

---

# Platform Cluster Manager version 1.2b TM (PCM 1.2b) User Guide

---

**Version 1.2b**

**April 2010**

**Platform Computing**

## **Contents**

- [About Platform Cluster Manager](#)
- [Basic administration](#)
- [Advanced administration](#)
- [Get technical support](#)
- [Copyright and trademarks](#)

[ [Top](#) ]

---

## **About Platform Cluster Manager**

Building a Linux® cluster is a challenging and time-consuming task. There are many tools in the community and on the Internet for building, configuring, and managing Linux clusters. However, these tools typically assume a familiarity with Linux cluster concepts.

Platform Cluster Manager (previously known as OCS - Open Cluster Stack), is a hardware vendor certified, software stack that enables the consistent deployment and management of scale-out Linux clusters. Backed by global 24x7 enterprise support, Platform Cluster Manager is a modular and hybrid stack that transparently integrates open source and commercial software into a single consistent cluster operating environment.

Platform Cluster Manager version 1.2b ("PCM 1.2b") is fully supported by Platform Computing Corporation and requires a Red Hat® based operating system.

## **Where to get Platform Cluster Manager**

You can download PCM 1.2b and find associated documentation at <http://my.platform.com/products/platform-cm>.

If you intend to install other third-party kits, obtain the CD or DVD containing those kits.

To purchase support for PCM 1.2b, contact Platform Computing.

---

## Basic administration

The following topics describe basic tasks and provide useful information for administering your PCM cluster:

- [Installation and Troubleshooting notes](#)
- [Access online documentation](#)
- [View Nagios information](#)
- [View Cacti graphs](#)
- [Switch NIC and view Ntop information](#)
- [Disable SSH forwarding](#)
- [Use PDSH to run parallel commands across nodes in cluster](#)
- [Add, remove, or upgrade kits](#)
- [Add or remove users](#)
- [Understand firewalls/iptables](#)
- [Understand Platform Cluster Manager services and utilities](#)
- [Find log files](#)

## Installation and Troubleshooting notes

For instructions on installing the Platform Cluster Manager, and troubleshooting common issues for Platform Cluster Manager 1.2b and related kits, download the latest PCM Installation Guide and PCM Customer Troubleshooting Guide at <http://my.platform.com/products/platform-cm>.

Here are some installation and upgrade precaution notes:

- Currently, PCM only partitions fully the first disk. External storage should be attached as a second disk and not be involved in the PCM partitioning process.
- To use an external storage for `/home`, fully customize the PCM schema in the installer node or use `NGEDIT` tools that allow you to edit the default schema to customize your needs. The PCM installer does not create any special user so you can mount an external partition over `/home`. Choose the **Use Existing** schema and create a schema from scratch. Be careful and ensure that the external storage is marked as **Preserve**. Otherwise, PCM will reformat the partition. A safer option is not to define `/home`. Manually edit `/etc/fstab` and mount the external storage as `/home` post install.
- If the compute node already has a Volume Group which spans across multiple HDDs, compute node installation stops and displays error message that it cannot be partitioned because the schema for the node would cause data loss. PCM tries to format only the first HDD (`sda1`) and it stops the installation if a VG, which spans across `sda`, `sdb`, etc., exists. The additional drives could belong to a networked storage and formatting could result in data loss. Ensure that there is no VG that spans across multiple hard disks prior to installing the compute node.

For more details on special installation and operational tips and solutions, refer to the PCM Customer Troubleshooting Guide.

## Access online documentation

Online documentation is provided online by the installer node. To access the documentation, open a browser on the installer node. It defaults to the Cluster page, which contains links to the following guides:

- Kit user guides
- Platform Cluster Manager User Guide (this guide)

- Project Kusu Homepage

Kit-specific documentation is available from the Installed Kits link. Follow the corresponding documentation links beside the kit.

## View Nagios information

Nagios is a NMS (Network Management Server) that monitors hosts and services you specify. It alerts you if problems occur within customizable thresholds you specify.

Nagios runs on the installer node and provides a web GUI to display the information collected from the nodes. Each compute node contains a daemon called NRPE that the installer node calls to collect data. This data is displayed in the Nagios web GUI showing very detailed statistical data as well as graphs. When compute nodes are added to a PCM cluster, Nagios monitoring is installed and configured on the nodes and the PCM 1.2b installs node re-configured to detect the new nodes.

To view Nagios information, go to the main cluster web page, click the "Nagios(R) Network Management System Monitoring" link, or point your browser to `https://localhost/nagios` (on the installer node).

## View Cacti graphs

Cacti is a complete network graphing solution built upon RRDTool (Round Robin Database Tool). PCM uses Cacti to collect and graph cluster data such as number of users logged in, load average on the nodes, available memory and swap, etc. Cacti is extensible and new metrics can be added to Cacti if needed.

Cacti displays information for each node in the cluster and is configured to group the nodes into their respective nodegroups. To view Cacti graphs, launch a local browser from the monitoring node and go to `http://primary_install_host_name/cacti/`

## Switch NIC and view Ntop information

Ntop is a network traffic analyzer designed to show the administrator the different protocol traffic passing through the installer node. Ntop can also show network traffic patterns to better diagnose network problems and network utilization issues.

By default, Ntop is configured for private traffic with one interface always listening. To switch on the network interface Ntop should listen on, navigate to **Plugins > Netflow > View/Configure**. The **Netflow Device Configuration** appears. Click **enable** and then click the **Admin** menu option then select **Switch NIC**. A new screen appears where you can select which network interface to listen on.

Ntop provides several plug-ins that can be enabled or disabled for further analysis of the traffic. See the plugins page within Ntop for more details.

To view the Ntop page, go to the main cluster web page and then click **Ntop Cluster Monitoring (SSL)**.

## Disable SSH forwarding

By default, the OpenSSH daemon is configured to enable X11 forwarding. This can sometimes slow down node connections. You can disable forwarding by using the `-X` option when connecting to a node to skip X11 forwarding.

This can also be disabled permanently by editing the `/etc/ssh/ssh_config` file and changing the line `ForwardX11 Yes` and setting this to `No`.

An SSH connection from one node to another may be slow in setting up. This is usually because of a name resolution failure, and subsequent timeout. This can occur if the installer node was installed with an invalid DNS server.

---

**Note:** This will also slow MPI jobs.

---

## Use PDSH to run parallel commands across nodes in cluster

PCM 1.2b comes with Parallel Distributed Shell (pdsh) installed and configured for operation in the cluster. As new compute nodes are added or removed from the cluster, PCM automatically generates the required configuration files for pdsh to ensure that parallel commands can be issued across all nodes in the cluster. By default, pdsh is configured to use ssh as the underlying shell mechanism for connecting to nodes. The PCM `genconfig` tool dynamically creates the list of nodes in the cluster and saves the list in `/etc/hosts.pdsh`. This file can be used to run parallel commands across all nodes in the cluster. For example:

Running a command on a node in the cluster

```
# /usr/bin/pdsh -w compute-00-00 uname -a
```

Running a command on all nodes in the cluster (in this example there is only one node in the cluster)

```
# /usr/bin/pdsh -a uname -a
```

Optionally, you can specify a static host list instead of using the dynamic host list. This means pdsh commands affect selected nodes instead of all the nodes in the cluster. To specify the nodes, make a copy of `hosts.pdsh`, modify it, and save under any name. To make the change take effect and use your modified file in place of `hosts.pdsh`, you must export the path to the modified file. Run:

```
# export WCOLL=path_to_modified_hosts_file
```

---

**Note:** Running pdsh sometimes produces an ssh exit message (for example, "ssh exited with exit code xx"). This simply tells you that command did not finish normally. When a command finishes on UNIX systems, it returns an exit status - normally this status is 0. A status of 0 means that the command finished and exited normally. When a command does not finish normally, it exits with a non-zero value which pdsh shows you.

---

## Add or remove kits

PCM 1.2b provides kits with a mechanism for packaging applications and installation scripts together for easy installation onto a PCM 1.2b cluster. A PCM 1.2b kit contains the following:

- A kit info defining kit information (such as kit version, release number, noarch, etc) and components.
- A meta-rpm containing the documentation for the kit and default nodegroup associations.
- Component rpms containing a list of rpm dependencies and installation/removal scripts for a component
- rpms - containing the actual software.

Kits can contain multiple components and components have a list of dependencies on other components or rpms. The kit mechanism provides PCM 1.2b with the ability to install and automatically configure applications or tools onto the entire cluster.

- Kits collect components into a simple package and components add a extra layer of dependency abstraction to the standard rpm mechanisms.
- Kits use the native package management system to resolve dependencies. For Red Hat based operating systems, PCM 1.2b relies on yum to resolve all dependencies in a kit.

### Add kits using the "kitops" command

The PCM 1.2b `kitops` command adds kits to a PCM 1.2b system. Everything in PCM 1.2b is a kit including operating systems such as Red Hat Enterprise Linux. The base kit and operating system kit are added during installation. These kits are used to create the repository and build the install node. In order to add more kits for operating systems or applications, use the `kitops` command.

```
# [root@master ~] # kitops -h
```

```
usage: KUSU kitops - kit operations tool
```

```
options:
```

```
-h, --help          show this help message and exit
-a, --add           add the kit specified with -m
-l, --list         list all the kits available on the system
-e, --remove       remove the specified kit
-m MEDIA           specify the media iso or mountpoint for the kit
-k KIT, --kit=KIT  kit
-o KITVERSION, --kitversion=KITVERSION
                  kitversion
-c KITARCH, --kitarch=KITARCH
                  kitarch
--dbdriver=DBDRIVER Database driver (sqlite, mysql)
--dbdatabase=DBDATABASE
                  Database
--dbuser=DBUSER   Database username
--dbpassword=DBPASSWORD
                  Database password
-y, --yes         Assume the answer to any question is Yes
-v, --version     Display version
```

### Add an operating system kit

Adding a new operating system to PCM 1.2b is very simple. All you need is an ISO or disk containing the operating system you want to add. When PCM 1.2b detects a new operating system media, it automatically creates an OS kit from the media and installs it on the cluster. For example:

#### Add an OS kit to a PCM 1.2b cluster

Insert OS kit into CD/DVD then run the following command:

```
# kitops -a -m /media/CDROM/ --kit=rhel
```

### Add a tool or application kit

The `kitops` command can also add pre-packaged tools and applications to the cluster. PCM 1.2b comes with the following application kits:

- **Base** - Contains PCM 1.2b tools and applications for managing and using the

cluster.

li>**Dell** - Installs drivers and utilities for Dell  PowerEdge servers. It provides support for IPMI and Dell OpenManage  (OM).

