Crowbar User Guide
Version 1.2

Revised 12/16/2011

For the most recent versions,

consult http://github.com/dellcloudedge/crowbar/wiki

DOCUMENT PROVIDED UNDER APACHE 2 LICENSE

Notes, Cautions, and Warnings

NOTE: A NOTE indicates important information that helps you make better use of your computer

CAUTION: A CAUTION indicates potential damage to hardware or loss of data if instructions are not followed.

WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

Information in this document is subject to change without notice.

© 2011 Dell Inc. All rights reserved.

Reproduction of these materials is allowed under the Apache 2 license.

Trademarks used in this text: Dell™, the DELL logo, Nagios™, Ganglia™, Opscode Chef™, OpenStack™, Canonical Ubuntu™ , VmWare™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, KACE™, FlexAddress™ and Vostro™ are trademarks of Dell Inc. Intel®, Pentium®, Xeon®, Core™ and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™, and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, MS-DOS® and Windows Vista® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat Enterprise Linux® and Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® is a registered trademark and SUSE ™ is a trademark of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, Virtual SMP®, vMotion®, vCenter®, and vSphere® are registered trademarks or trademarks of VMWare, Inc. in the United States or other countries.

Other trademarks and trade names may be used in this publication to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

December 2011

Contents

INTRODUCTION **5**

Concepts 5
Opscode Chef Server 5
Dell Specific Options 5

THE CROWBAR FRAMEWORK **6**

Architecture 6
System End State 6
Node Provisioning 7
NTP 7
DNS 8
Nagios™ 8
Ganglia™ 8
Logging 8
Network Setup 8
Default Networks 8
IP Addressing *9*

USER INTERFACE **10**

Using Crowbar 10
Nodes *11*
Barclamps *11*
General Layout 11
Nodes (system dashboard) 12
Node Switch Groups *12*
Node Details *12*
Bulk Edit *16*
Barclamps 16
Barclamp List *17*
Proposal View/Edit *18*
Deployment Functions 18
Life Cycle *19*
PXE State Machine *20*
Discovering Node *21*
Allocating Node *21*
Hardware Install *21*
Base OS Install *21*
Ready for Role *21*
Hardware Update *21*
Applying Role *22*
Included Barclamps *22*

BARCLAMP DETAILS 22

CROWBAR BARCLAMP 22
DEPLOYER BARCLAMP 23
PROVISIONER BARCLAMP 24
NETWORK BARCLAMP 25
RAID BARCLAMP 27
BIOS BARCLAMP 27
IPMI BARCLAMP 27
NTP BARCLAMP 28
LOGGING BARCLAMP 29
NAGIOS BARCLAMP 29
GANGLIA BARCLAMP 29
TEST BARCLAMP 30

SUPPLEMENTAL MATERIAL 30

SYSTEM VERIFICATION 30
MANAGING BARCLAMPS 30
Introduction 30
Importing a Barclamp 31
More Information 31

SUPPORT 31

CROWBAR SUPPORT 31

Introduction

This document provides instructions you for operating Crowbar. Please refer to additional user guide for specific products that are deployed by Crowbar such as OpenStack™ or Apache™ Hadoop™.

Concepts

The purpose of this guide is to explain the user interface of Crowbar. Use the Crowbar User Guide for assistance with installing Crowbar and configuring the target system.

Note: Concepts beyond the scope of this guide are introduced as needed in notes and references to other documentation.

Opscode Chef Server

Crowbar makes extensive use of Opscode Chef Server**,** http://opscode.com. To explain Crowbar actions, you should understand the underlying Chef implementation.

To use Crowbar, it is not necessary to log into the Chef Server; consequently, use of the Chef UI is not covered in this guide. Supplemental information about Chef is included.

This guide provides this additional Chef information as notes flagged with the Opscode logo.

Dell Specific Options

The Dell EULA version of Crowbar provides additional functionality and color pallets than the open source version. When divergences are relevant, they are identified.

To perform some configuration options and provide some integration, we use libraries that cannot be distributed using open source.

Crowbar is not limited to managing Dell servers and components. Due to driver requirements, some barclamps, for example: BIOS & RAID, must be targeted to specific hardware; however, those barclamps are not required for system configuration.

The Crowbar Framework

Architecture

Crowbar provides a modular platform containing the building blocks to provision, monitor, and operate a large scale cloud deployment. Starting with bare metal installation, Crowbar automates all the required installation and configuration tasks. The core capabilities provided are:

- Hardware configuration – updating and configuring BIOS and BMC boards.
- Deployment of base operating system.
- Deployment of cloud components.
- Providing core network infrastructure services (NTP, DNS, DHCP).
- Monitoring availability and performance of all deployed components.

To allow easy integration into your existing environments, Crowbar allows customization of its components. You can disable the default monitoring tools (Nagios and Ganglia) if you prefer to use your own existing monitoring tools, and internal cloud services can be connected to extant services; for example, the cloud's NTP service can be configured to synchronize with existing servers.

Each function in the cloud is controlled by a Crowbar component called a Barclamp. There are Barclamps for Nagios, Ganglia, NTP, and a variety of other basic services. Each Barclamp is responsible for all the aspects of the underlying technology required to make it usable. To control the operation of a Barclamp, you create a proposal for the Barclamp (or may edit one already in place). A proposal comprises several parts:

- Parameters to customize the operation of the function; for example: upstream DNS resolvers.
- List of machines in the deployment that fulfill the different roles in the function.
- Internal system information.

When provisioning a function, you start with a proposal generated by Crowbar. Each core service running on the admin server has a default proposal included as part of the Crowbar installation. You can edit these proposals before installing these services on the admin node.

When a proposal is committed, Crowbar configures the Chef server and other components in the systems (TFTP, DHCP, and so on) to build the setup described in the proposal. Machines in the deployment affected by the proposals have their configuration updated using Chef client-side components. At the end of the process, the function described by the proposal is ready for use.

Finally, a cloud deployment is dynamic. Machines come and go, break down, or get repurposed. Crowbar's operational model makes sure that machines are hooked into the key infrastructure services. Critical services (for example Nagios monitoring) are installed automatically on newly provisioned machines by default, and those machines may be easily allocated for use with any additional Crowbar services desired.

System End State

The sections that follow describe the services and capabilities available assuming the system is installed with defaults. As mentioned above, the Crowbar framework allows for many customizations. This section focuses on the primary use

cases for Crowbar, namely integrating all the functions into an existing network environment. Later sections describe more advanced customization options.

Node Provisioning

When a new node is added to the system, set it up to allow PXE booting. Once a machine is powered on, Crowbar uses the PXE boot protocol to manage its provisioning process.

After a system is fully installed by Crowbar, it has the following characteristics:

- BIOS is updated and configured based on the system's usage.
- BMC (Board Management Controller) is configured to allow management and IPMI support.
- Administration access to the OS is configured – IP addressing and SSH keys are installed.
- Nagios and Ganglia monitoring scripts are installed for the functions deployed on the system.
- Chef-client daemon is configured to maintain the system's state in sync.
- NTP sync client is configured.

Additionally, the system is configured to fulfill the functions that are deployed on it and is added to the appropriate cluster.

