ASSESSMENT ON NGINX: WEB SERVER

TO THE NEW

1.What is the advantage of using a “reverse proxy server”?

Ans:

Advantages :

- Avoid the expense of installing another web server. A reverse proxy server increases the capacity of existing servers.
- Serve more requests for static content and thus free up bandwidth to serve more dynamic content.
- Reduce operating expense by increasing bandwidth.
- Provide a single point of control over who can access HTTP servers, and which servers can be accessed.
- Decrease response time of web pages and accelerate download time, enhancing the experience of web site users.
- Provide another layer of protection by hiding the internal IP address.

Disadvantages :

- If the reverse proxy is compromised and a failover is not in place, the site’s HTTP presence is shut down.
- If an outside attacker compromises the reverse proxy server, the attacker may also be able to get access into your HTTP server architecture. If the HTTP servers are inside your firewall, an attacker could compromise your internal network.
- A large amount of translations must be done between the reverse proxy and the firewall, so initial requests may be filled slightly slower.
- Cache content must have a single security access point and cannot be personalized.
- Because all HTTP traffic is going through a reverse proxy, content filtering must be added to ensure that HTTP requests are not actually attacks.

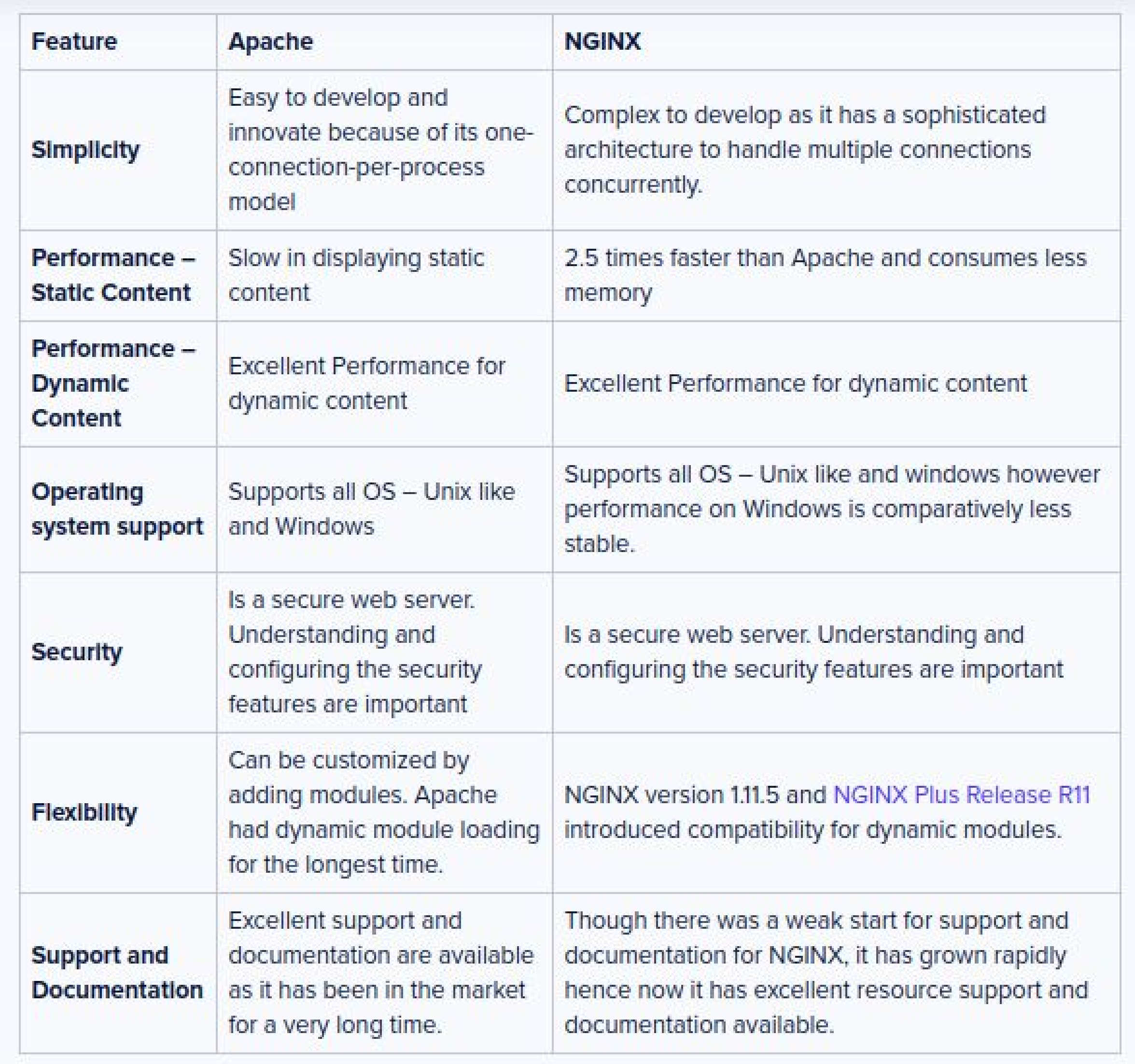
2.Why and where Nginx is a better choice than apache.

Apache	Nginx

Based on process-driven architecture.	Based on event-driven architecture.
Creates a new process for each request.	Doesn’t create a new process for a new request.
Nature of creating new process for each request leads to memory consumption.	Memory consumption is very low for serving static pages.
Supports much wider range of Operating Systems.	Do not support Operating Systems like OpenVMS and IBMi.

If your website is predominantly Python or Ruby, Apache might be preferred for your application, as Apache does not have to use CGI, because it does not have to load the Python interpreter for each request.

If your website is PHP dependent, Nginx is the far best way to host application.



Feature	Apache	NGINX
Simplicity	Easy to develop and innovate because of its one-connection-per-process model	Complex to develop as it has a sophisticated architecture to handle multiple connections concurrently.
Performance – Static Content	Slow in displaying static content	2.5 times faster than Apache and consumes less memory
Performance – Dynamic Content	Excellent Performance for dynamic content	Excellent Performance for dynamic content
Operating system support	Supports all OS – Unix like and Windows	Supports all OS – Unix like and windows however performance on Windows is comparatively less stable.
Security	Is a secure web server. Understanding and configuring the security features are important	Is a secure web server. Understanding and configuring the security features are important
Flexibility	Can be customized by adding modules. Apache had dynamic module loading for the longest time.	NGINX version 1.11.5 and NGINX Plus Release R11 introduced compatibility for dynamic modules.
Support and Documentation	Excellent support and documentation are available as it has been in the market for a very long time.	Though there was a weak start for support and documentation for NGINX, it has grown rapidly hence now it has excellent resource support and documentation available.

3.What are worker nodes and worker connections? How to calculate the max server capacity using the above two?

Ans:

`worker_processes` – The number of NGINX worker processes (the default is 1). In most cases, running one worker process per CPU core works well, and we recommend setting this directive to `auto` to achieve that. There are times when you may want to increase this number, such as when the worker processes have to do a lot of disk I/O.

`Worker connections` – The maximum number of connections that each worker process can handle simultaneously. The default is 512, but most systems have enough resources to support a larger number. The appropriate setting depends on the size of the server and the nature of the traffic, and can be discovered through testing

4.From what directory will NGINX automatically load server (virtual host) configurations when using the default /etc/nginx/nginx.conf configuration?

Ans: From sites-enabled

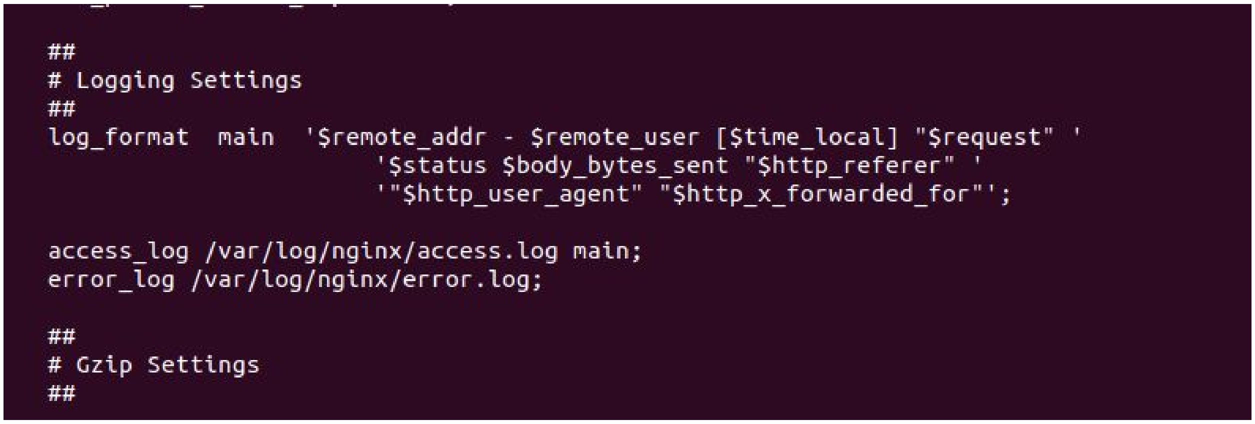
5.How to configure different log_format for different "location" block/directive?

Ans: Configuring the Access Log

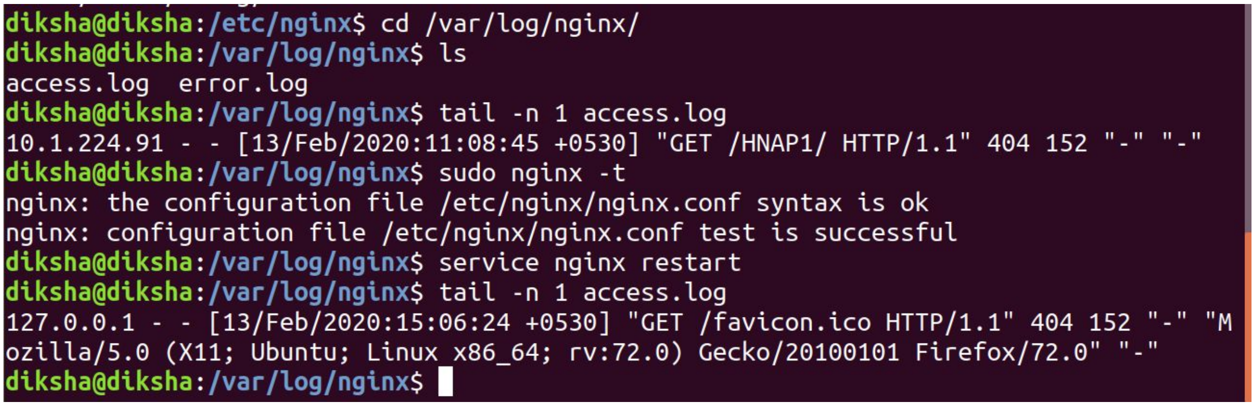
Whenever a client request is processed, Nginx generates a new event in the access log. Each event record contains a timestamp and includes various information about the client and the requested resource. Access logs can show you the location of the visitors, the page they visit, how much time they spend on the page, and much more.

