

<https://domology.es/integracion-proxmox-en-homeassistant/>

Proxmox integration in Homeassistant

December 17, 2019 mig77angel 0 comments hass , homeassistant , proxmox , ssh

After taking a while with [Proxmox](#) you realize that it is usually a bit slow to have to go to its interface to just turn on or stop a [ct](#) or a [vm](#) , so I have looked for a way to integrate it into [Homeassistant](#) to make it more comfortable and also You can do it as if you turned on or off a light or a plug, with alexa or Google Home.

Contents [[hide](#)]

- [1 Disk space sensors](#)
- [2 Switch to start / stop ct](#)
- [3 Switch to start / stop vm](#)

Disk space sensors

First we are going to get the storage of the Proxmox system in some sensors, with the command `pvesm status`(with `sudo` in front if applicable) you can see the status of the Proxmox system storage:

```
root@pve:~# pvesm status
```

```
Name Type Status Total Used Available %
local dir active 59600812 30646720 25896840 51.42%
local-lvm lvmthin active 158646272 90777396 67868875 57.22%
```

With this data we can take one or more sensors and show it in a progress bar in Homeassistant or in the way that each one wants.

In my example it is like this:

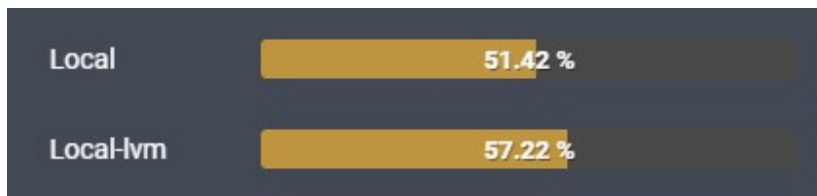
sensor:

```
- platform: command_line
  name: proxmox_local_used
  unit_of_measurement: '%'
  command: >-
    ssh root@192.168.1.3 pvesm status | awk 'NR==2' | awk '{ print $7 }' | cut -d
  '%' -f 1
  scan_interval: 120
- platform: command_line
  name: proxmox_local_lvm_used
  unit_of_measurement: '%'
  command: >-
    ssh root@192.168.1.3 pvesm status | awk 'NR==3' | awk '{ print $7 }' | cut -d
  '%' -f 1
  scan_interval: 120
```

This generates two sensors with the % of used disk of each Proxmox storage, if you have more you can put the ones you want.

To do this as I do, you need the system where Homeassistant is installed to have ssh trust with the Proxmox system.

Then we can make that data a little more beautiful with a progress bar for each storage:



This is created with the custom-component [custom: bar-card](#) , this is my example code of one of the sensors in lovelace:

```
- type: custom:bar-card
  title: Local
  title_position: left
  height: 20px
  padding: 2px
  entity: sensor.proxmox_local_used
  severity:
    - value: 33
      color: '#40bf40' #verde
    - value: 66
      color: '#bf9540' #ocre
    - value: 100
      color: '#bf4040' #rojo
  card_style:
    border-radius: 0px
    background: 'rgba(0, 0, 0, 0.0)'
    box-shadow: 'inset 2px 2px 3px 1px rgba(0, 0, 0, 0.0)'
```

Switch to start / stop ct

To start and stop the containers from the command line we use the command `pct start/stop`(also with `sudo` if applicable) and to see the status with the command `pct status`, with those commands we make the [template switch](#) corresponding to each ct for Homeassistant.

Each ct is named by its name or code, we can find out what we have with the command `pct list`, and it shows you a list and its status:

```
root@pve:~# pct list
VMID Status Lock Name
100 stopped debian9
101 stopped plex
102 stopped emby
104 stopped calibre
108 stopped ftp
109 stopped sql
```

This is one of my example switches:

```
switch:
  - platform: command_line
    switches:
      proxmox_ct_debian9:
        friendly_name: 'Debian 9'
        command_on: "ssh root@192.168.1.3 'pct start 100'"
        command_off: "ssh root@192.168.1.3 'pct stop 100'"
        command_state: "ssh root@192.168.1.3 pct status 100 | grep -oPc 'status:
running'"
        value_template: '{{ value == "1" }}'
```