- Networking administration.

Crowbar's network Barclamp carries on responsibilities related to L2/L3 management, namely:

- Physical NIC configuration – BMC port allocation (teamed or not).
- VLAN configuration on nodes.
- IP address location service, used by the rest of Crowbar. Addresses can be allocated from different pools, meant for different usages; for example: Admin network, BMC, Storage, and Public.

The above functions involve managing information on the server and executing operations on nodes as they are provisioned. On a node, NICs are defined to match VLANs and appropriate addressing information is configured.

NTP

The admin node runs an NTP server to synchronize time on all the machines in the cluster. Optionally, the NTP server can synchronize with upstream servers, in which case nodes are configured to sync their local time to those servers instead.

DNS

The admin node runs a DNS server to allow resolution of internal and (optionally) external names. The DNS server can be configured with the following:

- A list of upstream DNS servers to contact.
- A set of static mappings.
- The default domain name suffix.
- Crowbar makes sure that when a new machine is added to the deployment, it has a default entry added to the DNS zone. The default host name is the machine's MAC address prefixed by the letter, 'd' (for example: d00-a4-b3-c2-d1-e0.yourdomain.com).

Nagios™

Nagios monitors provisioned services for availability. Each cluster instance is represented as a host group, letting you quickly identify the health of a given instance.

Install the Nagios server on the admin node. It is configured to monitor all the nodes in the system. As new nodes are brought online, Crowbar dynamically updates Nagios to include them.

Ganglia™

Ganglia monitors the installed cluster for capacity and performance information, letting you easily gauge the cluster's capacity and check recent activity.

Logging

The admin node serves as a central log repository. Its syslog daemon is configured to accept remote messages and each node is configured to forward all messages there.

Network Setup

Due to the nature of the different software used, the network is set up as flat as possible using a dedicated BMC port and bonded LOMs. Crowbar manages all networks, and comes out of the box preconfigured to allow the initial configuration to come up quickly by predefining the storage, admin, public, and BMC networks.

The Crowbar network configuration can be customized to better map to site specific networking needs and conventions. These changes include adding additional vLANs, changing vLAN mappings, and teaming NICs.

Please refer to the *Network Barclamp* section for specific details.

NOTE: Networks for the environment are configured when the crowbar installation is performed. They cannot be changed without re-installing crowbar. For specific information on how to configure the network JSON file please see the Deployment Guide.

Default Networks

The default networks are presented in the following table. These defaults can be modified prior to installing crowbar for your specific environment.

Table 1-1: Default Networks

Usage	Description	Default reserved vLAN tag	Tagged
Admin/Internal vLAN	Used for administrative functions such as Crowbar node installation, TFTP booting, DHCP assignments, KVM, system logs, backups, and other monitoring. There is only one vLAN set up for this function and it is spanned across the entire network.	100	Not tagged
BMC vLAN	Used for connecting to the BMC of each node.	100	Not tagged
Storage vLAN	Used by the Swift storage system for replication of data between machines, monitoring of data integrity, and other storage specific functions (802.1q Tagged).	200	Tagged
Edge/External vLANs	Used for connections to devices external to the Cloud infrastructure; these include externally visible services such as load balancers and web servers. Use one or many of these networks, dependent on the need to segregate traffic among groups of servers (802.1q Tagged).	300	Tagged

Note: The admin and BMC networks are expected to be in the same L2 network.

Each network defined in the system has the following parameters:

Table 1-2: Network Parameters

Name	Default	Description
vlan	Integer	The vlan to use on the switch and interfaces for this network
use_vlan	true	A value of true indicates that the vlan should applied to the interface. A value of false assumes that the node will receive untagged traffic for this network.
add_bridge	false	indicates if the network should have a bridge built on top of it. The bridge will be br.
subnet	IP Address	The subnet for this network
netmask	Netmask	The netmask for this network
router	IP Address	The default router for this network
broadcast	IP Address	The default broadcast address for this network
ranges	map	This contains a map of strings to start and stop values for network. This allows for sub-ranges with the network for specific uses. e.g. dhcp, admin, bmc, hosts.

IP Addressing

The IP address can be assigned in this fashion, using large subnets to support many machines on the production network. The following table shows the default networks which are installed when the default Network JSON is used. In each network, the first 10 IP addresses are reserved for switches, routers, and firewalls.

Table 1-3: Default Network Addresses

LAN	VLAN	Network	Subnet	Gateway	802.1q	Bridged
Storage	200	192.168.125.0	255.255.255.0	none	Yes	No
Public	300	192.168.122.0	255.255.255.0	192.168.122.1	Yes	No
Admin	100	192.168.124.0	255.255.255.0	192.168.124.1	No	No
BMC	100	192.168.124.0	255.255.255.0	192.168.124.1	No	No
BMC_Vlan	100	192.168.124.0	255.255.255.0	192.168.124.1	No	No

Note: Each network's ".1" address is reserved for the network gateway.

User Interface

The Crowbar interface has two primary concepts: nodes and barclamps. All actions are focused on management of these two elements.

Using Crowbar

Crowbar is delivered as a Web application available on the admin node using HTTP on port 3000. By default, you can access it using http://192.168.124.10:3000 (see table below). Additionally, the default installation contains an implementation of Nagios and Ganglia for status and performance monitoring of the installation.

Note: Nagios, Ganglia and Chef can be accessed directly from a web browser or via selecting one of the links on the Dashboard in the Crowbar UI.

Table 2-1: Service URLs

Service	URL	Credentials
SSH	root@192.168.124.10	crowbar
Crowbar UI	http://192.168.124.10:3000/	crowbar / crowbar
Nagios UI	http://192.168.124.10/nagios3	nagiosadmin / password
Ganglia UI	http://192.168.124.10/ganglia	nagiosadmin / password
Chef UI	http://192.168.124.10:4040/	admin / password

Note: Crowbar has been tested on the following browsers: FireFox 3.5+, FireFox 4.0, Internet Explorer 8, and Safari 5. HTML5 compatibility and a minimum screen resolution of 1024x768 are recommended.

The IP address (192.168.124.10) is the default address. Replace it with the address assigned to the Admin node.

Crowbar has two primary concepts for users Nodes and Barclamps. Before talking about the UI, it's important to understand how they are used by Crowbar.

Nodes

Nodes represent distinct servers in your environment. A server is a single operating system that can be physical or virtual with multiple NICs and HDDs. Each server is identified uniquely by the MAC address of the NIC on the administrative network.

Crowbar nodes map directly to Chef nodes. In fact, all data used in Crowbar is stored entirely in Chef. Chef is the database for Crowbar. Changing a node's data in Chef changes it in Crowbar.

Barclamps

- Barclamps represent modular capabilities that you can install onto none, some, or all of the nodes in your environment.
- Barclamps are activated by generating a Proposal for that Barclamp. It is possible to generate multiple proposals for a barclamp.
- Once a proposal is reviewed, you must activate it before it becomes an Active in the system.

In addition to logic for Crowbar, barclamps are decomposed in Chef as multiple components: Crowbar data bag entries, cookbooks, recipes, and roles. Our objective is to allow the Chef components used by Barclamps to operate in Chef even without Crowbar.

