If the administrator certificate and key are replaced, then any other certificates signed by the original administrator certificate must be generated again using the replacement, because otherwise they will no longer function.

Certificate generation in general, including the generation and use of non-administrator certificates, is described in greater detail section 6.4.

Replacing A Temporary Or Evaluation License

In the preceding section, if a license is replaced, then regular user certificates need to be generated again. Similarly, if a temporary or evaluation license is replaced, regular user certificates need to be generated again. This is because the old user certificates are signed by a key that is no longer valid. The generation of non-administrator certificates and how they function is described in section 6.4.

2.3.4 Profiles

Certificates that authenticate to CMDaemon contain a *profile*.

A profile determines which cluster management operations the certificate holder may perform. The administrator certificate is created with the `admin` profile, which is a built-in profile that allows all cluster management operations to be performed. In this sense it is similar to the `root` account on unix systems. Other certificates may be created with different profiles giving certificate owners access to a pre-defined subset of the cluster management functionality (section 6.4).

2.4 Cluster Management GUI

This section introduces the basics of the cluster management GUI (`cmgui`). This is the graphical interface to cluster management in Bright Cluster Manager. It can be run from the head node or on a login node of the cluster using X11-forwarding:

Example

```
user@desktop:~> ssh -X root@mycluster cmgui
```

However, typically it is installed and run on the administrator's desktop computer. This saves user-discernable lag time if the user is hundreds of kilometers away from the head node.

2.4.1 Installing Cluster Management GUI On The Desktop

Installation packages are available for Linux, for Windows XP/Vista/Windows 7/Windows 8, and for Mac OS.

To install `cmgui` on a desktop computer a Firefox browser, version 10 or greater, must first be installed onto the operating system. The `cmgui` installation package corresponding to the CMDaemon version must then be installed to the desktop. The package can be downloaded from from the internet or from the head node.

Downloading From The Internet

The latest `cmgui` packages are publicly available at `http://support.brightcomputing.com/cmgui-download`. Mac OS, MS Windows, or Linux packages are available. The administrator should use these with up-to-date Bright Cluster Manager releases.

Downloading From The Head Node

If no internet access is available, then the packages can be picked up from the head node as follows:

For a default repository configuration, doing a `yum update`, or `zypper up` for SLES-based distributions, ensures version compatibility on the head node. The appropriate installation package for the desktop can then be copied over from the head node of any Bright Cluster Manager cluster under the directory:

```
/cm/shared/apps/cmgui/dist/
```

Installing cmgui On Mac OS, MS Windows, And Linux

The installation package can be placed in a convenient and appropriately accessible location on the desktop.

On the Mac OS desktop, cmgui is installed from the Mac OS install.cmgui.macosx.7.2.r7498.pkg file by clicking on it and following the installation procedure. The cmgui front end is then run by clicking on the cmgui icon.

On the MS Windows desktop, cmgui is installed from the install.cmgui.7.2.r7498.exe installer file by running it and following the installation procedure.

If there is an antivirus application preventing its installation, then the file and folders into which cmgui is installed should be excluded from the antivirus checks as suggested by the antivirus software vendor.

After installation, cmgui is started through the Start menu or through the desktop shortcut.

For Linux, the installation of a 64-bit Firefox browser on the desktop, and not a 32-bit version, is mandatory, before cmgui is installed. The version of Firefox on the desktop can be checked and replaced if necessary. The check can be done similarly to the following (some output elided):

Example

```
root@work:~# file /usr/bin/firefox
/usr/bin/firefox:  ELF 32-bit LSB executable, Intel 80386,...
root@work:~# apt-get install firefox:amd64
...
root@work:~# exit
```

For the Linux desktop, cmgui can then be installed and run by:

- copying over the .tar.bz2 file
- untarring the .tar.bz2 file
- running cmgui from inside the directory that the untar process created.

