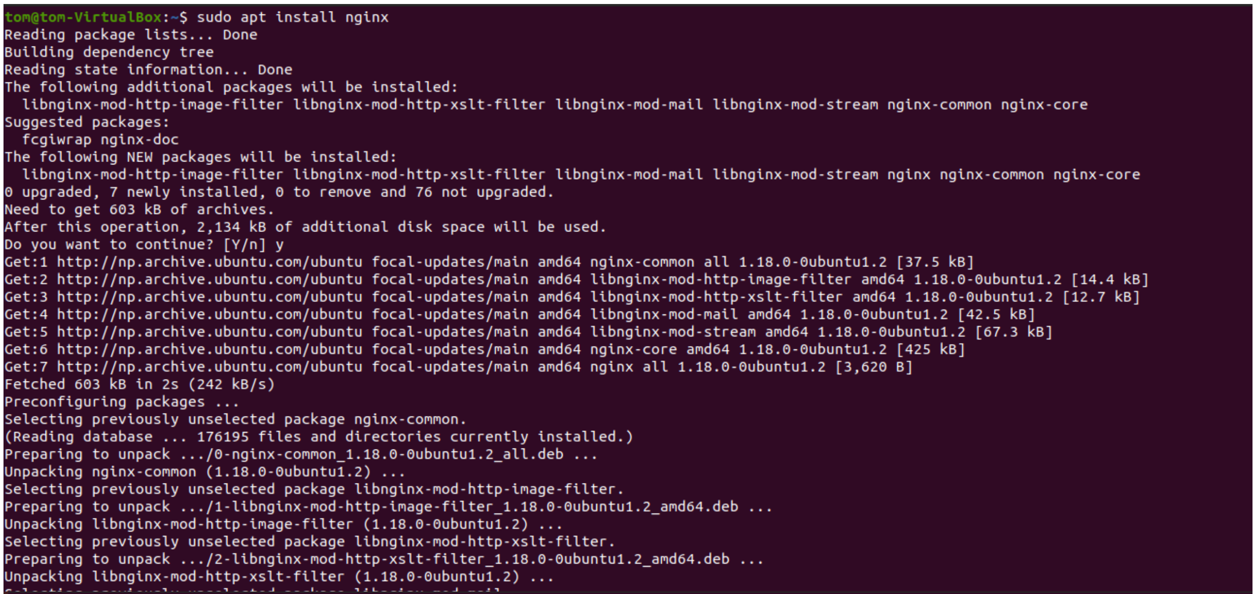
Nginx Assignment

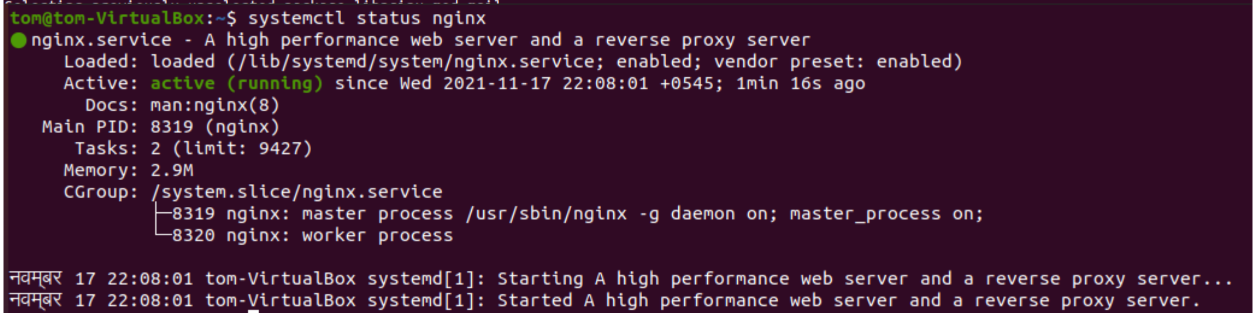
1. Install nginx and host a simple index.html with message “hello nginx”

To Install nginx we will first need to run the Linux command:

```
~ sudo apt update
~ sudo apt install nginx
```



```
tom@tom-VirtualBox:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx nginx-common nginx-core
0 upgraded, 7 newly installed, 0 to remove and 76 not upgraded.
Need to get 603 kB of archives.
After this operation, 2,134 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-common all 1.18.0-0ubuntu1.2 [37.5 kB]
Get:2 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-0ubuntu1.2 [14.4 kB]
Get:3 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-0ubuntu1.2 [12.7 kB]
Get:4 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-mail amd64 1.18.0-0ubuntu1.2 [42.5 kB]
Get:5 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-stream amd64 1.18.0-0ubuntu1.2 [67.3 kB]
Get:6 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-core amd64 1.18.0-0ubuntu1.2 [425 kB]
Get:7 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx all 1.18.0-0ubuntu1.2 [3,620 B]
Fetched 603 kB in 2s (242 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 176195 files and directories currently installed.)
Preparing to unpack .../0-nginx-common_1.18.0-0ubuntu1.2_all.deb ...
Unpacking nginx-common (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package libnginx-mod-http-image-filter.
Preparing to unpack .../1-libnginx-mod-http-image-filter_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking libnginx-mod-http-image-filter (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package libnginx-mod-http-xslt-filter.
Preparing to unpack .../2-libnginx-mod-http-xslt-filter_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.2) ...
```