1)cd /etc/nginx

2)vim nginx.conf



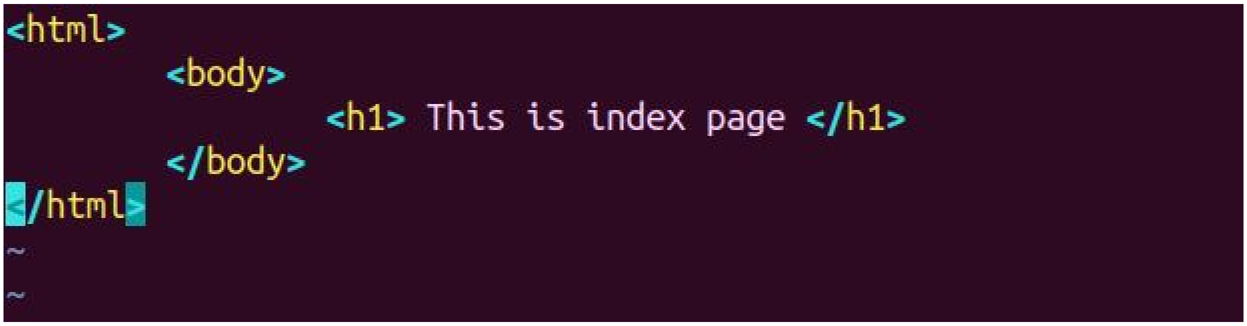
```
        ##
        # Logging Settings
        ##
        log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                          '$status $body_bytes_sent "$http_referer" '
                          '"$http_user_agent" "$http_x_forwarded_for"';

        access_log /var/log/nginx/access.log main;
        error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##
```



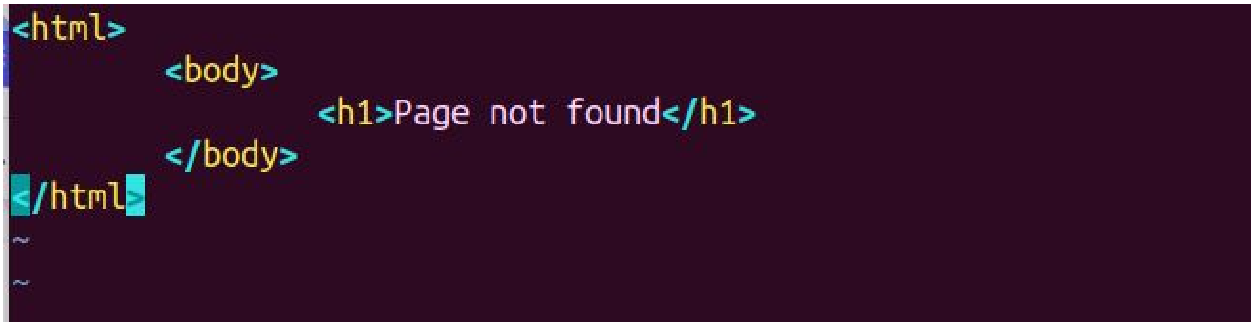
```
diksha@diksha:/etc/nginx$ cd /var/log/nginx/
diksha@diksha:/var/log/nginx$ ls
access.log  error.log
diksha@diksha:/var/log/nginx$ tail -n 1 access.log
10.1.224.91 - - [13/Feb/2020:11:08:45 +0530] "GET /HNAP1/ HTTP/1.1" 404 152 "-" "-"
diksha@diksha:/var/log/nginx$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
diksha@diksha:/var/log/nginx$ service nginx restart
diksha@diksha:/var/log/nginx$ tail -n 1 access.log
127.0.0.1 - - [13/Feb/2020:15:06:24 +0530] "GET /favicon.ico HTTP/1.1" 404 152 "-" "M
ozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0" "-"
diksha@diksha:/var/log/nginx$
```

6.Host a site ABC.COM

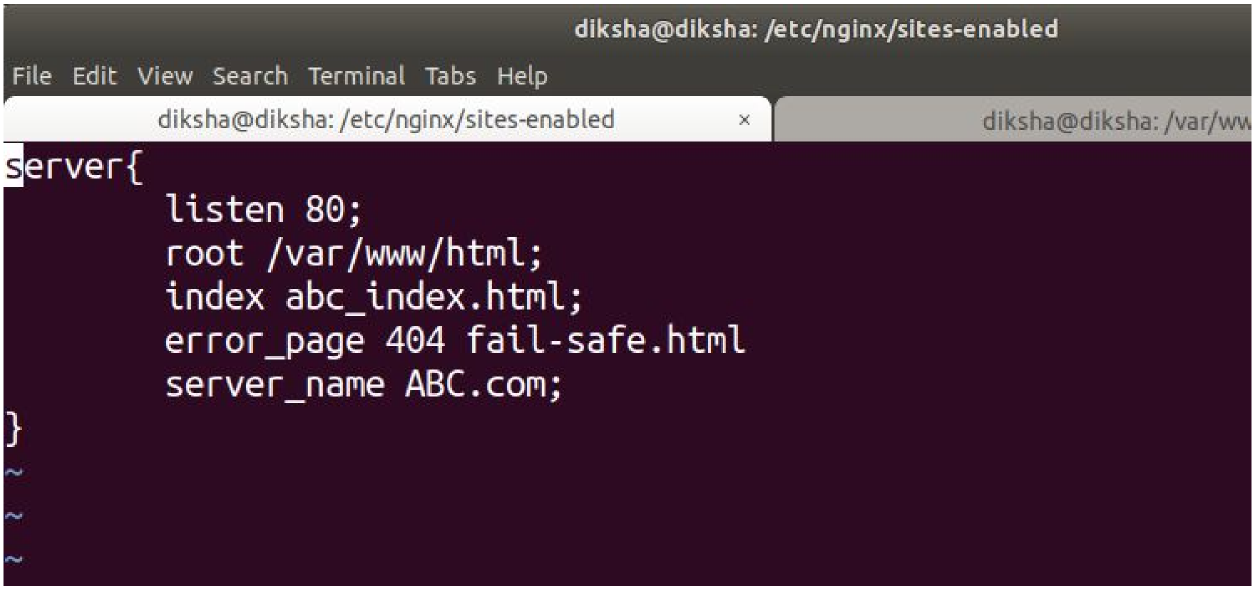
A. **Create an index page and a fail-safe page. If a page for URI is not available, the fail-safe page is served.**



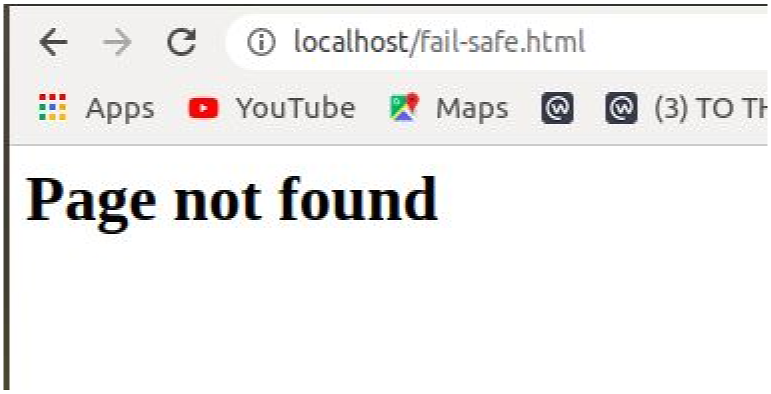
```
<html>
        <body>
                <h1> This is index page </h1>
        </body>
</html>
~
~
```



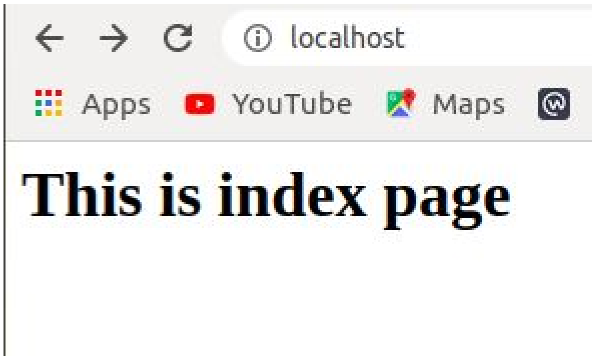
```
<html>
        <body>
                <h1>Page not found</h1>
        </body>
</html>
~
~
```


diksha@diksha: /etc/nginx/sites-enabled
File Edit View Search Terminal Tabs Help
diksha@diksha: /etc/nginx/sites-enabled
diksha@diksha: /var/ww

```
server{
        listen 80;
        root /var/www/html;
        index abc_index.html;
        error_page 404 fail-safe.html
        server_name ABC.com;
}
~
~
~
```



