Installation and Administration of Nginx Web Server

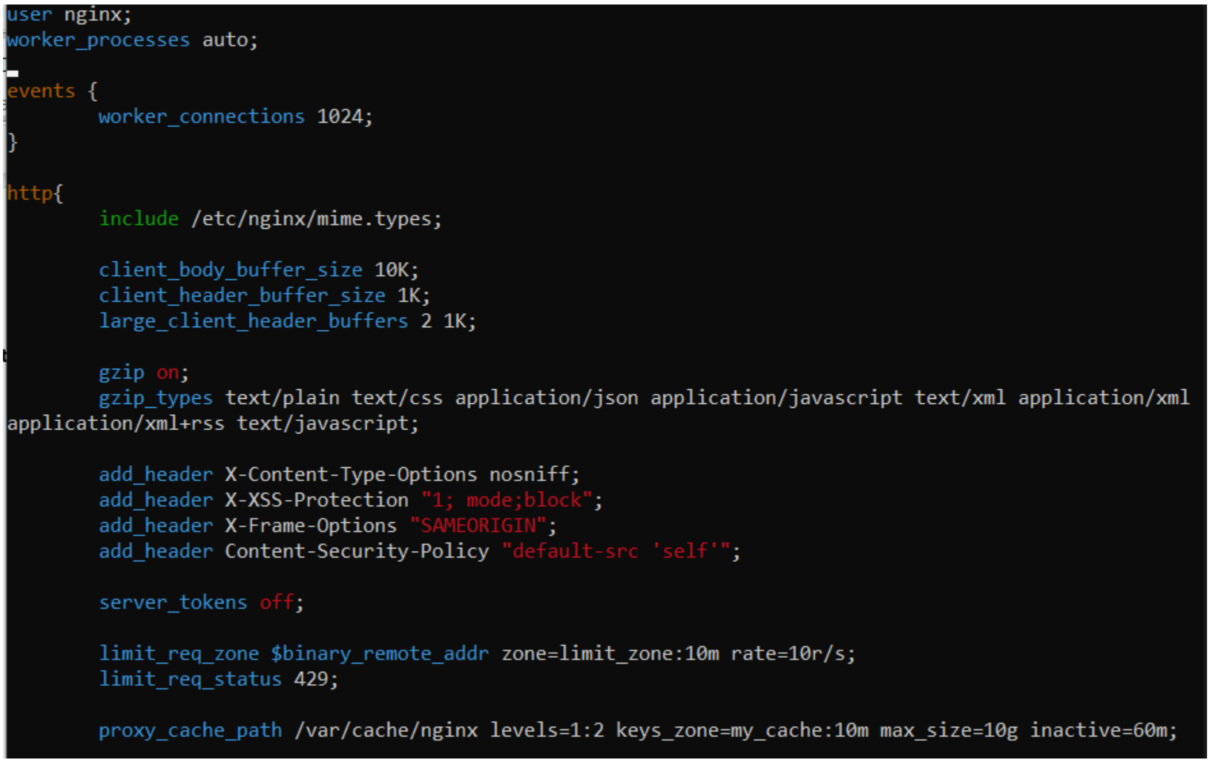
Nginx installation and management will be performed within the scope of the project. For this, nginx, php-fpm applications were installed on the server. Then the nginx.conf file was edited. The nginx.conf file can be found in the "/etc/nginx/nginx.conf" directory.

1. **“user nginx; and worker_processes auto;”:** These lines specify which user account and how many processes Nginx will execute. In this example, Nginx will run under the user account nginx and the number of processes will be set automatically.
2. **events:** This block configures the Nginx event module. The value specified by worker_connections specifies the number of connections to be opened at the same time.
3. **http:** The main HTTP configuration block starts here. All HTTP settings are defined in this block.
4. **“include /etc/nginx/mime.types;”:** This line specifies the path to the MIME types file. MIME types help to correctly identify file types.
5. **“client_body_buffer_size 10K; and client_header_buffer_size 1K;”:** Used by Nginx to limit the size of the body and header data of HTTP requests sent by the client. This header "client_boddy_buffer_size" specifies the maximum size of the body of sent requests. The reason for specifying 10K is to ensure that the sent request cannot exceed 10 kilobytes. This prevents oversized requests from overloading the server's resources. "client_header_buffer_size" specifies the maximum size of HTTP headers. If 1K is specified, HTTP headers cannot exceed 1 kilobyte. If the headers sent by the client exceed this size, Nginx will close the connection to the client before receiving any more header data.
6. **"large_client_header_buffers 2 1K;":** In some cases, clients may need to send special or very large headers, in which case the default memory buffer sizes may be insufficient. This directive specifies the size of the memory buffers that Nginx can use to receive large headers sent by the client in such cases. In the configuration shown in the figure, Nginx is configured to use 2 1 kilobyte buffers, totalling 2 kilobytes of memory.
7. **“’gzip on;’ and ‘gzip_types ...’;”:** Enables the gzip compression feature and specifies which MIME types are used for compression.
8. **add_header ...;:** Adds additional headers to HTTP responses. Here the headers "X-Content-Type-Options", "X-XSS-Protection", "X-Frame-Options" and "Content-Security-Policy" are added. These headers provide basic protection against XSS and clickjacking vulnerabilities.
9. **server_tokens off;:** Hides the server header so that it does not give any information about the Nginx version.
10. **“limit_req_zone ...; ve limit_req_status 429;”:** The limit_req_zone and limit_req_status directives are used to configure Nginx's IP-based query restrictions and limitations. These configurations are used to limit requests from specific IP addresses or to keep query speed under control. limit_req_zone: This directive defines a zone to be used for query limitations. limit_req_zone directive has three parameters:
    a. $binary_remote_addr: IP addresses to which the limitation will be applied.
    b. zone: The zone to be used for the limitation.
    c. rate: Specifies the maximum query rate rate: Specifies the maximum query rate allowed in a given period of time.

