

# Linux - Server Load Monitor Script

This is to monitor various objects on a given server and email notifications out when thresholds are exceeded

IT will send out something like this

```
High CPU usage on SFL-WEB-001.onling.com - [ 76% ]

R root <root@onling.com>
To ● Steve Ling

top - 11:00:01 up 2 days, 20:07, 1 user, load average: 0.05, 0.08, 0.08
Tasks: 253 total, 2 running, 251 sleeping, 0 stopped, 0 zombie
%Cpu(s): 62.5 us, 9.4 sy, 0.0 ni, 28.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7684.1 total, 823.3 free, 4797.5 used, 2513.4 buff/cache
MiB Swap: 8064.0 total, 7215.9 free, 848.1 used. 2886.6 avail Mem

  PID USER   PR NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 67745 root    20  0   3372  1664  1152 S  75.0   0.0   0:00.21 gzip
   1029 mysql   20  0 5930736 836124 9856 S  31.2  10.6  55:21.38 mariadb
 67744 root    20  0   14188  6912  5760 R  25.0   0.1   0:00.08 mysqldu+
 67764 root    20  0    8144  4096  3328 R   6.2   0.1   0:00.01 top
     1 root    20  0 106664  7808  4992 S   0.0   0.1   0:40.00 systemd
```

```
#!/bin/bash

#####

#

# Author: Steve Ling 5/2/25

#

# Purpose: Monitor system metrics including:

# - Average Load

# - CPU Usage

# - Disk Space

# - Memory Usage

#

# Installation:

# 1. Copy to /opt/scripts/
```

```

# 2. Set permissions: chmod 755 /opt/scripts/loadmon.sh
# 3. Set ownership: chown remuser:kiwiplan /opt/scripts/loadmon.sh
# 4. Add cron job as root to run every 5 minutes:
#    */5 * * * * /opt/scripts/loadmon.sh >/dev/null 2>&1 | logger
#
# Improvements:
# - Added error handling for commands and email sending
# - Improved logging with timestamps
# - Made thresholds configurable via environment variables
# - Added hostname to alerts for clarity
# - Replaced mutt with mail (more common)
# - Optimized command execution
# - Added input validation
#
#####

# Exit on any error
set -e

# Configuration (can be overridden via environment variables)
: "${LOAD_THRESHOLD:=10.00}"    # Load average threshold
: "${DISK_THRESHOLD:=85}"      # Disk usage threshold (%)
: "${CPU_THRESHOLD:=65}"       # CPU usage threshold (%)
: "${MEM_THRESHOLD:=85}"       # Memory usage threshold (%)
: "${RECIPIENTS:=steve.ling@sflservicesllc.com}" # Space-separated email addresses
: "${HOSTNAME:=$(hostname -s)}" # Short hostname for alerts
: "${LOG_FILE:=/var/log/loadmon.log}" # Log file location

# Ensure required commands are available
for cmd in awk df top free mail logger; do
    if ! command -v "$cmd" &>/dev/null; then
        echo "Error: Required command '$cmd' not found" | logger -t loadmon
        exit 1
    fi
done

# Function to log messages with timestamp
log_message() {
    local message="$1"
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $message" | logger -t loadmon
}

```

```

    echo "$(date '+%Y-%m-%d %H:%M:%S') - $message" >> "$LOG_FILE"
}

# Function to send email alerts
send_alert() {
    local subject="$1"
    local body="$2"
    if echo -e "$body" | mail -s "$subject" $RECIPIENTS 2>/dev/null; then
        log_message "Alert sent: $subject"
    else
        log_message "Error: Failed to send alert: $subject"
    fi
}

# Collect system metrics
load=$(awk '{print $1}' /proc/loadavg 2>/dev/null || log_message "Error: Failed to read load average")
disk_usage=$(df / | awk 'NR==2 {print $5}' | sed 's/%//' 2>/dev/null || log_message "Error: Failed to read disk
usage")
cpu_usage=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8}' | cut -d. -f1 2>/dev/null || log_message "Error:
Failed to read CPU usage")
mem_usage=$(free | awk '/Mem:/ {printf "%.0f", $3/$2 * 100}' 2>/dev/null || log_message "Error: Failed to read
memory usage")

# Validate collected metrics
if ! [[ "$load" =~ ^[0-9]+([.][0-9]+)?$ ]]; then
    log_message "Error: Invalid load value: $load"
    exit 1
fi
if ! [[ "$disk_usage" =~ ^[0-9]+$ ]]; then
    log_message "Error: Invalid disk usage value: $disk_usage"
    exit 1
fi
if ! [[ "$cpu_usage" =~ ^[0-9]+$ ]]; then
    log_message "Error: Invalid CPU usage value: $cpu_usage"
    exit 1
fi
if ! [[ "$mem_usage" =~ ^[0-9]+$ ]]; then
    log_message "Error: Invalid memory usage value: $mem_usage"
    exit 1
fi

```

```
# Check thresholds and send alerts
if (( $(echo "$load > $LOAD_THRESHOLD" | bc -l) )); then
    body=$(sar -q 2>/dev/null || echo "Error collecting sar data")
    send_alert "High load on $HOSTNAME - [ $load ]" "$body"
fi

if (( disk_usage > DISK_THRESHOLD )); then
    body=$(df -h / 2>/dev/null || echo "Error collecting df data")
    send_alert "High disk usage on $HOSTNAME - [ ${disk_usage}% ]" "$body"
fi

if (( cpu_usage > CPU_THRESHOLD )); then
    body=$(top -bn1 | head -n 12 2>/dev/null || echo "Error collecting top data")
    send_alert "High CPU usage on $HOSTNAME - [ ${cpu_usage}% ]" "$body"
fi

if (( mem_usage > MEM_THRESHOLD )); then
    body=$(free -h 2>/dev/null || echo "Error collecting free data")
    send_alert "High memory usage on $HOSTNAME - [ ${mem_usage}% ]" "$body"
fi

log_message "Monitoring completed: Load=$load, Disk=${disk_usage}%, CPU=${cpu_usage}%,
Mem=${mem_usage}%"
```

.

---

Revision #3

Created 14 August 2025 14:29:34 by Steve Ling

Updated 14 August 2025 16:03:40 by Steve Ling