Nginx Web server

Q1. What is the advantage of using a “reverse proxy server”?

Client ------------------> Nginx webserver -----------------------------> main webserver (apache)

<------------------- <-----------------------------

- Load Balancing – A reverse proxy can perform load balancing which helps distribute client requests evenly across backend servers.
- Increased Security - A reverse proxy also acts as a line of defense for your backend servers.
- Better Performance -- Nginx has been known to perform better in delivering static content over Apache.
- Easy Logging and Auditing -- Since there is only one single point of access when a reverse proxy is implemented, this makes logging and auditing much simpler.

Q2. Why and where Nginx is a better choice than apache.

- Nginx have single processing method and apache have multiple processing method .
- In nginx there is no .htaccess file concept and in apache have the concept of .htaccess for directory listing in apache .

Q3.What are worker nodes and worker connections? How to calculate the max server capacity using the above two?

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.

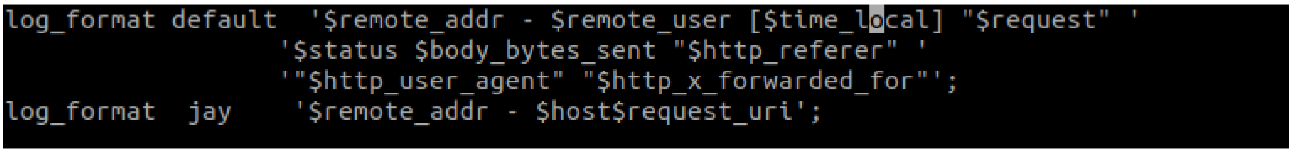
Worker nodes- The number of NGINX worker processes (the default is 1). In most cases, running one worker process per CPU core works well, and we recommend setting this directive to auto to achieve that. There are times when you may want to increase this number, such as when the worker processes have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

Maximum number of connections = worker_processes * worker_connections

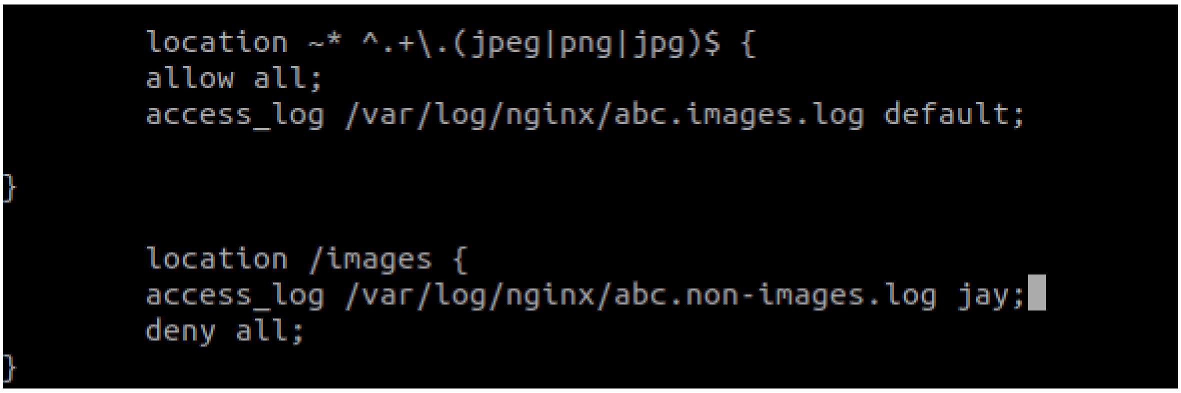
Q4.From what directory will NGINX automatically load server (virtual host) configurations when using the default /etc/nginx/nginx.conf configuration?

/var/www/html

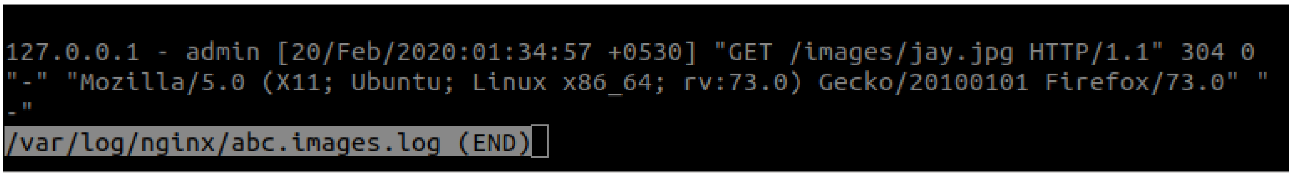
Q5.How to configure different log_format for different “location” block/directive?



