

## **INSPARK Data Center**

### **ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ (серверная часть)**

Листов 15

Москва 2024г.

## ОГЛАВЛЕНИЕ

<b>1 Введение .....</b>	<b>3</b>
<b>2 Назначение и условия применения.....</b>	<b>4</b>
2.1 Назначение системы INSPARK DATA CENTER .....	4
2.2 Условия применения.....	4
2.3 Требования к совместимости серверного ПО.....	4
2.3.1 Мощность среды.....	4
2.3.2 Рекомендации по аппаратным ресурсам .....	5
2.3.3 Рекомендации по программной совместимости.....	6
<b>3 Описание операций .....</b>	<b>7</b>
3.1 Подготовка к установке серверных компонент.....	7
3.1.1 Подготовка ОС .....	7
3.1.2 Установка HTTP.....	7
3.1.3 Установка J2EE сервера WildFly .....	8
3.1.4 Установка СУБД.....	10
3.2 Настройка серверов INSPARK DATA CENTER .....	10
3.2.1 Развертывание БД.....	10
3.2.2 Создание кластера PostgreSQL .....	11
3.2.3 Создание базы данных .....	12
3.2.4 Настройка конфигурационных данных.....	12
3.2.5 Настройка источников данных.....	12
3.2.6 Настройка связи с базой данных.....	13
3.2.7 Настройка JMS.....	14
3.2.8 Настройка кодировки .....	15
3.3 Установка серверных приложений.....	15

# 1 ВВЕДЕНИЕ

В настоящем документе описаны действия администратора системы INSPARK Data Center по установке и настройке серверных компонент системы.

INSPARK Data Center предоставляется в виде публичной (общедоступной) облачной услуги по модели подписки, либо в виде установленного экземпляра ПО на сайте Пользователя.

Материал ниже предназначен для выполнения работ по установке и настройке INSPARK Data Center только для установки ПО на сайте Пользователя.

## 2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМИНЕНИЯ

### 2.1 Назначение системы INSPARK Data Center

INSPARK Data Center является программной платформой для сбора, агрегации, обработки и представления информации от различных источников данных, основанных на принципах и способах IoT технологии.

INSPARK Data Center предоставляет совокупность программных решений и технологий, которые позволяют осуществлять автоматизацию и диспетчеризацию центров обработки данных (ЦОД) всех уровней, а также объекты критической информационной инфраструктуры (КИИ).

INSPARK Data Center предоставляет три базовые возможности IoT платформы:

- управление подключением (connectivity management) систем, датчиков и устройств;
- агрегацию и хранение данных (IoT core);
- поддержку приложений (IoT analytics and application).

INSPARK Data Center содержит:

- серверная платформа;
- web приложения;
- ПО для контроллеров.

### 2.2 Условия применения

Обязательным условием администрирования INSPARK Data Center является навыки администрирования Linux, умение пользоваться командной строкой, текстовыми редакторами, управлением файловой системой, сетевыми настройками, создание и управлениями пользователями, правами доступа к файлам и т.д.

Для сетевого взаимодействия на межсетевых экранах должны быть открыты следующие порты в соответствующих направлениях:

Исходящий адрес	Входящий адрес:порт	пояснение
IP сервера приложений	IP сервера БД: 5432	Взаимодействие сервера приложения с СУБД
IP администратора	IP сервера приложения: 80,443	Взаимодействие администратора с WEB интерфейсом системы

### 2.3 Требования к совместимости серверного ПО

В разделе даны рекомендации по применению аппаратных и программных сред для работы серверных компонент платформы INSPARK Data Center.

#### 2.3.1 Мощность среды

Требования к аппаратным средствам представим с учетом мощности среды.

Среда	Количество параметров конфиг.	Количество сессий пользователей
Proof of Concept (POC)	3000	10
Small	200 000	200
Medium	1 000 000	400
Large	3 000 000	1000

Рост мощности среды будет зависеть от той нагрузки, которую должны выдерживать сервера INSPARK Data Center. В таблице приведены рекомендуемое деление, которое позволяет провести начальную оценку потребности в аппаратных средствах.

