Documentation

Release 5.0.1

Administering XSQUARE-PGHS 5

Jan. 31, 2025

Table of Contents

CHAPTER 1

General information

1.1 Introduction to PGHS

XSQUARE-PGHS (**PGHS**) is an application server that powers web applications developed in the XSQUARE-RAD (**XRAD**) builder.

PGHS - mediates between the client's browser and databases and fully powers the developed web application by providing ready-made tools for managing business logic and visualization using an internal API.

PGHS is powered by PostgreSQL and performs:

- serving and balancing HTTP requests between the web client and the database,
- creating a structured hierarchy of application pages,
- creation of dynamic web pages with unified processing in different browsers,
- Web managed page rendering,
- data visualization in the form of forms, reports, graphs and other web components,
- authentication and authorization using popular schemes,
- database connection pool support.

This documentation describes:

- architecture,
- various installation options,
- configuration.

1.2 Localization

All our products do support all PostgreSQL locales. Very important at the beginning create your PostgreSQL database with compatible localization to correct order of sorting rows, for this the **LC_COLLATE** and **LC_CTYPE** parameters are responsible. We recommend setting your locale as primary on the operating system so that PostgreSQL inherits the OS locale by default.

Attention! During installation, the default English locale is specified, it is highlighted in red font in the text. If you want to set a different locale, change the highlighted value to the desired locale. Example:

```
echo "UTC" > /etc/timezone && ¥
```

CHAPTER 2

Architecture and system requirements

2.1 Architecture

The architecture of an application depends on its purpose and is subdivided:

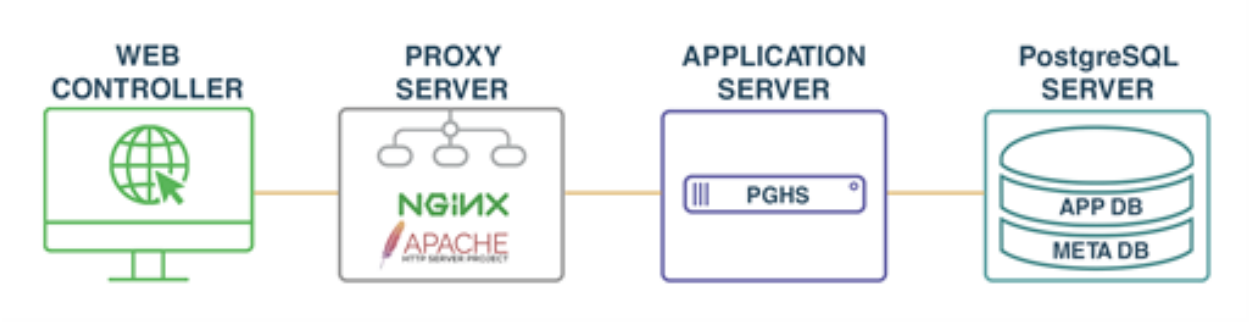
- Basic
- Basic plus development environment (described in the developer documentation)
- Basic for a highly loaded environment

and consists of four main components:

- HTTP Proxy Server
- PGHS - application server
- App DB - An application database that contains business data or business DB. An application can use many different App DBs
- XRAD DB is the application's meta database, which stores the data to build the application itself.

