смартСПОРТ

АИС «СМАРТСПОРТ 2:0»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

СОДЕРЖАНИЕ

1 Наименование Системы 4
2 Назначение Системы 4
3 Сведения о системном и прикладном программном обеспечении 4
4 Требования к квалификации администратора 4
5 Подготовка серверов к развертыванию 5
5.1 Общие требования 5
5.2 Установка системных утилит и вспомогательных пакетов 5
5.3 Синхронизация времени 6
6 Установка АИС «СмартСпорт 2:0» 7
6.1 Сервера файлового хранилища 7
6.2 Сервер баз данных 8
6.3 Сервера приложений 9
6.3.1 Установка OpenJDK 9
6.3.2 Установка КриптоПро 9
6.3.3 Установка Apache Tomcat 10
6.3.4 Установка RabbitMQ 11
6.3.5 Установка nginx 13
7 Установка ППО 14
7.1 Сервера баз данных 14
7.2 Сервера приложений 14
7.3 Сервера файлового хранилища 17
8 Установка обновления 18
9 Резервное копирование сервера баз данных 19
10 Аварийное восстановление 21
10.1 При скачивании файлов возникает ошибка 21
10.2 Не работает авторизация через ЕСИА 21

Перечень сокращений

В настоящем документе использованы следующие сокращения (см. Таблицу 1).

Таблица 1 – Список используемых сокращений

Сокращение	Расшифровка
БД	База данных
ЕСИА	Единая система идентификации и аутентификации
АИС «СмартСпорт 2:0», Система	Автоматизированная информационная система «СмартСпорт 2:0»
ОС	Операционная система
ППО	Прикладное программное обеспечение
СУБД	Система управления базами данных

1 Наименование Системы

Полное наименование: Автоматизированная информационная система «СмартСпорт 2:0»

Страна происхождения АИС «СмартСпорт 2:0» - Российская Федерация.

Сокращенное наименование: АИС «СмартСпорт 2:0», Система.

2 Назначение Системы

АИС «СмартСпорт 2:0» предназначена для автоматизации деятельности организаций в сфере спорта (РОИВ, Спортивные федераций, клубы, школы), а также автоматизации управления процессами спортивной подготовки.

3 Сведения о системном и прикладном программном обеспечении

АИС «СмартСпорт 2:0» разработана в трехзвенной архитектуре «клиент – сервер» и представляет собой совокупность следующих компонент:

- сервер баз данных – включает в себя данные и систему управления над ними (СУБД),
- сервер приложения – отвечает за бизнес-логику Системы,
- клиентское приложение - «тонкий» клиент в виде web-браузера на рабочих местах пользователей.

Программное обеспечение, применяемое на серверах Системы:

- Операционная система - Debian 10/11;
- Система управления базами данных - Postgres;
- Объектное хранилище – Minio;
- Прикладное ПО - Apache Tomcat, RabbitMQ, OpenJDK, КриптоПро, nginx.

4 Требования к квалификации администратора

Администратор Системы отвечает за поддержание работоспособности программной инфраструктуры Системы.

Для выполнения инструкций, изложенных в настоящем документе, администратор должен обладать квалификацией не ниже оператора обслуживания Системы:

- обладать базовыми знаниями информационных технологий:
    - базовые знания (представление) о стеке протоколов TCP/IP и базовых утилит (http, https, DNS, route, ping, netstat, ifconfig);
    - представление об устройстве компьютера и ОС (RAM, HDD, CPU, Core, файл, директория, файловая система, таблица разделов HD, volume manager);
    - представление о SSL, TLS, открытый/закрытый ключ, сертификаты;

- представление о работе firewall (NAT, proxy, port forwarding);

— иметь базовые навыки работы с Linux:

- работа в консоли Linux (bash, текстовые редакторы, cp, mv, grep, cat, tar, gzip, less, ssh, scp, kill);
- представление о process/thread, файловых дескрипторах, перенаправление, pipe, код возврата, сигналы;
- представление по управлению памятью в Linux (swap/paging/IPC);
- текстовые средства мониторинга sar/vmstat/top/iostat/ps /df;
- представление о работе/настройка сети в Linux (nmtui, ifconfig, ip addr, nmcli);
- базовые навыки управление системой (reboot, shutdown, systemctl, mount/umount, ssm);

— иметь базовые навыки для управления серверами приложений Apache Tomcat:

- запуск/остановка/проверка запуска серверов приложений;

— иметь базовые навыки для управления серверами балансировки нагрузки nginx:

- запуск/остановка/проверка запуска серверов балансировки нагрузки.

