Luxms BI

Luxms BI 8.0

визуальный управленческий контроль

Руководство системного администратора

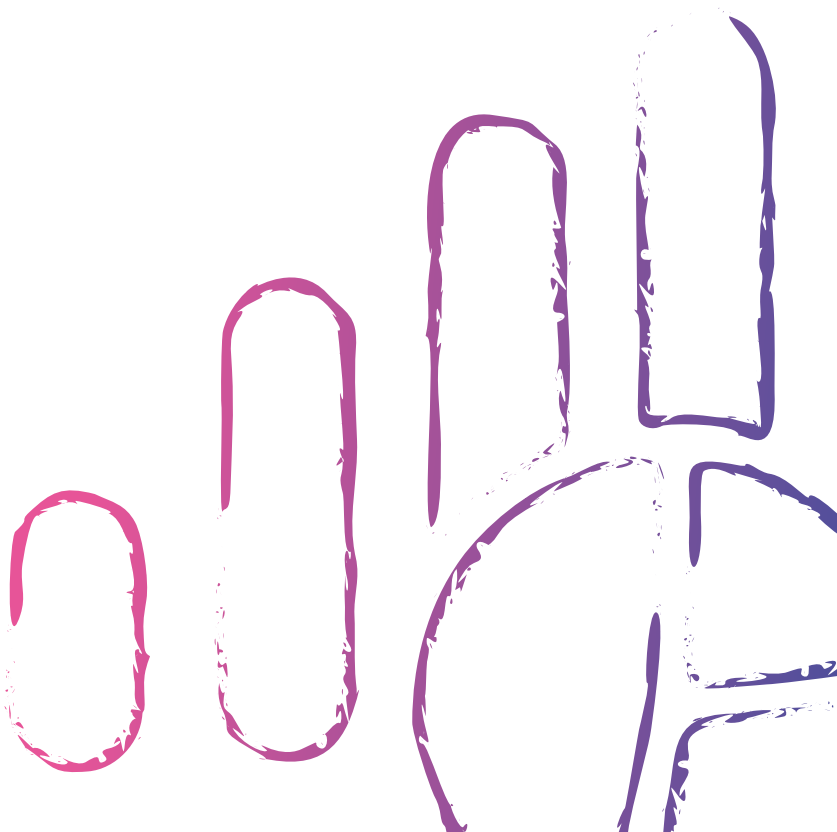
УСТАНОВКА

НАСТРОЙКА

ОБНОВЛЕНИЕ

РЕЗЕРВНОЕ КОПИРОВАНИЕ

МОНИТОРИНГ



2022

Оглавление

Краткое описание

Документ подготовлен для системных администраторов, которые занимаются планированием и подготовкой инфраструктуры, развертыванием 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;
- 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:

- Представления (Front-End Layer);
- Обработки (Back-End Layer);
- Хранения (Storage Layer);

1.1. Слой “Представления”

Реализован на базе NGinx (компонент `luxmsbi-web`) и усилен использованием Lua-скрипт. Дополнительно, функционал Front-End использует HTTP API компонета `luxmsbi-appseerver` (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.

1.2. Слой “Обработки”

1) `luxmsbi-appserver` (Java) - предоставляет API для управления настройками приложения Luxms BI и функционал импорта файлов в формате Excel. Имеет следующие интерфейсы:

- HTTP API - 8080/TCP.

Требует взаимодействия с:

- KeyDB сервером;
- БД Luxms BI.

Содержит сервер диагностики(Spring Boot Admin), при необходимости использования требуется:

- настройки конфигурации Java-приложений на подключение к функционалу диагностики
- настройки разрешений локального fairwall-а, при необходимости доступа с других узлов

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

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

Требует взаимодействия с:

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

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

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

Требует взаимодействия с:

- KeyDB сервером
- БД Luxms BI

1.3. Слой Хранения

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 Импорто-замещающего решения, дистрибутив для Astra Linux использует **PostgresProSql - полностью российский дистрибутив. Входит в Единый реестр, имеет сертификат ФСТЭК.

2. Варианты развертывания

В целях обучения и тестирования, мы предлагаем поставку Luxms BI в виде Appliance - это виртуальная машина ESXi с установленными базовыми компонентами и демонстрационными датасетами.

Для продуктовых инсталляций мы рекомендуем производить развертывание схем с учетом требований резервирования и масштабирования компонентов для поддержки высокой нагрузки.

При развертывании продуктовых инсталляций мы предлагаем архитектуру развертывания в виде графической схемы. И, после ее согласования, предоставляем Ansible-сценарии для выполнения развертывания Luxms BI.

Ниже приведены несколько примеров архитектуры. В схемах не указаны протоколы, используемые для кластеризации.

1. Базовая конфигурация включает в себя резервирование компонентов для каждого слоя системы Luxms BI. Схема, расположенная ниже, включает в себя резервирование на выделенных узлах:

- компонентов слоев “Представления” и “Обработки”;
- слоя “Хранения”.

2. Для высоконагруженных продуктовых инсталляций мы рекомендуем также выделение компонентов, осуществляющих интеграцию Luxms BI с Источниками данных. Ниже приведена схема с расширенной конфигурацией для продуктовой схемы.

