Bright Cluster Manager 7.0

Cloudbursting Manual

Revision: 6181

Date: Thu, 07 May 2015

Trademarks

Linux is a registered trademark of Linus Torvalds. PathScale is a registered trademark of Cray, Inc. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. SUSE is a registered trademark of Novell, Inc. PGI is a registered trademark of The Portland Group Compiler Technology, STMicroelectronics, Inc. SGE is a trademark of Sun Microsystems, Inc. FLEXlm is a registered trademark of Globetrotter Software, Inc. Maui Cluster Scheduler is a trademark of Adaptive Computing, Inc. ScaleMP is a registered trademark of ScaleMP, Inc. All other trademarks are the property of their respective owners.

Rights and Restrictions

All statements, specifications, recommendations, and technical information contained herein are current or planned as of the date of publication of this document. They are reliable as of the time of this writing and are presented without warranty of any kind, expressed or implied. Bright Computing, Inc. shall not be liable for technical or editorial errors or omissions which may occur in this document. Bright Computing, Inc. shall not be liable for any damages resulting from the use of this document.

Limitation of Liability and Damages Pertaining to Bright Computing, Inc.

The Bright Cluster Manager product principally consists of free software that is licensed by the Linux authors free of charge. Bright Computing, Inc. shall have no liability nor will Bright Computing, Inc. provide any warranty for the Bright Cluster Manager to the extent that is permitted by law. Unless confirmed in writing, the Linux authors and/or third parties provide the program as is without any warranty, either expressed or implied, including, but not limited to, marketability or suitability for a specific purpose. The user of the Bright Cluster Manager product shall accept the full risk for the quality or performance of the product. Should the product malfunction, the costs for repair, service, or correction will be borne by the user of the Bright Cluster Manager product. No copyright owner or third party who has modified or distributed the program as permitted in this license shall be held liable for damages, including general or specific damages, damages caused by side effects or consequential damages, resulting from the use of the program or the un-usability of the program (including, but not limited to, loss of data, incorrect processing of data, losses that must be borne by you or others, or the inability of the program to work together with any other program), even if a copyright owner or third party had been advised about the possibility of such damages unless such copyright owner or third party has signed a writing to the contrary.

Table of Contents

Preface

Welcome to the *Cloudbursting Manual* for Bright Cluster Manager 7.0.

0.1 About This Manual

This manual is aimed at helping cluster administrators install, understand, configure, and manage the cloud capabilities of Bright Cluster Manager. The administrator is expected to be reasonably familiar with the *Administrator Manual*.

0.2 About The Manuals In General

Regularly updated versions of the Bright Cluster Manager 7.0 manuals are available on updated clusters by default at `/cm/shared/docs/cm`. The latest updates are always online at `http://support.brightcomputing.com/manuals`.

- The *Installation Manual* describes installation procedures for the basic cluster.
- The *Administrator Manual* describes the general management of the cluster.
- The *User Manual* describes the user environment and how to submit jobs for the end user.
- The *Developer Manual* has useful information for developers who would like to program with Bright Cluster Manager.
- The *OpenStack Deployment Manual* describes how to deploy OpenStack with Bright Cluster Manager.
- The *Hadoop Deployment Manual* describes how to deploy Hadoop with Bright Cluster Manager.
- The *UCS Deployment Manual* describes how to deploy the Cisco UCS server with Bright Cluster Manager.

If the manuals are downloaded and kept in one local directory, then in most pdf viewers, clicking on a cross-reference in one manual that refers to a section in another manual opens and displays that section in the second manual. Navigating back and forth between documents is usually possible with keystrokes or mouse clicks.

For example: `<Alt>-<Backarrow>` in Acrobat Reader, or clicking on the bottom leftmost navigation button of xpdf, both navigate back to the previous document.

The manuals constantly evolve to keep up with the development of the Bright Cluster Manager environment and the addition of new hardware and/or applications. The manuals also regularly incorporate customer feedback. Administrator and user input is greatly valued at Bright

Computing. So any comments, suggestions or corrections will be very gratefully accepted at manuals@brightcomputing.com.

0.3 Getting Administrator-Level Support

Unless the Bright Cluster Manager reseller offers support, support is provided by Bright Computing over e-mail via support@brightcomputing.com. Section 10.2 of the *Administrator Manual* has more details on working with support.

1

Introduction

In weather, a cloudburst is used to convey the idea that a sudden flood of cloud contents takes place. In cluster computing, the term *cloudbursting* conveys the idea that a flood of extra cluster capacity is made available when needed from a cloud computing services provider such as Amazon.

Bright Cluster Manager implements cloudbursting for two scenarios:

1. A "Cluster-On-Demand", or a "pure" cloud cluster (chapter 2). In this scenario, the entire cluster can be started up on demand from a state of non-existence. All nodes, including the head node, are instances running in a coordinated manner entirely inside the cloud computing service.

2. A "Cluster Extension", or a "hybrid" cloud cluster (chapter 3). In this scenario, the head node is kept outside the cloud. Zero or more regular nodes are also run outside the cloud. When additional capacity is required, the cluster is extended via cloudbursting to make additional nodes available from within the cloud.

Chapters 2 and 3 deal with GUI configuration of the two cloudbursting scenarios.

Chapter 4 looks at cloudbursting configuration using command line tools.

Chapter 5 discusses some miscellaneous aspects of cloudbursting.

Chapter 6 describes the concepts, including networking, behind setting up a "private" cloud cluster on a virtual private network using the Amazon VPC infrastructure.

2

Cluster-On-Demand Cloudbursting

Requirements

If the cloud provider is Amazon, then Cluster-On-Demand cloudbursting (the case of starting up a "pure" cloud cluster) requires:

- an Amazon account
- registration on the Bright Computing Customer Portal website at `http://www.brightcomputing.com/Customer-Login.php`
- a Bright Cluster Manager product key. The key is obtained at the Customer Portal website specifically for a Cluster-On-Demand setup, from the `Burst!` menu. This key is later activated when the license is installed (section 2.2) on the head node. The head node and regular nodes in this case are in the cloud.

Steps

The following steps are then carried out to start up the head node and regular nodes of the cloud cluster:

- a head node instance is launched from a browser, using the Amazon management console (section 2.1)
- the head node instance is logged into via `ssh` and the cluster is configured (section 2.2)
- the regular nodes are started up from the head node using `cmsh` or `cmgui` to power them up (section 2.4)

These steps are now covered in more detail.

2.1 Cluster-On-Demand: Launching The Head Node From The Cloud Provider

Launching a head node from within Amazon is covered in this section.

2.1.1 Getting To The "Launch Instance" Button

The Amazon management console can be logged into from `https://console.aws.amazon.com/console/` by using the e-mail address and password of the Amazon account (figure 2.1).

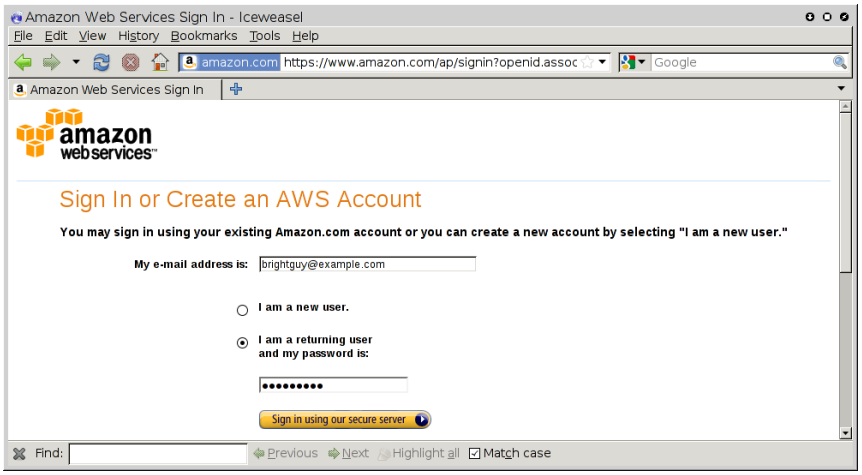

Figure 2.1: Logging Into The Amazon Management Console

By default, on login, the management console displays a list of accessible Amazon web services (figure 2.2).

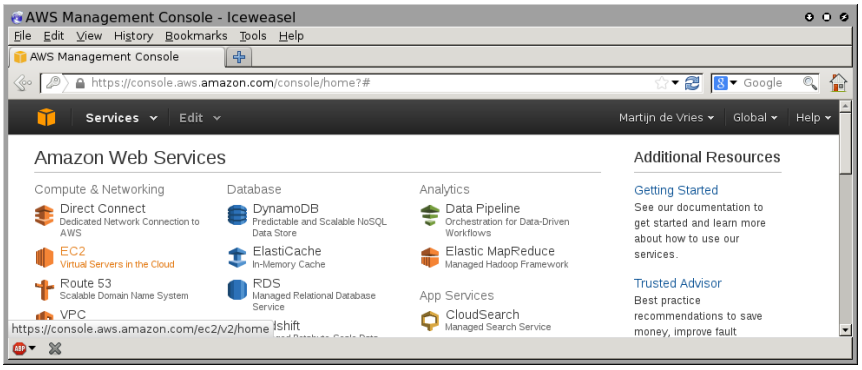

Figure 2.2: Amazon Management Console: Accessible Services

To set up the Cluster-On-Demand cluster, the `EC2` service within the `Compute & Networking` grouping should be clicked. This brings up the `EC2 Dashboard`, which is also the top link of a resource tree that is displayed in a `Navigation` pane (figure 2.3).

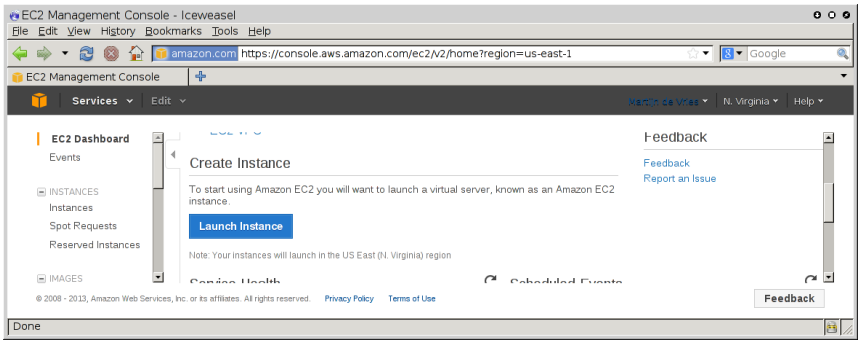

Figure 2.3: The EC2 Dashboard With The "`Launch Instance`" Button

In the main pane of the dashboard is the `Launch Instance` button. Clicking it starts up Amazon's Launch Instance Wizard. Amazon documentation for the wizard is at `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html`.

Using the wizard to launch a head node instance is described next.

2.1.2 Launching The Head Node Instance

To start a Cluster-On-Demand cluster, a head node instance must first be launched. This can be done as follows:

- `Step 1: Choose an Amazon Machine Image (AMI)`: The first step in the wizard offers a choice of `Select` buttons to launch an instance from an AMI image (figure 2.4).

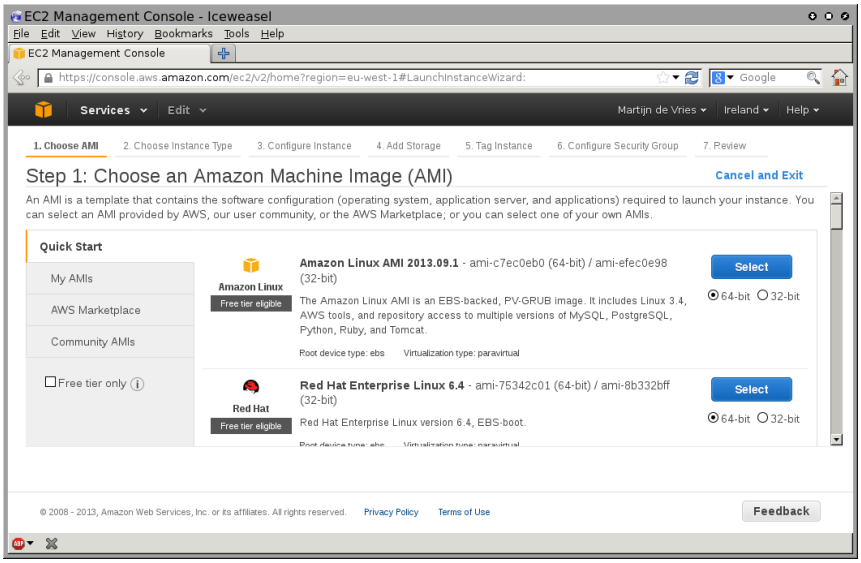

Figure 2.4: EC2: Choosing An AMI, Step 1

The default AMIs can be ignored. Clicking on the `Community AMIs` link in the left navigation pane brings up a new display of community AMIs. Entering a search text of "`brightheadnode`" then shows only the AMIs appropriate for a Bright Cluster Manager head node instance in a Cluster-On-Demand cluster. These are:

1. An AMI that uses standard XEN paravirtualization technology. This is available for all regions. If this image is used,

hardware virtualization extensions acceleration is not implemented, even if available in the underlying cloud node hardware.

2. An AMI with `hvm` in the name. This is available for some regions. It is intended for use in regions that support HVM (Hardware Virtual Machines). HVM requires that the CPU used has the Intel VT or AMD-V virtualization extensions, to implement hardware acceleration for virtualized machines. At the time of checking (April 2013):

   – Regions supporting HVM are `eu-west-1`, `us-east-1`, and `us-west-2`.
   – Instance types supporting HVM are the `m3.xlarge` instance type, and higher. Instance types (`http://aws.amazon.com/ec2/instance-types/`) are a way of characterizing machine specifications, such as whether it has more RAM, more cores, or HVM.

   Updated details on the regions and instance types that Amazon EC2 supports can be found via the Amazon website, `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html`.

Clicking on the `Select` button for the appropriate XEN or HVM head node AMI then brings up the next step in the launch wizard:

- `Step 2: Choose an Instance Type`: This displays a *micro instance* by default (figure 2.5).

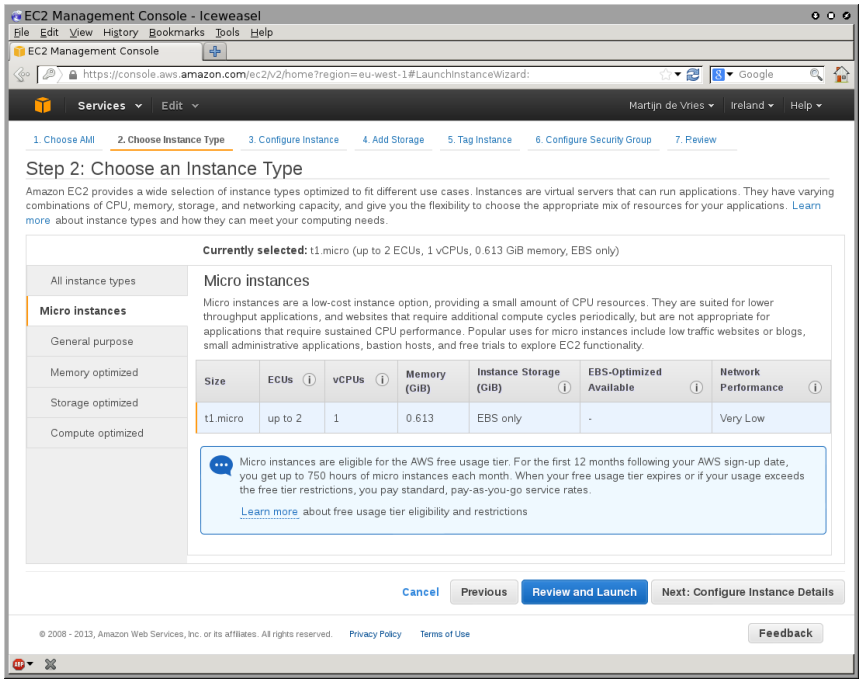

Figure 2.5: EC2: Choosing An AMI, Step 2

The `t1.micro` is the smallest and least powerful type of instance that a head node can run as, but is only useful for quite minor

testing. It is likely to be overwhelmed when doing any significant work. A more reasonable node type to use for testing is therefore the `m1.small` type, which is available under the `General purpose` navigation option of this window.

