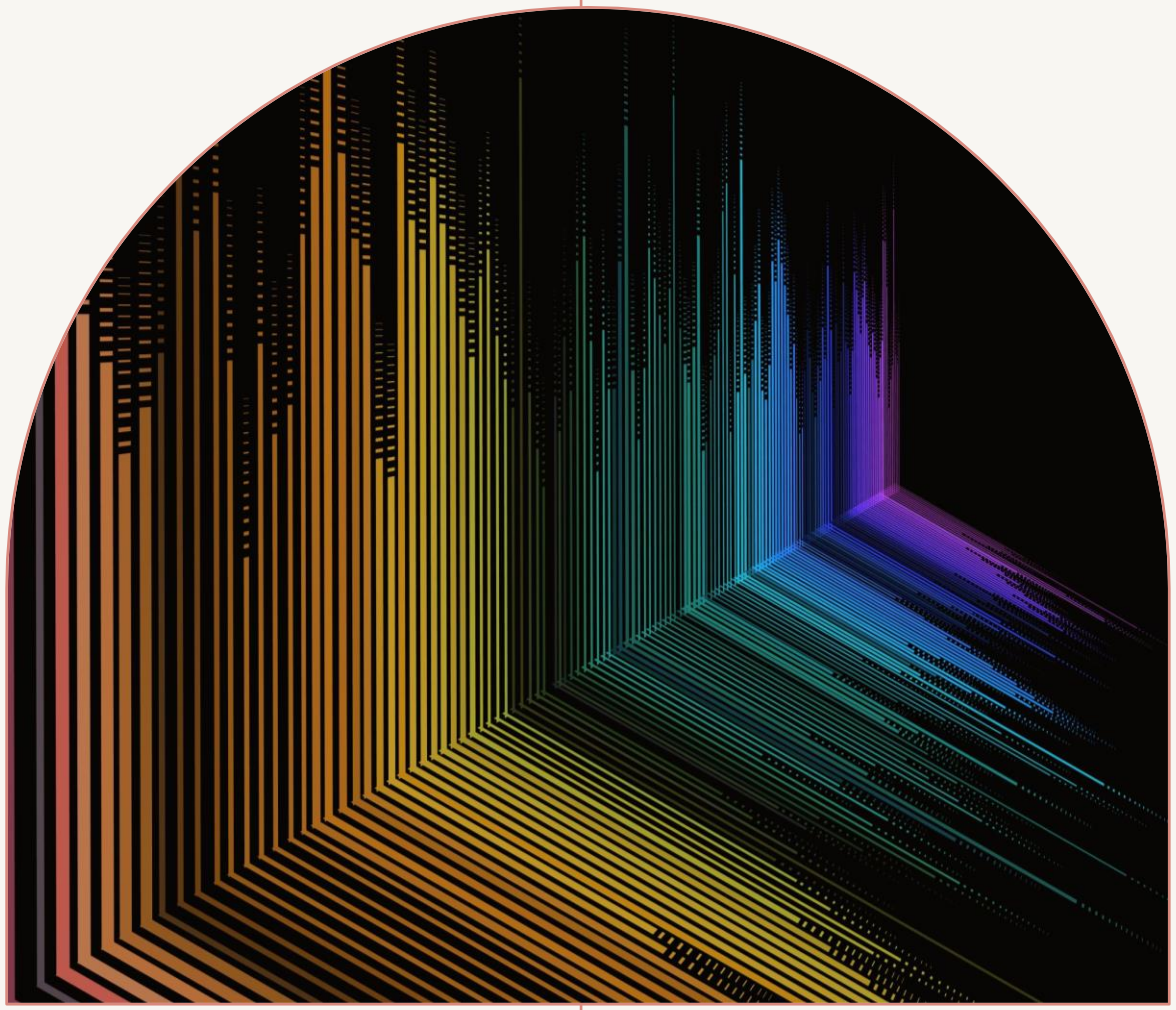
LOGGING and its importance

By Khaja

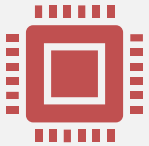


Why ? An Educated Version

Why? A Tollywood Version

Log: Diary of software

What is a log?