2.1. Базовая конфигурация

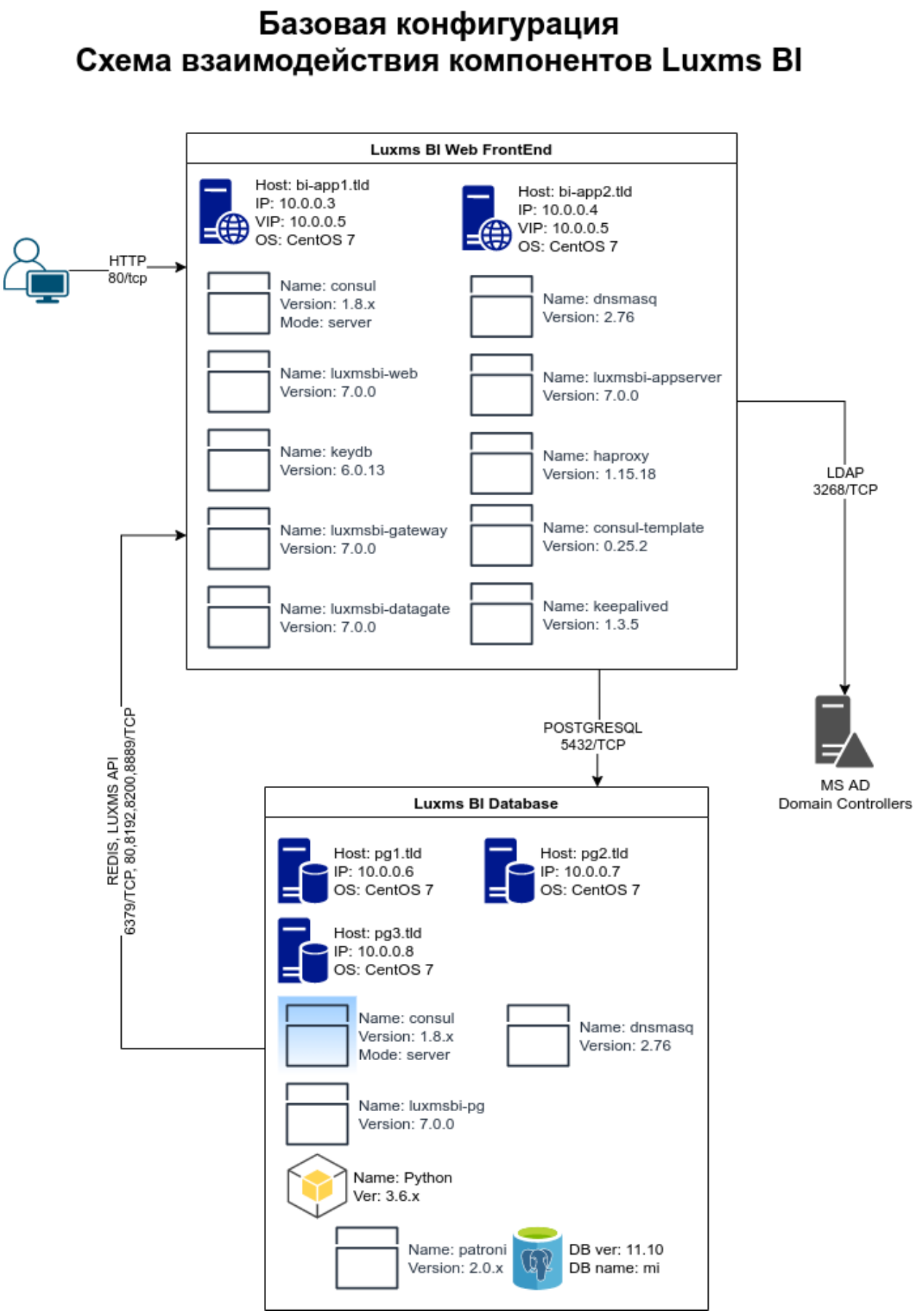


Рис. 2.1. **luxmsbi-deployment-base_schema.png**

2.2. Расширенная конфигурация

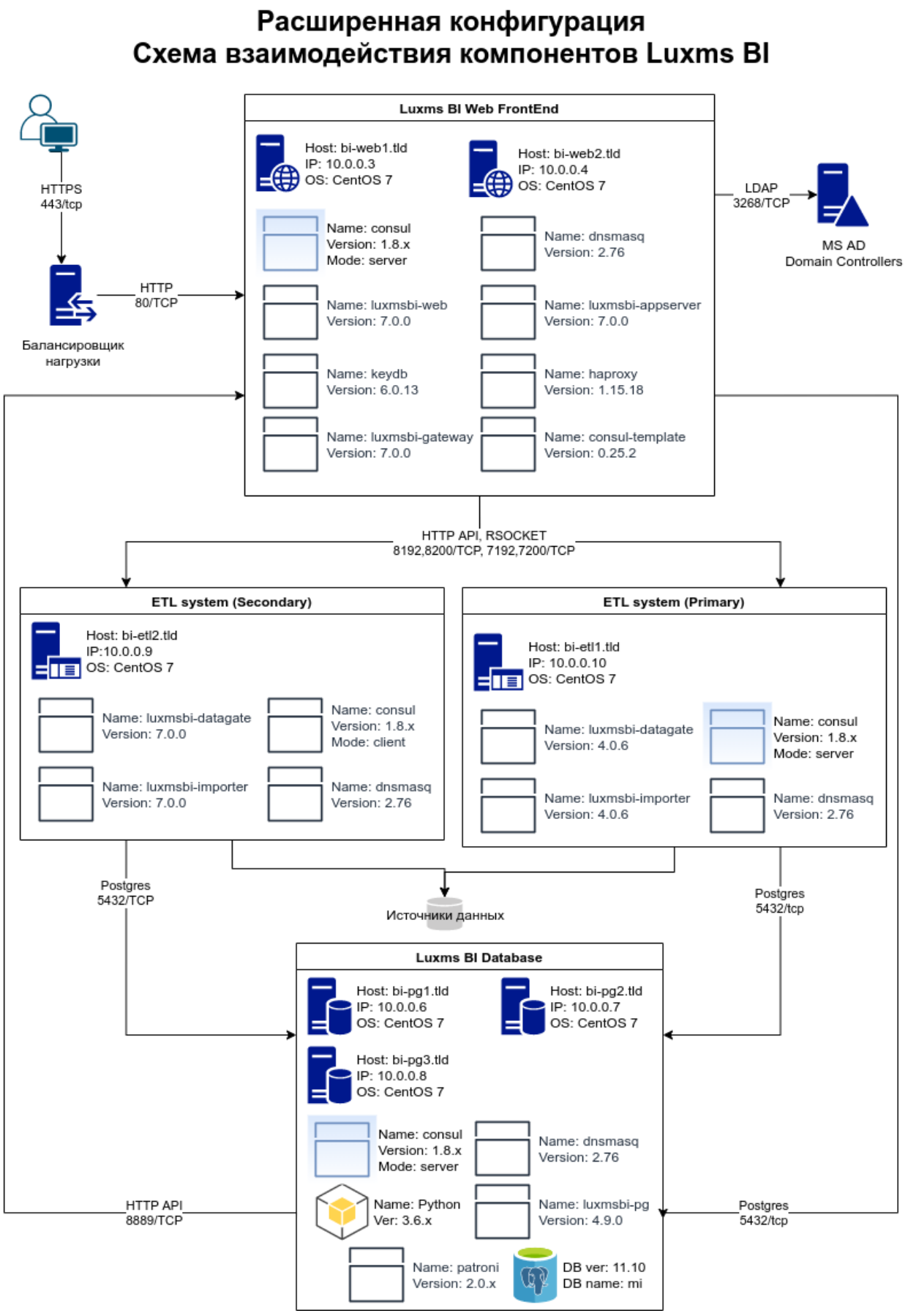


Рис. 2.2. **luxmsbi-deployment-extended_schema.png**

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 учётными 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.

Датасет (*DataSet*, так же *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 датасет.

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 развертыванию Luxms BI

Данные рекомендации предназначены для ознакомления Клиентов с требованиям Luxms BI при первичном развертывании. В документе перечислены точки монтирования файловой системы, предполагаюшие рост объема 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 безболезнено увеличить необходимые параметры систем.

При необходимости, Вы можете запросить расчет 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:

- RHEL/CentOS 7
- Astra Linux редакция “Орел”(Debian 9, stretch)

Доступ к персональным репозитариям 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 серверногого 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 одно-узловой 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
* /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. {.is-info}

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 содержит папку для постоянного хранения отчетных файлов - `/opt/luxmsbi/resources`. Размер данного раздела зависит от использования функционала Luxms BI в конкретной инсталяции.

3. **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. Для использования в условиях высокой нагрузки Мы предлагаем разнесение компонетов системы на следующие уровни:

- уровень Базы данных
- уровень Приложения
- вспомогательный уровень загрузки аггрегированных 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:

- **PostgreSQL 11**, для RedHat-based дистрибутивов;
- **PostgreProSQL 11**, для Debian-based дестрибутивов.

