WORDPRESS USING PROXY

APURVA ANIL JAGDALE

PROBLEM STATEMENT

The problem statement requires deploying a sample WordPress website, protecting it with a Nginx reverse proxy, and allowing admin login from a specific IP address only. Additionally, the candidate must enable log rotation, write a script to analyze Nginx logs, and provide a report. They must also automate the deployment using either cloud infrastructure automation technology or containers.

Requirements:

1)Installing Wordpress on Ubuntu OS.

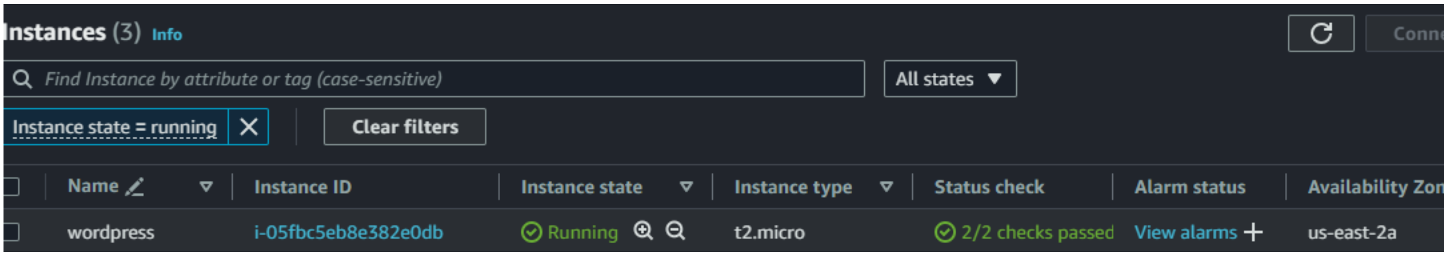
2)Reverse Proxy Setup using Nginx.

3)Enabling Log Rotation

4) Allowing Admin login from the specific IP only.

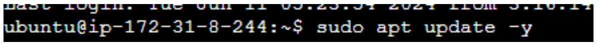
STEP I : INSTALLING WORDPRESS ON UBUNTU OS.

Step 1: Launch the EC2 Instance using Ubuntu.



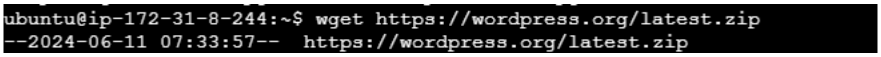
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
□	wordpress	i-05fbc5eb8e382e0db	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a

Step 2: Then update the instance using Below Command.



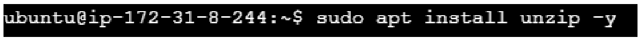
```
ubuntu@ip-172-31-8-244:~$ sudo apt update -y
```

Step 3 : Download the Wordpress using https://wordpress.org/latest.zip this link address.

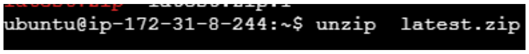


```
ubuntu@ip-172-31-8-244:~$ wget https://wordpress.org/latest.zip
--2024-06-11 07:33:57--  https://wordpress.org/latest.zip
```

Step 4 : then unzip it.

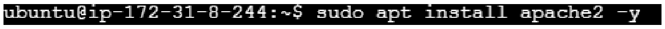


```
ubuntu@ip-172-31-8-244:~$ sudo apt install unzip -y
```

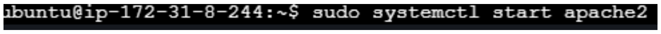


```
ubuntu@ip-172-31-8-244:~$ unzip  latest.zip
```

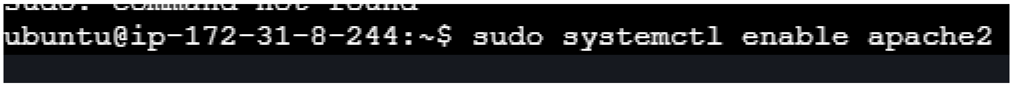
Step 5: then, download the apache server then start and enable it.



```
ubuntu@ip-172-31-8-244:~$ sudo apt install apache2 -y
```

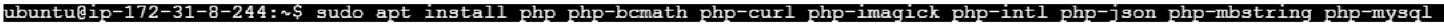


```
ubuntu@ip-172-31-8-244:~$ sudo systemctl start apache2
```



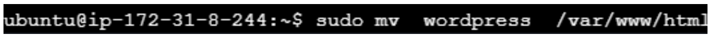
```
ubuntu@ip-172-31-8-244:~$ sudo systemctl enable apache2
```

Step 6 : Download the php with supported Packages.



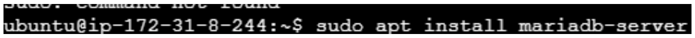
```
ubuntu@ip-172-31-8-244:~$ sudo apt install php php-bcmath php-curl php-imagick php-intl php-json php-mbstring php-mysql
```

Step 7 : Move the extracted wordpress directory to /var/www/html/

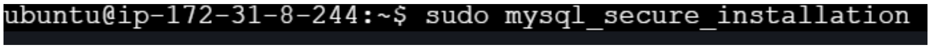


```
ubuntu@ip-172-31-8-244:~$ sudo mv  wordpress  /var/www/html
```

Step 8 : Then Install Mariadb Server.



