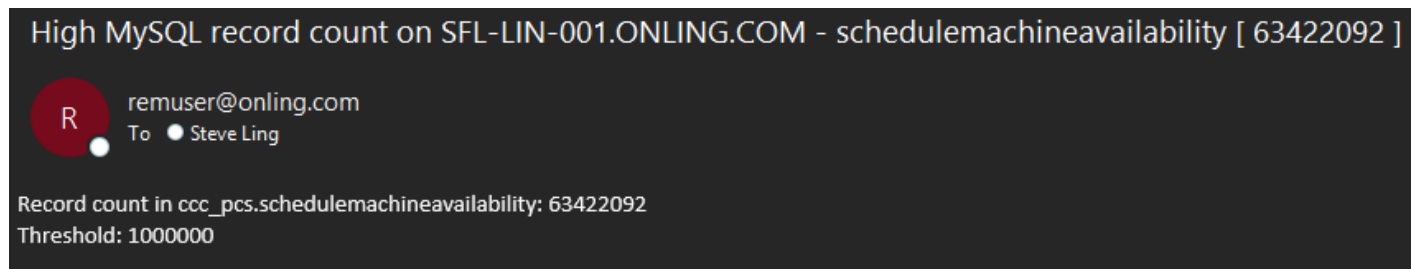
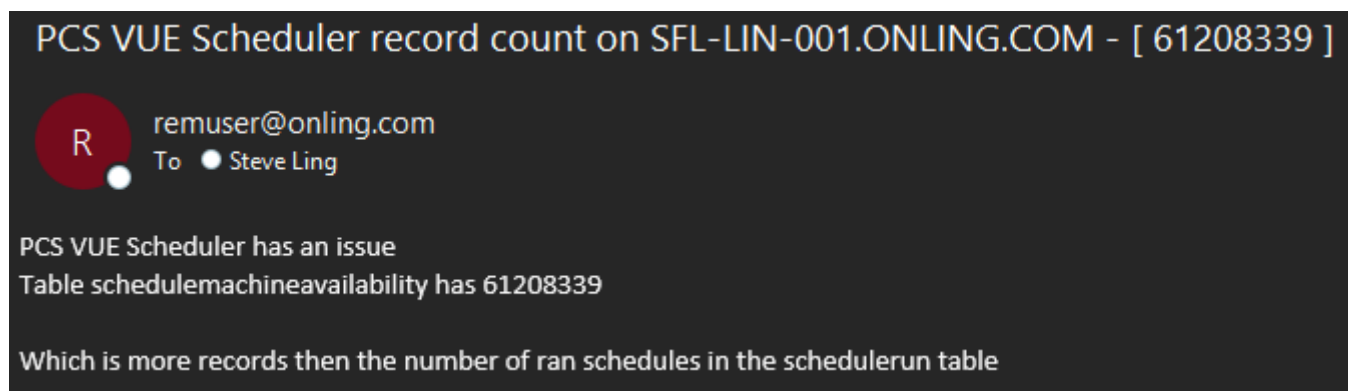


Kiwiplan - Server Load Monitor

This script is based on this initial [loadmon](#) script

This is to monitor specific objects at a Advantive/Kiwiplan sites

It will send something like this for PCS



```
#!/bin/bash

#####

#

# Author: Steve Ling 5/2/25

# Modified:

#   Added PCS Scheduler looping monitoring 8/14/25

#   Added PCS table record counts 8/14/25

#

# Purpose: Monitor system metrics including:

# - Average Load

# - CPU Usage
```

```
# - Disk Space
# - Memory Usage
# - MySQL Table Record Counts (multiple tables with individual thresholds)
#
# 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:
# To supply parameters or not:
#    */5 * * * * /opt/scripts/loadmon.sh >/dev/null 2>&1 | logger
#    */5 * * * * root MYSQL_USER="your_user" MYSQL_PASSWORD="your_password"
MYSQL_DATABASE="your_db" MYSQL_TABLES_AND_THRESHOLDS="users:50000,orders:100000"
/opt/scripts/loadmon.sh >/dev/null 2>&1 | logger
#    */5 * * * * root MYSQL_USER="your_user" MYSQL_PASSWORD="your_password" /opt/scripts/loadmon.sh
>/dev/null 2>&1 | logger
#
# MySQL Configuration:
# Set the following environment variables:
# - MYSQL_HOST: MySQL host (default: localhost)
# - MYSQL_USER: MySQL username
# - MYSQL_PASSWORD: MySQL password
# - MYSQL_PCS_DATABASE: PCS Database name
# - MYSQL_TABLES_AND_THRESHOLDS: Comma-separated list of table:threshold pairs
# (e.g., "table1:50000,table2:100000,table3"). If threshold is omitted, uses MYSQL_RECORD_THRESHOLD.
# - MYSQL_PCS_RECORD_THRESHOLD: Default record count threshold
#
# Improvements:
# - Added support for multiple MySQL tables with individual thresholds
# - Sends individual alerts for each table exceeding its threshold
# - Added error handling for MySQL commands
# - 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
#
#####
```

```

# Uncomment to "Exit" the script 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 (%)
: "${MYSQL_RECORD_THRESHOLD:=100000}" # Default MySQL table record count threshold
: "${MYSQL_PCS_RECORD_THRESHOLD:=100000}" # Default PCS Scheduler table record count 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
: "${MYSQL_HOST:=localhost}"      # MySQL host
: "${MYSQL_USER:=}"               # MySQL username
: "${MYSQL_PASSWORD:=}"           # MySQL password
: "${MYSQL_PCS_DATABASE:=ccc_pcs}" # MySQL database prefix
: "${MYSQL_TABLES_AND_THRESHOLDS:=schedulemachineavailability:1000000}" # Comma-separated
table:threshold pairs

# Ensure required commands are available
for cmd in awk df top free mail logger mysql sed tr bc; 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 system 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

# Collect MySQL table record counts (if configured)
mysql_records_summary=""
if [ -n "$MYSQL_USER" ] && [ -n "$MYSQL_PCS_DATABASE" ] && [ -n "$MYSQL_TABLES_AND_THRESHOLDS" ];
then
    IFS=' ' read -r -a table_pairs <<< "$MYSQL_TABLES_AND_THRESHOLDS"
    for pair in "${table_pairs[@]"; do
        pair=$(echo "$pair" | sed 's/^[[:space:]]*//;s/[[:space:]]*$//') # Trim whitespace

