Luxms BI

Руководство системного администратора

УСТАНОВКА

НАСТРОЙКА

ОБНОВЛЕНИЕ

РЕЗЕРВНОЕ КОПИРОВАНИЕ

МОНИТОРИНГ

2024-04-02

Оглавление

Введение

Документ подготовлен для системных администраторов, которые занимаются планированием и подготовкой инфраструктуры, развёртыванием и эксплуатацией программного обеспечения «Визуальный управленческий контроль Luxms BI» (далее – Luxms BI). Документ описывает:

- Организацию доступа к пакетным репозиториям.
- Установку компонентов системы из пакетов и их настройку.

Также документ содержит необходимую техническую информацию по обеспечению отказоустойчивости, резервному копированию и мониторингу ПО.

Документ предполагает наличие базовых знаний в области администрирования серверных операционных систем на базе Linux. А именно:

- Навыки работы в shell операционных систем Linux.
- Навыки работы с пакетными менеджерами ОС Linux.
- Навыки настройки и сопровождения Web-сервера NGinx.
- Навыки эксплуатации базы данных PostgreSQL.

1. Описание системы Luxms BI

Платформа Luxms BI создана для интерактивной визуализации данных с целью проведения экспресс-анализа их структуры и динамики. Реализовано на 3-х слоях:

- Представление (Front-End Layer).
- Обработка (Back-End Layer).
- Хранение (Storage Layer).

1.1. Слой “Представления”

1. Реализован на базе NGinx (компонент `luxmsbi-web`) и усилен использованием Lua-скриптов. Дополнительно, функционал Front-End-а использует HTTP API компонента `luxmsbi-appserver` (Java). Предоставляет следующие интерфейсы:

- HTTP/HTTPS - 80,443/TCP.

Требует взаимодействия с:

- KeyDB сервером.
- БД Luxms BI.
- Компонентом `luxmsbi-appserver`.
- Компонентом `luxmsbi-datagate`.
- Компонентом `luxmsbi-importer`.

Компонент `luxmsbi-web` включает в себя базовые конфигурационные файлы для организации SSO-авторизации и шифрование сессий с помощью SSL.

2) `luxmsbi-bins` (Javascript) - предоставляет функционал обработки WebSocket соединений с браузером клиента. Имеет следующие интерфейсы:

    - HTTP API - 8888/TCP.

Требует взаимодействия с:

- KeyDB сервером;
- БД Luxms BI.

1.2. Слой “Обработки”

Текущая версия требует наличие только компонента `luxmsbi-appserver`, поставляемый с пакетом **luxmsbi-appserver-mono**, который включает в себя консолидированный функционал по управлению системой и взаимодействия с источниками данных.

Но при необходимости выноса взаимодействия с внешними источниками данных, можно использовать выделенные компоненты: - `luxmsbi-appserver`, поставляемый пакетом **luxmsbi-appserver**, без функционала работы с Источниками данных; - `luxmsbi-datagate`, поставляемый пакетом **luxmsbi-datagate**.

Компонент `luxmsbi-importer` - устаревщий компонент для обеспечения интеграции с внешними Источниками данных. Поддерживаемый для существующих Клиентских инсталяций, но не рекомендуемый для новых развертываний.

1) `luxmsbi-appserver` (Java) - предоставляет API для управления настройками приложения Luxms BI и функционал работы с Источниками данных, в консолидированном варианте. Имеет следующие интерфейсы:

    - HTTP API - 8080/TCP;

- RSocket - 7200/TCP, при использовании консолидированного компонента **luxmsbi-appserver-mono**.

Требует взаимодействия с:

- NATS сервером;
- БД Luxms BI.

2) `luxmsbi-datagate`(Java) - обеспечивает взаимодействие со сторонними источниками данных. Имеет следуюшие интерфейсы:

- HTTP API - 8200/TCP.
- RSocket - 7200/TCP.

Требует взаимодействия с:

- NATS сервером;
- БД Luxms BI.

3) `luxmsbi-importer`(Java) - реализует функционал импорта, обработки и загрузки данных по расписанию. Имеет следующие интерфейсы:

- HTTP API - 8192/TCP;
- RSocket - 7192/TCP.

Требует взаимодействия с:

- KeyDB сервером;
- NATS сервером;
- БД Luxms BI;
- Компонентом `luxmsbi-appserver`;
- Компонентом `luxmsbi-datagate`.

1.3. Слой Хранения

1. Хранение данных реализовано на **PostgreSQL** - свободная объектно-реляционная система управления базами данных. Предоставляет другим компонентам интерфейс доступа:

- PostgreSQL - 5432/TCP.

Требует взаимодействия с:

- KeyDB сервером.
- Компонентом `luxmsbi-appserver`.
- Компонентом `luxmsbi-datagate`.

Компоненты Luxms BI поддерживают подключение к БД с использованием SSL-шифрования, но для снижения ресурсной нагрузки рекомендуется использовать нешифрованные соединения, особенно во внутренних закрытых сегментах сети.

Для предоставления импортозамещающего решения дистрибутивы для

- Astra Linux Special Edition
- RedOS Linux

использует российскую СУБД **Postgres Pro** (входит в Единый реестр, имеет сертификат ФСТЭК).

2. NATS сервер - распределённое хранилище объектов, в том числе в формате key:value и брокер функционала pub/sub. Используется для консолидации конфигурационных объектов и обеспечения доступности сгенерированных отчетов в распределенной архитектуре. Предоставляет другим компонентам интерфейсы доступа:

- NATS cluster protocol - 6222/TCP
- NATS client protocol - 4222/TCP

1.4. ETL-решение

`luxms-databoring`(NodeJS) - обеспечивает исполнения сценариев получения и трансформации данных из сторонних источников данных. Имеет следующие интерфейсы:

- HTTP API - 1880/TCP.

Требует взаимодействия с:

- Компонентом `luxmsbi-importer`, если он используется;
- Компонентом `luxmsbi-appserver` и/или `luxmsbi-datagate`, если он используется;
- Компонентом `luxmsbi-web`.

2. Варианты развёртывания

Luxms BI может быть поставлен в различных вариантах:

1. В виде VMWare Appliance - для обучения и тестирования. Это виртуальная машина ESXi с установленными компонентами и демонстрационными датасетами на базе ОС CentOS7. Выбор ОС связан с требованиями Правообладателей по лицензированию ОС.

> Во втором квартале 2024 года этот вариант поставки будет использовать ОС Rocky Linux 9.

2. В виде образа жесткого диска, для развертывания в облачной инфраструктуре следующих провайдеров:

- Яндекс
- ВКонтакте
- СберКлауд

> В 2024 году появится возможность приобретения Приложения в облаке ВКонтакте - сейчас мы на стадии завершения тестирования.

3. Для развертывания ПО на внутренних, корпоративных облачных решениях, мы рекомендуем производить установку с учётом требований резервирования и масштабирования компонентов для поддержания высокой нагрузки и отказоустойчивости.

Для каждого Клиента, купившего корпоративную лицензию, мы и наши официальные Партнеры прорабатываем архитектуру развёртывания и предоставляем в виде графических схем и сопутствующей документации. После её согласования предоставляем Ansible-сценарии для автоматизации развёртывания.

Ниже приведены несколько вариантов архитектуры, предлагаемых клиентам - базовые варианты. Окончательный вариант архитектуры будет зависеть от расчетов, основанных на:

- объеме Ваших данных;
- количества конкурентных пользователей;
- сложности элементов визуализации в Атласах.

> Просим учитывать - группировка хостов скрывает протоколы взаимодействия между узлами одной группы. Просим изучить дополнительно соответствующие Приложения к данному руководсту, которые содержат полное описание взаимодействия

2.1. Конфигурация для сред Разработки

Оптимальная конфигурация для выполнения разработки. Не включает в себя компоненты для интеграции с корпоративным каталогом учетных записей и поддержку Web-Socket для браузеров. Достаточная для разработки и настройки визуализации данных.

Схема не включает в себя решения для “горячих данных”, но рекомендуется дополнительное подключение к экземпляру Clickhouse.

Рис. 2.1. **Пример среды для разработки**

2.2. Минимальная продуктовая конфигурация

Минимальное резервирование компонентов для продуктовой конфигурации. Эта схема обеспечивает резервирование компонентов и отказоустойчивость среды. Количество конкурентных пользователей ограничено и зависит от объема данных, подлежащих визуализации и сложности Атласов(количества элементов визуализации).

Схема балансировки, на базе HAProxy, приведена как возможное, бесплатное решение. Мы можем рекомендовать это решение как работоспособное, но без гарантии его достаточности в Вашей конкретной продуктовой среде.

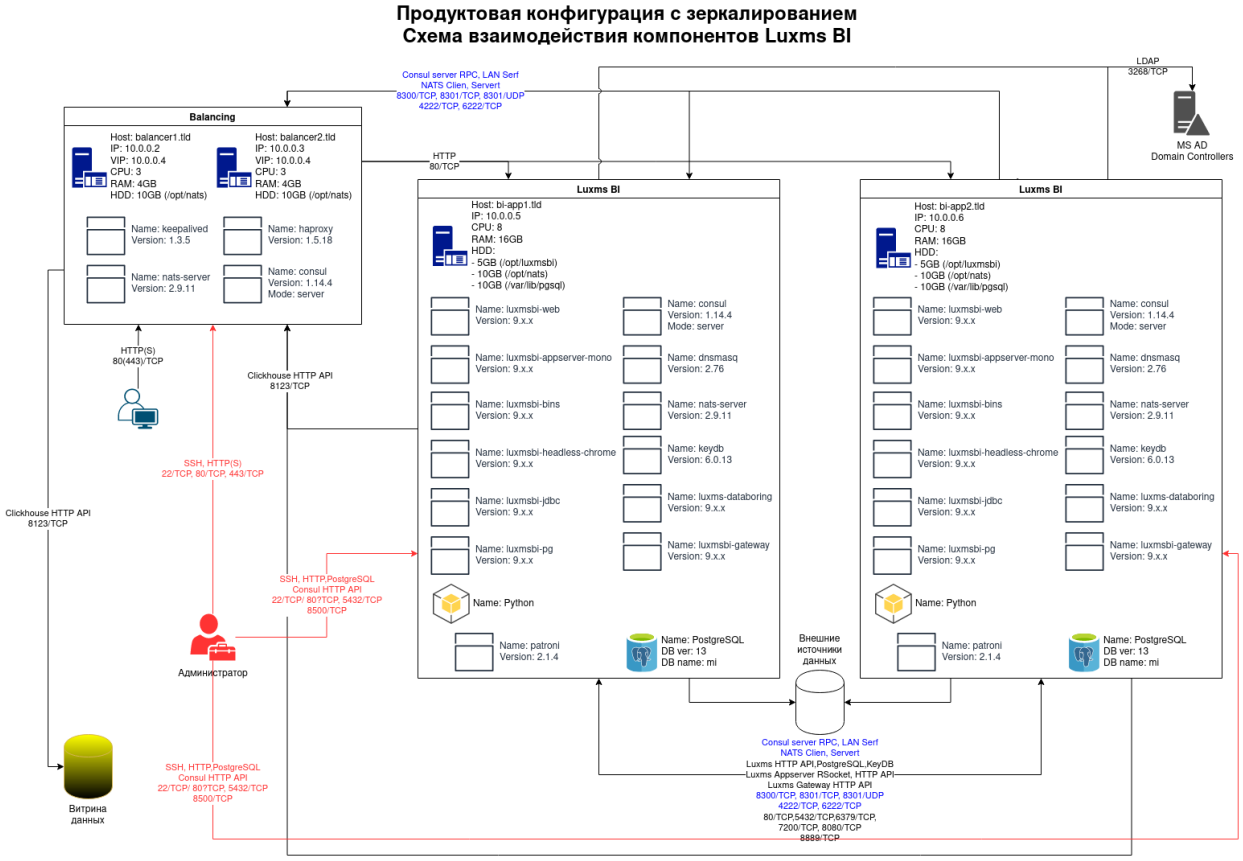


Рис. 2.2. **Пример продуктовой среды с резервированием**

2.3. Продуктовая конфигурация

Конфигурация, поддерживающая отказоустойчивость и возможность масштабирования решения с учетом растущей нагрузки. Количество узлов каждого типа не ограничено.

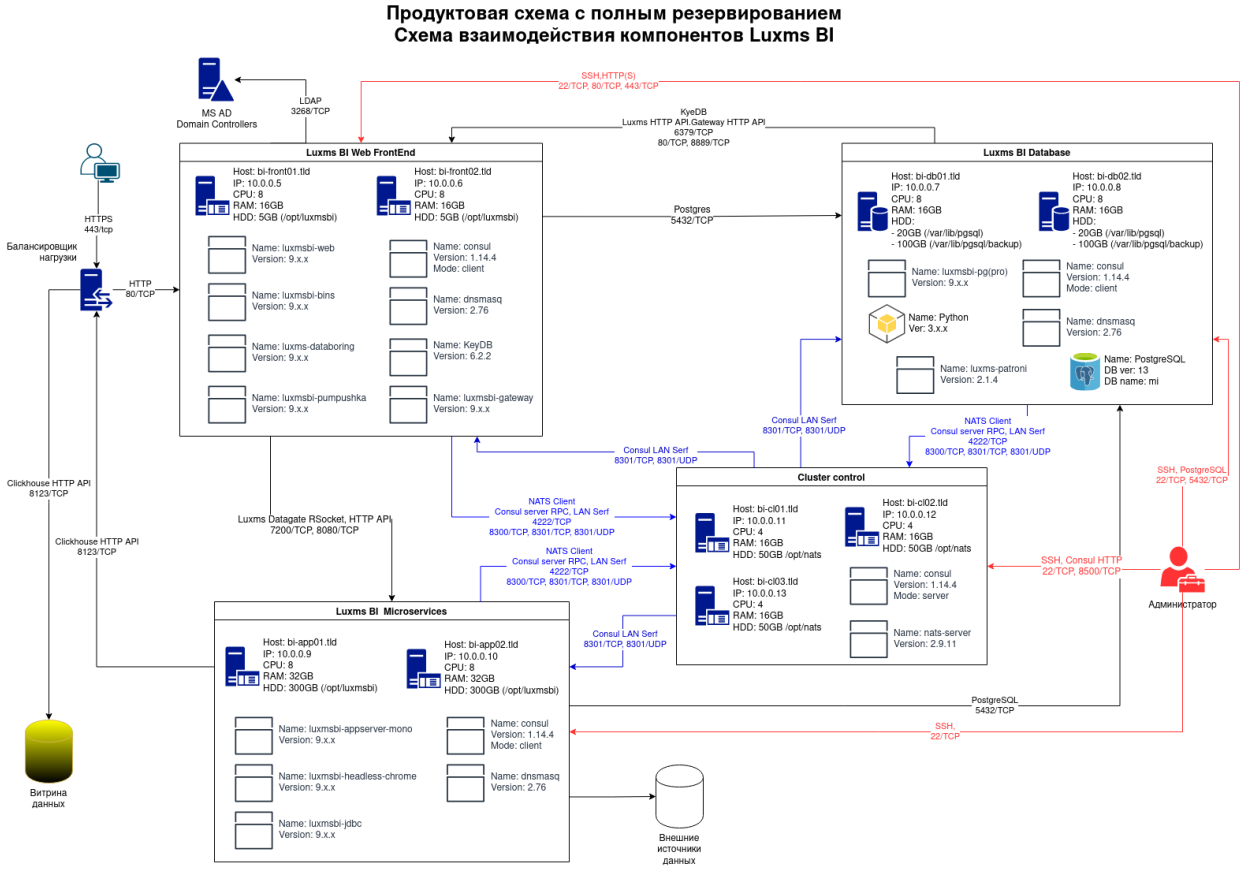


Рис. 2.3. **Пример продуктовой среды с возможностью масштабирования**

3. Терминология

ОС - операционная система, под управлением которой работает хост/узел.

RPM-based (*RedHat-based*) - операционные системы Linux, основанные на открытом коде операционной системы Red Hat Enterprise Linux.

Deb-based (*Debian-based*) - операционные системы Linux, основанные на открытом коде операционной системы Debian Linux.

Front-end (Клиент) – веб-приложение Luxms BI для пользователей и администраторов, реализованное в виде HTML5/Javascript приложения для браузеров.

DB (*БД*, так же *База данных*) - база данных, в данном документе под *Базой данных* имеются ввиду экземпляры PostgreSQL или PostgreProSQL.

Административная панель – часть Front-end Luxms BI, предназначенная для управления учётными записями, датасетами, дэшбордами, правами доступа, подключениями к источникам данных, кубами и загрузками.

Администратор – именованный пользователь с доступом на чтение через пользовательский интерфейс, а также расширенным доступом на управление учетными записями и правами доступа, датасетами и дэшбордами, подключениями к источникам, кубами и загрузками через административную панель Luxms BI.

Браузер – программа для работы с Web ресурсами.

Атлас (устаревшее *DataSet*, также *Набор данных*) – логическая единица хранения агрегированных данных, готовых дэшбордов и их настроек, полностью подготовленных для показа на Front-end.

Импорт – операция по добавлению данных или документов в слой горячих данных, в редких случаях в Атлас.

Источник данных – любое хранилище данных, в том числе файл Excel или CSV.

Пользователь – именованный пользователь с доступом на чтение через пользовательский интерфейс Luxms BI.

Права доступа – совокупность правил, регламентирующих условия доступа пользователя к ресурсам ОС.

Учётная запись – совокупность сведений об именованном пользователе, необходимая для его аутентификации.

4. Рекомендации по развёртыванию Luxms BI

Данные рекомендации предназначены для ознакомления Клиентов с требованиям Luxms BI при первичном развёртывании. В документе перечислены точки монтирования файловой системы, предполагающие рост объёма хранимых данных и рекомендации по организации файловых систем.

Указанные в документе числовые значения носят рекомендательный характер и не могут быть применены для промышленной эксплуатации. Но предлагаемые решения по управлению ресурсами позволят безболезненно увеличить необходимые параметры систем.

При необходимости вы можете запросить расчёт требований необходимых ресурсов (сайзинг) и архитектуру реализации решения у Продавца решения или у Производителя.

4.1. Пакетные репозитории

Программное обеспечение Luxms BI доставляется с использованием пакетных репозиториев под следующие операционные системы:

- CentOS 7, совместимо с ОС RHEL 7, OracleLinux 7;
- Astra Linux Special Edition 1.7;
- RedOS Linux 7.3.x.
- Rocky Linux 8, совместимо с ОС RHEL 8, OracleLinux 8

Доступ к репозиториям для Клиентов предоставляется с использованием аутентификации.

4.2. Запуск компонентов на одном хосте

Luxms BI может успешно работать при развертывании на одном хосте при небольшой пользовательской нагрузке, обычно до 100 активных пользователей. Но для гарантирования доступности приложения и при количестве активных пользователей больше 100 рекомендуется использование горячего резервирования:

- Дублирование сервисов Luxms BI (горячее резервирование);
- Организация кластера базы данных.

Кластеризация и дублирование компонентов позволит вам обеспечить доступность системы не только при нештатных ситуациях (отказ серверного оборудования), но и при проведении регламентных работ на ОС и при установке обновлений Luxms BI.

4.2.1. Требования к вычислительным ресурсам

Рекомендуемые ресурсы для одноузловой системы в среде виртуализации:

- От 8 virtual CPU;
- От 32GB virtual RAM.

Мы предоставляем демонстрационный образ виртуальной машины с меньшими ресурсами, но для обеспечения надежности работы решения в Вашей ИТ-инфраструктуре рекомендуем вам запросить расчет требований необходимых ресурсов.

4.2.2. Рекомендации по организации файловой системы

Выделение файловых систем под различные точки монтирования обеспечивает стабильную работу ОС, независимо от заполнения файловой системы в этих разделах. Определение типового разбиения стандартной файловой системы для хостов с компонентами Luxms BI определяется внутренней политикой клиента или отраслевыми стандартами.

Для Luxms BI мы определяем следующие дополнительные минимальные требования к конфигурации файловой системы:

```
/opt/luxmsbi                        - 2GB   LVM  EXT4
/opt/nats                           - 10GB  LVM  EXT4
/var/lib/pgsql или /var/lib/pgpro   - 10GB  LVM  EXT4
/var/log                            - 8GB   LVM  EXT4
```

Предлагаемые минимальные значения могут быть недостаточными для вашей инсталяции. Размер файловой системы для указанных точек монтирования зависит от планируемой нагрузки на систему Luxms BI.

4.2.3. Пояснения к рекомендациям по файловой системе

Рекомендуем использовать менеджер логических дисков (LVM).

> Рано или поздно возникает необходимость оперативного добавления места в файловой системе. И наиболее простой способ для решения этого вопроса - добавление или расширение физического тома (PV), с последующим расширением VG/LV и файловой системы.

Для обеспечения работы Luxms BI настоятельно рекомендуем выделить отдельные файловые системы для следующих нужд:

1. **Файлы базы данных - начните с 10ГБ.**

Рекомендуем использование отдельного раздела файловой системы для хранения файлов базы данных PostgreSQL:

- Рекомендуемая файловая система EXT4 (сравнение производительности других файловых систем не показывает существенного повышения производительности и/или возможностей файловой системы).
- Рекомендуемые параметры монтирования - отключите фиксацию времени доступа к файлу (**noatime**).
- Рекомендуемая точка монтирования - использование стандартного пути расположения для PostgreSQL `/var/lib/pgsql/` или другой точки монтирования, регулируемой внутренней политикой клиента.
- Рекомендуемый размер файловой системы - зависит от планируемого объема Ваших данных.

2. **Раздел приложения `/opt/luxmsbi` - начните с 2 ГБайт.**

Кроме хранения файлов приложения, данный раздел предполагает место для временных файлов при импорте транспортных файлов с данными.

3. **Распределенное хранилище объектов - начните с 10Гбайт.**

В этом разделе файловой системы будут храниться шаблоны для отчетов, и сами сгенерированные очеты. Также используется для временного хранения транспортных файлов ETL-процессов. Размер данного раздела зависит от использования функционала Luxms BI в конкретной инсталяции.

4. **Журнальные файлы приложений - начните с 8 Гбайт.**

Компонеты Luxms BI используют два варианта журналирования:

- Журналирование на файловую систему `/var/log/luxmsbi`.

В этом разделе файловой системы хранятся журнальные файлы Web-сервера NGinx. Размер файловой системы определяется установленными параметрами журналирования. В большинстве случаев раздел `/var/log` может не требовать монтирования дополнительного раздела диска. Но мы это **рекомендуем** для избежания необходимости переноса приложения Luxms BI на другой хост, с большим разделом для журналирования событий.

- Журналирование событий Systemd Journal `/var/log/journal`.

Большинство компонентов Luxms BI используют системную службу журналирования Journald. Размер файловой системы для хранения этих данных зависит от ваших корпоративных требований, от объема подгружаемых в систему данных из сторонних источников. Поэтому настоятельно рекомендуем монтирование отдельной файловой системы в данной точке.

Просим дополнительно ознакомиться с Приложением #2.

4.3. Масштабирование сервисов Luxms BI

Выделение отдельных хостов для разных сервисов Luxms BI требуется при инсталяции решений, обрабатывающих значительные объемы данных или большого числа одновременных пользователей. Для использования в условиях высокой нагрузки мы предлагаем разнесение компонетов системы на следующие уровни:

- Уровень Базы данных;
- Уровень Приложения;
- Вспомогательный уровень загрузки агрегированных данных;
- Вспомогательный уровень доступа к внешним источникам данных.

4.3.1. Выделенные сервера Базы данных

На текущий момент продуктовое решение использует в качестве базы данных:

RedHat-based дистрибутивы:

- **PostgreSQL 11**;
- **PostgreSQL 13**.

AstraLinux Special Edition 1.7:

- **Postgres Pro 13**;
- **PostgreSQL 13**.

RedOS 7.3.1:

- **Postgres Pro 13**;
- **PostgreSQL 13**.

Для обеспечения доступности системы и резервирования данных рекомендуется использование кластеризации базы данных. В качестве кластерного решения мы рекомендуем Patroni с использованием HarshiCorp Consul как управляющего кластера.

Вне зависимости от использования решения кластеризации для базы данных, мы настоятельно рекомендуем использование отделной файловой системы для файлов базы данных. Размер файловой системы зависит от планируемого объема обрабатываемых данных.

Дополнительные рекомендации:

> Не рекомендуем использовать архивирование журнальных файлов Базы данных. Luxms BI в большей степени аналитическая система и предполагает периодичную пакетную загрузку больших объемов данных. Вероятность необходимости откатить состояние Базы данных на какой-то определенный момент времени в прошлом при пакетной загрузке консолидированных данных очень мала - намного дешевле и быстрее выполнить повторную загрузку данных. А накапливание архивированных журналов может неожиданно остановить работу системы, если архивные журналы по какой-то причине заполнят файловую систему Базы данных.

Резервное копирование Базы данных определяется внутренней политикой Клиента. Мы рекомендуем ежедневное снятие резервных копий и небольшой срок хранения резервных копий - от 3 до 7 дней.

Для надёжной работы кластеров требуется точная синхронизация времени на различных серверах. Если используется среда виртуализации (например, VMWare), то, обычно, это обеспечивается через настройки виртуализатора. Если же ОС запускается на железе, то требуется установка и настройка NTP сервисов на каждом сервере кластера.

4.3.2. Выделенные сервера приложений

Выделение отдельных серверов для уровня Приложений позволяет обеспечить балансировку нагрузки между несколькими узлами, повысить доступность системы при нештатных ситуациях и обеспечить возможность проведения работ по обслуживанию узлов без ограничения доступа к Luxms BI.

В качестве решений по балансировке нагрузки могут быть использованны различные аппаратные и программные комплексы, работающие с HTTP(S) трафиком.

Для уровня Приложений существует две точки монтирования, где возможен рост использования файловой системы:

1. Журнальные файлы приложений - `/var/log`.

Необходимый размер файловой системы для журнальных файлов полностью зависит от объема загружаемых данных и установленного уровня журналирования. Использование LVM менеджера упростит решение вопросов по увеличению размера файловой системы, поэтому мы рекомендуем создать файловую систему с минимальным размером в 8 ГБайт и предусмотреть возможность увеличения размера за счет добавления дополнительных дисковых устройств.

2. Файлы данных и отчеты - `/opt/nats`.

При загрузке данных в Luxms BI из файлов загруженные файлы временно сохраняются в файловой системе для обеспечения возможности анализа первичных данных после загрузки. При генерации отчетов и презентаций результаты генерации также сохраняются в распределенной файловой системе и доступны пользователям через Web-приложение Luxms BI.

4.3.3. Выделенные сервера для импорта и доступа к данным

Развертывание компонентов Luxms BI Importer и Datagate может быть необходимым для обеспечения доступа к данным, расположенным в сетях с ограничением доступа. Установка выделенного сервера на границе закрытого сегмента сети позволит обеспечить безопасность критичных данных.

Рекомендации по размеру файловой системы для Luxms BI Importer и Datagate идентичны рекомендациям по выделенным серверам Приложений.

4.4. Использование SELinux и Firewalld/UFW

При установке приложений мы не рекомендуем Клиентам отключение стандартных средств защиты операционной системы Linux.

RPM-пакеты приложения Luxms BI и Инструкции по установке содержат конфигурационные файлы и рекомендации по настройке SELinux и Firewalld.

DEB-пакеты приложения Luxms BI содержат пост-инсталляционные скрипты настраивающие UFW, если он активен на сервере.

В сценариях установки Ansible мы предусматриваем также использование клиентом IPTables, но рекомендуем переход на Firewalld/UFW - эти решения служат упрощенным фронт-ендом для работы с функциями фильтрации сетевого трафика, поставляются производителеми ОС по-умолчанию.

Современные дистрибутивы ОС Linux включают в свое ядро API nftables. Официальная документация которого содержит 450 страниц. Конечно **nftables** позволяет производить более тонкую настройку сетевой фильтрации, но в 99% случаев этот избыточный функционал Вам не понадобится.

Мы рекомендуем использовать средства защиты встроенные в операционную систему как обязательное дополнение к существующим решениям защиты в инфраструктуре клиентов. Увеличение количества уровней защиты ИТ-решений:

- Увеличивает вероятность обнаружения злоумышленника;
- Снижает эффективность атаки и вероятность доступа к информации.

При установке ***Luxms BI*** из пакетных репозиториев в ПО регистрируется учетная запись Администратора системы(**adm**) с паролем по умолчанию **luxmsbi**. Мы настоятельно рекомендуем изменить пароль для этой учетной записи.

5. Использование пакетных менеджеров и репозиториев

Установка из компонентов Luxms BI с использованием пакетов облегчает развертывание, обновление и восстановление на предыдущую версию компонентов. Все пакеты компонентов поддерживают версию пакета, состоящую из 3 чисел, и временную метку, содержащую дату сборки пакета.

При установке пакетов Luxms BI может потребоваться доступ к публичным репозиториям стороннего программного обеспечения. Информация о необходимых дополнительных репозиториях указана далее для всех компонентов.

Доступ к пакетным репозиториям может быть настроен несколькими путями, мы рассмотрим два наиболее популярных:

- создание собственного Инфраструктурного решения по зеркалированию пакетных репозиториев;
- подключение к пакетным репозиториям Производитея ПО.

> Рекомендуем использование зеркалирования репозиториев для обеспечения независимости Клиентов от стабильности каналов связи и для исключения многократной утилизации канала связи для повторного получения пакетов.

В качестве решений по реализации собственного зеркала пакетных репозиториев можем порекомендовать следующий перечень решений:

1. Foreman(with Katello plugin) - бесплатное OpenSource решение предоставляющее, кроме управления зеркалами репозиториев, функционал управления серверами. Поддерживает как YUM, так и DEB-репозитории.

2. Sonatype Nexus repository OSS - бесплатный вариант проксирующего сервера контента. Может быть использован как проксирующий сервер для различных артефактов , включая Maven, PyPi, NodeJS и других.

3. Aptly бесплатное OpenSource решение для полноценного управления Deb-репозиториями.

5.1. Подключение к собственному зеркалу репозиториев

В зависимости от действующего в Вашей ИТ-Инфраструктуре решения по организации доступа к пакетным репозиториям, подключение хостов должно производиться в соответсвии с внутренними нормативными документами. И не может быть произведено с помощью “релизных” пакетов из репозиториев Производителей ПО, в том числе и к зеркалам репозиториев Luxms BI.

5.2. Подключение к репозиториям Luxms BI

Для подключения к персонализированному репозиторию необходимы аутентификационные данные, выдаваемые клиентам.

Репозитории Luxms BI содержат “релизные” пакеты:

- для RPM-based ОС - `luxmsbi-release-[version]-[release].noarch.rpm`
- для Deb-based ОС - `luxmsbi-release_[version]-[release]_amd64.deb`

значения для `[version]` и `[release]` обозначают версию Luxms BI и дату выпуска пакета.

При поключении к репозиториям Производителя Вам необходимо настроить и сохранить аутентификационные данные, для доступа к репозиториям, на каждом подключаемом хосте.

Рекомендуем вам загрузить самый свежий пакет для вашей ОС через Web-интерфейс сервера обновлений Luxms BI.

5.2.1. Обновление корневых сертификатов

Подключение репозиториев или установка из них пакетов может не работать на ваших серверах, если корневые сертификаты ОС давно не обновлялись вручную и/или не обновляются автоматически.

В этом случае вам нужно загрузить пакет с корневыми сертификатами вручную и установить его.

Для RPM-based ОС:

```
sudo dnf -y update ca-certificates
sudo update-ca-trust
```

Если Вы используете собственные центры сертификации, то необходимо создать файлы сертификатов корневого(и промежуточных) CA в папке `/etc/pki/ca-trust/source/anchors/` в PEM-формате, с расширением `.crt`. После чего выполнить ручное обновление сертификатов командой:

```
sudo update-ca-trust
```

Для Deb-based ОС:

```
apt update
apt install ca-certificates
```

Если Вы используете собственные центры сертификации, то необходимо создать файлы сертификатов корневого(и промежуточных) CA в папке `/usr/local/share/ca-certificates/` в PEM-формате, с расширением `.crt`. После чего выполнить ручное обновление сертификатов командой:

```
update-ca-certificates
```

5.2.2. Пакетное подключение репозиториев

Установка “релизного” пакета, опубликованного в репозитории, позволяет Вам установить публичный ключ GPG, для обеспечения проверки целостности пакетов, и создаст шаблон конфигурационного файла репозиториев.

Но настройка аутентификационных данных для доступа к репозиториям должна быть произведена на каждом хосте в ручном режиме. Описание настройки приведено в следующих разделах. Пакеты Luxms BI не поставляются с критичными аутентификационными данными, тем более что каждому Клиенту предоставляется возможность смены секретной фразы для доступа к репозиториям.

Установите эти пакеты в ОС ваших серверов, используя необходимый пакетный менеджер - `apt` или `dnf(yum)`.

Если по каким-либо причинам вы не можете установить релизный пакет, то настройка доступа к ним может быть проведена в ручном режиме.

5.2.3. Настройка подключение к YUM-репозиторию

“Релизный” пакет устанавливает шаблон конфигурации для подключения к YUM-репозиторию `/etc/yum.repos.d/luxmsbi.repo`, который выглядит следующим образом:

```
[luxms-thirdparty]
name=Luxms 3rd-party packages
baseurl=https://download.luxms.com/repository/thirdparty/el/$releasever/$basearch/
enabled=1
gpgcheck=0
repo_gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms

[luxms-bi9]
name=Luxms BI 9 Repository
baseurl=https://download.luxms.com/repository/[REPO]/9/el/$releasever/$basearch/
enabled=1
gpgcheck=1
repo_gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms
#username=
#password=
```

Для окончательной настройки доступа к репозиториям LuxmsBI необходимо:

- Указать имя репозитория в параметре `baseurl=` вместо `[REPO]` (иногда название репозитория совпадает с именем учетной записи);
- Удалить знак комментария и указать имя учетной записи в параметре `username=`;

- Удалить знак коментария и указать пароль учетной записи в параметре `password=`.

> При развертывании на других RPM-based дистрибутивах необходимо корректно изменить URI с учетом значения переменной `$releasever` для конкретной ОС. Например для Oracle Linux 7, поможет простая замена на статическую версию:
> `baseurl=https://download.luxms.com/repository/[REPO]/9/el/↩ 7/$basearch/`
> `baseurl=https://download.luxms.com/repository/thirdparty/↩ 8/el/7/$basearch/`

Для облегчения процесса получения RPM-пакетов программного обеспечения с открытым исходным кодом сторонних разработчиков ПО, дополнительно настраивается репозиторий `luxms-thirdparty`. Этот репозиторий не требует аутентификации.

5.2.4. Настройка подключения к DEB-репозиторию

“Релизный” пакет устанавливает шаблон конфигурации для подключения к репозиторию `/etc/apt/source.list.d/luxmsbi.list`. Может содержать одну или несколько строк, выглядит следующим образом:

```
# Replace password with yours in links.
deb [ arch=amd64 ] https://[customer]:[password]↩
  @download.luxms.com/repository/alse-bi9 1.7_x86-64 main
```

где:

- `[customer]` - учетная запись клиента, для доступа к репозиториям;
- `[password]` - пароль к учетной записи.

Количество строк в конфигурационном файле зависит от наличия заказной разработки ПО. Если Вы получаете от нас дополнительное ПО, то к основному репозиторию может быть добавлен персонализированный, например:

```
deb [ arch=amd64 ] https://[customer]:[password]↩
  @download.luxms.com/repository/alse-[customer] 1.7_x86-64 main
```

Для окончательной настройки доступа к репозиториям Luxms BI необходимо:

- Проверить корректность имени учетной записи в URL-е репозитория, вместо **[customer]**.
- Заменить **password** на пароль учетной предоставленной записи.

