The Data Computing Division of EMC

EMC Greenplum® HD Enterprise Edition

Administrator Guide

P/N: 300-013-062

Rev: A01

EMC Greenplum HD EE

Welcome to Greenplum HD EE

Welcome to Greenplum HD EE! If you are not sure how to get started, here are a few places to find the information you are looking for:

- Quick Start - Small Cluster - Set up a Hadoop cluster with a small to moderate number of nodes
- Installation Guide - Learn how to set up a production cluster, large or small
- User Guide - Read more about what you can do with a Greenplum HD EE cluster

Welcome to Greenplum HD EE

Greenplum HD EE, a fully Apache Hadoop Interface-compatible, is the easiest, most dependable, and fastest Hadoop distribution on the planet. It is the only Hadoop distribution that allows direct data input and output via Direct Access NFS, and the first to provide true High Availability (HA) at all levels. Greenplum HD EE introduces logical volumes to Hadoop. A volume is a way to group data and apply policy across an entire data set. Greenplum HD EE provides hardware status and control with the Greenplum HD EE Control System, a comprehensive UI including a Heatmap that displays the health of the entire cluster at a glance. Read on to learn about how the unique features of Greenplum HD EE provide the highest-performance, lowest cost Hadoop available.

To get started right away, read the Quick Start guide:

- Quick Start - Small Cluster

To learn more about Greenplum HD EE, read on!

Ease of Use

With Greenplum HD EE, it is easy to run Hadoop jobs reliably, while isolating resources between different departments or jobs, applying data and performance policies, and tracking resource usage and job performance:

1. Create a volume and set policy. The Greenplum HD EE Control System makes it simple to set up a volume and assign granular control to users or groups. Use replication, mirroring, and snapshots for data protection, isolation, or performance.
2. Provision resources. You can limit the size of data on a volume, or place the volume on specific racks or nodes for performance or protection.
3. Run the Hadoop job normally. Proactive cluster monitoring lets you track resource usage and job performance, while Direct Access NFS gives you easy data input and direct access to the results.

Greenplum HD EE lets you control data access and placement, so that multiple concurrent Hadoop jobs can safely share the cluster.

With Greenplum HD EE, you can mount the cluster on any server or client and have your applications write data and log files directly into the cluster, instead of the batch processing model of the past. You do not have to wait for a file to be closed before reading it; you can tail a file as it is being written. Direct Access NFS even makes it possible to use standard shell scripts to work with Hadoop data directly.

Provisioning resources is simple. You can easily create a volume for a project or department in a few clicks. Greenplum HD EE integrates with NIS and LDAP, making it easy to manage users and groups. The Greenplum HD EE Control System makes it a breeze to assign user or group quotas, to limit how much data a user or group can write; or volume quotas, to limit the size of a volume. You can assign topology to a volume, to limit it to a specific rack or set of nodes. Setting recovery time objective (RTO) and recovery point objective (RPO) for a data set is a simple matter of scheduling snapshots and mirrors on a volume through the Greenplum HD EE Control System. You can set read and write permissions on volumes directly via NFS or using hadoop fs commands, and volumes provide administrative delegation through ACLs; for example, through the Greenplum HD EE Control System you can control who can mount, unmount, snapshot, or mirror a volume.

Greenplum HD EE is 100% Hadoop API compatible. You can run Hadoop jobs the way you always have. Greenplum HD EE is backwards and will be forwards-compatible across all versions of the Hadoop API, so you don't have to change your applications to use Greenplum HD EE.

For more information:

- Read about Provisioning Applications
- Learn about Direct Access NFS

Dependability

With clusters growing to thousands of nodes, hardware failures are inevitable even with the most reliable machines in place. Greenplum HD EE Distribution for Apache Hadoop has been designed from the ground up to tolerate hardware failure seamlessly.

Greenplum HD EE is the first Hadoop distibution to provide true HA and failover at all levels, including a Greenplum HD EE Distributed HA NameNode™. If a disk or node in the cluster fails, Greenplum HD EE automatically restarts any affected processes on another node without requiring administrative intervention. The HA JobTracker ensures that any tasks interrupted by a node or disk failure are re-started on another TaskTracker node. In the event of any failure, the job's completed task state is preserved and no tasks are lost. For additional data reliability, every bit of data on the wire is compressed and CRC-checked.

With volumes, you can control access to data, set replication factor, and place specific data sets on specific racks or nodes for performance or data protection. Volumes control data access to specific users or groups with Linux-style permissions that integrate with existing LDAP and NIS directories. Volumes can be size-limited with volume quotas to prevent data overruns from using excessive storage capacity. One of the most powerful aspects of the volume concept is the ways in which a volume provides data protection:

- To enable point-in-time recovery and easy backups, volumes have manual and policy-based snapshot capability.
- For true business continuity, you can manually or automatically mirror volumes and synchronize them between clusters or datacenters to enable easy disaster recovery.
- You can set volume read/write permission and delegate administrative functions, to control access to data.

Volumes can be exported with Direct Access NFS with HA, allowing data to be read and written directly to Hadoop without the need for temporary storage or log collection. You can load-balance across NFS nodes; clients connecting to different nodes see the same view of the cluster.

The Greenplum HD EE Control System provides powerful hardware insight down to the node level, as well as complete control of users, volumes, quotas, mirroring, and snapshots. Filterable alarms and notifications provide immediate warnings about hardware failures or other conditions that require attention, allowing a cluster administrator to detect and resolve problems quickly.

For more information:

- Take a look at the Heatmap
- Learn about Volumes, Snapshots, and Mirroring
- Explore Data Protection scenarios

Performance

Greenplum HD EE for Apache Hadoop achieves up to three times the performance of any other Hadoop distribution.

Greenplum HD EE Direct Shuffle uses the Distributed NameNode to improve Reduce phase performance drastically. Unlike Hadoop distributions that use the local filesystem for shuffle and HTTP to transport shuffle data, Greenplum HD EE shuffle data is readable directly from anywhere on the network. Greenplum HD EE stores data with Lockless Storage Services, a sharded system that eliminates contention and overhead from data transport and retrieval. Automatic, transparent client-side compression reduces network overhead and reduces footprint on disk, while direct block device I/O provides throughput at hardware speed with no additional overhead.

Greenplum HD EE gives you ways to tune the performance of your cluster. Using mirrors, you can load-balance reads on highly-accessed data to alleviate bottlenecks and improve read bandwidth to multiple users. You can run Direct Access NFS on many nodes – all nodes in the cluster, if desired – and load-balance reads and writes across the entire cluster. Volume topology helps you further tune performance by allowing you to place resource-intensive Hadoop jobs and high-activity data on the fastest machines in the cluster.

For more information:

- Read about Provisioning for Performance

Get Started

Now that you know a bit about how the features of Greenplum HD EE for Apache Hadoop work, take a quick tour to see for yourself how they can work for you:

- To explore cluster installation scenarios, see Planning the Deployment
- For more about provisioning, see Provisioning Applications
- For more about data policy, see Working with Data

Quick Start - Small Cluster

Choose the Quick Start guide that is right for your operating system:

- RHEL or CentOS

RHEL or CentOS

Use the following steps to install a simple Greenplum HD EE cluster up to 100 nodes with a basic set of services. To build a larger cluster, or to build a cluster that includes additional services (such as Hive or Pig), see the Installation Guide. To add services to nodes on a running cluster, see Adding Roles.

Setup

Follow these instructions to install a small Greenplum HD EE cluster (3-100 nodes) on machines that meet the following requirements:

- 64-bit RHEL 5.x or 6.0, or CentOS 5.x
- RAM: 4 GB or more
- At least one free unmounted drive or partition, 50 GB or more
- At least 10 GB of free space on the operating system partition
- Sun Java JDK 6 (not JRE)
- The `root` password, or `sudo` privileges
- A Linux user chosen to have administrative privileges on the cluster
    - Make sure the user has a password (using `sudo passwd <user>` for example)

Each node must have a unique hostname, and keyless SSH set up to all other nodes.

> ⚠ This procedure assumes you have free, unmounted physical partitions or hard disks for use by Greenplum HD EE. If you are not sure, please read Setting Up Disks for Greenplum HD EE.

- Create a text file `/tmp/disks.txt` listing disks and partitions for use by Greenplum HD EE. Each line lists a single disk, or partitions on a single disk. Example:

```
/dev/sdb
/dev/sdc1 /dev/sdc2 /dev/sdc4
/dev/sdd
```

Later, when you run `disksetup` to format the disks, specify the disks and partitions file. Example:

```
disksetup -F /tmp/disks.txt
```

For the steps that follow, make the following substitutions:

- `<user>` - the chosen administrative username
- `<node 1>`, `<node 2>`, `<node 3>`... - the IP addresses of nodes 1, 2, 3 ...
- `<proxy user>`, `<proxy password>`, `<host>`, `<port>` - proxy server credentials and settings

> ⚠ If you are installing a Greenplum HD EE cluster on nodes that are not connected to the Internet, contact Greenplum for assistance. If you are installing a cluster larger than 100 nodes, see the Installation Guide. In particular, CLDB nodes on large clusters should not run any other service (see Isolating CLDB Nodes).

Deployment

Refer to the Greenplum HD EE Release Notes for deployment information.

Next Steps

Using Hadoop

Now that Greenplum HD EE is installed, you can use Hadoop normally. Let's try a few simple Hadoop commands you probably already know: accessing data with the `hadoop fs` command, then running a simple example MapReduce job.

Example: Hadoop FileSystem Shell

Try a few Hadoop FileSystem commands:

1. List the contents of the root directory by typing `hadoop fs -ls /`
2. Create a directory called `foo` by typing `hadoop fs -mkdir /foo`
3. List the root directory again to verify that `/foo` is there: `hadoop fs -ls /`

Example:

```
# hadoop fs -ls
Found 3 items
drwxr-xr-x   - pconrad supergroup          0 2011-01-03 13:50 /foo
drwxr-xr-x   - pconrad supergroup          0 2011-01-04 13:57 /user
drwxr-xr-x   - mapred supergroup           0 2010-11-25 09:41 /var
```

Example: MapReduce

The following example performs a MapReduce job to estimate the value of Pi using 2 map tasks, each of which computes 50 samples:

```
# hadoop jar /opt/mapr/hadoop/hadoop-0.20.2/hadoop-0.20.2-dev-examples.jar pi 2 50
```

By the way, the directory you created in the previous example will be useful in the next step.

Mounting the Cluster via NFS

With Greenplum HD EE, you can export and mount the Hadoop cluster as a read/write volume via NFS from the machine where you installed Greenplum HD EE, or from a different machine.

- If you are mounting from the machine where you installed Greenplum HD EE, replace `<host>` in the steps below with `localhost`
- If you are mounting from a different machine, make sure the machine where you installed Greenplum HD EE is reachable over the network and replace `<host>` in the steps below with the hostname of the machine where you installed Greenplum HD EE.

Try the following steps to see how it works:

1. Change to the `root` user (or use `sudo` for the following commands).
2. See what is exported from the machine where you installed Greenplum HD EE:

   ```
   showmount -e <host>
   ```

3. Set up a mount point for the NFS share:

   ```
   mkdir /mapr
   ```

4. Mount the cluster via NFS:

   ```
   mount <host>:/mapr /mapr
   ```

5. Notice that the directory you created is there:

   ```
   # ls /mapr
   Found 3 items
   drwxr-xr-x   - pconrad supergroup          0 2011-01-03 13:50 /foo
   drwxr-xr-x   - pconrad supergroup          0 2011-01-04 13:57 /user
   drwxr-xr-x   - mapred supergroup           0 2010-11-25 09:41 /var
   ```

6. Try creating a directory via NFS:

   ```
   mkdir /mapr/foo/bar
   ```

7. List the contents of `/foo`:

```
hadoop fs -ls /foo
```

Notice that Hadoop can see the directory you just created with NFS.

> ⚠ If you are already running an NFS server, Greenplum HD EE will not run its own NFS gateway. In that case, you will not be able to mount the single-node cluster via NFS, but your previous NFS exports will remain available.

Installation Guide

Getting Started

To get started installing a basic cluster, take a look at the Quick Start guide:

- RHEL or CentOS

To design and configure a cluster from the ground up, perform the following steps:

1. PREPARE all nodes, making sure they meet the hardware, software, and configuration requirements.
2. PLAN which services to run on which nodes in the cluster.
3. INSTALL Greenplum HD EE Software:
    - On each node, INSTALL the planned Greenplum HD EE services.
    - On all nodes, RUN configure.sh.
    - On all nodes, FORMAT disks for use by Greenplum HD EE.
    - START the cluster.
    - SET UP node topology.
    - SET UP NFS for HA.
4. CONFIGURE the cluster:
    - SET UP the administrative user.
    - CHECK that the correct services are running.
    - SET UP authentication.
    - CONFIGURE cluster email settings.
    - CONFIGURE permissions.
    - SET user quotas.
    - CONFIGURE alarm notifications.

More Information

Once the cluster is up and running, you will find the following documents useful:

- Integration with Other Tools - guides to third-party tool integration with Greenplum HD EE
- Setting Up the Client - set up a laptop or desktop to work directly with a Greenplum HD EE cluster
- Uninstalling Greenplum HD EE - completely remove Greenplum HD EE software
- Cluster Upgrade - upgrade an entire cluster to the latest version of Greenplum HD EE software

Architecture

Greenplum HD EE is a complete Hadoop distribution, implemented as a number of services running on individual nodes in a cluster. In a typical cluster, all or nearly all nodes are dedicated to data processing and storage, and a smaller number of nodes run other services that provide cluster coordination and management. The following table shows the services corresponding to roles in a Greenplum HD EE cluster.

CLDB	Maintains the container location database (CLDB) and the Greenplum HD EE Distributed NameNode. The CLDB maintains the Greenplum HD EE FileServer storage (MapR-FS) and is aware of all the NFS and FileServer nodes in the cluster. The CLDB process coordinates data storage services among Greenplum HD EE FileServer nodes, Greenplum HD EE NFS Gateways, and Greenplum HD EE Clients.

FileServer	Runs the Greenplum HD EE FileServer (MapR-FS) and Greenplum HD EE Lockless Storage Services.
HBaseMaster	HBase master (optional). Manages the region servers that make up HBase table storage.
HRegionServer	HBase region server (used with HBase master). Provides storage for an individual HBase region.
JobTracker	Hadoop JobTracker. The JobTracker coordinates the execution of MapReduce jobs by assigning tasks to TaskTracker nodes and monitoring their execution.
NFS	Provides read-write Greenplum HD EE Direct Access NFS access to the cluster, with full support for concurrent read and write access. With NFS running on multiple nodes, Greenplum HD EE can use virtual IP addresses to provide automatic transparent failover, ensuring high availability (HA).
TaskTracker	Hadoop TaskTracker. The process that starts and tracks MapReduce tasks on a node. The TaskTracker registers with the JobTracker to receive task assignments, and manages the execution of tasks on a node.
WebServer	Runs the Greenplum HD EE Control System and provides the Greenplum HD EE Heatmap
Zookeeper	Enables high availability (HA) and fault tolerance for Greenplum HD EE clusters by providing coordination.

A process called the *warden* runs on all nodes to manage, monitor, and report on the other services on each node. The Greenplum HD EE cluster uses ZooKeeper to coordinate services. ZooKeeper runs on an odd number of nodes (at least three, and preferably five or more) and prevents service coordination conflicts by enforcing a rigid set of rules and conditions that determine which instance of each service is the master. The warden will not start any services unless ZooKeeper is reachable and more than half of the configured ZooKeeper nodes are live.

Hadoop Compatibility

Greenplum HD EE is compatible with the following version of the Apache Hadoop API:

- Apache Hadoop 0.20.2

For more information, see Hadoop Compatibility in This Release.

Requirements

Before setting up a Greenplum HD EE cluster, ensure that every node satisfies the following hardware and software requirements.

If you are setting up a large cluster, it is a good idea to use a configuration management tool such as Puppet or Chef, or a parallel ssh tool, to facilitate the installation of Greenplum HD EE packages across all the nodes in the cluster. The following sections provide details about the prerequisites for setting up the cluster.

Node Hardware

Minimum Requirements	Recommended
• 64-bit processor • 4G DRAM • 1 network interface • At least one free unmounted drive or partition, 100 GB or more • At least 10 GB of free space on the operating system partition • Twice as much swap space as RAM (if this is not possible, see Memory Overcommit)	• 64-bit processor with 8-12 cores • 32G DRAM or more • 2 GigE network interfaces • 3-12 disks of 1-3 TB each • At least 20 GB of free space on the operating system partition • 32 GB swap space or more (see also: Memory Overcommit)

In practice, it is useful to have 12 or more disks per node, not only for greater total storage but also to provide a larger number of storage pools available. If you anticipate a lot of big reduces, you will need additional network bandwidth in relation to disk I/O speeds. Greenplum HD EE can detect multiple NICs with multiple IP addresses on each node and load-balance throughput automatically. In general, the more network bandwidth you can provide, the faster jobs will run on the cluster. When designing a cluster for heavy CPU workloads, the processor on each node is more important than networking bandwidth and available disk space.

Disks

Set up at least three unmounted drives or partitions, separate from the operating system drives or partitions, for use by MapR-FS. For information on setting up disks for MapR-FS, see Setting Up Disks for Greenplum HD EE. If you do not have disks available for Greenplum HD EE, or to test with a small installation, you can use a flat file instead.

It is not necessary to set up RAID on disks used by MapR-FS. Greenplum HD EE uses a script called `disksetup` to set up storage pools. In most cases, you should let Greenplum HD EE calculate storage pools using the default stripe width of two or three disks. If you anticipate a high volume of random-access I/O, you can use the `-W` option with `disksetup` to specify larger storage pools of up to 8 disks each.

You can set up RAID on each node at installation time, to provide higher operating system performance (RAID 0), disk mirroring for failover (RAID 1), or both (RAID 10), for example. See the following instructions from the operating system websites:

- CentOS
- Red Hat

Software

Install a compatible 64-bit operating system on all nodes. Greenplum HD EE currently supports the following operating systems:

- CentOS 5.x
- Red Hat Enterprise Linux 5.x or 6.0

Each node must also have the following software installed:

- Sun Java JDK 6 (not JRE)

  If Java is already installed, check which versions of Java are installed: `java -version`
  If JDK 6 is installed, the output will include a version number starting with 1.6, and then below that the text `Java(TM)`.
  Example:

  ```
  java version "1.6.0_24"
  Java(TM) SE Runtime Environment (build 1.6.0_24-b07)
  ```

  Use update-alternatives to make sure JDK 6 is the default Java: `sudo update-alternatives --config java`

Configuration

Each node must be configured as follows:

- Unique hostname
- Keyless SSH set up between all nodes
- SELinux disabled
- Able to perform forward and reverse host name resolution with every other node in the cluster
- Administrative user - a Linux user chosen to have administrative privileges on the cluster
  - Make sure the user has a password (using `sudo passwd <user>` for example)

NTP

To keep all cluster nodes time-synchronized, Greenplum HD EE requires NTP to be configured and running on every node. If server clocks in the cluster drift out of sync, serious problems will occur with HBase and other Greenplum HD EE services. Greenplum HD EE raises a Time Skew alarm on any out-of-sync nodes. See http://www.ntp.org/ for more information about obtaining and installing NTP. In the event that a large adjustment must be made to the time on a particular node, you should stop ZooKeeper on the node, then adjust the time, then restart ZooKeeper.

DNS Resolution

For Greenplum HD EE to work properly, all nodes on the cluster must be able to communicate with each other. Each node must have a unique hostname, and must be able to resolve all other hosts with both normal and reverse DNS name lookup.

You can use the `hostname` command on each node to check the hostname. Example:

```
$ hostname -f
swarm
```

If the command returns a hostname, you can use the `getent` command to check whether the hostname exists in the hosts database. The `getent` command should return a valid IP address on the local network, associated with a fully-qualified domain name for the host. Example:

```
$ getent hosts `hostname`
10.250.1.53 swarm.corp.example.com
```

If you do not get the expected output from the `hostname` command or the `getent` command, correct the host and DNS settings on the node. A common problem is an incorrect loopback entry (`127.0.x.x`), which prevents the correct IP address from being assigned to the hostname.

Pay special attention to the format of `/etc/hosts`. For more information, see the hosts(5) man page. Example:

```
127.0.0.1 localhost
10.10.5.10 mapr-hadoopn.maprtech.prv mapr-hadoopn
```

Users and Groups

Greenplum HD EE uses each node's native operating system configuration to authenticate users and groups for access to the cluster. Any user or group you wish to grant access to the cluster must be present on all nodes and any client machines that will use the cluster. If you are deploying a large cluster, you should consider configuring all nodes to use LDAP or another user management system. You can use the Greenplum HD EE Control System to give specific permissions to particular users and groups. For more information, see Managing Permissions.

Choose a specific user to be the administrative user for the cluster. By default, Greenplum HD EE gives the user `root` full administrative permissions. If the nodes do not have an explicit `root` login (as is sometimes the case with Ubuntu, for example), you can give full permissions to the chosen administrative user after deployment. See Cluster Configuration.

On the node where you plan to run the `mapr-webserver` (the Greenplum HD EE Control System), install Pluggable Authentication Modules (PAM). See PAM Configuration.

Network Ports

The following table lists the network ports that must be open for use by Greenplum HD EE.

Service	Port
SSH	22
NFS	2049
MFS server	5660
ZooKeeper	5181
CLDB web port	7221
CLDB	7222
Web UI HTTP	8080 (set by user)
Web UI HTTPS	8443 (set by user)
JobTracker	9001
NFS monitor (for HA)	9997
NFS management	9998
JobTracker web	50030
TaskTracker web	50060
HBase Master	60000

LDAP	Set by user
SMTP	Set by user

The Greenplum HD EE UI runs on Apache. By default, installation does not close port 80 (even though the Greenplum HD EE Control System is available over HTTPS on port 8443). If this would present a security risk to your datacenter, you should close port 80 manually on any nodes running the Greenplum HD EE Control System.

Licensing

You can obtain and install a license through the License Manager after installation.

> If installing a new cluster, make sure to install the latest version of Greenplum HD EE software. If applying a new license to an existing Greenplum HD EE cluster, make sure to upgrade to the latest version of Greenplum HD EE first. If you are not sure, check the contents of the file `MapRBuildVersion` in the `/opt/mapr` directory . If the version is `1.0.0` and includes `GA` then you must upgrade before applying a license. Example:
>
> ```
> \# cat  /opt/mapr/MapRBuildVersion
> 1.0.0.10178GA-0v
> ```
>
> For information about upgrading the cluster, see MapR:Cluster Upgrade.

PAM Configuration

Greenplum HD EE uses Pluggable Authentication Modules (PAM) for user authentication in the Greenplum HD EE Control System. Make sure PAM is installed and configured on the node running the `mapr-webserver`.

There are typically several PAM modules (profiles), configurable via configuration files in the `/etc/pam.d/` directory. Each standard UNIX program normally installs its own profile. Greenplum HD EE can use (but does not require) its own `mapr-admin` PAM profile. The Greenplum HD EE Control System webserver tries the following three profiles in order:

1. mapr-admin (Expects that user has created the `/etc/pam.d/mapr-admin` profile)
2. sudo (`/etc/pam.d/sudo`)
3. sshd (`/etc/pam.d/sshd`)

The profile configuration file (for example, `/etc/pam.d/sudo`) should contain an entry corresponding to the authentication scheme used by your system. For example, if you are using local OS authentication, check for the following entry:

```
auth sufficient pam_unix.so # For local OS Auth
```

The following sections provide information about configuring PAM to work with LDAP or Kerberos.

> The file `/etc/pam.d/sudo` should be modified only with care and only when absolutely necessary.

LDAP

To configure PAM with LDAP:

1. Install the appropriate PAM packages:
   - On Redhat/Centos, `sudo yum install pam_ldap`
2. Open `/etc/pam.d/sudo` and check for the following line:

```
auth sufficient pam_ldap.so # For LDAP Auth
```

Kerberos

To configure PAM with Kerberos:

1. Install the appropriate PAM packages:

- On Redhat/Centos, `sudo yum install pam_krb5`

2. Open `/etc/pam.d/sudo` and check for the following line:

```
auth sufficient pam_krb5.so # For kerberos Auth
```

Setting Up Disks for Greenplum HD EE

In a production environment, or when testing performance, Greenplum HD EE should be configured to use physical hard drives and partitions. In some cases, it is necessary to reinstall the operating system on a node so that the physical hard drives are available for direct use by Greenplum HD EE. Reinstalling the operating system provides an unrestricted opportunity to configure the hard drives. If the installation procedure assigns hard drives to be managed by the Linux Logical Volume manger (LVM) by default, you should explicitly remove from LVM configuration the drives you plan to use with Greenplum HD EE. It is common to let LVM manage one physical drive containing the operating system partition(s) and to leave the rest unmanaged by LVM for use with Greenplum HD EE.

To determine if a disk or partition is ready for use by Greenplum HD EE:

1. Run the command `sudo lsof <partition>` to determine whether any processes are already using the partition.
2. There should be no output when running `sudo fuser <partition>`, indicating there is no process accessing the specific partition.
3. The partition should not be mounted, as checked via the output of the `mount` command.
4. The partition should be accessible to standard Linux tools such as `mkfs`. You should be able to successfully format the partition using a command like `sudo mkfs.ext3 <partition>` as this is similar to the operations Greenplum HD EE performs during installation. If `mkfs` fails to access and format the partition, then it is highly likely Greenplum HD EE will encounter the same problem.

Any disk or partition that passes the above testing procedure can be added to the list of disks and partitions passed to the `disksetup` command.

To specify disks or partitions for use by Greenplum HD EE:

- Create a text file `/tmp/disks.txt` listing disks and partitions for use by Greenplum HD EE. Each line lists a single disk, or partitions on a single disk. Example:

```
/dev/sdb
/dev/sdc1 /dev/sdc2 /dev/sdc4
/dev/sdd
```

Later, when you run `disksetup` to format the disks, specify the disks and partitions file. Example:

```
disksetup -F /tmp/disks.txt
```

⚠ You should run `disksetup` only after running `configure.sh`.

To test without formatting physical disks:

If you do not have physical partitions or disks available for reformatting, you can test Greenplum HD EE by creating a flat file and including a path to the file in the disk list file. You should create at least a 4GB file or larger.

The following example creates a 20 GB flat file (`bs=1G` specifies 1 gigabyte, multiply by `count=20`):

```
$ dd if=/dev/zero of=/root/storagefile bs=1G count=20
```

Using the above example, you would add the following to `/tmp/disks.txt`:

```
/root/storagefile
```

Working with a Logical Volume Manager

The Logical Volume Manager creates symbolic links to each logical volume's block device, from a directory path in the form: `/dev/<volume group>/<volume name>`. Greenplum HD EE needs the actual block location, which you can find by using the `ls`

`-l` command to list the symbolic links.

1. Make sure you have free, unmounted logical volumes for use by Greenplum HD EE:
    - Unmount any mounted logical volumes that can be erased and used for Greenplum HD EE.
    - Allocate any free space in an existing logical volume group to new logical volumes.
2. Make a note of the volume group and volume name of each logical volume.
3. Use `ls -l` with the volume group and volume name to determine the path of each logical volume's block device. Each logical volume is a symbolic link to a logical block device from a directory path that uses the volume group and volume name: `/dev/<volume group>/<volume name>`
   The following example shows output that represents a volume group named `mapr` containing logical volumes named `mapr1`, `mapr2`, `mapr3`, and `mapr4`:

    ```
    # ls -l /dev/mapr/mapr*
    lrwxrwxrwx 1 root root 22 Apr 12 21:48 /dev/mapr/mapr1 -> /dev/mapper/mapr-mapr1
    lrwxrwxrwx 1 root root 22 Apr 12 21:48 /dev/mapr/mapr2 -> /dev/mapper/mapr-mapr2
    lrwxrwxrwx 1 root root 22 Apr 12 21:48 /dev/mapr/mapr3 -> /dev/mapper/mapr-mapr3
    lrwxrwxrwx 1 root root 22 Apr 12 21:48 /dev/mapr/mapr4 -> /dev/mapper/mapr-mapr4
    ```

4. Create a text file `/tmp/disks.txt` containing the paths to the block devices for the logical volumes (one path on each line). Example:

    ```
    $ cat  /tmp/disks.txt
    /dev/mapper/mapr-mapr1
    /dev/mapper/mapr-mapr2
    /dev/mapper/mapr-mapr3
    /dev/mapper/mapr-mapr4
    ```

5. Pass `disks.txt` to `disksetup`

    ```
    # sudo /opt/mapr/server/disksetup -F -D /tmp/disks.txt
    ```

Planning the Deployment

Planning a Greenplum HD EE deployment involves determining which services to run in the cluster and where to run them. The majority of nodes are *worker* nodes, which run the TaskTracker and MapR-FS services for data processing. A few nodes run *control* services that manage the cluster and coordinate MapReduce jobs.

The following table provides general guidelines for the number of instances of each service to run in a cluster:

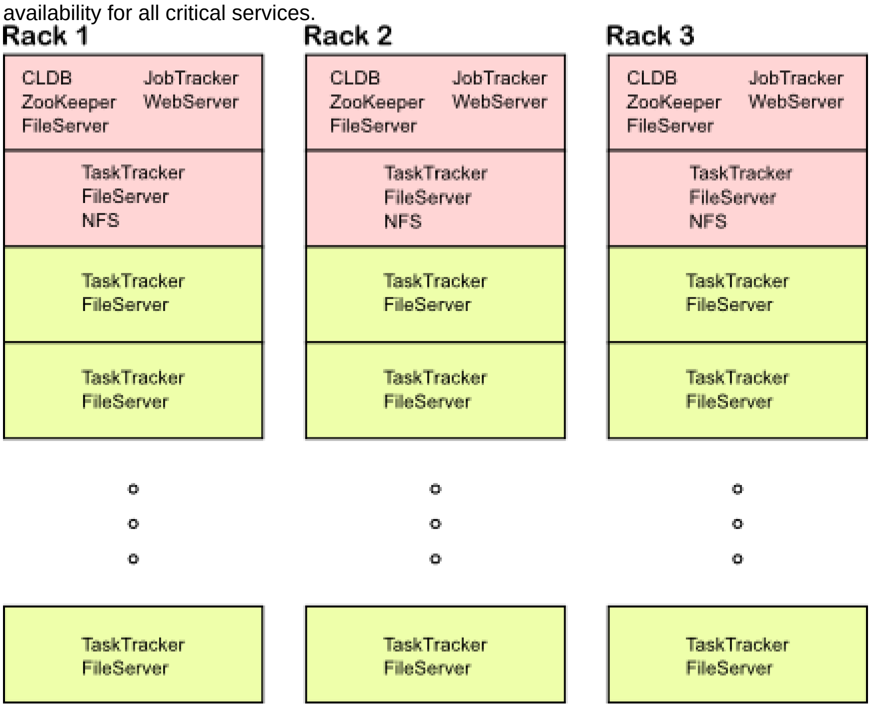
Service	Package	How Many
CLDB	mapr-cldb	1-3
FileServer	mapr-fileserver	Most or all nodes
HBase Master	mapr-hbase-master	1-3
HBase RegionServer	mapr-hbase-regionserver	Varies
JobTracker	mapr-jobtracker	1-3
NFS	mapr-nfs	Varies
TaskTracker	mapr-tasktracker	Most or all nodes
WebServer	mapr-webserver	One or more
Zookeeper	mapr-zookeeper	1, 3, 5, or a higher odd number

Sample Configurations

The following sections describe a few typical ways to deploy a Greenplum HD EE cluster.

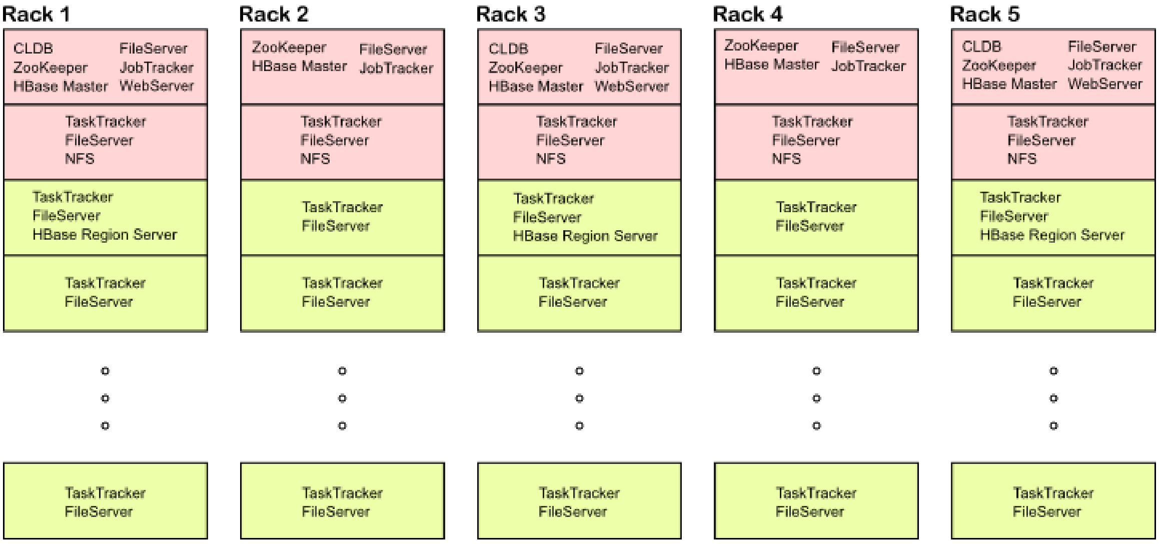
Small Cluster

A small cluster runs control services on three nodes and data services on the remaining nodes, providing failover and high availability for all critical services.



Larger Cluster

A large cluster (over 100 nodes) should isolate CLDB nodes from the TaskTracker and NFS nodes.



⚠ In large clusters, you should not run TaskTracker and ZooKeeper together on any nodes.

Planning NFS

The `mapr-nfs` service lets you access data on a licensed Greenplum HD EE cluster via the NFS protocol.

At cluster installation time, plan which nodes should provide NFS access according to your anticipated traffic. You can set up virtual IP addresses (VIPs) for NFS nodes in a Greenplum HD EE cluster, for load balancing or failover. VIPs provide multiple addresses that can be leveraged for round-robin DNS, allowing client connections to be distributed among a pool of NFS nodes. VIPs also make high availability (HA) NFS possible; in the event an NFS node fails, data requests are satisfied by other NFS nodes in the pool.

How you set up NFS depends on your network configuration and bandwidth, anticipated data access, and other factors. You can provide network access from MapR clients to any NFS nodes directly or through a gateway to allow access to data. Here are a few examples of how to configure NFS:

- On a few nodes in the cluster, with VIPs using DNS round-robin to balance connections between nodes (use at least as many VIPs as NFS nodes)
- On all file server nodes, so each node can NFS-mount itself and native applications can run as tasks
- On one or more dedicated gateways (using round-robin DNS or behind a hardware load balancer) to allow controlled access

Here are a few tips:

- Set up NFS on at least three nodes if possible.
- All NFS nodes must be accessible over the network from the machines where you want to mount them.
- To serve a large number of clients, set up dedicated NFS nodes and load-balance between them. If the cluster is behind a firewall, you can provide access through the firewall via a load balancer instead of direct access to each NFS node. You can run NFS on all nodes in the cluster, if needed.
- To provide maximum bandwidth to a specific client, install the NFS service directly on the client machine. The NFS gateway on the client manages how data is sent in or read back from the cluster, using all its network interfaces (that are on the same subnet as the cluster nodes) to transfer data via Greenplum HD EE APIs, balancing operations among nodes as needed.
- Use VIPs to provide High Availability (HA) and failover. See Setting Up NFS HA for more information.

NFS Memory Settings

The memory allocated to each Greenplum HD EE service is specified in the `/opt/mapr/conf/warden.conf` file, which Greenplum HD EE automatically configures based on the physical memory available on the node. You can adjust the minimum and maximum memory used for NFS, as well as the percentage of the heap that it tries to use, by setting the `percent`, `max`, and `min` parameters in the `warden.conf` file on each NFS node. Example:

```
...
service.command.nfs.heapsize.percent=3
service.command.nfs.heapsize.max=1000
service.command.nfs.heapsize.min=64
...
```

The percentages need not add up to 100; in fact, you can use less than the full heap by setting the `heapsize.percent` parameters for all services to add up to less than 100% of the heap size. In general, you should not need to adjust the memory settings for individual services, unless you see specific memory-related problems occurring.

Planning Services for HA

When properly licensed and configured for HA, the Greenplum HD EE cluster provides automatic failover for continuity throughout the stack. Configuring a cluster for HA involves running redundant instances of specific services, and configuring NFS properly. In HA clusters, it is advisable to have 3 nodes run CLDB and 5 run ZooKeeper. In addition, 3 Hadoop JobTrackers and/or 3 HBase Masters are appropriate depending on the purpose of the cluster. Any node or nodes in the cluster can run the Greenplum HD EE WebServer. In HA clusters, it is appropriate to run more than one instance of the WebServer with a load balancer to provide failover. NFS can be configured for HA using virtual IP addresses (VIPs). For more information, see High Availability NFS.

The following are the minimum numbers of each service required for HA:

- CLDB - 2 instances
- ZooKeeper - 3 instances (to maintain a quorum in case one instance fails)
- HBase Master - 2 instances
- JobTracker - 2 instances
- NFS - 2 instances

You should run redundant instances of important services on separate racks whenever possible, to provide failover if a rack goes down. For example, the top server in each of three racks might be a CLDB node, the next might run ZooKeeper and other control services, and the remainder of the servers might be data processing nodes. If necessary, use a worksheet to plan the services to run on each node in each rack.

Tips:

- If you are installing a large cluster (100 nodes or more), CLDB nodes should not run any other service and should not contain any cluster data (see Isolating CLDB Nodes).
- In HA clusters, it is advisable to have 3 nodes run CLDB and 5 run ZooKeeper. In addition, 3 Hadoop JobTrackers and/or 3 HBase Masters are appropriate depending on the purpose of the cluster.

Installing Greenplum HD EE

⚠ Before performing these steps, make sure all nodes meet the Requirements, and that you have planned which services to run on each node. You will need a list of the hostnames or IP addresses of all CLDB nodes, and the hostnames or IP addresses of all ZooKeeper nodes.

Perform the following steps, starting the installation with the *control nodes* running CLDB and ZooKeeper:

1. On each node, INSTALL the planned Greenplum HD EE services.
2. On all nodes, RUN configure.sh.
3. On all nodes, FORMAT disks for use by Greenplum HD EE.
4. START the cluster.
5. SET UP node topology.
6. SET UP NFS for HA.

The following sections provide details about each step.

Installing Greenplum HD EE Services

The Greenplum package installer will configure each node in the cluster to have one of three specific roles: Master, ZooKeeper, Worker. The installer includes all necessary rpm components and is designed to be run directly on each node in the cluster.

To install Greenplum HD EE:

1. Download the following binary from the EMC FeedbackCentral Beta Home Page: `emc-gphd-ee-1.x.x.x.bin`
2. As root, run the script on each node in your cluster.

   On node 1 run the script with the --master_node | -m option to install the master node RPMS.
   On nodes 2 and 3 run the script with the --zookeeper option | -z option to install the zooker node RPMS.
   On all other nodes, run the script with the --worker| -w option to install the worker node RPMS.option.

Optionally add additional components by running the script with the `--additional_pkgs` option. The available additional components are: Hbase, hive, pig, client

Running configure.sh

Run the script `configure.sh` to create `/opt/mapr/conf/mapr-clusters.conf` and update the corresponding `*.conf` and `*.xml` files. Before performing this step, make sure you have a list of the hostnames of the CLDB and ZooKeeper nodes. Optionally, you can specify the ports for the CLDB and ZooKeeper nodes as well. If you do not specify them, the default ports are:

- CLDB – 7222
- ZooKeeper – 5181

The script `configure.sh` takes an optional cluster name and log file, and comma-separated lists of CLDB and ZooKeeper host names or IP addresses (and optionally ports), using the following syntax:

```
/opt/mapr/server/configure.sh -C <host>[:<port>][,<host>[:<port>]...] -Z
<host>[:<port>][,<host>[:<port>]...] [-L <logfile>][-N <cluster name>]
```

Example:

```
/opt/mapr/server/configure.sh -C r1n1.sj.us:7222,r3n1.sj.us:7222,r5n1.sj.us:7222 -Z
r1n1.sj.us:5181,r2n1.sj.us:5181,r3n1.sj.us:5181,r4n1.sj.us:5181,r1n1.sj.us5:5181 -N
MyCluster
```

If you have not chosen a cluster name, you can run `configure.sh` again later to rename the cluster.

Formatting the Disks

On all nodes, use the following procedure to format disks and partitions for use by Greenplum HD EE.

⚠ This procedure assumes you have free, unmounted physical partitions or hard disks for use by Greenplum HD EE. If you are not sure, please read Setting Up Disks for Greenplum HD EE.

- Create a text file `/tmp/disks.txt` listing disks and partitions for use by Greenplum HD EE. Each line lists a single disk, or partitions on a single disk. Example:

```
/dev/sdb
/dev/sdc1 /dev/sdc2 /dev/sdc4
/dev/sdd
```

Later, when you run `disksetup` to format the disks, specify the disks and partitions file. Example:

```
disksetup -F /tmp/disks.txt
```

> The script `disksetup` removes all data from the specified disks. Make sure you specify the disks correctly, and that any data you wish to keep has been backed up elsewhere. Before following this procedure, make sure you have backed up any data you wish to keep.

1. Change to the `root` user (or use `sudo` for the following command).
2. Run `disksetup`, specifying the disk list file.
   Example:

   ```
   /opt/mapr/server/disksetup -F /tmp/disks.txt
   ```

Bringing Up the Cluster

In order to configure the administrative user and license, bring up the CLDB, Greenplum HD EE Control System, and ZooKeeper; once that is done, bring up the other nodes. You will need the following information:

- A list of nodes on which `mapr-cldb` is installed
- <MCS node> - the node on which the `mapr-webserver` service is installed
- <user> - the chosen Linux (or LDAP) user which will have administrative privileges on the cluster

To Bring Up the Cluster

1. Start ZooKeeper on all nodes where it is installed, by issuing the following command:

   ```
   /etc/init.d/mapr-zookeeper start
   ```

2. On one of the CLDB nodes and the node running the `mapr-webserver` service, start the warden:

   ```
   /etc/init.d/mapr-warden start
   ```

3. On the running CLDB node, issue the following command to give full permission to the chosen administrative user:

   ```
   /opt/mapr/bin/maprcli acl edit -type cluster -user <user>:fc
   ```

4. On a machine that is connected to the cluster and to the Internet, perform the following steps to install the license:
   - In a browser, view the Greenplum HD EE Control System by navigating to the node that is running the Greenplum HD EE Control System:
     https://<MCS node>:8443
     Your computer won't have an HTTPS certificate yet, so the browser will warn you that the connection is not trustworthy. You can ignore the warning this time.
   - On a machine that is connected to the cluster, open a browser and view the Greenplum HD EE Control System by navigating to the node that is running the WebServer: `https://<node 1>:8443`. Your computer won't have an HTTPS certificate yet, so the browser will warn you that the connection is not trustworthy. You can ignore the warning this time.
   - Log in to the Greenplum HD EE Control System as the administrative user you designated earlier. In the navigation pane, expand the System Settings group and click MapR Licenses to display the License Management dialog.
   - Send the Cluster ID number, along with your company name and the number of nodes in your cluster, to EMC FeedbackCentral (see “Licensing” on page 4).
   - Once you receive the license number back from EMC Greenplum, enter it in the text box, then click Activate.
5. Execute the following command on the running CLDB node (node 1)

```
/opt/mapr/bin/maprcli node services -nodes <node 1> -nfs start
```

6. On all other nodes, execute the following command:

```
/etc/init.d/mapr-warden start
```

7. Log in to the Greenplum HD EE Control System.
8. Under the Cluster group in the left pane, click **Dashboard**.
9. Check the Services pane and make sure each service is running the correct number of instances.

- Instances of the FileServer, NFS, and TaskTracker on all nodes
- 3 instances of the CLDB, JobTracker, and ZooKeeper
- 1 instance of the WebServer

Setting up Topology

Topology tells Greenplum HD EE about the locations of nodes and racks in the cluster. Topology is important, because it determines where Greenplum HD EE places replicated copies of data. If you define the cluster topology properly, Greenplum HD EE scatters replication on separate racks so that your data remains available in the event an entire rack fails. Cluster topology is defined by specifying a topology path for each node in the cluster. The paths group nodes by rack or switch, depending on how the physical cluster is arranged and how you want Greenplum HD EE to place replicated data.

Topology paths can be as simple or complex as needed to correspond to your cluster layout. In a simple cluster, each topology path might consist of the rack only (e. g. `/rack-1`). In a deployment consisting of multiple large datacenters, each topology path can be much longer (e. g. `/europe/uk/london/datacenter2/room4/row22/rack5/`). Greenplum HD EE uses topology paths to spread out replicated copies of data, placing each copy on a separate path. By setting each path to correspond to a physical rack, you can ensure that replicated data is distributed across racks to improve fault tolerance.

After you have defined node topology for the nodes in your cluster, you can use volume topology to place volumes on specific racks, nodes, or groups of nodes. See Setting Volume Topology.

Setting Node Topology

You can specify a topology path for one or more nodes using the node topo command, or in the Greenplum HD EE Control System using the following procedure.

To set node topology using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **Cluster** group and click the Nodes view.
2. Select the checkbox beside each node whose topology you wish to set.
3. Click the **Change Topology** button to display the Change Node Topology dialog.
4. Set the path in the **New Path** field:
    - To define a new path, type a topology path. Topology paths must begin with a forward slash ('/').
    - To use a path you have already defined, select it from the dropdown.
5. Click **Move Node** to set the new topology.

Setting Up NFS HA

You can easily set up a pool of NFS nodes with HA and failover using virtual IP addresses (VIPs); if one node fails the VIP will be automatically reassigned to the next NFS node in the pool. If you do not specify a list of NFS nodes, then Greenplum HD EE uses any available node running the Greenplum HD EE NFS service. You can add a server to the pool simply by starting the Greenplum HD EE NFS service on it. Before following this procedure, make sure you are running NFS on the servers to which you plan to assign VIPs. You should install NFS on at least three nodes. If all NFS nodes are connected to only one subnet, then adding another NFS server to the pool is as simple as starting NFS on that server; the Greenplum HD EE cluster automatically detects it and adds it to the pool.

You can restrict VIP assignment to specific NFS nodes or MAC addresses by adding them to the NFS pool list manually. VIPs are not assigned to any nodes that are not on the list, regardless of whether they are running NFS. If the cluster's NFS nodes have multiple network interface cards (NICs) connected to different subnets, you should restrict VIP assignment to the NICs that are on the correct subnet: for each NFS server, choose whichever MAC address is on the subnet from which the cluster will be NFS-mounted, then add it to the list. If you add a VIP that is not accessible on the subnet, then failover will not work. You can only set up VIPs for failover between network interfaces that are in the same subnet. In large clusters with multiple subnets, you can set up multiple groups of VIPs to provide NFS failover for the different subnets.

You can set up VIPs with the virtualip add command, or using the Add Virtual IPs dialog in the Greenplum HD EE Control System. The Add Virtual IPs dialog lets you specify a range of virtual IP addresses and assign them to the pool of servers that are running the NFS service. The available servers are displayed in the left pane in the lower half of the dialog. Servers that have been added to the NFS VIP pool are displayed in the right pane in the lower half of the dialog.

To set up VIPs for NFS using the Greenplum Control System:

1. In the Navigation pane, expand the **NFS HA** group and click the **NFS Setup** view.
2. Click **Start NFS** to start the NFS Gateway service on nodes where it is installed.
3. Click **Add VIP** to display the Add Virtual IPs dialog.
4. Enter the start of the VIP range in the **Starting IP** field.
5. Enter the end of the VIP range in the **Ending IP** field. If you are assigning one one VIP, you can leave the field blank.
6. Enter the Netmask for the VIP range in the **Netmask** field. Example: `255.255.255.0`
7. If you wish to restrict VIP assignment to specific servers or MAC addresses:
    a. If each NFS node has one NIC, or if all NICs are on the same subnet, select NFS servers in the left pane.
    b. If each NFS node has multiple NICs connected to different subnets, select the server rows with the correct MAC addresses in the left pane.
8. Click **Add** to add the selected servers or MAC addresses to the list of servers to which the VIPs will be assigned. The servers appear in the right pane.
9. Click **OK** to assign the VIPs and exit.

Cluster Configuration

After installing Greenplum HD EE Services and bringing up the cluster, perform the following configuration steps.

Setting Up the Administrative User

Give the administrative user full control over the cluster:

1. Log on to any cluster node as `root` (or use `sudo` for the following command).
2. Execute the following command, replacing `<user>` with the administrative username:
   `sudo /opt/mapr/bin/maprcli acl edit -type cluster -user <user>:fc`

For general information about users and groups in the cluster, see Users and Groups.

Checking the Services

Use the following steps to start the Greenplum HD EE Control System and check that all configured services are running:

1. Start the Greenplum HD EE Control System: in a browser, go to the following URL, replacing `<host>` with the hostname of the node that is running the WebServer: `https://<host>:8443`
2. Log in using the administrative username and password.
3. The first time you run the Greenplum HD EE Control System, you must accept the Terms of Service. Click **I Accept** to proceed.
4. Under the Cluster group in the left pane, click Dashboard.
5. Check the Services pane and make sure each service is running the correct number of instances. For example: if you have configured 5 servers to run the CLDB service, you should see that 5 of 5 instances are running.

If one or more services have not started, wait a few minutes to see if the warden rectifies the problem. If not, you can try to start the services manually. See Managing Services.
If too few instances of a service have been configured, check that the service is installed on all appropriate nodes. If not, you can add the service to any nodes where it is missing. See Reconfiguring a Node.

Configuring Authentication

If you use Kerberos, LDAP, or another authentication scheme, make sure PAM is configured correctly to give Greenplum HD EE access. See PAM Configuration.

Configuring Email

Greenplum HD EE can notify users by email when certain conditions occur. There are three ways to specify the email addresses of Greenplum HD EE users:

- From an LDAP directory
- By domain
- Manually, for each user

To configure email from an LDAP directory:

1. In the Greenplum HD EE Control System, expand the System Settings group and click Email Addresses to display the **Configure Email Addresses** dialog.
2. Select **Use LDAP** and enter the information about the LDAP directory into the appropriate fields.
3. Click **Save** to save the settings.

To configure email by domain:

1. In the Greenplum HD EE Control System, expand the System Settings group and click Email Addresses to display the **Configure Email Addresses** dialog.
2. Select **Use Company Domain** and enter the domain name in the text field.
3. Click **Save** to save the settings.

To configure email manually for each user:

1. Create a volume for the user.
2. In the Greenplum HD EE Control System, expand the MapR-FS group and click User Disk Usage.
3. Click the **username** to display the User Properties dialog.
4. Enter the user's email address in the **Email** field.
5. Click **Save** to save the settings.

Configuring SMTP

Use the following procedure to configure the cluster to use your SMTP server to send mail:

1. In the Greenplum HD EE Control System, expand the System Settings group and click SMTP to display the **Configure Sending Email** dialog.
2. Enter the information about how Greenplum HD EE will send mail:
   - **Provider:** assists in filling out the fields if you use Gmail.
   - **SMTP Server:** the SMTP server to use for sending mail.
   - **This server requires an encrypted connection (SSL):** specifies an SSL connection to SMTP.
   - **SMTP Port:** the SMTP port to use for sending mail.
   - **Full Name:** the name Greenplum HD EE should use when sending email. Example: `Greenplum Cluster`
   - **Email Address:** the email address Greenplum HD EE should use when sending email.
   - **Username:** the username Greenplum HD EE should use when logging on to the SMTP server.
   - **SMTP Password:** the password Greenplum HD EE should use when logging on to the SMTP server.
3. Click **Test SMTP Connection**.
4. If there is a problem, check the fields to make sure the SMTP information is correct.
5. Once the SMTP connection is successful, click **Save** to save the settings.

Configuring Permissions

By default, users are able to log on to the Greenplum HD EE Control System, but do not have permission to perform any actions. You can grant specific permissions to individual users and groups. See Managing Permissions.

Setting Quotas

Set default disk usage quotas. If needed, you can set specific quotas for individual users and groups. See Managing Quotas.

Configuring alarm notifications

If an alarm is raised on the cluster, Greenplum HD EE sends an email notification by default to the user associated with the object on which the alarm was raised. For example, if a volume goes over its allotted quota, Greenplum HD EE raises an alarm and sends email to the volume creator. You can configure Greenplum HD EE to send email to a custom email address in addition or instead of the default email address, or not to send email at all, for each alarm type. See Notifications.

Integration with Other Tools

- Compiling Pipes Programs
- Ganglia
- HBase Best Practices
- Mahout
- Nagios Integration

Compiling Pipes Programs

To facilitate running `hadoop pipes` jobs on various platforms, Greenplum HD EE provides `hadoop pipes`, `util`, and `pipes-example` sources.

> When using `pipes`, all nodes must run the same distribution of the operating system. If you run different distributions (Red Hat and CentOS, for example) on nodes in the same cluster, the compiled application might run on some nodes but not others.

To compile the pipes example:

1. Install `libssl` on all nodes.
2. Change to the `/opt/mapr/hadoop/hadoop-0.20.2/src/c++/utils` directory, and execute the following commands:

```
chmod +x configure
./configure # resolve any errors
make install
```

3. Change to the `/opt/mapr/hadoop/hadoop-0.20.2/src/c++/pipes` directory, and execute the following commands:

```
chmod +x configure
./configure # resolve any errors
make install
```

4. The APIs and libraries will be in the `/opt/mapr/hadoop/hadoop-0.20.2/src/c++/install` directory.
5. Compile `pipes-example`:

```
cd /opt/mapr/hadoop/hadoop-0.20.2/src/c++
g++ pipes-example/impl/wordcount-simple.cc -Iinstall/include/ -Linstall/lib/
-lhadooputils -lhadooppipes -lssl -lpthread -o wc-simple
```

To run the pipes example:

1. Copy the pipes program into MapR-FS.
2. Run the `hadoop pipes` command:

```
hadoop pipes -Dhadoop.pipes.java.recordreader=true -Dhadoop.pipes.java.recordwriter
=true -input <input-dir> -output <output-dir> -program <MapR-FS path to program>
```

Ganglia

Ganglia is a scalable distributed system monitoring tool that allows remote viewing live or historical statistics for a cluster. The Ganglia system consists of the following components:

- A PHP-based web front end
- Ganglia monitoring daemon (`gmond`): a multi-threaded monitoring daemon
- Ganglia meta daemon (`gmetad`): a multi-threaded aggregation daemon
- A few small utility programs

The daemon `gmetad` aggregates metrics from the `gmond` instances, storing them in a database. The front end pulls metrics from the database and graphs them. You can aggregate data from multiple clusters by setting up a separate `gmetad` for each, and then a master `gmetad` to aggregate data from the others. If you configure Ganglia to monitor multiple clusters, remember to use a separate port for each cluster.

Greenplum HD EE with Ganglia

The CLDB reports metrics about its own load, as well as cluster-wide metrics such as CPU and memory utilization, the number of active FileServer nodes, the number of volumes created, etc. For a complete list of metrics, see below.

MapRGangliaContext collects and sends CLDB metrics, FileServer metrics, and cluster-wide metrics to Gmon or Gmeta, depending on the configuration. On the Ganglia front end, these metrics are displayed separately for each FileServer by hostname. The ganglia monitor only needs to be installed on CLDB nodes to collect all the metrics required for monitoring a Greenplum HD EE cluster. To monitor other services such as HBase and MapReduce, install Gmon on nodes running the services and configure them as you normally would.

The Ganglia properties for the `cldb` and `fileserver` contexts are configured in the file `$INSTALL_DIR/conf/hadoop-metrics.properties`. Any changes to this file require a CLDB restart.

Installing Ganglia

To install Ganglia on Red Hat:

1. Download the following RPM packages for Ganglia version 3.1 or later:
   - ganglia-gmond
   - ganglia-gmetad
   - ganglia-web
2. On each CLDB node, install `ganglia-monitor`: `rpm -ivh <ganglia-gmond>`
3. On the machine where you plan to run the Ganglia meta daemon, install `gmetad`: `rpm -ivh <gmetad>`
4. On the machine where you plan to run the Ganglia front end, install `ganglia-webfrontend`: `rpm -ivh <ganglia-web>`

For more details about Ganglia configuration and installation, see the Ganglia documentation.

To start sending CLDB metrics to Ganglia:

1. Make sure the CLDB is configured to send metrics to Ganglia (see Service Metrics).
2. As `root` (or using `sudo`), run the following commands:

```
maprcli config save -values '{"cldb.ganglia.cldb.metrics":"1"}'
maprcli config save -values '{"cldb.ganglia.fileserver.metrics":"1"}'
```

To stop sending CLDB metrics to Ganglia:

- As `root` (or using `sudo`), run the following commands:

```
maprcli config save -values '{"cldb.ganglia.cldb.metrics":"0"}'
maprcli config save -values '{"cldb.ganglia.fileserver.metrics":"0"}'
```

Metrics Collected

CLDB	FileServers
Number of FileServers Number of Volumes Number of Containers Cluster Disk Space Used GB Cluster Disk Space Available GB Cluster Disk Capacity GB Cluster Memory Capacity MB Cluster Memory Used MB Cluster Cpu Busy % Cluster Cpu Total Number of FS Container Failure Reports Number of Client Container Failure Reports Number of FS RW Container Reports Number of Active Container Reports Number of FS Volume Reports Number of FS Register Number of container lookups Number of container assign Number of container corrupt reports Number of rpc failed Number of rpc received	FS Disk Used GB FS Disk Available GB Cpu Busy % Memory Total MB Memory Used MB Memory Free MB Network Bytes Received Network Bytes Sent

HBase Best Practices

* The HBase write-ahead log (WAL) writes many tiny records, and compressing it would cause massive CPU load. Before using

HBase, turn off compression for directories in the HBase volume (normally mounted at `/hbase`. Example:

```
hadoop mfs \-setcompression off /hbase
```

* You can check whether compression is turned off in a directory or mounted volume by using `[hadoop mfs]` to list the file contents. Example:

```
hadoop mfs \-ls /hbase
```

The letter `Z` in the output indicates compression is turned on; the letter `U` indicates compression is turned off. See `hadoop mfs` for more information.
* On any node where you plan to run both HBase and MapReduce, give more memory to the FileServer than to the RegionServer so that the node can handle high throughput. For example, on a node with 24 GB of physical memory, it might be desirable to limit the RegionServer to 4 GB, give 10 GB to MapR-FS, and give the remainder to TaskTracker. To change the memory allocated to each service, edit the `/opt/mapr/conf/warden.conf` file. See Tuning MapReduce for more information.

Mahout

Mahout is an Apache TLP project to create scalable, machine learning algorithms. For information about installing Mahout, see the Mahout Wiki.

To use Mahout with Greenplum HD EE, set the following environment variables:

- `HADOOP_HOME` - tells Mahout where to find the Hadoop directory (`/opt/mapr/hadoop/hadoop-0.20.2`)
- `HADOOP_CONF_DIR` - tells Mahout where to find information about the JobTracker (`/opt/mapr/hadoop/hadoop-0.20.2/conf`)

You can set the environment variables permanently by adding the following lines to /etc/environment:

```
HADOOP_HOME=/opt/mapr/hadoop/hadoop-0.20.2
HADOOP_CONF_DIR=/opt/mapr/hadoop/hadoop-0.20.2/conf
```

Nagios Integration

Nagios is an open-source cluster monitoring tool. Greenplum HD EE can generate a Nagios Object Definition File that describes the nodes in the cluster and the services running on each. You can generate the file using the Greenplum HD EE Control System or the nagios generate command, then save the file in the proper location in your Nagios environment.

To generate a Nagios file using the Greenplum HD EE Control System:

1. In the Navigation pane, click **Nagios**.
2. Copy and paste the output, and save as the appropriate Object Definition File in your Nagios environment.

For more information, see the Nagios documentation.

Setting Up the Client

Greenplum HD EE provides several interfaces for working with a cluster from a client computer:

- Greenplum HD EE Control System - manage the cluster, including nodes, volumes, users, and alarms
- Direct Access NFS - mount the cluster in a local directory
- Greenplum HD EE client - work with Greenplum HD EE Hadoop directly

Greenplum HD EE Control System

The Greenplum HD EE Control System is web-based, and works with the following browsers:

- Chrome
- Safari
- Firefox 3.0 and above
- Internet Explorer 7 and 8

To use the Greenplum HD EE Control System, navigate to the host that is running the WebServer in the cluster. Greenplum HD EE Control System access to the cluster is typically via HTTP on port 8080 or via HTTPS on port 8443; you can specify the protocol and port in the Configure HTTP dialog. You should disable pop-up blockers in your browser to allow Greenplum HD EE to open help links in new browser tabs.

Direct Access NFS

You can mount a Greenplum HD EE cluster locally as a directory on a Mac or Linux computer.

Before you begin, make sure you know the hostname and directory of the NFS share you plan to mount.
Example:

- `usa-node01:/mapr` - for mounting from the command line
- `nfs://usa-node01/mapr` - for mounting from the Mac Finder

Make sure the client machine has the appropriate username and password to access the NFS share. For best results, the username and password for accessing the Greenplum HD EE cluster should be the same username and password used to log into the client machine.