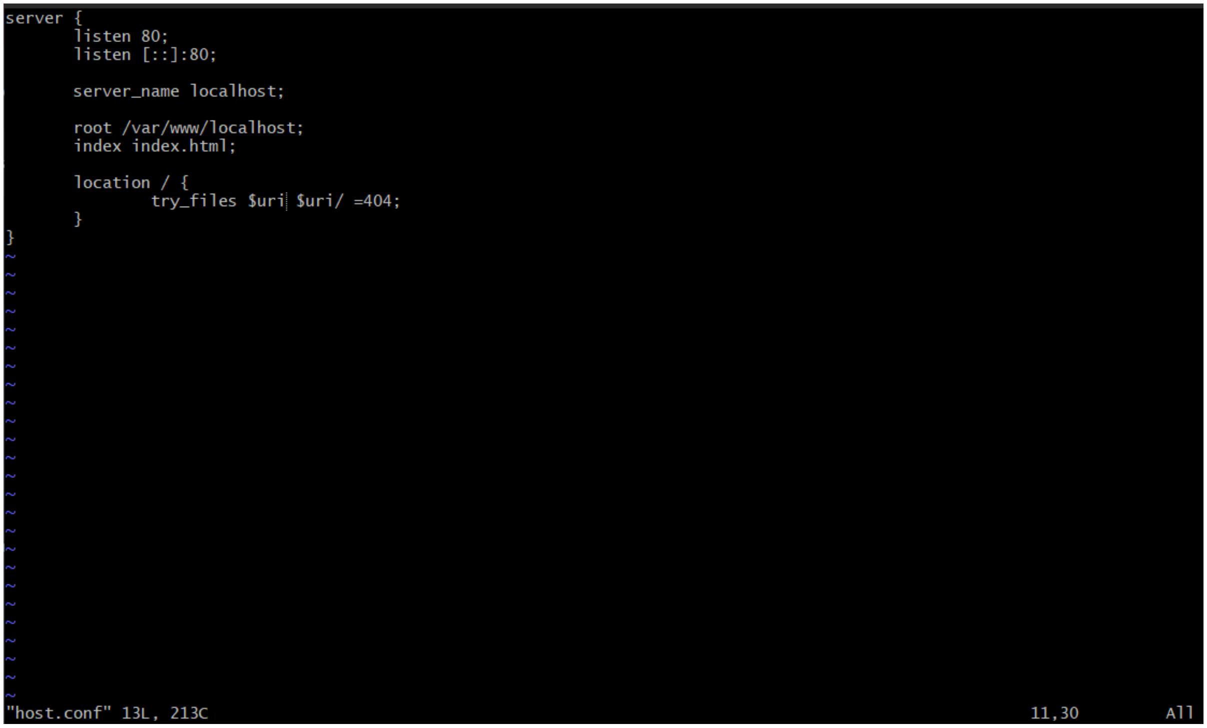
```
tom@tom-VirtualBox:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2021-11-17 22:08:01 +0545; 1min 16s ago
       Docs: man:nginx(8)
   Main PID: 8319 (nginx)
      Tasks: 2 (limit: 9427)
     Memory: 2.9M
     CGroup: /system.slice/nginx.service
             ├─8319 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─8320 nginx: worker process

नवम्बर 17 22:08:01 tom-VirtualBox systemd[1]: Starting A high performance web server and a reverse proxy server...
नवम्बर 17 22:08:01 tom-VirtualBox systemd[1]: Started A high performance web server and a reverse proxy server.
```

Now to configure the nginx server, we create a configuration file at **/etc/nginx/sites-available/host.conf:**

```
server {
      listen 80;
      listen [::]:80;
      server_name localhost;
```

```
        root /var/www/localhost;
        index index.html;
        location / {
                try_files $uri $uri/ =404;
        }
}
```



```
server {
        listen 80;
        listen [::]:80;

        server_name localhost;

        root /var/www/localhost;
        index index.html;

        location / {
                try_files $uri $uri/ =404;
        }
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"host.conf" 13L, 213C                                                  11,30         All
```

Create symlink of **/etc/nginx/sites-available/host.conf** with **/etc/nginx/sites-enabled/** using the command:

```
~ ln -s /etc/nginx/sites-available/host.conf /etc/nginx/sites-enabled/
```

And to test the configuration file is correct use the command:

```
~ nginx -t
```

