

Systematica Radius Enterprise Infrastructure / Руководство по технической установке для OS Linux & СУБД PostgreSQL

Редакция: 09 июля 2024

Оглавление

О документе	4
Введение.....	5
Требования к специалистам, производящим техническую установку серверного программного обеспечения «Systematica Radius Enterprise Infrastructure»	6
Способы получения программного обеспечения «Systematica Radius Enterprise Infrastructure» для установки.....	7
Состав файлов пакета установки (дистрибутива) программного обеспечения «Systematica Radius Enterprise Infrastructure» для установки.....	8
Файлы для установки на сервере базы данных	8
Файлы для установки серверных компонент программного обеспечения «Systematica Radius Enterprise Infrastructure» на отдельный от СУБД сервер	8
Краткий обзор архитектуры программного обеспечения «Systematica Radius Enterprise Infrastructure»	9
Утилита мониторинга работоспособности компонент программного обеспечения «Systematica Radius Enterprise Infrastructure».....	10
[Шаг 1] Требования к системе. Проверка	11
[Шаг 2] Требования к правам пользователя операционной системы производящего установку ПО. Проверка.....	12
Структура файлов и папок.....	13
Установка компонентов «Systematica Radius Enterprise Infrastructure».....	15
Зависимости на сторонние библиотеки.....	16
Описание настроек, стандартных для большинства компонентов	17
Введение.....	17
Пример конфигурационного файла	17
Описание	17
MessageBus.....	17
Log	17
MonitorAgent.....	17
DBConnection.....	18
Шифрование паролей.....	19
Введение.....	19
Пример	19
[Шаг 3] Установка и настройка «Systematica DB Back-end Server Extensions».....	20
Введение.....	20
Установка.....	20

[Шаг 4] Установка и настройка «Systematica Database Event Manager»	21
Введение.....	21
Используемые директории	21
Зависимости.....	21
Пример конфигурационного файла.....	21
Установка.....	21
[Шаг 5] Установка моделей данных Программного Обеспечения «Systematica Radius Enterprise Infrastructure» (объекты системы и соответствующие SQL таблицы).....	22
Введение.....	22
Шаги по установке	22
[Шаг 5.1] Настройка доступа	22
[Шаг 5.2] Установка структур данных Core	22
[Шаг 5.3] Установка и настройка Event Manager	22
[Шаг 5.4] Установка структур данных Common	23
[Шаг 5.5] Установка структур данных Balance	23
Проверка.....	23
[Шаг 6] Установка и настройка «Systematica Message Bus Daemon».....	25
Введение.....	25
Используемые директории	25
Пример конфигурационного файла	25
Установка.....	26
[Шаг 7] Установка и настройка «Systematica License Server».....	27
Введение.....	27
Используемые директории	27
Пример конфигурационного файла	27
Установка.....	27
После установки.....	27
[Шаг 8] Установка и настройка «Systematica Repository Server».....	29
Введение.....	29
Используемые директории	29
Зависимости.....	29
Пример конфигурационного файла	29
Установка.....	29
[Шаг 9] Установка и настройка «Systematica Jongleur Server»	30
Введение.....	30
Используемые директории	30
Зависимости.....	30
Пример конфигурационного файла	30
Установка.....	31
[Шаг 10] Установка и настройка «Systematica Market Data Server».....	32
Введение.....	32
Используемые директории	32
Зависимости.....	32
Пример конфигурационного файла	32
Установка.....	34

[Шаг 11] Установка и настройка «Systematica Database Adapter»	35
Введение.....	35
Используемые директории	35
Пример конфигурационного файла	35
Установка	36
[Шаг 12] Установка и настройка «Systematica PgMbProху»	37
Введение.....	37
Используемые директории	37
Зависимости.....	37
Пример конфигурационного файла	37
Установка.....	37
[Шаг 13] Установка и настройка PgBouncer	38
Введение.....	38
Установка.....	38
Настройка.....	38
Директории и файлы программного обеспечения «Systematica Radius Enterprise Infrastructure» на сервере после установки пакетов, настройки и запуска сервисов	40
Каталог с исполняемыми файлами	40
Каталог с настройками сервисов ("домашние каталоги сервисов").....	40
Каталог для файлов, используемых сервисами Systematica для своих нужд (например, файлы шаблонов для Repository Server).....	41
Каталог для файлов аудита и логов	42

О документе

Данный документ предназначен для технических администраторов платформы «Systematica Radius Enterprise».

По всем вопросам, связанным с установкой, настройкой и использованием системы, обращайтесь к разработчику для консультаций.

e-mail: support@systematica.ru

тел.: + 7 (495) 755-78-39

Введение

«Systematica Radius Enterprise Infrastructure» – серверное программное обеспечение платформы «Systematica Radius Enterprise», предназначенное для автоматизации фронт- и мидл- офис подразделений банков, инвестиционных, брокерских и страховых компаний в части учета операций на финансовых рынках (фондовый, валютный, денежный рынки и рынок производных финансовых инструментов), включая аналитическую обработку сделок, платежей и пр.

Требования к специалистам, производящим техническую установку серверного программного обеспечения «Systematica Radius Enterprise Infrastructure»

Установка программного обеспечения «Systematica Radius Enterprise Infrastructure» проводится:

- Сотрудниками Разработчика (Лицензиара), либо
- Обученными Разработчиком и сертифицированными Разработчиком сотрудниками Заказчика (Лицензиата).

Способы получения программного обеспечения «Systematica Radius Enterprise Infrastructure» для установки

Серверное программное обеспечение «Systematica Radius Enterprise Infrastructure» и его компоненты поставляются Заказчикам для установки на оборудование Заказчика после подписания с Заказчиком Лицензионного Договора.

Программное обеспечение «Systematica Radius Enterprise Infrastructure» доставляется (предоставляется) Заказчику одним из следующих способов:

- **На физическом носителе (компакт диск или флэшка).**
- **С интернет ресурса Разработчика (Лицензиара) после ввода выданного Заказчику Лицензиаром логина и пароля.**

Интернет-ресурс: support.systematica.ru

Состав файлов пакета установки (дистрибутива) программного обеспечения «Systematica Radius Enterprise Infrastructure» для установки

Ниже приведен список файлов, включая их названия и версии, актуальные на дату редакции данного Руководства по технической установке.

Файлы для установки на сервере базы данных

Директория `db-server` – пакеты для установки сервисов и библиотек

```
systematica-event-manager_0.1-3_amd64.deb
systematica-pg-mb-proxy_0.1.4-5_amd64.deb
systematica-postgresql-atrium-pgpro-std-15_0.3-1_amd64.deb
systematica-postgresql-db-adapter-pgpro-std-15_0.1-2_amd64.deb
systematica-postgresql-xml-pgpro-std-15_0.1-2_amd64.deb
```

Директория `db` – скрипты для создания и наполнения базы данных

```
install_PG_Core_024.sql
install_PG_Common_024.sql
install_PG_Balance_013.sql
check_EventManager.sql
```

Файлы для установки серверных компонент программного обеспечения «Systematica Radius Enterprise Infrastructure» на отдельный от СУБД сервер

Директория `app-server` – пакеты для установки сервисов и библиотек

Сервисы

```
systematica-db-adapter_2.4.0.283-5_amd64.deb
systematica-jongleur_1.6.0.2-5_amd64.deb
systematica-license-server_1.1.1.1-5_amd64.deb
systematica-market-data-server_1.6.0.1-5_amd64.deb
systematica-mbdaemon_1.4.0.0-5_amd64.deb
systematica-radius-server_2.5.0.1-5_amd64.deb
systematica-utils_0-5_amd64.deb
```

Зависимости и библиотеки

```
systematica-base_0-1_all.deb
systematica-sqlapi_5.2.3.j2-7_amd64.deb
```

Вспомогательные утилиты

```
systematica-mb-tools_0-5_amd64.deb
```


Краткий обзор архитектуры программного обеспечения «Systematica Radius Enterprise Infrastructure»

Серверные компоненты Программного обеспечения «Systematica Radius Enterprise Infrastructure» представляют набор Linux сервисов, которые могут быть установлены и запущены на одном или нескольких серверах в зависимости от конфигурации и настройки Программного обеспечения «Systematica Radius Enterprise Infrastructure» для конкретного Заказчика.