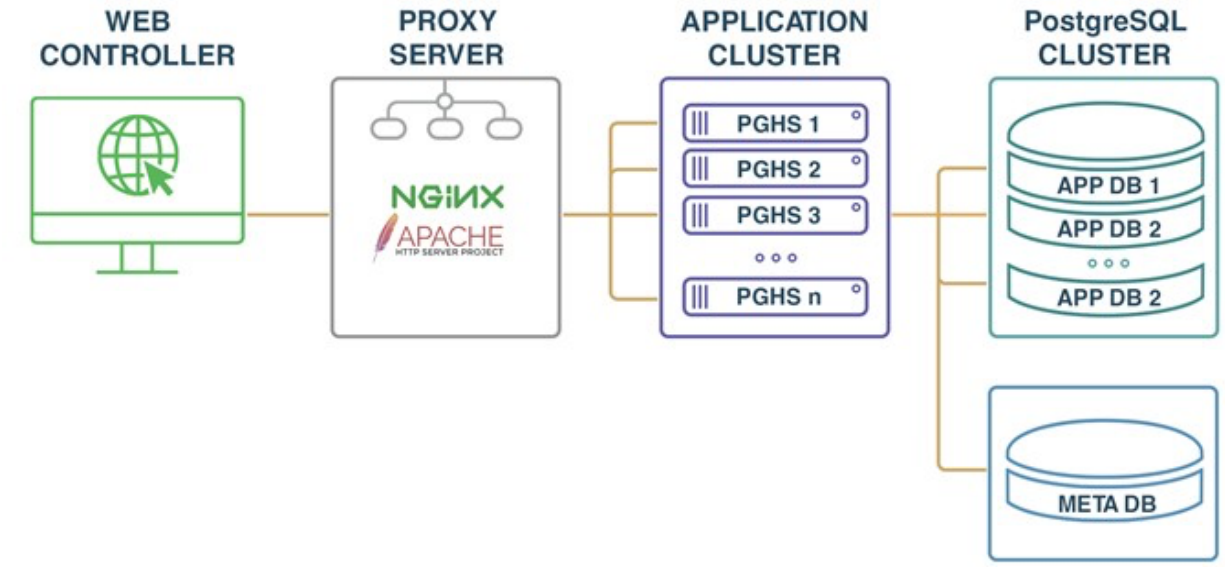
2.2 Basic architecture

The principle or basic architecture of the application is as follows for an industrial environment:



2.3 Architecture for highly loaded systems

The principle or basic architecture of an application for highly loaded systems is as follows:



2.4 Operating environment

Supported architecture:

- x86-64
- ARM
- Loongson

Supported OS:

- **DEB-based** - any
- **RPM-based** - any
- Debian 12 - recommended

Databases:

- PostgreSQL 13+
- PostgreSQL 15 - recommended HTTP/Proxy Server:
- Apache 2.4+
- NGINX 19+

2.5 System Requirements

Application Server:

- CPU - 1 Core
- RAM - 100 MB
- HDD - 100 MB + Logs

XRAD DB:

- CPU - 1 Core
- RAM - 50 MB
- HDD - 10 MB PostgreSQL

App DB:

- CPU - 1 Core
- RAM - 50 MB
- HDD - 10 MB PostgreSQL database (depends on the application)

Installation of virtualization/containerization system, operating system, database is at the discretion of the Administrator based on the needs of the Organization.

CHAPTER 3

Quick installation

3.1 Quick installation on a DEB-based OS

Below we review the quick installation of **PGHS** with Debian OS as an example. The detailed description of the installation steps can be found in the 'Advanced installation' section.

All commands should be run with root privileges

1. Create a directory for the PGHS distribution

```
mkdir /root/xsquare
```

Go to the directory

```
cd /root/xsquare
```

2. Download/retrieve the distribution to the created directory

```
wget https://lcdp.xsquare.dev/files/pghs/xsquare.lcdp.v5/xsquare.lcdp.5.0_latest_
→release.zip
```

3. Unzip the distribution

```
apt -y install unzip
unzip xsquare.lcdp.5.0.0.0.0.0.0.0_release.zip
```

4. Go to the directory with the PGHS distribution files

```
cd xsquare.lcdp.5.0.0.0.0.0.0.0_release
```

5. Configure the time zone and OS localization

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

(continues on next page)

(continued from previous page)

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

echo 'LANG="en_US.UTF-8"'>/etc/default/locale && \
dpkg-reconfigure --frontend=noninteractive locales && \
export LANG=en_US.UTF-8
```

5. Install PostgreSQL

