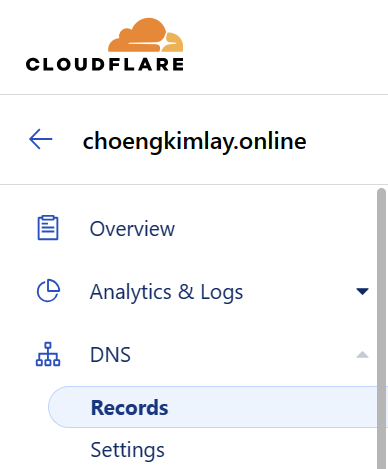
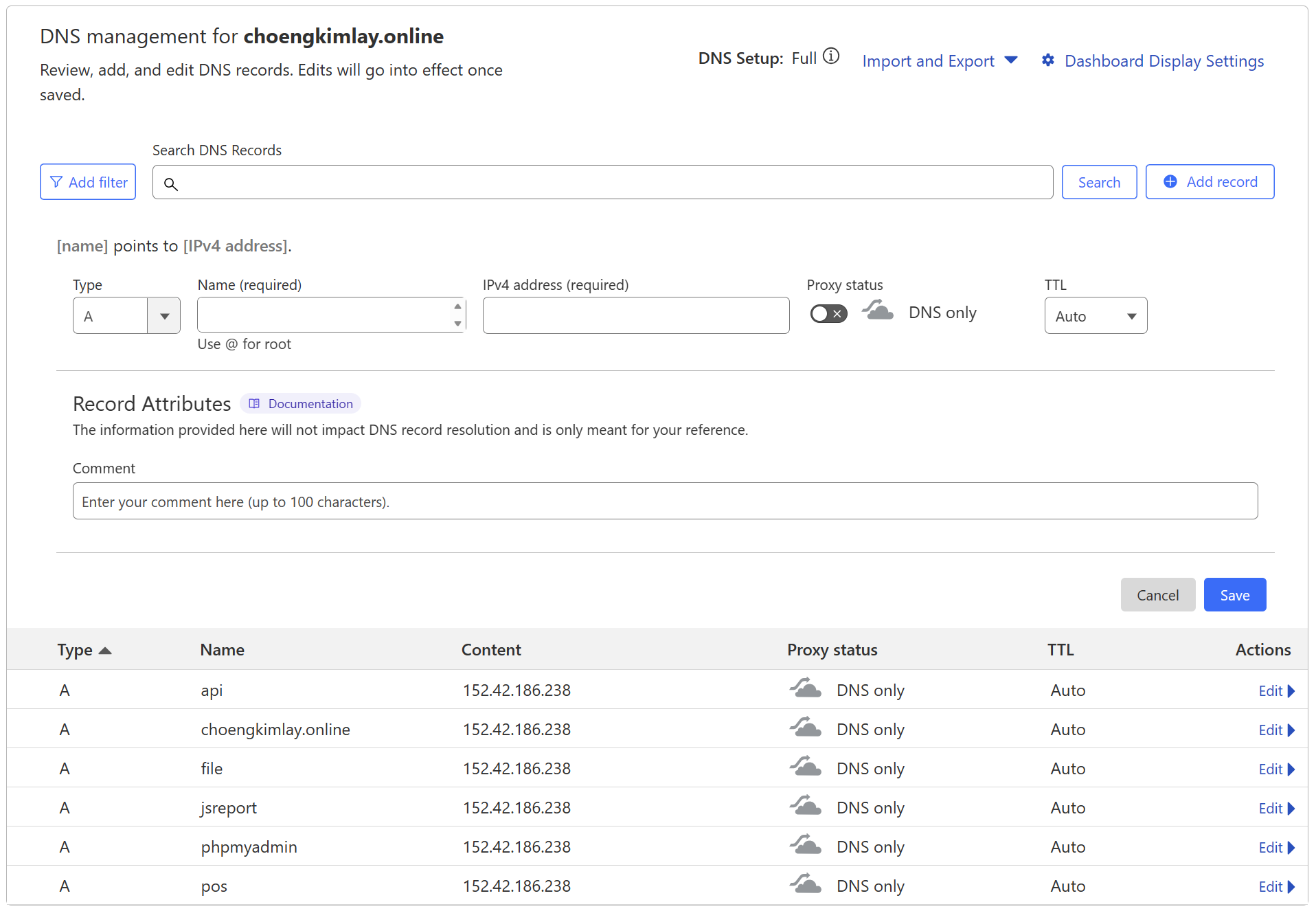