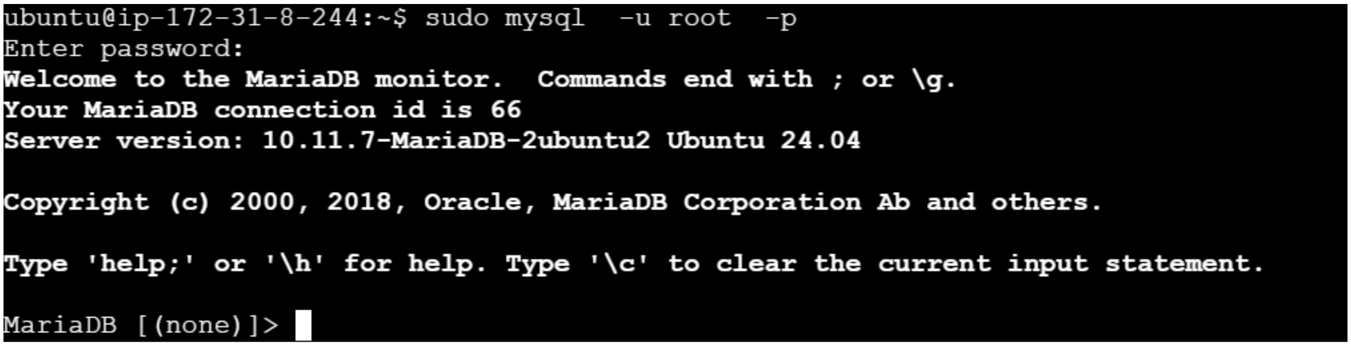
```
sudo: command not found
ubuntu@ip-172-31-8-244:~$ sudo apt install mariadb-server
```



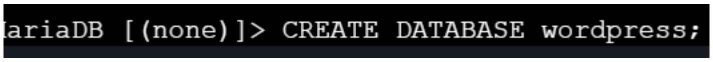
```
ubuntu@ip-172-31-8-244:~$ sudo mysql_secure_installation
```

Type y here for secure installation and create password for root user.

Step 9 : Create the Wordpress Database.

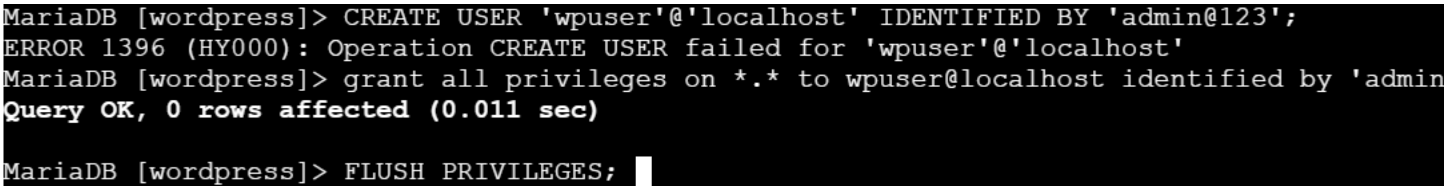


```
ubuntu@ip-172-31-8-244:~$ sudo mysql  -u root  -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 66
Server version: 10.11.7-MariaDB-2ubuntu2 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> 
```



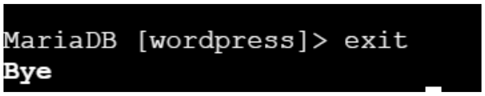
```
ariaDB [(none)]> CREATE DATABASE wordpress;
```

Step 10 : Create user for wordpress database and grant the all privillages.



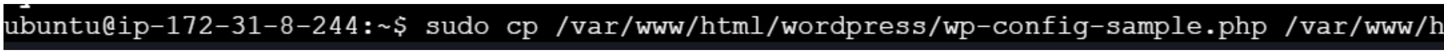
```
MariaDB [wordpress]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'admin@123';
ERROR 1396 (HY000): Operation CREATE USER failed for 'wpuser'@'localhost'
MariaDB [wordpress]> grant all privileges on *.* to wpuser@localhost identified by 'admin
Query OK, 0 rows affected (0.011 sec)

MariaDB [wordpress]> FLUSH PRIVILEGES; 
```

Step 11: then exit.



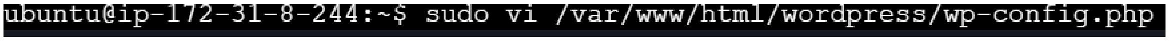
```
MariaDB [wordpress]> exit
Bye
```

Step 12: Configure wordpress using this command .



