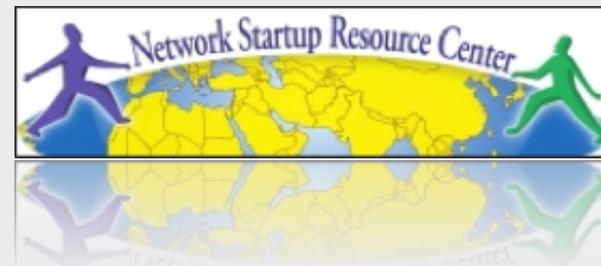


# System Administration: Linux

**Track 2 Workshop  
June 2010  
Pago Pago, American Samoa**



# Concepts to be Covered

We will do numerous exercises to reinforce these ideas:

- Run levels
- Terminal and Console Access
  - Out of Band (OoB)
- Configuring Your Network Interfaces
- RAID
- Backup Strategies
- Deployment
- System Monitoring
- Using SSH (intro)



# Classic Linux Runlevels

Looks like this:

ID	Name	Description
0	Halt	Shuts down the system.
S	Single-User Mode	Does not configure network interfaces or start daemons. <sup>[2]</sup>
6	Reboot	Reboots the system.

ID	Name	Description
1	Single-User Mode	Does not configure network interfaces, start daemons, or allow non-root logins. <sup>[3]</sup>
2	Multi-User Mode	Does not configure network interfaces or start daemons. <sup>[4]</sup>
3	Multi-User Mode with Networking	Starts the system normally. <sup>[5]</sup>
4	Not used/User-definable	For special purposes.
5	X11	As runlevel 3 + display manager.

For more details see: <http://en.wikipedia.org/wiki/Runlevel>



# How do we Use This?

In modern Linux we largely don't, except for:

- init 0
- init 1
- init 5 (or “exit” from runlevel 1)
- init 6

See “man init” for more details.

- init 0 → Stop our system
- init 1 → Single User Mode or “emergency mode”
- init 5 → Day to day use
- init 6 → Reboot our server



# How do we Use This?

## What happens at each runlevel?

- **init 0** → Links in /etc/rc0.d are executed.
- **init 1** → Links in /etc/rc1.d are executed.  
Login as root user only.  
Minimal file system access.
- **init 5** → Links in /etc/rc5.d are executed.  
Gui is started.  
Day-to-day working state.
- **init 6** → Scripts in /etc/rc6.d are executed.

# With Ubuntu We *Actually* Do...

## What happens at each runlevel?

- **init 0** → Links in /etc/rc0.d are executed.
- **init 1** → Links in /etc/rc1.d are executed.  
Login as root user only.  
Minimal file system access.
- **init 2-5** → Links in /etc/rc5.d are executed.  
Gui is started.  
Day-to-day working state.
- **init 6** → Scripts in /etc/rc6.d are executed.

# With Ubuntu We *Actually, Actually* Do...



As of Ubuntu Server 9.10 the System V init system has largely been replaced with an asynchronous initialization system called *upstart*.

Upstart is written/created by Canonical, the makers of Ubuntu:

<http://en.wikipedia.org/wiki/Upstart>

<http://upstart.ubuntu.com/>



# What Starts X/X.Org/Gnome/KDE?

- Previously GUI started at runlevel 5.
- Default runlevel was defined in `/etc/inittab`
- For runlevel 5 in `/etc/inittab` X/X.Org (GUI) was often started.
- Now we define default runlevel in `/etc/init/rc-sysinit.conf`
- Now we define if/how Gnome (KDE, etc.) start in `/etc/init/gdm.conf`

# It's All Pretty Cool

You have complete control over your system!

The default configuration is very likely all you need, but you can change system bootup and behavior *exactly* as needed if you wish.

**Now we'll look at a short Terminal and Consoles presentation and exercise.**



**This page intentionally blank**



# Configuring Your Network Interfaces

You can do this manually:

```
# ifconfig eth0 192.168.100.10 netmask 255.255.255.0  
(Anyone know what size network this is?)
```

You might need a gateway...

```
# route add default gw 192.168.100.1  
(We'll describe this in detail later)
```

Upon reboot your changes will be lost



# Configuring Your Network Interfaces

To make changes stick edit:

```
/etc/network/interfaces
```

For one of our classroom PCs:

```
# The primary network interface
auto eth0
iface eth0 inet static
    address 67.218.55.101
    netmask 255.255.255.192
    network 67.218.55.64
    broadcast 67.218.55.127
    gateway 67.218.55.65
    dns-nameservers 67.218.55.67
    dns-search pacnog.bluesky.as
```



# Virtual Interfaces or Network Aliases

You can make a single NIC look like 2, or more...

```
# ifconfig eth0:0 192.168.100.10 netmask 255.255.255.0
```

What's the "eth0:0"?

How you can bind multiple IP addresses to one NIC on a host.

To make this permanent...



# Network Aliases cont.

In the file `/etc/network/interfaces`:

```
# The primary network interface
auto eth0
iface eth0 inet static
    address 67.218.55.101
    netmask 255.255.255.192
    network 67.218.55.64
    broadcast 67.218.55.127
    gateway 67.218.55.65
    dns-nameservers 67.218.55.67
    dns-search pacnog.bluesky.as
```

```
auto eth0:0
iface eth0:0 inet static
    address 192.168.100.10
    netmask 255.255.255.0
    gateway 192.168.1.1
```

See “Man interfaces” for more information



# Network Configuration

Telling your system where to look for DNS:

```
/etc/resolv.conf
```

Telling your system what your host's name is:

```
/etc/hosts
```

```
/etc/hostname
```

The file `/etc/hosts` is “poor man’s DNS”



# Exercises

# RAID and Backup

**Separate presentation**



# Deployment

## Deploying lots of servers...

- Should we do this by hand?
- You could use “the cloud”
  - Google
  - Amazon
  - Cisco
  - others...
- Virtual images:
  - VMWare
  - VirtualBox (truly free, so far)
  - Citrix
  - Parallels
  - others...



# Deployment Continued

## Deployment tools:

- rsync
- G4L (Ghost 4 Linux)  
<http://sourceforge.net/projects/g4l/>
- Cfengine  
<http://www.cfengine.org/>
- Puppet  
<http://www.puppetlabs.com/>

The SourceForge logo, featuring the word "source" in black and "forge" in white on a blue rectangular background.

# System Monitoring

A massive topic!

Bottom line – something needs to be monitoring your box in some way so that you know when something unusual happens...

Logging? Requires secondary tools to parse your logs and notify you of events.



# System Monitoring Continued

A few possibilities

Simple to verify services are still up and running:

- Nagios
- Cacti
- Munin
- OpenNMS
- Zabbix
- Home grown scripts

Notification is a bit trickier. Nagios is very good at this.

# System Monitoring Continued

Monitoring for other items...

What about:

- System load
- Responsiveness
- Disk space
- RAID volume in degraded state
- Excessive processes
- Overheating
- etc

Previous tool set can do this. Generally will need installed client, snmp service or scripts using ssh.



# SSH – Your most important tool

To admin a Linux/Unix box ssh is *the* tool. With ssh you can:

- Connect securely and remotely to boxes you administer.
- Connect to boxes on private networks using ssh tunneling tricks.
- Connect using public/private key pairs to avoid the use of passwords.
- Write scripts to back up your boxes remotely and automatically.
- Create simple, easy-to-use admin tools:
  - We have done this for this class and will demonstrate now.



# SSH – Your most important tool

We'll do more in-depth SSH exercises later this week.

For now, we are setting the tone:

*SSH is how you work as a system administrator in the world of Linux and Unix.*

