Dell | Cloudera Apache Hadoop Solution Crowbar Administration User Guide

A Dell User Guide for Apache Hadoop Deployment

Revision 1.6

Dell Next Generation Compute Solutions

The power to do more

Table of Contents

Tables and Figures

Information in this document is subject to change without notice.

© 2012 Dell Inc. All rights reserved.

April 2012

Notes, Cautions, and Warnings

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

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

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

Abbreviations

Abbreviation	Definition
BMC	Baseboard management controller.
DBMS	Database management system.
EDW	Enterprise data warehouse.
EoR	End-of-row switch/router.
HDFS	Hadoop Distributed File System.
IPMI	Intelligent Platform Management Interface.
LAG	Link aggregation group.
LOM	Local Area Network on Motherboard.
NIC	Network interface card.
ToR	Top-of-rack switch/router.

Overview

Hadoop is an Apache project being built and used by a global community of contributors, written in 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.

Introduction

This document provides instructions you to use when deploying Cloudera Manager and Apache Hadoop Eco-System components with Crowbar**.** This guide is for use with the Crowbar Users Guide, and is *not* a stand-alone document. It specifically covers Cloudera Manager, Apache Hadoop and the deployment steps from a Crowbar prospective. Please refer to the Crowbar User Guide for assistance with installing common Crowbar components and configuring the target systems.

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

The Dell | Cloudera Apache Hadoop Solution is based on the Cloudera CDH 3 Enterprise distribution of Hadoop. Dell's solution includes:

- Dell Reference architecture (RA) and best practices documentation.
- Optimized hardware and network infrastructure.
- Cloudera CDH software (CDH Community-provided for customer-deployed solutions).
- Cloudera Manager free edition with the ability to upgrade to enterprise level via Cloudera issued license key.
- Cloudera Manager provided Hadoop infrastructure management tools.
- Dell Crowbar software framework.

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

Document Scope

The focus of this guide is the use of Crowbar, *not* Cloudera Manager or Apache Hadoop. While Crowbar includes substantial components to assist in the deployment of Cloudera Manager and Apache Hadoop, its operational aspects are completely independent. For more detailed information, please refer to the links below;

- Cloudera's Distribution including Apache Hadoop (CDH)
- CDH3 Update 4 Release Notes
- CDH3 Update 3 Release Notes
- CDH3 Update 2 Release Notes
- Cloudera Manager Free Edition User Guide
- Cloudera Manager Free Edition 3.7.x Release Notes
- Cloudera Manager Free Edition Installation Guide
- Configuring Ports for Cloudera Manager Free Edition
- Configuring TLS Security for Cloudera Manager Free Edition

- Hue Open Source Applications User Guide
- Hadoop at Apache.org

This guide provides this additional information about Cloudera as notes flagged with the Cloudera logo. For detailed operational support for Hadoop, we suggest visiting the Cloudera documentation web site at http://www.cloudera.com.

Opscode Chef Server

Crowbar makes extensive use of Opscode Chef Server, http://opscode.com. To explain Crowbar actions, you should understand the underlying Chef implementation. This guide provides this additional Chef information as notes flagged with the Opscode logo.

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

Dell Specific Options

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

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

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

Dell | Cloudera Apache Hadoop Solution

Network Setup

The network configuration assumes a flat L2 wiring. All network connections should be accessible at that layer. Where isolation between different logical networks is required, VLANs are used.

Managing Growth

The system architecture is organized into three components, for sizing as the Hadoop environment grows. From smallest to largest, they are:

- Rack
- Pod
- Cluster

Each has specific characteristics and sizing considerations. You can scale the environment by adding additional capacity as needed, without the need to replace any existing components.

Rack

A rack is the smallest component in a Hadoop environment, and consists of all of the power, network cabling, and two Ethernet switches required to support up to 20 data nodes. These nodes should utilize their own power connectivity and data center space – separate from other racks – and be treated as a fault zone.

Pod

A pod is a single set of stacked Ethernet switches. For the Dell | Cloudera Reference Architecture, both the maximum and minimum are six. A pod consists of the administration and operation infrastructure to support three racks.

Cluster

A cluster is a set of greater than one pod, up to a maximum of 12 pods. A cluster is a set of Hadoop nodes that share the same Network Node and management tools for operating the Hadoop environment.

Note: Please see the Dell | Cloudera Solution Reference Architecture Guide for more detailed information.

Default Networks

The default networks are presented in the following table.

Table 1-1: Default Networks

Usage	Description	Default reserved vLAN tag	Tagged
Admin/Internal vLAN	Used for administrative functions such as Crowbar node installation, TFTP booting, DHCP assignments, KVM, system logs, backups, and other monitoring. There is only one vLAN set up for this function and it is spanned across the entire network.	100	Not tagged
BMC vLAN	Used for connecting to the BMC of each node.	100	Not tagged
Storage vLAN	Used by the Swift storage system for replication of data between machines, monitoring of data integrity, and other storage specific functions (802.1q Tagged).	200	Tagged
Edge vLANs	Used for connections to devices external to the Hadoop cluster	300	Tagged

Usage	Description	Default reserved vLAN tag	Tagged
	infrastructure; these include externally visible services such as load balancers and web servers. Use one or many of these networks, dependent on the need to segregate traffic among groups of servers (802.1q Tagged).		

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

Layout

Due to the nature of Crowbar's network layout, addresses are assigned to a whole network based upon interface, Network Type (Production, Management, and External) and teaming type.

Table 1-2: Master/Secondary (Admin) Name Nodes Network Connections

Interface	Network Type	Teaming Type
BMC	Management LAN	Single
LOM1	Production LAN	Teamed
LOM2	Production LAN	Teamed
Eth1	Production LAN	Teamed
Eth2	Management LAN	Single

Table 1-3: Edge Nodes Network Connections

Interface	Network Type	Teaming Type
BMC	Management LAN	Single
LOM1	Production LAN	Teamed 1
LOM2	Production LAN	Teamed 1
Eth1	External LAN	Teamed 2
Eth2	External LAN	Teamed 2

Table 1-4: Slave Nodes Network Connections

Interface	Network Type	Teaming Type
BMC	Management LAN	Single
LOM1	Production LAN	Teamed 1
LOM2	Production LAN	Teamed 1

IP Addressing

The IP address can be assigned in this fashion, using large subnets to support many machines on the production network. The management network is a Class C network with 254 IP addresses. The Production network is what is known as a /23 with 512 IP addresses. In each network, the first 10 IP addresses are reserved for switches, routers, and firewalls.

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

Table 1-5: IP Addressing Schema

LAN	Network	Subnet	Gateway	Reserved
Management LAN	172.16.0.0	255.255.255.0	172.16.0.1	0.1 – 0.10
Production LAN	172.16.2.0	255.255.254.0	172.16.2.1	2.1-2.20
Name Nodes	DHCP Allocated			

LAN	Network	Subnet	Gateway	Reserved
Slave Nodes	DHCP Allocated			
External LAN	TBD by Customer			

Rack Awareness

With the network set up using Top of Rack (ToR) switches, Rack Awareness can be programmed using the Chef information about which switch the LOM1 is plugged into. A simple script has been added to the Hadoop configuration to pull the information out of Chef, and then use it for Rack Awareness.

Table 1-6: Pod 1 IP Example Addressing Layout

Network: 172.16.0.0	Netmask: 255.255.252.0
Multicast: 172.16.0.0	Broadcast 172.16.3.255

Pod	Rack Number	Network	Server Type	IP Range	Subnet Mask	Gateway
1	1	Production	Slave	172.16.0.1-42	255.255.252.0	172.16.0.1
1	2	Production	Slave	172.16.1. 1-42	255.255.252.0	172.16.0.1
1	3	Production	Slave	172.16.2. 1-42	255.255.252.0	172.16.0.1
1		Production	Master Name	172.16.3.1-19	255.255.252.0	172.16.0.1
1		Production	Secondary Name	172.16.3.20-30	255.255.252.0	172.16.0.1
1		Production	Edge	172.16.3.41-50	255.255.252.0	172.16.0.1
1	1	BMC	Slave	172.16.0.200-242	255.255.252.0	172.16.0.1
1	2	BMC	Slave	172.16.1.200-242	255.255.252.0	172.16.0.1
1	3	BMC	Slave	172.16.2.200.242	255.255.252.0	172.16.0.1
1		BMC	Master Name	172.16.3.201-219	255.255.252.0	172.16.0.1
1		BMC	Secondary Name	172.16.3.220-230	255.255.252.0	172.16.0.1
1		BMC	Edge	172.16.3.231-250	255.255.252.0	172.16.0.1

Table 1-7: Pod 2 IP Example Addressing Layout

Network: 172.16.0.0	Netmask: 255.255.252.0
Multicast: 172.16.0.0	Broadcast: 172.16.3.255

Pod	Rack Number	Network	Server Type	IP Range	Subnet Mask	Gateway
2	1	Production	Slave	172.16.4.1-42	255.255.252.0	172.16.4.1
2	2	Production	Slave	172.16.5. 1-42	255.255.252.0	172.16.4.1
2	3	Production	Slave	172.16.6. 1-42	255.255.252.0	172.16.4.1
2		Production	Master Name	172.16.7.1-19	255.255.252.0	172.16.4.1
2		Production	Secondary Name	172.16.7.20-30	255.255.252.0	172.16.4.1
2		Production	Edge	172.16.7.41-50	255.255.252.0	172.16.4.1
2	1	BMC	Slave	172.16.4.200-242	255.255.252.0	172.16.4.1
2	2	BMC	Slave	172.16.5.200-242	255.255.252.0	172.16.4.1
2	3	BMC	Slave	172.16.6.200.242	255.255.252.0	172.16.4.1
2		BMC	Master Name	172.16.7.201-219	255.255.252.0	172.16.4.1
2		BMC	Secondary Name	172.16.7.220-230	255.255.252.0	172.16.4.1
2		BMC	Edge	172.16.7.231-250	255.255.252.0	172.16.4.1
2		External	Edge	TBD by Customer	TBD	TBD

Hadoop Basics

The Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using a simple programmatic driven processing model. Hadoop is designed to scale up from a minimum of three servers to thousands of machines, each offering local computation and storage.

Rather than rely on hardware to deliver high-availability, the Hadoop library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on a cluster of computers, each of which may be prone to failures.

Hadoop is ideal for organizations with a growing need to store and process massive application datasets. It enables applications to work with thousands of nodes and petabytes of data. This Crowbar barclamp provides the ability to deploy and maintain Hadoop cluster Admin, Master, Slave and Edge nodes. It also provides the capability to configure and deliver Hadoop HDFS and MapReduce components.

Hadoop Overview

- **Hadoop Core**: The common libraries and utilities that provide the basic Hadoop runtime environment. A set of components and interfaces which implement a distributed filesystem and provide general I/O access for the Hadoop framework (serialization, Java RPC and persistent data storage).
- **Hadoop Distributed File System (HDFS)**: A distributed file system that provides redundant, high-throughput access to application data.
- **Hadoop MapReduce**: A software framework for distributed processing of large data sets on compute clusters.

HDFS Overview

HDFS is a core component of the Hadoop framework and it is the underlining Hadoop virtual file system.

HDFS has the underlining concepts of three node classes:

- Master name node which is responsible for managing the file system metadata and transactions.
- Secondary name node which is responsible for checkpointing the name node's persistent state.
- Slave data nodes which are responsible for actually storing the file data.

HDFS stores files as a series of blocks, each of which is by default 64MB in size. A block is the unit of storage for data nodes. Data nodes store and retrieve blocks, and have no concept of the actual physical files that these blocks are composed of.

