NGINX - WEB SERVER

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

Ans 1. Advantages

A site can benefit in several ways by implementing a reverse proxy server:

- 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.

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

Ans 2 . NGINX is about 2.5 times faster than Apache based on the results of a benchmark test running up to 1,000 concurrent connections. Clearly, NGINX serves static content much faster than Apache. If you need to serve a lot of static content at high concurrency levels, NGINX can be a real help.

it can handle a high volume of connections, **NGINX** is commonly used as a reverse proxy and load balancer to manage incoming traffic and distribute it to slower upstream servers – anything from legacy database servers to microservices.

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

Ans 3. Worker nodes 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.

Defines maximum number of simultaneous connection.
Default value is 768.

Maximum number of connections = worker_processes * worker_connections.

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

Ans 4.
conf .d
Sites-enabled
From the above directory NGINX automatically load server (virtual host) configurations when using the default /etc/nginx/nginx.conf configuration.

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

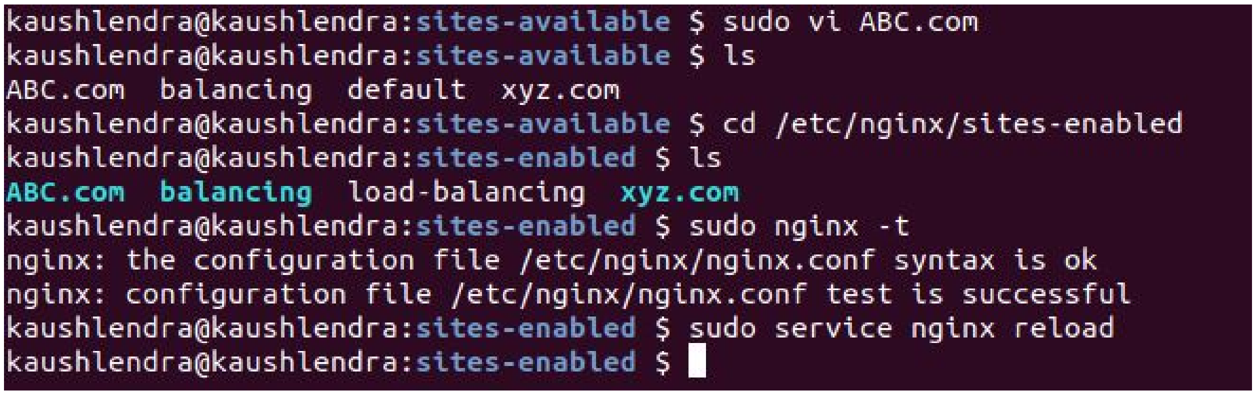
Ans 5. access_log /path/to/file format(Optional)

```
Log format :

log_format combined
'$remote_addr - $remote_user [$time_local]'
'"$request" $status $body_bytes_sent '
  '"$http_referer" "$http_user_agent"';
```

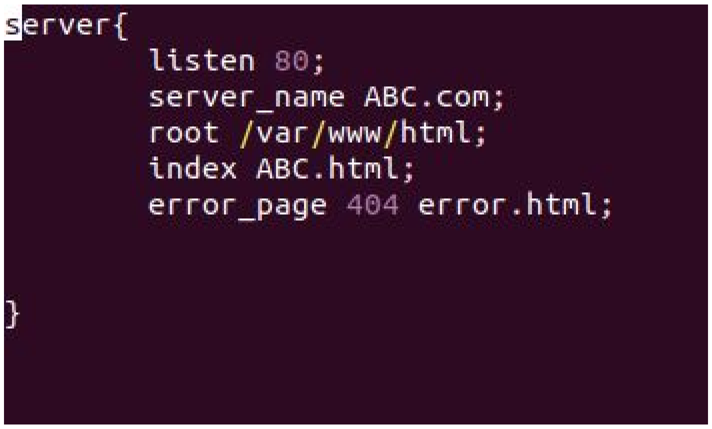
Ques 6. Host a site ABC.COM

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

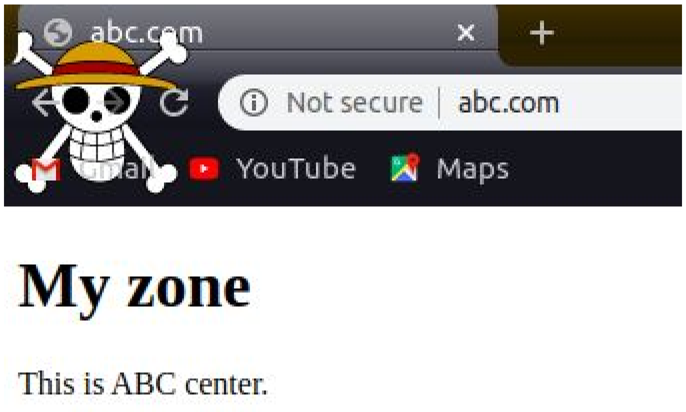
Ans 6. 1.



```
kaushlendra@kaushlendra:sites-available $ sudo vi ABC.com
kaushlendra@kaushlendra:sites-available $ ls
ABC.com  balancing  default  xyz.com
kaushlendra@kaushlendra:sites-available $ cd /etc/nginx/sites-enabled
kaushlendra@kaushlendra:sites-enabled $ ls
ABC.com  balancing  load-balancing  xyz.com
kaushlendra@kaushlendra:sites-enabled $ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
kaushlendra@kaushlendra:sites-enabled $ sudo service nginx reload
kaushlendra@kaushlendra:sites-enabled $ 
```

