Nginx-WebServer

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

Ans 1:-

- Avoid the expense of installing another web server. A reverse proxy server increases the capacity of existing servers.
- Reduce operating expense by increasing bandwidth.
- Provide another layer of protection by hiding the internal IP address.
- Provide a single point of control over who can access HTTP servers, and which servers can be accessed.

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
- Siteslocal.d
- Sites-avialable
- 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)

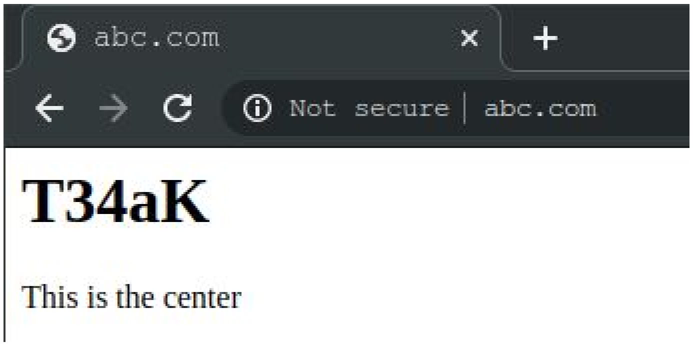
Log format :

```
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

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

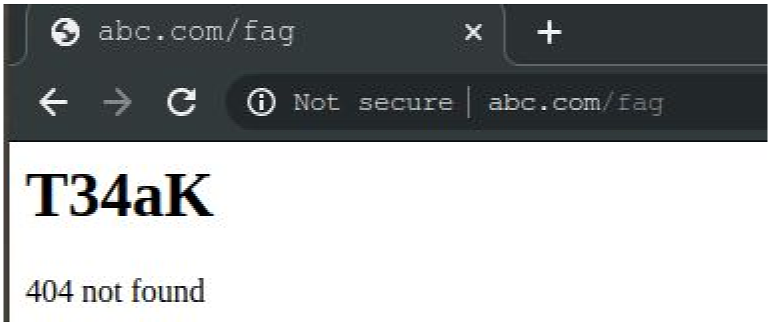
Index page



T34aK

This is the center

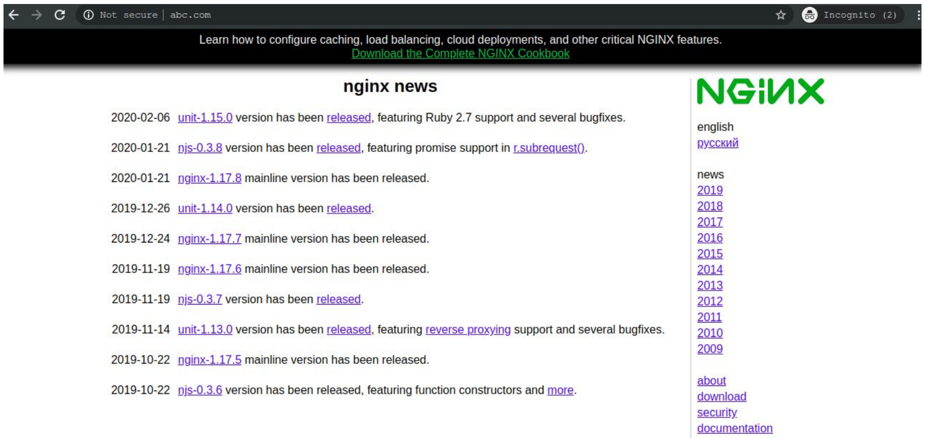
Fail-safe page



T34aK

404 not found

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



Learn how to configure caching, load balancing, cloud deployments, and other critical NGINX features.
Download the Complete NGINX Cookbook

nginx news

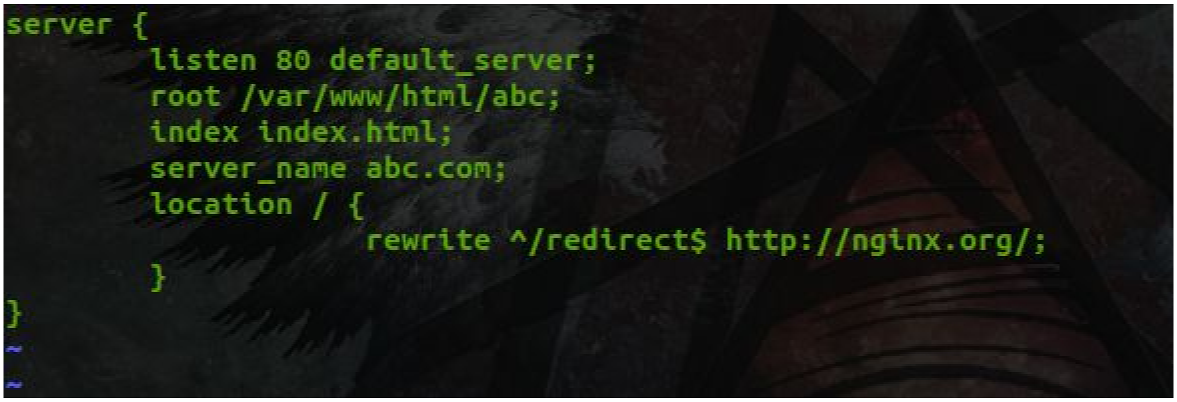
2020-02-06	unit-1.15.0 version has been released, featuring Ruby 2.7 support and several bugfixes.
2020-01-21	njs-0.3.8 version has been released, featuring promise support in r.subrequest().
2020-01-21	nginx-1.17.8 mainline version has been released.
2019-12-26	unit-1.14.0 version has been released.
2019-12-24	nginx-1.17.7 mainline version has been released.
2019-11-19	nginx-1.17.6 mainline version has been released.
2019-11-19	njs-0.3.7 version has been released.
2019-11-14	unit-1.13.0 version has been released, featuring reverse proxying support and several bugfixes.
2019-10-22	nginx-1.17.5 mainline version has been released.
2019-10-22	njs-0.3.6 version has been released, featuring function constructors and more.

english
русский

news
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009

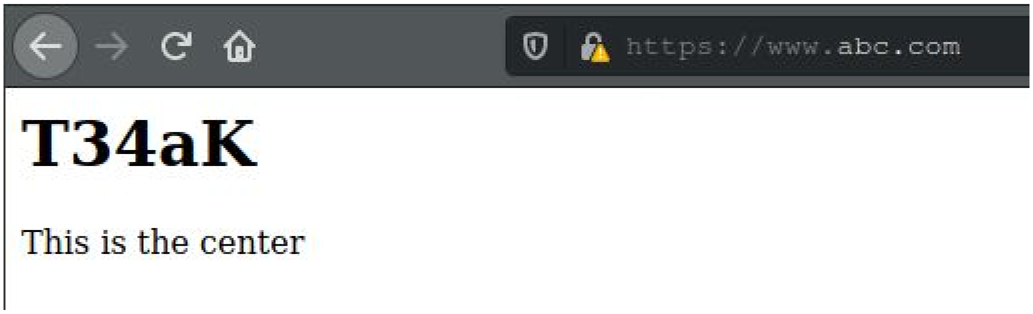
about
download
security
documentation

c:-redirect to above URI on /redirect/



