

---

Инструкция по инсталляции  
серверных приложений системы  
"ROBIN PLATFORM v2"

*Выпуск 2.0.1*

ROBIN RPA Team

дек. 24, 2020

<b>1</b>	<b>Установка серверной части (вне докера)</b>	<b>2</b>
1.1	Установка и настройка PostgreSQL в CentOS 7 . . . . .	2
1.2	Установка и настройка FlyWay 6.4.3 . . . . .	3
1.3	Установка, настройка и запуск WildFly 18 в standalone режиме . . . . .	4
1.4	Настройка шифрования пакета аутентификации . . . . .	5
1.5	Настройка логирования Wildfly . . . . .	6
1.6	Развертывание приложений RMS 2.0 . . . . .	7
1.7	Создание демона для Wildfly . . . . .	7
<b>2</b>	<b>Установка серверной части в docker</b>	<b>9</b>
2.1	Установка и конфигурирование СУБД PostgreSQL (1_postgres_setup) . . . . .	9
2.2	Создание/обновление схем БД (2_migrate_db) . . . . .	10
2.3	Установка и конфигурирование сервера приложений WildFly (3_wildfly_setup) . . . . .	10
2.4	Развертывание приложения RMS на сервер WildFly (4_deploy_rms) . . . . .	11
<b>3</b>	<b>Nginx SSL proxy</b>	<b>12</b>
<b>4</b>	<b>RDP Manager</b>	<b>14</b>
4.1	Установка . . . . .	14
4.2	Логирование . . . . .	14



---

## Установка серверной части (вне докера)

---

### 1.1 Установка и настройка PostgreSQL в CentOS 7

Добавляем репозиторий PostgreSQL в yum и устанавливаем все пакеты PostgreSQL 11:

```
sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-  
↪redhat-repo-latest.noarch.rpm  
  
sudo yum install postgresql11 postgresql11-server postgresql11-contrib
```

Инициализируем базу данных PostgreSQL следующей командой:

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

Добавляем СУБД в автозапуск и запускаем:

```
sudo systemctl enable postgresql-11  
sudo systemctl start postgresql-11
```

Задать пароль для пользователя postgres:

```
su postgres
```

Запускаем интерпретатор командной строки PostgreSQL

```
psql
```

И устанавливаем пароль для пользователя postgres

```
\password postgres
```

В `/var/lib/pgsql/11/data/postgresql.conf` в параметре `listen_addresses` указываем IP, который СУБД будет слушать

В `/var/lib/pgsql/11/data/pg_hba.conf` указываем адреса с которых можно будет подключаться к СУБД. В качестве метода проверки указываем `md5`.

Перезагружаем СУБД для принятия изменений:

```
sudo systemctl restart postgresql-11
```

Суперпользователем postgres выполнить первоначальную инициализацию БД:

Создать пользователя user\_robin

```
CREATE ROLE user_robin WITH  
LOGIN  
NOSUPERUSER  
NOCREATEDB  
NOCREATEROLE  
INHERIT  
NOREPLICATION  
CONNECTION LIMIT -1  
PASSWORD 'xxxxxx';
```

Создать БД robin и назначить владельцем user\_robin

```
CREATE DATABASE robin_2  
WITH  
OWNER = user_robin  
ENCODING = 'UTF8'  
CONNECTION LIMIT = -1;
```

Создать пользователей схем

```
CREATE USER user_orch LOGIN PASSWORD 'Qwerty123';  
CREATE USER user_quartz LOGIN PASSWORD 'Qwerty123';  
CREATE USER user_rpm LOGIN PASSWORD 'Qwerty123';
```

Выполнить скрипт initial.sql с созданием схем и функций C

Внимание: в файле initial.sql содержатся скрипты указанные выше, их необходимо закомментировать или удалить.

## 1.2 Установка и настройка FlyWay 6.4.3

Скачиваем Flyway с официального сайта по ссылке: <https://flywaydb.org/documentation/commandline/#download-and-installation>

Сконфигурировать инструмент миграции FlyWay: conf/flyway.conf:

```
flyway.url=jdbc:postgresql://localhost:5432/robin_2  
flyway.user=user_robin  
flyway.password=Qwerty123  
flyway.schemas=flyway  
flyway.baselineVersion=1.0  
flyway.baselineDescription=Base Migration
```

Далее распаковываем скрипты из архива sql.zip в каталог flyway/sql/

Выполнить:

```
flyway baseline
```

Запускаем миграцию базы данных в Flyway следующей командой:

```
flyway migrate
```

Статус работы Flyway и скриптов можно посмотреть командой

```
flyway info
```

### 1.3 Установка, настройка и запуск WildFly 18 в standalone режиме

Скачиваем издание Java EE Full & Web Distribution с официального сайта WildFly <https://wildfly.org/downloads/> (приложения протестированы на версии wildfly-18.0.1.Final)

На сервере должна быть установлена JDK не ниже 8\_202!