Linux

1. Make sure the NFS client is installed.
2. List the NFS shares exported on the server. Example:
   `showmount -e usa-node01`
3. Set up a mount point for an NFS share. Example:
   `sudo mkdir /mapr`
4. Mount the cluster via NFS. Example:
   `sudo mount usa-node01:/mapr /mapr`

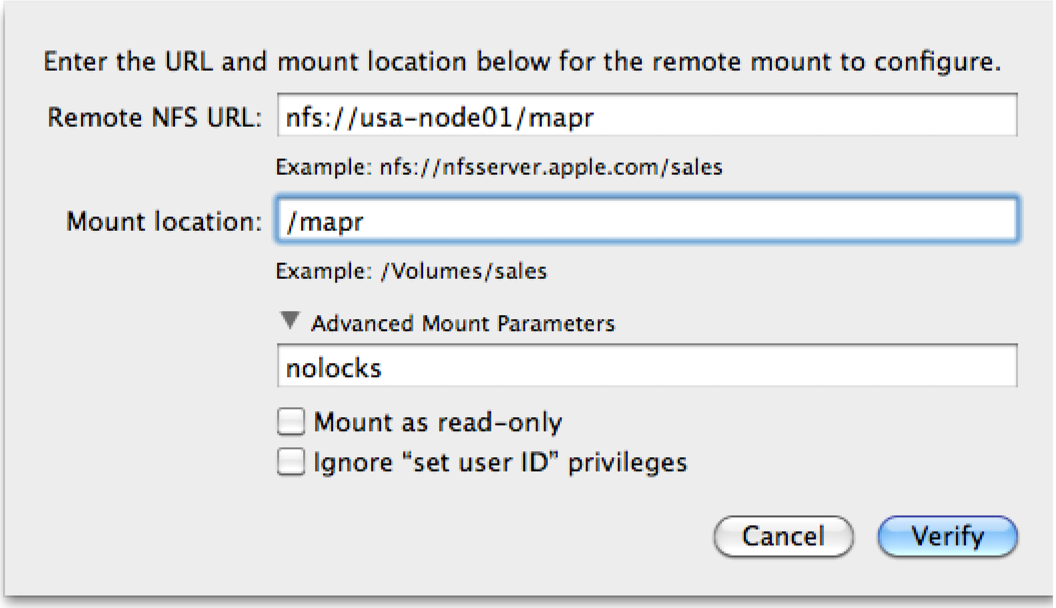
You can also add an NFS mount to `/etc/fstab` so that it mounts automatically when your system starts up. Example:

```
# device mountpoint fs-type options dump fsckorder
...
usa-node01:/mapr /mapr nfs rw 0 0
...
```

Mac

To mount the cluster from the Finder:

1. Open the Disk Utility: go to **Applications > Utilities > Disk Utility**.
2. Select **File > NFS Mounts**.
3. Click the **+** at the bottom of the NFS Mounts window.
4. In the dialog that appears, enter the following information:
   - **Remote NFS URL:** The URL for the NFS mount. If you do not know the URL, use the `showmount` command described below. Example: `nfs://usa-node01/mapr`
   - **Mount location:** The mount point where the NFS mount should appear in the local filesystem.



Enter the URL and mount location below for the remote mount to configure.

Remote NFS URL: nfs://usa-node01/mapr

Example: nfs://nfsserver.apple.com/sales

Mount location: /mapr

Example: /Volumes/sales

Advanced Mount Parameters: nolocks

□ Mount as read-only
□ Ignore "set user ID" privileges

5. Click the triangle next to **Advanced Mount Parameters**.
6. Enter **nolocks** in the text field.
7. Click **Verify**.
8. **Important:** On the dialog that appears, click **Don't Verify** to skip the verification process.

The Greenplum HD EE cluster should now appear at the location you specified as the mount point.

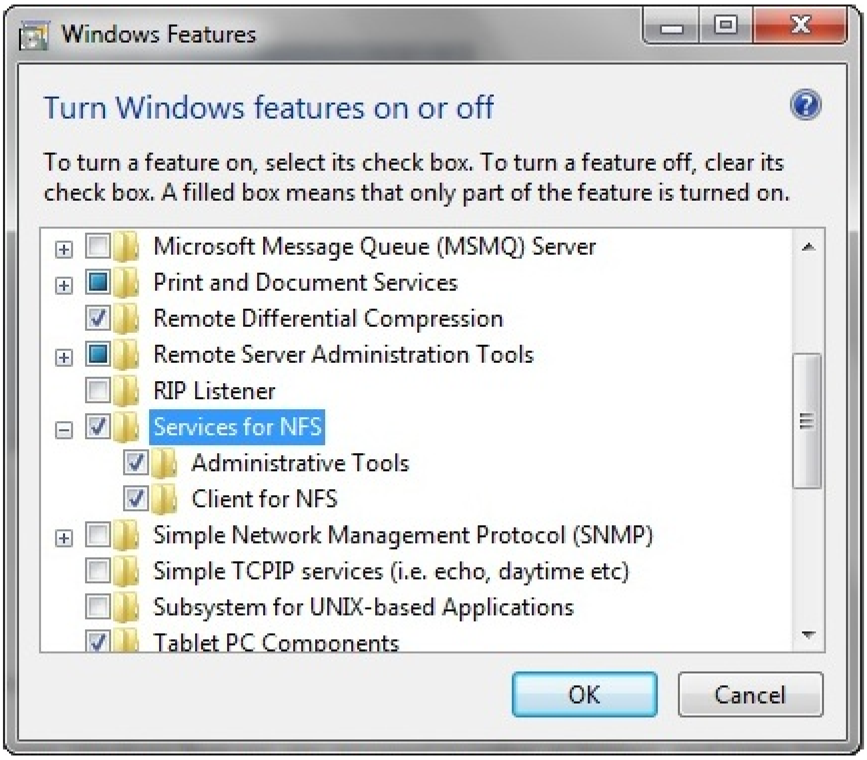
To mount the cluster from the command line:

1. List the NFS shares exported on the server. Example:
   ```
   showmount -e usa-node01
   ```
2. Set up a mount point for an NFS share. Example:
   ```
   sudo mkdir /mapr
   ```
3. Mount the cluster via NFS. Example:
   ```
   sudo mount -o nolock usa-node01:/mapr /mapr
   ```

Windows

> ⚠ Because of Windows directory caching, there may appear to be no .snapshot directory in each volume's root directory. To work around the problem, force Windows to re-load the volume's root directory by updating its modification time (for example, by creating an empty file or directory in the volume's root directory).

To mount the cluster on Windows 7 Ultimate or Windows 7 Enterprise:



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ⊞ □ Microsoft Message Queue (MSMQ) Server
- ⊞ ■ Print and Document Services
- ☑ Remote Differential Compression
- ⊞ ■ Remote Server Administration Tools
- □ RIP Listener
- ⊟ ☑ Services for NFS
  - ☑ Administrative Tools
  - ☑ Client for NFS
- ⊞ □ Simple Network Management Protocol (SNMP)
- □ Simple TCPIP services (i.e. echo, daytime etc)
- □ Subsystem for UNIX-based Applications
- ☑ Tablet PC Components

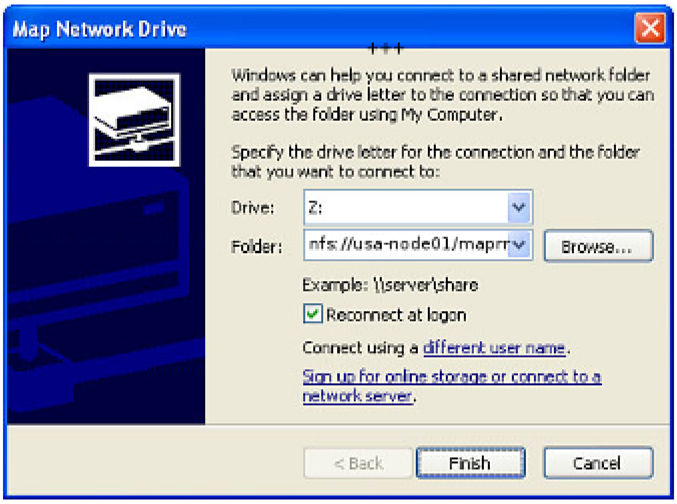
1. Open **Start > Control Panel > Programs**.
2. Select **Turn Windows features on or off**.
3. Select **Services for NFS**.
4. Click **OK**.
5. Mount the cluster and map it to a drive using the Map Network Drive tool or from the command line. Example:
   ```
   mount -o nolock usa-node01:/mapr z:
   ```

To mount the cluster on other Windows versions:

1. Download and install Microsoft Windows Services for Unix (SFU). You only need to install the NFS Client and the User Name Mapping.
2. Configure the user authentication in SFU to match the authentication used by the cluster (LDAP or operating system users). You can map local Windows users to cluster Linux users, if desired.
3. Once SFU is installed and configured, mount the cluster and map it to a drive using the Map Network Drive tool or from the command line. Example:

```
mount -o nolock usa-node01:/mapr z:
```

To map a network drive with the Map Network Drive tool:



Map Network Drive

Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

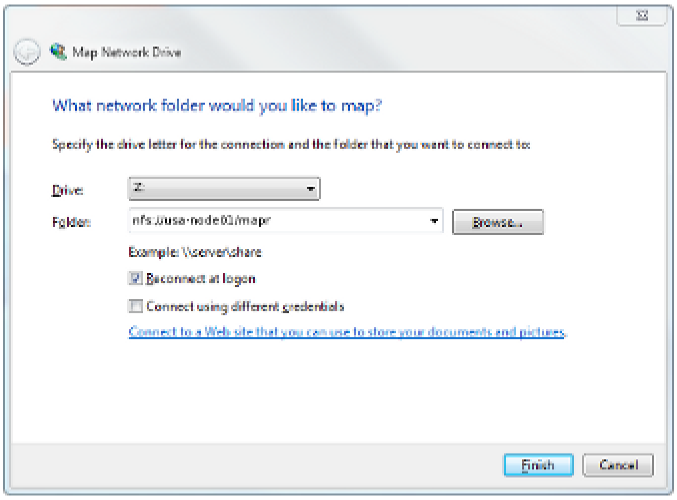
Folder: nfs://usa-node01/maprr

Example: \\server\share

☑ Reconnect at logon

Connect using a different user name.

Sign up for online storage or connect to a network server.



Map Network Drive

What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: nfs://usa-node01/mapr

Example: \\server\share

☑ Reconnect at logon

☐ Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures.

1. Open **Start > My Computer**.
2. Select **Tools > Map Network Drive**.
3. In the Map Network Drive window, choose an unused drive letter from the **Drive** drop-down list.
4. Specify the **Folder** by browsing for the Greenplum HD EE cluster, or by typing the hostname and directory into the text field.
5. Browse for the Greenplum HD EE cluster or type the name of the folder to map. This name must follow UNC. Alternatively, click the Browse… button to find the correct folder by browsing available network shares.
6. Select **Reconnect at login** to reconnect automatically to the Greenplum HD EE cluster whenever you log into the computer.
7. Click Finish.

See Direct Access NFS for more information.

Greenplum HD EE Client

The Greenplum HD EE client lets you interact with Greenplum HD EE directly. With the Greenplum HD EE client, you can submit Map/Reduce jobs and run `hadoop fs` and `hadoop mfs` commands. The Greenplum HD EE client is compatible with the following operating systems:

- CentOS 5.5 or above
- Red Hat Enterprise Linux 5.5 or above
- Mac OS X

> ⚠ Do not install the client on a cluster node. It is intended for use on a computer that has no other Greenplum HD EE software installed. Do not install other Greenplum HD EE software on a Greenplum HD EE client computer. To run MapR CLI commands, establish an `ssh` session to a node in the cluster.

To configure the client, you will need the cluster name and the IP addresses and ports of the CLDB nodes on the cluster. The configuration script `configure.sh` has the following syntax:

```
configure.sh [-N <cluster name>] -c -C <CLDB node>[:<port>][,<CLDB node>[:<port>]...]
```

Example:

```
/opt/mapr/server/configure.sh -N MyCluster -c -C 10.10.100.1:7222
```

Installing the Greenplum HD EE Client on CentOS or Red Hat

1. Change to the `root` user (or use sudo for the following commands).
2. Create a text file called `maprtech.repo` in the directory `/etc/yum.repos.d/` with the following contents:

```
[maprtech]
name=MapR Technologies
baseurl=http://package.mapr.com/releases/v1.1.1/redhat/
enabled=1
gpgcheck=0
protect=1
```

3. If your connection to the Internet is through a proxy server, you must set the `http_proxy` environment variable before installation:

```
http_proxy=http://<host>:<port>
export http_proxy
```

4. Remove any previous Greenplum HD EE software. You can use `rpm -qa | grep mapr` to get a list of installed Greenplum HD EE packages, then type the packages separated by spaces after the `rpm -e` command. Example:

```
rpm -qa | grep mapr
rpm -e mapr-fileserver mapr-core
```

5. Install the Greenplum HD EE client: `yum install mapr-client`
6. Run `configure.sh` to configure the client, using the `-C` (uppercase) option to specify the CLDB nodes, and the `-c` (lowercase) option to specify a client configuration. Example:

```
/opt/mapr/server/configure.sh -N MyCluster -c -C 10.10.100.1:7222
```

Installing the MapR Client on Mac OS X

1. Use the `extract` option to extract the contents of `mapr-client-1.1.1.11051GA-1.x86_64.tar.gz` into `/opt`
2. Run `configure.sh` to configure the client, using the `-C` (uppercase) option to specify the CLDB nodes, and the `-c` (lowercase) option to specify a client configuration.
   For example:

```
sudo /opt/mapr/server/configure.sh -N MyCluster -c -C 10.10.100.1:7222
```

Uninstalling Greenplum HD EE

To re-purpose machines, you may wish to remove nodes and uninstall Greenplum HD EE software.

Removing Nodes from a Cluster

To remove nodes from a cluster: first uninstall the desired nodes, then run `configure.sh` on the remaining nodes. Finally, if you are using Ganglia, restart all `gmeta` and `gmon` daemons in the cluster.

To uninstall a node:

On each node you want to uninstall, perform the following steps:

1. Change to the root user (or use sudo for the following commands).
2. Stop the Warden:
   `/etc/init.d/mapr-warden stop`
3. Remove the node (see Removing a Node).
4. If Pig is installed, remove it:
   - `erase mapr-pig-internal` (Red Hat or CentOS)
5. If Hive is installed, remove it:
   - `erase mapr-hive-internal` (Red Hat or CentOS)
6. If HBase (Master or RegionServer) is installed, remove it:
   - `erase mapr-hbase-internal` (Red Hat or CentOS)
7. Remove the package mapr-core:
   - `erase mapr-core` (Red Hat or CentOS)
8. If ZooKeeper is installed on the node, stop it:
   `/etc/init.d/mapr-zookeeper stop`
9. If ZooKeeper is installed, remove it:
   - `erase mapr-zk-internal` (Red Hat or CentOS)
10. If the node you have decommissioned is a CLDB node or a ZooKeeper node, then run `configure.sh` on all other nodes in the cluster (see Configuring a Node).

To reconfigure the cluster:

Run the script `configure.sh` to create `/opt/mapr/conf/mapr-clusters.conf` and update the corresponding `*.conf` and `*.xml` files. Before performing this step, make sure you have a list of the hostnames of the CLDB and ZooKeeper nodes. Optionally, you can specify the ports for the CLDB and ZooKeeper nodes as well. If you do not specify them, the default ports are:

- CLDB – 7222
- ZooKeeper – 5181

The script `configure.sh` takes an optional cluster name and log file, and comma-separated lists of CLDB and ZooKeeper host names or IP addresses (and optionally ports), using the following syntax:

```
/opt/mapr/server/configure.sh -C <host>[:<port>][,<host>[:<port>]...] -Z
<host>[:<port>][,<host>[:<port>]...] [-L <logfile>][-N <cluster name>]
```

Example:

```
/opt/mapr/server/configure.sh -C r1n1.sj.us:7222,r3n1.sj.us:7222,r5n1.sj.us:7222 -Z
r1n1.sj.us:5181,r2n1.sj.us:5181,r3n1.sj.us:5181,r4n1.sj.us:5181,r1n1.sj.us5:5181 -N
MyCluster
```

If you have not chosen a cluster name, you can run `configure.sh` again later to rename the cluster.

> ⚠ If you are using Ganglia, restart all gmeta and gmon daemons in the cluster. See Ganglia.

User Guide

This guide provides information about using Greenplum HD EE for Apache Hadoop, including the following topics:

- MapReduce - Provisioning resources and running Hadoop jobs
- Working with Data - Managing data protection, capacity, and performance with volumes and NFS
- Users and Groups - Working with users and groups, quotas, and permissions
- Managing the Cluster - Managing nodes, monitoring the cluster, and upgrading the Greenplum HD EE software
- Troubleshooting - Diagnosing and resolving problems

Volumes

A volume is a logical unit that allows you to apply policies to a set of files, directories, and sub-volumes. Using volumes, you can enforce disk usage limits, set replication levels, establish ownership and accountability, and measure the cost generated by different projects or departments. You can create a special type of volume called a *mirror*, a local or remote copy of an entire volume. Mirrors are useful for load balancing or disaster recovery. You can also create a *snapshot*, an image of a volume at a specific point in time. Snapshots are useful for rollback to a known data set. You can create snapshots manually or using a *sched ule*.

See also:

- Mirrors
- Snapshots
- Schedules

Greenplum HD EE lets you control and configure volumes in a number of ways:

- Replication - set the number of physical copies of the data, for robustness and performance
- Topology - restrict a volume to certain physical racks or nodes
- Quota - set a hard disk usage limit for a volume
- Advisory Quota - receive a notification when a volume exceeds a soft disk usage quota
- Ownership - set a user or group as the accounting entity for the volume
- Permissions - give users or groups permission to perform specified volume operations
- File Permissions - Unix-style read/write permissions on volumes

The following sections describe procedures associated with volumes:

- To create a new volume, see Creating a Volume (requires `cv` permission on the volume)
- To view a list of volumes, see Viewing a List of Volumes
- To view a single volume's properties, see Viewing Volume Properties
- To modify a volume, see Modifying a Volume (requires `m` permission on the volume)
- To mount a volume, see Mounting a Volume (requires `mnt` permission on the volume)
- To unmount a volume, see Unmounting a Volume (requires `m` permission on the volume)
- To remove a volume, see Removing a Volume (requires `d` permission on the volume)
- To set volume topology, see Setting Volume Topology (requires `m` permission on the volume)

Creating a Volume

When creating a volume, the only required parameters are the volume type (normal or mirror) and the volume name. You can set the ownership, permissions, quotas, and other parameters at the time of volume creation, or use the Volume Properties dialog to set them later. If you plan to schedule snapshots or mirrors, it is useful to create a schedule ahead of time; the schedule will appear in a drop-down menu in the Volume Properties dialog.

By default, the root user and the volume creator have full control permissions on the volume. You can grant specific permissions to other users and groups:

Code	Allowed Action
dump	Dump the volume
restore	Mirror or restore the volume
m	Modify volume properties, create and delete snapshots
d	Delete a volume
fc	Full control (admin access and permission to change volume ACL)

You can create a volume using the volume create command, or use the following procedure to create a volume using the Greenplum HD EE Control System.

To create a volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Click the **New Volume** button to display the New Volume dialog.
3. Use the **Volume Type** radio button at the top of the dialog to choose whether to create a standard volume, a local mirror, or a remote mirror.
4. Type a name for the volume or source volume in the **Volume Name** or **Mirror Name** field.
5. If you are creating a mirror volume:
    - Type the name of the source volume in the **Source Volume Name** field.
    - If you are creating a remote mirror volume, type the name of the cluster where the source volume resides, in the **Source Cluster Name** field.
6. You can set a mount path for the volume by typing a path in the **Mount Path** field.
7. You can specify which rack or nodes the volume will occupy by typing a path in the **Topology** field.

8. You can set permissions using the fields in the **Ownership & Permissions** section:
    a. Click **[+ Add Permission]** to display fields for a new permission.
    b. In the left field, type either u: and a user name, or g: and a group name.
    c. In the right field, select permissions to grant to the user or group.
9. You can associate a standard volume with an accountable entity and set quotas in the **Usage Tracking** section:
    a. In the **Group/User** field, select **User** or **Group** from the dropdown menu and type the user or group name in the text field.
    b. To set an advisory quota, select the checkbox beside **Volume Advisory Quota** and type a quota (in megabytes) in the text field.
    c. To set a quota, select the checkbox beside **Volume Quota** and type a quota (in megabytes) in the text field.
10. You can set the replication factor and choose a snapshot or mirror schedule in the Replication and Snapshot section:
    a. Type the desired replication factor in the **Replication Factor** field.
    b. To disable writes when the replication factor falls before a minimum number, type the minimum replication factor in the **Disable Writes...** field.
    c. To schedule snapshots or mirrors, select a schedule from the **Snapshot Schedule** dropdown menu or the **Mirror Update Schedule** dropdown menu respectively.
11. Click **OK** to create the volume.

Viewing a List of Volumes

You can view all volumes using the volume list command, or view them in the Greenplum HD EE Control System using the following procedure.

To view all volumes using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.

Viewing Volume Properties

You can view volume properties using the volume info command, or use the following procedure to view them using the Greenplum HD EE Control System.

To view the properties of a volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Display the Volume Properties dialog by clicking the volume name, or by selecting the checkbox beside the volume name, then clicking the **Properties** button.
3. After examining the volume properties, click **Close** to exit without saving changes to the volume.

Modifying a Volume

You can modify any attributes of an existing volume, except for the following restriction:

- You cannot convert a normal volume to a mirror volume.

You can modify a volume using the volume modify command, or use the following procedure to modify a volume using the Greenplum HD EE Control System.

To modify a volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Display the Volume Properties dialog by clicking the volume name, or by selecting the checkbox beside the volume name then clicking the **Properties** button.
3. Make changes to the fields. See Creating a Volume for more information about the fields.
4. After examining the volume properties, click **Modify Volume** to save changes to the volume.

Mounting a Volume

You can mount a volume using the volume mount command, or use the following procedure to mount a volume using the Greenplum HD EE Control System.

To mount a volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the name of each volume you wish to mount.
3. Click the **Mount** button.

You can also mount or unmount a volume using the **Mounted** checkbox in the Volume Properties dialog. See Modifying a

Volume for more information.

Unmounting a Volume

You can unmount a volume using the volume unmount command, or use the following procedure to unmount a volume using the Greenplum HD EE Control System.

To unmount a volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the name of each volume you wish to unmount.
3. Click the **Unmount** button.

You can also mount or unmount a volume using the Mounted checkbox in the Volume Properties dialog. See Modifying a Volume for more information.

Removing a Volume or Mirror

You can remove a volume using the volume remove command, or use the following procedure to remove a volume using the Greenplum HD EE Control System.

To remove a volume or mirror using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Click the checkbox next to the volume you wish to remove.
3. Click the **Remove** button to display the Remove Volume dialog.
4. In the Remove Volume dialog, click the **Remove Volume** button.

Setting Volume Topology

You can place a volume on specific racks, nodes,or groups of nodes by setting its topology to an existing node topology. For more information about node topology, see Node Topology.

To set volume topology, choose the path that corresponds to the node topology of the rack or nodes where you would like the volume to reside. You can set volume topology using the Greenplum HD EE Control System or with the volume modify command.

To set volume topology using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Display the Volume Properties dialog by clicking the volume name or by selecting the checkbox beside the volume name, then clicking the **Properties** button.
3. Click **Move Volume** to display the Move Volume dialog.
4. Select a topology path that corresponds to the rack or nodes where you would like the volume to reside.
5. Click **Move Volume** to return to the Volume Properties dialog.
6. Click **Modify Volume** to save changes to the volume.

Setting Default Volume Topology

By default, new volumes are created with a topology of `/` (root directory). To change the default topology, use the `config save` command to change the `cldb.default.volume.topology` configuration parameter. Example:

```
maprcli config save -values "{\"cldb.default.volume.topology\":\"/default-rack\"}"
```

After running the above command, new volumes have the volume topology `/default-rack` by default.

Example: Setting Up CLDB-Only Nodes

In a large cluster, it might be desirable to prevent nodes that contain the CLDB volume from storing other data, by creating CLDB-only nodes. This configuration provides additional control over the placement of the CLDB data, for load balancing, fault tolerance, or high availability (HA). Setting up CLDB-only nodes involves restricting the CLDB volume to its own topology and making sure all other volumes are on a separate topology. By default, new volumes have no topology when they are created, and reside at the root topology path: "`/`". Because both the CLDB-only path and the non-CLDB path are children of the root topology path, new non-CLDB volumes are not guaranteed to keep off the CLDB-only nodes unless their topologies are set explicitly on creation. Similarly, any node added to the cluster after setting up CLDB-only nodes must be moved explicitly to the non-CLDB topology or the CLDB-only topology, depending on its intended use.

To restrict the CLDB volume to specific nodes:

1. Move all CLDB nodes to a CLDB-only topology (e. g. `/cldbonly`) using the Greenplum HD EE Control System or the following command:
   `maprcli node move -serverids <CLDB nodes> -topology /cldbonly`
2. Restrict the CLDB volume to the CLDB-only topology. Use the Greenplum HD EE Control System or the following command:
   `maprcli volume move -name mapr.cldb.internal -topology /cldbonly`
3. If the CLDB volume is present on nodes not in /cldbonly, increase the replication factor of mapr.cldb.internal to create enough copies in
   /cldbonly using the Greenplum HD EE Control System or the following command:
   `maprcli volume modify -name mapr.cldb.internal -replication <replication factor>`
4. Once the volume has sufficient copies, remove the extra replicas by reducing the replication factor to the desired value using the Greenplum HD EE Control System or the command used in the previous step.

To move all other volumes to a topology separate from the CLDB-only nodes:

1. Move all non-CLDB nodes to a non-CLDB topology (e. g. `/defaultRack`) using the Greenplum HD EE Control System or the following command:
   `maprcli node move -serverids <all non-CLDB nodes> -topology /defaultRack`
2. Restrict all existing volumes to the topology `/defaultRack` using the Greenplum HD EE Control System or the following command:
   `maprcli volume move -name <volume> -topology /defaultRack`
   All volumes except (mapr.cluster.root) get re-replicated to the changed topology automatically.

> ⚠ To prevent subsequently created volumes from encroaching on the CLDB-only nodes, set a default topology that excludes the CLDB-only topology.

Mirrors

A *mirror volume* is a read-only physical copy of another volume, the *source volume*. Creating mirrors on the same cluster (local mirroring) is useful for load balancing and local backup. Creating mirrors on another cluster (remote mirroring) is useful for wide distribution and disaster preparedness. Creating a mirror is similar to creating a normal (read/write) volume, except that you must specify a source volume from which the mirror retrieves its contents (the *mirroring operation*). When you mirror a volume, read requests to the source volume can be served by any of its mirrors on the same cluster via a volume link of type `mirror`. A volume link is similar to a normal volume mount point, except that you can specify whether it points to the source volume or its mirrors.

- To write to (and read from) the source volume, mount the source volume normally. As long as the source volume is mounted below a non-mirrored volume, you can read and write to the volume normally via its direct mount path. You can also use a volume link of type `writeable` to write directly to the source volume regardless of its mount point.
- To read from the mirrors, use the volume link create command to make a volume link (of type `mirror`) to the source volume. Any reads via the volume link will be distributed among the volume's mirrors. It is not necessary to also mount the mirrors, because the volume link handles access to the mirrors.

Any mount path that consists entirely of mirrored volumes will refer to a mirrored copy of the target volume; otherwise the mount path refers to the specified volume itself. For example, assume a mirrored volume `c` mounted at `/a/b/c`. If the root volume is mirrored, then the mount path `/` refers to a mirror of the root volume; if `a` in turn is mirrored, then the path `/a` refers to a mirror of `a` and so on. If all volumes preceding `c` in the mount path are mirrored, then the path `/a/b/c` refers to one of the mirrors of `c`. However, if any volume in the path is not mirrored then the source volume is selected for that volume and subsequent volumes in the path. If `a` is not mirrored, then although `/` still selects a mirror, `/a` refers to the source volume `a` itself (because there is only one) and `/a/b` refers to the source volume `b` (because it was not accessed via a mirror). In that case, `/a/b/c` refers to the source volume `c`.

Any mirror that is accessed via a parent mirror (all parents are mirror volumes) is implicitly mounted. For example, assume a volume `a` that is mirrored to `a-mirror`, and a volume `b` that is mirrored to `b-mirror-1` and `b-mirror-2`; `a` is mounted at `/a`, `b` is mounted at `/a/b`, and `a-mirror` is mounted at `/a-mirror`. In this case, reads via `/a-mirror/b` will access one of the mirrors `b-mirror-1` or `b-mirror-2` without the requirement to explicitly mount them.

At the start of a mirroring operation, a temporary snapshot of the source volume is created; the mirroring process reads from the snapshot so that the source volume remains available for both reads and writes during mirroring. To save bandwidth, the mirroring process transmits only the deltas between the source volume and the mirror; after the initial mirroring operation (which creates a copy of the entire source volume), subsequent updates can be extremely fast.

Mirroring is extremely resilient. In the case of a *network partition* (some or all machines where the source volume resides cannot communicate with machines where the mirror volume resides), the mirroring operation will periodically retry the connection, and will complete mirroring when the network is restored.

Working with Mirrors

The following sections provide information about various mirroring use cases.

Local and Remote Mirroring

Local mirroring (creating mirrors on the same cluster) is useful for load balancing, or for providing a read-only copy of a data set.

Although it is not possible to directly mount a volume from one cluster to another, you can mirror a volume to a remote cluster (*remote mirroring*). By mirroring the cluster's root volume and all other volumes in the cluster, you can create an entire mirrored cluster that keeps in sync with the source cluster. Mount points are resolved within each cluster; any volumes that are mirrors of a source volume on another cluster are read-only, because a source volume from another cluster cannot be resolved locally.

To transfer large amounts of data between physically distant clusters, you can use the `volume dump create` command to create volume copies for transport on physical media. The `volume dump create` command creates backup files containing the volumes, which can be reconstituted into mirrors at the remote cluster using the `volume dump restore` command. These mirrors can be reassociated with the source volumes (using the `volume modify` command to specify the source for each mirror volume) for live mirroring.

Local Mirroring Example

Assume a volume containing a table of data that will be read very frequently by many clients, but updated infrequently. The data is contained in a volume named `table-data`, which is to be mounted under a non-mirrored user volume belonging to `jsmith`. The mount path for the writeable copy of the data is to be `/home/private/users/jsmith/private-table` and the public, readable mirrors of the data are to be mounted at `/public/data/table`. You would set it up as follows:

1. Create as many mirror volumes as needed for the data, using the Greenplum HD EE Control System or the volume create command (See Creating a Volume).
2. Mount the source volume at the desired location (in this case, `/home/private/users/jsmith/private-table`) using the Greenplum HD EE Control System or the volume mount command.
3. Use the volume link create command to create a volume link at `/public/data/table` pointing to the source volume. Example:

   ```
   maprcli volume link create -volume table-data -type mirror -path /public/data/table
   ```

4. Write the data to the source volume via the mount path `/home/private/users/jsmith/private-table` as needed.
5. When the data is ready for public consumption, use the volume mirror push command to push the data out to all the mirrors.
6. Create additional mirrors as needed and push the data to them. No additional steps are required; as soon as a mirror is created and synchronized, it is available via the volume link.

When a user reads via the path `/public/data/table`, the data is served by a randomly selected mirror of the source volume. Reads are evenly spread over all mirrors.

Remote Mirroring Example

Assume two clusters, `cluster-1` and `cluster-2`, and a volume `volume-a` on `cluster-1` to be mirrored to `cluster-2`. Create a mirror volume on `cluster-2`, specifying the remote cluster and volume. You can create remote mirrors using the Greenplum HD EE Control System or the volume create command:

- In the Greenplum HD EE Control System on `cluster-2`, specify the following values in the New Volume dialog:
  1. Select **Remote Mirror Volume**.
  2. Enter `volume-a` or another name in the **Volume Name** field.
  3. Enter `volume-a` in the **Source Volume** field.
  4. Enter `cluster-1` in the **Source Cluster** field.

- Using the volume create command `cluster-2`, specify the following parameters:
  - Specify the source volume and cluster in the format `<volume>@<cluster>`, provide a `name` for the mirror volume, and specify a `type` of `1`. Example:

    ```
    maprcli volume create -name volume-a -source volume-a@cluster-1 -type 1
    ```

After creating the mirror volume, you can synchronize the data using volume mirror start from `cluster-2` to pull data to the mirror volume on `cluster-2` from its source volume on `cluster-1`.

When you mount a mirror volume on a remote cluster, any mirror volumes below it are automatically mounted. For example, assume volumes `a` and `b` on `cluster-1` (mounted at `/a` and `/a/b`) are mirrored to `a-mirror` and `b-mirror` on `cluster-2`. When you mount the volume `a-mirror` at `/a-mirror` on `cluster-2`, it contains a mount point for `/b` which gets mapped to the mirror of `b`, making it available at `/a-mirror/b`. Any mirror volumes below `b` will be similarly mounted, and so on.

Mirroring the Root Volume

The most frequently accessed volumes in a cluster are likely to be the root volume and its immediate children. In order to load-balance reads on these volumes, it is possible to mirror the root volume (typically `mapr.cluster.root`, which is mounted at `/`). There is a special writeable volume link called `.rw` inside the root volume, to provide access to the source volume. In other words, if the root volume is mirrored:

- The path `/` refers to one of the mirrors of the root volume
- The path `/.rw` refers to the source (writeable) root volume

Mirror Cascades

A *mirror cascade* is a series of mirrors that form a chain from a single source volume: the first mirror receives updates from the source volume, the second mirror receives updates from the first, and so on. Mirror cascades are useful for propagating data over a distance, then re-propagating the data locally instead of transferring the same data remotely again.

You can create or break a mirror cascade made from existing mirror volumes by changing the source volume of each mirror in the Volume Properties dialog.

Creating, Modifying, and Removing Mirror Volumes

ou can create a mirror manually or automate the process with a schedule. You can set the topology of a mirror volume to determine the placement of the data, if desired. The following sections describe procedures associated with mirrors:

- To create a new mirror volume, see Creating a Volume (requires `M5` license and `cv` permission)
- To modify a mirror (including changing its source), see Modifying a Volume
- To remove a mirror, see Removing a Volume or Mirror

You can change a mirror's source volume by changing the source volume in the Volume Properties dialog.

Starting a Mirror

To *start* a mirror means to pull the data from the source volume. Before starting a mirror, you must create a mirror volume and associate it with a source volume. You should start a mirror operation shortly after creating the mirror volume, and then again each time you want to synchronize the mirror with the source volume. You can use a schedule to automate the synchronization. If you create a mirror and synchronize it only once, it is like a snapshot except that it uses the same amount of disk space used by the source volume at the point in time when the mirror was started. You can start a mirror using the volume mirror start command , or use the following procedure to start mirroring using the Greenplum HD EE Control System.

To start mirroring using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the name of each volume you wish to mirror.
3. Click the **Start Mirroring** button.

Stopping a Mirror

To *stop* a mirror means to cancel the replication or synchronization process. Stopping a mirror does not delete or remove the mirror volume, only stops any synchronization currently in progress.
You can stop a mirror using the volume mirror stop command, or use the following procedure to stop mirroring using the Greenplum HD EE Control System.

To stop mirroring using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the name of each volume you wish to stop mirroring.
3. Click the **Stop Mirroring** button.

Pushing Changes to Mirrors

To *push* a mirror means to start pushing data from the source volume to all its local mirrors. See Pushing Changes to All Mirrors. You can push source volume changes out to all mirrors using the volume mirror push command, which returns after the data has been pushed.

Schedules

A schedule is a group of rules that specify recurring points in time at which certain actions are determined to occur. You can use schedules to automate the creation of snapshots and mirrors; after you create a schedule, it appears as a choice in the scheduling menu when you are editing the properties of a task that can be scheduled:

- To apply a schedule to snapshots, see Scheduling a Snapshot.

- To apply a schedule to volume mirroring, see Creating a Volume.

The following sections provide information about the actions you can perform on schedules:

- To create a schedule, see Creating a Schedule
- To view a list of schedules, see Viewing a List of Schedules
- To modify a schedule, see Modifying a Schedule
- To remove a schedule, see Removing a Schedule

Creating a Schedule

You can create a schedule using the schedule create command, or use the following procedure to create a schedule using the Greenplum HD EE Control System.

To create a schedule using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Schedules view.
2. Click **New Schedule**.
3. Type a name for the new schedule in the **Schedule Name** field.
4. Define one or more schedule rules in the **Schedule Rules** section:
    a. From the first dropdown menu, select a frequency (Once, Yearly, Monthly, etc.))
    b. From the next dropdown menu, select a time point within the specified frequency. For example: if you selected Monthly in the first dropdown menu, select the day of the month in the second dropdown menu.
    c. Continue with each dropdown menu, proceeding to the right, to specify the time at which the scheduled action is to occur.
    d. Use the **Retain For** field to specify how long the data is to be preserved. For example: if the schedule is attached to a volume for creating snapshots, the Retain For field specifies how far after creation the snapshot expiration date is set.
5. Click **[+ Add Rule]** to specify additional schedule rules, as desired.
6. Click **Save Schedule** to create the schedule.

Viewing a List of Schedules

You can view a list of schedules using the schedule list command, or use the following procedure to view a list of schedules using the Greenplum HD EE Control System.

To view a list of schedules using the Greenplum HD EE Control System:

- In the Navigation pane, expand the **MapR-FS** group and click the Schedules view.

Modifying a Schedule

When you modify a schedule, the new set of rules replaces any existing rules for the schedule.

You can modify a schedule using the schedule modify command, or use the following procedure to modify a schedule using the Greenplum HD EE Control System.

To modify a schedule using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Schedules view.
2. Click the name of the schedule to modify.
3. Modify the schedule as desired:
    a. Change the schedule name in the **Schedule Name** field.
    b. Add, remove, or modify rules in the **Schedule Rules** section.
4. Click **Save Schedule** to save changes to the schedule.

For more information, see Creating a Schedule.

Removing a Schedule

You can remove a schedule using the schedule remove command, or use the following procedure to remove a schedule using the Greenplum HD EE Control System.

To remove a schedule using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Schedules view.
2. Click the name of the schedule to remove.
3. Click **Remove Schedule** to display the Remove Schedule dialog.
4. Click **Yes** to remove the schedule.

Snapshots

A snapshot is a read-only image of a volume at a specific point in time. On a Greenplum HD EE cluster, you can create a snapshot manually or automate the process with a schedule. Snapshots are useful any time you need to be able to roll back to a known good data set at a specific point in time. For example, before performing a risky operation on a volume, you can create a snapshot to enable "undo" capability for the entire volume. A snapshot takes no time to create, and initially uses no disk space, because it stores only the incremental changes needed to roll the volume back to the point in time when the snapshot was created.

The following sections describe procedures associated with snapshots:

- To view the contents of a snapshot, see Viewing the Contents of a Snapshot
- To create a snapshot, see Creating a Volume Snapshot (requires `M5` license)
- To view a list of snapshots, see Viewing a List of Snapshots
- To remove a snapshot, see Removing a Volume Snapshot

Viewing the Contents of a Snapshot

At the top level of each volume is a directory called `.snapshot` containing all the snapshots for the volume. You can view the directory with `hadoop fs` commands or by mounting the cluster with NFS. To prevent recursion problems, `ls` and `hadoop fs -ls` do not show the `.snapshot` directory when the top-level volume directory contents are listed. You must navigate explicitly to the `.snapshot` directory to view and list the snapshots for the volume.

Example:

```
root@node41:/opt/mapr/bin# hadoop fs -ls /myvol/.snapshot
Found 1 items
drwxrwxrwx   - root root          1 2011-06-01 09:57 /myvol/.snapshot/2011-06-01.09-57-49
```

Creating a Volume Snapshot

You can create a snapshot manually or use a schedule to automate snapshot creation. Each snapshot has an expiration date that determines how long the snapshot will be retained:

- When you create the snapshot manually, specify an expiration date.
- When you schedule snapshots, the expiration date is determined by the Retain parameter of the schedule.

For more information about scheduling snapshots, see Scheduling a Snapshot.

Creating a Snapshot Manually

You can create a snapshot using the volume snapshot create command, or use the following procedure to create a snapshot using the Greenplum HD EE Control System.

To create a snapshot using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the name of each volume for which you want a snapshot, then click the **New Snapshot** button to display the Snapshot Name dialog.
3. Type a name for the new snapshot in the **Name...** field.
4. Click **OK** to create the snapshot.

Scheduling a Snapshot

You schedule a snapshot by associating an existing schedule with a normal (non-mirror) volume. You cannot schedule snapshots on mirror volumes; in fact, since mirrors are read-only, creating a snapshot of a mirror would provide no benefit. You can schedule a snapshot by passing the ID of a schedule to the volume modify command, or you can use the following procedure to choose a schedule for a volume using the Greenplum HD EE Control System.

To schedule a snapshot using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Display the Volume Properties dialog by clicking the volume name, or by selecting the checkbox beside the name of the volume then clicking the **Properties** button.
3. In the Replication and Snapshot Scheduling section, choose a schedule from the **Snapshot Schedule** dropdown menu.
4. Click **Modify Volume** to save changes to the volume.

For information about creating a schedule, see Schedules.

Viewing a List of Snapshots

Viewing all Snapshots

You can view snapshots for a volume with the volume snapshot list command or using the Greenplum HD EE Control System.

To view snapshots using the Greenplum HD EE Control System:

- In the Navigation pane, expand the **MapR-FS** group and click the Snapshots view.

Viewing Snapshots for a Volume

You can view snapshots for a volume by passing the volume to the volume snapshot list command or using the Greenplum HD EE Control System.

To view snapshots using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Click the **Snapshots** button to display the Snapshots for Volume dialog.

Removing a Volume Snapshot

Each snapshot has an expiration date and time, when it is deleted automatically. You can remove a snapshot manually before its expiration, or you can preserve a snapshot to prevent it from expiring.

Removing a Volume Snapshot Manually

You can remove a snapshot using the volume snapshot remove command, or use the following procedure to remove a snapshot using the Greenplum HD EE Control System.

To remove a snapshot using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Snapshots view.
2. Select the checkbox beside each snapshot you wish to remove.
3. Click **Remove Snapshot** to display the Remove Snapshots dialog.
4. Click **Yes** to remove the snapshot or snapshots.

To remove a snapshot from a specific volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the volume name.
3. Click Snapshots to display the Snapshots for Volume dialog.
4. Select the checkbox beside each snapshot you wish to remove.
5. Click **Remove** to display the Remove Snapshots dialog.
6. Click **Yes** to remove the snapshot or snapshots.

Preserving a Volume Snapshot

You can preserve a snapshot using the volume snapshot preserve command, or use the following procedure to create a volume using the Greenplum HD EE Control System.

To remove a snapshot using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Snapshots view.
2. Select the checkbox beside each snapshot you wish to preserve.
3. Click **Preserve Snapshot** to preserve the snapshot or snapshots.

To remove a snapshot from a specific volume using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the volume name.
3. Click Snapshots to display the Snapshots for Volume dialog.
4. Select the checkbox beside each snapshot you wish to preserve.
5. Click **Preserve** to preserve the snapshot or snapshots.

Direct Access NFS

Unlike other Hadoop distributions which only allow cluster data import or import as a batch operation, Greenplum HD EE lets you mount the cluster itself via NFS so that your applications can read and write data directly. Greenplum HD EE allows direct file modification and multiple concurrent reads and writes via POSIX semantics. With an NFS-mounted cluster, you can read and write data directly with standard tools, applications, and scripts. For example, you could run a MapReduce job that outputs to a CSV file, then import the CSV file directly into SQL via NFS.

Greenplum HD EE exports each cluster as the directory `/mapr/<cluster name>` (for example, `/mapr/default`). If you create a mount point with the local path `/mapr`, then Hadoop FS paths and NFS paths to the cluster will be the same. This makes it easy to work on the same files via NFS and Hadoop. In a multi-cluster setting, the clusters share a single namespace, and you can see them all by mounting the top-level `/mapr` directory.

Mounting the Cluster

Before you begin, make sure you know the hostname and directory of the NFS share you plan to mount.
Example:

- `usa-node01:/mapr` - for mounting from the command line
- `nfs://usa-node01/mapr` - for mounting from the Mac Finder

Make sure the client machine has the appropriate username and password to access the NFS share. For best results, the username and password for accessing the Greenplum HD EE cluster should be the same username and password used to log into the client machine.

Linux

1. Make sure the NFS client is installed.
2. List the NFS shares exported on the server. Example:
   `showmount -e usa-node01`
3. Set up a mount point for an NFS share. Example:
   `sudo mkdir /mapr`
4. Mount the cluster via NFS. Example:
   `sudo mount usa-node01:/mapr /mapr`

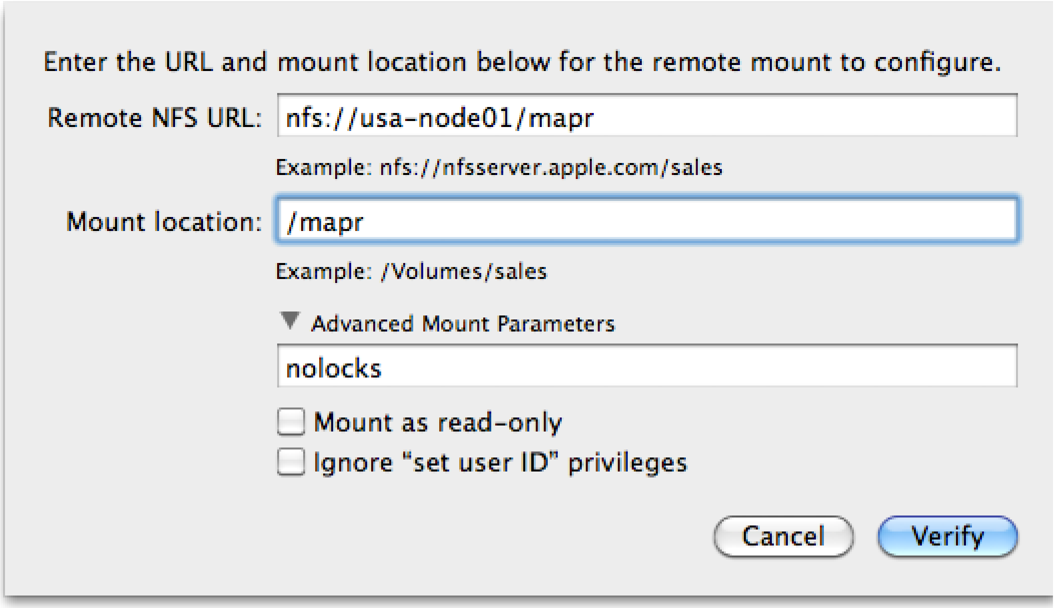
You can also add an NFS mount to `/etc/fstab` so that it mounts automatically when your system starts up. Example:

```
# device mountpoint fs-type options dump fsckorder
...
usa-node01:/mapr /mapr nfs rw 0 0
...
```

Mac

To mount the cluster from the Finder:

1. Open the Disk Utility: go to **Applications > Utilities > Disk Utility**.
2. Select **File > NFS Mounts**.
3. Click the **+** at the bottom of the NFS Mounts window.
4. In the dialog that appears, enter the following information:
   - **Remote NFS URL:** The URL for the NFS mount. If you do not know the URL, use the `showmount` command described below. Example: `nfs://usa-node01/mapr`
   - **Mount location:** The mount point where the NFS mount should appear in the local filesystem.



Enter the URL and mount location below for the remote mount to configure.

Remote NFS URL: nfs://usa-node01/mapr

Example: nfs://nfsserver.apple.com/sales

Mount location: /mapr

Example: /Volumes/sales

▼ Advanced Mount Parameters

nolocks

□ Mount as read-only

□ Ignore "set user ID" privileges

5. Click the triangle next to **Advanced Mount Parameters**.
6. Enter **nolocks** in the text field.
7. Click **Verify**.
8. **Important:** On the dialog that appears, click **Don't Verify** to skip the verification process.

The Greenplum HD EE cluster should now appear at the location you specified as the mount point.

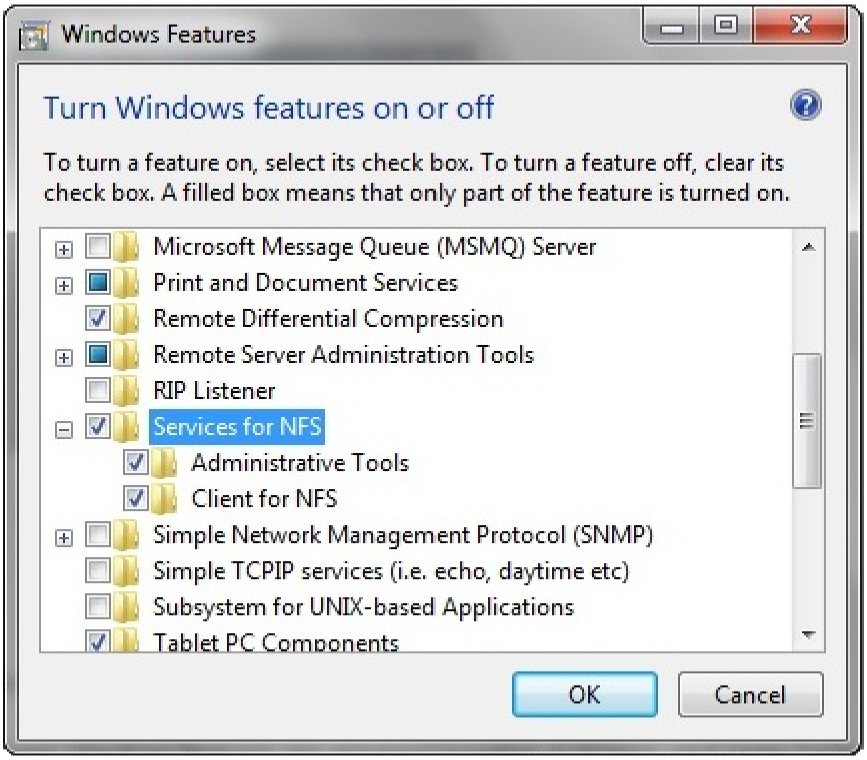
To mount the cluster from the command line:

1. List the NFS shares exported on the server. Example:
   ```
   showmount -e usa-node01
   ```
2. Set up a mount point for an NFS share. Example:
   ```
   sudo mkdir /mapr
   ```
3. Mount the cluster via NFS. Example:
   ```
   sudo mount -o nolock usa-node01:/mapr /mapr
   ```

Windows

⚠ Because of Windows directory caching, there may appear to be no .snapshot directory in each volume's root directory. To work around the problem, force Windows to re-load the volume's root directory by updating its modification time (for example, by creating an empty file or directory in the volume's root directory).

To mount the cluster on Windows 7 Ultimate or Windows 7 Enterprise:



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

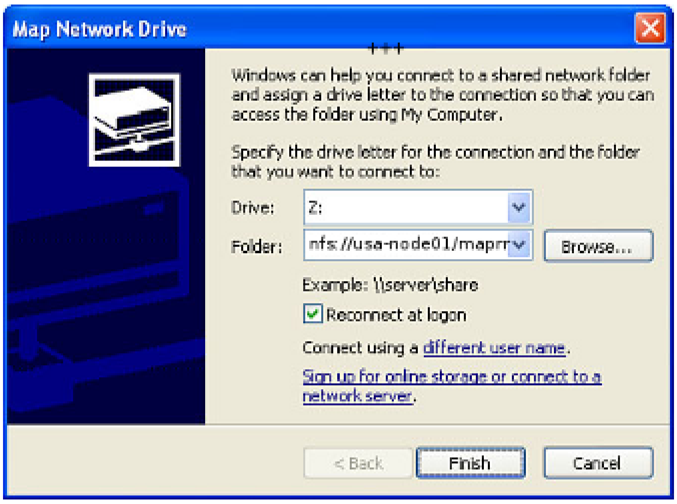
- ☐ Microsoft Message Queue (MSMQ) Server
- ■ Print and Document Services
- ☑ Remote Differential Compression
- ■ Remote Server Administration Tools
- ☐ RIP Listener
- ☑ Services for NFS
  - ☑ Administrative Tools
  - ☑ Client for NFS
- ☐ Simple Network Management Protocol (SNMP)
- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ☐ Subsystem for UNIX-based Applications
- ☑ Tablet PC Components

1. Open **Start > Control Panel > Programs**.
2. Select **Turn Windows features on or off**.
3. Select **Services for NFS**.
4. Click **OK**.
5. Mount the cluster and map it to a drive using the Map Network Drive tool or from the command line. Example:
   `mount -o nolock usa-node01:/mapr z:`

To mount the cluster on other Windows versions:

1. Download and install Microsoft Windows Services for Unix (SFU). You only need to install the NFS Client and the User Name Mapping.
2. Configure the user authentication in SFU to match the authentication used by the cluster (LDAP or operating system users). You can map local Windows users to cluster Linux users, if desired.
3. Once SFU is installed and configured, mount the cluster and map it to a drive using the Map Network Drive tool or from the command line. Example:
   `mount -o nolock usa-node01:/mapr z:`

To map a network drive with the Map Network Drive tool:



Map Network Drive

Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

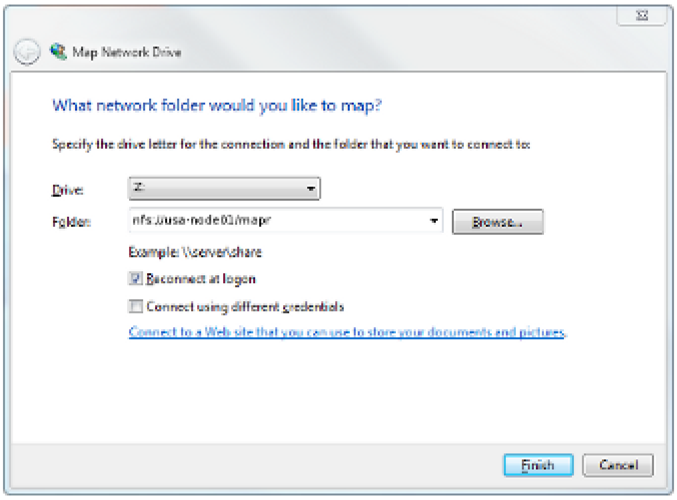
Folder: nfs://usa-node01/mapr

Example: \\server\share

☑ Reconnect at logon

Connect using a different user name.

Sign up for online storage or connect to a network server.



What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: nfs://usa-node01/mapr

Example: \\server\share

☑ Reconnect at logon

□ Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures.

1. Open **Start > My Computer**.
2. Select **Tools > Map Network Drive**.
3. In the Map Network Drive window, choose an unused drive letter from the **Drive** drop-down list.
4. Specify the **Folder** by browsing for the Greenplum HD EE cluster, or by typing the hostname and directory into the text field.
5. Browse for the Greenplum HD EE cluster or type the name of the folder to map. This name must follow UNC. Alternatively, click the Browse… button to find the correct folder by browsing available network shares.
6. Select **Reconnect at login** to reconnect automatically to the Greenplum HD EE cluster whenever you log into the computer.
7. Click Finish.

Setting Compression and Chunk Size

Each directory in Greenplum hD EE storage contains a hidden file called `.dfs_attributes` that controls compression and chunk size. To change these attributes, change the corresponding values in the file.

Valid values:

- Compression: `true` or `false`
- Chunk size (in bytes): a multiple of 65535 (64 K) or zero (no chunks). Example: `131072`

You can also set compression and chunksize using the hadoop mfs command.

By default, Greenplum HD EE does not compress files whose filename extension indicate they are already compressed. The default list of filename extensions is as follows:

- bz2
- gz
- tgz
- tbz2
- zip
- z
- Z
- mp3
- jpg
- jpeg
- mpg
- mpeg
- avi
- gif
- png

The list of fileneme extensions not to compress is stored as comma-separated values in the `mapr.fs.nocompression` configuration parameter, and can be modified with the `config save` command. Example:

```
maprcli config save -values {"mapr.fs.nocompression":"bz2,gz,tgz,tbz2,zip,z,Z,mp3,jpg,jpe
g,mpg,mpeg,avi,gif,png"}
```

The list can be viewed with the `config load` command. Example:

```
maprcli config load -keys mapr.fs.nocompression
```

MapReduce

If you have used Hadoop in the past to run MapReduce jobs, then running jobs on Greenplum HD EE for Apache Hadoop will be very familiar to you. Greenplum HD EE is a full Hadoop distribution, API-compatible with all versions of Hadoop. Greenplum HD EE provides additional capabilities not present in any other Hadoop distribution. This section contains information about the following topics:

- Tuning MapReduce - Strategies for optimizing resources to meet the goals of your application

ExpressLane

Greenplum HD EE provides an express path for small MapReduce jobs to run when all slots are occupied by long tasks. Small jobs are only given this special treatment when the cluster is busy, and only if they meet the criteria specified by the following parameters in `mapred-site.xml`:

Parameter	Value	Description
mapred.fairscheduler.smalljob.schedule.enable	true	Enable small job fast scheduling inside fair scheduler. TaskTrackers should reserve a slot called ephemeral slot which is used for smalljob if cluster is busy.
mapred.fairscheduler.smalljob.max.maps	10	Small job definition. Max number of maps allowed in small job.
mapred.fairscheduler.smalljob.max.reducers	10	Small job definition. Max number of reducers allowed in small job.
mapred.fairscheduler.smalljob.max.inputsize	10737418240	Small job definition. Max input size in bytes allowed for a small job. Default is 10GB.
mapred.fairscheduler.smalljob.max.reducer.inputsize	1073741824	Small job definition. Max estimated input size for a reducer allowed in small job. Default is 1GB per reducer.
mapred.cluster.ephemeral.tasks.memory.limit.mb	200	Small job definition. Max memory in mbytes reserved for an ephermal slot. Default is 200mb. This value must be same on JobTracker and TaskTracker nodes.

MapReduce jobs that appear to fit the small job definition but are in fact larger than anticipated are killed and re-queued for normal execution.

Secured TaskTracker

You can control which users are able to submit jobs to the TaskTracker. By default, the TaskTracker is secured; All TaskTracker nodes should have the same user and group databases, and only users who are present on all TaskTracker nodes (same user ID on all nodes) can submit jobs. You can disallow certain users (including `root` or other superusers) from submitting jobs, or remove user restrictions from the TaskTracker completely.
/opt/mapr/hadoop/hadoop-0.20.2/conf/mapred-site.xml

To disallow `root`:

1. Edit `mapred-site.xml` and set `mapred.tasktracker.task-controller.config.overwrite = false` on all TaskTracker nodes.
2. Edit `taskcontroller.cfg` and set `min.user.id=0` on all TaskTracker nodes.
3. Restart all TaskTrackers.

To disallow all superusers:

1. Edit `mapred-site.xml` and set `mapred.tasktracker.task-controller.config.overwrite = false` on all TaskTracker nodes.

2. Edit `taskcontroller.cfg` and set `min.user.id=1000` on all TaskTracker nodes.
3. Restart all TaskTrackers.

To disallow specific users:

1. Edit `mapred-site.xml` and set `mapred.tasktracker.task-controller.config.overwrite = false` on all TaskTracker nodes.
2. Edit `taskcontroller.cfg` and add the parameter `banned.users` on all TaskTracker nodes, setting it to a comma-separated list of usernames. Example:

   ```
   banned.users=foo,bar
   ```

3. Restart all TaskTrackers.

To remove all user restrictions, and run all jobs as `root`:

1. Edit `mapred-site.xml` and set `mapred.task.tracker.task.controller = org.apache.hadoop.mapred.DefaultTaskController` on all TaskTracker nodes.
2. Restart all TaskTrackers.

> When you make the above setting, the tasks generated by all jobs submitted by any user will run with the same privileges as the TaskTracker (`root` privileges), and will have the ability to overwrite, delete, or damage data regardless of ownership or permissions.

Standalone Operation

You can run MapReduce jobs locally, using the local filesystem, by setting `mapred.job.tracker=local` in `mapred-site.xml`. With that parameter set, you can use the local filesystem for both input and output, use MapR-FS for input and output to the local filesystem, or use the local filesystem for input and output to MapR-FS.

Examples

Input and output on local filesystem

```
./bin/hadoop jar hadoop-0.20.2-dev-examples.jar grep -Dmapred.job.tracker=local file:///o
pt/mapr/hadoop/hadoop-0.20.2/input file:///opt/mapr/hadoop/hadoop-0.20.2/output
'dfs[a-z.]+'
```

Input from MapR-FS

```
./bin/hadoop jar hadoop-0.20.2-dev-examples.jar grep -Dmapred.job.tracker=local input
file:///opt/mapr/hadoop/hadoop-0.20.2/output 'dfs[a-z.]+'
```

Output to MapR-FS

```
./bin/hadoop jar hadoop-0.20.2-dev-examples.jar grep -Dmapred.job.tracker=local file:///o
pt/mapr/hadoop/hadoop-0.20.2/input output 'dfs[a-z.]+'
```

Tuning MapReduce

Greenplum HD EE automatically tunes the cluster for most purposes. A service called the *warden* determines machine resources on nodes configured to run the TaskTracker service, and sets MapReduce parameters accordingly.

On nodes with multiple CPUs, Greenplum HD EE uses taskset to reserve CPUs for Greenplum HD EE services:

- On nodes with five to eight CPUs, CPU 0 is reserved for Greenplum HD EE services
- On nodes with nine or more CPUs, CPU 0 and CPU 1 are reserved for Greenplum HD EE services

In certain circumstances, you might wish to manually tune Greenplum HD EE to provide higher performance. For example, when running a job consisting of unusually large tasks, it is helpful to reduce the number of slots on each TaskTracker and adjust the Java heap size. The following sections provide MapReduce tuning tips. If you change any settings in mapred-site.xml, restart the

TaskTracker.

