Dell | Apache Hadoop Solution

Dell | Cloudera Solution Deployment Guide v1.6

A Dell Deployment Guide

Dell Next Generation Compute Solutions

The power to do more

Table of Contents

Tables

Figures

August 2011 Revision A00

Overview

Summary

This document presents the Deployment Guide of the Dell™ | Cloudera™ Hadoop™ Solution. The deployment guide describes the steps to install Dell | Cloudera Solution on predefined hardware and network configuration specified in the "Dell | Cloudera Solution Reference Architecture v1.6" document. It covers the steps required to prepare hardware platforms for the deployment of Cloudera Manager or Cloudera Hadoop (CDH). Use the *Dell | Cloudera Apache Hadoop Solution Crowbar Administration User Guide* for deployment of Cloudera Manager.

Abbreviations

Abbreviation	Definition
BMC	Baseboard management controller
CDH	Cloudera Distribution for Hadoop
DMBS	Database management system
EDW	Enterprise data warehouse
EoR	End-of-row switch/router
HDFS	Hadoop File System
IPMI	Intelligent Platform Management Interface
NIC	Network interface card
LOM	Local area Network on Motherboard
OS	Operating system
ToR	Top-of-rack switch/router

Dell | Cloudera Solution

Solution Overview

Hadoop is an Apache project being built and used by a global community of contributors, using the Java programming language. Yahoo! has been the largest contributor to the project, and uses Hadoop extensively across its businesses. Other contributors and users include Facebook, LinkedIn, eHarmony, and eBay. Cloudera has created a quality-controlled distribution of Hadoop and offers commercial management software, support, and consulting services.

Dell developed a solution for Hadoop that includes optimized hardware, software, and services to streamline deployment and improve the customer experience.

The Dell | Cloudera Solution is based on the Cloudera CDH Enterprise distribution of Hadoop. Dell's solution includes:

- Reference architecture and best practices
- Optimized hardware and network infrastructure
- Cloudera CDH Enterprise software (CDH community-provided for customer deployed solutions)
- Hadoop infrastructure management tools
- Dell Crowbar software

This solution provides Dell a foundation to offer additional solutions as the Hadoop environment evolves and expands.

The solution is designed to address the following use cases:

Table 1: Hadoop Use Cases

Use case	Description
Data storage	The user would like to be able to collect and store unstructured and semi-structured data in a fault-resilient scalable data store that can be organized and sorted for indexing and analysis.
Batch processing of unstructured data	The user would like to batch-process (index, analyze, etc.) large quantities of unstructured and semi-structured data.
Data archive	The user would like medium-term (12–36 months) archival of data from EDW/DBMS to increase the length that data is retrained or to meet data retention policies/compliance.
Integration with data warehouse	The user would like to transfer data stored in Hadoop into a separate DBMS for advanced analytics. Also the user may want to transfer the data from DMBS back on Hadoop.

Aside from the Hadoop core technology (HDFS, MapReduce, etc.) Dell had designed additional capabilities meant to address specific customer needs:

- Monitoring, reporting, and alerting of the hardware and software components
- Infrastructure configuration automation

The Dell | Cloudera Solution lowers the barrier to adoption for organizations looking to use Hadoop in production. Dell's customer-centered approach is to create rapidly deployable and highly optimized end-to-end Hadoop solutions running on commodity hardware. Dell provides all the hardware and software components and resources to meet your requirements, and no other supplier need be involved.

The hardware platform for the Dell | Cloudera Solution is the Dell™ PowerEdge™ R-series. Dell PowerEdge R-series servers are focused on hyper-scale and cloud capabilities. Rather than emphasizing gigahertz and gigabytes, these servers deliver maximum density, memory, and serviceability while minimizing total cost of ownership. It's all about getting the processing customers need in the least amount of space and in an energy-efficient package that slashes operational costs.

For this release (v1.6), Dell recommends Red Hat Enterprise Linux 6.2 for use in Cloudera Hadoop deployments. The recommended Java Virtual Machine (JVM) is the Oracle Sun JVM 1.6u27 or above.

The hardware platforms, the operating system, and the Java Virtual Machine make up the foundation on which the Hadoop software stack runs.

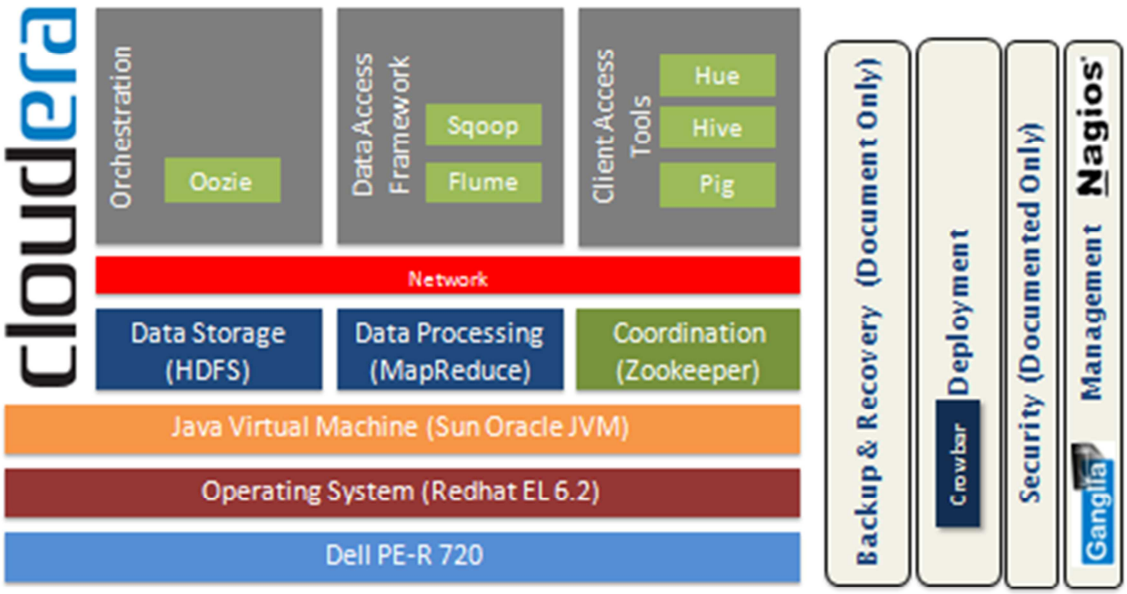


Figure 1: Dell | Cloudera Hadoop Solution Taxonomy

The bottom layer of the Hadoop stack (Figure 1) comprises two frameworks:

1. The *Data Storage Framework (HDFS)* is the file system that Hadoop uses to store data on the cluster nodes. Hadoop Distributed File System (HDFS) is a distributed, scalable, and portable file system.
2. The *Data Processing Framework (MapReduce)* is a massively-parallel compute framework inspired by Google's MapReduce papers.

The next layer of the stack in the Dell | Cloudera Solution design is the network layer. Dell recommends implementing the Hadoop cluster on a dedicated network for two reasons:

1. Dell provides network design blueprints that have been tested and qualified.
2. Network performance predictability—sharing the network with other applications may have detrimental impact on the performance of the Hadoop jobs.

The next three frameworks—the *Data, Access Framework*, the *Client Access Framework* and the *Orchestration*—comprise utilities that are part of the Hadoop ecosystem and provided by CDH.

Dell listened to its customers and designed a Hadoop solution that is unique in the marketplace. Dell's end-to-end solution approach means that you can be in production with Hadoop in a shorter time than is traditionally possible with homegrown solutions. The Dell | Cloudera Solution embodies *all* the software functions and services needed to run Hadoop in a production environment. One of Dell's chief contributions to Hadoop is a

method to rapidly deploy and integrate Hadoop in production. These complementary functions are designed and implemented side-by-side with the core Hadoop core technology.

Installing and configuring Hadoop is non-trivial. There are different roles and configurations that need to be deployed on various nodes. Designing, deploying, and optimizing the network layer to match Hadoop's scalability requires consideration for the type of workloads that will be running on the Hadoop cluster. The deployment mechanism that Dell designed for Hadoop automates the deployment of the cluster from "bare-metal" (no operating system installed) all the way to installing and configuring the Hadoop software components to your specific requirements. Intermediary steps include system BIOS update and configuration, RAID/SAS configuration, operating system deployment, Hadoop software deployment, Hadoop software configuration, and integration with your data center applications (i.e. monitoring and alerting).

Data backup and recovery is another topic that was brought up during customer roundtables. As Hadoop becomes the *de facto* platform for business-critical applications, the data that is stored in Hadoop is crucial for ensuring business continuity. Dell's approach is to offer several enterprise-grade backup solutions and let the customer choose while providing reference architectures and deployments guides for streamlined, consistent, low-risk implementations. Contact your Dell sales representative for additional information.

Lastly, Dell's open, integrated approach to enterprise-wide systems management enables you to build comprehensive system management solutions based on open standards and integrated with industry-leading partners. Instead of building a patchwork of solutions leading to systems management sprawl, Dell integrates the management of the Dell hardware running theHadoop cluster with the "traditional" Hadoop management consoles (Ganglia, Nagios).

To summarize, Dell has added Hadoop to its data analytics solutions portfolio. Dell's end-to-end solution approach means that Dell will provide readily available software interfaces for integration between the solutions in the portfolio.

In the current design, the Dell | Cloudera Solution contains the core components of a typical Hadoop deployment (HDFS, MapReduce, etc.) and auxiliary services (monitoring, reporting, security, etc.) that span the entire solution stack.

Dell | Cloudera Solution Hardware Architecture

The Dell | Cloudera Solution hardware consists of:

Master Node (aka Name Node)—runs all the services needed to manage the HDFS data storage and MapReduce task distribution and tracking.

