Vapor Server Commands

1. Install Swift

```
sudo apt update
sudo apt install clang libicu-dev build-essential pkg-config libz-dev
wget https://download.swift.org/swift-5.7.3-release/ubuntu2004/swift-5.7.3-RELEASE/swift-5.7.3-RELEASE-
ubuntu20.04.tar.gz
tar xzf swift-5.7.3-RELEASE-ubuntu20.04.tar.gz
sudo mkdir /swift
sudo mv swift-5.7.3-RELEASE-ubuntu20.04 /swift/5.7.3
sudo ln -s /swift/5.7.3/usr/bin/swift /usr/bin/swift
swift --version
```

2. Build Project & Install nginx, supervisor

```
git clone https://github.com/YuankaiZhu/Server_Example.git
cd Vapor_Server_Example/
swift build
swift run -v
sudo apt-get install supervisor
sudo apt-get install nginx
sudo vim /etc/supervisor/conf.d/Server_Example.conf
```

3. Set up supervisor config

```
[program: Vapor_Server_Example]
command=/home/vcm/Vapor_Server_Example/.build/debug/Run serve --env production
directory=/home/vcm/Vapor_Server_Example/
user=vcm
stdout_logfile=/var/log/supervisor/Server_Example-stdout.log
stderr_logfile=/var/log/supervisor/Server_Example-stderr.log
```

4. Set up nginx config

```
cd /etc/nginx/sites-enabled/
sudo cp default default_backup
sudo vim default
```

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
##server {
##  listen 80 default_server;
##  listen [::]:80 default_server;
```

```
  # SSL configuration
  #
  # listen 443 ssl default_server;
  # listen [::]:443 ssl default_server;
  #
  # Note: You should disable gzip for SSL traffic.
  # See: https://bugs.debian.org/773332
  #
  # Read up on ssl_ciphers to ensure a secure configuration.
  # See: https://bugs.debian.org/765782
  #
  # Self signed certs generated by the ssl-cert package
  # Don't use them in a production server!
  #
  # include snippets/snakeoil.conf;

##  root /var/www/html;

  # Add index.php to the list if you are using PHP
##  index index.html index.htm index.nginx-debian.html;

##  server_name _;

##  location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
##    try_files $uri $uri/ =404;
##  }

  # pass PHP scripts to FastCGI server
  #
  #location ~ \.php$ {
  # include snippets/fastcgi-php.conf;
  #
  # # With php-fpm (or other unix sockets):
  # fastcgi_pass unix:/run/php/php7.4-fpm.sock;
  # # With php-cgi (or other tcp sockets):
  # fastcgi_pass 127.0.0.1:9000;
  #}

  # deny access to .htaccess files, if Apache's document root
  # concurs with nginx's one
  #
  #location ~ /\.ht {
  # deny all;
  #}
##}


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
# listen 80;
# listen [::]:80;
#
# server_name example.com;
#
# root /var/www/example.com;
# index index.html;
#
# location / {
#   try_files $uri $uri/ =404;
# }
#}

server {
 server_name yz726.colab.duke.edu;
 listen 80;
 root /var/www/html;
 index index.html index.htm index.nginx-debian.html;
 try_files $uri @proxy;
 location @proxy{
 proxy_pass http://127.0.0.1:8080;
 proxy_pass_header Server;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_pass_header Server;
 proxy_connect_timeout 3s;
 proxy_read_timeout 10s;
 }

}
```

5. Install Postgresql Database & Vapor

```
sudo apt install postgresql postgresql-contrib
cd ~
git clone https://github.com/vapor/toolbox.git
cd toolbox/
git checkout 18.5.1
swift build -c release
sudo mv .build/release/vapor /usr/local/bin
```

6. Config Database

```
sudo -i -u postgres
psql
ALTER ROLE postgres WITH PASSWORD 'password';
CREATE DATABASE "Server_Example";
exit
exit
cd Server_Example/
vapor run migrate
^c
```

7. Reload supervisorctl

```
sudo supervisorctl reread
sudo supervisorctl add Vapor_Server_Example
sudo supervisorctl status
```

```
Server_Example         RUNNING  pid 12664, uptime 0:00:05
```

8. Reload nginx

```
sudo service nginx restart
sudo service nginx status
```

```
Active: active (running) since Thu 2023-03-09 19:28:10 UTC; 6s ago
```

9. Set up SSL for Web

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d yz726.colab.duke.edu

sudo supervisorctl restart Vapor_Server_Example
sudo service nginx restart
sudo service nginx status
```

10. Open our Web on Safari