- Master name node - The master name node is responsible for managing the filesystem metadata and data node mappings. The master name node holds the mapping from files to blocks, which it stores in memory as well as in a persistent metadata store on disk (e.g., the image file and edit log). The mapping between blocks and the data nodes they reside on is not stored persistently. Instead, it is stored in the name node's memory, and is built up from the periodic block reports that data nodes send to the name node. This is the primary metadata store for the cluster.
- Secondary name node - The secondary name is a checkpointing mechanism which can take over the primary name node's functional aspects for this particular operation. During system operation, the name node maintains two on-disk data structures to represent the filesystem state (an image file and an edit log). The image file is a checkpoint of the filesystem metadata at a point in time and the edit log is a transactional redo log of every filesystem metadata mutation since the image file was created. Incoming changes to the filesystem metadata (such as creating a new file) are written to the edit log. When the name node starts, it reconstructs the current state by replaying the edit log. To ensure that the log doesn't grow without bounds, at periodic intervals the edit log is rolled, and a new checkpoint is created by applying the old edit log to the image. This process is performed by the secondary name node daemon, often on a separate machine to the primary since creating a checkpoint has similar memory requirements to the name node itself. A side effect of the checkpointing mechanism is that the secondary name node holds an out-of-date copy of the primary's

persistent state, which, in extreme cases, can be used to recover the filesystem's state. Blocks are stored on the underlying filesystem of the data node, as opposed to the data node managing their own storage, as native kernel level filesystems do.

- Slave nodes - Slave nodes are the distributed collection points for data storage. Functioning data nodes send heartbeats to the name node every 3 seconds. This mechanism forms the communication channel between data node and name node. Occasionally, the name node will piggyback a command to a data node on the heartbeat response, for instance, "send a copy of block e to data node b." One of the first things that a data node does upon startup is send a block report to the name node, and this allows the name node to rapidly form a picture of the block distribution across the cluster.

Cluster Deployment Topology

The Crowbar Hadoop barclamp framework has expanded the concept of node deployment beyond HDFS in order to introduce the notion of a cloud edge node. The cloud edge node sits on the cloud boundary and provides the underlying interface between the data/processing capacity within the Hadoop cluster and the data consumer/end user environment. The addition of the cloud edge node serves to off-load external transactional processing requests from the data nodes and provide an additional level of security between the private cloud and the outside world.

- Master and secondary name nodes - Runs all the basic 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 a data and processing capacity available in the Hadoop cluster and a user of that capacity. Most of the Hadoop eco-system sub-components run on the edge node.
- Admin Node – Provides cluster deployment/management capabilities and is used to deploy Hadoop to all the nodes in the cluster (The Crowbar administration node).

Note: The Hadoop secondary name node runs on the Crowbar admin node by default.

There may be cases when the customer may choose to deploy the add-on services on slave nodes or even the name nodes. Also when the cluster grows beyond a certain size the customer may need to run the Name Node (the HDFS manager) daemon and the JobTracker (the MapReduce manager) on different machines. In that case the customer needs to be able to terminate/uninstall the JobTracker daemon on the original name node and bring it up on the new JobTracker machine.

Eco-system sub-components need to be able to scale independently of the cluster configuration and/or capacity. For example, there may be cases when the data transfer capacity between the Hadoop and data warehouse (i.e. Aster Data) may exceed the max capacity of a single edge node. Adding a second edge node may be a viable alternative.

The design of the Hadoop add-on services need to separate the core Hadoop components (HDFS, MapReduce) from the add-on services and allow the customer to manipulate and deploy the services configuration that makes sense in his environment regardless of the size or topology of the actual Hadoop cluster.

Apache Hadoop Component Deployment

For Hadoop (Cloudera Manager) and Eco-System components (Hive, Sqoop and Pig), employ Crowbar tools to construct a starting proposal and then edit any parameters to fit the specific needs of your environment. Once the proposal is ready, apply the proposal to deploy each system components.

Note: The Base Hadoop system (HDFS and Map Reduce), Zookeeper, Hbase, Oozie and Hue are deployed using the Cloudera Manager administration console. Crowbar also provided some supplemental Hadoop Eco-System Barclamps (Hive, Sqoop and Pig) and you must install the base Hadoop system (HDFS and Map Reduce) using Cloudera Manager before deploying any of these add-ons.

Table 2 Supported Apache Hadoop Components

Component	Deployment Method	Description
Hadoop Core (HDFS/Map Reduce)	Cloudera Manager	Common libraries and utilities that provides the basic Hadoop runtime environment (HDFS/map reduce), a set of components and interfaces which implements a distributed filesystem and provides general i/o access for the Hadoop framework (serialization, java rpc and persistent data storage).
HUE	Cloudera Manager	HUE (Hadoop User Experience) is a user interface framework and SDK platform for visual Hadoop applications. It delivers a suite of web base UI applications which can be used to access and modify the Hadoop Distributed File System (HDFS) and Map Reduce job queue. HUE provides UI application portals for HDFS file browsing, Map/Reduce job control, user account administration and web based on-line help.
HBase	Cloudera Manager	HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed Filesystem), providing BigTable -like capabilities for Hadoop. That is, it provides a fault-tolerant way of storing large quantities of sparse data. HBase features compression, in-memory operation, and Bloom filters on a per-column basis as outlined in the original BigTable paper. Tables in HBase can serve as the input and output for MapReduce jobs run in Hadoop, and may be accessed through the Java API but also through REST, Avro or Thrift gateway APIs. HBase is not a direct replacement for a classic SQL Database, although recently its performance has improved, and it is now serving several data-driven websites, including Facebook's Messaging Platform
ZooKeeper	Cloudera Manager	High-performance coordination service for distributed applications. ZooKeeper provides primitives such as distributed locks which can be used for building large scale distributed processing applications.
Oozie	Cloudera Manager	Oozie is an open-source workflow/coordination service to manage data processing jobs for Apache Hadoop™. It is an extensible, scalable and data-aware service to orchestrate dependencies between jobs running on Hadoop (including HDFS, Pig and MapReduce).
Hive	Crowbar Barclamp	Data warehouse infrastructure that provides SQL based data summarization and ad hoc querying.
Pig	Crowbar Barclamp	Platform for analyzing large data sets that consists of a high-level language for expressing data algorithms.
Sqoop	Crowbar Barclamp	SQL based command-line tool to assist with HDFS data import/export (SQL-to-Hadoop).

For more information about Hadoop, please visit http://hadoop.apache.org/.

Crowbar User Interface

Crowbar is delivered as a Web application available on the admin node using HTTP on port 3000. By default, you can access it using http://192.168.124.10:3000. Additionally, the default installation contains an implementation of Hadoop specific components (see table below).

Note: Crowbar has been tested on the following browsers: FireFox 3.5+, FireFox 4.0, Internet Explorer 7, and Safari 5. A minimum screen resolution of 1024x768 or higher is recommended.

Table 3 User Interface Service URLs

User Interface Service	Default Location	Port	Example URL
Crowbar	Crowbar Admin Node	3000	http:// <crowbar_admin_node >:3000
Cloudera Manager	Crowbar Admin Node	7180	http:// <crowbar_admin_node >:7180
Hadoop Name Node	Hadoop Name Node	50070	http:// <master_name_node >:50070
Hadoop Secondary Name Node	Hadoop Secondary Name Node	50090	http:// <secondary_name_node >:50090
Hadoop Data Node	Hadoop Data Node	50075	http:// <data _node>: 50075
Hadoop Job Tracker Web	Hadoop Job Tracker Node	50030	http:// <job_tracker_node>: 50030
Hadoop Task Tracker Web	Task Tracker Node	50060	http://<task_tracker_node>:50060

Note: The IP address (192.168.124.10) is the default address. Replace it with the address assigned to the Crowbar Admin node. Nagios, Ganglia and Chef can be accessed directly from a web browser or via selecting one of the links on the Crowbar Dashboard.

Cloudera Manager Overview

Cloudera Manager deploys and centrally operates a complete Hadoop stack. The application automates the installation process, reducing deployment time from weeks to minutes, gives you a cluster-wide, real time view of the services running and the status of their hosts, provides a single, central place to enact configuration changes across your cluster; and incorporates a full range of reporting and diagnostic tools to help you optimize cluster performance and utilization. Cloudera Manager provides full lifecycle management for Hadoop deployments.

Functionality Outline

- Installs the complete Hadoop stack in minutes via a wizard-based interface.
- Gives you complete, end-to-end visibility and control over your Hadoop cluster from a single interface.
- Let's you set server roles and configure services across the cluster.
- Let's you gracefully start, stop and restart of services as needed.
- Shows information pertaining to hosts in your cluster including status, resident memory, virtual memory and roles.

Table 4 Differences between Cloudera Manager Free Edition and Cloudera Manager

Feature	Cloudera Manager Free Edition	Cloudera Manager
Max number of hosts supported	50	Unlimited
Automated installer	✅	✅
Host Monitoring	✅	✅
Secure communication between server and agents (TLS)	✅	✅
Service and Configuration Management	✅	✅
Manage HDFS, MapReduce , HBase, Hue, Oozie, and Zookeeper	✅	✅
Configuration audit trails	✅	✅
Workflows to start, stop, restart, add, delete, and decommission role	✅	✅

instances		
Configuration versioning and history	⚠	✓
Support for Kerberos	⚠	✓
Service Monitoring	⚠	✓
Status and health summary	⚠	✓
Proactive health tests	⚠	✓
Log Search and Management	⚠	✓
Events Management	⚠	✓
Alerts	⚠	✓
Activity Monitoring	⚠	✓
Operational Reports	⚠	✓
Global Time Control for historical diagnosis	⚠	✓
Support integration		

Barclamps

Table 5 Barclamp Descriptions

Barclamp	Description
Cloudera Manager	Provides end-to-end management for apache Hadoop with the ability to deploy and centrally operate a complete Hadoop stack gives you a cluster wide, real time view of nodes and services running and provides a single central place to enact configuration changes across your cluster. Cloudera Manager incorporates a full range of reporting and diagnostic tools to help you optimize cluster performance and utilization.
Hive	Data warehouse that infrastructure provides SQL based data summarization and ad hoc querying.
Pig	Platform for analyzing large data sets that consists of a high-level language for expressing data algorithms.
Sqoop	SQL based command-line tool to assist with HDFS data import/export (SQL-to-Hadoop).
ZooKeeper	High-performance coordination service for distributed applications. ZooKeeper provides primitives such as distributed locks which can be used for building large scale distributed processing applications.

Cloudera Manager Barclamp

The Cloudera Manager Barclamp performs all the low level operating system configuration setup for the Hadoop cluster and installs the Cloudera Manager server setup in order to prepare for Hadoop cluster deployment.

Table 6 Open File Handles Configuration Parameters

Name	Description	Required	Default
Map/Reduce	Maximum number of Map/Reduce open file handles.	True	32768
HDFS	Maximum number of HDFS open file handles.	True	32768
HBASE	Maximum number of HBASE open file handles.	True	32768

Note: The values specified below must match the configuration setup in Cloudera Manager. See the section titled "Monitoring Database Setup Screen" within this document.

Table 7 Cloudera Service Monitor Database Parameters.

Name	Description	Required	Default
Database Name	Database name.	True	service_monitor
Database User	Login user name.	True	scm
Database Password	Login password.	True	crowbar

Table 8 Cloudera Activity Monitor Database Parameters

Name	Description	Required	Default
Database Name	Database name.	True	activity_monitor
Database User	Database Login user name.	True	scm
Database Password	Database Login password.	True	crowbar

Table 9 Cloudera Resource Manager Database Parameters.

Name	Description	Required	Default
Database Name	Database name.	True	resource_manager
Database User	Database Login user name.	True	scm
Database Password	Database Login password.	True	crowbar

Cloudera Manager Installation Overview

After the Cloudera Manager Barclamp has been deployed from Crowbar, you must run the Cloudera Manager configuration wizard in order to fully deploy the Hadoop cluster. This operation will perform the following tasks:

- Using SSH, discover the cluster hosts you specify via IP address ranges or hostnames.
- Install the Cloudera Manager Agent and CDH3 (including Hue) on the cluster data nodes.
- Install the Oracle JDK if it's not already installed on the cluster hosts.
- Configures the package repositories for Cloudera Manager, CDH3 and the Oracle JDK.
- Allow you to select and configure optional eco-system components.
- Determine mapping of services to host.
- Suggest a Hadoop configuration and automatically starts the Hadoop services.

You can choose to abort the Cloudera Manager Agent and CDH3 installation process and Cloudera Manager wizard will automatically revert and completely rollback the installation process for any uninstalled components. Installed components are not uninstalled during an abort.

Cloudera Manager Node Inventory Page

Once the Cloudera barclamp has been deployed, from the Edit Proposal page, there is a link below the Proposal Attributes section called "Cloudera Manager Nodes." Clicking on this link will display a page titled "Cloudera Node Inventory." This screen is pictured in Figure 4-1 below. You may print this page as it will be very useful during the Cloudera Manager installation to ensure the correct nodes are selected for their intended Cloudera Manager roles.

