HỆ ĐIỀU HÀNH: UBUNTU VERSION 18.04

Sau khi cấu hình xong thì nginx reload + restart thì mới ăn config: service nginx reload & service nginx restart

Có một lưu ý hết sức quan trọng, liên quan tới dữ liệu. Đó là việc mysql sẽ bị update khi chạy lệnh apt update.. gì đó, nó update lên version mới là 8, khi này migration laravel tạo cột sẽ không còn tương thích nữa và làm table bị lock hoàn toàn. Cách giải quyết lúc này

+ tắt server và mở lại

+ tắt apache, mở lại nginx và mysql (sudo service apache2 stop, sudo systemctl restart nginx, sudo service mysql restart)

+ tạo 1 table mới với cấu trúc tương tự như table cũ, lưu ý không tạo cột được sửa mới từ migration vì cột đó là nguyên nhân gây lỗi.

+ ta truy cập /var/lib/mysql/thư mục có tên là db của bạn rồi tìm đổi tên cái file ibd của table cũ sang tên nào đó để backup , tiếp đó ta chạy lệnh ALTER TABLE table\_name\_cu\_can\_xoa DISCARD TABLESPACE;

Rồi chạy tiếp ALTER TABLE table\_name\_mới tương ưng với file ibd mới được copy qua IMPORT TABLESPACE; lưu ý nếu bị lỗi table is missing space thì là do chưa đổi quyền sang mysq:mysql với file mới tạo, đổi lại là được

Sau đó xóa table cũ trong mysql bằng lệnh, vì nếu không xóa lúc dump lại db sẽ gây ra lỗi tiếp => lặp lại từ đầu và bỏ qua bước import data bằng ibd

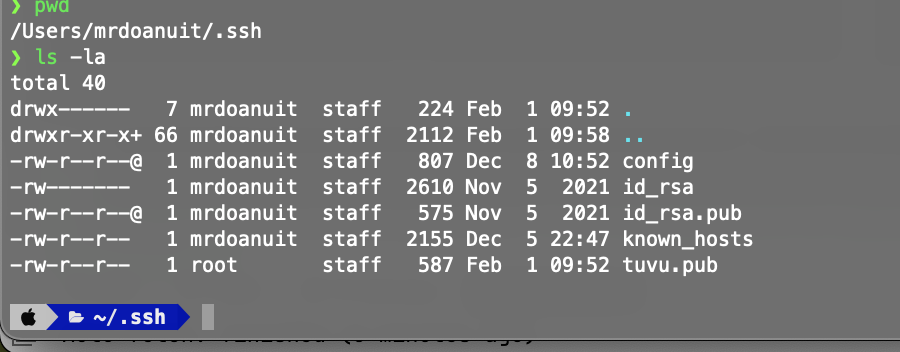
Sau khi đã cứu được data thì xóa mysql 8 đi, cài mysql 7, rồi chạy migrate, xong vô file sql đã dump ra rồi lấy data nhét vô, chú ý chỉ dump từng table, ko dump nguyên db vì cái table lỗi kia sẽ làm lock và đứng luôn system

Chú ý: bất cứ tương tác nào với table bị lỗi, kể cả xóa đều phát sinh lỗi, lúc này ta chỉ dump những table ko lỗi

+add ssh public key to access

Thủ công:  
sudo nano ~/.ssh/authorized\_keys => thêm public key vào

Public key lấy ở đây



Sử dụng lệnh: cat id\_rsa.pub để lấy content của file

+ Phần cấu hình upload max size gì gì đó, nó có mấy file config lận, nhiều file lắm nên check rồi sửa hết

**apache2** **cgi** **cli** **fpm** **mods-available**

/etc/php/7.2/cgi

Cả rổ ở trong á, update thôi, chỉnh xong thì phải restart server tương ứng thì mới ăn config

Nếu sử dụng linode vps, sau khi đã tạo xong machine thì tiến hành ssh bằng terminal để add signature key: ssh [root@103.3.60.240](mailto:root@103.3.60.240) thì mới có thể ssh qua ./ssh trên mac được(Nhớ add public key lúc tạo machine để bỏ qua bước nhập password)

+ Phần clone code từ git thì dùng ssh clone để khỏi phải nhập pass, chú ý 1 public key chỉ được phép dùng bởi 1 tài khoản git, nên là nếu mún add thêm thì tạo thêm public key của server

Git clone ssh://git@gitlab.beeknights.com:10022/root/chwm-outlet-and-production.git

Public key ở đây như là một chìa khóa cho phép access. Nếu muốn xác thực là ai thì phải add public key từ cái đó bỏ vô

+CÁC THỨ CƠ BẢN CẦN CÀI

* Php
* Mysql
* Supervisor
* Nginx
* Queue, job
* Config domain(trỏ qua machine) và domain trong nginx
* Có thể sẽ cần open remote với root account hoặc tạo 1 account khác để remote vào(Nên tạo 1 account khác)
* Cài đặt cert để có thể run https domain