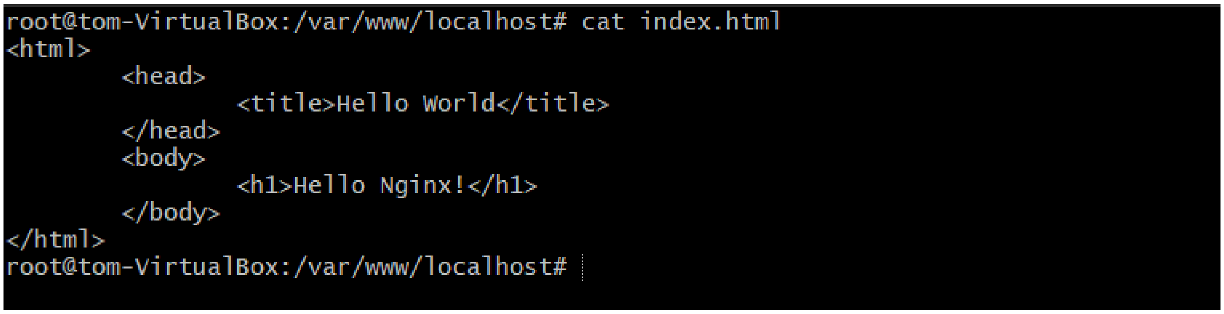
Now to restart the nginx server after configuration changes,

```
~ sudo systemctl reload nginx
```

Now to create **index.html** at **/var/www/localhost/** as

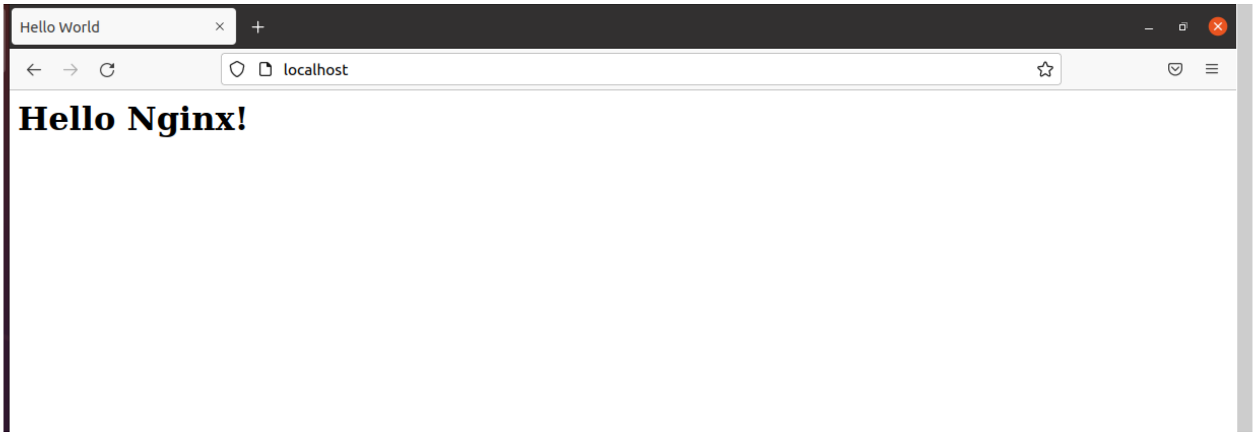
```
<html>
        <head>
                <title>Hello World</title>
        </head>
        <body>
                <h1>Hello Nginx!</h1>
```

```
          </body>
</html>
```



```
root@tom-VirtualBox:/var/www/localhost# cat index.html
<html>
        <head>
                <title>Hello World</title>
        </head>
        <body>
                <h1>Hello Nginx!</h1>
        </body>
</html>
root@tom-VirtualBox:/var/www/localhost#
```

So now the nginx server will response with **index.html** as such:



2. What are nginx header security and its uses. And also implement in the test.conf file.

The security HTTP headers are the response HTTP headers, that server can add in order to harden the security of HTTP exchange.

There are a few, and as the web evolves, more are being added. Each security header serves its own purpose.

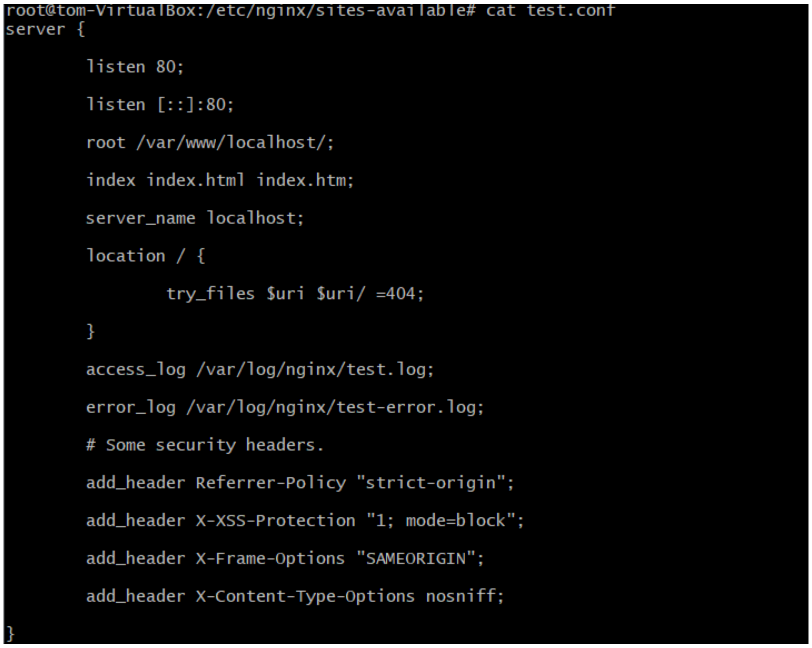
- HTTP Strict Transport Security (HSTS)
- Public Key Pinning Extension for HTTP (HPKP)
- X-Frame-Options
- X-XSS-Protection
- X-Content-Type-Options
- Content-Security-Policy
- X-Permitted-Cross-Domain-Policies
- Referrer-Policy
- Expect-CT
- Feature-Policy

HTTP security headers are added to responses, so that the browsers behave in a more secure way.