Все без исключений компоненты и сервисы Программного обеспечения «Systematica Radius Enterprise Infrastructure» и модули обмениваются данными по шине данных «Systematica Message Bus», которая является реализацией проприетарной шинной технологии, разработанной ООО «СИСТЕМАТИКА».

Подключение и информационный обмен с шиной «Systematica Message Bus» осуществляется по TCP/IP.

Для хранения данных используется СУБД «PostgreSQL».

Подробное описание в документе

«SystematicaRadiusEnterprise_TechnialArchitecture_09jul2024.pdf»

Утилита мониторинга работоспособности компонент программного обеспечения «Systematica Radius Enterprise Infrastructure»

Каждый серверный компонент Systematica Radius Enterprise Infrastructure имеет встроенный мини web сервер, который позволяет администраторам просматривать статистику, состояние и работоспособность сервиса с использованием Web Browser.

Для запроса статистики используется любой Internet Browser.

В строке URL необходимо указать адрес в формате

```
http://<ip/machine name>:<port>
```

Номер порта определяется в настройке каждого компонента.

[Шаг 1] Требования к системе. Проверка

Перед установкой убедитесь в том, что на оборудование установлено следующее программное обеспечение:

- **ОС: «Astra Linux» не ниже версии 1.7 или «Debian» версии не ниже 10 "buster";**

Проверить текущую версию операционной системы можно, выполнив следующую команду в консоли Linux:

```
cat /etc/os-release
```

Пример результата:

```
PRETTY_NAME="Astra Linux"  
NAME="Astra Linux"  
ID=astra  
ID_LIKE=debian  
ANSI_COLOR="1;31"  
HOME_URL="https://astralinux.ru"  
SUPPORT_URL="https://astralinux.ru/support"  
LOGO=astra  
VERSION_ID=1.7_x86-64  
VERSION_CODENAME=1.7_x86-64
```

- **СУБД: «PostgreSQL» не ниже 14 (рекомендуется 15 версия) - «Postgres Pro» или «ванильная» сборка от PGDG.**

Проверить текущую версию СУБД можно , выполнив следующую SQL-команду:

```
select version();
```

Пример результата:

```
PostgreSQL 15.7 on x86_64-pc-linux-gnu, compiled by gcc (AstraLinuxSE 8.3.0-6) 8.3.0, 64-bit
```

Дополнительно требуется, чтобы на сервере с СУБД был установлен и настроен PgBouncer (см. для информации <https://www.pgбouncer.org/>). Информация по установке и настройке приведена в разделе «Установка и настройка PgBouncer». Выполнять можно на любом этапе процесса установки системы независимо от других шагов.

[Шаг 2] Требования к правам пользователя операционной системы производящего установку ПО. Проверка

Пользователь, выполняющий установку и настройку системы, должен обладать правами суперпользователя в ОС и СУБД.

Проверить наличие необходимых прав можно, выполнив следующую команду в консоли Linux:

```
sudo -s
```

Если прав достаточно, то промт ввода поменяется с "<username>@<servername>" на "**root**@<servername>".

Вернитесь в сессию вашего пользователя, выполнив команду:

```
exit
```

Дополнительно, в некоторых Astra Linux (например, с уровнем защищённости Усиленный и выше) для установки пакетов и конфигурации сервисов нужна сессия с высоким уровнем целостности. Проверить уровень целостности текущей сессии можно, выполнив в консоли Linux команду:

```
pdp-id
```

Пример ожидаемого результата:

```
Уровень конф.=0(Уровень_0), Уровень целостности:63(Высокий), Категории=0x0(Нет)  
Роли=()
```

Структура файлов и папок

По умолчанию все серверные компоненты «Systematica Radius Enterprise» используют следующие директории:

ID в документации	Директория	Необходимые права	Описание
BIN_DIR	/opt/systematica/bin	Read	Папка для всех исполняемых файлов Systematica
CFG_SAMPLES_DIR	/opt/systematica/share/config-samples	Read	Папка с примерами конфигурационных файлов.
CFG_ROOT_DIR	/etc/opt/systematica	Read	Общая папка для всех файлов настроек конфигурации компонентов
CFG_DIR	CFG_ROOT_DIR/<type>[_<name>]	Read*	Папка файлов конфигурации одного экземпляра компонента
HOME_ROOT_DIR	/var/opt/systematica		Общая папка для всех используемых компонентами файлов (создаваемых в процессе работы приложений)
HOME_DIR	HOME_ROOT_DIR/<type>[_<name>]	Read/Write	Папка для файлов, используемых приложениями для своих нужд. Например файлы кэша данных Highway Broadcast Server'a
LOG_ROOT_DIR	/var/log/systematica/	Read/Write	Общая папка для всех файлов аудита и логов
LOG_DIR	LOG_ROOT_DIR/<type>[_<name>]	Read/Write	Папка для файлов аудита и логов
LOCK_DIR	/run/lock/systematica/<type>[_<name>]	Read/Write	Папка для механизма локфайлов, используемых для защиты от запуска нескольких экземпляров приложения

*Доступ на запись в папку с конфигами(/etc/opt/systematica) может требоваться для отдельных компонентов. Например, License Server при первом запуске.

В указанных папках каждый компонент, как правило, использует свою подпапку с именем, отображающим тип и экземпляр приложения. Далее в документе будут использоваться указанные в таблице ID директорий.

Например, папка для экземпляра «Test» компонента Systematica Jongleur будет:

“/etc/opt/systematica/Jongleur_Test”.

Стандартные директории компонентов могут быть переопределены переменными окружения:

Директория	Переменная окружения, переопределяющая путь по умолчанию
CFG_DIR	CONFIGURATION_DIRECTORY

CFG_ROOT_DIR	CONFIGURATION_ROOT_DIRECTORY
HOME_DIR	STATE_DIRECTORY
LOG_DIR	LOGS_DIRECTORY

Важно! При изменении путей по умолчанию, или настройке каких-либо дополнительных путей (например для файлов аудита) за пределами директорий компонентов по умолчанию, необходимо добавлять разрешения на доступ в конфигурационных файлах служб systemd соответствующих компонентов.

Установка компонентов «Systematica Radius Enterprise Infrastructure»

Все компоненты ПО Systematica Radius Enterprise Infrastructure» поставляются в виде отдельных пакетов (RPM для RHEL, DEB для Astra Linux и Debian).

Для их установки необходимы права администратора.

Все компоненты устанавливаются в виде системных служб (systemd services).

Установка пакетов происходит обычным способом

- для RHEL:

```
sudo dnf install <package-path-name.rpm>
```

- для DEB:

```
sudo apt install <package-path-name.deb>
```

Пакеты могут иметь зависимости на другие компоненты Systematica. Зависимости описаны в описании каждого компонента ниже.