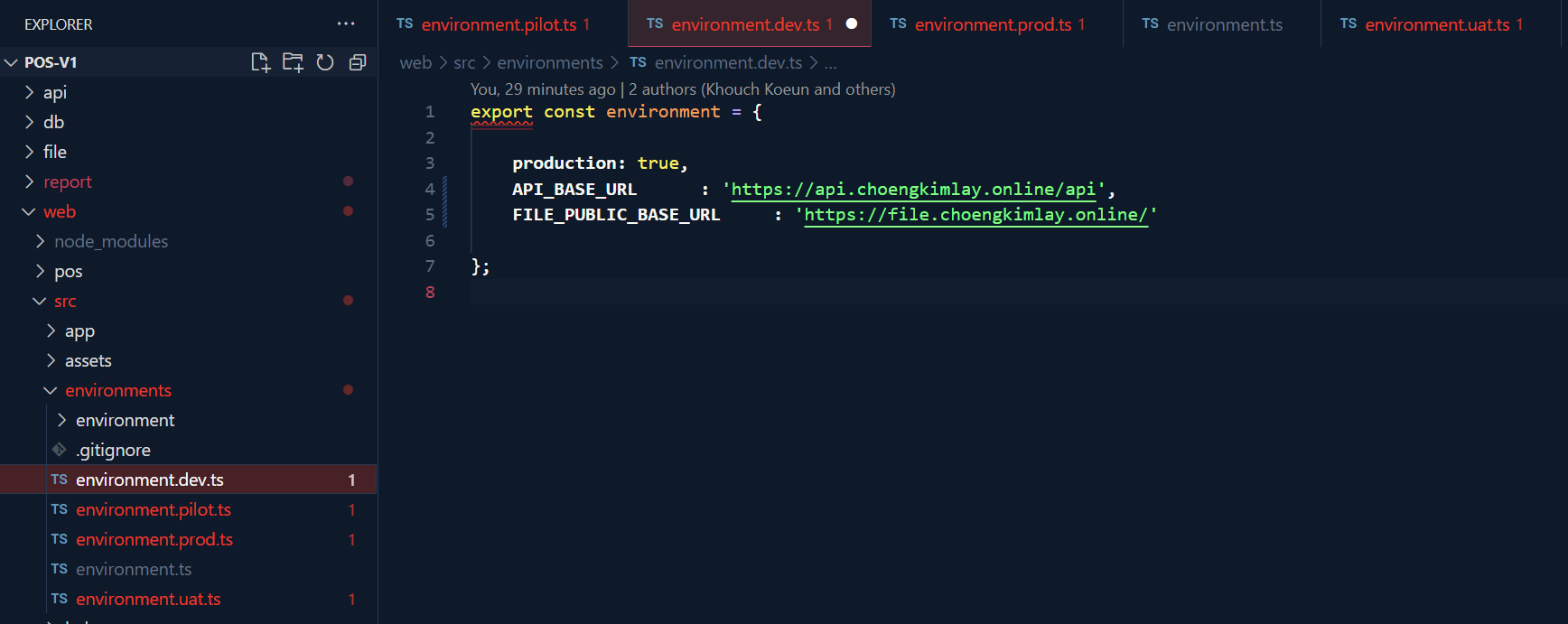
**Go to Cloudflare and go to DNS / Records**

