РСХБ-ИНТЕХ

Электронная Площадка РСХБ (ЭП РСХБ).

Руководство по установке

Всего 12 листов

Москва, 2022

Оглавление

1 Общие сведения

1.1 О документе

Настоящий документ представляет собой руководство по установке ЭП РСХБ с использованием установочных файлов, входящих в комплект поставки.

1.2 О продукте

Электронная площадка РСХБ (далее, ЭП РСХБ) обеспечивает взаимодействие и организует совместную работу различных прикладных информационных систем Банка с внешними информационными системами или источниками информации.

1.3 Состав продукта

Для реализации сервисов ЭП РСХБ используются open source компоненты:

- Linux - версия ядра не ниже 2.6.x
- Java – 8 и выше
- React
- Nginx/HAProxy -- версия не ниже 1.18.0

Для реализации механизмов криптографической защиты информации используется сертифицированный программный продукт – КриптоПро JCP 2.0[1] (сборка 39014 и выше).

2 Системные требования

2.1 Требования к аппаратному обеспечению

2.1.1 Минимальные характеристики и конфигурация серверов

Производитель заявляет, что для промыленной эксплуатации ЭП РСХБ в минимальной концифигурации требуется шесть выделенных серверов:

[1] КриптоПро JCP 2.0 реализует российские криптографические стандарты и разработано в соответствии со спецификацией JCA (Java Cryptography Architecture)

Таблица 1

Назначение	Количество серверов	Характеристики сервера		
		CPU	RAM, GB	Disk space, GB
Сервер базы данных	1	4	8	Минимум 50 GB HDD (без учета дальнейшего прироста)
Сервер приложений	2	4	8	50 GB HDD
Web-сервер	1	2	4	50 GB HDD

2.1.2 Рекомендуемые характеристики и конфигурация серверов

Требования к промышленной нагрузке рассчитываются индивидуально в зависимости от типа и количества используемых модулей и потоков.

2.2 Требования к программному обеспечению

Для успешной установки ЭП РСХБ на выделенных серверах должно быть установлено соответствующее программное обеспечение. Для каждого типа сервера указан свой набор ПО и требования к нему.

2.2.1 Сервер базы данных

Таблица 2

Компонент	Версия
Linux	Версия ядра не ниже 2.6.x
PostgreSQL	Не ниже 9.5

2.2.2 Сервер приложений

Таблица 3

Компонент	Версия
Linux	Версия ядра не ниже 2.6.x
Java JRE	8 и выше
КриптоПро JCP 2.0	Сборка 39014 и выше
Nginx/HAProxy	Не ниже 1.18.0

Компонент	Версия
Stunnel	Не ниже 5.3.0

2.2.3 Web-сервер

Таблица 4

Компонент	Версия
Linux	Версия ядра не ниже 2.6.x
Nginx	Не ниже 1.18.0
Stunnel	Не ниже 5.3.0

2.3 Требования к программному обеспечению пользователей

При разработке и обновлении продукта могут изменяться как frontend, так и backend решения. К рабочим станциям таких специалистов предъявляются отдельные требования к установленному ПО.

2.3.1 Рабочая станция web-разработчика

Таблица 5

Компонент	Версия
Операционная система	Windows: 7/8/10 MacOS Ubuntu – не ниже 16.04.6 LTS
Visual Studio Code	Latest
Node JS	14.17.6
NPM	6.14.15
Браузер (совместимость)	Google Chrome, Yandes Browser -- Latest
Postman	Latest

2.3.2 Рабочая станция backend разработчика

Таблица 6

Компонент	Версия
Операционная система	Windows: 7/8/10 MacOS Ubuntu – не ниже 16.04.6 LTS
Java JRE	8 и выше

Компонент	Версия
IDE (IntelliJ IDEA)	Latest
Maven	Не ниже 3.3.3
КриптоПро JCP 2.0	Сборка 39014 и выше
Postman	Latest
SOAP UI	Latest
DBeaver	Не ниже 6.2.1
Браузер (совместимость)	Google Chrome, Yandes Browser -- Latest

3 Комплект поставки

В состав поставки программного продукта входят:

- Установочный комплект файлов

Таблица 7

Компонент	Дистрибутив	Хэш-сумма
Сервис авторизации OAuth2.0[2]	oauth2-authorization-server.jar	
Сервис FileApi	file-api.jar	
Сервис СМЭВ 3	smev3.0-service.jar	
Сервис ЭЦП	crypto-pro-service-1.0.jar	

- Документация:
  - Руководство по установке (настоящий документ)
  - Руководство администратора

4 Чек-лист перед началом установки

- Выделены сервера под установку продукта в количестве, указанном в п. 2.1
- Сервера имеют сетевую связность

[2] В отдельных случаях допускается поставка сервиса авторизации на базе KeyCloak. В таком случае готовится отдельный дистрибутив keycloak-authorization-server.jar. Решение об изменении технологии авторизации принимается на этапе согласования поставки продукта.

