CO513: Advanced Computer Communication Networks - Lab 01

E/19/446: Wijerathna I.M.K.D.I.

04/05/2025

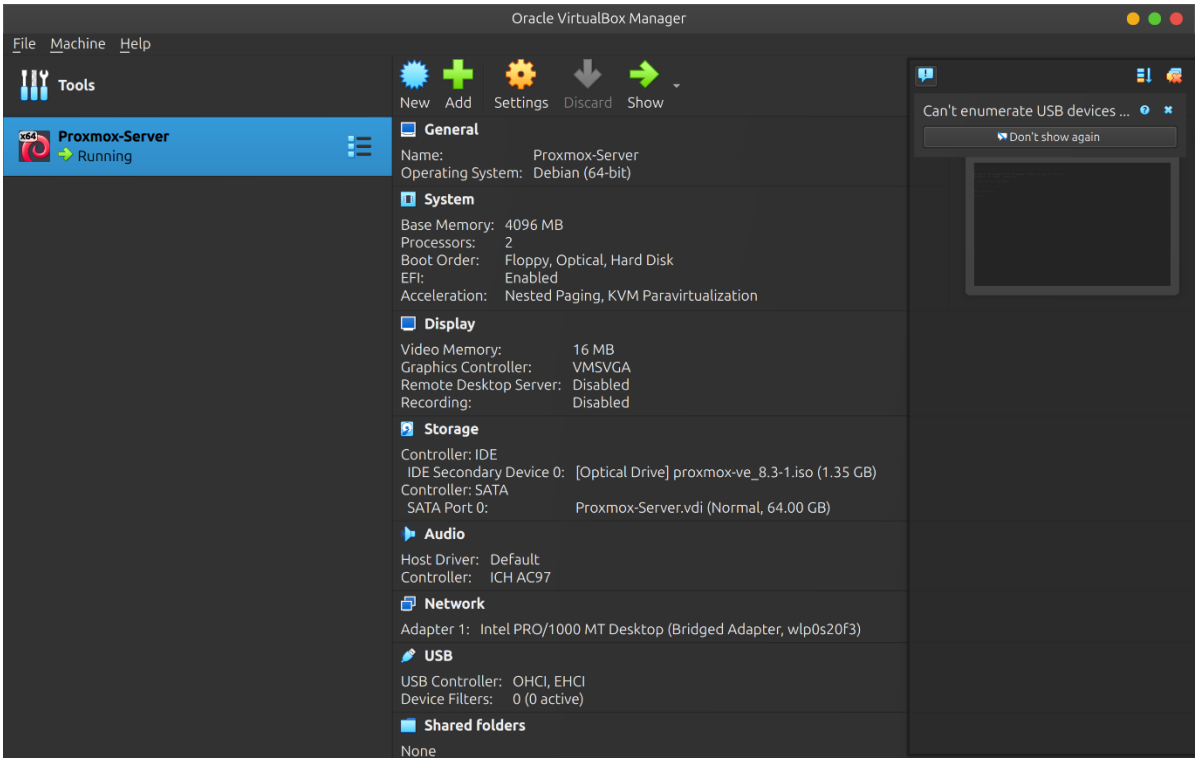
Installation Steps Summary

To install Proxmox VE, I began by downloading and installing VirtualBox 7.1 on my laptop. I then downloaded the latest Proxmox VE ISO and created a virtual machine with 8 GB RAM, 64 GB disk space, 2 CPUs, and bridged networking. After mounting the ISO, I proceeded with the Proxmox installation, configuring the timezone, root password, and network settings. Once installation was complete, I rebooted the VM to access the Proxmox web interface using the assigned IP address.

