



ОБЩЕСТВО  
С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«БЮДЖЕТНЫЕ И ФИНАНСОВЫЕ ТЕХНОЛОГИИ»

129085, г. Москва,  
ул. Годовикова, д. 9, стр. 17.  
+7 (495) 784-70-00  
ineed@bftcom.com, www.bftcom.com

## **Инструкция по установке экземпляра программного обеспечения и запуску системы «БФТ. Управление активами»**

На 25 листах

2024

## Оглавление

Оглавление .....	2
<b>1. УСТАНОВКА СИСТЕМЫ «БФТ. УПРАВЛЕНИЕ АКТИВАМИ».....</b>	<b>3</b>
<b>1.1 Установка стандартного ПО.....</b>	<b>3</b>
<b>1.1.1 Установка операционной системы .....</b>	<b>3</b>
<b>1.1.3 Установка OpenJRE.....</b>	<b>4</b>
<b>1.1.4 Установка сервера приложений Tomcat 9 .....</b>	<b>5</b>
<b>1.1.5 Установка PostgreSQL 12 .....</b>	<b>11</b>
<b>1.1.6 Настройка Tomcat .....</b>	<b>12</b>
<b>1.1.7 Установка PostGIS .....</b>	<b>16</b>
<b>1.1.8 Создание базы приложения .....</b>	<b>17</b>
<b>1.2 Установка приложения БФТ. Управление активами.....</b>	<b>18</b>
<b>1.2.1 Установка приложения БФТ. Управление активами ручным способом .....</b>	<b>18</b>
<b>1.2.2 Установка приложения БФТ. Управление активами посредством Ansible-скриптов .....</b>	<b>20</b>
<b>1.3 Установка приложения БФТ. Управление активами.....</b>	<b>25</b>

# 1. УСТАНОВКА СИСТЕМЫ «БФТ. УПРАВЛЕНИЕ АКТИВАМИ»

## 1.1 Установка стандартного ПО

### 1.1.1 Установка операционной системы

Подробно процесс установки описан в документации разработчика ОС, которую можно скачать со страницы:

Astra

Linux

2.12

—

<https://wiki.astralinux.ru/pages/viewpage.action?pageId=37290417>;

RedOS 7.3 – <https://redos.red-soft.ru/product/red-os/>;

ОС Альт 8 СП – <https://altsp.su/polzovatelyam/dokumentatsiya/>.

### 1.1.2 Первоначальная подготовка операционной системы

Установить ssh-клиент PuTTY на рабочей станции, с которой планируется производить подключение к удаленному серверу. Скачать клиент по ссылке: <https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.76-installer.msi>.

Открыть клиент PuTTY, в навигационной области панели «Category» выбрать раздел «Session». В поле «Host Name (or Ip address)» ввести IP-адрес сервера или его доменное имя, нажать «Open» (Рисунок 1).

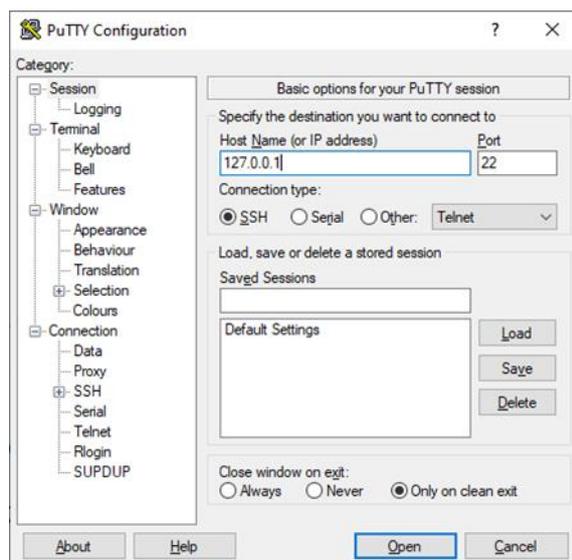


Рисунок 1 – Настройка подключения через ssh-клиент PuTTY

В окне консоли ввести логин и пароль от УЗ пользователя (пользователь должен обладать правами sudo). В командной строке (по

очереди) выполнить следующий набор команд:

1) Обновить текущий репозиторий, выполнив команду:

```
sudo yum update
```

По итогу установки подтвердить действие, нажать «Y».