Steps 3 to 6 that follow are optional and can be skipped, by going ahead to `Step 7: Review Instance Launch`.

- `Step 3: Configure Instance Details`: Among other instance options, this optional step allows the following to be set:
  - `Purchasing option`, for spot instances (section 5.3.1)
  - `Network` This is a choice of `EC2-Classic` or EC2-VPC instances (section 6.1.1)
  - `Availability Zone`, for if there is a preference for the location of the instance. Nodes within the same availability zone can connect with low latency and high bandwidth to each other. They are also isolated from other availability zones in the same region, which reduces the risk of network outages from another zone affecting them. By default, no preference is specified for the head node, nor for the cloud nodes later. This is because spot pricing can increase as nodes from an availability zone become scarce, which may conflict with the intention of the administrator. The default availability setting for a cloud account can be set in `cmsh` from within `cloud` mode:

    Example

    ```
    [bright70->cloud[Spare Capacity]]% set defaultavailabilityzone
    ```

  - `Shutdown behavior`, to decide if the instance should be stopped (kept around) or terminated (removed).
- `Step 4: Add Storage`: Among other storage options, this optional step allows the following options to be set:
  - `Size (GB)`: The size of storage to be added to the instance
  - `Type`: Whether the storage is EBS or ephemeral
  - `Device`: A device name, chosen from `/dev/sdb` onwards, since `/dev/sda` is already taken
  - `Delete on Termination`: Whether the device is deleted when the instance terminates

  By default, the instance has a `Type` called `Root`, which is a special EBS volume. It has a default `Size (GB)` of 80, which can be edited.

  For most instances other than `micro`, a certain amount of ephemeral storage is provided by Amazon for free, and can then be set for the root device in the `Storage Device Configuration` options of this screen. The EBS and ephemeral storage types are described in section 3.2.1.
- `Step 5: Tag instance`: This optional step allows the addition of metadata to an instance, via assignment of key-value pairs. A default key of `Name` is presented, and the administrator should put in a name for the instance as the associated value. The associated value can be arbitrary.

- `Step 6: Configure Security Group`: This optional step allows a *security group* to be defined. A security group is a configuration that defines how network access to the instance is allowed. By default all access to the instance is blocked, apart from SSH access.

  – Default: SSH inbound allowed. This means that `cmsh` can be used to control the Cluster-On-Demand cluster via SSH just like a regular cluster.

  Inbound connections can be defined, based on protocol, packet type, port, and source in CIDR specification. For example, allowing inbound connections via TCP port 8081 from anywhere (0.0.0.0/0) allows `cmgui` to communicate via its custom protocol with the default CMDaemon back end on the head node.

  The default security group setting should also be modified by the administrator at this point if a standalone `cmgui` is to be used to control the cluster (section 2.3). For regular use in a cluster-on-demand setup, lag is reduced if a standalone `cmgui` is used rather than running a `cmgui` originating from the head node via an `ssh -X` connection.

- `Step 7: Review Instance Launch`: The configuration so far can be reviewed. On clicking the `Launch` button, a pop-up dialog for "`Select an existing key pair or create a new key pair`" is displayed (figure 2.6).

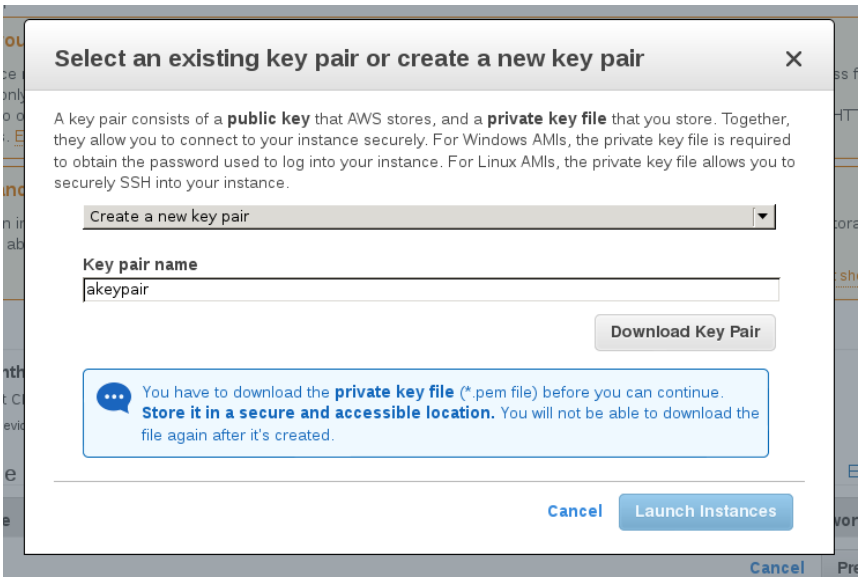

Figure 2.6: EC2: Choosing An AMI, Step 7 - Keypair generation/creation dialog

This dialog allows the creation and storage of a cryptographic key pair. It can alternatively allow an existing pair to be used from the "`Select a key pair`" selection. The private key of the key pair is used in order to allow SSH access to the head node instance when it is up and running.

After the instance is launched, the web session displays a window informing that the instance is being started up.

2.1.3 Managing A Head Node Instance With The AWS EC2 Management Console

A *newly-launched* head node instance, after it is fully up, is a fully-booted and running Linux instance, but it is not yet a fully-configured head node. That is, it is capable of running Bright Cluster Manager, but it is not yet running it, nor is it working with compute nodes at this point. The steps to make it a fully-configured head node are covered in section 2.2.

For now, the newly-launched head node instance can be watched and managed without Bright Cluster Manager in the following ways.

Status checking via instance selection from instances list:
Clicking the `Instances` menu resource item from the navigation pane opens up the "`Instances`" pane. This lists instances belonging to the account owner. An instance can be marked by ticking its checkbox. Information for the selected instance is then displayed in the lower main pane (figure 2.7).

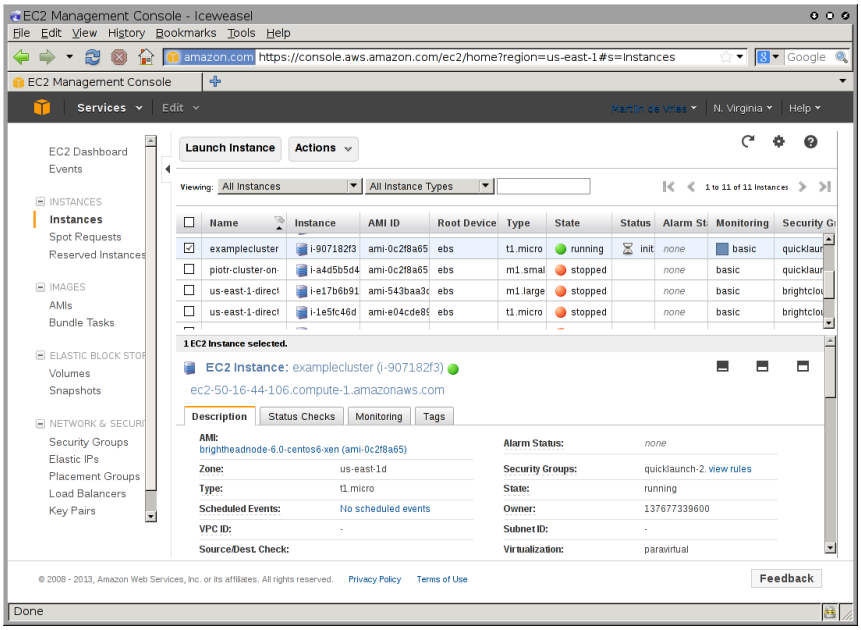

Figure 2.7: The EC2 Instances List

System (Amazon machine infrastructure) and instance (instance running under the infrastructure) reachabilities are similarly shown under the neighboring "`Status Checks`" tab (figure 2.8).

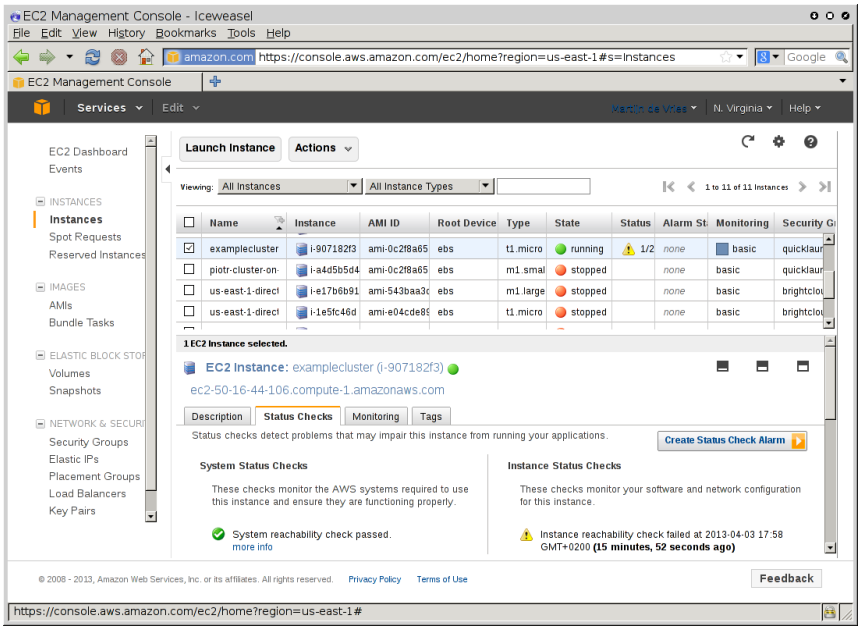

Figure 2.8: Reachability Status For An EC2 Instance

Acting on an instance from the AWS EC2 Management Console:
An instance can be marked by clicking on it. Clicking the `Actions` button near the top of the main center pane, or equivalently from a right-mouse-button click in the pane, brings up a menu of possible actions. These actions can be executed on the marked instance, and include the options to `Start`, `Stop` or `Terminate` the instance.

Connecting to an instance from the AWS EC2 Management Console:
A marked and running instance can have an SSH connection made to it. Clicking on the `Connect` button near the top of the main center pane displays a pop-up text that guides the user through the connection options for a running instance. These connection options are via:

- **a standalone SSH client**

  There is further documentation on this at:

  - `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html` for Linux clients
  - `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html` for PuTTY users

- **a browser-based Java SSH client, MindTerm**

  There is further documentation on this at:

  - `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mindterm.html`

Most administrators should find the pop-up text enough, and the further documentation unnecessary.

The standalone SSH client help text displays instructions (figure 2.9) on how to run `ssh` from the command line to access the marked instance.

If the launched head node is fully up then a login using those instructions succeeds.

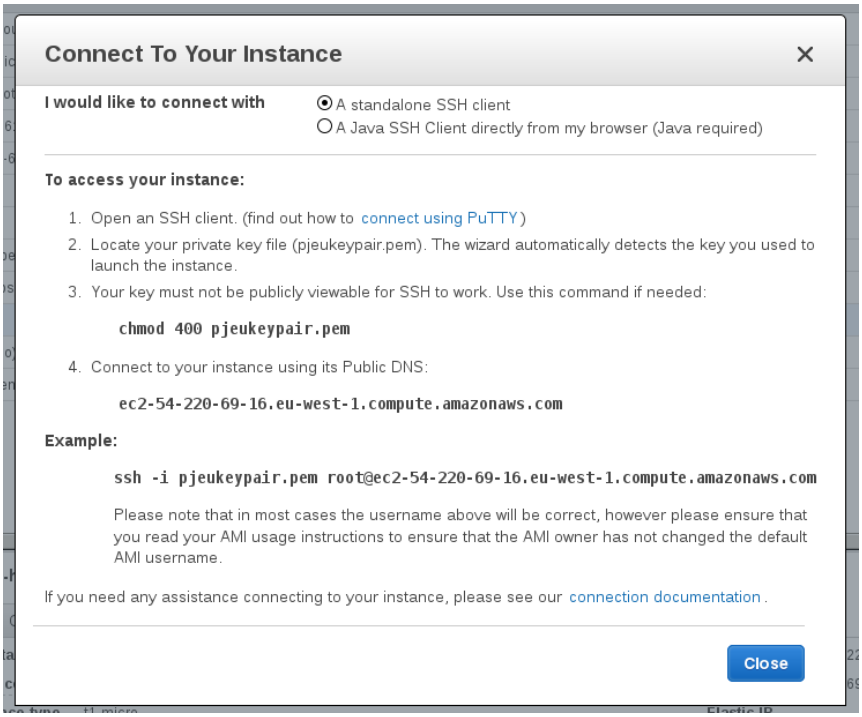

Figure 2.9: SSH Instructions To Connect To The Marked Instance

Viewing the head node console:

The head node takes about 2 minutes to start up. If, on following the instructions, an SSH connection cannot be made, then it can be worth checking the head node system log to check if the head node has started up correctly. The log is displayed on right-clicking on the "`Actions`" button and selecting the "`Get System Log`" menu item (figure 2.10). A successful start of the system generates a log with a tail similar to that of figure 2.10.

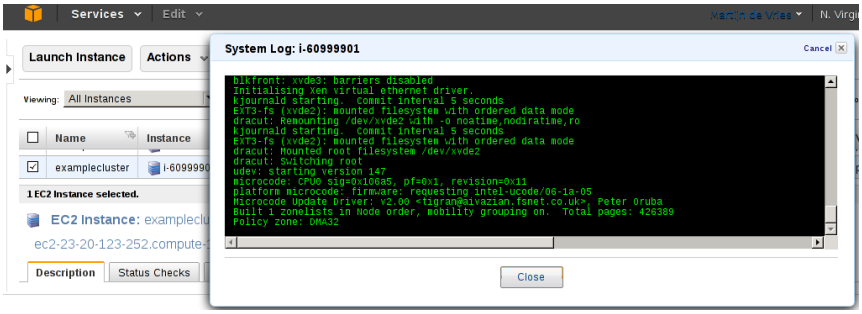

Figure 2.10: System Log Of The Checkboxed Instance

If the system and network are working as expected, then an SSH connection can be made to the head node to carry out the next step, which is the configuration of the head node and cluster.

2.2 Cluster-On-Demand: Head Node Login And Cluster Configuration

After the Amazon console manager has started up a head node instance, the head node instance and cluster must be configured. Logging into the head node via `ssh` allows this.

On logging in for the first time, the system suggests that the `bright-setup` script be run:

Example

```
pj@office:~$ ssh -i pjkeypair.pem root@ec2-176-34-160-197.eu-west-1.com\
pute.amazonaws.com
The authenticity of host 'ec2-176-34-160-197.eu-west-1.compute.amazonaw\
s.com (176.34.160.197)' can't be established.
RSA key fingerprint is 66:1e:f3:77:83:f8:3f:42:c1:b7:d2:d5:56:d8:c3:58.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-176-34-160-197.eu-west-1.compute.amazon\
aws.com,176.34.160.197' (RSA) to the list of known hosts.
Welcome to Bright Cluster Manager

                                                  Based on Scientific Linux 5
                                                  Cluster Manager ID: #999915

--------------------------------------------------------------------------------

To set up your cluster, type

  bright-setup

and follow the instructions

Creating DSA key for ssh
[root@headnode ~]#
```

Running the `bright-setup` script goes through several screens, some of which prompt for input. At some prompts, it is hinted that typing "`I`" gives further explanation about the input.

The screens go through the following issues:

- The license agreement.
- Amazon Web Services account information. This asks for the AWS Username, AWS Account ID, Access Key ID, and Secret Access Key. These are needed by Bright Cluster Manager to manage the cloud node instances.
- The installation of the Bright Computing product key (formatted like 868868-797099-979199-091301-134414). This is the cloud version of the `request-license` command in section 4.3 of the *Installation Manual*, and asks for:
  - The organization information for the license. This requires input for the fields: country, state, locality, organizational unit, unit and cluster name.

  - The values to be used for the head node machine name and its administrative password (used for the root and MySQL passwords).

