

1. Общие сведения

Для корректной работы программного комплекса необходима операционная система Ubuntu 14.04

2. Установка серверной подсистемы

2.1. Добавление в систему кодировки ru_RU.UTF-8

Для корректной работы с русскими символами на сервере должна быть установлена кодировка ru_RU.UTF-8. Для этого необходимо выполнить следующую команду в консоли:

```
sudo locale-gen ru_RU.UTF-8  
sudo dpkg-reconfigure locales
```

2.2. Необходимые компоненты

Для работы серверной подсистемы на сервере должны быть установлены следующие пакеты:

```
python  
python-pip  
python-dev  
python-psycopg2  
libjpeg62  
zlib1g-dev  
libjpeg-dev  
libpq-dev  
uwsgi  
uwsgi-plugin-python  
remcached  
redis  
postgresql  
nginx
```

2.3. Настройка СУБД Postgresql

2.4. Создание пользователя octopus_backend

Для создания пользователя, под которым система будет работать с СУБД, необходимо выполнить в psql следующую команду:

```
CREATE USER octopus_backend_demo WITH LOGIN CREATEDB PASSWORD  
'JoiJfowefOIJFWE98098LKJ';
```

2.5. Создание базы данных octopus_backend

Для создания базы данных, с которой будет работать система, необходимо выполнить следующую команду psql:

```
CREATE DATABASE octopus_backend_demo WITH TEMPLATE = template0 ENCODING =  
'UTF8' LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8' OWNER  
octopus_backend_demo;
```

2.6. Подготовка сервера

Необходимо выполнить следующий набор команд:

```
sudo mkdir /var/www/octopus  
sudo chown www-data /var/www/octopus
```

2.7. Настройка серверной подсистемы

Для настройки серверной подсистемы необходимо создать файл /etc/octopus/octopus-backend-config.ini со следующим содержимым:

[database]

*engine = django.db.backends.postgresql_psycopg2
name = octopus_backend_demo
user = octopus_backend_demo
password = JoijfowefOIJFWE98098LKJ
host = localhost
port = 5432*

[django]

*secret_key = LKJPOIjpoijefpoweijfLKJLIJOFIJEWOF890378098324
debug = True
allowed_hosts = *
static_root = /var/www/octopus/backend/static/
static_url = /static/*

*media_root = /var/www/octopus/backend/media/
media_url = /media/*

*password_salt = IOewweoqpoovmskJPuBHRsjlOm
jwt_token_signature = cnNiuniuefLKJOIJWEFjjoiefjoefijoIJ*

application_base_url = http://localhost:8000

sdb_constructor_url = /sdb-constructor/fieldSet

use_mailgun = false

[memcache]

*host = 127.0.0.1
port = 11211*

[redis]

*host = localhost
port = 6379*

[notifier]

*host = localhost
port = 9094*

[file-manager]

*host = localhost
port = 9091*

2.8. Установка основного пакета

Для установки серверной подсистемы необходимо установить, выполнив следующую команду shell:

```
sudo dpkg -i octopus-backend_3.9.0_all.deb
```

2.9. Настройка uwsgi

Для настройки uwsgi необходимо создать файл `/etc/uwsgi/apps-enabled/octopus-backend.ini` со следующим содержанием:

```
[uwsgi]  
vhost = true  
socket = localhost:9092  
uid = www-data  
gid = www-data  
module = octopus_backend.app.wsgi  
env = DJANGO_SETTINGS_MODULE=octopus_backend.app.settings  
wsgi-file = wsgi.py  
post-buffering = on  
buffer-size=32768
```

После этого необходимо выполнить команду shell:

```
sudo service uwsgi restart
```

3. Установка графической подсистемы

Для установки клиентской подсистемы необходимо выполнить следующую команду shell:

```
sudo dpkg -i octopus-frontend-ng5_3.9.0-408_amd64.deb
```

3.1. Настройка графической подсистемы:

Для настройки графической подсистемы необходимо в файл `/var/www/octopus/frontend-ng5/application/config.json` поместить следующее содержимое:

```
{
  "fileManager": {
    "protocol": "http",
    "address": "octopus.abibits.net",
    "port": "80",
    "encodings": ["windows-1250", "windows-1251", "UTF-8", "KOI8-R", "KOI8-U"]
  },
  "backend": {
    "protocol": "http",
    "address": "octopus.abibits.net",
    "port": "80",
    "defaultPath": "api/v2"
  },
  "api": {
    "filesUrl": "http://localhost:8080/api/v1/file-manager",
    "rdsUrl": "http://localhost:8080/api/v1/rds",
    "rdsTunnelEndPoint": "http://localhost:8080/api/v1/remote-session/tunnel"
  },
  "notificationServiceUrl": "http://localhost:9094/subscribe",
  "oldFrontEndUrl": "http://localhost",
  "avatarBaseURI": ""
}
```

4. Настройка web-сервера nginx

Для настройки web-сервера nginx необходимо в файл `/etc/nginx/sites-enabled/octopus.ini` поместить следующее содержимое:

```

server {
    listen 8080;
    resolver 127.0.0.1;
    root /var/www/octopus/backend;

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

    location / {
        root /var/www/octopus/frontend-ng5/application;
        index index.html;
        try_files $uri /index.html =404;
    }

    location ~ ^/new(.*) {
        root /var/www/octopus/frontend-ng5/application;
        index index.html;
        try_files $1 /index.html =404;
    }

    location ~ ^/sdb-creator(.*) {
        root /var/www/octopus/sdb-creator/application;
        index index.html;
        try_files $1 /index.html =404;
    }

    location ~ ^(/api/v1/|/api/v2/) {
        rewrite /api/(.+)/$1 break;
        uwsgi_pass localhost:9092;
        include uwsgi_params;
        uwsgi_intercept_errors on;
        error_page 301 302 = @handle_api_redirect;
    }

    location ~ ^(/backend|/admin|/static|/media|/sdb-creator) {
        uwsgi_pass localhost:9092;
        include uwsgi_params;
    }

    location @handle_api_redirect {
        set $saved_redirect_location '$upstream_http_location';
        proxy_pass $saved_redirect_location;
    }

```

```
    add_header Cache-Control no-cache;  
  }  
}
```

5. Работа с системой

Для начала работы с системой необходимо в браузере Chrome перейти по адресу <http://SERVERADDRESS:8080/new>

где SERVERADDRESS - адрес сервера, на котором осуществлялась установка системы.