Безопасность аутентификационных данных для доступа к репозиториям может быть обеспечена использованием apt_auth.conf. Также можно воспользоваться командой `man apt_auth.conf`

Аутентификационные данные для репозитория Luxms BI устанавливаются в конфигурационном файле `/etc/apt/auth.conf.d/luxmsbi.conf`:

В примере использованы нереальные значения для учетной записи и пароля. Получите Ваш пароль через менеджера партнера или производителя Luxms BI.

```
machine download.luxms.com
login AstraLinux
password CoolPassword
```

После настройки Вы можете смело поменять разрешения на конфигурации файлов, содержащие пароли:

```
chmod 600 /etc/apt/auth.conf /etc/apt/auth.conf.d/*.conf
chmod 700 /etc/apt/auth.conf.d
```

Замечена неработоспособность при использовании специального символа “@”, поэтому при настройке рекомендуем проверить совместимость содержимого пароля с этим функционалом.

5.2.5. Настройка верификации пакетов

Ручная установка репозитория требует дополнительно регистрацию публичного PGP-ключа для проверки цифровой подписи пакетов при установке.

Для RPM-based ОС:

```
sudo curl -o /etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms \
          https://download.luxms.com/repository/thirdparty/RPM-GPG-KEY-Luxms
sudo rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms
```

Для Deb-based ОС:

```
wget -q -O - \
     https://download.luxms.com/repository/thirdparty/RPM-GPG-KEY-Luxms \
     | sudo apt-key add -
```

После настройки репозитория рекомендуем обновить локальный кеш пакетов:

```
sudo apt update
```

6. Установка и настройка сервера БД

Наше ПО совместимо с любыми деривативами PostgreSQL, но все же, для его функционирования, необходим набор расширений БД. Список расширений не большой, приведен ниже:

- PLV8 - это расширение для PostgreSQL, обеспечивающее использование Javascript-языка в процедурах и функциях БД, популярный opensource-проект;
- KeyDB-FDW - opensource расширение для работы с On-memory БД на базе безопасной и современной алтернативы Redis - сервера KeyDB;
- RedisPubSub - opensource расширение для БД позволяющее публиковать сообщения в каналах сервера KeyDB;
- PostgreSQL HTTP Client - популярное расширение БД PostgreSQL реализующее HTTP-client-а.

На текущий момент мы поставляем все эти расширения для следующих вариантов PostgreSQL DB:

- PostgreSQL 11, 13, 15. Для PostgreSQL 11 наступил EOL;
- PostgresPro 13, для Стандартной и Расширенной редакции;
- Jatoba 4 (за исключением jatoba4-http, стандартного расширения входящего в поставку этой БД)

Список совместимых движков БД не велик, но обоснован следующими фактами: - PostgreSQL - является свободно-распространяемой opensource Базой данных; - PostgresPro - Российское ПО, обладающее сертификатом ФСТЭК. Входит в Единый реестр Минкомсвязи. Совместим с СКЗИ «Крипто БД 2.0»; - Jatoba - Российское ПО, обладающее сертификатом ФСТЭК. Входит в Единый реестр Минкомсвязи. - все расширения проходят функциональное и нагрузочное тестирование. Индивидуально и в составе Luxms BI.

Для установки отказоустойчивого кластера PostgreSQL , дополнительно ознакомтесь с Приложением #1

6.1. Настройки файловой системы

При использовании нестандартного расположения файлов БД, например в соответсвии с внутренними нормативными документиами, или при монтировании отдельной файловой системы необходимо откорректировать настройки прав доступа. Например:

```
sudo chown -R postgres.postgres /data/pgdata
```

Нестандартное расположение файлов БД также потребует дополнительной настройки профиля(переменных окружения) владельца процессов и файлов БД.

Обращаем внимание, если каталог базы данных отличается от дефолтного, необходимо переопределить переменную окружения **PGDATA**

Изменения фиксируются в следующих местах:

- профайле сервисной учетной записи `postgres`
- файле переменных среды сервиса или в самого Systemd service unit,

6.2. Установка PostgreSQL

Подключение необходимых репозиториев для установки PostgreSQL зависит от ОС хоста, на котором производится установка.

Здесь и далее указываются URL официальных репозиториев для обозначения целевого контента, при наличии других публичных или локальных ***зеркал*** репозиториев, желательно использовать локальные ***зеркала***.

Использование зеркалированных репозиториев потребует откорректировать параметры команд, описанных ниже.

6.2.1. CentOS 7

Необходимо подключить дополнительные публичные репозитории или имеющиеся у клиента зеркала этих репозиториев:

- Extra Packages for Enterprise Linux 7 - x86_64;
- PostgreSQL 13 for RHEL / CentOS 7 - x86_64

1. Установка дополнительных репозиториев:

```
sudo yum -y install epel-release \
  https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-↩
  redhat-repo-latest.noarch.rpm
```

Опционально, можно отключить не планируемые к использованию версии PostgreSQL, оставив доступными пакеты только 13 версии:

```
sudo yum-config-manager --disable pgdg11 pgdg12 pgdg14 pgdg15
```

2. Установка пакетов сервера `postgresql-13`:

```
sudo yum -y install postgresql13 \
               postgresql13-server \
               postgresql13-contrib
```

3. Настройка профиля сервисной учетной записи

Мы рекомендуем создать файл профиля для сервисной учетной записи `postgres`:

```
echo -e '\nPATH=/usr/pgsql-13/bin:$PATH\nexport PATH\n' \
      | sudo /usr/bin/tee -a /var/lib/pgsql/.bash_profile
```

4. Инициализация PostgreSQL:

```
sudo -iu postgres initdb
```

Если предполлагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

4. Запуск postgresql сервиса:

```
sudo systemctl enable --now postgresql-13
```

После завершения установки, не забудьте выполнить проверку в соостветсвии с Проверка после установки

6.2.2. RedOS 7.3

Достаточно подключения стандартных репозиториев производителя ОС:

- RedOS - Base
- RedOS - Updates

1. Установка пакетов сервера `postgresql-13`:

```
sudo dnf -y install postgresql13 \
               postgresql13-server \
               postgresql13-contrib
```

2. Настройка профиля сервисной учетной записи

Мы рекомендуем создать файл профиля для сервисной учетной записи `postgres`:

```
echo -e '\nPATH=/usr/pgsql-13/bin:$PATH\nexport PATH\n' \
      | sudo /usr/bin/tee -a /var/lib/pgsql/.bash_profile
```

3. Инициализация PostgreSQL:

```
sudo -iu postgres initdb
```

Если предполлагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

4. Запуск postgresql сервиса:

```
sudo systemctl enable --now postgresql-13
```

После завершения установки, не забудьте выполнить проверку в соостветсвии с Проверка после установки

6.2.3. Astra Linux Special Edition 1.7

Этот вариант установки не полностью соответсвует рекомендациям ГК Астра и тем более не будет легитимным для повышенных уровней безопасности ОС, поскольку требует включения пакетов стороннего репозитория с более высоким приоритетом, чем пакеты ОС. Но на “базовом”(**Орёл**) уровне безопасности это допустимо.

Необходимо подключение публичного репозитория:

- [PostgreSQL Apt Repository] `deb http://apt.postgresql.org/pub/repos/apt buster-pgdg main`

1. Установка репозитория PostgreSQL:

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg main" \
     | sudo /usr/bin/tee -a /etc/apt/sources.list.d/pgdg.list
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc \
     | sudo apt-key add -
sudo apt update
```

После установки репозитория, необходимо создать дополнительный конфигурационный файл `/etc/apt/preferences.d/postgresql13` со следующим содержимым:

```
Package: *
Pin: origin apt.postgresql.org
Pin-Priority: 1001
```

2. Установка пакетов сервера `postgresql-13`:

Если нет требований по нестандартному расположению файлов БД, то достаточно одной команды:

```
sudo apt -y install postgresql-13
```

Для установки БД с измененным местоположением(и других параметров) можно после выполнения предыдущей команды пыполнить следующую последовательность:

```
sudo pg_lsclusters # посмотреть с какими параметрами поднята текущая БД

Ver Cluster Port Status Owner    Data directory              Log file
13  main    5432 online postgres /var/lib/postgresql/13/main ↩
  /var/log/postgresql/postgresql-13-main.log
```

```
sudo pg_ctlcluster 13 main stop # остановить существующую БД

sudo pg_dropcluster 13 main # удалить существующую БД

sudo pg_createcluster -d /data/pgdata 13 main # создать БД с новым расположением
   файлов

sudo pg_ctlcluster 13 main start # запустим новую БД
```

3. Настройка профиля сервисной учетной записи

Мы рекомендуем создать файл профиля для сервисной учетной записи `postgres`:

```
sudo install -o postgres -g postgres -m 600 \
             /etc/skel/.profile /var/lib/postgresql/
echo -e '\nPATH=/usr/lib/postgresql/13/bin/:$PATH\nexport PATH\n' \
      | sudo /usr/bin/tee -a /var/lib/postgresql/.profile
```

После завершения установки, не забудьте выполнить проверку в соответсвии с Проверка после установки

6.2.4. Rocky Linux 8

Доступ к пакетам PostgreSQL настраивается подключением ***виртуального репозитория(модуля)*** из физического репозитория ***AppStream***/

1. Настройка модуля(виртуального репозитория) PostgreSQL и подключение EPEL:

```
sudo dnf module enable postgresql:13
```

Использование публичного репозитория EPEL [Опцонально] Если Вы используете локальное “зеркало” для EPEL-репозитория, настройте подключение хоста к этому репозиторию. Если нет:

```
sudo dnf -y install epel-release
```

2. Установка пакетов

```
sudo dnf -y install postgresql \
                    postgresql-server \
                    postgresql-contrib
```

3. Инициализация БД:

```
sudo -iu postgres initdb
```

Если предполлагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

4. Запуск postgresql сервиса:

Обращаем внимание, если каталог базы данных отличается от дефолтного, необходимо переопределить переменную окружения **PGDATA** в systemd скрипте - `/lib/systemd/system/postgresql-13.service`

```
sudo systemctl enable --now postgresql
```

После завершения установки, не забудьте выполнить проверку в соответсвии с Проверка после установки

6.2.5. Rocky Linux 9 (в тестировании)

Доступ к пакетам PostgreSQL настраивается подключением ***виртуального репозитория(модуля)*** из физического репозитория ***AppStream***/

1. Настройка модуля(виртуального репозитория) PostgreSQL и подключение EPEL:

```
sudo dnf -y module enable postgresql:15
```

[Опцонально] Использование публичного репозитория EPEL Если Вы используете локальное “зеркало” для EPEL-репозитория, настройте подключение хоста к этому репозиторию. Если нет:

```
sudo dnf -y install epel-release
```

2. Установка пакетов

```
sudo dnf -y install postgresql \
                    postgresql-server \
                    postgresql-contrib
```

3. Инициализация БД:

```
sudo -iu postgres initdb
```

Если предполлагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

4. Запуск postgresql сервиса:

Обращаем внимание, если каталог базы данных отличается от дефолтного, необходимо переопределить переменную окружения **PGDATA** в systemd скрипте - `/lib/systemd/system/postgresql-13.service`

```
sudo systemctl enable --now postgresql
```

После завершения установки, не забудьте выполнить проверку в соответсвии с Проверка после установки

6.3. Установка PostgresPro

Использование PostgresPro для Российских компаний гарантирует возможность/непрерывность поддержки БД и защищает от санкционных рисков.

Luxms BI может быть развернут на PostgresPro двух редакций - Стандартной и Расширенной. Ниже приведенные примеры конфигураций написаны для Стандартной редакции. Для Расширенной редакции расположение файлов и имена комманд отличаются.

6.3.1. RedOS 7.3

Необходимо подключение публичного репозитория:

- PostgresPro Standard 13 for RedOS

1. Подключение репозитория PostgresPro

Самый простой вариант - использовать скрипт предлагаемый производителем:

```
curl https://repo.postgrespro.ru/pgpro-13/keys/pgpro-repo-add.sh | sudo sh
```

При использовании локальных “зеркал” получите инструкцию по подключению от администраторов Корпоративного сервера репозиториев.

2. Установка пакетов сервера `postgrespro-std-13`

БД можно установить в ***облегченном*** варианте, одной командой. Установка единственного пакета совмещена с инициализацией БД и настройкой авто-запуска сервиса. Но этот вариант не подходит при использовании нестандартного места расположения файлов БД.

```
sudo dnf -y install postgrespro-std-13
```

В большинстве продуктовых решений используется, отличный от стандартного, путь до файлов БД. В этом случае необходимо установить совокупность пакетов:

```
sudo dnf -y install postgrespro-std-13-server \
                    postgrespro-std-13-contrib
```

После такой установки Вам необходимо произвести инициализацию БД, и создать конфигурационный файл `/etc/default/postgrespro-std-13` и указать в нем значение для переменной PGDATA, например:

```
PGDATA=/data/pgdata
```

3. Настройка профиля сервисной учетной записи

Можно воспользоваться поставляевой PostgresPro утилитой `pg-wrapper`:

```
/opt/pgpro/std-13/bin/pg-wrapper links update
```

Но мы рекомендуем создать файл профиля для сервисной учетной записи `postgres`:

```
sudo install -o postgres -g postgres -m 600 \
             /etc/skel/.bash_profile \
             /etc/skel/.bashrc \
             /var/lib/pgsql/
echo -e '\nPATH=/opt/pgpro/std-13/bin:$PATH\nexport PATH\n' \
      | sudo /usr/bin/tee -a /var/lib/pgsql/.bash_profile
```

4. Инициализация PostgresPro:

```
sudo -iu postgres /opt/pgpro/std-13/bin/initdb
```

Если предполагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

Если предполлагается отличный от стандартного путь до файлов БД, укажите его для параметра `-D`.

5. Запуск и настройка автозапуска PostgresPro сервиса:

Для оригинальной версии PostgreSQL необходимо выполнить следующие действия:

```
sudo systemctl enable --now postgrespro-std-13.service
```

После завершения установки, не забудьте выполнить проверку в соответсвии с Проверка после установки

6.3.2. Astra Linux Special Edition 1.7

Необходимо подключение публичного репозитория:

- [PostgresPro Standard 13 for Astra Linus SE] `deb http://repo.postgrespro.ru/std-13/astra-smolensk/1.7 1.7_x86-64 main`

1. Подключение репозитория PostgresPro:

```
echo "deb http://repo.postgrespro.ru/std-13/astra-smolensk/1.7 1.7_x86-64 main" ↩
  \
     | sudo /usr/bin/tee -a  /etc/apt/sources.list.d/postgrespro-std-13.list
wget -q -O - https://repo.postgrespro.ru/pgpro-13/keys/GPG-KEY-POSTGRESPRO \
     | sudo apt-key add -
sudo apt update
```

2. Установка пакетов сервера `postgrespro-std-13`:

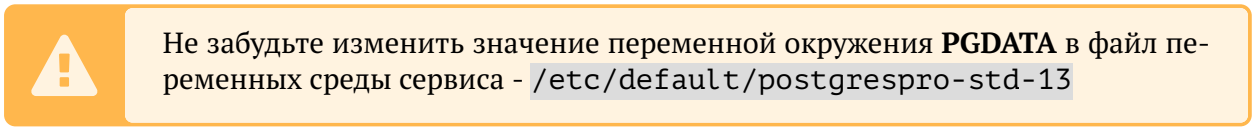
БД можно установить в ***облегченном*** варианте, одной командой. Установка единственного пакета совмещена с инициализацией БД и настройкой авто-запуска сервиса.Но этот вариант не подходит при использовании нестандартного места расположения файлов БД.

```
sudo apt -y install postgrespro-std-13
```

В большинстве продуктовых решений используется, отличный от стандартного, путь до файлов БД. В этом случае необходимо установить совокупность пакетов:

```
sudo apt -y install postgrespro-std-13-client postgrespro-std-13-server ↩
  postgrespro-std-13-contrib
```

После такой установки Вам необходимо произвести инициализацию БД, которая не означает автоматический запуск БД. Такой вариант установки особенно удобен при развертывании кластерного решения для БД, например Patroni.



Не забудьте изменить значение переменной окружения **PGDATA** в файл переменных среды сервиса - `/etc/default/postgrespro-std-13`

3. Настройка профиля сервисной учетной записи

Можно воспользоваться поставляевой PostgresPro утилитой `pg-wrapper`:

```
/opt/pgpro/std-13/bin/pg-wrapper links update
```

Но мы рекомендуем создать файл профиля для сервисной учетной записи `postgres`:

```
sudo install -o postgres -g postgres -m 600 \
             /etc/skel/.profile /var/lib/postgresql/
echo -e '\nPATH=/opt/pgpro/std-13/bin:$PATH\nexport PATH\n' \
      | sudo /usr/bin/tee -a /var/lib/postgresql/.profile
```

4. Инициализация PostgresPro:

```
sudo -iu postgres initdb
```

Если предполагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

5. Запуск и настройка автозапуска PostgresPro сервиса:

Для оригинальной версии PostgreSQL необходимо выполнить следующие действия:

```
sudo systemctl enable --now postgrespro-std-13.service
```

После завершения установки, не забудьте выполнить проверку в соответсвии с Проверка после установки

6.4. Установка Jatoba

СУБД Jatoba основана на базе PostgreSQL, при установке требует получения лицензионного ключа. Запуск БД

6.4.1. Astra Linux Special Edition 1.7

Необходимо подключение публичного репозитория:

- Jatoba 4 for Astra Linus SE Доступ к репозиторию предоставляется Производителем, поэтому в описании ниже будет использован не существующий URL для репозитория. Для установки пакетов Jatoba 4 необходим GPG-ключ(доступен в репозитории) - DEB-GPG-KEY-Jatoba

1. Подключение репозитория Jatoba4:

```
echo "deb https://repo.jatoba.ru/.../jatoba/astra-1_7-jatoba-4/ default all" \
     | sudo /usr/bin/tee -a  /etc/apt/sources.list.d/jatoba-4.list
sudo apt-key add DEB-GPG-KEY-Jatoba
sudo apt update
```

2. Установка пакетов сервера `Jatoba4`:

```
sudo apt -y install jatoba4-client \
                    jatoba4-contrib \
                    jatoba4-libs \
                    jatoba4-server
```

3. Настройка профиля сервисной учетной записи

Мы рекомендуем создать файл профиля для сервисной учетной записи `postgres`:

```
echo -e '\nPATH=/usr/jatoba-4/bin:$PATH\nexport PATH\n' \
      | sudo /usr/bin/tee -a /var/lib/jatoba/.bash_profile
```

4. Инициализация БД:

```
sudo -iu postgres initdb
```

Если предполлагается отличный от стандартного путь до файлов БД, укажите его для параметра, например: `-D /data/pgdata`.

5. Получение лицензионного ключа

Процедура получения ключа активации предполагает online-режим:

```
root@astra-template:~# cd /usr/jatoba-4/bin/
root@astra-template:/usr/jatoba-4/bin# ./jactivator
Добро пожаловать в центр активации Jatoba
Введите лицензионный ключ
XXXXX-XXXXX-XXXXX-XXX
Введите email адрес администратора
vpupkin@company.tld
```

```
Выберите способ активации:
  Online-активация (введите 1)
  Offline-активация (введите 2)
> 1
Используется сервер лицензирования: https://license.gaz-is.ru
Выберите режим активации:
 Обычная активация (введите 1)
 Реактивация (введите 2)
> 1
Время для активации 20 минут
Введите ключ активации с почты администратора
XXX-111-222-XXX
Введите путь для сохранения файла лицензии
/usr/jatoba-4/bin/-------------------------------------------------

Лицензия выпущена, файл лицензии успешно сохранен
Файл: /usr/jatoba-4/bin//jatoba.cer---------------------------------------------
  ----
```

После получения ключа необходимо поменять права доступа на сертификат:

```
chown postgres.postgres /usr/jatoba-4/bin/jatoba.cer
```

И добавить параметры для проверки лицензии в конфигурационный файл `${PGDATA}/postgresql.conf`:

```
lic_product_name = 'Jatoba'
lic_file_path = '/usr/jatoba-4/bin/jatoba.cer'
lic_server_addr = 'https://license.gaz-is.ru'
```

6. Запуск и настройка автозапуска Jatoba-4 сервиса:

```
sudo systemctl enable --now jatoba-4.service
```

6.5. Проверка после установки

Для корректной работы Luxms BI необходима не только работоспособная БД, но и доступность утилит БД для сервисной учетной записи. Поэтому после установки БД желательно провести следующие проверки.

1. Проверка работоспособности БД

```
netstat -nlpt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      951/sshd: /usr/sbin
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN
```

```
  2660/postgres
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      ↩
  1066/master
tcp6       0      0 :::22                   :::*                    LISTEN      ↩
  951/sshd: /usr/sbin
tcp6       0      0 ::1:5432                :::*                    LISTEN      ↩
  2660/postgres
tcp6       0      0 ::1:25                  :::*                    LISTEN      ↩
  1066/master
```

Вы должны увидеть процесс, который слушает на порту 5432. На этом этапе не важно, на каком IP-адресе слушает сервис.

2. Переключиться в shell сервисной учетной записи и проверить доступность необходимых комманд:

```
sudo -iu postgres
[sudo] password for admin:

[postgres@demo-bi ~]$ pg_dump -V
pg_dump (PostgreSQL) 13.13
[postgres@demo-bi ~]$ pg_ctl -V
pg_ctl (PostgreSQL) 13.13
[postgres@demo-bi ~]$ psql
psql (13.13)
Type "help" for help.

postgres=# exit
[postgres@demo-bi ~]$
```

7. Установка компонентов Luxms BI

Для установки компонентов Luxms BI необходимо подключение дополнительных репозиториев, в зависимости от ОС узла и компонента, устанавливаемого на хост. Ниже для каждого компонента Luxms BI будет дополнительно указано:

- какие дполнительные репозитории должны быть подключены
- какие дополнительные пакеты нужно установить

7.1. Развертывание БД Luxms BI

При установке пакета производится проверка существующих БД, при обнаружении существующей БД Luxms BI изменения данных БД не происходит.

Компоненты Luxms BI используют парольную аутентификацию для подключения к БД. Это защищает данные приложения от несанкционированного доступа. Единственным исключением является настройка **peer** аутентификации для суперпользователя БД - **postgres**. Это позволяет обеспечить установку обновлений и работу внутренней бизнес-логики БД.

При установке доступ к базе данных Luxms BI (**mi**) настроен для пользователя **bi** с паролем по умолчанию. **Обязательно измените его после установки.**

База данных Luxms BI может устанавливаться в автоматизированном режиме либо в ручном. Минимум условий для автоматизированной установки - запущенный экземпляр БД, до установки пакета с метаданными. Установка в “ручном” режиме - развертывание/перезапись метаданных после установки пакета **luxms-pg/luxmsbi-pgpro** и настройка ограничений на подключение к БД - `pg_hba.conf`.

7.1.1. Автоматизированная установка БД LuxmsBI

Для установки БД Luxms BI требуется установка пакета ***luxmsbi-pg/luxmsbi-pgpro***:

1. Установка пакета метаданных при работающем экземпляре БД выполнит донастройку конфигурационных файлов и создаст БД для системы Luxms BI. Для установки пакета необходимо выполнить команду:

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-pg
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-pgpro
```

Автоматическая установка БД из пакета включает в себя:

- корректировку конфигурации подключения к БД - вносит изменения в конфигурационный файл `pg_hba.conf` с сохранением оригинальной конфигурации;
- развертывание инициирующего дампа конфигурации системы Luxms BI.

Но только при условии использования базой данных стандартного порта и стандартного расположения файлов БД. При нестандартных параметрах необходимо до установки настроить переменные окружения в провиле сервисной учетной записи `postgres`, минимально достаточно добавить переменную `PGPORT`.

7.1.2. Ручная установка базы

Ручная установка базы данных Luxms BI необходима в ситуации, когда при установке пакета БД была не доступна или не определена переменная окружения `PGDATA`, указывающая на нестандартное расположение файлов БД.

1. Выполнить в ручном режиме, используя поставляемый с пакетом luxmsbi-pg скрипт:

```
su - postgres -c '/usr/share/luxmsbi-pg/luxmsbi-setupdb.sh /usr/share/luxmsbi-↩
  pg/luxmsbi-dump.sql.gz'
```

2. Корректировка ограничений подключения к БД выполняется при установке RPM-пакета. При ручной установке дампа БД необходимо проверить настройки файла pg_hba.conf (при необходимости изменить):

```
# TYPE  DATABASE     USER          ADDRESS          METHOD

local   all          postgres                      peer
local   bi           all                            md5
host    all          all           127.0.0.1/32     md5
host    all          all           ::1/128          md5
local   replication  all                            trust
host    replication  all           127.0.0.1/32     trust
host    replication  all           ::1/128          trust
```

При необходимости обеспечения доступа к БД с других хостов, при разнесении компонентов системы между разными узлами, необходимо добавить разрешения для подключения в соответсвии с документацией PostgreSQL сервера:

```
# Allow external connection
host    mi              bi              0.0.0.0/0               md5
```

7.2. Установка KeyDB сервера

Взаимодействия между компонентами Luxms BI построено с использованием быстрой in-memory database KeyDB. С помощью KeyDB реализован дополнительный функционал по ограничениям безопасности и контроля за пользовательскими сессиями. Установка KeyDB производится в большинстве случаев совместно с компонентом `luxmsbi-web`.

1. Установка:

Для RPM-based ОС:

```
sudo dnf -y install keydb
```

Для DEB-based ОС:

```
sudo apt -y install keydb-server
```

2. При установке компонентов Luxms BI на нескольких узлах необходимо обеспечить сетевую доступность:

Для RPM-based ОС:

```
sudo firewall-cmd --add-service=redis
sudo firewall-cmd --runtime-to-permanent
```

Для DEB-based ОС:

```
sudo ufw allow 6379/tcp comment 'KeyDB server'
```

3. Необходима настройка автоматического запуска сервера.

Для RPM-based ОС:

```
sudo systemctl enable keydb --now
```

Для DEB-based ОС:

```
sudo systemctl enable keydb-server --now
```

7.2.1. Обеспечение доступа по сети

При разнесении на разные хосты БД и KeyDB-сервера, необходимо настроить доступность API и БД для организации доступа к сервису.

Для настройки доступности сервиса с других хостов нужно изменить параметры стандартной конфигурации в файле `/etc/keydb/keydb.conf`. Закомментируйте и измените параметры:

```
# bind 127.0.0.1 ::1
protected-mode no
```

Необходим перезапуск сервиса. Для RPM-based ОС:

```
sudo systemctl restart keydb
```

Для DEB-based ОС:

```
sudo systemctl restart keydb-server
```

7.2.2. Резервирование(кластеризация) KeyDB

При развертывании нескольких экземпляров KeyDB необходимо настроить репликацию данных. В конфигурационном файле `/etc/keydb/keydb.conf` необходимо установить следующие параметры:

```
active-replica yes
replicaof [IP-адрес соседа] 6379
replica-readonly no
```

Необходим перезапуск сервиса. Для RPM-based ОС:

```
sudo systemctl restart keydb
```

Для DEB-based ОС:

```
sudo systemctl restart keydb-server
```

7.3. Развертывание Web приложения

Web-приложение Luxms BI, `luxmsbi-web`, базируется на HTTP сервере NGinx и требует взаимодействия с:

- KeyDB сервером;
- Java-приложением `luxmsbi-appserver`;
- Java-приложением `luxmsbi-datagate`;
- Java-приложением `luxmsbi-importer`;
- БД Luxms BI.

Устанавливаемые компоненты не вносят изменения в стандартную конфигурацию NGinx, а создают отдельную папку с конфигурацией **/opt/luxmsbi/conf/nginx** и собственный systemd сервис для запуска - `luxmsbi-web.service`.

Установка пакета Nginx производится как зависимость от пакета `luxmsbi-web` и, в большинстве случаев, использует самую последнюю версию, доступную в пакетных репозиториях ОС.

Исключение! Установка пакета на ОС CentOS7 или EL8(Rocky Linux 8, Oracle Linux 8 ...).

Поддержка ОС CentOS7 завершается 30 июня 2024 года. Настоятельно рекомендуется миграция на другую ОС.

Поставляемый в EL8 дистрибутивах пакет `NGinx` собран с устаревшей версией `OpenSSL`. Поэтому мы рекомендуем использовать пакет из официального репозитория NGinx с версией **1.25.4**. Или получить копию пакета из нашего репозитория

До установки пакета на ОС, базирующихся на EL8, необходимо ОТключить виртуальные репозитории:

```
sudo dnf module disable nginx php
```

До установки пакета на ОС, базирующихся на EL9, необходимо ПОДключить виртуальные репозиторий:

```
sudo dnf module enable nginx:1.22
```

Для установки необходимо выполнить следующую последовательность действий:

1. Установить пакет `luxmsbi-web`:

При установке пакета на RPM-based ОС автоматически производится настройка политики Selinux. Можно отключить SELinux, но рекомендуем не отказываться от дополнительной защиты ОС.

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-web
```

Для DEB-based ОС:

Версии пакета **luxmsbi-web** < **9.4.0** требуют дополнительной настройки. До установки пакета на Astra Linux SE 1.7 необходимо установить приоритет для пакета NGinx. Необходимо создать файл `/etc/apt/preferences.d/nginx` со следующим содержимым:

```
Package: nginx
Pin: version 1.21.4*
Pin-Priority: 1001
```

Начиная с версии пакета **luxmsbi-web** >= **9.4.0**, никаких дополнительных настроек не требуется. И при выполнении обновления на версию **9.4.0** требуется удалить ограничение установленное в файле `/etc/apt/preferences.d/nginx`.

И только после этого выполнять установку:

```
sudo apt -y install luxmsbi-web
```

2. Откорректировать конфигурационные файлы приложений:

Для LUA-скриптов, используемых в NGinx, необходимо настроить данные учетной записи для подключения к БД и KeyDB-сервера, конфигурационный файл `/opt/luxmsbi/conf/nginx/lua/bicfg.lua`:

```
return {
  dbhost="127.0.0.1",
  dbport=5432,
  dbname="mi",
  dbpool="pg-mi-4.0",
  dbuser="bi",
  dbpass="bi",
  dbcompact=false,
  dbpool_size=10, -- FIXME: NOT YET propogated default is 30  https://github.com/openresty/lua-nginx-module#lua_socket_pool_size
  kdbhost = "127.0.0.1",
  kdbport = 6379,
  debug = false,
  rpmlimit = 0, -- no limit
  max_login_attempts = 0, -- no checks
  login_attempts_ttl = 60, --seconds to make a continious session
  extra_headers = false, --add debug headers about pool usage, version and execution time.
}
```

4. Еще один ключевой файл конфигурации Web-приложения `/opt/luxmsbi/conf/nginx/conf.d/upstreams.conf`. Играет важную роль в интеграции Web-приложения и бизнес-логики БД Luxms BI/

```
# Define upstream_cookie for hash upstreams
map $http_cookie $upstream_cookie {
    default "";
    "~*LuxmsBI-User-Session=(.*?)($|;.*)" "$1";
}

# Defined upstreams
upstream binserver {
        server      127.0.0.1:8888;
}
upstream appserver {
        hash        $upstream_cookie consistent;
        server      127.0.0.1:8080;
}
upstream datagate {
        hash        $upstream_cookie consistent;
        server      127.0.0.1:8080;
}
upstream importer {
        hash        $upstream_cookie consistent;
        server      127.0.0.1:8192;
}
```

```
upstream gateway {
        hash        $upstream_cookie consistent;
        server      127.0.0.1:8889;
}
upstream lua-webapi {
        server      127.0.0.1:8282;
        keepalive   8;
}
```

Выше показана конфигурация файла для установки Luxms BI на едином хосте, устанавливаемая по-умолчанию. В случае более сложной инфраструктуры, с разнесением сервисов по выделенным хостам, этот конфигурационный файл должен быть откорректирован.

1. При использовании DCS Consul, откорректированная конфигурация должна сожержать DNS-имена зарегистрированных сервисов, например:

> При использовании параметра `resolve` необходимо убедиться в наличии определения для директивы `resolver 127.0.0.1;` в секции `http` конфигурации nginx `/opt/luxmsbi/conf/nginx/nginx.conf`

```
# Define upstream_cookie for hash upstreams
map $http_cookie $upstream_cookie {
    default "";
    "~*LuxmsBI-User-Session=(.*?)($|;.*)" "$1";
}

# Defined upstreams
upstream binserver {
        server      127.0.0.1:8888;
}
upstream appserver {
        hash        $upstream_cookie consistent;
    server      luxmsbi-appserver.service.consul:8080 resolve;
}
upstream datagate {
        hash        $upstream_cookie consistent;
        server      luxmsbi-datagate.service.consul:8080 resolve;
}
upstream importer {
        hash        $upstream_cookie consistent;
        server      luxmsbi-importer.service.consul:8192 resolve;
}
upstream gateway {
        hash        $upstream_cookie consistent;
        server      luxmsbi-gateway.service.consul:8889 resolve;
}
upstream lua-webapi {
        server      127.0.0.1:8282;
        keepalive   8;
}
```

2. При использовании нескольких инстансов компонентов, без регистрации сервисов в Consul, необходимо прописать IP-адреса сервисов, например:

За более подробной инструкцией по настройке обратитесь к документации по модулю NGinx ngx_http_upstream_module

```
# Defined upstreams
upstream binserver {
    server      127.0.0.1:8888;
}
upstream appserver {
    hash        $upstream_cookie consistent;
    server      127.0.0.1:8888;
}
upstream datagate {
    hash        $upstream_cookie consistent;
    server      10.0.0.5:8080;
    server      10.0.0.6:8080;
}
upstream importer {
    hash        $upstream_cookie consistent;
    server      10.0.0.5:8080;
    server      10.0.0.6:8080;
}
upstream gateway {
    server      127.0.0.1:8889;
}
upstream lua-webapi {
    server      127.0.0.1:8282;
    keepalive   8;
}
```

3. Выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable --now luxmsbi-web
```

7.3.1. Настройка безопасности Web-приложения

Конфигурационный файл `/opt/luxmsbi/conf/nginx/lua/bicfg.lua` включает в себя параметры подключения к базе Метаданных Luxms BI и настройки безопасности, связанные с защитой Web-приложения.

Для ограничения количества попыток подключения, защиты ПО от подбора пароля, необходимо установить параметры:

- **max_login_attempts** - количество неудачных попыток подключения за период, определяемый параметром login_attempts_ttl
- **login_attempts_ttl** - период времени, за который суммируются неудачные попытки подключения

По умолчанию ограничение количества неудачных входов отключено.

> Для того, чтобы работало ограничение неудачных попыток входа в Luxms BI, необходимо настроить подключение к KeyDB.

Для ограничения количества API-запросов с одного и того же клиента/IP-адреса необходимо настроить параметр:

- **rpmlimit** - определяет максимальное количество API-запросов с одного клиента в течение минуты

По умолчанию ограничение количества API-запросов отключено.

> Для того, чтобы работало ограничение количества API-запросов, необходимо настроить подключение к KeyDB.

Если для Вашей инсталляции необходима настройка SSO-авторизации, ознакомьтесь с разделом Приложение #4

> Для высоконагруженных инсталяций Luxms BI мы НЕ РЕКОМЕНДУЕМ настройку HTTPS на Web-серверах Luxms BI.
> Рекомендуем для этого функционала использовать аппаратные балансировщики нагрузки или выделенные сервера балансировки.

В случае необходимости настроить доступ к приложению по HTTPS и невозможности использования/отсутствия систем балансировки нагрузки, ознакомьтесь с Приложением #5

7.4. Развертывание BINS

Компонент `luxmsbi-bins` требует взаимодействия с:

- KeyDB сервером.
- БД Luxms BI.

Установка `luxmsbi-bins` производится совместно с компонентом `luxmsbi-web` и требует установки NodeJS версии 16, который устанавливается как зависимость из репозитория Luxms.