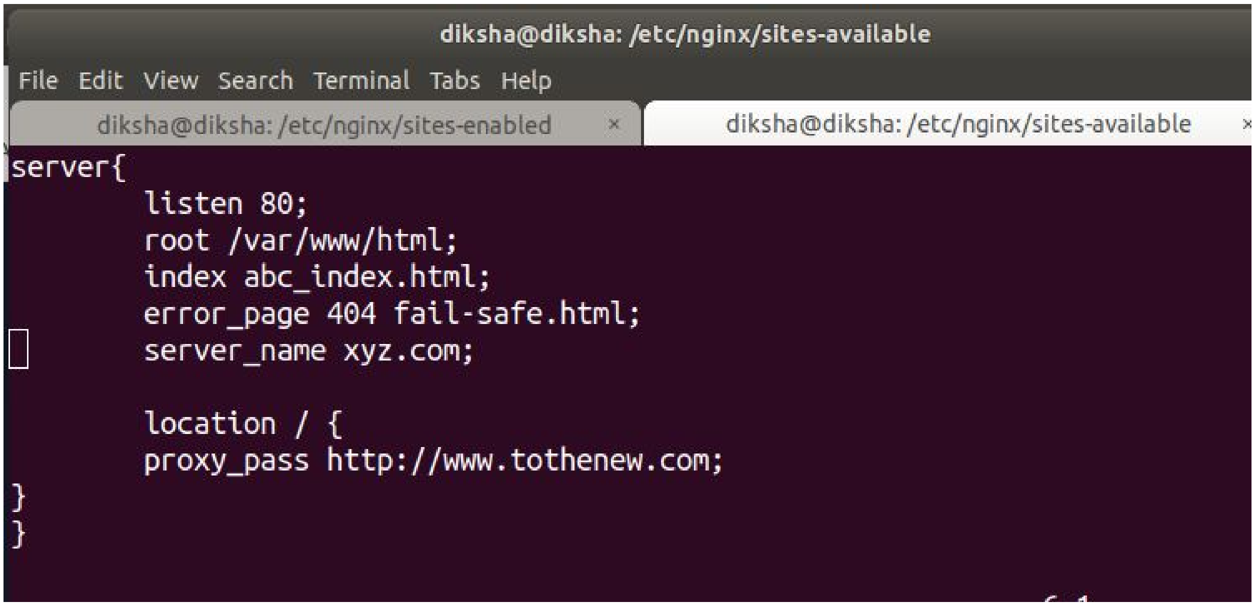
Page not found



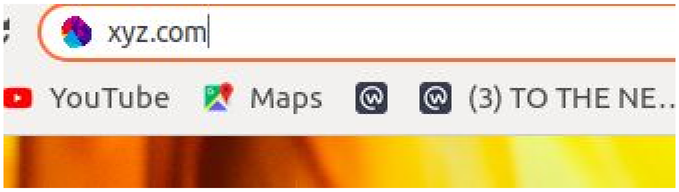
This is index page

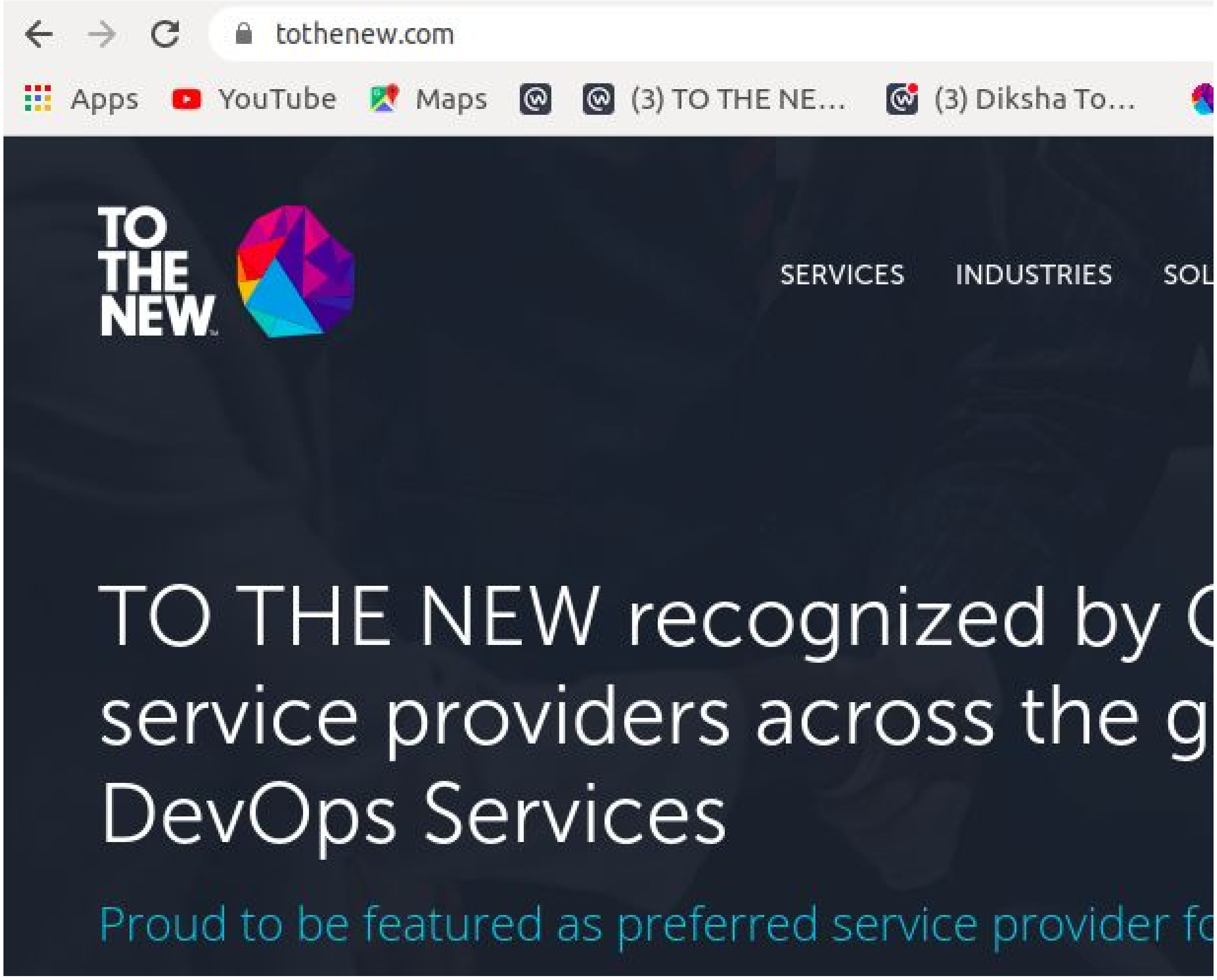
B.proxy pass to a website xyz.com on a particular URI.

ANS: Make a new configuration file in sites-available and then make its link in sites-enabled.Add the server name in hosts file.


diksha@diksha: /etc/nginx/sites-available
File Edit View Search Terminal Tabs Help
diksha@diksha: /etc/nginx/sites-enabled
diksha@diksha: /etc/nginx/sites-available

```
server{
        listen 80;
        root /var/www/html;
        index abc_index.html;
        error_page 404 fail-safe.html;
        server_name xyz.com;

        location / {
        proxy_pass http://www.tothenew.com;
}
}
```





TO THE NEW recognized by C
service providers across the g
DevOps Services

Proud to be featured as preferred service provider fo

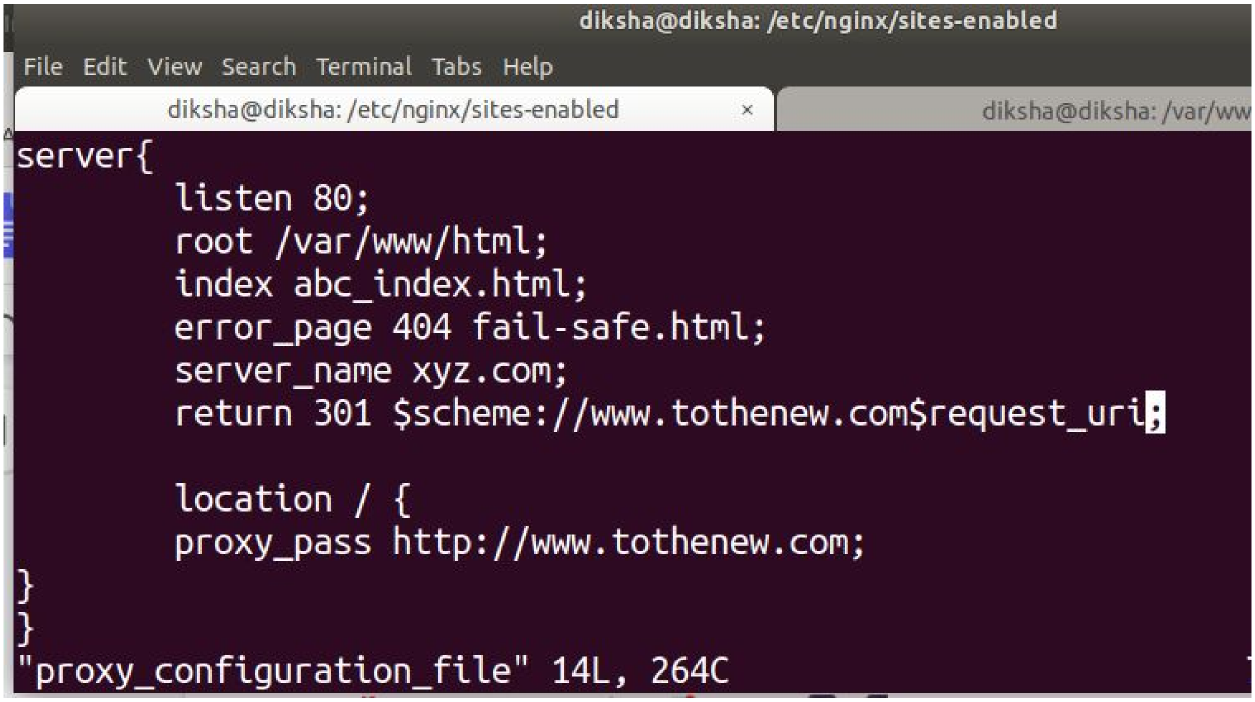
C.Redirect to above URI on /redirect/.

Ans:A redirect is a web server function that will redirect traffic from one URL to another. Redirects are an important feature when the need arises. There are several different types of redirects, but the more common forms are temporary and permanent

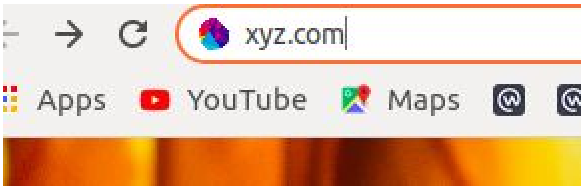
Common Methods for Redirects

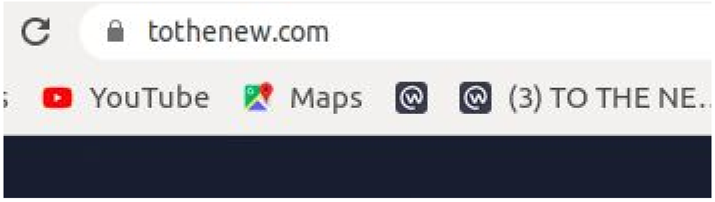
Temporary redirects (response code: 302 Found) are helpful if a URL is temporarily being served from a different location. For example, these are helpful when performing maintenance and can redirect users to a maintenance page.

However, permanent redirects (response code: 301 Moved Permanently) inform the browser there was an old URL that it should forget and not attempt to access anymore. These are helpful when content has moved from one place to another.


diksha@diksha: /etc/nginx/sites-enabled
File Edit View Search Terminal Tabs Help
diksha@diksha: /etc/nginx/sites-enabled
diksha@diksha: /var/ww

```
server{
        listen 80;
        root /var/www/html;
        index abc_index.html;
        error_page 404 fail-safe.html;
        server_name xyz.com;
        return 301 $scheme://www.tothenew.com$request_uri;

        location / {
        proxy_pass http://www.tothenew.com;
}
}
"proxy_configuration_file" 14L, 264C
```





To inform clients that the resource they're requesting now resides at a different location. Example use cases are when your website's domain name has changed, when you want clients to use a canonical URL format (either with or without the **www** prefix), and when you want to catch and correct common misspellings of your domain name. The `return` and `rewrite` directives are suitable for these purposes.

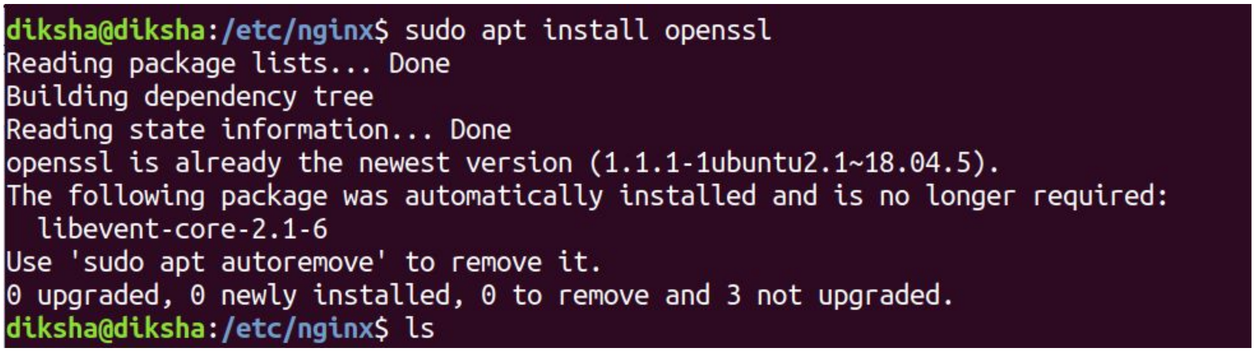
The `return` directive tells NGINX to stop processing the request and immediately send code `301 (Moved Permanently)` and the specified rewritten URL to the client. The rewritten URL uses two NGINX variables to capture and replicate values from the original request URL: `$scheme` is the protocol (**http** or **https**) and `$request_uri` is the full URI including arguments.

