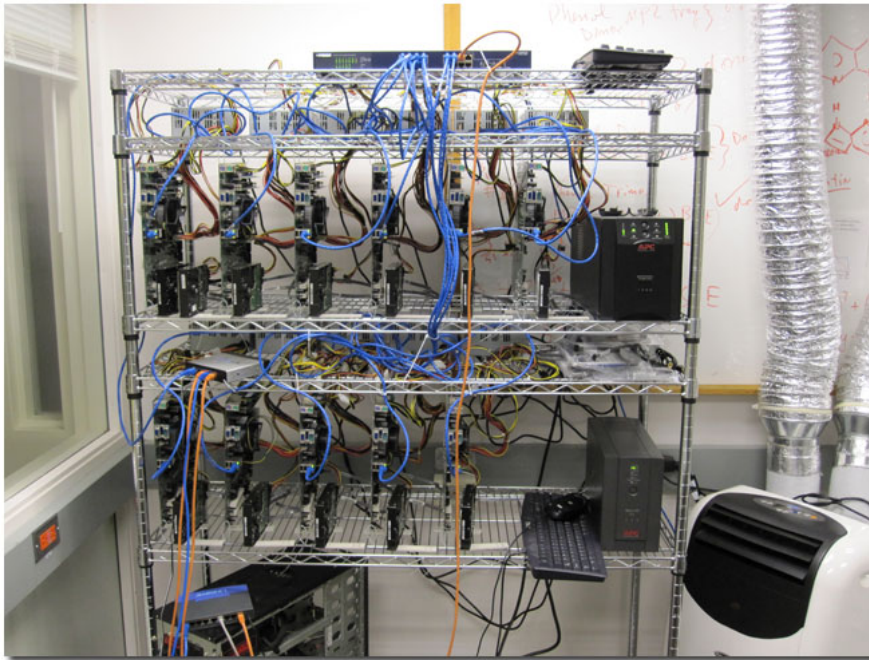


## Internet Connection Sharing – Ubuntu 10.04 NAT Gateway Setup (Abridged Version)

Had I known it would be this straightforward, I would have done it much sooner.

As is the case in many posts here, what follows is a summary of what I had to do to do what it is I'm writing about, kept largely for reference purposes but around just in case someone wants to do the same and would rather not take several web pages and combine them into one procedure that they may still have the stumble through to make work.

This page would not have been possible without the procedure outlined at [ubuntuforums.org/showthread.php?t=713874](http://ubuntuforums.org/showthread.php?t=713874).



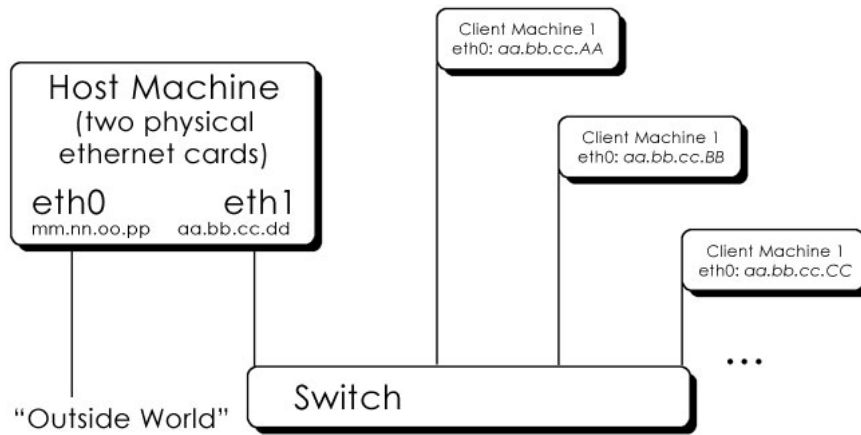
My current cluster configuration (say, as the one shown above) contains one system board dedicated to web access (with a single available IP address from my provider) and general file transfer hosting, then several boards dedicated to computation. As an [Ubuntu](#) user, every fresh install of some new program may require the addition of libraries or subordinate programs not available in the “standard” Ubuntu installation.

With a single IP address to the room, the procedure for performing a proper install of a library or program involves either running **sudo apt-get install program/library** on the web-accessible machine and saving all of the .deb files from a first machine, transferring those .deb files to all other machines, and installing, or, still more involved, “walking” the ethernet cable down the line of ports on each system board, reassigning IP addresses in **/etc/network/interfaces**, and performing installations. This long involved set of different processes are then also required for system updates. Not only is this tedious, it requires taking machines “offline” from your cluster one-at-a-time (that is, if you don’t have second ethernet cards on each system board to keep each ready for walking the ethernet cable down the line). It’s the “other” IPv4 exhaustion.

The easiest solution that keeps your hardware from otherwise ever being touched (and keeps you from having to sit in front of it) is to set up a [NAT Gateway](#), an approach that uses one host machine to share the outside world connection to the other machines. The whole setup procedure consists of a simple hardware setup, some configuration of the machine that has the hard-coded (or not) IP address, and modifications to the **/etc/network/interfaces** file of all associated machines to direct their search for the outside. After, all machines attached to the switch also connected to the Host Machine can use the “world” IP address of the Host Machine to update OS, transfer files, host pages (with some port work, which I won’t go into), etc.

The steps are as below that are all I have to follow to get this working in Ubuntu 10.04 (and should be very, very similar in all cases).

### Hardware Setup



The image above tells it all. This procedure uses 1 machine with 2 ethernet cards (which likely means one on the system board and a used PCI slot), 1 switch, and X number of other machines.

## Software Setup On Host Machine

Starting from a fresh installation on the Host Machine (although not necessary. This first pass was set up on a machine already the recipient of several updates and software installations) which properly recognizes the two ethernet cards, the procedure is as follows.

### 1. sudo pico /etc/network/interfaces

First we change the interfaces file to reflect that the two ethernet cards are two be divided into “public” (the internet. Here, **eth0**) and “private” (the intranet. Here, **eth1**). The contents of my file are as below:

```

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address mm.nn.oo.pp
netmask qq.rr.ss.tt
gateway uu.vv.xx.yy

auto eth1
iface eth1 inet static
address aa.bb.cc.dd
netmask 255.0.0.0
network aa.bb.cc.0
broadcast aa.bb.cc.255
    
```

Several things. **eth0** connects to the outside world. **mm.nn.oo.pp** is the IP address provided by your provider. The values for **netmask** and **gateway** (and **network** and **broadcast** if you have them) should be available from your provider (they may not be obvious).

**eth1** is being used for connection to the intranet. Letter pairs can be anything (but keep all **aa**’s the same, **bb**’s the same, etc.).

### 2. sudo pico /etc/resolv.conf

```

nameserver ee.ff.gg.hh
nameserver ee.ff.gg.ii
    
```

**nameserver** information should be available from your provider. Add this info to **resolv.conf**.

### 3. sudo pico /etc/sysctl.conf

From within this long file...

```

#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 4 4 1 7

#####3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
#net.ipv4.tcp_syncookies=1
    
```

```
# Uncomment the next line to enable packet forwarding for IPv4
# net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Ignore ICMP broadcasts
#net.ipv4.icmp_echo_ignore_broadcasts = 1
#
# Ignore bogus ICMP errors
#net.ipv4.icmp_ignore_bogus_error_responses = 1
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
```

Uncomment the line that reads:

```
# net.ipv4.ip_forward=1
```

so it reads:

```
net.ipv4.ip_forward=1
```

And save. This change will be made upon restart but can be started up on-the-fly by typing the following:

### 3a. `sudo sysctl -w net.ipv4.ip_forward=1`

### 4. `sudo pico /etc/rc.local`

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

exit 0
```

To the `rc.local` file add the following two lines at the bottom (above the `exit 0`):

```
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables --table nat -A POSTROUTING -o eth0 -j MASQUERADE
```

And save. This change will be made upon restart but can be started up on-the-fly by typing the following:

### 4a. `sudo iptables -P FORWARD ACCEPT`

### 4b. `sudo iptables -table nat -A POSTROUTING -o eth0 -j MASQUERADE`

### 5. `sudo apt-get install dhcp3-server`

This is the online machine after all. As one of the ubuntuforum pages ([ubuntuforums.org/showthread.php?t=713874](http://ubuntuforums.org/showthread.php?t=713874)) I worked from stated: “The start will fail, but that is nothing to worry about.”

### 6. `sudo pico /etc/dhcp3/dhcpd.conf`

Now we configure the dhcp server on this Host Machine so it knows how to assign addresses.

The dhcpd.conf example at [ubuntuforums.org/showthread.php?t=713874](http://ubuntuforums.org/showthread.php?t=713874) worked good enough.

```
ddns-update-style none;
option domain-name "mynetwork";
option domain-name-servers ee.ff.gg.hh, ee.ff.gg.ii;
```

```
option routers nn.mm.oo.pp;

default-lease-time 42300;
max-lease-time 84600;
authoritative;

log-facility local7;

subnet aa.bb.cc.0 netmask 255.0.0.0 {
    range aa.bb.cc.AA aa.bb.cc.ZZ;
}
```

Note: letter pairs here need to match up with the interfaces file. The **range** is for the Client Machines looking for IP addresses. My strong preference is to hard-code IP addresses for these Client Machines, so just have the range account for the number of machines from AA to ZZ.

## 7. sudo pico /etc/default/dhcp3-server

Finally, one last change on this Host Machine, to configure the dhcp server interface (for this cluster, eth1).

```
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/dhcp3-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
```

Simply add the following to the bottom of this file

```
INTERFACES="eth1"
```

My **strong** bias at this point is to restart this machine and confirm that the network works (that you can open web pages and ping accordingly), but you can restart the dhcp server by typing:

### 7a. sudo /etc/init.d/dhcp3-server start

## Client Machines

This could not be easier.

### 1. sudo pico /etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address aa.bb.cc.AA (up to ZZ)
    netmask 255.0.0.0
    network aa.bb.cc.0
    broadcast aa.bb.cc.255
    gateway aa.bb.cc.dd
```

Note! The gateway should be the IP address of the Host Machine eth1 card (**aa.bb.cc.dd**). That is all of the direction that is needed.

### 2. sudo pico /etc/resolv.conf

```
nameserver ee.ff.gg.hh
nameserver ee.ff.gg.ii
```

### 3. sudo /etc/init.d/networking restart

Either reboot the machine or use the above to restart the network. Following the above, the Client Machine can see the outside world, run system updates (that is, **sudo aptitude update** and **sudo aptitude upgrade**), etc.

Share this:

Twitter 1

Facebook 4

Google

StumbleUpon

LinkedIn

Email

Tags: [client machine](#) • [host machine](#) • [NAT gateway](#) • [ubuntu](#)

This entry was posted on Sunday, August 29th, 2010 at 7:31 PM and is filed under [linux issues](#), [scripts](#), [ubuntu](#). You can follow any responses to this entry through the [RSS 2.0](#) feed. You can [leave a response](#), or [trackback](#) from your own site.

## One Response to “Internet Connection Sharing – Ubuntu 10.04 NAT Gateway Setup (Abridged Version)”

1. [Ubuntu 12.04 IPv4 NAT Gateway and DHCP Server «Code Ghar May 2nd, 2012 18:00](#)  
1

[...] I begin this post, I want to thank Internet Connection Sharing – Ubuntu 10.04 NAT Gateway Setup (Abridged Version) for providing the bulk of the tutorial. I have made some modifications for Ubuntu [...]

## Leave a Reply

You must be [logged in](#) to post a comment.

« [Some Light Science Reading. The Constellations: Cygnus chown/chmod Your Way To A Backup Free Of The MemeodHelper Error in OSX 10.6.x \(Snow Leopard\) & A Seagate Vs. Western Digital Experience](#) »  
www.somewhereville.com is proudly powered by [WordPress](#), [Entries \(RSS\)](#) and [Comments \(RSS\)](#). 23 queries. 0.972 seconds. © #8^(=)+<, ph.d.

# Home

## About

quantum chemist computer geek nanotechnologist graphic artist amateur astronomer jazz/rock drummer.

## Connect

email • [damian@svw](mailto:damian@svw)  
aim • somewhereville  
gtalk • somewhereville  
skype • somewhereville  
facebook • [profile](#)  
google+ • [profile](#)  
myspace • [profile](#)  
academia • [profile](#)  
twitter • [profile](#)  
researchgate • [profile](#)  
linkedin • [profile](#)  
flickr • [profile](#)

## Search

## Family Links

- [Christopher Allis](#)
- [Rachel Hiller](#)
- [Live High Five](#)
- [Instant Classic](#)
- [W T H T](#)
- [George Wong](#)
- [Uncle Daddy](#)
- [2 Js And An I](#)

## CV, Press & Publications

- [full cv \(pdf\)](#)
- [cv \(html\)](#)
- [phd thesis \(153 MB\)](#)
- [papers](#)
- [books and chapters](#)
- [posters](#)
- [patents](#)
- [talks](#)
- [interviews and essays](#)
- [slidecasts](#)
- [journal & mag covers](#)
- [ubuntu mpich crystal06 intel-fort tutorial](#)

## Galleries

- [nano gallery](#)
- [aerial photography](#)
- [syracuse scenes](#)
- [panoramas](#)

## Categories

- [\(pc-\)gammess-us](#) (7)
- [abinit](#) (6)
- [amber](#) (11)
- [astronomy](#) (51)

- [audio and production](#) (7)
- [bioinformatics](#) (9)
- [biomedical theory](#) (37)
- [chem3d/draw](#) (1)
- [crystal06](#) (8)
- [crystal09](#) (4)
- [dmo3](#) (29)
- [drums & percussion](#) (10)
- [fftw](#) (3)
- [free press](#) (70)
- [gaussian98/03](#) (18)
- [gromacs](#) (15)
- [legislation](#) (2)
- [linux issues](#) (17)
- [molecular electronics theory](#) (5)
- [molecular manufacturing and nanotechnology](#) (27)
- [mpi](#) (12)
- [namd/vmd](#) (6)
- [NanoEngineer-1](#) (28)
- [nanotech gallery](#) (12)
- [neutron scattering spectroscopy](#) (10)
- [openmpi](#) (6)
- [operating systems](#) (18)
- [osx](#) (4)
- [povray](#) (23)
- [publications](#) (55)
- [qc/mm/md progs](#) (73)
- [quantum chemistry](#) (28)
- [scripts](#) (11)
- [site miscellany](#) (9)
- [six degrees](#) (22)
- [slidecasts](#) (2)
- [structural dna nanotechnology](#) (12)
- [supramolecular chemistry and molecular building blocks](#) (20)
- [terahertz \(THz\) spectroscopy](#) (25)
- [ubuntu](#) (16)
- [various photos](#) (21)

## Feeds

[RSS Feed](#)  
[RSS2 Feed](#)  
[Atom Feed](#)

## Meta

[Log in](#)

## Obligatory



## Ubuntu 4 Nano



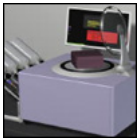
## CNYO



## NMT Review



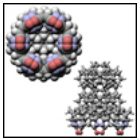
**N-Fact. Collab.**



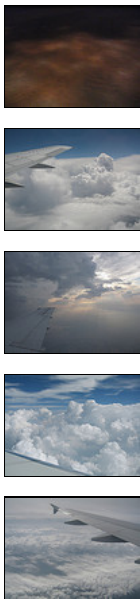
**T R P Nanosys**



**Nano Gallery**



**Aerial Photos**



More @ [flickr.com](#)

**Syracuse Scenes**





More @ [flickr.com](#) ↗