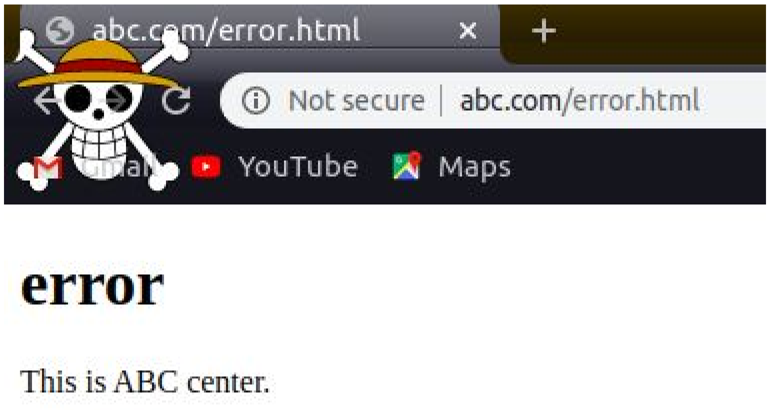


```
server{
        listen 80;
        server_name ABC.com;
        root /var/www/html;
        index ABC.html;
        error_page 404 error.html;


}
```



My zone

This is ABC center.

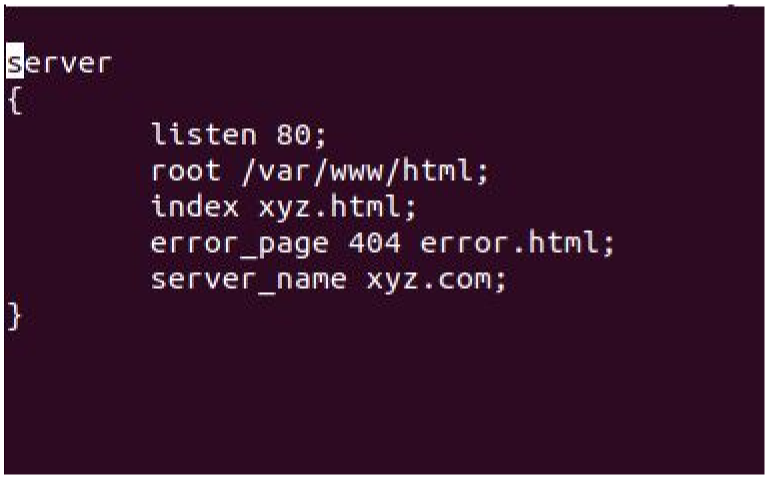


error

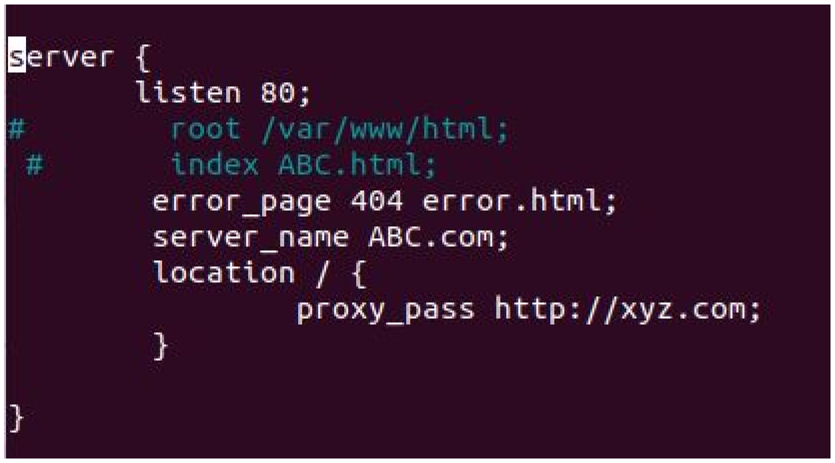
This is ABC center.

Ques 6.2 proxy pass to a website xyz.com on a particular URI.

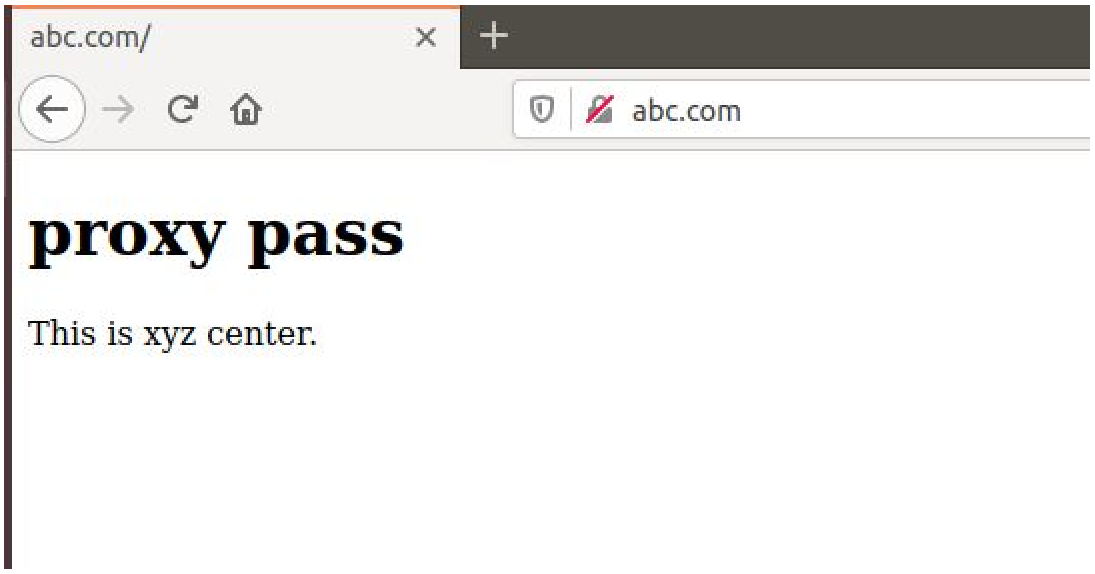
Ans 6.2



```
server
{
        listen 80;
        root /var/www/html;
        index xyz.html;
        error_page 404 error.html;
        server_name xyz.com;
}
```



```
server {
       listen 80;
#        root /var/www/html;
 #       index ABC.html;
        error_page 404 error.html;
        server_name ABC.com;
        location / {
                proxy_pass http://xyz.com;
        }

}
```

