь до сертификата центра сертификации, который Jadog использует для подключения к СУБД.

Значение по умолчанию – []

#### **6.7.5.8 Подменю 8) Jadog to database CRL file (param\_jadog\_to\_postgres: jadog\_to\_db\_crl\_file)**

Jadog to database CRL file (jadog\_to\_db\_crl\_file) - путь до CRL файла, который Jadog использует для подключения к СУБД.

Значение по умолчанию – []

#### **6.7.5.9 Подменю 9) Jadog to database cert file (param\_jadog\_to\_postgres: jadog\_to\_db\_cert\_file)**

Jadog to database cert file (jadog\_to\_db\_cert\_file) - путь до файла сертификата, который Jadog использует для подключения к СУБД.

Значение по умолчанию – []

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

#### 6.7.5.10 Подменю 10) Jadog to database key file (param\_jadog\_to\_postgres: jadog\_to\_db\_key\_file)

Jadog to database key file (jadog\_to\_db\_key\_file) - путь до файла с ключом, который Jadog использует для подключения к СУБД.

Значение по умолчанию – []

#### 6.7.5.11 Подменю 11) Jadog to database SSL mode (param\_jadog\_to\_postgres: jadog\_to\_db\_ssl\_mode)

Jadog to database SSL mode (jadog\_to\_db\_ssl\_mode) – параметр уровень защиты SSL-соединения утилиты «jadog\_ctl» к СУБД.

Значение по умолчанию - [verify-full]

Возможные значения параметра приведены в таблице 6.2.

Таблица 6.2 – Возможные значения параметра jadog\_to\_db\_ssl\_mode

SSL режим	По умолчанию	Защита от прослушивания	MITM защита	Мотивация к использованию
disable	—	—	—	Меня не волнует безопасность, и я не хочу оплачивать накладные расходы, связанные с шифрованием.
allow	—	Возможно	—	Я не забочусь о безопасности, но я оплачу накладные расходы на шифрование, если сервер будет настаивать на этом.
prefer	—	Возможно	—	Меня не волнует шифрование, но я хочу оплатить накладные расходы на шифрование, если сервер его поддерживает.
require	—	X	—	Я хочу, чтобы мои данные были зашифрованы, и я принимаю накладные расходы. Я верю, что сеть гарантирует, что я всегда подключусь к нужному мне серверу.
verify-ca	—	X	Зависит от СА политики	Я хочу, чтобы мои данные были зашифрованы, и я принимаю накладные расходы. Я хочу быть уверен, что подключаюсь к серверу, которому доверяю.
verify-full	X	X	X	Я хочу, чтобы мои данные были зашифрованы, и я принимаю накладные расходы. Я хочу быть



SSL режим	По умолчанию	Защита от прослушивания	MITM защита	Мотивация к использованию
				уверен, что подключаюсь к серверу, которому доверяю, и что это тот, который я указываю.

#### 6.7.5.12 Подменю 12) Jadog database user name (param\_postgres: db\_jadog\_user)

Jadog database user name (db\_jadog\_user) – имя пользователя СУБД.

Значение по умолчанию – [jadog\_user].

В дальнейшем, от имени и с правами роли «jadog\_user» будет проводиться репликация данных между узлами кластера.

Роль «jadog\_user» создается в СУБД специальной функцией «add\_jadog\_user» после [установки расширения «jadog»](#).

#### 6.7.5.13 Подменю 14) Database server OS user (param\_system: system\_user)

Database server OS user (system\_user) – имя служебного пользователя ОС, от имени и с правами которого будут запускаться служба (демон):

– в GNU Linux:

- jatoba-<ver>;

– в ОС Windows:

- JatobaServer;
- JadogRunAsService.

По умолчанию используется имя – [postgres].

Имя системного пользователя в ОС Windows устанавливается на одном из этапов установки СУБД, как представлено на рисунке 6.23.



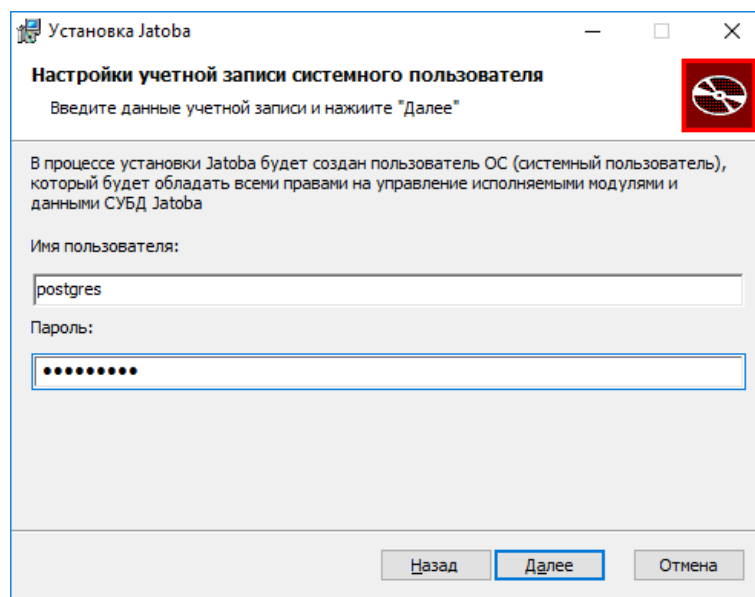


Рисунок 6.23 – Этап установки системного пользователя для СУБД

#### 6.7.5.14 Подменю 15) Replication node name (param\_replication: replication\_slot\_name)

Replication node name (param\_replication: replication\_slot\_name) – имя слота репликации. Это произвольное имя, которое должно быть уникально в каждом узле кластера.

Параметр обязательный. По умолчанию значение не установлено – [].

В рассматриваемом примере устанавливается значение – [node1].



Рисунок 6.24 – Подменю 15) Replication node name (replication\_slot\_name)

#### 6.7.6. Пункт меню 6 «Failover setting»

Компонент имеет настройку для ручного или автоматического режима отказоустойчивости (Failover).

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

Устанавливаются два основных режима отказоустойчивости:

- Auto failover mode (autofailover\_mode) – для автоматической смены роли «Master» в кластере;
- Auto dc promote mode (dc\_autofailover\_mode) – для автоматической смены дата-центра и назначения роли «Master» серверу в резервном дата-центре.

Администратор кластера может регулировать поведение кластера в режиме Failover (DC\_Failover). Режимы взаимосвязаны.

#### **6.7.6.1 Подменю 1) Auto failover mode [true/false] (param\_cluster\_behavior: autofailover\_mode)**

Auto failover mode (param\_cluster\_behavior: autofailover\_mode) – режим автоматической обработки отказа сервера кластера с ролью «Master». Работа в автоматическом режиме переназначения роли «Master» на резервный узел описана в п. «Выход из строя Master в DC1» во второй части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-02.

По умолчанию установлен параметр – [true].

Параметр не обязательный. По умолчанию значение не установлено – [true].



Данный параметр должен быть установлен на всех узлах кластера, и он должен быть одинаков для всех

Отключенный автоматический режим, не производит переназначение роли «Master» на резервный узел.

Администратор получает соответствующее сообщение в журнал событий компонента.

При отключенном автоматическом режиме переназначение роли «Master» на резервный узел, возможно только прямой командой, описанной в подразделе 7.15:

```
switchover [host_ip/host_name/hosts] [Jadog PORT number (port)]
```

#### **6.7.6.2 Подменю 2) Auto dc promote mode [true/false] (param\_cluster\_behavior: dc\_autofailover\_mode)**

Auto dc promote mode (param\_cluster\_behavior: dc\_autofailover\_mode) – режим автоматической смены дата-центра. Работа в режиме «Auto dc promote mode» описана в п

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

«10.2.3. dc\_failover. Аварийный режим № 1. Последовательный выход из строя Master-серверов» во второй части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-02.

По умолчанию установлен параметр – [false].



Данный параметр должен быть установлен на всех узлах кластера, и он должен быть одинаков для всех. Значение параметра проверяется при попытке включить узел в кластер.

```
Fallover settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
1) Auto failover mode [true/false] (param_cluster_behavior:autofailover_mode) [true]
2) Auto dc promote mode [true/false] (param_cluster_behavior:dc_autofailover_mode)[false]
3) Reset all settings to the default values
>
```

Рисунок 6.25 – Параметр Auto dc promote mode [true/false] (dc\_autofailover\_mode)

Если установлен параметр «false», т.е. режим отключен, смена дата-центра в кластере возможна только на узле, принадлежащему дата-центру, в который требуется перенести нагрузку, с помощью команды администратора:

```
datacenter promote
```

Команда описана в п. 7.11 настоящего документа.

### 6.7.7. Пункт меню 7 «Replication setting»

СУБД поддерживает синхронную и асинхронную репликацию.

Синхронная репликация в СУБД – это режим репликации, при котором ведущий и ведомый серверы ждут подтверждения записи с обеих сторон, прежде чем продолжить обработку следующей записи. Этот режим обеспечивает более высокую надежность и согласованность данных, но может снизить производительность системы, так как каждый сервер должен ждать, пока другой сервер подтвердит запись.

Асинхронная репликация в СУБД – это процесс, при котором данные записываются на ведущий сервер, и затем эти изменения реплицируются на один или несколько ведомых серверов в асинхронном режиме. Это означает, что **ведомые сервера не ждут**, пока ведущий сервер подтвердит запись, прежде чем начать обработку этой записи. Вместо этого, ведомые сервера начинают обработку записи сразу после получения ее от ведущего сервера, и

продолжают обработку до тех пор, пока не получают подтверждение записи от ведущего сервера.

В меню «Replication setting» устанавливаются параметры для синхронной репликации кластера.



Для использования асинхронной репликации устанавливать параметры в меню не требуется

#### **6.7.7.1 Подменю 1) Replication node name (param\_replication: replication\_slot\_name)**

Replication node name (param\_replication: replication\_slot\_name) – имя слота репликации.

Зависимый параметр.

Значение подставляется из установленного параметра, описанного в п. 6.7.5.14 «Подменю 15) Replication node name (param\_replication: replication\_slot\_name)» и доступно для редактирования.

#### **6.7.7.2 Подменю 2) Synchronous commit setting (param\_synchronous: synchronous\_commit)**

Synchronous commit setting (param\_synchronous: synchronous\_commit) – настройка типа синхронной репликации.

Параметр «synchronous\_commit» определяет, будет ли сервер при фиксировании транзакции ждать, пока записи из WAL сохранятся на диске, прежде чем сообщить клиенту об успешном завершении операции.

Значение по умолчанию не установлено – [].

Доступны для установки следующие значения параметра:

- off;
- local;
- remote\_write;
- on;
- remote\_apply.

После процедуры конфигурирования кластера установленный параметр сохранится в конфигурационном файле «postgresql.conf» в разделе «WRITE-AHEAD LOG».

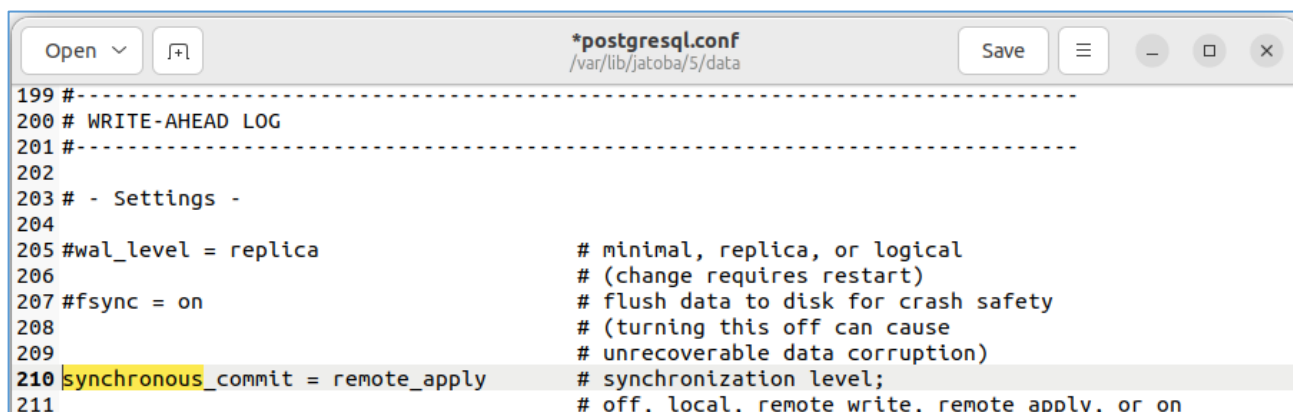


Рисунок 6.26 – Расположение параметра в конфигурационном файле «postgresql.conf»

В зависимости от установленного параметра достигается уровень гарантий сохранности транзакций, представленный в таблице 6.3.

Таблица 6.3 – Уровни гарантий сохранности транзакций

Значение «synchronous commit»	Гарантированная локальная фиксация	Гарантированная фиксация на ведомом сервере после сбоя СУБД	Гарантированная фиксация на ведомом сервере после сбоя ОС	Согласованность запросов на ведомом сервере
remote_apply	X	X	X	X
on	X	X	X	—
remote_write	X	X	—	—
local	X	—	—	—
off	—	—	—	—

```

root@node1: /usr/jatoba-6/bin

Replication settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Synchronous commit can take the following values: on, remote_apply, remote_write, local, off.
* "on" - commit transaction after apply (wal will reply) on synchronous nodes. Not guarantee of consistency;
* "remote_apply" - commit transaction after apply (wal will reply) on synchronous nodes. Guarantee of consistency;
* "remote_write" - commit transaction after write WAL into the filesystem on synchronous nodes;
* "local" - allow only local synchronous commit;
* "off" - switch off synchronous commit and synchronous replication.

Synchronous type can take the following values: FIRST, ANY
Wal_level parameter can take the following values: replica, logical

1) Replication node name (param_replication:replication_slot_name) [node1]
2) Synchronous commit setting (param_synchronous:synchronous_commit) [remote_apply]
3) Synchronous type (param_synchronous:synchronous_type) []
4) The number of synchronous standbys (param_synchronous:synchronous_commit_max_nodes)[]
5) The wal_level value (param_postgres:wal_level) [replica]
6) Reset all settings to the default values

>

```

Рисунок 6.27 – Установленный параметр «synchronous\_commit»

### 6.7.7.3 Подменю 3) Synchronous type (param\_synchronous: synchronous\_type)

Synchronous type (param\_synchronous: synchronous\_type) – тип синхронной репликации.

Зависимый параметр.

Значение по умолчанию не установлено – [].

Доступны для установки следующие значения параметра:

- ANY;
- FIRST.

Параметры «FIRST» и «ANY» задают тип выбора серверов с ролью «Slave».

Значение «ANY», в сочетании с числом резервных узлов, устанавливаемых в параметре «synchronous\_commit\_max\_nodes» в подменю 4, выбирает синхронную репликацию **на основе кворума**, когда транзакции фиксируются только после того, как их записи в WAL реплицируются на как минимум число резервных перечисленных узлов кластера. Транзакция фиксируется после получения ответа от указанного количества серверов.

Значение «FIRST», в сочетании с числом резервных узлов, устанавливаемых в параметре «synchronous\_commit\_max\_nodes» в подменю 4, выбирает синхронную репликацию **на основе приоритетов**, когда транзакции фиксируются только после того, как

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

их записи в WAL реплицируются на число резервных узлов и выбираемых **согласно приоритетам**.

```

root@node1: /usr/jatoba-6/bin

Replication settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Synchronous commit can take the following values: on, remote_apply, remote_write, local, off.
* "on" - commit transaction after apply (wal will reply) on synchronous nodes. Not guarantee of consistency;
* "remote_apply" - commit transaction after apply (wal will reply) on synchronous nodes. Guarantee of consistency;
* "remote_write" - commit transaction after write WAL into the filesystem on synchronous nodes;
* "local" - allow only local synchronous commit;
* "off" - switch off synchronous commit and synchronous replication.

Synchronous type can take the following values: FIRST, ANY
Wal_level parameter can take the following values: replica, logical

1) Replication node name (param_replication:replication_slot_name) [node1]
2) Synchronous commit setting (param_synchronous:synchronous_commit) [remote_apply]
3) Synchronous type (param_synchronous:synchronous_type) [ANY]
4) The number of synchronous standbys (param_synchronous:synchronous_commit_max_nodes)[]
5) The wal_level value (param_postgres:wal_level) [replica]
6) Reset all settings to the default values

>

```

Рисунок 6.28 – Установленный параметр «synchronous type»

#### 6.7.7.4 Подменю 4) The number of synchronous standbys (param\_synchronous:synchronous\_commit\_max\_nodes)

The number of synchronous standbys (param\_synchronous:synchronous\_commit\_max\_nodes) – количество резервных узлов, т.е. серверов с ролью «Slave» требующих подтверждения синхронной репликации.

Значение по умолчанию не установлено – [].

В качестве значения используется целое, положительное число не равное «0».

При указании количества резервных узлов, т.е. серверов с ролью «Slave», следует указывать разумно достаточное количество. Допустимо указать меньшее количество от реального, в этом случае указанное количество будет гарантировано выполнять синхронную репликацию, согласно установленного типа синхронизации в параметре «synchronous type», а дополнительные резервные узлы будут выполнять репликацию с меньшим приоритетом.



В случае если значение параметра «The number of synchronous standbys» не задано (пустое значение), для добавляемых в кластер резервных узлов по умолчанию будет настроен асинхронный тип репликации.



### 6.7.7.5 Подменю 5) The wal\_level value (param\_postgres:wal\_level)

The wal\_level value (param\_postgres:wal\_level) определяет, какой объем информации записывается в WAL.

Значение по умолчанию – [replica].

### 6.7.8. Пункт меню 8 «WAL archive and restore settings»

Пункт меню 8 «WAL archive and restore settings» – настройки архивирования и восстановления WAL, предназначен для установки параметров и команд резервного копирования. При первоначальной настройке кластера включение механизма резервного копирования WAL необязательно.

Пример настройки монтирования сетевого каталога для настройки архивирования WAL приведен в Приложении 1 документа «Компонент jaDog. Управление режимом работы узлов кластера (версия 3.2, часть 2)» 643.72410666.00067-07 98 02-02.

```
WAL archive and restore settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
1) System directory for WAL archive/restore (param_archive:wal_archive_directory)[]
2) WAL archive copy command (param_archive:wal_archive_command) []
3) WAL archive restore command (param_restore:wal_restore_command) []
4) WAL archive cleanup process on (param_archive:wal_archive_cleanup_needed) [false]
5) WAL archive cleanup command (param_archive:wal_archive_cleanup_command) []
6) WAL archive cleanup timeout (param_archive:wal_archive_cleanup_timeout) [30000]
7) Reset all settings to the default values
>
```

Рисунок 6.29 – Меню 6 «WAL archive and restore settings»

#### 6.7.8.1 Подменю 1) System directory for WAL archive/restore (param\_archive:wal\_archive\_directory)

System directory for WAL archive/restore (param\_archive: wal\_archive\_directory) – системный каталог для архивирования/восстановления WAL.

Значение по умолчанию не установлено – [].

Допустимо указывать как прямой, так и относительный путь к каталогу.

В рассматриваемом примере примонтирован сетевой каталог:

```
/nfs/arhive_wal
```



```

WAL archive and restore settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
1) System directory for WAL archive/restore (param_archive:wal_archive_directory)[]
2) WAL archive copy command (param_archive:wal_archive_command) []
3) WAL archive restore command (param_restore:wal_restore_command) []
4) WAL archive cleanup process on (param_archive:wal_archive_cleanup_needed) [false]
5) WAL archive cleanup command (param_archive:wal_archive_cleanup_command) []
6) WAL archive cleanup timeout (param_archive:wal_archive_cleanup_timeout) [30000]
-----
7) Reset all settings to the default values
-----
> 1
Enter new parameter value
> /nfs/arhive_wal

```

Рисунок 6.30 – Установки параметра «System directory for WAL archive/restore»



После установки параметра «System directory for WAL archive/restore» остальные параметры установятся автоматически.

```

WAL archive and restore settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
1) System directory for WAL archive/restore (param_archive:wal_archive_directory)[/nfs/arhive_wal]
2) WAL archive copy command (param_archive:wal_archive_command) [cp %p /nfs/arhive_wal/%f]
3) WAL archive restore command (param_restore:wal_restore_command) [cp /nfs/arhive_wal/%f %p]
4) WAL archive cleanup process on (param_archive:wal_archive_cleanup_needed) [false]
5) WAL archive cleanup command (param_archive:wal_archive_cleanup_command) [/usr/jatoba-6/bin/pg_archivecleanup /nfs/arhive_wal
 %s]
6) WAL archive cleanup timeout (param_archive:wal_archive_cleanup_timeout) [30000]
-----
7) Reset all settings to the default values
-----
>

```

Рисунок 6.31 – Автоматическая установка параметров

#### 6.7.8.2 Подменю 2) WAL archive copy command (param\_archive: wal\_archive\_command)

WAL archive copy command (param\_archive: wal\_archive\_command) – команда копирования архива WAL.

Значение по умолчанию не установлено – [].

#### 6.7.8.3 Подменю 3) WAL archive restore command (param\_restore: wal\_restore\_command)

WAL archive restore command (param\_restore: wal\_restore\_command) – команда восстановления архива WAL.

Значение по умолчанию не установлено – [].

#### 6.7.8.4 Подменю 4) WAL archive cleanup process on (param\_archive: wal\_archive\_cleanup\_needed)

WAL archive cleanup process on (param\_archive: wal\_archive\_cleanup\_needed) – признак необходимости запуска процесса очистки архива WAL.

По умолчанию установлено значение – [false].

#### 6.7.8.5 Подменю 5) WAL archive cleanup command (param\_archive: wal\_archive\_cleanup\_command)

WAL archive cleanup command (param\_archive: wal\_archive\_cleanup\_command) – команда очистки архива WAL.

Значение по умолчанию не установлено – [].

#### 6.7.8.6 Подменю 6) WAL archive cleanup timeout (param\_archive: wal\_archive\_cleanup\_timeout)

WAL archive cleanup timeout (param\_archive: wal\_archive\_cleanup\_timeout) – тайм-аут очистки архива WAL.

Значение по умолчанию – [30000].

### 6.7.9. Пункт меню 9 «Reporting and logging»

Пункт меню 9 «Reporting and logging» – отчетность и ведение журнала, служит для настройки журнала событий кластера и событий безопасности кластера.

Установка параметра Log format (logs\_type) в значение «csv» позволит компоненту централизованного сбора записей событий СУБД «ja\_Log», собирать и сохранять события в служебную БД.

```
Jadog 3.1.0 Setup
Reporting and logging Menu

Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Log format can take the following comma-separated values: txt, csv, json
And special security values: security.txt, security.csv, security.json
Max log file usage duration can take the following units: s, min, h, d
Max log file size can take the following units: B, kB, MB, GB, TB
Log verbosity can take the following values: fatal, error, warning, info, debug1, debug2, debug3

1) Log directory (param_path:log_path) [usr/jatoba-5/var/log/jadog]
2) Log filename (param_log:logs_filename) [jadog-%a]
3) Log file mode (param_log:logs_file_mode) [0600]
4) Log format (param_log:logs_type) [csv]
5) Max log file usage duration (param_log:logs_rotation_age) [1d]
6) Max log file size (param_log:logs_rotation_size) [10MB]
7) Truncate, not append, logs (param_log:logs_truncate_on_rotation) [false]
8) Log verbosity (param_log:logs_level) [info]
9) Security log directory (param_security_log:security_log_path) [usr/jatoba-5/var/log/jadog]
10) Security log filename (param_security_log:security_logs_filename) [jadog-%a]
11) Security log file mode (param_security_log:security_logs_filemode) [0600]
12) Allow to write into file (param_log:logs_file) [true]
13) Allow to write into stdout (param_log:logs_screen) [false]

14) Reset all settings to the default values

> |
```

Рисунок 6.32 – Пункт меню 9 «Reporting and logging»

#### 6.7.9.1 Подменю 1) Log directory (param\_path: log\_path)

Log directory (param\_path: log\_path) – директория хранения журнала событий кластера.

Значение по умолчанию

- В GNU Linux – [/usr/jatoba-6/var/log/jadog];
- В ОС Windows - [C:\Program Files\GIS\Jatoba\6\var\log\jadog].

Путь к директории указывается как абсолютный.

#### 6.7.9.2 Подменю 2) Log filename (param\_log: logs\_filename)

Log filename (param\_log: logs\_filename) – имя файла журнала событий кластера.

Значение по умолчанию – [jadog-%a].

Допускается синтаксис шаблонов имен файлов для журналов событий.

#### Например

```
logs_filename = 'jadog-%Y-%m-%d_%H%M%S'
```

#### 6.7.9.3 Подменю 3) Log file mode (param\_log: logs\_file\_mode)

Log file mode (param\_log: logs\_file\_mode) – режим доступа к файлам журнала событий кластера.

Значение по умолчанию – [0600].

В GNU Linux устанавливаются права доступа на файлы журналов кластера и используется значение по умолчанию 0600, т. е. только владелец сервера может читать и писать в файлы журналов.

Может использоваться значение 0640, разрешающее чтение файлов членам группы.

В ОС семейства Windows данный параметр игнорируется.

#### 6.7.9.4 Подменю 4) Log format (param\_log: logs\_type)

Log format (param\_log: logs\_type) – формат файла журнала событий кластера

Значение по умолчанию – [txt].

Поддерживаются форматы файла журнала:

- txt;

- csv;
- json.

Допустимо установка нескольких значения. Кроме того, поддерживаются специальные форматы журнала событий ИБ:

- security.json;
- security.csv;
- security.txt.

### Например

```
logs_type = 'csv,json'
```

Для интеграции с компонентом «ja\_Log» рекомендуется установить формат файла \*.csv.

#### 6.7.9.5 Подменю 5) Max log file usage duration (param\_log: logs\_rotation\_age)

Max log file usage duration – параметр определяет максимальное количество времени для использования отдельного файла журнала, по истечении которого будет создан новый файл журнала, т.е. время ротации.

Если это значение указано без единиц измерения, оно принимается за минуты. Значение по умолчанию равно 24 часам.

Установите значение ноль, чтобы отключить создание новых файлов журнала на основе времени.

Значение по умолчанию – [1d].

Поддерживаются следующие единицы измерения:

- s (секунды);
- min (минуты);
- h (часы);
- d (дни).

В качестве значения используется целое, положительное число.

#### 6.7.9.6 Подменю 6) Max log file size (param\_log: logs\_rotation\_size)

Max log file size – параметр определяет максимальный размер отдельного файла журнала. После того, как этот объем данных будет передан в файл журнала, будет создан новый файл журнала. Если это значение указано без единиц измерения, оно принимается в килобайтах. Значение по умолчанию равно 10 мегабайтам.

Установите значение ноль, чтобы отключить создание новых файлов журнала на основе размера.

Значение по умолчанию – [100MB].

Поддерживаются следующие единицы измерения:

- B (байт);
- kB (килобайт);
- MB (мегабайт);
- GB (гигабайт);
- TB (терабайт).

В качестве значения используется целое, положительное число «0».

#### 6.7.9.7 Подменю 7) Truncate, not append, logs (param\_log: logs\_truncate\_on\_rotation)

Truncate, not append, logs – установка параметра приведет к тому, что компонент «jaDog» перезапишет файл журнала (признак перезаписи файла логов), а не добавит к любому существующему с таким же именем.

Удаление данных файла журнала событий будет происходить только при открытии нового файла из-за ротации по времени, а не во время запуска сервера или ротации по размеру. Если выключено, во всех случаях будут добавлены ранее существовавшие файлы.

Значение по умолчанию – [false].



Значение «true» не должно использоваться при контроле событий ИБ

**Например**

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

Использование этого параметра в сочетании с `log_filename`, таким как `jadog-%H.csv` приведет к созданию двадцатичетырехчасовых файлов журнала, а затем к их циклической перезаписи.

### Пример 1

Для хранения журналов за 7 дней необходимо использовать один файл журнала в день с именем `jadog_.Mon.csv`, `jadog_Thu.csv` и т.д., и автоматически перезаписывать журнал прошлой недели журналом этой недели, установить для `log_filename` значение `jadog_log.%a`, для `log_truncate_on_rotation` значение `on`, а для `log_rotation_age` значение `1440`.

### Пример 2

Для ведения журналов в течение 24 часов по одному файлу журнала в час, но также выполнять ротацию раньше, если размер файла журнала превышает 1 ГБ, необходимо установить для `log_filename` значение `jadog_log.%H%M`, для `log_truncate_on_rotation` значение `on`, для `log_rotation_age` значение `60`, а для `log_rotation_size` значение `1000000`. Включение `%M` в `log_filename` позволяет выполнять любые зависящие от размера ротации, которые могут произойти, чтобы выбрать имя файла, отличное от исходного имени файла часа.

#### 6.7.9.8 Подменю 8) Log verbosity (param\_log: logs\_level)

Log verbosity (`logs_level`) – уровень детализации журнала сообщений. Доступно установить значения `FATAL`, `ERROR`, `WARNING`, `INFO`, `DEBUG1`, `DEBUG2`, `DEBUG3`.

Значение по умолчанию – `[info]`.

#### 6.7.9.9 Подменю 9) Security log directory (param\_security\_log: security\_log\_path)

Security log directory (`security_log_path`) – директория хранения журнала событий безопасности кластера.

Значение по умолчанию:

- В GNU Linux – `[/usr/jatoba-6/var/log/jadog]`;
- В ОС Windows - `[C:\Program Files\GIS\Jatoba\6\var\log\jadog]`.

Путь к директории указывается как абсолютный.

В данном подменю реализована функциональная возможность компонента хранить события безопасности отдельно от журнала событий кластера.

По умолчанию установлен путь /usr/jatoba-6/var/log/jadog, как в «Подменю 1) Log directory (log\_path)» (6.7.9.1). В этом случае события безопасности будут записываться в журнал событий кластера.

Для отдельного хранения событий безопасности путь по умолчанию необходимо изменить.

В соответствии с «ГОСТ Р 59548-2022. Национальный стандарт Российской Федерации. Защита информации. Регистрация событий безопасности. Требования к регистрируемой информации» будут храниться:

- события безопасности, связанные с идентификацией и аутентификацией субъекта доступа;
- события безопасности, связанные с управлением учетными записями пользователей;
- события безопасности, связанные с изменением настроек общего программного обеспечения;
- события безопасности, связанные с управлением запуском/остановкой компонентов программного обеспечения;
- события безопасности, связанные с выполнением резервного копирования;
- события безопасности, связанные с управлением резервным копированием;
- события безопасности, связанные с выполнением восстановления информации;
- события безопасности, связанные с управлением журналами (записями) регистрации событий безопасности.

**6.7.9.10 Подменю 10) Security log filename (param\_security\_log: security\_logs\_filename)**

Log filename (security\_logs\_filename) – имя файла журнала событий безопасности кластера.

Значение по умолчанию – [jadog-%a].

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

Допускается синтаксис шаблонов имен файлов для журналов событий, как описано в «Подменю 2) Log filename (logs\_filename)».

#### 6.7.9.11 Подменю 11) Security log file mode (param\_security\_log: security\_logs\_filemode)

Security log file mode (security\_logs\_filemode) – режим доступа к файлам журнала событий безопасности кластера.

Значение по умолчанию – [0600]. Параметр аналогичен описанному в «Подменю 3) Log file mode (logs\_file\_mode)».

#### 6.7.9.12 Подменю 12) Allow to write into file (param\_log: logs\_file)

Allow to write into file (logs\_file) – разрешить запись в файл.

Значение по умолчанию – [true].

#### 6.7.9.13 Подменю 13) Allow to write into stdout (param\_log: logs\_screen)

Allow to write into stdout (logs\_screen) – разрешить запись в стандартный вывод (консоль).

Значение по умолчанию – [false].

Сброс установленных параметров.

### 6.7.10. Пункт меню 10 «REST API Settings»

Пункт меню 10 «REST API Settings» – настройки управления REST API.

```
REST API settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
1) REST API use (param_rest_api:rest_api_use) [false]
2) REST API listen address (param_rest_api:rest_api_listen_address) [0.0.0.0]
3) REST API listen port (param_rest_api:rest_api_listen_port) [54443]
4) REST API TLS server certificate (param_rest_api:rest_api_cert_file) []
5) REST API TLS server private key (param_rest_api:rest_api_key_file) []
6) REST API TLS CA bundle (param_rest_api:rest_api_ca_file) []
7) REST API TLS server revocation list (param_rest_api:rest_api_crl_file) []
8) Reset all settings to the default values
> |
```

Рисунок 6.33 – Пункт меню 10 «REST API Settings»

#### 6.7.10.1 Пункт 1) REST API use (param\_rest\_api:rest\_api\_use)

Включение доступа к REST API.

Значение по умолчанию – [false].



**6.7.10.2 Пункт 2) REST API listen address (param\_rest\_api:rest\_api\_listen\_address)**

IP-адрес, по которому доступен REST API.

Значение по умолчанию – [0.0.0.0].

**6.7.10.3 Пункт 3) REST API listen port (param\_rest\_api:rest\_api\_listen\_port)**

Сетевой порт, по которому доступен REST API.

Значение по умолчанию – [54443].

**6.7.10.4 Пункт 4) REST API TLS server certificate (param\_rest\_api:rest\_api\_cert\_file)**

REST API TLS server certificate (param\_rest\_api:rest\_api\_cert\_file) – директория хранения файла сертификата, который используется для подключения к REST API.

Значение по умолчанию – [].

**6.7.10.5 Пункт 5) REST API TLS server private key (param\_rest\_api:rest\_api\_key\_file)**

REST API TLS server private key (param\_rest\_api:rest\_api\_key\_file) ) – директория хранения файла с ключом, который используется для подключения к REST API.

Значение по умолчанию – [].

**6.7.10.6 Пункт 6) REST API TLS CA bundle (param\_rest\_api:rest\_api\_ca\_file)**

REST API TLS CA bundle (param\_rest\_api:rest\_api\_ca\_file) – директория хранения до сертификата центра сертификации, который используется для подключения к REST API.

Значение по умолчанию – [].

**6.7.10.7 Пункт 7) REST API TLS server revocation list (param\_rest\_api:rest\_api\_crl\_file)**

REST API TLS server revocation list (param\_rest\_api:rest\_api\_crl\_file) – список аннулированных сертификатов, которые использовались для подключения к REST API.

Значение по умолчанию – [].

**6.7.11. Пункт меню 11 «Recovery settings»**

Пункт меню 11 «Recovery settings» – настройки восстановления, предназначен для установки параметров и команд восстановления состояния кластера после устранения сбоев.

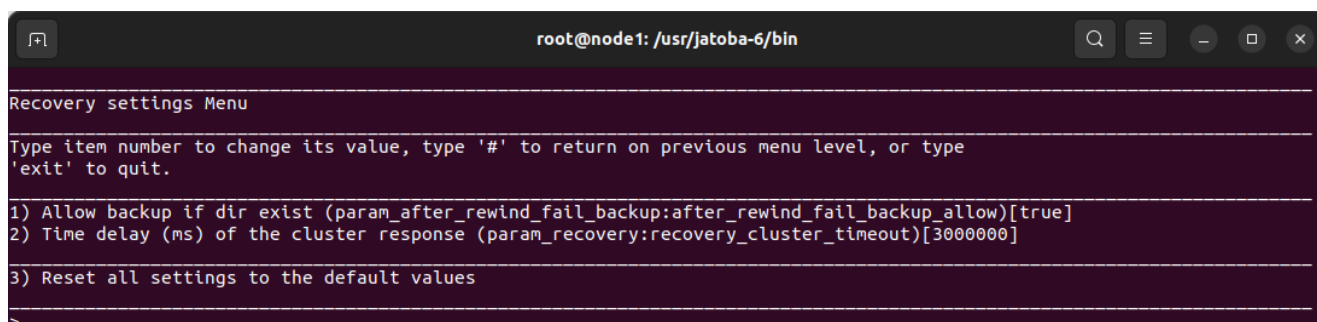


Рисунок 6.34 – Пункт меню 10 «Recovery settings»

#### 6.7.11.1 Пункт 1) Allow backup if dir exist (param\_after\_rewind\_fail\_backup:after\_rewind\_fail\_backup\_allow)

Allow backup if dir exist

(param\_after\_rewind\_fail\_backup:after\_rewind\_fail\_backup\_allow) – разрешение на создание резервной копии восстанавливаемых данных в процессе восстановления состояния кластера.

Значение по умолчанию – [true].

#### 6.7.11.2 Пункт 2) Time delay (ms) of the cluster response (param\_recovery:recovery\_cluster\_timeout)

Time delay (ms) of the cluster response (param\_recovery:recovery\_cluster\_timeout) – время (мс) в течении которого будет выполняться попытка восстановления состояния кластера.

Значение по умолчанию – [3000000].

#### 6.7.11.3 Пункт 3) Reset all settings to the default values

Reset all settings to the default values – перезагрузка всех установленных параметров в данном меню.

#### 6.7.12. Пункт меню 12 «Reset all settings to the default values»

Reset all settings to the default values – перезагрузка всех значений и возврат к настройкам по умолчанию.

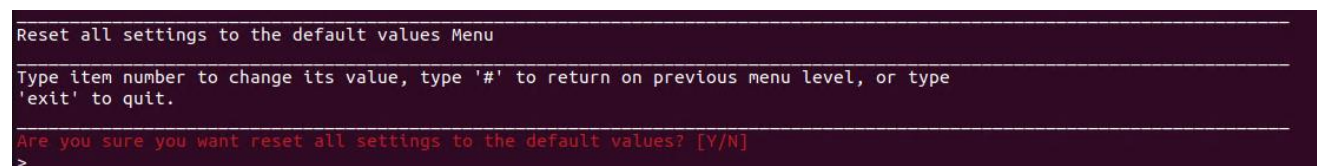


Рисунок 6.35 – Диалог сброса всех настроек к значениям по умолчанию

### 6.7.13. Пункт меню 13 «Check and show all settings»

Проконтролировать все введенные настройки, выбрав пункт 13 «Check and show all settings» (рисунок 6.36). Далее настройки будут записаны в конфигурационный файл компонента «jaDog». Подтвердить настройки нажатием ENTER и вернуться в главное меню.



В дальнейшем эти параметры могут быть изменены в конфигурационном файле компонента «jaDog» вручную или через jadog setup.

```

root@node1: /usr/jatoba-5/bin
param_security_log:security_log_path/usr/jatoba-5/var/log/jadog
param_ssl:ssl false
param_ssl:ssl_ca_file
param_ssl:ssl_cert_file
param_ssl:ssl_crl_file
param_ssl:ssl_key_file
param_ssl:ssl_engine
param_archive:wal_archive_directory/nfs/arhive_wal
param_archive:wal_archive_commandcp %p /nfs/arhive_wal/%f
param_restore:wal_restore_commandcp /nfs/arhive_wal/%f %p
param_archive:wal_archive_cleanup_command/usr/jatoba-5/bin/pg_archivecleanup /nfs/arhive_wal %s
param_archive:wal_archive_cleanup_neededfalse
param_archive:wal_archive_cleanup_timeout30000
param_cluster_behavior:autofailover_modetrue
param_cluster_behavior:dc_autofailover_modetrue
param_synchronous:synchronous_commitremote_apply
param_synchronous:synchronous_typeANY
param_synchronous:synchronous_commit_max_nodes1
param_cluster_behavior:master_sync_state_after_switchoversync
-----users.txt-----
Account #1 [admin/BD831BC3BCB76A812EA8F023C9521E80AA316AC4AA29CDD3636479DABD9B1003]
Account #2 [admin1/139F38C5AF047FF8355B388060CDFB8C0B299D4F0D74FE918A929353C9B4A48F]
Account #3 [admin2/EB625A0EC07D159C97D579444881347DEADFAAB04282AC79293F83B41FEFFF3B]

[PRESS ENTER TO CONTINUE]

```

Рисунок 6.36 – Сводные данные по введенным настройкам

Если настройки корректны, то становится доступно их сохранение.

### 6.7.14. Пункт меню 14 «Save setting and setup jadog»

Сохранить введенные настройки и запустить инициализацию сервисов компонента «jaDog», для чего выбрать пункт 14 «Save setting and setup jadog» (рисунок 6.37). Запустится проверка корректности применения введенных параметров. Например, проверка существования системного пользователя, проверка существования директорий СУБД и так далее.

```
Save settings and setup jadog Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
CHECK PARAMETERS:
ERRORS in parameters are not detected

CHECK USERS:
SUCCESS

SETUP DIRECTORIES:
ALL Directories were created succesfully
SETUP SERVICES :
jadog was successfully installed
RESULT::Configuration completed successfully

[PRESS ENTER TO CONTINUE]
```

Рисунок 6.37 – Сохранение настроек

#### 6.7.14.1 Параметры созданного узла (jadog.yml)

Параметры созданного узла кластера сохраняются в конфигурационном файле в формате YAML.

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

```
mc [root@node1]:/usr/jatoba-6/etc/jadog
/usr/jatoba-6/etc/jadog/jadog.yml 1854/12385 14%
# -----
# jaDog configuration file
# -----
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the jaDog.
#
# This file is read on jaDog startup.
# If you edit the file on a running system, you have to reload the
# service for the changes to take effect
# Some parameters, which are marked below,
# require a service shutdown and restart to take effect.
#
# Memory units: B = bytes           Time units: us = microseconds
#               kB = kilobytes       ms = milliseconds
#               MB = megabytes        s = seconds
#               GB = gigabytes        min = minutes
#               TB = terabytes        h = hours
#                                   d = days
#
#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 # Строка подключения к субд
  max_admin_connections: 10                    # Максимальное количество пользовательских подключений к субд
#Параметры ja hipe
param_ja_hipe:
  db_name: ""                                  # Имя базы данных ja hipe
```

Рисунок 6.38 – Содержание конфигурационного файла jadog.yml

Конфигурационный файл содержит необходимые для понимания параметров комментарии.

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

Структура файла и описание параметров приведены в приложении 1. Настоящего документа.