- Optionally, setting up the secondary drives using Amazon's EBS service.
- Optionally, setting up extra storage for `/home` using Amazon's EBS service.
- Optionally, setting up extra storage for monitoring data (recommended for more than 500 nodes).
- Setting up cloud node instance types. Amazon instance types (`http://aws.amazon.com/ec2/instance-types/`) are choices presented from node specifications consisting of memory, storage, cores, GPUs, and others. The setting up of the instance type step is looped through if necessary to allow more than one instance type to be configured.
  - Setting up
    * the number <*N*> of cloud compute nodes and
    * their base name (the `cnode` part of the name if the nodes have the names `cnode001` to `cnode`<*N*>).
  - Setting up the amount of storage for the cloud compute node instance.

    The default disk partitioning layout for nodes can be modified as described in Appendix D of the *Administrator Manual*. Using diskless nodes is also possible if the cloud compute node instance has enough RAM—about 2GB at the time of writing.

  Setting these values causes the cloud node objects to be created in the CMDaemon database. Cloud nodes are not however actually started up at this stage. Starting up must be done explicitly, and is covered in section 2.4.
- Setting up the workload manager, along with the number of slots, and if the head node is to be used for compute jobs too.

After accepting the input, the `bright-setup` script runs through the configuration of the head node and the cloud nodes. Its progress is indicated by output similar to the following:

Example

```
[root@headnode ~]# bright-setup
Retrieving Amazon instance information

------------------------------------------------------------------------
|                            License agreements                        |
------------------------------------------------------------------------
The end user license agreement will be shown in a new screen. To exit
this screen, type 'q'. Press any key to continue
Do you agree with the license terms and conditions? (yes, no, show): yes
```

```
------------------------------------------------------------------------
|                        Amazon account information                    |
------------------------------------------------------------------------
AWS username (I for information): exampleuser@brightcomputing.com
AWS Account ID (I for information): 313234312222
Access Key ID (I for information): OUPOUASOUDSSSAOU
Secret Access Key (I for information): Aighei8EooLi1Dae8Nio5ohl4ieXiAiaiV

Verifying Amazon credentials

------------------------------------------------------------------------
|                               Bright License                         |
------------------------------------------------------------------------
Bright Product Key (I for information):
                                        423112-231432-134234-132423-134221
   Country: US
   State: CA
   Locality: San Francisco
   Organization name: Bright Computing
   Organizational Unit: Development
   Cluster Name: demo-cluster

------------------------------------------------------------------------
|                                 Head Node                            |
------------------------------------------------------------------------
Hostname: bright70
Administrative Password:
Verify password:
Do you want to create a second drive for more storage capacity?
(I for information) [YES|no] no
Extra storage for /home (I for information)? [NO|yes] n
Extra storage for monitoring data (I for information)? [NO|yes] n

------------------------------------------------------------------------
|                               Compute Nodes                          |
------------------------------------------------------------------------
Instance Type (I for information):
  m1.small
  m1.medium
  c1.medium
  m1.large
  t1.micro
  m2.xlarge
  m2.2xlarge
  m2.4xlarge
  c1.xlarge
  [t1.micro]
>
t1.micro
Node Count [2]:
2
Base name (I for information) [cnode]:
cnode
Instances of type t1.micro need to use EBS storage
Size of EBS (GB) [40]: 15
```

```
Do you want to configure more node types? [NO|yes]
no

------------------------------------------------------------------------
|                          Workload Management system                  |
------------------------------------------------------------------------
Which workload management system do you want to use? (I for information)?
  slurm
  sge
  torque
  [slurm]
>
slurm
Number of slots per node [8]:
8
Do you want to use the head node for compute jobs? [NO|yes]
no

The following information will be used to configure this head node:

  Amazon information:
    AWS username:             exampleuser@brightcomputing.com
    AWS Account ID:           313234312222
    Access Key ID:            OUPOUASOUDSSSAOU
    Secret Access Key:        Aighei8EooLi1Dae8Nio5ohl4ieXiAin5eeRoaiV
  Bright Product Key:         423112-231432-134234-132423-134221
  License information
    Country:                  US
    State:                    CA
    Locality:                 San Francisco
    Organization name:        Bright Computing
    Organizational Unit:      Development
    Cluster Name:             demo-cluster
  Hostname:                   bright70
  Second drive size:          no
  Instance Type:              t1.micro
  Node Count:                 2
  Base name:                  cnode
  Storage type:               EBS
  Size of storage:            15 GB
  Workload management system: slurm
  Number of slots:            8
  Head node for compute jobs: no

The information to configure this head node has been collected, and shown
above. The next phase will be to process this information. A new Bright
license will be installed, the Bright Cluster Manager software will be
initialized and the workload management software will be initialized
Do you want to continue? [YES|no]
yes

Starting to configure this head node
Successfully retrieved the license
Installed license
Initializing Bright Cluster Manager
```

```
Installing admin certificates
Configuring default scheduler, slurm
Set up finished
```

It is recommended that the system administrator log out and login again after the script has been run, in order to enable the new environment for the shell that the administrator is in. If the hostname was changed in the `bright-setup` script, for example, the name change shows up in the shell prompt only after the re-login.

Once there is a head in the cloud, the other cloud nodes can be started up.

2.3 Cluster-On-Demand: Connecting To The headnode Via `cmsh` or `cmgui`

Amazon provides a security group to each instance. By default, this configures network access so that only inbound SSH connections are allowed from outside the cloud. A new security group can be configured, or an existing one modified, using the `Edit details` button in figure 2.11. Security groups can also be accessed from the navigation menu on the left side of the EC2 Management Console.

2.3.1 Cluster-On-Demand: Access With A Remote, Standalone `cmgui`

The security group defined by Amazon for the head node can be modified by the administrator to allow remote connections to CMDaemon running on the head node (figure 2.11).

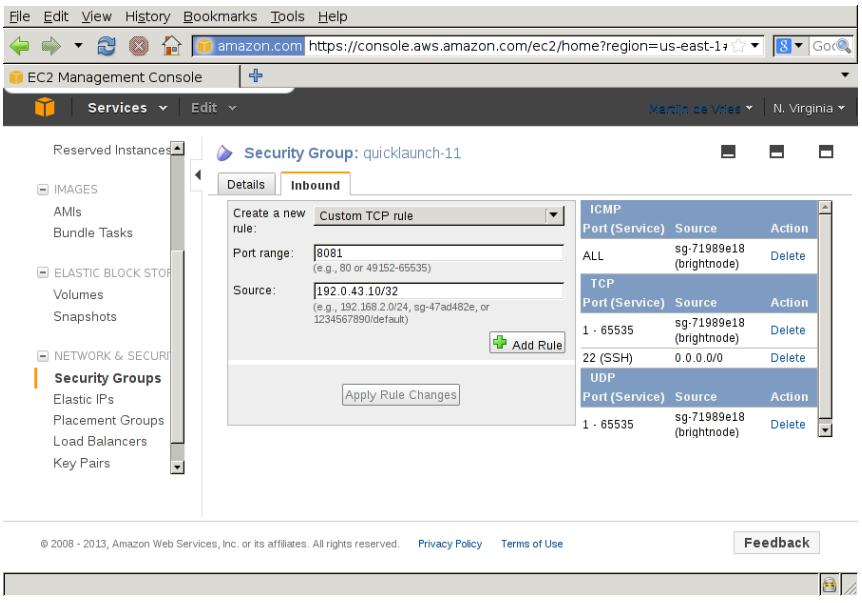

Figure 2.11: Security Group Network And Port Access Restriction

- To allow only a specific network block to access the instance, the network from which remote connections are allowed can be specified in CIDR format.
- Explicitly allowing inbound connections to port 8081 on the head node allows the standalone `cmgui` (section 2.4 of the *Administrator*

Manual) to connect to the head node. This is because the `cmgui` back end, which is CMDaemon, communicates via port 8081.

2.3.2 Cluster-On-Demand: Access With A Local `cmsh`

The security group created by Amazon by default already allows inbound SSH connections from outside the cloud to the instance running in the cloud, even if the incoming port 8081 is blocked. Launching a `cmsh` session within an SSH connection running to the head node is therefore possible, and works well.

2.3.3 Cluster-On-Demand: Access With A Local `cmgui`

It is possible to run an X-forwarded `cmgui` session from within an `ssh -X` connection that is already running to the head node. However, it suffers from significant X-protocol lag due to the various network encapsulation layers involved. The procedure described earlier for cluster-on-demand access with the remote, standalone `cmgui` from outside the cloud is therefore recommended instead for a more pleasant experience.

2.4 Cluster-On-Demand: Cloud Node Start-up

Cloud nodes must be explicitly started up. This is done by powering them up, assuming the associated cloud node objects exist. The cloud node objects are typically specified in the `bright-setup` script—in the preceding example the cloud node objects are `cnode001` and `cnode002`.

However, more cloud node objects can be created if needed after the `bright-setup` script has run. The maximum number that may be created is set by the license purchased.

Large numbers of cloud node objects can be created with Bright Cluster Manager as follows:

- In `cmgui` they are conveniently created with the Node Creation Wizard as described in section 3.3. Several of the steps described in that section are specific to Cluster Extension clusters. These steps are not needed for Cluster-On-Demand clusters, and therefore do not come up when the wizard is used in this case.
- In `cmsh` a large number of cloud node objects can conveniently be created with the "`foreach --clone`" command instead, as described in section 4.3.

After creation, individual cloud nodes can be powered up from within `cmgui` by a right-click on the cloud node resource item (figure 2.12).

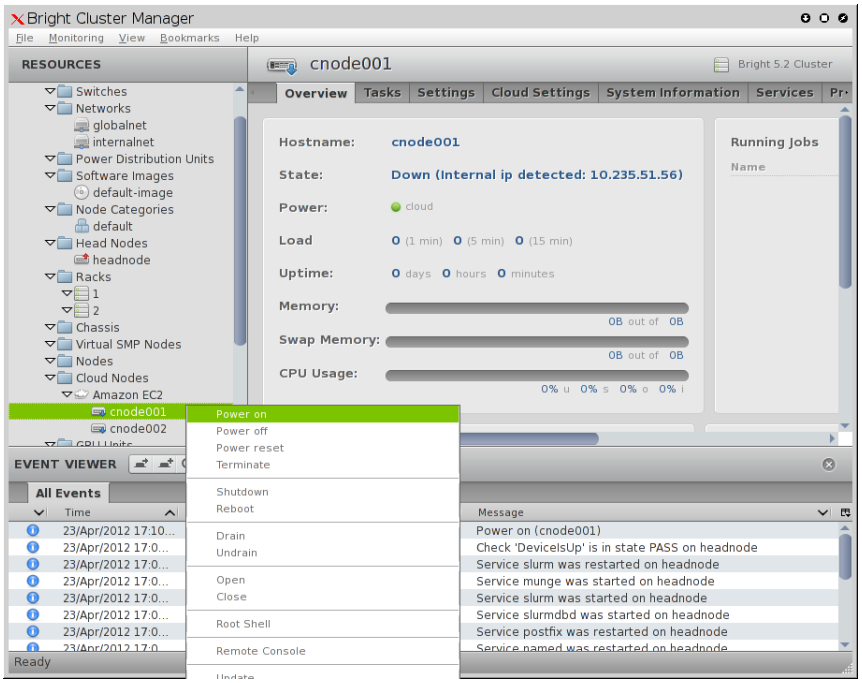

Figure 2.12: Powering on a cloud node with `cmgui`

As with regular non-cloud nodes, multiple cloud nodes can be powered up in `cmgui` by selecting them from the `Overview` tabbed pane. Switching to the `Tasks` tabbed pane and clicking on the power on button then powers them up.

As with regular non-cloud nodes, cloud nodes can also be powered up from within the `device` mode of `cmsh`. The initial power status (section 4.1 of the *Administrator Manual*) of cloud nodes is `FAILED`, because they cannot be communicated with. As they start up, their power status changes to `OFF`, and then to `ON`. Some time after that they are connected to the cluster and ready for use. The device status (as opposed to the power status) remains `DOWN` until it is ready for use, at which point it switches to `UP`:

Example

```
[head1->device]% power status
cloud ................. [  FAILED ] cnode001 (Cloud instance ID not set)
cloud ................. [  FAILED ] cnode002 (Cloud instance ID not set)
No power control ...... [ UNKNOWN ] head1
[head1->device]% power on -n cnode001
cloud ................. [   ON    ] cnode001
[head1->device]% power status
cloud ................. [   OFF   ] cnode001 (pending)
cloud ................. [  FAILED ] cnode002 (Cloud instance ID not set)
No power control ...... [ UNKNOWN ] head1
[head1->device]% power on -n cnode002
cloud ................. [   ON    ] cnode002
[head1->device]% power status
cloud ................. [   ON    ] cnode001 (running)
cloud ................. [   OFF   ] cnode002 (pending)
No power control ...... [ UNKNOWN ] head1
```

```
[head1->device]% !ping -c1 cnode001
ping: unknown host cnode001
[head1->device]% status
head1 ..................... [   UP   ]
node001 ................... [   UP   ]
node002 ................... [  DOWN  ]
[head1->device]% !ping -c1 cnode001
PING cnode001.cm.cluster (10.234.226.155) 56(84) bytes of data.
64 bytes from cnode001.cm.cluster (10.234.226.155): icmp_seq=1 ttl=63 t\
ime=3.94 ms
```

Multiple cloud nodes can be powered up at a time in `cmsh` with the "`power on`" command using ranges and other options (section 4.2.2 of the *Administrator Manual*).

2.4.1 IP Addresses In The Cluster-On-Demand Cloud

- The IP addresses assigned to cloud nodes on powering them up are arbitrarily scattered over the 10.0.0.0/8 network and its subnets
  - No pattern should therefore be relied upon in the addressing scheme of cloud nodes
- Shutting down and starting up head and regular cloud nodes can cause their IP address to change.
  - However, Bright Cluster Manager managing the nodes means that a regular cloud node re-establishes its connection to the cluster when it comes up, and will have the same node name as before.

3

Cluster Extension Cloudbursting

Cluster Extension cloudbursting ("hybrid" cloudbursting) in Bright Cluster Manager is the case when a cloud service provider is used to provide nodes that are in the cloud as an extension to the number of regular nodes in a cluster. The head node in a Cluster Extension configuration is always outside the cloud, and there may be some regular nodes that are outside the cloud too.

Requirements

Cluster Extension cloudbursting requires:

- **An activated cluster license.**

  Some administrators skip on ahead to try out cloudbursting right away in a Cluster Extension configuration, without having made the license active earlier on. That will not work.

  If activation is indeed needed, then it is most likely a case of simply running the `request-license` command with the product key. Further details on activating the license are in section 4 of the *Administrator Manual*.

- **Registration of the product key.**

  The product key must also be registered on the Bright Computing Customer Portal website at `http://www.brightcomputing.com/Customer-Login.php`. A Customer Portal account is needed to do this.

  The product key is submitted at the Customer Portal website specifically for a Cluster Extension setup, from the `Burst!` menu. The customer portal account is then automatically associated with the license installed (section 2.2) on the head node. The key is also needed to activate the cluster license, if that has not been done before.

- **An Amazon account**, if the cloud provider is Amazon.

- **An open UDP port.**

  By default, this is port 1194. It is used for the OpenVPN connection from the head node to the cloud and back. To use TCP, and/or ports other than 1194, the Bright Computing knowledgebase at `http://`

kb.brightcomputing.com can be consulted using the keywords "openvpn port".

Outbound ssh access from the head node is also useful, but not strictly required.

By default, Shorewall as provided by Bright Cluster Manager on the head node is configured to allow all outbound connections, but other firewalls may need to be considered too.

Steps

Cluster Extension cloudbursting uses a *cloud director*. A cloud director is a specially connected cloud node used to manage regular cloud nodes, and is described more thoroughly in section 3.2. Assuming the administrator has ownership of a cloud provider account, the following steps can be followed to launch Cluster Extension cloud nodes:

1. The cloud provider is logged into from cmgui, and a cloud director is configured (section 3.1).
2. The cloud director is started up (section 3.2).
3. The cloud nodes are provisioned from the cloud director (section 3.3).

The cloud nodes then become available for general use by the cluster.

Cluster Extension Cloudbursting With A Hardware VPN

Bright Cluster Manager recommends, and provides, OpenVPN by default for Cluster Extension cloudbursting VPN connectivity. If there is a wish to use a hardware VPN, for example if there is an existing hardware VPN network already in use at the deployment site, then Bright Cluster Manager can optionally be configured to work with the hardware VPN. The configuration details can be found in the Bright Computing knowledgebase at http://kb.brightcomputing.com by carrying out a search on the site using the keywords "cloudbursting without openvpn".