We can add security headers in nginx responses using **add_header** directive, example:

```
add_header X-XSS-Protection "1; mode=block";
```

Let us implement it on **test.conf** as given:



```
root@tom-VirtualBox:/etc/nginx/sites-available# cat test.conf
server {

        listen 80;

        listen [::]:80;

        root /var/www/localhost/;

        index index.html index.htm;

        server_name localhost;

        location / {

                try_files $uri $uri/ =404;

        }

        access_log /var/log/nginx/test.log;

        error_log /var/log/nginx/test-error.log;

        # Some security headers.

        add_header Referrer-Policy "strict-origin";

        add_header X-XSS-Protection "1; mode=block";

        add_header X-Frame-Options "SAMEORIGIN";

        add_header X-Content-Type-Options nosniff;

}
```

Create symlink of **/etc/nginx/sites-available/test.conf** with **/etc/nginx/sites-enabled/** using the command:

```
~ ln -s /etc/nginx/sites-available/test.conf /etc/nginx/sites-enabled/
```

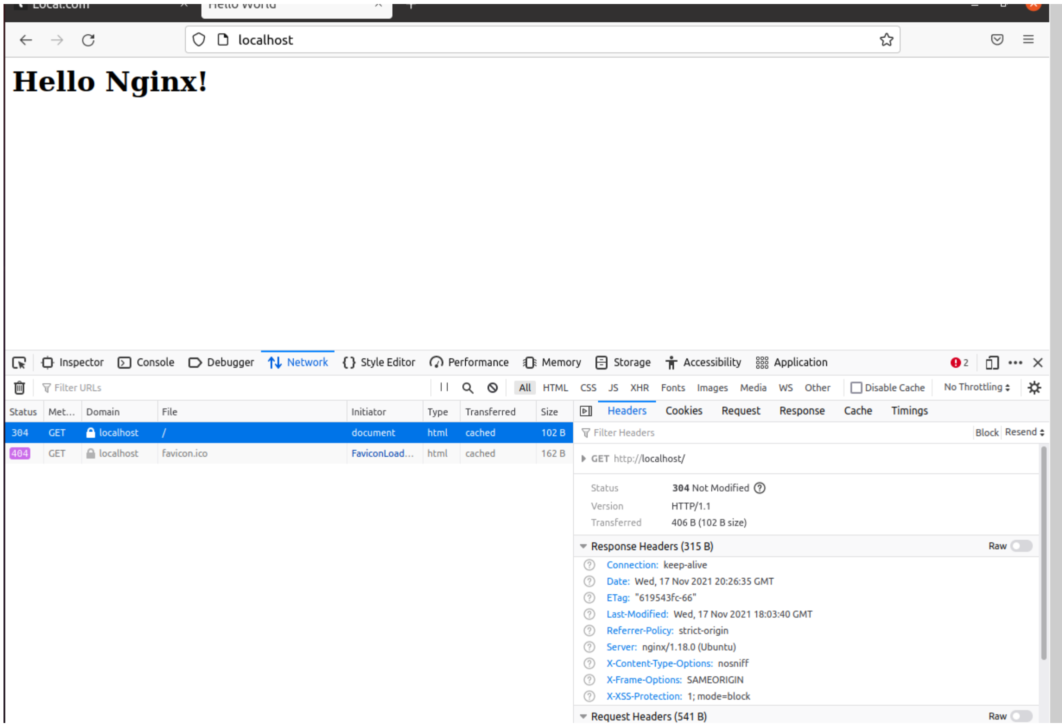
And to test the configuration file is correct use the command:

```
~ nginx -t
```

Now to restart the nginx server after configuration changes,

```
~ sudo systemctl reload nginx
```

We can inspect the security headers in the browser's debug panel.



Hello Nginx!

Status	Met...	Domain	File	Initiator	Type	Transferred	Size
304	GET	localhost	/	document	html	cached	102 B
404	GET	localhost	favicon.ico	FaviconLoad...	html	cached	162 B

GET http://localhost/

Status 304 Not Modified
Version HTTP/1.1
Transferred 406 B (102 B size)

Response Headers (315 B)

```
Connection: keep-alive
Date: Wed, 17 Nov 2021 20:26:35 GMT
ETag: "619543fc-66"
Last-Modified: Wed, 17 Nov 2021 18:03:40 GMT
Referrer-Policy: strict-origin
Server: nginx/1.18.0 (Ubuntu)
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
```

Request Headers (541 B)