#### 6.7.15. Запуск серверной части компонента «jaDog» на главном узле

Запуск серверной части компонента «jaDog» выполняется в следующем порядке:

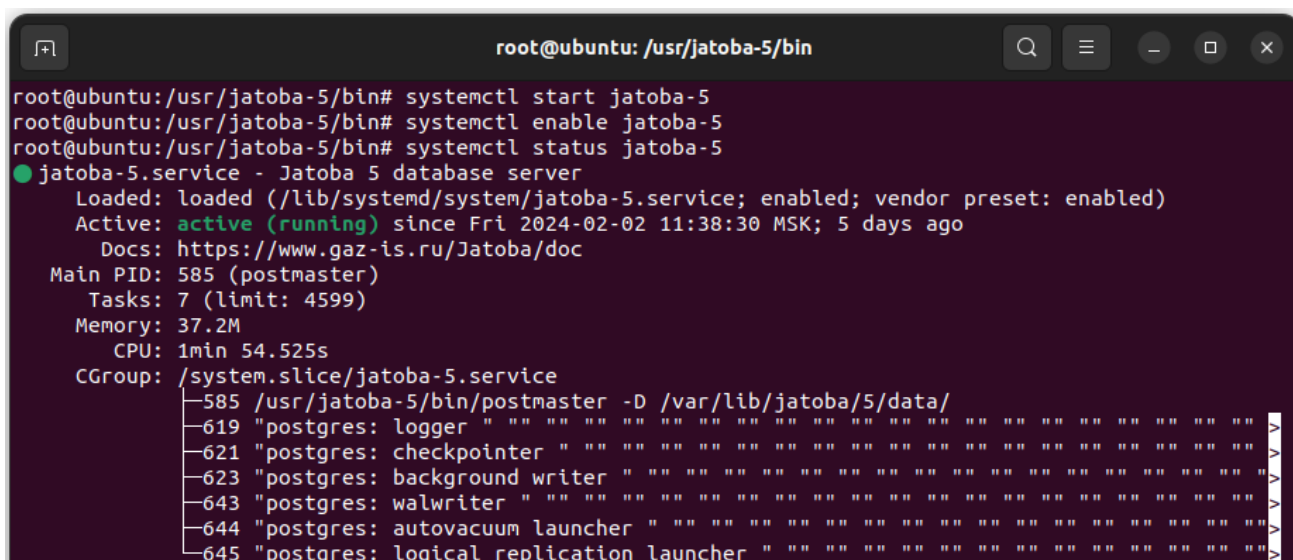
- 1) Запустить сервис jatoba.
- 2) Добавить сервис jatoba в автозапуск:
  - В GNU Linux:

```
systemctl start jatoba-6
systemctl enable jatoba-6
systemctl status jatoba-6
```

- в ОС Windows:

```
sc query jatobaserver
```

- 3) Проверить статус службы jatoba.

A screenshot of a terminal window titled 'root@ubuntu: /usr/jatoba-5/bin'. The terminal shows the following commands and output:

```
root@ubuntu: /usr/jatoba-5/bin# systemctl start jatoba-5
root@ubuntu: /usr/jatoba-5/bin# systemctl enable jatoba-5
root@ubuntu: /usr/jatoba-5/bin# systemctl status jatoba-5
● jatoba-5.service - Jatoba 5 database server
   Loaded: loaded (/lib/systemd/system/jatoba-5.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-02-02 11:38:30 MSK; 5 days ago
     Docs: https://www.gaz-is.ru/Jatoba/doc
   Main PID: 585 (postmaster)
    Tasks: 7 (limit: 4599)
   Memory: 37.2M
      CPU: 1min 54.525s
   CGroup: /system.slice/jatoba-5.service
           └─585 /usr/jatoba-5/bin/postmaster -D /var/lib/jatoba/5/data/
             └─619 "postgres: logger"
               └─621 "postgres: checkpointer"
                 └─623 "postgres: background writer"
                   └─643 "postgres: walwriter"
                     └─644 "postgres: autovacuum launcher"
                       └─645 "postgres: logical replication launcher"
```

Рисунок 6.39 – Статус службы jatoba-6

- 4) Запустить сервис jadog.
- 5) Добавить сервис jadog в автозапуск:
  - В GNU Linux:

```
systemctl start jadog
```

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

– в ОС Windows:

```
sc start jadogservice  
sc query jadogservice
```

6) Проверить статус сервиса jadog.



```
root@node1: /usr/jatoba-5/bin  
root@node1:/usr/jatoba-5/bin# systemctl start jadog  
root@node1:/usr/jatoba-5/bin# systemctl enable jadog  
root@node1:/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 Fri 2024-06-21 09:51:47 MSK; 3 days ago  
     Main PID: 17184 (jadog)  
       Tasks: 11 (limit: 8738)  
      Memory: 4.0M  
     CGroup: /system.slice/jadog.service  
             └─17184 /usr/jatoba-5/bin/jadog -C /usr/jatoba-5/etc/jadog  
  
Jun 21 09:51:47 node1 systemd[1]: Started "Jatoba failover control service".  
root@node1:/usr/jatoba-5/bin#
```

Рисунок 6.40 – Статус службы jadog

На данном шаге настройка главного узла кластера закончена.

## 6.8. Настройка и запуск компонента «jaDog» на резервном узле

В первую очередь необходимо уточнить, что резервные узлы (с ролью «Slave») в составе кластера могут работать в двух режимах:

– Асинхронный режим: Главному узлу кластера не нужно ждать подтверждения от резервных узлов, прежде чем сообщать о коммите. Этот подход быстрее, но если главный узел кластера выйдет из строя до того, как данные будут реплицированы на резервные узлы, это может привести к потере данных. Добавление в кластер резервного узла в асинхронном режиме приведено в п.п. 6.15;

– Синхронный режим: В этом режиме главный узел (с ролью «Master») должен дожидаться, пока первый доступный резервный узел получит и сохранит файл журнала транзакций, прежде чем сообщать об успешном коммите. Это полезно при настройке высокой доступности, хотя может возникнуть небольшая задержка. Добавление в кластер резервного узла в синхронном режиме приведено в п.п. 6.16.



### 6.8.1. Конфигурирование и запуск компонента «jaDog» на резервном узле

Конфигурирование компонента на резервном узле выполняется аналогично настройке главного узла.

Перейдите в каталог СУБД и запустите файл «jadow.exe» с параметром «setup» для перехода в режим конфигурирования компонента:

```
sudo su
cd /usr/jatoba-6/bin
./jadow setup
```

Требуется установить параметры компонента в пунктах меню:

- [Пункт меню 1 «Database server and jadow directory settings»](#) (п. 6.7.1);
- [Пункт меню 2 «Inter-jadow communication setting»](#) (п. 6.7.2);

В данном пункте меню устанавливается IP-адрес резервного узла кластера. В рассматриваемом примере устанавливается значение – 10.116.102.55/24.

2) Jadow IP address (param_jadow:ip)	[10.116.102.55/24]
--------------------------------------	--------------------

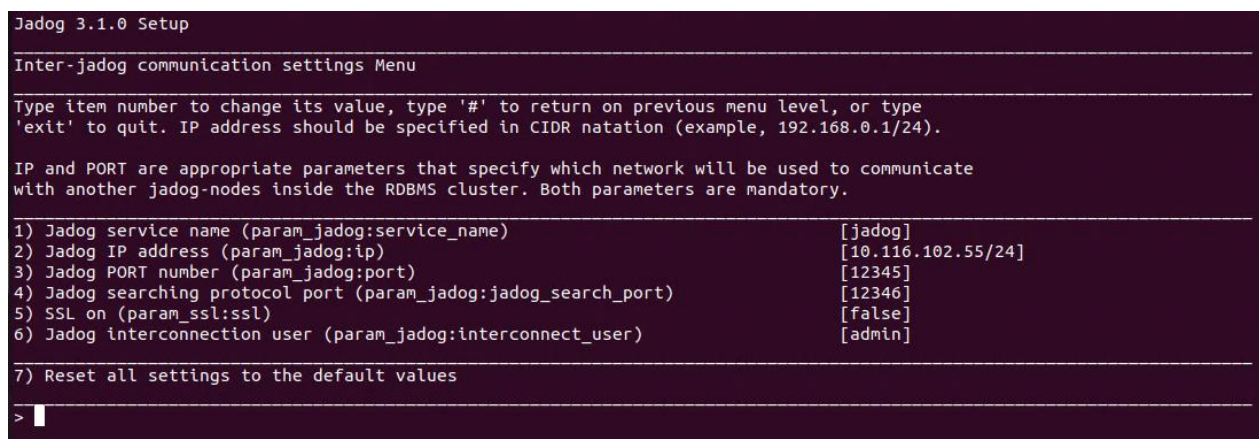


Рисунок 6.41 – Устанавливаемые параметры в подменю «Inter-jadow communication setting»

- [Пункт меню 3 «User / Admin access network setting»](#) (п. 6.7.3);

В пункте меню устанавливается:

#### Подменю 1) Public IP address (param\_jadow: public\_address)

Public IP address – публичный IP-адрес. В рассматриваемом примере устанавливается значение – 10.116.102.81/24, тот же IP-адрес, как и на главном узле (п. 5.8.3.1).

1) Public IP address (public\_address) [10.116.102.81/24]

**Подменю 2) Public address control interval (param\_jadog:  
public\_address\_control\_interval)**

Устанавливается тот же параметр, как и на главном узле (п. 6.7.3.2).

2) Public address control interval  
(param\_jadog:public\_address\_control\_interval) [5000]

**Подменю 3) Public address control attempts (param\_jadog:  
public\_address\_control\_attempts)**

Устанавливается тот же параметр, как и на главном узле (п. 6.7.3.3).

3) Public address control attempts  
(param\_jadog:public\_address\_control\_attempts) [3]

**Подменю 4) DB monitor timeout (param\_postgres:db\_check\_interval)**

Устанавливается тот же параметр, как и на главном узле (п. 6.7.5.4).

4) DB monitor timeout (param\_postgres:db\_check\_interval) [5000]

**Подменю 5) Trusted IP address (param\_jadog: trusted\_address)**

Trusted IP address (trusted\_address) – доверенный адрес.

Устанавливается тот же параметр, как и на главном узле (п. 6.7.3.5).

5) Trusted IP address (trusted\_address) []

**Подменю 6) RDBMS trusted ip (param\_jadog: rdbms\_trusted\_ip)**

RDBMS trusted port (rdbms\_trusted\_port) – доверенный порт системы управления  
реляционными базами данных.

Устанавливается тот же параметр, как и на главном узле (п. 6.7.3.6).

6) RDBMS trusted ip (rdbms\_trusted\_ip) []

**Подменю 7) RDBMS trusted port (param\_jadog: rdbms\_trusted\_port)**



RDBMS trusted port (rdbms\_trusted\_port) – доверенный порт системы управления реляционными базами данных.

Устанавливается тот же параметр, как и на главном узле (п. 6.7.3.7).

7) RDBMS trusted port (rdbms_trusted_port)	[ ]
--	-----

#### Подменю 8) Network interface name (param\_jadog: network\_interface)

Network interface name (network\_interface) – имя интерфейса на данном узле кластера, на который осуществляется привязка общего адреса.

Устанавливается параметр, полученный от ОС (п. 6.7.3.8).

8) Network interface name (network_interface)	[ens33]
---	---------

#### Подменю 9) Jadog TCP port (param\_jadog:user\_interface\_port)

Jadog TCP port (user\_interface\_port) – порт доступа административных подключений для управления кластером через утилиту «jadog\_ctl».

Устанавливается тот же параметр, как и на главном узле (п. 6.7.3.9).

9) Jadog TCP port (user_interface_port)	[54321]
---	---------

```
Jadog 3.1.0 Setup
User / Admin access network settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).
Public Ip address is the address that will be migrated between cluster nodes during failover
for reliable user access. Trusted address is the address to check user network accessibility.
Network interface is the network device name to which public address will be binded.

1) Public address (param_jadog:public_address) [10.116.102.81/24]
2) Public address control interval (param_jadog:public_address_control_interval)[5000]
3) Public address control attempts (param_jadog:public_address_control_attempts)[3]
4) DB monitor timeout (param_postgres:db_check_interval) [5000]
5) Trusted IP address (param_jadog:trusted_address) [ ]
6) RDBMS trusted ip (param_jadog:rdbms_trusted_ip) [ ]
7) RDBMS trusted port (param_jadog:rdbms_trusted_port) [ ]
8) Network interface name (param_jadog:network_interface) [ens33]
9) Jadog TCP port (param_jadog:user_interface_port) [54321]
10) Reset all settings to the default values
> 
```

Рисунок 6.42 – Устанавливаемые параметры в подменю «User / Admin access network setting»

– [Пункт меню 4 «Administrator account list»](#) (п. 6.7.4);

В списке учетных записей администратора(ов) для доступа к компоненту «jaDog» должна быть идентичная учетная запись, установленная на главном узле.

– [Пункт меню 5 «Database server system account and connection «settings»](#) (п. 6.7.5);

В данном пункте меню устанавливаются те же параметры, как на главном узле.

```
Jadog 3.1.0 Setup
Database server system account and connection settings Menu
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Database auth method can take the following values: password, ssl

1) Database host (db_connection_settings:host) [127.0.0.1]
2) Database port (db_connection_settings:port) [5432]
3) Database service name (param_postgres:db_service_name) [jatoba-5]
4) Database name (db_connection_settings:database) [postgres]
5) Jadog password file (db_connection_settings:passfile) [/usr/jatoba-5/bin/db_passfile]
6) Database auth method (db_connection_settings:db_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]
13) Jadog database user password (db_connection_settings:user_pass) HIDDEN VALUE
14) Database server OS user (param_system:system_user) [postgres]
15) Replication node name (param_replication:replication_slot_name) [node2]

16) Reset all settings to the default values

> |
```

Рисунок 6.43 – Устанавливаемые параметры в подменю «Database server system account and connection settings»

Аутентификационные данные роли «jadog\_user» сохраняются для случая, если узел сменит роль резервного узла (Slave) на роль главного узла (Master).

В строке Replication slot name (param\_replication: replication\_slot\_name) должно отображаться уникальное имя слота репликации, т.е. имя слота репликации на основном узле должно отличаться от имен слотов репликации на резервных узлах.

– [Пункт меню 6 «Failover settings»](#) (п. 4.3.3.6);

В данном пункте меню устанавливаются те же параметры, как на главном узле, отвечающие за поведение кластера при обработке отказа сервера с ролью «Master» и при смене дата-центра.

```
Jadog 3.1.0 Setup
-----
Failover settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
1) Auto failover mode [true/false] (param_cluster_behavior:autofailover_mode) [true]
2) Auto dc promote mode [true/false] (param_cluster_behavior:dc_autofailover_mode)[true]
-----
3) Reset all settings to the default values
-----
>
```

Рисунок 6.44 – Параметр Auto dc promote mode [true/false] (dc\_autofailover\_mode) на резервном узле

– [Пункт меню 7\) Replication settings](#) (п. 6.7.7);

В данном пункте меню устанавливаются параметры репликации между узлами кластера и приведены в таблице 6.4.

Таблица 6.4 – Параметры «Replication settings» на резервном узле

Наименование подменю	Тип параметра	Параметр примера
Подменю 1) Replication node name	Уникальный	node2
Подменю 2) Synchronous commit setting	Аналогичный главному узлу	remote_apply
Подменю 3) Synchronous type	Аналогичный главному узлу	ANY
Подменю 4) The number of synchronous standbys	Аналогичный главному узлу	1
Подменю 5) Master sync state after switchover	Аналогичный главному узлу	sync

```
Jadog 3.1.0 Setup
-----
Replication settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
Synchronous commit can take the following values: on, remote_apply, remote_write, local, off.
* "on" - commit transaction after apply (wal will reply) on synchronous nodes. Not guarantee of consistency;
* "remote_apply" - commit transaction after apply (wal will reply) on synchronous nodes. Guarantee of consistency;
* "remote_write" - commit transaction after write WAL into the filesystem on synchronous nodes;
* "local" - allow only local synchronous commit;
* "off" - switch off synchronous commit and synchronous replication.
-----
Synchronous type can take the following values: FIRST, ANY
-----
1) Replication node name (param_replication:replication_slot_name) [node2]
2) Synchronous commit setting (param_synchronous:synchronous_commit) [remote_apply]
3) Synchronous type (param_synchronous:synchronous_type) [ANY]
4) The number of synchronous standbys (param_synchronous:synchronous_commit_max_nodes)[1]
-----
5) Reset all settings to the default values
-----
> |
```

Рисунок 6.45 – Параметры «Replication settings» на резервном узле

– Пункт меню 8 «[WAL archive and restore settings](#)» (п. 6.7.7.5);

В данном пункте меню устанавливаются те же параметры, как на главном узле.



```
Jadog 3.1.0 Setup
-----
WAL archive and restore settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
-----
1) System directory for WAL archive/restore (param_archive:wal_archive_directory) [/nfs/arhive_wal]
2) WAL archive copy command (param_archive:wal_archive_command) [cp %p /nfs/arhive_wal/%f]
3) WAL archive restore command (param_restore:wal_restore_command) [cp /nfs/arhive_wal/%f %p]
4) WAL archive cleanup process on (param_archive:wal_archive_cleanup_needed) [false]
5) WAL archive cleanup command (param_archive:wal_archive_cleanup_command) [/usr/jatoba-5/bin/pg_archivecleanup /nfs/arhive_wal
%s]
6) WAL archive cleanup timeout (param_archive:wal_archive_cleanup_timeout) [30000]
-----
7) Reset all settings to the default values
-----
> |
```

Рисунок 6.46 – Параметры «System directory for WAL archive/restore»

– Пункт меню 9 «Reporting and logging» (п. 6.7.9);

В данном пункте меню устанавливаются те же параметры, как на главном узле.

```
root@node2: /usr/jatoba-6/bin
Jadog 3.3.0 Setup
-----
Reporting and logging Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.
Log format can take the following comma-separated values: txt, csv, json
And special security values: security.txt, security.csv, security.json
Max log file usage duration can take the following units: s, min, h, d
Max log file size can take the following units: B, kB, MB, GB, TB
Log verbosity can take the following values: fatal, error, warning, info, debug1, debug2, debug3
-----
1) Log directory (param_path:log_path) [/usr/jatoba-6/var/log/jadog]
2) Log filename (param_log:logs_filename) [jadog-%a]
3) Log file mode (param_log:logs_file_mode) [0600]
4) Log format (param_log:logs_type) [csv]
5) Max log file usage duration (param_log:logs_rotation_age) [1d]
6) Max log file size (param_log:logs_rotation_size) [10MB]
7) Truncate, not append, logs (param_log:logs_truncate_on_rotation) [false]
8) Log verbosity (param_log:logs_level) [info]
9) Security log directory (param_security_log:security_log_path) [/usr/jatoba-6/var/log/jadog]
10) Security log filename (param_security_log:security_logs_filename) [jadog-%a]
11) Security log file mode (param_security_log:security_logs_filemode) [0600]
12) Allow to write into file (param_log:logs_file) [true]
13) Allow to write into stdout (param_log:logs_screen) [false]
-----
14) Reset all settings to the default values
-----
>
```

Рисунок 6.47 – Параметры пункта меню 9 «Reporting and logging»

– Пункт меню 10 «[Reset all settings to the default values](#)» (п. 6.7.11);

Применяется для сброса всех установленных параметров.

– Пункт меню 11 «[Check and show all settings](#)» (п. 6.7.13);

Применяется для проверки установленных параметров.

– Пункт меню 12 «[Save setting and setup jadog](#)» (п. 6.7.14);

Проверьте корректность внесенных параметров выбрав пункт меню 11 «[Check and show all settings](#)» (п. 6.7.13).

После чего сохраните настройки выбрав пункт меню 12 «Save setting and setup jadog» (п. 6.7.14).

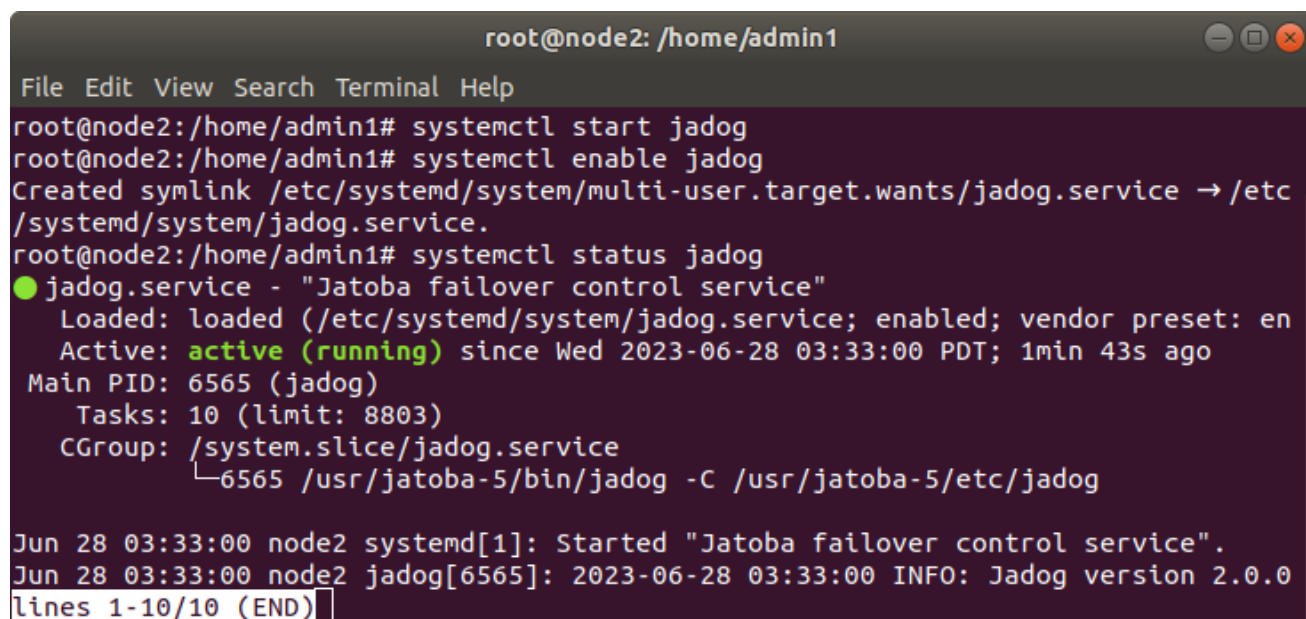
На данном шаге конфигурирование компонента на резервном узле закончено и требуется перейти к запуску серверной части компонента на резервном узле (см. п. 6.8.2).

### 6.8.2. Запуск серверной части компонента «jaDog» на резервном узле

Запуск серверной части компонента «jaDog» выполняется в следующем порядке:

- 1) Запустить сервис «jadog».
- 2) Добавить сервис «jadog» в автозапуск.
- 3) Проверить статус сервиса «jadog».

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



The screenshot shows a terminal window titled 'root@node2: /home/admin1'. The user has entered the following commands: 'systemctl start jadog', 'systemctl enable jadog', and 'systemctl status jadog'. The output of the status command shows that the 'jadog.service' is loaded and active (running). It also displays the main PID (6565), tasks (10), and CGroup. At the bottom, there are two log entries: 'Jun 28 03:33:00 node2 systemd[1]: Started "Jatoba failover control service".' and 'Jun 28 03:33:00 node2 jadog[6565]: 2023-06-28 03:33:00 INFO: Jadog version 2.0.0'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'.

```
root@node2: /home/admin1  
File Edit View Search Terminal Help  
root@node2:/home/admin1# systemctl start jadog  
root@node2:/home/admin1# systemctl enable jadog  
Created symlink /etc/systemd/system/multi-user.target.wants/jadog.service → /etc/  
/systemd/system/jadog.service.  
root@node2:/home/admin1# systemctl status jadog  
● jadog.service - "Jatoba failover control service"  
   Loaded: loaded (/etc/systemd/system/jadog.service; enabled; vendor preset: en  
   Active: active (running) since Wed 2023-06-28 03:33:00 PDT; 1min 43s ago  
   Main PID: 6565 (jadog)  
     Tasks: 10 (limit: 8803)  
    CGroup: /system.slice/jadog.service  
            └─6565 /usr/jatoba-5/bin/jadog -C /usr/jatoba-5/etc/jadog  
  
Jun 28 03:33:00 node2 systemd[1]: Started "Jatoba failover control service".  
Jun 28 03:33:00 node2 jadog[6565]: 2023-06-28 03:33:00 INFO: Jadog version 2.0.0  
lines 1-10/10 (END)
```

Рисунок 6.48 – Статус сервиса jadog

– в ОС Windows:

```
sc start jadogservice  
sc query jadogservice
```

На данном этапе установку резервного узла можно считать законченной.



Установка расширения и создание пользователя не потребуется, т.к. в процессе синхронизации главного и резервного узлов они создадутся автоматически.

## 6.9. Настройка кластера из двух узлов (главный-резервный)

Настройка кластера узлов выполняется с помощью утилиты компонента «jaDog» – «jadowctl».



Большинство команд утилиты «jadowctl» выполняются в фоновом режиме. Если в окне после ввода очередной команды выведено сообщение «Waiting reply from jadow...», это означает, что выполняется процесс. Прерывать процесс не рекомендуется. Для отображения командной строки необходимо нажать клавишу ENTER.



Все имена (пользователь, кластер, узел) вводятся в виде строки в одинарных кавычках

Для настройки кластера из двух узлов (главный-резервный) через утилиту «jadowctl» необходимо выполнить следующие действия.

## 6.10. Подключение к сетевому узлу (connect ip port 'login')

Выбрать сетевой узел, с которого есть доступ к подсети кластера (ко всем узлам кластера), и запустить утилиту «jadowctl» (рисунок 6.49).

```
cd /usr/jatoba-6/bin  
./jadowctl
```

```
root@node1:~# cd /usr/jatoba-5/bin  
root@node1:/usr/jatoba-5/bin# ./jadowctl  
Supported Jadow version 3.1.0  
Terminate parser with \q command  
Use \h for help  
  
Password:
```

Рисунок 6.49 – Запуск утилиты «jadowctl»



Соединение устанавливается с узлом, который будет первоначально играть роль главного в кластере

### 6.11. Создание кластера (add cluster 'name')

Далее необходимо создать кластер (в примере создается кластер с именем «TEST»). Для создания кластера необходимо ввести команду «add cluster» (рисунок 6.50):

```
add cluster 'test'
```

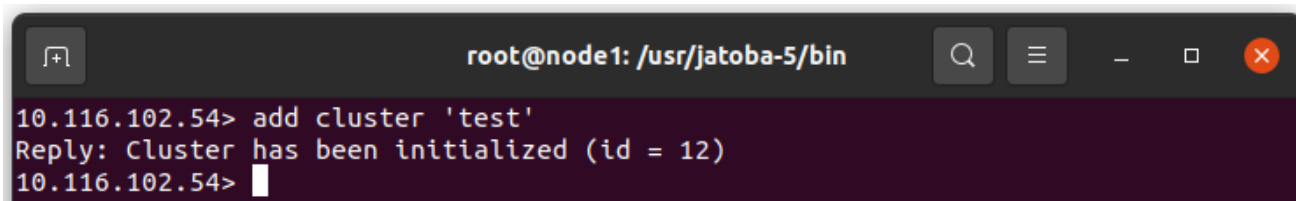


Рисунок 6.50 – Создание кластера

После успешного создания кластера в окне будет выведено сообщение «Cluster has been initialized».

### 6.12. Добавление узла кластера в качестве главного (cluster add master ip)

Далее необходимо добавить узлы кластера, начиная с главного узла (к которому выполнено подключение). Добавление главного узла выполняется с помощью команды «cluster add» имеющей синтаксис:

```
cluster add master [host_ip/host_name/hosts] [Jadog PORT number (port)]
```

В качестве адреса могут использоваться параметры:

- host\_ip – IP-адрес хоста;
- host\_name – имя хоста.

Параметр «Jadog PORT number (port)» описан в п. 6.7.2 настоящего документа.

В рассматриваемом примере используется IP-адрес 10.116.102.54 и команда будет иметь вид:

```
cluster add master 10.116.102.54 12345
```

По окончании успешного добавления узла в приглашении командной строки утилиты «jadog\_ctl» будет выведено сообщение «Node has been successfully initialized» (рисунок 6.51).

При необходимости, для возврата курсора командной строки необходимо продублировать нажатие клавиши ENTER.

```

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

10.116.102.54> 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 (id = 13)
10.116.102.54>

```

Рисунок 6.51 – Добавление главного узла

**i** В случаях, когда главный узел, добавляемый в существующий кластер, ранее уже был в составе другого кластера в качестве главного узла, требуется предварительно выполнить переименование/перенос каталога СУБД /var/lib/jatoba/6/data/. В противном случае при выполнении команды cluster add master [ip] [port] as [node\_name] будет выведено сообщение «Error: Command exit error: Database in recovery state. Please fix it».

### 6.13. Статус узлов кластера (cluster status)

Статус узлов кластера запрашивается следующей командой (рисунок 6.52):

```
cluster status
```

```

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

10.116.102.54> cluster status
Reply: (id = 20)
+-----+
|Jadog version|3.0.0|
+-----+

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

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

+-----+
|Node name |Node    |State   |Connection state|Replication type|Write LSN|Replication slot name|Slot status|Public IP|Node status|
+-----+
|Datacenter: DEFAULT|
+-----+
|test_node1 |10.116.102.54:12345(node1)|Master (ACTIVE)|t       |-       |0/15ACCD8|-       |f       |f       |
+-----+
10.116.102.54>

```

Рисунок 6.52 – Состояние узлов кластера

**i** Отображение вывода команды «cluster status» может потребовать изменения размера шрифта (размер кегля) терминала, чтобы избежать смещения столбцов.



Рекомендуется, предварительно сохранить настройки по умолчанию, создав отдельный профиль.

Таблица 6.5– Параметры таблицы статуса дата-центров

Название параметра	Описание
Datcenter	Имя дата-центра
Status	Состояние дата-центра. Возможные состояния: ACTIVE – активный, EMPTY – пустой, DIED – не активный

Таблица 6.6 – Параметры таблицы статуса узлов

Название параметра	Описание
Datcenter	Имя дата-центра
Node	Сетевой идентификатор узла
State	Состояние узла
Connection state	Состояние сетевого подключения на текущий момент. Возможные состояния: t – узел подключен, f – узел отключен
Replication type	Тип репликации. Возможные состояния: «-» – узел не реплицируется, «ASYNС» – асинхронная репликация
Write LSN	Номер LSN, который записан на диск для этого узла
Replication slot name	Имя слота репликации
Slot status	Состояние слота репликации. Возможные состояния: f – не подключен/отключен, t – активен
Public IP	Публичный IP-адрес. Возможные состояния: f – не подключен/отключен, t – активен
Node status	Текущее состояние узла

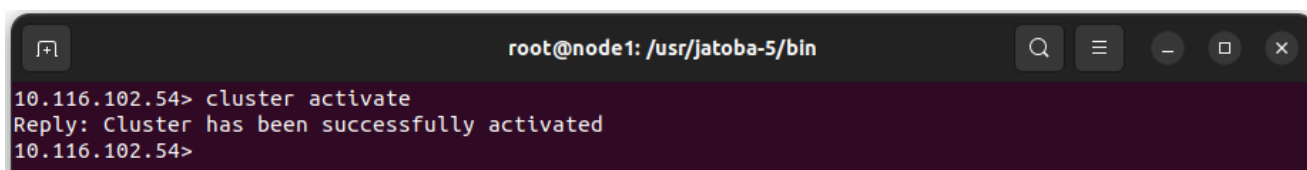
#### 6.14. Присвоение публичного IP-адреса главному узлу кластера (cluster activate)

Присвоение публичного IP-адреса главному узлу кластера выполняется после завершения его формирования и входящих в него узлов.

Присвоение публичного IP-адреса главному узлу кластера выполняется при помощи следующей команды:

```
cluster activate
```

Далее данную операцию делать не требуется.



```
root@node1: /usr/jatoba-5/bin
10.116.102.54> cluster activate
Reply: Cluster has been successfully activated
10.116.102.54>
```

Рисунок 6.53 – Присвоение публичного IP-адреса

Устанавливаемый публичный IP-адрес (Public IP address) описан в п. 6.7.3.

### 6.15. Добавление резервного узла в асинхронном режиме (cluster add slave ip)



Перед выполнением процедуры добавления нового резервного узла в кластер необходимо убедиться в том, что на нем отсутствует каталог `/var/lib/jatoba/6/data` (не выполнена инициализация БД), либо для каталога `data` установлены права доступа `drwx-----` по умолчанию для системного пользователя `postgres`.

```
postgres@node3:~$ chmod 700 /var/lib/jatoba/6/data
```

Далее необходимо добавить резервный узел. Для чего также используется команда «cluster add» с указанием ключевого слова «slave» и соответствующего IP-адреса из списка свободных узлов (рисунок 6.54):

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

В качестве адреса могут использоваться параметры:

- `host_ip` – IP-адрес хоста;
- `port` – номер сетевого порта;
- `node_name` – имя узла в кластере.

Параметр «port» описан в п. 6.7.2 настоящего документа.

В рассматриваемом примере используется IP-адрес сервера с потенциальной ролью «Slave» (10.116.102.55), и команда будет иметь вид:

```
cluster add slave 10.116.102.55 12345 as 'node2'
```

После успешного добавления узла в кластер будет выведено следующее информационное сообщение: «Node was successfully added», в списке узлов кластера появится второй узел – резервный (роль «slave»).

```

root@node1: /usr/jatoba-5/bin
10.116.102.54> cluster add slave 10.116.102.55 12345
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Node was successfully added (id = 3)
10.116.102.54>

```

Рисунок 6.54 – Добавление резервного узла

Аналогично, можно ввести команду «cluster status», чтобы получить актуальное состояние узлов кластера. В соответствии с рисунком 6.55 оба узла активны и в штатных рабочих ролях.

```

root@node1: /usr/jatoba-5/bin
10.116.102.54> cluster status
Reply: (id = 4)
+-----+
|Jadog version|3.0.0|
+-----+

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

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

+-----+
|Node name|Node|State|Connection state|Replication type|Write LSN|Replication slot name|Slot status|Public IP|Node status|
+-----+
|Datacenter: DEFAULT|
+-----+
|test_node1|10.116.102.54:12345(node1)|Master(ACTIVE)|t|-|0/4000180|-|f|t|t|
+-----+
|test_node2|10.116.102.55:12345|Slave(ACTIVE)|t|async|0/4000180|node2|t|f|t|
+-----+
10.116.102.54>

```

Рисунок 6.55 – Контроль статуса узлов кластера

Информацию о статусе добавленного резервного узла можно узнать при помощи консольной утилиты «jadog\_ctl» (см. п.п. 7.22).

## 6.16. Добавление резервного узла в синхронном режиме (cluster add sync slave ip)



Перед выполнением процедуры добавления нового резервного узла в кластер необходимо убедиться в том, что на нем отсутствует каталог /var/lib/jatoba/6/data (не выполнена инициализация БД), либо для каталога data установлены права доступа drwx----- по умолчанию для системного пользователя postgres.

```
postgres@node3:~$ chmod 700 /var/lib/jatoba/6/data
```

Для того чтобы добавить резервный, который будет работать в составе кластера в синхронном режиме, используется команда «cluster add sync» с указанием ключевого слова «slave» и соответствующего IP-адреса/номера сетевого порта из списка свободных узлов (рисунок 6.54):

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

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

В качестве адреса могут использоваться параметры:

- host\_ip – IP-адрес хоста;
- port – номер сетевого порта;
- node\_name – имя узла в кластере.

Параметр «port» описан в п. 6.7.2 настоящего документа.

В рассматриваемом примере используется IP-адрес сервера с потенциальной ролью «Slave» (10.116.102.55), и команда будет иметь вид:

```
cluster add sync slave 10.116.102.55 12345 as 'node2'
```

После успешного добавления резервного узла в синхронном режиме в кластер будет выведено следующее информационное сообщение: «Node was successfully added», в списке узлов кластера появится второй узел – резервный (роль «slave»). Для резервного узла с асинхронным режимом работы.

Информацию о статусе добавленного резервного узла можно узнать при помощи консольной утилиты «jalog\_ctl» (см. п.п. 7.22).

### 6.17. Изменение типа репликации резервного узла

Администратору предоставляется возможность изменять тип репликации резервного узла кластера.

Для примера изменения типа репликации узла будет рассмотрен кластер, состав которого приведен на рисунке 6.55.

Изменение типа репликации возможно в случае если настроен параметр «The number of synchronous standbys (param\_synchronous:synchronous\_commit\_max\_nodes)»



В случае если при изменении типа репликации резервного узла отображается сообщение «Reply: Error: Unable to change replication to sync type in cluster. Please setup synchronous parameters in cluster» необходимо указать количество

резервных узлов, т.е. серверов с ролью «Slave» требующих подтверждения синхронной репликации. Описание данного параметра см. в п.п. 6.7.7.4

После того как резервный узел с IP-адресом 10.116.102.55 добавлен в кластер в асинхронном (см. п.п. 6.15) или синхронном (см. п.п. 6.16) режиме репликации данный параметр возможно изменить следующим образом:

1) Подключиться на главном узле кластера к компоненту «jaDog» при помощи консольной утилиты «jadowctl».

2) Выполнить команду, изменяющую настройки типа репликации резервного узла на **синхронный**:

```
alter node [ip] [port] set replication type = ['sync']
```

**Пример:**

```
alter node 10.116.102.55 12345 set replication type = 'sync'
```

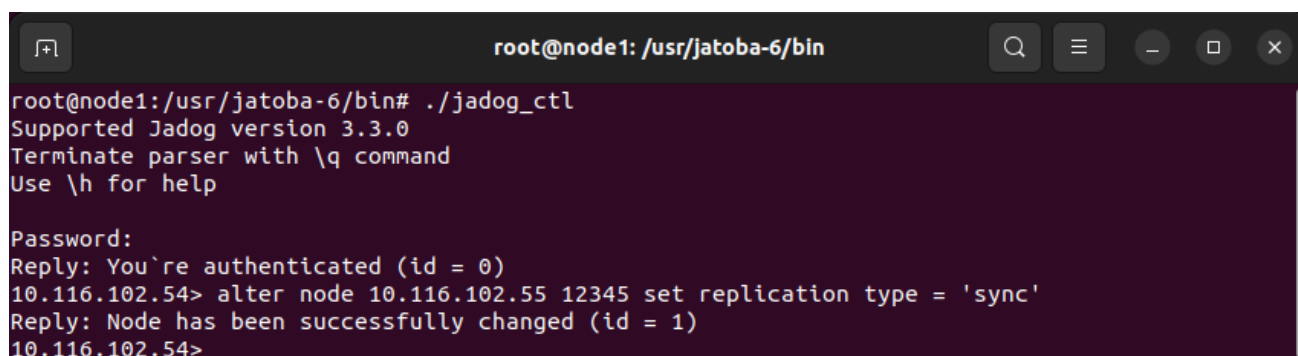


Рисунок 6.56 – Изменение типа репликации на синхронный для резервного узла кластера

3) Или выполнить команду, изменяющую настройки типа репликации резервного узла на **асинхронный**:

```
alter node [ip] [port] set replication type = ['async']
```

**Пример:**

```
alter node 10.116.102.55 12345 set replication type = 'async'
```

```
root@node1: /usr/jatoba-6/bin
root@node1:usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> alter node 10.116.102.55 12345 set replication type = 'sync'
Reply: Node has been successfully changed (id = 1)
10.116.102.54>
```

Рисунок 6.57 – Изменение типа репликации на асинхронный для резервного узла кластера

4) После завершения изменения типа репликации необходимо проверить состояние узлов кластера при помощи команды:

```
cluster status
```



Получение дополнительной информации о типе репликации резервного узла кластера приведено в п.5) и п.6) подраздела 7.22.

Если резервный узел должен входить в кластер с каскадной репликацией в качестве подчиненного каскадного (Cascade Slave), то для изменения его роли необходимо:

1) Подключиться на главном узле кластера к компоненту «jaDog» при помощи консольной утилиты «jadog\_ctl».

2) Выполнить команду:

```
alter node [ip] [port] set primary = ['primary_node']
```

Где primary\_node – название подчиненного резервного узла кластера (Primary Slave).

### Пример:

```
alter node 10.116.102.55 12345 set primary = 'node2'
```

Другим вариантом изменения роли резервного узла на подчиненный каскадный (Cascade Slave) является использование команды:

```
alter node [ip] [port] set primary [primary_ip] [primary_port]
```

Где значения primary\_ip и primary\_port соответствуют IP-адресу и номеру сетевого порта подчиненного резервного узла. (Primary Slave).

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

### Пример:

```
alter node 10.116.102.56 12345 set primary 10.116.102.55 12345
```

3) После завершения внесения изменений типа репликации необходимо перевести кластер в рабочий режим и проверить состояние его узлов при помощи команды:

```
cluster status
```

### 6.18. Автоматическая настройка параметров резервного узла

После добавления резервного узла компонент «jaDog»:

- выполнит репликацию директории данных (data);
- запустит службу jatoba-<ver> на резервном узле;
- скопирует каталог данных, содержащий конфигурационные файлы pg\_hba.conf и postgresql.conf, с главного узла кластера;
- синхронизирует список по УЗ пользователей в файле /usr/jatoba-<ver>/etc/jadog/user.txt.

Внесенные параметры позволят пользователю СУБД «jadog\_user» подключаться к СУБД и выполнять репликацию.

Пример внесенных параметров представлен на рисунке 6.58.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all jadog_user 127.0.0.1/32 md5
host all jadog_user 10.116.102.0/24 md5
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all md5
host replication jadog_user 10.116.102.0/24 md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
```

Рисунок 6.58 – Параметры конфигурационного файла /var/lib/jatoba/5/pg\_hba.conf

### 6.19. Файл состояния кластера jalog\_state.yml

На всем этапе конфигурирования кластера формируется файл состояния кластера jalog\_state.yml, располагающийся по пути:

```
/usr/jatoba-6/etc/jalog/jalog_state.yml
```

В файле jalog\_state.yml отражаются основные параметры состояния кластера и обновляются параметры по мере администрирования кластера. При этом не затрагивается основной файл конфигурации кластера.

Каждый узел кластера имеет свой файл состояния.

