

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ЭКСПЛУАТАЦИИ**  
**программного обеспечения**  
**«Digit Платформа»**

## Содержание

- 1 Перечень программного обеспечения4
- 2 Требования к серверному оборудованию5
- 3 Инструкция по установке компонентов в ручном режиме6
  - 3.1 Подготовка6
  - 3.2 Настройка Java6
  - 3.3 Настройка БД (OrientDB)6
  - 3.4 Настройка Apache Solr8
  - 3.5 Настройка Файлового сервиса9
  - 3.6 Настройка адресного сервиса13
  - 3.7 Настройка сервиса подписания15
  - 3.8 Настройка Digit Платформы17
  - 3.9 Настройка подсистемы Интеграции19
- 4 Описание действий по проверке корректности установки компонентов и их настройке21
- 5 Инструкция по эксплуатации22

## СПИСОК СОКРАЩЕНИЙ

<b>БД</b>	База данных
<b>АРМ</b>	Автоматизированное рабочее место - индивидуальный комплекс технических и программных средств, предназначенный для автоматизации профессионального труда специалиста и обеспечивающий подготовку, редактирование, поиск, отображение на экране и печать необходимых ему документов и данных.
<b>КСПД</b>	Корпоративная сеть передачи данных – телекоммуникационная сеть, объединяющая в единое информационное пространство все краевые структурные подразделения органов власти

# 1 ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для развертывания программного обеспечения «Digit Платформа» программно-аппаратный комплекс должен удовлетворять следующему требованию: сервер должен функционировать под управлением операционной системы Astra Linux.

Требования к операционной системе Astra Linux:

- astra 1.7.5
- Kernel: Linux 6.1.90-1-generic
- Архитектура: x86\_64

## **2 ТРЕБОВАНИЯ К СЕРВЕРНОМУ ОБОРУДОВАНИЮ**

Требования к серверному оборудованию для размещения программного обеспечения «Digit Платформа»:

- 8-16 Гб оперативной памяти;
- 200 Гб физической памяти;
- 8 процессора с 4 логическими ядрами каждый.

### 3 ИНСТРУКЦИЯ ПО УСТАНОВКЕ КОМПОНЕНТОВ В РУЧНОМ РЕЖИМЕ

Для развертывания программного обеспечения «Digit Платформа» необходимо выполнить следующие действия в установленном порядке:

#### 3.1 Подготовка

Скачать архив с <https://cloud.smart-consulting.ru/s/SEeAwSAnLGsN3TE>

Password: по запросу

Разархивировать его. Zip архив содержит дистрибутивы для установки.

#### 3.2 Настройка Java

Скачать jre bellsoft:

```
wgethttps://download.bell-sw.com/java/8u312+7/bellsoft-jre8u312+7-linux-amd64.tar.gz -O /var/tmp/bellsoft-jre8u312+7-linux-amd64.tar.gz
```

Подготовить java:

```
tar -xf/tmp/bellsoft-jre8u312+7-linux-amd64.tar.gz -C /opt/  
chmod-R 755 /opt/jre8u312
```

Проверка установки:

```
java -version
```

#### 3.3 Настройка БД (OrientDB)

##### 1. Настройка фаерволов

Разрешить порт 2480 для настройки и 2424 для сервера приложения

```
ufwdefaultdenyincoming
```

```
ufwdefaultallowoutgoing
```

```
ufwallowssh
```

```
ufwallow2480
```

```
ufwallow2424
```

```
ufwenable
```

```
gtyrfpcnfj,h
```

2. Скачать сервер баз данных orientdb-community-2.2.21.zipи vis\_test.zip в /tmp

4. Подготовить java

```
tar-xf/tmp/bellsoft-jre8u312+7-linux-amd64.tar.gz -C /opt/  
chmod-R 755 /opt/jre8u312
```

5. Скопировать данные из /tmp/digit-platform-db/orientdb в директорию /opt/  
unzip/tmp/orientdb-community-2.2.21.zip-d /opt/ &&mv/opt/orientdb-community-  
2.2.21/opt/orientdb

6. Создать пользователя orient:

```
useraddorient  
chown-Rorient:orient/opt/orientdb  
chmod-R 755 /opt/orientdb/bin/  
chmod-R 755 /opt/jre8u312
```

