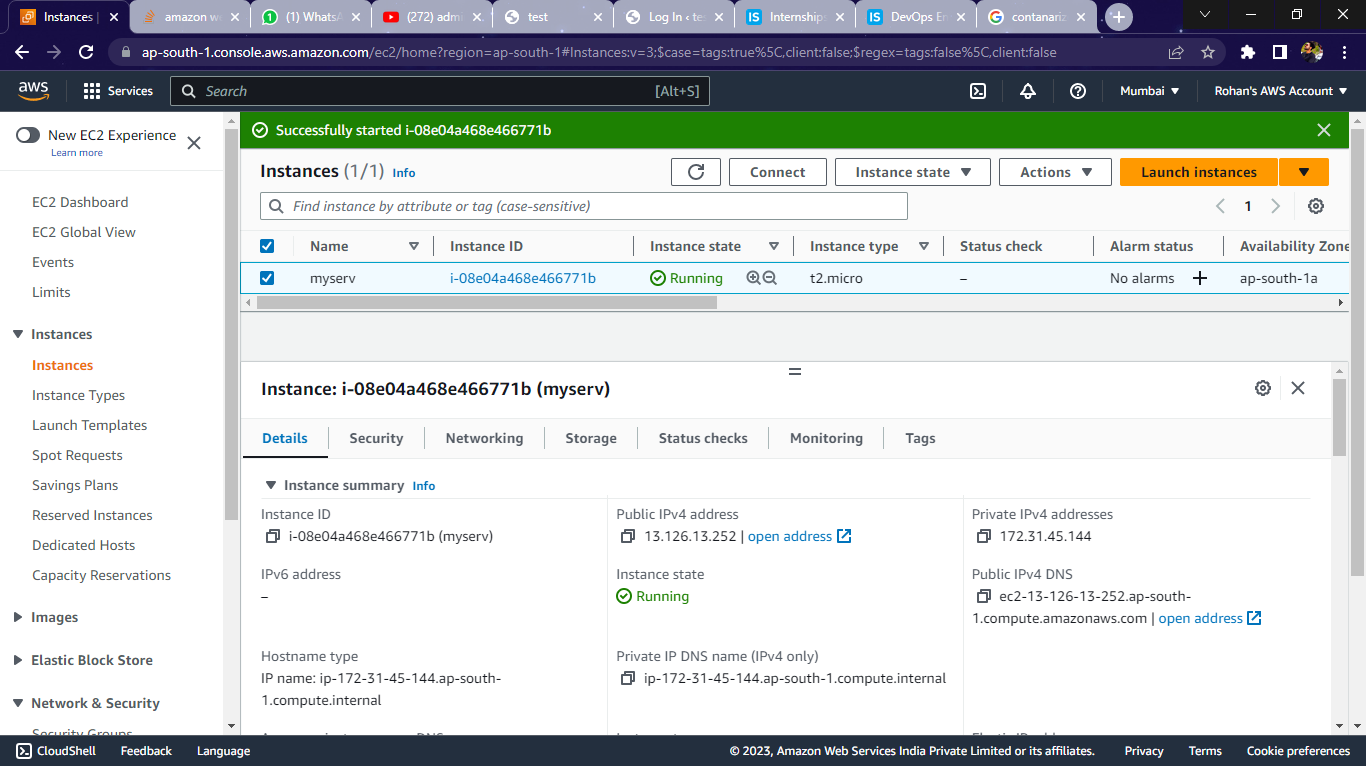
**Project**

**Name:- Rohan Kamble Github:-** [**https://github.com/rohankamble103**](https://github.com/rohankamble103) **Linkedin:- https://www.linkedin.com/in/rohan-kamble-79705a217/**