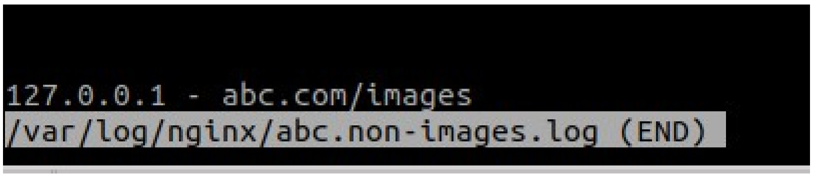
```
log_format default  '$remote_addr - $remote_user [$time_local] "$request" '
                  '$status $body_bytes_sent "$http_referer" '
                  '"$http_user_agent" "$http_x_forwarded_for"';
log_format  jay    '$remote_addr - $host$request_uri';
```



```
        location ~* ^.+\.(jpeg|png|jpg)$ {
        allow all;
        access_log /var/log/nginx/abc.images.log default;


}


        location /images {
        access_log /var/log/nginx/abc.non-images.log jay;
        deny all;
}
```

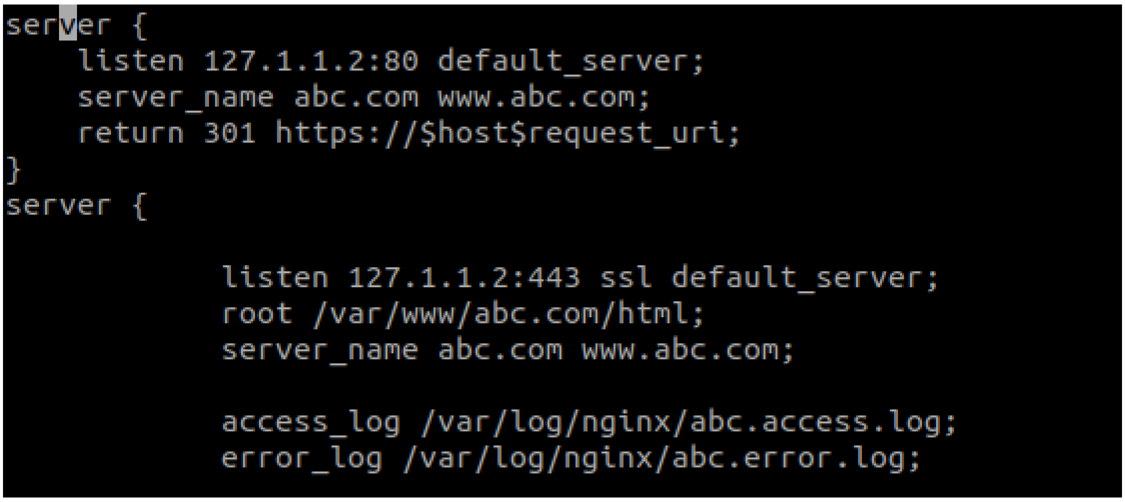


```
127.0.0.1 - admin [20/Feb/2020:01:34:57 +0530] "GET /images/jay.jpg HTTP/1.1" 304 0
"-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0" "
-"
/var/log/nginx/abc.images.log (END)
```



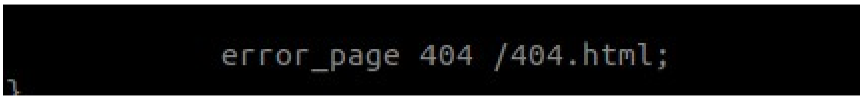
```
127.0.0.1 - abc.com/images
/var/log/nginx/abc.non-images.log (END)
```

Q6.Host a site ABC.COM



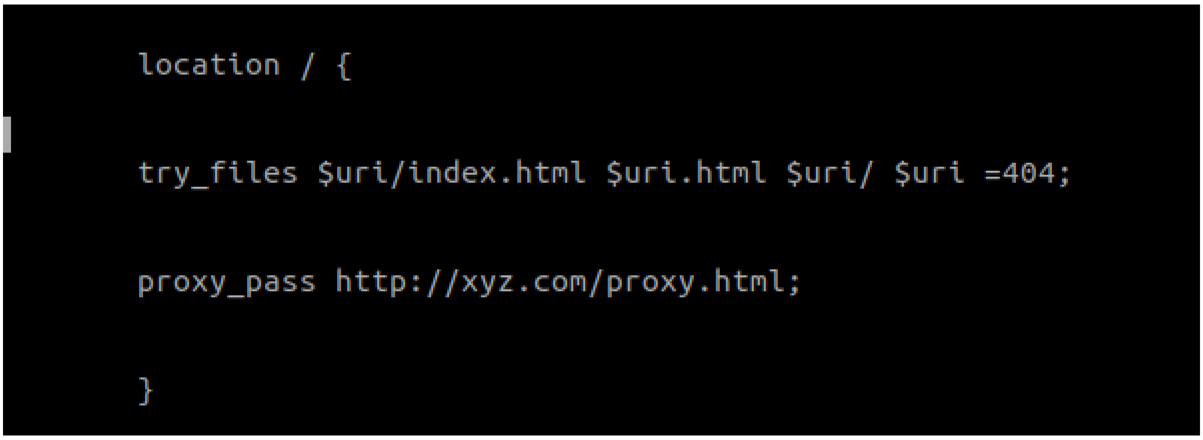
```
server {
    listen 127.1.1.2:80 default_server;
    server_name abc.com www.abc.com;
    return 301 https://$host$request_uri;
}
server {

            listen 127.1.1.2:443 ssl default_server;
            root /var/www/abc.com/html;
            server_name abc.com www.abc.com;

            access_log /var/log/nginx/abc.access.log;
            error_log /var/log/nginx/abc.error.log;
```