Memory Settings

Memory for Greenplum HD EE Services

The memory allocated to each Greenplum HD EE service is specified in the `/opt/mapr/conf/warden.conf` file, which Greenplum HD EE automatically configures based on the physical memory available on the node. For example, you can adjust the minimum and maximum memory used for the TaskTracker, as well as the percentage of the heap that the TaskTracker tries to use, by setting the appropriate `percent`, `max`, and `min` parameters in the `warden.conf` file:

```
...
service.command.tt.heapsize.percent=2
service.command.tt.heapsize.max=325
service.command.tt.heapsize.min=64
...
```

The percentages of memory used by the services need not add up to 100; in fact, you can use less than the full heap by setting the `heapsize.percent` parameters for all services to add up to less than 100% of the heap size. In general, you should not need to adjust the memory settings for individual services, unless you see specific memory-related problems occurring.

MapReduce Memory

The memory allocated for MapReduce tasks normally equals the total system memory minus the total memory allocated for Greenplum HD EE services. If necessary, you can use the parameter mapreduce.tasktracker.reserved.physicalmemory.mb to set the maximum physical memory reserved by MapReduce tasks, or you can set it to `-1` to disable physical memory accounting and task management.

If the node runs out of memory, MapReduce tasks are killed by the OOM-killer to free memory. You can use `mapred.child.oom_adj` (copy from `mapred-default.xml` to adjust the `oom_adj` parameter for MapReduce tasks. The possible values of `oom_adj` range from -17 to +15. The higher the score, more likely the associated process is to be killed by OOM-killer.

Job Configuration

Map Tasks

Map tasks use memory mainly in two ways:

- The MapReduce framework uses an intermediate buffer to hold serialized (key, value) pairs.
- The application consumes memory to run the map function.

MapReduce framework memory is controlled by `io.sort.mb`. If `io.sort.mb` is less than the data emitted from the mapper, the task ends up spilling data to disk. If `io.sort.mb` is too large, the task can run out of memory or waste allocated memory. By default `io.sort.mb` is 100mb. It should be approximately 1.25 times the number of data bytes emitted from mapper. If you cannot resolve memory problems by adjusting `io.sort.mb`, then try to re-write the application to use less memory in its map function.

Reduce Tasks

If tasks fail because of an Out of Heap Space error, increase the heap space (the `-Xmx` option in `mapred.reduce.child.java.opts`) to give more memory to the tasks. If map tasks are failing, you can also try reducing `io.sort.mb`.
(see mapred.map.child.java.opts in mapred-site.xml)

TaskTracker Configuration

Greenplum HD EE sets up map and reduce slots on each TaskTracker node using formulas based on the number of CPUs present on the node. The default formulas are stored in the following parameters in `mapred-site.xml`:

- `mapred.tasktracker.map.tasks.maximum`: (CPUS > 2) ? (CPUS * 0.75) : 1 (At least one Map slot, up to 0.75 times the number of CPUs)
- `mapred.tasktracker.reduce.tasks.maximum`: (CPUS > 2) ? (CPUS * 0.50) : 1 (At least one Map slot, up to 0.50 times the number of CPUs)

You can adjust the maximum number of map and reduce slots by editing the formula used in `mapred.tasktracker.map.tasks.maximum` and `mapred.tasktracker.reduce.tasks.maximum`. The following variables are used in the formulas:

- `CPUS` - number of CPUs present on the node
- `DISKS` - number of disks present on the node
- `MEM` - memory reserved for MapReduce tasks

Ideally, the number of map and reduce slots should be decided based on the needs of the application. Map slots should be based on how many map tasks can fit in memory, and reduce slots should be based on the number of CPUs. If each task in a MapReduce job takes 3 GB, and each node has 9GB reserved for MapReduce tasks, then the total number of map slots should be 3. The amount of data each map task must process also affects how many map slots should be configured. If each map task processes 256 MB (the default chunksize in Greenplum HD EE), then each map task should have 800 MB of memory. If there are 4 GB reserved for map tasks, then the number of map slots should be 4000MB/800MB, or 5 slots.

Greenplum HD EE allows the JobTracker to over-schedule tasks on TaskTracker nodes in advance of the availability of slots, creating a pipeline. This optimization allows TaskTracker to launch each map task as soon as the previous running map task finishes. The number of tasks to over-schedule should be about 25-50% of total number of map slots. You can adjust this number with the parameter mapreduce.tasktracker.prefetch.maptasks.

Working with Data

This section contains information about working with data:

- Copying Data from Apache Hadoop - using `distcp` to copy data to Greenplum HD EE from an Apache cluster
- Data Protection - how to protect data from corruption or deletion
- Direct Access NFS - how to mount the cluster via NFS
- Volumes - using volumes to manage data
    - Mirrors - local or remote copies of volumes
    - Schedules - scheduling for snapshots and mirrors
    - Snapshots - point-in-time images of volumes

Copying Data from Apache Hadoop

To enable data copying from an Apache Hadoop cluster to a Greenplum HD EE cluster using `distcp`, perform the following steps from a Greenplum HD EE client or node (any computer that has either `mapr-core` or `mapr-client` installed). For more information about setting up a Greenplum HD EE client, see Setting Up the Client.

1. Log in as the `root` user (or use `sudo` for the following commands).
2. Create the directory `/tmp/maprfs-client/` on the Apache Hadoop JobClient node.
3. Copy the following files from a Greenplum HD EE client or any Greenplum HD EE node to the `/tmp/maprfs-client/` directory:
    - /opt/mapr/hadoop/hadoop-0.20.2/lib/maprfs-0.1.jar,
    - /opt/mapr/hadoop/hadoop-0.20.2/lib/zookeeper-3.3.2.jar
    - /opt/mapr/hadoop/hadoop-0.20.2/lib/native/Linux-amd64-64/libMapRClient.so
4. Install the files in the correct places on the Apache Hadoop JobClient node:

```
cp /tmp/maprfs-client/maprfs-0.1.jar $HADOOP_HOME/lib/.
cp /tmp/maprfs-client/zookeeper-3.3.2.jar $HADOOP_HOME/lib/.
cp /tmp/maprfs-client/libMapRClient.so $HADOOP_HOME/lib/native/Linux-amd64-64/libMa
pRClient.so
```

    - If you are on a 32-bit client, use `Linux-i386-32` in place of `Linux-amd64-64` above.
5. If the JobTracker is a different node from the JobClient node, copy and install the files to the JobTracker node as well using the above steps.
6. On the JobTracker node, set `fs.maprfs.impl=com.mapr.fs.MapRFileSystem` in `$HADOOP_HOME/conf/core-site.xml`.
7. Restart the JobTracker.

You can now copy data to the Greenplum HD EE cluster by running `distcp` on the JobClient node of the Apache Hadoop cluster. In the following example, `$INPUTDIR` is the HDFS path to the source data; `$OUTPUTDIR` is the MapR-FS path to the target directory; and `<MapR CLDB IP>` is the IP address of the master CLDB node on the MapR cluster. Example:

```
./bin/hadoop distcp -Dfs.maprfs.impl=com.mapr.fs.MapRFileSystem -libjars
/tmp/maprfs-client/maprfs-0.1.jar,/tmp/maprfs-client/zookeeper-3.3.2.jar -files
/tmp/maprfs-client/libMapRClient.so $INPUTDIR maprfs://<MapR CLDB IP>:7222/$OUTPUTDIR
```

Data Protection

You can use Greenplum HD EE to protect your data from hardware failures, accidental overwrites, and natural disasters. Greenplum HD EE organizes data into volumes so that you can apply different data protection strategies to different types of data. The following scenarios describe a few common problems and how easily and effectively Greenplum HD EE protects your data from loss.

Scenario: Hardware Failure

Even with the most reliable hardware, growing cluster and datacenter sizes will make frequent hardware failures a real threat to business continuity. In a cluster with 10,000 disks on 1,000 nodes, it is reasonable to expect a disk failure more than once a day and a node failure every few days.

Solution: Topology and Replication Factor

Greenplum HD EE automatically replicates data and places the copies on different nodes to safeguard against data loss in the event of hardware failure. By default, Greenplum HD EE assumes that all nodes are in a single rack. You can provide Greenplum HD EE with information about the rack locations of all nodes by setting topology paths. Greenplum HD EE interprets each topology path as a separate rack, and attempts to replicate data onto different racks to provide continuity in case of a power failure affecting an entire rack. These replicas are maintained, copied, and made available seamlessly without user intervention.

To set up topology and replication:

1. In the Greenplum HD EEControl System, open the MapR-FS group and click **Nodes** to display the Nodes view.
2. Set up each rack with its own path. For each rack, perform the following steps:
    a. Click the checkboxes next to the nodes in the rack.
    b. Click the **Change Topology** button to display the Change Node Topology dialog.
    c. In the Change Node Topology dialog, type a path to represent the rack. For example, if the cluster name is `cluster1` and the nodes are in rack 14, type `/cluster1/rack14`.
3. When creating volumes, choose a **Replication Factor** of 3 or more to provide sufficient data redundancy.

Scenario: Accidental Overwrite

Even in a cluster with data replication, important data can be overwritten or deleted accidentally. If a data set is accidentally removed, the removal itself propagates across the replicas and the data is lost. Users or applications can corrupt data, and once the corruption spreads to the replicas the damage is permanent.

Solution: Snapshots

With Greenplum HD EE, you can create a point-in-time snapshot of a volume, allowing recovery from a known good data set. You can create a manual snapshot to enable recovery to a specific point in time, or schedule snapshots to occur regularly to maintain a recent recovery point. If data is lost, you can restore the data using the most recent snapshot (or any snapshot you choose). Snapshots do not add a performance penalty, because they do not involve additional data copying operations; a snapshot can be created almost instantly regardless of data size.

Example: Creating a Snapshot Manually

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Select the checkbox beside the name the volume, then click the **New Snapshot** button to display the Snapshot Name dialog.
3. Type a name for the new snapshot in the **Name...** field.
4. Click **OK** to create the snapshot.

Example: Scheduling Snapshots

This example schedules snapshots for a volume hourly and retains them for 24 hours.

To create a schedule:

1. In the Navigation pane, expand the **MapR-FS** group and click the Schedules view.
2. Click **New Schedule**.
3. In the **Schedule Name** field, type "Every Hour".
4. From the first dropdown menu in the Schedule Rules section, select **Hourly**.
5. In the **Retain For** field, specify 24 Hours.
6. Click **Save Schedule** to create the schedule.

To apply the schedule to the volume:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Display the Volume Properties dialog by clicking the volume name, or by selecting the checkbox beside the volume name then clicking the **Properties** button.
3. In the **Replication and Snapshot Scheduling** section, choose "Every Hour."
4. Click **Modify Volume** to apply the changes and close the dialog.

Scenario: Disaster Recovery

A severe natural disaster can cripple an entire datacenter, leading to permanent data loss unless a disaster plan is in place.

Solution: Mirroring to Another Cluster

Greenplum HD EE makes it easy to protect against loss of an entire datacenter by mirroring entire volumes to a different datacenter. A mirror is a full read-only copy of a volume that can be synced on a schedule to provide point-in-time recovery for critical data. If the volumes on the original cluster contain a large amount of data, you can store them on physical media using the volume dump create command and transport them to the mirror cluster. Otherwise, you can simply create mirror volumes that point to the volumes on the original cluster and copy the data over the network. The mirroring operation conserves bandwidth by transmitting only the deltas between the source and the mirror, and by compressing the data over the wire. In addition, Greenplum HD EE uses checksums and a latency-tolerant protocol to ensure success even on high-latency WANs. You can set up a cascade of mirrors to replicate data over a distance. For instance, you can mirror data from New York to London, then use lower-cost links to replicate the data from London to Paris and Rome.

To set up mirroring to another cluster:

1. Use the volume dump create command to create a full volume dump for each volume you want to mirror.
2. Transport the volume dump to the mirror cluster.
3. For each volume on the original cluster, set up a corresponding volume on the mirror cluster.
    a. Restore the volume using the volume dump restore command.
    b. In the Greenplum HD EE Control System, click Volumes under the MapR-FS group to display the Volumes view.
    c. Click the name of the volume to display the Volume Properties dialog.
    d. Set the **Volume Type** to Remote Mirror Volume.
    e. Set the **Source Volume Name** to the source volume name.
    f. Set the **Source Cluster Name** to the cluster where the source volume resides.
    g. In the **Replication and Mirror Scheduling** section, choose a schedule to determine how often the mirror will sync.

To recover volumes from mirrors:

1. Use the volume dump create command to create a full volume dump for each mirror volume you want to restore. Example:
   ```
   maprcli volume create -e statefile1 -dumpfile fulldump1 -name volume@cluster
   ```
2. Transport the volume dump to the rebuilt cluster.
3. For each volume on the mirror cluster, set up a corresponding volume on the rebuilt cluster.
    a. Restore the volume using the volume dump restore command. Example:
       ```
       maprcli volume dump restore -name volume@cluster -dumpfile fulldump1
       ```
    b. Copy the files to a standard (non-mirror) volume.

Provisioning Applications

Provisioning a new application involves meeting the business goals of performance, continuity, and security while providing necessary resources to a client, department, or project. You'll want to know how much disk space is needed, and what the priorities are in terms of performance, reliability. Once you have gathered all the requirements, you will create a volume to manage the application data. A volume provides convenient control over data placement, performance, protection, and policy for an entire data set.

Make sure the cluster has the storage and processing capacity for the application. You'll need to take into account the starting and predicted size of the data, the performance and protection requirements, and the memory required to run all the processes required on each node. Here is the information to gather before beginning:

Access	How often will the data be read and written? What is the ratio of reads to writes?
Continuity	What is the desired recovery point objective (RPO)? What is the desired recovery time objective (RTO)?
Performance	Is the data static, or will it change frequently? Is the goal data storage or data processing?
Size	How much data capacity is required to start? What is the predicted growth of the data?

The considerations in the above table will determine the best way to set up a volume for the application.

About Volumes

Volumes provide a number of ways to help you meet the performance, access, and continuity goals of an application, while managing application data size:

- Mirroring - create read-only copies of the data for highly accessed data or multi-datacenter access
- Permissions - allow users and groups to perform specific actions on a volume
- Quotas - monitor and manage the data size by project, department, or user

- Replication - maintain multiple synchronized copies of data for high availability and failure protection
- Snapshots - create a real-time point-in-time data image to enable rollback
- Topology - place data on a high-performance rack or limit data to a particular set of machines

See Volumes.

Mirroring

Mirroring means creating *mirror volumes*, full physical read-only copies of normal volumes for fault tolerance and high performance. When you create a mirror volume, you specify a source volume from which to copy data, and you can also specify a schedule to automate re-synchronization of the data to keep the mirror up-to-date. After a mirror is initially copied, the synchronization process saves bandwidth and reads on the source volume by transferring only the deltas needed to bring the mirror volume to the same state as its source volume. A mirror volume need not be on the same cluster as its source volume; Greenplum HD EE can sync data on another cluster (as long as it is reachable over the network). When creating multiple mirrors, you can further reduce the mirroring bandwidth overhead by daisy-chaining the mirrors. That is, set the source volume of the first mirror to the original volume, the source volume of the second mirror to the first mirror, and so on. Each mirror is a full copy of the volume, so remember to take the number of mirrors into account when planning application data size. See Mirrors.

Permissions

Greenplum HD EE provides fine-grained control over which users and groups can perform specific tasks on volumes and clusters. When you create a volume, keep in mind which users or groups should have these types of access to the volume. You may want to create a specific group to associate with a project or department, then add users to the group so that you can apply permissions to them all at the same time. See Managing Permissions.

Quotas

You can use quotas to limit the amount of disk space an application can use. There are two types of quotas:

- User/Group quotas limit the amount of disk space available to a user or group
- Volume quotas limit the amount of disk space available to a volume

When the data owned by a user, group, or volume exceeds the quota, Greenplum HD EE prevents further writes until either the data size falls below the quota again, or the quota is raised to accommodate the data.

Volumes, users, and groups can also be assigned *advisory quotas*. An advisory quota does not limit the disk space available, but raises an alarm and sends a notification when the space used exceeds a certain point. When you set a quota, you can use a slightly lower advisory quota as a warning that the data is about to exceed the quota, preventing further writes.

Remember that volume quotas do not take into account disk space used by sub-volumes (because volume paths are logical, not physical).

You can set a User/Group quota to manage and track the disk space used by an accounting entity (a department, project, or application):

- Create a group to represent the accounting entity.
- Create one or more volumes and use the group as the Accounting Entity for each.
- Set a User/Group quota for the group.
- Add the appropriate users to the group.

When a user writes to one of the volumes associated with the group, any data written counts against the group's quota. Any writes to volumes not associated with the group are not counted toward the group's quota. See Managing Quotas.

Replication

When you create a volume, you can choose a replication factor to safeguard important data. Greenplum HD EE manages the replication automatically, raising an alarm and notification if replication falls below the minimum level you have set. A replicate of a volume is a full copy of the volume; remember to take that into account when planning application data size.

Snapshots

A snapshot is an instant image of a volume at a particular point in time. Snapshots take no time to create, because they only record changes to data over time rather than the data itself. You can manually create a snapshot to enable rollback to a particular known data state, or schedule periodic automatic snapshots to ensure a specific recovery point objective (RPO). You can use snapshots and mirrors to achieve a near-zero recovery time objective (RTO). Snapshots store only the deltas between a volume's current state and its state when the snapshot is taken. Initially, snapshots take no space on disk, but they can grow arbitrarily as a volume's data changes. When planning application data size, take into account how much the data is likely to change, and how often snapshots will be taken. See Snapshots.

Topology

You can restrict a volume to a particular rack by setting its physical topology attribute. This is useful for placing an application's data on a high-performance rack (for critical applications) or a low-performance rack (to keep it out of the way of critical applications). See Setting Volume Topology.

Scenarios

Here are a few ways to configure the application volume based on different types of data. If the application requires more than one type of data, you can set up multiple volumes.

Data Type	Strategy
Important Data	High replication factor Frequent snapshots to minimize RPO and RTO Mirroring in a remote cluster
Highly Accessed Data	High replication factor Mirroring for high-performance reads Topology: data placement on high-performance machines
Scratch data	No snapshots, mirrors, or replication Topology: data placement on low-performance machines
Static data	Mirroring and replication set by performance and availability requirements One snapshot (to protect against accidental changes) Volume set to read-only

The following documents provide examples of different ways to provision an application to meet business goals:

- Provisioning for Capacity
- Provisioning for Performance

Setting Up the Application

Once you know the course of action to take based on the application's data and performance needs, you can use the Greenplum HD EE Control System to set up the application.

- Creating a Group and a Volume
- Setting Up Mirroring
- Setting Up Snapshots
- Setting Up User or Group Quotas

Creating a Group and a Volume

Create a group and a volume for the application. If you already have a snapshot schedule prepared, you can apply it to the volume at creation time. Otherwise, use the procedure in Setting Up Snapshots below, after you have created the volume.

Setting Up Mirroring

- If you want the mirror to sync automatically, use the procedure in Creating a Schedule to create a schedule.
- Use the procedure in Creating a Volume to create a mirror volume. Make sure to set the following fields:
    - Volume Type - Mirror Volume
    - Source Volume - the volume you created for the application
    - Responsible Group/User - in most cases, the same as for the source volume

Setting Up Snapshots

To set up automatic snapshots for the volume, use the procedure in Scheduling a Snapshot.

Provisioning for Capacity

You can easily provision a volume for maximum data storage capacity by setting a low replication factor, setting hard and advisory quotas, and tracking storage use by users, groups, and volumes. You can also set permissions to limit who can write data to the volume.

The replication factor determines how many complete copies of a volume are stored in the cluster. The actual storage requirement for a volume is the volume size multiplied by its replication factor. To maximize storage capacity, set the replication factor on the volume to 1 at the time you create the volume.

Volume quotas and user or group quotas limit the amount of data that can be written by a user or group, or the maximum size of a specific volume. When the data size exceeds the advisory quota, Greenplum HD EE raises an alarm and notification but does not prevent additional data writes. Once the data exceeds the hard quota, no further writes are allowed for the volume, user, or group. The advisory quota is generally somewhat lower than the hard quota, to provide advance warning that the data is in danger of exceeding the hard quota. For a high-capacity volume, the volume quotas should be as large as possible. You can use the advisory quota to warn you when the volume is approaching its maximum size.

To use the volume capacity wisely, you can limit write access to a particular user or group. Create a new user or group on all nodes in the cluster.

In this scenario, storage capacity takes precedence over high performance and data recovery; to maximize data storage, there will be no snapshots or mirrors set up in the cluster. A low replication factor means that the data is less effectively protected against loss in the event that disks or nodes fail. Because of these tradeoffs, this strategy is most suitable for risk-tolerant large data sets, and should not be used for data with stringent protection, recovery, or performance requirements.

To create a high-capacity volume:

1. Set up a user or group that will be responsible for the volume. For more information, see Users & Groups.
2. In the Greenplum HD EE Control System, open the MapR-FS group and click **Volumes** to display the Volumes view.
3. Click the **New Volume** button to display the New Volume dialog.
4. In the **Volume Setup** pane, set the volume name and mount path.
5. In the **Usage Tracking** pane:
   a. In the **Group/User** section, select **User** or **Group** and enter the user or group responsible for the volume.
   b. In the **Quotas** section, check **Volume Quota** and enter the maximum capacity of the volume, based on the storage capacity of your cluster. Example: 1 TB
   c. Check **Volume Advisory Quota** and enter a lower number than the volume quota, to serve as advance warning when the data approaches the hard quota. Example: 900 GB
6. In the **Replication & Snapshot Scheduling** pane:
   a. Set **Replication** to `1`.
   b. Do not select a snapshot schedule.
7. Click OK to create the volume.
8. Set the volume permissions on the volume via NFS or using `hadoop fs`. You can limit writes to root and the responsible user or group.

See Volumes for more information.

Provisioning for Performance

You can provision a high-performance volume by creating multiple mirrors of the data and defining volume topology to control data placement: store the data on your fastest servers (for example, servers that use SSDs instead of hard disks).

When you create mirrors of a volume, make sure your application load-balances reads across the mirrors to increase performance. Each mirror is an actual volume, so you can control data placement and replication on each mirror independently. The most efficient way to create multiple mirrors is to cascade them rather than creating all the mirrors from the same source volume. Create the first mirror from the original volume, then create the second mirror using the first mirror as the source volume, and so on. You can mirror the volume within the same cluster or to another cluster, possibly in a different datacenter.

You can set node topology paths to specify the physical locations of nodes in the cluster, and volume topology paths to limit volumes to specific nodes or racks.

To set node topology:

Use the following steps to create a rack path representing the high-performance nodes in your cluster.

1. In the Greenplum HD EE Control System, open the MapR-FS group and click **Nodes** to display the Nodes view.
2. Click the checkboxes next to the high-performance nodes.
3. Click the **Change Topology** button to display the Change Node Topology dialog.
4. In the Change Node Topology dialog, type a path to represent the high-performance rack. For example, if the cluster name is `cluster1` and the high-performance nodes make up rack 14, type `/cluster1/rack14`.

To set up the source volume:

1. In the Greenplum HD EE Control System, open the MapR-FS group and click **Volumes** to display the Volumes view.
2. Click the **New Volume** button to display the New Volume dialog.
3. In the **Volume Setup** pane, set the volume name and mount path normally.
4. Set the **Topology** to limit the volume to the high-performance rack. Example: `/default/rack14`

To Set Up the First Mirror

1. In the Greenplum HD EE Control System, open the MapR-FS group and click **Volumes** to display the Volumes view.
2. Click the **New Volume** button to display the New Volume dialog.
3. In the **Volume Setup** pane, set the volume name and mount path normally.
4. Choose **Local Mirror Volume**.
5. Set the **Source Volume Name** to the original volume name. Example: `original-volume`
6. Set the **Topology** to a different rack from the source volume.

To Set Up Subsequent Mirrors

1. In the Greenplum HD EE Control System, open the MapR-FS group and click **Volumes** to display the Volumes view.
2. Click the **New Volume** button to display the New Volume dialog.
3. In the **Volume Setup** pane, set the volume name and mount path normally.
4. Choose **Local Mirror Volume**.
5. Set the **Source Volume Name** to the previous mirror volume name. Example: `mirror1`
6. Set the **Topology** to a different rack from the source volume and the other mirror.

See Volumes for more information.

Managing the Cluster

Greenplum HD EE provides a number of tools for managing the cluster. This section describes the following topics:

- Nodes - Viewing, installing, configuring, and moving nodes
- Monitoring - Getting timely information about the cluster

Monitoring

This section provides information about monitoring the cluster:

- Alarms and Notifications
- Monitoring Tools

Alarms and Notifications

Greenplum HD EE raises alarms and sends notifications to alert you to information about a cluster:

- Cluster health, including disk failures
- Volumes that are under-replicated or over quota
- Services not running

You can see any currently raised alarms in the Alarms view of the Greenplum HD EE Control System, or using the alarm list command. For a list of all alarms, see Troubleshooting Alarms.

To view cluster alarms using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **Cluster** group and click the Dashboard view.
2. All alarms for the cluster and its nodes and volumes are displayed in the Alarms pane.

To view node alarms using the Greenplum HD EE Control System:

- In the Navigation pane, expand the **Alarms** group and click the Node Alarms view.

You can also view node alarms in the Node Properties view, the NFS Alarm Status view, and the Alarms pane of the Dashboard view.

To view volume alarms using the Greenplum HD EE Control System:

- In the Navigation pane, expand the **Alarms** group and click the Volume Alarms view.

You can also view node alarms in the Alarms pane of the Dashboard view.

Notifications

When an alarm is raised, Greenplum HD EE can send an email notification to either or both of the following addresses:

- The owner of the cluster, node, volume, or entity for which the alarm was raised (standard notification)
- A custom email address for the named alarm.

You can set up alarm notifications using the alarm config save command or from the Alarms view in the Greenplum HD EE Control System.

To set up alarm notifications using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **Alarms** group and click the Alarm Notifications view.
2. Display the **Configure Alarm Subscriptions** dialog by clicking **Alarm Notifications**.
3. For each **Alarm**:

    - To send notifications to the owner of the cluster, node, volume, or entity: select the **Standard Notification** check box.
    - To send notifications to an additional email address, type an email address in the **Additional Email Address** field.
4. Click **Save** to save the configuration changes.

Monitoring Tools

Greenplum HD EE works with the following third-party monitoring tools:

- Ganglia
- Nagios

Service Metrics

Greenplum HD EE services produce metrics that can be written to an output file or consumed by Ganglia. The file metrics output is directed by the `hadoop-metrics.properties` files.
By default, the CLDB and FileServer metrics are sent via unicast to the Ganglia gmon server running on localhost. To send the metrics directly to a Gmeta server, change the `cldb.servers` property to the hostname of the Gmeta server. To send the metrics to a multicast channel, change the `cldb.servers` property to the IP address of the multicast channel.

To configure metrics for a service:

1. Edit the appropriate `hadoop-metrics.properties` file on all CLDB nodes, depending on the service:
    - For Greenplum HD EE-specific services, edit `/opt/mapr/conf/hadoop-metrics.properties`
    - For standard Hadoop services, edit `/opt/mapr/hadoop/hadoop-<version>/conf/hadoop-metrics.properties`
2. In the sections specific to the service:
    - Un-comment the lines pertaining to the context to which you wish the service to send metrics.
    - Comment out the lines pertaining to other contexts.
3. Restart the service.

To enable metrics:

1. As root (or using sudo), run the following commands:

```
maprcli config save -values '{"cldb.ganglia.cldb.metrics":"1"}'
maprcli config save -values '{"cldb.ganglia.fileserver.metrics":"1"}'
```

To disable metrics:

1. As root (or using sudo), run the following commands:

```
maprcli config save -values '{"cldb.ganglia.cldb.metrics":"0"}'
maprcli config save -values '{"cldb.ganglia.fileserver.metrics":"0"}'
```

Example

In the following example, CLDB service metrics will be sent to the Ganglia context:

```
#CLDB metrics config - Pick one out of null,file or ganglia.
#Uncomment all properties in null, file or ganglia context, to send cldb metrics to that
context

# Configuration of the "cldb" context for null
#cldb.class=org.apache.hadoop.metrics.spi.NullContextWithUpdateThread
#cldb.period=10

# Configuration of the "cldb" context for file
#cldb.class=org.apache.hadoop.metrics.file.FileContext
#cldb.period=60
#cldb.fileName=/tmp/cldbmetrics.log

# Configuration of the "cldb" context for ganglia
cldb.class=com.mapr.fs.cldb.counters.MapRGangliaContext31
cldb.period=10
cldb.servers=localhost:8649
cldb.spoof=1
```

Nodes

This section provides information about managing nodes in the cluster:

- Viewing a List of Nodes - displaying all the nodes recognized by the Greenplum HD EE cluster
- Adding a Node - installing a new node on the cluster (requires `fc` or `a` permission)
- Managing Services - starting or stopping services on a node (requires `ss`, `fc`, or `a` permission)
- Reformatting a Node - reformatting a node's disks
- Removing a Node - removing a node temporarily for maintenance (requires `fc` or `a` permission)
- Decommissioning a Node - permanently uninstalling a node (requires `fc` or `a` permission)
- Reconfiguring a Node - installing, upgrading, or removing hardware or software, or changing roles

Viewing a List of Nodes

You can view all nodes using the node list command, or view them in the Greenplum HD EE Control System using the following procedure.

To view all nodes using the Greenplum HD EE Control System:

- In the Navigation pane, expand the **Cluster** group and click the Nodes view.

Adding a Node

To Add Nodes to a Cluster

1. PREPARE all nodes, making sure they meet the hardware, software, and configuration requirements.
2. PLAN which services to run on the new nodes.
3. INSTALL Greenplum HD EE Software:
   - On each new node, INSTALL the planned Greenplum HD EE services.
   - On all new nodes, RUN configure.sh.
   - On all new nodes, FORMAT disks for use by Greenplum HD EE.
   - On any previously used Greenplum HD EE cluster node, use the script `zkdatacleaner.sh` to clean up old ZooKeeper data:

```
/opt/mapr/server/zkdatacleaner.sh
```

If you have made any changes to configuration files such as `warden.conf` or `mapred-site.xml`, copy these configuration changes from another node in the cluster.
Start each node:

- On any new nodes that have ZooKeeper installed, start it:

  ```
  /etc/init.d/mapr-zookeeper start
  ```

- On all new nodes, start the warden:

```
/etc/init.d/mapr-warden start
```

If any of the new nodes are CLDB or ZooKeeper nodes (or both):

- RUN `configure.sh` on all new and existing nodes in the cluster, specifying all CLDB and ZooKeeper nodes.

SET UP node topology for the new nodes.
On any new nodes running NFS, SET UP NFS for HA.

Managing Services

You can manage node services using the node services command, or in the Greenplum HD EE Control System using the following procedure.

To manage node services using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **Cluster** group and click the Nodes view.
2. Select the checkbox beside the node or nodes you wish to remove.
3. Click the **Manage Services** button to display the Manage Node Services dialog.
4. For each service you wish to start or stop, select the appropriate option from the corresponding drop-down menu.
5. Click **Change Node** to start and stop the services according to your selections.

You can also display the Manage Node Services dialog by clicking **Manage Services** in the Node Properties view.

Reformatting a Node

1. Change to the root user (or use sudo for the following commands).
2. Stop the Warden:
   `/etc/init.d/mapr-warden stop`
3. Remove the `disktab` file:
   `rm /opt/mapr/conf/disktab`
4. Create a text file `/tmp/disks.txt` that lists all the disks and partitions to format for use by Greenplum HD EE. See Setting Up Disks for Greenplum HD EE.
5. Use `disksetup` to re-format the disks:
   `disksetup -F /tmp/disks.txt`
6. Start the Warden:
   `/etc/init.d/mapr-warden start`

Removing a Node

You can remove a node using the node remove command, or in the Greenplum HD EE Control System using the following procedure. Removing a node detaches the node from the cluster, but does not remove the Greenplum HD EE software from the cluster.

To remove a node using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **Cluster** group and click the NFS Nodes view.
2. Select the checkbox beside the node or nodes you wish to remove.
3. Click **Manage Services** and stop all services on the node.
4. Wait 5 minutes. The Remove button becomes active.
5. Click the **Remove** button to display the Remove Node dialog.
6. Click **Remove Node** to remove the node.

⚠ If you are using Ganglia, restart all gmeta and gmon daemons in the cluster. See Ganglia.

You can also remove a node by clicking **Remove Node** in the Node Properties view.

Decommissioning a Node

Use the following procedures to remove a node and uninstall the Greenplum HD EE software. This procedure detaches the node from the cluster and removes the Greenplum HD EE packages, log files, and configuration files, but does not format the disks.

⛔ **Before Decommissioning a Node**
Make sure any data on the node is replicated and any needed services are running elsewhere. For example, if decommissioning the node would result in too few instances of the CLDB, start CLDB on another node beforehand; if you are decommissioning a ZooKeeper node, make sure you have enough ZooKeeper instances to meet a quorum after the node is removed. See Planning the Deployment for recommendations.

To decommission a node permanently:

1. Change to the root user (or use sudo for the following commands).
2. Stop the Warden:
   `/etc/init.d/mapr-warden stop`
3. Remove the node (see Removing a Node).
4. If Pig is installed, remove it:
   - `erase mapr-pig-internal` (Red Hat or CentOS)
5. If Hive is installed, remove it:
   - `erase mapr-hive-internal` (Red Hat or CentOS)
6. If HBase (Master or RegionServer) is installed, remove it:
   - `erase mapr-hbase-internal` (Red Hat or CentOS)
7. Remove the package mapr-core:
   - `erase mapr-core` (Red Hat or CentOS)
8. If ZooKeeper is installed on the node, stop it:
   `/etc/init.d/mapr-zookeeper stop`
9. If ZooKeeper is installed, remove it:
   - `erase mapr-zk-internal` (Red Hat or CentOS)
10. If the node you have decommissioned is a CLDB node or a ZooKeeper node, then run `configure.sh` on all other nodes in the cluster (see Configuring a Node).

⚠ If you are using Ganglia, restart all gmeta and gmon daemons in the cluster. See Ganglia.

Reconfiguring a Node

You can add, upgrade, or remove services on a node to perform a manual software upgrade or to change the roles a node serves. There are four steps to this procedure:

- Stopping the Node
- Formatting the Disks (optional)
- Installing or Removing Software or Hardware
- Configuring the Node
- Starting the Node

This procedure is designed to make changes to existing Greenplum HD EE software on a machine that has already been set up as a Greenplum HD EE cluster node. If you need to install software for the first time on a machine to create a new node, please see Adding a Node instead.

Stopping a Node

1. Change to the root user (or use sudo for the following commands).
2. Stop the Warden:
   `/etc/init.d/mapr-warden stop`
3. If ZooKeeper is installed on the node, stop it:
   `/etc/init.d/mapr-zookeeper stop`

Installing or Removing Software or Hardware

Before installing or removing software or hardware, stop the node using the procedure described in Stopping the Node.

Once the node is stopped, you can add, upgrade or remove software or hardware.

To add or remove individual Greenplum HD EE packages, use the standard package management commands for your Linux distribution:

For information about the packages to install, see Planning the Deployment.

After installing or removing software or hardware, follow the procedures in Configuring the Node and Starting the Node.

After you install new services on a node, you can start them in two ways:

- Use the Greenplum HD EE Control System, the API, or the command-line interface to start the services individually
- Restart the warden to stop and start all services on the node
  If you start the services individually, the node's memory will not be reconfigured to account for the newly installed services. This can cause memory paging, slowing or stopping the node. However, stopping and restarting the warden can take the node out of service.

Setting Up a Node

Formatting the Disks

> The script `disksetup` removes all data from the specified disks. Make sure you specify the disks correctly, and that any data you wish to keep has been backed up elsewhere. Before following this procedure, make sure you have backed up any data you wish to keep.

1. Change to the `root` user (or use `sudo` for the following command).
2. Run `disksetup`, specifying the disk list file.
   Example:

```
/opt/mapr/server/disksetup -F /tmp/disks.txt
```

Configuring the Node

Run the script `configure.sh` to create `/opt/mapr/conf/mapr-clusters.conf` and update the corresponding `*.conf` and `*.xml` files. Before performing this step, make sure you have a list of the hostnames of the CLDB and ZooKeeper nodes. Optionally, you can specify the ports for the CLDB and ZooKeeper nodes as well. If you do not specify them, the default ports are:

- CLDB – 7222
- ZooKeeper – 5181

The script `configure.sh` takes an optional cluster name and log file, and comma-separated lists of CLDB and ZooKeeper host names or IP addresses (and optionally ports), using the following syntax:

```
/opt/mapr/server/configure.sh -C <host>[:<port>][,<host>[:<port>]...] -Z
<host>[:<port>][,<host>[:<port>]...] [-L <logfile>][-N <cluster name>]
```

Example:

```
/opt/mapr/server/configure.sh -C r1n1.sj.us:7222,r3n1.sj.us:7222,r5n1.sj.us:7222 -Z
r1n1.sj.us:5181,r2n1.sj.us:5181,r3n1.sj.us:5181,r4n1.sj.us:5181,r1n1.sj.us5:5181 -N
MyCluster
```

If you have not chosen a cluster name, you can run `configure.sh` again later to rename the cluster.

Starting the Node

1. If ZooKeeper is installed on the node, start it:
   `/etc/init.d/mapr-zookeeper start`
2. Start the Warden:
   `/etc/init.d/mapr-warden start`

Adding Roles

To add roles to an existing node:

1. Install the packages corresponding to the new roles
2. Run `configure.sh` with a list of the CLDB nodes and ZooKeeper nodes in the cluster.

The warden picks up the new configuration and automatically starts the new services.

Memory Overcommit

There are two important memory management settings related to overcommitting memory:

- overcommit_memory - determines the strategy for overcommitting system memory (default: `0`)

- overcommit_ratio - determines how extensively memory can be overcommitted (default: `50`)

For more information, see the Linux kernel documentation about Overcommit Accounting.

In most cases, you should make sure the node has twice as much swap space as RAM and use an `overcommit_memory` setting of `0` to allow memory to be overcommitted to reduce swap usage while rejecting spurious or excessive overcommits. However, if the node does not have any swap space, you should set `overcommit_memory` to `1`. If you have less than twice as much swap space as RAM, you can set `overcommit_memory` to mode `2` and increase the `overcommit_ratio` to `100`.

To configure memory management on a node:

1. Use the `free` command to determine whether you have swap space on the node. Look for a line that starts with `Swap:`. Example:

```
$ free
             total       used       free     shared    buffers     cached
Mem:       2503308    2405524      97784          0      18192     575720
-/+ buffers/cache:    1811612     691696
Swap:      5712888     974240    4738648
```

2. If possible, ensure that you have at least twice as much swap space as physical RAM.
3. Set `overcommit_memory` according to whether there is swap space:
   - If the node has swap space, type `sysctl -w vm.overcommit_memory=0`
   - If the node does not have swap space, type `sysctl -w vm.overcommit_memory=1`

If you have a compelling reason to use `vm.overcommit_memory=2`, you should set `overcommit_ratio` to `100` by typing: `sysctl -w vm.overcommit_ratio=100`

Node Topology

Topology tells Greenplum HD EE about the locations of nodes and racks in the cluster. Topology is important, because it determines where Greenplum HD EE places replicated copies of data. If you define the cluster topology properly, Greenplum HD EE scatters replication on separate racks so that your data remains available in the event an entire rack fails. Cluster topology is defined by specifying a topology path for each node in the cluster. The paths group nodes by rack or switch, depending on how the physical cluster is arranged and how you want Greenplum HD EE to place replicated data.

Topology paths can be as simple or complex as needed to correspond to your cluster layout. In a simple cluster, each topology path might consist of the rack only (e. g. `/rack-1`). In a deployment consisting of multiple large datacenters, each topology path can be much longer (e. g. `/europe/uk/london/datacenter2/room4/row22/rack5/`). Greenplum HD EE uses topology paths to spread out replicated copies of data, placing each copy on a separate path. By setting each path to correspond to a physical rack, you can ensure that replicated data is distributed across racks to improve fault tolerance.

After you have defined node topology for the nodes in your cluster, you can use volume topology to place volumes on specific racks, nodes, or groups of nodes. See Setting Volume Topology.

Setting Node Topology

You can specify a topology path for one or more nodes using the node topo command, or in the Greenplum HD EE Control System using the following procedure.

To set node topology using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **Cluster** group and click the Nodes view.
2. Select the checkbox beside each node whose topology you wish to set.
3. Click the **Change Topology** button to display the Change Node Topology dialog.
4. Set the path in the **New Path** field:
   - To define a new path, type a topology path. Topology paths must begin with a forward slash ('/').
   - To use a path you have already defined, select it from the dropdown.
5. Click **Move Node** to set the new topology.

Shutting Down a Cluster

To safely shut down an entire cluster, preserving all data and full replication, you must follow a specific sequence that stops writes so that the cluster does not shut down in the middle of an operation:

1. Shut down the NFS service everywhere it is running.
2. Shut down the CLDB nodes.
3. Shut down all remaining nodes.

This procedure ensures that on restart the data is replicated and synchronized, so that there is no single point of failure for any data.

To shut down the cluster:

1. Change to the `root` user (or use `sudo` for the following commands).
2. Before shutting down the cluster, you will need a list of NFS nodes, CLDB nodes, and all remaining nodes. Once the CLDB is shut down, you cannot retrieve a list of nodes; it is important to obtain this information at the beginning of the process. Use the `node list` command as follows:
   - Determine which nodes are running the NFS gateway. Example:

     ```
     /opt/mapr/bin/maprcli node list -filter "[rp==/*]and[svc==nfs]" -columns
     id,h,hn,svc, rp
     id                   service
     hostname             health  ip
     6475182753920016590  fileserver,tasktracker,nfs,hoststats
     node-252.cluster.us 0       10.10.50.252
     8077173244974255917  tasktracker,cldb,fileserver,nfs,hoststats
     node-253.cluster.us 0       10.10.50.253
     5323478955232132984  webserver,cldb,fileserver,nfs,hoststats,jobtracker
     node-254.cluster.us 0       10.10.50.254
     ```

   - Determine which nodes are running the CLDB. Example:

     ```
     /opt/mapr/bin/maprcli node list -filter "[rp==/*]and[svc==cldb]" -columns
     id,h,hn,svc, rp
     ```

   - List all non-CLDB nodes. Example:

     ```
     /opt/mapr/bin/maprcli node list -filter "[rp==/*]and[svc!=cldb]" -columns
     id,h,hn,svc, rp
     ```

3. Shut down all NFS instances. Example:

   ```
   /opt/mapr/bin/maprcli node services -nfs stop -nodes
   node-252.cluster.us,node-253.cluster.us,node-254.cluster.us
   ```

4. SSH into each CLDB node and stop the warden. Example:

   ```
   /etc/init.d/mapr-warden stop
   ```

5. SSH into each of the remaining nodes and stop the warden. Example:

   ```
   /etc/init.d/mapr-warden stop
   ```

CLDB Failover

The CLDB automatically replicates its data to other nodes in the cluster, preserving at least two (and generally three) copies of the CLDB data. If the CLDB process dies, it is automatically restarted on the node. All jobs and processes wait for the CLDB to return, and resume from where they left off, with no data or job loss.

If the node itself fails, the CLDB data is still safe, and the cluster can continue normally as soon as the CLDB is started on another node. A failed CLDB node automatically fails over to another CLDB node without user intervention and without data loss.

Users and Groups

Greenplum HD EE detects users and groups from the operating system running on each node. The same users and groups must be configured on all nodes; in large clusters, you should configure nodes to use an LDAP or NIS setup. If you are creating a group, be sure to add the appropriate users to the group. Adding a Greenplum HD EE user or group simply means adding a user or group in your existing scheme, then creating a volume for the user or group.

To create a volume for a user or group

1. In the Volumes view, click **New Volume**.
2. In the New Volume dialog, set the volume attributes:
    - In **Volume Setup**, type a volume name. Make sure the Volume Type is set to Normal Volume.
    - In **Ownership & Permissions**, set the volume owner and specify the users and groups who can perform actions on the volume.
    - In **Usage Tracking**, set the accountable group or user, and set a quota or advisory quota if needed.
    - In **Replication & Snapshot Scheduling**, set the replication factor and choose a snapshot schedule.
3. Click **OK** to save the settings.

See Volumes for more information. You can also create a volume using the volume create command.

You can see users and groups that own volumes in the User Disk Usage view or using the entity list command.

Managing Permissions

Greenplum HD EE manages permissions using two mechanisms:

- Cluster and volume permissions use *access control lists (ACLs)*, which specify actions particular users are allowed to perform on a certain cluster or volume
- MapR-FS permissions control access to directories and files in a manner similar to Linux file permissions. To manage permissions, you must have `fc` permissions.

Cluster and Volume Permissions

Cluster and volume permissions use ACLs, which you can edit using the Greenplum HD EE Control System or the `acl` command s.

Cluster Permissions

The following table lists the actions a user can perform on a cluster, and the corresponding codes used in the cluster ACL.

Code	Allowed Action	Includes
login	Log in to the Greenplum HD EE Control System, use the API and command-line interface, read access on cluster and volumes	cv
ss	Start/stop services	
cv	Create volumes	
a	Admin access	All permissions except fc
fc	Full control (administrative access and permission to change the cluster ACL)	a

Setting Cluster Permissions

You can modify cluster permissions using the acl edit and acl set commands, or using the Greenplum HD EE Control System.

To add cluster permissions using the Greenplum HD EE Control System:

1. Expand the System Settings group and click Permissions to display the **Edit Permissions** dialog.
2. Click **[+ Add Permission]** to add a new row. Each row lets you assign permissions to a single user or group.
3. Type the name of the user or group in the empty text field:
    - If you are adding permissions for a user, type `u:<user>`, replacing `<user>` with the username.
    - If you are adding permissions for a group, type `g:<group>`, replacing `<group>` with the group name.
4. Click the **Open Arrow** (▾) to expand the Permissions dropdown.
5. Select the permissions you wish to grant to the user or group.
6. Click **OK** to save the changes.

To remove cluster permissions using the Greenplum HD EE Control System:

1. Expand the System Settings group and click Permissions to display the **Edit Permissions** dialog.
2. Remove the desired permissions:
3. To remove all permissions for a user or group:
    - Click the delete button (⊗) next to the corresponding row.
4. To change the permissions for a user or group:

    - Click the **Open Arrow** (⌄) to expand the Permissions dropdown.
    - Unselect the permissions you wish to revoke from the user or group.
5. Click **OK** to save the changes.

Volume Permissions

The following table lists the actions a user can perform on a volume, and the corresponding codes used in the volume ACL.

Code	Allowed Action
dump	Dump the volume
restore	Mirror or restore the volume
m	Modify volume properties, create and delete snapshots
d	Delete a volume
fc	Full control (admin access and permission to change volume ACL)

To mount or unmount volumes under a directory, the user must have read/write permissions on the directory (see MapR-FS Permissions).

You can set volume permissions using the acl edit and acl set commands, or using the Greenplum HD EE Control System.

To add volume permissions using the Greenplum HD EE Control System:

1. Expand the MapR-FS group and click Volumes.
    - To create a new volume and set permissions, click **New Volume** to display the New Volume dialog.
    - To edit permissions on a existing volume, click the volume name to display the Volume Properties dialog.
2. In the **Permissions** section, click **[+ Add Permission]** to add a new row. Each row lets you assign permissions to a single user or group.
3. Type the name of the user or group in the empty text field:
    - If you are adding permissions for a user, type `u:<user>`, replacing `<user>` with the username.
    - If you are adding permissions for a group, type `g:<group>`, replacing `<group>` with the group name.
4. Click the **Open Arrow** (⌄) to expand the Permissions dropdown.
5. Select the permissions you wish to grant to the user or group.
6. Click **OK** to save the changes.

To remove volume permissions using the Greenplum HD EE Control System:

1. Expand the MapR-FS group and click Volumes.
2. Click the volume name to display the Volume Properties dialog.
3. Remove the desired permissions:
4. To remove all permissions for a user or group:
    - Click the delete button (⊗) next to the corresponding row.
5. To change the permissions for a user or group:
    - Click the **Open Arrow** (⌄) to expand the Permissions dropdown.
    - Unselect the permissions you wish to revoke from the user or group.
6. Click **OK** to save the changes.

MapR-FS Permissions

MapR-FS permissions are similar to the POSIX permissions model. Each file and directory is associated with a user (the *owner*) and a group. You can set read, write, and execute permissions separately for:

- The owner of the file or directory
- Members of the group associated with the file or directory
- All other users.

The permissions for a file or directory are called its *mode*. The mode of a file or directory can be expressed in two ways:

- Text - a string that indicates the presence of the read (`r`), write (`w`), and execute (`x`) permission or their absence (`-`) for the owner, group, and other users respectively. Example:
  `rwxr-xr-x`
- Octal - three octal digits (for the owner, group, and other users), that use individual bits to represent the three permissions. Example:
  `755`

Both `rwxr-xr-x` and `755` represent the same mode: the owner has all permissions, and the group and other users have read and execute permissions only.

Text Modes

String modes are constructed from the characters in the following table.

Text	Description
u	The file's owner.
g	The group associated with the file or directory.
o	Other users (users that are not the owner, and not in the group).
a	All (owner, group and others).
=	Assigns the permissions Example: "a=rw" sets read and write permissions and disables execution for all.
-	Removes a specific permission. Example: "a-x" revokes execution permission from all users without changing read and write permissions.
+	Adds a specific permission. Example: "a+x" grants execution permission to all users without changing read and write permissions.
r	Read permission
w	Write permission
x	Execute permission

Octal Modes

To construct each octal digit, add together the values for the permissions you wish to grant:

- Read: 4
- Write: 2
- Execute: 1

Syntax

You can change the modes of directories and files in the Greenplum HD EE storage using either the `hadoop fs` command with the `-chmod` option, or using the `chmod` command via NFS. The syntax for both commands is similar:

- `hadoop fs -chmod [-R] <MODE>[,<MODE>]... | <OCTALMODE> <URI> [<URI> ...]`
- `chmod [-R] <MODE>[,<MODE>]... | <OCTALMODE> <URI> [<URI> ...]`

Parameters and Options

Parameter/Option	Description
-R	If specified, this option applies the new mode recursively throughout the directory structure.
MODE	A string that specifies a mode.
OCTALMODE	A three-digit octal number that specifies the new mode for the file or directory.
URI	A relative or absolute path to the file or directory for which to change the mode.

Examples

The following examples are all equivalent:

- `chmod 755 script.sh`
- `chmod u=rwx,g=rx,o=rx script.sh`

- `chmod u=rwx,go=rx script.sh`

Managing Quotas

Quotas limit the disk space used by a volume or an *entity* (user or group) on a cluster, by specifying the amount of disk space the volume or entity is allowed to use:

- A volume quota limits the space used by a volume.
- A user/group quota limits the space used by all volumes owned by a user or group.

Quotas are expressed as an integer value plus a single letter to represent the unit:

- B - bytes
- K - kilobytes
- M - megabytes
- G - gigabytes
- T - terabytes
- P - petabytes

Example: 500G specifies a 500 gigabyte quota.

If a volume or entity exceeds its quota, further disk writes are prevented and a corresponding alarm is raised:

- AE_ALARM_AEQUOTA_EXCEEDED - an entity exceeded its quota
- VOLUME_ALARM_QUOTA_EXCEEDED - a volume exceeded its quota

A quota that prevents writes above a certain threshold is also called a *hard quota*. In addition to the hard quota, you can also set an *advisory* quota for a user, group, or volume. An advisory quota does not enforce disk usage limits, but raises an alarm when it is exceeded:

- AE_ALARM_AEADVISORY_QUOTA_EXCEEDED - an entity exceeded its advisory quota
- VOLUME_ALARM_ADVISORY_QUOTA_EXCEEDED - a volume exceeded its advisory quota

In most cases, it is useful to set the advisory quota somewhat lower than the hard quota, to give advance warning that disk usage is approaching the allowed limit.

To manage quotas, you must have a or `fc` permissions.

Quota Defaults

You can set hard quota and advisory quota defaults for users and groups. When a user or group is created, the default quota and advisory quota apply unless overridden by specific quotas.

Setting Volume Quotas and Advisory Quotas

You can set a volume quota using the volume modify command, or use the following procedure to set a volume quota using the Control System.

To set a volume quota using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **MapR-FS** group and click the Volumes view.
2. Display the Volume Properties dialog by clicking the volume name, or by selecting the checkbox beside the volume name then clicking the **Properties** button.
3. In the Usage Tracking section, select the **Volume Quota** checkbox and type a quota (value and unit) in the field. Example: `500G`
4. To set the advisory quota, select the **Volume Advisory Quota** checkbox and type a quota (value and unit) in the field. Example: `250G`
5. After setting the quota, click **Modify Volume** to exit save changes to the volume.

Setting User/Group Quotas and Advisory Quotas

You can set a user/group quota using the entity modify command, or use the following procedure to set a user/group quota using the Greenplum HD EE Control System.

To set a user or group quota using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the MapR-FS group and click the User Disk Usage view.
2. Select the checkbox beside the user or group name for which you wish to set a quota, then click the **Edit Properties** button to display the User Properties dialog.
3. In the Usage Tracking section, select the **User/Group Quota** checkbox and type a quota (value and unit) in the field. Example: `500G`

4. To set the advisory quota, select the **User/Group Advisory Quota** checkbox and type a quota (value and unit) in the field. Example: `250G`
5. After setting the quota, click **OK** to exit save changes to the entity.

Setting Quota Defaults

You can set an entity quota using the entity modify command, or use the following procedure to set an entity quota using the Greenplum HD EE Control System.

To set quota defaults using the Greenplum HD EE Control System:

1. In the Navigation pane, expand the **System Settings** group.
2. Click the Quota Defaults view to display the **Configure Quota Defaults** dialog.
3. To set the user quota default, select the **Default User Total Quota** checkbox in the User Quota Defaults section, then type a quota (value and unit) in the field.
4. To set the user advisory quota default, select the **Default User Advisory Quota** checkbox in the User Quota Defaults section, then type a quota (value and unit) in the field.
5. To set the group quota default, select the **Default Group Total Quota** checkbox in the Group Quota Defaults section, then type a quota (value and unit) in the field.
6. To set the group advisory quota default, select the **Default Group Advisory Quota** checkbox in the Group Quota Defaults section, then type a quota (value and unit) in the field.
7. After setting the quota, click **Save** to exit save changes to the entity.

Best Practices

File Balancing

Greenplum HD EE distributes volumes to balance files across the cluster. Each volume has a name container that is restricted to one storage pool. The greater the number of volumes, the more evenly Greenplum HD EE can distribute files. For best results, the number of volumes should be greater than the total number of storage pools in the cluster. To accommodate a very large number of files, you can use disksetup with the `-W` option when installing or re-formatting nodes, to create storage pools larger than the default of three disks each.

Disk Setup

It is not necessary to set up RAID on disks used by MapR-FS. Greenplum HD EE uses a script called `disksetup` to set up storage pools. In most cases, you should let Greenplum HD EE calculate storage pools using the default stripe width of two or three disks. If you anticipate a high volume of random-access I/O, you can use the `-W` option with `disksetup` to specify larger storage pools of up to 8 disks each.

Setting Up NFS

The `mapr-nfs` service lets you access data on a licensed Greenplum HD EE cluster via the NFS protocol.

At cluster installation time, plan which nodes should provide NFS access according to your anticipated traffic. You can set up virtual IP addresses (VIPs) for NFS nodes in a Greenplum HD EE cluster, for load balancing or failover. VIPs provide multiple addresses that can be leveraged for round-robin DNS, allowing client connections to be distributed among a pool of NFS nodes. VIPs also make high availability (HA) NFS possible; in the event an NFS node fails, data requests are satisfied by other NFS nodes in the pool.

How you set up NFS depends on your network configuration and bandwidth, anticipated data access, and other factors. You can provide network access from MapR clients to any NFS nodes directly or through a gateway to allow access to data. Here are a few examples of how to configure NFS:

- On a few nodes in the cluster, with VIPs using DNS round-robin to balance connections between nodes (use at least as many VIPs as NFS nodes)
- On all file server nodes, so each node can NFS-mount itself and native applications can run as tasks
- On one or more dedicated gateways (using round-robin DNS or behind a hardware load balancer) to allow controlled access

Here are a few tips:

- Set up NFS on at least three nodes if possible.
- All NFS nodes must be accessible over the network from the machines where you want to mount them.
- To serve a large number of clients, set up dedicated NFS nodes and load-balance between them. If the cluster is behind a firewall, you can provide access through the firewall via a load balancer instead of direct access to each NFS node. You can run NFS on all nodes in the cluster, if needed.
- To provide maximum bandwidth to a specific client, install the NFS service directly on the client machine. The NFS gateway on the client manages how data is sent in or read back from the cluster, using all its network interfaces (that are

on the same subnet as the cluster nodes) to transfer data via Greenplum HD EE APIs, balancing operations among nodes as needed.

- Use VIPs to provide High Availability (HA) and failover. See Setting Up NFS HA for more information.

NFS Memory Settings

The memory allocated to each Greenplum HD EE service is specified in the `/opt/mapr/conf/warden.conf` file, which Greenplum HD EE automatically configures based on the physical memory available on the node. You can adjust the minimum and maximum memory used for NFS, as well as the percentage of the heap that it tries to use, by setting the `percent`, `max`, and `min` parameters in the `warden.conf` file on each NFS node. Example:

```
...
service.command.nfs.heapsize.percent=3
service.command.nfs.heapsize.max=1000
service.command.nfs.heapsize.min=64
...
```

The percentages need not add up to 100; in fact, you can use less than the full heap by setting the `heapsize.percent` parameters for all services to add up to less than 100% of the heap size. In general, you should not need to adjust the memory settings for individual services, unless you see specific memory-related problems occurring.

NIC Configuration

For high performance clusters, use more than one network interface card (NIC) per node. Greenplum HD EE can detect multiple IP addresses on each node and load-balance throughput automatically.

Isolating CLDB Nodes

In a large cluster (100 nodes or more) create CLDB-only nodes to ensure high performance. This configuration also provides additional control over the placement of the CLDB data, for load balancing, fault tolerance, or high availability (HA). Setting up CLDB-only nodes involves restricting the CLDB volume to its own topology and making sure all other volumes are on a separate topology. Unless you specify a default volume topology, new volumes have no topology when they are created, and reside at the root topology path: "`/`". Because both the CLDB-only path and the non-CLDB path are children of the root topology path, new non-CLDB volumes are not guaranteed to keep off the CLDB-only nodes. To avoid this problem, set a default volume topology. See Setting Default Volume Topology.

To set up a CLDB-only node:

1. SET UP the node as usual:
   - PREPARE the node, making sure it meets the requirements.
2. INSTALL only the following packages:
   - mapr-cldb
   - mapr-webserver
   - mapr-core
   - mapr-fileserver
3. RUN `configure.sh`.
4. FORMAT the disks.
5. START the warden:

   ```
   /etc/init.d/mapr-warden start
   ```

To restrict the CLDB volume to specific nodes:

1. Move all CLDB nodes to a CLDB-only topology (e. g. `/cldbonly`) using the Greenplum HD EE Control System or the following command:
   `maprcli node move -serverids <CLDB nodes> -topology /cldbonly`
2. Restrict the CLDB volume to the CLDB-only topology. Use the Greenplum HD EE Control System or the following command:
   `maprcli volume move -name mapr.cldb.internal -topology /cldbonly`
3. If the CLDB volume is present on nodes not in /cldbonly, increase the replication factor of mapr.cldb.internal to create enough copies in
   /cldbonly using the Greenplum HD EE Control System or the following command:
   `maprcli volume modify -name mapr.cldb.internal -replication <replication factor>`
4. Once the volume has sufficient copies, remove the extra replicas by reducing the replication factor to the desired value using the Greenplum HD EE Control System or the command used in the previous step.

To move all other volumes to a topology separate from the CLDB-only nodes:

1. Move all non-CLDB nodes to a non-CLDB topology (e. g. `/defaultRack`) using the Greenplum HD EE Control System or the following command:
   `maprcli node move -serverids <all non-CLDB nodes> -topology /defaultRack`
2. Restrict all existing volumes to the topology `/defaultRack` using the Greenplum HD EE Control System or the following command:
   `maprcli volume move -name <volume> -topology /defaultRack`
   All volumes except (mapr.cluster.root) get re-replicated to the changed topology automatically.

> ⚠ To prevent subsequently created volumes from encroaching on the CLDB-only nodes, set a default topology that excludes the CLDB-only topology.

Isolating ZooKeeper Nodes

For large clusters (100 nodes or more), isolate the ZooKeeper on nodes that do not perform any other function, so that the ZooKeeper does not compete for resources with other processes. Installing a ZooKeeper-only node is similar to any typical node installation, but with a specific subset of packages. Importantly, do not install the FileServer package, so that Greenplum HD EE does not use the ZooKeeper-only node for data storage.

To set up a ZooKeeper-only node:

1. SET UP the node as usual:
   - PREPARE the node, making sure it meets the requirements.
2. INSTALL only the following packages:
   - mapr-zookeeper
   - mapr-zk-internal
   - mapr-core
3. RUN `configure.sh`.
4. FORMAT the disks.
5. START ZooKeeper (as `root` or using `sudo`):

```
/etc/init.d/mapr-zookeeper start
```

> ⚠ Do not start the warden.

Setting Up RAID on the Operating System Partition

You can set up RAID on each node at installation time, to provide higher operating system performance (RAID 0), disk mirroring for failover (RAID 1), or both (RAID 10), for example. See the following instructions from the operating system websites:

- CentOS
- Red Hat

Tuning MapReduce

Greenplum HD EE automatically tunes the cluster for most purposes. A service called the *warden* determines machine resources on nodes configured to run the TaskTracker service, and sets MapReduce parameters accordingly.