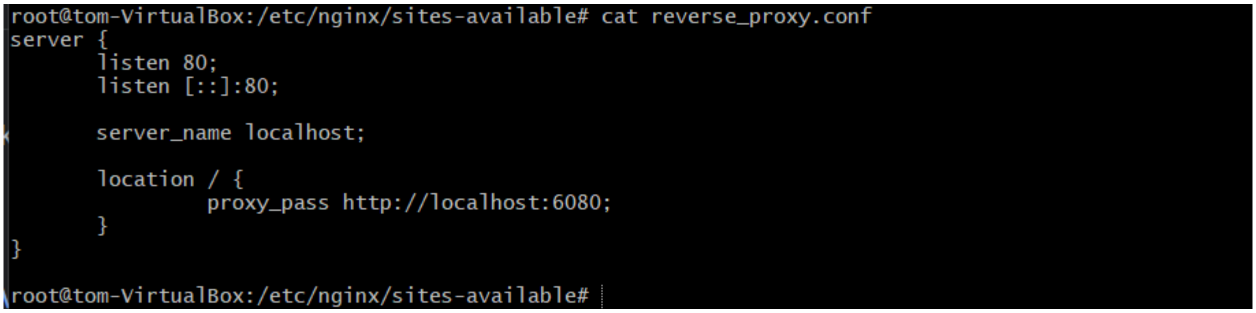
3. Nginx Reverse proxy all http requests to nodes js api.

Reverse Proxy from all http requests (port **80**) is directed towards node js app which is being served at port **6080**:
So the configuration file, **reverse_proxy.conf**, becomes:

```
server {
        listen 80;
        listen [::]:80;

        server_name localhost;

        location / {
                  proxy_pass http://localhost:6080;
        }
}
```



```
root@tom-VirtualBox:/etc/nginx/sites-available# cat reverse_proxy.conf
server {
        listen 80;
        listen [::]:80;

        server_name localhost;

        location / {
                  proxy_pass http://localhost:6080;
        }
}

root@tom-VirtualBox:/etc/nginx/sites-available#
```

Create symlink of **/etc/nginx/sites-available/reverse_proxy.conf** with **/etc/nginx/sites-enabled/** using the command:

```
~ ln -s /etc/nginx/sites-available/reverse_proxy.conf
  /etc/nginx/sites-enabled/
```

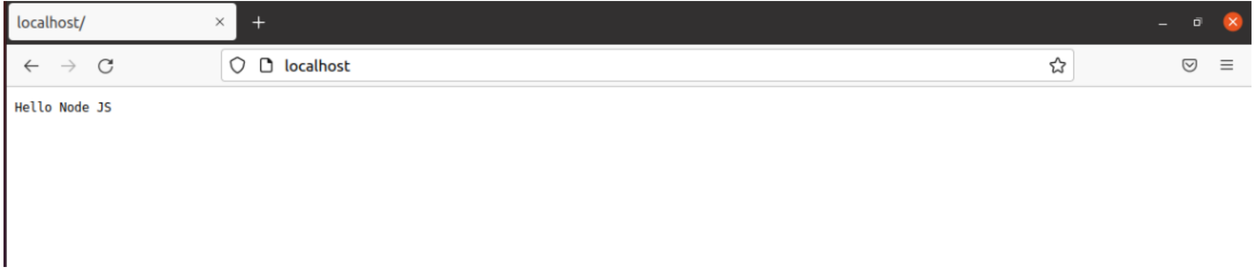
And to test the configuration file is correct use the command:

```
~ nginx -t
```

Now to restart the nginx server after configuration changes,

```
~ sudo systemctl reload nginx
```

So now the the port 80 proxy to port 6080 which is hosting node app



```
Hello Node JS
```

4. Create a test2.conf and listen on port 82 and to “ location /test/” with message “ test is successful”.

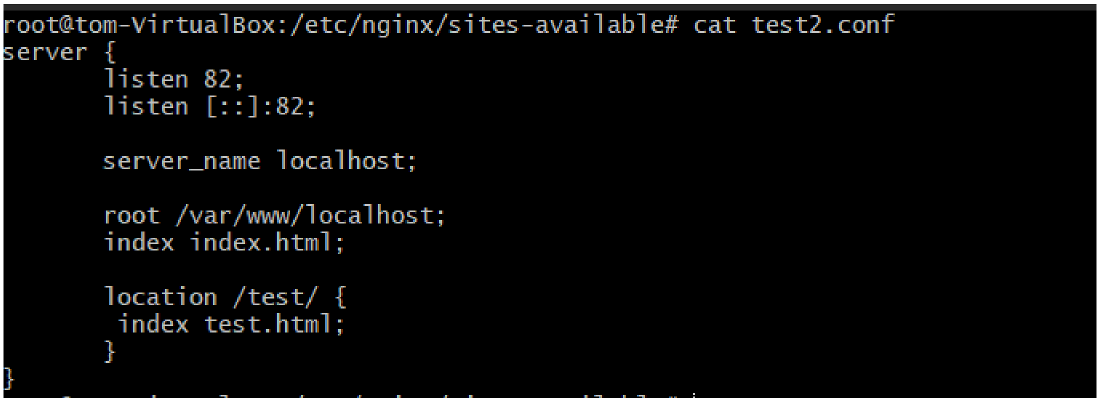
Create a test2.conf configuration file and add the following:

```
server {
        listen 82;
        listen [::]:82;

        server_name localhost;

        root /var/www/localhost;
        index index.html;

        location /test/ {
         index test.html;
        }
}
```



```
root@tom-VirtualBox:/etc/nginx/sites-available# cat test2.conf
server {
        listen 82;
        listen [::]:82;

        server_name localhost;

        root /var/www/localhost;
        index index.html;

        location /test/ {
         index test.html;
        }
}
```