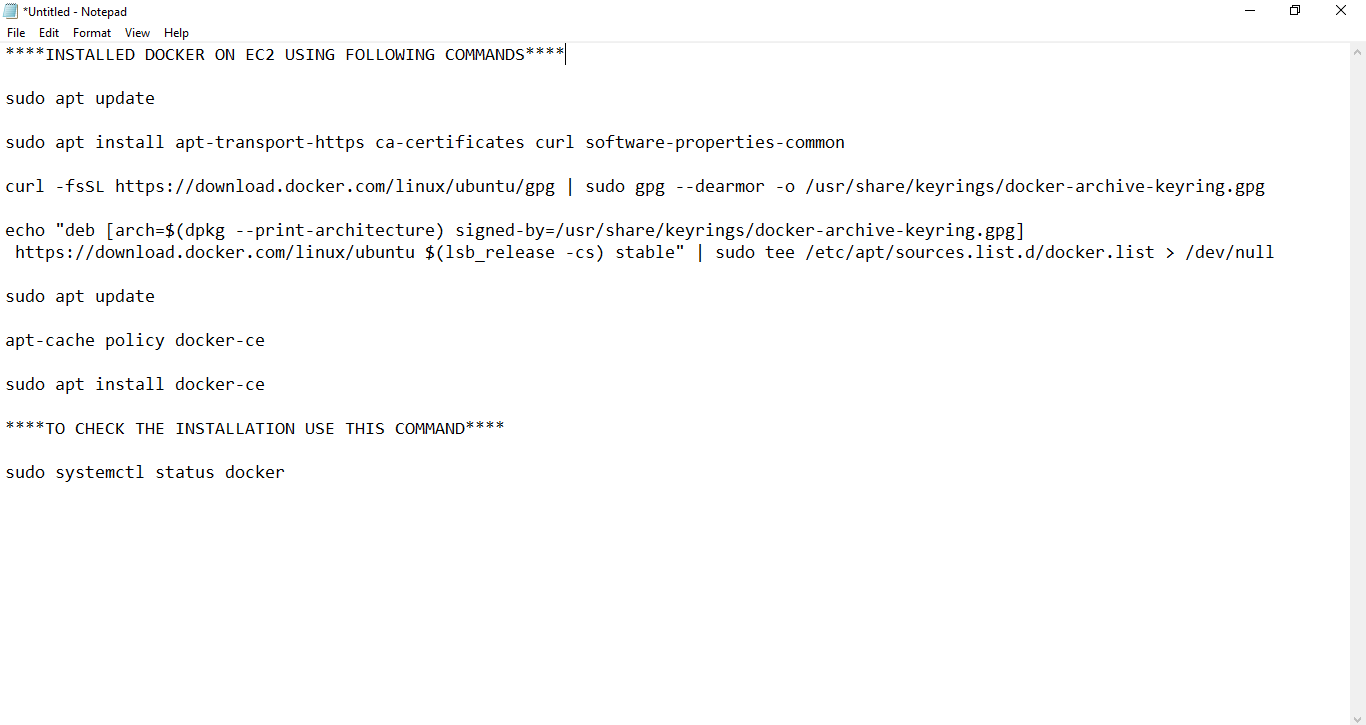
**The problem statement requires deploying a sample WordPress website, protecting it with a Nginx reverse proxy, and allowing admin login from a specific IP address only. Additionally, the candidate must enable log rotation, write a script to analyze Nginx logs, and provide a report.**

