Task: Track Request & Response Time Using Nginx

To do the above task we have to install Nginx and also we have to deploy a website on Nginx Server.

Track Request and Response Time generally refers to monitoring and measuring the time it takes for a system or application to send a request and receive a response.

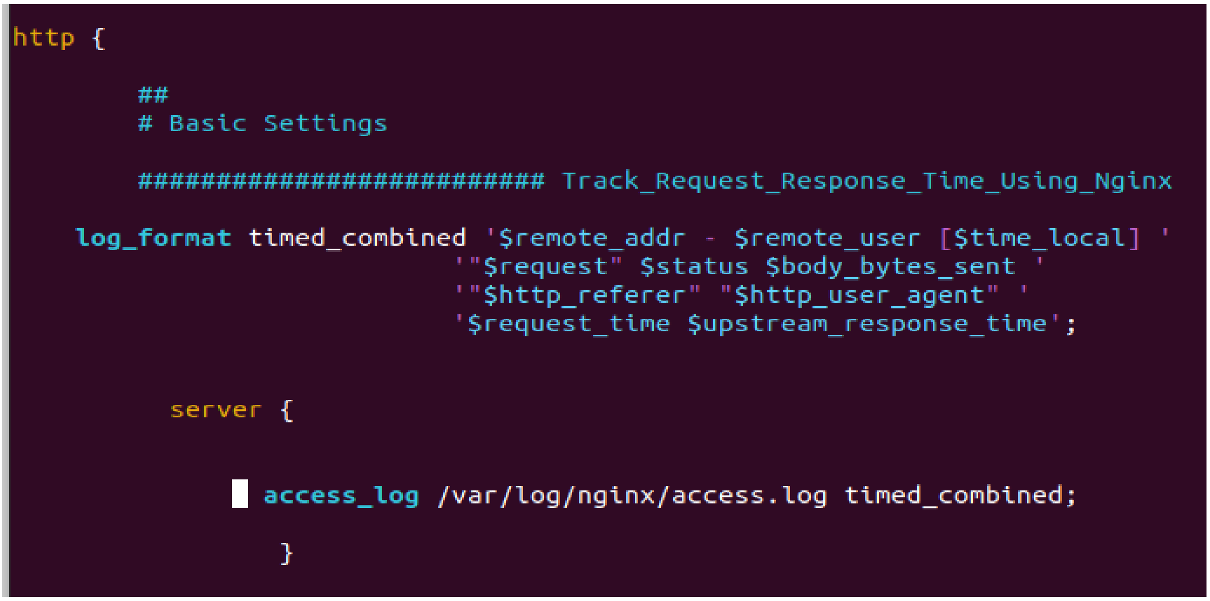
Monitoring track request and response time is important for several reasons :

1) **Performance Optimization :** Monitoring response times helps identify potential bottlenecks or slow components in a system.
2) **Troubleshooting :** When users report slow or unresponsive behavior, tracking request and response times can help diagnose the cause of the issue and implement appropriate fixes.
3) **Troubleshooting :** When users report slow or unresponsive behavior, tracking request and response times can help diagnose the cause of the issue and implement appropriate fixes.

Following are the steps to configure settings to Track Request and Response Time -

Steps :

1) Open the “nginx.conf” file located in “/etc/nginx/” and add the following code into it.



```
http {

        ##
        # Basic Settings

        ########################## Track_Request_Response_Time_Using_Nginx

    log_format timed_combined '$remote_addr - $remote_user [$time_local] '
                            '"$request" $status $body_bytes_sent '
                            '"$http_referer" "$http_user_agent" '
                            '$request_time $upstream_response_time';


          server {

                access_log /var/log/nginx/access.log timed_combined;

                 }
```

“**log_format timed_combined**” : This line defines a custom log format named “**timed_combined**”.
It specifies the format of the log entries that will be recorded when clients access the server.
The format includes various placeholders (variables) that will be replaced with actual values during logging.

$remote_addr : IP address of the client.
$remote_user : Remote user (if provided during authentication).
$time_local : Local time of the server when the request was received.
$request : The full HTTP request line.
$status : HTTP response status code (e.g., 200, 404, etc.).
$body_bytes_sent : Size of the response body sent to the client.
$http_referer : Referring URL from which the client was directed.
$http_user_agent : User agent (web browser or client) string.
$request_time : Time taken to process the request (in seconds with millisecond precision).
$upstream_response_time : Time taken by the upstream server to process the request.

2) Under server block, We have to tell the access_log to use the above log format.

   ```
   access_log /var/log/nginx/access.log timed_combined;
   ```

3) After configuring the “nginx.conf" file, restart the nginx service.

4) To see the output hit the url/IP in different browser with different device
   and to see the output run the following command.

