



7634e6a4042e5e1b78be5a7058a9c689cfd473



Руководство системного администратора

УСТАНОВКА

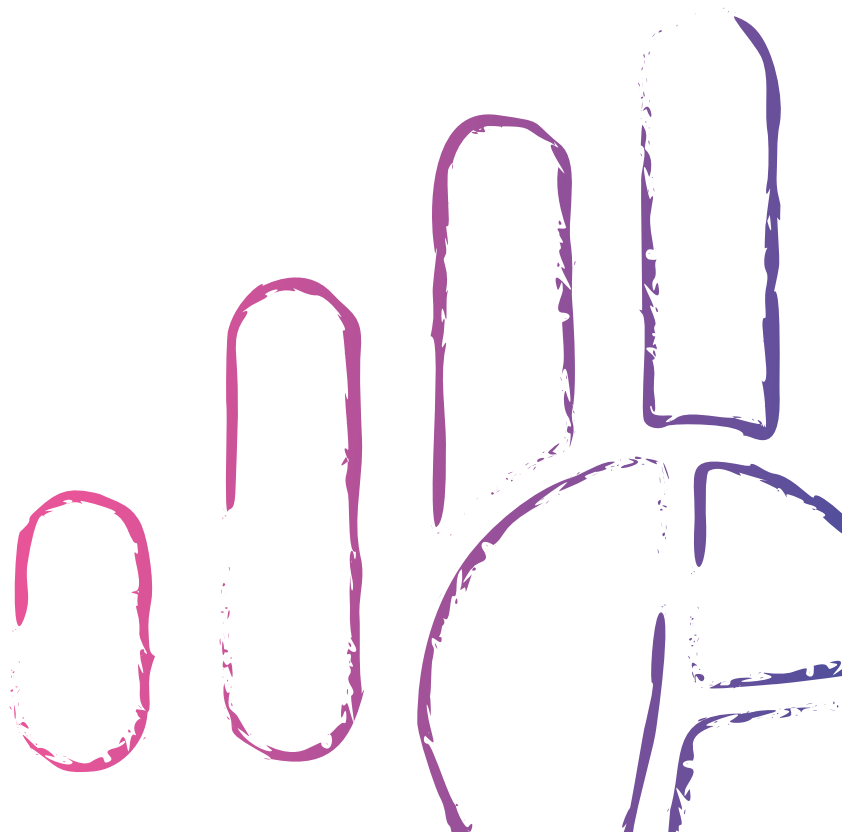
НАСТРОЙКА

ОБНОВЛЕНИЕ

РЕЗЕРВНОЕ КОПИРОВАНИЕ

МОНИТОРИНГ

2023-02-27



Оглавление

Краткое описание	1
1. Описание системы Luxms BI	2
1.1. Слой “Представления”	2
1.2. Слой “Обработки”	2
1.3. Слой Хранения	4
2. Варианты развертывания	5
2.1. Базовая кластерная конфигурация	7
2.2. Расширенная кластерная конфигурация	9
3. Терминология	10
4. Рекомендации по развертыванию Luxms BI	11
4.1. Пакетные репозитории	11
4.2. Запуск компонентов на одном хосте	11
4.2.1. Требования к вычислительным ресурсам	12
4.2.2. Рекомендации по организации файловой системы	12
4.2.3. Пояснения к рекомендациям по файловой системе	12
4.3. Масштабирование сервисов Luxms BI	13
4.3.1. Выделенные сервера Базы данных	14
4.3.2. Выделенные сервера приложений	15
4.3.3. Выделенные сервера для импорта и доступа к данным	15
4.4. Использование SELinux и FirewallD/UFW	16
5. Использование пакетных менеджеров и репозиториях	17
5.1. Обновление корневых сертификатов	17
5.2. Пакетное подключение репозиториях	18
5.3. Ручное подключение репозиториях	18
5.3.1. Подключение к YUM-репозиторию	18
5.3.2. Подключение к DEB-репозиторию	19
5.3.3. Настройка верификации пакетов	20
6. Установка и настройка сервера БД	22
6.1. Установка на RPM-based ОС	22
6.2. Установка на DEB-based ОС	23
6.2.1. Установка PostgreSQL	23
7. Установка компонентов Luxms BI	26
7.1. Развертывание БД Luxms BI	26
7.1.1. Автоматизированная установка БД LuxmsBI	26
7.1.2. Ручная установка базы	27
7.2. Установка KeyDB сервера	28
7.2.1. Обеспечение доступа по сети	28

7.2.2. Резервирование(кластеризация) Redis БД	29
7.3. Развертывание Web приложения	29
7.4. Развертывание BINS	32
7.5. Установка Java Runtime	33
7.6. Установка Luxms BI Appserver	34
7.7. Установка Luxms BI Importer	35
7.8. Установка Luxms BI Datagate	37
7.9. Драйвера JDBC для доступа к данным	38
7.9.1. Подключение дополнительных драйверов	38
7.10. Установка Luxms Databoring	39
7.11. (Только для инсталляций с версией ниже 8.8.x) Установка Luxms Admin	41
7.12. Тюнинг параметров приложений с учетом ресурсов	42
8. Управление компонентами системы Luxms BI	44
8.1. Управление DCS Consul	44
8.2. Настройка параметров БД	45
8.3. Управление кластером Patroni	46
8.4. Управление сервисами приложений	48
8.5. Рекомендации по просмотру журнальных файлов	48
8.5.1. Предоставление прав на просмотр журнала	49
9. Установка обновлений Luxms BI	50
9.1. Установка обновлений компонентов, кроме БД	50
9.1.1. Для RPM-based ОС:	50
9.1.2. Для DEB-based ОС:	50
9.2. Установка обновлений пакета БД luxmsbi-pg	50
9.2.1. Очистка, возврат первоначального состояния БД	51
9.2.2. Обновление БД	51
9.2.3. Обновление БД по требованиям Клиента	52
10. Резервное копирование	53
10.1. Настройка резервного копирования конфигурации	53
10.2. Настройка резервного копирования БД	53
10.2.1. Настройка разрешений доступа к БД	54
10.2.2. Снятие резервной копии	55
10.2.3. Восстановление данных из резервной копии	56
11. Мониторинг компонентов Luxms BI	58
11.1. Мониторинг БД	58
11.2. Мониторинг сервиса Core (luxmsbi-pg)	58
11.3. Мониторинг сервиса App Server (luxmsbi-appserver)	59
11.4. Мониторинг сервиса Luxms BI Importer (luxmsbi-importer)	59
11.5. Мониторинг сервиса Luxms BI Datagate (luxmsbi-datagate)	59
12. Процедура удаления компонентов Luxms BI	60
Приложение А. Установка отказоустойчивой БД	61
А.1. Планирование DCS Consul	62
А.1.1. Типовая схема кластера	62
А.1.2. Планирование DCS Consul	62
А.2. Установка и настройка Consul DCS	63
А.3. Настройка разрешения ресурсов зоны .consul	66
А.3.1. Установка и настройка DNSMasq	67

A.3.2. Дополнительная настройка ОС по разрешению имен	67
A.3.3. Проверка разрешения DNS имен	68
A.4. Установка и настройка Patroni	68
A.4.1. Установка на RPM-based ОС	69
A.4.2. Установка на DEB-based ОС	69
A.4.3. Установка конфигурации Patroni	69
A.4.4. Проверка работоспособности кластера БД	72
A.5. Рекомендации по подключению к БД	72
Приложение В. Настройка журналирования событий	73
В.1. Рекомендации по настройке Journald	73
В.2. Рекомендации по хранению журнальных записей	73
В.3. Проверка текущей конфигурации	74
В.4. Настройка учетных записей для просмотра журналов	75
В.5. Альтернативный вариант для более современных ОС	76
Приложение С. Использование HAProxy	77
С.1. HAProxy в роли менеджера пула соединений	77
С.1.1. Подключение к web-интерфейсу HAProxy для просмотра статистики и управления	79
С.1.2. Тюнинг операционной системы	79
С.2. HAProxy как балансировщик для кластера	80
С.3. Consul-Template. Установка и настройка	80
С.4. HAProxy. Установка и конфигурирование	81
С.4.1. Шаблоны конфигурационных файлов	81
Приложение D. Настройка SSO	85
D.1. Настройка конфигурации Web-сервера	85
D.1.1. Проверка работоспособности Web-сервера	86
D.1.2. Проверка работы модуля SPNEGO	86
D.2. Интеграция с LDAP-каталогами	87
D.2.1. Проверка конфигурации Luxmsbi-gateway	89
D.3. Настройка пользовательских браузеров	90
D.3.1. Internet Explorer:	90
D.3.2. Windows 10 EDGE:	94
D.3.3. Firefox:	96
D.3.4. Yandex & Chrome	97
D.4. Генерация Kerberos-ключей	97
D.4.1. Создание сервисной учетной записи	97
D.4.2. Регистрация Service Principal Name (SPN)	97
D.4.3. Проверка сгенерированных SPN	98
D.4.4. Генерация ключей	99
D.4.5. Установка и проверка работоспособности	100
D.4.6. Настройка NGinx	101
D.5. Настройка прав в приложении Luxms BI	101
Приложение E. Настройка SSL	102
E.1. Настройка конфигурации	102
E.2. Проверка работоспособности	103

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.

Также документ содержит необходимую техническую информацию по обеспечению отказоустойчивости и 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.

Документ не подлежит копированию и/или распространению, а также использованию в целях, отличающихся от прямой цели ее предоставления, без согласия автора и правообладателя — ООО «ЯСП».

1. Описание системы Luxms BI

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

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

1.1. Слой “Представления”

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

- HTTP/HTTPS - 80,443/TCP.

Требует взаимодействия с:

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

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

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 firewall при необходимости доступа с других узлов.

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

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

Требует взаимодействия с:

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

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

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

Требует взаимодействия с:

- KeyDB сервером.
- БД Luxms BI.

4) `luxms-databoring`(NodeJS) - обеспечивает взаимодействие со сторонними источниками данных. Имеет следующие интерфейсы:

- HTTP API - 1880/TCP.

Требует взаимодействия с:

- Компонентом `luxmsbi-importer`,
- Компонентом `luxmsbi-datagate`,
- Компонентом `luxmsbi-web`.

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 импортозамещающего решения дистрибутив для Astra Linux использует российскую СУБД **Postgres Pro Sql** (входит в Единый реестр, имеет сертификат ФСТЭК).

2. Варианты развертывания

В целях обучения и тестирования мы предлагаем поставку Luxms VI в виде Appliance - это виртуальная машина ESXi с установленными компонентами и демонстрационными дата-сетями.

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

При развертывании продуктовых инсталляций мы предлагаем архитектуру развертывания в виде графической схемы. После ее согласования предоставляем Ansible-сценарии для выполнения развертывания Luxms VI.

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

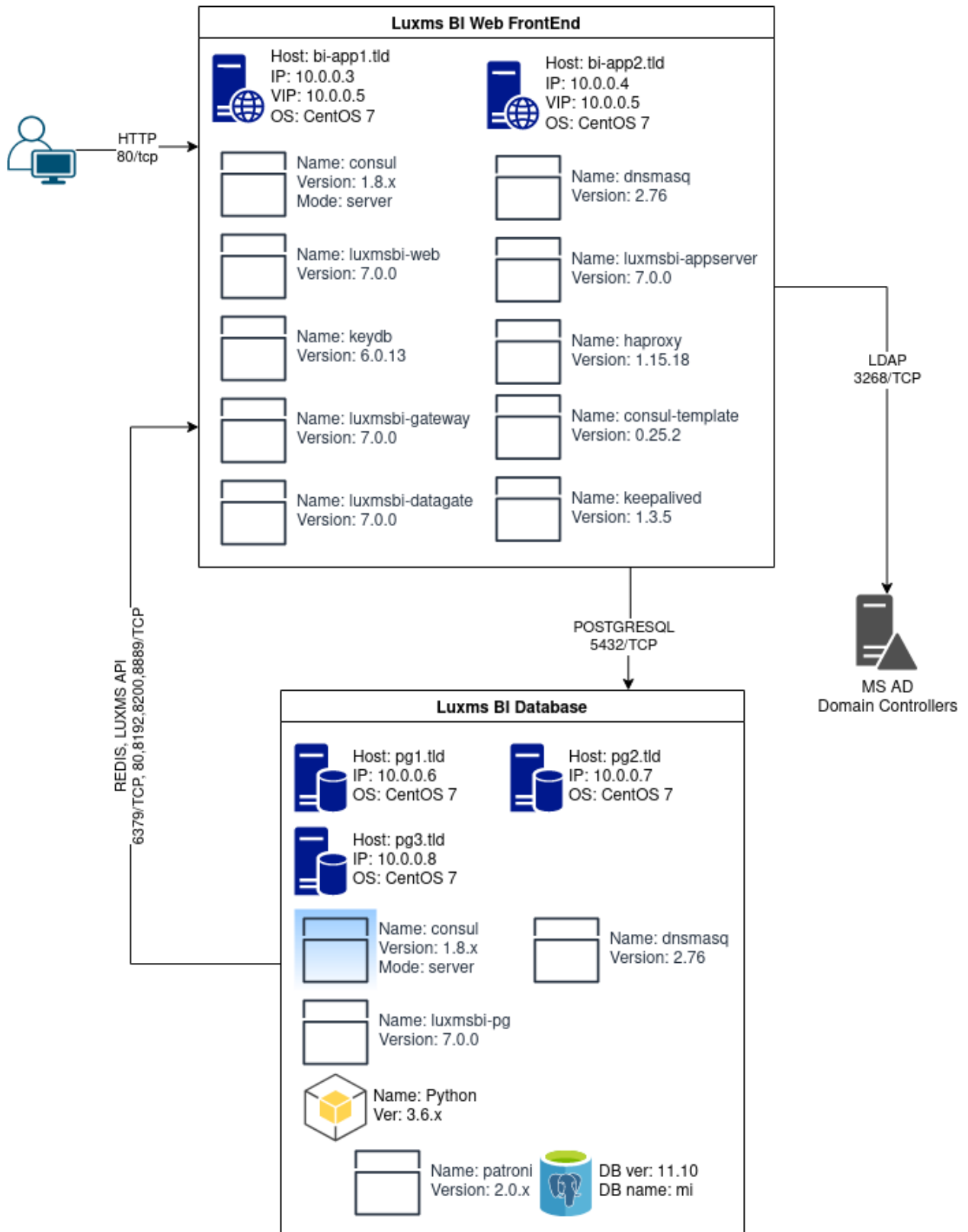
1. Базовая кластерная конфигурация включает в себя резервирование компонентов для каждого слоя системы Luxms VI и обеспечивает работоспособность системы при отказе одного любого сервера. Схема, расположенная ниже, включает в себя резервирование на выделенных узлах:

- Компонентов слоев “Представления” и “Обработки”.
- Слоя “Хранения”.