Slave Node— runs all the services required to store blocks of data on the local hard drives and execute processing tasks against that data

Edge Node—provides the interface between the data and processing capacity available in the Hadoop cluster and a user of that capacity

Admin Node—provides cluster deployment and management capabilities

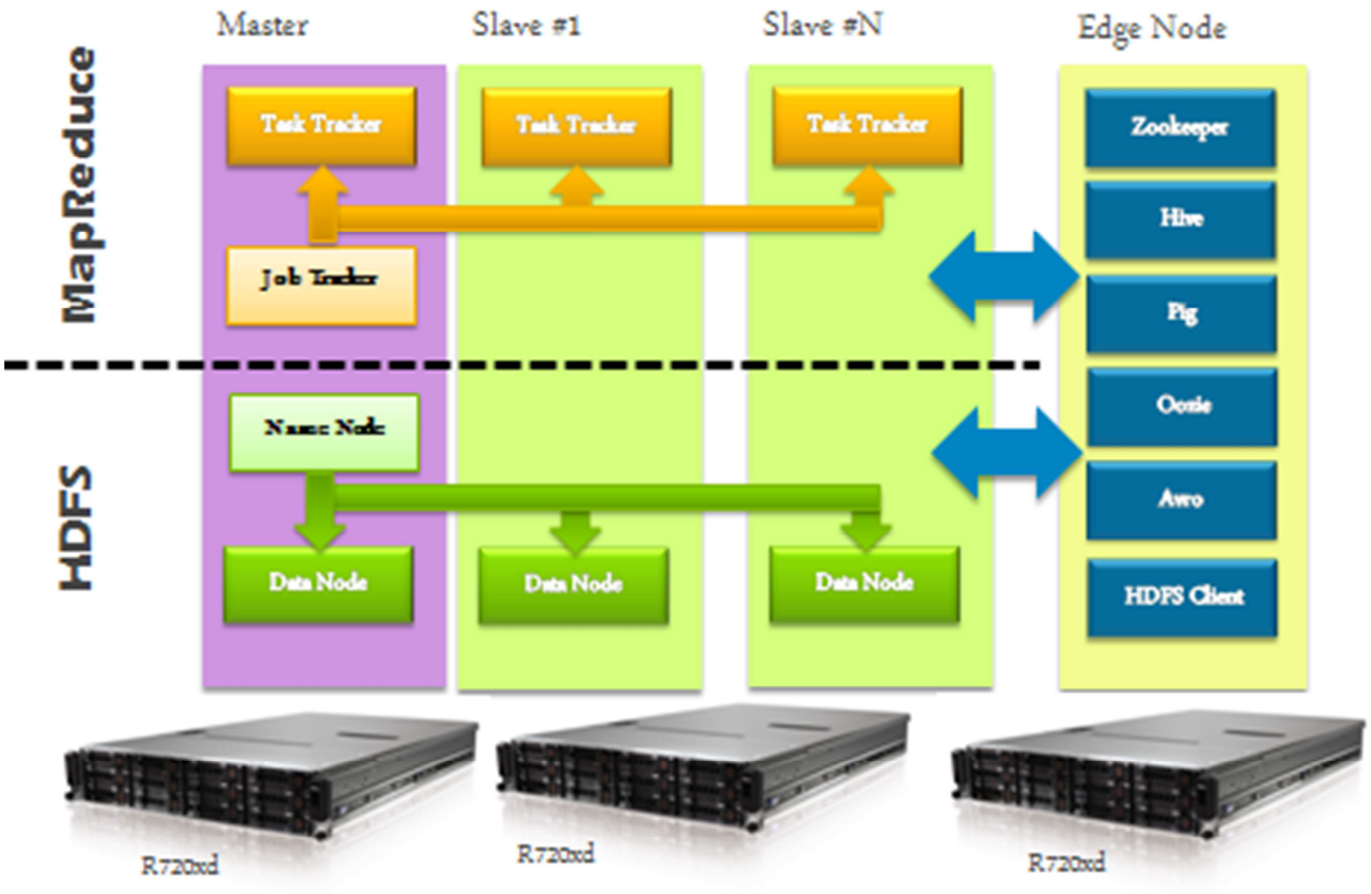


Figure 2: Dell | Cloudera Solution Hardware Architecture

High-level Architecture

The hardware configurations for the Dell | Cloudera Solution are:

Table 2: Dell | Cloudera Hardware Configurations – PowerEdge R720xd

Machine Function	Master Node	Secondary Master Node (serves as Admin Node)	Edge Node	Slave Node
Platform	PowerEdge R720xd			
CPU	2x E5-2640 (6-core)			
RAM (Minimum)	96GB			48GB
LOM	4x1GbE			
DISK	6x 600GB 10K SAS 2.5"			24 x 1TB SATA 7.2K 2.5"
Storage Controller	PERC H710			
RAID	RAID 10			JBOD
Min per Rack	1	1	1	1
Max Per Rack	1	1	1	20
Min per Pod	1	1	1	3*
Max per Pod	1	1		60
Min per cluster	1	1	1	36
Max per Cluster	1	1	To be determined based on sizing criteria	720

- Be sure to consult your Dell Account Representative before changing the recommended disk sizes.
- A minimum of five Data Nodes are needed if Zookeeper will be used in the environment
- Secondary Master Node serves as the Admin Node for Dell Crowbar functionality.

Table **5**: Dell | Cloudera Hardware Configurations – PowerEdge **R720/**R720xd				
Machine Function	**Master Node**	**Secondary Master Node (serves as Admin Node)**	**Edge Node**	**Slave Node**
Platform	PowerEdge R720			PowerEdge 720xd
CPU	2x E5-2640 (6-core)			
RAM (Minimum)	96GB			48GB
LOM	4x1GbE			
DISK	6x 600GB 10K SAS 3.5"			24 x 1TB SATA 7.2K 2.5"
Storage Controller	PERC H710			
RAID	RAID 10			JBOD
Min per Rack	1	1	1	1
Max Per Rack	1	1	1	20
Min per Pod	1	1	1	3*
Max per Pod	1	1		60
Min per cluster	1	1	1	36

Table 3: Dell | Cloudera Hadoop Solution Software Locations

Daemon	Primary Location	Secondary Location
JobTracker	MasterNode01	MasterNode02
TaskTracker	SlaveNode(x)	
Master Node	MasterNode01	MasterNode02
Operating System Provisioning	MasterNode02	MasterNode01

Chef	MasterNode02	MasterNode01
Yum Repositories	MasterNode02	MasterNode01
Crowbar Admin	MasterNode02	**A separate server from the redundant Name Node functionality**
Cloudera Management Suite	EdgeNode(x)	
Zookeeper	MasterNode(06-08)	MasterNode(09-10)
HMaster	MasterNode(11+)	
RegionServer	SlaveNode(x)	

Table 4: Dell | Cloudera Solution Support Matrix

RA Version	OS Version	Hadoop Version	Available Support
1.6	Red Hat Enterprise Linux 6.2	Cloudera Enterprise	Dell Hardware Support Cloudera Hadoop Support Red Hat Linux Support
1.6	CentOS 6.2	Cloudera Distribution including Apache Hadoop Cloudera Enterprise	Dell Hardware Support

High-level Network Architecture

Network Overview

The Dell | Cloudera solution implements at a minimum three distinct, separate VLANs:

- **Hadoop Cluster Production LAN** —connects the compute node NICs into the fabric used for sharing data and distributing work tasks among compute nodes
- **Hadoop Cluster Management LAN**—connects all the iDRAC/BMCs in the cluster nodes
- **Hadoop Cluster Edge LAN**—connects the cluster to the outside world

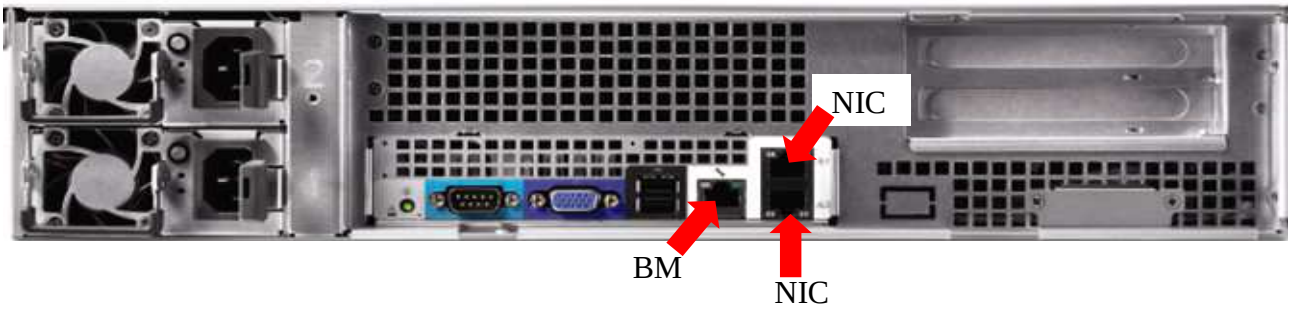
All servers in a Hadoop cluster are tied together using TCP/IP networks. These networks form a data interconnect across which individual servers pass data back and forth, return query results, and load/unload data. These networks are also used for management.

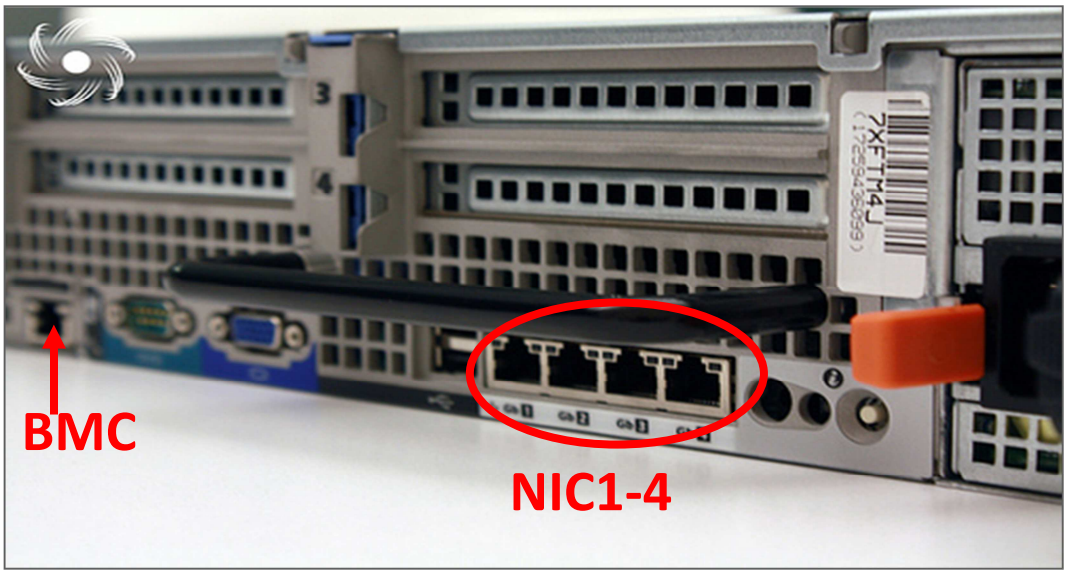
The admin node manages all the cluster nodes. It assigns the other nodes IP addresses; PXE boots them, configures them, and provides them the necessary software for their roles. To provide these services, the admin node runs Crowbar, Chef, DHCP, TFTP, NTP, and other services, and this must be the only DHCP server visible to the compute and storage nodes. Details follow:

- **DHCP server**—assigns and manages IPs for the compute and storage nodes.
- **NTP server** (Network Time Protocol server)—makes sure all nodes are keeping the same clock.

TFTP server—PXE boots compute and storage nodes with a Linux kernel. The TFTP server services any PXE boot request it receives with its default options. **DNS server**—manages the name resolution for the nodes and can be configured to provide external name forwarding. 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. If Crowbar is used to deploy the cluster, it manages all networks, and comes out of the box preconfigured to allow the initial configuration to come up quickly by predefining the 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.