Все компоненты требуют установки «базового» пакета Systematica (systematica-base-<version>.rpm). Данный пакет создает необходимую для всех компонентов структуру директорий и пользователя «systematica», из-под которого по умолчанию будут запускаться все сервисы Systematica. Данный пользователь ограничен в своих правах.

Пакеты также устанавливают unit-файлы для systemd services. Для именованных компонентов имя является обязательным параметром соответствующего unit-файла.

Согласно поставляемым unit-файлам, службы Systematica дополнительно ограничиваются в правах до прав исключительно на необходимые директории и функционал системы.

В случае необходимости использовать директории, отличные от используемых по умолчанию, необходимо отредактировать (по возможности используя drop-in) соответствующие unit-файлы и права пользователя (или сменить самого пользователя).

После установки пакета необходимо создать все необходимые директории для каждого экземпляра приложений. Для неименованных (существующих всегда только в одном экземпляре служб) эти папки создаются автоматически при установке пакета.

Пакеты устанавливают примеры конфигурационных файлов, которые необходимо отредактировать по необходимости и поместить в CFG_DIR соответствующих компонентов. Основной конфиг должен иметь имя “config.xml”. Некоторые компоненты могут иметь дополнительные опциональные или обязательные конфигурационные файлы, описанные в документации ниже.

Необходимыми директориями являются CFG_DIR, LOG_DIR, HOME_DIR (их имена формируются исходя из имени и типа компонента). Так же отдельные компоненты могут обращаться к другим папкам, расположенным в CFG_ROOT_DIR.

Зависимости на сторонние библиотеки

Как правило, компоненты Systematica требуют для своей работы библиотеки:

- `zlib (libz)`
- `libbsd` – В RHEL данная библиотека доступна для автоматической установки пакетным менеджером после включения EPEL (`sudo dnf install epel-release`), иначе её необходимо скачать и установить вручную.
- `libuuid`
- `libxml2`
- `libffi`
- `SQLAPI++` – Данная библиотека используется в компонентах ПО Systematica, где необходима работа с СУБД. Версия библиотеки поставляется Systematica.

Для работы с БД PostgreSQL SQLAPI++ требует библиотеку `libpq` версии 14 или новее.

Все эти зависимости указаны в зависимостях пакетов, и они будут доставлены автоматически при необходимости.

Описание настроек, стандартных для большинства компонентов

Введение

Настройки компонентов «Systematica Radius Enterprise Infrastructure» как правило представляют собой XML файл, расположенный по умолчанию по пути CFG_DIR/config.xml

Ниже описаны общие правила настроек, актуальные для большинства компонентов. Подробные настройки каждого компонента описаны в соответствующих разделах документа.

Пример конфигурационного файла

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <Description value="Sample Adapter" />
  <MessageBus daemon="Radius" />
  <Log level="3" depth="30" />
  <MonitorAgent name="Sample Adapter" path="Radius" port="8850" />
  <DBConnection>
    <Client value="POSTGRES" />
    <ConnectionString value="207.1.175.65@radius" />
    <User value="radius" />
    <Password value="XXXXXX" />
    <Options value="" />
  </DBConnection>
</Configuration>
```

Ниже представлено описание основных параметров из данного примера.

Описание

Данный параметр отвечает за описание компонента, видимое в веб статистике и мониторинге.

MessageBus

Раздел, отвечающих за подключение к шине Message Bus Daemon. По умолчанию (если настройки пустые), компонент будет подключаться к локальной шине. Важно не забыть открыть порт для доступа по TCP/IP. По данному порту используется проприетарный бинарный протокол. В случае использования прокси-серверов, они должны быть полностью «прозрачными», и передавать весь трафик “as-is”, не привнося в него ничего лишнего.

Настраиваемые параметры:

daemon – хост(или IP адрес) и порт для подключения. Указывается в формате `<hostname/IP>:<port>`.

Дополнительно могут быть заданы параметры **lan**, **network**, **service**, которые используются только при необходимости иметь более одной независимой шины на одной машине, и для явного указания используемого интерфейса\подсети. Должны быть пустыми в большинстве случаев.

Log

Данная секция задает уровень логирования компонента.

level = 1 – Только ошибки (error)

level = 2 – Предупреждения (warning) и ошибки

level = 3 – Информационные сообщения, предупреждения и ошибки

level = 4 – Полное логирование, включая отладочные сообщения. Не рекомендуется включать данный уровень без необходимости, т.к. он замедляет работу приложений и пишет огромные логи.

MonitorAgent

Данный раздел отвечает за мониторинг работы компонента:

name – отображаемое имя компонента в мониторе компонентов

path – раздел, к которому относится компонент (Highway, Common, Modular, Radius, и т.п.)

port – если задан (не равен 0), то на данном порту приложение будет отображать http страницу статистики работы компонента, которую можно использовать при поиске проблем и ошибок. Порты, обычно используемые для каждого компонента ПО Systematica, приведены в примерах соответствующих конфигурационных файлов. По умолчанию, веб статистика отключена (port=0).

DBConnection

Данный раздел описывает основные параметры подключения к БД. Нужен только в компонентах, работающих с СУБД (DB Adapter, Jongleur, и т.п.).

Client – тип БД, к которой подключаемся. На данный момент Linux версии компонентов работают только с POSTGRES.

ConnectionString – строка подключения к БД в формате <HOST>[\<DB>]@<DBName>

User – имя пользователя для подключения

Password – пароль (в зашифрованном виде)

Options – опции подключения

Шифрование паролей

Введение

Пароли, хранящиеся в конфигурационных файлах компонентов ПО Systematica, в большинстве случаев должны быть зашифрованы.

Для шифрования паролей используется утилита Systematica Encrypt Util, поставляемая в пакете systematica-utils.

Для шифрования пароля воспользуйтесь командой:

```
/opt/systematica/bin/encrypt-util <Путь> <Пароль>
```

Где <Путь> - это путь к конфигурационному файлу компонента, где будет использоваться пароль (без слеша в конце). На основании данного параметра строится ключ шифрования пароля.

Полученную в результате выполнения данной команды строку зашифрованного пароля необходимо скопировать в конфигурационный файл вручную.

Пример

Для получения зашифрованного пароля «foobar», который будет использован в сервисе DbAdapter_FooBar, вызываем команду:

```
/opt/systematica/bin/encrypt-util /etc/opt/systematica/DbAdapter_FooBar  
foobar
```

В результате выполнения получаем строку:

```
D5B0507D98C773D2F2F8C7AE051C2769
```

Это и есть зашифрованный пароль, который необходимо указать в параметре <DbConnection password="..."/> компонента.

[Шаг 3] Установка и настройка «Systematica DB Back-end Server Extensions»

Введение

«Systematica DB Back-end Server Extensions» - программное обеспечение сервиса, который поддерживает программные расширения для специализированной обработки данных, полученных из СУБД PostgreSQL

Установка

Для работы с Postgres Pro установите пакеты:

```
systematica-postgresql-atrium-pgpro-std-15-<version>.rpm (или .deb).  
systematica-postgresql-xml-pgpro-std-15-<version>.rpm (или .deb).  
systematica-postgresql-db-adapter-pgpro-std-15-<version>.rpm (или .deb).
```

Для работы с PostgreSQL (Debian) установите пакеты:

```
systematica-postgresql-atrium-pg15-<version>.rpm (или .deb).  
systematica-postgresql-xml-pg15-<version>.rpm (или .deb).  
systematica-postgresql-db-adapter-pg15-<version>.rpm (или .deb).
```

Настроек и папок у данных компонентов нет, после установки они подхватываются сервером Postgres как плагины.

[Шаг 4] Установка и настройка «Systematica Database Event Manager»

Введение	Systematica Event Manager - программное обеспечение сервиса, который реализовывает в базе данных Программного обеспечения «Systematica Radius Enterprise Infrastructure» внутренние очереди и обработчики событий из этих очередей.
Используемые директории	LOG_DIR = LOG_ROOT_DIR
Зависимости	<p>libpq</p> <p>Единственная конфигурация, которая нужна Event Manager – это параметры подключения к PostgreSQL. Поскольку Event Manager для подключения к серверу PostgreSQL пользуется библиотекой libpq, и соответственно для настройки подключения используются переменные окружения, поддерживаемые libpq, см. https://postgrespro.ru/docs/postgrespro/15/libpq-envars</p> <p>Конкретно, вероятно возникнет желание установить PGHOSTADDR (сервер должен быть локальным, то есть обычно это будет 127.0.0.1), PGUSER, PGPASSWORD. Это предлагается делать с помощью директивы Environment в systemd unit-файле, лучше всего с использованием drop-in.</p>
Пример конфигурационного файла	<p>Помещаем по пути /etc/systemd/system/event-manager@atrium.service.d/override.conf файл со следующим содержимым:</p> <pre>[Service] Environment="PGHOSTADDR=127.0.0.1" Environment="PGUSER=atrium" Environment="PGPASSWORD=atrium"</pre> <p>Указанные выше настройки являются настройками по умолчанию и уже заданы после установки.</p>
Установка	<p>Установите пакет systematica-event-manager-<version>.rpm (или .deb).</p> <p>Данный компонент является «именованным», т.е. допускает работу нескольких именованных экземпляров. Для настройки экземпляра сервиса необходимо:</p> <ul style="list-style-type: none">• выполнить регистрацию службы: <pre>sudo systemctl enable event-manager@<instance_name>.service</pre> <p>Обратите внимание, что запуск Event Manager возможен только после создания соответствующих таблиц (dbo."SettingValue", dbo."SystemEvent", dbo."SystemEventState") в базе данных.</p> <p>Запуск сервиса необходимо выполнить в процессе выполнения Шага 5 (Установка моделей данных) .</p>

[Шаг 5] Установка моделей данных Программного Обеспечения «Systematica Radius Enterprise Infrastructure» (объекты системы и соответствующие SQL таблицы)

Введение

«Systematica Radius Enterprise Infrastructure Database Data Structures & Procedures» – набор моделей данных для всех объектов, хранимых в базе данных программного обеспечения «Systematica Radius Enterprise Infrastructure», которые описываются в специализированном формате XML документов, и программного кода по обработке данных.

Шаги по установке

Базовая поставка БД «Systematica Radius Enterprise Infrastructure» содержит три скрипта, которые необходимо выполнить последовательно:

- *install_PG_Core_<version>.sql* - создание базы и наполнение базы основными объектами ядра;
- *install_PG_Common_<version>.sql* - наполнение базы бизнес-объектами общего модуля системы;
- *install_PG_Balance_<version>.sql* - наполнение базы объектами модуля Balance (расчет и ведение остатков);

<NNN> в именах файлов будет

Внимание! До выполнения скриптов на сервере с СУБД обязательно должен быть выполнен [Шаг 3] (Установка и настройка «Systematica DB Back-end Server Extensions»).

[Шаг 5.1] Настройка доступа

Перед выполнением скрипта *install_PG_Core_<version>.sql* в нём можно изменить некоторые параметры по умолчанию.

Все места в скрипте, которые могут кастомизироваться, отмечены меткой \$SETUP\$. Вы можете изменить следующие параметры:

- пароль пользователя-владельца БД atrium - по умолчанию *atrium*;
- пароль пользователя для импорт-адаптера (для обслуживания импорт-очереди) - по умолчанию *importadapter*;
- заменить имя БД atrium на другое имя - по умолчанию *atrium*;
- поменять параметры подключения к PgBouncer – по умолчанию порт *6432*, имя БД *atrium*, пароль пользователя atrium – *atrium*.

[Шаг 5.2] Установка структур данных Core

Выполните скрипт *install_PG_Core_<version>.sql* – выполнять от имени суперпользователя в СУБД.

Пример команды:

```
sudo --user postgres /opt/pgpro/std-15/bin/psql -q  
<install_PG_Core_024.sql |& tee ./install_PG_Core_024.sql.log
```

После выполнения скрипта проверьте файл лога *install_PG_Core_024.sql.log* – в нем не должно быть ошибок.

[Шаг 5.3] Установка и настройка Event Manager

Перед выполнением следующих скриптов необходимо стартовать сервис Systematica Event Manager (предварительно сервис должен быть установлен и настроен на [Шаге 4]).

В консоли Linux выполнить команду:

```
sudo systemctl start event-manager@atrium.service
```

Выполнение команды должно пройти без ошибок. В случае ошибок причины можно посмотреть с помощью команды:

```
sudo systemctl status event-manager@atrium.service
```

или более детально :

```
sudo journalctl -u event-manager@atrium.service -e
```

Проверьте успешность запуска Event Manager:

```
sudo systemctl status event-manager@atrium.service
```

Проверьте возможность обращения к сервису из БД:

```
/opt/pgpro/std-15/bin/psql -h localhost -q -U atrium -d atrium -f  
check_EventManager.sql
```

Ожидаемый результат выполнения скрипта – отсутствие ошибок (в этом случае можно переходить к следующему шагу по устновке скриптов БД), например:

```
psql:check_EventManager.sql:7: ИНФОРМАЦИЯ: Server [127.0.0.1],  
database [atrium], server time [2024-07-08 10:46:15]
```

```
psql:check_EventManager.sql:20: ЗАМЕЧАНИЕ: Checking Event  
Manager...
```

```
psql:check_EventManager.sql:22: ИНФОРМАЦИЯ: DONE. (server time  
2024-07-08 10:46:15)
```

[Шаг 5.4] Установка структур данных Common

Выполните скрипт *install_PG_Common_<version>.sql* - выполнять от имени пользователя atrium и в контексте БД atrium

Пример команды:

```
/opt/pgpro/std-15/bin/psql -h localhost -q -U atrium -d atrium -f  
install_PG_Common_024.sql |& tee ./install_PG_Common_024.sql.log
```

После выполнения скрипта проверьте файл лога *install_PG_Common_024.sql.log* – в нем не должно быть ошибок.

[Шаг 5.5] Установка структур данных Balance

Выполните скрипт *install_PG_Balance_<version>.sql* - выполнять от имени пользователя atrium и в контексте БД atrium

Пример команды:

```
/opt/pgpro/std-15/bin/psql -h localhost -q -U atrium -d atrium -f  
install_PG_Balance_013.sql |& tee ./install_PG_Balance_013.sql.log
```

После выполнения скрипта проверьте файл лога *install_PG_Common_024.sql.log* – в нем не должно быть ошибок.

Проверка

По окончании процесса установки вы можете проверить факт наличия таблиц и функций в базе данных, выполнив запросы ниже в БД atrium (от имени пользователя atrium).

Список сущностей / объектов «Systematica Metadata Processor» в базе данных, являющихся частью ПО «Systematica Radius Enterprise Infrastructure» (более 400 объектов):

