Dell Cloud Solution for OpenStack™ Solutions

OpenStack Barclamps Users Guide

Version 1.3

OpenStack Version Essex

Updated June 2012

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

Figures

Tables

Introduction

This document provides instructions to use when deploying **OpenStack** components using **Crowbar 1.3**. 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, June 2012)
- Crowbar Users Guide (June 2012)
- Bootstrapping Open Source Clouds (Dell Tech White Paper, updated Dec 2011)
- CloudOps White Paper (Dell Tech White Paper, Oct 2011)

Concepts

The purpose of this guide is to explain the special aspects of OpenStack on Crowbar. Please consult the Crowbar Users Guide and Crowbar Deployment Guide for assistance with installing and using Crowbar.

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

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, visit the OpenStack documentation web site at http://docs.openstack.org/.

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

Dell Specific Options

The Dell End User License Agreement (EULA) version of Crowbar provides additional functionality beyond that in the open source version. It also uses a color palette that is different from the open source version.

Crowbar is not limited to managing Dell servers and components. Due to driver requirements, some barclamps (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 an 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 automatically reflected.

Architecture

The Crowbar OpenStack deployment includes both core and incubated OpenStack components. Crowbar deploys each component as a module, known as a barclamp. All shared components are

broken out as independent barclamps. Crowbar automatically detects and integrates 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.

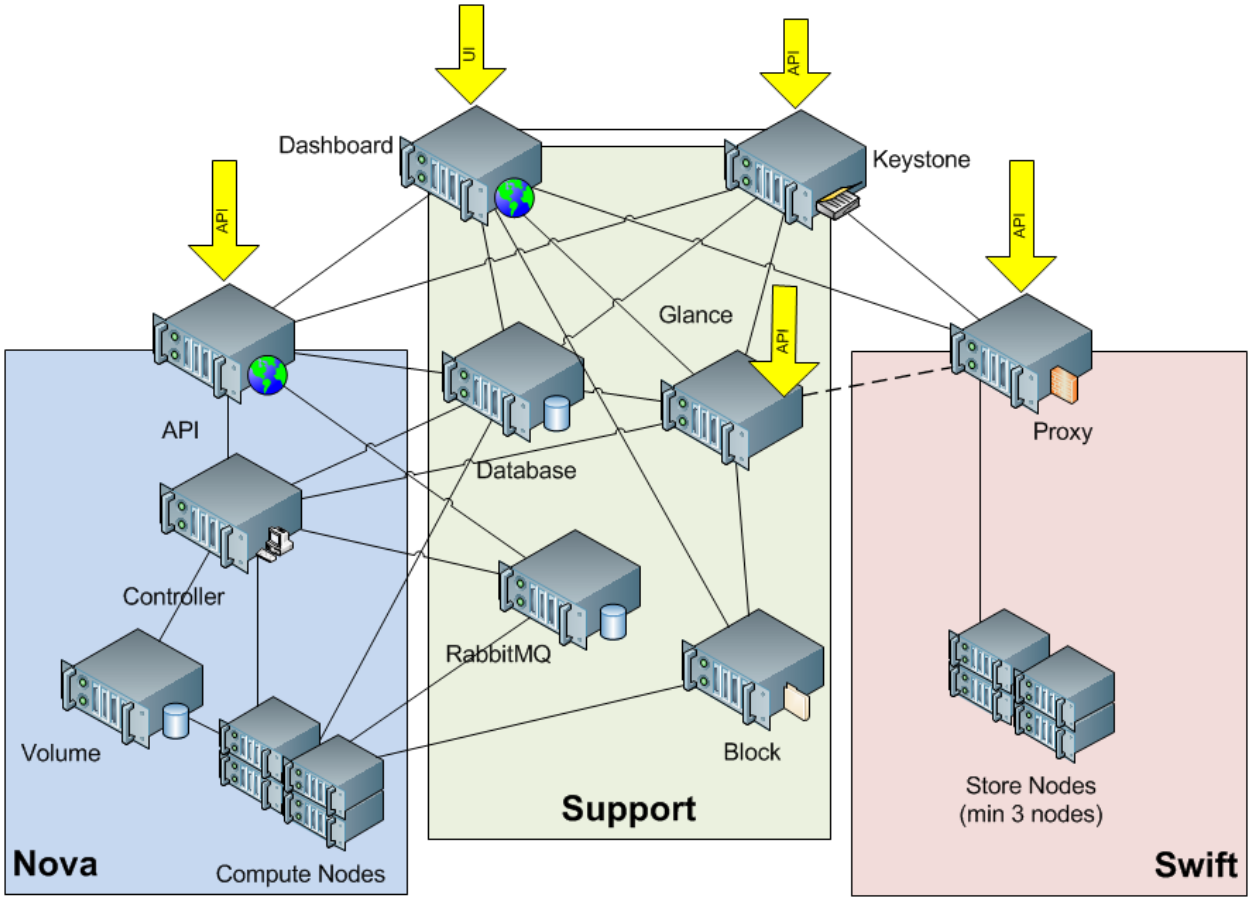

Figure 1 : Crowbar Target for Openstack Deployment

OpenStack Barclamp Suite

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.

