
Документация

Выпуск 5.0.1

Администрирование XSQUARE-PGHS 5

янв. 31, 2025

1	Общие сведения	1
1.1	Знакомство с PGHS	1
2	Архитектура и системные требования	2
2.1	Архитектура	2
2.2	Базовая архитектура	2
2.3	Архитектура для высоконагруженных систем	3
2.4	Среда исполнения	3
2.5	Системные требования	4
3	Быстрая установка	5
3.1	Быстрая установка на DEB-based ОС	5
3.2	Быстрая установка на RPM-based ОС	7
4	Расширенная установка	10
4.1	Настройка DEB-based ОС	11
4.2	Настройка RPM-based ОС	12
4.3	Установка и настройка PostgreSQL	12
4.4	Установка и настройка HTTP Server (NGINX/Apache2)	14
4.5	Установка и настройка NGINX	14
4.6	Установка и настройка Apache2	16
5	Конфигурация файлов	20
5.1	Файл конфигурации config.json	20
5.2	Файл конфигурации схем аутентификации auth_config.json	21
6	Эксплуатация	24
6.1	Сервер приложений	24
6.2	База данных	25

1.1 Знакомство с PGHS

XSQUARE-PGHS (**PGHS**) - это сервер приложений, который обеспечивает работу веб-приложений, разработанных в конструкторе XSQUARE-RAD (**XRAD**).

PGHS - является посредником между браузером клиента и базами данных и полностью обеспечивает функционирование разработанного веб-приложения, предоставляя готовые инструменты для управления бизнес-логикой и визуализацией при помощи внутреннего API.

PGHS работает на базе PostgreSQL и выполняет:

- обслуживание и балансировку HTTP запросов между веб-клиентом и базой данных;
- создание структурированной иерархии страниц приложения;
- создание динамических веб-страниц с унификацией обработки в различных браузерах;
- рендеринг веб-управляемых страниц;
- визуализацию данных в виде форм, отчетов, графиков и других веб-компонентов;
- аутентификацию и авторизацию с использованием популярных схем;
- поддержку пула соединений с базой данных.

Настоящая документация описывает:

- архитектуру;
- различные варианты установки;
- конфигурирование.

Архитектура и системные требования

2.1 Архитектура

Архитектура приложения зависит от назначения и подразделяется:

- Базовая
- Базовая плюс среда разработки (описывается в документации разработчика)
- Базовая для высоконагруженной среды

и состоит из четырех основных компонентов:

- HTTP Proxy Server
- PGHS - сервер приложений
- App DB - база данных приложения, в которой содержатся бизнес-данные или бизнес DB. Приложение может использовать множество разных App DB.
- XRAD DB - мета база данных приложения, в которой хранятся данные для построения самого приложения.

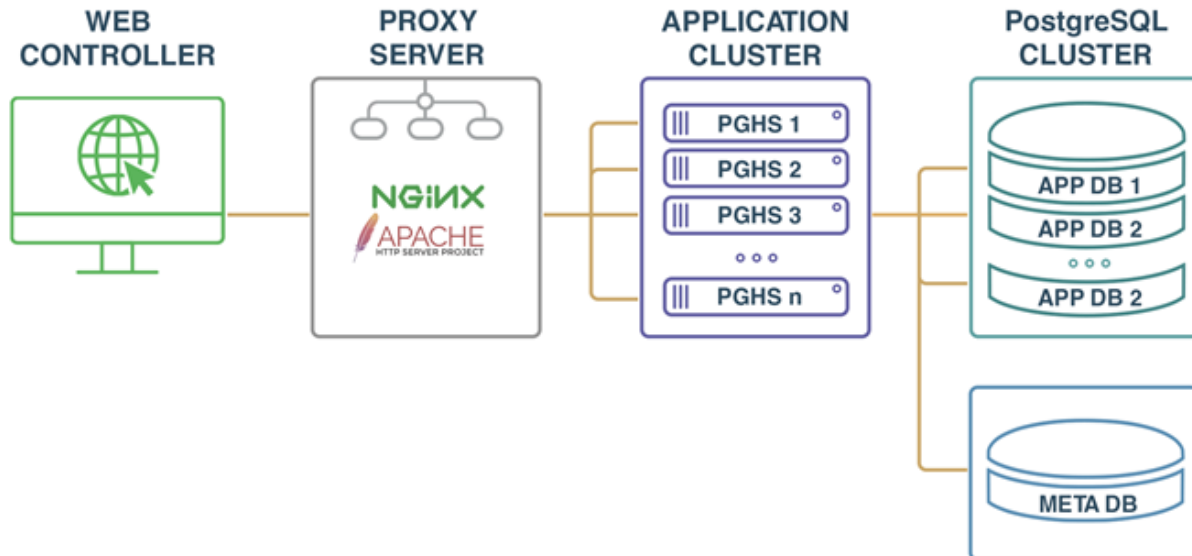
2.2 Базовая архитектура

Принципиальная или базовая архитектура приложения выглядит следующим образом для промышленного окружения:



2.3 Архитектура для высоконагруженных систем

Принципиальная или базовая архитектура приложения для высоконагруженных систем выглядит следующим образом:



2.4 Среда исполнения

Поддерживаемые архитектуры:

- x86-64
- ARM
- Loongson

Поддерживаемые ОС:

- **DEB**-based – любые
- **RPM**-based – любые
- Debian 12 - рекомендуемая

Базы данных:

- PostgreSQL 13+
- PostgreSQL 15 – рекомендуемая

HTTP/Proxy Server:

- Apache 2.4+
- NGINX 19+

2.5 Системные требования

Сервер приложений:

- CPU - 1 Ядро
- RAM - 100 Мб
- HDD - 100 Мб + Логи

XRAD DB:

- CPU - 1 Ядро
- RAM - 50 Мб
- HDD - 10 Мб БД PostgreSQL

App DB:

- CPU - 1 Ядро
- RAM - 50 Мб
- HDD - 10 Мб БД PostgreSQL (зависит от приложения)

Установка системы виртуализации/контейнеризации, операционной системы, базы данных осуществляется на усмотрение Администратора исходя из потребностей Организации.

3.1 Быстрая установка на DEB-based ОС

Рассмотрим быструю установку **PGHS** на примере ОС Debian. Подробное описание шагов установки можно посмотреть в разделе «Расширенная установка».

Все команды следует выполнять с правами суперпользователя (root)

1. Создаем каталог для дистрибутива PGHS

```
mkdir /root/xsquare
```

переходим в каталог

```
cd /root/xsquare
```

2. Скачиваем/получаем дистрибутив в созданный каталог

```
wget https://lcdp.xsquare.ru/files/pghs/xsquare.lcdp.v5/xsquare.lcdp.5.0_latest_
↪release.zip
```

3. Распаковываем дистрибутив

```
apt -y install unzip
unzip xsquare.lcdp.5.0.0.0.0.0_release.zip
```

4. Переходим в каталог с файлами дистрибутива PGHS

```
cd xsquare.lcdp.5.0.0.0.0.0_release
```

5. Настраиваем часовой пояс и локализацию ОС

```
echo "Europe/Moscow" > /etc/timezone && \
dpkg-reconfigure -f noninteractive tzdata && \
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sed -i -e 's/# en_US.UTF-8 UTF-8/en_US.UTF-8 UTF-8/' /etc/locale.gen && \
sed -i -e 's/# ru_RU.UTF-8 UTF-8/ru_RU.UTF-8 UTF-8/' /etc/locale.gen && \
echo 'LANG="ru_RU.UTF-8"'>/etc/default/locale && \
dpkg-reconfigure --frontend=noninteractive locales && \export LANG=ru_RU.UTF-8
```

6. Устанавливаем PostgreSQL

```
apt -y install postgresql
```

7. Подготавливаем PostgreSQL

переключаемся на пользователя postgres su - postgres

создаем пользователей БД xrad_user и app_user

```
psql -c "create user xrad_user with encrypted password 'xrad_user';"
psql -c "create user app_user with encrypted password 'app_user';"
```

создаем базы appdb и xraddb

```
psql -c "CREATE DATABASE \"appdb\" WITH OWNER \"app_user\" ENCODING 'UTF8' LC_COLLATE_
↪= 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
psql -c "CREATE DATABASE \"xraddb\" WITH OWNER \"xrad_user\" ENCODING 'UTF8' LC_
↪COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
```

назначаем пользователям xrad_user и app_user максимальные привилегии

```
psql -c "ALTER USER xrad_user WITH SUPERUSER;"
psql -c "ALTER USER app_user WITH SUPERUSER;"
```

выходим из сеанса учетной записи postgres

```
exit
```

8. Импортируем базы данных

```
export PGPASSWORD='xrad_user';
psql -U xrad_user -h 127.0.0.1 xraddb < db/xraddb.xsquare.pgsql
export PGPASSWORD='app_user';
psql -U app_user -h 127.0.0.1 appdb < db/appdb.xsquare.pgsql
```