Распакуем скачанный дистрибутив в директорию /opt/wildfly/

```
# mkdir -p /opt/wildfly/  
# unzip wildfly*.zip -d /opt/wildfly/
```

Активируем и обновляем пароль для учетной записи admin. Для этого запускаем \$wildfly\_home/bin/add-user.sh В интерактивном режиме пошагово выполняем следующие действия. 1. Выбираем пункт «а» и в поле username указываем admin. 2. Выбираем пункт «b», затем пункт «а».

- При выводе сообщения - *What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[PowerUser,BillingAdmin,]* - нажимаем Enter
- При выводе сообщения - *Is this new user going to be used for one AS process to connect to another AS process? e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls. yes/no?* - отвечаем «no»

Из дистрибутива RMS 2.0 распаковываем архив rms2.zip в директорию \$wildfly\_home/modules/ В данном архиве находятся wildfly-модуль с файлами свойств приложений. В файлах свойств /rms2/main/properties/rms2.properties, /rms2/main/properties/quartz2.properties, /rms2/main/properties/rpm.properties прописываем свои параметры для подключения к RabbitMQ и LDAP.

Скачиваем драйвер PostgreSQL для JDBC по следующей ссылке <https://jdbc.postgresql.org/download.html>

Запускаем сервер приложений WildFly следующей командой:

```
$wildfly_home/bin/standalone.sh
```

Выполняем команду, указав путь до скачанного драйвера:

```
$wildfly_home/bin/jboss-cli.sh -c command="module add --name=org.postgresql --resources=/opt/  
↳ postgresql-42.2.12.jar --dependencies=javax.api,javax.transaction.api"
```

Выполняем команду:

```
$wildfly_home/bin/jboss-cli.sh -c command="/subsystem=datasources/jdbc-  
↳ driver=postgresql:add(driver-name=postgresql,driver-module-name=org.postgresql,driver-xa-  
↳ datasource-class-name=org.postgresql.xa.PGXADatasource)"
```

Выполняем следующие команды, изменив адрес СУБД с localhost и значения --password на свои соответственно:

```
$wildfly_home/bin/jboss-cli.sh -c command="data-source add  
--name=OrchestratorDS2  
--jndi-name=java:jboss/datasources/OrchestratorDS2  
--driver-name=postgresql  
--connection-url=jdbc:postgresql://localhost/robin_2?currentSchema=orchestrator&  
↳ ApplicationName=RobinManagementServer  
--user-name=user_orch  
--password=Qwerty123"
```

```
--use-java-context=true  
--min-pool-size=5  
--max-pool-size=30  
--pool-prefill=true  
--statistics-enabled=true"
```

```
$wildfly_home/bin/jboss-cli.sh -c command="data-source add  
--name=QuartzDS2  
--jndi-name=java:jboss/datasources/QuartzDS2  
--driver-name=postgresql --connection-url=jdbc:postgresql://localhost/robin_2?  
↪currentSchema=quartz&ApplicationName=RobinManagementServer --user-name=user_quartz  
--password=Qwerty123  
--use-java-context=true  
--min-pool-size=5  
--max-pool-size=30  
--pool-prefill=true  
--statistics-enabled=true"
```

```
$wildfly_home/bin/jboss-cli.sh -c command="data-source add  
--name=PackageManagerDS2  
--jndi-name=java:jboss/datasources/PackageManagerDS2  
--driver-name=postgresql --connection-url=jdbc:postgresql://localhost/robin_2?  
↪currentSchema=package_manager --user-name=user_rpm  
--password=Qwerty123  
--use-java-context=true  
--min-pool-size=5  
--max-pool-size=30  
--pool-prefill=true  
--statistics-enabled=true"
```

Меняем ip для интерфейсов:

```
$wildfly_home/bin/jboss-cli.sh -c command="/interface=management/:write-attribute(name=inet-  
↪address,value=${jboss.bind.address.management:0.0.0.0})"  
$wildfly_home/bin/jboss-cli.sh -c command="/interface=public/:write-attribute(name=inet-  
↪address,value=${jboss.bind.address:0.0.0.0})"
```

Перезапускаем WildFly следующей командой:

```
$wildfly_home/bin/jboss-cli.sh --connect command=:reload
```

## 1.4 Настройка шифрования пакета аутентификации

Для того чтобы клиентское приложение RMS могло проходить аутентификацию при подключении к серверу RMS на машине пользователя должен быть установлен сертификат, используемый для шифрования пакета аутентификации. Чтобы решить проблему с его отсутствием на машине пользователя сервер WildFly предоставляет точку входа, откуда скачивается данный сертификат. RMS расшифровывает полученный от клиента пакет приватным ключом.