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:

1. Не рекомендуем использовать архивирование журнальных файлов Базы данных. Luxms BI в большей степени аналитическая система и предполагает переодичную 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 мало - 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.

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

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 системму с минимальным размером в 8 ГБайт и предусмотреть возможность увеличения размера за счет добавления дополнительных дисковых устройств.

2. Файлы данных и отчеты - `/opt/luxmsbi/resources`

При загрузке данных в Luxms BI из файлов загруженные файлы сохраняются в файловой системе для обеспечения возможности анализа первичных данных после загрузки. При генерации отчетов и презентаций результаты генерации также сохраняются на файловой системе и доступны пользователям через Web-приложение Luxms BI. Определение необходимого размера файловой системы зависит от режимов использования Luxms BI пользователями системы. Рекомендуемый минимальный размер, 2 Гбайта, для точки монтирования `/opt/luxmsbi` в большинстве случаев будет недостаточным. Поэтомы Мы рекомендуем предусмотреть возможность увеличения размера файловой системы в процессе эксплуатации с помощью LVM-менеджера.

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 - IPTables уже исключен из последнего релиза RedHat 8.

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

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, содержашую дату сборки пакета. Пакеты программного обеспечения Luxms BI опубликованы в персонализированных репозитариях, использующим аутентификационную учетную запись.

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

5.1. Обновление корневых сертификатов

Подключение репозитариев или установка из них пакетов может не работать на Ваших серверах, если корневые сертификаты ОС не самые свежие и/или не обновляются автоматически.

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

Для RPM-based ОС:

```
sudo yum -y update ca-certificates
sudo update-ca-trust extract
```

Для Deb-based ОС:

```
wget http://ftp.ru.debian.org/debian/pool/main/c/ca-certificates/
ca-certificates_20211004_all.deb
sudo dpkg -i ca-certificates_20211004_all.deb
sudo update-ca-certificates
```

5.2. Пакетное подключение репозитариев

Установка “релизного” пакета, опубликованного в Вашем персональном репозитарии, позволяет Вам настроить подключение к репозитариям и установить публичный ключ GPG. Рекомендуем Вам загрузить самый свежий пакет, для Вашей ОС, через Web-интерфейс сервера обновлений Luxms BI.

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

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

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

Установите эти пакеты в ОС ваших серверов использую необходимый пакетный менеджер - `apt` или `yum`.

5.3. Ручное подключение репозитариев

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

5.3.1. Подключение к YUM-репозиторию

Предоставляемые репозитории RPM-пакетов позволяют использовать собственные корпоративные решения по зеркалированию репозиториев и управлению пакетами, например **Spacewalk, Sattelite/Katello** или просто локальный репозитарий, созданный утилитой **createrepo**. Для настройки доступа к зеркалу репозитария, обратитесь к документации Вашего ПО зеркалирования репозитария.

“Релизный”-пакет устанавливает конфигурацию YUM-репозитория и GPG-ключи, необходимые для проверки полученного пакета на целостность.

Конфигурационный файл репозитария `/etc/yum.repos.d/luxmsbi.repo` выглядит следующим образом:

```
1 [luxms-thirdparty]
2 name=Luxms 3rd-party packages
3 baseurl=https://download.luxms.com/repository/thirdparty/el/↩
  $releasever/$basearch/
4 enabled=1
```

```
gpgcheck=0
repo_gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms

[luxms-bi]
name=Luxms BI Repository
baseurl=https://download.luxms.com/repository/[CLIENT]/7/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 учетной записи в параметре **baseurl=**, ‘[CLIENT];
- удалить знак комментария и указать имя учетной записи в параметре **username=**;
- удалить знак коментария и указать пароль учетной записи в параметре **password=**

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

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

5.3.2. Подключение к DEB-репозиторию

Предоставляемые репозитории DEB-пакетов позволяют использовать собственные корпоративные решения по зеркалированию репозиториев и управлению пакетами, например **reprepro**. Для настройки доступа к зеркалу репозитария, обратитесь к документации Вашего ПО зеркалирования репозитария.

“Релизный”-пакет устанавливает конфигурацию APT-репозитория и GPG-ключи, необходимые для проверки полученного пакета на целостность.

Конфигурационный файл репозитария `/eta/apt/source.list.d/luxmsbi.list` выглядит следующим образом:

```
# Replace password with yours in links.
deb https://customer:password@download.luxms.com/repository/stretch-customer stretch main
```

Для окончательной настройки доступа к репозиториям Luxms BI необходимо:

- проверить корректность имени учетной записи в URL-е репозитария, вместо **customer**;
- заменить **password** на пароль учетной предоставленной записи.

После настройки репозитария рекомендуем обновить локальный кеш пакетов:

```
sudo apt update
```

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

Для облегчения процесса получения RPM-пакетов программного обеспечения с открытым исходным кодом сторонних разработчиков ПО, мы предоставляем в персонализированном репозитарии клиента такие пакеты после проверки и подписания нашим GPG-ключом.

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

Для Deb-based ОС:

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

6. Установка и настройка сервера БД

6.1. Установка на RPM-based ОС

Для установке PostgreSQL сервера необходимо подключить дополнительные публичные репозитории или имеющиеся у клиента зеркала этих репозиториев:

- postgresql.org
- Extra Packages for Enterprise Linux (**EPEL**)

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
```

2. Установка пакетов сервера postgresql-11

```
sudo yum -y install postgresql11 \
postgresql11-server \
postgresql11-contrib
```

3. Инициализация PostgreSQL

Не забудьте предоставить разрешения файловой системы при монтировании отдельного раздела или использовании нестандартного расположения файлов БД.
`sudo chown -R postgres.postgres /var/lib/pgsql` При использовании нестандартного расположения, не забывайте определить переменную окружения **PGDATA**

```
su - postgres -c '/usr/pgsql-11/bin/initdb'
```

или из под суперпользователя

```
# sudo /usr/pgsql-11/bin/postgresql-11-setup initdb
```

> Обращаем внимание, если каталог базы данных отличается от дефолтного, необходимо переопределить переменную окружения **PGDATA** в: - профайле сервисной учетной записи `postgres` - `/var/lib/pgsql/.bash_profile` - systemd скрипте - `/lib/systemd/system/postgresql-11.service` {is.warning}

4. Запуск postgresql сервиса

```
sudo systemctl enable --now postgresql-11
```

6.2. Установка на DEB-based ОС

6.2.1. Проверка и установка русского языка

До установки PostgreSQL необходимо убедится в наличии локализации для русского языка, используется в базе данных для обеспечения поиска, выполните команду :

```
locale -a
C
C.UTF-8
en_US.utf8
POSIX
ru_RU.utf8
russian
```

Если в выводе команды отсутствует `ru_RU.utf8` необходимо выполнить последовательность комманд”

```
sudo apt-get install locales

sudo sed -i '/^# ru_RU\.UTF-8/ s/^#\s*//' /etc/locale.gen
sudo locale-gen
```

