**Руководство Разработчика - Развертывание системы**

Каталог реализован по системе клиент-серверное приложение, где используется REST API запросы. Клиент - использует библиотеку “reactjs”, Сервер - “python 2.7”, база данных - “postgresql 10”.

1. Требования по системе

Для сервера можно использовать любую версию linux операционных систем (Рекомендуется “Debian >7, Ubuntu 18.04” или же “Centos”). Программное обеспечение может быть развернуто как на физическом, так и на виртуализированном оборудовании(На данный момент работа ПО была протестирована в виртуальном окружении “VDS”).

Минимальные требования: “Debian >7, 64-bit, 4 GB RAM, 15 GB free disk space”

.

1. Первоначальные настройки операционной системы

* Произвести обновление обновление системы

sudo apt-get update && sudo apt-get upgrade

* Установить пакет “curl”:

sudo apt install curl

* Далее, необходимо установить пакет “Nodejs”

sudo apt install nodejs

* Чтобы проверить, какую версию “Node.js” введится:

nodejs -v

* Установка python 2.7

apt install python

* Чтобы проверить версию “Python” по умолчанию, запрашивается команда “python” и ее версию:

$ python --version

Python 2.7

* Установка “pip” для “Python 2” и все его зависимости:

sudo apt install python-pip

* После завершения установки проверка введя следующую команду, которая напечатает версию “pip”:

pip --version

* “Nginx” является одним из самых популярных веб-серверов в мире и отвечает за размещение некоторых из крупнейших сайтов с наибольшим трафиком в Интернете. В большинстве случаев он более дружественен к ресурсам, чем “Apache”, и его можно использовать в качестве веб-сервера или обратного прокси-сервера.

sudo apt install nginx

* В “nginx” необходимо настроить “site-available” и ярлык “site-enabled”

upstream catalog-api{

server 127.0.0.1:3031;

}

server {

listen 80;

server\_name 176.126.166.21

www.example.kg

example.kg;

client\_max\_body\_size 100M;

location / {

root /var/www/example.kg/client;

try\_files $uri $uri.html $uri/index.html @api;

}

location @api {

uwsgi\_pass catalog-api;

include uwsgi\_params;

}

}

* Команды для “nginx”:

service nginx stop|start|restart|reload|status

* Теперь можно установить сервер “uWSGI”. Его пакет называется uwsgi (однако не следует путать пакет сервера “uWSGI” с протоколом “uwsgi”).

sudo apt-get install uwsgi

* Чтобы убедиться, что установка прошла успешно, запрос версии:

uwsgi --version

* Команда должна вернуть версию сервера “uWSGI”.
* После необходимо настроить “uwsgi”. Нужно создать директории “/etc/uwsgi/apps-available, /etc/uwsgi/apps-enabled и /etc/uwsgi/sites”
* В директории “/etc/uwsgi/apps-available создать файл example.ini”

[uwsgi]

socket = 0.0.0.0:3031

chdir = /var/www/example.kg/server/

wsgi-file = /var/www/example.kg/server/run.py

callable = app

stats = 0.0.0.0:9191

plugins = python,gevent\_python

touch-reload = /etc/uwsgi/apps-available/catalog.ini

* В директориях “/etc/uwsgi/apps-enabled и /etc/uwsgi/sites” создать ярлык на файл “/etc/uwsgi/apps-available/example.ini”.
* Команды для “uwsgi”:

service uwsgi stop|start|restart|reload|status

* Установка “git”

sudo apt install git

* Можно подтвердить, что правильно установили “Git”, выполнив следующую команду:

git --version

* Произвести обновление обновление системы

sudo apt-get update && sudo apt-get upgrade

* Обновление локального кэш пакетов “apt” и установка “Redis”, набрав:

sudo apt install redis-server

* После установки пакета, проверка статуса “redis-server”:

sudo systemctl restart redis

* Установка “postgresql 10” на Debian

echo "deb [arch=amd64] http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg main" > /etc/apt/sources.list.d/pgdg.list

curl -sSL https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

sudo apt update

sudo apt install -y postgresql-10 postgresql-10 postgresql10-contrib

* Если по понятной причине сервер не работает, можно запустить его в любое время, используя следующую команду:

systemctl start postgresql

* Можно получить доступ к работающему серверу, используя:

sudo -u postgres psql

# или su - postgresql -c "psql"

* Теперь можно уже создать пользователя и создать базу.

1. Развертывание Front-end Электронного каталога

*Примечание: Для более удобного и быстрого развертывания можно настроить для проекта CI/CD (на данный момент приложение клиента Электронного каталога настроена на “gitlab CI/CD”)*

Для этого нужно скачать проект локально, установить пакеты из “package.json”

npm install

Далее, необходимо собрать сборку клиента командой:

npm build

После окончания сборки скопировать файлы из директории “web-build” на сервер по пути “/var/www/example.kg/client”

1. Развертывание Backend Электронного каталога

*Примечание: Для более удобного и быстрого развертывания можно настроить для проекта “CI/CD” (На данный момент приложение backend Электронного каталога настроена на “gitlab CI/CD”)*

После клонирования серверной части приложения Электронного каталога на локальный компьютер, нужно будет скопировать на сервер по пути “/var/www/example.kg/server”

Далее, установить зависимости “pip” из “requirements.txt” в корне директории проекта

- cd /var/www/example.kg/server/

- pip install -r requirements.txt

После завершения 4 шага Вы должны будете выполнить команды “restart” для сервисов “nginx” и “uwsgi”:

- service nginx restart

- service uwsgi restart