1) Создать хранилище и сгенерировать пару ключей указав свои параметры

```
keytool -genkey -alias orchestrator-client -keyalg RSA -keysize 2048 -sigalg SHA256withRSA -  
↪validity 3650 -keystore orchestrator-client.jks -storepass Qwerty123 -keypass Qwerty123 -  
↪dname "cn=RMS, ou=Robin, o=ITBR, L=Moscow, ST=Moscow, C=RU"
```

2) Экспортировать публичный сертификат из хранилища

```
keytool -v -exportcert -alias orchestrator-client -keystore orchestrator-client.jks -rfc -file  
↪orchestrator-client.cer
```

3) Скопировать полученные файлы в директории сервера Wildfly

```
# mv orchestrator-client.cer $wildfly_home/certificate/client/  
# mv orchestrator-client.jks $wildfly_home/certificate/server/
```

4) Создать Wildfly Handler - endpoint для скачивания сертификата. Настройка осуществляется командами jboss-cli

```
$wildfly_home/bin/jboss-cli.sh -connect
```

В интерактивном режиме JBoss CLI выполняем следующие команды:

```
/subsystem=undertow/configuration=handler/file=client-cert:add(path="{jboss.home.dir}/  
↪certificate/client", directory-listing="true")  
/subsystem=undertow/server=default-server/host=default-host/location="/rms/client-cert  
↪":add(handler="client-cert")  
quit
```

5) Проверить и исправить свойства приложения rms2.properties В конфигурационном файле \$wildfly\_home/modules/rms2/main/properties/rms2.properties в блоке настройки криптографии прописываются соответствующие сертификатам параметры.

Значения по умолчанию:

```
# Блок настроек сервиса криптографии  
# Настройка позволяющие включать/выключать сервис.  
crypto.enable=true  
# Путь к каталогу в файловой системе до keystore, в котором хранится пара закрытый и публичный ↪  
↪ключ.  
# Публичный ключ из пары используется клиентом RMC для шифрования данных пакета аутентификации.  
crypto.keyStore.path=/opt/wildfly/certificate/server/  
# Название keystore в котором хранится пара закрытый и публичный ключ.  
# Публичный ключ из пары используется клиентом RMC для шифрования данных пакета аутентификации.  
crypto.keyStore.name=orchestrator-client.jks  
# Тип keystore (JKS или PKCS12). JKS по умолчанию.  
crypto.keyStore.type=JKS  
# Пароль от keystore.  
crypto.keyStore.password=Qwerty123  
# Алиас пары закрытый ключ/публичный сертификат, хранящихся в keystore.  
crypto.keyPair.alias=orchestrator-client  
# Пароль для пары закрытый ключ/публичный сертификат, хранящихся в keystore.  
crypto.keyPair.password=Qwerty123
```

В crypto.keyStore.path указывается абсолютный путь

## 1.5 Настройка логирования Wildfly

Открыть файл \$wildfly\_home/standalone/configuration/standalone.xml

По умолчанию ротация логов происходит ежедневно без удаления старых логов:

```
<periodic-rotating-file-handler name="FILE">  
  <formatter>  
    <named-formatter name="PATTERN"/>  
  </formatter>  
  <file relative-to="jboss.server.log.dir" path="server.log"/>  
  <suffix value=".yyyy-MM-dd"/>  
</periodic-rotating-file-handler>
```

Необходимо заменить этот блок настроек следующим блоком (добавить две новые строки с тегами rotate-size и max-backup-index:

```
<periodic-size-rotating-file-handler name="FILE">
  <formatter>
    <named-formatter name="PATTERN"/>
  </formatter>
  <file relative-to="jboss.server.log.dir" path="server.log"/>
  <suffix value=".yyyy-MM-dd"/>
  <rotate-size value="300m"/>
  <max-backup-index value="256"/>
</periodic-size-rotating-file-handler>
```

В этом случае ротация будет происходить после того, как размер файла логов превысит 300 Мбайт, а максимальное число архивных логов равно 256, при достижении которого старые файлы логов будут удаляться. Месторасположение файлов логов в домашнем каталоге Wildfly: `./standalone/log`.