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

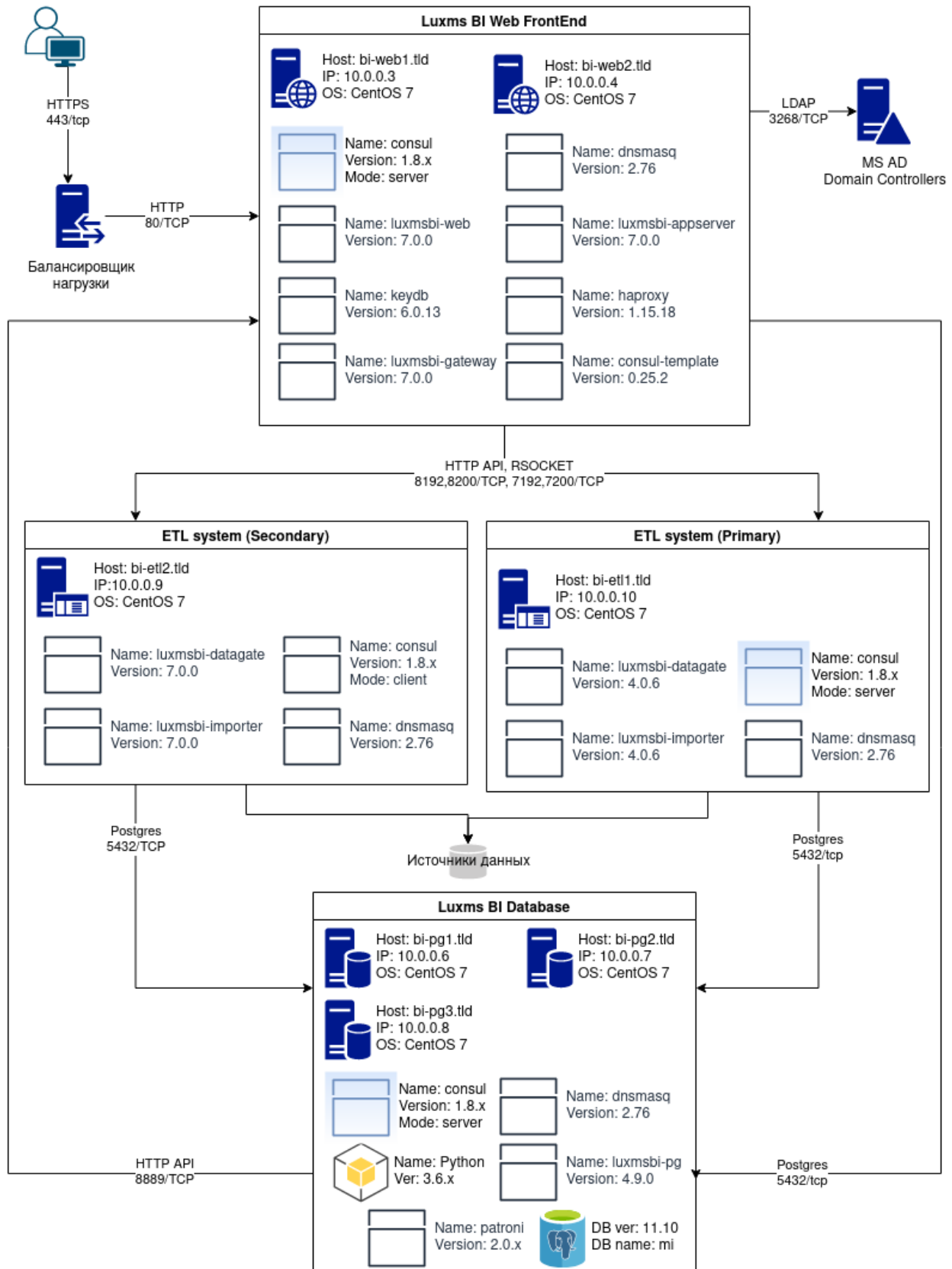
2.1. Базовая кластерная конфигурация

Базовая кластерная конфигурация Схема взаимодействия компонентов Luxms BI



2.2. Расширенная кластерная конфигурация

Расширенная кластерная конфигурация Схема взаимодействия компонентов Luxms BI



3. Терминология

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

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

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

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

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

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

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

Браузер – программа для работы с Web ресурсами.

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

Импорт – операция по добавлению данных или документов в 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, предполагающие 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 безболезненно 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 репозиториях под следующие операционные системы:

- CentOS 7, совместимо с ОС RHEL 7/ OracleLinux 7;
- Astra Linux Special Edition 1.7;
- RedOS Linux 7.3.1.

Доступ к персональным репозиториям предоставляется с использованием аутентификации.

4.2. Запуск компонентов на одном хосте

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

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

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

4.2.1. Требования к вычислительным ресурсам

Рекомендуемые ресурсы для одноузловой системы в среде виртуализации:

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

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

4.2.2. Рекомендации по организации файловой системы

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

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

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

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

4.2.3. Пояснения к рекомендациям по файловой системе

Рекомендуем использовать менеджер логических дисков (LVM).



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

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

1. Файлы базы данных - начните с 10ГБ.

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

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

2. Раздел приложения `/opt/luxmsbi` - начните с 2 Гбайт.

Кроме хранения файлов приложения, данный раздел содержит папку для постоянного хранения отчетных файлов `/opt/luxmsbi/resources`. Размер данного раздела зависит от использования функционала Luxms BI в конкретной инсталляции.

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 инсталляции решений, обрабатывающих значительные объемы данных или большого числа одновременных пользователей. Для использования в условиях высокой нагрузки мы предлагаем разнесение компонент системы на следующие уровни:

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

4.3.1. Выделенные сервера Базы данных

На текущий момент продуктивное решение использует в качестве базы данных:

RedHat-based дистрибутивы:

- **PostgreSQL 11**;
- **PostgreSQL 13** - на стадии тестирования и подготовки миграционной документации.

AstraLinux Special Edition 1,7:

- **PostgrePro 13**;
- **PostgreSQL 13** - на стадии тестирования.

RedOS 7.3.1:

- **PostgrePro 13**.

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

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

Дополнительные рекомендации:



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



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



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

4.3.2. Выделенные сервера приложений

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

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

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

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

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

2. Файлы данных и отчеты - `/opt/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. Поэтому 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 - это решение по фильтрации сетевого трафика поставляется производителем ОС по-умолчанию, поддерживает зоны и синтаксис команд `iptables` при использовании команды:

```
1 firewall-cmd --add-rich-rule ...
```

А в последних выпусках EL команда `iptables-nft` используется как интерфейс для доступа к API `nftables`. Команда `iptables` не работает вообще.

Официальная документация по управлению сетевыми интерфейсами содержит 450 страниц. Из них 40 страниц используются для введения в использование **nftables**. Конечно **nftables** позволяет производить более тонкую настройку сетевой фильтрации, но в 99% случаев этот избыточный функционал Вам не понадобится.

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

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

5. Использование пакетных менеджеров и репозиториев

Установка из компонентов Luxms VI с использованием пакетов облегчает развертывание, обновление и восстановление на предыдущую версию компонентов. Все пакеты компонентов поддерживают версию пакета, состоящую из 3 чисел, и временную метку, содержащую дату сборки пакета. Пакеты программного обеспечения Luxms VI опубликованы в персонализированных репозиториях, использующим аутентификационную учетную запись.

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



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

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

Для RPM-based ОС:

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

Для Deb-based ОС:

```
1 wget http://ftp.ru.debian.org/debian/pool/main/c/ca-certificates/ca-certificates_20211016_all.deb
2 sudo dpkg -i ca-certificates_20211016_all.deb
```

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

```
1 update-ca-certificates
```

5.2. Пакетное подключение репозитариев

Установка “релизного” пакета, опубликованного в вашем персональном репозитории, позволяет вам настроить подключение к репозиториям 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 репозиторию 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`.

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

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

5.3. Ручное подключение репозитариев

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 репозитория и управлению пакетами, например **Spacewalk**, **Sattelite/Katello** или просто локальный репозиторий, созданный утилитой **createrepo**. Для настройки доступа к зеркалу репозитория обратитесь к документации Вашего ПО зеркалирования репозитория.

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

Конфигурационный файл репозитория `/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
5 gpgcheck=0
6 repo_gpgcheck=0
7 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms

9 [luxms-bi]
10 name=Luxms BI Repository
11 baseurl=https://download.luxms.com/repository/[REPO]/8/el/$releasever/$basearch/
12 enabled=1
13 gpgcheck=1
14 repo_gpgcheck=0
15 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Luxms
16 #username=
17 #password=
```

Для окончательной настройки доступа к репозитариям LuxmsBI необходимо:

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



При разворачивании на других RPM-based дистрибутивах необходимо корректно изменить URI с учетом значения переменной `$releasever` для конкретной ОС. Например для Oracle Linux 7, поможет простая замена на статическую версию:

```
baseurl=https://download.luxms.com/repository/[REPO]/8/el/↵
7/$basearch/
baseurl=https://download.luxms.com/repository/thirdparty/↵
8/el/7/$basearch/
```

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 репозитория и управлению пакетами, например **reprepro**. Для настройки доступа к зеркалу репозитория обратитесь к документации вашего ПО зеркалирования репозитория.

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

Конфигурационный файл репозитория `/etc/apt/source.list.d/luxmsbi.list` может содержать одну или несколько строк, выглядит следующим образом:

```

1 # Replace password with yours in links.
2 deb [ arch=amd64 ] https://[customer]:[password]@download.luxms.com/repository/alse-bi8 1.7_x86-64 main

```

где:

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

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

```

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

```

Для окончательной настройки доступа к репозиториям Luxms BI необходимо:

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

Безопасность аутентификационных данных для доступа к репозиториям может быть обеспечена использованием `apt_auth.conf`. Также можно воспользоваться командой `man apt_auth.conf` После настройки Вы можете смело поменять разрешения на конфигурации файлов, содержащие пароли:

```

1 chmod 600 /etc/apt/auth.conf /etc/apt/auth.conf.d/*.conf
2 chmod 700 /etc/apt/auth.conf.d

```



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

5.3.3. Настройка верификации пакетов

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

Для RPM-based ОС:

```

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

```

Для Deb-based ОС:

```

1 sudo wget -q -O - \
2 https://download.luxms.com/repository/thirdparty/RPM-GPG-KEY-Luxms | \

```

```
3 apt-key add -
```

После настройки репозитария рекомендуем обновить локальный кеш пакетов:

```
1 sudo apt update
```

6. Установка и настройка сервера БД

6.1. Установка на RPM-based ОС

Для установки PostgreSQL сервера необходимо подключить дополнительные публичные репозитории или имеющиеся у клиента зеркала этих репозиторий:

- [postgresql.org](https://www.postgresql.org).
- [Extra Packages for Enterprise Linux \(EPEL\)](https://www.fedoraproject.org/epel).

1. Установка дополнительных репозиторий:

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

2. Установка пакетов сервера postgresql-11:

```
1 sudo yum -y install postgresql11 \  
2 postgresql11-server \  
3 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**.

```
1 su - postgres -c '/usr/pgsql-11/bin/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:

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

6.2. Установка на DEB-based ОС

6.2.1. Установка PostgreProSQL

1. Установка необходимых репозитариев:

Должен быть доступным репозиторий Астра Linux, один из двух вариантов в `/etc/apt/sources.list`:

- [Рекомендуется] Подключение к собственному зеркалу репозитория Astra Linux Special Edition 1.7
- Подключение к репозиторию производителя ОС, `/etc/apt/sources.list`:

```
1 deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/ 1.7_x86-64 main,contrib,non-free
2 deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_x86-64 main,contrib,non-free
```

Проверьте наличие подключенных репозитариев PostgresPro и Luxms в ОС. При необходимости подключите недостающий репозиторий:

```
1 # Luxms repository
2 sudo echo "deb https://download.luxms.com/repository/alse-bi8/ 1.7_x86-64 main" \
3 > /etc/apt/sources.list.d/luxmsbi.list
4 wget -q -O - https://download.luxms.com/repository/thirdparty/RPM-GPG-KEY-Luxms | sudo apt-key add -
```

После установки репозитория необходимо добавить данные аутентификации для репозитория Luxms BI в конфигурационный файл `/etc/apt/auth.conf.d/luxmsbi.conf`:



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

```
1 machine download.luxms.com
2 login AstraLinux
3 password CoolPassword
```

В зависимости от используемой БД, подключите необходимый репозиторий:

- [Рекомендовано] PostgresPro Российской сборки

```
1 # PostgresPro 13 Standard repository
2 wget -O - https://repo.postgrespro.ru/pgpro-13/keys/pgpro-repo-add.sh | sh
```

- Оригинальную версию PostgreSQL

```

1 echo "deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg main" > ↵
  /etc/apt/sources.list.d/pgdg.list
2 echo "deb https://deb.debian.org/debian/ buster main contrib non-free" > ↵
  /etc/apt/sources.list.d/debian-buster.list
3 wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-↵
  key add -
4 apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 648ACFD622F3D138

```



Почему мы не рекомендуем Оригинальный PostgreSQL? Он требует подключения репозитория Debian Buster для получения необходимых библиотек. Эта процедура нарушает целостность ОС и ее функционал обеспечения контроля безопасности.

Обновите информацию по доступным пакетам в репозиториях:

```
1 sudo apt update
```

2. Установка PostgrePro и необходимых расширений:

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

3. Инициализация PostgreSQL:

Установка PostgresPro из пакета `postgrespro-std-13` не требует настройки автоматического запуска сервиса (`postgrespro-std-13.service`) и инициализации БД. Но в большинстве продуктовых решений используется, отличное от стандартного, месторасположение файлов БД. Поэтому, для установки выделенного(-ых) серверов БД необходимо установить совокупность пакетов:

```
1 apt -y install postgrespro-std-11-client postgrespro-std-11-server postgrespro-↵
  std-11-contrib
```

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

```
1 su - postgres -c '/opt/pgpro/std-13/bin/initdb -D /var/lib/pgpro/std-13/data'
```



Обращаем внимание, если каталог базы данных отличается от дефолтного, необходимо переопределить переменную окружения **PGDATA** в: - профайле сервисной учетной записи `postgres` - `/var/lib/postgresql/↵`
`.bash_profile` - файле переменных среды сервиса - `/etc/default/↵`
`postgrespro-std-13` `{is.warning}`



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

4. Запуск postgresql сервиса:

Для оригинальной версии PostgreSQL необходимо выполнить следующие действия:

```
1 systemctl enable postgresql-13.service --now
```

Для установки отказоустойчивого кластера PostgreSQL , ознакомьтесь с [Приложением #1](#)

7. Установка компонентов Luxms BI

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

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](#);
- альтернативный вариант Java, зеркало Yandex - [OpenJDK](#)
- Luxms DEB Repo см. [5.3.1 Подключение к YUM-репозиторию](#).

7.1. Развертывание БД Luxms BI



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

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



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

База данных Luxms BI может устанавливаться в автоматизированном режиме либо в ручном.

7.1.1. Автоматизированная установка БД LuxmsBI

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

1. Установка пакета **luxmsbi-pg/luxmsbi-pgpro** при работающей БД выполнит донастройку конфигурационных файлов и создаст БД для системы Luxms BI. Для установки пакета необходимо выполнить команду:

Для RPM-based ОС:


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

Для DEB-based ОС:

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

Автоматическая установка БД из пакета включает в себя:


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

Но только при условии использования базой данных стандартного порта и стандартного расположения файлов БД.

7.1.2. Ручная установка базы

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

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

```
1 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):

```
1 # TYPE DATABASE USER ADDRESS METHOD
3 local all postgres peer
4 local bi all md5
5 host all all 127.0.0.1/32 md5
6 host all all ::1/128 md5
7 local replication all trust
8 host replication all 127.0.0.1/32 trust
9 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 соответствии с документацией PostgreSQL сервера:

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

7.2. Установка KeyDB сервера

Взаимодействия между компонентами Luxms BI построено с использованием быстрой in-memory database KeyDB. 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:

```
1 sudo yum -y install keydb
```

Для DEB-based ОС:

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

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

Для RPM-based ОС:

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

Для DEB-based ОС:

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

3. Необходима настройка автоматического запуска сервера.

Для RPM-based ОС:

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

Для DEB-based ОС:

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

7.2.1. Обеспечение доступа по сети

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

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

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

Необходим перезапуск сервиса. Для RPM-based ОС:

```
1 sudo systemctl restart keydb
```

Для DEB-based ОС:

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

7.2.2. Резервирование(кластеризация) Redis БД

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

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

Необходим перезапуск сервиса. Для RPM-based ОС:

```
1 sudo systemctl restart keydb
```

Для DEB-based ОС:

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

7.3. Развертывание Web приложения

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

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

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



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

Для установки необходимо выполнить следующую последовательность действий:

1. Установить пакет `luxmsbi-web`:

Для RPM-based ОС:

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

Для DEB-based ОС:

```
1 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`:

```
1 return {
2   dbhost="127.0.0.1",
3   dbport=5432,
4   dbname="mi",
5   dbpool="pg-mi-4.0",
6   dbuser="bi",
7   dbpass="bi",
8   dbcompact=false,
9   dbpool_size=10, -- FIXME: NOT YET propogated default is 30 https://github.com/openresty/lua-nginx-module#lua_socket_pool_size
10  dbbacklog=20, -- FIXME: Not yet propogated down to the db:connect!--
11  https://github.com/openresty/lua-nginx-module#tcpsockconnect    needs version 0.10.14
12  debug = true,
13 }
```

3. Выполнить настройку автоматического запуска systemd сервиса:

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

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



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

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

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

```
1 # Defined upstreams
2 upstream binserver {
3     server 127.0.0.1:8888;
4 }
5 upstream appserver {
6     server 127.0.0.1:8080;
7 }
8 upstream datagate {
9     server 127.0.0.1:8200;
10 }
11 upstream importer {
12     server 127.0.0.1:8192;
13 }
14 upstream gateway {
15     server 127.0.0.1:8889;
16 }
17 upstream lua-webapi {
18     server 127.0.0.1:8282;
19     keepalive 8;
20 }
```

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

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



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

```
1 # Defined upstreams
2 upstream binserver {
3     server 127.0.0.1:8888;
4 }
5 upstream appserver {
6     server luxmsbi-appserver.service.consul:8080 resolve;
7 }
8 upstream datagate {
9     server luxmsbi-datagate.service.consul:8200 resolve;
10 }
11 upstream importer {
12     server luxmsbi-importer.service.consul:8192 resolve;
13 }
14 upstream gateway {
15     server luxmsbi-gateway.service.consul:8889 resolve;
16 }
17 upstream lua-webapi {
18     server 127.0.0.1:8282;
19     keepalive 8;
20 }
```

20 }

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



За более подробной инструкцией по настройке обратитесь к документации по модулю NGinx [ngx_http_upstream_module](#)

```
1 # Defined upstreams
2 upstream binserver {
3     server 127.0.0.1:8888;
4 }
5 upstream appserver {
6     server 127.0.0.1:8888;
7 }
8 upstream datagate {
9     server 10.0.0.5:8080;
10    server 10.0.0.6:8080;
11 }
12 upstream importer {
13    server 10.0.0.5:8080;
14    server 10.0.0.6:8080;
15 }
16 upstream gateway {
17    server 127.0.0.1:8889;
18 }
19 upstream lua-webapi {
20    server 127.0.0.1:8282;
21    keepalive 8;
22 }
```

7.4. Развертывание BINS

Компонент `luxmsbi-bins` требует взаимодействия с:

- KeyDB сервером.
- БД Luxms BI.