NIC
BM
NIC

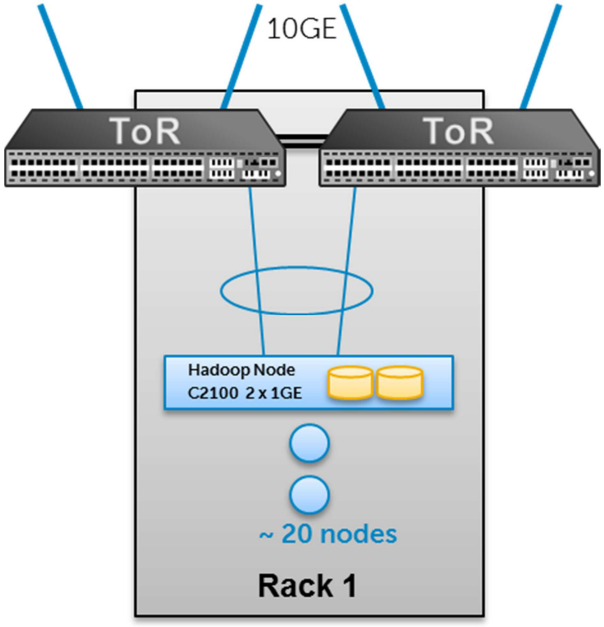

BMC
NIC1-4

Configuring the Force 10 Network Solution

Single rack configuration

Using 1G nodes the Dell Force10 recommends using S60 ToR switches in the rack. Each rack could have a maximum of 20 servers. Each rack has two ToR S60 switches that are stacked and this stack connects to the two S4810 switches. The S60 stack offers a single switch view to the servers. Each StorageNode has 2 data 1xG NIC ports. It forms a LAG of 2 ports with 1 port on each switch in the stack. The LAG of 2G offers a switch redundancy within the rack and offer high availability.

Figure 3: Single rack view



There would be some configurations that are not present on the switches out of the box which are also required prior to these major features. An example of these is enabling the interfaces ('no shut'), configuration of IPs on the management interfaces, enable ssh (telnet is enabled by default) and authorization details. The configuration guide, would help get started on these steps.

Stacking S60s

The following configuration helps stack the two s60s together within the rack. This configuration assumes the stacking module in both s60s is in module 0 (IO facing side) and the 10G uplink module is in slot 1 (power supply and fan side).

Connect the port 49 on module 0 (IO side from the left) to the port 49 of the second S60 and similarly connect port 50 on both switches. The stack is automatically detected and formed without a user configuration. Using the CLI command 'show system brief' verify that the stacking module is detected by the S60.

When you are adding units to a stack, you can either:

• Allow FTOS to automatically assign the new unit a position in the stack, or

• Manually determine each unit's position in the stack by configuring each unit to correspond with the stack before connecting it. Three configurable system variables affect how a new unit joins a stack: priority, stack number, and provision.

After the new unit loads, it synchronizes its running and startup configurations with the stack.

```
TOR-Rack1#stack-unit renumber
TOR-Rack1(conf)# stack-unit priority <higher priority determines primary role>
```

After connecting the switches together run the following command to check the status of the stack

```
TOR-Rack1#show system brief

Stack MAC : 00:01:e8:d5:ef:81

-- Stack Info --

Unit    UnitType   Status   ReqTyp   CurTyp   Version   Ports
--------------------------------------------------------------------------
0       Standby    online    S60     S60     8.3.3.7    52
1       Management online    S60     S60     8.3.3.7    52
```

Up linking the S60s

The following configuration helps create configurations for the uplink the stack. This configuration assumes the 10G uplink module is in slot 1 (power supply and fan side). The uplink ports are going to be numbered 0/51,0/52 and 1/51,1/52 respectively. All four 10G interfaces would part of a single LAG or port-channel. The following configurations show that.

```
# Put the user ports in the switchport mode

TOR-Rack1(config)# interface range gigabitethernet 0/1 - 47

TOR-Rack1(config-if-range-gi-0/1-47)# no shutdown

TOR-Rack1(config-if-range-gi-0/1-47)#switchport

TOR-Rack1(config-if-range-gi-0/1-47)#end



# Repeat the same for ports on the second unit

TOR-Rack1(config)# interface range gigabitethernet 1/1 - 47

<snip>…
```

```
# Create port-channel of the 4 10G ports.The example below shows it for 1 port.

# Repeat the same configs for other 10G ports 0/52,1/51 and 1/52.


TOR-Rack1(conf)#interface Gigabitethernet 0/51

TOR-Rack1(conf-if-gi-3/15)#no shutdown

TOR-Rack1(conf-if-gi-3/15)#port-channel-protocol lacp

TOR-Rack1(conf-if-gi-3/15-lacp)#port-channel 1 mode active


# Change the defaults on the port-channel that gets created automatically

# From the above commands.


TOR-Rack1(conf)#interface port-channel 1

TOR-Rack1(conf-if-po-1)#no shutdown

TOR-Rack1(conf-if-po-1)#switchport


# Add the Data ports 0 through 30 and the port-channel 1 to vlan 100


TOR-Rack1#config

TOR-Rack1 (conf)#int vlan 100

TOR-Rack1 (conf-if-vlan)#tagged po 1

TOR-Rack1 (conf-if-vlan)#untagged gi 0/0-21

TOR-Rack1 (conf-if-vlan)#untagged gi 1/0-21

TOR-Rack1 (conf-if-vlan)#show conf

!

interface Vlan 100

no ip address
```

```
tagged Port-channel 1

untagged gi 0/0-21

untagged gi 1/0-21



TOR-Rack1#config

TOR-Rack1 (conf)#int vlan 300

TOR-Rack1 (conf-if-vlan)#tagged po 1

TOR-Rack1 (conf-if-vlan)#untagged gi 0/29-41

TOR-Rack1 (conf-if-vlan)#show conf
!

interface Vlan 300

no ip address

tagged Port-channel 1

untagged gi 0/29-41
```

So far the configuration is sufficient to link the nodes to the ToR switches, Stacking the ToR and up links from ToR.

The uplink port-channel links are all active and forward traffic to the aggregate switches. Each flow, unique combination of a source and destination, gets hashed internally and gets load-balanced across the port-channel.

Server Gateway

The nodes in a rack have a single virtual IP as their gateway for routing purpose. The VRRP protocol runs on the aggregation S4810s. It does not need any configuration on the ToR. The VRRP master owns the virtual IP and does the routing but the combination of VLT and VRRP ensures that backup also routes or switches the traffic if it has a path in its forwarding table. This is an active-active brained capability where routing is independent of which switch owns the virtual IP.

Management network

The BMC ports from all the nodes connect to the same ToR switches as the data ports. However the management vlan is separate from the data vlan. Ports 0 to 30 on the ToR are reserved for data connections and 31 to 48 are

configured for management network. This is achieved by creating a separate VLAN on the the ToR and adding all the management ports as part of that VLAN.

```
TOR-Rack1(conf)#int vlan 300

TOR-Rack1(conf-if-vlan)#tagged po 1

TOR-Rack1(conf-if-vlan)#untagged gi 0/31-47

TOR-Rack1(conf-if-vlan)#untagged gi 1/31-47
```

Multi-Rack configuration

Once the single rack is deployed from the server and network perspective we can take a look at the multi-rack view and then move on to configure the aggregation switches that connect the racks together. This section shows the S4810 aggregating the clusters together to enable inter-rack traffic as well as the management network. As we saw there are two separate VLANs for data and management, all port-channels on S4810 and ToR are tagged in these two VLANs.

The following table shows the network inventory details in a full cluster of 3 racks.

60 node network

Total Racks	3 (15-20 nodes per rack)
Top of Rack switch	6 S60 (2 per rack)
Pod-interconnect switch	2 S4810
Server	2RU R720/R720xd
Over-subscription at ToR	1:1
Modules in each ToR	1x 12-2port Stacking, 1x 10G -2 port uplink

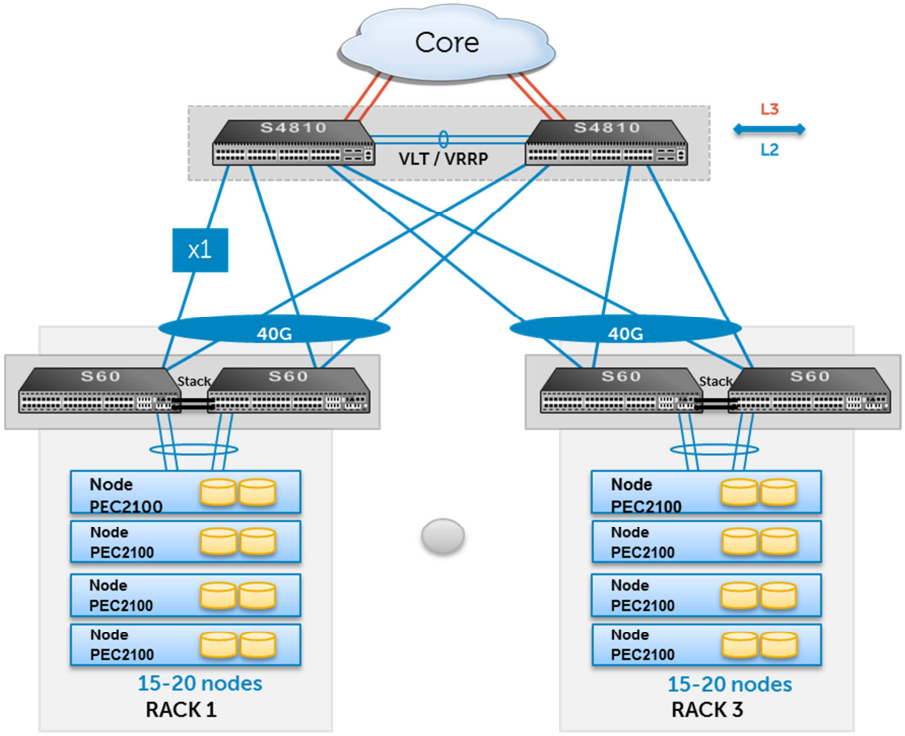


Figure 4: Multi-Rack view

VRRP on S4810

The following configuration shows a sample VRRP configuration on the s4810s. This configuration is created on the VLAN interfaces of the S4810. Since there is only a single VLAN 100 in the cluster of three racks, a single instance of this configuration is needed.

```
Force10_VLTpeer1(conf)#int vlan 100

Force10_VLTpeer1(conf-if-gi-1/1)#vrrp-group 100

Force10_VLTpeer1(conf-if-gi-1/1-vrid-111)#virtual-address 10.10.10.1

#One or more these virtual IP addresses can be configured, which can be used

#as the unique gateway per rack or cluster.

Force10_VLTpeer1(conf-if-gi-1/1-vrid-111)# priority 125

# Priority from 1-255 can be used to determine which switch owns the VIP and becomes the VRRP master.


# Repeat the same configuration on the second VLT peer, except for a different priority.
```

VLT on S4810

The second part of configuration is the pod-interconnect switches that run VLT with each other.