The install and run procedure may look similar to (some output elided):

Example

```
me@work:~$ scp root@bright72:/cm/shared/apps/cmgui/dist/cmgui.7.2.r7498.tar.bz2 .
...
me@work:~$ tar -xjf cmgui.7.2.r7498.tar.bz2
...
me@work:~/cmgui-7.2-r7498$ ./cmgui
```

Bright Cluster Manager `cmgui` Welcome Screen

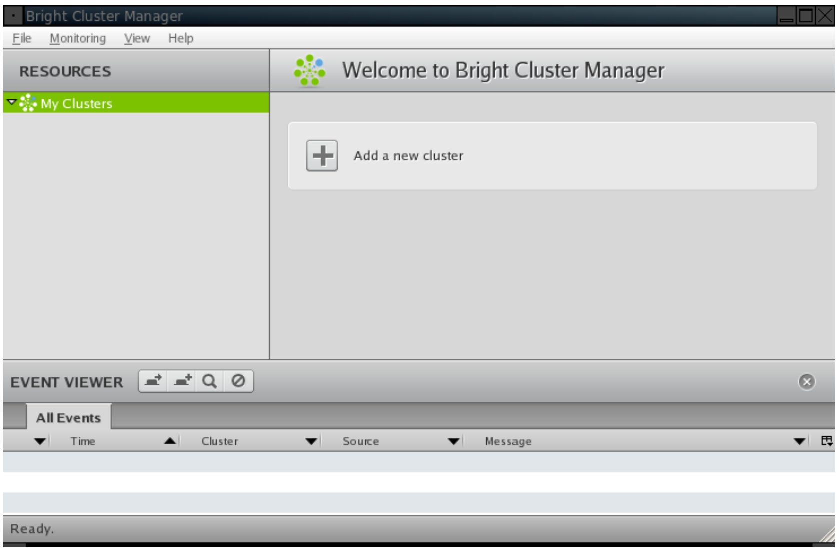
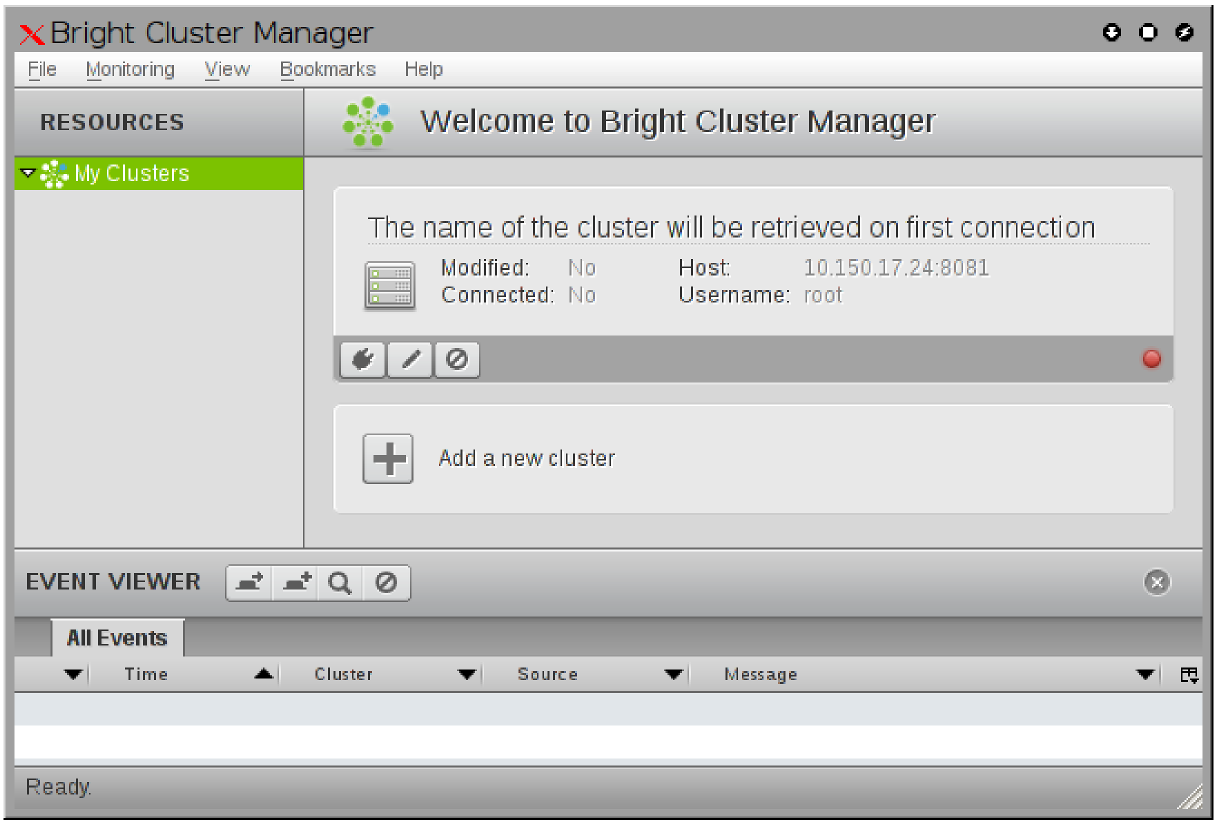


Figure 2.1: Cluster Manager GUI welcome screen

When `cmgui` is started for the first time, the welcome screen (figure 2.1) is displayed.