Среда POC используется для тестовых исследований и сред разработки. Среда Small закрывает потребности, например, небольшой организации до 200 пользователей. Среды Medium и Large развертываются для действительно большого числа пользователей (больше 5 000).

### 2.3.2 Рекомендации по аппаратным ресурсам

**ВНИМАНИЕ !** Сведения, приведенные ниже, носят рекомендательный характер, и могут быть применены только в рамках оценки потребностей.

Компонент	POC	Small	Medium	Large
Сервер web приложений	Не требуется: устанавливается вместе с сервером приложений	1 сервер: 4 CPU CORE 8 Gb RAM 5 Gb disk	2 сервера: CPU core 12 Gb RAM 5 Gb disk	3 сервера: 8 CPU core 16 Gb RAM 5 Gb disk
Сервер приложений	1 сервер: 8 CPU core; 16 Gb RAM 50 Gb disk	1 сервер: 16 CPU core 32 Gb RAM 50 Gb disk	1 сервер: 32 CPU core 64 Gb RAM 50 Gb disk	2 сервера: 32 CPU core 96 Gb RAM 50 Gb disk
Сервер БД	Не требуется: устанавливается вместе с сервером приложений	1 сервер: 16 CPU core 64 Gb RAM 500 Gb disk	1 сервер: 32 CPU core 128 Gb RAM 1 Tb disk	1 сервера: 64 CPU core 128 Gb RAM 2 Tb disk

### 2.3.3 Рекомендации по программной совместимости

Компонент	POC	Small	Medium	Large
Уровень ОС	Ubuntu Server 16.x и выше, Windows Server 2016, Red Hat 6 и выше		Windows Server 2016, Red Hat 6 и выше	
Сервер приложений, web-сервер	J2EE WildFly 11, Apache Tomcat 9			
Сервер СУБД	PostGresSQL 11			

### 2.3.4 Совместимость с браузерами

Приложение поддерживает следующие браузеры:

Браузер	Поддерживаемая версия
Chrome для всех ОС	71.x и выше
FireFox для всех ОС	64.x и выше
MS Internet Explorer	11.x и выше
MS Edge	40.x и выше
Safari для всех ОС	11.x , 12.x
Android	Nougat (7.0) Marshmallow (6.0) Lollipop (5.0, 5.1) KitKat (4.4)

## 3 ОПИСАНИЕ ОПЕРАЦИЙ

### 3.1 Подготовка к установке серверных компонент

В зависимости от схемы серверов, на каждом ролевом сервере должны быть установлены обязательные общесистемные пакеты. Ниже приведены материалы для подготовки серверов INSPARK Data Center на свободно распространяемом ПО. В тоже время, в разделе совместимости приведен полный перечень поддерживаемого ПО. В случае необходимости материалов для развертывания платформы INSPARK Data Center на других средах, следует обратиться в поддержку.

#### 3.1.1 Подготовка ОС

##### 3.1.1.1 ОС Ubuntu

Проверить версию ОС: `lsb_release -a`

Версия ОС должна быть не ниже 16.0

Выполнить обновление индекса пакетов: `apt-get update`

Выполнить установку последних пакетов ОС: `apt-get upgrade`

Установить JDK 8

##### 3.1.1.2 Установка JAVA (JDK 8)

Рекомендуем использовать официальную версию Java Oracle JDK, распространяемую компанией Oracle (на момент написания статьи рекомендуем использовать 8 версию)

Добавить PPA компании Oracle: `add-apt-repository ppa:webupd8team/java`

Обновить список пакетов: `apt-get update`

Установить Oracle JDK 8: `apt-get install oracle-java8-installer`

Проверить версию: `java -version`

#### 3.1.2 Установка HTTP

Рекомендуем использовать один из поддерживаемых веб-серверов: `nginx`, `apache`. Гибридная конфигурация `Nginx+ Apache` также возможна.