```
server {
        listen 80 default_server;
        root /var/www/html/abc;
        index index.html;
        server_name abc.com;
        location / {
                rewrite ^/redirect$ http://nginx.org/;
        }
}
~
~
```

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

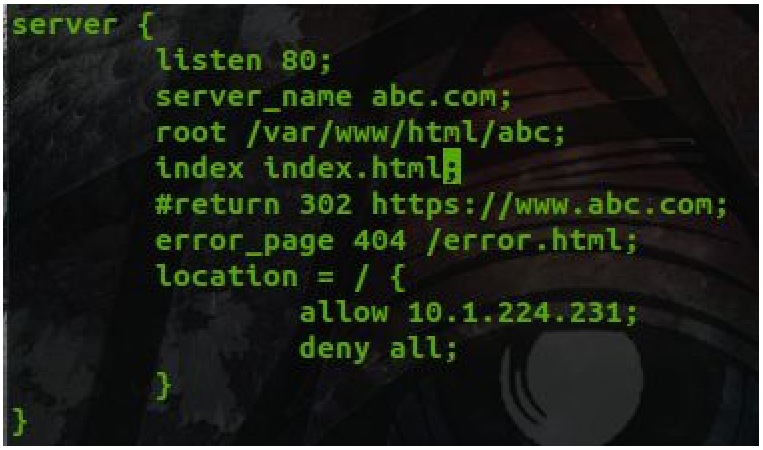


https://www.abc.com

T34aK

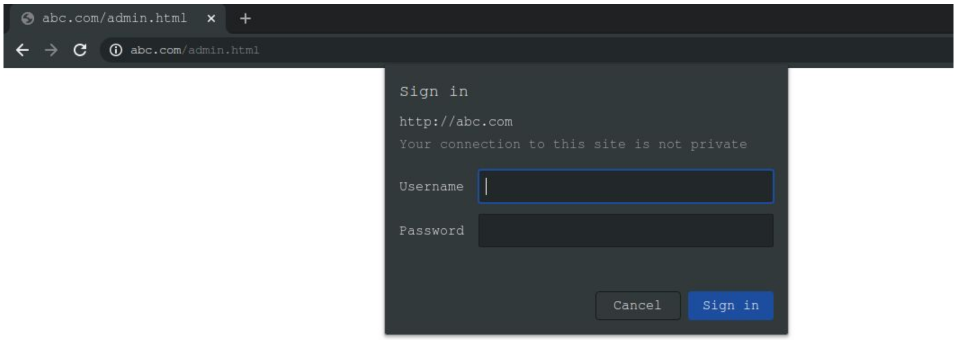
This is the center

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.



