Dell Cloud Solution for
OpenStack™ Solutions

OpenStack Barclamps Users Guide

Crowbar Version 1.2

Notes, Cautions, and Warnings

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

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.

Contents

1 Introduction 5
1.1 Concepts 5
1.2 OpenStack 5
1.3 Opscode Chef Server 5
1.4 Dell Specific Options 6
2 The Crowbar Solution ..... **Error! Bookmark not defined.**
2.1 Architecture ..... **Error! Bookmark not defined.**
2.2 System end state ..... **Error! Bookmark not defined.**
- Node provisioning ..... **Error! Bookmark not defined.**
- Networking administration ..... **Error! Bookmark not defined.**
- NTP ..... **Error! Bookmark not defined.**
- DNS ..... **Error! Bookmark not defined.**
- Nagios ..... **Error! Bookmark not defined.**
- Ganglia ..... **Error! Bookmark not defined.**
- Logging ..... **Error! Bookmark not defined.**

2.3 Network setup ..... **Error! Bookmark not defined.**
3 User Interface ..... **Error! Bookmark not defined.**
3.1 Using Crowbar ..... **Error! Bookmark not defined.**
3.2 Nodes ..... **Error! Bookmark not defined.**
3.3 Barclamps ..... **Error! Bookmark not defined.**
3.4 General Layout ..... **Error! Bookmark not defined.**
3.5 Dashboard ..... **Error! Bookmark not defined.**
3.6 Node Switch Groups ..... **Error! Bookmark not defined.**
3.7 Node Details ..... **Error! Bookmark not defined.**
4 Overview 6
5 Deployment Functions ..... **Error! Bookmark not defined.**
5.1 Life Cycle ..... **Error! Bookmark not defined.**
5.2 PXE State Machine ..... **Error! Bookmark not defined.**
5.3 Barclamps 6
5.4 Crowbar Barclamp ..... **Error! Bookmark not defined.**

5.5 Deployer Barclamp ..... **Error! Bookmark not defined.**
5.6 Provisioner Barclamp ..... **Error! Bookmark not defined.**
5.7 Network Barclamp ..... **Error! Bookmark not defined.**
5.8 RAID Barclamp ..... **Error! Bookmark not defined.**
5.9 BIOS Barclamp ..... **Error! Bookmark not defined.**
5.10 IPMI Barclamp ..... **Error! Bookmark not defined.**
5.11 NTP Barclamp ..... **Error! Bookmark not defined.**
5.12 Logging Barclamp ..... **Error! Bookmark not defined.**
5.13 Nova Barclamp 7
5.14 Swift Barclamp 10
5.15 Glance Barclamp 12
5.16 Nagios Barclamp ..... **Error! Bookmark not defined.**
5.17 Ganglia Barclamp ..... **Error! Bookmark not defined.**
5.18 Test Barclamp ..... **Error! Bookmark not defined.**
5.19 Proposals ..... **Error! Bookmark not defined.**
5.20 Active Roles ..... **Error! Bookmark not defined.**
6 Supplemental Material ..... **Error! Bookmark not defined.**
6.1 System Verification ..... **Error! Bookmark not defined.**
6.2 Deactivating Barclamps ..... **Error! Bookmark not defined.**
6.3 Creating Barclamps ..... **Error! Bookmark not defined.**
6.4 Gathering Log Information ..... **Error! Bookmark not defined.**

1 Introduction

This document provides instructions you to use when deploying **OpenStack** components on Crowbar 1.2. This guide is for use with the Crowbar Users Guide, it is not a stand-alone document.

Other suggested materials:

- OpenStack Reference Architecture (RA, July 2011)
- Crowbar Getting Started Guide (July 2011)
- Bootstrapping OpenStack Clouds (Dell Tech White Paper , Feb 2011)
- CloudOps White Paper (Dell Tech White Paper, Oct 2011)

1.1 Concepts

The purpose of this guide is to explain the special aspects of OpenStack on the Crowbar. Please consult the users' guide and getting started guide for assistance with installing Crowbar and configuring the target system.