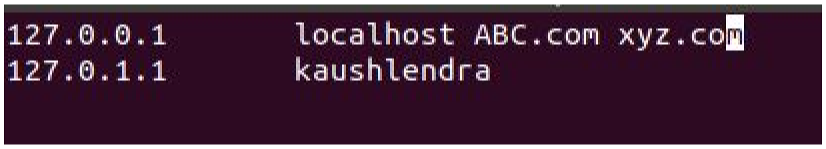


proxy pass

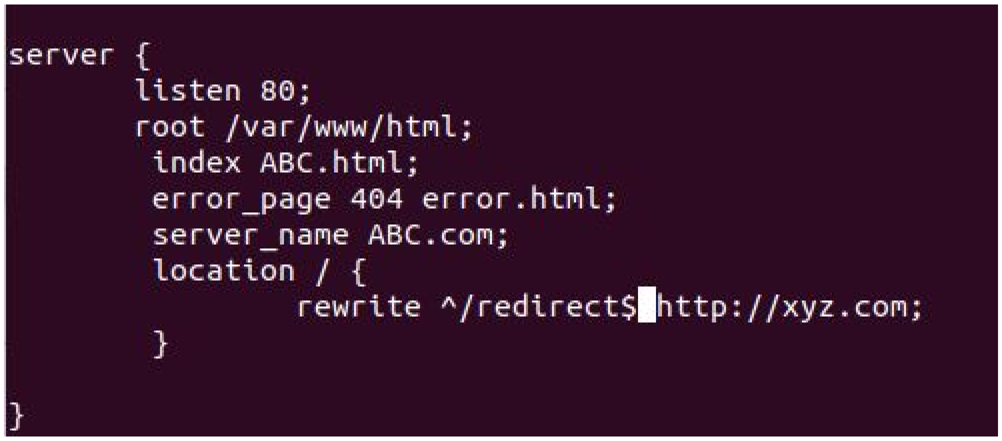
This is xyz center.

QUES 6.3 redirect to above URI on /redirect/

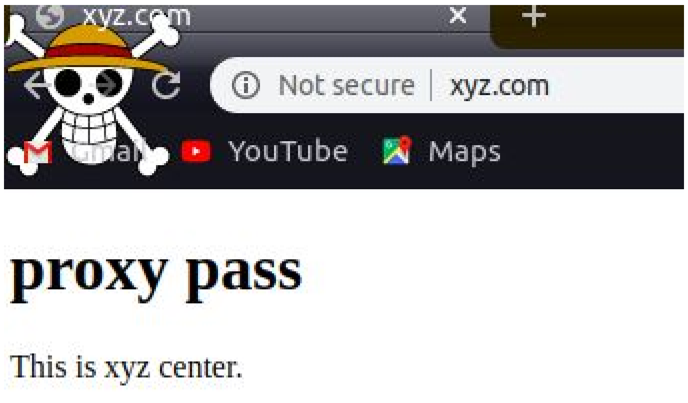
Ans 6.3



```
127.0.0.1       localhost ABC.com xyz.com
127.0.1.1       kaushlendra
```



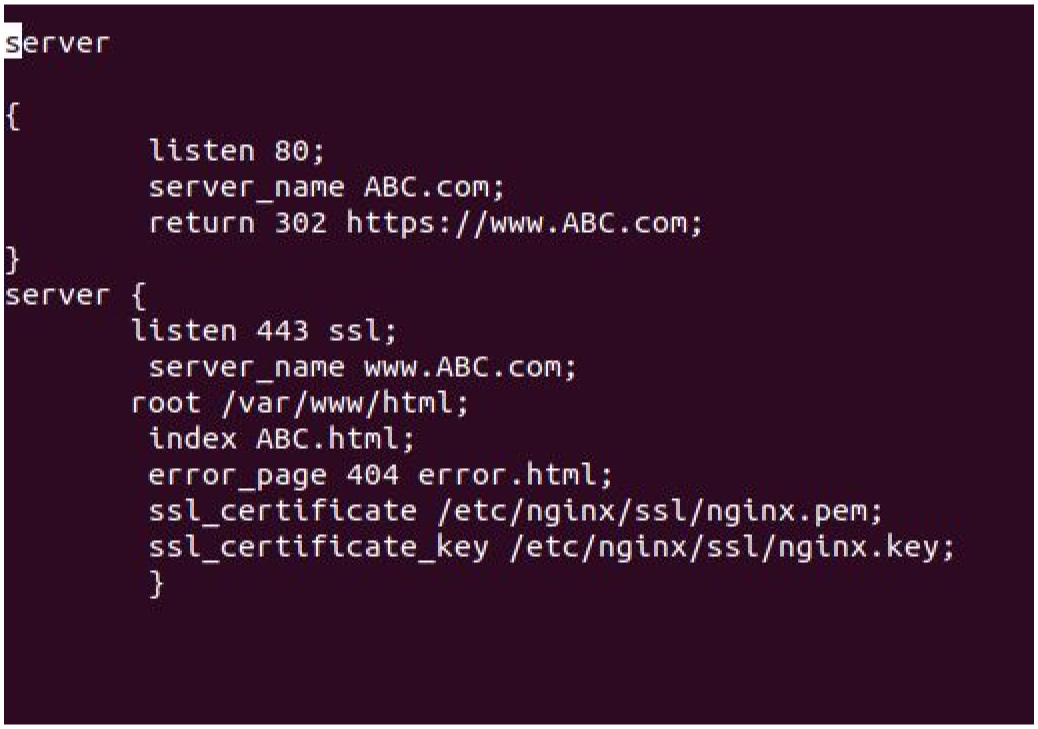
```
server {
       listen 80;
       root /var/www/html;
        index ABC.html;
        error_page 404 error.html;
        server_name ABC.com;
        location / {
                rewrite ^/redirect$ http://xyz.com;
        }

}
```



