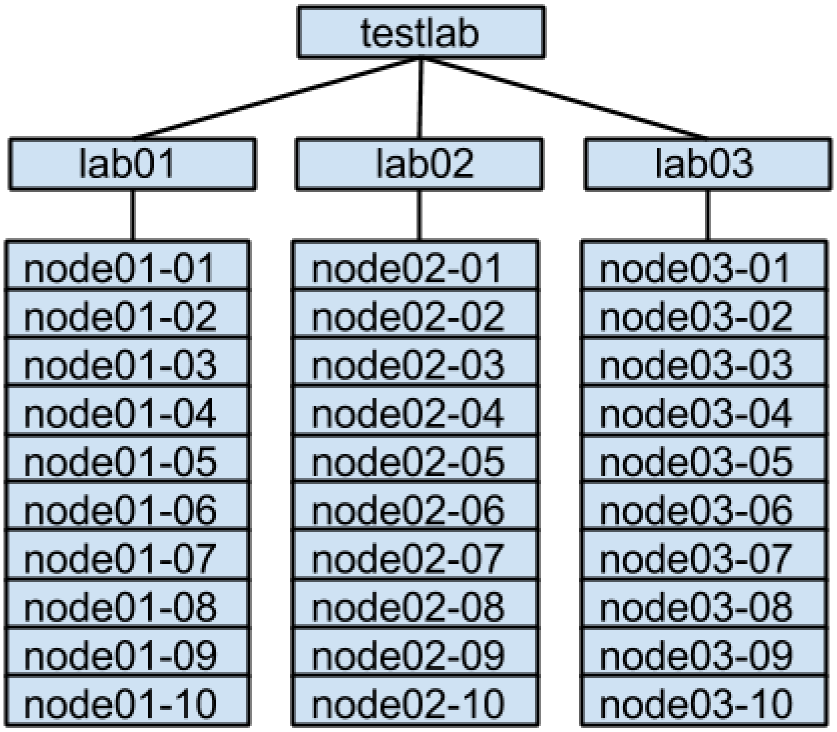
Test Lab

1.0 - Introduction to the Test Lab Linux Cluster Environment

Test lab is a set of three disposable cluster environments that can be used for systems research. All three environments are accessible through their respective head nodes once you have logged into the main head node testlab.its.iastate.edu. Each lab environment consists of a lab head node (lab01, lab02, lab03) which in turn is connected to a private network containing 10 compute nodes.



All lab systems are running CentOS 6.6 and a local repository of the install packages has been created on testlab. We decided to use CentOS instead of RHEL to eliminate the need for registration with the Red Hat Network.

When a user is assigned an environment on testlab (e.g. lab01, lab02, lab03) they will be given an ssh key to login to the environment as root and the IPMI login and password for their set of systems.

Lab head nodes and compute nodes on testlab are always PXE booted - this allows for easy reinstallation of the operating system in case of a problem. Lab head nodes (lab01, lab02, lab03) receive PXE configs from testlab, and the compute nodes receive PXE configs from their respective lab head nodes. The default PXE config will direct the machines to boot from local disk but gives the option to do a fresh install. The PXE boot menu is accessible by connecting to the IPMI console over ssh, <hostname>-ipmi. Further instructions for connecting to IPMI will be provided in the next section.

2.0 - Initializing Lab Environment

When you are assigned a test lab user environment, the lab head node and compute nodes should already be initialized to a default CentOS 6.6 installation. However, you may wish to modify this setup or reinitialize the environment to ensure that it is running a clean, default configuration. This section will walk you through the steps of this initialization/reinstallation process. This example is written for the “lab01” environment, but can be easily adapted to the “lab02” or “lab03” environments.

2.1 - Intelligent Platform Management Interface (IPMI)

The primary way that users will be administering the hardware in their assigned testlab environment is via the servers’ Intelligent Platform Management Interface, or IPMI. IPMI allows remote management of a server via a network connection. IPMI allows for remotely powering a server on or off, and also allows remote access to the serial console, among many other capabilities. If you are unfamiliar with IPMI, a good place to start is http://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface.

The two primary means of interacting with a server’s IPMI interface are via the command line utility ipmitool (used mainly from a different system for checking status of a remote system and powering on/off, etc), and via an ssh session (which will primarily be used for remote access to a system’s serial console for initial installation and other cases where a standard login shell is unavailable).

