**Cloudera Offline Installation (Path “B”) Plan**

Contents

[Overview 3](#_Toc476215178)

[Software Placement 3](#_Toc476215179)

[Required Software 3](#_Toc476215180)

[Steps 3](#_Toc476215181)

[Installation Instructions 3](#_Toc476215182)

[Overview 3](#_Toc476215183)

[Build the Virtual Servers for the Cluster 4](#_Toc476215184)

[Install Support Software and Configure VMs 6](#_Toc476215185)

[Cloudera Manager Installation 6](#_Toc476215186)

[Configure Cloudera Manager to Use Local Parcel Repository 6](#_Toc476215187)

# Overview

There are two pieces that comprise the Cloudera Hadoop Cluster (CDH) Build; the Cloudera Manager Framework and the Cloudera Hadoop Cluster System. You can use the Cloudera Manager to administer the CDH cluster, to include installing all the CDH Agents and services. Essentially, we’ll use YUM package manager and a local repo to install the Cloudera Manager Server, then use Parcels from within Cloudera Manager Server to install agents on cluster nodes.

# Software Placement

We will be using CDH 5.9.1.4 for this installation.

Since this will be an offline installation, we must create our own custom repositories for the Cloudera Manager package and the Cloudera Hadoop (CDH) parcels.

## Required Software

Cloudera Manager Server, Cloudera Manager Server DB

## Steps

1. Update Cobbler Repositories to Get Desired Version

**cobbler reposync --name=cloudera\_manager**

1. Grab the correct Parcel

On Agora Spacewalk Server:

**mkdir /var/www/cobbler/repo\_mirror/cloudera\_parcels**

**cd /var/www/cobbler/repo\_mirror/cloduera\_parcels**

**wget <http://archive.cloudera.com/cdh5/parcels/5.9.1.4/CDH-5.9.1-1.cdh5.9.1.p0.4-el6.parcel>**

**wget** [**http://archive.cloudera.com/cdh5/parcels/5.9.1.4/CDH-5.9.1-1.cdh5.9.1.p0.4-el6.parcel.sha1**](http://archive.cloudera.com/cdh5/parcels/5.9.1.4/CDH-5.9.1-1.cdh5.9.1.p0.4-el6.parcel.sha1)

1. Tar the directories and prepare for transfer to RADON

From within /var/www/cobbler/repo\_mirror do:

**tar -cvf /shared/p\_drive/cloudera.tar cloudera\_manager cloudera\_parcels**

Complete ATA Paperwork to transfer to RADON and untar (-xvf) in same location on RADON Spacewalk server.

# Hardware Specifications

## Selecting Hardware for Your CDH Cluster

The first step in choosing a machine configuration is to understand the type of hardware your operations team already manages. Operations teams often have opinions or hard requirements about new machine purchases, and will prefer to work with hardware with which they’re already familiar. Hadoop is not the only system that benefits from efficiencies of scale. Again, as a general suggestion, if the cluster is new or you can’t accurately predict your ultimate workload, we advise that you use balanced hardware.

There are four types of roles in a basic Hadoop cluster: NameNode (and Standby NameNode), JobTracker, TaskTracker, and DataNode. (A node is a machine performing a particular task.) Most machines in your cluster will perform two of these roles, functioning as both DataNode (for data storage) and TaskTracker (for data processing).

Here are the recommended specifications for DataNode/TaskTrackers in a balanced Hadoop cluster:

* 12-24 1-4TB hard disks in a JBOD (Just a Bunch Of Disks) configuration
* 2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz
* 64-512GB of RAM
* Bonded Gigabit Ethernet or 10Gigabit Ethernet (the more storage density, the higher the network throughput needed)

The NameNode role is responsible for coordinating data storage on the cluster, and the JobTracker for coordinating data processing. (The Standby NameNode should not be co-located on the NameNode machine for clusters and will run on hardware identical to that of the NameNode.) Cloudera recommends that customers purchase enterprise-class machines for running the NameNode and JobTracker, with redundant power and enterprise-grade disks in RAID 1 or 10 configurations.

The NameNode will also require RAM directly proportional to the number of data blocks in the cluster. A good rule of thumb is to assume 1GB of NameNode memory for every 1 million blocks stored in the distributed file system. With 100 DataNodes in a cluster, 64GB of RAM on the NameNode provides plenty of room to grow the cluster. We also recommend having HA configured on both the NameNode and JobTracker, features that have been available in the CDH4 line for some time.

Here are the recommended specifications for NameNode/JobTracker/Standby NameNode nodes. The drive count will fluctuate depending on the amount of redundancy:

* 4–6 1TB hard disks in a JBOD configuration (1 for the OS, 2 for the FS image [RAID 1], 1 for Apache ZooKeeper, and 1 for Journal node)
* 2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz
* 64-128GB of RAM
* Bonded Gigabit Ethernet or 10Gigabit Ethernet

|  |
| --- |
| **Remember, the Hadoop ecosystem is designed with a parallel environment in mind.** |

If you expect your Hadoop cluster to grow beyond 20 machines, we recommend that the initial cluster be configured as if it were to span two racks, where each rack has a top-of-rack 10 GigE switch. As the cluster grows to multiple racks, you will want to add redundant core switches to connect the top-of-rack switches with 40GigE. Having two logical racks gives the operations team a better understanding of the network requirements for intra-rack and cross-rack communication.

With a Hadoop cluster in place, the team can start identifying workloads and prepare to benchmark those workloads to identify hardware bottlenecks. After some time benchmarking and monitoring, the team will understand how additional machines should be configured. Heterogeneous Hadoop clusters are common, especially as they grow in size and number of use cases – so starting with a set of machines that are not “ideal” for your workload will not be a waste of time. Cloudera Manager offers templates that allow different hardware profiles to be managed in groups, making it simple to manage heterogeneous clusters.

