



ОБЩЕСТВО
С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«БЮДЖЕТНЫЕ И ФИНАНСОВЫЕ ТЕХНОЛОГИИ»

127018, г. Москва,
ул. Складочная, д. 3, строение 1
+7 (495) 784-70-00,
ineed@bftcom.com, www.bftcom.com

Система хранения и предоставления электронных документов «БФТ.е-Архив»

Инструкция по скачиванию, установке экземпляра программного обеспечения и запуску Системы хранения и предоставления электронных документов «БФТ.е-Архив»

Листов 18

СОДЕРЖАНИЕ

1.	Скачивание дистрибутива.....	3
2.	Установка системы «БФТ.е-Архив».....	3
2.1	Установка OpenJRE 8.....	3
2.2	Установка PostgreSQL-9.5.....	4
2.3	Создание базы.....	4
2.4	Установка и настройка Tomcat 8.....	4
2.5	Установка приложения «БФТ.е-Архив».....	7
2.6	Установка Hadoop.....	9
2.7	Установка ApacheMQ.....	13
2.8	Настройка портов брандмауэра.....	15
2.9	Настройка логирования.....	15
3.	Запуск Системы хранения и предоставления электронных документов «БФТ.е-Архив».....	17

1. Скачивание дистрибутива

Для получения дистрибутива необходимо пройти по ссылке, ввести пароль и скачать файл (app.war)из облачного хранилища.

Ссылка: <https://bftcloud.bftcom.com/index.php/s/hT8SA0MRkGYi60R>

Пароль: *Lp5iivj7*

2. Установка системы «БФТ.е-Архив»

Порядок установки

- 1) Установка OpenJRE 8;
- 2) Установка PostgreSQL-9.5;
- 3) Создание базы приложения;
- 4) Установка и настройка Tomcat 8;
- 5) Установка приложения «БФТ.е-Архив»;
- 6) Установка Nadoor;
- 7) Установка ApacheMQ;
- 8) Настройка портов брандмауэра;
- 9) Настройка логирования.

2.1 Установка OpenJRE 8

Для установки OpenJRE 8 необходимо выполнить следующие действия:

- 1) Перейти в home директорию:

```
cd ~
```

- 2) Обновить кэш:

```
sudo yum makecache
```

- 3) Установить OpenJDK:

```
sudo yum install java-1.8.0-openjdk
```

4) Выбрать java 8 по умолчанию:

```
sudo alternatives --config java
```

2.2 Установка PostgreSQL-9.5

Установить PostgreSQL-9.5

```
sudo yum install postgresql95-server postgresql95-contrib
```

Создать экземпляр базы:

```
sudo postgresql-setup initdb
```

Изменить аутентификации по паролю:

```
sudo vi /var/lib/pgsql/data/pg_hba.conf
```

```
host all all 127.0.0.1/32 ident
```

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

Изменить на:

```
host all all 127.0.0.1/32 md5
```

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

Установить пароль для пользователя postgres:

```
sudo -u postgres psql postgres
```

Ввести команду:

```
\password postgres
```

Ввести пароль.

Выйти из psql:

```
\q
```

2.3 Создание базы

Создать базу ice_docarch:

```
CREATE DATABASE ice_docarch;
```

2.4 Установка и настройка Tomcat 8

Установка Tomcat 8

Для установки Tomcat 8 необходимо:

Установить Tomcat:

```
sudo yum install tomcat8-user tomcat8
```

Создать инстанс в необходимой директории (название модуля и порты указать свои):

```
tomcat8-instance-create -p 8083 -c 8015 mdm
```

Настройка Tomcat 8

Положить файл application.properties в папку с томкатом.

```
dm.fieldSetPackages=com.bftcom.docarch.commons
```

```
ice.servicesPackages=com.bftcom.docarch.commons.service,  
com.bftcom.ice.common.service
```

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

```
spring.datasource.url=jdbc:postgresql://localhost:5432/ice_docarch
```

```
spring.datasource.username=postgres
```

```
spring.datasource.password=postgres
```

```
#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

docarch.archiveStartYear=2014

docarch.jms.url = tcp://localhost:61616
docarch.jms.username = admin
docarch.jms.password = admin
docarch.jms.outbound.queue.name = docArchiveJsonOutgoingQueue
docarch.jms.inbound.queue.name = docArchiveJsonIncomingQueue

docarch.file.encoding=UTF-8
docarch.file.directory=files/exchange
docarch.file.concurrency=2
docarch.file.pollInterval=1000

docarch.fsDirectory=files/attaches

docarch.hadoop.defaultFS=hdfs://localhost:9000
docarch.hadoop.user.name=hadoop
docarch.hadoop.home.dir=C:/winutils

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
```