##### 3.1.2.1 Установка веб-сервера Nginx

Инсталляция `nginx`:

```
apt-get install nginx
```

Настройка профиля `Nginx`:

```
ufw app list
ufw allow 'Nginx Full'
```

Запуск службы `nginx`:

```
service nginx start
```

Проверка версии:

```
nginx -v
```

### 3.1.2.2 Установка веб-сервера Apache

Инсталляция apache2:

```
apt-get install apache2
```

Настройка профиля:

```
ufw app list  
ufw allow 'Apache Full'
```

Запуск службы:

```
service apache2 start
```

Проверка версии:

```
apache2 -v
```

### 3.1.3 Установка J2EE сервера WildFly

*{На примере wildfly-10.1.0.Final}*

Скачать дистрибутив с официального сайта:

```
http://www.wildfly.org/downloads/
```

Перейти в каталог загрузки и извлечь архив в /opt/wildfly:

```
tar -xvzf wildfly-10.1.0.Final.tar.gz -C /opt/wildfly
```

Назначить права на каталог:

```
chmod -R 755 wildfly
```

Создать пользователя ОС, под которым он будет работать:

```
adduser --no-create-home --disabled-password --disabled-login wildfly  
chown -R wildfly.wildfly /opt/wildfly
```

Так как Wildfly распространяется в виде rpm - у него есть скрипты демонизации:

```
/opt/wildfly/docs/contrib/scripts/init.d
```

Скопировать и активировать init-скрипт:

```
cp /opt/wildfly/docs/contrib/scripts/init.d/wildfly-init-debian.sh /etc  
/init.d/wildfly  
update-rc.d wildfly defaults
```

Скопировать конфигурационный файл в `/etc/default/wildfly`

```
cp /opt/wildfly/docs/contrib/scripts/init.d/wildfly.conf /etc/default/wildfly
```

Редактировать конфигурационный файл:

```
vi /etc/default/wildfly
```

Внести следующие изменения:

```
JBOSS_HOME="/opt/wildfly"  
JBOSS_USER=wildfly  
JBOSS_MODE=standalone  
JBOSS_CONFIG=standalone-full.xml  
STARTUP_WAIT=300  
SHUTDOWN_WAIT=300  
JBOSS_CONSOLE_LOG="/var/log/wildfly/console.log"
```

Создать директорию с консольными логами:

```
mkdir /var/log/wildfly/  
chown wildfly:wildfly /var/log/wildfly/
```

Запустить wildfly в качестве службы:

```
service wildfly start
```

Проверить:

```
wget http://localhost:8080
```

По умолчанию Wildfly слушает только 127.0.0.1. Чтобы слушал на всех интерфейсах нужно во всех местах конфига заменить 127.0.0.1 на 0.0.0.0:

```
service wildfly stop  
vi /opt/wildfly/standalone/configuration/standalone-full.xml  
    :%s/127.0.0.1/0.0.0.0/g  
    :wq  
service wildfly start
```

Теперь доступен на внешнем IP и порту 8080

Добавить память и включить нужные опции JVM в конфигурационном файле:

```
/opt/wildfly/bin/standalone.conf  
JAVA_OPTS="-Xms1024m -Xmx4096m -XX:MetaspaceSize=512M -XX:MaxMetaspaceSize=1024m -Djava.net.preferIPv4Stack=true"
```

```
JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
```

Перезагрузить wildfly:

```
service wildfly start
```

### 3.1.4 Установка СУБД

Платформа СУБД для оперативного хранилища использует реляционную СУБД. Платформа не предъявляет ограничений к типу СУБД. В текущей версии платформа использует СУБД PostgreSQL.