A log is a file that records events, errors, or transactions that occur while software or a system is running

It's like a diary for a computer program, noting down what happens, when it happens, and sometimes why it happens.

Logs are crucial for troubleshooting problems, monitoring system performance, and ensuring security.

Types of Logs

Application Logs

System Logs

Security Logs

Audit Logs

Event Logs

Application Logs

Record events related to the application's behavior, such as user actions, system errors, and transactions

An application log typically records activities, events, and errors that occur during the execution of an application. Here's a simplified example of what entries in an application log might look like for a web application

Application Logs: Example

2024-02-09 08:45:12 INFO User 'john_doe' logged in successfully.

2024-02-09 08:47:30 WARNING Attempt to access undefined variable in module 'UserProfile'.

2024-02-09 08:48:05 ERROR Database connection failed: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool.

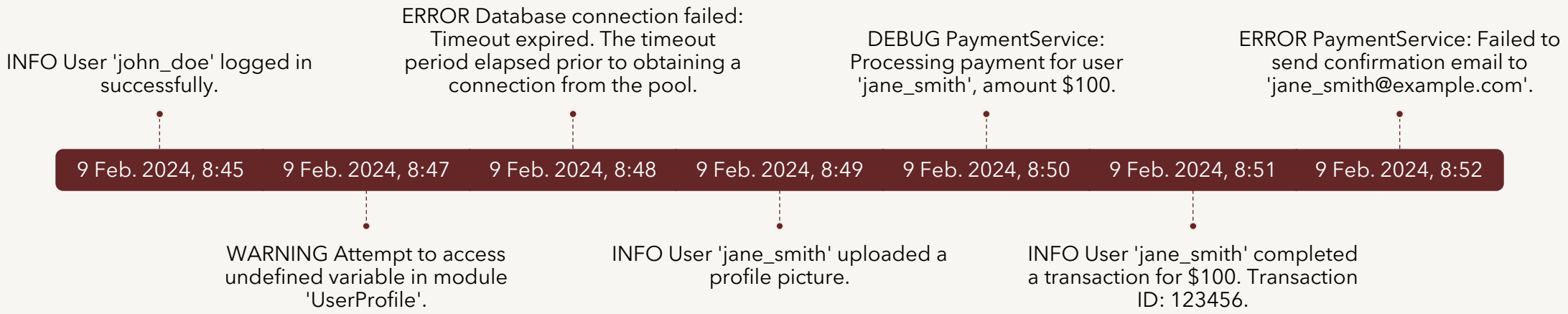
2024-02-09 08:49:17 INFO User 'jane_smith' uploaded a profile picture.

2024-02-09 08:50:03 DEBUG PaymentService: Processing payment for user 'jane_smith', amount $100.

2024-02-09 08:51:00 INFO User 'jane_smith' completed a transaction for $100. Transaction ID: 123456.

2024-02-09 08:52:00 ERROR PaymentService: Failed to send confirmation email to 'jane_smith@example.com'.

Application Logs: Refined view



Who creates Application logs ?

- Application logs are created by the developers of the application. During the development process, programmers write code to generate log messages for various events, actions, or errors that occur within the application. This involves deciding:
  - **What to Log:** Identifying the key events, actions, and errors that should be recorded. This could include user actions, system errors, transaction completions, and more.
  - **Log Severity Levels:** Assigning a severity level to each log message, such as DEBUG, INFO, WARNING, ERROR, or CRITICAL, based on the importance and impact of the event being logged.
  - **Log Format:** Defining a consistent format for log messages, which might include a timestamp, the severity level, a description of the event, and possibly other contextual information like user IDs or transaction IDs.
  - **Log Destination:** Deciding where the log messages should be stored, which could be a file on the server, a database, or a centralized log management system.

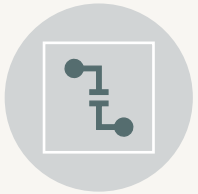
What are log severity Levels ?