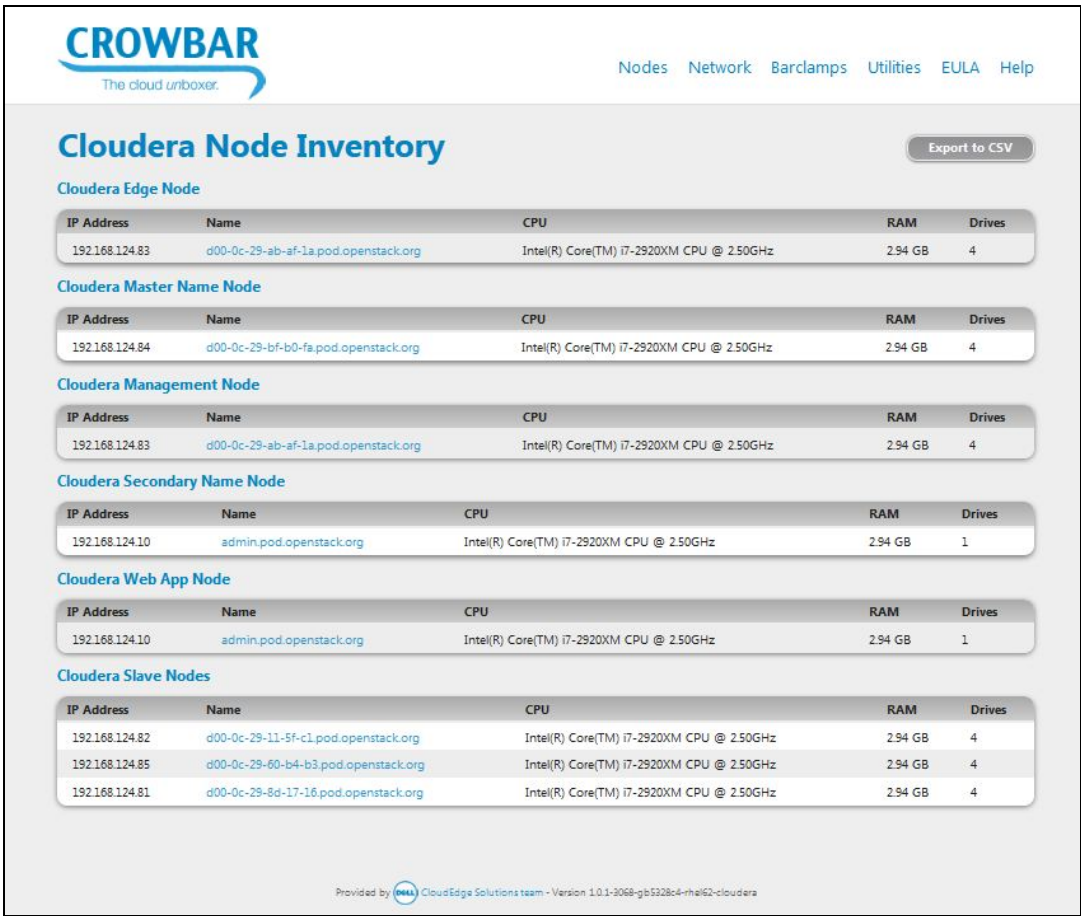


Figure 4-1: Cloudera Node Inventory page.

Note: You may also export this data to a comma separated value file by selecting the "Export to CSV" button at the top of the page.

Note: The Cloudera Manager Administration console supports the following web browsers:

- Internet Explorer 8 and 9.
- Google Chrome.
- Safari 5.
- Firefox 3.6 and later.

To start the Cloudera Manager Administration Console;

- In a web browser, type the following URL: http(s):// IP_ADDRESS: PORT_NUMBER.
  - IP_ADDRESS is the name or IP address of the host machine where the Cloudera Manager Server is installed.
  - PORT_NUMBER is the default port number (7180).
  - Crowbar Installation defaults are Crowbar Admin Node on port 7180 (http://192.168.124.10:7180).
- Log into the Cloudera Manager Admin Console. The default login credentials are;

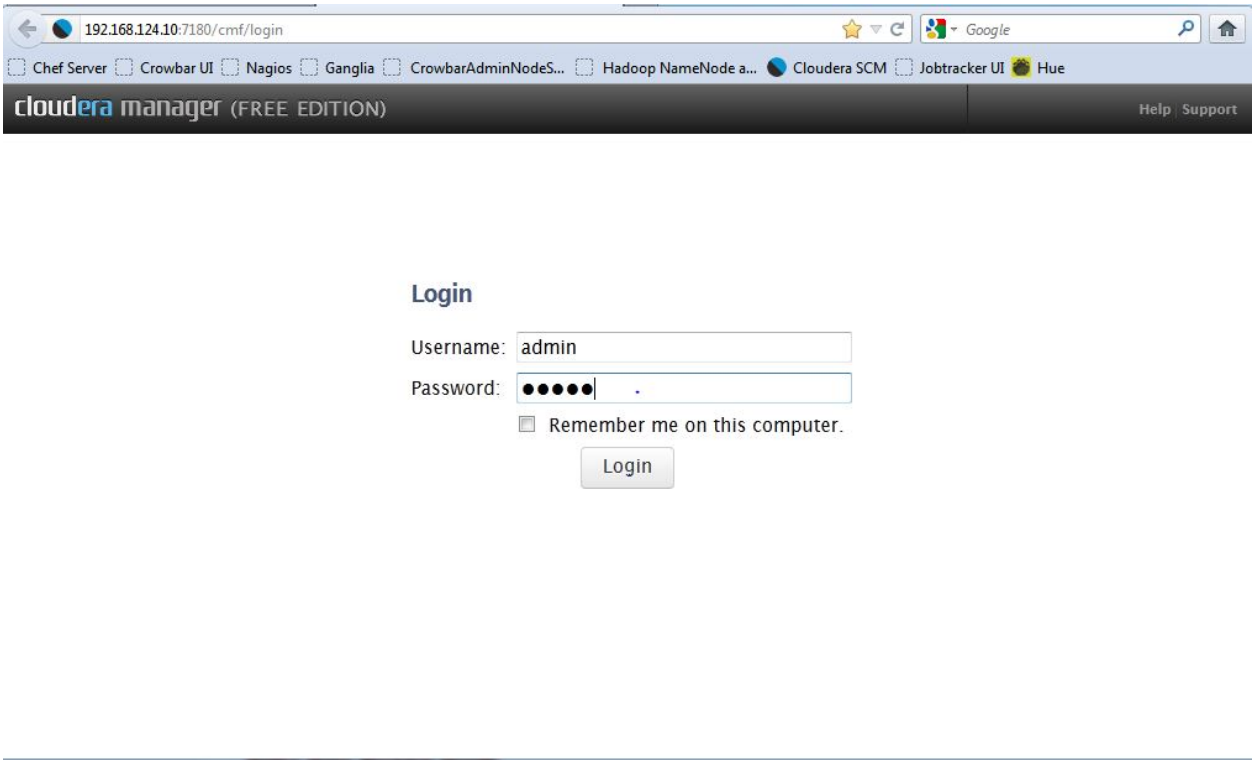
  - Username: admin
  - Password: admin
- You can also access the Cloudera Manager Administration Console from the Crowbar User Interface using the link located on the crowbar admin node view page (Cloudera Manager).

Note: For security, you should change the password for the default admin user account as soon as possible. This option from the Cloudera Manager application under the Administration->Password tab.

Login Screen

- Enter the user login name and password (default=admin, admin).
- Check ***Remember me on this computer*** to store the browser session data if desired.
- Click the ***Login*** button to proceed.

Figure 1 Login Page

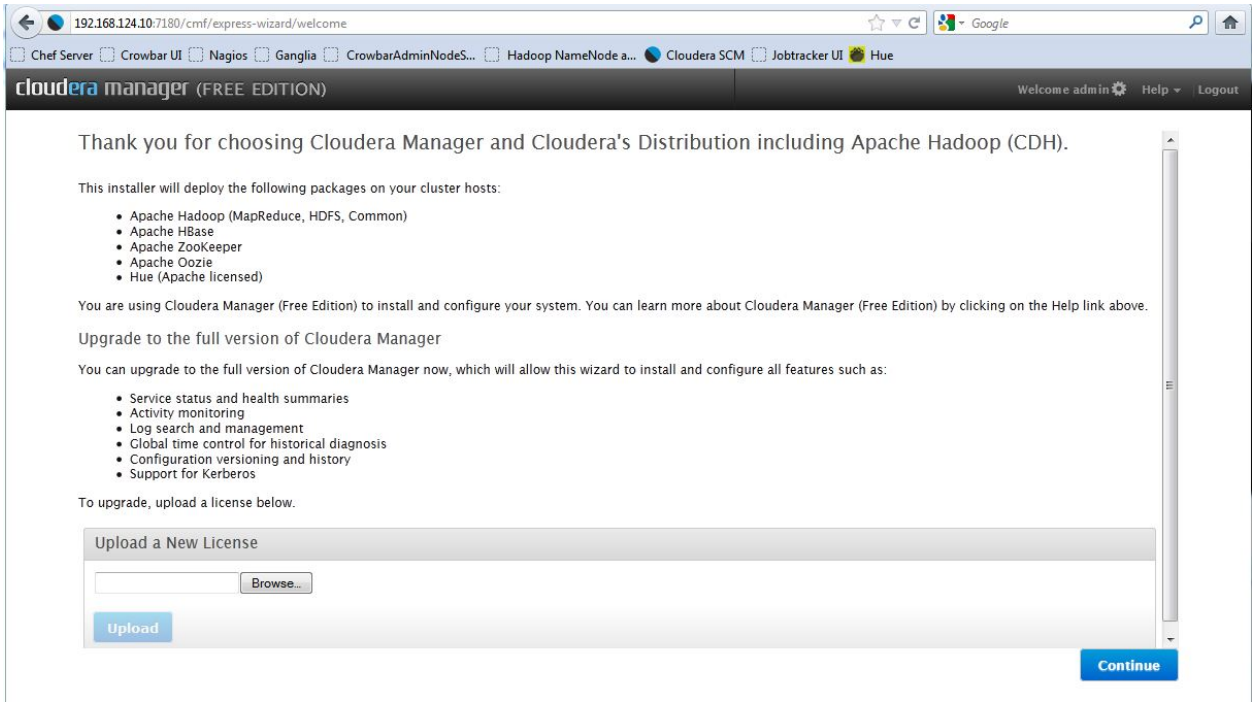


License Key Entry Screen

Note: Applying the license key is an optional step and you can always enter the license key later on in the process by clicking on the ***welcome->admin*** link in the Cloudera Manager administration console. This menu option is located at the top right side of the Cloudera Manager administration console screen.

- If you have obtained a Cloudera Manager license key and you wish to upgrade to the Cloudera Manager Enterprise edition, you can enter the license key at this point.
- Hit the *Browse* button and select the license key location on the local file system.
- Hit the *Upload* button to register the license key.
- Hit the *Continue* Button to proceed.
- Once the license key has been uploaded, the Cloudera Manager application will ask you to restart the Cloudera Manager server for it to take effect. You need to open an SSH console into the node which has the Crowbar web application role applied to it (login=root/crowbar) and type the following commands;
    - cd /etc/init.d
    - service cloudera-scm-server restart
    - Once the Cloudera manager server has been restarted, you need to restart the browser and log back into the Cloudera Manager Administration console.