D. Perform an HTTP to HTTPS redirection including non-www to www redirection.

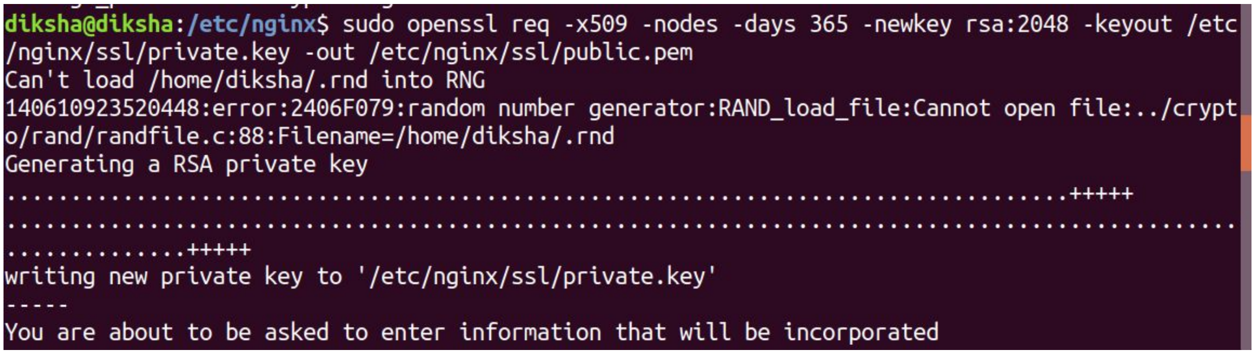
There are many benefits of using HTTPS over HTTP, such as:

- All the data is encrypted in both directions. As a result, sensitive information cannot be read if intercepted.
- Google Chrome and all other popular browsers will mark your website as safe.
- HTTPS allows you to use the HTTP/2 protocol, which significantly improves the site performance.
- Google favors HTTPS websites. Your site will rank better if served via HTTPS.

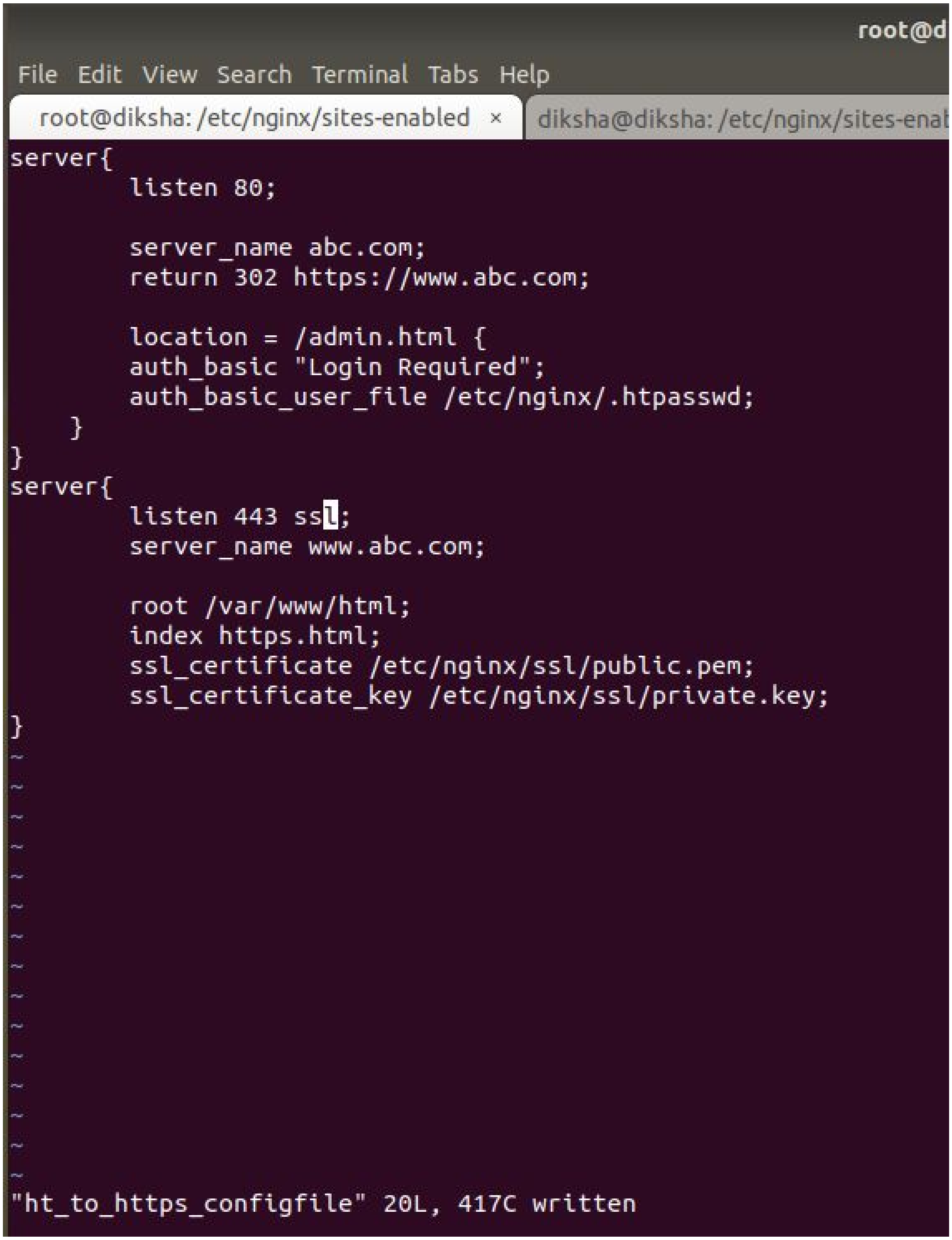
The preferred method to redirect HTTP to HTTPS in Nginx is to configure a separate server block for each version of the site



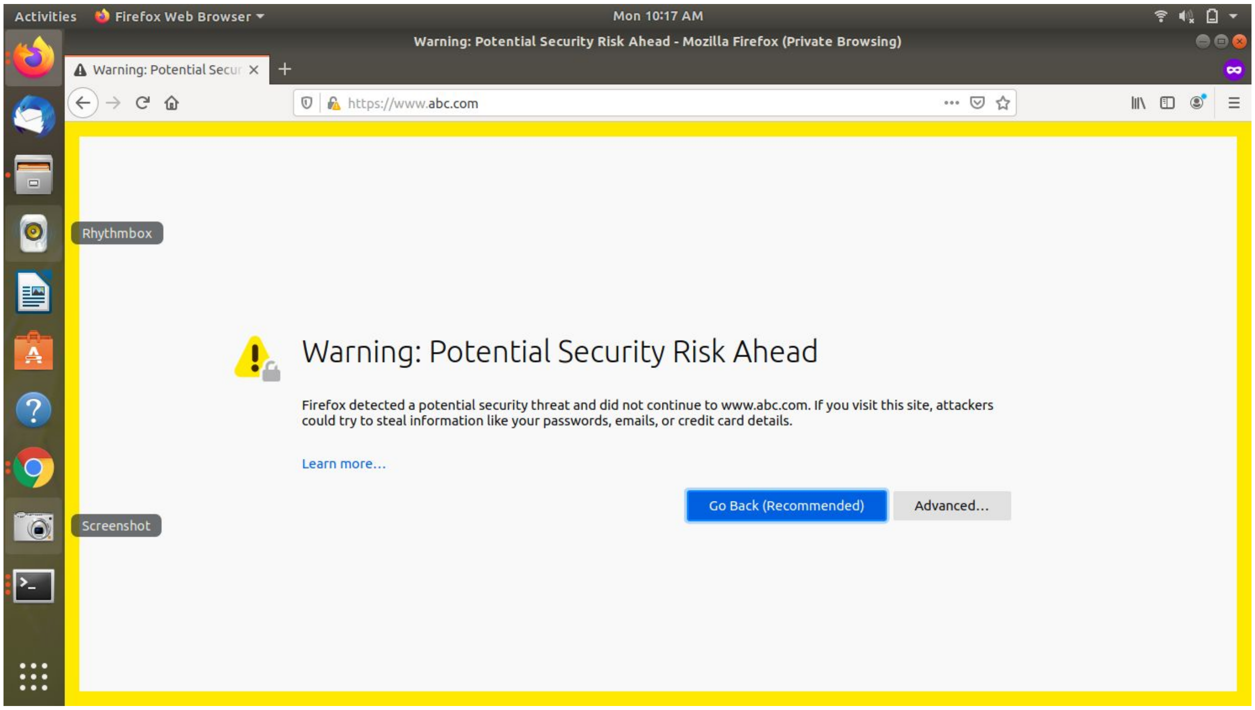
```
diksha@diksha:/etc/nginx$ sudo apt install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1-1ubuntu2.1~18.04.5).
The following package was automatically installed and is no longer required:
  libevent-core-2.1-6
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
diksha@diksha:/etc/nginx$ ls
```



```
diksha@diksha:/etc/nginx$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc
/nginx/ssl/private.key -out /etc/nginx/ssl/public.pem
Can't load /home/diksha/.rnd into RNG
140610923520448:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypt
o/rand/randfile.c:88:Filename=/home/diksha/.rnd
Generating a RSA private key
...................................................................................+++++
................................................................................................
..............+++++
writing new private key to '/etc/nginx/ssl/private.key'
-----
You are about to be asked to enter information that will be incorporated
```

