

РУКОВОДСТВО ПО УСТАНОВКЕ ПО

Установка подсистем «Централизованная бухгалтерия»
и «Зарплата и кадры»

Содержание

1. Требования к программному обеспечению серверов3

2. Установка PostgreSQL4

3. Установка приложения6

3.1. Отключение SELinux6

3.2. Установка зависимостей6

3.3. Создание пользователя7

3.4. Установка python 37

3.5. Установка пакета с приложением8

4. Настройка ОС11

4.1. Для CentOS 711

4.2. NGINX11

4.3. Supervisor13

4.4. RabbitMQ13

4.5. Redis14

1. Требования к программному обеспечению серверов

- Программное обеспечение сервера приложений: ОС CentOS 7.6 x86_64.
- Программное обеспечение сервера баз данных: ОС CentOS 7.6 x86_64; СУБД PostgreSQL 11.

2. Установка PostgreSQL

Важно! Установка производится на сервере для СУБД.

Для установки СУБД PostgreSQL необходимо в рабочей консоли выполнить следующие команды:

- добавить репозиторий с установочным пакетом postgres:

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

- установить СУБД PostgreSQL и необходимые библиотеки из репозитория:

```
# sudo yum install -y postgresql13-server
```

- инициализировать БД, т.е. подготовить ее к работе:

```
# sudo /usr/pgsql-9.6/bin/postgresql13-setup initdb
```

- добавить СУБД PostgreSQL в автозагрузку:

```
# sudo systemctl enable postgresql-13
```

- запустить СУБД PostgreSQL:

```
# sudo systemctl start postgresql-13
```

- создать для приложения пользователя:

```
# sudo -u postgres psql -c "create user bars_web_bb password 'bars_web_bb'"
```

- создать для приложения БД:

```
# sudo -u postgres psql -c "create database bars_web_bb owner bars_web_bb"
```

Примечание. После создания БД необходимо оптимизировать ее работу под мощности сервера. Подробная информация о порядке оптимизации доступна на странице:

<http://postgresql.leopard.in.ua>. Для первоначальной настройки, например, можно воспользоваться информацией на странице: <http://pgtune.leopard.in.ua>.

- Настроить права доступа к базе данных. Для этого нужно прописать правило в файле `pg_hba.conf`, например:

#	TYPE	DATABASE	USER	ADDRESS	METHOD
host		all	all	127.0.0.1/32	md5

Примечание. Если приложение на другом хосте, то вместо 127.0.0.1 указывается IP-адрес сервера приложения.

Если база данных размещается на отдельном хосте, то в этом случае необходимо в файле `postgresql.conf` раскомментировать параметр `listen_addresses` и указать значение:

```
listen_addresses = '*'
```

Чтобы применить все сделанные изменения рекомендуется перезапустить СУБД PostgreSQL:

```
# sudo systemctl restart postgresql-13
```

3. Установка приложения

3.1. Отключение SELinux

Переведите SELinux в режим **permissive**:

```
$ sudo setenforce 0
```

Измените в файле `/etc/selinux/config` параметр **SELINUX** на значение **disabled**. В результате должно получиться примерно так:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3.2. Установка зависимостей

Установка репозитория **epel-release**:

```
$ sudo yum install -y epel-release
```

Обновление списка доверенных сертификатов:

```
$ sudo yum install -y ca-certificates nss nss-util
```

Установка репозитория **ius**:

```
$ sudo yum localinstall -y https://centos7.iuscommunity.org/ius-release.rpm
```

Установка репозитория **postgresql**:

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

Создайте файл по пути `/etc/yum.repos.d/nginx.repo` и заполните его:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=0
enabled=1
```

Скачайте ключ и импортируйте его:

```
$ curl -L -O https://nginx.org/keys/nginx_signing.key
$ sudo rpm --import nginx_signing.key
```

Установка системных пакетов:

```
$ sudo yum install -y python python-devel supervisor python-setuptools \
python-virtualenv postgresql96 libxml2-devel \
libxslt-devel libjpeg-devel ncurses-devel readline-devel \
xz-libs zlib-devel rabbitmq-server redis nginx git gcc \
gcc-c++ bzip2 bzip2-devel sqlite sqlite-devel openssl-devel \
xz xz-devel libffi-devel openldap-devel
```

3.3. Создание пользователя

Создайте пользователя, от имени которого будут запускаться процессы приложения. Например, пользователь **www-data** с домашней директорией **/var/www**:

```
$ sudo useradd -m -d /var/www -U -s /bin/bash www-data
```

Смените пользователя. Дальнейшие действия будут выполняться от пользователя, созданного в предыдущем пункте:

```
$ sudo -u www-data -i
```

3.4. Установка python 3

Склонируйте репозиторий **pyenv**, пропишите переменные в скрипты инициализации окружения:

```
$ git clone https://github.com/pyenv/pyenv.git ~/.pyenv
$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile
$ echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n    eval "$(pyenv init -)"\nfi' >>
~/.bash_profile
```

Переинициализируйте shell:

```
$ source $HOME/.bash_profile
```

Установите **python 3.9.6**:

```
$ pyenv install 3.9.6
```

3.5. Установка пакета с приложением

Создайте виртуальное окружение:

```
$ /var/www/.pyenv/versions/3.9.6/bin/python -m venv /var/www/venv/web_bb
```

Распакуйте архив с пакетами обновления в отдельный каталог. Например, `/tmp/web_bb_update`.

Пропишите в переменную **PATH** путь до **pg_config**:

```
$ export PATH=/usr/pgsql-13/bin:$PATH
```

Активируйте **virtualenv** окружение:

```
$ source venv/web_bb/bin/activate
```

Запустите установку пакета:

```
$ pip install -U setuptools
$ pip install pip==22.1
$ pip install m3-builder==1.2.0 --extra-index-url http://pypi.bars-open.ru/simple/ --
trusted-host pypi.bars-open.ru
$ pip install --extra-index-url https://pypi.org/simple --trusted-host https://pypi.bars-
open.ru web-bb-app -f /tmp/web_bb_update
```

Создайте по пути `/var/www/web_bb_config` конфигурационные файлы `gunicorn.conf.py` и `supervisord_web_bb.conf`.

`gunicorn.conf.py`:

```
import os, multiprocessing
from gunicorn import util

short_name = "web_bb"
proc_name = "%s" % (short_name)
run_dir = "/var/run"
log_dir = "/var/www/log/%s" % (short_name)

bind = "unix:%s/%s/%s.sock" % (run_dir, short_name, short_name)
pidfile = "%s/%s/%s.pid" % (run_dir, short_name, short_name)
workers = multiprocessing.cpu_count() * 2 + 1
user = "www-data"
group = "www-data"
errorlog = "%s/gunicorn.log" % (log_dir)
loglevel = "info"
timeout = 3000
max_requests = 1000
```

В параметрах **user** и **group** укажите имя пользователя и группу, которые будут использоваться для запуска процессов приложения. Замените значение параметра **log_dir** на

предпочтительный путь для хранения логов приложения. Создайте соответствующий каталог, если еще не создан:

```
$ mkdir -p /var/www/log/web_bb
```

supervisor_web_bb.conf:

```
[program:web_bb]
environment=LANG="ru_RU.UTF-8",WEB_BB_CONF="/var/www/web_bb_config/project.conf",DJANGO_SETTINGS_MODULE="web_bb_app.settings",PYTHONIOENCODING='UTF-8'
command=/var/www/venv/web_bb/bin/gunicorn -c /var/www/web_bb_config/gunicorn.conf.py wsgi
directory=/var/www/venv/web_bb/lib/python3.6/site-packages/web_bb_app/
user=www-data
group=www-data
daemon=False
debug=False
autostart=True
autorestart=True
redirect_stderr=True
stdout_logfile=/var/www/log/web_bb/main.log

[program:web_bb_celery]
environment=LANG="ru_RU.UTF-8",WEB_BB_CONF="/var/www/web_bb_config/project.conf",DJANGO_SETTINGS_MODULE="web_bb_app.settings",PYTHONIOENCODING='UTF-8'
command=/var/www/venv/web_bb/bin/django-admin celery worker -l info -Q celery,highpriority,tracking,waterfall,salary,salary_highpriority,salary_copy_queue,oss
directory=/var/www/venv/web_bb/lib/python3.6/site-packages/web_bb_app/
user=www-data
group=www-data
daemon=False
debug=False
autostart=True
autorestart=True
redirect_stderr=True
stdout_logfile=/var/www/log/web_bb/celery.log
```

Аналогично конфигурационному файлу выше, отредактируйте значения параметров **stdout_logfile**, **user**, **group** на собственные, если они отличаются.

При установке пакета по пути **/var/www/web_bb_config** должен был появиться файл **project.conf.example**. Его необходимо использовать для создания конфигурационного файла приложения. Для этого скопируйте файл по пути, по которому должен располагаться ваш конфигурационный файл(значение переменной **WEB_BB_CONF** в **supervisor_web_bb.conf**):

```
$ cp /var/www/web_bb_config/project.conf.example /var/www/web_bb_config/project.conf
```

Отредактируйте параметры конфигурационного файла на собственные. Обязательно переопределите значения параметров в секции **static**. Например, на следующие:

```
[static]
ROOT = /var/www/web_bb_config/static
DOWNLOADS = /var/www/web_bb_config/downloads
SCREENSHOTS = /var/www/web_bb_config/screenshots
```

А также указать необходимые для работы приложения плагины в секции:

```
[plugins]
ACTIVATED_PLUGINS =
```

Создайте базу для приложения, настройте подключение и выполните management-команды:

```
$ export WEB_BB_CONF=/var/www/web_bb_config/project.conf
DJANGO_SETTINGS_MODULE=web_bb_app.settings
$ django-admin migrate --noinput
$ django-admin collectstatic --noinput
```

Разлогиньтесь. Последующие шаги должны выполняться от имени привилегированного пользователя.

4. Настройка ОС

4.1. Для CentOS 7

По пути **/etc/tmpfiles.d** создайте файл(например, **web_bb.conf**). В файл добавьте:

```
D /var/run/web_bb 0775 www-data www-data -
```

Выполните команду:

```
$ sudo systemd-tmpfiles --create
```

4.2. NGINX

Отредактируйте в конфигурационном файле **nginx** (путь по умолчанию: **/etc/nginx/nginx.conf**) имя пользователя, от которого выполняется запуск веб-сервера. Для этого измените значение параметра **user** на имя пользователя, под которым будет работать приложение.

Определите путь, по которому nginx будет выполнять поиск дополнительных конфигурационных файлов (определяется параметром **include**, значение по умолчанию:

`/etc/nginx/conf.d/*.conf`). Создайте файл, который подходил бы по маске значению параметра. Например, `/etc/nginx/conf.d/web_bb.conf`. В файл добавьте:

```
upstream web_bb_server {
    server unix:/var/run/web_bb/web_bb.sock fail_timeout=0;
}

log_format custom '$remote_addr - [$time_local] "$host" "$request" '
                  '$status ($bytes_sent) "$http_referer" '
                  '"$uri $args" [$request_time] [$upstream_response_time]';

server {
    listen      *:80;
    keepalive_timeout 65;
    access_log  /var/log/nginx/nginx_access.log custom;
    error_log   /var/log/nginx/nginx_error.log;

    location / {
        if ( -e /var/www/html/servicework ) {
            return 503;
        }
        proxy_redirect      off;
        proxy_set_header    Host $host;
        proxy_set_header    X-Url-Scheme $scheme;
        proxy_set_header    X-Host $http_host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    Proxy-host $proxy_host;
        client_max_body_size 400m;
        client_body_buffer_size 128k;
        proxy_buffering      off;
        proxy_connect_timeout 7200;
        proxy_send_timeout   7200;
        proxy_read_timeout   7200;
        proxy_buffers         8 32k;
        proxy_pass http://web_bb_server;
    }

    location /static/ {
        alias /var/www/web_bb_config/static/;
    }
    location /static/downloads/ {
        alias /var/www/web_bb_config/downloads/;
    }
    location /static/screenshots/ {
        alias /var/www/web_bb_config/screenshots/;
    }
}
```

Выполните перезапуск сервиса:

```
$ sudo systemctl restart nginx
```

Добавьте в автозагрузку:

```
$ sudo systemctl enable nginx
```

4.3. Supervisor

В основной конфигурационном файле (путь по умолчанию: `/etc/supervisord.conf`) переопределите следующие строки:

```
[unix_http_server]
chmod=0777
chown=www-data:www-data

[include]
files = /etc/supervisord.d/*.conf
```

Создайте каталог `/etc/supervisord.d`, если его не существует:

```
$ sudo mkdir -p /etc/supervisord.d
```

Создайте символическую ссылку на конфигурационный файл `supervisor_web_bb.conf`, который создавался в предыдущих пунктах:

```
$ sudo ln -s /var/www/web_bb_config/supervisor_web_bb.conf /etc/supervisord.d/web_bb.conf
```

Перезапустите сервис:

```
$ sudo systemctl restart supervisord
```

Добавьте в автозагрузку:

```
$ sudo systemctl enable supervisord
```

4.4. RabbitMQ

Запустите сервис:

```
$ sudo systemctl start rabbitmq-server
```

Добавьте в автозапуск:

```
$ sudo systemctl enable rabbitmq-server
```

Создайте виртуальный хост:

```
$ sudo rabbitmqctl add_vhost web_bb
```

Создайте пользователя:

```
$ sudo rabbitmqctl add_user web_bb web_bb
```

Раздайте права на виртуальный хост:

```
$ sudo rabbitmqctl set_permissions -p web_bb web_bb ".*" ".*" ".*"
```

В конфигурационном файле приложения измените значение параметра **CELERY_BROKER_URL** в секции **waterfall** согласно шаблону (подставьте вместо ПОЛЬЗОВАТЕЛЬ, ПАРОЛЬ, ВИРТУАЛЬНЫЙ_ХОСТ свои значения):

```
CELERY_BROKER_URL = amqp://ПОЛЬЗОВАТЕЛЬ:ПАРОЛЬ@192.168.228.172:5672/ВИРТУАЛЬНЫЙ_ХОСТ
```

4.5. Redis

Запустите сервис:

```
$ sudo systemctl start redis
```

Добавьте в автозагрузку:

```
$ sudo systemctl enable redis
```