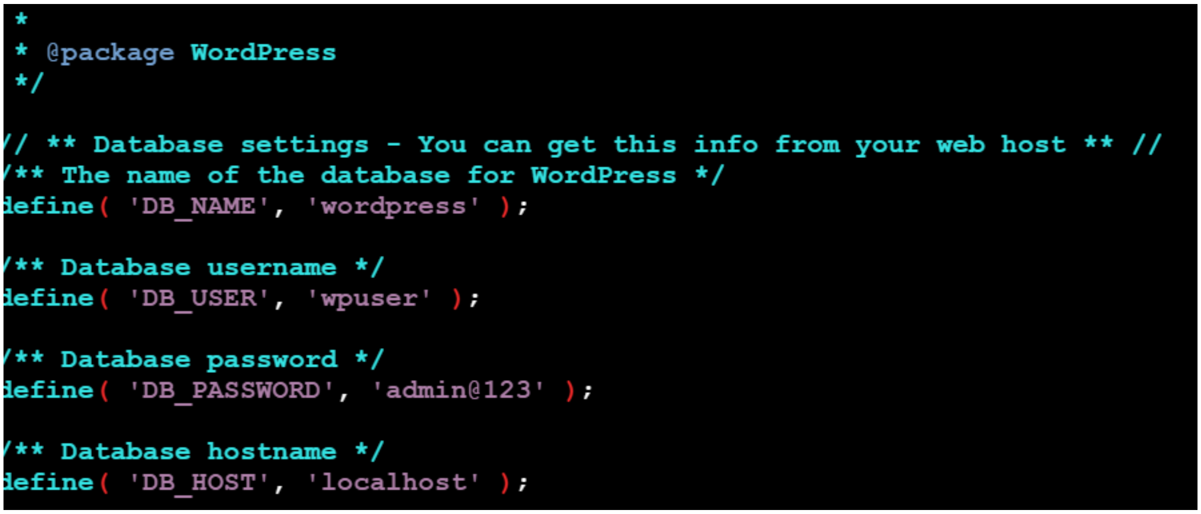
```
ubuntu@ip-172-31-8-244:~$ sudo cp /var/www/html/wordpress/wp-config-sample.php /var/www/h
```

Step 13: then open wordpress configure file using below command.



```
ubuntu@ip-172-31-8-244:~$ sudo vi /var/www/html/wordpress/wp-config.php
```

Step 14: then give the db name,username and password here.

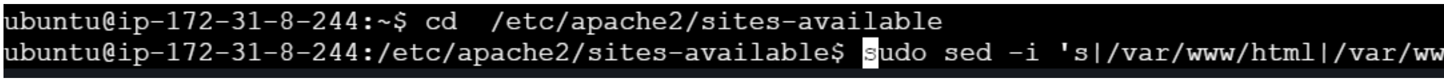


```
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'admin@123' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Step 15: then save and close it.

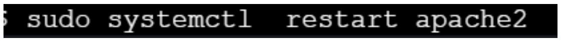
Step 16: after that using http://<public-ip>/wordpress/wp-admin/install.php then enter the site title,username and password for your website.

Step 17: you want to hit your website using only ip then give below commands



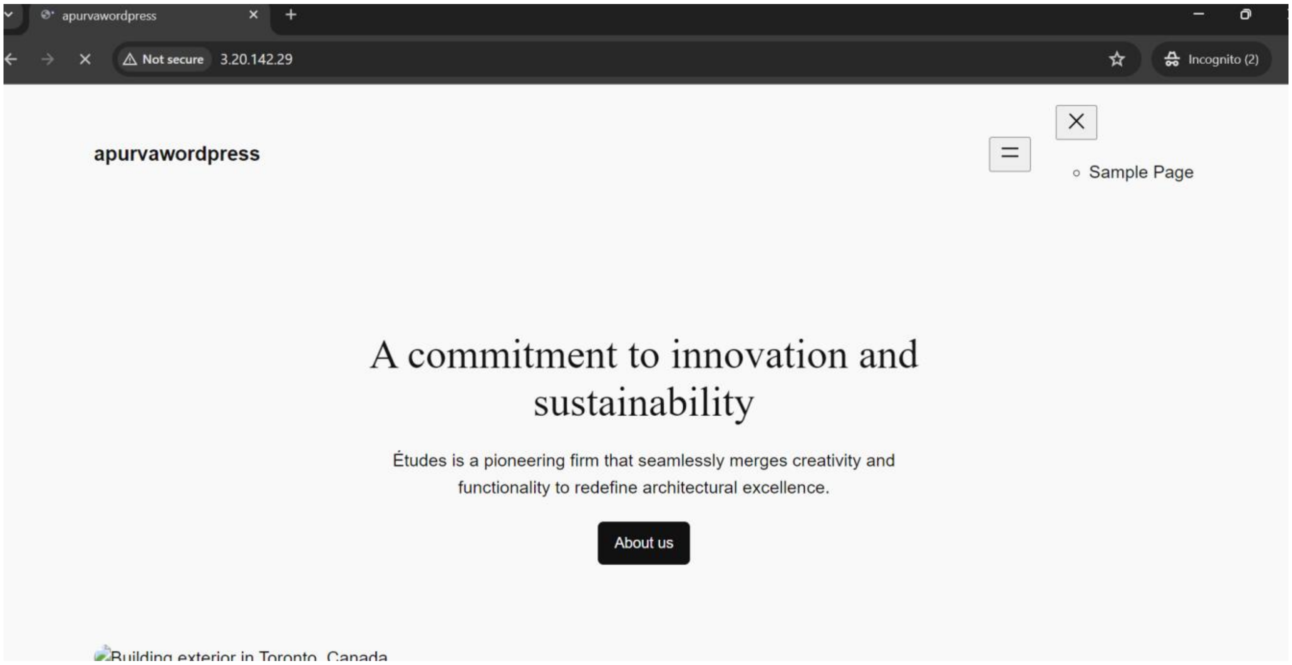
```
ubuntu@ip-172-31-8-244:~$ cd  /etc/apache2/sites-available
ubuntu@ip-172-31-8-244:/etc/apache2/sites-available$ sudo sed -i 's|/var/www/html|/var/ww
```

Step 18:then restart your apache tomcat