Barclamps have a specific life cycle that is discussed in more detail as we explore the user interface. Information about using, creating, and extending barclamps is included in the Supplemental Material section.

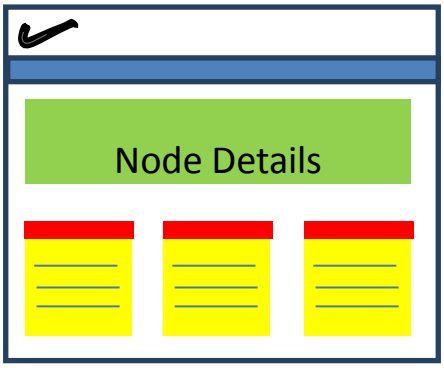
General Layout

The menu for Crowbar is displayed on the upper right side of the page. Clicking on one of the menu sections causes related content to display on the lower section of the screen.

Alerts or confirmation messages may be displayed between the menu and the page content. Most Crowbar screens automatically update state information so you should not have to refresh the page to get the most current information.

Nodes (system dashboard)

The Dashboard shows all the nodes in the system and lets you manipulate their power and configuration states.



Node Switch Groups

Nodes are shown organized by the physical switching infrastructure, which is discovered automatically during the configuration process. Use the switch and port of each node's administrative network interface to determine groups and order within groups.

Note: If you use a consistent pattern for connecting nodes to switches then the Crowbar display matches your nodes' physical location.

The top of the group box (red in illustration) shows the Switch MAC address and a pie chart of the nodes' status within each group. Nodes (yellow in illustration) connected to the switch display in port order (lowest on top) with their current deployment state shown by the status light.

Table 2-2: Deployment States

Status	Icon	Comment	User Action
Ready		Requested services provisioned.	Configure as needed
Waiting	blinking	Waiting for user input	Node waiting to be allocated. See "Bulk Edit" or include node in a proposal.
Pending	solid	Hardware and Operating System Installation	*None: Crowbar is working*
In Process	spinning	Crowbar and Chef actively deploying	*None: Crowbar is working.*
Failed	blinking	Failure detected operating on Node	Correct issue
Unknown		In between states or not reporting for 20 minutes (likely powered off).	Restart server if desired

The Admin node is the node that runs the Crowbar, Chef, and other core services. It exists in the system before the user interface is available.

Node Details

Clicking on a node's name displays details about the selected node in the details panel (green in illustration). The detail panel displays important information about the node including its FQDN, uptime, switch connectivity hardware profile, and a detailed list of all active network connections.

Node detail only shows a tiny fraction of the total details that Chef tracks for each node. To see the complete list, examine the Run List and Attributes for each node in Chef.

The Links list is barclamp specific and expands depending on which barclamps are using the selected node. Links open a new window to view additional information about the node. The Barclamps and Roles lists indicate what capabilities have been assigned to the node. The Bulk Edit

The Build Edit screen (added in v1.2) allows you to quickly update the description, RAID, BIOS, and allocation state for all the nodes in the system.

Note: If you allocate a node(s) then Crowbar will immediately begin to deploy it (them). Allocation is a one way process. You cannot "unallocate" a node.

The RAID and BIOS selections highlight the nodes' current value using [brackets]. The choices offered for BIOS and RAID selectors are determined by the Bios and Raid Barclamps. They can be expanded after installation.

If no change it made then the node will not be updated.

Barclamps

The Barclamps page lets you create, edit, review and deploy proposals for Barclamps. These activities are the way that Crowbar decides which nodes to deploy and how to configure them.

Note: Previous versions of Crowbar split the Barclamp lifecycle into multiple pages. All three phases of the lifecycle (barclamp, proposal, active role) are now represented together on the Barclamps page.

Barclamp List

The Barclamps page shows a list of all available barclamps (see "Included Barclamps" table). The barclamps are represented as the blue lines in the figure to the right. Expanding a barclamp (by clicking ▷) displays the associated proposals for the selected barclamp (red box in the figure). You jump directly to the relevant proposal by clicking on its name under its barclamp.

A barclamp will show the status of the proposals that are attached to it using a status light (see table below). If multiple proposals are assigned, then multiple light are displayed. If there are no proposals, a diamond is displayed. Hovering over the light will show you the name and status of the matching proposal. The proposals status updates automatically without a refresh.

Table 4-59: Proposal Status

Status	Icon	Next Step	Comment	User Interaction
None	◇	Create	No proposal has assigned to the barclamp	Create a proposal for the barclamp if desired
User Input	●	Delete or Apply	Proposal waiting for user input and activation	Edit the proposal. Apply proposal after review.

Ready		Deactivate or Apply	Proposal has been deployed.	Ready for use.
Pending	blinking	Dequeue	Queued for deployment.	Needs additional resources and/or configuration.
Working	spinning	None	Proposal is being configured.	*None: Crowbar is working*
Failed	blinking	Apply	Proposal failed during Apply	Correct error and reapply proposal

From the Barclamp list, you may take actions on the proposals based on their state as shown in the table above. Please review the *Life Cycle* section for more information about the different proposal states.

All core barclamps automatically create "*default*" proposals and do not allow users to create additional proposal. Other barclamps allow the creation of multiple proposals. These additional proposals can be used to manage deployment configurations or control which parts of the system are active in which barclamps.

To create a new proposal, expand the barclamp row to expose the create form for new proposal. You must supply a name for the proposal but descriptions are optional. Clicking create will take you to the proposal editor (*details below*). The create form will not be shown for barclamps that only allow a single proposal.

Note: Naming for proposals is limited to lowercase letters and numbers only (not "spaces). Capitalization is automatic for formatting only.

> This limitation is necessary because activated proposals are created as roles in Chef and follow a prescribed naming convention.
>
> Crowbar stores barclamps in the Chef under the Crowbar data bag using *bc-template-[barclamp]* as the naming pattern. When a proposal is created, the instance copy is also stored in the Crowbar data bag. Only active proposals have roles created for them.

Proposal View/Edit

Selecting a proposal from on the list navigates to the proposal details page.

> **Note:** If a proposal is active, you will be initially taken to a read only view of the proposal. It is acceptable to edit an active proposal and re-apply. To access the proposal editor from the read only view, click the edit button.

Selecting the name of the proposal on the list opens the Edit Proposal page. All proposals have two primary edit areas: Attributes (yellow in figure) and Deployment. Attributes are configurable data that is used by the Chef recipes. Deployment is the Chef roles and nodes assigned to those roles.

Since each barclamp has unique attributes and roles you should consult the documentation for each barclamp if you plan to change its defaults.

Each barclamp may provide a custom editor for its attributes and node deployment information. The typical custom editor lets you set attribute values using a form and drag and drop nodes from the available list (left column) into the roles associated with the barclamp (right, red on figure). Each barclamp may have specific logic that requires minimum or maximums for node assignments.

Note: While most barclamps coordinate with Chef to perform node deployments, Crowbar includes some special function barclamps that can be used to change how Crowbar operates.