2) Установить «wget» , выполнив команду:

```
sudo yum install wget
```

По итогу установки подтвердить действие, нажать «Y».

3) Установить Midnight commander , выполнив команду:

```
sudo yum install mc
```

По итогу установки подтвердить действие, нажать «Y».

4) Установить «unzip», выполнив команду:

```
sudo yum install unzip
```

По итогу установки подтвердить действие, нажать «Y».

При установке ОС во время настройки некоторых параметров необходимо отключить Firewall и службу SELinux.

### **1.1.3 Установка OpenJRE**

1 . Перейти в директорию /usr/lib

```
cd /usr/lib
```

2 . Создать там каталог jvm

```
sudo mkdir jvm
```

3 . Из предоставленного набора дистрибутивов скопировать архив `jdk-11.0.13_linux-x64_bin.tar.gz` в директорию /usr/lib/jvm

```
sudo cp jdk-11.0.13_linux-x64_bin.tar.gz /usr/lib/jvm/
```

4 . Перейти в каталог /usr/lib/jvm и распаковать архив

```
cd /usr/lib/jvm
```

```
sudo tar -xvf jdk-11.0.13_linux-x64_bin.tar.gz
```

5 . Убедиться, что распаковка прошла успешно и удалить архив

```
sudo rm jdk-11.0.13_linux-x64_bin.tar.gz
```

6 . Далее необходимо установить переменную `$JAVA_HOME` и добавить путь к java в переменную `$PATH`. Для этого следует открыть файл `/etc/profile` в любом редакторе, например

```
sudo gedit /etc/profile
```

и прописать их ниже:

```
export JAVA_HOME=/usr/lib/jvm/jdk-11.0.13
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

7 . Далее следует перезагрузить систему и проверить, что java 11 установлена

```
java --version
```

должна быть выведена версия java:

```
java 11.0.13 2021-10-19 LTS
```

### 1.1.4 Установка сервера приложений Tomcat 9

Для установки «Tomcat» на сервере приложения «БФТ. Управление активами» необходимо:

Для обозначения каталога установки любой копии Tomcat будет использоваться <ТМС>

– Создать каталог

```
sudo mkdir /opt/_Tomcat
```

– Создать группу tomcat

```
sudo groupadd tomcat
```

– Создать пользователя **tomcat**

```
sudo useradd -M -s /sbin/nologin -g tomcat -d /opt/_Tomcat  
tomcat
```

– В каталоге /opt/\_Tomcat создать каталог application1-8080 (указание порта в названии каталога, по которому будет работать приложение предусмотрено специально — это поможет при отладке и в дальнейшей работе с сервером)

```
sudo mkdir /opt/_Tomcat/application1-8080
```

– Из предоставленного набора дистрибутивов скопировать и распаковать архив apache-tomcat-9.0.36.tar.gz в каталог /opt/\_Tomcat/application1-8080:

```
sudo tar xvf apache-tomcat-9.0.36.tar.gz -C /opt/_Tomcat/application1-8080 --strip-components=1
```

– Установить права доступа:

```
sudo chown -R tomcat:tomcat /opt/_Tomcat
sudo find /opt/_Tomcat/* -type d -exec chmod 770 {} \;
sudo find /opt/_Tomcat/* -type f -name *.sh -exec chmod 770 {} \;
```

– Создать каталоги для журналов

```
sudo mkdir -p /var/log/tomcat/application1-8080/archiv
sudo rmdir /opt/_Tomcat/application1-8080/logs
sudo ln -s /var/log/tomcat/logs
sudo ln -s /var/log/tomcat/application1-8080 /opt/_Tomcat/application1-8080/logs
sudo chmod -R 770 /var/log/tomcat
sudo chown -R tomcat:tomcat /var/log/tomcat
sudo restorecon -Rv /var/log/tomcat
```

– Создать скрипт для архивации журналов (архивирует все журналы за определённую дату и сохраняет архивы в каталоге <ТМС>/logs/archiv в течение недели):

```
sudo mkdir -p /root/_Scripts/
sudo touch /root/_Scripts/arch_log_date-tomcat.sh
```

и привести его к виду:

```
#!/bin/bash

mkdir -p $1/archiv
```

```
find $1/*$(date --date '-1 day' +%Y-%m-%d)*.* -exec tar -
r -f $1/archiv/$(date --date '-1 day' +%Y%m%d).tar --remove-files
{ } \; && find $1/archiv/*.tar -exec gzip { } \; && find $1/archiv/*.
tar.gz -mtime +5 -delete
```

```
chown $2:$3 $1/archiv/*.tar.gz
```

```
chmod 640 $1/archiv/*.tar.gz
```

– Для logrotate создать файл-конфигурацию ротации основного файла журнала Tomcat-a (при активной работе с приложениями в Tomcat он увеличивается очень быстро) /etc/logrotate.d/tomcat следующего содержания:

```
/var/log/tomcat/application1-8080/catalina.out
```

```
{
```

```
rotate 10
```

```
size 200M
```

```
compress
```

```
notifempty
```

```
missingok
```

```
copytruncate
```

```
su tomcat tomcat
```

```
}
```

– В файл конфигурации "планировщика" /etc/crontab добавить строки:

```
0 */12 * * * root logrotate --force /etc/logrotate.d/tomcat
> /dev/null 2>&1
```

```
0 1 * * * root /root/_Scripts/arch_log_date-tomcat.sh
/var/log/tomcat/application1-8080 tomcat tomcat > /dev/null
2>&1
```

По умолчанию в конфигурационных файлах Tomcat (Tomcat/opt/\_Tomcat/application1-8080/conf/server.xml) в качестве основного порта доступа указан порт 8080.

– При необходимости настройки работы по HTTPS необходимо добавить дополнительные настройки:

```
<Connector      port="8443"      SSLEnabled="true"
maxHttpHeaderSize="8192"
      maxThreads="150"          minSpareThreads="25"
maxSpareThreads="200"
      enableLookups="false" disableUploadTimeout="true"
      acceptCount="100" scheme="https" secure="true"
      clientAuth="false" sslProtocol="TLS"
      keystoreFile="/opt/_Tomcat/application1-
8080/conf/ssl/keystore.p12"      keystorePass="*****"
keystoreType="PKCS12"
      keyAlias="application1"/>
```

И создать сами ключи:

```
sudo mkdir -p /opt/_Tomcat/application1-8080/conf/ssl
sudo keytool -genkeypair -keyalg RSA -keysize 2048 -
storetype PKCS12 -keystore keystore.p12 -validity 3650 -alias
application1      -file      /opt/_Tomcat/application1-
8080/conf/ssl/keystore.p12
```

– В интерактивном режиме указать требуемые параметры и получить по окончании файл-хранилище с ключами (закрытым (секретным) и публичным). В файл <ТМС>/conf/context.xml перед закрывающим тегом </Context> добавить строку:

...

```

        <Resources cachingAllowed="true"
cacheMaxSize="100000" cacheTtl="2000" />
    </Context>

```

– Создать файлы systemd-юнитов для запуска Tomcat-ов в качестве сервиса. Для Application1 /etc/systemd/system/tomcat-application1-8080.service:

```
# Systemd unit file for Tomcat - Application1
```

```
[Unit]
```

```
Description=Apache Tomcat Web Application Container
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=forking
```

```
###-- Каталог установки JDK
```

```
### OpenJDK - JRE
```

```
# Environment='JAVA_HOME=/usr/lib/jvm/jre'
```

```
Environment='JAVA_HOME=/usr/lib/jvm/jdk-11.0.13'
```

```
### OpenJDK - JRE
```

```
# Environment='JAVA_HOME=/usr/java/latest/jre'
```

```
Environment='CATALINA_PID=/opt/_Tomcat/application1-8080/temp/tomcat.pid'
```

```
Environment='CATALINA_HOME=/opt/_Tomcat/application1-8080'
```

```
Environment='CATALINA_BASE=/opt/_Tomcat/application1-8080'
```

```
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
```

```
Environment='JAVA_OPTS=-Djava.awt.headless=true -  
Djava.security.egd=file:/dev/./urandom'
```

```
WorkingDirectory=/opt/_Tomcat/application1-8080  
ExecStart=/opt/_Tomcat/application1-8080/bin/startup.sh  
ExecStop=/bin/kill -15 $MAINPID
```

```
User=tomcat  
Group=tomcat  
UMask=0007  
RestartSec=10  
Restart=always
```

[Install]