```
select "ObjectType_Code" from "ObjectType" ot order by  
"ObjectType_Code";
```

Список таблиц в базе данных, являющихся частью ПО «Systematica Radius Enterprise Infrastructure» (более 880 объектов):

```
select * from pg_catalog.pg_tables where schemaname not in ('pg_catalog', 'information_schema') order by schemaname, tablename;
```

Список процедур и функций, созданных скриптами или автоматически сгенерированных метапроцессором в процессе установки системы в схемах 'dbo', 'calc', 'import' (выборка процедур в схеме 'dbo' возвращает более 5 тыс. записей):

```
select proname from pg_catalog.pg_proc where pronamespace = to_regnamespace('dbo') order by proname;
```

```
select proname from pg_catalog.pg_proc where pronamespace = to_regnamespace('calc') order by proname;
```

```
select proname from pg_catalog.pg_proc where pronamespace = to_regnamespace('import') order by proname;
```


[Шаг 6] Установка и настройка «Systematica Message Bus Daemon»

Введение

Systematica Message Bus Daemon - программное обеспечение сервиса, который реализует высокопроизводительную шину для организации обмена сообщениями в режиме реального времени между всеми компонентами и сервисами программного обеспечения «Systematica Radius Enterprise Infrastructure», включая Модули. Работает полностью в памяти.

Используемые директории

CFG_DIR = CFG_ROOT_DIR/*mbDaemon*

LOG_DIR = LOG_ROOT_DIR/*mbDaemon*

HOME_DIR = HOME_ROOT_DIR/*mbDaemon*

Пример конфигурационного файла

Конфигурационный файл компонента расположен по пути:

CFG_DIR\config.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <Description value="Message Bus Daemon"/>
  <ClientHeartbeatPeriod value="30"/>
  <ClientTimeout value="90"/>
  <ListenPort value="7700"/>
  <TimerPeriod value="5"/>
  <SlowConsumerTimeout value="600"/>
  <HttpListenPort value="8800"/>
  <HttpListenHost value=""/>
  <DefaultLan value=""/>
  <Log level="3" depth="30"/>
</Configuration>
```

Описание параметров	
ClientHeartbeatPeriod	Интервал между отправляемыми heartbeat сообщениями (в секундах).
ClientTimeout	Таймаут (в секундах) для отключения клиентов, не отвечающих на heartbeat'ы и не присылающих никакие данные за этот период.
ListenPort	Порт, на котором Message Bus Daemon слушает подключение от других компонентов. По умолчанию 7700. В случае изменения дефолтного значения, необходимо поменять порт подключения у всех клиентов шиины.
TimerPeriod	Период (в секундах), с которым проверяется статус клиентов.
SlowConsumerTimeout	Если компонент в течение данного интервала не успевает забрать всю накопленную очередь сообщений, Message Bus Daemon принудительно его отключит. Используется как защита от накапливания гигантских очередей сообщений, когда очередь растёт быстрее, чем клиент успевает её забирать.
HttpListenPort	Порт веб статистики компонента. 0 – не использовать веб статистику.
HttpListenHost	Подсеть, откуда доступна веб статистика. Если параметр не задан, то вебстатистика доступна с любого IP. Если указан localhost или 127.0.0.1 – только локально.

DefaultLan

Интерфейс, на котором слушаются подключения. По умолчанию пусто (с любых IP).

Установка

Установите пакет `systematica-mbdaemon-<version>.rpm` (или `.deb`).

Проверьте существование папки `CFG_DIR` и наличие файла `config.xml` в ней, а также наличие папок `HOME_DIR` и `LOG_DIR`, при необходимости создайте их и выдайте права.

Сервис Message Bus Daemon поддерживает сокетную активацию (автоматический запуск службы при попытке подключиться к его порту). Для включения этого режима запуска выполните команду:

```
sudo systemctl enable --now mbdaemon.socket
```

В случае использования нестандартного порта для подключения к шине, необходимо указать используемый порт не только в настройках компонента (`config.xml`), но и в `unit`-файле `"mbdaemon.socket"` (по возможности используя `drop-in`).

[Шаг 7] Установка и настройка «Systematica License Server»

Введение «Systematica License Server» программное обеспечение сервиса, который реализует сервер лицензий для активации программного обеспечения «Systematica Radius Enterprise Infrastructure» и Модулей.

Используемые директории

CFG_DIR = CFG_ROOT_DIR/LicenseServer
LOG_DIR = LOG_ROOT_DIR/LicenseServer
HOME_DIR = HOME_ROOT_DIR/LicenseServer

Пример конфигурационного файла

Конфигурационный файл компонента должен быть расположен по пути:
CFG_DIR\config.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <Description value="License Server for Linux"/>
  <MessageBus daemon="localhost"/>
  <Log level="3" depth="30"/>
  <MonitorAgent name="License Server" path="Common" port="8801"/>
  <UniqueUserPolicy value="N"/>
  <ClientHeartbeatPeriod value="30"/>
  <ClientTimeout value="300"/>
</Configuration>
```

Описание параметров	
UniqueUserPolicy	Политика выдачи лицензий для пользователей с одинаковой позицией. Актуально для работы приложения Radius Terminal.
ClientHeartbeatPeriod	Период опроса клиентов, получивших лицензии.
ClientTimeout	Если за данный интервал клиент не отвечает на запросы, то он считается отключенным, и лицензия освобождается.
MessageBus	Данный компонент всегда должен подключаться к локальной шине Message Bus. Если указать другое значение, оно будет проигнорировано. Допускается указание произвольного порта для подключения, например: «localhost:12345»

Установка Компонент должен быть установлен на той же машине, где и шина данных. Установите пакет systematica-license-server-<version>.rpm (или .deb).

Проверьте существование папки CFG_DIR и наличие файла config.xml в ней, а так же наличие папок HOME_DIR и LOG_DIR, при необходимости создайте их и выдайте права.

Данный компонент при первом запуске требует доступ на запись в данную директорию для генерации ключа файлов лицензий. Для этого рекомендуем сделать пользователя systematica owner'ом этой директории.

Для регистрации службы выполните команду:

```
sudo systemctl enable --now license-server.service
```

После установки При первом запуске License Server должен сгенерировать ключ лицензии в файле CFG_DIR/license.txt.

Данный файл необходимо прислать в Systematica для генерации лицензий.

После получения заполненного файла лицензий, необходимо заменить им имеющийся файл.

Перезапуск компонента при этом не требуется, новый файл должен подхватиться автоматически.

[Шаг 8] Установка и настройка «Systematica Repository Server»

Введение

Systematica User Authorization & UI Templates Repository Server - программное обеспечение сервиса, который реализует единый централизованно обновляемый репозиторий экранных форм и бизнес логики форм для предоставления этого репозитория рабочему месту пользователя.

Используемые директории

```
CFG_DIR = CFG_ROOT_DIR/RadiusServer{<instance_name>}
LOG_DIR = LOG_ROOT_DIR/ RadiusServer{<instance_name>}
HOME_DIR = HOME_ROOT_DIR/ RadiusServer{<instance_name>}
```

Зависимости

-

Пример конфигурационного файла

Конфигурационный файл компонента расположен по пути:

CFG_DIR/config.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <Description value="Radius Server"/>
  <MessageBus daemon="radius"/>
  <Log level="3" depth="30"/>
  <MonitorAgent name="Radius Server" path="Radius" port="8895"/>
  <SystemMode value="ATRIUM"/>
</Configuration>
```

Описание параметров

SystemMode

Режим работы компонента. На данный момент поддерживается только режим «ATRIUM».