Concepts beyond the scope of this guide will be introduced as needed in notes and references to other documentation..

1.2 OpenStack

The focus of this guide is the use of Crowbar, *not* OpenStack. While Crowbar includes substantial components to assist in the deployment of OpenStack, its operational aspects are completely independent of OpenStack.

For detailed operational support for OpenStack, we suggest visiting the OpenStack documentation web site at http://docs.openstack.org/.

This guide will provide this additional information about OpenStack as notes flagged with the OpenStack logo.

1.3 Opscode Chef Server

Crowbar makes extensive use of Opscode Chef Server**,** http://opscode.com. To explain Crowbar actions, it is helpful (but not required) to 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 will be including using this formation.

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

1.4 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 will be identified.

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

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

2 Overview

This user screen is available in the Dell version.

The Overview page shows a reference taxonomy for a Cactus version OpenStack deployment. Crowbar highlights the sections of the taxonomy that have been enabled in the system. Like most Crowbar pages, this page updates automatically so changes in the system status are automated reflected.

There are no user actions for the Overview page. It is a view only page.

For the API components of the taxonomy (Nova, Glance and Swift), Crowbar will provide the IP address on which these components may be accessed. This is the fastest way to determine which identify which nodes are hosting API services.

2.1 Barclamps

The Barclamp page shows a list of all available barclamps (see table below). Selecting a barclamp displays the details for the selected barclamp.

The details page shows the Proposals created and the Active Roles deployed for the barclamp. You can jump directory to the relevant proposal or role by clicking on its name.

From the barclamp details page, you may create a new proposal for the system.

Naming for proposals is limited to letters and numbers only (not spaces). Capitalization is allowed.

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

As of November 2011, the following barclamps are included with the OpenStack barclamps.

Barclamp	Function	Comments
Nova	Compute	Installs and configures the Cactus release of Openstack Nova. It relies upon the network and glance barclamps for normal operation.
Swift	Object Store	part of Openstack, and provides a distributed blob storage
Glance	Image Cache	Glance service (Nova image management) for the cloud
Dashboard	User Interface	
Keystone	Central Identity	
-		
-		

2.2 Nova Barclamp

The Nova Barclamp installs and configures the Openstack Nova component. It relies upon the network and glance barclamps for normal operation.

The Nova Proposal requires that you have previously deployed a Glance proposal. You must have at LEAST 3 NODES before you can apply the Nova proposal.

Nova Barclamp parameters. A slash indicates a hash level.

Name	Default	Description
use_barclamp	true	A placeholder value for indicating that we are in a barclamp. Lame, but hey. Will probably go away.
libvirt_type	kvm	The type of hypervisor to use. qemu is the one to use in virtualization mode. Other options are available but untested.
network_type	flat	The networking model to use for nova. Options are `flat`, `flatdhcp`, and `dhcpvlan`.
flat_network/flat_network_bridge	br500	The bridge to use for the main network in flat mode
flat_network/flat_injected	true	Should the system attempt to inject networking information into the image?
flat_dhcp_network/flat_network_bridge	br500	The bridge to use for the main network in flatdhcp mode
flat_dhcp_network/flat_network_dhcp_start	10	The starting address of the DHCP range. This is used to reserve some addresses out of the pool.
flat_dhcp_network/flat_interface	eth0.500	The interface that the flat network is really on. This may or may not be