Below is a list of various hardware configurations for different workloads, including our original “balanced” recommendation:

* Light Processing Configuration (1U/machine): Two hex-core CPUs, 24-64GB memory, and 8 disk drives (1TB or 2TB)
* Balanced Compute Configuration (1U/machine): Two hex-core CPUs, 48-128GB memory, and 12 – 16 disk drives (1TB or 2TB) directly attached using the motherboard controller. These are often available as twins with two motherboards and 24 drives in a single 2U cabinet.
* Storage Heavy Configuration (2U/machine): Two hex-core CPUs, 48-96GB memory, and 16-24 disk drives (2TB – 4TB). This configuration will cause high network traffic in case of multiple node/rack failures.
* Compute Intensive Configuration (2U/machine): Two hex-core CPUs, 64-512GB memory, and 4-8 disk drives (1TB or 2TB)

(Note that Cloudera expects to adopt 2×8, 2×10, and 2×12 core configurations as they arrive.)

The following diagram shows how a machine should be configured according to workload:

# Installation Instructions

## Overview

Because we are doing an offline installation, we are going to use Cloudera recommended “Path B” with Parcels. Below is an overview of the installation paths.

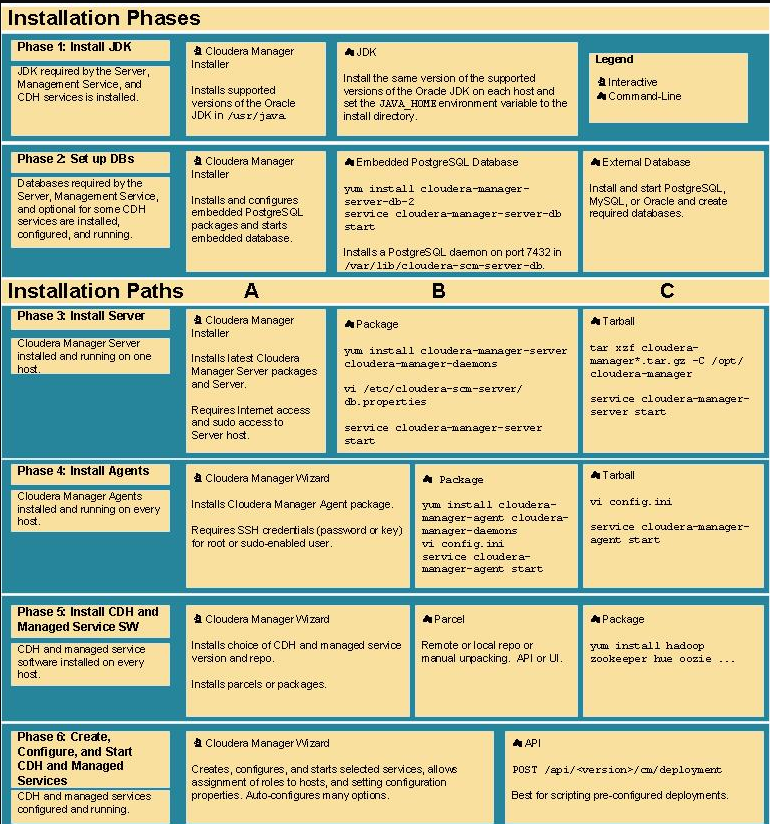


Figure 1- Overview of Install Process

## 

Figure 2 - General CDH Architecture

## Hadoop Cluster Active Ports

The following ports need to be allowed in/out of your local server FW on all nodes.