7. Поправить файл в /etc/systemd/system/orientdb.service:

```
[Unit]  
Description=OrientDBServer  
After=network.target  
After=syslog.target  
[Install]  
WantedBy=multi-user.target  
[Service]  
Environment=JAVA_HOME=/opt/jre8u312  
User=orient  
Group=orient  
ExecStart=/opt/orientdb/bin/server.sh  
ExecStop=/opt/orientdb/bin/shutdown.sh
```

8. До того, как стартовать БД, изменить пароль для рута в конфигурационном файле:/opt/orientdb/config/orientdb-server-config.xml в секции <users>:

```
<userresources="*"password="root"name="root"/>
```

8. Стартовать сервер и добавить orientdb в автозагрузку:

```
systemctlenable--noworientdb
```

10. Ручное создание БД:

Создать БД через консоль.

```
su-c
```

```
"sh/opt/orientdb/bin/console.sh"orientdbcreatedatabaseplocal:/opt/orientdb/databases/visdbrootroot  
plocaldocument
```

10.1 Импорт БД:

Разархивировать vis\_test.zip в /opt/orientdb/databases

10.2 Настройка адреса сервиса подписания для сервера приложения:

Проверить адрес для сервиса подписания можно поглядеть запросом в бд:

```
select*fromPersistedVisConfigwherename= 'signServerUrl';
```

Если значение отсутствует, выполнить:

```
insertintoPersistedVisConfig(name,value)values('signServerUrl',      'http://адрес_сервиса  
подписания:8181/rest/sign')
```

### **3.4 Настройка Apache Solr**

1. Настройка фаерволов

Разрешить порт 8983 для сервера приложения

```
ufw default deny incoming
```

```
ufw default allow outgoing
```

```
ufw allow ssh
```

```
ufw allow 8983
```

```
ufw enable
```

2. Скачать solr-7.2.0.zip и solrDigitConf.zip в vis\_test.zip /tmp/

3. Установить solr:

```
unzip /tmp/solr-7.2.0.zip -d /opt/ && mv /opt/solr-7.2.0 /opt/solr
```

4. Создать пользователя solr

```
adduser solr
```

```
chown solr:solr -R /opt/solr
```

```
chmod -R 755 /opt/solr
```

```
chmod -R 755 /opt/jre8u312
```

5. Создать юнит-скрипт в /etc/systemd/system/solr.service

```
[Unit]
```

```
Description=Apache SOLR
```

```
After=syslog.target network.target
Before=multi-user.target graphical.target
Conflicts=shutdown.target
[Service]
Environment=SOLR_INCLUDE=/opt/solr/bin/solr.in.sh
Type=forking
WorkingDirectory=/opt/solr/
User=solr
PIDFile=/opt/solr/bin/solr-8983.pid
ExecStart=/opt/solr/bin/solr start
ExecStop=/opt/solr/bin/solr stop
[Install]
WantedBy=multi-user.target graphical.target
```

6. Для создания ядра использовать шаблон

```
unzip /tmp/solrDigitConf.zip -d /opt/solr/server/solr/configsets/solrDigitConf
unzip /tmp/vis_test.zip -d /opt/solr/server/solr/vis_test
```

7. Запустить solr и добавить в автозагрузку solr.

```
systemctl enable --now solr
```

8. Добавить ядро солр:

Зайти по адресу вм солр <http://solr.ip.address:8983/solr>

И создать ядро vis\_test

### **3.5 Настройка Файлового сервиса**

1. Настройка фаерволов

Разрешить порт 8088 для сервера приложения

```
ufw default deny incoming
```

```
ufw default allow outgoing
```

```
ufw allow ssh
```

```
ufw allow 8088
```

```
ufw enable
```

2. Скачать digitfs-server.jar и digit\_fs\_schema.sql в /tmp/

### 3. Скачать jre bellsoft

```
wget https://download.bell-sw.com/java/8u312+7/bellsoft-jre8u312+7-linux-amd64.tar.gz -  
O /var/tmp/bellsoft-jre8u312+7-linux-amd64.tar.gz
```