On nodes with multiple CPUs, Greenplum HD EE uses taskset to reserve CPUs for Greenplum HD EE services:

- On nodes with five to eight CPUs, CPU 0 is reserved for Greenplum HD EE services
- On nodes with nine or more CPUs, CPU 0 and CPU 1 are reserved for Greenplum HD EE services

In certain circumstances, you might wish to manually tune Greenplum HD EE to provide higher performance. For example, when running a job consisting of unusually large tasks, it is helpful to reduce the number of slots on each TaskTracker and adjust the Java heap size. The following sections provide MapReduce tuning tips. If you change any settings in mapred-site.xml, restart the TaskTracker.

Memory Settings

Memory for Greenplum HD EE Services

The memory allocated to each Greenplum HD EE service is specified in the `/opt/mapr/conf/warden.conf` file, which

Greenplum HD EE automatically configures based on the physical memory available on the node. For example, you can adjust the minimum and maximum memory used for the TaskTracker, as well as the percentage of the heap that the TaskTracker tries to use, by setting the appropriate `percent`, `max`, and `min` parameters in the `warden.conf` file:

```
...
service.command.tt.heapsize.percent=2
service.command.tt.heapsize.max=325
service.command.tt.heapsize.min=64
...
```

The percentages of memory used by the services need not add up to 100; in fact, you can use less than the full heap by setting the `heapsize.percent` parameters for all services to add up to less than 100% of the heap size. In general, you should not need to adjust the memory settings for individual services, unless you see specific memory-related problems occurring.

MapReduce Memory

The memory allocated for MapReduce tasks normally equals the total system memory minus the total memory allocated for Greenplum HD EE services. If necessary, you can use the parameter mapreduce.tasktracker.reserved.physicalmemory.mb to set the maximum physical memory reserved by MapReduce tasks, or you can set it to `-1` to disable physical memory accounting and task management.

If the node runs out of memory, MapReduce tasks are killed by the OOM-killer to free memory. You can use `mapred.child.oom_adj` (copy from `mapred-default.xml` to adjust the `oom_adj` parameter for MapReduce tasks. The possible values of `oom_adj` range from -17 to +15. The higher the score, more likely the associated process is to be killed by OOM-killer.

Job Configuration

Map Tasks

Map tasks use memory mainly in two ways:

- The MapReduce framework uses an intermediate buffer to hold serialized (key, value) pairs.
- The application consumes memory to run the map function.

MapReduce framework memory is controlled by `io.sort.mb`. If `io.sort.mb` is less than the data emitted from the mapper, the task ends up spilling data to disk. If `io.sort.mb` is too large, the task can run out of memory or waste allocated memory. By default `io.sort.mb` is 100mb. It should be approximately 1.25 times the number of data bytes emitted from mapper. If you cannot resolve memory problems by adjusting `io.sort.mb`, then try to re-write the application to use less memory in its map function.

Reduce Tasks

If tasks fail because of an Out of Heap Space error, increase the heap space (the `-Xmx` option in `mapred.reduce.child.java.opts`) to give more memory to the tasks. If map tasks are failing, you can also try reducing `io.sort.mb`.
(see mapred.map.child.java.opts in mapred-site.xml)

TaskTracker Configuration

Greenplum HD EE sets up map and reduce slots on each TaskTracker node using formulas based on the number of CPUs present on the node. The default formulas are stored in the following parameters in `mapred-site.xml`:

- `mapred.tasktracker.map.tasks.maximum`: (CPUS > 2) ? (CPUS * 0.75) : 1 (At least one Map slot, up to 0.75 times the number of CPUs)
- `mapred.tasktracker.reduce.tasks.maximum`: (CPUS > 2) ? (CPUS * 0.50) : 1 (At least one Map slot, up to 0.50 times the number of CPUs)

You can adjust the maximum number of map and reduce slots by editing the formula used in `mapred.tasktracker.map.tasks.maximum` and `mapred.tasktracker.reduce.tasks.maximum`. The following variables are used in the formulas:

- `CPUS` - number of CPUs present on the node
- `DISKS` - number of disks present on the node
- `MEM` - memory reserved for MapReduce tasks

Ideally, the number of map and reduce slots should be decided based on the needs of the application. Map slots should be based on how many map tasks can fit in memory, and reduce slots should be based on the number of CPUs. If each task in a MapReduce job takes 3 GB, and each node has 9GB reserved for MapReduce tasks, then the total number of map slots should be 3. The amount of data each map task must process also affects how many map slots should be configured. If each map task processes 256 MB (the default chunksize in Greenplum HD EE), then each map task should have 800 MB of memory. If there are 4 GB reserved for map tasks, then the number of map slots should be 4000MB/800MB, or 5 slots.

Greenplum HD EE allows the JobTracker to over-schedule tasks on TaskTracker nodes in advance of the availability of slots, creating a pipeline. This optimization allows TaskTracker to launch each map task as soon as the previous running map task finishes. The number of tasks to over-schedule should be about 25-50% of total number of map slots. You can adjust this number with the parameter mapreduce.tasktracker.prefetch.maptasks.

Troubleshooting Out-of-Memory Errors

When the aggregated memory used by MapReduce tasks exceeds the memory reserve on a TaskTracker node, tasks can fail or be killed. Greenplum HD EE attempts to prevent out-of-memory exceptions by killing MapReduce tasks when memory becomes scarce. If you allocate too little Java heap for the expected memory requirements of your tasks, an exception can occur. The following steps can help configure Greenplum HD EE to avoid these problems:

- If a particular job encounters out-of-memory conditions, the simplest way to solve the problem might be to reduce the memory footprint of the map and reduce functions, and to ensure that the partitioner distributes map output to reducers evenly.
- If it is not possible to reduce the memory footprint of the application, try increasing the Java heap size (-Xmx) in the client-side MapReduce configuration.
- If many jobs encounter out-of-memory conditions, or if jobs tend to fail on specific nodes, it may be that those nodes are advertising too many TaskTracker slots. In this case, the cluster administrator should reduce the number of slots on the affected nodes.

To reduce the number of slots on a node:

1. Stop the TaskTracker service on the node:

```
$ sudo maprcli node services -nodes <node name> -tasktracker stop
```

2. Edit the file `/opt/mapr/hadoop/hadoop-<version>/conf/mapred-site.xml`:
   - Reduce the number of map slots by lowering `mapred.tasktracker.map.tasks.maximum`
   - Reduce the number of reduce slots by lowering `mapred.tasktracker.reduce.tasks.maximum`
3. Start the TaskTracker on the node:

```
$ sudo  maprcli node services -nodes <node name> -tasktracker start
```

ExpressLane

Greenplum HD EE provides an express path for small MapReduce jobs to run when all slots are occupied by long tasks. Small jobs are only given this special treatment when the cluster is busy, and only if they meet the criteria specified by the following parameters in `mapred-site.xml`:

Parameter	Value	Description
mapred.fairscheduler.smalljob.schedule.enable	true	Enable small job fast scheduling inside fair scheduler. TaskTrackers should reserve a slot called ephemeral slot which is used for smalljob if cluster is busy.
mapred.fairscheduler.smalljob.max.maps	10	Small job definition. Max number of maps allowed in small job.
mapred.fairscheduler.smalljob.max.reducers	10	Small job definition. Max number of reducers allowed in small job.
mapred.fairscheduler.smalljob.max.inputsize	10737418240	Small job definition. Max input size in bytes allowed for a small job. Default is 10GB.
mapred.fairscheduler.smalljob.max.reducer.inputsize	1073741824	Small job definition. Max estimated input size for a reducer allowed in small job. Default is 1GB per reducer.
mapred.cluster.ephemeral.tasks.memory.limit.mb	200	Small job definition. Max memory in mbytes reserved for an ephermal slot. Default is 200mb. This value must be same on JobTracker and TaskTracker nodes.

MapReduce jobs that appear to fit the small job definition but are in fact larger than anticipated are killed and re-queued for normal execution.

HBase

* The HBase write-ahead log (WAL) writes many tiny records, and compressing it would cause massive CPU load. Before using HBase, turn off compression for directories in the HBase volume (normally mounted at `/hbase`. Example:

```
hadoop mfs \-setcompression off /hbase
```

* You can check whether compression is turned off in a directory or mounted volume by using `[hadoop mfs]` to list the file contents. Example:

```
hadoop mfs \-ls /hbase
```

The letter `Z` in the output indicates compression is turned on; the letter `U` indicates compression is turned off. See `hadoop mfs` for more information.
* On any node where you plan to run both HBase and MapReduce, give more memory to the FileServer than to the RegionServer so that the node can handle high throughput. For example, on a node with 24 GB of physical memory, it might be desirable to limit the RegionServer to 4 GB, give 10 GB to MapR-FS, and give the remainder to TaskTracker. To change the memory allocated to each service, edit the `/opt/mapr/conf/warden.conf` file. See Tuning MapReduce for more information.

Troubleshooting

This section provides information about troubleshooting cluster problems:

- Disaster Recovery
- Troubleshooting Alarms

Disaster Recovery

It is a good idea to set up an automatic backup of the CLDB volume at regular intervals; in the event that all CLDB nodes fail, you can restore the CLDB from a backup. If you have more than one Greenplum HD EE cluster, you can back up the CLDB volume for each cluster onto the other clusters; otherwise, you can save the CLDB locally to external media such as a USB drive.

To back up a CLDB volume from a remote cluster:

1. Set up a cron job on the remote cluster to save the container information to a file by running the following command:
   `/opt/mapr/bin/maprcli dump cldbnodes -zkconnect <IP:port of ZooKeeper leader>  > <path to file>`
2. Set up a cron job to copy the container information file to a volume on the local cluster.
3. Create a mirror volume on the local cluster, choosing the volume `mapr.cldb.internal` from the remote cluster as the source volume. Set the mirror sync schedule so that it will run at the same time as the cron job.

To back up a CLDB volume locally:

1. Set up a cron job to save the container information to a file on external media by running the following command:
   `/opt/mapr/bin/maprcli dump cldbnodes -zkconnect <IP:port of ZooKeeper leader>  > <path to file>`
2. Set up a cron job to create a dump file of the local volume `mapr.cldb.internal` on external media. Example:
   `/opt/mapr/bin/maprcli volume dump create  -name mapr.cldb.internal -dumpfile <path_to_file>`

For information about restoring from a backup of the CLDB, contact Greenplum Support.

Out of Memory Troubleshooting

When the aggregated memory used by MapReduce tasks exceeds the memory reserve on a TaskTracker node, tasks can fail or be killed. Greenplum HD EE attempts to prevent out-of-memory exceptions by killing MapReduce tasks when memory becomes scarce. If you allocate too little Java heap for the expected memory requirements of your tasks, an exception can occur. The following steps can help configure Greenplum HD EE to avoid these problems:

- If a particular job encounters out-of-memory conditions, the simplest way to solve the problem might be to reduce the memory footprint of the map and reduce functions, and to ensure that the partitioner distributes map output to reducers evenly.

- If it is not possible to reduce the memory footprint of the application, try increasing the Java heap size (-Xmx) in the client-side MapReduce configuration.
- If many jobs encounter out-of-memory conditions, or if jobs tend to fail on specific nodes, it may be that those nodes are advertising too many TaskTracker slots. In this case, the cluster administrator should reduce the number of slots on the affected nodes.

To reduce the number of slots on a node:

1. Stop the TaskTracker service on the node:

```
$ sudo maprcli node services -nodes <node name> -tasktracker stop
```

2. Edit the file `/opt/mapr/hadoop/hadoop-<version>/conf/mapred-site.xml`:
   - Reduce the number of map slots by lowering `mapred.tasktracker.map.tasks.maximum`
   - Reduce the number of reduce slots by lowering `mapred.tasktracker.reduce.tasks.maximum`
3. Start the TaskTracker on the node:

```
$ sudo  maprcli node services -nodes <node name> -tasktracker start
```

Troubleshooting Alarms

User/Group Alarms

User/group alarms indicate problems with user or group quotas. The following tables describe the Greenplum HD EE user/group alarms.

Entity Advisory Quota Alarm

UI Column	User Advisory Quota Alarm
Logged As	AE_ALARM_AEADVISORY_QUOTA_EXCEEDED
Meaning	A user or group has exceeded its advisory quota. See Managing Quotas for more information about user/group quotas.
Resolution	No immediate action is required. To avoid exceeding the hard quota, clear space on volumes created by the user or group, or stop further data writes to those volumes.

Entity Quota Alarm

UI Column	User Quota Alarm
Logged As	AE_ALARM_AEQUOTA_EXCEEDED
Meaning	A user or group has exceeded its quota. Further writes by the user or group will fail. See Managing Quotas for more information about user/group quotas.
Resolution	Free some space on the volumes created by the user or group, or increase the user or group quota.

Cluster Alarms

Cluster alarms indicate problems that affect the cluster as a whole. The following tables describe the Greenplum HD EE cluster alarms.

Blacklist Alarm

UI Column	Blacklist Alarm
Logged As	CLUSTER_ALARM_BLACKLIST_TTS

Meaning	The JobTracker has blacklisted a TaskTracker node because tasks on the node have failed too many times.
Resolution	To determine which node or nodes have been blacklisted, see the JobTracker status page (click **JobTracker** in the Navigation Pane). The JobTracker status page provides links to the TaskTracker log for each node; look at the log for the blacklisted node or nodes to determine why tasks are failing on the node.

License Near Expiration

UI Column	License Near Expiration Alarm
Logged As	CLUSTER_ALARM_LICENSE_NEAR_EXPIRATION
Meaning	The license associated with the cluster is within 30 days of expiration.
Resolution	Renew the license.

License Expired

UI Column	License Expiration Alarm
Logged As	CLUSTER_ALARM_LICENSE_EXPIRED
Meaning	The license associated with the cluster has expired.
Resolution	Renew the license.

Cluster Almost Full

UI Column	Cluster Almost Full
Logged As	CLUSTER_ALARM_CLUSTER_ALMOST_FULL
Meaning	The cluster storage is almost full. The percentage of storage used before this alarm is triggered is 90% by default, and is controlled by the configuration parameter `cldb.cluster.almost.full.percentage`.
Resolution	Reduce the amount of data stored in the cluster. If the cluster storage is less than 90% full, check the `cldb.cluster.almost.full.percentage` parameter via the `config load` command, and adjust it if necessary via the `config save` command.

Cluster Full

UI Column	Cluster Full
Logged As	CLUSTER_ALARM_CLUSTER_FULL
Meaning	The cluster storage is full. MapReduce operations have been halted.
Resolution	Free up some space on the cluster.

Upgrade in Progress

UI Column	Software Installation & Upgrades
Logged As	CLUSTER_ALARM_UPGRADE_IN_PROGRESS
Meaning	A rolling upgrade of the cluster is in progress.

Resolution	No action is required. Performance may be affected during the upgrade, but the cluster should still function normally. After the upgrade is complete, the alarm is cleared.

VIP Assignment Failure

UI Column	VIP Assignment Alarm
Logged As	CLUSTER_ALARM_UNASSIGNED_VIRTUAL_IPS
Meaning	Greenplum HD EE was unable to assign a VIP to any NFS servers.
Resolution	Check the VIP configuration, and make sure at least one of the NFS servers in the VIP pool are up and running. See Configuring NFS for HA.

Node Alarms

Node alarms indicate problems in individual nodes. The following tables describe the Greenplum HD EE node alarms.

CLDB Service Alarm

UI Column	CLDB Alarm
Logged As	NODE_ALARM_SERVICE_CLDB_DOWN
Meaning	The CLDB service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the CLDB service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the CLDB service, the alarm is cleared. If the warden is unable to restart the CLDB service, it may be necessary to contact technical support.

Core Present Alarm

UI Column	Core files present
Logged As	NODE_ALARM_CORE_PRESENT
Meaning	A service on the node has crashed and created a core dump file.
Resolution	Contact technical support.

Debug Logging Active

UI Column	Excess Logs Alarm
Logged As	NODE_ALARM_DEBUG_LOGGING
Meaning	Debug logging is enabled on the node.
Resolution	Debug logging generates enormous amounts of data, and can fill up disk space. If debug logging is not absolutely necessary, turn it off: either use the Manage Services pane in the Node Properties view or the setloglevel command. If it is absolutely necessary, make sure that the logs in /opt/mapr/logs are not in danger of filling the entire disk.

Disk Failure

UI Column	Disk Failure Alarm
Logged As	NODE_ALARM_DISK_FAILURE
Meaning	A disk has failed on the node.

Resolution	Check the disk health log (/opt/mapr/logs/faileddisk.log) to determine which disk failed and view any SMART data provided by the disk.

FileServer Service Alarm

UI Column	FileServer Alarm
Logged As	NODE_ALARM_SERVICE_FILESERVER_DOWN
Meaning	The FileServer service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the FileServer service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the FileServer service, the alarm is cleared. If the warden is unable to restart the FileServer service, it may be necessary to contact technical support.

HBMaster Service Alarm

UI Column	HBase Master Alarm
Logged As	NODE_ALARM_SERVICE_HBMASTER_DOWN
Meaning	The HBMaster service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the HBMaster service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the HBMaster service, the alarm is cleared. If the warden is unable to restart the HBMaster service, it may be necessary to contact technical support.

HBRegion Service Alarm

UI Column	HBase RegionServer Alarm
Logged As	NODE_ALARM_SERVICE_HBREGION_DOWN
Meaning	The HBRegion service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the HBRegion service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the HBRegion service, the alarm is cleared. If the warden is unable to restart the HBRegion service, it may be necessary to contact technical support.

Hoststats Alarm

UI Column	Hoststats process down
Logged As	NODE_ALARM_HOSTSTATS_DOWN
Meaning	The Hoststats service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the Hoststats service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the service, the alarm is cleared. If the warden is unable to restart the service, it may be necessary to contact technical support.

Installation Directory Full Alarm

UI Column	Installation Directory full
Logged As	NODE_ALARM_OPT_MAPR_FULL
Meaning	The partition `/opt/mapr` on the node is running out of space.
Resolution	Free up some space in `/opt/mapr` on the node.

JobTracker Service Alarm

UI Column	JobTracker Alarm
Logged As	NODE_ALARM_SERVICE_JT_DOWN
Meaning	The JobTracker service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the JobTracker service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the JobTracker service, the alarm is cleared. If the warden is unable to restart the JobTracker service, it may be necessary to contact technical support.

NFS Service Alarm

UI Column	NFS Alarm
Logged As	NODE_ALARM_SERVICE_NFS_DOWN
Meaning	The NFS service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the NFS service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the NFS service, the alarm is cleared. If the warden is unable to restart the NFS service, it may be necessary to contact technical support.

Root Partition Full Alarm

UI Column	Root partition full
Logged As	NODE_ALARM_ROOT_PARTITION_FULL
Meaning	The root partition ('/') on the node is running out of space.
Resolution	Free up some space in the root partition of the node.

TaskTracker Service Alarm

UI Column	TaskTracker Alarm
Logged As	NODE_ALARM_SERVICE_TT_DOWN
Meaning	The TaskTracker service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the TaskTracker service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the TaskTracker service, the alarm is cleared. If the warden is unable to restart the TaskTracker service, it may be necessary to contact technical support.

Time Skew Alarm

UI Column	Time Skew Alarm

Logged As	NODE_ALARM_TIME_SKEW
Meaning	The clock on the node is out of sync with the master CLDB by more than 20 seconds.
Resolution	Use NTP to synchronize the time on all the nodes in the cluster.

Version Alarm

UI Column	Version Alarm
Logged As	NODE_ALARM_VERSION_MISMATCH
Meaning	One or more services on the node are running an unexpected version.
Resolution	Stop the node, Restore the correct version of any services you have modified, and re-start the node. See Managing Nodes.

WebServer Service Alarm

UI Column	WebServer Alarm
Logged As	NODE_ALARM_SERVICE_WEBSERVER_DOWN
Meaning	The WebServer service on the node has stopped running.
Resolution	Go to the Manage Services pane of the Node Properties View to check whether the WebServer service is running. The warden will try several times to restart processes automatically. If the warden successfully restarts the WebServer service, the alarm is cleared. If the warden is unable to restart the WebServer service, it may be necessary to contact technical support.

Volume Alarms

Volume alarms indicate problems in individual volumes. The following tables describe the Greenplum HD EE volume alarms.

Data Unavailable

UI Column	Data Alarm
Logged As	VOLUME_ALARM_DATA_UNAVAILABLE
Meaning	This is a potentially very serious alarm that may indicate data loss. Some of the data on the volume cannot be located. This alarm indicates that enough nodes have failed to bring the replication factor of part or all of the volume to zero. For example, if the volume is stored on a single node and has a replication factor of one, the Data Unavailable alarm will be raised if that volume fails or is taken out of service unexpectedly. If a volume is replicated properly (and therefore is stored on multiple nodes) then the Data Unavailable alarm can indicate that a significant number of nodes is down.
Resolution	Investigate any nodes that have failed or are out of service. • You can see which nodes have failed by looking at the Cluster Node Heatmap pane of the Dashboard. • Check the cluster(s) for any snapshots or mirrors that can be used to re-create the volume. You can see snapshots and mirrors in the MapR-FS view.

Data Under-Replicated

UI Column	Replication Alarm

Logged As	VOLUME_ALARM_DATA_UNDER_REPLICATED
Meaning	The volume replication factor is lower than the minimum replication factor set in Volume Properties. This can be caused by failing disks or nodes, or the cluster may be running out of storage space.
Resolution	Investigate any nodes that are failing. You can see which nodes have failed by looking at the Cluster Node Heatmap pane of the Dashboard. Determine whether it is necessary to add disks or nodes to the cluster. This alarm is generally raised when the nodes that store the volumes or replicas have not sent a heartbeat for five minutes. To prevent re-replication during normal maintenance procedures, Greenplum HD EE waits a specified interval (by default, one hour) before considering the node dead and re-replicating its data. You can control this interval by setting the `cldb.fs.mark.rereplicate.sec` parameter using the config save command.

Mirror Failure

UI Column	Mirror Alarm
Logged As	VOLUME_ALARM_MIRROR_FAILURE
Meaning	A mirror operation failed.
Resolution	Make sure the CLDB is running on both the source cluster and the destination cluster. Look at the CLDB log (/opt/mapr/logs/cldb.log) and the MapR-FS log (/opt/mapr/logs/mfs.log) on both clusters for more information. If the attempted mirror operation was between two clusters, make sure that both clusters are reachable over the network. Make sure the source volume is available and reachable from the cluster that is performing the mirror operation.

No Nodes in Topology

UI Column	No Nodes in Vol Topo
Logged As	VOLUME_ALARM_NO_NODES_IN_TOPOLOGY
Meaning	The path specified in the volume's topology no longer corresponds to a physical topology that contains any nodes, either due to node failures or changes to node topology settings. While this alarm is raised, Greenplum HD EE places data for the volume on nodes outside the volume's topology to prevent write failures.
Resolution	Add nodes to the specified volume topology, either by moving existing nodes or adding nodes to the cluster. See Node Topology.

Snapshot Failure

UI Column	Snapshot Alarm
Logged As	VOLUME_ALARM_SNAPSHOT_FAILURE
Meaning	A snapshot operation failed.
Resolution	Make sure the CLDB is running. Look at the CLDB log (/opt/mapr/logs/cldb.log) and the MapR-FS log (/opt/mapr/logs/mfs.log) on both clusters for more information. If the attempted snapshot was a scheduled snapshot that was running in the background, try a manual snapshot.

Volume Advisory Quota Alarm

UI Column	Vol Advisory Quota Alarm

Logged As	VOLUME_ALARM_ADVISORY_QUOTA_EXCEEDED
Meaning	A volume has exceeded its advisory quota.
Resolution	No immediate action is required. To avoid exceeding the hard quota, clear space on the volume or stop further data writes.

Volume Quota Alarm

UI Column	Vol Quota Alarm
Logged As	VOLUME_ALARM_QUOTA_EXCEEDED
Meaning	A volume has exceeded its quota. Further writes to the volume will fail.
Resolution	Free some space on the volume or increase the volume hard quota.

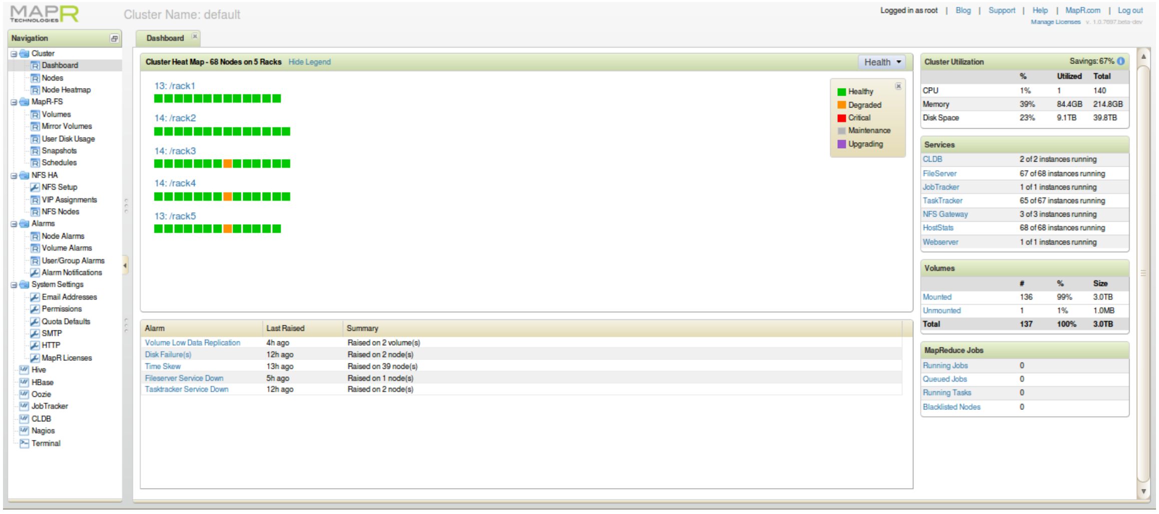
Reference Guide

This guide contains reference information:

- API Reference - information about the command-line interface and the REST API
- Greenplum HD EE Control System Reference - user interface reference guide
- Glossary - essential terms and definitions
- Release Notes - known issues and new features, by release

Greenplum HD EE Control System Reference

The Greenplum HD EE Control System main screen consists of a navigation pane to the left and a view to the right. Dialogs appear over the main screen to perform certain actions.



Alarm	Last Raised	Summary
Volume Low Data Replication	4h ago	Raised on 2 volume(s)
Disk Failure(s)	12h ago	Raised on 2 node(s)
Time Skew	13h ago	Raised on 39 node(s)
Fileserver Service Down	5h ago	Raised on 1 node(s)
Tasktracker Service Down	12h ago	Raised on 2 node(s)

Cluster Utilization — Savings: 67%

	%	Utilized	Total
CPU	1%	1	140
Memory	39%	84.4GB	214.8GB
Disk Space	23%	9.1TB	39.8TB

Services	
CLDB	2 of 2 instances running
FileServer	67 of 68 instances running
JobTracker	1 of 1 instances running
TaskTracker	65 of 67 instances running
NFS Gateway	3 of 3 instances running
HostStats	68 of 68 instances running
Webserver	1 of 1 instances running

Volumes

	#	%	Size
Mounted	136	99%	3.0TB
Unmounted	1	1%	1.0MB
Total	**137**	**100%**	**3.0TB**

MapReduce Jobs	
Running Jobs	0
Queued Jobs	0
Running Tasks	0
Blacklisted Nodes	0

The Navigation pane to the left lets you choose which *view* to display on the right.

The main view groups are:

- Cluster - information about the nodes in the cluster
- MapR-FS - information about volumes, snapshots and schedules
- NFS HA - NFS nodes and virtual IP addresses
- Alarms - node and volume alarms
- System Settings - configuration of alarm notifications, quotas, users, groups, SMTP, and HTTP

Some other views are separate from the main navigation tree:

- Hive - information about Hive on the cluster
- HBase - information about HBase on the cluster
- Oozie - information about Oozie on the cluster

- JobTracker - information about the JobTracker
- CLDB - information about the container location database
- Nagios - generates a Nagios script
- Terminal - an ssh terminal for logging in to the cluster

Views

Views display information about the system. As you open views, tabs along the top let you switch between them quickly.

Clicking any column name in a view sorts the data in ascending or descending order by that column.

Most views contain the following controls:

- a Filter toolbar that lets you sort data in the view, so you can quickly find the information you want
- an info symbol (ⓘ) that you can click for help

Some views contain collapsible panes that provide different types of detailed information. Each collapsible a control at the top left that expands and collapses the pane. The control changes to show the state of the pane:

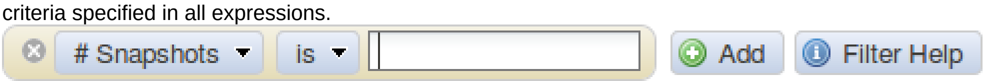
- ▼ - pane is collapsed; click to expand
- ▲ - pane is expanded; click to collapse

Views that contain many results provide the following controls:

- ⏮ (**First**) - navigates to the first screenful of results
- ◀ (**Previous**) - navigates to the previous screenful of results
- ▶ (**Next**) - navigates to the next screenful of results
- ⏭ (**Last**) - navigates to the last screenful of results
- 🔄 (**Refresh**) - refreshes the list of results

The Filter Toolbar

The Filter toolbar lets you build search expressions to provide sophisticated filtering capabilities for locating specific data on views that display a large number of nodes. Expressions are implicitly connected by the AND operator; any search results satisfy the criteria specified in all expressions.



There are three controls in the Filter toolbar:

- The close control (⊗) removes the expression.
- The **Add** button adds a new expression.
- The **Filter Help** button displays brief help about the Filter toolbar.

Expressions

Each expression specifies a semantic statement that consists of a field, an operator, and a value.

- The first dropdown menu specifies the field to match.
- The second dropdown menu specifies the type of match to perform:
- The text field specifies a value to match or exclude in the field. You can use a wildcard to substitute for any part of the string.

Cluster

The Cluster view group provides the following views:

- Dashboard - a summary of information about cluster health, activity, and usage
- Nodes - information about nodes in the cluster
- Node Heatmap - a summary of the health of nodes in the cluster

Dashboard

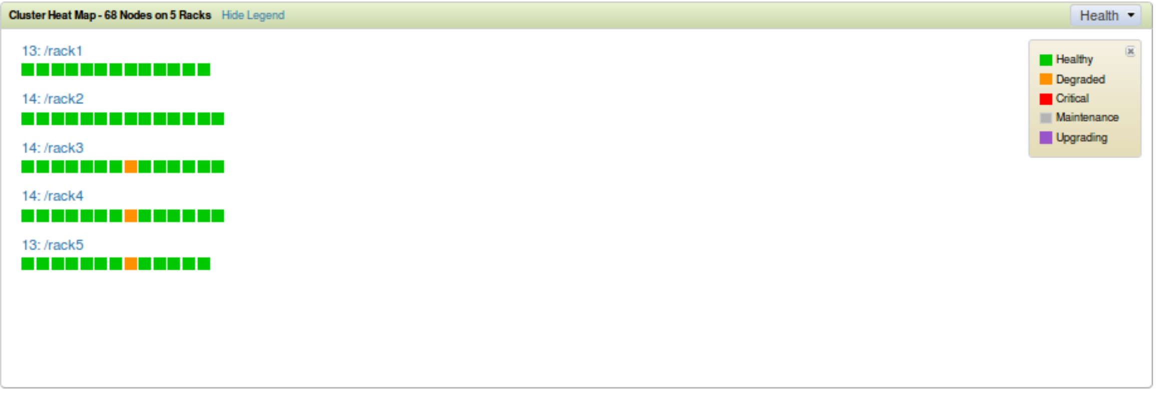
The Dashboard displays a summary of information about the cluster in five panes:

- Cluster Heat Map - the alarms and health for each node, by rack
- Alarms - a summary of alarms for the cluster
- Cluster Utilization - CPU, Memory, and Disk Space usage
- Services - the number of instances of each service
- Volumes - the number of available, under-replicated, and unavailable volumes
- MapReduce Jobs - the number of running and queued jobs, running tasks, and blacklisted nodes

Links in each pane provide shortcuts to more detailed information. The following sections provide information about each pane.

Cluster Heat Map

The Cluster Heat Map pane displays the health of the nodes in the cluster, by rack. Each node appears as a colored square to show its health at a glance.



Cluster Heat Map - 68 Nodes on 5 Racks Hide Legend

Health ▾

13: /rack1
🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩

14: /rack2
🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩

14: /rack3
🟩🟩🟩🟩🟩🟩🟩🟧🟩🟩🟩🟩🟩🟩

14: /rack4
🟩🟩🟩🟩🟩🟩🟩🟧🟩🟩🟩🟩🟩🟩

13: /rack5
🟩🟩🟩🟩🟩🟩🟩🟧🟩🟩🟩🟩🟩

🟩 Healthy
🟧 Degraded
🟥 Critical
⬜ Maintenance
🟪 Upgrading

The Show Legend/Hide Legend link above the heatmap shows or hides a key to the color-coded display.

The drop-down menu at the top right of the pane lets you filter the results to show the following criteria:

- Health
  - 🟩 (green): healthy; all services up, MapR-FS and all disks OK, and normal heartbeat
  - 🟧 (orange): degraded; one or more services down, or no heartbeat for over 1 minute
  - 🟥 (red): critical; Mapr-FS Inactive/Dead/Replicate, or no heartbeat for over 5 minutes
  - ⬜ (gray): maintenance
  - 🟪 (purple): upgrade in process
- CPU Utilization
  - 🟩 (green): below 50%; 🟧 (orange): 50% - 80%; 🟥 (red): over 80%
- Memory Utilization
  - 🟩 (green): below 50%; 🟧 (orange): 50% - 80%; 🟥 (red): over 80%
- Disk Space Utilization
  - 🟩 (green): below 50%; 🟧 (orange): 50% - 80%; 🟥 (red): over 80% or all disks dead
- Disk Failure(s) - status of the NODE_ALARM_DISK_FAILURE alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Excessive Logging - status of the NODE_ALARM_DEBUG_LOGGING alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Software Installation & Upgrades - status of the NODE_ALARM_VERSION_MISMATCH alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Time Skew - status of theNODE_ALARM_ TIME_SKEW alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- CLDB Service Down - status of the NODE_ALARM_SERVICE_CLDB_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- FileServer Service Down - status of the NODE_ALARM_SERVICE_FILESERVER_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- JobTracker Service Down - status of the NODE_ALARM_SERVICE_JT_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- TaskTracker Service Down - status of the NODE_ALARM_SERVICE_TT_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- HBase Master Service Down - status of the NODE_ALARM_SERVICE_HBMASTER_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- HBase Regionserver Service Down - status of the NODE_ALARM_SERVICE_HBREGION_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- NFS Service Down - status of the NODE_ALARM_SERVICE_NFS_DOWN alarm

  - 🟥 (red): raised; 🟩 (green): cleared
- WebServer Service Down - status of the NODE_ALARM_SERVICE_WEBSERVER_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Hoststats Service Down - status of the NODE_ALARM_SERVICE_HOSTSTATS_DOWN alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Root Partition Full - status of the NODE_ALARM_ROOT_PARTITION_FULL alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Installation Directory Full - status of the NODE_ALARM_OPT_MAPR_FULL alarm
  - 🟥 (red): raised; 🟩 (green): cleared
- Cores Present - status of the NODE_ALARM_CORE_PRESENT alarm
  - 🟥 (red): raised; 🟩 (green): cleared

Clicking a rack name navigates to the Nodes view, which provides more detailed information about the nodes in the rack.

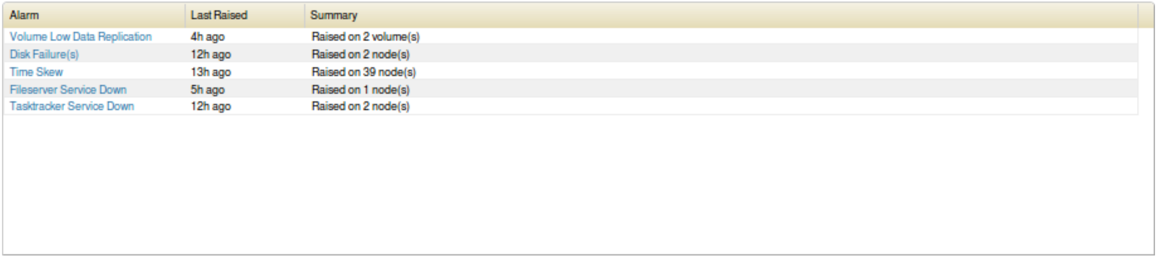
Clicking a colored square navigates to the Node Properties View, which provides detailed information about the node.

Alarms

The Alarms pane displays the following information about alarms on the system:

- Alarm - a list of alarms raised on the cluster
- Last Raised - the most recent time each alarm state changed
- Summary - how many nodes or volumes have raised each alarm

Clicking any column name sorts data in ascending or descending order by that column.



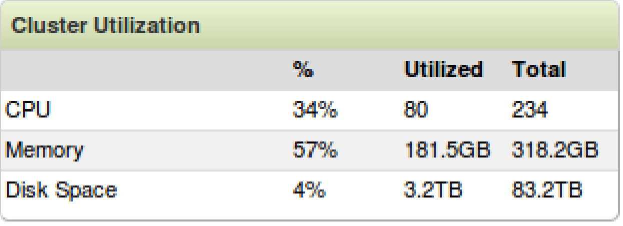
Alarm	Last Raised	Summary
Volume Low Data Replication	4h ago	Raised on 2 volume(s)
Disk Failure(s)	12h ago	Raised on 2 node(s)
Time Skew	13h ago	Raised on 39 node(s)
Fileserver Service Down	5h ago	Raised on 1 node(s)
Tasktracker Service Down	12h ago	Raised on 2 node(s)

Cluster Utilization

The Cluster Utilization pane displays a summary of the total usage of the following resources:

- CPU
- Memory
- Disk Space

For each resource type, the pane displays the percentage of cluster resources used, the amount used, and the total amount present in the system.

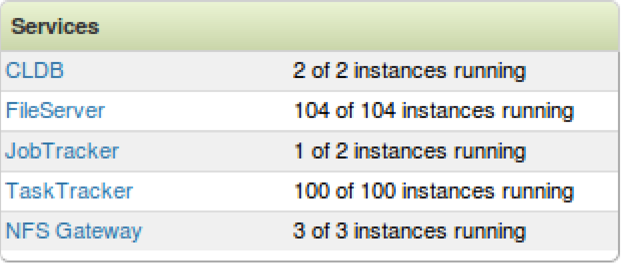


Cluster Utilization

	%	Utilized	Total
CPU	34%	80	234
Memory	57%	181.5GB	318.2GB
Disk Space	4%	3.2TB	83.2TB

Services

The Services pane shows information about the services running on the cluster. For each service, the pane displays how many instances are running out of the total possible number of instances.

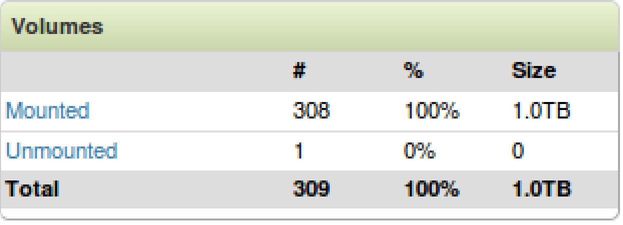


Services	
CLDB	2 of 2 instances running
FileServer	104 of 104 instances running
JobTracker	1 of 2 instances running
TaskTracker	100 of 100 instances running
NFS Gateway	3 of 3 instances running

Clicking a service navigates to the Services view.

Volumes

The Volumes pane displays the total number of volumes, and the number of volumes that are mounted and unmounted. For each category, the Volumes pane displays the number, percent of the total, and total size.



Volumes

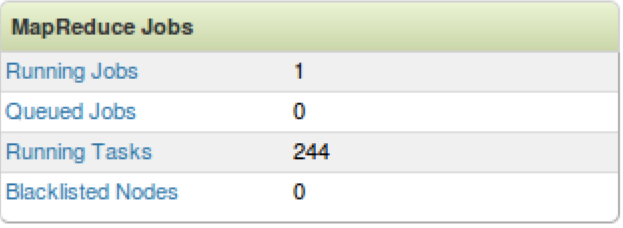
	#	%	Size
Mounted	308	100%	1.0TB
Unmounted	1	0%	0
Total	**309**	**100%**	**1.0TB**

Clicking **mounted** or **unmounted** navigates to the Volumes view.

MapReduce Jobs

The MapReduce Jobs pane shows information about MapReduce jobs:

- Running Jobs - the number of MapReduce jobs currently running
- Queued Jobs - the number of MapReduce jobs queued to run
- Running Tasks - the number of MapReduce tasks currently running
- Blacklisted Nodes - the number of nodes that have been eliminated from the MapReduce pool

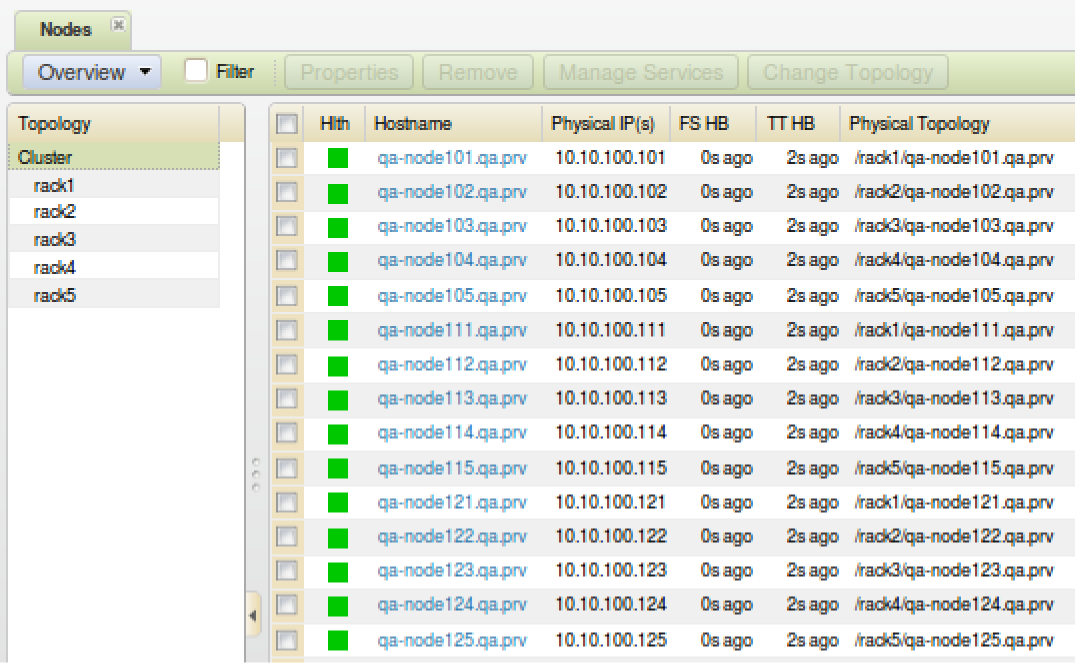


MapReduce Jobs	
Running Jobs	1
Queued Jobs	0
Running Tasks	244
Blacklisted Nodes	0

Nodes

The Nodes view displays the nodes in the cluster, by rack. The Nodes view contains two panes: the Topology pane and the Nodes pane. The Topology pane shows the racks in the cluster. Selecting a rack displays that rack's nodes in the Nodes pane to the right. Selecting **Cluster** displays all the nodes in the cluster.

Clicking any column name sorts data in ascending or descending order by that column.



□	Hlth	Hostname	Physical IP(s)	FS HB	TT HB	Physical Topology
□	🟩	qa-node101.qa.prv	10.10.100.101	0s ago	2s ago	/rack1/qa-node101.qa.prv
□	🟩	qa-node102.qa.prv	10.10.100.102	0s ago	2s ago	/rack2/qa-node102.qa.prv
□	🟩	qa-node103.qa.prv	10.10.100.103	0s ago	2s ago	/rack3/qa-node103.qa.prv
□	🟩	qa-node104.qa.prv	10.10.100.104	0s ago	2s ago	/rack4/qa-node104.qa.prv
□	🟩	qa-node105.qa.prv	10.10.100.105	0s ago	2s ago	/rack5/qa-node105.qa.prv
□	🟩	qa-node111.qa.prv	10.10.100.111	0s ago	2s ago	/rack1/qa-node111.qa.prv
□	🟩	qa-node112.qa.prv	10.10.100.112	0s ago	2s ago	/rack2/qa-node112.qa.prv
□	🟩	qa-node113.qa.prv	10.10.100.113	0s ago	2s ago	/rack3/qa-node113.qa.prv
□	🟩	qa-node114.qa.prv	10.10.100.114	0s ago	2s ago	/rack4/qa-node114.qa.prv
□	🟩	qa-node115.qa.prv	10.10.100.115	0s ago	2s ago	/rack5/qa-node115.qa.prv
□	🟩	qa-node121.qa.prv	10.10.100.121	0s ago	2s ago	/rack1/qa-node121.qa.prv
□	🟩	qa-node122.qa.prv	10.10.100.122	0s ago	2s ago	/rack2/qa-node122.qa.prv
□	🟩	qa-node123.qa.prv	10.10.100.123	0s ago	2s ago	/rack3/qa-node123.qa.prv
□	🟩	qa-node124.qa.prv	10.10.100.124	0s ago	2s ago	/rack4/qa-node124.qa.prv
□	🟩	qa-node125.qa.prv	10.10.100.125	0s ago	2s ago	/rack5/qa-node125.qa.prv

Selecting the checkboxes beside one or more nodes makes the following buttons available:

- **Manage Services** - displays the Manage Node Services dialog, which lets you start and stop services on the node.
- **Remove** - displays the Remove Node dialog, which lets you remove the node.
- **Change Topology** - displays the Change Node Topology dialog, which lets you change the topology path for a node.

Selecting the checkbox beside a single node makes the following button available:

- **Properties** - navigates to the Node Properties View, which displays detailed information about a single node.

The dropdown menu at the top left specifies the type of information to display:

- Overview - general information about each node
- Services - services running on each node
- Machine Performance - information about memory, CPU, I/O and RPC performance on each node
- Disks - information about disk usage, failed disks, and the MapR-FS heartbeat from each node
- MapReduce - information about the JobTracker heartbeat and TaskTracker slots on each node
- NFS Nodes - the IP addresses and Virtual IPs assigned to each NFS node
- Alarm Status - the status of alarms on each node

Clicking a node's Hostname navigates to the Node Properties View, which provides detailed information about the node.

Selecting the **Filter** checkbox displays the Filter toolbar, which provides additional data filtering options.

Overview

The Overview displays the following general information about nodes in the cluster:

- Hlth - each node's health: healthy, degraded, or critical
- Hostname - the hostname of each node
- Phys IP(s) - the IP address or addresses associated with each node
- FS HB - time since each node's last heartbeat to the CLDB
- JT HB - time since each node's last heartbeat to the JobTracker
- Physical Topology - the rack path to each node

Services

The Services view displays the following information about nodes in the cluster:

- Hlth - eact node's health: healthy, degraded, or critical
- Hostname - the hostname of eact node
- Services - a list of the services running on each node
- Physical Topology - each node's physical topology

Machine Performance

The Machine Performance view displays the following information about nodes in the cluster:

- Hlth - each node's health: healthy, degraded, or critical
- Hostname - the hostname of each node
- Memory - the percentage of memory used and the total memory
- # CPUs - the number of CPUs present on each node
- % CPU Idle - the percentage of CPU usage on each node
- Bytes Received - the network input
- Bytes Sent - the network output
- # RPCs - the number of RPC calls
- RPC In Bytes - the RPC input, in bytes
- RPC Out Bytes - the RPC output, in bytes
- # Disk Reads - the number of RPC disk reads
- # Disk Writes - the number of RPC disk writes
- Disk Read Bytes - the number of bytes read from disk
- Disk Write Bytes - the number of bytes written to disk
- # Disks - the number of disks present

Disks

The Disks view displays the following information about nodes in the cluster:

- Hlth - each node's health: healthy, degraded, or critical
- Hostname - the hostname of each node
- # bad Disks - the number of failed disks on each node
- Usage - the amount of disk used and total disk capacity, in gigabytes

MapReduce

The MapReduce view displays the following information about nodes in the cluster:

- Hlth - each node's health: healthy, degraded, or critical
- Hostname - the hostname of each node
- JT HB - the time since each node's most recent JobTracker heartbeat
- TT Map Slots - the number of map slots on each node
- TT Map Slots Used - the number of map slots in use on each node
- TT Reduce Slots - the number of reduce slots on each node
- TT Reduce Slots Used - the number of reduce slots in use on each node

NFS Nodes

The NFS Nodes view displays the following information about nodes in the cluster:

- Hlth - each node's health: healthy, degraded, or critical
- Hostname - the hostname of each node
- Phys IP(s) - the IP address or addresses associated with each node
- VIP(s) - the virtual IP address or addresses assigned to each node

Alarm Status

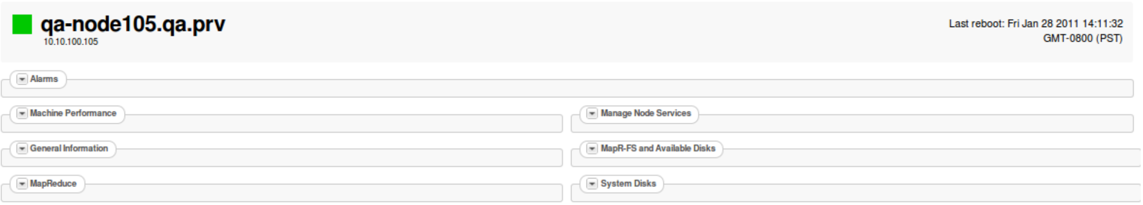
The Alarm Status view displays the following information about nodes in the cluster:

- Hlth - each node's health: healthy, degraded, or critical
- Hostname - the hostname of each node
- Version Alarm - whether the NODE_ALARM_VERSION_MISMATCH alarm is raised
- Excess Logs Alarm - whether the NODE_ALARM_DEBUG_LOGGING alarm is raised
- Disk Failure Alarm - whether the NODE_ALARM_DISK_FAILURE alarm is raised
- Time Skew Alarm - whether the NODE_ALARM_TIME_SKEW alarm is raised
- Root Partition Alarm - whether the NODE_ALARM_ROOT_PARTITION_FULL alarm is raised
- Installation Directory Alarm - whether the NODE_ALARM_OPT_MAPR_FULL alarm is raised
- Core Present Alarm - whether the NODE_ALARM_CORE_PRESENT alarm is raised
- CLDB Alarm - whether the NODE_ALARM_SERVICE_CLDB_DOWN alarm is raised
- FileServer Alarm - whether the NODE_ALARM_SERVICE_FILESERVER_DOWN alarm is raised
- JobTracker Alarm - whether the NODE_ALARM_SERVICE_JT_DOWN alarm is raised
- TaskTracker Alarm - whether the NODE_ALARM_SERVICE_TT_DOWN alarm is raised
- HBase Master Alarm - whether the NODE_ALARM_SERVICE_HBMASTER_DOWN alarm is raised
- HBase Region Alarm - whether the NODE_ALARM_SERVICE_HBREGION_DOWN alarm is raised
- NFS Gateway Alarm - whether the NODE_ALARM_SERVICE_NFS_DOWN alarm is raised
- WebServer Alarm - whether the NODE_ALARM_SERVICE_WEBSERVER_DOWN alarm is raised

Node Properties View

The Node Properties view displays detailed information about a single node in seven collapsible panes:

- Alarms
- Machine Performance
- General Information
- MapReduce
- Manage Node Services
- MapR-FS and Available Disks
- System Disks



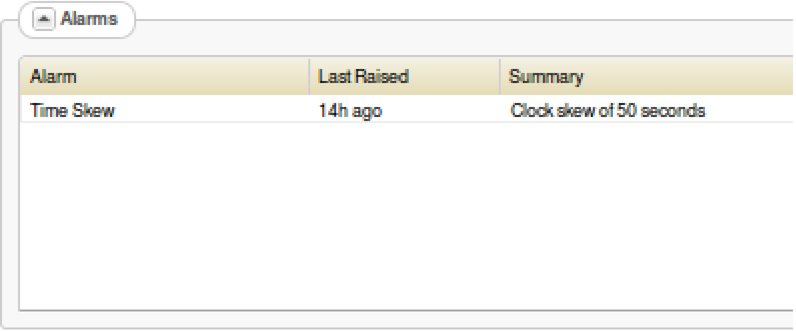
Buttons:

- **Remove Node** - displays the Remove Node dialog

Alarms

The Alarms pane displays a list of alarms that have been raised on the system, and the following information about each alarm:

- Alarm - the alarm name
- Last Raised - the most recent time when the alarm was raised
- Summary - a description of the alarm

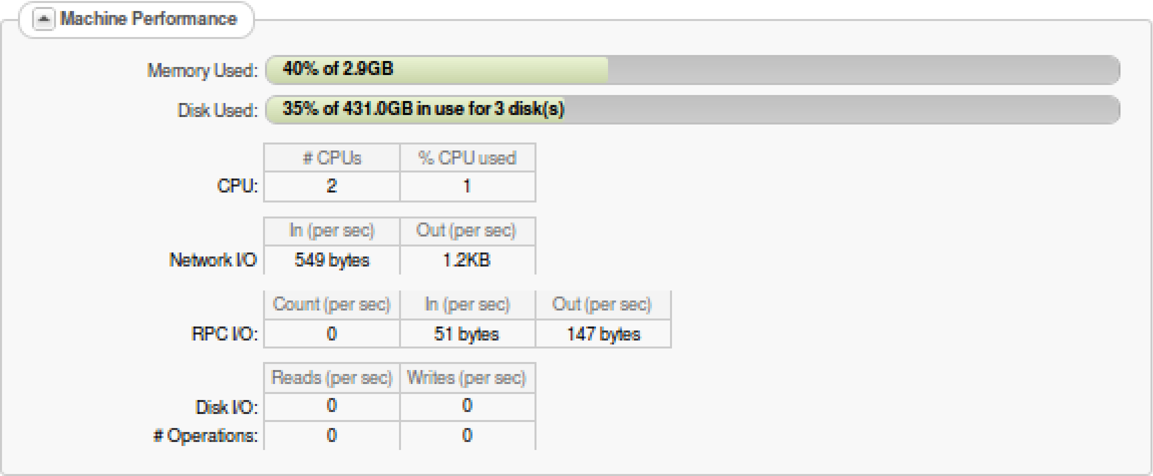


Alarm	Last Raised	Summary
Time Skew	14h ago	Clock skew of 50 seconds

Machine Performance

The Activity Since Last Heartbeat pane displays the following information about the node's performance and resource usage since it last reported to the CLDB:

- Memory Used - the amount of memory in use on the node
- Disk Used - the amount of disk space used on the node
- CPU - The number of CPUs and the percentage of CPU used on the node
- Network I/O - the input and output to the node per second
- RPC I/O - the number of RPC calls on the node and the amount of RPC input and output
- Disk I/O - the amount of data read to and written from the disk
- # Operations - the number of disk reads and writes



Machine Performance

Memory Used: 40% of 2.9GB

Disk Used: 35% of 431.0GB in use for 3 disk(s)

	# CPUs	% CPU used
CPU:	2	1

	In (per sec)	Out (per sec)
Network I/O	549 bytes	1.2KB

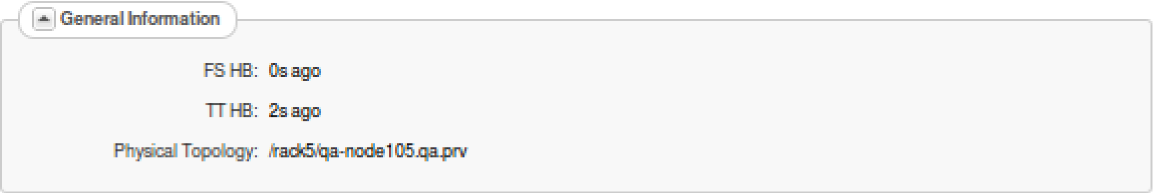
	Count (per sec)	In (per sec)	Out (per sec)
RPC I/O:	0	51 bytes	147 bytes

	Reads (per sec)	Writes (per sec)
Disk I/O:	0	0
# Operations:	0	0

General Information

The General Information pane displays the following general information about the node:

- FS HB - the amount of time since the node performed a heartbeat to the CLDB
- JT HB - the amount of time since the node performed a heartbeat to the JobTracker
- Physical Topology - the rack path to the node



General Information

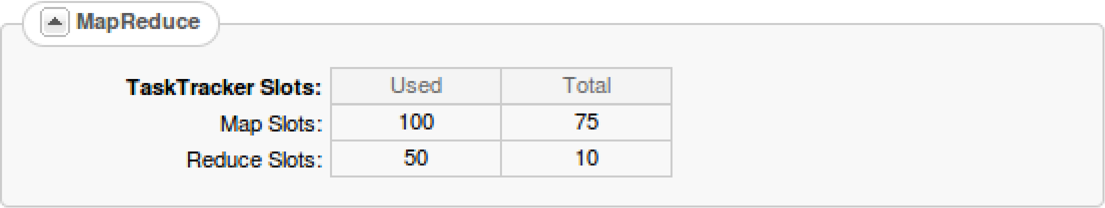
FS HB: 0s ago

TT HB: 2s ago

Physical Topology: /rack5/qa-node105.qa.prv

MapReduce

The MapReduce pane displays the number of map and reduce slots used, and the total number of map and reduce slots on the node.



MapReduce

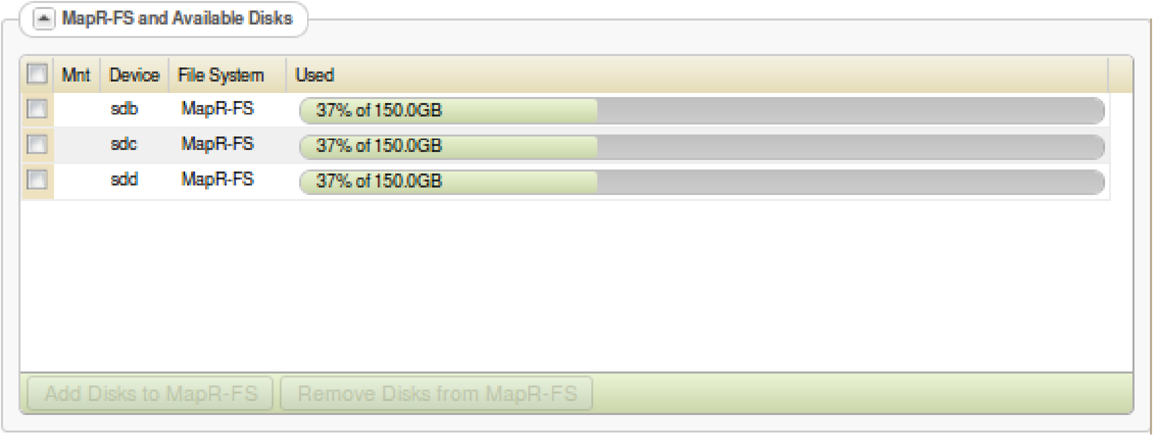
TaskTracker Slots:	Used	Total
Map Slots:	100	75
Reduce Slots:	50	10

MapR-FS and Available Disks

The MapR-FS and Available Disks pane displays the disks on the node, and the following information about each disk:

- Mnt - whether the disk is mounted or unmounted
- Disk - the disk name
- File System - the file system on the disk
- Used -the percentage used and total size of the disk

Clicking the checkbox next to a disk lets you select the disk for addition or removal.