```
apt -y install postgresql
```

6. Prepare PostgreSQL

- switch to user postgres su - postgres
- create database users xrad_user and app_user

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

- create appdb and xraddb databases

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

- assign maximum privileges to users xrad_user and app_user

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

- log out of the postgres account session

```
exit
```

7. Import databases

```
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
```

8. Install nginx

```
apt -y install nginx
```

- disable the default site

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

- copy the PGHS web controller files from the distribution kit

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

- copy nginx configuration files from the distribution kit

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

9. Restart nginx

```
systemctl restart nginx
```

- check its condition

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

11. Copy PGHS executable and configuration files

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

12. Start PGHS as a service and check the status

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

13. Check if the default web application is available in the browser

Note: In case of problems with accessing http, you need to check your inginx settings and firewall permissions.

3.2 Quick installation on an RPM-based OS

Below we review quick installation of **PGHS** using Fedora OS as an example. The detailed description of the installation steps can be found in the 'Advanced installation' section. All commands should be run with superuser privileges (root).

1. Create a directory for the PGHS distribution

```
mkdir /root/xsquare
```

Go to the directory

```
cd /root/xsquare
```

2. Download the distribution to the created directory

```
wget https://lcdp.xsquare.dev/files/pghs/xsquare.lcdp.v5/xsquare.lcdp.5.0.0.0.0.0_
→release.zip
```

3. Unzip the distribution

```
dnf install -y unzip
unzip xsquare.lcdp.5.0.0.0.0.0.0.0_release.zip
```

4. Go to the directory with the PGHS distribution files

```
cd xsquare.lcdp.5.0.0.0.0.0.0.0_release
```

5. Configure the time zone and OS localization

```
timedatectl set-timezone UTC
localectl set-locale LANG=en_US.UTF-8
export LANG=en_US.UTF-8
```

6. Install and run PostgreSQL

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

7. Prepare PostgreSQL switch to the postgres user

```
su - postgres
```

create database users xrad_user and app_user

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

create appdb and xraddb databases

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

assign maximum privileges to users xrad_user and app_user

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

log out of the postgres account session

```
 exit
```

8. Import databases

```
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. Install nginx

```
dnf install -y nginx
```

disable the default site

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

copy the PGHS web controller files from the distribution kit

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

copy nginx configuration files from the distribution kit

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

10. Disable Security-Enhanced Linux for HTTP requests

```
setsebool -P httpd_can_network_connect 1
```

11. Restart nginx

```
systemctl restart nginx
```

check its condition

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

12. Copy PGHS executable and configuration files

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

13. Start PGHS as a service and check the status

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

14. Check if the default web application is available in the browser

Note: In case of problems with accessing http, you need to check your inginx settings and firewall permissions.

CHAPTER 4

Advanced installation

Below we review the steps to install PGHS in detail. IMPORTANT: All commands must be run with root privileges.

The following manual assumes that the XSQUARE-LCDP distribution is located in the /root/xsquare directory.

Follow the steps below to download the distribution:

1. Create a directory for the distribution

```
mkdir /root/xsquarel
```

Go to the directory

```
cd /root/xsquare
```

2. Download the distribution to the created directory

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

3. Unzip the distribution

```
unzip xsquare.lcdp.5.0.0.0.0.0.0.0_release.zip
```

Note: To install the unzip utility, run

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

4. Go to the directory with the PGHS distribution files

```
cd xsquare.lcdp.5.0.0.0.0.0.0.0_release
```

For the correct operation of the developed web application, it is necessary to configure regional OS settings (locale) and time zone.

4.1 Configuring DEB-based OS

To manually configure the time zone, run the dpkg-reconfigure utility with the tzdata parameter:

```
dpkg-reconfigure tzdata
```

To manually configure the locale, run the dpkg-reconfigure utility with the locale parameter:

```
dpkg-reconfigure locale
```

The same actions can be automated: Write the name of the required time zone to the file /etc/timezone.