Установка `luxmsbi-bins` производится совместно с компонентом `luxmsbi-web` и требует установки NodeJS версии 14, который устанавливается как зависимость из репозитория Luxms.

1. Установка BINS:

Для RPM-based ОС:

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

Для DEB-based ОС:

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

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

```
1 {
2 // Get changes from database
3 "db": "postgres://bi:big@127.0.0.1/mi",
4 // Subscribe to changes channel on Redis-server
5 "db.rt": "redis://127.0.0.1"
6 }
```

3. Выполнить настройку автоматического запуска systemd сервиса:

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

7.5. Установка Java Runtime

Установка JRE/JDK для компонентов Luxms BI, реализованных как Java-приложение, не требуется для RPM-based ОС.

Для ОС AstraLinux пакеты Java-приложений могут работать с двумя вариантами JVM:

- по умолчанию, с использованием OpenJDK 11, доступного из репозитория **Extended** ГК Астра;
- с использованием импортозамещающего решения от ООО “БеллСофт”. Для использования сертифицированного ФСТЭК решения **Axiom** JDK необходимо приобретение лицензии или использование пакетов, доступных с американского **репозитория**

Для использования OpenJDK 11, поставляемого производителем ОС необходимо настроить публичный репозиторий (или локальное зеркало репозитория):

```
1 echo -e "\ndeb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-
2 extended/ 1.7_x86-64 main contrib non-free\n" >> /etc/apt/sources.list
apt update
```

Установка OpenJDK 11 после настройки репозитория не требуется, все необходимые пакеты будут установлены как зависимости при установке Java-компонентов.

Для использования импортозамещающего решения требуется дополнительная установка репозитория компании BelSoft и пакета `belsoft-java11`. На примере бесплатной версии, доступной из зарубежного репозитория:

```
1 wget -q -O - https://download.bell-sw.com/pki/GPG-KEY-bellsoft | sudo apt-key
add -
```

```
2 echo "deb [arch=amd64] https://apt.bell-sw.com/ stable main" \  
3 | sudo tee /etc/apt/sources.list.d/bellsoft.list  
4 sudo apt update  
5 sudo apt -y install bellsoft-java13
```

После установки Java-компонентов Luxms BI необходимо исправить конфигурационные файлы Java-компонентов, убрать комментарии с соответствующего значения для `JAVA_HOME`:

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

7.6. 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:

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

Для DEB-based ОС:

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

2. Настройка используемого Java

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

```
1 # Systemd enviroment variables  
  
3 # According to recommended BellSoft Java usage  
4 JAVA_HOME=/usr/lib/jvm/bellsoft-java13-amd64  
5 # Uncomment to Java11-OpenJDK  
6 #JAVA_HOME=/usr/lib/jvm/java-13-openjdk-amd64  
  
8 PATH=/usr/bin:$JAVA_HOME/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin  
  
10 # Options for JVM  
11 # Example, to increase max HEAP size  
12 # JAVA_TOOL_OPTIONS="-Xmx32g"
```



```
13 JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true"
15 # Options for ExecStart
16 OPTIONS="--spring.config.additional-location=↵
/opt/luxmsbi/conf/appserver/application.properties"
```

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

Необходимо настроить подключение к БД в конфигурационном файле:

```
1 # LuxmsBI database properties
2 luxmsbi.datasource.url=jdbc:postgresql://127.0.0.1:5432/mi
3 luxmsbi.datasource.username=bi
4 luxmsbi.datasource.password=bi
```

4. Выполнить настройку автоматического запуска systemd сервиса:

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

5. При необходимости обеспечения SpringBootAdmin дать разрешения на доступ к приложению, при наличии локального firewall в ОС:

Для RPM-based ОС:

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

Для DEB-based ОС:

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

7.7. 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:

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

Для DEB-based ОС:

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

2. Настройка используемого Java

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

```
1 # Systemd enviroment variables
3 # According to recommended BellSoft Java usage
4 JAVA_HOME=/usr/lib/jvm/bellsoft-java13-amd64
5 # Uncomment to Java11-OpenJDK
6 #JAVA_HOME=/usr/lib/jvm/java-13-openjdk-amd64
8 PATH=/usr/bin:$JAVA_HOME/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin
10 # Options for JVM
11 # Example, to increase max HEAP size
12 # JAVA_TOOL_OPTIONS="-Xmx32g"
13 JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true"
15 # Options for ExecStart
16 OPTIONS="--spring.config.additional-location=
```

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

Необходимо настроить подключение к БД в конфигурационном файле:

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

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

Для RPM-based ОС:

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

Для DEB-based ОС:

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

5. Выполнить настройку автоматического запуска systemd сервиса:

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

7.8. Установка Luxms BI Datagate

Компонент `luxmsbi-datagate` требует взаимодействия с:

- KeyDB сервером;
- БД Luxms BI.

1. Установка приложения:

Для RPM-based ОС:

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

Для DEB-based ОС:

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

2. Настройка используемого Java

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

```
1 # Systemd enviroment variables
3 # According to recommended BellSoft Java usage
4 JAVA_HOME=/usr/lib/jvm/bellsoft-java13-amd64
5 # Uncomment to Java11-OpenJDK
6 #JAVA_HOME=/usr/lib/jvm/java-13-openjdk-amd64
8 PATH=/usr/bin:$JAVA_HOME/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin
10 # Options for JVM
11 # Example, to increase max HEAP size
12 # JAVA_TOOL_OPTIONS="-Xmx32g"
13 JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true"
15 # Options for ExecStart
16 OPTIONS="--spring.config.additional-location=↵
   /opt/luxmsbi/conf/appserver/application.properties"
```

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

Необходимо настроить подключение к БД в конфигурационном файле:

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

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

Для RPM-based ОС:

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

Для DEB-based ОС:

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

5. Выполнить настройку автоматического запуска systemd сервиса:

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

7.9. Драйвера JDBC для доступа к данным

На текущий момент(версия 8) Luxms BI поставляется со следующим набором JDBC для доступа к **Источникам данных**:

- PostgreSQL, v42.2.5
- Clickhouse, v0.2.4
- SQLserver, v7.0.0
- Oracle, v12.2.0.1
- Oracle, v19.9.0.0
- Mysql, v5.1.47
- Linter, v1.8
- SQLite, v3.20.1
- Olap4j, v1.1.0
- Hive2, v3.1.2
- Teradata, v16.20.0.13

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

7.9.1. Подключение дополнительных драйверов

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

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

Требования к настройке драйверов JDBC:

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

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

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

- 2) Каждая папка драйвера должна содержать:

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

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

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

Элемент "config" может содержать перечень свойств, поддерживаемых конкретным драйвером, включая свойства конфигурации пула соединений, например:

- max-pool-size;
- timeout;
- и прочие.

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

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

Добавление драйверов требует перезапуска компонентов:

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

7.10. Установка Luxms Databoring

Компонент `luxms-databoring` требует взаимодействия с:

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

1. Установка приложения:

Для RPM-based ОС:

```
1 sudo yum -y install luxms-databoring
```

Для DEB-based ОС:

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

2. Параметры приложения настраиваются в конфигурационном файле сервиса:

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

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

```
1 # RSocket endpoint of Datagate service. In format [IP|DNS]:[PORT].
2 # If empty set default to "127.0.0.1:7200"
3 DATAGATE_HOST=""
4
5 # RSocket endpoint of Importer service. In format [IP|DNS]:[PORT].
6 # If empty set default to "127.0.0.1:7192"
7 IMPORTER_HOST=""
8
9 # Luxms BI API HTTP Proxy In format [IP|DNS]:[PORT].
10 # If empty set default to "http://127.0.0.1/"
11 LUXMSBI_HTTP_API_URL=""
12
13 NODE_OPTIONS=""
```

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

Для RPM-based ОС:

```
1 sudo firewall-cmd --add-port=1880/TCP
2 sudo firewall-cmd --runtime-to-permanent
```

Для DEB-based ОС:

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

4. Выполнить настройку автоматического запуска systemd сервиса:

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

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

```
1 location /databoring/ {
2 proxy_pass http://127.0.0.1:1880/;
3 proxy_http_version 1.1;
4 proxy_read_timeout 7d;
5 proxy_send_timeout 7d;
6 proxy_connect_timeout 7d;
7 proxy_set_header Upgrade $http_upgrade;
8 proxy_set_header Connection $connection_upgrade;
9 proxy_set_header Host $host;
10 }
11 }
```

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

7.11. (Только для инсталляций с версией ниже 8.8.x) 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 инсталляциях.

Требует взаимодействия с:

- Компонентом `luxmsbi-web`.
- БД Luxms BI.

1. Установка приложения:

Для RPM-based ОС:

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

Для DEB-based ОС:

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

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

```
1 package Conf;
3 use strict;
5 use vars qw(%Conf);
7 my $database_db = 'mi';
```

```
8 my $database_user = 'bi';
9 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 соответствии с документацией PostgreSQL [The Password File](#) - `/opt/luxmsbi/.pgpass`:

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

3. Выполнить настройку автоматического запуска systemd сервисов:

Сервис для запуска CGI-приложения:

```
1 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:

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

7.12. Тюнинг параметров приложений с учетом ресурсов

Настройка параметров компонентов зависит от ресурсов аппаратного или виртуального хоста. В первую очередь производится тюнинг параметров БД исходя из ожидаемой нагрузки. Для продуктовой схемы с вышеуказанными параметрами предлагается установка следующих параметров:

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

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

2) После установки параметра необходимо перезапустить узлы БД командой:

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

Во вторую очередь производится тюнинг параметров Web-приложения `luxmsbi-web-portal`. Для продуктовой схемы рекомендуется выставить следующие параметры:

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

worker_processes 8;

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

dbpool_size=20,

В третью очередь определяются параметры определяющие объем используемой оперативной памяти компонентами и размеры пула соединений к БД. Для продуктовой схемы рекомендуется выставить следующие параметры:

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

luxmsbi.datasource.max-pool-size=10

- 2) В конфигурационном файле /etc/default/luxmsbi-datagate

```
1 # Options for JVM
2 JAVA_TOOL_OPTIONS="-Djava.net.preferIPv4Stack=true -Xmx=16G"
```

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

luxmsbi.datasource.max-pool-size=25

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

luxmsbi.datasource.max-pool-size=5

8. Управление компонентами системы Luxms BI

8.1. 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.

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

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

Кроме управления из командной строки, Consul предоставляет функционал управления:

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

8.2. Настройка параметров БД

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

```
1  --
2  Generated by PGConfig 2.0 beta----
3  http://pgconfig.org--
4
5  Memory Configuration
6  ALTER SYSTEM SET shared_buffers TO '1GB';
7  ALTER SYSTEM SET effective_cache_size TO '3GB';
8  ALTER SYSTEM SET work_mem TO '20MB';
9  ALTER SYSTEM SET maintenance_work_mem TO '512MB';--
10
11 Checkpoint Related Configuration
12 ALTER SYSTEM SET min_wal_size TO '2GB';
13 ALTER SYSTEM SET max_wal_size TO '6GB';
14 ALTER SYSTEM SET checkpoint_completion_target TO '0.9';
15 ALTER SYSTEM SET wal_buffers TO '16MB';--
16
17 Network Related Configuration
18 ALTER SYSTEM SET listen_addresses TO '*';
19 ALTER SYSTEM SET max_connections TO '150';--
20
21 Storage Configuration
22 ALTER SYSTEM SET random_page_cost TO '4.0';
23 ALTER SYSTEM SET effective_io_concurrency TO '2';--
24
25 Worker Processes
26 ALTER SYSTEM SET max_worker_processes TO '2';
27 ALTER SYSTEM SET max_parallel_workers_per_gather TO '1';
28 ALTER SYSTEM SET max_parallel_workers TO '2';
```

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

- 1) Используйте для изменения конфигурации сервера команды `ALTER SYSTEM`, это позволяет избежать ошибок при редактировании `postgresql.conf`. При этом конфигурационные параметры применяются при каждом рестарте экземпляров БД из конфигурационного файла `postgresql.auto.conf`.
- 2) Для генерации конфигурационных команд можно использовать любой калькулятор, мы рекомендуем [PGConfig](#).