| **Component** | **Service** | **Port** | **Protocol** | **Configuration** | **Comment** |
| --- | --- | --- | --- | --- | --- |
| Hadoop HDFS | DataNode | 50010 | TCP | dfs.datanode.  address | DataNode HTTP server port |
|  | DataNode | 1004 | TCP | dfs.datanode.  address |  |
|  | DataNode | 50075 | TCP | dfs.datanode.http.  address |  |
|  | DataNode | 1006 | TCP | dfs.datanode.http.  address |  |
|  | DataNode | 50020 | TCP | dfs.datanode.ipc.  address |  |
|  | NameNode | 8020 | TCP | fs.default.  name  or  fs.defaultFS | fs.default.  name  is deprecated (but still works) |
|  | NameNode | 50070 | TCP | dfs.http.  address  or  dfs.namenode.  http-address | dfs.http.  address  is deprecated (but still works) |
|  | NameNode | 50470 | TCP | dfs.https.  address  or  dfs.namenode.  https-address | dfs.https.  address  is deprecated (but still works) |
|  | Secondary NameNode | 50090 | TCP | dfs.secondary.  http.address  or  dfs.namenode.  secondary.  http-address | dfs.secondary.  http.address  is deprecated (but still works) |
|  | Secondary NameNode | 50495 | TCP | dfs.secondary.  https.address |  |
|  | JournalNode | 8485 | TCP | dfs.namenode.  shared.edits.dir |  |
|  | JournalNode | 8480 | TCP |  |  |
| Hadoop MapReduce (MRv1) | JobTracker | 8021 | TCP | mapred.job.  tracker |  |
|  | JobTracker | 50030 | TCP | mapred.job.  tracker.  http.address |  |
|  | JobTracker | 9290 | TCP | jobtracker.  thrift.address | Required by Hue and Cloudera Manager Activity Monitor |
|  | TaskTracker | 50060 | TCP | mapred.task.  tracker.http.  address |  |
|  | TaskTracker | 0 | TCP | mapred.task.  tracker.report.  address | Communicating with child (umbilical) |
| Hadoop YARN (MRv2) | ResourceManager | 8032 | TCP | yarn.  resourcemanager.  address |  |
|  | ResourceManager | 8030 | TCP | yarn.  resourcemanager.  scheduler.address |  |
|  | ResourceManager | 8031 | TCP | yarn.  resourcemanager.  resource-tracker.  address |  |
|  | ResourceManager | 8033 | TCP | yarn.  resourcemanager.  admin.address |  |
|  | ResourceManager | 8088 | TCP | yarn.  resourcemanager.  webapp.address |  |
|  | NodeManager | 8040 | TCP | yarn.  nodemanager.  localizer.  address |  |
|  | NodeManager | 8042 | TCP | yarn.  nodemanager.  webapp.address |  |
|  | NodeManager | 8041 | TCP | yarn.  nodemanager.  address |  |
|  | MapReduce JobHistory Server | 10020 | TCP | mapreduce.  jobhistory.  address |  |
|  | Shuffle HTTP | 13562 | TCP |  |  |
|  | MapReduce JobHistory Server | 19888 | TCP | mapreduce.  jobhistory.  webapp.address |  |
| HBase | Master | 60000 | TCP | hbase.master.  port | IPC |
|  | Master | 60010 | TCP | hbase.master.  info.port | HTTP |
|  | RegionServer | 60020 | TCP | hbase.  regionserver.  port | IPC |
|  | RegionServer | 60030 | TCP | hbase.  regionserver.  info.port | HTTP |
|  | HQuorumPeer | 2181 | TCP | hbase.  zookeeper.  property.  clientPort | HBase-managed ZK mode |
|  | HQuorumPeer | 2888 | TCP | hbase.  zookeeper.  peerport | HBase-managed ZK mode |
|  | HQuorumPeer | 3888 | TCP | hbase.  zookeeper.  leaderport | HBase-managed ZK mode |
|  | REST | 8080 | TCP | hbase.rest.  port |  |
|  | REST UI | 8085 | TCP |  |  |
|  | ThriftServer | 9090 | TCP | Pass -p <port> on CLI |  |
|  | ThriftServer | 9095 | TCP |  |  |
|  |  | 9090 | TCP | Pass --port <port> on CLI |  |
| Hive | Metastore | 9083 | TCP |  |  |
|  | HiveServer2 | 10000 | TCP | hive.  server2.  thrift.port |  |
| Sqoop | Metastore | 16000 | TCP | sqoop.  metastore.  server.port |  |
| Sqoop 2 | Sqoop 2 server | 12000 | TCP |  |  |
|  | Sqoop 2 | 12001 | TCP |  | Admin port |
| ZooKeeper | Server (with CDH 5 and/or Cloudera Manager 5) | 2181 | TCP | clientPort | Client port |
|  | Server (with CDH 5 only) | 2888 | TCP | X in server.N  =host:X:Y | Peer |
|  | Server (with CDH 5 only) | 3888 | TCP | X in server.N  =host:X:Y | Peer |
|  | Server (with CDH 5 and Cloudera Manager 5) | 3181 | TCP | X in server.N  =host:X:Y | Peer |
|  | Server (with CDH 5 and Cloudera Manager 5) | 4181 | TCP | X in server.N  =host:X:Y | Peer |
|  | ZooKeeper FailoverController (ZKFC) | 8019 | TCP |  | Used for HA |
|  | ZooKeeper JMX port | 9010 | TCP |  | ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must *EITHER*   * Open up all ports when the connection originates from the Cloudera Manager server; *OR* * Do the following:   1. Open a non-ephemeral port (such as 9011) in the firewall.   2. Install Oracle Java 7u4 JDK or later.   3. Add the port configuration to the safety valve, for example:Dcom.sun.management.jmxremote.rmi.port=9011   4. Restart ZooKeeper. |
| Hue | Server | 8888 | TCP |  |  |
|  | Beeswax Server | 8002 |  |  |  |
|  | Beeswax Metastore | 8003 |  |  |  |
| Oozie | Oozie Server | 11000 | TCP | OOZIE\_HTTP\_  PORT  in  oozie-env.sh | HTTP |
|  | Oozie Server | 11001 | TCP | OOZIE\_ADMIN\_  PORT  in  oozie-env.sh | Shutdown port |
| Spark | Default Master RPC port | 7077 | TCP |  |  |
|  | Default Worker RPC port | 7078 | TCP |  |  |
|  | Default Master web UI port | 18080 | TCP |  |  |
|  | Default Worker web UI port | 18081 | TCP |  |  |
| HttpFS | HttpFS | 14000 | TCP |  |  |
|  | HttpFS | 14001 | TCP |  |  |

Table 1 - Hadoop Ports

## Build the Virtual Servers for the Cluster

1. Build out your host servers for the cluster (9 in our case). Should use Ansible Playbook for this.

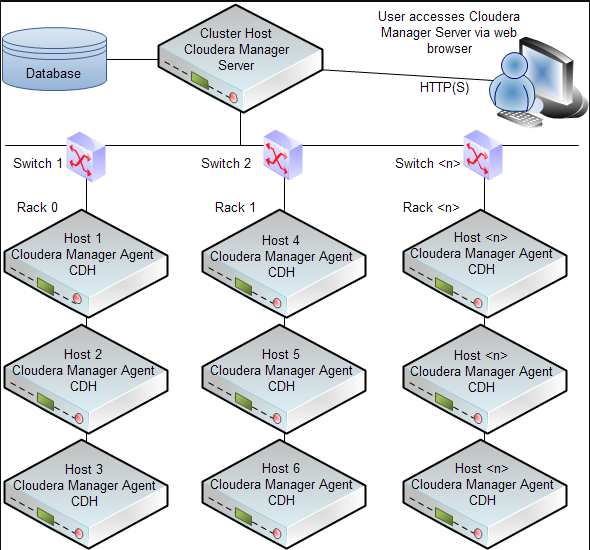


Figure 3 - Typical Cluster Configuration

* 1. Each node must have /etc/hosts updated for all other nodes
  2. DNS entries need to be in place for all nodes (ideally)
  3. Requirements for CM Server itself are:
     1. **Disk Space**
        1. **Cloudera Manager Server**
        2. 5 GB on the partition hosting /var.
        3. 500 MB on the partition hosting /usr.
        4. For parcels, the space required depends on the number of parcels you download to the Cloudera Manager Server and distribute to Agent hosts.:
           1. CDH 4.6 - ~700 MB per parcel, CDH 5 - ~1 GB per parcel
        5. **Cloudera Management Service** - The Host Monitor and Service Monitor databases are stored on the partition hosting /var. Ensure that you have at least 20 GB available on this partition. For further information
        6. **Agents** - On Agent hosts each unpacked parcel requires about three times the space of the downloaded parcel on the Cloudera Manager Server. By default unpacked parcels are located in /opt/cloudera/parcels.
     2. **RAM** - 4 GB is appropriate for most cases, and is required when using Oracle databases. 2 GB may be sufficient for non-Oracle deployments involving fewer than 100 hosts. However, if you want to run the Cloudera Manager Server on a machine with 2 GB of RAM, you must tune down its maximum heap size (by modifying -Xmx in /etc/default/cloudera-scm-server). Otherwise the kernel may kill the Server for consuming too much RAM.
     3. **Python** - Cloudera Manager uses Python. All supported operating systems contain a Python version 2.4 or higher. Cloudera Manager and CDH 4 require at least Python 2.4, but Hue in CDH 5 requires Python 2.6 or 2.7.
  4. Configure data directory
     1. Each host should be built on local storage (not shared).
     2. A /data directory should be built to host the HDFS.

