**8 - R710 Proxmox Container - LXC**

**Log in:**

1. In Firefox, go to : https://192.168.124.**ppn**:8006

User: root

Password : <whatever>

**Storage area for uplading ISO’s into**

1. See ~7 mins into video:  
    Proxmox VE 6.0 Beginner Tutorial - Installing Proxmox & Creating a virtual machine.  
    <https://www.youtube.com/watch?v=I-e1_CTa4s0>  
   **NOTE:** in above video => At 10:58 as pointed out by Pierre Lavoie, we accidentally selected the storage location "ISOs" while this should have been "VM1". Please select the earlier created VM1 storage location instead.

**Templates**

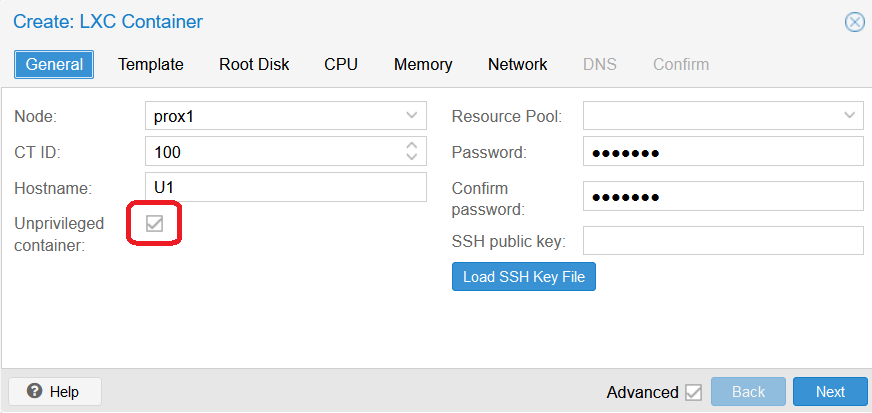
1. See video:

Creating an LXC Container in Proxmox Virtual Environment

<https://www.youtube.com/watch?v=cyjXxsQ8Igw>

1. Get a Ubuntu 18.04 template (which is stored in ‘local’)  
   [add here name of exact one] and note that it’s the minimum server version, NOT Desktop.

**Containers ( Ubuntu 18.04 )**

1. When creating Container, Ensure "Unprivileged container" is ticked (so that containers are self contained).  
   
2. Containers are stored in the “local-lvm” which is of type LVM-Thin (which does not pre-allocate all of the storage space – it is allocated as needed).
3. Setting up first Ubuntu container, keep following instructions in above mentioned video.  
   Hard Disk: Bus/Device: VirtIO Block [ for best performance in linux, possibly SCSI for windows ]  
     
   [ investigate:  
   cache for performace = writeback, but at risk of losing information if the physical host crashes or is breaks … try this as I have hardware RAID 10

cache for security = writethrough, isn't fast, but is the more sure, you will not lose information, and writethrough also do cache of writes on the physical host (a more RAM in the physical host, more amount of cache will have)

Bus/Device: VirtIO

Format: LVM as block device for your VMs

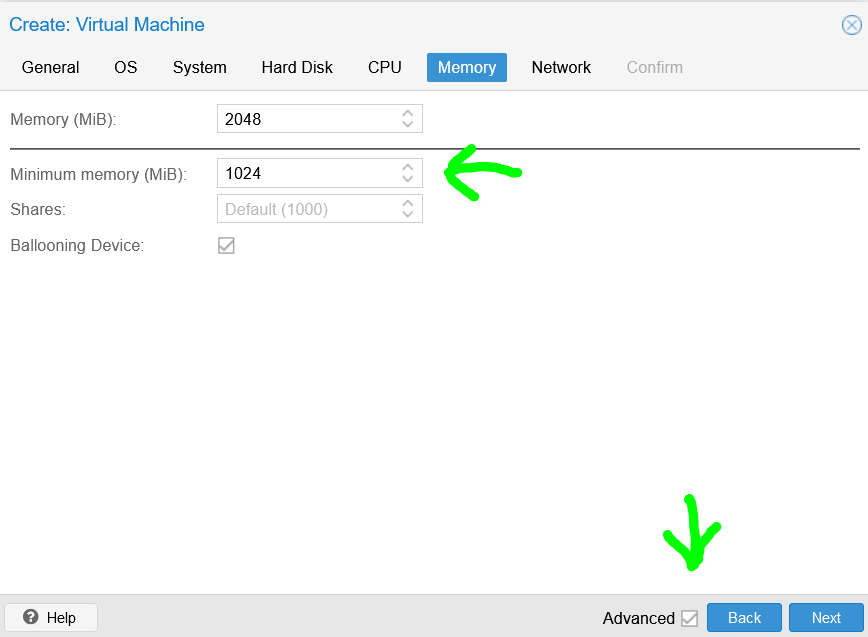
I/O scheduler for the VM = noop (no optimized, the PVE host have his own I/O scheduler)  
]  
  
Use Hostname of : u18-master [this will be a template to create others from]

User: root

Password: u1-1234  
root disk size: 40GB

CPU: 5 cores Type: host [for best performance]

RAM: 10240 MB, swap 512 [ look into option to set min and max memory]

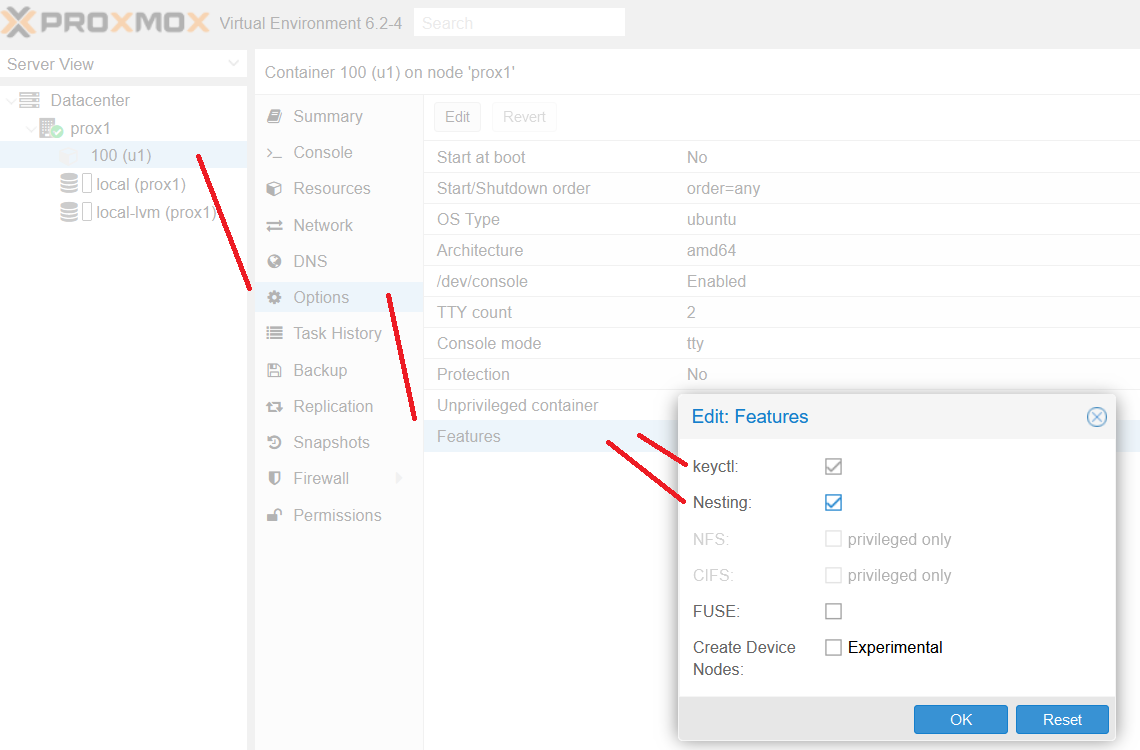


[ add screen shot here of how I configured static IP ]  
Network: static IPv4 : 192.168.124.202/24  
 gateway : 192.168.124.1  
DNS: click next  
on ‘Confirm’, click finish  
When in scroll list window, you see ‘TASK OK’, close the window.

1. Before running container:

Select Options, and double click on 'Features'

and turn on 'Keyctl' and 'Nesting'

(where 'Nesting' allow Docker to run inside Container)  


1. Select the Container: 100 (u1)  
   Start the container.  
   Click the ‘Console’ button and log in as root
2. Do updates:  
   apt-get update

apt-get upgrade -y

apt-get dist-upgrade -y

apt-get autoremove -y

apt-get autoclean –y  
apt install net-tools

1. To check IP address, execute command:

Ifconfig  
Then on machine that Firefox is running from, open a terminal (command window), and enter:  
 ping 192.168.124.202

1. At command line in console (for **u1** in proxmox, in Firefox):  
    shutdown –h now
2. Select node ‘prox1’ and click on the “Shutdown” button.
3. OR --- I might need to install it from a SERVER INSTALL ISO and … When installing server … go thru all this again .. ensure there is something to do with selecting and installing ssh.  
    and then create my own template from this.  
   **See: How to Create a VM Template in Proxmox VE** [**https://www.youtube.com/watch?v=PLFQbO\_FHUU&list=PLC53fzn9608DcSu6RrrB8uVkkO3N69Y-1&index=7&t=0s**](https://www.youtube.com/watch?v=PLFQbO_FHUU&list=PLC53fzn9608DcSu6RrrB8uVkkO3N69Y-1&index=7&t=0s)
4. OR --- try minimal install from following:  
    Creating your own Virtual Machine with Proxmox Virtual Environment (VE)  
    <https://www.youtube.com/watch?v=BiIFLFhXByE>   
   pay attention to ~9 mins into above video & also ~10:30 mins.  
   ---  
   Then creating a template from this with cloud-init, see video:  
   [ MUST DO this … ]  
    Creating a template in Proxmox Virtual Environment with cloud-init support  
    <https://www.youtube.com/watch?v=8qwnXd1yRK4>

**Docker in Container**

1. To actually install Docker into the container, see Step 1 at the following website without using sudo.

(You're root and don't need to use sudo for the commands.)

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>

1. After this step – look into saving container as a template from which to create others
2. Create new container from my new template and look into changing resources of containers as per this table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| node resource allocation in proxmox for various LXC's | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | left | GB for disk | GB - SWAP | node # | RAM | left | CPU's | left |
| prox1 | **170** | 30 | 2 | U0 | 10 | **48** | 4 | **24** |
|  | 138 | 30 | 2 | U1 | 10 | 38 | 4 | 20 |
|  | 106 | 20 | 1 | U2 | 4 | 28 | 3 | 16 |
|  | 85 | 20 | 1 | U3 | 4 | 24 | 3 | 13 |
|  | 64 | 20 | 1 | U4 | 4 | 20 | 3 | 10 |
|  | 43 | 10 | 1 | U5 | 4 | 16 | 2 | 7 |
|  | 32 | 10 | 1 | U6 | 4 | 12 | 2 | 5 |
|  | 21 | 10 | 1 | U7 | 4 | 8 | 2 | 3 |
|  | **10** |  |  |  |  | **4** |  | **1** |
|  |  |  |  |  |  |  |  |  |
|  | left | GB for disk |  | node # | RAM | left | CPU's | left |
| prox2 | **170** | 30 | 2 | U20 | 10 | **48** | 4 | **24** |
|  | 138 | 30 | 2 | U21 | 10 | 38 | 4 | 20 |
|  | 106 | 20 | 1 | U22 | 4 | 28 | 3 | 16 |
|  | 85 | 20 | 1 | U23 | 4 | 24 | 3 | 13 |
|  | 64 | 20 | 1 | U24 | 4 | 20 | 3 | 10 |
|  | 43 | 10 | 1 | U25 | 4 | 16 | 2 | 7 |
|  | 32 | 10 | 1 | U26 | 4 | 12 | 2 | 5 |
|  | 21 | 10 | 1 | U27 | 4 | 8 | 2 | 3 |
|  | **10** |  |  |  |  | **4** |  | **1** |

1. Ideally Terraform can take initial template and create other machines with the above resources (which all need docker, ssh and ansible in them and maybe kubernetes ???).
2. and :  
   Try running up an apache or nginx docker container and accessing its ip and port number from firefox on same machine that proxmox firefox viewer is running from.
3. From ML1 Get Ansible script going to do the above … need an ssh key thing set up, document all steps

**Extending VM disk size (not yet tried)**

(This sequence found on proxmox forum):  
Extended the drive from 8GB to 30GB.

1) Turned off the VM

2) Backed up the VM

3) extended the disk using web GUI

4) booted the VM using Ubuntu Desktop

5) launched GParted using TRY Ubuntu

6) Right Clicked both sd2 and sd5 and chose "Deactivate".

7) Resized the extended (sda2) partition.

8) Resized the lvm (sda5) partition.

9) Booted up the VM normally

10) executed the following Commands:

a) lvextend -l +100%FREE [mount point] {/dev/vrtemlsrv-vg/root}

b) resize2fs [mount point] {/dev/vrtemlsrv-vg/root}​

11) Restarted and confirmed size

12) Backed up the VM

**Cluster and Ceph storage creation in Virtualbox, including HA (container and VM restart after node failure => see 36:12 mins into vid)  
 [NOTE: Ceph is not compatible with disks backed with a hardware RAID controller]**

See video:  
 Proxmox VE 6.0 new features | Easy Cluster creation | Easy Ceph storage  
 <https://www.youtube.com/watch?v=GgliWaOfvsA>

-=-=-

14th Jun 2020

Alternative disk setup for ZFS, that might work for perc6i (that I have), tough I doubt if the following will play ball:

H700/710 cards does **not** have pass-through capability. As a workaround, you can configure each disk as an independent RAID0 array and build your ZFS pool from these "fake", single-disk arrays. **This is not recommended, though,** as ZFS really expect to have low-level access to your disks. You should be better off using a classical RAID5/10 array and formatting is with ZFS.

-=-=-

Thought, apparently a H200 card is needed that can be re-programmed with pass though, so that the OS can see the drives directly.

-=-=-