

→ Convenciones:

```
# En todos los nodos como 'sudo su'.  
[root@srv1 ~]# Solo en servidor 'srv1' → como 'sudo su'.  
[root@srv2 ~]# Solo en servidor 'srv2' → como 'sudo su'.
```

364.1 Hardware and Resource High Availability (weight: 2)

| | |
|--------------------|---|
| Weight | 2 |
| Description | Candidates should be able to monitor a local node for potential hardware failures and resource shortages. |

Key Knowledge Areas:

- Understand and monitor S.M.A.R.T values using smartmontools, including triggering frequent disk checks
- Configure system shutdown at specific UPC events
- Configure monit for alerts in case of resource exhaustion

Partial list of the used files, terms and utilities:

- smartctl
- /etc/smartd.conf
- smartd
- nvme-cli
- apcupsd
- apctest
- monit

```
redhat00 ~ # dnf install smartmontools
```

```
redhat00 ~ # systemctl status smartd.service
```

- smartd.service - Self Monitoring and Reporting Technology (SMART) Daemon

Loaded: loaded (/usr/lib/systemd/system/smartd.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2021-03-26 03:13:45 CET; 2 days ago

Docs: man:smartd(8)

man:smartd.conf(5)

Main PID: 1095 (smartd)

Status: "Next check of 4 devices will start at 11:13:44"

Tasks: 1 (limit: 126461)

Memory: 4.4M

CGroup: /system.slice/smartd.service

└─1095 /usr/sbin/smartd -n -q never

mar 27 22:43:49 redhat00.cadilinea.lan smartd[1095]: Device: /dev/sdb [SAT], CHECK POWER STATUS spins up disk (>

mar 27 23:13:49 redhat00.cadilinea.lan smartd[1095]:

...

redhat00 ~ # smartctl --

```
--abort      --device=      --log=          --saveauto=     --vendorattribute=
--all         --drivedb=      --nocheck=      --scan           --version
--attributes  --firmwarebug=  --offlineauto=  --scan-open     --xall
--badsum=     --health        --presets=      --smart=
--capabilities --help          --quietmode=    --test=
--captive     --info          --report=       --tolerance=
```

redhat00 ~ # smartctl --scan

/dev/sda -d scsi # /dev/sda, SCSI device

/dev/sdb -d scsi # /dev/sdb, SCSI device

/dev/sdc -d scsi # /dev/sdc, SCSI device

/dev/sdd -d scsi # /dev/sdd, SCSI device

/dev/sde -d sat # /dev/sde [SAT], ATA device

redhat00 ~ # smartctl -a /dev/sde

smartctl 7.1 2020-04-05 r5049 [x86_64-linux-4.18.0-240.15.1.el8_3.x86_64] (local build)

Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===

Model Family: Seagate Samsung SpinPoint M8 (AF)

Device Model: ST1000LM024 HN-M101MBB

Serial Number: S2RXJ9CCC01244

LU WWN Device Id: 5 0004cf 20916e757

Firmware Version: 2AR10002

User Capacity: 1.000.204.886.016 bytes [1,00 TB]
Sector Sizes: 512 bytes logical, 4096 bytes physical
Rotation Rate: 5400 rpm
Form Factor: 2.5 inches
Device is: In smartctl database [for details use: -P show]
ATA Version is: ATA8-ACS T13/1699-D revision 6
SATA Version is: SATA 3.0, 3.0 Gb/s (current: 3.0 Gb/s)
Local Time is: Sun Mar 28 10:20:19 2021 CEST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
=== START OF READ SMART DATA SECTION ===
SMART Status not supported: Incomplete response, ATA output registers missing
SMART overall-health self-assessment test result: PASSED
Warning: This result is based on an Attribute check.
General SMART Values:
Offline data collection status: (0x00) Offline data collection activity
was never started.
Auto Offline Data Collection: Disabled.
Self-test execution status: (0) The previous self-test routine completed
without error or no self-test has ever
been run.
Total time to complete Offline
data collection: (13380) seconds.
Offline data collection
capabilities: (0x5b) SMART execute Offline immediate.
Auto Offline data collection on/off support.
Suspend Offline collection upon new
command.

Offline surface scan supported.

Self-test supported.

No Conveyance Self-test supported.

Selective Self-test supported.

SMART capabilities: (0x0003) Saves SMART data before entering power-saving mode.

Supports SMART auto save timer.

Error logging capability: (0x01) Error logging supported.

General Purpose Logging supported.

Short self-test routine

recommended polling time: (2) minutes.

Extended self-test routine

recommended polling time: (223) minutes.

SCT capabilities: (0x003f) SCT Status supported.

SCT Error Recovery Control supported.

SCT Feature Control supported.

SCT Data Table supported.

SMART Attributes Data Structure revision number: 16

Vendor Specific SMART Attributes with Thresholds:

| ID# | ATTRIBUTE_NAME | FLAG | VALUE | WORST | THRESH | TYPE | UPDATED |
|-------------|------------------------|--------|-------|-------|--------|-----------------|---------|
| WHEN_FAILED | RAW_VALUE | | | | | | |
| 1 | Raw_Read_Error_Rate | 0x002f | 100 | 100 | 051 | Pre-fail Always | - 0 |
| 2 | Throughput_Performance | 0x0026 | 252 | 252 | 000 | Old_age Always | - 0 |
| 3 | Spin_Up_Time | 0x0023 | 086 | 086 | 025 | Pre-fail Always | - 4466 |
| 4 | Start_Stop_Count | 0x0032 | 086 | 086 | 000 | Old_age Always | - 14653 |
| 5 | Reallocated_Sector_Ct | 0x0033 | 252 | 252 | 010 | Pre-fail Always | - 0 |
| 7 | Seek_Error_Rate | 0x002e | 252 | 252 | 051 | Old_age Always | - 0 |
| 8 | Seek_Time_Performance | 0x0024 | 252 | 252 | 015 | Old_age Offline | - 0 |