1. Установка BINS:

Для RPM-based ОС:

До установки пакета на ОС, базирующихся на EL8, необходимо подключить виртуальный репозиторий:

```
sudo dnf -y module enable nodejs:16
```

и установить пакет:

```
sudo dnf -y install luxmsbi-bins
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-bins
```

2. Настройки источников событий, используемых для трансляции в браузер пользователя, производятся в конфигурационном файле `/opt/luxmsbi/conf/bins.json`. В зависимости от схемы инсталяции, может потребоваться корректировка строки подключения:

```
{
  // Get changes from database
  "db": "postgres://bi:bi@127.0.0.1/mi",
  // Subscribe to changes channel on Redis-server
  "db.rt": "redis://127.0.0.1"
}
```

3. Выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable luxmsbi-bins --now
```

7.5. Установка NATS-Server

Компонент `NATS-Server` предоставляет распределенное объектное хранилище для компонентов Luxms BI.

1. При установке одно-узлового Luxms BI необходимо выполнить команды::

Для RPM-based ОС:

```
sudo dnf -y install nats-server
```

Для DEB-based ОС:

```
sudo apt -y install nats-server
```

2. При установке создается конфигурационный файл по умолчанию `/opt/nats/nats-server.conf`:

```
# Example config file

server_name: change_me-please
port: 4222
monitor_port: 8222
```

```
accounts: {
    SYS: {
        users: [
            { user: "x", pass: "y" }
        ]
    },
}

system_account: SYS

#cluster {
#  name: nats-cluster
#  port: 6222
  # Routes are actively solicited and connected to from this server.
#  routes = ["nats://nats-seed:6222","nats://nats-server-1:6223","nats://nats-server-2:6224"]
#}

max_payload: 16MB

jetstream: enabled

jetstream {
    store_dir: /opt/nats
    max_mem: 1G
}
```

Конфигурационный файл требует изменений:

- server_name - уникальный идентификатор, нужно установить значение. Например - имя хоста;
- accounts.SYS.users - аутентификационные данные для управления системными параметрами. Необходимо установить безопасные значения для атрибутов `user` и `password`.

При развертывании продуктовых, много-хостовых решений необходимо провести планирование инфраструктуры и выполнить развертывание кластерного решения в соответсвии с Приложением G

3. И выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable nats-server --now
```

7.6. Установка Java Runtime 11

Для работы Java-приложений требуется установка JRE/JDK. Можно использовать как поставляемый ОС так и Российские сертифицированные продукты.

1. Со всеми ОС, совместимыми с нашим ПО, поставляется OpenJDK 11:

- не требует приобретения отдельной лицензии;
- не требует отдельной установки, автоматически устанавливается как зависимость;
- не требует настройка конфигурации Java-компонентов Luxms BI.

2. ГосJava - Российское решение от компании “ООО Лаборотория 50”:

- доступны пакеты бесплатной версии для Astra Linux SE 1.7;
- предоставляется платная поддержка;
- требует отдельной установки;
- после установки требуется настройка конфигурации Java-компонентов Luxms BI.

3. Axiom JDK Pro/Certified - Российская компания ООО «БЕЛЛСОФТ»

- требует приобретения лицензии;
- предоставляется платная поддержка;
- требует отдельной установки;
- после установки требуется настройка конфигурации Java-компонентов Luxms BI.

После установки JRE/JDK, отличного от поставляемых с ОС, необходимо исправить конфигурационные файлы Java-компонентов, указать корректное значения для JAVA_HOME.

Для DEB-based ОС, в конфигурационных файлах:

- /etc/default/luxmsbi-appserver
- /etc/default/luxmsbi-datagate
- /etc/default/luxmsbi-importer

Для RPM-based ОС, в конфигурационных файлах:

- /etc/sysconfig/luxmsbi-appserver
- /etc/sysconfig/luxmsbi-datagate
- /etc/sysconfig/luxmsbi-importer

7.7. Установка Java Runtime 17 (план 1 квартал 2024 года)

В связи с возрастающим количеством уязвимостей безопасности и окончанием поддержки JDK 11:

- Oracle JDK 11, бесплатная поддержка закончилась в сентябре 2023 года;
- OpenJDK 11 от RHEL, бесплатная поддержка заканчивается в октябре 2024 года;
- по Российским ОС сведениями о сроке поддержки не обладаем.

Мы планируем в ближайшее время переход на 17 версию. К сожалению, JRE/JDK 17 на текущий момент поставляются только в ОС EL 8 и 9. Для отечественных ОС существуют несколько других вариантов.

1. Мы предоставляем свою сборку OpenJDK 17 JRE, доступные в наших репозиториях

- не требует отдельной установки, автоматически устанавливается как зависимость;
- не требует настройка конфигурации Java-компонентов Luxms BI. > > Наша сборка включает в себя ограниченный состав модулей, достаточный для работы приложения. > Но не позволяющий проводить отладку и разработку. > Наша сборка выполнена на основе Oracle JDK 17 с использованием стандартной утилиты jlink. {.is-info}

2. Oracle JDK 17

- требует отдельной установки;
- после установки требуется настройка конфигурации Java-компонентов Luxms BI.

3. Axiom JDK Pro/Certified - Российская компания ООО «БЕЛЛСОФТ»

- требует приобретения лицензии;
- предоставляется платная поддержка;
- требует отдельной установки;
- после установки требуется настройка конфигурации Java-компонентов Luxms BI.

7.8. Установка Luxms BI Appserver

Компонент `luxmsbi-appserver` поставляется в двух-вариантах, пакеты:

- **luxmsbi-appserver-mono**, предпочтительный, консолидированный функционал;
- **luxmsbi-appserver**, без функционала интеграции с Источниками данных.

Если Вы не знаете какой выбрать, выбирайте консолидированный пакет **luxmsbi-appserver-mono**. И у Вас отпадет необходимость устанавливать дополнительные Java-компоненты.

Требует взаимодействия с:

- NATS сервером;
- БД Luxms BI.

1. Установка `luxmsbi-appserver` производится в большинстве случаев совместно с компонентом `luxmsbi-web`:

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-appserver
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-appserver
```

2. Настройка используемого Java

В файле конфигурации переменных окружения для systemd unit-а необходимо указать или раскомментировать определение переменной **JAVA_HOME**:

```
# Systemd enviroment variables

# Uncomment to Java11-OpenJDK
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

PATH=/usr/bin:$JAVA_HOME/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin

# Options for JVM
# Example, to increase max HEAP size
# JAVA_TOOL_OPTIONS="-Xmx32g"
JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true"

# Options for ExecStart
OPTIONS="--spring.config.additional-location=↩
  /opt/luxmsbi/conf/appserver/application.properties"
```

3. Параметры приложения настраиваются в конфигурационном файле `/opt/luxmsbi/conf/appserver/application.properties` и имеют комментарии, кратко описывающие их назначение.

Необходимо настроить подключение к БД и серверу NATS в конфигурационном файле:

```
# LuxmsBI database properties
luxmsbi.datasource.url=jdbc:postgresql://127.0.0.1:5432/mi
luxmsbi.datasource.username=bi
luxmsbi.datasource.password=bi
luxmsbi.nats.servers=nats://localhost:4222
```

4. Выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable luxmsbi-appserver --now
```

5. Для работы функционала генерации отчетов в PDF-формате из табличных(XLSX) отчетов, дополнительно, необходима установка LibreOffice Calc:

Для RPM-based ОС:

```
sudo dnf -y install libreoffice-calc
```

Для DEB-based ОС:

```
sudo apt -y install libreoffice-calc
```

6. При установке консолидированного `luxmsbi-appserver` из пакета **luxmsbi-appserver-mono** необходимо внести корректировку в конфигурацию системы, через Web-приложение:

- парметер `datagate.url.prefix`

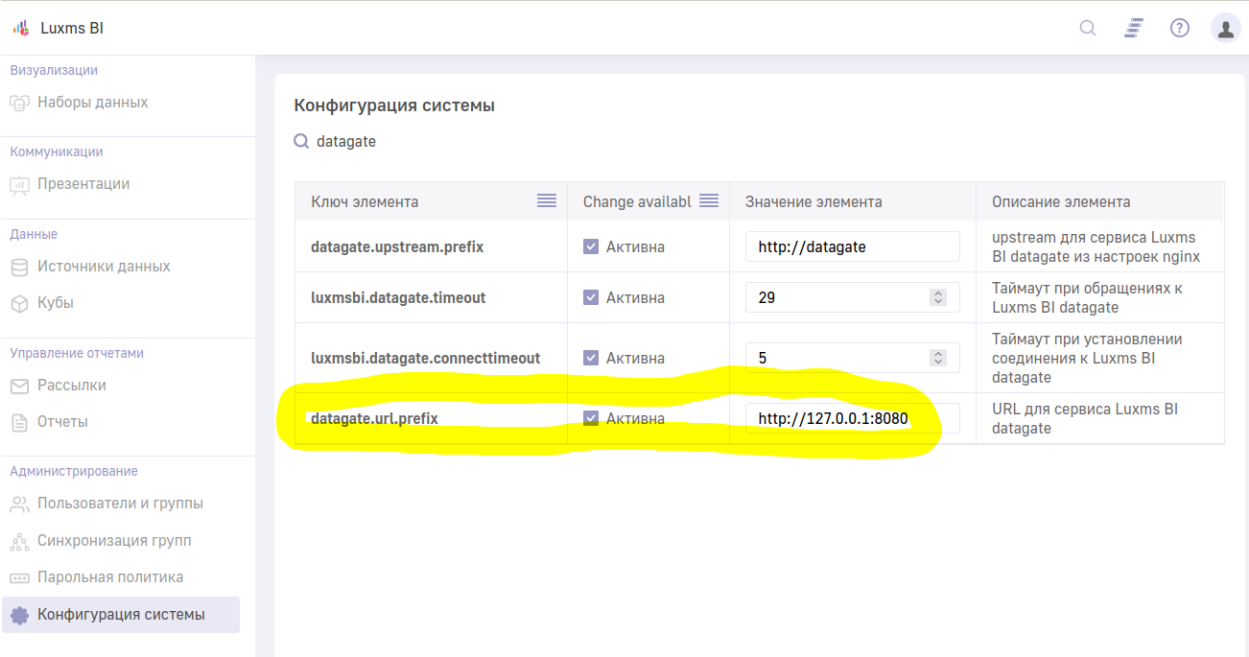


Рис. 7.1. **datagate-url-prefix.png**

7.9. Установка Luxms BI Datagate

При установленном пакете **luxmsbi-appserver-mono** устанвливать компонент luxmsbi-datagate не требуется! Пакет luxmsbi-datagate требуется устанавливать только при установленном пакете **luxmsbi-appserver**

В версиях Luxms BI начиная с 9 рекомендуется использование пакета **luxmsbi-appserver-mono**, который содержит в себе два сервиса сразу: Appserver и Datagate.

Компонент `luxmsbi-datagate` используется как выделенный сервис для взаимодействия с Источниками данных с использованием JDBC.

Требует взаимодействия с:

- NATS сервером;
- Компонентом `luxmsbi-appserver`;
- БД Luxms BI.

1. Установка приложения:

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-datagate
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-datagate
```

2. Настройка используемого Java

В файле конфигурации переменных окружения для systemd unit-а необходимо указать или раскомментировать определение переменной **JAVA_HOME**:

```
# Systemd enviroment variables

# Uncomment to Java11-OpenJDK
JAVA_HOME=/usr/lib/jvm/java-13-openjdk-amd64

PATH=/usr/bin:$JAVA_HOME/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin

# Options for JVM
# Example, to increase max HEAP size
# JAVA_TOOL_OPTIONS="-Xmx32g"
JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true"

# Options for ExecStart
OPTIONS="--spring.config.additional-location=↩
  /opt/luxmsbi/conf/appserver/application.properties"
```

3. Параметры приложения настраиваются в конфигурационном файле `/opt/luxmsbi/conf/datagate/application.properties` и имеют комментарии, кратко описывающие их назначение.

Необходимо настроить подключение к БД и серверу NATS в конфигурационном файле:

```
luxmsbi.datasource.url=jdbc:postgresql://127.0.0.1:5432/mi
luxmsbi.datasource.username=bi
luxmsbi.datasource.password=bi
luxmsbi.nats.servers=nats://localhost:4222
```

4. При развертывании компонента на выделенном узле необходимо обеспечить сетевую доступность:

Для RPM-based ОС:

```
sudo firewall-cmd --add-service=luxmsbi-datagate
sudo firewall-cmd --runtime-to-permanent
```

Для DEB-based ОС:

```
sudo ufw allow Luxmsbi-Datagate
```

5. Выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable luxmsbi-datagate --now
```

7.10. Драйвера JDBC для доступа к данным

На текущий момент Luxms BI поставляется со следующим набором JDBC для доступа к `Источникам данных`:

- PostgreSQL, v42.5.1
- Clickhouse, v0.4.6
- SQLserver, v7.0.0
- MSAccess, v5.0.1
- Oracle, v19.16.0.0
- Mysql, v8.0.33
- MariaDB, v3.0.4
- Linter, v1.8
- SQLite, v3.20.1
- Olap4j, v1.1.0
- Teradata, v16.20.0.13

Список драйверов может быть расширен и версии драйверов могут быть обновлены в дальнейшем. Для уточнения используемых в Вашей инсталяции драйверов необходима проверка установленных драйверов. Ниже описана настройка и месторасположение в файловой системе, которое подлежит проверке.

7.10.1. Подключение дополнительных драйверов

Драйвера JDBC используются компонентами `luxmsbi-importer` и `luxmsbi-datagate`. Расположение драйверов JDBC в папках файловой системы, определяется параметром конфигурации компонента(**/opt/luxmsbi/config/[Имя Компонента]/application.properties**):

```
luxmsbi.drivers-config.location=/opt/luxmsbi/lib/jdbc
```

Требования к настройке драйверов JDBC:

1) Имя папок должно соответсвовать следующим правилам: **[Vendor name]_[Major version]_[Minor version]**,

где `Vendor name` - должно совпадать с именем используемым в url, для подключения к источнику данных '**jdbc:<vendor>:**//'

Например: `/opt/luxmsbi/lib/jdbc/mysql_5_1/`

2) Каждая папка драйвера должна содержать:

- набор JAR-файлов драйвера;
- файл описания `info.json`, в формате JSON.

3) Формат файла описания `info.json`, например для MariaDB:

```
{"vendor":"mysql","majorVersion":5,"minorVersion":1,"className":↩
  "com.mysql.jdbc.Driver","config":{}}
```

Элемент “config” может содержать перечень свойств, поддерживаемых конкретным драйвером. Полный перечень параметров корректнее искать в документации по драйверу.

4) Владелец и права на папку драйвера, содержащиеся в ней файлы, должны быть следующие:

```
[root@host ~]# ls -la /opt/luxmsbi/lib/jdbc/mysql_5_1/
total 1004
drwxrwx---.  2 bi   bi        62 Jun  2 04:00 .
drwxrwx---. 11 bi   bi     12288 Jun  1 11:05 ..-
rw-r-----.  1 bi   bi       100 Jun  1 11:04 info.json-
rw-r-----.  1 bi   bi   1007502 Jun  1 11:04 mysql-connector-java-5.1.47.jar
```

Добавление драйверов требует перезапуска компонентов:

```
sudo systemctl restart luxmsbi-datagate luxmsbi-importer
```

7.11. Установка Luxms BI Importer

Luxms BI Importer - поддерживаемый, но устаревший компонент интеграции с внешними Источниками данных. Начиная с версии Luxms BI 9 не требуется, но может быть установлен для поддержки старых проектов.

Компонент `luxmsbi-importer` требует взаимодействия с:

- NATS сервером;
- БД Luxms BI;
- Компонентом `luxmsbi-appserver`;
- Компонентом `luxmsbi-datagate`.

1. Установка приложения:

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-importer
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-importer
```

2. Настройка используемого Java

В файле конфигурации переменных окружения для systemd unit-а необходимо указать или раскомментировать определение переменной **JAVA_HOME**:

```
# Systemd enviroment variables

# Uncomment to Java11-OpenJDK
JAVA_HOME=/usr/lib/jvm/java-13-openjdk-amd64

PATH=/usr/bin:$JAVA_HOME/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin

# Options for JVM
# Example, to increase max HEAP size
# JAVA_TOOL_OPTIONS="-Xmx32g"
JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true"

# Options for ExecStart
OPTIONS="--spring.config.additional-location=↩
  /opt/luxmsbi/conf/appserver/application.properties"
```

3. Параметры приложения настраиваются в конфигурационном файле `/opt/luxmsbi/conf/importer/application.properties` и имеют комментарии, кратко описывающие их назначение.

Необходимо настроить подключение к БД и серверу NATS в конфигурационном файле:

```
luxmsbi.datasource.url=jdbc:postgresql://127.0.0.1:5432/mi
luxmsbi.datasource.username=bi
luxmsbi.datasource.password=bi
luxmsbi.nats.servers=nats://localhost:4222
```

4. При развертывании компонента на выделенном узле необходимо обеспечить сетевую доступность:

Для RPM-based ОС:

```
sudo firewall-cmd --add-service=luxmsbi-importer
sudo firewall-cmd --runtime-to-permanent
```

Для DEB-based ОС:

```
sudo ufw allow Luxmsbi-Importer
```

5. Выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable luxmsbi-importer --now
```

7.12. Установка Luxms Databoring

Компонент `luxms-databoring` требует взаимодействия с:

- Luxmsbi-Datagate компонентом;
- Luxmsbi-Importer компонентом.

1. Установка приложения:

Для RPM-based ОС:

```
sudo dnf -y install luxms-databoring
```

Для DEB-based ОС:

```
sudo apt -y install luxms-databoring
```

2. Параметры приложения настраиваются в конфигурационном файле сервиса:

- Для RPM-based -`/etc/sysconfig/luxms-databoring`,
- Для DEB-based ОС - `/etc/default/luxms-databoring`

Конфигурационные файлы имеют комментарии, кратко описывающие их назначение. В случае размещения компонентов `luxmsbi-datagate` и/или `luxmsbi-importer` и/или `luxmsbi-web` не на том же хосте, где располагается компонент `luxms-databoring` необходимо указать IP-адрес или DNS-имя для соответствующего компонента.

```
# RSocket endpoint of Datagate service. In format [IP|DNS]:[PORT].
# If empty set default to "127.0.0.1:7200"
DATAGATE_HOST=""

# RSocket endpoint of Importer service. In format [IP|DNS]:[PORT].
# If empty set default to "127.0.0.1:7192"
IMPORTER_HOST=""

# Luxms BI API HTTP Proxy In format [IP|DNS]:[PORT].
# If empty set defualt to "http://127.0.0.1/"
LUXMSBI_HTTP_API_URL=""

NODE_OPTIONS=""
```

3. При развертывании компонента на выделенном узле необходимо обеспечить сетевую доступность:

Для RPM-based ОС:

```
sudo firewall-cmd --add-port=1880/tcp
sudo firewall-cmd --runtime-to-permanent
```

Для DEB-based ОС:

```
sudo ufw allow 1880/tcp
```

4. Выполнить настройку автоматического запуска systemd сервиса:

```
sudo systemctl enable luxms-databoring --now
```

5. В состав пакета входит конфигурационный файл Web-сервера NGinx `/opt/luxmsbi/conf/nginx/conf.d/luxms-databoring.location`:

```
location /databoring/ {
    if ($arg_port != '') {
        return 301 $scheme://$host/databoring/$arg_port/;
    }

    proxy_http_version 1.1;
    proxy_send_timeout 300s;
    proxy_set_header   Upgrade $http_upgrade;
    proxy_set_header   Connection $connection_upgrade;
    proxy_set_header   Host $host;

    location ~ ^/databoring/(\d*)/(.*) {
        #access_log  /var/log/luxmsbi/nginx/boring.access.log;
        error_log   /var/log/luxmsbi/nginx/boring.error.log;
        proxy_pass  http://127.0.0.1:$1/$2$is_args$args;
    }

    proxy_pass http://127.0.0.1:1880/;
}
```

В случае размещения `luxms-databoring` на выделенном хосте необходимо перенести этот конфигурационный файл на хост с установленным компонентом `luxmsbi-web`.

7.13. Установка пакета с Документацией

Настоящая документация поставляется в виде пакета `luxmsbi-docs` для его установки необходимо выполнить команду:

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-docs
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-docs
```

Пакет поставляет дополнительный конфигурационный файл для Web-приложения Luxms BI. После первичной установки пакета, необходимо выполнить перезапуск сервиса^

```
systemctl restart luxmsbi-web
```

После выполнения этих действий Вы можете найти документацию по ссылке:

`http(s)://<DNS-имя/IP-адрес Вашего сервера>/docs/`

7.14. Проверка корректности установки и настройки

Для проверки корректности установки компонентов и их параметров рекомендуем следующий набор комманд:

1) Проверка работоспособности сервисов - позволит Вам увидеть проблемный сервис

```
systemctl list-units | grep '^nats\|luxms\|keydb\|postgres'
```

2) Проверка ошибки конкреного сервиса `journalctl -u <имя сервиса>`

```
journalctl -u luxmsbi-appserver.service

Aug 03 05:24:25 bi9-el8.ci-test.luxms.com systemd[1]: Started LuxmsBI AppServer.
Aug 03 05:24:25 bi9-el8.ci-test.luxms.com luxmsbi-appserver[19579]: Picked up ↩
  JAVA_TOOL_OPTIONS: -Djava.net.preferIPv4Stack=true
Aug 03 05:24:26 bi9-el8.ci-test.luxms.com luxmsbi-appserver[19579]: Exception ↩
  in thread "main" java.lang.Error: java.io.FileNotFoundException: ↩
  /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-2.el8.x86_64/lib/tzdb.dat (No such ↩
  file or directory)
```

7.15. Настройка параметров компонентов с учетом ресурсов

Настройка параметров компонентов зависит от ресурсов аппаратных или виртуальных хостов.

7.15.1. Настройка Java-Heap

При работе с большим объемом при загрузке данных, через ETL или с использованием импорта файла, есть необходимость в увеличении объема оперативной памяти для обработки данных. Этот выставляется в виде параметра JVM для пакетов **luxmsbi-appserver-mono** и **luxmsbi-datagate**, в зависисмости от используемого пакета.

В этом случае производиться настройка одного из двух конфигурационных файлов:

- `/etc/{sysconfig,default}/luxmsbi-appserver`
- `/etc/{sysconfig,default}/luxmsbi-datagate`

```
# Options for JVM
JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true -Xmx16G"
```

7.15.2. Настройка параметров соединения с БД-метаданных

База метаданных не самый нагруженый компонент Luxms BI, если она не используется для хранения данных визуализации. Но при большом количестве пользователей возникает необходимость увеличения количества соединений для БД.

Главное обеспечить количество входящих соединение в БД Метаданных не выше количества соединений, предоставляемых Базой данных. Учитывайте также соединения служб мониторинга.

В первую очередь производится тюнинг параметров БД исходя из ожидаемой нагрузки. Для продуктовой схемы с вышеуказанными параметрами предлагается установка следующих параметров:

1) Установите параметр max_connections в значение 400. Для изменения параметра необходимо использование команды:

```
patronictl -c /etc/patroni/patroni.yml edit-config
```

2) После установки параметра необходимо перезапустить узлы БД командой:

```
patronictl -c /etc/patroni/patroni.yml restart
```

Во вторую очередь производится тюнинг параметров Web-приложения luxmsbi-web:

1) В конфигурационном файле /opt/luxmsbi/conf/nginx/nginx.conf

```
worker_processes 8;
```

2) В конфигурационном файле /opt/luxmsbi/conf/nginx/lua/bicfg.lua

```
dbpool_size=20,
```

В третью очередь определяются размеры пула соединений к БД из Java-приложений. Например:

1) В конфигурационном файле /opt/luxmsbi/conf/appserver/application.properties

```
luxmsbi.datasource.max-pool-size=10
```

3) В конфигурационном файле /opt/luxmsbi/conf/datagate/application.properties

```
luxmsbi.datasource.max-pool-size=25
```

4) В конфигурационном файле /opt/luxmsbi/conf/importer/application.properties

```
luxmsbi.datasource.max-pool-size=5
```

8. Инструкция по настройке подключения к почтовому сервису

Есть две части, необходимые для отправки писем:

- Почтовый сервер, который настраивается на стороне компании-клиента.
- Настройка `/opt/luxmsbi/conf/appserver/application.properties`.

Рассылка может осуществляться либо с настроенного локального сервиса (например, `sendmail`) или через существующий ящик на действующем сервере по протоколу `smtp`.

В зависимости от выбранного варианта существуют разные настройки в `/opt/luxmsbi/conf/appserver/application.properties`.

Например, это настройка через корпоративный ящик через `smtp`:

```
# Mail sender used for Presentations and Reports

luxmsbi.mail-sender.enabled=true
# 'From' value for outgoing emails.
luxmsbi.mail-sender.sender-email=kt@waldorf-spb.ru
luxmsbi.mail-sender.retry-count=3
luxmsbi.mail-sender.retry-interval=10000

# Mail server connection
# @see https://docs.spring.io/spring-boot/docs/current/reference/html/application-properties.html#appendix.application-properties.mai
# https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html

spring.mail.host=smtp.mail.ru
spring.mail.port=465
#spring.mail.username=kt[redacted]
spring.mail.password= xxxxxxxxxxxx
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.ssl.enable=true
spring.mail.properties.mail.transport.protocol=smtp
spring.mail.properties.mail.smtp.ssl.trust=*
spring.mail.properties.mail.debug=true
```

Для тестирования можно изменить в последней строчке `*.debug=true` и запустить команду:

```
journalctl -u luxmsbi-appserver -f
```

Пример успешной отправки:

```
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: bHOtkc9qwzAMh1/F6L446WCMUbeXMei1fx5A2EocmtjG0trl7esy2FLoYIeehGX0fT+k5fprHNSJ
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: MvcxGGiqGhQFG10fOgOH/cfTKygWDA6HGMjARAzr1XJLA0oZYd8nVoUR2IAXSW9as/U0IlcxUSg/
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: bcwjSnnmTie0R+xIL+r6Rec5A26ZauMM5I1rQO0xdyQGWKaBuCrE0psS/ccX27a39B7t50hB7mj1
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: NxX0ff3iV3+O+cieSMrMtTSPTvIj+CvM82wXHjO5neRypsevZA6/htE3115dAFBLBwj2cWkV1wAA
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: ADQCAABQSwMEFAAICAgAQIm4VgAAAAAAAAAAAAAABgAAAB4bC93b3Jrc2hlZXRzL3NoZWV0MS54
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: bWy1WM1y2zYQvvspOLxH/LH+rJGUSey66YxTZyqnmemlA5GgiDEJsAAk2T721jfoKe+Q6aW99BmU
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: N+oC/BEsqw6soXQSFosP++0Hkosdv77LM2eFuSCMTtyg47sOphGLCV1M3I83l6+G7uvpyXjN+K1I
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: MZYO+FMxcVMpi5HnisjFORIdVmAKMwnjOZIw5AtPFByjWC/KMy/0/b6XI0LdEmGURzYgOeK3y+JV
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: xPICSTInGZH3GguG4TYoLElIhC9YtMwxlWUsHGeAyKhISSFqtLvYCi/maA0JqmkZTC/KmQbvKVxO
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: Is4ES2QHSFWRPclV0IX4VkSJUkOhAwIz8w3Ugq5dNPguwjoK/xlqyBatiqnEMyOi4uxl8ewwoiLw
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: Xwgw9MISYnriwG+siX3gTkIyifl7FuOJm6BM4MpBOxVogWdYfiyOo7xhH8BQ+3kVkldBVcOYwElT
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: 8jkcJxP3TTB6G/QaZ+37M8FrYWzTGB2RsvUlqLrMkKj30cbvOYmvCMVglXxZGX9i63OWvQOZ4Ck2
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: J37BkInawMkihdCvcCIbSInmM5zhSOLYXHe9lBlsMrvP5yxrAGKcoGUmVQiwHeO1fQURT1yqDmEG
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: kKxQW5zjLFOsXSdSvj8Afr/rOg+M5bMIZZC9wPeN8Y96+a5V5fkK3bOlTks1q95Ec8ZulUnh+oZU
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: ZR41JZX6AlHYqQrJdRBYV3gb2nZc4jjit0qsWihDWbXdI6kN+UrpLvVzCGekyhRk6ROJZTpxh53+
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: cNBrUgiCvcNKDmDUUy/cBxCqtlTSsFKDK7zCGfjr8EwbgCvbowN1gSQy4uZs7YBKQHS7U7QUkuX1
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: XqWAJuyedEYKRKVLaCxYJnZctNvK21nnRfuA3tZAL4UYe8DnKbuwFXahDsov2Um+N7hp2O34vU7o
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: h6dOMBgF/sj3x97KjnX4LPQBvE9b4X2qeT8f3BQQqPxV3hfYmq4VbsSWgFuiU5R/A/1/8tBtJQ9d
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: m3jhhcSldQpMSLofEIqtgzj3WuHcs9OIJmRhTbp3PNL9Vkj3rUhDySPtD3v/eKQHrZAe2JCO4av5
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: AtKD45EetkJ6aPVqi4n9Ez08HuWzViif2VBecETtOZ8dj3Pgt1OY+DasM7Yg1Jr1I8y2abdUjwWW
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: tAHNnndwRN7tVGpBaMNbYKHuXvbEwyMSb6dUC6xqKnUZs2d9+m3W4aGs2ynMAqvKbPN58+XrH5u/
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: N/84mz+//r75d/PXyD4LNpXaweK3U6sFvQNvTSbPQzEMatXVd3vRHOeYL/QVWji6nleJNJY203Ub
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: pGf2Qfa7gMe2q7LFrywFJlReF7pz56SMkwdGJcrO4RaBuepjVBfoFRTrJNozkWKkmpvbxspi21TZ
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: tcxwrVMdkOoFvUccvibCyXQvxe8Mqu5K9V+yQv+DfM+ZBMHrUapbNGrUC4JhEMBNsh+GfhfWJIzJ
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: /VPmxhDOsnAKVGA+Iw9Y1w3CaKfo5lTVcwiqYdNpcB0Fcc11CDFb05sU02tIE5xBTiBLuhs6cQvG
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: JUdEQvQZim7f0PhTSmTT73JijowWUgTSnLNctVWF6gJRleFGlYuCwFVVhVbLsbVErCBYH8iaYpmf
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: S50KJyZJArqB1nH8HdwRt/vXE5eEC7mnXcfiuGyGeY+NJXBzsszdpidjr+lwT/8DUEsHCOwgPsTb
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: BAAAFBcAAFBLAQIUABQACAgIAECJuFaoRjetYQEAAGYFAAATAAAAAAAAAAAAAAAAAAAAABbQ29u
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: dGVudF9UeXBlc10ueG1sUEsBAhQAFAAICAgAQIm4Vqf9iNTlAAAASgIAAAsAAAAAAAAAAAAAAAAA
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: ogEAAF9yZWxzLy5yZWxzUEsBAhQAFAAICAgAQIm4Vn0Ir+n8AAAAAgIAABAAAAAAAAAAAAAAAAAA
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: wAIAAGRvY1Byb3BzL2FwcC54bWxQSwECFAAUAAgICABAibhW3kmwRiQBAADtAQAAEQAAAAAAAAAA
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: AAAAAAD6AwAAZG9jUHJvcHMvY29yZS54bWxQSwECFAAUAAgICABAibhWP2RU04gBAABIBQAAFAAA
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: AAAAAAAAAAAAAABdBQAAeGwvc2hhcmVkU3RyaW5ncy54bWxQSwECFAAUAAgICABAibhWNs1QwlgD
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: AAAuGgAADQAAAAAAAAAAAAAAAAAnBwAAeGwvc3R5bGVzLnhtbFBLAQIUABQACAgIAECJuFZpVEAZ
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: oQIAADEGAAAPAAAAAAAAAAAAAAAALoKAAB4bC93b3JrYm9vay54bWxQSwECFAAUAAgICABAibhW
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: 9nFpFdcAAAA0AgAAGgAAAAAAAAAAAAAAACYDQAAeGwvX3JlbHMvd29ya2Jvb2sueG1sLnJlbHNQ
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: SwECFAAUAAgICABAibhW7CA+xNsEAAAUFwAAGAAAAAAAAAAAAAAAAAC3DgAAeGwvd29ya3NoZWV0
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: cy9zaGVldDEueG1sUEsFBgAAAAAJAAkAPwIAANgTAAAAAA==
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: ------=_Part_6_1810642774.1684937400285--
май 24 17:10:00 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: .
май 24 17:10:01 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: 250 OK id=1q1pBg-00Hah0-K9
май 24 17:10:01 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: DEBUG SMTP: message successfully delivered to mail server
май 24 17:10:01 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: QUIT
май 24 17:10:01 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: 221 smtp29.i.mail.ru closing connection
май 24 17:10:01 dev-oberezenko.spb.luxms.com luxmsbi-appserver[11731]: 2023-05-24 17:10:01.062 11731 [pool-10-thread-2] INFO  c.l.b.m.s.MailSenderServiceImpl :: Emai
```

8.1. Тестовая отправка при настройке локального почтового сервера

Пример команды в терминале, замените email на значение, настроенное у вас:

```
curl -v 'http://localhost:8080/ipc/service' \-
H 'Content-Type: application/json;charset=UTF-8' \-
d '{
   "service":"MailSenderRemoteService.sendMail",
   "args":[
      {
      "from":"user1@mail.com",
      "fromName":"Pablo",
      "subject":"Hello",
      "body":"World",
      "to":"user2@gmail.com",
      "cc":"everhax@gmail.com",
      "attachment":
         [
            {"type":"base64",
            "name":"smth.txt",
            "body":"SCBFIEwgTCBPCg=="}
         ]
      }
   ]
}'
```

9. Управление компонентами системы Luxms BI

9.1. Управление DCS Consul

Consul - универсальное и комплексное решение, предоставляющее функционал распределенного кластера для управления сервисами в ИТ-инфраструктуре. Данный документ не содержит исчерпывающей информации по методам управления этого решения. Вам в любом случае необходимо изучить документацию по данному ПО.

