Setting Up Proxmox VE

manjit28.medium.com/setting-up-proxmox-ve-9f85e158f127

Manjit Singh 11 сентября 2024 г.

Manjit Singh

This post is about setting up Proxmox on a bare metal server. But first of all, what is Proxmox? It’s a virtualization platform that we can use to run virtual machines and containers. Compared to some other alternatives out there, it comes with Web UI, storage and backup solutions right out of the box. Installation process is pretty simple but for any hypervisor, we need to prepare the network, storage etc. in advance. I went over that in my previous post about preparing to install multiple hypervisors:

Multi-Hypervisor Homelab -Exploring VMWare ESXi Alternatives

This post is about my journey of setting up multi hypervisor home lab, exploring ESXi, Proxmox, XCP-ng, Nutanix and…

medium.com

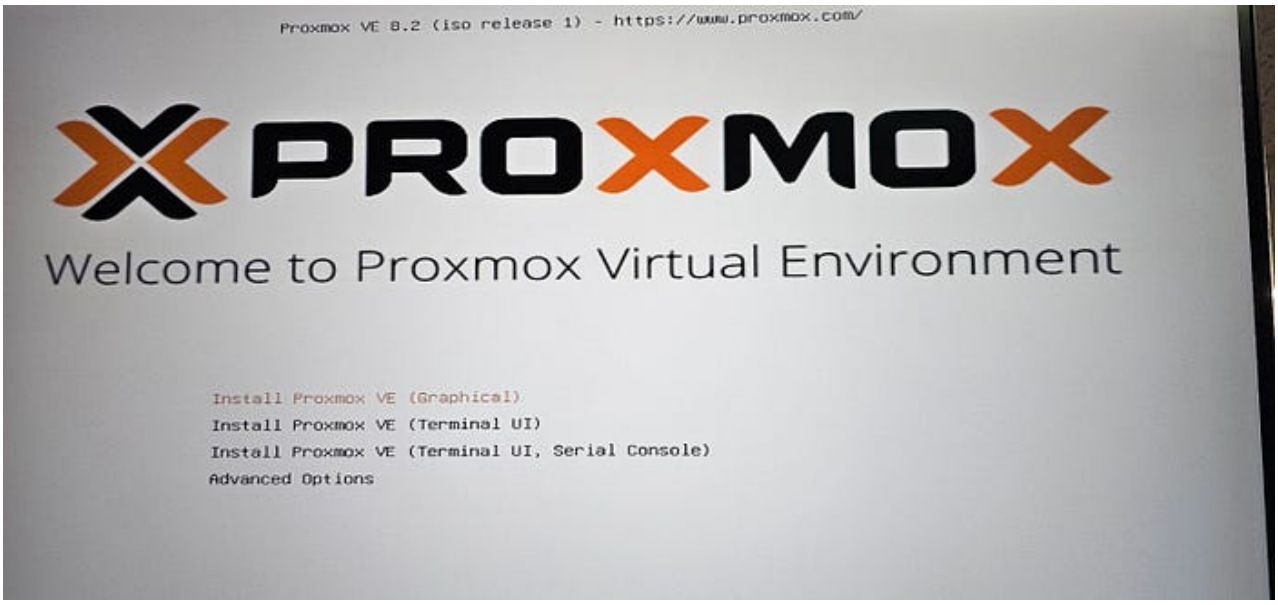
Very first thing is to get Proxmox VE ISO from following location:

Downloads

Edit description

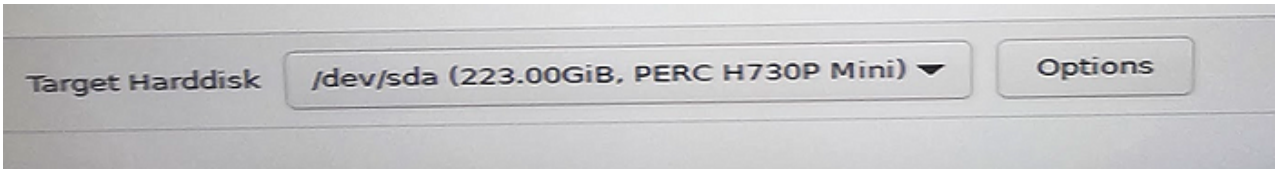
www.proxmox.com

Get this on USB media and boot the server with that ISO.



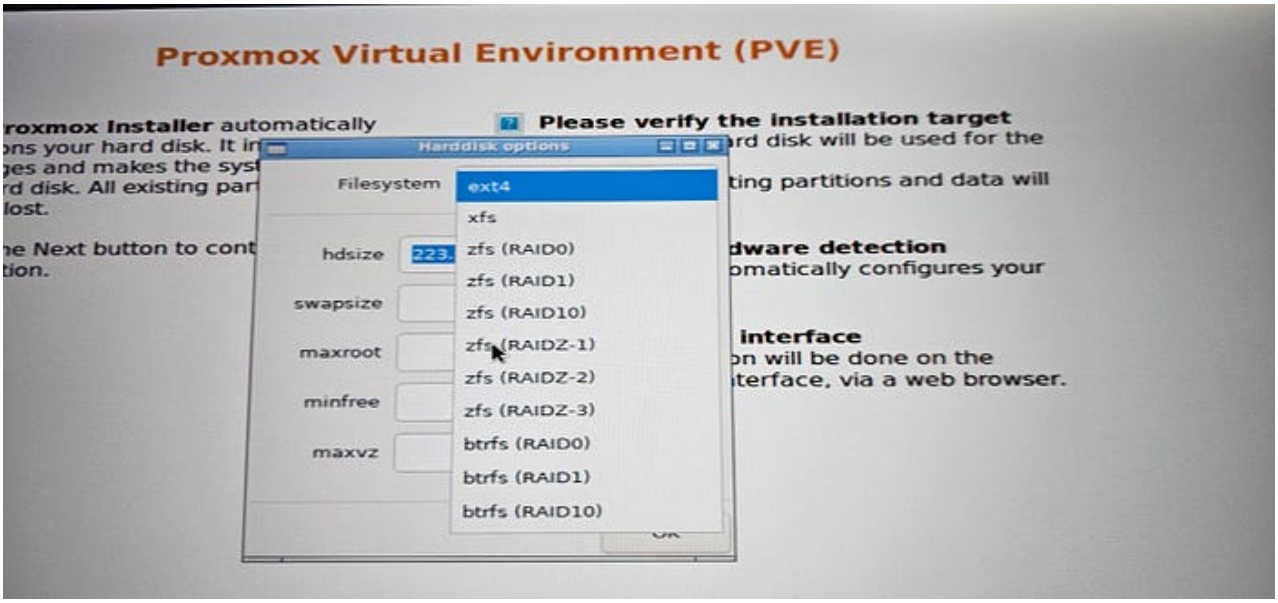
Proxmox Initial Screen

Select disk:

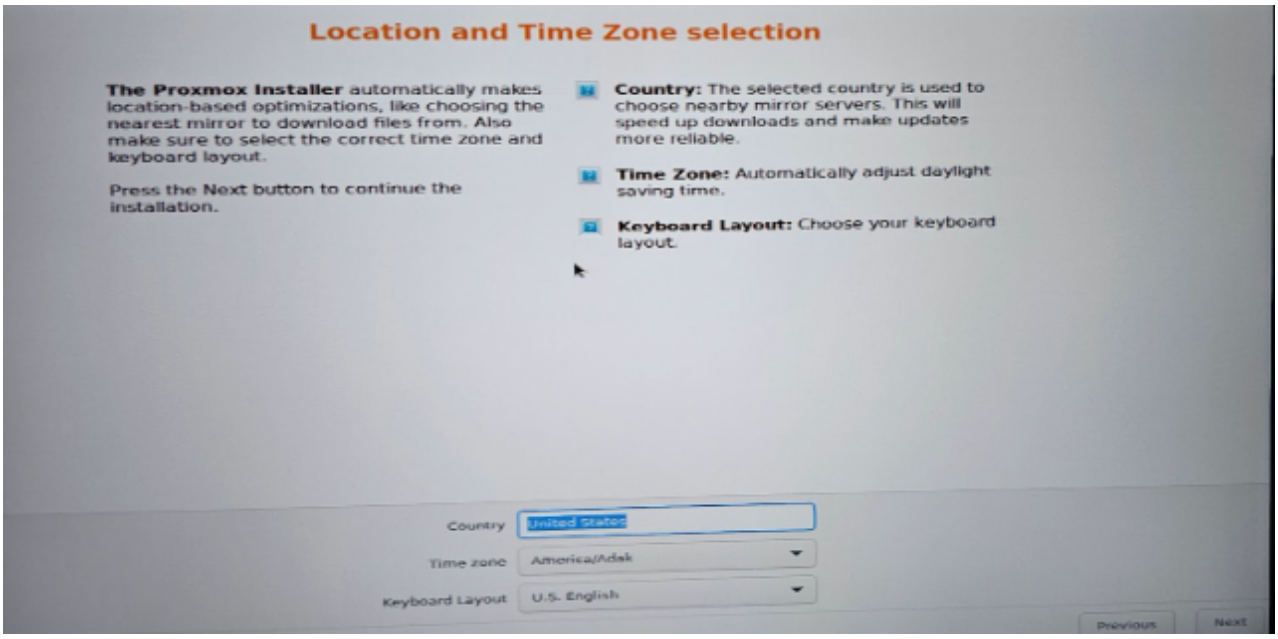


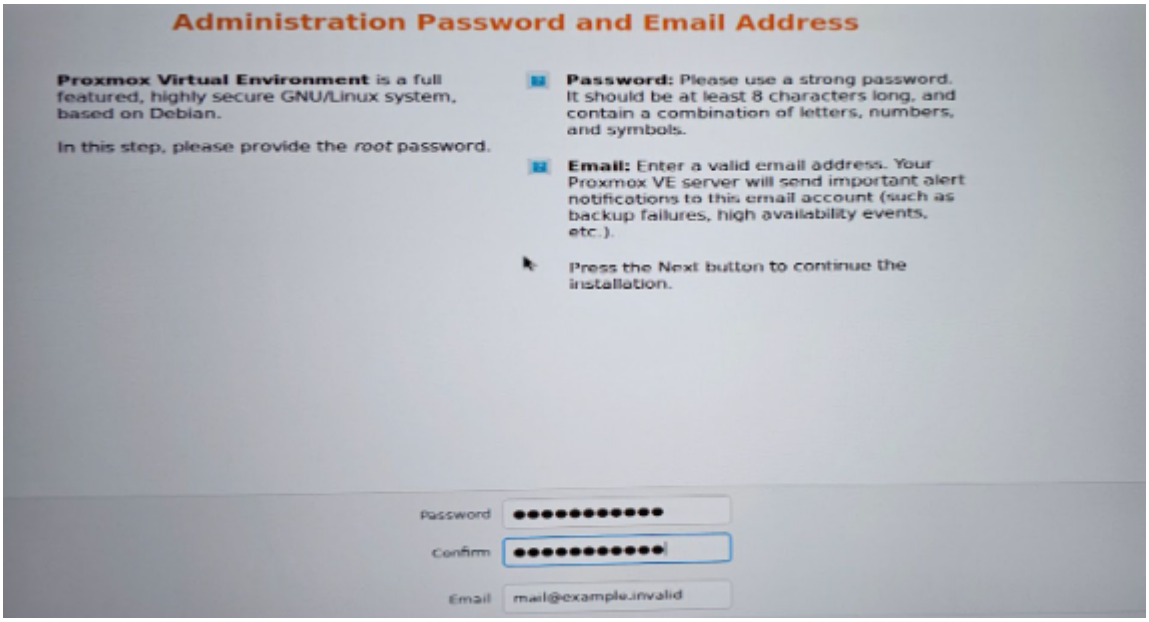
Select Disk for VE installation

NOTE:
At this stage decide what filesystem (ext4/zfs etc.) need to be used for Proxmox OS. I already had Raid 1 at hardware level, so I picked that drive. Otherwise at this step, we can create software based array:



Select Location:

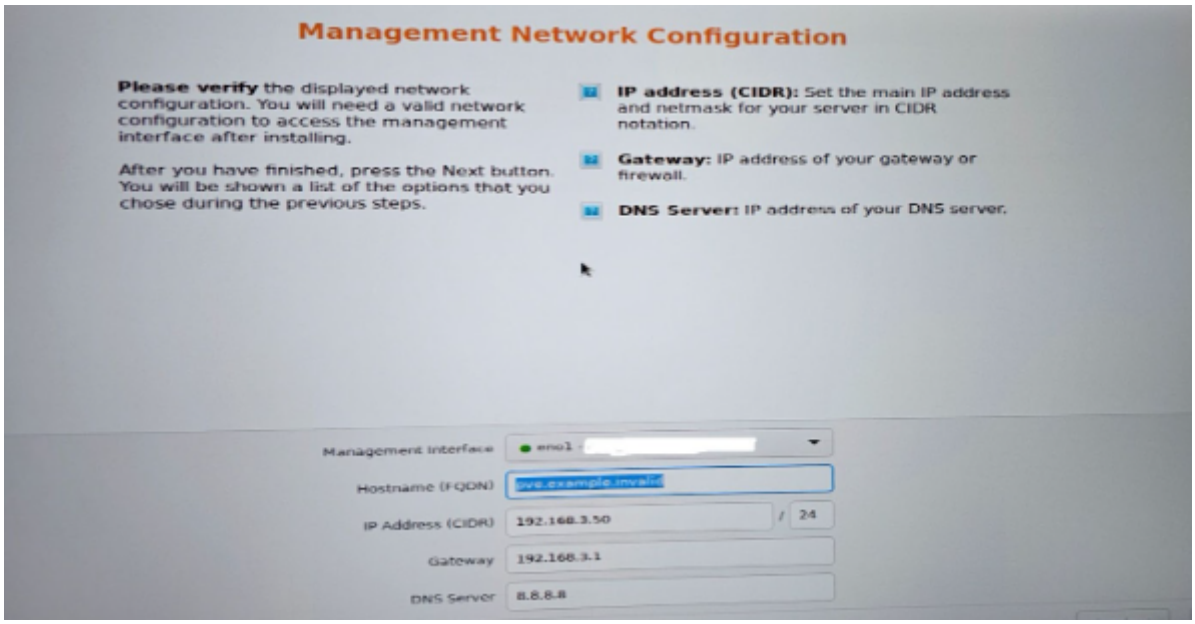




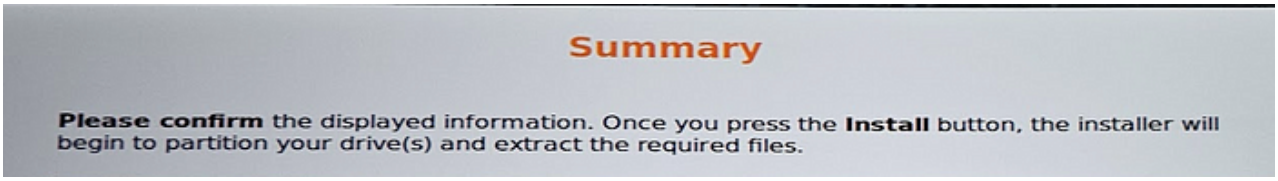
Network

This is a very important step. As I mentioned in my previous post above, I had prepared to have a separate network for management planes and VMs. So, I selected a network interface connected to the management plane. Also, the hostname has to be FQDN. If no name available, we can just use something like prx01.lan or prx01.local

If the DHCP server is in picture, please make sure that IP address given here is outside the range. We need it to be static as changing it later is a bit involved and has to be done at multiple places.

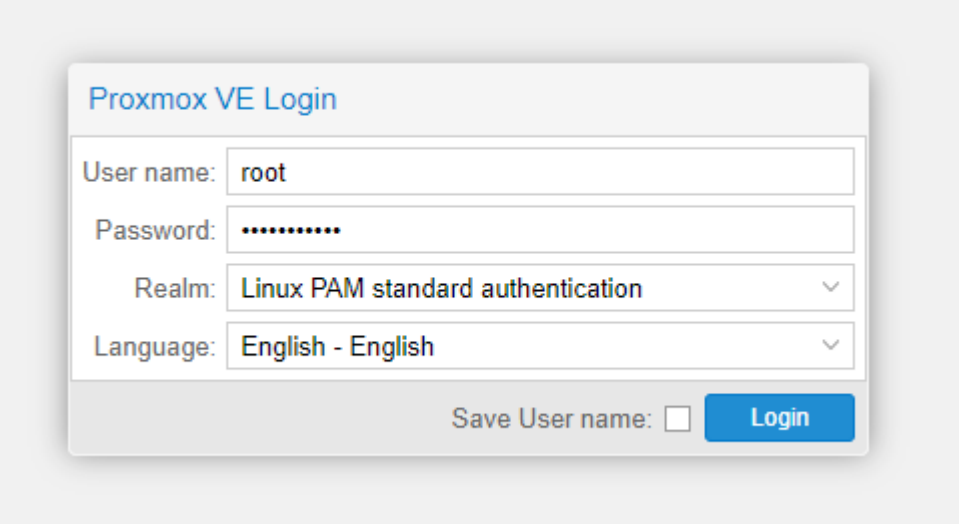


If everything looks good in confirmation screen, we can start installation:



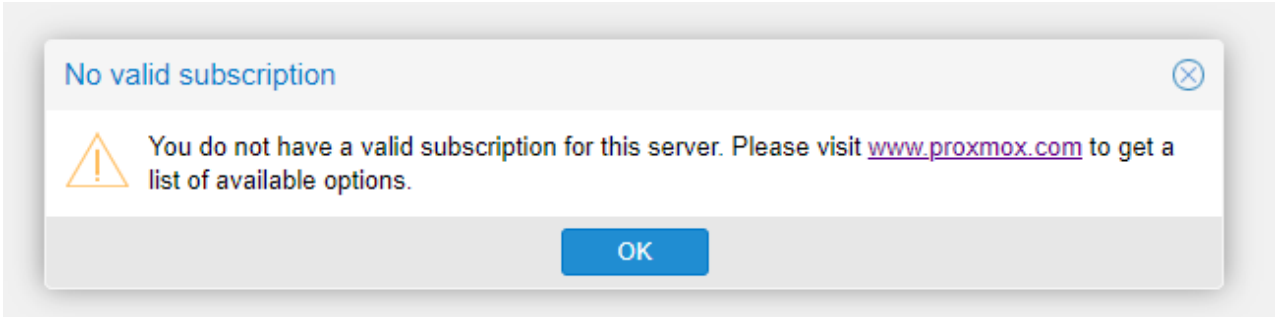
At this step, Proxmox installation will begin. It does not take very long. After installation, remove the USB drive and reboot. After the server reboots, it will show us URL to connect to the web interface. Typically it is at port 8006 of the IP address that we gave in the network screen.