```

```

table="${pair%%:*}" # Extract table name
threshold="${pair#*:}" # Extract threshold (if provided)

# Use default threshold if not specified or invalid
if [ -z "$threshold" ] || ! [[ "$threshold" =~ ^[0-9]+$ ]]; then
    threshold="$MYSQL_PCS_RECORD_THRESHOLD"
    log_message "Warning: Using default threshold ($threshold) for table $table"
fi

# Validate table name (basic check for non-empty and no special characters)
if [ -z "$table" ] || ! [[ "$table" =~ [^a-zA-Z0-9_] ]]; then
    log_message "Error: Invalid table name: $table"
    continue
fi

# Query record count
record_count=$(mysql -h "$MYSQL_HOST" -u "$MYSQL_USER" -p"$MYSQL_PASSWORD" -D
"$MYSQL_PCS_DATABASE" -N -e "SELECT COUNT(*) FROM ` $table `" 2>/dev/null) || {
    log_message "Error: Failed to query MySQL table $MYSQL_PCS_DATABASE.$table"
    record_count=0
}

# Validate record count
if ! [[ "$record_count" =~ ^[0-9]+$ ]]; then
    log_message "Error: Invalid record count for table $table: $record_count"
    record_count=0
fi

# Append to summary
mysql_records_summary="$mysql_records_summary $table:$record_count"

# Check threshold and send alert if exceeded
if [ "$record_count" -gt "$threshold" ]; then
    body="Record count in $MYSQL_PCS_DATABASE.$table: $record_count\nThreshold: $threshold"
    send_alert "High MySQL record count on $HOSTNAME - $table [ $record_count ]" "$body"
fi
done
mysql_records_summary=$(echo "$mysql_records_summary" | sed 's/^ //') # Trim leading space
else
    log_message "Warning: MySQL configuration incomplete (USER, DATABASE, or TABLES_AND_THRESHOLDS

```

```

missing)"
    mysql_records_summary="N/A"
fi

# Collect PCS Scheduler record count
mysql_pcs_record_count=0
if [ -n "$MYSQL_USER" ] && [ -n "$MYSQL_PCS_DATABASE" ]; then
    mysql_pcs_record_count=$(mysql -h "$MYSQL_HOST" -u "$MYSQL_USER" -p"$MYSQL_PASSWORD" -D
"$MYSQL_PCS_DATABASE" -N -e "SELECT COUNT(*) FROM schedulemachineavailability WHERE pcsSchedule NOT
IN (SELECT pcsSchedule FROM schedulerun)" 2>/dev/null) || {
    log_message "Error: Failed to query PCS Scheduler table $MYSQL_PCS_DATABASE"
    mysql_pcs_record_count=0
}
else
    log_message "Warning: PCS Scheduler configuration incomplete (USER or DATABASE missing)"
fi

# Validate collected system metric
if ! [[ "$mysql_pcs_record_count" =~ ^[0-9]+$ ]]; then
    log_message "Error: Invalid PCS Scheduler record count: $mysql_pcs_record_count"
    mysql_pcs_record_count=0
fi

# Check thresholds and send alerts for system metrics
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

if [ "$mysql_pcs_record_count" -gt "$MYSQL_PCS_RECORD_THRESHOLD" ]; then
    body="PCS VUE Scheduler has an issue\nTable schedulemachineavailability has
    $mysql_pcs_record_count\n\nWhich is more records then the number of ran schedules in the schedulerun table"
    send_alert "PCS VUE Scheduler record count on $HOSTNAME - [ $mysql_pcs_record_count ]" "$body"
fi

log_message "Monitoring completed: Load=$load, Disk=${disk_usage}%, CPU=${cpu_usage}%,
Mem=${mem_usage}%, PCS Scheduler Count=${mysql_pcs_record_count}, MySQL
Records=[$mysql_records_summary]"
```

Revision #14

Created 14 August 2025 14:28:49 by Steve Ling

Updated 15 August 2025 01:13:57 by Steve Ling