

Managing a Hadoop cluster with Cloudera Manager

Mark Stetzer
@stetzer

Why use CM?

- Point & click management
- Central location to configure all nodes & services

Capabilities Overview

- Hosts dashboard showing load avg, disk usage, etc.
- Bundled config downloads
- CDH3 & CDH4 support
- HDFS HA (CDH4)
- Lucid, Precise, & Squeeze support
- API
- Gateway role (a.k.a. client box)
- ...and much more!

What does it look like?

The screenshot displays the Cloudera Manager interface for a host named **master00.chacha.com**. The top navigation bar includes the Cloudera Manager logo, a search bar, and links to Support Portal, Help, and the user profile (admin). The main content area is divided into several sections:

- Hosts:** A green banner at the top indicates the host is in **Good Health**.
- Details:** A table showing host information.

IP	Rack	Health	Last Update	Number of Cores	Load	Physical Memory	Swap Space	Host Agent
10.0.0.200	/default	Good	0ms	2	2.86 2.62 1.91	762.5 MiB / 7.3 GiB	0 B / 0 B	Details
- File Systems:** A table showing disk usage for mounted file systems.

Disk	Mount Point	Usage
/dev/xvda1	/	3.1 GiB / 8.0 GiB
/dev/xvdb	/mnt	21.2 GiB / 413.4 GiB
/dev/xvdf	/data	20.9 GiB / 99.8 GiB
- Processes:** A table showing the status of various services.

Service	Instance	Name	Links	Status	PID	Uptime	Full log file	Stderr	Stdout
None	None	deploy-client-config		Stopped			Full log file	Full stderr log	Full stdout log
None	None	host-inspector		Stopped			Full log file	Full stderr log	Full stdout log
hdfs	namenode (master00)	hdfs-NAMENODE	NameNode Web UI	Running	1189	2.4h	Full log file	Full stderr log	Full stdout log
hdfs	secondarynamenode (master00)	hdfs-SECONDARYNAMENODE	SecondaryNameNode Web UI	Running	1209	2.4h	Full log file	Full stderr log	Full stdout log
mapreduce	jobtracker (master00)	mapreduce-JOBTRACKER	JobTracker Web UI fairscheduler	Running	1198	2.4h	Full log file	Full stderr log	Full stdout log

First, some AWS setup...

Setting up a VPC

Create an Amazon Virtual Private Cloud

Cancel

Select a VPC configuration below:

☐ **VPC with a Single Public Subnet Only**

Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

☒ **VPC with Public and Private Subnets**

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation.

☐ **VPC with Public and Private Subnets and Hardware VPN Access**

This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your datacenter - effectively extending your datacenter to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.

☐ **VPC with a Private Subnet Only and Hardware VPN Access**

Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate datacenter via an IPsec Virtual Private Network (VPN) tunnel.

The diagram illustrates a VPC configuration. At the top, a cloud icon labeled 'Internet' contains the text 'Amazon S3, EC2, SimpleDB, RDS'. A line connects this cloud to a box labeled 'VPC'. Inside the 'VPC' box, there are two subnets: a 'Public Subnet' and a 'Private Subnet'. The 'Public Subnet' is connected to the 'Internet' cloud. The 'Private Subnet' is connected to the 'Public Subnet' via a 'NAT' instance, represented by a small server icon.

Creates: a /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply)

Continue

Configuring private IPs

- Set up DHCP to give instances same internal IP after restart

Assigning elastic IPs

- Instances in private subnet use NAT to get to Internet
- Instances in public subnet need elastic IPs

Mount EBS volume(s)

- `mkfs.ext4 /dev/xvdf`
- `mkdir -m 000 /data`
- `echo "/dev/xvdf /data auto noatime 0 0" |
sudo tee -a /etc/fstab`
- `mount /data`

DNS

- Must configure DNS so hosts can address each other & reverse DNS works
- If instances can't identify themselves, Cloudera agents won't work correctly

Now take me through:

- Examining cluster health
- Adding a node to a cluster
- Configuring services on nodes
- Adding a new service

Questions?



Want to work with Hadoop at
your day job?

<http://about.chacha.com/about/careers/>