**Add Record as seen below. Turn off Proxy Status. Change choengkimlay.online to your own domain.**

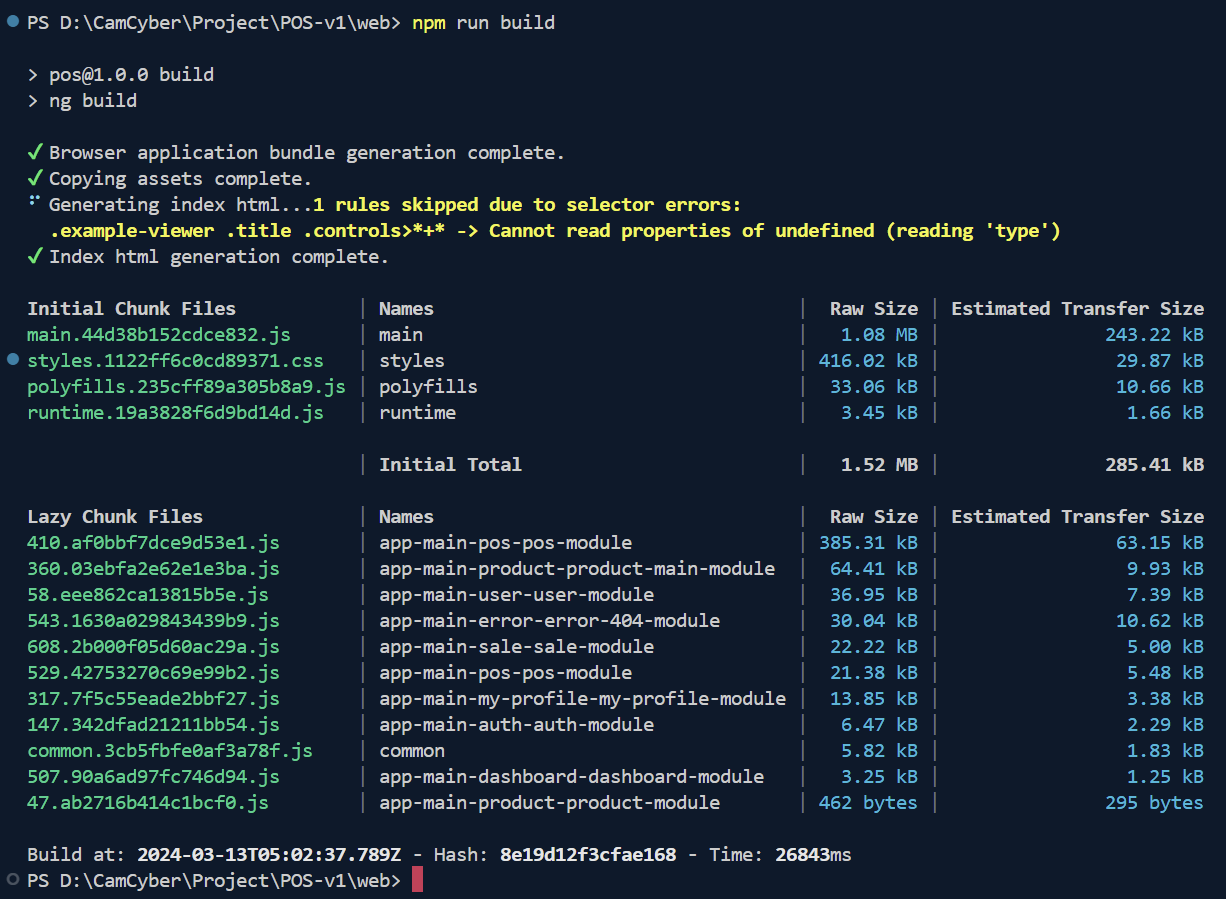


**Go to web and change environment of all 5 environment files as below. Make sure it is https and change choengkimlay.online to your own domain name.**

