АО «БАРС Груп»

БАРС.Бюджет-Автотранспорт

Инструкция по установке программного обеспечения

2024

Содержание

Аннотация

В данном документе приводится необходимая информация по установке и настройке компонентов ПО «БАРС.Бюджет-Автотранспорт» на операционной системе Ubuntu Server 14.04 x86_64.

Примечание. Для других Linux систем команды схожи, но могут отличаться.

Перечень терминов и сокращений

Обозначение	**Описание**
БД	База данных
Репозиторий	Хранилище файлов, доступных для дальнейшего распространения по сети
СУБД	Система управления базой данных

1 Установка и настройка

Установка компонентов ПО «БАРС.Бюджет-Автотранспорт» и всех его составляющих выполняется с помощью команд в рабочей консоли (Рис. 1), которую можно открыть, например, с помощью нажатия клавиш Ctrl+Alt+F1:

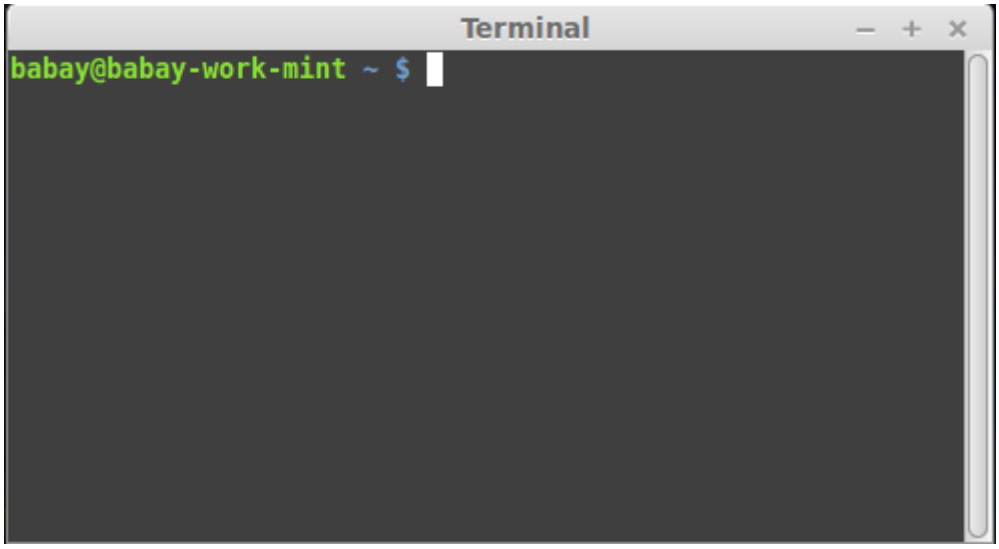

Рис. 1. Рабочая консоль

Для установки зависимостей необходимо выполнить в терминале сервера следующую команду.

```
root@host:~# apt-get install python-dev python-virtualenv python-pip libxml2 libxml2-dev libxslt1-dev ghostscript zlib1g-dev python-virtualenv libffi_dev libxmlsec1_dev swig
```

Где:

- libxml2, libxml2-dev и libxslt1-dev - потребуется для работы с xml,
- ghostscript - потребуется для печатных форм с встроенными изображениями,
- libffi_dev - потребуется для сборки pip-пакета cryptography,
- libxmlsec1_dev и swig - потребуются для сборки зависимостей python-saml.