- **Cacti** - Installs the Cacti open-source reporting tool. Configure Cacti to monitor and collect node metrics.
- **Console** - Installs the Platform Management Console to allow PCM administrators to administer, monitor, and manage workload via a GUI web-console. It allows users without administrative rights to submit and control the workload manager (Lava).
- **Ganglia** - Installs the Ganglia open-source package for monitoring node availability, and displaying system load, network usage, and other resource information for high performance computing (HPC) clusters.
- **HPC** - Contains tools, libraries, and utilities for high performance computing clusters. Includes libraries such as MPICH1 and 2, MVAPICH1 and 2, OpenMPI, Linpack and various other benchmarking tools and utilities.
- **Lava** - Installs Platform Lava Open Source Batch system for running jobs on the cluster
- **LSF** - Installs the Platform LSF application for managing and accelerating high performance computing (HPC) mission-critical workload, and intelligently scheduling parallel and serial workload.
- **MPI** - Installs the Platform MPI (PMPI) application and its license manager component on your PCM 1.2b cluster to support the centralized licensing mode of PMPI. It allows you to take advantage of leading interconnect technologies to build high performance applications and support burden of multiple software distributions.
- **Nagios** - Installs the open source Nagios monitoring package on your PCM 1.2b cluster. Nagios is an NMS (Network Management Server) that runs on the installer node and monitors your cluster hosts, services, and network.
- **Ntop** - Installs a tool to monitor network bandwidth and analyze traffic. Ntop allows you to examine the network patterns of your PCM 1.2b cluster.
- **Mellanox** - Installs and configures MPI software and OFED hardware drivers (which support server and storage clustering and grid connectivity, and allows the configuration of IP over IB), and updates the firmware for any available Mellanox cards.
- **OFED** - Automatically configures OFED in the cluster. It includes OFED drivers, which support server and storage clustering and grid connectivity, and allows the configuration of IP over Infiniband.

Adding an application kit to the cluster is very similar to adding an operating system kit. For example:

```
# kitops -a -m kit-ntop-3.3-14.x86_64.iso
Added Kit ntop-3.3-noarch
```

#### Add a single kit or many kits in a meta-kit

```
kitops -a
[0]: base-5.1-noarch
[1]: cacti-0.1-x86_64
[2]: hpc-0.1-x86_64
[3]: lava-1.0-noarch
[4]: ntop-3.3-noarch
Provide a comma separated list of kits to install,
'all' to install all kits or ENTER to quit::
```

#### Add kit to a repository

To add kit to a repository on the installer node, use the `repoman` command.

1. Determine which repository, on the installer node, you want to add the kit. To do this, list the repositories currently in the system:

```
# repoman -l
```

2. Add the kit to the desired repository:

```
repoman -r <repository name> -a -k <kit_name>
```

3. Refresh the repository:

```
# repoman -r <repository name> -u
```

## Remove a kit

Kits can be removed from PCM 1.2b, however if a kit is in use by a repository or nodegroup PCM 1.2b will not remove it from the cluster.

To remove a kit from PCM 1.2b, run the following command:

```
# kitops -e -k <kit_name>
```

```
+-----+-----+-----+
| Kit   | Version | Architecture |
+-----+-----+-----+
| ntop  | 3.3     | noarch       |
+-----+-----+-----+
The above kits will be removed.
```

```
Confirm [y/N]:y
```

## Understand kits, repositories and nodegroups

After adding kits to a PCM 1.2b installer node, you need to run the `repoman` command to make the kit available to the repository. If a new repository is created and new nodegroups are associated with the repository then you will need to add all the kits to the new repository and then associate the components in the kits with the appropriate nodegroups. Refer to [Creating a New Repository](#) and [Associating kit components to nodegroups](#) for more information.

### Disable a kit and components

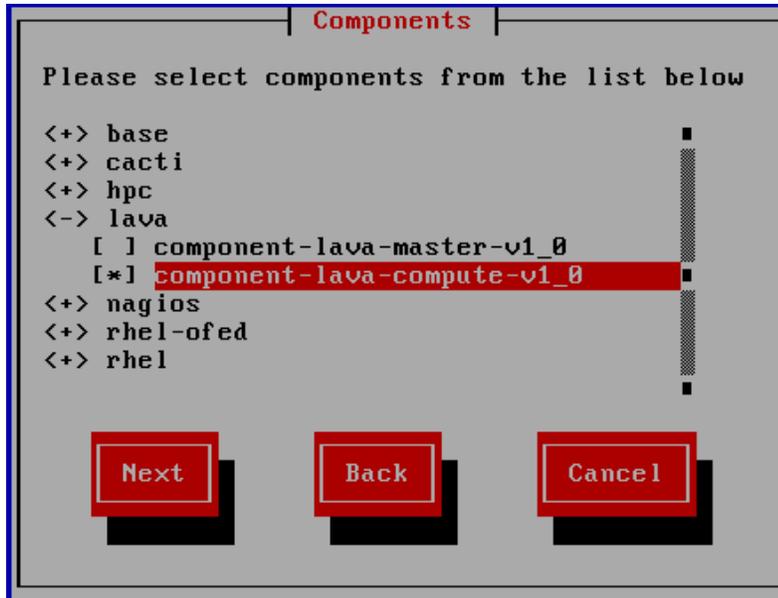
To disable a kit or to prevent components in a kit from being installed on a compute node, follow these steps:

1. Run `ngedit` as root.
2. Select the nodegroup (or the nodegroup you wish to edit) and select **Next**.
3. Select **Next** until you arrive at the Components screen.
4. Expand the component you wish to remove using the spacebar.
5. Clear the component selection field with the space bar.
6. Proceed through the rest of the `ngedit` screens by selecting **Next**.

Your changes are confirmed and applied.

The kit is now disabled from the nodes in the nodegroup.

Figure: Ngedit display disabling Lava compute node component



## Add or remove users

PCM 1.2b is configured by default to share all of the user names and passwords defined on the installer node across all nodes and nodegroups in the cluster. Use the standard Linux command line or GUI tools to add a user to PCM 1.2b. Once the user is added, the PCM 1.2b cfm tool (Configuration File Manager) must be called to synchronize the user names and passwords across the cluster.

### Add a user from the command line

- To specify the password on the same command-line:

```
# adduser -p <encrypted_password> <user_name>
# cfmsync -f
```

- To specify the password using the `passwd` command:

```
# adduser <user_name>
# passwd <user_name>
# cfmsync -f
```

---

**Warning:** You must run the `cfmsync` command prior to rebooting the installer; otherwise, user changes might be lost.

---

### Remove a user from the command line

To remove a user, run the following command:

```
# userdel <user_name>
# cfmsync -f
```

## Remove a user from the GUI

- Start the GUI from System->Administration->Users and Groups
- Remove the user with the GUI interface
- Exit the GUI
- Run cfm from a terminal:

```
# cfmsync -f
```

## Understand firewalls/iptables

The installer node is configured with some basic forwarding rules. From a network security standpoint, the installer and compute nodes are not secure. Evaluate the security risks at your site and create appropriate firewall rules to secure the cluster.

---

**Warning:** The installer node should never be connected to the Internet without first restricting the type of packets allowed by customizing the iptables rules.

---

The installer node is configured with network address translation (NAT), allowing compute nodes access to the public network. By default, nodes on the public network do not have a route to the provision network.

HTTP and HPPTS are enabled by default. To change this and keep services visible only to the private network, edit `/etc/sysconfig/iptables`.

### Disable HTTP and HTTPS over the public network

1. In the `/etc/sysconfig/iptables` file, find and comment the following lines:

```
-A INPUT -i eth1 -m state --state NEW -p tcp --dport 80 -j ACCEPT
-A INPUT -i eth1 -m state --state NEW -p tcp --dport 443 -j ACCEPT
```

2. Then, restart iptables:

```
# service iptables restart
```

For details on customizing your firewall, see <http://www.netfilter.org>.

## Understand PCM 1.2b services and utilities

### cfm

Platform HPC provides a service called cfm. This is very similar to NIS. It is used to synchronize files across a cluster. This is done via multicasting a notification of change from the installer node, then having the nodes download the file over an encrypted channel. Users and groups are one example of information passed over cfm. Cfm can also synchronize yum repositories in the cluster. Each node in the cluster has a yum repository and when notified via `cfmsync` the nodes will automatically update from the installer node repository using the `httpd` server. Whenever you run `useradd` or `userdel`, `cfmsync` must be run to update the user information on all nodes in the cluster.

By default the following files are propagated throughout all nodegroups in the cluster by cfm:

- `/etc/passwd`
- `/etc/shadow`

- /etc/group
- /etc/services
- /etc/fstab
- /etc/hosts
- /etc/hosts.equiv
- /etc/ssh/\* files (ssh key files and signatures)

## DHCP and TFTP

PCM 1.2b uses DHCP and the TFTP services to handle installation or reinstallation of nodes. The services are automatically configured when running the addhost tool. Use addhost to configure the DHCP settings for each node.

