



Установка BPMSoft под PostgreSQL (Linux)



Оглавление

Настройка сервера БД.....	3
Настройка сервера Redis	7
Настройка сервера приложения.....	8
Настройка логирования	10

Настройка сервера БД

Подключение репозитория PostgreSQL

```
#репозиторий PostgreSQL
echo -e "deb http://apt.postgresql.org/pub/repos/apt/ \$(lsb_release -sc)-pgdg main" >
/etc/apt/sources.list.d/pgdg.list

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

Установка PostgreSQL 12

```
apt-get update
apt-get install -y postgresql-12 postgresql-client-common
```

Создание пользователей и БД

```
sudo su - postgres

psql

CREATE USER sa;
ALTER ROLE sa WITH SUPERUSER;
ALTER ROLE sa WITH LOGIN;
ALTER ROLE sa WITH PASSWORD '123';
CREATE USER someuser;
ALTER ROLE someuser WITH LOGIN;
ALTER ROLE someuser WITH PASSWORD '111';
CREATE DATABASE <ИМЯ_СОЗДАВАЕМОЙ_БД> WITH OWNER = someuser ENCODING = 'UTF8'
CONNECTION LIMIT = -1;

\q

exit
```

Добавить в /etc/postgresql/12/main/pg_hba.conf (/var/lib/pgpro/std-12/data/pg_hba.conf) строчку:

```
host all all 0.0.0/0 md5
```

Изменить в /etc/postgresql/12/main/postgresql.conf (/var/lib/pgpro/std-12/data/postgresql.conf) строчку:

```
listen_addresses = 'localhost' на listen_addresses = '*'
```

Перезапустить postgresql:

```
systemctl restart postgresql
# или
systemctl restart postgrespro-std-12
```

Выполнить преобразование типов при помощи скрипта **CreateTypeCastsPostgreSql.sql**:

```
CreateTypeCastsPostgreSql.sql
```

```
--Enabling implicit type conversions

DROP CAST IF EXISTS (varchar AS integer);
CREATE CAST (varchar AS integer) WITH INOUT AS IMPLICIT;

DROP CAST IF EXISTS (varchar AS uuid);
CREATE CAST (varchar AS uuid) WITH INOUT AS IMPLICIT;

DROP CAST IF EXISTS (text AS integer);
CREATE CAST (text AS integer) WITH INOUT AS IMPLICIT;

DROP CAST IF EXISTS (uuid AS text);
CREATE CAST (uuid AS text) WITH INOUT AS IMPLICIT;

DROP CAST IF EXISTS (text AS boolean);
CREATE CAST (text AS boolean) WITH INOUT AS IMPLICIT;

DROP CAST IF EXISTS (text AS numeric);
CREATE CAST (text AS numeric) WITH INOUT AS IMPLICIT;

DROP CAST IF EXISTS (text AS uuid);
CREATE CAST (text AS uuid) WITH INOUT AS IMPLICIT;

DROP FUNCTION IF EXISTS public."fn_CastTimeToDateTime" CASCADE;
CREATE FUNCTION public."fn_CastTimeToDateTime"(sourceValue TIME WITHOUT TIME ZONE)
RETURNS TIMESTAMP WITHOUT TIME ZONE
AS $BODY$
BEGIN
    RETURN (MAKE_DATE(1900, 01, 01) + sourceValue)::TIMESTAMP WITHOUT TIME ZONE;
END;
$BODY$
LANGUAGE PLPGSQL;

DROP CAST IF EXISTS (TIME WITHOUT TIME ZONE AS TIMESTAMP WITHOUT TIME ZONE);
CREATE CAST (TIME WITHOUT TIME ZONE AS TIMESTAMP WITHOUT TIME ZONE)
WITH FUNCTION public."fn_CastTimeToDateTime"(TIME WITHOUT TIME ZONE) AS IMPLICIT;
```

И ввести команды:

```
export PGPASSWORD=123
```

```
psql --host localhost --port 5432 --username=sa --dbname=<ИМЯ_БД> --
file=/path/to/CreateTypeCastsPostgreSql.sql
```

Восстановить БД из резервной копии, прилагаемой с ПО.

Для этого необходимо из архива с дистрибутивом скопировать папку db, а также скопировать скрипт **ChangeDbObjectsOwner_Postgres11.v2.sql** в любую папку на хосте.

