Overview of the problem:

*Need to create a command designed for managing system resources and tasks. The new command will be called **sysopctl** and will focus on managing system services, processes, and system health.*

General Instructions

1. **Problem Solving Approach:** The approach to solving the problem is at the discretion of the candidate.
2. **Documentation:** Use Draw.io to create workflow diagrams and system architecture visuals.
3. **Version Control:** All code and configuration files should be committed to a private Git repository.
4. **Confidentiality:** The documentation and code must not be shared with anyone outside of the project team, including colleagues.

Scenario

A customer requires a custom command to enhance their system administration capabilities. Your task is to develop a Bash script that acts as a Linux command to manage system resources effectively.

Command Specifications

- **Command Name:** sysopctl
- **Command Version:** v0.1.0

Section A: Documentation and Basic Features

- **Manual Page:**
  - Create a detailed manual page for sysopctl so users can access full documentation using man sysopctl.
- **Help Option:**
  - Implement a --help option that outlines usage and examples, akin to sysopctl --help.

- **Version Information:**
    - Users should be able to view the command version with: sysopctl --version.

Section B: System Management Operations

Part 1 | Level Easy

- **List Running Services:**
    - Command: $ sysopctl service list
    - Expected Output: List of all active services, similar to systemctl list-units --type=service.
- **View System Load:**
    - Command: $ sysopctl system load
    - Expected Output: Current system load averages, akin to the output from the uptime command.

Part 2 | Level Intermediate

- **Manage System Services:**
    - Start a service: $ sysopctl service start <service-name>
    - Stop a service: $ sysopctl service stop <service-name>
    - Expected Output: Status updates confirming the start or stop of services, similar to systemctl start/stop.
- **Check Disk Usage:**
    - Command: $ sysopctl disk usage
    - Expected Output: Disk usage statistics by partition, similar to df -h.

Part 3 | Advanced Level

- **Monitor System Processes:**
    - Command: $ sysopctl process monitor
    - Expected Output: Real-time process activity, akin to top or htop.
- **Analyze System Logs:**
    - Command: $ sysopctl logs analyze
    - Expected Output: Summary of recent critical log entries, utilizing tools like journalctl.
- **Backup System Files:**
    - Command: $ sysopctl backup <path>

- Expected Output: Confirmation of backup initiation and status, potentially using rsync for file transfers.