### pdsh

This shell provides the ability to execute commands on a cluster wide basis. For example:

```
# WCOLL=/etc/hosts.pdsh;export WCOLL
pdsh uname
compute-00-00-eth0: Linux
compute-00-01-eth0: Linux
compute-00-02-eth0: Linux
```

### boothost

The boothost command manages the PXE files, kernel, initrd and boot state of nodes in a PCM cluster. The boothost tool reads cluster and node information from the PCM 1.2b database and constructs PXE files for each node. The files are stored in /tftpboot/kusu/pxelinux.cfg/, one file per unique MAC address. Boothost can override the default kernel and initrd settings for a node or nodegroup. Overriding kernels and initrd can be used to test custom kernels and initrds without affecting all of the nodes in a cluster. The boothost command is also used by the PCM administrator to trigger a re-install of a node or a whole nodegroup. Examples:

List Boot information:

```
# boothost -l
```

Rebuild the PXE configuration files for nodes compute-00-00 and compute-00-01:

```
# boothost -m compute-00-00,compute-00-01
```

Reboot all of the nodes in the RHEL 5 nodegroup:

```
# boothost -r -n compute-rhel
```

## Find log files

PCM 1.2b generates the following logs:

- Kit installation logs
- System logs

System logs may be found in the `/var/log` directory. Some important logs include:

- System log (`/var/log/messages`)
- Installation log (`/var/log/anaconda.log`)
- Kernel log (`/var/log/dmesg`)

- Web server logs (`/var/log/httpd/*_log`)
- Xorg log (`/var/log/Xorg.0.log`)

[ [Top](#) ]

---

## Advanced administration

The following topics describe advanced tasks when administrating your PCM 1.2b:

- [Manage nodegroups](#)
- [Manage repositories](#)
- [Add nodes to the cluster as diskless and imaged](#)
- [Connect to NFS servers](#)
- [Append external nodes information to genconfig hosts output](#)
- [Manage network interfaces](#)
- [Synchronize files in the cluster](#)
- [Move hosts between nodegroups](#)
- [Update a repository](#)
- [Create a new repository](#)
- [Update the installer node and compute node](#)
- [PCM 1.2b NIS/LDAP authentication](#)
- [Backup and restore PostgreSQL database in newer clusters](#)
- [Backup and restore MySQL database in older clusters](#)
- [Configure a dedicated logging server in PCM cluster](#)
- [Configure add-on NIC as provisioning network](#)

### Manage nodegroups

PCM 1.2b is built around the concept of nodegroups. Nodegroups are a powerful template mechanism that allows the cluster administrator to define common shared characteristics among a group of nodes. PCM 1.2b ships with a default set of node groups for, installer nodes and packaged installed compute nodes. The default node groups can be modified or new nodegroups can be created from the default nodegroups. All of the nodes in a node group share the following:

- Node name format
- Operating system repository
- Kernel parameters
- Kits and components
- Network configuration and available networks
- Additional rpm packages
- Custom scripts (for automated configuration of tools)
- Partitioning

A typical HPC cluster is created from a single installer node and many compute nodes. Normally compute nodes are exactly the same as each other with just a few exceptions, like the node name or other host specific configuration files. A nodegroup for compute nodes makes it easy to configure and manage 1 or 100 nodes all from the same nodegroup. The **ngedit** command is a graphical TUI (Text User Interface) run by the root user to create, delete and modify nodegroups. The **ngedit** tool modifies cluster information in the Postgres database and also automatically calls other tools and plugins to perform actions or update configuration files automatically. For example, modifying the set of packages associated with a nodegroup in **ngedit** automatically calls **cfm** (configuration file manager) to synchronize all of the nodes in the cluster using **yum** to add and remove the new packages, while modifying the partitioning on the nodegroup notifies the

administrator that a re-install must be performed on all nodes in the cluster in order to change the partitioning on all nodes. The Postgres database keeps track of the nodegroup state, thus several changes can be made to a nodegroup simultaneously and the physical nodes in the group can be updated immediately or at a future time and date using the **cfmsync** command.

### Reinitialize a partition table

A known issue in Red Hat is that Anaconda raises an error when a system with one or more uninitialized disks is provisioned using a kickstart file. The kickstart installation halts and shows an interactive message when provisioning a RHEL compute node in a new disk or zero out partition. When you have a corrupted partition table, a workaround is to reinitialize the partition on the system by using RHEL install disk, in rescue mode (`boot: linux_rescue`), to successfully proceed installing the compute nodes with PCM 1.2b. For example:

1. To reinitialize a partition table, run:

```
fdisk dev/sda
```

2. Type `c` to create a new empty DOS partition table.
3. Press `w` to write.

The partition table is reinitialized. Reboot and proceed with the compute node installation.

### Create a new nodegroup

1. Open a Terminal and run the Node Group Editor as root.

```
# ngedit
```

2. From the list, select an existing nodegroup that is similar to what you want to create (the new nodegroup is only initially based on the selected nodegroup; you will later edit it, as required).
3. Scroll to the bottom of the Node Group Editor page and then select **Copy**.
4. Edit the newly created nodegroup as desired. For example, you may want to rename the nodegroup or change the format that the machine uses when naming new nodes based on this group.

---

**Tips:**

N= Node number (automatic)

NN= Double node-numbering used (for example, 01, 02, 03, etc.)

R= Rack number

RR= Double rack-numbering used

---

5. Make any other required changes, or leave the default settings as-is.

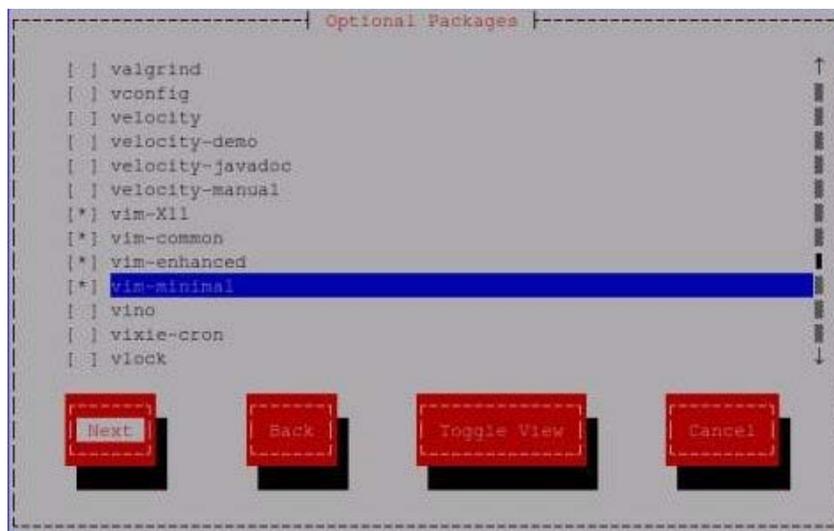
### Add RPM packages in RHEL to nodegroups

1. Open a Terminal and run the Node Group Editor as root:

```
# ngedit
```

2. Select a nodegroup and move through the Text User Interface screens by pressing F8 or by selecting **Next** on the screen. Stop at the Optional Packages screen.

Figure: Optional Packages screen



3. Add additional rpm packages by choosing the package in the tree list.

Press the space bar to expand or contract the list to display all of the available packages. By default, packages are sorted alphabetically. To re-sort the list of packages by groups, select **Toggle View**. Choose additional packages using the spacebar. When a package is selected, an asterisk displays beside the package name.

Package dependencies are automatically handled by *yum*. If a selected package requires other packages, they are automatically included when the package is installed on the cluster nodes. **Ngedit** automatically calls **cfm** to synchronize the nodes and to install the new package--this synchronizes but does not automatically remove packages from nodes in the cluster (this is by design). If required, **pdsh** and **rpm** can be used to completely remove packages from the rpm database on each node in the cluster.

### Add RPM packages not in RHEL to nodegroups

Red Hat HPC maintains a repository containing all of the rpm packages that ship with Red Hat Enterprise Linux. For most customers, this repository is sufficient. Rpm packages that are not in Red Hat Enterprise Linux can also be added to a Red Hat HPC repository by placing the rpms into the appropriate `/depot/contrib` directory. For example:

1. Starting with the rpms that are not in Red Hat Enterprise Linux or in a Red Hat HPC kit, create the appropriate subdirectories in `/depot/contrib`:

```
# cd /depot/contrib
# mkdir -p rhel/5/x86_64
```

```
# cp foo.rpm /depot/contrib/rhel/5/x86_64
```

2. Rebuild the Red Hat HPC repository with **repoman**:

```
# repoman -u -r rhel-<5.x>-x86_64
```

It will take some time to rebuild the repository and associated images.

3. Run **nqedit** and navigate to the Optional Packages screen.

4. Select the new package by navigating within the package tree and using the spacebar to select.

5. Continue through the **nqedit** screens and either allow **nqedit** to synchronize the nodes immediately or perform the node synchronization manually with **cfmsync** at a later time.

Figure: Example--Selecting an rpm that is not included in Red Hat Enterprise Linux



The **contrib** directory may not exist in **/depot/**. If it does not exist, create the directory. Contributions can be added to more than one Red Hat HPC repository. The directory structure is as follows:

```
/depot/contrib/<os_kit_name>/<version>/<architecture>
```

For example, adding contributions to a RHEL repository requires the following directory structure in **/depot/contrib**:

```
/depot/contrib/rhel/5/x86_64
```

### Add RHEL repository to the installer node

Adding other operating systems to Red Hat HPC requires a few steps. To add RHEL to the installer node, you will need a copy of its media or iso.

1. Add the OS using the **kitops** command, mounted on **/media/CDROM** to Red Hat HPC:

```
# kitops -a -m /media/CDROM/ --kit=rhel
```

Adding a kit to Red Hat HPC makes the software available for use in a repository.

2. Create a RHEL repository.

```
# repoman -n -r rhel5.4
```

3. Add the required operating system kit to the repository.

```
# repoman -a -r rhel5.4 -kit=rhel
```

4. Add the Red Hat HPC base kit to the repository. The base kit contains all of the tools required by Red Hat HPC for managing the cluster.

```
# repoman -a -r rhel5.4 -kit=base
```

The operating system and base kits are always required in a repository. At this point the repository can be used to install nodes or you can add more kits to the repository.

5. Rebuild the repository with the new operating system and base kit.

```
# repoman -u -r rhel5.4
```

Congratulations, you have added a new repository to your cluster. View the available repositories with the following command:

```
# repoman -l
```

### Associate a repository with nodegroups

A single Red Hat HPC installer node can contain more than one Red Hat operating system repository. Adding a new operating system to Red Hat HPC involves several steps:

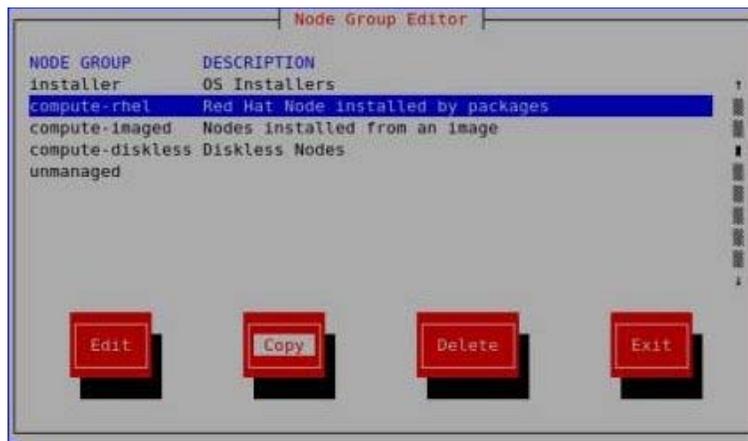
1. Add RHEL operating system CDs/DVD/iso as a Red Hat HPC kit with this command: `kitops .`
2. Create a new repository for RHEL with this command: `repoman -r .`
3. Add the RHEL kit to the new repository with this command: `repoman -a .`
4. Add the Red Hat HPC base kit to the repository with this command: `repoman -a .`
5. Update the repository with this command: `repoman -r .`

This assembles all of the kits into a complete repository.

