NGINX

- It is a http server that is used in many high traffic websites.
- It is designed for both web and proxy serverr.
- It provides Load Balancing, caching, Reverse Proxy

HOW TO INSTALL NGINX:-

sudo apt-get update
sudo apt-get install nginx

HOW TO APPLY REVERSE PROXY:-

Once nginx is installed, then, we need to create a file within the `/etc/nginx/sites-available` directory that contains the reverse proxy information. We can name this `reverse-proxy.conf` for example.

```
server {
    listen 80;
    location / {
    proxy_pass http://192.x.x.2;
    }
}
```

The important part here is the `proxy_pass` directive which is essentially telling any requests coming through the Nginx reverse proxy to be passed along to the Apache remote socket `192.x.x.2:80`

Once you've added the appropriate directives to your .conf file, activate it by linking to /sites-enabled/ using the following command.

```
ln -s /etc/nginx/sites-available/reverse-proxy.conf
/etc/nginx/sites-enabled/reverse-proxy.conf
```


```
ln -s /etc/nginx/sites-available/reverse-proxy.conf
/etc/nginx/sites-enabled/reverse-proxy.conf
```


Lastly, run an Nginx configuration test and restart Nginx.

```
service nginx configtest
service nginx restart
```

Now after this if you will type localhost in your browser then you will get the same response as it was on `http://192.x.x.2;`

Now if you want the above process to execute with a proper dns like abc.com then you need to do

server {

```
listen 80;

server_name abc.com;

location / {

    proxy_pass http://192.x.x.2;

    }

}
```

Once you've added the appropriate directives to your .conf file, activate it by linking to /sites-enabled/ using the following command.

```
ln -s /etc/nginx/sites-available/<filename> /etc/nginx/sites-enabled/<filename>
```

After that you to enter the server name i.e. abc.com into your host directory by using

sudo nano /etc/hosts/

localhost(127.0.0.1) abc.com

Lastly, run an Nginx configuration test and restart Nginx.

```
service nginx configtest
service nginx restart
```

Display “Hello i am nginx” by using lua when user hits abc.hello:-

For this also you need to create a .conf file and in that file write

```
server {
        listen 80;
        server_name abc.hello;
         location / {
         return 200 “Hello i am nginx”;
         add-header Content-Type text/plain;
         }
}
```

Once you’ve added the appropriate directives to your .conf file, activate it by linking to /sites-enabled/ using the following command.

```
ln -s /etc/nginx/sites-available/<filename> /etc/nginx/sites-enabled/<filename>
```

After that you to enter the server name i.e. abc.hello into your host directory by using

sudo nano /etc/hosts/

localhost(127.0.0.1) abc.hello

Lastly, run an Nginx configuration test and restart Nginx.

```
service nginx configtest
service nginx restart
```

SHOW A STATIC JSON AT PARTICULAR ENDPOINT (EX:- abc.hello):-

```
server {
        listen 80;
        server_name abc.hello;
         location / {
         return 200 “{“fruit” : “mango” , “color” : “yellow” , “size” : “large”}”;
         add-header Content-Type text/plain;
         }
}
```

Once you’ve added the appropriate directives to your .conf file, activate it by linking to /sites-enabled/ using the following command.

```
ln -s /etc/nginx/sites-available/<filename> /etc/nginx/sites-enabled/<filename>
```

After that you to enter the server name i.e. abc.hello into your host directory by using

sudo nano /etc/hosts/

localhost(127.0.0.1) abc.hello

Lastly, run an Nginx configuration test and restart Nginx.

```
service nginx configtest
service nginx restart
```

NGINX HEADER TASK

```
server {
 listen 80;
 location /msisdn/fetch {
  return 200 '{"msisdn":"$http_x_msisdn"}';
   add_header Content-Type application/json;
  }
}
```