```
/usr/jatoba-5/etc/jalog/jalog_state.yml
cluster_nodes:
  Epoch: 0
  Datacenters:
    - Datacenter: DEFAULT
      SyncCount: 0
      nodes:
        - ip: 10.116.102.54
          port: 12345
          ClusterState: Master
          NodeState: ACTIVE
          ReplicationSlot: node1
          NodeName: test_node1
          DBPort: 5432
          SyncState: Async
          PrimaryIP: ""
          PrimaryPort: ""
          PublicIP: 10.116.102.81/24
        - ip: 10.116.102.55
          port: 12345
          ClusterState: Slave
          NodeState: ACTIVE
          ReplicationSlot: node2
          NodeName: test_node2
          DBPort: 5432
          SyncState: Async
          PrimaryIP: 10.116.102.54
          PrimaryPort: 12345
          PublicIP: 10.116.102.81
  Activated: true
  cluster_name: test
```

Рисунок 6.59 – Содержание файла состояния кластера jalog\_state.yml

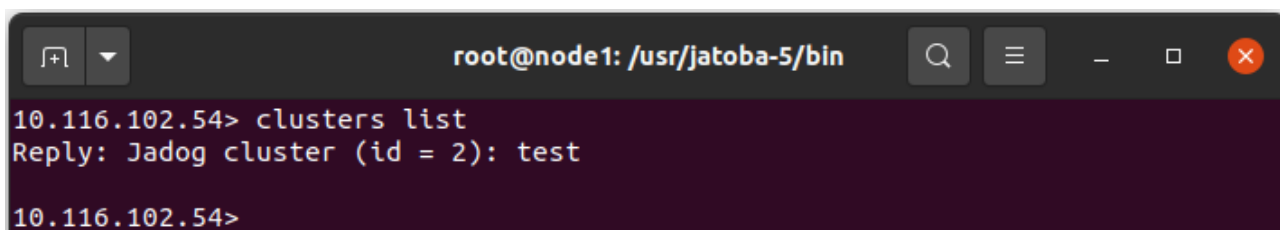


## 7. ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

### 7.1. Просмотр имени кластера (clusters list)

Вывести установленное имя кластера возможно в утилите «jadog\_ctl» при помощи команды:

```
clusters list
```



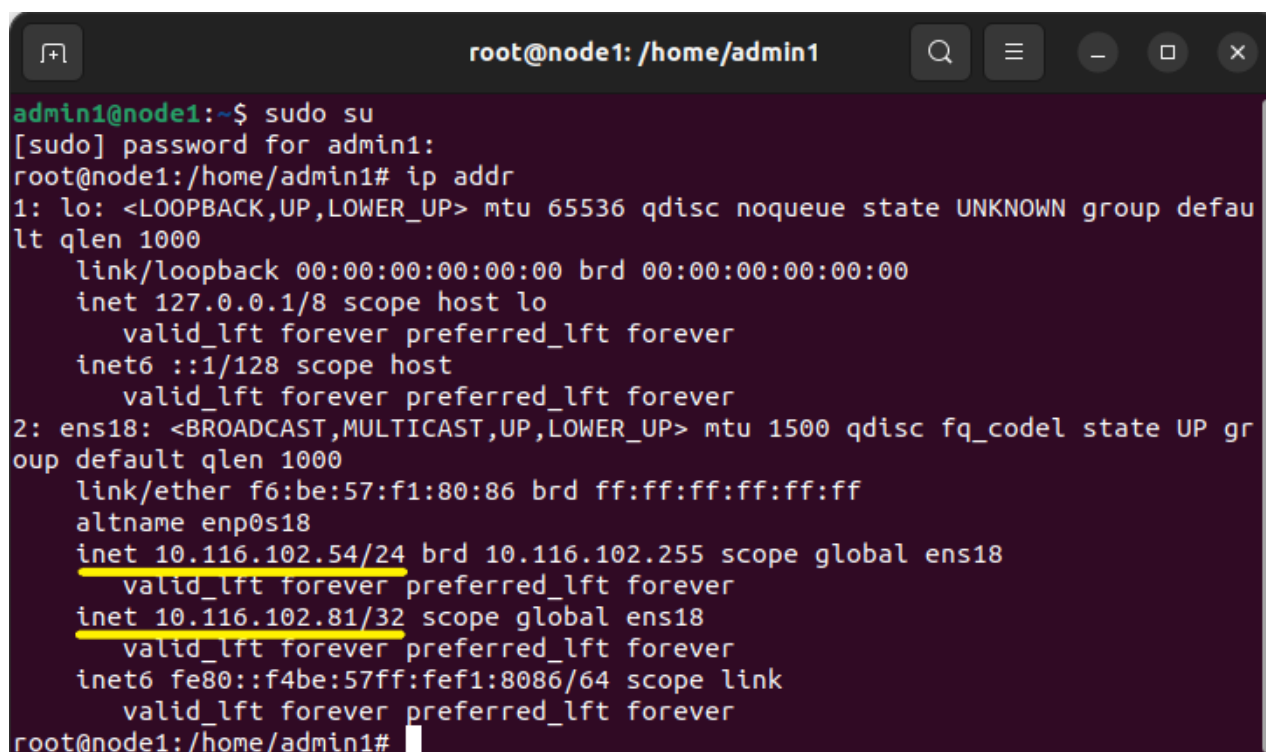
```
root@node1: /usr/jatoba-5/bin
10.116.102.54> clusters list
Reply: Jadog cluster (id = 2): test
10.116.102.54>
```

Рисунок 7.1 – Вывод имени кластера

### 7.2. Деактивация публичного IP-адреса главного узла кластера (cluster deactivate)

При настройке кластера назначался публичный IP-адрес (см. п. 6.14), установленный в настройках основного и резервного узлов (см. п.п. 6.7.3, 6.8.1). Проверка назначенного публичного адреса выполняется командой в терминале ОС.

```
ip addr
```



```
root@node1: /home/admin1
admin1@node1:~$ sudo su
[sudo] password for admin1:
root@node1: /home/admin1# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f6:be:57:f1:80:86 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.116.102.54/24 brd 10.116.102.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet 10.116.102.81/32 scope global ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::f4be:57ff:fef1:8086/64 scope link
        valid_lft forever preferred_lft forever
root@node1: /home/admin1#
```

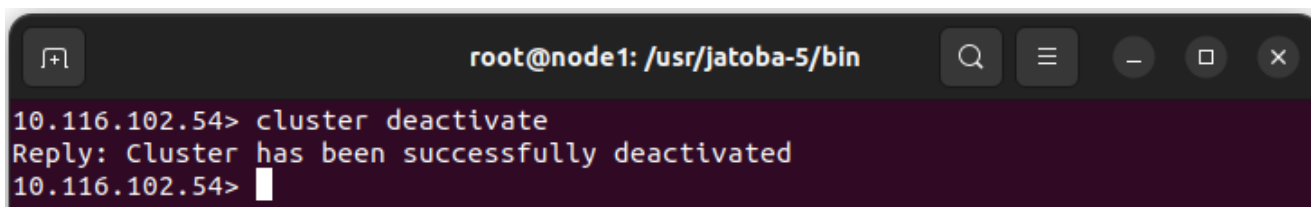
Рисунок 7.2 – Проверка назначенных IP-адресов

На представленном рисунке видно, что назначены два IP-адреса:

- 10.116.102.81 – публичный IP-адрес (Public IP);
- 10.116.102.54 – реальный IP-адрес сетевой карты.

Операция деактивации, т.е. запрета внешнего доступа к кластеру путем удаления «Public IP», выполняется командой в утилите «jadow\_ctl»:

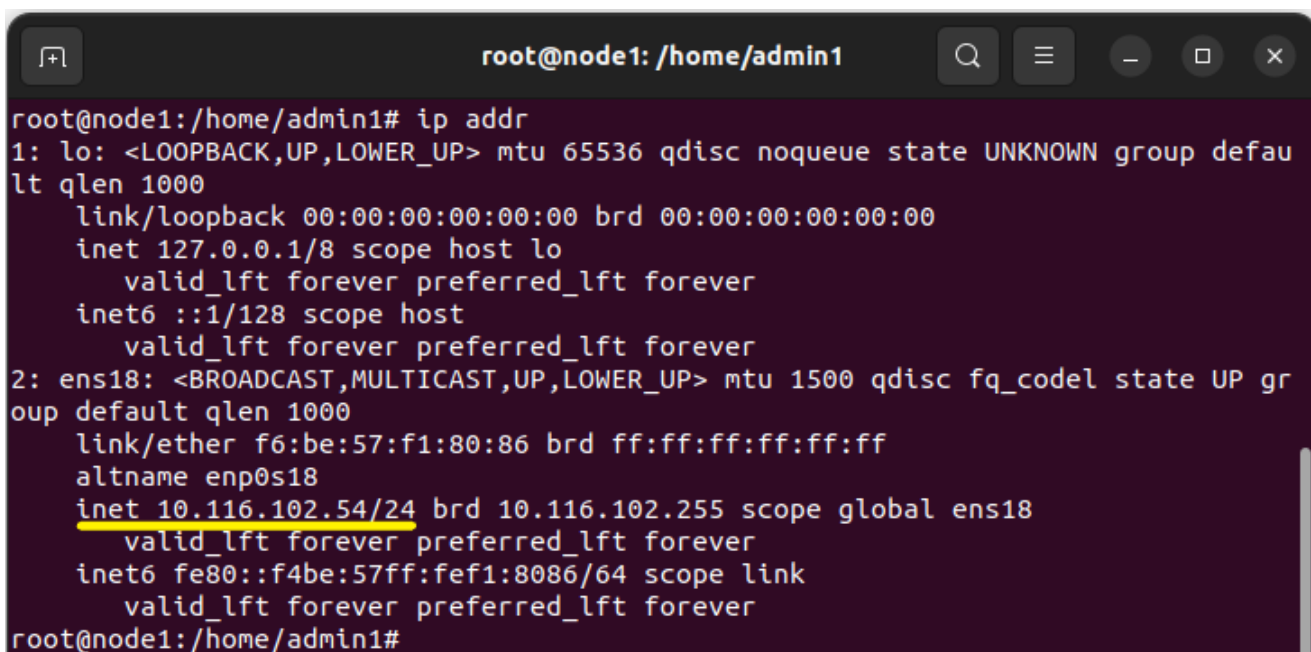
```
cluster deactivate
```



```
root@node1: /usr/jatoba-5/bin
10.116.102.54> cluster deactivate
Reply: Cluster has been successfully deactivated
10.116.102.54>
```

Рисунок 7.3 – Удаление Public IP

Просмотрев назначенные IP-адреса убедимся, что отсутствует публичный IP-адрес – 10.96.1.40.



```
root@node1: /home/admin1
root@node1:/home/admin1# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f6:be:57:f1:80:86 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.116.102.54/24 brd 10.116.102.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::f4be:57ff:fef1:8086/64 scope link
        valid_lft forever preferred_lft forever
root@node1:/home/admin1#
```

Рисунок 7.4 – Проверка назначенных IP-адресов после удаления Public IP

Восстановление внешнего доступа к кластеру выполняется командой как описано в п. 6.14:

```
cluster activate
```

### 7.3. Удаление узла кластера (cluster delete node)

Удаление сетевого узла из кластера с указанным IP - адресом выполняется командой:

```
cluster delete node [host_ip/host_name/hosts] [Jadog PORT  
number (port)]
```

#### Например

Сетевой узел кластера с IP 10.116.102.55, выполняющий роль резервного узла, т.е. «Slave», удаляется командой:

```
cluster delete node 10.116.102.55 12345
```

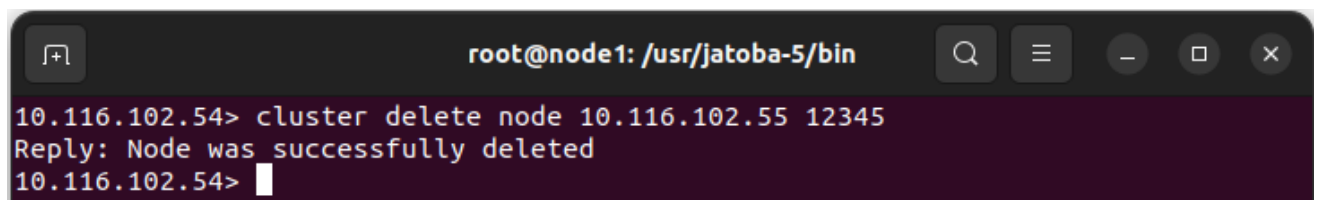


Рисунок 7.5 – Удаление узла кластера

### 7.4. Версия протокола jadog (protocol version)

Вывести информацию о версии протокола jadog в утилите «jadog\_ctl», возможно при помощи команды:

```
protocol version
```

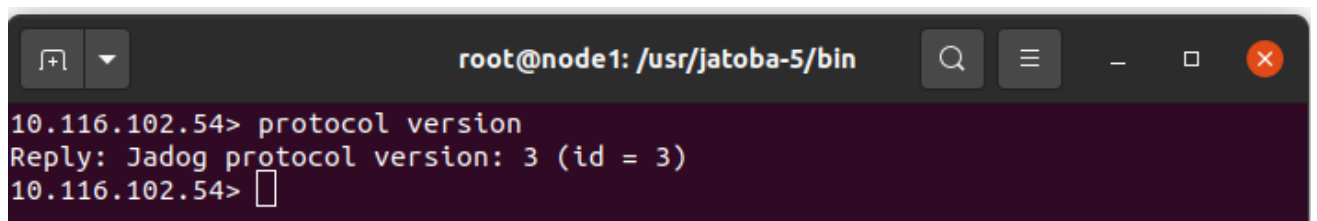


Рисунок 7.6 – Вывод версии протокола jadog

### 7.5. Вывод справочной информации (\h)

Вывод справочной информации утилиты «jadog\_ctl» осуществляется командой:

```
\h
```

```
root@node1: /usr/jatoba-6/bin
10.116.102.54> \h
Reply: Supported commands:
Commands for jaDog:
\h - this help
\password - change current user password.
\q - terminate console.
protocol version - show jadog protocol version.
clear - clear console.
connect {ip} {port} {'login'} - connect to jadog node.
disconnect - disconnect from jadog node.
reload jadog - reload the configuration parameters of this jadog if it is located in the Master-node or it is not in the cluster.

Commands for work with parameters of jadog:
set parameter param_name = param_value - single parameter setting.
set parameters - setting a set of parameters in a transaction.
save - save parameter set transaction.
cancel - cancel parameter set transaction.

set async mode - setup asynchronous command processing.
set sync mode - setup synchronous command processing.
get last response - print last message from server excluding status.
get result id - print last message from server by the given id excluding status.

bundle create {'bundle_name'} - create new cluster bundle.
bundle delete {'bundle_name'} - delete empty bundle.
bundle attach cluster {ip} {port} - attach cluster to bundle with specified ip, port, user.
bundle attach cluster {ip} {port} {'interconnect_user'} - attach cluster to bundle with specified ip, port, interconnect_user.
bundle detach cluster {ip} {port} - detach cluster from bundle with specified ip, port.
bundle detach cluster {cluster_name} - detach cluster from bundle with specified cluster name.
bundle status - show list of clusters in bundle.

Commands for cluster:
add cluster {'cluster_name'} - create new cluster.
delete cluster - delete empty cluster, reverse command to add cluster.
cluster add master {ip} {port} as {'node_name'} - add jadog with specified ip, port and node name as master.
cluster add master {ip} {port} - add jadog with specified ip and port as master.
cluster add cascade slave {ip} {port} primary {primary_ip} {primary_port} as {'node_name'} - add jadog with specified ip, port and node name as cascade slave with primary.
cluster add cascade slave {ip} {port} primary {primary_ip} {primary_port} - add jadog with specified ip and port as cascade slave with primary.
```

Рисунок 7.7 – Вывод справочной информации

## 7.6. Выход из консоли jadog\_ctl (\q)

Выход из утилиты «jadog\_ctl» осуществляется командой:

```
\q
```

## 7.7. Создание дата-центра (datacenter create 'name')

Использование дата-центров описано в разделе «Геораспределенный, отказоустойчивый кластер. Решение JA\_DTC\_AS» второй части документа «Компонент jaDog. Управление режимом работы узлов кластера» 643.72410666.00067-07 98 02-02.

Создать новый дата-центр возможно командой:

```
datacenter create 'name'
```

Например

```
datacenter create 'dc1'
```

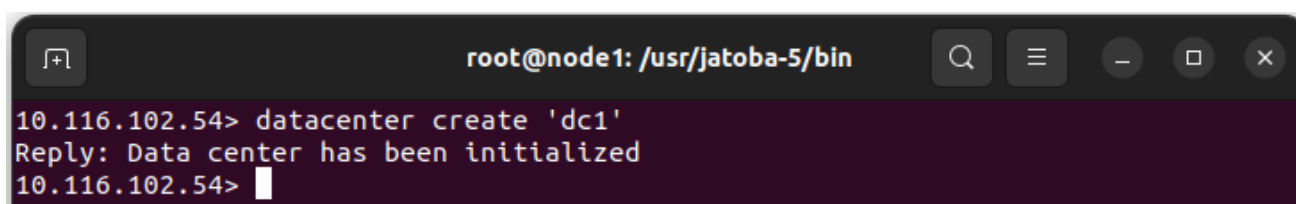


Рисунок 7.8 – Создание дата-центра

### 7.8. Удаление дата-центра (datacenter delete 'name')

Удаление дата-центра выполняется командой со следующим синтаксисом:

```
datacenter delete 'name '
```

#### Например

Удалим ранее созданный дата-центр с именем «dc1», выполнив команду:

```
datacenter delete 'dc1'
```

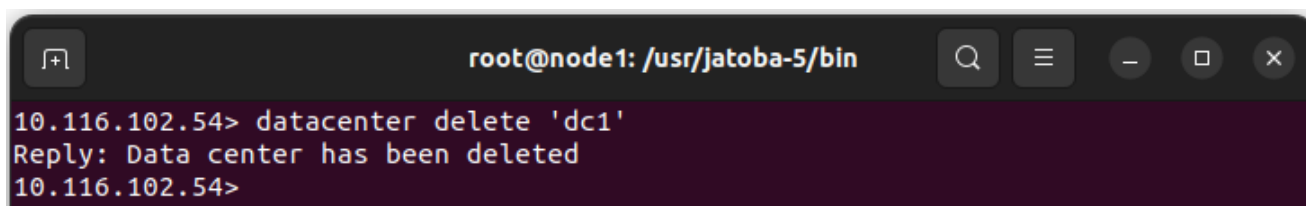


Рисунок 7.9 – Команда удаления дата-центра

### 7.9. Включение узла кластера в дата-центр (datacenter 'name' attach node ip)

Включение узла кластера в дата-центр выполняется командой:

```
datacenter 'name' attach node [host_ip/host_name/hosts] [Jadog  
PORT number (port)]
```

#### Например

При создании кластер по умолчанию привязывается к дата-центру по умолчанию (DEFAULT). Создав новый дата-центр, как описано в п. 7.7, возможно в него включить узел(ы) кластера и таким образом распределить узлы по дата-центрам (площадкам).

Относительно рассматриваемого примера в настоящем документе, включим с созданный дата-центр «dc1» узел кластера Node1 с IP 10.96.1.20, выполнив команду:

```
datacenter 'dc1' attach node 10.116.102.54 12345
```

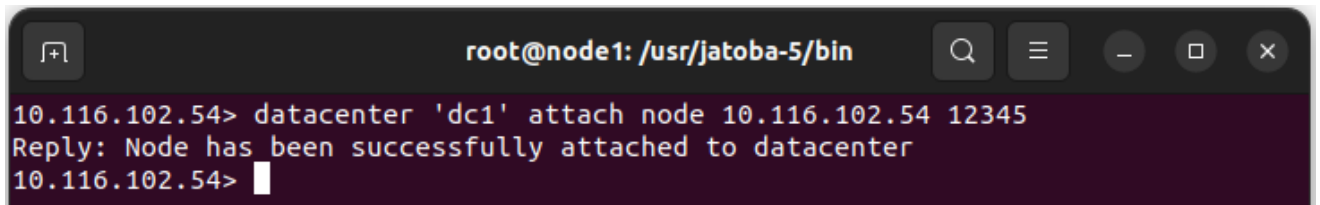


Рисунок 7.10 – Включение узла в дата-центр

Проверить выполнение команды возможно выполнив команду:

```
cluster status
```



Рисунок 7.11 – Просмотр состояния узлов кластера в дата-центре

Из вывода видно, что узел 10.116.102.54 включен в дата-центр «dc1».

### 7.10. Выключение узла кластера из дата-центра (datacenter 'name' detach node ip)

В случае, когда узел кластера включен в какой-либо дата-центр, существует функциональная возможность выключить его из текущего дата-центра. Выключение выполняется командой:

```
datacenter 'name' detach node [host_ip/host_name/hosts] [Jadog
PORT number (port)]
```

#### Например

Ранее включенный узел кластера с IP – 10.116.102.54 в дата-центр «dc1» выключим из него командой:

```
datacenter 'dc1' detach node 10.116.102.54 12345
```



```

root@node1: /usr/jatoba-5/bin
10.116.102.54> datacenter 'dc1' detach node 10.116.102.54 12345
Reply: Node successfully detach from Data Center
10.116.102.54>

```

Рисунок 7.12 – Выключение узла кластера из дата-центра

В результате узел кластера с IP – 10.116.102.54 будет выключен из дата-центра «dc1».

Проверить выполнение команды возможно с помощью команды:

```
cluster status
```

```

root@node1: /usr/jatoba-5/bin
10.116.102.54> cluster status
+-----+-----+
|Datacenter|STATUS|
+-----+-----+
|DEFAULT   |ACTIVE|
+-----+-----+
|dc1       |EMPTY |
+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+
|Datacenter|Node                               |State           |Connection state|Replication type|Write LSN|Replication slot name|Slot status|
+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT   |[10.116.102.54:12345(node1)]|Master(ACTIVE)|t               |-               |[0/9000198]-       |f          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT   |[10.116.102.55:12345()]   |Slave(ACTIVE)|t               |ASync          |[0/9000198]|node2      |t          |
+-----+-----+-----+-----+-----+-----+-----+-----+
10.116.102.54>

```

Рисунок 7.13 – Состояние узлов кластера в дата-центре

Таким образом узел кластера будет включен в дата-центр по умолчанию (DEFAULT).

### 7.11. Принудительное переключение на текущий дата-центр (datacenter promote)

Если главный узел или весь дата-центр, где он расположен, вышел из строя, есть возможность перевести нагрузку в другой дата-центр.

Команду необходимо выполнить на узле, принадлежащему дата-центру, в который требуется перенести нагрузку:

```
datacenter promote
```

### 7.12. Множественные подключения

Утилита «jadog\_ctl» поддерживает множественные подключения. Такой режим требуется, если с кластером работают несколько администраторов и/или используется компонент «Jatoba Data Safe» для управления кластером.

По умолчанию, в конфигурационном файле /usr/jatoba-6/etc/jadog/jadog.yml установлен параметр, максимального количества подключений администраторов кластера:

```
max_admin_connections = 10
```

Это максимальное значение параметра.

Поступающие команды утилита будет обрабатывать в асинхронном режиме, т.е. последовательно, чтобы избежать обрушения кластера.

### 7.13. Управление учетными записями

Для выполнения подключения к компонентам «jaDog» применяются различные типы учетных записей (УЗ):

- Учетная запись пользователя СУБД;
- Технологическая учетная запись пользователя кластера (Interconnect).

#### 7.13.1. Управление учетными записями компонента кластера

Управление учетными записями компонента кластера возможно выполнить:

- при первоначальном конфигурировании кластера через интерфейс сервиса «jadog» (п. 6.7), в пункте меню 4 «Administrator account list» (п. 6.7.4);
- в режиме редактирования параметров кластера, через интерфейс сервиса «jadog» (п. 7.14.1), в пункте меню 4 «Administrator account list» (п. 6.7.4);
- в терминальном режиме утилиты «jadog\_ctl» (п.п. 4.2.2, 7.14.3).

В терминальном режиме утилиты «jadog\_ctl» реализованы следующие функциональные возможности управления учетными записями компонента:

- Создание новой учетной записи без пароля;

Команда имеет синтаксис:

```
create user 'user_name'
```

#### Например

```
create user 'admin4'
```



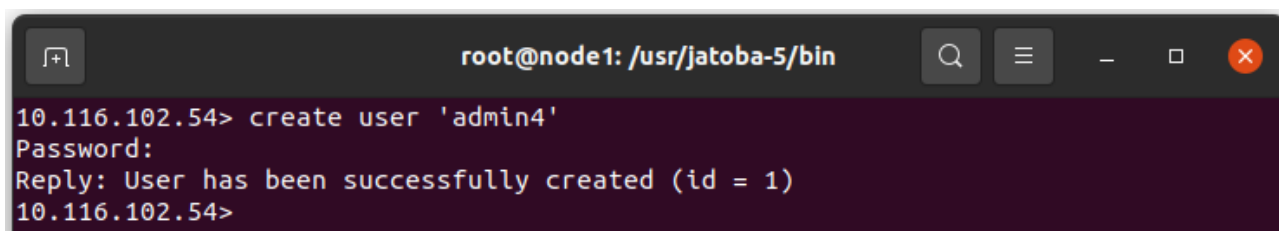


Рисунок 7.14 – Создание новой учетной записи

Создание учетной записи пользователя без ввода пароля необходимо в случае ее использования в качестве технологической (interconnect\_user) для взаимодействия компонентов «jaDog» по TLS соединению (см. п.п 6.7.2.6). В этом случае аутентификация учетной записи без пароля производится на основании доверенных сертификатов SSL.

Создание учетной записи пользователя без ввода пароля также необходимо в случае, если аутентификация любых учетных записей (в том числе администраторов) в компоненте «jaDog» настроена с использованием доверенных сертификатов SSL.

- Создание новой учетной записи с указанием пароля в команде

Команда имеет синтаксис:

```
create user 'user_name' with password 'password'
```

**Например**

```
create user 'admin5' with password 'password'
```

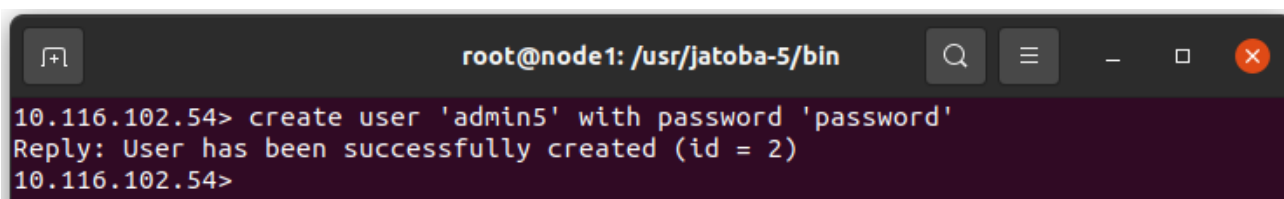


Рисунок 7.15 – Создание новой учетной записи с указанием пароля в команде

- Смена пароля учетной записи;

Команда имеет синтаксис:

```
alter user 'user_name' with password 'new_password'
```

**Например**

```
alter user 'admin4' with password 'new_password'
```

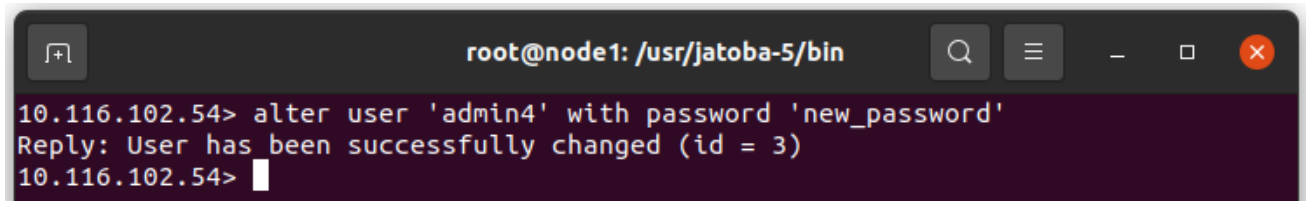


Рисунок 7.16 – Смена пароля учетной записи

- Переименование учетной записи

Команда имеет синтаксис:

```
alter user 'user_name' rename to 'new name'
```

### Например

```
alter user 'admin4' rename to 'test_admin'
```

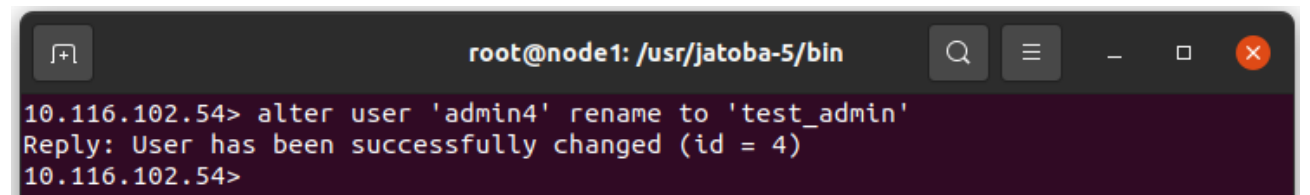


Рисунок 7.17 – Переименование учетной записи

- Удаление учетной записи;

Команда имеет синтаксис:

```
drop user 'name'
```

### Например

```
drop user 'admin5'
```

Удаление учетной записи и последующее создание ее без пароля (см. данный подраздел) требуется в случае, если компонент «jaDog» настраивается на TLS аутентификацию учетных записей при помощи сертификатов SSL.

```
root@node1: /usr/jatoba-5/bin
10.116.102.54> show users
Reply: (id = 6)
+-----+
|User      |blocked|
+-----+
|admin     |f      |
+-----+
|admin1    |f      |
+-----+
|admin2    |f      |
+-----+
|admin5    |f      |
+-----+
|test_admin|f      |
+-----+

10.116.102.54> drop user 'admin5'
Reply: User has been successfully dropped (id = 7)
10.116.102.54> show users
Reply: (id = 8)
+-----+
|User      |blocked|
+-----+
|admin     |f      |
+-----+
|admin1    |f      |
+-----+
|admin2    |f      |
+-----+
|test_admin|f      |
+-----+

10.116.102.54> 
```

Рисунок 7.18 – Удаление учетной записи

После выполнения команды, УЗ отсутствует в списке пользователей утилиты «jadog\_ctl».

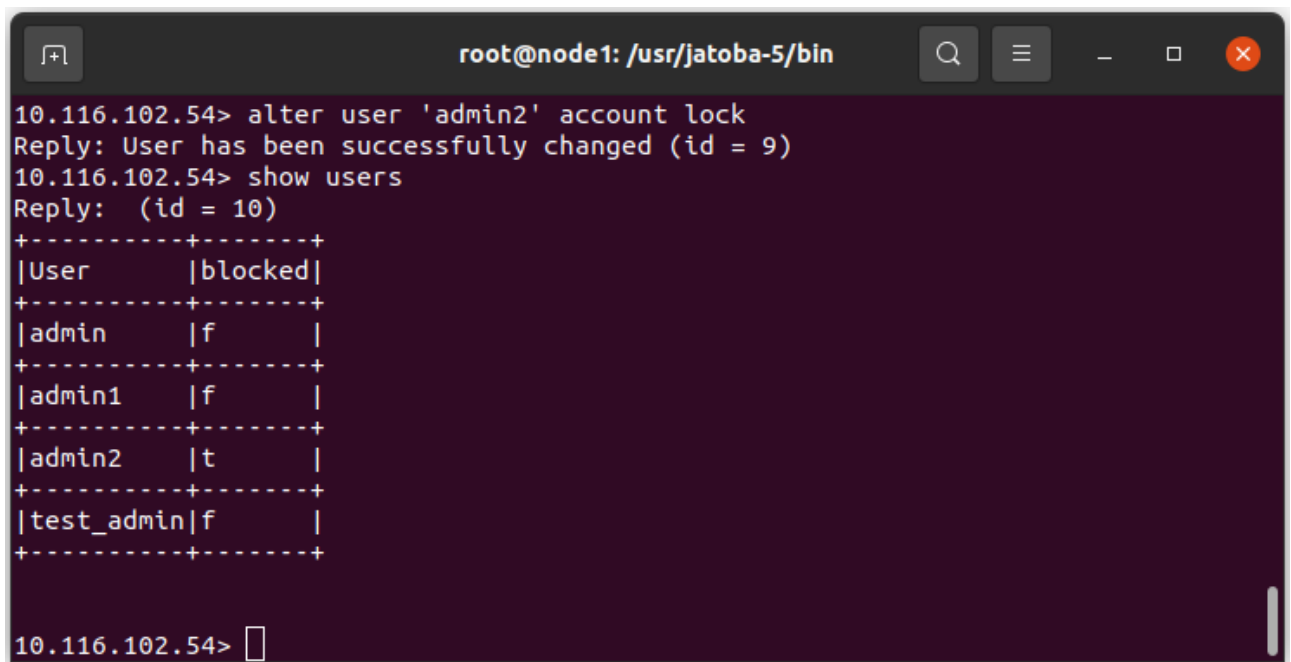
- Блокирование учетной записи;

Команда имеет синтаксис:

```
alter user 'user_name' account lock
```

**Например**

```
alter user 'admin2' account lock
```



```
root@node1: /usr/jatoba-5/bin
10.116.102.54> alter user 'admin2' account lock
Reply: User has been successfully changed (id = 9)
10.116.102.54> show users
Reply: (id = 10)
+-----+-----+
|User      |blocked|
+-----+-----+
|admin     |f      |
+-----+-----+
|admin1    |f      |
+-----+-----+
|admin2    |t      |
+-----+-----+
|test_admin|f      |
+-----+-----+
10.116.102.54> 
```

Рисунок 7.19 – Блокирование учетной записи

При выводе списка пользователей у заблокированного пользователя в поле «blocked» появится значение «t» (true).

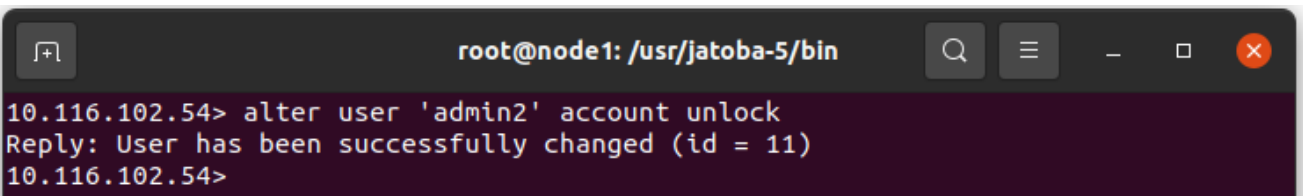
- Разблокирование учетной записи;

Команда имеет синтаксис:

```
alter user 'user_name' account unlock
```

**Например**

```
alter user 'admin2' account unlock
```



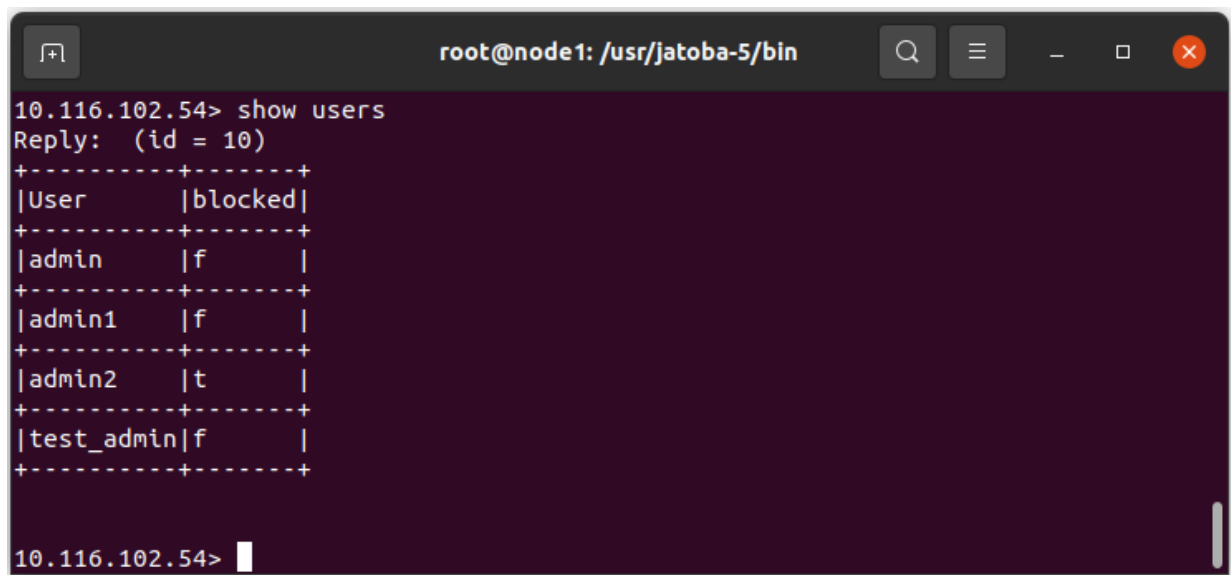
```
root@node1: /usr/jatoba-5/bin
10.116.102.54> alter user 'admin2' account unlock
Reply: User has been successfully changed (id = 11)
10.116.102.54> 
```

Рисунок 7.20 – Разблокирование учетной записи

- Просмотр учетных записей в кластере.

Команда имеет синтаксис:

```
show users
```



```
10.116.102.54> show users
Reply: (id = 10)
+-----+-----+
|User      |blocked|
+-----+-----+
|admin     |f      |
+-----+-----+
|admin1    |f      |
+-----+-----+
|admin2    |t      |
+-----+-----+
|test_admin|f      |
+-----+-----+

10.116.102.54> 
```

Рисунок 7.21 – Просмотр учетных записей в кластере

Все изменения в УЗ пользователей кластера будут синхронизированы автоматически, т.е. внесенные изменения на одном узле отобразятся на всех узлах кластера.

### 7.13.2. Управление учетными записями компонента для СУБД

Управление учетными записями компонента для СУБД осуществляется через утилиту «psql».

В терминальном режиме утилиты «psql» реализованы следующие функциональные возможности управления учетными записями СУБД:

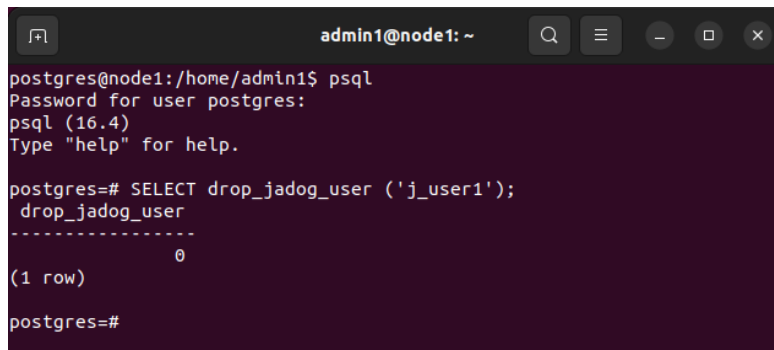
- Удаление пользователя специальной роли «jalog\_user» (см. п.п. 2.1.5)

Команда имеет следующий синтаксис:

```
SELECT drop_jalog_user ('jalog_user');
```

**Например:**

```
SELECT drop_jalog_user ('admin2');
```



```
admin1@node1: ~  
postgres@node1:/home/admin1$ psql  
Password for user postgres:  
psql (16.4)  
Type "help" for help.  
  
postgres=# SELECT drop_jadog_user ('j_user1');  
drop_jadog_user  
-----  
0  
(1 row)  
  
postgres=#
```

Рисунок 7.22 – Удаление пользователя специальной роли «jadog\_user»

- Удаление роли пользователей для взаимодействия с Citus/Hipe Cluster;

Команда имеет следующий синтаксис:

```
SELECT drop_hipe_role ('jadog_user');
```

**Например:**

```
SELECT drop_hipe_role ('admin2');
```

- Добавление групповой роли «jadog\_hipe\_role» для указанного пользователя (пользователь должен быть заранее создан);

Команда имеет следующий синтаксис:

```
SELECT grant_hipe_role_to_jadog_user ('jadog_user');
```

**Например:**

```
SELECT grant_hipe_role_to_jadog_user ('admin2');
```

- Удаление групповой роли «jadog\_hipe\_role» для указанного пользователя (пользователь должен быть заранее создан).

Команда имеет следующий синтаксис:

```
SELECT drop_hipe_role ('jadog_user');
```

**Например:**

```
SELECT drop_hipe_role ('admin2');
```

Пользователя компонента «jaDog» можно удалить с использованием команды:

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

```
DROP ROLE ('role');
```

если он был создан через функцию «grant\_jadog\_role\_to\_jadog\_user».

– Добавление групповой роли «jadog\_repl\_acc» для пользователя (пользователь должен быть заранее создан).

Команда имеет следующий синтаксис:

```
SELECT grant_jadog_role_to_jadog_user ('jadog_user');
```

**Например:**

```
SELECT grant_jadog_role_to_jadog_user ('admin2');
```

Групповая роль «jadog\_repl\_acc» удаляется через использование функции «drop\_jadog\_user» (см. выше).

#### **7.14. Редактирование и получение параметров кластера «jaDog»**

Редактирование и получение параметров кластера возможно выполнить через:

- режим «setup» сервиса «jadog» (п. 7.14.1);
- интерфейс утилиты «jadog\_ctl» (п. 7.14.2);
- терминальный режим утилиты «jadog\_ctl» (п. 7.14.3);

Режим «setup» сервиса «jadog» позволяет изменять ряд основных параметров кластера.

Утилита «jadog\_ctl» позволяет просматривать и редактировать параметры кластера. Описание части команд приведено ниже.

Примеры команд изменения параметров приведены в Приложении 1.

##### **7.14.1. Редактирование параметров через интерфейс сервиса «jadog»**

Для перехода в режим «setup» редактирования конфигурации компонента «jaDog» в GNU Linux необходимо выполнить команды:

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

В ОС Windows необходимо выполнить следующие действия:

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

- 1) Открыть командную строку ОС Windows (cmd.exe).
- 2) Перейти в директорию «C:\Program Files\GIS\Jatoba\6\bin» и запустить исполняемый файл «jadog» с параметром «setup» (далее – jadog setup) для перехода в режим редактирования конфигурации компонента «jaDog».