Создать файл:

```
touch /etc/apt/sources.list.d/PostgreSQL.list
```

Добавить строку для репозитория:

```
deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -sc)-pgdg main
```

используя, к примеру команду echo:

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -sc)-pgdg main" > /etc/apt/sources.list.d/PostgreSQL.list'
```

Импортировать ключ репозитория:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

Обновить список пакетов:

```
apt-get update
```

Установить postgresql:

```
apt-get install postgresql-11
```

Проверить версию:

```
psql --version
```

Рекомендуется версия PostgreSQL не ниже 11.

## 3.2 Настройка серверов INSPARK DATA CENTER

### 3.2.1 Развертывание БД

Создать каталог БД:

```
mkdir /pgstore
```

Определить UUID разделов:

```
blkid
```

в файле:

```
/etc/fstab
```

добавить необходимый раздел:

```
UUID=4b2a649f-9668-4637-9d2c-c1d5eaf4fad1 /pgstore          ext4
errors=remount-ro      1
```

выполнить:

```
mount -a
```

Создать каталог /pgstore/11:

```
mkdir /pgstore/11
```

### 3.2.2 Создание кластера PostgreSQL

Создать кластер PostgreSQL в локали ru\_RU.UTF-8 Если в системе локали нету, то:

```
locale-gen ru_RU.UTF-8
```

При этом локаль должна быть установлена:

```
dpkg-reconfigure locales
```

Включить локаль в консоли:

```
export LC_ALL=ru_RU.UTF-8
```

Перейти под пользователя postgres:

```
sudo su - postgres
```

Удалить старый кластер базы данных:

```
pg_dropcluster --stop 11(версия кластера) main(имя кластера, получить м  
ожно pg_lsclusters)
```

Остановить процесс postgresql и выйти из-под postgres:

```
sudo systemctl stop postgresql@11-main
logout
```

Убедиться, что процесс остановлен:

```
systemctl stop postgresql@11-main
```

```
systemctl daemon-reload
```

изменить владельца каталога:

```
chown -R postgres:postgres /pgstore
```

Перейти под пользователя postgres:

```
sudo su - postgres
```

Создать новый кластер:

```
pg_createcluster -d /pgstore/11 --locale ru_RU.UTF-8 --start 11 main
```

Выйти из-под postgres и запустить службу СУБД:

```
logout
sudo systemctl daemon-reload
sudo systemctl start postgresql@11-main
```

В конфигурационном файле СУБД:

```
/etc/postgresql/11/main/pg_hba.conf
```

открыть внешние соединения с БД:

```
host all all 0.0.0.0/0 md5
```

### 3.2.3 Создание базы данных

Пользователем postgres в psql выполнить:

```
CREATE DATABASE SEMDB WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';
ALTER DATABASE SEMDB OWNER TO postgres;
```

### 3.2.4 Настройка конфигурационных данных

В конфигурационном файле **postgresql.conf** для

параметра *max\_prepared\_transactions* установить значение равным 200, значение по умолчанию для ПО (SEMNEXT) INSPARK Data Center

```
max_prepared_transactions = 200          # zero disables the feature
```

### 3.2.5 Настройка источников данных

Создать директорию `/opt/wildfly/modules/org/postgres/main` и скопировать в нее JDBC-драйвер `postgresql-9.4.1212.jre6.jar` и `module.xml`

Перезапустить сервер приложений

```
service wildfly restart
```

Зайти в административную консоль

```
http://ip_сервера_приложений:9990/
```

перейти в *Configuration*→*Subsystem*→*Datasources*→*Non XA* нажать *Add* выбираем «**PostgreSQL Datasource**», задать:

```
Name: DefaultDataSource
JNDI name: java:/jdbc/SEM
```

перейти на вкладку *Detected Drivers* и выбираем JDBC-драйвер **postgresql-9.4.1212.jre6.jar**, настроить параметры в конфигурационном файле **/opt/wildfly/standalone/configuration/standalone-full.xml**. Все ниже описанные настройки производятся в секциях конфигурационного файла **standalone-full.xml**

### 3.2.6 Настройка связи с базой данных

В конфигурационном файле **/opt/wildfly/standalone/configuration/standalone-full.xml** Найти секцию `<subsystem xmlns=«urn:jboss:domain:datasources:4.0»>`. Настроить параметры для связи с базой данных:

```
<subsystem xmlns="urn:jboss:domain:datasources:4.0">
  <datasources>
    <xa-datasource jndi-name="java:/jdbc/SEM" pool-name="DefaultDataSource" enabled="true" use-ccm="true">
      <xa-datasource-property name="ServerName">
        ИМЯ_СЕРВЕРА_БД
      </xa-datasource-property>
      <xa-datasource-property name="PortNumber">
        ПОРТ_СЕРВЕРА_БД_ПО_УМОЛЧАНИЮ_5433
      </xa-datasource-property>
      <xa-datasource-property name="DatabaseName">
        ИМЯ_БД_РЕКОМЕНДУЕМ_SEMDB//
      </xa-datasource-property>
      <xa-datasource-property name="currentSchema">
        ИМЯ_СХЕМЫ_РЕКОМЕНДУЕМ_sem
      </xa-datasource-property>
      <driver>postgres</driver>
    </xa-datasource>
  </datasources>
</subsystem>
```

```

        <min-pool-size>10</min-pool-size>
        <max-pool-size>20</max-pool-size>
        <prefill>true</prefill>
    </xa-pool>
    <security>
        <user-name>ИМЯ_ПОЛЬЗОВАТЕЛЯ_БД_SEM_NEXT_ПО_УМОЛ
ЧАНИЮ_postgres</user-name>
        <password>ПАРОЛЬ_ПОЛЬЗОВАТЕЛЯ_БД_SEM_NEXT</pass
word>
    </security>
    <validation>
        <valid-connection-checker class-name="org.jboss
.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker
"/>
        <background-validation>true</background-validat
ion>
        <exception-sorter class-name="org.jboss.jca.ada
pters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"/>
    </validation>
</xa-datasource>
<drivers>
    <driver name="postgres" module="org.postgres">
        <xa-datasource-class>org.postgresql.xa.PGXADat
aSource</xa-datasource-class>
    </driver>
</drivers>
</datasources>
</subsystem>

```

### 3.2.7 Настройка JMS

Переходим в секцию `<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">` И добавляем очереди:

```

<jms-queue name="ExpiryQueue" entries="java:/jms/queue/ExpiryQueue"/>
<jms-queue name="DLQ" entries="java:/jms/queue/DLQ"/>
<jms-queue name="Measure" entries="java:/jndi/Measure"/>
<jms-queue name="Event" entries="java:/jndi/Event"/>
<jms-queue name="AdminLog" entries="java:/jndi/AdminLog"/>
<jms-queue name="Command" entries="java:/jndi/Command"/>
<jms-queue name="CalcParam" entries="java:/jndi/CalcParam"/>

```

```
<jms-queue name="Collector" entries="java:/jndi/Collector"/>
<jms-queue name="ParamControl" entries="java:/jndi/ParamControl"/>
<jms-queue name="Rule" entries="java:/jndi/Rule"/>
<jms-queue name="RuleExec" entries="java:/jndi/RuleExec"/>
<jms-queue name="SetValue" entries="java:/jndi/SetValue"/>
<jms-queue name="Notify" entries="java:/jndi/Notify"/>
<jms-queue name="EventNotify" entries="java:/jndi/EventNotify"/>
<jms-queue name="EventImport" entries="java:/jndi/EventImport"/>
<jms-topic name="EventExport" entries="java:/jndi/EventExport"/>
<jms-topic name="ParamChange" entries="java:/jndi/ParamChange"/>
```

Так же очереди можно добавить через административную консоль.

### 3.2.8 Настройка кодировки

Переходим в секцию `<subsystem xmlns=«urn:jboss:domain:undertow:3.1»>` находим `<servlet-container name=«default»>` и меняем на:

```
<servlet-container name="default" default-encoding="UTF-8" use-listener-encoding="true">
```

## 3.3 Установка серверных приложений

Установить приложения на сервер J2EE. Скопировать приложения **semona.ear**, **datacenter.war** в директорию развертывания `opt/wildfly/standalone/deployments`.

Перезапустить сервер приложений:

```
service wildfly restart
```

Проверить, что выполняется команда:

```
curl -X POST "http://localhost:8080/sem-restservices/auth/login?username=root&password=masterkey"
```

Проверить через браузер доступность основного ресурса:

[http://ip\\_сервера\\_приложений:8080/datacenter/](http://ip_сервера_приложений:8080/datacenter/)