9. Устанавливаем nginx

```
apt -y install nginx
```

отключаем сайт по умолчанию

```
rm -f /etc/nginx/sites-enabled/default
```

копируем из дистрибутива файлы веб-контроллера PGHS

```
cp -R ./var/www/pghs.xsquare* /var/www/
```

копируем из дистрибутива конфигурационные файлы nginx

```
cp -R ./etc/nginx /etc/
```

10. Перезапускаем nginx


```
systemctl restart nginx
```

проверяем его состояние

```
systemctl status nginx
systemctl enable nginx
```

11. Копируем исполняемые и конфигурационные файлы PGHS

```
cp -R ./etc/systemd /etc/
cp -R ./usr /
```

12. Запускаем PGHS как службу и проверяем статус

```
systemctl start xsquare.pghs.service
systemctl enable xsquare.pghs.service
systemctl status xsquare.pghs.service
```

13. Проверяем доступность дефолтного веб-приложения в браузере

Примечание: в случае проблем с доступом по http необходимо проверить настройки nginx и разрешения в брандмауэре.

3.2 Быстрая установка на RPM-based ОС

Рассмотрим быструю установку **PGHS** на примере ОС Fedora. Подробное описание шагов установки можно посмотреть в разделе «Расширенная установка». Все команды следует выполнять с правами суперпользователя (root).

1. Создаем каталог для дистрибутива PGHS

```
mkdir /root/xsquare
```

переходим в каталог

```
cd /root/xsquare
```

2. Скачиваем дистрибутив в созданный каталог

```
wget https://lcdp.xsquare.ru/files/pghs/xsquare.lcdp.v5/xsquare.lcdp.5.0.0.0.0.0_
↪release.zip
```

3. Распаковываем дистрибутив

```
dnf install -y unzip
unzip xsquare.lcdp.5.0.0.0.0.0_release.zip
```

4. Переходим в каталог с файлами дистрибутива PGHS

```
cd xsquare.lcdp.5.0.0.0.0.0_release
```

5. Настраиваем часовой пояс и локализацию ОС

```
timedatectl set-timezone Europe/Moscow
localectl set-locale LANG=ru_RU.UTF-8
export LANG=ru_RU.UTF-8
```

6. Устанавливаем и запускаем PostgreSQL

```
dnf install -y postgresql
postgresql-setup --initdb
systemctl start postgresql
systemctl enable postgresql
```

7. Подготавливаем PostgreSQL

переключаемся на пользователя postgres

```
su - postgres
```

создаем пользователей БД xrad_user и app_user

```
psql -c "create user xrad_user with encrypted password 'xrad_user';"
psql -c "create user app_user with encrypted password 'app_user';"
```

создаем базы appdb и xraddb

```
psql -c "CREATE DATABASE \"appdb\" WITH OWNER \"app_user\" ENCODING 'UTF8' LC_COLLATE_
↪= 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
psql -c "CREATE DATABASE \"xraddb\" WITH OWNER \"xrad_user\" ENCODING 'UTF8' LC_
↪COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
```

назначаем пользователям xrad_user и app_user максимальные привилегии

```
psql -c "ALTER USER xrad_user WITH SUPERUSER;"
psql -c "ALTER USER app_user WITH SUPERUSER;"
```

выходим из сеанса учетной записи postgres

```
exit
```

8. Импортируем базы данных

```
export PGPASSWORD='xrad_user';
psql -U xrad_user -h 127.0.0.1 xraddb < db/xraddb.xsquare.pgsql
export PGPASSWORD='app_user';
psql -U app_user -h 127.0.0.1 appdb < db/appdb.xsquare.pgsql
```

9. Устанавливаем nginx

```
dnf install -y nginx
```

отключаем сайт по умолчанию

```
rm -f /etc/nginx/sites-enabled/default
```

копируем из дистрибутива файлы веб-контроллера PGHS

```
cp -R ./var/www/pghs.xsquare* /var/www/
```

копируем из дистрибутива конфигурационные файлы nginx

```
cp -R ./etc/nginx /etc/
```

10. Отключаем Security-Enhanced Linux для HTTP запросов

```
setsebool -P httpd_can_network_connect 1
```

11. Перезапускаем nginx

```
systemctl restart nginx
```

проверяем его состояние

```
systemctl --no-pager status nginx
```

12. Копируем исполняемые и конфигурационные файлы PGHS

```
cp -R ./etc/systemd /etc/  
cp -R ./usr /
```

13. Запускаем PGHS как службу и проверяем статус

```
systemctl start xsquare.pghs.service  
systemctl enable xsquare.pghs.service  
systemctl --no-pager status xsquare.pghs.service
```

14. Проверяем доступность дефолтного веб-приложения в браузере

Примечание: в случае проблем с доступом по http необходимо проверить настройки nginx и разрешения в брандмауэре.