3.1 Cluster Extension: Cloud Provider Login And Cloud Director Configuration

To access the Amazon cloud service from cmgui, the "Cloud Nodes" resource is selected, and the "Cloud Accounts" tabbed pane opened. This allows a cloud provider account to be edited or deleted from the available ones already there.

It also allows a new cloud account provider to be added and configured. This is done by clicking on the ⊞ button beside the text "Add a new cloud account", which opens up the "Add Cloud Provider Wizard" window (figure 3.1).

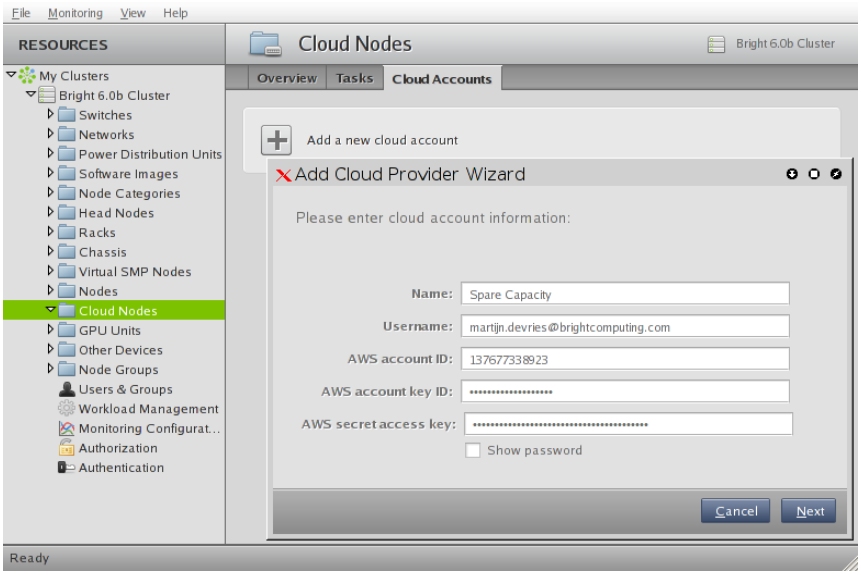

Figure 3.1: Logging Into A Cloud Provider With `cmgui`

In the first screen, the cloud account subscription information is added. The subscription information could be from Amazon or from another supported provider.

In the case of Amazon, the information is obtainable after signing up for Amazon Web Services (AWS) at `http://aws.amazon.com`. After sign-up, the Access Identifiers section of the subscription, at `http://aws-portal.amazon.com/gp/aws/developer/account/index.html?action=access-key`, provides the required information. If that URL does not work, then the Amazon documentation at `http://docs.amazonwebservices.com/fws/latest/GettingStartedGuide/index.html?AWSCredentials.html` can be followed instead.

For Amazon, the fields to enter in the wizard are:

- The `Name`: A convenient, arbitrary value.
- The `Username`: The e-mail address associated with the AWS account.
- The `AWS account ID`: The AWS Account ID.
- The `AWS account key ID`: The AWS Access Key ID.
- The `AWS secret access key ID`: The AWS Secret Access Key.

The "`show password`" checkbox toggles the visibility of the sensitive input. Clicking the `Next` button submits the details, and inputs for the next screen are retrieved from Amazon.

The next screen (figure 3.2) displays options for the Amazon cloud service.

Figure 3.2: Selecting Options At The Cloud Provider With `cmgui`

In figure 3.2, the following options are shown:

- `Default region`: These are regions from which the service can be provided. Amazon, for example, offers a choice out of capacity on the East Coast of the USA, Western Europe, the Asia Pacific region and others.
- `Default AMI`: This is the Amazon Machine Instance image that Bright Computing provides. The node-installer from this image installs the cloud director and cloud nodes.
- `Default type`: A choice out of a selection of possible virtual machine types (`http://aws.amazon.com/ec2/instance-types/`) made available by Amazon for the cloud node. The choices presented are from node specifications consisting of memory, storage, cores, GPUs, and others. In `cmsh`, running `cmsh -c "cloud types"` also shows the types available.
- `Default director type`: A choice for the cloud director node, made from a selection of possible virtual machine types made available by Amazon. This virtual machine type usually needs to be more powerful than a regular cloud node, and is by default set to `m1.large`.

The default settings are normally a good choice. On clicking the `Next` button, the choices are processed.

The next screen (figure 3.3) displays the NetMap network name and addressing scheme. This is a network mapping that assigns extra IP addresses to local nodes to make them accessible from the cloud. The addressing scheme can be changed if needed to another unused subnet. By default it uses 172.31.0.0/16.

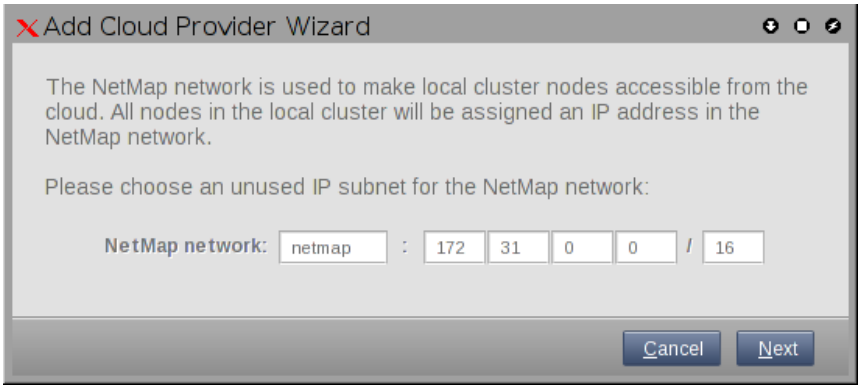

Figure 3.3: Setting The NetMap Network With `cmgui`

The default values are normally a good choice. On clicking the `Next` button, the values are processed.

The next screen (figure 3.4) displays the cloud network name and addressing scheme. This can be changed if needed, but for Amazon the 10.0.0.0/8 range is expected.

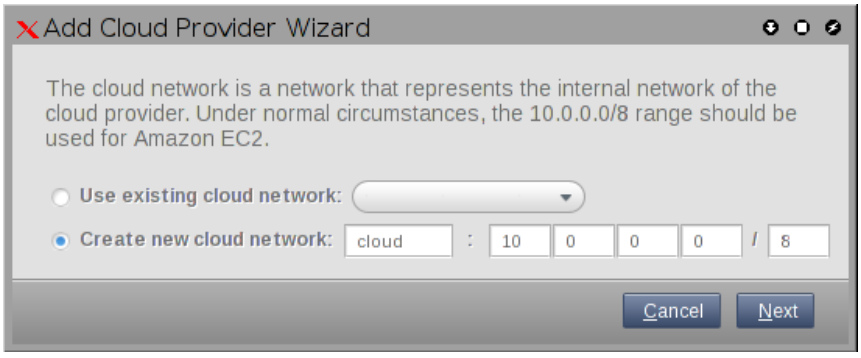

Figure 3.4: Setting The Cloud Network At The Cloud Provider With `cmgui`

On clicking the `Next` button, the configuration is processed.

The next screen (figure 3.5) displays a proposed Bright Cluster Manager tunnel network naming and addressing scheme for each checkboxed cloud region. These can be changed if needed from the suggested defaults. For Amazon the `us-east-1` region shown in the figure has a default tunnel network value of 172.21.0.0/16.

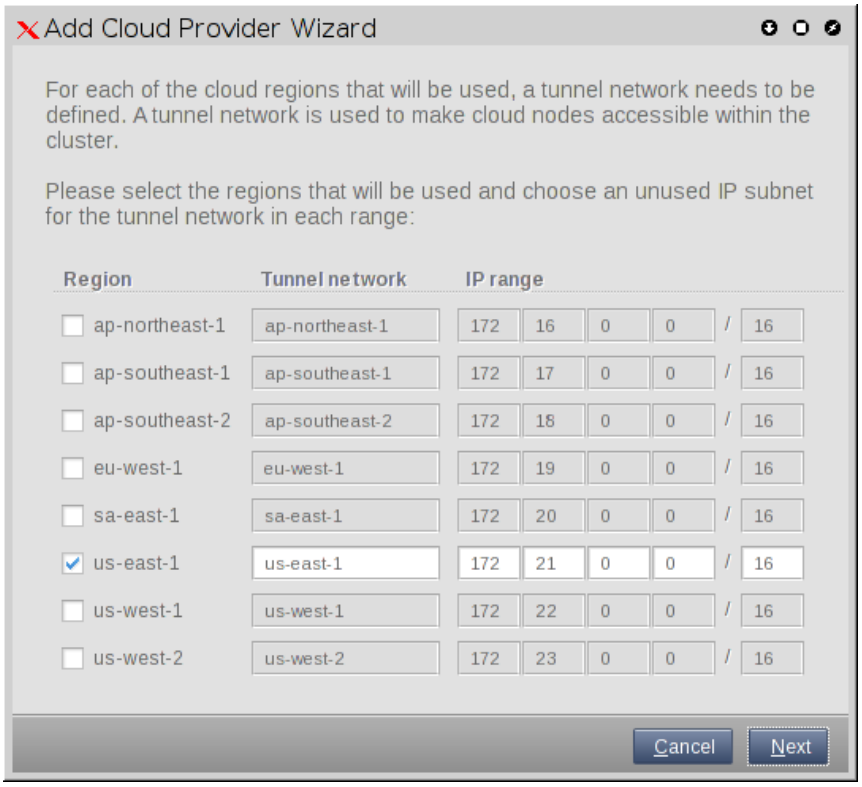

Figure 3.5: Setting The Tunnel Network For Regions With `cmgui`

On clicking the `Next` button, the configuration is processed.

The next screen (figure 3.6) displays a proposed Bright Cluster Manager tunnel interface name and IP address for the head node(s). A tunnel interface is defined for the head node for each tunnel network. By default, the address ending in .255.254 is used, and appended to the first two parts of the dotted quad (for example, 172.21 for us-east-1), so that the suggested default IP address in this case becomes 172.21.255.254. The default suggested device name is tun0.

These can be changed if needed from the suggested defaults.

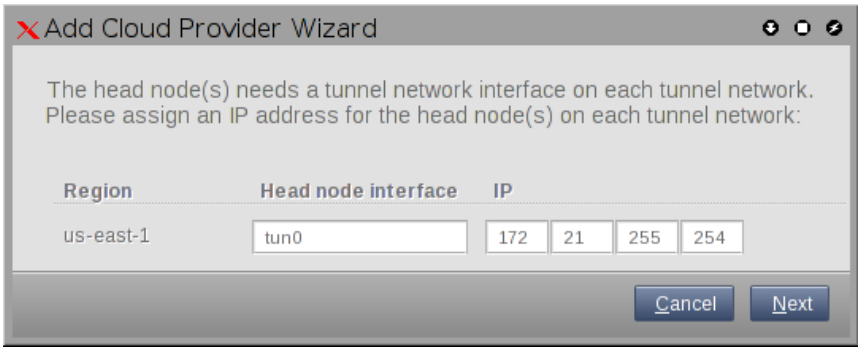

Figure 3.6: Setting The Tunnel Network Interface For The Head Node(s) With `cmgui`

On clicking the `Next` button, the configuration is processed.

The next screen (figure 3.7) displays a proposed Bright Cluster Manager hostname and tunnel IP address for the cloud director node(s). By

default, the suggested hostname is the region name with `-director` as the suffix. For example, `us-east1-director` for the region `us-east1`. By default, an address ending in .255.251 is suggested for appending to the first two parts of the dotted quad (for example, the prefix 172.21 for us-east-1), so that the suggested default IP address in this case becomes 172.21.255.251. The addresses ending in 252 and 253 may be required by head nodes that implement failover (Chapter 12 of the *Administrator Manual*).

These can be changed if needed from the suggested defaults, but should be consistent with the network address.

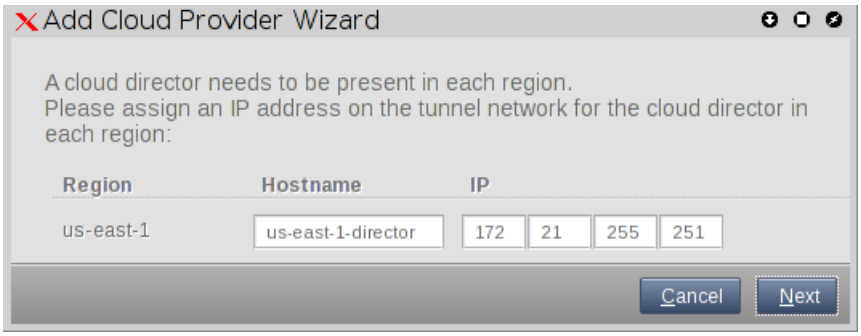

Figure 3.7: Setting The Tunnel Network Interface For The Cloud Director(s) With `cmgui`

On clicking the `Next` button, the configuration is processed.

The next screen (figure 3.8) displays the proposed assignment of IP addresses in the NetMap network. These can be changed from the suggested defaults, but should be consistent with the addressing schemes already defined.

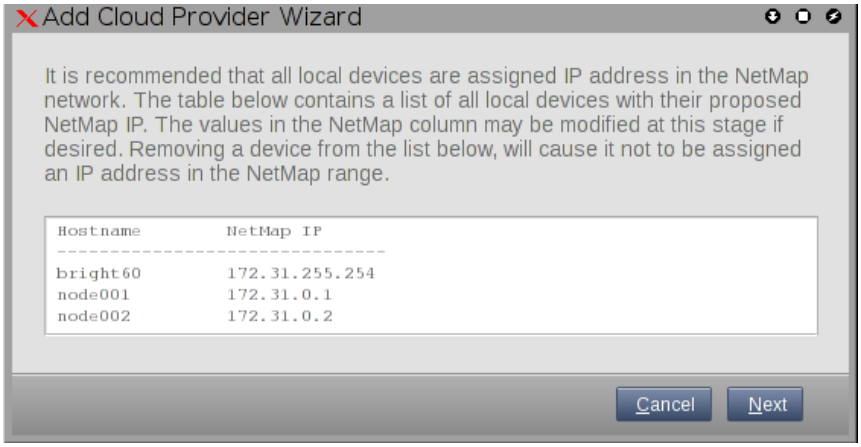

Figure 3.8: Setting The Tunnel Network Interface IP Addresses With `cmgui`

On clicking the `Next` button, the configuration is processed.

It should be noted that the default suggested NetMap, cloud network, and cloud region addresses configured by the wizard are all compliant with RFC1918 private network addressing, and are thus not public IP addresses.

If all is well, the configuration is successfully processed. A message is

then displayed indicating that the cloud provider service has been added to the existing cluster and configured successfully, and that the wizard is finished with its job.

No nodes are activated yet within the cloud provider service. To start them up, the components of the cloud provider service must be started up by

- powering up the cloud directors (section 3.2)
- powering on the cloud nodes after the cloud directors are up. Often this involves creating new cloud nodes by using the "`Create Cloud Nodes`" wizard (section 3.3).

3.2 Cluster Extension: Cloud Director Start-up

The cloud director can take some time to start up the first time. The bottleneck is usually due to several provisioning stages, where the bandwidth between the head node and the cloud director means that the provisioning runs typically take tens of minutes to complete. The progress of the cloud director can be followed in the event viewer (section 9.6 of the *Administrator Manual*).

This bottleneck is one of the reasons why the cloud director is put in the cloud in the first place. The next time the cloud director powers up, and assuming persistent storage is used—as is the default—the cloud director runs through the provisioning stages much faster, and completes within a few minutes.

The cloud director acts as a helper instance in the cloud, providing some of the functions of the head node within the cloud in order to speed up communications and ensure greater resource efficiency. Amongst the functions it provides are:

- Cloud nodes provisioning
- Exporting a copy of the shared directory `/cm/shared` to the cloud nodes so that they can mount it
- Providing routing services using an OpenVPN server. While cloud nodes within a region communicate directly with each other, cloud nodes in one region use the OpenVPN server of their cloud director to communicate with the other cloud regions and to communicate with the head node of the cluster.

Cloud directors are not regular nodes, so they have their own category, `cloud-director`, into which they are placed by default.

The cloud-related properties of the cloud director are displayed in the "`Cloud Settings`" tab of the Cloud Nodes director item.