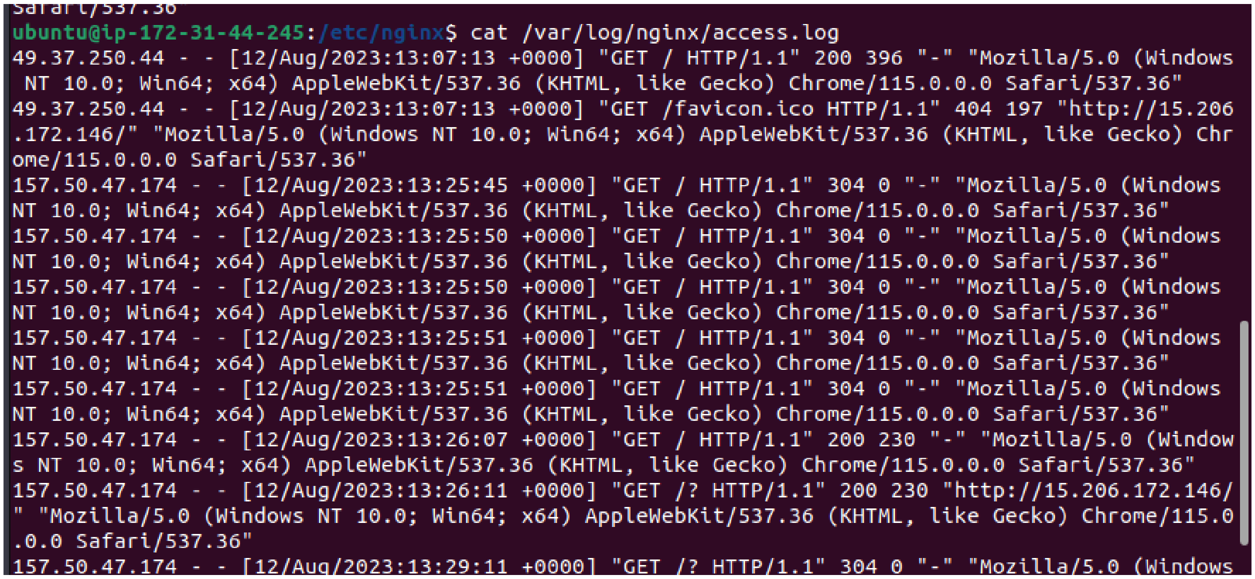
   cat /var/log/nginx/access.log

   OR

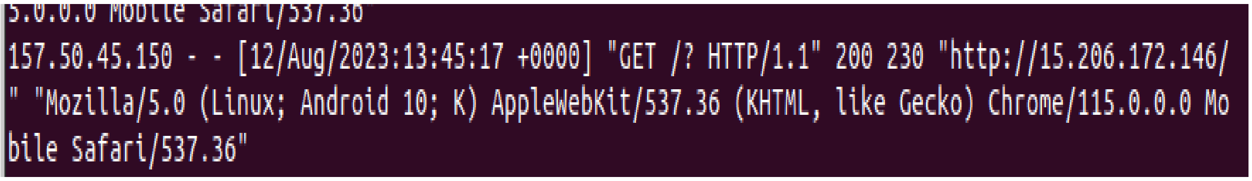
   less /var/log/nginx/access.log

   OR

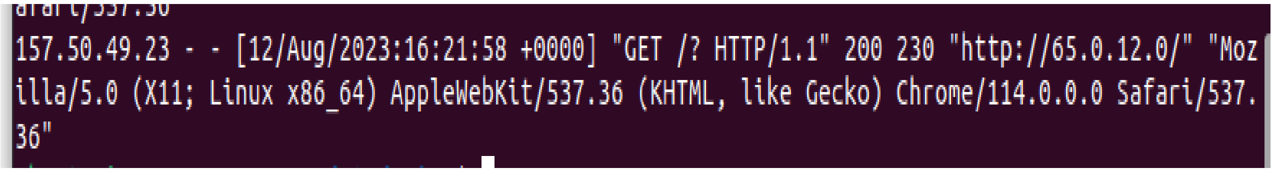
   tail /var/log/nginx/access.log



```
Safari/537.36
ubuntu@ip-172-31-44-245:/etc/nginx$ cat /var/log/nginx/access.log
49.37.250.44 - - [12/Aug/2023:13:07:13 +0000] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (Windows
 NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
49.37.250.44 - - [12/Aug/2023:13:07:13 +0000] "GET /favicon.ico HTTP/1.1" 404 197 "http://15.206
.172.146/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
ome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:25:45 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:25:50 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:25:50 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:25:51 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:25:51 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:26:07 +0000] "GET / HTTP/1.1" 200 230 "-" "Mozilla/5.0 (Window
s NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:26:11 +0000] "GET /? HTTP/1.1" 200 230 "http://15.206.172.146/
" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0
.0.0 Safari/537.36"
157.50.47.174 - - [12/Aug/2023:13:29:11 +0000] "GET /? HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows
```



```
5.0.0.0 Mobile Safari/537.36
157.50.45.150 - - [12/Aug/2023:13:45:17 +0000] "GET /? HTTP/1.1" 200 230 "http://15.206.172.146/
" "Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Mo
bile Safari/537.36"
```



```
afari/537.36
157.50.49.23 - - [12/Aug/2023:16:21:58 +0000] "GET /? HTTP/1.1" 200 230 "http://65.0.12.0/" "Moz
illa/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.
36"
```

Conclusion :

As there are multiple APM (Application performance Monitoring) tools available to investigate the performance of the applications, it’s quite time consuming and expensive to set up.
This is the one best simple and easy solution to immediately analyze and rectify the issues within the applications.