Once steps 1-5 are completed, the new repository can be added to nodegroups with the `ngedit` tool.

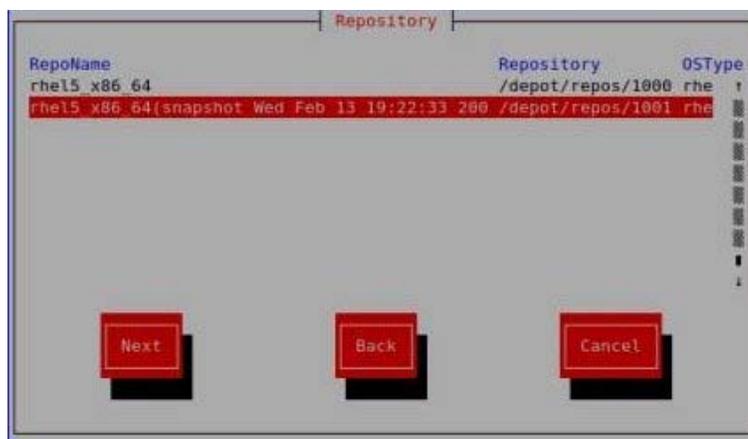
6. Run `ngedit` from a terminal, and create a copy of an existing nodegroup. In our example we will copy the `rhel-compute` nodegroup.

*Figure: Node Group Editor screen*



7. Edit the newly created nodegroup. Then, on the Repository screen, change the repository to RHEL (or your snapshot repository).

Figure: Repository screen



By changing to your new repository, you have effectively added this new nodegroup to your new repository. Continue moving through the rest of the `ngedit` screens, selecting or modifying settings as needed. Upon exit, `ngedit` automatically updates the database.

### Add kit components to nodegroups

Adding kit components to nodes in a nodegroup is very similar to adding additional rpm packages.

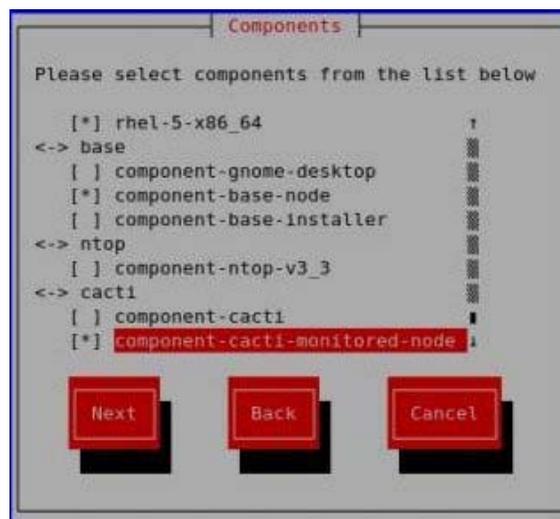
1. Open a terminal and start the `ngedit` tool.
2. Select the `compute-rhel` nodegroup, press F8 or select **Next** and then proceed to the Components screen.

Each Red Hat HPC kit installs an application or a set of applications. The kit also contains components that are meta-rpm packages, designed for installing and configuring applications onto a cluster. By choosing the appropriate components it is easy to configure all nodes in a nodegroup.

For example, the Cacti kit contains two components: `component-cacti` and `component-cacti-monitored-node`. The `component-cacti` installs and configures the Cacti utility, and sets up web pages and connections to the database. This component is normally installed on the cluster

installer node, or any other node (or set of nodes) designated as a management node. The other component in the Cacti kit, *component-cacti-monitored-node*, contains a Cacti agent that runs on the compute nodes in the cluster. Most Red Hat HPC kits come configured with automatic nodegroup association and component selection, making the process of adding kits to nodegroups much easier than manually selecting them in `ngedit`. For example, the Platform Lava kit automatically associates the Lava master with the install nodegroup, and the Lava compute nodes with the compute-rhel nodegroup.

Figure: Components screen



### Add hosts to a nodegroup

Once an installer node is configured and all of the necessary kits are installed on it, you can then add nodes to the cluster. The installer node runs `dhcpd` and is configured to respond to PXE requests on one or more "provision" networks. The quickest way to add hosts is to physically connect them to the same "provision" network, run the "addhost" tool and then PXE-boot the nodes. The following steps detail this procedure:

1. Physically connect the new nodes to the same "provision" network as the installer node.
2. Log on to the installer node as root and run the following command:
 

```
# addhost
```
3. Choose a PCM 1.2b nodegroup.

A nodegroup is a template that defines how a group of nodes will be configured and what software will be installed on the node. When an installer node is created, default nodegroups are built from the operating system supplied during the installation. Different nodegroups include the following:

- install nodeCompute node Compute node, imaged Compute node, diskless
- Unmanaged

If this is your first PCM 1.2b installation, choose the **compute node** nodegroup. Choosing this nodegroup performs a standard

package-based installation onto a new host. Although it is the most reliable method for installation, it is also the slowest method. Once a nodegroup is chosen, select **Next** to proceed.

4. Choose which network to listen on for new hosts. In most cases there is only one network, but for complex clusters there may be more than one network interface. Choose a network and proceed to the next screen.
5. Once addhost is listening on a network, boot the new node. Ensure the new node properly PXE boots.
6. Go to the new node and start the boot process.

If your node is not configured to PXE boot, change the boot order in the BIOS or press F12 while the node is booting to enter PXE boot. If everything is connected properly (that is, if the new node is physically on the same network as the installer node and addhost is listening on that network), you should see the new node download an initrd (initial ram disk) and start a full operating system install. Addhost will report that it has detected the new node and the install is proceeding.

The install on a 100 MB network should take no more than 5 minutes. Once the node installation is complete the machine will reboot and join the cluster.

### Assign IP address to a managed node

Use the `addhost` command to assign an IP address to a managed node. Run the following command to specify an IP address while provisioning a managed node:

```
addhost -n <node_group_name> -x <ip_addr_str>
```

### Manage repositories

PCM 1.2b can support multiple repositories on a single installer node. In a simple PCM 1.2b cluster, there is usually only one repository; however, as the cluster environment becomes more complex there is a need for different operating system repositories or even copies of existing repositories. The PCM 1.2b `repoman` tool creates, deletes, manages kits and snapshots of PCM 1.2b repositories. Repositories are attached.

Repository commands and examples are given below.

---

**Note:** If there is a space in the repository name, enclose the name in quotation marks. For example:

```
repoman -n -r "repository name"
```

---

### Create a repository

```
# repoman -n -r testrepo
```

**Repo: testrepo created.**

**You can now add kits, including OS kits, to the new repository.**

### Delete a repository

```
# repoman -e -r testrepo
```

### Add kits to a repository

```
# repoman -a -r testrepo -k base
```

**Kit: base, version 5.1, architecture noarch, has been added to the repo: testrepo.  
Remember to refresh with -u**

### Remove kits from a repository

```
# repoman -e -r testrepo -k base
```

**Kit: base-5.1-noarch removed from repo: testrepo.  
Remember to refresh with -u**

### Update a repository

```
# repoman -u -r testrepo
```

**Refreshing repo: testrepo. This may take a while...**

### Create a repository snapshot

```
# repoman -s -r rhel5_x86_64
```

## Add nodes to the cluster as diskless and imaged

Imaged node provisioning uses a repository on the installer node and creates a disk image (essentially a pre-installed version of the OS) that is saved in the PCM 1.2b /depot/images directory. When a node is added to the compute-imaged nodegroup, a special initial ram disk and kernel are sent to the node. The disk image is then sent across the network and written to the disk. Once the disk image is installed, the node reboots and starts from the disk instead of the network. An imaged installation is much faster than a standard package-based installation, and is theoretically more customizable than the package-based installation. Images are useful in situations where an application requires a specific version of an operating system with specific packages installed.

Diskless installations are very similar to imaged installations, with one big difference: the image created for a diskless install must fit entirely in the RAM on the compute node. Diskless installs are much smaller than package-based or imaged installs. PCM 1.2b comes configured with a compute-diskless nodegroup. The compute-diskless nodegroup is configured to create a small image that can fit in the RAM of a node. The image can be further customized to reduce the size of the image if needed. Diskless installations are very quick, usually taking less than 30 seconds to install a single node with an operating system. Diskless installations do not require nodes with physical disks attached to them.

Adding nodes to the cluster as diskless or imaged is very simple. The steps for installing are exactly the same as installing a package-based node, the only difference being that the nodegroup compute-diskless or compute-imaged is chosen for the nodes. To add nodes to the cluster as either diskless or imaged, complete the following:

1. Log on as root, and then run the addhost tool:

```
# addhost
```

2. Choose compute-imaged or compute-diskless nodegroup, and then select **Next**.
3. Choose the network interface to listen for new nodes, and then select **Next**.

Addhost waits for the new nodes to boot.

4. PXE boot the new nodes either manually or by logging into the iDRAC (Integrated Dell Remote Access Controller)/BMC (Base Management Controller) on the compute node.
  - If the node is connected to the proper network, it PXE boots and installs either a diskless or disk image. When the installation is complete, the node reboots.
  - If addhost properly detected the new nodes, their MAC addresses appear on the addhost TUI. Exit the TUI when finished.

## Connect to NFS servers

### Home directories in PCM 1.2b

PCM 1.2b automatically configures /home on the installer node as an NFS export. The compute nodes automatically NFS mount /home. This default configuration makes it easy to add new users and their home directories to the cluster. Add them to the installer node and the files in the cluster using the command **cfmsync -f**.