If the barclamp does not have customer editor or your browser does not support the editor, Crowbar automatically uses a raw JSON editor. You can also use this view if you want to see the entire configuration details. Selecting the Raw view option on the right side of the Attributes or Deployment panel opens the JSON editor for that section. This option lets you directly edit the JSON configuration details for the proposal. This option is typically used when developing new barclamps or for advanced users only.

When you have finished editing the proposal, you may save or apply it. Save retains your configuration settings. Apply save and then applies your proposal so that Crowbar to begin deploying the barclamp on the selected nodes. Deleting a proposal removes it from the system and you lose your configuration.

Note: If you attempt to apply a proposal that does not have sufficient nodes then Crowbar shows that proposal as queued. This is likely to happen if you select nodes that have not been allocated.

When you apply a proposal, Crowbar creates Chef roles, and then puts them into the run list of the selected nodes.

Crowbar uses a naming pattern for Roles that let you quickly figure out which barclamp and proposal is being applied to a node's run list in Chef. The instantiated barclamp naming pattern is [barclamp]-config-[proposal]. Barclamps then use additional roles to control node proposal membership (aka the Run List)

Deployment Functions section has a much more detailed discussion of Barclamps and Roles.

Note: When a role or barclamp is selected in the details panel, the nodes that share the same barclamp or role are highlighted in the group panel. This helps quickly identify groups of nodes that are similarly configured.

The buttons on the top of the details panel (Identify, Power On, Shutdown and Reboot) use the node's IPMI interface to change the node's physical state. These states cause the node status to be unknown. These buttons are only available if the system is able to successfully configure the BMC on the target system.

Table 2-3: Details Panel Buttons

Button	Action	Useful When
Identify	Causes the identify light to blink for 15 seconds.	Trying to identify a node within a rack.
Power On	Sends a power on signal to the BMC of the selected system.	Remotely powering on a system.
Shutdown	Sends a power off signal to the BMC of the selected system.	Remotely powering on a system.
Reboot	Sends a power cycle signal to the BMC of the selected system.	Remotely power cycling a system, which has stopped responding.

The buttons on the bottom of the details panel (Delete, Reset, Reinstall and Hardware Update) reset the node's deployment state. These functions are very useful during lab configurations when the system is being continuously reconfigured. The buttons take the following actions:

Table 2-4: Buttons on Bottom of Details Panel

Button	Action	Config Lost?	Reboot?	Useful When

Delete	Completely removes all records of the node from the Crowbar/Chef database. If a node is deleted, it is rediscovered if it reboots.	Yes	No	Removing nodes.
Reset	Removes all the roles assigned to the node and reimage it back to an unassigned node.	Yes	Yes	Reallocate the node for a new purpose.
Reinstall	Reimages the node and then reapply the current deployment profile to the node. This effectively rebuilds the server back to a pristine state.	No	Yes	Tuning the Chef recipes or configuration details.
Hardware Update	Keeps the current configuration, but forces the node to reboot.	Yes	Yes	To apply BIOS or RAID updates.

Using the Edit link (after the node name in the top left) lets you make per node decisions about how the node is deployed.

Note: You **must** allocate the node to let Crowbar complete the deployment group panel. The allocate step acts as a pause state for deployment so that you have time choose a node's role in the system before Crowbar provisions it. In addition, use it to simulate white listing.

To Allocate a node, manually allocate the node from the edit page or you may include that node in an applied barclamp proposal.

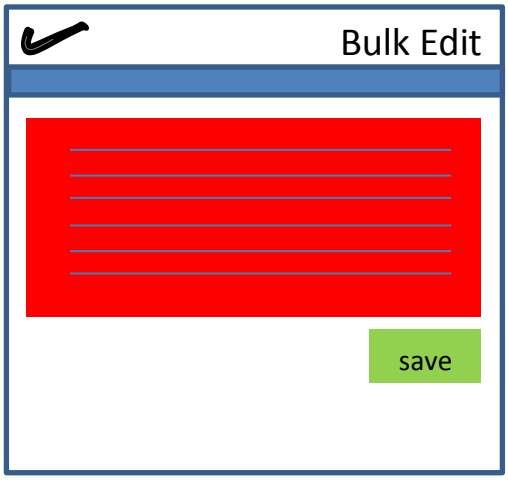
Bulk Edit

The Build Edit screen (added in v1.2) allows you to quickly update the description, RAID, BIOS, and allocation state for all the nodes in the system.

Note: If you allocate a node(s) then Crowbar will immediately begin to deploy it (them). Allocation is a one way process. You cannot "unallocate" a node.

The RAID and BIOS selections highlight the nodes' current value using [brackets]. The choices offered for BIOS and RAID selectors are determined by the Bios and Raid Barclamps. They can be expanded after installation.

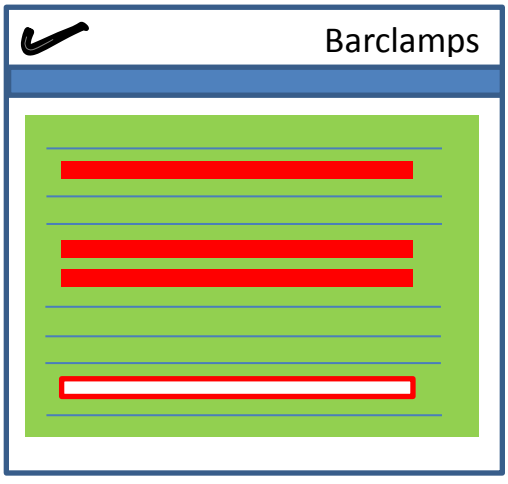
If no change it made then the node will not be updated.



Barclamps

The Barclamps page lets you create, edit, review and deploy proposals for Barclamps. These activities are the way that Crowbar decides which nodes to deploy and how to configure them.

Note: Previous versions of Crowbar split the Barclamp lifecycle into multiple pages. All three phases of the lifecycle (barclamp, proposal, active role) are now represented together on the Barclamps page.



Barclamp List

The Barclamps page shows a list of all available barclamps (see "Included Barclamps" table). The barclamps are represented as the blue lines in the figure to the right. Expanding a barclamp (by clicking ▷) displays the associated proposals for the selected barclamp (red box in the figure). You jump directly to the relevant proposal by clicking on its name under its barclamp.

A barclamp will show the status of the proposals that are attached to it using a status light (see table below). If multiple proposals are assigned, then multiple light are displayed. If there are no proposals, a diamond is displayed. Hovering over the light will show you the name and status of the matching proposal. The proposals status updates automatically without a refresh.

Table 4-59: Proposal Status

Status	Icon	Next Step	Comment	User Interaction
None	◇	Create	No proposal has assigned to the barclamp	Create a proposal for the barclamp if desired
User Input		Delete or Apply	Proposal waiting for user input and activation	Edit the proposal. Apply proposal after review.
Ready		Deactivate or Apply	Proposal has been deployed.	Ready for use.
Pending	blinking	Dequeue	Queued for deployment.	Needs additional resources and/or configuration.
Working	spinning	None	Proposal is being configured.	*None: Crowbar is working*
Failed	blinking	Apply	Proposal failed during Apply	Correct error and reapply proposal

From the Barclamp list, you may take actions on the proposals based on their state as shown in the table above. Please review the *Life Cycle* section for more information about the different proposal states.