Если вы хотите изменить месторасположение файлов логов, то необходимо заменить строку:

```
<file relative-to="jboss.server.log.dir" path="server.log"/>
```

следующей строкой:

```
<file path="[Абсолютный путь к каталогу с логами]/server.log"/>
```

После всех изменений перезапустите Wildfly.

## 1.6 Развертывание приложений RMS 2.0

Загрузить приложения в Wildfly командой, указав путь до файлов `rms2.war`, `rpm.war`

```
$wildfly_home/bin/jboss-cli.sh --connect --command="deploy /opt/rms2.war --force"
$wildfly_home/bin/jboss-cli.sh --connect --command="deploy /opt/rpm.war --force"
```

Теперь можно подключить клиента RMC

## 1.7 Создание демона для Wildfly

Предполагается что домашняя директория сервера Wildfly - `/opt/wildfly/`. Если другая, то необходимо это учесть в скриптах и конфигурационных файлах

Создаем пользователя для запуска демона Wildfly

```
# useradd -r -d /opt/wildfly -s /sbin/nologin wildfly
```

Назначаем созданного пользователя владельцем директории Wildfly

```
# chown -RH wildfly:wildfly /opt/wildfly
```

Создаем сервис Systemd для WildFly, копируя пример из дистрибутива

```
# mkdir /etc/wildfly
# cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
```

Содержимое `wildfly.conf`, если Wildfly запускается в `standalone` режиме

```
# The configuration you want to run
WILDFLY_CONFIG=standalone.xml

# The mode you want to run
```

```
WILDFLY_MODE=standalone  
  
# The address to bind to  
WILDFLY_BIND=0.0.0.0
```

Копируем сценарий launch.sh в /opt/wildfly/bin/. Этот файл необходим для запуска WildFly как службы.

```
# cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/
```

Копируем служебный модуль WildFly в каталог конфигурации Systemd

```
# cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /usr/lib/systemd/system/
```

Создаем каталог для хранения PID-файла для службы WildFly

```
# mkdir /var/run/wildfly/  
# chown -R wildfly:wildfly /var/run/wildfly/
```

Запускаем службу WildFly

```
# systemctl enable wildfly  
# systemctl start wildfly
```

Перезапустить Wildfly можно командой

```
# systemctl restart wildfly
```

---

## Установка серверной части в docker

---

Скрипты написаны на Python 3 и выполняют следующие автоматизированные операции:

1. Установку и конфигурирование базы данных PostgreSQL
2. Запуск sql-скриптов создания и обновления схем БД
3. Установку и конфигурирование сервера приложений WildFly
4. Развертывание приложения RMS на WildFly

Скрипты \*.py вместе с необходимыми для выполнения файлами сгруппированы по папкам, запускаются из соответствующих папок. Для первичной установки RMS скрипты выполняются в последовательности, указанной в названии папок (postgres\_setup.py, migrate\_db.py, wildfly\_setup.py, deploy\_rms.py) Если требуется обновить RMS, из артефакта новой версии, запускаются скрипты migrate\_db.py и deploy\_rms.py

---

**Примечание:** Для установки Postgresql вне докера перейдите в раздел - Ручная установка и первичная настройка PostgreSQL в CentOS 7, затем в раздел - Создание/обновление схем БД (2\_migrate\_db)

---

### 2.1 Установка и конфигурирование СУБД PostgreSQL (1\_postgres\_setup)

В файле config/postgres\_setup.config указываются параметры установки БД (по умолчанию запуск в докер контейнере, имя докер контейнера robin-postgres, имя сети robin)

В папке images находится первоначально сконфигурированный докер образ PostgreSQL. В папке build - файлы для создания этого образа

Для запуска выполнить скрипт postgres\_setup.py

В ходе установки будет установлен пароль суперпользователя "postgres" - postgres. Так же будут созданы пользователи "user\_robin", "user\_orch", "user\_quartz" с паролем Qwerty123

Данные пароли при необходимо поменять и внести правки в конфигурационные файлы migrate\_db.config, wildfly\_setup.config

## 2.2 Создание/обновление схем БД (2\_migrate\_db)

Создание/обновление схем БД осуществляется инструментом миграции FlyWay

В файле `config/migrate_db.config` находятся параметры запуска команд FlyWay (запуск FlyWay в докере, параметры доступа к БД и т.д.)

