

First of all you need to follow this complete guide:

<https://coderrocketfuel.com/article/deploy-a-create-react-app-website-to-digitalocean>

Then, you can start configuring your react app:

1 - Change your **package.json**, add **homepage** variable and **make it empty**

```
{  
  "name": "example app",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    ...  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test",  
    "eject": "react-scripts eject"  
  },  
  "homepage": "",  
}
```

2 - npm run build and deploy to digital ocean

3 - now we need to change configurations for nginx

```
sudo nano /etc/nginx/sites-enabled/default
```

Then change location field just like on the screenshot

```
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/smartdeals.uz/html;

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

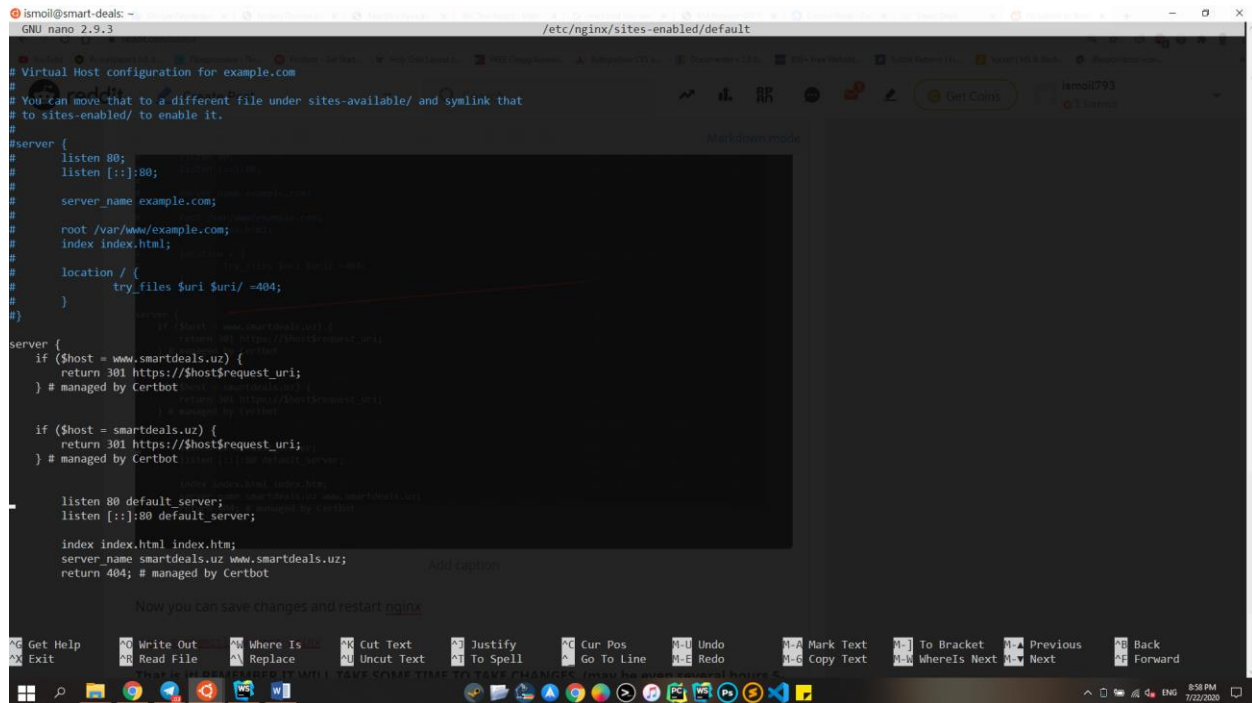
    server_name smartdeals.uz www.smartdeals.uz;

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

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #
    #    # With php-fpm (or other unix sockets):
    #    fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    #    # With php-cgi (or other tcp sockets):
```

location

There is 1 more server object inside of default file, it should be similar to



```
GNU nano 2.9.3 /etc/nginx/sites-enabled/default
# 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 {
    listen 80 default_server;
    listen [::]:80 default_server;

    index index.html index.htm;

    server_name smartdeals.uz www.smartdeals.uz;
    return 404; # managed by Certbot
}
```

Now you can save changes and restart nginx

`sudo systemctl restart nginx`

REMEMBER IT WILL TAKE SOME TIME TO TAKE CHANGES

HOW TO OPTIMIZE YOUR files img,css and etc. and add it for caching to speed up your app ?

Add the following to sites-enabled default parameter of nginx:

```
ismoil@smart-deals: ~  
GNU nano 2.9.3 /etc/nginx/sites-enabled/default  
# Add index.php to the list if you are using PHP  
index index.html index.htm index.nginx-debian.html;  
server_name smartdeals.uz www.smartdeals.uz;  
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    try_files $uri $uri/ /index.html;  
}  
location ~* \.(jpg|jpeg|gif|png|svg)$ {  
    expires 365d;  
}  
location ~* \.(pdf|css|html|js|swf)$ {  
    expires 2d;  
}  
# pass PHP scripts to FastCGI server  
#  
#location ~ \.php$ {  
#    include snippets/fastcgi-php.conf;  
#  
#    # With php-fpm (or other unix sockets):  
#    fastcgi_pass unix:/var/run/php/php7.0-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;  
#}  
listen [::]:443 ssl ipv6only=on; # managed by Certbot  
listen 443 ssl; # managed by Certbot  
ssl_certificate /etc/letsencrypt/live/smartdeals.uz/fullchain.pem; # managed by Certbot  
ssl_certificate_key /etc/letsencrypt/live/smartdeals.uz/privkey.pem; # managed by Certbot  
Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket Previous Back  
Exit Read File Replace Uncut Text To Spell Go To Line Redo Copy Text WhereIs Next Next Forward
```