```
cd "C:\Program Files\GIS\Jatoba\6\bin"  
jadog setup -C "C:\Program Files\GIS\Jatoba\6\etc\jadog"
```

После выполнения команд откроется интерфейс сервиса «jadog».

#### 7.14.2. Редактирование и получение параметров через утилиту «jadog\_ctl»

Редактирование параметров через утилиту «jadog\_ctl» возможно через вход в режим редактирования командой:

```
set parameters
```

Команда может использоваться как для входа в режим транзакционного редактирования параметров (см. п.п. 7.14.2.2), так и в составе команды на редактирование одного параметра (см. п.п. 7.14.2.1).



В утилите «jadog\_ctl» реализован механизм автодополнения.

Чтобы воспользоваться этим сочетанием, нужно начать вводить первые буквы команды и нажать ТАВ. Строка будет автодополнена. Это делает набор команд и путей к файлам намного быстрее.



Некоторые редактируемые параметры требуют выполнения перезагрузки компонентов кластера или всего кластера. В этом случае после изменения параметра администратору выводится предупреждение.

«Параметр установлен. Требуется перезагрузка сервиса jadog»

Перезагрузка компонентов осуществляется Администратором кластера самостоятельно вне рамок данной сессии.

Таблица П.1.1 содержит список параметров и информацию о необходимости выполнения перезагрузки компонента.



При изменении некоторых параметров сообщение пользователю может отличаться от выводимого по умолчанию. В этом случае в таблице П.1.1 для изменяемого параметров приводится текст выводимого сообщения.

Для параметров компонента, которые не требуют перезагрузки выводится сообщение пользователю после изменения параметра:

«Параметр установлен. Перезагрузка не требуется.»

#### 7.14.2.1 Редактирование единственного параметра

Изменение параметра «trusted\_address» (доверенного IP-адреса) доступно после аутентификации в утилите «jadowctl».

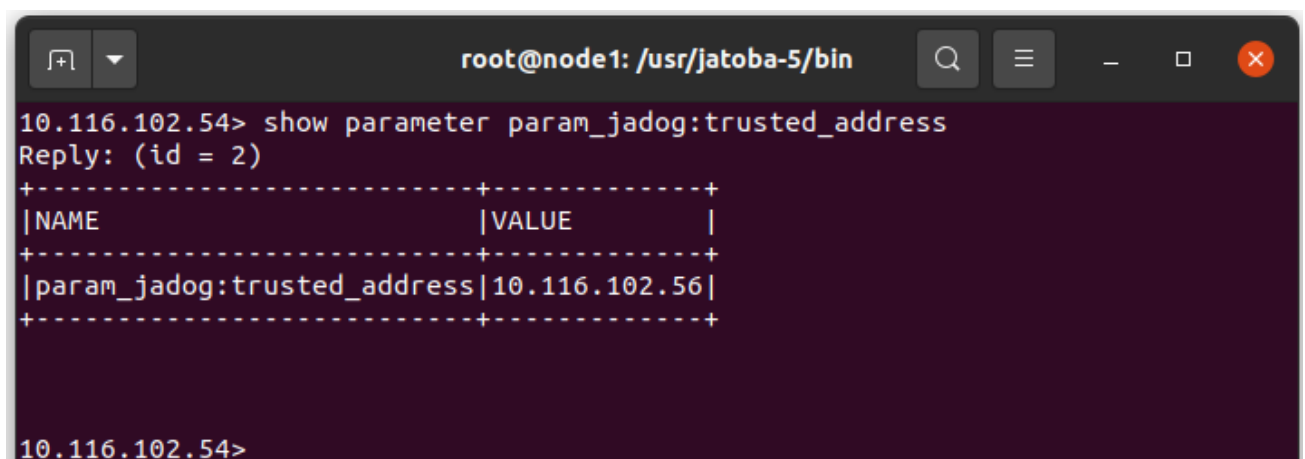
Команда изменения единственного параметра имеет синтаксис:

```
set parameter <parameter category>:<parameter> = 'the value of the parameter'
```

#### Например

Установим доверенный IP-адрес 10.116.102.56/24 командой в утилите «jadowctl»:

```
set parameter param_jadow:trusted_address = '10.116.102.56'
```

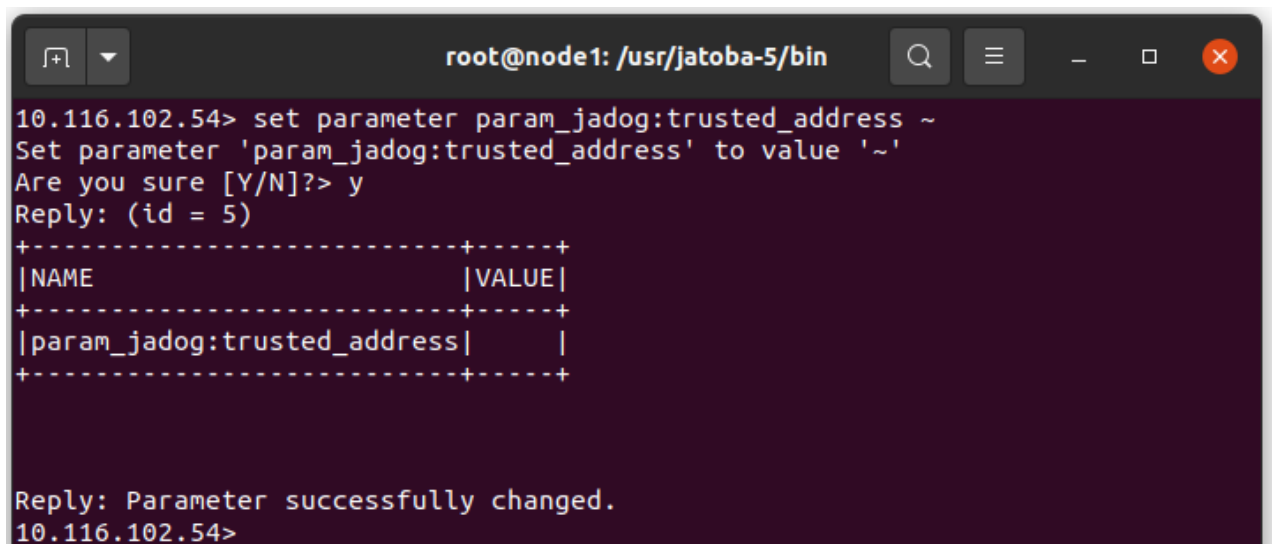


```
root@node1: /usr/jatoba-5/bin
10.116.102.54> show parameter param_jadow:trusted_address
Reply: (id = 2)
+-----+-----+
|NAME                |VALUE                |
+-----+-----+
|param_jadow:trusted_address|10.116.102.56|
+-----+-----+
10.116.102.54>
```

Рисунок 7.23 – Изменение доверенного адреса

Возврат к значению по умолчанию выполняется командой (знак тильда «~» в качестве принимаемого значения):

```
set parameter param_jadow:trusted_address ~
```



```
root@node1: /usr/jatoba-5/bin
10.116.102.54> set parameter param_jadog:trusted_address ~
Set parameter 'param_jadog:trusted_address' to value '~'
Are you sure [Y/N]?> y
Reply: (id = 5)
+-----+-----+
|NAME                |VALUE|
+-----+-----+
|param_jadog:trusted_address|    |
+-----+-----+

Reply: Parameter successfully changed.
10.116.102.54>
```

Рисунок 7.24 – Сброс значения параметра до значения по умолчанию



Изменение параметра не требует перезагрузки кластера

#### 7.14.2.2 Редактирование группы параметров (транзакционный метод)

Для редактирования группы параметров требуется:

- Войти в режим транзакционного редактирования параметров:

```
set parameters
```

- Изменить первый параметр. Например «public ip»:

```
set parameter param_jadog:public_address = '10.116.102.100'
```

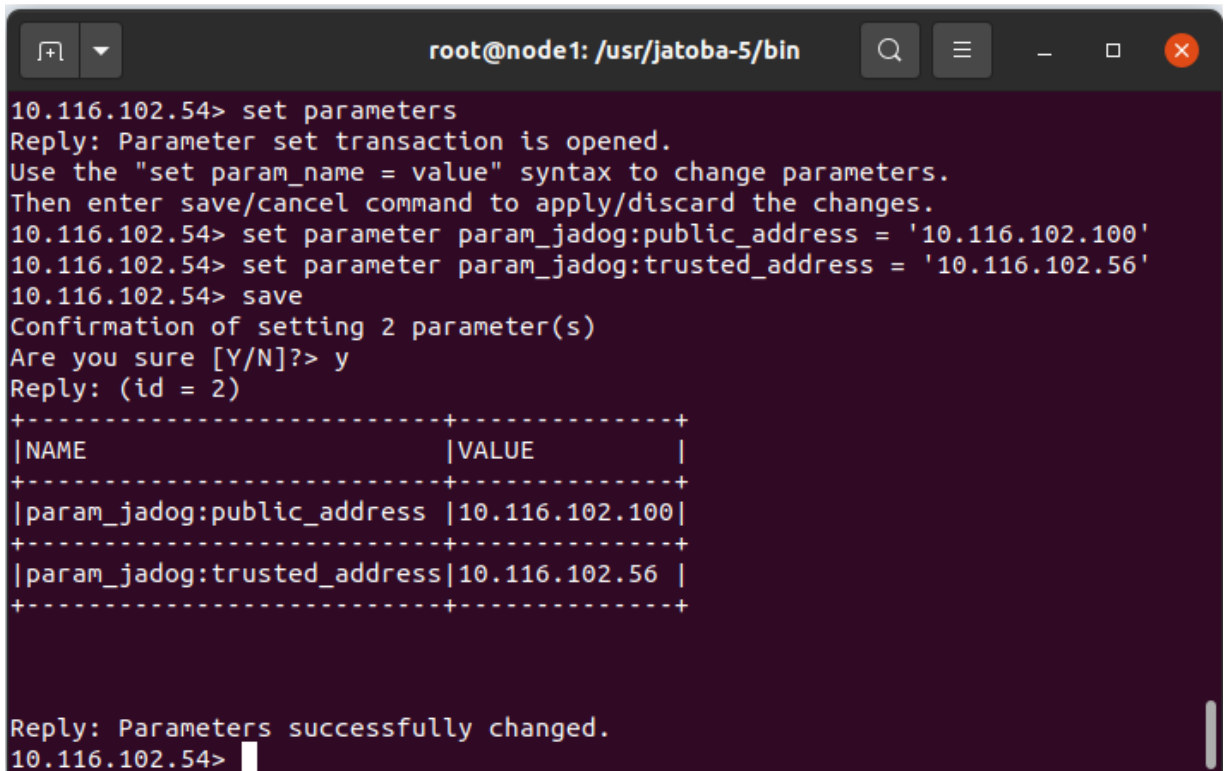
- Изменить второй параметр. Например «trusted address»:

```
set parameter param_jadog:trusted_address = '10.116.102.56'
```

- Сохранить изменения командой:

```
save
```

- Подтвердить сохранение изменений.



```
root@node1: /usr/jatoba-5/bin
10.116.102.54> set parameters
Reply: Parameter set transaction is opened.
Use the "set param_name = value" syntax to change parameters.
Then enter save/cancel command to apply/discard the changes.
10.116.102.54> set parameter param_jadog:public_address = '10.116.102.100'
10.116.102.54> set parameter param_jadog:trusted_address = '10.116.102.56'
10.116.102.54> save
Confirmation of setting 2 parameter(s)
Are you sure [Y/N]?> y
Reply: (id = 2)
+-----+-----+
|NAME                                |VALUE                                |
+-----+-----+
|param_jadog:public_address |10.116.102.100|
+-----+-----+
|param_jadog:trusted_address|10.116.102.56 |
+-----+-----+

Reply: Parameters successfully changed.
10.116.102.54> 
```

Рисунок 7.25 – Редактирование группы параметров

### 7.14.2.3 Просмотр единственного параметра

Просмотр имени кластера возможен через утилиту «jadog\_ctl» командой, имеющей синтаксис:

```
show parameter <parameter category>:<parameter>
```

Например

Просмотр установленного доверенного IP-адреса выполняется командой:

```
show parameter param_jadog:trusted_address
```

```

root@node1: /usr/jatoba-5/bin
10.116.102.54> show parameter param_jadog:trusted_address
Reply: (id = 7)
+-----+-----+
|NAME                |VALUE|
+-----+-----+
|param_jadog:trusted_address|    |
+-----+-----+

10.116.102.54>

```

Рисунок 7.26 – Просмотр доверенного IP адреса

#### 7.14.2.4 Просмотр всех параметров кластера (show parameter all)

Просмотр параметров кластера возможен через утилиту «jadog\_ctl» командой:

```
show parameter all
```

```

root@node1: /usr/jatoba-5/bin
10.116.102.54> show parameter all
Reply: (id = 2)
+-----+-----+
|NAME                |VALUE|
+-----+-----+
|param_path:state_cfg_path|/usr/jatoba-5/etc/jadog|
+-----+-----+
|param_path:module_path  |/usr/jatoba-5/share/jadog/scripts|
+-----+-----+

```

Рисунок 7.27 – Вывод установленных параметров кластера

В результате будет выведен список всех параметров кластера, приведенных в Приложении 1 настоящего документа.

#### 7.14.3. Редактирование и получение параметров в терминальном режиме через утилиту «jadog\_ctl»

Синтаксис подключения к утилите аналогичен синтаксису подключения к утилите «psql» СУБД.

В режиме транзакционного (то есть накопительного) редактирования не указываемые параметры остаются в значениях по умолчанию.

#### Например

Команда изменения доверенного IP-адреса будет иметь вид:

```
./jadog_ctl -h localhost -p 54321 -U admin -W test -C "set
parameters set parameter param_jadog:trusted_address =
'192.168.31.4' save"
```

Команда изменения доверенного IP-адреса и публичного IP-адреса будет иметь вид:

```
./jadog_ctl -h localhost -p 54321 -U admin -W test -C "set
parameters set parameter param_jadog:trusted_address =
'192.168.31.5' set parameter param_jadog:public_address =
'192.168.31.201' save"
```

### 7.14.3.1 Просмотр параметров кластера (./jadog\_ctl -C "show parameter all")

Просмотр параметров кластера в терминальном режиме позволяет вывести установленные параметры без аутентификации в утилите «jadog\_ctl».

Вывод параметров кластера выполняется в терминале ОС командой:

```
./jadog_ctl -C "show parameter all"
```



Рисунок 7.28 – Вывод параметров в терминальном режиме



Синтаксис команды позволяет использовать как одинарные, так и двойные кавычки

### 7.14.3.2 Вывод параметров и команд в формате JSON

Результаты выполнения команд в терминальном режиме, с ключами «-C», «-f» возможно вывести в формате JSON.

**Например**

```
./jadog_ctl -h localhost -p 54321 -U admin -W admin -C "cluster
status" -q -f json
```

```
root@node1:/usr/jatoba-5/bin# ./jadog_ctl -h localhost -p 54321 -U admin -W admin -C "cluster status" -q -f json
{
  "JadogVersion": "3.1.0",
  "connectionFrom": "127.0.0.1",
  "errCode": 2,
  "id": 0,
  "message": "Wrong authentication password for user: admin",
  "srcip": "10.116.102.54",
  "srcport": "12345",
  "state": "false"
}
root@node1:/usr/jatoba-5/bin#
```

Рисунок 7.29 – Вывод в формате JSON

## 7.15. Процедура принудительной смены ролей серверов (switchover)



Операция принудительной смены ролей серверов (switchover) всегда производится при подключении только к главному узлу

Для выполнения операции «switchover» необходимо выполнить следующие действия.

1) Подключиться к главному узлу через утилиту «jdogctl».

2) В списке серверов определить резервный узел, на который необходимо переключить роль главного (master), командой:

```
cluster status
```

```
10.116.102.54> cluster status
+-----+
|Jadog version|2.2.0|
+-----+

+-----+
|Cluster name|AutoFailover|AutoDCFailover|Public IP  |
+-----+
|test       |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        |            |
+-----+

10.116.102.54>
```

Рисунок 7.30 – Подключение к главному узлу через утилиту «jdogctl»

В представленном примере с ролью «Slave» узел с IP – 10.116.102.55

3) После этого запустить команду «switchover» имеющую синтаксис:

```
switchover [host_ip/host_name/hosts] [Jadog PORT number (port)]
```

Например:

```
switchover 10.116.102.55 12345
```

```

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

10.116.102.54> switchover 10.116.102.55 12345
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: SWITCHOVER has successfully completed
10.116.102.54>

```

Рисунок 7.31 – Выполнение команды «switchover»

В результате узлы кластера поменяются ролями.

```

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

10.116.102.54> switchover 10.116.102.55 12345
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: Waiting reply from jadog. Check jadog node logs for details.
Reply: SWITCHOVER has successfully completed
10.116.102.54> cluster status

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

+-----+-----+-----+-----+
|Cluster name|AutoFailover|AutoDCFailover|Public IP|
+-----+-----+-----+-----+
|test        |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)|Slave(ACTIVE) |t              |-              |0/40090A8|jadog_slot         |f          |f        |t          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|DEFAULT   |10.116.102.55:12345()    |Master(ACTIVE)|t              |-              |0/3000198|-                 |t          |t        |t          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

10.116.102.54>

```

Рисунок 7.32 – Проверка ролей узлов в кластере

По журналам событий компонента «jaDog» на узлах можно наблюдать порядок выполняемых действий.

Например, на основном сервере будет сообщение подобное следующему:

```
2023-11-25 13:27:56 WARNING: UISwitchover command was successfully completed
```

Процедуру принудительной смены ролей узлов кластера допустимо проводить только с узла с ролью «Master».

## 7.16. Режим технического обслуживания (maintenance)

Режим технического обслуживания (ТО) может быть установлен для всего кластера или его отдельных составных узлов.

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

В режиме ТО кластер или отдельный его узел принимает команды и выполняет все необходимые действия администратора, но автоматические смены состояний кластера не отрабатывает (FO).

Установка режима ТО не влияет на синхронизацию данных согласно установленных правил и режимов работы кластера.

Управление режимом ТО кластера и/или его отдельных узлов осуществляется через интерфейс утилиты «`jadog_ctl`» путем ввода команд.

Основные принципы поведения узлов кластера в режиме ТО:

- В случае если режим ТО устанавливается для узла кластера с ролью «Master» это автоматически приводит к вводу в этот режим и всех узлов кластера с ролями «Slave» (т.е. на каждый узел кластера принудительно отправляется команда установки в режим ТО);
- Если режим ТО установлен на всех узлах кластера кроме узла с ролью «Master», то это автоматически переводит весь кластер в данный режим;
- Перезагрузка узла кластера, находящегося в режиме ТО, не является аварийной;
- Включение режима ТО производится только через главный узел кластера с ролью «Master»;
- В режиме ТО администратору доступны все параметры настройки как кластера, так и отдельного его узла кластера;
- После отключения режима ТО отдельный узел или весь кластер работают в активном режиме с учетом текущего статуса и состояния всего кластера.
- Режимы синхронной/асинхронной репликации данных влияния на ТО и поведение кластера в режиме ТО не производят.

Для получения информации о статусе кластера или входящих в него узлов применяется команда:

```
cluster status
```

Узлы, находящиеся в режиме ТО, будут иметь статус MAINTENANCE.



В качестве примера использования режима ТО: если администратору кластера необходимо выполнить перезагрузку узла с ролью «Master» без перехода этой роли на другой узел (без выполнения процедур FAILOVER) он получает возможность установить для кластера или узла с ролью «Master» режим ТО, и выполнить перезагрузку кластера.

#### 7.16.1. Включение режима технического обслуживания на узле

Включение режима ТО осуществляется командой:

```
set maintenance on node ['node_name']
```

Или с указанием IP-адреса и номера порта узла:

```
set maintenance on node [ip] [port]
```

#### 7.16.2. Включение режима технического обслуживания кластера

Осуществляется командой:

```
set maintenance
```

```
10.116.102.54> set maintenance
Reply: SetMaintenance SUCCESS (id = 5)
10.116.102.54> █
```

Рисунок 7.33 – Включение режима технического обслуживания кластера

Для получения информации о статусе кластера в режиме технического обслуживания или входящих в него узлов применяется команда:

```
cluster status
```

Узлы кластера, для которого установлен режим технического обслуживания, будут находиться в режиме «Master» (MAINTENANCE) или «Slave» (MAINTENANCE)

```
10.116.102.54> cluster status
Reply: (id = 6)
+-----+
|Jadog version|3.1.0|
+-----+

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

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

+-----+
|Node name  |Node      |State      |Connection state|Replication type|Write LSN|Replication slot name|Slot status|Public IP|Node status|
+-----+
|Datacenter: DEFAULT|
+-----+
|test_node1 |10.116.102.54(:12345)|Master(MAINTENANCE)|t        |-        |0/1589248|-        |f        |f        |f        |f        |
+-----+
```

Рисунок 7.34 – Проверка статуса узлов в кластере

### 7.16.3. Вывод кластера из режима технического обслуживания

Осуществляется командой:

```
reset maintenance
```

```
10.116.102.54> reset maintenance
Reply: ResetMaintenance SUCCESS (id = 7)
10.116.102.54>
```

Рисунок 7.35 – Выключение режима технического обслуживания кластера

Вывод кластера из режима ТО осуществляется по следующим правилам:

- Первым из режима ТО выводится главный узел с ролью «Master», затем узлы с ролями «Slave»;
- При использовании каскадной репликации (см. документ Компонент jaDog. Управление режимом работы узлов кластера 643.72410666.00067-07 98 02-02) узлы с ролями «Slave» выводятся из режима ТО в следующем порядке: сначала Primary, затем – Secondary.

Если главный узел кластера с ролью «Master» не выполнил выход из режима ТО, то узлы с ролями «Slave» также из него не выводятся и сохраняют статус MAINTENANCE.

Если узел кластера не может быть выведен из режима ТО по каким-либо причинам (в том числе из-за возникших ошибок), данное событие фиксируется в журнале работы. В этом случае узлы сохраняют свой статус MAINTENANCE и остаются в режиме ТО.

#### 7.16.4. Вывод отдельного узла из режима технического обслуживания

Осуществляется командой с указанием названия узла:

```
reset maintenance on node ['node_name']
```

Или с указанием IP-адреса и номера порта узла:

```
reset maintenance on node [ip] [port]
```

Если узел кластера с ролью «Master» не выполнил выход из режима ТО, то узлы с ролями «Slave» также из него не выводятся и сохраняют статус MAINTENANCE.

Если узел кластера не может быть выведен из режима ТО по каким-либо причинам (в том числе из-за возникших ошибок), данное событие фиксируется в журнале работы. В этом случае узел сохраняет свой статус MAINTENANCE и остается в режиме ТО.

#### 7.17. Процедура обработки отказа (failover)

Компоненты «jaDog» на узлах кластера непрерывно следят за статусами сетевых соединений между собой. В случае прерывания соединения между резервным и главным узлами компонентом «jaDog» автоматически запускается процедура обработки отказа «failover».

В случае отказа резервного узла – главный узел продолжит обслуживание в обычном режиме, ожидая в соответствии с заданными параметрами возобновления связи с резервным узлом (количество попыток в заданный интервал времени). Если в процессе ожидания узел активируется, он автоматически включается в кластер и восстанавливается.

В случае отказа главного узла – резервный узел автоматически переводит роль главного узла на себя, переводит общий адрес на себя, переводит статус главного узла в статус «DIED» (вышедший из строя) и продолжит ожидание восстановления вышедшего из строя узла.

Пользователь имеет возможность запросить статус узлов кластера командой «cluster status» в утилите «jalogctl».

В журнале событий резервного узла (ставшего новым главным узлом) будут отображаться следующие сообщения:

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

```
2021-02-04 19:02:40 ERROR: connection attempt 9/10 till
failover
2021-02-04 19:02:43 ERROR: TCPSocket::connect::connection
failed
2021-02-04 19:02:43 ERROR: NetworkAPI::errorHandler
2021-02-04 19:02:50 ERROR: Master jadog with ip 10.10.41.158 is
unconnected
2021-02-04 19:02:50 ERROR:      connection attempt 10/10 till
failover
2021-02-04 19:02:53 ERROR: TCPSocket::connect::connection
failed
2021-02-04 19:02:53 ERROR: NetworkAPI::errorHandler
2021-02-04 19:03:00 DEBUG: jaDog with ip 10.10.41.158 is died
2021-02-04 19:03:00 DEBUG: ClusterManager::failover
2021-02-04 19:03:00 DEBUG: ClusterManager::failover winner is
10.10.41.159
2021-02-04 19:03:00 DEBUG: SWITCHTOMASTER
2021-02-04 19:03:00 INFO: ожидание повышения сервера.... готово
сервер повышен
2021-02-04 19:03:00 DEBUG: PGPromote success
```

При возвращении резервного узла можно увидеть сообщение:

```
INFO: jaDog connection with ip 10.10.41.159 was stabilized
```

Кроме того, в компоненте «jaDog» предусмотрена возможность обработки ситуации «раздвоения» («split brain»), которая может возникнуть при четном количестве узлов в кластере (2, 4 и т.д.) при отказе сетевых соединений любого из узлов (например, отказ порта коммутатора, обрыв кабеля и т.п. ситуации). В этой ситуации главный и резервный узлы продолжают работать, при этом между ними пропадает связь и каждый из них не может точно определить, по какой причине не отвечает соседний узел: он действительно физически вышел из строя (отказ диска, отказ блока питания, отказ других компонент) или возникли неполадки сетевого характера. Принять решение о дальнейших действиях на каждом узле без дополнительных данных невозможно. По умолчанию, как было описано ранее, алгоритм предполагает следующие действия узлов:

- резервный узел переводит себя в роль главного;
- главный узел ожидает восстановления связи с резервным узлом (количество попыток в заданный интервал времени), по окончании ожидания и отсутствии связи – присваивает резервному узлу статус «DIED».

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

Таким образом, в случае «split brain» в результате получится два узла в роли «MATSER», не связанные друг с другом.

Во избежание такой ситуации в компоненте «jaDog» предусмотрен хост-арбитра в виде узла с ролью «Referee». Узел с ролью «Referee» является основным инструментом решения проблемы сетевого разделения узлов кластера в случае возникновения внештатных ситуаций. Описание применения узла с ролью «Referee» приведено в п.п. 7.27.

Альтернативным вариантом решения проблемы сетевого разделения узлов кластера является применение параметра «trusted\_address», который рекомендуется устанавливать всегда, когда в кластере четное количество узлов. «trusted\_address» является третьей независимой стороной, которая располагается в назначенном сетевым администратором надежном сетевом сегменте, и доступность которой прямо или косвенно обозначает доступность кластера СУБД со стороны пользователя. Задача сетевого администратора – выбрать такой ресурс или узел (настройка маршрутизации, введение дополнительного хоста-арбитра), доступность которого можно было бы проверить по протоколу ICMP (команда «ping»).

Параметр «trusted\_address» используется главным и резервным узлом для проверки доступности сетевой инфраструктуры и правильного принятия решения о произошедшем отказе.

При потере связи между узлами фиксируется отказ, после чего, если в настройках компонента «jaDog» задан параметр «trusted\_address»:

- 1) и он не доступен (не «пингуется»), то:
  - главный узел выполнит свое огораживание (fencing), удалит общий адрес с заданного сетевого интерфейса, остановит СУБД и перейдет в режим ожидания;
  - резервный узел также выполнит свое огораживание (fencing), остановит СУБД и перейдет в режим ожидания.
- 2) и он доступен, то при отказе:
  - резервного узла – главный узел останется главным и продолжит обслуживание пользователей;

– главного узла – резервный узел повысит свою роль до главного и включится в работу по обслуживанию запросов пользователей.

### 7.18. Восстановление после отказа (fail)

При восстановлении узла кластера после отказа, сначала узел для возобновления работы соединяется с главным на текущий момент узлом и синхронизируется с новой линией времени. В большинстве случаев этого шага достаточно для введения узла в рабочее состояние.

Если синхронизация заканчивается ошибкой, как правило, требуется делать повторное бэкапирование данных с главного сервера, что может занимать достаточное количество времени при больших объемах информации.

Чтобы обезопасить данные от утери, компонент «jaDog» не удаляет директорию с данными, а переименовывает ее.

Регулируют данное поведение следующие параметры:

- `after_rewind_fail_backup_allow(true/false)` – разрешает/запрещает использование бэкапирования после неудавшейся перемотки;
- `after_rewind_fail_backup_coeff` – коэффициент, регулирующий размер пространства на диске, которое должно быть свободно для новых данных в зависимости от директории с данными.

Например, если задано значение параметра «`after_rewind_fail_backup_coeff`» – «0.5», размер директории с данными равен 2 Гб, то минимальный объем пространства под новые данные, определяемый по формуле:  $\langle \text{значение параметра after\_rewind\_fail\_backup\_coeff} \rangle * \langle \text{размер директории с данными} \rangle$ , составит 1 Гб.

Бэкапирование после отказа перемотки выполнится только при следующих условиях:

- параметр «`after_rewind_fail_backup_allow`» установлен в значение «true»;
- размер свободного пространства на разделе больше, чем значение, вычисленное по формуле:

$$1 + \langle \text{значение параметра after\_rewind\_fail\_backup\_coeff} \rangle * \langle \text{размер директории с данными} \rangle .$$

В результате бэкапирования в директории данных компонентом «jaDog» будет создана копия прошлой (на момент отказа) директории данных с числовым порядковым суффиксом «N» – «dataN» (если таких директорий накопилось несколько).

Администратору необходимо проанализировать содержимое копии директории и определить наличие или отсутствие в аварийной копии требующих сохранения данных. При отсутствии в копиях «dataN» важных данных, копии можно удалить.



Основной директорией всегда остается директория «data», указанная в параметре «db\_data\_path»

При других аварийных ситуациях возвращение в работу узла выполняется с соблюдением следующих рекомендаций:

Для главного узла:

1) В случае кратковременного отключения прошлый главный узел включается в работу автоматически. Автоматическое восстановление запускается после включения бывшего главного узла в работу.

2) В случае длительных отключений прошлый главный узел может не включиться в восстановление автоматически. Требуется выполнить следующие процедуры на аварийном узле.

– отключить общий адрес, если таковой остался при аварии на главном узле:

для ОС Windows:

```
netsh interface ip delete address name="Ethernet" 10.10.41.163;
```

для ОС Ubuntu:

```
ip addr del 10.10.41.95 dev eth0
```

– перезапустить службу jadog:

для ОС Windows:

```
sc stop jadogservice  
sc start jadogservice
```

для ОС Ubuntu:

```
systemctl restart jadog
```

3) В случае отказа восстановления, при наличии ошибок в файле «jadog.log», касающихся запуска процедур «rewind» и «backup» необходимо:

- удалить файлы \*.BACKUP в директории данных «jatoba»;
- переместить или удалить старую директорию данных;
- повторить перезапуск службы jadog:

для ОС Windows:

```
sc stop jadogservice  
sc start jadogservice
```

для ОС Ubuntu:

```
systemctl restart jadog
```

Для резервного узла:

1) В случае успешного восстановления аварийного узла в роли резервного (всегда в роли резервного) в журнале событий будет сообщение вида:

```
2021-02-04 19:20:06 DEBUG: RECOVERY: local node state: Slave  
2021-02-04 19:20:06 DEBUG: RECOVERY: send state to 10.10.41.159  
2021-02-04 19:20:06 DEBUG: RECOVERY: remote node state: Master  
2021-02-04 19:20:06 DEBUG: RECOVERY: over
```

2) В случае ошибок администратору необходимо:

- проверить файл «jadog.log» на наличие ошибок, касающихся запуска процедур «rewind» и «backup»;
- устранить причины возникновения ошибок, обеспечить нормальный запуск процедур «rewind» и «backup»;
- после восстановления – проверить наличие старых директорий с данными, удостовериться в отсутствии исключительных записей и удалить лишние директории из текущего раздела.





Ответственность за утерянные при удалении данные несет администратор

Пользователь имеет возможность проконтролировать восстановление состояния кластера, для чего необходимо запросить статус узлов кластера командой «cluster status» в утилите «jadowctl». В ходе восстановления резервного узла его статус «Rewind», после успешного восстановления статус изменился на «Slave».

В редких случаях, если на резервном узле кластера произошла остановка сервиса «jatoba», повторный запуск СУБД при помощи сторонних средств, например использование команды

```
pg_ctl -D /var/lib/jatoba/6/data start
```

может привести к неконтролируемому переходу резервного узла в циклический режим восстановления (Recovery).

Для устранения возникшей аварийной ситуации необходимо убедиться при помощи утилиты мониторинга процессов ОС «top» в наличии на резервном узле запущенного процесса postgres -D /var/lib/jatoba/6/data. В этом случае выполняется его остановка средствами ОС и перезапуск сервиса «jadow» при помощи команды:

```
systemctl restart jadow
```

После выполнения указанных выше процедур резервный узел будет успешно подключен к кластеру.

### 7.19. Настройка SSL соединения в ОС GNU/Linux компонента «jaDog» с СУБД

В конфигурационном файле СУБД «pg\_hba.conf» требуется установить метод аутентификации «SSL»

#	TYPE	DATABASE	USER	ADDRESS	METHOD
hostssl	replication		jadow	IP/32	cert
hostssl	all		jadow	IP/32	cert
hostssl	all		jadow	127.0.0.1/32	cert

Также установить следующие параметры в конфигурационном файле СУБД postgres.conf

```
ssl = on
ssl_ca_file = '/path/to/ca.crt'
ssl_cert_file = '/path/to/server.crt'
ssl_crl_file = '/path/to/ca.crl'
ssl_key_file = '/path/to/server.key'
```

## 7.20. Настройка SSL соединения в ОС GNU/Linux для узлов кластера

Компонент «jaDog» поддерживает SSL соединение между узлами кластера.

Схема взаимодействия представлена на рисунке 7.36.

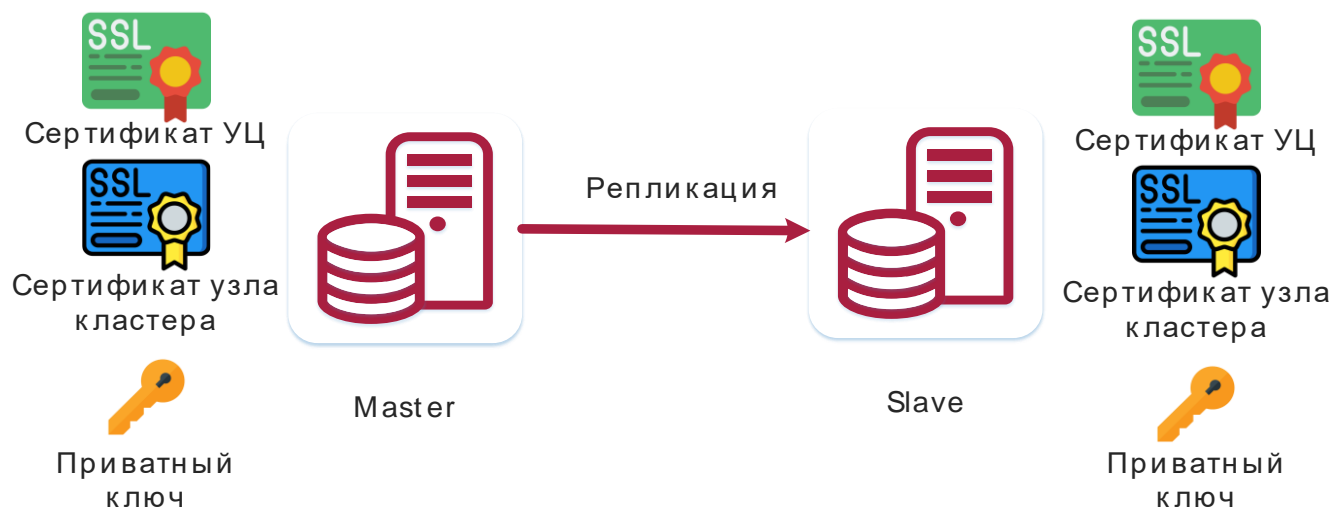


Рисунок 7.36 – Схема взаимодействия узлов кластера по протоколу SSL

Описанный ниже пример настройки компонента «jaDog» для SSL соединения подходит для подключения администратора кластера по протоколу SSL, используя строку подключения описанную в п 4.2.7.

### 7.20.1. Установка криптопровайдера

Сборки продуктов загружаются с официального сайта ООО «Крипто-Про» <https://www.cryptopro.ru/>. Потребуется криптопровайдер библиотеки OpenSSL и пакет OpenSSL.

На сайте ООО «Крипто-Про» перейти по следующему пути:

Скачать – КриптоПро CSP – Сертифицированные версии – «Крипто Про CSP 5.0» – для – «Unix» – КриптоПро CSP 5.0 для Linux (x64, deb)

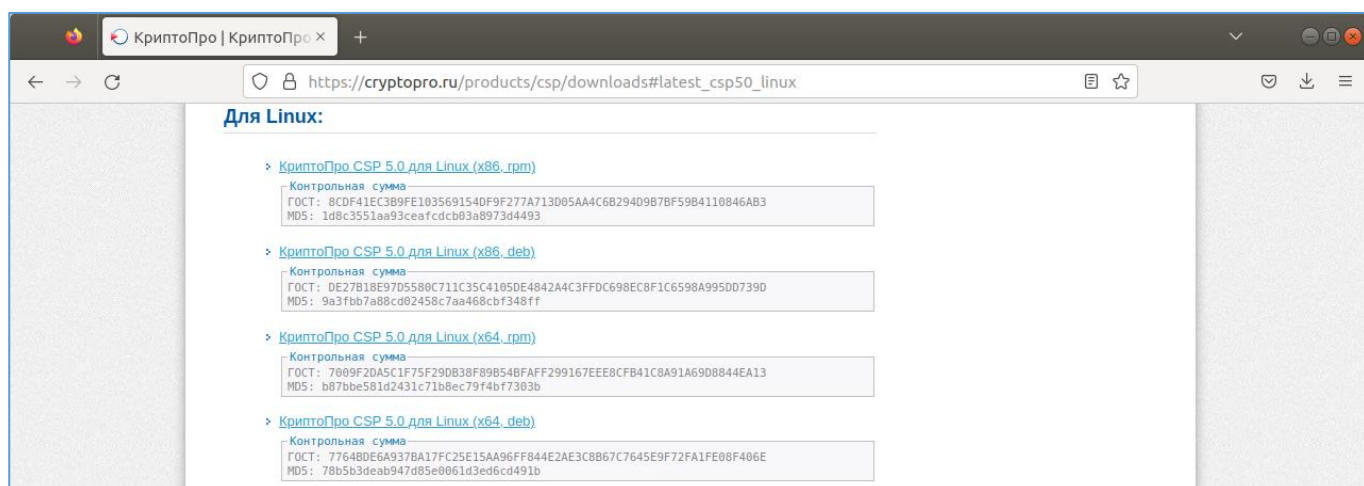


Рисунок 7.37 – Страница загрузки

Распаковать полученный архив linux-amd64\_deb.tgz, например, в каталог «Download».

Перейти в каталог от имени и с правами привилегированного пользователя и выполнить команду:

```
./install.sh
```

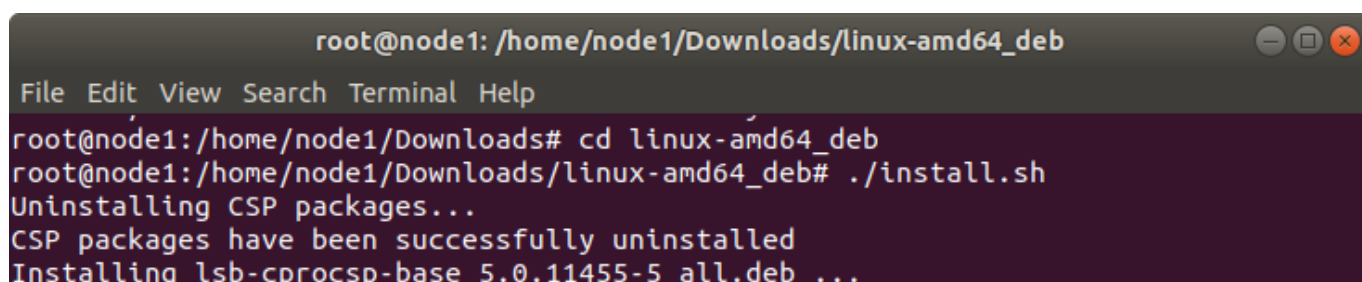


Рисунок 7.38 – Команда установки

После чего начнется установка дистрибутива.

В результате будет выведено сообщение:

```
CSP packages have been successfully installed
```

### 7.20.2. Установка библиотеки OpenSSL с поддержкой ГОСТ 2012

Библиотека OpenSSL с поддержкой ГОСТ 2012 необходимо загрузить по адресу:

<https://update.cryptopro.ru/support/nginx-gost/bin/174535/>

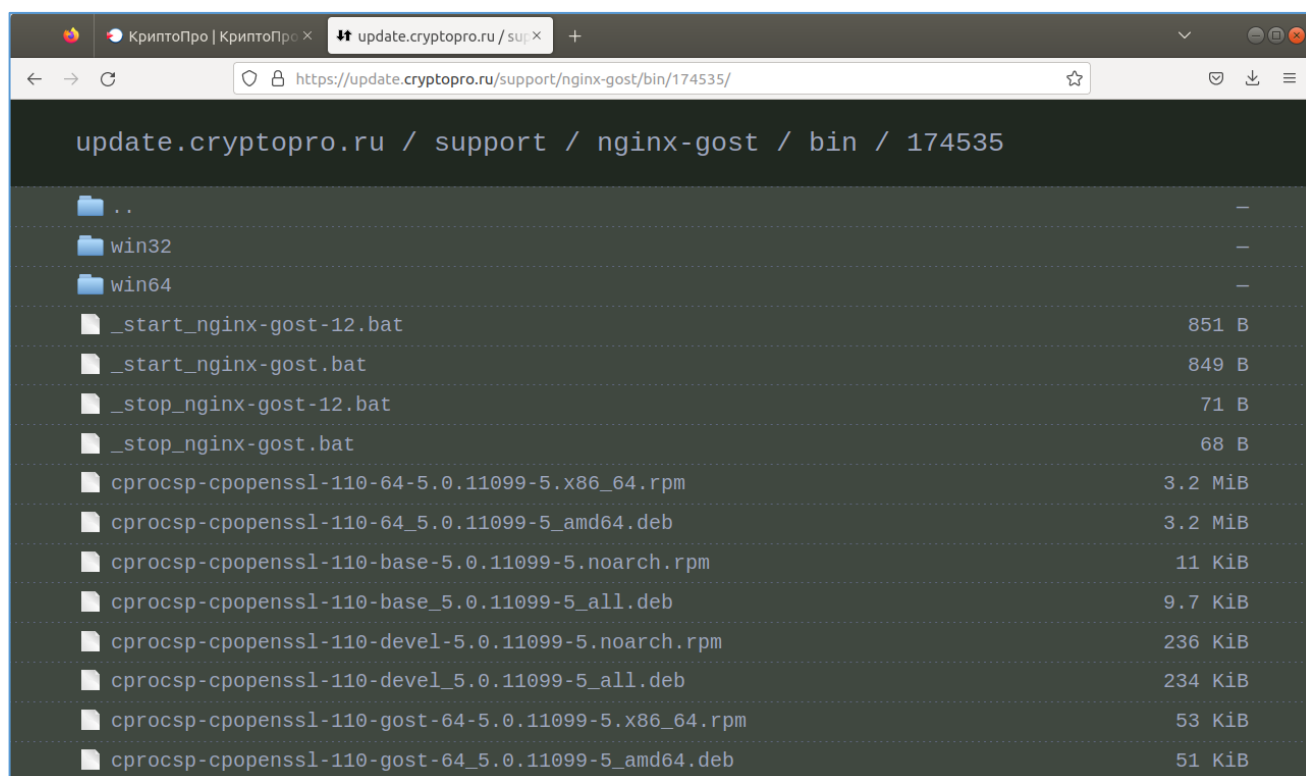


Рисунок 7.39 – Страница загрузки библиотеки OpenSSL

В терминале, от имени и с правами привилегированного пользователя, перейти к полученному файлу `cprocsp-cpopenSSL-110-gost-64_5.0.11099-5_amd64.deb`. По умолчанию загруженный файл сохранится в каталоге пользователя `/home/<user_name>/Downloads`.