## Install Support Software and Configure VMs

1. Install Latest JDK (make sure executable is in /usr/java…..)
2. Set up local repository on cloudera-manager server.
   1. /etc/yum.repos.d/cloudera-manager.repo

[cloudera\_manager]

name=cloudera\_manager

baseurl=http://spacewalk/cobbler/repo\_mirror/cloudera\_manager/

enabled=1

gpgcheck=0

1. Install Python >2.6 on all nodes if not installed.
2. Set up root ssh access to nodes from CM Server
   1. as root: **ssh-copy-id 192.168.1.2, .3, and so on**
3. Disable SELINUX
   1. vi /etc/sysconfig/selinux (set to disable)
4. Turn off firewall (at least during initial config) do for ip6tables as well.
   1. **service iptables save**
   2. service iptables stop
   3. chkconfig iptables off

## Cloudera Manager Installation

1. Install the Cloudera Manager Database on the CM Server
   1. **yum install cloudera-manager-server-db-2**
   2. **service cloudera-scm-server-db start**
   3. **chkconfig cloudera-scm-server-db on**
2. Install the Cloudera Manager Server Packages
   1. **yum install cloudera-manager-daemons cloudera-manager-server cloudera-manager-agent**
   2. **service cloudera-scm-server start**
   3. **service cloudera-scm-agent start** (optional)
   4. **chkconfig cloudera-scm-server on**

## Configure Cloudera Manager

1. Connect to Cloudera Manager Server on Port 7180
   1. <http://cloudera.local:7180>
   2. Default User: admin Pass: admin
2. Utilize Installation Wizard to Configure Server and set up configured host nodes.
   1. Add cluster nodes

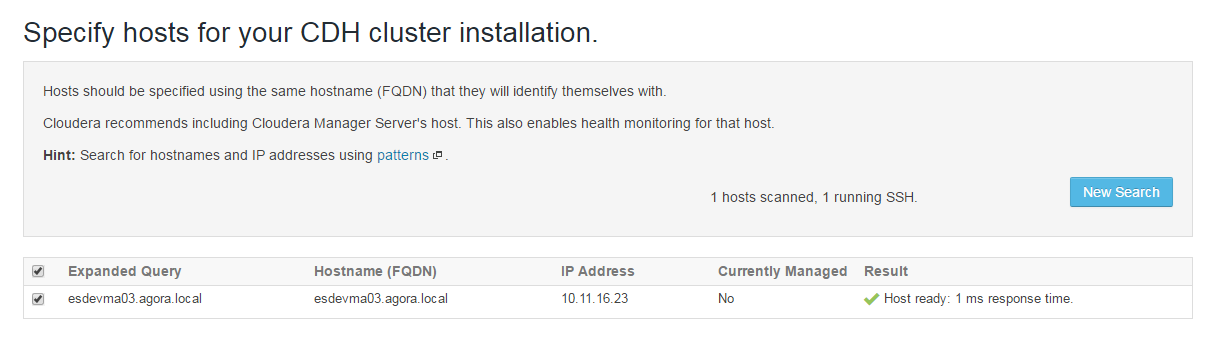


Figure 4 - Add Cluster Nodes

* 1. Select Repository (Add custom Repos for Parcels)