		needed.
dhcp_vlan_network/vlan_interface	eth0	The interface that vlans should be created upon
dhcp_vlan_network/vlan_start	1000	The starting vlan ID
dhcp_vlan_network/vpn_start	10000	The starting VPN port number for accessing the project's vlan.
num_networks	1	The number of networks to cut the fixed range up into. Really only used for dhcpvlan, i think
network_size	256	Power-of-2 number that is the size of each network to give to a project. Really only used for dhcpvlan, I think
user	nova	User to run the nova services as
user_group	nova	Group gather things together under
project	admin	The default initial admin project
access_key	auto-generated	EC2 and other access key for the initial admin user
secret_key	auto-generated	EC2 and other access key for the initial admin user
user_dir	/var/lib/nova	The default user directory for the nova user.
images	two amis	A list of URLs to get images. The defaults are the ones on the admin node.
rabbit/user	nova	The user to access the rabbit MQ server
rabbit/password	auto-generated	The password to access the rabbit MQ server
rabbit/vhost	/nova	The exchange for nova messages
rabbit/port	unset	The port to use for the rabbitmq server. Unset to allow the system to choose its default.
db/user	nova	The user to access the database server
db/password	auto-generated	The password to access the database server
db/database	nova	The name of the nova database.

There are additional parameters specified in the nova attributes file that haven't been replicated in the barclamp configuration file.

- Nova Networking

This section is called out separately because of its complexity and scope.

Nova has 3 networking modes available. They are integrated with the networking barclamp modes and 10gb networking. Initially, the nova modes will be described and then the integration with the networking barclamp will be described. While the three modes are different, they use a consistent underlying networking mode.

The Nova Barclamp assumes that the Networking Barclamp is running and handling the networks. It will use the information about the topology from the networking barclamp. Nova assumes that three networks are available, admin, public, and nova_fixed_network. Currently, a nova_floating_network is defined, but not used or required. That is evolving in the community. The admin network is for service communication. Public is used for outward facing public services of Nova. Nova_fixed_network is used for the VMs. It is assumed that nova_fixed_network is a completely owned subnet. public network may be partially presented. In all cases, the nova-network node will act as the router between public and nova_fixed_network.

- Flat Network

In this mode, the nova-compute gets an address from nova-network and injects that address into the VMs image (linux-only). In our setup, this image then pulls from nova-api to get its custom configuration files (keys and stuff).

To make this work nicer, the nova-network node acts as the router between the public facing networks. This is NOT part of normal Nova Flat Network. It is part of Nova for the other modes.

The network parameters should be to chop the nova_fixed_network into a single network with all addresses available. This is specified in the num_networks and network_size parameters. The DHCP start parameter of the nova_fixed_network acts a reservation section of addresses for that range. This allows to remove shared network pieces.

This mode will use the interfaces specified by the network barclamp. By default, it will use eth0.500 for fixed network, eth0.300 for public, and eth0 for admin. Bridges will be created as appropriate. If the network mode is changed in the network barclamp, it will switch to using the teamed network or dual nic for the fixed network and public. If these are on 10g, those interfaces will be pulled as appropriate.

- Flat DHCP Network

In this mode, the nova-compute doesn't modify the VM image or allocates an address. The VM is assumed to run DHCP to get its address and then talk to nova-api for custom configuration. The nova-network node runs dnsmasq to provide DHCP to the nova_fixed_network.

The network parameters should be to chop the nova_fixed_network into a single network with all addresses available. This is specified in the num_networks and network_size parameters. The DHCP start parameter of the nova_fixed_network acts the DHCP starting address for the nova-network agent.

This mode will use the interfaces specified by the network barclamp. By default, it will use eth0.500 for fixed network, eth0.300 for public, and eth0 for admin. Bridges will be created as appropriate. If the network mode is changed in the network barclamp, it will switch to using the teamed network or dual nic for the fixed network and public. If these are on 10g, those interfaces will be pulled as appropriate.

- VLAN DHCP Network

In this mode, the nova-compute doesn't modify the VM image and uses dnsmasq to hand out addresses. There are two big differences from Flat DHCP. First, a custom VLAN is allocated for each project. The

project gets the next free vlan after the nova_fixed_network vlan and each project gets a subset of the nova_fixed_network vlan defined by the num_networks and network_size. So, if nova_fixed_network is a class B and num_networks is 1024 and network_size is 64, this will support 1024 projects. The external assumption is that the networking barclamp has setup the single, dual or teamed network and that the reserved vlans are already trunk by the switch. Default switch configs already trunk all vlans to all ports.