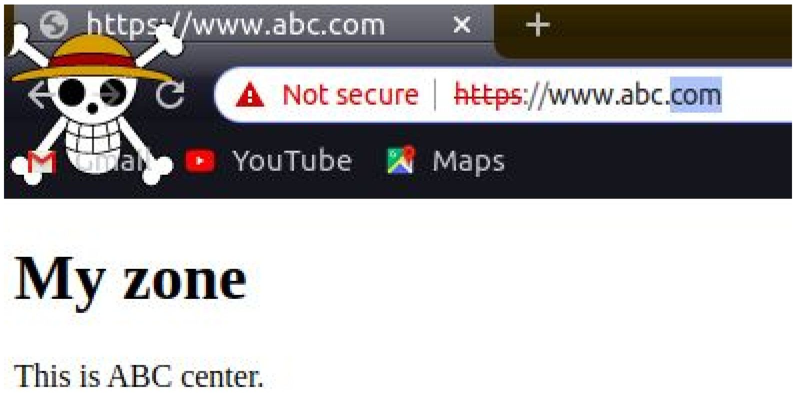
proxy pass

This is xyz center.

Ques 6.4 perform an HTTP to HTTPS redirection including non-www to www redirection.
ANS 6.4



```
server

{
        listen 80;
        server_name ABC.com;
        return 302 https://www.ABC.com;
}
server {
       listen 443 ssl;
        server_name www.ABC.com;
       root /var/www/html;
        index ABC.html;
        error_page 404 error.html;
        ssl_certificate /etc/nginx/ssl/nginx.pem;
        ssl_certificate_key /etc/nginx/ssl/nginx.key;
        }
```



https://www.abc.com

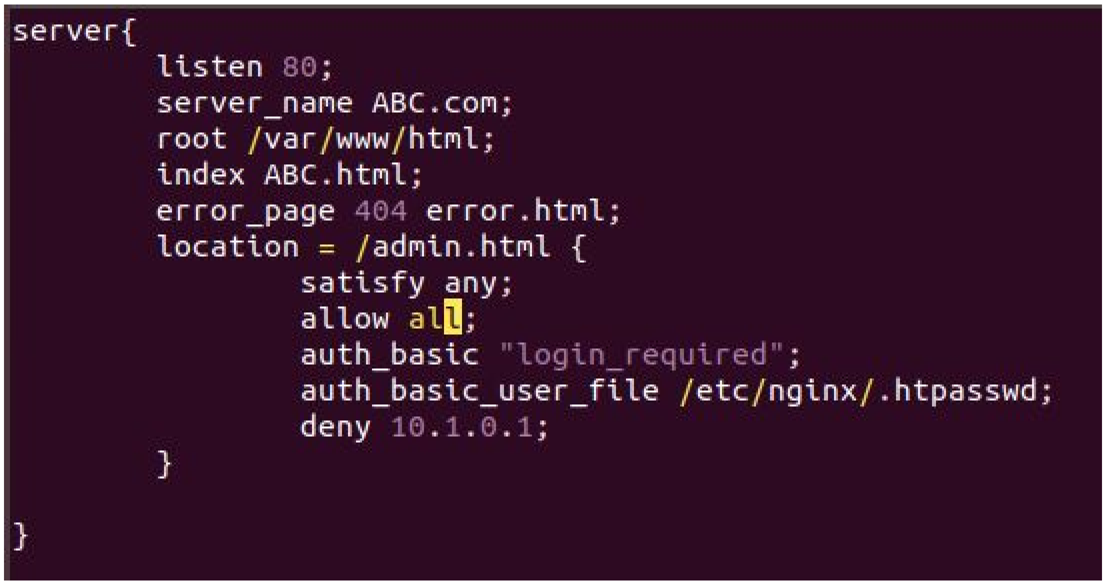
Not secure | ~~https~~://www.abc.com

My zone

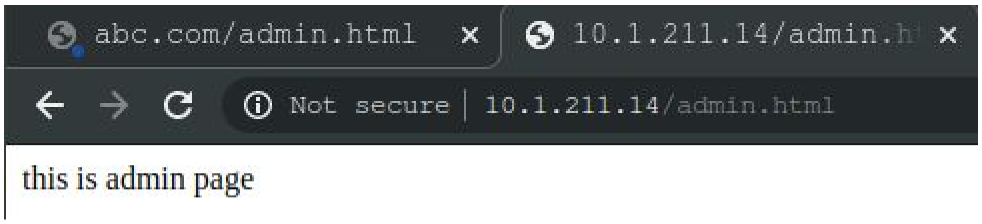
This is ABC center.

Ques 6.5 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 6.5



```
server{
        listen 80;
        server_name ABC.com;
        root /var/www/html;
        index ABC.html;
        error_page 404 error.html;
        location = /admin.html {
                satisfy any;
                allow all;
                auth_basic "login_required";
                auth_basic_user_file /etc/nginx/.htpasswd;
                deny 10.1.0.1;
        }

}
```



this is admin page

Ques 6.6 Place your images at /var/www/html/images. Only accept jpg/png/jpeg. Discard rest

Ans 6.6

```
Location ~(.*.jpeg/.*.png){
        root /var/www/html/images/;
}
```

Ques 7. Create a load balancer with 5 backends. Explain different types of load balancing methods.