### 4. Установить Postgres pro 12

```
curl -o apt-repo-add.sh https://repo.postgrespro.ru/pgpro-12/keys/apt-repo-add.sh  
sh apt-repo-add.sh  
apt-get install postgrespro-12 postgrespro-contrib-12
```

5. В файле `/etc/postgresql/12/main/pg_hba.conf` прописать возможность подключения к бд:

```
# "local" is for Unix domain socket connections only  
local all all trust  
# IPv4 local connections:  
host all all 127.0.0.1/32 md5  
host all all x.x.x.x/32 md5
```

где `x.x.x.x/32` — адрес сервера приложений digitFS (если на той же машине, то только 127.0.0.1)

6. В файле `/etc/postgresql/9.6/main/postgresql.conf` в Connection Settings привести к виду:

```
listen_addresses = '*'  
port = 5432  
max_connections = 100
```

7. Запуск сервиса `postgresql-12` и автозапуск

```
systemctl enable --now postgresql
```

8. Проверка портов, поднятых базой данных (процесс `postgres`)

```
netstat -plnt
```

9. Создать пользователя и базу:

```
su postgres
```

```
psql
```

```
CREATE ROLE atcfs;
```

```
CREATE DATABASE atcfsdb OWNER atcfs;
```

```
ALTER ROLE atcfs ENCRYPTED PASSWORD 'atcfs';
```

```
GRANT CONNECT ON DATABASE atcfsdb TO atcfs;
ALTER ROLE atcfs LOGIN;
\q
```

10. Залить схему данных:

```
psql -h 127.0.0.1 -p 5432 -W -U atcfs atcfsdb <
/tmp/digit_platcform_atcfs/digit_fs_schema.sql
```

11. Проверяем:

```
psql -h 127.0.0.1 -p 5432 -W -U atcfs atcfsdb
```

12. Подготовить сервер файлового хранилища:

```
mkdir -p /opt/atcfs/atcfs_storage
mkdir /opt/atcfs/log
mkdir /opt/atcfs/treefilestorage
cp -r /tmp/digitfs-server.jar /opt/atcfs
```

13. Создать файл /opt/atcfs/application.properties с содержимым:

```
spring.datasource.url=jdbc:postgresql://127.0.0.1:5432/atcfsdb?charset=UTF8
spring.datasource.username=atcfs
spring.datasource.password=atcfspass
spring.datasource.driverClassName=org.postgresql.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database=POSTGRESQL
spring.jpa.show-sql=false
filestorage.directory=/opt/atcfs/atcfs_storage
DBFileStorage.enabled=false
filestorage.filesperdirectory=500
filestorage.partialdirnameprefix=_
treefilestorage.directory=/opt/atcfs/treefilestorage
tmplinks.lifetime.seconds=300
server.port=8088
server.contextPath=/atcfs
logging.file=/opt/atcfs/log/atcfs.log
```

```
logging.path=/opt/atcfs/log
verifySign.checkOnlySign=true
verifySign.url=http://smev-mvf.test.gosuslugi.ru:7777/gateway/services/SID0003064
writelog=true
setSignStatus.enabled=false
spring.datasource.testOnBorrow=true
spring.datasource.validationQuery=SELECT 1;
spring.datasource.validationInterval=30000
spring.datasource.testWhileIdle=true
spring.datasource.timeBetweenEvictionRunsMillis=10000
```

15. Создать юнит-скрипт в /etc/systemd/system/atcfs.service

[Unit]

Description=ATCFS file storage daemon

After=network.target

[Service]

StandardOutput=null

StandardError=journal

WorkingDirectory=/opt/atcfs/

ExecStart=/opt/jre8u312/bin/java -Xmx4G -jar digitfs-server.jar

User=atcfs

Group=atcfs

[Install]

WantedBy=multi-user.target

16. Добавить пользователя atcfs и назначить его владельцем /opt/atcfs

```
useradd atcfs
```

```
chown -R atcfs: /opt/atcfs
```

17. Запустить сервис atcfs и добавить его в автозагрузку

```
systemctl enable --now atcfs
```

19. Проверить работоспособность

```
curl -s http://127.0.0.1:8088/atcfs/monitoring
```