Once we navigate to that url and ignore certificate warning, we are presented with following screen (password that we gave at installation time with root user):

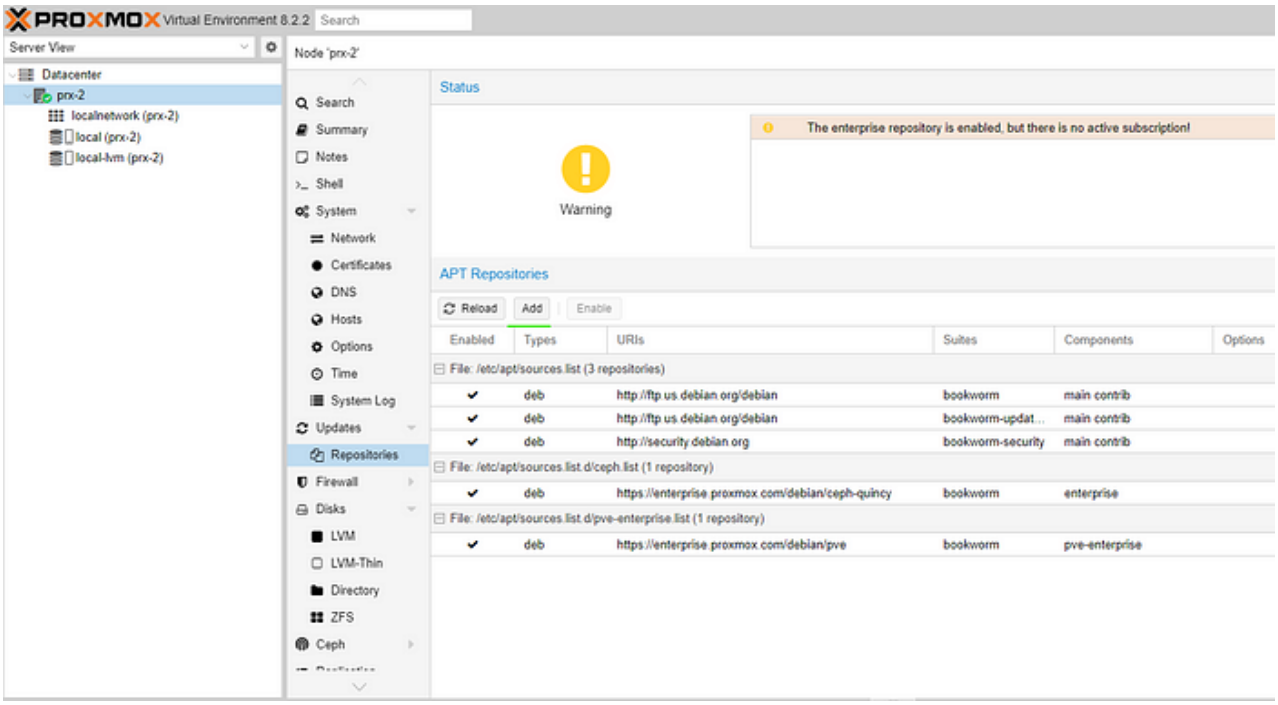


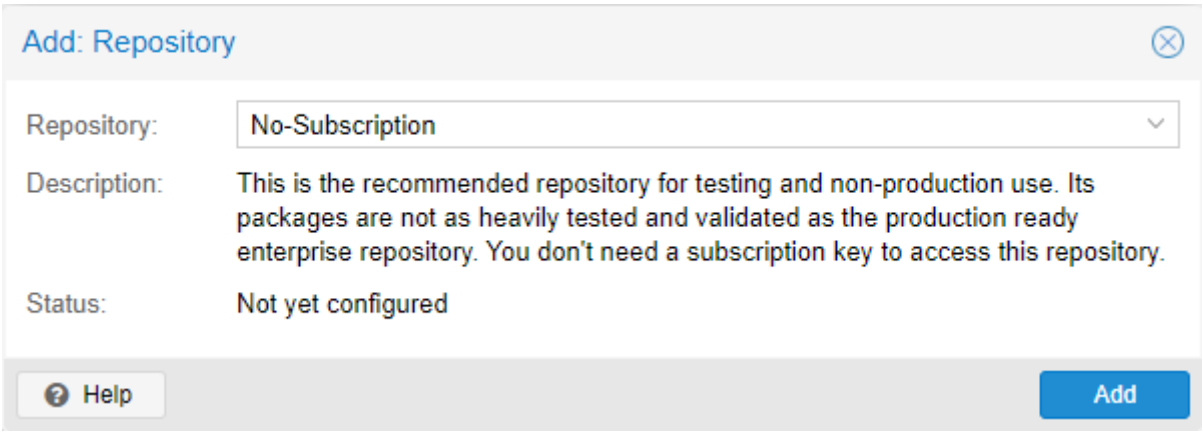
Proxmox Web Management Login

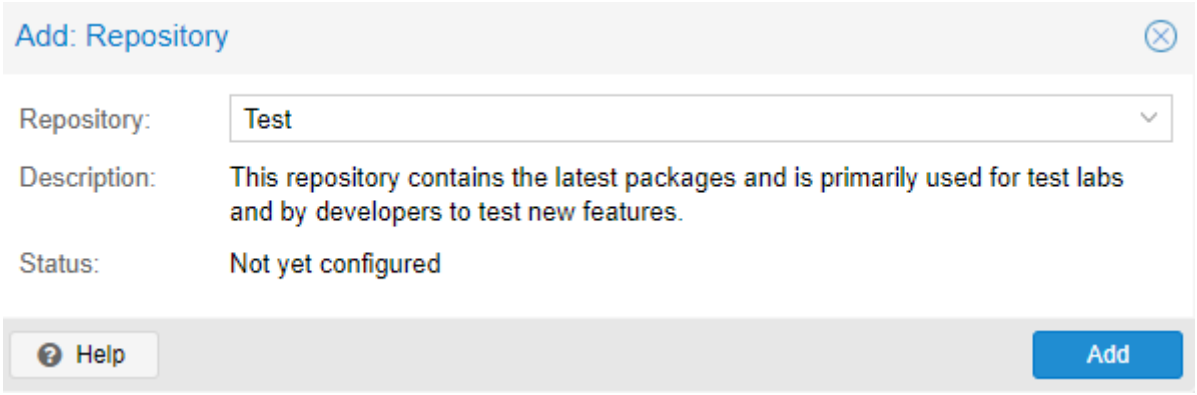
We'll see following message as we do not have paid subscription:



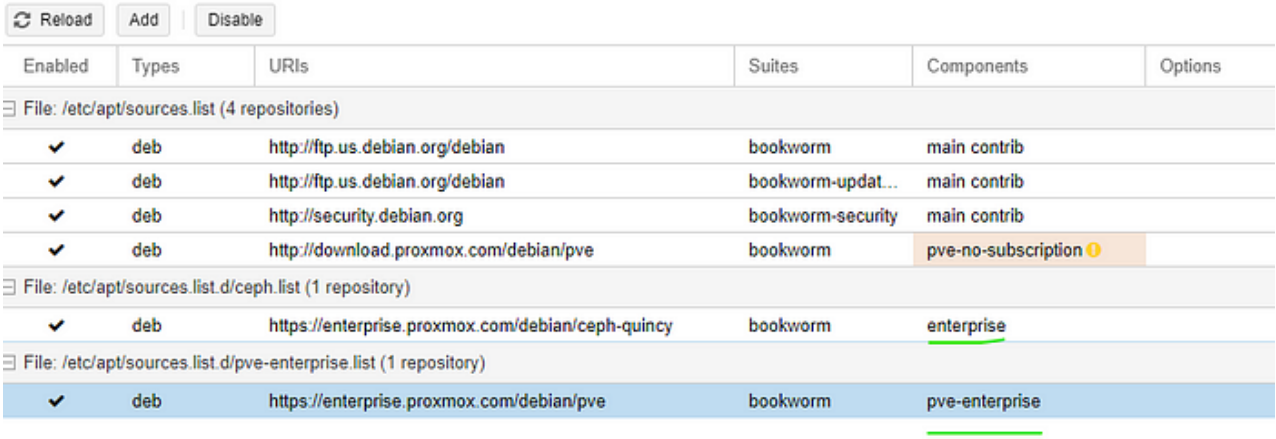
As I did not have plans to purchase for personal use, I added a No Subscription repository. This free repository lets us get security updates. Select the node name and then click on Updates/Repositories:





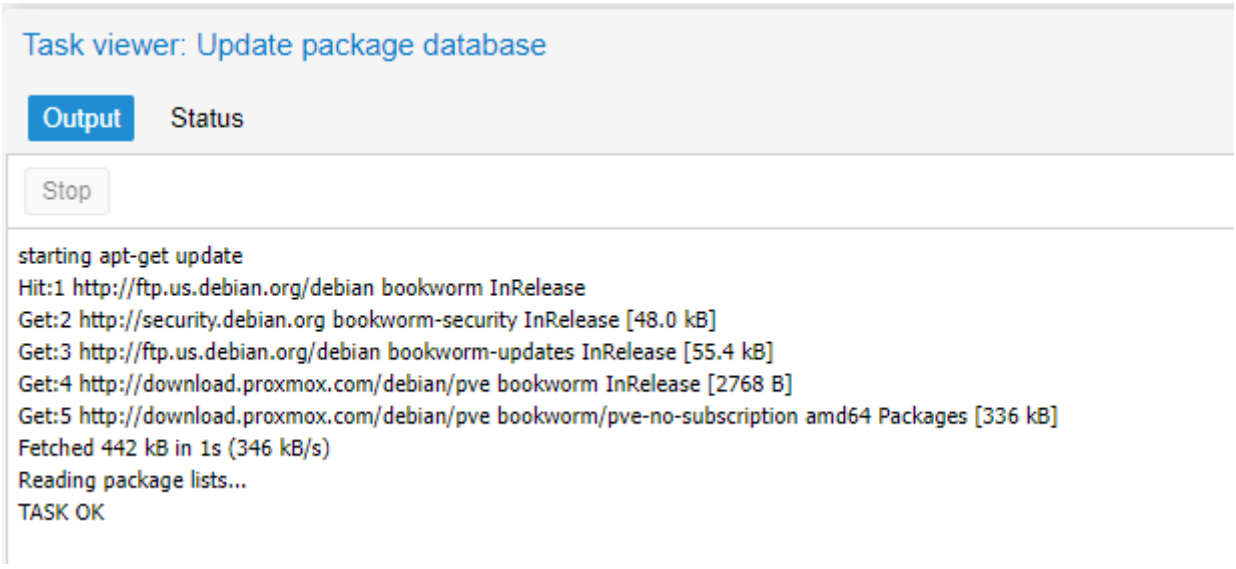


I have not explored Test one yet. Just went with No Subscription one. Disable paid ones.