Create symlink of **/etc/nginx/sites-available/test2.conf** with **/etc/nginx/sites-enabled/** using the command:

```
~ ln -s /etc/nginx/sites-available/test2.conf /etc/nginx/sites-enabled/
```

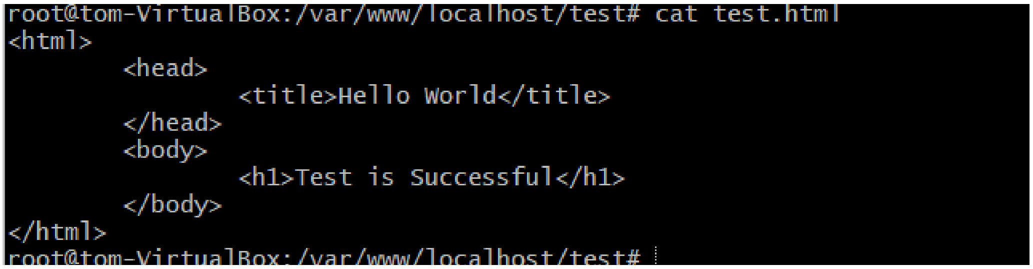
And to test the configuration file is correct use the command:

```
~ nginx -t
```

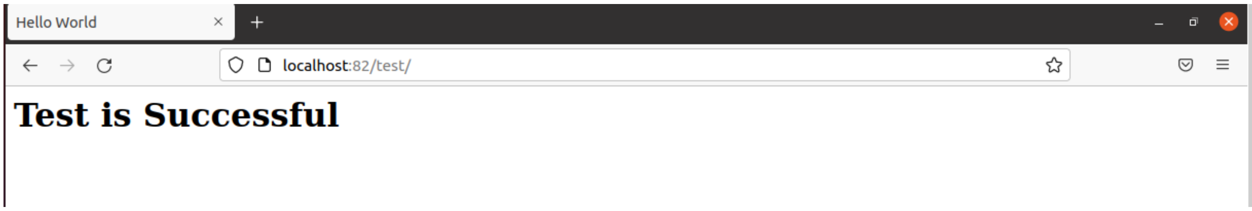
Now to restart the nginx server after configuration changes,

```
~ sudo systemctl reload nginx
```

The html file is now inside **/var/www/localhost/test/test.html,**



```
root@tom-VirtualBox:/var/www/localhost/test# cat test.html
<html>
        <head>
                <title>Hello World</title>
        </head>
        <body>
                <h1>Test is Successful</h1>
        </body>
</html>
root@tom-VirtualBox:/var/www/localhost/test#
```



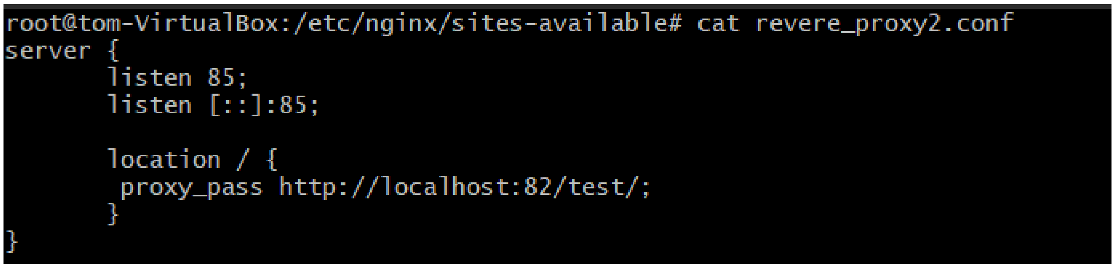
Test is Successful

5. Reverse proxy all http traffic of port 82 to port 85.

Creating reverse proxy with configuration file

```
server {
       listen 85;
       listen [::]:85;

       location / {
        proxy_pass http://localhost:82/test/;
       }
}
```



```
root@tom-VirtualBox:/etc/nginx/sites-available# cat revere_proxy2.conf
server {
       listen 85;
       listen [::]:85;

       location / {
        proxy_pass http://localhost:82/test/;
       }
}
```

Create symlink of **/etc/nginx/sites-available/revere_proxy2.conf** with **/etc/nginx/sites-enabled/** using the command:

```
~ ln -s /etc/nginx/sites-available/revere_proxy2.conf
  /etc/nginx/sites-enabled/
```