For example:

```
echo "UTC" > /etc/timezone
```

apply the specified time zone settings:

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

To configure the locale, you need to write the locale names in the /etc/locale.gen file

For example, to install en_US.UTF-8 and ru_RU.UTF-8 locales, the contents of the /etc/locale.gen file should be set to the following: en_US.UTF-8 UTF-8 UTF-8

Since most often /etc/locale.gen already contains a commented list of locales, you can automate the uncommenting of necessary lines using the sed utility. For example:

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

To set the default locale, you need to write the locale name to the /etc/default/locale file.

For example, to set the default locale to ru_RU.UTF-8:

```
LANG=en_US.UTF-8
```

You can also set the default locale using the update-locale utility.

For example:

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

After the /etc/locale.gen and /etc/default/locale files are brought to the correct contents - you should run the automatic reconfiguration using the dpkg-reconfigure utility:

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

The final step in preparing the OS is to write the LANG environment variable with the default locale value.

For example:

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

4.2 Configuring RPM-based OS

To set the time zone, run the timedatectl utility with the timezone parameter and the required time zone.

For example:

```
timedatectl set-timezone UTC
```

To get the current time zone:

```
timedatectl status
```

To get a list of possible time zones:

```
timedatectl list-timezones
```

To set the locale, run the localectl utility with the set-locale parameter and the required locale:

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

To get the current locale:

```
localectl status
```

To get a list of possible locales:

```
localectl list-locales
```

You can also enable locale support by adding the LANG line= locale to the /etc/sysconfig/i18n file.

For example:

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

The final step in preparing the OS is to write the LANG environment variable with the default locale value.

For example:

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

4.3 Installing and configuring PostgreSQL

4.3.1 Installing PostgreSQL on a DEB-based OS

To install PostgreSQL, run:

```
apt install postgresql
```

Note: You may need to perform a system update before installation

```
apt update
apt upgrade
```

4.3.2 Installing PostgreSQL on an RPM-based OS

To install PostgreSQL, run:

```
dnf install postgresql
```

You also need to run the database initialization:

```
postgresql-setup initdb
```

Note: You may need to perform a system update before installation

```
dnf update
dnf upgrade
```

4.3.3 Configuring PostgreSQL

After installing PostgreSQL, add the service to autoloader and start it:

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

Check if the startup was successful:

```
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
```

Switch to the postgres user

```
su - postgres
```

Create database users xrad_user and app_user

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

Create appdb and xraddb databases

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

Assign maximum privileges to users xrad_user and app_user

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

log out of the postgres account session

```
exit
```

Import databases

```
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
```

Note: You may need to configure the remote connection and PostgreSQL access policy.

To enable remote connectivity, you need to uncomment and edit the listen_addresses line in /etc/postgresql/[postgresql version]/main/postgresql.conf file:

```
listen_addresses= '*'
```

To configure the access policy, add a line to the /etc/postgresql/[postgresql version]/main/pg_hba.conf file:

```
host all all 0.0.0.0.0/0 md5
```

This line will allow connecting to all databases, all users with any IP address, using MD5 hashed passwords.

PostgreSQL must be restarted to apply the settings:

```
systemctl restart postgresql
```

4.4 Installing and configuring HTTP Server (NGINX/Apache2)

For PGHS to operate, an HTTP server must be installed. Find below the info on installation of HTTP server on the example of two most popular solutions NGINX and Apache2.

Note: To make the web application available by name, you must add the server name to the /etc/hosts file. For example:

```
127.0.0.1 pghs.xsquare
```

4.5 Installing and configuring NGINX

Installing NGINX on a DEB-based OS:

```
apt install nginx
```

Installing NGINX on an RPM-based OS:

```
dnf install -y nginx
```

4.5.1 Configuring NGINX

Disable the default site

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

copy the PGHS web controller files from the distribution kit

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

copy nginx configuration files from the distribution kit

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

and edit file /etc/nginx/conf.d/pghs.xsquare.conf, making necessary changes
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;
    }
}
```

For RPM-based systems, disable Security-Enhanced Linux for HTTP requests