1. First of all create a instance of your desired size and configure it as per your need.



2. We need to install docker and docker-compose using following commands to containerize the Images

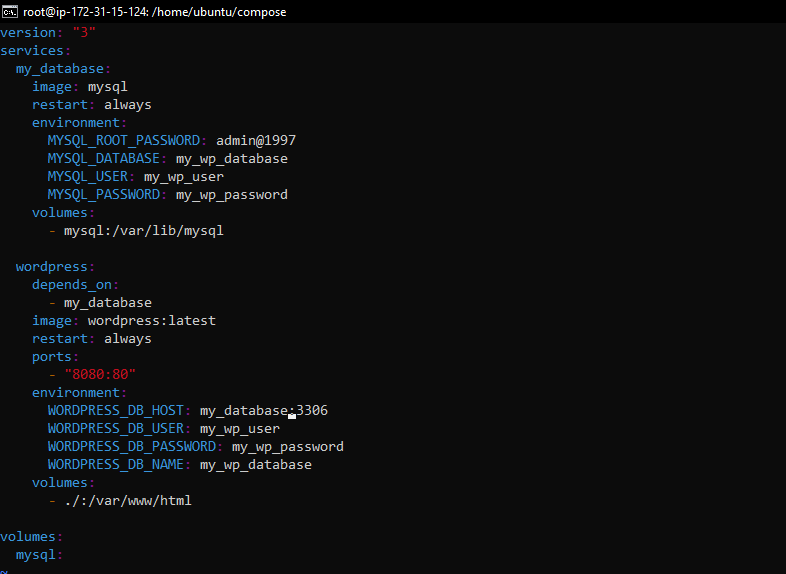
#sudo apt install docker-compose -y



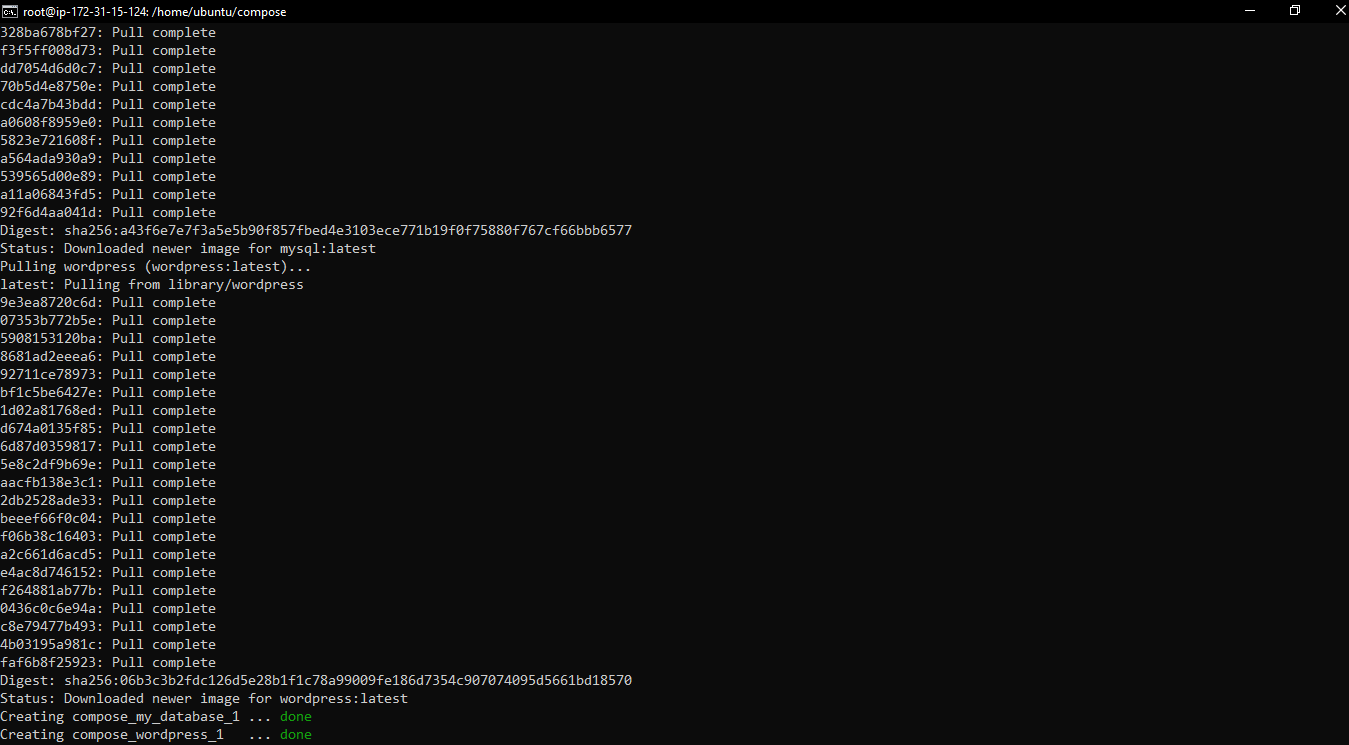
3. Creating a folder named as compose and in that folder I have created a docker-compose file named as docker-compose.yml by using commands

#mkdir compose

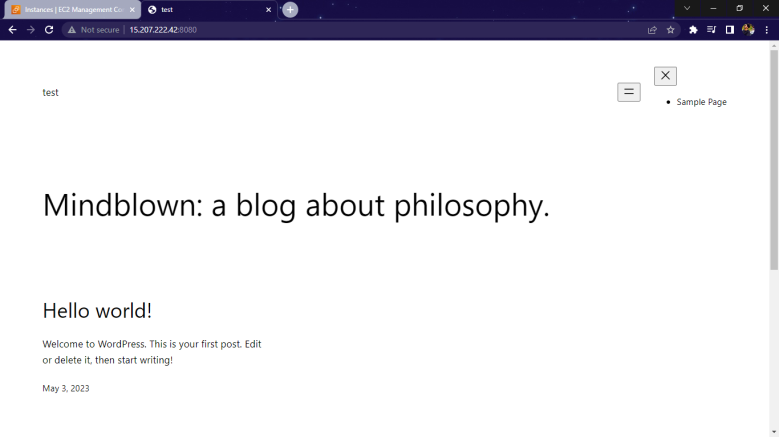
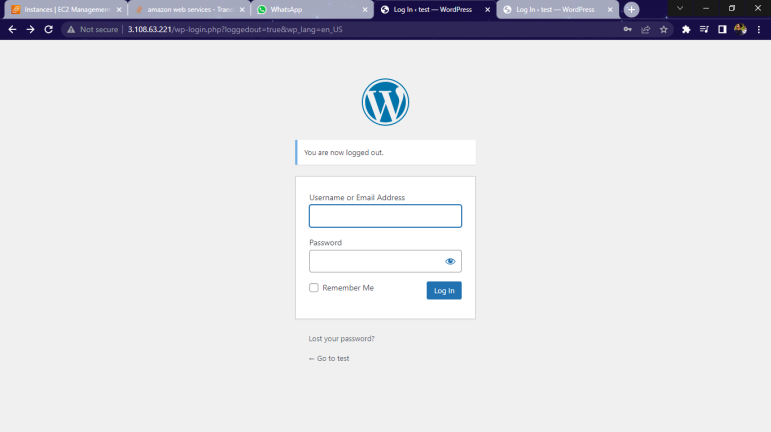
#vi docker-compose.yml



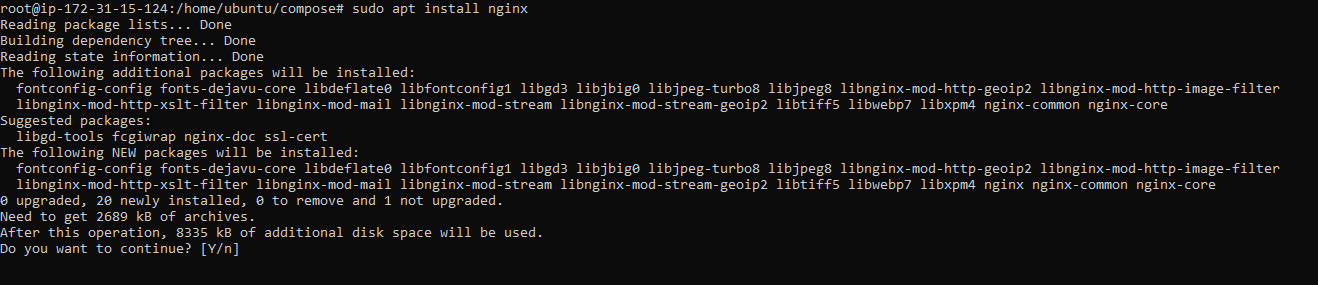
4. Launching two containers of wordpress sample website and database with the help of docker compose