6.2.2. Установка PostgreProSQL

1. Установка необходимых репозитариев

Проверьте наличие подключенных репозитариев PostgreProSQL 11, Debian Stretch и Luxms в ОС. При необходимости подключите недостающий репозиторий:

```
# Luxms repository
sudo echo "deb https://download.luxms.com/repository/alo-bi7/ orel main" \
> /etc/apt/sources.list.d/luxmsbi.list
wget -q -O - https://download.luxms.com/repository/thirdparty/RPM-GPG-KEY-Luxms| apt-key add -
# PostgresPro 11 Standard repository
wget apt-repo-add.sh https://repo.postgrespro.ru/pgpro-11/keys/apt-repo-add.sh \   &&
sudo /bin/sh apt-repo-add.sh
# Debian stretch repository
sudo echo "deb https://ftp.debian.org/debian stretch main contrib" \
> /etc/apt/sources.list.d/stretch.list
sudo wget -q -O - https://ftp-master.debian.org/keys/archive-key-9.asc| apt-key add -
apt update
```

После установки репозитариев необходимо добавить данные аутентификации для репозитария Luxms BI в конфигурационный файл `/etc/apt/sources.list.d/luxmsbi.list`:

```
deb https://Имя_учетной_записи:Пароль_учетной_записи]@download.luxms.com/repository/alo-bi7/ orel main
```

2. Установка PostgreProSQL и необходимых расширений:

```
sudo apt -y install postgrespro-std-11 \
postgrespro-std-11-server \
postgrespro-std-11-contrib \
pgpro11-tap \
pgpro11-redis-pubsub \
pgpro11-http \
pgpro11-plv8 \
pgpro11-keydb-fdw
```

Установка PostgresProSQL не требует настройки автоматического запуска сервера БД - `postgrespro-std-11` - ОС AstraLinux выполняет это автоматически. Но эта же особенность требует дополнительных действий для установки БД в нестандартном местоположении. В том числе и удаление созданной по умолчанию БД.

7. Установка кластерной БД (patroni[consul])

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
- HAProxy - для поддержки пула соединений в БД и разделения запросов между Leader-ом и Replica-ми БД

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-ух узлов, для запуска экземпляра БД.

Если Ваша инсталяция содержит большое количество узлов, рекомендуем ознакомиться с Приложением “Планирование DCS Consul” данного документа.

До начала установки DCS Consul необходимо определить единый ***сектретный токен***, используемый для шифрования информационного обмена между всеми экземплярами Consul. Выполните генерацию токена одним из двух способов:

```
# /usr/bin/consul keygen
```

или используйте сторонние утилиты, например `openssl`:

```
openssl rand -base64 32
```

Установку Consul DCS рекомендуем выполнять из репозитория компании HashiCorp, детальная информация по подключению репозитория для различных операционных систем описаны на сайте производителя.

7.1. Установка и настройка Consul DCS

1. Настройка репозитария и установка.

Для RPM-based ОС:

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://
rpm.releases.hashicorp.com/RHEL/hashicorp.repo
sudo yum -y install consul
```

Для DEB-based ОС:

```
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key
add -
sudo apt-add-repository "deb [arch=amd64] https://
apt.releases.hashicorp.com $(lsb_release -cs) main"
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",
"domain": "consul",
"enable_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": [
"centos-1.local",
"centos-2.local",
"centos-3.local"  ],

"server": true,
"bootstrap_expect": 3,
"ui": 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",
"domain": "consul",
"enable_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": [
"centos-1.local",
```

```
"centos-2.local",
"centos-3.local"  ]

}
```

- **bind_addr** — адрес, на котором будет слушать наш сервер консул. Это может быть IP любого из наших сетевых интерфейсов или, как в данном примере, все.
- **bootstrap_expect** — ожидаемое количество серверов в кластере.
- **client_addr** — адрес, к которому будут привязаны клиентские интерфейсы.
- **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.
- **domain** — домен, в котором будет зарегистрирован сервис.
- **enable_script_checks** — разрешает на агенте проверку работоспособности.
- **dns_config** — параметры для настройки DNS.
- **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 завершиться 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/config.hcl

...
Configuration is valid!
```

3. Настройка разрешений локального firewall-а

Для обеспечения сетевого взаимодействия агентов DCS Consul, мы рекомендуем использование комплексных конфигурационных файлов.

Для RPM-based ОС создайте файл сервиса для Firewalld, следующего содержания:

Шаблон конфигурации сервиса Firewalld, consul.xml

```
<?xml version="1.0" encoding="utf-8"?>
<service>
<short>consul</short>
<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 </description>
<port protocol="tcp" port="8300"/>
<port protocol="tcp" port="8301"/>
<port protocol="tcp" port="8302"/>
<port protocol="tcp" port="8500"/>
<port protocol="tcp" port="8501"/>
<port protocol="tcp" port="8502"/>
<port protocol="tcp" port="8600"/>

<port protocol="udp" port="8301"/>
<port protocol="udp" port="8302"/>
<port protocol="udp" port="8600"/>
</service>
```

Для добавление сервиса в конфигурацию Firewalld, выполните следующую команду:

```
sudo firewall-cmd --permanent --new-service-from-file=consul.xml
sudo firewall-cmd --reload
sudo firewall-cmd --add-service=consul
sudo firewall-cmd --runtime-to-permanent
```

Для DEB-based ОС создайте файл сервиса для UFW, следующего содержания:

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

```
[Consul]
title=HashiCorp Consul DCS
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.
ports=8300:8302,8500:8502,8600/tcp|8301:8302,8600/udp
```

Для добавление application в конфигурацию UFW, выполните следующую команду:

```
sudo ufw app update --add-new Consul
sudo ufw allow Consul
```

4. Запуск DCS Consul и проверка работоспособности:

Запускаем сервис Consul-а на всех узлах:

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

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

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

или через Web-интерфейс, который доступен по ссылке `http://<node>:8500/`

7.2. Настройка разрешения ресурсов зоны .consul

Для обеспечения актуальности сведений о зарегестрированных 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.

7.2.1. Установка и настройка DNSMasq

Пакет **dnsmasq** опубликован в стандартных репозиториях, для его установки необходимо использовать следующую команду:

Для RPM-based ОС:

```
sudo yum -y install dnsmasq
```

Для DEB-based ОС:

```
sudo apt -y install dnsmasq
```

После установки пакета необходимо откорректировать файл конфигурации или установить следующую мимнимальную 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
```

7.2.2. 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.

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
```

7.2.3. Проверка разрешения DNS имен

Если конфигурация компонентов выполнена корректно, то выполнение проверочного запроса должно вернуть перечень адресов:

```
nslookup consul.service.luxmsbi.consul
```

```
...
Name:  consul.service.luxmsbi.consul
Address: 10.0.2.5
Name:  consul.service.luxmsbi.consul
Address: 10.0.2.7
Name:  consul.service.luxmsbi.consul
Address: 10.0.2.6
```

7.3. Установка и настройка Patroni

До установки Patroni необходимо установить пакеты БД и необходимых расширений в соответсвии с Разделом Установка и настройка сервера БД без развертывания БД Luxms BI.

ПО Patroni реализовано и поставляется в виде Python-модулей. Поэтому процесс установки требует доступа к Python Pacckage Index или к его локальному зеркалу.

7.3.1. Установка на RPM-based ОС

При установке модуля `patroni` большое значение версия модуля `pip` - на версии поставляемой RPM-пакетом `python3-pip` установка Patroni потребует сборки некоторых зависимых модулей из исходного кода. Поэтому необходимо обновление модуля `pip`.