В папке `images` находится официальный докер образ `flyway:6.1.1`

В папке `sql` - список всех миграций БД. (будут применены только новые, по отношению к текущей версии схемы БД)

Для запуска выполнить скрипт `migrate_db.py`

## 2.3 Установка и конфигурирование сервера приложений WildFly (3\_wildfly\_setup)

В файле `config/wildfly_setup.config` указываются параметры запуска WildFly (запуск WildFly в докере, параметры `datasource` и переменные среды `JAVA_OPTS`)

Необходимо выделить достаточное количество ресурсов для сервера приложений. Для этого в переменных `JAVA_OPTS` нужно изменить следующие параметры:

1. `/Xmx2G` - максимальный размер оперативной памяти выделенной виртуальной машине JVM
2. `/XX:MaxMetaspaceSize=512M` - максимальный размер памяти выделенной под метаданные приложения

В файле `config/logging.cli` - пакет команд `jboss-cli` для настройки логирования. В них прописаны параметры ротации логов.

В папке `certificate` находятся файлы ключей и хранилища сертификатов.

В папке `images` - официальный докер образ `wildfly:18.0.0.Final` и образ утилиты `keytool`

В папке `jboss` - файлы модулей WildFly необходимых приложению RMS

В папке `jdbc-driver` - драйвер `postgresql-42.2.8.jar`

Для запуска выполнить скрипт `wildfly_setup.py`

В ходе установки будет запрошен и установлен пароль администратора (для доступа веб консоли WildFly. Консоль доступна на порту 9990)

Так же установщик попросит указать сертификаты, ввести пароли существующих и создаваемых хранилищ ключей.

В папке `certificate/server/` уже лежат тестовые сертификаты. При необходимости заменить на актуальные.

1. SSL сертификат агента: `client.crt`
2. SSL сертификат сервера: `dev-orch.robin.it.ru.pfx` (пароль источника - 1)
3. обязательный сертификат верификации RMC: `orchestrator-client.p12` (пароль - Qwerty123, alias - orchestrator-client). Alias и пароль должны быть прописаны в `application.properties` приложения RMS.

Настройку SSL можно отключить в конфигурационном файле `config/wildfly_setup.config`

## 2.4 Развертывание приложения RMS на сервер WildFly (4\_deploy\_rms)

В файле `config/deploy_rms.config` указываются параметры подключения к серверу WildFly

В папке `properties` - файлы свойств приложения RMS.

В папке `target` - war-файл приложения RMS

В папке `wildfly_updates` - файлы `jboss-cli` скрипты обновления конфигурации WildFly (соответствующие версиям RMS)

Для запуска выполнить скрипт `deploy_rms.py`

RMS подключается к RMS исключительно по защищенному протоколу https. Если SSL не настраивается на сервере WildFly, то необходимо настроить SSL прокси. Для этих целей можно использовать веб-сервер Nginx.

Ниже пример файла конфигурации Nginx `nginx.conf`. Необходимо указать ip сервера WildFly и путь до сертификатов.

`/etc/nginx/conf.d/nginx.conf`

```
server {
    listen 80;
    server_name 172.28.1.57;
    return 301 https://172.28.1.57$request_uri;
}

server {
    listen 443 ssl;
    server_name 172.28.1.57;

    # location of key and certificate files
    ssl_certificate /etc/nginx/robin.crt;
    ssl_certificate_key /etc/nginx/robin.rsa;
    ssl_protocols SSLv2 SSLv3 TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;

    # cache ssl sessions
    #ssl_session_cache builtin:1000 shared:SSL:10m;

    # prefer server ciphers (safer)
    ssl_prefer_server_ciphers on;

    # important! protects your website against MITM attacks.
    add_header Strict-Transport-Security "max-age=604800";

    location / {
        # these headers contains information about original request
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $http_host;
        proxy_set_header Upgrade $http_upgrade;
```

```
proxy_set_header Connection "Upgrade";  
proxy_http_version 1.1;  
proxy_pass http://172.28.1.57:8080;  
}  
}
```

## 4.1 Установка

Запустить инсталлятор

```
RdpManagerSetup.exe
```

Выбрать язык установки, путь к папке и нажать кнопку **Установить**.

Открыть файл:

```
RobinRdpManager\config\application
```

Заменить в файле `%hostname%` на адрес сервера Orchestrator.

Открыть службы Windows и запустить

```
Robin RdpManagerService
```

## 4.2 Логирование

Для просмотра логов RDP Manager открыть папку

```
C:\Windows\System32\config\systemprofile\AppData\Local\RobinRdpManager\logs
```