```
cd /home/node1/Downloads
```

Установить пакет при помощи команды:

```
dpkg -i cprocsp-cpopenSSL-110-gost-64_5.0.11099-5_amd64.deb
```

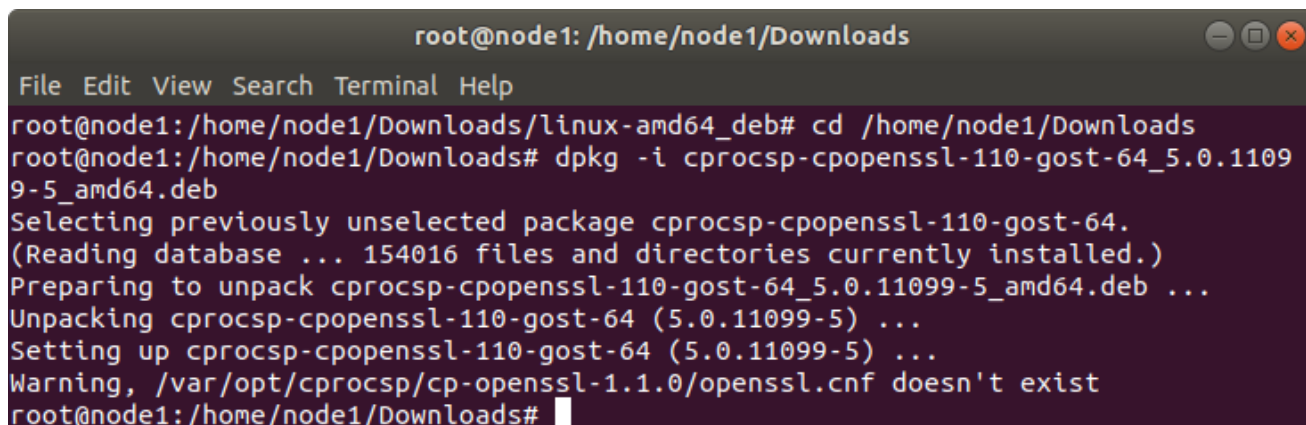
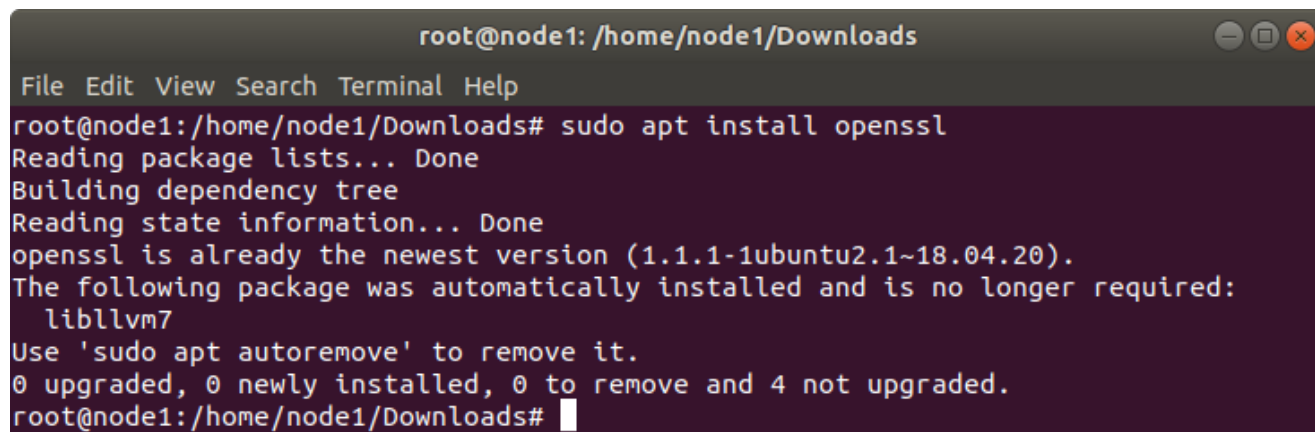


Рисунок 7.40 – Установка пакета библиотеки OpenSSL

### 7.20.3. Пакет OpenSSL

Пакет OpenSSL устанавливается командой:

```
sudo apt install openssl
```



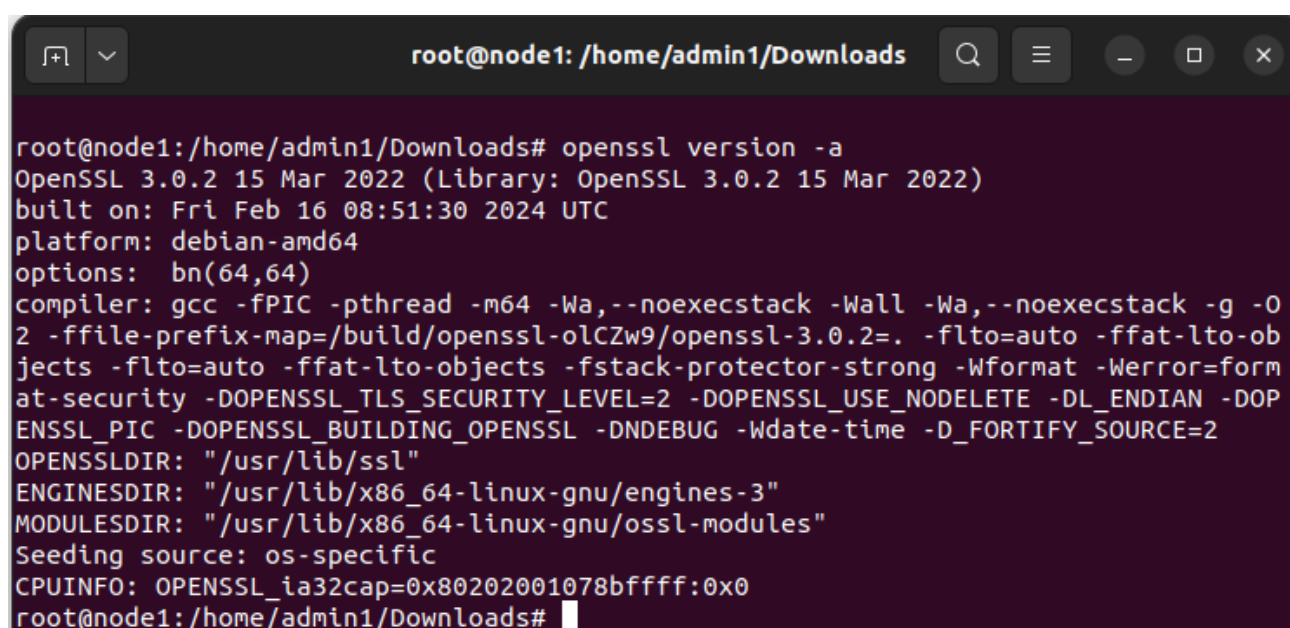
```
root@node1: /home/node1/Downloads
File Edit View Search Terminal Help
root@node1:/home/node1/Downloads# sudo apt install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1-1ubuntu2.1~18.04.20).
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
root@node1:/home/node1/Downloads#
```

Рисунок 7.41 – Установка пакета OpenSSL

Определение используемой версии OpenSSL является важным первым шагом в подготовке генерации приватного ключа и запроса CSR. Версия OpenSSL определяет криптографический алгоритм, который может использоваться для генерации ключей и поддерживаемый протокол.

Выполнить определение версии OpenSSL командой:

```
openssl version -a
```



```
root@node1: /home/admin1/Downloads
root@node1:/home/admin1/Downloads# openssl version -a
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
built on: Fri Feb 16 08:51:30 2024 UTC
platform: debian-amd64
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -O2 -ffile-prefix-map=/build/openssl-olCZw9/openssl-3.0.2=. -flto=auto -ffat-lto-objects -flto=auto -ffat-lto-objects -fstack-protector-strong -Wformat -Werror=format-security -DOPENSSL_TLS_SECURITY_LEVEL=2 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=2
OPENSSLDIR: "/usr/lib/ssl"
ENGINESDIR: "/usr/lib/x86_64-linux-gnu/engines-3"
MODULESDIR: "/usr/lib/x86_64-linux-gnu/openssl-modules"
Seeding source: os-specific
CPUINFO: OPENSSL_ia32cap=0x80202001078bffff:0x0
root@node1:/home/admin1/Downloads#
```

Рисунок 7.42 – Определение версии OpenSSL

В этой команде ключ «-a» включит отображение следующей информации:

- OpenSSL – номер версии и дата выхода;
- options – опции библиотеки;
- OPENSSLDIR – название директории, где находятся сертификат и приватные ключи.

Для последующей настройки требуется информация о параметре OPENSSLDIR.

#### 7.20.4. Настройка конфигурационного файла «openssl\_conf»

Перейти в каталог:

```
cd /usr/lib/ssl
```

Открыть файл «openssl\_conf» для редактирования командой:

```
nano openssl.cnf
```

В файле нужно проверить наличие параметров и закомментировать их:

```
#openssl_conf=default_modules  
#openssl_conf=default_conf
```

Внести в начало конфигурационного файла «openssl\_conf» строку:

```
openssl_conf=openssl_def
```

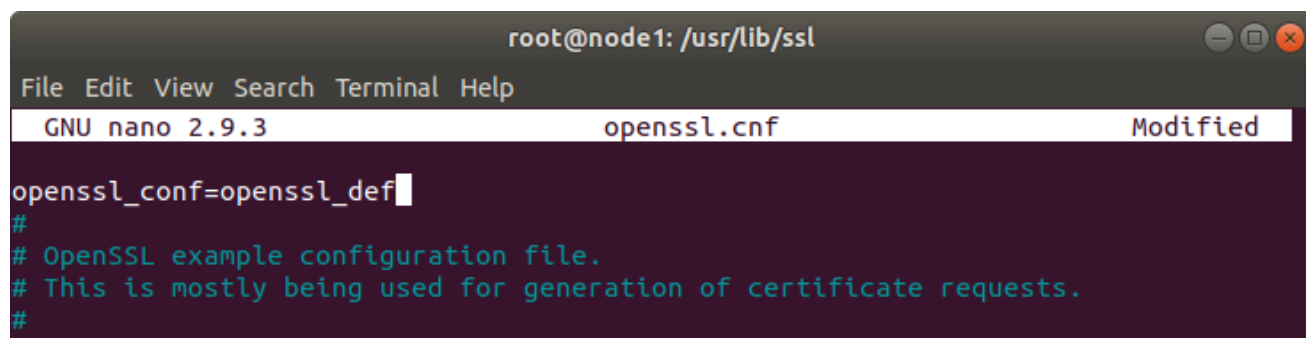


Рисунок 7.43 – Первая строка в конфигурационном файле «openssl.cnf»

Внести в конец конфигурационного файла «openssl\_conf» строки:

```
[openssl_def]
```

```
engines = engine_section
[engine_section]
gost_engy = gost_engy_section
[gost_engy_section]
engine_id = gostengy

dynamic_path = /opt/cprocsp/cp-openssl-
1.1.0/lib/amd64/engines/libgostengy.so

default_algorithms = CIPHERS, DIGESTS, PKEY, PKEY_CRYPT,
PKEY_ASN1
```

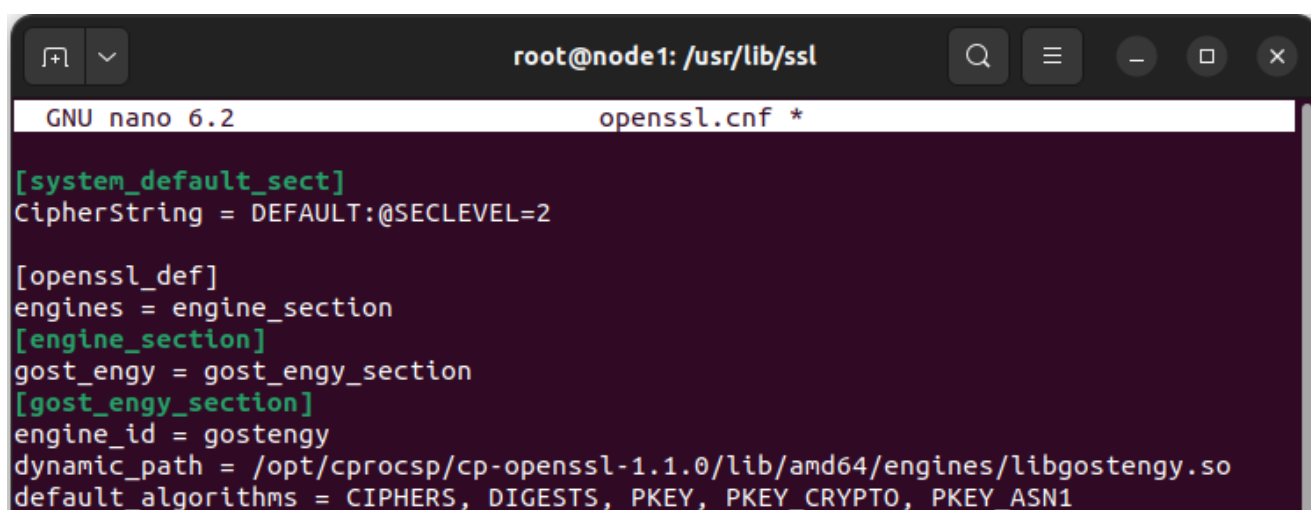


Рисунок 7.44 – Параметры в конфигурационном файле «openssl.cnf»

Для проверки работоспособности внесенных параметров выполнить команду:

```
openssl engine
```

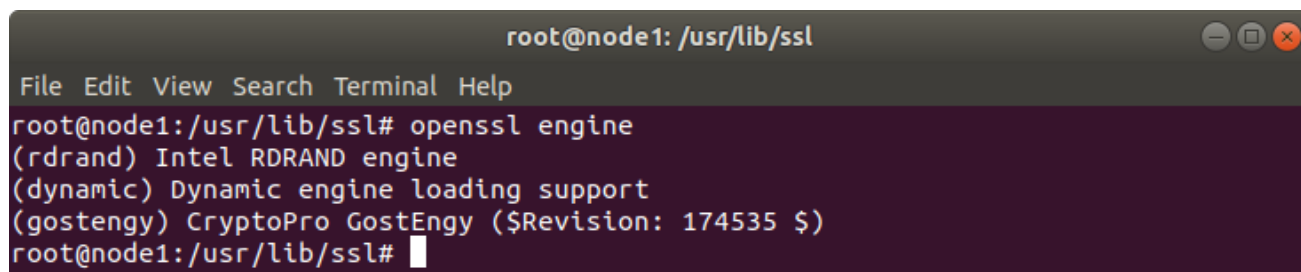


Рисунок 7.45 – Вывод команды «openssl engine»

Параметр «gostengy» должен показывать активную библиотеку OpenSSL (п. 7.20.2).

#### 7.20.5. Формирование сертификата

Формирование сертификата должно выполняться от имени и с правами учетной записи ОС root.

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



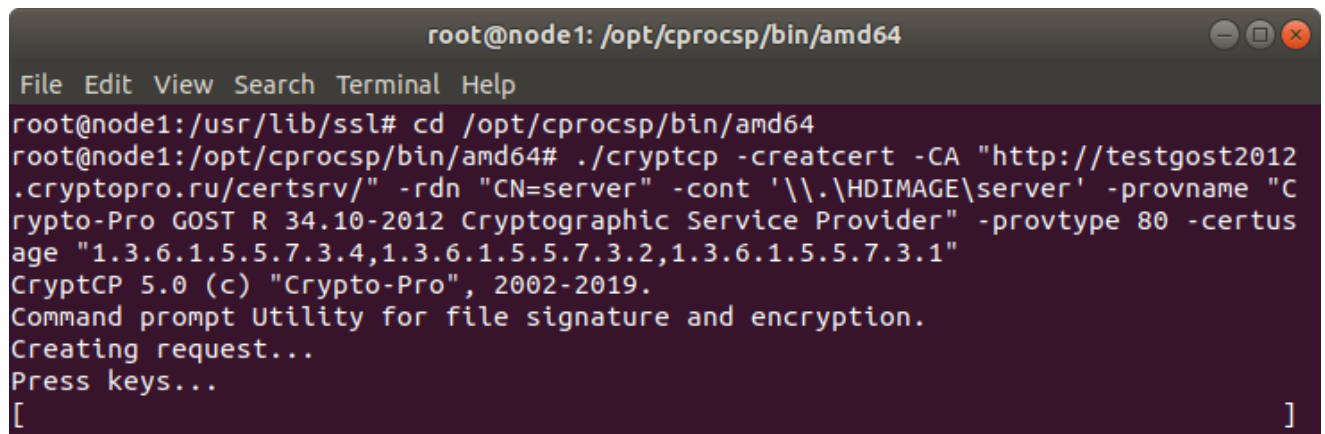
Перейти в каталог криптопровайдера:

```
cd /opt/cprocsp/bin/amd64
```

Выполнить команду для формирования сертификата:

```
./cryptcp -creatcert -CA  
"http://testgost2012.cryptopro.ru/certsrv/" -rdn "CN=server" -  
cont '\\.\HDIMAGE\server' -provname "Crypto-Pro GOST R 34.10-  
2012 Cryptographic Service Provider" -provtype 80 -certusage  
"1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2,1.3.6.1.5.5.7.3.1"
```

После чего станет активным датчик случайных чисел (ДСЧ), срабатывающий на ввод случайных символов с клавиатуры.

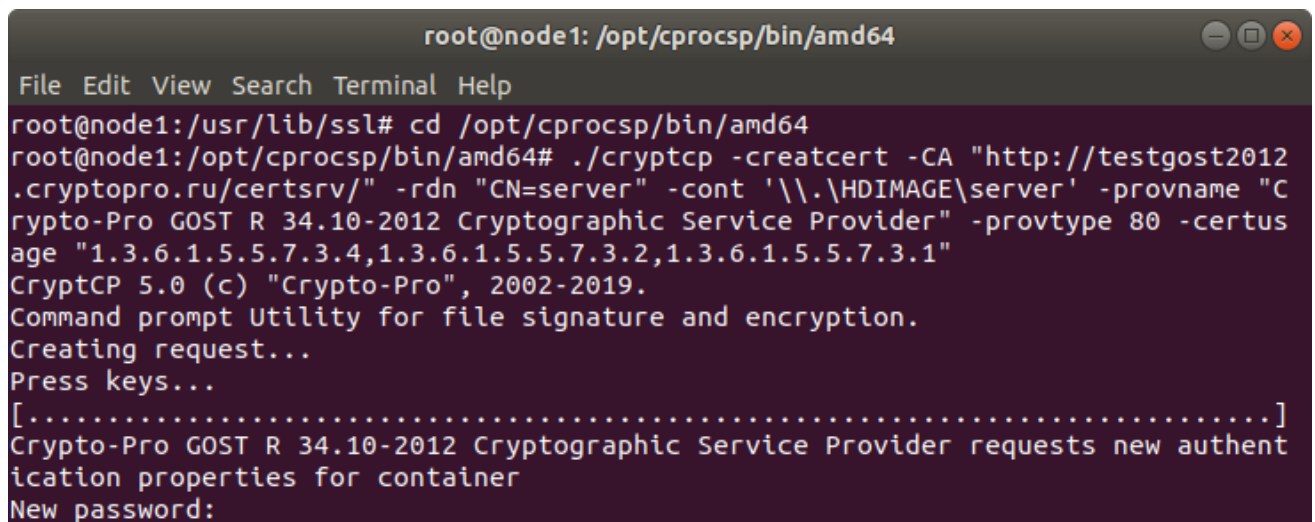


```
root@node1: /opt/cprocsp/bin/amd64  
File Edit View Search Terminal Help  
root@node1:/usr/lib/ssl# cd /opt/cprocsp/bin/amd64  
root@node1:/opt/cprocsp/bin/amd64# ./cryptcp -creatcert -CA "http://testgost2012  
.cryptopro.ru/certsrv/" -rdn "CN=server" -cont '\\.\HDIMAGE\server' -provname "C  
rypto-Pro GOST R 34.10-2012 Cryptographic Service Provider" -provtype 80 -certus  
age "1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2,1.3.6.1.5.5.7.3.1"  
CryptCP 5.0 (c) "Crypto-Pro", 2002-2019.  
Command prompt Utility for file signature and encryption.  
Creating request...  
Press keys...  
[
```

Рисунок 7.46 – ДСЧ криптопровайдера

При окончании работы ДСЧ предложит ввод пароля. Перед вводом пароля рекомендуется удалить случайно набранные символы после окончания работы ДСЧ.





```
root@node1: /opt/cprocsp/bin/amd64
File Edit View Search Terminal Help
root@node1:/usr/lib/ssl# cd /opt/cprocsp/bin/amd64
root@node1:/opt/cprocsp/bin/amd64# ./cryptcp -creatcert -CA "http://testgost2012
.cryptopro.ru/certsrv/" -rdn "CN=server" -cont '\\.\HDIMAGE\server' -provname "C
rypto-Pro GOST R 34.10-2012 Cryptographic Service Provider" -provtype 80 -certus
age "1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2,1.3.6.1.5.5.7.3.1"
CryptCP 5.0 (c) "Crypto-Pro", 2002-2019.
Command prompt Utility for file signature and encryption.
Creating request...
Press keys...
[.....]
Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider requests new authent
ication properties for container
New password:
```

Рисунок 7.47 – Ввод пароля

В процессе формирования сертификата будет выведено сообщение:

Do you want to add it to ROOT store

с предложением добавить сертификат в корневое хранилище. Следует отказаться от операции, т.к. далее сертификат будет экспортирован в каталог:

/usr/jatoba-6/var

как описано в п. 7.20.6 настоящего документа.

```
root@node1: /opt/cprocsp/bin/amd64
File Edit View Search Terminal Help
.cryptopro.ru/certsrv/" -rdn "CN=server" -cont '\\.\HDIMAGE\server' -provname "C
rypto-Pro GOST R 34.10-2012 Cryptographic Service Provider" -provtype 80 -certus
age "1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2,1.3.6.1.5.5.7.3.1"
CryptCP 5.0 (c) "Crypto-Pro", 2002-2019.
Command prompt Utility for file signature and encryption.
Creating request...
Press keys...
[.....]
Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider requests new authent
ication properties for container
New password:
Confirm password:
Sending request to CA...
Installing certificate...
Message contains root certificate that is absent in your root store.
RDN:"Тестовый УЦ ООО ""КРИПТО-ПРО""", "ООО ""КРИПТО-ПРО""", Москва, г. Москва, R
U, ул. Суцёвский вал д. 18, 001234567890, 1234567890123
Valid from 12.09.2018 10:19:30 to 12.09.2023 10:28:55

Do you want to add it to Root store? [Y]es or [N]o: n
Not adding certificate to Root store.
Certificate is installed.
[ErrorCode: 0x00000000]
root@node1: /opt/cprocsp/bin/amd64#
```

Рисунок 7.48 – Команда формирования сертификата

В результате сертификат сформируется и будет выведено сообщение:

```
Certificate is installed.
[ErrorCode: 0x00000000]
```

#### 7.20.6. Экспорт сформированного сертификата

Сформированный сертификат экспортировать в защищенную директорию:

```
/usr/jatoba-6/var
```

Экспорт сертификата выполняется командой:

```
./certmgr -export -store uMy -dn "server" -dest /usr/jatoba-
6/var/server.crt -base64
```

```

root@node1: /opt/cprosp/bin/amd64
root@node1:/opt/cprosp/bin/amd64# ./certmgr -export -store uMy -dn "server" -dest /usr/jatoba-5/var/server.crt -base64
Certmgr 1.1 (c) "Crypto-Pro", 2007-2019.
program for managing certificates, CRLs and stores

Exporting:
=====
1-----
Issuer       : OGRN=1234567890123, INN=001234567890, STREET=ул. Суцёвский вал д. 18, C=RU, S=г. Москва, L=Москва,
O="000 ""КРИПТО-ПРО""", CN="Тестовый УЦ 000 ""КРИПТО-ПРО""
Subject      : CN=server
Serial       : 0x7C0013CA34C114BFC5BC9139A300060013CA34
SHA1 Hash    : 142ab490b207bd391e3122e85ce70fdb8ef72444
SubjKeyID    : 64a725805e8c5c48df6130c87277bd977d5b06ba
Signature Algorithm : ГОСТ Р 34.11-2012/34.10-2012 256 бит
PublicKey Algorithm : ГОСТ Р 34.10-2012 (512 bits)
Not valid before : 19/03/2024 09:28:53 UTC
Not valid after  : 05/05/2024 12:25:00 UTC
PrivateKey Link  : Yes
Container       : HDIMAGE\server.000\16DC
Provider Name    : Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider
Provider Info    : ProvType: 80, KeySpec: 1, Flags: 0x0
OCSP URL        : http://testgost2012.cryptopro.ru/ocsp2012g/ocsp.srf
CA cert URL     : http://testgost2012.cryptopro.ru/ocsp2012gst/ocsp.srf
CA cert URL     : http://testgost2012.cryptopro.ru/CertEnroll/testgost2012(6).crt
CA cert URL     : http://testgost2012.cryptopro.ru/CertEnroll/root2024-6.crt
CA cert URL     : http://testgost2012.cryptopro.ru/CertEnroll/testroot.p7b
CDP             : http://testgost2012.cryptopro.ru/CertEnroll/!0422!0435!0441!0442!043e!0432!044b!0439%20!0423!0426%2
0!041e!041e!041e%20!0022!041a!0420!0418!041f!0422!041e-!041f!0420!041e!0022(6).crl
CDP             : http://testgost2012.cryptopro.ru/CertEnroll/testgost2012(6).crl
Extended Key Usage : 1.3.6.1.5.5.7.3.4
                  : 1.3.6.1.5.5.7.3.2
                  : 1.3.6.1.5.5.7.3.1
=====
Export complete

[ErrorCode: 0x00000000]
root@node1:/opt/cprosp/bin/amd64#

```

Рисунок 7.49 – Окно формирования сертификата

По окончании экспорта будет выведено сообщение:

```

Export complete
[ErrorCode: 0x00000000]

```

### 7.20.7. Редактирование сформированного сертификата

Экспортированный сертификат требует внесения дополнительных строк.

Для чего потребуется перейти в директорию командой:

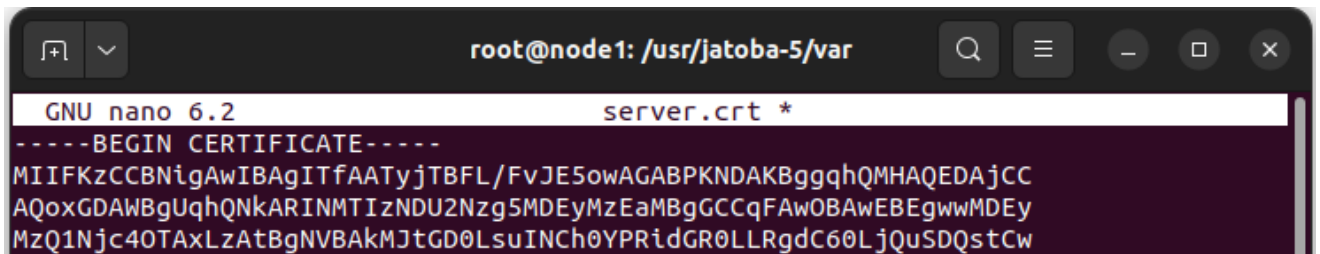
```
cd /usr/jatoba-6/var
```

Открыть для редактирования файл сертификата командой:

```
nano server.crt
```

Вставить первую строку:

```
-----BEGIN CERTIFICATE-----
```



The screenshot shows a terminal window with the title bar 'root@node1: /usr/jatoba-5/var'. The editor is GNU nano 6.2, editing a file named 'server.crt'. The content of the file is a Base64-encoded certificate, starting with '-----BEGIN CERTIFICATE-----' followed by several lines of text, including 'MIIFKzCCBNigAwIBAgITfAATyjTBFL/FvJE5owAGABPKNDABgqghQMHAQEDAJCC'.

Рисунок 7.50 – Первая строка сертификата

Вставить последнюю строку:

```
-----END CERTIFICATE-----
```



The screenshot shows the same terminal window as Figure 7.50, but now the file 'server.crt' contains the full certificate. The content ends with '-----END CERTIFICATE-----'.

Рисунок 7.51 – Последняя строка сертификата

Сохранить сформированный файл.

#### 7.20.8. Установка корневого сертификата УЦ и списка отозванных сертификатов CRL

Загрузка корневого сертификата центра сертификации осуществляется по адресу:

<https://www.cryptopro.ru/certsrv/certcarc.asp>

В открывшейся странице установить метод шифрования «Base64». После чего перейти по ссылке «Загрузка сертификата ЦС».

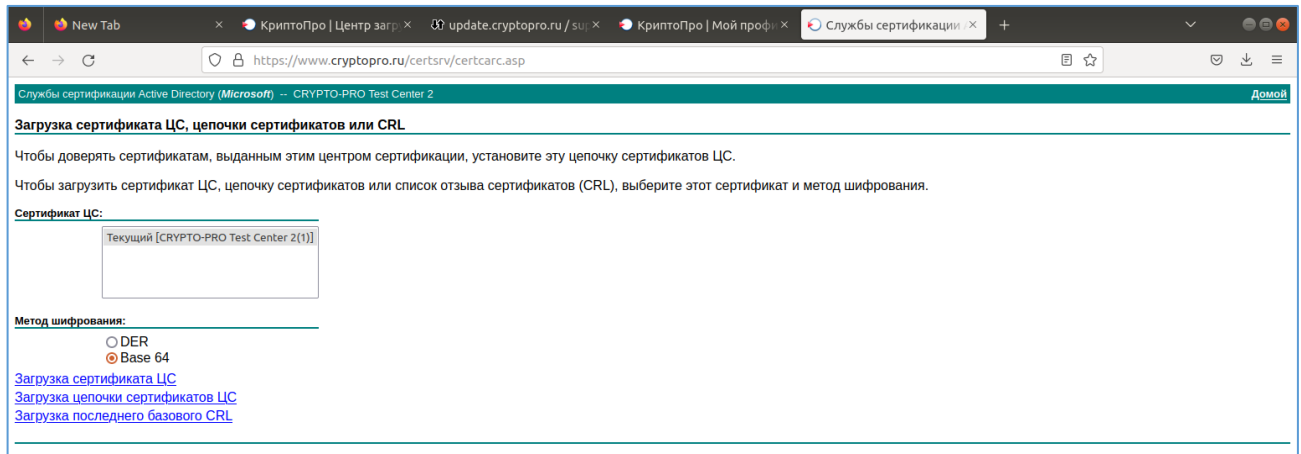


Рисунок 7.52 – Страница загрузки корневого сертификата УЦ

Скачать список отозванных сертификатов CRL через гиперссылку «Загрузка последнего базового CRL».

Полученные сертификаты ЦС из каталога «Downloads» переместить в каталог:

```
/usr/jatoba-6/var
```

В результате проделанных операций в каталоге /usr/jatoba-6/var будут храниться сертификаты удостоверяющего центра, сформированный сертификат и список отозванных сертификатов.

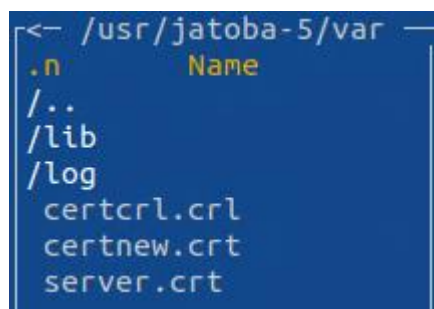


Рисунок 7.53 – Содержание каталога /usr/jatoba-6/var

### 7.20.9. Настройка главного узла для SSL соединения

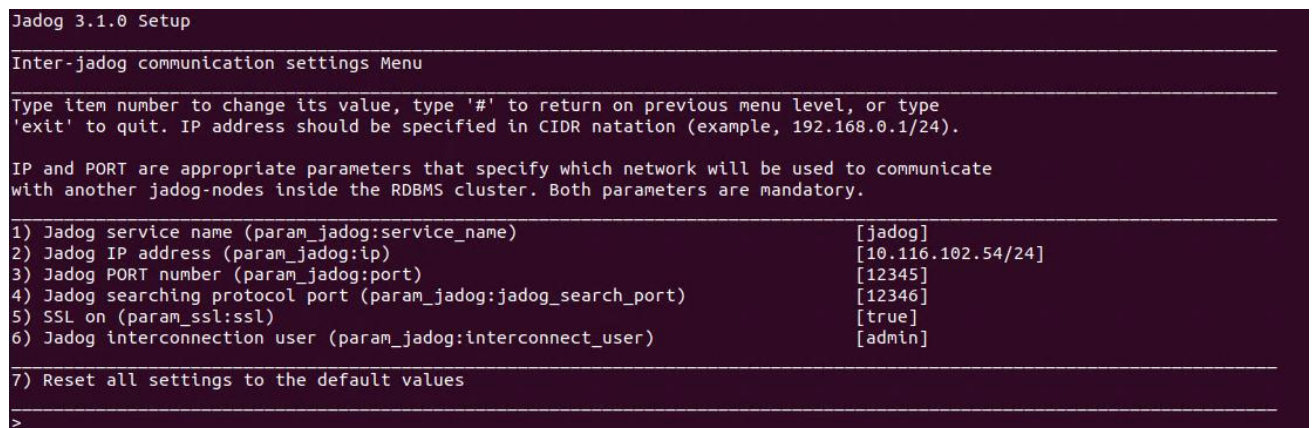
Для перехода в режим редактирования конфигурации компонента «jaDog» необходимо выполнить команды.

Настройка главного узла кластера для SSL соединения выполняется через запуск сервиса «jadog». В случае, когда настройка кластера уже проводилась, утилита запускается в режиме редактирования существующих настроек:

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

При запуске откроется главное меню jadog setup, в котором:

- выбрать пункт 2 «Inter-jadog communication setting»;
- в пункте 5 SSL on (SSL) установить параметр «true»;



```
Jadog 3.1.0 Setup  
-----  
Inter-jadog communication settings Menu  
-----  
Type item number to change its value, type '#' to return on previous menu level, or type  
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).  
  
IP and PORT are appropriate parameters that specify which network will be used to communicate  
with another jadog-nodes inside the RDBMS cluster. Both parameters are mandatory.  
-----  
1) Jadog service name (param_jadog:service_name) [jadog]  
2) Jadog IP address (param_jadog:ip) [10.116.102.54/24]  
3) Jadog PORT number (param_jadog:port) [12345]  
4) Jadog searching protocol port (param_jadog:jadog_search_port) [12346]  
5) SSL on (param_ssl:ssl) [true]  
6) Jadog interconnection user (param_jadog:interconnect_user) [admin]  
-----  
7) Reset all settings to the default values  
-----  
>
```

Рисунок 7.54 – Активация режима «SSL»

- перейти в главное меню при помощи ввода символа #;

В главном меню добавится меню «Security connection settings» под номером «4».



```
Jadog 3.1.0 Setup
-----
Main Menu
-----
Type item number to pick settings group you want to adjust or
type 'exit' to quit.
-----
1) Database server and jadog directory settings
2) Inter-jadog communication settings
3) User / Admin access network settings
4) Security connection settings
5) Administrator account list
6) Database server system account and connection settings
7) Failover settings
8) Replication settings
9) WAL archive and restore settings
10) Reporting and logging
11) Reset all settings to the default values
12) Check and show all settings
13) Save settings and setup jadog
-----
> |
```

Рисунок 7.55 – Вид главного меню с меню «Security connection settings»

- выбрать пункт 4 «Security connection settings»;
- в пункте 1 «SSL Center Authority certificate» (ssl\_ca\_file) указать путь к сертификату удостоверяющего центра;

```
Jadog 3.1.0 Setup
-----
Security connection settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).
-----
1) SSL Center Authority certificate (param_ssl:ssl_ca_file)                [/usr/jatoba-5/var/certnew.crt]
2) SSL User certificate (param_ssl:ssl_cert_file)                        []
3) SSL Certificate Revocation List (param_ssl:ssl_crl_file)              []
4) SSL User private key (param_ssl:ssl_key_file)                        []
-----
5) Reset all settings to the default values
-----
>
```

Рисунок 7.56 – Путь к сертификату УЦ

Сертификат УЦ был сохранен по пути:

```
/usr/jatoba-6/var/certnew.crt
```

как было описано в п. 7.20.8.

- в пункте 2 «SSL User certificate» (ssl\_cert\_file) указать путь к сертификату, созданного для узла кластера;

```
Jadog 3.1.0 Setup
-----
Security connection settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).
-----
1) SSL Center Authority certificate (param_ssl:ssl_ca_file)                [/usr/jatoba-5/var/certnew.crt]
2) SSL User certificate (param_ssl:ssl_cert_file)                        []
3) SSL Certificate Revocation List (param_ssl:ssl_crl_file)              []
4) SSL User private key (param_ssl:ssl_key_file)                        []
5) Reset all settings to the default values
-----
>
```

Рисунок 7.57 – Путь к сертификату узла кластера

Сертификат узла кластера был экспортирован по пути:

```
/usr/jatoba-6/var/server.crt
```

как было описано в п. 7.20.6.

– в пункте 3 «SSL Certificate Revocation List» (ssl\_crl\_file) указать путь к списку отозванных сертификатов CRL:

```
Jadog 3.1.0 Setup
-----
Security connection settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).
-----
1) SSL Center Authority certificate (param_ssl:ssl_ca_file)                [/usr/jatoba-5/var/certnew.crt]
2) SSL User certificate (param_ssl:ssl_cert_file)                        [/usr/jatoba-5/var/server.crt]
3) SSL Certificate Revocation List (param_ssl:ssl_crl_file)              [/usr/jatoba-5/var/certcrt.crt]
4) SSL User private key (param_ssl:ssl_key_file)                        []
5) Reset all settings to the default values
-----
>
```

Рисунок 7.58 – Путь к списку отозванных сертификатов

```
/usr/jatoba-6/var/certcrt.crt
```

– в пункте 4 «SSL User private key» (ssl\_key\_file) указать имя файла приватного ключа, который был сформирован при генерировании сертификата.

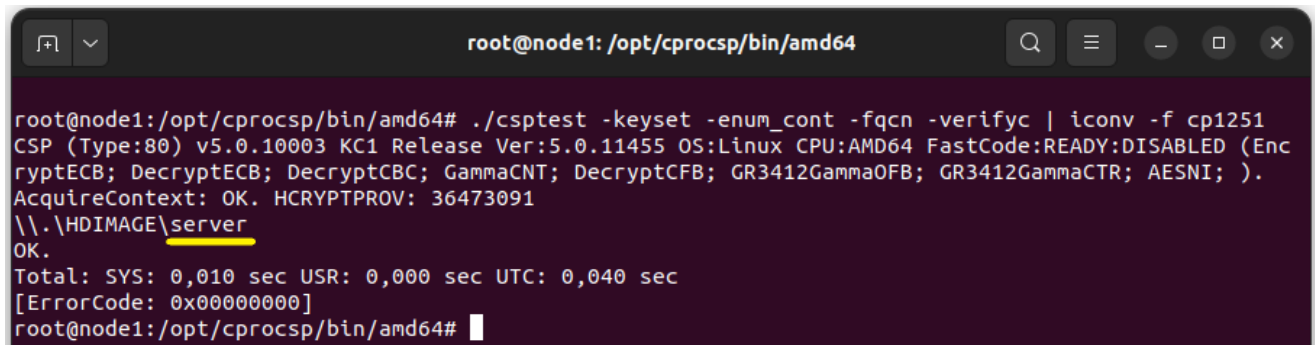
В ОС CNU/Linux нет собственных контейнеров для хранения ключей. Криптопровайдер КриптоПро CSP организует такие контейнеры. Чтобы просмотреть имя приватного ключа потребуется перейти в директорию:

```
cd /opt/cproscsp/bin/amd64/
```

и получить имя файла приватного ключа командой:



```
./csptest -keyset -enum_cont -fqcn -verifyc | iconv -f cp1251
```



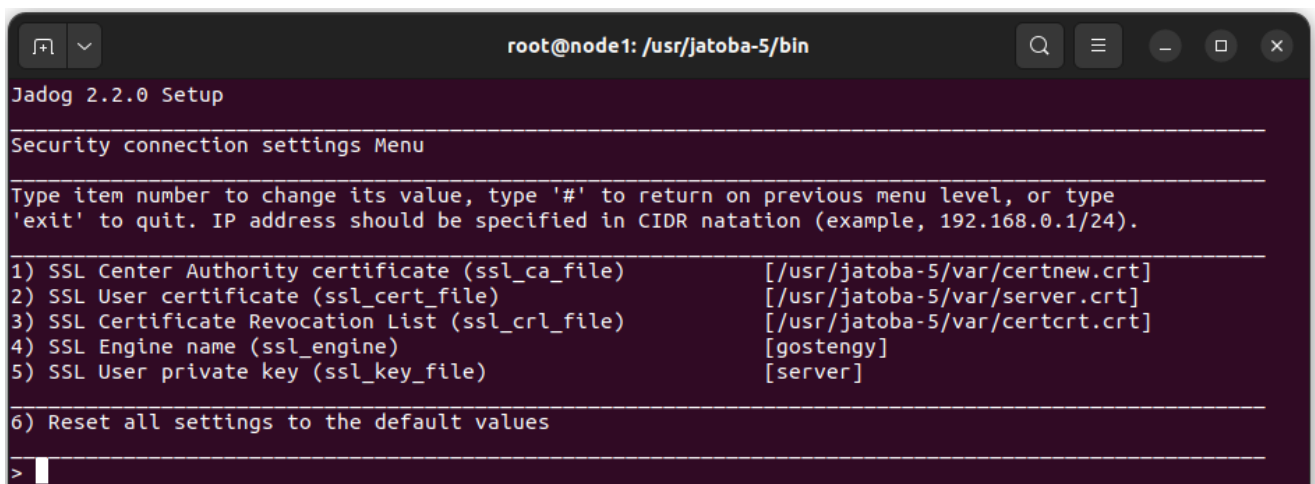
```
root@node1: /opt/cprocsp/bin/amd64
root@node1:/opt/cprocsp/bin/amd64# ./csptest -keyset -enum_cont -fqcn -verifyc | iconv -f cp1251
CSP (Type:80) v5.0.10003 KC1 Release Ver:5.0.11455 OS:Linux CPU:AMD64 FastCode:READY:DISABLED (Enc
ryptECB; DecryptECB; DecryptCBC; GammaCNT; DecryptCFB; GR3412GammaOFB; GR3412GammaCTR; AESNI; ).
AcquireContext: OK. HCRYPTPROV: 36473091
\\.\HDIMAGE\server
OK.
Total: SYS: 0,010 sec USR: 0,000 sec UTC: 0,040 sec
[ErrorCode: 0x00000000]
root@node1:/opt/cprocsp/bin/amd64#
```

Рисунок 7.59 – Команда получения имени файла приватного ключа

Полученное значение:

```
server
```

Указывается в пункте 4 «SSL User private key» (ssl\_key\_file).

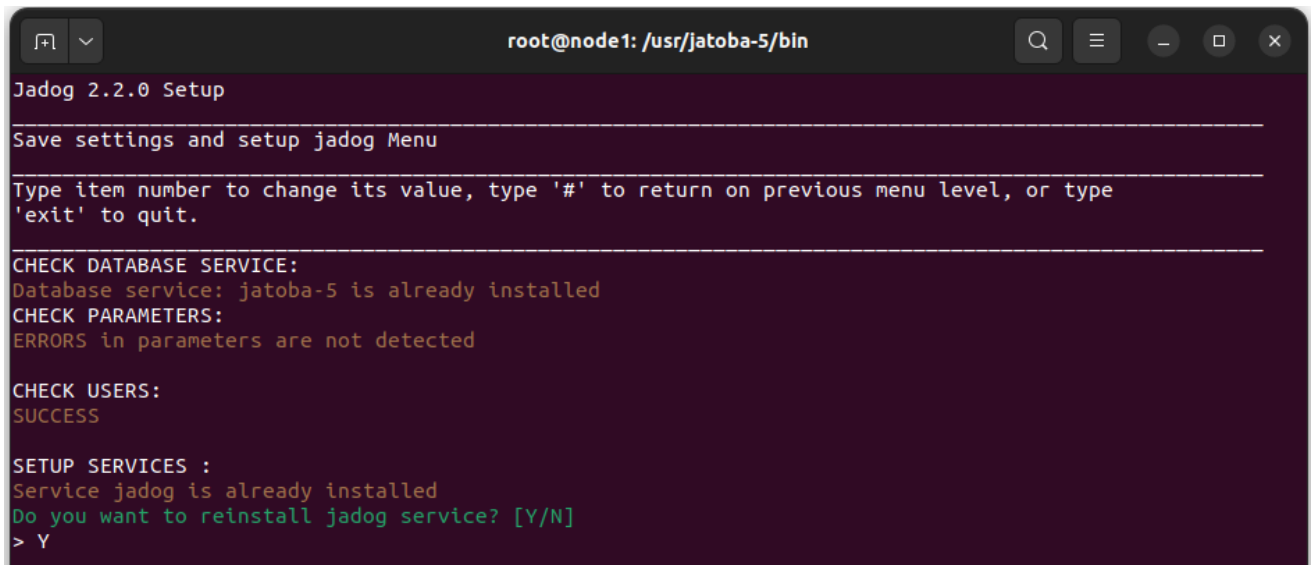


```
root@node1: /usr/jatoba-5/bin
Jadog 2.2.0 Setup
Security connection settings Menu
-----
Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit. IP address should be specified in CIDR notation (example, 192.168.0.1/24).
-----
1) SSL Center Authority certificate (ssl_ca_file)           [/usr/jatoba-5/var/certnew.crt]
2) SSL User certificate (ssl_cert_file)                    [/usr/jatoba-5/var/server.crt]
3) SSL Certificate Revocation List (ssl_crl_file)          [/usr/jatoba-5/var/certcrt.crt]
4) SSL Engine name (ssl_engine)                            [gostengy]
5) SSL User private key (ssl_key_file)                     [server]
-----
6) Reset all settings to the default values
-----
>
```

Рисунок 7.60 – Имя файла приватного ключа

- перейти в главное меню при помощи ввода символа #;
- повторно внести параметры значений в меню 6) Database server system account and connection settings;
- проверить корректность внесенных изменений через пункт основного меню «Check and show all setting»;
- сохранить внесенные изменения через пункт основного меню «Save setting and setup jadog»;

– подтвердить реинсталляции службы «jadog» в ОС, что позволит применить установленные параметры.



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

Jadog 2.2.0 Setup

Save settings and setup jadog Menu

Type item number to change its value, type '#' to return on previous menu level, or type
'exit' to quit.

CHECK DATABASE SERVICE:
Database service: jatoba-5 is already installed
CHECK PARAMETERS:
ERRORS in parameters are not detected

CHECK USERS:
SUCCESS

SETUP SERVICES :
Service jadog is already installed
Do you want to reinstall jadog service? [Y/N]
> Y
```

Рисунок 7.61 – Подтверждение перезапуска службы «jadog»

– подтвердить перезапустить службу «jadog» в ОС, что позволит применить установленные параметры командами:

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

#### 7.20.10. Настройка резервного узла для SSL соединения

Для настройки резервного узла будет достаточно выполнить следующие действия, описанные в пунктах:

- 7.20.1 «Установка криптопровайдера»;
- 7.20.2 «Установка библиотеки OpenSSL с поддержкой ГОСТ 2012»;
- 7.20.3 «Пакет OpenSSL»;
- 7.20.4 «Настройка конфигурационного файла «openssl\_conf»;
- 7.20.5 «Формирование сертификата»;
- 7.20.6 «Экспорт сформированного сертификата»;
- 7.20.7 «Редактирование сформированного сертификата»;

- 7.20.8 «Установка корневого сертификата УЦ»;
- 7.20.9 «Настройка главного узла для SSL соединения».