```
$ sudo systemctl  restart apache2
```

Step 19: now,the the public ip and see your wordpress site.



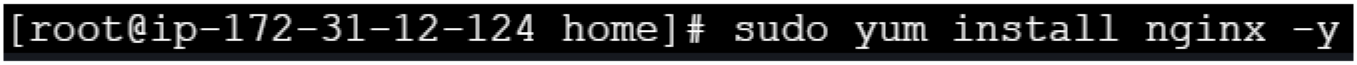
A commitment to innovation and sustainability

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

STEP II : REVERSE PROXY USING NGINX.

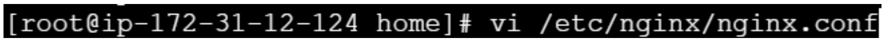
Step 1 : Launch ec2 instance using amazon linux ami.

Step 2: then install nginx in it using “sudo yum install nginx -y”



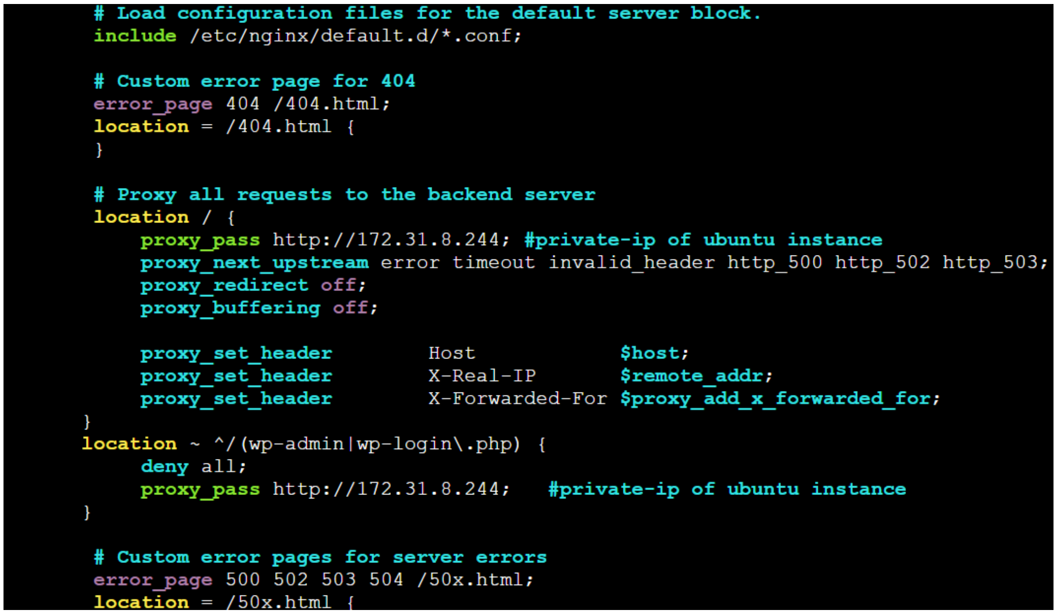
```
[root@ip-172-31-12-124 home]# sudo yum install nginx -y
```

Step 3 : then configure nginx.conf file using below command.



```
[root@ip-172-31-12-124 home]# vi /etc/nginx/nginx.conf
```

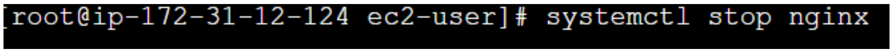
Step 4: Add the below content their.



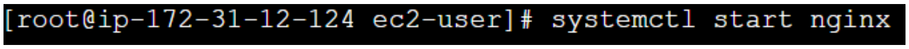
```
        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        # Custom error page for 404
        error_page 404 /404.html;
        location = /404.html {
        }

        # Proxy all requests to the backend server
        location / {
            proxy_pass http://172.31.8.244; #private-ip of ubuntu instance
            proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;
            proxy_redirect off;
            proxy_buffering off;

            proxy_set_header        Host            $host;
            proxy_set_header        X-Real-IP       $remote_addr;
            proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
       }
       location ~ ^/(wp-admin|wp-login\.php) {
            deny all;
            proxy_pass http://172.31.8.244;   #private-ip of ubuntu instance
       }

