CSC 210/BIF 521 Analysis and Design of Algorithms | CSC218 Database Systems

An Overall step-by-step Infrastructure documentation

Thanadon Teeratadapong, Student ID: 61130500240,

Mail : thanadon.rh@mail.kmutt.ac.th

2019-11-05

Contents

1 Travelaloha-Infrastructure

1.1 Introduction

This Doumentation contains all efforts that were conducted in order to configure the server to deploy the project.

1.2 Summary

I was tasked with configuring the server to deploy our project as Infrastructure Technical Leader. The project is part of CSC 210/BIF 521 Analysis and Design of Algorithms and CSC218 Database Systems. The focus of this project is to put our theory into practice. My overall objective was to configure the network, and systems to be ready for the application.

I identified the requirement for the applications as follows.

- Nginx
  - SSL Certificate
- Node
  - PM2
- MySQL 8
- Jenkins

2 Install Node & PM2

2.1 Introduction / Rational

Debian 10 already comes with Nodejs version `10.15.2` via apt. But I wanted to use Nodejs version `12.x.x` because the V8 engine update. The newer version comes with fast `async/await` and `spread operators` implementation. This is where PPA comes to the rescue (Maintained by Nodesource).

2.2 Install Node Via PPA

```
cd ~ # Or your Working directory
curl -sL https://deb.nodesource.com/setup_12.x -o nodesource_setup.sh
sudo bash nodesource_setup.sh
sudo apt install nodejs build-essential
```

2.3 Install PM2 Via npm

PM2 is a daemon process manager that will help you manage and keep your application online.

```
sudo npm install pm2@latest -g
```

3 Install and Configure Nginx.

3.1 Introduction

Nginx is one of the most popular web servers in the world and responsible for hosting some of the largest and highest-traffic sites on the internet. It is more resource-friendly than Apache in most cases and can be used as a web server or reverse proxy. For our purpose, We're going to be using it in reverse proxy mode.

3.2 Installation

```
sudo apt update
sudo apt install nginx
```

3.3 Configure Nginx

We create a config file with these options

```
sudo nano /etc/nginx/sites-available/project
```

```
server {

  listen 80;
  listen [::]:80;
  server_name travelaloha.tech www.travelaloha.tech;
  return 301 https://$host$request_uri;
}
server {
```

```
  root /var/www/project;
  index index.php index.html index.htm;

  server_name travelaloha.tech www.travelaloha.tech;

  location / {

    proxy_pass http://localhost:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
  }
  location ~ \.php$ {

    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
  }
  location /phpmyadmin {

    root /usr/share/;
    index index.php index.html index.htm;
    location ~ ^/phpmyadmin/(.+\.php)$ {

      try_files $uri = 404;
      root /usr/share;
      fastcgi_pass unix:/run/php/php7.3-fpm.sock;
      fastcgi_index index.php;
      fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
      include /etc/nginx/fastcgi_params;
    }
    location ~* ^/phpmyadmin/(.+\.(jpg|jpeg|gif|css|png|js|ico|html|xml|txt))$ {

      root /usr/share/;
    }
  }
```

```
  listen [::]:443 ssl http2;
  listen 443 ssl http2;
  ssl_certificate /etc/letsencrypt/live/travelaloha.tech/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/travelaloha.tech/privkey.pem;
  include /etc/letsencrypt/options-ssl-nginx.conf;
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;


}
```

Then we need to enable this config

```
sudo ln -s /etc/nginx/sites-available/project /etc/nginx/sites-enabled/
sudo nginx -t
# To check if our config is valid.
#Don't Worry if there is an error. Complete all other steps first.
sudo systemctl restart ngnix
```

4 Adding SSL Certificate using Certbot

Certbot is my choice to setup SSL Certificate for the website. Certbot is a free, open source software tool for automatically using Let's Encrypt certificates on manually-administrated websites to enable HTTPS.

4.1 Install Certbot

```
sudo apt-get update
sudo apt-get install certbot python-certbot-nginx
sudo certbot --nginx
```

Enter Your information. Select No `redirect` when prompted to redirect HTTP to HTTPS, Since we will be configuring that in the Nginx Setting.