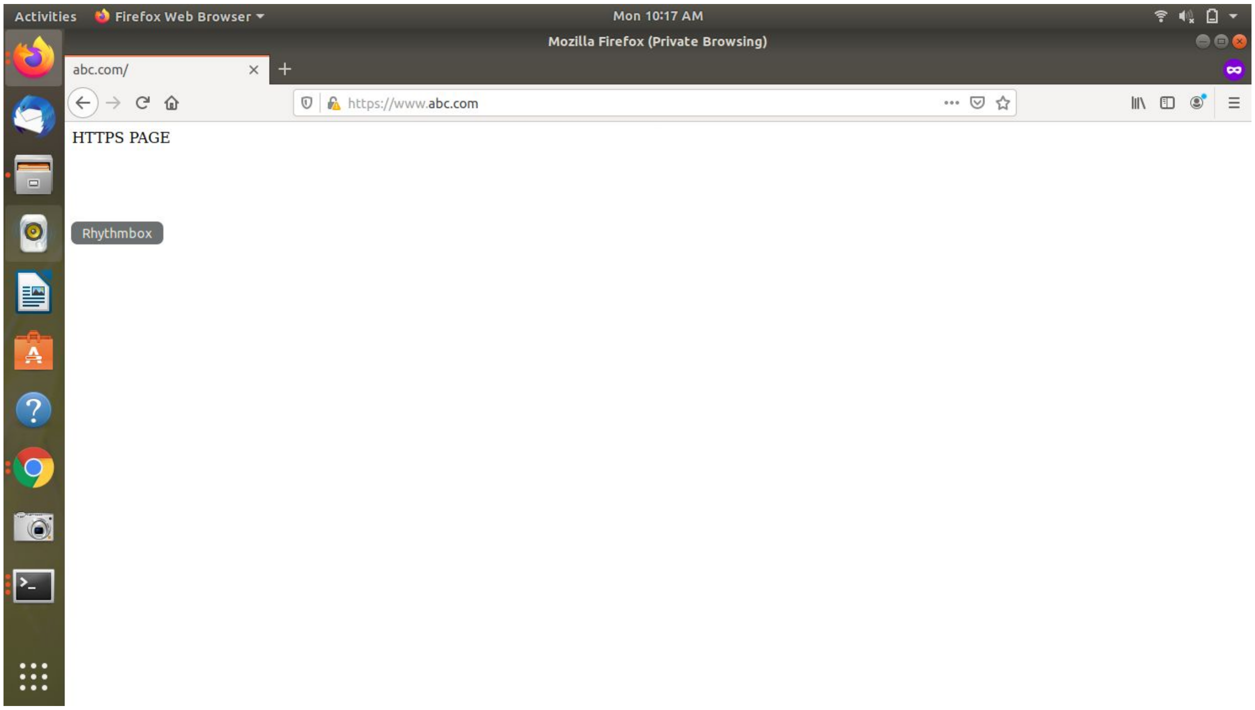

root@d
File Edit View Search Terminal Tabs Help
root@diksha: /etc/nginx/sites-enabled ×
diksha@diksha: /etc/nginx/sites-enab

```
server{
        listen 80;

        server_name abc.com;
        return 302 https://www.abc.com;

        location = /admin.html {
        auth_basic "Login Required";
        auth_basic_user_file /etc/nginx/.htpasswd;
    }
}
server{
        listen 443 ssl;
        server_name www.abc.com;

        root /var/www/html;
        index https.html;
        ssl_certificate /etc/nginx/ssl/public.pem;
        ssl_certificate_key /etc/nginx/ssl/private.key;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"ht_to_https_configfile" 20L, 417C written
```



Warning: Potential Security Risk Ahead

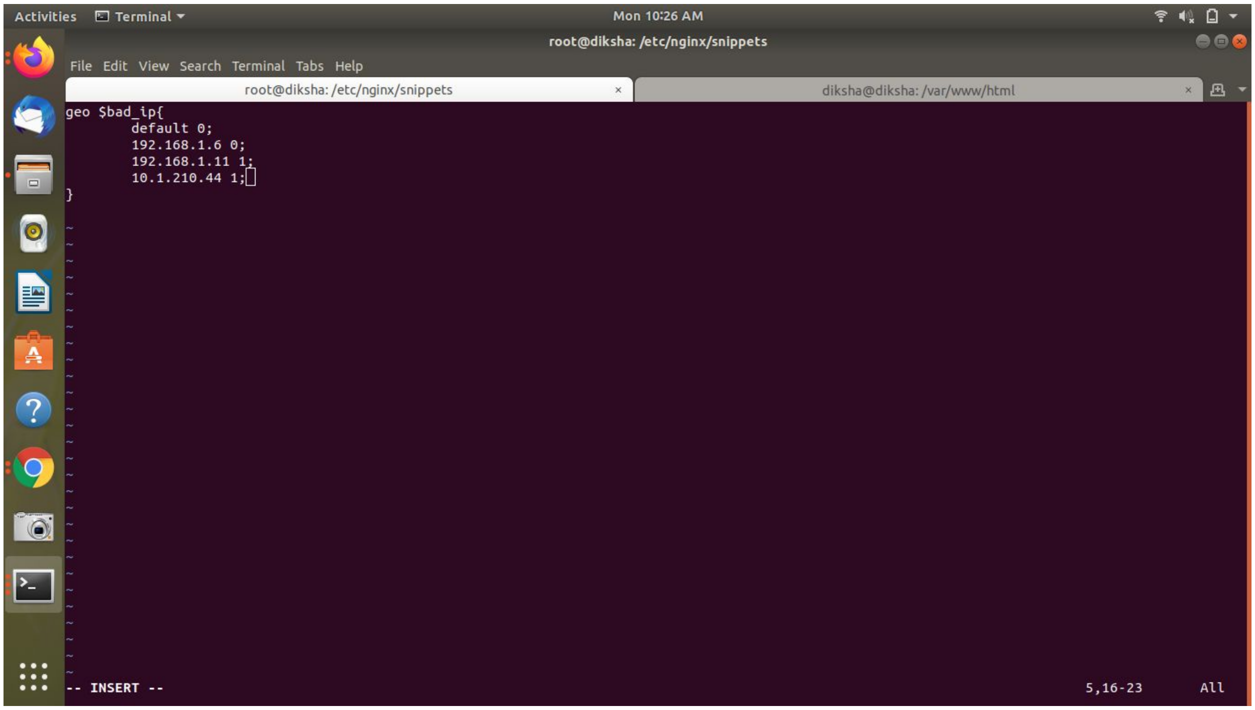
Firefox detected a potential security threat and did not continue to www.abc.com. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.



HTTPS PAGE

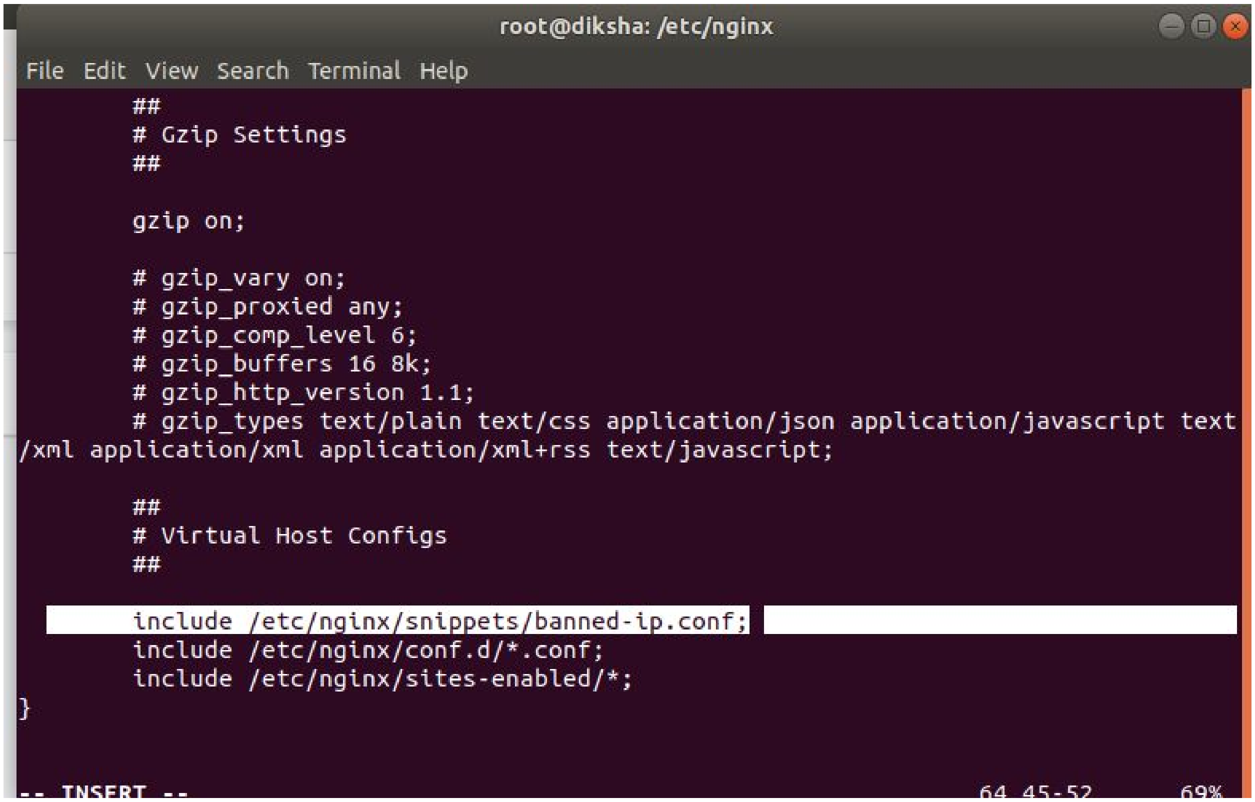
E. Allow access to a set of particular IPs on a location block and return 405 to other IPs no matter if the page in that location exists.

Ans:

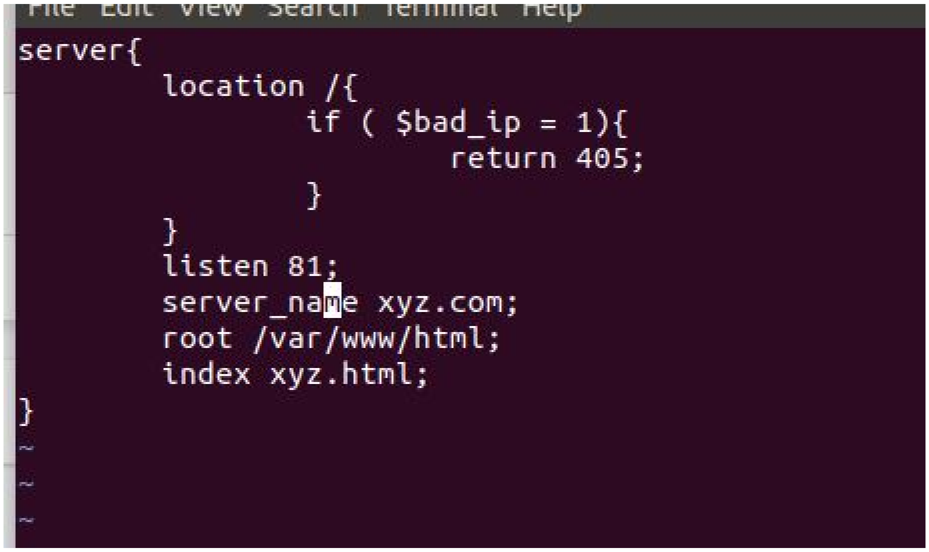


Activities
Terminal
Mon 10:26 AM
root@diksha: /etc/nginx/snippets
File Edit View Search Terminal Tabs Help
root@diksha: /etc/nginx/snippets
diksha@diksha: /var/www/html

```
geo $bad_ip{
        default 0;
        192.168.1.6 0;
        192.168.1.11 1;
        10.1.210.44 1;
}

-- INSERT --                                                                    5,16-23       All
```