На данном этапе конфигурирование кластера для SSL-соединения закончено.

### **7.21. Применение узла с ролью «Referee»**

Сетевое разделение кластера (потеря связности между узлами кластера) приводит к разделению кластера и возникновению нескольких главных узлов с ролью «Master». Это в свою очередь может приводить в итоге к потере данных из-за того, что данные отправляются в оба возникших кластера.

Для устранения потерь данных в кластер может быть добавлен специализированный узел, называемый «Referee».

Узел кластера с ролью «Referee» предназначен для:

- Получения дополнительного значения голосования в процедуре отработки отказа (failover);
- Выполнения проверок и синхронизации состояния кластера.

Узел кластера с ролью «Referee»:

- Выполняет подключение к СУБД главного узла кластера с использованием Public Address, но при этом не управляет СУБД других узлов кластера;
- Периодически опрашивает кластер на предмет состояния кластера и СУБД главного узла с ролью «Master»;

Узел кластера с ролью «Referee» в отличии от других узлов кластера не переходит в режим технического обслуживания и таким образом сохраняет свое состояние в независимости от режимов работы кластера.

Для узла кластера с ролью «Referee» рекомендуется использовать сетевой адрес, отличный от диапазона адресов, применяемого для других узлов кластера.

При активации узла с ролью «Referee» выполняется проверка и синхронизация ролей узлов кластера. Состояние кластера при этом сохраняется в БД главного узла с ролью «Master».

### 7.21.1. Добавление узла с ролью «Referee» с помощью «jadog\_ctl»

Для добавления существующего узла кластера с ролью «Referee» необходимо на главном узле кластера с ролью «Master» подключиться к утилите «jadog\_ctl» и выполнить команду:

```
cluster add referee [ip] [port] as ['node_name']
```

При выполнении указанной команды узлу присваивается имя ['node\_name'].

После добавления узла с ролью «Referee» необходимо выполнить перезагрузку конфигурации кластера с помощью выполнения команды:

```
reload jadog
```

Для проверки статуса узла необходимо воспользоваться встроенным инструментарием консольной утилиты «jadog\_ctl» (см. п.п. 7.22).

Альтернативным вариантом добавления существующего узла кластера с ролью «Referee» является команда:

```
cluster add referee [ip] [port]
```

### 7.21.2. Добавление узла с ролью «Referee» в процессе автоматизированной настройки кластера

В случае автоматизированной настройки кластера узел с ролью «Referee» указывается в шаблоне файла ответов (см. п.п. 3.4).

Для того чтобы реализовать возможность настройки узла с ролью «Referee» необходимо в шаблоне файла ответов «jadog\_make\_ms\_cluster.yml» добавить секцию «cluster\_referee\_nodes» и перечислить список узлов, для которых определяется роль «Referee»:

```
cluster_settings:
cluster_referee_nodes:      # Массив нод, являющимися
referee. Не обязательный параметр.
    - node-referee          # Нода, являющаяся referee.
```

В дальнейшем процедура автоматизированной настройки кластера не отличается от указанной в п.п. 3.4 данного руководства.

После успешного добавления узла с ролью «Referee» в файле состояния /usr/jatoba-6/etc/jadog/jadog\_state.yml для него будет указано новое состояние:

```
ClusterState: Referee
```

### 7.21.3. Удаление узла с ролью «Referee»

Для того чтобы удалить узел кластера с ролью «Referee» необходимо на главном узле кластера с ролью «Master» подключиться к утилите «jadog\_ctl» и выполнить команду:

```
cluster delete node ['node_name']
```

Также удалить узел кластера с ролью «Referee» возможно с указанием его IP-адреса и следующей команды:

```
cluster delete [ip] [port]
```

### 7.22. Получение информации о статусе узлов кластера

Администратору или ИС для выполнения действий с кластером необходимо предварительно убедиться, что узел, к которому он подключен, является мастером. Для получения подобной информации можно воспользоваться консольной утилитой «jadog\_ctl», при помощи которой администратор кластера или ИС может получить информацию о статусе узла.

Информацию о статусе узла возможно получить только подключившись напрямую к узлу.

Узел кластера может принимать следующие статусы:

- MASTER;
- SLAVE;
- PRIMARY.

Помимо статусов узел кластера может находиться в одном из следующих состояний:

- active – узел кластера находится в штатном режиме работы;

- maintenance – узел кластера находится в режиме технического обслуживания (см. п.п. 7.16);
- recovery – узел кластера находится в режиме восстановления после сбоя (см. п.п. 7.17).

Для резервных узлов кластера с использованием каскадной репликации может отображаться информация о режиме репликации:

- sync – узел кластера находится в синхронном режиме репликации (см. п.п. 6.16);
- async – узел кластера находится в асинхронном режиме репликации (см. п.п. 6.15).

Для того чтобы получить при помощи утилиты «jadowctl» информацию о статусе узла необходимо:

- 1) На главном узле кластера в терминале от имени суперпользователя перейти в каталог с консольной утилитой «jadowctl» и выполнить подключение к компоненту «jaDog»:

```
cd /usr/jatoba-6/bin  
./jadowctl
```

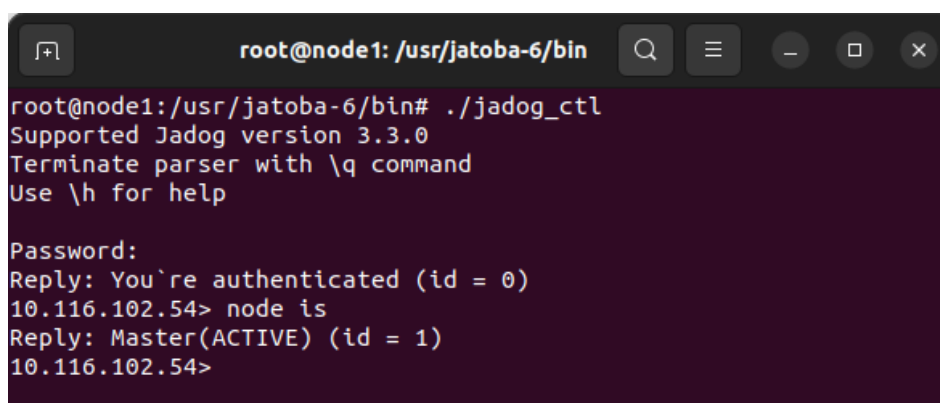
- 2) После ввода пароля пользователя компонента «jaDog» и вывода сообщения «You're authenticated» необходимо выполнить команду вывода информации о статусе узла:

```
node is
```

- 3) Консольная утилита «jadowctl» выведет ответ, например (см. рисунок 7.62):

```
Reply: Master(ACTIVE) (id = 1)
```

В примере узел является главным узлом кластера и находится в штатном состоянии.



```
root@node1: /usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You`re authenticated (id = 0)
10.116.102.54> node is
Reply: Master(ACTIVE) (id = 1)
10.116.102.54>
```

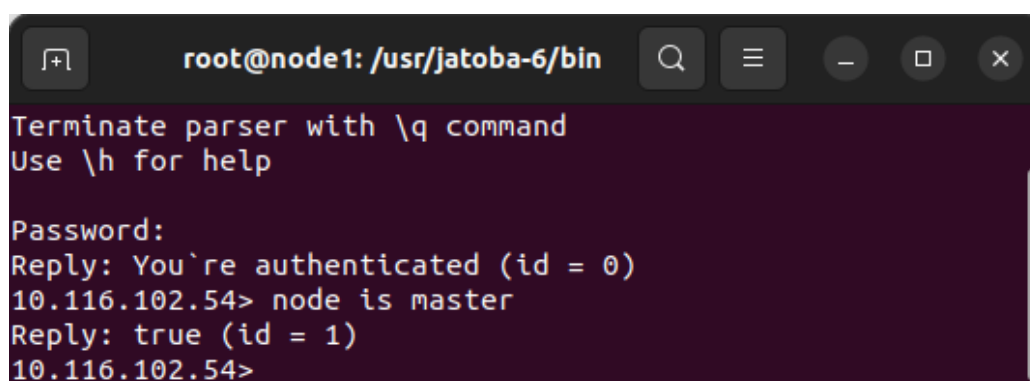
Рисунок 7.62 – Получение информации о статусе узла при помощи консольной утилиты «jadog\_ctl»

Консольная утилита «jadog\_ctl» может возвращать значение статуса узла кластера в виде «true | false». Для получения информации о статусе узла кластера в таком виде применяются следующие команды:

- 1) Если узел кластера является главным с ролью MASTER (обе команды равнозначны):

```
node is m
node is master
```

Будет возвращено значение «Reply: true» (см. рисунок 7.63), в противном случае результат будет «Reply: false».



```
root@node1: /usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You`re authenticated (id = 0)
10.116.102.54> node is master
Reply: true (id = 1)
10.116.102.54>
```

Рисунок 7.63 – Получение информации о роли узла при помощи консольной утилиты «jadog\_ctl»

- 2) Если узел кластера является резервным (обе команды равнозначны):

```
node is s
node is slave
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false».

3) Если узел кластера является ведущим для каскадного типа кластера (обе команды равнозначны):

```
node is p  
node is primary
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false».

4) Если узел кластера является координатором кластера (обе команды равнозначны):

```
node is r  
node is referee
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false».

5) Если резервный узел кластера с каскадной репликацией должен работать в асинхронном режиме:

```
node in async
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false». Для главного узла кластера с каскадной репликацией возвращаемое значение всегда «Reply: false».

6) Если резервный узел кластера с каскадной репликацией должен работать в синхронном режиме:

```
node in sync
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false». Для главного узла кластера с каскадной репликацией возвращаемое значение всегда «Reply: false».

7) Если резервный узел добавлен в кластер с каскадной репликацией:



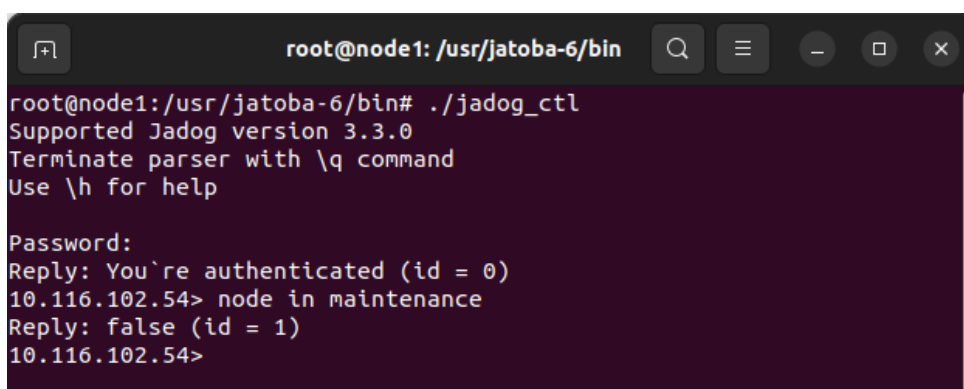
```
node in cascade
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false». Для главного узла кластера с каскадной репликацией возвращаемое значение всегда «Reply: false».

- 8) Если узел находится в режиме технического обслуживания:

```
node in maintenance
```

Будет возвращено значение «Reply: true», в противном случае результат будет «Reply: false» (см. рисунок 7.64).



```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jalogctl
Supported Jalog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> node in maintenance
Reply: false (id = 1)
10.116.102.54>
```

Рисунок 7.64 – Получение информации о режиме технического обслуживания узла при помощи консольной утилиты «jalogctl»

### 7.23. Планировщик заданий кластера

Планировщик заданий кластера (далее по тексту - планировщик) предназначен для автоматизации выполнения операций с узлами кластера при помощи настройки заданий.

Планировщик позволяет выполнять задания с задержкой по таймеру или установить определенное время их выполнения.

Это дает возможность гибко планировать и контролировать выполнение критически важных операций в нужное время и строить несложные последовательности выполнения команд.

Доступ к планировщику осуществляется через интерфейс консольной утилиты «jalogctl» (см. раздел 5 данного руководства).

Примеры использования планировщика:

- Изменение IP-адресов узлов кластера по расписанию в процессе выполнения ТО;
- Запуск/остановка/перезагрузка компонент кластера в определенное администратором время.

### 7.23.1. Ограничения планировщика заданий кластера

Планировщик имеет следующие функциональные ограничения:

- Список команд заданий ограничен командами компонента «jaDog»;
- Запуск заданий планировщика выполняется только на главном (с ролью Master) узле кластера;
- Тип выполнения команд заданий – асинхронный;
- Для заданий возможно указать конкретные дату и время;
- Для заданий возможно указать задержку времени выполнения команды (минимальный 1 минута, максимальный 1 год);
- В планировщик невозможно добавить задание, которое добавляет другое задание;
- Запрет на выполнение команд изменения паролей учетных записей пользователей;
- Запрет на Запрет на выполнение команд создания и/или удаления учетных записей пользователей;

Полный список команд недоступных для использования при создании заданий планировщика приведен в таблице 7.1.

Таблица 7.1 – Список общих команд компонента «jaDog», недоступных для использования в заданиях планировщика

Команда	Описание
cluster status	Отображение списка и состояния узлов кластера
show users	Отображение списка пользователей

create user	Создание пользователя
drop user	Удаление пользователя
alter user	Изменение пароля пользователя
node show	Отображение информации о узле кластера
node ls	Отображение списка узлов кластера
node set	Переименование узла кластера
node is	Отображение актуальной роли узла (главный, резервный)
\password	Изменение пароля пользователя компонента «jaDog»
\h	Справочная информация по командам компонента «jaDog»
connect	Подключение к узлу кластера
get last response	Отображение последнего сообщения с сервера, исключая статус
get result id	Отображение последнего сообщения с сервера по указанному идентификатору, исключая статус
datacenter promote	Принудительное переключение на текущий центр обработки данных
add cluster	Создание кластера
cluster add master	Добавление в кластер главного узла с ролью Master
delete cluster	Удаление кластера
schedule	Отображение списка запланированных задач планировщика
Schedule set	Добавление задания в планировщик

Проверка допустимости выполнения команд производится и при добавлении задания в планировщик и при попытке запуска задания. Результаты отказа в выполнении команды зафиксированы в функциональном логе компонента.

В планировщике реализована очередь выполнения заданий. Задание добавляется в очередь выполнения в двух случаях:

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

- Наступило установленное дата и время задания (см. п.п. 7.23.2.1);
- Истекла задержка выполнения задания (см. п.п. 7.23.2.2).

Если в очереди выполнения планировщика несколько заданий, они выполняются последовательно.



Не рекомендуется добавлять в планировщик несколько заданий с одними и теми же датой и временем выполнения. Чтобы этого избежать перед добавлением нового задания необходимо ознакомиться с списком запланированных заданий (см. п.п. 7.23.3)

Если задание было в очереди планировщика дольше, чем значение параметра `param_scheduler:timer_inaccuracy` (см. п.п. 7.23.5.2), то в этом случае оно удаляется из очереди планировщика.



В начале выполнения каждого задания необходимо в обязательном порядке проверять все условия и ограничения для возможности ее выполнения. Временное окно - одно из таких ограничений. Если последующее задание не попадает в отведенное временное окно - оно отклоняется и удаляется из очереди заданий.

Например, если временное окно выполнения заданий +/- 5 сек и если в планировщик добавлены два задания:

23.00 Задача1

23.00 Задача2

В случае, если Задача1 выполнится быстрее чем за 5 секунд, то у Задачи2 есть шанс запуститься, то есть время выполнения попадает во временное окно. Если Задача1 будет выполняться больше 5 сек, то Задача2 не будет выполнена, так как она не проходит проверку по соответствию временному окну. Со соответствующей записью в логах.

## 7.23.2. Добавление задания в планировщик

### 7.23.2.1 С указанием даты и времени выполнения

Общий синтаксис команды добавления нового задания с датой и временем выполнения в планировщик выглядит следующим образом:

```
schedule set [job_name] [job_time] ["job_command"]
```

Где [job\_name] – уникальное, в рамках кластера, наименование задания для идентификации и управления. Наименование задания должно содержать не более не более 32-х символов без пробелов, без специальных символов; [job\_time] – дата и время в формате TIMESTAMP ('ГГГГ-ММ-ДД ЧЧ:ММ:СС'); [job\_command] – полная команда для выполнения (например "switchover 'node2'").



При совпадении наименования job\_name добавляемого нового задания с уже существующим в планировщике заданием последнее будет обновлено. В этом случае если новое задание содержит новую команду job\_command этот параметр так же будет перезаписан.

### Пример с указанием конкретного времени выполнения задания:

```
schedule set job1 "2025-10-19 10:23:54+02 " "switchover to 'node2' "
```

### 7.23.2.2 С указанием задержки выполнения задания

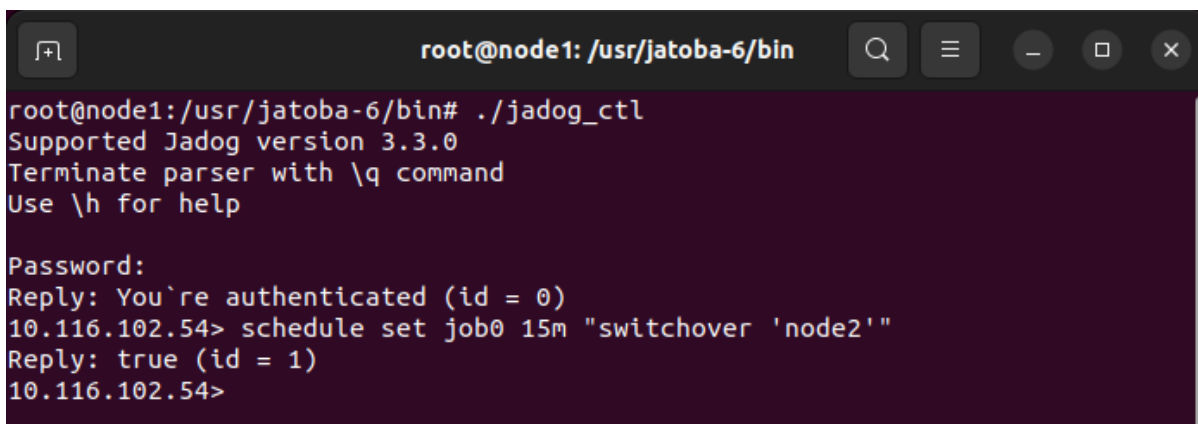
Общий синтаксис команды добавления нового задания с задержкой выполнения в планировщик выглядит следующим образом:

```
schedule set [job_name] [job_delay_interval] ["job_command"]
```

Где [job\_name] – уникальное, в рамках кластера, наименование задания для идентификации и управления. Наименование задания должно содержать не более не более 32-х символов без пробелов, без специальных символов; [job\_delay\_interval] – задержка времени выполнения задания; [job\_command] – полная команда для выполнения (например "switchover 'node2'").

### Пример с указанием времени задержки выполнения задания:

```
schedule set job0 15m "switchover 'node2'"
```



```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> schedule set job0 15m "switchover 'node2'"
Reply: true (id = 1)
10.116.102.54>
```

Рисунок 7.65 – Добавление нового задания в планировщик

### 7.23.3. Получение списка заданий

После добавления в очередь планировщика параметры задания записываются в файл `jadog_shedule.yml`, который расположен в каталоге `/usr/jatoba-6/etc/jadog`.

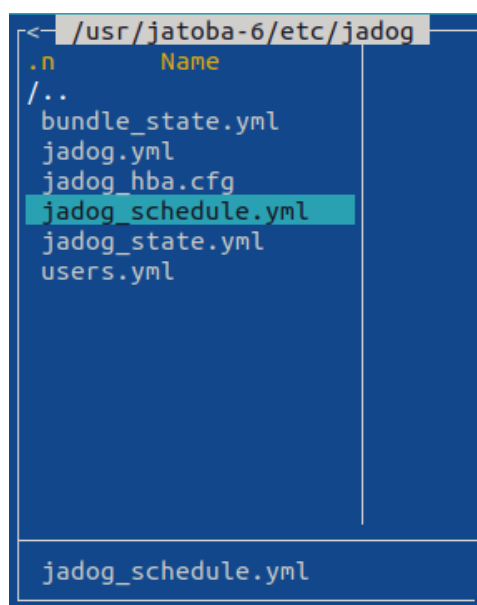


Рисунок 7.66 – Расположение файла `jadog_shedule.yml`

Каждая запись в файле `jadog_shedule.yml` относится к одному заданию.



В случае обнаружения ошибок в параметрах запускаемого задания необходимо отредактировать его параметры в конфигурационном файле `jadog_shedule.yml`, и перезагрузить конфигурацию при помощи консольной утилиты `jadog_ctl`:

```
reload jadog
```

Файл jadog\_schedule.yml реплицируется с главного узла (с ролью Master) на все узлы кластера. Таким образом при выполнении процедуры смены главного узла задания планировщика продолжают выполняться с нового главного узла.

Общий синтаксис команды отображения списка запланированных задач планировщика выглядит следующим образом:

```
schedule
```

В результате выполнения команды будет выведена следующая информация:

- job\_id – идентификатор задания (ID);
- job\_name – наименование задания;
- job\_time – дата время или временной интервал выполнения задания;
- job\_command – команда задания с параметрами к выполнению.



```
root@node1: /usr/jatoba-6/bin
root@node1:usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> schedule
Reply: (id = 1)
+-----+-----+-----+-----+
|job_id|job_name|job_time          |job_command      |
+-----+-----+-----+-----+
|1     |job000  |2025-05-20 17:43:29+03:00|switchover 'node2'|
+-----+-----+-----+-----+

10.116.102.54>
```

Рисунок 7.67 – Список заданий планировщика

Для того чтобы просмотреть выполненные ранее задания необходимо выполнить следующую команду:

```
schedule completed
```

Данная команда выводит список последних 10 выполненных заданий с результатом выполнения команд компонента «jaDog».

```

root@node1: /usr/jatoba-6/bin
root@node1: /usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> schedule completed
Reply: (id = 1)
+-----+-----+-----+-----+-----+
|job_id|job_name|job_time          |job_command          |job_result          |
+-----+-----+-----+-----+-----+
|0      |job0     |2025-05-20 16:21:59+03:00|switchover 'node2'|findAddrByNodeName Node with name: node2 doesn't found|
+-----+-----+-----+-----+-----+
|0      |job00    |2025-05-20 16:48:29+03:00|switchover 'node2'|findAddrByNodeName Node with name: node2 doesn't found|
+-----+-----+-----+-----+-----+
10.116.102.54>

```

Рисунок 7.68 – Список выполненных заданий планировщика



При выполнении перезагрузки сервисов компонента «jaDog» список выполненных заданий очищается.

#### 7.23.4. Удаление задания из планировщика

Для того чтобы удалить задание из планировщика необходимо его наименование (см. п.п. 7.23.3).

Общий синтаксис команды удаления задания из планировщика выглядит следующим образом:

```
schedule reset [job_name]
```

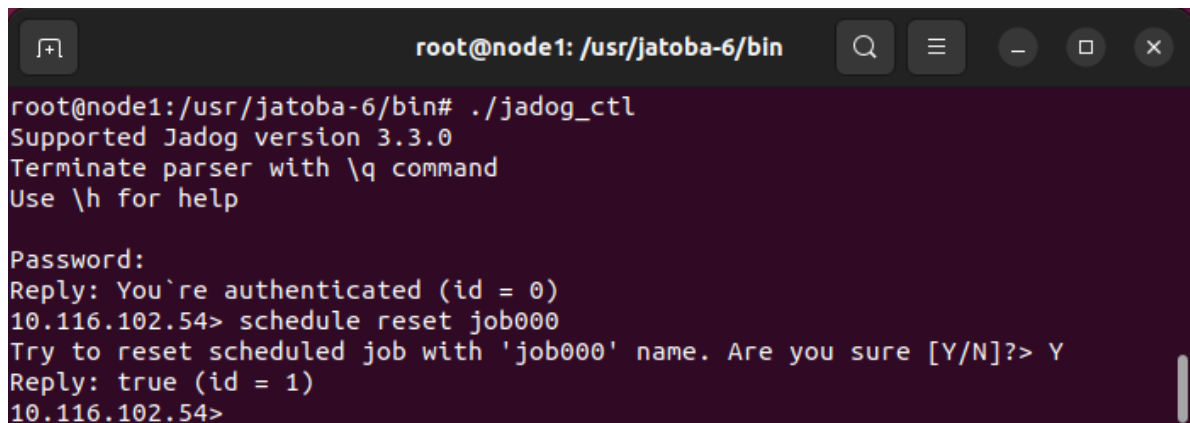
Где [job\_name] – уникальное, в рамках кластера, наименование задания для идентификации и управления

#### Пример:

```
schedule reset job000
```

При удалении задания из планировщика пользователю выводится сообщение «Try to reset scheduled job with 'job\_name' name. Are you sure [Y/N]?>»





```
root@node1: /usr/jatoba-6/bin
root@node1:/usr/jatoba-6/bin# ./jadog_ctl
Supported Jadog version 3.3.0
Terminate parser with \q command
Use \h for help

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> schedule reset job000
Try to reset scheduled job with 'job000' name. Are you sure [Y/N]?> Y
Reply: true (id = 1)
10.116.102.54>
```

Рисунок 7.69 – Удаление задания из планировщика

### 7.23.5. Параметры планировщика

Параметры планировщика располагаются в конфигурационном файле jadog.yml.

Изменение параметров допускается с использованием консольной утилиты «jadog\_ctl» (см. п.п. 7.14.2).

Примеры команд изменения параметров планировщика приведены в таблице П.1.1.

#### 7.23.5.1 param\_scheduler:timer\_interval

Интервал внутреннего таймера на опрос текущего времени и времени выполнения, мс.

Значение по умолчанию – 1000.

#### 7.23.5.2 param\_scheduler:timer\_inaccuracy

Погрешность отставания времени выполнения от фактически установленного времени, мс.

Значение по умолчанию – 5000.

#### 7.23.5.3 param\_scheduler:max\_jobs\_count

Максимальное количество задач в очереди планировщика.

Значение по умолчанию – 8.

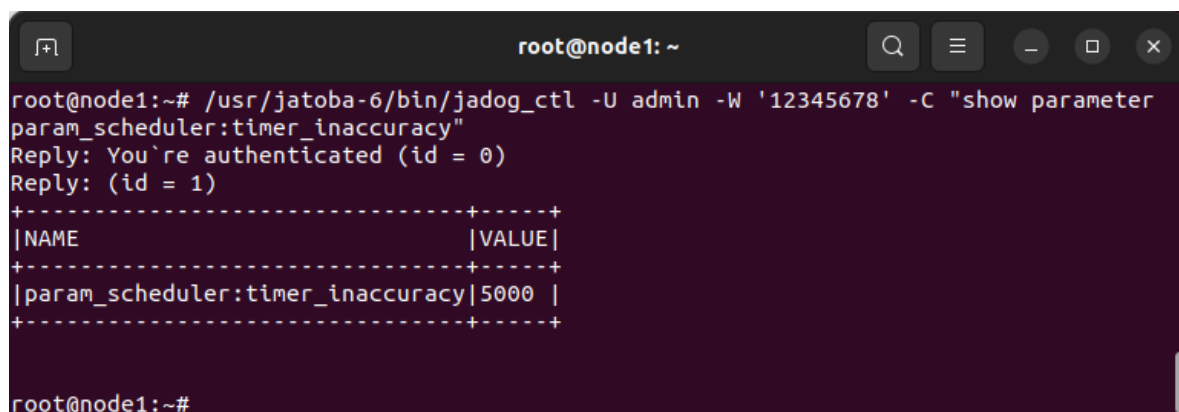
#### 7.23.5.4 Получение значений параметров планировщика заданий

Для того чтобы получить значение параметра планировщика применяется команда с следующим синтаксисом:

```
/usr/jatoba-6/bin/jadog_ctl -U admin -W '[пароль УЗ admin]' -C  
"show parameter param_scheduler:[название_параметра]"
```

### Пример:

```
/usr/jatoba-6/bin/jadog_ctl -U admin -W '12345678' -C "show  
parameter param_scheduler:timer_inaccuracy"
```



```
root@node1: ~  
root@node1:~# /usr/jatoba-6/bin/jadog_ctl -U admin -W '12345678' -C "show parameter  
param_scheduler:timer_inaccuracy"  
Reply: You're authenticated (id = 0)  
Reply: (id = 1)  
+-----+-----+  
|NAME                                |VALUE|  
+-----+-----+  
|param_scheduler:timer_inaccuracy|5000 |  
+-----+-----+  
root@node1:~#
```

Рисунок 7.70 – Получение значений параметров планировщика заданий

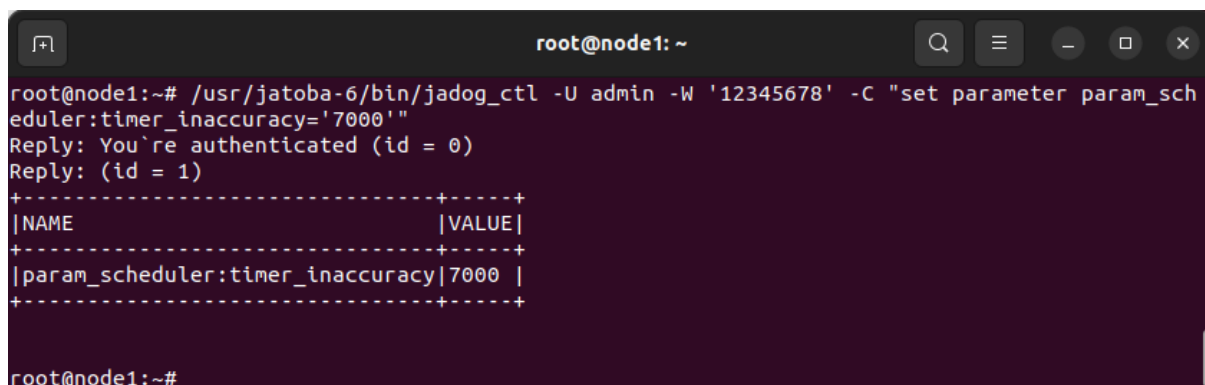
### 7.23.5.5 Изменение параметров планировщика заданий

Для того чтобы изменить значение параметра планировщика применяется команда с следующим синтаксисом:

```
/usr/jatoba-6/bin/jadog_ctl -U admin -W '[пароль УЗ admin]' -C  
"set parameter param_scheduler:[название_параметра]"
```

### Пример:

```
/usr/jatoba-6/bin/jadog_ctl -U admin -W '12345678' -C "set  
parameter param_scheduler:timer_inaccuracy='7000'"
```



```
root@node1: ~  
root@node1:~# /usr/jatoba-6/bin/jadog_ctl -U admin -W '12345678' -C "set parameter param_sch  
eduler:timer_inaccuracy='7000'"  
Reply: You're authenticated (id = 0)  
Reply: (id = 1)  
+-----+-----+  
|NAME                                |VALUE|  
+-----+-----+  
|param_scheduler:timer_inaccuracy|7000 |  
+-----+-----+  
root@node1:~#
```

Рисунок 7.71 – Изменение значений параметров планировщика заданий

## 7.24. Синхронизация файлов в кластере

Компонент «jaDog» предоставляет функционал автоматизированной доставки на узлы кластера файлов, необходимых для функционирования кластера (сертификаты TLS, конфигурационные файлы, файлы ответов и так далее).

Формирование кластеров и бандлов по файлам ответов («тихое» формирование) требует установки цифровых сертификатов на узлы, участвующие в развертывании. На текущий момент эта функция должна выполняться администратором, как предусловие, в ручном режиме.

Автоматизированное распространение позволяет размещать сертификаты и необходимые конфигурационные файлы вовремя развертывания системы с минимальным количеством ручных действий, что позволяет значительно упростить процесс и минимизировать участие администратора в больших конфигурациях.

Настройка параметров синхронизации файлов в кластере выполняется через внесение изменений в конфигурационный файл `file_manager.yml`.



Конфигурационный файл `file_manager.yml` располагается в каталоге `/usr/jatoba-6/etc/jadog`

### 7.24.1. Ограничения менеджера синхронизации файлов

Менеджер синхронизации файлов имеет следующие функциональные ограничения:

- Синхронизация файлов и/или каталогов выполняется в направлении с главного узла на резервные/referee. В случае смены главного узла в кластере выполнения синхронизации производится с нового главного узла. В случае отсутствия на новом главном узле директорий, указанных в списке синхронизации, данные записи будут удалены.
- Синхронизация файлов и/или каталогов для узлов кластера с ролью Referee (см. п.п. 7.21) выполняется только при использовании соответствующего ключа.
- Формирование записей в файле `file_manager.yml` – автоматизированное.
- Максимальный размер синхронизируемых файлов – 10 Мб.
- Максимальное количество записей в файле `file_manager.yml` – не более 32 (см. п.п. 7.24.8.2). В случае ручного изменения файла `file_manager.yml` лишние записи при выполнении синхронизации будут удалены.

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

- При добавлении новые записи записываются в конец списка.
- При достижении в файле файл `file_manager.yml` максимального количества записей (см. п.п. 7.24.8.2) добавление новой записи невозможно.
- В качестве записей для синхронизации допускается указывать файлы.
- В качестве записей для синхронизации допускается указывать каталоги.
- Расположение файлов и каталогов узла-источника при выполнении синхронизации совпадает с расположением файлов и каталогов на узле-получателе. Администратор при необходимости создает на узлах все необходимые каталоги для синхронизации файлов.
- Атрибуты и права доступа к файлам и каталогам при выполнении синхронизации копируются без изменений.
- Не выполняется синхронизация файлов и/или каталогов между узлами кластера в том случае, если атрибуты или права доступа не позволяют завершить синхронизацию. Информация об этом записывается в журнал информационной безопасности.
- При выполнении синхронизации имеющиеся в каталогах файлы будут заменены новыми.

### 7.24.2. Настройка синхронизации файлов в кластере

Файл `file_manager.yml` располагается в каталоге `/usr/jatoba-6/etc/jadog` и содержит список файлов, которые необходимо синхронизировать между узлами кластера при помощи файловой синхронизации.

Внесение изменений в конфигурационный файл `file_manager.yml` отражается записями в журнале информационной безопасности.



При отсутствии в каталоге `/usr/jatoba-6/etc/jadog` конфигурационного файла `file_manager.yml` при использовании консольной утилиты `«jadog_ctl»` выводится сообщение `«File manager configuration file does not exist»`.

В этом случае необходимо вручную создать конфигурационный файл:

```
# touch /usr/jatoba-6/etc/jadog/file_manager.yml
```

и выполнить перезагрузку сервиса компонента `«jaDog»`:

```
# systemctl restart jadog
```

Перезагрузку компонента «jaDog» следует проводить в установленные администратором СУБД технологические окна обслуживания кластера. Перед выполнением операции кластер нужно перевести в режим maintenance (см. п.п. 7.16).

В случае отсутствия настроек параметров в конфигурационном файле `file_manager.yml`, при использовании для синхронизации консольной утилиты «`jadog_ctl`», выводится сообщение «Synchronization file list is empty».

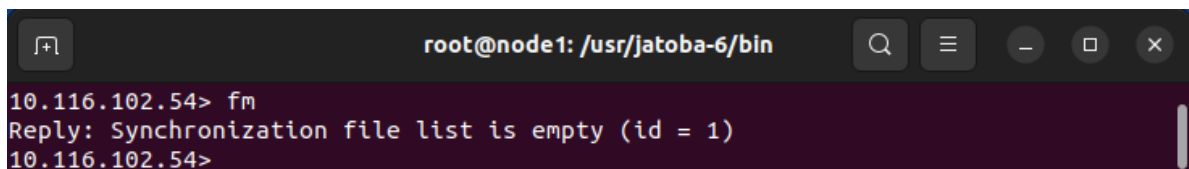


Рисунок 7.72 – Сообщение об отсутствии параметров в конфигурационном файле `file_manager.yml`

### 7.24.3. Добавление нового файла(ов) в список синхронизации

Для того чтобы добавить файлы, которые требуется синхронизировать между узлами кластера, необходимо на главном узле кластера воспользоваться командой, имеющей следующий синтаксис:

```
fm set '[path/to/file.cfg]'
```

где `path/to/file.cfg` – путь к каталогу с файлом.

#### Пример:

```
fm set /var/lib/jatoba/certs/srv.crt
```



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

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> fm set /var/lib/jatoba/certs/srv.crt
Reply: Add path to sync files: /var/lib/jatoba/certs/srv.crt (id = 1)
10.116.102.54>
```

Рисунок 7.73 – Добавление нового файла для синхронизации

После успешного выполнения команды в файл `file_manager.yml` добавляется строка с номером в списке синхронизации (см. 7.24.5).

#### 7.24.4. Добавление нового каталога для синхронизации

Для того чтобы добавить каталоги, которые требуется синхронизировать между узлами кластера, необходимо на главном узле кластера воспользоваться командой, имеющей следующий синтаксис:

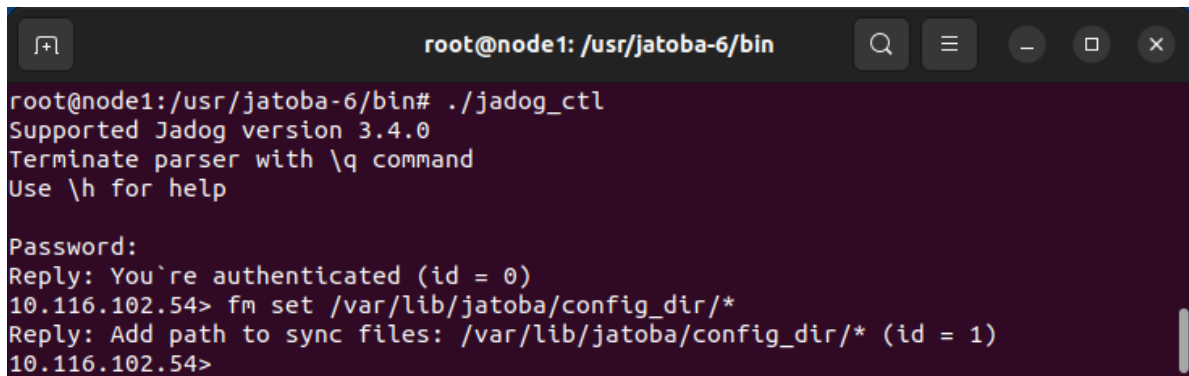
```
fm set '[path/to/folder/*]'
```

#### Пример:

```
fm set /var/lib/jatoba/config_dir/*
```

При указании каталога, копируются только файлы входящие в него. Подкаталоги в синхронизации файлов не участвуют. Для того чтобы добавить файлы из подкаталогов необходимо добавить их при помощи команды `fm`, например:

```
fm set /var/lib/jatoba/root_folder/*
fm set /var/lib/jatoba/root_folder/sub_folfer1/*
fm set /var/lib/jatoba/root_folder/sub_folfer2/*
```



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

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> fm set /var/lib/jatoba/config_dir/*
Reply: Add path to sync files: /var/lib/jatoba/config_dir/* (id = 1)
10.116.102.54>
```

Рисунок 7.74 – Добавление нового каталога для синхронизации

После успешного выполнения команды в файл `file_manager.yml` добавляется строка с номером в списке синхронизации (см. 7.24.5).