```
server {
        listen 80;
        server_name abc.com;
        root /var/www/html/abc;
        index index.html;
        #return 302 https://www.abc.com;
        error_page 404 /error.html;
        location = / {
                allow 10.1.224.231;
                deny all;
        }
}
```

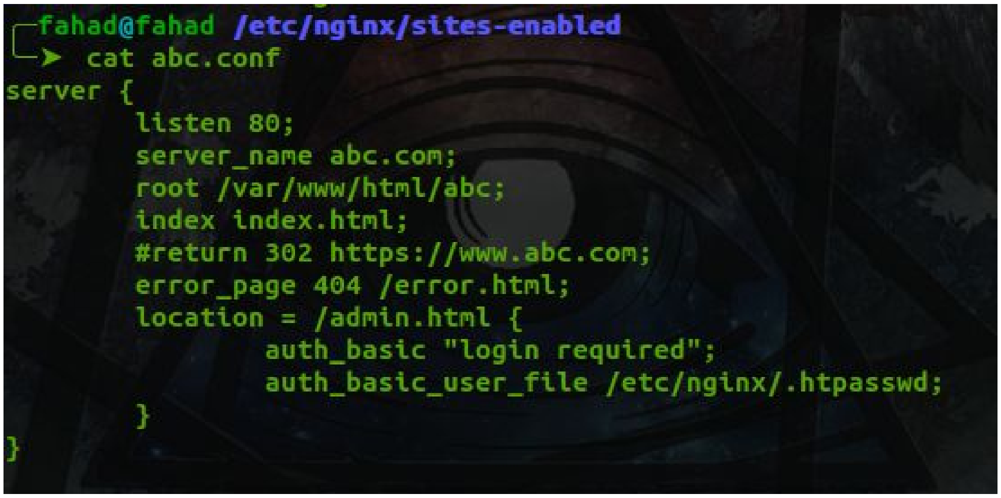
f:-Setup Basic Auth (Popup asking for username and password) in a particular location block. (The Basic Auth should not be asked for TTN IP)



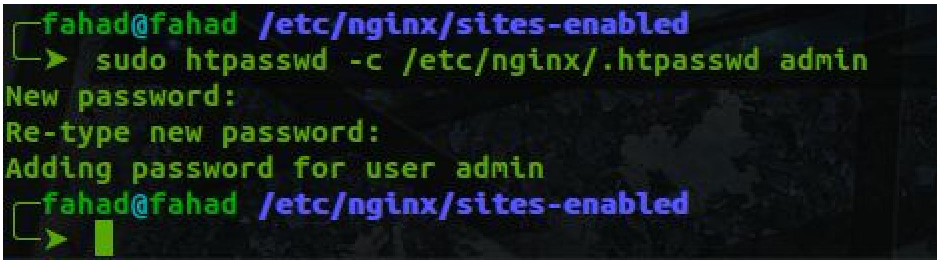
Sign in

http://abc.com

Your connection to this site is not private