Для установки, выполните следующую последовательность команд:

```
sudo yum -y install python3
python3 -m pip -y install pip --upgrade
sudo pip3 install patroni[consul] psycopg2-binary
```

Выполните настройку разрешения для Firewalld, для обеспечения доступа к БД с других узлов.

```
sudo firewall-cmd --permanent --add-service=postgresql
sudo firewall-cmd --reload
```

7.3.2. Установка на DEB-based ОС

К сожалению, Deb-пакет `python3-pip` содержит зависимости от пакетов разработки и компиляторов.
Для исключения установки среды разработки на продуктовые сервера, выполните операцию сборки WHL-пакетов на выделенном сервере и перенесите готовые бинарные пакеты на продуктовый сервер.

Для установки, выполните следующую последовательность команд:

```
sudo apt -y install python3 python3-pip python3-requests
sudo python3 -m pip install pip --upgrade
sudo pip3 install patroni[consul] psycopg2-binary
```

7.3.3. Установка конфигурации Patroni

Мы планируем сборку пакетов для инсталляции Patroni для RPM-based и DEB-based ОС в ближайшее время и предоставление этих пакетов своим партнерам и клиентам.
На данный момент эта секция описывает ручную инсталляцию, которую мы рекомендуем заменить на инсталляцию с использованием сценариев Ansible.

Базовая конфигурация сервиса Patroni предоставлена ниже. Необходимо заменить значения параметров, на соотвествующие вашей инсталяции 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.authentication
- postgresql.pg_rewind
- postgresql.replication
- postgresql.superuser

Детальное описание параметров рекомендуется прочитать на сайте производителя.

Шаблон конфигурации, /etc/patroni/patroni.yml

```
name: centos-2.local
scope: postgresdb

watchdog:
mode: off
```

```
consul:
host: "localhost:8500"
register_service: true
## token: <ACL-token if used>

restapi:
listen: 0.0.0.0:8008
connect_address: "10.0.2.4:8008"
authenticate: 'username: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
max_wal_senders: 10
wal_keep_segments: 8
max_replication_slots: 5
hot_standby: "on"
wal_log_hints: "on"

initdb:  -
encoding: UTF8  -
data-checksums

pg_hba:  # Add following lines to pg_hba.conf after running ↩
'initdb'  -
local all all trust  -
host replication replicator 10.0.2.0/24 md5  -
host replication replicator 127.0.0.1/32 trust  -
host all all 0.0.0.0/0 md5

postgresql:
pgpass: /var/lib/postgresql/.pgpass
listen: 0.0.0.0:5432
connect_address: "10.0.2.4:5432"
data_dir: /var/lib/postgresql/data
bin_dir: /usr/pgsql-11/bin/
```

```
pg_rewind:
username: postgres
password: password
pg_hba:    -
local all all trust    -
host replication replicator 10.0.2.0/24 md5    -
host replication replicator 127.0.0.1/32 trust    -
host all all 0.0.0.0/0 md5
replication:
username: replicator
password: password
superuser:
username: postgres
password: password
```

После корректного заполнения конфигурационного файла, необходимо создать сервисный файл для Systemd:

Шаблон systemd service unit, /etc/systemd/system/patroni.service

```
[Unit]
Description=Runners to orchestrate a high-availability PostgreSQL
Documentation=https://github.com/zalando/patroni/tree/master/docs
After=syslog.target network.target

[Service]
Type=simple
User=postgres
Group=postgres
ExecStart=/usr/local/bin/patroni /etc/patroni/patroni.yml
KillMode=process
TimeoutSec=30
Restart=no

[Install]
WantedBy=multi-user.target
```

После создания сервисного файла, необходимо настроить его автоматический запуск b pfgecnbnm tuj:

```
sudo systemctl enable patroni --now
```

7.3.4. 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 -c /etc//patroni/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

7.4. Рекомендации по подключению к БД

При кластеризации особенно важно организовать подключение к БД с использованием постоянного пула соединений. Мы рекомендуем использование HAProxy для организации взаимодействия, с установкой этого решения на узлах с Web-приложением Luxms BI. Для настройки постоянного пула соединений ознакомтесь с Приложением 3

8. Установка компонентов Luxms BI

Для установки компонентов Luxms BI из пакетов необходимо подключение дополнительных репозиториев.

Для RPM-based ОС:

- Extra Packages for Enterprise Linux (**EPEL**)
- Luxms RPM Repo см. 5 Использование пакетных менеджеров и репозиториев

Для DEB-based ОС:

- BelSoft Liberica
- Luxms DEB Repo см. 5.3.1 Подключение к YUM-репозиторию

8.1. 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.

8.1.1. Автоматизированная установка БД LuxmsBI

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

1. Установка пакета luxmsbi-pg при работающей БД выполнит до-настройку конфигурационных файлов и создаст БД для системы Luxms BI. Для установки пакета необходимо выполнить команду.

Для RPM-based ОС:

```
sudo yum -y install luxmsbi-pg
```

Для DEB-based ОС:

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

8.1.2. 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   all          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.

8.1.3. Создание администратора приложения Luxms BI (adm).

```
su - postgres
psql -U postgres -d mi -c "SELECT adm.create_user('Default Admin','adm','luxmsbi','','','{}','{}',false);"
```

8.2. Установка KeyDB сервера

Взаимодействия между компонентами Luxms BI построено с использованием быстрой in-memory database KeyDB. В KeyDB реализованы дополнительный функционал, расширяющий стандартные возможности Redis. Установка KeyDB производится, в большинстве случаев, совместно с компонентом `luxmsbi-web`.

1. Установка

Для RPM-based ОС

```
sudo yum -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. Для RPM-based ОС, необходима настройка автоматического запуска сервера:

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

4. При развертывании нескольких экземпляров KeyDB, необходимо настроить репликацию данных. В конфигурационном файле необходимо установить следующие параметры:

```

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

8.3. Развертывание Web приложения

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

- KeyDB сервером
- с Java-приложением `luxmsbi-appserver`
- 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 - `lixmsbi-web.service`.

Для установки необходимо выполнить следующую последовательность действий:

1. Установить пакет `luxmsbi-web`

Для RPM-based ОС

```
sudo yum -y install luxmsbi-web
```

Для DEB-based ОС:

```
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, 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
dbbacklog=20, -- FIXME: Not yet propogated down to the db:connect!--
https://github.com/openresty/lua-nginx-module#tcpsockconnect
```

```
needs version 0.10.14
debug = true,
}
```

3. Для RPM-based ОС. Настроить полиси SELinux.

Если нет особых требований, то можно отключить SELinux, но рекомендуем не отказываться от защиты ОС. Для установки параметров SELinux потребуется инсталяция пакета **policycoreutils-python**.

```
sudo yum -y install policycoreutils-python
```

После инсталяции необходимо настроить параметры:

```
#Add tcp ports to permited HTTP ports
semanage port -a -t http_port_t -p tcp 8080
semanage port -a -t http_port_t -p tcp 8200
semanage port -a -t http_port_t -p tcp 8282
semanage port -a -t http_port_t -p tcp 8888

# Allow to use execmem
setsebool -P httpd_execmem on
# Allow to set resource limits
setsebool -P httpd_setrlimit on
# Allow to connect network sockets
setsebool -P httpd_can_network_connect on
# Allow connect to db
setsebool -P httpd_can_network_connect_db on
```

4. Выполнить настройку автоматического запуска systemd сервиса.

