Dell Cloud Solution for
OpenStack™ Solutions

OpenStack Barclamps Users Guide

OpenStack Stack Version Diablo Final

Crowbar Version 1.2

Updated 12/19/2011

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

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 (Dell Internal, RA, July 2011)
- Crowbar Getting Started Guide (July 2011)
- Bootstrapping Open Source Clouds (Dell Tech White Paper, updated Dec 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 Crowbar Users guide and Deployment 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 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 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.

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.

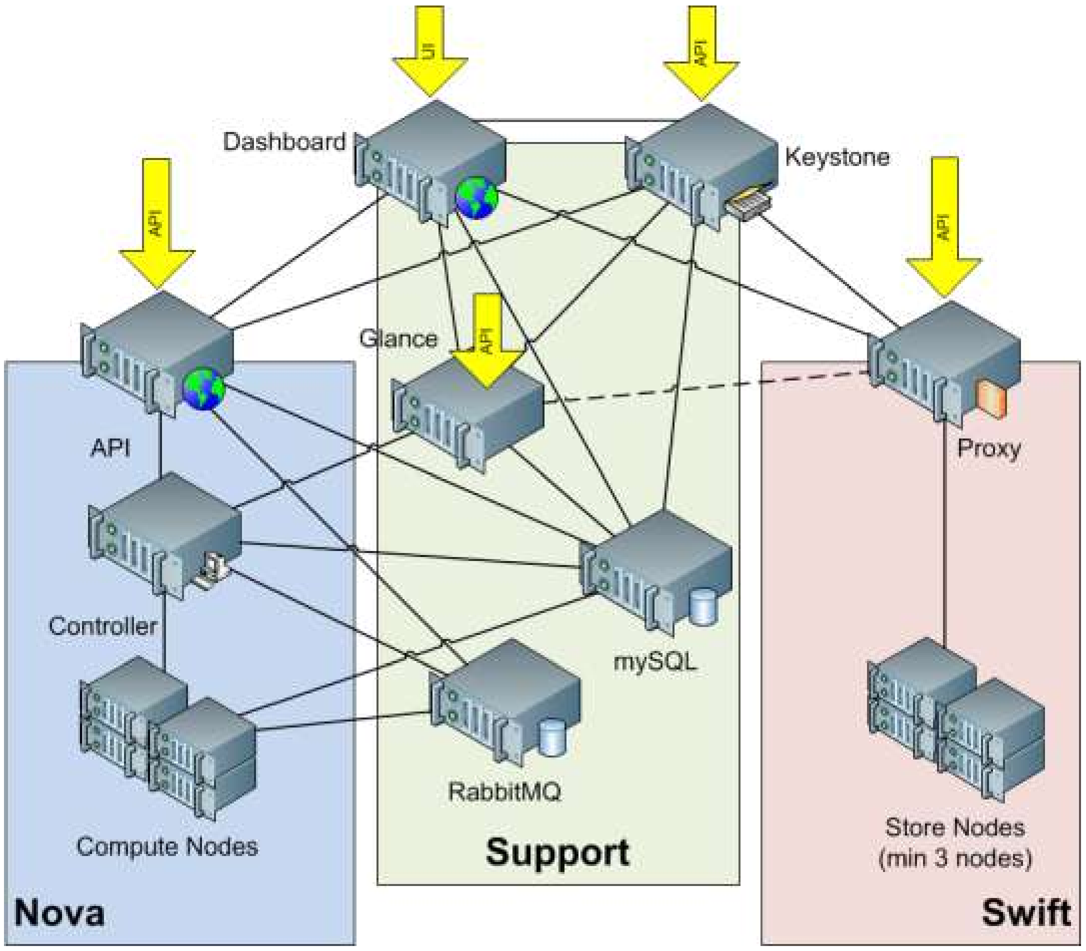
The Overview page shows an interactive reference taxonomy for a OpenStack deployment. Crowbar highlights the sections of the taxonomy that have been enabled in the system. Like most Crowbar pages, this page updates automatically so changes in the system status are automated reflected.

2 Architecture

The Crowbar OpenStack deployment includes both core and incubated OpenStack components. Crowbar deploys each component as a module, known as a barclamp. For this release, all shared components have been broken out as independent barclamps. Crowbar will automatically detect and integrate connections between barclamps as they are deployed.

It is important to deploy the barclamps in the correct order because of the dependencies between barclamps!

The figure below shows Crowbar's target OpenStack deployment with both shared and stand-alone components. Crowbar both installs the components and integrates them together as needed.



3 OpenStack Barclamp Suite

Crowbar 1.2 introduced product suite pages for barclamps. The Barclamps→OpenStack page shows only the barclamps that pertain to the OpenStack deployments.