```
# consul --help
Usage: consul [--version] [--help] <command> [<args>]

Available commands are:
    acl             Interact with Consul's ACLs
    agent           Runs a Consul agent
    catalog         Interact with the catalog
    config          Interact with Consul's Centralized Configurations
    connect         Interact with Consul Connect
    debug           Records a debugging archive for operators
    event           Fire a new event
    exec            Executes a command on Consul nodes
    force-leave     Forces a member of the cluster to enter the "left" state
    info            Provides debugging information for operators.
    intention       Interact with Connect service intentions
    join            Tell Consul agent to join cluster
    keygen          Generates a new encryption key
    keyring         Manages gossip layer encryption keys
    kv              Interact with the key-value store
    leave           Gracefully leaves the Consul cluster and shuts down
    lock            Execute a command holding a lock
    login           Login to Consul using an auth method
    logout          Destroy a Consul token created with login
    maint           Controls node or service maintenance mode
    members         Lists the members of a Consul cluster
    monitor         Stream logs from a Consul agent
    operator        Provides cluster-level tools for Consul operators
    reload          Triggers the agent to reload configuration files
    rtt             Estimates network round trip time between nodes
    services        Interact with services
    snapshot        Saves, restores and inspects snapshots of Consul server state
    tls             Builtin helpers for creating CAs and certificates
    validate        Validate config files/directories
    version         Prints the Consul version
    watch           Watch for changes in Consul
```

Кроме управления из командной строки, Consul предоставляет функционал управления:

- Через Web-интерфейс.
- Через API-интерфейс.

9.2. Настройка параметров БД

При установке БД Luxms BI из пакета `luxmsbi-pg` устанавливаются параметры сервера БД, расчитанные для минимальных ресурсов:

```
--
 Generated by PGConfig 2.0 beta----
 http://pgconfig.org--

 Memory Configuration
ALTER SYSTEM SET shared_buffers TO '1GB';
ALTER SYSTEM SET effective_cache_size TO '3GB';
ALTER SYSTEM SET work_mem TO '20MB';
ALTER SYSTEM SET maintenance_work_mem TO '512MB';--

 Checkpoint Related Configuration
ALTER SYSTEM SET min_wal_size TO '2GB';
ALTER SYSTEM SET max_wal_size TO '6GB';
ALTER SYSTEM SET checkpoint_completion_target TO '0.9';
ALTER SYSTEM SET wal_buffers TO '16MB';--

 Network Related Configuration
ALTER SYSTEM SET listen_addresses TO '*';
ALTER SYSTEM SET max_connections TO '150';--

 Storage Configuration
ALTER SYSTEM SET random_page_cost TO '4.0';
ALTER SYSTEM SET effective_io_concurrency TO '2';--

 Worker Processes
ALTER SYSTEM SET max_worker_processes TO '2';
ALTER SYSTEM SET max_parallel_workers_per_gather TO '1';
ALTER SYSTEM SET max_parallel_workers TO '2';
```

После установки рекомендуем рассчитать параметры под ваши ресурсы. Наши рекомендации:

1) Используйте для изменения конфигурации сервера команды `ALTER SYSTEM`, это позволяет избежать ошибок при редактировании `postgresql.conf`. При этом конфигурационные параметры применяются при каждом рестарте экземпляров БД из конфигурационного файла `postgresql.auto.conf`.

2) Для генерации конфигурационных команд можно использовать любой калькулятор, мы рекомендуем PGConfig.

9.3. Управление кластером Patroni

Управление сервисом Patroni выполняется утилитой `systemctl`. Поддерживаются следующие команды:

- `start`.
- `reload`.
- `restart`.
- `stop`.

События, генерируемые Patroni, регистрируются Journald. Для получения журнальных записей вам необходимо выполнить команду:

```
journalctl -u patroni
```

Часто используемые опции утилиты `journactl` вы можете найти в этом документе

Для управления кластером БД пакет Patroni устанавливает утилиту `patronictl`. Перечень доступных команд, приведенных ниже, требует изучения документации от производителя ПО. Утилита предоставляет вывод перечня доступных команд:

```
patronictl -c /opt/patroni/etc/patroni.yml --help
Usage: patronictl [OPTIONS] COMMAND [ARGS]...

Options:
  -c, --config-file TEXT  Configuration file
  -d, --dcs TEXT          Use this DCS
  -k, --insecure          Allow connections to SSL sites without certs
  --help                  Show this message and exit.

Commands:
  configure    Create configuration file
  dsn          Generate a dsn for the provided member, defaults to a dsn of...
  edit-config  Edit cluster configuration
  failover     Failover to a replica
  flush        Discard scheduled events
  history      Show the history of failovers/switchovers
  list         List the Patroni members for a given Patroni
  pause        Disable auto failover
  query        Query a Patroni PostgreSQL member
  reinit       Reinitialize cluster member
  reload       Reload cluster member configuration
  remove       Remove cluster from DCS
  restart      Restart cluster member
  resume       Resume auto failover
  scaffold     Create a structure for the cluster in DCS
  show-config  Show cluster configuration
  switchover   Switchover to a replica
  topology     Prints ASCII topology for given cluster
  version      Output version of patronictl command or a running Patroni...
```

Мы приведем минимальный перечень команд, который необходим вам в начале эксплуатации этого решения, но в любом случае вам необходимо обратиться к первоисточнику для изучения возможностей данной утилиты.

1. Проверка статуса узлов кластера:

```
patronictl -c /opt/patroni/etc/patroni.yml list postgresdb
```

Cluster	Member	Host	Role	State	TL	Lag in MB
postgresdb	centos-1.local	10.0.2.5	Leader	running	25	
postgresdb	centos-2.local	10.0.2.4		running	25	0.0
postgresdb	centos-3.local	10.0.2.15		running	25	0.0

2. Передача роли `Leader` на другой хост:

```
[root@centos-3 ~]# patronictl -c /opt/patroni/etc/patroni.yml list postgresdb
```

Cluster	Member	Host	Role	State	TL	Lag in MB
postgresdb	centos-1.local	10.0.2.5	Leader	running	25	
postgresdb	centos-2.local	10.0.2.4		running	25	0.0
postgresdb	centos-3.local	10.0.2.15		running	25	0.0

```
[root@centos-3 ~]# patronictl -c /opt/patroni/etc/patroni.yml failover postgresdb
Candidate ['centos-2.local', 'centos-3.local'] []: centos-2.local
Current cluster topology
```

Cluster	Member	Host	Role	State	TL	Lag in MB
postgresdb	centos-1.local	10.0.2.5	Leader	running	25	
postgresdb	centos-2.local	10.0.2.4		running	25	0.0
postgresdb	centos-3.local	10.0.2.15		running	25	0.0

```
Are you sure you want to failover cluster postgresdb, demoting current master ↩
  centos-1.local? [y/N]: y
2020-02-07 18:21:10.02461 Successfully failed over to "centos-2.local"
```

Cluster	Member	Host	Role	State	TL	Lag in MB
postgresdb	centos-1.local	10.0.2.5		stopped		unknown
postgresdb	centos-2.local	10.0.2.4	Leader	running	25	
postgresdb	centos-3.local	10.0.2.15		running	25	0.0

9.4. Управление сервисами приложений

Управление компонентами приложения Luxms BI реализовано с использованием Systemd Units. Минимально поддерживаемый перечень команд:

```
systemctl enable <component-name>
systemctl start <component-name>
systemctl restart <component-name>
systemctl stop <component-name>
systemctl disable <component-name>
```

События и ошибки компонентов Luxms BI регистрируются в Journald. Рекомендации по просмотру этих событий описаны в следующем разделе.

Исключением являются журналы Web-сервера, использующего NGinx. Журналы Web-сервера находятся в файловой системе - `/var/log/luxmsbi`.

9.5. Рекомендации по просмотру журнальных файлов

Утилита **journalctl** использует **less** как средство просмотра вывода. Что позволяет выполнять контекстный поиск и фильтрацию по шаблону.

1. Используйте фильтрацию вывода журналов с помощью указания имени сервиса, параметр `-u`:

```
journalctl -u luxmsbi-importer
```

2. При необходимости получения вывода с переносом строк, можно воспользоваться двумя способами:

- Установите переменную окружения для пользователя или запускайте утилиту с измененным окружением, по умолчанию journalctl использует настройку `SYSTEMD_LESS=FRSXMK` :

```
SYSTEMD_LESS=FRXMK journalctl -u luxmsbi-importer
```

Изменения переменной среды позволит использовать поиск и фильтрацию **less**

- Используйте перенаправление вывода в файл или параметр `-no-pager`:

```
journalctl -u luxmsbi-importer --no-pager
journalctl -u luxmsbi-importer > dump.log
```

Недостаток этого метода - вывод полного содержимого журнального файла

3. Используйте параметры `-since` и `-until`. Параметры позволяют ограничить период событий для вывода:

```
journalctl -u luxmsbi-importer --since="2012-10-30 18:17:16" --until "4 hours ago"
```

9.5.1. Предоставление прав на просмотр журнала

Для предоставления прав доступа для ЧТЕНИЯ журнальных файлов компонентов Luxms BI нужно добавить учетную запись пользователя в следующие группы:

- `bi`.
- `systemd-journal`.

Пример:

```
usermod -aG bi,systemd-journal johndoe
```

10. Установка обновлений Luxms BI

При обновлении компонентов Luxms BI пакетами мы рекомендуем руководствоваться следующими принципами.

1. При наличии тестовой среды:

- поддержка актуальности тестовой среды;
- создание снимка файловой системы(snapshot) перед обновлением, если это возможно;
- выполнение обновления;
- актуализация конфигурационных файлов;
- проведение функционального тестирования с привлечением Бизнес-пользователей;
- принятия решения об обновлении продуктового контура;
- создание снимка файловой системы(snapshot) перед обновлением продуктовой среды, если это возможно;
- обновление продуктовой среды;
- актуализация конфигурационных файлов;
- проверка работоспособности;
- принятие решения об успешности обновления.

2. При отсутсвии актуальной тестовой схемы:

- создание снимка файловой системы(snapshot) перед обновлением продуктовой среды, если это возможно;
- обновление продуктовой среды;
- актуализация конфигурационных файлов;
- проверка работоспособности;
- принятие решения об успешности обновления.

3. Критичные ресурсы, требующие обязательное снятие резервной копии:

- конфигурационные файлы, расположенные по пути `/opt/luxmsbi/conf`;
- метаданные Luxms BI - БД `mi`.

4. При обновлении среды с кластером Postgres, возможно обновление без снятия копии БД:

- остановка сервиса Patroni на хосте со статусом `Replica`;
- выполнение обновление на хосте со статусом `Primary`;
- проверка работоспособности;
- принятие решения оо успешности обновления;
- при неудачном обновлении, остановка обновленного узла кластера БД и запуск ранее остановленной реплики.

10.1. Установка обновлений компонентов, кроме БД

10.1.1. Для RPM-based ОС

Для получения списка и версий установленных пакетов, выполните команду:

```
sudo rpm -qa | grep luxms
```

Установка обновлений компонентов Luxms BI производится обновлением пакетов:

```
sudo dnf -y update luxmsbi-web
```

При необходимости отката на предыдущую версию компонента используйте команду:

```
sudo dnf -y downgrade luxmsbi-web
```

10.1.2. Для DEB-based ОС

Для получения списка и версий установленных пакетов, выполните команду:

```
sudo apt list --installed | grep luxms
```

Установка обновлений компонентов Luxms BI производится обновлением пакетов:

```
sudo apt -y install luxmsbi-web
```

При необходимости отката на предыдущую версию компонента используйте команду с указанием конкретной версии:

```
sudo apt -y install luxmsbi-web=8.9.0-20220913.alse-1.7
```

10.2. Актуализация конфигурационных файлов

Обновления пакетов может включать в себя обновление конфигурационных файлов. Например в связи с появлением нового или изменением существующего функционала. Что требует приведения конфигурационных файлов в соответсвии с обновленной версией очень **Важным**. Поскольку конфигурационные файлы содержат значения относящиеся к конкретной инсталляции, например данные по подключению или DNS/IP-адреса, выполнение такой операции в автоматическом режиме пакеты не предусматривают. Актуализация конфигурационных файлов - это ручная часть работы Системного администратора при обновлениях.

10.2.1. Для RPM-based ОС

В RPM-пакетах это решается просто, при обновлении/удалении конфигурационного файла всегда создаются резервные копии и текущий конфигурационный файл не изменяется.

Все что нужно сделать после обновления RPM-пакета.

1. Определить список конфигурационных файлов конкретного пакета, например:

```
rpm -qc luxmsbi-appserver-mono
/etc/sysconfig/luxmsbi-appserver
/opt/luxmsbi/conf/appserver/application.properties
/usr/lib/firewalld/services/luxmsbi-appserver.xml
```

2. Проверить наличие указанных файлов с дополнительным суффиксом `.rpmnew`

> Конфигурационные файлы с суффиксом `.rpmnew` создаются при установке другой версии пакета, только если текущий конфигурационный файл отличается от эталонного в уже установленном пакете.

```
ls -la /etc/sysconfig/luxmsbi-appserver* \
       /opt/luxmsbi/conf/appserver/application.properties* \
       /usr/lib/firewalld/services/luxmsbi-appserver.xml*-
rw-r--r--. 1 root root  321 Jan 11 19:34 /etc/sysconfig/luxmsbi-appserver-
rw-r-----. 1 bi   bi   5346 Jan 12 04:19 ↩
  /opt/luxmsbi/conf/appserver/application.properties-
rw-r-----. 1 bi   bi   5350 Jan 12 14:28 ↩
  /opt/luxmsbi/conf/appserver/application.properties.rpmnew-
rw-r--r--. 1 root root  190 Jan 11 10:42 /usr/lib/firewalld/services/luxmsbi-↩
  appserver.xml
```

3. При наличии таких файлов, сверить изменения и при необходимости перенести из `.rpmnew` в текущий конфигурационный файл недостающие директивы.

```
diff /opt/luxmsbi/conf/appserver/application.properties*
30c30
< luxmsbi.datasource.password=non-bi---

> luxmsbi.datasource.password=bi
```

> В данном случае изменен пароль в текущем конфиге, а новый конфиг содержит пароль по умолчанию. Эти изменения не требует актуализации, переноса

10.2.2. Для DEB-based ОС

В мире Debian совсем другой подход к конфигурационным файлам. Если Вы выполняете все обновления “линейно” и правильно, то различия ***базового(входящего в состав пакета)*** конфигурационного файла новоко пакета по сравнению с установленным пакетом, породит диалог. Диалог с возможными опциями при изменении ***“базового”*** конфигурационного файла:

> Применяемое решение для визуализации Диалога с опциями зависти от переменной окружения **DEBIAN_FRONTEND**. В наших примерах мы предоставляем вывод для значения **readline**

```
apt install luxmsbi-web
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  luxmsbi-web
1 upgraded, 0 newly installed, 0 to remove and 259 not upgraded.
Need to get 0 B/29.1 MB of archives.
After this operation, 6105 kB of additional disk space will be used.
(Reading database ... 118877 files and directories currently installed.)
Preparing to unpack .../luxmsbi-web_9.2.14-20231219.alse-1.7_amd64.deb ...
Unpacking luxmsbi-web (9.2.14-20231219.alse-1.7) over (9.2.12-20231122.alse-
  1.7) ...
Setting up luxmsbi-web (9.2.14-20231219.alse-1.7) ...

Configuration file '/opt/luxmsbi/conf/luxmsbi-web-settings.js'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** luxmsbi-web-settings.js (Y/I/N/O/D/Z) [default=N] ?
Progress: [ 40%] [####......]
```

Если Вы до обновления внесли изменения в конфигурационный файл, то эти изменения сохранятся, диалог с опциями не визуализируется. При условии, что ***“базовый”*** конфигурационный файл в новом пакете не отличается от такого же ***“базового”*** конфигурационного файла установленной ранее версии.

Сложности возникнут только в одном случае - конфигурационный файл будет поврежден/-потерян до выполнения обновления. Вы не сможете восстановить конфигурационный файл из пакета - здесь поможет только резервное копирование `/opt/luxmsbi/conf`.

Кстати, посмотреть список конфигурационных файлов можно с помощью команды:

```
 cat /var/lib/dpkg/info/luxmsbi-web.conffiles
/opt/luxmsbi/conf/luxmsbi-web-settings.js
```

```
/opt/luxmsbi/conf/nginx/nginx.conf
/opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl
/opt/luxmsbi/conf/nginx/lua/aalcfg.lua
/opt/luxmsbi/conf/nginx/lua/bicfg.lua
/opt/luxmsbi/conf/nginx/conf.d/entrypoint.conf
/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-core.conf
/opt/luxmsbi/conf/nginx/conf.d/upstreams.conf
/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-appserver.location
/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-bins.location
```

10.3. Установка обновлений пакета БД luxmsbi-pg

При установке пакета `luxmsbi-pg` в пост-инсталляционном скрипте реализована следующая логика:

- При установке пакета скрипт использует переменную окружения `PGDATA` для определения расположения файлов БД. Используйте `PGDATA` при установке БД в нестандартном расположении.
- Установка пакета на “чистую” БД автоматически создает БД для Luxms BI с именем `mi`.
- Установка пакета на БД с уже существующей базой данных `mi` не вносит изменения в существующую БД;

Установка пакета `luxmsbi-pg` во всех случаях сохраняет в файловой системе сервера, `/usr/share/luxmsb-pg/`:

- Дамп БД соответсвующий версии пакета, сохраняется только одна версия дампа.
- SQL-скрипты обновлений для БД.
- Shell-скрипты для установки дампа и обновлений БД в ручном режиме.

Если при установке пакета БД была недоступна или не выставлена переменная окружения `PGDATA`, то развертывание бызы `mi` может быть выполненно в ручном режиме:

```
su - postgres -c /usr/share/luxmsbi-pg/luxmsbi-setupdb.sh /usr/share/luxmsbi-↩
  pg/luxmsbi-dump.sql.gz
```

10.3.1. Очистка, возврат первоначального состояния БД

При необходимости восстановления первоначального состояния БД нужно запустить предыдущую команду с ключом `-force`:

```
su - postgres -c /usr/share/luxmsbi-pg/luxmsbi-setupdb.sh /usr/share/luxmsbi-↩
  pg/luxmsbi-dump.sql.gz --force
```

При этом существующая БД `mi` не будет утрачена, а переименована в `mi_$(date +%Y%m%d_%H%M%S)`.

10.3.2. Обновление БД

До начала процесса обновления БД рекомендуем снять резервную копию БД, смотрите раздел Резервное копирование.

Скрипты обновления БД поставляются в составе пакетов `luxmsbi-pg/luxmsbi-pgpro`, поэтому необходимо установить необходимую версию соответсвующего пакета на ВСЕ узлы кластера или хост содержащий БД.

Для RPM-based ОС:

```
sudo dnf -y install luxmsbi-pg
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-pg
```

Установка обновлений БД может выполняться только из командной строки, два варианта запуска обновления:

1) Кумулятивная установка - установка всех необходимых обновлений:

```
su - postgres -c "/usr/share/luxmsbi-pg/luxmsbi-setupdb.sh --upgrade"
```

2) Выборочная установка - установка скрипта конкретной версии обновления БД:

```
su - postgres -c "/usr/share/luxmsbi-pg/luxmsbi-setupdb.sh --upgrade ↩
  /usr/share/luxmsbi-pg/upgradeDB-7.0.0.sql"
```

При установке обновлений БД проверяется текущая версия существующей БД `mi` и, при отсутствии необходимости обновления, изменения в БД не выполняются.

SQL-скрипты обновления БД выполняются в транзакции. При возникновении ошибки в процессе установки все изменения БД отменяются и shell-скрипт выдает информацию о возникшей ошибке.

10.3.3. Обновление БД по требованиям Клиента

При аварийных ситуациях или в случае дополнительного изменения БД под требования клиентов поставка обновлений может производиться в виде SQL-файла с прилагаемой инструкцией по применению.

Это не обычный вариант внесения изменения, но иногда и это решение используется.

11. Резервное копирование

Резервное копирование Luxms BI должно включать в себя, но не ограничиваться, следующим перечнем ресурсов:

- Конфигурационные файлы компонентов.
- Данные БД.

Периодичность снятия резервных копий определяется владельцем инсталляции Luxms BI в соответствии с внутренней политикой или отраслевыми стандартами. Мы можем только рекомендовать параметры резервирования.

11.1. Настройка резервного копирования конфигурации

Конфигурация компонентов Luxms BI консолидирована в папке `/opt/luxmsbi/conf` файловой системы. Изменение конфигурации компонентов производится только в ручном режиме. Установка новых или возврат к старым версиям пакетов включает в себя функционал сохранения конфигурационных файлов.

Мы рекомендуем создание резервных копий конфигурационных файлов не реже одного раза в день. Период хранения резервных копий - не менее 7 дней.

11.2. Настройка резервного копирования БД

> Настоятельно рекомендуем не хранить резервные копии локально, на той же машине, что и сама база.

Мы рекомендуем создание резервных копий данных БД не реже одного раза в день. Период хранения резервных копий - не менее 7 дней.

> Рекомендуем учитывать при утверждении планов резервного копирования следующее:
> “Временные затраты на восстановление журналов БД после восстановления резервной копии может быть существенно больше затрат на повторную загрузку данных из первичных источников.”

11.2.1. Настройка разрешений доступа к БД

Оптимальное решение - снятие резервной копии базы данных с внешнего хоста. Для настройки разрешений доступа к кластерной БД, управляемой с помощью Patroni, необходимо:

1. На всех хостах кластера (т.к. роль Leader может быть передана любому члену кластера) добавить разрешение для доступа к БД для системы резервного копирования.

Нам требуется внести строчку с IP адресом хоста, который будет участвовать в резервировании, в массив `postgresql.pg_hba`:

```
postgresql:
  pgpass: /pgdata/.pgpass
  listen: 0.0.0.0:5432
  connect_address: "172.16.32.112:5432"
  data_dir: /pgdata/data
  bin_dir: /usr/pgsql-11/bin/
  pg_rewind:
    username: postgres
    password: "password"
  pg_hba:
    - local all postgres peer
    - host all all 0.0.0.0/0 md5
    - host replication replicator 127.0.0.1/32 md5
    - host replication replicator 172.16.32.112/32 md5
    - host replication replicator 172.16.32.113/32 md5
    - host replication replicator 172.16.32.155/32 md5 <--- добавить сюда хост, с которого будут делаться бэкап

  replication:
    username: replicator
    password: "password"
  superuser:
    username: postgres
    password: "password"
```

2. Затем нужно дать команды сервису patroni, чтобы он перезапустил свою службу и обновил измененную конфигурацию на всех узлах кластера. Данные команды подаются на одном из узлов кластера и распространяются на все автоматически:

```
 patronictl -c /opt/patroni/etc/patroni.yml reload db-main


|   Cluster    |        Member        |      Host      |  Role  |  State   | TL | Lag in MB |
|--------------|----------------------|----------------|--------|----------|----|-----------|
| db-main      | bi-pg1               | 172.16.32.112 | Leader | running |  1 |           |
| db-main      | bi-pg2               | 172.16.32.113 |        | running |  1 |           |
```

```
        0 |

Are you sure you want to reload members bi-pg1, bi-pg2? [y/N]: y
Reload request received for member bi-pg1 and will be processed within 10 seconds
Reload request received for member bi-pg2 and will be processed within 10 seconds
```

3. После обновления конфигурации перезапустим сервис patroni вместе с PostgreSQL:

```
[root@bi-pg1 ~]# patronictl -c /opt/patroni/etc/patroni.yml restart db-main

|   Cluster   |        Member       |      Host     |  Role  |  State  | TL | Lag in MB |
|-------------|---------------------|---------------|--------|---------|----|--------|
| db-main     | bi-pg1              | 172.16.32.112 | Leader | running |  1 |        |
| db-main     | bi-pg2              | 172.16.32.113 |        | running |  1 |        0 |


When should the restart take place (e.g. 2021-03-15T14:38)  [now]:
Are you sure you want to restart members bi-pg1, bi-pg2? [y/N]: y
Restart if the PostgreSQL version is less than provided (e.g. 9.5.2)  []:
Success: restart on member bi-pg1
Success: restart on member bi-pg2
```

11.2.2. Снятие резервной копии

Для получения резервной копии БД необходимо выполнить команду:

```
pg_basebackup -d postgresql://replicator:password@172.16.32.113 \
              --checkpoint=fast \
              -D /backup \
              -P -Ft -z -Xs
```

По окончании снятия резервной копии, получим следующие файлы:

```
[root@localhost backup]# ls -l
total 11848-
rw-------. 1 root root 12110482 Mar 15 15:50 base.tar.gz-
rw-------. 1 root root    18318 Mar 15 15:50 pg_wal.tar.gz
```

Используются следующие ключи:

- `-D` - директория, куда будут складываться бэкапы (должна быть пустой).
- `-F` - формат выходного файла, в нашем случае tar архив.
- `-z` - включаем gzip сжатие.
- `-Xs` - передавать журнал предзаписи в процессе создания резервной копии.

> Рекомендация по безопасности: вместо явного указания в строке подключения имени и пароля пользователя используйте параметром `-U` и вводом пароля в командной строке.

Полученные файлы должны перемещаться на устройства долговременного хранения.

11.2.3. Восстановление данных из резервной копии

Останавливаем весь кластер PostgreSQL. Нужно выполнить на каждом хосте:

```
systemctl stop patroni
```

Чтобы убедиться, что все хосты остановлены выполняем команду на всех хостах кластера:

```
[root@bi-pg1 ~]# patronictl -c /opt/patroni/etc/patroni.yml list
```

Cluster	Member	Host	Role	State	TL	Lag in MB
db-main	bi-pg1	172.16.32.112		stopped	0	

Копируем данные резервной копии на один из хостов кластера, на котором будет поднята роль Leader. И выполняем на нём следующие команды:

```
rm -rf /pgdata/data/*
tar -xzf base.tar.gz -C /pgdata/data/
tar -xzf pg_wal.tar.gz -C /pgdata/data/pg_wal/
```

Пробуем запустить хост с базой:

```
systemctl start patroni
```

Проверяем на ошибки лог файлы: `/pgdata/data/log/postgredsql-*.log`. При удачном восстановлении в лог файлах будут такие строчки:

```
2021-03-15 15:30:03.727 MSK [14685] LOG:  archive recovery complete
2021-03-15 15:30:03.732 MSK [14682] LOG:  database system is ready to accept ↩
  connections
```

Если всё в порядке, то запускаем другие узлы кластера, через systemctl start patroni.

Проверяем, все ли работает правильно:

```
patronictl -c /opt/patroni/etc/patroni.yml list


|   Cluster    |        Member        |      Host      |  Role  |  State   | TL | ↩
  Lag in MB |
|--------------|----------------------|----------------|--------|----------|----|---↩
  --------|
```

```
| db-main     | bi-pg1              | 172.16.32.112 | Leader | running |  4 |
          |
| db-main     | bi-pg2              | 172.16.32.113 |        | running |  3 |
      109 |
```

Видно, что после восстановления у нас появился Lag in 109MB, для того, чтобы реплика не расходилась с мастером, её нужно реинициализировать. После чего она догонит мастера.

```
patronictl -c /opt/patroni/etc/patroni.yml reinit db-main

|   Cluster   |        Member       |      Host     |  Role  |  State  | TL | Lag in MB |
|-------------|---------------------|---------------|--------|---------|----|-------|
| rmr-db-main | rzd-skimcss-bi-pg-1 | 172.16.32.112 | Leader | running |  4 |
          |
| rmr-db-main | rzd-skimcss-bi-pg-2 | 172.16.32.113 |        | running |  3 |
      109 |

Which member do you want to reinitialize [bi-pg1, bi-pg-]? []: bi-pg2
Are you sure you want to reinitialize members bi-pg2? [y/N]: y
Success: reinitialize for member bi-pg2
```

12. Мониторинг компонентов Luxms BI

Каждая конкретная инсталляция Luxms BI может иметь различное ПО для мониторинга работоспособности и доступности как самой системы Luxms BI, так и ее компонентов. Поэтому мы предоставляем минимальные рекомендации по мониторингу компонентов.

Перечень элементов мониторинга включает в себя:

- Мониторинг очевидных критичных точек, влияющих на работоспособность системы.
- Дополнительные элементы, обнаруженные при нештатных ситуациях в инфраструктуре наших клиентов, в продуктовой эксплуатации.

При развертывании системы совместно с Consul DCS рекомендуется использование Consul API для мониторинга сервисов.

Дополнительно рекомендуется организовать мониторинг содержимого журнальных файлов.

Мониторинг параметров аппаратного обеспечения и ОС узлов должен быть реализован в соответствии с внутренним регламентом или отраслевыми стандартами.

12.1. Мониторинг БД

Мониторинг не резервируемого сервера БД должен включать в себя:

- Мониторинг доступности БД;
- Мониторинг свободного места файловой системы, используемой для хранения БД и журналов БД.

12.2. Мониторинг сервиса Core (luxmsbi-pg)

- URI: `/api/healthcheck`;
- Тип запроса: `HEAD`;
- Ожидаемый HTTP статус ответа: `204`.

Доступно, начиная с версии luxmsbi-pg-8.8.11

12.3. Мониторинг сервиса App Server (luxmsbi-appserver)

12.3.1. Health

- HTTP API port: `8080`;
- URI `/actuator/health` (отправка запроса с localhost);
- Тип запроса `GET`;
- Ожидаемый HTTP статус ответа: `200`;
- Ожидаемый ответ (JSON): `{"status":"UP"}`.

12.3.2. Prometheus metrics

- HTTP API port: `8080`;
- URI `/actuator/prometheus` (отправка запроса с localhost);
- Тип запроса `GET`;
- Ожидаемый HTTP статус ответа: `200`;
- Ожидаемый ответ (TEXT): метрики с описанием и значениями

12.4. Мониторинг сервиса Luxms BI Datagate (luxmsbi-datagate)

12.4.1. Health

- HTTP API port: `8200`;
- URI `/actuator/health` (отправка запроса с localhost);
- Тип запроса `GET`;
- Ожидаемый HTTP статус ответа: `200`;
- Ожидаемый ответ (JSON): `{"status":"UP"}`.

12.4.2. Prometheus metrics

- HTTP API port: `8200`;
- URI `/actuator/prometheus` (отправка запроса с localhost);
- Тип запроса `GET`;
- Ожидаемый HTTP статус ответа: `200`;
- Ожидаемый ответ (TEXT): метрики с описанием и значениями

13. Обращения в службу Поддержки

При эксплуатации Luxms BI могут возникать нештатные ситуации, которые не могут быть оперативно разрешены техническими специалистами Клиента. Это не недостаток Luxms BI как ПО, в большинстве случаев такие обращения возникают из-за:

- ошибок в конфигурационных параметрах
- ошибках в первичных данных
- авариях ИТ-инфраструктуры Клиента

Но иногда нештатные ситуации возникают на совокупности различных причин и приводят к некорректной работе Luxms BI.

В случае, если у Клиента есть договор Технической поддержки, можно оформить Обращение в службу Технической поддержки.

13.1. Подготовка диагностической информации

> Диагностическая информация может и содержит “чувствительную” информацию о настройках среды Luxms BI.
> Не допускайте ее передачи через открытые/публичные каналы связи.

Для полноценного анализа возникших технических проблем, нашей Службе Технической поддержки обязательно необходима информация о настройках Вашего экземпляра Luxms BI.

Передача вместе с Обращением диагностической информации, позволит исключить дополнительное взаимодействие по запросу, подготовки и отправки необходимой информации. Что значительно ускорит обработку Обращения.

13.1.1. Автоматизированный сбор диагностической информации

Утилита, входящая в пакет **luxmsbi-check**, позволяет в автоматизированном режиме выполнить сбор диагностической информации.

1. При установке необходимо выполнить команды:

Для RPM-based ОС:

```
sudo yum install luxmsbi-check
```

Для DEB-based ОС:

```
sudo apt install luxmsbi-check
```

2. Запуск сбора информации:

```
sudo /opt/luxmsbi/bin/check.sh
```

2.1. Запуск сбора информации при многоузловой сборке:

> В случае, если у Вас компоненты Luxms BI развернуты на нескольких узлах, отличных от места запуска утилиты, необходимо воспользоваться дополнительными ключами:
> -dbh=[IP | DNS] - для указания адреса базы данных -a=[IP | DNS] - для указания адреса компонента luxmsbi-appserver -dg=[IP | DNS] - для указания адреса компонента luxmsbi-datagate -db=[IP | DNS] - для указания адреса компонента luxms-databoring -w=[IP | DNS] - для указания адреса компонента luxmsbi-web -b=[IP | DNS] - для указания адреса компонента luxmsbi-bins -g=[IP | DNS] - для указания адреса компонента luxmsbi-gateway

```
sudo /opt/luxmsbi/bin/check.sh -dbh=192.168.101.27
```

3. После запуска необходимо ввести логин и пароль пользователя веб-версии.

```
sudo /opt/luxmsbi/bin/check.sh
Please, input username from Luxmsbi:
Please, input password from Luxmsbi:
```

После чего будут выведены статусы компонентов:

```
#######################################################################
|                        Check running services                       |
#######################################################################
             Service         Version      State    Running
             luxmsbi-web:     9.3.2        OK       9.3.2
        luxms-databoring:     9.3.1        OK       9.3.1
         luxmsbi-gateway:     9.3.0        OK       9.3.0
                   keydb:     6.3.3        OK
            luxmsbi-bins:     9.3.0        OK       9.3.0
               postgres*:      15.6        OK        15.6
 luxmsbi-headless-chrome:     9.3.0        OK
       luxmsbi-appserver:     9.3.1        OK       9.3.1
              luxmsbi-pg:     9.3.3        OK       9.3.3
        luxmsbi-datagate:     9.3.1       FAIL
             nats-server:   2.10.11        OK     2.10.11
```

4. Формирование полного отчета: После вывода статуса компонентов, от пользователя требуется выбрать формировать полный отчет или нет:

```
Do you want to generate full report? (y/n)
```

Отчет формируется на сервере в директории “/tmp” в формате *.rpt.

13.2. Данные ошибок из Веб браузера:

Необходимо открыть консоль разработчика в браузере (клавиша f12), выбрать вкладку Сеть (Network), после чего воспроизвести проблему и нажать правой кнопкой мыши на любой строке выбрав “Сохранить все как HAR с контентом”.

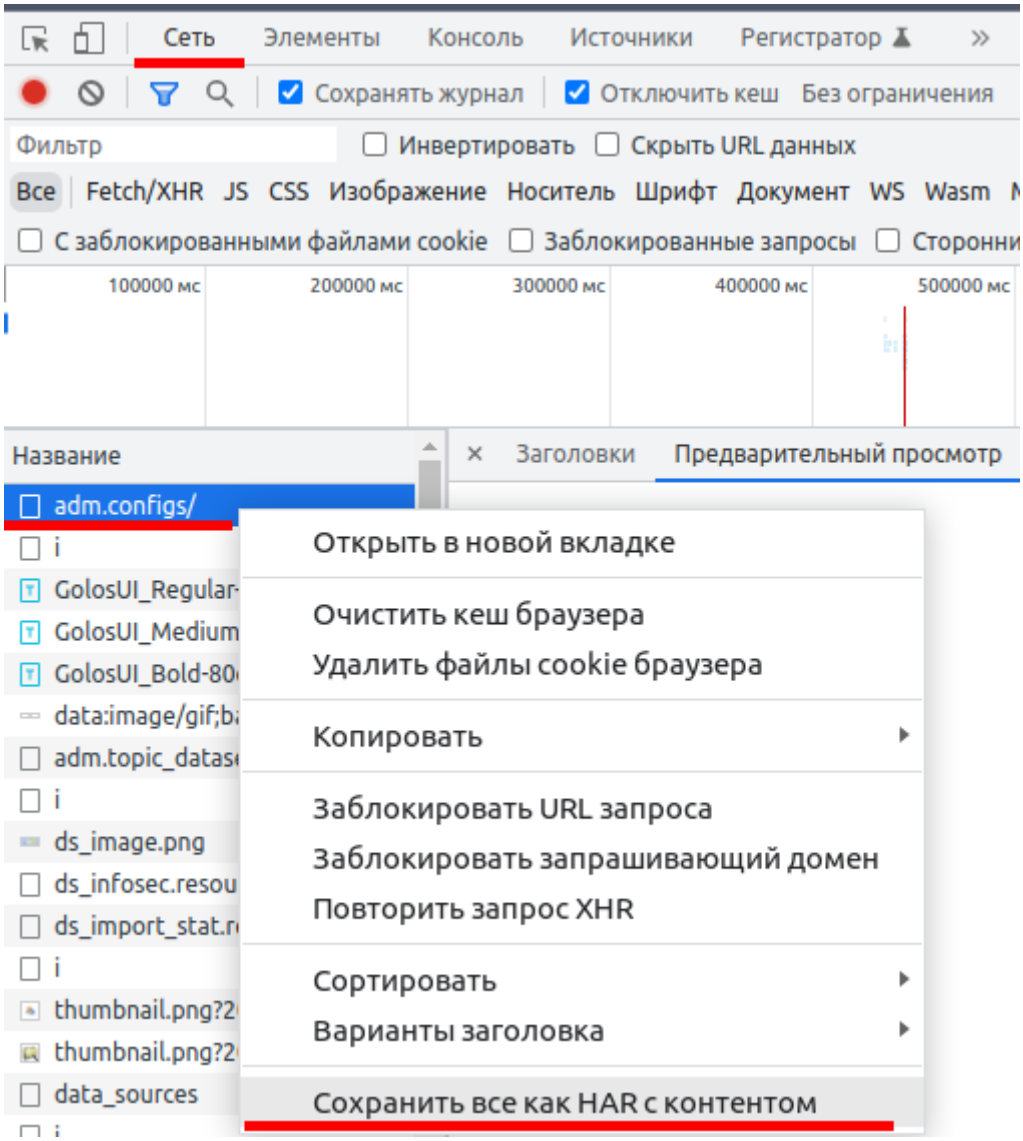


Рис. 13.1. **requesttosupport.png**

13.3. Оформление обращений

Файлы логов и/или ответов, необходимо направлять на почту support@yasp.ru

14. Процедура удаления компонентов Luxms BI

Luxms BI - сервисное (серверное) программное обеспечение. Данные, которыми манипулирует это ПО, могут быть конфиденциальными или обладать грифом секретности. Поэтому в большинстве случаев мы рекомендуем производить удаление данных без возможности восстановления с накопителей. Как минимум, этот принцип должен применяться к файлам Базы Данных, хранящим данные и метрики.

> **Безопасное удаление данных** На текущий момент не существует ГОСТа, определяющего требования и методики по програмному удалению данных, исключающему их восстановление. Мы рекомендуем изучить материалы по способам безопасного удаления данных, например, ознакомьтесь со статьей и комментариям к ней Shred и безвозвратное удаление файлов

В случае тестового развертывания ПО на локальной системе возможно удаление ПО и данных в менее строгом режиме. Но только в случае отсутствия конфиденциальных и секретных данных.

14.1. Удаление пакетов системы Luxms BI

Для удаления пакетов ПО необходимо выполнение команд, в зависимости от пакетного менеджера ОС серверов, на всех серверах.

Для RPM-based ОС:

```
dnf -y remove luxms*
```

для DEB-based ОС:

```
apt remove --purge luxms*
apt autoremove
```

14.2. Удаление конфигурационных файлов

Для удаления конфигурационных файлов достаточно рекурсивного удаления файлов по пути `/opt/luxmsbi`, после удаления пакетов ПО.