```
WantedBy=multi-user.target
```

- Указать systemd, чтобы он прочитал новые юниты:  
sudo systemctl daemon-reload
- Запустить, проверить, поставить в автозапуск:  
sudo systemctl enable tomcat-application1-8080.service  
sudo systemctl start tomcat-application1-8080.service  
sudo systemctl status tomcat-application1-8080.service
- Если запуск Tomcat прошёл успешно, можно разворачивать приложение. При необходимости передавать какие-либо параметры в приложение, их необходимо предварительно прописывать в файл <ТМС>/conf/catalina.properties.
- Для установки приложения необходимо выбрать файл app.war из предоставленного набора дистрибутивов и поместить его в папку /opt/\_Tomcat/application1-8080/webapps

### 1.1.5 Установка PostgreSQL 12

– Для установки PostgreSQL 12 из исходников потребуются следующие пакеты, доступные из ОС: gcc make, libreadline-dev, zlibc zlib1g-dev — для их установки следует воспользоваться командой apt install

```
apt install gcc make
```

```
apt install libreadline-dev
```

```
apt install zlibc zlib1g-dev
```

– Создать на жестком диске папку, скопировать туда и распаковать архив postgresql-12.6.tar.gz из набора предоставленных дистрибутивов

```
mkdir pg
```

```
cp postgresql-12.6.tar.gz ~/pg
```

```
cd ~/pg
```

```
tar xvf postgresql-12.6.tar.gz
```

– Для установки СУБД последовательно выполнить:

```
sudo ./configure
```

```
sudo make
```

```
sudo make install
```

– Создать каталог для хранения данных

```
sudo mkdir /usr/local/pgsql/data
```

– Создать пользователя postgres. Во время создания система попросит назначить пароль и прочие пользовательские данные

```
sudo adduser postgres
```

– Сделать его владельцем каталога data и установить права

```
chown postgres /usr/local/pgsql/data/
```

```
chmod 750 /usr/local/pgsql/data/
```

– Добавить в /etc/profile переменную PGDATA и путь к PostgreSQL

```
export PGDATA=/usr/local/pgsql/data
```

```
export PATH=$PATH:/usr/local/pgsql/bin
```

– Зайти под пользователем postgres

```
su – postgres
```

– Проинициализировать кластер баз данных

```
initdb -k
```

– Изменить аутентификацию по паролю:

```
sudo gedit /usr/local/pgsql/data/pg_hba.conf
```

В блоке:

```
host    all    all    127.0.0.1/32    ident
host    all    all    ::1/128         ident
```

Заменить данные на:

```
host    all    all    127.0.0.1/32    md5
host    all    all    ::1/128         md5
```

### 1.1.6 Настройка Tomcat

1. Для всех компонентов в указанный файл <ТМС>/conf/catalina.properties надо добавить параметр. <ТМС> - это домашний каталог конкретного экземпляра установленной копии Tomcat (в данном примере это /opt/\_Tomcat/application1-8080).

Добавить параметр:

```
ice.projectRoot=/opt/_Tomcat/application1-8080
```

2. Создать соответствующий каталог:

```
sudo mkdir -p <ТМС>/ice
```

```
sudo chown tomcat:tomcat <ТМС>/ice
```

```
sudo chmod 750 <ТМС>/.ice
```

3. Так же создать файл **/spring-shell.log** (его необходимость зависит от параметров, указанных в **<ТМС>/conf/catalina.properties**):

```
sudo touch /spring-shell.log
```

```
sudo chown tomcat:tomcat /spring-shell.log
```

```
sudo chmod 750 /spring-shell.log
```

4. Открыть доступ к порту:

- на межсетевом экране firewalld:

```
# открываем доступ к контейнерам и интерфейсу управления для
указанной сети
```

```
# добавляем требуемые порты 8080
```

```
sudo firewall-cmd --permanent --new-service=tomcat-http
```

```
sudo firewall-cmd --permanent --service=tomcat-http --add-
port=8080/tcp
```

```
sudo firewall-cmd --permanent --add-service=tomcat-http
```

```
sudo firewall-cmd --reload
```

```
### Далее - примеры (!) для настройки более "тонкой" настройки
доступа:
```

```
sudo firewall-cmd --permanent --add-rich-rule 'rule family="ipv4"
source address="192.168.1.0/24" service name="http" accept'
```