2 Установка и конфигурирование PostgreSQL

Установка PostgreSQL

Для установки СУБД PostgreSQL 9.6 необходимо в терминале сервера выполнить следующие команды:

```
# yum install postgresql13-server
```

> **Примечание.** Если в репозиториях нет установочного пакета postgresql133-server, то СУБД PostgreSQL 13 следует установить из источника ниже:
>
> http://wiki.postgresql.org/wiki/YUM_Installation

Настройка PostgreSQL

Для корректной работы модуля, необходимо в папке с установленной СУБД PostgreSQL в файле *postgresql.conf* найти и исправить параметр *datestyle*:

```
datestyle = 'iso, dmy'
```

Создание базы данных и пользователя для системы

Необходимо войти в консоль управления PostgreSQL:

```
# su postgres -c 'psql'
```

Создать пользователя с паролем:

```
create user bars_web_bb password 'bars_web_bb';
```

Создать базу данных и указать созданного пользователя ее владельцем:

```
create database bars_web_bb owner bars_web_bb;
```

Выйти из консоли PostgreSQL:

```
\q
```

Загрузить первоначальные данные в базу данных из резервной копии:

```
# su postgres

$ pg_restore --username postgres --dbname bars_web_bb web_bb_demo.backup
```

3 Установка и конфигурирование RabbitMQ

Установка необходимых пакетов

Необходимо установить пакет rabbitmq-server можно из репозитория epel (его нужно подключить) или скачать с официального сайта разработчиков. Для того чтобы установить пакет из репозитория epel необходимо выполнить следующие команды:

Установить репозиторий и пакет:

```
# su -c 'rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm'

# yum install rabbitmq-server
```

Запустить сервер rabbitmq:

```
# /etc/init.d/rabbitmq-server start
```

Добавить сервер rabbitmq в автозагрузку:

```
# chkconfig rabbitmq-server on
```

Настройка rabbitmq

Создать пользователя:

```
# rabbitmqctl add_user web_bb web_bb
```

Создать хост:

```
# rabbitmqctl add_vhost web_bb
```

Настроить права на хост:

```
# rabbitmqctl set_permissions -p web_bb web_bb ".*" ".*" ".*"
```

В файле настроек приложения *project.conf* необходимо указать хост и пользователя для подключения к RabbitMQ, а также включить фоновые процессы:

```
[waterfall]
```

```
WTF_ASYNC = True

CELERY_BROKER_URL = amqp://web_bb:web_bb@localhost:5672/web_bb
```

4 Установка и конфигурирование Supervisor

Установить Supervisor:

```
# pip install supervisor
```

Создать файл конфигураций в supervisor:

```
# echo_supervisord_conf > /etc/supervisord.conf
```

В папке /etc/ появится файл конфигурации *supervisord.conf*, необходимо открыть его и изменить следующие строки:

```
chmod=0700                 ; socket file mode (default 0700)

chown=www-data:www-data    ; socket file uid:gid owner
```

В нижней части файла конфигурации *supervisord.conf* следует добавить строки:

```
[include]

files = /etc/supervisor/conf.d/*.conf
```

Создать папку *supervisor* и в ней *conf.d* – в этой папке будут храниться файлы конфигураций для запуска приложений:

```
# mkdir -p /etc/supervisor/conf.d
```

Скопировать в папку файл конфигурации для проекта *web_bb.conf* или его создать:

```
# cp /var/www/web_bb_config/supervisord.conf.example /etc/supervisor/conf.d/web_bb.conf
```