The cloud director can be started up in `cmgui` by right-clicking on the cloud director item from the `Cloud Nodes` resource, and selecting `Power on` from the menu. Any cloud settings options that have been set are frozen as the instance starts up, until the instance terminates.

3.2.1 Setting The Cloud Director Disk Storage Device Type

Amazon provides two kinds of storage types as part of EC2:

1. **Instance storage, using ephemeral devices:** Instance storage is not provided for the following instance types:

   - t1.micro
   - m3.xlarge
   - m3.2xlarge
   - cr1.8xlarge

   However, Amazon by default currently provides 150GB or more of instance storage for all other instance types.

   Details on instance storage can be found at http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#StorageOnInstanceTypes. Ephemeral means that the device is temporary, which means that whatever is placed on it is lost on reboot.

2. **Elastic Block Storage (EBS) volumes:** Normally, EBS is suggested for cloud director and cloud node use. The reasons for this include:

   - it can be provided to all nodes in the same availability zone
   - unlike instance storage, EBS remains available for use when an instance using it reboots
   - instance storage is not available for some instances types such as t1.micro.

Using the ephemeral device as the drive for the cloud director:
Since the cloud provider instance type is essential, and contains so much data, it is rarely wise to use ephemeral devices as the drive for the cloud provider.

However, if for some reason the administrator would like to avoid using EBS, and use the instance storage, then this can be done by removing the default EBS volume suggestion for the cloud director provided by Bright Cluster Manager. When doing this, the ephemeral device that is used as the replacement must be renamed. It must take over the name that the EBS volume device had before it was removed.

- In cmgui, this can be done in the "Cloud Settings" tab of the Cloud Nodes director item.
- In cmsh, this can be done in device mode, by going into the cloudsettings submode for the cloud director, and then going a level deeper into the storage submode. Within the storage submode, the list command shows the values of the storage devices associated with the cloud director. The values can be modified as required with the usual object commands. The set command can be used to modify the values.

  Example

```
[bright70]% device use us-east-1-director
[bright70->device[us-east-1-director]]% cloudsettings
[bright70->device[us-east-1-director]->cloudsettings]% storage
[bright70->...->cloudsettings->storage]% list
```

```
Type       Name (key)   Drive     Size    Volume ID
---------- ------------ --------- ------- ---------
ebs        ebs          sdb       42GB
ephemeral  ephemeral0   sdb       0B
[bright70->...->cloudsettings->storage]% remove ebs
[bright70->...->cloudsettings*->storage*]% set ephemeral0 drive sdb
[bright70->...->cloudsettings*->storage*]% list
Type       Name (key)   Drive     Size    Volume ID
---------- ------------ --------- ------- ---------
ephemeral  ephemeral0   sdb       0B
[bright70->...->cloudsettings*->storage*]% commit
```

3.2.2 Setting The Cloud Director Disk Size

The disk size for the cloud director can be set with `cmgui` in the `Cloud Settings` tab.

By default, an EBS volume size of 42GB is suggested. This is as for a standard node layout (section D.3 of the *Administrator Manual*), and no use is then made of the ephemeral device.

42GB on its own is unlikely to be enough for most purposes other than running basic `hello world` tests. In actual use, the most important considerations are likely to be that the cloud director should have enough space for:

- the user home directories (under `/home/`)
- the cluster manager shared directory contents, (under `/cm/shared/`)
- the software image directories (under `/cm/images/`)

The cluster administrator should therefore properly consider the allocation of space, and decide if the disk layout should be modified. An example of how to access the disk setup XML file to modify the disk layout is given in section 3.9.3 of the *Administrator Manual*.

For the cloud director, an additional sensible option may be to place `/tmp` and the swap space on an ephemeral device, by appropriately modifying the XML layout for the cloud director.

3.2.3 Tracking Cloud Director Start-up

Tracking cloud director start-up from the EC2 management console:
the boot progress of the cloud director can be followed by watching the status of the instance in the Amazon EC2 management console, as illustrated in figure 2.8. The `Instance ID` that is used to identify the instance can be found

- with `cmgui`, within the `Cloud Settings` tab for the cloud director node
- with `cmsh`, by running something like:

  Example

```
[bright70]% device cloudsettings us-east-1-director
[bright70->device[us-east-1-director]]% get instanceid
```

Tracking cloud director start-up from `cmgui`:
the boot progress of the cloud director can also be followed by

- watching the icon changes (as in section 5.5.1 of the *Administrator Manual*)
- watching the `State` in the `Overview` tabbed window
- watching the `Console log` from the `Tasks` tabbed window

Tracking cloud director start-up from the bash shell of the head node:
there are some further possibilities to view the progress of the cloud director after it has reached at least the initrd stage. These possibilities include:

- an SSH connection to the cloud director can be made during the pre-init, initrd stage, after the cloud director system has been set up via an rsync. This allows a login to the node-installer shell.
- an SSH connection to the cloud director can be also be made after the initrd stage has ended, after the init process runs making an SSH daemon available again. This allows a login on the cloud director when it is fully up.

During the initrd stage, the cloud director is provisioned first. The cloud node image(s) and shared directory are then provisioned on the cloud director, still within the initrd stage. To see what rsync is supplying to the cloud director, the command "`ps uww -C rsync`" can be run on the head node. Its output can then be parsed to make obvious the source and target directories currently being transferred:

Example

```
[root@bright70 ~]# ps uww -C rsync | cut -f11- -d" " #11th part onwards
/cm/shared/ syncer@172.22.255.251::target//cm/shared/
```

Tracking cloud director start-up from `cmsh`:
the `provisioningstatus` command in `cmsh` can be used to view the provisioning status (some output elided):

Example

```
[root@bright70 ~]# cmsh -c "softwareimage provisioningstatus"
...
+ us-east-1-director
...
  Up to date images:        none
  Out of date images:       default-image
```

In the preceding output, the absence of an entry for "`Up to date images`" shows that the cloud director does not yet have an image that it can provision to the cloud nodes. After some time, the last few lines of output should change to something like:

Example

```
+ us-east-1-director
...
  Up to date images:        default-image
```

This indicates the image for the cloud nodes is now ready.

With the -a option, the provisioningstatus -a command gives details that may be helpful. For example, while the cloud director is having the default software image placed on it for provisioning purposes, the source and destination paths are /cm/images/default-image:

Example

```
[root@bright70 ~]# cmsh -c "softwareimage provisioningstatus -a"
Request ID(s):        4
Source node:          bright70
Source path:          /cm/images/default-image
Destination node:     us-east-1-director
Destination path:     /cm/images/default-image
...
```

After some time, when the shared filesystem is being provisioned, the source and destination paths should change to the /cm/shared directory:

```
[root@bright70 ~]# cmsh -c "softwareimage provisioningstatus -a"
Request ID(s):        5
Source node:          bright70
Source path:          /cm/shared
Destination node:     us-east-1-director
Destination path:     /cm/shared
...
```

After the shared directory and the cloud node software images are provisioned, the cloud director is fully up. Cloud node instances can then be powered up and provisioned from the cloud director.

3.3 Cluster Extension: Cloud Node Start-up

The "Create Cloud Nodes" wizard button in cmgui conveniently creates cloud node objects. The wizard is accessed from within the "Cloud Nodes" resource, by selecting the provider item, and then choosing the Overview tab. Cloud node objects can also be created in cmsh as described in section 4.3.

A working cloud director is not needed to configure the regular cloud nodes. However the cloud director must be up, and the associated networks to the regular cloud nodes and to the head node must be configured correctly, in order for the regular cloud nodes to boot up properly. If needed, additional cloud provisioning nodes (section 5.2 of the *Administrator Manual*) can be configured by assigning the provisioning role to cloud nodes, along with appropriate nodegroups (page 156 of the *Administrator Manual*) values, in order to create a provisioning hierarchy.

By default, the first screen of the wizard (figure 3.9) allows the administrator to do the following:

- The first regular cloud node and last regular cloud node can be set. By default, 16 regular cloud nodes are suggested. The names of the nodes have a prefix of cnode by default, and end in three digit numbers, for example cnode001, cnode002 and so on.

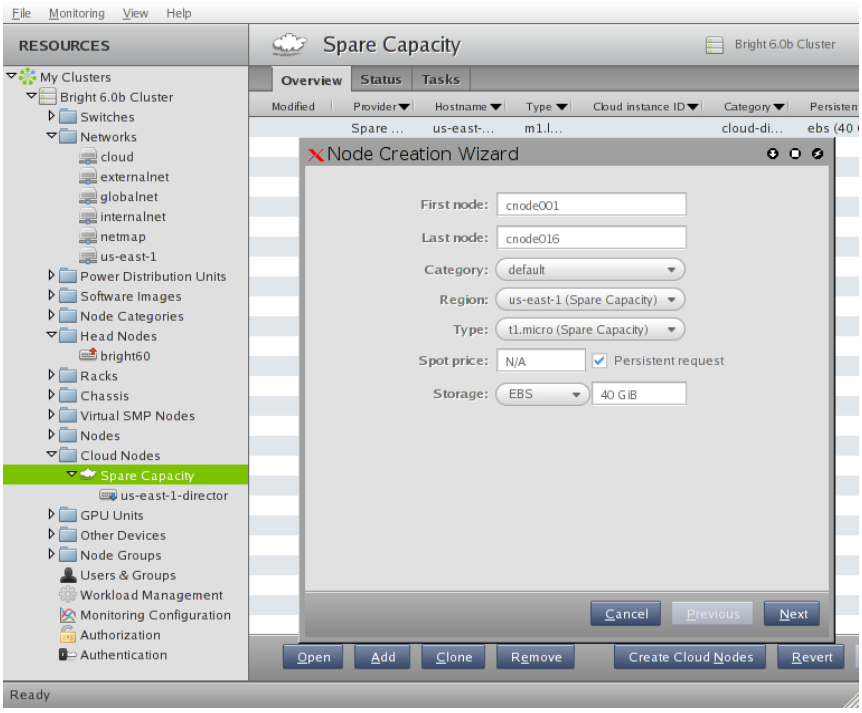

Figure 3.9: Main Cloud Node Creation Wizard Configuration Screen

- The category can be set for these nodes. By default it is set to the `default` category
- The region for the regular cloud nodes can be set. By default it matches the cloud director region.
- The regular cloud node instance type can be set. By default, `t1.micro` is chosen.
- A spot price (section 5.3.1) can be set in this screen to take advantage of cheaper pricing to launch regular cloud nodes. By default, no spot price is set.
- The storage type and size used can be set. By default, it is EBS, and 42GB. If the `t1.micro` instance type has been chosen, then there is no ephemeral device storage available, in accordance with Amazon policies.

The next screen of the wizard (figure 3.10) applies to the region chosen in the previous screen (in figure 3.9 the region is `us-east-1`). Within the region, IP offsets (footnote on page 28 of the *Installation Manual*) can be set:

- for nodes in the associated cloud network
- for nodes in the associated tunnel network

By default, both these IP offsets are `0.0.0.0`.

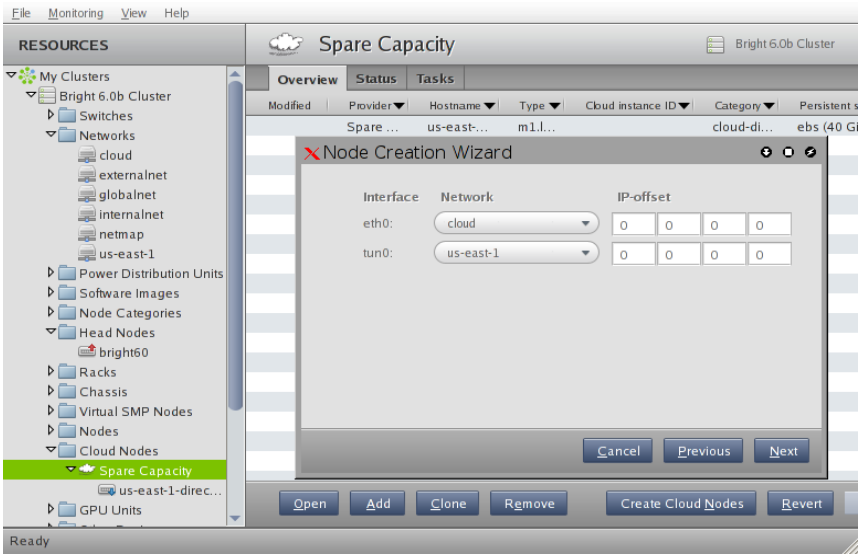

Figure 3.10: Cloud Node Wizard Network And IP Offset Configuration Screen

The last screen of the wizard (figure 3.11) shows a summary screen of the proposed IP address allocations. If the cloud IP addresses are to be assigned using DHCP, then their values are 0.0.0.0.

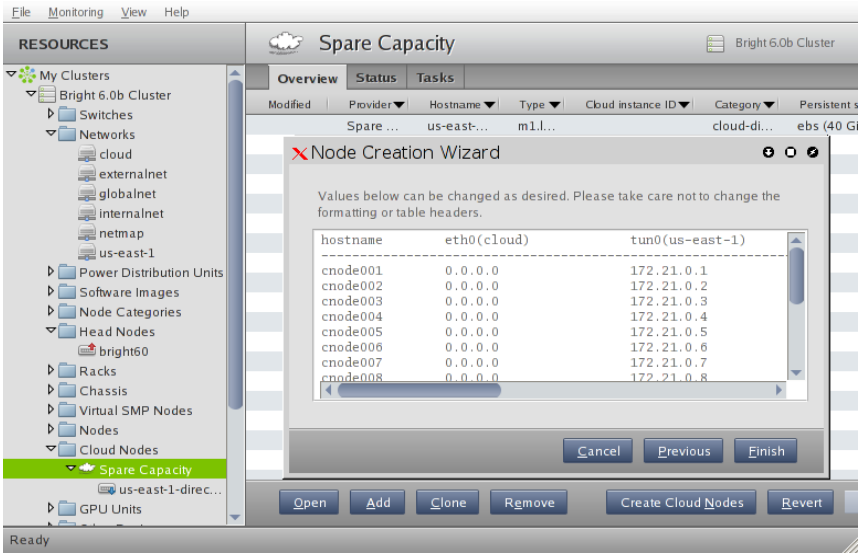

Figure 3.11: Cloud Node Wizard Network And IP Layout Screen

When the wizard is finished, the regular cloud nodes must be saved. This adds them to the default category by default.

If the cloud director is up, then the cloud nodes can be booted up by powering them up (section 4.2 of the *Administrator Manual*) by category, or individually.

4

Cloudbursting Using The Command Line And cmsh

The command line and cmsh can be used to set up Cluster-On-Demand and Cluster Extension clusters.

For Cluster-On-Demand setups, a GUI web browser is needed initially to launch the head node AMI from Amazon. However, once an ssh connection is made by the administrator to the head node, cloudbursting can be continued from command line. Thus, the bright-setup script is run from the command line as in section 2.2, and the cloud nodes can be powered up from the command line as in section 2.4.

For Cluster Extension setups, cloudbursting can be carried out entirely from the command line. The cm-cloud-setup script (section 4.1) sets up the cloud provider login and cloud director configuration GUI steps of section 3.1 in a guided manner on a command line dialog, after which cmsh power commands can launch the required cloud nodes (sections 4.2 and 4.3).