Figure 1. **S4810 VLT interconnect**

Following these steps we will configure VLT on the pair of s4810s that interconnect the racks. To configure virtual link trunking, you must create a VLT domain, configure a backup link and interconnect trunk, and connect the peer switches in a VLT domain to an attached access device (switch or server). But first RSTP should be configured as a best practice on the s4810 as well as the S60s.

```
Force10_VLTpeer1(conf)#protocol spanning-tree rstp

Force10_VLTpeer1(conf-rstp)#no disable

Force10_VLTpeer1(conf-rstp)#bridge-priority 4096



#Repeat the same on VLTPeer2 with a different bridge priority to make it the root.



Force10_VLTpeer2(conf-rstp)#bridge-priority 0
```

The next figures show a sample configuration on VLT. The VLT works over a primary link and a backup link. Therefore the VLT configuration consists of configuring the IP connectivity details of each other. In addition each port-channel to the layer-2 switch, S60 stack in this case, gets a configuration specifying the port-channel that acts as the ICL link. In absence of a direct path to the destination the ICL link would carry the traffic to the peer. The backup link is only for heartbeat status of the peer, no data traffic flows over it.

Figure 2. **VLT configuration on peer1**

```
Force10_VLTpeer1(conf)#vlt domain 999
Force10_VLTpeer1(conf-vlt-domain)#peer-link port-channel 100
Force10_VLTpeer1(conf-vlt-domain)#back-up destination 10.11.206.35
Force10_VLTpeer1(conf-vlt-domain)#exit


Force10_VLTpeer1(conf)#interface ManagementEthernet 0/0
Force10_VLTpeer1(conf-if-ma-0/0)#ip address 10.11.206.23/16
Force10_VLTpeer1(conf-if-ma-0/0)#no shutdown
Force10_VLTpeer1(conf-if-ma-0/0)#exit


Force10_VLTpeer1(conf)#interface port-channel 100
Force10_VLTpeer1(conf-if-po-100)#no ip address
Force10_VLTpeer1(conf-if-po-100)#channel-member fortyGigE 0/56,60
Force10_VLTpeer1(conf-if-po-100)#no shutdown
Force10_VLTpeer1(conf-if-po-100)#exit


Force10_VLTpeer1(conf)#interface port-channel 110
Force10_VLTpeer1(conf-if-po-110)#no ip address
Force10_VLTpeer1(conf-if-po-110)#switchport
Force10_VLTpeer1(conf-if-po-110)#channel-member fortyGigE 0/52
Force10_VLTpeer1(conf-if-po-110)#no shutdown
Force10_VLTpeer1(conf-if-po-110)#vlt-peer-lag port-channel 110
Force10_VLTpeer1(conf-if-po-110)#end


Force10_VLTpeer1# show vlan id 10
Codes: * - Default VLAN, G - GVRP VLANs, P - Primary, C - Community, I - Isolated
Q: U - Untagged, T - Tagged
   x - Dot1x untagged, X - Dot1x tagged
   G - GVRP tagged, M - Vlan-stack, H - Hyperpull tagged

    NUM    Status    Description                  Q Ports
    10     Active                                 U Po110(Fo 0/52)
                                                  T Po100(Fo 0/56,60)
```


H

Figure 5: Dell | Cloudera Solution Network Interconnects

The network cabling within the Dell | Cloudera Solution is described in the following table.

Table 5: Dell | Cloudera Solution Network Cabling

: Dell | Cloudera Solution Network Cabling

Component	NICs to Switch Port				
	LOM1	LOM2	LOM3	LOM4	BMC
Admin Node					
Master Node(s)				N/A	
Data Node(s)			N/A	N/A	
Edge Node(s)					

Legend	
	Cluster Production LAN
	Cluster Management LAN
	Cluster Edge LAN

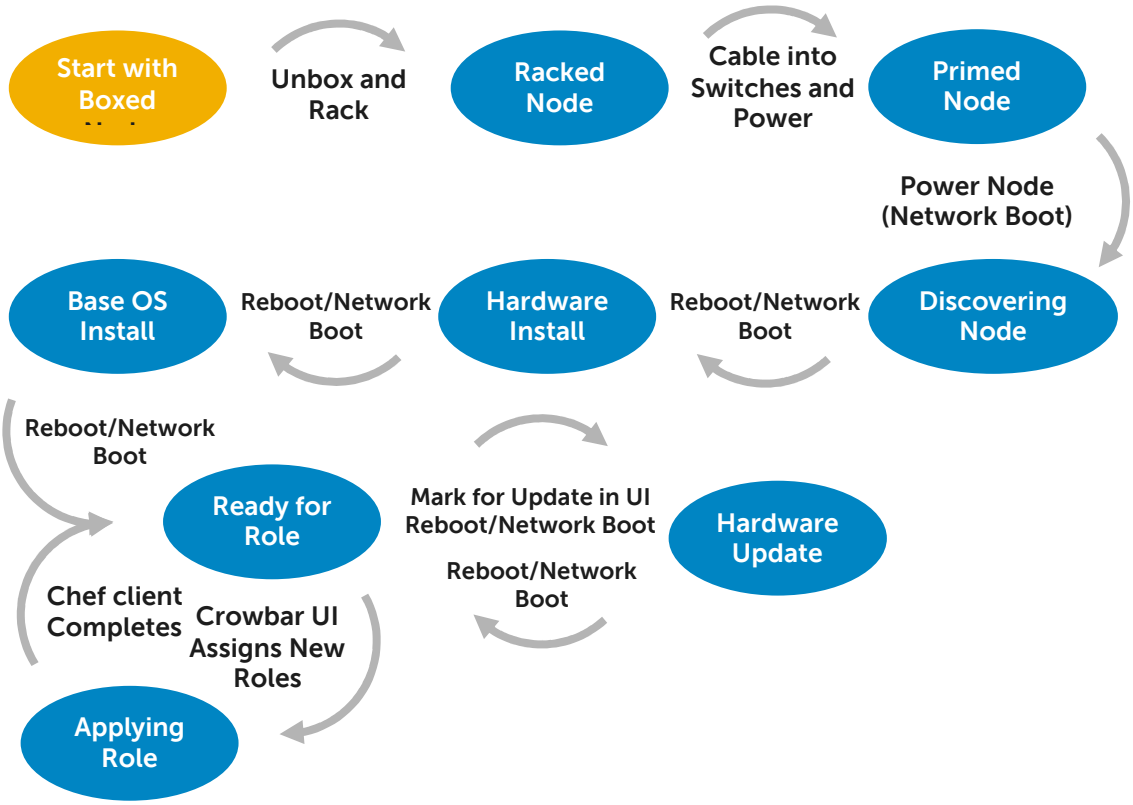
Rack Configuration

Table 6 Rack Configuration

RU	RACK1	RACK2	RACK3
42	R1- Switch2: Force10 S60	R2- Switch2: Force10 S60	R3- Switch2: Force10 S60
41	R1- Switch1 Force10 S60	R2- Switch1: Force10 S60	R3- Switch1: Force10 S60
40	Cable Management	Cable Management	Cable Management
39	Cable Management	Cable Management	Cable Management
38	Master01:R720xd	Edge01: R720xd	R3 - Switch 1 Force 10 S4810
37			R3 - Switch 2 Force 10 S4810
36	Cable Management	Cable Management	Cable Management
35	Cable Management	Cable Management	Cable Management
34	Empty	Empty	Master02_Admin: R720xd
33			
32			Cable Management
31			Cable Management
30			Empty
29			
28			
27			
26			
25			
24			
23			
22			
21			
20	R1- Chassis10: R720xd	R2- Chassis10: R720xd	R3- Chassis10: R720xd
19			
18	R1- Chassis09: R720xd	R2- Chassis09: R720xd	R3- Chassis09: R720xd
17			
16	R1- Chassis08: R720xd	R2- Chassis08: R720xd	R3- Chassis08: R720xd
15			
14	R1- Chassis07 R720xd	R2- Chassis07: R720xd	R3- Chassis07: R720xd
13			
12	R1- Chassis06: R720xd	R2- Chassis06: R720xd	R3- Chassis06: R720xd
11			

10	R1- Chassis05: R720xd	R2- Chassis05: R720xd	R3- Chassis05: R720xd
9			
8	R1- Chassis04: R720xd	R2- Chassis04 R720xd	R3- Chassis04: R720xd
7			
6	R1- Chassis03: R720xd	R2- Chassis03: R720xd	R3- Chassis03: R720xd
5			
4	R1- Chassis02: R720xd	R2- Chassis02: R720xd	R3- Chassis02: R720xd
3			
2	R1- Chassis02: R720xd	R1- Chassis02: R720xd	R1- Chassis02: R720xd

Dell | Cloudera Solution Deployment Process Overview



Dell | Cloudera Solution Hardware Configuration

Edge Node Hardware Configuration

Component	Setting	Parameter
BIOS	Boot Order	1) LOM 1 PXE 2) Internal Boot Device PERC H710 LUN 0
	PXE Boot LOM 1	Enable
	PXE Boot LOM 2	Disable
	C-State	Disable
PERC H710 BIOS	RAID	Enabled
	LUN 0	Disk 0-5 RAID 10
	Boot Order	1) LUN 0

Master Node, Secondary Master Node Hardware Configuration

Component	Setting	Parameter
BIOS	Boot Order	1) LOM 1 PXE 2) Internal Boot Device PERC H710 LUN 0
	PXE Boot LOM 1	Enable
	PXE Boot LOM 2	Disable
	C-State	Disable
PERC H710 BIOS	RAID	Enabled
	LUN 0	Disk 0-5 RAID 10
	Boot Order	1) LUN 0

Slave Node Hardware Configuration

Component	Setting	Parameter
BIOS	Boot Order	1) LOM 1 PXE 2) Internal Boot Device
	PXE Boot LOM 1	Enable
	PXE Boot LOM 2	Disable
	C-State	Disable
PERC H710 Controller BIOS	RAID	Enabled
	LUN0	Disk0 RAID0
	LUN1	Disk1 RAID0
	.	.
	LUN23	Disk23 RAID0
	Boot Order	1) Disk 0 2) Disk 1

Network Switch Configuration

Setting	Parameter	Ports
Spanning-Tree	Disable	ALL
Port-Fast	Disable	ALL
Flow-Control	Enable	ALL

Dell | Cloudera Solution Network Configuration

Table 7: IP Scheme

A	B	C	D	Use
First POD				
172	16	0/22		Rack Number
			1-42	Slave Node[XX] bond0, by Rack Unit
		4/22		Rack Number (1xx)
			200-242	Slave Node [XX] BMC, by Rack Unit
172	16	3	1-19	Master Node[XX]
		3	20-30	Slave Node[XX]
		3	41-50	Edge Node[XX]
172	16	7	1-19	Master Node[XX]
		7	20-30	Master Node[XX]
		7	41-50	Edge Node[XX]
Second POD				
172	16	8/22		Rack Number
			1-42	Slave Node[XX] bond0, by Rack Unit
		12/22		Rack Number (1xx)
			200-242	Slave Node [XX] BMC, by Rack Unit
172	16	11	20-30	Slave Node[XX]
		11	41-50	Edge Node[XX]
172	16	15	41-50	Edge Node[XX]

- All Master Nodes will be addressed in the first Pod only. Additional Pods will not contain additional Master Nodes
- Master Nodes running Zookeeper related services will be distributed among Pods for larger deployments. Please consult with your Dell sales team when designing your solution

Dell | Cloudera Solution Automated Software Installation

Admin Node Installation

To use Crowbar, you must first install an Admin Node. Installing the Admin Node requires installing the base operating system, optionally customizing the Crowbar configuration, and installing Crowbar itself.