Ans 7.

There are three algo used by nginx to distribute the load:-

- **Round-Robin.**
- **Least Connection.**
- **Hashing**

Round Robin

Round Robin is the default load‑balancing technique for both NGINX Plus and NGINX. The load balancer runs through the list of upstream servers in sequence, assigning the next connection request to each one in turn.

```
upstream backend {
    server web1;
    server web2;
    server web3;
}

server {
    server_name www.example.com;

    location / {
       proxy_pass http://backend;
    }
}
```

Hashing

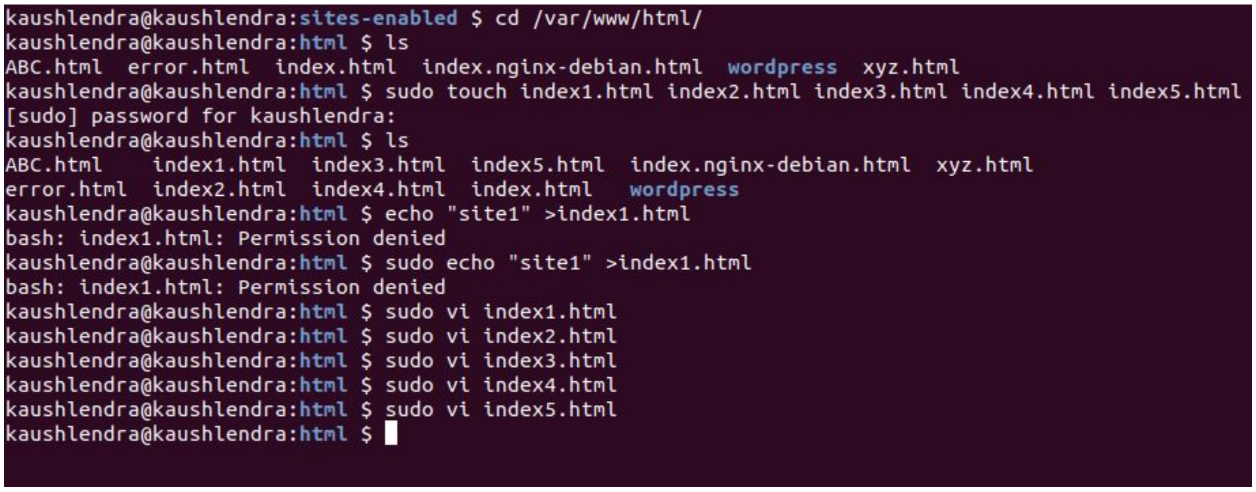
With the Hash method, for each request the load balancer calculates a hash that is based on the combination of text and NGINX variables you specify, and associates the hash with one of the servers. It sends all requests with that hash to that server, so this method establishes a basic kind of session persistence.

```
upstream backend {

   hash $scheme$request_uri;



   server web1;

   server web2;

   server web3;

}



server {

   server_name www.example.com;



   location / {

      proxy_pass http://backend;

   }
}
```

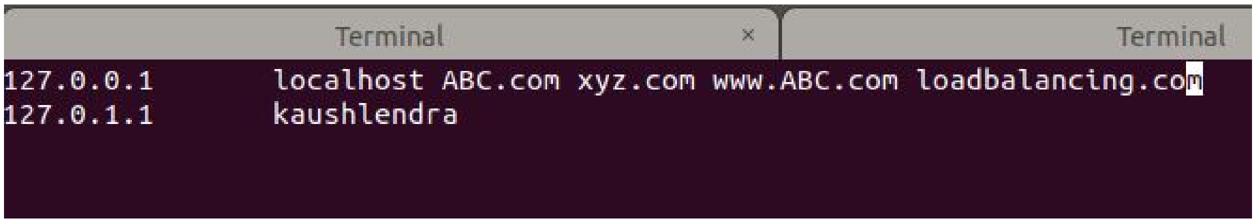
Least Connections

With the Least Connections method, the load balancer compares the current number of active connections it has to each server, and sends the request to the server with the fewest connections. You configure it with the `least_conn` directive.

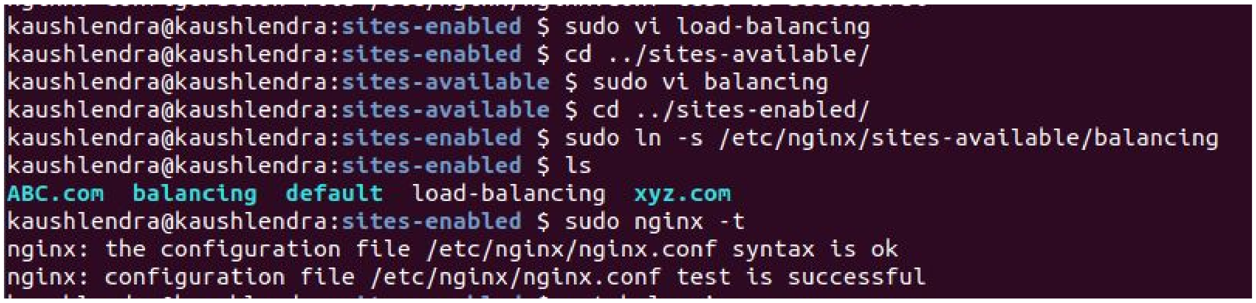
```
upstream backend {

    least_conn;



    server web1;

    server web2;

    server web3;

}



server {

    server_name www.example.com;



    location / {
        proxy_pass http://backend;

    }

}
```



