

- 1) When we add a new VM in the OLD UI, now we have 2 options of "**VM Type**", one is "**Virtual Machine**" and another one is "**Docker VM**", basically there are some differences between both, here I describe what are the differences and what need to be worked.
  - a) First of all, we need to have a switch where users can choose to add a '**VM Machine**' or add a '**Docker VM**'.
  - b) The first screen is slightly different for both options, for 'VM Machine' you can choose the '**Boot Method**' and for 'Docker VM' it is hidden as 'Docker VM' uses **GRUB**. [Fig. 1]

Virtual Machine	RancherOS (Docker)
<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #333; color: white; padding: 2px; display: flex; justify-content: space-between; align-items: center;"> <span>Add VM</span> <span>✕</span> </div> <div style="padding: 10px;"> <p><b>VM Type:</b> <span style="border: 1px solid #ccc; padding: 2px;">Virtual Machine</span> <span style="font-size: 0.8em;">▼</span></p> <p><b>Name:</b> <input style="width: 100%;" type="text" value=""/></p> <p><b>Description:</b> <input style="width: 100%;" type="text" value=""/></p> <p><b>Virtual CPUs:</b> <input style="width: 100%;" type="text" value="1"/></p> <p><b>Memory Size (MiB):</b> <input style="width: 100%;" type="text" value=""/></p> <p><b>Boot Method:</b> <span style="border: 1px solid #ccc; padding: 2px;">UEFI</span> <span style="font-size: 0.8em;">▼</span> <span style="font-size: 0.8em;">i</span></p> <p><b>Autostart:</b> <input type="checkbox"/> <span style="font-size: 0.8em;">i</span></p> <p style="text-align: right;"><span style="border: 1px solid #ccc; padding: 2px 10px;">OK</span> <span style="border: 1px solid #ccc; padding: 2px 10px;">Cancel</span></p> </div> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #333; color: white; padding: 2px; display: flex; justify-content: space-between; align-items: center;"> <span>Add VM</span> <span>✕</span> </div> <div style="padding: 10px;"> <p><b>VM Type:</b> <span style="border: 1px solid #ccc; padding: 2px;">Docker VM</span> <span style="font-size: 0.8em;">▼</span></p> <p><b>Name:</b> <input style="width: 100%;" type="text" value=""/></p> <p><b>Description:</b> <input style="width: 100%;" type="text" value=""/></p> <p><b>Virtual CPUs:</b> <input style="width: 100%;" type="text" value="1"/></p> <p><b>Memory Size (MiB):</b> <input style="width: 100%;" type="text" value=""/></p> <p><b>Autostart:</b> <input type="checkbox"/> <span style="font-size: 0.8em;">i</span></p> <p style="text-align: right;"><span style="border: 1px solid #ccc; padding: 2px 10px;">OK</span> <span style="border: 1px solid #ccc; padding: 2px 10px;">Cancel</span></p> </div> </div>

Fig. 1 - Boot Method difference.

- 2) Raw disk for "Docker VM" vs "Virtual Machine".
  - a) The RAW disk for "Virtual Machines" has only few fields in its form, as we can see on Fig.2.
  - b) The "Docker VM" has extra fields, also the "**Raw File**" field can't be used with an existent RAW file, basically the user needs to type the path and the file name and middlewared will copy the pre-built image and resize it using truncate. If user select an existent RAW file, the middlewared won't do any action, as it believes the RAW image contains data.
  - c) The extra fields for "Docker VM" are:
    - i) "**Disk Boot**": tells grub this is a bootable disk.
    - ii) "**Password**": Sets the rancher user password.
    - iii) "**Disk size**": It will use truncate to resize the pre-built image based on this value.