— иметь базовые навыки PostgreSQL:

- знакомство с утилитами (pg_ctl, psql, pg_basebackup);
- знакомство с SQL (простейшие SQL команды);
- назначение WAL сегментов;
- базовые команды и общее понимание работы HA pacemaker/corosync (pcs,crm_mon).

5 Подготовка серверов к развертыванию

5.1 Общие требования

Установка программного обеспечения проводится под пользователем root в операционной системе Debian 10.

5.2 Установка системных утилит и вспомогательных пакетов

Выполните от имени суперпользователя (root) следующие команды:

```
apt install wget mc nano curl mlocate gnupg apt-transport-
https gnupg2 ca-certificates lsb-release debian-archive-
keyring -y
```

5.3 Синхронизация времени

Для корректной работы системы, время на всех серверах должно быть синхронизировано. Для синхронизации времени в Debian используется сервис chrony.

Выполните установку пакета chrony:

```
apt install chrony
```

Для работы chrony необходимо внести изменения в файл настроек. Для этого в конце файла /etc/chrony/chrony.conf добавьте строки формата:

```
pool server 123.123.123.123
```

Если нет серверов точного времени, можно использовать международный сервера:

```
pool 2.debian.pool.ntp.org iburst
```

После установки и настойки запустите сервис и активируйте автоматический запуск:

```
systemctl start chrony
systemctl enable chrony
```

6 Установка АИС «СмартСпорт 2:0»

6.1 Сервера файлового хранилища

Установку minio произвести путем запуска бинарного файла, который необходимо скачать, выполнив команду:

```
wget https://dl.minio.io/server/minio/release/linux-amd64/minio
```

Сделать файл исполняемым:

```
chmod +x minio
```

Переместить исполняемый файл в каталог /usr/local/bin:

```
mv minio /usr/local/bin
```

В целях безопасности не рекомендуется запускать сервер minio от суперпользователя root. Необходимо создать пользователя и группу, от которого будем запускать minio:

```
useradd -r minio-user -s /sbin/nologin
```

Сменить владельца бинарного файла

```
chown minio-user:minio-user /usr/local/bin/minio
```

Далее необходимо создать каталог, в котором Minio будет хранить файлы

```
mkdir /var/share/minio -p
```

Передать права на каталог пользователю minio:

```
chown minio-user:minio-user /var/share/minio
```

Создать файл конфигурации minio

```
mkdir /etc/minio
chown minio-user:minio-user /etc/minio
nano /etc/default/minio
```

Настроить переменное окружение в файле /etc/default/minio

```
MINIO_ACCESS_KEY=”ACCESSKEY”
MINIO_SECRET_KEY=”SECRETKEY”
MINIO_VOLUMES=”/var/share/minio/”
MINIO_OPTS=”-C /etc/minio -address :9000 -console-address :9001”
```

Установка загрузочного скрипта Minio Systemd

```
curl -O https://raw.githubusercontent.com/minio/minio-service/master/linux-systemd/minio.service
mv minio.service /etc/systemd/system
systemctl daemon-reload
systemctl enable minio
```

Запустить сервер Minio:

```
systemctl start minio
```

Проверить статус:

```
systemctl status minio
```

Войти в веб-интерфейс minio можно на порту 9000, консоль администратора будет доступна на порту 9001.