Расширенная установка

Рассмотрим все шаги по установке PGHS подробно. ВАЖНО: все команды следует выполнять с правами суперпользователя (root).

Далее в руководстве подразумевается, что дистрибутив XSQUARE-LCDP находится в каталоге /root/xsquare.

Для загрузки дистрибутива выполните следующие шаги:

1. Создаем каталог для дистрибутива

Создаем

```
mkdir /root/xsquare
```

переходим в каталог

```
cd /root/xsquare
```

2. Скачиваем дистрибутив в созданный каталог

```
:: wget https://lcdp.xsquare.ru/files/pghs/xsquare.lcdp.v5/xsquare.lcdp.5.0_latest_release.zip
```

3. Распаковываем дистрибутив

```
unzip xsquare.lcdp.5.0.0.0.0.0_release.zip
```

Примечание: для установки утилиты unzip выполните

```
apt -y install unzip (DEB-based ОС)  
dnf install -y unzip (RPM-based ОС)
```

4. Переходим в каталог с файлами дистрибутива PGHS

```
cd xsquare.lcdp.5.0.0.0.0.0_release
```

Для корректного функционирования разработанного веб-приложения необходимо сконфигурировать региональные настройки ОС (локаль) и часовой пояс.

4.1 Настройка DEB-based ОС

Для ручной настройки часового пояса необходимо запустить утилиту `dpkg-reconfigure` с параметром `tzdata`:

```
dpkg-reconfigure tzdata
```

Для ручной настройки локали необходимо запустить утилиту `dpkg-reconfigure` с параметром `locale`:

```
dpkg-reconfigure locale
```

Эти же действия можно автоматизировать: Записываем наименование необходимого часового пояса в файл `/etc/timezone`.

Например:

```
echo "Europe/Moscow" > /etc/timezone
```

применяем указанные настройки часового пояса:

```
dpkg-reconfigure -f noninteractive tzdata
```

Для настройки локали необходимо записать наименование локалей в файл `/etc/locale.gen`

Например для установки локалей `en_US.UTF-8` и `ru_RU.UTF-8` необходимо привести содержимое файла `/etc/locale.gen` к следующему виду: `en_US.UTF-8 UTF-8 ru_RU.UTF-8 UTF-8`

Так как чаще всего `/etc/locale.gen` уже содержит закомментированный список локалей, можно автоматизировать раскомментирование необходимых строк с помощью утилиты `sed`. Например:

```
sed -i -e 's/# en_US.UTF-8 UTF-8/en_US.UTF-8 UTF-8/' /etc/locale.gen  
sed -i -e 's/# ru_RU.UTF-8 UTF-8/ru_RU.UTF-8 UTF-8/' /etc/locale.gen
```

Для установки локали по умолчанию необходимо записать наименование локали в файл `/etc/default/locale`.

Например для установки по умолчанию локали `ru_RU.UTF-8`:

```
LANG=ru_RU.UTF-8
```

Установить локаль по умолчанию можно также с помощью утилиты `update-locale`.

Например:

```
update-locale LANG=ru_RU.UTF-8
```

После того, как файлы `/etc/locale.gen` и `/etc/default/locale` приведены к правильному содержанию – необходимо запустить автоматическое перекомментирование с помощью утилиты `dpkg-reconfigure`:

```
dpkg-reconfigure --frontend=noninteractive locales
```

Заключительным шагом подготовки ОС является запись переменной окружения `LANG` значением локали по умолчанию.

Например:

```
export LANG=ru_RU.UTF-8
```

4.2 Настройка RPM-based ОС

Для настройки часового пояса необходимо запустить утилиту `timedatectl` с параметром `timezone` и необходимым часовым поясом.

Например:

```
timedatectl set-timezone Europe/Moscow
```

Для получения текущего часового пояса:

```
timedatectl status
```

Для получения списка возможных часовых поясов:

```
timedatectl list-timezones
```

Для настройки локали необходимо запустить утилиту `localectl` с параметром `set-locale` и необходимой локалью:

```
localectl set-locale LANG=ru_RU.UTF-8
```

Для получения текущей локали:

```
localectl status
```

Для получения списка возможных локалей:

```
localectl list-locales
```

Также включить поддержку локали можно путем добавления строки `LANG=locale` в файл `/etc/sysconfig/i18n`.

Например:

```
echo "LANG=ru_RU.UTF-8" > /etc/sysconfig/i18n
```

Заключительным шагом подготовки ОС является запись переменной окружения `LANG` значением локали по умолчанию.