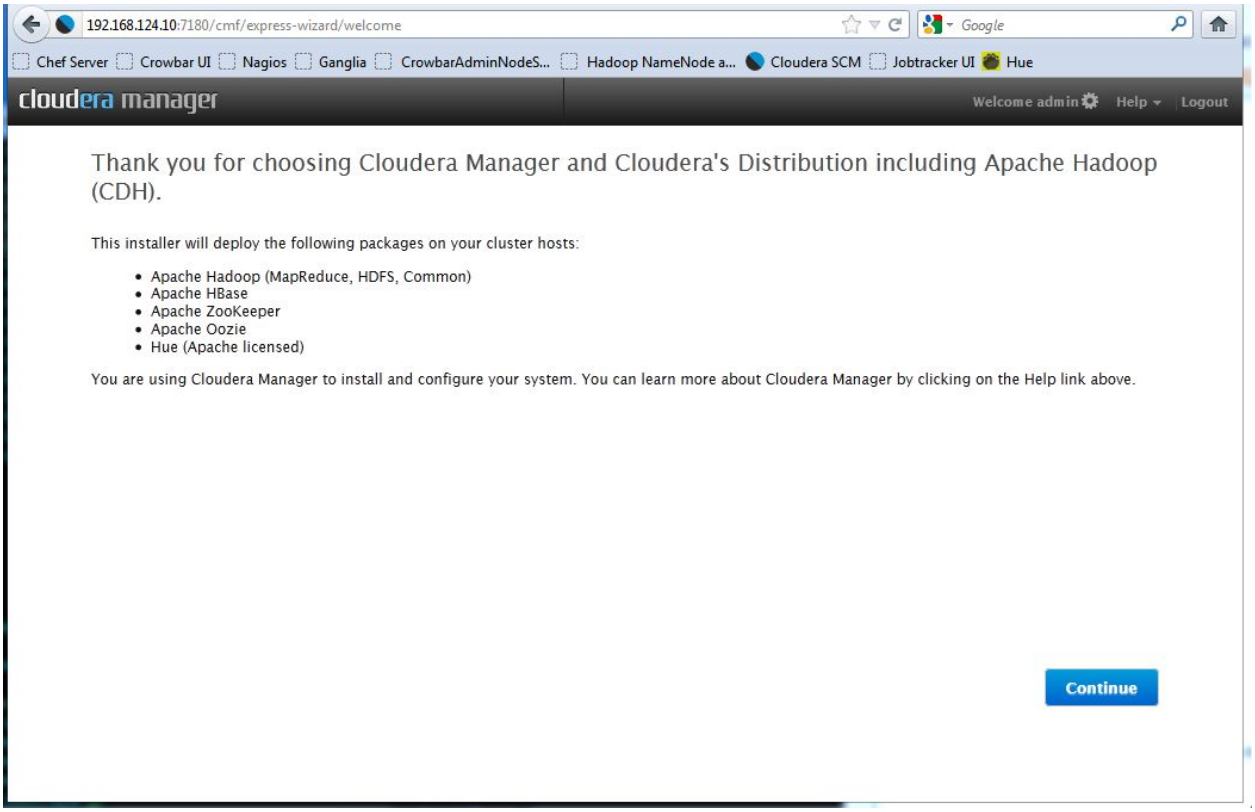
Figure 2 License Key Entry Screen



License Key Confirmation Screen

- If you entered the Cloudera Manager License key, you will see this additional screen.
- Hit the *Continue* Button to proceed.

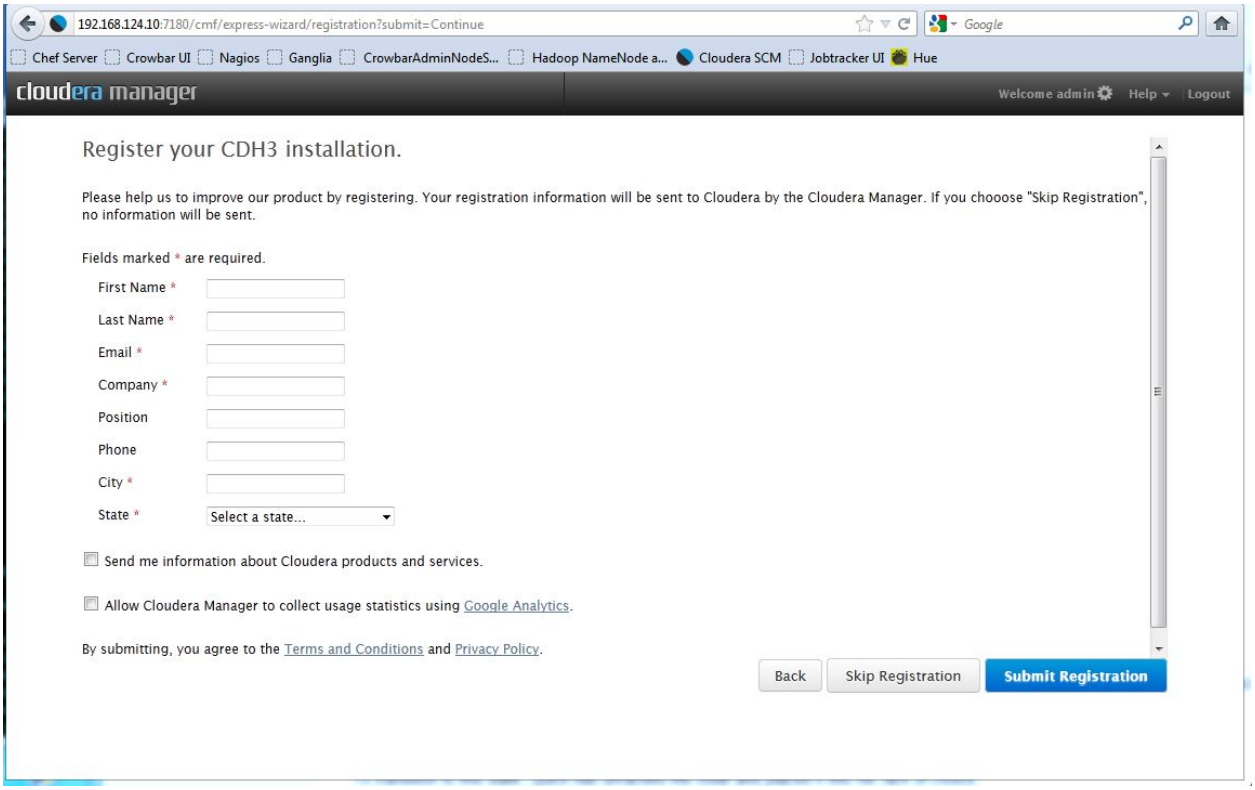
Figure 3 License Key Confirmation Screen



Cloudera Registration Screen

- If you wish to register with Cloudera, fill out the registration form information and click the ***Submit Registration*** button.
- If you do not want to register, click the ***Skip Registration*** button to proceed.

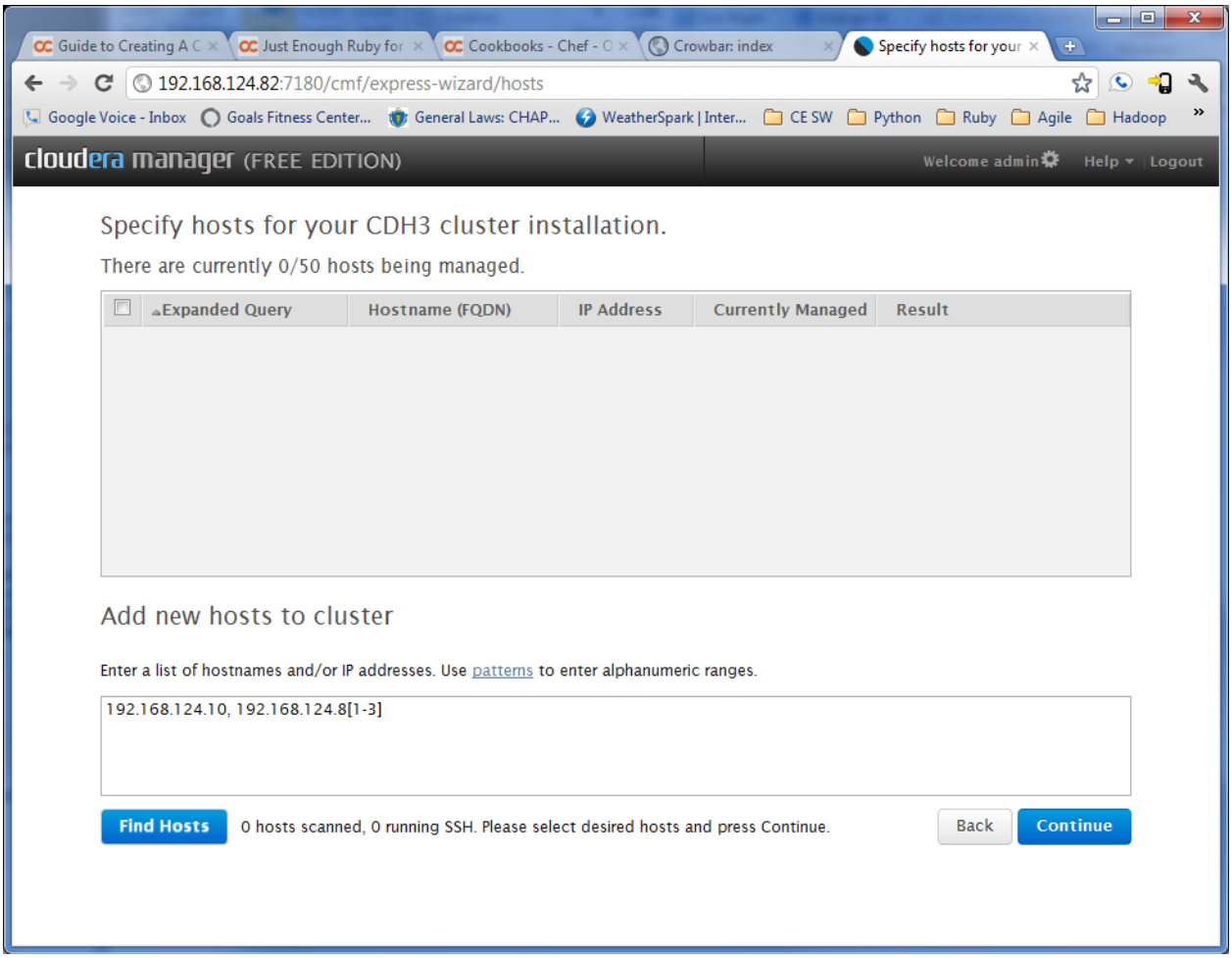
Figure 4 Registration Screen



Node Search Screen

- Enter the IP address range for all nodes on Hadoop cluster. This should be an alphanumeric search pattern and the crowbar admin node should be included in the search. For example;
  - **192.168.124.10, 192.168.124.8[1-3]** will attempt to discover 192.168.124.10, 192.168.124.81, 192.168.124.82 and 192.168.124.83,
- Click the ***Find Hosts*** button to search for available hosts within the specified IP address range.