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

8.4. 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 14, который устанавливается как зависимость из репозитария Luxms.

1. Установка BINS.

Для RPM-based ОС

```
sudo yum -y install luxmsbi-bins
```

Для DEB-based ОС:

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

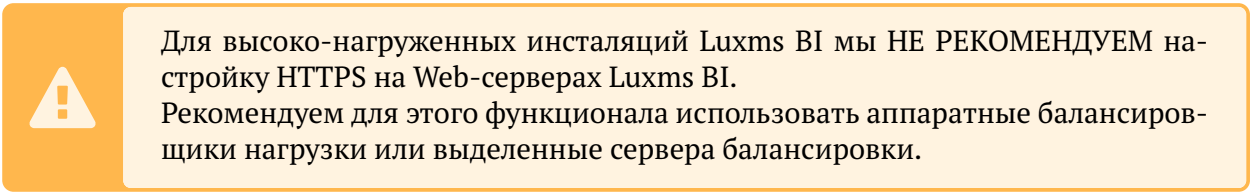
2. Настройки источников событий, используемых для трансляции в браузер пользователя производится в конфигурационном файле `/opt/luxmsbi/conf/bins.json`. В зависимости от схемы инсталяции, может потребовать корректировку строки подключения:

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

3. Для RPM-based ОС. Выполнить настройку автоматического запуска systemd сервиса.

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

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



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

8.5. Установка Java Runtime

Установка JRE/JDK для компонентов Luxms BI реализованых как Java-приложение не требуется для RPM-based ОС. Для ОС AstraLinux пакеты Java-приложений настроены на использование импорто-замещающих решений. Поэтому требуется дополнительная установка репозитария компании BelSoft и пакета `belsoft-java11`:

```
wget -q -O - https://download.bell-sw.com/pki/GPG-KEY-bellsoft | sudo apt-key add -
echo "deb [arch=amd64] https://apt.bell-sw.com/ stable main" \
| sudo tee /etc/apt/sources.list.d/bellsoft.list
sudo apt update
sudo apt -y install bellsoft-java11
```

8.6. Установка Luxms BI Appserver

Компонент `luxmsbi-appserver` требует взаимодействия с:

- KeyDB сервером
- БД Luxms BI

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

Для RPM-based ОС

```
sudo yum -y install luxmsbi-appserver
```

Для DEB-based ОС:

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

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

Необходимо настроить подключение к БД 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
```

3. Для RPM-based ОС. Выполнить настройку автоматического запуска systemd сервиса.

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

4. При необходимости обеспечения SpringBootAdmin дать разрешения на доступ к приложению, при наличии локального fairwall-а в ОС:

Для RPM-based ОС

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

Для DEB-based ОС:

```
sudo ufw allow to any app Luxmsbi-Appserver
```

8.7. Установка Luxms BI Importer

Компонент `luxmsbi-importer` требует взаимодействия с: - KeyDB сервером - БД Luxms BI - компонентом `luxmsbi-appserver` - компонентом `luxmsbi-datagate`

1. Установка приложения: Для RPM-based ОС

```
sudo yum -y install luxmsbi-importer
```

Для DEB-based ОС:

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

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

Необходимо настроить подключение к БД в конфигурационном файле:

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

3. 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
```

4. Для RPM-based ОС. Выполнить настройку автоматического запуска systemd сервиса.

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

8.8. Установка Luxms BI Datagate

Компонент `luxmsbi-datagate` требует взаимодействия с: - KeyDB сервером - БД Luxms BI

1. Установка приложения: Для RPM-based ОС

```
sudo yum -y install luxmsbi-datagate
```

Для DEB-based ОС:

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

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

Необходимо настроить подключение к БД в конфигурационном файле:

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

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

Для RPM-based ОС

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

Для DEB-based ОС:

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

4. Для RPM-based ОС. Выполнить настройку автоматического запуска systemd сервиса.

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

8.9. Установка Luxms Admin

Компонент `luxmsbi-admin` предоставляет расширение средств для управления конфигурацией приложения Luxms BI. Это функционал уже устарел и почти весь перенесен в компонент `luxmsbi-appserver`. Но еще существует в некоторых продуктовых инсталяциях. Требует взаимодействия с:

- компонентом `luxmsbi-web`
- БД Luxms BI

1. Установка приложения:

Для RPM-based ОС

```
sudo yum -y install luxmsbi-admin
```

Для DEB-based ОС:

```
sudo apt -y install luxmsbi-admin
```

2. Необходимо настроить подключение к БД в конфигурационном файле `/opt/luxmsbi/conf/luxmsbi-admin.pm`:

```
package Conf;

use strict;

use vars qw(%Conf);

my $database_db = 'mi';
my $database_user = 'bi';
my $database_password = 'bi';
```

Текущая версия пакета использует запуск команд в shell для создания датасетов, в связи с чем необходимо дополнительно настроить безпарольную аутентификацию.

В соответсвии с документацией PostgreSQL The Password File - `/opt/luxmsbi/.pgpass`:

```
# hostname:port:database:username:password
*:*:mi:bi:bi
```

3. Выполнить настройку автоматического запуска systemd сервисов:

Сервис для запуска CGI-приложения:

```
sudo systemctl enable luxmsbi-admin --now
```

Сервис для запуска загрузчика данных. Для периодического перезапуска загрузчика данных используется systemd timer, поэтому необходимо разрешить использование таймера для перезапуска:

```
sudo systemctl enable luxmsbi-timporter --now
sudo systemctl enable restart-luxmsbi-timporter.timer --now
```

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 коммандной строки, Consul предоставляет функционал управления:

- через Web-интерфейс
- 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';
```

Рекомендуем после установки, расчитать 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, регестрируются 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 комманд, приведенных ниже, требует изучения документации от производителя ПЗ. Утилита предоставляет вывод перечня доступных команд:

```
patronictl --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 при начале эксплуатации этого решения, но, в любом случае, Вам необходимо обратится к первоисточнику для изучения возможностей данной утилиты.