The following is required to bootstrap the Admin Node by PXE booting:

1. The user is expected to make the physical arrangements to connect this VM to the network such that the (soon to be) Admin Node can PXE boot from it. A network crossover cable might be required.
2. All BIOS and RAID configuration for the Crowbar Admin node will need to be completed manually, prior to the installation from the Crowbar ISO image.
3. A VM image provides an initial TFTP/DHCP/Boot server. A VMware Player (free download from VMware) is required to execute it.

In preparation for running VMware player on a particular machine, please make sure that:

- **support for Intel VT is enabled in BIOS**
- **There is only one NIC enabled (turn off the wireless NIC if there is one and leave only the wired NIC enabled)**

Procedure:

1. Make sure you have VMware Player installed.
2. Open the VMware machine configuration distributed with Crowbar. (e.g Crowbar_Installer-1.3.tgz)
3. Edit the machine settings and ensure that (see images below):
    - The CD/DVD drive is mounting the Crowbar ISO distribution
    - The Network adapter is configured to use Bridged Networking
4. Obtain the ISO of Crowbar (From your Dell Account Representative), and configure VMware Player to mount it as a DVD in the VM.
5. Plug the crossover cable into eth0 of the server and your network port on the laptop.
6. Start the WMware Player and configure it to use the network port.
7. Power on the admin node, and ensure that:
    - It is set up to boot from the hard disk for subsequent boots
    - The first boot is a network boot

The machine would obtain its image from the VMware Player VM and start the installation process.

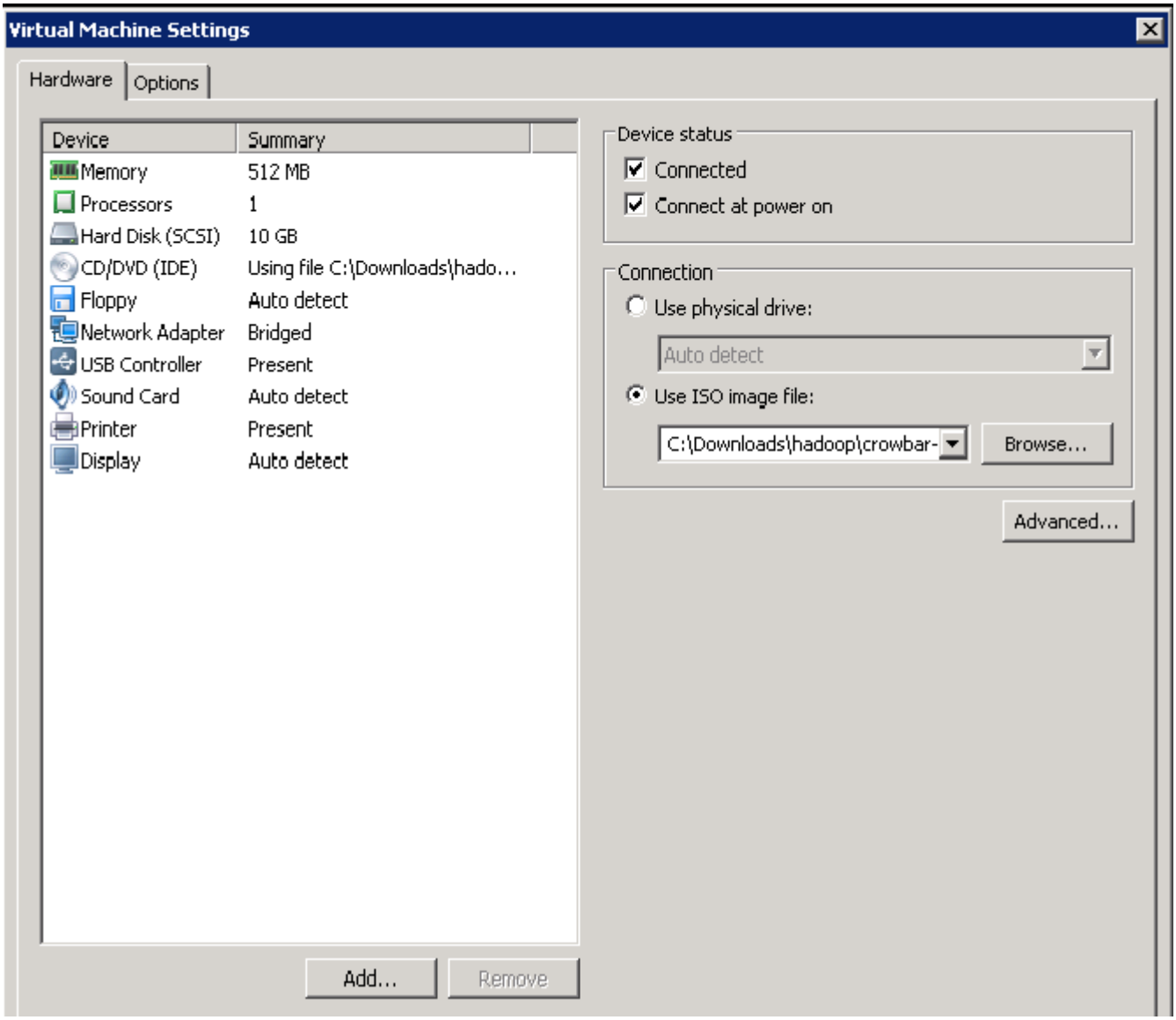


Figure 6: VMware Player Configuration for DVD

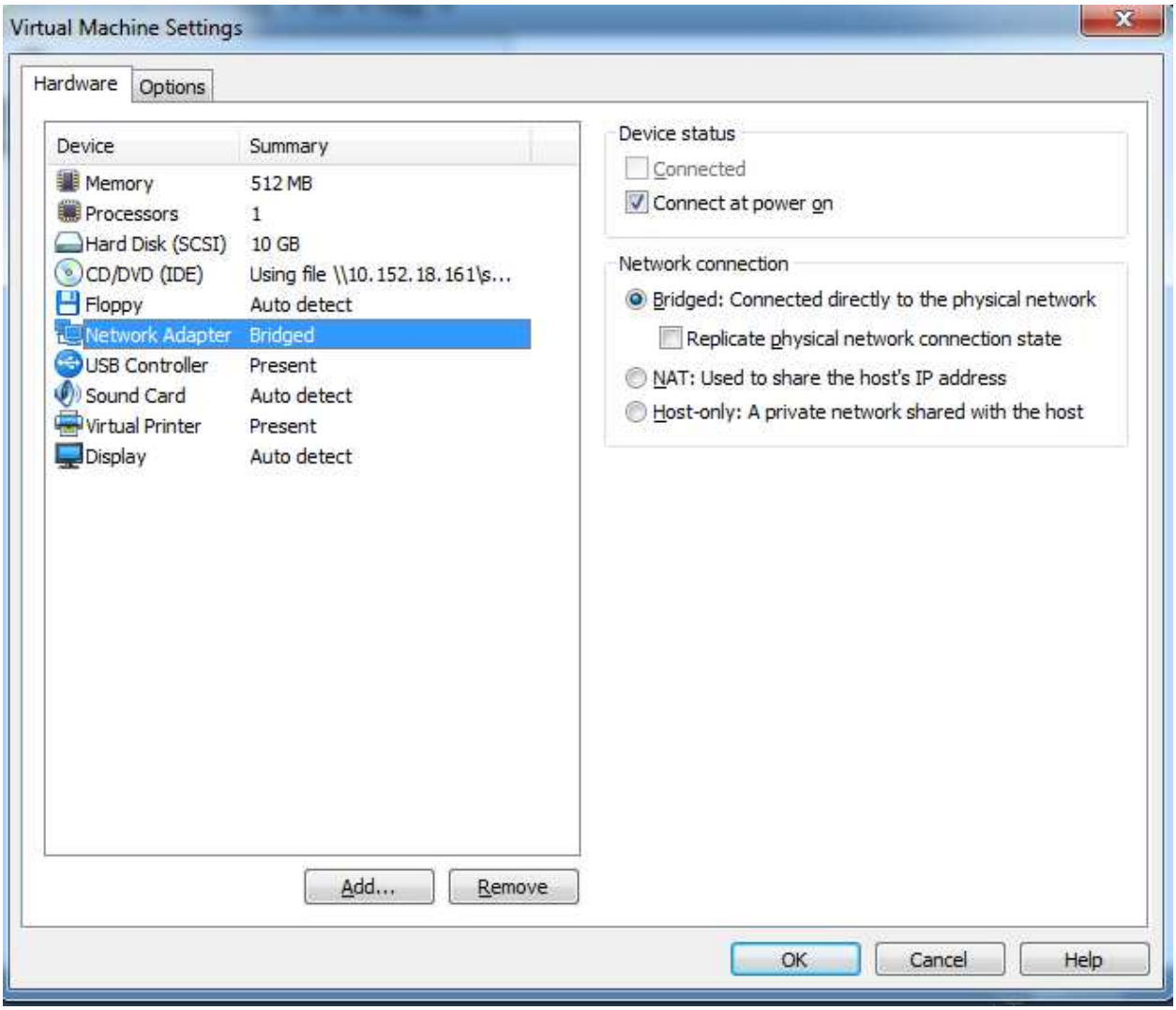


Figure 7: VMware Player Configuration for Network Adapter

1. Installing Crowbar

The image installed in the previous steps includes all the required Crowbar components. Before actually installing Crowbar, there is the opportunity to customize the installation to fit into the deployment environment. The steps below assume default configuration.

To install Crowbar:

- Log onto the Admin node. The default username is `root`, password: `crowbar`.
- If necessary edit the file /opt/dell/chef/data_bags/crowbar/ bc-template-network.json to customize the network information for the deployment. A detailed description of how to edit the network json can be found in the next section.
- **The networks cannot be reconfigured once the system is installed**.
- `cd /tftpboot/redhat_dvd/extra`
- `./install admin.your.cluster.fqdn`

where `admin.your.cluster.fqdn` is the hostname of the admin machine, for example admin.dell.com

This will install Crowbar.