Barclamps

The table below shows the barclamps that are available with the Crowbar v1.3 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.

The following OpenStack barclamps are included with Crowbar.

Table 1: Openstack Barclamps

Barclamp	Function	Comments
MySQL	Database	Used by Keystone, Nova, Horizon, and Glance.
Keystone	Centralized Authentication & Authorization	Not core, but strongly recommended. When installed, the identity service is automatically leveraged by all other components.
Swift	Object Store	Provides distributed object storage
Glance	Image Cache	Glance service (Nova image management) for the cloud. Used by Nova.
Nova Dashboard ("Horizon")	User Interface	Not core, but strongly recommended. Provides a web user interface and configuration capabilities for other OpenStack components.
Nova	Compute	Supports many network modes.

MySQL

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

Background

MySQL (http://mysql.com) is a widely adopted open source database that stores relational data for several OpenStack components.

This barclamp can support multiple proposals.

Table 2: MySQL Barclamp Parameters

Name	Default	Description
Datadir	/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 nodes are configured as servers and which are configured as clients.

Barclamps that require a MySQL client will cause a client to be automatically deployed on the appropriate nodes, and will update the MySQL proposal to reflect the nodes that have clients installed. As a result, when creating or editing a MySQL proposal, it is only necessary to select nodes to be configured as MySQL servers by assigning them to the mysql-server role.

Keystone

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

Background

The Keystone Identity Service (http://keystone.openstack.org) provides unified authentication across all OpenStack projects and integrates with existing authentication systems.

Table 3: Keystone Barclamp Parameters

Name	Default	Description
SQL Engine	MySQL	Choose MySQL or SQLite as a backing store
MySQL Instance	[generated]	Select a MySQL proposal to use as a database (requires choosing MySQL as the SQL Engine)
Default Tenant	openstack	Default tenant
Regular User Username	crowbar	Default user name
Regular User Password	crowbar	Default user password
Administrator Username	admin	Administrator user name
Administrator Password	crowbar	Administrator password

Barclamp Roles

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

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

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. All data is stored in structures called partitions, which are replicated a minimum of three times, ensuring data permanence. Should a node fail, OpenStack works to serve its content from other active nodes and create new replicas of the objects. 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.

The Swift barclamp includes the following components:

- Proxy node - provides the API to the cluster, including authentication.
- Storage nodes - provide storage for cluster.
- Ring node – generates the ring file which is distributed to all nodes to provide the logical lookup information to determine where objects are stored in the cluster.

Table 4: Swift Barclamp Parameters

Name	Default	Description
Keystone instance	[generated]	The Keystone proposal to use
Keystone Service User	swift	The user that Swift uses when authenticating with Keystone
Keystone Service Password	[generated]	The password for the Swift Keystone authentication user
Zones	2	The number of zones in this cluster (should be >= # of replicas)
Partitions	18	The number of bits to represent the partition count
Minimum Partitions per Hour	1	The minimum amount of time a partition should stay put, in hours
Replicas	1	The number of replicas that should be made for each object
Cluster Hash	[generated]	Shared among all nodes in a swift cluster. Can be generated using od -t x8 -N 8 -A n </dev/random
Cluster Admin Password	swauth	Super user password - used for managing users
User	swift	The uid to be used for swift processes
Group	swift	The gid to be used for swift processes
Debug	true	Indicates the service should run in debug

mode

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

Barclamp Roles

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

The infrastructure roles are Swift-ring-compute and Swift-proxy-acct. Swift-ring-compute configures a node to provide ring file generation services, and Swift-proxy-acct provides the external access and control functions for a Swift cluster. The default node allocation is to use the same node as the MySQL barclamp for these roles.

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.