| | | | | | | | | | |
|------------|-------------------------|---------------|------------|------------|------------|----------------|---------------|----------|-----------------------|
| 9 | Power_On_Hours | 0x0032 | 100 | 100 | 000 | Old_age | Always | - | 1645 |
| 10 | Spin_Retry_Count | 0x0032 | 252 | 252 | 051 | Old_age | Always | - | 0 |
| 11 | Calibration_Retry_Count | 0x0032 | 100 | 100 | 000 | Old_age | Always | - | 1 |
| 12 | Power_Cycle_Count | 0x0032 | 100 | 100 | 000 | Old_age | Always | - | 263 |
| 191 | G-Sense_Error_Rate | 0x0022 | 252 | 252 | 000 | Old_age | Always | - | 0 |
| 192 | Power-Off_Retract_Count | 0x0022 | 252 | 252 | 000 | Old_age | Always | - | 0 |
| 194 | Temperature_Celsius | 0x0002 | 064 | 064 | 000 | Old_age | Always | - | 22 (Min/Max 18/47) |
| 195 | Hardware_ECC_Recovered | 0x003a | 100 | 100 | 000 | Old_age | Always | - | 0 |
| 196 | Reallocated_Event_Count | 0x0032 | 252 | 252 | 000 | Old_age | Always | - | 0 |
| 197 | Current_Pending_Sector | 0x0032 | 252 | 252 | 000 | Old_age | Always | - | 0 |
| 198 | Offline_Uncorrectable | 0x0030 | 252 | 252 | 000 | Old_age | Offline | - | 0 |
| 199 | UDMA_CRC_Error_Count | 0x0036 | 200 | 200 | 000 | Old_age | Always | - | 0 |
| 200 | Multi_Zone_Error_Rate | 0x002a | 100 | 100 | 000 | Old_age | Always | - | 2 |
| 223 | Load_Retry_Count | 0x0032 | 100 | 100 | 000 | Old_age | Always | - | 1 |
| 225 | Load_Cycle_Count | 0x0032 | 097 | 097 | 000 | Old_age | Always | - | 31539 |

SMART Error Log Version: 1

No Errors Logged

SMART Self-test log structure revision number 1

No self-tests have been logged. [To run self-tests, use: smartctl -t]

SMART Selective self-test log data structure revision number 0

Note: revision number not 1 implies that no selective self-test has ever been run

SPAN MIN_LBA MAX_LBA CURRENT_TEST_STATUS

| | | | |
|---|---|---|--------------------------------|
| 1 | 0 | 0 | Completed [00% left] (0-65535) |
| 2 | 0 | 0 | Not_testing |
| 3 | 0 | 0 | Not_testing |
| 4 | 0 | 0 | Not_testing |

```
5 0 0 Not_testing
```

Selective self-test flags (0x0):

After scanning selected spans, do NOT read-scan remainder of disk.

If Selective self-test is pending on power-up, resume after 0 minute delay.

redhat00 ~ # lsblk --output NAME,SIZE,MODEL,MOUNTPOINT

```
NAME                SIZE MODEL          MOUNTPOINT
sda                 931,5G WDC WD1003FZEX-0
├─sda1              2,9G          /boot
└─sda2              925,5G
   ├─rhel-pool00_tmeta 1,8G
   │ └─rhel-pool00-tpool 3,5T
   │   ├─rhel-root      2T           /
   │   ├─rhel-pool00    3,5T
   │   └─rhel-home      1,5T         /home
   └─rhel-pool00_tdata 3,5T
       └─rhel-pool00-tpool 3,5T
           ├─rhel-root      2T           /
           ├─rhel-pool00    3,5T
           └─rhel-home      1,5T         /home
└─rhel-swap         20,8G        [SWAP]
sdb                 931,5G MB1000GCEEK
└─sdb1              927,5G
   └─rhel-pool00_tdata 3,5T
       └─rhel-pool00-tpool 3,5T
           ├─rhel-root      2T           /
           ├─rhel-pool00    3,5T
           └─rhel-home      1,5T         /home
```

sdc 931,5G WDC WD1003FZEX-0

└─sdc1 927,5G

└─rhel-pool00_tdata 3,5T

└─rhel-pool00-tpool 3,5T

└─rhel-root 2T /

└─rhel-pool00 3,5T

└─rhel-home 1,5T /home

sdd 931,5G WDC WD10EURX-73F

└─sdd1 927,5G

└─rhel-pool00_tdata 3,5T

└─rhel-pool00-tpool 3,5T

└─rhel-root 2T /

└─rhel-pool00 3,5T

└─rhel-home 1,5T /home

sde 931,5G External USB 3.0

└─sde1 931,5G

sr0 1024M DVD-RAM GH82N

→ **Discos SSD con tecnologia NVME**

redhat00 ~ # dnf install nvme-cli

redhat00 ~ # nvme list

redhat00 ~ # nvme smart-log /dev/nvme0n1

```
Terminal File Edit View Search Terminal Help
Smart Log for NVME device:nvme0n1 namespace-id:ffffff
critical_warning      : 0
temperature           : 31 C
available_spare       : 100%
available_spare_threshold : 10%
percentage_used       : 0%
data_units_read       : 426 713
data_units_written    : 417 181
host_read_commands    : 6 642 465
host_write_commands   : 3 648 785
controller_busy_time  : 28
power_cycles           : 43
power_on_hours        : 21
unsafe_shutdowns      : 14
media_errors          : 0
num_err_log_entries   : 115
Warning Temperature Time : 0
Critical Composite Temperature Time : 0
Temperature Sensor 1  : 31 C
Temperature Sensor 2  : 40 C
Temperature Sensor 3  : 0 C
Temperature Sensor 4  : 0 C
Temperature Sensor 5  : 0 C
Temperature Sensor 6  : 0 C
Temperature Sensor 7  : 0 C
Temperature Sensor 8  : 0 C
eera5607@eera5607:~$
```

Análisis de los datos

Load Cycle Count o Power Cycles:

Esta es una información muy útil ya que por lo general los fabricantes establecen dentro de las especificaciones del disco la cantidad de ciclos de carga como una forma de indicar la vida útil de un disco duro.

