

Set 1: Monitoring System Resources for a Proxy Server

"shell script file lamp.sh"

```
#!/bin/bash
```

```
# Function to display the top 10 most used applications
```

```
top_apps() {  
    echo "Top 10 Most Used Applications:"  
    echo "-----"  
    ps aux --sort=-%cpu | awk 'NR<=10 {print $0}' | head -n 10  
    echo ""  
    ps aux --sort=-%mem | awk 'NR<=10 {print $0}' | head -n 10  
    echo ""  
}
```

```
# Function to display network monitoring details
```

```
network_monitoring() {  
    echo "Network Monitoring:"  
    echo "-----"  
    netstat -an | grep ':80\|:443' | wc -l # Number of concurrent connections on ports 80 and 443  
    echo "Packet Drops: $(netstat -i | awk '/^eth0/ {print $6}')"  
    echo "Network In/Out (MB):"  
    ifstat -i eth0 1 1 | awk 'NR==3 {print $1 " MB In, " $2 " MB Out"}'  
    echo ""  
}
```

```
# Function to display disk usage
```

```
disk_usage() {  
    echo "Disk Usage:"
```

```
    echo "-----"
    df -h
    df -h | awk '$5 > 80 {print $0}'
    echo ""
}
```

Function to display system load

```
system_load() {
    echo "System Load:"
    echo "-----"
    uptime
    echo "CPU Usage:"
    top -bn1 | grep "Cpu(s)"
    echo ""
}
```

Function to display memory usage

```
memory_usage() {
    echo "Memory Usage:"
    echo "-----"
    free -h
    echo ""
}
```

Function to display process monitoring

```
process_monitoring() {
    echo "Process Monitoring:"
    echo "-----"
    echo "Number of Active Processes: $(ps aux | wc -l)"
    echo "Top 5 Processes by CPU Usage:"
    ps aux --sort=-%cpu | awk 'NR<=6 {print $0}'
}
```

```

    echo "Top 5 Processes by Memory Usage:"
    ps aux --sort=-%mem | awk 'NR<=6 {print $0}'
    echo ""
}

# Function to display service monitoring
service_monitoring() {
    echo "Service Monitoring:"
    echo "-----"

    for service in sshd nginx apache2 iptables; do
        echo "$service status: $(systemctl is-active $service)"
    done
    echo ""
}

# Parse command-line arguments
while getopts "cpu memory network disk load processes services" opt; do
    case $opt in
        cpu) top_apps ;;
        memory) memory_usage ;;
        network) network_monitoring ;;
        disk) disk_usage ;;
        load) system_load ;;
        processes) process_monitoring ;;
        services) service_monitoring ;;
        \?) echo "Usage: $0 [-cpu] [-memory] [-network] [-disk] [-load] [-processes] [-services]" ;;
    esac
done

# If no options are provided, display the full dashboard
if [ $OPTIND -eq 1 ]; then

```

top_apps

network_monitoring

disk_usage

system_load

memory_usage

process_monitoring

service_monitoring

fi