Note: Because there are many dependencies some transient errors might be visible on the console. This is expected

Editing the Network JSON

The json is located at /opt/dell/chef/data_bags/crowbar/bc-template-network.json. The file should be edited before the install command is run to create your admin node.

The information you will need:

1. Vlan ID for each of the VLAN's used by crowbar.
2. Network subnet for each vlan
3. Netmask for each vlan
4. Gateway for the public network and possibly the BMC ranges
5. . This is a section from the json

Example network (admin)

```
"admin": {
"vlan": 100,
"use_vlan": false,
"add_bridge": false,
"subnet": "192.168.124.0",
"netmask": "255.255.255.0",
"broadcast": "192.168.124.255",
"ranges": {
"admin"
{ "start": "192.168.124.10", "end": "192.168.124.11" },
"dhcp"
{ "start": "192.168.124.21", "end": "192.168.124.80" },
"host"
{ "start": "192.168.124.81", "end": "192.168.124.160" },
"switch"
{ "start": "192.168.124.241", "end": "192.168.124.250" }
}
}
```

The biggest issue that many users hit is putting a comma at the end of the section. Other networks are specified in the same manner. The default file contains a public network, an admin network, a BMC Vlan and a BMC network.

Network configuration options are:

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

The teaming sub-parameters are:

Name	Default	Description
mode	5	The teaming algorithm to use for the bonding driver in Linux used on all Nova Compute Nodes.
mode	6	The default teaming algorithm to use for the bonding driver in Linux used on all nodes except Nova Compute Nodes.

The system provides the following 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 on.
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 will be used to generate a vlan tagged interface on the admin nodes that can access the bmc lan.
public	Public network for crowbar and other components	A router, if wanted, is external to the system.

Each network has the following 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. This is mostly for Nova compute.
subnet	IP	The subnet for this network

	Address	
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.

The range map has a string key that is the name and map defining the range.

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

JSON Configuration by Section

1. Attributes
    1. start up delay set to 30 seconds to allow Spanning tree to settle down
    2. Mode - This sets weather to build up single nics or bonded nics options are single, teamed
    3. teaming - sets the mode of the teaming in this case 6
2. Interface Maps - Setup the interface map for figuring out and defining eth0, eth1, eth2 on particular hardware models.
    1. Pattern - Pattern of the hardware model/type
    2. Bus order - order to start enumerating, enumeration begins at eth0, eth1, eth2, eth3, … if a bus is not defined, then it will be enumerated at the end in order of how presented.
3. Conduit Maps - Determines what network, gets mapped to which interface based on what role
    1. Pattern this matched to the attribute variable, the nic type and role
        1. mode (single, team)
        2. nic typ 1g or 10g
        3. role - mastername, crowbar-config-default
4. Conduit or Network Lists to use
    1. conduit name (prod, mgmt, admin)
        1. if_list - what interfaces to use
        2. team_mode - how to team when needed
    2. repeat for other conduits
5. Networks - Define the network, IP ranges, available scopes, etc…
    1. Name of Network (admin, mgmt, prod…)
    2. Conduit - Name of conduit used in step 4
    3. vLan - vlan to use

4. add_bridge - whether to use bridging protocol or vlan tagging
5. subnet - the IP subnet
6. netmask - subnet netmask
7. broadcast - broadcast IP
8. ranges - the ip ranges in the subnet broken down by usage
    1. admin, host, dhcp, are all possible examples.

Special Note: The following Networks are required: bmc, bmc_lan and admin. Admin must have ranges set to the dhcp, admin and host.

JSON Example

"admin": {
"conduit": "prod",
"vlan": 100,
"use_vlan": false,
"add_bridge": false,
"subnet": "172.16.2.0",
"netmask": "255.255.254.0",
"broadcast": "172.16.3.255",
"ranges": {
"host": { "start": "172.16.2.21", "end": "172.16.2.254" },
"dhcp": { "start": "172.16.3.1", "end": "172.16.3.240" },
"admin" : {"start": "172.16.2.18", "end": "172.16.2.20" }

Below is trimmed down version of the json

{
"id": "bc-template-network",
"description": "Instantiates network interfaces on the crowbar managed systems. Also manages the address pool",
"attributes": {
"network": {
"start_up_delay": 30,
"mode": "team",
"teaming": {
"mode": 6
},
"interface_map": [
{
"pattern": "PowerEdge R610",
"bus_order": [
"/0/100/1",
"/0/100/3"
]
{
"pattern": "product",

```
"bus_order": [
"/0/100/1",
"/0/100/2"
]
}
],
"conduit_map": [
{
"pattern": "team/.*/crowbar-config-default",
"conduit_list": {
"prod": {
"if_list": [ "1g1", "1g2", "1g3" ],
"team_mode": 6
},
"mgmt": {
"if_list": [ "1g4" ]
}
}
},

{
"pattern": ".*/.*/.*",
"conduit_list": {
"prod": {
"if_list": [ "1g1" ]
},
"admin": {
"if_list": [ "1g1" ]
},
"external": {
"if_list": [ "1g1" ]
}
}
}
],
"networks": {
"bmc": {
"conduit": "bmc",
"vlan": 300,
"use_vlan": true,
"add_bridge": true,
"subnet": "172.16.0.0",
"netmask": "255.255.255.0",
"broadcast": "172.16.0.255",
"router": "172.16.0.1",
"ranges": {
"router": { "start": "172.16.0.1", "end": "172.16.0.10" },
```

```
"host": { "start": "172.16.0.50", "end": "172.16.2.254" }
}
},
"storage": {
"conduit": "intf1",
"vlan": 200,
"use_vlan": true,
"add_bridge": false,
"subnet": "192.168.125.0",
"netmask": "255.255.255.0",
"broadcast": "192.168.125.255",
"ranges": {
"host": { "start": "192.168.125.10", "end": "192.168.125.239" }
}
},
"bmc_vlan": {
"conduit": "mgmt",
"vlan": 300,
"use_vlan": true,
"add_bridge": true,
"subnet": "172.16.0.0",
"netmask": "255.255.255.0",
"broadcast": "172.16.0.255",
"router": "172.16.0.1",
"ranges": {
"host": { "start": "172.16.0.21", "end": "172.16.2.50" }
}
},
"admin": {
"conduit": "prod",
"vlan": 100,
"use_vlan": false,
"add_bridge": false,
"subnet": "172.16.2.0",
"netmask": "255.255.254.0",
"broadcast": "172.16.3.255",
"ranges": {
"host": { "start": "172.16.2.21", "end": "172.16.2.254" },
"dhcp": { "start": "172.16.3.1", "end": "172.16.3.240" },
"admin" : {"start": "172.16.2.18", "end": "172.16.2.20" }
}
}
}
}
},
"deployment": {
"network": {
```

"crowbar-revision": 0,
"elements": {},
"element_order": [
["network"]
],
"config": {
"environment": "network-base-config",
"mode": "full",
"transitions": true,
"transition_list": ["discovered"]
}
}
}
}

How to add a public IP to a node

From a command prompt on the Admin node, you can execute the following:

- crowbar network allocate_ip default <machine name> public host

To validate address, you can run:

- crowbar machines show <machine name>

You should then have your system setup with a public IP. From the admin section above, you could do "admin switch" instead of "public host", and the IP allocated will be from the switch range of the admin network.

To edit the DNS or NTP time server, please modify the DNS and NTP Barclamps.

How to add an external interface for access to the admin node

You will need to do two things prior to installing the admin node.

First, you will need to add a new network stanza that defines your external network. We will assume for this example that you have one address that you want to assign to the admin node and you are going to run this as a native (non-tagged) interface.

You make up a **vlan** number since we aren't going to use it, make sure that **use_vlan** and **add_bridge** are false, and the rest of the parameters are correct for your network. The admin range will be used to assign the address to the admin node from this pool. Place the assigned address in the start and end fields. The final field is the **conduit** field. We make up an unused value to use in the conduit map in step 2. This example uses "bastion1".

Something like this for example.

```
 "bastion": {
   "conduit": "bastion1",
   "vlan": 50,
   "use_vlan": false,
   "add_bridge": false,
   "subnet": "192.168.235.0",
   "netmask": "255.255.255.0",
   "broadcast": "192.168.235.255",
   "ranges": {
          "admin": { "start": "192.168.235.10", "end": "192.168.235.10" }
   }
 }
```

Second, you will need to update the conduit map for you mode. For this example, we will assume that you are in single mode and have a second interface to use.

The normal conduit map for single mode, any role, and any interface (which is what the admin uses by default), looks like this:

```
       {
         "pattern": "single/.*/.*",
         "conduit_list": {
           "intf0": {
             "if_list": [ "1g1" ]
           },
           "intf1": {
             "if_list": [ "1g1" ]
           },
```

```
    "intf2": {
      "if_list": [ "1g1" ]
    }
  }
},
```

You will need to add a new entry in this stanza. It looks like this:

```
    "bastion1": {
      "if_list": [ "1g2" ]
    },
```

Add it before the **intf0** definition. This will ensure commas match and fun JSON stuff.

This tells the node to make a conduit logically called **bastion1** on the second physical interface.

Save the network json file and install the admin node. Once the admin node is installed, exit out & log back in.

You will need to do two commands to make sure the node gets this new IP address.

crowbar network allocate_ip default <admin name> bastion admin

chef-client

Once the chef-client has finished, you should have access to the admin node through the new interface.

Configuring the network for external connectivity

1. Chose the json that matches your environment best,
    1. /opt/dell/barclamps/network/chef/data_bags/crowbar
        1. bc-network-template.json b. /tftpboot/redhat_dvd/extra/config/network-hadoop-noteam-admin.json ii. network-hadoop-team-admin.json
2. Backup the old file
    1. opt/dell/barclamps/network/chef/data_bags/crowbar/bc-network-template.json
3. Copy the file you want to use to
    1. opt/dell/barclamps/network/chef/data_bags/crowbar/bc-network-template.json (this will overwrite the existing one.)
4. Using your favorite editor edit that file
    1. Change the section of the public IP ranges to match your network

"public": {

"conduit"

"public",

"vlan"

500,

"use_vlan"

false,

"add_bridge"

false,

"subnet"

"192.168.1.0",

"netmask"

"255.255.255.0",

"broadcast"

"192.168.1.255",

"router"

"192.168.1.1",

"ranges"

{

"host"

{ "start": "192.168.1.10", "end": "192.168.1.25" } } }, b. Change the netmask,broadcast, router and ranges c. Verify the file and save it.