- Log severity levels are a classification system used in logging to indicate the importance or urgency of a log message.
- These levels help in categorizing log entries based on their nature and severity, making it easier for developers, system administrators, and automated monitoring tools to prioritize actions and filter log data for analysis

Common log severity levels

DEBUG: Used for detailed diagnostic information, helpful during development or debugging. These messages are usually intended for developers and are not typically enabled in a production environment due to their verbose nature.

ERROR: Represents a failure or issue that affects the operation of the application but does not necessarily halt the entire system. Error logs require immediate attention as they can impact user experience or result in data loss.



INFO: Indicates general operational information that shows the system is functioning correctly. For example, successful completion of a process or a transaction.

CRITICAL (or FATAL): Indicates a severe problem that has caused or is about to cause the application to stop functioning entirely. These messages highlight system crashes or critical conditions that require urgent investigation.

WARNING (or WARN): Signals a potential issue or unexpected event that might not interrupt current operations but should be noted. Warnings may require attention to prevent future errors or issues.

Relevance of Log Severity Levels

Prioritization: Severity levels allow for the prioritization of issues. By categorizing log messages, developers and system administrators can quickly identify and address the most critical problems first.

Filtering and Analysis: When reviewing log files, it's possible to filter messages by severity level, focusing on errors and critical issues while ignoring less severe messages during crisis management.

Automated Monitoring and Alerts: Many log monitoring systems can be configured to trigger alerts based on the severity of log messages. For example, critical errors might immediately alert the IT team via email or SMS, ensuring rapid response to urgent issues.

Performance Optimization: By analyzing warnings and errors, developers can identify areas of the application that need optimization or refactoring, leading to improved performance and user experience.

Log Formatting Standards

Common Event Format (CEF): A text-based format that makes logs more uniform across different devices and applications. It's widely used in security information and event management (SIEM) systems.

Log Event Extended Format (LEEF): Developed by IBM for QRadar SIEM, LEEF is another format used for logging security events in a consistent manner

Syslog Protocol: Syslog is the de facto standard for system logging, widely used across Unix-like systems, networking devices, and even in some Windows-based environments

System Logs

- Capture events at the operating system level, including system errors, boot messages, and status of system services.
- System logs are essential for several reasons, serving as a comprehensive record of events, errors, and operations that occur within an operating system and its components

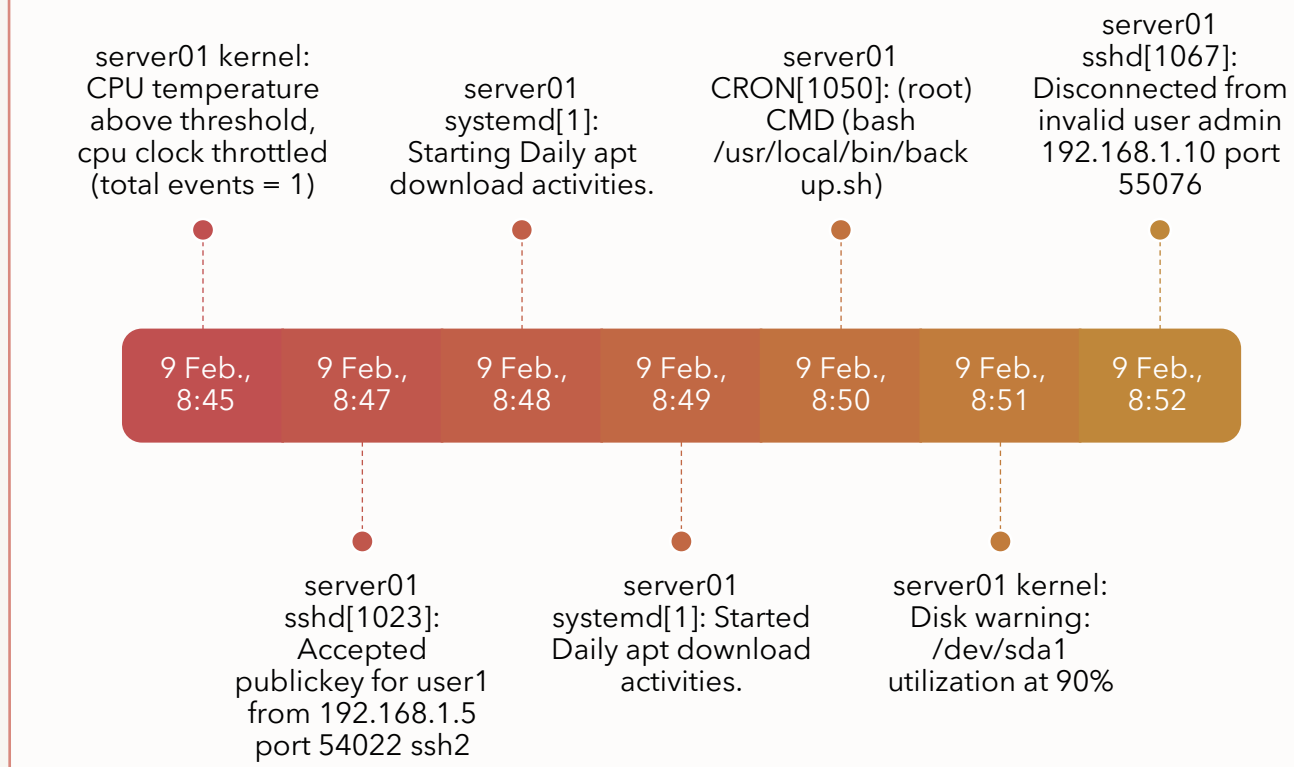
Why System Logs ?