```
# открываем полный доступ с сервера мониторинга Zabbix
```

```
sudo firewall-cmd --permanent --add-rich-rule 'rule family="ipv4"
source address="192.168.19.34/32" accept'
```

- в SELinux:

5. Проверить, что SELinux активен:

```
sudo sestatus
```

6. Если в ответе на эту команду в строке "SELinux status:" будет enabled - проверяем целевые порты и добавляем по необходимости:

```
# Посмотреть, открыты ли порты наших серверов Tomcat
sudo semanage port -l | grep -E "8080"
```

7. Так же проверяем, если ли блокировки со стороны SELinux:

```
sudo cat /var/log/audit/audit.log | grep denied | grep catalina.sh
```

8. Если в выводе будут строки, то создаём и применяем на основе их разрешающее правило:

```
sudo cat /var/log/audit/audit.log | grep denied | grep catalina.sh |
audit2allow -M my_tomcat
sudo semodule -i my_tomcat.pp
```

9. При случае, когда утилита semanage не установлена на сервере - установить её (и повторить затем предыдущие команды):

```
sudo yum install polycoreutils-python
```

10. Положить файл application.properties в папку с /opt/\_Tomcat/application1-8080 и внести в него следующие настройки:

```
ice.servicesPackages=com.bftcom.mdm.common,
com.bftcom.ice.common.service
```

```
spring.profiles.active=postgresql,postgresql95
```

```
spring.datasource.url=jdbc:postgresql://localhost:5433/saumi3
```

```
spring.datasource.username=admin
```

```
spring.datasource.password=admin
```

```
#spring.devtools.restart.trigger-file = application.properties
```

```
spring.devtools.restart.enabled = false
spring.devtools.livereload.enabled = false

server.servlet.contextPath=/app
# Charset of HTTP requests and responses. Added to the "Content-
Type" header if not set explicitly.
spring.http.encoding.charset=UTF-8
# Enable http encoding support.
spring.http.encoding.enabled=true
# Force the encoding to the configured charset on HTTP requests and
responses.
spring.http.encoding.force=true

spring.jackson.serialization.write-dates-as-timestamps=true
spring.jackson.default-property-inclusion = non_null

spring.servlet.multipart.maxFileSize=100MB
spring.servlet.multipart.maxRequestSize=100MB

spring.cache.jcache.provider=org.ehcache.jsr107.EhcacheCachingPr
vider
spring.cache.jcache.config=classpath:ehcache.xml

security.jwt.secret=random_secret_key
security.jwt.expirationTime=604800000

spring.quartz.job-store-type=jdbc
spring.quartz.jdbc.initialize-schema=never
spring.quartz.properties.org.quartz.jobStore.tablePrefix=quartz.qrtz_
```

```
spring.quartz.properties.org.quartz.jobStore.driverDelegateClass =  
org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

```
mdm.jms.url = tcp://localhost:61616  
mdm.jms.username = admin  
mdm.jms.password = admin  
mdm.jms.outbound.queue.name = mdmOutgoingQueue  
mdm.jms.inbound.queue.name = mdmIncomingQueue  
  
mdm.elasticsearch.url = http://localhost:9200
```

11. Прописать полный путь до этого файла в  
/opt/\_Tomcat/application1-8080/bin/startup.sh:

```
export CATALINA_OPTS="-  
Dspring.config.location=file:полный_путь_до_application.properties"
```

### 1.1.7 Установка PostGIS

1. Для установки PostGIS потребуются дополнительно установленные пакеты: libxml2-dev, libgeos-dev, libproj-dev, libgdal-dev, postgres-server-dev-12, доступные из ОС по команде `sudo apt install`

```
sudo apt install libxml2-dev  
sudo apt install libgeos-dev  
sudo apt install libproj-dev  
sudo apt install libgdal-dev  
sudo apt install postgres-server-dev-12
```

2. Из предоставленного набора дистрибутивов нужно скопировать на жесткий диск и распаковать архив `postgis-3.0.5.tar.gz`

```
tar -xvf postgis-3.0.5.tar.gz
```

3. Перейти в распакованный каталог

```
cd postgis-3.0.5
```

4. Последовательно выполнить команды для установки PostGIS:

```
./configure
```

```
make
```

```
make install
```

5. Далее необходимо установить дистрибутив postgresql-server-dev-12\_12.9-0ubuntu0.20.04.1.deb из предоставленного набора.