```
ChangeDbObjectsOwner_Postgres11.v2.sql

SELECT set_config('var.newOwner', :'owner', FALSE);

do
$$
declare
    newOwner TEXT;
    rec record;
begin
    newOwner := current_setting('var.newOwner');
    execute 'ALTER TYPE varchar OWNER TO '||newOwner||'';
    execute 'ALTER TYPE bool OWNER TO '||newOwner||'';
    execute 'ALTER TYPE text OWNER TO '||newOwner||'';
    execute 'ALTER TYPE timestamp OWNER TO '||newOwner||'';
    --Schemas
    for rec in(select 'alter schema '||nspname||' owner to '||newOwner||'' as
query from pg_namespace)
    loop
        execute rec.query;
    end loop;
    --Tables
    for rec in(SELECT 'ALTER TABLE '|| schemaname || '.'|| tablename ||'' OWNER
TO '||newOwner||'' as query FROM pg_tables WHERE NOT schemaname IN ('pg_catalog',
'information_schema'))
    loop
        execute rec.query;
    end loop;
    --Sequences
    for rec in(SELECT 'ALTER SEQUENCE '|| sequence_schema || '.'||
sequence_name ||'' OWNER TO '||newOwner||'' as query FROM
information_schema.sequences WHERE NOT sequence_schema IN ('pg_catalog',
'information_schema'))
    loop
        execute rec.query;
    end loop;
    --Views
    for rec in(SELECT 'ALTER VIEW '|| table_schema || '.'|| table_name ||''
OWNER TO '||newOwner||'' as query FROM information_schema.views WHERE NOT
table_schema IN ('pg_catalog', 'information_schema'))
    loop
        execute rec.query;
    end loop;
```

```
--Materialized views
for rec in(SELECT 'ALTER TABLE '|| oid::regclass::text ||' OWNER TO
'||newOwner||';' as query FROM pg_class WHERE relkind = 'm')
loop
    execute rec.query;
end loop;
--Functions
for rec in(SELECT 'ALTER FUNCTION
'||n.nspname||'.'||p.proname||'"('||pg_catalog.pg_get_function_identity_arguments(p.
oid)||') OWNER TO '||newOwner||';' as query
FROM pg_catalog.pg_proc p
JOIN pg_catalog.pg_namespace n ON n.oid = p.pronamespace
where n.nspname NOT IN ('pg_catalog', 'information_schema') AND
p.prokind IN ('f', 'a', 'w'))
loop
    execute rec.query;
end loop;
--Procedures
for rec in(SELECT 'ALTER PROCEDURE
'||n.nspname||'.'||p.proname||'"('||pg_catalog.pg_get_function_identity_arguments(p.
oid)||') OWNER TO '||newOwner||';' as query
FROM pg_catalog.pg_proc p
JOIN pg_catalog.pg_namespace n ON n.oid = p.pronamespace
where n.nspname NOT IN ('pg_catalog', 'information_schema') AND
p.prokind = 'p')
loop
    execute rec.query;
end loop;
end$$;
```

И ввести команды:

```
export PGPASSWORD=123
```

```
pg_restore --host localhost --port 5432 --username=sa --dbname=<ИМЯ_БД> --no-owner --no-privileges
/path/to/db.backup
```

```
psql --host localhost --port 5432 --username=sa --dbname=<ИМЯ_БД> -c "ALTER DATABASE <ИМЯ_БД>
OWNER TO someuser"
```

```
psql --host localhost --port 5432 --username=sa --dbname=<ИМЯ_БД> --
file=/path/to/ChangeDbObjectsOwner_Postgres11.v2.sql --variable owner=someuser --variable
ON_ERROR_STOP=1
```

При возникновении ошибки:

ERROR: out of shared memory

HINT: You might need to increase max_locks_per_transaction.

```
/etc/postgresql/12/main/postgresql.conf (/var/lib/pgpro/std-12/data/postgresql.conf)
```

```
max_locks_per_transaction = 256  
max_pred_locks_per_transaction = 1000
```

Настройка сервера Redis

Установка пакетов

```
sudo su
apt-get update
apt-get install redis-server
systemctl enable redis-server
```

```
mcedit /etc/redis/redis.conf
```

Найдите запись "supervised no". Замените запись на "supervised systemd"

Найдите запись "bind 127.0.0.1". Замените запись на "#bind 127.0.0.1"

Найдите запись "protected-mode yes". Замените запись на "protected-mode no"

```
systemctl restart redis-server
exit
```


Настройка сервера приложения

Для того, чтобы работал скрипт деплоя, необходимо отключить ввод пароля при использовании sudo:

/etc/sudoers

```
%sudo    ALL=(ALL) NOPASSWD: ALL
```

Установка пакетов

```
sudo su

apt-get update && apt-get install apt-transport-https

wget -q https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb #Debian 9, Astra (Orel)
# или
wget -q https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb #Ubuntu 20.04

dpkg -I packages-microsoft-prod.deb

apt-get update && apt-get install -y dotnet-sdk-3.1 libgdiplus libc6-dev unzip
#Debian, Ubuntu, Astra (Orel)
# или
apt-get update && apt-get install -y dotnet-sdk-3.1 libgdiplus unzip dotnet-aspnetcore-3.1 #АльтЛинукс

