

Trustroots Production Setup (updated 2021-07-25)

This setup procedure is based on with [Server.md](#) and with Passenger as part of setup. This was created in the effort to create a federated hospitality system under the [Open Hospitality Network](#) project.

System Setup

Setup a new Ubuntu 18.04 server with 4gb RAM. Non-root user below is called dsterry.

Run the following commands to setup a user and login via ssh from that user, just as you can to root (if keys were added during OS install by the host)

Source: <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04>

```
adduser dsterry
usermod -aG sudo dsterry
ufw allow OpenSSH
ufw enable
rsync --archive --chown=dsterry:dsterry ~/.ssh /home/dsterry
```

Install Nginx

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

```
sudo apt install nginx
sudo ufw allow 'Nginx HTTP'
```

Passenger

Now we setup Passenger by running these commands one at a time.

Source: <https://www.phusionpassenger.com/library/install/nginx/install/oss/bionic/>

```
sudo apt-get install -y dirmngr gnupg
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
561F9B9CAC40B2F7
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates
sudo sh -c 'echo deb https://oss-binaries.phusionpassenger.com/apt/passenger bionic
main > /etc/apt/sources.list.d/passenger.list'
sudo apt-get update
```

```
if [ ! -f /etc/nginx/modules-enabled/50-mod-http-passenger.conf ]; then sudo ln -s
/usr/share/nginx/modules-available/mod-http-passenger.load
/etc/nginx/modules-enabled/50-mod-http-passenger.conf ; fi
```

```
sudo ls /etc/nginx/conf.d/mod-http-passenger.conf
```

And you should see a file listed there. If not, make an issue in the [fedi-trustroots](#) repo.

```
sudo service nginx restart
sudo /usr/bin/passenger-config validate-install
```

If there are any errors, follow the onscreen directions.

```
sudo /usr/sbin/passenger-memory-stats
```

And you should see some nginx and Passenger processes. If not, something is wrong.

Replace `subdomain.example.com` with your domain and do the following to create a server block for your domain.

```
export DOMAIN=subdomain.example.com
sudo mkdir -p /var/www/$DOMAIN/html
sudo chown -R $USER:$USER /var/www/$DOMAIN/html
sudo mkdir -p /var/www/$DOMAIN/ft
sudo chown -R $USER:$USER /var/www/$DOMAIN/ft
sudo chmod -R 755 /var/www/$DOMAIN
nano /var/www/$DOMAIN/html/index.html
```

The last line opens a file, paste in the following html and save it.

```
<html>
  <head>
    <title>Welcome to Example.com!</title>
  </head>
  <body>
    <h1>Success! The example.com server block is working!</h1>
  </body>
</html>
```

The next line opens a file. Paste in the configuration that follows, swap your domain in for `example.com` in three places and save the file.

```
sudo nano /etc/nginx/sites-available/$DOMAIN
```

```
server {
    listen 80;
    listen [::]:80;
```

```
    root /var/www/example.com/html;
    index index.html index.htm index.nginx-debian.html;

    server_name example.com www.example.com;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

```
sudo ln -s /etc/nginx/sites-available/$DOMAIN /etc/nginx/sites-enabled/
sudo nano /etc/nginx/nginx.conf
```

Remove the # to uncomment the line that says `server_names_hash_bucket_size 64;`

Test your nginx config with

```
sudo nginx -t
```

Check the result to make sure your config is ok, then

```
sudo systemctl restart nginx
```

Add the following in the http section of `/etc/nginx/nginx.conf`

```
# passenger_ruby /usr/bin/passenger_free_ruby;
passenger_nodejs /usr/local/bin/node;
passenger_show_version_in_header off;
```

SSL

Next we'll enable SSL using a Let's Encrypt certificate

Source:

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

Note: This ppa method is deprecated in 18.04.5 LTS

```
sudo add-apt-repository ppa:certbot/certbot
sudo apt install python-certbot-nginx
sudo ufw allow 'Nginx Full'
sudo ufw delete allow 'Nginx HTTP'
sudo certbot --nginx -d $DOMAIN -d www.$DOMAIN
```

Fedi-trustroots

Next cd to your `/var/www/subdomain.domain.tld` directory and clone the repo
`git clone https://github.com/OpenHospitalityNetwork/fedi-trustroots.git` ft

Install nvm from the nvm-sh repo. It's good practice to visit the repo and make sure there are no security incidents that might have compromised this script.

Source: <https://github.com/nvm-sh/nvm#installing-and-updating>

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
nvm install node
```

Install MongoDB

Source:

<https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-18-04-source>

```
curl -fsSL https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
bionic/mongodb-org/4.4 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-4.4.list
```

Continuing to setup per <https://team.trustroots.org/Install.html>

```
sudo apt update
sudo apt install mongodb-org
sudo systemctl start mongod.service
sudo systemctl enable mongod
```

For production it is also recommended to further secure mongodb as described here:
<https://www.digitalocean.com/community/tutorials/how-to-secure-mongodb-on-ubuntu-18-04>

This part requires 4gb of ram.

Todo: Check if we can skip npm ci and do npm run build instead.

```
sudo apt install make build-essential
npm ci
```

Add "port: 3001," to config/env/production.js because webpack:server proxies from port 3000 to 3001.

Compare your nginx config to this one and make any necessary changes:
<https://gist.github.com/weex/add4a96da52be1ca32e9698ce713b366>

Run this command once to build assets.

```
NODE_ENV=production npm run build
```

Then run these two in separate shells

```
NODE_ENV=production npm run start:prod
```

```
NODE_ENV=production npm run webpack:server
```

At this point you should be able to visit the site!

Todo: Work through API issues

Todo: Ensure system comes back up on boot

Todo: Document common maintenance tasks

If you have any problems, please make an issue at [fedi-trustroots](https://github.com/fedi-trustroots).