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

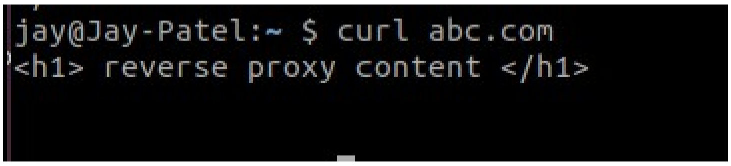


```
            error_page 404 /404.html;
```

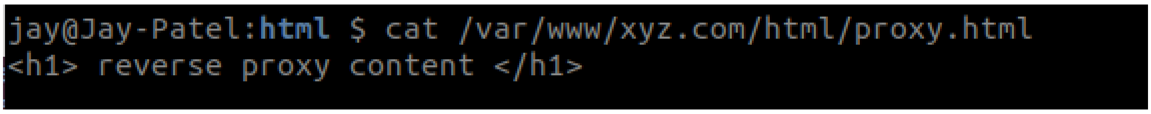
- proxy pass to a website xyz.com on a particular URI.



```
       location / {


       try_files $uri/index.html $uri.html $uri/ $uri =404;


       proxy_pass http://xyz.com/proxy.html;


       }
```

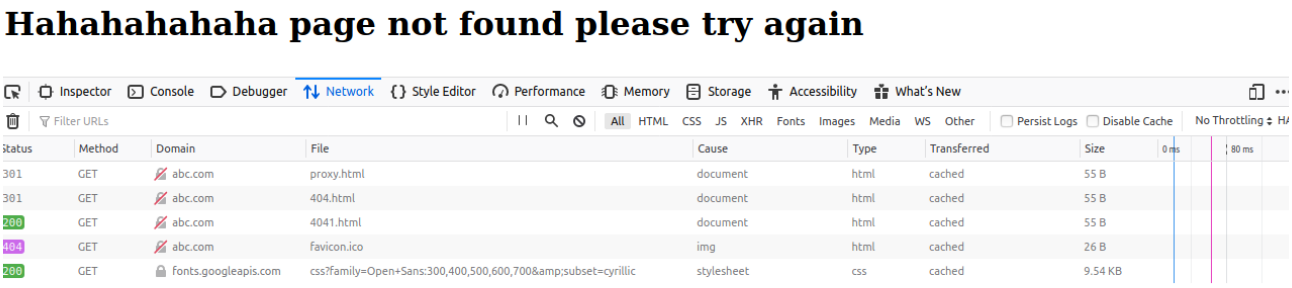


```
jay@Jay-Patel:~ $ curl abc.com
<h1> reverse proxy content </h1>
```



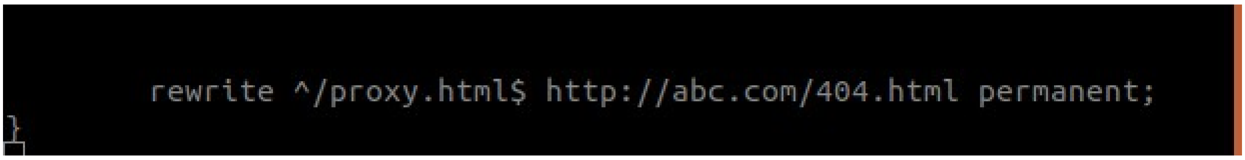
```
jay@Jay-Patel:html $ cat /var/www/xyz.com/html/proxy.html
<h1> reverse proxy content </h1>
```

- redirect to above URI on /redirect/



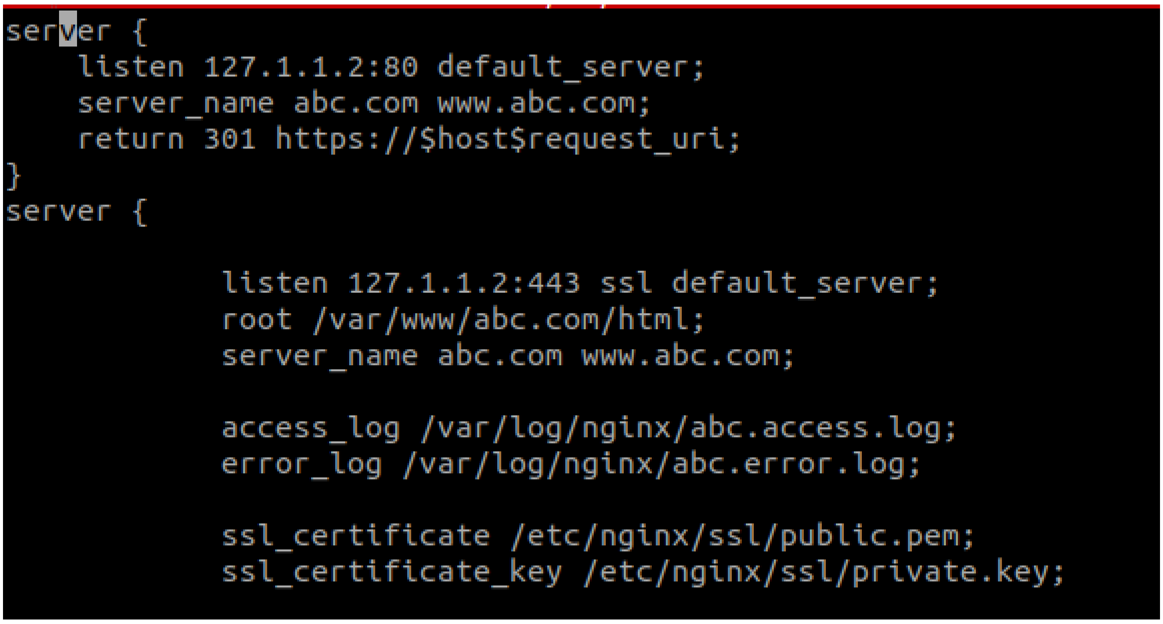
Hahahahahaha page not found please try again