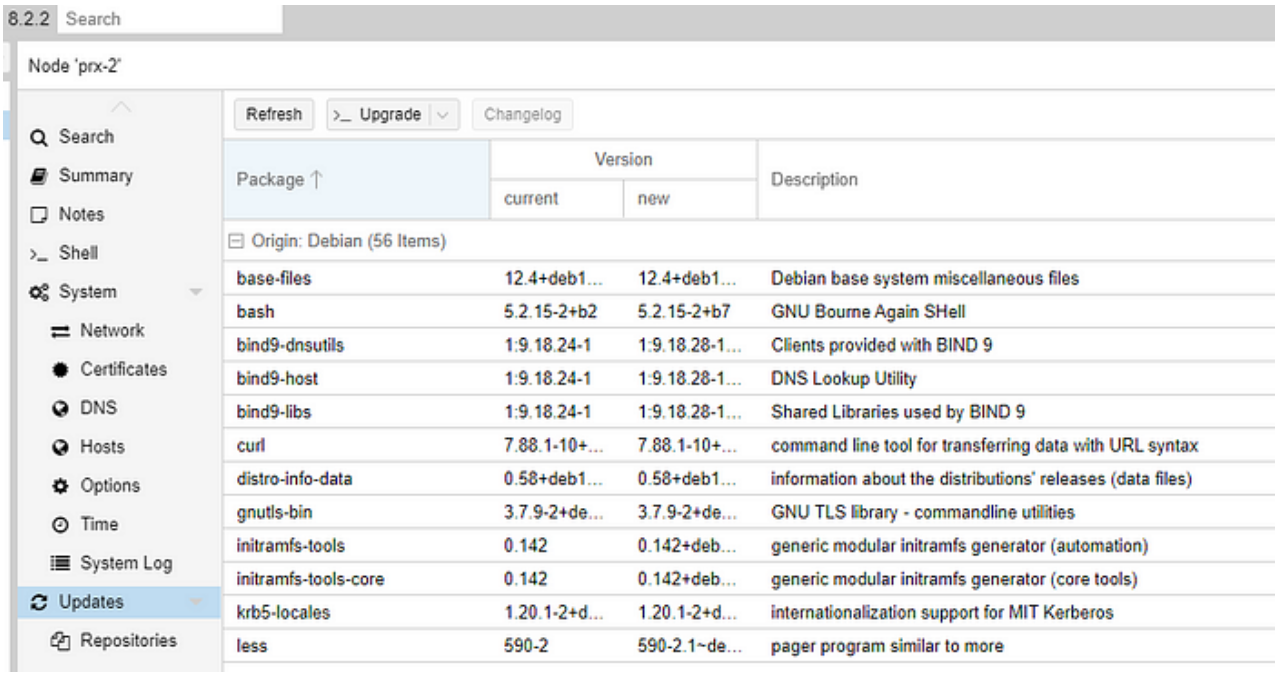


Package Repositories

Now we can go to Updates and that should work:



Once we get the Updates, we can Upgrade:

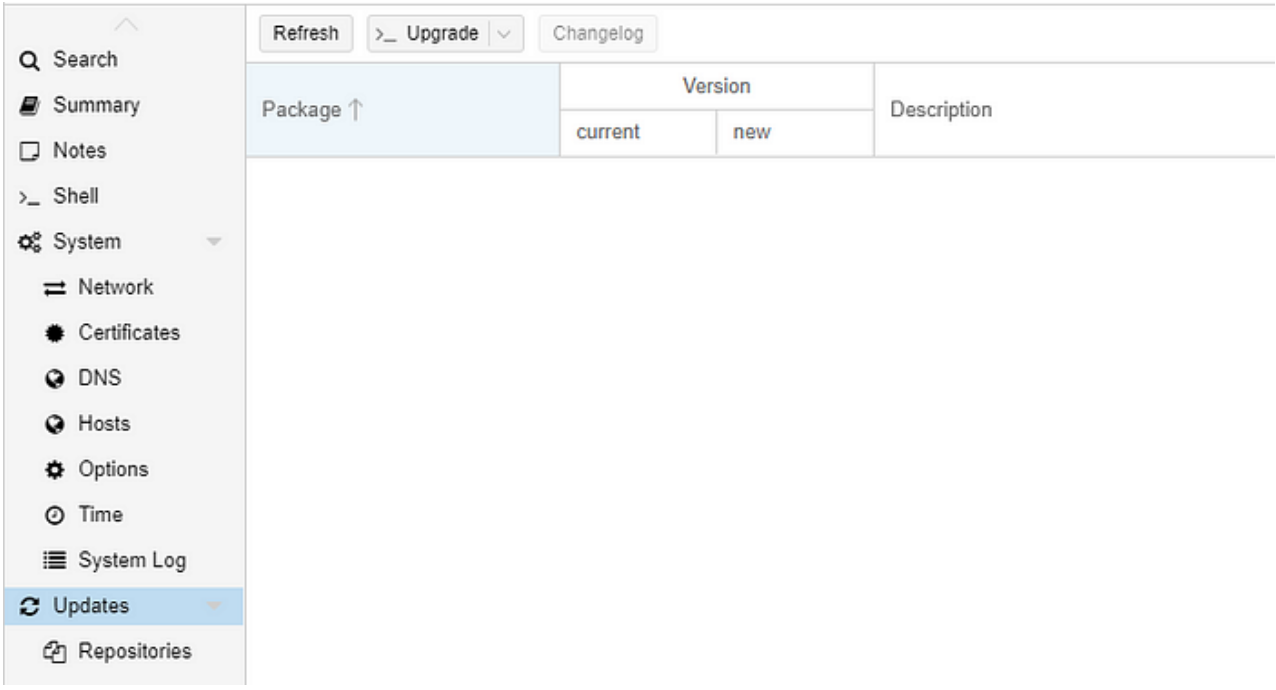



```
Starting system upgrade: apt-get dist-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  proxmox-kernel-6.8.12-1-pve-signed
The following packages will be upgraded:
  base-files bash bind9-dnsutils bind9-host bind9-libs curl distro-info-data
  gnutls-bin ifupdown2 initramfs-tools initramfs-tools-core krb5-locales less
  libarchive13 libc-bin libc-l10n libc6 libcurl3-gnutls libcurl4 libfreetype6
  libglib2.0-0 libgnutls-dane0 libgnutls30 libgnutlsxx30 libgssapi-krb5-2
  libgstreamer-plugins-base1.0-0 libk5crypto3 libkrb5-3 libkrb5support0
  libnss-systemd libnvpair3linux libopeniscsiusr libpam-systemd
  libproxmox-acme-perl libproxmox-acme-plugins libpve-cluster-api-perl
  libpve-cluster-perl libpve-common-perl libpve-guest-common-perl
  libpve-notify-perl libpve-rs-perl libpve-storage-perl libpython3.11-minimal
  libpython3.11-stdlib libqt5core5a libqt5dbus5 libqt5network5 libseccomp2 libssl3
  libsystemd-shared libsystemd0 libudev1 libuutil3linux libzfs4linux libzpool5linux
  locales nano open-iscsi openssh-client openssh-server openssh-sftp-server openssl
  postfix proxmox-backup-client proxmox-backup-file-restore proxmox-firewall
  proxmox-kernel-6.8 proxmox-termproxy proxmox-widget-toolkit pve-cluster
  pve-container pve-docs pve-esxi-import-tools pve-firewall pve-firmware
  pve-ha-manager pve-manager pve-qemu-kvm python3-idna python3.11
  python3.11-minimal qemu-server shim-helpers-amd64-signed shim-signed
  shim-signed-common shim-unsigned spl ssh systemd systemd-boot systemd-boot-efi
  systemd-sysv udev zfs-initramfs zfs-zed zfsutils-linux
96 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 336 MB of archives.
After this operation, 570 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

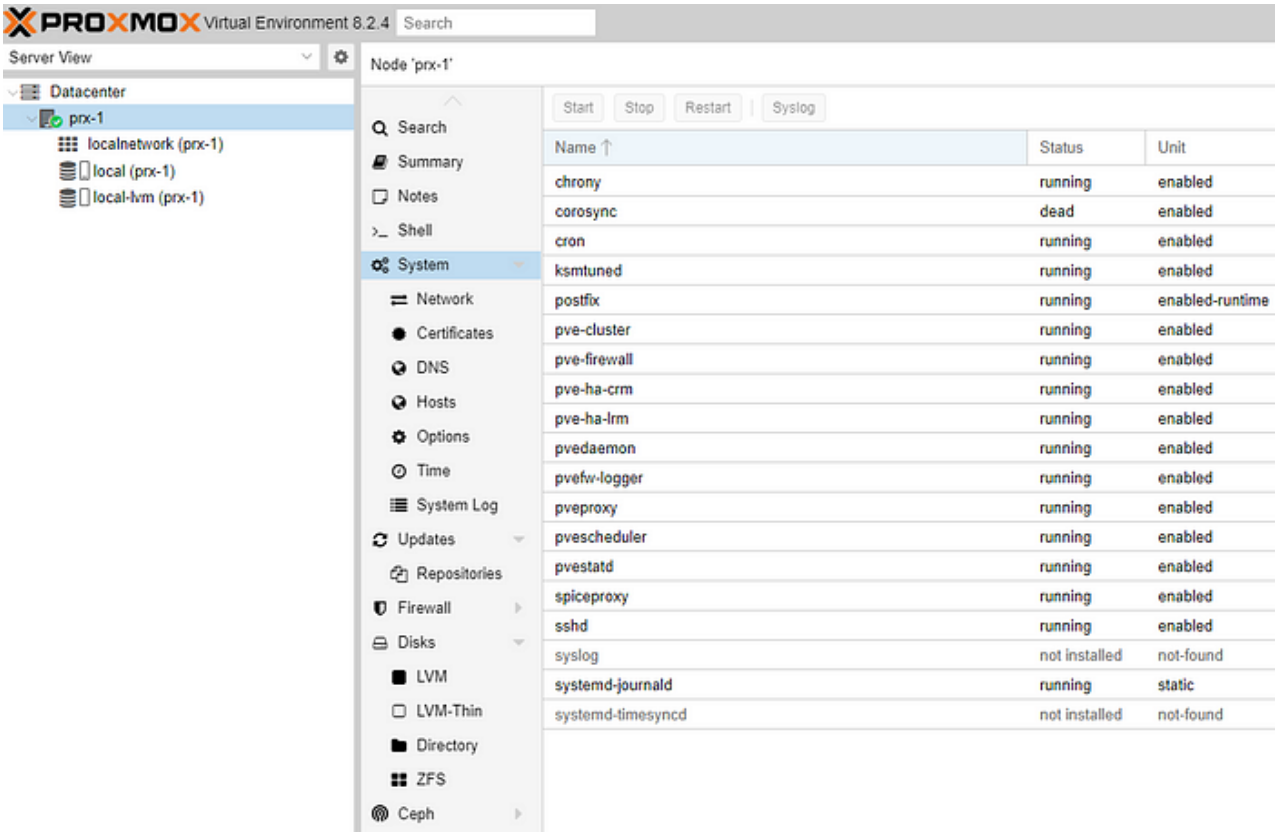
Normally if I was already running workloads, I would check individual updates and their possible impact but as this is a fresh install, I would just Upgrade all. In fact, at the end it did tell me to reboot because of some update:

```
Your System is up-to-date


Seems you installed a kernel update - Please consider rebooting
this node to activate the new kernel.

starting shell
```

After reboot, I checked again and had no more Updates to install:

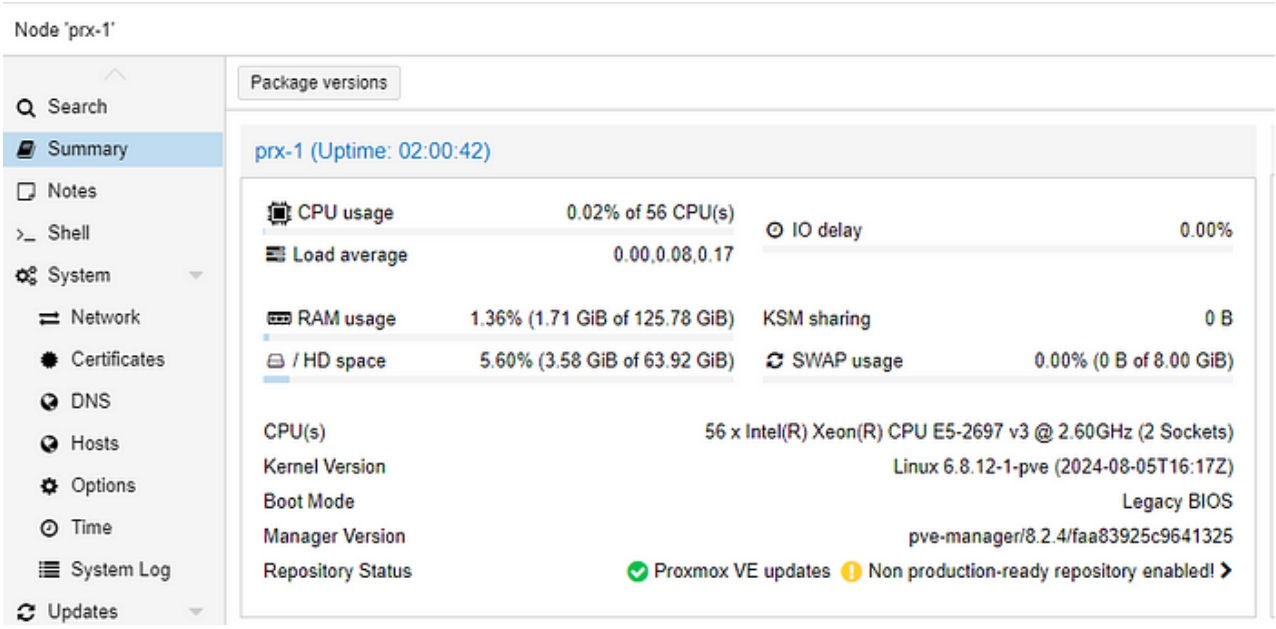


At this stage, I had launched my first node for the cluster I was going to set up:



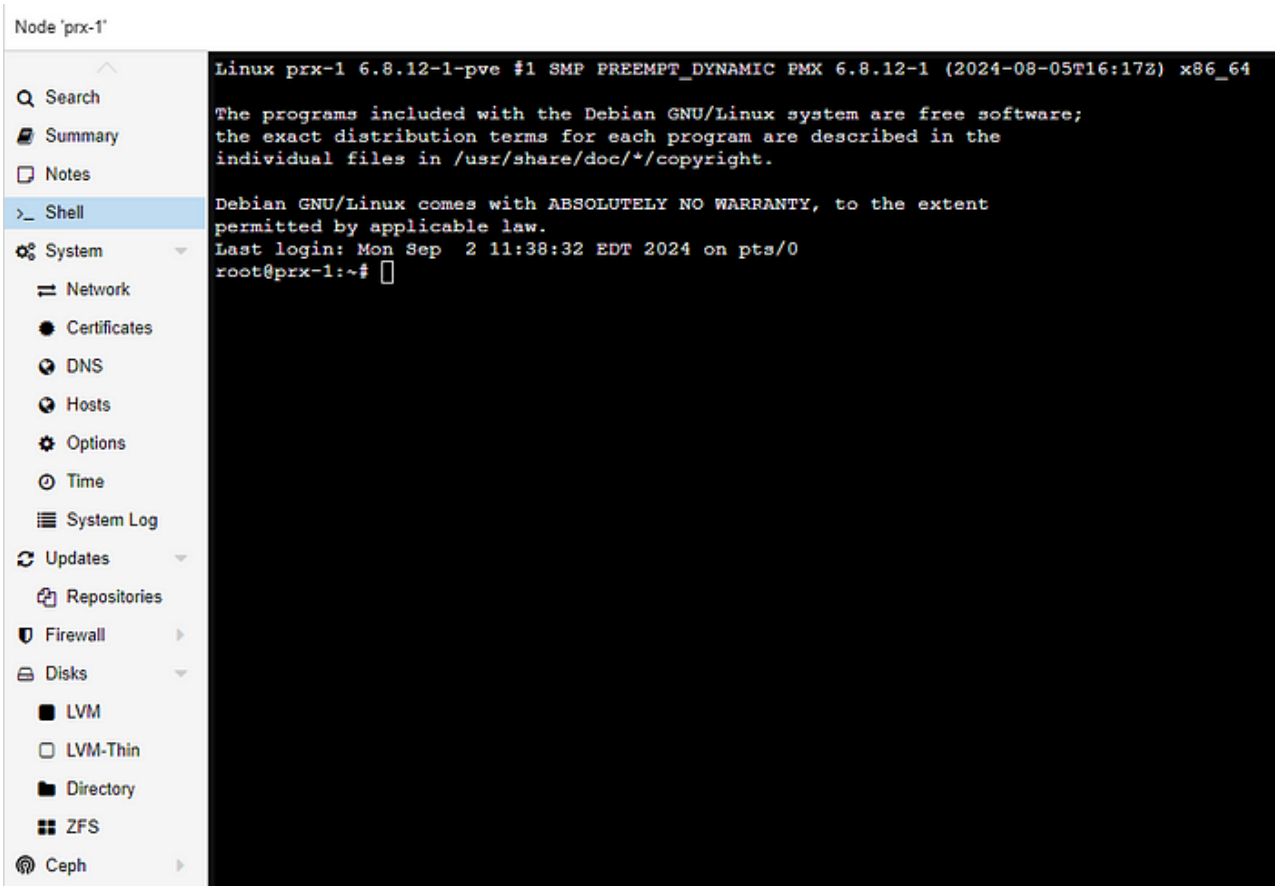
First Proxmox Node

Summary view will show resource consumption. As we can see Proxmox itself has very small footprint:



Resource Consumption

Before moving on, it is really a good idea to explore and be comfortable with most options in the following screen like network, dns, sys logs, disks etc.:



In my case, clicking on Disks got me following:

Reload | Show S.M.A.R.T. values | Initialize Disk with GPT | Wipe Disk

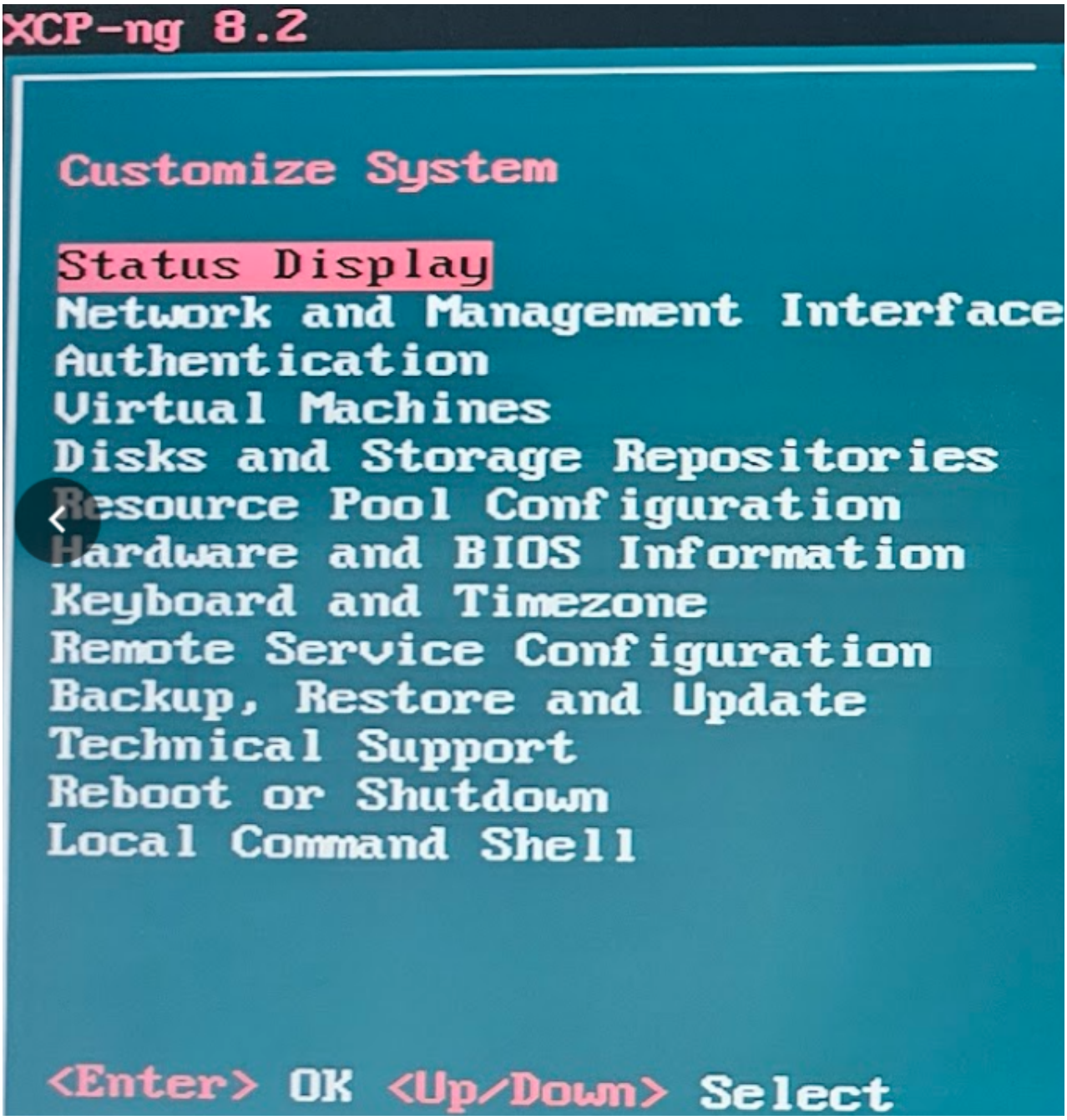
Device	Type	Usage	Size	GPT	Model
/dev/sda	unknown	partitions	239.44 GB	Yes	PERC_H730P_Mini
/dev/sda1	partition	BIOS boot	1.03 MB	Yes	