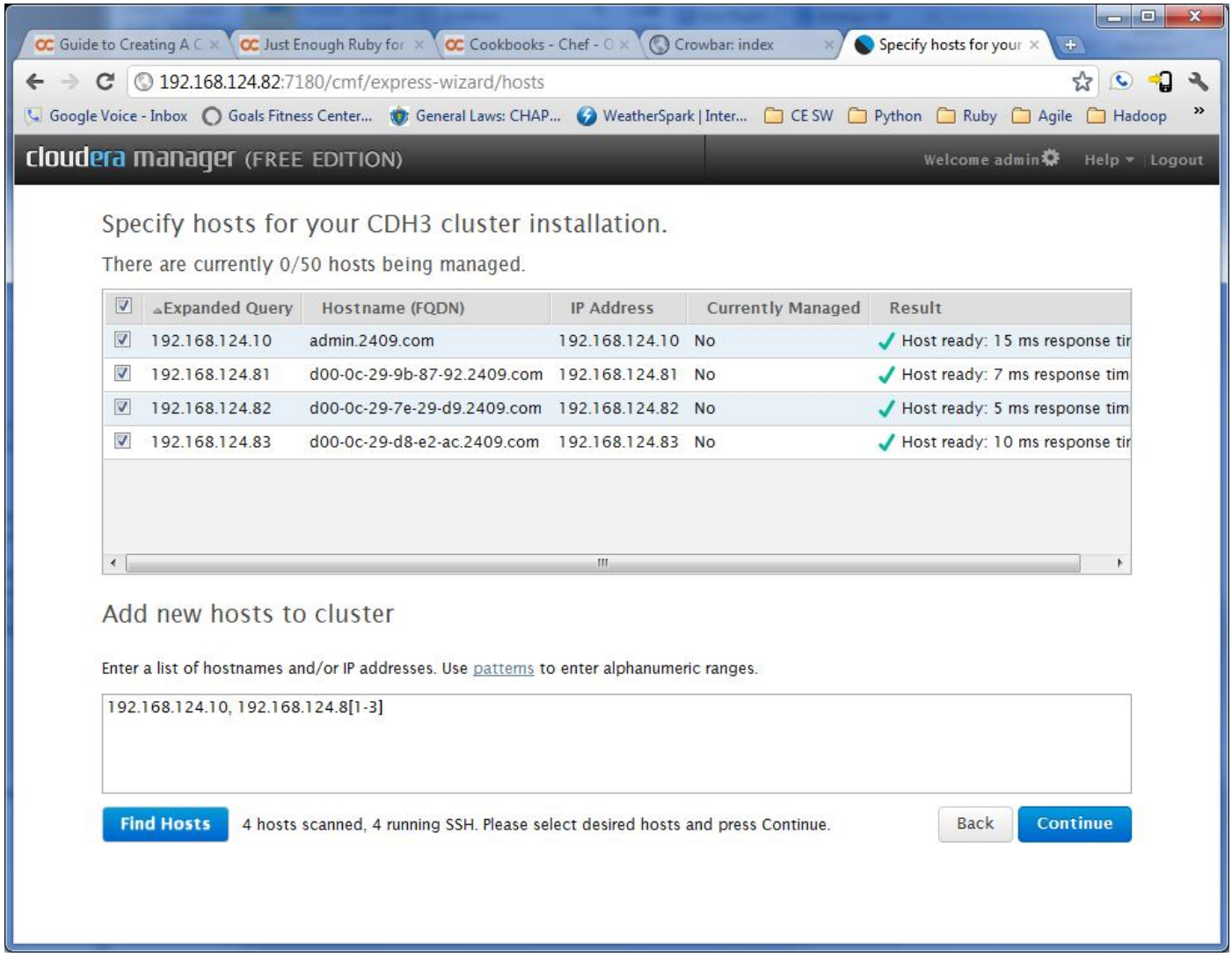
Figure 5 Cloudera Cluster Node Search Screen



Node Search Results Screen

- Verify that all your hosts are discovered.
- Make any cluster configuration adjustments by deselecting nodes.
- Click the *Continue* button to proceed.

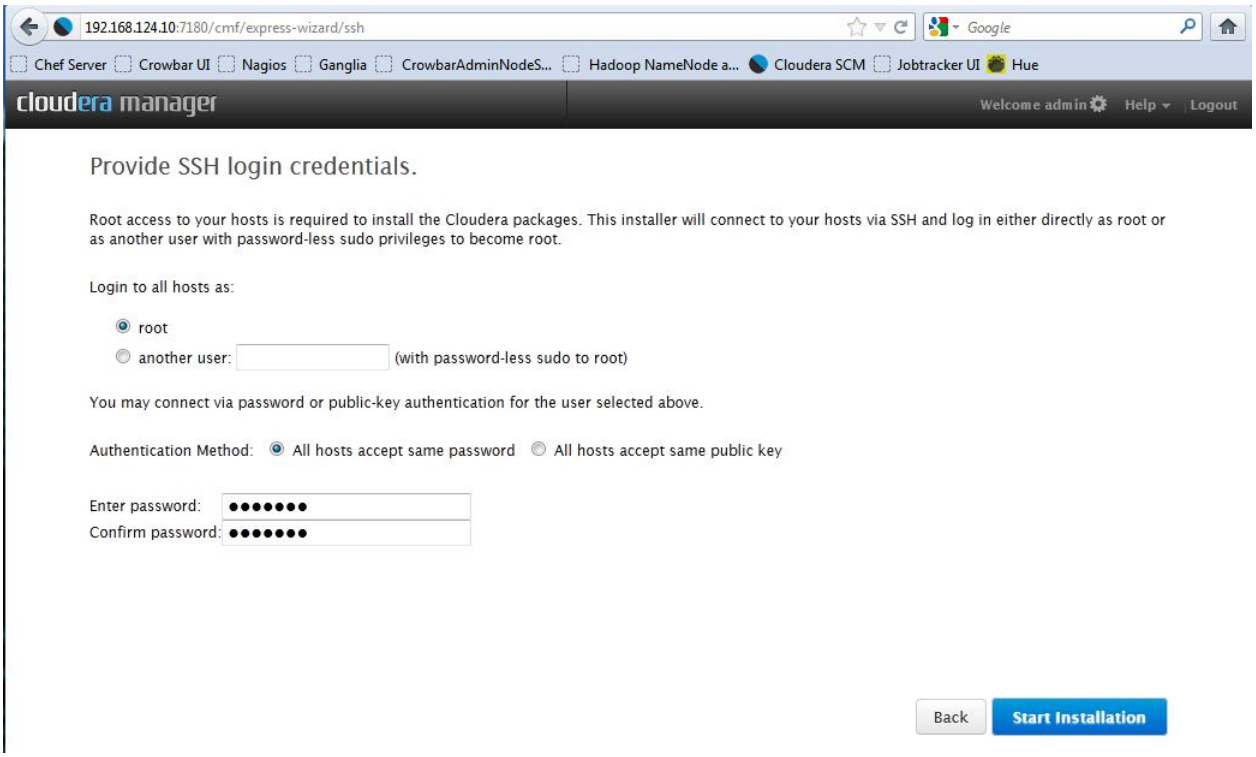
Figure 6 Node Search Results Screen



SSH Credentials Screen

- Select the *Login to all hosts as root* radio button.
- Select the *All hosts accept same password* radio button.
- Enter the SSH login credentials for the cluster (default=root, crowbar).
- Click the *Start Installation* button to proceed.

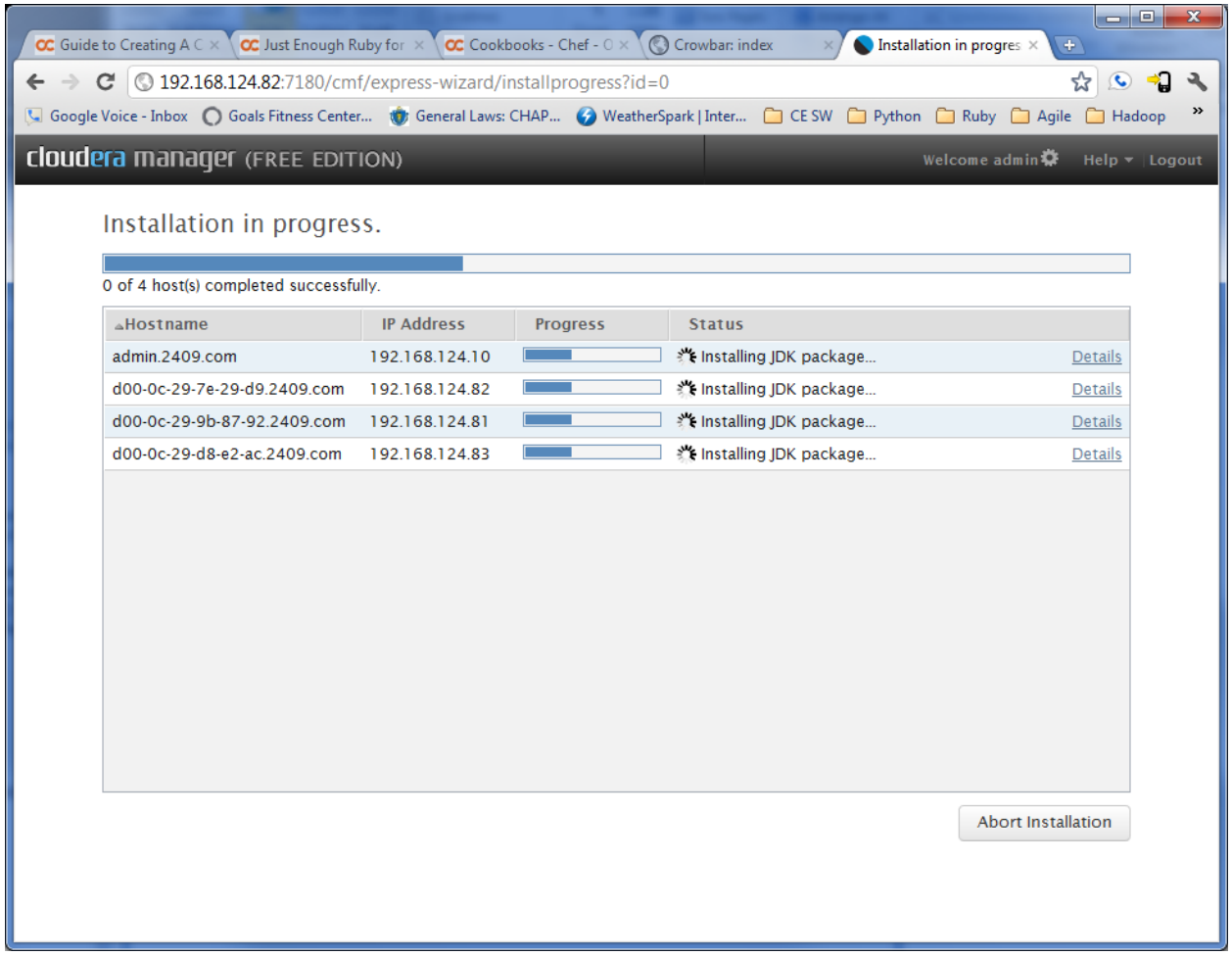
Figure 7 SSH Credentials Screen



Package Install Screen

- You will see bar graphs next to each node and the name of the package it is installing.
- Wait for the installation process to complete.

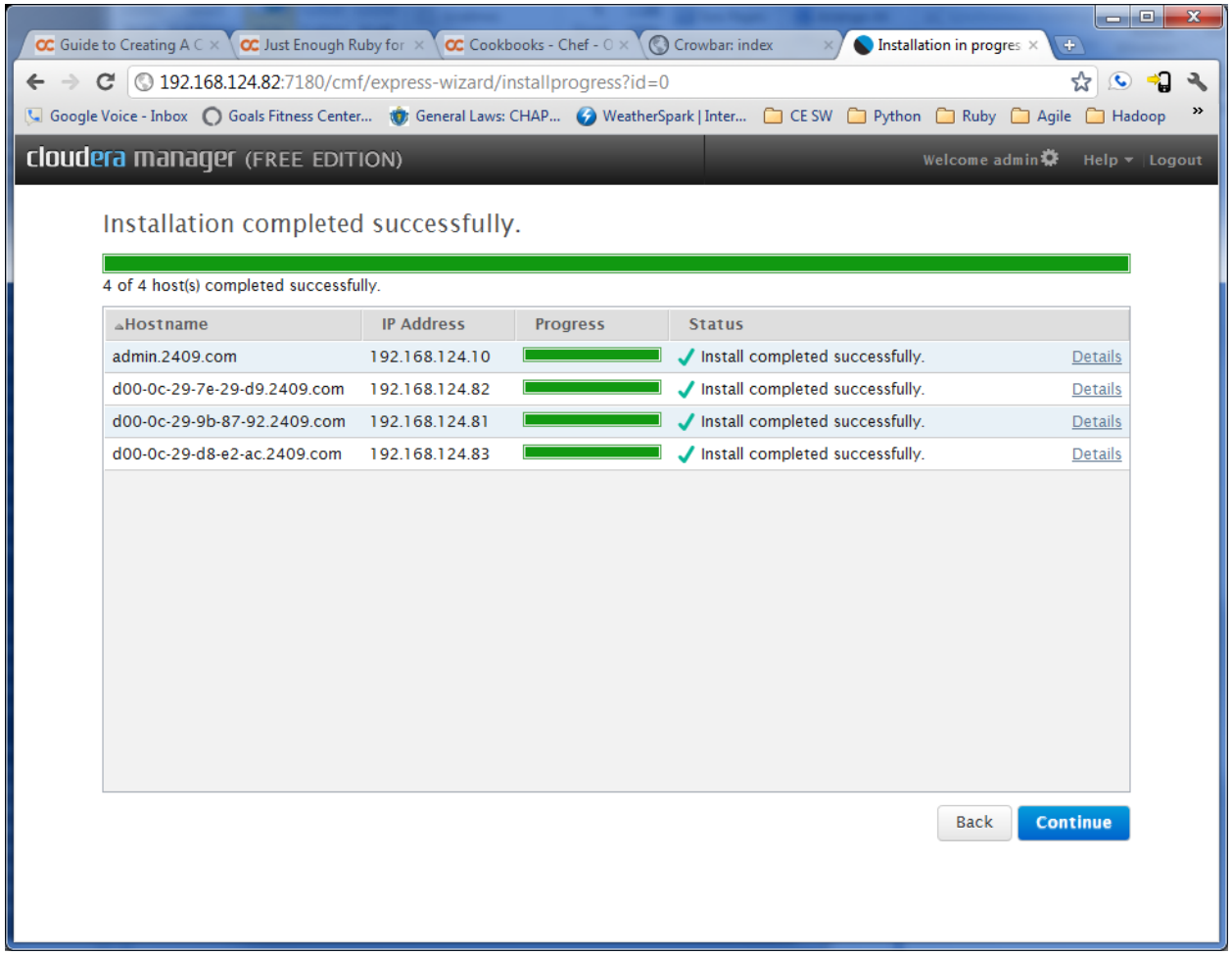
Figure 8 Package Install Screen



Package Install Completion Screen

- You will see the following screen when the installation process completes.

