Lab 5: Serving Content: Location

Estimated time for completion: **15 minutes**

Requirements

The following tasks must be completed before beginning this lab:

- Getting Started with NGINX, (the Getting Started Guide in LearnF5)
- Log into Hosted Environment, your lab initialization instructions are located in the LearnF5 course

Scenario

In this exercise, find and replace the included default configuration file. Use two server blocks with different listen directives and determine which server responds.

Objectives

At the end of this lab you will be able to:

- Create and test a new configuration file
- Determine which location block will serve a request

Lab Contents

Exercise 1: Update a configuration file and test it.

IMPORTANT
You can copy and paste the commands and text from the examples to your terminal or editor, (just make sure you don't copy and paste the $ prompt!)

Exercise 1: Update a configuration file and test it.

1. Rename the **`return_test.conf`** file:

   ```
   $ cd /etc/nginx/conf.d
   $ sudo mv return_test.{conf,bak}
   ```

   This command renames (moves) the file **`return_test.conf`** to a file named **`return_test.bak.`**

2. Rename the **`mywebserver.bak`** file to **`mywebserver.conf`** to make it active:

   ```
   $ sudo vim mywebserver.{conf,bak}
   ```

   This command renames (moves) the file **`mywebserver.bak`** to a file named **`mywebserver.conf`**.

3. Edit the **`mywebserver.conf`** file and insert the below **`highlighted text`** to create 3 locations. You are creating two locations that have content already in them, and adding a root directive to the **`/images`** location:

NOTE
Ensure that you insert the text BEFORE the last } character, and not after it. An easy way to do this (in VIM) is to place your cursor on the root directive line and then type **`o`** (a little o) to open a new line below where you can paste the locations.

```
$ sudo vim mywebserver.conf
```

There should be a main (master) NGINX process and at least one worker process displayed, with their corresponding Process ID's (highlight added below).

```
server {
  listen 80;
    root /home/ubuntu/public_html;

    location /application1 {
    }

    location /application2 {
    }

    location /images {
        root /data;
    }
}
```

Your file should look exactly like this:

```
server {
  listen 80;
    root /home/ubuntu/public_html;

    location /application1 {
    }

    location /application2 {
    }

    location /images {
        root /data;
    }
}
```

4. Save the file and reload NGINX. (**esc** and **:wq**).

   $ sudo nginx -s reload

5. Test the locations in a browser as follows (test all locations before you take the next step):

   http://<FQDN>
   http://<FQDN>/application1

```
http://<FQDN>/application2
http://<FQDN>/images/logo.png
```

6. Why do you think you received a "403 Forbidden" on hte application URLs?

7. List the files in the **`application1`** directory:

   ```
   $ ls /home/ubuntu/public_html/application1
   ```

   Note here is no **`index.html`** file, which NGINX looks for by default.

8. Update your **`mywebserver.conf`** file by adding the below **`highlighted text`** to configure the correct index files:

   ```
   $ sudo vim mywebserver.conf
   ```

```
location /application1 {
  index app1.html;
}

location /application2 {
    index app2.html;
}
```

9. Save the file and reload NGINX. (**`esc`** and **`:wq`**).

   ```
   $ sudo nginx -s reload
   ```

10. Retest the **`application1`** and **`application2`** endpoints in your browser.

    ```
    http://< FQDN>/application1
    http://< FQDN>/application2
    ```

    Now that your location block has been set to use the file **`/home/ubuntu/application1/app1.html`** instead of **`index.html`** the webpage will display properly and not give 403 Forbidden.

Expected Results

In these exercises, you were able to find and replace the included default configuration file. Use two server blocks with different listen directives and determine which server responds.


STOP