DevOps Bash script to check Server Health

Server Health Check Script

This script provides a quick overview of the server's health by displaying important system metrics like hostname, uptime, CPU usage, memory usage, disk space, and top processes.

Line-by-Line Breakdown:

1. Shebang

#!/bin/bash

This tells the system to use the Bash shell to execute the script.

2. Print header

echo "== Server Health Check =="

Displays a title to make the output clear

3. Display Hostname

echo "Hostname: $(hostname)"

hostname command prints the server's name

$(command) runs a command inside echo

4. Show system Uptime

echo "Uptime: $(uptime -p)"

uptime -p shows how long the system has been running in human-readable format

5. Show CPU Usage

echo "--- CPU Usage ---"

top -bn1 | grep "Cpu(s)"

top -bn1 runs the top command in batch mode once (**-b** = batch mode, **-n1** = 1 iternation)

grep "Cpu(s)" extracts the CPU usage line

Notes: this helps monitor **CPU load**

6. Show Memory Usage

echo “--- Memory Usage ---”

free -h

free -h shows available and used memory (**-h** makes it human-readable)

7. Show Disk Usage

echo “--- Disk Usage ---”

df -h /

df -h / shows disk space usage for the root directory (**/**) in human-readable format

8. Show Top 5 Memory-consuming processes

echo “--- Top 5 Processes ---”

ps -eo pid, ppid, cmd, %mem, %cpu –sort=-%mem | head -6

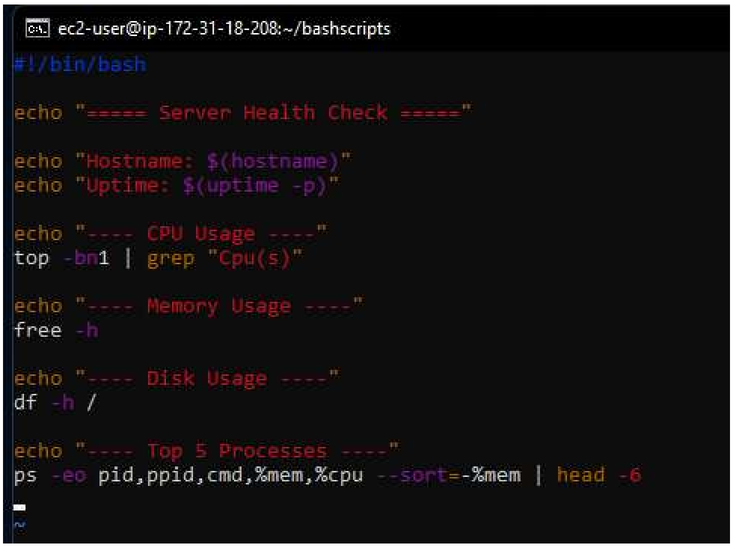
ps -eo pid, ppid, cmd, %mem, %cpu:

- **pid** – process ID
- **ppid** – parent process ID
- **cmd** – command running the process
- **%mem** – memory usage
- **%cpu** – CPU usage

–sort=-%mem – sorts by highest memory usage first

Head -6 – shows top 5 processes (**head -6** instead of **head -5** because the first row is a header)

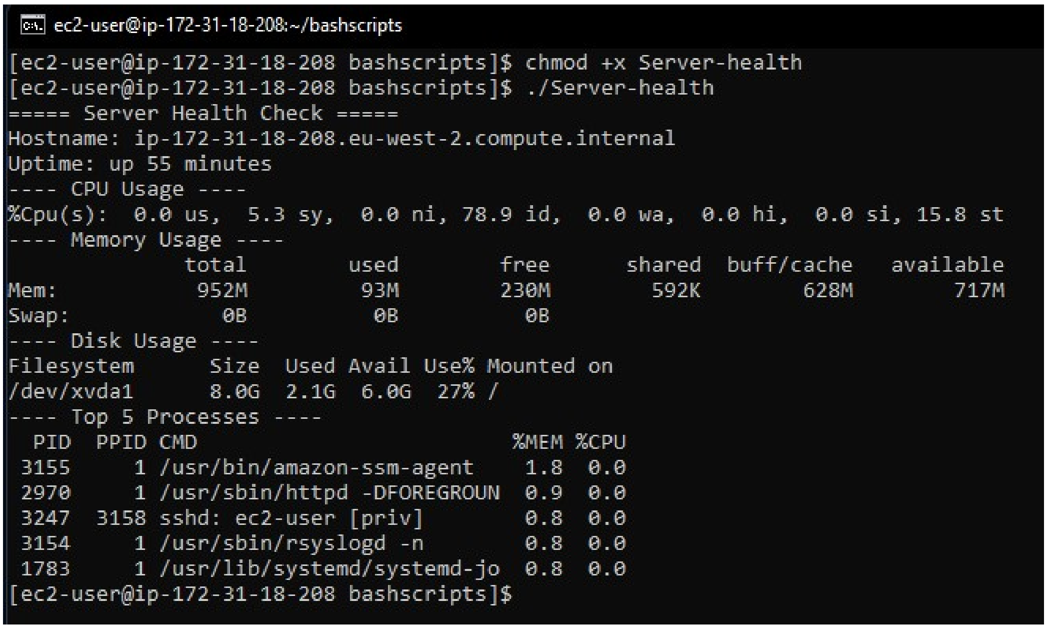
Example Input via vi:



```
#!/bin/bash

echo "===== Server Health Check ====="

echo "Hostname: $(hostname)"
echo "Uptime: $(uptime -p)"

echo "---- CPU Usage ----"
top -bn1 | grep "Cpu(s)"

echo "---- Memory Usage ----"
free -h

echo "---- Disk Usage ----"
df -h /

echo "---- Top 5 Processes ----"
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head -6

~
```

Make sure to change file permissions to execute the bash script.

Example output in Linux:



```
[ec2-user@ip-172-31-18-208 bashscripts]$ chmod +x Server-health
[ec2-user@ip-172-31-18-208 bashscripts]$ ./Server-health
===== Server Health Check =====
Hostname: ip-172-31-18-208.eu-west-2.compute.internal
Uptime: up 55 minutes
---- CPU Usage ----
%Cpu(s):  0.0 us,  5.3 sy,  0.0 ni, 78.9 id,  0.0 wa,  0.0 hi,  0.0 si, 15.8 st
---- Memory Usage ----
              total        used        free      shared  buff/cache   available
Mem:           952M         93M        230M        592K        628M        717M
Swap:            0B          0B          0B
---- Disk Usage ----
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      8.0G  2.1G  6.0G  27% /
---- Top 5 Processes ----
  PID  PPID CMD                         %MEM %CPU
 3155     1 /usr/bin/amazon-ssm-agent    1.8  0.0
 2970     1 /usr/sbin/httpd -DFOREGROUN  0.9  0.0
 3247  3158 sshd: ec2-user [priv]        0.8  0.0
 3154     1 /usr/sbin/rsyslogd -n        0.8  0.0
 1783     1 /usr/lib/systemd/systemd-jo  0.8  0.0
[ec2-user@ip-172-31-18-208 bashscripts]$
```