- Для установки и настройки программного продукта на каждом сервере создана техническая учетная запись с административными (sudo) правами
- На каждом сервере установлено ПО в соответствии с требованиями, изложенными в п. 2.2
- Комплект поставки дистрибутивов продукта соответствует п. 3
- Хэш-суммы файлов дистрибутивов соответствуют значениям, указанным в сопроводительной записке к комплекту поставки

5 Установка продукта

Ниже описывается процесс установки ЭП РСХБ для случая минимальной конфигурации (см. п. .2.1.1). Процесс установки продукта на ином аппаратном обеспечении (для промышленной нагрузки) — идентичен описываемому в настоящем документе.

5.1 Сервис авторизации

Сервис авторизации устанавливается в единственном экземпляре на **одном** сервере приложений. При этом не имеет значения на каком из двух серверов будет установлен сервис авторизации.

Для установки сервиса:

- Авторизуйтесь на сервере приложения под учетной записью с sudo правами
- Проверьте, что на сервере установлена база данных PostgreSQL нужной версии и консоль для работы с базой данных pgAdmin
- Подключитесь к базе данных
- Создайте учетную запись пользователя БД **oauth**
- Проверьте, что пользователь создался
- Авторизуйтесь под учетной записью *oauth*
- Создайте базу данных **oauth**
- В корневом каталоге sudo-пользователя создайте файл конфигурации /application.yml

```
touch application.yml
```

- В файле *application.yml* задайте настройки для подключения к базе данных *oauth*

```
spring:
```

```
  datasource:
    hikari:
      connection-test-query: SELECT 1
      minimum-idle: 1
      maximum-pool-size: 5
    driver-class-name: org.postgresql.Driver
    url: ${PG_URL}
    username: ${PG_USER}
    password: ${PG_PASSWORD}
```

- Задайте порт для сервера

```
server.port: ${PORT}
```

- В файле *oauth2-authrization-server.sh* задайте переменную USER_JAVA

```
USER_JAVA= $JAVA_HOME/bin/java
```

- Запустите сервис авторизации

```
$ oauth2-authorization-server.sh start
Started AuthorizationServer
```

При первом запуске сервис создаст в БД все необходимые объекты.

- Откройте браузер, в адресной строке введите *http://localhost:${PORT}/*, далее нажмите **Enter**
- В случае успешной установки откроется консоль управления сервисом авторизации
- Для проверки работоспособности сервиса, авторизуйтесь под дефолтной учетной записью (логин/пароль) **admin/admin**

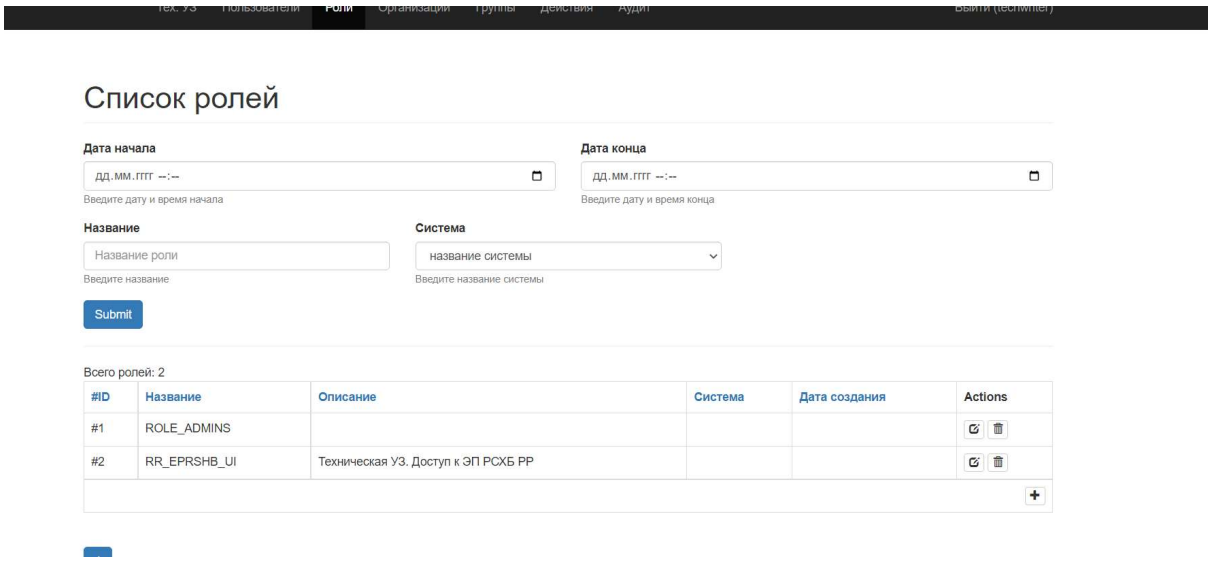
5.2 Сервис FileApi

Сервис FileApi устанавливается в единственном экземпляре на **одном** сервере приложений. При этом не имеет значения на каком из двух серверов будет установлен сервис FileApi.