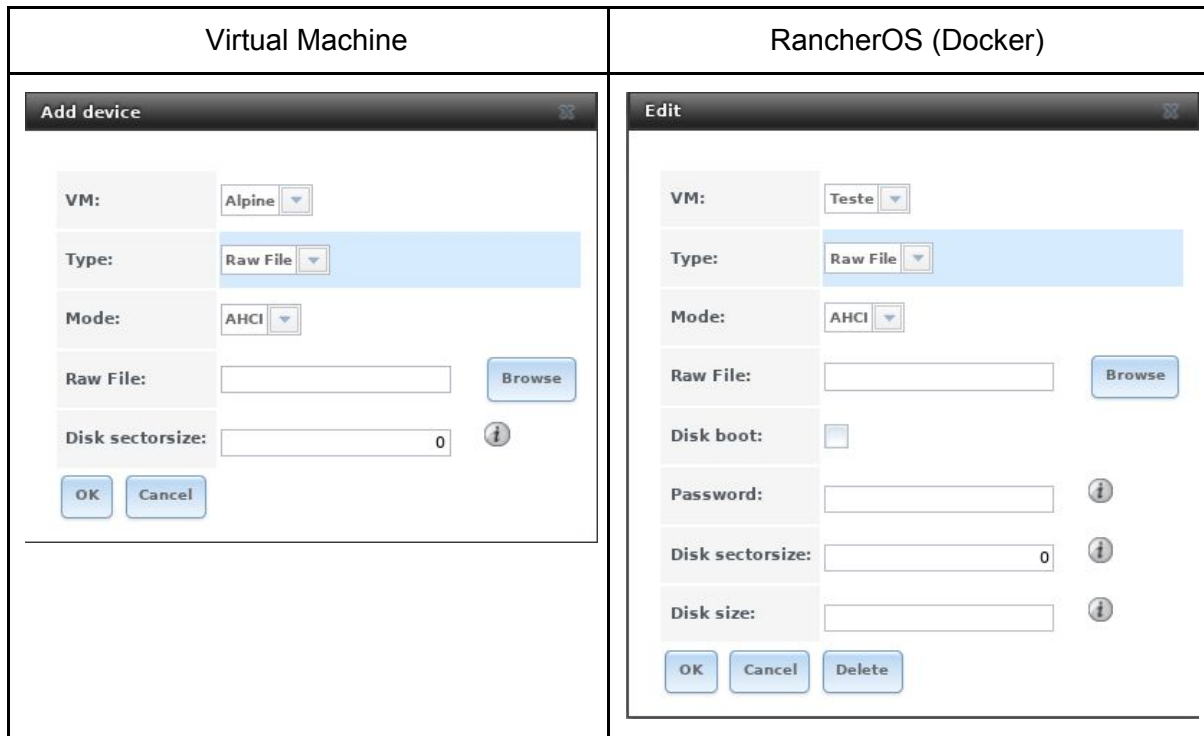


Fig 2. RAW disk difference.

- 3) Prior to start a "Docker VM" we need to setup a **common sharefs** place where we will download a pre-built RancherOS image and also we will store the guests configuration file.
  - a) There are two websocket methods for that purpose, the first one is: `midctl call vm.activate_sharefs "pool name"`, we can omit the "pool name" and it will take the first pool available in alphabetical order. We do it in the OLD UI, because of some Django limitations, but would be nice to have a form to allow users to choose which **"pool"** they want use to download the pre-built image as well as to store the guest grub configuration files.
  - b) Second command is: `midctl call vm.get_sharefs` that will returns the current sharefs area used among the guests "Docker VM".
  - c) When we start a "Docker VM" at first time, it will check if the pre-built image exists, if not it will download the image from our CDN using the websocket call `vm.fetch_image "RancherOS"` it will return a **job ID**, it will place a job at `core.get_jobs` with the name '**container**', we need to track the progress there to be able to create a 'progress bar'. Please, use the old UI as a reference for the "progress bar". You also can use the websocket `vm.get_download_status <job id>` to track the job progress.
  - d) After the download is finished, you can use another websocket call to get the relative path of the image, ex.: `midctl call vm.image_path 'RancherOS'`.
  - e) To finish the 'Docker VM' setup, there are other 2 webcalls to decompress and resize the image according to RAW disk settings.
    - i) `midctl call vm.decompress_gzip` that needs two arguments, first the relative path of the pre-built image and second the relative path of RAW DISK defined by the user.