8.3. Управление кластером Patroni

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

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

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

```
1 journalctl -u patroni
```

Часто используемые опции утилиты `journalctl` вы можете найти в [этом документе](#)

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

```
1 patronictl --help
2 Usage: patronictl [OPTIONS] COMMAND [ARGS]...
3
4 Options: -
5 c, --config-file TEXT  Configuration file -
6 d, --dcs TEXT           Use this DCS -
7 k, --insecure          Allow connections to SSL sites without certs --
8 help                  Show this message and exit.
9
10 Commands:
11 configure              Create configuration file
12 dsn                    Generate a dsn for the provided member, defaults to a dsn of...
13 edit-config            Edit cluster configuration
14 failover               Failover to a replica
15 flush                  Discard scheduled events
16 history                Show the history of failovers/switchovers
17 list                   List the Patroni members for a given Patroni
18 pause                  Disable auto failover
19 query                  Query a Patroni PostgreSQL member
20 reinit                 Reinitialize cluster member
21 reload                 Reload cluster member configuration
22 remove                 Remove cluster from DCS
23 restart                Restart cluster member
24 resume                 Resume auto failover
25 scaffold               Create a structure for the cluster in DCS
26 show-config            Show cluster configuration
27 switchover             Switchover to a replica
28 topology               Prints ASCII topology for given cluster
29 version                Output version of patronictl command or a running Patroni...
```

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

1. Проверка статуса узлов кластера:

```
1 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:

```
1 [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

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

```
1 Are you sure you want to failover cluster postgresdb, demoting current master ↩
  centos-1.local? [y/N]: y
2 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

8.4. Управление сервисами приложений

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

```
1 systemctl enable <component-name>
2 systemctl start <component-name>
3 systemctl restart <component-name>
4 systemctl stop <component-name>
5 systemctl disable <component-name>
```

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

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

8.5. Рекомендации по просмотру журнальных файлов

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

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

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

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

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

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

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

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

```
1 journalctl -u luxmsbi-importer --no-pager
2 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:

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

8.5.1. 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:

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

9. Установка обновлений Luxms BI

9.1. Установка обновлений компонентов, кроме БД

9.1.1. Для RPM-based ОС:

Установка обновлений компонентов Luxms BI производится обновлением пакетов:

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

При необходимости отката на предыдущую версию компонента используйте команду:

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

9.1.2. Для DEB-based ОС:

Установка обновлений компонентов Luxms BI производится обновлением пакетов:

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

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

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

9.2. 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 соответствующий 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:

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

9.2.1. Очистка, возврат первоначального состояния БД

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

```
1 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)`.

9.2.2. Обновление БД



До начала процесса обновления БД рекомендуем снять резервную копию БД, смотрите раздел [Резервное копирование](#).

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

- 1) Кумулятивная установка - установка всех необходимых обновлений:

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

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

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

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

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

9.2.3. Обновление БД по требованиям Клиента

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

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

10. Резервное копирование

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

- Конфигурационные файлы компонентов.
- Данные БД.

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

10.1. Настройка резервного копирования конфигурации

Конфигурация компонентов Luxms BI консолидирована в папке `/opt/luxmsbi/conf` файловой системы. Изменение 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.

10.2. Настройка резервного копирования БД



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

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



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


10.2.1. Настройка разрешений доступа к БД

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

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

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

```

1 postgresql:
2 pgpass: /pgdata/.pgpass
3 listen: 0.0.0.0:5432
4 connect_address: "172.16.32.112:5432"
5 data_dir: /pgdata/data
6 bin_dir: /usr/pgsql-11/bin/
7 pg_rewind:
8 username: postgres
9 password: "password"
10 pg_hba: -
11 local all postgres peer -
12 host all all 0.0.0.0/0 md5 -
13 host replication replicator 127.0.0.1/32 md5 -
14 host replication replicator 172.16.32.112/32 md5 -
15 host replication replicator 172.16.32.113/32 md5 -
16 host replication replicator 172.16.32.155/32 md5 <--- добавить сюда хост, с 
    которого будут делаться бэкап
18 replication:
19 username: replicator
20 password: "password"
21 superuser:
22 username: postgres
23 password: "password"

```

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

```
1 [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

```

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

```

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

```

1 [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

```

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

```

10.2.2. Снятие резервной копии

Для получения резервной копии БД необходимо выполнить команду:

```

1 [root@localhost backup]# pg_basebackup \-
2 d postgresql://replicator:password@172.16.32.113 \--
3 checkpoint=fast \-
4 D /backup \-
5 P -Ft -z -Xs
7 [root@localhost backup]# ls -l
8 total 11848-
9 rw----- . 1 root root 12110482 Mar 15 15:50 base.tar.gz-
10 rw----- . 1 root root 18318 Mar 15 15:50 pg_wal.tar.gz

```

Используются следующие ключи:

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



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

Полученные файлы должны перемещаться на устройства долговременного хранения.

10.2.3. Восстановление данных из резервной копии

Останавливаем весь кластер PostgreSQL. Нужно выполнить на каждом хосте:

```
1 systemctl stop patroni
```

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

```
1 [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 команды:

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

Пробуем запустить хост с базой:

```
1 systemctl start patroni
```

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

```
1 2021-03-15 15:30:03.727 MSK [14685] LOG: archive recovery complete
2 2021-03-15 15:30:03.732 MSK [14682] LOG: database system is ready to accept
connections
```

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

Проверяем, все ли работает правильно:

```
1 [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	
db-main	bi-pg2	172.16.32.113		running	3	109



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

```
1 [root@bi-pg2 ~]# patronictl -c /etc/patroni/patroni.yml reinit db-main
```

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

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

11. Мониторинг компонентов Luxms BI

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

Перечень элементов мониторинга включает в себя:

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

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

Дополнительно рекомендуется организовать мониторинг содержимого журнальных файлов.

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

11.1. Мониторинг БД

Мониторинг не резервируемого сервера БД должен включать в себя:

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

11.2. Мониторинг сервиса Core (luxmsbi-pg)

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



Доступно, начиная с версии luxmsbi-pg-8.8.11

11.3. Мониторинг сервиса App Server (luxmsbi-appserver)

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

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

- HTTP API port: 8192;
- URI /actuator/health (отправка запроса с localhost);
- Тип запроса GET;
- Ожидаемый HTTP статус ответа: 200;
- Ожидаемый ответ (JSON): {"status": "UP"}.

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

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



12. Процедура удаления компонентов Luxms BI

Приложение А. Установка отказоустойчивой БД

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

- Hashicorp Consul DCS - в роле арбитра для кластерных 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-х узлов, для запуска экземпляра БД.

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

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

Или используйте сторонние утилиты, например, `openssl`:

```
1 openssl rand -base64 32
```

Установку Consul DCS рекомендуем выполнять из репозитория компании HashiCorp, детальная информация по подключению репозитория для различных операционных систем описана на [сайте](#) производителя.

A.1. Планирование DCS Consul

A.1.1. Типовая схема кластера

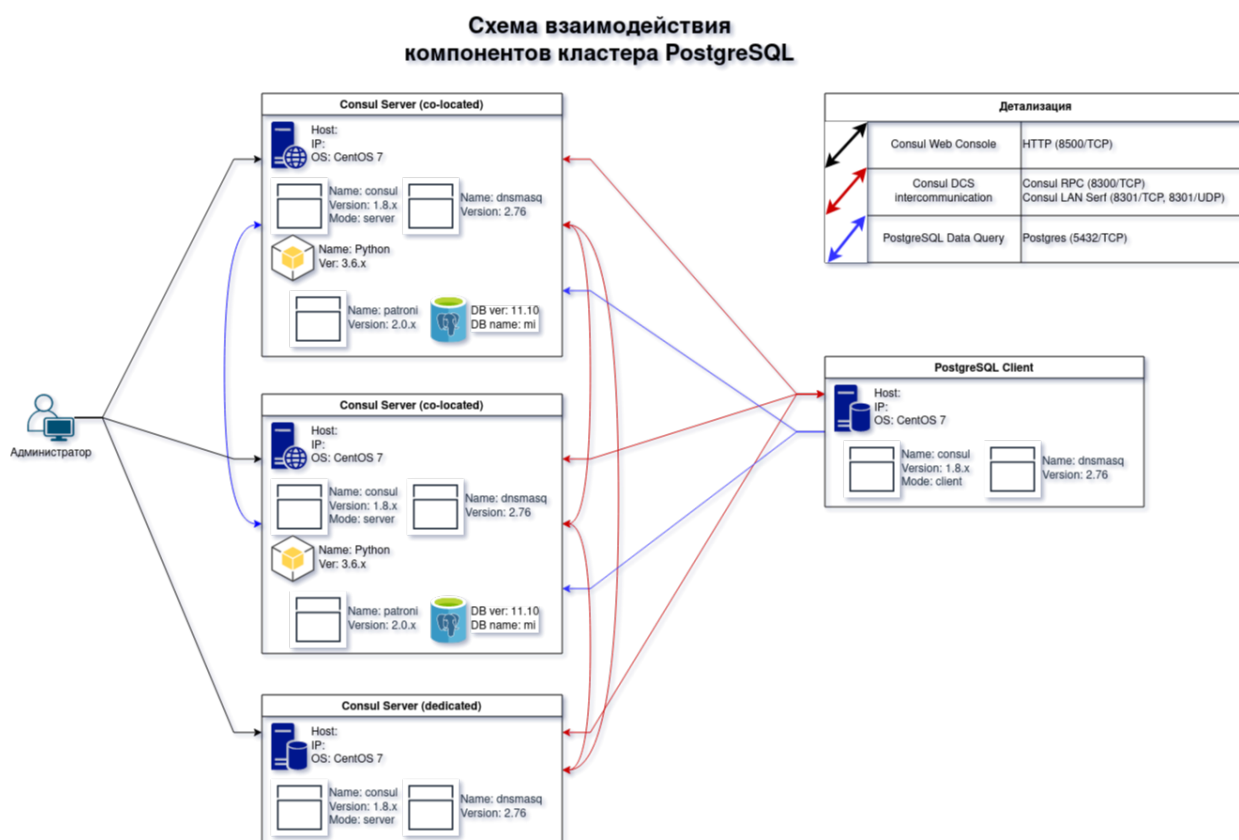


Рис. А.1. Схема взаимодействия компонентов кластера PostgreSQL

A.1.2. Планирование DCS Consul

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



Необходимость установки **Consul** на все сервера программного комплекса обусловлена использованием его **DNS**-интерфейса для настройки доступа к БД в приложениях. Альтернативное решение - настройка `/etc/resolv.conf` может привести к значительному снижению времени отклика приложений, в случае выхода/вывода из рабочего режима одного из настроенных серверов имен.

Для начала установки необходимо:

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

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

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

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

```
1 openssl rand -base64 32
```

Или при установке первого узла запустить команду:

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

A.2. Установка и настройка Consul DCS

1. Настройка репозитория и установка:

Для RPM-based ОС:

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

Для DEB-based ОС:

```
1 curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
2 sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"
3 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 предварительно сгенерированный **секретный токен**:

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

```
1 {
2   "datacenter": "luxmsbi",
3   "data_dir": "/opt/consul",
4   "bind_addr": "0.0.0.0",
```

```
5 "client_addr": "0.0.0.0",
6 "domain": "consul",
7 "enable_script_checks": true,
8 "dns_config": {
9   "enable_truncate": true,
10  "only_passing": true
11 },
12 "recursors": [ "127.0.0.1" ],
13 "enable_syslog": true,
14 "encrypt": "5wDTh+YLWG5DTDDfEeWkQ1j9J72+aJ3NOavqTRaLLUA=",
15 "leave_on_terminate": true,
16 "log_level": "INFO",
17 "rejoin_after_leave": true,
18 "retry_join": [
19   "centos-1.local",
20   "centos-2.local",
21   "centos-3.local" ],
22
23 "server": true,
24 "bootstrap_expect": 3,
25 "ui": true
26 }
```

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

```
1 {
2   "datacenter": "luxmsbi",
3   "data_dir": "/opt/consul",
4   "bind_addr": "0.0.0.0",
5   "client_addr": "0.0.0.0",
6   "domain": "consul",
7   "enable_script_checks": true,
8   "dns_config": {
9     "enable_truncate": true,
10    "only_passing": true
11  },
12  "recursors": [ "127.0.0.1" ],
13  "enable_syslog": true,
14  "encrypt": "5wDTh+YLWG5DTDDfEeWkQ1j9J72+aJ3NOavqTRaLLUA=",
15  "leave_on_terminate": true,
16  "log_level": "INFO",
17  "rejoin_after_leave": true,
18  "retry_join": [
19    "centos-1.local",
20    "centos-2.local",
21    "centos-3.local" ]
22
23 }
```

- **bind_addr** — адрес, на котором будет слушать наш сервер Consul. Это может быть IP любого из наших сетевых интерфейсов или, как в данном примере, все.
- **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 завершится успешно.
- **server** — режим работы сервера.
- **start_join** — список узлов кластера, к которым пробуем присоединиться при загрузке сервера.
- **ui_config** — конфигурация для графического веб-интерфейса.

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

```

1 # /usr/bin/consul validate /etc/consul.d/config.hcl
3 ...
4 Configuration is valid!
```

3. Настройка разрешений локального firewall:

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

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <service>
3 <short>consul</short>
4 <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>
5 <port protocol="tcp" port="8300"/>
6 <port protocol="tcp" port="8301"/>
7 <port protocol="tcp" port="8302"/>
8 <port protocol="tcp" port="8500"/>
9 <port protocol="tcp" port="8501"/>
```

```
10 <port protocol="tcp" port="8502" />
11 <port protocol="tcp" port="8600" />
13 <port protocol="udp" port="8301" />
14 <port protocol="udp" port="8302" />
15 <port protocol="udp" port="8600" />
16 </service>
```

Для добавление сервиса в конфигурацию Firewalld выполните следующую команду:

```
1 sudo firewall-cmd --permanent --new-service-from-file=consul.xml
2 sudo firewall-cmd --reload
3 sudo firewall-cmd --add-service=consul
4 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

```
1 [Consul]
2 title=HashiCorp Consul DCS
3 description=Consul makes it simple for services to register themselves and to
  discover other services via a DNS or HTTP interface. https://www.consul.io/docs/install/ports.html.
4 ports=8300:8302,8500:8502,8600/tcp|8301:8302,8600/udp
```

Для добавление application в конфигурацию UFW выполните следующую команду:

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

4. Запуск DCS Consul и проверка работоспособности:

Запускаем сервис Consul-а на всех узлах:

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

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

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

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

A.3. Настройка разрешения ресурсов зоны .consul

Для обеспечения актуальности сведений о зарегистрированных сервисах Consul DNS-интерфейс разрешает имена без кэширования с TTL=0. Consul позволяет перенаправлять запросы на другие DNS-сервера, но более оптимальное решение - использование интеграции Consul и Dnsmasq для разрешения всех запросов.

A.3.1. Установка и настройка DNSMasq

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

Для RPM-based ОС:

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

Для DEB-based ОС:

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

После установки пакета необходимо откорректировать файл конфигурации или установить следующую минимальную конфигурацию:

Минимальный конфигурационный файл, `/etc/dnsmasq.conf`

```
1 # Configuration file for dnsmasq.
2 #
3 # Format is one option per line, legal options are the same
4 # as the long options legal on the command line. See
5 # "/usr/sbin/dnsmasq --help" or "man 8 dnsmasq" for details.
6
7 # Never forward plain names (without a dot or domain part)
8 domain-needed
9
10 # Add other name servers here, with domain specs if they are for
11 # non-public domains.
12 server=/consul/127.0.0.1#8600
13
14 # If you don't want dnsmasq to read /etc/hosts, uncomment the
15 # following line.
16 no-hosts
17
18 # Include all files in /etc/dnsmasq.d except RPM backup files
19 conf-dir=/etc/dnsmasq.d,.rpmnew,.rpmsave,.rpmorig
```

Разрешаем и запускаем сервис dnsmasq:

```
1 sudo systemctl enable dnsmasq --now
```

A.3.2. 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`:

```
1 prepend domain-name-servers 127.0.0.1;
```

Перезапускаем сервис сети для применения настроек:

```
1 sudo systemctl restart NetworkManager
```

A.3.3. Проверка разрешения DNS имен

Разрешение имен сервисов производится запросом:

- для разрешения узлов

```
1 <node>.node.<partition>.ap.<datacenter>.dc.<domain>
```

- для разрешения сервисов

```
1 [<tag>.<service>.service[.<datacenter>].<domain>
```

Для разрешения имен сервисов в Consul DNS нет необходимости указывать имя, указанное в параметре конфигурации Consul-агента - `datacenter`. Если этот параметр отсутствует, то предполагается текущее значение у самого Consul-агента. т.е. следующие запросы вернут корректный IP-адрес:

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

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

```
1 nslookup consul.service.consul
2 ...
3 Name: consul.service.consul
4 Address: 10.0.2.5
5 Name: consul.service.consul
6 Address: 10.0.2.7
7 Name: consul.service.consul
8 Address: 10.0.2.6
```

A.4. Установка и настройка Patroni

До установки Patroni необходимо установить пакеты БД и необходимых расширений в соответствии 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 Package Index](#) или к его локальному зеркалу.

A.4.1. Установка на RPM-based ОС

При установке модуля `patroni` большое значение имеет 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:

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

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

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

A.4.2. 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:

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

A.4.3. 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 соответствующие вашей инсталляции в файле конфигурации `/etc/patroni/patroni.yml` (YAML):

- `name` - имя узла, рекомендуется установить DNS-имя узла.