### Append external nodes information to genconfig hosts output

Running `genconfig hosts` on its initial state only generates information on existing nodes (including PCM installer node, compute nodes, and unmanaged nodes). Initially, the optional `/etc/hosts.append` file does not exist on your local machine so you need to create this file if you want to append external nodes information to the genconfig hosts output.

Sample output on initial state of genconfig hosts:

```
.. ...
192.168.0.100  master-node
192.168.0.101  compute-00-00
192.168.0.102  compute-01-00

# Unmanaged nodes
192.168.1.123  unmanaged-00-00
```

To append external nodes information to genconfig hosts output:

1. Create and edit `/etc/hosts.append` file on the PCM installer node.

```
111.111.111.111  external-unmanaged-00-00
```

### 222.222.222.222 external-unmanaged-00-01

Refer to the validation notes specified below when providing line entries in the `/etc/hosts.append` file.

2. Run `genconfig hosts`.

The generated output shows both the details for existing nodes and external nodes, including validation comments on line entries that were ignored during the verification process of importing external nodes information.

Note that if the same hostname is assigned to different IP addresses, the line of information using the same hostname but referencing to a different IP address will be ignored.

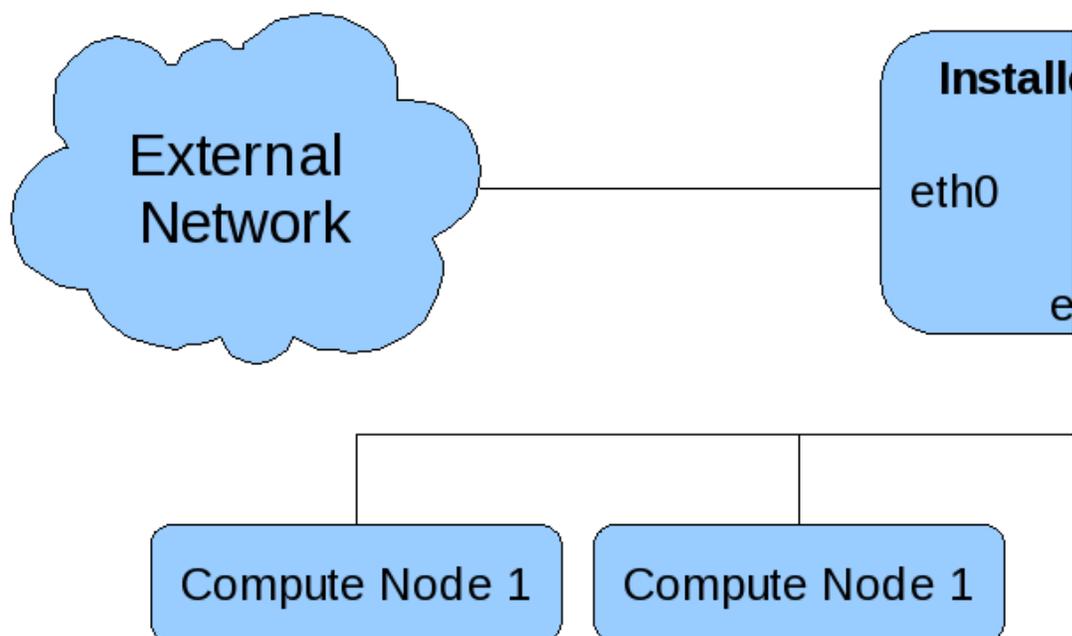
Also note that the whole line of external nodes information will be ignored if any of the following entries are found in certain lines of the `/etc/hosts.append` file during validation:

- IP is not a valid IPv4 address.
- IP is reserved by managed or unmanaged nodes (including PCM master node and other compute nodes) in PCM cluster.
- Host name is not a fully qualified host name (containing invalid characters, of invalid format).
- Host name is reserved by managed or unmanaged nodes (including PCM master node and other compute nodes) in PCM cluster.
- DNS zone is not a fully qualified domain name (containing invalid characters, of invalid format).
- DNS zone is same as the provisioning DNS zone of the current PCM cluster.

## Manage network interfaces

Unlike other HPC cluster toolkits, PCM 1.2b allows for a lot of flexibility in how networks are defined and used in a cluster. In a PCM 1.2b cluster, all possible network configurations are defined using the `netedit` tool. The networks are then associated with the appropriate nodegroups.

*Figure: Sample network configuration*



The key to configuring networks with `netedit` is that each entry in the `netedit` table is a template for a network adaptor attached to a particular network. For example, if the network adaptor is `eth0` attached to a `10.10.0.0` network, then the entry in the `netedit` table represents all nodes that have a `eth0` network attached to a `10.10.0.0` network. There are some exceptions to this rule:

- On the installer nodes, the networks may be provision networks. In these cases, the network definition in the networks table must be unique since installer node provision networks will have a DHCP server bound to them. In the table below there is a entry for `eth0` on the installer node and on the compute nodes connected to the same network interface (`10.10.0.0`) - so even though the network adaptors are the same (`eth0`) and the network is the same (`10.10.0.0`) the network definitions in `netedit` are different simply because the network on the installer node is a provision network and the network on the compute nodes (1 and 2) are public.
- When the network adaptors are the same, or when the networks are the same but the starting IP address for numbering the nodes is different or the gateway IP address for the nodes is different. In either case, defining an entry in `netedit` for each adaptor attached to a particular network is always the best way to start configuring networks in PCM 1.2b.
- During installation, some networks are automatically configured by PCM 1.2b. In particular, the networks attached to the installer node must always be configured during installation since it is difficult to add new networks to the installer node after installation.

The following table lists the networks that must be configured using `netedit`:

Device	Network	Subnet	Description	Type
<code>eth0</code>	<code>10.10.0.0</code>	<code>255.0.0.0</code>	Eth0 network on installer node	provision
<code>eth0</code>	<code>20.20.0.0</code>	<code>255.0.0.0</code>	Eth0 network on node 5	public
<code>eth0</code>	<code>10.10.0.0</code>	<code>255.0.0.0</code>	Eth0 network on node 1,2	public
<code>eth1</code>	<code>20.20.0.0</code>	<code>255.0.0.0</code>	Eth1 on installer node	provision
<code>eth1</code>	<code>20.20.0.0</code>	<code>255.0.0.0</code>	Eth1 on node 6	public

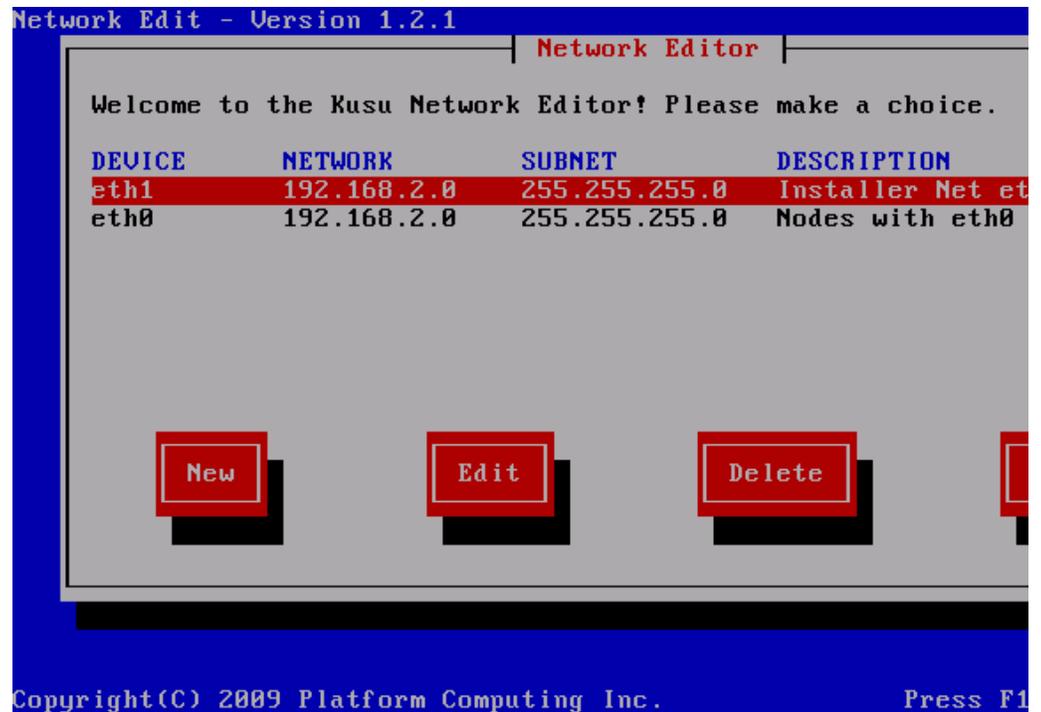
eth1	10.10.0.0	255.0.0.0	Eth 1 on node 3, 4	public
lb0	30.30.0.0	255.0.0.0	Infiniband on nodes 3,4,5,6	provision

## Configure new network templates

1. Log on as root.
2. From a terminal prompt, run the `netedit` tool:

```
# netedit
```

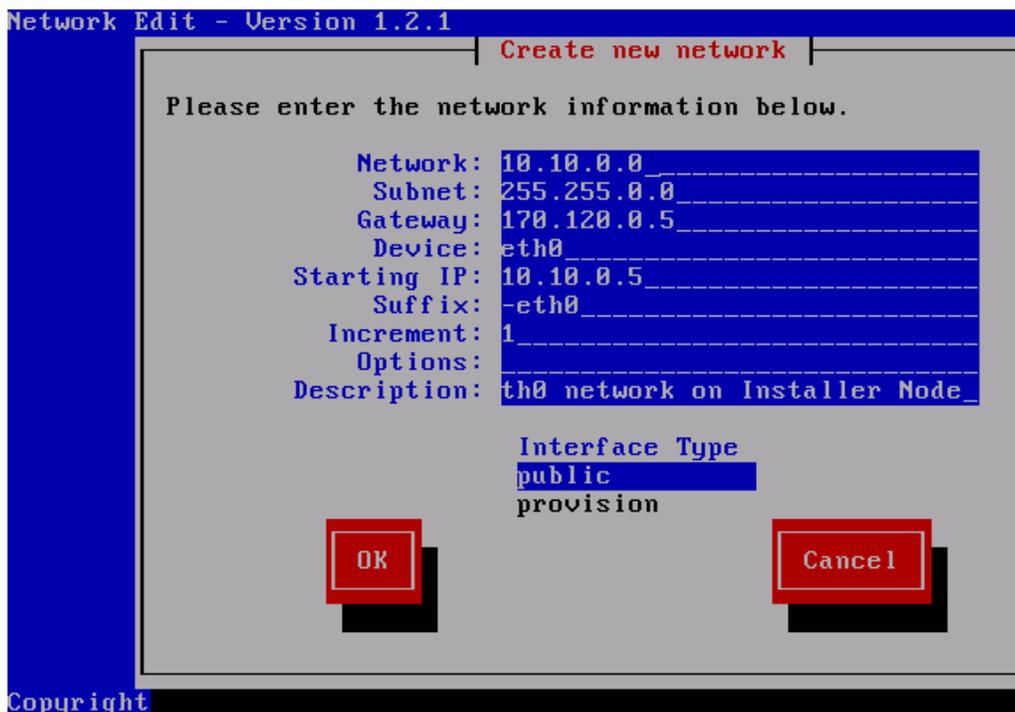
*Figure: Network Editor screen*



3. Create a new network definition by selecting **New**.
4. Fill in the fields to create a new network.

Note that some networks do not need a Gateway IP address. For example: In the network diagram above, the Infiniband network does not require a route to the public internet network, thus no Gateway IP address is required for the Infiniband network configuration.

