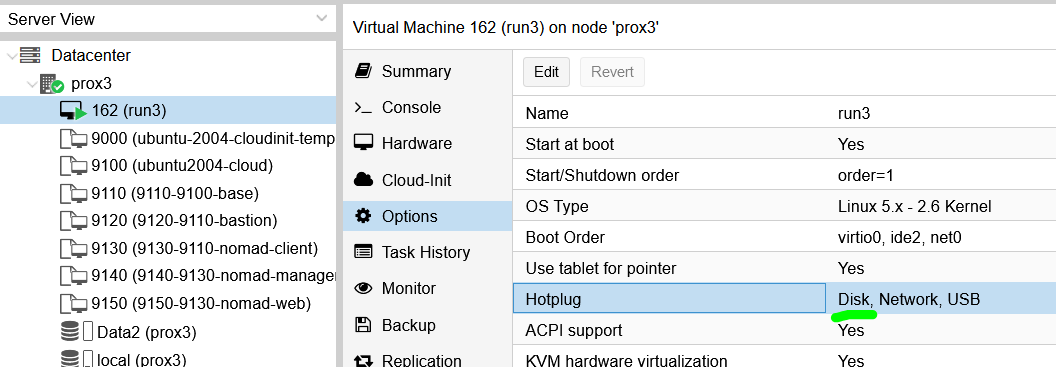
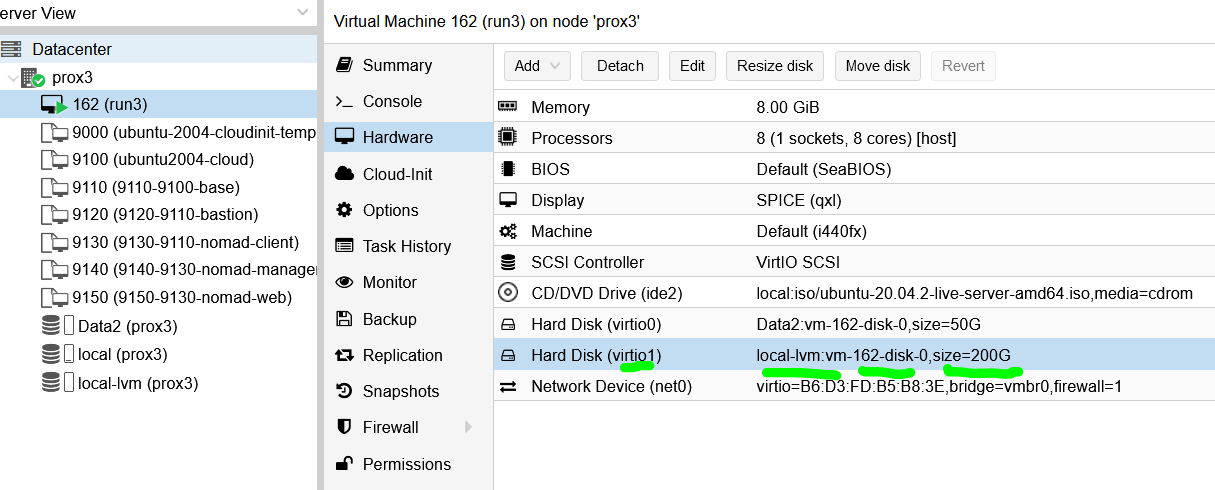
**24 - R710 Proxmox Add 2nd disk to run host and Docker for localstack S3 and SQS**

These notes cover adding a 2nd drive the run3 host and Docker to utilise the localstack dockerfile to provide S3 and SQS services

This document builds upon the previous documents.

# Add 2nd disk:

1. For the run3 host, in Proxmox GUI, ensure Disk has been selected for Hotplug  
   
2. Again in Proxmox GUI, for the run3 host, for Hardware, click on Add and fill out the info to end up with the 2nd hard disk shown as per:  
   
3. Ssh into run3 host and switch to root with:

**sudo su -**

1. Do:  
   **fdisk -l**  
     
   to confirm that you can see the new 2nd drive as:  
   **/dev/vdb**
2. Enter the following commands:  
   **parted /dev/vdb mklabel gpt**

**parted -a opt /dev/vdb mkpart primary ext4 0% 100%**  
  
**mkfs.ext4 -L S3andSQS /dev/vdb1  
  
mkdir -p /mnt/S3andSQS \*\*\* may need to not create this as ‘root’, and do as ‘rhys’**

1. Edit:  
   **/etc/fstab**  
     
   and add line:  
     
   **LABEL=S3andSQS /mnt/S3andSQS ext4 defaults 0 2**
2. Doing:  
   **fdisk -l**  
     
   should now show something like:  
   *Disk /dev/vdb: 200 GiB, 214748364800 bytes, 419430400 sectors*