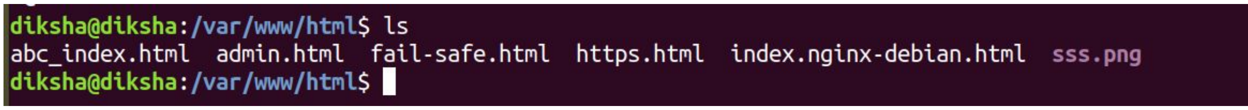


```
        ##
        # Gzip Settings
        ##

        gzip on;

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript text
/xml application/xml application/xml+rss text/javascript;

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/snippets/banned-ip.conf;
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}

-- INSERT --                                          64,45-52      69%
```

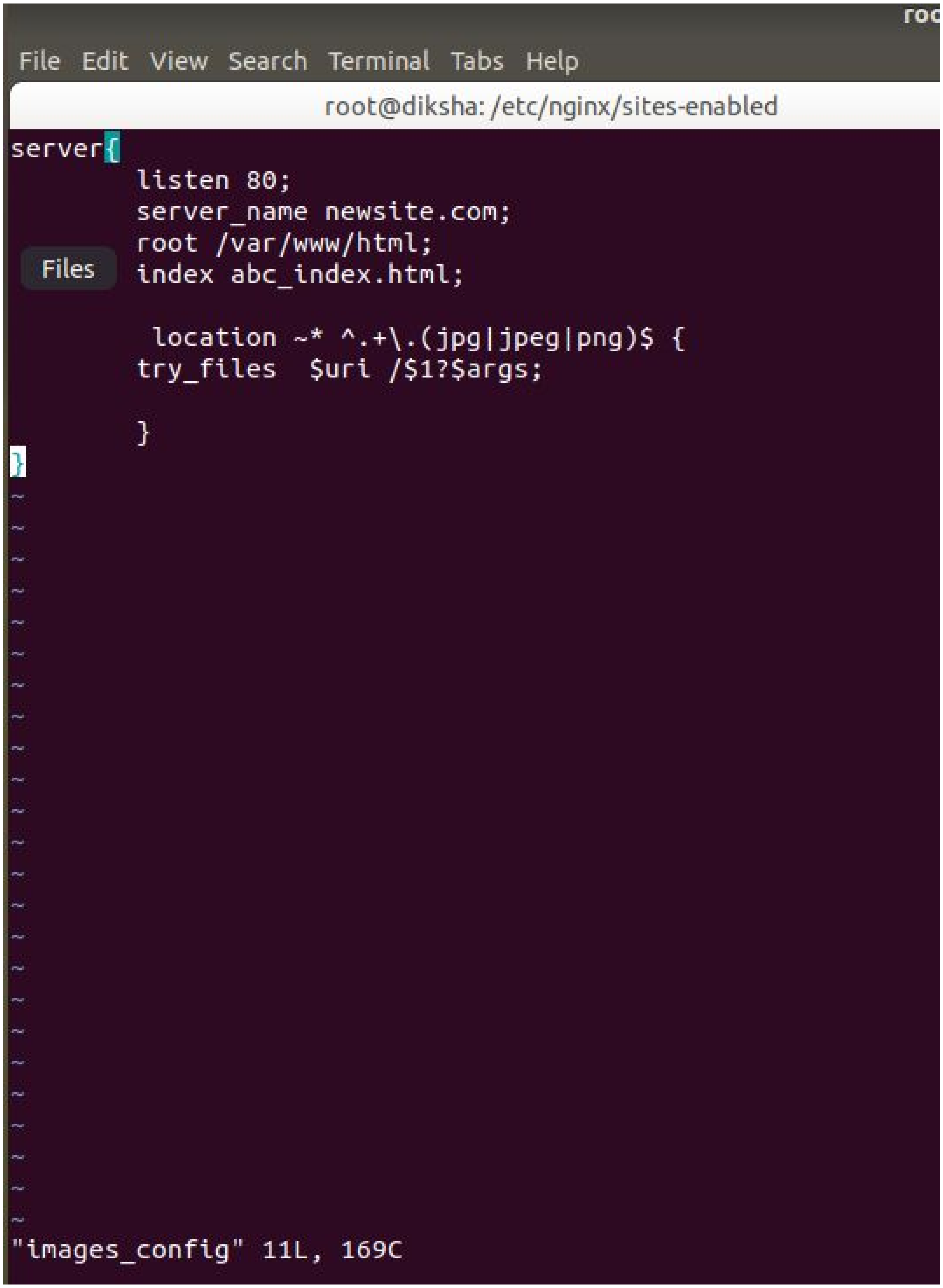


```
server{
        location /{
                if ( $bad_ip = 1){
                        return 405;
                }
        }
        listen 81;
        server_name xyz.com;
        root /var/www/html;
        index xyz.html;
}
~
~
~
```

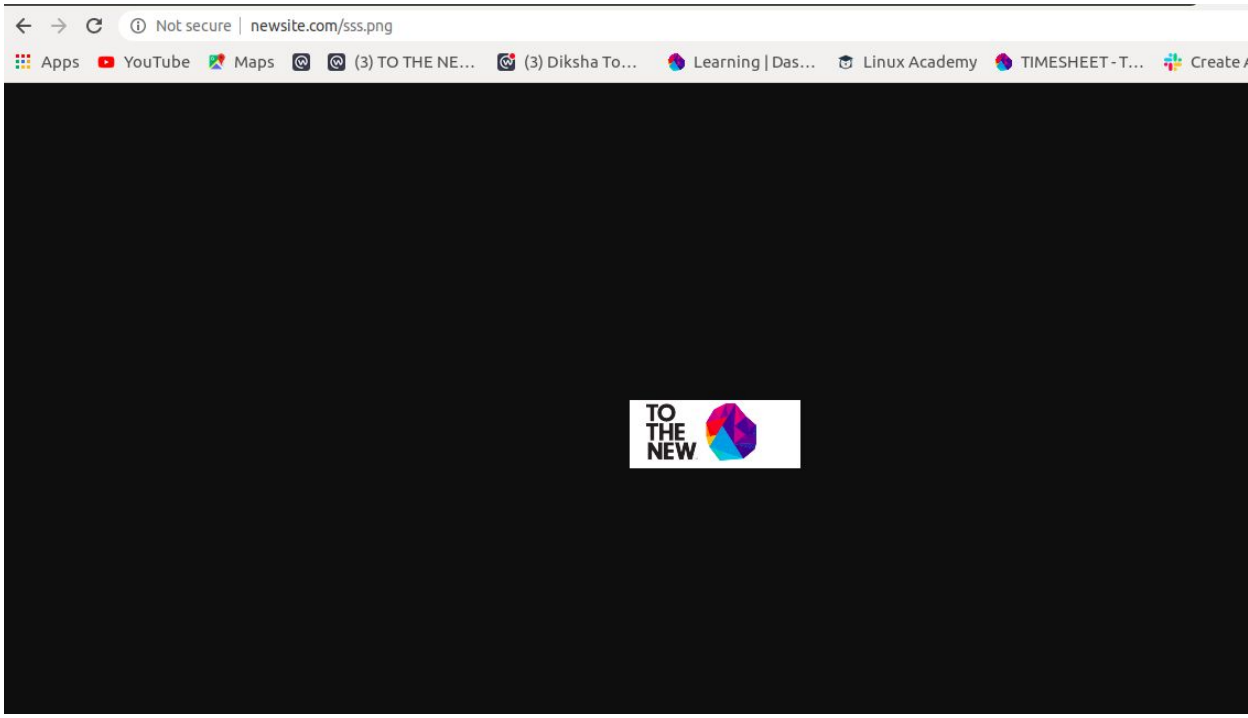
F. Place your images at /var/www/html/images. Only accept jpg/png/jpeg. Discard rest



```
diksha@diksha:/var/www/html$ ls
abc_index.html  admin.html  fail-safe.html  https.html  index.nginx-debian.html  sss.png
diksha@diksha:/var/www/html$ 
```



```
server{
        listen 80;
        server_name newsite.com;
        root /var/www/html;
        index abc_index.html;

         location ~* ^.+\.(jpg|jpeg|png)$ {
        try_files  $uri /$1?$args;

        }
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"images_config" 11L, 169C
```



G.Create a load balancer with 5 backends. Explain different types of load balancing methods.

Ans:

a. Round Robin (Default)

b. Least Connection:

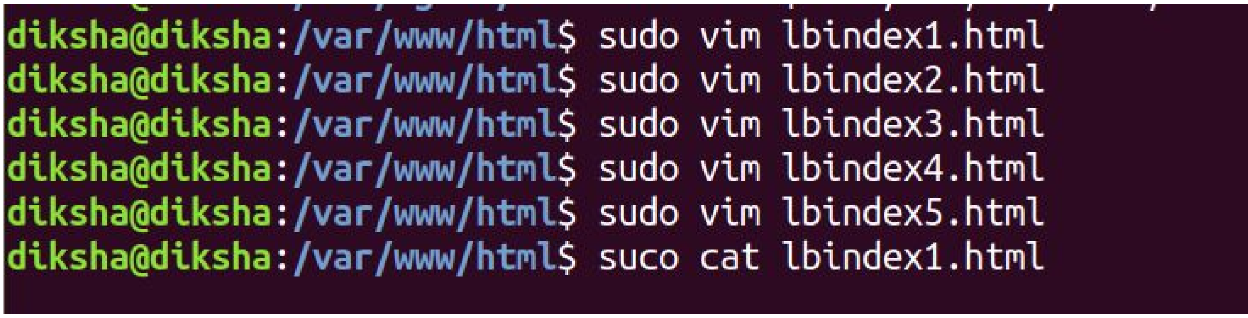
```
upstream backend{

least_conn;

server backend1.example.com;

server backend2.example.com;

}
```

c. IP Hash

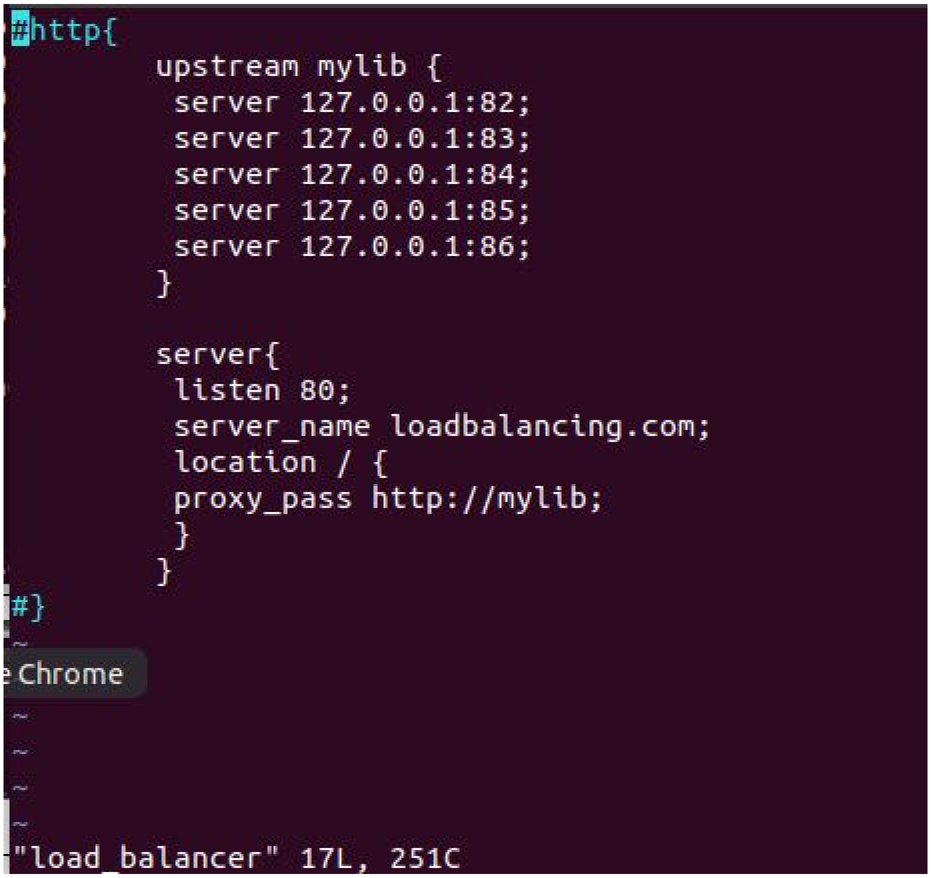
```
upstream backend{

ip_hash;

server backend1.example.com;

server backend2.example.com;

}
```

d. Least time:

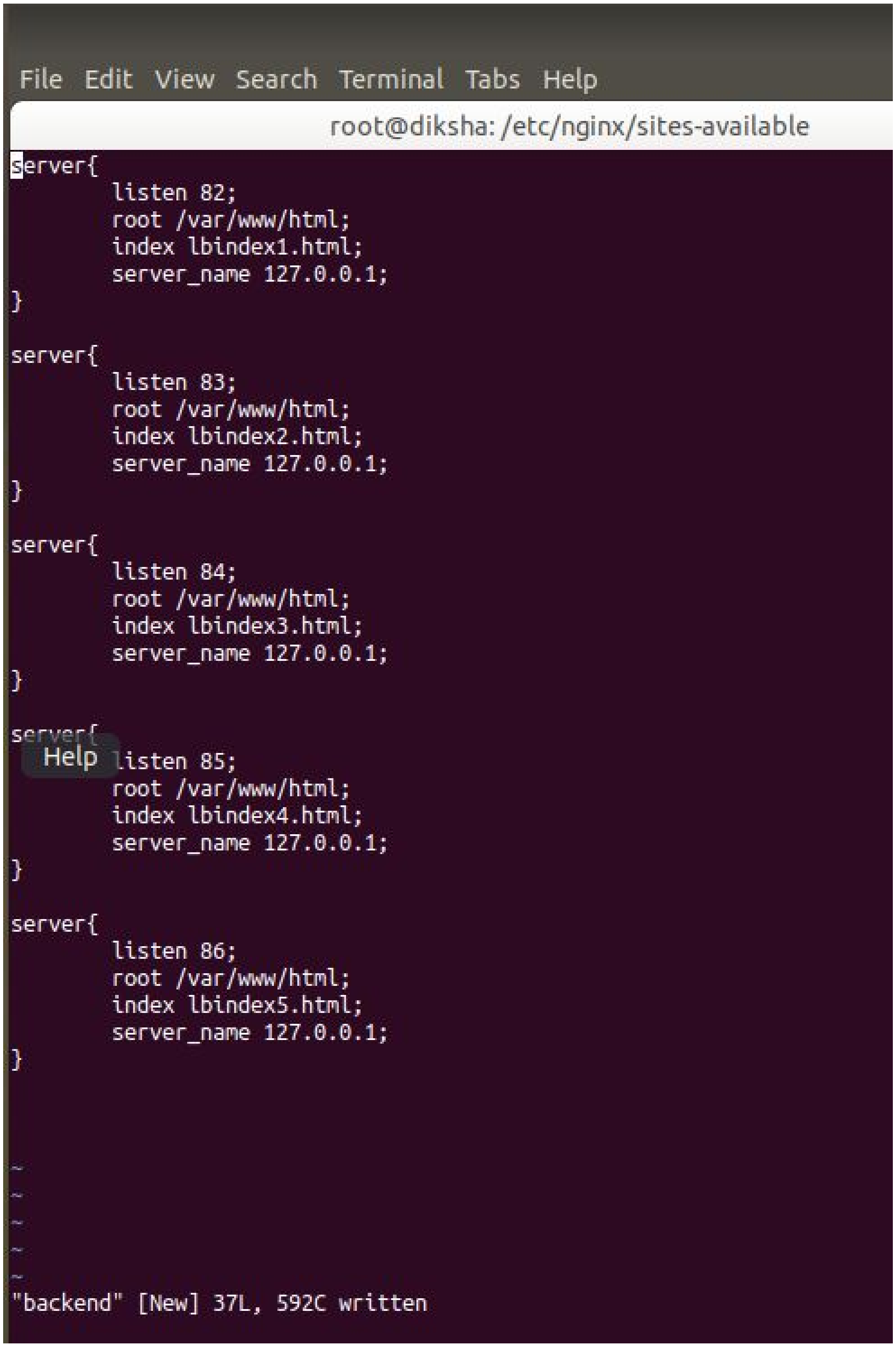
```
upstream backend{

least_time header;

server backend1.example.com;

server backend2.example.com;

}
```



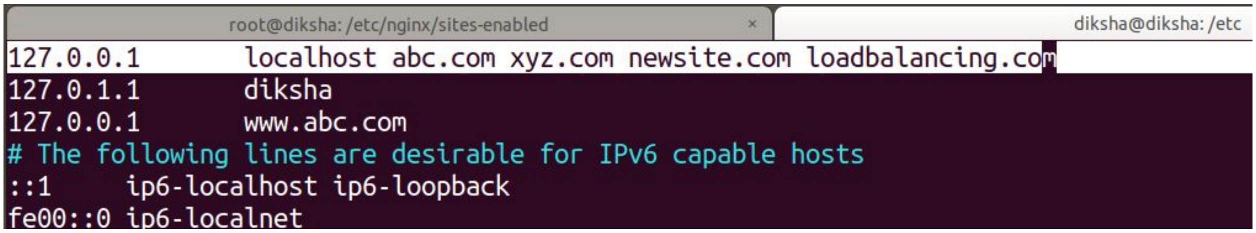
```
diksha@diksha:/var/www/html$ sudo vim lbindex1.html
diksha@diksha:/var/www/html$ sudo vim lbindex2.html
diksha@diksha:/var/www/html$ sudo vim lbindex3.html
diksha@diksha:/var/www/html$ sudo vim lbindex4.html
diksha@diksha:/var/www/html$ sudo vim lbindex5.html
diksha@diksha:/var/www/html$ suco cat lbindex1.html
```



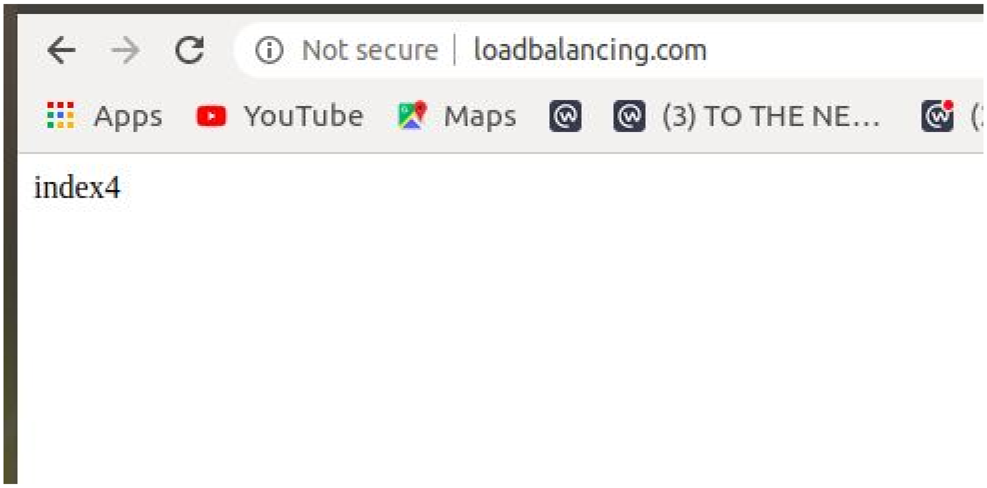
```
#http{
        upstream mylib {
         server 127.0.0.1:82;
         server 127.0.0.1:83;
         server 127.0.0.1:84;
         server 127.0.0.1:85;
         server 127.0.0.1:86;
        }

        server{
         listen 80;
         server_name loadbalancing.com;
         location / {
         proxy_pass http://mylib;
         }
        }
#}
~
~
~
~
~
"load_balancer" 17L, 251C
```


File Edit View Search Terminal Tabs Help
root@diksha: /etc/nginx/sites-available
Help

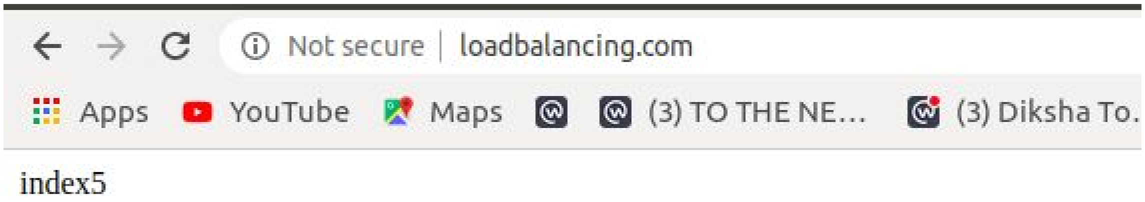
```
server{
        listen 82;
        root /var/www/html;
        index lbindex1.html;
        server_name 127.0.0.1;
}

server{
        listen 83;
        root /var/www/html;
        index lbindex2.html;
        server_name 127.0.0.1;
}

server{
        listen 84;
        root /var/www/html;
        index lbindex3.html;
        server_name 127.0.0.1;
}

server{
        listen 85;
        root /var/www/html;
        index lbindex4.html;
        server_name 127.0.0.1;
}

server{
        listen 86;
        root /var/www/html;
        index lbindex5.html;
        server_name 127.0.0.1;
}



~
~
~
~
~
"backend" [New] 37L, 592C written
```



```
127.0.0.1       localhost abc.com xyz.com newsite.com loadbalancing.com
127.0.1.1       diksha
127.0.0.1       www.abc.com
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
```