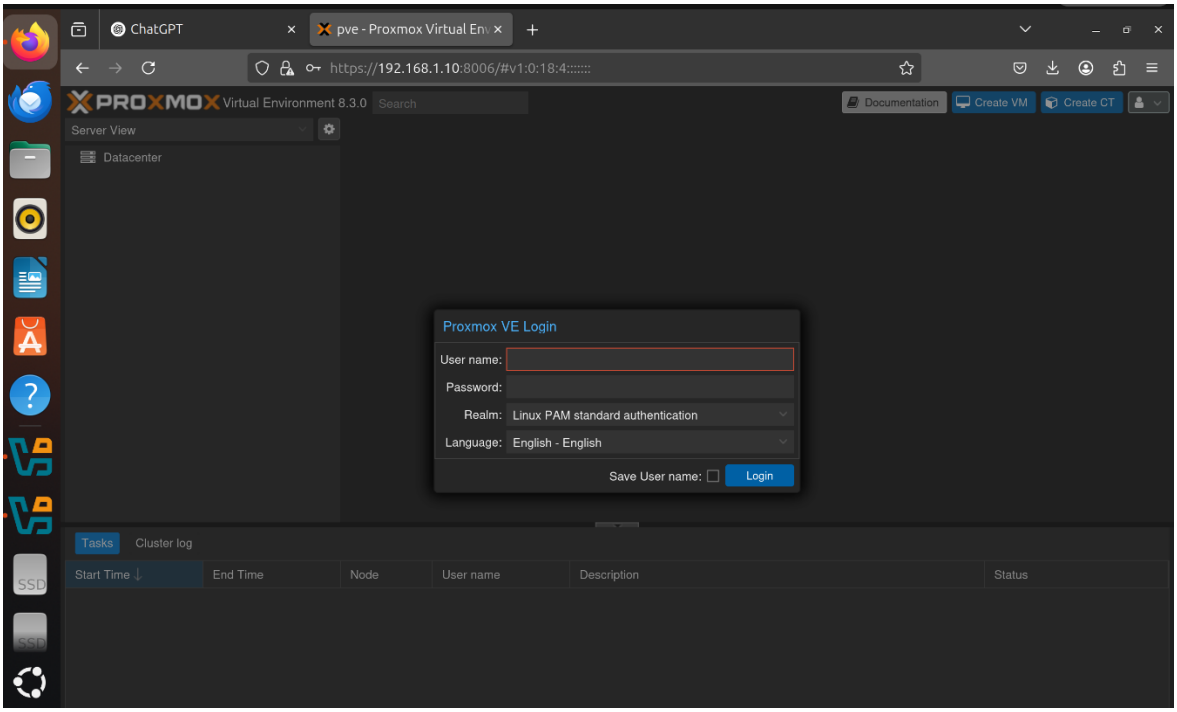
I did encounter a network-related issue during the process, as Proxmox by default configured a static IP and gateway during installation process, which caused connectivity problems when switching between networks (ie: from Uni WIFI network to Home WIFI). This was resolved by editing the /etc/network/interfaces file inside the Proxmox VM to enable DHCP, allowing it to obtain a valid IP address dynamically.

Screenshots

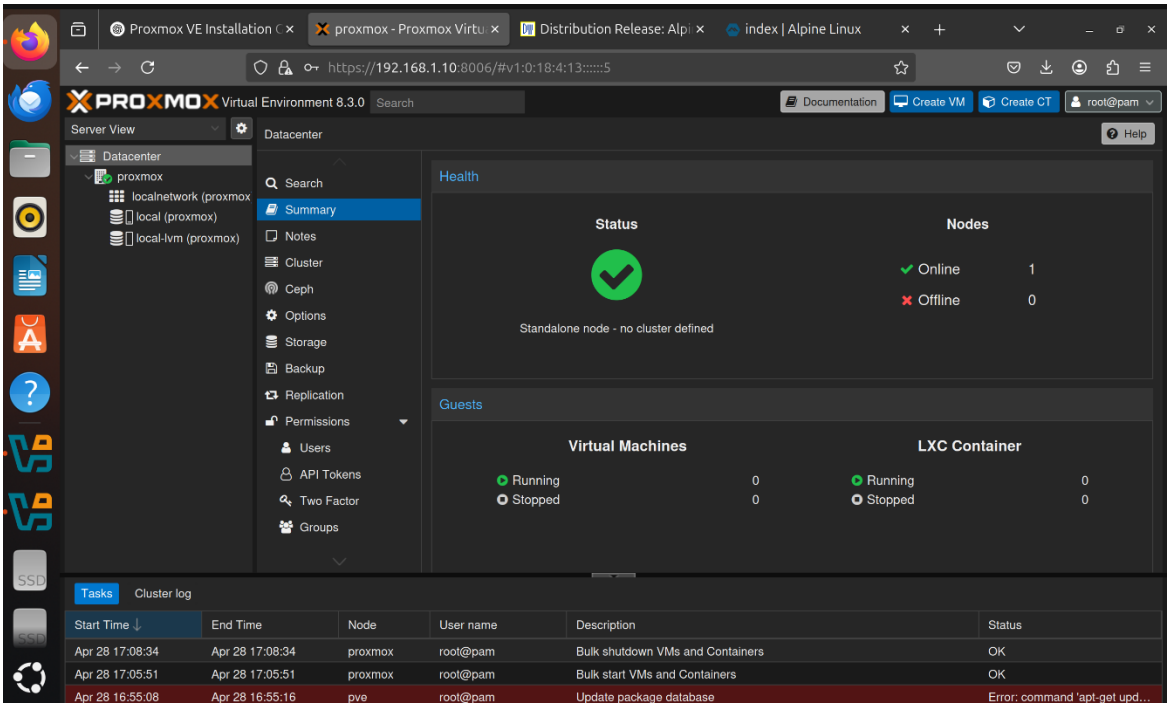
- VirtualBox settings for the Proxmox VM.



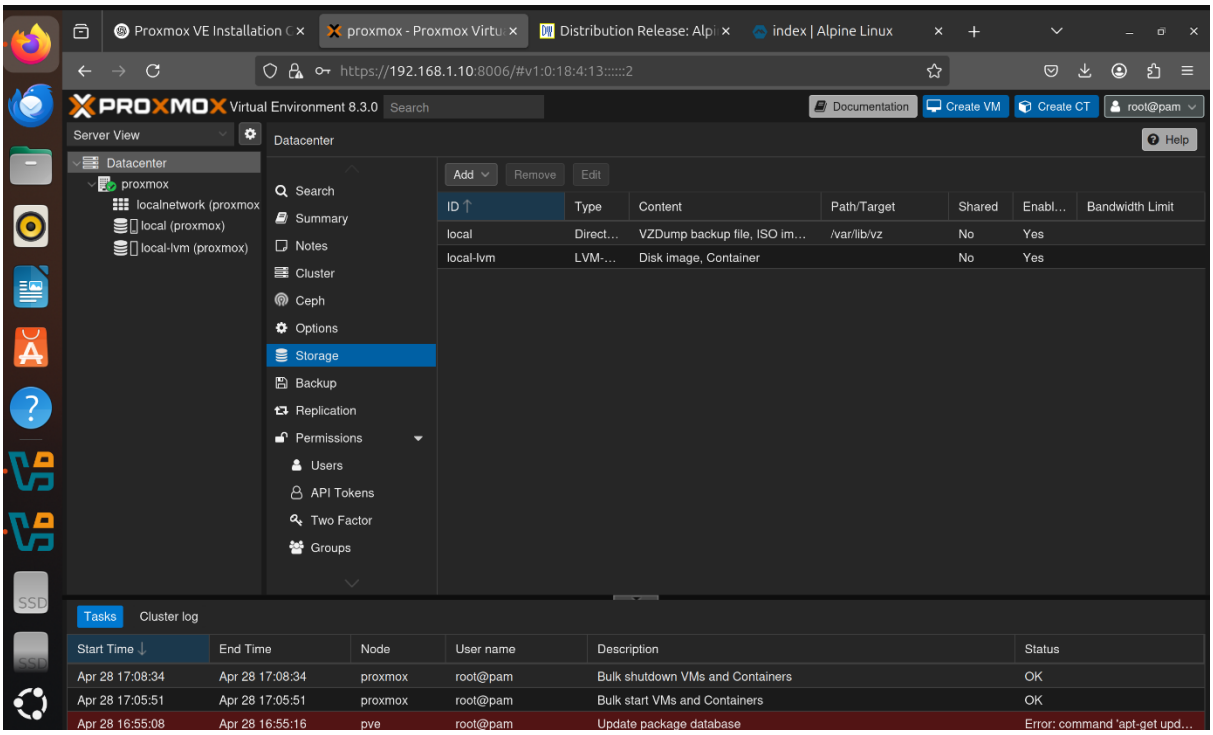
- Proxmox web login page.



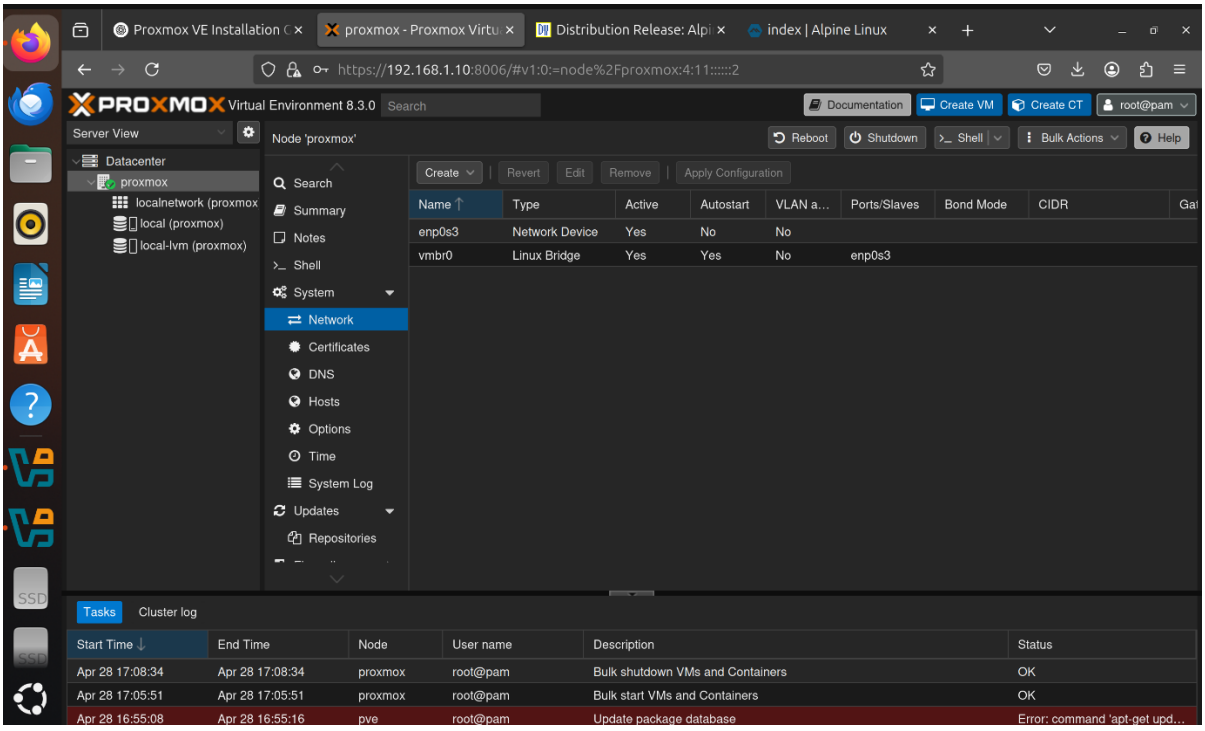
- Proxmox dashboard (Datacenter view).



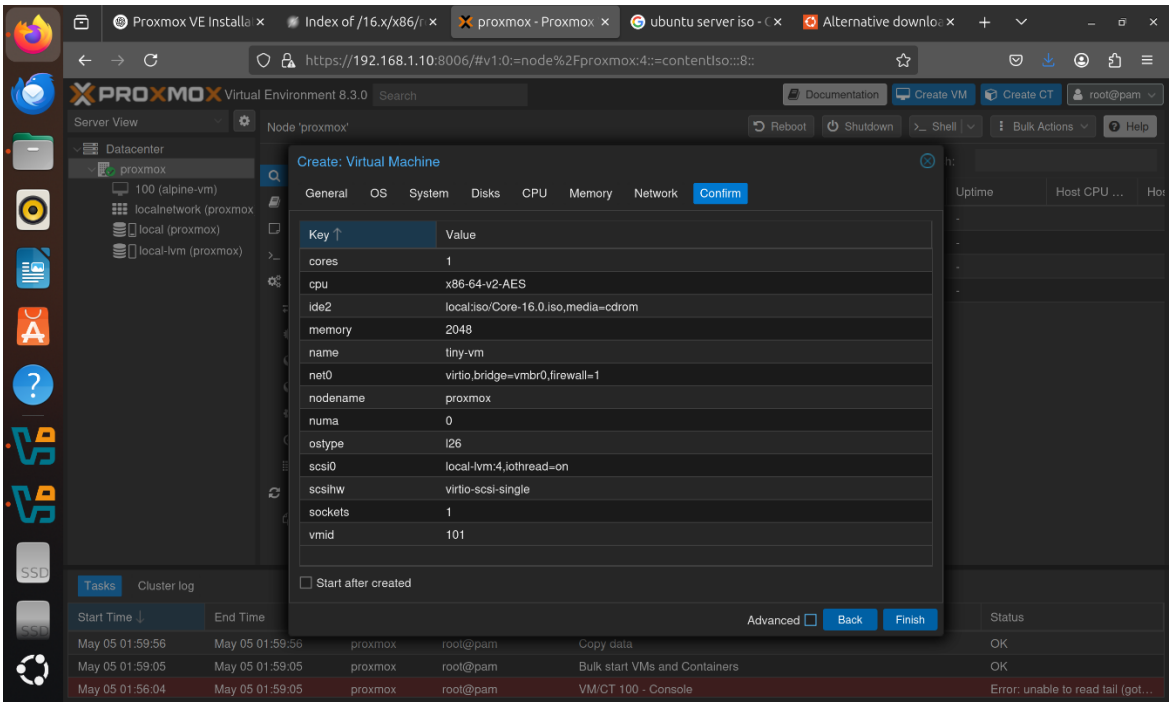
- Storage overview.



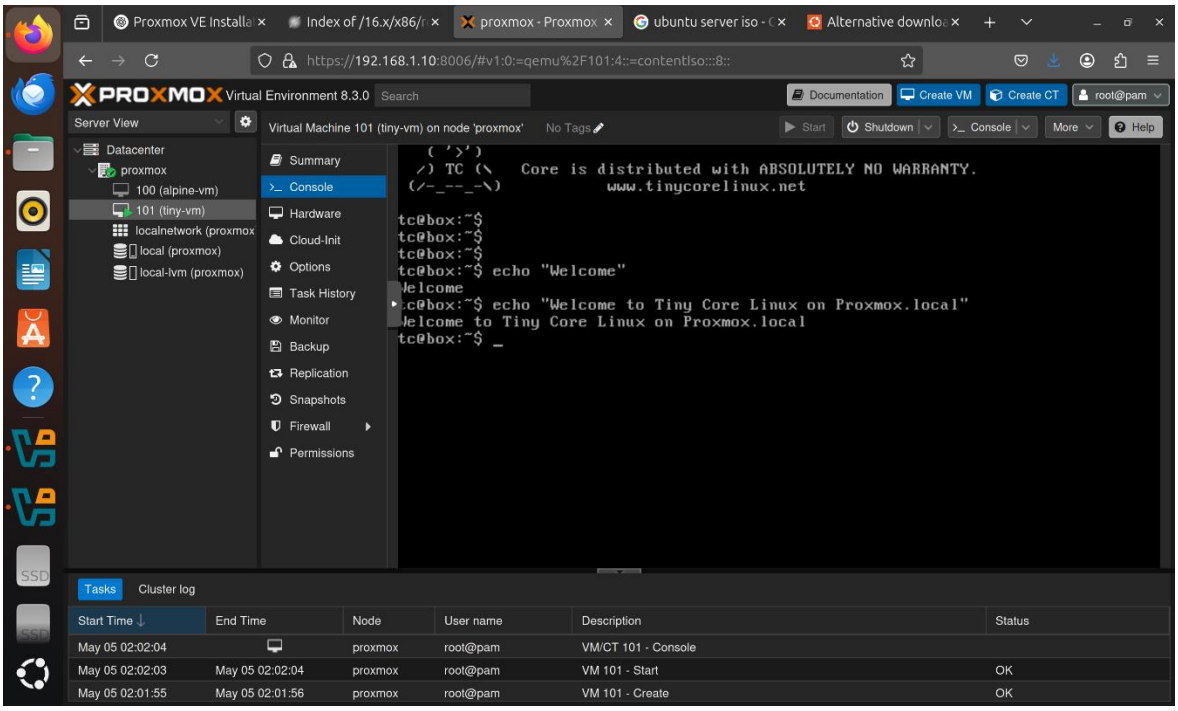
- Network settings page.



- Create VM wizard (summary before creating VM).



- Console with the new VM booting or installing an OS. (Used a Linux distro called tiny-core linux)



Observations

1. Installation:

   The installation process was straightforward and well-documented in the Proxmox installer. I noticed that the installer automatically detected and configured the virtual disk and network interface, which made setup easier. No unexpected issues occurred during this step.

2. Configuration:

   During configuration, the Proxmox interface allowed me to set the timezone, password, and network settings with minimal input. The system automatically obtained a valid IP address via DHCP, and the user interface provided clear confirmation of successful configuration. I was pleasantly surprised by how fast the web interface was available just after the configuration and reboot.

3. VM Creation:

   Creating a virtual machine inside Proxmox was intuitive, with a guided wizard that made the process straightforward. The ability to allocate resources and mount an ISO during setup was particularly convenient. I initially attempted to use an Alpine Linux ISO, but encountered issues where both the VM and the host machine would crash during the initial boot. Due to this instability, I switched to a lighter Linux distribution, Tiny Core Linux, which booted smoothly and ran reliably.

   Interestingly, the VM creation process in Proxmox felt quite similar to other Type 2 hypervisors like VirtualBox, despite Proxmox itself being a Type 1 hypervisor, unlike more complex platforms such as VMware ESXi.

Deployment

To transform the Proxmox installation into a fully deployed cloud environment, several additional steps are necessary.

1. First, storage should be configured properly using either local disks, LVM, or network-based storage like NFS or Ceph, depending on the intended scale.
2. Next, networking should be refined by setting up static IPs, VLANs, or bridges for better isolation and connectivity.

3. User and role management must be implemented to control access securely across different VMs or containers.
4. Finally, enabling features like backups, high availability (HA), and clustering (if multiple nodes are available) would complete the transition from a basic setup to a production-ready private cloud infrastructure.

Bonus Task

For the bonus task, I accessed the official Tiny Core Linux download page and downloaded the appropriate ISO file. Using this ISO, I successfully created my first virtual machine inside Proxmox VE, as shown in the last two screenshots provided in the **Screenshots** section.