```
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ cat abc.conf
server {
        listen 80;
        server_name abc.com;
        root /var/www/html/abc;
        index index.html;
        #return 302 https://www.abc.com;
        error_page 404 /error.html;
        location = /admin.html {
                auth_basic "login required";
                auth_basic_user_file /etc/nginx/.htpasswd;
        }
}
```



```
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ sudo htpasswd -c /etc/nginx/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ 
```

g:-Create a load balancer with 5 backends. Explain different types of load balancing methods.

i) Round Robin

ii) Least connection:

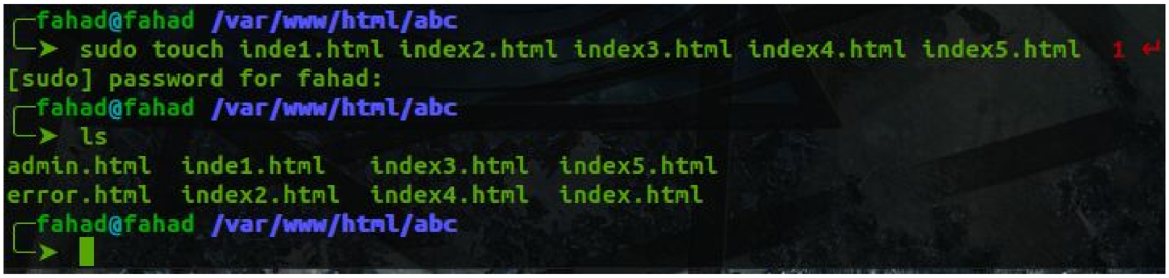
```
        Upstream backend{

        Least_conn;

        Server backend1.example.com;

        Server backend2.example.com;

        }
```

iii) IP hash:

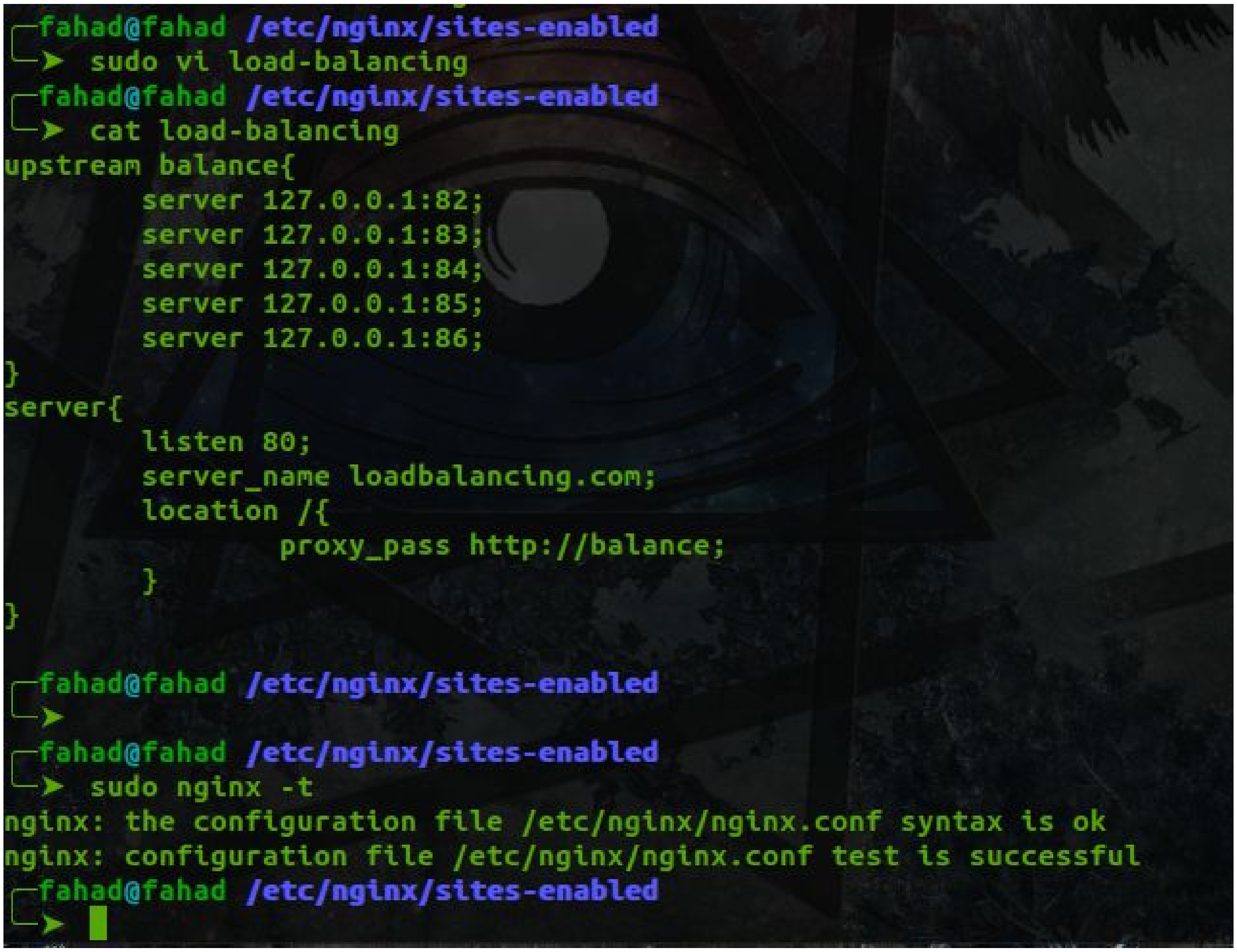
```
        Upstream backend{