The barclamps on this page are listed in deploy order from top (deploy first) to bottom (deploy last). This ordering is intended to aid users in performing the installation in the correct order. Not all barclamps are required; the next section explores each barclamp in detail.

Please review the barclamp use and life cycle information in the Crowbar Users Guide to learn about the status and management process for barclamps.

As a convenience, a deployment type pull down box is helps identify which barclamps should be included if the target deployment is compute (Nova/IaaS) or storage (Swift/STaaS) focused. Choosing a deployment will temporarily hide the non-applicable barclamps.

4 Barclamps

The table below shows the barclamps that are available with the Crowbar v1.2 OpenStack deployment.

From each barclamp, 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 December 2011, the following barclamps are included with the OpenStack barclamps.

Barclamp	Function	Use	Comments
MySQL	Database	All	Used by Keystone, Nova, Horizon, and Glance.
Keystone	Central Identity	All	Not core, but strongly recommended. When installed, the identify service is automatically leveraged by all other components.
Swift	Object Store	STaaS	Provides a distributed object storage
Glance	Image Cache	IaaS	Glance service (Nova image management) for the cloud. Used by Nova. Will detect & use Swift as backing store if Swift configured.
Nova Dashboard ("Horizon")	User Interface	All	Not core, but strongly recommended. Provides a web user interface and configuration capabilities for other OpenStack components.
Nova	Compute	IaaS	Supports many network modes.
Kong	Test	All	Optional. Provides a test framework for other components.

4.1 MySQL

Please see https://github.com/dellcloudedge/crowbar/wiki/Mysql-barclamp for the latest updates.

Background

MySQL, http://mysql.com/, is widely adopted an open source database that stores relational data for several OpenStack components. It was split into an independent barclamp for this Crowbar version because it is used by several other barclamps.

This barclamp will ultimately be split out as an independent barclamps.

This barclamp **can** support multiple proposals.

Swift Barclamp Parameters

Name	Default	Description
Data Directory	/var/lib/mysql	Location where database files will be stored.

Roles

MySQL has two roles: mysql-server and mysql-client. The roles are used to identify which node is the server and which are configured as client.

Barclamps that consume the MySQL barclamp will automatically update the client information. There is no need to configuration client roles at this time.

4.2 Keystone

Please see https://github.com/dellcloudedge/crowbar/wiki/Keystone-barclamp for the latest updates.

Background from ***http://openstack.org/projects/***

Identity Service provides unified authentication across all OpenStack projects and integrates with existing authentication systems.

Barclamp Parameters

Name	Default	Description
SQL Engine	MySQL	Choose mysql or sqlite as a backing store
MySQL Instance	[generated]	Select Crowbar mysql barclamp to use as a database (requires choosing MySQL)
Default Tenant	openstack	
Default Username	crowbar	
Default Password	crowbar	
Admin Username	admin	
Admin Password	crowbar	
Admin Token	[generated]	
Admin Token Expiration	[generated]	

Baclamp Roles

Keystone has one roles: keystone-server. Select which server should be the Keystone server.

The default node allocation is to use the same as the MySQL barclamp. This is not required.

4.3 Swift

Please see https://github.com/dellcloudedge/crowbar/wiki/Swift-barclamp for the latest updates.

Background from http://openstack.org/projects/storage/

OpenStack Object Storage (code-named Swift) is open source software for creating redundant, scalable object storage using clusters of standardized servers to store petabytes of accessible data. It is not a file system or real-time data storage system, but rather a long-term storage system for a more permanent type of static data that can be retrieved, leveraged, and then updated if necessary. Primary examples of data that best fit this type of storage model are virtual machine images, photo storage, email storage and backup archiving. Having no central "brain" or master point of control provides greater scalability, redundancy and permanence.

Objects are written to multiple hardware devices in the data center, with the OpenStack software responsible for ensuring data replication and integrity across the cluster. Storage clusters can scale horizontally by adding new nodes. Should a node fail, OpenStack works to replicate its content from other active nodes. Because OpenStack uses software logic to ensure data replication and distribution across different devices, inexpensive commodity hard drives and servers can be used in lieu of more expensive equipment.

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.

Barclamp Parameters

Name	Default	Description
Auth Method	Keystone	Choose to use integrated or stand alone authentication
Keystone Instance	[generated]	If using Keystone, link to the Crowbar's Keystone proposal(s)
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.

For swift, parameters should not be changed after applying the proposal. Addition or removal of devices from the proposal will be dynamicaly reconfigured in the swift configuration after the initial proposal has been applied.