All core barclamps automatically create "*default*" proposals and do not allow users to create additional proposal. Other barclamps allow the creation of multiple proposals. These additional proposals can be used to manage deployment configurations or control which parts of the system are active in which barclamps.

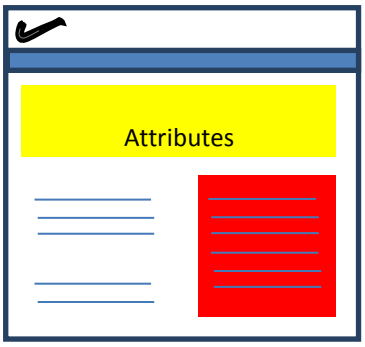
To create a new proposal, expand the barclamp row to expose the create form for new proposal. You must supply a name for the proposal but descriptions are optional. Clicking create will take you to the proposal editor (*details below*). The create form will not be shown for barclamps that only allow a single proposal.

Note: Naming for proposals is limited to lowercase letters and numbers only (not "spaces). Capitalization is automatic for formatting only.

This limitation is necessary because activated proposals are created as roles in Chef and follow a prescribed naming convention.

Crowbar stores barclamps in the Chef under the Crowbar data bag using *bc-template-[barclamp]* as the naming pattern. When a proposal is created, the instance copy is also stored in the Crowbar data bag. Only active proposals have roles created for them.

Proposal View/Edit



Selecting a proposal from on the list navigates to the proposal details page.

Note: If a proposal is active, you will be initially taken to a read only view of the proposal. It is acceptable to edit an active proposal and re-apply. To access the proposal editor from the read only view, click the edit button.

Selecting the name of the proposal on the list opens the Edit Proposal page. All proposals have two primary edit areas: Attributes (yellow in figure) and Deployment. Attributes are configurable data that is used by the Chef recipes. Deployment is the Chef roles and nodes assigned to those roles.

Since each barclamp has unique attributes and roles you should consult the documentation for each barclamp if you plan to change its defaults.

Each barclamp may provide a custom editor for its attributes and node deployment information. The typical custom editor lets you set attribute values using a form and drag and drop nodes from the available list (left column) into the roles associated with the barclamp (right, red on figure). Each barclamp may have specific logic that requires minimum or maximums for node assignments.

Note: While most barclamps coordinate with Chef to perform node deployments, Crowbar includes some special function barclamps that can be used to change how Crowbar operates.

If the barclamp does not have customer editor or your browser does not support the editor, Crowbar automatically uses a raw JSON editor. You can also use this view if you want to see the entire configuration details. Selecting the Raw view option on the right side of the Attributes or Deployment panel opens the JSON editor for that section. This option lets you directly edit the JSON configuration details for the proposal. This option is typically used when developing new barclamps or for advanced users only.

When you have finished editing the proposal, you may save or apply it. Save retains your configuration settings. Apply save and then applies your proposal so that Crowbar to begin deploying the barclamp on the selected nodes. Deleting a proposal removes it from the system and you lose your configuration.

Note: If you attempt to apply a proposal that does not have sufficient nodes then Crowbar shows that proposal as queued. This is likely to happen if you select nodes that have not been allocated.

When you apply a proposal, Crowbar creates Chef roles, and then puts them into the run list of the selected nodes.

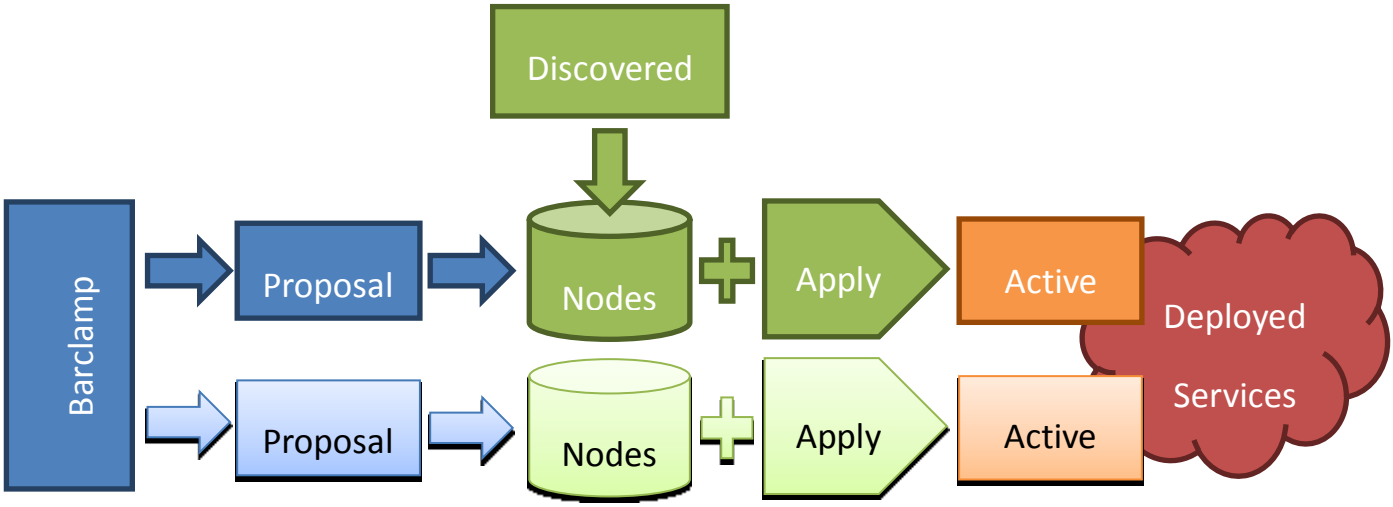
Crowbar uses a naming pattern for Roles that let you quickly figure out which barclamp and proposal is being applied to a node's run list in Chef. The instantiated barclamp naming pattern is [barclamp]-config-[proposal]. Barclamps then use additional roles to control node proposal membership (aka the Run List)

Deployment Functions

To allow control of the process, Crowbar breaks deployment into phases. You must activate and allocate nodes before Crowbar implements optional services.

Life Cycle

Understanding the Barclamp life cycle is essential to understanding Crowbar interface layout.



Crowbar progresses all deployments through a fixed lifecycle. It is important to understand this lifecycle to use Crowbar.

Figure 2 Life Cycle of a Barclamp from Concept to Proposal and Deployment.

Figure 2 shows the entirety of a barclamp within the Crowbar user interface. A Barclamp defines the capability for a service but cannot be deployed. To deploy a barclamp, you must create a Proposal. Once the Proposal is created, you must select nodes to operate on. As discussed in the next sections, you may also edit the Proposal's attributes as needed.

Applying the Proposal tells Crowbar to deploy the proposal onto the nodes. While deploying, nodes return to the Ready state when deployment is completed. Once a proposal has become an Active Role, you cannot edit it. You must delete the Role and repeat the Apply process.

Note: Many barclamps, require multiple nodes before they can be deployed. A proposal is not deploy until it has sufficient capacity.