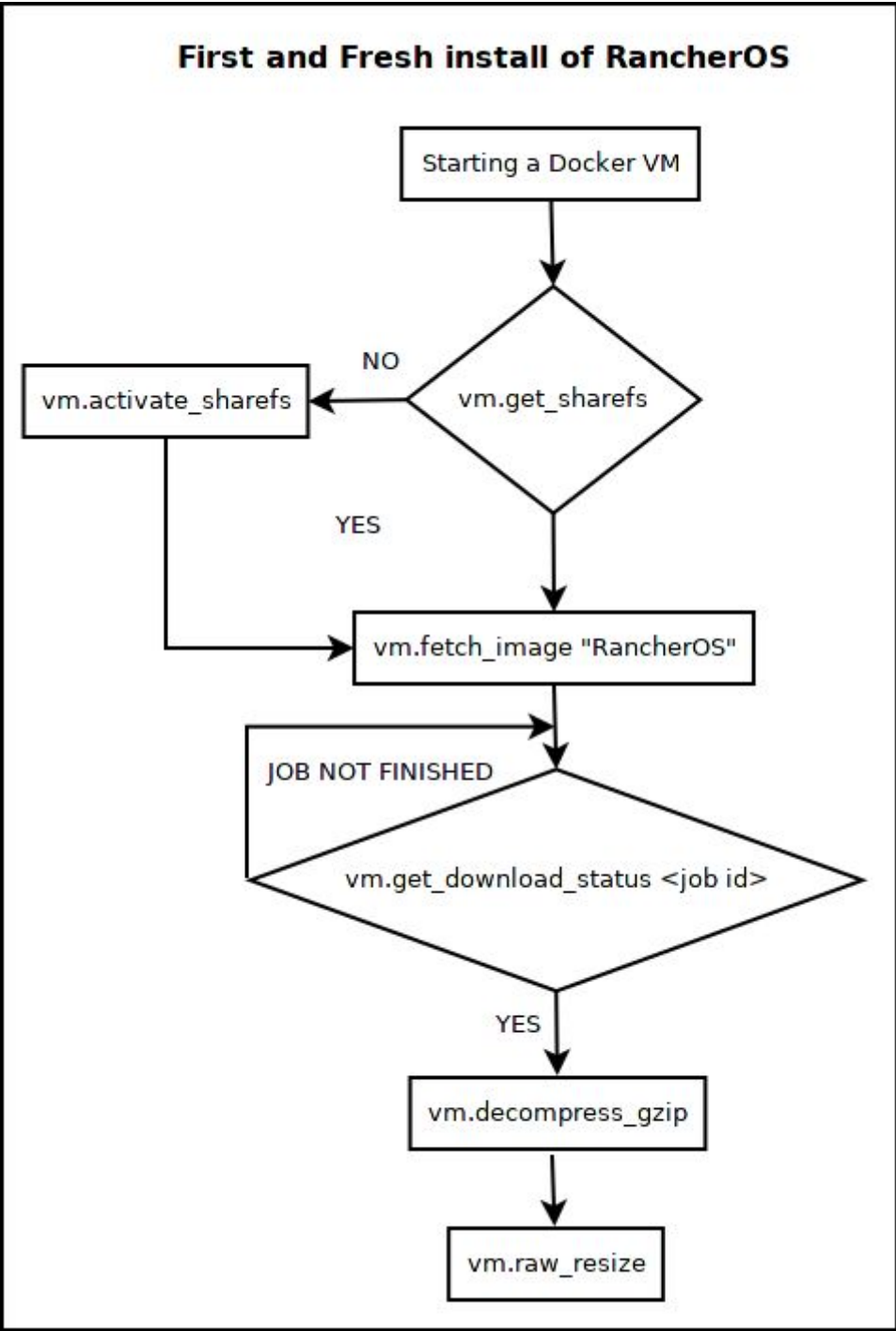
- ii) Midclt call **vm.raw\_resize** followed by 2 arguments, the relative path of RAW DISK and the size.

**NOTE:** In the OLD UI, I use `vm.activate_sharefs` without pass a "pool name", however for the new UI would be good to have a form just to set the sharefs area. The best approach would be use the same sharefs area for all "Docker VM".

4) Deleting a "Docker VM", we basically need to call a websocket to remove the configuration files related with that guest vm.

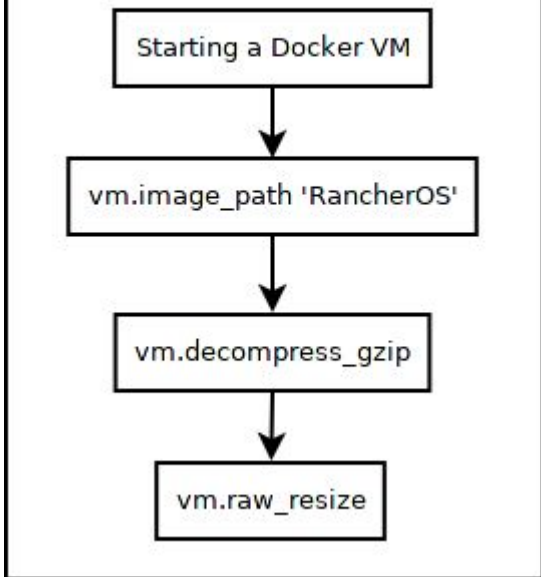
- a) We must call: midclt **call vm.rm\_container\_conf** with an argument that is the VM ID.

5) Here is a diagram of all calls related when we try to install at first time a Docker VM.



6) Here is a diagram related with all calls when we Start and Remove a Docker VM.

### Start a new Docker VM



### Remove a Docker VM