□	Mnt	Device	File System	Used
□		sdb	MapR-FS	37% of 150.0GB
□		sdc	MapR-FS	37% of 150.0GB
□		sdd	MapR-FS	37% of 150.0GB

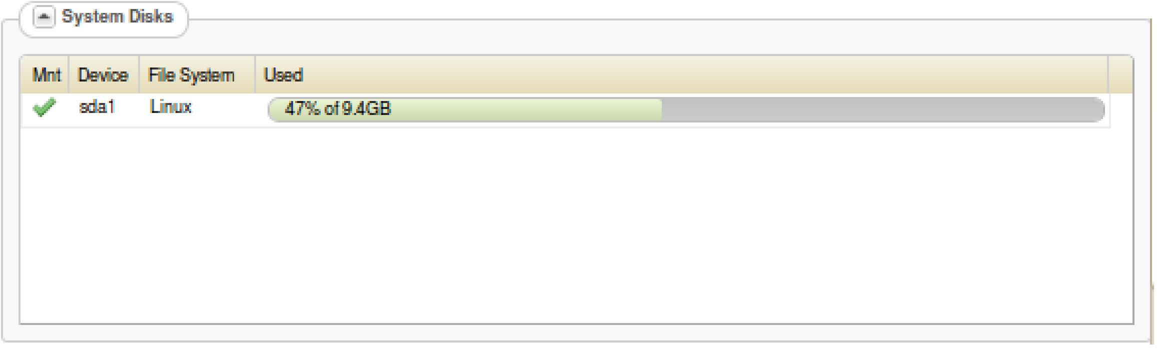
Buttons:

- **Add Disks to MapR-FS** - with one or more disks selected, adds the disks to the MapR-FS storage
- **Remove Disks from MapR-FS** with one or more disks selected, removes the disks from the MapR-FS storage

System Disks

The System Disks pane displays information about disks present and mounted on the node:

- Mnt - whether the disk is mounted
- Device - the device name of the disk
- File System - the file system
- Used - the percentage used and total capacity

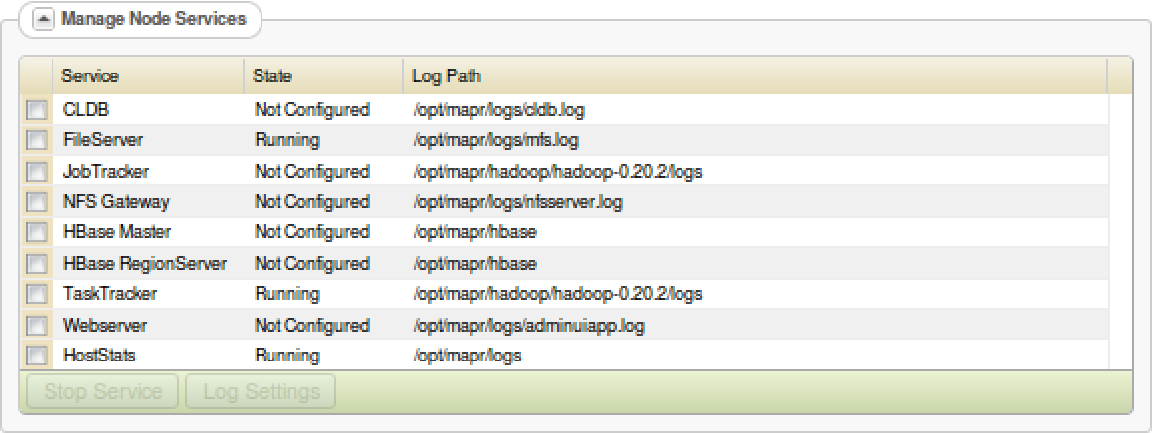


Mnt	Device	File System	Used
✓	sda1	Linux	47% of 9.4GB

Manage Node Services

The Manage Node Services pane displays the status of each service on the node:

- Service - the name of each service
- State:
  - 0 - NOT_CONFIGURED: the package for the service is not installed and/or the service is not configured (configure.sh has not run)
  - 2 - RUNNING: the service is installed, has been started by the warden, and is currently executing
  - 3 - STOPPED: the service is installed and `configure.sh` has run, but the service is currently not executing
- Log Path - the path where each service stores its logs



Manage Node Services

	Service	State	Log Path
□	CLDB	Not Configured	/opt/mapr/logs/cldb.log
□	FileServer	Running	/opt/mapr/logs/mfs.log
□	JobTracker	Not Configured	/opt/mapr/hadoop/hadoop-0.20.2/logs
□	NFS Gateway	Not Configured	/opt/mapr/logs/nfsserver.log
□	HBase Master	Not Configured	/opt/mapr/hbase
□	HBase RegionServer	Not Configured	/opt/mapr/hbase
□	TaskTracker	Running	/opt/mapr/hadoop/hadoop-0.20.2/logs
□	Webserver	Not Configured	/opt/mapr/logs/adminuiapp.log
□	HostStats	Running	/opt/mapr/logs

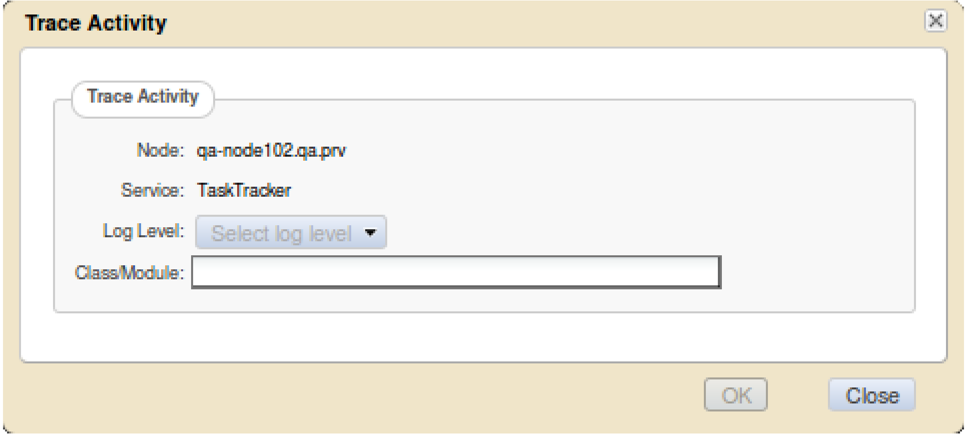
Buttons:

- **Start Service** - starts the selected services
- **Stop Service** - stops the selected services
- **Log Settings** - displays the Trace Activity dialog

You can also start and stop services in the the Manage Node Services dialog, by clicking **Manage Services** in the Nodes view.

Trace Activity

The Trace Activity dialog lets you set the log level of a specific service on a particular node.



Trace Activity

Node: qa-node102.qa.prv

Service: TaskTracker

Log Level: Select log level

Class/Module:

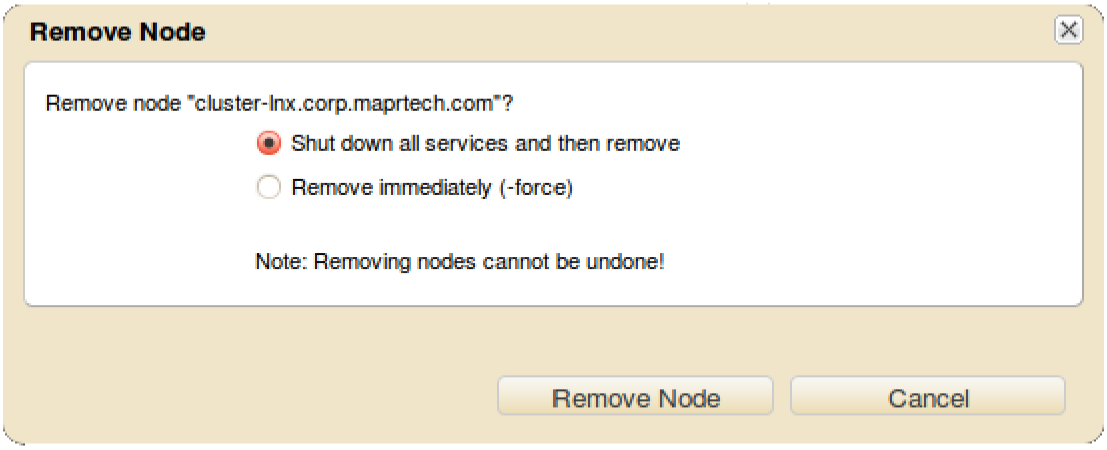
The **Log Level** dropdown specifies the logging threshold for messages.

Buttons:

- **OK** - save changes and exit
- **Close** - exit without saving changes

Remove Node

The Remove Node dialog lets you remove the specified node.



Remove Node

Remove node "cluster-lnx.corp.maprtech.com"?

● Shut down all services and then remove

○ Remove immediately (-force)

Note: Removing nodes cannot be undone!

The Remove Node dialog contains a radio button that lets you choose how to remove the node:

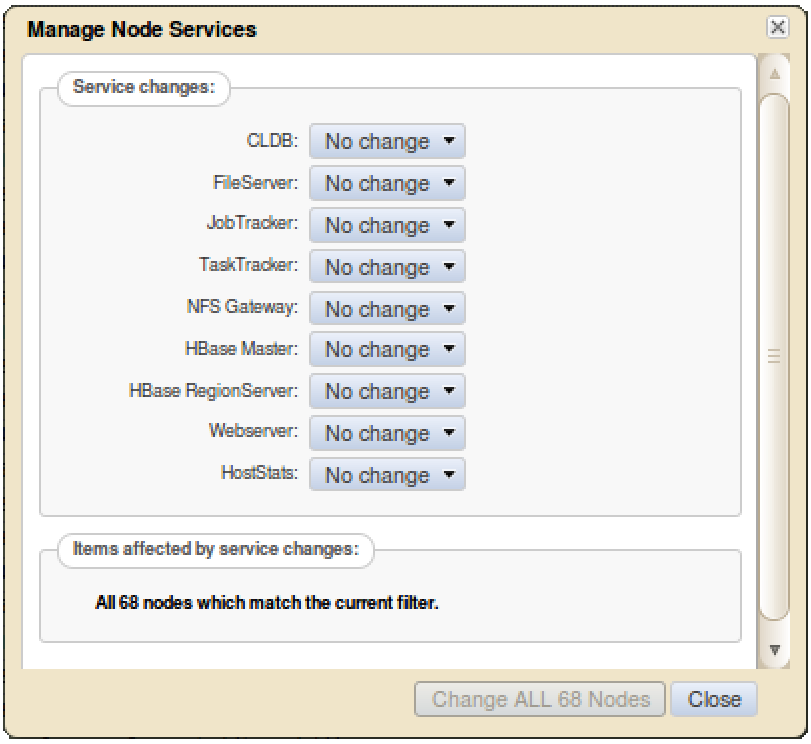
- **Shut down all services and then remove** - shut down services before removing the node
- **Remove immediately (-force)** - remove the node without shutting down services

Buttons:

- **Remove Node** - removes the node
- **Cancel** - returns to the Node Properties View without removing the node

Manage Node Services

The Manage Node Services dialog lets you start and stop services on the node.



Manage Node Services

Service changes:

CLDB:	No change
FileServer:	No change
JobTracker:	No change
TaskTracker:	No change
NFS Gateway:	No change
HBase Master:	No change
HBase RegionServer:	No change
Webserver:	No change
HostStats:	No change

Items affected by service changes:

All 68 nodes which match the current filter.

The Service Changes section contains a dropdown menu for each service:

- **No change** - leave the service running if it is running, or stopped if it is stopped
- **Start** - start the service
- **Stop** - stop the service

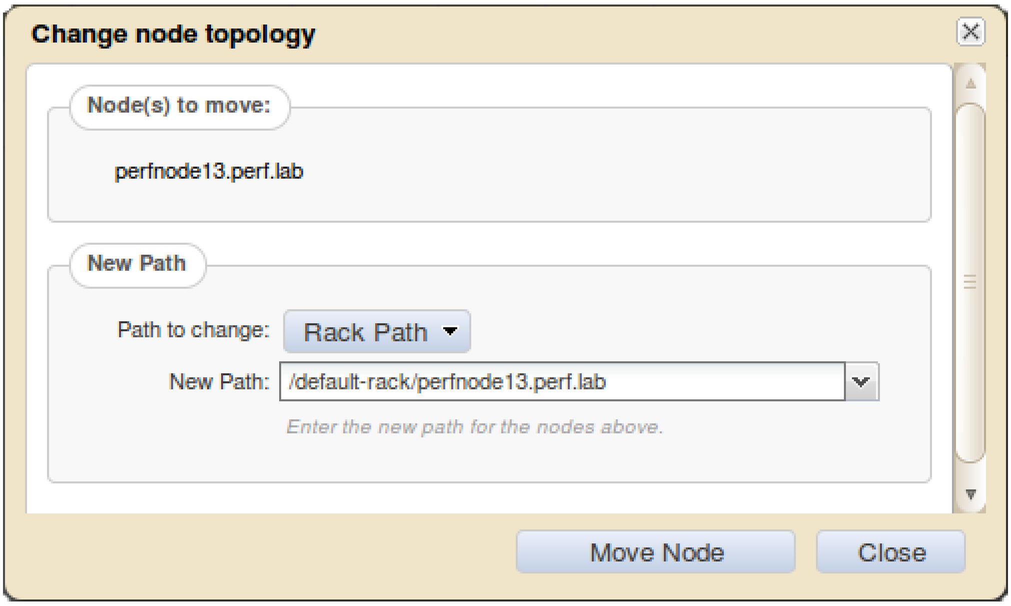
Buttons:

- **Change Node** - start and stop the selected services as specified by the dropdown menus
- **Cancel** - returns to the Node Properties View without starting or stopping any services

You can also start and stop services in the the Manage Node Services pane of the Node Properties view.

Change Node Topology

The Change Node Topology dialog lets you change the rack or switch path for one or more nodes.



Change node topology

Node(s) to move:

perfnode13.perf.lab

New Path

Path to change: Rack Path

New Path: /default-rack/perfnode13.perf.lab

Enter the new path for the nodes above.

The Change Node Topology dialog consists of two panes:

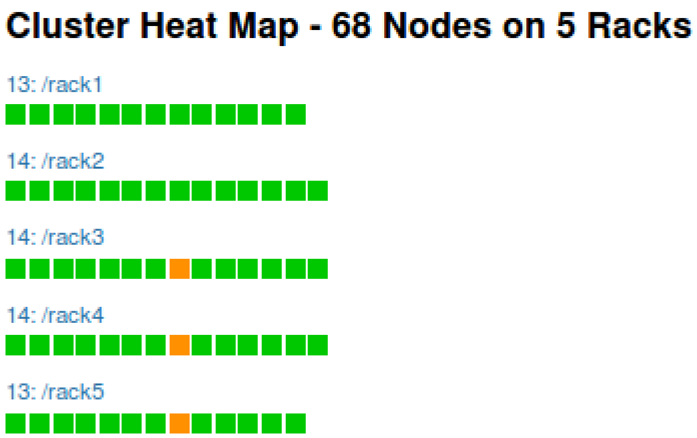
- Node(s) to move shows the node or nodes specified in the Nodes view.
- New Path contains the following fields:
    - **Path to Change** - rack path or switch path
    - **New Path** - the new node topology path

The Change Node Topology dialog contains the following buttons:

- **Move Node** - changes the node topology
- **Close** - returns to the Nodes view without changing the node topology

Node Heatmap

The Node Heatmap view displays information about each node, by rack



Cluster Heat Map - 68 Nodes on 5 Racks

13: /rack1

14: /rack2

14: /rack3

14: /rack4

13: /rack5

The dropdown menu above the heatmap lets you choose the type of information to display. See Cluster Heat Map.

Selecting the **Filter** checkbox displays the Filter toolbar, which provides additional data filtering options.

MapR-FS

The MapR-FS group provides the following views:

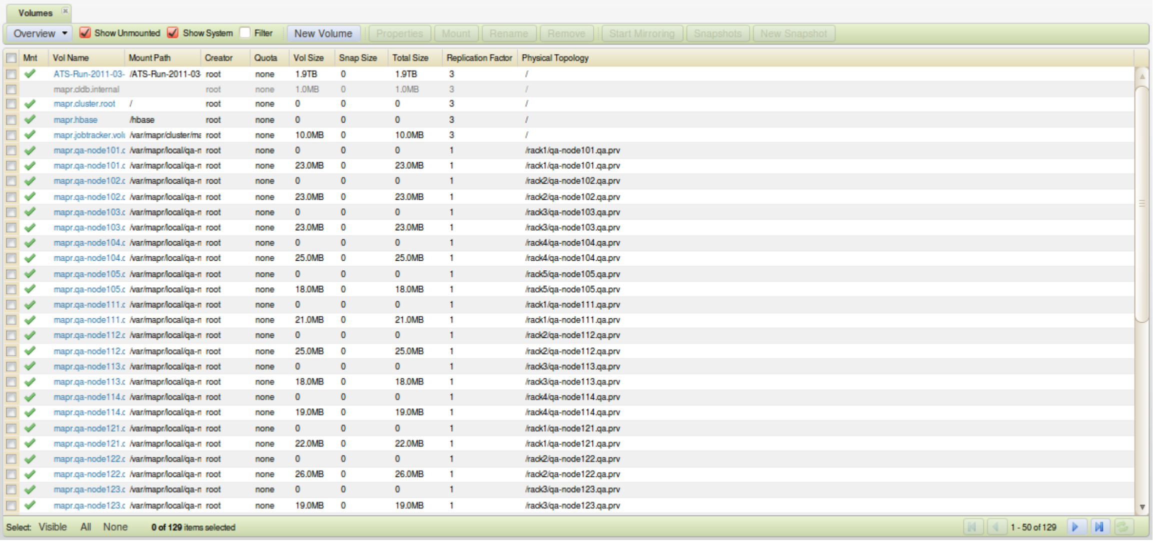
- Volumes - information about volumes in the cluster
- Mirror Volumes - information about mirrors
- User Disk Usage - cluster disk usage
- Snapshots - information about volume snapshots
- Schedules - information about schedules

Volumes

The Volumes view displays the following information about volumes in the cluster:

- Mnt - whether the volume is mounted (✓)
- Vol Name - the name of the volume
- Mount Path - the path where the volume is mounted
- Creator - the user or group that owns the volume
- Quota - the volume quota
- Vol Size - the size of the volume
- Snap Size - the size of the volume snapshot
- Total Size - the size of the volume and all its snapshots
- Replication Factor - the number of copies of the volume
- Physical Topology - the rack path to the volume

Clicking any column name sorts data in ascending or descending order by that column.



□	Mnt	Vol Name	Mount Path	Creator	Quota	Vol Size	Snap Size	Total Size	Replication Factor	Physical Topology
□	✓	ATS-Run-2011-03-	/ATS-Run-2011-03-	root	none	1.9TB	0	1.9TB	3	/
□		mapr.cldb.internal		root	none	1.0MB	0	1.0MB	3	/
□	✓	mapr.cluster.root	/	root	none	0	0	0	3	/
□	✓	mapr.hbase	/hbase	root	none	0	0	0	3	/
□	✓	mapr.jobtracker.volı	/var/mapr/cluster/ma	root	none	10.0MB	0	10.0MB	3	/
□	✓	mapr.qa-node101.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack1/qa-node101.qa.prv
□	✓	mapr.qa-node101.c	/var/mapr/local/qa-n	root	none	23.0MB	0	23.0MB	1	/rack1/qa-node101.qa.prv
□	✓	mapr.qa-node102.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack2/qa-node102.qa.prv
□	✓	mapr.qa-node102.c	/var/mapr/local/qa-n	root	none	23.0MB	0	23.0MB	1	/rack2/qa-node102.qa.prv
□	✓	mapr.qa-node103.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack3/qa-node103.qa.prv
□	✓	mapr.qa-node103.c	/var/mapr/local/qa-n	root	none	23.0MB	0	23.0MB	1	/rack3/qa-node103.qa.prv
□	✓	mapr.qa-node104.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack4/qa-node104.qa.prv
□	✓	mapr.qa-node104.c	/var/mapr/local/qa-n	root	none	25.0MB	0	25.0MB	1	/rack4/qa-node104.qa.prv
□	✓	mapr.qa-node105.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack5/qa-node105.qa.prv
□	✓	mapr.qa-node105.c	/var/mapr/local/qa-n	root	none	18.0MB	0	18.0MB	1	/rack5/qa-node105.qa.prv
□	✓	mapr.qa-node111.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack1/qa-node111.qa.prv
□	✓	mapr.qa-node111.c	/var/mapr/local/qa-n	root	none	21.0MB	0	21.0MB	1	/rack1/qa-node111.qa.prv
□	✓	mapr.qa-node112.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack2/qa-node112.qa.prv
□	✓	mapr.qa-node112.c	/var/mapr/local/qa-n	root	none	25.0MB	0	25.0MB	1	/rack2/qa-node112.qa.prv
□	✓	mapr.qa-node113.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack3/qa-node113.qa.prv
□	✓	mapr.qa-node113.c	/var/mapr/local/qa-n	root	none	18.0MB	0	18.0MB	1	/rack3/qa-node113.qa.prv
□	✓	mapr.qa-node114.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack4/qa-node114.qa.prv
□	✓	mapr.qa-node114.c	/var/mapr/local/qa-n	root	none	19.0MB	0	19.0MB	1	/rack4/qa-node114.qa.prv
□	✓	mapr.qa-node121.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack1/qa-node121.qa.prv
□	✓	mapr.qa-node121.c	/var/mapr/local/qa-n	root	none	22.0MB	0	22.0MB	1	/rack1/qa-node121.qa.prv
□	✓	mapr.qa-node122.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack2/qa-node122.qa.prv
□	✓	mapr.qa-node122.c	/var/mapr/local/qa-n	root	none	26.0MB	0	26.0MB	1	/rack2/qa-node122.qa.prv
□	✓	mapr.qa-node123.c	/var/mapr/local/qa-n	root	none	0	0	0	1	/rack3/qa-node123.qa.prv
□	✓	mapr.qa-node123.c	/var/mapr/local/qa-n	root	none	19.0MB	0	19.0MB	1	/rack3/qa-node123.qa.prv

The **Show Unmounted** checkbox specifies whether to show unmounted volumes:

- selected - show both mounted and unmounted volumes
- unselected - show mounted volumes only

The **Show System** checkbox specifies whether to show system volumes:

- selected - show both system and user volumes
- unselected - show user volumes only

Selecting the **Filter** checkbox displays the Filter toolbar, which provides additional data filtering options.

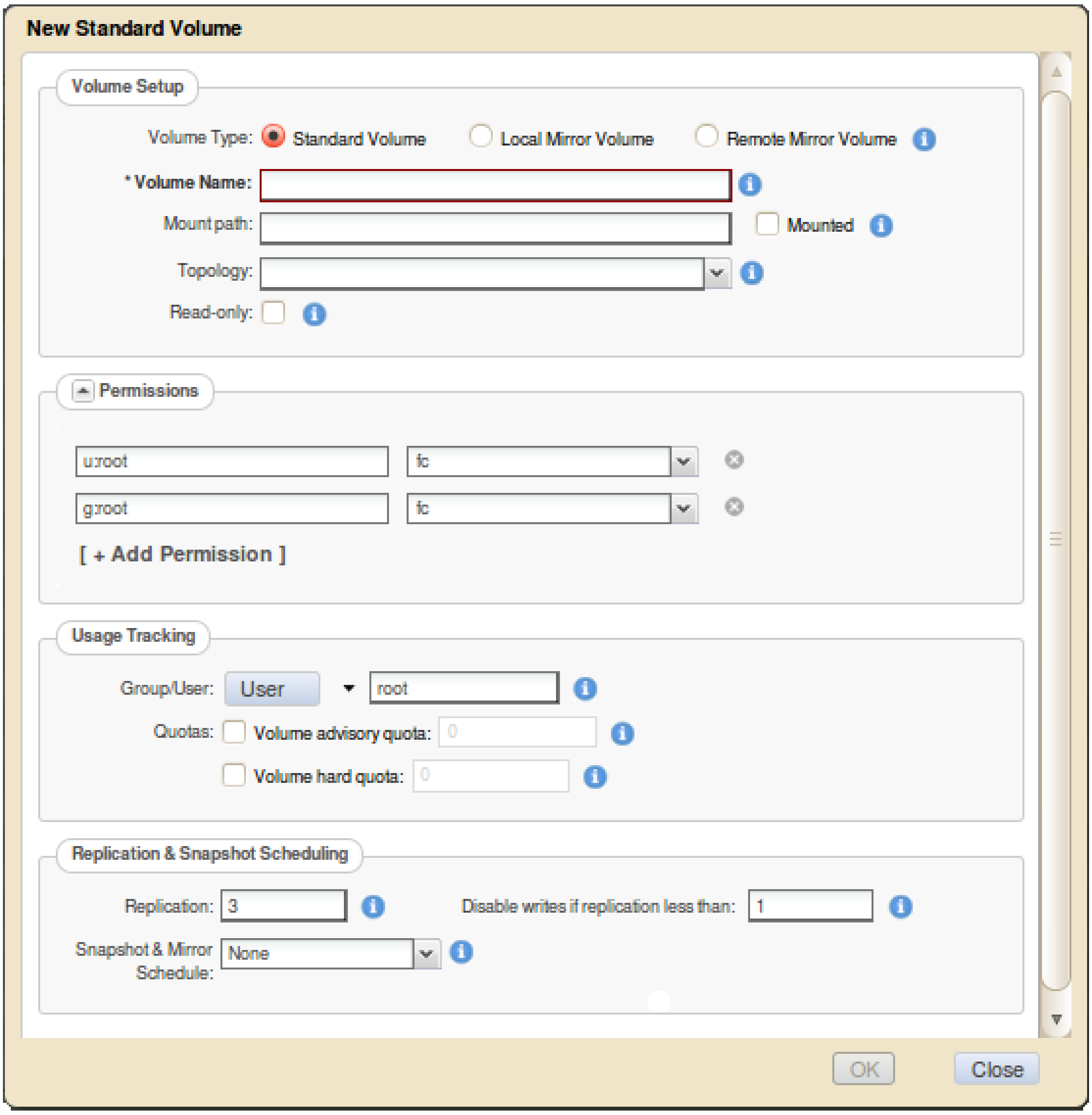
Clicking **New Volume** displays the New Volume dialog.

Selecting one or more checkboxes next to volumes enables the following buttons:

- **Remove** - displays the Remove Volume dialog
- **Properties** - displays the Volume Properties dialog (becomes **Edit X Volumes** if more than one checkbox is selected)
- **Snapshots** - displays the Snapshots for Volume dialog
- **New Snapshot** - displays the Snapshot Name dialog

New Volume

The New Volume dialog lets you create a new volume.



New Standard Volume

Volume Setup

Volume Type: ● Standard Volume ○ Local Mirror Volume ○ Remote Mirror Volume

* Volume Name:

Mount path: □ Mounted

Topology:

Read-only: □

Permissions

u:root	fc
g:root	fc

[+ Add Permission]

Usage Tracking

Group/User: User root

Quotas: □ Volume advisory quota: 0

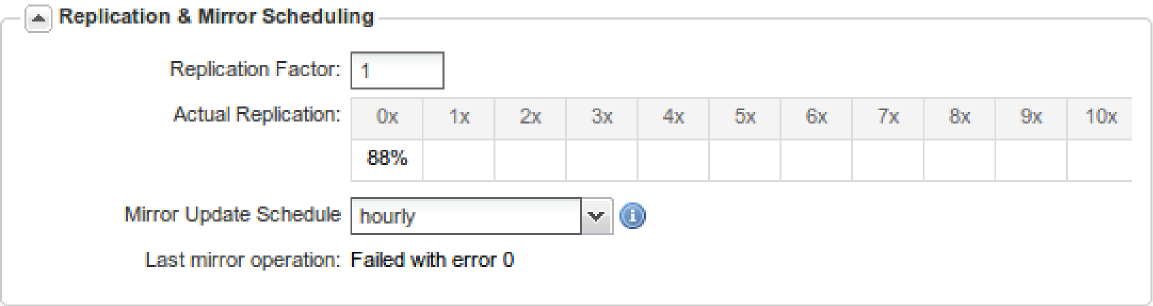
□ Volume hard quota: 0

Replication & Snapshot Scheduling

Replication: 3 Disable writes if replication less than: 1

Snapshot & Mirror Schedule: None

For mirror volumes, the Replication & Snapshot Scheduling section is replaced with a section called Replication & Mirror Scheduling:



Replication & Mirror Scheduling

Replication Factor: 1

Actual Replication:

0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	10x
88%										

Mirror Update Schedule hourly

Last mirror operation: Failed with error 0

The Volume Setup section specifies basic information about the volume using the following fields:

- **Volume Type** - a standard volume, or a local or remote mirror volume
- **Volume Name** (required) - a name for the new volume
- **Mount Path** - a path on which to mount the volume
- **Mounted** - whether the volume is mounted at creation
- **Topology** - the new volume's rack topology
- **Read-only** - if checked, prevents writes to the volume

The Ownership & Permissions section lets you grant specific permissions on the volume to certain users or groups:

- User/Group field - the user or group to which permissions are to be granted (one user or group per row)
- Permissions field - the permissions to grant to the user or group (see the Permissions table below)

- Delete button (⊗) - deletes the current row
- **[+ Add Permission]** - adds a new row

Volume Permissions

Code	Allowed Action
dump	Dump the volume
restore	Mirror or restore the volume
m	Modify volume properties, create and delete snapshots
d	Delete a volume
fc	Full control (admin access and permission to change volume ACL)

The Usage Tracking section sets the accountable entity and quotas for the volume using the following fields:

- **Group/User** - the group/user that is accountable for the volume
- Quotas - the volume quotas:
  - **Volume Advisory Quota** - if selected, the advisory quota for the volume as an integer plus a single letter to represent the unit
  - **Volume Quota** - if selected, the quota for the volume as an integer plus a single letter to represent the unit

The Replication & Snapshot Scheduling section (normal volumes) contains the following fields:

- **Replication** - the desired replication factor for the volume
- **Minimum Replication** - the minimum replication factor for the volume
- **Snapshot Schedule** - determines when snapshots will be automatically created; select an existing schedule from the pop-up menu

The Replication & Mirror Scheduling section (mirror volumes) contains the following fields:

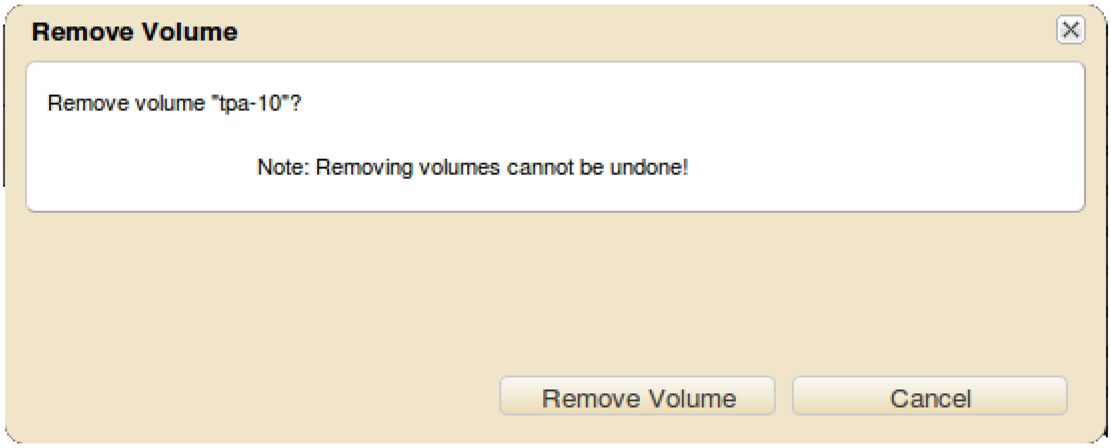
- **Replication Factor** - the desired replication factor for the volume
- **Actual Replication** - what percent of the volume data is replicated once (1x), twice (2x), and so on, respectively
- **Mirror Update Schedule** - determines when mirrors will be automatically updated; select an existing schedule from the pop-up menu
- **Last Mirror Operation** - the status of the most recent mirror operation.

Buttons:

- **Save** - creates the new volume
- **Close** - exits without creating the volume

Remove Volume

The Remove Volume dialog prompts you for confirmation before removing the specified volume or volumes.


Remove Volume

Remove volume "tpa-10"?

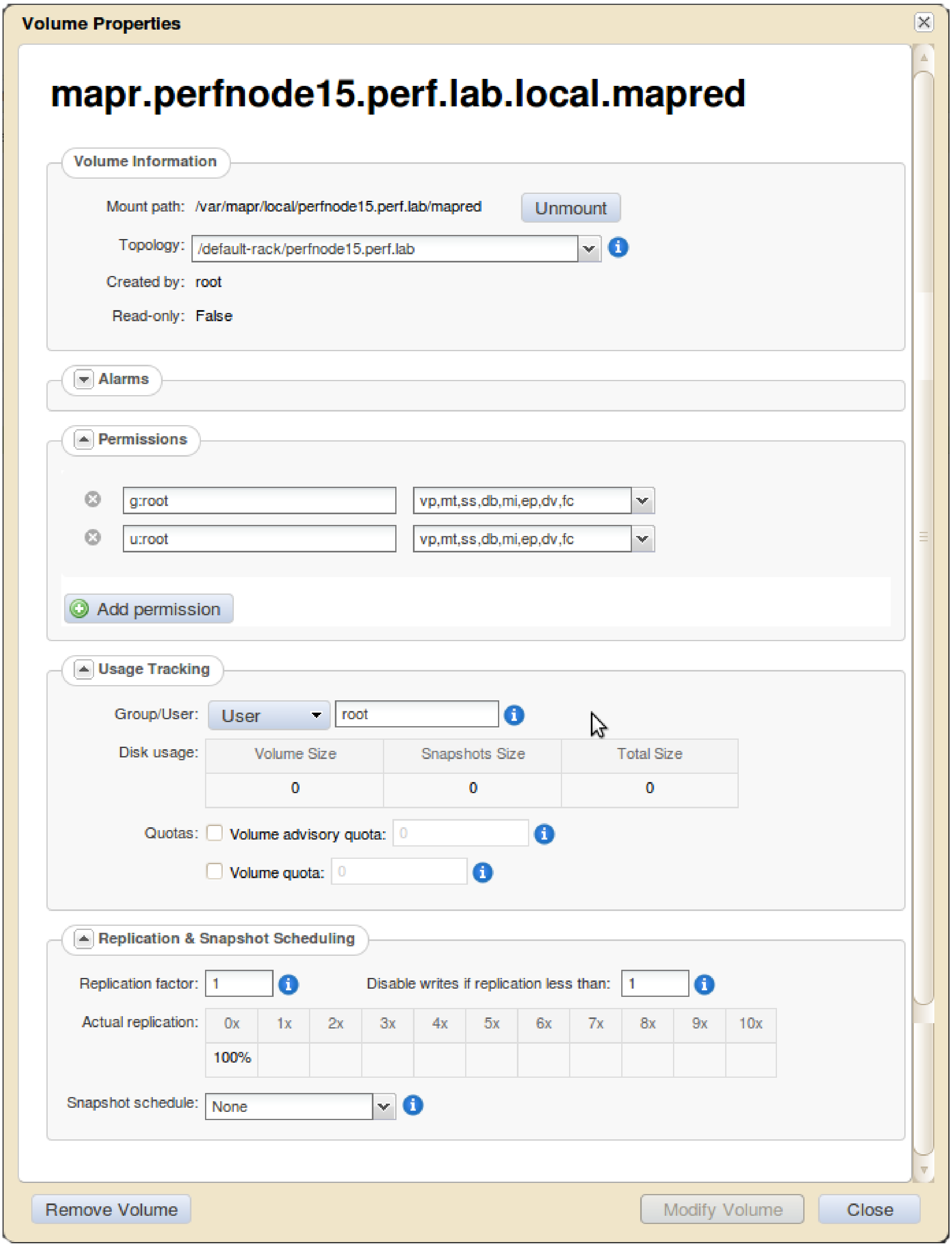
Note: Removing volumes cannot be undone!

Buttons:

- **Remove Volume** - removes the volume or volumes
- **Cancel** - exits without removing the volume or volumes

Volume Properties

The Volume Properties dialog lets you view and edit volume properties.



Volume Properties

mapr.perfnode15.perf.lab.local.mapred

Volume Information

Mount path: /var/mapr/local/perfnode15.perf.lab/mapred

Topology: /default-rack/perfnode15.perf.lab

Created by: root

Read-only: False

Alarms

Permissions

g:root	vp,mt,ss,db,mi,ep,dv,fc
u:root	vp,mt,ss,db,mi,ep,dv,fc

Usage Tracking

Group/User: User root

Disk usage:

Volume Size	Snapshots Size	Total Size
0	0	0

Quotas: □ Volume advisory quota: 0

□ Volume quota: 0

Replication & Snapshot Scheduling

Replication factor: 1

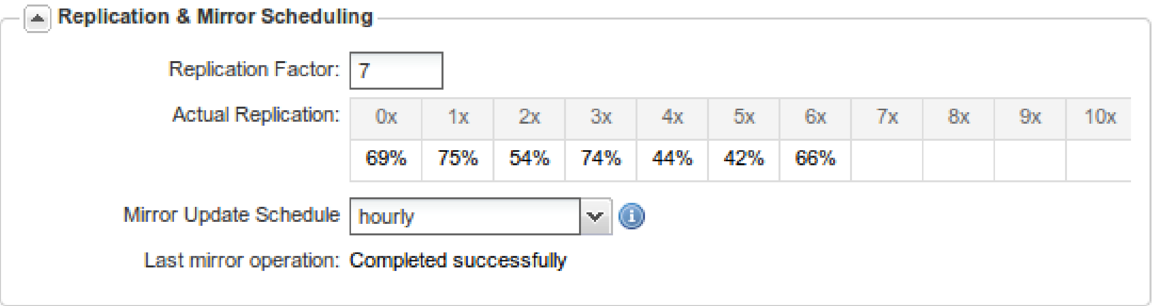
Disable writes if replication less than: 1

Actual replication:

0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	10x
100%										

Snapshot schedule: None

For mirror volumes, the Replication & Snapshot Scheduling section is replaced with a section called Replication & Mirror Scheduling:



Replication & Mirror Scheduling

Replication Factor: 7

Actual Replication:

0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	10x
69%	75%	54%	74%	44%	42%	66%				

Mirror Update Schedule: hourly

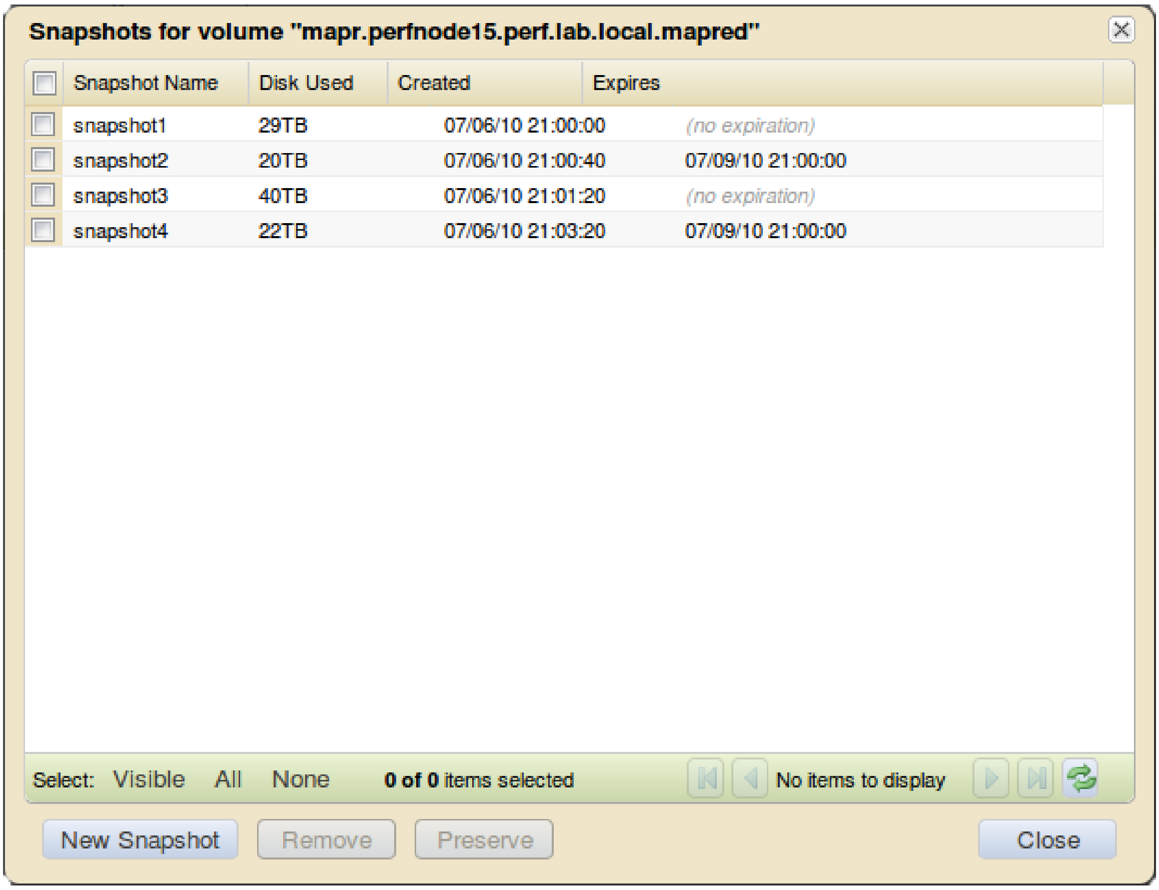
Last mirror operation: Completed successfully

For information about the fields in the Volume Properties dialog, see New Volume.

Snapshots for Volume

The Snapshots for Volume dialog displays the following information about snapshots for the specified volume:

- Snapshot Name - the name of the snapshot
- Disk Used - the disk space occupied by the snapshot
- Created - the date and time the snapshot was created
- Expires - the snapshot expiration date and time



Snapshots for volume "mapr.perfnode15.perf.lab.local.mapred"

□	Snapshot Name	Disk Used	Created	Expires
□	snapshot1	29TB	07/06/10 21:00:00	*(no expiration)*
□	snapshot2	20TB	07/06/10 21:00:40	07/09/10 21:00:00
□	snapshot3	40TB	07/06/10 21:01:20	*(no expiration)*
□	snapshot4	22TB	07/06/10 21:03:20	07/09/10 21:00:00

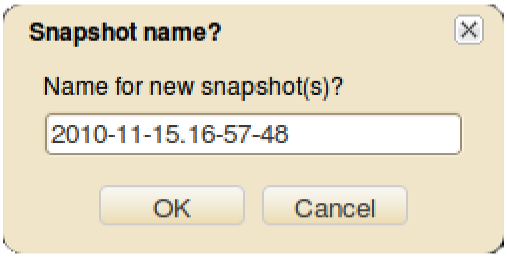
Select: Visible All None **0 of 0** items selected No items to display

Buttons:

- **New Snapshot** - displays the Snapshot Name dialog.
- **Remove** - when the checkboxes beside one or more snapshots are selected, displays the Remove Snapshots dialog
- **Preserve** - when the checkboxes beside one or more snapshots are selected, prevents the snapshots from expiring
- **Close** - closes the dialog

Snapshot Name

The Snapshot Name dialog lets you specify the name for a new snapshot you are creating.



Snapshot name?

Name for new snapshot(s)?

2010-11-15.16-57-48

The Snapshot Name dialog creates a new snapshot with the name specified in the following field:

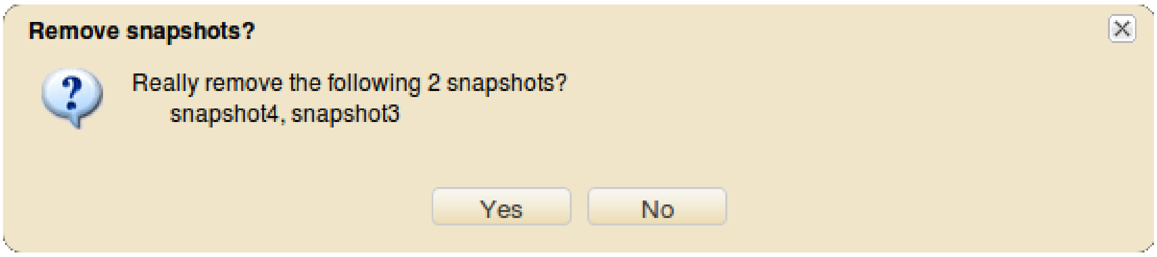
- **Name For New Snapshot(s)** - the new snapshot name

Buttons:

- **OK** - creates a snapshot with the specified name
- **Cancel** - exits without creating a snapshot

Remove Snapshots

The Remove Snapshots dialog prompts you for confirmation before removing the specified snapshot or snapshots.



Remove snapshots?

Really remove the following 2 snapshots?
snapshot4, snapshot3

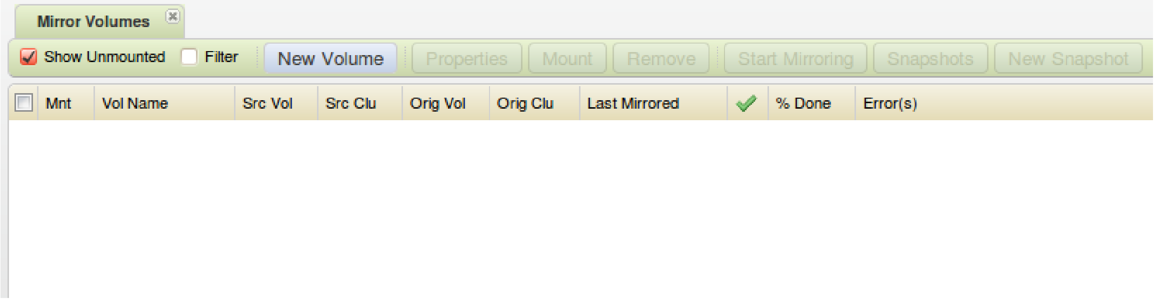
Buttons

- **Yes** - removes the snapshot or snapshots
- **No** - exits without removing the snapshot or snapshots

Mirror Volumes

The Mirror Volumes pane displays information about mirror volumes in the cluster:

- Mnt - whether the volume is mounted
- Vol Name - the name of the volume
- Src Vol - the source volume
- Src Clu - the source cluster
- Orig Vol -the originating volume for the data being mirrored
- Orig Clu - the originating cluster for the data being mirrored
- Last Mirrored - the time at which mirroring was most recently completed
- ✓ - status of the last mirroring operation
- % Done - progress of the mirroring operation
- Error(s) - any errors that occurred during the last mirroring operation

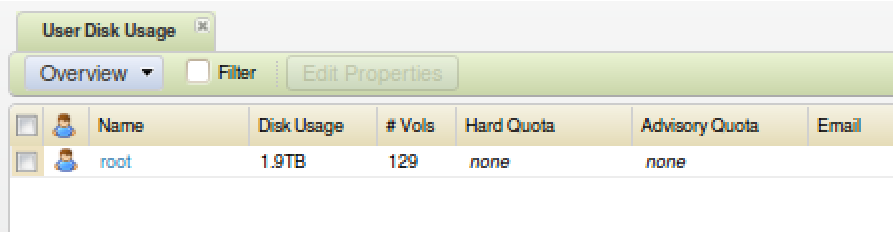


□	Mnt	Vol Name	Src Vol	Src Clu	Orig Vol	Orig Clu	Last Mirrored	✓	% Done	Error(s)

User Disk Usage

The User Disk Usage view displays information about disk usage by cluster users:

- Name - the username
- Disk Usage - the total disk space used by the user
- # Vols - the number of volumes
- Hard Quota - the user's quota
- Advisory Quota - the user's advisory quota
- Email - the user's email address



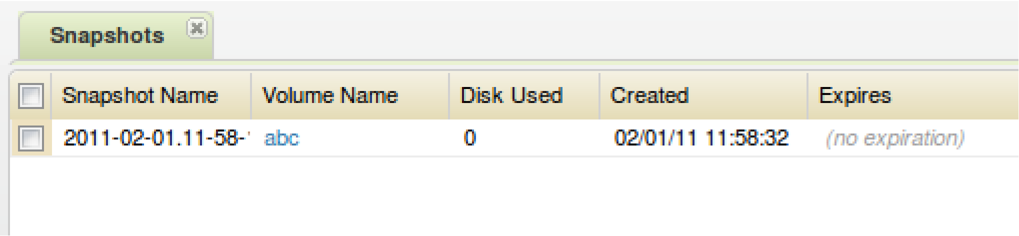
Name	Disk Usage	# Vols	Hard Quota	Advisory Quota	Email
root	1.9TB	129	*none*	*none*	

Snapshots

The Snapshots view displays the following information about volume snapshots in the cluster:

- Snapshot Name - the name of the snapshot
- Volume Name - the name of the source volume volume for the snapshot
- Disk Space used - the disk space occupied by the snapshot
- Created - the creation date and time of the snapshot
- Expires - the expiration date and time of the snapshot

Clicking any column name sorts data in ascending or descending order by that column.



Snapshot Name	Volume Name	Disk Used	Created	Expires
2011-02-01.11-58-'	abc	0	02/01/11 11:58:32	*(no expiration)*

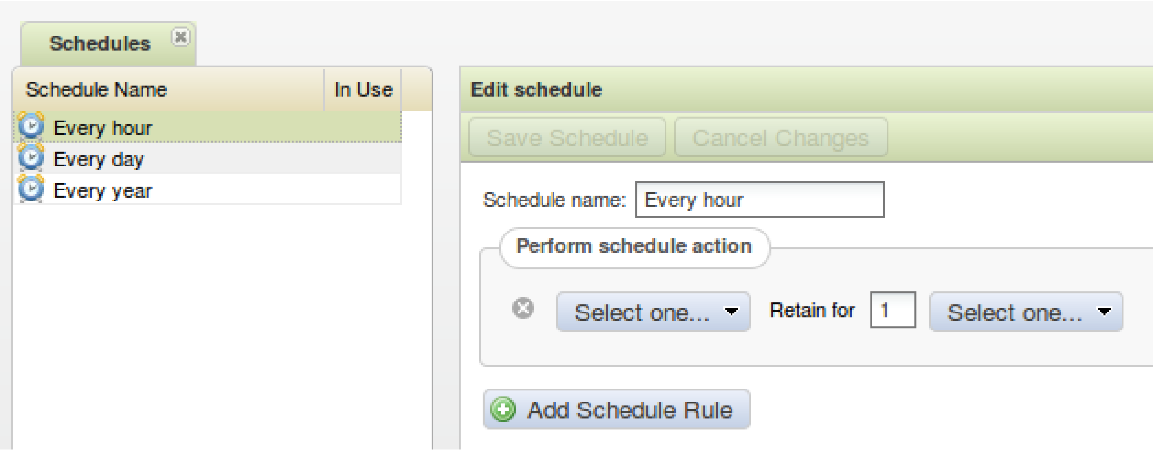
Selecting the **Filter** checkbox displays the Filter toolbar, which provides additional data filtering options.

Buttons:

- **Remove Snapshot** - when the checkboxes beside one or more snapshots are selected, displays the Remove Snapshots dialog
- **Preserve Snapshot** - when the checkboxes beside one or more snapshots are selected, prevents the snapshots from expiring

Schedules

The Schedules view lets you view and edit schedules, which can then can be attached to events to create occurrences. A schedule is a named group of rules that describe one or more points of time in the future at which an action can be specified to take place.



Every hour
Every day
Every year

Schedule name: Every hour

Retain for 1

The left pane of the Schedules view lists the following information about the existing schedules:

- Schedule Name - the name of the schedule; clicking a name displays the schedule details in the right pane for editing
- In Use - indicates whether the schedule is *in use* (✓), or attached to an action

The right pane provides the following tools for creating or editing schedules:

- **Schedule Name** - the name of the schedule
- **Schedule Rules** - specifies schedule rules with the following components:
  - A dropdown that specifies frequency (Once, Yearly, Monthly, Weekly, Daily, Hourly, Every X minutes)
  - Dropdowns that specify the time within the selected frequency
  - **Retain For** - the time for which the scheduled snapshot or mirror data is to be retained after creation
- **[+Add Rule]** - adds another rule to the schedule

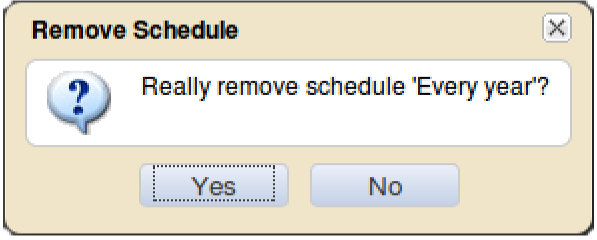
Navigating away from a schedule with unsaved changes displays the Save Schedule dialog.

Buttons:

- **New Schedule** - starts editing a new schedule
- **Remove Schedule** - displays the Remove Schedule dialog
- **Save Schedule** - saves changes to the current schedule
- **Cancel** - cancels changes to the current schedule

Remove Schedule

The Remove Schedule dialog prompts you for confirmation before removing the specified schedule.



Really remove schedule 'Every year'?

Buttons

- **Yes** - removes the schedule
- **No** - exits without removing the schedule

NFS HA

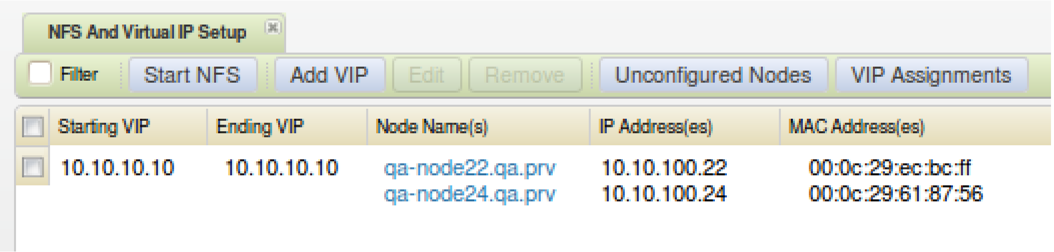
The NFS view group provides the following views:

- NFS Setup - information about NFS nodes in the cluster
- VIP Assignments - information about virtual IP addresses (VIPs) in the cluster
- NFS Nodes - information about NFS nodes in the cluster

NFS Setup

The NFS Setup view displays information about NFS nodes in the cluster and any VIPs assigned to them:

- Starting VIP - the starting IP of the VIP range
- Ending VIP - the ending IP of the VIP range
- Node Name(s) - the names of the NFS nodes
- IP Address(es) - the IP addresses of the NFS nodes
- MAC Address(es) - the MAC addresses associated with the IP addresses



□	Starting VIP	Ending VIP	Node Name(s)	IP Address(es)	MAC Address(es)
□	10.10.10.10	10.10.10.10	qa-node22.qa.prv qa-node24.qa.prv	10.10.100.22 10.10.100.24	00:0c:29:ec:bc:ff 00:0c:29:61:87:56

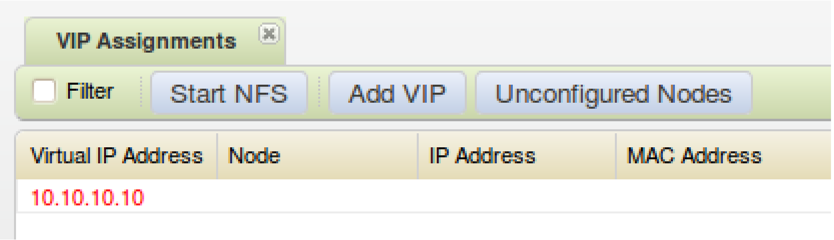
Buttons:

- **Start NFS** - displays the Manage Node Services dialog
- **Add VIP** - displays the Add Virtual IPs dialog
- **Edit** - when one or more checkboxes are selected, edits the specified VIP ranges
- **Remove**- when one or more checkboxes are selected, removes the specified VIP ranges
- **Unconfigured** Nodes - displays nodes not running the NFS service (in the Nodes view)
- **VIP Assignments** - displays the VIP Assignments view

VIP Assignments

The VIP Assignments view displays VIP assignments beside the nodes to which they are assigned:

- Virtual IP Address - each VIP in the range
- Node Name - the node to which the VIP is assigned
- IP Address - the IP address of the node
- MAC Address - the MAC address associated with the IP address



Virtual IP Address	Node	IP Address	MAC Address
10.10.10.10			

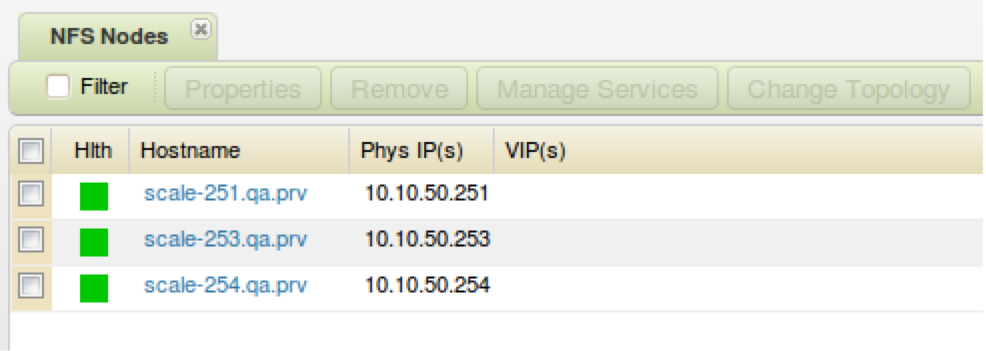
Buttons:

- **Start NFS** - displays the Manage Node Services dialog
- **Add VIP** - displays the Add Virtual IPs dialog
- **Unconfigured Nodes** - displays nodes not running the NFS service (in the Nodes view)

NFS Nodes

The NFS Nodes view displays information about nodes running the NFS service:

- Hlth - the health of the node
- Hostname - the hostname of the node
- Phys IP(s) - physical IP addresses associated with the node
- VIP(s) - virtual IP addresses associated with the node



☐	Hlth	Hostname	Phys IP(s)	VIP(s)
☐	🟩	scale-251.qa.prv	10.10.50.251	
☐	🟩	scale-253.qa.prv	10.10.50.253	
☐	🟩	scale-254.qa.prv	10.10.50.254	

Buttons:

- **Properties** - when one or more nodes are selected, navigates to the Node Properties View
- **Manage Services** - navigates to the Manage Node Services dialog, which lets you start and stop services on the node
- **Remove** - navigates to the Remove Node dialog, which lets you remove the node
- **Change Topology** - navigates to the Change Node Topology dialog, which lets you change the rack or switch path for a node

Alarms

The Alarms view group provides the following views:

- Node Alarms - information about node alarms in the cluster
- Volume Alarms - information about volume alarms in the cluster
- User/Group Alarms - information about users or groups that have exceeded quotas
- Alarm Notifications - configure where notifications are sent when alarms are raised

Node Alarms

The Node Alarms view displays information about node alarms in the cluster.

- Hlth - a color indicating the status of each node (see Cluster Heat Map)
- Hostname - the hostname of the node
- Version Alarm - last occurrence of the NODE_ALARM_VERSION_MISMATCH alarm
- Excess Logs Alarm - last occurrence of the NODE_ALARM_DEBUG_LOGGING alarm
- Disk Failure Alarm - of the NODE_ALARM_DISK_FAILURE alarm
- Time Skew Alarm - last occurrence of the NODE_ALARM_ TIME_SKEW alarm
- Root Partition Alarm - last occurrence of the NODE_ALARM_ROOT_PARTITION_FULL alarm
- Installation Directory Alarm - last occurrence of the NODE_ALARM_OPT_MAPR_FULL alarm
- Core Present Alarm - last occurrence of the NODE_ALARM_CORE_PRESENT alarm
- CLDB Alarm - last occurrence of the NODE_ALARM_SERVICE_CLDB_DOWN alarm
- FileServer Alarm - last occurrence of the NODE_ALARM_SERVICE_FILESERVER_DOWN alarm
- JobTracker Alarm - last occurrence of the NODE_ALARM_SERVICE_JT_DOWN alarm
- TaskTracker Alarm - last occurrence of the NODE_ALARM_SERVICE_TT_DOWN alarm
- HBase Master Alarm - last occurrence of the NODE_ALARM_SERVICE_HBMASTER_DOWN alarm
- HBase Regionserver Alarm - last occurrence of the NODE_ALARM_SERVICE_HBREGION_DOWN alarm
- NFS Gateway Alarm - last occurrence of the NODE_ALARM_SERVICE_NFS_DOWN alarm
- WebServer Alarm - last occurrence of the NODE_ALARM_SERVICE_WEBSERVER_DOWN alarm
- Hoststats Alarm - last occurrence of the NODE_ALARM_SERVICE_HOSTSTATS_DOWN alarm

See Troubleshooting Alarms.

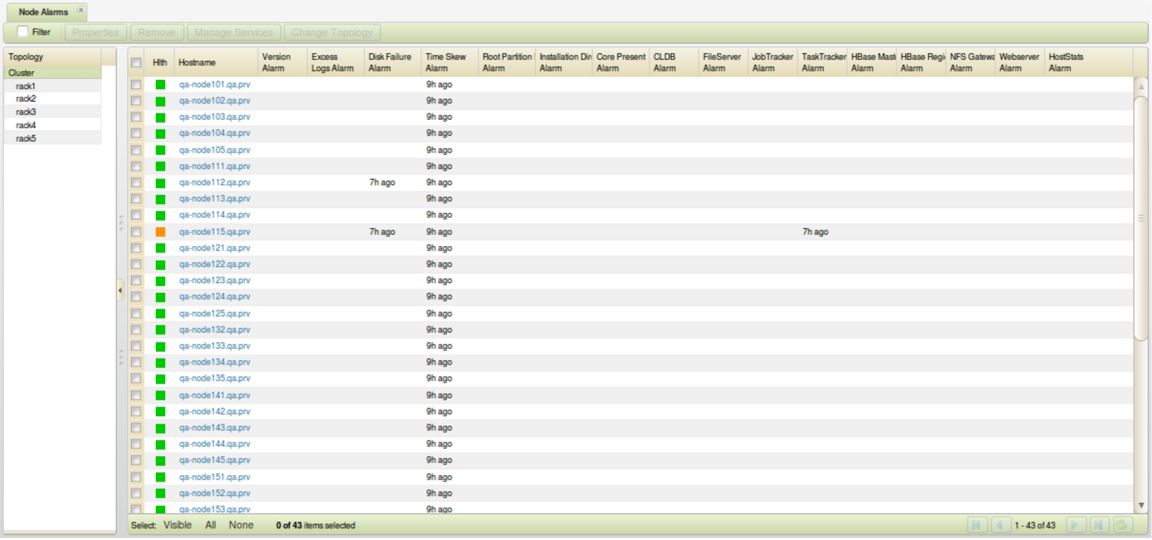
Clicking any column name sorts data in ascending or descending order by that column.

The left pane of the Node Alarms view displays the following information about the cluster:

- Topology - the rack topology of the cluster

Selecting the **Filter** checkbox displays the Filter toolbar, which provides additional data filtering options.

Clicking a node's Hostname navigates to the Node Properties View, which provides detailed information about the node.