        # Custom error pages for server errors
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
```

Step 5 : then save it and close it.

Step 6 : then stop the nginx server and start and enable it.

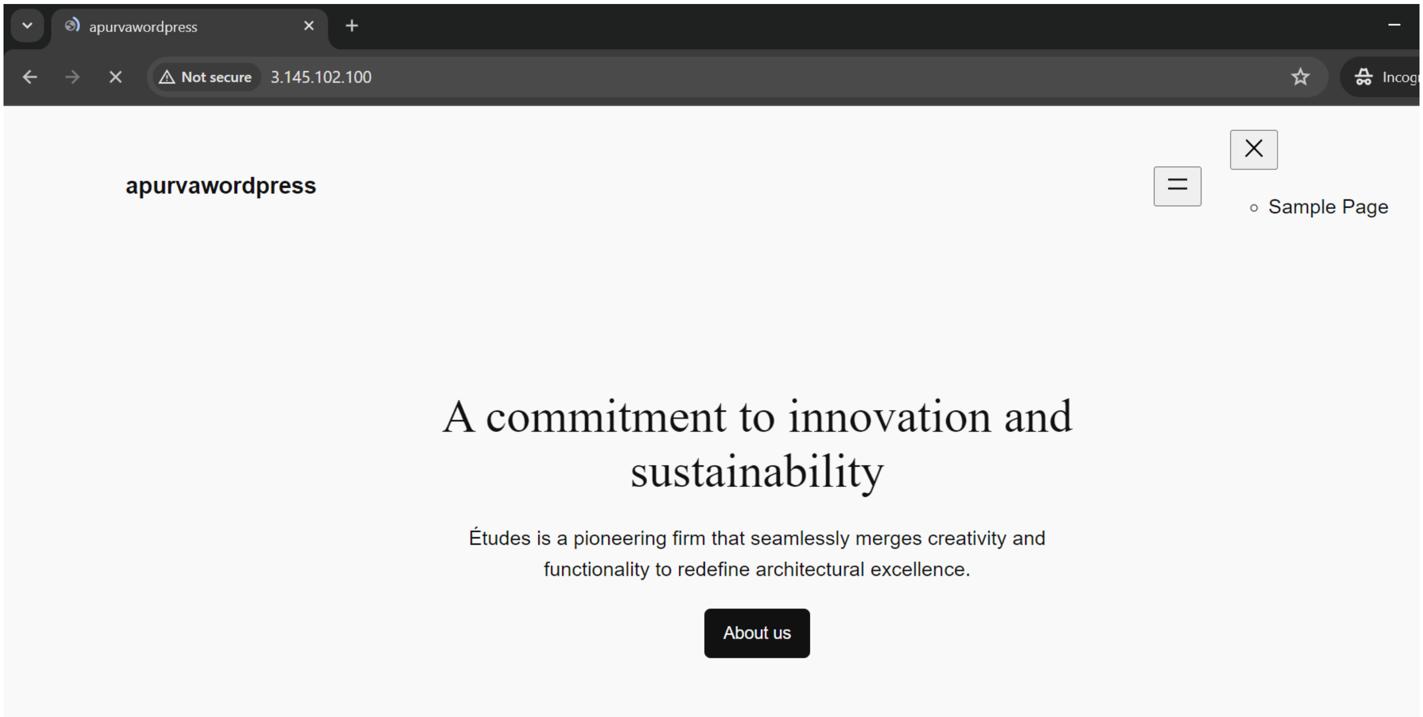