1. Проверка статуса узлов кластера:

```
patronictl -c /etc/patroni/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 /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 /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 на 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, параметер `-u`:

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

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

- Установити 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 перенапрвление вывода в файл или параметер `-no-pager`

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

Недостатк 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 vpupkin
```

10. Установка обновлений Luxms BI

10.1. Установка обновлений компонентов, кроме БД

Установка обновлений компонентов Luxms BI производится обновлением пакетов:

```
sudo yum -y update luxmsbi-web
```

При необходимости отката на предыдущую версию компонента, используйте команду:

```
sudo yum -y downgrade luxmsbi-web
```

10.2. 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 выполненным в ручном режиме:

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

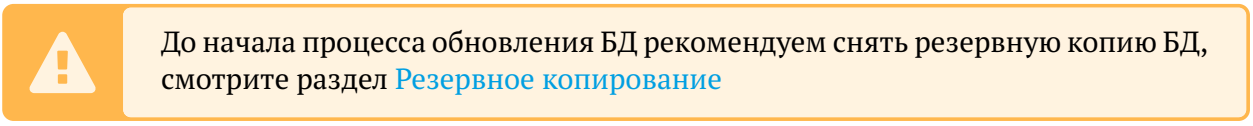
10.2.1. Очистка, возврат первоначального состояния БД

При необходимости восстановления первоначального состояния БД, необходимо 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)`

10.2.2. Обновление БД



Установка обновлений БД может выполняться только из командной строки. Установка обновлений работает в двух режимах:

1) Куммулятивная установка - установка всех необходимых обновлений

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

2) Выворочная установка - установка скрипта обновления на до конкретной версии

```
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 устанвки 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.

10.2.3. Обновление БД по требованиям Клиента

При аварийных ситуациях или дополнительного изменения БД под требования клиентов, поставка обновлений может производится 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

Переодичность снятия резервных копий определяется владельцем инсталяции Luxms BI, в соответсвии 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. Изменеие конфигурации компонентов производится в только 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 следущее:
“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. 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 что бы он перезапустил свою службу и обновил измененную конфигурацию на всех узлах кластера. Данные команды подаются на одном из узлов кластера и распространяется на все автоматически:

```
[root@bi-pg1 ~]# patronictl -c /etc/patroni/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 /etc/patroni/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:

```
[root@localhost backup]# pg_basebackup -d postgresql://replicator:password@172.16.32.113 --checkpoint=fast -D /backup -P -Ft -z -Xs

[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:

-U пользователь у которого есть replication слот -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.

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
```

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

```
[root@bi-pg1 ~]# patronictl -c /etc/patroni/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 комманды:

```
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 логфайлах будут вот 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
```

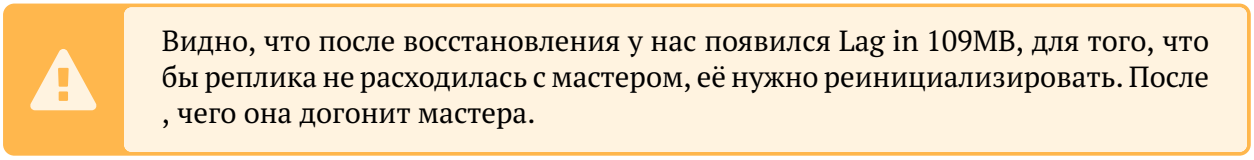
Если всё в порядке, то запускаем другие узлы кластера, через systemctl start patroni

Проверяем всё ли работает хорошо:

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

Cluster	Member	Host	Role	State	TL	Lag in MB
db-main	bi-pg1	172.16.32.112	Leader	running	4	

Cluster	Member	Host	Role	State	TL	Lag in MB
db-main	bi-pg2	172.16.32.113		running	3	109




```
[root@bi-pg2 ~]# patronictl -c /etc/patroni/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 мимнимальные 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 соответсвии 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 жкрналов БД

12.2. Мониторинг сервиса Core (luxmsbi-pg)

- URI: `/api/helthcheck`
- тип запроса: `HEAD`
- ожидаемый HTTP статус ответа: `200`

12.3. Мониторинг сервиса App Server (luxmsbi-appserver)

- URI `/actuator/helthcheck` (отправка запрсоа с localhost)
- тип запроса `GET`
- ожидаемый HTTP статус ответа: `200`

12.4. Мониторинг сервиса Luxms BI Importer (luxmsbi-importer)

- URI `/actuator/helthcheck` (отправка запрсоа с localhost)
- тип запроса `GET`
- ожидаемый HTTP статус ответа: `200`

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

- URI `/actuator/helthcheck` (отправка запрсоа с localhost)
- тип запроса `GET`
- ожидаемый HTTP статус ответа: `200`

Приложение А. Планирование DCS Consul

А.1. Типовая схема кластера

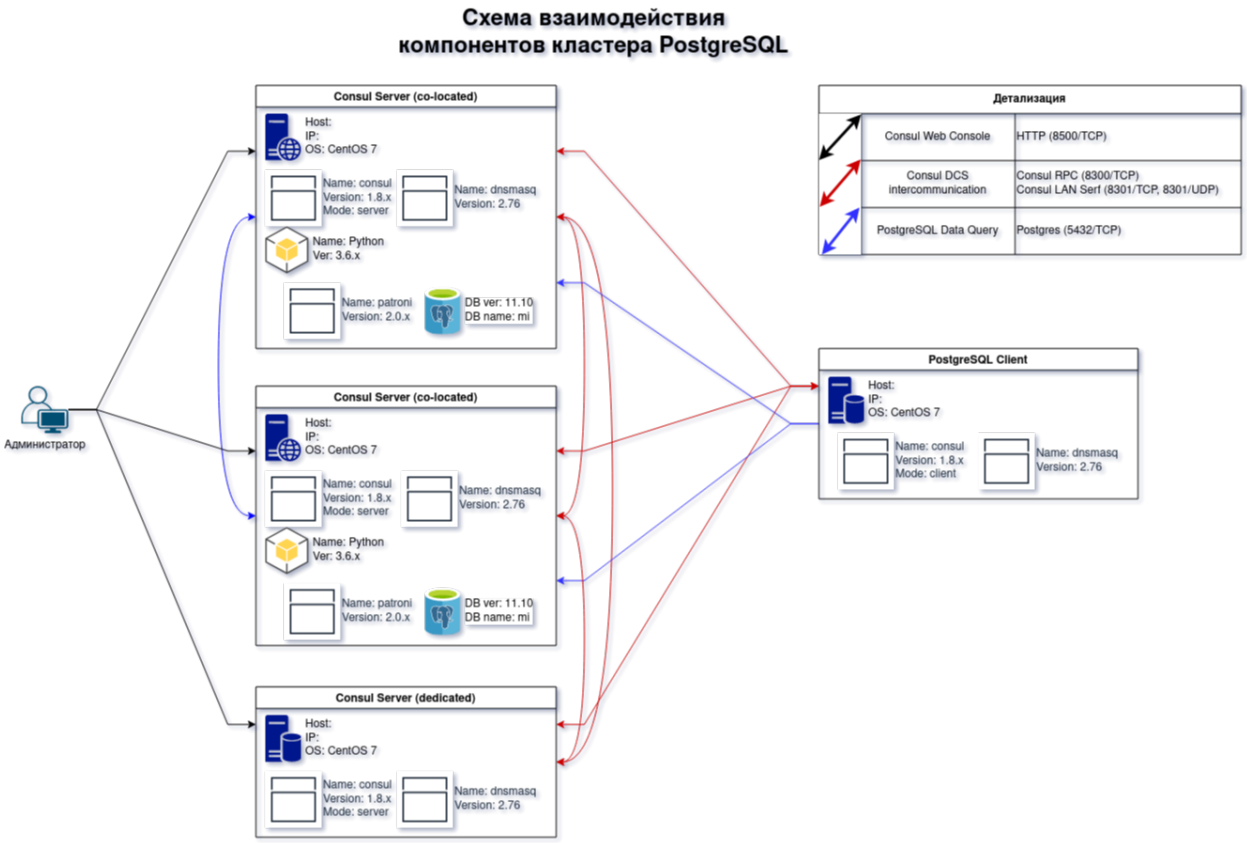


Рис. А.1. **consulcluster.svg**

А.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 асортиментом 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 сектретный token**

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

```
openssl rand -base64 32
```

или, при установке первого узла, запустить команду:

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

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`.