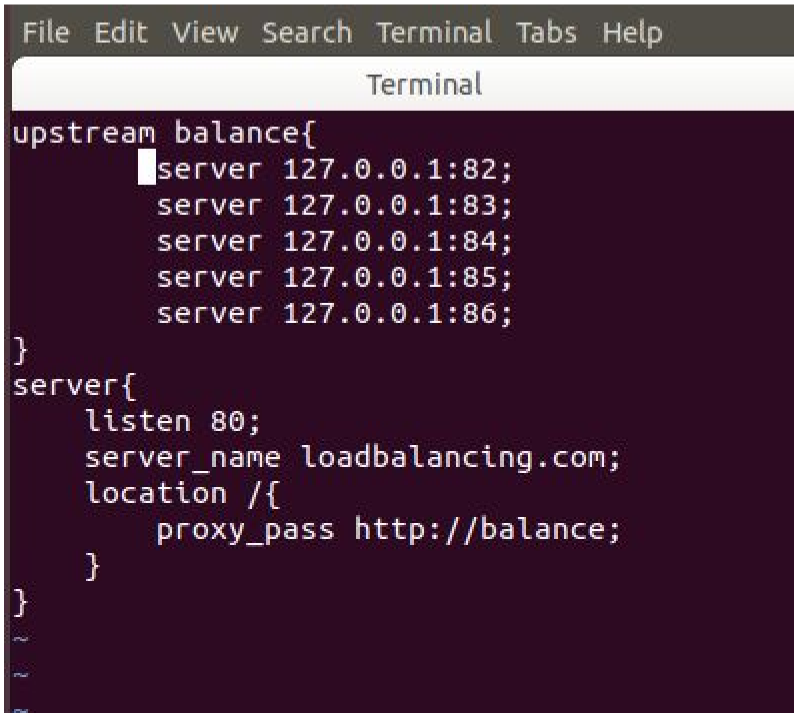
```
kaushlendra@kaushlendra:sites-enabled $ cd /var/www/html/
kaushlendra@kaushlendra:html $ ls
ABC.html  error.html  index.html  index.nginx-debian.html  wordpress  xyz.html
kaushlendra@kaushlendra:html $ sudo touch index1.html index2.html index3.html index4.html index5.html
[sudo] password for kaushlendra:
kaushlendra@kaushlendra:html $ ls
ABC.html    index1.html  index3.html  index5.html  index.nginx-debian.html  xyz.html
error.html  index2.html  index4.html  index.html   wordpress
kaushlendra@kaushlendra:html $ echo "site1" >index1.html
bash: index1.html: Permission denied
kaushlendra@kaushlendra:html $ sudo echo "site1" >index1.html
bash: index1.html: Permission denied
kaushlendra@kaushlendra:html $ sudo vi index1.html
kaushlendra@kaushlendra:html $ sudo vi index2.html
kaushlendra@kaushlendra:html $ sudo vi index3.html
kaushlendra@kaushlendra:html $ sudo vi index4.html
kaushlendra@kaushlendra:html $ sudo vi index5.html
kaushlendra@kaushlendra:html $
```



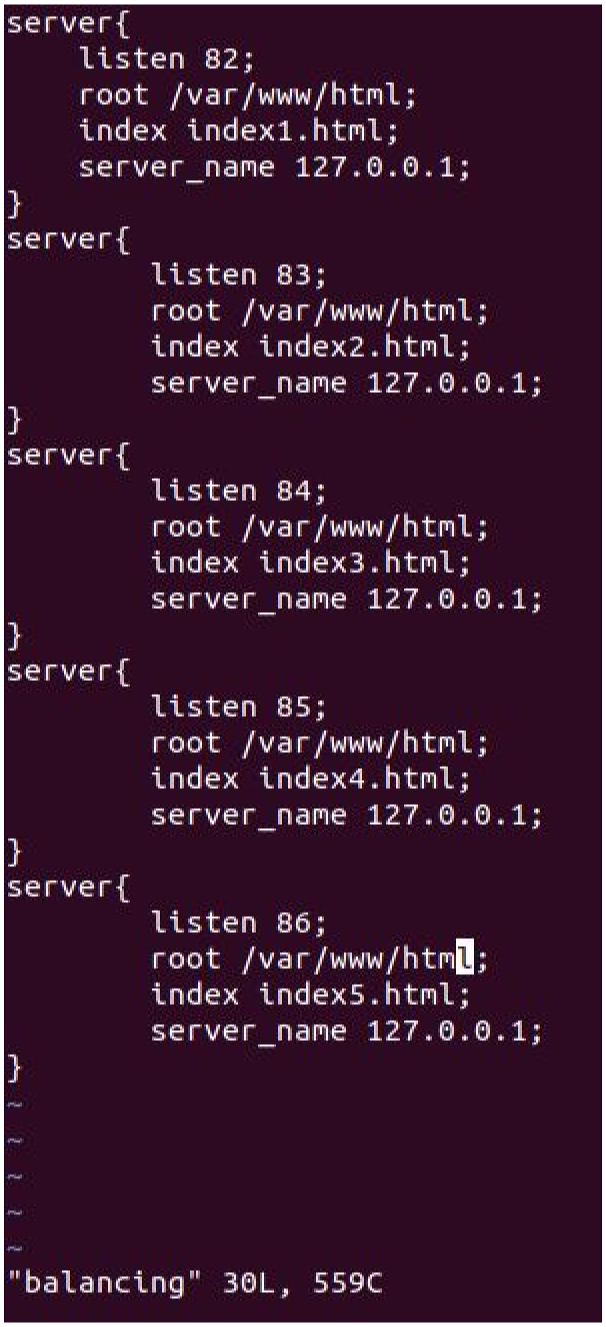
```
127.0.0.1       localhost ABC.com xyz.com www.ABC.com loadbalancing.com
127.0.1.1       kaushlendra
```