```
[root@ip-172-31-12-124 ec2-user]# systemctl stop nginx
```



```
[root@ip-172-31-12-124 ec2-user]# systemctl start nginx
```

Step 7 : hit the public ip of server and see here it is access successfully using nginx proxy.



apurvawordpress

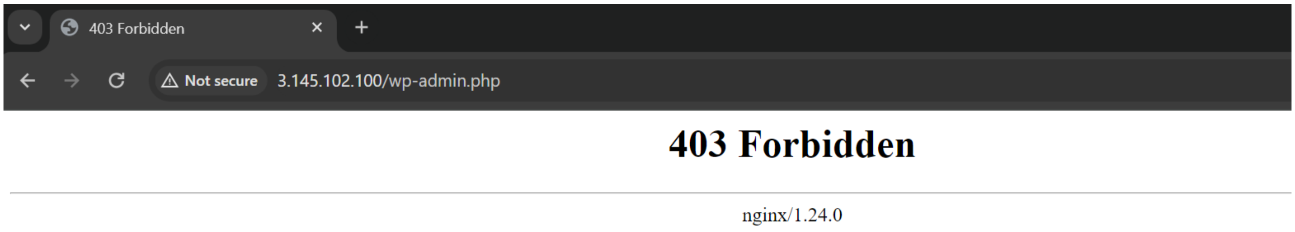
Sample Page

A commitment to innovation and sustainability

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

About us

Step 8 : using proxy normal user cannot access the admin page.

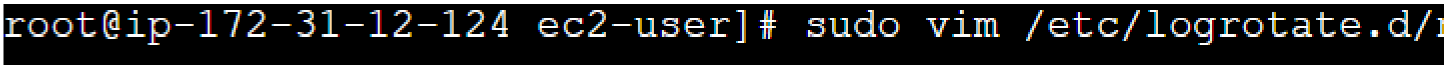


403 Forbidden

nginx/1.24.0

STEP III : ENABLING LOG ROTATION.

Step 1: Create log rotation configuration file using below command in nginx proxy instance.

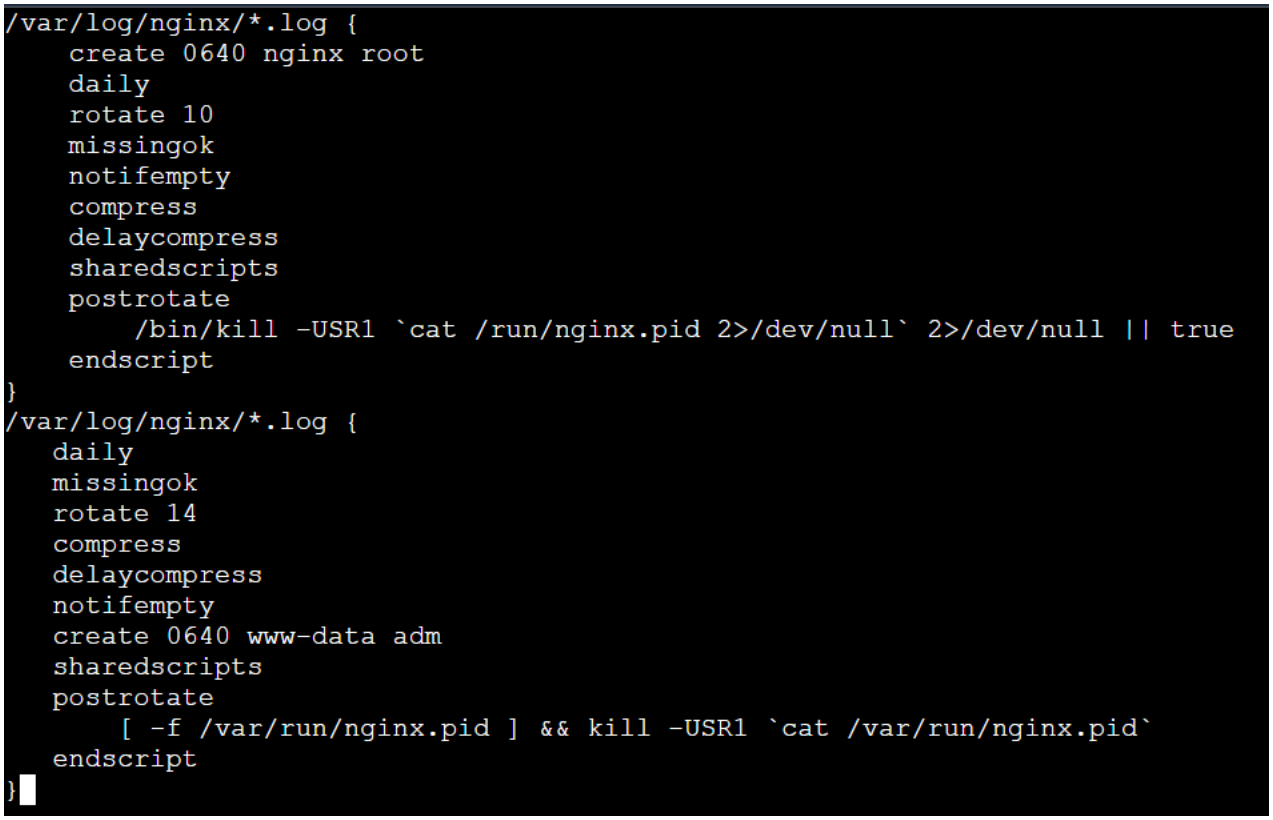


```
root@ip-172-31-12-124 ec2-user]# sudo vim /etc/logrotate.d/
```

Step 2: add the below configuration there.

/var/log/nginx/*.log {

```
  daily
  missingok
  rotate 14
  compress
  delaycompress
  notifempty
  create 0640 www-data adm
  sharedscripts
  postrotate
    [ -f /var/run/nginx.pid ] && kill -USR1 `cat /var/run/nginx.pid`
  endscript
  }
```