1. Run the Install command
2. If you need to deploy the external network to the admin node continue or go to step 12
3. Before starting any slave nodes

4. Connect to the admin node at 172.16.2.18(unless you change the IP ranges of the Admin net) via ssh
5. Execute from the root command prompt
    1. "crowbar network allocate_ip default "admin node FQDN" public host b. "chef-client" c. /etc/init.d/chef-server-webui restart
6. From the Crowbar gui Modify the DNS and NTP barclamps to use the external server and apply them
7. From a command line you can do an ntpq -p
    1. [root@admin config]# ntpq -p
    2. remote refid st t when poll reach delay offset jitter
    3. ==
    4. *172.26.1.50 132.163.4.103 2 u 40 64 377 0.287 -0.433 0.169
8. When the "*" shows up the ntp server is now synced with your server and your server is now ready for the slaves to come online

6. Verifying master node state

When the admin node finishes installation, it will remain at a shell prompt. At this point, all Crowbar services have started. Consult the table below to access these services.

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

Logging into the UI requires acceptance of the EULA. It can be found on the Dashboard under EULA, in Appendix B of this document or at this web page:
http://www.dell.com/content/topics/global.aspx/policy/en/policy?c=us&l=en&s=gen&~section=015#dsla

Set CROWBAR Parameter

export CROWBAR_KEY=$(`cat /etc/crowbar.install.key`)

export CROWBAR_KEY=crowbar:crowbar

Slave Node Installation

Nodes other than the Admin Nodes are installed when they are first powered up. A sequence boot phase is executed (rebooting multiple times) which culminates in deploying a minimal OS image installed on the local drive. Part of the basic installation includes "hooking" the nodes into the infrastructure services—NTP, DNS, Nagios, and Ganglia.

Once known to Crowbar, the node can be managed; it can be powered on and off, rebooted, and components can be installed on it.

Functional components are installed on nodes by including them in one or more barclamps' proposals. For example, when a node is mentioned in a proposal for swift as a storage node, the relevant packages, services, and configuration are deployed to that node when the proposal is committed.

The next section describes details for installing the different components.

Installing components

The general workflow to install any component is the same:

A. Obtain a default proposal which includes the parameters for the component and a mapping of nodes to the roles they are assigned.
B. Edit the proposal to match the desired configuration.
C. Upload the proposal to Crowbar.
D. Commit the proposal.

All these activities are achieved by using the Crowbar command line tool or the Web-based UI. The sections that follow use the command line tool: `/opt/dell/bin/crowbar.`

In the sections that follow, this tool is referred to as "Crowbar."

General installation process

Obtain a proposal

Crowbar can inspect the current known nodes and provide a proposal that best utilizes the available systems for the component being installed. To obtain and inspect this proposed configuration:

```
/opt/dell/bin/crowbar <component> proposal create <name>
```

```
/opt/dell/bin/crowbar <area> proposal show <name> > <local_file_name>
```

Where:

- `<area>` – is the area for which the proposal is made; e.g. Clouderamanager, Pig.
- `<name>` – is the name assigned to this proposal. This name should be unique for the component; i.e. if two hadoop clusters are being installed, the proposals for each should have unique names.
- `<local_file_name>` - Is any file name into which the proposal will be written

Update a proposal

The local file created above can be inspected and modified. The most common changes are:

- Change default passwords and other barclamp parameters (e.g. swift replica count).
- Change assignment of machines to roles.

Once edits are completed, Crowbar must be updated. To update Crowbar with a modified proposal, execute:

```
/opt/dell/bin/crowbar <area> proposal –-file=<local_file_name> edit <name>
```

Where the parameters in this command are exactly as mentioned above, Crowbar will validate the proposal for syntax and basic sanity rules as part of this process.

Committing a proposal

Once the proposal content is satisfactory, the barclamp instance can be activated. To achieve that, execute:

```
/opt/dell/bin/crowbar <area> proposal commit <name>
```

This might take a few moments, as Crowbar is deploying the required software to the machines mentioned in the proposal.

Modifying an active configuration

When committing a proposal that was previously committed, Crowbar compares the new configuration to the currently active state and applies the deltas.

To force Crowbar to reapply a proposal, the active state needs to be deleted:

```
/opt/dell/bin/crowbar <area> delete <name>
```

Installing Cloudera Manager

Use the *Dell | Cloudera Apache Hadoop Solution Crowbar Administration User Guide* for instructions on how to deploy Cloudera Manager.

Dell | Cloudera Solution Software Configuration

Dell | Cloudera Solution Configuration Parameters Recommended Values

Table 8: hdfs-site.xml

Property	Description	Value
dfs.block.size	Lower value offers parallelism	134217728 (128Mb)
dfs.name.dir	Comma-separated list of folders (no space) where a Slave Node stores its blocks	/mnt/hdfs/hdfs01/meta1
dfs.datanode.handler.count	Number of handlers dedicated to serve data block requests in Hadoop Slave Nodes	16 (Start 2 x CORE_COUNT in each SlaveNode)
dfs.namenode.handler.count	More Master Node server threads to handle RPCs from large number of Slave Nodes	Start with 10, increase large clusters (Higher count will drive higher CPU, RAM, and network utilization)
dfs.namenode.du.reserved	The amount of space on each storage volume that HDFS should not use, in bytes.	10M
dfs.replication	Data replication factor. Default is 3.	3 (default)
fs.trash.interval	Time interval between HDFS space reclaiming	1440 (minutes)
dfs.permissions		true (default)
dfs.datanode.handler.count		8
dfs.data.dir	Hadoop Data Node Location	/mnt/hdfs/hdfs01/data1/hdfs comma-separated through /mnt/hdfs/hdfs01/dataN/hdfs

Table 9: mapred-site.xml

Property	Description	Value
mapred.child.java.opts	Larger heap-size for child JVM's of maps/reduces.	-Xmx1024M
mapred.job.tracker	Hostname or IP address and port of the JobTracker.	namenode:8021
mapred.job.tracker.handler.count	More JobTracker server threads to handle RPCs from large number of TaskTrackers.	Start with 32, increase large clusters (higher count will drive higher CPU, RAM and Network utilization)
mapred.reduce.tasks	The number of Reduce tasks per job.	Set to a prime close to the number of available hosts
mapred.local.dir	Comma-separated list of folders (no space) where a TaskTracker stores runtime information	/mnt/hdfs/hdfs01/data1/mapred comma-separated through /mnt/hdfs/hdfs01/dataN/mapred
mapred.tasktracker.map.tasks.maximum	Maximum number of map tasks to run on the node	2 + (2/3) * number of cores per node
mapred.tasktracker.reduce.tasks.maximum	Maximum number of reduce tasks to run per node	2 + (1/3) * number of cores per node
mapred.child.ulimit		2097152
mapred.map.tasks.speculative.execution		FALSE
mapred.reduce.tasks.speculative.execution		FALSE
mapred.job.reuse.jvm.num.tasks		1

Table 10: default.xml

Property	Description	Value
SCAN_IPC_CACHE_LIMIT	Number of rows cached in search engine for each scanner next call over the wire. It reduces the network round trip by 300 times caching 300 rows in each trip.	100
LOCAL_JOB_HANDLER_COUNT	Number of parallel queries executed at one go. Query requests above than this limit gets queued up.	30

Table 11: hadoop-env.sh

Property	Description	Value
java.net.preferIPv4Stack		true
JAVA_HOME		
HADOOP_*_OPTS		**-Xmx2048m**

Table 12: /etc/fstab

Property	Description	Value
File system mount options		data=writeback,nodiratime, noatime

Table 13: core-site.xml

Property	Description	Value
io.file.buffer.size	The size of buffer for use in sequence files. The size of this buffer should probably be a multiple of hardware page size (4096 on Intel x86), and it determines how much data is buffered during read and write operations.	65536 (64Kb)
fs.default.name	The name of the default files system. A URI whose scheme and authority determine the file system implementation.	hdfs://namenode:8020
fs.checkpoint.dir	Comma-separated list of directories on the local file system of the Secondary Master Node where its checkpoint images are stored	TBD
io.sort.factor		80
Io.sort.mb		512

Table 14: /etc/security/limits.conf

Property	Description	Value
mapred – nofile		32768
hdfs – nofile		32768
hbase – nofile		32768

Dell | Cloudera Solution Monitoring and Alerting

The following components will be monitored by the Hadoop monitoring console:

Service Type	Resource	Warning	Critical	Nodes to Monitor	Tool
Disk	HDFS_DISK_[00-10]	60	90	SlaveNode[]	Nagios
SWAP	SWAP	60	90	SlaveNode[]	Nagios
		60	90	Master Node[]	Nagios
		60	90	EdgeNode[]	Nagios
Ping_Node_From_Admin		DELAY	NO RESPONSE	SlaveNode[]	Nagios
		DELAY	NO RESPONSE	Master Node[]	Nagios
		DELAY	NO RESPONSE	EdgeNode[]	Nagios
NIC Bonding		DELAY	1 NIC in Bond	SlaveNode[]	Nagios
		DELAY	1 NIC in Bond	Master Node[]	Nagios
		DELAY	1 NIC in Bond	EdgeNode[]	Nagios
DNS_From_Node		DELAY	NO RESPONSE	SlaveNode[]	Nagios
		DELAY	NO RESPONSE	Master Node[]	Nagios
		DELAY	NO RESPONSE	EdgeNode[]	Nagios
DNS_About_Node		DELAY	NO RESPONSE	SlaveNode[]	Nagios
		DELAY	NO RESPONSE	Master Node[]	Nagios
		DELAY	NO RESPONSE	EdgeNode[]	Nagios
JobTracker_Daemon		DELAY	DAEMON NOT RUNNING	Master Node[]	Nagios
TaskTracker_Daemon		DELAY	DAEMON NOT RUNNING	SlaveNode[]	Nagios
SlaveNode_Daemon		DELAY	DAEMON NOT RUNNING	SlaveNode[]	Nagios
Master Node_Daemon		DELAY	DAEMON NOT RUNNING	Master Node[]	Nagios
SecondaryMaster Node		DELAY	DAEMON NOT RUNNING	Master Node[]	Nagios
SSH		DELAY	NO RESPONSE	SlaveNode[]	Nagios
		DELAY	NO RESPONSE	Master Node[]	Nagios
Zombie_Processes		5	10	SlaveNode[]	Nagios
		5	10	Master Node[]	Nagios
		5	10	EdgeNode[]	Nagios
CPU_Load		80	90	SlaveNode[]	Nagios
		80	90	Master Node[]	Nagios
		80	90	EdgeNode[]	Nagios
Zookeeper_Client		DELAY	DAEMON NOT RUNNING	SlaveNode[]	Nagios
Zookeeper_Server		DELAY	DAEMON NOT RUNNING	Master Node[]	Nagios
JobTracker_Submit_Job		DELAY	NO RESPONSE	Master Node[]	Nagios
Chef_Daemon		DELAY	NO RESPONSE	SlaveNode[]	Nagios
		DELAY	NO RESPONSE	Master Node[]	Nagios
		DELAY	NO RESPONSE	EdgeNode[]	Nagios
Disk	MAPRED_DIR	60	90	SlaveNode[]	Nagios
		60	90	Master Node[]	Nagios
		60	90	EdgeNode[]	Nagios
Memory_Capacity_Used	System Memory	80	90	SlaveNode[]	Nagios
		80	90	Master Node[]	Nagios
		80	90	EdgeNode[]	Nagios
Disk	HDFS01_Capacity	60	90	Master Node[]	Nagios