5. Create Admin User for the wordpress sample website and launching both the admin page and the website page

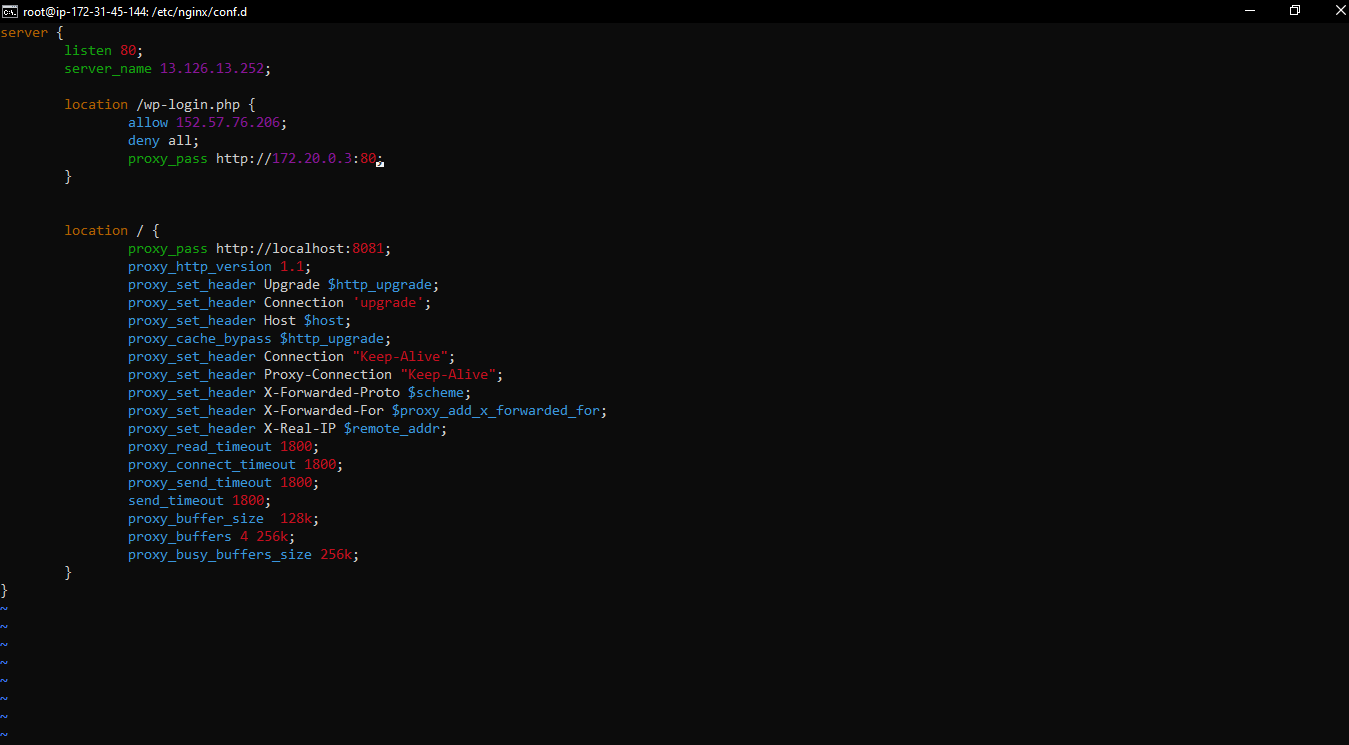
6. Installing Nginx web server to apply security in the form of reverse proxy to the container.



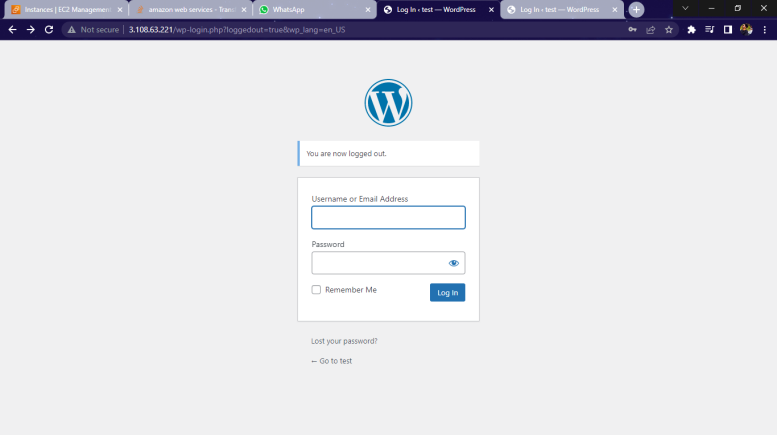
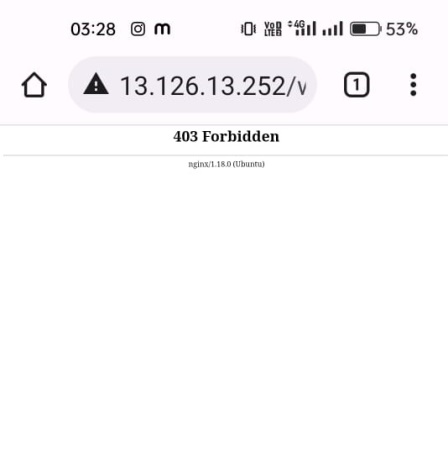
7. Creating a conf file in the “/etc/nginx/conf.d/wordpress.rohan.conf”, By using this file we are applying reverse proxy and also we are restricting the admin access of the website using allow “private IP”;