```
rm -rf /opt/luxmsbi
```

14.3. Удаление БД и данных

Luxms BI использует собственную базу данных для хранения конфигурации системы. Эта БД включает в себя учетные записи для интеграции с внешними источниками данных. Данные, предоставляемые через Web-интерфейс Luxms BI, в большинстве случаев расположены во внешних источниках. Тип и ПО используемое для хранения во внешних источниках данных могут быть различными. Вопрос удаления таких данных не входит в рамки данного раздела документации.

> До удаления экземпляра БД мы рекомендуем выполнить операции по безопасному удалению файлов данных БД, в соответствии с внутренними нормативными документами. В том числе и удаление структуры папок в файловой системе.

Для удаления можно использовать следующие команды, в зависимости от используемого типа БД:

Для RPM-based ОС:

```
dnf -y remove postgresql* postgrespro*
```

для DEB-based ОС:

```
apt remove --purge postgresql* postgrespro*
apt autoremove
```

14.4. Удаление сопутствующего ПО

Наше программное обеспечение использует дополнительное ПО сторонних разработчиков для обеспечения отказоустойчивости и масштабирования решения. Ниже приведен список сторонних продуктов с соответствующими командами по удалению.

14.4.1. Сервис KeyDB

Решение для организации взаимодействия компонентов системы Luxms BI с хранением данных в памяти, KeyDB требует выполнения следующих команд:

> В приведенных скриптах используются значения по-умолчанию для рабочей директории сервиса и места хранения журнальных файлов. Укажите другие пути для удаления файлов при использовании других значений. Уточните их расположение в конфигурационном файле `/etc/keydb/keydb.conf`.

Для RPM-based ОС:

```
dnf -y remove keydb
```

```
rm -rf /var/log/keydb /var/lib/keydb
```

для DEB-based ОС:

```
apt remove --purge keydb
apt autoremove
rm -rf /var/log/keydb /var/lib/keydb
```

14.4.2. Среда исполнения Java

Если не требуется последующее использование среды исполнения Java 11 версии, необходимо выполнить следующие команды удаления:

Для RPM-based ОС:

```
dnf -y remove java-11-openjdk*
```

для DEB-based ОС:

```
apt remove --purge openjdk-11*
apt autoremove
```

14.4.3. Среда исполнения NodeJS

Часть компонентов Luxms BI используют среду исполнения на базе NodeJS 16 версии. Данная среда отличается от поставляемых в пакетных репозиториях версий:

Для RPM-based ОС:

```
dnf -y remove nodejs*
```

для DEB-based ОС:

```
apt remove --purge nodejs*
apt autoremove
```

14.4.4. DCS Consul

При развертывании Luxms BI с применением решений по масштабированию сервисов и обеспечению отказоустойчивости системы мы рекомендуем использование DCS Consul. Удаление этого ПО требует выполнения следующих команд:

> В приведенных скриптах используются значения по-умолчанию для рабочей директории сервиса и места хранения журнальных файлов. Укажите другие пути для удаления файлов при использовании других значений.

Для RPM-based ОС:

```
dnf -y remove consul*
rm -rf /var/lib/consul /etc/consul-template.
```

для DEB-based ОС:

```
apt remove --purge consul*
apt autoremove
rm -rf /var/lib/consul /etc/consul-template.d
```

Приложение А. Установка отказоустойчивой БД

Для обеспечения отказоустойчивости БД мы рекомендуем использование в качестве кластерного решения использование следующих компонентов:

- Hashicorp Consul DCS - в роле арбитра для кластернных ресурсов и сервисов.
- Patroni - в роли управляющего кластером PostgreSQL ПО.
- Dnsmasq - как кеширующий DNS-сервер, для разрешения имен зарегистрированных сервисов Consul.
- Дополнительное конфигурирование DHCP-клиента, при использовании динамического получения адреса.

Все вышеназванное ПО имеет открытый код и длительное время показывает стабильную работу. Мы включаем в рекомендации только протестированные нами версии ПО.

> Установка кластерной БД под управлением Patroni требует согласования параметров всех компонентов. Поэтому при развертывании мы рекомендуем использовать сценарии Ansible.

> Процесс установки ПО для кластеризации БД в данном разделе описывается для пояснения взаимодействия между компонентами.

Исходные параметры:

- На всех узлах с компонентами Luxms BI устанавливается HashiCorp Consul с режимом работы `server` или `client`.
- Для кластера HashiCorp Consul используется нечетное количество узлов в режиме `server` - 3.
- Для кластера БД PostgreSQL используется не менее 2-х узлов, для запуска экземпляра БД.

До начала установки DCS Consul необходимо определить единый ***секретный токен***, используемый для шифрования информационного обмена между всеми экземплярами Consul.

Установку Consul DCS рекомендуем выполнять из нашего репозитория. Отличие нашей сборки - мы добавили в пакет наш шаблон конфигурационного файла в формате HCL.

> Вы можете использовать репозиторий производителя. Для использования репозитория Производителя ПО, в связи с санкционными ограничениями, требуется использование прокси-сервера с отличным от отечественного GEO-локацией по IP-адресу.

A.1. Лицензионные ограничения Hashicorp Consul

В связи с изменением лицензионной политики компании Hashicorp с Mozilla Public License v2.0 на Business Source License v1.1, мы не можем более предлагать к развертыванию Consul DCS с версией выше чем **1.16.1**.

Версия Consul 1.16.1 - это последняя версия выпущенная под лицензией, не содержащей ограничений по распространению и коммерческому использованию.

Consul, в большей степени, используется в роли DCS для организации отказоустойчивого кластера БД Postgres. Дополнительно мы вложили в него логику проверок (HealthChecks) работоспособности сервисов Luxms BI и обеспечения их доступности.

Для организации работоспособности нашего решения эта версия содержала весь необходимый функционал. При этом за все время эксплуатации не было зафиксировано инцидентов информационный безопасности. В данный момент версия 1.16.1 доступна в наших репозиториях для установки и использования.

Мы начали процесс пересмотра технологий и поиск нового подхода для обеспечения контроля за узлами и компонентами Luxms BI. Это потребует достаточно много времени.

Наши ближайшие планы - это собрать пакеты Consul с версией 1.16.1 (и даже исправить в ней уязвимости, которые Производитель закрыл в декабре 2023 года), провести комплексное тестирование и предоставлять эту версию через свои репозитории в соответствие с открытой лицензией.

Так как мы не можем гарантировать своевременное предоставление исправлений безопасности для этого продукта, мы предлагаем Вам учесть необходимость дополнительных компенсационных мер в вопросах информационной безопасности:

- при развертывание Consul необходимо использовать уникальный ключ шифрования;
- при развертывании Consul использовать встроенные средства фильтрации сетевого трафика(Firewall) для ограничения доступа;
- ограничивать доступ к Web-консоли DCS Consul с использованием аутентификации пользователей.

A.2. Планирование DCS Consul

A.2.1. Типовая схема кластера

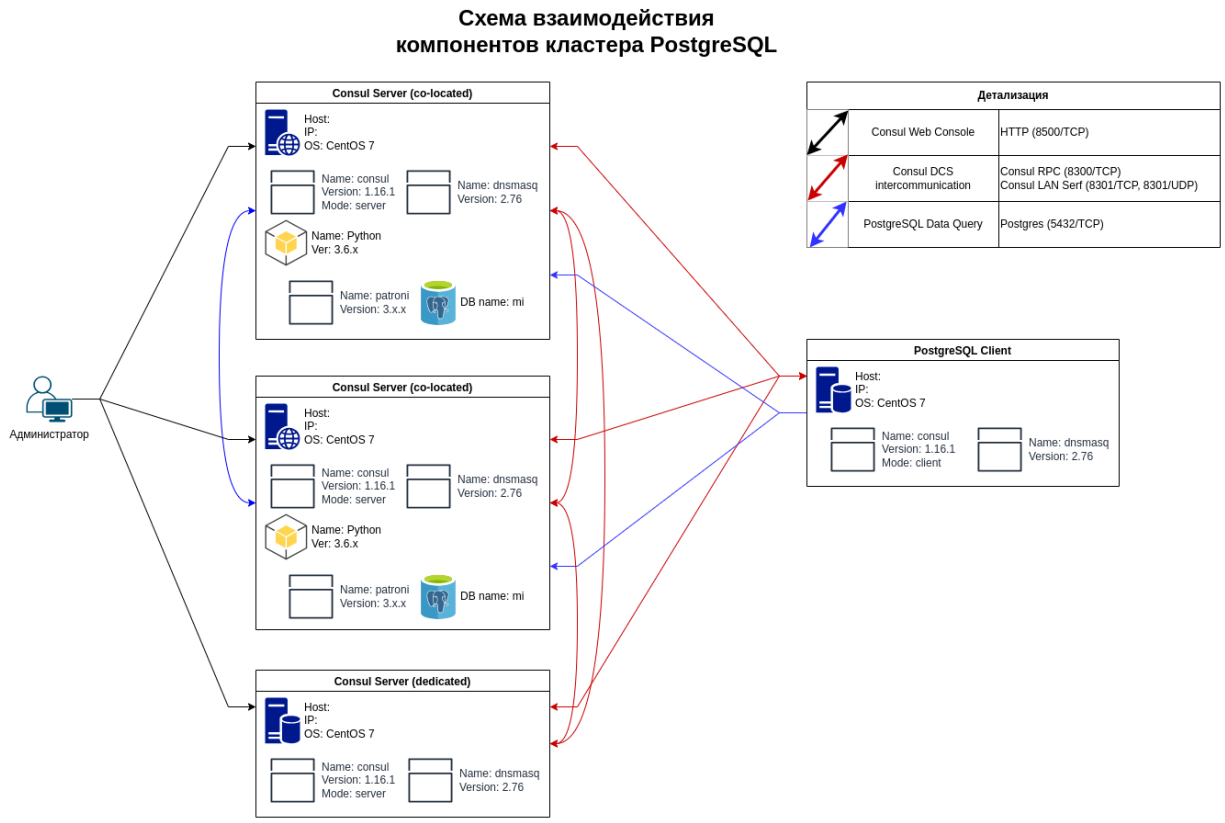


Рис. A.1. **Схема взаимодействия компонентов кластера PostgreSQL**

A.2.2. Планирование DCS Consul

Consul - это Distributed Control Service (**DCS**), гарантированно предоставляющий актуальную информацию по состоянию зарегистрированных сервисов. Продукт хорошо документирован производителем и обладает широким ассортиментом интеграционных решений. Для поддержки отказоустойчивости кластера PostgreSQL и разнесения нагрузки мы воспользуемся функционалом **DNS**-интерфейса.

> Необходимость установки ***Consul*** на все сервера программного комплекса обусловлена использованием его **DNS**-интерфейса для настройки доступа к БД в приложениях.

> Альтернативное решение - настройка **/etc/resolv.conf** может привести к значительному снижению времени отклика приложений, в случае выхода/вывода из рабочего режима одного из настроенных серверов имен.

Для начала установки необходимо:

1. Определить перечень узлов, на которых Consul агент должен запускаться в серверном режиме.

На каждый узел программного комплекса **Luxms BI** должен быть установлен агент ***Consul***, работающий в серверном или клиентском режиме. Так как для обеспечения работы Consul требуется нечетное количество агентов в серверном режиме (рекомендуемое количество **3** и не более **7**) большая часть узлов использует агента в клиентском режиме. Режим работы определяется флагом `server(bool)` файла конфигурации.

2. Для настройки файла конфигурации необходимо сгенерировать ***единый секретный token***.

Общий секретный token используется на всех узлах программного комплекса. Можно использовать собственный генератор или любую из предложенных ниже команд:

```
openssl rand -base64 32
```

Или при установке первого узла запустить команду:

```
/usr/sbin/consul keygen
```

A.3. Установка и настройка Consul DCS

Как было сказано выше, пакет consul находится в репозиториях Luxms. Поэтому установку можно выполнить без дополнительных настроек.

1. Установка:

Для RPM-based ОС:

```
sudo dnf -y install consul
```

Для DEB-based ОС:

```
sudo apt-get update && sudo apt-get -y install consul
```

2. Настройка конфигурационного файла:

Для первоначального запуска предлагаем использовать следующие конфигурационные файлы. Замените значение параметра `encrypt` на преварительно сгенерированный ***секретный токен***:

Конфигурационный файл для работы в режиме **server**, /etc/consul.d/consul.hcl

```
datacenter = "luxmsbi"
data_dir = "/opt/consul"
bind_addr = "0.0.0.0"
client_addr = "0.0.0.0"
advertise_addr = "10.0.0.11"
domain = "consul"
```

```
enable_local_script_checks = true
dns_config = {
  enable_truncate = true
  only_passing = true
}
recursors = [ "127.0.0.1" ]
enable_syslog = true
encrypt = "5wDTh+YLWG5DTDDfEeWkQ1j9J72+aJ3NOavqTRaLlUA="
leave_on_terminate = true
log_level = "INFO"
rejoin_after_leave = true
retry_join = [
  "consul-server-01.localnet",
  "consul-server-02.localnet",
  "consul-server-03.localnet"]

server = true
bootstrap_expect = 3
ui_config = {
  enabled = true
}
```

Конфигурационный файл для работы в режиме **client**, /etc/consul.d/consul.hcl

```
datacenter = "luxmsbi"
data_dir = "/opt/consul"
bind_addr = "0.0.0.0"
client_addr = "0.0.0.0"
advertise_addr = "10.0.0.5"
domain = "consul"
enable_local_script_checks = true
dns_config = {
  enable_truncate = true
  only_passing = true
}
recursors = [ "127.0.0.1" ]
enable_syslog = true
encrypt = "5wDTh+YLWG5DTDDfEeWkQ1j9J72+aJ3NOavqTRaLlUA="
leave_on_terminate = true
log_level = "INFO"
rejoin_after_leave = true
retry_join = [
  "consul-server-01.localnet",
  "consul-server-02.localnet",
  "consul-server-03.localnet"]
```

- **datacenter** — привязка сервера к конкретному датацентру. Нужен для логического разделения. Серверы с одинаковым датацентром должны находиться в одной локальной сети.
- **data_dir** — каталог для хранения данных.
- **bind_addr** — адрес, на котором будет слушать наш сервер Consul. Это может быть IP любого из наших сетевых интерфейсов или, как в данном примере, все.
- **client_addr** — адрес, к которому будут привязаны клиентские интерфейсы.

- **advertise_addr** — адрес, которой мы анонсируем другим узлам кластера. Это должен быть адрес, назначенный на одном из интерфейсов сервера.
- **domain** — домен, в котором будет зарегистрирован сервис.
- **enable_local_script_checks** — разрешает на агенте проверку работоспособности.
- **dns_config** — параметры для настройки DNS.
- **recursors** — адреса вышестоящих DNS-серверов, которые используются для рекурсивного разрешения запросов, если они не находятся внутри домена службы Consul.
- **enable_syslog** — разрешение на ведение лога.
- **encrypt** — ключ для шифрования сетевого трафика. В качестве значения используем сгенерированный ранее.
- **leave_on_terminate** — при получении сигнала на остановку процесса консула, корректно отключать ноду от кластера.
- **log_level** — минимальный уровень события для отображения в логе. Возможны варианты “trace”, “debug”, “info”, “warn”, and “err”.
- **rejoin_after_leave** — по умолчанию, нода, покидающая кластер, не присоединяется к нему автоматически. Данная опция позволяет управлять данным поведением.
- **retry_join** — перечисляем узлы, к которым можно присоединять кластер. Процесс будет повторяться, пока не завершится успешно.
- **server** — режим работы сервера.
- **start_join** — список узлов кластера, к которым пробуем присоединиться при загрузке сервера.
- **ui_config** — конфигурация для графического веб-интерфейса.

Проверяем корректность конфигурационного файла. Мы должны увидеть подтверждение корректности конфигурации в выводе:

```
/usr/bin/consul validate /etc/consul.d/consul.hcl

...
Configuration is valid!
```

3. Настройка разрешений локального firewall:

Для обеспечения сетевого взаимодействия агентов DCS Consul мы рекомендуем использование комплексных конфигурационных файлов для локальных решений по фильтрации сетевого трафика.

Для RPM-based ОС при установке consul создаются файлы описания сервисов для Firewalld следующего содержания:

Шаблон конфигурации сервиса Firewalld, consul-server.xml

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>consul-server</short>
  <description>Server access. Consul makes it simple for services to register ↩
  themselves and to discover other services via a DNS or HTTP interface. https:↩
  //www.consul.io/docs/install/ports.html </description>
  <port protocol="tcp" port="8300"/>
  <port protocol="tcp" port="8301"/>
  <port protocol="tcp" port="8302"/>
```

```
8   <port protocol="tcp" port="8502"/>
9   <port protocol="tcp" port="8600"/>

11  <port protocol="udp" port="8301"/>
12  <port protocol="udp" port="8302"/>
13  <port protocol="udp" port="8600"/>
14 </service>
```

Шаблон конфигурации сервиса Firewalld, consul-web.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <service>
3   <short>consul-web</short>
4   <description>Web access to Consul. Consul makes it simple for services to ↩
    register themselves and to discover other services via a DNS or HTTP ↩
    interface. https://www.consul.io/docs/install/ports.html </description>
5   <port protocol="tcp" port="8500"/>
6   <port protocol="tcp" port="8501"/>
7 </service>
```

Шаблон конфигурации сервиса Firewalld, consul-client.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <service>
3   <short>consul-client</short>
4   <description>Client access. Consul makes it simple for services to register ↩
    themselves and to discover other services via a DNS or HTTP interface. https:↩
    //www.consul.io/docs/install/ports.html </description>
5   <port protocol="tcp" port="8300"/>
6   <port protocol="tcp" port="8301"/>

8   <port protocol="udp" port="8301"/>
9 </service>
```

Правила описанные ниже базируются на схеме Продуктовая конфигурация с полным резервированием

Для обеспечения взаимодействия между узлами с ролью **server**. Добавление Firewalld-сервисов производится на на этих узлах, для примера на сервере **10.0.0.11** :

```
1 sudo firewall-cmd --permanent --new-service-from-file=consul-server.xml

3 sudo firewall-cmd --permanent \
4   --add-rich-rule='rule family="ipv4" source address="10.0.0.12" service name=↩
    "consul-server" accept'
5 sudo firewall-cmd --permanent \
6   --add-rich-rule='rule family="ipv4" source address="10.0.0.13" service name=↩
    "consul-server" accept'

8 sudo firewall-cmd --reload
```

Для ограничения доступа к Web-консоли Consul только с хостов принадлежащих администраторам, например **192.168.1.200**, на узлах с ролью **server** выполняются следующие ко-

манды:

```
sudo firewall-cmd --permanent --new-service-from-file=consul-web.xml

sudo firewall-cmd --permanent \
  --add-rich-rule='rule family="ipv4" source address="192.168.1.200" service name="consul-web" accept'

sudo firewall-cmd --reload
```

Для обеспечения взаимодействия между узлами(обнаружение узлов и проверка их работоспособности) с запущенным Consul, необходимо настроить сетевые ограничения. Наиболее оптимально использование сегментацию сетей, адреса сети и/или `ipset`.

На каждом узле с ролью **server** нужно запустить следующий перечень команд, например для сегментированной сети `10.0.0.0/28`:

```
sudo firewall-cmd --permanent --new-service-from-file=consul-client.xml

sudo firewall-cmd --permanent \
  --add-rich-rule='rule family="ipv4" source address="10.0.0.0/28" service name="consul-client" accept'

sudo firewall-cmd --reload
```

На каждом узлах с ролью **client** нужно запустить следующий перечень команд, например для `ipset`(небольшого списка IP-адресов и/или подсетей):

```
sudo firewall-cmd --permanent --new-service-from-file=consul-client.xml

sudo firewall-cmd --permanent --new-ipset=consulAgent --type=hash:net
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.5
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.6
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.7
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.8
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.9
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.10
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.11
sudo firewall-cmd --permanent --ipset=consulAgent --add-entry=10.0.0.12

sudo firewall-cmd --permanent \
  --add-rich-rule='rule family="ipv4" source ipset="consulAgent" service name="consul-client" accept'

sudo firewall-cmd --reload
```

Для DEB-based ОС при установке consul создаются файлы приложений UFW следующего содержания:

Шаблон конфигурации сервиса UFW, /etc/ufw/applications.d/consul-server

```
[consul-server]
title=HashiCorp Consul DCS
```

```
3 description=Consul makes it simple for services to register themselves and to ↩
    discover other services via a DNS or HTTP interface. https:↩
    //www.consul.io/docs/install/ports.html.
4 ports=8300:8302,8502,8600/tcp|8301:8302,8600/udp
```

Шаблон конфигурации сервиса UFW, /etc/ufw/applications.d/consul-client

```
1 [consul-client]
2 title=HashiCorp Consul Client
3 description=Consul makes it simple for services to register themselves and to ↩
    discover other services via a DNS or HTTP interface. https:↩
    //www.consul.io/docs/install/ports.html.
4 ports=8300:8301/tcp|8301/udp
```

Шаблон конфигурации сервиса UFW, /etc/ufw/applications.d/consul-web

```
1 [consul-web]
2 title=HashiCorp Consul Web UI
3 description=Consul makes it simple for services to register themselves and to ↩
    discover other services via a DNS or HTTP interface. https:↩
    //www.consul.io/docs/install/ports.html.
4 ports=8500:8501/tcp
```

Для добавление application в конфигурацию UFW выполните следующие команды:

```
1 sudo ufw app update --add-new
2 sudo ufw allow from 10.0.0.4 to any app consul-server
3 sudo ufw allow from 10.0.0.5 to any app consul-server
4 sudo ufw allow from 10.0.0.6 to any app consul-client
5 sudo ufw allow from 10.0.0.0 to any app consul-web
```

4. Запуск DCS Consul и проверка работоспособности:

Запускаем сервис Consul-а на всех узлах:

```
1 sudo systemctl enable consul.service --now
```

После запуска сервиса на всех узлах необходимо проверить статус узлов, используя командную строку:

```
1 # /usr/bin/consul members
```

Или через Web-интерфейс, который доступен по ссылке, на резрешенном сетевыми правилами сервере с ролью `server` - `http://<node>:8500/`.

A.4. Настройка разрешения ресурсов зоны .consul

Для обеспечения актуальности сведений о зарегистрированных сервисах Consul DNS-интерфейс разрешает имена без кэширования с TTL=0. Consul позволяет перенаправлять запросы на другие DNS-сервера, но более оптимальное решение - использование интеграции Consul и Dnsmasq для разрешения всех запросов.

A.4.1. Установка и настройка DNSMasq

Пакет **dnsmasq** опубликован в стандартных репозиториях, для его установки необходимо использовать следующую команду:

Для RPM-based ОС:

```
sudo dnf -y install dnsmasq
```

Для DEB-based ОС:

```
sudo apt -y install dnsmasq
```

После установки пакета необходимо откорректировать файл конфигурации или установить следующую минимальную конфигурацию:

Минимальный конфигурационный файл, /etc/dnsmasq.conf

```
# Configuration file for dnsmasq.
#
# Format is one option per line, legal options are the same
# as the long options legal on the command line. See
# "/usr/sbin/dnsmasq --help" or "man 8 dnsmasq" for details.

# Never forward plain names (without a dot or domain part)
domain-needed

# Add other name servers here, with domain specs if they are for
# non-public domains.
server=/consul/127.0.0.1#8600

# If you don't want dnsmasq to read /etc/hosts, uncomment the
# following line.
no-hosts

# Include all files in /etc/dnsmasq.d except RPM backup files
conf-dir=/etc/dnsmasq.d,.rpmnew,.rpmsave,.rpmorig
```

Разрешаем и запускаем сервис dnsmasq:

```
sudo systemctl enable dnsmasq --now
```

A.4.2. Дополнительная настройка ОС по разрешению имен

Если сервера используют статические IP-адреса, необходимо добавить в конфигурационный файл основного сетевого интерфейса параметр `DNS1=127.0.0.1`, назначая **DNSMasq** первичным.

Наиболее сложная тема - выбор между `стандартной` обработкой запросов на разрешение DNS-имен или использование `systemd-resolved`. Если Вы не готовы глубоко “копать” эту тему, позаботьтесь об наличии в секции `[main]` параметра:

```
[main]
...
dns=default
```

Если сервера используют динамическое получение IP-адреса, необходимо настроить DHCP-клиент. Создайте или добавьте следующую строку в конфигурационный файл `/etc/dhcp/dhclient.conf`:

```
prepend domain-name-servers 127.0.0.1;
```

Перезапускаем сервис сети для применения настроек:

```
sudo systemctl restart NetworkManager
```

A.4.3. Проверка разрешения DNS имен

Разрешение имен сервисов производится запросом:

- для разрешения узлов

```
<node>.node.<partition>.ap.<datacenter>.dc.<domain>
```

- для разрешения сервисов

```
[<tag>.]<service>.service[.<datacenter>].<domain>
```

Для разрешения имен сервисов в Consul DNS нет необходимости указывать имя, указанное в параметре конфигурации Consul-агента - datacentr. Если этот параметр отсутствует, то предполагается текущее значение у самого Consul-агента. т.е. следующие запросы вернут корректный IP-адрес:

- nslookup consul.service.luxmsbi.consul
- nslookup consul.service.consul

Если конфигурация компонентов выполнена корректно, то выполнение проверочного запроса должно вернуть перечень адресов:

```
nslookup consul.service.consul
...
Name:  consul.service.consul
Address: 10.0.2.5
Name:  consul.service.consul
Address: 10.0.2.7
Name:  consul.service.consul
Address: 10.0.2.6
```

A.5. Установка и настройка Patroni

С 2023 года, с учетом стремления Российских дистрибутивов Linux включить в себя пакеты `patroni`, мы поставляем свои пакеты для развертывания Менеджера отказоустойчивого кластера Postgres. В связи с чем, поставляемый нами пакет распологает(prefix) свои файлы в `/opt`. Основные цели:

- использование изолированного(VENV) экземпляра Python
- предоставление шаблона конфигурации, для Luxms BI

До установки Patroni необходимо установить пакеты БД и необходимых расширений в соответсвии с Разделом Установка и настройка сервера БД без развертывания БД Luxms BI.

A.5.1. Установка на RPM-based ОС

В зависимости от дистрибутива Linux, команды немного различаются.

а) Для Linux RPM (РЕД ОС, CentOS, Rocky и другие):

```
dnf install luxms-patroni
```

б) Для Linux DEB (Astra Linux):

```
apt install luxms-patroni
```

A.5.2. Установка конфигурации Patroni

Базовая конфигурация сервиса Patroni предоставлена ниже. Необходимо заменить некоторые значения параметров на соответствующие Вашим хостам в файле конфигурации `/etc/patroni/patroni.yml` (YAML):

- name - имя узла, рекомендуется установить DNS-имя узла.
- restapi.connect_adress - IP-адрес узла.
- postgresql.connect_address - IP-адрес узла.
- pg_hba.host(replication replicator), postgresql.pg_hba.host(replication replicator): сегмент сети узлов БД. Или несколько записей для IP-адресов узлов с маской `/32`.

ОБЯЗАТЕЛЬНО замените значения (идентичные на всех узлах БД) для паролей и, если необходимо, имен учетных записей.

Данные учетных записей:

- `restapi.auth`.
- `postgresql.pg_rewind`.
- `postgresql.replication`.
- `postgresql.superuser`.

Детальное описание параметров рекомендуется прочитать на сайте производителя.

Исходный вид файла конфигурации /opt/patroni/etc/patroni.yml:

```
name: hostname
scope: db_cluster

watchdog:
  mode: off

consul:
  host: "localhost:8500"
  register_service: true
  #token: <consul-acl-token>

restapi:
  listen: 0.0.0.0:8008
  #!#connect_address: "192.168.0.10:8008"
  auth: "patroni_rest_user:patroni_rest_password"

bootstrap:
  dcs:
    ttl: 30
    loop_wait: 10
    maximum_lag_on_failover: 1048576 # 1 megabyte in bytes
    postgresql:
      use_pg_rewind: true
      use_slots: true
      parameters:
        archive_mode: "off"
        wal_level: hot_standby
        archive_command: "mkdir -p ../wal_archive && test ! -f ..
  /wal_archive/%f && cp %p ../wal_archive/%f"
        max_wal_senders: 10
        wal_keep_segments: 8
        archive_timeout: 1800s
        max_replication_slots: 5
        max_connections: 100
        hot_standby: "on"
        wal_log_hints: "on"

initdb:
  - encoding: UTF8
  - locale: ru_RU.UTF8
  - data-checksums

pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
  - local all postgres peer
  - host all all 0.0.0.0/0 md5
  - host replication replicator 127.0.0.1/32 md5
  #!#- host replication replicator 192.168.0.20/32 md5


postgresql:
  use_unix_socket: true
  listen: 0.0.0.0:5432
  #!#connect_address: "192.168.0.10:5432"
```

```
  data_dir: /var/lib/pgpro/std-13/data
  bin_dir: /opt/pgpro/std-13/bin
  pg_rewind:
    username: postgres
  pg_hba:
    - local all postgres peer
    - host all all 0.0.0.0/0 md5
    - host replication replicator 127.0.0.1/32 md5
    #!#- host replication replicator 192.168.0.20/32 md5

  parameters:
    unix_socket_directories: /tmp

  replication:
    username: replicator
    password: "replicator_password"
  superuser:
    username: postgres
    password: "postgres_password"
```

После корректного заполнения конфигурационного файла необходимо разрешить автозапуск сервиса patroni и запустить его:

```
sudo systemctl enable patroni --now
```

A.5.3. Проверка работоспособности кластера БД

После запуска Patroni DNS-интерфейс Consul разрешает запросы для сервиса PostgreSQL, например:

```
[root@centos-4 ~]# dig master.postgresdb.service.consul +short
10.0.2.5
[root@centos-4 ~]# dig replica.postgresdb.service.consul +short
10.0.2.4
[root@centos-4 ~]# dig replica.postgresdb.service.consul SRV +short
1 1 5432 centos-2.local.node.dc0.consul.
```

Проверка статуса узлов кластера:

```
patronictl list
```

> После установки пакета `luxms-patroni` в shell пользователя root добавляется alias, который позволяет набирать команду `patronictl` без необходимости прописывать полный путь до конфигурационного файла. Чтобы данный alias применился, нужно зайти под пользователем root или, если мы уже под ним работает, зайти еще раз.

Если же мы работает не под пользователем root, команде `patronictl` нужно передавать аргумент с указанием пути до конфигурационного файла, например:

```
patronictl -c /etc/patroni/patroni.yml list
```

{.is-info}

Статус конкретного (например, postgresdb) кластера узла мы можем посмотреть командой:

```
patronictl list postgresdb
```

Пример ответа:

Cluster	Member	Host	Role	State	TL	Lag in MB
postgresdb	centos-1.local	10.0.2.5	Leader	running	25	
postgresdb	centos-2.local	10.0.2.4		running	25	0.0

Приложение B. Настройка журналирования событий

B.1. Рекомендации по настройке Journald

Одно из преимуществ Journald - возможность ограничивать поток сообщений для хранения. Этот механизм защищает файловую систему сервера от переполнения, т.е. обеспечивает работоспособность. Иногда есть необходимость корректировки конфигурации по умолчанию для параметров `RateLimitInterval` и `RateLimitBurst`.

Хранение журнальных записей производится в binary-формате, что позволяет существенно снизить объем и обеспечить высокую скорость доступа. Расположение журнальных файлов зависит от параметра конфигурации **Storage**, но фактически это 2 варианта.

1. Хранение в памяти - `/run/log/journal`:

- `Storage=auto` при отсутствии директории `/var/log/journal` (по умолчанию)
- `Storage=volatile`

2. Хранение в файловой системе - ‘/var/log/journal:

- `Storage=auto` при существовании директории `/var/log/journal`
- `Storage=persistent`

B.2. Рекомендации по хранению журнальных записей

1. Хранение журнальных записей в файловом виде в папке файловой системы `/var/log/journal/`.
2. Обеспечение необходимого дискового пространства для хранения журналов на срок не менее 7 дней, желательно до 30 дней.

При необходимости создайте дополнительное дисковое устройство для точки монтирования `/var/log/journal/`.

3. Минимальная конфигурация, конфигурационный файл `/etc/systemd/journald.conf`:

```
#Storage=auto
#Compress=yes
#Seal=yes
#SplitMode=uid
#SyncIntervalSec=5m
RateLimitInterval=1
RateLimitBurst=10000
#SystemMaxUse=
SystemKeepFree=20
#SystemMaxFileSize=
#RuntimeMaxUse=
#RuntimeKeepFree=
#RuntimeMaxFileSize=
#MaxRetentionSec=
#MaxFileSec=1month
#ForwardToSyslog=yes
#ForwardToKMsg=no
#ForwardToConsole=no
#ForwardToWall=yes
#TTYPath=/dev/console
#MaxLevelStore=debug
#MaxLevelSyslog=debug
#MaxLevelKMsg=notice
#MaxLevelConsole=info
#MaxLevelWall=emerg
#LineMax=48K
```

4. Проверить существование папки в файловой системе, при необходимости создать и выполнить перезапуск сервиса журнальных файлов:

```
[[ -d /var/log/journal ]] && \
    ( sudo mkdir -p /var/log/journal && sudo systemctl restart systemd-journald)
```

B.3. Проверка текущей конфигурации

1. Выполняем проверку режима работы Journald:

```
systemctl status journald
```

Проверьте полученный статус. Строка `Runtime journal is using` в статусе означает использование оперативной памяти для хранения журнальных записей. Т.е. после перезагрузки или аварийного отключения хоста журналы не сохранятся. Пример:

```
systemctl status systemd-journald
  systemd-journald.service - Journal Service
   Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static; ↩
  vendor preset: disabled)
   Active: active (running) since Thu 2020-11-19 21:01:43 MSK; 18s ago
     Docs: man:systemd-journald.service(8)
```

```
           man:journald.conf(5)
 Main PID: 2671 (systemd-journal)
   Status: "Processing requests..."
   CGroup: /system.slice/systemd-journald.service
             2671 /usr/lib/systemd/systemd-journald

Nov 19 21:01:43 localhost.localdomain systemd-journal[2671]: Runtime journal is
  using 8.0M (max allowed 91.9M, trying to leave 137.8M free of 910.5M
  available → current limit 91.9M).
Nov 19 21:01:43 localhost.localdomain systemd-journal[2671]: Journal started
```

Строка `Permanent journal is using` в статусе означает использование дисковой подсистемы. Т.е. после перезагрузки или аварийного отключения хоста журналы сохранятся.