Table 5: Glance Barclamp Parameters

Name	Default	Description
Working Directory	/var/lib/glance	Glance working directory
PID Directory	/var/run/glance	Location of Glance's PID files
Notifier Strategy	Noop	The only option is "No Operation"
Image Store Directory	/var/lib/glance/images	Location of images
Scrubber:		
Log File	/var/log/glance/scrubber.log	The location where the scrubber will log
Config File	/etc/glance/glance-scrubber.conf	The configuration file for the scrubber
Debug	false	Indicates if the scrubber will run in debug mode
Verbose	true	Indicates if the scrubber will run in verbose mode
Reaper:		
Log File	/var/log/glance/reaper.log	The location where the reaper will log
Config File	/etc/glance/glance-reaper.conf	The configuration file for the reaper
Debug	false	Indicates if the reaper will run in debug mode
Verbose	true	Indicates if the reaper will run in verbose mode
Pruner:		
Log File	/var/log/glance/pruner.log	The location where the pruner will log
Config File	/etc/glance/glance-pruner.conf	The configuration file for the pruner
Debug	false	Indicates if the pruner will run in debug mode
Verbose	true	Indicates if the pruner will

		run in verbose mode
Prefetcher:		
Log File	/var/log/glance/prefetcher.log	The location where the prefetcher will log
Config File	/etc/glance/glance- prefetcher.conf	The configuration file for the prefetcher
Debug	false	Indicates if the prefetcher will run in debug mode
Verbose	true	Indicates if the prefetcher will run in verbose mode
API:		
Log File	/var/log/glance/api.log	The location where the API will log
Config File	/etc/glance/glance-api.conf	The configuration file for the API
Paste INI File	/etc/glance/glance-api-paste.ini	Paste Deploy configuration file for the API
Debug	false	Indicates if the API will run in debug mode
Verbose	true	Indicates if the API will run in verbose mode
Bind to All Addresses	true	Controls if the API will bind to all addresses or the public address only
Access Port	9292	The port the API service will run on
Registry:		
Log File	/var/log/glance/registry.log	The location where the registry will log
Config File	/etc/glance/glance-registry.conf	The configuration file for the registry
Paste INI File	/etc/glance/glance-registry-paste.ini	Paste Deploy configuration file for the registry
Debug	false	Indicates if the registry will run in debug mode
Verbose	true	Indicates if the registry will run in verbose mode
Bind to All Addresses	true	Controls if the registry will bind to all addresses or the public address only
Access Port	9191	The port the registry service will run on
Caching:		
Enable Caching	false	Indicates if caching should be on
Turn On Cache Management	false	Enables the use of glance-cache-manage CLI & the corresponding API

Directory	/var/lib/glance/image-cache	The location where images are cached
Grace Period	3600	The timeout for accessing the image
Stall Timeout	86400	The timeout to wait for a stalled GET request
Database:		
Database Type	MySQL	Type of database (MySQL or SQLite)
SQL Idle Timeout	3600	MySQL idle time check
SQLite Connection String		String for SQLite connection. Only used if not using MySQL
MySQL Instance	[generated]	The Crowbar MySQL proposal to use
Use Keystone	True	Indicates to Crowbar if Keystone is to be used for authentication
Keystone Instance	[generated]	The Crowbar Keystone proposal to use
Service User	glance	The user that Glance uses when authenticating with Keystone
Service Password	[generated]	The password for the Swift Keystone authentication user
Use Syslog	False	Indicates to Glance to not log to syslog

Barclamp Roles

Glance provides the glance-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 node as the MySQL barclamp. This is not required.

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.

Table 6: Nova Dashboard Barclamp Parameters

Name	Default	Description
SQL Engine	MySQL	Choose database type (MySQL or SQLite)
MySQL Instance	[generated]	Select the Crowbar MySQL proposal to use
Keystone Instance	[generated]	Select the Crowbar Keystone proposal to use

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 node as the MySQL barclamp. This is not required.

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.

Table 7: Nova Barclamp Parameters

Name	Default	Description
MySQL	[generated]	The MySQL proposal to use
Keystone	[generated]	The Keystone proposal to use
Keystone Service User	nova	The user that Nova uses when authenticating with Keystone
Keystone Service Password	[generated]	The password for the Nova Keystone authentication user
Glance	[generated]	The Glance proposal to use
Verbose	true	Indicates if Nova will run in verbose mode
Use NoVNC (otherwise VPN-VNC)	true	Indicates what VNC package to use
Hypervisor	kvm	Indicates what hypervisor Nova should use when spinning up virtual machines (select qemu if running Nova on virtual machines). The default is kvm, but will be switched to qemu if virtual machines are detected.
Network Options:		
Use Tenant Vlans	false	Indicates if Nova should use VLANs for each tenant
DHCP Enabled	true	Indicates if Nova should hand out IP addresses using DHCP
High Availability Enabled	true	Indicates if Nova should use HA networking mode
Allow Same-Network Traffic	false	Network security option that isolates VMs from same network traffic
Number of Networks	1	The number of subnets to split the nova-fixed network into from the network barclamp. Used for VLAN mode
Network Size	256	The number of IP addresses in a single network. Used for VLAN mode
Volume Options:		
Name of Volume	nova-volumes	The name of the volume-group created on the nova-volume node