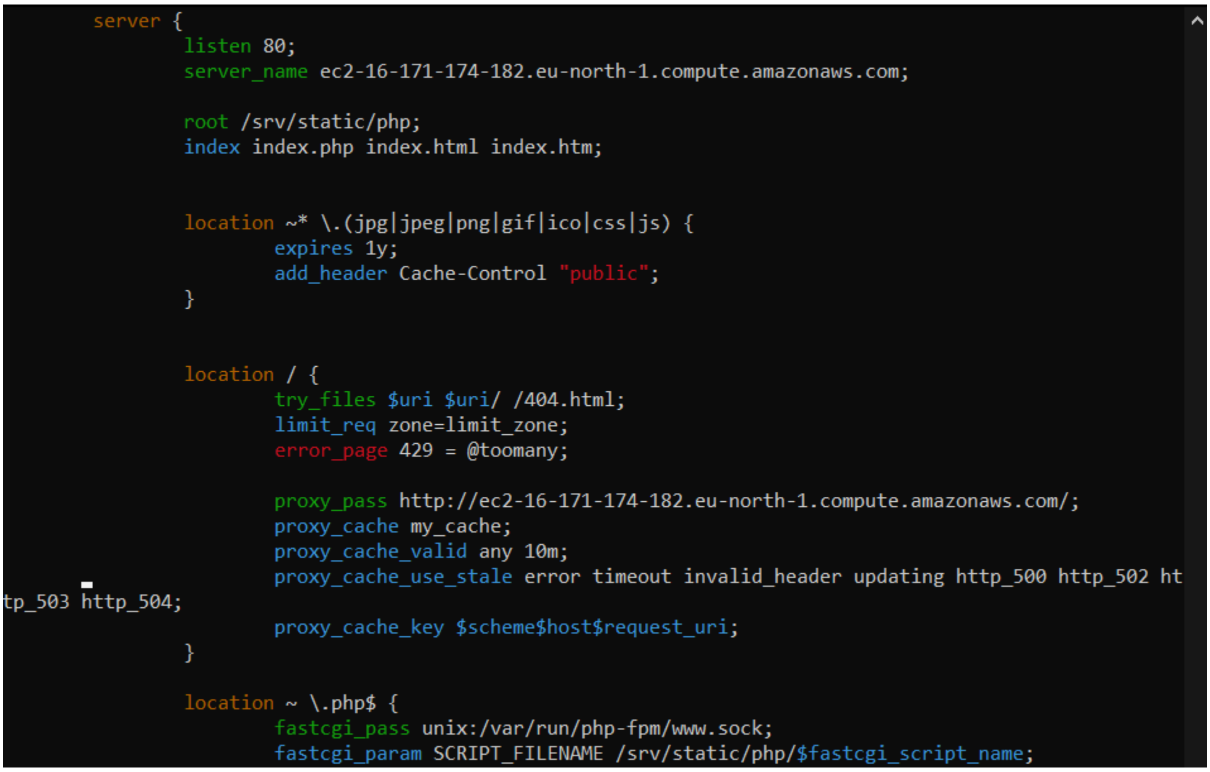
    For example, in a configuration such as limit_req_zone $binary_remote_addr zone=limit_zone:10m rate=10r/s;, queries from the same IP address are limited to 10 queries/second for 10 minutes. This means that queries from the same IP address are allowed one request every 10 seconds.

limit_req_status: This directive specifies which HTTP status code the server should send when the query limit is exceeded. For example, if limit_req_status is specified as 429;, the server sends the status code 429 Too Many Requests when the query limit is exceeded.