The lab head node IPMI interfaces (lab01-ipmi, lab02-ipmi, lab03-ipmi) are available from the main testlab system. The compute node IPMI interfaces are available from their respective lab head node (e.g. node01-03-ipmi is available from lab01, node03-07-ipmi is available from lab03, etc).

Note that while there is an ipmitool command line option to pass the IPMI user password on the command line, for security reasons it is recommended that you do not use this option (as the password will show up in plaintext on the system process list and in your shell command history). It is recommended you allow the remote system to prompt for the password, or alternatively use the ipmitool “-f” command line option and a file containing only your IPMI user password. Make sure the file containing your ipmi password is only readable by you (e.g. `chmod 600 ipmi-password-file`).

Below you will find some IPMI examples. Commands typed in by the user are in **`bold font`**. Note that the shell prompt and any text returned from the shell is displayed with an `underlined` font… do not type in the underlined text. The prompts and returned output may vary slightly from these examples, and some returned output has been left out here for brevity. In the following examples, replace `user` with your assigned testlab username, and replace `ipmi-user` with your assigned ipmi username. You may receive the warning message “Get HPM.x Capabilities request failed, compcode = d4” when you execute an ipmitool command… this message can be safely ignored.

Example 1: Check power status of lab01 from a shell on testlab with ipmitool

```
[user@testlab ~]# ipmitool -I lanplus -H lab01-ipmi -U ipmi-user power status
Chassis Power is on
```

Example 2: Power down node01-03 from a shell on lab01 with ipmitool

```
[user@lab01 ~]# ipmitool -I lanplus -H node01-03-ipmi -U ipmi-user power down
Chassis Power Control: Down/Off
```

Example 3: Power cycle node01-05 from a shell on lab01 with ipmitool (this is a “hard reset” which turns the power completely off and back on again)

```
[user@lab01 ~]# ipmitool -I lanplus -H node01-05-ipmi -U ipmi-user power cycle
Chassis Power Control: Cycle
```

Example 4: Power reset node01-05 from a shell on lab01 with ipmitool (this does a “warm reset” which does not actually turn the power off).

```
[user@lab01 ~]# ipmitool -I lanplus -H node01-05-ipmi -U ipmi-user power reset
Chassis Power Control: Reset
```

Example 5: Connect to remote serial console on lab01 from testlab (note some output has been removed for brevity)

```
[user@testlab ~]# ssh ipmi-user@lab01-ipmi
/SP -> cd /SP/AgentInfo/console
/SP/AgentInfo/console -> start
press ESC ( to terminate session…
lab01.testlab.lan login:
```

Note that you can have only one active connection to the ipmi console at a time. When you are done with the system console, if you were logged in to the compute node make sure you log out before you disconnect. Then disconnect from the compute node IPMI console session with the key sequence “`<esc> (`“ or in other words the escape key followed by the left parenthesis character (pressed in sequence, not at the same time).

2.2 - Re-installing the Lab Head Node

The lab head node will need to be installed and running prior to reinstalling any of its compute nodes, as the compute nodes get their PXE boot configuration information from this node. As mentioned previously, the lab head node will likely be configured already but you can follow this procedure should you choose to reinstall. While this procedure is written for lab01, it is easily adapted to lab02 or lab03. **NOTE: THIS PROCEDURE WILL ERASE ANY EXISTING DATA FROM THE LAB HEAD NODE!**

1. Initiate a remote IPMI serial console connection from testlab to lab01. See (Example 5 from section 2.1).
2. If lab01 is booted up to the OS and you are able to log in, reboot the system from a login shell. Otherwise, use ipmitool to do a "power reset" (see Example 4 from section 2.1).
3. As the lab01 system begins to boot, it will bring up a PXE network boot menu that it receives from testlab. It may have to cycle through more than one network interface before it tries the correct interface to talk with testlab - it may take several seconds or up to a minute for it to timeout and try the next interface. Once the PXE boot menu comes up, select menu item "lab01-install-no-prompt". This option will automatically reinstall the CentOS 6.6 operating system and correctly configure the system as the "lab01" head node.
4. The system will eventually reboot and provide you with a login prompt at the remote serial console. At this point, you should be able to exit the remote serial console and log in via a standard ssh connection from testlab.