Status	Method	Domain	File	Cause	Type	Transferred	Size
301	GET	abc.com	proxy.html	document	html	cached	55 B
301	GET	abc.com	404.html	document	html	cached	55 B
200	GET	abc.com	4041.html	document	html	cached	55 B
404	GET	abc.com	favicon.ico	img	html	cached	26 B
200	GET	fonts.googleapis.com	css?family=Open+Sans:300,400,500,600,700&amp;subset=cyrillic	stylesheet	css	cached	9.54 KB



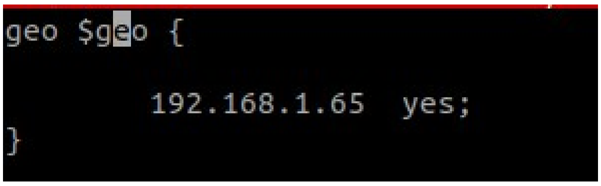
```
        rewrite ^/proxy.html$ http://abc.com/404.html permanent;
}
```

- perform an HTTP to HTTPS redirection including non-www to www redirection.

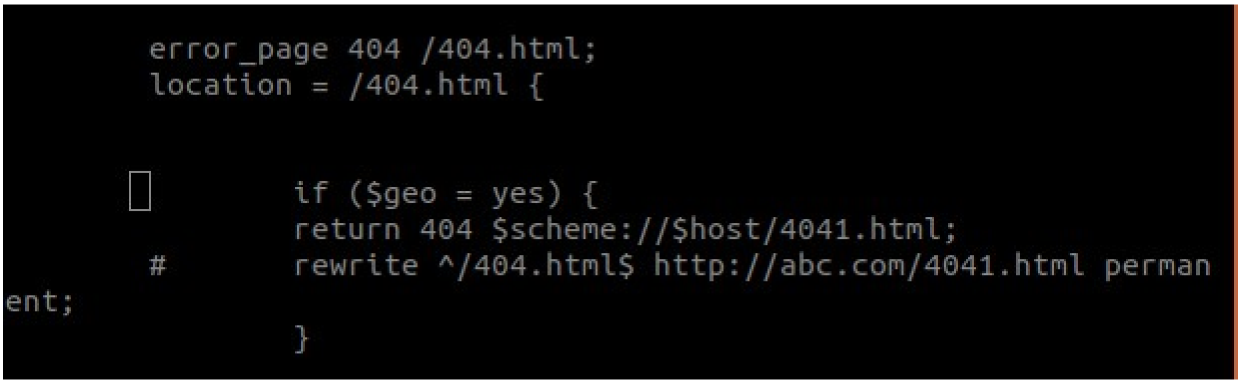


```
server {
    listen 127.1.1.2:80 default_server;
    server_name abc.com www.abc.com;
    return 301 https://$host$request_uri;
}
server {

            listen 127.1.1.2:443 ssl default_server;
            root /var/www/abc.com/html;
            server_name abc.com www.abc.com;

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

            ssl_certificate /etc/nginx/ssl/public.pem;
            ssl_certificate_key /etc/nginx/ssl/private.key;
```

- 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.

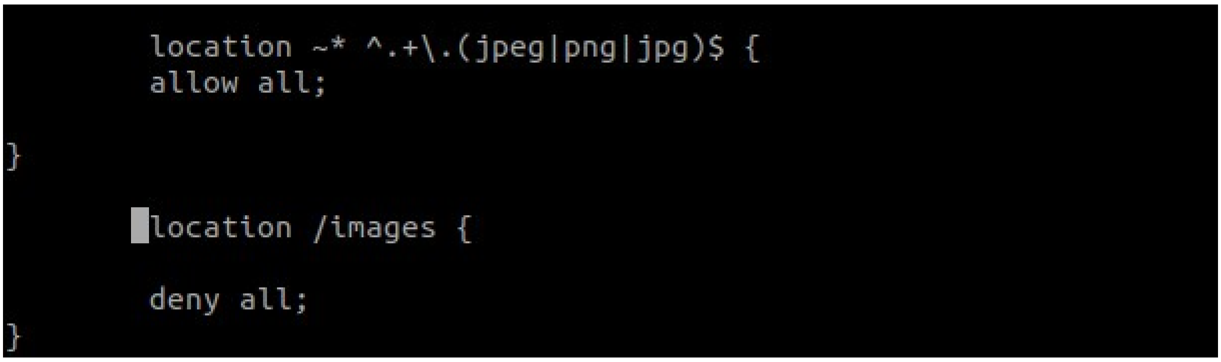


```
geo $geo {

        192.168.1.65  yes;
}
```