4.1 The cm-cloud-setup Script

The cm-cloud-setup script is run on the head node, and allows the administrator to specify settings to launch a cloud using the command line. The help text for this utility shows:

```
USAGE: /cm/local/apps/cluster-tools/bin/cm-cloud-setup <OPTIONS>

OPTIONS:
---------
  -p | --provider <name>       Provider name (default: amazon)
  -n | --name <name>           Account name (default: Amazon EC2)
  -u | --username <username>   Username used in Amazon AWS console
  -w | --password <password>   Password used in Amazon AWS console (if
                               omitted you will be prompted, unless
                               account keys are provided)
  -a | --accountId             Amazon account ID
  -k | --awskey <key>          AWS access key ID
  -s | --secretkey <key>       AWS secret access ID

  -d | --delete                Delete Amazon account if it exists
  -y | --yes                   Use all
```

```
  -h | --help                  Print this help

NOTES:
------
  --password option does not work yet
```

It can be used as follows from the prompt:

Example

```
[root@bright70 ~]# cm-cloud-setup -u rotwang@example.com -a 123923792991 \
-k OIQQOWU9LJJEI1232PJP -s ra9xaG7oUiy1uqu0ahW4aixuThee5ahmUGoh9cha
```

The administrator is then guided through the steps needed to launch the cloud. The session output may show up as something like (some text elided):

Example

```
Connecting to cluster
Waiting for data from cluster
Adding cloud provider Amazon EC2 ... ok
Waiting for cloud provider data ...
Got 7 regions, 29 images, 12 types
Default region (default: eu-west-1), options:
      ap-northeast-1,
      ...
      us-west-2
> eu-west-1
Default AMI (default: latest), options
      ...
      brightinstaller-074,
      brightinstaller-075
>
Default type (default: t1.micro), options
      c1.xlarge,
      ...
      t1.micro
>
Default cloud director type (default: m1.large), options
      c1.xlarge,
      ...
      t1.micro
>
Update cloud provider Amazon EC2... ok
Got 6 networks
Found tunnel network for eu-west-1: 172.16.0.0/16
Using NetMap network: 172.31.0.0/16
Using cloud network: 10.0.0.0/8
Use regions: (default eu-west-1, space separated / all), options:
      ap-northeast-1,
      ...
      us-west-2
>
Updating head node bright70 ... ok
Updating tunnel network eu-west-1 ... ok
```

```
Cloud director ip on eu-west-1 (default 172.16.255.251)
>
Adding cloud director eu-west-1-director ... ok
Provisioning update started
[root@bright70 ~]#
```

After `cm-cloud-setup` has run, the cloud nodes (the cloud director and regular cloud nodes) can be launched.

4.2 Launching The Cloud Director

Launching the cloud requires that the cloud director and cloud nodes be powered up. This can be done using `cmgui` as described in sections 3.2 and 3.3. It can also be carried out in `cmsh`, for example, the cloud director `eu-west-1-director` can be powered up from device mode with:

Example

```
cmsh -c "device power on -n eu-west-1-director"
```

If the administrator is unsure of the exact cloud director name, one way it can easily be found is via tab-completion within the `device` mode of `cmsh`.

As explained in section 3.2, the cloud director takes some time to power up. Its status can be followed in the notice messages sent to the `cmsh` session, or in the `cmgui` event viewer. The status can also be queried via the `status` command in device node. For example, a `watch` instruction such as:

```
[root@bright70 ~]# watch 'cmsh -c "device status -n eu-west-1-director"
```

will show a series of outputs similar to:

```
eu-west-1-director ....... [ PENDING ] (Waiting for instance to start)
eu-west-1-director ....... [ PENDING ] (Waiting for instance to start)
eu-west-1-director ....... [ PENDING ] (IP assigned: 54.220.240.166)
eu-west-1-director ....... [ PENDING ] (setting up tunnel)
eu-west-1-director ....... [ INSTALLER_REBOOTING ]
eu-west-1-director ....... [ INSTALLING  ] (recreating partitions)
eu-west-1-director ....... [ INSTALLING  ] (FULL provisioning to "/")
eu-west-1-director ....... [ INSTALLING  ] (provisioning started)
...
```

4.3 Launching The Cloud Nodes

Once the cloud director is up, the cloud nodes can be powered up. This first requires that the cloud node objects exist and each have an IP address assigned to them that is consistent with that of the cloud director that manages them. With `cmgui`, this can be done with the help of a wizard to assign the IP addresses (section 3.3). With `cmsh`, assignment can be done for an individual cloud node, or for many cloud nodes, as follows:

4.3.1 Creating And Powering Up An Individual Node

In the example that follows, a single cloud node is assigned a management network, a tunnel IP address, and a tunnel network so that it can communicate with the cloud director.

Example

```
[root@bright70 ~]# cmsh
[bright70]% device
[bright70->device]% add cloudnode cnode001
Warning: tunnel ip of cnode001 not set. This CloudNode will not start!
[bright70->device*[cnode001*]]% set managementnetwork eu-west-1
[bright70->device*[cnode001*]]% show
Parameter              Value
---------------------- --------------------------
...
Management network     eu-west-1
Network                eu-west-1
...
[bright70->device*[cnode001*]]% interfaces
[bright70->device*[cnode001*]->interfaces]% list
Type      Network device name   IP          Network
--------- --------------------- ----------- ----------------
physical  eth0 [prov,dhcp]      0.0.0.0     cloud-ec2classic
tunnel    tun0                  0.0.0.0
[bright70->device*[cnode001*]->interfaces]% set tun0 ip 172.16.0.1
[bright70->device*[cnode001*]->interfaces*]% set tun0 network eu-west-1
[bright70->device*[cnode001*]->interfaces*]% list
Type      Network device name   IP          Network
--------- --------------------- ----------- ----------------
physical  eth0 [prov,dhcp]      0.0.0.0     cloud-ec2classic
tunnel    tun0                  172.16.0.1  eu-west-1
[bright70->device*[cnode001*]->interfaces*]% commit
```

The preceding session completes the cloud node object configuration. The cloud node itself can now be launched with an explicit power command such as:

```
[bright70->device[cnode001]->interfaces]% device power on -n cnode001
```

4.3.2 Creating And Powering Up Many Nodes

For a large number of cloud nodes, the creation and assignment of IP addresses can be done with the clone option of the `foreach` command, (section 2.5.5 of the *Administrator Manual*), together with a node range specification. This is the same syntax as used to create non-cloud regular nodes with `cmsh`. Continuing on with the preceding session, where a node `cnode001` was configured:

```
[bright70->device]% foreach --clone cnode001 -n cnode002..cnode010 ()
The IP of network interface: eth0 was not updated
Warning: The Ethernet switch settings were not cloned, and have to be se\
t manually
...
[bright70->device*]% commit
Mon Apr 23 04:19:41 2012 [alert] cnode002: Check 'DeviceIsUp' is in stat\
e FAIL on cnode002
[bright70->device]%
Mon Apr 23 04:19:41 2012 [alert] cnode003: Check 'DeviceIsUp' is in stat\
e FAIL on cnode003
...
Successfully committed 9 Devices
```

```
[bright70->device]%
```

The IP addresses are assigned via heuristics based on the value of cnode001 and its cloud director.

As before, an alert warns each cloud node is down. The list of cloud nodes can be powered up using cmsh with the node range option:

Example

```
[bright70->device]% foreach -n cnode002..cnode010 (power on)
```

4.4 Submitting Jobs With cmsub

The cmsub command is a user command wrapper to submit job scripts to a workload manager in a Cluster Extension cluster, so that jobs are considered for running in the cloud. Its usage for an end user is covered in section 4.7 of the *User Manual*.

The cmsub command is available from the Bright Cluster Manager repository as part of the cmsub package. The package is installed by default on the head node.

The cmsub command requires that an environment module (section 2.2 of the *Administrator Manual*) called cmsub is loaded before it can be used. The cmsub environment module is not loaded by default on the head node.

When the cmsub command is run by the user to submit a job, the job is submitted to the workload manager, and the data-aware scheduling mechanism is initiated. A cluster with *data-aware scheduling* is a cluster that ensures that it has the data needed for the cloud computing job already in the cloud before the job is executed in the cloud.

4.4.1 Installation And Configuration of cmsub For Data-aware Scheduling To The Cloud

The configuration of data-aware scheduling means configuring the cluster so that the tools that allow data-aware scheduling to work correctly are configured. The configuration that is carried out depends on the workload manager that is to be used.

If cmsub has not yet been set up, or if it needs reconfiguration, then the following steps should be carried out:

1. The cmsub package is installed, if needed, on the head node and in the software image used for compute cloud nodes. The following dependencies are installed automatically:
   - cmdaemon-pythonbinding
   - python-boto
   - cm-cloud-copy
2. The cmsub-setup utility is run on the head node:

```
$ module load cmsub
$ cmsub-setup
```
a series of questions appears that need answering

3. The instructions that are displayed at the end of the cmsub-setup execution should then be followed. The instructions include the following:

   - Cloud queues that have been specified in the previous step must be assigned to the appropriate cloud nodes.
   - For workload managers that are not Slurm:
     - The software image that is to be used for cloud compute nodes must be checked for having home directories and having the correct permissions for these directories. Only home directories for users that will submit cloud jobs need to exist.
   - The cloud nodes must be provisioned.

One of the changes that cmsub-setup carries out is to create, or update, /cm/local/apps/cmsub/etc/cmsub.conf. The file does not normally need changing manually, but it can, for example, be used to turn debug messages on and off, or be used to change the name of the current cloud provider.

The name of the so-called cloud transfer queue is specified when cmsub-setup is run. The transfer of data to and from the cloud is carried out by jobs that are submitted to the cloud transfer queue, and the default properties of the queue depend on the workload manager that is used. The cluster administrator may therefore wish to tune the queue in the workload manager so that data transfer is optimized according to the characteristic requirements of the submitted jobs.

The cmsub-setup configuration utility can be used to set up cmsub for a particular workload manager, cloud queue, and software image. If another workload manger, cloud queue or software image needs to be used, then cmsub-setup should be executed again.

The cmsub-setup configuration utility retains the values in cmsub.conf, except for the following parameters which are set during cmsub-setup execution:

- TRANSFER_QUEUE
- CLOUD_PROVIDER
- CLOUD_REGION

The cmsub-setup actions are logged in /var/log/cmsub/cmsub-setup.log

4.4.2 How Data-aware Scheduling To The Cloud Works

Data-aware scheduling logic is described in this section. Figure 4.1 shows the flow of data transfer that takes place during a cmsub run.

The numbers on the "data flow" arrows in the figure indicate the data flow steps. The steps are, in sequence:

Step 1: The user creates a job script and submits it to a workload manager using the cmsub utility. All files required by a job should be readable by the job user on the transfer node.

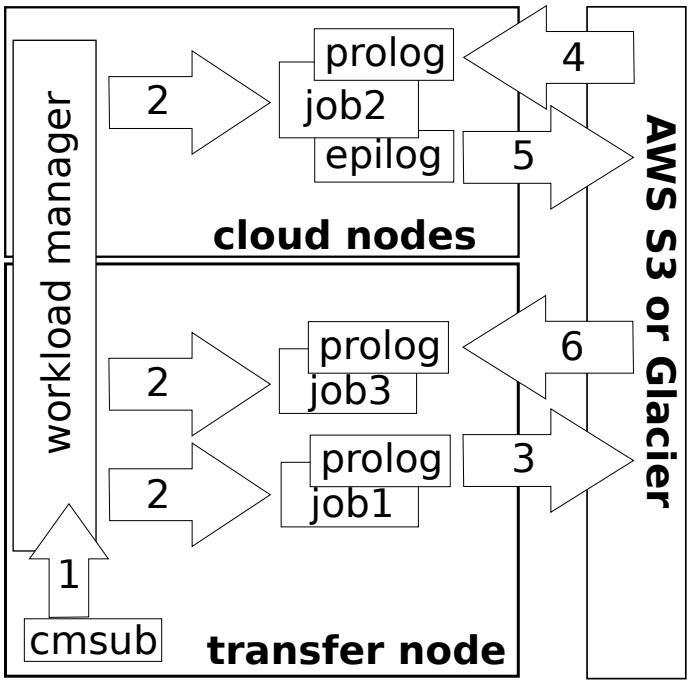

Figure 4.1: Flow Sequence Of Data-aware Scheduling

Step 2: The workload manager starts three jobs (`job1`, `job2`, and `job3` in figure 4.1), as requested by the `cmsub` utility.

- The first job (`job1`) uploads user data to AWS S3 or Glacier storage. The `cmsub` utility also creates a job description file in user home directory with a name `.cmsub-`<*jobid*> that is uploaded to S3 storage (never Glacier). The number <*jobid*> is the ID of `job1`. The `cmsub` utility puts information in the description file that is required by the prolog and epilog scripts running in the cloud.
- The second job (`job2`) is run only after the first job is finished, and is the user job that is submitted to the workload manager with `cmsub`. The job is specified by the last argument of the `cmsub` command that the user runs.
- The third job (`job3`) downloads the data produced by `job2` from the cloud back to the transfer host. This job starts only after the second job is finished.

Step 3: The Prolog script (`/cm/local/apps/cmsub/scripts/prolog-cmsub`) of the first job creates a new container: a bucket (in the case of S3) or a vault (in the case of Glacier), with the name:

cmsub-<*user*>-<*jobid*>-<*account*>
where

- <*user*> is the job owner name
- <*jobid*> is the ID (2) of the user job (`job2` in Step 2)
- <*account*> is the account number that is configured for the cloud provider, which is AWS here.

When the container is created, the prolog script uploads user data to that container.

Step 4: The prolog script (`/cm/local/apps/cmsub/scripts/prolog-cmsub`) of the second job now downloads the description file from the S3 bucket. Based on the description file content, the prolog script downloads data from S3 or Glacier to the main job node, typically to the home director of the user.

Step 5: When the second job has finished, its epilog script (`/cm/local/apps/cmsub/scripts/epilog-cmsub`) is started. The script uploads what has been produced by the `job2` files and directories to S3 or to Glacier, based on the description file that was downloaded by the prolog script earlier during Step 4. Any STDOUT and STDERR files created by the workload manager for the job are also uploaded.

The epilog script normally removes all the data downloaded during Step 4 and produced by `job2`. The user can specify the `--keep-data` option for `cmsub` to keep the data.

Step 6: The prolog script (`/cm/local/apps/cmsub/scripts/prolog-cmsub`) of the third job downloads data produced by `job2` to the transfer host. On finishing it removes the description file from the user directory and removes the container from AWS storage, unless the `--keep-data` option has been used for `cmsub`.

Data transfer is carried out with prolog and epilog scripts for `job1` and `job2`, instead of using job scripts. This is because prolog and epilog scripts can be started with root permissions, and can therefore connect to CMDaemon. Connecting to CMDaemon allows information about the cloud provider and jobs to be obtained, which is needed for the `cmsub` run to be carried out. In order to connect to CMDaemon, the scripts use key files generated by `cmsub-setup`. The key files are located by default in `/cm/local/apps/cmsub/etc/` and are copied to the cloud compute node software image specified during `cmsub-setup`. The prolog and epilog scripts that run in the cloud are then able to use the keys. After having accessed the keys, the prolog and epilog scripts change their own UID/GID and EUID/EGID execution bits to that of the job owner, so that all file operations are performed only with the permissions of the job owner.

The prolog and epilog scripts log all their operations on the nodes where they are executed, to the following files:

- `/var/log/cmsub/<`*username*`>/<`*jobid*`>-prolog`
- `/var/log/cmsub/<`*username*`>/<`*jobid*`>-epilog`

In the file paths, <*username*> is the job owner name, and <*jobid*> is the ID of a job that "owns" the prolog or epilog script

Thus the prolog from Step 3 creates a log file with the ID of the first job, while prolog and epilog from Steps 4 and 5 use the ID of the second job, and the prolog from Step 6 uses the ID of the third job, when setting their log file names.

By transfer node, a node is meant where `job1` and `job2` are started. Usually this is a head node, but it can also be any other node, such as a login node. The transfer node must have access to AWS services in order to allow prolog scripts of `job1` and `job3` to upload and download data to and from S3 or Glacier. The hostname of the transfer node is asked for during the `cmsub-setup` run.

The `cm-cloud-copy` tool (section 4.5.1) is used in the backend during the transfer of data to and from AWS storage.

4.4.3 Troubleshooting `cmsub` Problems

SGE Issues

- A second job, that is the user job `job2` in figure 4.1, gets stuck in the 't' state according to `qstat`.

  This is often due to an `sgeexe` daemon failing on a cloud node that has executed the job, some time between the steps 4 and 5 described in section 4.4.2. A search through the error messages in `/cm/local/apps/sge/var/spool/<`*hostname*`>/messages` on the cloud nodes may help uncover the cause.

- A second job gets stuck in the 'Eqw' state according to `qstat -j <`*jobid*`>`. The error message is:
  `error: can't chdir to <path>: No such file or dir`

  This can happen if the `-cwd` option to `qsub` is specified in the jobscript to change the directory to the working directory. The problem with using the `-cwd` option is that SGE checks if the directory where the job was submitted from exists before the prolog of the second job has started. Since the prolog of the second job has not yet created all the required directories, SGE stops the job with an error message based on the 'Eqw' state.