Например:

```
export LANG=ru_RU.UTF-8
```

4.3 Установка и настройка PostgreSQL

4.3.1 Установка PostgreSQL на DEB-based ОС

Для установки `postgresql` выполните:

```
apt install postgresql
```

Примечание: возможно перед установкой вам понадобится выполнить обновление системы

```
apt update  
apt upgrade
```

4.3.2 Установка PostgreSQL на RPM-based ОС

Для установки PostgreSQL выполните:

```
dnf install postgresql
```

Также необходимо запустить инициализацию базы данных:

```
postgresql-setup initdb
```

Примечание: возможно перед установкой вам понадобится выполнить обновление системы

```
dnf update
dnf upgrade
```

4.3.3 Настройка PostgreSQL

После установки PostgreSQL добавляем службу в автозагрузку и запускаем:

```
systemctl enable postgresql
systemctl start postgresql
```

Проверяем успешность запуска:

```
systemctl status postgresql

postgresql.service - PostgreSQL RDBMS
Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
Active: active (exited) since Wed 2024-10-09 09:05:17 CDT; 1 month 1 day ago
Process: 1052 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
Main PID: 1052 (code=exited, status=0/SUCCESS)
CPU: 5ms
```

Переключаемся на пользователя postgres

```
su - postgres
```

Создаем пользователей БД xrad_user и app_user

```
psql -c "create user xrad_user with encrypted password 'xrad_user';"
psql -c "create user app_user with encrypted password 'app_user';"
```

Создаем базы appdb и xraddb

```
psql -c "CREATE DATABASE \"appdb\" WITH OWNER \"app_user\" ENCODING 'UTF8' LC_COLLATE_
↪= 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
psql -c "CREATE DATABASE \"xraddb\" WITH OWNER \"xrad_user\" ENCODING 'UTF8' LC_
↪COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
```

Назначаем пользователям xrad_user и app_user максимальные привилегии

```
psql -c "ALTER USER xrad_user WITH SUPERUSER;"
psql -c "ALTER USER app_user WITH SUPERUSER;"
```

выходим из сеанса учетной записи postgres

```
exit
```

Импортируем базы данных

```
export PGPASSWORD='xrad_user';
psql -U xrad_user -h 127.0.0.1 xraddb < db/xraddb.xsquare.pgsq1
export PGPASSWORD='app_user';
psql -U app_user -h 127.0.0.1 appdb < db/appdb.xsquare.pgsq1
```

Примечание: может понадобиться настроить удаленное подключение и политику доступа PostgreSQL.

Для включения возможности удаленного подключения необходимо в файле `/etc/postgresql/[версия postgresql]/main/postgresql.conf` раскомментировать и отредактировать строку `listen_addresses`:

```
listen_addresses = '*'
```

Для настройки политики доступа необходимо в файл `/etc/postgresql/[версия postgresql]/main/pg_hba.conf` добавить строку:

```
host all all 0.0.0.0/0 md5
```

Данная строка позволит подключаться ко всем базам данных, всем пользователям с любым IP-адресом, с использованием MD5-хешированных паролей.

Для применения настроек необходимо перезапустить PostgreSQL:

```
systemctl restart postgresql
```

4.4 Установка и настройка HTTP Server (NGINX/Apache2)

Для работы PGHS необходимо установить HTTP-сервер. Рассмотрим установку HTTP-сервера на примере двух самых популярных решений NGINX и Apache2.

Примечание: для обеспечения доступности веб-приложения по имени необходимо добавить имя сервера в файл `/etc/hosts`. Например:

```
127.0.0.1 pghs.xsquare
```

4.5 Установка и настройка NGINX

Установка NGINX на DEB-based ОС:

```
apt install nginx
```

Установка NGINX на RPM-based ОС:

```
dnf install -y nginx
```


4.5.1 Настройка NGINX

Отключаем сайт по умолчанию

```
rm -f /etc/nginx/sites-enabled/default
```

копируем из дистрибутива файлы веб-контроллера PGHS

```
cp -R ./var/www/pghs.xsquare* /var/www/
```

копируем из дистрибутива конфигурационные файлы nginx

```
cp -R ./etc/nginx /etc/
```

и редактируем файл `/etc/nginx/conf.d/pghs.xsquare.conf`, внося необходимые изменения `vi /etc/nginx/conf.d/pghs.xsquare.conf`

```
server {
    listen 80;
    server_name pghs.xsquare;

    root /var/www/pghs.xsquare;
    index index.html;

    location /files{
        alias /var/www/pghs.xsquare.files.local;
        try_files $uri $uri/ =404;
    }

    location /pghs{
        proxy_pass http://127.0.0.1:8888/pghs;
    }

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Для RPM-based систем отключаем Security-Enhanced Linux для HTTP запросов