2. Проверка доступного дискового пространства для хранения журнальных записей:

```
df -h
```

Проверьте, достаточно ли места на файловой системе, содержащей папку `/var/log/journal/`.

3. Проверка корректности конфигурационных параметров.

Наличие сообщений в журнальном файле `Suppressed xxxx messages` говорит о недостаточном значении параметра **RateLimitBurst** в конфигурационном файле `/etc/systemd/journald.conf` или о том, что приложение сконфигурировано неверно в части журналирования событий.

```
journalctl -u systemd-journald--

 Logs begin at Thu 2019-05-16 13:56:01 MSK, end at Thu 2020-11-19 22:42:04 MSK.
  --
Oct 28 08:26:01 rzd-skimn-d-app-1 systemd-journal[1580]: Suppressed 7894
  messages from /system.slice/luxmsbi_appserver.service
Oct 28 08:27:01 rzd-skimn-d-app-1 systemd-journal[1580]: Suppressed 7894
  messages from /system.slice/luxmsbi_appserver.service
```

B.4. Настройка учетных записей для просмотра журналов

При необходимости предоставления доступа на чтение к журналам приложений необходимо:

1) Для просмотра журнальных записей в системном журнале необходимо добавить учетную запись пользователя в группу **systemd-journal**:

```
usermod -aG systemd-journal username
```

2) Для просмотра журнальных записей в файлах:

```
usermod -aG bi username
```

B.5. Альтернативный вариант для более современных ОС

Операционные системы RHEL-based 7 (RedHat/CentOS/Oracle) Linux использует Systemd версии 219, которая не поддерживает расширенный функционал управления журнальными файлами. Начиная с версии Systemd 231, Journald поддерживает разделение потоков регистрации журнальных записей.
Например, для ОС CentOS 8 возможно настроить регистрацию событий для конкретного сервиса отдельным потоком, со своими ограничениями - Per unit size limit

Приложение C. Использование HAProxy (в процессе переработки)

Работоспособность PostgreSQL сервера сильно зависит от количества активных соединений к БД. Оптимальное количество процессов для обработки соединений 100-200. Настройка количества соединений производится в конфигурационных файлах сервера PostgreSQL. Например, `/var/lib/pgsql/11/data/postgresql.conf`.

> Рекомендуем использовать настройку конфигурации сервера с использованием команд `ALTER SYSTEM`. Это позволяет обеспечить применение измененных параметров при старте экземпляра PostgreSQL и сохраняет возможность отката к настройкам по умолчанию или предыдущим настройка с помощью корректировки/удаления файла `postgresql.auto.conf`.

По умолчанию система Luxms BI настраивает 150 соединений к БД. Это значение может быть изменено при эксплуатации при необходимости.

При увеличении количества активных пользователей системы Luxms BI установленное количество соединений может быть недостаточным и вызвать отказ в обслуживании. Для обеспечения работоспособности при высокой нагрузке мы рекомендуем использование HAProxy в качестве менеджера пула соединений к БД.

C.1. HAProxy в роли менеджера пула соединений

Для установки HAProxy необходимо выполнить следующий перечень команд:

```
sudo yum -y install haproxy
sudo setsebool -P haproxy_connect_any=1
cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.default
```

Ниже расположен конфигурационный файл HAProxy, включающий в себя настройки для:

- Журналирования событий балансировщика.
- Предоставления web-интерфейса для управления и просмотра статистики по балансировщику нагрузки.
- Обеспечение очереди для запросов к БД PostgreSQL.

Замените содержимое конфигурационного файла `/etc/haproxy/haproxy.cfg`, тем более что предыдущие команды создали его резервную копию:

```
global
    daemon
    user    haproxy
    group   haproxy
    pidfile /var/run/haproxy.pid
    log     /dev/log local0
    maxconn 102400

defaults
    log     global
    mode    tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s

listen stats
    bind    127.0.0.1:2000
    maxconn 100
    mode    http
    option  httplog
    stats   uri /stats
    stats   enable
    stats   refresh 1s
    stats   admin if LOCALHOST

listen postgres
    bind           *:5432
    maxconn        10240
    timeout        queue 30s
    server         local localhost:9898 maxconn 100
```

Поскольку запуск HAProxy произведен не в chroot-окружении, не требуется дополнительной настройки для организации журналирования событий сервиса - журнальные записи сохраняются в системном `journald`. Для просмотра журнальных записей достаточно набрать команду:

```
sudo journalctl -u haproxy
```

После корректировки/создания конфигураций, необходимо изменить порт для экземпляра PostgreSQL. Изменение порта экземпляра БД позволяет не производить многочисленных корректировок конфигурационных файлов компонентов системы Luxms BI. Выполните следующие команды:

```
su - postgres -c '/usr/pgsql11/bin/psql "ALTER SYSTEM SET PORT TO 9898;"'
sudo systemctl restart postgresql-11 haproxy
```

С этого момента система Luxms BI будет использовать HAProxy как менеджер пула соединений к БД PostgreSQL.

> Порт 9898/TCP уже зарегистрирован в SELinux как `postgresql_port_t`, поэтому дополнительных настроек безопасности не требуется.

Не забудьте добавить в профиль сервисной учетной записи `postgres` измененное значение порта, это облегчит работу с утилитами PostgreSQL:

```
---
 .bash_profile.old 2021-06-05 00:47:06.303382240 +0300+++
 .bash_profile 2021-06-05 00:37:04.264916442 +0300
@@ -1,6 +1,7 @@
 [ -f /etc/profile ] && source /etc/profile
 PGDATA=/var/lib/pgsql/11/data-
export PGDATA+
PGPORT=9898+
export PGDATA PGPORT
 # If you want to customize your settings,
 # Use the file below. This is not overridden
 # by the RPMS.
```

C.1.1. Подключение к web-интерфейсу HAProxy для просмотра статистики и управления

Рекомендуем использование SSH SOCKS-прокси и плагина для Вашего браузера, например, для FireFox добавьте FoxyProxy Standard.

И настройте расширение:

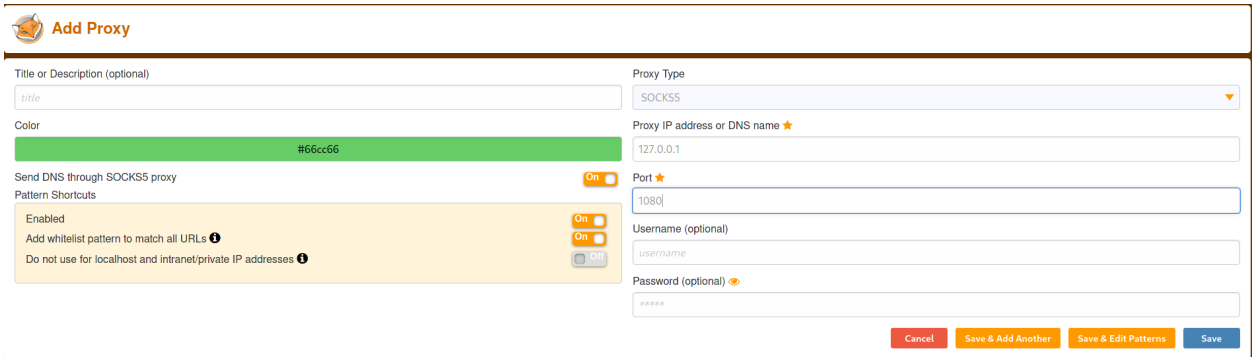


Рис. C.1. **foxyproxy.png**

Обеспечьте создание SOCKS-прокси через SSH-соединение

```
ssh -D 1080 <Luxms BI host IP/DNS>
```

В качестве альтернативы Вы можете откорректировать конфигурационный файл HAProxy (директиву bind, разрешив доступ к интерфейсу других хостов) и добавить разрешения для доступа в фаервол сервера.

C.1.2. Тюнинг операционной системы

Настройка сетевого стека ядра для хоста с установленным HAProxy в чем-то похожа на настройку тестирующего хоста, создающего нагрузку:

`/etc/sysctl.d/98-luxmsbi.conf`:

```
net.core.netdev_max_backlog = 5000
net.core.somaxconn = 65535
net.ipv4.ip_local_port_range = 1025 65000
net.ipv4.tcp_max_syn_backlog = 5000
net.ipv4.tcp_tw_reuse = 1
```

C.2. HAProxy как балансировщик для кластера

Кластер **PostgreSQL** под управлением **Patroni[consul]** позволяет предоставлять доступ к экземпляру базы данных с возможностью добавления/изменения данных и к нескольким экземплярам с возможностью только чтения данных. Перенаправление запросов на чтение на выделенные сервера позволяет снизить общую нагрузку на основной экземпляр и обеспечить устойчивую работоспособность системы.

В случае выхода/вывода из рабочего режима одного из узлов кластера PostgreSQL Patroni оперативно выполняет передачу ролей и реконфигурацию кластера. Что требует такого же оперативного изменения конфигураций на серверах приложений.

В текущей архитектуре для балансировки нагрузки используется HAProxy. А для динамического изменения конфигурации HAProxy при изменении в кластере PostgreSQL используется решение Consul-Template.

C.3. Consul-Template. Установка и настройка

1. Поместить поставляемые шаблоны конфигурационных файлов (см. ниже) и архив с приложением на хост, в папку **/tmp/consul**. И выполнить команды:

```
cd /tmp/consul
#curl -kO https://releases.hashicorp.com/consul-template/0.24.1/consul-↩
  template_0.24.1_linux_amd64.tgz
sudo -- sh -c 'unzip -u -d /usr/sbin consul-template_0.24.1_linux_amd64.tgz \
              && chmod ug+x /usr/sbin/consul \
              && rm -r /tmp/consul-template_0.24.1_linux_amd64.tgz'

sudo mkdir -p /etc/consul-template.d /var/lib/consul/templates
sudo cp consul-template.hcl /etc/consul-template.d/00-consul-template.hcl
sudo cp haproxy.hcl /etc/consul-template.d/
sudo cp haproxy.ctmpl /var/lib/consul/templates/
sudo chown -R consul.consul /etc/consul-template.d /var/lib/consul/templates
```

```
sudo -- sh -c 'cp consul-template.service /etc/systemd/system/ \
               && sudo systemctl daemon-reload
               && sudo systemctl enable consul-template'
```

C.4. HAProxy. Установка и конфигурирование

> Для разрешения проблемы ***HAProxy*** “**Cannot bind socket**” необходимо установить флаг ***SELinux***:
>
> ```
> setsebool -P haproxy_connect_any=1
> ```

Для установки HAProxy необходимо выполнить следующий перечень команд:

```
sudo yum -y haproxy
sudo setsebool -P haproxy_connect_any=1
sudo systemctl enable haproxy
sudo systemctl start consul-template haproxy
```

C.4.1. Шаблоны конфигурационных файлов

`/etc/consul-template.d/00-consul-template.hcl`

```
consul {
  address = "127.0.0.1:8500"
  token = "{{ consul_token }}"
  retry {
    enabled = true
    attempts = 12
    backoff = "250ms"
    max_backoff = "10s"
  }
}

reload_signal = "SIGHUP"

kill_signal = "SIGINT"

max_stale = "10m"

log_level = "warn"

# pid_file = "/run/consul-template.pid"

wait {
  min = "2s"
  max = "5s"
}
```

```
deduplicate {
  enabled = true
  prefix = "consul-template/dedup/"
}
```

/etc/consul-template.d/haproxy.hcl

```
template {
  source = "/var/lib/consul/templates/haproxy.ctmpl"
  destination = "/etc/haproxy/haproxy.cfg"
  command = "systemctl reload haproxy"
  command_timeout = "10s"
  error_on_missing_key = false
  backup = true
  wait {
    min = "2s"
    max = "10s"
  }
}
```

Документация по функциям и встроенным переменным для написания шаблонов Consul Template language.

`/var/lib/consul/templates/haproxy.ctmpl`

```
# Rendered by consul-template.service {{ timestamp }}

global
    daemon
    chroot  /var/lib/haproxy
    user    haproxy
    group   haproxy
    pidfile /var/run/haproxy.pid
    log     /dev/log local0
    maxconn 102400

defaults
    log     global
    mode    tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s

listen stats
    bind    *:2000
    maxconn 100
    mode    http
    option  httplog
    stats   uri /stats
```

```
    stats   enable
    stats   refresh 10s
    stats   admin if LOCALHOST

### Listener for PostgreSQL LEADER database
listen  db-rw
        bind       127.0.0.1:5432
        maxcon     10240
        timeout    queue 30s
        option     httpchk OPTIONS /master
        http-check expect status 200
        default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions {{range service "master.db-main"}}
        server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}

### Listener for PostgreSQL REPLICA database
listen  db-ro
        bind       127.0.0.1:5433
        maxconn    10240
        timeout    queue 30s
        option     httpchk OPTIONS /replica
        http-check expect status 200
        balance    roundrobin
        default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions {{range service "replica.db-main"}}
        server {{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}
```

Для использования шаблона в ansible (Jinja2 template) необходимо экранировать переменные consul-template с помощью конструкции `{%raw%} ... {%endraw%}`

```
# Rendered by consul-template.service {%raw%}{{ timestamp }}{%endraw%}

global
    daemon
    chroot  /var/lib/haproxy
    user    haproxy
    group   haproxy
    pidfile /var/run/haproxy.pid
    log     /dev/log local0
    maxconn 102400

defaults
    log     global
    mode    tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s

listen stats
    bind    *:2000
    maxconn 100
```

```
24      mode    http
25      option  httplog
26      stats   uri /stats
27      stats   enable
28      stats   refresh 10s
29      stats   admin if LOCALHOST

31  ### Listener for PostgreSQL LEADER database
32  listen  db-rw
33          bind       127.0.0.1:{{ db_rw_port}}
34          maxcon     10240
35          timeout    queue 30s
36          option     httpchk OPTIONS /master
37          http-check expect status 200
38          default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions {↩
    %raw%}{{range service "master.{%endraw%}{{ consul_service }}{%raw%}"}}{%endraw%}
39          server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port ↩
    8008{{end}}{%endraw%}

41  ### Listener for PostgreSQL REPLICA database
42  listen db-ro
43          bind       127.0.0.1:{{ db_ro_port}}
44          maxconn    10240
45          timeout    queue 30s
46          option     httpchk OPTIONS /replica
47          http-check expect status 200
48          balance    roundrobin
49          default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions {↩
    %raw%}{{range service "replica.{%endraw%}{{ consul_service }}{%raw%}"}}{%endraw%}
50          server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port ↩
    8008{{end}}{%endraw%}
```

`/etc/systemd/system/consul-template.service`

```
1  [Unit]
2  Description=Consul Template Service
3  Documentation=https://github.com/hashicorp/consul-template/
4  After=network-online.target
5  Wants=network-online.target

7  [Service]
8  Type=simple
9  #User=consul
10 #Group=consul
11 ExecStart=/usr/sbin/consul-template -config=/etc/consul-template.d/
12 ExecReload=/bin/kill -HUP $MAINPID
13 KillSignal=SIGINT
14 TimeoutStopSec=5
15 Restart=on-failure
16 SyslogIdentifier=consul

18 [Install]
19 WantedBy=multi-user.target
```

Приложение D. Настройка SSO

Конфигурация Web-сервера, поставляемая для нашего приложения, содержит отключенные (комментированные) директивы для подключения SSO-авторизации. Web-приложение поддерживает интеграцию с kerberos инфраструктурой и LDAP-каталогами, в том числе MS AD и FreeIPA.

D.1. Настройка конфигурации Web-сервера

В файле `/opt/luxmsbi/conf/nginx/nginx.conf` раскомментировать строку с подключением модуля `ngx_http_auth_spnego_module.so`:

```
---
 nginx.conf.old 2021-10-05 17:19:58.562466594 +0300+++
 nginx.conf 2021-10-05 17:19:49.449517250 +0300
@@ -7,7 +7,7 @@

 load_module /usr/lib64/nginx/modules/ngx_http_lua_module.so;
 #load_module /usr/lib64/nginx/modules/ngx_http_auth_spnego_module.so;-
#load_module /usr/lib64/nginx/modules/ngx_http_auth_spnego_module_debug.so;+
load_module /usr/lib64/nginx/modules/ngx_http_auth_spnego_module_debug.so;
```

Модуль собирается и тестируется нами, и доступен в публичном репозитарии Luxms BI RPM ThirdParty.

Необходимо переименовать/скопировать конфигурационный файл `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sso.location.off` в `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sso.location`:

```
mv /opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sso.location.off \
   /opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sso.location
```

Изменить значения в конфигурационном файле `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sso.location`:

```
# You should change next 3 lines according to your environment
# - put your KDC domain name
# - put keytab file full path.
#   Make sure the file has read permissions for user bi
# - put SPN name from keytab file without KDC domain suffix
auth_gss_realm                EXAMPLE.TLD;
auth_gss_keytab               /etc/bi.keytab;
auth_gss_service_name         "HTTP/bi.example.tld";
```

Вы можете посмотреть корректные значения для конфигурационного файла в вашем Kerberos-ключе, вызовите утилиту *klist*:

```
klist -ke /opt/luxmsbi/conf/nginx/bi.keytab
Keytab name: FILE:/opt/luxmsbi/conf/nginx/bi.keytab
KVNO Principal----
 ---------------------------------------------------------------------------
   3 HTTP/bi.example.tld@EXAMPLE.TLD (des-cbc-crc)
   3 HTTP/bi.example.tld@EXAMPLE.TLD(des-cbc-md5)
   3 HTTP/bi.example.tld@EXAMPLE.TLD (arcfour-hmac)
   3 HTTP/bi.example.tld@EXAMPLE.TLD (aes256-cts-hmac-sha1-96)
   3 HTTP/bi.example.tld@EXAMPLE.TLD (aes128-cts-hmac-sha1-96)
```

Сверьте настройки в файле конфигурации со значениями из *keytab*-файла.

D.1.1. Проверка работоспособности Web-сервера

До применения измененной конфигурации запустите команду проверки:

```
sudo nginx -c /opt/luxmsbi/conf/nginx/nginx.conf -t
```

Перезапустить сервис *luxmsbi-web* и проверить журналы на отсутствие ошибок после перезапуска:

```
sudo systemctl restart luxmsbi-web
sudo systemctl status luxmsbi-web -l
sudo journalctl -u luxmsbi-web
```

D.1.2. Проверка работы модуля SPNEGO

Если целевой пользователь с заведенной в домен машиной заходит в систему с включеным `отладочным модулем` веб-сервера nginx `ngx_http_auth_spnego_module_debug.so` при включенном логировании, то в лог-файле `/var/log/luxmsbi/nginx/luxmsbi-sso.debug.log` отображается информация следующего вида:

```
2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling IN: token.len=0, ↩
  head=0, ret=401
2022/05/20 18:05:26 [debug] 958#958: *78798 Begin auth
2022/05/20 18:05:26 [debug] 958#958: *78798 Detect SPNEGO token
2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling OUT: token.len=0, ↩
  head=1, ret=401
2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling IN: token.len=0, ↩
  head=0, ret=401
2022/05/20 18:05:26 [debug] 958#958: *78798 Begin auth
2022/05/20 18:05:26 [debug] 958#958: *78798 Detect SPNEGO token
2022/05/20 18:05:26 [debug] 958#958: *78798 Token decoded: YIII+wYGKwYB...
        [Тело Kerberos-ticket]
2022/05/20 18:05:26 [debug] 958#958: *78798 Client sent a reasonable Negotiate ↩
  header
```

```
2022/05/20 18:05:26 [debug] 958#958: *78798 GSSAPI authorizing
2022/05/20 18:05:26 [debug] 958#958: *78798 Use keytab /etc/bi.keytab
2022/05/20 18:05:26 [debug] 958#958: *78798 Using service principal: HTTP/hostname@MAIN.DOMAIN.LOCAL
2022/05/20 18:05:26 [debug] 958#958: *78798 my_gss_name HTTP/hostname@MAIN.DOMAIN.LOCAL
2022/05/20 18:05:26 [debug] 958#958: *78798 ngx_http_auth_spnego_set_bogus_authorization: bogus user set
2022/05/20 18:05:26 [debug] 958#958: *78798 user is USER_TEST@MAIN.DOMAIN.LOCAL
2022/05/20 18:05:26 [debug] 958#958: *78798 GSSAPI auth succeeded
2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling OUT: token.len=2303, head=1, ret=0
```

Данный лог информирует, что со стороны целевой пользовательской машины настройки правильные. Естественно, могут быть другие проблемы. Например, пользователь не в той группе или залогинился не под своим логином.

Если машина целевого пользователя не в домене, то будет приходить NTLM-ный токен (он намного меньше по длине) следующего вида:

```
2022/05/20 14:11:16 [debug] 958#958: *75240 SSO auth handling IN: token.len=0, head=0, ret=401
2022/05/20 14:11:16 [debug] 958#958: *75240 Begin auth
2022/05/20 14:11:16 [debug] 958#958: *75240 Detect SPNEGO token
2022/05/20 14:11:16 [debug] 958#958: *75240 Token decoded: TlRDFSDFSDGDDBAAAAlAAASDFDSFAAAAAAKALpHAAAADw==
2022/05/20 14:11:16 [debug] 958#958: *75240 Client sent a reasonable Negotiate header
2022/05/20 14:11:16 [debug] 958#958: *75240 GSSAPI authorizing
2022/05/20 14:11:16 [debug] 958#958: *75240 Use keytab /etc/bi.keytab
2022/05/20 14:11:16 [debug] 958#958: *75240 Using service principal: HTTP/hostname@MAIN.DOMAIN.LOCAL
2022/05/20 14:11:16 [debug] 958#958: *75240 my_gss_name HTTP/hostname@MAIN.DOMAIN.LOCAL
2022/05/20 14:11:16 [debug] 958#958: *75240 gss_accept_sec_context() failed: Unknown error:
2022/05/20 14:11:16 [debug] 958#958: *75240 GSSAPI failed
```

При подобных логах машина целевого пользователя настроена неправильно. И, скорее всего, машина не заведена в домен.

D.2. Интеграция с LDAP-каталогами

При необходимости настройки распределения прав в системе Luxms BI по членству в группах LDAP-каталога требуется установка компонента `luxmsbi-gateway`, предоставляющего API для проверки учетной записи и получения списка групп, в которой состоит пользователь.

Компонент **Luxms BI Gateway** использует конфигурационный файл `/opt/luxmsbi/conf/luxmsbi-gateway.yml`. Пример конфигурации:

```yaml
# general HTTP server configuration
# address:port that server will be listening and serving on
listen: 'localhost:8889'
# logs all incoming/outgoing HTTP requests/responses
http-trace: false
# can be: error, warn, info, debug (default: "info")
log-level: 'info'
# serve HTTPS requests instead of HTTP
use-tls: false
# path to *.crt/*.pem certificate file, ignored if use-tls: false
cert-file: './localhost.crt'
# path to *.key key file, ifnored if use-tls: false
key-file: './localhost.key'
# connections timeout, global setting for all connections (default: "30s")
timeout: '30s'


# Simple config for MS AD
ad:
    base: 'dc=example,dc=org'
    host: 'dc-01.example.org'
    port: 3268
    usessl: false
    binddn: 'bind-user@example.org'
    bindpw: 'BindPassword'
    # Specify LDAP attribute to provide as "login" at output JSON
    returnAsLogin: userPrincipalName
    # Filter groups to output JSON
    groupFilter:
      - groupname


# Simle config for OpenLDAP config:
ldap:
    base: 'dc=example,dc=com'
    host: 'dc-01.example.org'
    port: 636
    usessl: true
    skipSSLCertVerify: true
    binddn: 'uid=bind-user,dc=example,dc=org'
    bindpw: 'BindPassword'
    userSearchFilter: '(uid=%s)'
# Used for pure OpenLDAP to search user group by user DN on its member attribute
    groupSearchFilter: '(&(objectClass=posixgroup)(member=%s))'
    attributes:
      - cn
      - mail
# Specify LDAP attribute to provide as "login" at output JSON
    returnAsLogin: uid
# Filter groups to output JSON
    groupFilter:
      - groupname
# Provide filtered group DN's
```

```
    includeGroupDNs: true

# Simple config for Free IPA:
ldap:
    base: 'dc=example,dc=com'
    host: 'dc-01.example.org'
    port: 636
    usessl: true
    skipSSLCertVerify: true
    binddn: 'uid=ldap-sso,dc=example,dc=com'
    bindpw: 'BindPassword'
    userSearchFilter: '(uid=%s)'
    attributes:
      - mail
# Specify LDAP attribute to provide as "login" at output JSON
    returnAsLogin: krbPrincipalName
# Filter groups to output JSON
    groupFilter:
      - group-*
# Provide filtered group DN's
    includeGroupDNs: true


# Simple config for PDF converter
# html2pdf:
#    chromeDevToolsURI: "http://localhost:9222"
#    tmpFilesDirectory: "/tmp/"
#    orientation: "landscape"
#    printBackground: true
#    marginTop: 0.5
#    marginBottom: 0.5
#    marginLeft: 0.5
#    marginRight: 0.5
#    paperWidth: 8.5
#    paperHeight: 11.0
```

После установки компонента и настройки конфигурации `luxmsbi-gateway` необходимо выполнить следующие дополнительные действия.

Для RPM-based ОС настроить автоматический запуск и запустить сервис:

```
sudo systemctl enable luxmsbi-gateway --now
```

Для DEB-based ОС перезапустить сервис:

```
sudo systemctl restart luxmsbi-gateway
```

D.2.1. Проверка конфигурации Luxmsbi-gateway

При выполнении команды `journalctl -u luxmsbi-gateway` можем увидеть логирование компонента `luxmsbi-gateway`, где будет отображаться подробная информация:

```
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [INFO ] AD ↩
  userAndGroupsHandler for USER_TEST@MAIN.DOMAIN.LOCAL
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↩
  AuthenticateExtendedLong: Getting UPN for USER_TEST@MAIN.DOMAIN.LOCAL
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↩
  AuthenticateExtendedLong: Will bind as sys_USER@MAIN.DOMAIN.LOCAL, getting UPN
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↩
  AuthenticateExtendedLong: Connecting to AD
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↩
  AuthenticateExtendedLong: Bind as sys_USER@MAIN.DOMAIN.LOCAL with known pass
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↩
  AuthenticateExtendedLong: Bind Ok
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] Got User: ↩
  USER_TEST@MAIN.DOMAIN.LOCAL
May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] Got cn ↩
  Attr: Userov User Testovich
```

В случае, если в логах данные сообщения не появляются необходимо проверить настройку `loglevel: debug` в файле `/opt/luxmsbi/conf/luxmsbi-gateway.yml`.

D.3. Настройка пользовательских браузеров

Настройки SSO на стороне сервера не всегда гарантирует работу его на клиентских машинах.

D.3.1. Internet Explorer:

Для настройки SSO в IE, нужно зайти в настройки и добавить сайт в надежные узлы.

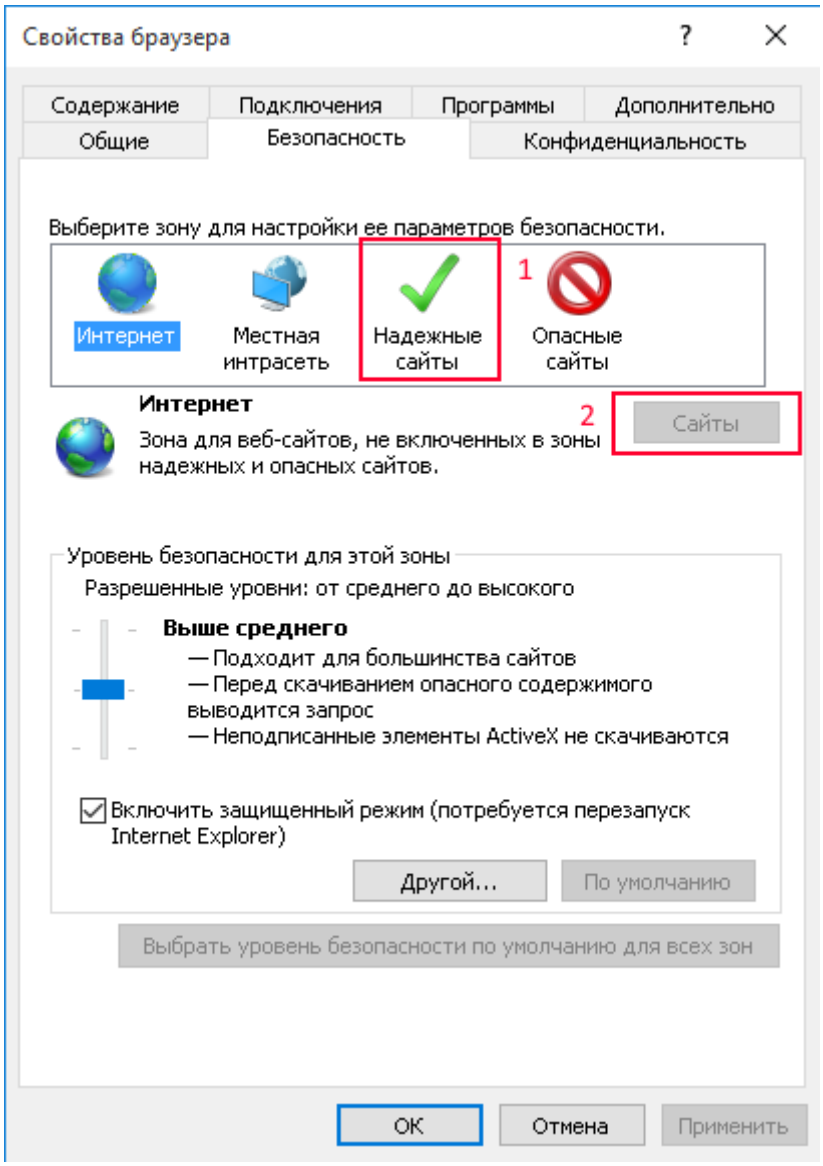


Рис. D.1. **1-sso-browser-settings.png**

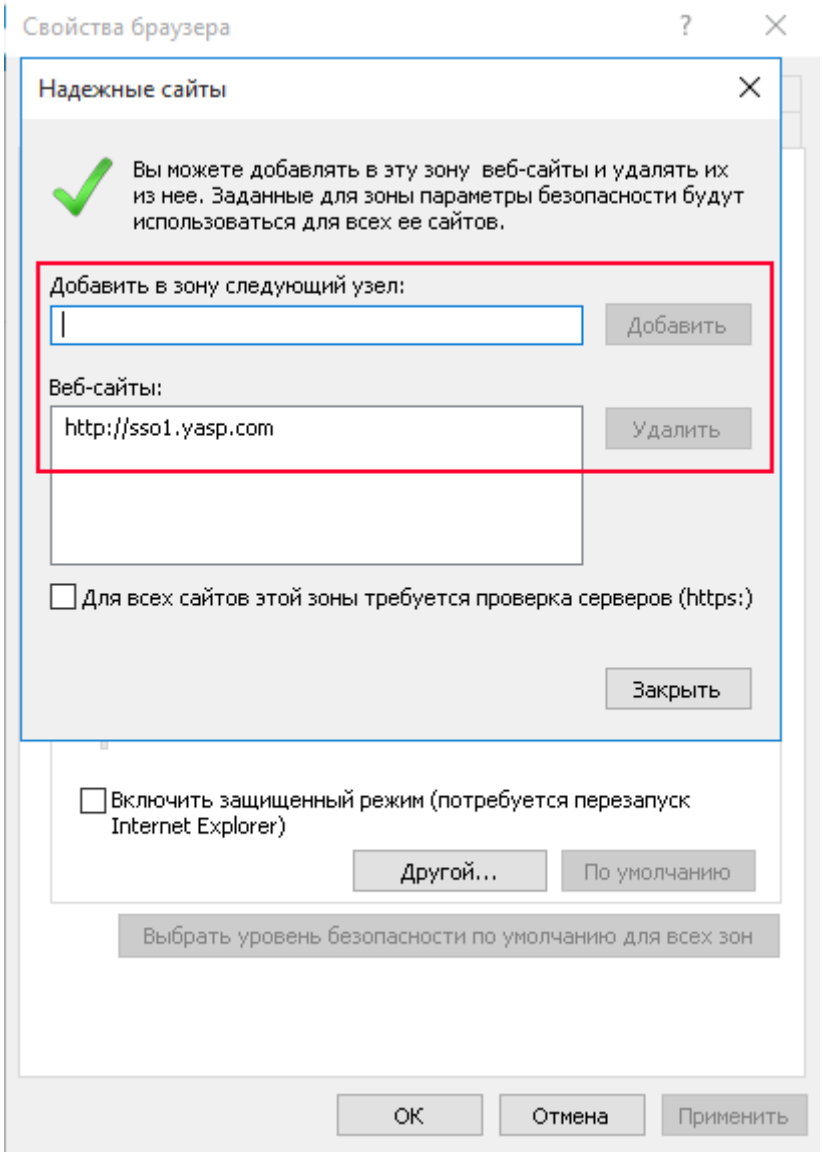


Рис. D.2. **2-sso-browser-settings.png**

Так же, в секции надежные узлы, нужно выбрать уровень безопасности (Другой) и выбрать проверку подлинности (Автоматический вход в сеть с текущим именем пользователя и паролем). Хочу обратить внимаение - это актуально, если сайт находится в локальной сети.

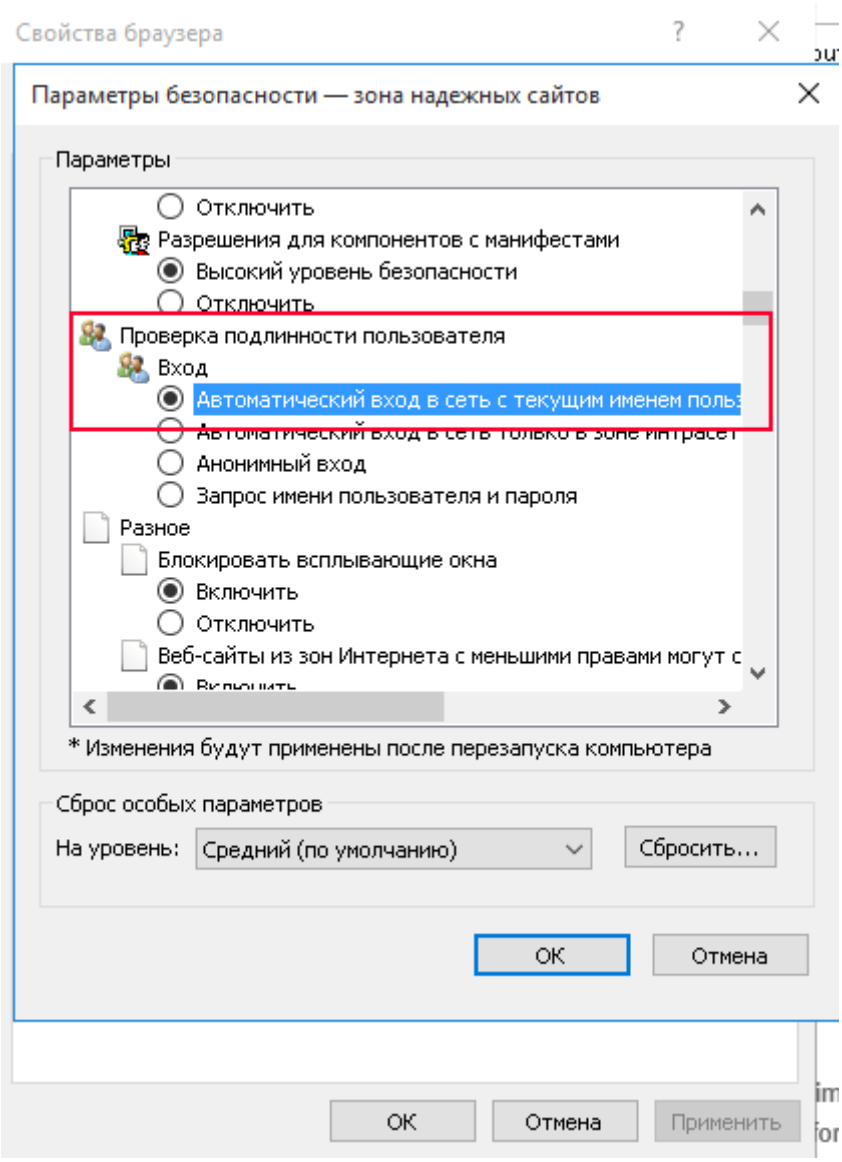


Рис. D.3. **3-sso-browser-settings.png**

D.3.2. Windows 10 EDGE:

Что бы тоже самое заработало и в Microsoft Edge browser. Нужно выбрать местную интрасеть и нажать кнопку (сайты).