#### 7.24.5. Просмотр списка файлов/каталогов для синхронизации

После добавления файлов и/или каталогов в конфигурационном файле `file_manager.yml` формируется список, содержащий следующие данные:

- Номер записи;
- Путь в файловой системе к каталогу или файлу.

Для того чтобы отобразить список файлов и/или каталогов, для которых выполняется синхронизация в кластере, необходимо на главном узле кластера выполнить следующую команду:

```
fm
```

```

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> fm
Reply: (id = 1)
+-----+
|Id|Path|
+-----+
|1|/var/lib/jatoba/certs/srv.crt|
+-----+
|2|/var/lib/jatoba/.pgpass|
+-----+
|3|/var/lib/jatoba/config_dir/logs.conf|
+-----+
|4|/var/lib/jatoba/config_dir/*|
+-----+

10.116.102.54>

```

Рисунок 7.75 – Просмотр списка файлов и/или каталогов для синхронизации

## 7.24.6. Выполнение синхронизации файлов и каталогов на узлах кластера

### 7.24.6.1 Синхронизация на всех узлах кластера

Для того чтобы выполнить синхронизацию всех файлов или каталогов с резервными узлами (кроме узлов с ролью «Referee») необходимо на главном узле кластера выполнить следующую команду:

```
fm copy
```

В случае если в составе кластера отсутствуют резервные узлы, доступные для синхронизации при выполнении в консольной утилите «jadog\_ctl» будет выведено сообщение: «There are no nodes to sync».

```

root@node1: /usr/jatoba-6/bin
10.116.102.54> fm copy
Reply: Error: There are no nodes to sync (id = 2)
10.116.102.54>

```

Рисунок 7.76 – Сообщение об отсутствии резервных узлов, доступных для синхронизации



#### 7.24.6.2 Синхронизация всех файлов с определенным узлом или группой узлов

Для того чтобы выполнить синхронизацию всех файлов или каталогов с определенным резервным узлом (кроме узлов с ролью «Referee») необходимо на главном узле кластера выполнить следующую команду, имеющую синтаксис:

```
fm copy -n {node name array}
```

где node name array – перечень резервных узлов через пробел (можно указать как один, так и несколько узлов для синхронизации файлов). Наименование резервного узла отображается при отображении статусов узлов кластера (см. п.п. 6.13).

#### Пример:

```
fm copy -n cluster1_node2 cluster1_node3
```

#### 7.24.6.3 Синхронизация файлов с узлом Referee

Для того чтобы выполнить синхронизацию всех файлов или каталогов с узлами кластера, включая узлы с ролью «Referee», необходимо на главном узле кластера выполнить следующую команду:

```
fm copy -referee
```

#### 7.24.6.4 Синхронизация файла/каталога на всех узлах кластера

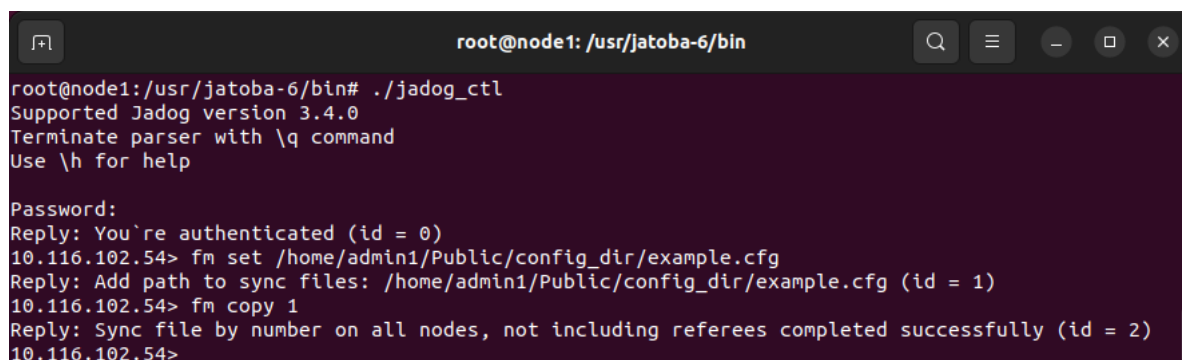
Для того чтобы выполнить синхронизацию определенных файлов или каталогов с резервными узлами (кроме узлов с ролью «Referee») необходимо на главном узле кластера выполнить следующую команду:

```
fm copy {file id}
```

где file id – идентификатор (номер) строки в списке файла file\_manager.yml.

#### Пример:

```
fm copy 1
```



```
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> fm set /home/admin1/Public/config_dir/example.cfg
Reply: Add path to sync files: /home/admin1/Public/config_dir/example.cfg (id = 1)
10.116.102.54> fm copy 1
Reply: Sync file by number on all nodes, not including referees completed successfully (id = 2)
10.116.102.54>
```

Рисунок 7.77 – Синхронизация файла со всеми узлами кластера

#### 7.24.6.5 Синхронизация файла/каталога с определенным узлом

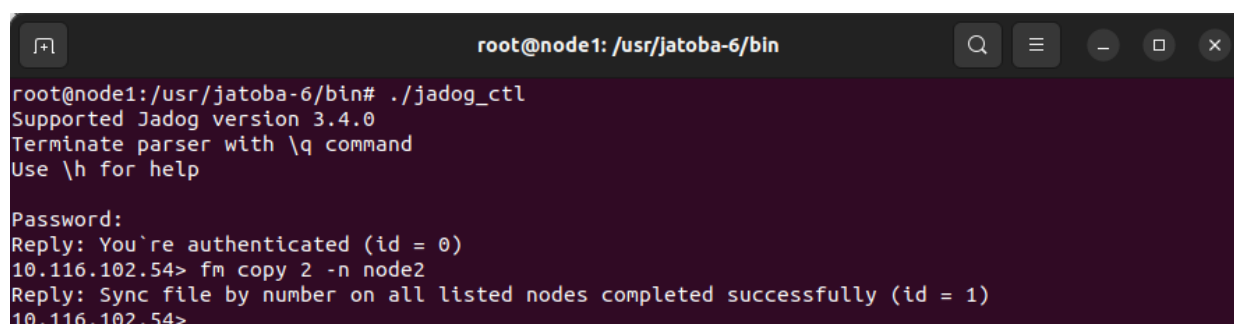
Для того чтобы выполнить синхронизацию определенных файлов или каталогов с определенным резервным узлом (кроме узлов с ролью «Referee») необходимо на главном узле кластера выполнить следующую команду:

```
fm copy {file id} -n {node name array}
```

где file id – идентификатор (номер) строки в списке файла file\_manager.yml, node name array – наименование резервного узла в кластере. Наименование резервного узла отображается при отображении статусов узлов кластера (см. п.п. 6.13).

#### Пример:

```
fm copy 2 -n node2
```



```
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> fm copy 2 -n node2
Reply: Sync file by number on all listed nodes completed successfully (id = 1)
10.116.102.54>
```

Рисунок 7.78 – Синхронизация определенного файла с конкретным узлом кластера

#### 7.24.6.6 Синхронизация файла/каталога с узлом Referee

Для того чтобы выполнить синхронизацию определенных файлов или каталогов с узлами кластера, включая узлы с ролью «Referee», необходимо на главном узле кластера выполнить следующую команду:

```
fm copy {file id} --referee
```

где file id – идентификатор (номер) строки в списке файла file\_manager.yml

**Пример:**

```
fm copy 3 --referee
```

### 7.24.7. Удаление файлов/каталогов из списка синхронизации

#### 7.24.7.1 Удаление всех записей синхронизации файлов

Для того чтобы очистить список в файле file\_manager.yml, содержащий информацию о файлах и/или каталогах, необходимо выполнить следующую команду:

```
fm delete
```

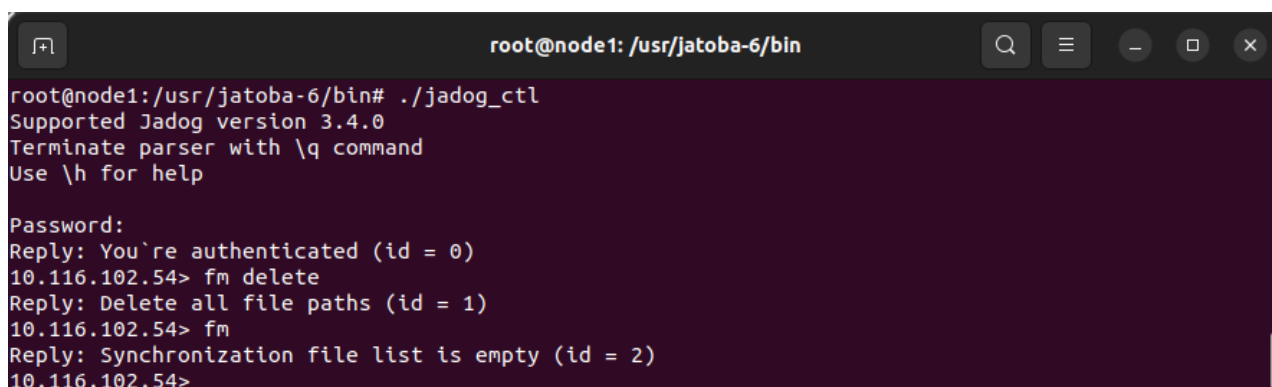


Рисунок 7.79 – Удаление всех записей синхронизации файлов и просмотр текущего списка записей (пусто)

#### 7.24.7.2 Удаление одной записи

Для того чтобы удалить одну запись из списка в файле file\_manager.yml, содержащего информацию о файлах и/или каталогах, необходимо выполнить следующую команду:

```
fm delete {file id}
```

где file id – идентификатор (номер) строки в списке файла file\_manager.yml

**Пример:**

```
fm delete 2
```

```

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

Password:
Reply: You're authenticated (id = 0)
10.116.102.54> fm delete 2
Reply: Delete file path with id: 2 (id = 1)
10.116.102.54>

```

Рисунок 7.80 – Удаление записи из списка синхронизации

```

root@node1: /usr/jatoba-6/bin
10.116.102.54> fm
Reply: (id = 2)
+-----+
|Id|Path|
+-----+
|1 | /var/lib/jatoba/certs/srv.crt |
+-----+
|3 | /var/lib/jatoba/config_dir/logs.conf|
+-----+
|4 | /var/lib/jatoba/config_dir/* |
+-----+
10.116.102.54>

```

Рисунок 7.81 – Результат удаления записи №2 из списка синхронизации

После удаления список идентификаторов записей синхронизации не изменяет нумерацию.

### 7.24.8. Параметры синхронизации файлов в кластере

Параметры синхронизации файлов в кластере располагаются в конфигурационном файле `jalog.yml`.

Изменение параметров допускается с использованием консольной утилиты «`jalogctl`» (см. п.п. 7.14.2).

Примеры команд изменения параметров синхронизации файлов в кластере приведены в таблице П.1.1.

#### 7.24.8.1 `file_manager:max_file_size`

Максимальный размер синхронизируемых файлов в мегабайтах. В случае превышения размера синхронизируемого файла, чем указанное значение в параметре

max\_file\_size, в журнал компонента записывается ошибка. При этом данный файл не синхронизируется, а работа по синхронизации продолжается.

Значение по умолчанию – 1.

#### **7.24.8.2 file\_manager: max\_files\_per\_session**

Максимальное количество синхронизируемых файлов и каталогов.

Значение по умолчанию – 32.

### **7.25. Запуск пользовательских сценариев при смене роли узла**

Механизм выполнения предварительных (pre) и завершающих (post) сценариев предназначен для автоматизации действий администратора при изменении состояния узлов отказоустойчивого кластера. Например предварительные и завершающие сценарии могут запускаться при смене роли узла с главного на резервный или наоборот.

Основные сценарии использования:

- Активация/деактивация дополнительного ПО при смене роли узла;
- Уведомление администратора о критических событиях (например, переходе узла в режим главного);
- Настройка компонентов СУБД «Jatoba», требующих ручного вмешательства при смене роли.

#### **7.25.1. Типы сценариев**

Типы сценариев, запускаемых компонентом «jaDog» при изменении состояния узлов указаны в таблице

Таблица 7.2 – Типы сценариев

Тип скрипта	Момент выполнения	Пример использования
Pre-скрипт	До выполнения основной команды (например, до promote)	Проверка готовности узла к повышению роли до главного
Post-скрипт	После выполнения основной команды (например, после promote)	Запуск служб на новом главном узле

#### **7.25.2. Поддерживаемые события**

При выполнении сценариев поддерживаются следующие события:

- Смена роли (promote, demote);

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

- Остановка/перезапуск кластера (stop, restart);
- Включение/отключение режима обслуживания (maintenance).

Таблица 7.3 – Типы событий выполнения сценариев

Тип события	Префикс названия сценария
Смена роли узла до главного	promote
Синхронизация	rewind
Смена роли узла на резервный	demote

### 7.25.3. Ограничения при запуске сценариев

При выполнении сценариев существуют следующие ограничения:

- Исполняемые файлы сценариев должны быть только \*.sh, \*.bat (SQL-скрипты не поддерживаются);
- Пути к сценариям указываются в локальной файловой системе;
- Компонент «jaDog» не контролирует содержимое сценариев – ответственность за их работоспособность лежит на администраторе кластера;
- Ошибка при выполнении сценария (ошибки при обработке исполняемых файлов \*.sh, \*.bat) не отменяет основную команду смены роли узла кластера.

### 7.25.4. Предварительные требования запуска сценариев

- Сценарии должны быть размещены на всех узлах кластера в идентичных путях;
- Права на выполнение сценариев должны быть назначены вручную (компонент «jaDog» использует учетную запись с правами, достаточными для запуска).

### 7.25.5. Алгоритм запуска и выполнения сценариев

Алгоритм запуска и выполнения сценариев состоит из следующих шагов:

- 1) Получение команды от компонента «jaDog» (например, promote).
- 2) Выполнение предварительного (pre) скрипта (если указан в файле jadog.yml).  
Результат (успех/ошибка) записывается в журнал работы компонента «jaDog».
- 3) Выполнение основной команды (например, смена роли узла на главный (Master)).
- 4) Выполнение завершающего (Post) скрипта (если указан). Результат (успех/ошибка) записывается в журнал работы компонента «jaDog».

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

### 7.25.6. Настройка конфигурации запуска сценариев

Пути к сценариям указываются в конфигурационном файле `/usr/jatoba-6/etc/jadog/jadog.yml`.

В конфигурационном файле `jadog.yml` располагается секция `scenarios`, предназначенная для указания пути к исполняемым файлам скриптов.

```
scenarios:
  events:promote:pre_script: /usr/jatoba-
6/share/jadog/event_scripts/pre_promote.sh # Предварительный
сценарий при смене роли узла до главного
  events:promote:post_script: /usr/jatoba-
6/share/jadog/event_scripts/post_promote.sh Завершающий
сценарий при смене роли узла до главного
  events:demote:pre_script: /usr/jatoba-
6/share/jadog/event_scripts/pre_demote.sh Предварительный
сценарий при смене роли узла до резервного
  events:demote:post_script: /usr/jatoba-
6/share/jadog/event_scripts/post_demote.sh Завершающий сценарий
при смене роли узла до резервного
  events:rewind:pre_script: /usr/jatoba-
6/share/jadog/event_scripts/pre_rewind.sh # Предварительный
сценарий при выполнении синхронизации узлов кластера
  events:rewind: /usr/jatoba-
6/share/jadog/event_scripts/post_rewind.sh # Завершающий
сценарий при выполнении синхронизации узлов кластера
  script_timeout: 3000 # Задержка выполнения сценария, мс
```

#### 7.25.6.1 promote:pre\_script

Предварительный сценарий при смене роли узла до главного.

Команда настройки параметра `promote:pre_script`:

```
set parameter scenarios:events:promote:pre_script =
"/path/to/script_pre_promote.sh"
```

The screenshot shows two windows. The top window is a terminal with the following content:

```
root@node1: /usr/jatoba-6/bin
10.116.102.54> set parameter scenarios:events:promote:pre_script = "/home/admin1/jatoba.sh"
Set parameter 'scenarios:events:promote:pre_script' to value '/home/admin1/jatoba.sh'
Are you sure [Y/N]?> y
Reply: (id = 1)
+-----+-----+
|NAME                                     |VALUE                                     |
+-----+-----+
|scenarios:events:promote:pre_script|/home/admin1/jatoba.sh|
+-----+-----+
```

The bottom window is a nano editor showing the file `/usr/jatoba-6/etc/jadog/jadog.yml`. The content is:

```
scenarios:
  events:promote:pre_script: /home/admin1/jatoba.sh
  events:promote:post_script: ""
  events:demote:pre_script: ""
  events:demote:post_script: ""
  events:rewind:pre_script: ""
  events:rewind:post_script: ""
  script_timeout: 3000
```

Рисунок 7.82 – Настройка параметра promote:pre\_script

#### 7.25.6.2 promote:post\_script

Завершающий сценарий при смене роли узла до главного.

Команда настройки параметра promote:post\_script:

```
set parameter scenarios:events:promote:post_script =
"/path/to/script_post_promote.sh"
```

The screenshot shows two windows. The top window is a terminal with the following content:

```
root@node1: /usr/jatoba-6/bin
10.116.102.54> set parameter scenarios:events:promote:post_script = "/home/admin1/jadog_script_post_promote.sh"
Set parameter 'scenarios:events:promote:post_script' to value '/home/admin1/jadog_script_post_promote.sh'
Are you sure [Y/N]?> y
Reply: (id = 2)
+-----+-----+
|NAME                                     |VALUE                                     |
+-----+-----+
|scenarios:events:promote:post_script|/home/admin1/jadog_script_post_promote.sh|
+-----+-----+
```

The bottom window is a nano editor showing the file `/usr/jatoba-6/etc/jadog/jadog.yml`. The content is:

```
scenarios:
  events:promote:pre_script: /home/admin1/jatoba.sh
  events:promote:post_script: /home/admin1/jadog_script_post_promote.sh
  events:demote:pre_script: ""
  events:demote:post_script: ""
  events:rewind:pre_script: ""
  events:rewind:post_script: ""
  script_timeout: 3000
```

Рисунок 7.83 – Настройка параметра promote:post\_script



### 7.25.6.3 demote:pre\_script

Предварительный сценарий при смене роли узла до резервного.

Команда настройки параметра demote:pre\_script:

```
set parameter scenarios:events:demote:pre_script =  
"/path/to/script_pre_demote.sh"
```

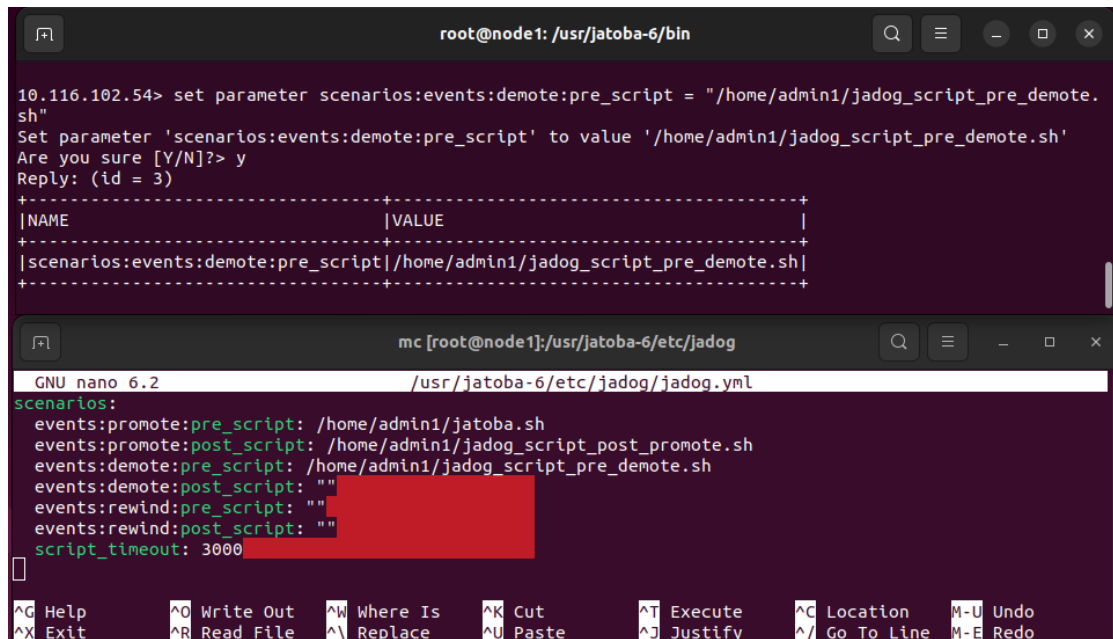


Рисунок 7.84 – Настройка параметра demote:pre\_script

### 7.25.6.4 demote:post\_script

Завершающий сценарий при смене роли узла до резервного.

Команда настройки параметра demote:post\_script:

```
set parameter scenarios:events:demote:post_script =  
"/path/to/script_post_demote.sh"
```

The screenshot shows two windows. The top window is a terminal with the prompt 'root@node1: /usr/jatoba-6/bin'. It shows the command 'set parameter scenarios:events:demote:post\_script = "/home/admin1/jadog\_script\_post\_demote.sh"' being entered. The prompt asks 'Are you sure [Y/N]?> y' and the user replies 'y'. The prompt then shows 'Reply: (id = 4)'. Below this, a table shows the configuration change:

NAME	VALUE
scenarios:events:demote:post_script	/home/admin1/jadog_script_post_demote.sh

The bottom window is a nano editor showing the file '/usr/jatoba-6/etc/jadog/jadog.yml'. The content of the file is:

```
scenarios:
  events:promote:pre_script: /home/admin1/jatoba.sh
  events:promote:post_script: /home/admin1/jadog_script_post_promote.sh
  events:demote:pre_script: /home/admin1/jadog_script_pre_demote.sh
  events:demote:post_script: /home/admin1/jadog_script_post_demote.sh
  events:rewind:pre_script: ""
  events:rewind:post_script: ""
  script_timeout: 3000
```

Рисунок 7.85 – Настройка параметра demote:post\_script

#### 7.25.6.5 rewind:pre\_script

Предварительный сценарий при выполнении синхронизации узлов кластера.

Команда настройки параметра rewind:pre\_script:

```
set parameter scenarios:events:rewind:pre_script =
"/path/to/script_pre_rewind.sh"
```

The screenshot shows two windows. The top window is a terminal with the prompt 'root@node1: /usr/jatoba-6/bin'. It shows the command 'set parameter scenarios:events:rewind:pre\_script = "/home/admin1/jadog\_script\_pre\_rewind.sh"' being entered. The prompt asks 'Are you sure [Y/N]?> y' and the user replies 'y'. The prompt then shows 'Reply: (id = 5)'. Below this, a table shows the configuration change:

NAME	VALUE
scenarios:events:rewind:pre_script	/home/admin1/jadog_script_pre_rewind.sh

The bottom window is a nano editor showing the file '/usr/jatoba-6/etc/jadog/jadog.yml'. The content of the file is:

```
scenarios:
  events:promote:pre_script: /home/admin1/jatoba.sh
  events:promote:post_script: /home/admin1/jadog_script_post_promote.sh
  events:demote:pre_script: /home/admin1/jadog_script_pre_demote.sh
  events:demote:post_script: /home/admin1/jadog_script_post_demote.sh
  events:rewind:pre_script: /home/admin1/jadog_script_pre_rewind.sh
  events:rewind:post_script: ""
  script_timeout: 3000
```

Рисунок 7.86 – Настройка параметра rewind:pre\_script

### 7.25.6.6 rewind:post\_script

Завершающий сценарий при выполнении синхронизации узлов кластера.

Команда настройки параметра `rewind:post_script`:

```
set parameter scenarios:events:rewind:post_script =  
"/path/to/script_post_rewind.sh"
```

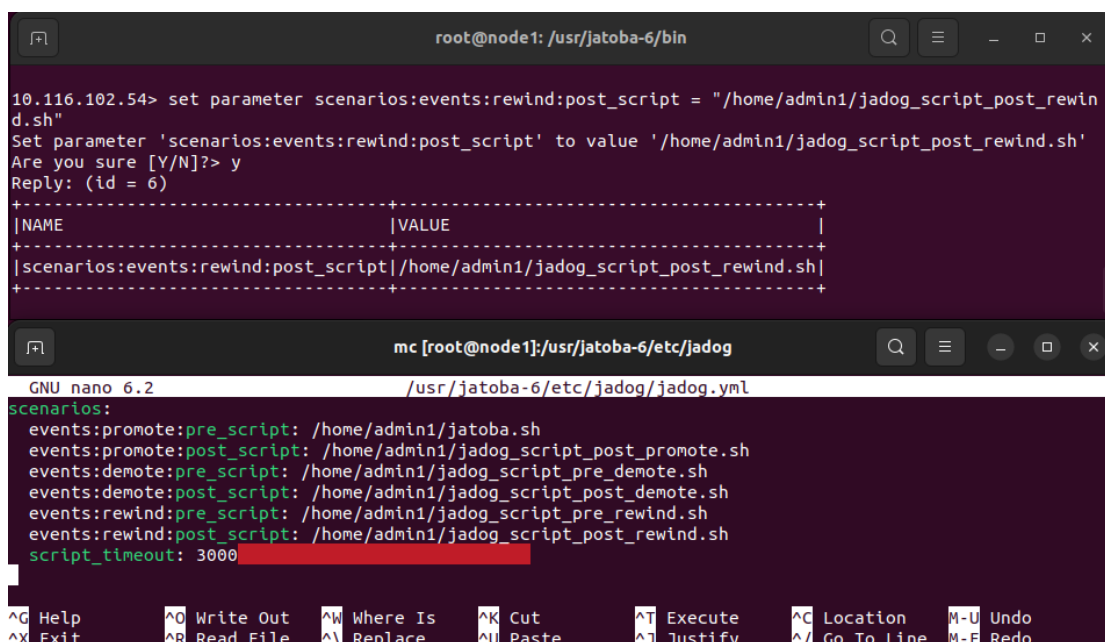


Рисунок 7.87 – Настройка параметра `rewind:post_script`

### 7.25.6.7 script\_timeout

Задержка выполнения сценариев в миллисекундах.

Команда настройки параметра `script_timeout`:

```
set parameter scenarios:script_timeout = "4000"
```

### 7.25.6.8 Ручное изменение параметров сценариев

При ручном изменении конфигурационного файла `jadog.yml` необходимо выполнить перезагрузку компонента «jaDog» при помощи команды в консольной утилите `jadog_ctl`:

```
reload jadog
```

### 7.25.7. Проверка работоспособности сценариев

Проверка работоспособности сценариев может быть выполнена в следующем порядке:

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

- 1) Разместить тестовый скрипт (например, `post_promote.sh`) на всех узлах кластера.
- 2) Настроить параметр в конфигурационном файле в `jadog.yml` на главном узле.
- 3) Инициировать переключение роли узла через процедуру `switchover (promote)`.
- 4) Ожидаемый результат:
  - Скрипт `post_promote.sh` выполнится на узле, у которого устанавливается роль главного;
  - Скрипт `pre_demote` выполнится на узле, у которого устанавливается роль резервного.

## 8. ОБНОВЛЕНИЕ И УДАЛЕНИЕ КОМПОНЕНТА

Обновление и удаление компонента «jaDog» выполняется соответственно при обновлении и удалении СУБД «Jatoba».

Правила и последовательность обновления и удаления СУБД «Jatoba» приведены в документе «Защищенная система управления базами данных «Jatoba». Руководство по установке».

При обновлении СУБД «Jatoba» необходимо учитывать, что компоненты «jaDog» на всех серверах кластера должны быть совместимы по версиям. Совместная работа разных версий компонентов «jaDog» не гарантируется и не поддерживается.

После завершения обновления СУБД «Jatoba» необходимо на обновленном сервере выполнить настройку отказоустойчивого кластера СУБД «Jatoba» в зависимости от типа ОС обновленного сервера.

### 8.1. Обновление (миграция) компонента «jaDog»

Процесс обновления (миграции) компонента «jaDog» 1.4, 2.0, 2.1, 2.2 на версию 3.4 состоит из следующих шагов:

- 1) Остановить кластер на предыдущей версии.

Важно соблюдать порядок остановки: сначала все каскадные узлы, потом все узлы с ролью Slave, потом узел Master. Такой Порядок нужен, чтобы избежать произвольного выполнения функциональной возможности «failover» (см. п. 7.17)

- 2) Первоначальная установка и настройка компонента «jaDog» версии 3.4.

Требуется установить и настроить компонент «jaDog» версии 3.4 согласно документации на всех узлах кластера.



Параметры каталогов БД должны совпадать с предыдущей конфигурацией

- 3) Формирование файла состояния /usr/jatoba-6/etc/jadog/jadog\_state.yml.

Пример такого файла состояния jadog\_state.yml:

```
cluster_nodes:
  Epoch: 0
  Datacenters:
```

```
- Datacenter: DEFAULT
  SyncCount: 0
  nodes:
    - ip: 173.16.3.2
      port: 12345
      ClusterState: Master
      NodeState: ACTIVE
      ReplicationSlot: node1
      NodeName: cl0_node1
      DBPort: 5432
      SyncState: 0
      PrimaryIP: ""
      PrimaryPort: ""
      PublicIP: 173.16.3.21/24
    - ip: 173.16.3.3
      port: 12345

      ClusterState: Slave
      NodeState: ACTIVE
      ReplicationSlot: node2
      NodeName: cl0_node2
      DBPort: 5432
      SyncState: 0
      PrimaryIP: 173.16.3.2
      PrimaryPort: 12345
      PublicIP: 173.16.3.18
    - ip: 173.16.3.4
      port: 12345
      ClusterState: Slave
      NodeState: ACTIVE
      ReplicationSlot: node3
      NodeName: cl0_node3
      DBPort: 5432
      SyncState: 0
      PrimaryIP: 173.16.3.2
      PrimaryPort: 12345
      PublicIP: 173.16.3.18
Activated: true
cluster_name: cl0
```

Формирование файла состояния `jadog_state.yml` сводится к тому, чтобы по предыдущей конфигурации нужно описать новую в формате, представленном выше.

Параметр «Epoch» не требует изменения.

Параметр «cluster\_name» меняется на имя предыдущего кластера.

Параметр «Activated» устанавливается в значение «true», если кластер был активирован или в значении «false», если не был.

### Описание узлов кластера

В разделе Datacenters описываются все датацентры кластера. По умолчанию всегда существует дата-центр с именем DEFAULT.

Допустим, что у нас простой отказоустойчивый кластер из двух узлов. Тогда описание в файле состояния jadog\_state.yml будет выглядеть следующим образом:

```
cluster_nodes:
  Epoch: 0
  Activated: true
  cluster_name: cl0
  Datacenters:
    - Datacenter: DEFAULT
      SyncCount: 0
      nodes:
        - ip: 173.16.3.2
          port: 12345
          ClusterState: Master
          NodeState: ACTIVE
          ReplicationSlot: node1
          NodeName: cl0_node1
          DBPort: 5432
          SyncState: 0
          PrimaryIP: ""
          PrimaryPort: ""
          PublicIP: 173.16.3.21/24
        - ip: 173.16.3.3
          port: 12345
          ClusterState: Slave
          NodeState: ACTIVE
          ReplicationSlot: node2
          NodeName: cl0_node2
          DBPort: 5432
          SyncState: 0
          PrimaryIP: 173.16.3.2
          PrimaryPort: 12345
          PublicIP: 173.16.3.18
```

Или кластер из трех узлов, один из которых в одном ДЦ, а остальные в другом:

```
cluster_nodes:
  Epoch: 5
  Datacenters:
    - Datacenter: DEFAULT
      SyncCount: 0
      nodes: ~
    - Datacenter: dc1
      SyncCount: 0
      nodes:
        - ip: 173.16.3.2
          port: 12345
          ClusterState: Master

          NodeState: ACTIVE
          ReplicationSlot: node1
          NodeName: cl0_node1
          DBPort: 5432
          SyncState: 0
          PrimaryIP: ""
          PrimaryPort: ""
          PublicIP: 173.16.3.21/24
        - Datacenter: dc2
          SyncCount: 0
          nodes:
            - ip: 173.16.3.3
              port: 12345
              ClusterState: Slave
              NodeState: ACTIVE
              ReplicationSlot: node2
              NodeName: cl0_node2
              DBPort: 5432
              SyncState: 0
              PrimaryIP: 173.16.3.2
              PrimaryPort: 12345
              PublicIP: 173.16.3.18
            - ip: 173.16.3.4
              port: 12345
              ClusterState: Slave
              NodeState: ACTIVE
              ReplicationSlot: node3
              NodeName: cl0_node3
              DBPort: 5432
              SyncState: 0
              PrimaryIP: 173.16.3.2
              PrimaryPort: 12345
              PublicIP: 173.16.3.18

  Activated: true
```



```
cluster_name: cl0
```

Аналогичным образом настраивается любая другая конфигурация.

В описании каждого датацентра есть параметр «SyncCount», который отвечает за количество синхронных реплик. Данный параметр требуется установить в актуальное значение для соответствующего дата-центра

Перейдем к описанию узлов кластера:

В каждом дата-центре есть категория nodes, в которой указывается список узлов, принадлежащих данному дата-центру.

Разберем описание узла подробнее:

```
- ip: 173.16.3.2
  port: 12345
  ClusterState: Master
  NodeState: ACTIVE
  ReplicationSlot: node1
  NodeName: cl0_node1
  DBPort: 5432
  SyncState: 0
  PrimaryIP: ""
  PrimaryPort: ""
  PublicIP: 173.16.3.21/24
```

В приведенном примере используются параметры:

- ip и port – сетевые параметры подключения к узлу;
- ClusterState - состояние узла в кластере (Master, Slave, Unknown, NotInitd);
- NodeState – состояние активности узла в кластере (ACTIVE, в процессе миграции другие состояния не учитываются);
- ReplicationSlot - имя слота репликации;
- NodeName - имя узла в кластере (параметр появился начиная с версии 3.0, поэтому требуется его определить);
- DBPort - порт СУБД;
- SyncState - определяет является ли резервный узел синхронным (Async - асинхронный, Sync - синхронный);

– PrimaryIP, PrimaryPort - ip, port с которого узел реплицируется. Для главного узла кластера эти параметры всегда пустые. У резервных узлов (не каскадных) - ip, port главного узла кластера. У каскадного узла - ip, port узла, с которого идет репликация;

– PublicIP - публичный адрес конкретного узла

Каждый узел в кластере описывается соответствующим образом.

4) Сохранить файл состояния под именем jadog\_state.yml и кладем в каталог с файлами конфигурации (/usr/jatoba-n/etc/jadog)

При формировании файла могут возникать ошибки форматирования, поэтому желательно перед установкой выполнить валидацию, к примеру (<https://jsonformatter.org/yaml-validator>).

5) Запускаем кластер и наблюдаем над восстановлением.

На данном шаге миграция компонента jaDog закончена.



Порядок запуска узлов в кластере не важен

## 8.2. Удаление компонента «jaDog»

Удаление компонента «jaDog» выполняется стандартными средствами встроенного менеджера пакетов для ОС GNU/Linux, или же при помощи деинсталлятора для ОС Microsoft Windows.

## 9. СООБЩЕНИЯ ОБ ОШИБКАХ

### 9.1. Ошибка добавления резервного узла

При добавлении резервного узла командой:

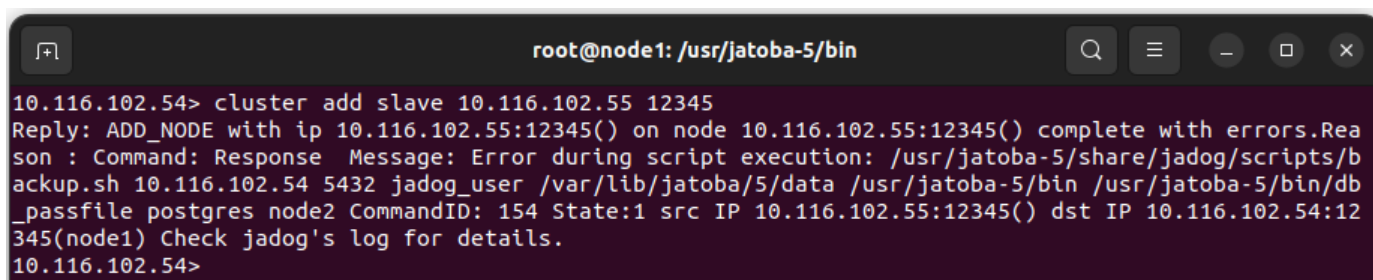
```
cluster add slave [host_ip/host_name] [Jadog PORT number  
(port)]
```

в приводимом примере выполняется команда:

```
10.116.102.54> cluster add slave 10.116.102.55 12345
```

может возникнуть следующая ошибка:

```
Reply: ADD_NODE with ip 10.116.102.55:12345() on node  
10.116.102.55:12345() complete with errors.Reason : Command:  
Response Message: Error during script execution: /usr/jatoba-  
6/share/jadog/scripts/backup.sh 10.116.102.54 5432 jadog_user  
/var/lib/jatoba/6/data /usr/jatoba-6/bin /usr/jatoba-  
6/bin/db_passfile postgres node2 CommandID: 154 State:1 src IP  
10.116.102.55:12345() dst IP 10.116.102.54:12345(node1) Check  
jadog's log for details.
```

A screenshot of a terminal window with a dark background. The title bar shows 'root@node1: /usr/jatoba-5/bin'. The terminal text is as follows:

```
10.116.102.54> cluster add slave 10.116.102.55 12345  
Reply: ADD_NODE with ip 10.116.102.55:12345() on node 10.116.102.55:12345() complete with errors.Reason : Command: Response Message: Error during script execution: /usr/jatoba-5/share/jadog/scripts/backup.sh 10.116.102.54 5432 jadog_user /var/lib/jatoba/5/data /usr/jatoba-5/bin /usr/jatoba-5/bin/db_passfile postgres node2 CommandID: 154 State:1 src IP 10.116.102.55:12345() dst IP 10.116.102.54:12345(node1) Check jadog's log for details.  
10.116.102.54>
```

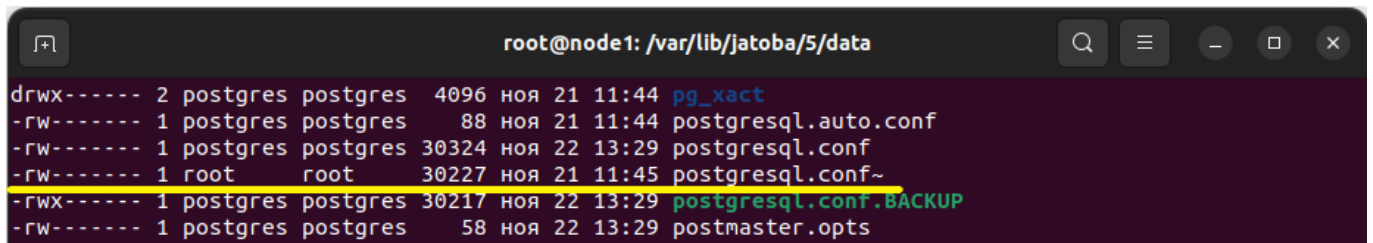
Рисунок 9.1 – Ошибка добавления резервного узла

Ошибка возникает на главном узле (Master) из-за открытия файла /var/lib/jatoba/6/data/postgresql.conf от имени и с правами пользователя «root» и некорректного его закрытия.