En este caso el disco que estoy analizando (con 4 años de uso) indica en sus especificaciones que el total de ciclos es de 600 000 y la información me indica lo siguiente:


```

Terminal File Edit View Search Terminal Help
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000b   100   100   062   Pre-fail Always         -         0
  2 Throughput_Performance  0x0005   100   100   040   Pre-fail Offline        -         0
  3 Spin_Up_Time            0x0007   171   171   033   Pre-fail Always         -         1
  4 Start_Stop_Count       0x0012   099   099   000   Old_age  Always        -       1720
  5 Reallocated_Sector_Ct   0x0033   100   100   005   Pre-fail Always         -         0
  7 Seek_Error_Rate        0x000b   100   100   067   Pre-fail Always         -         0
  8 Seek_Time_Performance  0x0005   100   100   040   Pre-fail Offline        -         0
  9 Power_On_Hours         0x0012   078   078   000   Old_age  Always        -     9893
 10 Spin_Retry_Count       0x0013   100   100   060   Pre-fail Always         -         0
 12 Power_Cycle_Count      0x0032   099   099   000   Old_age  Always        -     1697
191 G-Sense_Error_Rate     0x000a   100   100   000   Old_age  Always         -         0
192 Power-Off_Retract_Count 0x0032   100   100   000   Old_age  Always         -         35
193 Load_Cycle_Count       0x0012   094   094   000   Old_age  Always         -     69687
194 Temperature_Celsius    0x0002   240   240   000   Old_age  Always         -         25 (Min/Max 18/44)
196 Reallocated_Event_Count 0x0032   100   100   000   Old_age  Always         -         0
197 Current_Pending_Sector 0x0022   100   100   000   Old_age  Always         -         0
198 Offline_Uncorrectable   0x0008   100   100   000   Old_age  Offline        -         0
199 UDMA_CRC_Error_Count   0x000a   200   200   000   Old_age  Always         -         0
223 Load_Retry_Count      0x000a   100   100   000   Old_age  Always         -         0

SMART Error Log Version: 1
No Errors Logged
    
```

Total LBAs Written

En este caso corresponde al parámetro 241 y el número indicado representa la **cantidad de unidades de 512 bytes escritas**. Por esta razón para obtener el dato en bytes es necesario multiplicar el valor indicado en los resultados por 512 y después dividirlo entre 1073741824 para obtener los gigabytes totales escritos (hay mil setenta y tres millones setecientos cuarenta y un mil ochocientos veinticuatro bytes en un gigabyte).

```

Terminal File Edit View Search Terminal Help
SMART Attributes Data Structure revision number: 10
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
 1 Raw_Read_Error_Rate     0x000f   083   075   006   Pre-fail Always    -       218903335
 3 Spin_Up_Time            0x0003   097   097   000   Pre-fail Always    -         0
 4 Start_Stop_Count       0x0032   100   100   020   Old_age  Always    -        41
 5 Reallocated_Sector_Ct   0x0033   100   100   010   Pre-fail Always    -         0
 7 Seek_Error_Rate         0x000f   100   253   045   Pre-fail Always    -       401565
 9 Power_On_Hours          0x0032   100   100   000   Old_age  Always    -        71
10 Spin_Retry_Count        0x0013   100   100   097   Pre-fail Always    -         0
12 Power_Cycle_Count       0x0032   100   100   020   Old_age  Always    -        40
183 Runtime_Bad_Block       0x0032   100   100   000   Old_age  Always    -         0
184 End-to-End_Error        0x0032   100   100   099   Old_age  Always    -         0
187 Reported_Uncorrect      0x0032   100   100   000   Old_age  Always    -         0
188 Command_Timeout        0x0032   100   100   000   Old_age  Always    -         0
189 High_Fly_Writes         0x003a   100   100   000   Old_age  Always    -         0
190 Airflow_Temperature_Cel 0x0022   074   059   040   Old_age  Always    -        26 (Min/Max 21/26)
193 Load_Cycle_Count        0x0032   100   100   000   Old_age  Always    -        41
194 Temperature_Celsius     0x0022   026   021   000   Old_age  Always    -        26 (0 21 0 0 0)
195 Hardware_ECC_Recovered  0x001a   007   003   000   Old_age  Always    -       218903335
197 Current_Pending_Sector  0x0012   100   100   000   Old_age  Always    -         0
198 Offline_Uncorrectable    0x0010   100   100   000   Old_age  Offline   -         0
199 UDMA_CRC_Error_Count    0x003e   200   200   000   Old_age  Always    -         0
240 Head_Flying_Hours       0x0000   100   253   000   Old_age  Offline   -        70 (242 73 0)
241 Total_LBAs_Written      0x0000   100   253   000   Old_age  Offline   -       228248782
242 Total_LBAs_Read         0x0000   100   253   000   Old_age  Offline   -       248221948

```

En el caso de la imagen el resultado de $248221948 * 512 / 1073741824$ da 108 Gigabytes. Es un **disco duro relativamente nuevo** con apenas 41 ciclos de carga completados (ver parámetro 193).

Nota: es importante aclarar que para algunos fabricantes como Kingston, el resultado para el parámetro 241 se muestra directamente en *gigabytes*. En otros casos como el del fabricante Intel, el contador aumenta en una unidad cada 32 *megabytes* escritos. Por esto, hay que multiplicar el dato por 32 y dividir entre 1024 para obtener el total de *gigabytes* escritos.