```
/var/log/nginx/*.log {
    create 0640 nginx root
    daily
    rotate 10
    missingok
    notifempty
    compress
    delaycompress
    sharedscripts
    postrotate
        /bin/kill -USR1 `cat /run/nginx.pid 2>/dev/null` 2>/dev/null || true
    endscript
}
/var/log/nginx/*.log {
   daily
   missingok
   rotate 14
   compress
   delaycompress
   notifempty
   create 0640 www-data adm
   sharedscripts
   postrotate
       [ -f /var/run/nginx.pid ] && kill -USR1 `cat /var/run/nginx.pid`
   endscript
}
```

Step 3 : Create log analysis script in “sudo nano /usr/local/bin/nginx_log_analysis.sh”.

Step 4: add the below script their.

#!/bin/bash

LOG_FILE="/var/log/nginx/access.log"

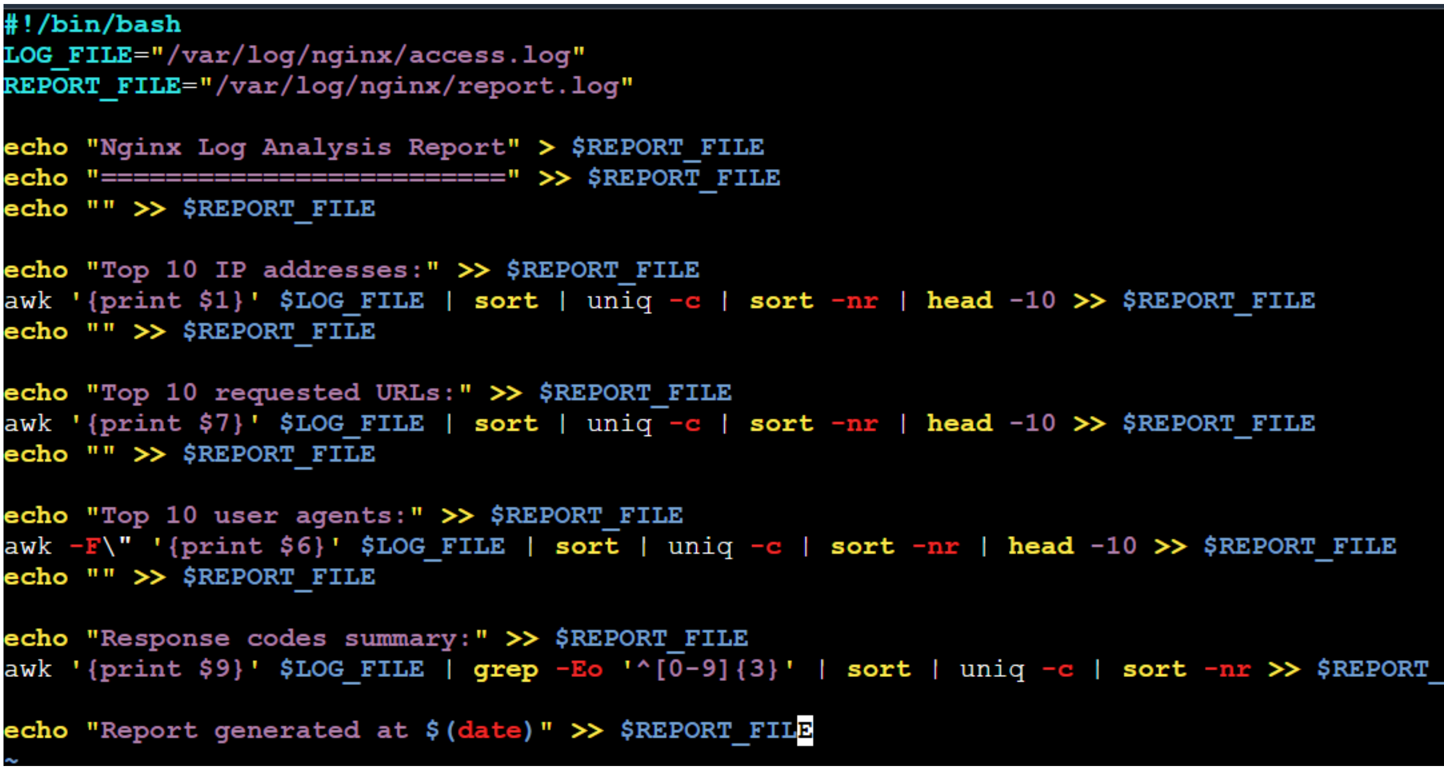
REPORT_FILE="/var/log/nginx/report.log"

echo "Nginx Log Analysis Report" > $REPORT_FILE

echo "==========================" >> $REPORT_FILE

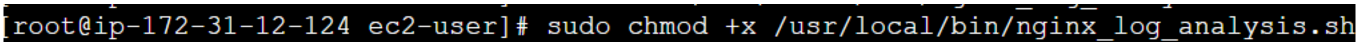
```
echo "" >> $REPORT_FILE

echo "Top 10 IP addresses:" >> $REPORT_FILE
awk '{print $1}' $LOG_FILE | sort | uniq -c | sort -nr | head -10 >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Top 10 requested URLs:" >> $REPORT_FILE
awk '{print $7}' $LOG_FILE | sort | uniq -c | sort -nr | head -10 >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Top 10 user agents:" >> $REPORT_FILE
awk -F\" '{print $6}' $LOG_FILE | sort | uniq -c | sort -nr | head -10 >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Response codes summary:" >> $REPORT_FILE
awk '{print $9}' $LOG_FILE | grep -Eo '^[0-9]{3}' | sort | uniq -c | sort -nr >> $REPORT_FILE
```

echo "Report generated at $(date)" >> $REPORT_FILE



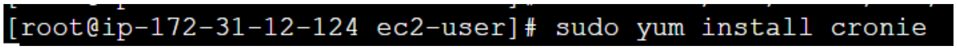
```
#!/bin/bash
LOG_FILE="/var/log/nginx/access.log"
REPORT_FILE="/var/log/nginx/report.log"

echo "Nginx Log Analysis Report" > $REPORT_FILE
echo "=========================" >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Top 10 IP addresses:" >> $REPORT_FILE
awk '{print $1}' $LOG_FILE | sort | uniq -c | sort -nr | head -10 >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Top 10 requested URLs:" >> $REPORT_FILE
awk '{print $7}' $LOG_FILE | sort | uniq -c | sort -nr | head -10 >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Top 10 user agents:" >> $REPORT_FILE
awk -F\" '{print $6}' $LOG_FILE | sort | uniq -c | sort -nr | head -10 >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Response codes summary:" >> $REPORT_FILE
awk '{print $9}' $LOG_FILE | grep -Eo '^[0-9]{3}' | sort | uniq -c | sort -nr >> $REPORT_

echo "Report generated at $(date)" >> $REPORT_FILE
~
```

Step 5 : then give the execute permission to this file