/dev/sda2	partition	EFI	1.07 GB	Yes	
/dev/sda3	partition	LVM	238.37 GB	Yes	
/dev/sdb	unknown	No	479.56 GB	No	PERC_H730P_Mini
/dev/sdc	unknown	No	479.56 GB	No	PERC_H730P_Mini

Networking shows following as we have not configured second bridge yet:

Create | Revert | Edit | Remove | Apply Configuration

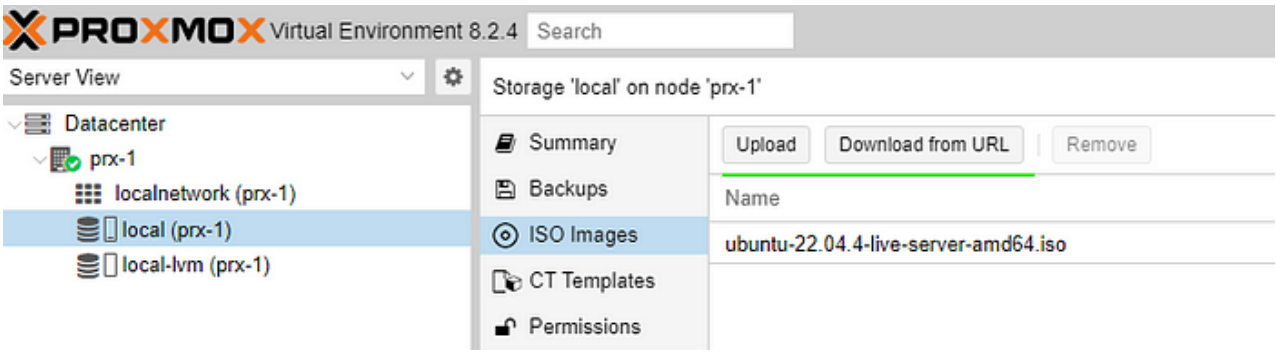
Name	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR
eno1	Network Device	Yes	No	No			
eno2	Network Device	No	No	No			
eno3	Network Device	No	No	No			
eno4	Network Device	No	No	No			
vmbr0	Linux Bridge	Yes	Yes	No	eno1		192.168.3.10/24

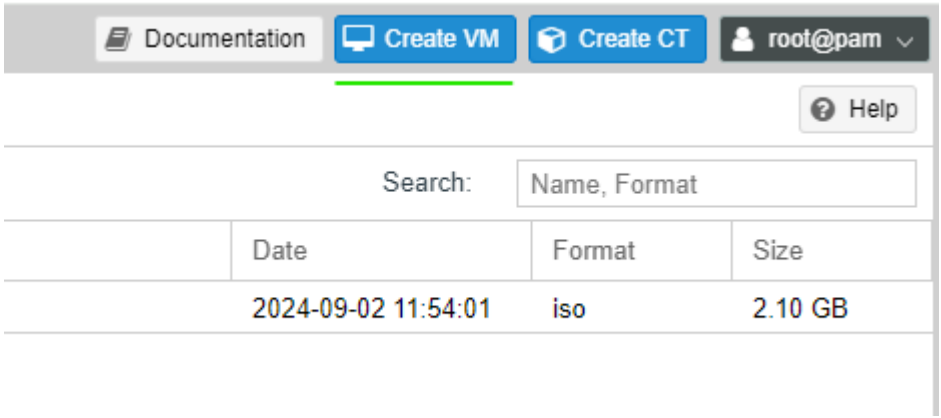
As I had mentioned in previous post, xcp-ng gives most of functionality of web interface on console itself:



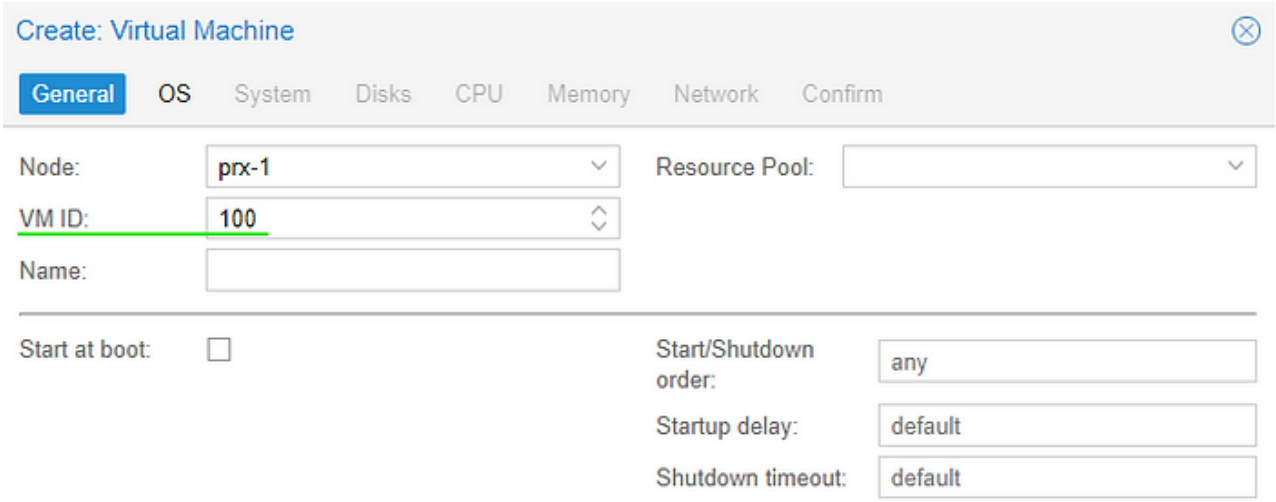
But in Proxmox, we do not have that. It is just a shell. On the other hand, using commands on shell, we can do whatever we could do using the web interface. I have not run into any scenario yet where I would miss that console functionality. I still have access through the web interface and through ssh connection to shell and using commands there.