Data Units Written (para NVME)

Este es un dato que aparece por lo regular en los discos de **tecnología NVME**. Es similar al caso anterior pero a diferencia la interpretación es diferente. Cada unidad de **Data Units Written** corresponde a su vez a 1000 unidades de 512 bytes escritas. En otras palabras, para obtener el dato legible, es necesario multiplicar el valor generado por 512000 y dividirlo entre 1073741824 para obtener el dato en gigabytes. (En algunos casos como el de **Samsung**, se sugiere redondear a 1000000000 para obtener el dato, de cualquier forma lo que interesa es un **dato aproximado**).

```

Terminal File Edit View Search Terminal Help
Smart Log for NVME device:nvme0n1 namespace-id:ffffffff
critical_warning      : 0
temperature          : 31 C
available_spare       : 100%
available_spare_threshold : 10%
percentage_used       : 0%
data_units_read      : 426 713
data_units_written    : 417 181
host_read_commands   : 6 642 465
host_write_commands  : 3 648 785
controller_busy_time  : 28
power_cycles          : 43
power_on_hours        : 21
unsafe_shutdowns     : 14
media_errors          : 0
num_err_log_entries  : 115
Warning Temperature Time : 0
Critical Composite Temperature Time : 0
Temperature Sensor 1  : 31 C
Temperature Sensor 2  : 40 C
Temperature Sensor 3  : 0 C
Temperature Sensor 4  : 0 C
Temperature Sensor 5  : 0 C
Temperature Sensor 6  : 0 C
Temperature Sensor 7  : 0 C
Temperature Sensor 8  : 0 C
eera5607@eera5607:~$

```

En el caso anterior al multiplicar 417181 por 512000 y dividirlo entre 1000000000 da como resultado 213 gigabytes escritos. Es importante consultar las **especificaciones del modelo** dadas por el fabricante para así hacernos una idea de que tan cerca nos encontramos del límite de garantía por ejemplo. En este caso **Samsung ofrece confiabilidad hasta 200 TBW** (Terabytes escritos) y 200 gigabytes son 0.2 terabytes por lo que corresponde aproximadamente a un 0.1 % del total. De nuevo, tiene únicamente 43 ciclos de carga por lo que es un disco nuevo.

redhat00 ~ # smartctl -i /dev/sda

smartctl 7.1 2020-04-05 r5049 [x86_64-linux-4.18.0-240.15.1.el8_3.x86_64] (local build)

Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===

Model Family: Western Digital Black

Device Model: WDC WD1003FZEX-00MK2A0

Serial Number: WD-WCC3FKZYVPY7

LU WWN Device Id: 5 0014ee 25fbf20b9

```
Firmware Version: 01.01A01
User Capacity: 1.000.204.886.016 bytes [1,00 TB]
Sector Sizes: 512 bytes logical, 4096 bytes physical
Rotation Rate: 7200 rpm
Device is: In smartctl database [for details use: -P show]
ATA Version is: ACS-2, ACS-3 T13/2161-D revision 3b
SATA Version is: SATA 3.1, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is: Sun Mar 28 10:56:04 2021 CEST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

The last two lines are the most important as they tell you if the [hard drive](#) is capable of running the SMART test. When you see that SMART support is available, but not enabled, you can run this command to enable it.

```
smartctl -s on -o on -S on /dev/vda
```

The output of the command above will look like this.

```
=== START OF ENABLE/DISABLE COMMANDS SECTION ===
SMART Enabled.
SMART Attribute Autosave Enabled.
SMART Automatic Offline Testing Enabled every four hours.
```

Now that we are sure SMART is available and enabled on our device, let's check some examples of smartctl commands. We will review how to use them and their purpose.

Disable smartctl

If you want to disable SMART capabilities on your hard drive, you can use this:

```
smartctl -s off /dev/vda
```

Check Drive Health

To display the overall health of your hard drive, you can use this one:

```
redhat00 ~ # smartctl -H /dev/sda
```

```
smartctl 7.1 2020-04-05 r5049 [x86_64-linux-4.18.0-240.15.1.el8_3.x86_64] (local build)
```

Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===

SMART overall-health self-assessment test result: PASSED

```
redhat00 ~ # smartctl /dev/sda --test=short
```

```
smartctl 7.1 2020-04-05 r5049 [x86_64-linux-4.18.0-240.15.1.el8_3.x86_64] (local build)
```

Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===

Sending command: "Execute SMART Short self-test routine immediately in off-line mode".

Drive command "Execute SMART Short self-test routine immediately in off-line mode" successful.

Testing has begun.

Please wait 2 minutes for test to complete.

Test will complete after Sun Mar 28 11:08:57 2021 CEST

Use smartctl -X to abort test.

Run Long/Short Test

When you want to run a short or long test on your HDD, you can use these two commands:

```
smartctl -test=short /dev/vda
smartctl -test=long /dev/vda
```

Background Testing

For any of the tests to run in the background, you can use -t flag:

```
smartctl -t short /dev/vda
smartctl -t long /dev/vda
smartctl -t conveyance /dev/vda
smartctl -t select /dev/vda
```

Foreground Testing

For the Foreground mode, the "-C" flag has to be added. Remember, run this only if the hard drive is not used by anything else!

```
smartctl -t short -C /dev/vda
smartctl -t long -C /dev/vda
```

```
smartctl -t conveyance -C /dev/vda
smartctl -t select -C /dev/vda
```

View Full Results

Let's check how we can view the results of our tests.

```
smartctl -a /dev/vda
```

Here is an example of the output that you will get, just a part of it because the whole output will be quite lengthy.