20. Важное замечание: при использовании БД для хранения файлов необходимо учитывать это и создавать раздел для `/var/lib/postgresql` максимально большим

### 3.6 Настройка адресного сервиса

#### 1. Настройка фаерволов

Разрешить порт 8983 для сервера приложения

```
ufw default deny incoming
```

```
ufw default allow outgoing
```

```
ufw allow ssh
```

```
ufw allow 8983
```

```
ufw allow 8180
```

```
ufw enable
```

2. Скачиваем `solr-8.11.2.zip` `fias.jar` `schema.xml` и `synonyms.txt` в `/tmp/`

3. Устанавливаем `solr`:

```
wget https://dlcdn.apache.org/lucene/solr/8.11.2/solr-8.11.2.zip -O /tmp/solr-8.11.2.zip
```

```
unzip /tmp/solr-8.11.2.zip -d /opt/ && mv /opt/solr-8.11.2 /opt/solr
```

4. Создаем ядро `/opt/solr/server/solr/fias_core`

5. Подкладываем в `/opt/solr/server/solr/fias_core` файлы `/tmp/schema.xml` и `/tmp/synonyms.txt`

6. Создаем пользователя `solr`

```
adduser solr
```

```
chown solr:solr -R /opt/solr
```

```
chmod -R 755 /opt/solr
```

```
chmod -R 755 /opt/jre8u312
```

7. Создаем юнит-скрипт в `/etc/systemd/system/solr.service`

```
[Unit]
```

```
Description=Apache SOLR
```

```

After=syslog.target network.target
Before=multi-user.target graphical.target
Conflicts=shutdown.target
[Service]
Environment=SOLR_INCLUDE=/opt/solr/bin/solr.in.sh
Type=forking
WorkingDirectory=/opt/solr/
User=solr
PIDFile=/opt/solr/bin/solr-8983.pid
ExecStart=/opt/solr/bin/solr start
ExecStop=/opt/solr/bin/solr stop
[Install]
WantedBy=multi-user.target graphical.target

```

8. Запускаем solr и Добавляем в автозагрузку solr.  
systemctl enable --now solr

#### 9. Установка ФИАС

```

mkdir /opt/fias
cp /tmp/fias.jar /opt/fias/

```

#### 10. Добавить скрипты запуска\остановки

```

/opt/fias/start.sh
#!/bin/bash
echo "Killing fias service if starting"
kill -15 `cat /opt/fias/fias.pid` || true
echo "Starting fias service"
cd /opt/fias/
/opt/jdk1.8.0_211/bin/java -jar /opt/fias/fias.jar --server.port=8180 --
fias.solr.url=http://127.0.0.1:8983/solr/fias-core </dev/null &>/dev/null & echo $! >/opt/fias/fias.pid
echo "Fias service started"

/opt/fias/stop.sh
#!/bin/bash
echo "Stopping fias service"
kill -15 `cat /opt/fias/fias.pid` || true

```

```
echo "Fias service success killed"
```

11. Создать директорию /opt/fiasdb/ и /opt/fiasdb/fiasData

12. Создать пользователя fias

```
adduser fias
```

```
chown solr:solr -R /opt/fias
```

```
chown solr:solr -R /opt/fiasdb
```

```
chmod -R 755 /opt/fias
```

```
chmod -R 755 /opt/fiasdb
```

13. Создать сервис /etc/systemd/system/fias.service

```
[Unit]
```

```
Description=Fias Service App
```

```
After=network.target
```

```
[Service]
```

```
Type=forking
```

```
ExecStart=/opt/fias/start.sh
```

```
ExecStop=/opt/fias/stop.sh
```

```
User=fias
```

```
Group=fias
```

```
[Install]
```

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

14. Запустить ФИАС

15. запустить процесс построения индекса для всех регионов командой:

```
curl http://127.0.0.1:8180/fias-address-rest/rest/regions/updateAll
```

Ориентировочное время завершения процесса - ~48 часов, в это время ФИАС будет недоступен

### **3.7 Настройка сервиса подписания**

1. Разрешить порт 8181 для сервиса подписания

```
ufw default deny incoming
```

```
ufw default allow outgoing
```

```
ufw allow ssh
```

```
ufw allow 8181
```

ufw enable

2. Используем сертифицированную CryptoPro JCP 2.0.41789 скачанную с официального сайта cryptopro, установка согласно инструкции

Для этого потребуется наличие лицензионного ключа для jcp2.0, при его наличии при инсталляции, которая будет указана ниже, будет необходимо указать его. В противном случае предстоит установить демо версию, которая работает три месяца.