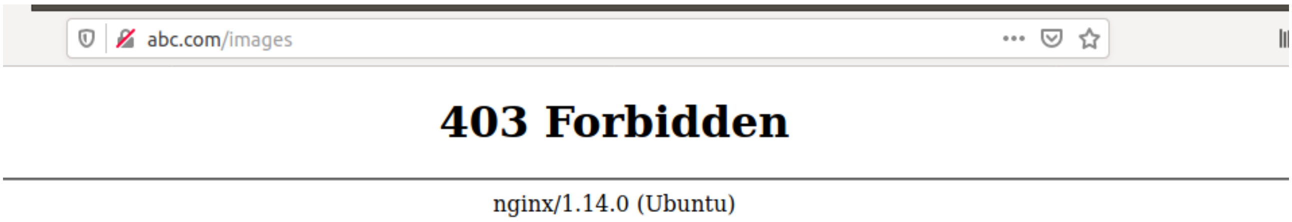


```
        error_page 404 /404.html;
        location = /404.html {


                if ($geo = yes) {
                return 404 $scheme://$host/4041.html;
        #       rewrite ^/404.html$ http://abc.com/4041.html perman
ent;

                }
```

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

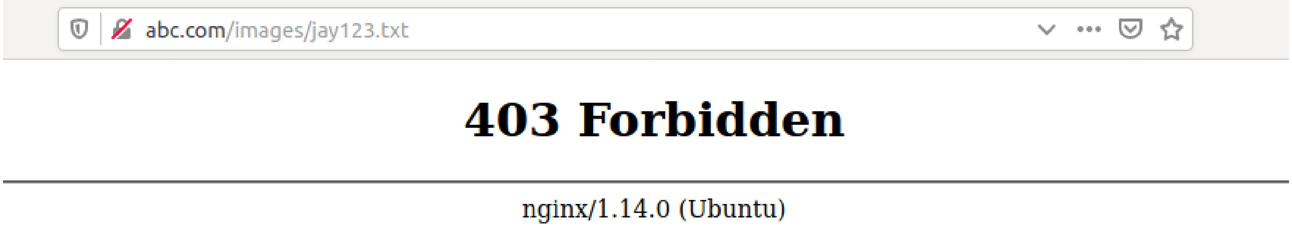


```
        location ~* ^.+\.(jpeg|png|jpg)$ {
        allow all;

}


        location /images {

        deny all;
}
```



403 Forbidden

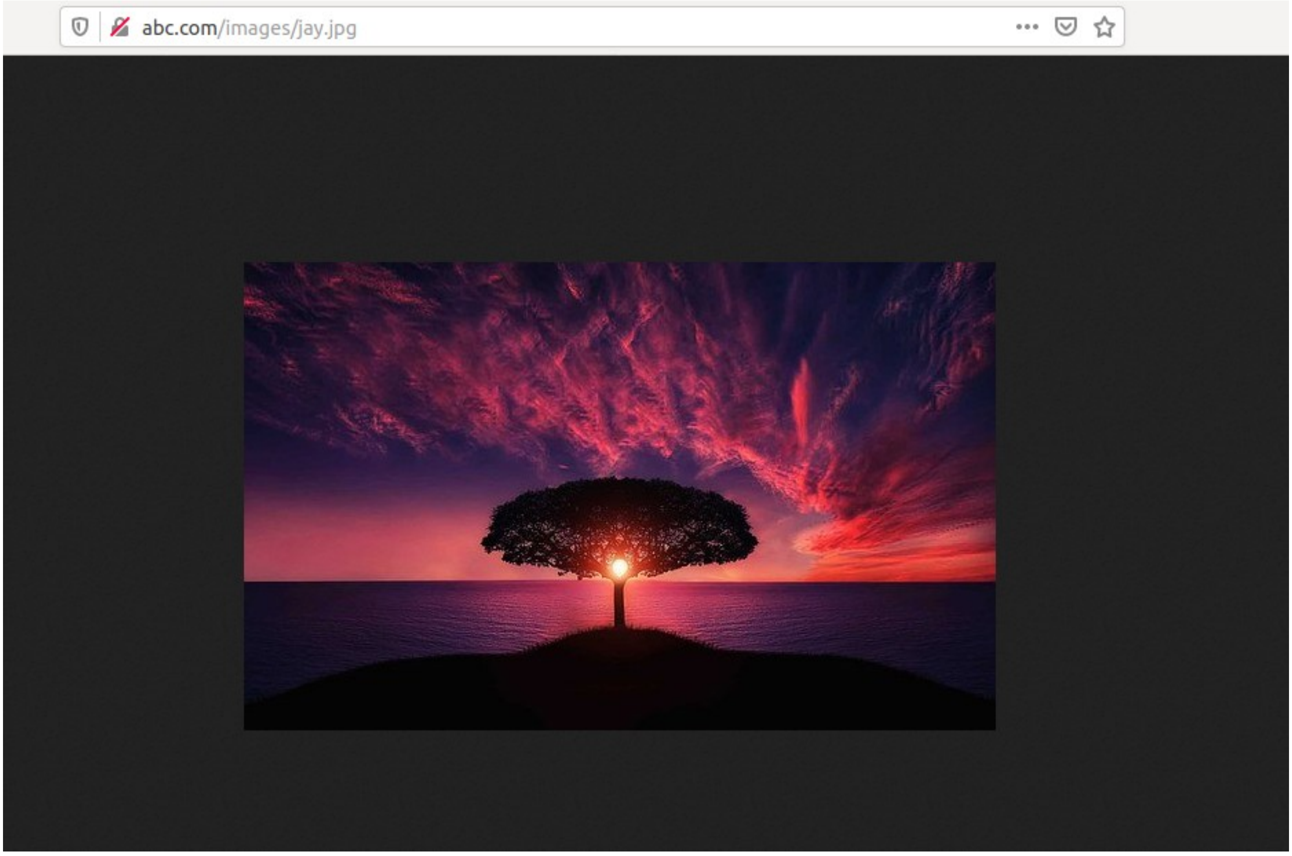
nginx/1.14.0 (Ubuntu)