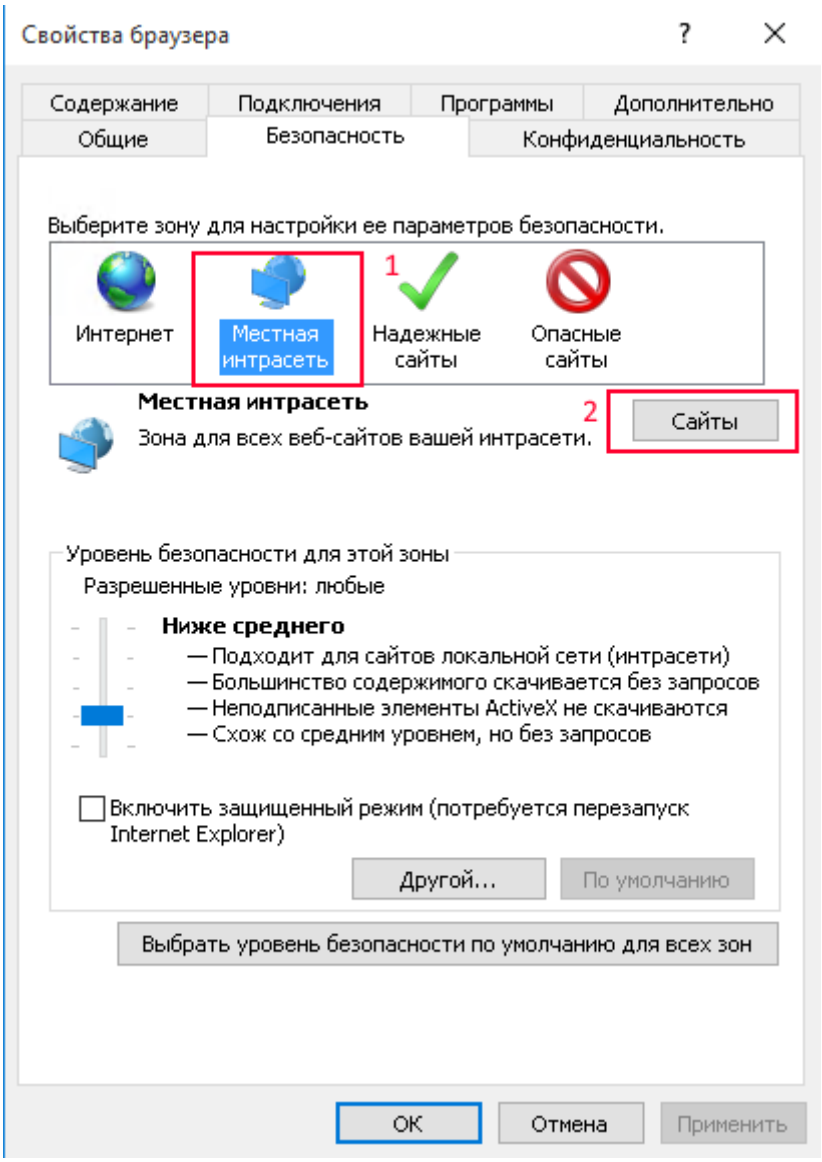


Рис. D.4. **4-sso-browser-settings.png**

Выбираем кнопку (дополнительно) и так же вставляем туда адрес нашего сайта.

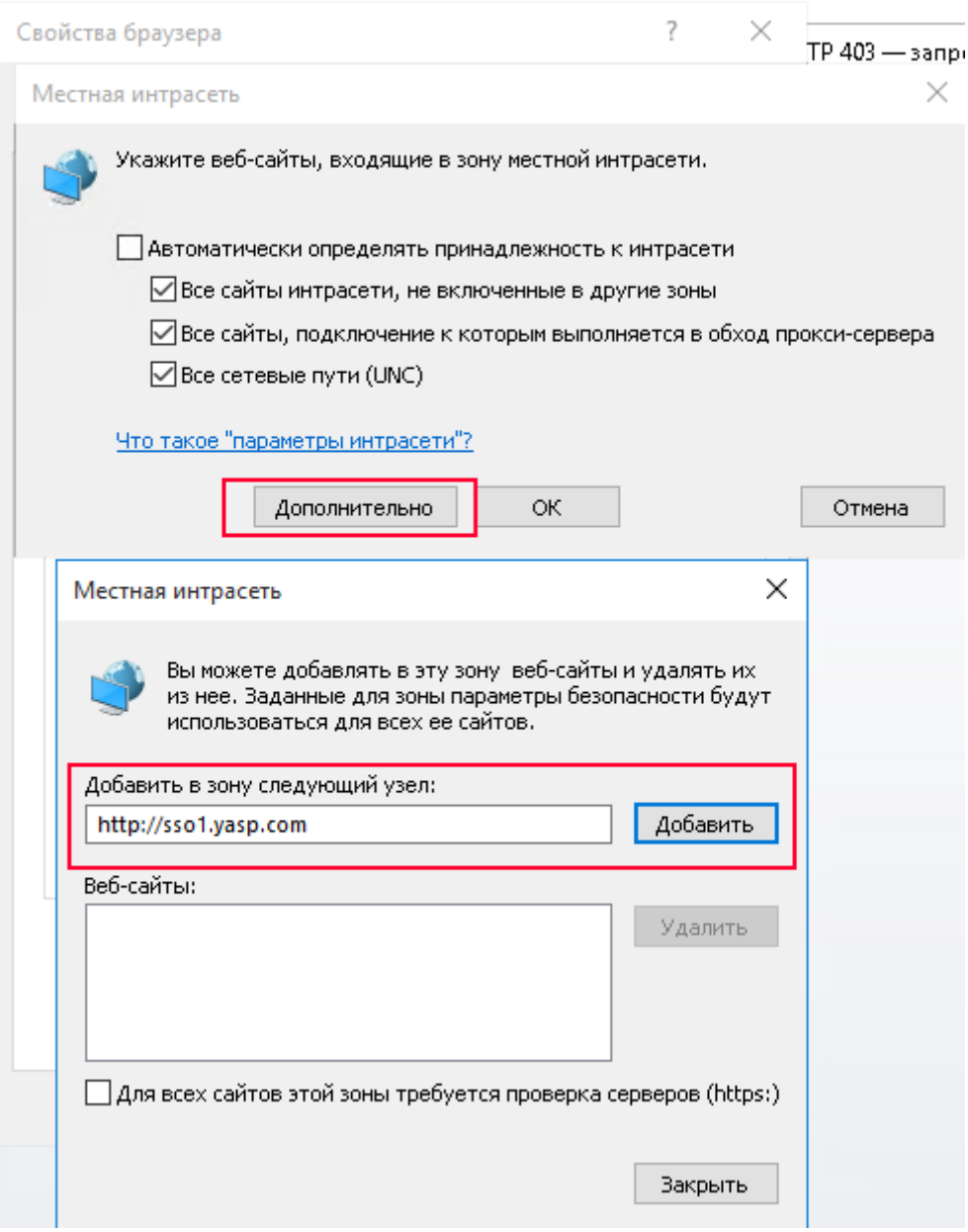


Рис. D.5. **5-sso-browser-settings.png**

Не забываем проверить в настройках броузера (Дополнительно - > Разрешить встроенную проверку подлинности Windows) это актуально для обоих броузеров IE & Edge.

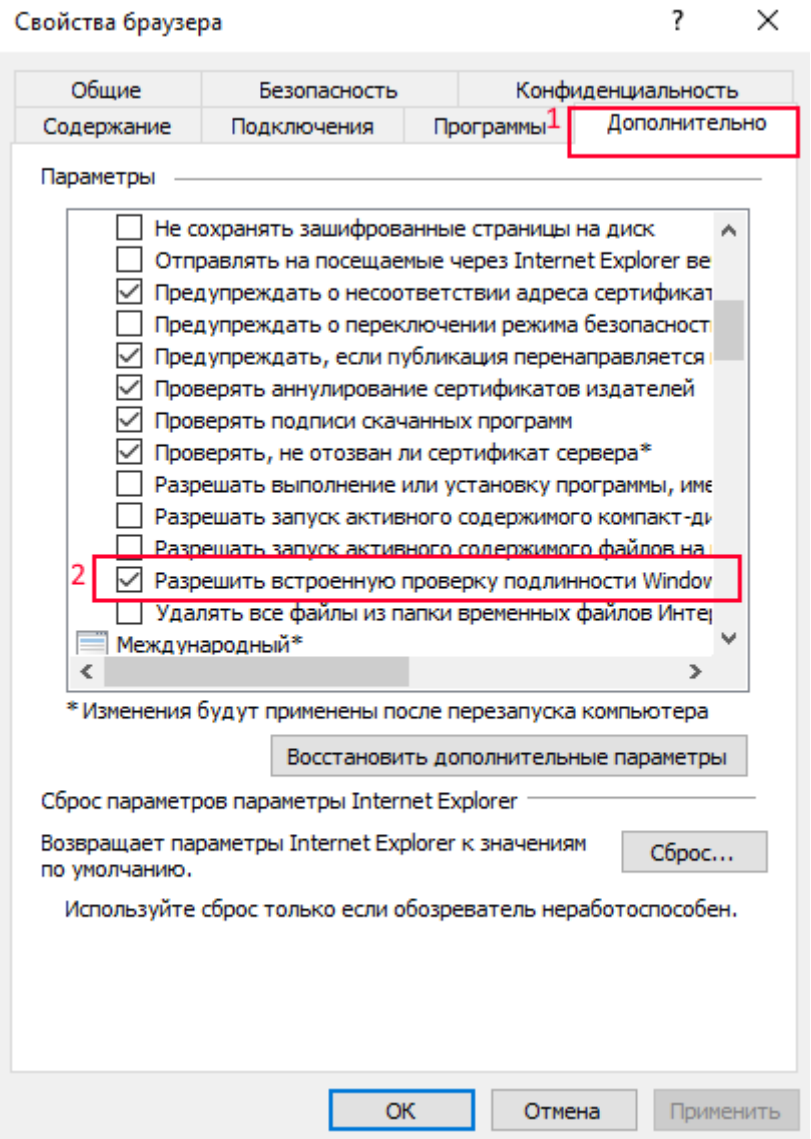


Рис. D.6. **7-sso-browser-settings.png**

D.3.3. Firefox

В строке броузера, нужно вписать about:config , согласиться (приняв риск и продолжив). Дальше нужно ввести (network.negotiate-auth.trusted-uris) далее ввести нужный сайт.

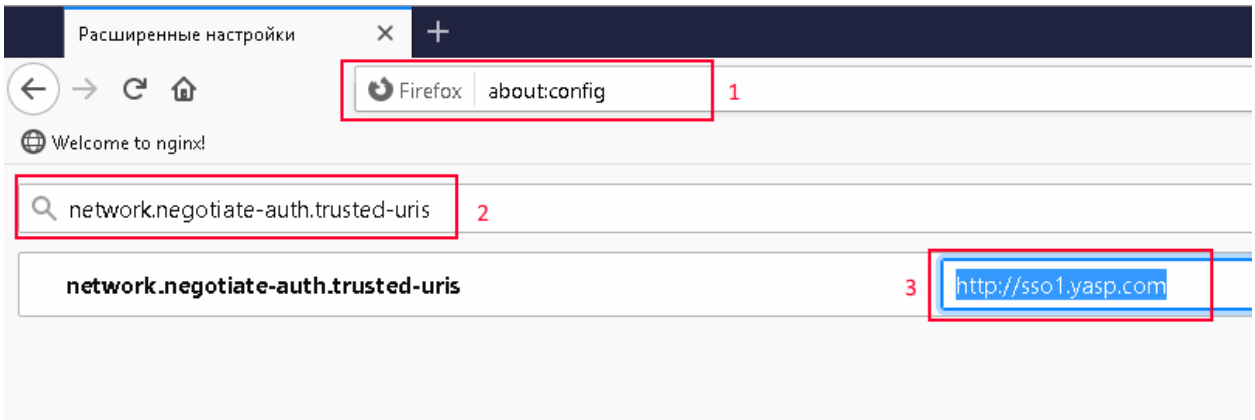


Рис. D.7. **6-sso-browser-settings.png**

D.3.4. Yandex & Chrome

Что касается этих броузеров, они берут свои настройки из IE, так что после настройки IE & Edge, остальные будут работать, как надо.

D.4. Генерация Kerberos-ключей

Описание процедуры организации Kerberos-аутентификации для Web(HTTP/HTTPS) приложений на ОС Linux при интеграции с каталогом MS AD.

D.4.1. Создание сервисной учетной записи

Создание учетной записи для обеспечения ***Kerberos*** - аутентификации необходимо помнить, что ***Kerberos(Secret Key)*** ключ генерируется на основе пароля учетной записи. Поэтому необходимо обеспечить неизменность пароля или перегенерацию ключей при его изменении.

В случае использования учетной записи рабочей станции(computer) изменение пароля производится автоматически *каждые 30 дней*, соответсвенно необходима настройка с такой же переодичностью механизма перегенерации ключей. В гетерогенных вычислительных сетях это требует регистрации серверов Linux в домене MS AD и использование скриптового решения по генерации ключей

Для горизонтально масштабированных решений более целесообразный вариант - использование пользовательской учетной записи с применением генерации пароля из набора случайных символов с длинной, обеспечивающей защиту от взлома методом перебора - например 16 символов, и запретом на изменние пароля учетной записи.

Исходные данные:

- Приложение доступно по адресу - http://www.example.org/application
- Домен MS AD - example.org
- Сервисная учетная запись - service-account

D.4.2. Регистрация Service Principal Name (SPN)

Формат имени сервисной учетной записи для Web-приложений, не зависимо от использования SSL-шифрования, имеет следующий вид:

> **HTTP/<*web-service*>**@**<REALM>**
> где:
>
> - web_service - URL, DNS-имя web-приложения
> - REALM - имя Kerberos REALM, обычно совпадает с именем домена MS AD, символами верхнего регистра
>
> например: HTTP/www.example.org@EXAMPLE.ORG

Регистрация сервисных учетных записей производится с использованием утилит командной строки в ОС Windows, под доменной учетной записью, обладающей полномочиями для изменения учетных записей, достаточно только предоставление прав на изменение конкретных учетных записей.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>setspn -A HTTP/www.example.org@EXAMPLE.ORG example\service-account
Checking domain DC=example,DC=org

Registering ServicePrincipalNames for CN=service-account,DC=example,DC=org
        HTTP/www.example.org@EXAMPLE.ORG
Updated object
```

Выполняется регистрация всех необходимых сервисных учетных записей. Возможно также регистрация сервисных учетных записей на короткие(NetBIOS) имена систем, IP-адреса.

D.4.3. Проверка сгенерированных SPN

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>setspn -L example\service-account
Registered ServicePrincipalNames for CN=service-account,DC=example,DC=org:
        HTTP/www.example.org@EXAMPLE.ORG
        HTTP/www@EXAMPLE.ORG


c:\>ldifde -d "CN=service-account,DC=example,DC=org" -l "userPrincipalName,↩
  servicePrincipalName,msDS-KeyVersionNumber" -f account.ldif
Connecting to "dc-01.example.org"
Logging in as current user using SSPI
Exporting directory to file account.ldif
Searching for entries...
Writing out entries.
```

```
1 entries exported

The command has completed successfully

c:\temp>type account.ldif
dn: CN=service-account,DC=example,DC=org
changetype: add
userPrincipalName: service-account@example.org
servicePrincipalName: HTTP/www.example.org@EXAMPLE.ORG
servicePrincipalName: HTTP/www@EXAMPLE.ORG
msDS-KeyVersionNumber: 2
```

D.4.4. Генерация ключей

При генерации ключей для нескольких SPN основной момент - сохранение KVNO(Key Value Number), изменение KVNO в процессе генерации ключей приведет к неработоспособности части ключей. Поэтому обратите внимание на ключ ***-setpass*** в вызове генерации ключей для второго и последующих SPN.

Для генерации последовательности случайных символов для пароля можно использовать утилиту командной строки openssl.

```
$ openssl rand -base64 20
jgDMj2KvZSqkEw2yWVxIVrfptGo=
```

Используем секретный пароль при генерации ключей

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

c:\temp>ktpass /mapuser example\service-account /princ ↩
  HTTP/www.example.org@EXAMPLE.ORG /ptype KRB5_NT_PRINCIPAL /pass ↩
  jgDMj2KvZSqkEw2yWVxIVrfptGo /crypto ALL /out key-1.keytab +answer
Targeting domain controller: dc-01.example.org
Successfully mapped HTTP/www.example.org to service-account.
Password successfully set!
Key created.
Key created.
Key created.
Key created.
Key created.
Output keytab to key-1.keytab:
Keytab version: 0x502
keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
  etype 0x1 (DES-CBC-CRC) keylength 8 (0x5e6befd37c4913ba)
keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
  etype 0x3 (DES-CBC-MD5) keylength 8 (0x5e6befd37c4913ba)
keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
  etype 0x17 (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
keysize 95 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
  etype 0x12 (AES256-SHA1) keylength 32 ↩
  (0xd520e8aed124bb5213cba436d3e9d6cd1d5ba54fdd5919e406aa185977dd121a)
```

```
19 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
     etype 0x11 (AES128-SHA1) keylength 16 (0xe14a7e0e13917c1165f5b8848f00c20f)

21 c:\temp>ktpass /mapuser example\service-account /princ HTTP/www@EXAMPLE.ORG ↩
     /ptype KRB5_NT_PRINCIPAL /pass jgDMj2KvZSqkEw2yWVxIVrfptGo -setpass /kvno 3 ↩
     /crypto ALL /in key-1.keytab /out http.keytab -setupn
22 Existing keytab:

24 Keytab version: 0x502
25 keysize 71 HTTP/www@EXAMPLE.ORGD ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x1 ↩
     (DES-CBC-CRC) keylength 8 (0x5e6befd37c4913ba)
26 keysize 71 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 ↩
     (DES-CBC-MD5) keylength 8 (0x5e6befd37c4913ba)
27 keysize 79 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x17 ↩
     (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
28 keysize 95 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 ↩
     (AES256-SHA1) keylength 32 ↩
     (0xd520e8aed124bb5213cba436d3e9d6cd1d5ba54fdd5919e406aa185977dd121a)
29 keysize 79 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x11 ↩
     (AES128-SHA1) keylength 16 (0xe14a7e0e13917c1165f5b8848f00c20f)
30 Targeting domain controller: gvc-dc-02.gvc.oao.rzd
31 Successfully mapped HTTP/www to service-account.
32 Key created.
33 Key created.
34 Key created.
35 Key created.
36 Key created.
37 Output keytab to http.keytab:
38 Keytab version: 0x502
39 keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
     etype 0x1 (DES-CBC-CRC) keylength 8 (0x5e6befd37c4913ba)
40 keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
     etype 0x3 (DES-CBC-MD5) keylength 8 (0x5e6befd37c4913ba)
41 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
     etype 0x17 (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
42 keysize 95 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
     etype 0x12 (AES256-SHA1) keylength 32 ↩
     (0xd520e8aed124bb5213cba436d3e9d6cd1d5ba54fdd5919e406aa185977dd121a)
43 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 ↩
     etype 0x11 (AES128-SHA1) keylength 16 (0xe14a7e0e13917c1165f5b8848f00c20f)
44 keysize 71 HTTP/www@EXAMPLE.ORGD ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x1 ↩
     (DES-CBC-CRC) keylength 8 (0x25a8e3403d4a342c)
45 keysize 71 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 ↩
     (DES-CBC-MD5) keylength 8 (0x25a8e3403d4a342c)
46 keysize 79 HTTP/www@EXAMPLE.ORGD ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x17 ↩
     (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
47 keysize 95 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 ↩
     (AES256-SHA1) keylength 32 ↩
     (0x452317c637a6ef2b236ef9e9232d03dc3ae95a85f5132e3274710a5cb9c0c9c4)
48 keysize 79 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x11 ↩
     (AES128-SHA1) keylength 16 (0x0917d8ff99a2b30e381a3e53ce0b5420)
```

По результатам генерации сохраняем `http.keytab` и можем также сохранить пароль - пароль можно использовать для проверки работоспособности сервисной учетной записи. Если

пароль случайно утерян, это не повлияет на работоспособность Kerberos-аутентификации.

D.4.5. Установка и проверка работоспособности

Установка файла `http.keytab`, классически производится в папку `/etc`, но в нашем случае установка возможна и в локальные папки приложения. Главное в обоих случаях дать разрешения файловой системы, достаточные для чтения файла владельцу процесса `NGinx`.

D.4.6. Настройка NGinx

Настройка ОС

```
includedir /etc/krb5.conf.d/

[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_keytab_name=bi5.keytab
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
 default_realm = EXAMPLE.ORG
 default_ccache_name = KEYRING:persistent:%{uid}

[realms]
 YASP.COM = {
  kdc = dc-01.example.org
  admin_server = dc-01.example.org
 }

[domain_realm]
 .example.org = EXAMPLE.ORG
 example.org = EXAMPLE.ORG
```

Установка файла http.keytab, классически производится в директрию /etc, но в нашем случае установка возможна и в локальные папки приложения. Главное в обоих случаях дать разрешения файловой системы, достаточные для чтения файла владельцу процесса ***NGinx***.

```
    auth_gss on;
    auth_gss_realm EXAMPLE.ORG;
    auth_gss_keytab /etc/http.keytab;
    auth_gss_service_name HTTP/www.example.org;
    auth_gss_format_full on;
```

```
    auth_gss_allow_basic_fallback off;
    proxy_set_header Authorization "";
    proxy_set_header X-Forwarded-User $remote_user;
```

D.5. Настройка прав в приложении Luxms BI

Настройка распределения прав выполняется прикладным Администратором приложения и не входит в область системного администрирования.

Приложение E. Настройка SSL

Настройка шифрования трафика между Пользователем и Luxms BI позволяет обеспечить защиту передаваемых данных, защиту Web приложения и безопасность использования API.

Наша основная рекомендация - при использовании Luxms BI как внутреннего корпоративного ИТ-сервиса, в доверенных(защищенных) сетях - нет необходимости настраивать максимально возможный уровень защищенности. Излишнее увеличение уровня защищенности в любом случае увеличит нагрузку на вычислительные ресурсы. Что может привести к необходимости их масштабирования. При этом вероятность угроз, которые будут компенсированы в доверенных сетях имеют очень низкое значение. Вы просто потратите ресурсы на то, что вообще никогда не случится или может быть компенсировано другими методами.

> Для высоконагруженных инсталляций Luxms BI подключение HTTPS на Web-серверах Luxms BI приводит к дополнительной нагрузке на CPU. Если это является существенной проблемой, рекомендуем подключение и использование аппаратных SSL ускорителей.
> Также рекомендуем терминировать SSL трафик на аппаратных балансировщиках нагрузки и использовать обычный HTTP между балансировщиками и Web-серверами Luxms BI во внутренних(доверенных) сетях.

Мы предоставляем минимальную конфигурацию, которая достаточна для обеспечения безопасности. Но Клиент может добавить в нее настройки, позволяющие соответсвовать требуемому уровню защиты в соответствии с ВНД.

E.1. Настройка конфигурации

Конфигурация Web-сервера, поставляемая для нашего приложения, содержит отключенные (комментированные) директивы для подключения SSL-шифрования сессий пользователя.

В файле `/opt/luxmsbi/conf/nginx/conf.d/entrypoint.conf` раскомментировать строку с подключением конфигурационного файла `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl`:

```
---
 entrypoint.conf.old 2021-10-08 17:36:21.173807998 +0300+++
 entrypoint.conf 2021-10-08 17:36:43.564734068 +0300
@@ -3,7 +3,7 @@
     listen       80;

     # Uncomment next line to allow SSL-
    #include /opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl+
```

```
    include /opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl

 #    access_log   /var/log/luxmsbi/nginx/luxmsbi.access.log with_timing;
     error_log    /var/log/luxmsbi/nginx/luxmsbi.errors.log;
```

Содержимое конфигурационного файла `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl` тоже подлежит корректировке:

```
    listen       443 ssl;

    ssl_certificate              /opt/luxmsbi/conf/ssl/host-full.cer;
    ssl_certificate_key          /opt/luxmsbi/conf/ssl/host.key;
    ssl_session_timeout          5m;
    ssl_protocols                TLSv1.1 TLSv1.2;
    ssl_ciphers                  'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';
    ssl_prefer_server_ciphers    on;
    ssl_session_cache            shared:SSL:10m;
    add_header                   Strict-Transport-Security "max-age=31536000; includeSubdomains;";
```

Необходимо откорректировать путь и имена файлов в параметрах конфигурации:

- `ssl_certificate`
- `ssl_certificate_key`

В конфигурационных файлах Nginx не забывайте писать символ `;` в конце строки при корректировке имени и пути к файлам.

Конфигурационный файл предполагает хранение сертификатов по пути `/opt/luxmsbi/conf/ssl/`. Убедитесь, что эти файлы расположены там, где они должны быть, и имеют правильные разрешения доступа и корректного владельца:

```
ls -la /opt/luxmsbi/conf/ssl

chown -R bi.bi /opt/luxmsbi/conf/ssl
chmod 640 /opt/luxmsbi/conf/ssl/*
```

E.2. Проверка работоспособности

До применения измененной конфигурации запустите команду проверки:

```
sudo nginx -c /opt/luxmsbi/conf/nginx/nginx.conf -t
```

Перезапустить сервис *luxmsbi-web* и проверить журналы на отсутствие ошибок после перезапуска:

```
sudo systemctl restart luxmsbi-web
sudo systemctl status luxmsbi-web -l
```

```
sudo journalctl -u luxmsbi-web
```

Приложение F. Развертывание и настройка NATS

NATS - это совокупность продуктов с открытым кодом предоставляющий для ИТ инфраструктуры функционал:

- по хранению данных в формате ключ:значение (key:value)
- по хранению больших объектов (в том числе файлов)
- по предоставлению решений для модели издатель:подписчик (pub/sub)

NATS включает в себя функционал масштабирования, позволяющий организовать распределенное хранение и гарантированную доступность данных.

F.1. Планирование

Для обеспечения отказоустойчивости необходимо построение инфраструктуры из серверов NATS на нечетном количестве узлов. Минимальное количество узлов для отказоустойчивого решения - 3. NATS Server может работать и в режиме одного инстанса, но мы не рекомендуем такое решение для продуктовых сред.

Хранение данных в NATS может потребовать достаточно много дискогово пространства. Необходимо заранее спланировать инфраструктуру, чтобы переполнение разделов не привело к деградации остальных сервисов. Правильнее всего, выделить под nats отдельный раздел и монтировать его в /opt/nats. В этом случае переполнение никак не повлияет на работу системы.

Хранение больших объектов(файлов) включает в себя хранение метаданных, поэтому максимальный размер хранимых файлов меньше предоставленного раздела. Кроме того, файловая система используется для хранения данных `ключ:значение` и очередей сообщений (pub/sub).

Определение размера раздела зависимсит от действующей в Вашей компании политики глубины хранения оперативных данных (отчетов). Мы рекомендуем начинать с размера в 10 ГБ.

F.1.1. Типовая схема

Ниже приведена типовая схема с указанием стартовых параметров по размерам файловой системы. Реальные размеры файловой системы зависят от функционала инсталяции.

Совмещение компонентов NATS с другими компонентами системы Luxms BI возможно. Но мы рекомендуем использовать выделенные хосты для развертывания.

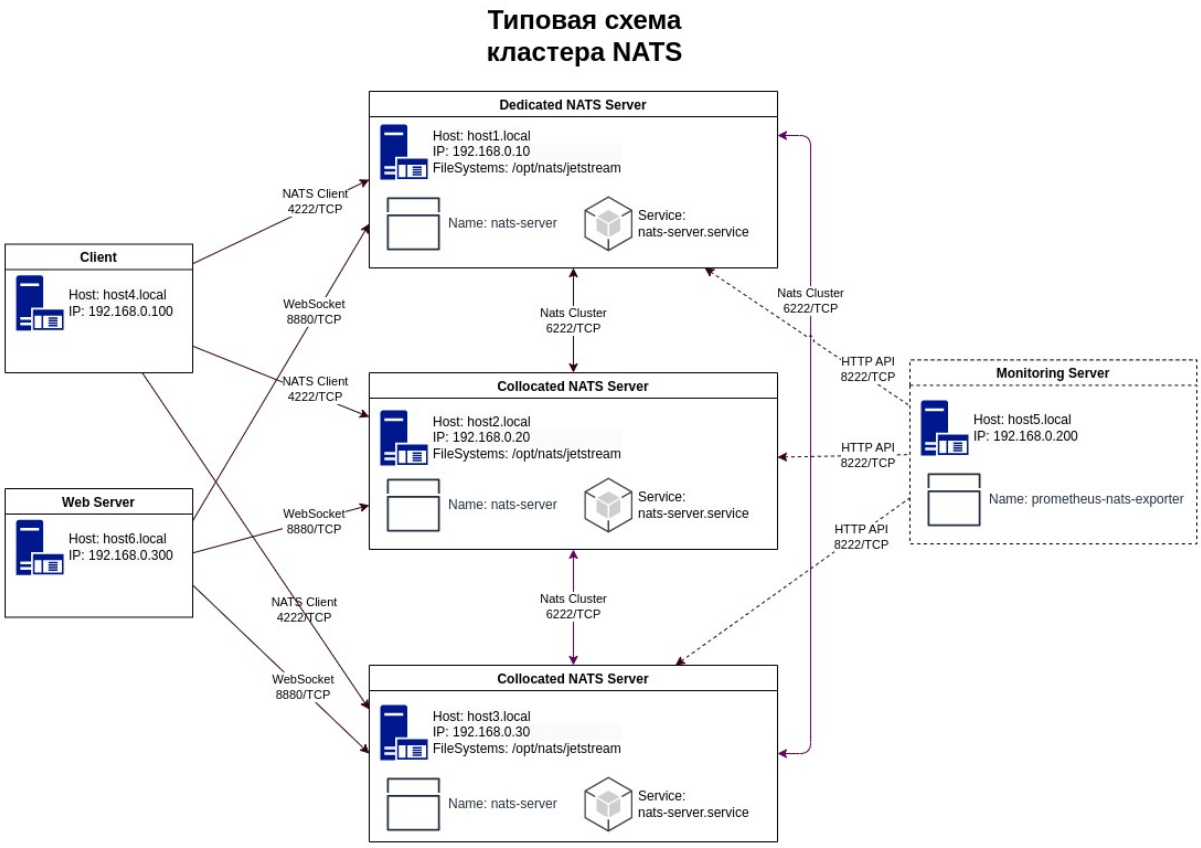


Рис. F.1. **nats-cluster-01.jpg**

F.2. Установка и настройка

Установка NATS производится с помощью пакета, который мы собираем из исходного кода проекта. Пакет содержит единый исполняемый бинарник и конфигурационные файлы, сервис, необходимый для развертывания распределенного хранилища.

Для ОС основаных на DEB-пакетах необходимо выполнить команды:

```
apt -y install nats-server
```

Для ОС основаных на RPM-пакетах необходимо выполнить команду:

```
dnf -y install nats-server
```

Проверяем, что nats-server может быть запущен:

```
# nats-server -v
nats-server: v2.9.11
```

При установке единственного экземпляра NATS Server-а дополнительной настройки конфигурации не требуется. Сервис работоспособен сразу после его запуска.

При развертывании кластерного решения необходима дополнительная корректировка конфигурационного файла.

F.2.1. Настройка кластера

Конфигурационный файл, поставляемый пакетом, содержит закомментированный блок параметров `cluster`. Необходимо удалить символы комментирования и указать в параметре `routes` правильный список серверов кластера, за исключением адреса хоста, на котором выполняется настройка:

На всех серверах необходимо отредактировать конфигурационный файл `/opt/nats/nats-server.conf`:

```
# Example config file

server_name: host1.local
port: 4222
#monitor_port: 8222

accounts: {
    SYS: {
        users: [
            { user: "x", pass: "y" }
        ]
    },
}

system_account: SYS

#cluster {
#  name: nats-cluster
#  port: 6222
  # Routes are actively solicited and connected to from this server.
#  routes: [
#    nats-route://host2.local:6222
#    nats-route://host3.local:6222
#  ]
#}

max_payload: 16MB

jetstream: enabled

jetstream {
    store_dir: /opt/nats
```

```
    max_mem: 1G
}
```

Изменяем значение параметра `server_name` на более осмысленное, рекомендуем использовать FQDN хоста. Обязательно, на всех хостах должно быть уникальное значение для `server_name`, в противном случае, кластер не запустится корректно.

> Для выполнения команд с административными привилегиями необходимо авторизованное подключение к контексту SYS. Необходимо поменять имя подключения и пароль после установки.

F.2.2. Настройка фильтра сетевых соединений

При использовании фаерволов необходимо обеспечить сетевую доступность между хостами с установленным NATS Server и компонентами Luxms BI.

1. При использовании FirewallD необходимо выполнить следующие команды.

- Для обеспечения сетевой доступности сервисов NATS, исключив команду с собственным адресом хоста.

а) На первом сервере NATS (192.168.0.10):

```
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.20/32 port port=6222 protocol=tcp accept'
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.30/32 port port=6222 protocol=tcp accept'
sudo firewall-cmd --runtime-to-permanent
```

б) На втором сервере NATS (192.168.0.20):

```
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.10/32 port port=6222 protocol=tcp accept'
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.30/32 port port=6222 protocol=tcp accept'
sudo firewall-cmd --runtime-to-permanent
```

в) На третьем сервере NATS (192.168.0.30):

```
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.10/32 port port=6222 protocol=tcp accept'
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.20/32 port port=6222 protocol=tcp accept'
sudo firewall-cmd --runtime-to-permanent
```

- Для обеспечения доступа Java-компонентов Luxms BI

На всех серверах NATS:

```
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.100/32 port port=4222 protocol=tcp accept'
sudo firewall-cmd --add-rich-rule='rule family=ipv4 source address=↩
  192.168.0.200/32 port port=4222 protocol=tcp accept'
sudo firewall-cmd --runtime-to-permanent
```

2. При использовании UFW необходимо выполнить следующие команды,

   - Для обеспечения сетевой доступности сервисов NATS, исключив команду с собственным адресом хоста.

а) На первом сервере NATS (192.168.0.10):

```
sudo ufw allow proto tcp from 192.168.0.20 to any port 6222
sudo ufw allow proto tcp from 192.168.0.30 to any port 6222
```

б) На втором сервере NATS (192.168.0.20):

```
sudo ufw allow proto tcp from 192.168.0.10 to any port 6222
sudo ufw allow proto tcp from 192.168.0.30 to any port 6222
```

в) На третьем сервере NATS (192.168.0.30):

```
sudo ufw allow proto tcp from 192.168.0.10 to any port 6222
sudo ufw allow proto tcp from 192.168.0.20 to any port 6222
```

   - Для обеспечения доступа Java-компонентов Luxms BI

На всех серверах NATS:

```
sudo ufw allow proto tcp from 192.168.0.100/32 to any port 4222
sudo ufw allow proto tcp from 192.168.0.200/32 to any port 4222
```

3. При использовании IPTables необходимо выполнить следующие команды,

   - Для обеспечения сетевой доступности сервисов NATS, исключив команду с собственным адресом хоста.

а) На первом сервере NATS (192.168.0.10):

```
sudo iptables -I INPUT -p tcp -m tcp \
                       -s 192.168.0.20,192.168.0.30 \
                       --dport 6222 \
                       -j ACCEPT
```

б) На втором сервере NATS (192.168.0.20):

```
sudo iptables -I INPUT -p tcp -m tcp \
                       -s 192.168.0.10,192.168.0.30 \
```

```
                        --dport 6222 \
                        -j ACCEPT
```

в) На третьем сервере NATS (192.168.0.30):

```
sudo iptables -I INPUT -p tcp -m tcp \
                        -s 192.168.0.10,192.168.0.20 \
                        --dport 6222 \
                        -j ACCEPT
```

- Для обеспечения доступа Java-компонентов Luxms BI

На всех серверах NATS:

```
sudo iptables -I INPUT -p tcp -m tcp \
                        -s 192.168.0.100,192.168.0.200 \
                        --dport 4222
                        -j ACCEPT
```

Не забываем сохранить правила с помощью iptables-save. В зависимости от системы и ее версии, варианты использования последней различаются. Подробнее читайте в документации к операционной системе.

F.2.3. Запуск сервисов

Для активации автоматического запуска NATS сервера после утановки и корректировки конфигурационного файла необходимо выполнить команду:

```
systemctl enable nats-server --now
```

F.2.4. Проверка работоспособности кластера

После запуска сервера можно проверить его работоспособность в журнале сервиса, мы должны увидеть что-то на подобие:

```
# journalctl -u nats-server

 Cluster name is test-nats
 Listening for route connections on 0.0.0.0:6222
 192.168.0.20:6222 - rid:4 - Route connection created
```

Для проверки работоспособности фунционала pub/sub можно на одном сервере запустить подписку из командной строки:

```
nats sub -s 127.0.0.1 "test-subject"
```

А на другом сервере отправить сообщение:

```
nats pub -s 127.0.0.1 "test-subject" "Test message"
```

В итоге, сообщение должно быть получено на первом сервере.

F.3. Встроенный мониторинг

Для включения мониторинга в кофигурационном файле необходимо снять комментарий с параметра `monitor_port` и перезапустить сервис.

По адресу http://(адрес сервера nats):8222 можно получить список метрик в формате json. Также данные метрики могут быть переданы в Prometheus с помощью экспортера prometheus-nats-exporter и визуализированы с помощью Grafana.

Подробнее настройка мониторинга изложена в официальной документации продукта:

- Включение мониторинга в NATS Server
- prometheus-nats-exporter
- NATS Server Dashboard

F.4. Полезные команды

Для получения информации по статусу и состоянию сервера NATS можно использовать CLI-интерфейс, выполнив следующие команды

F.4.1. Server Info

```
# nats --user=x --password=y server info
Server information for NDD2QZ3ERNLOVTRH2KA7OFM5Y46NX4EHHTNWD2PUALOOQ4TPOQWO4WU3

Process Details:

         Version: 2.9.11
      Git Commit:
      Go Version: go1.19.4
      Start Time: 2023-04-13 11:52:41.605030061 +0000 UTC
          Uptime: 13d20h28m53s

Connection Details:

   Auth Required: true
    TLS Required: false
            Host: 0.0.0.0:4222
     Client URLs:
```

```
JetStream:

              Domain:
   Storage Directory: /tmp/jetstream
          Max Memory: 954 MiB
            Max File: 63 GiB
      Active Acconts: 1
       Memory In Use: 0 B
         File In Use: 29 MiB
        API Requests: 2,636
          API Errors: 40

Limits:

        Max Conn: 65536
        Max Subs: 0
     Max Payload: 1.0 MiB
     TLS Timeout: 2s
  Write Deadline: 10s

Statistics:

       CPU Cores: 4 0.00%
          Memory: 26 MiB
     Connections: 3
   Subscriptions: 177
            Msgs: 58,403,928 in 114,840,129 out
           Bytes: 26 GiB in 52 GiB out
  Slow Consumers: 1
```

F.4.2. Server Ping

Для проверки доступности и работоспособности сервера:

```
# nats --user=x --password=y server ping
NDD2QZ3ERNLOVTRH2KA7OFM5Y46NX4EHHTNWD2PUALOOQ4TPOQWO4WU3     rtt=752.387µs----

 ping statistics ----
1 replies max: 0.00 min: 0.00 avg: 0.00
```

Приложение G. Руководство по миграции на Postgres 13

В рамках данной инструкции будет рассмотрен пример миграции с СУБД PostgreSQL 11 на PostgreSQL 13 с переносом данных. Также мы разберем несколько разных сценариев: 1. Обновление PostgreSQL на отдельном сервере. 1. Обновление PostgreSQL на кластере. 1. Обновление PostgreSQL с переходом на новую операционную систему.

Начнем с предварительной подготовки к миграции.

G.1. Подготовка к миграции

Прежде чем перейти к обновлению СУБД, выполним предварительные действия. Подразумевается, что мы будем работать с базой **mi**.

В зависимости от того, работает ли мы с кластером PostgreSQL или одним сервером СУБД, наши действия будут немного различаться. Соответствующие нюансы будут описаны.

G.1.1. Отключение активных соединений с базой данных

Необходимо убедиться, что с базой данных нет активных соединений. Для начала, остановим службы:

```
systemctl stop luxmsbi-datagate luxmsbi-importer luxmsbi-web luxmsbi-appserver
```

* обратите внимание, что данные службы могут быть запущены на разных серверах кластера LuxmsBI.

Входим в командную оболочку SQL (для кластера СУБД, только на мастере):

```
su - postgres
```

Посмотреть активные подключения можно командой:

```
SELECT * FROM pg_stat_activity;
```

Необходимо добиться, чтобы их не было. Для этого можно использовать команду:

```
SELECT pg_terminate_backend(pid) FROM pg_stat_activity WHERE datname = 'mi';
```

G.1.2. Создание резервной копии

Операция по обновлению PostgreSQL, потенциально, опасна. Поэтому стоит позаботиться о создании резервной копии.

Если мы работает на виртуальной машине, можно создать снапшот. Только стоит иметь ввиду, что снапшоты плохо влияют на обслуживание виртуальной машины. Рекомендуется удалить его после успешного выполнения обновления PostgreSQL.

Помимо этого, рекомендуется создать дамп базы данных.

> Важно отметить, что резервная копия, созданная с помощью утилиты `pg_basebackup` или на ее базе не позволяет восстанавливать данные на другую версию PostgreSQL. Если мы снимим дамп с ее помощью, то восстановление нужно выполнять на сервере с той же версией СУБД.

Так как он может занять много места на диске, убедитесь в наличие свободного пространства на носителе.

Посмотреть список баз и их размер можно sql-командой:

```
=
# \l+
```

Для самого резервного копирования PostgreSQL могут использоваться профессиональные средства или штатная утилита **pg_dump/pg_basebackup**.

Выбор конкретного инструмента должен соответствовать внутренним нормативным документам вашей организации.

G.1.3. Получение списка расширений

Заранее посмотрим список расширений, которые мы используем в текущем PostgreSQL (это делается из консоли psql):

```
su - postgres -c "psql mi"
```

```
=
# \dx
```

Мы можем увидеть что-то на подобие:

```
     Name      | Version |   Schema   |             Description--------------+---↩
  ------+------------+-------------------------------------

 btree_gin    | 1.3     | public     | support for indexing common datatypes
 btree_gist   | 1.5     | public     | support for indexing common datatypes
```

В данной таблице представлен список установленных расширений postgresql. Вам нужно будет установить те же расширения для новой версии СУБД, поэтому фиксируем список.

После окончания работы, выходим из оболочки psql:

```
=
# quit
```

G.1.4. Обновление Luxmsbi-pg

Перед обновлением СУБД, необходимо обновить пакет luxmsbi-pg до последней версии.

Выполним команды (если мы работает на кластере PostgreSQL, на всех нодах СУБД):

```
yum makecache
```

```
yum update luxmsbi-pg
```

И обновим базу (если мы работает на кластере PostgreSQL,только на мастере):

```
su - postgres -c "/usr/share/luxmsbi-pg/luxmsbi-setupdb.sh --upgrade"
```

При обновлении пакета с версии 8.9.3 на 8.9.4 мы получим ошибку:

```
schema "ag_catalog" does not exist
```

Проблема появляется из-за удаления поддержки расширения age в версии 8.9.3. Но в самой СУБД расширение продолжает перехватывать sql-запросы.

Необходимо удалить настройку shared_preload_libraries из конфигурации postgres и перезагрузить СУБД:

```
su - postgres -c 'psql -c "ALTER system reset shared_preload_libraries;"'
```

```
su - postgres -c 'psql -c "ALTER USER bi reset shared_preload_libraries;"'
```

```
systemctl restart postgresql-11
```

и снова выполнить обновление:

```
su - postgres -c "/usr/share/luxmsbi-pg/luxmsbi-setupdb.sh --upgrade"
```

G.2. Обновление PostgreSQL на CentOS (один сервер)

В нашей инструкции мы рассмотрим пример обновления СУБД PostgreSQL с версии 11 на версию 13. В качестве рабочей операционной системы будет использоваться CentOS 7.

Процедура обновления состоит из нескольких шагов:

1. Установка и запуск PostgreSQL новой версии (она будет работать параллельно со старой).

2. Запуск pg_upgrade для проверки возможности обновления.
3. Запуск pg_upgrade для выполнения обновления.
4. Проверка работоспособности СУБД.
5. Настройка новой версии в качестве основного экземпляра сервера баз данных.

Предполагается, что у нас уже установлена одна СУБД, которую мы и будем обновлять.

G.2.1. Установка и запуск PostgreSQL 13

В нашей инструкции мы планируем обновление до версии 13. Установим нужный нам пакет.

Для этого необходимо установить репозиторий.

Так как в нашей системе уже установлен PostgreSQL, скорее всего, репозиторий уже настроен, но мы все же, рассмотрим его установку.

Вводим команду:

```
yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-↩
  x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

* в данном примере мы установим репозиторий для CentOS 7 с архитектурой x86_64 (EL-7-x86_64).

Если мы получим ошибку:

```
...
Error: Nothing to do
```

... значит репозиторий уже настроен. Идем дальше.

Устанавливаем postgresql:

```
yum install postgresql13 postgresql13-server postgresql13-contrib
```

* где:

- postgresql13 — клиент.
- postgresql13-server — сервер.
- postgresql13-contrib — набор дополнительных утилит и расширений для postgresql.

Для приложения Lixms BI также требуется установить следующие расширения:

```
yum install pgsql13-http pgsql13-keydb-fdw pgsql13-plv8 pgsql13-redis-pubsub
```

Инициализируем базу для нового postgresql:

```
/usr/pgsql-13/bin/postgresql-13-setup initdb
```

Откроем конфигурационный файл для postgresql 13:

```
vi /var/lib/pgsql/13/data/postgresql.conf
```

Как минимум, нам нужно поменять порт, на котором должен запуститься наш сервер:

```
port = 5433
```

* порт по умолчанию 5432 и, скорее всего, на нем работает наш сервер версии 11, который мы будем обновлять. Поэтому мы поменяли порт, например, на 5433.

Стоит сравнить настройки для файлов postgresql.conf и pg_hba.conf. Некоторые настройки, которые явно менялись для текущей версии СУБД, стоит перенести в конфигурационные файлы нового postgresql.

Запускаем сервис для установленного PostgreSQL:

```
systemctl start postgresql-13
```

И разрешаем его автозапуск:

```
systemctl enable postgresql-13
```

Стоит сразу проверить, запустилась ли служба и слушает ли сервис на нужном порту (мы настроили 5433):

```
systemctl status postgresql-13
```

```
ss -tunlp | grep :5433
```

G.2.2. Обновление postgresql

Процесс состоит из двух этапов - тестирование и обновление.

G.2.2.1. Сбор данных и запуск проверки на возможность обновления

И так, на текущий момент у нас запущены две версии postgresql (в нашем примере 11 и 13). Так как СУБД позволяет системному администратору тонко настроить расположение путей до рабочих данных и конфигов, выполним запросы, которые позволят однозначно определить их локацию.

Для текущей версии:

```
su - postgres -c "psql"
```

```
=
# SELECT current_setting('data_directory'), current_setting('config_file');
```

```
=
# quit
```

Для новой:

```
su - postgres -c "PGPORT=5433 psql"
```

```
=
# SELECT current_setting('data_directory'), current_setting('config_file');
```

```
=
# quit
```

Фиксируем полученные ответы. Они нам понадобятся для теста конфигурации.

Останавливаем службу postgresql для новой версии:

```
systemctl stop postgresql-13
```

Выполняем тест — в моем случае получилась такая команда:

```
su - postgres -c " \
/usr/pgsql-13/bin/pg_upgrade \--
old-datadir=/var/lib/pgsql/11/data \--
new-datadir=/var/lib/pgsql/13/data \--
old-bindir=/usr/pgsql-11/bin \--
new-bindir=/usr/pgsql-13/bin \--
old-options '-c config_file=/var/lib/pgsql/11/data/postgresql.conf' \--
new-options '-c config_file=/var/lib/pgsql/13/data/postgresql.conf' \--
check \
"
```

* еще раз стоит отметить, что пути зависят от версий postgresql и индивидуальных настроек.

Описание опций pg_upgrade:

- **-b bindir или –old-bindir=bindir или системная переменная PGBINOLD** Путь до каталога с бинарными файлами старого PostgreSQL.
- **-B bindir или –new-bindir=bindir или системная переменная PGBINNEW** Путь до каталога с бинарными файлами нового PostgreSQL.
- **-c или –check** Не вносить изменений, только проверка на возможность выполнить обновление.
- **-d configdir или –old-datadir=configdir или системная переменная PGDATAOLD** Путь до рабочего каталога с данными старого PostgreSQL.
- **-D configdir или –new-datadir=configdir или системная переменная PGDATANEW** Путь до рабочего каталога с данными нового PostgreSQL.
- **-j njobs или –jobs=njobs** Количество одновременных процессов для использования

- **-k или –link** Не копировать данные из старой СУБД в новую. Вместо этого использовать жесткую ссылку.
- **-o options или –old-options options** Опции, с которыми запускается старый PostgreSQL.
- **-O options или –new-options options** Опции, с которыми запускается новый PostgreSQL.
- **-p port или –old-port=port или системная переменная PGPORTOLD** Порт, на котором слушает старый PostgreSQL.
- **-P port или –new-port=port или системная переменная PGPORTNEW** Порт, на котором слушает новый PostgreSQL.
- **-r или –retain** Сохранить файлы SQL и журналов даже после успешного обновления
- **-s dir или –socketdir=dir или системная переменная PGSOCKETDIR** Каталог для сокетов postmaster во время обновления. По умолчанию текущий рабочий каталог.
- **-U username или –username=username или системная переменная PGUSER** Пользователь, под которым запускать процедуру обновления.
- **-v или –verbose** Подробный вывод информации.
- **-V или –version** Показать версию.
- **–clone** Эффективное клонирование файлов (также известное как «ссылки» в некоторых системах) вместо копирования файлов в новый кластер. Это может привести к почти мгновенному копированию файлов данных, предоставляя преимущества скорости -k/–link, оставляя старый кластер нетронутым. В настоящее время он поддерживается в Linux (ядро 4.5 или новее) с Btrfs и XFS (в файловых системах, созданных с поддержкой reflink), а также в macOS с APFS.

Если все хорошо, то мы увидим:

```
Performing Consistency Checks on Old Live Server--------------------------------
----------------

Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for system-defined composite types in user tables  ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for tables WITH OIDS                               ok
Checking for invalid "sql_identifier" user columns          ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  ok
Checking for prepared transactions                          ok
Checking for new cluster tablespace directories             ok
```

```
*Clusters are compatible*
```

Мы готовы обновить СУБД.

G.2.2.2. Обновление PostgreSQL

Остается выполнить само обновление.

Сначала нужно остановить текущий экземпляр СУБД и запретить его автозапуск (если данный экземпляр не содержит других нужных для работы баз):

```
systemctl stop postgresql-11
```

```
systemctl disable postgresql-11
```

Для обновления используем такую же команду, как при проверке, за исключением опции check:

```
su - postgres -c " \
/usr/pgsql-13/bin/pg_upgrade \--
old-datadir=/var/lib/pgsql/11/data \--
new-datadir=/var/lib/pgsql/13/data \--
old-bindir=/usr/pgsql-11/bin \--
new-bindir=/usr/pgsql-13/bin \--
old-options '-c config_file=/var/lib/pgsql/11/data/postgresql.conf' \--
new-options '-c config_file=/var/lib/pgsql/13/data/postgresql.conf' \--
link \
"
```

Обратите внимание на опцию –link. Она создает hard link вместо полноценных копий данных. Это экономит дисковое пространство.

После ее работы мы должны увидеть:

```
Upgrade Complete----------------

Optimizer statistics are not transferred by pg_upgrade so,
once you start the new server, consider running:
    ./analyze_new_cluster.sh

Running this script will delete the old cluster's data files:
    ./delete_old_cluster.sh
```

В данном тексте предлагается перенести статистику оптимизатора на новый сервер. В двух словах, данная статистика позволяет делать большие запросы быстрее. Также, в сообщении предлагается удалить данные старого сервера.

Открываем конфигурационный файл:

```
vi /var/lib/pgsql/13/data/postgresql.conf
```

Меняем порт, на котором должен слушать сервер:

```
port = 5432
```

* ранее мы использовали порт 5433.

Стартуем новый сервер:

```
systemctl start postgresql-13
```

G.2.3. Тест сервера и завершение настройки

Напоследок, проверим, что наш сервер выполняет запросы и настроим ему порт по умолчанию.

Зайдем в командную оболочку нового сервера:

```
su - postgres -c "psql"
```

На свое усмотрение, сделаем несколько запросов, чтобы убедиться в базовой работоспособности СУБД. Если запросы прошли, выходим из оболочки:

```
=
# quit
```

Проверяем работу портала Luxms BI.

Обновление PostgreSQL можно считать завершенным.

Если все хорошо, можно перенести статистику командой (для кластера это делаем на лидере):

```
su - postgres -c "/var/lib/pgsql/analyze_new_cluster.sh"
```

И если мы, совсем, уверены в работе нашего сервера, удаляем файлы старого PostgreSQL:

```
su - postgres -c "/var/lib/pgsql/delete_old_cluster.sh"
```

G.2.4. Возможные проблемы

В данном разделе рассмотрим проблемы, с которыми можно столкнуться при обновлении PostgreSQL.

G.2.4.1. Checking for presence of required libraries

Ошибка также сопровождается текстом:

```
Your installation references loadable libraries that are missing from the
new installation.  You can add these libraries to the new installation,
or remove the functions using them from the old installation.  A list of
problem libraries is in the file:
    loadable_libraries.txt
```

Причина: в новой версии PostgreSQL нет нужных библиотек для расширений, используемых в старой.

Решение: смотрим содержимое файла loadable_libraries.txt:

```
cat /var/lib/pgsql/loadable_libraries.txt
```

В нем перечислены библиотеки, которые нужно доустановить в новой версии. Установка расширений для postgresql, как правило, выполняется с помощью пакетного менеджера, например:

```
yum install pgsql13-tap
```

Однако, некоторые расширения нужно будет собирать, поэтому решение проблемы имеет индивидуальный характер.

G.2.4.2. Проблема подключения к СУБД после обновления

После завершения работы утилиты pg_upgrade и запуска службы, мы можем подключиться к СУБД от пользователя postgres, но не можем подключиться по сети или из приложения.

Причина: как правило, проблема в конфигурации pg_hba.

Решение: файл pg_hba.conf регламентирует условия подключения к СУБД — с каких узлов, для каких учетных записей, к каким базам и с помощью какого метода аутентификации. Необходимо привести в соответствие наши файлы. Данный файл находится в том же каталоге, что и основной файл конфигурации, в нашем примере это:

```
vi /var/lib/pgsql/11/data/pg_hba.conf
```

Файл для нового postgresql:

```
vi /var/lib/pgsql/13/data/pg_hba.conf
```

Также необходимо обратить внимание на методы шифрования паролей. Например, в 11 версии по умолчанию используется md5, например:

```
host    all             all             127.0.0.1/32            md5
```

В то время, как в 13 версии уже используется scram-sha-256:

```
host    all             all             127.0.0.1/32            scram-sha-256
```

Таким образом, при миграции данных в новую базу были перенесены и пароли с алгоритмом шифрования md5, а при подключении система пытается использовать scram-sha-256. Полученная таким образом последовательность не соответствует записанной, что приводит к ошибкам аутентификации. Для решения проблемы можно поменять scram-sha-256 на md5 в файле pg_hba.conf.

G.3. Обновление PostgreSQL на CentOS (кластер Patroni)

В нашей инструкции мы рассмотрим пример обновления СУБД PostgreSQL с версии 11 на версию 13. В качестве рабочей операционной системы будет использоваться CentOS 7.

Процедура обновления состоит из нескольких шагов:

1. Установка и запуск PostgreSQL новой версии (она будет работать параллельно со старой).
2. Запуск pg_upgrade для проверки возможности обновления.
3. Запуск pg_upgrade для выполнения обновления.
4. Проверка работоспособности СУБД.
5. Настройка новой версии в качестве основного экземпляра сервера баз данных.

Предполагается, что у нас уже работает кластер Patroni + Consul с PostgreSQL версии 11.

G.3.1. Установка и запуск PostgreSQL 13 (на всех узлах кластера)

В нашей инструкции мы планируем обновление до версии 13. Установим нужный нам пакет.

Так как в нашей системе уже установлен PostgreSQL, скорее всего, репозиторий уже настроен.

Но если репозитория нет, то необходимо его установить. В зависимости от сценария безопасности, есть два варианта: 1. Если используются внешние репозитории - устанавливаем его. 1. Если доступ к внешним репозиям отсутствует, используем свои внутренние репозитории с PostgreSQL 13.

Устанавливаем postgresql:

```
sudo yum install postgresql13 postgresql13-server postgresql13-contrib
```

* где:

- postgresql13 — клиент.
- postgresql13-server — сервер.
- postgresql13-contrib — набор дополнительных утилит и расширений для postgresql.

Для приложения Lixms BI также требуется установить следующие расширения:

```
sudo yum install pgsql13-http pgsql13-keydb-fdw pgsql13-plv8 pgsql13-redis-↩
  pubsub pgsql13-tap
```

G.3.2. Обновление postgresql

Процесс состоит из пяти этапов: 1. остановка узлов replica. 1. проверка состояния перед обновлением. 1. обновление. 1. тестирование. 1. включение узлов replica.

G.3.2.1. Останавливаем узлы с репликой

Чтобы не прописывать путь к конфигурационному файлу патрони, вводим:

```
export PATRONICTL_CONFIG_FILE=/etc/patroni/patroni.yml
```

Определяем, какая в данный момент узел работает в режиме Replica:

```
sudo patronictl list
```

Мы должны увидеть что-то на подобие:

```
+
 Cluster: pgdb (7151035635083057487) -------+---------+----+-----------+
| Member         | Host             | Role    | State   | TL | Lag in MB |+------↩
  ----------+-----------------+---------+---------+----+-----------+

| test01         | 192.168.0.200    | Replica | running |  2 |         0 |
| test02         | 192.168.0.210    | Leader  | running |  2 |           |+------↩
  ----------+-----------------+---------+---------+----+-----------+
```

В нашем примере Replica на сервере test01 - заходим на него и останавливаем патрони:

```
sudo systemctl stop patroni
```

После переходим на сервер с ролью Leader. Обновление будем выполнять на нем.

G.3.2.2. Сбор данных и запуск проверки на возможность обновления

И так, на текущий момент у нас установлены две версии postgresql (в нашем примере patroni - 11 и pgsql - 13).

Инициализируем базу для pgsql13:

```
sudo su - postgres -c "/usr/pgsql-13/bin/pg_ctl -D /pgdata/newdata initdb"
```

* новая база будет находиться в каталоге /pgdata/newdata. После обновления мы планируем вернуть рабочий каталог /pgdata/data.

Изучаем конфигурационный файл патрони:

```
cat /etc/patroni/patroni.yml
```

Необходимо обратить внимание на опции: - **data_dir** - путь хранения данных. - **bin_dir** - каталог с бинарниками.

Данные пути нам нужны для выполнения теста:

```
sudo su - postgres -c " \
/usr/pgsql-13/bin/pg_upgrade \--
old-datadir=/pgdata/data \--
new-datadir=/pgdata/newdata \--
old-bindir=/usr/pgsql-11/bin \--
new-bindir=/usr/pgsql-13/bin \--
old-options '-c config_file=/pgdata/data/postgresql.conf' \--
```

```
new-options '-c config_file=/pgdata/newdata/postgresql.conf' \--
check \
"
```

Если все хорошо, то мы увидим:

```
Performing Consistency Checks on Old Live Server--------------------------------
  ----------------

Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for system-defined composite types in user tables  ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for tables WITH OIDS                               ok
Checking for invalid "sql_identifier" user columns          ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  ok
Checking for prepared transactions                          ok
Checking for new cluster tablespace directories             ok

*Clusters are compatible*
```

Мы готовы обновить СУБД.

G.3.2.3. Обновление PostgreSQL

Остается выполнить само обновление.

Сначала нужно остановить patroni уже на лидере:

```
sudo systemctl stop patroni
```

Для обновления используем такую же команду, как при проверке, за исключением опции check:

```
sudo su - postgres -c " \
/usr/pgsql-13/bin/pg_upgrade \--
old-datadir=/pgdata/data \--
new-datadir=/pgdata/newdata \--
old-bindir=/usr/pgsql-11/bin \--
new-bindir=/usr/pgsql-13/bin \--
old-options '-c config_file=/pgdata/data/postgresql.conf' \--
new-options '-c config_file=/pgdata/newdata/postgresql.conf' \--
link \
"
```

Обратите внимание на опцию –link. Она создает hard link вместо полноценных копий данных. Это экономит дисковое пространство.

После ее работы мы должны увидеть:

```
Upgrade Complete----------------

Optimizer statistics are not transferred by pg_upgrade so,
once you start the new server, consider running:
    ./analyze_new_cluster.sh

Running this script will delete the old cluster's data files:
    ./delete_old_cluster.sh
```

В данном тексте предлагается перенести статистику оптимизатора на новый сервер. В двух словах, данная статистика позволяет делать большие запросы быстрее. Также, в сообщении предлагается удалить данные старого сервера.

Откроем конфигурационный файл patroni:

```
sudo vi /etc/patroni/patroni.yml
```

Внесем изменения в путь к бинарникам:

```
...
postgresql:
  ...
  bin_dir: /usr/pgsql-13/bin/
  ...
```

Удаляем каталог со старыми данными:

```
rm -rf /pgdata/data
```

Переносим каталог с новыми данным в каталог, который использовался до обновления:

```
mv /pgdata/newdata /pgdata/data
```

Открываем файл:

```
vi /pgdata/data/postmaster.opts
```

Меняем в путях /pgdata/newdata на /pgdata/data.

Переходим на сервер консула. Нам нужно удалить ключ-значение initialize для нашего кластера. Это можно сделать в веб-интерфейсе, перейдя в раздел Key/Value - service - pgdb (данное имя зависит от настройки scope в patroni) - initialize.

Также мы можем удалить данный ключ из командной строки:

```
sudo consul kv delete service/pgdb/initialize
```

* еще раз напомню, pgdb зависит от названия кластера patroni (опция scope).

Стартуем патрони:

```
sudo systemctl start patroni
```

Проверить, что служба запустилась и наш сервер стал лидером можно командой:

```
patronictl list
```

Запускаем службы luxms, которые были нами выключены перед началом работ:

```
sudo systemctl start luxmsbi-datagate luxmsbi-importer luxmsbi-web luxmsbi-↩
  appserver
```

G.3.2.4. Тест сервера и завершение настройки

Проверяем, что наш сервер выполняет запросы.

Зайдем в командную оболочку нового сервера:

```
sudo su - postgres -c "psql"
```

На свое усмотрение, сделаем несколько запросов, чтобы убедиться в базовой работоспособности СУБД. Если запросы прошли, выходим из оболочки:

```
=
# quit
```

Проверяем работу портала Luxms BI.

Изучаем журнал работы patroni:

```
journalctl -u patroni -f
```

А также журанал работы postgresql:

```
tail -f /pgdata/data/log/postgresql-<День недели>.log
```

Если детальное изучение работы приложения, запросов и логов не выявили ошибок, обновление PostgreSQL на одном из хостов можно считать завершенным.

Можно перенести статистику командой (делаем только на лидере):

```
sudo su - postgres -c "/var/lib/pgsql/analyze_new_cluster.sh"
```

G.3.2.5. Запуск узлов replica

Убедившись в работоспособности лидера, переключаем реплики на новую базу. Для этого откроем конфигурационный файл:

```
sudo vi /etc/patroni/patroni.yml
```

Внесем изменения в путь с bin_dir:

```
...
postgresql:
  bin_dir: /usr/pgsql-13/bin/
  ...
```

Удаляем содержимое старого каталога:

```
sudo rm -rf /pgdata/data/*
```

Стартуем патрони:

```
sudo systemctl start patroni
```

Проверяем статус:

```
patronictl list
```

G.3.3. Откат базы

Если в результате проверки работоспособности базы мы обнаружили ошибки и приняли решение вернуть старую версию базы, выполняем слудеющие действия.

На лидере останавливаем патрони:

```
sudo systemctl stop patroni
```

Переходим на сервер консула. Нам нужно удалить ключ-значение initialize для нашего кластера. Это можно сделать в веб-интерфейсе, перейдя в раздел Key/Value - service - pgdb (данное имя зависит от настройки scope в patroni) - initialize.

Также мы можем удалить данный ключ из командной строки:

```
sudo consul kv delete service/pgdb/initialize
```

* еще раз напомню, pgdb зависит от названия кластера patroni (опция scope).

На реплике, где мы не обновляли базу, запускаем patroni:

```
sudo systemctl start patroni
```

Проверяем. Команда:

```
patronictl list
```

... должна вернуть нам что-то на подобие:

```
+
 Cluster: pgdb (7151035635083057487) -------+---------+----+-----------+
| Member         | Host             | Role    | State   | TL | Lag in MB |+------↩
  ----------+-----------------+---------+---------+----+-----------+
```

```
| test01          | 192.168.0.200   | Leader  | running |  2 |         0 |+------↩
  ----------+-----------------+---------+---------+----+-----------+
```

Теперь лидером является test01. Идем на сервер, где ранее обновлял базу. Удаляем данные из каталога postgresql:

```
sudo rm -rf /pgdata/data/*
```

Открываем на редактирование конфигурайионный файл патрони:

```
sudo vi /etc/patroni/patroni.yml
```

Вернем значение для пути с bin_dir:

```
...
postgresql:
  bin_dir: /usr/pgsql-11/bin/
  ...
```

Запускаем патрони:

```
sudo systemctl start patroni
```

G.4. Обновление PostgreSQL с переходом на новую операционную систему

При необходимости смены операционной системы, например, при ее обновлении на более свежую версию или переходе на отечественный аналог, мы можем также обновить СУБД. Наши действия будут сведены к нескольким пунктам:

1. Установке и настройке СУБД.
2. Установке необходимых пакетов.
3. Восстановлении базы из дампа.

> В данной инструкции мы рассмотрим пример перехода с CentOS 7 + PostgreSQL 11 на РЕД ОС 7.3 + Postgres Pro 13. Действия для других систем и СУБД будут схожими.

G.4.1. Установка и настройка СУБД

Подробнее про установку СУБД PostgreSQL рассказано в инструкции “Руководство системного администратора”, разделе “Установка и настройка сервера БД”.

Для развертывания кластера СУБД прочитайте приложение “Установка отказоустойчивой БД”.

После установки, инициализации базы и запуска службы СУБД вводим команду:

```
sudo -iu postgres psql -c "ALTER SYSTEM SET password_encryption = 'md5';"
```

Далее нужно выполнить тюнинг базы в соответствии с вашей аппаратной состовляющей. Для более эффективного решения данной задачи можно использовать онлайн калькуляторы для оптимизации postgresql.

G.4.2. Установка расширений для СУБД

Для того, чтобы восстановления из дампа на новом сервере прошло без ошибок, необходимо установить расширения для postgresql. Точное название пакетов зависит от установленного PostgreSQL - это могут быть пакеты `pgsql` или `pgpro`. В нашем примере рассматривается новый сервер с Postgres Pro версии 13, поэтому команда установки расширений будет такой:

```
dnf -y install pgpro13-plv8 \
               pgpro13-http \
               pgpro13-redis-pubsub \
               pgpro13-keydb-fdw
```

Переходим к восстановлению данных.

G.4.3. Восстановление из резервной копии

Создаем роль bi. Для этого выполняем в SQL-оболочке команду под пользователем postgres:

```
sudo -iu postgres psql -c "CREATE ROLE bi WITH LOGIN PASSWORD 'bi';"
```

> В данном примере мы создадим роль `bi` с паролем `bi`. Последний используется только для примера, и в вашей инфраструктуре он должен быть таким же, как на старом сервере СУБД. В случае смены пароля также необходимо отредактировать настройки компонентов Luxms BI.
> Стоит иметь ввиду, что для функционирования Luxms BI достаточно только роли `bi`. Мы, хоть, и не рекомендуем, но не ограничиваем Клиента при создании дополнительных ролей. Поэтому, Если в Вашей БД существуют роли, созданные дополнительно, для успешного переноса дампа, может потребоваться предварительно перенести эти роли на новый сервер.

В зависимости от выбранного способа снятия дампа, наши действия по восстановлению будут отличаться. Так как резервная копия, сделанная с помощью pg_basebackup нам не подходит для восстановления данных на другой версии СУБД, рассмотрим процесс на примере утилиты `pg_dump`.

Предположим, что мы запускаем команду для снятия дампа на новом сервере. При этом, мы подключимся к старому серверу. Для этого вводим:

```
pg_dump -d postgresql://bi:bi@192.168.0.10/mi -Fc -C -c -f /tmp/mi.sqlc
```

* где `192.168.0.10` - адрес сервера, где находится старая версия PostgreSQL. `bi`:`bi` - логин и пароль для подклчюения.

Теперь восстанавливаем данные. Для нашего примера команда будет такой:

```
sudo -iu postgres pg_restore -C -d postgres /tmp/mi.sqlc
```

Мы можем получить ошибку при создании расширения pgtap. Ее можно проигнорировать.

В нашем примере дамп был снят с опцией -C, которая добавляет автоматическое создание базы. При необходимости создать пустую базу mi вручную, выполняем команду:

```
sudo -iu postgres createdb -E UTF-8 -O bi --lc-collate=ru_RU.UTF-
  8 --lc-ctype=ru_RU.UTF-8 -T template0 mi
```

После завершения восстановления данных потребуется инициализировать lpe (выполняется в SQL-оболочке):

```
sudo -iu postgres psql
```

```
ALTER DATABASE mi SET plv8.start_proc = '"lpe"."init"';
```

```
quit
```

Миграция завершена. Проверяем работу портала.

Если мы получили ошибку “auth return type not support” необходимо проверить, чтобы строка подключения в файле pg_hba.conf имела метод аутентификации `md5`, например:

```
host     all     all     127.0.0.1/32     md5
```

* подробнее про настройку pg_hba.conf можно почитать в документации “Руководство системного администратора”, раздел “Установка компонентов Luxms BI”.

В завершении наших работ стоит также установить пакет `luxmsbi-pgpro` (`luxmsbi-pg` для PostgreSQL версии не Pro):

```
dnf install luxmsbi-pgpro
```

В процессе мы получим ошибку:

```
ERROR:  DATABASE mi already exists.
```

Ее нужно проигнорировать.