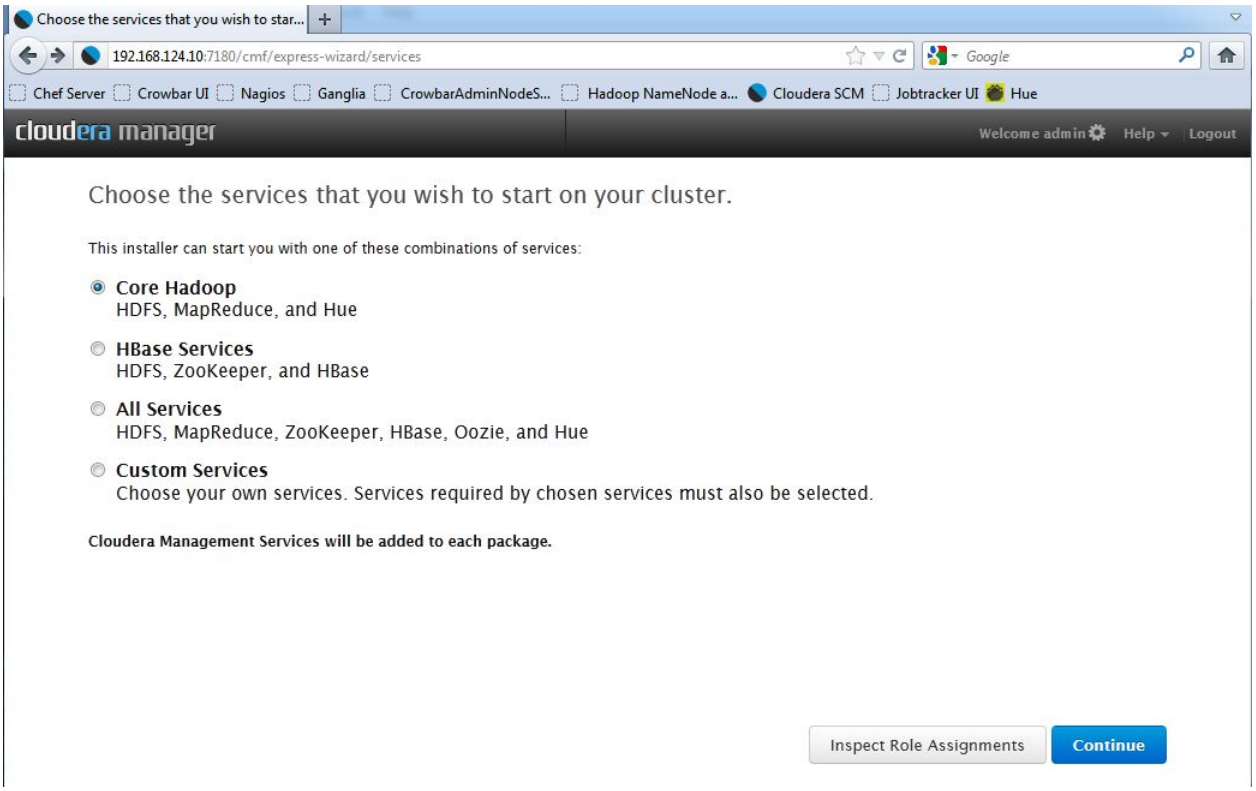
Figure 9 Package Install Completion Screen



Service Selection Screen

- Select the services that you want to install.
- You can install *All Services* now or *Core Services* and add others as needed.
- Click on the *Inspect Role Assignments* button to configure the Hadoop cluster services. Do not select *Continue* as this will give you the default role assignments which are probably not what you want.

Figure 10 Service Selection Screen

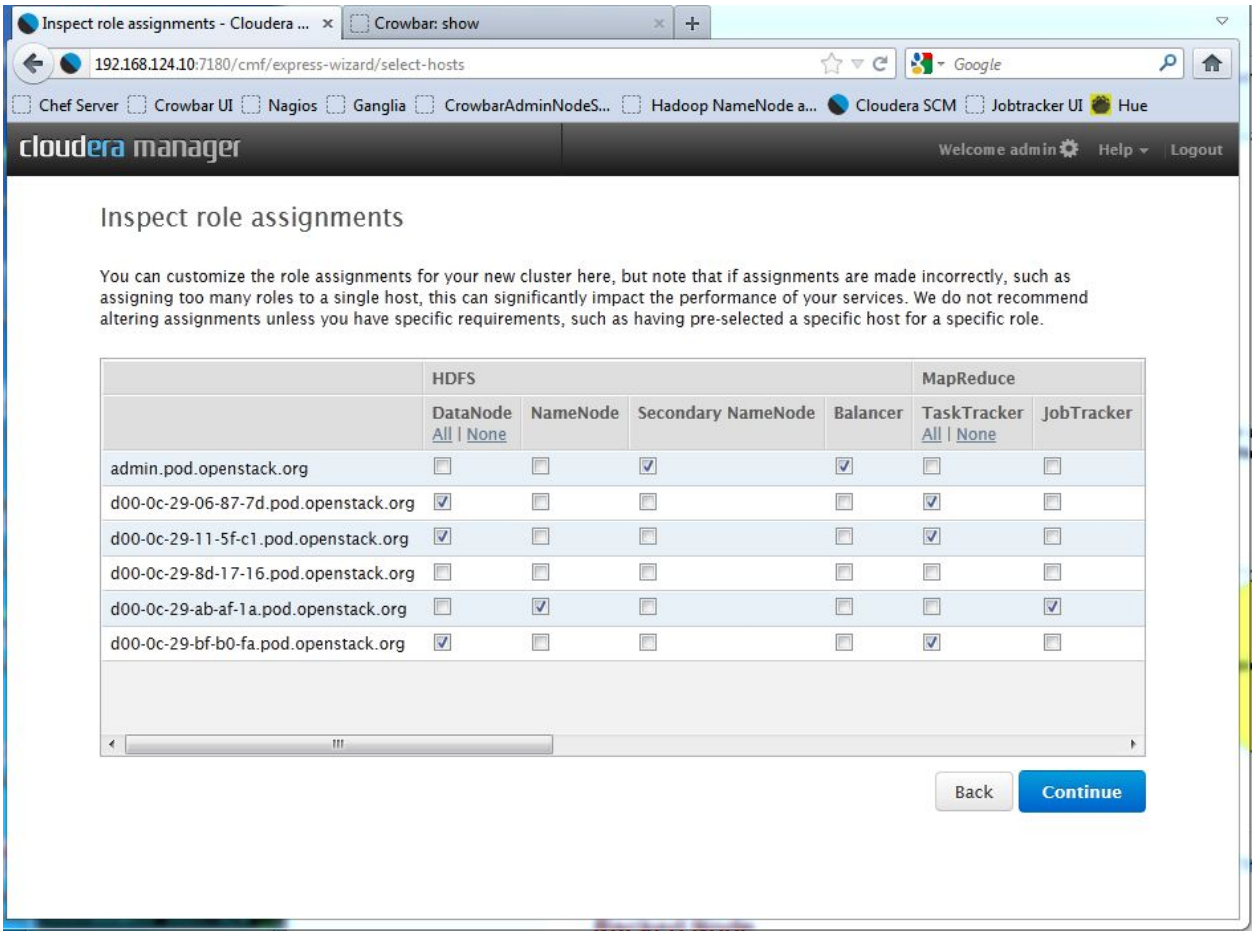


Inspect Role Assignments Screen # 1

- Select the role assignments for Hadoop cluster deployment (HDFS/Map Reduce).
- Recommended settings for the Dell Reference Architecture:
  - CM DataNode role – Crowbar nodes which contains the *clouderamanager-slavenode* role.
  - CM NameNode – Crowbar node which contains the *clouderamanager-masternamenode* role.
  - CM Secondary NameNode – Crowbar node which contains the *clouderamanager-secondarynamenode* role.
  - CM Balancer – Any node of your choice.
  - CM TaskTracker roles – Crowbar node which contains the *clouderamanager-slavenode* role.
  - CM JobTracker role– Crowbar node which contains the *clouderamanager-masternamenode* role.
- Please refer to the next diagram (Figure 8 – Screen #2) before clicking the ***Continue*** button.

Note: The Cloudera Node Inventory page you printed /exported from within the Cloudera Manager barclamp page in Crowbar is a very useful document to have for this step to ensure the roles selected in Cloudera Manager are assigned to nodes which have been provisioned (RAID, BIOS, etc ...) specifically for that purpose.

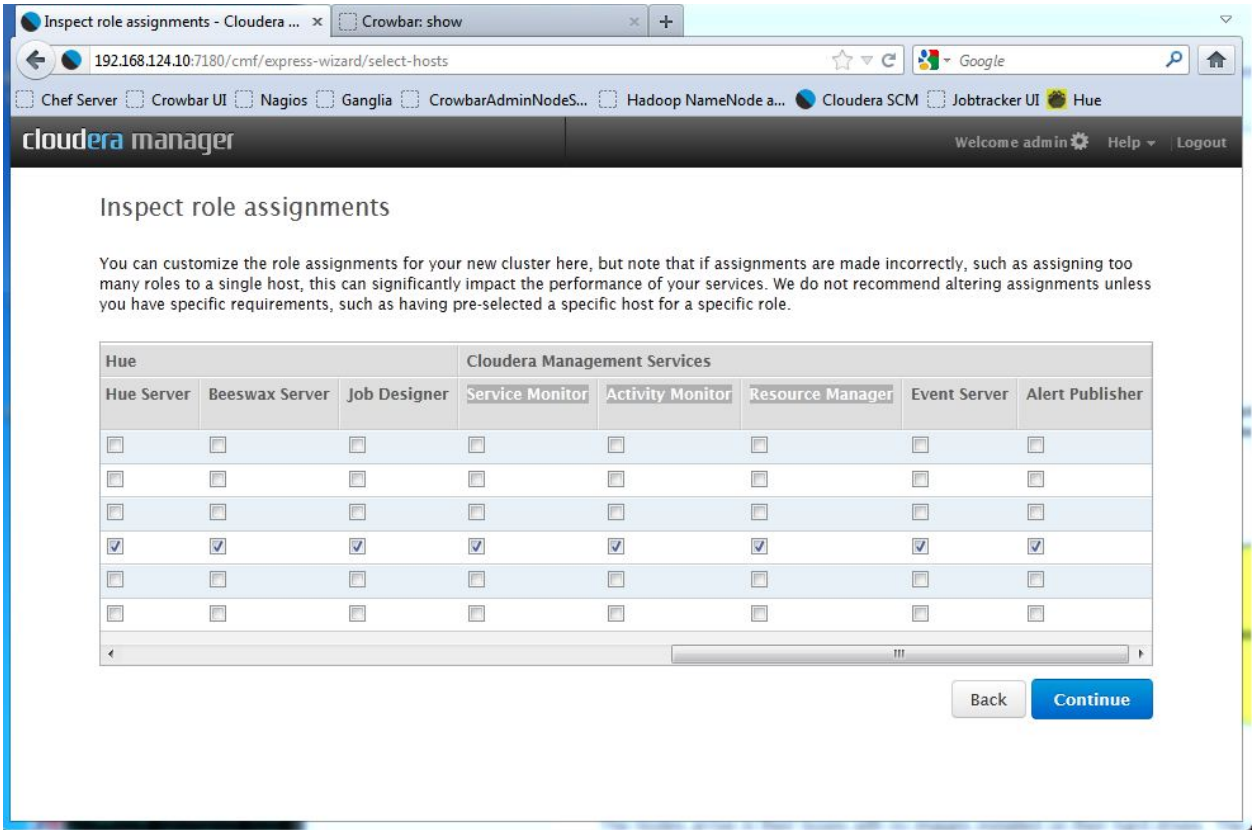
Figure 11 Inspect Role Assignments Screen # 1



Inspect Role Assignments Screen # 2

- If you entered the Cloudera Manager License key, you will see this additional screen.
- Select the role assignments for Hadoop add-ons services and monitoring services (Activity Monitor, Service Monitor Resource Manager).
- The Cloudera Manager monitoring services roles (Activity Monitor, Service Monitor Resource Manager) need to be assigned to the Crowbar node that has the *cloueramanager-mgmtservices role applied to it*. The other services can be assigned to any node on the cluster (recommended setting –Crowbar edge node).
- Click the ***Continue*** button to proceed.