6. Кроме того необходимо настроить взаимодействие PostgreSQL и PostGIS для возможности использовать расширения при настройке баз данных. Из предоставленного набора необходимо установить на компьютер дистрибутив postgresql-12-postgis-3\_3.0.0+dfsg-6ubuntu4\_amd64.deb с помощью стандартного менеджера пакетов (дважды кликнув по файлу дистрибутива, откроется окно установки).

### 1.1.8 Создание базы приложения

1. Создать базу geo\_saumi3 и выполнить на ней:

```
CREATE EXTENSION postgis;
```

```
CREATE EXTENSION postgis_raster;
```

```
CREATE EXTENSION postgis_sfcgal;
```

```
CREATE EXTENSION address_standardizer;
```

```
CREATE EXTENSION fuzzystrmatch;
```

```
CREATE EXTENSION postgis_topology;
```

```
CREATE EXTENSION postgis_tiger_geocoder;
```

2. Создать базу ice\_esb

3. Создать базу данных saumi3 (в данном примере на ней будет развернуто приложение)

```
CREATE DATABASE saumi3;
```

4. Кроме того, для успешного восстановления БД из дампа, потребуется создать следующих пользователей и настроить одного из них

```
CREATE USER ice_user;  
CREATE USER admin;  
ALTER USER admin WITH SUPERUSER;  
ALTER DATABASE saumi3 OWNER TO admin;  
GRANT ALL PRIVILEGES ON DATABASE "saumi3" to admin;
```

5. Восстановить базу из дампа

```
pg_restore -d saumi3 db_backup_08_02_2022.tar.gz
```

## **1.2 Установка приложения БФТ. Управление активами**

### **1.2.1 Установка приложения БФТ. Управление активами ручным способом**

Для установки приложения БФТ.Управление активами следует выполнить следующие действия:

1) Скопировать веб-архив приложения app.war во временную папку на сервере приложение БФТ. Управление активами, например /usr/tmp.

2) Остановить Apache Tomcat.

3) Скопировать веб-архив приложения app.war в папку \$PATH\_TOMCAT/webapps.

4) Добавить глобальных настроек ./conf/catalina.properties Apache Tomcat следующие строки параметров:

```
DAS: app.war
```

```
-----
```

```
### DocArchive v1.8.0
```

```
jodconverter.local.enabled = true
```

```
jodconverter.local.port-numbers = 2003
```

```
jodconverter.local.office-home = /opt/libreoffice
```

```
jline.terminal = org.springframework.shell.core.IdeTerminal
```

*dm.fieldSetPackages = com.bftcom.docarch.commons*  
*ice.servicesPackages = com.bftcom.docarch.commons.service,*  
*com.bftcom.ice.common.service*  
*spring.profiles.active = postgresql*  
*spring.datasource.url = jdbc:postgresql://10.225.32.101:5432/db\_das\_uat*  
*spring.datasource.username = \*\*\*\*\**  
*spring.datasource.password = \*\*\*\*\**  
*spring.devtools.restart.enabled = false*  
*spring.devtools.livereload.enabled = false*  
*server.servlet.contextPath = /app*  
*spring.http.encoding.charset = UTF-8*  
*spring.http.encoding.enabled = true*  
*spring.http.encoding.force = true*  
*spring.jackson.serialization.write-dates-as-timestamps = true*  
*docarch.archiveStartYear = 2014*  
*docarch.file.encoding = UTF-8*  
*docarch.file.directory = files/exchange*  
*docarch.file.concurrency = 2*  
*docarch.file.pollInterval = 1000*  
*docarch.fsDirectory = files/attaches*  
*spring.quartz.job-store-type = jdbc*  
*spring.quartz.jdbc.initialize-schema = never*  
*spring.quartz.properties.org.quartz.jobStore.tablePrefix = quartz.qrtz\_*  
*spring.quartz.properties.org.quartz.jobStore.driverDelegateClass =*  
*org.quartz.impl.jdbcjobstore.PostgreSQLDelegate*  
*spring.shell.interactive.enabled = false*  
*docarch.eds.url = http://10.225.32.104:8090/eds*  
*security.ignore.urls =*  
*/attach,/attach/auth,/attach/data,/attach/info,/attach/download,/attach/check*

```
### CAS
sso.provider = cas
sso.useOnlySsoLogin = true
cas.host = crsso-test.dpc.fss.ru
# cas.port = 80
# cas.context-path = /cas
# cas.web-protocol = http://
# cas.login-form = login
# cas.logout-form = logout
cas.callback-url = http://10.225.32.102:8080/app/login/cas
# cas.logout-cas-url = /logout
# fss.user-roles-prefix = fss_ea_
# fss.user-role-super-user = fss_ea_super_user
```