You must edit a barclamp in the Crowbar data bag in Chef to change the attribute defaults of a barclamp when not using the Crowbar UI.

At the time a proposal is applied, Crowbar updates the Run List of the selected nodes.

The following table shows options for changes to proposals based on their current state

State	Icon	Forwards	Backwards
None	◇	Create	None
User Input	●	Apply	Delete

Ready		Apply	Deactivate
Pending	blinking	None: Proposal Queued	Dequeue
Not Ready	spinning	None: System Working	None: System Working
Failed	blinking	Apply (after fixing issue)	Delete

PXE State Machine

While Crowbar brings systems up the systems run through a series of states each of which preforms different actions on the systems. This is controlled by the Provisioner barclamp and the status of any of the systems is indicated by the icon next to the representation of the system on the Dashboard page. You can see a description of the state which a system is in by hovering over the status icon.

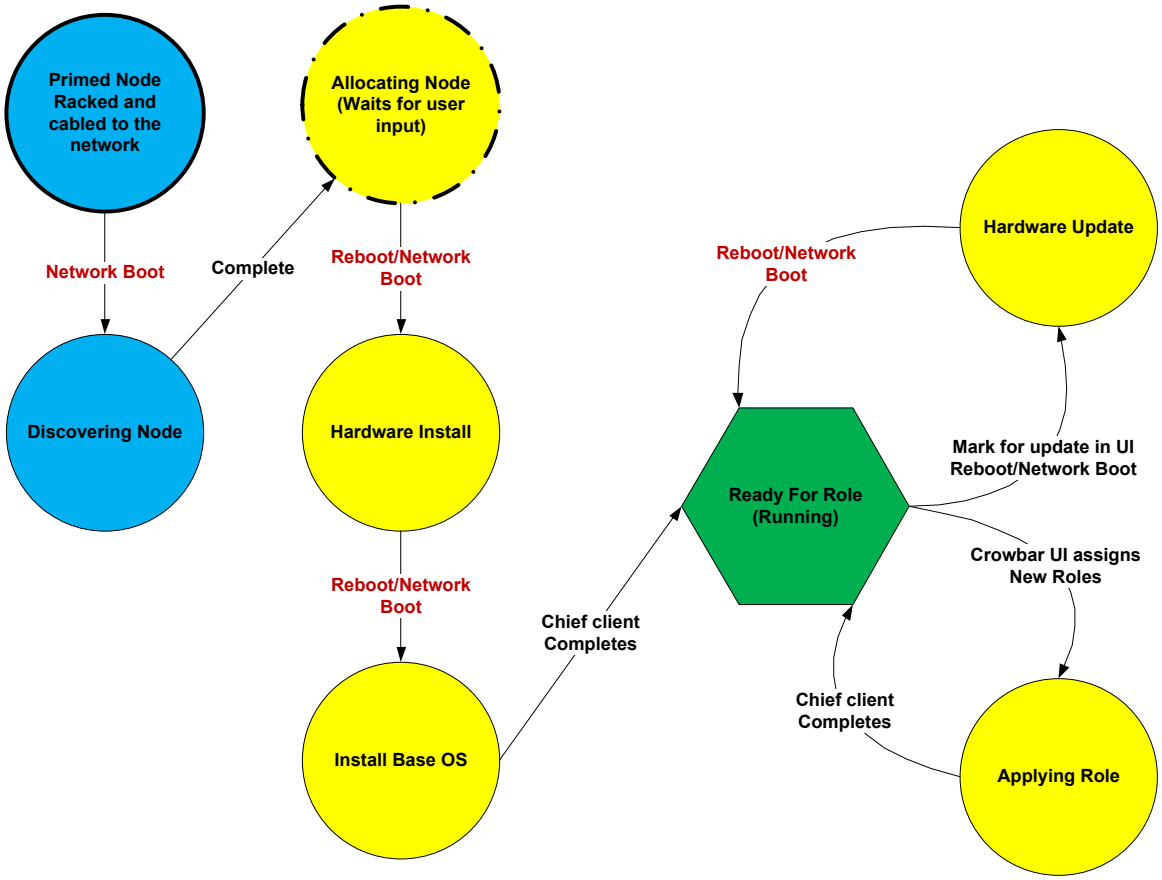


Figure 2 shows the states a system progresses through to fully provision a system

Discovering Node

During the Discovering Node state, the node network boots a CentOS LiveCD image to make sure that the node can run some basic components. Once this is complete, the LiveCD image registers with the Admin node to declare its state. The admin node adds the machine to the Chef database, allocates a BMC and Admin network address for the node.

Allocating Node

By default the nodes pause after the discovered state. The node awaits allocation either manually through the API (CLI or UI) or the being included in a proposal that is committed to the system. Since the node is not in the ready state, the proposal is queue until the newly allocated node achieves is ready. Allocation is required to allow the system to use what the node is going to become to influence the configuration of the bios and raid systems.

Hardware Install

The LiveCD image reboots forcing another network boot. During the Hardware Install state, the node network boots a CentOS LiveCD image to make sure that the BIOS/BMC/RAID controller are up-to-date. The CentOS LiveCD also update the BIOS and RAID configurations. Once this is complete, the LiveCD image registers with the Admin node to declare its state. The admin node updates the Chef database and resets the DHCP entry to installation state.

Base OS Install

LiveCD image reboots forcing another network boot. The node boots into a network installation of Ubuntu. The installation process reboots the node. Upon reboot and completing the one-time setup script, the system transitions to the Ready for Role state. The first time chef-client is run, the node gets a base set of configuration (nagios client, ganglia client, ntp client, dns resolver).

Ready for Role

This is the default state for normal operation. The node runs the chef client periodically to ensure that it is up to date. It is waiting for changes to its configuration. If the node reboots, the node network boots the local boot image again and return to this state.

Hardware Update

To transition to this state, you mark the node as needing hardware updates in the Crowbar UI and then reboot the node. The node network boots and does the same actions as the Hardware Install state. Once this is complete, the liveCD image registers with the Admin node to declare its state. The admin node updates the Chef database and resets the DHCP entry to installation state. The return state is back to the Ready for Role state. This state allows for the update of BIOS without having to re-install the system.

Applying Role

This is a transient state that represents the running chef-client. It is the time that the node is applying new configuration before returning to Ready for Role. The transition to this state happens periodically or can be forced by the admin node to run chef-client.

Included Barclamps

Table 4-1: Barclamps

Barclamp	Function / Comments
Crowbar	The roles and recipes to set up the barclamp framework. References other barclamps. Modify the default proposal to change the Usernames and passwords for access to the Crowbar UI.
Deployer	Initial classification system for the Crowbar environment (aka the state machine)
Provisioner	The roles and recipes to set up the provisioning server and a base environment for all nodes
Network	Instantiates network interfaces on the Crowbar managed systems. Also manages the address pool.
RAID	Sets up LSI RAID controllers in a variety of configurations. If missing, you can performed it manually.
BIOS	Configures BIOS options for Dell PowerEdge C servers. If missing, can be performed manually.
IPMI	Allows management of the IP Management Interface (IPMI) on servers when the BMC network is enabled.
NTP	Common NTP service for the cluster (required for secure access). An NTP server can be specified.
DNS	Manages the DNS subsystem for the cluster
Logging	Centralized logging system based on syslog
Nagios	System monitoring service for the cluster that can be used by other barclamps
Ganglia	Performance monitoring service for the cluster that can be used by other barclamps
Test	Provides a shell for writing tests against

Details about individual barclamps are including in the Barclamps section below.