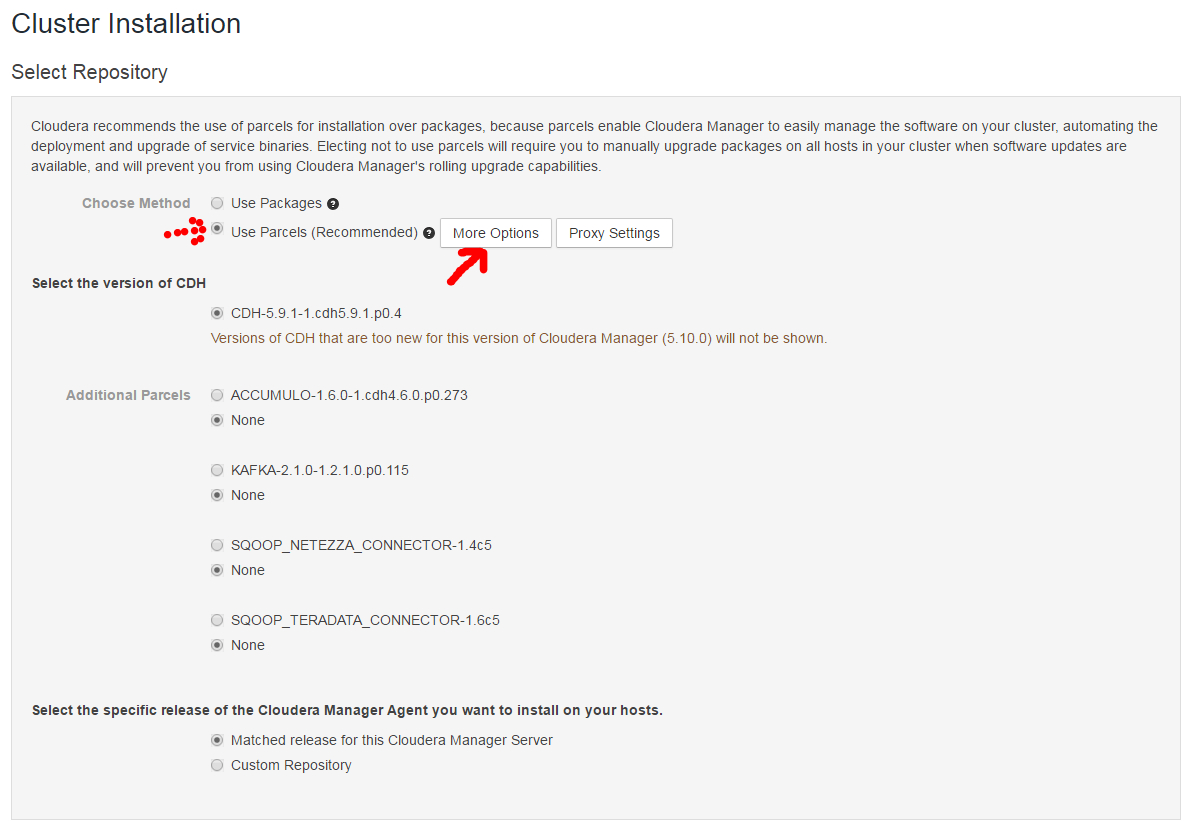


Figure 5 - Add Custom Parcel Repos

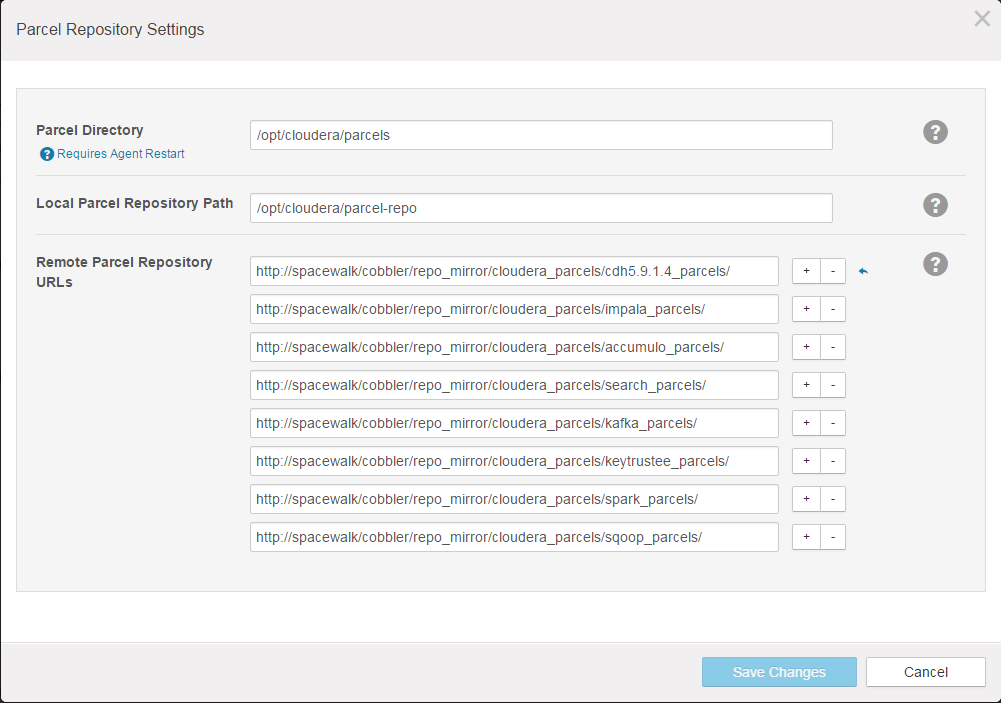


Figure 6 - Add Custom Parcel Repos (Cont'd)

* 1. Skip JDK Install (This has to be installed manually)

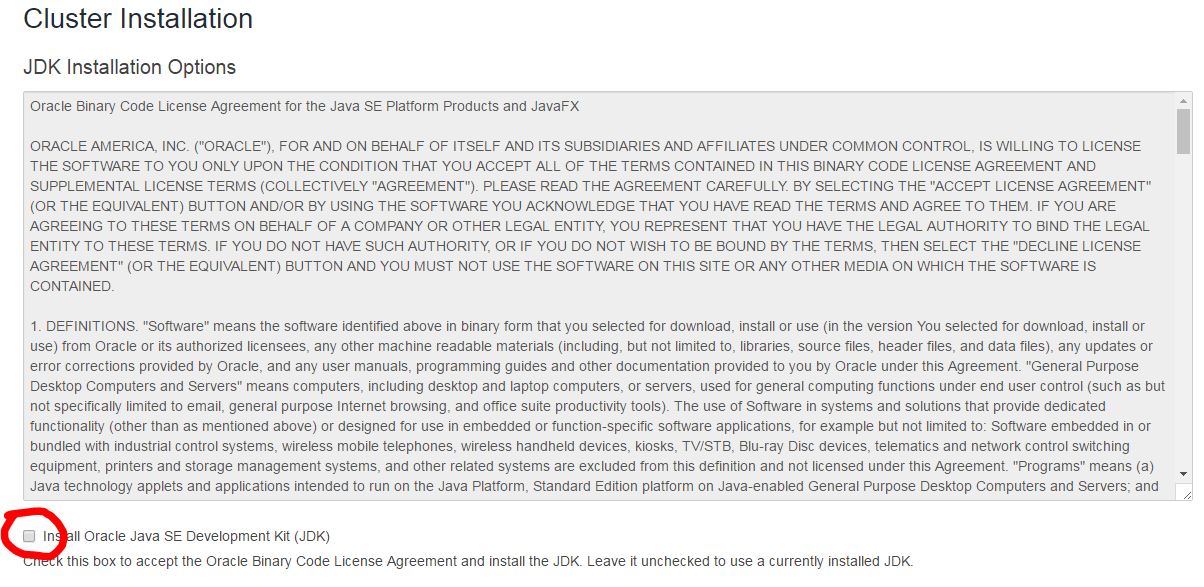


Figure 7 - Skip JDK Installation

* 1. DO NOT select “Single User Mode”
  2. Put in your SSH credentials for the nodes in cluster