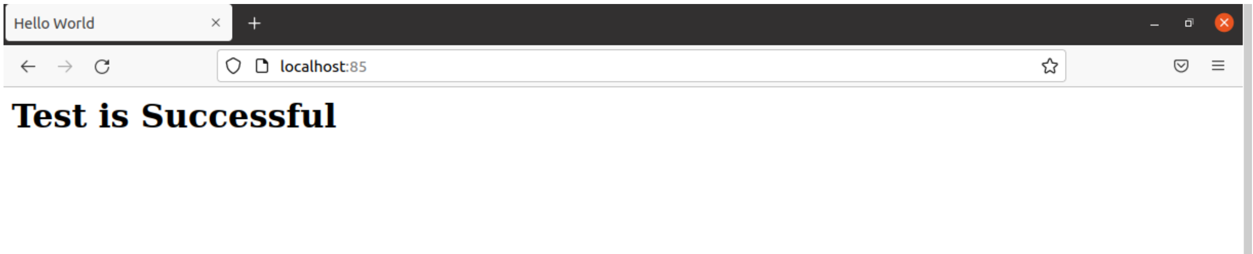
And to test the configuration file is correct use the command:

```
~ nginx -t
```

Now to restart the nginx server after configuration changes,

```
~ sudo systemctl reload nginx
```

Now all traffic is proxy to 82

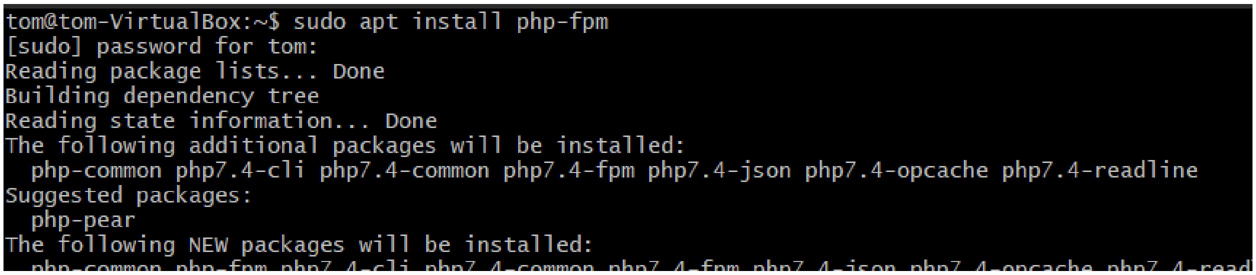


Test is Successful

6. Install LEMP stack (avoid installing mysql) and open info.php on port 80 and print message info.php

LEMP stands for Linux, Nginx, MySQL, Php stack. First to process php, nginx need **php-fpm** which can be installed as

```
~ sudo apt install php-fpm
```



```
tom@tom-VirtualBox:~$ sudo apt install php-fpm
[sudo] password for tom:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  php-common php7.4-cli php7.4-common php7.4-fpm php7.4-json php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  php-common php-fpm php7.4-cli php7.4-common php7.4-fpm php7.4-json php7.4-opcache php7.4-read
```

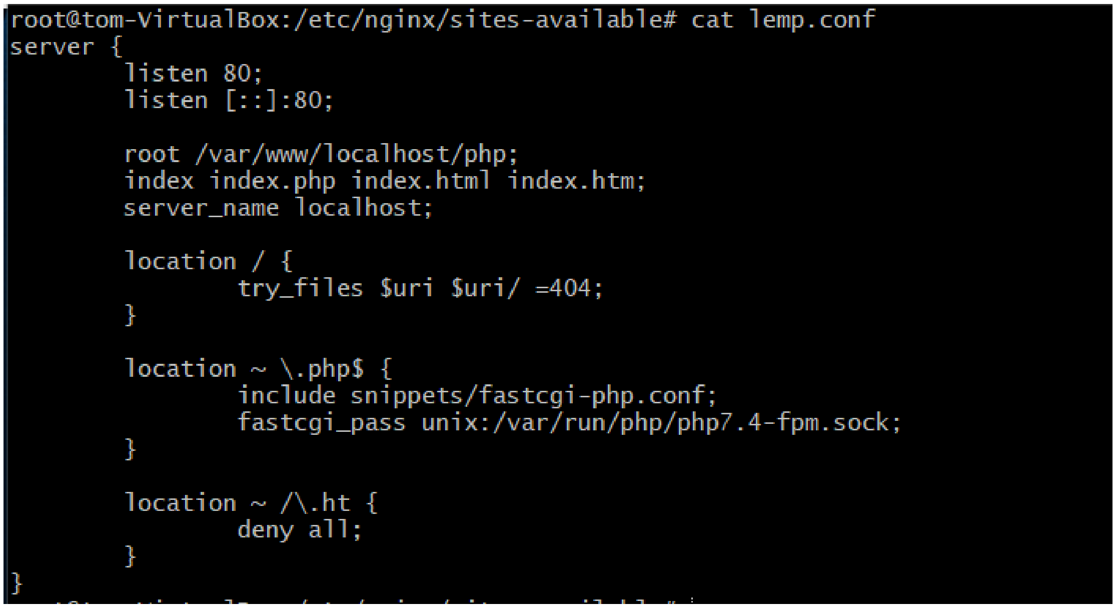
So now the configuration file needs to be changed as follows:

```
server {
        listen 80;
        root /var/www/localhost;
        index index.php index.html index.htm;
        server_name localhost;

        location / {
                try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
        }

        location ~ /\.ht {
                deny all;
        }
}
```



```
root@tom-VirtualBox:/etc/nginx/sites-available# cat lemp.conf
server {
        listen 80;
        listen [::]:80;

        root /var/www/localhost/php;
        index index.php index.html index.htm;
        server_name localhost;

        location / {
                try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        }

        location ~ /\.ht {
                deny all;
        }
}
```