6.2 Сервер баз данных

Подключите репозитарий:

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

Добавьте публичный ключ в доверие:

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

Обновите список пакетов:

```
apt-get update
```

Выполните:

```
apt install postgresql-11 -y
```

Отредактируйте файл /etc/postgresql/11/main/postgresql.conf – разрешите слушать любой IP:

```
listen_addresses = '*'
```

Так же увеличьте количество подключений до 1000:

```
max_connections = 1000
```

Отредактируйте файл /etc/postgresql/11/main/pg_hba.conf – добавьте в конец:

```
host    all             all             all                     md5
```

Произведите запуск сервиса и включите автоматический запуск:

```
systemctl enable postgresql
systemctl restart postgresql
```

Установите пароль на суперпользователя PostgreSQL:

```
passwd postgres
su - postgres
psql
ALTER USER postgres WITH PASSWORD ‘postgres’;
\q
exit
```

6.3 Сервера приложений

6.3.1 Установка OpenJDK

Скачайте и распакуйте актуальный дистрибутив OpenJDK (актуальную ссылку можно получить на сайт https://github.com/adoptium/temurin11-binaries/releases/ или https://adoptium.net/):

```
wget https://github.com/adoptium/temurin11-
binaries/releases/download/jdk-11.0.17%2B8/OpenJDK11U-
jdk_x64_linux_hotspot_11.0.17_8.tar.gz
mkdir /opt/jdk
tar -zxf OpenJDK11U-jdk_x64_linux_hotspot_11.0.17_8.tar.gz --directory
/opt/jdk  --strip-components=1
```

Подключите OpenJDK, задав переменные в конце файла /etc/profile и /root/.bashrc:

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

Изменения вступят в силу после перезагрузки. Для проверки успешности установки выполните следующую команду:

```
java -version
```

6.3.2 Установка КриптоПро

Важно, перед установкой КриптоПро убедитесь, что у Вас работает X11 сервер и клиент.

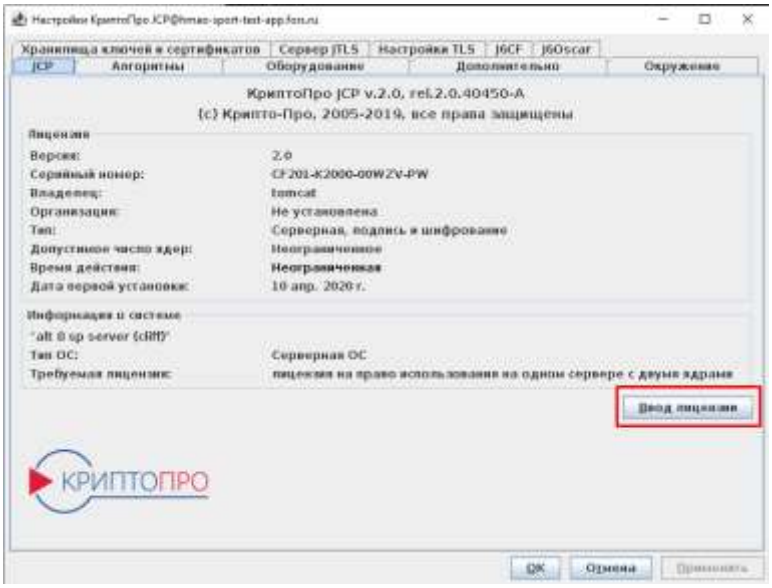
Дистрибутив КриптоПро JCP 2.0.40450-A необходимо скачать с сайта https://www.cryptopro.ru/products/csp/jcp/downloads, распаковать и загрузить на сервер в папку /opt/jcp. После чего выполните команды:

```
chmod 777 /opt/jcp/*.sh
sh /opt/jcp/configure.sh
```

При первом запуске будет сгенерирована тестовая лицензия сроком на 3 месяца. Запускать КриптоПро необходимо под пользователем tomcat:

```
./ControlPane.sh /opt/jdk
```

Для установки постоянной лицензии, в открывшемся окне нажмите «Ввод лицензии» и заполните данные лицензии в новом окне:



6.3.3 Установка Apache Tomcat

Создайте пользователя tomcat:

```
groupadd tomcat
useradd -M -s /bin/nologin -g tomcat -d /opt/tomcat tomcat
mkdir /home/tomcat/
chown -R tomcat:tomcat /home/tomcat
```

Скачайте и распакуйте Apache Tomcat (Актуальную ссылку можно получить на сайте https://tomcat.apache.org/):

```
wget https://dlcdn.apache.org/tomcat/tomcat-
9/v9.0.70/bin/apache-tomcat-9.0.70.tar.gz
mkdir /opt/tomcat
tar xvf apache-tomcat-9.0.70.tar.gz --directory /opt/tomcat
--strip-components=1
chown –R tomcat:tomcat /opt/tomcat/
```

Создайте файл /etc/systemd/system/tomcat.service со следующим содержимым:

```
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target

[Service]
Type=forking

Environment=JAVA_HOME=/opt/jdk
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
```

```
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -
XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -
Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID

User=tomcat
Group=tomcat
Umask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

В файл /opt/tomcat/conf/tomcat-users.xml перед </tomcat-users> добавьте пользователя:

```
<user username="tomcat" password="tomcat" roles="admin,admin-
gui,manager-gui,manager-script,manager-jmx,manager-status"/>
```

В файлах /opt/tomcat/webapps/host-manager/META-INF/context.xml и /opt/tomcat/webapps/manager/META-INF/context.xml экранируйте строку проверки IP адреса, добавив <!–в начале и → в конце:

```
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow=»127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1» />→
```

Обновите конфигурацию сервисов, запустите и включите автозапуск Apache Tomcat

```
systemctl daemon-reload
systemctl start tomcat
systemctl enable tomcat
```

Для проверки запуска Apache Tomcat используйте команду:

```
systemctl status tomcat
```

Так же сервер должен быть доступен в Web браузере по 8080 порту.

6.3.4 Установка RabbitMQ

Подключите репозитории:

```
curl -1sLf "https://keys.openpgp.org/vks/v1/by-
fingerprint/0A9AF2115F4687BD29803A206B73A36E6026DFCA" | gpg --dearmor
| tee /usr/share/keyrings/com.rabbitmq.team.gpg > /dev/null
curl -1sLf
"https://keyserver.ubuntu.com/pks/lookup?op=get&search=0xf77f1eda57ebb
1cc" | gpg --dearmor | tee
/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg > /dev/null
curl -1sLf "https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey" |
gpg --dearmor | tee /usr/share/keyrings/io.packagecloud.rabbitmq.gpg >
/dev/null

tee /etc/apt/sources.list.d/rabbitmq.list <<EOF
## Provides modern Erlang/OTP releases
##
## "bionic" as distribution name should work for any reasonably recent
Ubuntu or Debian release.
## See the release to distribution mapping table in RabbitMQ doc
guides to learn more.
deb [signed-
by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg]
http://ppa.launchpad.net/rabbitmq/rabbitmq-erlang/ubuntu bionic main
deb-src [signed-
by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg]
http://ppa.launchpad.net/rabbitmq/rabbitmq-erlang/ubuntu bionic main

## Provides RabbitMQ
##
## "bionic" as distribution name should work for any reasonably recent
Ubuntu or Debian release.
## See the release to distribution mapping table in RabbitMQ doc
guides to learn more.
deb [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg]
https://packagecloud.io/rabbitmq/rabbitmq-server/ubuntu/ bionic main
deb-src [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg]
https://packagecloud.io/rabbitmq/rabbitmq-server/ubuntu/ bionic main
EOF
```

Обновите список репозиториев

```
apt-get update -y
```

Установите пакет RabbitMQ, запустите сервис и включите автоматический запуск:

```
apt-get install rabbitmq-server
systemctl start rabbitmq-server
systemctl enable rabbitmq-server
```

Установите пакет Web-панели администрирования RabbitMQ:

```
rabbitmq-plugins enable rabbitmq_management
```

Для проверки работы RabbitMQ используйте команду:

```
systemctl status rabbitmq-server
```

Так же в браузере по порту 15672 должна быть доступна Web-панель управления RabbitMQ

При использовании более 1 сервера приложений, как того требует архитектура, необходимо настроить кластеризацию сервиса RabbitMQ.

Создайте политику синхронизации на основном сервере:

```
rabbitmqctl set_policy ha-all "" '{"ha-mode":"all","ha-sync-mode":"automatic"}'
```

Убедитесь, что политика создалась:

```
rabbitmqctl list_policies
```

Скопируйте файл /var/lib/rabbitmq/.erlang.cookie с основного сервера на все остальные

На остальных серверах остановите сервис:

```
systemctl restart rabbitmq-server
rabbitmqctl stop_app
```

Укажите серверам адрес основного сервера (вместо app01 – хост основного сервера):

```
rabbitmqctl join_cluster rabbit@app01
```

Снова запустите сервис:

```
rabbitmqctl start_app
```

6.3.5 Установка nginx

Скачайте ключ:

```
curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor \
    | tee /usr/share/keyrings/nginx-archive-keyring.gpg >/dev/null
```

Для подключения apt-репозитория для стабильной версии nginx, выполните следующую команду:

```
echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] \
http://nginx.org/packages/debian `lsb_release -cs` nginx" \
    | tee /etc/apt/sources.list.d/nginx.list
```

Для использования пакетов из репозитария вместо распространяемых в дистрибутиве, настройте закрепление:

```
echo -e "Package: *\nPin: origin nginx.org\nPin: release o=nginx\nPin-
Priority: 900\n" \
    | tee /etc/apt/preferences.d/99nginx
```

Чтобы установить nginx, выполните следующие команды:

```
apt update
apt install nginx
```

7 Установка ППО

7.1 Сервера баз данных

Установка базы данных предполагается с рабочей станции под управлением операционной системы Windows с установленной утилитой psql (входит в состав pgAdmin). Установка так же возможна Linux, ниже приведен пример установки с сервера баз данных.

Для папок security и sport поочередно отредактируйте файл /install/define/define.sql – задайте имя будущих баз данных, отредактируйте файлы install.bat – укажите параметры подключения к базе данных.

Поочередно в папках sport и security запустите файл /install/install.bat

В случае установки с Linux объявите переменные:

```
export PGHOST=localhost
export PGPORT=5432
export PGDATABASE=postgres
export PGUSER=postgres
export PGPASSWORD=postgres
```

Поочередно в папках sport и security запустите:

```
psql <install.sql >install.log 2>&1
```

7.2 Сервера приложений

Скопируйте файлы конфигурации cas в папку /etc/cas/, Catalina в папку /opt/tomcat/conf/, скопируйте файлы context.xml, server.xml в папку /opt/tomcat/conf/, файл setenv.sh в папку /opt/tomcat/bin/ Во всех скопированных файлах скорректируйте ссылки на приложения. Во всех xml файлах скорректируйте настройки (имена серверов, параметры БД). Файл postgresql-42.2.8.jar необходимо скопировать в /opt/tomcat/lib/.

Скопируйте файлы sport.war, security.war и cas.war, файл public.war получается путем копирования smartsport.war. Файлы располагаются в папке /opt/tomcat/webapps/.

smartsports.war -> /opt/tomcat/webapps/sport.war

smartsport-security.war -> /opt/tomcat/webapps/security.war

cas.war -> /opt/tomcat/webapps/cas.war

smartsports.war -> /opt/tomcat/webapps/public.war

Скопируйте (при их наличии) контейнер КриптоПРО в папку /var/opt/cprocsp/keys/tomcat/. В файле /etc/cas/config/cas.properties скорректируйте название контейнера и пароль к нему.

Владельцем папок и всех файлов по путям /etc/cas/, /opt/tomcat/, /var/opt/cprocsp/keys/tomcat/ должен быть пользователь tomcat и группа tomcat.