Ошибка регистрируется на резервном узле в файле /usr/jatoba-6/var/log/jadog/jadog.log:

```
pg_basebackup: error: backup failed: ERROR: could not open  
file "./postgresql.conf~": Permission denied
```

В итоге возникает ошибка доступа.



```
root@node1: /var/lib/jatoba/5/data
drwx----- 2 postgres postgres 4096 ноя 21 11:44 pg_xact
-rw----- 1 postgres postgres  88 ноя 21 11:44 postgresql.auto.conf
-rw----- 1 postgres postgres 30324 ноя 22 13:29 postgresql.conf
-rw----- 1 root      root      30227 ноя 21 11:45 postgresql.conf~
-rwx----- 1 postgres postgres 30217 ноя 22 13:29 postgresql.conf.BACKUP
-rw----- 1 postgres postgres  58 ноя 22 13:29 postmaster.opts
```

Рисунок 9.2 – Вывод списка файлов

Ошибку возможно устранить:

– назначением прав доступа, выполнив команду:

```
chown postgres:postgres /var/lib/jatoba/6/data/postgresql.conf~
```

– удалением файла выполнив команду:

```
rm -rf /var/lib/jatoba/6/data/postgresql.conf~
```

## ПРИЛОЖЕНИЕ 1

### Структура конфигурационного файла jadog.yml

Таблица П.1.1 – Структура конфигурационного файла jadog.yml компонента «jaDog» для ОС Windows/Linux

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузк и кластера	Описание параметра
param_path					<b>Рабочие директории</b>
module_path	6.7.1.4	module_path: "C:\Program Files\GIS\Jatoba\ 6\share\jadog\scri pts"	set parameter param_path: module_path = '/usr/jatoba- 6/share/jadog /scripts'	—	Полный путь к дополнительным скриптам, необходимым для проведения системных действий
		module_path: /usr/jatoba- 6/share/jadog/scri pts			
state_path		state_path: "C:\Program Files\GIS\Jatoba\ 6\var\lib\jadog"	set parameter param_path:s tate_path = '/usr/jatoba- 6/etc/jadog'	—	Полный путь к дополнительным файлам, необходимым для работы (в данной версии не используется!)
		state_path: /usr/jatoba- 6/var/lib/jadog			

№ изменения: \_\_\_\_\_ Подпись отв. лица: \_\_\_\_\_ Дата внесения изм: \_\_\_\_\_

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
log_path	6.7.9.1	log_path: "C:\Program Files\GIS\Jatoba\6\var\log\jadog"	set parameter param_path:log_path = '/usr/jatoba-6/var/log/jadog'	X	Полный путь к директории хранения файла журнала сообщений компонента «jaDog».
		log_path: /usr/jatoba-6/var/log/jadog			
param_postgres					Параметры postgres
db_bin_path	6.7.1.1	db_bin_path: "C:\Program Files\GIS\Jatoba\6\bin"	—	X	Полный путь к системным утилитам СУБД*
		db_bin_path: /usr/jatoba-6/bin			
db_data_path	6.7.1.2	db_data_path: "C:\Program Files\GIS\Jatoba\6\data"	—	X	Полный путь к директории данных узла кластера
		db_data_path: /var/lib/jatoba/6/data			

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
db_service_name	6.7.2.1	db_service_name: JatobaServer	—	—	Параметр, отвечающий за имя сервиса БД в системе
		db_service_name: jatoba-6			
db_fail_connection_attempts		5	set parameter param_postgr es: db_fail_conn ection_attem pts = '5'	—	Количество попыток восстановления подключения при возникновении ошибки
db_check_interval	6.7.3.4	5000	set parameter param_postgr es: db_check_int erval = '5000'	—	Интервал проверки подключения к СУБД (мс)
wal_level	6.7.7.5	replica	set parameter param_postgr es: wal_level = 'replica'	—	Определяет, какой объем информации записывается в WAL
<b>param_replication</b>					<b>Параметры репликации</b>
replication_slot_name	6.7.5.14	—	—	—	Имя слота репликации, уникальное в каждом узле кластера
№ изменения: _____		Подпись отв. лица: _____		Дата внесения изм: _____	

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
<b>db_connection_settings</b>					<b>Параметры подключения к СУБД</b>
db_auth_method	6.7.5.6	password	—	—	Метод аутентификации при подключении к СУБД. Поддерживаемые типы: password, ssl
passfile	6.7.5.5	passfile: " C:\Program Files\GIS\Jatoba\ 6\bin\db_passfile"	—	—	Путь к файлу с паролями от СУБД
		passfile: /usr/jatoba- 6/bin/db_passfile			
<b>param_synchronous</b>					<b>Параметры синхронизации</b>
synchronous_commit	6.7.7.2	remote_apply	set parameter param_synchronous: synchronous_commit = ' remote_apply '	—	Настройка синхронной фиксации
synchronous_type	6.7.7.3	ANY	set parameter param_synchronous: synchronous	—	Тип синхронной репликации Доступны для установки следующие значения параметра:



Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
			ronous_type = 'ANY'		<ul style="list-style-type: none"> <li>– ANY;</li> <li>– FIRST.</li> </ul>
synchronous_commit_max_nodes	6.7.7.4	3	set parameter param_synchronous_commit_max_nodes = "	—	Количество резервных узлов, т.е. серверов с ролью «Slave»
param_jadog					<b>Параметры jadog</b>
ip	6.7.2.2	ip:	—	X	<p>IP-адрес/ маска в CIDR-нотации, через который будет производиться обмен данными с другими компонентами «jaDog» на других узлах кластера.</p> <p>Параметр обязателен к заполнению.</p> <p>*Допускается редактирование, если узел не проинициализирован. Иначе редактирование запрещено.</p>
port	6.7.2.3	port: 12345	—	X	<p>Порт обмена служебными данными компонентов «jaDog».</p> <p>*Допускается редактирование, если узел не проинициализирован. Иначе редактирование запрещено.</p>

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
user_interface_port	6.7.3.9	user_interface_port: 54321	—	X	Порт доступа административных подключений для управления кластером через утилиту «jadog_ctl»  Сообщение пользователю после изменения параметра: «Параметр установлен. Требуется перезагрузка сервиса jadog. Для продолжения работы откройте новую сессию на установленном порту!»
jadog_search_port	6.7.2.4	jadog_search_port: 12346	—	—	Порт для поиска "свободных" узлов в подсети
public_address	6.7.3.1	public_address:	—	—	IP-адрес/ маска в CIDR-нотации, через который будет происходить доступ пользователей к СУБД. Данный адрес является общим для кластера, поддерживается его миграция при выходе из строя узлов кластера. Параметр обязателен к заполнению.
public_address_control_interval	6.7.3.2	5000	—	—	Контрольный интервал проверки виртуального IP-адреса кластера (мс)

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
public_address_control_attempts	6.7.3.3	3	—	—	Количество контрольных вызовов проверки виртуального IP-адреса
trusted_address	6.7.3.5	trusted_address:	—	—	Доверенный адрес, служит дополнительным индикатором принятия решения при обработке отказа «failover». Если узел доступен по сети, то резервный узел перейдет в статус главного. Если не доступен, то узел будет считать, что он сам находится в сетевой изоляции и войдет в состояние «fencing» (запрет «failover» + удаление «public_address», если присутствует). Если параметр не заполнен, то опция считается отключенной.
rdbms_trusted_ip	6.7.3.6	rdbms_trusted_ip:	—	—	Доверенный IP адрес системы управления реляционными базами данных
rdbms_trusted_port	6.7.3.7	rdbms_trusted_ip:	—	—	Доверенный порт системы управления реляционными базами данных
network_interface	6.7.3.8	network_interface :	—	—	Имя интерфейса на данном узле кластера, на который осуществляется привязка общего адреса (используется в скриптах addip, checkip и delip).

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
					Параметр обязателен к заполнению. Перезагрузка не требуется. Но все действующие соединения прервутся. (Понадобится замена IP адреса)
failover_connection_attempts		10000	set parameter failover_connections_timeout = '1000'	—	Время в миллисекундах (числовой параметр) проверки доступности главного узла до момента объявления его вышедшим из строя
failover_connection_timeout		10	set parameter failover_connections_attempts = '1'	—	Количество попыток, по истечении которых, если резервный узел не смог установить связь с главным, то объявляется аварийная ситуация – отказ, и начинается выполняться процедура обработки отказа
interconnect_user	6.7.2.6	interconnect_user :	—	—	Название технологической учетной записи для взаимодействия с другими сервисами jaDog
<b>param_log</b>					<b>Параметры логирования</b>
logs_filename	6.7.9.2	logs_filename: jadog-%a	set parameter logs_filename = 'jadog-%a '	X	Имя файла журнала. Создается в директории, указанной в «log_path»

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузк и кластера	Описание параметра
logs_screen	6.7.9.13	logs_screen: false	set parameter logs_screen = 'true'	X	Вывод всех сообщений на экран (удобно для ручных и отладочных запусков)
logs_file	6.7.9.12	logs_file: true	set parameter logs_file = 'false'	X	Вывод всех сообщений в файл, имя которого задается параметром «logs_filename»
logs_file_mode	6.7.9.3	logs_file_mode: 0600	—	—	Режим доступа к файлам журнала событий кластера
logs_level	6.7.9.8	logs_level: info	—	X	<p>Уровень детализации сообщений в порядке увеличения объема выдаваемых сообщений:</p> <ul style="list-style-type: none"> <li>– fatal;</li> <li>– error;</li> <li>– warning;</li> <li>– info;</li> <li>– debug1;</li> <li>– debug2;</li> <li>– debug3.</li> </ul> <p>Для тестирования и поиска ошибок рекомендуется ставить значение «debug1» и выше (с выводом детальной отладочной информации). Для</p>

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
					обычного режима эксплуатации достаточно уровня «info»
logs_type	6.7.9.4	logs_type: csv	set parameter logs_type = 'csv, security.csv '	—	Формат файла журнала событий кластера. Поддерживаются форматы файла журнала: <ul style="list-style-type: none"> <li>– txt;</li> <li>– csv;</li> <li>– json.</li> </ul>
logs_rotation_age	6.7.9.5	logs_rotation_age : 1d	set parameter log_rotation_ age = '1d'	—	Максимальное количество времени для использования отдельного файла журнала, по истечении которого будет создан новый файл журнала
logs_rotation_size	6.7.9.6	logs_rotation_size : 10MB	set parameter log_rotation_ size = '10MB'	—	Максимальный размер отдельного файла журнала
logs_truncate_on_rotation	6.7.9.7	logs_truncate_on _rotation: false	set parameter log_truncate_ on_rotation = 'true'	—	Перезапись файла журнала событий кластера
param_security_log					<b>Параметры журнала событий информационной безопасности (ИБ)</b>

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
security_logs_filename	6.7.9.10	security_logs_filename: jalog-%a	set parameter logs_filename = 'jalog-%Y-%m-%d_%H%M%S.log'	—	Имя файла журнала событий безопасности кластера
security_logs_filemode	6.7.9.11	security_logs_filemode: 0600	set parameter log_file_mode = '0777'	—	Режим доступа к файлам журнала событий безопасности кластера
security_log_path	6.7.9.9	security_log_path : /usr/jatoba-6/var/log/jalog	set parameter log_path = '/usr/jatoba-6/var/log/jalog'	—	Директория хранения журнала событий безопасности кластера
<b>param_recovery</b>					<b>Характеристики отказоустойчивости</b>
recovery_cluster_timeout		3000000		—	Количество микросекунд кратное 3000. Результат деления значения этого параметра на 3000 определяет количество попыток ожидания восстановления статуса кластера после перезапуска (так называемая стадия RECOVER, когда компонент «jaDog» должен запуститься, проверить

№ изменения: \_\_\_\_\_ Подпись отв. лица: \_\_\_\_\_ Дата внесения изм: \_\_\_\_\_

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузк и кластера	Описание параметра
					конфигурацию кластера, убедиться в наличии главного узла и вступить в работу по контролю доступности всех узлов)
<b>param_after_rewing_fail_backup</b>					<b>Параметры после перемотки неудачного резервного копирования</b>
after_rewind_fail_backup_allow	7.18	after_rewind_fail _backup_allow: true	set parameter after_rewind _fail_backup _allow = 'false'	—	Восстановление работоспособности «старого» главного сервера после отказа по умолчанию выполняется методом отката сохранившегося журнала транзакций на последнее согласованное с «новым» главным сервером состояние. В случае ошибки отката данный параметр разрешает создание новой полной реплики с «нового» главного сервера
after_rewind_fail_backup_coeff	7.18	after_rewind_fail _backup_coeff: 1	set parameter after_rewind _fail_backup _coeff = '0.5'	—	Коэффициент, регулирующий размер пространства на диске, которое должно быть свободно для новых данных в зависимости от размера текущей директории с данными при восстановлении отказавшего узла



Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
					методом создания новой полной реплики. <ul style="list-style-type: none"> <li>– 1 – размер свободного пространства должен быть не меньше размера директории с данными;</li> <li>– 0.5 – половине и т.д</li> </ul> Коэффициент для расчета ожидаемой свободной памяти после бэкапа рассчитывается по формуле: $*(1+K)$ , где K - часть от размера директории данных, которая должна остаться свободной.
<b>param_tcp</b>					<b>Параметры TCP</b>
tcp_keepalive_count		tcp_keepalive_count: 1	set parameter tcp_keepalive_count = '1'	X	Параметр TCP сокета для определения разрыва соединения. Отвечает за отправляемое максимальное количество отправляемых "оживляющих" (keep-alive) проверок перед тем, как соединение считается разорванным
tcp_keepalive_time		tcp_keepalive_time: 10	set parameter tcp_keepalive_time = '10'	X	Параметр TCP сокета для определения разрыва соединения. Отвечает за время в секундах, через которое будут

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
					отправляться валидирующие соединения пакеты после последнего принятого сообщения
tcp_keepalive_interval		tcp_keepalive_interval: 3	set parameter tcp_keepalive_interval = '4'	X	Параметр TCP сокета для определения разрыва соединения. Отвечает за интервал времени между посылкой валидирующих пакетов
param_ssl					<b>Параметры SSL</b>
ssl	6.7.2.5	ssl: false	set parameter ssl = 'true'	X	Активация/деактивация режима соединения по протоколу SSL между узлами кластера. Сообщение пользователю после изменения параметра: «Параметр установлен. Потребуется изменение метода аутентификации и перезагрузка Jadog»
ssl_ca_file	7.20.9	ssl_ca_file:	set parameter ssl_ca_file = '/home/user/newCert/ca.crt'	X	Путь до сертификата удостоверяющего центра (CA)

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
ssl_cert_file	7.20.9	ssl_cert_file:	set parameter ssl_cert_file = '/home/user/newCert/server.crt'	X	Путь до сертификата, который используется для аутентификации и установки tls/ssl соединения (параметр для Linux)
ssl_crl_file	7.20.9	ssl_crl_file:	set parameter ssl_crl_file = '/home/user/newCert/ca.crl'	X	Путь до списка отзыва сертификатов, в случае если список хранится локально (параметр для Linux)
ssl_key_file	7.20.9	ssl_key_file:	set parameter ssl_key_file= '/home/user/newCert/server.key'	X	Путь до закрытого ключа сертификата (параметр для Linux)
<b>param_system</b>					<b>Пользователь системы</b>
system_user	6.7.5.13	system_user: postgres	set parameter param_system: system_user = 'postgres'	X	Имя служебного пользователя ОС от имени и с правами которого будут запускаться службы jaDogService и jaDogRunAsService

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
					Имя служебного пользователя ОС от имени и с правами которого будут запускаться службы (демон) jadog.service и jatoba-<ver>
<b>param_archive</b>					<b>Параметры архивации</b>
wal_archive_directory	6.7.8.1	wal_archive_directory:	set parameter wal_archive_directory = '/home/arhive_wal/'	X	Системный каталог для архивирования/восстановления WAL. Сообщение пользователю после изменения параметра: «Параметр установлен. Требуется выполнить pg_reload_conf() в СУБД»
wal_archive_command	6.7.8.2	wal_archive_command: "cp %p /nfs/arhive_wal/%f"	set parameter wal_archive_command = 'cp %p /home/arhive_wal/%f'	X	Команда копирования архива WAL. Сообщение пользователю после изменения параметра: «Параметр установлен. Требуется выполнить pg_reload_conf() в СУБД»
wal_archive_cleanup_command	6.7.8.5	wal_archive_cleanup_command: "/usr/jatoba-6/bin/pg_archivecleanup	set parameter wal_archive_cleanup_command = '/usr/jatoba-	—	Запущен процесс очистки архива WAL

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
		/nfs/arhive_wal %s"	6/bin/pg_archivecleanup /home/arhive_wal/ %s'		
wal_archive_cleanup_needed	6.7.8.4	false	set parameter wal_archive_cleanup_needed = 'true'	—	Команда очистки архива WAL
wal_archive_cleanup_timeout	6.7.8.6	30000	set parameter wal_archive_cleanup_timeout = '500000'	—	Тайм-аут очистки архива WAL
<b>param_restore</b>					<b>Параметры восстановления</b>
wal_restore_command	6.7.8.3	wal_restore_command: "cp /nfs/arhive_wal/%f %p"	set parameter wal_restore_command = 'cp /home/arhive_wal/%f %p'	X	Команда восстановления архива WAL. Сообщение пользователю после изменения параметра: «Параметр установлен. Требуется выполнить pg_reload_conf() в СУБД»
<b>param_cluster_behavior</b>					<b>Параметры поведения кластера</b>

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
autofailover_mode	6.7.6.1	true	set parameter autofailover_ mode = 'true'	—	Режим автоматической обработки отказа сервера кластера с ролью «Master»
dc_autofailover_mode	6.7.6.2	false	set parameter dc_autofailov er_mode = 'false'	—	Режим автоматической смены дата- центра
<b>param_connection</b>					<b>Параметры подключения jadog</b>
max_admin_connections	7.12	10	set parameter max_admin_ connections = '10'		Максимальное количество подключений администраторов кластера
conn_string		—	set parameter param_conne ction:conn_st ring = 'host=127.0.0 .1 port=5432 user=jadog_u ser dbname=post gres passfile=/usr/	X	Параметры подключения (host, port, user, dbname, passfile)

№ изменения: \_\_\_\_\_ Подпись отв. лица: \_\_\_\_\_ Дата внесения изм: \_\_\_\_\_

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
			jatoba- 6/bin/db_pass file'		
<b>param_bundle</b>					<b>Параметры синхронизации bundle</b>
update_bundle_status_time		20	set parameter update_bundl e_status_time = '20'	X	Временной промежуток отправки статуса bundle от главного узла на подчиненные узлы (мс)
check_bundle_status_time		60	set parameter check_bundle _status_time = '60'	X	Временной промежуток проверки статуса bundle (мс)
update_cluster_status_time		20	set parameter update_cluste r_status_time = '20'	X	Временной промежуток отправки статуса кластера между главными узлами с ролью Master внутри bundle (мс)
<b>param_scheduler</b>					<b>Параметры планировщика заданий кластера</b>
timer_interval	7.23.5.1	1000	set parameter timer_interva l = '1000'	X	Интервал внутреннего таймера на опрос текущего времени и времени выполнения, мс

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузк и кластера	Описание параметра
timer_inaccuracy	7.23.5.2	5000	set parameter timer_inaccu racy = '5000'	X	Погрешность отставания времени выполнения от фактически установленного времени, мс
max_jobs_count	7.23.5.3	8	set parameter max_jobs_co unt = '8'	X	Максимальное количество задач в очереди планировщика
<b>file_manager</b>					<b>Параметры синхронизации файлов кластера</b>
max_file_size	7.24.8.1	1	set parameter max_file_siz e = '1'	X	Максимальный размер синхронизируемых файлов
max_files_per_session	7.24.8.2	32	set parameter max_files_pe r_session = '16'	X	Максимальное количество синхронизируемых файлов и каталогов
<b>scenarios</b>					<b>Параметры запускаемых сценариев</b>
events:promote:pre_script	7.25.6.1		set parameter scenarios:eve nts:promote:p re_script = "/path/to/	X	Предварительный сценарий при смене роли узла до главного



Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузки и кластера	Описание параметра
			script_pre_promote.sh"		
events:promote:post_script	7.25.6.2		set parameter scenarios:events:promote:post_script = "/path/to/script_post_promote.sh"	X	Завершающий сценарий при смене роли узла до главного
events:demote:pre_script	7.25.6.3		set parameter scenarios:events:demote:pre_script = "/path/to/script_pre_demote.sh"	X	Предварительный сценарий при смене роли узла до резервного
events:demote:post_script	7.25.6.4		set parameter scenarios:events:demote:post_script = "/path/to/script_post_demote.sh"	X	Завершающий сценарий при смене роли узла до резервного

Наименование параметра	Пункт	Значение по умолчанию Windows/Linux	Пример команды изменения параметра	Параметр, требующий перезагрузк и кластера	Описание параметра
events:rewind:pre_script	7.25.6.5		set parameter scenarios:events:rewind:pre_script = "/path/to/ script_pre_rewind.sh"	X	Предварительный сценарий при выполнении синхронизации узлов кластера
events:rewind:post_script	7.25.6.6		set parameter scenarios:events:rewind:post_script = "/path/to/ script_post_rewind.sh"	X	Завершающий сценарий при выполнении синхронизации узлов кластера
script_timeout	7.25.6.7	3000	set parameter scenarios: script_timeout = "4000"	X	Задержка выполнения сценариев в миллисекундах

*\*параметр – параметр не подлежит редактированию.*

### Структура файла состояния кластера jadog\_state.yml

Таблица П.1.2 – Структура файла состояния кластера jadog\_state.yml компонента «jaDog» для ОС Windows/Linux

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

Наименование параметра	Значение по умолчанию Windows/Linux	Описание параметра
cluster_nodes	cluster_nodes:	Список узлов кластера
cluster_name	cluster_name:	Название кластера
Epoch		
Datacenters:	DEFAULT	Название дата-центра, к которому относится данный кластер
SyncCount		
nodes		Список узлов, входящий в данный кластер
ip		IP-адрес узла кластера
port	12345	Сетевой порт, на котором запущен компонент кластера «jaDog»
ClusterState		Роль узла в кластере. Может принимать значения: Master, Slave, Unknown, NotInit
NodeState	ACTIVE	Состояние узла кластера. Может принимать значения: INIT (только для узла с ролью Master), BACKUP (только для узла с ролью Slave), STARTING, ACTIVE, STOPPED, DEAD, FENCED, RECOVERY, PROMOTING (только для узла с ролью Slave), DEMOTING (только для узла с ролью Master) CANDIDATE
ReplicationSlot		Слот репликации узла кластера
NodeName		Название узла кластера
DBPort	5432	Сетевой порт подключения СУБД «Jatoba»
SyncState		Состояние синхронизации узла кластера. Может принимать значения: Async,
PrimaryIP		IP-адрес главного узла кластера, через который будет производиться обмен данными с другими компонентами «jaDog» на других узлах кластера.

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

Наименование параметра	Значение по умолчанию Windows/Linux	Описание параметра
		Параметр обязателен к заполнению. Указывается для резервных узлов кластера
PrimaryPort		Сетевой порт главного узла кластера, через который будет производится обмен служебными данными компонентов «jaDog». Указывается для резервных узлов кластера
PublicIP		IP-адрес, через который будет происходить доступ пользователей к СУБД. Данный адрес является общим для кластера, поддерживается его миграция при выходе из строя узлов кластера
Activated		Статус активации кластера. Может принимать значения: true, false

# **Перечень исполняемых скриптов в ОС Windows и GNU Linux**

Таблица П.1.3 – Перечень исполняемых скриптов

<b>Linux /usr/jatoba- 6/share/jadog/scripts</b>	<b>Windows C:\Program Files\GIS\Jatoba\6\share\jadog\scripts</b>	<b>Описание</b>
addip.sh	addip.bat	Добавление общего адреса на заданный интерфейс
backup.sh	backup.bat	Создание реплики
checkip.sh	checkip.bat	Проверка наличия общего адреса
checknoservice.sh	checknoservice.bat	Проверка наличия установленного сервиса
delip.sh	delip.bat	Удаление общего адреса с заданного интерфейса
demote.sh	demote.bat	В текущей версии компонента «jaDog» не используется
fixrecovery-12.sh	fixrecovery-12.bat	Восстановление файла «recovery.conf», если таковой был удален, для версии Jatoba 2x и выше
fixrecovery-13.sh		Восстановление файла «recovery.conf», если таковой был удален, для версии Jatoba 3x и выше
fixrecovery.sh	fixrecovery.bat	Восстановление файла «recovery.conf», если таковой был удален
ping.sh	ping.bat	Проверка доступности сетевого узла
–	ping.ps1	Плагин для ping.bat
installservice.sh	–	Установка jadog-сервиса (через опцию install; делается из jadog-setup автоматически)
promote.sh	promote.bat	Перевод узла в роль главного
rewind-13.sh	rewind-13.bat	Восстановление узла в роли главного после падения и перевода обслуживания на резервный узел для версии Jatoba 3x и выше
rewind.sh	rewind.bat	Восстановление узла в роли главного после падения и перевода обслуживания на резервный узел для версии Jatoba 1x - 2x
–	runasstart.bat	Запуск вспомогательного сервиса «jadogrunas»
–	runasstop.bat	Остановка вспомогательного сервиса «jadogrunas»
servicestatus.ch	servicestatus.bat	Проверка статусов сервисов операционной системы
start.sh	start.bat	Запуск процессов СУБД
stop.sh	stop.bat	Остановка процессов СУБД

<b>Linux</b> <b>/usr/jatoba-</b> <b>6/share/jadog/scripts</b>	<b>Windows</b> <b>C:\Program</b> <b>Files\GIS\Jatoba\6\share\jadog\scripts</b>	<b>Описание</b>
trusted_rdbms.sh		Проверка доступности IP адреса доверенной СУБД
uninstallservice.sh	—	Удаление jadog-сервиса

## ПРИЛОЖЕНИЕ 2

### Пример установки СУБД «Jatoba» из локального репозитория для ОС Ubuntu

Установка СУБД «Jatoba» из локального репозитория для ОС Ubuntu проводится в следующем порядке:

- 1) В терминале войти в режим суперпользователя, выполнив команду:

```
sudo su
```

- 2) Если команды `sudo` не существует – установить:

```
su -l  
apt-get install sudo -y
```

- 3) Выполнить обновление системы:

```
sudo apt update && sudo apt upgrade -y  
sudo apt -s dist-upgrade  
sudo apt dist-upgrade
```

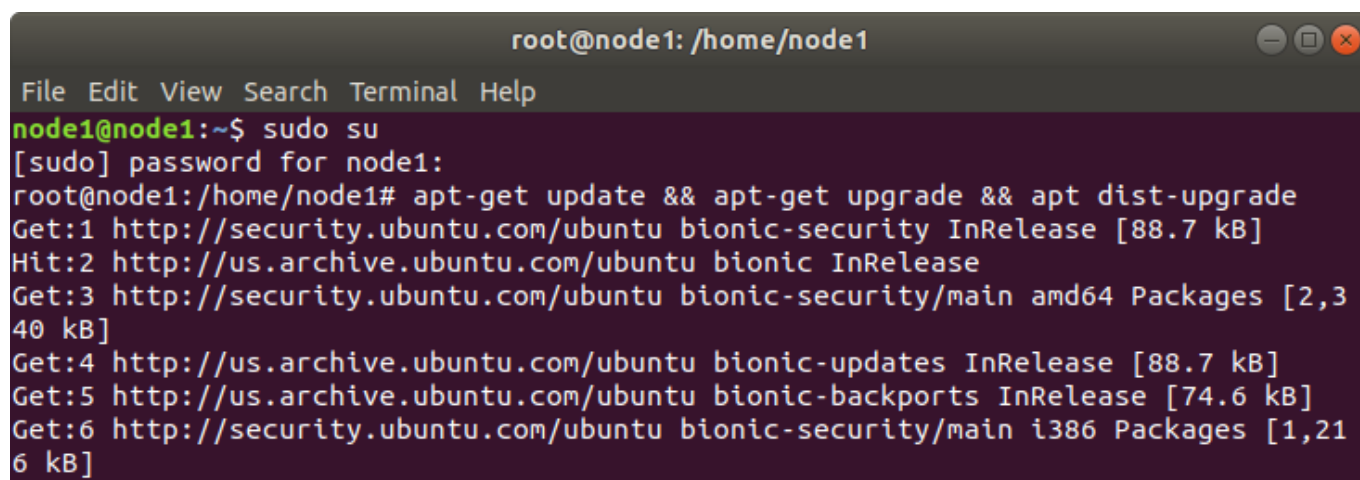


Рисунок 2.1 – Обновление системы

- 4) Создать папку `localrepo` в корневом каталоге:

```
mkdir /localrepo
```

- 5) В созданную папку скопировать:

- каталог `<pool>`
- каталог `<dist>`

- файл <DEB-GPG-KEY-Jatoba>



Рисунок 2.2 – Структура каталога «localrepo»

- 6) Установить открытый ключ репозитория:

```
apt-key add /localrepo/DEB-GPG-KEY-Jatoba
```

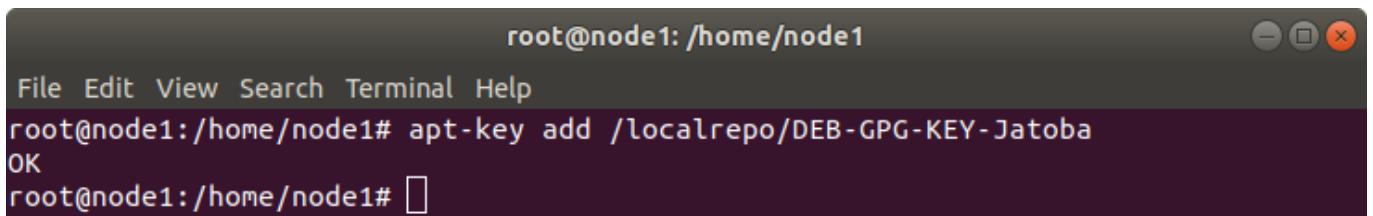


Рисунок 2.3 – Установка открытого ключа репозитория

- 7) Добавить описание локального репозитория в систему:

```
nano /etc/apt/sources.list.d/jatoba-6.list
```

- 8) Вставить в файл следующее содержимое и сохранить:

```
deb file:///localrepo stable non-free
```

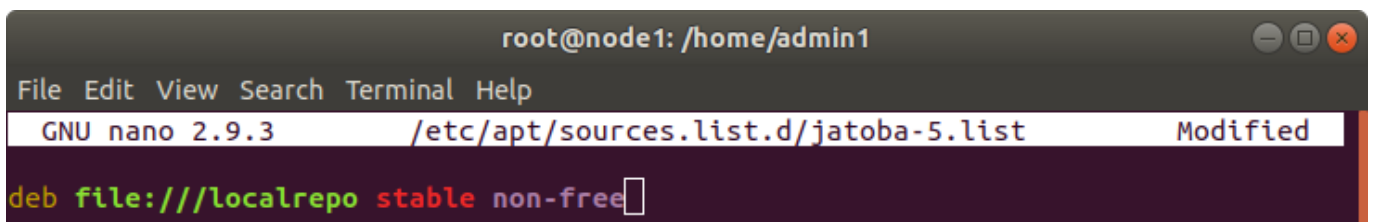
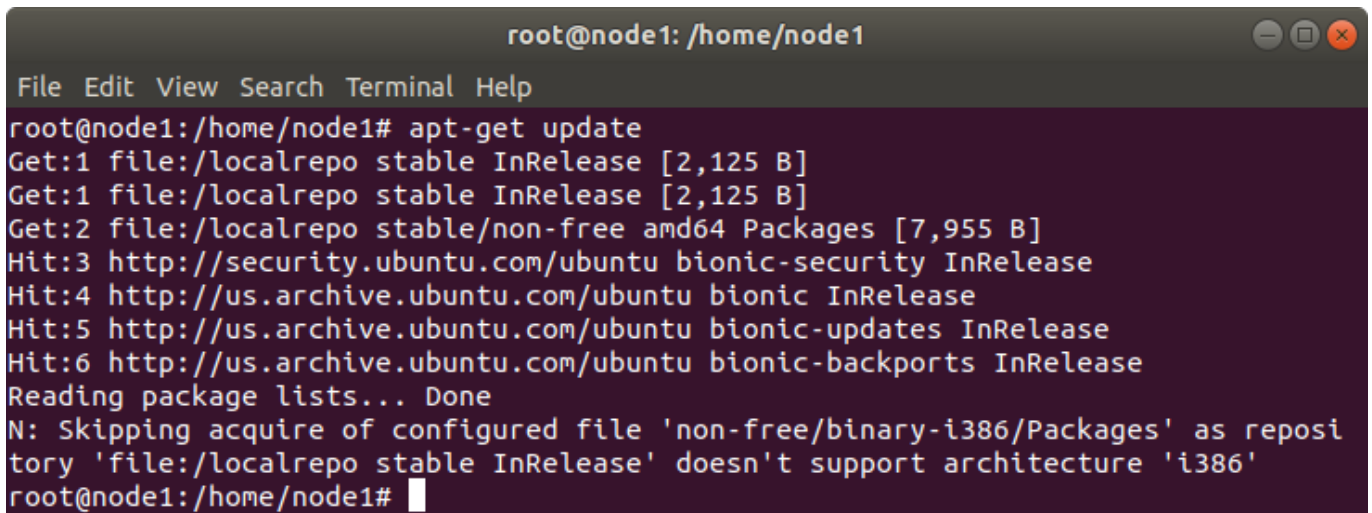


Рисунок 2.4 – Содержание файла «jatoba-6.list»

- 9) Проиндексировать обновленное состояние репозитория:

```
apt-get update
```



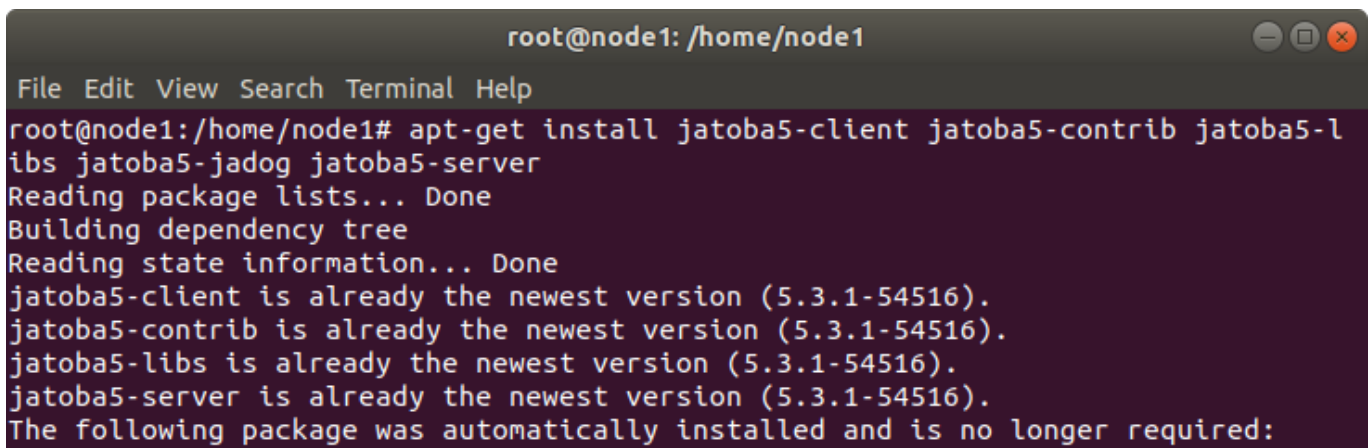


```
root@node1: /home/node1
File Edit View Search Terminal Help
root@node1:/home/node1# apt-get update
Get:1 file:/localrepo stable InRelease [2,125 B]
Get:1 file:/localrepo stable InRelease [2,125 B]
Get:2 file:/localrepo stable/non-free amd64 Packages [7,955 B]
Hit:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
N: Skipping acquire of configured file 'non-free/binary-i386/Packages' as repository 'file:/localrepo stable InRelease' doesn't support architecture 'i386'
root@node1:/home/node1#
```

Рисунок 2.5 – Индексация репозитория

- 10) Установить СУБД Jatoba при помощи команды:

```
apt-get install jatoba6-client jatoba6-contrib jatoba6-libs
jatoba6-jadog jatoba6-server
```



```
root@node1: /home/node1
File Edit View Search Terminal Help
root@node1:/home/node1# apt-get install jatoba5-client jatoba5-contrib jatoba5-libs
jatoba5-jadog jatoba5-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
jatoba5-client is already the newest version (5.3.1-54516).
jatoba5-contrib is already the newest version (5.3.1-54516).
jatoba5-libs is already the newest version (5.3.1-54516).
jatoba5-server is already the newest version (5.3.1-54516).
The following package was automatically installed and is no longer required:
```

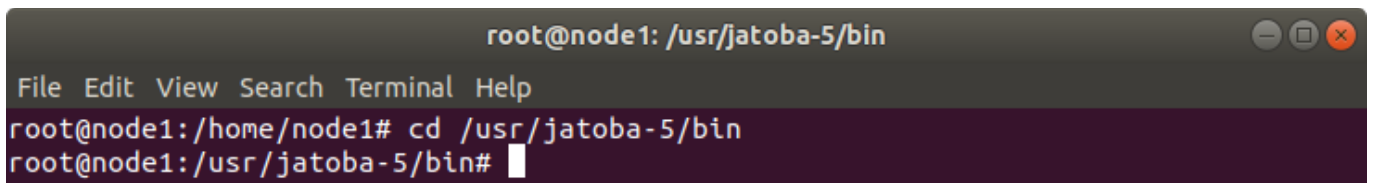
Рисунок 2.6 – Установка пакетов

- 11) Убедиться, что отсутствуют ошибки зависимостей:

```
for f in $(LANG=C find /usr/jatoba-6 -type f -exec file {} \; |
grep "ELF 64-bit LSB" | awk 'BEGIN {FS=":"} { print $1}' |
sort); do echo $f; ldd $f | grep "not found"; done
```

- 12) Перейти в директорию исполняемых файлов СУБД:

```
cd /usr/jatoba-6/bin
```



```
root@node1: /usr/jatoba-5/bin
File Edit View Search Terminal Help
root@node1:/home/node1# cd /usr/jatoba-5/bin
root@node1:/usr/jatoba-5/bin#
```

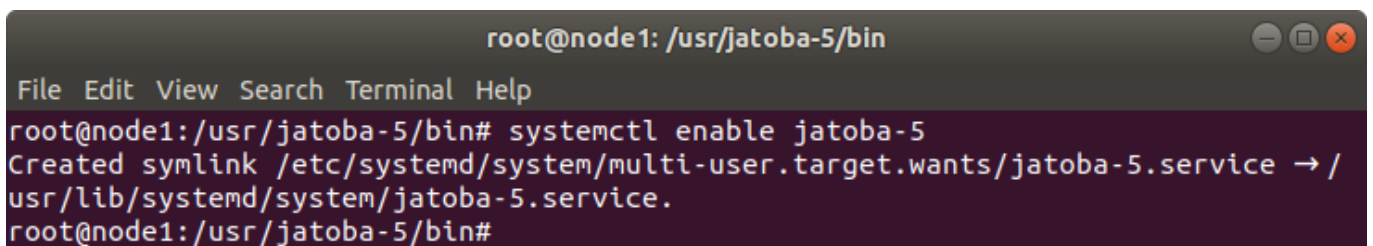
Рисунок 2.7 – Переход в каталог

- 13) Инициализировать каталог данных СУБД при помощи команды:

```
./jatoba-setup initdb jatoba-6
```

- 14) Добавить сервис в список автозапуска:

```
systemctl enable jatoba-6
```

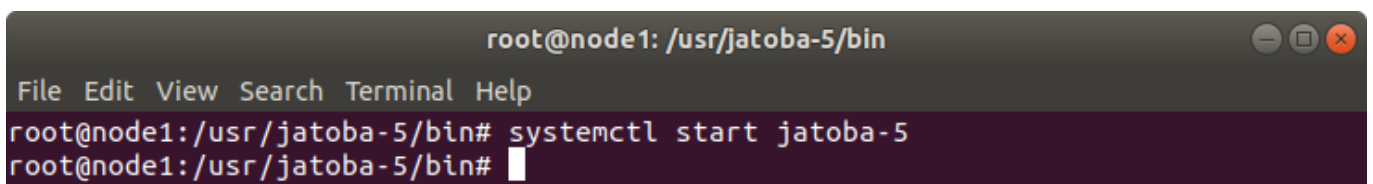


```
root@node1: /usr/jatoba-5/bin
File Edit View Search Terminal Help
root@node1:/usr/jatoba-5/bin# systemctl enable jatoba-5
Created symlink /etc/systemd/system/multi-user.target.wants/jatoba-5.service → /usr/lib/systemd/system/jatoba-5.service.
root@node1:/usr/jatoba-5/bin#
```

Рисунок 2.8 – Добавление сервиса jatoba-6 в автозагрузку ОС

- 15) Запустить службу:

```
systemctl start jatoba-6
```



```
root@node1: /usr/jatoba-5/bin
File Edit View Search Terminal Help
root@node1:/usr/jatoba-5/bin# systemctl start jatoba-5
root@node1:/usr/jatoba-5/bin#
```

Рисунок 2.9 – Запуск службы jatoba-6

- 16) Проверить статус службы:

```
systemctl status jatoba-6
```

```
root@node1: /usr/jatoba-5/bin
File Edit View Search Terminal Help
root@node1:/usr/jatoba-5/bin# systemctl status jatoba-5
● jatoba-5.service - Jatoba 5 database server
   Loaded: loaded (/usr/lib/systemd/system/jatoba-5.service; enabled; vendor pre
   Active: active (running) since Thu 2023-06-15 06:07:30 PDT; 1min 17s ago
     Docs: https://www.gaz-is.ru/Jatoba/doc
   Process: 7915 ExecStartPre=/usr/jatoba-5/bin/jatoba-check-db-dir ${PGDATA} (co
 Main PID: 7923 (postmaster)
    Tasks: 6 (limit: 8803)
   CGroup: /system.slice/jatoba-5.service
           └─7923 /usr/jatoba-5/bin/postmaster -D /var/lib/jatoba/5/data/
             └─7924 postgres: checkpointer
               └─7925 postgres: background writer
                 └─7927 postgres: walwriter
                   └─7928 postgres: autovacuum launcher
                     └─7929 postgres: logical replication launcher

Jun 15 06:07:30 node1 systemd[1]: Starting Jatoba 5 database server...
Jun 15 06:07:30 node1 postmaster[7923]: 2023-06-15 06:07:30.727 PDT [7923] LOG:
Jun 15 06:07:30 node1 postmaster[7923]: 2023-06-15 06:07:30.728 PDT [7923] LOG:
Jun 15 06:07:30 node1 postmaster[7923]: 2023-06-15 06:07:30.728 PDT [7923] LOG:
Jun 15 06:07:30 node1 postmaster[7923]: 2023-06-15 06:07:30.732 PDT [7926] LOG:
```

Рисунок 2.10 – Проверка статуса службы jatoba-6

- 17) Установить пароль для системного пользователя ОС «postgres»:

```
sudo passwd postgres
```

```
node1@node1: ~
File Edit View Search Terminal Help
node1@node1:~$ sudo passwd postgres
[sudo] password for node1:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
node1@node1:~$
```

Рисунок 2.11 – Установка пароля для пользователя ОС

- 18) Авторизоваться в psql:

Нажать сочетание клавиш:

```
CTRL + D
```

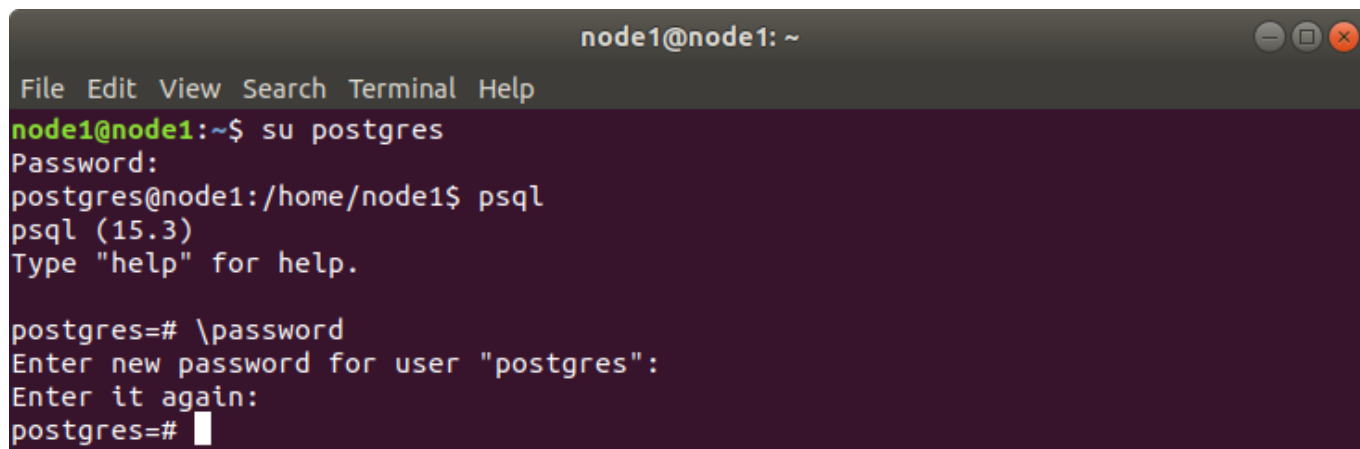
Войти в psql:

```
su -l postgres
```

```
psql
```

19) Установить пароль для пользователя СУБД «postgres»:

```
\password
```



```
node1@node1: ~  
File Edit View Search Terminal Help  
node1@node1:~$ su postgres  
Password:  
postgres@node1:/home/node1$ psql  
psql (15.3)  
Type "help" for help.  
  
postgres=# \password  
Enter new password for user "postgres":  
Enter it again:  
postgres=#
```

Рисунок 2.12 – Установка пароля для пользователя ОС

На этом этапе установку СУБД можно считать оконченной. Можно переходить к установке и конфигурированию кластера «jaDog».

## 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
FO	–	Failover – процедура обработки отказа узла/узлов кластера
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
ТО	–	Режим технического обслуживания
УЗ	–	Учетная запись
УЦ	–	Удостоверяющий центр
ЭВМ	–	Электронно-вычислительная машина

[illegible]

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

