

# Электронная Площадка РСХБ (ЭП РСХБ).

Руководство по установке

Всего 11 листов

---

# Оглавление

<b>1</b>	<b>ОБЩИЕ СВЕДЕНИЯ .....</b>	<b>3</b>
1.1	О ДОКУМЕНТЕ .....	3
1.2	О ПРОДУКТЕ .....	3
1.3	СОСТАВ ПРОДУКТА .....	3
<b>2</b>	<b>СИСТЕМНЫЕ ТРЕБОВАНИЯ.....</b>	<b>3</b>
2.1	ТРЕБОВАНИЯ К АППАРАТНОМУ ОБЕСПЕЧЕНИЮ.....	3
2.1.1	<i>Минимальные характеристики и конфигурация серверов.....</i>	<i>3</i>
2.1.2	<i>Рекомендуемые характеристики и конфигурация серверов .....</i>	<i>4</i>
2.2	ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ .....	4
2.2.1	<i>Сервер базы данных.....</i>	<i>4</i>
2.2.2	<i>Сервер приложений .....</i>	<i>4</i>
2.2.3	<i>Web-сервер.....</i>	<i>4</i>
2.3	ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ ПОЛЬЗОВАТЕЛЕЙ .....	5
2.3.1	<i>Рабочая станция web-разработчика .....</i>	<i>5</i>
2.3.2	<i>Рабочая станция backend разработчика.....</i>	<i>5</i>
<b>3</b>	<b>КОМПЛЕКТ ПОСТАВКИ.....</b>	<b>6</b>
<b>4</b>	<b>ЧЕК-ЛИСТ ПЕРЕД НАЧАЛОМ УСТАНОВКИ .....</b>	<b>6</b>
<b>5</b>	<b>УСТАНОВКА ПРОДУКТА.....</b>	<b>6</b>
5.1	СЕРВИС АВТОРИЗАЦИИ.....	7
5.2	СЕРВИС FILEAPI.....	8
5.3	СЕРВИС СМЭВ 3 .....	9
5.4	СЕРВИС ЭЦП.....	10

## 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 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<sup>1</sup> (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 промышленной эксплуатации ЭП РСХБ в минимальной конфигурации требуется шесть выделенных серверов:

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)

<sup>1</sup> КриптоПро JCP 2.0 реализует российские криптографические стандарты и разработано в соответствии со спецификацией JCA (Java Cryptography Architecture)

Назначение	Количество серверов	Характеристики сервера		
		CPU	RAM, GB	Disk space, GB
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
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

Компонент	Версия
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, Yandex Browser — Latest
Postman	Latest

### 2.3.2 Рабочая станция backend разработчика

Таблица 6

Компонент	Версия
Операционная система	Windows: 7/8/10 Mac OS Ubuntu — не ниже 16.04.6 LTS
Java JRE	8 и выше
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, Yandex Browser — Latest

### 3 Комплект поставки

В состав поставки программного продукта входят:

- Установочный комплект файлов

Таблица 7

Компонент	Дистрибутив	Хэш-сумма
Сервис авторизации OAuth2.0 <sup>2</sup>	oauth2-authorization-server.jar	Определяется на момент поставки
Сервис FileApi	file-api.jar	Определяется на момент поставки
Сервис СМЭВ 3	smev3.0-service.jar	Определяется на момент поставки
Сервис ЭЦП	crypto-pro-service-1.0.jar	Определяется на момент поставки

- 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
- 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.

<sup>2</sup> В отдельных случаях допускается поставка сервиса авторизации на базе Keycloak. Тогда готовится отдельный дистрибутив *keycloak-authorization-server.jar*. Решение об изменении технологии авторизации принимается на этапе согласования поставки продукта.

## 5.1 Сервис авторизации

Сервис авторизации устанавливается в единственном экземпляре на **одном** сервере приложений. При этом не имеет значения на каком из двух серверов приложений 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 создан
- 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}
```

- Порт для сервера

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

- В файле *oauth2-authorization-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}](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

Список ролей

Дата начала: DD.MM.YYYY --> | Дата конца: DD.MM.YYYY -->

Название: Название роли | Система: название системы

Submit

Всего ролей: 2

ID	Название	Описание	Система	Дата создания	Actions
#1	ROLE_ADMINS				[i] [d]
#2	RR_EPRSHB_UI	Техническая УЗ. Доступ к ЭП РОСБ			[i] [d]

- В поле *Название* 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 создан
- 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}
```



- Порт для сервера

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

- 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**
- В файле конфигурации *application.yml* задайте:
  - 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}
```

- Порт

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

- Настройки для проверки авторизации

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

- 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` создайте 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 строка «*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*
- В файле `application.yml` задайте:
  - Порт сервера

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

- Настройки для проверки авторизации:

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

- 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
```