Type of Volume	Raw	This field indicates the type of volume to create. If raw is specified, the system attempts to use the remaining unused disks to create a volume group. If the system doesn't have additional free drives, the system will switch to local. Local uses a local file in the existing file system based upon other parameters.
Volume File Name	/var/lib/nova/volume.raw	When local type is chosen or fallen back to, this field is the name of the file in the file system to use
Maximum File Size	2000	This parameter is specified in gigabytes. When local type is chosen or fallen back to, this field defines the maximum size of that file. If the file is too big for the file system, the size of the file will be capped to 90% of the free space in that file system (at the time of creation)
Disk selection method	all	When raw type is chosen, this field indicates how to select the disks to use for volume construction. "all" means use all available. "first" means use the first one detected. "selected" means use the disks selected in the list below this option.

Barclamp Roles

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

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

The Nova-multi-volume role identifies a single node on which a Nova volume will be created. The default node allocation for the volume role is to use the same node as the controller role.

Nova Volume

For Essex, the OpenStack dashboard requires a nova-volume service to function and display properly. Crowbar's Nova barclamp has been updated to have a new Nova-multi-volume role. A node with the Nova-multi-volume role will be deployed with a RAID10 configuration to enable redundancy for the volume store. The volume is created when the Nova proposal is applied. Changing Nova-multi-volume parameters after initial application may not work correctly. At the present time, Nova volumes are not cleaned up or removed.

After both the Nova and Nova Dashboard proposals have been applied, the OpenStack Dashboard can be used to create, attach, detach, and destroy volumes. The "Instances & Volumes" tab of the navigation column allows for manipulation of volumes. Volumes can be snapshotted and should be visible in the "Images & Snapshots" tab. Attached volumes can be validated by logging in to the VM and running "fdisk –l".

Note that at the present time, volumes cannot be attached for VMs in systems using the qemu hypervisor.

Nova Networking

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

Nova has three networking modes available. They are integrated with the networking barclamp modes. 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 uses the information about the network topology from the networking barclamp. Nova assumes that three networks are available: admin, public, and nova_fixed_network. The nova_floating_network is defined, but not used or required. The usage of this network is evolving in the community. The admin network is used for service communication. The public network is used for the 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. The public network may be partially presented. In all cases, the nova-network node will act as the router between public and nova_fixed_network. Note that in a standard Crowbar Nova deployment, the nova-network node is the same as the Nova-multi-controller node.

Flat Network

In this mode, the nova-compute node gets an address from nova-network and injects that address into the VM's image (linux-only). In our setup, this image then pulls from the node with the nova-api role to get its custom configuration files (keys, etc).

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.

For the flat network, the network parameters should be configured to separate the nova_fixed_network into a single network with all addresses available. This is specified in the "Number of Networks" and "Network Size" parameters. The DHCP start/end parameters of the nova_fixed network act as 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 the nova_fixed network, eth0.300 for the public network, 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 and public networks.

Flat DHCP Network

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

The network parameters should be configured to separate the nova_fixed_network into a single network with all addresses available. This is specified in the "Number of Networks" and "Network Size" parameters. The DHCP start parameter of the nova_fixed_network acts as 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 the nova_fixed network, eth0.300 for the public network, and eth0 for the admin network. 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 and public networks.

VLAN DHCP Network

In this mode, the nova-compute node doesn't modify the VM image and uses dnsmasq to hand out addresses. There are two important differences between this mode and Flat DHCP mode.

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 "Number of Networks" and "Network Size" parameters. If the nova_fixed network is a class B, "Number of Networks" is 1024, and "Network Size" is 64, then this will support 1024 projects. The external assumptions are that the networking barclamp has setup the single, dual or teamed network, and that the reserved VLANs are already trunked 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 the nova_fixed network, eth0.300 for the public network, and eth0 for the admin network. 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 and public networks. Custom VLANs that are allocated to each project will start at 501 and continue upwards.

The second important 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 a cloud-pipe. The cloud-pipe image needs to be in Glance and setup in a way that openvpn configuration can be injected into it.