Debian GNU/Linux

Install Basic Operating System

The latest version of Debian GNU/Linux 5.0 (5.0.5 as of this writing) should be installed on each physical host used by a HUBzero installation.

To install Debian GNU/Linux, you can easily obtain a copy, and then follow the installation instructions for your architecture (only 64bit [AMD64] is currently supported).

Installing Debian GNU/Linux using a a small bootable CD is the simplest method.

When installing Debian GNU/Linux be sure to do the following:

- Ensure the disk(s) are partitioned to have at least:
    - A bootable partition at least 100.0 GB in size for the root filesystem.
    - An empty partition at least 50.0 GB in size (note the device name of this partition for later)
    - An appropriately sized swap partition.
- When prompted to select an installation package just select "Standard System", other packages will be added later

When the installation is complete your system will reboot into a minimal Debian GNU/Linux system.

Don't forget to remove your installation media and/or change your server's boot media order if you changed them prior to installation.

Set hostname

Optional. If you didn't specify the fully qualified domain name when running setup you will need to set it here.

HUBzero expects the `hostname` command to return the fully qualified hostname for the system.

```
# hostname myhub.org
```

To make the change permanent you must also edit the file /etc/hostname, this can simply with:

```
# echo "myhub.org" > /etc/hostname
```

Delete local user

If you created a local user account when installing the operating system now would be a good time to delete it before it causes you future problems.

```
# deluser username
```

Configure Networking

Optional. If you didn't configure networking during installation you will need to do so now.

For help with networking setup try this link.

Setting up your IP address.

The IP addresses associated with any network cards you might have are read from the file **/etc/network/interfaces**. This file has documentation you can read with:

```
# man interfaces
```

A sample entry for a machine with a static address would look something like this:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
 address 192.168.1.90
 gateway 192.168.1.1
 netmask 255.255.255.0
 network 192.168.1.0
 broadcast 192.168.1.255
```

Here we've setup the IP addresss, the default gateway, and the netmask.

For a machine running DHCP the setup would look much simpler:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface - use DHCP to find our address
auto eth0
iface eth0 inet dhcp
```

(If you're using a DHCP based setup you must have a DHCP client package installed - usually one of pump or dhcp-client.)

If you make changes to this file you can cause them to take effect by running:

```
# /etc/init.d/networking restart
```

Setting up DNS

Use whatever nameserver and other options as recommended by your ISP. If you used DHCP to set up networking it is likely this has already been set.

When it comes to DNS setup Debian doesn't differ from other distributions. To cause your machine to consult with a particular server for name lookups you simply add their addresses to /etc/resolv.conf.

For example a machine which should perform lookups from the DNS server at IP address 192.168.1.10 would have a resolv.conf file looking like this:

```
nameserver 192.168.1.10
```

Configure Advanced Package Tool

Now configure what debian distribution mirror to use and the location of the HUBzero package repository by editing /etc/apt/sources.list to look like:

```
deb http://ftp.us.debian.org/debian/ lenny main
deb-src http://ftp.us.debian.org/debian/ lenny main
```

```
deb http://security.debian.org/ lenny/updates main
deb-src http://security.debian.org/ lenny/updates main

deb http://volatile.debian.org/debian-volatile lenny/volatile main
deb-src http://volatile.debian.org/debian-volatile lenny/volatile main

deb http://packages.hubzero.org/deb lenny main contrib non-free
deb-src http://packages.hubzero.org/deb lenny main contrib non-free
```

You will need to get and install the hubzero archive key to be able to verify packages from the hubzero archive:

```
# wget http://packages.hubzero.org/deb/hubzero-signing-
key.asc -q -O - | apt-key add -
```

Once the public key for http://packages.hubzero.org has been install you can then upgrade the current packages to their latest releases.

```
# apt-get update
# apt-get upgrade
```

SSH

Next we install fail2ban and ssh

```
# apt-get install fail2ban ssh
```

At this point you can continue configuration and setup remotely if that is more convenient.

Enable OpenVZ

If you are installing this in a VirtualBox VM you must enable PAE/NX support. Go to system ->

processor of your VM, select "Enable PAE/NX".

To use OpenVZ you must use an OpenVZ enabled kernel which is easily installed.

HUBzero makes extensive use of OpenVZ containers so it is recommended to just use the OpenVZ enabled kernel on all HUBzero servers.

```
# apt-get install linux-image-2.6-openvz-amd64
```

You will need to reboot the server to activate the new kernel.

```
# reboot
```

Once you have rebooted you can verify the new kernel is active

```
# uname -a
Linux myhub.hubzero.org 2.6.26-2-openvz-
amd64 #1 SMP Thu Nov 5 03:06:00 UTC 2009 x86_64 GNU/Linux
```

With the new kernel active you can remove the old one

```
# apt-get purge linux-image-2.6.26-2-amd64
```

Prepare Fileystem

The root filesystem ('/') runs with quotas disabled and contains the primary operating system for the server and for each OpenVZ container hosted on the server.

Each HUBzero server may use an addition partition for use appropriate to the function of the server (web document root, project data, home directories, etc).

If you did not create an empty partition during setup, create one now using your favorite disk partitioning tool. Be sure to note the device name for the partition you create as it will be used below.

Once you have an empty partition ready we can install a filesystem. Replace "/dev/PART" with the device name for the empty partition you have created (e.g., /dev/sda2). The command "fdisk -l" will list all paritions the system knows about.

```
# mke2fs -j /dev/PART
# e2fsck -f -C 0 /dev/PART
# mkdir /export
```

Then make sure the following line appears in /etc/fstab

```
/dev/PART     /export      ext3    defaults,quota,errors=remount-
ro     0     2
```

Then mount the new filesystem

```
# mount /export
```

Bind mount /home

Create a 'home' directory in our new /export filesystem. move the contents of the default home directory to the new location, then bind mount new location over the old.

```
# mkdir -p /export/home/myhub
# mv /home/* /export/home
# mount --bind /export/home /home
```

Bind mount /opt

Currently HUBzero uses the /opt directory for storing subversion and trac data as well as some of hubzero supporting software as well. We recognize this may not be the best organization.

Create a 'opt' directory in our new /export filesystem. move the contents of the default /opt directory to the new location, then bind mount new location over the old.

```
# mkdir /export/opt
# mv /opt/* /export/opt
# mount --bind /export/opt /opt
```

Bind mount /apps

Currently HUBzero uses the /apps directory for storing installed tools and other software that needs to be available to each execution container.

Create a 'apps' directory in our new /export filesystem and in the root filesystem. Then bind mount /export/apps over /apps.

```
# mkdir /export/apps
# mkdir /apps
# mount --bind /export/apps /apps
```

Bind mount /www

HUBzero uses the /www directory for storing the document root and supporting directories needed by the web server.

Create a 'www' directory in our new /export filesystem and in the root filesystem. Then bind mount /export/www over /www.

```
# mkdir -p /export/www
# mkdir /www
# mount --bind /export/www /www
```

Update /etc/fstab

Now edit **/etc/fstab** with the bind mounts we created above by adding the following lines

```
/export/opt  /opt        none    bind,defaults
0     0
/export/apps /apps       none    bind,defaults
0     0
/export/home /home       none    bind,defaults
0     0
/export/www  /www        none    bind,defaults
0     0
```