Barclamp Details

Crowbar Barclamp

The Crowbar Barclamp provides the roles and recipes to set up the barclamp framework.

The Crowbar Barclamp initializes the system, creates initial instances of other barclamps defined in its configuration, and creates the users to access the Crowbar API and UI. Any number of barclamp instances can be started. By default, the system starts a network, ganglia, nagios, ntp, dns, provisioner, deployer, ipmi, raid, and BIOS barclamp based upon their default configurations. The initialization function of the Crowbar barclamp works exactly like the other barclamps. A proposal is created and can be committed during installation.

The main post-installation function is to provide the main transition entry point for the system. All barclamps' transition functions can be called directly, but the Crowbar barclamp calls these in an order specified in its configuration which is determined by their priority. The default unspecified priority is 100. The special cases are the Provisioner, which is last, and the Deployer and network, which are first and second.

Table 4-2: Crowbar Barclamp Parameters

Name	Default	Description
barclamps	A list of all barclamps we ship.	A list of supported barclamps. This is not used in the product currently.
instances	The starting barclamps using their default configurations.	A map of barclamp names that reference a list of json files (default is special to mean to take the defaults) that represent starting barclamp instances to create.
run_order	A map of barclamp names to integer priorities.	The map is used to define the order that the barclamps are called when handling a transition request. lower is higher priority. Unspecified items are assumed to have a priority of 100.
users	A map of users - containing Crowbar.	This map defines the users allowed to access Crowbar's UI and REST API.

The users map contains a map. The key is the user name and the rest of the required fields are:

Table 4-3: User Name Key

Name	Description
password	Clear text password of the user
description	A description of the user.

Deployer Barclamp

The Deployer provides an initial classification system for the Crowbar environment. As nodes are discovered, the Deployer makes sure that discovery tools are run on the node by making sure that the Deployer-client role is assigned to the node, the results of that discovery are classified, and the node's attributes are updated to reflect its potential usage. The Deployer also builds a map of valid and usable disks.

The Deployer gives the primary name to the node at the discovered state. The names default to the letter 'd' and the mac address (with dashes instead of colons). The Deplorer also allocates the admin and BMC addresses from the network barclamp.

In addition, the Deployer defines and provides the node's configuration for raid and bios. These values are assigned part of the hardware-installing state transition. The Deployer uses a list of role name patterns that define what the raid and

bios configurations should be. These are applied as values in the node attributes under crowbar -> hardware. bios_set can in the set of Virtualization or Storage. raid_set can be in the set of JBODOnly and SingleRaid10.

The Deployer is also responsible for manipulating the run-list during the hardware-installing and update (or hardware-updating) states. The run list should only include bios, raid, and ipmi operations.

The Deployer also controls the allocate flag on the node. The allocate flag is used to pause the node during after discovery. The node waits for it to be allocated to contain. The Deployer has a configuration option to indicate if the allocate flag should be set to false (and cause a pause) or just allocate all nodes.

Table 4-4: Deployer Barclamp Parameters

Name	Default	Description
bios_map	A list of default settings for bios and raid for swift and nova.	The map defines a list of patterns that would apply a configuration setting for bios and raid.
use_allocate	true	A Boolean value - true indicates that a pause should be injected after the discovered state to allow the admin to accept and allocate the node.

Table 4-5: BIOS Map Entry Keys

Name	Description
pattern	Regular expression applied to the role names on the node.
bios_set	The bios set of parameters to apply. Values are: Virtualization or Storage.
raid_set	The raid set of parameters to apply. Values are: JBODOnly or SingeRaid10.

Provisioner Barclamp

The Provisioner provides the roles and recipes to set up the provisioning server and a base environment for all provisioned nodes. The Provisioner also provides the transition entry point for nodes that need to have DHCP transitions done. The Provisioner assumes that addressing is handled outside of this barclamp.

Table 4-6: Provisioner Barclamp Parameters

Name	Default	Description
default_user	crowbar	User to create for external login.
default_password	unset	Clear text password to use for external login.
default_password_hash	Hash of crowbar	MD5 hash of password to use for external login. printf 'password' | mkpassed -s -m md5

		will generate the hash.
web_port	8091	The default Web port that the repository web server uses.
use_local_security	true	This defaults the security updates path in the install to use the admin node instead of the internet.
dhcp	map	This is a map that contains the DHCP parameters (lease-time and state_machine).
lease-time	60	The number of seconds a DHCP lease is valid for the system.
state_machine	map	This is the state machine that DHCP server uses in this instance of the barclamp.

Note:While neither is required, one of default_password or default_password_hash is required.

Network Barclamp

The Network barclamp provides two functions for the system. The first is a common role to instantiate network interfaces on the Crowbar managed systems. The other function is address pool management.

The network interfaces are controlled by the network role that is applied by the barclamp as a node transition to "installed". Based upon assigned addresses, the network recipe creates the appropriate single, dual, or team mode interface sets.

The network assignment function is handled by the creation of an API extension of the base barclamp. The barclamp adds the allocate_ip REST API call. This function allocates an IP address from a requested network and updates the node's attributes and the network's data bag. The available networks (and their parameters) are defined in the configuration for the barclamp.

Modification of the following parameters should only be done at install time.

Table 4-7: Network Configuration Options

Name	Default	Description
mode	single	A string value of either single, dual, or team. This specifies the default network interface construction model.
teaming	map	A map of values specific to teaming.
networks	map	A map of networks that this barclamp should manage.

Table 4-8: Teaming Sub-Parameters

Name	Default	Description
mode	6	The default teaming algorithm to use for the bonding driver in Linux.

Table 4-9: Default Networks

Name	Usage	Notes
admin	Private network for node to node communication	A router, if wanted, is external to the system. This network must be owned by the Crowbar system to run DHCP.
bmc	Private network for bmc communication	This can be the same as the admin network by using the ranges to limit what IP goes where. A router, if wanted, is external to the system.
bmc_vlan	Private network for admin nodes on the bmc network	This must be the same as the BMC network and have the same VLAN. This is used to generate a vlan tagged interface on the admin nodes that can access the BMC LAN.
storage	Private network for storage traffic	A router, if wanted, is external to the system.
public	Public network for Crowbar and other components	A router, if wanted, is external to the system.

Table 4-10: Network Parameters

Name	Default	Description
vlan	Integer	The VLAN to use on the switch and interfaces for this network
use_vlan	true	A value of true indicates that the VLAN should apply to the interface. A value of false assumes that the node receives untagged traffic for this network.
add_bridge	false	Indicates if the network should have a bridge built on top of it. The bridge will be br. This is mostly for Nova compute.
subnet	IP Address	The subnet for this network.
netmask	Netmask	The netmask for this network.
router	IP Address	The default router for this network.
broadcast	IP Address	The default broadcast address for this network.
ranges	map	This contains a map of strings to start and stop values for network. This allows for sub-ranges with the network for specific uses. For example: DHCP, admin, BMC, hosts.