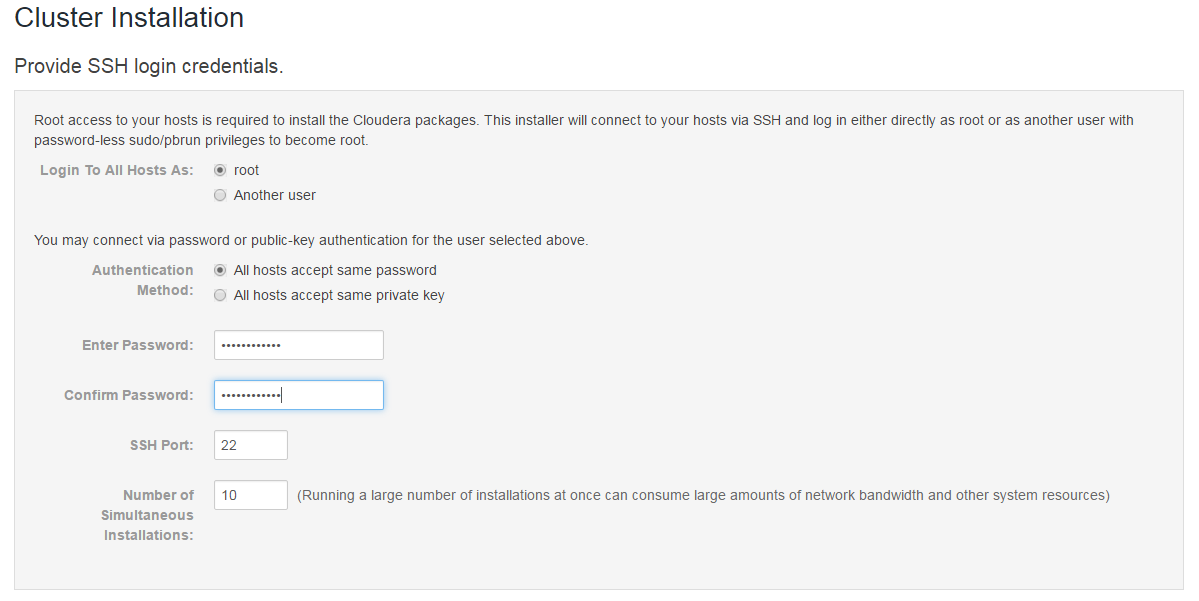


Figure 8 - SSH Credentials

* 1. You should see the progress moving at this point

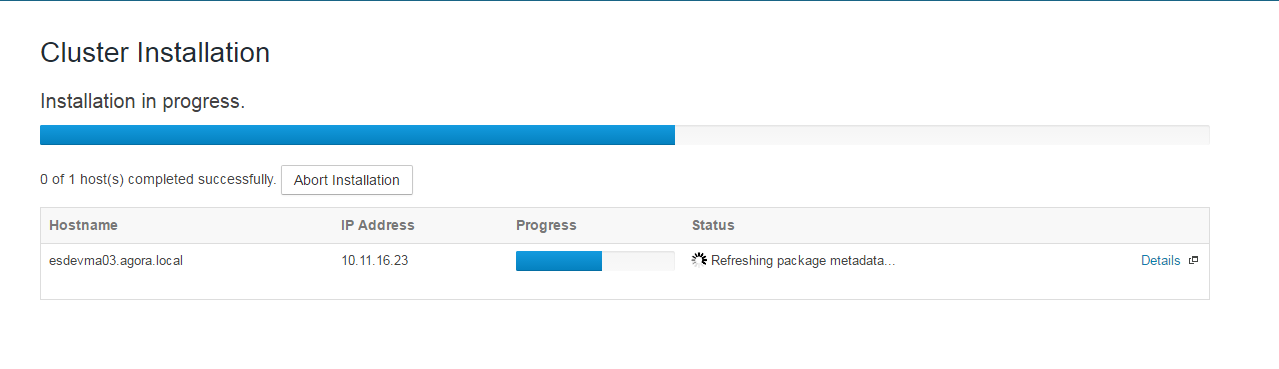


Figure 9 - Installation Started

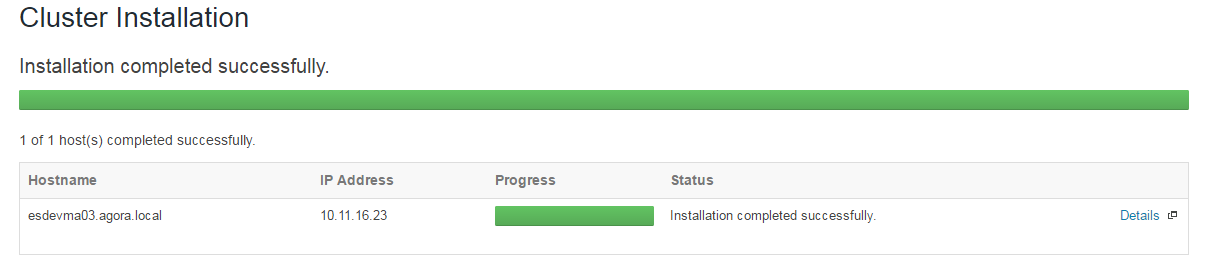


Figure 10 - Successful Installation

* 1. Begin Parcel installation

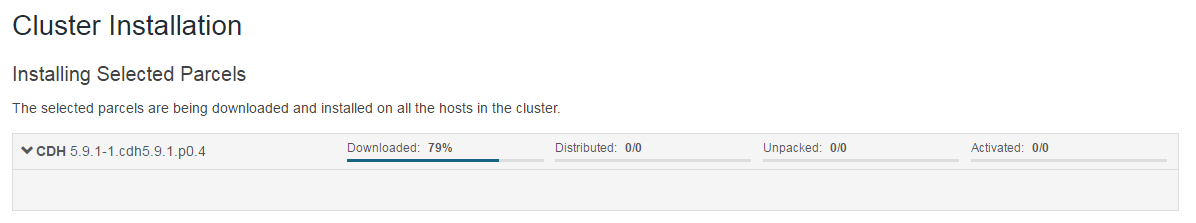


Figure 11 - Parcel Installation

1. After initial installation, if you have to add the local Parcel Repository. From within Cloudera Manager do:
   1. Click https://www.cloudera.com/documentation/manager/5-1-x/Images/parcels_icon.png in the top navigation bar
   2. Click the **Edit Settings** button.
   3. Select **Administration** > **Settings**.
   4. Click the **Parcels** category.
   5. Click the **Hosts** tab.
   6. Click the **Configuration** tab.
   7. Click the **Parcels** category.
   8. Click the **Edit Settings** button
   9. Add <http://spacewalk/cobbler/repo_mirror/cloudera_parcels>