#Распаковать архив с ПО.
unzip -d /etc/bpmsoft/ /path/to/BPM_SOFT_ARCHIVE_NAME.zip

exit
```

Настроить файл /etc/bpmsoft/ConnectionStrings.config (Важно! Номер базы данных Redis должен быть уникальным для каждого сайта приложения).

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <add name="db" connectionString="Pooling=true; Database=<ИМЯ_БД>; Host=<database.server.host>;
Port=5432; Username=someuser; Password=111; Timeout=5; CommandTimeout=400" />
  <add name="redis" connectionString="host=<redis.server.host>;db=0;port=6379" />
  <add name="tempDirectoryPath" connectionString="%TEMP%/%USER%/%APPLICATION%" />
  <add name="elasticsearchCredentials" connectionString="User=gs-es; Password=DEQpJMfKqUVTWg9wYVgi;"
/>
</connectionStrings>
```

Скопировать сертификат .pfx в папку приложения.

При необходимости настроить файл /etc/bpmsoft/appsettings.json (Важно. Предварительно нужно получить цифровой сертификат от центра сертификации в формате *.pfx).

```
"Https": { "Url": "https://::5002", "Certificate": { "Path":  
"C:\\Projects\\site\\20210215_103239\\localhost.pfx", "Password": "Password" }
```

Создать пользователя, от имени которого будет запускаться приложение:

Создание пользователя для сервиса

```
sudo useradd -m dotnetuser # по поводу необходимости задания пароля пользователю есть  
сомнения (ключ -p dotnetpass)
```

```
# Выдача прав на папку приложения  
chown -R dotnetuser /etc/bpmsoft
```

где **dotnetuser** — это имя пользователя, от которого запускается приложение (у разных клиентов оно может отличаться);

Для того, чтобы приложение запускалось как сервис нужно создать файл `/etc/systemd/system/dotnet-bpmsoft.service` следующего содержания:

`/etc/systemd/system/dotnet-bpmsoft.service`

```
[Unit]  
Description=DotNet BPMSoft Service  
  
[Service]  
Environment="COMPlus_ThreadPool_ForceMinWorkerThreads=100"  
ExecStart=/usr/bin/dotnet BPMSoft.WebHost.dll  
WorkingDirectory=/etc/bpmsoft/  
User=dotnetuser  
Group=dotnetuser  
#Restart=on-failure  
Restart=always  
SyslogIdentifier=dotnet-bpmsoft-service  
PrivateTmp=true  
  
[Install]  
WantedBy=multi-user.target
```

где **dotnet-bpmsoft** – это имя сервиса (у разных клиентов оно может отличаться).

Далее выполнить:

```
systemctl daemon-reload  
systemctl enable dotnet-bpmsoft  
systemctl start dotnet-bpmsoft  
systemctl status dotnet-bpmsoft -l
```

Настройка логирования

Для настройки логирования приложения нужно создать файлы следующего содержания:

/etc/rsyslog.d/01-json-template.conf

```
template (name="json-template"
type="list") {
constant (value="{")
constant (value="\">@timestamp\":"\")          property (name="timereported"
dateFormat="rfc3339")
constant (value="\",\">@version\":"\@1")
constant (value="\",\@message\":"\@")          property (name="msg" format="json")
constant (value="\",\@host\":"\@")             property (name="hostname")
constant (value="\",\@severity\":"\@")         property (name="syslogseverity-text")
constant (value="\",\@facility\":"\@")          property (name="syslogfacility-text")
constant (value="\",\@programname\":"\@")      property (name="programname")
constant (value="\",\@procid\":"\@")           property (name="procid")
constant (value="\}\n")
}
```

/etc/rsyslog.d/dotnet-bpmssoft-service.conf

```
#Отправка логов в файл
if $programname == 'dotnet-bpmssoft-service' then /var/log/dotnet-bpmssoft-service.log
#Отправка логов на syslog сервер
if $programname == 'dotnet-bpmssoft-service' then @@elk.domain.ru:10514;json- template
& stop
```

Перезапустить rsyslog:
systemctl restart rsyslog

Для ротации логов сервиса необходимо создать файл **/etc/logrotate.d/dotnet-bpmssoft-service** следующего содержания:

/etc/logrotate.d/dotnet-bpmssoft-service

```
/var/log/dotnet-bpmssoft-service.log {
daily
rotate 7
size 10M
compress
delaycompress
}
```

Вручную приложение можно запустить так:

```
cd /etc/bpmsoft
```

```
COMPlus_ThreadPool_ForceMinWorkerThreads=100 dotnet BPMSoft.WebHost.dll
```

Запуск системы:

```
https://<stand.host.name>:5002
```