Установка

Установите пакет `systematica-radius-server-<version>.rpm` (или `.deb`).

Проверьте существование папки `CFG_DIR` и наличие файла `config.xml` в ней, а так же наличие папок `HOME_DIR` и `LOG_DIR`, при необходимости создайте их и выдайте права.

Скопируйте шаблоны UI, файлы шаблонов печати, файлы изображений в соответствующие подпапки `HOME_DIR`.

Для регистрации службы выполните команду:

```
sudo systemctl enable --now radius-server.service
```

[Шаг 9] Установка и настройка «Systematica Jongleur Server»

Введение

Systematica Jongleur - программное обеспечение сервиса, который обеспечивает публикацию иерархических сообщений из базы данных программного обеспечения «Systematica Radius Enterprise Infrastructure» в шину для организации обмена сообщениями с другими компонентами и сервисами программного обеспечения «Systematica Radius Enterprise Infrastructure», включая Модули.

Используемые директории

CFG_DIR = CFG_ROOT_DIR/Jongleur_<instance_name>

LOG_DIR = LOG_ROOT_DIR/Jongleur_<instance_name>

HOME_DIR = HOME_ROOT_DIR/Jongleur_<instance_name>

Зависимости

Пакет systematica-sqlapi – Библиотека SQLAPI для работы с БД

Пример конфигурационного файла

Основной конфигурационный файл компонента расположен по пути:

CFG_DIR/config.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration>
  <Description value="Jongleur Server Test"/>
  <MessageBus daemon="radius"/>
  <Log level="3" depth="30"/>
  <MonitorAgent name="Jongleur Server (Test)" path="Radius"
port="8875"/>
  <TrackTable value="MessageBusCmd"/>
  <PollingCmd value="SELECT 1"/>
  <DataReadTemplate value="select * from %s where %s"/>
  <UseRangesMinimumCmds value="50"/>
  <ThrottleTimeout value="800"/>
  <UnfinishedMsgGroupTTL value="30"/>
  <CmdsProcessPortion value="400"/>
  <SoftReconnectTimeout value="1440"/>
</Configuration>
```

Описание параметров

TrackTable	Имя командной таблицы, в которую помещаются команды, выполняемые компонентом. Jongleur регулярно вычитывает содержимое данной таблицы. Данная таблица создается скриптом БД, поставляемым Systematica.
PollingCmd	Шаблон SQL запроса для «пинга» БД.
DataReadTemplate	Шаблон SQL для начитывания данных из произвольных таблиц БД.
UseRangesMinimumCmds	Порог количества команд, для обработки которого будут использованы диапазоны вместо перечисления (т.е. синтаксис id > X and id < Y вместо перечисления id IN (id1, id2, id3...idN). Значение по умолчанию 50.
ThrottleTimeout	Интервал (в мс) между опросами командной таблицы, если после предыдущего запроса командная таблица была пуста. Иначе данные запрашиваются сразу после выполнения начитанных ранее команд.

UnfinishedMsgGroupTTL	Время жизни групповой записи, которая еще не закрыта (в минутах). По умолчанию 30.
CmdsProcessPortion	Количество команд, начитываемых и выполняемых за один цикл работы Jongleur. По умолчанию 400.
SoftReconnectTimeout	Интервал (в минутах) принудительного реконнекта к БД в профилактических целях. По умолчанию 14400 (24 часа).

Установка

Установите пакет `systematica-jongleur-<version>.rpm` (или `.deb`).

Компонент потребует установки пакета `systematica-sqlapi`.

Данный компонент является «именованным», т.е. допускает работу нескольких именованных экземпляров. Для настройки экземпляра сервиса необходимо:

- вручную создать необходимые папки с выбранным именем экземпляра
- дать права на них
- создать на основе примера конфигурационный файл (`CFG_DIR/config.xml`)
- выполнить регистрацию службы:

```
sudo systemctl enable --now jongleur@<instance_name>.service
```

[Шаг 10] Установка и настройка «Systematica Market Data Server»

Введение	Systematica Market Data Server - программное обеспечение сервиса, который обеспечивает работу с внешними источниками рыночной информации.
Используемые директории	CFG_DIR = CFG_ROOT_DIR/MarketDataServer LOG_DIR = LOG_ROOT_DIR/MarketDataServer HOME_DIR = HOME_ROOT_DIR/MarketDataServer
Зависимости	Пакет systematica-sqlapi – Библиотека SQLAPI для работы с БД.

Пример конфигурационного файла

Конфигурационный файл компонента расположен по пути:

CFG_DIR\config.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <Description value="Market Data Server"/>
  <MessageBus daemon="radius"/>
  <Log level="3" depth="30"/>
  <MonitorAgent name="Market Data Server" path="Radius"
port="8915"/>
  <DBConnection>
    <Client value="POSTGRES"/>
    <ConnectionString value="207.1.175.65@radius"/>
    <User value="radius"/>
    <Password value="XXXXXX"/>
    <Options value=""/>
  </DBConnection>
  <PollingCmd value="SELECT 1"/>
  <UseRangesMinimumCmds value="50"/>
  <ThrottleTimeout value="800"/>
  <CmdsProcessPortion value="400"/>
  <SoftReconnectTimeout value="1440"/>
  <DBCmdThrottleTimeout value="300"/>
  <DBRecordThrottleTimeout value="3000"/>
  <BondRecalcTime value="00:00"/>
  <HighwayPublishers>
    <Publisher name="MDS">
      <MessageBus daemon="radius"/>
    </Publisher>
    <Publisher name="/MDS" compressionPeriod="1000"
compressionSlots="10">
      <MessageBus daemon=" radius "/>
    </Publisher>
  </HighwayPublishers>
</Configuration>
```

Описание параметров

PollingCmd

Шаблон SQL запроса для «пинга» БД.

UseRangesMinimumCmds	<p>Порог количества команд, для обработки которого будут использованы диапазоны, вместо перечисления(т.е. синтаксис <code>id > X and id < Y</code> вместо перечисления <code>IN (id1, id2, id3...idN)</code>.</p> <p>Значение по умолчанию 50.</p>
ThrottleTimeout	<p>Интервал (в мс) между опросами командной таблицы, если после предыдущего запроса командная таблица была пуста. Иначе данные запрашиваются сразу после выполнения начитанных ранее команд.</p>
CmdsProcessPortion	<p>Количество команд, начитываемых и выполняемых за один цикл работы MarketDataServer. По умолчанию 400.</p>
SoftReconnectTimeout	<p>Интервал (в минутах) принудительного реконнекта к БД в профилактических целях. По умолчанию 14400 (24 часа).</p>
DBCmdThrottleTimeout	<p>Интервал (в мс) между отправляемыми в БД порциями команд с данными (публикация данных в БД). По умолчанию 300.</p>
DBRecordThrottleTimeout	<p>Минимальный интервал (в мс) между двумя публикациями в БД обновлений одного инструмента. По умолчанию 3000.</p>
BondRecalcTime	<p>Время пересчета рассчитываемых сервисом полей облигаций.</p>
HighwayPublishers	<p>Данный раздел позволяет настроить источники данных (Highway Publisher).</p> <p>Каждый источник должен иметь уникальное имя в рамках всей шины Message Bus, куда он подключается.</p> <p>name – имя источника данных (Highway Publisher). Не может быть пустым, рекомендуется использовать только заглавные буквы.</p> <p>compressionPeriod, compressionSlots – эти два параметра всегда должны задаваться в паре, они отвечают за прореживание данных, если оно необходимо. Если параметры не заданы или значение хотя бы одного из них равно 0 – прореживание не используется, все данные публикуются сразу (real-time).</p> <p>Если оба параметра заданы, то <code>compressionPeriod</code> задает минимальный интервал (в мс) между двумя последовательными обновлениями одного инструмента. А <code>compressionSlots</code> задает количество «слотов», на которые разбиваются все инструменты. Обновления инструментов, попавших в один «слот», публикуются одновременно, а обновления разных слотов распределены равномерно по периоду <code>compressionPeriod</code>. Т.е. это механизм, позволяющий распределить нагрузку, публикуя данные порциями.</p> <p>Имена источников Highway, в которых используется прореживание данных, рекомендуем начинать с символа «/».</p> <p>Например: <code>compressionPeriod="1000" compressionSlots="10"></code></p> <p>При такой настройке, один инструмент будет обновляться не чаще чем раз в секунду, а обновления разных наборов инструментов будут публиковаться раз в 100мс.</p> <p>Так же каждый источник имеет свою настройку подключения к шине Message Bus, что позволяет одному компоненту публиковать данные в разные шины.</p>