  As a workaround, the "`cd <`*directory*`>`" command can be used instead.

4.5 Miscellaneous Cloud Commands

4.5.1 The `cm-cloud-copy` Tool

In order to transfer data to and from AWS storage (section 4.4.2), the prolog and epilog scripts can use the `cm-cloud-copy` tool. It allows AWS storage containers in AWS S3 and Glacier to be created or removed, and allows the upload and download of files and directories to those containers. The tool can also be used as a standalone tool by users directly. More information, including examples, about the `cm-cloud-copy` tool can be found in the `cm-cloud-copy(1)` man page.

4.5.2 The `cm-cloud-check` Utility

The `cm-cloud-check` utility checks the current cloud-bursting configuration for possible problems and misconfigurations. It reports any potential issues. The tool also tests communications with Amazon using some simple Amazon API calls.

Only a handful of tests are performed at present. More tests will be added over time.

4.5.3 The cm-scale-cluster Utility

The cm-scale-cluster utility is a Bright Cluster Manager utility that allows workload managers to scale a cluster up or down in size, depending on job requirements and administrator preferences. This can improve cluster efficiency by cutting down on needless energy consumption.

The development of the utility was originally aimed at cloud use, but it is now a general cluster management utility. Its use is covered in section 7.9.2 of the *Administrator Manual*.

4.5.4 The cm-cloud-remove-all Utility

This utility simply removes all clouds and associated objects:

Example

```
[root@bright70 ~]# cm-cloud-remove-all
Connecting to cluster
Removing all normal cloud nodes ...
Removing all cloud director nodes ...
 removed eu-west-1-director
Removing all netmap and tunnel interfaces ...
  remove interface tun0 of bright70
  remove interface map0 of bright70
  remove interface map0:0 of bright70
 updated bright70
  remove interface map0 of node001
 updated node001
  remove interface map0 of node002
 updated node002
Removing all tunnel networks ...
 removed eu-west-1
Removing all cloud categories ...
 removed cloud-director
Removing all cloud networks ...
 removed cloud-ec2classic
Removing all netmap networks ...
 removed netmap
Removing all cloud providers ...
 removed Amazon EC2
Done.
[root@bright70 ~]#
```

If the -d|--dryrun option is used, then it shows what the utility intends to remove during a run, but without actually removing it.

5

Cloud Considerations And Issues With Bright Cluster Manager

5.1 Differences Between Cluster-On-Demand And Cluster Extension

Some explicit differences between Cluster-On-Demand and Cluster Extension clusters are:

Cluster-On-Demand	Cluster Extension
cloud nodes only in 1 region	cloud nodes can use many regions
no cloud director	uses one or more cloud directors per region
no failover head node	failover head node possible
no VPN or NetMap	VPN and NetMap
no externalnet interface on head	can have an external interface
cluster has publicly accessible IP address	cloud directors have publicly accessible IP addresses

A note about the last entry: The access to the cloud director addresses can be restricted to an administrator-defined set of IP addresses, using the "`Externally visible IP`" entry in figure 3.1 of the *Administrator Manual*.

5.2 Hardware And Software Availability

Bright Computing head node AMIs are available for the following distributions: RHEL5/RHEL6, SL5/SL6, CentOS5/CentOS6, and SLES 11 SP1/SP2.

AMIs with GPU computing instances are available with Amazon cloud computing services only in the US East (Virginia) region the last time this was checked (April 2012). These can be used with Bright Computing AMIs with `hvm` in the name (not `xen` in the name).

To power the system off, a `shutdown -h now` can be used, or the power commands for `cmgui` or `cmsh` can be executed. These commands stop the instance, without terminating it. Any associated extra drives that were created need to be removed manually, via the `Volumes` screen in the `Elastic Block Store` resource item in the navigation menu of the AWS Management Console.

5.3 Reducing Running Costs

5.3.1 Spot Pricing

The spot price field is a mechanism to take advantage of cheaper pricing made available at irregular[1] times. The mechanism allows the user to decide a threshold spot price (a price quote) in US dollars per hour for instances. Instances that run while under the threshold are called *spot instances*. Spot instances are described further at `http://aws.amazon.com/ec2/spot-instances/`.

With the pricing threshold set:

- If the set spot price threshold is above the instantaneous spot price, then the spot instances run.
- If the set spot price threshold is below the instantaneous spot price, then the spot instances are killed.
- If the set spot price threshold is `N/A`, then no conditions apply, and the instances will run on demand regardless of the instantaneous spot price.

An *on demand instance* is one that runs regardless of the price, according to the pricing at `http://aws.amazon.com/ec2/pricing/`.

A *persistent request* is one that will retry running a spot instance if the conditions allow it.

5.3.2 Storage Space Reduction

Reducing the amount of EBS disk storage used per cloud node or per cloud director is often feasible. 15 GB is usually enough for a cloud director, and 5 GB is usually enough for a cloud node with common requirements. In `cmsh` these values can be set with:

Example

```
[bright70]% device cloudsettings eu-west-1-director
[bright70->device[eu-west-1-director]->cloudsettings]% storage
[bright70->...->cloudsettings->storage]% set ebs size 15GB; commit
[bright70->...->cloudsettings->storage]% device cloudsettings cnode001
[bright70->device[cnode001]->cloudsettings]% storage
[bright70->...->cloudsettings->storage]% set ebs size 5GB; commit
```

The value for the cloud node EBS storage can also be set in the cloud node wizard (fig. 3.9) for a Cluster Extension configuration.

[1] irregular turns out to be random within a tight range, bound to a reserve price. Or rather, that was the case during the period 20th January–13th July, 2010 that was analyzed by Ben-Yehuda et al, `http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2011/CS/CS-2011-09`

5.4 Address Resolution In Cluster Extension Networks

5.4.1 Resolution And globalnet

The globalnet network is introduced in section 3.2.3 of the *Administrator Manual*. It allows an extra level of redirection during node resolution. The reason for the redirection is that it allows the resolution of node names across the entire cluster in a hybrid cluster, regardless of whether the node is a cloud node (cloud director node or regular cloud node) or a non-cloud node (head node, regular node or networked device). A special way of resolving nodes is needed because the Amazon IP addresses are in the 10.0.0.0/8 network space, which conflicts with some of the address spaces used by Bright Cluster Manager.

There are no IP addresses defined by globalnet itself. Instead, a node, with its domain defined by the globalnet network parameters, has its name resolved by another network to an IP address. The resolution is done by the nameserver on the head node for all nodes.

5.4.2 Resolution In And Out Of The Cloud

The networks, their addresses, their types, and their domains can be listed from the network mode in cmsh:

```
[bright70->network]% list
Name (key)    Type       Netmask bits  Base address  Domain name
------------ --------- ------------- ------------- -----------
bmcnet        Internal   16            10.148.0.0    bmc.cluster
cloud         Cloud      8             10.0.0.0      cloud.cluster
externalnet   External   16            10.2.0.0      brightcomputing.com
globalnet     Global     0             0.0.0.0       cm.cluster
ibnet         Internal   16            10.149.0.0    ib.cluster
internalnet   Internal   16            10.141.0.0    eth.cluster
netmap        NetMap     16            172.31.0.0
us-east-1     Tunnel     16            172.21.0.0
```

In a Type 1 network (section 3.3.6 of the *Installation Manual*), the head node is connected to internalnet. When a cloud service is configured, the head node is also "connected" to the CMDaemon-managed NetMap "network". It is useful to think of NetMap as a special network, although it is actually a network mapping from the cloud to internalnet. That is, it connects (maps) from the nodes in one or more cloud networks such as the us-east-1 network provided by Amazon, to IP addresses provided by netmap. The mapping is set up when a cloud extension is set up. With this mapping, packets using NetMap go from the cloud, via an OpenVPN connection to the NetMap IP address. Once the packets reach the OpenVPN interface for that address, which is actually on the head node, they are forwarded via Shorewall's IPtables rules to their destination nodes on internalnet.

With default settings, nodes on the network internalnet and nodes in a cloud network such as us-east-1 are both resolved with the help of the cm.cluster domain defined in globalnet. For a cluster with default settings and using the cloud network us-east-1, the resolution of the IP address of 1. a regular node and 2. a regular cloud node, takes place as follows:

1. node001, a regular node in the internalnet network, is resolved for node001.cm.cluster to

   (a) 10.141.0.1, when at the head node. The cluster manager assigns this address, which is on internalnet. It could also be an ibnet address instead, such as 10.149.0.1, if InfiniBand has been configured for the nodes instead of Ethernet.

   (b) 172.31.0.1 when at the cloud director or regular cloud node. The cluster manager assigns this address, which is a NetMap address. It helps route from the cloud to a regular node. It is not actually an IP address on the interface of the regular node, but it is convenient to think of it as being the IP address of the regular node.

2. cnode001, a regular cloud node in the us-east-1 network, is resolved for cnode001.cm.cluster to:

   (a) 172.21.0.1 when at the head node. The cluster manager assigns this address, which is an OpenVPN tunnel address on us-east-1.

   (b) an IP address within 10.0.0.0/8 (10.0.0.1–10.255.255.254) when at a regular cloud node or at a cloud director. The Amazon cloud network service assigns the addresses in this network to the cloud director and regular cloud nodes.

An explanation of the networks mentioned in the preceding list follows:

- The nodes within all available cloud networks (all networks such as for example, us-east-1, us-west-1, and so on) are given CMDaemon-assigned addresses in the cloud node space range 172.16.0.0–172.30.255.255. In CIDR notation that is: 172.16.0.0/12 (172.16.0.0–172.31.255.255), except for 172.31.0.0/16 (172.31.0.0–172.31.255.255).

- The network address space 172.31.0.0/16 (172.31.0.0–172.31.255.255) is taken by the CMDaemon-assigned NetMap network, explained shortly. The addressing scheme for each cloud network is assigned as suggested in figure 3.5.

- Each node in a cloud network is also assigned an address in the network addressing space provided by Amazon. The assignment of IP addresses to nodes within the 10.0.0.0/8 range is decided by Amazon via DHCP.

- The netmap "network" (figure 5.1) is a helper mapping reserved for use in routing from the cloud (that is, from a cloud director or a cloud node) to a regular node. The mapping uses the 172.31.0.0/16 addressing scheme. Its routing is asymmetrical, that is, a NetMap mapping from a regular node to the cloud does not exist. Packets from a regular node to the cloud do however resolve to the cloud network as indicated by 2(a) in the preceding.

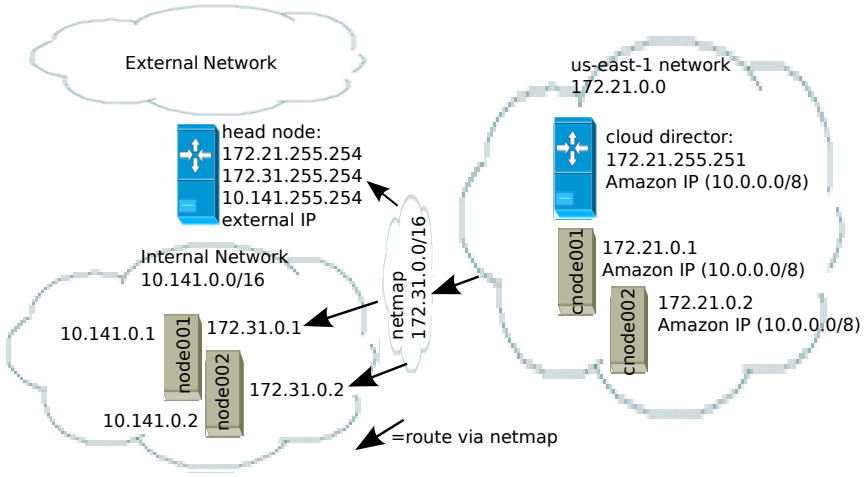

Figure 5.1: NetMap In Relation To The General Network Scheme

As pointed out in the introduction to this section (5.4), the main reason for the IP addressing network scheme used is to avoid IP address conflicts between nodes within the cloud and nodes outside the cloud.

The *difference* in resolution of the IP address for the nodes as listed in points 1 and 2 in the preceding text is primarily to get the lowest overhead route between the source and destination of the packet being routed. Thus, for example, a packet gets from the regular cloud node to the cloud director with less overhead if using the Amazon cloud IP addressing scheme (10.0.0.0/8) than if using the Bright OpenVPN addressing scheme (172.21.0.0/16). A secondary reason is convenience and reduction of networking complexity. For example, a node in the cloud may shut down and start up, and get an arbitrary Amazon IP address, but using an OpenVPN network such as `us-east-1` allows it to retain its OpenVPN address and thus stay identified instead of having the properties that have been assigned to it under Bright Cluster Manager become useless.

6

Virtual Private Clouds

A *virtual private cloud* is an implementation of a cluster on a virtual network in a cloud service provider. The Amazon Virtual Private Cloud (Amazon VPC) is an implementation of such a virtual private cloud. The Amazon VPC is documented more fully at `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-vpc.html`.

Managing VPCs would normally require significant networking expertise. Bright Cluster Manager makes it easier to do this, so that the administrator can focus on using them productively, instead of on working out VPC configurations.

The following VPC-related terms are explained and compared in this chapter:

- *EC2-Classic* (page 51)
- *EC2-VPC* (page 52)
- *classic cloud* (page 52)
- *defaultVPC* (page 52)
- *private cloud* (page 53)
- *custom VPC* (page 53)
- *elastic IP addresses* (page 57)

6.1 EC2-Classic And EC2-VPC

6.1.1 EC2-Classic Vs EC2-VPC Overview

So far, this manual has discussed configuring clusters within Amazon EC2. The EC2 designation actually covers two kinds of platforms:

- EC2-Classic: This platform provides an environment that corresponds to a physical network. Instances in the same region exist on the same physical network and rely on explicitly configured security groups to restrict unauthorized access from other instances on the same network. A cloud instance that is created in such a network can be called a classic cloud cluster, or simply a cloud cluster.

  Amazon is gradually phasing out the EC2-Classic platform.

- EC2-VPC: This platform is replacing EC2-Classic. It provides an environment corresponding to an isolated virtual network. A cloud cluster instance implemented on this virtual network is thus a virtual private cloud, or VPC, as described at the start of this section (section 6).

  The EC2-VPC platform offers some extra features that are not available, or not as easy to configure, on the EC2-Classic platform:

  - Multiple VPCs can be configured per region
  - The inherent isolation of Amazon VPCs makes them more secure by default
  - their network properties can be customized

The isolated network design of a VPC means that instances started within a VPC cannot by default communicate with instances outside. *Elastic IP* addresses (page 57) are used to explicitly allow communication with the outside.

6.1.2 EC2-Classic Vs EC2-VPC And AWS Account Creation Date

The type of platform that can be accessed by an AWS account varies as indicated by the following table:

Account Creation Date	Typical Platform Offered
Before start of 2013	EC2-Classic only
In first half of 2013	EC2-Classic or EC2-VPC*
After first half of 2013	EC2-VPC only, in most or all regions

*Typically depends on the region accessed.

Most new AWS accounts do not provide an EC2-Classic platform. However, to maintain backward compatibility for users who are migrating to EC2-VPC, and who have applications that run on the EC2-Classic platform, Amazon provides the defaultVPC instance on the EC2-VPC platform.

6.1.3 The Classic Cloud And The DefaultVPC Instances

The *classic cloud* is a cloud instance that EC2-Classic supports.

The `defaultVPC` instance is a special VPC instance that emulates EC2-Classic behavior on the EC2-VPC platform. This allows legacy applications that do not support EC2-VPC to run on it. A legacy application that runs in a defaultVPC instance may be thought of as having its EC2-Classic API calls translated into EC2-VPC API calls. The defaultVPC instance is available in all regions that do not offer the EC2-Classic platform.

There is one major difference between the network environments of EC2-Classic and the defaultVPC instance: For EC2-Classic instances, the base address of network inside Amazon is 10.0.0.0/8. In contrast, for defaultVPC instances the base address is 172.31.0.0/16.

When creating a new cloud provider account, Bright Cluster Manager automatically detects which regions offer the EC2-Classic platform, and which do not. The suggested base address of the cloud network that is

to be created is then automatically matched according to the regions. The platform supported, EC2-Classic or EC2-VPC, is also displayed in `cmgui` when the cloud director is being created.

