Installation of Ubuntu on Mac Mini

Contents

1 Overview

These notes describe the installation of Ubuntu 14.04 on a Mac Mini. This is the most powerful server we have found for its size, especially considering that the PSU is internal.

1.1 Mac Mini versions

The recommended configuration of a workshop server is:

- Quad-core i7 ("server" version)
- 16GB RAM
- Two 250GB SSDs (e.g. OCZ Vertex 4)

These instructions have been tested on the following models:

- Mid 2010 Core 2 (Macmini4,1)
- Mid 2011 i7 (Macmini5,3)
- Mid 2013 i7 (Macmini6,2)

When you have Linux running, you can determine your model using

```
sudo dmidecode -s system-product-name
```

For more information on the different models of Mac Mini see:

- https://support.apple.com/kb/HT3476
- https://en.wikipedia.org/wiki/Mac_Mini

1.2 Items required

For installation only, you will need:

- A mini-displayport to VGA adapter, or a HDMI cable
- A monitor

- A USB keyboard
- A 1GB+ USB flash drive or a USB CD-R drive

We also recommend an Apple USB to Ethernet adapter for the external Internet connection. This leaves the internal gigabit Ethernet port available for the lab internal network and avoids having to trunk VLANs; it also helps if you have a newer Mac Mini where the internal ethernet adapter is not supported in the default kernel.

2 Preparing the install image

First you need to choose which ISO image to use and download it. Start with the standard 64-bit server edition (ubuntu-14.04-server-amd64.iso) which should work fine with recent Mac Minis.

If you have problems booting, then try one of the amd64+mac[1] variants from the alternative downloads page:

- ubuntu-14.04-server-amd64+mac.iso
- ubuntu-14.04-desktop-amd64+mac.iso

You may burn the image onto a CD-R if you have a USB CD-ROM drive to boot from, such as the Apple Superdrive.

To write the image to a USB stick you can simply dd it. To do this on a Mac running OSX:

```
diskutil list  # note the devices seen
# insert flash drive
diskutil list   # note the new device seen, e.g. /dev/disk2
diskutil unmountDisk /dev/diskN
sudo dd if=ubuntu-14.04-server-amd64.iso of=/dev/rdiskN bs=1m
diskutil eject /dev/diskN
# remove flash drive
```

To install older versions of Ubuntu you need to use a Mac which is running OSX and follow these instructions

```
hdiutil convert -format UDRW -o ubuntu-12.04.3-server-amd64.img ubuntu-12.04.3-server-amd64.
# Note: the output filename may be .img.dmg
diskutil list   # note the devices seen
# insert flash drive
```

[1] See http://askubuntu.com/questions/37999/what-is-different-about-the-mac-iso-image

```
diskutil list   # note the new device seen, e.g. /dev/disk2
diskutil unmountDisk /dev/diskN
sudo dd if=ubuntu-12.04.3-server-amd64.img.dmg of=/dev/rdiskN bs=1m
diskutil eject /dev/diskN
# remove flash drive
```

3 Installation

Insert the CD-ROM or flash disk into the Mac Mini, and connect the keyboard and screen. Connect ethernet uplink - using the USB-to-ethernet adapter if you have one.

Power on and hold the alt/option key while it is starting. You may get a choice of boot icons - the USB key will be "EFI boot".

After this you should get a menu of Ubuntu options:

```
Install Ubuntu Server
Install in expert mode
Multiple server install with MAAS
Check disc for defects
Rescue a broken system
```

Select "Install Ubuntu Server" and proceed as normal, choosing country, language and keyboard layout.

If you are using the USB-to-ethernet adapter for your external Internet connection, then you will be prompted for the primary network interface:

```
eth0: Broadcom Corporation NetXtreme BCM57765 Gigabit Ethernet PCIe
eth1: Ethernet
```

Choose the one which is your ethernet uplink (i.e. eth1)

- When you see "Configuring network with DHCP" press ENTER to cancel. Do this quickly!
- Press "Continue" on the Network autoconfiguration failed screen.
- Select "Do not configure the network at this time"
- Hostname: `kit<number>` or whatever unique identifier for this setup
- Full name for the new user: NSRC instructor
- Username: nsrc

- (Use a suitable instructor password)
- Encrypt your home directory? No
- Select timezone. If you are going to be travelling around the world you should not accept the local timezone, and instead scroll down to "UTC" at the very end of the worldwide list.
- Partition
  - If you see a screen that the install has detected a mounted partition select Yes to unmount the partition before continuing.
- Partitioning:
  - Guided - use entire disk and set up LVM
  - "Select disk to partition" SCSI1 (0,0,0) sda
  - "Write changes to disk and configure LVM?" Yes
  - Amount of volume group to use for guided partitioning: 50G (This will give about 31G root and 16G swap, but these are easy to grow later if required)
  - The partitioning generated should look something like this:

    ```
    LVM VG hostname-vg, LV root as ext4
    LVM VG hostname-vg, LV swap_1 as swap
    partition #1 of SCSI1 (0,0,0) (sda) as EFIboot
    partition #2 of SCSI1 (0,0,0) (sda) as ext2

    Write the changes to disks?
    ```

    Answer Yes.
- Use HTTP proxy? *Check with your instructors* whether there is a local cache you should use to speed up installation. If not, leave this blank, and packages will be fetched over the Internet.
- No automatic updates (you don't want changes in the middle of a workshop)
- Software to install: select only "OpenSSH server". Use the space bar to toggle selections.
- Installation is complete: hit Enter, then unplug CD-ROM or USB stick when the screen goes black.

4 Login after reboot

Your Mac should be up and running - login as user `nsrc`.

Use `ifconfig -a` to find its external IP address, and you should be able to ssh into it from outside. Then you no longer need the keyboard and monitor.

Hint: also take a note of the MAC address of the USB-ethernet adapter, and label it. It helps when locating the machine without a monitor.

Since all the following commands need to be done as root, get a root shell:

```
$ sudo -s
#
```

4.1 Add the second hard drive

The second hard drive can now be made available for storing data.

The simplest way to do this is to make the whole second drive an LVM physical volume, and add it to our volume group.

First, check your current volume group name:

```
# vgs
```

Look under the first column "VG": if it was created by the Ubuntu installer it will be something like "-vg".

Label your second drive as a physical volume, and add it to the volume group:

```
# pvcreate /dev/sdb
# vgextend <hostname>-vg /dev/sdb
```

Now check your work:

```
# pvs   # should show both /dev/sda3 and /dev/sdb as physical volumes
# vgs   # The "#PV" (number of physical volumes) and VSize/VFree should increase
```

> Alternatively you could choose to partition it in the same way as the first drive, and it might be possible to boot off this drive if the first one fails - but you'd almost certainly have lost your OS installation anyway.

Now let's create a new logical volume and mount it under `/data`. By creating a "striped" volume, half of the extents will be on each disk; this increases performance by dividing the read and write load between the disks. Choosing a large stripe size ensures that *smaller* reads and writes hit only a single disk.

```
# lvcreate --size 300G --stripes 2 --stripesize 4096 --name data <hostname>-vg
# mkfs.ext4 -m 0 /dev/<hostname>-vg/data
# mkdir /data
# vi /etc/fstab
... add this line ...
/dev/<hostname>-vg/data  /data  ext4    noatime         0       2

# mount /data
# df -h
```

To see which physical volume(s) each logical volume is using:

```
# lvs -a -o +devices
or
# lvs -a -o name,lv_size,devices
```

4.2 Optimise the swap volume

You can move the swap volume to the second disk, which helps balance the usage slightly.

```
# pvmove -n swap_1 /dev/sda3 /dev/sdb
```

Optionally, you could also reduce its size. By default a 16GB RAM system will have a 16GB swap volume. If you don't intend to use hibernate functionality you can reduce it to (say) 4GB, as you never want your machine to go this far into swap anyway.

```
# swapoff -a
# lvresize --size 4G /dev/<hostname>-vg/swap_1
  WARNING: Reducing active logical volume to 4.00 GiB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce swap_1? [y/n]: y
  Reducing logical volume swap_1 to 4.00 GiB
  Logical volume swap_1 successfully resized
# mkswap /dev/<hostname>-vg/swap_1
```

Finally enable the swap partitions again

```
# swapon -a
# free   # should have 4GB of swap space
```

4.3 Configure the internal network interface

We purposely did not configure our network interfaces during installation so that the installation would go faster and we could configure out interfaces as we want now. You can see that neither eth0 or eth1 are configured by doing:

```
$ ifconfig
```

To fix this, as root, at the command line prompt type:

```
# vi /etc/network/interfaces
```

Make sure the file looks like this:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.0.241
netmask 255.255.255.0

auto eth1
iface eth1 inet dhcp
```

Now to bring up both interfaces with their new configurations do:

```
# ifup eth0
# ifup eth1
```

Wait a few seconds for the script to complete then type:

```
# ifconfig
```

to see that your network is now configured. Trying pinging some external sites to verify that your network is working.

4.4 Update all packages

Now that our base operating system is installed we need to pull down a list of current versions of available packages and then download and install the packages.

Let's first see what version of the Linux kernel we are running:

```
$ uname -a
```

Remember this and see if you have a newer kernel version once we are finished with this exercise. To update apt and then packages that need updating do:

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

Say "Yes" to downloading and installing the new packages.

This could take some time as a considerable amount of information is going to be downloaded. While this is happening we will go on to the next set of configuration items for your workshop kit which includes configuring your switch and access point.

When your machine finishes with the apt-get update and the apt-get dist-upgrade processes you will likely have a new kernel image. This is one of the few times you need to reboot Linux to see change. To do this do:

```
# reboot
```

Wait for the MacMini to restart and log back in as the user nsrc, then type:

```
$ uname -a
```

To see the version of the Linux kernel that you are running. Is it different from what you had before running "apt-get update" and "apt-get dist-upgrade"?

At this point your MacMini is ready and in its initial state. We'll be updating your MacMini environment using a tool called Ansible a bit later.

Assuming your network interfaces are properly configured you can now remove the monitor and keyboard attached to your MacMini and connect to it using ssh instead.

4.5 Update /etc/hosts file

By updating your /etc/hosts file to include the lab.nsrc.org domain your machine's public IP address will be registered on the public Internet using dynamic DNS. This will be configured using Ansible in our upcoming exercises. In addition for class exercises and infrastructure your Mac Mini is referred to as "s1" or "s1.ws.nsrc.org". We will add both these entries to your /etc/hosts file.

The first two lines in your /etc/hosts file should look like:

```
127.0.0.1       localhost
127.0.1.1       kitX
```

The "X" is the number of your kit. We will now edit this file by doing:

```
$ sudo vi /etc/hosts
```

And change the first two lines to look like this:

```
127.0.0.1       localhost
10.10.0.241     kitX.lab.nsrc.org kitX s1.ws.nsrc.org s1
```

Be sure you substitue your kit's number for "X". Once done, save the file and exit.

5 Troubleshooting post installation problems

5.1 Your Mac does not boot

If this is the first time you are installing Linux on your MacMini it's possible that the Linux installer will not set the boot order correctly. Your Mac may be set to boot Mac OS X, but this is no longer available. To correct this you need to do:

- Connect a current Ubuntu version installation media, such as an Ubuntu 14.04 LTS USB disk.
- Start your MacMini and hold down the option key until you see an option to boot as "EFI Boot"
- Select "EFI Boot" and when presented with the Ubuntu Install screen select "Rescue a broken system"
- You are now asked to run through the initial stages of the Ubuntu installation.
- Select keyboard, country, primary network interface (eth1), set hostname and time zone. Note that you can just use all defaults as none of these items will be saved.
- When you see the screen for "Device to use as root file system:" select `/dev/s1-vg/root`
- When asked whether to mount the separate /boot partition, say Yes.
- Select "Execute a shell in /dev/-vg/root" and select "Continue"

At the bottom of your screen you will see the "#" command prompt. You are now in a rescue Linux shell where we can set some hardware options. You can type:

```
efibootmgr
```

You'll probably see MacOS X listed as the BootCurrent item (0080). We want the Ubuntu entry to become the BootCurrent item. To do this type:

```
efibootmgr -o 00
```

And, that's it.

- Type "exit" to leave the rescue shell
- Select "Reboot the system" and remove the USB key.

Your MacMini should now boot. It may pause for a moment with the white screen, but give it a few seconds to complete booting.

5.2 Problems with video

If the attached monitor displays the grub prompt but then goes blank after booting, your monitor may not support the default resolution. If this happens, try booting with parameter "nomodeset".

- Hit any key at the grub menu to stop booting
- Hit 'e' to edit
- Go to the end of the "linux ..." line, and add `nomodeset` to the end
- Ctrl-X or F10 to continue booting

Then to make this change permanent:

```
# vi /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT="nomodeset"

# update-grub
```

5.3 Problems with ethernet

On the Internet there is another set of instructions for installing Ubuntu 12.04 on a Mac Mini 6,2. The author describes building the tg3 network driver from source and installing with the "noapic" option. However we found we didn't have to do this.

6 Installing on older MacMinis

The following instructions were from previous tests with a Macmini4,1 and are kept in case they help debugging problems on older machines.

6.1 Boot using Mac OS X install CD

6.2 Wipe out all partitions

It's probably safer to wipe out the existing Mac OS X and Linux partitions, which should also get rid of rEFIt. That way we make sure that rEFIt is not changing how Linux views the system.

- Use Disk Utility to remove all partitions on both drives

6.3 Get the right Ubuntu install CD

The image used successfully was the alternate-amd64+mac.iso image

The problem is that this installs all the desktop GUI crap, which must be removed after installation.

6.4 Install Ubuntu

- Boot from CD
- Press F6 and select:
  - acpi=off
  - nomodeset
- Continue as above, including same partitioning scheme
- Reboot

6.5 Edit grub defaults

```
# vi /etc/default/grub

GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX="noacpi nomodeset reboot=acpi"
```

Disable lightdm (replacement for gdm):

```
echo 'manual' | sudo tee /etc/init/lightdm.override
```

Then:

```
# update-grub
```

6.6 Convert into server

Remove all the desktop-related packages:

```
# apt-get --purge remove 'gnome-*' xerver-xorg
```

And install the right kernel:

```
# apt-get install linux-headers-server linux-image-server linux-server
```

All set.

Aside: there may be another way to do this using tasksel