Имена источников в данном случае так же должны быть уникальными в рамках компонента.

Установка

Установите пакет `systematica-market-data-server-<version>.rpm` (или `.deb`).

Компонент потребует установки пакета `systematica-sqlapi`.

Проверьте существование папки `CFG_DIR` и наличие файла `config.xml` в ней, а так же наличие папок `HOME_DIR` и `LOG_DIR`, при необходимости создайте их и выдайте права.

Для регистрации службы выполните команду:

```
sudo systemctl enable --now market-data-server.service
```

[Шаг 11] Установка и настройка «Systematica Database Adapter»

Введение

Systematica Database Adapter - программное обеспечение сервиса, который обеспечивает управление и выполнение запросов (полученных из шины от компонент и сервисов программного обеспечения «Systematica Radius Enterprise Infrastructure» включая Модули) к базе данных программного обеспечения «Systematica Radius Enterprise Infrastructure».

Используемые директории

CFG_DIR = CFG_ROOT_DIR/DbAdapter_<instance_name>

LOG_DIR = LOG_ROOT_DIR/DbAdapter_<instance_name>

HOME_DIR = HOME_ROOT_DIR/DbAdapter_<instance_name>

Пример конфигурационного файла

Конфигурационный файл компонента расположен по пути:

CFG_DIR\config.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <Description value="Database Adapter"/>
  <MessageBus daemon="radius"/>
  <Log level="3" depth="30"/>
  <MonitorAgent name="Database Adapter" path="Radius" port="8905"/>
  <DbConnection client="POSTGRES" connstring="server@base"
user="user" password="pwd"/>
  <DbOptions></DbOptions>
  <DbConnectSetup></DbConnectSetup>
  <DbPollingStatement>select 1</DbPollingStatement>
  <DbConnectionPool size="10" disconnect_threshold="5"
idle_timeout="600" reconnect_timeout="0" stop_timeout="60"
command_timeout="3600" query_threshold="120"/>
  <DbQueueLimits></DbQueueLimits>
  <DbCacheTTL value="120"/>
  <DbResultsetLimit value="-1"/>
  <TimingAudit value="N"/>
</Configuration>
```

Описание параметров

DbConnection	Настройки соединения к БД. Атрибуты client, connstring, user, password соответствуют настройкам соединения, описанным в разделе стандартных настроек
DbOptions	Опции соединения к БД
DbConnectSetup	Скрипт, выполняющийся при каждом открытии соединения. Опционально.
DbPollingStatement	Скрипт, проверяющий работоспособность соединения. По умолчанию select 1.
DbConnectionPool	Параметры пула соединений. Имеет следующие атрибуты: size = размер пула соединений; disconnect_threshold = допустимое количество «мёртвых» соединений; idle_timeout = допустимый период неактивности соединения, в секундах. При превышении его соединение закрывается, и

	<p>открывается при необходимости. При 0 закрывается сразу же, при -1 – не закрывается никогда;</p> <p>reconnect_timeout = периодичность переоткрытия активных соединений, в секундах. При 0 – соединение не переоткрывается;</p> <p>stop_timeout = таймаут ожидания закрытия соединения при остановке адаптера;</p> <p>command_timeout = таймаут для выполнения запроса по умолчанию в секундах;</p> <p>query_treshold = таймаут для записи предупреждения о долго выполняющейся команде в лог;</p>
DbQueueLimits	<p>Очереди запросов и их ограничения на заполнение пула соединений. Задается в формате: Queue1=N; Queue2=N2; Queue3 = P%. Доля пула соединений указывается либо в штуках, либо в процентах. Запросы, которые не указали желаемую очередь, попадают в DEFAULT.</p>
DbCacheTTL	<p>Настройка времени хранения кешированных запросов – в секундах.</p>
DbResultsetLimit	<p>Ограничение на максимальный объем данных, возвращаемый запросом, для отправки в Message Bus.</p>
TimingAudit	<p>Включает подробное логирование запросов и времени их исполнения.</p>

Установка

Установите пакет `systematica-database-adapter-<version>.rpm` (или `.deb`).

Компонент потребует установки пакета `systematica-sqlapi`.

Данный компонент является «именованным», т.е. допускает работу нескольких именованных экземпляров. Для настройки экземпляра сервиса необходимо:

- вручную создать необходимые папки с выбранным именем экземпляра
- дать права на них
- создать на основе примера конфигурационный файл (`CFG_DIR/config.xml`)
- выполнить регистрацию службы:

```
sudo systemctl enable --now database-adapter@<instance_name>.service
```

[Шаг 12] Установка и настройка «Systematica PgMbProxy»

Введение	Systematica PostgreSQL to MessageBus Proxy – компонент, который обеспечивает работу SQL-функций MessageBus_Publish и MessageBus_Request.
Используемые директории	CFG_DIR = CFG_ROOT_DIR/PgMbProxy_<instance_name> LOG_DIR = LOG_ROOT_DIR/PgMbProxy_<instance_name> HOME_DIR = HOME_ROOT_DIR/PgMbProxy_<instance_name>
Зависимости	(нет)
Пример конфигурационного файла	Основной конфигурационный файл компонента расположен по пути: CFG_DIR/config.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration>
  <MessageBus daemon="example.com" service="1234"/>
  <Log level="3" depth=""/>
  <HttpDiagnostics host="" port="8802"/>
  <MonitorAgent name="PgMbProxy" path="Radius"/>
  <Dashboards aggregateTimeout="1">
  </Dashboards>
</Configuration>
```

Описание параметров

Dashboards

Список Дашбордов (Dashboards – умных отчётов в реальном времени), которые обслуживаются текущим компонентом. Список разделяется через ‘;’.

Установка

Установите пакет systematica-pg-mb-proxy-<version>.rpm (или .deb).

Данный компонент является «именованным», т.е. допускает работу нескольких именованных экземпляров. Для настройки экземпляра сервиса необходимо:

- вручную создать необходимые папки с выбранным именем экземпляра
- дать права на них
- создать на основе примера конфигурационный файл (CFG_DIR/config.xml)
- выполнить регистрацию службы:

```
sudo systemctl enable --now pg-mb-proxy@<instance_name>.service
```

Сервис поддерживает сокетную активацию (автоматический запуск службы при попытке подключиться к его порту). Для включения этого режима запуска выполните команду: “sudo systemctl enable --now pg-mb-proxy@<instance_name>.socket”.