Before wrapping up, what good is a hypervisor if we do not launch a VM there? But first we need an image to launch a VM from. We can either Upload from local machine or just give it direct web url to download image from on following screen:

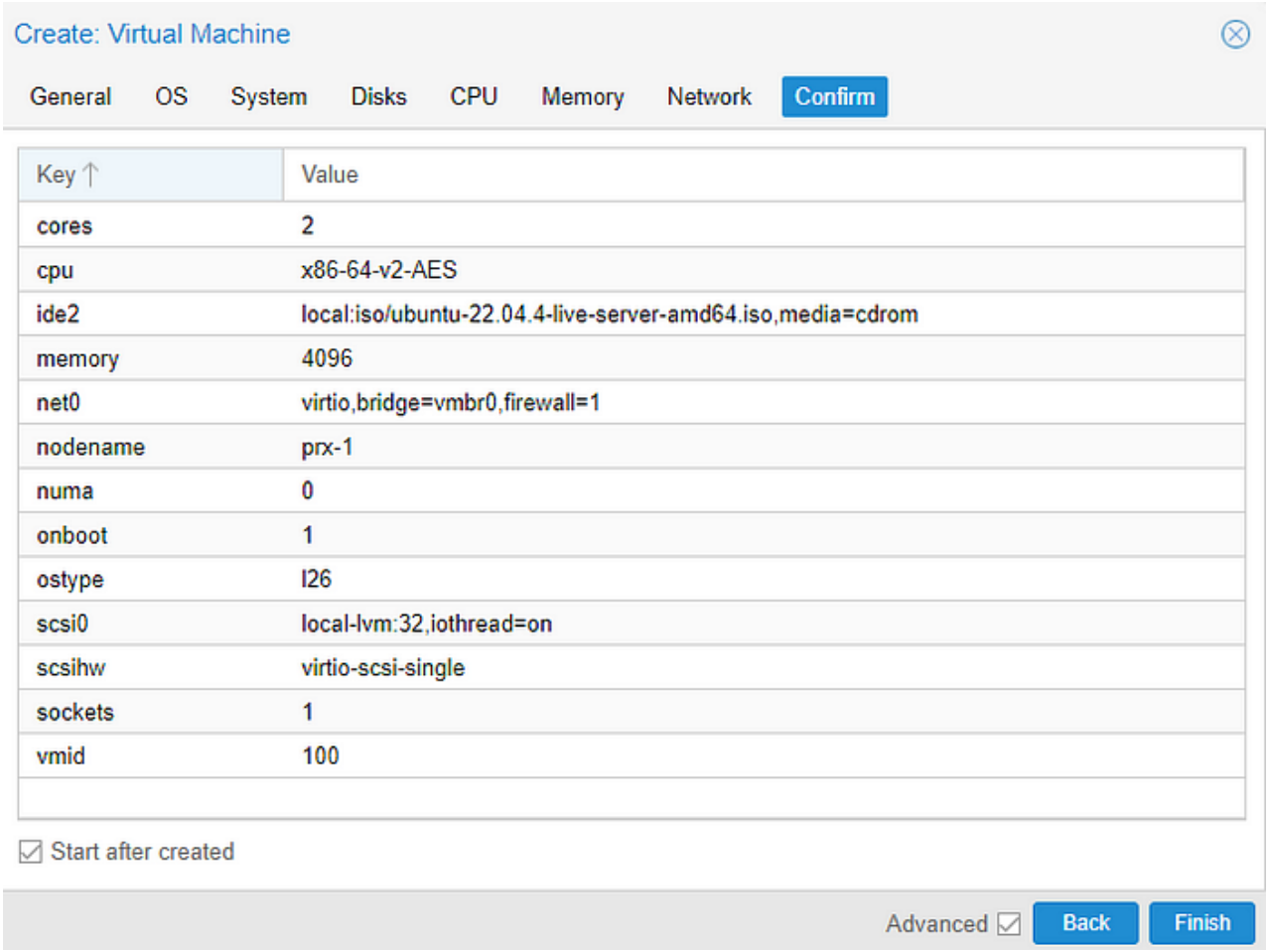




VM Id has to be unique but it can be anything:

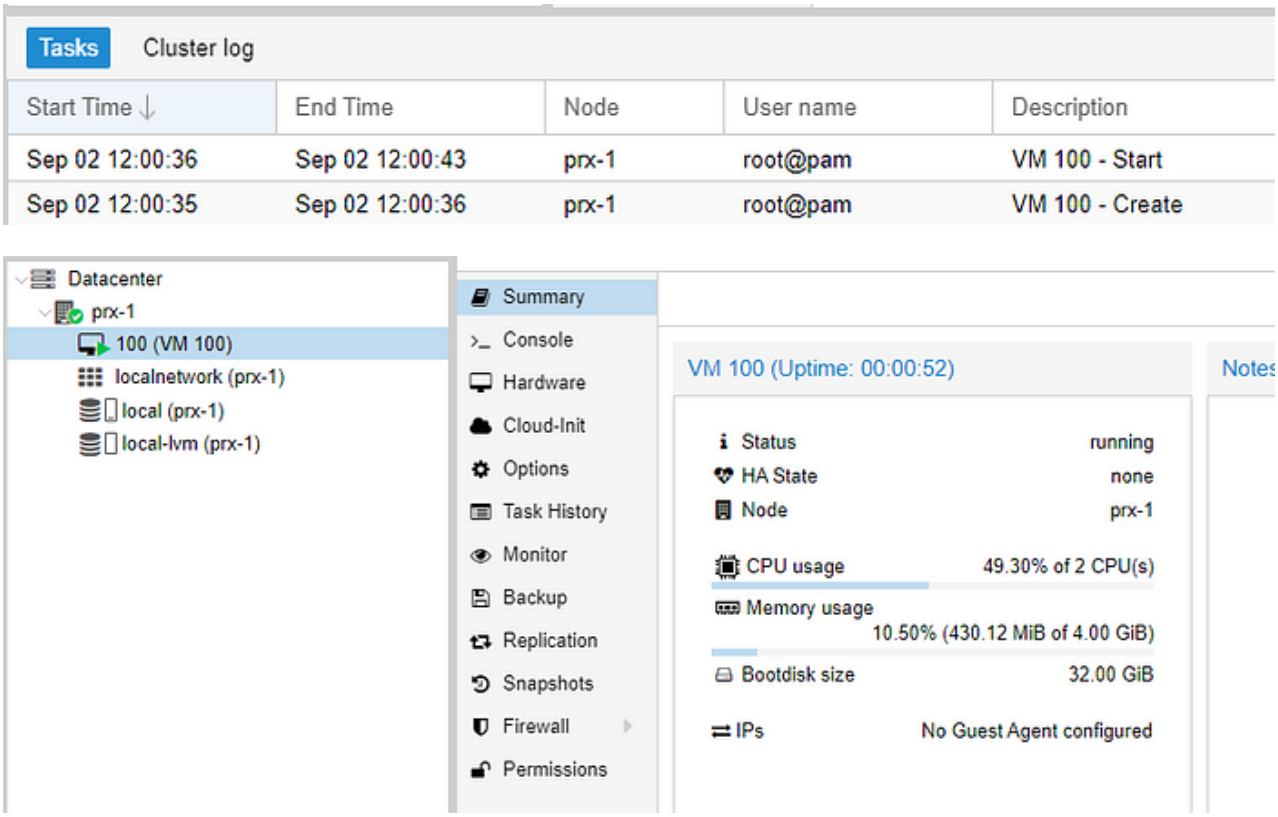


After going through each screen, ended up with following summary:

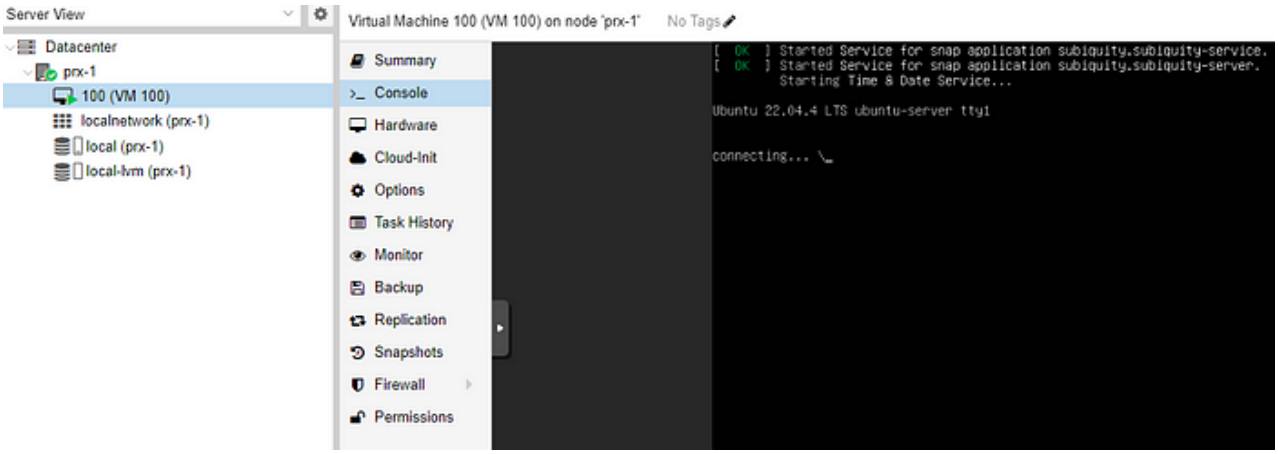


Summary screen for VM creation

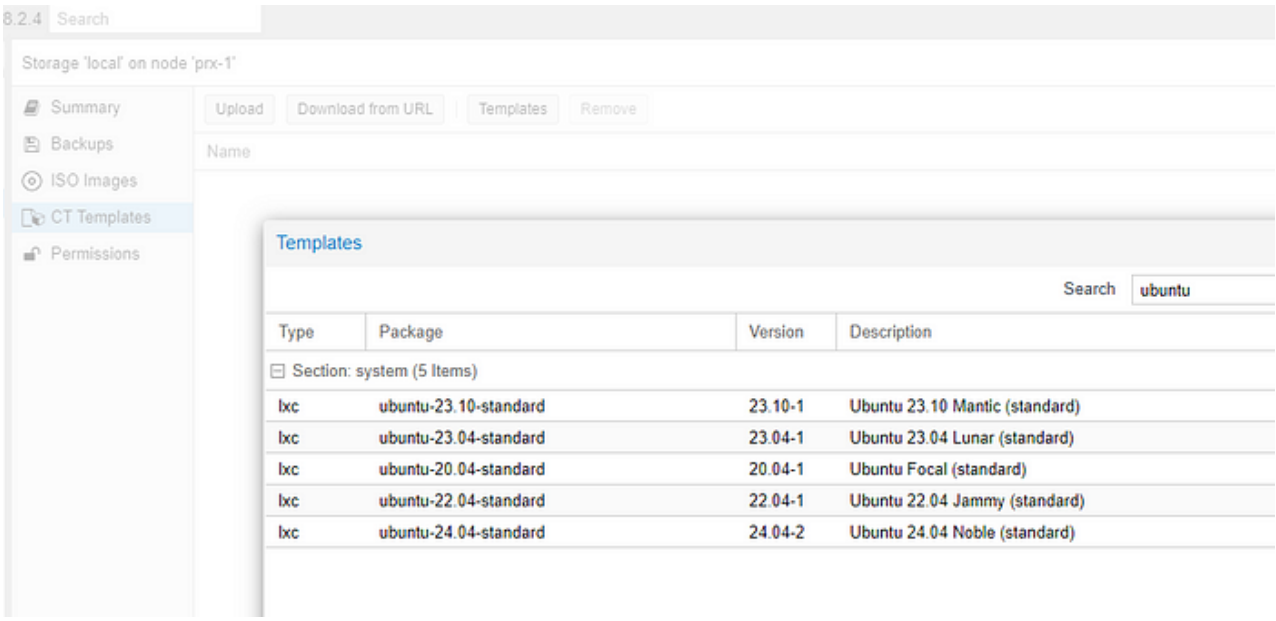
Logs at bottom of screen shows:



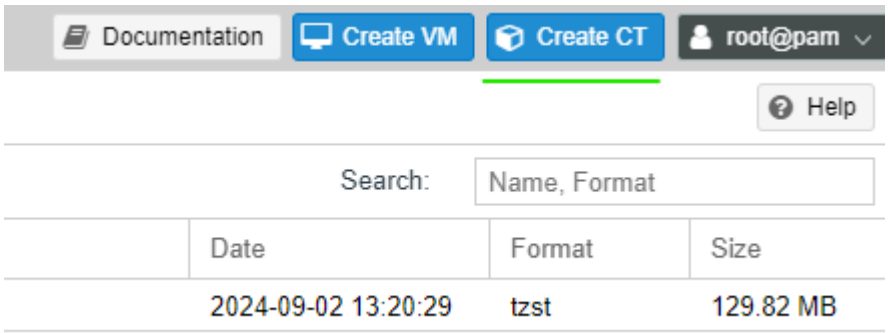
Once it is started, we can connect to console and do a typical Ubuntu installation (as that is what our ISO was):



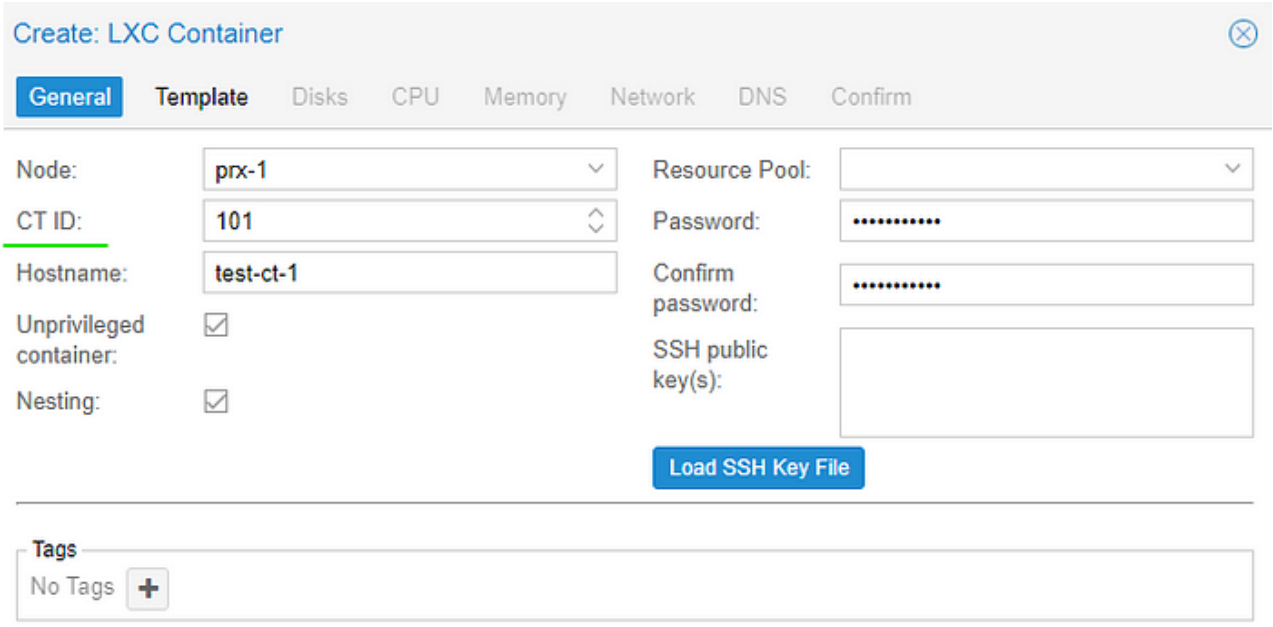
We can also create containers directly on Proxmox. First, we need to get Templates:



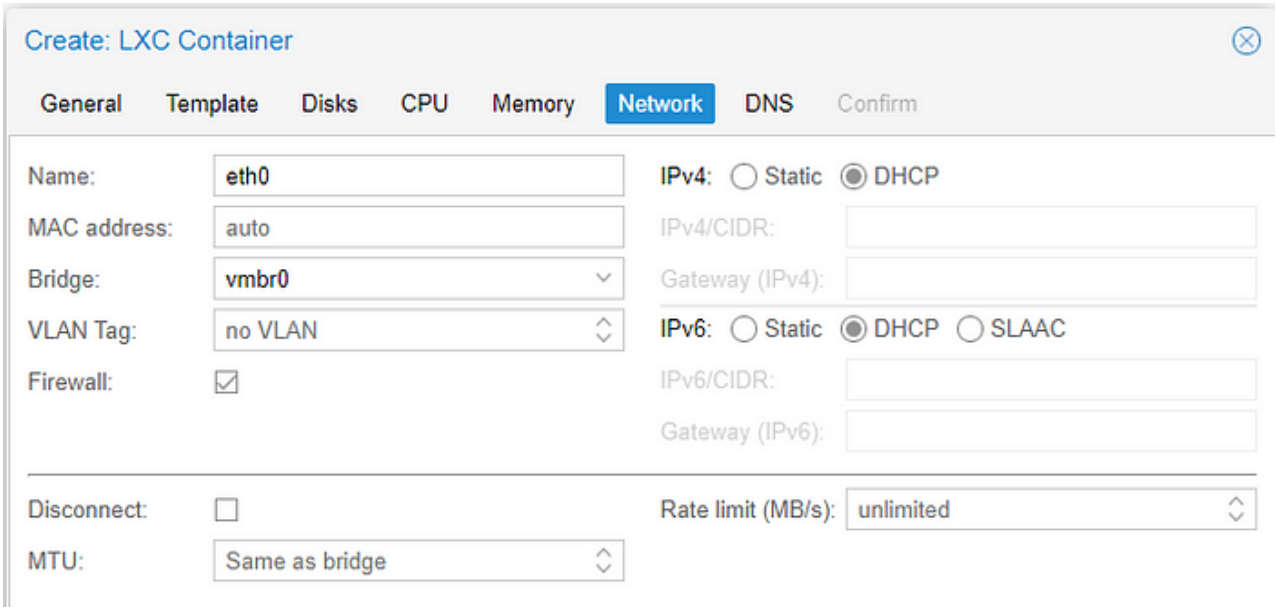
Now, let us create a container:

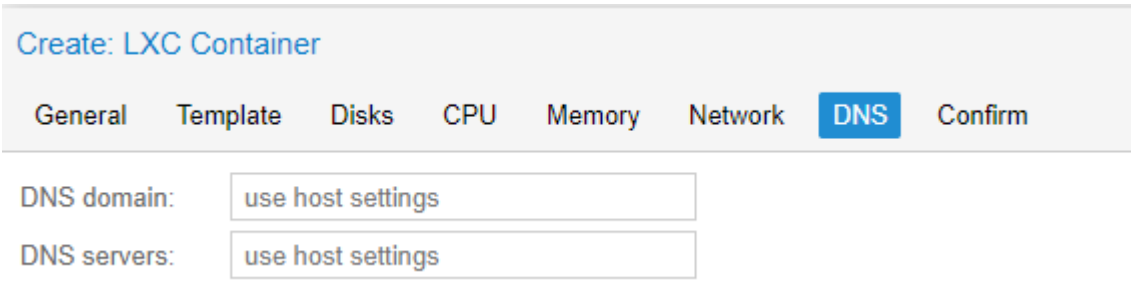


Create Container

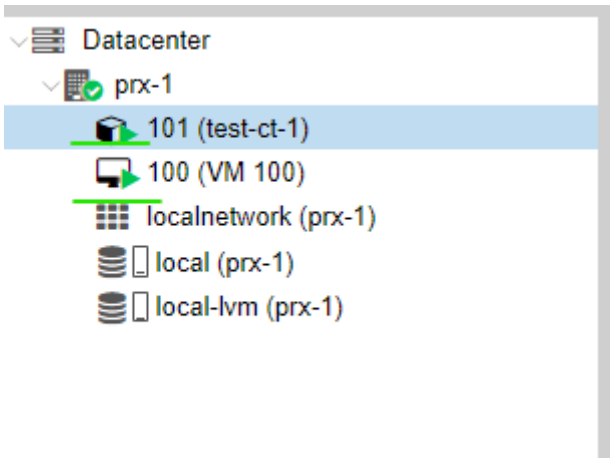


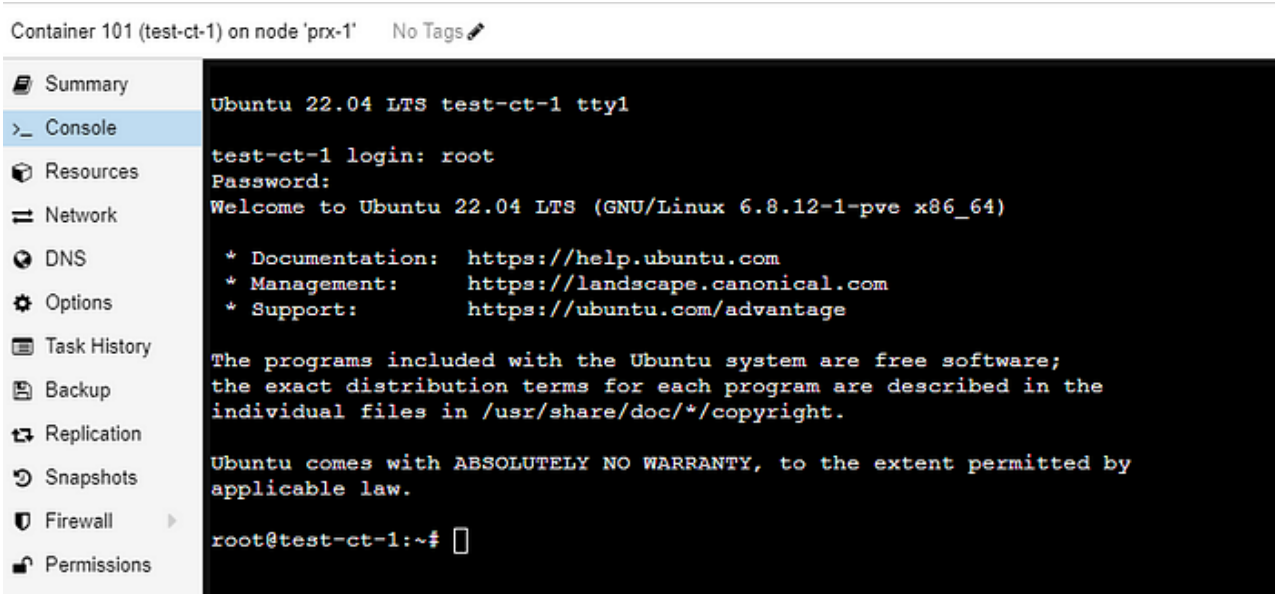
CT ID and VM ID are not separate pools, ID needs to be unique across VMs and Containers.





Also, containers will come up much faster than a VM. Login with user ‘root’ and password that was selected on first screen:





NOTE: Unlike containers in Kubernetes, these containers are on the same network (because we picked that in the wizard above). So, if we check the IP address of the container, we would see that it belongs to the same DHCP range that the host is on and can be accessed directly. So, it works just like VMs in that sense.

There is still a lot to do for storage, networking, cluster and HA. That will be covered in a future post. In the meantime I'll set up other nodes to be added to the cluster.

Thanks!