```
SMART Selective self-test log data structure revision number 1
SPAN MIN_LBA MAX_LBA CURRENT_TEST_STATUS
  1 0 0 Not_testing
  2 0 0 Not_testing
  3 0 0 Not_testing
  4 0 0 Not_testing
  5 0 0 Not_testing
```

```
redhat00 ~ # smartctl -l selftest /dev/sda
```

```
smartctl 7.1 2020-04-05 r5049 [x86_64-linux-4.18.0-240.15.1.el8_3.x86_64] (local build)
```

```
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org
```

```
==== START OF READ SMART DATA SECTION ====
```

```
SMART Self-test log structure revision number 1
```

```
Num Test_Description      Status              Remaining LifeTime(hours) LBA_of_first_error
# 1 Short offline        Completed without error    00%    13594      -
```

Disable smartctl

To disable SMART capabilities on the hard drive, we can use this:

```
# smartctl -s off /dev/vda
```

```
redhat00 ~ # vim /etc/smartmontools/smartd.conf
```

Advanced: Running as Smartmontools as a Daemon

You can run Smartmontools in the background and have it check drives and email when there are issues:

Open the file `/etc/default/smartmontools` with your favourite text editor. For example (using vim): `sudo vim /etc/default/smartmontools`. Uncomment the line

`start_smartd=yes.`

How `smartd` is going to scan the disks and what it will do in case of errors is controlled by the daemon configuration file, `/etc/smartd.conf`. Again, use your favourite text editor to open this file. There should be one uncommented line, similar to:

```
DEVICESCAN -m root -M exec /usr/share/smartmontools/smartd-runner
```

In this example (which is the default for Karmic), `smartd` will:

- scan for all ATA/SCSI devices (`DEVICESCAN`). The rest of the file will be ignored;
- mail a report to the 'root' account in case of trouble (`-m`);
- but instead of the `mail` command, it will execute `/usr/share/smartmontools/smartd-runner` and feed the report to it (`-M exec program`).

`/usr/share/smartmontools/smartd-runner` is a script that basically saves the report to a temporary file, and then runs anything it finds in `/etc/smartmontools/run.d/`; take a look there to understand what you already have (there should be a script that mails the report).

There are several `-M` directives that change when and how often reports are sent. You need to specify (`-m something`) in order to use them, even if you're not sending any mail.

You may include some useful options:

```
DEVICESCAN -H -l error -l selftest -f -s  
(0/../../../../5/11|L/../../../../5/13|C/../../../../5/15) -m root -M exec  
/usr/share/smartmontools/smartd-runner
```

In this example, `smartd` will:

- check the SMART health status (`-H`);
- report increases in both SMART error logs (`-l`);
- check for failure of any Usage Attributes (`-f`);
- schedule an Offline Immediate Test every Friday at 11 am, a Long Self-Test every Friday at 1 pm, and a Conveyance Self-Test every Friday at 3 pm (`-s`) -- see the `smartd` manual page for what these tests do so you can choose what suits you.

You may also replace `DEVICESCAN` with the path of the device which you'd like to be monitored (e.g. `/dev/sda`), and the daemon will only monitor this drive. You'll need one such line for each device.

Actions in case of trouble

You'll want to configure the actions `smartd` will take in case of trouble. If all you want is a notification shown on your desktop, skip to "Personal computer" below.

Most of the time, you only need to place a script in `/etc/smartmontools/run.d/`.

Whenever `smartd` wants to send a report, it will execute `smart-runner` and the latter will run your script.

You have several variables available to your script (again, see the `smartd` manpage). These come from a test run:

```
SMARTD_MAILER=/usr/share/smartmontools/smartd-runner
SMARTD_SUBJECT=SMART error (EmailTest) detected on host: XXXXX
SMARTD_ADDRESS=root
SMARTD_TFIRSTEPOCH=1267409738
SMARTD_FAILTYPE=EmailTest
SMARTD_TFIRST=Sun Feb 28 21:45:38 2010 VET
SMARTD_DEVICE=/dev/sda
SMARTD_DEVICETYPE=sat
SMARTD_DEVICESTRING=/dev/sda
SMARTD_FULLMESSAGE=This email was generated by the smartd daemon running on:
SMARTD_MESSAGE=TEST EMAIL from smartd for device: /dev/sda
```

Your script also has a *temporary* copy of the report available as "\$1". It will be deleted after you finish but the same content is written to `/var/log/syslog`.

redhat00 ~ # man smartd.conf

→ `apcupsd`

`Apcupsd`, abreviatura del demonio de fuente de alimentación ininterrumpida American Power Conversion, es una utilidad que se ejecuta en Linux, UNIX, macOS y Windows. Permite que la computadora interactúe con los UPS de APC. `Apcupsd` también funciona con algunas fuentes de alimentación OEM APC.

<http://www.apcupsd.org/manual/>

<https://wiki.debian.org/apcupsd>

→ `apctest`

`apctest` is a program that enables the running of certain low-level tests to check the operation of the attached simple signalling American Power Conversion Corp (APC) UPS and to check that your `apcupsd`(8) configuration is correctly setup and can establish communication with the UPS.

<https://linux.die.net/man/8/apctest>

→ `monit`

[Monit](#) is an open-source process [monitoring tool](#) for Linux operating system which helps you to monitor the system process. Whenever the service/process goes down, it automatically does the maintenance and repair of the particular process and ensures it is brought back online.

[Monit](#) can also be used for managing and monitoring of programs, files, directories, filesystems, TCP/IP network checks, protocol checks, and can utilize SSL for such checks.

It logs to its log file and notifies the user via customizable messages.