-----  
*DAS API: api.war*  
-----

```
das.base-url = http://10.225.32.102:8080
das.login = *****
das.password = *****
```

-----

Запустить Apache Tomcat.

### **1.2.2 Установка приложения БФТ. Управление активами посредством Ansible-скриптов**

Установка БФТ. Управление активами посредством Ansible-скриптов производится с компьютера под управлением ОС Linux или ОС Windows с поддержкой WSL (Linux-среда внутри windows).

Для установки БФТ. Управление активами посредством Ansible-скриптов необходимо выполнить следующие шаги:

Подключиться к серверу, на который будет производиться установка,

по SSH.

1) Создать на сервере пользователя `deployer`, назначить ему пароль (для примера `Pass0rd`), выполнив команду:

```
useradd -m deployer -p Pass0rd
```

2) Добавить пользователя `deployer` в системную группу `wheel`, члены которой обладают возможностью повышения прав до суперпользователя, выполнив команду:

```
usermod -a -G wheel deployer
```

3) Создать ключ для доступа к серверу посредством команды `ssh-keygen`. При требовании ввести парольную фразу необходимо оставить поле пустым, так как Ansible не работает с ключами, защищенными парольной фразой.

4) Публичный ключ, полученный в п.4, добавить в файл, расположенный по пути: `/home/deployer/.ssh/authorized_keys`

5) Приватный ключ, полученный в п.4, сохранить на компьютер, с которого будет производиться установка.

6) Зайти на компьютер с ОС linux или ОС Windows с поддержкой WSL, с которого будет производиться установка, и создать директорию в linux-среде командой:

```
mkdir /var/das_deploy
```

7) Распаковать архив с Ansible-скриптами в директорию `/var/das_deploy` посредством команды:

```
tar -xvzf das.tgz
```

8) Приватный ключ, сохраненный в п.6, скопировать в директорию:

```
/var/das_deploy/DAS/keys/id_rsa
```

9) Дистрибутив приложения скопировать в директорию:

```
/var/das_deploy/DAS/roles/DAS/files/ под именем DAS.war
```

10) Установить на приватный ключ права 400 командой:

```
chmod 400 /var/das_deploy/DAS/keys/id_rsa
```

11) Установить необходимые Ansible модули для выполнения скриптов:

```
/usr/local/bin/ansible-galaxy collection install community.general
```

```
/usr/local/bin/ansible-galaxy collection install ansible.posix
```

```
/usr/local/bin/ansible-galaxy collection install community.docker
```

12) В файле Ansible-playbook

*/var/das\_deploy/DAS/playbooks/DAS.yml* внести необходимые изменения:

```
name: DAS DEV
```

```
hosts: bft_das_dev
```

```
remote_user: deployer
```

```
vars:
```

```
ansible_ssh_private_key_file: './keys/id-rsa'
```

```
roles:
```

```
- { role: './roles/DAS',
```

```
become: yes,
```

```
become_user: "root",
```

```
ansible_become_password: 'Pass0rd'}
```

13) В файле */var/das\_deploy/DAS/gt\_inventory/static\_inventory*

указать адрес сервера, на который будет произведена установка, в формате:

```
[bft_das_dev] //название группы серверов
```

```
172.24.17.108 //IP адрес сервера
```

14) В файле переменных для группы серверов

*/var/das\_deploy/DAS/gt\_inventory/group\_vars/bft\_das\_dev.yml* внести необходимые изменения в конфигурацию (ниже представлен пример):

```
java_home: "/usr/lib/jvm/java-11" – путь к домашней директории
```

```
java
```

```
server_addr: "111.11.11.111" – ip адрес сервера
```

```
domain_name: "DAS-tes.test" – доменное имя
```

```
DAS_app_name: "DAS-test.test" - имя приложения
```

```
DAS_project_port: "8080" – http порт Tomcat
```

```
DAS_srv_port: "8005" – сервисный порт tomcat
```

```
DAS_redirect_port: "443" – https порт Tomcat
```

```
DAS_ssl_config:
```