To configure `cmgui` for connections to a new Bright Cluster Manager cluster, the cluster is added to `cmgui` by clicking the ⊞ button in the welcome screen. More clusters can be added within `cmgui` as needed.

After a cluster is added, the screen displays the connection parameters for the cluster (figure 2.2).


Bright Cluster Manager
File Monitoring View Bookmarks Help
RESOURCES
My Clusters
Welcome to Bright Cluster Manager
Add a new cluster
EVENT VIEWER
All Events
Time Cluster Source Message
Ready.

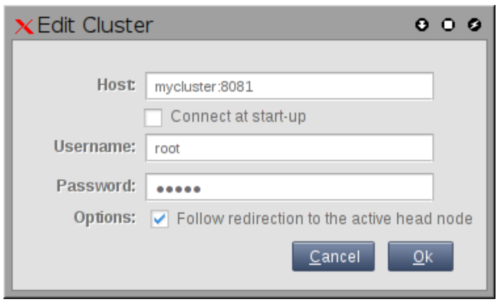
The name of the cluster will be retrieved on first connection

Modified:	No	Host:	10.150.17.24:8081
Connected:	No	Username:	root

Figure 2.2: Connecting to a cluster

Bright Cluster Manager `cmgui` Connection Parameters Dialog Window

Figure 2.3 shows the dialog window in which the connection parameters can be entered.



Edit Cluster

Host: mycluster:8081

☐ Connect at start-up

Username: root

Password: •••••

Options: ☑ Follow redirection to the active head node

Figure 2.3: Editing The Cluster Connection Parameters

The dialog window in figure 2.3 has several options:

- The `Host` field can be a name or an IP address. If the port on the host is not specified, then port 8081 is added automatically.
- The "`Connect at start-up`" checkbox option offers convenience by attempting to connect to the cluster right away when starting up `cmgui`, without waiting for the administrator to click on the connect button of figure 2.2.
- The "`Username`" field accepts the name of the user that `cmgui` is to run as. So, `cmgui` can be run by a non-root user as user `root` as well as run the other way round.
- The `Password` field affects connection behavior when starting up `cmgui`, or when clicking on the connect button of figure 2.2. The behavior is generally obvious, and depends on whether a password has been saved in the `Password` field:

  1. With a correct password set, and all other fields in the dialog set correctly, a connection to the cluster is established when starting up `cmgui` if the "`Connect at start-up`" state has been set. The cluster overview screen (figure 2.4) for the cluster is then displayed.
  2. With a blank password, but all other fields in the dialog set correctly, and if continuing in a current `cmgui` session where a previous successful connection to the cluster has been disconnected, then clicking on the connect button of figure 2.2 establishes a successful connection to the cluster, and the cluster overview screen (figure 2.4) for the cluster is then displayed.
  3. With a blank password, but all other fields in the dialog set correctly, and if the cluster is being connected to for the first time during a new `cmgui` session, then a password prompt dialog is displayed if:

     – starting `cmgui` and if the "`Connect at start-up`" state has been set
       or

     – if `cmgui` is already running and the connect button in the screen shown in figure 2.2 is clicked.

     The cluster overview screen (figure 2.4) for the cluster in this case (case number 3) is only displayed after a successful authentication.

The administrator should be aware that storing the password is unsafe if untrusted people can access the `cmgui` files in which the password is stored. These are kept on the machine that runs the `cmgui` process, which may not be the machine that displays `cmgui` on its screen.