4.2 Footnotes / Important Details

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/travelaloha.tech/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/travelaloha.tech/privkey.pem
   Your cert will expire on 2020-02-03. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
```

5 MySQL 8 Installation

5.1 Introduction / Rational

MySQL is a prominent open source database management system used to store and retrieve data for a wide variety of popular applications. My DB TechLead wanted MySQL version 8+. So Installing it the normal way wouldn't work.

5.2 Installation

We need to install the prerequisite GnuPG package, an open-source implementation of the OpenPGP standard.

```
sudo apt update
sudo apt install gnupg
```

Next visit this site `https://dev.mysql.com/downloads/repo/apt/` Click *Download* Then copy the link from *No thanks, just start my download.*

```
cd ~ # Or your Working directory
wget The_link_you_copied
#wget https://dev.mysql.com/get/mysql-apt-config_0.8.13-1_all.deb
sudo dpkg -i mysql-apt-config* # Select default options
sudo apt update
sudo apt install mysql-server
```

Enter root password when prompted

Next we need to secure MySQL

```
mysql_secure_installation
```

Select no for ***validate password plugin***. Answering yes for all the options.

6 Deploying the app

6.1 First we need to clone the app.

```
git clone https://github.com/CS19-SIT/travel-aloha.git /var/www/project
```

6.2 Next, we need to install all the project dependencies.

```
cd /var/www/project
npm i
sudo chmod -R 755 ./*
```

6.3 After that, We need to setup our Environment File

```
cp .env.example .env
```

edit `.env` with your editor of choice. I will be using nano

```
nano .env
```

```
APP_PORT=3000
DB_HOST="WWW.YOUR_WEBSITE.EXT"
DB_PORT=3306
DB_USER="YOUR_USER"
DB_PASSWORD="YOURSECRETPASSWORD"
DB_DATABASE="Development"
SESSION_KEY="YOUR_VERY_IMPORTANT_SESSION_KEY"
SESSION_PASSWORD="YOUR_VERY_IMPORTANT_SESSION_KEY_PASSWORD"
```

6.4 Running the App with PM2.

```
pm2 start npm -- start -i max
pm2 save
pm2 startup #Select systemd if prompted
```

7 Installing Jenkins

7.1 Introduction

Jenkins is an open-source tool that tests and compiles the code. Jenkins is used to minimizes the testing time and automates the deployment of new commits from github.

7.2 We need to install Java.

```
sudo apt-get update
sudo apt-get install openjdk-11-jre-headless
```

7.3 Add the Jenkins key to apt.

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add
```

7.4 Creates the source list for Jenkins.

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sour
```

7.5 Install Jenkins

```
sudo apt-get update
sudo apt-get install jenkins
```

7.6 Complete Jenkins Setup

Go to http://35.247.178.19:8080/ The initial password is in

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

7.7 Create new Item in Jenkins

7.7.1 Create a Freestyle Project

7.7.2 In General

Select Github project. Add Project URL.

```
https://github.com/CS19-SIT/travel-aloha
```

7.7.3 In Source code Management

Select Git. Add Repository URL.

```
https://github.com/CS19-SIT/travel-aloha.git
```

Add Credentials. Add Branch Specifier.

7.7.4 In Build Triggers

Select Github hook trigger for GITScm polling.

7.7.5 In Build

Select Add build step. Select Execute shell. Add in desired commands, For this project it will be

```
/bin/bash
cd /var/www/project
git pull origin master
npm i
sudo pm2 restart 0
```

7.8 Post Setup

normally Jenkins user does not have permission in most of the system folders. We need to change this by giving sudoers group to Jenkins user.

```
sudo visudo
```

Navigate to the end of the file, Then add.

```
jenkins ALL=(ALL) NOPASSWD: ALL
```

8 Additional Items

8.1 References:

https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-debian-10

https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-debian-10

https://medium.com/@darkrubyist/how-to-fix-git-push-error-insufficient-permission-for-adding-an-object-to-repository-database-git-53d7dc9649e2

https://certbot.eff.org/lets-encrypt/debianbuster-nginx

https://www.digitalocean.com/community/tutorials/how-to-install-the-latest-mysql-on-debian-10

https://www.digitalocean.com/community/tutorials/how-to-install-phpmyadmin-from-source-debian-10