



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«Digit PROконтакт»

Инструкция по установке и эксплуатации ПО

Листов 13

СОДЕРЖАНИЕ

1	ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	4
2	ТРЕБОВАНИЯ К СЕРВЕРНОМУ ОБОРУДОВАНИЮ	5
3	ИНСТРУКЦИЯ ПО УСТАНОВКЕ КОМПОНЕНТОВ ПРИЛОЖЕНИЯ В РУЧНОМ РЕЖИМЕ	6
3.1	Подготовка	6
3.2	Настройка Java	6
3.3	Установка докера.....	6
3.4	Разворот окружения	7
3.5	Настройка файлового сервиса	7
3.6	Разворот ПО «Digit PROконтакт».....	9
4	ИНСТРУКЦИЯ ПО УСТАНОВКЕ КОМПОНЕНТОВ АТС В РУЧНОМ РЕЖИМЕ	10
4.1	Подготовка сервера	10
4.2	Установка пакетов, необходимых для сборки Asterisk	10
4.3	Сборка Asterisk	11
4.4	Конфигурация Asterisk	11
4.5	Настройка базы данных Asterisk.....	11
4.6	Запуск и проверка работы сервера Asterisk	12
5	ОПИСАНИЕ ДЕЙСТВИЙ ПО ПРОВЕРКЕ КОРРЕКТНОСТИ УСТАНОВКИ КОМПОНЕНТОВ И ИХ НАСТРОЙКЕ.....	13

СПИСОК СОКРАЩЕНИЙ

Сокращение	Наименование
АРМ	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
АТС	Автоматическая телефонная станция
БД	База данных
КСЦД	Корпоративная сеть передачи данных – телекоммуникационная сеть, объединяющая в единое информационное пространство все краевые структурные подразделения органов власти
ПО	Программное обеспечение

1 ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для развертывания программного обеспечения «Digit PROконтакт» программно-аппаратный комплекс должен иметь сервер АТС и сервер приложений, функционирующие под управлением операционными системы Astra Linux / ALT Linux.

Требования к операционным системам Astra Linux / ALT Linux:

- Архитектура: x86_64.

2 ТРЕБОВАНИЯ К СЕРВЕРНОМУ ОБОРУДОВАНИЮ

Требования к серверному оборудованию для размещения программного обеспечения «Digit PROконтакт» (приложение):

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

Требования к серверному оборудованию для размещения программного обеспечения «Digit PROконтакт» (АТС):

- 8-16 ГБ оперативной памяти;
- 200 ГБ физической памяти;
- 8 процессоров с 4 логическими ядрами каждый.

3 ИНСТРУКЦИЯ ПО УСТАНОВКЕ КОМПОНЕНТОВ ПРИЛОЖЕНИЯ В РУЧНОМ РЕЖИМЕ

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

- 1) подготовка;
- 2) настройка Java;
- 3) установка докера;
- 4) разворот окружения;
- 5) настройка файлового сервиса;
- 6) разворот ПО «Digit PROконтакт».

3.1 Подготовка

Скачать архив с <ftp://ftp.smart-consulting.ru/> из папки *rop_ftp/ucc_2022/ucc.zip*.

Username: *по запросу*

Password: *по запросу*

Разархивировать его. Zip-архив содержит дистрибутивы для установки.

3.2 Настройка Java

Скачать jre bellsoft:

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

```
wget https://download.bell-sw.com/java/17.0.4+8/bellsoft-jre17.0.4+8-linux-amd64.tar.gz -  
O /tmp/bellsoft-jre17.0.4+8-linux-amd64.tar.gz
```

Подготовить Java:

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

```
tar -xf /tmp/bellsoft-jre17.0.4+8-linux-amd64.tar.gz -C /opt/
```

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

```
chmod-R 755 /opt/jre17.0.4
```