**Delete the POS folder (if exist) and run command npm run build**



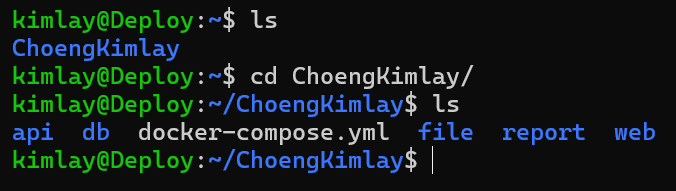
**Then push web to gitlab**



**ssh to server**

**Note: need to ask Hiek Lymonyratanak for ssh key to server**

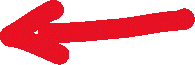


**create folder with your name and clone file, api, web of your own upgrading-name branch**

* **sudo nano docker-compose.yml (password is 123)**

**Copy the docker-compose below but Read and Change the comments I put in. I also attach the docker-compose file as well.**

****



version: "3.8"

services:

*#==============================================>> Database*

    db:

        image: mysql:5.7.36

        container\_name: !!-pos-db-v1 *# change !! to anything so that there is no container name conflict (ex ck.pos.db-v1)*

        restart: unless-stopped

        command: --default-authentication-plugin=mysql\_native\_password

        tty: true

        ports:

            - "!!!6:3306" *# change !!! to the first three digits of the ports you are assigned*

        environment:

            MYSQL\_DATABASE: homestead

            MYSQl\_USER: homestead

            MYSQL\_PASSWORD: secret

            MYSQL\_ROOT\_PASSWORD: secret

            SERVICE\_TAGS: dev

            SERVICE\_NAME: mysql

        volumes:

            - ./db/mysql:/var/lib/mysql

        networks:

            - v1

*#==============================================>> PhpMyAdmin*

    db-tool:

        image: phpmyadmin/phpmyadmin

        container\_name: !!-pos-pma-v1 *# change !! to anything so that there is no container name conflict (ex ck-pos-pma-v1)*

        restart: always

        environment:

            PMA\_HOST: db

            PMA\_PORT: 3306

            PMA\_ARBITRARY: 1

            MYSQL\_ROOT\_PASSWORD: secret

        ports:

            - ~~!!!!:80~~ *# change !!!! to your pma port that is assigned to you*

        networks:

            - v1

*#==============================================>> API*

    api-v1.1:

        container\_name: !!-pos-api-v1.1 *# change !! to anything so that there is no container name conflict (ex ck-pos-api-v1.1)*

        restart: unless-stopped

        tty: true

        build:

            context: ./api/.

            dockerfile: Dockerfile

        volumes:

            - ./api/app:/var/www/app

            - ./api/resources:/var/www/resources

            - ./api/routes:/var/www/routes

            - ./api/storage:/var/www/storage

            - ./api/database:/var/www/database

        command: php artisan --host=0.0.0.0 serve

        ports:

            - "!!!!:8000" *# change !!!! to your api port that is assigned to you*

        environment:

*#===========================>> App Info*

            APP\_KEY: base64:mNe36k6bz1cidhh1dLtMtjRsiMnMOyjztjti7a6rAf4=

*#===========================>> JWT*

            JWT\_SECRET: MTO7OmBzXhbPI2bErLjV9XEFbpCn3MHCPxKdcBxiFHjuWJirIayTB9npa4cGNAFX

*#==========================>> DB Config*

            DB\_CONNECTION: mysql

            DB\_HOST: !!-pos-db-v1 *# change !! to the same as in the db section*

            DB\_PORT: 3306

            DB\_DATABASE: minimart

            DB\_USERNAME: root

            DB\_PASSWORD: secret

*#==========================>> FILE*

            FILE\_BASE\_URL: http://file-v1.1:8000

*#==========================>> JS Report*

            JS\_BASE\_URL : !!-pos-report-v1:5488 *# change !! to the same as in the jsreport section*

            JS\_USERNAME : admin

            JS\_PASSWORD : secret

*#=========================>> Telegram Bot*

*# TELEGRAM\_BOT\_TOKEN: 6288151605:AAH6UsaVEhpzxgVWdQ7ZEJp1yFFz0e8HMtg*

*# TELEGRAM\_CHAT\_ID: -1002142257134*

*# you can use your own bot or use mine*

*# this is the telegram link*

*# https://t.me/+arf8lYDf4aRhYWE1*

*#===========================>> Mail*

            MAIL\_MAILER         : ''

