

Xen DOM0

Base Install

```
apt -y install xen-system
# if you don't do this, it will boot into non-xen kernel
dpkg-divert --divert /etc/grub.d/08_linux_xen --rename
/etc/grub.d/20_linux_xen
# now, limit DOM0 memory to 4G
echo '# Xen boot parameters for all Xen boots' >> /etc/default/grub
echo 'GRUB_CMDLINE_XEN="dom0_mem=4096M,max:4096M"' >> /etc/default/grub
echo '# do not probe anything but the boot drive for os' >>
/etc/default/grub
echo 'GRUB_DISABLE_OS_PROBER=true' >> /etc/default/grub
echo 'autoballoon="0"' >> /etc/xen/xl.conf
# and tell grub to update itself
update-grub
# tell Xen to use the xl tool stack. It appears with newer Debian releases,
this is the default
# so probably not necessary
# NOT TESTED, you may have to hand edit.
echo 'TOOLSTACK=xl' >> /etc/default/xen
# at this point, we are ready to go, so reboot the system into the Xen
kernel
reboot
```

Note: There is an option to limit the number of vcpu's DOM0 can have, but my notes are possibly incorrect. I had:

```
GRUB_CMDLINE_XEN="dom0_mem=4096M,max:4096M dom0_max_vcpus=4"
```

Limit DOM0 memory

At times, the DOM0 will attempt to grab all RAM and make it where no virtuals can exist. There may be other reasons to limit the amount of memory available to the DOM0, though I generally like to give it a bunch so it can cache things.

====

DOM0 and UEFI

I have run into some serious issues trying to get a DOM0 to boot with UEFI. A fresh Devuan (Debian) install works just fine, then when the Xen kernel is brought up, the boot fails. I don't remember the exact issue, but it has been around for several years.

The article at https://wiki.xenproject.org/wiki/Xen_EFI describes how to build a custom kernel.

However, my fix is to simply set the server to boot into legacy BIOS mode. As long as that is an option, it appears to be better for the time being.

Adding iSCSI and NFS

For some of our systems, we are using an iSCSI target for the device back ends (vbd's). We also put common configuration files on an NFS share, then mount it at /etc/xen/iscsi_configs. This allows us to migrate virtuals (DOMU's) rapidly between the Xen Hypervisors.

Install the packages for the client (aka initiator for iSCSI)

```
apt -y install open-iscsi nfs-common
```

And, change /etc/iscsi/iscsid.conf to automatically start on boot. Default is to do manual startup, so uncomment automatic and comment node.startup = manual

[iscsid.conf](#)

```
# To request that the iscsi initd scripts startup a session set to
"automatic".
node.startup = automatic
#
# To manually startup the session set to "manual". The default is
manual.
# node.startup = manual
```

set up NFS mounts

Now, create an entry in /etc/fstab to automount /etc/xen/iscsi_configs

```
mkdir -p /etc/xen/iscsi_configs
echo 'ip.of.iscsi.target:/media/nfs_root/xen/configs /etc/xen/iscsi_configs
nfs defaults 0 0' >> /etc/fstab
mount /etc/xen/iscsi_configs
```

set up iSCSI targets

This is a manual operation, but there is a script that will do it automagically for you, if you like, located at [iscsi_tricks_and_techniques](#)

Links

1. https://wiki.xen.org/wiki/Xen_Project_Best_Practices
2. <https://wiki.debian.org/Xen>

From:

<https://wiki.linuxservertech.com/> - **Unix Server Tech Knowledge Base**

Permanent link:

<https://wiki.linuxservertech.com/unix/virtualization/xen/dom0?rev=1601937165>

Last update: **2020/10/05 22:32**