3.3 Установка докера

Все окружение будет развернуто в докере, поэтому сначала необходимо установить сам докер:

```
sudo yum install -y yum-utils
```

```
sudo yum-config-manager --add-repo \
```

```
https://download.docker.com/linux/centos/docker-ce.repo
```

```
systemctl enable --now docker
```

```
sudo curl -L "https://github.com/docker/compose/releases/download/v2.6.1/docker-compose-  
linux-x86_64" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
sudo groupadd docker
```

```
sudo usermod -aG docker $USER
```

Проверка:

```
docker -v
```

```
docker-compose -v
```

3.4 Разворот окружения

На этом этапе будут развернуты БД, *solr*, *apache kafka* и *zookeeper*:

```
cp -R /tmp/ucc/environment/ /opt/
```

```
cd /opt/environment/
```

```
docker-compose up -d
```

Проверка:

```
docker ps
```

Будет выведено 4 docker-процесса (*postgres14*, *solr8*, *zookeeper*, *broker*).

3.5 Настройка файлового сервиса

1. Скачать *digitfs-server.jar* в */tmp/*.

2. 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
```

3. Создать файл */opt/atcfs/application.properties* с содержимым:

```
spring.datasource.url=jdbc:postgresql://127.0.0.1:5434/atcfs?charset=UTF8
```

```
spring.datasource.username=ucc
```

```
spring.datasource.password=ucc
```

```
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
```

```
flyway.enabled=true
```

```
flyway.checkLocation=false
```

flyway.locations=classpath:atc/fs/migration/postgresql
flyway.baseline-version=00001
flyway.baseline-on-migrate=true
flyway.sql-migration-prefix=V
flyway.sql-migration-separator=_
flyway.placeholders.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
```

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

3.6 Разворот ПО «Digit ПРОконтакт»

1. Достаем сборки:

```
cp -R /tmp/ucc/ucc-core/ucc /opt/
```

```
cp -R /tmp/ucc/ucc-core/ami /opt/
```

```
cp -R /tmp/ucc/ucc-core/ucc-media /opt/
```

```
cp -R /tmp/ucc/ucc-core/callservice /opt/
```

```
cp -R /tmp/ucc/ucc-core/ucc-messaging/ /opt/
```

2. Переносим сервисы:

```
mv /opt/ucc/ucc.service /etc/systemd/system/
```

```
mv /opt/ami/ami.service /etc/systemd/system/
```

```
mv /opt/callservice/callservice.service /etc/systemd/system/
```

```
mv /opt/ucc-media/ucc-media.service /etc/systemd/system/
```

```
mv /opt/ucc-media/ucc-messaging.service /etc/systemd/system/
```

3. Запускаем:

```
systemctl enable --now ucc
```

```
systemctl enable --now callservice
```

```
systemctl enable --now ucc-media
```

```
systemctl enable --now ami
```

```
systemctl enable --now ucc-messaging
```

4 ИНСТРУКЦИЯ ПО УСТАНОВКЕ КОМПОНЕНТОВ АТС В РУЧНОМ РЕЖИМЕ

Установка компонента АТС выполняется на хосте под управлением операционных систем Astra Linux / ALT Linux и состоит из следующих этапов:

- 1) подготовка сервера;
- 2) установка пакетов, необходимых для сборки Asterisk;
- 3) сборка Asterisk;
- 4) конфигурация Asterisk;
- 5) настройка базы данных Asterisk;
- 6) запуск и проверка работы сервера Asterisk.

4.1 Подготовка сервера

Отключение SELinux: `nano /etc/sysconfig/selinux`

Изменить: `SELINUX=disabled`

Выполнить рестарт: `shutdown -r now`

Настройка таймзоны: `timedatectl set-timezone Asia/Novosibirsk`

Отключение firewalld: `systemctl disable firewalld`

`systemctl stop firewalld`

4.2 Установка пакетов, необходимых для сборки Asterisk

Выполнить команды:

`yum clean all`

`yum upgrade`

`yum install make gcc gcc-c++ pkgconfig libedit-devel jansson-devel libuuid-devel sqlite-devel libxml2-devel speex-devel libogg-devel libvorbis-devel alsa-lib-devel portaudio-devel libcurl-devel xmlstarlet bison flex postgresql-devel unixODBC-devel neon-devel gmime-devel lua-devel uriparser-devel libxslt-devel openssl-devel mariadb-devel bluez-libs-devel radcli-devel freetds-devel jack-audio-connection-kit-devel net-snmp-devel corosynclib-devel newt-devel popt-devel libical-devel spandsp-devel uw-imap-devel binutils-devel libsrtp-devel gsm-devel doxygen graphviz zlib-devel openldap-devel patch -y`

Установить ODBC для postgresql14:

`yum -y install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm`

`yum update`

`yum install postgresql14-odbc.x86_64`

4.3 Сборка Asterisk

Копирование конфигурационных файлов Asterisk:

```
cd /usr/src
```

Скачать исходники Asterisk:

```
wget -c http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-18.13.0.tar.gz
```

```
tar -xf asterisk-18.13.0.tar.gz
```

```
mv asterisk-18.13.0 asterisk
```

```
cd asterisk
```

```
./configure --with-jansson-bundled
```

Выбрать в меню `core_sounds_package` -> `en-wav`, `ru-wav`

```
make
```

```
make install
```

```
make config
```

Создание пользователя Asterisk:

```
adduser asterisk -c "Asterisk User"
```

```
passwd asterisk
```

```
usermod -aG wheel asterisk
```

```
chown asterisk:asterisk -R /var/run/asterisk
```

```
chown asterisk:asterisk -R /etc/asterisk
```

```
chown asterisk:asterisk -R /var/{lib,log,spool}/asterisk
```

```
chown asterisk:asterisk -R /var/cache/asterisk
```

```
chown asterisk:asterisk -R /var/run/asterisk
```

```
systemctl enable asterisk
```

4.4 Конфигурация Asterisk

Копирование конфигурационных файлов Asterisk:

```
cd /usr/src
```

Скопировать конфигурационные файлы Asterisk:

```
cp -Rf /usr/src/ropo_ucc/asterisk/config/* /etc/asterisk
```

Скопировать конфигурационные файлы драйвера ODBC:

```
cp -f /usr/src/ropo_ucc/asterisk/odbc/odbc.ini > /etc/odbc.ini
```

```
cp -f /usr/src/ropo_ucc/asterisk/odbc/odbcinst.ini > /etc/odbcinst.ini
```

4.5 Настройка базы данных Asterisk

Отредактировать файл `/etc/odbc.ini`:

```
Servername = ip_docker_host
```

Создать таблицы базы данных Asterisk:

```
pg_restore -h ip_docker_host -p 5432 -d asterisk -U asterisk -f  
/usr/src/ropo_ucc/asterisk/database/asterisk.sql
```

4.6 Запуск и проверка работы сервера Asterisk

Выполнить запуск процесса Asterisk: `systemctl start asterisk`

Проверить подключение Asterisk к базе данных postgresql:

```
asterisk -rx "odbc show" | grep «Number of active connections»
```

Результат выполнения команды должен быть:

Number of active connections: 1 (out of 1) – данный результат означает, что подключение к базе данных выполнено успешно.

Выполнить команду: `asterisk -rx "pjsip show endpoints"`

Результатом выполнения команды должен быть список эндпоинтов 0000-0100.

5 ОПИСАНИЕ ДЕЙСТВИЙ ПО ПРОВЕРКЕ КОРРЕКТНОСТИ УСТАНОВКИ КОМПОНЕНТОВ И ИХ НАСТРОЙКЕ

Проверка работоспособности установленного программного обеспечения проводится через интерфейс пользователя.

Для организации доступа к программному обеспечению «Digit PROконтакт» на АРМ пользователя выполните следующие действия:

- 1) проверьте, что в целях обеспечения возможности получения авторизованного доступа рабочее место сотрудника имеет подключение к КСПД или подключение к сети Интернет;
- 2) проверьте, что на АРМ пользователя установлен веб-браузер Google Chrome или Яндекс.Браузер актуальной версии.
- 3) в адресной строке веб-браузера введите адрес приложения *http://hostName/*.