Avoiding Lag With `cmgui`

The `cmgui` front-end is often run on the head node of the cluster via an ssh -X login, with the -X option passing the X11 protocol instructions. With such a connection, more changes being tracked on the front end means more X11 protocol instructions, and therefore a more sluggish interface. Additionally, the transport lag for X11 instructions is greater if the X-display server, where the administrator is viewing `cmgui`, is some distance away from the head where `cmgui` is actually running. This can lead to an irritating, or even an unusable `cmgui` in extreme cases.

Then there is the common case when there is a bastion host or a similar "jump" host in the way, which prevents running `cmgui` from the head node directly via ssh -X. A way to deal with the bastion/jump host problem is to carry out an ssh -X login to the bastion host first, and then to carry out an ssh -X login to the head node to run the `cmgui` session. This *double-ssh -X* connection is easy enough to implement. However it has several problems.

- The X11-protocol overhead and lag becomes larger because there are now two jumps between the cluster and the remote display server. Viewing `cmgui` in this way is therefore likely to be even more irritating than before.
- Microsoft windows `cmgui` clients will need to be displayed over an X-display server too, which means installing an X-display server for MS windows. This is often undesirable.

Because of these problems, alternatives to double-ssh -X are therefore often used:

- One workaround is to run `cmgui` over an ssh -X connection from the bastion host, and then use VNC (a remote desktop viewing software) from the remote display-server to the bastion host. This is usually more pleasant for interactive use than double-ssh -X because
  - the bastion host is usually close to the cluster, thus minimizing the X11 lag from the cluster head node to the bastion host.
  - using VNC in the connection to transport the desktop image from the bastion host to the remote display avoids the X11 protocol transport lag issue entirely for that part of the connection.

    For implementing VNC, an administrator should note that:
    * most modern standard versions of VNC can connect with compression
    * encryption and authentication for VNC must be enabled in order to avoid unauthorized VNC access. If enabling these is a problem, then the well-trusted OpenVPN utility can be used to provide these in a separate layer instead with good results.
- Usually the best workaround to set up `cmgui` to display on the desktop is to run `cmgui` as the desktop client on the remote display, and set up ssh port-forwarding on the bastion host. This again avoids the transport overhead and lag of X11 over double-ssh -X, and indeed avoids transporting the X11 protocol entirely. Instead, the ssh connection transports the CMDaemon SOAP calls directly. These SOAP calls run encrypted over the SSL port 8081 already, so that all that is needed is to forward them, which is what the ssh port-forwarding feature is able to do quite easily. The additional encryption from ssh on top of the SSL encryption already operating on the SOAP calls does no harm. It is unlikely to be a performance bottleneck, and in any case can be switched off or speeded up in some implementations or compiler options of ssh.

  The ssh port-forwarding connection can be configured by following these steps:
  - For failover clusters, to connect to the active node, `cmgui` needs to be aware that it should not follow redirection. To arrange this, the `cmgui` settings file `~/.cm/cmgui/clusters.dat` should be modified on the machine where the `cmgui` desktop client is running. The safest way to carry out the modification is to first make sure that the file is created by starting up the desktop `cmgui`, then stopping it. The `followRedirect` line in the `clusters.dat` file should then be changed to:

```
followRedirect = false;
```

Single-head clusters require no such change.

– Port-forwarding can then be set up from the cluster.

If these parameters are used:

* *cluster-ip*: the cluster address. This is the shared alias IP address for a failover cluster
* *bast*: the bastion host.
* *user*: the user name that the administrator is using
* *admin-desktop*: the remote machine

Then:

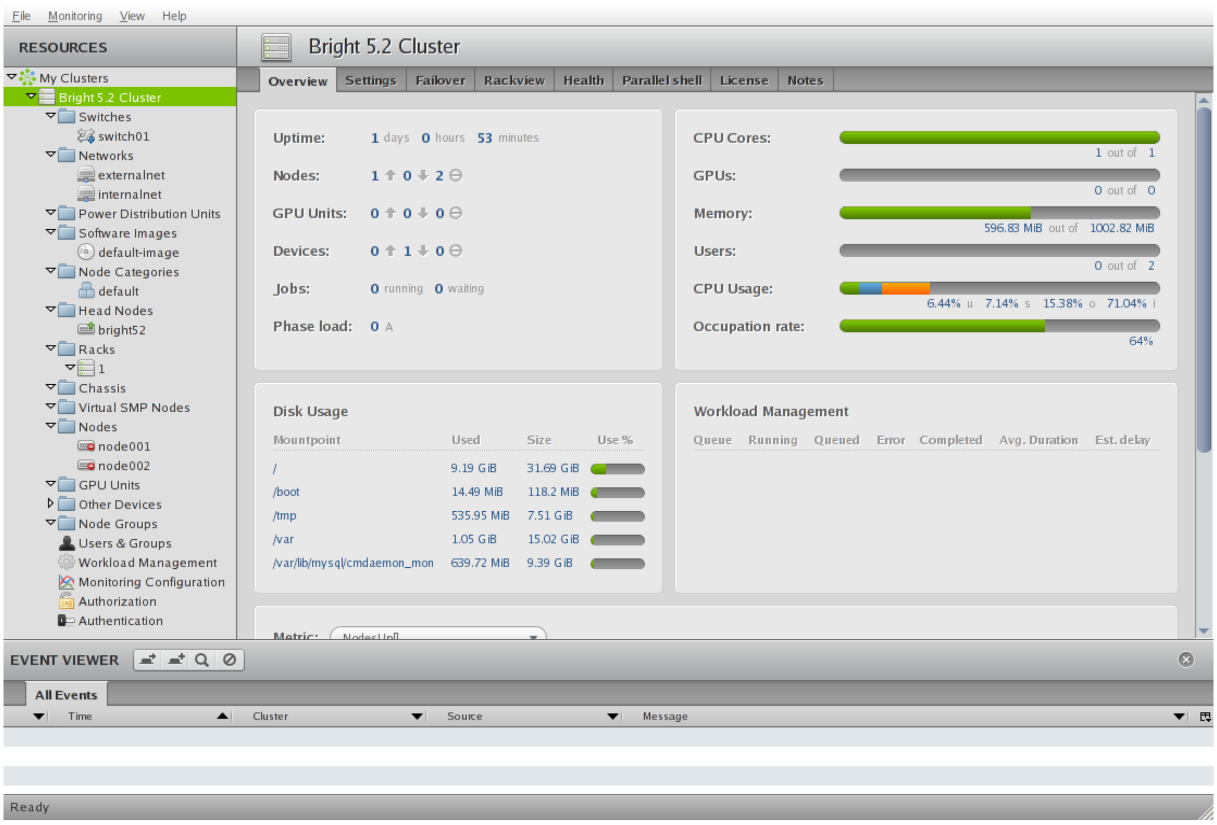
* If the bastion host allows no inbound ssh connections from the administrator's remote desktop, then:
  · a remote-port-forwarding tunnel can be set up on it by running:

    ```
    bast:~$ ssh -R 8081:cluster-ip:8081 user@admin-desktop
    ```

    A login to *admin-desktop* is prompted for, as part of the execution of the preceding command, and the login should be carried out.
  · on the remote desktop, after `cmgui` is started up, a connection to `localhost:8081` should be made. The value `localhost:8081` can be set in the `cmgui` cluster connection parameters dialog shown in figure 2.3.

Bright Cluster Manager `cmgui` Default Display On Connection

Clicking on the `Connect` button establishes a connection to the cluster, and `cmgui` by default then displays a tabbed pane overview screen of the cluster. This default screen presents summarized data for the entire cluster (figure 2.4):



Bright 5.2 Cluster

Uptime: 1 days 0 hours 53 minutes
Nodes: 1 ↑ 0 ↓ 2 ⊖
GPU Units: 0 ↑ 0 ↓ 0 ⊖
Devices: 0 ↑ 1 ↓ 0 ⊖
Jobs: 0 running 0 waiting
Phase load: 0 A

CPU Cores: 1 out of 1
GPUs: 0 out of 0
Memory: 596.83 MiB out of 1002.82 MiB
Users: 0 out of 2
CPU Usage: 6.44% u 7.14% s 15.38% o 71.04% i
Occupation rate: 64%

Disk Usage

Mountpoint	Used	Size	Use %
/	9.19 GiB	31.69 GiB	
/boot	14.49 MiB	118.2 MiB	
/tmp	535.95 MiB	7.51 GiB	
/var	1.05 GiB	15.02 GiB	
/var/lib/mysql/cmdaemon_mon	639.72 MiB	9.39 GiB	

Workload Management

Queue	Running	Queued	Error	Completed	Avg. Duration	Est. delay

Metric: NodesUp

Figure 2.4: Cluster Overview