**28 - R710 Rebuild run3 host as Ubuntu Workstation**

This describes saving off info from existing run3 host that was built on top of Ubuntu server, Replacing everything in the VM with a fresh install of **ubuntu-22.04-desktop-amd64**, configuring all needed applications and reinstating any data saved from old run3 host.

The need for this arose after some grief with ansible and python3 and I mistakenly deleted python3 and before restoring it I rebooted run3, which lost networking and messed everything up.

So a complete rebuild is now the only option.

This document aims to capture a list of everything that needs to be installed to get back the whole of the run3 host, but starting from a Ubuntu workstation, because adding a GUI to a server configuration did not seem to work so well in so far as I could not select the version of gnome I wanted in remote desktop access.

Here goes …

# 1 – Saving existing data from run3 host

1. Shut run3 host down
2. In Proxmox GUI, configure run3 host to boot Ubuntu GUI ISO:  
   
3. Then in Options set the boot order to boot from the ISO first
4. Click on Console and start run3 VM.
5. This will boot the Ubuntu GUI and as it boots, select Try Ubuntu
6. The reason for doing this is to backup various directories whilst they are not being used in a running run3 VM
7. On the left hand side there are icons. Move mouse over them and scroll down to 50G volume and open this.
8. In the folder view window that opens, open up a terminal at that location.
9. Cd into home directory and create a tar.gz file (preserving permissions, etc) of the ‘rhys’ directory. A file of about 6.7GB is created.
10. Open up firefox and log into my online jupyter notebook and drag and drop the created file into a folder. This will take an hour or so to upload.
11. Once uploaded, download it onto windows machine in folder red-run3
12. Remove the tar file that was created in the VM
13. Cd into /
14. Create a tar.gz of /etc and copy to jupyter, download to windows machine and remove the tar file
15. Close the GUI folder for the 50GB partition
16. Open a GUI directory view of the 200GB volume named S3andSQS
17. Right click in that window and select ‘open in terminal’
18. Create a tar.gz of tmp directory and copy to jupyter, download to windows machine and remove the tar file
19. Close the directory, power off the GUI (and press enter to eject the CD).
20. On windows machine, inspect the saved tar files with 7-zip … if any problems found, then parts of the previous steps may need redoing until backups are OK
21. Take a snapshot in proxmox of the current run3 host, just in case future steps need something else from it.
22. That completes the backup of old data

# 2 – Install Ubuntu-22.04-desktop-amd64.iso

1. From the previous steps, run3 VM is setup to boot from **ubuntu-22.04-desktop-amd64.iso**
2. In proxmox for the run3 VM, select the Console view for the VM and start the VM
3. Then referring to Previous document: “**12 – R710 Proxmox vM – Ubuntu Workstation with Static IP**”, follow step 12 onwards under the section titled: “**VM – Full Virtual Machine ( Ubuntu 20.04 LTS )**” … but note we are now using Ubuntu 22.04
4. Also, when asked ‘Who are you’, use name of ‘**rhys**’ and computer name of ‘**run3**’.
5. NOTE: in the document that is being worked thru, where it sets up a static IP, use the ip of 162.198.124.162 for this new setup
6. Then complete the rest of the steps in that document.

# 3 – Check the remote desktop log in works

1. From windows machine try the remote desktop login … it works … FAB.

# 4 – The previous S3andSQS volume mount

1. Within the proxmox console Ubuntu GUI login it turns out the new install has mounted the previous hosts S3andSQS mount with all of its contents … FAB.
2. However in the remote desktop, opening up the disks icon in the utilities it is not mounted …
3. So in remote desktop, in a terminal, do:  
   **sudo su -**  
   **mkdir -p /mnt/S3andSQS**Edit **/etc/fstab,** and add line:  
   **LABEL=S3andSQS /mnt/S3andSQS ext4 defaults 0 2**and reboot run3 VM
4. In remote desktop, open a terminal and cd into /mnt/S3andSQS and do ls, where you should see the original tmp directory and lost+found. Within tmp you sould see minio and other original directories.

# 5 – Ubuntu Desktop additional software, including ‘go’

1. Follow the steps in this document: “**15 - R710 Proxmox VM – Ubuntu Desktop additional Software**”
2. Edit .bashrc and adjust history size:  
   HISTSIZE=100000

HISTFILESIZE=200000

1. At end of the .bashrc, replace the exporting of the go bath with this:  
     
   **myssh\_agent () {**

**umask 077**

**local f=~/.ssh/spy kee=**

**if [[ ! -f $f ]]; then**

**ssh-agent -s | grep --color -v '^echo' > $f**

**fi**

**. $f**

**if [[ -z $SSH\_AGENT\_PID || -z "$(ps -p $SSH\_AGENT\_PID | grep ssh-agent)" ]]; then**

**\rm $f**

**myssh\_agent**

**else**

**if [[ -z "$(ssh-add -l | grep '^[0-9]')" ]]; then**

**ssh-add**

**fi**

**for kee in ~/.ssh/id\_rsa4k ~/.ssh/id\_rsa\_$myHOST; do**

**[[ -f $kee ]] || continue**

**local fing=$(ssh-keygen -l -f $kee | awk '{print $2}')**

**[[ -n $fing && -z "$(ssh-add -l | grep " $fing ")" ]] || continue**

**ssh-add $kee**

**done**

**fi**

**}**

**# Go Global variables**

**export GOROOT="/usr/local/go"**

**export GOPATH="$HOME/Go"**

**export PATH="$PATH:$GOPATH/bin:$GOROOT/bin"**

**alias on='cd ~/public/src/github.com/redhug1'**

**set +e**

**set +o posix**

**# !!! put back in when nomad has been installed ... complete -C /usr/bin/nomad nomad**

1. Exit and restart the terminal

# 6 – Terraform, ansible and further run3 setup

1. do:  
   **ssh-keygen -t rsa -b 4096**
2. do:  
   **sudo apt-get update && sudo apt-get install -y gnupg software-properties-common**
3. We will install Terraform manually to avoid any unwanted updates:  
   Open Firefox and go to:  
   <https://developer.hashicorp.com/terraform/downloads>  
     
   There select and download:  
   **terraform\_1.3.7\_linux\_amd64.zip**
4. In a terminal, in Downloads folder, do:  
   **unzip terraform\_1.3.7\_linux\_amd64.zip**
5. Then do:  
   **sudo mv terraform /usr/bin**  
   **sudo chown root:root /usr/bin/terraform**
6. Verify terraform installed with:

**terraform version**

1. Install “auto-complete” Terraform extension, with:  
   **terraform -install-autocomplete**  
   **source ~/.bashrc**

Install ansible with:  
**sudo apt install ansible  
sudo apt install sshpass**(as of 4th Feb 2023 this installed version 2.10.8 of ansible, python is 3.10.6)

# 7 – Samba on run3 host:

1. do:  
   **bla bla !!! fill in**

Copy bits needed from document 18 – samba on run host, check host ip’s on WSL, change vm host name ?

As the following steps are followed, also detail the restoration of any data

!!! also after each step take a snapshot … and at some point if possible clean out previous snapshots.

Copy bits needed from document 21

Copy bits needed from document 22

Copy bits needed from document 23

Copy bits needed from document 24

Don’t need to create the dirs, but create users and change owners of dirs. + do other bits for minio … and then the rest of the document.

Copy bits needed from document 25

Copy bits needed from document 26

Copy bits needed from document 27