        ip_hash;

        Server backend1.example.com;

        Server backend2.example.com;

        }
```

iv) Least time:

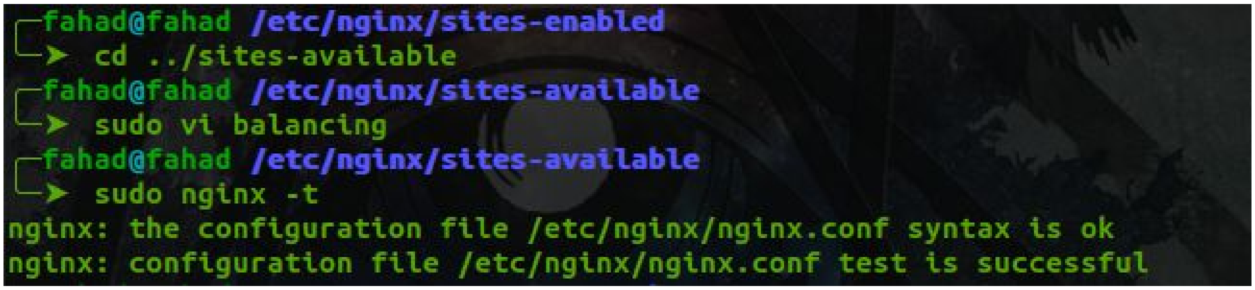
```
        Upstream backend{

        Least_time header;

        Server backend1.example.com;

        Server backend2.example.com;

