

Linux Shell Script for VM Health Monitoring

Linux Shell Script for VM Health Monitoring

I've created two shell scripts for you to monitor the VoIP service and automatically reboot your Proxmox VM when needed.

Simple Version.

Here's the basic script that does exactly what you asked for:

```
#!/bin/bash

# Simple VM Health Monitor
# Checks if http://www.your-domain.com responds, reboots VM 105 if not

URL="http://www.your-domain.com"
VM_ID="105"

# Check if URL responds (with 10 second timeout)
if ! curl -s --connect-timeout 10 --max-time 10 "$URL" > /dev/null 2>&1; then
    echo "$(date): Service not responding, rebooting VM $VM_ID"
    qm reboot "$VM_ID" && sleep 30 && qm start "$VM_ID"
    echo "$(date): VM reboot completed"
else
    echo "$(date): Service is responding normally"
fi
```

Enhanced Version (Production Ready)

```
#!/bin/bash

# VM Health Monitor Script
# This script checks if a web service is responding and reboots the VM if it's not
```

```
# Configuration
TARGET_URL="http://www.your-domain.com"
VM_ID="105"
TIMEOUT=10
MAX_RETRIES=3
LOG_FILE="/var/log/vm-health-monitor.log"

# Function to log messages with timestamp
log_message() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $1" | tee -a "$LOG_FILE"
}

# Function to check if URL is responding
check_url() {
    local url="$1"
    local timeout="$2"

    # Use curl to check if the URL responds
    if curl -s --connect-timeout "$timeout" --max-time "$timeout" "$url" > /dev/null 2>&1;
then
        return 0 # Success
    else
        return 1 # Failure
    fi
}

# Function to reboot VM
reboot_vm() {
    local vm_id="$1"

    log_message "Attempting to reboot VM $vm_id"

    # Stop the VM first
    if qm stop "$vm_id"; then
        log_message "VM $vm_id stopped successfully"
        sleep 30

        # Start the VM
        if qm start "$vm_id"; then
            log_message "VM $vm_id started successfully"
        fi
    fi
}
```

```

        return 0
    else
        log_message "ERROR: Failed to start VM $vm_id"
        return 1
    fi
else
    log_message "ERROR: Failed to stop VM $vm_id"
    return 1
fi
}

# Main monitoring function
main() {
    log_message "Starting health check for $TARGET_URL"

    local retry_count=0

    # Retry loop
    while [ $retry_count -lt $MAX_RETRIES ]; do
        if check_url "$TARGET_URL" "$TIMEOUT"; then
            log_message "Service is responding normally (attempt $((retry_count + 1))/$MAX_RETRIES)"
            exit 0
        else
            retry_count=$((retry_count + 1))
            log_message "Service not responding (attempt $retry_count/$MAX_RETRIES)"

            if [ $retry_count -lt $MAX_RETRIES ]; then
                sleep 5
            fi
        fi
    done

    # If we reach here, all retries failed
    log_message "Service failed to respond after $MAX_RETRIES attempts. Initiating VM reboot."

    if reboot_vm "$VM_ID"; then
        log_message "VM reboot completed successfully"
    else
        log_message "VM reboot failed - manual intervention required"
    fi
}

```

```
        exit 1
    fi
}

# Check if running as root (required for qm commands)
if [ "$EUID" -ne 0 ]; then
    echo "This script must be run as root (required for Proxmox qm commands)"
    exit 1
fi

# Create log directory if it doesn't exist
mkdir -p "$(dirname "$LOG_FILE")"

# Run main function
main

exit 0
```

Installation and Usage

1. Save the Script

Save either script to a file (e.g., vm-health-monitor.sh) on your Proxmox host:

```
nano vm-health-monitor.sh
```

Paste the script content

2. Make it Executable

```
chmod +x vm-health-monitor.sh
```

3. Test the Script

```
sudo ./vm-health-monitor.sh
```

4. Set up Automatic Monitoring (Cron Job)

To run the script automatically every 5 minutes:

```
sudo crontab -e
```

Add this line:

```
*/5 * * * * /path/to/vm-health-monitor.sh >> /var/log/vm-monitor.log 2>&1
```

Key Features

Simple Script:

- Direct implementation of your requirements
- Uses curl to check if the URL responds
- Executes qm reboot 105 && sleep 30 && qm start 105 if the service is down
- Minimal logging with timestamps

Enhanced Script:

- Retry Logic: Attempts 3 times before declaring service down
- Comprehensive Logging: Detailed logs with timestamps
- Error Handling: Proper error checking for VM operations
- Root Check: Ensures script runs with proper privileges
- Graceful Shutdown: Uses qm stop before qm start for cleaner reboot

Important Notes

- Root Privileges Required: The qm command requires root access, so run the script as root or with sudo
- Curl Dependency: Ensure curl is installed on your Proxmox host
- Network Timeout: Set to 10 seconds to avoid hanging on network issues
- VM ID: Currently set to 105 as specified - modify if needed
- Logging: Enhanced version logs to /var/log/vm-health-monitor.log

Both scripts will effectively monitor your VoIP service and automatically reboot VM 105 when the service becomes unresponsive, ensuring minimal downtime for your infrastructure.

Download the scripts.

Simple version

Attachment link : [vm_health_monitoring_simple.sh](#)

Enhanced version

Attachment link : [vm_health_monitoring_enhanced.sh](#)

Ins0mniA

Revision #11

Created 2025-08-03 17:33:38 EEST by Green

Updated 2025-09-04 02:06:51 EEST by Green