```
hp ~ # dnf install monit
```

```
hp ~ # monit
```

```
Starting Monit 5.26.0 daemon with http interface at [localhost]:2812
```

```
hp ~ # monit status
```

```
Monit 5.26.0 uptime: 1m
```

```
System 'hp.cadilinea.lan'
```

```
status          OK
monitoring status      Monitored
monitoring mode       active
on reboot            start
load average          [0.31] [0.53] [0.47]
cpu                  1.3%us 0.5%sy 0.2%wa
memory usage          2.3 GB [19.8%]
swap usage            0 B [0.0%]
uptime                3h 1m
boot time              Sun, 28 Mar 2021 09:00:56
data collected         Sun, 28 Mar 2021 12:02:02
```

```
hp ~ # firewall-cmd --permanent --add-port=2812/tcp
```

```
hp ~ # firewall-cmd --reload
```

```
hp ~ # systemctl start monit.service
```

```
hp ~ # systemctl status monit.service
```

- monit.service - Pro-active monitoring utility for unix systems

Loaded: loaded (/usr/lib/systemd/system/monit.service; disabled; vendor preset: disabled)

Active: active (running) since Sun 2021-03-28 12:05:37 CEST; 40s ago

Main PID: 8492 (monit)

Tasks: 2 (limit: 14180)

Memory: 1.0M

CPU: 37ms

CGroup: /system.slice/monit.service

└─8492 /usr/bin/monit -I

mar 28 12:05:37 hp.cadilinea.lan systemd[1]: Started Pro-active monitoring utility for unix systems.

mar 28 12:05:37 hp.cadilinea.lan monit[8492]: Starting Monit 5.26.0 daemon with http interface at [localhost]:2>

mar 28 12:05:37 hp.cadilinea.lan monit[8492]: 'hp.cadilinea.lan' Monit 5.26.0 started

mar 28 12:05:37 hp.cadilinea.lan monit[8492]: Starting Monit 5.26.0 daemon with http interface at [localhost]:2>

mar 28 12:05:37 hp.cadilinea.lan monit[8492]: 'hp.cadilinea.lan' Monit 5.26.0 started

hp ~ # vim /usr/lib/systemd/system/monit.service

[Unit]

Description=Pro-active monitoring utility for unix systems

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/monit -I

ExecStop=/usr/bin/monit quit

ExecReload=/usr/bin/monit reload

[Install]

WantedBy=multi-user.target

hp ~ # vim /etc/monitrc

Check a network link status (up/down), link capacity changes, saturation

and bandwidth usage.

#

check network public with interface eth0

if failed link then alert

if changed link then alert

if saturation > 90% then alert

if download > 10 MB/s then alert

if total uploaded > 1 GB in last hour then alert

#

#

Check custom program status output.

#

check program myscript with path /usr/local/bin/myscript.sh

if status != 0 then alert

#

#

#####

Includes

#####

##

It is possible to include additional configuration parts from other files or

directories.

#

include /etc/monit.d/*

#

GENERAL OPERATION

The behaviour of Monit is controlled by command-line options *and* a run control file, [monitrc](#), the syntax of which we describe in a later section. Command-line options override *.monitrc* declarations.

The default location for *monitrc* is *~/monitrc*. If this file does not exist, Monit will try */etc/monitrc* and a few other places. See [FILES](#) for details. You can also specify the control file directly by using the *-c* command-line switch to *monit*. For instance,

```
$ monit -c /var/monit/monitrc
```

Before Monit is started the first time, you can test the control file for syntax errors:

```
$ monit -t
$ Control file syntax OK
```

If there was an error, Monit will print an error message to the console, including the line number in the control file from where the error was found.

Once you have a working Monit control file, simply start Monit from the console, like so:

```
$ monit
```

You can change some configuration directives via command-line switches, but for simplicity it is recommended that you put these in the control file.

Monit will detach from the terminal and run as a background process, i.e. as a daemon process. As a daemon, Monit runs in cycles; It monitor services, then goes to sleep for a configured period, then wakes up and start monitoring again in an endless loop.

Options

The following options are recognized by Monit. However, it is recommended that you set options (when applicable) directly in the *.monitrc* control file.

-c *file* Use this control file

-d *n* Run Monit as a daemon once per *n* seconds. Or use "*set daemon*" in *monitrc*.

-g *name* Set group name for start, stop, restart, monitor, unmonitor, status and summary action.

-l *file* Print log information to this file. Or use "*set log*" in *monitrc*.

-p *pidfile* Use this lock file in daemon mode. Or use "*set pidfile*" in *monitrc*.

-s *statefile* Write state information to this file. Or use "*set statefile*" in *monitrc*.

-B Batch command line mode (no tabular output and no colors). Or use "*set terminal batch*" in *monitrc*.

- I Do not run in background mode (needed to run from init). Or use "*set init*" in monitrc.
- i Print Monit's unique ID
- r Reset Monit's unique ID. Use with caution
- t Run syntax check for the control file
- v Verbose mode, work noisy (diagnostic output)
- vv Very verbose mode, same as -v plus log stack-trace on error
- H [*filename*] Print MD5 and SHA1 hashes of the file or of stdin if the filename is omitted; Monit will exit afterwards
- V Print version number and patch level
- h Print a help text

Arguments

Once you have Monit running as a daemon process, you can call Monit with one of the following arguments. Monit will then connect to the Monit daemon (on TCP port 127.0.0.1:2812 by default) and ask the Monit daemon to perform the requested action. In other words; calling monit without arguments starts the Monit daemon, and calling monit *with* arguments enables you to communicate with the Monit daemon process.

start all

Start all services listed in the control file and enable monitoring for them. If the group option is set (-g), only start and enable monitoring of services in the named group ("all" is not required in this case).

start <name>

Start the named service and enable monitoring for it. The name is a service entry name from the monitrc file.

stop all

Stop all services listed in the control file and disable their monitoring. If the group option is set, only stop and disable monitoring of the services in the named group ("all" is not required in this case).

stop <name>

Stop the named service and disable its monitoring. The name is a service entry name from the monitrc file.

restart all

Stop and start *all* services. If the group option is set, only restart the services in the named group ("all" is not required in this case).

restart <name>

Restart the named service. The name is a service entry name from the monitrc file.

monitor all

Enable monitoring of all services listed in the control file. If the group option is set, only start monitoring of services in the named group ("all" is not required in this case).

monitor <name>

Enable monitoring of the named service. The name is a service entry name from the monitrc file. Monit will also enable monitoring of all services this service depends on.

unmonitor all

Disable monitoring of all services listed in the control file. If the group option is set, only disable monitoring of services in the named group ("all" is not required in this case).

unmonitor <name>

Disable monitoring of the named service. The name is a service entry name from the monitrc file. Monit will also disable monitoring of all services that depends on this service.

status [name]

Print service status information.

summary [name]

Print a short status summary.

report [up | down | initialising | unmonitored | total]

Report services state. The output can easily be parsed by scripts. Without options, prints a short overview of the state of all services managed by Monit. The option, *up* prints the number of all services in this state, *down* likewise and so on.

reload

Reinitialise a running Monit daemon, the daemon will reread its configuration, close and reopen log files.

quit

Kill the Monit daemon process

validate

Check all services listed in the control file. This action is also the default behaviour when Monit runs in daemon mode.

procmatch <regex>

Allows for easy testing of pattern for process match check. The command takes regular expression as an argument and displays all running processes matching the pattern.

<http://localhost:2812/>

admin:monit

| System | Status | Load | CPU | Memory | Swap |
|-----------------|--------|----------------------|------------------------|----------------|------------|
| hp.cadlinea.lan | OK | [0.65] [0.43] [0.42] | 6.7%us, 2.3%sy, 0.5%wa | 21.1% [2.4 GB] | 0.0% [0 B] |

Copyright © 2001-2019 [Troxlab](#). All rights reserved. [Monit web site](#) | [Monit Wiki](#) | [M3Mon](#)

→ **Añadimos servicio nginx**

hp ~ # dnf install nginx

hp ~ # systemctl start nginx

hp ~ # vim /etc/monit.d/nginx-monitor

check process nginx with pidfile /run/nginx.pid

start program "/usr/bin/systemctl start nginx.service"

stop program "/usr/bin/systemctl stop nginx.service"

if failed port 80 protocol http then restart

hp ~ # monit -t

Control file syntax OK

hp ~ # monit reload

Reinitializing monit daemon

hp ~ # monit status

Monit 5.26.0 uptime: 35m

Process 'nginx'

| | |
|--------------------|---|
| status | OK |
| monitoring status | Monitored |
| monitoring mode | active |
| on reboot | start |
| pid | 11156 |
| parent pid | 1 |
| uid | 0 |
| effective uid | 0 |
| gid | 0 |
| uptime | 2m |
| threads | 1 |
| children | 8 |
| cpu | - |
| cpu total | - |
| memory | 0.0% [1.2 MB] |
| memory total | 0.3% [33.6 MB] |
| security attribute | system_u:system_r:httpd_t:s0 |
| port response time | 31.781 ms to localhost:80 type TCP/IP protocol HTTP |

data collected Sun, 28 Mar 2021 12:41:04

Home > Use M.Monitor to manage all your Monit instances Monit 5.26.0

Monit Service Manager

Monit is running on hp.cadlinea.lan and monitoring:

| System | Status | Load | CPU | Memory | Swap |
|-----------------|--------|----------------------|------------------------|----------------|------------|
| hp.cadlinea.lan | OK | [0.40] [0.42] [0.36] | 1.2%us, 0.3%sy, 0.2%wa | 21.6% [2.5 GB] | 0.0% [0 B] |

| Process | Status | Uptime | CPU Total | Memory Total | Read | Write |
|---------|--------|--------|-----------|----------------|------|-------|
| nginx | OK | 3m | 0.0% | 0.3% [33.6 MB] | - | - |

Copyright © 2001-2019 [Tobias](#). All rights reserved. [Monit web site](#) | [Monit Wiki](#) | [M.Monitor](#)

hp ~ # systemctl stop nginx

Home > Use M.Monitor to manage all your Monit instances Monit 5.26.0

Monit Service Manager

Monit is running on hp.cadlinea.lan and monitoring:

| System | Status | Load | CPU | Memory | Swap |
|-----------------|--------|----------------------|------------------------|----------------|------------|
| hp.cadlinea.lan | OK | [0.72] [0.52] [0.40] | 2.9%us, 0.9%sy, 0.1%wa | 21.9% [2.5 GB] | 0.0% [0 B] |

| Process | Status | Uptime | CPU Total | Memory Total | Read | Write |
|---------|----------------|--------|-----------|--------------|------|-------|
| nginx | Does not exist | - | - | - | - | - |

Copyright © 2001-2019 [Tobias](#). All rights reserved. [Monit web site](#) | [Monit Wiki](#) | [M.Monitor](#)

Home > Use MMonit to manage all your Monit instances Monit 5.26.0

Monit Service Manager
Monit is running on hp.cadilinea.lan and monitoring:

| System | Status | Load | CPU | Memory | Swap | |
|------------------|--------|----------------------|---------------------------|----------------|------------|-------|
| hp.cadilinea.lan | OK | [2.16] [0.95] [0.56] | 11.0%us, 3.7%sy, 17.3%swa | 25.2% [2.9 GB] | 0.0% [0 B] | |
| Process | Status | Uptime | CPU Total | Memory Total | Read | Write |
| noinx | OK | 1m | 0.0% | 0.3% [34.0 MB] | - | - |

Copyright © 2001-2019 [TixieLab](#). All rights reserved. [Monit web site](#) | [Monit Wiki](#) | [MMonit](#)

What you can monitor

- Processes
- Server
- Cloud
- Disks
- Files
- Folders

hp ~ # vim /etc/monit.d/fs_check

check filesystem my_fs with path /

if failed permission 660 then unmonitor

if failed uid root then unmonitor

if failed gid disk then unmonitor

if space usage > 80% for 5 times within 15 cycles then alert

if space usage > 50% then alert

if inode usage > 30000 then alert

if inode usage > 99% then stop