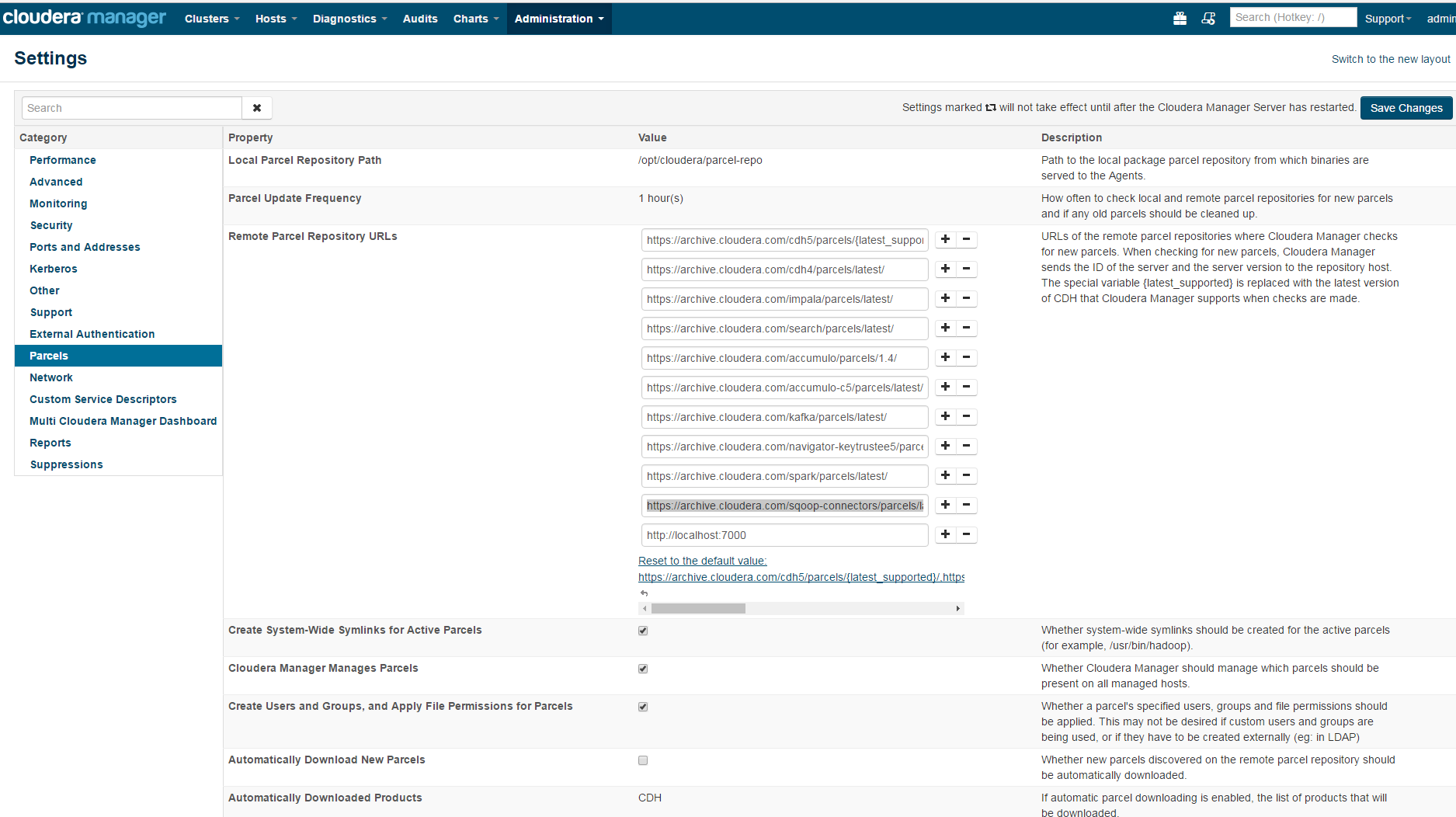
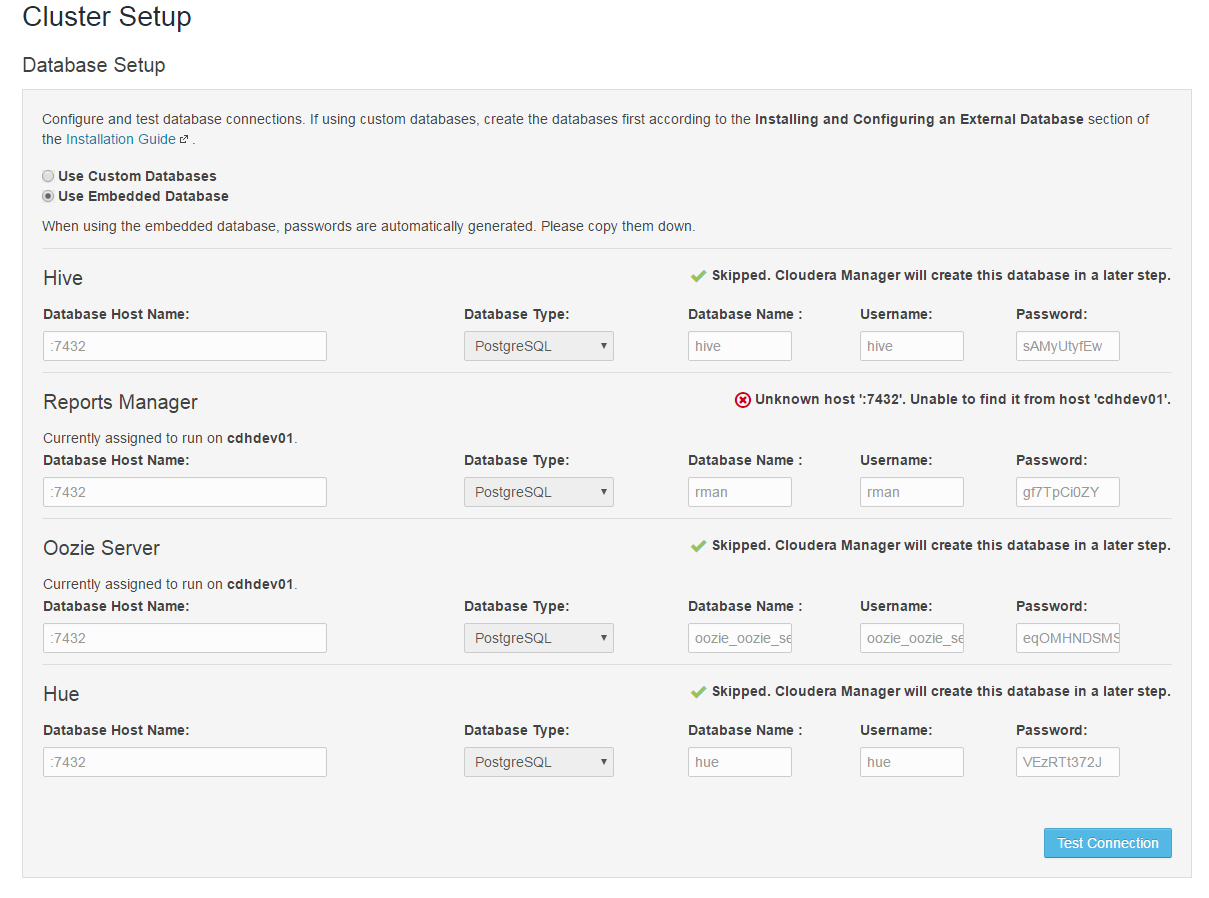
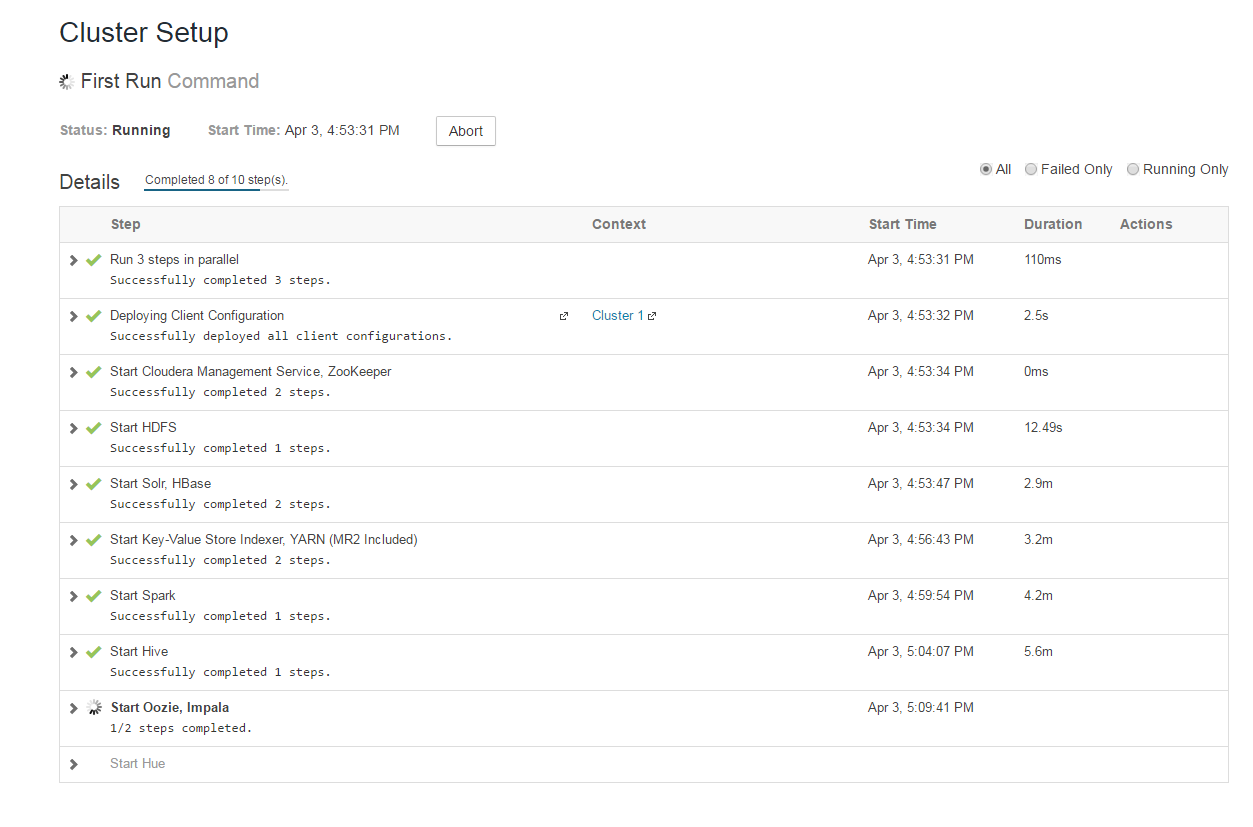


Figure 12 - Custom Parcel Repo Example

NOTE: If you get the following error about Reports database:



Toward the end of the installation, you should see



Do the following:

To fix it, look at the files in /etc/cloudera-scm-server/db.mgmt.properties and find all the places where you see :7432. You'll notice that there's no host name before the colon. Put the fully qualified domain name (NOT localhost) before the colon in each place where that occurs.  
  
Then restart CM server:  
service cloudera-scm-server restart