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/lib/journal ]] && \
( sudo mkdir -p /var/lib/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 не 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 сконфигурированно не верно 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 нсастроить регистрацию событий для конкретного сервиса отдельным потоком, со своими ограничениями - Per unit size limit

Приложение С. Использование HAProxy

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.

С.1. HAProxy в роли менеджера пула соединений

Для установки HAProxy необходимо выполнить следующий перечень команд:

```
sudo yum -y 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:

- журналирование 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 кломанду:

```
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 зарегестрирован 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-прохи и плагина для Вашего браузера, например для FireFox FoxyProxy Standard.

Пример подключения:

Настройте расширение:

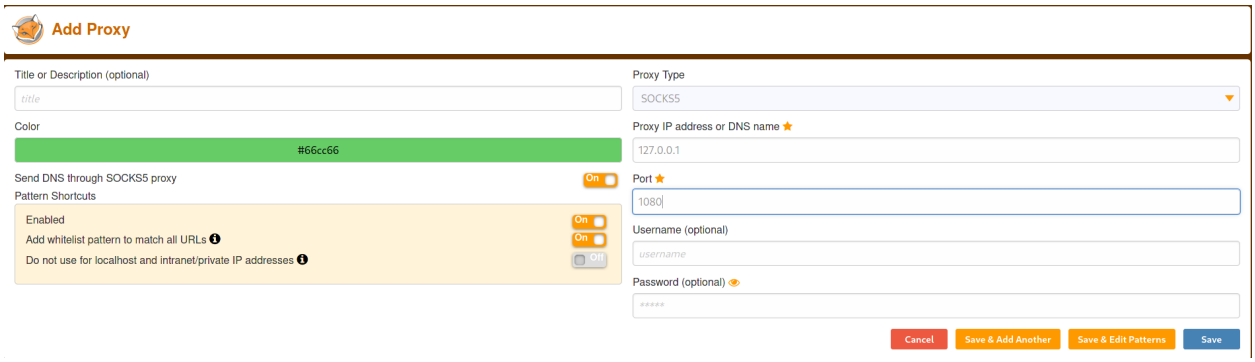


Рис. C.1. **foxyproxy.png**

И обеспечте создание SOCKS-проху через SSH-соединение

```
ssh -D 1080 <Luxms BI host IP/DNS>
```

Как альтернатива, 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. Тюннинг 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 нашрузки 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(смотри 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-conul-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
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:{{ db_rw_port}}
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 {%raw%}{{range service "master.{%endraw%}{{ consul_service }}{%raw%}"}}{%endraw%}
server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}{%endraw%}

### Listener for PostgreSQL REPLICA database
listen db-ro
bind        127.0.0.1:{{ db_ro_port}}
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 {%raw%}{{range service "replica.{%endraw%}{{ consul_service }}{%raw%}"}}{%endraw%}
server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}{%endraw%}
```

`/etc/systemd/system/consul-template.service`

```
[Unit]
Description=Consul Template Service
Documentation=https://github.com/hashicorp/consul-template/
After=network-online.target
Wants=network-online.target

[Service]
```

```
Type=simple
#User=consul
#Group=consul
ExecStart=/usr/sbin/consul-template -config=/etc/consul-template.d/
ExecReload=/bin/kill -HUP $MAINPID
KillSignal=SIGINT
TimeoutStopSec=5
Restart=on-failure
SyslogIdentifier=consul

[Install]
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

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";
```

Чтобы проверить информацию по Kerberos-ключам, сверьте настройки в файле конфигурации со значениями в *keytab*-файле нужно вызвать 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)
```

D.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
```

D.3. 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 **Lucms BI Gateway** использует конфигурасционный файл `/opt/luxmsbi/conf/luxmsbi-gateway.yml`, пример конфигурации:

```
listen: "127.0.0.1:8889"
http-trace: false

## Simple config for LDAP integration (FreeIPA)
# ldap:
#    base: 'cn=accounts,dc=example,dc=tld'
#    host: 'ipa.example.tld'
#    loglevel: debug
#    port: 636
#    usessl: true
#    skipSSLCertVerify: true
#    timeout: 5s
#    binddn: 'uid=bind-user,cn=users,cn=accounts,dc=example,dc=tld'
#    bindpw: 'secret'
#    userfilter: '(uid=%s)'
#    groupfilter: '(member=%s)'
#    useDNForGroupSearch: true
#    attributes:
#      - mail
#      - uid
#    returnAsLogin: uid

## Simple config for MS AD integration (MS Domain Tree)
ad:
base: dc=example,dc=tld
host: dc-01.domainA.example.tld
loglevel: debug
port: 3268
usessl: false
timeout: 5s
binddn: 'bind-user@domainA.example.tld'
bindpw: "secret"
attributes:      -
mail      -
userPrincipalName
```

```
36 returnAsLogin: userPrincipalName
```

После установки компонента и настройки конфигурации `luxmsbi-gateway` необходимо:

Для RPM-based ОС настроить автоматический запуск и запустить сервис:

```
1 sudo systemctl enable luxmsbi-gateway --now
```

Для DEB-based ОС, перезапустить сервис:

```
1 sudo systemctl restart luxmsbi-gateway
```

D.4. Настройка прав в приложении Luxms BI

Настройка распределения прав выполняется прикладным Администратором приложения и не входит в область системного администрирования.

Приложение Е. Настройка SSL

Для высоко-нагруженных 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.

Конфигурация Web-сервера поставляемая для нашего приложения содержит отключенные(комментированные) директивы для подключения SSД-шифрования сессий пользователя.

Е.1. Настройка конфигурации

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;
```

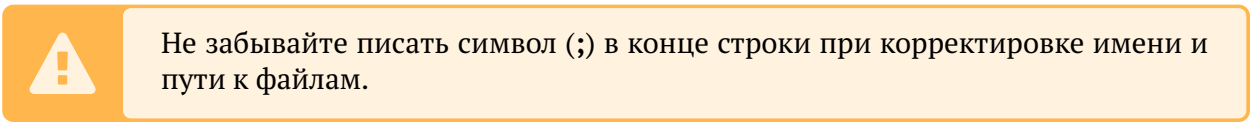
Содержимое конфигурационного файла `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi.sso` тоже подлежит корретировке:

```

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;
ssl_dhparam                     /etc/nginx/ssl/dhparam.pem;
add_header                      Strict-Transport-Security "max-age=31536000; includeSubdomains;";
```

Необходимо откорректировать путь и имена файлов в параметрах конфигурации:

- `ssl_certificate`
- `ssl_certificate_key`



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/nginx/ssl

chown -R bi.bi /opt/luxmsbi/conf/nginx/ssl
chmod 640 /opt/luxmsbi/conf/nginx/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
```