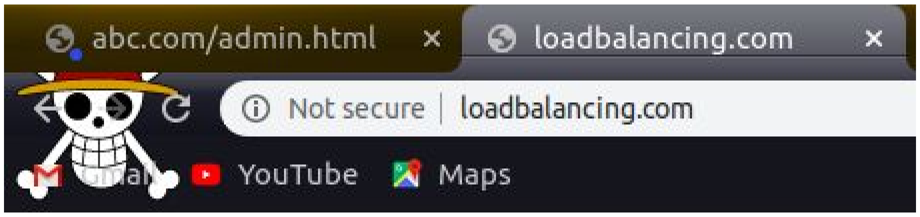
```
kaushlendra@kaushlendra:sites-enabled $ sudo vi load-balancing
kaushlendra@kaushlendra:sites-enabled $ cd ../sites-available/
kaushlendra@kaushlendra:sites-available $ sudo vi balancing
kaushlendra@kaushlendra:sites-available $ cd ../sites-enabled/
kaushlendra@kaushlendra:sites-enabled $ sudo ln -s /etc/nginx/sites-available/balancing
kaushlendra@kaushlendra:sites-enabled $ ls
ABC.com  balancing  default  load-balancing  xyz.com
kaushlendra@kaushlendra:sites-enabled $ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```



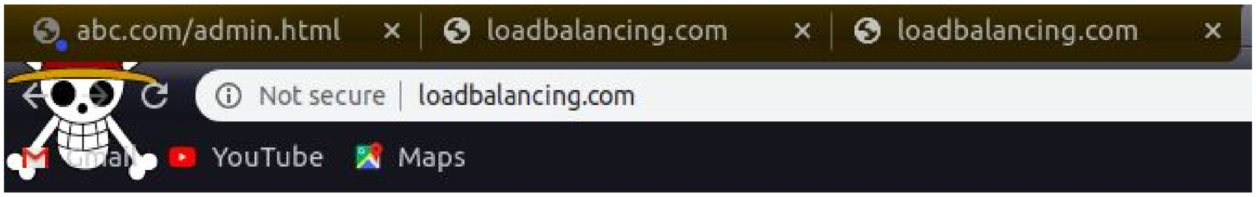
```
upstream balance{
        server 127.0.0.1:82;
        server 127.0.0.1:83;
        server 127.0.0.1:84;
        server 127.0.0.1:85;
        server 127.0.0.1:86;
}
server{
    listen 80;
    server_name loadbalancing.com;
    location /{
        proxy_pass http://balance;
    }
}
~
~
~
```



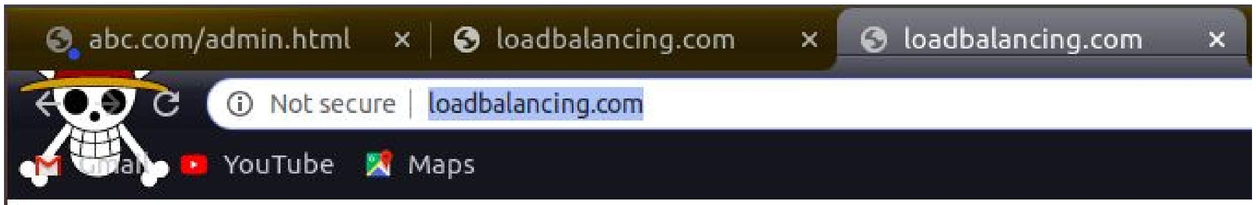
```
server{
    listen 82;
    root /var/www/html;
    index index1.html;
    server_name 127.0.0.1;
}
server{
        listen 83;
        root /var/www/html;
        index index2.html;
        server_name 127.0.0.1;
}
server{
        listen 84;
        root /var/www/html;
        index index3.html;
        server_name 127.0.0.1;
}
server{
        listen 85;