□	Hlth	Hostname	Version Alarm	Excess Logs Alarm	Disk Failure Alarm	Time Skew Alarm	Root Partition I Alarm	Installation Dir Alarm	Core Present Alarm	CLDB Alarm	FileServer Alarm	JobTracker Alarm	TaskTracker Alarm	HBase Mast Alarm	HBase Regi Alarm	NFS Gatewa Alarm	Webserver Alarm	HostStats Alarm
□	🟩	qa-node101.qa.prv				9h ago												
□	🟩	qa-node102.qa.prv				9h ago												
□	🟩	qa-node103.qa.prv				9h ago												
□	🟩	qa-node104.qa.prv				9h ago												
□	🟩	qa-node105.qa.prv				9h ago												
□	🟩	qa-node111.qa.prv				9h ago												
□	🟩	qa-node112.qa.prv			7h ago	9h ago												
□	🟩	qa-node113.qa.prv				9h ago												
□	🟩	qa-node114.qa.prv				9h ago												
□	🟧	qa-node115.qa.prv			7h ago	9h ago							7h ago					
□	🟩	qa-node121.qa.prv				9h ago												
□	🟩	qa-node122.qa.prv				9h ago												
□	🟩	qa-node123.qa.prv				9h ago												
□	🟩	qa-node124.qa.prv				9h ago												
□	🟩	qa-node125.qa.prv				9h ago												
□	🟩	qa-node132.qa.prv				9h ago												
□	🟩	qa-node133.qa.prv				9h ago												
□	🟩	qa-node134.qa.prv				9h ago												
□	🟩	qa-node135.qa.prv				9h ago												
□	🟩	qa-node141.qa.prv				9h ago												
□	🟩	qa-node142.qa.prv				9h ago												
□	🟩	qa-node143.qa.prv				9h ago												
□	🟩	qa-node144.qa.prv				9h ago												
□	🟩	qa-node145.qa.prv				9h ago												
□	🟩	qa-node151.qa.prv				9h ago												
□	🟩	qa-node152.qa.prv				9h ago												
□	🟩	qa-node153.qa.prv				9h ago												

Buttons:

- **Properties** - navigates to the Node Properties View
- **Remove** - navigates to the Remove Node dialog, which lets you remove the node
- **Manage Services** - navigates to the Manage Node Services dialog, which lets you start and stop services on the node
- **Change Topology** - navigates to the Change Node Topology dialog, which lets you change the rack or switch path for a node

Volume Alarms

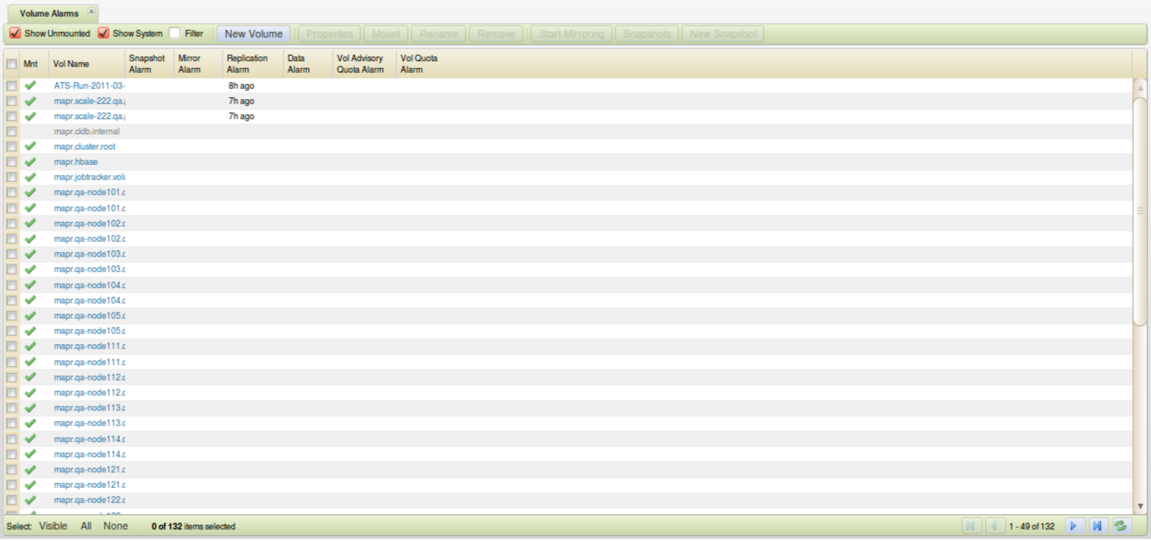
The Volume Alarms view displays information about volume alarms in the cluster:

- Mnt - whether the volume is mounted
- Vol Name - the name of the volume
- Snapshot Alarm - last Snapshot Failed alarm
- Mirror Alarm - last Mirror Failed alarm
- Replication Alarm - last Data Under-Replicated alarm
- Data Alarm - last Data Unavailable alarm
- Vol Advisory Quota Alarm - last Volume Advisory Quota Exceeded alarm
- Vol Quota Alarm- last Volume Quota Exceeded alarm

Clicking any column name sorts data in ascending or descending order by that column. Clicking a volume name displays the Volume Properties dialog

Selecting the **Show Unmounted** checkbox shows unmounted volumes as well as mounted volumes.

Selecting the **Filter** checkbox displays the Filter toolbar, which provides additional data filtering options.



□	Mnt	Vol Name	Snapshot Alarm	Mirror Alarm	Replication Alarm	Data Alarm	Vol Advisory Quota Alarm	Vol Quota Alarm
□	✓	ATS-Run-2011-03-			8h ago			
□	✓	mapr.scale-222.qa.			7h ago			
□	✓	mapr.scale-222.qa.			7h ago			
□		mapr.cldb.internal						
□	✓	mapr.cluster.root						
□	✓	mapr.hbase						
□	✓	mapr.jobtracker.vol						
□	✓	mapr.qa-node101.c						
□	✓	mapr.qa-node101.c						
□	✓	mapr.qa-node102.c						
□	✓	mapr.qa-node102.c						
□	✓	mapr.qa-node103.c						
□	✓	mapr.qa-node103.c						
□	✓	mapr.qa-node104.c						
□	✓	mapr.qa-node104.c						
□	✓	mapr.qa-node105.c						
□	✓	mapr.qa-node105.c						
□	✓	mapr.qa-node111.c						
□	✓	mapr.qa-node111.c						
□	✓	mapr.qa-node112.c						
□	✓	mapr.qa-node112.c						
□	✓	mapr.qa-node113.c						
□	✓	mapr.qa-node113.c						
□	✓	mapr.qa-node114.c						
□	✓	mapr.qa-node114.c						
□	✓	mapr.qa-node121.c						
□	✓	mapr.qa-node121.c						
□	✓	mapr.qa-node122.c						

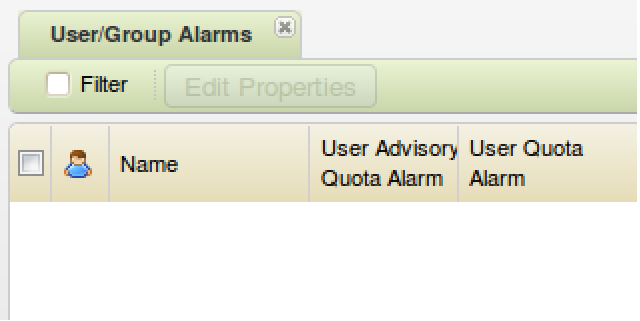
Buttons:

- **New Volume** displays the New Volume Dialog.
- **Properties** - if the checkboxes beside one or more volumes is selected,displays the Volume Properties dialog
- **Mount (Unmount)** - if an unmounted volume is selected, mounts it; if a mounted volume is selected, unmounts it
- **Remove** - if the checkboxes beside one or more volumes is selected, displays the Remove Volume dialog
- **Start Mirroring** - if a mirror volume is selected, starts the mirror sync process
- **Snapshots** - if the checkboxes beside one or more volumes is selected,displays the Snapshots for Volume dialog
- **New Snapshot** - if the checkboxes beside one or more volumes is selected,displays the Snapshot Name dialog

User/Group Alarms

The User/Group Alarms view displays information about user and group quota alarms in the cluster:

- Name - the name of the user or group
- User Advisory Quota Alarm - the last Advisory Quota Exceeded alarm
- User Quota Alarm - the last Quota Exceeded alarm



Buttons:

- **Edit Properties**

Alarm Notifications

The Configure Global Alarm Notifications dialog lets you specify where email notifications are sent when alarms are raised.

Fields:

- **Alarm Name** - select the alarm to configure
- **Standard Notification** - send notification to the default for the alarm type (the cluster administrator or volume creator, for example)
- **Additional Email Address** - specify an additional custom email address to receive notifications for the alarm type

Buttons:

- **Save** - save changes and exit
- **Close** - exit without saving changes

System Settings

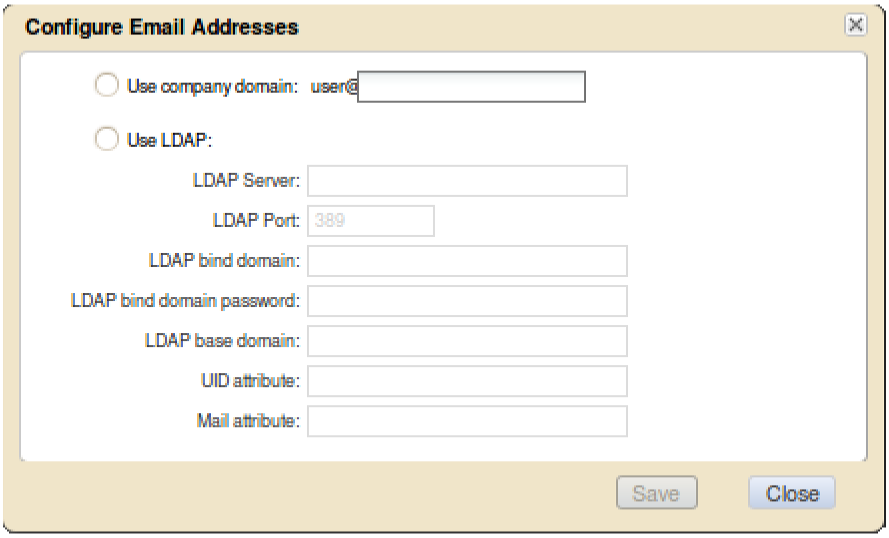
The System Settings view group provides the following views:

- Email Addresses - specify Greenplum HD EE user email addresses
- Permissions - give permissions to users
- Quota Defaults - settings for default quotas in the cluster
- SMTP - settings for sending email from Greenplum HD EE
- HTTP - settings for accessing the Greenplum HD EE Control System via a browser
- Greenplum HD EE Licenses - Greenplum HD EE license settings

Email Addresses

The Configure Email Addresses dialog lets you specify whether Greenplum HD EE gets user email addresses from an LDAP directory, or uses a company domain:

- **Use Company Domain** - specify a domain to append after each username to determine each user's email address
- **Use LDAP** - obtain each user's email address from an LDAP server



Configure Email Addresses

○ Use company domain: user@

○ Use LDAP:

LDAP Server:

LDAP Port: 389

LDAP bind domain:

LDAP bind domain password:

LDAP base domain:

UID attribute:

Mail attribute:

Buttons:

- **Save** - save changes and exit
- **Close** - exit without saving changes

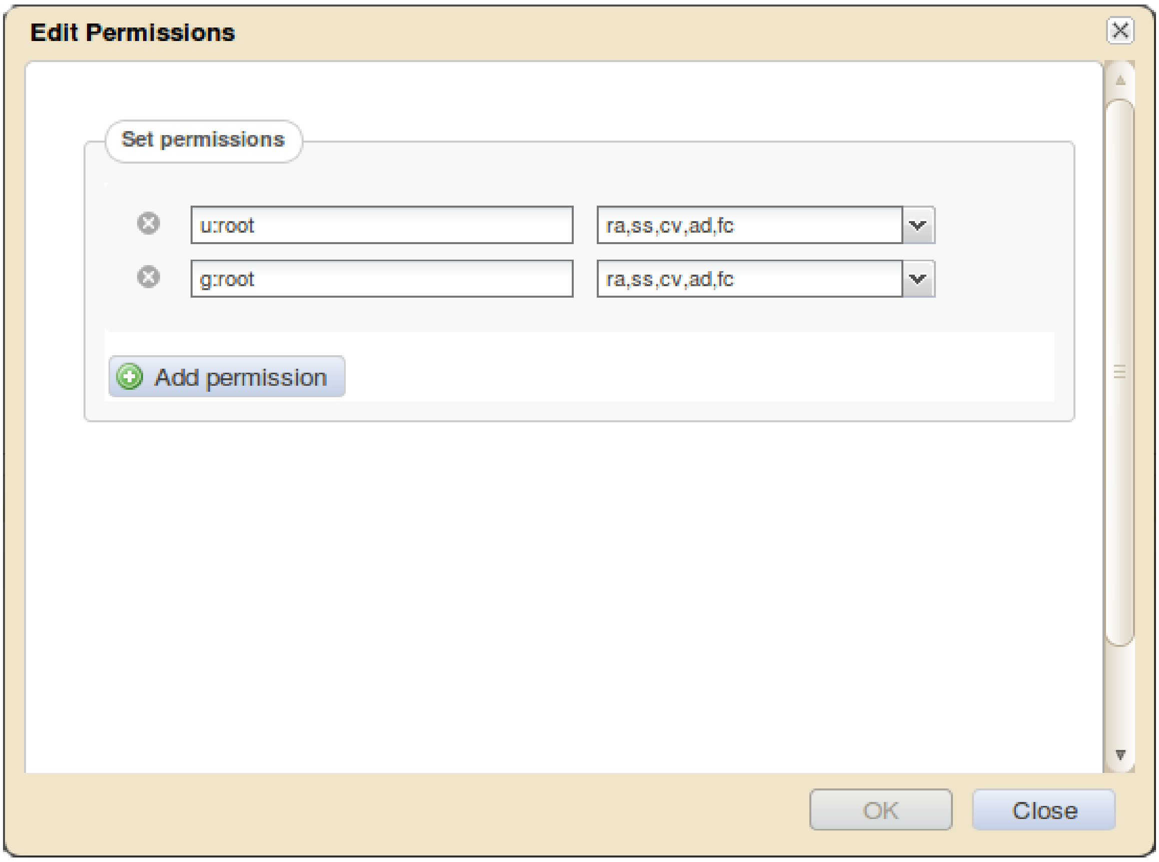
Permissions

The Edit Permissions dialog lets you grant specific cluster permissions to particular users and groups.

- User/Group field - the user or group to which permissions are to be granted (one user or group per row)
- Permissions field - the permissions to grant to the user or group (see the Permissions table below)
- Delete button (⊗) - deletes the current row
- **[+ Add Permission]** - adds a new row

Cluster Permissions

Code	Allowed Action	Includes
login	Log in to the Greenplum HD EE Control System, use the API and command-line interface, read access on cluster and volumes	cv
ss	Start/stop services	
cv	Create volumes	
a	Admin access	All permissions except fc
fc	Full control (administrative access and permission to change the cluster ACL)	a

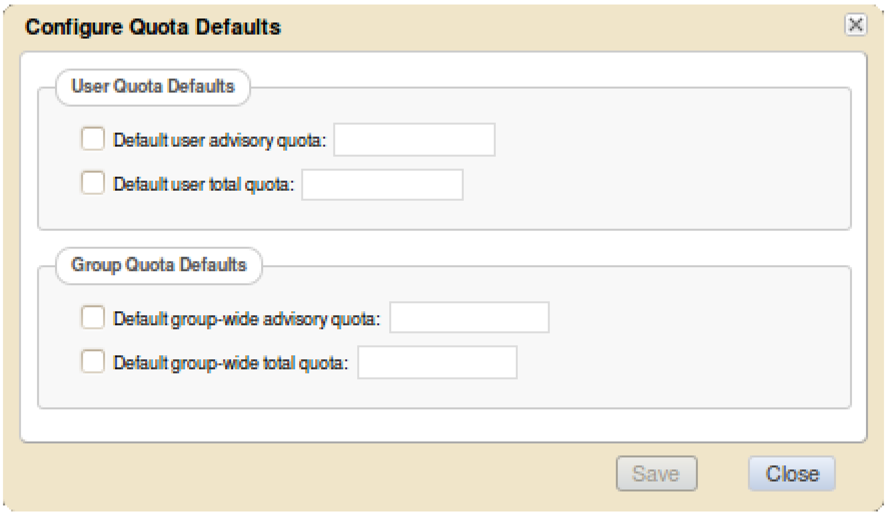


Buttons:

- **OK** - save changes and exit
- **Close** - exit without saving changes

Quota Defaults

The Configure Quota Defaults dialog lets you set the default quotas that apply to users and groups.



Configure Quota Defaults
User Quota Defaults
□ Default user advisory quota:
□ Default user total quota:
Group Quota Defaults
□ Default group-wide advisory quota:
□ Default group-wide total quota:
Save
Close

The User Quota Defaults section contains the following fields:

- **Default User Advisory Quota** - if selected, sets the advisory quota that applies to all users without an explicit advisory quota.
- **Default User Total Quota** - if selected, sets the advisory quota that applies to all users without an explicit total quota.

The Group Quota Defaults section contains the following fields:

- **Default Group Advisory Quota** - if selected, sets the advisory quota that applies to all groups without an explicit

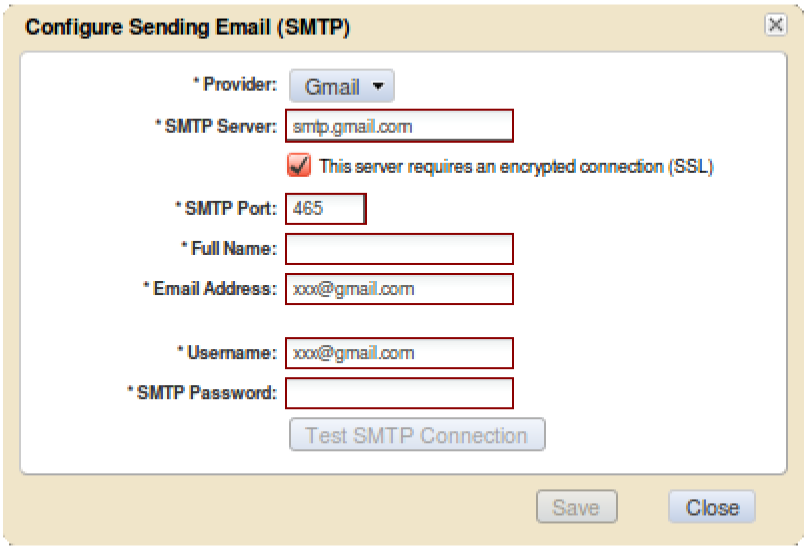
advisory quota.
- **Default Group Total Quota** - if selected, sets the advisory quota that applies to all groups without an explicit total quota.

Buttons:

- **Save** - saves the settings
- **Close** - exits without saving the settings

SMTP

The Configure Sending Email dialog lets you configure the email account from which the Greenplum HD EE cluster sends alerts and other notifications.



* Provider: Gmail

* SMTP Server: smtp.gmail.com

☑ This server requires an encrypted connection (SSL)

* SMTP Port: 465

* Full Name:

* Email Address: xxx@gmail.com

* Username: xxx@gmail.com

* SMTP Password:

The Configure Sending Email (SMTP) dialog contains the following fields:

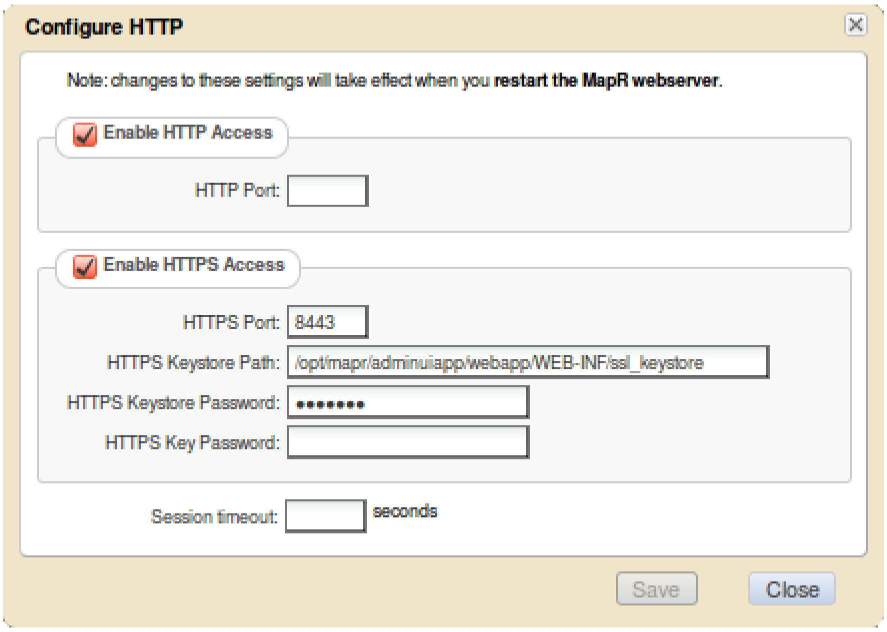
- **Provider** - selects Gmail or another email provider; if you select Gmail, the other fields are partially populated to help you with the configuration
- **SMTP Server** specifies the SMTP server to use when sending email.
- **The server requires an encrypted connection (SSL)** - use SSL when connecting to the SMTP server
- **SMTP Port** - the port to use on the SMTP server
- **Full Name** - the name used in the From field when the cluster sends an alert email
- **Email Address** - the email address used in the From field when the cluster sends an alert email.
- **Username** - the username used to log onto the email account the cluster will use to send email.
- **SMTP Password** - the password to use when sending email.

Buttons:

- **Save** - saves the settings
- **Close** - exits without saving the settings

HTTP

The Configure HTTP dialog lets you configure access to the Greenplum HD EE Control System via HTTP and HTTPS.



Configure HTTP

Note: changes to these settings will take effect when you **restart the MapR webserver**.

☑ Enable HTTP Access

HTTP Port:

☑ Enable HTTPS Access

HTTPS Port: 8443

HTTPS Keystore Path: /opt/mapr/adminuiapp/webapp/WEB-INF/ssl_keystore

HTTPS Keystore Password: •••••••

HTTPS Key Password:

Session timeout: seconds

The sections in the Configure HTTP dialog let you enable HTTP and HTTPS access, and set the session timeout, respectively:

- **Enable HTTP Access** - if selected, configure HTTP access with the following field:
    - **HTTP Port** - the port on which to connect to the Greenplum HD EE Control System via HTTP
- **Enable HTTPS Access** - if selected, configure HTTPS access with the following fields:
    - **HTTPS Port** - the port on which to connect to the Greenplum HD EE Control System via HTTPS
    - **HTTPS Keystore Path** - a path to the HTTPS keystore
    - **HTTPS Keystore Password** - a password to access the HTTPS keystore
    - **HTTPS Key Password** - a password to access the HTTPS key
- **Session Timeout** - the number of seconds before an idle session times out.

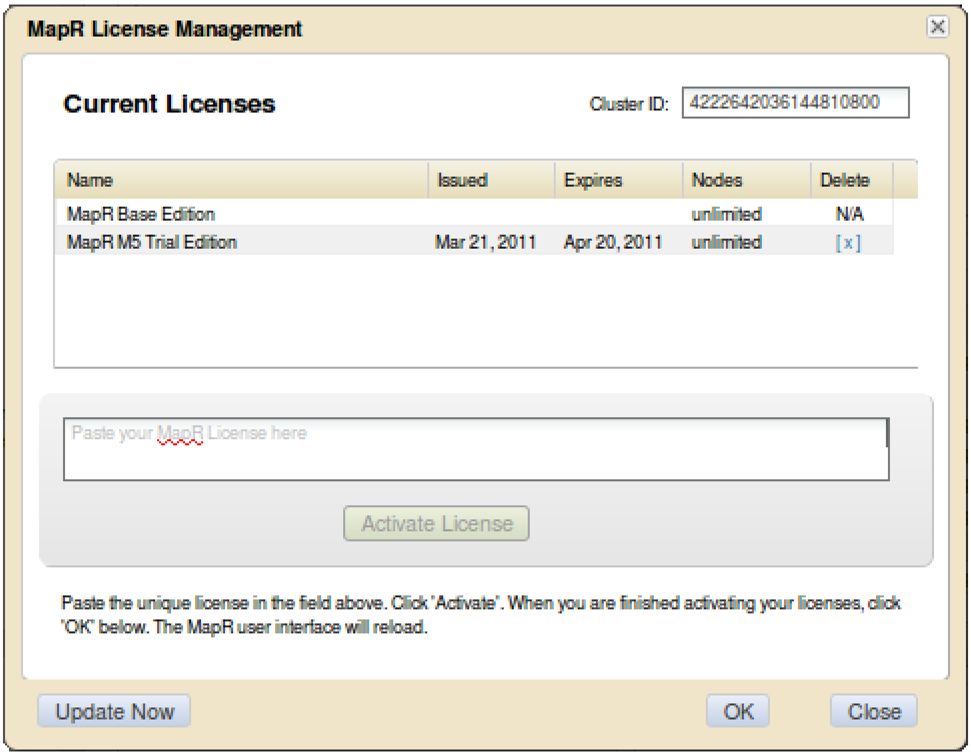
Buttons:

- **Save** - saves the settings
- **Close** - exits without saving the settings

Greenplum HD EE Licenses

The Greenplum HD EE License Management dialog lets you add and activate licenses for the cluster, and displays the Cluster ID and the following information about existing licenses:

- **Name** - the name of each license
- **Issued** - the date each license was issued
- **Expires** - the expiration date of each license
- **Nodes** - the nodes to which each license applies



MapR License Management

Current Licenses

Cluster ID: 4222642036144810800

Name	Issued	Expires	Nodes	Delete
MapR Base Edition			unlimited	N/A
MapR M5 Trial Edition	Mar 21, 2011	Apr 20, 2011	unlimited	[x]

Paste your MapR License here

Paste the unique license in the field above. Click 'Activate'. When you are finished activating your licenses, click 'OK' below. The MapR user interface will reload.

Fields:

- license field - a field for entering (or pasting) a new license

Buttons:

- **Delete** - deletes the corresponding license
- **Activate License** - activates the license pasted into the license field
- **Update Now** - helps you check for updates and additional features
- **OK** - saves changes and exits
- **Close** - exits without saving changes

Other Views

In addition to the Greenplum HD EE Control System views, there are views that display detailed information about the system:

- Hive - information about Hive on the cluster
- HBase - information about HBase on the cluster
- Oozie - information about Oozie on the cluster
- JobTracker - information about the JobTracker
- CLDB - information about the CLDB
- Nagios - information about Nagios
- Terminal - command-line interface to the cluster

Scripts and Commands

This section contains information about the following scripts and commands:

- configure.sh - configures a node or client to work with the cluster
- disksetup - sets up disks for use by Greenplum HD EE storage
- Hadoop MFS - enhanced `hadoop fs` command
- mapr-support-collect.sh - collects information for use by Greenplum HD EE Support
- rollingupgrade.sh - upgrades software on a Greenplum HD EE cluster
- zkdatacleaner.sh - cleans up old ZooKeeper data

configure.sh

Sets up a Greenplum HD EE cluster or client, creates `/opt/mapr/conf/mapr-clusters.conf`, and updates the corresponding `*.conf` and `*.xml` files.

The normal use of `configure.sh` is to set up a Greenplum HD EE cluster, or to set up a Greenplum HD EE client for communication with one or more clusters.

- To set up a cluster, run `configure.sh` on all nodes specifying the cluster's CLDB and ZooKeeper nodes, and a cluster name if desired. If setting up a cluster on virtual machines, use the `-isvm` parameter.
- To set up a client, run `configure.sh` on the client machine, specifying the CLDB and ZooKeeper nodes of the cluster or clusters. On a client, use both the `-c` and `-C` parameters.
- If you change the location or number of CLDB or ZooKeeper services in a cluster, run `configure.sh` and specify the new lineup of CLDB and ZooKeeper nodes.
- To rename a cluster, run `configure.sh` on all nodes with the `-N` option

Syntax

```
/opt/mapr/server/configure.sh
-C <host>[:<port>][,<host>[:<port>]...]
-Z <host>[:<port>][,<host>[:<port>]...]
[ -c ]
[ --isvm ]
[ -J <CLDB JMX port> ]
[ -L <log file> ]
[ -N <cluster name> ]
```

Parameters

Parameter	Description
-C	A list of the CLDB nodes in the cluster.
-Z	A list of the ZooKeeper nodes in the cluster. The `-Z` option is required unless `-c` (lowercase) is specified.
--isvm	Specifies virtual machine setup. Required when `configure.sh` is run on a virtual machine.
-c	Specifies client setup. See Setting Up the Client.
-J	Specifies the JMX port for the CLDB. Default: `7220`
-L	Specifies a log file. If not specified, `configure.sh` logs errors to `/opt/mapr/logs/configure.log` .
-N	Specifies the cluster name, to prevent ambiguity in multiple-cluster environments.

Examples

Add a node (not CLDB or ZooKeeper) to a cluster that is running the CLDB and ZooKeeper on three nodes:

On the new node, run the following command:

```
/opt/mapr/server/configure.sh -C 10.10.100.1,10.10.100.2,10.10.100.3 -Z
10.10.100.1,10.10.100.2,10.10.100.3
```

Configure a client to work with MyCluster, which has one CLDB at 10.10.100.1:

On the client, run the following command:

```
/opt/mapr/server/configure.sh -N MyCluster -c -C 10.10.100.1:7222
```

Rename the cluster to Cluster1 without changing the specified CLDB and ZooKeeper nodes:

On all nodes, run the following command:

```
/opt/mapr/server/configure.sh -N Cluster1 -R
```

disksetup

Formats specified disks for use by Greenplum HD EE storage.

For information about when and how to use `disksetup`, see Setting Up Disks for Greenplum HD EE.

To specify disks:

- Create a text file `/tmp/disks.txt` listing disks and partitions for use by Greenplum HD EE. Each line lists a single disk, or partitions on a single disk. Example:

```
/dev/sdb
/dev/sdc1 /dev/sdc2 /dev/sdc4
/dev/sdd
```

Later, when you run `disksetup` to format the disks, specify the disks and partitions file. Example:

```
disksetup -F /tmp/disks.txt
```

To test without formatting physical disks:

If you do not have physical partitions or disks available for reformatting, you can test Greenplum HD EE by creating a flat file and including a path to the file in the disk list file. You should create at least a 4GB file or larger.

The following example creates a 20 GB flat file (`bs=1G` specifies 1 gigabyte, multiply by `count=20`):

```
$ dd if=/dev/zero of=/root/storagefile bs=1G count=20
```

Using the above example, you would add the following to `/tmp/disks.txt`:

```
/root/storagefile
```

Syntax

```
/opt/mapr/server/disksetup
<disk list file>
[-F]
[-G]
[-W <stripe_width>]
```

Parameters

Parameter	Description
-F	Forces formatting of all specified disks. If not specified, `disksetup` does not re-format disks that have already been formatted for Greenplum HD EE.
-G	Generates `disktab` contents from input disk list, but does not format disks. This option is useful if disk names change after a reboot, or if the `disktab` file is damaged.

-W	Specifies the number of disks per storage pool.

Examples

Set up disks specified in the file /tmp/disks.txt:

```
/opt/mapr/server/disksetup -F /tmp/disks.txt
```

Hadoop MFS

The `hadoop mfs` command performs operations on directories in the cluster. The main purposes of `hadoop mfs` are to display directory information and contents, to create symbolic links, and to set compression and chunk size on a directory.

```
hadoop mfs
    [ -ln <target> <symlink> ]
    [ -ls <path> ]
    [ -lsd <path> ]
    [ -lsr <path> ]
    [ -lss <path> ]
    [ -setcompression on|off <dir> ]
    [ -setchunksize <size> ]
    [ -help <command> ]
```

Options

The normal command syntax is to specify a single option from the following table, along with its corresponding arguments. If compression and chunk size are not set explicitly for a given directory, the values are inherited from the parent directory.

Option	Description
-ln	Creates a symbolic link `<symlink>` that points to the target path `<target>`, similar to the standard Linux `ln -s` comman d.
-ls	Lists files in the directory specified by `<path>`. The `hadoop mfs -ls` command corresponds to the standard `hadoop fs -ls` command, but provides the following additional information: • Blocks used for each file • Server where each block resides
-lsd	Lists files in the directory specified by `<path>`, and also provides information about the specified directory itself: • Whether compression is enabled for the directory (indicated by `z` • The configured chunk size (in bytes) for the directory.
-lsr	Lists files in the directory and subdirectories specified by `<pa th>`, recursively. The `hadoop mfs -lsr` command corresponds to the standard `hadoop fs -lsr` command, but provides the following additional information: • Blocks used for each file • Server where each block resides |
-lss <path>	Lists files in the directory specified by `<path>`, with an additional column that displays the number of disk blocks per file. Disk blocks are 8192 bytes.
-setcompression	Turns compression on or off on the specified directory.

-setchunksize	Sets the chunk size in bytes for the specified directory. The <`size`> parameter must be a multiple of 65536.
-help	Displays help for the `hadoop mfs` command.

Output

When used with `-ls`, `-lsd`, `-lsr`, or `-lss`, `hadoop mfs` displays information about files and directories. For each file or directory `hadoop mfs` displays a line of basic information followed by lines listing the chunks that make up the file, in the following format:

> *{mode} {compression} {replication} {owner} {group} {size} {date} {chunk size} {name}*
> *{chunk} {fid} {host} [{host}...]*
> *{chunk} {fid} {host} [{host}...]*
> *...*

Volume links are displayed as follows:

> *{mode} {compression} {replication} {owner} {group} {size} {date} {chunk size} {name}*
> *{chunk} {target volume name} {writability} {fid} -> {fid} [{host}...]*

For volume links, the first `fid` is the chunk that stores the volume link itself; the `fid` after the arrow (`->`) is the first chunk in the target volume.

The following table describes the values:

mode	A text string indicating the read, write, and execute permissions for the owner, group, and other permissions. See also Managing Permissions.
compression	• U - directory is not compressed • Z - directory is compressed
replication	The replication factor of the file (directories display a dash instead)
owner	The owner of the file or directory
group	The group of the file of directory
size	The size of the file or directory
date	The date the file or directory was last modified
chunk size	The chunk size of the file or directory
name	The name of the file or directory
chunk	The chunk number. The first chunk is a primary chunk labeled "p", a 64K chunk containing the root of the file. Subsequent chunks are numbered in order.
fid	The file ID.
host	The host on which the chunk resides. When several hosts are listed, the first host is the first copy of the chunk and subsequent hosts are replicas.
target volume name	The name of the volume pointed to by a volume link.
writability	Displays whether the volume is writable.

mapr-support-collect.sh

Collects information about a cluster's recent activity, to help Greenplum HD EE Support diagnose problems.

Syntax

```
/opt/mapr/support/tools/mapr-support-collect.sh
    [ -h|--hosts <host file> ]
    [ -H|--host <host entry> ]
    [ -Q|--no-cldb ]
    [ -n|--name <name> ]
    [ -l|--no-logs ]
    [ -s|--no-statistics ]
    [ -c|--no-conf ]
    [ -i|--no-sysinfo ]
    [ -x|--exclude-cluster ]
    [ -u|--user <user> ]
    [ -p|--par <par> ]
    [ -t|--dump-timeout <dump timeout> ]
    [ -T|--scp-timeout <SCP timeout> ]
    [ -C|--cluster-timeout <cluster timeout> ]
    [ -y|--yes ]
    [ -S|--scp-port <SCP port> ]
    [ --collect-cores ]
    [ --move-cores ]
    [ --port <port> ]
    [ -?|--help ]
```

Parameters

Parameter	Description
-h or --hosts	A file containing a list of hosts. Each line contains one host entry, in the format `[user@]host[:port]`
-H or --host	One or more hosts in the format `[user@]host[:port]`
-Q or --no-cldb	If specified, the command does not query the CLDB for list of nodes
-n or --name	Specifies the name of the output file
-l or --no-logs	If specified, the command output does not include log files
-s or --no-statistics	If specified, the command output does not include statistics
-c or --no-conf	If specified, the command output does not include configurations
-i or --no-sysinfo	If specified, the command output does not include system information
-x or --exclude-cluster	If specified, the command output does not collect cluster diagnostics
-u or --user	The username for ssh connections
-p or --par	The maximum number of nodes from which support dumps will be gathered concurrently (default: 10)
-t or --dump-timeout	The timeout for execution of the `mapr-support-dump` command on a node (default: 120 seconds or 0 = no limit)
-T or --scp-timeout	The timeout for copy of support dump output from a remote node to the local file system (default: 120 seconds or 0 = no limit)
-C or --cluster-timeout	The timeout for collection of cluster diagnostics (default: 300 seconds or 0 = no limit)
-y or --yes	If specified, the command does not require acknowledgement of the number of nodes that will be affected
-S or --scp-port	The local port to which remote nodes will establish an SCP session
--collect-cores	If specified, the command collects cores of running mfs processes from all nodes (off by default)

--move-cores	If specified, the command moves mfs and nfs cores from /opt/cores from all nodes (off by default)
--port	The port number used by FileServer (default: 5660)
-? or --help	Displays usage help text

Examples

Collect support information and dump it to the file /tmp/support-output.txt:

```
/opt/mapr/support/tools/mapr-support-collect.sh -n /tmp/support-output.txt
```

rollingupgrade.sh

Upgrades a Greenplum HD EE cluster to a specified version of the Greenplum HD EE software, or to a specific set of Greenplum HD EE packages.

Syntax

```
/opt/upgrade-mapr/rollingupgrade.sh
    [-c <cluster name>]
    [-d]
    [-h]
    [-i <identity file>]
    [-p <directory>]
    [-s]
    [-u <username>]
    [-v <version>]
```

Parameters

Parameter	Description
-c	Cluster name.
-d	If specified, performs a dry run without upgrading the cluster.
-h	Displays help text.
-i	Specifies an identity file for SSH. See the SSH man page.
-p	Specifies a directory containing the upgrade packages.
-s	Specifies SSH to upgrade nodes
-u	A username for SSH
-v	upgrade_version

zkdatacleaner.sh

Removes old ZooKeeper data that might otherwise interfere with installation or proper operation of a node or client.

Syntax

```
/opt/mapr/server/zkdatacleaner.sh
```

Examples

Clean up old ZooKeeper data on the current node:

```
/opt/mapr/server/zkdatacleaner.sh
```

Configuration Files

hadoop-metrics.properties

The `hadoop-metrics.properties` files direct Greenplum HD EE where to output service metric reports: to an output file (`FileContext`) or to Ganglia 3.1 (`MapRGangliaContext31`). A third context, `NullContext`, disables metrics. To direct metrics to an output file, comment out the lines pertaining to Ganglia and the `NullContext`; for the chosen service; to direct metrics to Ganglia, comment out the lines pertaining to the metrics file and the `NullContext`. See Service Metrics.

There are two `hadoop-metrics.properties` files:

- `/opt/mapr/hadoop/hadoop-<version>/conf/hadoop-metrics.properties` specifies output for standard Hadoop services
- `/opt/mapr/conf/hadoop-metrics.properties` specifies output for Greenplum HD EE-specific services

The following table describes the parameters for each service in the `hadoop-metrics.properties` files.

Parameter	Example Values	Description
<service>.class	• org.apache.hadoop.metrics.spi.NullContextWithUpdateThread • apache.hadoop.metrics.file.FileContext • com.mapr.fs.cldb.counters.MapRGangliaContext31	The class that implements the interface responsible for sending the service metrics to the appropriate handler. When implementing a class that sends metrics to Ganglia, set this property to the class name.
<service>.period	• 10 • 60	The interval between 2 service metrics data exports to the appropriate interface. This is independent of how often are the metrics updated in the framework.
<service>.fileName	/tmp/cldbmetrics.log	The path to the file where service metrics are exported when the cldb.class property is set to FileContext.
<service.servers	localhost:8649	The location of the gmon or gmeta that is aggregating metrics for this instance of the service, when the cldb.class property is set to GangliaContext.
<service>.spoof	1	Specifies whether the metrics being sent out from the server should be spoofed as coming from another server. All our fileserver metrics are also on cldb, but to make it appear to end users as if these properties were emitted by fileserver host, we spoof the metrics to Ganglia using this property. Currently only used for the FileServer service.

Examples

The `hadoop-metrics.properties` files are organized into sections for each service that provides metrics. Each section is divided into subsections for the three contexts.

/opt/mapr/hadoop/hadoop-<version>/conf/hadoop-metrics.properties

```
# Configuration of the "dfs" context for null
dfs.class=org.apache.hadoop.metrics.spi.NullContext

# Configuration of the "dfs" context for file
#dfs.class=org.apache.hadoop.metrics.file.FileContext
#dfs.period=10
#dfs.fileName=/tmp/dfsmetrics.log

# Configuration of the "dfs" context for ganglia
# Pick one: Ganglia 3.0 (former) or Ganglia 3.1 (latter)
# dfs.class=org.apache.hadoop.metrics.ganglia.GangliaContext
# dfs.class=org.apache.hadoop.metrics.ganglia.GangliaContext31
# dfs.period=10
# dfs.servers=localhost:8649


# Configuration of the "mapred" context for null
mapred.class=org.apache.hadoop.metrics.spi.NullContext

# Configuration of the "mapred" context for file
#mapred.class=org.apache.hadoop.metrics.file.FileContext
#mapred.period=10
#mapred.fileName=/tmp/mrmetrics.log

# Configuration of the "mapred" context for ganglia
# Pick one: Ganglia 3.0 (former) or Ganglia 3.1 (latter)
# mapred.class=org.apache.hadoop.metrics.ganglia.GangliaContext
# mapred.class=org.apache.hadoop.metrics.ganglia.GangliaContext31
# mapred.period=10
# mapred.servers=localhost:8649


# Configuration of the "jvm" context for null
#jvm.class=org.apache.hadoop.metrics.spi.NullContext

# Configuration of the "jvm" context for file
#jvm.class=org.apache.hadoop.metrics.file.FileContext
#jvm.period=10
#jvm.fileName=/tmp/jvmmetrics.log

# Configuration of the "jvm" context for ganglia
# jvm.class=org.apache.hadoop.metrics.ganglia.GangliaContext
# jvm.period=10
# jvm.servers=localhost:8649


# Configuration of the "ugi" context for null
ugi.class=org.apache.hadoop.metrics.spi.NullContext


# Configuration of the "fairscheduler" context for null
#fairscheduler.class=org.apache.hadoop.metrics.spi.NullContext

# Configuration of the "fairscheduler" context for file
#fairscheduler.class=org.apache.hadoop.metrics.file.FileContext
#fairscheduler.period=10
#fairscheduler.fileName=/tmp/fairschedulermetrics.log

# Configuration of the "fairscheduler" context for ganglia
# fairscheduler.class=org.apache.hadoop.metrics.ganglia.GangliaContext
# fairscheduler.period=10
# fairscheduler.servers=localhost:8649
#
```

/opt/mapr/conf/hadoop-metrics.properties

```
#########################################################################################
##################################
hadoop-metrics.properties
#########################################################################################
##################################

#CLDB metrics config - Pick one out of null,file or ganglia.
#Uncomment all properties in null, file or ganglia context, to send cldb metrics to that
context

# Configuration of the "cldb" context for null
#cldb.class=org.apache.hadoop.metrics.spi.NullContextWithUpdateThread
#cldb.period=10

# Configuration of the "cldb" context for file
#cldb.class=org.apache.hadoop.metrics.file.FileContext
#cldb.period=60
#cldb.fileName=/tmp/cldbmetrics.log

# Configuration of the "cldb" context for ganglia
cldb.class=com.mapr.fs.cldb.counters.MapRGangliaContext31
cldb.period=10
cldb.servers=localhost:8649
cldb.spoof=1

#FileServer metrics config - Pick one out of null,file or ganglia.
#Uncomment all properties in null, file or ganglia context, to send fileserver metrics to
that context

# Configuration of the "fileserver" context for null
#fileserver.class=org.apache.hadoop.metrics.spi.NullContextWithUpdateThread
#fileserver.period=10

# Configuration of the "fileserver" context for file
#fileserver.class=org.apache.hadoop.metrics.file.FileContext
#fileserver.period=60
#fileserver.fileName=/tmp/fsmetrics.log

# Configuration of the "fileserver" context for ganglia
fileserver.class=com.mapr.fs.cldb.counters.MapRGangliaContext31
fileserver.period=37
fileserver.servers=localhost:8649
fileserver.spoof=1

#########################################################################################
#######################
```

mapr-clusters.conf

The configuration file `/opt/mapr/conf/mapr-clusters.conf` specifies the CLDB nodes for one or more clusters that can be reached from the node or client on which it is installed.

Format:

```
clustername1 <CLDB> <CLDB> <CLDB>
clustername2 <CLDB> <CLDB> <CLDB>
```

The <CLDB> string format is one of the following:

- `host,ip:port` - To support hostnames even with DNS down.
- `host,:port` - Skip the IP
- `ip:port` - Skip the host
- `host` - skip the ip and port (default)
- `ip` - skip host and port, and avoid DNS

mapred-default.xml

The configuration file `mapred-default.xml` provides defaults that can be overridden using `mapred-site.xml`, and is located in the Hadoop core JAR file (`/opt/mapr/hadoop/hadoop-<version>/lib/hadoop-<version>-dev-core.jar`).

> ⚠ Do not modify `mapred-default.xml` directly. Instead, copy parameters into `mapred-site.xml` and modify them there. If mapred-site.xml does not already exist, create it.

The format for a parameter in both `mapred-default.xml` and `mapred-site.xml` is:

```
<property>
<name>io.sort.spill.percent</name>
<value>0.99</value>
<description>The soft limit in either the buffer or record collection
buffers. Once reached, a thread will begin to spill the contents to disk
in the background. Note that this does not imply any chunking of data to
the spill. A value less than 0.5 is not recommended.</description>
</property>
```

The `<name>` element contains the parameter name, the `<value>` element contains the parameter value, and the optional `<description>` element contains the parameter description. You can create XML for any parameter from the table below, using the example above as a guide.

Parameter	Value	Description
hadoop.job.history.location		If job tracker is static the history files are stored in this single well known place on local filesystem. If No value is set here, by default, it is in the local file system at $<hadoop.log.dir>/history. History files are moved to mapred.jobtracker.history.completed.location which is on MapRFs JobTracker volume. |
hadoop.job.history.user.location		User can specify a location to store the history files of a particular job. If nothing is specified, the logs are stored in output directory. The files are stored in "_logs/history/" in the directory. User can stop logging by giving the value "none".
hadoop.rpc.socket.factory.class.JobSubmissionProtocol		SocketFactory to use to connect to a Map/Reduce master (JobTracker). If null or empty, then use hadoop.rpc.socket.class.default.
io.map.index.skip	0	Number of index entries to skip between each entry. Zero by default. Setting this to values larger than zero can facilitate opening large map files using less memory.
io.sort.factor	256	The number of streams to merge at once while sorting files. This determines the number of open file handles.
io.sort.mb	100	Buffer used to hold map outputs in memory before writing final map outputs. Setting this value very low may cause spills. If a average input to map is "MapIn" bytes then typically value of io.sort.mb should be '1.25 times MapIn' bytes.

io.sort.record.percent	0.17	The percentage of io.sort.mb dedicated to tracking record boundaries. Let this value be r, io.sort.mb be x. The maximum number of records collected before the collection thread must block is equal to (r * x) / 4
io.sort.spill.percent	0.99	The soft limit in either the buffer or record collection buffers. Once reached, a thread will begin to spill the contents to disk in the background. Note that this does not imply any chunking of data to the spill. A value less than 0.5 is not recommended.
job.end.notification.url	\\	Indicates url which will be called on completion of job to inform end status of job. User can give at most 2 variables with URI : $jobId and $jobStatus. If they are present in URI, then they will be replaced by their respective values.
job.end.retry.attempts	0	Indicates how many times hadoop should attempt to contact the notification URL
job.end.retry.interval	30000	Indicates time in milliseconds between notification URL retry calls
jobclient.completion.poll.interval	5000	The interval (in milliseconds) between which the JobClient polls the JobTracker for updates about job status. You may want to set this to a lower value to make tests run faster on a single node system. Adjusting this value in production may lead to unwanted client-server traffic.
jobclient.output.filter	FAILED	The filter for controlling the output of the task's userlogs sent to the console of the JobClient. The permissible options are: NONE, KILLED, FAILED, SUCCEEDED and ALL.
jobclient.progress.monitor.poll.interval	1000	The interval (in milliseconds) between which the JobClient reports status to the console and checks for job completion. You may want to set this to a lower value to make tests run faster on a single node system. Adjusting this value in production may lead to unwanted client-server traffic.
map.sort.class	org.apache.hadoop.util.QuickSort	The default sort class for sorting keys.
mapr.localoutput.dir	output	The path for local output
mapr.localspill.dir	spill	The path for local spill
mapr.localvolumes.path	/var/mapr/local	The path for local volumes
mapred.acls.enabled	false	Specifies whether ACLs should be checked for authorization of users for doing various queue and job level operations. ACLs are disabled by default. If enabled, access control checks are made by JobTracker and TaskTracker when requests are made by users for queue operations like submit job to a queue and kill a job in the queue and job operations like viewing the job-details (See mapreduce.job.acl-view-job) or for modifying the job (See mapreduce.job.acl-modify-job) using Map/Reduce APIs, RPCs or via the console and web user interfaces.

mapred.child.env		User added environment variables for the task tracker child processes. Example : 1) A=foo This will set the env variable A to foo 2) B=$B:c This is inherit tasktracker's B env variable.
mapred.child.java.opts		Java opts for the task tracker child processes. The following symbol, if present, will be interpolated: (taskid) is replaced by current TaskID. Any other occurrences of '@' will go unchanged. For example, to enable verbose gc logging to a file named for the taskid in /tmp and to set the heap maximum to be a gigabyte, pass a 'value' of: -Xmx1024m -verbose:gc -Xloggc:/tmp/ (taskid).gc The configuration variable mapred.child.ulimit can be used to control the maximum virtual memory of the child processes.
mapred.child.oom_adj	10	Increase the OOM adjust for oom killer (linux specific). We only allow increasing the adj value. (valid values: 0-15)
mapred.child.renice	10	Nice value to run the job in. on linux the range is from -20 (most favorable) to 19 (least favorable). We only allow reducing the priority. (valid values: 0-19)
mapred.child.taskset	true	Run the job in a taskset. man taskset (linux specific) 1-4 CPUs: No taskset 5-8 CPUs: taskset 1- (processor 0 reserved for infrastructure processes) 9-n CPUs: taskset 2- (processors 0,1 reserved for infrastructure processes)
mapred.child.tmp	./tmp	To set the value of tmp directory for map and reduce tasks. If the value is an absolute path, it is directly assigned. Otherwise, it is prepended with task's working directory. The java tasks are executed with option -Djava.io.tmpdir='the absolute path of the tmp dir'. Pipes and streaming are set with environment variable, TMPDIR='the absolute path of the tmp dir'
mapred.child.ulimit		The maximum virtual memory, in KB, of a process launched by the Map-Reduce framework. This can be used to control both the Mapper/Reducer tasks and applications using Hadoop Pipes, Hadoop Streaming etc. By default it is left unspecified to let cluster admins control it via limits.conf and other such relevant mechanisms. Note: mapred.child.ulimit must be greater than or equal to the -Xmx passed to JavaVM, else the VM might not start.
mapred.cluster.map.memory.mb	-1	The size, in terms of virtual memory, of a single map slot in the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single map task via mapred.job.map.memory.mb, upto the limit specified by mapred.cluster.max.map.memory.mb, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.

mapred.cluster.max.map.memory.mb	-1	The maximum size, in terms of virtual memory, of a single map task launched by the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single map task via mapred.job.map.memory.mb, upto the limit specified by mapred.cluster.max.map.memory.mb, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.
mapred.cluster.max.reduce.memory.mb	-1	The maximum size, in terms of virtual memory, of a single reduce task launched by the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single reduce task via mapred.job.reduce.memory.mb, upto the limit specified by mapred.cluster.max.reduce.memory.mb, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.
mapred.cluster.reduce.memory.mb	-1	The size, in terms of virtual memory, of a single reduce slot in the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single reduce task via mapred.job.reduce.memory.mb, upto the limit specified by mapred.cluster.max.reduce.memory.mb, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.
mapred.compress.map.output	false	Should the outputs of the maps be compressed before being sent across the network. Uses SequenceFile compression.
mapred.healthChecker.interval	60000	Frequency of the node health script to be run, in milliseconds
mapred.healthChecker.script.args		List of arguments which are to be passed to node health script when it is being launched comma seperated.
mapred.healthChecker.script.path		Absolute path to the script which is periodically run by the node health monitoring service to determine if the node is healthy or not. If the value of this key is empty or the file does not exist in the location configured here, the node health monitoring service is not started.
mapred.healthChecker.script.timeout	600000	Time after node health script should be killed if unresponsive and considered that the script has failed.
mapred.hosts.exclude		Names a file that contains the list of hosts that should be excluded by the jobtracker. If the value is empty, no hosts are excluded.
mapred.hosts		Names a file that contains the list of nodes that may connect to the jobtracker. If the value is empty, all hosts are permitted.

mapred.inmem.merge.threshold	1000	The threshold, in terms of the number of files for the in-memory merge process. When we accumulate threshold number of files we initiate the in-memory merge and spill to disk. A value of 0 or less than 0 indicates we want to DON'T have any threshold and instead depend only on the ramfs's memory consumption to trigger the merge.
mapred.job.map.memory.mb	-1	The size, in terms of virtual memory, of a single map task for the job. A job can ask for multiple slots for a single map task, rounded up to the next multiple of mapred.cluster.map.memory.mb and upto the limit specified by mapred.cluster.max.map.memory.mb, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off iff mapred.cluster.map.memory.mb is also turned off (-1).
mapred.job.map.memory.physical.mb		Maximum physical memory limit for map task of this job. If limit is exceeded task attempt will be FAILED.
mapred.job.queue.name	default	Queue to which a job is submitted. This must match one of the queues defined in mapred.queue.names for the system. Also, the ACL setup for the queue must allow the current user to submit a job to the queue. Before specifying a queue, ensure that the system is configured with the queue, and access is allowed for submitting jobs to the queue.
mapred.job.reduce.input.buffer.percent	0.0	The percentage of memory- relative to the maximum heap size- to retain map outputs during the reduce. When the shuffle is concluded, any remaining map outputs in memory must consume less than this threshold before the reduce can begin.
mapred.job.reduce.memory.mb	-1	The size, in terms of virtual memory, of a single reduce task for the job. A job can ask for multiple slots for a single map task, rounded up to the next multiple of mapred.cluster.reduce.memory.mb and upto the limit specified by mapred.cluster.max.reduce.memory.mb, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off iff mapred.cluster.reduce.memory.mb is also turned off (-1).
mapred.job.reduce.memory.physical.mb		Maximum physical memory limit for reduce task of this job. If limit is exceeded task attempt will be FAILED..
mapred.job.reuse.jvm.num.tasks	-1	How many tasks to run per jvm. If set to -1, there is no limit.
mapred.job.shuffle.input.buffer.percent	0.70	The percentage of memory to be allocated from the maximum heap size to storing map outputs during the shuffle.
mapred.job.shuffle.merge.percent	0.66	The usage threshold at which an in-memory merge will be initiated, expressed as a percentage of the total memory allocated to storing in-memory map outputs, as defined by mapred.job.shuffle.input.buffer.percent.

mapred.job.tracker.handler.count	10	The number of server threads for the JobTracker. This should be roughly 4% of the number of tasktracker nodes.
mapred.job.tracker.history.completed.location	/var/mapr/cluster/mapred/jobTracker/history/done	The completed job history files are stored at this single well-known location. If nothing is specified, the files are stored at $<hadoop.job.history.location>/done in local filesystem.
mapred.job.tracker.http.address	0.0.0.0:50030	The job tracker http server address and port the server will listen on. If the port is 0 then the server will start on a free port.
mapred.job.tracker.persist.jobstatus.active	false	Indicates if persistency of job status information is active or not.
mapred.job.tracker.persist.jobstatus.dir	/var/mapr/cluster/mapred/jobTracker/jobsInfo	The directory where the job status information is persisted in a file system to be available after it drops of the memory queue and between jobtracker restarts.
mapred.job.tracker.persist.jobstatus.hours	0	The number of hours job status information is persisted in DFS. The job status information will be available after it drops of the memory queue and between jobtracker restarts. With a zero value the job status information is not persisted at all in DFS.
mapred.job.tracker	localhost:9001	jobTracker address ip:port or use uri maprfs:/// for default cluster or maprfs:///mapr/san_jose_cluster1 to connect 'san_jose_cluster1' cluster.
mapred.jobtracker.completeuserjobs.maximum	100	The maximum number of complete jobs per user to keep around before delegating them to the job history.
mapred.jobtracker.instrumentation	org.apache.hadoop.mapred.JobTrackerMetricsInst	Expert: The instrumentation class to associate with each JobTracker.
mapred.jobtracker.job.history.block.size	3145728	The block size of the job history file. Since the job recovery uses job history, its important to dump job history to disk as soon as possible. Note that this is an expert level parameter. The default value is set to 3 MB.
mapred.jobtracker.jobhistory.lru.cache.size	5	The number of job history files loaded in memory. The jobs are loaded when they are first accessed. The cache is cleared based on LRU.
mapred.jobtracker.maxtasks.per.job	-1	The maximum number of tasks for a single job. A value of -1 indicates that there is no maximum.
mapred.jobtracker.plugins		Comma-separated list of jobtracker plug-ins to be activated.
mapred.jobtracker.port	9001	Port on which JobTracker listens.
mapred.jobtracker.restart.recover	true	"true" to enable (job) recovery upon restart, "false" to start afresh
mapred.jobtracker.retiredjobs.cache.size	1000	The number of retired job status to keep in the cache.
mapred.jobtracker.taskScheduler.maxRunningTasksPerJob		The maximum number of running tasks for a job before it gets preempted. No limits if undefined.
mapred.jobtracker.taskScheduler	org.apache.hadoop.mapred.JobQueueTaskScheduler	The class responsible for scheduling the tasks.

mapred.line.input.format.linespermap	1	Number of lines per split in NLineInputFormat.
mapred.local.dir.minspacekill	0	If the space in mapred.local.dir drops under this, do not ask more tasks until all the current ones have finished and cleaned up. Also, to save the rest of the tasks we have running, kill one of them, to clean up some space. Start with the reduce tasks, then go with the ones that have finished the least. Value in bytes.
mapred.local.dir.minspacestart	0	If the space in mapred.local.dir drops under this, do not ask for more tasks. Value in bytes.
mapred.local.dir	$<hadoop.tmp.dir>/mapred/local	The local directory where MapReduce stores intermediate data files. May be a comma-separated list of directories on different devices in order to spread disk i/o. Directories that do not exist are ignored.
mapred.map.child.env		User added environment variables for the task tracker child processes. Example : 1) A=foo This will set the env variable A to foo 2) B=$B:c This is inherit tasktracker's B env variable.
mapred.map.child.java.opts	-XX:ErrorFile=/opt/cores/hadoop/java_error%p.log	Java opts for the map tasks. The following symbol, if present, will be interpolated: (taskid) is replaced by current TaskID. Any other occurrences of '@' will go unchanged. For example, to enable verbose gc logging to a file named for the taskid in /tmp and to set the heap maximum to be a gigabyte, pass a 'value' of: -Xmx1024m -verbose:gc -Xloggc:/tmp/ (taskid).gc The configuration variable mapred.<map/reduce>.child.ulimit can be used to control the maximum virtual memory of the child processes. MapR: Default heapsize(-Xmx) is determined by memory reserved for mapreduce at tasktracker. Reduce task is given more memory than a map task. Default memory for a map task = (Total Memory reserved for mapreduce) * (#mapslots/ (#mapslots + 1.3*#reduceslots))
mapred.map.child.ulimit		The maximum virtual memory, in KB, of a process launched by the Map-Reduce framework. This can be used to control both the Mapper/Reducer tasks and applications using Hadoop Pipes, Hadoop Streaming etc. By default it is left unspecified to let cluster admins control it via limits.conf and other such relevant mechanisms. Note: mapred.<map/reduce>.child.ulimit must be greater than or equal to the -Xmx passed to JavaVM, else the VM might not start.
mapred.map.max.attempts	4	Expert: The maximum number of attempts per map task. In other words, framework will try to execute a map task these many number of times before giving up on it.
mapred.map.output.compression.codec	org.apache.hadoop.io.compress.DefaultCodec	If the map outputs are compressed, how should they be compressed?
mapred.map.tasks.speculative.execution	true	If true, then multiple instances of some map tasks may be executed in parallel.