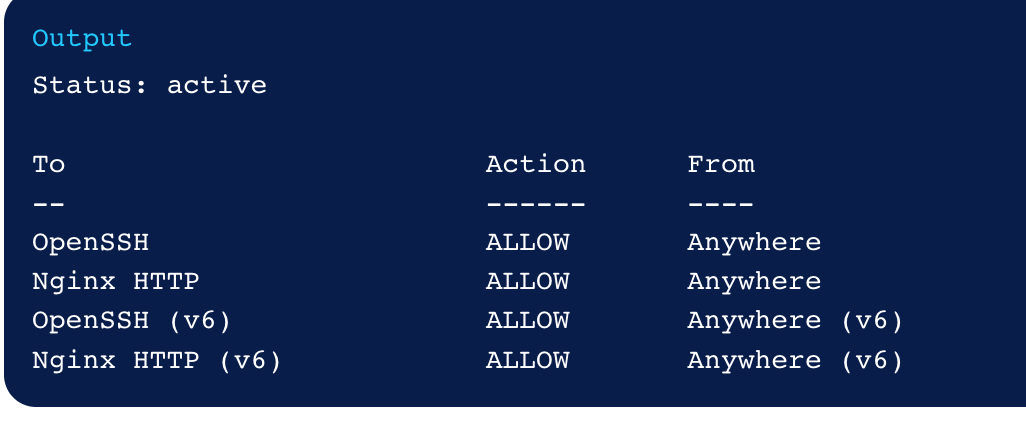
+UPDATE UBUNTU VIRTUAL SERVER  
 sudo apt update, sudo apt upgrade

+INSTALL NGINX  
 <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04>

sudo apt update, sudo apt install nginx

Adjusting the Firewall: sudo ufw app list

sudo ufw status => inactive => sudo ufw enable



Kích hoạt hết như cái list(xài `sudo ufw app list` để hiển thị cái cần kích hoạt)

Nếu muốn sử dụng một port để deploy mà không sử dụng domain thì phải mở port ra: sudo ufw allow 1701(1701 là port cần mở)

Nếu muốn đóng port thì: sudo ufw deny 22

Một số câu lệnh tương tác: sudo systemctl restart nginx | sudo systemctl stop nginx | sudo systemctl start nginx

-CÀI ĐẶT PHP 7.4

sudo apt-get update

sudo apt -y install software-properties-common

sudo add-apt-repository ppa:ondrej/php

sudo apt-get update

sudo apt -y install php7.4

sudo apt-get install -y php7.4-cli php7.4-json php7.4-common php7.4-mysql php7.4-zip php7.4-gd php7.4-mbstring php7.4-curl php7.4-xml php7.4-bcmath

sudo apt -y install php7.4-fpm php7.4-bcmath php7.4-curl php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xml php7.4-zip

sudo update-alternatives --config php => để chọn 1 version php vì đôi khi sẽ cài php 8

Lúc config nhớ kiểm tra path của php fpm

/var/run/php/php7.4-fpm.sock

-CÀI ĐẶT COMPOSER

(Cách 1)

(<https://linuxize.com/post/how-to-install-and-use-composer-on-ubuntu-18-04/>)

sudo apt install wget php-cli php-zip unzip

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

HASH="$(wget -q -O - <https://composer.github.io/installer.sig>)"

php -r "if (hash\_file('SHA384', 'composer-setup.php') === '$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"

sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer

alias composer='/usr/local/bin/composer'

(Cách 2)

sudo apt update && sudo apt install wget php-cli php-zip unzip curl

curl -sS https://getcomposer.org/installer |php

sudo mv composer.phar /usr/local/bin/composer

alias composer='/usr/local/bin/composer'

-CÀI ĐẶT MYSQL 5.7

sudo wget <https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb>

sudo dpkg -i mysql-apt-config\_0.8.15-1\_all.deb => Chọn bản 5.7 rồi chọn nút OK

sudo apt-get update

sudo apt-get install mysql-server

mysql\_secure\_installation

Tạo remote user

mysql> CREATE USER 'fosstechnix'@'%' IDENTIFIED BY 'fosstechnix@123';

GRANT ALL PRIVILEGES ON \* . \* TO 'fosstechnix'@'%';

FLUSH PRIVILEGES;

Mở remote mysql trên ubuntu

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf => đổi cái thông số của cái này thành như dưới: bind-address = 0.0.0.0

sudo systemctl restart mysql.service

Nhớ mở port(3306) để remote access vô => ufw allow 3306

Khi muốn access port nào thì phải mở port đó ra

Một số câu lệnh tương tác

sudo systemctl start mysql.service

sudo systemctl restart mysql.service

sudo systemctl stop mysql.service

-CÀI ĐẶT SUPERVISOR

Trong dự án sẽ có 2 chỗ sử dụng, đó là queue và schedule

crontab -e và chỉnh như ở dưới(sẽ run job mỗi phút vào project Laravel)

\* \* \* \* \* cd /var/www/speech-academy-laravel-backend && php artisan schedule:run >> /dev/null 2>&1