            MAIL\_HOST           : ''

            MAIL\_PORT           : ''

            MAIL\_USERNAME       : ''

            MAIL\_PASSWORD       : ''

            MAIL\_ENCRYPTION     : ''

            MAIL\_FROM\_ADDRESS   : ''

            MAIL\_FROM\_NAME      : ''

        networks:

            - v1

*#==============================================>> File*

    file-v1.1:

        container\_name: !!-pos-file-v1 *# change !! to anything so that there is no container name conflict*

        build:

            context: ./file/.

            dockerfile: Dockerfile

        volumes:

            - ./file/public:/var/www/public

        command: php artisan --host=0.0.0.0 serve

        ports:

            - "!!!!:8000" *# change !!!! to your file port that is assigned to you*

        environment:

*#===========================>> App Info*

            APP\_KEY: base64:mNe36k6bz1cidhh1dLtMtjRsiMnMOyjztjti7a6rAf4=

        networks:

            - v1

*#==============================================>> Report Generator*

    report:

        image: jsreport/jsreport

        container\_name: !!-pos-report-v1 *# change !! to anything so that there is no container name conflict*

*# volumes:*

*#     - ./report/data:/app/data     #delete volumes*

        ports:

            - "!!!!:5488" *# change !!!! to your jsreport port that is assigned to you*

        networks:

            - v1

*#==============================================>> Web*

    web:

        container\_name: !!-pos-web-v1 *# change !! to anything so that there is no container name conflict*

        build:

            context: ./web/.

            dockerfile: Dockerfile

        ports:

            - "!!!!:80" *# change !!!! to your web port that is assigned to you*

        networks:

            - v1

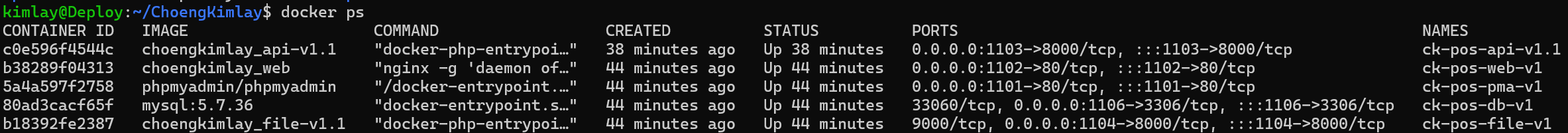
networks:

    v1:

* **docker-compose up -d**

**(make sure it is “docker-compose” and not “docker comopose”)**

* **docker ps**

****

**Migrate and seed**

* **docker exec -it (Container ID) bash**
* **php artisan migrate (To create minimart database)**
* **php artisan db:seed**

**make sure it is running in the correct port**

* **cd /etc/nginx/conf.d**
* **sudo nano phpMyAdmin.domain\_name.conf**

**(ex phyMyAdmin.choengkimlay.online.conf)**

server {

    listen 80;

    server\_name phpMyadmin.domain.name; *# change domain.name to your domain name*

    location / {

        proxy\_connect\_timeout 140;

        proxy\_send\_timeout 180;

        proxy\_read\_timeout 180;

        proxy\_set\_header Host $host;

        proxy\_pass http://localhost:!!!!; *# change to your phpMyAdmin port*

    }

    access\_log /var/log/nginx/phpMyadmin.choendgkimlay.online\_access.log; *# change choengkimlay.online to your domain name*

    error\_log /var/log/nginx/phpMyadmin.choengkimlay.online\_error.log; *# change choengkimlay.online to your domain name*

}

* **sudo nano pos.domain\_name.conf**

**(pos.choengkimlay.online.conf)**

server {

    listen 80;

    server\_name pos.domain.name; *# change domain.name to your domain name*

    location / {

        proxy\_connect\_timeout 140;

        proxy\_send\_timeout 180;

        proxy\_read\_timeout 180;

        proxy\_set\_header Host $host;

        proxy\_pass http://localhost:!!!!; *# change to your web port*

    }

    access\_log /var/log/nginx/pos.domain.name\_access.log; *# change domain.name to your domain name*

    error\_log /var/log/nginx/pos.domain.name\_error.log; *# change domain.name to your domain name*