        }
```



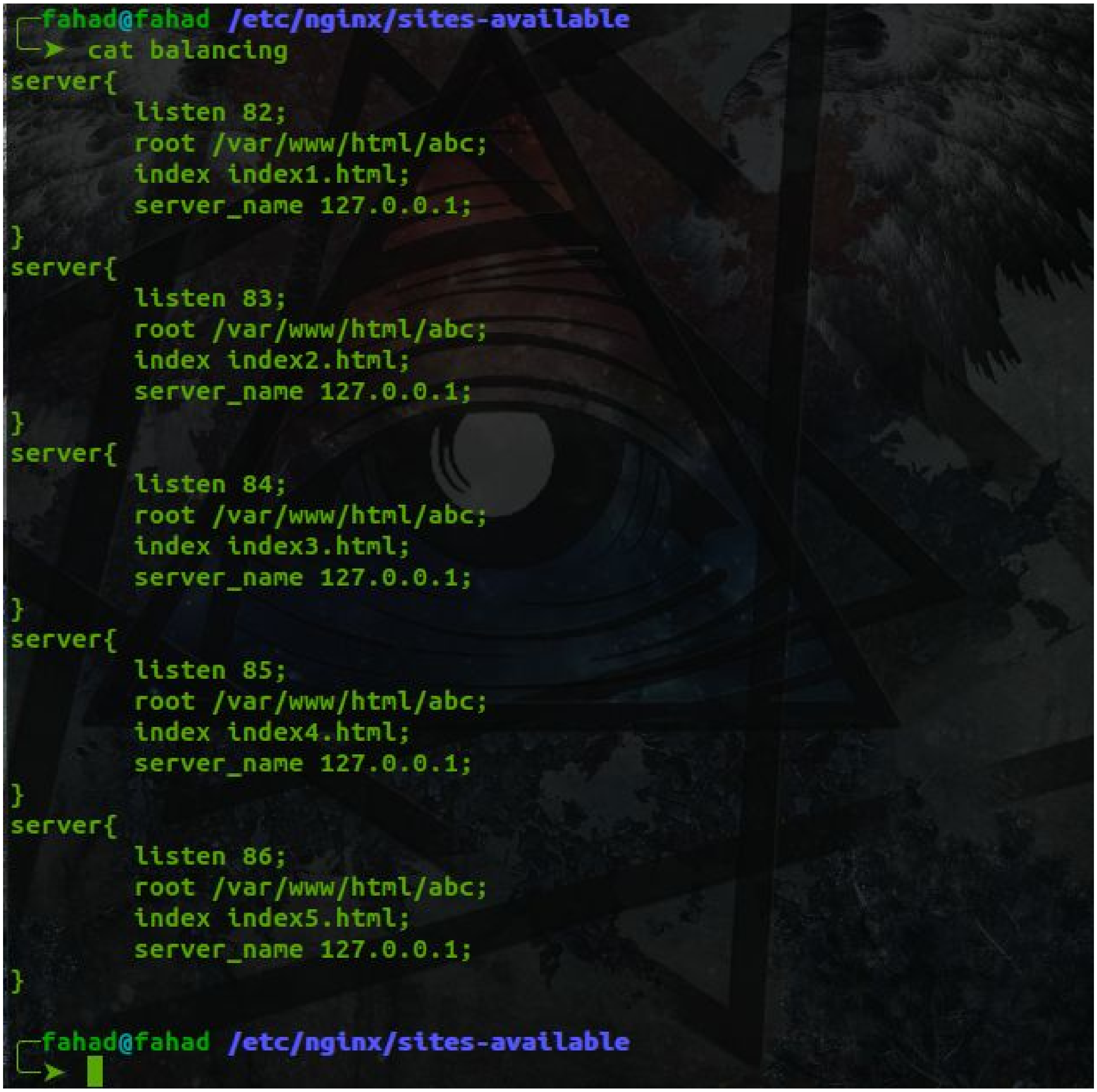
```
┌fahad@fahad /var/www/html/abc
└➤ sudo touch inde1.html index2.html index3.html index4.html index5.html   1 ↵
[sudo] password for fahad:
┌fahad@fahad /var/www/html/abc
└➤ ls
admin.html  inde1.html   index3.html  index5.html
error.html  index2.html  index4.html  index.html
┌fahad@fahad /var/www/html/abc
└➤ 
```



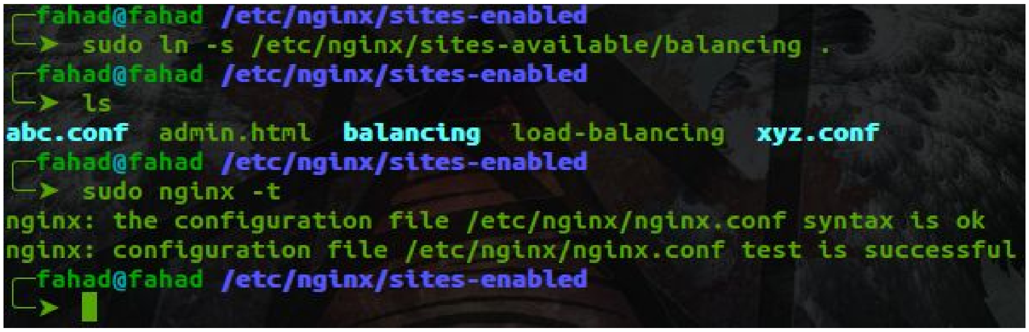
```
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ sudo vi load-balancing
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ cat load-balancing
upstream balance{
        server 127.0.0.1:82;
        server 127.0.0.1:83;
        server 127.0.0.1:84;
        server 127.0.0.1:85;
        server 127.0.0.1:86;
}
server{
        listen 80;
        server_name loadbalancing.com;
        location /{
                proxy_pass http://balance;
        }
}