Figure 12 Inspect Role Assignments Screen #2

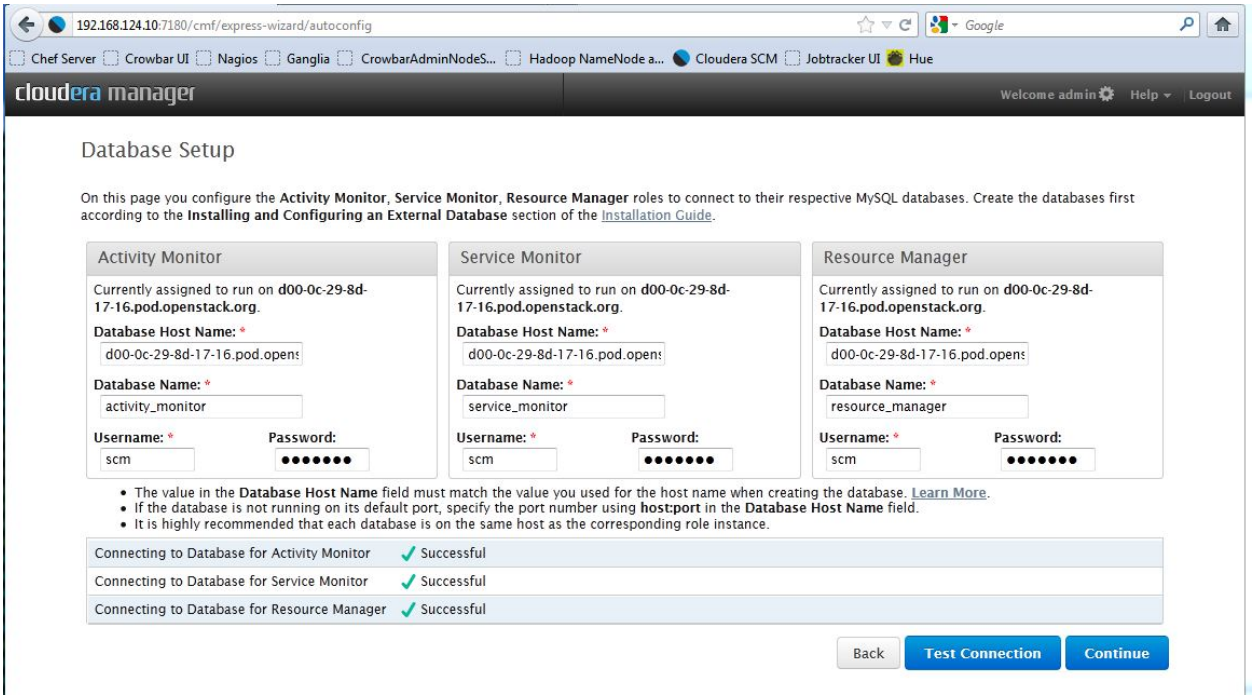


Monitoring Database Setup Screen

Note: These database settings need to match the values specified in the crowbar cloudera-manager barclamp proposal.

- If you entered the Cloudera Manager License key, you will see this additional screen.
- The Cloudera Manager monitoring services roles (Activity Monitor, Service Monitor Resource Manager) need to be assigned to the Crowbar node that has the *clouderamanager-mgmtservices role applied to it*.
- Under the ***Activity Monitor*** section, enter the database name, username and password (default=activity_monitor, scm, crowbar).
- Under the ***Service Monitor*** section, enter the database name, username and password (default=service_monitor, scm, crowbar).
- Under the ***Resource Manager*** section, enter the database name, username and password (default= resource_manager, scm, crowbar).
- Click the ***Test Connection*** button to make sure you can connect to all the databases.
- Click the ***Continue*** button to proceed.

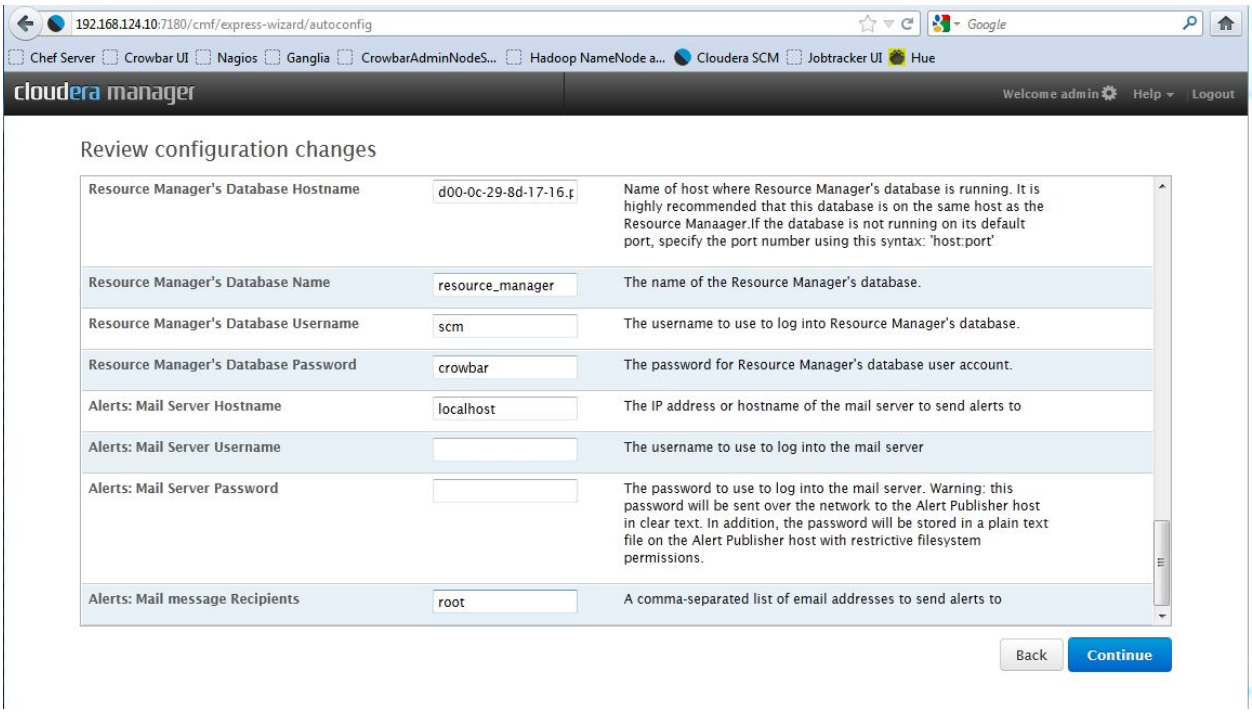
Figure 13 Monitoring Database Setup Screen



Review Configuration Changes Screen

- If you entered the Cloudera Manager License key, you will see this additional screen.
- Set the mail server hostname for alerts (localhost).
- Set the mail server message recipients for alerts.
- Click the *Continue* button to proceed.

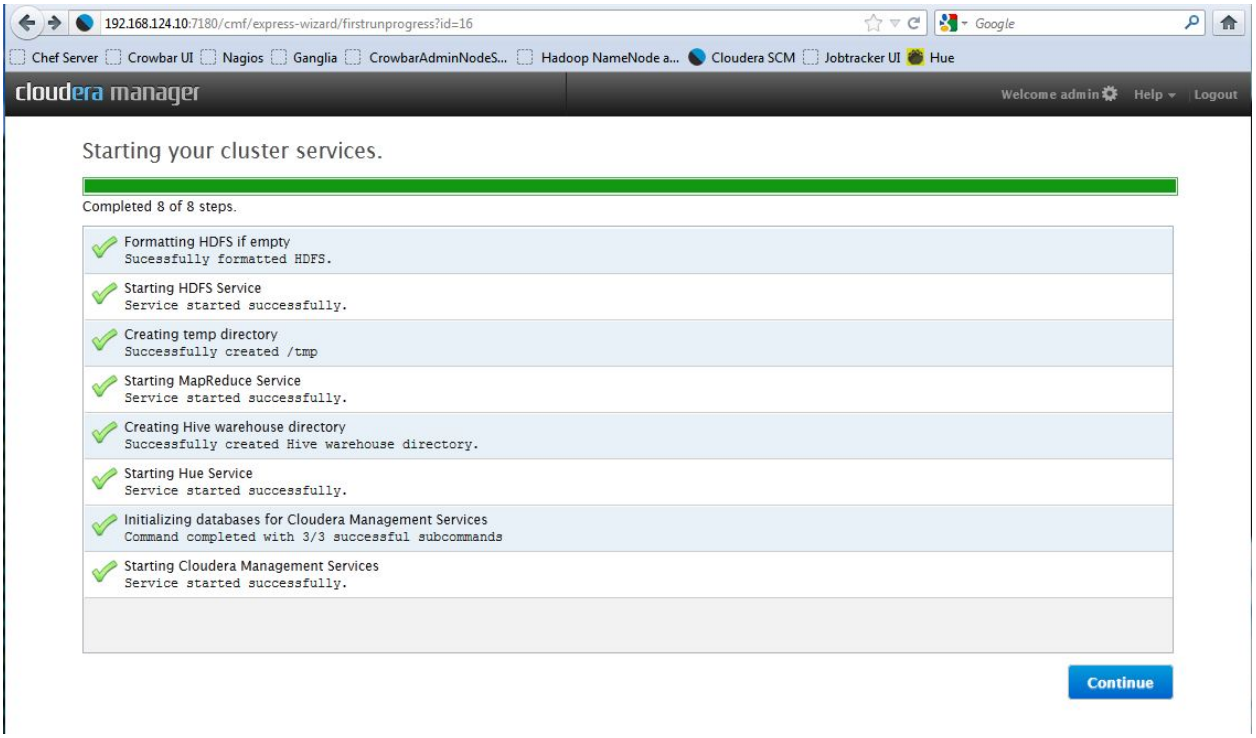
Figure 14 Review Configuration Changes Screen



Cluster Services Initialization Screen

- Wait for the Hadoop cluster configuration process to complete.
- Click the *Continue* button to proceed.