abc.com/images/jay123.txt

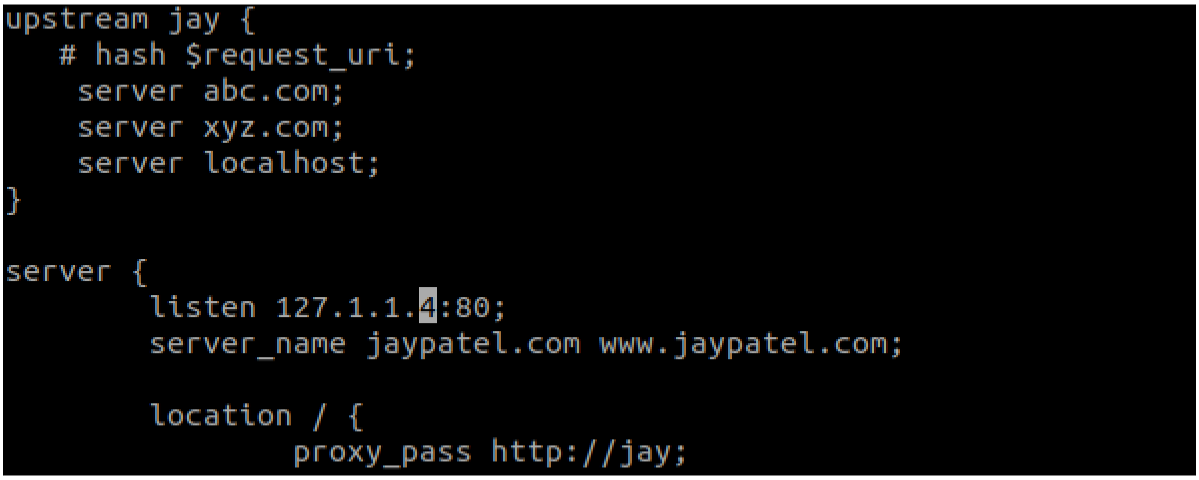
403 Forbidden

nginx/1.14.0 (Ubuntu)



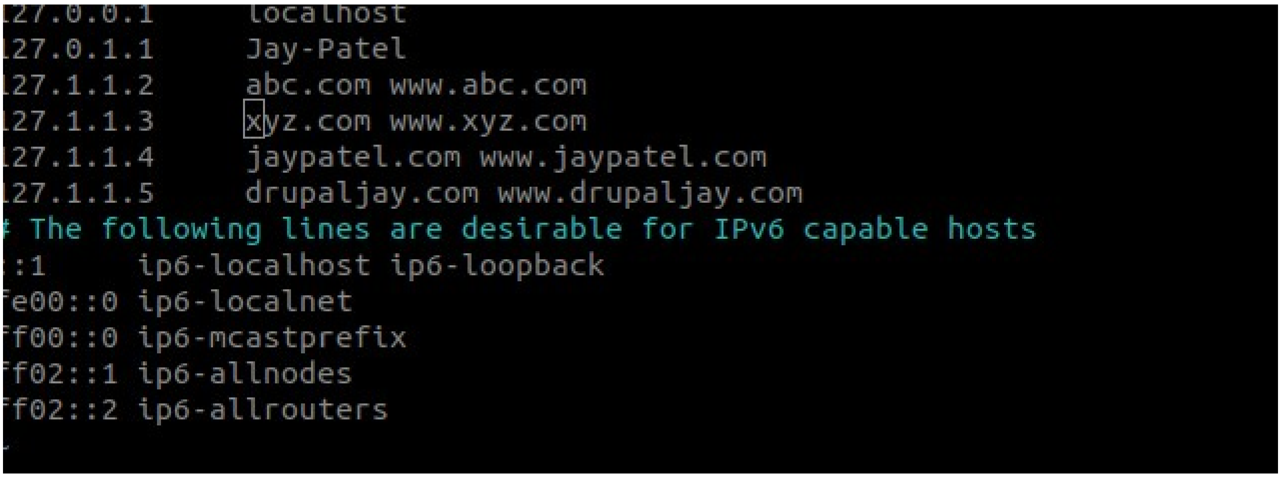
Q7.Create a load balancer with 3 backends. Explain different types of load balancing methods.