}

* **sudo nano api.domain\_name.conf**

**(api.choengkimlay.online.conf)**

server {

    listen 80;

    server\_name api.domain.name; *# change domain.name to your domain name*

    location / {

        proxy\_connect\_timeout 140;

        proxy\_send\_timeout 180;

        proxy\_read\_timeout 180;

        proxy\_set\_header Host $host;

        proxy\_pass http://localhost:!!!! *# change to your file port*

    }

    access\_log /var/log/nginx/file.domain.name\_access.log; *# change domain.name to your domain name*

    error\_log /var/log/nginx/file.domain.name\_error.log; *# change domain.name to your domain name*

}

* **sudo nano jsreport.domain\_name.conf**

**(api.choengkimlay.online.conf)**

server {

    listen 80;

    server\_name jsreport.domain.name; *# change domain.name to your domain name*

    location / {

        proxy\_connect\_timeout 140;

        proxy\_send\_timeout 180;

        proxy\_read\_timeout 180;

        proxy\_set\_header Host $host;

        proxy\_pass http://localhost:!!!!; *# change to your jsreport port*

    }

    access\_log /var/log/nginx/jsreport.domain.name\_access.log; *# change domain.name to your domain name*

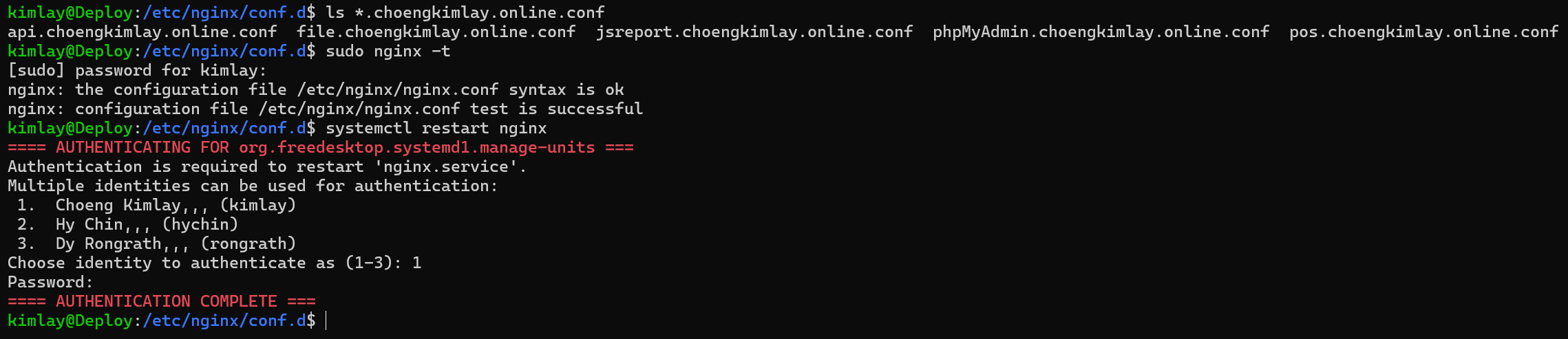
    error\_log /var/log/nginx/jsreport.domain.name\_error.log; *# change domain.name to your domain name*

}

**When you have all five files**

**Run these command**

* **sudo nginx -t**
* **systemctl restart nginx**

****

**After that run these command to Configure from HTTP to HTTPs**

* **sudo apt update**
* **sudo apt install certbot python3-certbot-nginx**
* **sudo certbot --nginx -d phpMyAdmin.domain.name pos.domain.name api.domain.name file.domain.name jsreport.domain.name**

**(change domain.name to your domain name)**

**Done!!**

**That should be it. Everything should be running. You can try to open on browser now.**