```
setsebool -P httpd_can_network_connect 1
```

Для применения новых настроек перезапускаем nginx

```
systemctl restart nginx
systemctl enable nginx
```

проверяем его состояние

```
systemctl --no-pager status nginx
```

4.6 Установка и настройка Apache2

Установка Apache2 на DEB-based ОС

```
apt-get install apache2
```

добавляем в автозагрузку и запускаем apache2

```
systemctl enable apache2
systemctl start apache2
```

проверяем состояние:

```
systemctl status apache2
```

устанавливаем дополнительные модули для apache2:

```
a2enmod proxy
a2enmod proxy_http
```

копируем из дистрибутива файлы веб-контроллера PGHS

```
cp -R ./var/www/pghs.xsquare* /var/www/
```

копируем из дистрибутива конфигурационные файлы apache2

```
cp -R ./etc/apache2 /etc/
```

и редактируем файл конфигурации VirtualHost /etc/apache2/sites-available/pghs.xsquare.conf, внося необходимые изменения: vi /etc/apache2/sites-available/pghs.xsquare.conf

```
<VirtualHost *:80>
    ServerAdmin info@xsquare.ru
    ServerName pghs.xsquare
    ServerAlias pghs.xsquare
    DocumentRoot /var/www/pghs.xsquare

    Alias /files "/var/www/pghs.xsquare.files.local"
    <Directory /var/www/pghs.xsquare.files.local>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    ProxyPass          /pghs http://127.0.0.1:8888/pghs
    ProxyPassReverse  /pghs http://127.0.0.1:8888/pghs

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Применяем новую конфигурацию

```
a2ensite pghs.xsquare.conf
```

и перезапускаем apache2

```
systemctl restart apache2
```

Установка Apache2 на RPM-based ОС

```
dnf install httpd
```

добавляем в автозагрузку и запускаем:

```
systemctl enable httpd
systemctl start httpd
```

проверяем состояние:

```
systemctl status httpd
```

копируем из дистрибутива файлы веб-контроллера PGHS

```
cp -R ./var/www/pghs.xsquare* /var/www/
```

копируем из дистрибутива конфигурационные файлы apache2

```
cp -R ./etc/httpd /etc/
```

и редактируем файл конфигурации VirtualHost `/etc/httpd/conf.d/pghs.xsquare.conf`, внося необходимые изменения: `vi /etc/httpd/conf.d/pghs.xsquare.conf`

```
<VirtualHost *:80>
    ServerAdmin info@xsquare.ru
    ServerName pghs.xsquare
    ServerAlias pghs.xsquare
    DocumentRoot /var/www/pghs.xsquare

    Alias /files "/var/www/pghs.xsquare.files.local"
    <Directory /var/www/pghs.xsquare.files.local>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    ProxyPass /pghs http://127.0.0.1:8888/pghs
    ProxyPassReverse /pghs http://127.0.0.1:8888/pghs

    ErrorLog /etc/httpd/logs/pghs-error.log
    CustomLog /etc/httpd/logs/pghs-access.log combined
</VirtualHost>
```

Отключаем Security-Enhanced Linux для HTTP запросов

```
setsebool -P httpd_can_network_connect 1
```

и перезапускаем apache

```
systemctl restart httpd
```

Установка и настройка PGHS

Для установки PGHS копируем исполняемые файлы из дистрибутива в каталог `/usr/local/xsquare.pghs`

Например:

```
cp -R ./usr /
```

Примечание: если необходимо назначьте права на исполнение `chmod +x /usr/local/xsquare.pghs/pghs`

Создаем сервис `vi /etc/systemd/system/xsquare.pghs.service`

```
[Unit]
Description=PGHS Services
After=syslog.target network.target
After=postgresql.service

[Service]
Type=simple
ExecStart=/usr/local/xsquare.pghs/pghs
WorkingDirectory=/usr/local/xsquare.pghs
Restart=on-failure
RestartSec=3

[Install]
WantedBy=default.target
```

Примечание: файл сервиса по умолчанию можно скопировать из дистрибутива.