- `restapi.connect_address` - 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`

```
1 name: centos-2.local
2 scope: postgresdb
3
4 watchdog:
5 mode: off
6
7 consul:
8 host: "localhost:8500"
9 register_service: true
10 ## token: <ACL-token if used>
11
12 restapi:
13 listen: 0.0.0.0:8008
14 connect_address: "10.0.2.4:8008"
15 authenticate: 'username:password'
16
17 bootstrap:
18 dcs:
19 ttl: 30
20 loop_wait: 10
21 maximum_lag_on_failover: 1048576 # 1 megabyte in bytes
22 postgresql:
23 use_pg_rewind: true
24 use_slots: true
25 parameters:
26 archive_mode: "off"
27 wal_level: hot_standby
28 max_wal_senders: 10
29 wal_keep_segments: 8
30 max_replication_slots: 5
31 hot_standby: "on"
32 wal_log_hints: "on"
```

```
34 initdb: -
35 encoding: UTF8 -
36 data-checksums

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

44 postgresql:
45 pgpass: /var/lib/postgresql/.pgpass
46 listen: 0.0.0.0:5432
47 connect_address: "10.0.2.4:5432"
48 data_dir: /var/lib/postgresql/data
49 bin_dir: /usr/pgsql-11/bin/
50 pg_rewind:
51 username: postgres
52 password: password
53 pg_hba: -
54 local all all trust -
55 host replication replicator 10.0.2.0/24 md5 -
56 host replication replicator 127.0.0.1/32 trust -
57 host all all 0.0.0.0/0 md5
58 replication:
59 username: replicator
60 password: password
61 superuser:
62 username: postgres
63 password: password
```

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

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

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

6 [Service]
7 Type=simple
8 User=postgres
9 Group=postgres
10 ExecStart=/usr/local/bin/patroni /etc/patroni/patroni.yml
11 KillMode=process
12 TimeoutSec=30
13 Restart=no

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

После создания сервисного файла необходимо настроить его автоматический запуск и запустить его:

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

А.4.4. Проверка работоспособности кластера БД

После запуска Patroni DNS-интерфейс Consul разрешает запросы для сервиса PostgreSQL, например:

```
1 [root@centos-4 ~]# dig master.postgresdb.service.consul +short
2 10.0.2.5
3 [root@centos-4 ~]# dig replica.postgresdb.service.consul +short
4 10.0.2.4
5 [root@centos-4 ~]# dig replica.postgresdb.service.consul SRV +short
6 1 1 5432 centos-2.local.node.dc0.consul.
```

Проверка статуса узлов кластера:

```
1 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

А.5. 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 ознакомьтесь с [Приложением 3](#)

Приложение В. Настройка журналирования событий

В.1. Рекомендации по настройке Journald

Одно из преимуществ Journald - возможность ограничивать поток сообщений для хранения. Этот механизм защищает файловую систему сервера от переполнения, т.е. обеспечивает работоспособность. Иногда есть необходимость корректировки конфигурации по умолчанию для параметров `RateLimitInterval` и `RateLimitBurst`.

Хранение журнальных записей производится в binary-формате, что позволяет существенно снизить объем и обеспечить высокую скорость доступа. Расположение журнальных файлов зависит от параметра конфигурации **Storage**, но фактически это 2 варианта.

1. Хранение в памяти - `/run/log/journal`:

- `Storage=auto` при отсутствии директории `/var/log/journal` (по умолчанию)
- `Storage=volatile`

2. Хранение в файловой системе - `/var/log/journal`:

- `Storage=auto` при существовании директории `/var/log/journal`
- `Storage=persistent`

В.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`:

```
1 #Storage=auto
2 #Compress=yes
3 #Seal=yes
4 #SplitMode=uid
5 #SyncIntervalSec=5m
6 RateLimitInterval=1
7 RateLimitBurst=10000
8 #SystemMaxUse=
9 SystemKeepFree=20
10 #SystemMaxFileSize=
11 #RuntimeMaxUse=
12 #RuntimeKeepFree=
13 #RuntimeMaxFileSize=
14 #MaxRetentionSec=
15 #MaxFileSec=1month
16 #ForwardToSyslog=yes
17 #ForwardToKMsg=no
18 #ForwardToConsole=no
19 #ForwardToWall=yes
20 #TTYPath=/dev/console
21 #MaxLevelStore=debug
22 #MaxLevelSyslog=debug
23 #MaxLevelKMsg=notice
24 #MaxLevelConsole=info
25 #MaxLevelWall=emerg
26 #LineMax=48K
```

4. Проверить существование папки в файловой системе, при необходимости создать и выполнить перезапуск сервиса журнальных файлов:

```
1 [[ -d /var/log/journal ]] && \
2 ( sudo mkdir -p /var/log/journal && sudo systemctl restart systemd-journald)
```

В.3. Проверка текущей конфигурации

1. Выполняем проверку режима работы Journald:

```
1 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:

```
1 systemctl status systemd-journald
2 systemd-journald.service - Journal Service
3 Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static; ↵
4         vendor preset: disabled)
5 Active: active (running) since Thu 2020-11-19 21:01:43 MSK; 18s ago
6 Docs: man:systemd-journald.service(8)
```



```
6 man:journald.conf(5)
7 Main PID: 2671 (systemd-journal)
8 Status: "Processing requests..."
9 CGroup: /system.slice/systemd-journald.service
10 2671 /usr/lib/systemd/systemd-journald
12 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).
13 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 сохраняются.

2. Проверка доступного дискового пространства для хранения журнальных записей:

```
1 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/journal.conf` или о том, что приложение сконфигурировано неверно в части журналирования событий.

```
1 journalctl -u systemd-journald--
3 Logs begin at Thu 2019-05-16 13:56:01 MSK, end at Thu 2020-11-19 22:42:04 MSK. -
4 Oct 28 08:26:01 rzd-skimn-d-app-1 systemd-journal[1580]: Suppressed 7894 messages from /system.slice/luxmsbi_appserver.service
5 Oct 28 08:27:01 rzd-skimn-d-app-1 systemd-journal[1580]: Suppressed 7894 messages from /system.slice/luxmsbi_appserver.service
```

В.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**:

```
1 usermod -aG systemd-journal username
```

2) Для просмотра журнальных записей в файлах:

```
1 usermod -aG bi username
```

В.5. Альтернативный вариант для более современных ОС

Операционные системы RHEL-based 7 (RedHat/CentOS/Oracle) Linux использует Systemd версии 219, которая не поддерживает расширенный функционал управления журнальными файлами. Начиная с версии Systemd 231, Journald поддерживает разделение потоков регистрации журнальных записей.

Например, для ОС CentOS 8 возможно настроить регистрацию событий для конкретного сервиса отдельным потоком, со своими ограничениями - **Per unit size limit**

Приложение 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 настройкам 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:

```
1 sudo yum -y install haproxy
2 sudo setsebool -P haproxy_connect_any=1
3 cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.default
```

Ниже расположен конфигурационный файл HAProxy, включающий в себя настройки для:

- Журналирования событий балансировщика.
- Предоставления web-интерфейса для управления и просмотра статистики по балансировщику нагрузки.
- Обеспечение очереди для запросов к БД PostgreSQL.

Замените содержимое конфигурационного файла `/etc/haproxy/haproxy.cfg`, тем более что предыдущие команды создали его резервную копию:

```
1 global
2 daemon
```

```
3 user haproxy
4 group haproxy
5 pidfile /var/run/haproxy.pid
6 log /dev/log local0
7 maxconn 102400

9 defaults
10 log global
11 mode tcp
12 retries 2
13 timeout client 30m
14 timeout connect 4s
15 timeout server 30m
16 timeout check 5s

18 listen stats
19 bind 127.0.0.1:2000
20 maxconn 100
21 mode http
22 option httplog
23 stats uri /stats
24 stats enable
25 stats refresh 1s
26 stats admin if LOCALHOST

28 listen postgres
29 bind *:5432
30 maxconn 10240
31 timeout queue 30s
32 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:

```
1 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:

```
1 su - postgres -c '/usr/pgsql11/bin/psql "ALTER SYSTEM SET PORT TO 9898;"'
2 sudo systemctl restart postgresql-11 haproxy
```

С этого момента система Luxms BI будет использовать HAProxy как менеджер пула соединений к БД PostgreSQL.



Порт 9898/TCP уже зарегистрирован в SELinux как `postgresql_port_t`, поэтому дополнительных настроек безопасности не требуется.

Не забудьте добавить в профиль сервисной учетной записи `postgres` измененное значение

порта, это облегчит работу с утилитами PostgreSQL:

```

1 ---
2 .bash_profile.old 2021-06-05 00:47:06.303382240 +0300+++
3 .bash_profile 2021-06-05 00:37:04.264916442 +0300
4 @@ -1,6 +1,7 @@
5 [ -f /etc/profile ] && source /etc/profile
6 PGDATA=/var/lib/pgsql/11/data-
7 export PGDATA+
8 PGPORT=9898+
9 export PGDATA PGPORT
10 # If you want to customize your settings,
11 # Use the file below. This is not overridden
12 # by the RPMS.
```

C.1.1. Подключение к web-интерфейсу HAProxy для просмотра статистики и управления

Рекомендуем использование SSH SOCKS-прокси и плагина для Вашего браузера, например, для Firefox добавьте [FoxyProxy Standard](#).

И настройте расширение:

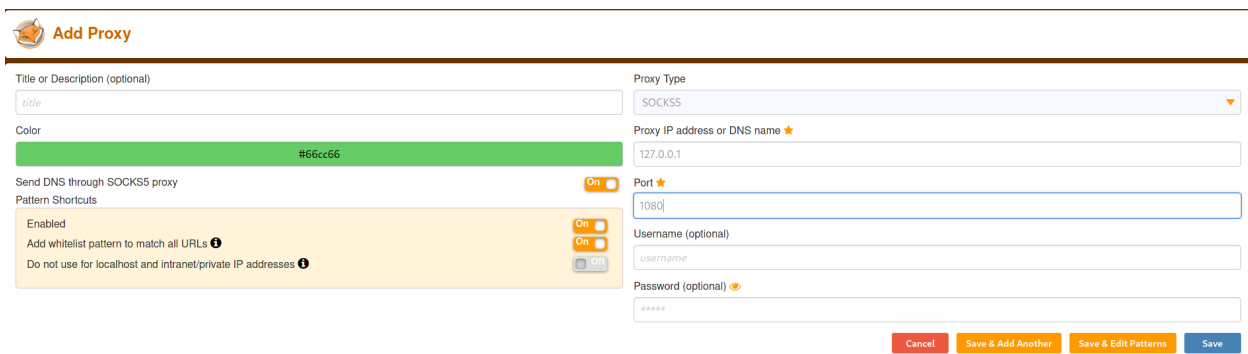


Рис. C.1. foxyproxy.png

Обеспечьте создание SOCKS-прокси через SSH-соединение

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

В качестве альтернативы Вы можете откорректировать конфигурационный файл HAProxy (директиву bind, разрешив доступ к интерфейсу других хостов) и добавить разрешения для доступа в фаервол сервера.

C.1.2. Тюнинг операционной системы

Настройка сетевого стека ядра для хоста с установленным HAProxy в чем-то похожа на настройку тестирующего хоста, создающего нагрузку:

```
/etc/sysctl.d/98-luxmsbi.conf:
```

```
1 net.core.netdev_max_backlog = 5000
2 net.core.somaxconn = 65535
3 net.ipv4.ip_local_port_range = 1025 65000
4 net.ipv4.tcp_max_syn_backlog = 5000
5 net.ipv4.tcp_tw_reuse = 1
```

С.2. HAProxy как балансировщик для кластера

Кластер PostgreSQL под управлением Patroni[consul] позволяет предоставлять доступ к экземпляру базы данных с возможностью добавления/изменения данных и к нескольким экземплярам с возможностью только чтения данных. Перенаправление запросов на чтение на выделенные сервера позволяет снизить общую нагрузку на основной экземпляр и обеспечить устойчивую работоспособность системы.

В случае выхода/вывода из рабочего режима одного из узлов кластера PostgreSQL Patroni оперативно выполняет передачу ролей и реконфигурацию кластера. Что требует такого же оперативного изменения конфигураций на серверах приложений.

В текущей архитектуре для балансировки нагрузки используется HAProxy. А для динамического изменения конфигурации HAProxy при изменении в кластере PostgreSQL используется решение Consul-Template.

С.3. Consul-Template. Установка и настройка

1. Поместить поставляемые шаблоны конфигурационных файлов (см. ниже) и архив с приложением на хост, в папку **/tmp/consul**. И выполнить команды:

```
1 cd /tmp/consul
2 #curl -k0 https://releases.hashicorp.com/consul-template/0.24.1/consul-template_0.24.1_linux_amd64.tgz
3 sudo -- sh -c 'unzip -u -d /usr/sbin consul-template_0.24.1_linux_amd64.tgz \
4 &&
5 chmod ug+x /usr/sbin/consul \
6 rm -r /tmp/consul-template_0.24.1_linux_amd64.tgz'

7 sudo mkdir -p /etc/consul-template.d /var/lib/consul/templates
8 sudo cp consul-template.hcl /etc/consul-template.d/00-consul-template.hcl
9 sudo cp haproxy.hcl /etc/consul-template.d/
10 sudo cp haproxy.ctmpl /var/lib/consul/templates/
11 sudo chown -R consul.consul /etc/consul-template.d /var/lib/consul/templates
12 sudo -- sh -c 'cp consul-template.service /etc/systemd/system/ \
13 &
14 sudo systemctl daemon-reload
15 sudo systemctl enable consul-template'
```

С.4. HAProxy. Установка и конфигурирование

“

Для разрешения проблемы *HAProxy* “Cannot bind socket” необходимо установить флаг *SELinux*:

```
1 setsebool -P haproxy_connect_any=1
```

Для установки HAProxy необходимо выполнить следующий перечень команд:

```
1 sudo yum -y haproxy
2 sudo setsebool -P haproxy_connect_any=1
3 sudo systemctl enable haproxy
4 sudo systemctl start consul-template haproxy
```

С.4.1. Шаблоны конфигурационных файлов

/etc/consul-template.d/00-consul-template.hcl

```
1 consul {
2   address = "127.0.0.1:8500"
3   token = "{{ consul_token }}"
4   retry {
5     enabled = true
6     attempts = 12
7     backoff = "250ms"
8     max_backoff = "10s"
9   }
10 }

12 reload_signal = "SIGHUP"

14 kill_signal = "SIGINT"

16 max_stale = "10m"

18 log_level = "warn"

20 # pid_file = "/run/consul-template.pid"

22 wait {
23   min = "2s"
24   max = "5s"
25 }

27 deduplicate {
28   enabled = true
29   prefix = "consul-template/dedup/"
30 }
```

/etc/consul-template.d/haproxy.hcl

```
1 template {
2   source = "/var/lib/consul/templates/haproxy.ctmpl"
3   destination = "/etc/haproxy/haproxy.cfg"
4   command = "systemctl reload haproxy"
5   command_timeout = "10s"
6   error_on_missing_key = false
7   backup = true
8   wait {
9     min = "2s"
10    max = "10s"
11  }
12 }
```



Документация по функциям и встроенным переменным для написания шаблонов [Consul Template language](#).

```
/var/lib/consul/templates/haproxy.ctmpl
```

```
1 # Rendered by consul-template.service {{ timestamp }}
2
3 global
4 daemon
5 chroot /var/lib/haproxy
6 user haproxy
7 group haproxy
8 pidfile /var/run/haproxy.pid
9 log /dev/log local0
10 maxconn 102400
11
12 defaults
13 log global
14 mode tcp
15 retries 2
16 timeout client 30m
17 timeout connect 4s
18 timeout server 30m
19 timeout check 5s
20
21 listen stats
22 bind *:2000
23 maxconn 100
24 mode http
25 option httplog
26 stats uri /stats
27 stats enable
28 stats refresh 10s
29 stats admin if LOCALHOST
30
31 ### Listener for PostgreSQL LEADER database
32 listen db-rw
33 bind 127.0.0.1:5432
34 maxcon 10240
```



```
35 timeout    queue 30s
36 option    httpchk OPTIONS /master
37 http-check expect status 200
38 default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions {{range ←
    service "master.db-main"}}
39 server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}

41 ### Listener for PostgreSQL REPLICa database
42 listen db-ro
43 bind      127.0.0.1:5433
44 maxconn   10240
45 timeout   queue 30s
46 option    httpchk OPTIONS /replica
47 http-check expect status 200
48 balance   roundrobin
49 default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions {{range ←
    service "replica.db-main"}}
50 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%}`

```
1 # Rendered by consul-template.service {%raw%}{{ timestamp }}{%endraw%}

3 global
4 daemon
5 chroot /var/lib/haproxy
6 user haproxy
7 group haproxy
8 pidfile /var/run/haproxy.pid
9 log /dev/log local0
10 maxconn 102400

12 defaults
13 log global
14 mode tcp
15 retries 2
16 timeout client 30m
17 timeout connect 4s
18 timeout server 30m
19 timeout check 5s

21 listen stats
22 bind *:2000
23 maxconn 100
24 mode http
25 option httplog
26 stats uri /stats
27 stats enable
28 stats refresh 10s
29 stats admin if LOCALHOST