Файл конфигурации проекта содержит настройки запуска проекта:

```
web_bb.conf

[program:web_bb]

#Переменные окружения для проекта.

environment=PATH="/home/venv/web_bb/bin/",WEB_BB_CONF="/var/www/web_bb_config/project.conf"
```

```
#команда запуска

command=/home/venv/web_bb/bin/python manage.pyc –c
/var/www/web_bb_config/gunicorn.conf.py

#Директория, в которой будет выполняться программа.

directory=/home/venv/web_bb/lib/python3.9/site-packages/web_bb/

#Пользователь и группа, от имени который будет запускаться приложение

user=www-data

group=www-data

daemon=False

debug=False

autostart=true

autorestart=true

redirect_stderr=True

stdout_logfile=/var/log/web_bb/main.log

[program:web_bb_celery]

environment=PATH="/home/venv/web_bb/bin/",
WEB_BB_CONF="/var/www/web_bb_config/project.conf"

command=/home/venv/web_bb/bin/python manage.pyc celery worker -l info

directory=/home/venv/web_bb/lib/python3.9/site-packages/web_bb/

user=celery

group=www-data

numprocs=1

autostart=true

autorestart=true

startsecs=10

redirect_stderr=true
```

```
stopwaitsecs = 600

stdout_logfile=/var/log/web_bb/celery.log

[program:pushme_server]

command=python -m pushme.server -p 9999

user=www-data

group=www-data

autostart=true

autorestart=true

startsecs=10

redirect_stderr=true

stdout_logfile=/var/log/web_bb/pushme_server.log

[program:pushme_queue]

command=python -m pushme.queue -p 4000

user=www-data

group=www-data

autostart=true

autorestart=true

startsecs=10

redirect_stderr=true

stdout_logfile=/var/log/web_bb/pushme_queue.log
```

Автозагрузка Supervisor

Добавление supervisor в автозагрузку зависит от дистрибутива Linux. На сайте supervisor есть все необходимые конфигурационные файлы.

5 Установка и конфигурирование Nginx

Копирование или создание файла web_bb.nginx.conf в каталог /etc/nginx/conf.d:

```
# cp /var/www/web_bb_config/nginx.conf.example /etc/nginx/conf.d/web_bb.nginx.conf
```

Ниже приведен пример виртуального хоста nginx:

Важно! Для корректной работы виртуального хоста nginx – в конфигурационном фале */etc/nginx/nginx.conf* должна быть раскомментирована строка: *include /etc/nginx/conf.d/*.conf*. В этой директории создаем файл web_bb.nginx.conf со следующим содержимым

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

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    10800;
        proxy_send_timeout       10800;
        proxy_read_timeout       10800;
        proxy_buffers           8 32k;

        add_header  PR-B-TOTAL  $request_time;

        proxy_pass http://web_bb_server;

    location /protected/downloads/ {
        alias /var/www/web_bb_config/share/downloads/;
        internal;
    }

    location /protected/media/ {
        alias /var/www/web_bb_config/share/media/;
        internal;
    }
    }
}
```

6 Установка и конфигурирование приложения

Создание пользователя www-data

Веб сервер и supervisor необходимо заускать от пользователя www-data. Если данного пользователя ещё нет в системе, то его необходимо создать.

```
root@host:~# useradd www-data –U -d /home/venv
```

Если пользователь уже есть, необходимо изменить его домашнюю директорию и login shell.

```
root@host:~# usermod www-data -s /bin/bash -d /home/venv
```

Создание необходимых директорий

Для создания необходимых дирректорий и назначения прав доступа необходимо выполнить в терминале сервера следующие команды.

```
root@host:~# mkdir /home/venv/
root@host:~# chown -R www-data:www-data /home/venv/
root@host:~# mkdir /var/log/paidserv/
root@host:~# mkdir -p /var/www/paidserv/
root@host:~# mkdir /var/www/paidserv/{media,static,downloads}
root@host:~# chown -R www-data:www-data /var/log/paidserv/
root@host:~# chown -R www-data:www-data /var/www/paidserv/
```

Создание виртуального окружения

Для создания виртуального окружения необходимо выполнить в терминале сервера следующие команды.

```
root@host:~# sudo -u www-data virtualenv --no-site-packages /home/venv/paidserv
```

Установка и конфигурирование приложения

Для установки приложения из пакета необходимо выполнить в терминале сервера следующие команды.

```
root@host:~# su www-data
```

```
www-data@host:~$ source /home/venv/paidserv/bin/activate

(paidserv)www-data@host:~$ pip install paidserv-XXX.tar.gz
```

После установки пакета необходимо произвести конфигурирование приложения.

Пример конфигурации находится в файле /var/www/paidserv/project.conf.example.