This mode will use the interfaces specified by the network barclamp. By default, it will use eth0.500 for fixed network, eth0.300 for public, and eth0 for admin. Bridges will be created as appropriate. If the network mode is changed in the network barclamp, it will switch to using the teamed network or dual nic for the fixed network and public. If these are on 10g, those interfaces will be pulled as appropriate. Additional VLANs will start at 501 and continue upwards.

The second big difference is the introduction of a VPN VM that is managed by nova-network to provide access to the network. The nova-network node acts as a router/firewall for the network and routes to the VPN to allow access to the VMs. The VM is controlled and managed by nova-network. It is often called cloud-pipe. The cloud-pipe image needs to be in glance and set-up in a way that openvpn configuration can be injected into it.

THIS MODE IS STILL UNDER DEVELOPMENT

2.3 Swift Barclamp

Swift is part of Openstack, and provides a distributed blob storage. This barclamp installs swift.

Swift includes the following components:

- Proxy node- provides the API to the cluster, including authentication.
- Storage nodes - provide storage for cluster.

Swift Barclamp Parameters

Name	Default	Description
cluster_hash	random generated	cluster hash is shared among all nodes in a swift cluster.can be generated using od -t x8 -N 8 -A n </dev/random
cluster_admin_pw	swauth	super user password - used for managing users.
replicas	1	how many replicas should be made for each object
zones	2	how many zones are in this cluster (should be >= # of replicas)
min_part_hours	1	minimum amount of time a partition should stay put, in

		hours
partitions	18	number of bits to represent the partitions count
user	swift	the uid to be used for swift processes
group	swift	the gid to be used for swift processes
admin_ip_expr	`"Chef::Recipe::Barclamp::Inventory .get_network_by_type(node, \"admin\").address"`	where to find IP for admin use
storage_ip_expr	`"Chef::Recipe::Barclamp::Inventory .get_network_by_type(node, \"admin\").address"`	where to find IP for admin use
disk_enum_expr	`"node[\"crowbar\"][\"disks\"]"`	expression to find a hash of possible disks to be used.
disk_test_expr	`"v[\"usage\"] == 'Storage'"`	expression accepting a k,v pair for evaluation. if expression returns true, then the disk will be used. by default, use any sdX or hdX that is not the first one (which will hold the OS).
disk_zone_assign _expr	`'$DISK_CNT | =0; $DISK_CNT= $DISK_CNT+1 ;[ $DISK_CNT % node[:swift][:zones] , 99]'`	see below

Note that the _expr parameters are meant to allow the recipes supporting the swift barclamp to be customized in their logic. Customers can embed alternative logic to e.g. disk usage decisions by the system. The same parameters allow the recipes to be utilized in a non-crowbar environment.

- Disk zone assignment expression

An expression to classify disks into zone's and assign them a weight. The expression will return either:

- nil: the disk is not included in the ring
- otherwise an array of [zone, weight]. Zone is an integer representing the zone # for the disk is expected, weight is the weight of the disk

The default expression below just assigns disks in a round robin fashion.

The expression is evaluated with the following context:

- node - the Chef node hash
- params - a hash with the following keys:
    - :ring=> one of "object", "account" or "container"
    - :disk=> disk partition information as created in disks.rb,contains: :name (e.g sdb) :size either :remaining (= all the disk) or an actual byte count.

Parameters should not be changed after committing the proposal. Addition or removal of devices from the proposal will be dynamicaly reconfigured in the swift configuration after the initial proposal has been committed.

2.4 Glance Barclamp

The glance barclamp provides the Glance service for the cloud

The Barclamp has a couple of list parameters.

Name	Default	Description
backup_type	"file"	The type of backing store to use. Options are file, s3, swift. Only supported currently is file.