Load balancer config



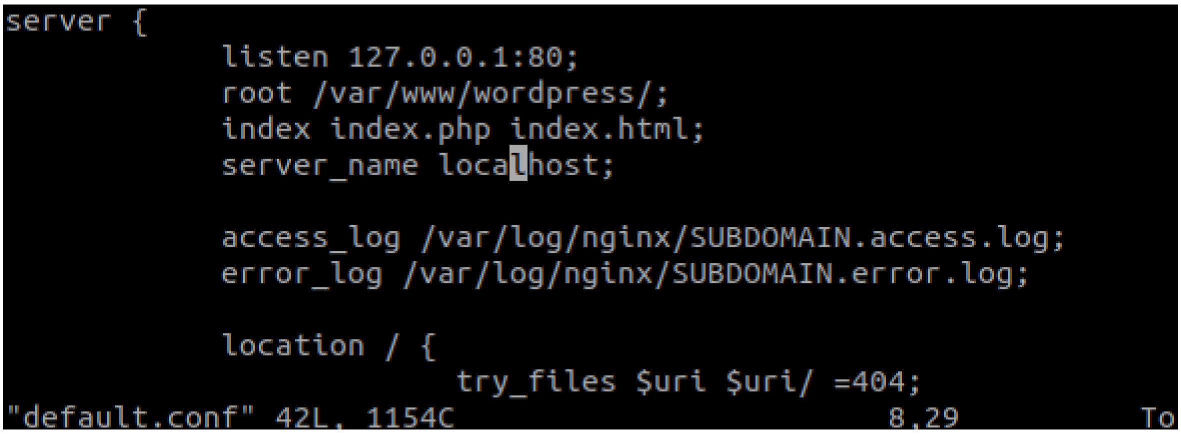
```
upstream jay {
   # hash $request_uri;
    server abc.com;
    server xyz.com;
    server localhost;
}

server {
        listen 127.1.1.4:80;
        server_name jaypatel.com www.jaypatel.com;

        location / {
                proxy_pass http://jay;
```

/etc/hosts file



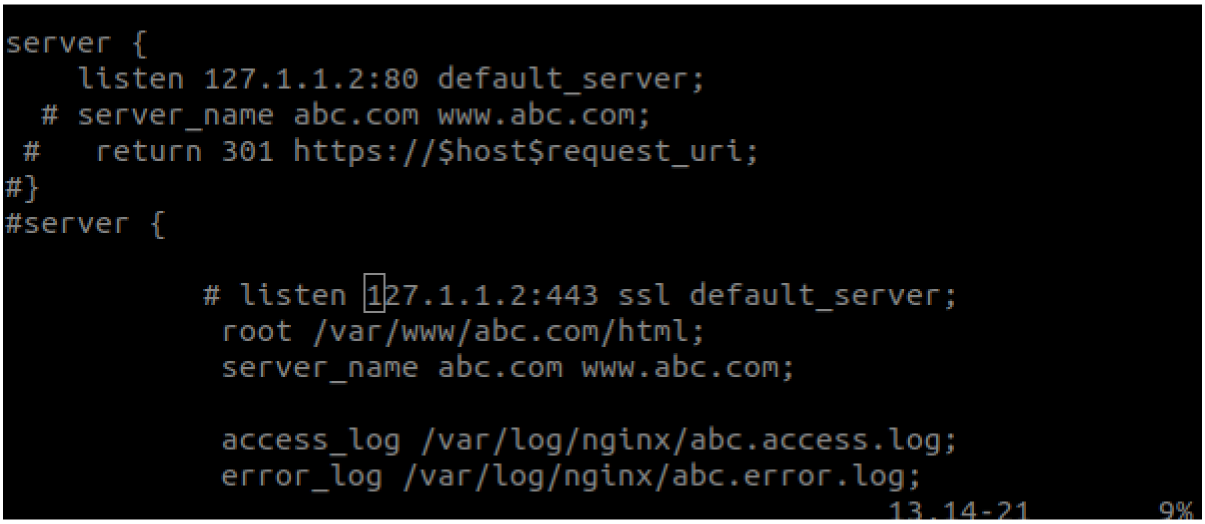
```
127.0.0.1       localhost
127.0.1.1       Jay-Patel
127.1.1.2       abc.com www.abc.com
127.1.1.3       xyz.com www.xyz.com
127.1.1.4       jaypatel.com www.jaypatel.com
127.1.1.5       drupaljay.com www.drupaljay.com
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Load balancer 1



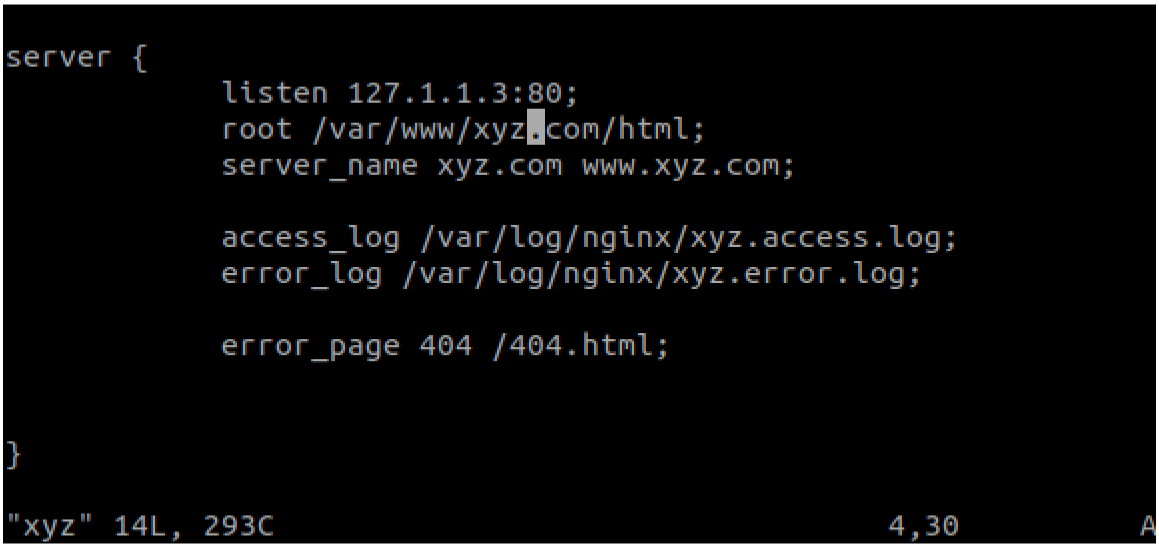
```
server {
            listen 127.0.0.1:80;
            root /var/www/wordpress/;
            index index.php index.html;
            server_name localhost;