31 ### Listener for PostgreSQL LEADER database
```

```

32 listen db-rw
33 bind      127.0.0.1:{{ db_rw_port}}
34 maxcon    10240
35 timeout   queue 30s
36 option    httpchk OPTIONS /master
37 http-check expect status 200
38 default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions ⌵
   {%raw%}{{range service "master.{{endraw%}}{{ consul_service ⌵
   }}{%raw%}}"}{{%endraw%}}
39 server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}{%endraw%}}

41 ### Listener for PostgreSQL REPLICa database
42 listen db-ro
43 bind      127.0.0.1:{{ db_ro_port}}
44 maxconn   10240
45 timeout   queue 30s
46 option    httpchk OPTIONS /replica
47 http-check expect status 200
48 balance   roundrobin
49 default-server inter 3s rise 2 fall 3 maxconn 100 shutdown-sessions ⌵
   {%raw%}{{range service "replica.{{endraw%}}{{ consul_service ⌵
   }}{%raw%}}"}{{%endraw%}}
50 server {%raw%}{{.Node}} {{.Address}}:{{.Port}} check port 8008{{end}}{%endraw%}}

```

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

```

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

7 [Service]
8 Type=simple
9 #User=consul
10 #Group=consul
11 ExecStart=/usr/sbin/consul-template -config=/etc/consul-template.d/
12 ExecReload=/bin/kill -HUP $MAINPID
13 KillSignal=SIGINT
14 TimeoutStopSec=5
15 Restart=on-failure
16 SyslogIdentifier=consul

18 [Install]
19 WantedBy=multi-user.target

```

Приложение D. Настройка SSO

Конфигурация Web-сервера, поставляемая для нашего приложения, содержит отключенные (комментированные) директивы для подключения SSO-авторизации. Web-приложение поддерживает интеграцию с kerberos инфраструктурой и LDAP-каталогами, в том числе MS AD и FreeIPA.

D.1. Настройка конфигурации Web-сервера

В файле `/opt/luxmsbi/conf/nginx/nginx.conf` раскомментировать строку с подключением модуля `ngx_http_auth_spnego_module.so`:

```
1 ---
2 nginx.conf.old 2021-10-05 17:19:58.562466594 +0300+++
3 nginx.conf 2021-10-05 17:19:49.449517250 +0300
4 @@ -7,7 +7,7 @@
5
6 load_module /usr/lib64/nginx/modules/ngx_http_lua_module.so;
7 #load_module /usr/lib64/nginx/modules/ngx_http_auth_spnego_module.so;-
8 #load_module /usr/lib64/nginx/modules/ngx_http_auth_spnego_module_debug.so;+
9 load_module /usr/lib64/nginx/modules/ngx_http_auth_spnego_module_debug.so;
```

Модуль собирается и тестируется нами, и доступен в публичном репозитории [Luxms BI RPM ThirdParty](#).

Необходимо переименовать/скопировать конфигурационный файл `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sslocation.off` в `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sslocation`:

```
1 mv /opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sslocation.off \
2 /opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sslocation
```

Изменить значения в конфигурационном файле `/opt/luxmsbi/conf/nginx/conf.d/luxmsbi-sslocation`:

```
1 # You should change next 3 lines according to your environment
2 # - put your KDC domain name
3 # - put keytab file full path.
4 #   Make sure the file has read permissions for user bi
5 # - put SPN name from keytab file without KDC domain suffix
6 auth_gss_realm          EXAMPLE.TLD;
7 auth_gss_keytab         /etc/bi.keytab;
8 auth_gss_service_name   "HTTP/bi.example.tld";
```

Вы можете посмотреть корректные значения для конфигурационного файла в вашем Kerberos-ключе, вызовите утилиту *klist*:

```
1 klist -ke /opt/luxmsbi/conf/nginx/bi.keytab
2 Keytab name: FILE:/opt/luxmsbi/conf/nginx/bi.keytab
3 KVNO Principal-----
4 -----
5 3 HTTP/bi.example.tld@EXAMPLE.TLD (des-cbc-crc)
6 3 HTTP/bi.example.tld@EXAMPLE.TLD(des-cbc-md5)
7 3 HTTP/bi.example.tld@EXAMPLE.TLD (arcfour-hmac)
8 3 HTTP/bi.example.tld@EXAMPLE.TLD (aes256-cts-hmac-sha1-96)
9 3 HTTP/bi.example.tld@EXAMPLE.TLD (aes128-cts-hmac-sha1-96)
```

Сверьте настройки в файле конфигурации со значениями из *keytab*-файла.

D.1.1. Проверка работоспособности Web-сервера

До применения измененной конфигурации запустите команду проверки:

```
1 sudo nginx -c /opt/luxmsbi/conf/nginx/nginx.conf -t
```

Перезапустить сервис *luxmsbi-web* и проверить журналы на отсутствие ошибок после перезапуска:

```
1 sudo systemctl restart luxmsbi-web
2 sudo systemctl status luxmsbi-web -l
3 sudo journalctl -u luxmsbi-web
```

D.1.2. Проверка работы модуля SPNEGO

Если целевой пользователь с заведенной в домен машиной заходит в систему с включенным отладочным модулем веб-сервера `nginx ngx_http_auth_spnego_module_debug.so` при включенном логировании, то в лог-файле `/var/log/luxmsbi/nginx/luxmsbi-ssso.debug.log` отображается информация следующего вида:

```
1 2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling IN: token.len=0,
   head=0, ret=401
2 2022/05/20 18:05:26 [debug] 958#958: *78798 Begin auth
3 2022/05/20 18:05:26 [debug] 958#958: *78798 Detect SPNEGO token
4 2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling OUT: token.len=0,
   head=1, ret=401
5 2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling IN: token.len=0,
   head=0, ret=401
6 2022/05/20 18:05:26 [debug] 958#958: *78798 Begin auth
7 2022/05/20 18:05:26 [debug] 958#958: *78798 Detect SPNEGO token
8 2022/05/20 18:05:26 [debug] 958#958: *78798 Token decoded: YIII+wYGkWyB...
9 [Тело Kerberos-ticket]
```

```

10 2022/05/20 18:05:26 [debug] 958#958: *78798 Client sent a reasonable Negotiate header
11 2022/05/20 18:05:26 [debug] 958#958: *78798 GSSAPI authorizing
12 2022/05/20 18:05:26 [debug] 958#958: *78798 Use keytab /etc/bi.keytab
13 2022/05/20 18:05:26 [debug] 958#958: *78798 Using service principal: HTTP/hostname@MAIN.DOMAIN.LOCAL
14 2022/05/20 18:05:26 [debug] 958#958: *78798 my_gss_name HTTP/hostname@MAIN.DOMAIN.LOCAL
15 2022/05/20 18:05:26 [debug] 958#958: *78798 ngx_http_auth_spnego_set_bogus_authorization: bogus user set
16 2022/05/20 18:05:26 [debug] 958#958: *78798 user is USER_TEST@MAIN.DOMAIN.LOCAL
17 2022/05/20 18:05:26 [debug] 958#958: *78798 GSSAPI auth succeeded
18 2022/05/20 18:05:26 [debug] 958#958: *78798 SSO auth handling OUT: token.len=2303, head=1, ret=0

```

Данный лог информирует, что со стороны целевой пользовательской машины настройки правильные. Естественно, могут быть другие проблемы. Например, пользователь не в той группе или залогинился не под своим логином.

Если машина целевого пользователя не в домене, то будет приходить NTLM-ный токен (он намного меньше по длине) следующего вида:

```

1 2022/05/20 14:11:16 [debug] 958#958: *75240 SSO auth handling IN: token.len=0, head=0, ret=401
2 2022/05/20 14:11:16 [debug] 958#958: *75240 Begin auth
3 2022/05/20 14:11:16 [debug] 958#958: *75240 Detect SPNEGO token
4 2022/05/20 14:11:16 [debug] 958#958: *75240 Token decoded: TLRDFSDFSDGDDBAAAAA\AAASDFSFAAAAAAKALpHAAAADw==
5 2022/05/20 14:11:16 [debug] 958#958: *75240 Client sent a reasonable Negotiate header
6 2022/05/20 14:11:16 [debug] 958#958: *75240 GSSAPI authorizing
7 2022/05/20 14:11:16 [debug] 958#958: *75240 Use keytab /etc/bi.keytab
8 2022/05/20 14:11:16 [debug] 958#958: *75240 Using service principal: HTTP/hostname@MAIN.DOMAIN.LOCAL
9 2022/05/20 14:11:16 [debug] 958#958: *75240 my_gss_name HTTP/hostname@MAIN.DOMAIN.LOCAL
10 2022/05/20 14:11:16 [debug] 958#958: *75240 gss_accept_sec_context() failed: Unknown error:
11 2022/05/20 14:11:16 [debug] 958#958: *75240 GSSAPI failed

```

При подобных логах машина целевого пользователя настроена неправильно. И, скорее всего, машина не заведена в домен.


D.2. Интеграция с LDAP-каталогами

При необходимости настройки распределения прав в системе Luxms BI по членству в группах LDAP-каталога требуется установка компонента `luxmsbi-gateway`, предоставляющего API для проверки учетной записи и получения списка групп, в которой состоит пользователь.

Компонент **Luxms BI Gateway** использует конфигурационный файл `/opt/luxmsbi/conf/luxmsbi-gateway.yml`. Пример конфигурации:

```
1 # general HTTP server configuration
2 # address:port that server will be listening and serving on
3 listen: 'localhost:8889'
4 # logs all incoming/outgoing HTTP requests/responses
5 http-trace: false
6 # can be: error, warn, info, debug (default: "info")
7 log-level: 'info'
8 # serve HTTPS requests instead of HTTP
9 use-tls: false
10 # path to *.cert/*.pem certificate file, ignored if use-tls: false
11 cert-file: './localhost.crt'
12 # path to *.key key file, ignored if use-tls: false
13 key-file: './localhost.key'
14 # connections timeout, global setting for all connections (default: "30s")
15 timeout: '30s'

18 # Simple config for MS AD
19 ad:
20 base: 'dc=example,dc=org'
21 host: 'dc-01.example.org'
22 port: 3268
23 usessl: false
24 binddn: 'bind-user@example.org'
25 bindpw: 'BindPassword'
26 # Specify LDAP attribute to provide as "login" at output JSON
27 returnAsLogin: userPrincipalName
28 # Filter groups to output JSON
29 groupFilter: -
30 groupname

33 # Simple config for OpenLDAP config:
34 ldap:
35 base: 'dc=example,dc=com'
36 host: 'dc-01.example.org'
37 port: 636
38 usessl: true
39 skipSSLCertVerify: true
40 binddn: 'uid=bind-user,dc=example,dc=org'
41 bindpw: 'BindPassword'
42 userSearchFilter: '(uid=%s)'
43 # Used for pure OpenLDAP to search user group by user DN on its member 
44 attribute
45 groupSearchFilter: '(&(objectClass=posixgroup)(member=%s))'
46 attributes: -
47 cn -
48 mail
49 # Specify LDAP attribute to provide as "login" at output JSON
50 returnAsLogin: uid
51 # Filter groups to output JSON
52 groupFilter: -
53 groupname
```

```
53 # Provide filtered group DN's
54 includeGroupDNs: true

56 # Simple config for Free IPA:
57 ldap:
58 base: 'dc=example,dc=com'
59 host: 'dc-01.example.org'
60 port: 636
61 usessl: true
62 skipSSLCertVerify: true
63 binddn: 'uid=ldap-sso,dc=example,dc=com'
64 bindpw: 'BindPassword'
65 userSearchFilter: '(uid=%s)'
66 attributes: -
67 mail
68 # Specify LDAP attribute to provide as "login" at output JSON
69 returnAsLogin: krbPrincipalName
70 # Filter groups to output JSON
71 groupFilter: -
72 group-*
73 # Provide filtered group DN's
74 includeGroupDNs: true

77 # Simple config for PDF converter
78 # html2pdf:
79 #   chromeDevToolsURI: "http://localhost:9222"
80 #   tmpFilesDirectory: "/tmp/"
81 #   orientation: "landscape"
82 #   printBackground: true
83 #   marginTop: 0.5
84 #   marginBottom: 0.5
85 #   marginLeft: 0.5
86 #   marginRight: 0.5
87 #   paperWidth: 8.5
88 #   paperHeight: 11.0
```

После установки компонента и настройки конфигурации `luxmsbi-gateway` необходимо выполнить следующие дополнительные действия.

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

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

Для DEB-based ОС перезапустить сервис:

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

D.2.1. Проверка конфигурации Luxmsbi-gateway

При выполнении команды `journalctl -u luxmsbi-gateway` можем увидеть логирование компонента `luxmsbi-gateway`, где будет отображаться подробная информация:

```
1 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [INFO ] AD ↔  
  userAndGroupsHandler for USER_TEST@MAIN.DOMAIN.LOCAL  
2 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↔  
  AuthenticateExtendedLong: Getting UPN for USER_TEST@MAIN.DOMAIN.LOCAL  
3 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↔  
  AuthenticateExtendedLong: Will bind as sys_USER@MAIN.DOMAIN.LOCAL, getting UPN  
4 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↔  
  AuthenticateExtendedLong: Connecting to AD  
5 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↔  
  AuthenticateExtendedLong: Bind as sys_USER@MAIN.DOMAIN.LOCAL with known pass  
6 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] ↔  
  AuthenticateExtendedLong: Bind Ok  
7 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] Got User: ↔  
  USER_TEST@MAIN.DOMAIN.LOCAL  
8 May 23 13:05:37 hostname.domain.local luxmsbi-gateway[11908]: [DEBUG] Got cn ↔  
  Attr: Userov User Testovich
```

В случае, если в логах данные сообщения не появляются необходимо проверить настройку `LogLevel: debug` в файле `/opt/luxmsbi/conf/luxmsbi-gateway.yml`.

D.3. Настройка пользовательских браузеров

Настройки SSO на стороне сервера не всегда гарантирует работу его на клиентских машинах.

D.3.1. Internet Explorer:

Для настройки SSO в IE, нужно зайти в настройки и добавить сайт в надежные узлы.

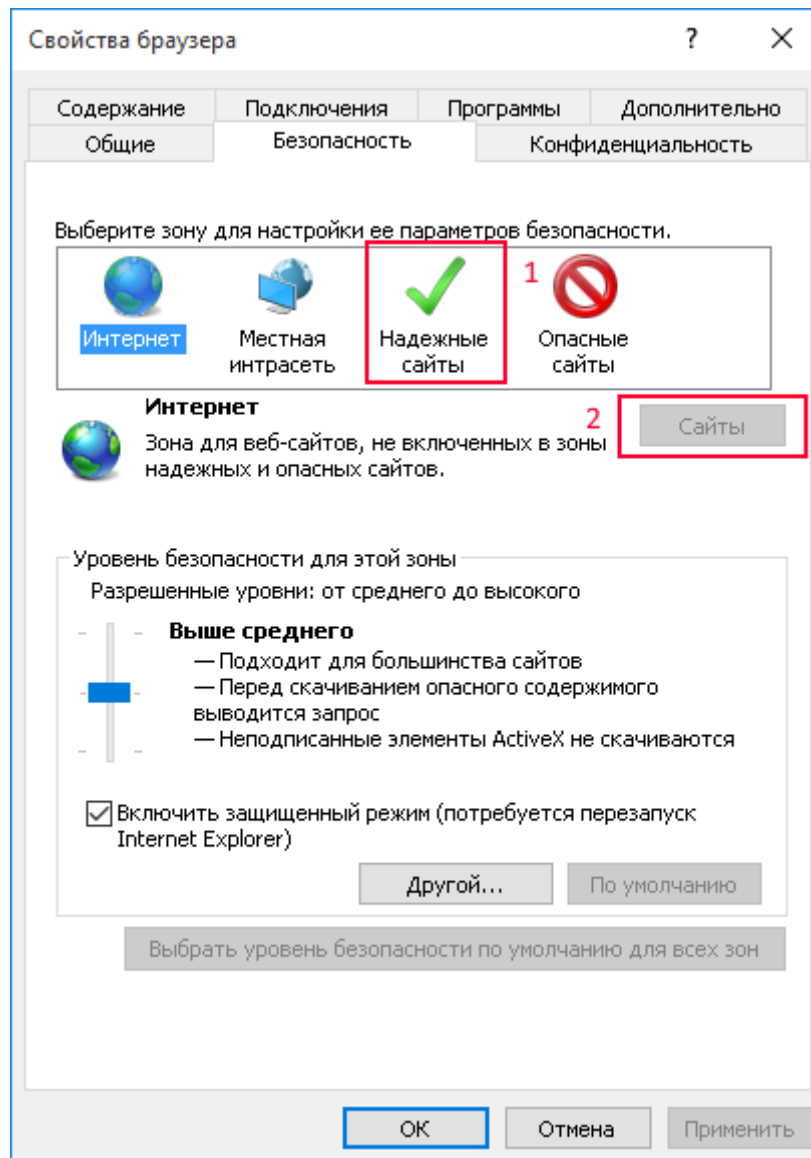
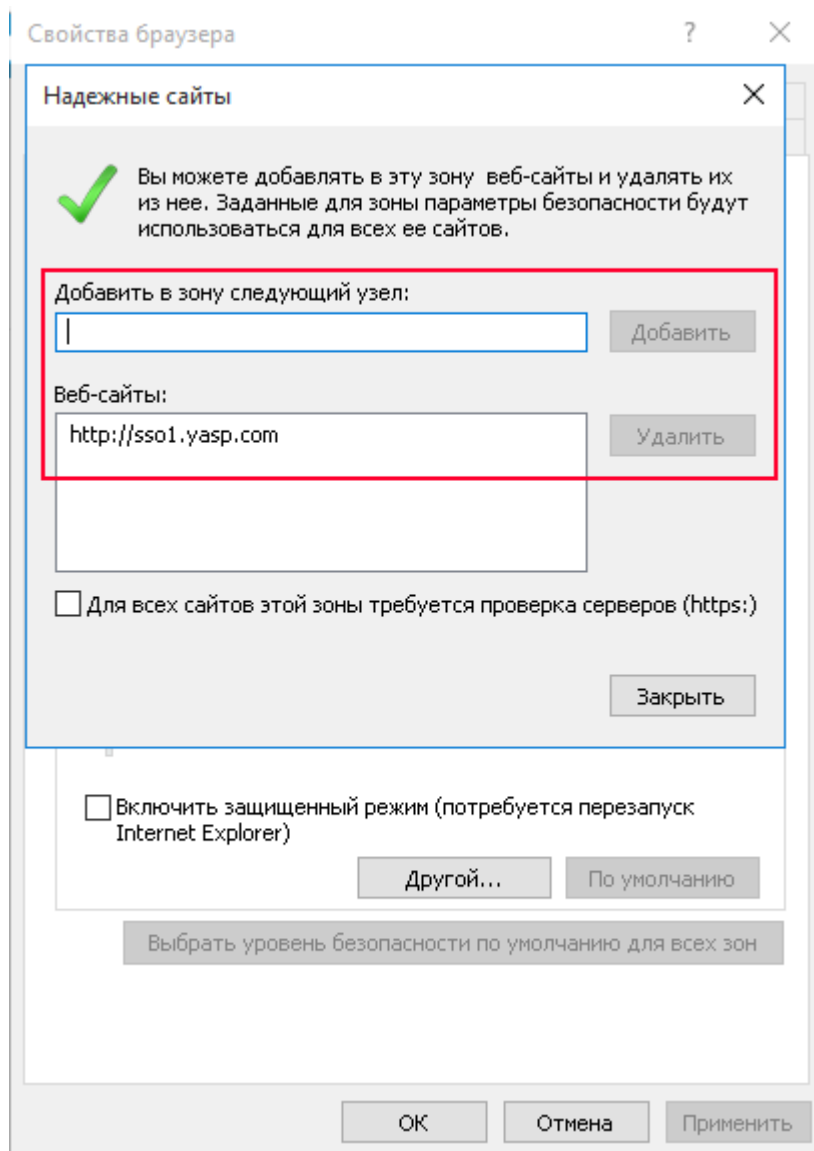
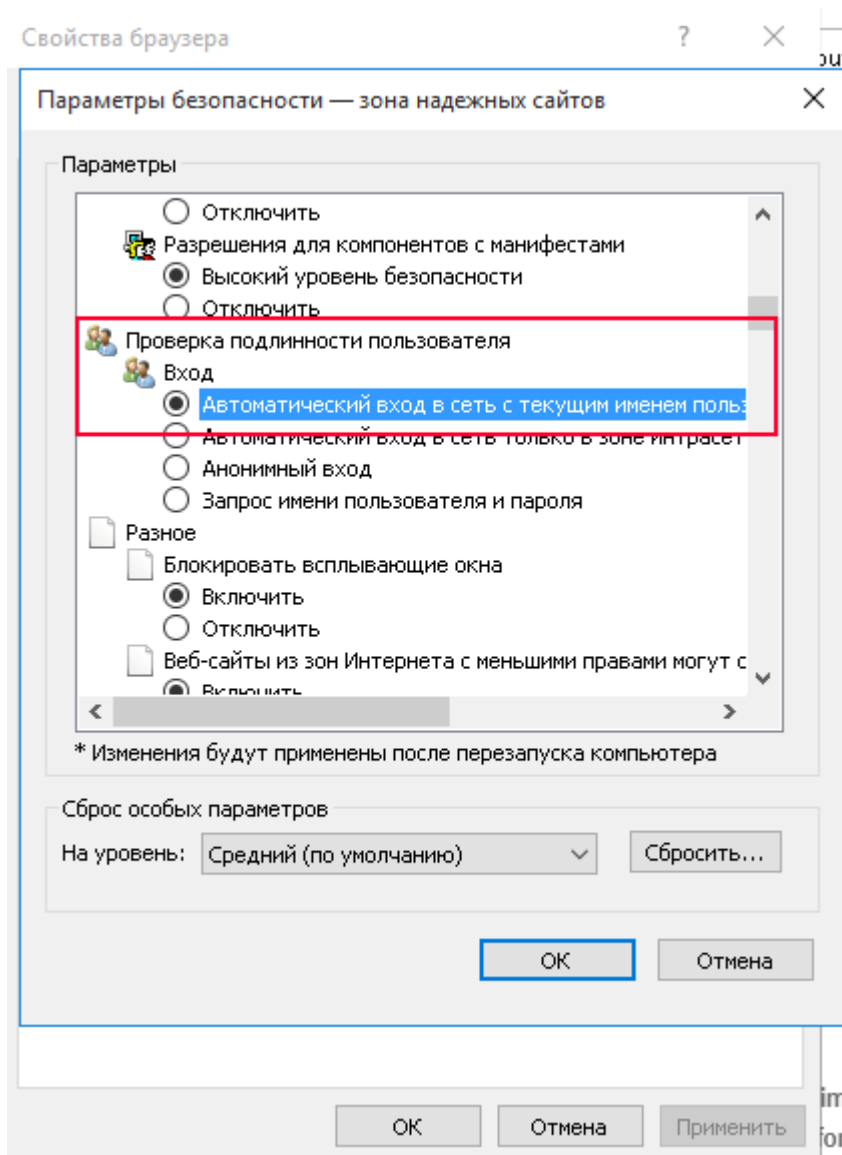


Рис. D.1. 1-sso-browser-settings.png

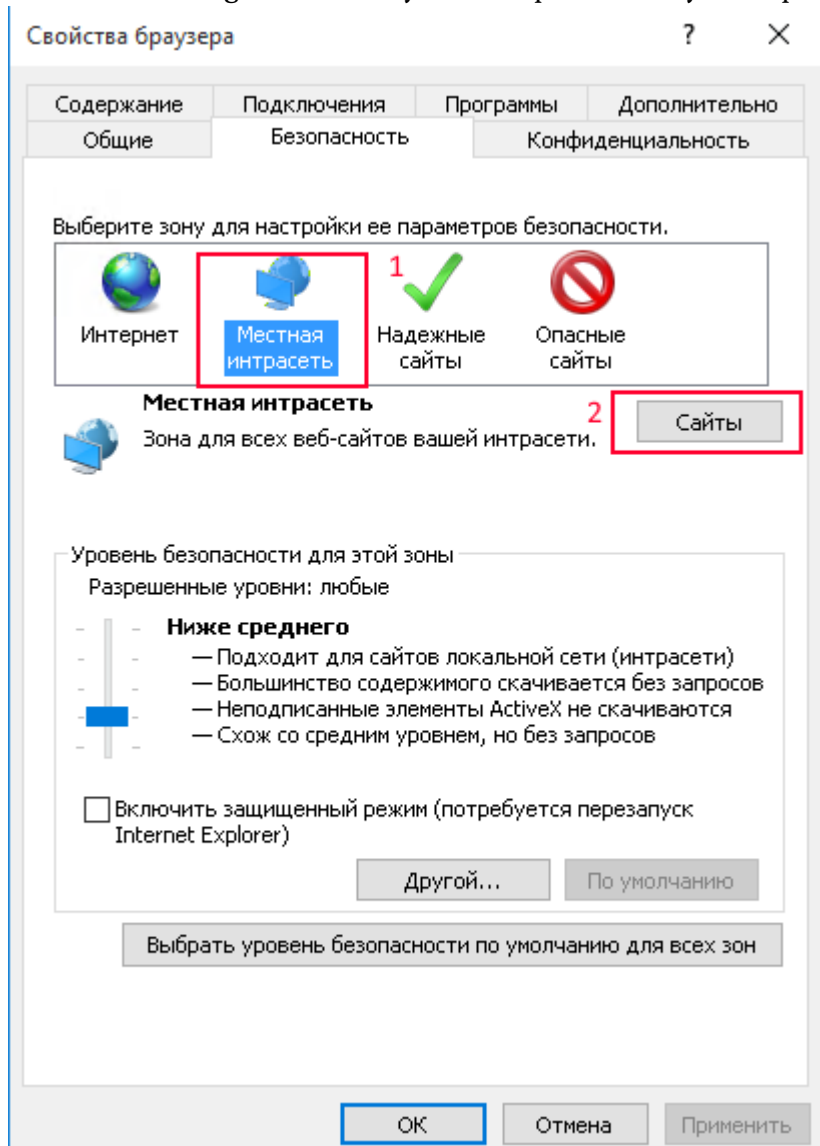


Так же, в секции надежные узлы, нужно выбрать уровень безопасности (Другой) и выбрать проверку подлинности (Автоматический вход в сеть с текущим именем пользователя и паролем). Хочу обратить внимание - это актуально, если сайт находится в локальной сети.



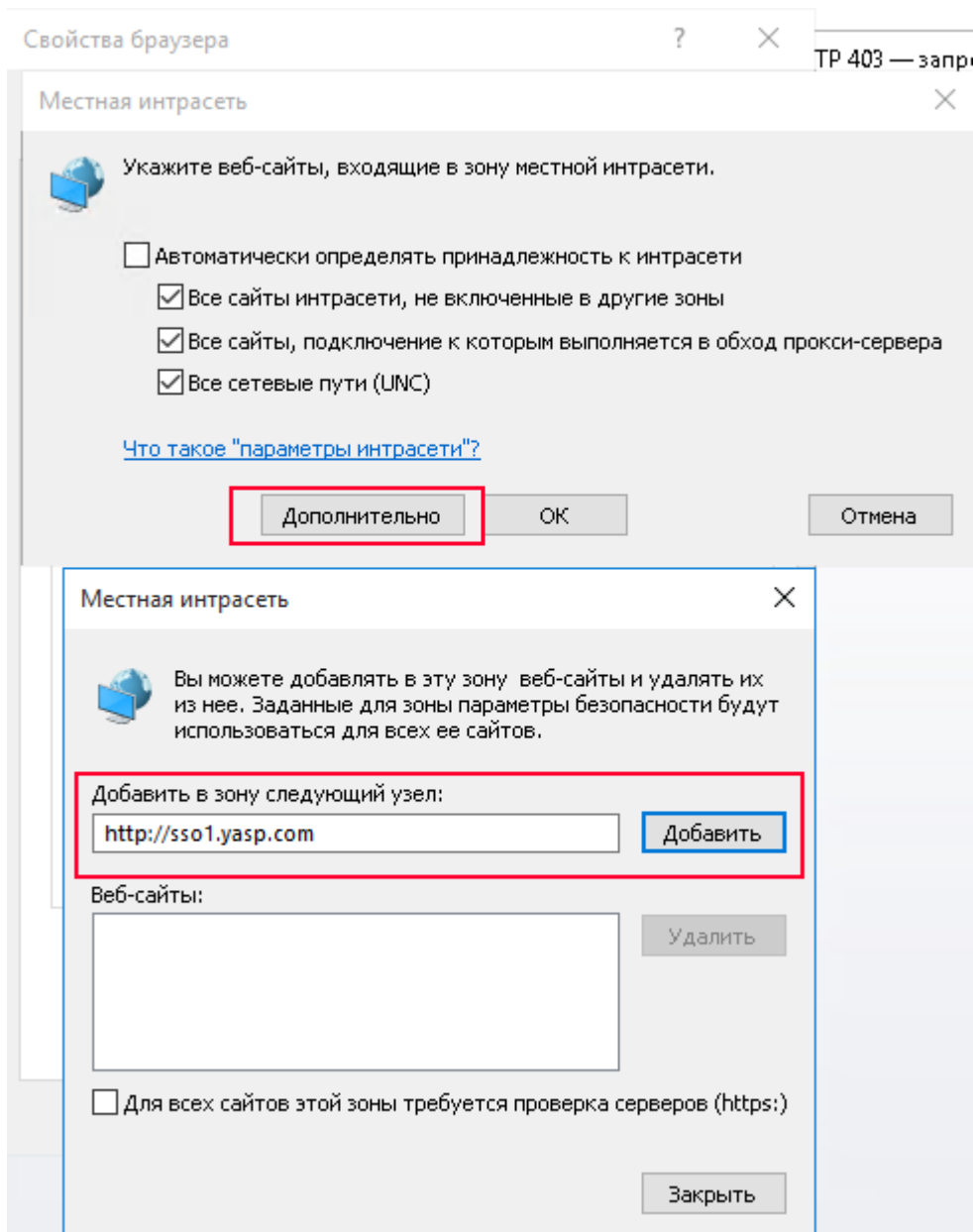
D.3.2. Windows 10 EDGE:

Что бы тоже самое заработало и в Microsoft Edge browser. Нужно выбрать местную интра-

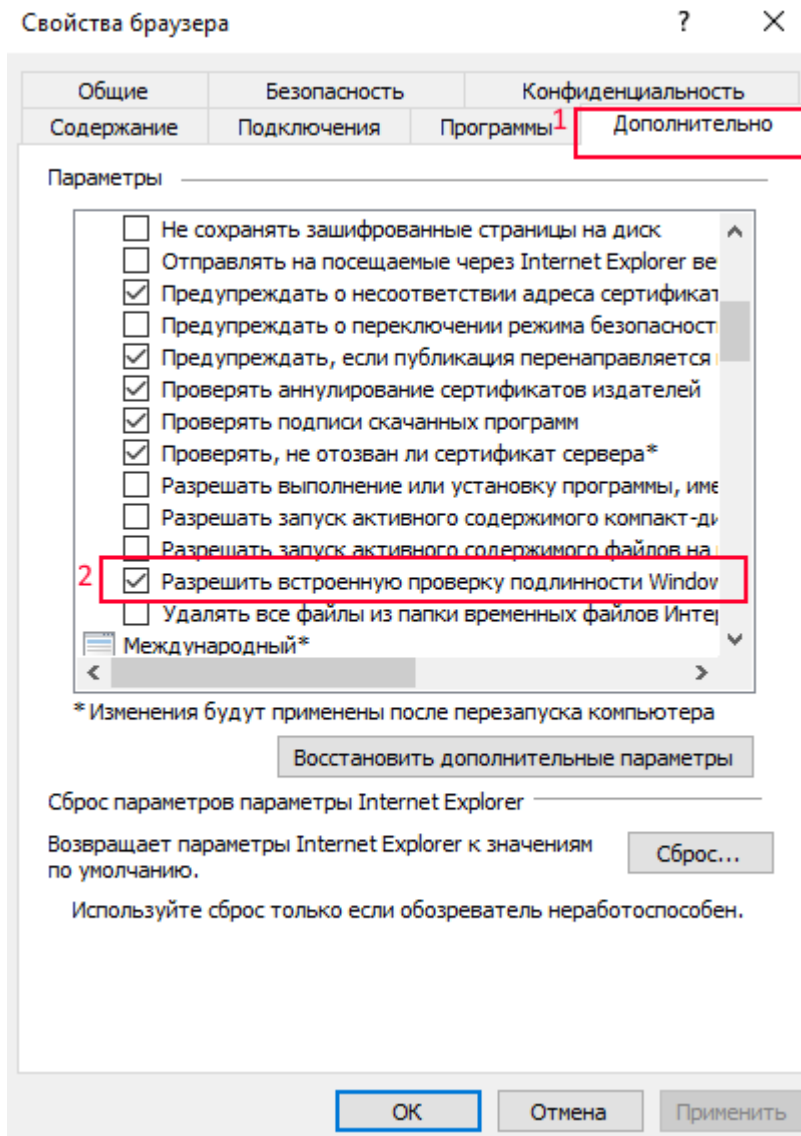


сеть и нажать кнопку (сайты).

Выбираем кнопку (дополнительно) и так же вставляем туда адрес нашего сайта.

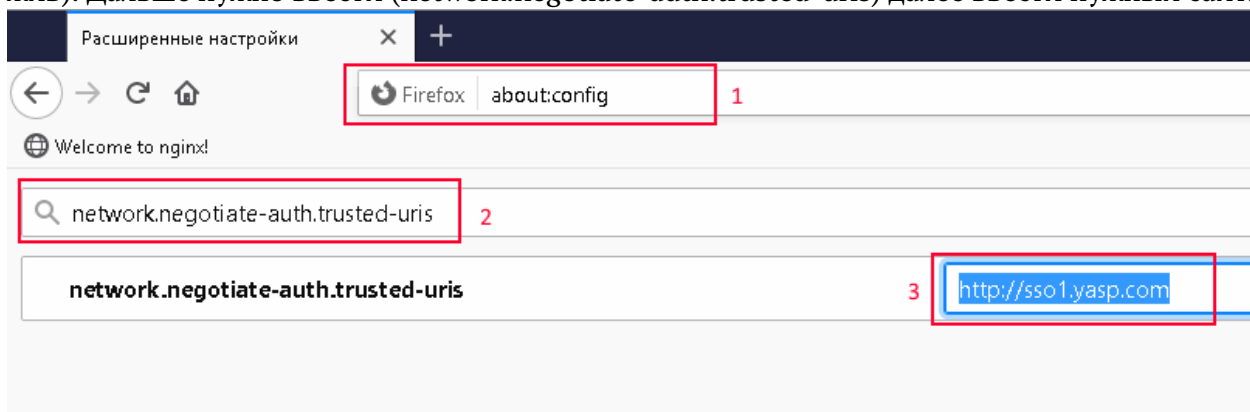


Не забываем проверить в настройках броузера (Дополнительно - > Разрешить встроенную проверку подлинности Windows) это актуально для обоих браузеров IE & Edge.



D.3.3. Firefox:

В строке браузера, нужно вписать `about:config`, согласиться (приняв риск и продолжив). Далее нужно ввести `(network.negotiate-auth.trusted-uris)` далее ввести нужный сайт.



D.3.4. Yandex & Chrome

Что касается этих браузеров, они берут свои настройки из IE, так что после настройки IE & Edge, остальные будут работать, как надо.

D.4. Генерация Kerberos-ключей

Описание процедуры организации Kerberos-аутентификации для Web(HTTP/HTTPS) приложений на ОС Linux при интеграции с каталогом MS AD.

D.4.1. Создание сервисной учетной записи

Создание учетной записи для обеспечения **Kerberos** - аутентификации необходимо помнить, что **Kerberos(Secret Key)** ключ генерируется на основе пароля учетной записи. Поэтому необходимо обеспечить неизменность пароля или регенерацию ключей при его изменении.

В случае использования учетной записи рабочей станции(computer) изменение пароля производится автоматически *каждые 30 дней*, соответственно необходима настройка с такой же периодичностью механизма регенерации ключей. В гетерогенных вычислительных сетях это требует регистрации серверов Linux в домене MS AD и использование скриптового решения по генерации ключей

Для горизонтально масштабированных решений более целесообразный вариант - использование пользовательской учетной записи с применением генерации пароля из набора случайных символов с длинной, обеспечивающей защиту от взлома методом перебора - например 16 символов, и запретом на изменение пароля учетной записи.

Исходные данные:

- Приложение доступно по адресу - <http://www.example.org/application>
- Домен MS AD - `example.org`
- Сервисная учетная запись - `service-account`

D.4.2. Регистрация Service Principal Name (SPN)

Формат имени сервисной учетной записи для Web-приложений, не зависимо от использования SSL-шифрования, имеет следующий вид:

HTTP/<web-service>@**<REALM>**

где:

- `web_service` - URL, DNS-имя web-приложения
- `REALM` - имя Kerberos REALM, обычно совпадает с именем домена MS AD, символами верхнего регистра

например: `HTTP/www.example.org@EXAMPLE.ORG`

Регистрация сервисных учетных записей производится с использованием утилит командной строки в ОС Windows, под доменной учетной записью, обладающей полномочиями для изменения учетных записей, достаточно только предоставление прав на изменение конкретных учетных записей.

