**LOGS**

**Log files: -** It is a textual data file that stores events, processes, messages, and other data from applications.

**Importance of log files: -** These files store valuable information used to recreate past events, find security flaws, or troubleshoot.

**It can improve:**

1. Improve Performance
2. Make troubleshooting easy
3. Security check
4. Help to understand the behavior of the application.

**Type of logs: -**

1. **Event logs: -** An event log is a high-level log that records system activity data to provide an audit trail for troubleshooting issues. Event logs are essential to understand the behavior of complex systems, particularly in the case of applications with little user interaction. For example, in networks, event logs record network traffic, access, and usage.
2. **System logs: -**A system log records operating system events, such as system changes, startup messages, errors, warnings, and unexpected shutdowns.
3. **Access logs**: -An access log records the list of all requests for individual files that people or applications request from a system. It includes information about user authentication, who requested a particular system file when they asked for it, and other associated information.
4. **Server logs**: - A server log is a log file that a server automatically creates and the number of page requests, client IP addresses, types of requests, and so on.
5. **Web server logs: - logs produced by web application. These logs provide us who visited the site (IP address) or which page is accessed by the client.**

**// In Linux, logs are stored in /var/log/**

**Log Rotation: -**

Process of managing log files periodically archiving or removing old log files to prevent the disk from being filled with unnecessary data.

**Rotation can be done based on:**

* File size (if file exceeds a particular size (for eg 1 MB))
* based on time (weekly, monthly, daily)

**Log rotation directory**: /etc/logrotation.d/

Log rotation can be automated using **Logrotate** software in Linux.

**Practical implementation:**

1. We will create the sample file for testing purposes.
2. In /var/log/ folder we will mkdir and then make .log file in that directory.
3. Then same we will perform in /etc/logrotate.d/
4. After that we will edit that log file using vi command and in that logrotate file we will add 🡪

Var/log/\*.log{

Daily/weekly/monthly

Minsize 10M}

1. We have specified min size 10mb of for the log file which means that whenever the size of the log file exceeds 10 MB it will be rotated.
2. And if we want to archive the log when can write oldir and provide the dir where we want to store our archive and even we can compress our archive log files.

Code 🡪 Var/log/\*.log{

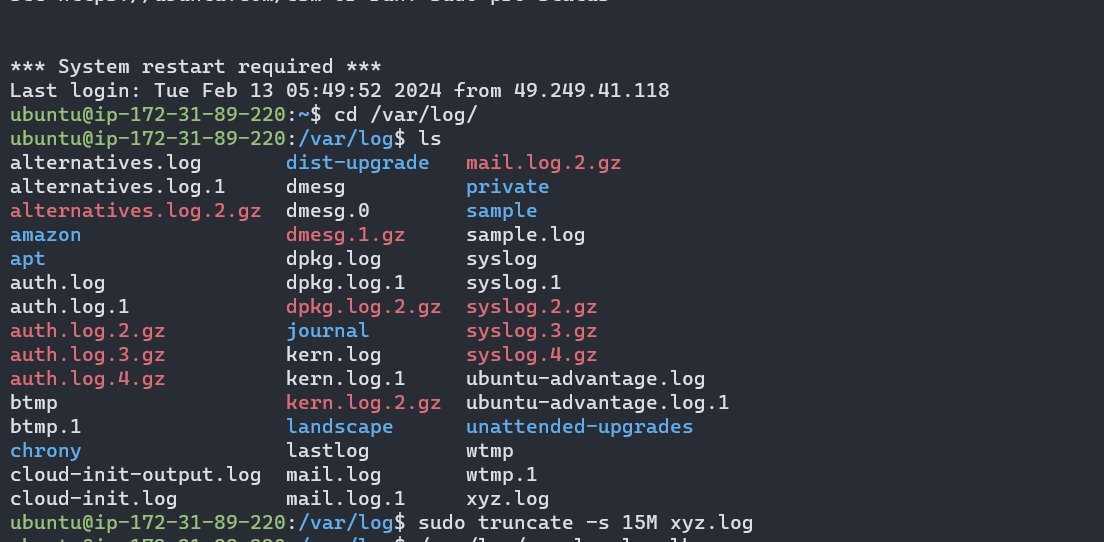
Daily/weekly/monthly

Minsize 10M

Olddir etc/archive.d/

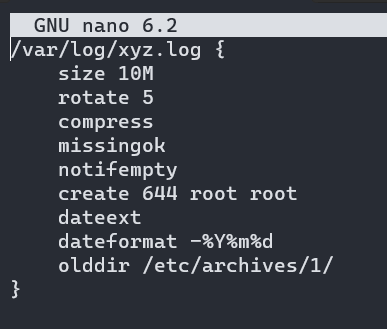
compress

}



Xyz.log is a sample log file

Now, with nano /etc/logrotate.d/xyz command

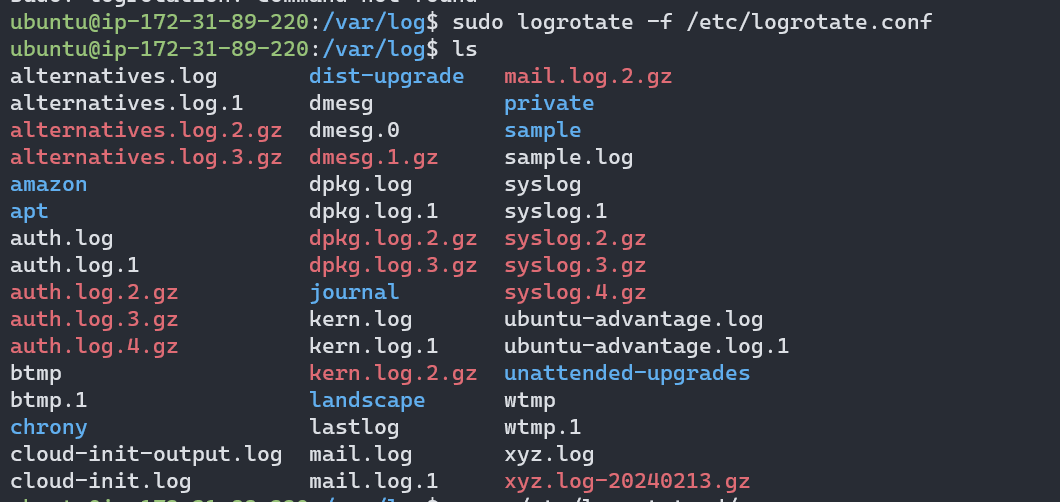


Explanation of the additional directives:

* **dateext**: Appends the date to rotated log files.
* **dateformat -%Y%m%d**: Specifies the date format. In this example, **%Y** represents the four-digit year, **%m** represents the two-digit month, and **%d** represents the two-digit day.
* **size 10M**: Rotate the log file if it exceeds 10 megabytes.
* **rotate 5**: Keep up to 5 rotated log files.
* **compress**: Compress rotated log files.
* **missingok**: Do not generate an error if the log file is missing.
* **notifempty**: Do not rotate the log if it is empty.
* **create 644 root root**: Create a new log file with the specified permissions and ownership if it doesn't exist.



With this command, we will increase the size of the log file by which it will exceed 10MB



And after forcing logrotation we can see our new log file and the old log file with date extension and compressed format.

Now to archive old logs in a specific location we can use olddir and /path/to/archive/