CPU_Utilizion				SlaveNode[]	Ganglia
				Master Node[]	Ganglia
				EdgeNode[]	Ganglia
Memory_Utilization				SlaveNode[]	Ganglia
				Master Node[]	Ganglia
				EdgeNode[]	Ganglia
NIG_LAG_Utilization				SlaveNode[]	Ganglia
				Master Node[]	Ganglia
				EdgeNode[]	Ganglia
CPU Temp		As defined by SDR (Sensor Data Record)	As defined by SDR	SlaveNode[]	Nagios
		As defined by SDR	PENDING	Master Node[]	Nagios
		As defined by SDR	As defined by SDR	EdgeNode[]	Nagios
Power Supplies		As defined by SDR	As defined by SDR	Master Node[]	Nagios
		As defined by SDR	As defined by SDR	Edge Node[]	Nagios
Master Node _NFS_Mount		DELAY	MOUNT MISSING	Master Node[]	Nagios
Hbase		DELAY	SELECT FAILED	EdgeNode[]	Nagios
		DELAY	INSERT FAILED	EdgeNode[]	Nagios
Hive		DELAY	SELECT FAILED	EdgeNode[]	Nagios
		DELAY	INSERT FAILED	EdgeNode[]	Nagios
Ping_From_Admin	IPMI Interface	DELAY	NO RESPONSE	SlaveNode[]	Nagios
		DELAY	NO RESPONSE	Master Node[]	Nagios
		DELAY	NO RESPONSE	EdgeNode[]	Nagios

Hadoop Ecosystem Components

Component	Master Node	Slave Node	Edge Node	Utilize From	Administer From
Pig	X	X	X	Edge Node	Edge Node
Hive		X	X	Edge Node	Edge Node
Sqoop			X	Edge Node	Edge Node
Zookeeper-Server		X (5)		Edge Node	Edge Node

X—Designates server location for the appropriate package binaries to be installed.

Pig

See https://ccp.cloudera.com/display/CDHDOC/Pig+Installation

Hive

See https://ccp.cloudera.com/display/CDHDOC/Hive+Installation

Sqoop

See https://ccp.cloudera.com/display/CDHDOC/Sqoop+Installation

ZooKeeper

Introduction: What is Apache ZooKeeper?

Apache ZooKeeper is a coordination system that has high availability built in. ZooKeeper allows distributed applications to coordinate with each other. For example, a group of nodes (i.e. web servers) can use ZooKeeper to deliver a highly-available service. They can use ZooKeeper to refer all clients to the master node. Also they can use ZooKeeper to assign a new master in case the original master node fails.

Distributed processes using ZooKeeper coordinate with each other via a shared hierarchical name space of data registers (called ZNodes). This name space is much like that of a standard file system. A name is a sequence of path elements separated by a slash (/). Every ZNode in ZooKeeper's name space is identified by a path. And every ZNode has a parent whose path is a prefix of the ZNode with one less element; the exception to this rule is root (/), which has no parent. Also, exactly like standard file systems, a ZNode cannot be deleted if it has any children.

The main differences between ZooKeeper and standard file systems are that every ZNode can have data associated with it (every file can also be a directory and vice-versa) and ZNodes are limited to the amount of data that they can have. ZooKeeper was designed to store coordination data: status information, configuration, location information, etc. This kind of meta-information is usually measured in kilobytes, if not bytes. ZooKeeper has a built-in sanity check of 1M, to prevent it from being used as a large data store, but in general it is used to store much smaller pieces of data.

ZooKeeper Service Topology

The ZooKeeper service is replicated over a set of machines that comprise the service. These machines maintain an in-memory image (hence, high throughput, low latency) of the data tree along with transaction logs and snapshots in a persistent store.

The machines that make up the ZooKeeper service must all know about each other. As long as a majority of the servers are available the ZooKeeper service will be available. Clients also must know the list of servers. The clients create a handle to the ZooKeeper service using this list of servers.

Because ZooKeeper requires majority, it is best to use an odd number of machines. For example, with four machines ZooKeeper can handle only the failure of a single machine; if two machines fail, the remaining two machines do not constitute a majority. However, with five machines ZooKeeper can handle the failure of two machines. Therefore, to sustain the failure of F machines, the ZooKeeper service should be deployed on (2xF + 1) machines.

Guidelines for ZooKeeper Deployment

The reliability of ZooKeeper rests on two basic assumptions:

- Only a minority of servers in a deployment will fail; size the number ZooKeeper machines in accordance to the 2xF+1 rule. If possible, you should try to make machine failures independent. For example, if most machines share the same switch or are installed in the same rack, failure of that switch or a rack power failure could cause a correlated failure and bring the service down.
- Deployed machines operate correctly, which means execute code correctly, have clocks that operate properly and have storage and network components that perform consistently. ZooKeeper has strong durability requirements, which means it uses storage media to log changes before the operation responsible for the change is allowed to complete. If ZooKeeper has to contend with other applications for access to resources like storage media, its performance will suffer. Ideally, ZooKeeper's transaction log should be on a dedicated device—a dedicated partition is not enough.

For additional information, please click on http://zookeeper.apache.org/doc/r3.3.2/zookeeperAdmin.html.

Installing ZooKeeper Server on Dell | Cloudera Solution Cluster

First, determine the machines that will run the ZooKeeper service. You should start with five HDFS Data Nodes installed in three different racks. For example, these machines can be:

Rack ID	DataNode Hostname/IP	ZooKeeper Machine ID
RACK1	R01-N04	1
	R01-N15	2
RACK2	R02-N07	3
	R02-N11	4
RACK5	R05-N10	5

Following steps should be performed on each ZooKeeper machine:

1. Install the ZooKeeper Server package:

   ```
   # yum –y install hadoop-zookeeper
   # yum –y install hadoop-zookeeper-server
   ```

2. Create a ZooKeeper data directory on the system drive for ZooKeeper logs. The installer creates a /var/zookeeper directory. You can use that directory or create a new one:

   ```
   # mkdir /var/my_zookeeper
   ```

3. Create unique IDs, between 1 and 255, for each ZooKeeper machine and store them in the file myid in the ZooKeeper data directory. For example, on ZooKeeper machine 1 run:

   ```
   # echo 1 > /var/my_zookeeper/myid
   ```

4. On ZooKeeper machine 2, run:

   ```
   # echo 2 > /var/my_zookeeper/myid
   ```

5. Edit the file /etc/zookeeper/zoo.cfg and append the ZooKeeper machine IDs of each machine and its IP address or DNS name.

   ```
   # The number of milliseconds of each tick
   tickTime=2000
   # The number of ticks that the initial
   # synchronization phase can take
   initLimit=10
   # The number of ticks that can pass between
   # sending a request and getting an acknowledgement
   syncLimit=5
   # the directory where the snapshot is stored.
   dataDir=/var/zookeeper
   # the port at which the clients will connect
   clientPort=2181
   server.1=172.16.0.1:2888:3888
   server.2=172.16.0.2:2888:3888
   server.3=172.16.0.3:2888:3888
   server.4=172.16.0.4:2888:3888
   server.6=172.16.0.6:2888:3888
   ```

   Note that the entries of the form server.X list the servers that make up the ZooKeeper service. When the ZooKeeper machine starts up, it determines which ZooKeeper machine it is by looking for the file myid in the data directory. That file contains the server number in ASCII, and it should match X in server.X in the left-hand side of this setting.

6. Increase the heapsize of the ZooKeeper-Server instance to 4GB. Edit the JVM settings in file /usr/bin/zookeeper-server:

```
# vi /usr/bin/zookeeper-server
 export JVMFLAGS="-Dzookeeper.log.threshold=INFO -Xmx4G"
```

Note: Don't forget the double-quotes (").

7. Save the file and start the ZooKeeper server:

```
# /etc/init.d/hadoop-zookeeper start
```

8. Verify that the ZooKeeper service started by reading the state of the service. On one the machines, not necessarily a ZooKeeper machine, run the following command:

```
# echo stat | nc ZKNode_IP 2181
```

Where ZKNode_IP is the IP address or the hostname of one of the ZooKeeper machines and 2181 is the client connect port specified in configuration file zoo.cfg.

The output should look something like this:

```
Zookeeper version: 3.3.3-cdh3u0--1, built on 03/26/2011 00:21 GMT
Clients:
 /172.16.3.20:49499[0](queued=0,recved=1,sent=0)

Latency min/avg/max: 0/0/0
Received: 1
Sent: 0
Outstanding: 0
Zxid: 0x100000004
Mode: leader
Node count: 4
```

For additional ZooKeeper commands click http://zookeeper.apache.org/doc/r3.3.2/zookeeperAdmin.html#sc_zkCommands.

Maintenance

http://zookeeper.apache.org/doc/r3.1.2/zookeeperAdmin.html#sc_maintenance.

Troubleshooting and Common Problems

http://archive.cloudera.com/cdh/3/zookeeper/zookeeperAdmin.html#sc_commonProblems

References

Ghemawat, S. Gobioff, H. and Leung, S.-T. The Google File System. Proceedings of the 19th ACM Symposium on Operating Systems Principles. pp 29--43. Bolton Landing, NY, USA. 2003. © 2003, ACM.

Borthakur, Dhruba. The Hadoop Distributed File System: Architecture and Design. © 2007, The Apache Software Foundation.

Hadoop DFS User Guide. © 2007, The Apache Software Foundation.

HDFS: Permissions User and Administrator Guide. © 2007, The Apache Software Foundation.

HDFS API Javadoc © 2008, The Apache Software Foundation.

HDFS source code

Pig – http://developer.yahoo.com/hadoop/tutorial/pigtutorial.html

Pig – http://pig.apache.org/docs/r0.6.0/setup.html

Zookeeper – http://zookeeper.apache.org/doc/r3.2.2/zookeeperOver.html

Zookeeper – https://ccp.cloudera.com/display/CDHDOC/ZooKeeper+Installation

Zookeeper – http://archive.cloudera.com/cdh/3/zookeeper/zookeeperAdmin.html#sc_zkMulitServerSetup

Nagios – http://www.nagios.org/

Ganglia – http://ganglia.sourceforge.net/

Additional information can be obtained at www.dell.com/hadoop or by e-mailing hadoop@dell.com.

To Learn More

For more information on the Dell | Cloudera Solution, visit:
www.dell.com/hadoop