Например:

```
cp -R ./etc/systemd /etc/
```

Запускаем PGHS как службу и проверяем статус `systemctl start xsquare.pghs.service systemctl enable xsquare.pghs.service systemctl --no-pager status xsquare.pghs.service`

Настраиваем конфигурацию PGHS для этого редактируем файл `config.json` в каталоге сервера приложений: `vi /usr/local/xsquare.pghs/config.json`

```
{
  "app": {
    "port": "8888"
  },
  "XRAD": {
    "login": "xrad_user",
    "password": "xrad_user",
    "host": "localhost",
    "port": 5432,
    "minCons": 1,
    "maxCons": 15,
    "dbName": "xraddb",
    "runtimeOptions": {
      "LC_NUMERIC": "ru_RU.UTF-8"
    }
  },
  "datasources": [
    {
      "login": "app_user",
      "password": "app_user",
      "host": "localhost",
      "name": "DEFAULT_APP",
      "port": 5432,
      "minCons": 1,
      "maxCons": 15,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"dbName": "appdb",
  "runtimeOptions": {
    "LC_NUMERIC": "ru_RU.UTF-8"
  }
}
]
```

Конфигурация файлов

5.1 Файл конфигурации config.json

Для работы сервера приложений PGHS необходимо, чтобы в директории сервера приложений присутствовал файл конфигурации config.json. Файл конфигурации содержит 3 раздела:

Описатель «app», где можно определить базовые настройки сервера:

```
{
  "app": {
    "port": "8888"
  },

```

«port» - строка. Определяет номер сетевого порта, на котором будет запущен сервер (по умолчанию - 8888)

Описатель «XRAD», где определяются настройки для работы с БД XRAD:

- «login» – строка. Имя пользователя для подключения к базе данных XRAD.
- «password» – строка. Пароль пользователя для подключения к базе данных XRAD.
- «host» – строка. IP-адрес сервера базы данных XRAD.
- «port» - число. Номер порта, на котором работает сервер базы данных XRAD.
- «dbName» – строка. Имя базы данных, к которой требуется подключиться XRAD.
- «minCons» – число. Минимальное количество одновременных соединений с базой данных.
- «maxCons» - число. Максимальное количество одновременных соединений с базой данных.

описатель runtimeOptions:

«LC_NUMERIC» – строка. Локальные настройки числового формата, используемого для работы с базой данных.

Описатель «datasources» определяет массив источников данных, используемых сервером приложений. Блок описания источника данных содержит тот же набор полей, что и описание БД XRAD и дополнительное поле:

- «name» - строка. Имя источника данных.

Например, в следующем блоке определяется два источника данных с именами DEFAULT_APP и DEFAULT_APP_TEST

```
"datasources": [
  {
    "login": "app_user",
    "password": "app_user",
    "host": "10.100.117.219",
    "name": "DEFAULT_APP",
    "port": 5432,
    "minCons": 1,
    "maxCons": 15,
    "dbName": "pghs",
    "runtimeOptions": {
      "LC_NUMERIC": "ru_RU.UTF-8"
    }
  },
  {
    "login": "app_user",
    "password": "app_user",
    "host": "10.100.117.219",
    "name": "DEFAULT_APP_TEST",
    "port": 5432,
    "minCons": 1,
    "maxCons": 15,
    "dbName": "pghs",
    "runtimeOptions": {
      "LC_NUMERIC": "ru_RU.UTF-8"
    }
  }
]
```

5.2 Файл конфигурации схем аутентификации auth_config.json

Сервер приложений PGHS поддерживает аутентификацию и авторизацию с использованием следующих схем:

- Microsoft LDAP
- Microsoft Kerberos SSO
- LDAP
- Kerberos SSO
- Open ID Connect

При разработке приложения в конструкторе XRAD разработчик определяет возможные схемы аутентификации посредством указания имени схемы и ее типа. При старте сервер приложений PGHS загружает схемы из файла auth_config.json и сопоставляет их по имени и типу с параметрами в БД XRAD.

Файл auth_config.json содержит массив описателей схем аутентификации следующего формата:

```
[
  {
    "name": "",
    "type": "",
    "options": {}
  }
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
}
]
```

- «name» - строка. Имя схемы аутентификации.
- «type» – строка. Тип схемы аутентификации.

Поддерживаемые значения:

- ldap
- kerberos_sso
- microsoft_ldap
- microsoft_kerberos_sso
- oidc
- «options» – блок параметров схемы, который зависит от типа схемы.

LDAP, MICROSOFT_LDAP

Для типа схемы ldap и microsoft_ldap блок «options» описывает параметры подключения к LDAP-серверу, которые используются для аутентификации и получения информации о пользователях.