```
[root@ip-172-31-12-124 ec2-user]# sudo chmod +x /usr/local/bin/nginx_log_analysis.sh
```

Step 6 : then install the cron for crontab then start it.



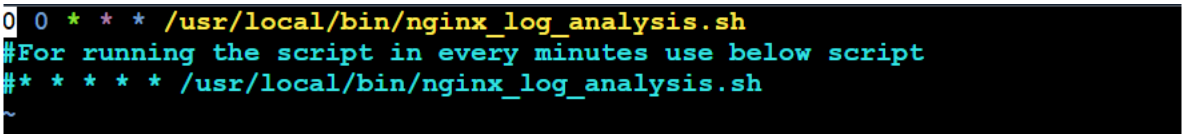
```
[root@ip-172-31-12-124 ec2-user]# sudo yum install cronie
```

Step 7 : using crontab -e schedule the below script.

0 0 * * * /usr/local/bin/nginx_log_analysis.sh

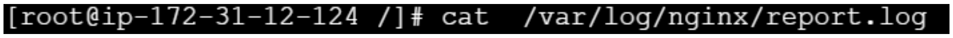
#For running the script in every minutes use below script

#* * * * * /usr/local/bin/nginx_log_analysis.sh



```
0 0 * * * /usr/local/bin/nginx_log_analysis.sh
#For running the script in every minutes use below script
#* * * * * /usr/local/bin/nginx_log_analysis.sh
~
```

Step 8 : now we successfully get the report of nginx.



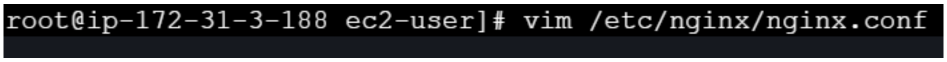
```
[root@ip-172-31-12-124 /]# cat  /var/log/nginx/report.log
```

STEP IV : ALLOW ADMIN LOGIN FOR SPECIFIC IP ONLY.

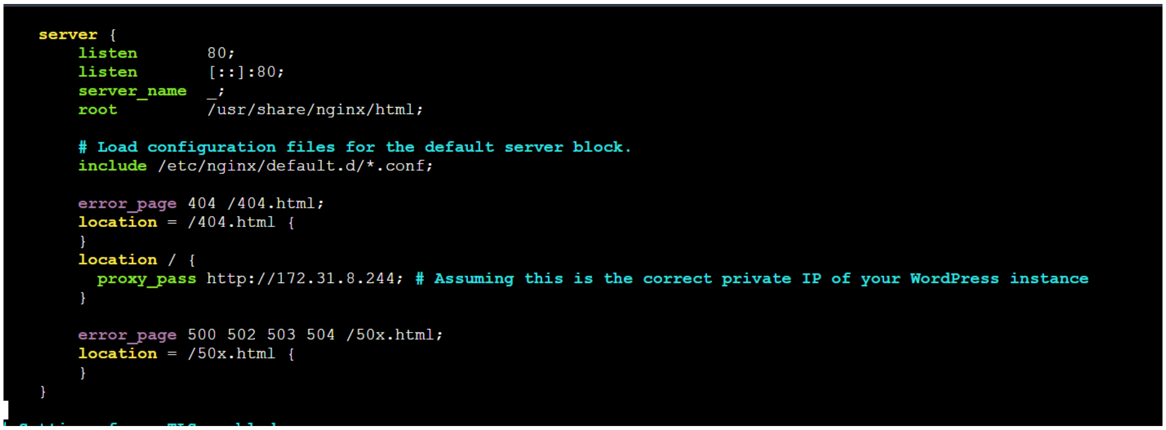
Step 1: Launch the instance using amazon linux.

Step 2: Install the nginx in it.

Step 3: Configure the nginx.conf file.



```
root@ip-172-31-3-188 ec2-user]# vim /etc/nginx/nginx.conf
```



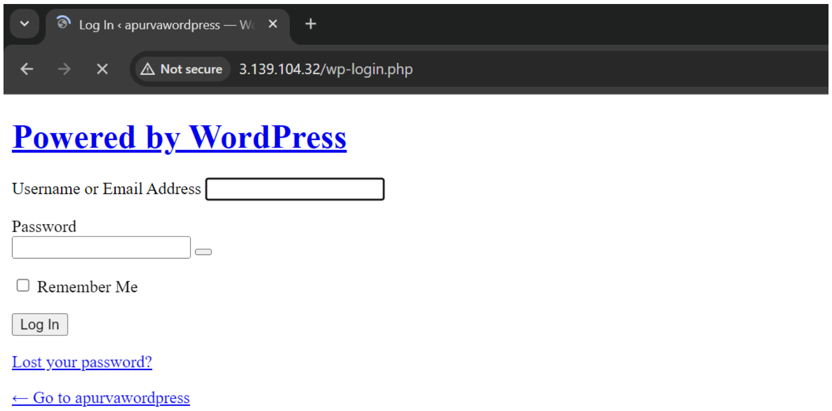
```
    server {
        listen       80;
        listen       [::]:80;
        server_name  _;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
        location = /404.html {
        }
        location / {
          proxy_pass http://172.31.8.244; # Assuming this is the correct private IP of your WordPress instance
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
```

Step 4 : save and close it.

Step 5: then stop nginx again start and enable it.

Step 6: edit the security group assign the public ip of admin.

Step 7 : see here, now only admin can access login page.



Powered by WordPress

Username or Email Address

Password

□ Remember Me

Log In

Lost your password?

← Go to apurvawordpress