mapred.map.tasks	2	The default number of map tasks per job. Ignored when mapred.job.tracker is "local".
mapred.max.tracker.blacklists	4	The number of blacklists for a taskTracker by various jobs after which the task tracker could be blacklisted across all jobs. The tracker will be given a tasks later (after a day). The tracker will become a healthy tracker after a restart.
mapred.max.tracker.failures	4	The number of task-failures on a tasktracker of a given job after which new tasks of that job aren't assigned to it.
mapred.merge.recordsBeforeProgress	10000	The number of records to process during merge before sending a progress notification to the TaskTracker.
mapred.min.split.size	0	The minimum size chunk that map input should be split into. Note that some file formats may have minimum split sizes that take priority over this setting.
mapred.output.compress	false	Should the job outputs be compressed?
mapred.output.compression.codec	org.apache.hadoop.io.compress.DefaultCodec	If the job outputs are compressed, how should they be compressed?
mapred.output.compression.type	RECORD	If the job outputs are to compressed as SequenceFiles, how should they be compressed? Should be one of NONE, RECORD or BLOCK.
mapred.queue.default.state	RUNNING	This values defines the state , default queue is in. the values can be either "STOPPED" or "RUNNING" This value can be changed at runtime.
mapred.queue.names	default	Comma separated list of queues configured for this jobtracker. Jobs are added to queues and schedulers can configure different scheduling properties for the various queues. To configure a property for a queue, the name of the queue must match the name specified in this value. Queue properties that are common to all schedulers are configured here with the naming convention, mapred.queue.$QUEUE-NAME.$PROPERTY-NAME, for e.g. mapred.queue.default.submit-job-acl. The number of queues configured in this parameter could depend on the type of scheduler being used, as specified in mapred.jobtracker.taskScheduler. For example, the JobQueueTaskScheduler supports only a single queue, which is the default configured here. Before adding more queues, ensure that the scheduler you've configured supports multiple queues.
mapred.reduce.child.env		
mapred.reduce.child.java.opts	-XX:ErrorFile=/opt/cores/hadoop/java_error%p.log	Java opts for the reduce tasks. MapR: Default heapsize(-Xmx) is determined by memory reserved for mapreduce at tasktracker. Reduce task is given more memory than map task. Default memory for a reduce task = (Total Memory reserved for mapreduce) * (1.3*#reduceslots / (#mapslots + 1.3*#reduceslots))

mapred.reduce.child.ulimit		
mapred.reduce.copy.backoff	300	The maximum amount of time (in seconds) a reducer spends on fetching one map output before declaring it as failed.
mapred.reduce.max.attempts	4	Expert: The maximum number of attempts per reduce task. In other words, framework will try to execute a reduce task these many number of times before giving up on it.
mapred.reduce.parallel.copies	12	The default number of parallel transfers run by reduce during the copy(shuffle) phase.
mapred.reduce.slowstart.completed.maps	0.95	Fraction of the number of maps in the job which should be complete before reduces are scheduled for the job.
mapred.reduce.tasks.speculative.execution	true	If true, then multiple instances of some reduce tasks may be executed in parallel.
mapred.reduce.tasks	1	The default number of reduce tasks per job. Typically set to 99% of the cluster's reduce capacity, so that if a node fails the reduces can still be executed in a single wave. Ignored when mapred.job.tracker is "local".
mapred.skip.attempts.to.start.skipping	2	The number of Task attempts AFTER which skip mode will be kicked off. When skip mode is kicked off, the tasks reports the range of records which it will process next, to the TaskTracker. So that on failures, tasktracker knows which ones are possibly the bad records. On further executions, those are skipped.
mapred.skip.map.auto.incr.proc.count	true	The flag which if set to true, SkipBadRecords.COUNTER_MAP_PROCESSED_RECORDS is incremented by MapRunner after invoking the map function. This value must be set to false for applications which process the records asynchronously or buffer the input records. For example streaming. In such cases applications should increment this counter on their own.
mapred.skip.map.max.skip.records	0	The number of acceptable skip records surrounding the bad record PER bad record in mapper. The number includes the bad record as well. To turn the feature of detection/skipping of bad records off, set the value to 0. The framework tries to narrow down the skipped range by retrying until this threshold is met OR all attempts get exhausted for this task. Set the value to Long.MAX_VALUE to indicate that framework need not try to narrow down. Whatever records(depends on application) get skipped are acceptable.
mapred.skip.out.dir		If no value is specified here, the skipped records are written to the output directory at _logs/skip. User can stop writing skipped records by giving the value "none".

mapred.skip.reduce.auto.incr.proc.count	true	The flag which if set to true, SkipBadRecords.COUNTER_REDUCE_PROCESSED_GROUPS is incremented by framework after invoking the reduce function. This value must be set to false for applications which process the records asynchronously or buffer the input records. For example streaming. In such cases applications should increment this counter on their own.
mapred.skip.reduce.max.skip.groups	0	The number of acceptable skip groups surrounding the bad group PER bad group in reducer. The number includes the bad group as well. To turn the feature of detection/skipping of bad groups off, set the value to 0. The framework tries to narrow down the skipped range by retrying until this threshold is met OR all attempts get exhausted for this task. Set the value to Long.MAX_VALUE to indicate that framework need not try to narrow down. Whatever groups(depends on application) get skipped are acceptable.
mapred.submit.replication	10	The replication level for submitted job files. This should be around the square root of the number of nodes.
mapred.system.dir	/var/mapr/cluster/mapred/jobTracker/system	The shared directory where MapReduce stores control files.
mapred.task.cache.levels	2	This is the max level of the task cache. For example, if the level is 2, the tasks cached are at the host level and at the rack level.
mapred.task.profile.maps	0-2	To set the ranges of map tasks to profile. mapred.task.profile has to be set to true for the value to be accounted.
mapred.task.profile.reduces	0-2	To set the ranges of reduce tasks to profile. mapred.task.profile has to be set to true for the value to be accounted.
mapred.task.profile	false	To set whether the system should collect profiler information for some of the tasks in this job? The information is stored in the user log directory. The value is "true" if task profiling is enabled.
mapred.task.timeout	600000	The number of milliseconds before a task will be terminated if it neither reads an input, writes an output, nor updates its status string.
mapred.task.tracker.http.address	0.0.0.0:50060	The task tracker http server address and port. If the port is 0 then the server will start on a free port.
mapred.task.tracker.report.address	127.0.0.1:0	The interface and port that task tracker server listens on. Since it is only connected to by the tasks, it uses the local interface. EXPERT ONLY. Should only be changed if your host does not have the loopback interface.
mapred.task.tracker.task-controller	org.apache.hadoop.mapred.DefaultTaskController	TaskController which is used to launch and manage task execution
mapred.tasktracker.dns.interface	default	The name of the Network Interface from which a task tracker should report its IP address.

mapred.tasktracker.dns.nameserver	default	The host name or IP address of the name server (DNS) which a TaskTracker should use to determine the host name used by the JobTracker for communication and display purposes.
mapred.tasktracker.expiry.interval	600000	Expert: The time-interval, in miliseconds, after which a tasktracker is declared 'lost' if it doesn't send heartbeats.
mapred.tasktracker.indexcache.mb	10	The maximum memory that a task tracker allows for the index cache that is used when serving map outputs to reducers.
mapred.tasktracker.instrumentation	org.apache.hadoop.mapred.TaskTrackerMetricsInst	Expert: The instrumentation class to associate with each TaskTracker.
mapred.tasktracker.map.tasks.maximum	(CPUS > 2) ? (CPUS * 0.75) : 1	The maximum number of map tasks that will be run simultaneously by a task tracker.
mapred.tasktracker.memory_calculator_plugin		Name of the class whose instance will be used to query memory information on the tasktracker. The class must be an instance of org.apache.hadoop.util.MemoryCalculatorPlugin. If the value is null, the tasktracker attempts to use a class appropriate to the platform. Currently, the only platform supported is Linux.
mapred.tasktracker.reduce.tasks.maximum	(CPUS > 2) ? (CPUS * 0.50): 1	The maximum number of reduce tasks that will be run simultaneously by a task tracker.
mapred.tasktracker.taskmemorymanager.monitoring-interval	5000	The interval, in milliseconds, for which the tasktracker waits between two cycles of monitoring its tasks' memory usage. Used only if tasks' memory management is enabled via mapred.tasktracker.tasks.maxmemory.
mapred.tasktracker.tasks.sleeptime-before-sigkill	5000	The time, in milliseconds, the tasktracker waits for sending a SIGKILL to a process, after it has been sent a SIGTERM.
mapred.temp.dir	$<hadoop.tmp.dir>/mapred/temp	A shared directory for temporary files.
mapred.user.jobconf.limit	5242880	The maximum allowed size of the user jobconf. The default is set to 5 MB
mapred.userlog.limit.kb	0	The maximum size of user-logs of each task in KB. 0 disables the cap.
mapred.userlog.retain.hours	24	The maximum time, in hours, for which the user-logs are to be retained after the job completion.
mapreduce.heartbeat.10	300	heartbeat in milliseconds for small cluster (less than or equal 10 nodes)
mapreduce.heartbeat.100	1000	heartbeat in milliseconds for medium cluster (11 - 100 nodes). Scales linearly between 300ms - 1s
mapreduce.heartbeat.1000	10000	heartbeat in milliseconds for medium cluster (101 - 1000 nodes). Scales linearly between 1s - 10s
mapreduce.heartbeat.10000	100000	heartbeat in milliseconds for medium cluster (1001 - 10000 nodes). Scales linearly between 10s - 100s

mapreduce.job.acl-modify-job		job specific access-control list for 'modifying' the job. It is only used if authorization is enabled in Map/Reduce by setting the configuration property mapred.acls.enabled to true. This specifies the list of users and/or groups who can do modification operations on the job. For specifying a list of users and groups the format to use is "user1,user2 group1,group". If set to '*', it allows all users/groups to modify this job. If set to ' '(i.e. space), it allows none. This configuration is used to guard all the modifications with respect to this job and takes care of all the following operations: o killing this job o killing a task of this job, failing a task of this job o setting the priority of this job Each of these operations are also protected by the per-queue level ACL "acl-administer-jobs" configured via mapred-queues.xml. So a caller should have the authorization to satisfy either the queue-level ACL or the job-level ACL. Irrespective of this ACL configuration, job-owner, the user who started the cluster, cluster administrators configured via mapreduce.cluster.administrators and queue administrators of the queue to which this job is submitted to configured via mapred.queue.queue-name.acl-administer-jobs in mapred-queue-acls.xml can do all the modification operations on a job. By default, nobody else besides job-owner, the user who started the cluster, cluster administrators and queue administrators can perform modification operations on a job.

mapreduce.job.acl-view-job		job specific access-control list for 'viewing' the job. It is only used if authorization is enabled in Map/Reduce by setting the configuration property mapred.acls.enabled to true. This specifies the list of users and/or groups who can view private details about the job. For specifying a list of users and groups the format to use is "user1,user2 group1,group". If set to '*', it allows all users/groups to modify this job. If set to ' '(i.e. space), it allows none. This configuration is used to guard some of the job-views and at present only protects APIs that can return possibly sensitive information of the job-owner like o job-level counters o task-level counters o tasks' diagnostic information o task-logs displayed on the TaskTracker web-UI and o job.xml showed by the JobTracker's web-UI Every other piece of information of jobs is still accessible by any other user, for e.g., JobStatus, JobProfile, list of jobs in the queue, etc. Irrespective of this ACL configuration, job-owner, the user who started the cluster, cluster administrators configured via mapreduce.cluster.administrators and queue administrators of the queue to which this job is submitted to configured via mapred.queue.queue-name.acl-administer-jobs in mapred-queue-acls.xml can do all the view operations on a job. By default, nobody else besides job-owner, the user who started the cluster, cluster administrators and queue administrators can perform view operations on a job.
mapreduce.job.complete.cancel.delegation.tokens	true	if false - do not unregister/cancel delegation tokens from renewal, because same tokens may be used by spawned jobs
mapreduce.job.split.metainfo.maxsize	10000000	The maximum permissible size of the split metainfo file. The JobTracker won't attempt to read split metainfo files bigger than the configured value. No limits if set to -1.
mapreduce.jobtracker.recovery.dir	/var/mapr/cluster/mapred/jobTracker/recovery	Recovery Directory
mapreduce.jobtracker.recovery.job.initialization.maxtime		Maximum time in seconds JobTracker will wait for initializing jobs before starting recovery. By default it is same as mapreduce.jobtracker.recovery.maxtime.
mapreduce.jobtracker.recovery.maxtime	480	Maximum time in seconds JobTracker should stay in recovery mode. JobTracker recovers job after talking to all running tasktrackers. On large cluster if many jobs are to be recovered, mapreduce.jobtracker.recovery.maxtime should be increased.
mapreduce.jobtracker.staging.root.dir	/var/mapr/cluster/mapred/jobTracker/staging	The root of the staging area for users' job files In practice, this should be the directory where users' home directories are located (usually /user)
mapreduce.maprfs.use.checksum	true	Deprecated; checksums are always used.

mapreduce.maprfs.use.compression	true	If true, then mapreduce will use checksums.
mapreduce.reduce.input.limit	-1	The limit on the input size of the reduce. If the estimated input size of the reduce is greater than this value, job is failed. A value of -1 means that there is no limit set.
mapreduce.task.classpath.user.precedence	false	Set to true if user wants to set different classpath.
mapreduce.tasktracker.group		Expert: Group to which TaskTracker belongs. If LinuxTaskController is configured via mapreduce.tasktracker.taskcontroller, the group owner of the task-controller binary should be same as this group.
mapreduce.tasktracker.heapbased.memory.management	false	Expert only: If admin wants to prevent swapping by not launching too many tasks use this option. Task's memory usage is based on max java heap size (-Xmx). By default -Xmx will be computed by tasktracker based on slots and memory reserved for mapreduce tasks. See mapred.map.child.java.opts/mapred.reduce.child.java.opts.
mapreduce.tasktracker.jvm.idle.time	10000	If jvm is idle for more than mapreduce.tasktracker.jvm.idle.time (milliseconds) tasktracker will kill it.
mapreduce.tasktracker.outofband.heartbeat	false	Expert: Set this to true to let the tasktracker send an out-of-band heartbeat on task-completion for better latency.
mapreduce.tasktracker.prefetch.maptasks	1.0	How many map tasks should be scheduled in-advance on a tasktracker. To be given in % of map slots. Default is 1.0 which means number of tasks overscheduled = total map slots on tasktracker.
mapreduce.tasktracker.reserved.physicalmemory.mb		Maximum phyiscal memory tasktracker should reserve for mapreduce tasks. If tasks use more than the limit, task using maximum memory will be killed. Expert only: Set this value iff tasktracker should use a certain amount of memory for mapreduce tasks. In MapR Distro warden figures this number based on services configured on a node. Setting mapreduce.tasktracker.reserved.physicalmemory.mb to -1 will disable physical memory accounting and task management.
mapreduce.tasktracker.volume.healthcheck.interval	60000	How often tasktracker should check for mapreduce volume at $<mapr.localvolumes.path>/mapred/. Value is in milliseconds.
mapreduce.use.fastreduce	false	Expert only. Reducer won't be able to tolerate failures.
mapreduce.use.maprfs	true	If true, then mapreduce uses maprfs to store task related data may be executed in parallel.

keep.failed.task.files	false	Should the files for failed tasks be kept. This should only be used on jobs that are failing, because the storage is never reclaimed. It also prevents the map outputs from being erased from the reduce directory as they are consumed.
keep.task.files.pattern	.*_m_123456_0	Keep all files from tasks whose task names match the given regular expression. Defaults to none.
tasktracker.http.threads	2	The number of worker threads that for the http server. This is used for map output fetching

mapred-site.xml

The file `/opt/mapr/hadoop/hadoop-<version>/conf/mapred-site.xml` specifies MapReduce formulas and parameters.

- `maprfs:///var/mapred/cluster/mapred/mapred-site.xml` - cluster-wide MapReduce configuration
- `/opt/mapr/hadoop/hadoop-<version>/conf/mapred-site.xml` - local MapReduce configuration on the node

Each parameter in the local configuration file overrides the corresponding parameter in the cluster-wide configuration unless the cluster-wide copy of the parameter includes `<final>true</final>`. In general, only job-specific parameters should be set in the local copy of `mapred-site.xml`.

There are three parts to `mapred-site.xml`:

- JobTracker configuration
- TaskTracker configuration
- Job configuration

Jobtracker Configuration

Should be changed by the administrator. When changing any parameters in this section, a JobTracker restart is required.

Parameter	Value	Description
mapred.job.tracker	maprfs:///	JobTracker address ip:port or use uri maprfs:/// for default cluster or maprfs:///mapr/san_jose_cluster1 to connect 'san_jose_cluster1' cluster. Replace localhost by one or more ip addresses for jobtracker.
mapred.jobtracker.port	9001	Port on which JobTracker listens. Read by JobTracker to start RPC Server.
mapreduce.tasktracker.outofband.heartbeat	false	Expert: Set this to true to let the tasktracker send an out-of-band heartbeat on task-completion for better latency.
webinterface.private.actions		If set to true, jobs can be killed from JT's web interface. Enable this option if the interfaces are only reachable by those who have the right authorization.

Jobtracker Directories

When changing any parameters in this section, a JobTracker restart is required.

Volume path = mapred.system.dir/../

Parameter	Value	Description
mapred.system.dir	/var/mapr/cluster/mapred/jobTracker/system	The shared directory where MapReduce stores control files.

mapred.job.tracker.persist.jobstatus.dir	/var/mapr/cluster/mapred/jobTracker/jobsInfo	The directory where the job status information is persisted in a file system to be available after it drops of the memory queue and between jobtracker restarts.
mapreduce.jobtracker.staging.root.dir	/var/mapr/cluster/mapred/jobTracker/staging	The root of the staging area for users' job files In practice, this should be the directory where users' home directories are located (usually /user)
mapreduce.job.split.metainfo.maxsize	10000000	The maximum permissible size of the split metainfo file. The JobTracker won't attempt to read split metainfo files bigger than the configured value. No limits if set to -1.
mapred.jobtracker.retiredjobs.cache.size	1000	The number of retired job status to keep in the cache.
mapred.job.tracker.history.completed.location	/var/mapr/cluster/mapred/jobTracker/history/done	The completed job history files are stored at this single well known location. If nothing is specified, the files are stored at ${hadoop.job.history.location}/done in local filesystem.
hadoop.job.history.location		If job tracker is static the history files are stored in this single well known place on local filesystem. If No value is set here, by default, it is in the local file system at ${hadoop.log.dir}/history. History files are moved to mapred.jobtracker.history.completed.location which is on MapRFs JobTracker volume.
mapred.jobtracker.jobhistory.lru.cache.size	5	The number of job history files loaded in memory. The jobs are loaded when they are first accessed. The cache is cleared based on LRU.

JobTracker Recovery

When changing any parameters in this section, a JobTracker restart is required.

Parameter	Value	Description
mapreduce.jobtracker.recovery.dir	/var/mapr/cluster/mapred/jobTracker/recovery	Recovery Directory. Stores list of known TaskTrackers.
mapreduce.jobtracker.recovery.maxtime	120	Maximum time in seconds JobTracker should stay in recovery mode.
mapred.jobtracker.restart.recover	true	"true" to enable (job) recovery upon restart, "false" to start afresh

Enable Fair Scheduler

When changing any parameters in this section, a JobTracker restart is required.

Parameter	Value	Description
mapred.fairscheduler.allocation.file	conf/pools.xml	
mapred.jobtracker.taskScheduler	org.apache.hadoop.mapred.FairScheduler	
mapred.fairscheduler.assignmultiple	true	
mapred.fairscheduler.eventlog.enabled	false	Enable scheduler logging in ${HADOOP_LOG_DIR}/fairscheduler/

mapred.fairscheduler.smalljob.schedule.enable	true	Enable small job fast scheduling inside fair scheduler. TaskTrackers should reserve a slot called ephemeral slot which is used for smalljob if cluster is busy.
mapred.fairscheduler.smalljob.max.maps	10	Small job definition. Max number of maps allowed in small job.
mapred.fairscheduler.smalljob.max.reducers	10	Small job definition. Max number of reducers allowed in small job.
mapred.fairscheduler.smalljob.max.inputsize	10737418240	Small job definition. Max input size in bytes allowed for a small job. Default is 10GB.
mapred.fairscheduler.smalljob.max.reducer.inputsize	1073741824	Small job definition. Max estimated input size for a reducer allowed in small job. Default is 1GB per reducer.
mapred.cluster.ephemeral.tasks.memory.limit.mb	200	Small job definition. Max memory in mbytes reserved for an ephermal slot. Default is 200mb. This value must be same on JobTracker and TaskTracker nodes.

TaskTracker Configuration

When changing any parameters in this section, a TaskTracker restart is required.

Should be changed by admin

Parameter	Value	Description
mapred.tasktracker.map.tasks.maximum	(CPUS > 2) ? (CPUS * 0.75) : 1	The maximum number of map tasks that will be run simultaneously by a task tracker.
mapreduce.tasktracker.prefetch.maptasks	1.0	How many map tasks should be scheduled in-advance on a tasktracker. To be given in % of map slots. Default is 1.0 which means number of tasks overscheduled = total map slots on TT.
mapred.tasktracker.reduce.tasks.maximum	(CPUS > 2) ? (CPUS * 0.50): 1	The maximum number of reduce tasks that will be run simultaneously by a task tracker.
mapred.tasktracker.ephemeral.tasks.maximum	1	Reserved slot for small job scheduling
mapred.tasktracker.ephemeral.tasks.timeout	10000	Maximum time in ms a task is allowed to occupy ephemeral slot
mapred.tasktracker.ephemeral.tasks.ulimit	4294967296>	Ulimit (bytes) on all tasks scheduled on an ephemeral slot
mapreduce.tasktracker.reserved.physicalmemory.mb		Maximum phyiscal memory tasktracker should reserve for mapreduce tasks. If tasks use more than the limit, task using maximum memory will be killed. Expert only: Set this value iff tasktracker should use a certain amount of memory for mapreduce tasks. In MapR Distro warden figures this number based on services configured on a node. Setting mapreduce.tasktracker.reserved.physicalmemory.mb to -1 will disable physical memory accounting and task management.

mapreduce.tasktracker.heapbased.memory.management	false	Expert only: If admin wants to prevent swapping by not launching too many tasks use this option. Task's memory usage is based on max java heap size (-Xmx). By default -Xmx will be computed by tasktracker based on slots and memory reserved for mapreduce tasks. See mapred.map.child.java.opts/mapred.reduce.child.java.opts.
mapreduce.tasktracker.jvm.idle.time	10000	If jvm is idle for more than mapreduce.tasktracker.jvm.idle.time (milliseconds) tasktracker will kill it.

Job Configuration

Users should set these values on the node from which you plan to submit jobs, before submitting the jobs. If you are using Hadoop examples, you can set these parameters from the command line. Example:

```
hadoop jar hadoop-examples.jar terasort -Dmapred.map.child.java.opts="-Xmx1000m"
```

When you submit a job, the JobClient creates `job.xml` by reading parameters from the following files in the following order:

1. `mapred-default.xml`
2. The local `mapred-site.xml` - overrides identical parameters in `mapred-default.xml`
3. Any settings in the job code itself - overrides identical parameters in `mapred-site.xml`

Parameter	Value	Description
keep.failed.task.files	false	Should the files for failed tasks be kept. This should only be used on jobs that are failing, because the storage is never reclaimed. It also prevents the map outputs from being erased from the reduce directory as they are consumed.
mapred.job.reuse.jvm.num.tasks	-1	How many tasks to run per jvm. If set to -1, there is no limit.
mapred.map.tasks.speculative.execution	true	If true, then multiple instances of some map tasks may be executed in parallel.
mapred.reduce.tasks.speculative.execution	true	If true, then multiple instances of some reduce tasks may be executed in parallel.
mapred.job.map.memory.physical.mb		Maximum physical memory limit for map task of this job. If limit is exceeded task attempt will be FAILED.
mapred.job.reduce.memory.physical.mb		Maximum physical memory limit for reduce task of this job. If limit is exceeded task attempt will be FAILED.
mapreduce.task.classpath.user.precedence	false	Set to true if user wants to set different classpath.
mapred.max.maps.per.node	-1	Per-node limit on running map tasks for the job. A value of -1 signifies no limit.
mapred.max.reduces.per.node	-1	Per-node limit on running reduce tasks for the job. A value of -1 signifies no limit.
mapred.running.map.limit	-1	Cluster-wide limit on running map tasks for the job. A value of -1 signifies no limit.

mapred.running.reduce.limit	-1	Cluster-wide limit on running reduce tasks for the job. A value of -1 signifies no limit.
mapred.reduce.child.java.opts	-XX:ErrorFile=/opt/cores/mapreduce_java_error%p.log	Java opts for the reduce tasks. Default heapsize(-Xmx) is determined by memory reserved for mapreduce at tasktracker. Reduce task is given more memory than map task. Default memory for a reduce task = (Total Memory reserved for mapreduce) * (2*#reduceslots / (#mapslots + 2*#reduceslots))
mapred.reduce.child.ulimit		
io.sort.mb		Buffer used to hold map outputs in memory before writing final map outputs. Setting this value very low may cause spills. By default if left empty value is set to 50% of heapsize for map. If a average input to map is "MapIn" bytes then typically value of io.sort.mb should be '1.25 times MapIn' bytes.
io.sort.factor	256	The number of streams to merge at once while sorting files. This determines the number of open file handles.
io.sort.record.percent	0.17	The percentage of io.sort.mb dedicated to tracking record boundaries. Let this value be r, io.sort.mb be x. The maximum number of records collected before the collection thread must block is equal to (r * x) / 4
mapred.reduce.slowstart.completed.maps	0.95	Fraction of the number of maps in the job which should be complete before reduces are scheduled for the job.
mapreduce.reduce.input.limit	-1	The limit on the input size of the reduce. If the estimated input size of the reduce is greater than this value, job is failed. A value of -1 means that there is no limit set.
mapred.reduce.parallel.copies	12	The default number of parallel transfers run by reduce during the copy(shuffle) phase.

Oozie

Parameter	Value	Description
hadoop.proxyuser.root.hosts	*	comma separated ips/hostnames running Oozie server
hadoop.proxyuser.mapr.groups	mapr,staff	
hadoop.proxyuser.root.groups	root	

taskcontroller.cfg

The file `/opt/mapr/hadoop/hadoop-<version>/conf/taskcontroller.cfg` specifies TaskTracker configuration parameters. The parameters should be set the same on all TaskTracker nodes. See also Secured TaskTracker.

Parameter	Value	Description
mapred.local.dir	/tmp/mapr-hadoop/mapred/local	The local MapReduce directory.
hadoop.log.dir	/opt/mapr/hadoop/hadoop-0.20.2/bin/../logs	The Hadoop log directory.

mapreduce.tasktracker.group	root	The group that is allowed to submit jobs.
min.user.id	-1	The minimum user ID for submitting jobs: • Set to `0` to disallow `root` from submitting jobs • Set to `1000` to disallow all superusers from submitting jobs
banned.users	(not present by default)	Add this parameter with a comma-separated list of usernames to ban certain users from submitting jobs

Hadoop Compatibility in This Release

Greenplum HD EE provides the following packages:

- Apache Hadoop 0.20.2
- hbase-0.90.2
- hive-0.7.0
- pig-0.8
- sqoop-1.2.0

Hadoop Common Patches

Greenplum HD EE 1.0 includes the following Apache Hadoop issues that are not included in the Apache Hadoop base version 0.20.2:

[HADOOP-1722] Make streaming to handle non-utf8 byte array
[HADOOP-1849] IPC server max queue size should be configurable
[HADOOP-2141] speculative execution start up condition based on completion time
[HADOOP-2366] Space in the value for dfs.data.dir can cause great problems
[HADOOP-2721] Use job control for tasks (and therefore for pipes and streaming)
[HADOOP-2838] Add HADOOP_LIBRARY_PATH config setting so Hadoop will include external directories for jni
[HADOOP-3327] Shuffling fetchers waited too long between map output fetch re-tries
[HADOOP-3659] Patch to allow hadoop native to compile on Mac OS X
[HADOOP-4012] Providing splitting support for bzip2 compressed files
[HADOOP-4041] IsolationRunner does not work as documented
[HADOOP-4490] Map and Reduce tasks should run as the user who submitted the job
[HADOOP-4655] FileSystem.CACHE should be ref-counted
[HADOOP-4656] Add a user to groups mapping service
[HADOOP-4675] Current Ganglia metrics implementation is incompatible with Ganglia 3.1
[HADOOP-4829] Allow FileSystem shutdown hook to be disabled
[HADOOP-4842] Streaming combiner should allow command, not just JavaClass
[HADOOP-4930] Implement setuid executable for Linux to assist in launching tasks as job owners
[HADOOP-4933] ConcurrentModificationException in JobHistory.java
[HADOOP-5170] Set max map/reduce tasks on a per-job basis, either per-node or cluster-wide
[HADOOP-5175] Option to prohibit jars unpacking
[HADOOP-5203] TT's version build is too restrictive
[HADOOP-5396] Queue ACLs should be refreshed without requiring a restart of the job tracker
[HADOOP-5419] Provide a way for users to find out what operations they can do on which M/R queues
[HADOOP-5420] Support killing of process groups in LinuxTaskController binary
[HADOOP-5442] The job history display needs to be paged
[HADOOP-5450] Add support for application-specific typecodes to typed bytes
[HADOOP-5469] Exposing Hadoop metrics via HTTP
[HADOOP-5476] calling new SequenceFile.Reader(...) leaves an InputStream open, if the given sequence file is broken
[HADOOP-5488] HADOOP-2721 doesn't clean up descendant processes of a jvm that exits cleanly after running a task successfully
[HADOOP-5528] Binary partitioner
[HADOOP-5582] Hadoop Vaidya throws number format exception due to changes in the job history counters string format (escaped compact representation).
[HADOOP-5592] Hadoop Streaming - GzipCodec
[HADOOP-5613] change S3Exception to checked exception
[HADOOP-5643] Ability to blacklist tasktracker
[HADOOP-5656] Counter for S3N Read Bytes does not work
[HADOOP-5675] DistCp should not launch a job if it is not necessary
[HADOOP-5733] Add map/reduce slot capacity and lost map/reduce slot capacity to JobTracker metrics
[HADOOP-5737] UGI checks in testcases are broken
[HADOOP-5738] Split waiting tasks field in JobTracker metrics to individual tasks
[HADOOP-5745] Allow setting the default value of maxRunningJobs for all pools

[HADOOP-5784] The length of the heartbeat cycle should be configurable.
[HADOOP-5801] JobTracker should refresh the hosts list upon recovery
[HADOOP-5805] problem using top level s3 buckets as input/output directories
[HADOOP-5861] s3n files are not getting split by default
[HADOOP-5879] GzipCodec should read compression level etc from configuration
[HADOOP-5913] Allow administrators to be able to start and stop queues
[HADOOP-5958] Use JDK 1.6 File APIs in DF.java wherever possible
[HADOOP-5976] create script to provide classpath for external tools
[HADOOP-5980] LD_LIBRARY_PATH not passed to tasks spawned off by LinuxTaskController
[HADOOP-5981] HADOOP-2838 doesnt work as expected
[HADOOP-6132] RPC client opens an extra connection for VersionedProtocol
[HADOOP-6133] ReflectionUtils performance regression
[HADOOP-6148] Implement a pure Java CRC32 calculator
[HADOOP-6161] Add get/setEnum to Configuration
[HADOOP-6166] Improve PureJavaCrc32
[HADOOP-6184] Provide a configuration dump in json format.
[HADOOP-6227] Configuration does not lock parameters marked final if they have no value.
[HADOOP-6234] Permission configuration files should use octal and symbolic
[HADOOP-6254] s3n fails with SocketTimeoutException
[HADOOP-6269] Missing synchronization for defaultResources in Configuration.addResource
[HADOOP-6279] Add JVM memory usage to JvmMetrics
[HADOOP-6284] Any hadoop commands crashing jvm (SIGBUS) when /tmp (tmpfs) is full
[HADOOP-6299] Use JAAS LoginContext for our login
[HADOOP-6312] Configuration sends too much data to log4j
[HADOOP-6337] Update FilterInitializer class to be more visible and take a conf for further development
[HADOOP-6343] Stack trace of any runtime exceptions should be recorded in the server logs.
[HADOOP-6400] Log errors getting Unix UGI
[HADOOP-6408] Add a /conf servlet to dump running configuration
[HADOOP-6419] Change RPC layer to support SASL based mutual authentication
[HADOOP-6433] Add AsyncDiskService that is used in both hdfs and mapreduce
[HADOOP-6441] Prevent remote CSS attacks in Hostname and UTF-7.
[HADOOP-6453] Hadoop wrapper script shouldn't ignore an existing JAVA_LIBRARY_PATH
[HADOOP-6471] StringBuffer -> StringBuilder - conversion of references as necessary
[HADOOP-6496] HttpServer sends wrong content-type for CSS files (and others)
[HADOOP-6510] doAs for proxy user
[HADOOP-6521] FsPermission:SetUMask not updated to use new-style umask setting.
[HADOOP-6534] LocalDirAllocator should use whitespace trimming configuration getters
[HADOOP-6543] Allow authentication-enabled RPC clients to connect to authentication-disabled RPC servers
[HADOOP-6558] archive does not work with distcp -update
[HADOOP-6568] Authorization for default servlets
[HADOOP-6569] FsShell#cat should avoid calling unecessary getFileStatus before opening a file to read
[HADOOP-6572] RPC responses may be out-of-order with respect to SASL
[HADOOP-6577] IPC server response buffer reset threshold should be configurable
[HADOOP-6578] Configuration should trim whitespace around a lot of value types
[HADOOP-6599] Split RPC metrics into summary and detailed metrics
[HADOOP-6609] Deadlock in DFSClient#getBlockLocations even with the security disabled
[HADOOP-6613] RPC server should check for version mismatch first
[HADOOP-6627] "Bad Connection to FS" message in FSShell should print message from the exception
[HADOOP-6631] FileUtil.fullyDelete() should continue to delete other files despite failure at any level.
[HADOOP-6634] AccessControlList uses full-principal names to verify acls causing queue-acls to fail
[HADOOP-6637] Benchmark overhead of RPC session establishment
[HADOOP-6640] FileSystem.get() does RPC retries within a static synchronized block
[HADOOP-6644] util.Shell getGROUPS_FOR_USER_COMMAND method name - should use common naming convention
[HADOOP-6649] login object in UGI should be inside the subject
[HADOOP-6652] ShellBasedUnixGroupsMapping shouldn't have a cache
[HADOOP-6653] NullPointerException in setupSaslConnection when browsing directories
[HADOOP-6663] BlockDecompressorStream get EOF exception when decompressing the file compressed from empty file
[HADOOP-6667] RPC.waitForProxy should retry through NoRouteToHostException
[HADOOP-6669] zlib.compress.level ignored for DefaultCodec initialization
[HADOOP-6670] UserGroupInformation doesn't support use in hash tables
[HADOOP-6674] Performance Improvement in Secure RPC
[HADOOP-6687] user object in the subject in UGI should be reused in case of a relogin.
[HADOOP-6701] Incorrect exit codes for "dfs -chown", "dfs -chgrp"
[HADOOP-6706] Relogin behavior for RPC clients could be improved
[HADOOP-6710] Symbolic umask for file creation is not consistent with posix
[HADOOP-6714] FsShell 'hadoop fs -text' does not support compression codecs
[HADOOP-6718] Client does not close connection when an exception happens during SASL negotiation
[HADOOP-6722] NetUtils.connect should check that it hasn't connected a socket to itself
[HADOOP-6723] unchecked exceptions thrown in IPC Connection orphan clients
[HADOOP-6724] IPC doesn't properly handle IOEs thrown by socket factory
[HADOOP-6745] adding some java doc to Server.RpcMetrics, UGI
[HADOOP-6757] NullPointerException for hadoop clients launched from streaming tasks
[HADOOP-6760] WebServer shouldn't increase port number in case of negative port setting caused by Jetty's race
[HADOOP-6762] exception while doing RPC I/O closes channel
[HADOOP-6776] UserGroupInformation.createProxyUser's javadoc is broken
[HADOOP-6813] Add a new newInstance method in FileSystem that takes a "user" as argument

[HADOOP-6815] refreshSuperUserGroupsConfiguration should use server side configuration for the refresh
[HADOOP-6818] Provide a JNI-based implementation of GroupMappingServiceProvider
[HADOOP-6832] Provide a web server plugin that uses a static user for the web UI
[HADOOP-6833] IPC leaks call parameters when exceptions thrown
[HADOOP-6859] Introduce additional statistics to FileSystem
[HADOOP-6864] Provide a JNI-based implementation of ShellBasedUnixGroupsNetgroupMapping (implementation of GroupMappingServiceProvider)
[HADOOP-6881] The efficient comparators aren't always used except for BytesWritable and Text
[HADOOP-6899] RawLocalFileSystem#setWorkingDir() does not work for relative names
[HADOOP-6907] Rpc client doesn't use the per-connection conf to figure out server's Kerberos principal
[HADOOP-6925] BZip2Codec incorrectly implements read()
[HADOOP-6928] Fix BooleanWritable comparator in 0.20
[HADOOP-6943] The GroupMappingServiceProvider interface should be public
[HADOOP-6950] Suggest that HADOOP_CLASSPATH should be preserved in hadoop-env.sh.template
[HADOOP-6995] Allow wildcards to be used in ProxyUsers configurations
[HADOOP-7082] Configuration.writeXML should not hold lock while outputting
[HADOOP-7101] UserGroupInformation.getCurrentUser() fails when called from non-Hadoop JAAS context
[HADOOP-7104] Remove unnecessary DNS reverse lookups from RPC layer
[HADOOP-7110] Implement chmod with JNI
[HADOOP-7114] FsShell should dump all exceptions at DEBUG level
[HADOOP-7115] Add a cache for getpwuid_r and getpwgid_r calls
[HADOOP-7118] NPE in Configuration.writeXml
[HADOOP-7122] Timed out shell commands leak Timer threads
[HADOOP-7156] getpwuid_r is not thread-safe on RHEL6
[HADOOP-7172] SecureIO should not check owner on non-secure clusters that have no native support
[HADOOP-7173] Remove unused fstat() call from NativeIO
[HADOOP-7183] WritableComparator.get should not cache comparator objects
[HADOOP-7184] Remove deprecated local.cache.size from core-default.xml

MapReduce Patches

Greenplum HD EE 1.0 includes the following Apache MapReduce issues that are not included in the Apache Hadoop base version 0.20.2:

[MAPREDUCE-112] Reduce Input Records and Reduce Output Records counters are not being set when using the new Mapreduce reducer API
[MAPREDUCE-118] Job.getJobID() will always return null
[MAPREDUCE-144] TaskMemoryManager should log process-tree's status while killing tasks.
[MAPREDUCE-181] Secure job submission
[MAPREDUCE-211] Provide a node health check script and run it periodically to check the node health status
[MAPREDUCE-220] Collecting cpu and memory usage for MapReduce tasks
[MAPREDUCE-270] TaskTracker could send an out-of-band heartbeat when the last running map/reduce completes
[MAPREDUCE-277] Job history counters should be avaible on the UI.
[MAPREDUCE-339] JobTracker should give preference to failed tasks over virgin tasks so as to terminate the job ASAP if it is eventually going to fail.
[MAPREDUCE-364] Change org.apache.hadoop.examples.MultiFileWordCount to use new mapreduce api.
[MAPREDUCE-369] Change org.apache.hadoop.mapred.lib.MultipleInputs to use new api.
[MAPREDUCE-370] Change org.apache.hadoop.mapred.lib.MultipleOutputs to use new api.
[MAPREDUCE-415] JobControl Job does always has an unassigned name
[MAPREDUCE-416] Move the completed jobs' history files to a DONE subdirectory inside the configured history directory
[MAPREDUCE-461] Enable ServicePlugins for the JobTracker
[MAPREDUCE-463] The job setup and cleanup tasks should be optional
[MAPREDUCE-467] Collect information about number of tasks succeeded / total per time unit for a tasktracker.
[MAPREDUCE-476] extend DistributedCache to work locally (LocalJobRunner)
[MAPREDUCE-478] separate jvm param for mapper and reducer
[MAPREDUCE-516] Fix the 'cluster drain' problem in the Capacity Scheduler wrt High RAM Jobs
[MAPREDUCE-517] The capacity-scheduler should assign multiple tasks per heartbeat
[MAPREDUCE-521] After JobTracker restart Capacity Schduler does not schedules pending tasks from already running tasks.
[MAPREDUCE-532] Allow admins of the Capacity Scheduler to set a hard-limit on the capacity of a queue
[MAPREDUCE-551] Add preemption to the fair scheduler
[MAPREDUCE-572] If #link is missing from uri format of -cacheArchive then streaming does not throw error.
[MAPREDUCE-655] Change KeyValueLineRecordReader and KeyValueTextInputFormat to use new api.
[MAPREDUCE-676] Existing diagnostic rules fail for MAP ONLY jobs
[MAPREDUCE-679] XML-based metrics as JSP servlet for JobTracker
[MAPREDUCE-680] Reuse of Writable objects is improperly handled by MRUnit
[MAPREDUCE-682] Reserved tasktrackers should be removed when a node is globally blacklisted
[MAPREDUCE-693] Conf files not moved to "done" subdirectory after JT restart
[MAPREDUCE-698] Per-pool task limits for the fair scheduler
[MAPREDUCE-706] Support for FIFO pools in the fair scheduler
[MAPREDUCE-707] Provide a jobconf property for explicitly assigning a job to a pool
[MAPREDUCE-709] node health check script does not display the correct message on timeout
[MAPREDUCE-714] JobConf.findContainingJar unescapes unnecessarily on Linux
[MAPREDUCE-716] org.apache.hadoop.mapred.lib.db.DBInputformat not working with oracle
[MAPREDUCE-722] More slots are getting reserved for HiRAM job tasks then required
[MAPREDUCE-732] node health check script should not log "UNHEALTHY" status for every heartbeat in INFO mode

[MAPREDUCE-734] java.util.ConcurrentModificationException observed in unreserving slots for HiRam Jobs
[MAPREDUCE-739] Allow relative paths to be created inside archives.
[MAPREDUCE-740] Provide summary information per job once a job is finished.
[MAPREDUCE-744] Support in DistributedCache to share cache files with other users after HADOOP-4493
[MAPREDUCE-754] NPE in expiry thread when a TT is lost
[MAPREDUCE-764] TypedBytesInput's readRaw() does not preserve custom type codes
[MAPREDUCE-768] Configuration information should generate dump in a standard format.
[MAPREDUCE-771] Setup and cleanup tasks remain in UNASSIGNED state for a long time on tasktrackers with long running high RAM tasks
[MAPREDUCE-782] Use PureJavaCrc32 in mapreduce spills
[MAPREDUCE-787] -files, -archives should honor user given symlink path
[MAPREDUCE-809] Job summary logs show status of completed jobs as RUNNING
[MAPREDUCE-814] Move completed Job history files to HDFS
[MAPREDUCE-817] Add a cache for retired jobs with minimal job info and provide a way to access history file url
[MAPREDUCE-825] JobClient completion poll interval of 5s causes slow tests in local mode
[MAPREDUCE-840] DBInputFormat leaves open transaction
[MAPREDUCE-842] Per-job local data on the TaskTracker node should have right access-control
[MAPREDUCE-856] Localized files from DistributedCache should have right access-control
[MAPREDUCE-871] Job/Task local files have incorrect group ownership set by LinuxTaskController binary
[MAPREDUCE-875] Make DBRecordReader execute queries lazily
[MAPREDUCE-885] More efficient SQL queries for DBInputFormat
[MAPREDUCE-890] After HADOOP-4491, the user who started mapred system is not able to run job.
[MAPREDUCE-896] Users can set non-writable permissions on temporary files for TT and can abuse disk usage.
[MAPREDUCE-899] When using LinuxTaskController, localized files may become accessible to unintended users if permissions are misconfigured.
[MAPREDUCE-927] Cleanup of task-logs should happen in TaskTracker instead of the Child
[MAPREDUCE-947] OutputCommitter should have an abortJob method
[MAPREDUCE-964] Inaccurate values in jobSummary logs
[MAPREDUCE-967] TaskTracker does not need to fully unjar job jars
[MAPREDUCE-968] NPE in distcp encountered when placing _logs directory on S3FileSystem
[MAPREDUCE-971] distcp does not always remove distcp.tmp.dir
[MAPREDUCE-1028] Cleanup tasks are scheduled using high memory configuration, leaving tasks in unassigned state.
[MAPREDUCE-1030] Reduce tasks are getting starved in capacity scheduler
[MAPREDUCE-1048] Show total slot usage in cluster summary on jobtracker webui
[MAPREDUCE-1059] distcp can generate uneven map task assignments
[MAPREDUCE-1083] Use the user-to-groups mapping service in the JobTracker
[MAPREDUCE-1085] For tasks, "ulimit -v -1" is being run when user doesn't specify mapred.child.ulimit
[MAPREDUCE-1086] hadoop commands in streaming tasks are trying to write to tasktracker's log
[MAPREDUCE-1088] JobHistory files should have narrower 0600 perms
[MAPREDUCE-1089] Fair Scheduler preemption triggers NPE when tasks are scheduled but not running
[MAPREDUCE-1090] Modify log statement in Tasktracker log related to memory monitoring to include attempt id.
[MAPREDUCE-1098] Incorrect synchronization in DistributedCache causes TaskTrackers to freeze up during localization of Cache for tasks.
[MAPREDUCE-1100] User's task-logs filling up local disks on the TaskTrackers
[MAPREDUCE-1103] Additional JobTracker metrics
[MAPREDUCE-1105] CapacityScheduler: It should be possible to set queue hard-limit beyond it's actual capacity
[MAPREDUCE-1118] Capacity Scheduler scheduling information is hard to read / should be tabular format
[MAPREDUCE-1131] Using profilers other than hprof can cause JobClient to report job failure
[MAPREDUCE-1140] Per cache-file refcount can become negative when tasks release distributed-cache files
[MAPREDUCE-1143] runningMapTasks counter is not properly decremented in case of failed Tasks.
[MAPREDUCE-1155] Streaming tests swallow exceptions
[MAPREDUCE-1158] running_maps is not decremented when the tasks of a job is killed/failed
[MAPREDUCE-1160] Two log statements at INFO level fill up jobtracker logs
[MAPREDUCE-1171] Lots of fetch failures
[MAPREDUCE-1178] MultipleInputs fails with ClassCastException
[MAPREDUCE-1185] URL to JT webconsole for running job and job history should be the same
[MAPREDUCE-1186] While localizing a DistributedCache file, TT sets permissions recursively on the whole base-dir
[MAPREDUCE-1196] MAPREDUCE-947 incompatibly changed FileOutputCommitter
[MAPREDUCE-1198] Alternatively schedule different types of tasks in fair share scheduler
[MAPREDUCE-1213] TaskTrackers restart is very slow because it deletes distributed cache directory synchronously
[MAPREDUCE-1219] JobTracker Metrics causes undue load on JobTracker
[MAPREDUCE-1221] Kill tasks on a node if the free physical memory on that machine falls below a configured threshold
[MAPREDUCE-1231] Distcp is very slow
[MAPREDUCE-1250] Refactor job token to use a common token interface
[MAPREDUCE-1258] Fair scheduler event log not logging job info
[MAPREDUCE-1285] DistCp cannot handle -delete if destination is local filesystem
[MAPREDUCE-1288] DistributedCache localizes only once per cache URI
[MAPREDUCE-1293] AutoInputFormat doesn't work with non-default FileSystems
[MAPREDUCE-1302] TrackerDistributedCacheManager can delete file asynchronously
[MAPREDUCE-1304] Add counters for task time spent in GC
[MAPREDUCE-1307] Introduce the concept of Job Permissions
[MAPREDUCE-1313] NPE in FieldFormatter if escape character is set and field is null
[MAPREDUCE-1316] JobTracker holds stale references to retired jobs via unreported tasks
[MAPREDUCE-1342] Potential JT deadlock in faulty TT tracking
[MAPREDUCE-1354] Incremental enhancements to the JobTracker for better scalability
[MAPREDUCE-1372] ConcurrentModificationException in JobInProgress

[MAPREDUCE-1378] Args in job details links on jobhistory.jsp are not URL encoded
[MAPREDUCE-1382] MRAsyncDiscService should tolerate missing local.dir
[MAPREDUCE-1397] NullPointerException observed during task failures
[MAPREDUCE-1398] TaskLauncher remains stuck on tasks waiting for free nodes even if task is killed.
[MAPREDUCE-1399] The archive command shows a null error message
[MAPREDUCE-1403] Save file-sizes of each of the artifacts in DistributedCache in the JobConf
[MAPREDUCE-1421] LinuxTaskController tests failing on trunk after the commit of MAPREDUCE-1385
[MAPREDUCE-1422] Changing permissions of files/dirs under job-work-dir may be needed sothat cleaning up of job-dir in all mapred-local-directories succeeds always
[MAPREDUCE-1423] Improve performance of CombineFileInputFormat when multiple pools are configured
[MAPREDUCE-1425] archive throws OutOfMemoryError
[MAPREDUCE-1435] symlinks in cwd of the task are not handled properly after MAPREDUCE-896
[MAPREDUCE-1436] Deadlock in preemption code in fair scheduler
[MAPREDUCE-1440] MapReduce should use the short form of the user names
[MAPREDUCE-1441] Configuration of directory lists should trim whitespace
[MAPREDUCE-1442] StackOverflowError when JobHistory parses a really long line
[MAPREDUCE-1443] DBInputFormat can leak connections
[MAPREDUCE-1454] The servlets should quote server generated strings sent in the response
[MAPREDUCE-1455] Authorization for servlets
[MAPREDUCE-1457] For secure job execution, couple of more UserGroupInformation.doAs needs to be added
[MAPREDUCE-1464] In JobTokenIdentifier change method getUsername to getUser which returns UGI
[MAPREDUCE-1466] FileInputFormat should save #input-files in JobConf
[MAPREDUCE-1476] committer.needsTaskCommit should not be called for a task cleanup attempt
[MAPREDUCE-1480] CombineFileRecordReader does not properly initialize child RecordReader
[MAPREDUCE-1493] Authorization for job-history pages
[MAPREDUCE-1503] Push HADOOP-6551 into MapReduce
[MAPREDUCE-1505] Cluster class should create the rpc client only when needed
[MAPREDUCE-1521] Protection against incorrectly configured reduces
[MAPREDUCE-1522] FileInputFormat may change the file system of an input path
[MAPREDUCE-1526] Cache the job related information while submitting the job , this would avoid many RPC calls to JobTracker.
[MAPREDUCE-1533] Reduce or remove usage of String.format() usage in CapacityTaskScheduler.updateQSIObjects and Counters.makeEscapedString()
[MAPREDUCE-1538] TrackerDistributedCacheManager can fail because the number of subdirectories reaches system limit
[MAPREDUCE-1543] Log messages of JobACLsManager should use security logging of HADOOP-6586
[MAPREDUCE-1545] Add 'first-task-launched' to job-summary
[MAPREDUCE-1550] UGI.doAs should not be used for getting the history file of jobs
[MAPREDUCE-1563] Task diagnostic info would get missed sometimes.
[MAPREDUCE-1570] Shuffle stage - Key and Group Comparators
[MAPREDUCE-1607] Task controller may not set permissions for a task cleanup attempt's log directory
[MAPREDUCE-1609] TaskTracker.localizeJob should not set permissions on job log directory recursively
[MAPREDUCE-1611] Refresh nodes and refresh queues doesnt work with service authorization enabled
[MAPREDUCE-1612] job conf file is not accessible from job history web page
[MAPREDUCE-1621] Streaming's TextOutputReader.getLastOutput throws NPE if it has never read any output
[MAPREDUCE-1635] ResourceEstimator does not work after MAPREDUCE-842
[MAPREDUCE-1641] Job submission should fail if same uri is added for mapred.cache.files and mapred.cache.archives
[MAPREDUCE-1656] JobStory should provide queue info.
[MAPREDUCE-1657] After task logs directory is deleted, tasklog servlet displays wrong error message about job ACLs
[MAPREDUCE-1664] Job Acls affect Queue Acls
[MAPREDUCE-1680] Add a metrics to track the number of heartbeats processed
[MAPREDUCE-1682] Tasks should not be scheduled after tip is killed/failed.
[MAPREDUCE-1683] Remove JNI calls from ClusterStatus cstr
[MAPREDUCE-1699] JobHistory shouldn't be disabled for any reason
[MAPREDUCE-1707] TaskRunner can get NPE in getting ugi from TaskTracker
[MAPREDUCE-1716] Truncate logs of finished tasks to prevent node thrash due to excessive logging
[MAPREDUCE-1733] Authentication between pipes processes and java counterparts.
[MAPREDUCE-1734] Un-deprecate the old MapReduce API in the 0.20 branch
[MAPREDUCE-1744] DistributedCache creates its own FileSytem instance when adding a file/archive to the path
[MAPREDUCE-1754] Replace mapred.persmissions.supergroup with an acl : mapreduce.cluster.administrators
[MAPREDUCE-1759] Exception message for unauthorized user doing killJob, killTask, setJobPriority needs to be improved
[MAPREDUCE-1778] CompletedJobStatusStore initialization should fail if {mapred.job.tracker.persist.jobstatus.dir} is unwritable
[MAPREDUCE-1784] IFile should check for null compressor
[MAPREDUCE-1785] Add streaming config option for not emitting the key
[MAPREDUCE-1832] Support for file sizes less than 1MB in DFSIO benchmark.
[MAPREDUCE-1845] FairScheduler.tasksToPeempt() can return negative number
[MAPREDUCE-1850] Include job submit host information (name and ip) in jobconf and jobdetails display
[MAPREDUCE-1853] MultipleOutputs does not cache TaskAttemptContext
[MAPREDUCE-1868] Add read timeout on userlog pull
[MAPREDUCE-1872] Re-think (user|queue) limits on (tasks|jobs) in the CapacityScheduler
[MAPREDUCE-1887] MRAsyncDiskService does not properly absolutize volume root paths
[MAPREDUCE-1900] MapReduce daemons should close FileSystems that are not needed anymore
[MAPREDUCE-1914] TrackerDistributedCacheManager never cleans its input directories
[MAPREDUCE-1938] Ability for having user's classes take precedence over the system classes for tasks' classpath
[MAPREDUCE-1960] Limit the size of jobconf.
[MAPREDUCE-1961] ConcurrentModificationException when shutting down Gridmix
[MAPREDUCE-1985] java.lang.ArrayIndexOutOfBoundsException in analysejobhistory.jsp of jobs with 0 maps
[MAPREDUCE-2023] TestDFSIO read test may not read specified bytes.

[MAPREDUCE-2082] Race condition in writing the jobtoken password file when launching pipes jobs
[MAPREDUCE-2096] Secure local filesystem IO from symlink vulnerabilities
[MAPREDUCE-2103] task-controller shouldn't require o-r permissions
[MAPREDUCE-2157] safely handle InterruptedException and interrupted status in MR code
[MAPREDUCE-2178] Race condition in LinuxTaskController permissions handling
[MAPREDUCE-2219] JT should not try to remove mapred.system.dir during startup
[MAPREDUCE-2234] If Localizer can't create task log directory, it should fail on the spot
[MAPREDUCE-2235] JobTracker "over-synchronization" makes it hang up in certain cases
[MAPREDUCE-2242] LinuxTaskController doesn't properly escape environment variables
[MAPREDUCE-2253] Servlets should specify content type
[MAPREDUCE-2256] FairScheduler fairshare preemption from multiple pools may preempt all tasks from one pool causing that pool to go below fairshare.
[MAPREDUCE-2289] Permissions race can make getStagingDir fail on local filesystem
[MAPREDUCE-2321] TT should fail to start on secure cluster when SecureIO isn't available
[MAPREDUCE-2323] Add metrics to the fair scheduler
[MAPREDUCE-2328] memory-related configurations missing from mapred-default.xml
[MAPREDUCE-2332] Improve error messages when MR dirs on local FS have bad ownership
[MAPREDUCE-2351] mapred.job.tracker.history.completed.location should support an arbitrary filesystem URI
[MAPREDUCE-2353] Make the MR changes to reflect the API changes in SecureIO library
[MAPREDUCE-2356] A task succeeded even though there were errors on all attempts.
[MAPREDUCE-2364] Shouldn't hold lock on rjob while localizing resources.
[MAPREDUCE-2366] TaskTracker can't retrieve stdout and stderr from web UI
[MAPREDUCE-2371] TaskLogsTruncater does not need to check log ownership when running as Child
[MAPREDUCE-2372] TaskLogAppender mechanism shouldn't be set in log4j.properties
[MAPREDUCE-2373] When tasks exit with a nonzero exit status, task runner should log the stderr as well as stdout
[MAPREDUCE-2374] Should not use PrintWriter to write taskjvm.sh
[MAPREDUCE-2377] task-controller fails to parse configuration if it doesn't end in \n
[MAPREDUCE-2379] Distributed cache sizing configurations are missing from mapred-default.xml

API Reference

Overview

This guide provides information about the Greenplum HD EE command API. Most commands can be run on the command-line interface (CLI), or by making REST requests programmatically or in a browser. To run CLI commands, use a Client machine or an ssh connection to any node in the cluster. To use the REST interface, make HTTP requests to a node that is running the WebServer service.

Each command reference page includes the command syntax, a table that describes the parameters, and examples of command usage. In each parameter table, required parameters are in **bold** text. For output commands, the reference pages include tables that describe the output fields. Values that do not apply to particular combinations are marked **NA**.

REST API Syntax

Greenplum HD EE REST calls use the following format:

`https://<host>:<port>/rest/<command>[/<subcommand>...]?<parameters>`

Construct the `<parameters>` list from the required and optional parameters, in the format `<parameter>=<value>` separated by the ampersand (&) character. Example:

`https://r1n1.qa.sj.ca.us:8443/api/volume/mount?name=test-volume&path=/test`

Values in REST API calls must be URL-encoded. For readability, the values in this document are presented using the actual characters, rather than the URL-encoded versions.

Authentication

To make REST calls using `curl` or `wget`, provide the username and password.

Curl Syntax

```
curl -k -u <username>:<password> https://<host>:<port>/rest/<command>...
```

Wget Syntax

```
wget --no-check-certificate --user <username> --password <password> https://<host>:<port>
/rest/<command>...
```

Command-Line Interface (CLI) Syntax

The Greenplum HD EE CLI commands are documented using the following conventions:

- `[Square brackets]` indicate an optional parameter
- `<Angle brackets>` indicate a value to enter

The following syntax example shows that the `volume mount` command requires the `-name` parameter, for which you must enter a list of volumes, and all other parameters are optional:

```
maprcli volume mount
    [ -cluster <cluster> ]
    -name <volume list>
    [ -path <path list> ]
```

For clarity, the syntax examples show each parameter on a separate line; in practical usage, the command and all parameters and options are typed on a single line. Example:

```
maprcli volume mount -name test-volume -path /test
```

Common Parameters

The following parameters are available for many commands in both the REST and command-line contexts.

Parameter	Description
cluster	The cluster on which to run the command. If this parameter is omitted, the command is run on the same cluster where it is issued. In multi-cluster contexts, you can use this parameter to specify a different cluster on which to run the command.
zkconnect	A ZooKeeper connect string, which specifies a list of the hosts running ZooKeeper, and the port to use on each, in the format: `'<host>[:<port>][,<host>[:<port>]...]'` Default: `'localhost:5181'` In most cases the ZooKeeper connect string can be omitted, but it is useful in certain cases when the CLDB is not running.

Common Options

The following options are available for most commands in the command-line context.

Option	Description
-noheader	When displaying tabular output from a command, omits the header row.
-long	Shows the entire value. This is useful when the command response contains complex information. When -long is omitted, complex information is displayed as an ellipsis (...).
-json	Displays command output in JSON format. When -json is omitted, the command output is displayed in tabular format.

Filters

Some Greenplum HD EE CLI commands use *filters*, which let you specify large numbers of nodes or volumes by matching specified values in specified fields rather than by typing each name explicitly.

Filters use the following format:

```
[<field><operator>"<value>"]<and|or>[<field><operator>"<value>"] ...
```

field	Field on which to filter. The field depends on the command with which the filter is used.

operator	An operator for that field: • == - Exact match • != - Does not match • > - Greater than • < - Less than

- >= - Greater than or equal to
- <= - Less than or equal to |

value	Value on which to filter. Wildcards (using *) are allowed for operators == and !=. There is a special value `all` that matches all values.

You can use the wildcard (*) for partial matches. For example, you can display all volumes whose owner is `root` and whose name begins with `test` as follows:

```
maprcli volume list -filter [n=="test*"]and[on=="root"]
```

Response

The commands return responses in JSON or in a tabular format. When you run commands from the command line, the response is returned in tabular format unless you specify JSON using the -json option; when you run commands through the REST interface, the response is returned in JSON.

Success

On a successful call, each command returns the error code zero (OK) and any data requested. When JSON output is specified, the data is returned as an array of records along with the status code and the total number of records. In the tabular format, the data is returned as a sequence of rows, each of which contains the fields in the record separated by tabs.

JSON	`{`
	` "status":"OK",`
	` "total":<number of records>,`
	` "data":[`
	`  {`
	`   <record>`
	`  }`
	`  ...`
	`         ]`
	`}`

Tabular	`status`
	`0`
	
	Or
	
	`<heading>   <heading>   <heading> ...`
	`<field>     <field>     <field> ...`
	`...`

Error

When an error occurs, the command returns the error code and descriptive message.

JSON	`{`
	`"status":"ERROR",`
	`"errors":[`
	`{`
	`"id":<error code>,`
	`"desc":"<command>: <error`
	`message>"`
	`}`
	`]`
	`}`

Tabular	`ERROR (<error code>) -  <command>: <error`
	`message>`

acl

The acl commands let you work with access control lists (ACLs):

- acl edit - modifies a specific user's access to a cluster or volume
- acl set - modifies the ACL for a cluster or volume
- acl show - displays the ACL associated with a cluster or volume

In order to use the `acl edit` command, you must have full control (`fc`) permission on the cluster or volume for which you are running the command. The following tables list the permission codes used by the `acl` commands.

Cluster Permission Codes

Code	Allowed Action	Includes
login	Log in to the Greenplum HD EE Control System, use the API and command-line interface, read access on cluster and volumes	cv
ss	Start/stop services	
cv	Create volumes	
a	Admin access	All permissions except fc
fc	Full control (administrative access and permission to change the cluster ACL)	a

Volume Permission Codes

Code	Allowed Action
dump	Dump the volume
restore	Mirror or restore the volume
m	Modify volume properties, create and delete snapshots
d	Delete a volume
fc	Full control (admin access and permission to change volume ACL)

acl edit

The `acl edit` command grants one or more specific volume or cluster permissions to a user. To use the `acl edit` command, you must have full control (`fc`) permissions on the volume or cluster for which you are running the command.

The permissions are specified as a comma-separated list of permission codes. See acl.