- «host» – строка. IP-адрес LDAP-сервера.
- «port» – число. Порт, используемый для связи с LDAP-сервером.
- «base» – строка. Определяет базовую точку входа в LDAP-каталог.
- «encryption» – строка. Определяет тип шифрования, используемого для связи с LDAP-сервером. Поддерживаемые значения:
 - start_tls – устанавливает защищенное TLS-соединение после начальной аутентификации по незашифрованному каналу. Обычно используется на порту 389.
 - simple_tls – устанавливает полностью зашифрованное TLS-соединение с самого начала. Обычно используется на порту 636.
 - plain – не использует шифрование и работает по незащищенному каналу. Обычно используется на порту 389.
- «bind_dn» – строка. Distinguished Name (DN) пользователя, используемого для аутентификации на LDAP-сервере.
- «password» – строка. Пароль, связанный с указанным bind_dn.
- «request_user_groups» – логическое значение. Определяет необходимость запроса групп, к которым принадлежит пользователь.

Например:

```
{
  "name": "User Auth LDAP",
  "type": "ldap",
  "options": {
    "host": "10.100.117.229",
    "port": 389,
    "base": "DC=ald,DC=dom",
    "encryption": "start_tls",
    "bind_dn": "uid=ldap,cn=users,cn=accounts,dc=ald,dc=dom",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"password": "password",
"request_user_groups": true
}
}

```

KERBEROS_SSO, MICROSOFT_KERBEROS_SSO

Для типа схемы `kerberos_sso` и `microsoft_kerberos_sso` блок «options» описывает параметры подключения к LDAP-серверу, которые используются для аутентификации и получения информации о пользователях.

- «keytab» – строка. Закодированное в BASE64 содержимое keytab файла, сгенерированного для сквозной доменной аутентификации.
- «request_user_groups» – логическое значение. Определяет необходимость запроса групп, к которым принадлежит пользователь.

Например:

```

{
  "name": "User Auth Kerberos SSO",
  "type": "kerberos_sso",
  "options": {
    "keytab": "BQIAAABVAAIAB0FMRC5ET00ABEhUVFAAEXBnaHM0LWRldi5hbGQuZG9tAAAAAWa
    ...
    ukHskC0mwuIQB0AAAAAE=",
    "request_user_groups": true
  }
},

```

OIDC

Для типа схемы `oidc` (Open Id Connect) блок «options» описывает конфигурацию аутентификации и авторизации пользователей с использованием OpenID Connect (OIDC).

- «scope»- строка. Список запрашиваемых областей, к которым требуется доступ при аутентификации.
- «issuer»- строка. URL-адрес сервера аутентификации (Issuer).
- `uid_field` - строка. Определяет поле, содержащее уникальный идентификатор пользователя (UID).
- «pkce»- логическое значение. Флаг, указывающий на необходимость использования Proof Key for Code Exchange (PKCE) для повышения безопасности.
- **client_options** - описатель блока параметров клиента, использующего OIDC-аутентификацию.
 - «id»- строка. Идентификатор клиента(Client ID).
 - «secret» - строка. Секретный ключ клиента(Client Secret).
 - «redirect_uri»- строка. URL-адрес, на который будет перенаправлен пользователь после успешной аутентификации.
- «request_user_groups» – логическое значение. Определяет необходимость запроса групп, к которым принадлежит пользователь.

Данный раздел описывает порядок поддержания работоспособности приложения и порядок загрузки компонентов.

6.1 Сервер приложений

Для загрузки `pghs` пользователю необходимо убедиться в наличие правильно настроенного конфигурационного файла `config.json`.

```
cat /usr/local/xsquare.pghs/config.json
```

Команда должна отобразить правильный конфигурационный файл, описанный в разделе “Конфигурационные файлы”.

Запуск сервера приложений:

```
systemctl start xsquare.pghs.service
```

Проверка состояния сервера приложений:

```
systemctl status xsquare.pghs.service
```

В случае возникновения ошибок они будут записаны в журнал. Проверить сообщения об ошибках можно выполнив команду:

```
journalctl -u xsquare.pghs.service
```

6.2 База данных

Запуск базы данных

```
systemctl start postgresql
```

Остановка базы данных

```
systemctl stop postgresql
```

Проверка статуса сервера базы данных:

```
systemctl status postgresql
```

Необходимо проверить, что база сервер баз данных прослушивает указанный адрес и порт

```
root@pghsdb:main# cat postgresql.conf | grep listen
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
root@pghsdb:main# cat postgresql.conf | grep port
port = 5432
```

Необходимо проверить, что пользователи имеют доступ подключения по ipv4

```
root@pghsdb:main# cat pg_hba.conf
# IPv4 local connections:
host    xraddb          xrad_user    127.0.0.1/32          md5
host    app_db          app_user     127.0.0.1/32          md5
```

В случае возникновения неполадок с БД необходимо обратиться к руководству администратора выбранного дистрибутива PostgreSQL.