A few Amazon AWS accounts provide the EC2-Classic platform for only a certain subset of all available regions, and provide EC2-VPC in other regions. In such a case, when a new cloud provider account is created in Bright Cluster Manager with a cloud director in both types of platforms, then two cloud networks can be created. If only a single cloud director is run, then only one network is created, and the network base address in that case depends on the platform, EC2-Classic or EC2-VPC, that it is run on. However, if two cloud directors are started up, with each cloud director on a different platform, then one cloud director runs on one platform and associated network, and the other cloud director on the other platform and associated network.

6.1.4 The Private Cloud And Custom VPC Instances

A *private cloud* (without the "virtual" in front) is the term used in the Bright Cluster Manager manuals, as well as by Amazon, and in general, for a general VPC instance.

A *custom VPC* is the term used in the manual to mean a general VPC instance, but one that is not a defaultVPC instance.

Thus, in terms of math sets:

`private clouds` = `custom VPCs` + `defaultVPCs`

In the context of Amazon VPCs, the term private cloud is often used by administrators, by convention and for convenience, to mean the more precise term of custom VPC as defined here, implicitly ignoring possible defaultVPC instances. The Bright Cluster Manager software itself also follows this convention. In this chapter of the manual (6), however, using the term "private cloud" for this is avoided, and the terms are adhered to precisely as defined, in order to avoid confusion.

Attempting to change a defaultVPC instance to a custom VPC instance by editing defaultVPC properties directly with Bright Cluster Manager is not possible, because these properties are hidden behind the EC2-Classic facade. This kind of change can be done via the Amazaon Webconsole instead. If Bright Cluster Manager requires that the custom VPC functionality of a general VPC instance is needed in Amazon VPC, then a custom VPC has to be created within Bright Cluster Manager. How to do this is described in section 6.3.

6.1.5 Cloud Cluster Terminology Summary

The cluster terminology used so far can be summarized as follows:

cluster term	platform	type and connectivity
classic cloud	EC2-Classic	classic cloud cluster that has direct connectivity to the outside
defaultVPC	EC2-VPC	a VPC that looks like it has direct connectivity to the outside because it emulates a classic cloud cluster
custom VPC	EC2-VPC	isolated VPC with no connectivity to the outside by default, and NAT gateway connectivity to the outside when made to connect
private cloud	EC2-VPC	both defaultVPC and custom VPC

6.2 Comparison Of EC2-Classic And EC2-VPC Platforms

There are several differences between EC2-Classic and EC2-VPC platforms. The most important ones are:

- Cloud nodes created inside the EC2-VPC platform do not have an external (public) IP address assigned to them by default. An exception to this is the case of nodes running in a defaultVPC instance, which emulates EC2-Classic network behaviour. Having no public IP address by default allows for a greater degree of out-of-the-box security.
- Custom VPCs are self-contained and securely isolated from the instance of other users.
- Custom VPCs are partitioned into multiple network segments, called *subnets* (section 6.3.1).
- It is possible to specify a custom base network address for the custom VPC. This is in contrast to the EC2-Classic platform, where a base network address always has the value of 10.0.0.0/8. For a defaultVPC instance the base network address takes the value of 172.31.0.0/8.

6.3 Setting Up And Creating A Custom VPC

By default, when Bright Cluster Manager creates a new cloud provider account, the cloud nodes created are EC2-Classic instances or defaultVPC instances inside the EC2-VPC platform. That is, they are not nodes in a custom VPC instance. This default behavior is expect to change in a later version of Bright Cluster Manager as Amazon and Bright Cluster Manager both evolve.

Bright Cluster Manager can be set to create custom VPC instances inside the EC2-VPC platform. The EC2-VPC platform is recommended for all new cloudbursting setups.

6.3.1 Subnets In A Custom VPC

The components of a custom VPC include subnets, the nodes that run in them, and static IP addresses. The subnets are logical network segments within the network range of that custom VPC. Subnets can be thought of as interconnected with a central "magic" router, with Bright Cluster Manager managing the routing tables on that router. The routing ensures correct subnet communication. Inside Bright Cluster Manager, subnets are represented as a type of network (section 3.2 of the *Administrator Manual*), with a value for type set to `CLOUD`.

Subnets for a custom VPC must have non-overlapping ranges. If there are multiple custom VPCs being managed by Bright Cluster Manager, then a particular subnet may be assigned to one custom VPC at the most.

Two series of valid network ranges could be:

Example

1. 10.0.0.0-10.0.31.255 (10.0.0.0/19),
   10.0.32.0-10.0.63.255 (10.0.32.0/19),
   10.0.64.0-10.0.95.255 (10.0.64.0/19).
2. 192.168.0.0-192.168.0.255 (192.168.0.0/24),
   192.168.1.0-192.168.1.255 (192.168.1.0/24).

The `sipcalc` command (page 59 of the *Administrator Manual*) is a useful tool for calculating appropriate subnet ranges. At least one subnet must be assigned to a custom VPC before an instance can be created in that cloud. Typically two or more subnets are assigned, as shown in the custom VPC creation example in the following section.

6.3.2 Creating The Custom VPC

After subnets have been configured, a custom VPC can be created by specifying:

- the name
- the default region
- base address
- number of netmask bits

The network of the custom VPC must obviously be a superset of its subnets. Any subnets of the custom VPC must also be specified. Subnets can be added to or removed from an already-created custom VPC, but only if any cloud node instances within them are terminated first.

There are several ways to set up and create the subnets and custom VPC instance in Bright Cluster Manager:

1. by using the command line `cm-cloud-setup-private-cloud` utility,
2. by using the `cmgui` private cloud creation dialog box,
3. by manually creating and configuring the `private cloud` object using `cmsh`.

These are described next:

6.3.3 1. Subnet Setup And Custom VPC Instance Creation Using cloud-setup-private-cloud

Once the cloud provider account has been configured, using the cm-cloud-setup utility (section 4.1), or by using the cmgui wizard (section 3.1), the cm-cloud-setup-private-cloud utility can then be run to set up a custom VPC.

The utility prompts the user to choose a cloud provider account, a region to create the VPC in, and the base address of the VPC. It goes on to create the custom VPC, and finishes by prompting whether to move any eligible cloud nodes to the custom VPC.

6.3.4 2. Subnet Setup And Custom VPC Creation Using cmgui

For the cloud provider resource item, inside the Private Clouds tab, clicking the Add button launches a dialog box to create a custom VPC.

6.3.5 3. Subnet Setup And Custom VPC Creation Using cmsh

Similarly with cmsh, the subnets to be used for the custom VPC are created first, before creating the private cloud, as shown in the following examples.

- **Subnet creation and cloning:** In the following example session, an arbitrary naming scheme is used for subnets, with a pattern of: <*name of custom VPC*>-sn-<*number*>. Here, sn is an arbitrary abbreviation for "subnet":

  Example

  ```
  [bright70->network]% add vpc-0-sn-0
  [bright70->network*[vpc-0-sn-0*]]% set type cloud
  [bright70->network*[vpc-0-sn-0*]]% set baseaddress 10.0.0.0
  [bright70->network*[vpc-0-sn-0*]]% set netmaskbits 24
  [bright70->network*[vpc-0-sn-0*]]% set ec2availabilityzone eu-west-1a
  [bright70->network*[vpc-0-sn-0*]]% commit
  ```

  Setting the ec2availabilityzone property is optional. It causes the subnet to be created in a specific availability zone. Leaving its value empty creates the subnet inside a randomly chosen availability zone. Having all subnets of the custom VPC inside the same availability zone is advised for better network performance. The availability zone set for the network must be one of the availability zones available for the region inside which the private cloud will be created.

  Once the first subnet has been created, it can be cloned:

  Example

  ```
  [bright70->network]% clone vpc-0-sn-0 vpc-0-sn-1
  [bright70->network*[vpc-0-sn-1*]]% set baseaddress 10.0.1.0
  [bright70->network*[vpc-0-sn-1*]]% commit
  ```

- **Custom VPC creation:** The following example session in the privateclouds submode of the cloud mode, creates a private

cloud called vpc-0. The private cloud is actually a custom VPC according to the strict definition of a private cloud instance in the section on page 53. It is of type ec2 and within a network that contains the two subnets specified earlier.

Example

```
[bright70->cloud[Amazon EC2]->privateclouds]%
[bright70->...->privateclouds]% add ec2privatecloud vpc-0
[bright70->...->privateclouds*[vpc-0*]]% set region eu-west-1
[bright70->...*[vpc-0*]]% set baseaddress 10.10.0.0
[bright70->...*[vpc-0*]]% set netmaskbits 16
[bright70->...*[vpc-0*]]% set subnets vpc-0-sn-0 vpc-0-sn-1
[bright70->...*[vpc-0*]]% commit
```

6.3.6 Elastic IP Addresses And Their Use In Configuring Static IP Addresses

Unlike defaultVPC and EC2-Classic instances, a custom VPC instance does not have an externally visible (public) IP address assigned to it by Amazon by default. Without an externally visible IP address, the custom PVC cannot communicate with the internet, and it cannot even be an endpoint to an outside connection. To solve this issue, Amazon *elastic IP addresses* (EIPs) can be used to assign a public IP address to a custom VPC cloud.

EIP addresses are the public IP addresses that Amazon provides for the AWS account. These addresses are associated with defaultVPC and EC2-Classic cloud instances by Amazon by default. These addresses can also be associated with custom VPC instances. The public addresses in the set of addresses can then be used to expose the custom VPC instance. In this manual and in Bright Cluster Manager, EIPs are referred to as "static IPs" in the cloud context. When allocating a static IP addres, the exact IP address that is allocated is a random IP address from the set of all public IP addresses made available in the specified region by the configured cloud provider.

Automatic allocation of static IP addresses:
When a cloud director instance is started inside a custom VPC, CMDaemon automatically allocates and assigns a static IP address to it. By default, the static IP address is automatically released when the cloud director instance is terminated. This behavior can be changed in the CMDaemon cloud settings for the cloud director.

Manual allocation of static IP addresses:
It is also possible to manually allocate a static IP address to a cloud director using cmgui or cmsh.

Allocating a static IP address in cmsh is done using the staticip allocate command, followed by the string indicating the region in which the static IP address is to be allocated. In cmsh, the command is issued inside a cloud provider object. A new static IP address is then made available and can be assigned to instances running within custom VPCs.

After allocation, the static IP address can be assigned and reassigned to any instance inside any custom VPC created within the region in which the IP address was allocated.

Example

```
[bright70] cloud use amazonec2
[bright70->cloud[Amazon EC2]]% staticip allocate us-west-1
Allocating Static IP. Please wait...
Successfully allocated the following static IP: 54.215.158.42
[bright70->cloud[Amazon EC2]]% staticip list
Cloud Provider   Cloud Region  Static IP      Assigned to
---------------- ------------- -------------- ---------------
Amazon EC2       us-west-1     54.215.158.42  <not assigned>
[bright70->cloud[Amazon EC2]]%
```

An allocated static IP can be released with the `staticip release` command in `cmsh`:

Example

```
[bright70->cloud[Amazon EC2]]% staticip release 54.215.158.42
Releasing static IP 54.215.158.42. Please wait...
Successfully released the static ip.
[bright70->cloud[Amazon EC2]]%
```

Once the IP address has been released, it may no longer be used for instances defined in the custom VPC.

The `staticips` command lists all allocated static IPs for all configured cloud providers.

The `staticip list` command lists static IP addresses for the currently active cloud provider object.

In `cmgui` the static IPs can be managed via the "Static IPs" tab of a cloud provider object.

6.3.7 Subnets With Static IP Addresses In A Custom VPC Recommendation

Subnets can be set up in many ways inside a custom VPC. The following is recommended:

- There must be exactly one network containing all the instances which have static IP addresses. This network should contain the cloud director. The network with the cloud director is arbitrarily referred to as the "public" network.
- There must be zero or more networks containing instances with no static IP addresses assigned to them. Such networks are arbitrarily referred to as the "private" subnets.

Instances in the private subnets have no static IP addresses assigned to them, so by default they do not communicate with outside networks. To allow them to connect to the outside, the cloud director instance is automatically configured by CMDaemon as a NAT gateway for outside-bound traffic, for the instances existing inside the private subnets.

6.3.8 Assignment Of Nodes To Subnets And Cloud Platforms

A cloud node instance is connected to a network by its `eth0` interface. The network is one of those covered in following table, that is: classic physical, classic emulated, or subnet of a custom VPC.

what cloud is the eth0 interface connected to?	cloud instance type and network that the node joins
classic cloud	classic cloud cluster instance, in classic physical network (10.0.0.0/8)
defaultVPC	defaultVPC instance, in classic emulated network (172.31.0.0/8)
custom VPC	inside VPC instance, in the connected subnet (if any) of that network

Therefore, when the cloud node is being created inside EC2, the CMDaemon must tell the EC2 environment which of these networks is going to be attached to the eth0 interface of the newly created cloud node.

This information is deduced by CMDaemon by looking at the interface configuration of the cloud node Bright Cluster Manager. More specifically, it is deduced from the value set for `network` in the cloud node's eth0 interface settings.

If that network is part of a custom VPC, that is, if it is a subnet, then the node starts inside the custom VPC instance. Otherwise, it starts inside the EC2-Classic or defaultVPC instance in that region.

For example, the cloud director node is started inside the `vpc-0-sn-0` network in the following session. It is considered a custom VPC node, and starts up inside the EC2-VPC platform:

Example

```
[bright70->device[us-west-1-director]->interfaces]% list
Type         Network device name  IP               Network
------------ -------------------- ---------------- ----------
physical     eth0 [dhcp]          0.0.0.0          vpc-0-sn-0
tunnel       tun0 [prov]          172.18.255.251   us-west-1
```

In contrast, if the cloud network assigned to the `eth0` interface is the cloud network representing the network environment of an EC2-Classic or defaultVPC cloud, then the node is considered to be an EC2-Classic node. It then starts up inside the EC2-Classic platform by default:

Example

```
[bright70->device[us-west-1-director]->interfaces]% list
Type         Network device name  IP               Network
------------ -------------------- ---------------- ----------
physical     eth0 [dhcp]          0.0.0.0          cloud
tunnel       tun0 [prov]          172.18.255.251   us-west-1
```

Once a cloud node instance has been instantiated inside a specified subnet it cannot be reassigned to a different subnet, nor can it be reassigned to a different custom VPC. The cloud instance must first be termi-

nated and reconfigured. Only then can it be powered on inside a different subnet.

6.3.9 Creating A Cloud Director In A Custom VPC

To create a cloud director in a custom VPC using `cmgui`, the cloud director must first be created inside the EC2-Classic platform region in which the custom VPC is created. This can be done via the `Add Cloud Director` button inside the `Overview` tab of a cloud provider account. After this has been done, the cloud director must be moved from the EC2-Classic platform to the custom VPC, as explained in section 6.3.11.

6.3.10 Creating Cloud Compute nodes In A Custom VPC

Creating cloud compute nodes inside a custom VPC can be done in a similar way to creating cloud compute nodes for the EC2-Classic platform. That is, by clicking the `Create Cloud Nodes` button in the overview tab of the cloud provider in `cmgui`. However, to create a cloud node inside the custom VPC, a subnet of the custom VPC must be specified when selecting the network of the eth0 interface of the node. To avoid confusion, it is sensible to make this a different subnet from the one in which the cloud director node for that particular custome VPC is assigned.

An alternative solution to creating cloud compute nodes in a custom VPC is to instruct the cluster manager to automatically move the existing ones while also moving the cloud director to the custom VPC, as explained in the following section.

6.3.11 Moving Existing Nodes To A Custom VPC

After a custom VPC has been configured, it is possible to automatically reconfigure the existing cloud nodes to make them start inside that custom VPC. This is an alternative to creating new nodes inside the custom VPC from scratch. An existing cloud director can be moved to a custom VPC cloud using the `Move Cloud Director` button in `cmgui`. This button can be clicked in the `Overview` tab of a cloud provider account, and it opens up a dialog box. After completion of the dialog, the cloud director is moved to a custom VPC in the same region. It can also move any other cloud compute nodes managed by the selected cloud director.

Moving a node to a custom VPC effectively terminates the current EC2 instance, and creates a new one inside the target custom VPC.