```
1 Microsoft Windows [Version 6.3.9600]
2 (c) 2013 Microsoft Corporation. All rights reserved.

4 C:\>setspn -A HTTP/www.example.org@EXAMPLE.ORG example\service-account
5 Checking domain DC=example,DC=org

7 Registering ServicePrincipalNames for CN=service-account,DC=example,DC=org
8 HTTP/www.example.org@EXAMPLE.ORG
9 Updated object
```

Выполняется регистрация всех необходимых сервисных учетных записей. Возможно также регистрация сервисных учетных записей на короткие (NetBIOS) имена систем, IP-адреса

D.4.3. Проверка сгенерированных SPN

```
1 Microsoft Windows [Version 6.3.9600]
2 (c) 2013 Microsoft Corporation. All rights reserved.

4 C:\>setspn -L example\service-account
5 Registered ServicePrincipalNames for CN=service-account,DC=example,DC=org:
6 HTTP/www.example.org@EXAMPLE.ORG
7 HTTP/www@EXAMPLE.ORG

10 c:\>ldifde -d "CN=service-account,DC=example,DC=org" -l "userPrincipalName,
    servicePrincipalName,msDS-KeyVersionNumber" -f account.ldif
11 Connecting to "dc-01.example.org"
12 Logging in as current user using SSPI
13 Exporting directory to file account.ldif
14 Searching for entries...
15 Writing out entries.
16 1 entries exported

18 The command has completed successfully

20 c:\temp>type account.ldif
21 dn: CN=service-account,DC=example,DC=org
22 changetype: add
23 userPrincipalName: service-account@example.org
24 servicePrincipalName: HTTP/www.example.org@EXAMPLE.ORG
25 servicePrincipalName: HTTP/www@EXAMPLE.ORG
26 msDS-KeyVersionNumber: 2
```


D.4.4. Генерация ключей

При генерации ключей для нескольких SPN основной момент - сохранение KVNO(Key Value Number), изменение KVNO в процессе генерации ключей приведет к неработоспособности части ключей. Поэтому обратите внимание на ключ **-setpass** в вызове генерации ключей для второго и последующих SPN.

Для генерации последовательности случайных символов для пароля можно использовать утилиту командной строки openssl.

```
1 $ openssl rand -base64 20
2 jgDMj2KvZSqkEw2yWVxIVrfptGo=
```

Используем секретный пароль при генерации ключей

```
1 Microsoft Windows [Version 6.3.9600]
2 (c) 2013 Microsoft Corporation. All rights reserved.
4 c:\temp>ktpass /mapuser example\service-account /princ HTTP/
  HTTP/www.example.org@EXAMPLE.ORG /ptype KRB5_NT_PRINCIPAL /pass
  jgDMj2KvZSqkEw2yWVxIVrfptGo /crypto ALL /out key-1.keytab +answer
5 Targeting domain controller: dc-01.example.org
6 Successfully mapped HTTP/www.example.org to service-account.
7 Password successfully set!
8 Key created.
9 Key created.
10 Key created.
11 Key created.
12 Key created.
13 Output keytab to key-1.keytab:
14 Keytab version: 0x502
15 keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3
  etype 0x1 (DES-CBC-CRC) keylength 8 (0x5e6befd37c4913ba)
16 keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3
  etype 0x3 (DES-CBC-MD5) keylength 8 (0x5e6befd37c4913ba)
17 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3
  etype 0x17 (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
18 keysize 95 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3
  etype 0x12 (AES256-SHA1) keylength 32
  (0xd520e8aed124bb5213cba436d3e9d6cd1d5ba54fdd5919e406aa185977dd121a)
19 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3
  etype 0x11 (AES128-SHA1) keylength 16 (0xe14a7e0e13917c1165f5b8848f00c20f)
21 c:\temp>ktpass /mapuser example\service-account /princ HTTP/www@EXAMPLE.ORG
  /ptype KRB5_NT_PRINCIPAL /pass jgDMj2KvZSqkEw2yWVxIVrfptGo -setpass /kvno 3
  /crypto ALL /in key-1.keytab /out http.keytab -setupn
22 Existing keytab:
24 Keytab version: 0x502
25 keysize 71 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x1
  (DES-CBC-CRC) keylength 8 (0x5e6befd37c4913ba)
26 keysize 71 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3
  (DES-CBC-MD5) keylength 8 (0x5e6befd37c4913ba)
```

```

27 keysize 79 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
28 keysize 95 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 (AES256-SHA1) keylength 32 (0xd520e8aed124bb5213cba436d3e9d6cd1d5ba54fdd5919e406aa185977dd121a)
29 keysize 79 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x11 (AES128-SHA1) keylength 16 (0xe14a7e0e13917c1165f5b8848f00c20f)
30 Targeting domain controller: gvc-dc-02.gvc.oao.rzd
31 Successfully mapped HTTP/www to service-account.
32 Key created.
33 Key created.
34 Key created.
35 Key created.
36 Key created.
37 Output keytab to http.keytab:
38 Keytab version: 0x502
39 keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x1 (DES-CBC-CRC) keylength 8 (0x5e6befd37c4913ba)
40 keysize 71 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 (DES-CBC-MD5) keylength 8 (0x5e6befd37c4913ba)
41 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
42 keysize 95 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 (AES256-SHA1) keylength 32 (0xd520e8aed124bb5213cba436d3e9d6cd1d5ba54fdd5919e406aa185977dd121a)
43 keysize 79 HTTP/www.example.org@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x11 (AES128-SHA1) keylength 16 (0xe14a7e0e13917c1165f5b8848f00c20f)
44 keysize 71 HTTP/www@EXAMPLE.ORGD ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x1 (DES-CBC-CRC) keylength 8 (0x25a8e3403d4a342c)
45 keysize 71 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 (DES-CBC-MD5) keylength 8 (0x25a8e3403d4a342c)
46 keysize 79 HTTP/www@EXAMPLE.ORGD ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0xe8256a4b795e15a4ade75b5faa040be1)
47 keysize 95 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 (AES256-SHA1) keylength 32 (0x452317c637a6ef2b236ef9e9232d03dc3ae95a85f5132e3274710a5cb9c0c9c4)
48 keysize 79 HTTP/www@EXAMPLE.ORG ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x11 (AES128-SHA1) keylength 16 (0x0917d8ff99a2b30e381a3e53ce0b5420)

```

По результатам генерации сохраняем `http.keytab` и можем также сохранить пароль - пароль можно использовать для проверки работоспособности сервисной учетной записи. Если пароль случайно утерян, это не повлияет на работоспособность Kerberos-аутентификации.

D.4.5. Установка и проверка работоспособности

Установка файла `http.keytab`, классически производится в папку `/etc`, но в нашем случае установка возможна и в локальные папки приложения. Главное в обоих случаях дать разрешения файловой системы, достаточные для чтения файла владельцу процесса `NGinx`.

D.4.6. Настройка NGinx

Настройка ОС

```
1 includedir /etc/krb5.conf.d/
3 [logging]
4 default = FILE:/var/log/krb5libs.log
5 kdc = FILE:/var/log/krb5kdc.log
6 admin_server = FILE:/var/log/kadmind.log
8 [libdefaults]
9 default_keytab_name=bi5.keytab
10 dns_lookup_realm = false
11 ticket_lifetime = 24h
12 renew_lifetime = 7d
13 forwardable = true
14 rdns = false
15 pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
16 default_realm = EXAMPLE.ORG
17 default_ccache_name = KEYRING:persistent:%{uid}
19 [realms]
20 YASP.COM = {
21 kdc = dc-01.example.org
22 admin_server = dc-01.example.org
23 }
25 [domain_realm]
26 .example.org = EXAMPLE.ORG
27 example.org = EXAMPLE.ORG
```

Установка файла `http.keytab`, классически производится в директорию `/etc`, но в нашем случае установка возможна и в локальные папки приложения. Главное в обоих случаях дать разрешения файловой системы, достаточные для чтения файла владельцу процесса **NGinx**.

```
1
2 auth_gss on;
3 auth_gss_realm EXAMPLE.ORG;
4 auth_gss_keytab /etc/http.keytab;
5 auth_gss_service_name HTTP/www.example.org;
6 auth_gss_format_full on;
7 auth_gss_allow_basic_fallback off;
8 proxy_set_header Authorization "";
9 proxy_set_header X-Forwarded-User $remote_user;
```

D.5. Настройка прав в приложении Luxms BI

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

Приложение Е. Настройка SSL



Для высоконагруженных инсталляций Luxms BI подключение HTTPS на Web-серверах Luxms BI приводит к дополнительной нагрузке на CPU. Если это является существенной проблемой, рекомендуем подключение и использование аппаратных SSL ускорителей.

Также рекомендуем терминировать SSL трафик на аппаратных балансировщиках нагрузки и использовать обычный HTTP или HTTP/2 между балансировщиками и Web-серверами Luxms BI.

Конфигурация Web-сервера, поставляемая для нашего приложения, содержит отключенные (комментированные) директивы для подключения SSL-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`:

```
1 ---
2 entrypoint.conf.old 2021-10-08 17:36:21.173807998 +0300+++
3 entrypoint.conf 2021-10-08 17:36:43.564734068 +0300
4 @@ -3,7 +3,7 @@
5 listen      80;
6
7 # Uncomment next line to allow SSL-
8 #include /opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl+
9 include /opt/luxmsbi/conf/nginx/conf.d/luxmsbi.ssl
10
11 #   access_log   /var/log/luxmsbi/nginx/luxmsbi.access.log with_timing;
12 error_log      /var/log/luxmsbi/nginx/luxmsbi.errors.log;
```

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

```
1
2 listen      443 ssl;
3
4 ssl_certificate      /opt/luxmsbi/conf/ssl/host-full.cer;
5 ssl_certificate_key  /opt/luxmsbi/conf/ssl/host.key;
6 ssl_session_timeout  5m;
7 ssl_protocols        TLSv1.1 TLSv1.2;
```

```
8 ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH'; ↵
9 ssl_prefer_server_ciphers on;
10 ssl_session_cache shared:SSL:10m;
11 ssl_dhparam /etc/nginx/ssl/dhparam.pem;
12 add_header Strict-Transport-Security "max-age=31536000; ↵
    includeSubdomains;";
```

Необходимо откорректировать путь и имена файлов в параметрах конфигурации:

- `ssl_certificate`
- `ssl_certificate_key`



В конфигурационных файлах Nginx не забывайте писать символ `;` в конце строки при корректировке имени и пути к файлам.

Конфигурационный файл предполагает хранение сертификатов по пути `/opt/luxmsbi/↵
conf/ssl/`. Убедитесь, что эти файлы расположены там, где они должны быть, и имеют правильные разрешения доступа и корректного владельца:

```
1 ls -la /opt/luxmsbi/conf/nginx/ssl
3 chown -R bi.bi /opt/luxmsbi/conf/nginx/ssl
4 chmod 640 /opt/luxmsbi/conf/nginx/ssl/*
```

E.2. Проверка работоспособности

До применения измененной конфигурации запустите команду проверки:

```
1 sudo nginx -c /opt/luxmsbi/conf/nginx/nginx.conf -t
```

Перезапустить сервис `luxmsbi-web` и проверить журналы на отсутствие ошибок после перезапуска:

```
1 sudo systemctl restart luxmsbi-web
2 sudo systemctl status luxmsbi-web -l
3 sudo journalctl -u luxmsbi-web
```