- **Troubleshooting and Diagnostics**
- **Security Monitoring**
- **Compliance and Auditing**
- **Performance Monitoring**
- **System Management**
- **Accountability**

System Log Example

- Feb 09 08:45:12 server01 kernel: CPU temperature above threshold, cpu clock throttled (total events = 1)
- Feb 09 08:47:30 server01 sshd[1023]: Accepted publickey for user1 from 192.168.1.5 port 54022 ssh2
- Feb 09 08:48:05 server01 systemd[1]: Starting Daily apt download activities.
- Feb 09 08:49:17 server01 systemd[1]: Started Daily apt download activities.
- Feb 09 08:50:03 server01 CRON[1050]: (root) CMD (bash /usr/local/bin/backup.sh)
- Feb 09 08:51:00 server01 kernel: Disk warning: /dev/sda1 utilization at 90%
- Feb 09 08:52:00 server01 sshd[1067]: Disconnected from invalid user admin 192.168.1.10 port 55076

System Log Example



Who Creates System Logs ?

- System logs are automatically generated by the operating system (OS) and various system-level components and services
- Here’s how system logs are created across different components:
  - **Operating System Kernel**
  - **System Services and Daemons**
  - **Security and Authentication Services**
  - **Application Software**
  - **Log Management Systems**

System Logs in Linux

- Syslog and Syslog-ng
- Rsyslog
- Journalctl/Systemd-journald
- Logwatch/Logcheck

Syslog

Syslog is one of the most traditional logging systems on Unix-like systems, including Linux. It handles the logging of kernel, system, and application messages. Syslog can be configured to route messages based on their severity, source, or facility to different destinations like files, remote syslog servers, or consoles.

Mar 15 10:22:14 hostname CRON[12345]: (root) CMD (command executed by cron)