┌─fahad@fahad /etc/nginx/sites-enabled
└─➤
┌─fahad@fahad /etc/nginx/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
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ █
```



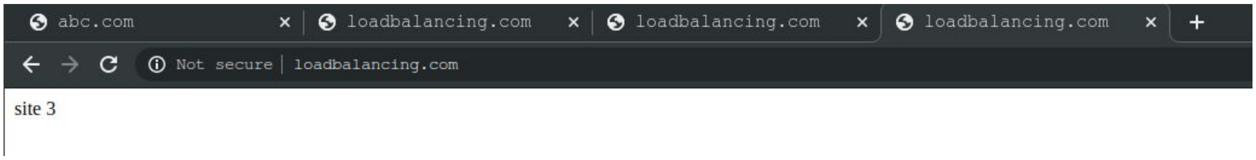
```
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ cd ../sites-available
┌─fahad@fahad /etc/nginx/sites-available
└─➤ sudo vi balancing
┌─fahad@fahad /etc/nginx/sites-available
└─➤ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```



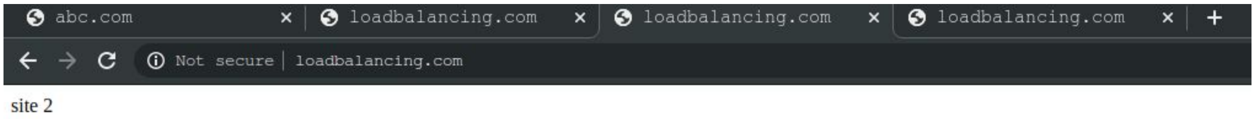
```
┌─fahad@fahad /etc/nginx/sites-available
└─➤ cat balancing
server{
        listen 82;
        root /var/www/html/abc;
        index index1.html;
        server_name 127.0.0.1;
}
server{
        listen 83;
        root /var/www/html/abc;
        index index2.html;
        server_name 127.0.0.1;
}
server{
        listen 84;
        root /var/www/html/abc;
        index index3.html;
        server_name 127.0.0.1;
}
server{
        listen 85;
        root /var/www/html/abc;
        index index4.html;
        server_name 127.0.0.1;
}
server{
        listen 86;
        root /var/www/html/abc;
        index index5.html;
        server_name 127.0.0.1;
}

┌─fahad@fahad /etc/nginx/sites-available
└─➤ 
```



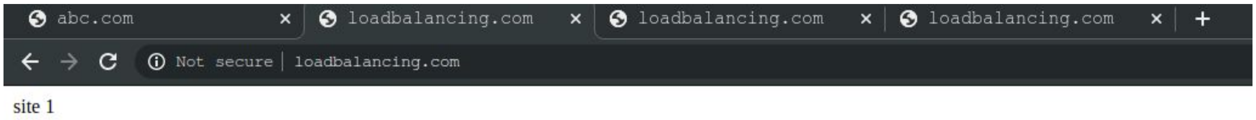
```
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ sudo ln -s /etc/nginx/sites-available/balancing .
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ ls
abc.conf  admin.html  balancing  load-balancing  xyz.conf
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
┌─fahad@fahad /etc/nginx/sites-enabled
└─➤ 
```



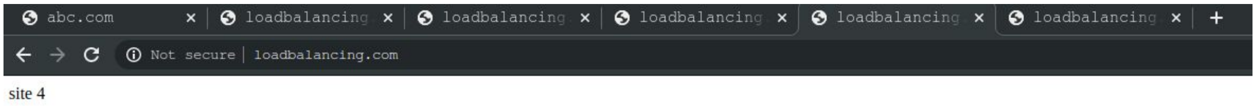
site 3



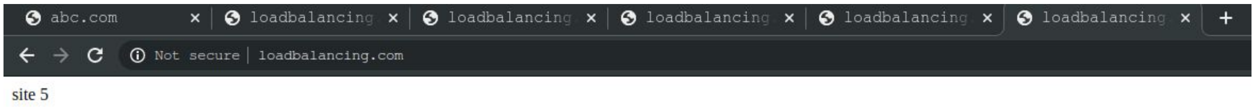
site 2



site 1



site 4



site 5