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

            location / {
                         try_files $uri $uri/ =404;
"default.conf" 42L, 1154C                                8,29          To
```

Load balancer 2



```
server {
    listen 127.1.1.2:80 default_server;
  # server_name abc.com www.abc.com;
 #   return 301 https://$host$request_uri;
#}
#server {

           # listen 127.1.1.2:443 ssl default_server;
            root /var/www/abc.com/html;
            server_name abc.com www.abc.com;

            access_log /var/log/nginx/abc.access.log;
            error_log /var/log/nginx/abc.error.log;
                                                       13,14-21      9%
```

Load balancer 3

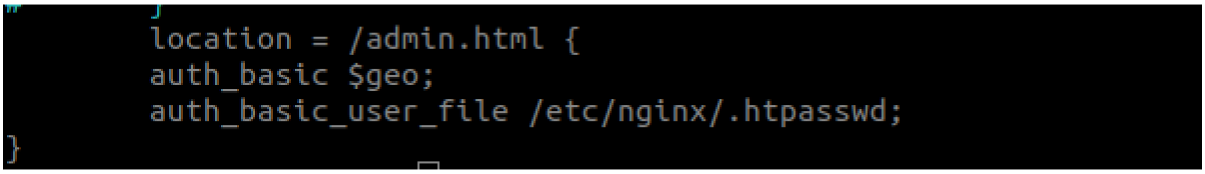


```
server {
            listen 127.1.1.3:80;
            root /var/www/xyz.com/html;
            server_name xyz.com www.xyz.com;

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

            error_page 404 /404.html;


}
"xyz" 14L, 293C                                             4,30          A
```

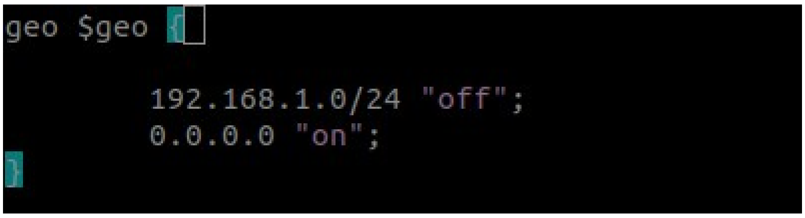
types of load balancing

1. round-robin” approach
2.least connections
3.hash – group of client to map with single server
4. ip_hash – particular client to map with single server
5. least time – nginx plus feature

Q8.Setup Basic Auth (Popup asking for username and password) in a particular location block. (The Basic Auth should not be asked for TTN IP)



```
        location = /admin.html {
        auth_basic $geo;
        auth_basic_user_file /etc/nginx/.htpasswd;
}
```



```
geo $geo {

        192.168.1.0/24 "off";
        0.0.0.0 "on";
}
```