*Figure: Edit existing network screen*



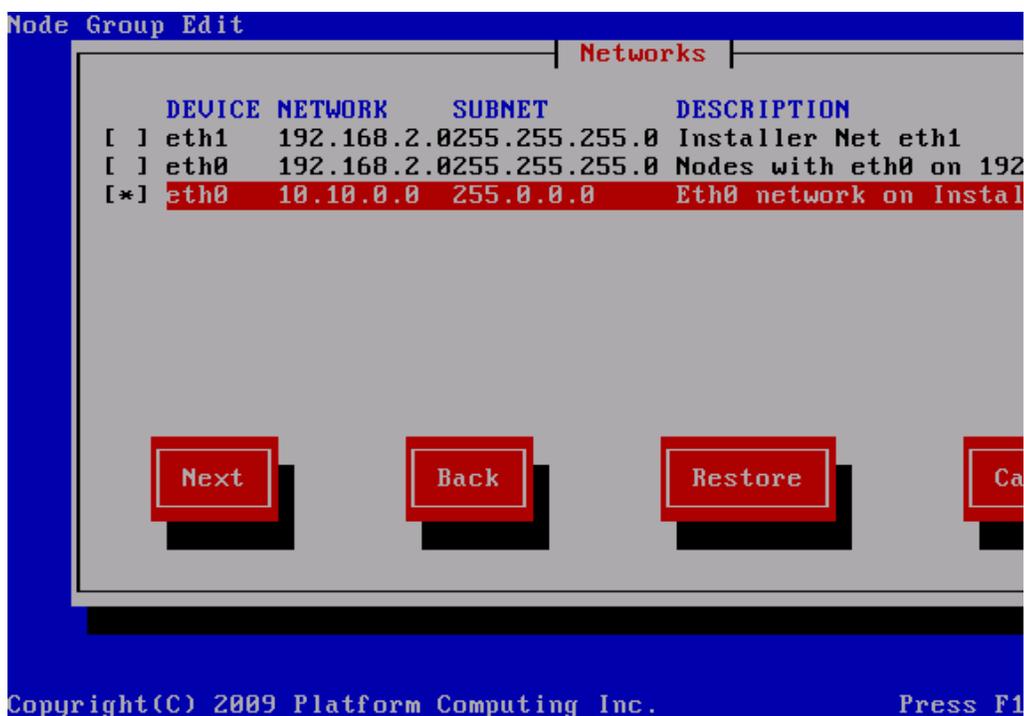
- Once networks are defined in the database with the `netedit` tool they can be attached to nodegroups using the `ngedit` tool. For example, once the 10.10.0.0 network is defined, log on as root and run the following tool:

```
# ngedit
```

- Edit any network of your choice and select **Next** until the Networks screen is displayed.

All of the available networks defined in `netedit` should display on the Networks screen.

Figure: Networks screen



- Assign a network to a nodegroup by selecting the network using the

space bar.

This puts an asterisk beside the network, and assigns all nodes in the chosen nodegroup with this network.

By combining `netedit` with the nodegroup editor `ngedit`, complex network configurations such as the one described above can be created and managed with PCM 1.2b.

---

**Important:** If you created multiple network interfaces on a node, PCM 1.2b automatically creates short names of hosts for these interfaces of the node:

- \* **First provision interface** (for master node)
  - \* **First bootable provision interface** (for compute node)
- 

## Synchronize files in the cluster

HPC clusters are built from many individual compute nodes. All of these nodes must have copies of common files such as `/etc/passwd`, `/etc/shadow`, `/etc/group`, and others. PCM 1.2b contains a file synchronization service called `cfm` (Configuration File Manager). The `cfm` service runs on each compute node in the cluster. When new files are available on the installer node, a message gets sent to all of the nodes notifying them that files are available. Each compute node connects to the installer node and copies the new files using the `httpd` daemon on the installer node. All of the files to be synchronized by `cfm` are located in the directory tree `/etc/cfm/<nodegroup>`. The `cfm` service organizes file synchronization trees by nodegroup. A directory exists for each nodegroup under `/etc/cfm`. Below the nodegroup name is a tree that replicates the file structure of the machines in the nodegroup.

*Figure: File structure example*

```

root@RH-HPCinstaller:/etc/cfm/compute-rhel/etc
File Edit View Terminal Tabs Help
[root@RH-HPCinstaller ~]# cd /etc/cfm
[root@RH-HPCinstaller cfm]# ls
compute-diskless  compute-rhel  group.new  passwd.getent  shadow.getent
compute-imaged   group.getent  installer  passwd.new     shadow.new
[root@RH-HPCinstaller cfm]# cd compute-rhel/
[root@RH-HPCinstaller compute-rhel]# ls
etc
[root@RH-HPCinstaller compute-rhel]# cd etc
[root@RH-HPCinstaller etc]# ls
fstab.append  group  hosts  hosts.equiv  passwd  shadow  ssh
[root@RH-HPCinstaller etc]# cat fstab.append
# Appended by CFM
# Entries below this come from the CFM's fstab.append
[root@RH-HPCinstaller etc]#

```

In the figure above, the `/etc/cfm` directory contains several nodegroup directories such as `compute-diskless` and `compute-rhel`. In each of those directories is a directory tree where the `/etc/cfm/<nodegroup>` directory

represents the root of the tree. The `/etc/cfm/compute-rhel/etc` directory contains several files or symbolic links to system files. These system files synchronize across all of the nodes in the nodegroup automatically by `cfm`. Creating symbolic links for the files in `cfm` allows the compute nodes to automatically synchronize with system files on the installer node.

To add files to `cfm`, create the new file in the appropriate directory. Make sure to create all of the directories and subdirectories for the file before placing the file in the correct location. Existing files can also have a `<filename>.append` file. The contents of a `<filename>.append` file is automatically appended to the existing `<filename>` file on all nodes in the nodegroup.

To notify all of the nodes in all nodegroups or nodes in a single nodegroup, run this command:

```
# cfmsync -f -n compute-rhel
```

This synchronizes all files in the compute-rhel nodegroup.

To synchronize all files in all nodegroups, run this command:

```
# cfmsync -f
```

For more information on `cfmsync`, view the man pages.

---

**Note:** Place a file in `/etc/cfm/<nodegroup>/<path to file>` to replace the existing file on all nodes in the nodegroup or to create the file if it does not exist. Create a file with `/etc/cfm/<nodegroup>/<path to file>.append` to append the contents of the file on all nodes in the nodegroup.

---

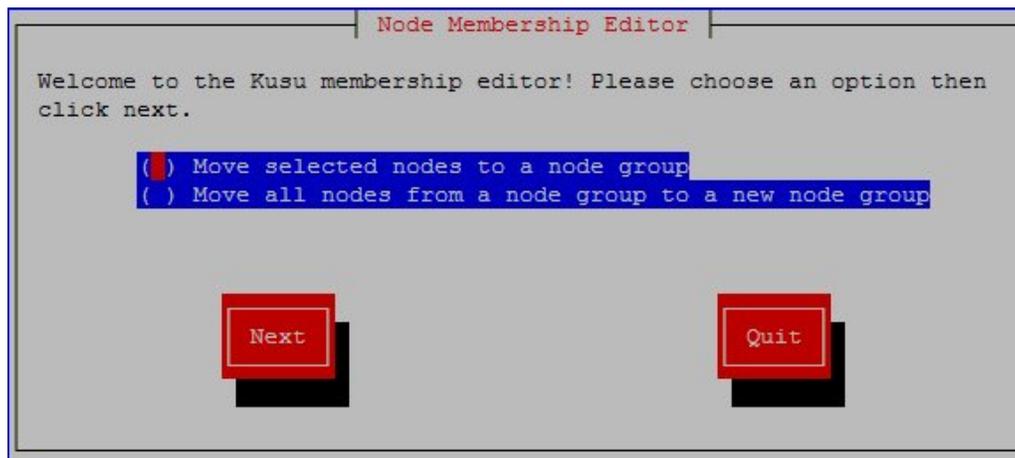
## Move hosts between nodegroups

During installation, PCM 1.2b assigns all nodes in the cluster to a nodegroup; however, HPC clusters rarely stay the same and change over time: new nodes are added, old nodes are removed, the applications, packages and even the operating system on the nodes will change. PCM 1.2b is designed with this flexibility in mind. It is very easy to move single nodes or a group of nodes from one nodegroup to another. When a node is moved, the "personality" of the node changes and the node is manually or automatically provisioned according to the configuration defined by the new nodegroup.