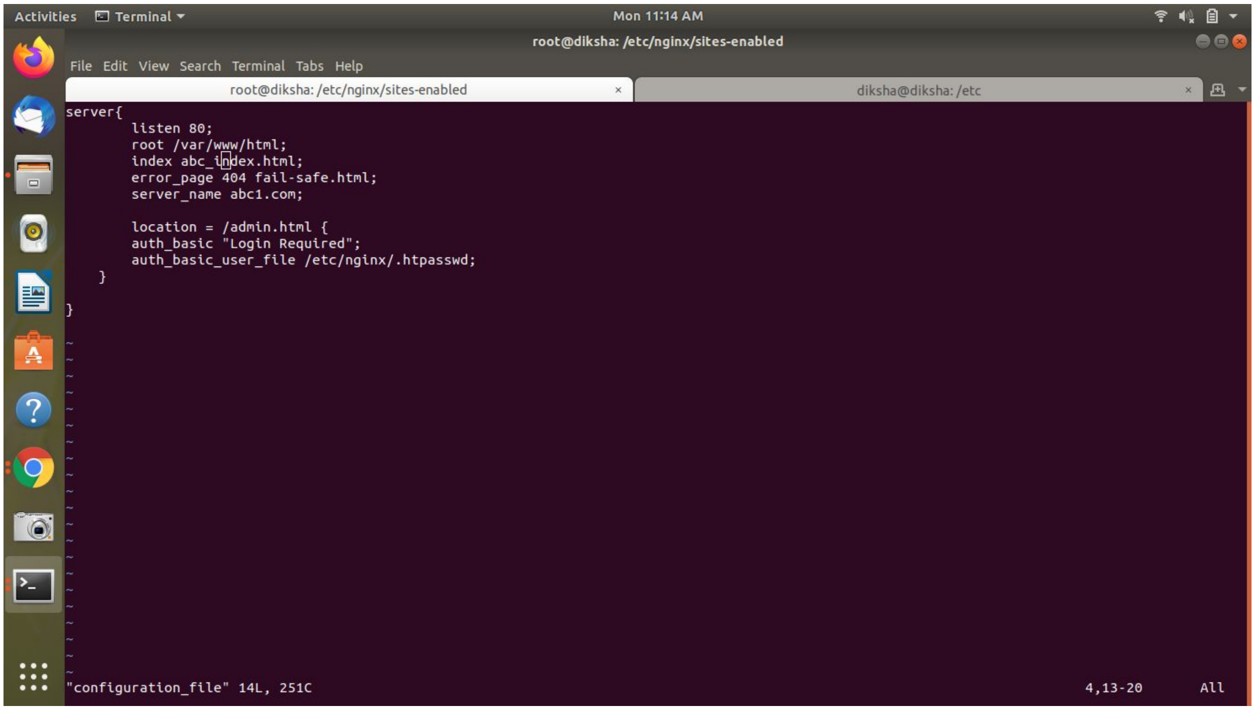


index4

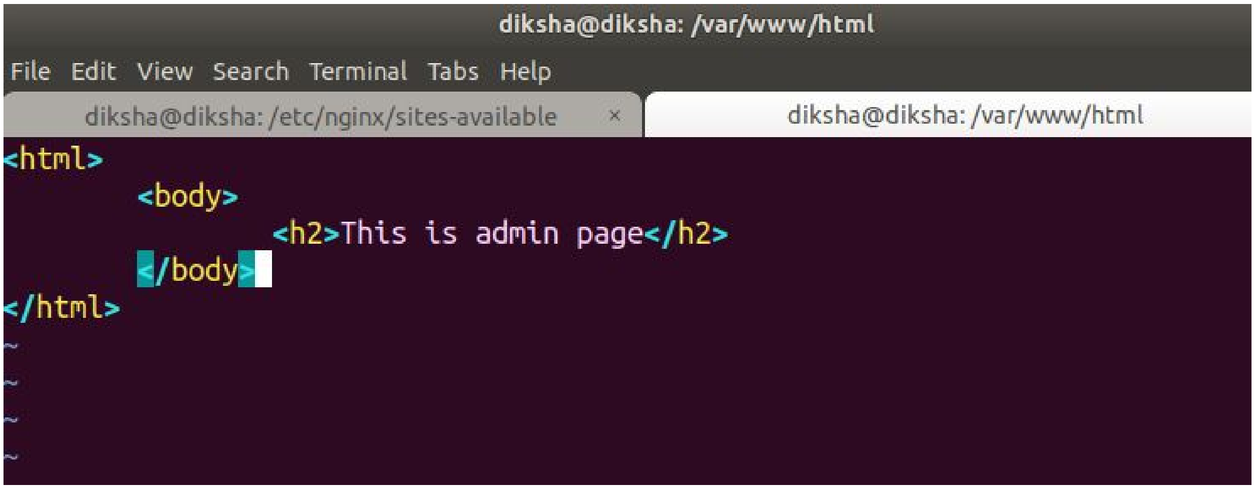


index5

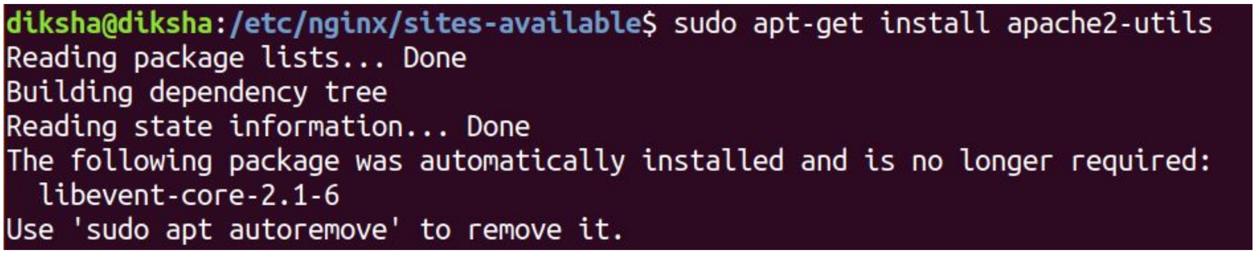
H. Setup Basic Auth (Popup asking for username and password) in a particular location block. (The Basic Auth should not be asked for TTN IP)


Activities
Terminal
Mon 11:14 AM
root@diksha: /etc/nginx/sites-enabled
File Edit View Search Terminal Tabs Help
root@diksha: /etc/nginx/sites-enabled
diksha@diksha: /etc

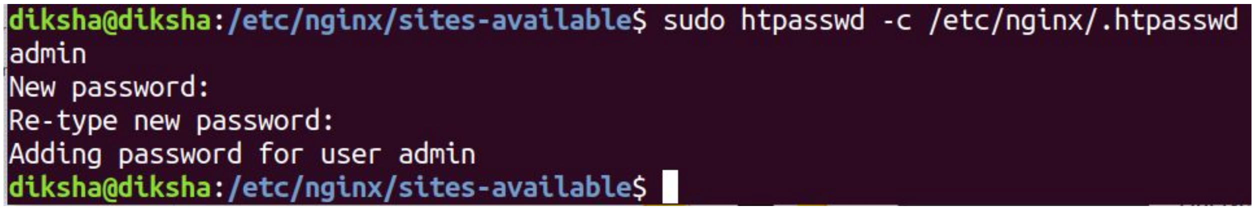
```
server{
        listen 80;
        root /var/www/html;
        index abc_index.html;
        error_page 404 fail-safe.html;
        server_name abc1.com;

        location = /admin.html {
        auth_basic "Login Required";
        auth_basic_user_file /etc/nginx/.htpasswd;
    }

}
~
"configuration_file" 14L, 251C                                    4,13-20       All
```


diksha@diksha: /var/www/html
File Edit View Search Terminal Tabs Help
diksha@diksha: /etc/nginx/sites-available
diksha@diksha: /var/www/html

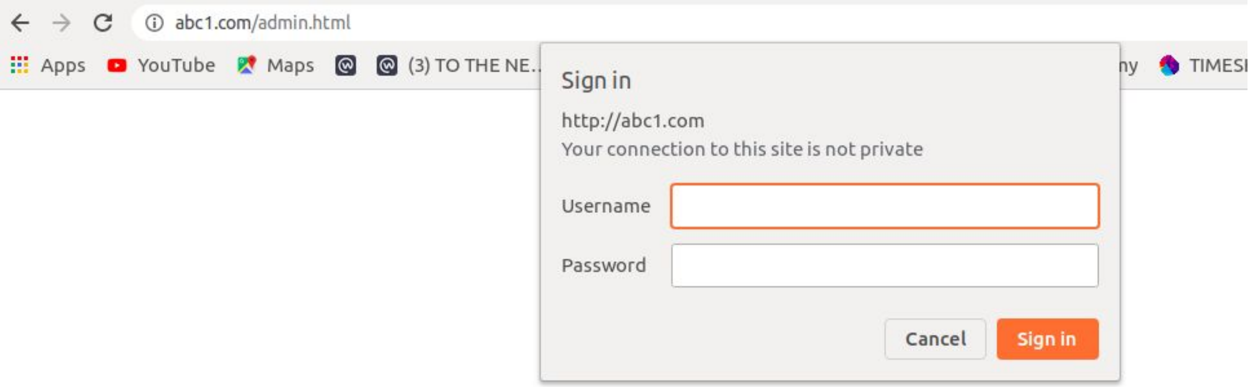
```
<html>
        <body>
                <h2>This is admin page</h2>
        </body>
</html>
~
~
~
~
```



```
diksha@diksha:/etc/nginx/sites-available$ sudo apt-get install apache2-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libevent-core-2.1-6
Use 'sudo apt autoremove' to remove it.
```



```
diksha@diksha:/etc/nginx/sites-available$ sudo htpasswd -c /etc/nginx/.htpasswd
admin
New password:
Re-type new password:
Adding password for user admin
diksha@diksha:/etc/nginx/sites-available$
```



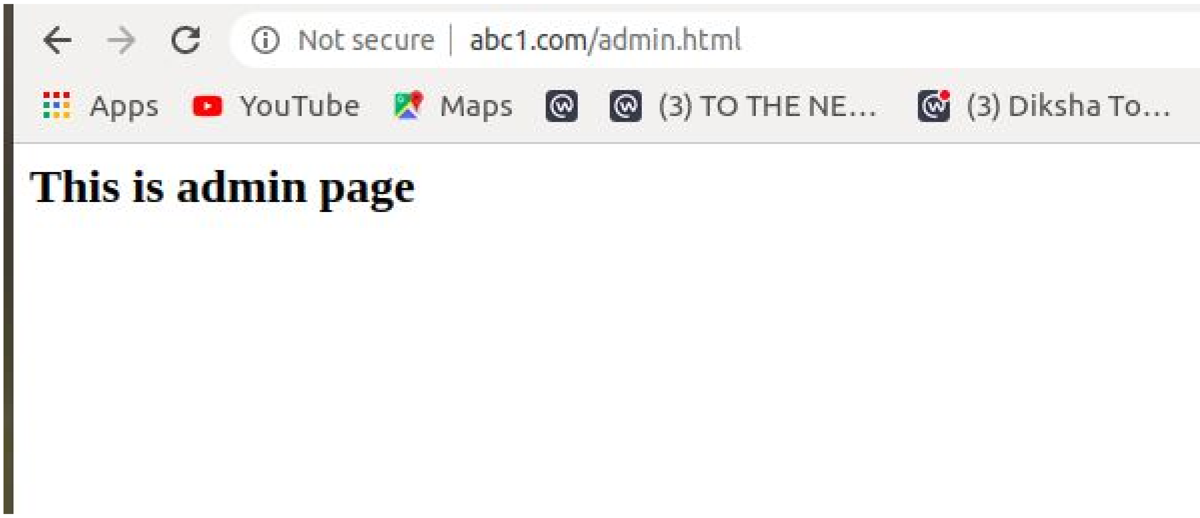
Sign in

http://abc1.com

Your connection to this site is not private

Username

Password



This is admin page