Baclamp Roles

Swift offers three roles for configuration. The primary role, swift-storage role, identifies the nodes that store the data.

The infrastructure roles, swift-ring-compute and swift-proxy-acct, provide external access and control functions for the swift cluster. The default node allocation is to use the same as the MySQL barclamp.

4.4 Glance

Please see https://github.com/dellcloudedge/crowbar/wiki/Glance-barclamp for the latest updates.

Background from http://openstack.org/projects/image-service

OpenStack Image Service (code-named Glance) provides discovery, registration, and delivery services for virtual disk images. The Image Service API server provides a standard REST interface for querying information about virtual disk images stored in a variety of back-end stores, including OpenStack Object Storage. Clients can register new virtual disk images with the Image Service, query for information on publicly available disk images, and use the Image Service's client library for streaming virtual disk images.

Barclamp Parameters

Name	Default	Description
Working Directory	/var/lib/glance	
PID Directory	/var/run/glance	
Notifier Strategy	Noop	Only option is "No Operation"
File System Store Data Directory	/bar/lib/glance/images	
Scrubber …	Many options	
Reaper …	Many options	
Pruner …	Many Options	
Prefetch …	Many Options	
API …	Many Options	
Registry …	Many Options	
Cache …	Many Options	
Database	MySQL	Type of database (MySQL or SQLite)
SQL Idle Timeout	3600	
SQLite Connection		Only used if not using MySQL
MySQL Instance	[generated]	Select the Crowbar MySQL proposal to use
Use Keystone	True	
Keystone Instance	[generated]	Select the Crowbar Keystone proposal to use
Use Syslog	False	

Baclamp Roles

Glace provides the glace-server role so that users can select a node as the glance server. This node should have adequate disk space to cache images. The default node allocation is to use the same as the MySQL barclamp. This is not required.

4.5 Nova Dashboard ("Horizon")

Please see https://github.com/dellcloudedge/crowbar/wiki/Nova-dashboard-barclamp for the latest updates.

Background from http://openstack.org/projects/

OpenStack Dashboard enables administrators and users to access and provision cloud-based resources through a self-service portal.

Barclamp Parameters

Name	Default	Description
Keystone instance	[generated]	Select the Crowbar Keystone proposal to use
SQL engine	mysql	Choose database type (MySQL or SQLite)
MySQL instance	[generated]	Select the Crowbar MySQL proposal to use
Show Swift	False	Show Swift integration. Required Swift

Barclamp Roles

Dashboard provides the nova-dashboard-server role so that users can select a node as the dashboard server. The default node allocation is to use the same as the MySQL barclamp. This is not required.

4.6 Nova

Please see https://github.com/dellcloudedge/crowbar/wiki/Nova--barclamp for the latest updates.

*Background from **http://openstack.org/projects/compute/***

OpenStack Compute is open source software designed to provision and manage large networks of virtual machines, creating a redundant and scalable cloud computing platform. It gives you the software, control panels, and APIs required to orchestrate a cloud, including running instances, managing networks, and controlling access through users and projects. OpenStack Compute strives to be both hardware and hypervisor agnostic, currently supporting a variety of standard hardware configurations and seven major hypervisors.

Barclamp Parameters

Name	Default	Description
MySQL Instance	[generated]	Select the Crowbar MySQL proposal to use
Keystone Instance	[generated]	Select the Crowbar Keystone proposal to use
Glance Instance	[generated]	Select the Crowbar Glance proposal to use
Verbose	True	
Libvirt Type	KVM	Virtualization model (switch to QEMU if testing on virtual machines)
Networking Config		
Tenant VLANs	False	
DHCP enabled	True	
HA enabled	True	Default mode is DHCP + HA
Allow Same Net Traffic	False	VM isolation
Num Networks	1	Used for VLAN mode
Network Size	256	Used for VLAN model (num * size = 256)

Barclamp Roles

The Nova Barclamp has two roles. The nova-multi- controller determine which node(s) perform the infrastructure management and API functions. The default node allocation is to use the same as the MySQL barclamp. This is not required.

The nova-multi-compute identifies nodes that act as virtualization hosts. The majority of the nodes in the nova deployment will perform this role.

Nova Networking

This section is called out separately because of its complexity and scope. Is it not a complete reference, please refer to the Crowbar wiki, https://github.com/dellcloudedge/crowbar/wiki/Network--barclamp, for complete networking details.

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 users 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.

4.7 Kong

Background: http://openstack.org/projects/

Kong is a test suite deployed with OpenStack. It is for development environments and not needed for pilots or production deployments.

Barclamp Parameters

Name	Default	Description
Param1	File	

Baclamp Roles

Kong has no roles.