1. Run the `nghosts` command to move one nodegroup to another.

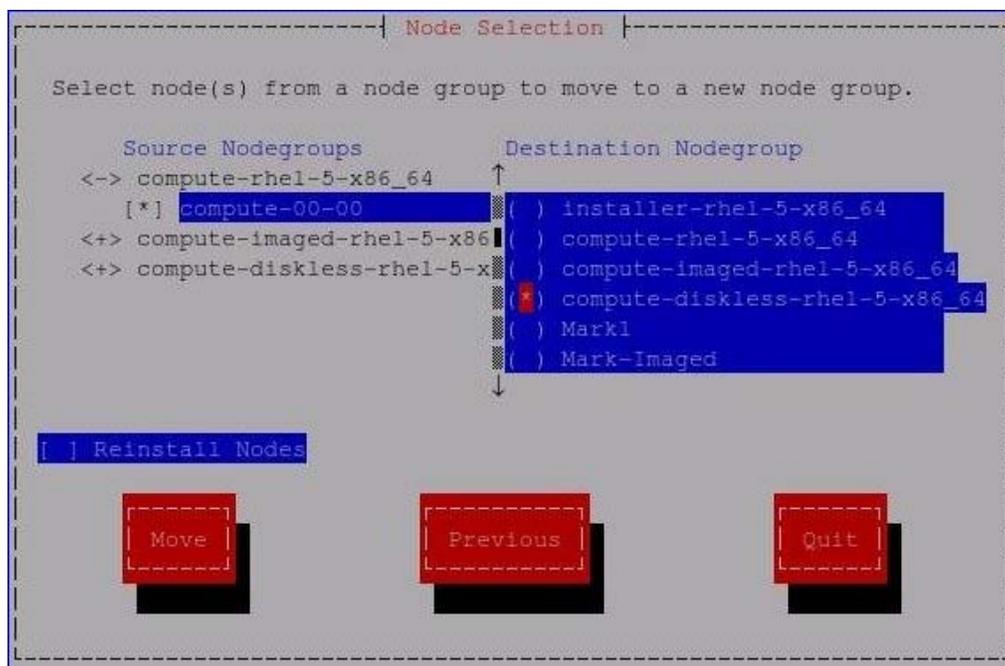
```
# nghosts
```

*Figure: Node Membership Editor screen*



2. Choose either **Move selected nodes to a nodegroup** or **Move all nodes from a nodegroup to a new nodegroup**.
3. Choose the nodes you wish to move and then choose a destination nodegroup.
4. Select **Move** to transfer the nodes to the new nodegroup.

Figure: Node Selection screen



5. Once the nodes have been moved, select **Quit** to exit.

The nodes are now assigned to a new nodegroup. The nodes, however, are not re-provisioned until the node (or nodes) is rebooted.

6. Shutdown the affected nodes and then restart them, ensuring that they PXE boot (by default PCM 1.2b expects all nodes to always PXE boot).

If nodes can be moved back to the original nodegroup using nghosts, they are provisioned back to their original state. Note: Each time a node is provisioned, it returns to its original installed state - thus any configuration files or applications on the nodes are also removed and re-installed. If you need to retain the state of the nodes, consider saving all shared configuration files on a

separate network attached storage or fileserver.

## Create a new repository

PCM 1.2b comes pre-configured with at least one repository. This repository is used to create the installer node and to create compute nodes or any other type of node needed for the cluster. There are two obvious cases when more than one repository is needed in the cluster:

- Updating a repository from a snapshot (see [Update a repository](#))
- Mixing two different operating systems in one cluster

Clusters tend to grow over time. As the needs of users increase, an administrator adds more nodes to the cluster. It is not uncommon for clusters to start on one version of an operating system and over time need new nodes with a new version of the operating system. PCM 1.2b can provision different operating system types and versions from a single installer node. This provides the cluster administrator with flexibility when designing a cluster, and makes it easy to migrate the cluster to a new operating system when it arrives.

To add a new operating system to a PCM 1.2b installer node, perform the following:

1. Create a new repository using the `repoman` command:

```
# repoman -n -r "Repo for rhel5.4-x86_64"
```

2. Add the appropriate kits to the repository using the `kitops` command.

The base kit is always needed for proper operation of PCM 1.2b. The operating system is also a kit and must be added to the installer node before it can be added to the repository.

1. Add the OS kit PCM 1.2b:

```
# kitops -a -m "/media/rhel5.4-x86_64" --kit=rhel
```

2. Add the base kit and the OS kit to the repository:

```
# repoman -a --kit=rhel -r "Repo for rhel5.4-x86_64"
```

```
# repoman -a --kit=base -r "Repo for rhel5.4-x86_64"
```

```
# repoman -u -r "Repo for rhel5.4-x86_64"
```

Note: If you have more than one kit called "sles" or "base", you will need to specify the kit version or kit architecture.

3. Update the repository (see [Update a repository](#)) .
4. Create new nodegroups (see [Create a new nodegroup](#)) .
5. Associate the new repository with one or more nodegroups (see [Associate a repository with nodegroups](#)).

Now the repository contains the necessary components the `ngedit` tool needs to create nodegroups and assign them to the new repository.

## Update a repository

PCM 1.2b repository management allows administrators to update operating

system repositories using the native package management tools. On Red Hat based systems, the tool is "yum". PCM 1.2b can connect to Red Hat Network and download updates to the repository (as long as you have a valid entitlement) or connect to any yum repository and download patches and updates.

Maintaining and updating repositories is critical for system administrators, particularly within HPC clusters. When security patches are issued, the natural response is to install the patches as quickly as possible on the cluster. In many cases, however, there are unintended side-effects caused by updating a cluster. Typically when an update tool like yum is used, the administrator downloads all updates and applies them to the operating system. Some of these updates may cause problems on a cluster while others fix security issues. What normally happens is the administrator decides which updates are needed and then manually installs them (or writes a script to do this) into the cluster. A better solution is to employ an "update and test" mechanism (that is, download the updates, test them out, and only if everything is okay, update the entire cluster). PCM 1.2b provides this mechanism.

If you are updating an existing software kit, you must first update the configuration file that gets read when you initiate the patch procedure. Edit the file `/opt/kusu/etc/updates.conf` and specify your Red Hat Network user name and password.

To safely update a repository, perform the following:

1. Create a snapshot of the existing repository that will be updated using the `repoman` command.

```
# repoman -s -r "repository name"
```

A snapshot is created for the repository and is added to the list of repositories.

The `repoman` command can perform the following actions on a repository:

- Create a repository
- Add kits to a repository
- Update/refresh a repository
- Snapshot a repository
- Delete a repository

You can list repositories, including the new snapshot repository, using the `repoman -l` command. The new repository is named similarly to the original repository on which it is based, with the addition of a snapshot date and time stamp. For example, if the original repository is named "myrepo," the new repository is named "myrepo(snapshot Tue Mar 18 11:43:06 2008)."

2. Update the snapshot with the command `repopatch`, and give it a new name.

```
# repopatch -r "new repository name"
```

---

**Note:** Updating the repository may take some time, as updates from various OS vendors (such as a new kernel) may need downloading. In some cases, you are required to interact with the updates (for example, you may need to click OK to continue or to approve a download); if you attempt an unsupervised overnight update to the repository, your

database connection may time out if interaction requests go unanswered.

---

The next time you run `repoman -i`, look for the new kit with an updated OS name in the list. You can also run `kitops -i` to see the new kit name and version number.

6. Test the updated repository snapshot on one or two machines.
  1. Run the command `ngedit`.
  2. From the original repository (on which you based the new repository), select a nodegroup and make a copy of it.
  3. Edit the copied nodegroup, making changes on the Repository page to include the new repository name to which you want to associate it.
  4. Review each configuration page and make any other changes to the new nodegroup, as required. Select **Accept** when your edits are completed.

The new nodegroup is now mapped to the new repository,

5. Exit the Node Group Editor.
  6. Run `nghosts`.
  7. Find one or two machines to use for testing purposes, and then select those machines to move to the newly created nodegroup.
  8. Choose to reinstall the entire machine.
  9. Log on to the testing machines and ensure everything is ok.
7. Merge the updates with one of these two methods:
    - Move all of the machines into the new nodegroup.

OR

    - Run `repoman -a -k new_kit_name -r repository_name` to associate the updates with the original repository.

You must then run `repoman -u -r repository_name` to update the repository to which you've added the new kit.

8. Update the entire cluster.

You can update the cluster by reinstalling nodes (cleanest method), or by updating existing node packages in the nodegroup (a faster method, but there is a risk of failed package upgrades). Both options are described below:

- Reinstall nodes: Run `boothost -n "nodegroup name" -r`

This performs a full reinstallation on all nodes in the nodegroup.
- Update node packages: Run `cfmsync -u -n "nodegroup name"`

All nodes in the nodegroup will check their repositories and make

any required updates to packages.

## Associate kit components to nodegroups

When you add a new kit to a nodegroup repository, set up a new nodegroup, or offload some functions from one machine to another, you will need to associate kit components to the nodegroup.

For information about individual components, see the corresponding kit documentation.

1. Run `ngedit`.
2. (Optional) If desired, create a nodegroup to reserve certain machines for a special purpose (for example, to run all web services).
  - Copy an existing nodegroup upon which to base the new one.
  - Configure the new nodegroup to make it distinctive and suitable for your purposes. (For example, rename it, provide a description, indicate the NN format and name, etc.)
3. Choose the nodegroup to which you want to associate kit components.
4. Navigate to the Components screen.

Here you can control the association of components with the parent kit. What you see on the Components page depends on the repository to which the nodegroup is associated.

5. Choose those components you want installed on the machines in the nodegroup.
6. Complete any other configuration changes within `ngedit`.
7. If desired, use `nghost` to move a node to a nodegroup, or `addhost` to add a new host to a nodegroup.

## Set compute node in BIOS

In BIOS, set the boot sequence of the compute node to **always boot from network**.

## Update the installer node and compute node

To install a security update, or other required updates across your PCM 1.2b cluster, you will need to update both the installer node and the compute node.

1. Use `repopatch` to update the installer node.

This command uses an existing repository to update files on the installer node. Ensure that you have updated this repository with required changes prior to running this command. See [Update a repository](#) for details.

---

**Note:** Depending on your repository configuration, the install script might visit certain OS and 3rd party application sites to download kernel or driver updates. Updates might take some time as a result. You will want

to monitor the update process and consent to or deny various update requests.

**Warning:** If you update a kernel, some drivers that were previously provided by 3rd parties may no longer work. Take precautions in deciding to update a kernel.