```
chown -R tomcat:tomcat /etc/cas/
chown -R tomcat:tomcat /opt/tomcat/
chown -R tomcat:tomcat /var/opt/cprocsp/keys/tomcat/
```

Скопируйте SSL сертификат и закрытый ключ для открытого контура в папку /etc/ssl/ с именем sport.cer и sport.key. Для повышения безопасности, сгенерируйте в этой же папке файл dhparam:

```
openssl dhparam -out dhparams.pem 4096
```

В файле /etc/nginx/nginx.conf внесите следующую настройку:

```
client_max_body_size 55M;
```

В этом же файле для открытого контура скорректируйте секции server (вместо sport используйте доменное имя) для редиректа на HTTPS подключение:

```
server {
    listen  *:80;
    server_name sport;
    return 301 https://$host$request_uri;
    access_log  /var/log/nginx/access.log;
}
```

Там же добавьте блок настроек для HTTPS подключения для открытого контура:

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    ssl_certificate /etc/ssl/sport.cer;
    ssl_certificate_key /etc/ssl/sport.key;
    ssl_session_timeout 1d;
    ssl_session_cache shared:MozSSL:10m;
    ssl_session_tickets off;
    ssl_dhparam /etc/ssl/dhparams.pem;
    ssl_protocols TLSv1.2 TLSv1.3;
```

```
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;
    add_header Strict-Transport-Security “max-age=63072000” always;
    ssl_stapling on;
    ssl_stapling_verify on;
```

Для открытого контура в секции server добавьте:

```
location /public {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/public;
}
location /security {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/public;
}
       location /cas {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/cas;
}
```

Для закрытого контура добавьте:

```
location /sport {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/sport;
}
```

```
location /security {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/security;
}
location /cas {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/cas;
}
```

Перезапустите сервера tomcat:

```
systemctl restart tomcat
```

7.3 Сервера файлового хранилища

Авторизируйте в панели управления на порту 9001 и создайте новый бакет «sport». Других настроек не требуется.

8 Установка обновления

Установка версии сводится к двум операциям: обновлению базы данных и обновлению приложений. Перед проведением обновления необходимо остановить сервера приложений:

```
systemctl stop tomcat
```

Обновление базы данных производится с помощью maven и утилиты liquibase. Для этого необходимо установить на машину OpenJDK и maven, а так же обеспечить доступ к открытым репозиториям, либо скопировать кеш .m2 в папку пользователя. Для установки обновления выполните в папке sport:

```
mvn -Dfile.encoding=UTF-8 -Dliquibase.username=sport -
Dliquibase.password=sport -
Dliquibase.url=jdbc:postgresql://localhost:5432/sport  liqui
base:update
```

В папке security:

```
mvn -Dfile.encoding=UTF-8 -Dliquibase.username=security -
Dliquibase.password=security -
Dliquibase.url=jdbc:postgresql://localhost:5432/security liq
uibase:update
```

В командах вместо localhost укажите адрес сервера базы данных.

Замените файлы sport.war и security.war, файл public.war получается путем копирования sport.war. Файл cas.war обновляется при необходимости и поставляется не со всеми версиями. Файлы располагаются в папке /opt/tomcat/webapps/.

Запустите сервера приложений:

```
systemctl start tomcat
```

9 Резервное копирование сервера баз данных

Данный документ предлагает базовую настройку резервного копирования базы данных. Более подробно о других вариантах настроек можно узнать из руководства по pgbackrest.

```
apt install pgbackrest
```

Сделайте пользователя postgres владельцем файла конфигурации

```
chown postgres:postgres /etc/pgbackrest.conf
```

Подключите сетевой диск или создайте папку /mnt/backup/.