*keystore\_path: "{{tomcat\_properties.service\_path}}/certs/certstore" – путь к хранилищу ключей tomcat которое содержит сертификат для https*

*keystore\_password: "123456" – пароль от хранилища*

*DAS\_service\_name: "DAS-{{DAS\_project\_port}}" – имя сервиса внутри системы для обращение через команду service*

*DAS\_service\_module\_name: "{{ DAS\_service\_name }}.service" – имя файла описывающего системный сервис в systemd*

*DAS\_service\_template: "DAS\_module.j2" – имя шаблона для сервис файла*

*DAS\_prop\_template: "DAS\_properties.j2" – имя шаблона для файла конфигурации tomcat*

*DAS\_config: - блок конфигурации приложения*

*postgres: - блок конфигурации базы данных*

*user: "bftUA"*

*password: "123456"*

*database: " bftUADB"*

*ice\_db\_role\_name: "bftUA"*

*addr: "172.24.19.41"*

*port: "5432"*

*avapost: - блок конфигурации IAM на примере системы*

*Аванпост*

*enabled: true*

*domain\_url: "https://{{domain\_name}}"*

*app\_code: "35481e37-61a8-49fc-862b-4a96a1ee3c56"*

*app\_secret: "123456789"*

*ice\_url: "https://{{domain\_name}}/app"*

*av\_admin\_url: "https://srv-idp.example.com"*

*roles\_field: " bftUARoles"*

*converter*: - блок конфигурации адреса сервиса для конвертации файлов в pdf

*url*: "http://localhost:8881/lool/convert-to"

*s3*: - блок конфигурации s3-совместимого хранилища

*enabled*: true

*access\_key*: "gtdas"

*secret\_key*: "123456"

*bucket*: "gtas"

*url*: "http://172.24.19.48:9000"

*audit*: - блок конфигурации событий аудита

*enabled*: true

*url*: "http://localhost:8000/app/test"

*metamodels*: | - блок настроек метамodelей (задается блоком прямого текста)

*DAS\_app\_profiles*: "postgresql, common" – загружаемые профили приложения

*tomcat\_properties*: - блок конфигурации tomcat

*archive*: "apache-tomcat-9.0.44.tar.gz" – имя архива внутри

дистрибутива

*service\_path*: "/opt/\_Tomcat/"

*log\_dir*: "/var/log/tomcat/"

*tomcat\_user*:

*name*: "tomcat"

*group*: "tomcat"

15) После изменения параметров запустить команду установки из папки DAS:

*ansible-playbook --extra-vars="root\_passwd= " playbooks/DAS.yaml -i gt\_inventory*

### 1.3 Установка приложения БФТ. Управление активами

Работа в Системе доступна только для зарегистрированных пользователей.

Для перехода к окну авторизации в строке браузера вводится адрес сервера Системы. Откроется окно авторизации пользователя:

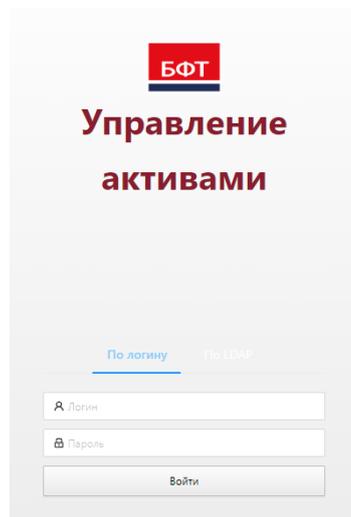


Рисунок 2 – Окно входа в систему

Для входа в систему необходимо указать следующую информацию:

- **Логин** – имя пользователя:
- **Пароль** – пароль пользователя.

Далее нажимается кнопка «**Войти**».

В случае ввода верной информации (указан зарегистрированный в системе пользователь и правильный пароль) осуществится вход в главное окно системы. При вводе неверного **Пользователя** или **Пароля** на экране появится сообщение об ошибке «**Неправильное имя пользователя или пароль**».