```
setsebool -P httpd_can_network_connect 1
```

To apply the new settings, restart nginx

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

check its condition

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

4.6 Installing and Configuring Apache2

Installing Apache2 on a DEB-based OS

```
apt-get install apache2
```

add to autorun and start apache2

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

check the status:

```
systemctl status apache2
```

install additional modules for apache2:

```
a2enmod proxy
a2enmod proxy_http
```

copy the PGHS web controller files from the distribution kit

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

copy apache2 configuration files from the distribution kit

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

and edit the VirtualHost configuration file /etc/apache2/sites-available/pghs.xsquare.conf, making the necessary changes: vi /etc/apache2/sites-available/pghs.xsquare.conf

```
<VirtualHost *:80>
        ServerAdmin info@xsquare.dev
        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
                Demand 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>
```

Apply the new configuration

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

and restart apache2

```
systemctl restart apache2
```

Installing Apache2 on an RPM-based OS

```
dnf install httpd
```

add it to the autoloader and run it:

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

check the status:

```
systemctl status httpd
```

copy the PGHS web controller files from the distribution kit

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

copy apache2 configuration files from the distribution kit

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

and edit the VirtualHost configuration file /etc/httpd/conf.d/pghs.xsquare.conf, making the necessary changes: vi /etc/httpd/conf.d/pghs.xsquare.conf

```
<VirtualHost *:80>
       ServerAdmin     info@xsquare.dev
       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
               Demand 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>
```

Disable Security-Enhanced Linux for HTTP requests

```
setsebool -P httpd_can_network_connect 1
```

and restart apache

```
systemctl restart httpd
```

PGHS installation and configuration

To install PGHS, copy the executables from the distribution to the /usr/local/xsquare.pghs directory

For example:

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

Note: if necessary, assign execution rights to chmod +x /usr/local/xsquare.pghs/pghs

Create service 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
```

Note: the default service file can be copied from the distribution.

For example:

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

Start PGHS as a service and check the status systemctl start xsquare.pghs.service systemctl enable xsquare.pghs.service systemctl -no-pager status xsquare.pghs.service

Set up the PGHS configuration by editing the config.json file in the application server directory: 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": "en_US.UTF-8" }
    },
    "datasources": [
       {
        "login": "app_user",
        "password": "app_user",
        "host": { "localhost",
        "name": "DEFAULT_APP",
        "port": 5432,
        "minCons":1,
        "maxCons":15,
```

(continues on next page)

(continued from previous page)

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

CHAPTER 5

File configuration

5.1 Configuration file config.json

The PGHS application server requires a config.json configuration file to be present in the application server directory. The configuration file contains 3 sections:

The "app" descriptor, where you can define the basic server settings:

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

"port" - string. Defines the number of the network port on which the server will be started (by default - 8888)

Descriptor "XRAD", where the settings for working with the XRAD database are defined:

- "login" - string. Username to connect to the XRAD database.
- "password" - string. User password to connect to XRAD database.
- "host" - string. IP address of the XRAD database server.
- "port" - number. The number of the port on which the XRAD database server is running.
- "dbName" - string. The name of the database to which XRAD needs to connect.
- "minCons" - number. Minimum number of simultaneous connections to the database.
- "maxCons" - number. Maximum number of simultaneous connections to the database.

runtimeOptions descriptor:

"LC_NUMERIC" - string. Local settings of the numeric format used to work with the database.

The "datasources" descriptor defines an array of data sources used by the application server. The datasource description block contains the same set of fields as the XRAD database description and an additional field:

- "name" - string. The name of the data source.

For example, the next block defines two data sources with names DEFAULT_APP and 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":"en_US.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":"en_US.UTF-8"
        }
   }
   ]
```

5.2 Authentication scheme configuration file auth_config.json

The PGHS Application Server supports authentication and authorization using the following schemes:

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

When developing an application in the XRAD Designer, the developer defines the possible authentication schemes by specifying the scheme name and type. At startup, the PGHS application server loads schemes from the auth_config.json file and maps them by name and type to parameters in the XRAD database.