Для установки сервиса:

- Авторизуйтесь в сервисе авторизации
- Создайте роль FILE_API. Для этого в верхнем меню навигации выберите Роли, далее нажмите на знак **+**

Рисунок 1



- В *Название роли* укажите **FILE_API**, далее нажмите **Submit**
- Авторизуйтесь на сервере приложения под учетной записью с sudo правами
- Проверьте, что на сервере установлена база данных PostgreSQL нужной версии и консоль для работы с базой данных pgAdmin
- Подключитесь к базе данных
- Создайте учетную запись пользователя БД **file_api**
- Проверьте, что пользователь создался
- Авторизуйтесь под учетной записью *file_api*
- Создайте базу данных **file_api**
- В корневом каталоге sudo-пользователя отредактируйте файл конфигурации /application.yml, добавив в него настройки для подключения к БД *file_api*

```
spring:
  datasource:
    hikari:
      connection-test-query: SELECT 1
      minimum-idle: 1
      maximum-pool-size: 5
    driver-class-name: org.postgresql.Driver
    url: ${PG_URL}
    username: ${PG_USER}
    password: ${PG_PASSWORD}
```

- Задайте порт для сервера

```
server.port: ${PORT}
```

- Заполните в файле конфигурации application.yml настройки для проверки авторизации

```
spring:
```

```
  security:
    oauth2:
      resourceserver:
        jwt:
          jwk-set-uri: http://${OAUTH_SERVICE}/.well-known/jwks.json
```

- В файле file-api.sh установите переменную USER_JAVA

```
USER_JAVA= $JAVA_HOME/bin/java
```

- Запустите сервис FileApi

```
$ file-api.sh start
$ Started FileApiApplication
```

- При первом запуске сервис создаст в БД все необходимые объекты.

5.3 Сервис СМЭВ 3

- Данный сервис размещается на втором сервере приложений;
- Для использования сервиса СМЭВ 3 создайте отдельную роль SMEV_USER внутри сервиса авторизации.
  У сервиса-потребителя должна быть соответствующая роль;
- Создайте пользователя БД **smev_adapter**
- Создайте БД **smev_adapter** и схему **smev_messages**
- Заполните в файле конфигурации application.yml настройки для подключения к БД

```
spring:
  datasource:
    hikari:
      connection-test-query: SELECT 1
      minimum-idle: 1
      maximum-pool-size: 5
    driver-class-name: org.postgresql.Driver
    url: ${PG_URL}
    username: ${PG_USER}
    password: ${PG_PASSWORD}
```

- Установите в application.yml порт

```
server.port: ${PORT}
```

- Заполните в файле конфигурации application.yml настройки для проверки авторизации:

```
spring:
  security:
    oauth2:
      resourceserver:
        jwt:
          jwk-set-uri: http://${OAUTH_SERVICE}/.well-known/jwks.json
```

- Заполните в файле конфигурации application.yml настройки для отправки сообщений в СМЭВ:

```
smev:
  adapter:
    mnemonic: 771602
    testMessage: false
  outDir: ${SMEV_ADAPTER_OUTPUT_FILES}/out/
  storageDir: ${SMEV_ADAPTER_OUTPUT_FILES}/in/
```

- В файле smev3.0-service.sh установите переменную USER_JAVA

```
USER_JAVA= $JAVA_HOME/bin/java
```

- Запустите сервис

```
$ smev3.0-service.sh start
$ Started Application
```

При первом запуске сервис создаст в БД все необходимые объекты. Сервис запустится, в логе должна отображаться строчка «Started Application».

5.4 Сервис ЭЦП

- Данный сервис размещается на первом сервере приложений;
- Для использования сервиса ЭЦП создайте две отдельных роли внутри сервиса авторизации
  - **HMAC_USER** – возможность использования метки целостности;
  - **CRYPTO_USER** – возможность использования ЭЦП и шифрования. *Примечание: У сервиса-потребителя должны быть созданы идентичные роли*
- Задайте в *application.yml* порт сервера

```
server.port: ${PORT}
```

- В файле конфигурации *application.yml* задайте настройки для проверки авторизации:

```
spring:
  security:
    oauth2:
      resourceserver:
        jwt:
          jwk-set-uri: http://${OAUTH_SERVICE}/.well-known/jwks.json
```

- В файле конфигурации *application.yml* задайте настройки для ключа по умолчанию и других доступных ключей:

```
  enc: false
  default-private-key:
    alias: ${DEFAULT_ALIAS}
    pswd: ${DEFAULT_PSWD}
  private-keys:
    - ${EMAIL_1};${KEY_ALIAS};${KEY_PSWD}
```

- В файле *crypto-pro-service.sh* задайте переменную USER_JAVA

```
USER_JAVA= $JAVA_HOME/bin/java
```

- Запустите сервис ЭЦП

```
$ crypto-pro-service.sh start
$ Started Application
```