

# Run an index.html on an new installation of Nginix

To run an index.html file on a Nginx web server, you can follow these steps:

Copy your index.html file to the Nginx web server's root directory. By default, this directory is **/var/www/html/**.

```
sudo cp /path/to/index.html /var/www/html/
```

Open the Nginx default configuration file **/etc/nginx/sites-available/default** using a text editor such as nano:

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

Locate the server block in the configuration file, which typically starts with `listen 80 default_server;`. This block defines the server's default behavior for incoming requests.

If there is already a `root` directive in the server block, make sure it points to the `/var/www/html` directory:

```
root /var/www/html;
```

If there is no `root` directive in the server block, add it as follows:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
    index index.html;  
  
    # other directives ...  
}
```

Save and close the file.

Restart the Nginx service to apply the changes:

```
sudo systemctl restart nginx
```

Now, if you open a web browser and enter your server's IP address in the address bar, Nginx should serve the index.html file from the `/var/www/html` directory. If you want to use a different file name or location, make sure to update the `root` and `index` directives in the Nginx configuration file accordingly.

# Steps to run the file on a separate subdomain

To run an index.html file on a subdomain using Nginx, you can follow these steps:

Create a new server block file in the `/etc/nginx/sites-available/` directory. You can use a different file name that represents your subdomain. For example, if your subdomain is `example.com`, you can create a file called `example.com`:

```
sudo nano /etc/nginx/sites-available/example.com
```

In the file, add the following configuration to define your server block. Replace `example.com` with your subdomain name and `/path/to/index.html` with the file path to your `index.html` file:

```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name example.com;  
  
    root /path/to/  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ /index.html;  
    }  
}
```

This configuration sets up a server block that listens on port 80 for requests to `example.com`. The `root` directive specifies the directory where the `index.html` file is located. The `index` directive specifies the default file name to serve when a request is made to the server. The `location` block adds a rule to try serving `index.html` if the requested resource is not found.

Create a symbolic link from the `sites-available` directory to the `sites-enabled` directory to enable the new server block:

```
sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/
```

Test the Nginx configuration to make sure there are no syntax errors:

```
sudo nginx -t
```

If there are no errors, restart the Nginx service to apply the changes:

```
sudo systemctl restart nginx
```

Now, if you enter your subdomain's URL in a web browser, Nginx should serve the index.html file from the specified directory. If you want to use a different file name or location, make sure to update the root and index directives in the Nginx configuration file accordingly.

---

Revision #3

Created 20 February 2023 16:05:10 by Daniel Azimi

Updated 3 June 2023 06:28:49 by Daniel Azimi