2.3 - Re-installing the Compute Nodes

Note again that this section is written with the lab01 cluster in mind, but can easily be adapted to the lab02 or lab03 environments.

Once the lab head node has been correctly installed and configured, the compute nodes that it controls can be reinstalled if desired. This is accomplished using the PXE boot mechanism as well, although the compute nodes will be getting their PXE boot configurations from the lab head node instead of the testlab system. The configuration files that control what boot options are sent to the compute nodes are on the lab head node in the directory `/var/ftpd/pxelinux.cfg`

PXE boot configurations can be modified for each individual client node if desired, and this is generally keyed from the DHCP assigned IP address of the compute node or subnet (although alternate methods such as MAC address or client system UUID are also possible… see http://www.syslinux.org/wiki/index.php/PXELINUX#Configuration). The compute nodes are assigned reserved DHCP addresses by dnsmasq based on each system's hardware MAC address using the `/etc/ethers` and `/etc/hosts` files (on lab01), so each time a compute node reboots it should receive the same reserved DHCP address.

In the `/var/ftpd/pxelinux.cfg` directory on lab01, there will be three separate configuration files named "`default`", "`memtest-console`", and "`node-install-no-prompt`". If there is no customized content for a particular compute node client, it will receive the "default" configuration. Under the "`default`" configuration, the boot process will automatically choose to boot the compute node client from its locally installed operating system, although options are also available in the menu to perform a memtest or reinstall the CentOS 6.6 operating system.

If you wish to send a customized boot menu to a particular system or group of systems, you would create a file or a symlink in the `/var/ftpd/pxelinux.cfg` directory on lab01 with a hexadecimal representation of the DHCP assigned IP address or subnet. For example, if you want to send node01-01 a customized boot menu that defaults to performing an automatic wipe reinstall of the operating system, you could create a copy of the file `/var/ftpd/pxelinux.cfg/node-install-no-prompt` called `/var/ftpd/pxelinux.cfg/AC180A01`. Note that AC180A01 is the hexadecimal representation of 172.24.10.1, which is the IP address assigned to node01-01 via DHCP. Next time you reboot node01-01, it will receive the boot menu from file "`AC180A01`" instead of the "`default`" file, while all other compute nodes will continue to receive the "`default`" version.

Another alternative to creating a separate copy of the `node-install-no-prompt` file would be to create a symlink called `AC180A01` that points to the `node-install-no-prompt` file. This is a good approach to use if you will have several different compute nodes booting from a specific configuration file (in that it avoids multiple copies of an identical file). If you are unfamiliar with symlinks, here is the command you would use:

```
        ln -s /var/ftpd/pxelinux.cfg/node-install-no-prompt /var/ftpd/pxelinux.cfg/AC180A01
```

Note to remove a symlink, you use the command `unlink` instead of the command `rm` that you would use for a regular file.

In this example, we will be reinstalling all compute nodes at the same time, so the easiest way to do this will be to set a customized symlink for our subnet which would be AC180A in hexadecimal (which corresponds to subnet 172.24.10.0/24), and the symlink will point to the node-install-no-prompt file.

```
        ln -s /var/ftpd/pxelinux.cfg/node-install-no-prompt /var/ftpd/pxelinux.cfg/AC180A
```

The next time any compute nodes reboot, they will grab the AC180A PXE configuration and start the automated re-installation process. One challenge with this method is that once each system has finished installing and rebooted, it will again grab the AC180A PXE configuration and begin reinstalling once again. One way to approach this is to use a shell script that uses the ipmitool command within a loop that will do a "power reset" on each compute node so that they all rebuild at roughly the same time. Once you have verified the last node has received its pxe configuration, then you can go ahead and remove the customized symlink from lab01:/var/ftpd/pxelinux.cfg so that after the rebuild each compute node will grab the default configuration which defaults to booting from the local disk instead of doing a reinstallation.

One method of more easily monitoring/managing multiple installations on the compute nodes is to use a utility such as "screen" or "tmux" to open and maintain an ipmi console session to each compute node in the cluster. These utilities have the added benefit of maintaining the remote shell connections even if you need to logout of the lab head node or if there is a network interruption between your system and the lab head node (you can reconnect to the "screen" or "tmux" session once you log back into the lab head node).