Необходимо будет установить crypto pro как провайдер Java, для этого необходимо воспользоваться встроенной командой CryptoPro JCP, где указать путь до используемой JDK

```
setup_console.sh path_to_JDK
```

3. Подложить ключи CryptoPro в контейнер либо использовать flash накопитель для используемой JDK

4. Установить Tomcat 8

```
sudo apt-get install tomcat8
```

Не забыть настроить переменную среды JAVA\_HOME, либо прописать ее в tomcat /etc/default/tomcat8

5. Скачиваем ROOT.war и подкладываем в директорию tomcat, после старта сервис будет доступен на порту 8181

Скорректировать скрипты и конфигурации:

```
/var/lib/tomcat8/webapps/ROOT/WEB_INF/classes/application.properties
```

```
server.port=8181
```

```
logging.file=logs/signservice.log
```

```
sign.store.path=/var/opt/cprosp/keys
```

```
sign.store.user=signuser
```

```
admin.username=admin
```

```
admin.password= по
```

```
spring.mvc.view.prefix=/public/
```

```
spring.mvc.view.suffix=.html
```

```
spring.main.allow-bean-definition-overriding=true
```

6. Запустить tomcat8

```
/etc/init.d/tomcat8 start
```

Проверить работу приложения

проверка со стороны администраторов заключается в curl запросе:

```
curl 127.0.0.1:8181/rest/keys
```

Сервис должен отдать 200й статус и контент страницы

## 3.8 Настройка Digit Платформы

### 1. Настройка фаерволов

Разрешить порт 8080 для сервиса

```
ufw default deny incoming
```

```
ufw default allow outgoing
```

```
ufw allow ssh
```

```
ufw allow 8080
```

```
ufw enable
```

### 2. Скачиваем компоненты приложения wildfly-20.0.1.Final.zip vis\_conf.zip и ROOT.war

в /tmp

### 3. Копируем скрипт запуска в /etc/systemd/system/wildfly.service

```
[Unit]
```

```
Description=WildFly application server
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
StandardOutput=null
```

```
StandardError=journal
```

```
WorkingDirectory=/opt/wildfly
```

```
User=wildfly
```

```
Group=wildfly
```

```
ExecStart=/opt/wildfly/bin/standalone.sh
```

```
[Install]
```

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

### 4. Добавляем пользователя wildfly.

```
useradd wildfly
```

### 5. Назначаем права в системе:

```
chown -R wildfly: /opt/wildfly
```

```
chmod -R 755 /opt/jre8u312
```

6. Предоставленный ROOT.war копируем в /opt/wildfly/standalone/deployments

7. Прописываем корректные адреса в следующих конфигурационных файлах:

```
/opt/wildfly/standalone/configuration/vis_conf
```

Полный список конфигураций тут

- /opt/wildfly/standalone/configuration/vis\_conf/atcfs.properties - информация по подключению сервиса digitfs(необходимо указать порт подключения к порту 8088 файлового хранилища)

- /opt/wildfly/standalone/configuration/vis\_conf/solr.properties - информация по подключению полнотекстового сервиса (указывается адрес и порт solr)

- /opt/wildfly/standalone/configuration/vis\_conf/orient.properties - информация по подключению сервера БД

- /opt/wildfly/standalone/configuration/vis\_conf/esia.properties - информация по подключению авторизации в esia (опционально)

- /opt/wildfly/standalone/configuration/vis\_conf/custom.properties - информация по подключению сервера федеральной информационной системы + кастомизированных настроек(опционально)

- /opt/wildfly/standalone/configuration/vis\_conf/workspaces.properties - дополнительное описание подключения полнотекстового поиска + БД

- /opt/wildfly/standalone/configuration/vis\_conf/email.properties - информация по добавлению возможности отправки данных через почту (опционально)

8. Активируем сервис и вносим в автозагрузку

```
systemctl enable --now wildfly
```