The auth_config.json file contains an array of authentication scheme descriptors in the following format:

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

(continues on next page)

(continued from previous page)

```
    }
]
```

- "name" - string. The name of the authentication scheme.
- "type" - string. Authentication scheme type.

Supported Values:

- ldap
- kerberos_sso
- microsoft_ldap
- microsoft_kerberos_sso
- oidc
- "options" - a block of scheme parameters that depends on the scheme type.

LDAP, MICROSOFT_LDAP

For the ldap and microsoft_ldap schema type, the "options" block describes the LDAP server connection parameters that are used to authenticate and retrieve user information.

- "host" - string. IP address of the LDAP server.
- "port" - number. The port used to connect to the LDAP server.
- "base" - string. Defines the base entry point to the LDAP directory.
- "encryption" - string. Specifies the type of encryption used to connect to the LDAP server. Supported values:
  - start_tls - establishes a secure TLS connection after initial authentication over an unencrypted channel. Usually used on port 389.
  - simple_tls - establishes a fully encrypted TLS connection from the start. Typically used on port 636.
  - plain - does not use encryption and works over an unprotected channel. Usually used on port 389.
- "bind_dn" - string. Distinguished Name (DN) of the user used for authentication on the LDAP server.
- "password" - string. The password associated with the specified bind_dn.
- "request_user_groups" - logical value. Determines the need to request the groups to which the user belongs.

For example:

```
{
"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)

(continued from previous page)

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

KERBEROS_SSO, MICROSOFT_KERBEROS_SSO

For the kerberos_sso and microsoft_kerberos_sso scheme types, the "options" block describes the LDAP server connection parameters that are used to authenticate and retrieve user information.

- "keytab" - string. BASE64 encoded content of keytab file generated for end-to-end domain authentication.
- "request_user_groups" - logical value. Determines the need to request the groups to which the user belongs.

For example:

```
{
     "name": "User Auth Kerberos SSO",
     "type": "kerberos_sso",
     "options": {
         "keytab": "BQIAAABVAAIAB0FMRC5ET00ABEhUVFAAEXBnaHM0LWRldi5hbGQuZG9tAAAAAAAAWa
       ...
       ukHskC0mwuIQB0AAAAAAAE= ",
         "request_user_groups": true
     }
 },
```

OIDC

For the oidc (Open Id Connect) scheme type, the "options" block describes the configuration of user authentication and authorization using OpenID Connect (OIDC).

- "scope"-string. A list of requested scopes that need to be accessed during authentication.
- "issuer"-string. The URL of the authentication server (Issuer).
- uid_field - string. It defines a field containing the unique user identifier (UID).
- "pkce" - boolean value. The flag indicates the need to use Proof Key for Code Exchange (PKCE) to enhance security.
- **client_options - descriptor of the block of parameters of the client using OIDC-authentication.**
  - "id"-string. Client ID.
  - "secret" - string. Client secret key (Client Secret).
  - "redirect_uri"-string.URL to which the user will be redirected after successful authentication.
- "request_user_groups" - boolean value. It determines the need to request the groups to which the user belongs.

CHAPTER 6

Operation

This section describes how to keep the application operational and the order of loading components.

6.1 Application Server

To download pghs, the user needs to make sure there is a properly configured config.json configuration file.

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

The command must display the correct configuration file described in the "Configuration files" section.

Application server startup:

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

Check the status of the application server:

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

If errors occur, they will be logged. One can check error messages by executing the command:

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

6.2 Database

Starting the database

```
systemctl start postgresql
```

Stopping the database

```
systemctl stop postgresql
```

Checking the status of the database server:

```
systemctl status postgresql
```

One needs to verify that the database server is listening to the specified address and port

```
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
```

One needs to verify that users have connection access by 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
```

In case of issues with the database, refer to the administrator's manual of the selected PostgreSQL distribution.