Отредактируйте файл конфигурации pgbackrest (расположение /etc/pgbackrest.conf). Вставьте следующие параметры:

```
[global]
repo1-path=/mnt/backup/11 --место расположение бэкапов
repo1-retention-full=2 --количество хранящихся полных
резервных копий
repo1-type=cifs --тип хранилища данных

[hms-delta]
pg1-path=/var/lib/postgresql/11 --директория базы данных
start-fast=y --параметр для быстрого чекпоинта

[global:archive-push]
compress-level=3 --степень сжатия бэкапов
```

Теперь можно проинициализировать раздел, выполнив следующую команду от имени пользователя postgres

```
pgbackrest --stanza=otus --log-level-console=info stanza-
create
```

Далее настройте архивацию wal-файлов в конфиге postgres (/var/lib/postgresql/11/postgresql.conf)

```
archive_mode = on
archive_command = 'pgbackrest --stanza=otus archive-push %p'
```

Перезапустите сервер postgres, от имени postgres выполняем

```
pg_ctl restart
```

Проверьте работу репозитария

```
pgbackrest --stanza=otus --log-level-console=info check
```

Ответ должен быть вида:

```
2020-06-22 16:05:38.391 P00   INFO: check command begin 2.26: --log-
level-console=info --pg1-path=/var/lib/postgresql/11 --repo1-
path=/mnt/backup/11 --repo1-type=cifs --stanza=hms-delta
2020-06-22 16:05:39.517 P00   INFO: WAL segment
000000010000024000000ED successfully archived to
'/mnt/backup/11/archive/hms-delta/11-
1/0000000100000240/000000010000024000000ED-
ea43ae56da6ae184cd93abdfd4d4f96f2a60e03a.gz'
2020-06-22 16:05:39.517 P00   INFO: check command end: completed
successfully (1127ms)
```

Сделайте полную резервную копию

```
pgbackrest --stanza=otus --type=full --log-level-console=info backup
```

Проверьте успешности бэкапа и информацию о бэкапах

```
pgbackrest info
```

Пример ответа:

```
stanza: hms-delta
    status: ok
    cipher: none

    db (current)
        wal archive min/max (11-1):
000000010000024000000B3/000000010000024000000F4

        full backup: 20200622-160812F
            timestamp start/stop: 2020-06-22 16:08:12 / 2020-06-22
17:53:40
            wal start/stop: 000000010000024000000EF /
000000010000024000000F4
            database size: 111.8GB, backup size: 111.8GB
            repository size: 22.9GB, repository backup size: 22.9GB
```

Добавьте задание по расписанию в crontab:

```
00 03 * * 0 pgbackrest --type=full --stanza=hms-delta backup --log-
level-stderr=info &>/tmp/backup_full.log
00 03 * * 1-6 pgbackrest --type=diff --stanza=hms_delta backup --log-
level-stderr=info &>/tmp/backup_diff.log
```

10 Аварийное восстановление

10.1 При скачивании файлов возникает ошибка

Наиболее вероятной причиной появления ошибок при попытке скачать файлы является остановка приложения minio на сервере файлового архива.

Проверьте работоспособность minio. Подключитесь к серверу файлового архива и выполните команду:

```
systemctl status minio
```

Если сервис не запущен – запустите его командой:

```
systemctl start minio
```

10.2 Не работает авторизация через ЕСИА

Наиболее вероятной причиной невозможности авторизаться через ЕСИА является истечение срока действия ключей, либо истечение срока лицензии на КриптоПро JCP.

Подключитесь под пользователем tomcat к серверу приложений (клиент должен поддерживать протокол X11), перейтите в папку /opt/jcp и выполните команду

```
./ControlPane.sh /u01/jdk-11.0.2
```

В появившемся окне проверьте срок лицензии. Если лицензия истекла – нажмите New license и укажите новый лицензионный ключ.

Перейдите во вкладку Keys and certificates stores, откройте вкладку HDImageStore и проверьте срок действия ключа.

Если ключи устарели, получите новые и скопируйте контейнер с ними в папку /var/opt/cprocsp/keys/tomcat/. Затем отредактируйте файл /etc/cas/config/cas.properties, укажите наименование контейнера и пароль к нему

```
crypto-pro.jcp.alias=name
crypto-pro.jcp.password=password
```

Далее необходимо перезапустить сервер приложения:

```
systemctl restart tomcat
```

Открытый ключ необходимо добавить в панели управления (кабинете) ЕСИА.