Nginx also look for wp-login.php in the localserver and not able to find the webpage,

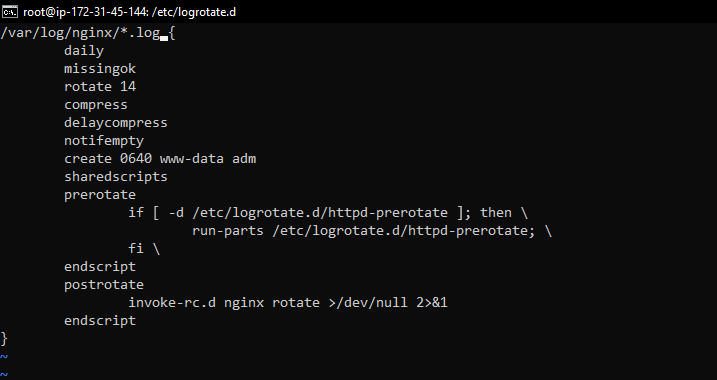
So we have to give the containers Ip (172.20.0.3:port) so that nginx can look for wp-login.php in container



8. Check the syntax by “ nginx –t ” and if the test is ok then restart the nginx by command systemctl restart nginx, ow you don’t need to mention port in the address bar and also no one else can find the admin page

9. Create a file nginx in /etc/logrotate.d/ and you will mention each and every detail of information you want from logs, for example :-

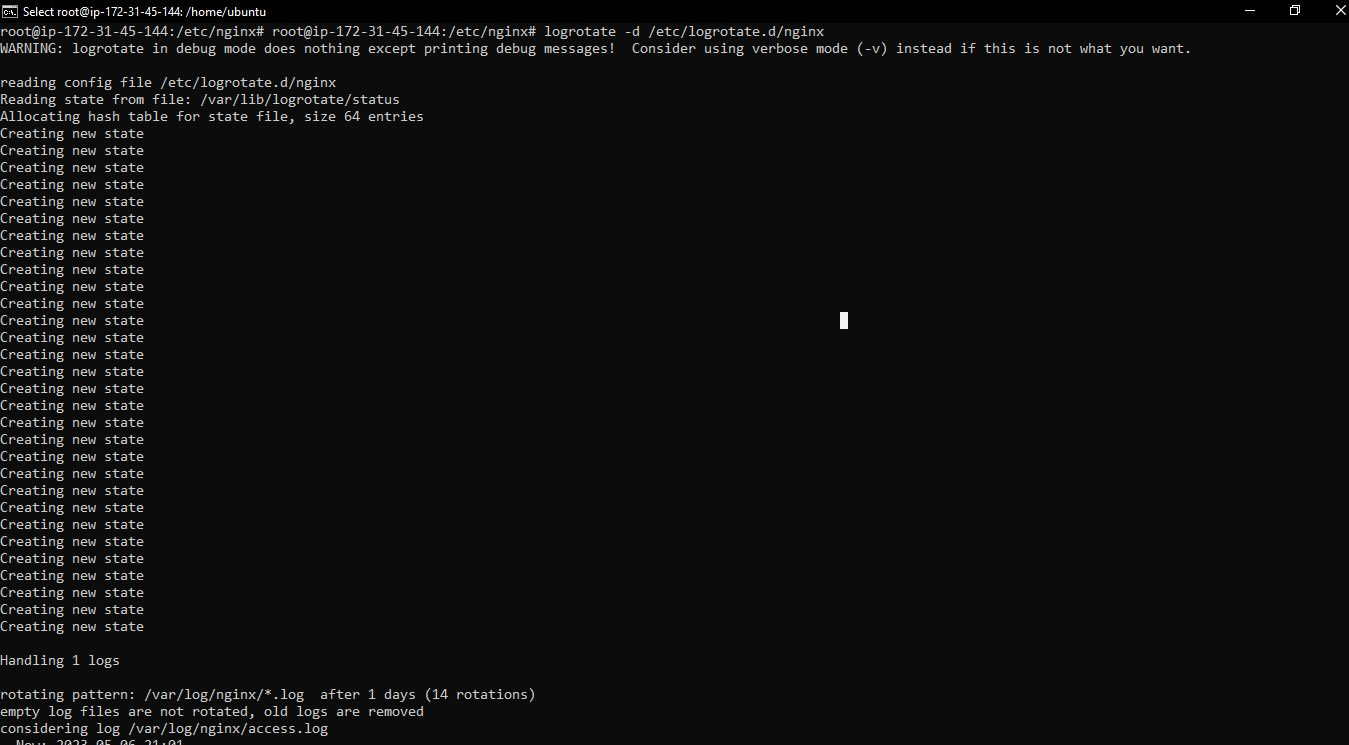


\***daily**:- log files should be rotated every day. \***missingok**:- log file is missing, the rotation should proceed without throwing an error. \***rotate 14**:- maximum of 14 rotated logs should be kept. \***compress**:- rotated log files should be compressed. \* **delaycompress**:-compression of rotated log files should be delayed until the next rotation cycle.\***notifempty**:- rotated log files should not be rotated if they are empty. \***create**:- create log with given permissions.\***sharedscripts**:- **postrotate** script should only be run once after all logs have been rotated.\***prerotate**:- commands enclosed in the block should be executed before the log files are rotated.\***postrotate**:- commands enclosed in the block should be executed after the log files have been rotated.

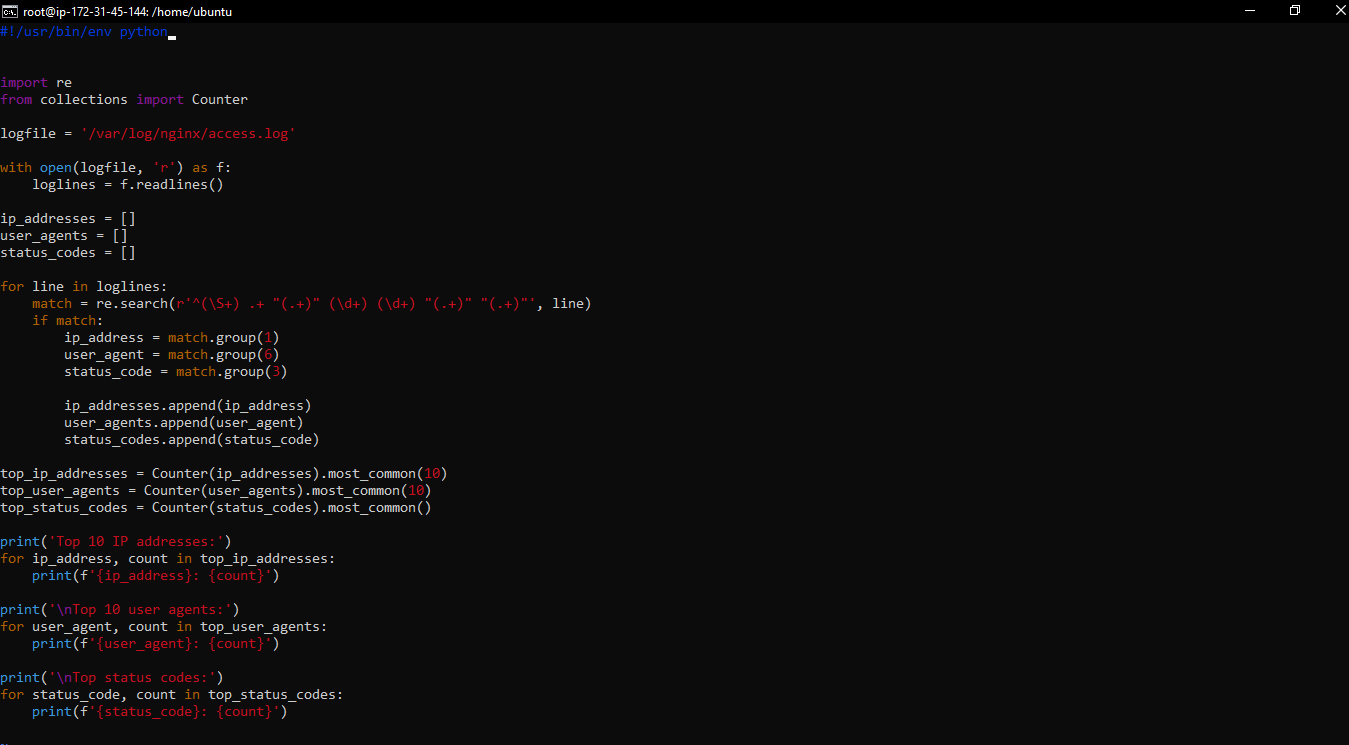
10. Test the configuration using the following command and also enable it with the second command.

# logrotate -d /etc/logrotate.d/nginx

# logrotate /etc/logrotate.d/nginx

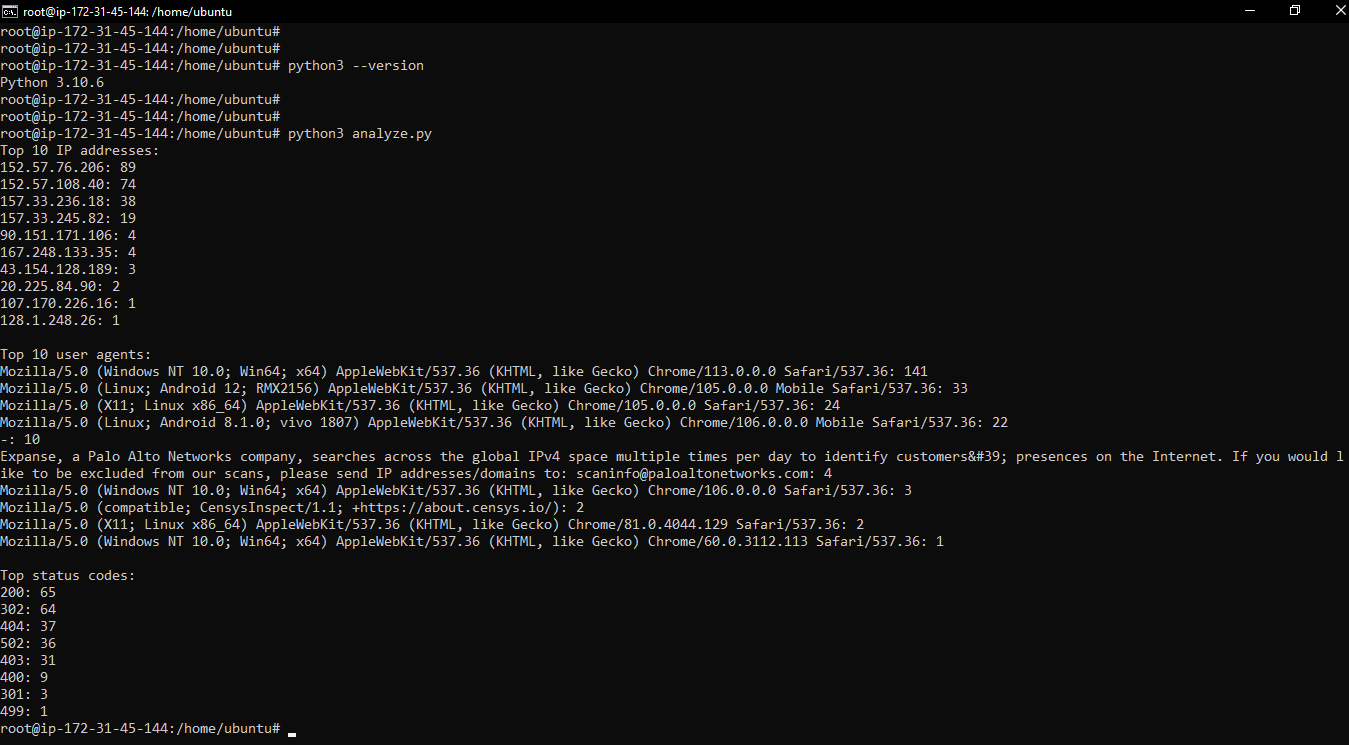


11. Create a Python Script to analyse the logs according to our need and print the output



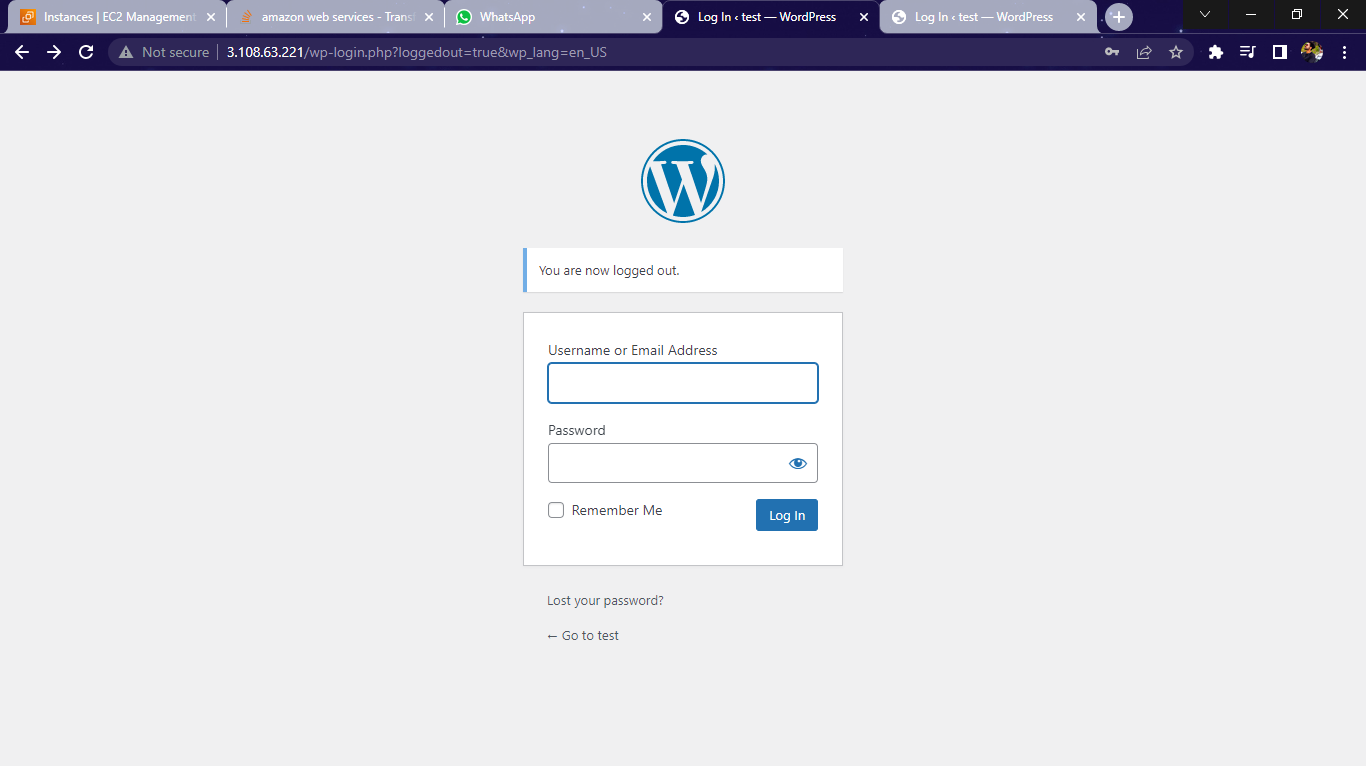
12. To Execute the file we need to change the permission of the file and add execute permission by using following command

#chmod 755 analyze.py

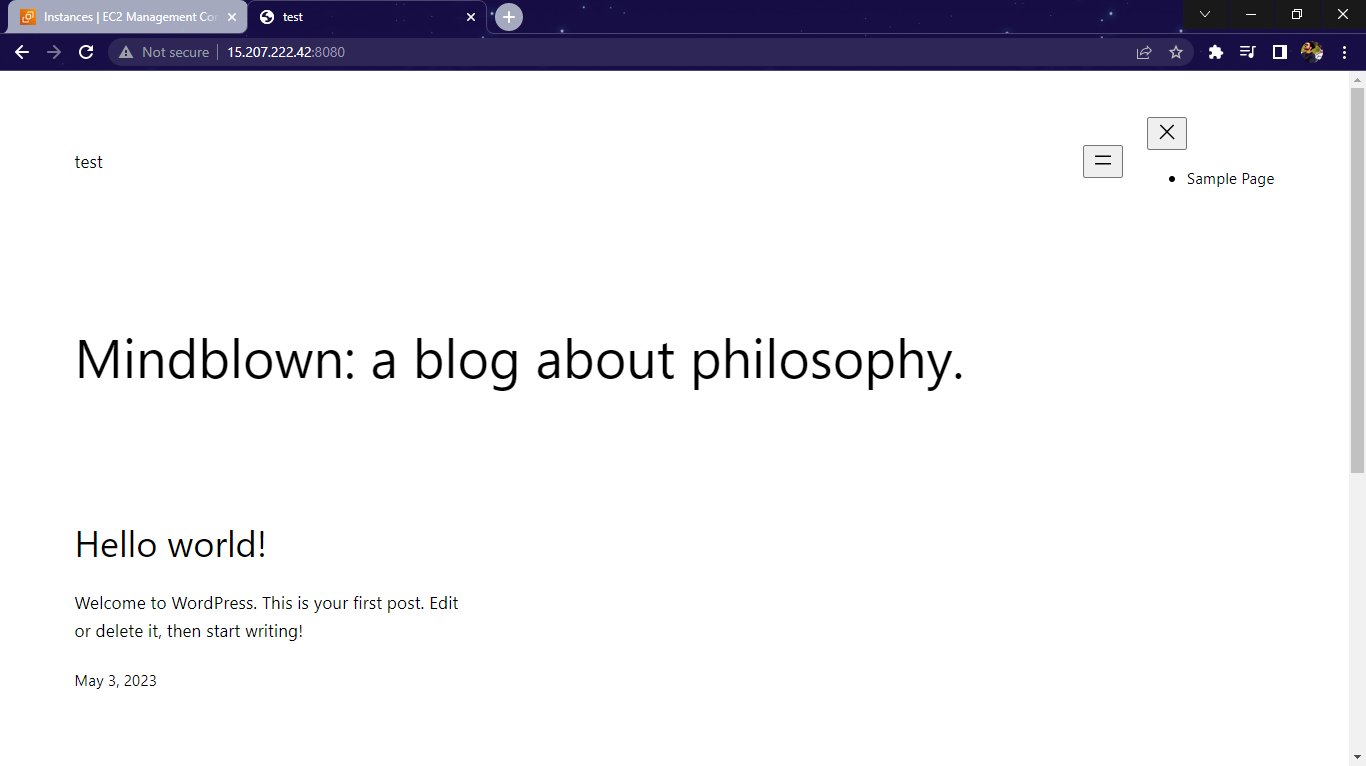


13. To Run the script we need to check is ther python installed or not, If it is not we need to install it to our system





14. Run the script by command “python3 analyze.py” and it will print the output as you needed and you can also story the output of the file into another file by using “python3 analyze.py > rk.txt” and the data will be stored in the file rk.txt



Regards,

Rohan Kamble