**Problem 1: Why was “/static” not working?**

**[loadbaancing file]**

**Location:**

Configure based on provided URI.

=: exactly provided value

~: case insensitive

[0-9]: regular exp range

+regular expressions as well

+you can add extensions

Root and alias are to direct to the file based on the specified path!

From the documentation:

A path to the file is constructed by merely adding a URI to the value of the root directive. If a URI has to be modified, the [alias](https://nginx.org/en/docs/http/ngx_http_core_module.html#alias) directive should be used.

If an alias is used inside a location defined with a regular expression then such regular expression should contain captures and the alias should refer to these captures.

<https://stackoverflow.com/questions/10631933/nginx-static-file-serving-confusion-with-root-alias>

root

* the location part is appended to the root part
* final path = root + location

alias

* the location part is replaced by the alias part
* final path = alias

So, the final path for root when /static is used will be :

/usr/share/nginx/html/static which doesn’t exist!

But the alias will just drop the location! = similar to symbolic link!

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem2: Learn: restart, rewrite and reload ❓**

<https://phoenixnap.com/kb/nginx-start-stop-restart>

[Use listen as listenn on any file and rewrite file]

Use the welcome page

Restart it with an error

Then reload it with an error

The restart will shut the server and then restart it again

Reload will not make any changes since there is an error and hence, whatever existed to the public will still stay the same.

So be sure of what you are trying to do!

And nginx -t will give you the line @ which the error exists

Exception and try out? I've experienced a case where I needed to restart nginx to listen to directives for a new IP address to kick in.

**Rewrite:**

301 - to redirect - but goes to another page;

To avoid this and get it done on the same page: we use rewrite

Rewrite /homepage /app1

Rewrite /homepage1 /profile

Location /profile

{ return 200 “ message “; }

**Redirect: 301 will cause moving to another page from the page to avoid this, use rewrite**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROBLEM3: NGINX 403 forbidden**

If your server is on Nginx, you wil have the following message: “403 Forbidden – nginx”. On Nginx, no *.htaccess* file.

NGINX 403 Forbidden is an HTTP error response message indicating that a client is not allowed to see the requested page. This is most commonly caused by a server-side issue, such as incorrect permissions or an incorrect index file.

**Incorrect Index File**

The most common cause of an NGINX 403 Forbidden error on a website is an incorrect index file. When a user visits a URL of the website, the system looks for a file named “index.html” in the requested directory or folder. If no such file is found, the server will return a 403 Forbidden error status.

To fix this issue, ensure that an appropriate index file exists in the requested directory/folder.

**Incorrectly Set Permissions**

Another common cause of an NGINX 403 Forbidden error is incorrectly set permissions. All directories within the web server’s directory should have the correct permissions set.

The user running the web server must have read and execute permissions. If these permissions are not set correctly, the system will return a 403 Forbidden error. To fix this issue, ensure that the permissions are set correctly for the user running the web server.

**Client-Side Cause of Error 403**

Finally, it is possible that the client side is causing the 403 Forbidden error. A few possible causes of this issue include:

* a security plugin,
* an outdated browser,
* or an issue with the cache.

To fix this issue, try disabling any security plugins, [clearing the browser’s cache](https://www.debugbar.com/how-clear-cache-internet-explorer/), or updating it to the latest version.

**Page not Found: 404**

The 404 Not Found message tells us that the document root of the website can not be found although the visitor's web browser is connected successfully to our server

You can use try\_files to display a 404 error instead;

$uri - whatever URI you give in the search bar is looked and if not present, 404 is raised

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROBLEM4: Worker connections and processors**

| Syntax: | **worker\_connections *number*;** |
| --- | --- |
| **Default:** | **worker\_connections 512;** |

Sets the maximum number of simultaneous connections that can be opened by a worker process. Includes all connections, even the proxy pass

Cannot exceed what you set in [worker\_rlimit\_nofile](https://nginx.org/en/docs/ngx_core_module.html#worker_rlimit_nofile).

| Syntax: | **worker\_processes** *number* | auto; |
| --- | --- |
| Default: | worker\_processes 1; |

Defines the number of worker processes.

The optimal value depends on many factors including (but not limited to) the number of CPU cores

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem: port 80 and port 90:**

Deleted the running docker tutorial container and works fine now!

The error of /static was not getting added to the error page because it was getting added to docker errors.

Still don’t know why if I gave port 80 at nginx it still displayed the nginx home page with curl command and docker tutorial on the website!

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem: netstat not showing the listening ports:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**