[Шаг 13] Установка и настройка PgBouncer

Введение

Для организации процесса технического логирования операций в базе данных в «Systematica Radius Enterprise Infrastructure» применяется функционал автономных транзакций.

Для работы функционала используется линк на эту же БД (atrium), а для экономии количества одновременно открытых сессий к БД используется PgBouncer (<https://www.pgouncer.org/>) в режиме `pool_mode=transaction`.

Требуется, чтобы данный компонент был установлен и настроен на сервере БД.

Установка

Установите пакет pgbouncer:

```
sudo apt install pgbouncer
```

Проверка установки:

```
apt list -a --installed 'pgbouncer*'
```

Зарегистрируйте службу (но не запускайте до выполнения настроек):

```
sudo systemctl enable pgbouncer.service
```

Настройка

До запуска `pgbouncer` необходимо выполнить ряд настроек.

Редактирование параметров в файле настроек.

Откройте на редактирование файл с настройками:

```
sudoedit /etc/pgbouncer/pgbouncer.ini
```

Проверьте и измените при необходимости следующие настройки:

```
;; fallback connect string // работаем с локальным сервером БД  
* = host=localhost
```

```
;;; Where to wait for clients // ограничиваем только локальными  
подключениями, принимаем соединения по порту 6432  
listen_addr = localhost  
listen_port = 6432
```

```
;;; Authentication settings // данные параметры менять не следует  
(оставить те, что в файле) – приведены для информации для  
следующего шага  
auth_type = md5  
auth_file = /etc/pgbouncer/userlist.txt
```

```
;; When server connection is released back to pool: // возвращаем  
соединение в пул доступных сразу после завершения транзакции  
;; transaction - after transaction finishes  
pool_mode = transaction
```

После выполнения изменений сохраните файл.

Редактирование параметров в файле со списком разрешенных пользователей

Добавить в список разрешенных пользователей запись для *atrium*. Список разрешенных пользователей находится в файле, указанном в параметре *auth_file* в *pgbouncer.ini* (по умолчанию «*/etc/pgbouncer/userlist.txt*»).

Открыть файл на редактирование:

```
$ sudoedit /etc/pgbouncer/userlist.txt
```

Добавить строку:

```
"atrium" "$password"
```

где *\$password* - значение, возвращаемое при выполнении SQL-команды (команду выполнить в БД от лица супер-пользователя)

```
SELECT passwd FROM pg_shadow where username = 'atrium'
```

После выполнения изменений сохраните файл.

После внесения изменений в настроечные файлы необходимо запустить (перезапустить) *pgbouncer*

```
sudo systemctl restart pgbouncer.service
```

Для проверки корректности выполненных настроек попробуйте подключиться к *pgbouncer* от имени пользователя *atrium*

```
$ /opt/pgpro/std-15/bin/psql -h localhost -p 6432 -U atrium
```

Директории и файлы программного обеспечения «Systematica Radius Enterprise Infrastructure» на сервере после установки пакетов, настройки и запуска сервисов

Каталог с исполняемыми файлами

```
/opt/systematica/  
├── bin  
│   ├── DbAdapter  
│   ├── encrypt-util  
│   ├── event-manager  
│   ├── Jongleur  
│   ├── LicenseServer  
│   ├── MarketDataServer  
│   ├── mbDaemon  
│   ├── mb-listen  
│   ├── mb-publish  
│   ├── PgMbProxy  
│   ├── RadiusServer  
│   ├── Systematica.PostgreSQL.Atrium.so  
│   ├── Systematica.PostgreSQL.DbAdapter.so  
│   └── Systematica.PostgreSQL.Xml.so  
├── lib  
│   └── x86_64-linux-gnu  
│       └── libsqlapi.so  
└── share  
    ├── config-samples  
    │   ├── DbAdapter.xml  
    │   ├── jongleur.xml  
    │   ├── LicenseServer.xml  
    │   ├── MarketDataServer.xml  
    │   ├── mbDaemon.xml  
    │   ├── PgMbProxy.xml  
    └── RadiusServer.xml
```

Каталог с настройками сервисов ("домашние каталоги сервисов")

```
/etc/opt/systematica/  
├── DbAdapter_atrium  
│   └── config.xml  
├── Jongleur_atrium  
│   └── config.xml  
├── LicenseServer  
│   ├── config.xml  
│   └── license.txt  
├── MarketDataServer  
│   └── config.xml  
├── mbDaemon  
│   └── config.xml  
├── PgMbProxy_atrium  
│   └── config.xml  
└── RadiusServer  
    └── config.xml
```


Каталог для файлов, используемых сервисами Systematica для своих нужд (например, файлы шаблонов для Repository Server)

```
/var/opt/systematica
├── DbAdapter_atrium
├── Jongleur_atrium
│   ├── local
│   │   └── agentKey
├── LicenseServer
│   ├── local
│   │   ├── agentKey
│   │   └── RunStat_LicenseServer.txt
├── MarketDataServer
│   ├── local
│   └── agentKey
├── mbDaemon
├── PgMbProxy_atrium
└── RadiusServer
    ├── DATA
    │   ├── Images
    │   │   └── Radius_Images_2021-02-18.dat
    │   ├── PrintTemplates
    │   ├── ReportTemplates
    │   ├── RFL
    │   ├── Templates
    │   │   └── Atrium_Templates_2024-06-23_028a.dat
    └── local
        ├── agentKey
        └── RunStat_RadiusServer.txt
```

Каталог для файлов аудита и логов

```
/var/log/systematica
├── DbAdapter_atrium
│   ├── 2024-07-06_DbAdapter_atrium.log
│   ├── 2024-07-07_DbAdapter_atrium.log
│   ├── 2024-07-08_DbAdapter_atrium.log
│   └── RunStat_DbAdapter_atrium.txt
├── Jongleur_atrium
│   ├── 2024-07-06_Jongleur_atrium.log
│   ├── 2024-07-07_Jongleur_atrium.log
│   └── 2024-07-08_Jongleur_atrium.log
├── LicenseServer
│   ├── 2024-07-04_LicenseServer.log
│   └── 2024-07-07_LicenseServer.log
├── MarketDataServer
│   ├── 2024-07-06_RadiusMDServer.log
│   ├── 2024-07-07_RadiusMDServer.log
│   ├── 2024-07-08_RadiusMDServer.log
│   └── Audit
│       ├── ATRIUM.Audit.20240706.txt
│       ├── ATRIUM.Audit.20240707.txt
│       └── ATRIUM.Audit.20240708.txt
├── mbDaemon
│   ├── 2024-07-03_mbDaemon.log
│   └── 2024-07-07_mbDaemon.log
├── PgMbProxy_atrium
│   ├── 2024-06-25_PgMbProxy_atrium.log
│   ├── 2024-06-26_PgMbProxy_atrium.log
│   ├── 2024-06-27_PgMbProxy_atrium.log
│   ├── 2024-06-28.log
│   ├── 2024-06-28_PgMbProxy_atrium.log
│   ├── 2024-07-03.log
│   ├── 2024-07-03_PgMbProxy_atrium.log
│   ├── CrashLog_PgMbProxy_atrium.txt
│   └── RunStat_PgMbProxy_atrium.txt
└── RadiusServer
    ├── 2024-07-03_RadiusServer.log
    └── 2024-07-04_RadiusServer.log
```