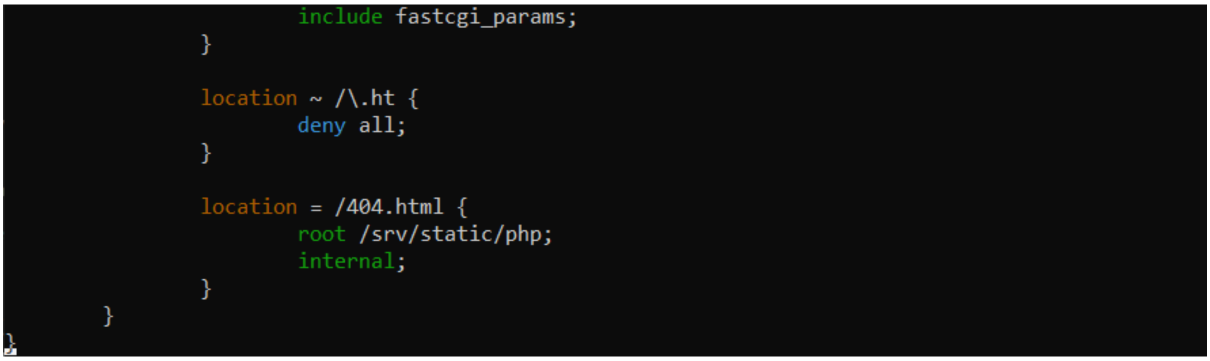
11. **“proxy_cache_path ...;”:** Defines the cache structure for the HTTP proxy.
12. **server:** This is the main configuration block where Nginx runs as a web server. The properties of the web server are defined here.
13. **“listen 80;”:** Specifies that the web server listens on port 80.
14. **“server_name ec2-16-171-174-182.eu-north-1.compute.amazonaws.com;”:** Specifies the domain name of the server.
15. **“root /srv/static/php;”:** The main root directory. Indicates that the web server will serve files from here.
16. **“index index.php index.html index.htm;”:** If there are specified files in the directory, indicates their priority.
17. **location:** These blocks define operations based on specific URL criteria.
18. **location ~* \.(jpg|jpeg|png|gif|ico|css|js) { ... }:** This block allows caching files with specified extensions and adding custom headers based on their extensions.
19. **location / { ... }:** Redirects to the home page and ensures that the specified URLs redirect to /404.html if there are no static files.
20. **location ~ \.php$ { ... }:** Redirects PHP files and makes PHP-FPM related settings.
21. **location ~ /\.ht { ... }:** Prevents direct access to .htaccess files.
22. **location = /404.html { ... }:** Redirects to /404.html in case of a 404 error.



```
user nginx;
worker_processes auto;

events {
        worker_connections 1024;
}

http{
        include /etc/nginx/mime.types;

        client_body_buffer_size 10K;
        client_header_buffer_size 1K;
        large_client_header_buffers 2 1K;

        gzip on;
        gzip_types text/plain text/css application/json application/javascript text/xml application/xml
application/xml+rss text/javascript;

        add_header X-Content-Type-Options nosniff;
        add_header X-XSS-Protection "1; mode;block";
        add_header X-Frame-Options "SAMEORIGIN";
        add_header Content-Security-Policy "default-src 'self'";

        server_tokens off;

        limit_req_zone $binary_remote_addr zone=limit_zone:10m rate=10r/s;
        limit_req_status 429;

        proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_cache:10m max_size=10g inactive=60m;
```



```
        server {
                listen 80;
                server_name ec2-16-171-174-182.eu-north-1.compute.amazonaws.com;

                root /srv/static/php;
                index index.php index.html index.htm;


                location ~* \.(jpg|jpeg|png|gif|ico|css|js) {
                        expires 1y;
                        add_header Cache-Control "public";
                }


                location / {
                        try_files $uri $uri/ /404.html;
                        limit_req zone=limit_zone;
                        error_page 429 = @toomany;

                        proxy_pass http://ec2-16-171-174-182.eu-north-1.compute.amazonaws.com/;
                        proxy_cache my_cache;
                        proxy_cache_valid any 10m;
                        proxy_cache_use_stale error timeout invalid_header updating http_500 http_502 ht
tp_503 http_504;
                        proxy_cache_key $scheme$host$request_uri;
                }

                location ~ \.php$ {
                        fastcgi_pass unix:/var/run/php-fpm/www.sock;
                        fastcgi_param SCRIPT_FILENAME /srv/static/php/$fastcgi_script_name;
```



```
                        include fastcgi_params;
                }

                location ~ /\.ht {
                        deny all;
                }

                location = /404.html {
                        root /srv/static/php;
                        internal;
                }
        }
}
```