*Units: sectors of 1 \* 512 = 512 bytes*

*Sector size (logical/physical): 512 bytes / 512 bytes*

*I/O size (minimum/optimal): 512 bytes / 512 bytes*

*Disklabel type: gpt*

*Disk identifier: 2F482D94-B19B-4E51-B87C-57CD729FD5F5*

*Device Start End Sectors Size Type*

*/dev/vdb1 2048 419428351 419426304 200G Linux filesystem*

1. Reboot the run3 host and as root confirm the 2nd disk is still there and the S3andSQS directory is still there.

# Docker in run3 host

1. To install Docker into the VM, follow Steps:  
   **sudo apt update**

**sudo apt install apt-transport-https ca-certificates curl software-properties-common**

**curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -**

**sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"**

**sudo apt update**

**apt-cache policy docker-ce**

**sudo apt install docker-ce**

**sudo systemctl status docker**

to see something like:

*● docker.service - Docker Application Container Engine*

*Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)*

*Active: active (running) since Mon 2022-03-07 16:09:06 UTC; 26s ago*

*TriggeredBy: ● docker.socket*

*Docs: https://docs.docker.com*

*Main PID: 4188 (dockerd)*

*Tasks: 13*

*Memory: 33.5M*

*CGroup: /system.slice/docker.service*

*└─4188 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock*

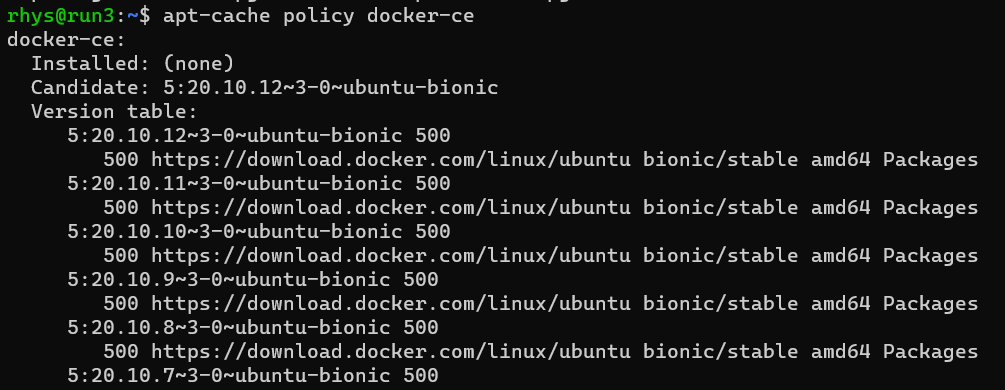
1. Do:

**sudo usermod -aG docker rhys**

then log out and log back in and do:

**id -aG**

to confirm that rhys is in the '**docker**' group.

1. And when you get to the stage of running command:  
   **apt-cache policy docker-ce**  
     
   it produces something like:  
   
2. Now edit file (to enable IPv4 packet forwarding):  
   **sudo nano /etc/sysctl.conf**  
   and look for line:  
     
   and uncomment line to be:  
     
   and then reboot run3 host for changes to take effect.
3. To check Docker is running:  
   **systemctl status docker**
4. To test Docker:

**docker run hello-world**

1. Add **portainer** for use from Ubuntu desktop to manage docker containers with:

**docker run -d -p 9100:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer-ce:2.6.0**

# localstack install:

1. To use / test localstack, awscli will be needed to be installed, with:

**sudo apt install awscli  
  
\*\*\* copy out instructions from below articles \*\*\***

1. Now work thru this to setup SQS and SNS and test them:  
   <https://onexlab-io.medium.com/localstack-sns-to-sqs-47a38f33b8f4>
2. Then test that commands can access the SQS, etc on run3 host from other terraformed VM’s  
   (add notes on what was done here to test)
3. Use any other commands from this:  
   <https://onexlab-io.medium.com/localstack-sqs-a0c36fd13108>  
     
   to do further checks.  
   (add notes on what was done here to test)
4. Do the same for S3 with this:  
   <https://onexlab-io.medium.com/localstack-s3-e28ad393c09>

1. And check I can access from other terraformed machines.  
   (add notes on what was done here to test)
2. Figure out that S3 files are in some s3 directory in the 2nd disk
3. Is there any app / web app that I can use to inspect S3 bucket and also the SQS and SNS queues ?
4. Sus running the docker compose in a detached manner such that it will run at power up / reset of run3 host, so that its always running (or not in a detached state, but running at boot)

1. Work through this article, to run in run3 and also on terraformed host:

<https://joerg-pfruender.github.io/software/docker/microservices/testing/2020/01/25/Localstack_in_Docker.html>

1. Also double check through this to see if its got anything of use:  
   <https://towardsaws.com/sns-and-sqs-with-localstack-using-golang-16b291f45e0b>