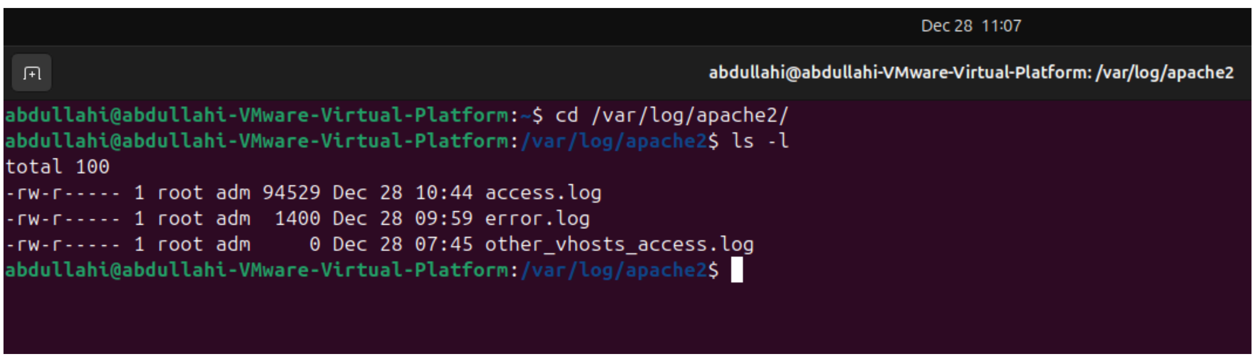
Apache Web Server Log Analysis

Introduction

In this project, I explored Apache log files to understand and analyze server activity by focusing on `access.log` and `error.log`. I began by navigating to the log directory and used commands like `head` and `grep` to filter log entries based on criteria such as IP addresses, HTTP status codes, and specific errors. Additionally, I utilized tools like `awk` to summarize data, identifying the number of requests per IP, requests per day, and the most accessed URLs. Through this analysis, I gained insights into server usage patterns and error events, demonstrating the practical value of log file examination in server management.

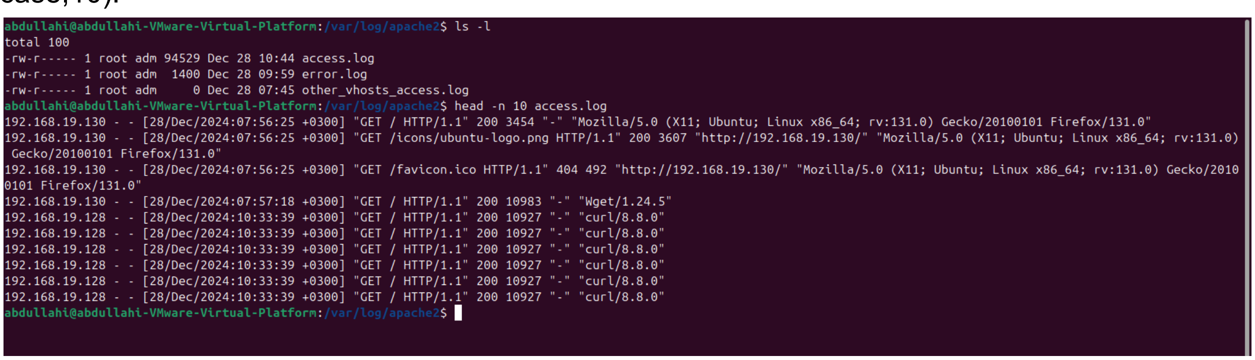
Accessing Apache log files

I first of all started with navigating to the apache log directory using the command `cd /var/log/apache2/` and then listed the available files in the directory which are 3 files as you can see in the screenshot below, for this project we are working on access.log and error.log.



Understanding access logs

I started by using the `head -n 10 access.log` to display the first 10 lines of the access.log file. The head command reads the file and outputs only the specified number of lines (in this case,10).



The output of the command shows the first 10 lines of the file and the following is the explanation of the first line;

192.168.19.130 - - [28/Dec/2024:07:56:25 +0300] "GET / HTTP/1.1" 200 3454 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0"

Here's a brief explanation of the log entry:

1. **`192.168.19.130`**: This is the IP address of the client (user) that made the request to the server. It shows where the request originated.
2. **`- -`**: These are placeholders for the identity and user authentication information, which are usually not used (hence, they appear as `-`).
3. **`[28/Dec/2024:07:56:25 +0300]`**: This is the timestamp of the request, indicating that it was made on December 28, 2024, at 07:56:25 AM, in the +0300 time zone.
4. **`"GET / HTTP/1.1"`**: This shows the HTTP request:
    - `GET`: The HTTP method used (GET requests are for retrieving resources).
    - `/`: The requested resource, in this case, the root of the web server.
    - `HTTP/1.1`: The HTTP protocol version used for the request.
5. **`200`**: The HTTP status code returned by the server. A `200` indicates the request was successful.
6. **`3454`**: The size of the response in bytes, indicating that the server sent 3,454 bytes of data to the client.
7. **`"-"`**: The referrer, which is empty (`"-"`) in this case, meaning no referring page was provided.
8. **`"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0"`**: The user agent string, indicating the client's browser and operating system. Here, it's Firefox 131.0 running on Ubuntu Linux (64-bit).

This log entry shows that a client with the IP `192.168.19.130` successfully accessed the server's root (`/`) using Firefox on Ubuntu, and the server returned a `200` status with 3,454 bytes of data.

Filtering log entries

I used `grep '192.168.19.128' access.log` command to filter log entries based on specific criteria such as a particular ip address.

```
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$ grep '192.168.19.128' access.log
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:39 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
```

```
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:33:52 +0300] "GET / HTTP/1.1" 200 10927 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:36:58 +0300] "GET /image.jpg HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:37:14 +0300] "GET /image.jpg HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.128 - - [28/Dec/2024:10:37:33 +0300] "GET /style.css HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:37:53 +0300] "GET /about.html HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:39:36 +0300] "GET / HTTP/1.1" 200 3454 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.19.128 - - [28/Dec/2024:10:39:36 +0300] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://192.168.19.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/2
0100101 Firefox/115.0"
192.168.19.128 - - [28/Dec/2024:10:39:36 +0300] "GET /favicon.ico HTTP/1.1" 404 492 "http://192.168.19.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Fir
efox/115.0"
192.168.19.128 - - [28/Dec/2024:10:41:04 +0300] "GET /nonexistentpage HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:41:16 +0300] "GET /nonexistentpage HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.128 - - [28/Dec/2024:10:44:04 +0300] "GET / HTTP/1.1" 200 3454 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.19.128 - - [28/Dec/2024:10:44:04 +0300] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://192.168.19.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/2
0100101 Firefox/115.0"
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$
```

This shows all requests from the specified ip address which is 192.168.19.128.

The next one is Filtering log entries by HTTP status code, e.g., for 404 errors and i used the command `grep ' 404 ' access.log`

```
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$ grep ' 404 ' access.log
192.168.19.130 - - [28/Dec/2024:07:56:25 +0300] "GET /favicon.ico HTTP/1.1" 404 492 "http://192.168.19.130/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/2010
0101 Firefox/131.0"
192.168.19.130 - - [28/Dec/2024:10:36:45 +0300] "GET /image.jpg HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.128 - - [28/Dec/2024:10:36:58 +0300] "GET /image.jpg HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:37:14 +0300] "GET /image.jpg HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.128 - - [28/Dec/2024:10:37:33 +0300] "GET /style.css HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:37:53 +0300] "GET /about.html HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.130 - - [28/Dec/2024:10:38:08 +0300] "GET /about.html HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.130 - - [28/Dec/2024:10:38:40 +0300] "GET /style.css HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.128 - - [28/Dec/2024:10:39:36 +0300] "GET /favicon.ico HTTP/1.1" 404 492 "http://192.168.19.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Fir
efox/115.0"
192.168.19.128 - - [28/Dec/2024:10:41:04 +0300] "GET /nonexistentpage HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:41:16 +0300] "GET /nonexistentpage HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.130 - - [28/Dec/2024:10:41:48 +0300] "GET /nonexistentpage HTTP/1.1" 404 493 "-" "Wget/1.24.5"
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$
```

This one i Combined both filters to find 404 errors from a specific IP address and i used the command `grep '192.168.19.128' access.log | grep ' 404 '`

```
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$ grep '192.168.19.128' access.log | grep ' 404 '
192.168.19.128 - - [28/Dec/2024:10:36:58 +0300] "GET /image.jpg HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:37:14 +0300] "GET /image.jpg HTTP/1.1" 404 493 "-" "Wget/1.24.5"
192.168.19.128 - - [28/Dec/2024:10:37:33 +0300] "GET /style.css HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:37:53 +0300] "GET /about.html HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:39:36 +0300] "GET /favicon.ico HTTP/1.1" 404 492 "http://192.168.19.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Fir
efox/115.0"
192.168.19.128 - - [28/Dec/2024:10:41:04 +0300] "GET /nonexistentpage HTTP/1.1" 404 437 "-" "curl/8.8.0"
192.168.19.128 - - [28/Dec/2024:10:41:16 +0300] "GET /nonexistentpage HTTP/1.1" 404 493 "-" "Wget/1.24.5"
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$
```

Analyzing error logs

I started with displaying the contents of the error.log using the `cat error.log` command

```
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$ cat error.log
[Sat Dec 28 07:45:54.778206 2024] [mpm_event:notice] [pid 6990:tid 6990] AH00489: Apache/2.4.62 (Ubuntu) configured -- resuming normal operations
[Sat Dec 28 07:45:54.778288 2024] [core:notice] [pid 6990:tid 6990] AH00094: Command line: '/usr/sbin/apache2'
[Sat Dec 28 08:18:05.484934 2024] [mpm_event:notice] [pid 6990:tid 6990] AH00492: caught SIGWINCH, shutting down gracefully
[Sat Dec 28 08:18:33.092808 2024] [mpm_event:notice] [pid 1701:tid 1701] AH00489: Apache/2.4.62 (Ubuntu) configured -- resuming normal operations
[Sat Dec 28 08:18:33.105169 2024] [core:notice] [pid 1701:tid 1701] AH00094: Command line: '/usr/sbin/apache2'
[Sat Dec 28 08:23:01.340227 2024] [mpm_event:notice] [pid 1701:tid 1701] AH00492: caught SIGWINCH, shutting down gracefully
[Sat Dec 28 08:23:27.039794 2024] [mpm_event:notice] [pid 1548:tid 1548] AH00489: Apache/2.4.62 (Ubuntu) configured -- resuming normal operations
[Sat Dec 28 08:23:27.044292 2024] [core:notice] [pid 1548:tid 1548] AH00094: Command line: '/usr/sbin/apache2'
[Sat Dec 28 09:59:16.696415 2024] [mpm_event:notice] [pid 1548:tid 1548] AH00492: caught SIGWINCH, shutting down gracefully
[Sat Dec 28 09:59:45.087907 2024] [mpm_event:notice] [pid 1728:tid 1728] AH00489: Apache/2.4.62 (Ubuntu) configured -- resuming normal operations
[Sat Dec 28 09:59:45.092564 2024] [core:notice] [pid 1728:tid 1728] AH00094: Command line: '/usr/sbin/apache2'
abdullahi@abdullahi-VMware-Virtual-Platform:/var/log/apache2$
```

Here's a brief explanation of the first error line:
`[Sat Dec 28 08:23:01.340227 2024]`:

- This is the timestamp, indicating that the event occurred on December 28, 2024, at 08:23:01 AM. The precise time is also logged to microseconds (`.340227`).

`[mpm_event:notice]`:

- `mpm_event`: Refers to the **Multi-Processing Module (Event MPM)** being used by Apache. It manages how Apache handles incoming requests.
- `notice`: The log level, which is used for informational messages that don't indicate an error or warning.

`[pid 1701:tid 1701]`:

- `pid`: Process ID (1701 in this case), indicating the specific Apache process responsible for this log entry.
- `tid`: Thread ID (1701 here, the same as the process ID because it's a single-threaded notice).

`AH00492: caught SIGWINCH, shutting down gracefully`:

- `AH00492`: A unique Apache error code identifying this specific event.

- `caught SIGWINCH`: Apache received a **SIGWINCH signal**, which usually indicates that the terminal window size has changed or that the server is being instructed to reload configuration files.
- `shutting down gracefully`: Apache is stopping its current processes in a controlled manner, ensuring that all requests in progress are completed before shutting down.

Summarizing log data

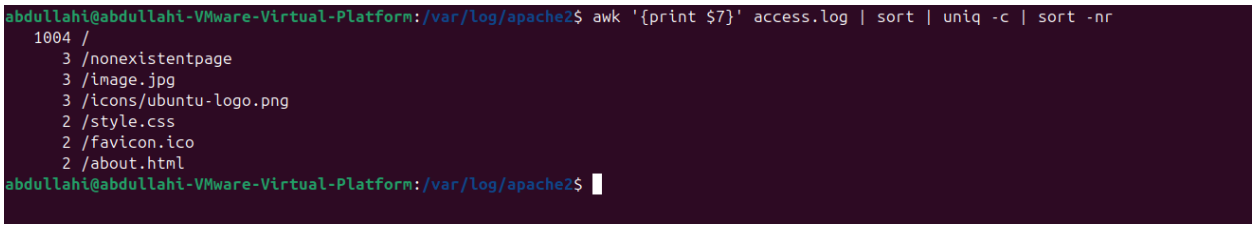
I Summarized log data using `awk '{print $1}' access.log | sort | uniq -c | sort -nr` command to find the number of requests from each IP address:

This will count and sort requests by IP address, showing which IPs are making the most requests.

I Summarized the number of requests per day to get or understand the estimate requests in a day in an organization by using `awk '{print $4}' access.log | cut -d: -f1 | sort | uniq -c` command .

This extracts the date portion from the timestamp and counts requests per day.

I Identified the most requested URLs by using the command awk '{print $7}' access.log | sort | uniq -c | sort -nr



This extracts the requested URLs and counts how many times each was requested.

Conclusion

In conclusion, this project provided me with valuable hands-on experience in working with Apache log files, enhancing my skills in log analysis and data interpretation. I gained a deeper understanding of server activity, including how to identify key patterns, filter specific entries, and summarize large datasets using commands like `grep`, `awk`, and `sort`. This experience has strengthened my analytical thinking and technical proficiency, equipping me with the skills needed to monitor and manage server environments effectively. With these skills, I feel confident and ready to apply log analysis techniques to real-world scenarios, contributing to improved server performance and security.