Table 4-11: Range Map String Key

Name	Type	Description
start	IP Address	First address in the range, inclusive.
end	IP Address	Last address in the range, inclusive.

△ Settings in the Network barclamp should not be changed after the installation of the Admin Node.

RAID Barclamp

RAID = Redundant Array Of independent Disks. This means that a RAID controller makes (or can) multiple disks look like 1 big/smart/safe disk.

The RAID controller can support up to 2 separate RAID volumes of RAID0, RAID1, RAID1E, or RAID10. Any disk not included in a RAID volume, are directly exposed to the Operating System (also known as JBOD = just a bunch of disks).

The Crowbar code makes sure that the configuration on the RAID controller matches that specified using the Crowbar configuration.

The parts that determine the configuration for a node are:

- A set of chef data bags, which contain the RAID configuration (in data bags/crowbar-data). The defaults are SingleRaid10 and JBODOnly.
- An attribute on the chef node of the machine which identifies which data bag (described above) should be applied to this node (the attribute is node[:crowbar][:hardware][:raid_set], and it should include the name of a data bag).
- Crowbar (the Deployer barclamp) sets the above property when a node is allocated to a proposal.

When invoked, the recipe uses a LWRP to inspect the current configuration on the system, and compare it to the desired state. If the two diverge, the code will:

- Delete any raid-sets that are not required any more
- Allocate available disks among the desired raid sets, according to the order attribute.
- Issue commands to apply the configuration.

BIOS Barclamp

The BIOS barclamp provides the following specific control features:

- Uploading a known BIOS firmware into flash. Setting parameters to defined a set, based on the machine's role.
- IPMI Barclamp
- LAN parameters (IP address, netmask gateway) and User credentials.
- The IPMI Barclamp has a couple of list parameters.

IPMI Barclamp

The IPMI barclamp configures IPMI access on platforms that support it. Specifically it configures:

Table 4-12: IPMI Barclamp Parameters

Name	Default	Description
bmc_enable	true	Controls if the barclamp attempts to work on the BMC.
bmc_password	crowbar	The password which will be configured on the BMC
bmc_user	crowbar	The username which will be configured on the BMC

debug	true	turns on more verbose output.

NTP Barclamp

The NTP Barclamp provides a common NTP service for the cluster. You can specify an NTP server or servers and all other nodes are clients of them.

Table 4-13: NTP Barclamp Parameters

Name	Default	Description
external_servers	empty list	A list of IP addresses or hostnames that should be used as external NTP servers. Hostname can be used if the DNS barclamp is configured to have access to an external resolver.
admin_ip_eval	"Chef::Recipe::Barclamp::Inventory.get_network_by_type(node, \"admin\").address"	The ruby eval expression that returns the admin IP address of a node.

Note: If you are setting up an external server it can take up to 5 minutes for the nodes to sync with the server. Systems should not be rebooted during this process. If systems are rebooted they pause as the sync occurs.

Logging Barclamp

The Logging Barclamp provides a centralized logging system based on syslog. The barclamp enables a centralized log server that can then forward information to external syslog servers. The Crowbar install process sends logs to the admin node by default, but the configuration from the logging barclamp overrides this initial configuration.

Table 4-14: Logging Barclamp Configuration Parameters

Name	Default	Description
External_servers	Empty list	A list of IP addresses for the logging-server to which to forward logs.

Nagios Barclamp

The Nagios barclamp provides a common Nagios service for the cluster. A Nagios server or servers can be specified and all other nodes are clients of them. The barclamp attempts to direct all traffic over the admin network.

Table 4-56. Nagios Barclamp Parameters

Name	Default	Description
admin_interface_eval	Chef::Recipe::Barclamp::Inventory.get_network_by_type(node, \"admin\").interface	The ruby eval expression that returns the admin interface of the node.
admin_ip_eval	"Chef::Recipe::Barclamp::Inventory.get_network_by_type(node, \"admin\").address"	The ruby eval expression that returns the admin IP address of a node.

Ganglia Barclamp

The Ganglia barclamp provides a common Ganglia service for the cluster. Ganglia server or servers can be specified and all other nodes are clients of them.

Table 4-57: Ganglia Barclamp Parameters.

Name	Default	Description
interface_eval	Chef::Recipe::Barclamp::Inventory.get_network_by_type(node, \"admin\").interface	The ruby evaluation string that gets the interface of the admin interface.

Test Barclamp

The Test barclamp provides a shell for writing tests against. It allows for failures to be injected and other barclamps can be validated against it.

Table 4-58: Test Barclamp Parameters.

Name	Description
barclamps	A list of supported barclamps that are used as the return value for the barclamp list API call
instances	A map of barclamp names that reference a list of json files (default is special to mean to take the defaults) that represent starting barclamp instances to create

Supplemental Material

System Verification

As a final step, it is important to verify that your deployment has succeeded. Crowbar does *not* provide specific feedback or updates to confirm that the Chef recipes were successfully deployed.

You should consult the getting started guide and barclamps specific to your system for details on verification of deployment.

Managing Barclamps

This section briefly describes barclamps, and how to import barclamps.

Note: For information about creating barclamps, please visit https://github.com/dellcloudedge/crowbar/wiki/Barclamp:-create-&-install-steps

Introduction

A barclamp is a deployment module that is imported from its own code repository into the Crowbar framework. A barclamp cannot operate without Crowbar, but you do not have to create a unique build of Crowbar in order to create a barclamp.

Note: You must install Crowbar before importing barclamps.

Importing a Barclamp

Once you created a barclamp, you can import the barclamp into Crowbar & Chef. Assuming that you already created the foo barclamp in /barclamps (see Creating a Barclamp), proceed as follows:

1. From the Crowbar server, become the super admin: sudo –i
2. Run the barclamp install script: /opt/dell/bin/barclamp_install /barclamps/foo.tar.gz
    a. "/barclamps/foo.tar.gz" is the file name of your barclamp. It could be anything.
    b. The core barclamps are in /opt/dell/barclamps.

Your barclamp should now appear in the Crowbar UI. You can also see it in Chef under the Crowbar data bag.

While barclamps are generally safe to install multiple times, you can uninstall a barclamp using "barclamp_uninstall.rb /path/to/barclamp".

More Information

Information on how to develop your own barclamp is available in the *Developers Guide*. If you are interested in creating, extending or contributing barclamps, please use one of the following contact methods.

Support

Crowbar Support

To obtain support for Crowbar:

- Gathering Log Information
- Email the Crowbar listserv: (join at https://lists.us.dell.com/mailman/listinfo/crowbar)

To help facilitate troubleshooting of the environment a utility to gather logs has been provided. Browse to http://<Admin_ip>:3000/support/logs. This creates a tar archive of the relevant logs and asks the user for a location to save the resulting archive.

Note: Depending on the size of the logs to be gathered this utility may take a while to run.

www.dell.com