        root /var/www/html;
        index index4.html;
        server_name 127.0.0.1;
}
server{
        listen 86;
        root /var/www/html;
        index index5.html;
        server_name 127.0.0.1;
}
~
~
~
~
~
"balancing" 30L, 559C
```



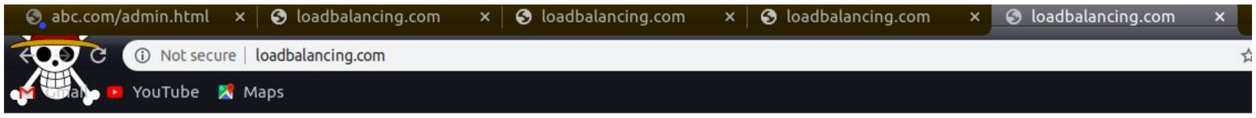
site 1



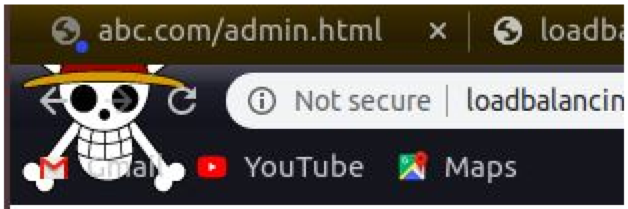
site 2



site 3



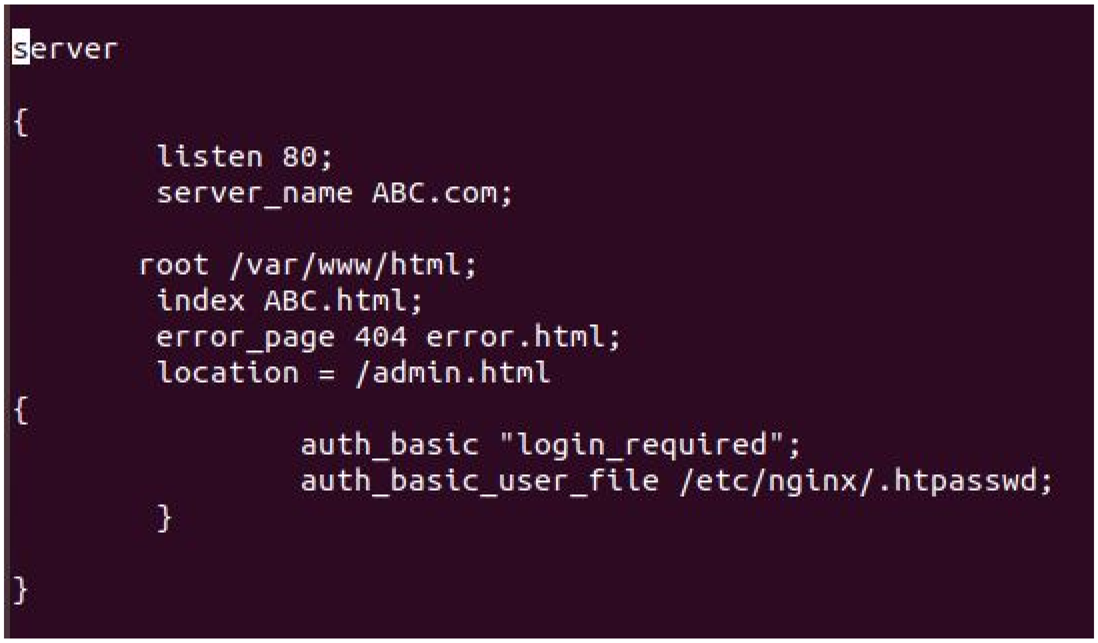
site 4



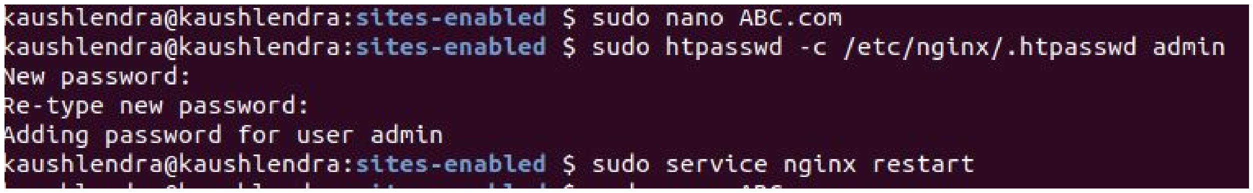
site 5

Ques 8. Setup Basic Auth (Popup asking for username and password) in a particular location block. (The Basic Auth should not be asked for TTN IP)

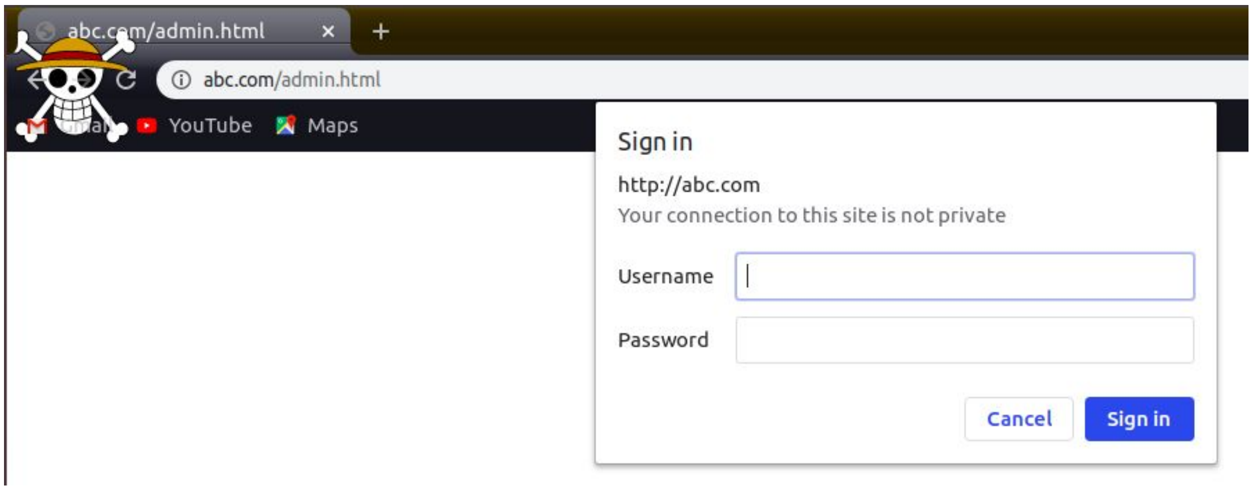
Ans 8.



```
server

{
        listen 80;
        server_name ABC.com;

       root /var/www/html;
        index ABC.html;
        error_page 404 error.html;
        location = /admin.html
{
                auth_basic "login_required";
                auth_basic_user_file /etc/nginx/.htpasswd;
        }

}
```



```
kaushlendra@kaushlendra:sites-enabled $ sudo nano ABC.com
kaushlendra@kaushlendra:sites-enabled $ sudo htpasswd -c /etc/nginx/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
kaushlendra@kaushlendra:sites-enabled $ sudo service nginx restart
```



Sign in

http://abc.com

Your connection to this site is not private

Username

Password