#!/bin/bash

# **Script: comprehensive\_server\_mgmt.sh**

# **Description: A comprehensive script for server health monitoring, backups, and log analysis.**

# **Author: Your Name**

# **Date: January 4, 2024**

# **Function to check disk space in /home directory**

check\_disk\_space() {  
echo “Disk space in /home directory on $1:”  
ssh $1 ‘df -h /home’  
}

# **Function to check running processes**

check\_running\_processes() {  
echo -e “\nRunning processes on $1:”  
ssh $1 ‘ps aux’  
}

# **Function to check memory usage**

check\_memory\_usage() {  
echo -e “\nMemory usage on $1:”  
ssh $1 ‘free -m’  
}

# **Function to check top CPU-consuming processes**

check\_top\_cpu() {  
echo -e “\nTop CPU-consuming processes on $1:”  
ssh $1 ‘ps aux --sort=-%cpu | head -n 5’  
}

# **Function to check top memory-consuming processes**

check\_top\_memory() {  
echo -e “\nTop memory-consuming processes on $1:”  
ssh $1 ‘ps aux --sort=-%mem | head -n 5’  
}

# **Perform comprehensive health checks on the local machine**

check\_disk\_space localhost  
check\_running\_processes localhost  
check\_memory\_usage localhost  
check\_top\_cpu localhost  
check\_top\_memory localhost

# **Perform comprehensive health checks on a remote server “hideho”**

check\_disk\_space hideho  
check\_running\_processes hideho  
check\_memory\_usage hideho  
check\_top\_cpu hideho  
check\_top\_memory hideho

# **------------------------------------------------------**

#!/bin/bash

# **Script: server\_alerts.sh**

# **Description: Monitors CPU, memory, and disk usage, sending alerts if thresholds are exceeded.**

# **Author: Your Name**

# **Date: January 4, 2024**

# **Set thresholds**

CPU\_THRESHOLD=90  
MEMORY\_THRESHOLD=90  
DISK\_THRESHOLD=90

# **Get system metrics**

CPU=$(top -b -n 1 | awk ‘/%Cpu/ {print $2}’ | cut -d. -f1)  
MEMORY=$(free | awk ‘/Mem/ {printf("%.0f", $3/$2\*100)}’)  
DISK=$(df -h / | awk ‘/// {print $(NF-1)}’ | cut -d% -f1)

# **Check and send alerts**

if [ $CPU -gt $CPU\_THRESHOLD ]; then  
echo “High CPU usage: $CPU%” | mail -s “Server Alert” admin@example.com  
fi

if [ $MEMORY -gt $MEMORY\_THRESHOLD ]; then  
echo “High memory usage: $MEMORY%” | mail -s “Server Alert” admin@example.com  
fi

if [ $DISK -gt $DISK\_THRESHOLD ]; then  
echo “High disk usage: $DISK%” | mail -s “Server Alert” admin@example.com  
fi

# 

# **------------------------------------------------------**

#!/bin/bash

# **Script: backup\_script.sh**

# **Description: Creates a compressed backup of important directories and performs cleanup.**

# **Author: Your Name**

# **Date: January 4, 2024**

# **Set backup directory and cleanup threshold**

BACKUP\_DIR="/path/to/backup"  
CLEANUP\_THRESHOLD=7 # Number of days to keep backups

# **Create backup**

tar -czf $BACKUP\_DIR/backup\_$(date +%Y%m%d).tar.gz /important/directory

# **Cleanup old backups**

find $BACKUP\_DIR -type f -name “backup\_\*” -mtime +$CLEANUP\_THRESHOLD -exec rm {} ;

**------------------------------------------------------**

#!/bin/bash

# **Script: log\_analysis.sh**

# **Description: Analyzes log files for specific patterns (e.g., errors) and sends notifications.**

# **Author: Your Name**

# **Date: January 4, 2024**

# **Set log file and error pattern**

LOG\_FILE="/var/log/application.log"  
ERROR\_PATTERN=“ERROR”

# **Check for errors in the log file**

if grep -q “$ERROR\_PATTERN” $LOG\_FILE; then  
echo “Error found in log file: $ERROR\_PATTERN” | mail -s “Log Analysis Alert” admin@example.com  
fi