```
hp ~ # monit -t
```

```
Control file syntax OK
```

```
hp ~ # monit reload
```

```
Reinitializing monit daemon
```

```
hp ~ # monit status
```

```
...
```

```
Filesystem 'my_fs'
```

```
status          Resource limit matched
monitoring status      Monitored
monitoring mode       active
on reboot            start
filesystem type       xfs
filesystem flags
rw,seclabel,relatime,attr2,inode64,logbufs=8,logbsize=32k,sunit=128,swidth=128,noquota
permission         555
uid                 0
gid                 0
block size          4 kB
space total          187.9 GB (of which 0.0% is reserved for root user)
space free for non superuser 148.0 GB [78.7%]
space free total      148.0 GB [78.7%]
inodes total         98578432
inodes free          98352203 [99.8%]
read                 13.5 MB/s [2.4 GB total], 217.4 reads/s [71758 reads total]
write                 1.3 MB/s [2.5 GB total], 72.6 writes/s [129004 writes total]
service time         1.191ms/operation (of which read 0.887ms, write 0.304ms)
data collected        Sun, 28 Mar 2021 13:02:22
```

```
...
```

Home > Use M.Monitor to manage all your Monit instances Monit 5.28.0

Monit Service Manager
Monit is **running** on hp.cadlinea.lan and monitoring:

| System | Status | Load | CPU | Memory | Swap | |
|-----------------|------------------------|--------------------|------------------------|----------------|------------|-------|
| hp.cadlinea.lan | OK | [1.05][1.17][1.00] | 2.2%us, 0.8%sy, 3.0%wa | 23.5% [2.7 GB] | 0.0% [0 B] | |
| Process | Status | Uptime | CPU Total | Memory Total | Read | Write |
| nginx | OK | 18m | 0.0% | 0.3% [34.0 MB] | - | - |
| Filesystem | Status | Space usage | Inodes usage | Read | Write | |
| my_fs | Resource limit matched | 21.3% [40.0 GB] | 0.2% [226235 objects] | 0 B/s | 0 B/s | |

Copyright © 2001-2018 Tixati s.p.a. All rights reserved. [Monit web site](#) | [Monit Wiki](#) | [M.Monitor](#)

hp ~ # monit unmonitor all

hp ~ # monit start all

hp ~ # monit stop all

hp ~ # monit --help

Usage: monit [options]+ [command]

Options are as follows:

- c file Use this control file
- d n Run as a daemon once per n seconds
- g name Set group name for monit commands
- l logfile Print log information to this file
- p pidfile Use this lock file in daemon mode
- s statefile Set the file monit should write state information to
- I Do not run in background (needed when run from init)
- id Print Monit's unique ID
- resetid Reset Monit's unique ID. Use with caution
- B Batch command line mode (do not output tables or colors)

- t Run syntax check for the control file
- v Verbose mode, work noisy (diagnostic output)
- vv Very verbose mode, same as -v plus log stacktrace on error
- H [filename] Print SHA1 and MD5 hashes of the file or of stdin if the filename is omitted; monit will exit afterwards
- V Print version number and patchlevel
- h Print this text

Optional commands are as follows:

- start all - Start all services
- start <name> - Only start the named service
- stop all - Stop all services
- stop <name> - Stop the named service
- restart all - Stop and start all services
- restart <name> - Only restart the named service
- monitor all - Enable monitoring of all services
- monitor <name> - Only enable monitoring of the named service
- unmonitor all - Disable monitoring of all services
- unmonitor <name> - Only disable monitoring of the named service
- reload - Reinitialize monit
- status [name] - Print full status information for service(s)
- summary [name] - Print short status information for service(s)
- report [up|down|..] - Report state of services. See manual for options
- quit - Kill the monit daemon process
- validate - Check all services and start if not running
- procmatch <pattern> - Test process matching pattern

→ **monit bash_completion**

hp ~ # vim /etc/bash_completion.d/monit

BIBLIOGRAFIA:

<https://robustiana.com/302-smart-linux-hdd-ssd>

<https://www.liquidweb.com/kb/how-to-install-and-configure-smartctl/>

<https://help.ubuntu.com/community/Smartmontools>

<https://wiki.debian.org/apcupsd>

<http://www.apcupsd.org/manual/>

<https://wiki.debian.org/apcupsd>

<https://linux.die.net/man/8/apctest>

<https://mmonit.com/monit/documentation/monit.html>

<https://linuxhostsupport.com/blog/check-file-systems-files-hosts-and-custom-scripts-with-monit/>

<https://www.itzgeek.com/how-tos/linux/centos-how-tos/how-to-install-monit-on-centos-8-rhel-8.html>

<https://raw.githubusercontent.com/scop/bash-completion/f8a5df2e876f02851fb8bfd504ae740f958b81c0/completions/monit>

Creative Commons**Reconocimiento-NoComercial-CompartirIgual 3.1 ESPAÑA**

© 2021 by carlos briso. Usted es libre de copiar, distribuir y comunicar públicamente la obra y hacer obras derivadas bajo las condiciones siguientes:

a) Debe reconocer y citar al autor original.

b) No puede utilizar esta obra para fines comerciales (incluyendo su publicación, a través de cualquier medio, por entidades con fines de lucro.

c) Si altera o transforma esta obra o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta. Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor. Los derechos derivados de usos legítimos u otras limitaciones no se ven afectados por lo anterior. Licencia completa en castellano.

→ La información contenida en este documento y los derivados de éste se proporcionan tal cual son y los autores no asumirán responsabilidad alguna si el usuario o lector hace mal uso de éstos.