```
(paidserv)www-data@host:~$ cp /var/www/paidserv/project.conf.example
/var/www/paidserv/project.conf

(paidserv)www-data@host:~$ nano /var/www/paidserv/project.conf
```

Далее необходимо обновить следующие параметры.

```
[datalogger]

shutup = True

[region]

REGION = tatarstan

[paidserv]

PLUGINS = paidserv.kinder

[rtk]

RTK = False

[locations]

MEDIA_ROOT = /var/www/paidserv/media/

STATIC_ROOT = /var/www/paidserv/static/

DOWNLOADS_ROOT = /var/www/paidserv/downloads/
```

После сохранения файла необходимо экспортировать переменные окружения.

Переменная DJANGO_SETTINGS_MODULE опциональна: при отсутствии используется значение 'paidserv.settings'.

```
(paidserv)www-data@host:~$ export DJANGO_SETTINGS_MODULE=paidserv.settings

(paidserv)www-data@host:~$ export PAIDSERV_CONFIG=/var/www/paidserv/project.conf
```

Далее необходимо подготовить БД.

```
(paidserv)www-data@host:~$ python -m paidserv.prepare_db
```

Был случай, когда команда prepare_db упала с ошибкой

```
django.db.utils.DatabaseError: ОШИБКА:  отношение "south_migrationhistory" не существует
```

В этом случае нужно в папке с виртуальным окружением найти папку site-packages/south. В этой папке не должно быть папки migrations, если есть то нужно удалить migrations, потом повторить prepare_db.

Если в ходе миграций появилась ошибка, что отношение finance не существует, то необходимо выполнить следующие команды:

```
python manage.pyc migrate finances 0001

python manage.pyc migrate finances 0002

python manage.pyc migrate finances 0003

....

python manage.pyc migrate finances 0016

python manage.pyc migrate finances 0017
```

А затем повторить prepare_db.

На запрос системы «Would you like to create one now? (yes/no):» необходимо ответить «yes» и ввести параметры суперпользователя.

Далее необходимо сгенерировать статичные файлы. Файлы будут сгенерированы в директорию /var/www/paidserv/static/.

```
(paidserv)www-data@host:~$ python -m paidserv.manage collectstatic --noinput
```

Если статика не собирается с ошибкой "No module named operational_journal.dbrouters". В таком случае сработает команда

```
(paidserv)www-data@host:~$ python /var/www/paidserv/src/paidserv/manage.pyc collectstatic --noinput

или

(paidserv)www-data@host:~$ python /var/www/venv/paidserv/lib/python2.7/site-packages/paidserv/manage.py collectstatic --noinput
```

7 Установка и конфигурирование Gunicorn

Для установки WSGI-сервера Gunicorn необходимо выполнить следующие команды.

Важно! В данном случае команды выполняются в виртуальном окружении.

```
# su www-data

$ source /home/venv/web_bb/bin/activate

(web_bb)$  pip install gunicorn
```

По умолчанию, после установки приложения, файл настройки для *gunicorn* будет находиться по адресу: */var/www/web_bb_config/gunicorn.conf.py*

Файл *gunicorn.conf.py* содержит параметры запуска приложения:

```
#coding: utf-8

import os, multiprocessing

short_name  = "web_bb"

run_dir     = "/var/tmp"

log_dir     = "/var/log/web_bb"


bind      = "unix:%s/%s.sock" % (run_dir, short_name)

proc_name = "%s/%s" % (run_dir, short_name)

workers   = multiprocessing.cpu_count() * 2 + 1

user      = "www-data"

group     = "www-data"

errorlog   = "%s/gunicorn.log" % log_dir

loglevel  = "info"

pidfile   = "%s/%s.pid" % (run_dir, short_name)

timeout   = 3000
```

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докумен-те	Номер докумен-та	Входящий номер сопроводи-тельного документа и дата	Под-пись	Дата
	измененных	замененных	новых	аннулированных					