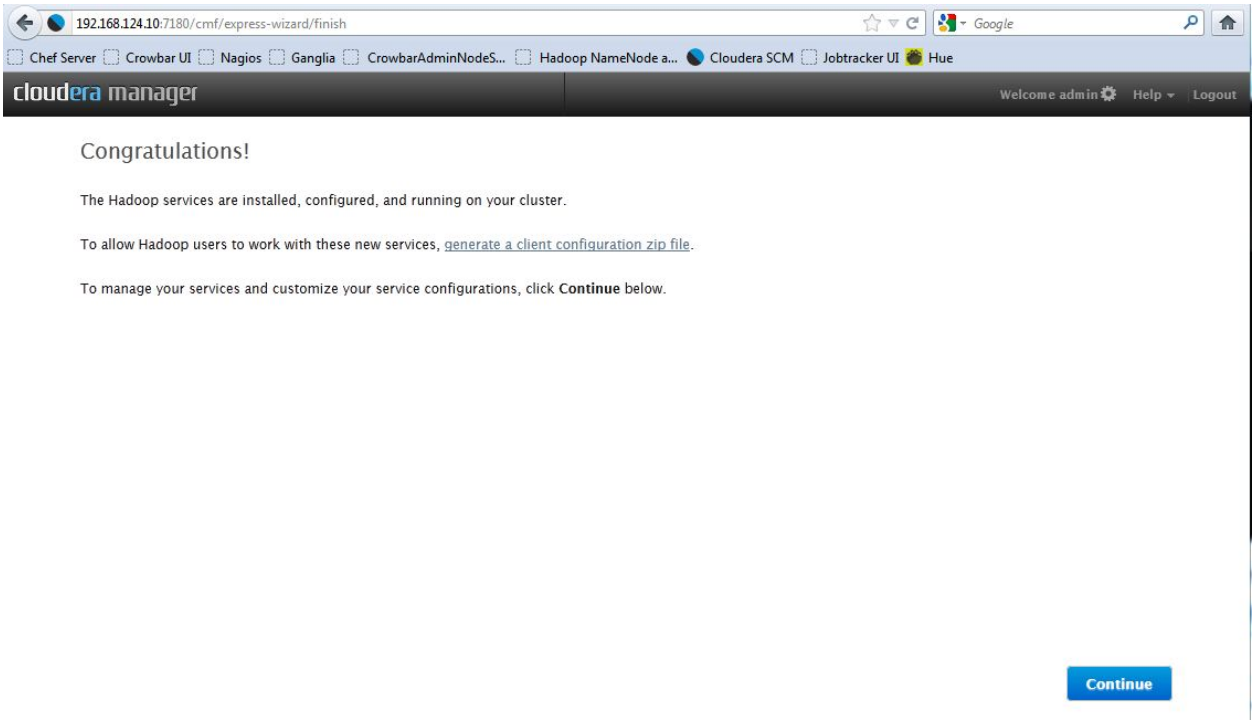
Figure 15 Cluster Services Initialization Screen



Configuration Completion Screen

- If the Hadoop configuration steps complete successfully, you will see the final Cloudera Manager confirmation screen.
- Click the *Continue* button to start Cloudera Manager.

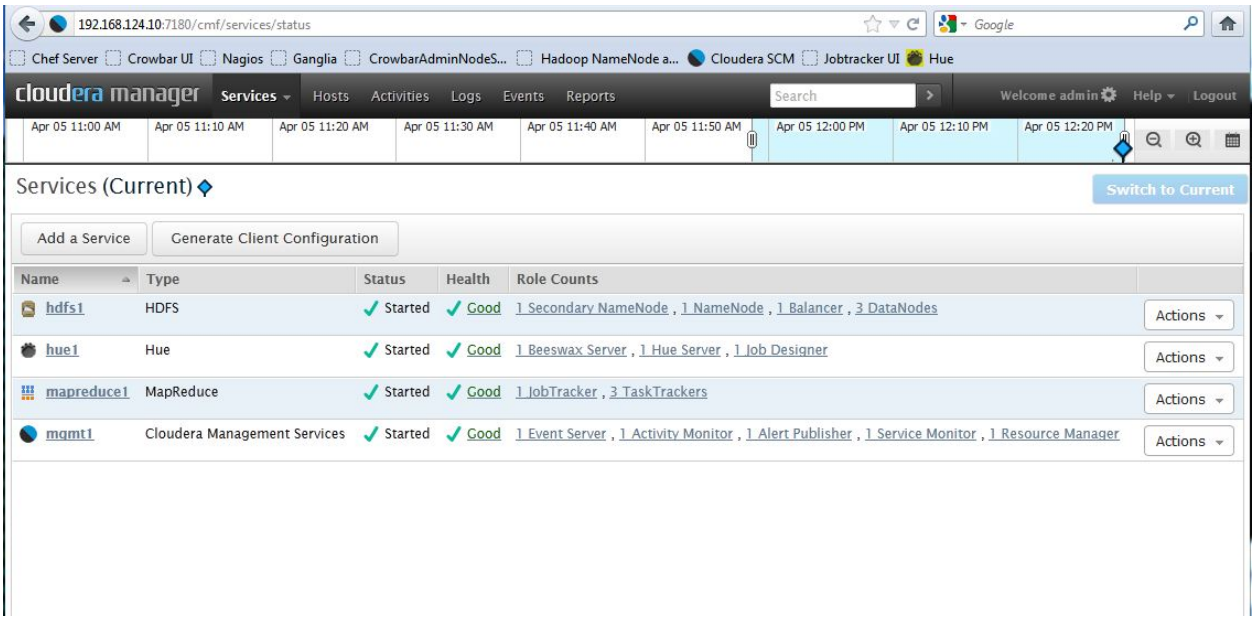
Figure 16 Configuration Completion Screen



Service Display Screen

- This is the normal startup screen after Cloudera Manager has completed the installation steps.
- Please refer to the Cloudera Manager User Guide for more detailed information on Cloudera Manager operation.

Figure 17 Service Display Screen



Pig Barclamp

Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

Pig's infrastructure layer consists of a compiler that produces sequences of MapReduce programs, for which large-scale parallel implementations already exist (e.g., the Hadoop subproject). Pig's language layer currently consists of a textual language called Pig Latin, which has the following key properties:

- **Ease of programming:** It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.
- **Optimization opportunities:** The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.
- **Extensibility:** Users can create their own functions to do special-purpose processing.

Table 4-34: Pig Barclamp Parameters

Name	Description	Required	Default
java_home	JAVA_HOME environment variable.	true	/usr/java/jdk1.6.0_27/jre
log4jconf	log4jconf log4j configuration file.	true	./conf/log4j.properties
brief	brief logging - no timestamps.	true	false
cluster	Clustername, name of the hadoop jobtracker. If no port is defined port 50020 will be used.	false	
debug_level	Debug level, INFO is default.	true	INFO
file	A file that contains pig script.	false	
jar	Load jarfile, colon separated.	false	
verbose	Verbose print all log messages to screen (default to print only INFO and above to screen).	true	false
exectype	Exectype local or mapreduce - mapreduce is default.	true	mapreduce
ssh_gateway	HOD gateway property.	false	
hod_expect_root	HOD expect root property.	false	
hod_expect_uselatest	HOD use latest root property.	false	
hod_command	HOD command root property.	false	
hod_config_dir	HOD config directory property.	false	
hod_param	HOD param property.	false	
pig_spill_size_threshold	Do not spill temp files smaller than this size (bytes).	true	5000000
pig_spill_gc_activation_size	EXPERIMENT: Activate garbage collection when spilling a file bigger than this size (bytes). This should help reduce the number of files being spilled.	true	40000000
log_file	Log file location.	false	

Hive Barclamp

Hive is a data warehouse system for Hadoop that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in Hadoop compatible file systems. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. This language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.

Table 4-41: Hive Barclamp Parameters

Name	Description	Required	Default
hive_exec_scratchdir	Scratch space for Hive jobs.	true	/tmp/hive-${user.name}
hive_metastore_local	Controls whether to connect to remove metastore server or open a new metastore server in Hive Client JVM.	true	true
javax_jdo_option_ConnectionURL	JDBC connect string for a JDBC metastore.	true	jdbc:derby:;databaseName=metastore_db;create=true
javax_jdo_option_ConnectionDriverName	Driver class name for a JDBC metastore.	true	org.apache.derby.jdbc.EmbeddedDriver
hive_metastore_metadb_dir	The location of filestore metadata base dir.	true	file:///var/metastore/metadb/
hive_metastore_uris	Comma separated list of URIs of metastore servers. The first server that can be connected to will be used.	true	file:///var/metastore/metadb/
hive_metastore_warehouse_dir	The location of the default database for the warehouse.	true	/user/hive/warehouse
hive_metastore_connect_retries	Number of retries while opening a connection to metastore.	true	5
hive_metastore_rawstore_impl	Name of the class that implements org.apache.hadoop.hive.metastore.rawstore interface. This class is used to store and retrieval of raw metadata objects such as table, database.	true	org.apache.hadoop.hive.metastore.ObjectStore
hive_default_fileformat	Default file format for CREATE TABLE statement. Options are TextFile and SequenceFile.	true	TextFile
hive_map_aggr	Whether to use map-side aggregation in Hive Group By queries.	true	false
hive_join_emit_interval	How many rows in the right-most join operand Hive should buffer before emitting the join result.	true	1000
hive_exec_script_maxerrsize	Maximum number of bytes a script is allowed to emit to standard error (per map-reduce task). This prevents runaway scripts from filling logs partitions to capacity .	true	100000
hive_exec_compress_output	Controls whether the final outputs of a query (to a local/hdfs file or a hive table) is compressed. The compression codec and other options are determined from hadoop config variables mapred.output.compress.	true	false
hive_exec_compress_intermediate	Controls whether intermediate files produced by hive between multiple map-	true	false

Name	Description	Required	Default
	reduce jobs are compressed. The compression codec and other options are determined from hadoop config variables mapred.output.compress.		

Sqoop Barclamp

Sqoop is an SQL based command-line tool to assist with HDFS data import/export (SQL-to-Hadoop). Sqoop is a tool designed to transfer data between Hadoop and relational databases. You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.

Sqoop automates most of this process by relying on the database to describe the schema for the data to be imported. Sqoop uses MapReduce to import and export the data, which provides parallel operation as well as fault tolerance.

Table 4-48: Sqoop Barclamp Parameters

Name	Description	Required	Default
sqoop_connection_factories	A comma-delimited list of ManagerFactory implementations which are consulted, in order, to instantiate ConnManager instances used to drive connections to databases.	false	
sqoop_tool_plugins	A comma-delimited list of ToolPlugin implementations which are consulted, in order, to register SqoopTool instances which allow third-party tools to be used.	false	
sqoop_metastore_client_enable_autoconnect	If true, Sqoop will connect to a local metastore for job management when no other metastore arguments are provided.	true	false
sqoop_metastore_client_autoconnect_url	The connect string to use when connecting to a job-management metastore. If unspecified, uses ~/.sqoop/. You can specify a different path here.	false	
sqoop_metastore_client_autoconnect_username	The username to bind to the metastore.	false	
sqoop_metastore_client_autoconnect_password	The password to bind to the metastore.	false	
sqoop_metastore_client_record_password	If true, allow saved passwords in the metastore.	false	
sqoop_metastore_server_location	Path to the shared metastore database files. If this is not set, it will be placed in ~/.sqoop/.	false	
sqoop_metastore_server_port	Port that this metastore should listen on.	false	

Support

Cloudera Support

To obtain support for Hadoop:

- Open a request at Cloudera's support portal. http://www.cloudera.com/hadoop-support/

Printed in USA

www.dell.com| support.dell.com

Appendix A: Dell | Hadoop Solution Components

- **Hadoop:** http://en.wikipedia.org/wiki/Hadoop
- **Hadoop Distributed File System (HDFS):** A distributed file system that provides high-throughput access to application data (http://en.wikipedia.org/wiki/Hadoop_Distributed_Filesystem#Hadoop_Distributed_File_System).
- **MapReduce:** A software framework for distributed processing of large data sets on compute clusters (http://en.wikipedia.org/wiki/MapReduce).
- **HBase:** A scalable, distributed database that supports structured data storage for large tables.
- **Hive:** A data warehouse infrastructure that provides data summarization and ad-hoc querying.
- **ZooKeeper:** A high-performance coordination service for distributed applications.
- **Pig:** A platform for analyzing large data sets that consists of high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.
- **Sqoop:** A tool designed to import data from relational databases into Hadoop. Sqoop uses JDBC to connect to a database.
- **Oozie:** An open-source workflow engine and coordination service to manage data processing jobs within Hadoop.
- **Hue:** A browser based interface for interacting with Hadoop clusters.
- **Crowbar:** A Dell provided, supported, and maintained toolset for system deployment and configuration automation. Crowbar supports the bare-metal bring-up of new hardware and configuration management of existing hardware.

Appendix B: External References

- Cloudera: http://www.cloudera.com
- Nagios: http://www.nagios.org
- Ganglia: http://ganglia.sourceforge.net

To Learn More

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