Lab 5.1: Embedded Variables

Use the credentials: root/training when accessing the server's command line.

1. Backup the default.conf file

```
mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.bak
```

2. Create a new configuration called variable_test.conf

```
vim /etc/nginx/conf.d/variable_test.conf
```

3. Add the following server block to the configuration:

```
server {
  listen 80 default_server;
  server_name localhost;
  return 200 "Host = $host | Request = $uri | Argument = $args \n";

}
```

4. Save and exit variable_test.conf.
5. Reload NGINX.
6. Use curl to test the following URIs:

```
curl http://localhost/test
```

```
curl http://localhost/test?1234
```

Lab 5.2 Setting variables

1. Open variable_test.conf

```
vim /etc/nginx/conf.d/variable_test.conf
```

2. Comment out the return directive in the server context

```
# return 200 "Host = $host | Request = $uri | Argument = $args \n";
```

3. Add the three location blocks below to the server context:

```
location / {
    return 200 "custom variable is $custom_variable \n";
}

location /test1 {
    set $custom_variable 42;
    return 200 "custom variable is $custom_variable \n";
}
```

```
location /test2 {
    set $custom_variable "null";
    return 200 "custom variable is $custom_variable \n";
}
```

4. Save and exit variable_test.conf
5. Reload NGINX.
6. Use curl to test the URIs below. What value is returned for "custom_variable"?

```
curl http://localhost/test1

curl http://localhost/test2

curl http://localhost/test
```

Lab 5.3 Variable scope

1. Open variable_test.conf
2. Add a set directive in the server context as shown below:

```
set $custom_variable 0;
```

3. Save and exit variable_test.conf
4. Reload NGINX
5. Use curl to test the URIs below. Any changes?

```
curl http://localhost/test1

curl http://localhost/test2

curl http://localhost/test
```

Each lab uses a fresh environment so remember to shut down the environment when done.