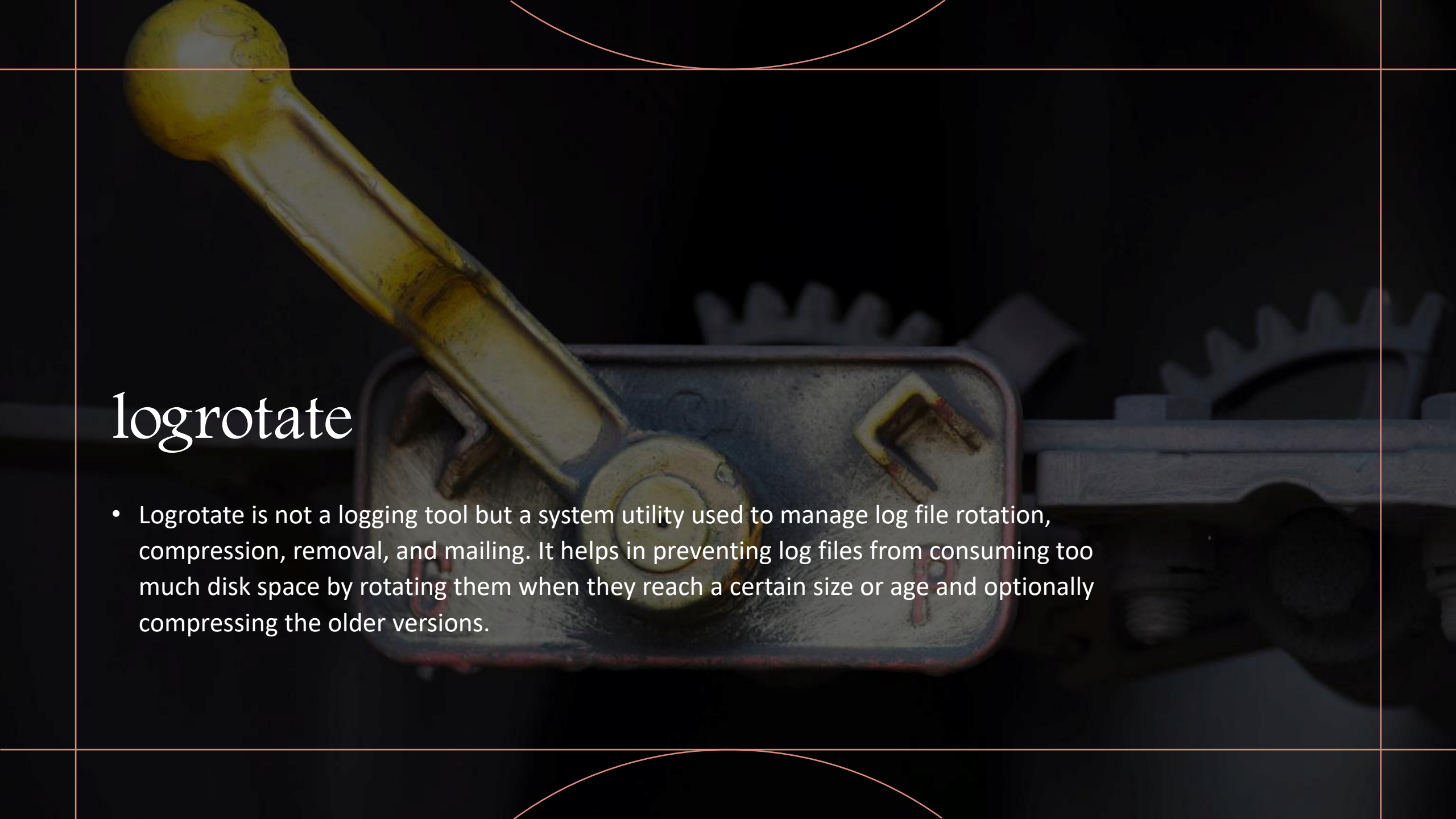
RSyslog

- An enhanced version of Syslog, rsyslog is a powerful and highly configurable logging daemon capable of forwarding log messages to remote servers, writing to databases, and more. It's the default logging solution on many Linux distributions.

journald

Systemd-journald is a journaling service that collects and manages logs from early boot and from applications running on the system. It is part of the systemd suite. Journald provides advanced features like indexing, consistent log formatting, and the ability to include binary data in logs.

Example Command to view Logs
`journalctl -u nginx.service`



logrotate

- Logrotate is not a logging tool but a system utility used to manage log file rotation, compression, removal, and mailing. It helps in preventing log files from consuming too much disk space by rotating them when they reach a certain size or age and optionally compressing the older versions.