Для выбора места хранения файлов - выбрать соответствующий набор параметров и добавить в application.properties.

```
#сохранение в файловую систему
```

```
docarch.storageType=fileStorage
```

```
docarch.fileStorage.directory=/tmp
```

```
#сохранение в хадуп
```

```
docarch.storageType=hadoop
```

```
docarch.hadoop.defaultFS=hdfs://172.21.11.218:9000
```

```
docarch.hadoop.user.name=hadoop
```

```
docarch.hadoop.home.dir=C:/winutils
```

```
#сохранение в базу
```

```
docarch.storageType=db
```

Прописать полный путь до этого файла в bin/startup.sh:

```
export CATALINA_OPTS="-
```

```
Dspring.config.location=file:полный_путь_до_application.properties"
```

2.5 Установка приложения «БФТ.е-Архив»

Скопировать файл приложения app.war в директорию \webapps сборки Tomcat;

Положить файл application.properties в папку с томкатом.

```
dm.fieldSetPackages=com.bftcom.docarch.common
```

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

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

```
spring.datasource.url=jdbc:postgresql://localhost:5432/ice_docarch
spring.datasource.username=postgres
spring.datasource.password=postgres
```

```
#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
```

```
docarch.archiveStartYear=2014
```

```
docarch.jms.url = tcp://localhost:61616
docarch.jms.username = admin
docarch.jms.password = admin
docarch.jms.outbound.queue.name = docArchiveJsonOutgoingQueue
docarch.jms.inbound.queue.name = docArchiveJsonIncomingQueue
```



```
docarch.file.encoding=UTF-8
docarch.file.directory=files/exchange
docarch.file.concurrency=2
docarch.file.pollInterval=1000
```

```
docarch.fsDirectory=files/attaches
```

```
docarch.hadoop.defaultFS=hdfs://localhost:9000
docarch.hadoop.user.name=hadoop
docarch.hadoop.home.dir=C:/winutils
```

```
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
```

Прописать полный путь до этого файла в bin/startup.sh :

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

2.6 Установка Hadoop

Создать пользователя hadoop:

```
useradd -d /opt/hadoop hadoop
passwd hadoop
```

Скачать hadoop:

```
curl -O http://apache.javapipeline.com/hadoop/common/hadoop-
2.7.2/hadoop-2.7.2.tar.gz
```

Разархивировать и поместить в opt:

```
tar xfz hadoop-2.7.2.tar.gz
cp -rf hadoop-2.7.2/* /opt/hadoop/
chown -R hadoop:hadoop /opt/hadoop/
```

Перейти под пользователя hadoop и отредактировать .bash_profile:

```
su - hadoop
vi .bash_profile
```

В конец файла добавить:

```
## JAVA env variables
export JAVA_HOME=/usr/java/default
export PATH=$PATH:$JAVA_HOME/bin
export
CLASSPATH=.:$JAVA_HOME/jre/lib:$JAVA_HOME/lib:$JAVA_HOME/
lib/tools.jar

## HADOOP env variables
export HADOOP_HOME=/opt/hadoop
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_OPTS="-
Djava.library.path=$HADOOP_HOME/lib/native"
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

Инициализировать переменные окружения:

```
source .bash_profile
echo $HADOOP_HOME
echo $JAVA_HOME
```

Настроить ssh соединение с пустой passphrase:

```
ssh-keygen -t rsa
ssh-copy-id master.hadoop.lan
```

Настройка hadoop кластера:

```
vi etc/hadoop/core-site.xml
```

Добавить следующие строки между `<configuration> ... </configuration>`, где `ip_comp` - ip вашего стенда:

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://ip_comp:9000/</value>
</property>
```

Отредактировать файл `hdfs-site.xml`:

```
vi etc/hadoop/hdfs-site.xml
```

Добавить следующие строки между `<configuration> ... </configuration>`:

```
<property>
  <name>dfs.data.dir</name>
  <value>file:///opt/volume/datanode</value>
</property>
<property>
```

```
<name>dfs.name.dir</name>
<value>file:///opt/volume/namenode</value>
</property>
```

Создать директории:

```
su root
mkdir -p /opt/volume/namenode
mkdir -p /opt/volume/datanode
chown -R hadoop:hadoop /opt/volume/
exit
```

Отредактировать mapred-site.xml:

```
vi etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Отредактировать yarn-site.xml:

```
vi etc/hadoop/yarn-site.xml
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
```

Настроить hadoop-env.sh.

Прописать путь до java:

```
export JAVA_HOME=/usr/java/default/ -- изменить на нужный путь
```

Сформировать hdfs:

```
hdfs namenode -format
```

Запускаем hdfs:

```
start-dfs.sh
```

```
start-yarn.sh
```

Настройка автозапуска:

```
su - root
```

```
vi /etc/rc.local
```

Добавить строки:

```
su - hadoop -c "/opt/hadoop/sbin/start-dfs.sh"
```

```
su - hadoop -c "/opt/hadoop/sbin/start-yarn.sh"
```

```
exit 0
```

```
chmod +x /etc/rc.d/rc.local
```

```
systemctl enable rc-local
```

```
systemctl start rc-local
```

```
systemctl status rc-local
```

2.7 Установка ApacheMQ

Скачать apacheMQ:

```
wget https://archive.apache.org/dist/activemq/5.14.3/apache-activemq-5.14.3-bin.tar.gz
```

Распаковать:

```
sudo tar -zxvf apache-activemq-5.14.3-bin.tar.gz -C /opt
```

Сделать линк:

```
sudo ln -s /opt/apache-activemq-5.14.3 /opt/activemq
```

Создать файл для запуска apacheMQ как сервиса:

```
sudo vi /usr/lib/systemd/system/activemq.service
```

```
[Unit]
```

```
Description=activemq message queue
```

```
After=network.target
```

```
[Service]
```

```
PIDFile=/opt/activemq/data/activemq.pid
```

```
ExecStart=/opt/activemq/bin/activemq start
```

```
ExecStop=/opt/activemq/bin/activemq stop
```

```
User=root
```

```
Group=root
```

```
[Install]
```

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

Сохранить.

Запустить как сервис:

```
sudo systemctl enable activemq.service
```

```
sudo systemctl start activemq.service
```

Запустить томкат:

```
./bin/startup.sh
```

2.8 Настройка портов брандмауэра

Для возможности подключения к серверу с других машин в сети на данной машине необходимо разрешить порту приложения (порт приложения указан в файле server.xml в поддиректории Tomcat'a conf – в параметре port тега Connector) принимать запросы извне.

```
sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp
```

Для применения правил необходимо их перезагрузить:

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

2.9 Настройка логирования

Для настройки логирования необходимо:

Создать файл logger.xml в директории Tomcat'a с содержимым:

```
<configuration>
  <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%d{YYYY-MM-dd HH:mm:ss.SSS} [%thread] %-5level
%logger{36} - %msg%n</pattern>
    </encoder>
  </appender>
  <appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender"> <!--
логгер
работает с файлом -->
  <File>logs/access.log</File> <!--расположение файла -->
  <rollingPolicy <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy"><!--
политика
для
```

разделения файлов логов по времени -->

```
<FileNamePattern>logs/access.%d{yyyy-MM-dd}.%i.log.zip</FileNamePattern> <!--
```

шаблон для

имени файла, в который будут архивироваться логи -->

```
<timeBasedFileNamingAndTriggeringPolicy  
class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">
```

```
    <maxFileSize>10MB</maxFileSize> <!--максимальный
```

размер файлов логов-->

```
</timeBasedFileNamingAndTriggeringPolicy>
```

```
    </rollingPolicy>
```

```
    <append>true</append>
```

```
    <immediateFlush>true</immediateFlush>
```

```
    <encoder>
```

```
        <pattern>%d{YYYY-MM-dd HH:mm:ss.SSS}
```

```
%msg%n</pattern>
```

```
    </encoder>
```

```
</appender>
```

```
<logger name="com.bftcom.ice.server.logging.AccessLoggingFilter">
```

```
    <appender-ref ref="FILE"/>
```

```
    <level value="DEBUG"/>
```

```
</logger>
```

```
<root level="INFO">
```

```
    <appender-ref ref="STDOUT"/>
```

```
</root>
```

```
<logger
```

```
name="com.bftcom.docarch.server.rest.DocArchiveDocumentController
```

```
" level="DEBUG"/>
```



```
<logger name="com.bftcom.ice.server.services.DataServiceImpl"  
level="OFF"/>  
<logger name="com.bftcom.ice.server.services.DeltaMachine"  
level="OFF"/>  
<logger  
name="com.bftcom.ice.server.security.SecuredDataServiceImpl"  
level="OFF"/>  
</configuration>
```

Прописать в catalina.sh:

```
export CATALINA_OPTS="-  
Dlogback.configurationFile=$CATALINA_HOME/logback.xml  
-Dlogging.config=file:$CATALINA_HOME/logback.xml  
-Dspring.config.location=file:$CATALINA_HOME/application.properties"
```

3. Запуск Системы хранения и предоставления электронных документов «БФТ.е-Архив»

Работа в платформе доступна только для зарегистрированных пользователей.

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

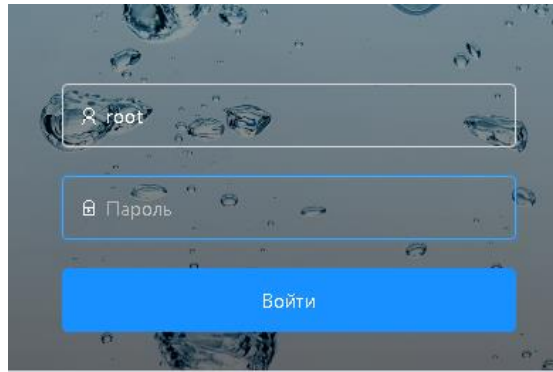


Рисунок 1 – Окно входа в систему

Для входа в систему необходимо указать следующую информацию:

- **Логин:** «root» - имя пользователя.
- **Пароль:** «root» - пароль пользователя.

Далее нажимается кнопка **Войти**.

В случае ввода верной информации (указаны 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.