Create symlink of **/etc/nginx/sites-available/lemp.conf** with **/etc/nginx/sites-enabled/** using the command:

```
~ ln -s /etc/nginx/sites-available/lemp.conf /etc/nginx/sites-enabled/
```

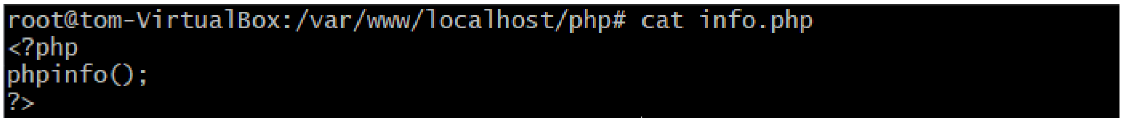
And to test the configuration file is correct use the command:

```
~ nginx -t
```

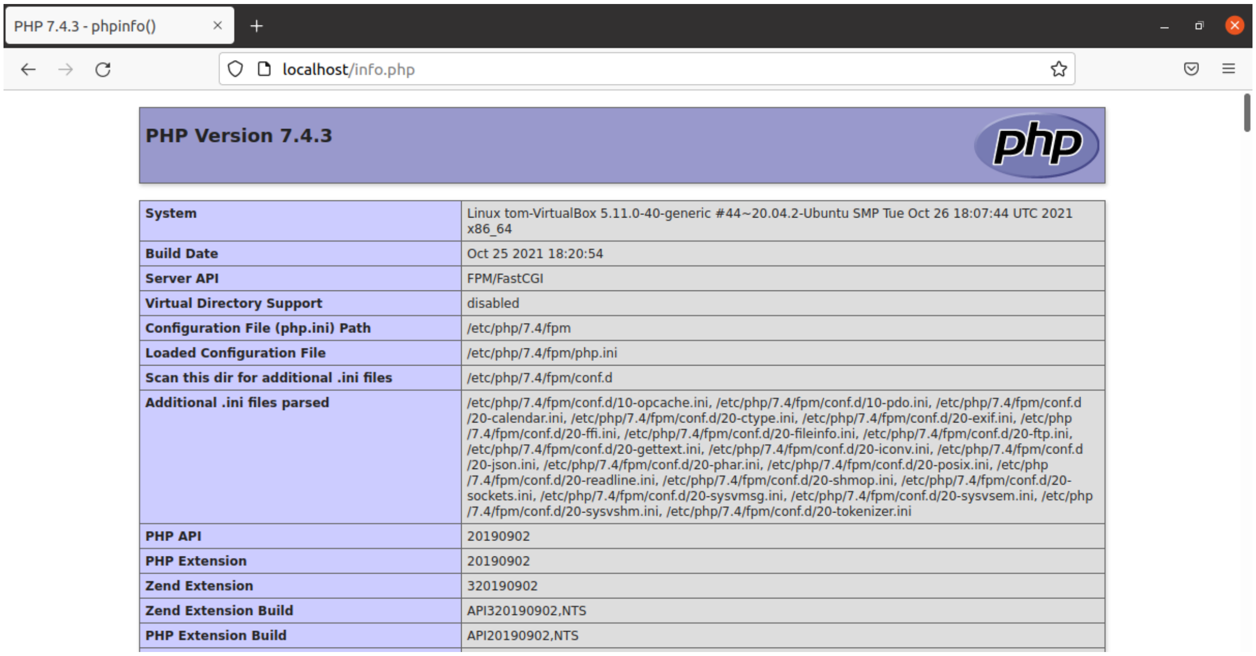
Now to restart the nginx server after configuration changes,

```
~ sudo systemctl reload nginx
```

Now to edit **info.php** as:



```
root@tom-VirtualBox:/var/www/localhost/php# cat info.php
<?php
phpinfo();
?>
```



PHP Version 7.4.3

System	Linux tom-VirtualBox 5.11.0-40-generic #44~20.04.2-Ubuntu SMP Tue Oct 26 18:07:44 UTC 2021 x86_64
Build Date	Oct 25 2021 18:20:54
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/fpm
Loaded Configuration File	/etc/php/7.4/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/fpm/conf.d
Additional .ini files parsed	/etc/php/7.4/fpm/conf.d/10-opcache.ini, /etc/php/7.4/fpm/conf.d/10-pdo.ini, /etc/php/7.4/fpm/conf.d/20-calendar.ini, /etc/php/7.4/fpm/conf.d/20-ctype.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-ffi.ini, /etc/php/7.4/fpm/conf.d/20-fileinfo.ini, /etc/php/7.4/fpm/conf.d/20-ftp.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-json.ini, /etc/php/7.4/fpm/conf.d/20-phar.ini, /etc/php/7.4/fpm/conf.d/20-posix.ini, /etc/php/7.4/fpm/conf.d/20-readline.ini, /etc/php/7.4/fpm/conf.d/20-shmop.ini, /etc/php/7.4/fpm/conf.d/20-sockets.ini, /etc/php/7.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvshm.ini, /etc/php/7.4/fpm/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS