1 Initial Setup

This part applies to nims2, since that's the one that I set up from scratch. Plugged ethernet cable into port nearest the edge, and registered on the PNNL network by trying to get to google.com using Firefox. In the network control panel, set the dhcp client ID to "nims2" so PNNL's DNS server picks it up.

Setting the hostname requires changing it in **/etc/hosts**

```
sudo echo nims2 > /etc/hostname
```

and also editing **/etc/hosts** to point 127.0.1.1 at nims2:

```
127.0.0.1 localhost
127.0.1.1 nims1
```

I changed the editor from **nano** to **vim.tiny**, as follows. This affects commands like **visudo** and **vipw**, after I screwed up the **/etc/sudoers** file by not realizing that it had invoked **nano**.

```
sudo update-alternatives --config editor
There are 3 choices for the alternative editor (providing /usr/bin/editor).

  Selection Path Priority Status
------------------------------------------------------------
* 0 /bin/nano 40 auto mode
  1 /bin/ed -100 manual mode
  2 /bin/nano 40 manual mode
  3 /usr/bin/vim.tiny 10 manual mode
```

For remote access, we also need to install the SSH server:

```
apt-get install openssh-server
```

At this point, it's basically functional on the network, and accessible as a headless server.

2 Users and Groups

I added a **nims** user and **amaxwell** user, for the NIMS executables and my user, respectively, using `adduser foo` (which creates home directories and sets up permissions appropriately). The **nims** user's password is the same as that of **owner**. I then set my user up with **sudo** access via `sudo visudo`, which invokes the **vi** editor. After adding the line

```
amaxwell ALL=(ALL) ALL
```

at the end of the file, **amaxwell** can execute commands as root.

I edited **/etc/passwd** manually to change UID and GID of **nims** as follows:

```
nims:x:200:200:NIMS user,,,:/home/nims:/bin/bash
```

Next, I edited **/etc/group** manually to change the **nims** group ID to 200 (this group was automatically created by `adduser nims`):

```
nims:x:200:
```

This signifies that **nims** is a system user, and prepares us for future usage of groups for filesystem permissions. Finally,

```
sudo chown nims:nims /home/nims
```

to fix the mess we just made with user and group ID.

3 Configuration

To set timezone:

```
dpkg-reconfigure tzdata
```

and follow the menu prompts. I set this to Los_Angeles, which is the tzdata name for Pacific time. Next, to install NTP support:

```
apt-get install system-config-date
apt-get install ntp
sudo system-config-date
```

Edit **/etc/ntp.conf** to include

```
server time.apple.com iburst
server 130.20.248.2 iburst prefer
server 130.20.128.83 iburst prefer
```

since our network blocks outside NTP servers. Note that this will likely need to change for deployment; ideally, we'd have a GPS receiver onboard. Reboot or restart the ntp service after changing the config file.

4 Startup

The following script in **/etc/init.d** is used to start the NIMS binaries.

```
#!/bin/sh

# System Startup for NIMS
```

```
# to be installed in /etc/init.d

APPLICATION_PATH=/home/nims/bin

killproc() {
       pid=‘pidof $1‘
       [ "$pid" != "" ] && kill -9 $pid
}

case $1 in

   start)
      echo "Starting NIMS applications."
      if [ -e $APPLICATION_PATH/nims ]; then
          pushd .
          cd $APPLICATION_PATH
          $APPLICATION_PATH/nims &
          popd
      fi
      ;;

   stop)
      echo "Stopping NIMS (Normal Mode)"
      killall nims
      ;;

   *)
     echo "Usage: nims.sh {start|stop}"
     exit 1
     ;;
esac

exit 0
```

To activate it, we use

```
sudo service nims start|stop
```

from the command prompt. Eventually we'll set it to run at boot with

```
# set to start in runlevels [2-5] at S99, stop in [0,1,6] at K1
# was using S99 only in testing (K number = 100 - S number)
/usr/sbin/update-rc.d nims start 99 2 3 4 5 . stop 1 0 1 6 .
```

or similar.