Setup worker: path: /etc/supervisor/conf.d in ubuntu

[program:speech-academy-laravel-backend-queue]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/speech-academy-laravel-backend/artisan queue:work

autostart=true

autorestart=true

user=www-data

numprocs=8

redirect\_stderr=true

stdout\_logfile=/var/www/speech-academy-laravel-backend/storage/logs/laravel.log

Phần cài đặt chi tiết:

sudo apt update

sudo apt install supervisor

sudo service supervisor restart

sudo service supervisor status

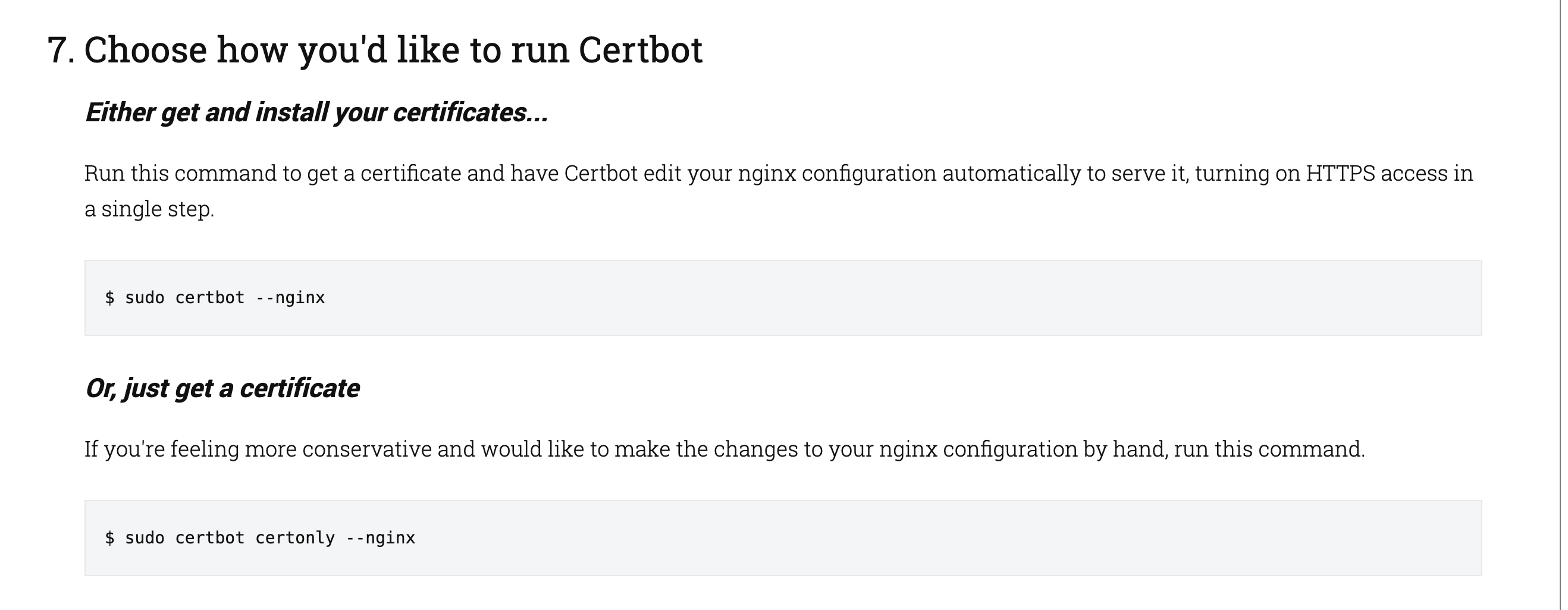
Cài đặt queue: <https://phambinh.net/bai-viet/cau-hinh-laravel-queue-tren-moi-truong-production/>

-CÀI ĐẶT CERT ĐỂ CẤU HÌNH HTTPS SSL

<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>

<https://certbot.eff.org/> Cài theo trang này có đầy đủ bộ cài

Nó có đưa 2 tùy chọn, 1 cái là tư động, 1 cái bằng cơm

Show list chứng chỉ: certbot certificates  


-CÀI ĐẶT NODE NVM

sudo apt update && sudo apt install curl -y

curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash

-CẤU HÌNH FILE NGINX

+Cấu hình với domain dạng port  
server {

listen 6152;//port này phải mở qua ufw để có thể access

root /var/www/chwm-outlet-and-production/public;

index index.php index.html index.htm index.nginx-debian.html;

location / {

try\_files $uri $uri/ /index.php?$query\_string;

}

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass unix:/var/run/php/php7.4-fpm.sock;

}

}

+Cấu hình với domain dạng link

server {

server\_name carrental.ftech.ltd;

client\_max\_body\_size 200M;

root /var/www/car-rental-laravel/public;

index index.php index.html index.htm index.nginx-debian,html;

location / {

try\_files $uri $uri/ /index.php$is\_args$args;

}

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass unix:/run/php/php8.2-fpm.sock;

}

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/carrental.ftech.ltd/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/carrental.ftech.ltd/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

if ($host = carrental.ftech.ltd) {

return 301 https://$host$request\_uri;

} # managed by Certbot

listen 80;

server\_name carrental.ftech.ltd;

return 404; # managed by Certbot

}