---

## 2. Update all compute nodes:

1. Update the repository (see [Update a repository](#)).

2. Perform a full reinstallation on all nodes in the nodegroup:

```
boothost -n "nodegroup name" -r
```

PCM 1.2b checks the database state, and then reboots all nodes and reinstalls required updates and drivers.

## PCM 1.2b NIS/LDAP authentication

1. Configure the installer node to authenticate using the corporate NIS or LDAP server.

To do this, run the RH tool *system\_config\_authentication* to configure the authentication services that the installer node uses.

2. Run `cfmsync -f`.

This command signals the nodes in the cluster to look for any configured NIS or LDAP servers you might have, and to update their configuration files and or packages/components accordingly.

3. Navigate to `/etc/cfm`, and look for the these auto-generated files: `shadow.getent`, `passwd.getent`, and `group.getent`.

These updated files now contain password and authentication information needed for the rest of the cluster.

4. Tell the nodegroup which authentication files to use, along with password information:

1. Navigate to `/etc/cfm/<nodegroup>/etc`.

Here you will find existing symbolic links that must be updated.

2. Change the symbolic link to point to the new file containing updated authentication information (the `*.getent` files).

## Backup and restore PostgreSQL database in newer clusters

PCM 1.2b uses a PostgreSQL (postgres) database to store cluster configuration information. This section provides an overview of the database tables, review the backup procedure, as well as how to restore the database.

The database has the following tables:

- appglobals
- components
- driverpacks

- o kits
- o modules
- o networks
- o ng\_has\_comp
- o ng\_has\_net
- o nics
- o nodegroups
- o nodes
- o os
- o packages
- o partitions
- o repos
- o repos\_have\_kits
- o scripts

Prior to performing server maintenance or upgrading your application, you will want to safely back up the database, and then later restore it.

### Back up the database

Use the following commands to back up the database.

```
# export PGPASSWORD=`cat /opt/kusu/etc/db.passwd`  
# pg_dump -U apache kusudb > db.backup
```

### Restore the database

Use the following commands to restore the database.

1. Authenticate access permission of kusudb for user postgres:

a) Edit the file:

```
/var/lib/pgsql/data/pg_hba.conf
```

and add the following line:

```
local all postgres trust
```

b) Restart postgres:

```
service postgresql restart
```

2. Drop database kusudb

```
dropdb -U postgres kusudb
```

3. Create database kusudb:

```
createdb -U postgres kusudb
```

4. Load backup file:

```
psql -U postgres kusudb < db.backup
```

5. Remove user postgres access permission to kusudb:

a) Edit the file:

```
/var/lib/pgsql/data/pg_hba.conf
```

and remove the line:

```
local all postgres trust
```

b) Restart postgres:

```
service postgres restart
```

### Verify the database restoration

To verify that the database has been successfully restored, run the following command:

- `genconfig nodes`: If the database restore was successful, this command returns a list of machines.

## Backup and restore MySQL database in older clusters

Older versions of PCM, use a MySQL database to store cluster configuration information. This section provides an overview of the database tables, review the backup procedure, as well as how to restore the database.

The database has the following tables:

- `app_globals`
- `components`
- `driverpacks`
- `kits`
- `modules`
- `networks`
- `ng_has_comp`
- `ng_has_net`
- `nics`
- `nodegroups`
- `nodes`
- `packages`
- `partitions`
- `repos`
- `repos_have_kits`
- `scripts`

Prior to performing server maintenance or upgrading your application, you will want to safely back up the database, and then later restore it.

### Back up the database

1. From a command prompt, run `genconfig debug`.

Once run, you will see the populated database file.

2. From the directory where you ran `genconfig debug`, specify a name for the backup file. For example:

```
# genconfig debug > db.backup
```

### Restore the database

1. To delete the existing database, run this command:

```
# mysqladmin drop kusudb
```

---

**Caution!** This command deletes the entire database. Ensure you have created a backup file first.

---

2. Run this command to create a new database:

```
# mysqladmin create kusudb
```

The newly created database is empty at this point.

3. Navigate to the location of your backup file (in this example, named `db.backup`), and then run this command:

```
# mysqladmin kusudb < db.backup
```

4. Navigate to `/opt/kusu/sql`, and look for a file called `kusu_dbperms.sql`.

This file sets the default password for Apache users.

5. Edit `kusu_dbperm.sql` and change the default password "Letmein" to your configured password.

6. Copy this file to the restored database with this command:

```
# mysqladmin kusudb < ./kusu_dbperms.sql
```

7. Change directories to `/opt/kusu/etc`, and then look for a file called `db.passwd`.

8. Change the password string in `db.passwd` to match the one previously entered in `kusu_dbperms.sql`.

### Verify the database restoration

To verify that the database has been successfully restored, run the following command:

- o **genconfig nodes:** If the database restore was successful, this command returns a list of machines.

## Configure a dedicated logging server in PCM cluster

The PCM installer node, by default, is set as the logging server. To know the default PCM logging server's address, run command:

```
# sqlrunner -q "select kvalue from appglobals where kname='SYSLOG_SERVER';"
```

You can set a dedicated logging server, in the PCM cluster, that supports forwarding of all logging messages from the PCM installer node and compute nodes but not to other dedicated nodes. Having a dedicated logging server, that collects all logged files, releases some pressure from the PCM installer node. This also allows you identify which messages come from which exact nodes and easily analyze problems.

To set a dedicated logging server in PCM cluster:

1. Specify a dedicated logging server address:

```
# sqlrunner -q "update appglobals set kvalue='IP address' where
kname='SYSLOG_SERVER';"
```

For example:

```
# sqlrunner -q "update appglobals set kvalue='172.20.7.35' where
kname='SYSLOG_SERVER';"
```

---

**Tip:** You can specify a logging server using an IP address or hostname.

---

2. Effect configuration changes by running these commands:

```
# kusrvc /etc/rc.kusu.d/S03KusuRsyslogMaster.rc.py
```

```
# cfmsync -f
```

or

```
# kusrvc /etc/rc.kusu.d/S03KusuRsyslogMaster.rc.py
```

```
# addhost -u
```

## Configure add-on NIC as provisioning network

This describes the steps for configuring a different provisioning network on the installer node, other than the default eth0. You can configure an add-on NIC (for example, 10GE NIC) as a provisioning network on the installer node.

**Case 1: During installation of Head node**

During installation, do the following on the "Network" screen:

1. Select "eth0" then uncheck "configure this device" option before clicking on "OK".
2. Select new network device "eth4" then click "configure".
  - Check the option "Configure this device".
  - Specify IP address and NetMask (for example, 172.20.0.1 and 255.255.0.0).
  - Update network name as "Cluster".
  - Select Network type as "provision".
  - Select "Activate on boot".
3. Click "OK" and continue with the installation.
4. Verify "netedit" output after the installation.
5. Check "ngedit" (Networks field) for the installer and compute nodegroups.
  - In the example, "eth4" is selected by default for both the nodegroups.
  - Change this selection to "eth0" for the compute nodegroup as NIC1 will be used for compute nodes connectivity to the network
6. Install compute nodes using `addhost`.

**Case 2: After installation of Head node**

Assume that the Head node installation is already completed with eth0 (provisioning) and eth1 (public) networks configured by default.

---

**Important:** If there are compute nodes which are already installed, remove these nodes by running "addhost -e" before configuring. Once the configuration is complete, all the nodes can be added back to the cluster using "addhost", selecting eth4 as the provisioning network. See examples below.

---

1. Unselect eth0 from the installer and compute nodegroups using `ngedit`.
2. Delete "eth0" using `netedit`.
3. Use `kusu-net-tool` to add a new provisioning network. The following command updates the configuration and refresh the repository.

```
kusu-net-tool addinstnic eth4 --netmask=255.255.0.0  
--ip-address=172.20.0.1 --start-ip=172.20.0.1 --provision  
--gateway=172.20.0.1 --desc="network2" --macaddr= --update-system
```

4. Verify "netedit" output.
5. Set "ONBOOT=no" in `/etc/sysconfig/network-scripts/ifcfg-eth0`.
6. Set "ONBOOT=yes" in `/etc/sysconfig/network-scripts/ifcfg-eth4`.
7. Reboot the head node.
8. Check "ngedit" (Networks field) for the installer and compute nodegroup.
  - In the example, "eth4" is selected by default for the Installer nodegroup provisioning network.
  - For the compute nodegroup, none of the networks will be selected by default.
  - Select "eth0" for the compute nodegroup, as NIC1 will be used for compute nodes connectivity to the network.
9. Install compute nodes using `addhost`.

[ [Top](#) ]

---

## Get technical support

### Contact Platform Computing

Contact Platform Computing or your PCM 1.2b vendor for technical support. Use one of the following to contact Platform technical support:

#### Web Portal eSupport

You can take advantage of our web-based self-support available 24 hours per day, 7 days a week ("24x7") by visiting <http://my.platform.com>.

The Platform eSupport and Support Knowledgebase site enables you to search for solutions, submit your support request, update your request, enquire about your request, as well as download product manuals, binaries and patches.

## Email Support

[support@platform.com](mailto:support@platform.com)

## Telephone Support

View contact information available at <http://www.platform.com/services/support>.

When contacting Platform, please include the full name of your company.

See the Platform web site at <http://www.platform.com/Company/Contact.Us.htm> for other contact information.

[ [Top](#) ]

---

## Copyright and trademarks

© 1994-2010 Platform Computing Corporation. All Rights Reserved.

Although the information in this document has been reviewed, Platform Computing Corporation ("Platform") does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

### Trademarks

LSF®, MPI®, and LSF HPC® are trademarks or registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

Platform Cluster Manager™, ACCELERATING INTELLIGENCE™, PLATFORM COMPUTING™, and the PLATFORM™ and PCM logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat® is a registered trademark of Red Hat, Inc.

Intel® is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Cisco® Systems is a registered trademark of Cisco Systems, Inc. and/or its affiliates in the U.S. and in other countries

Nagios® and the Nagios Logo are servicemarks, trademarks, registered trademarks owned by or licensed to Ethan Galstad.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

[ [Top](#) ]

---

Date Modified: April 2010

Platform Computing: [www.platform.com](http://www.platform.com)

Platform Support: [support@platform.com](mailto:support@platform.com)

Platform Information Development: [doc@platform.com](mailto:doc@platform.com)

1994-2010 Platform Computing Corporation. All rights reserved.