9. Проверяем работоспособность системы при помощи мониторинга.

```
curl localhost:8080/rest/monitoring
```

## 3.9 Настройка подсистемы Интеграции

### 1. Настройка фаерволов

Разрешить порт 9080 для сервера приложения

```
ufw default deny incoming
```

```
ufw default allow outgoing
```

```
ufw allow ssh
```

```
ufw allow 9080
```

```
ufw enable
```

### 2. Скачиваем jre bellsoft

```
wget -q -O - https://download.bell-sw.com/pki/GPG-KEY-bellsoft | sudo apt-key add - echo  
"deb [arch=amd64] https://apt.bell-sw.com/ stable main" | sudo tee /etc/apt/sources.list.d/bellsoft.list  
sudo apt-get update  
sudo apt-get install bellsoft-java8
```

### 3. Подготавливаем сервер файлового хранилища:

```
mkdir -p /opt/dataserver/
```

```
mkdir /opt/dataserver/deployments
```

```
mkdir /opt/dataserver/jslibraries
```

```
mkdir /opt/dataserver/logs
```

распаковываем архив `dataserver.zip` в папку `/opt/dataserver/`

### 4. Корректируем файл `/opt/dataserver/deployments/dataset/server/ datasources.js`:

в разделе подключить хранилище `atcfs` (если необходимо проверить Базу данных `postgres`)

```
"localStorage": {  
  type: "JDBC",  
  connection: {  
    url: "jdbc:postgresql://localhost:5432/etl",  
    username: "postgres",  
    password: "postgres"  
  }  
}
```

где указать в url адрес хоста atcfs (доступный по сети) из раздела 3.5 и указать корректные username и password

5. Создаем юнит-скрипт в /etc/systemd/system/dataserver.service

[Unit]

Description=Dataserver daemon

After=network.target

[Service]

StandardOutput=null

StandardError=journal

WorkingDirectory=/opt/dataserver/

ExecStart=/opt/jdk8u312/bin/java -Xmx4G -jar digitfs-server.jar

User=dataserver

Group= dataserver

[Install]

WantedBy=multi-user.target

6. Добавляем пользователя dataserver и назначаем его владельцем /opt/ dataserver  
useradd dataserver

chown -R dataserver: /opt/dataserver

7. Запускаем сервис dataserver и добавляем его в автозагрузку  
systemctl enable --now dataserver

8. Проверяем работоспособность

curl -s http://127.0.0.1:9080/actuator/

## 4 ОПИСАНИЕ ДЕЙСТВИЙ ПО ПРОВЕРКЕ КОРРЕКТНОСТИ УСТАНОВКИ КОМПОНЕНТОВ И ИХ НАСТРОЙКЕ

Проверка работоспособности системы производится через интерфейс пользователя. Для проверки работоспособности необходимо выполнить следующие действия:

Для организации доступа к программному обеспечению «Digit Платформа» на АРМ пользователя, необходимо выполнить следующие действия:

- 1) проверить, что в целях обеспечения возможности получения авторизованного доступа рабочее место сотрудника имеет подключение к КСПД или подключение к сети Интернет.
- 2) проверить, что на АРМ пользователя установлен веб-браузер Google Chrome или Mozilla Firefox актуальной версии.
- 3) в адресной строке веб-браузера ввести адрес приложения `http://hostName/`  
ВАЖНО: нужно обращаться к приложению через доменное имя, можно локально зарегистрировать его через hosts.
- 4) Для проверки того, что электронные подписи, подложенные в `/opt/Docker/certs/` доступны системе необходимо перейти в браузере по адресу:  
[http:// hostName:8082/rest/keys](http://hostName:8082/rest/keys)
- 5) Проверка доступности сервиса приема сообщений извне – перейти в браузере по адресу:  
`http://hostName/OrderService?wsdl`

## **5 ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ**

Эксплуатация программно-аппаратного комплекса, на котором развернуто программное обеспечение «Digit Платформа» производится по компонентно, в зависимости от состава компонент, входящих в программно-аппаратный комплекс.

Эксплуатация программного обеспечения «Digit Платформа» описана в документах «Digit Платформа: руководство администратора» и «Digit Платформа: руководство пользователя».