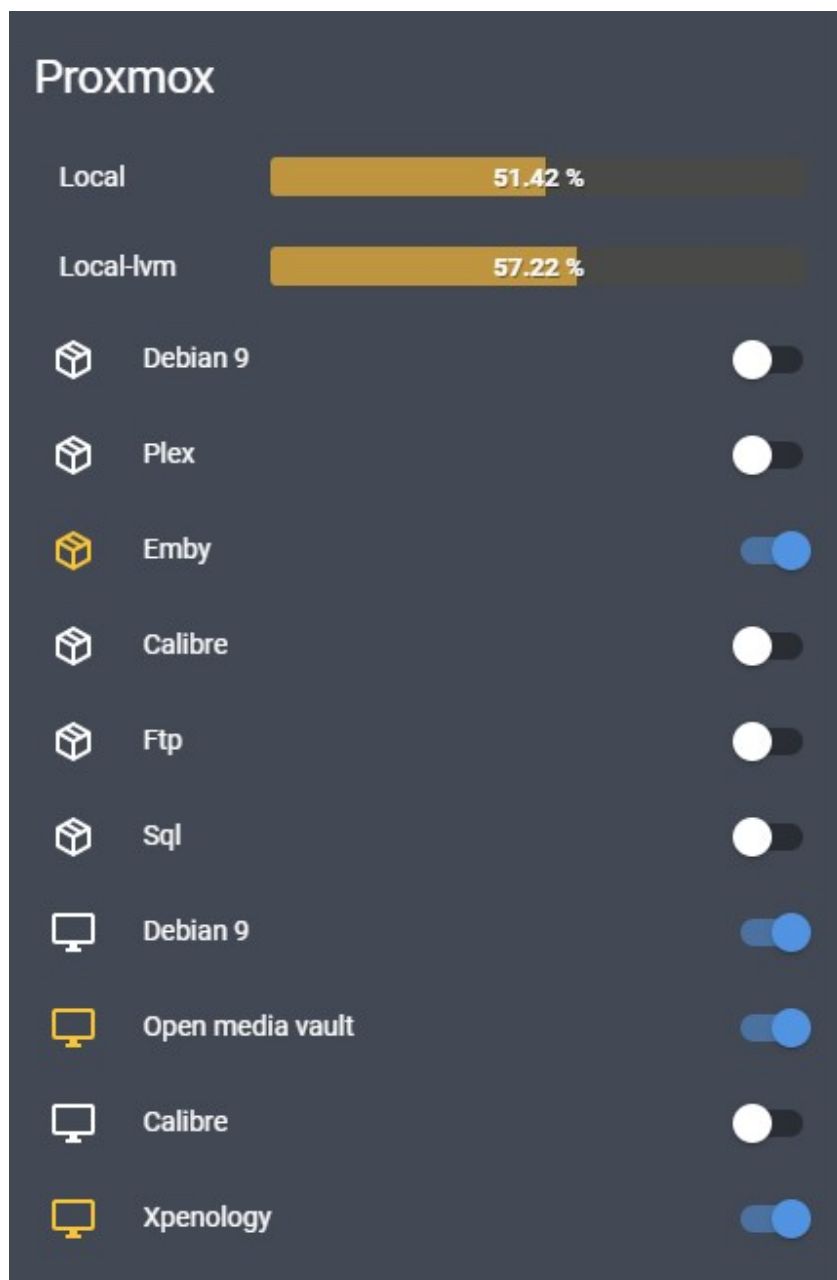
Switch to start / stop vm

To turn on and stop the vm is done in the same way as the ct but using qm instead of pct in the commands:

An example:

```
switch:
  - platform: command_line
    switches:
      proxmox_vm_omv:
        friendly_name: 'Open media vault'
        command_on: "ssh root@192.168.1.3 'qm start 105'"
        command_off: "ssh root@192.168.1.3 'qm stop 105'"
        command_state: "ssh root@192.168.1.3 qm status 105 | grep -oPc 'status:
running'"
        value_template: '{{ value == "1" }}'
```

This would be the final result with all the switches and the progress bars with the storage:



To leave those icons you can use this in the customize part:

homeassistant:

customize:

```
switch.proxmox_ct_debian9:  
  icon: mdi:package-variant-closed  
switch.proxmox_vm_debian9:  
  icon: mdi:monitor
```