Syntax

CLI	`maprcli acl edit`
	`    [ -cluster <cluster name> ]`
	`    [ -group <group> ]`
	`    [ -name <name> ]`
	`    -type cluster|volume`
	`    [ -user <user> ]`

REST	`http[s]://<host:port>/rest/acl/edit?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
group	A group, and permissions to specify for the group. See acl. Format: `<group>:<action>[,<action>...]`
name	The object name.
type	The object type (`cluster` or `volume`).
user	A user, and allowed actions to specify for the user. See acl. Format: `<user>:<action>[,<action>...]`

Examples

Give the user jsmith dump, restore, and delete permissions for "test-volume":

CLI	`maprcli acl edit -type volume -name`
	`test-volume -user jsmith:dump,restore,d`

acl set

The `acl set` command specifies the entire ACL for a cluster or volume. Any previous permissions are overwritten by the new values, and any permissions omitted are removed. To use the `acl set` command, you must have full control (`fc`) permissions on the volume or cluster for which you are running the command.

The permissions are specified as a comma-separated list of permission codes. See acl.

Syntax

CLI	`maprcli acl set`
	`    [ -cluster <cluster name> ]`
	`    [ -group <group> ]`
	`    [ -name <name> ]`
	`    -type cluster|volume`
	`    [ -user <user> ]`

REST	`http[s]://<host:port>/rest/acl/edit?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
group	A group, and permissions to specify for the group. See acl. Format: `<group>:<action>[,<action>...]`
name	The object name.
type	The object type (`cluster` or `volume`).
user	A user, and allowed actions to specify for the user. See acl. Format: `<user>:<action>[,<action>...]`

Examples

Give the users jsmith and rjones specific permissions for "test-volume", and remove all permissions for all other users:

CLI	`maprcli acl set -type volume -name test-volume -user jsmith:dump,restore,m rjones:fc`

acl show

Displays the ACL associated with an object (cluster or a volume). An ACL contains the list of users who can perform specific actions.

Syntax

CLI

```
maprcli acl show
    [ -cluster <cluster> ]
    [ -group <group> ]
    [ -name <name> ]
    [ -output long|short|terse ]
    [ -perm ]
    -type cluster|volume
    [ -user <user> ]
```

REST	None

Parameters

Parameter	Description
cluster	The name of the cluster on which to run the command
group	The group for which to display permissions
name	The cluster or volume name

output	The output format: • long • short • terse
perm	When this option is specified, `acl show` displays the permissions available for the object type specified in the `type` parameter.
type	Cluster or volume.
user	The user for which to display permissions

Output

The actions that each user or group is allowed to perform on the cluster or the specified volume. For information about each allowed action, see acl.

```
Principal   Allowed actions
User root   [r, ss, cv, a, fc]
Group root  [r, ss, cv, a, fc]
All users   [r]
```

Examples

Show the ACL for "test-volume":

CLI	`maprcli acl show -type volume -name test-volume`

Show the permissions that can be set on a cluster:

CLI	`maprcli acl show -type cluster -perm`

alarm

The alarm commands perform functions related to system alarms:

- alarm clear - clears one or more alarms
- alarm clearall - clears all alarms
- alarm config load - displays the email addresses to which alarm notifications are to be sent
- alarm config save - saves changes to the email addresses to which alarm notifications are to be sent
- alarm list - displays alarms on the cluster
- alarm names - displays all alarm names
- alarm raise - raises a specified alarm

Alarm Notification Fields

The following fields specify the configuration of alarm notifications.

Field	Description
alarm	The named alarm.

individual	Specifies whether individual alarm notifications are sent to the default email address for the alarm type. • 0 - do not send notifications to the default email address for the alarm type • 1 - send notifications to the default email address for the alarm type
email	A custom email address for notifications about this alarm type. If specified, alarm notifications are sent to this email address, regardless of whether they are sent to the default email address

Alarm Types

See Troubleshooting Alarms.

Alarm History

To see a history of alarms that have been raised, look at the file `/opt/mapr/logs/cldb.log` on the master CLDB node. Example:

```
grep ALARM /opt/mapr/logs/cldb.log
```

alarm clear

Clears one or more alarms. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli alarm clear`
	`    -alarm <alarm>`
	`    [ -cluster <cluster> ]`
	`    [ -entity <host, volume, user, or group name> ]`

REST	`http[s]://<host>:<port>/rest/alarm/clear?<parameters>`

Parameters

Parameter	Description
alarm	The named alarm to clear. See Alarm Types.
cluster	The cluster on which to run the command.
entity	The entity on which to clear the alarm.

Examples

Clear a specific alarm:

CLI	`maprcli alarm clear -alarm NODE_ALARM_DEBUG_LOGGING`
REST	`https://r1n1.sj.us:8443/rest/alarm/clear?alarm=NODE_ALARM_DEBUG_LOGGING`

alarm clearall

Clears all alarms. Permissions required: `fc` or a

Syntax

CLI	`maprcli alarm clearall`
	`    [ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/alarm/clearall?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

Examples

Clear all alarms:

CLI	`maprcli alarm clearall`
REST	`https://r1n1.sj.us:8443/rest/alarm/clearall`

alarm config load

Displays the configuration of alarm notifications. Permissions required: `fc` or a

Syntax

CLI	`maprcli alarm config load`
	`    [ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/alarm/config/load`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

Output

A list of configuration values for alarm notifications.

Output Fields

See Alarm Notification Fields.

Sample output

```
alarm                                 individual  email
CLUSTER_ALARM_BLACKLIST_TTS           1
CLUSTER_ALARM_UPGRADE_IN_PROGRESS     1
CLUSTER_ALARM_UNASSIGNED_VIRTUAL_IPS  1
VOLUME_ALARM_SNAPSHOT_FAILURE         1
VOLUME_ALARM_MIRROR_FAILURE           1
VOLUME_ALARM_DATA_UNDER_REPLICATED    1
VOLUME_ALARM_DATA_UNAVAILABLE         1
VOLUME_ALARM_ADVISORY_QUOTA_EXCEEDED  1
VOLUME_ALARM_QUOTA_EXCEEDED           1
NODE_ALARM_CORE_PRESENT               1
NODE_ALARM_DEBUG_LOGGING              1
NODE_ALARM_DISK_FAILURE               1
NODE_ALARM_OPT_MAPR_FULL              1
NODE_ALARM_VERSION_MISMATCH           1
NODE_ALARM_TIME_SKEW                  1
NODE_ALARM_SERVICE_CLDB_DOWN          1
NODE_ALARM_SERVICE_FILESERVER_DOWN    1
NODE_ALARM_SERVICE_JT_DOWN            1
NODE_ALARM_SERVICE_TT_DOWN            1
NODE_ALARM_SERVICE_HBMASTER_DOWN      1
NODE_ALARM_SERVICE_HBREGION_DOWN      1
NODE_ALARM_SERVICE_NFS_DOWN           1
NODE_ALARM_SERVICE_WEBSERVER_DOWN     1
NODE_ALARM_SERVICE_HOSTSTATS_DOWN     1
NODE_ALARM_ROOT_PARTITION_FULL        1
AE_ALARM_AEADVISORY_QUOTA_EXCEEDED    1
AE_ALARM_AEQUOTA_EXCEEDED             1
```

Examples

Display the alarm notification configuration:

CLI	`maprcli alarm config load`
REST	`https://r1n1.sj.us:8443/rest/alarm/config/load`

alarm config save

Sets notification preferences for alarms. Permissions required: `fc` or a

Alarm notifications can be sent to the default email address and a specific email address for each named alarm. If `individual` is set to `1` for a specific alarm, then notifications for that alarm are sent to the default email address for the alarm type. If a custom email address is provided, notifications are sent there regardless of whether they are also sent to the default email address.

Syntax

CLI	`maprcli alarm config save`
	`[ -cluster <cluster> ]`
	`-values <values>`

REST	`http[s]://<host>:<port>/rest/alarm/config/save?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
values	A comma-separated list of configuration values for one or more alarms, in the following format: `<alarm>,<individual>,<email>` See Alarm Notification Fields.

Examples

Send alert emails for the AE_ALARM_AEQUOTA_EXCEEDED alarm to the default email address and a custom email address:

CLI	`maprcli alarm config save -values`
	`"AE_ALARM_AEQUOTA_EXCEEDED,1,test@example.com"`

REST	`https://r1n1.sj.us:8443/rest/alarm/config/save?values=AE_ALARM_AEQUOTA_EXCEEDED,1,test@example.com`

alarm list

Lists alarms in the system. Permissions required: `fc` or a

You can list all alarms, alarms by type (Cluster, Node or Volume), or alarms on a particular node or volume. To retrieve a count of all alarm types, pass `1` in the `summary` parameter. You can specify the alarms to return by filtering on type and entity. Use `start` and `limit` to retrieve only a specified window of data.

Syntax

CLI	
	```
	maprcli alarm list
	    [ -alarm <alarm ID> ]
	    [ -cluster <cluster> ]
	    [ -entity <host or volume> ]
	    [ -limit <limit> ]
	    [ -output (terse|verbose) ]
	    [ -start <offset> ]
	    [ -summary (0|1) ]
	    [ -type <alarm type> ]
	```

REST	`http[s]://<host>:<port>/rest/alarm/list?<parameters>`

Parameters

Parameter	Description
alarm	The alarm type to return. See Alarm Types.
cluster	The cluster on which to list alarms.
entity	The name of the cluster, node, volume, user, or group to check for alarms.
limit	The number of records to retrieve. Default: 2147483647
output	Whether the output should be terse or verbose.
start	The list offset at which to start.
summary	Specifies the type of data to return: • 1 = count by alarm type • 0 = List of alarms
type	The entity type: • `cluster` • `node` • `volume` • `ae`

Output

Information about one or more named alarms on the cluster, or for a specified node, volume, user, or group.

Output Fields

Field	Description
alarm state	State of the alarm: • `0` = Clear • `1` = Raised
description	A description of the condition that raised the alarm
entity	The name of the volume, node, user, or group.

alarm name	The name of the alarm.
alarm statechange time	The date and time the alarm was most recently raised.

Sample Output

```
alarm state  description
entity                             alarm name                         alarm
statechange time
1            Volume desired replication is 1, current replication is 0
mapr.qa-node173.qa.prv.local.logs    VOLUME_ALARM_DATA_UNDER_REPLICATED  1296707707872
1            Volume data unavailable
mapr.qa-node173.qa.prv.local.logs    VOLUME_ALARM_DATA_UNAVAILABLE       1296707707871
1            Volume desired replication is 1, current replication is 0
mapr.qa-node235.qa.prv.local.mapred  VOLUME_ALARM_DATA_UNDER_REPLICATED  1296708283355
1            Volume data unavailable
mapr.qa-node235.qa.prv.local.mapred  VOLUME_ALARM_DATA_UNAVAILABLE       1296708283099
1            Volume desired replication is 1, current replication is 0
mapr.qa-node175.qa.prv.local.logs    VOLUME_ALARM_DATA_UNDER_REPLICATED  1296706343256
```

Examples

List a summary of all alarms

CLI	`maprcli alarm list -summary 1`
REST	`https://r1n1.sj.us:8443/rest/alarm/list?summary=1`

List cluster alarms

CLI	`maprcli alarm list -type 0`
REST	`https://r1n1.sj.us:8443/rest/alarm/list?type=0`

alarm names

Displays a list of alarm names. Permissions required `fc` or a.

Syntax

CLI	`maprcli alarm names`
REST	`http[s]://<host>:<port>/rest/alarm/names`

Examples

Display all alarm names:

CLI	`maprcli alarm names`
REST	`https://r1n1.sj.us:8443/rest/alarm/names`

alarm raise

Raises a specified alarm or alarms. Permissions required `fc` or a.

Syntax

CLI	`maprcli alarm raise`
	`     -alarm <alarm>`
	`    [ -cluster <cluster> ]`
	`    [ -description <description> ]`
	`    [ -entity <cluster, entity, host, node, or volume> ]`

REST	`http[s]://<host>:<port>/rest/alarm/raise?<parameters>`

Parameters

Parameter	Description
alarm	The alarm type to raise. See Alarm Types.
cluster	The cluster on which to run the command.
description	A brief description.
entity	The entity on which to raise alarms.

Examples

Raise a specific alarm:

CLI	`maprcli alarm raise -alarm NODE_ALARM_DEBUG_LOGGING`
REST	`https://r1n1.sj.us:8443/rest/alarm/raise?alarm=NODE_ALARM_DEBUG_LOGGING`

config

The config commands let you work with configuration values for the Greenplum HD EE cluster:

- config load displays the values
- config save makes changes to the stored values

Configuration Fields

Field	Description
cldb.cluster.almost.full.percentage	The percentage at which the CLUSTER_ALARM_CLUSTER_ALMOST_FULL alarm is triggered.
cldb.default.volume.topology	The default topology for new volumes.
cldb.fs.mark.rereplicate.sec	The number of seconds a node can fail to heartbeat before it is considered dead. Once a node is considered dead, the CLDB re-replicates any data contained on the node.
cldb.min.fileservers	The minimum CLDB fileservers.
cldb.volume.default.replication	The default replication for the CLDB volumes.
mapr.domainname	The domain name Greenplum HD EE uses to get operating system users and groups (in domain mode).
mapr.entityquerysource	Sets MapR to get user information from LDAP (LDAP mode) or from the operating system of a domain (domain mode): • ldap • domain
mapr.fs.permissions.supergroup	The super group of the MapR-FS layer.
mapr.fs.permissions.superuser	The super user of the MapR-FS layer.
mapr.ldap.attribute.group	The LDAP server group attribute.
mapr.ldap.attribute.groupmembers	The LDAP server groupmembers attribute.
mapr.ldap.attribute.mail	The LDAP server mail attribute.
mapr.ldap.attribute.uid	The LDAP server uid attribute.
mapr.ldap.basedn	The LDAP server Base DN.
mapr.ldap.binddn	The LDAP server Bind DN.
mapr.ldap.port	The port Greenplum HD EE is to use on the LDAP server.
mapr.ldap.server	The LDAP server Greenplum HD EE uses to get users and groups (in LDAP mode).
mapr.ldap.sslrequired	Specifies whether the LDAP server requires SSL: • 0 == no • 1 == yes
mapr.quota.group.advisorydefault	The default group advisory quota; see Managing Quotas.
mapr.quota.group.default	The default group quota; see Managing Quotas.
mapr.quota.user.advisorydefault	The default user advisory quota; see Managing Quotas.
mapr.quota.user.default	The default user quota; see Managing Quotas.
mapr.smtp.port	The port Greenplum HD EE uses on the SMTP server (1-65535).
mapr.smtp.sender.email	The reply-to email address Greenplum HD EE uses when sending notifications.
mapr.smtp.sender.fullname	The full name Greenplum HD EE uses in the Sender field when sending notifications.
mapr.smtp.sender.password	The password Greenplum HD EE uses to log in to the SMTP server when sending notifications.
mapr.smtp.sender.username	The username Greenplum HD EE uses to log in to the SMTP server when sending notifications.
mapr.smtp.server	The SMTP server that Greenplum HD EE uses to send notifications.

mapr.smtp.sslrequired	Specifies whether SSL is required when sending email: • `0` == no • `1` == yes
mapr.webui.http.port	The port Greenplum HD EE uses for the Control System over HTTP (0-65535); if 0 is specified, disables HTTP access.
mapr.webui.https.certpath	The HTTPS certificate path.
mapr.webui.https.keypath	The HTTPS key path.
mapr.webui.https.port	The port Greenplum HD EE uses for the Control System over HTTPS (0-65535); if 0 is specified, disables HTTPS access.
mapr.webui.timeout	The number of seconds the Greenplum HD EE Control System allows to elapse before timing out.
mapreduce.cluster.permissions.supergroup	The super group of the MapReduce layer.
mapreduce.cluster.permissions.superuser	The super user of the MapReduce layer.

config load

Displays information about the cluster configuration. You can use the `keys` parameter to specify which information to display.

Syntax

<table>
<tr><td><b>CLI</b></td><td><pre>
maprcli config load
    [-cluster <cluster>]
    -keys <keys>
</pre></td></tr>
<tr><td><b>REST</b></td><td><pre>
http[s]://<host>:<port>/rest/config/load?<parameters>
</pre></td></tr>
</table>

Parameters

Parameter	Description
cluster	The cluster for which to display values.
keys	The fields for which to display values; see the Configuration Fields table

Output

Information about the cluster configuration. See the Configuration Fields table.

Sample Output

```
{
 "status":"OK",
 "total":1,
 "data":[
  {
   "mapr.webui.http.port":"8080",
   "mapr.fs.permissions.superuser":"root",
   "mapr.smtp.port":"25",
   "mapr.fs.permissions.supergroup":"supergroup"
  }
 ]
}
```

Examples

Display several keys:

CLI	`maprcli config load -keys mapr.webui.http.port,mapr.webui.https.port,mapr.webui.https.keystorepath,mapr.webui.https.keystorepassword,mapr.webui.https.keypassword,mapr.webui.timeout`
REST	`https://r1n1.sj.us:8443/rest/config/load?keys=mapr.webui.http.port,mapr.webui.https.port,mapr.webui.https.keystorepath,mapr.webui.https.keystorepassword,mapr.webui.https.keypassword,mapr.webui.timeout`

config save

Saves configuration information, specified as key/value pairs. Permissions required: `fc` or a.

See the Configuration Fields table.

Syntax

CLI	`maprcli config save`
	`    [ -cluster <cluster> ]`
	`    -values <values>`

REST	`http[s]://<host>:<port>/rest/config/save?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
values	A JSON object containing configuration fields; see the Configuration Fields table.

Examples

Configure MapR SMTP settings:

CLI	`maprcli config save -values {"mapr.smtp.provider":"gmail","mapr.smtp.server":"smtp.gmail.com","mapr.smtp.sslrequired":"true","mapr.smtp.port":"465","mapr.smtp.sender.fullname":"Ab Cd","mapr.smtp.sender.email":"xxx@gmail.com","mapr.smtp.sender.username":"xxx@gmail.com","mapr.smtp.sender.password":"abc"}`
REST	`https://r1n1.sj.us:8443/rest/config/save?values={"mapr.smtp.provider":"gmail","mapr.smtp.server":"smtp.gmail.com","mapr.smtp.sslrequired":"true","mapr.smtp.port":"465","mapr.smtp.sender.fullname":"Ab Cd","mapr.smtp.sender.email":"xxx@gmail.com","mapr.smtp.sender.username":"xxx@gmail.com","mapr.smtp.sender.password":"abc"}`

dashboard

The dashboard info command displays a summary of information about the cluster.

dashboard info

Displays a summary of information about the cluster. For best results, use the `-json` option when running `dashboard info` from the command line.

Syntax

CLI	`maprcli dashboard info`
	`    [ -cluster <cluster> ]`
	`    [ -zkconnect <ZooKeeper connect string> ]`

REST	`http[s]://<host>:<port>/rest/dashboard/info?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
zkconnect	ZooKeeper Connect String

Output

A summary of information about the services, volumes, mapreduce jobs, health, and utilization of the cluster.

Output Fields

Field	Description
services	The number of total and active services on the following nodes: • CLDB • File server • Job tracker • Task tracker • HB master • HB region server
volumes	The number and size (in GB) of volumes that are: • Available • Under-replicated • Unavailable
mapreduce	The following mapreduce information: • Queue time • Running jobs • Queued jobs • Running tasks • Blacklisted jobs
maintenance	The following information about system health: • Failed disk nodes • Cluster alarms • Node alarms • Versions
utilization	The following utilization information: • CPU: • Memory • Disk space

Sample Output

```
{
    "status":"OK",
    "total":1,
    "data":[
        {
            "volumes":{
                "available":{
                    "total":3,
                    "size":0
                },
                "underReplicated":{
                    "total":0,
                    "size":0
                },
                "unavailable":{
                    "total":1,
                    "size":0
                }
            },
            "utilization":{
```

```
            "cpu":{
                "total":0,
                "active":0
            },
            "memory":{
                "total":0,
                "active":0
            },
            "diskSpace":{
                "total":1,
                "active":0
            }
        },
        "maintenance":{
            "failedDiskNodes":0,
            "clusterAlarms":0,
            "nodeAlarms":0,
            "versions":1
        },
        "services":{
            "cldb":{
                "total":"1",
                "active":0
            },
            "fileserver":{
                "total":0,
                "active":0
            },
            "jobtracker":{
                "total":"1",
                "active":0
            },
            "nfs":{
                "total":"1",
                "active":0
            },
            "hbmaster":{
                "total":"1",
                "active":0
            },
            "hbregionserver":{
                "total":0,
                "active":0
            },
            "tasktracker":{
                "total":0,
                "active":0
            }
        }
    }
```

```
        ]
    }
```

Examples

Display dashboard information:

CLI	`maprcli dashboard info -json`
REST	`https://r1n1.sj.us:8443/rest/dashboard/info`

disk

The disk commands lets you work with disks:

- disk add adds a disk to a node
- disk list lists disks
- disk listall lists all disks
- disk remove removes a disk from a node

Disk Fields

The following table shows the fields displayed in the output of the disk list and disk listall commands. You can choose which fields (columns) to display and sort in ascending or descending order by any single field.

Field	Description
hn	Hostname of node which owns this disk/partition.
n	Name of the disk or partition.
st	Disk status: • 0 = Good • 1 = Bad disk
pst	Disk power status: • 0 = Active/idle (normal operation) • 1 = Standby (low power mode) • 2 = Sleeping (lowest power mode, drive is completely shut down)
mt	Disk mount status • 0 = unmounted • 1 = mounted
fs	File system type
mn	Model number
sn	Serial number
fw	Firmware version
ven	Vendor name
dst	Total disk space, in MB

dsu	Disk space used, in MB
dsa	Disk space available, in MB
err	Disk error message, in English. Note that this will **not** be translated. Only sent if st == 1.
ft	Disk failure time, Greenplum HD EE disks only. Only sent if st == 1.

disk add

Adds one or more disks to the specified node. Permissions required: `fc` or a

Syntax

CLI	`maprcli disk add`
	`    [ -cluster ]`
	`    -disks <disk names>`
	`    -host <host>`

REST	`http[s]://<host>:<port>/rest/disk/add?<parameters>`

Parameters

Parameter	**Description**
cluster	The cluster on which to add disks.
disks	A comma-separated list of disk names.Examples: • ["disk"] • ["disk","disk","disk"...]
host	The hostname or IP address of the machine on which to add the disk.

Output

Output Fields

Field	**Description**
ip	The IP address of the machine that owns the disk(s).
disk	The name of a disk or partition. Example "sca" or "sca/sca1"
all	The string `all`, meaning all unmounted disks for this node.

Examples

Add a disk:

CLI	`maprcli disk add -disks ["sda1"] -host 10.250.1.79`
REST	`https://r1n1.sj.us:8443/rest/disk/add?disks=["sda1"]`

disk list

Lists the disks on a node.

Syntax

CLI	`maprcli disk list`
	`-host <host>`
	`[ -output terse|verbose ]`
	`[ -system 1|0 ]`

REST	`http[s]://<host>:<port>/rest/disk/list?<parameters>`

Parameters

Parameter	Description
host	The node on which to list the disks.
output	Whether the output should be `terse` or `verbose`.
system	Show only operating system disks: • 0 - shows only MapR-FS disks • 1 - shows only operating system disks • Not specified - shows both MapR-FS and operating system disks

Output

Information about the specified disks. See the Disk Fields table.

Examples

List hosts on a host:

CLI	`maprcli disk list -host 10.10.100.22`

REST	`https://r1n1.sj.us:8443/rest/disk/list?host=10.10.100.22`

disk listall

Lists all disks

Syntax

CLI

```
maprcli disk listall
    [ -cluster <cluster> ]
    [ -columns <columns>]
    [ -filter <filter>]
    [ -limit <limit>]
    [ -output terse|verbose ]
    [ -start <offset>]
```

REST

```
http[s]://<host>:<port>/rest/disk/listall?<parameters>
```

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
columns	A comma-separated list of fields to return in the query. See the Disk Fields table.
filter	A filter specifying snapshots to preserve. See Filters for more information.
limit	The number of rows to return, beginning at start. Default: 0
output	Always the string `terse`.
start	The offset from the starting row according to sort. Default: 0

Output

Information about all disks. See the Disk Fields table.

Examples

List all disks:

CLI	`maprcli disk listall`

REST	`https://r1n1.sj.us:8443/rest/disk/listall`

disk remove

Removes a disk from MapR-FS. Permissions required: `fc` or `a`

The `disk remove` command does not remove a disk containing unreplicated data unless forced. To force disk removal, specify `force` with the value `1`.

Syntax

CLI	`maprcli disk remove`
	`    [ -cluster <cluster> ]`
	`    -disks <disk names>`
	`    [ -force 0|1 ]`
	`    -host <host>`

REST	`http[s]://<host>:<port>/rest/disk/remove?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
disks	A list of disks in the form: `["disk"]or["disk","disk","disk"...]or[]`
force	Whether to force • `0` (default) - do not remove the disk or disks if there is unreplicated data on the disk • `1` - remove the disk or disks regardless of data loss or other consequences
host	The hostname or ip address of the node from which to remove the disk.

Output

Output Fields

Field	Description
disk	The name of a disk or partition. Example: `sca` or `sca/sca1`
`all`	The string `all`, meaning all unmounted disks attached to the node.

disks	A comma-separated list of disks which have non-replicated volumes.<eg> "sca" or "sca/sca1,scb"</eg>

Examples

Remove a disk:

CLI	`maprcli disk remove -disks ["sda1"]`
REST	`https://r1n1.sj.us:8443/rest/disk/remove?disks=["sda1"]`

entity

The entity commands let you work with *entities* (users and groups):

- entity info shows information about a specified user or group
- entity list lists users and groups in the cluster
- entity modify edits information about a specified user or group

entity info

Displays information about an entity.

Syntax

CLI

```
maprcli entity info
    [ -cluster <cluster> ]
    -name <entity name>
    [ -output terse|verbose ]
    -type <type>
```

REST

```
http[s]://<host>:<port>/rest/entity/info?<parameters>
```

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
name	The entity name.
output	Whether to display terse or verbose output.
type	The entity type

Output

```
DiskUsage  EntityQuota  EntityType  EntityName  VolumeCount  EntityAdvisoryquota
EntityId
864415     0            0           root        208          0                    0
```

Output Fields

Field	Description
DiskUsage	Disk space used by the user or group
EntityQuota	The user or group quota
EntityType	The entity type
EntityName	The entity name
VolumeCount	The number of volumes associated with the user or group
EntityAdvisoryquota	The user or group advisory quota
EntityId	The ID of the user or group

Examples

Display information for the user 'root':

CLI	`maprcli entity info -type 0 -name root`
REST	`https://r1n1.sj.us:8443/rest/entity/info?type=0&name=root`

entity list

Syntax

CLI

```
maprcli entity list
    [ -alarmedentities true|false ]
    [ -cluster <cluster> ]
    [ -columns <columns> ]
    [ -filter <filter> ]
    [ -limit <rows> ]
    [ -output terse|verbose ]
    [ -start <start> ]
```

REST

```
http[s]://<host>:<port>/rest/entity/list?<parameters>
```

Parameters

Parameter	Description
alarmedentities	Specifies whether to list only entities that have exceeded a quota or advisory quota.
cluster	The cluster on which to run the command.
columns	A comma-separated list of fields to return in the query. See the Fields table below.
filter	A filter specifying entities to display. See Filters for more information.
limit	The number of rows to return, beginning at start. Default: 0
output	Specifies whether output should be `terse` or `verbose`.
start	The offset from the starting row according to sort. Default: 0

Output

Information about the users and groups.

Fields

Field	Description
EntityType	Entity type - `0` = User - `1` = Group
EntityName	User or Group name
EntityId	User or Group id
EntityQuota	Quota, in MB. `0` = no quota.
EntityAdvisoryquota	Advisory quota, in MB. `0` = no advisory quota.
VolumeCount	The number of volumes this entity owns.
DiskUsage	Disk space used for all entity's volumes, in MB.

Sample Output

```
DiskUsage  EntityQuota  EntityType  EntityName  VolumeCount  EntityAdvisoryquota
EntityId
5859220    0            0           root        209          0                    0
```

Examples

List all entities:

CLI	`maprcli entity list`
REST	`https://r1n1.sj.us:8443/rest/entity/list`

entity modify

Modifies a user or group quota or email address. Permissions required: `fc` or a

Syntax

CLI	`maprcli entity modify`
	`    [ -advisoryquota <advisory quota>`
	`    [ -cluster <cluster> ]`
	`    [ -email <email>]`
	`    -name <entityname>`
	`    [ -quota <quota> ]`
	`    -type <type>`

REST	`http[s]://<host>:<port>/rest/entity/modify?<parameters>`

Parameters

Parameter	Description
advisoryquota	The advisory quota.
cluster	The cluster on which to run the command.
email	Email address.
name	The entity name.
quota	The quota for the entity.
type	The entity type: • 0=user • 1-group

Examples

Modify the email address for the user 'root':

CLI	`maprcli entity modify -name root -type 0 -email test@example.com`
REST	`https://r1n1.sj.us:8443/rest/entity/modify?name=root&type=0&email=test@example.com`

license

The license commands let you work with Greenplum HD EE licenses:

- license add - adds a license
- license addcrl - adds a certificate revocation list (CRL)
- license apps - displays the features included in the current license
- license list - lists licenses on the cluster
- license listcrl - lists CRLs
- license remove - removes a license

- license showid - displays the cluster ID

license add

Adds a license. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli license add`
	`    [ -cluster <cluster> ]`
	`    [ -is_file true|false ]`
	`    -license <license>`

REST	`http[s]://<host>:<port>/rest/license/add?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
is_file	Specifies whether the `license` specifies a file. If `false`, the `license` parameter contains a long license string.
license	The license to add to the cluster. If `file` is true, `license` specifies the filename of a license file. Otherwise, `license` contains the license string itself.

license addcrl

Adds a certificate revocation list (CRL). Permissions required: `fc` or `a`

Syntax

CLI	`maprcli license addcrl`
	`    [ -cluster <cluster> ]`
	`    -crl <crl>`
	`    [ -is_file true|false ]`

REST	`http[s]://<host>:<port>/rest/license/addcrl?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

crl	The CRL to add to the cluster. If file is set, crl specifies the filename of a CRL file. Otherwise, crl contains the CRL string itself.
is_file	Specifies whether the license is contained in a file.

license apps

Displays the features authorized for the current license. Permissions required: `fc` or a

Syntax

CLI	`maprcli license apps`
	`[ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/license/apps?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

license list

Lists licenses on the cluster. Permissions required: `fc` or a

Syntax

CLI	`maprcli license list`
	`[ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/license/list?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

license listcrl

Lists certificate revocation lists (CRLs) on the cluster. Permissions required: `fc` or a

Syntax

CLI	`maprcli license listcrl`
	`[ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/license/listcrl?`
	`<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

license remove

Adds a license. Permissions required: `fc` or a

Syntax

CLI	`maprcli license remove`
	`[ -cluster <cluster> ]`
	`-license_id <license>`

REST	`http[s]://<host>:<port>/rest/license/remove?<`
	`parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
license_id	The license to remove.

license showid

Displays the cluster ID for use when creating a new license. Permissions required: `fc` or a

Syntax

CLI	`maprcli license showid`
	`[ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/license/showid?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

nagios

The nagios generate command generates a topology script for Nagios

nagios generate

Generates a Nagios Object Definition file that describes the cluster nodes and the services running on each.

Syntax

CLI	`maprcli nagios generate`
	`    [ -cluster <cluster> ]`

REST	`http[s]://<host>:<port>/rest/nagios/generate?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

Output

Sample Output

```
############# Commands #############

define command {
  command_name    check_fileserver_proc
  command_line    $USER1$/check_tcp -p 5660
}

define command {
   command_name    check_cldb_proc
   command_line    $USER1$/check_tcp -p 7222
}

define command {
   command_name    check_jobtracker_proc
   command_line    $USER1$/check_tcp -p 50030
}

define command {
   command_name    check_tasktracker_proc
   command_line    $USER1$/check_tcp -p 50060
}

define command {
   command_name    check_nfs_proc
   command_line    $USER1$/check_tcp -p 2049
}

define command {
   command_name    check_hbmaster_proc
   command_line    $USER1$/check_tcp -p 60000
}

define command {
   command_name    check_hbregionserver_proc
   command_line    $USER1$/check_tcp -p 60020
}

define command {
   command_name    check_webserver_proc
   command_line    $USER1$/check_tcp -p 8443
}

################# HOST: host1 ###############

define host {
  use linux-server
  host_name host1
  address 192.168.1.1
  check_command check-host-alive
}

################# HOST: host2 ###############

define host {
  use linux-server
  host_name host2
  address 192.168.1.2
  check_command check-host-alive
}
```

Examples

Generate a nagios configuration, specifying cluster name and ZooKeeper nodes:

CLI	`maprcli nagios generate -cluster cluster-1`
REST	`https://host1:8443/rest/nagios/generate?cluster=cluster-1`

Generate a nagios configuration and save to the file `nagios.conf`:

CLI	`maprcli nagios generate > nagios.conf`

nfsmgmt

The nfsmgmt refreshexports command refreshes the NFS exports on the specified host and/or port.

nfsmgmt refreshexports

Refreshes the NFS exports. Permissions required: `fc` or a

Syntax

CLI	`maprcli nfsmgmt refreshexports`
	`    [ -nfshost <host> ]`
	`    [ -nfsport <port> ]`

REST	`http[s]://<host>:<port>/rest/license/nfsmgmt/refreshexports?<parameters>`

Parameters

Parameter	Description
nfshost	The host on which to refresh NFS exports.
nfsport	The port to use.

node

The node commands let you work with nodes in the cluster:

- node heatmap
- node list
- node path
- node remove
- node services
- node topo

node heatmap

Displays a heatmap for the specified nodes.

Syntax

CLI	`maprcli node heatmap`
	`    [ -cluster <cluster> ]`
	`    [ -filter <filter> ]`
	`    [ -view <view> ]`

REST	`http[s]://<host>:<port>/rest/node/heatmap?<parameters>`

Parameters

Parameter	**Description**
cluster	The cluster on which to run the command.
filter	A filter specifying snapshots to preserve. See Filters for more information.
view	Name of the heatmap view to show: • `status` = Node status: ◦ `0` = Healthy ◦ `1` = Needs attention ◦ `2` = Degraded ◦ `3` = Maintenance ◦ `4` = Critical • `cpu` = CPU utilization, as a percent from 0-100. • `memory` = Memory utilization, as a percent from 0-100. • `diskspace` = MapR-FS disk space utilization, as a percent from 0-100. • `DISK_FAILURE` = Status of the DISK_FAILURE alarm. `0` if clear, `1` if raised. • `SERVICE_NOT_RUNNING` = Status of the SERVICE_NOT_RUNNING alarm. `0` if clear, `1` if raised. • `CONFIG_NOT_SYNCED` = Status of the CONFIG_NOT_SYNCED alarm. `0` if clear, `1` if raised.

Output

Description of the output.

```
{
 status:"OK",
 data: [{
   "{{rackTopology}}" : {
     "{{nodeName}}" : {{heatmapValue}},
     "{{nodeName}}" : {{heatmapValue}},
     "{{nodeName}}" : {{heatmapValue}},
     ...
    },
   "{{rackTopology}}" : {
     "{{nodeName}}" : {{heatmapValue}},
     "{{nodeName}}" : {{heatmapValue}},
     "{{nodeName}}" : {{heatmapValue}},
     ...
    },
    ...
 }]
}
```

Output Fields

Field	Description
rackTopology	The topology for a particular rack.
nodeName	The name of the node in question.
heatmapValue	The value of the metric specified in the view parameterfor this node, as an integer.

Examples

Display a heatmap for the default rack:

CLI	`maprcli node heatmap`
REST	`https://r1n1.sj.us:8443/rest/node/heatmap`

Display memory usage for the default rack:

CLI	`maprcli node heatmap -view memory`
REST	`https://r1n1.sj.us:8443/rest/node/heatmap?view=memory`

node list

Lists nodes in the cluster.

Syntax

CLI	`maprcli node list`
	`    [ -alarmednodes 1 ]`
	`    [ -cluster <cluster ]`
	`    [ -columns <columns>]`
	`    [ -filter <filter> ]`
	`    [ -limit <limit> ]`
	`    [ -nfsnodes 1 ]`
	`    [ -output terse|verbose ]`
	`    [ -start <offset> ]`
	`    [ -zkconnect <ZooKeeper Connect String> ]`

REST	`http[s]://<host>:<port>/rest/node/list?<parameters>`

Parameters

Parameter	Description
alarmednodes	If set to 1, displays only nodes running NFS. Cannot be used if nfsnodes is set.
cluster	The cluster on which to run the command.
columns	A comma-separated list of fields to return in the query. See the Fields table below.
filter	A filter specifying nodes on which to start or stop services. See Filters for more information.
limit	The number of rows to return, beginning at start. Default: 0
nfsnodes	If set to 1, displays only nodes with raised alarms. Cannot be used if alarmednodes is set.
output	Specifies whether the output should be terse or verbose.
start	The offset from the starting row according to sort. Default: 0
zkconnect	ZooKeeper Connect String

Output

Information about the nodes.See the Fields table above.

Sample Output

```
bytesSent  dreads  davail  TimeSkewAlarm  servicesHoststatsDownAlarm
ServiceHBMasterDownNotRunningAlarm  ServiceNFSDownNotRunningAlarm  ttmapUsed
DiskFailureAlarm  mused  id                   mtotal  cpus  utilization  rpcout
ttReduceSlots  ServiceFileserverDownNotRunningAlarm  ServiceCLDBDownNotRunningAlarm
dtotal  jt-heartbeat  ttReduceUsed  dwriteK  ServiceTTDownNotRunningAlarm
ServiceJTDownNotRunningAlarm  ttmapSlots  dused  uptime                        hostname
health  disks  faileddisks  fs-heartbeat  rpcin  ip           dreadK  dwrites
ServiceWebserverDownNotRunningAlarm  rpcs  LogLevelAlarm
ServiceHBRegionDownNotRunningAlarm  bytesReceived  service  topo(rack)           MapRfs
disks  ServiceMiscDownNotRunningAlarm  VersionMismatchAlarm
8300       0       269     0              0                           0
0                              75         0                 4058   6394230189818826805
7749    4     3            141     50             0                                     0
286     2             10           32       0                             0
100         16     Thu Jan 15 16:58:57 PST 1970  whatsup   0       1      0            0
51     10.250.1.48  0       2        0                                    0     0
0                                   8236                    /third/rack/whatsup  1
0                               0
```

Fields

Field	Description
bytesReceived	Bytes received by the node since the last CLDB heartbeat.
bytesSent	Bytes sent by the node since the last CLDB heartbeat.
corePresentAlarm	Cores Present Alarm (NODE_ALARM_CORE_PRESENT): • 0 = Clear • 1 = Raised
cpus	The total number of CPUs on the node.
davail	Disk space available on the node.
DiskFailureAlarm	Failed Disks alarm (DISK_FAILURE): • 0 = Clear • 1 = Raised
disks	Total number of disks on the node.
dreadK	Disk Kbytes read since the last heartbeat.
dreads	Disk read operations since the last heartbeat.
dtotal	Total disk space on the node.
dused	Disk space used on the node.
dwriteK	Disk Kbytes written since the last heartbeat.
dwrites	Disk write ops since the last heartbeat.
faileddisks	Number of failed MapR-FS disks on the node.
failedDisksAlarm	Disk Failure Alarm (NODE_ALARM_DISK_FAILURE) • 0 = Clear • 1 = Raised
fs-heartbeat	Time since the last heartbeat to the CLDB, in seconds.

health	Overall node health, calculated from various alarm states: • 0 = Healthy • 1 = Needs attention • 2 = Degraded • 3 = Maintenance • 4 = Critical
hostname	The host name.
id	The node ID.
ip	A list of IP addresses associated with the node.
jt-heartbeat	Time since the last heartbeat to the JobTracker, in seconds.
logLevelAlarm	Excessive Logging Alarm (NODE_ALARM_DEBUG_LOGGING): • 0 = Clear • 1 = Raised
MapRfs disks	
mtotal	Total memory, in GB.
mused	Memory used, in GB.
optMapRFullAlarm	Installation Directory Full Alarm (ODE_ALARM_OPT_MAPR_FULL): • 0 = Clear • 1 = Raised
rootPartitionFullAlarm	Root Partition Full Alarm (NODE_ALARM_ROOT_PARTITION_FULL): • 0 = Clear • 1 = Raised
rpcin	RPC bytes received since the last heartbeat.
rpcout	RPC bytes sent since the last heartbeat.
rpcs	Number of RPCs since the last heartbeat.
service	A comma-separated list of services running on the node: • cldb - CLDB • fileserver - MapR-FS • jobtracker - JobTracker • tasktracker - TaskTracker • hbmaster - HBase Master • hbregionserver - HBase RegionServer • nfs - NFS Gateway Example: "cldb,fileserver,nfs"
ServiceCLDBDownAlarm	CLDB Service Down Alarm (NODE_ALARM_SERVICE_CLDB_DOWN) • 0 = Clear • 1 = Raised
ServiceFileserverDownNotRunningAlarm	Fileserver Service Down Alarm (NODE_ALARM_SERVICE_FILESERVER_DOWN) • 0 = Clear • 1 = Raised

serviceHBMasterDownAlarm	HBase Master Service Down Alarm (NODE_ALARM_SERVICE_HBMASTER_DOWN) • 0 = Clear • 1 = Raised
serviceHBRegionDownAlarm	HBase Regionserver Service Down Alarm" (NODE_ALARM_SERVICE_HBREGION_DOWN) • 0 = Clear • 1 = Raised
servicesHoststatsDownAlarm	Hoststats Service Down Alarm (NODE_ALARM_SERVICE_HOSTSTATS_DOWN) • 0 = Clear • 1 = Raised
serviceJTDownAlarm	Jobtracker Service Down Alarm (NODE_ALARM_SERVICE_JT_DOWN) • 0 = Clear • 1 = Raised
ServiceMiscDownNotRunningAlarm	• 0 = Clear • 1 = Raised
serviceNFSDownAlarm	NFS Service Down Alarm (NODE_ALARM_SERVICE_NFS_DOWN): • 0 = Clear • 1 = Raised
serviceTTDownAlarm	Tasktracker Service Down Alarm (NODE_ALARM_SERVICE_TT_DOWN): • 0 = Clear • 1 = Raised
servicesWebserverDownAlarm	Webserver Service Down Alarm (NODE_ALARM_SERVICE_WEBSERVER_DOWN) • 0 = Clear • 1 = Raised
timeSkewAlarm	Time Skew alarm (NODE_ALARM_TIME_SKEW): • 0 = Clear • 1 = Raised
topo(rack)	The rack path.
ttmapSlots	TaskTracker map slots.
ttmapUsed	TaskTracker map slots used.
ttReduceSlots	TaskTracker reduce slots.
ttReduceUsed	TaskTracker reduce slots used.
uptime	The number of seconds the machine has been up since the last restart.
utilization	CPU use percentage since the last heartbeat.

versionMismatchAlarm	Software Version Mismatch Alarm (NODE_ALARM_VERSION_MISMATCH): • 0 = Clear • 1 = Raised

Examples

List all nodes:

CLI	`maprcli node list`
REST	`https://r1n1.sj.us:8443/rest/node/list`

node move

Moves one or more nodes to a different topology. Permissions required: `fc` or `a`

Syntax

CLI

```
maprcli node move
    [ -cluster <cluster> ]
     -serverids <server IDs>
     -topology <topology>
```

REST

```
http[s]://<host>:<port>/rest/node/move?<parameters>
```

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
serverids	The server IDs of the nodes to move.
topology	The new topology.

node path

Changes the path of the specified node or nodes. Permissions required: `fc` or `a`

Syntax

CLI	maprcli node path
	    [-cluster <cluster>]
	    [-filter <filter>]
	    [-nodes <node names>]
	    -path <path>
	    [-which switch|rack|both]
	    [-zkconnect <ZooKeeper Connect String>]

REST	http[s]://<host>:<port>/rest/node/path?<parameters>

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
filter	A filter specifying nodes on which to start or stop services. See Filters for more information.
nodes	A list of node names, separated by spaces.
path	The path to change.
which	Which path to change: switch, rack or both. Default: rack
zkconnect	ZooKeeper Connect String.

node remove

Remove one or more server nodes from the system. Permissions required: `fc` or a

Syntax

CLI	maprcli node remove
	    [-filter <filter>]
	    [-force true|false]
	    [-nodes <node names>]
	    [-zkconnect <ZooKeeper Connect String>]

REST	http[s]://<host>:<port>/rest/node/remove?<parameters>

Parameters

Parameter	Description
filter	A filter specifying nodes on which to start or stop services. See Filters for more information.
force	Forces the service stop operations. Default: false
nodes	A list of node names, separated by spaces.
zkconnect	ZooKeeper Connect String. Example: 'host:port,host:port,host:port,...'. default: localhost:5181

node services

Starts, stops, restarts, suspends, or resumes services on one or more server nodes. Permissions required: `ss`, `fc` or `a`

The same set of services applies to all specified nodes; to manipulate different groups of services differently, send multiple requests.

> ⚠ Note: the suspend and resume actions have not yet been implemented.

Syntax

<table>
<tr><td><b>CLI</b></td><td><pre>
maprcli node services
    [-action restart|resume|start|stop|suspend]
    [-cldb restart|resume|start|stop|suspend]
    [-cluster <cluster>]
    [-fileserver restart|resume|start|stop|suspend]
    [-filter <filter>]
    [-hbmaster restart|resume|start|stop|suspend]
    [-hbregionserver restart|resume|start|stop|suspend]
    [-jobtracker restart|resume|start|stop|suspend]
    [-name <service>]
    [-nfs restart|resume|start|stop|suspend]
    [-nodes <node names>]
    [-tasktracker restart|resume|start|stop|suspend]
    [-zkconnect <ZooKeeper Connect String>]
</pre></td></tr>
<tr><td><b>REST</b></td><td><pre>
http[s]://<host>:<port>/rest/node/services?<parameters>
</pre></td></tr>
</table>

Parameters

When used together, the `action` and `name` parameters specify an action to perform on a service. To start the JobTracker, for example, you can either specify `start` for the `action` and `jobtracker` for the `name`, or simply specify `start` on the `jobtracker`.

Parameter	Description
action	An action to perform on a service specified in the `name` parameter: restart, resume, start, stop, or suspend
cldb	Starts or stops the cldb service. Values: restart, resume, start, stop, or suspend
cluster	The cluster on which to run the command.
fileserver	Starts or stops the fileserver service. Values: restart, resume, start, stop, or suspend
filter	A filter specifying nodes on which to start or stop services. See Filters for more information.
hbmaster	Starts or stops the hbmaster service. Values: restart, resume, start, stop, or suspend
hbregionserver	Starts or stops the hbregionserver service. Values: restart, resume, start, stop, or suspend

jobtracker	Starts or stops the jobtracker service. Values: restart, resume, start, stop, or suspend
name	A service on which to perform an action specified by the `action` parameter.
nfs	Starts or stops the nfs service. Values: restart, resume, start, stop, or suspend
nodes	A list of node names, separated by spaces.
tasktracker	Starts or stops the tasktracker service. Values: restart, resume, start, stop, or suspend
zkconnect	ZooKeeper Connect String

node topo

Lists cluster topology information.

Lists internal nodes only (switches/racks/etc) and not leaf nodes (server nodes).

Syntax

CLI	`maprcli node topo`
	`[ -cluster <cluster> ]`
	`[ -path <path> ]`

REST	`http[s]://<host>:<port>/rest/node/topo?<parameters>`

Parameters

Parameter	**Description**
cluster	The cluster on which to run the command.
path	The path on which to list node topology.

Output

Node topology information.

Sample output

```
{
 status:"OK",
 total:recordCount,
 data: [
  {
   path:'path',
   status:[errorChildCount,OKChildCount,configChildCount],
  },
  ...additional structures above for each topology node...
 ]
}
```

Output Fields

Field	Description
path	The physical topology path to the node.
errorChildCount	The number of descendants of the node which have overall status 0.
OKChildCount	The number of descendants of the node which have overall status 1.
configChildCount	The number of descendants of the node which have overall status 2.

schedule

The schedule commands let you work with schedules:

- schedule create creates a schedule
- schedule list lists schedules
- schedule modify modifies the name or rules of a schedule by ID
- schedule remove removes a schedule by ID

A schedule is a JSON object that specifies a single or recurring time for volume snapshot creation or mirror syncing. For a schedule to be useful, it must be associated with at least one volume. See volume create and volume modify.

Schedule Fields

The schedule object contains the following fields:

Field	Value
id	The ID of the schedule.
name	The name of the schedule.
inuse	Indicates whether the schedule is associated with an action.
rules	An array of JSON objects specifying how often the scheduled action occurs. See Rule Fields below.

Rule Fields

The following table shows the fields to use when creating a rules object.

Field	Values
frequency	How often to perform the action: • `once` - Once • `yearly` - Yearly • `monthly` - Monthly • `weekly` - Weekly • `daily` - Daily • `hourly` - Hourly • `semihourly` - Every 30 minutes • `quarterhourly` - Every 15 minutes • `fiveminutes` - Every 5 minutes • `minute` - Every minute

retain	How long to retain the data resulting from the action. For example, if the schedule creates a snapshot, the retain field sets the snapshot's expiration. The retain field consists of an integer and one of the following units of time: • mi - minutes • h - hours • d - days • w - weeks • m - months • y - years
time	The time of day to perform the action, in 24-hour format: HH
date	The date on which to perform the action: • For single occurrences, specify month, day and year: MM/DD/YYYY • For yearly occurrences, specify the month and day: MM/DD • For monthly occurrences occurrences, specify the day: DD Daily and hourly occurrences do not require the date field.

Example

The following example JSON shows a schedule called "snapshot," with three rules.

```
{
            "id":8,
            "name":"snapshot",
            "inuse":0,
            "rules":[
                {
                    "frequency":"monthly",
                    "date":"8",
                    "time":14,
                    "retain":"1m"
                },
                {
                    "frequency":"weekly",
                    "date":"sat",
                    "time":14,
                    "retain":"2w"
                },
                {
                    "frequency":"hourly",
                    "retain":"1d"
                }
            ]
        }
```

schedule create

Creates a schedule. Permissions required: `fc` or `a`

A schedule can be associated with a volume to automate mirror syncing and snapshot creation. See volume create and volume modify.

Syntax

CLI	`maprcli schedule create`
	`    [ -cluster <cluster> ]`
	`    -schedule <JSON>`

REST	`http[s]://<host>:<port>/rest/schedule/create?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
schedule	A JSON object describing the schedule. See Schedule Objects for more information.

Examples

Scheduling a Single Occurrence

CLI	`maprcli schedule create -schedule '{"name":"Schedule-1","rules":[{"frequency":"once","retain":"1w","time":13,"date":"12/5/2010"}]}'`
REST	`https://r1n1.sj.us:8443/rest/schedule/create?schedule={"name":"Schedule-1","rules":[{"frequency":"once","retain":"1w","time":13,"date":"12/5/2010"}]}`

A Schedule with Several Rules

CLI	`maprcli schedule create -schedule '{"name":"Schedule-1","rules":[{"frequency":"weekly","date":"sun","time":7,"retain":"2w"},{"frequency":"daily","time":14,"retain":"1w"},{"frequency":"hourly","retain":"1w"},{"frequency":"yearly","date":"11/5","time":14,"retain":"1w"}]}'`
REST	`https://r1n1.sj.us:8443/rest/schedule/create?schedule={"name":"Schedule-1","rules":[{"frequency":"weekly","date":"sun","time":7,"retain":"2w"},{"frequency":"daily","time":14,"retain":"1w"},{"frequency":"hourly","retain":"1w"},{"frequency":"yearly","date":"11/5","time":14,"retain":"1w"}]}`

schedule list

Lists the schedules on the cluster.

Syntax

CLI	`maprcli schedule list`
	`    [ -cluster <cluster> ]`
	`    [ -output terse|verbose ]`

REST	`http[s]://<host>:<port>/rest/schedule/list?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
output	Specifies whether the output should be terse or verbose.

Output

A list of all schedules on the cluster. See Schedule Objects for more information.

Examples

List schedules:

CLI	`maprcli schedule list`
REST	`https://r1n1.sj.us:8443/rest/schedule/list`

schedule modify

Modifies an existing schedule, specified by ID. Permissions required: `fc` or `a`

To find a schedule's ID:

1. Use the schedule list command to list the schedules.
2. Select the schedule to modify
3. Pass the selected schedule's ID in the -id parameter to the `schedule modify` command.

Syntax

CLI	`maprcli schedule modify`
	`    [ -cluster <cluster> ]`
	`    -id <schedule ID>`
	`    [ -name <schedule name ]`
	`    [ -rules <JSON>]`

REST	`http[s]://<host>:<port>/rest/schedule/modify?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
id	The ID of the schedule to modify.
name	The new name of the schedule.
rules	A JSON object describing the rules for the schedule. If specified, replaces the entire existing rules object in the schedule. For information about the fields to use in the JSON object, see Rule Fields.

Examples

Modify a schedule

CLI	`maprcli schedule modify -id 0 -name Newname -rules '[{"frequency":"weekly","date":"sun","time":7,"retain":"2w"},{"frequency":"daily","time":14,"retain":"1w"}]'`
REST	`https://r1n1.sj.us:8443/rest/schedule/modify?id=0&name=Newname&rules=[{"frequency":"weekly","date":"sun","time":7,"retain":"2w"},{"frequency":"daily","time":14,"retain":"1w"}]`

schedule remove

Removes a schedule.

A schedule can only be removed if it is not associated with any volumes. See volume modify.

Syntax

CLI	`maprcli schedule remove`
	`    [ -cluster <cluster> ]`
	`    -id <schedule ID>`

REST	`http[s]://<host>:<port>/rest/schedule/remove?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
id	The ID of the schedule to remove.

Examples

Remove schedule with ID 0:

CLI	`maprcli schedule remove -id 0`
REST	`https://r1n1.sj.us:8443/rest/schedule/remove?id=0`

service list

Lists all services on the specified node, along with the state and log path for each service.

Syntax

CLI	`maprcli service list`
	`-node <node name>`

REST	`http[s]://<host>:<port>/rest/service/list?<parameters>`

Parameters

Parameter	Description
node	The node on which to list the services

Output

Information about services on the specified node. For each service, the status is reported numerically:

- 0 - NOT_CONFIGURED: the package for the service is not installed and/or the service is not configured (`configure.sh` has not run)
- 2 - RUNNING: the service is installed, has been started by the warden, and is currently executing
- 3 - STOPPED: the service is installed and `configure.sh` has run, but the service is currently not executing

setloglevel

The setloglevel commands set the log level on individual services:

- setloglevel cldb - Sets the log level for the CLDB.
- setloglevel hbmaster - Sets the log level for the HB Master.
- setloglevel hbregionserver - Sets the log level for the HBase RegionServer.
- setloglevel jobtracker - Sets the log level for the JobTracker.
- setloglevel fileserver - Sets the log level for the FileServer.
- setloglevel nfs - Sets the log level for the NFS.
- setloglevel tasktracker - Sets the log level for the TaskTracker.

setloglevel cldb

Sets the log level on the CLDB service. Permissions required: `fc` or a

Syntax

CLI	`maprcli setloglevel cldb -classname <class> -loglevel DEBUG|ERROR|FATAL|INFO|TRACE|WARN -node <node>`
REST	`http[s]://<host>:<port>/rest/setloglevel/cldb?<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.
loglevel	The log level to set: • DEBUG • ERROR • FATAL • INFO • TRACE • WARN
node	The node on which to set the log level.

setloglevel fileserver

Sets the log level on the FileServer service. Permissions required: `fc` or a

Syntax

CLI	`maprcli setloglevel fileserver -classname <class> -loglevel DEBUG|ERROR|FATAL|INFO|TRACE|WARN -node <node>`
REST	`http[s]://<host>:<port>/rest/setloglevel/fileserver?<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.

loglevel	The log level to set: • DEBUG • ERROR • FATAL • INFO • TRACE • WARN
node	The node on which to set the log level.

setloglevel hbmaster

Sets the log level on the HB Master service. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli setloglevel hbmaster`
	`    -classname <class>`
	`    -loglevel`
	`DEBUG|ERROR|FATAL|INFO|TRACE|WARN`
	`    -node <node>`

REST	`http[s]://<host>:<port>/rest/setloglevel/hbmaster?<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.
loglevel	The log level to set: • DEBUG • ERROR • FATAL • INFO • TRACE • WARN
node	The node on which to set the log level.

setloglevel hbregionserver

Sets the log level on the HB RegionServer service. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli setloglevel hbregionserver`
	`    -classname <class>`
	`    -loglevel`
	`DEBUG|ERROR|FATAL|INFO|TRACE|WARN`
	`    -node <node>`

REST	`http[s]://<host>:<port>/rest/setloglevel/hbregionserver?<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.
loglevel	The log level to set: - DEBUG - ERROR - FATAL - INFO - TRACE - WARN
node	The node on which to set the log level.

setloglevel jobtracker

Sets the log level on the JobTracker service. Permissions required: `fc` or a

Syntax

CLI	`maprcli setloglevel jobtracker`
	`    -classname <class>`
	`    -loglevel`
	`DEBUG|ERROR|FATAL|INFO|TRACE|WARN`
	`    -node <node>`

REST	`http[s]://<host>:<port>/rest/setloglevel/jobtracker?<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.

loglevel	The log level to set: • DEBUG • ERROR • FATAL • INFO • TRACE • WARN
node	The node on which to set the log level.

setloglevel nfs

Sets the log level on the NFS service. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli setloglevel nfs`
	`    -classname <class>`
	`    -loglevel`
	`DEBUG|ERROR|FATAL|INFO|TRACE|WARN`
	`    -node <node>`

REST	`http[s]://<host>:<port>/rest/setloglevel/nfs?`
	`<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.
loglevel	The log level to set: • DEBUG • ERROR • FATAL • INFO • TRACE • WARN
node	The node on which to set the log level.

setloglevel tasktracker

Sets the log level on the TaskTracker service. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli setloglevel tasktracker`
	`    -classname <class>`
	`    -loglevel DEBUG|ERROR|FATAL|INFO|TRACE|WARN`
	`    -node <node>`

REST	`http[s]://<host>:<port>/rest/setloglevel/tasktracker?<parameters>`

Parameters

Parameter	Description
classname	The name of the class for which to set the log level.
loglevel	The log level to set: • DEBUG • ERROR • FATAL • INFO • TRACE • WARN
node	The node on which to set the log level.

trace

The trace commands let you view and modify the trace buffer, and the trace levels for the system modules. The valid trace levels are:

- DEBUG
- INFO
- ERROR
- WARN
- FATAL

The following pages provide information about the trace commands:

- trace dump
- trace info
- trace print
- trace reset
- trace resize
- trace setlevel
- trace setmode

trace dump

Dumps the contents of the trace buffer into the MapR-FS log.

Syntax

CLI	`maprcli trace dump`
	`    [ -host <host> ]`
	`    [ -port <port> ]`

REST	None.

Parameters

Parameter	Description
host	The IP address of the node from which to dump the trace buffer. Default: `localhost`
port	The port to use when dumping the trace buffer. Default: `5660`

Examples

Dump the trace buffer to the MapR-FS log:

CLI	`maprcli trace dump`

trace info

Displays the trace level of each module.

Syntax

CLI	
	```
	maprcli trace info
	    [ -host <host> ]
	    [ -port <port> ]
	```

REST	None.

Parameters

Parameter	Description
host	The IP address of the node on which to display the trace level of each module. Default: `localhost`
port	The port to use. Default: `5660`

Output

A list of all modules and their trace levels.

Sample Output

```
RPC Client Initialize
**Trace is in DEFAULT mode.
**Allowed Trace Levels are:
FATAL
ERROR
WARN
INFO
DEBUG
**Trace buffer size: 2097152
**Modules and levels:
Global : INFO
RPC : ERROR
MessageQueue : ERROR
CacheMgr : INFO
IOMgr : INFO
Transaction : ERROR
Log : INFO
Cleaner : ERROR
Allocator : ERROR
BTreeMgr : ERROR
BTree : ERROR
BTreeDelete : ERROR
BTreeOwnership : INFO
MapServerFile : ERROR
MapServerDir : INFO
Container : INFO
Snapshot : INFO
Util : ERROR
Replication : INFO
PunchHole : ERROR
KvStore : ERROR
Truncate : ERROR
Orphanage : INFO
FileServer : INFO
Defer : ERROR
ServerCommand : INFO
NFSD : INFO
Cidcache : ERROR
Client : ERROR
Fidcache : ERROR
Fidmap : ERROR
Inode : ERROR
JniCommon : ERROR
Shmem : ERROR
Table : ERROR
Fctest : ERROR
DONE
```

Examples

Display trace info:

CLI	`maprcli trace info`

trace print

Manually dumps the trace buffer to stdout.

Syntax

CLI	`maprcli trace print`
	`    [ -host <host> ]`
	`    [ -port <port> ]`
	`    -size <size>`

REST	None.

Parameters

Parameter	Description
host	The IP address of the node from which to dump the trace buffer to stdout. Default: `localhost`
port	The port to use. Default: `5660`
size	The number of kilobytes of the trace buffer to print. Maximum: 64

Output

The most recent `<size>` bytes of the trace buffer.

```
-----------------------------------------------------
2010-10-04 13:59:31,0000 Program: mfs on Host: fakehost IP: 0.0.0.0, Port: 0, PID: 0

-----------------------------------------------------
DONE
```

Examples

Display the trace buffer:

CLI	`maprcli trace print`

trace reset

Resets the in-memory trace buffer.

Syntax

CLI	`maprcli trace reset`
	`    [ -host <host> ]`
	`    [ -port <port> ]`

REST	None.

Parameters

Parameter	Description
host	The IP address of the node on which to reset the trace parameters. Default: `localhost`
port	The port to use. Default: `5660`

Examples

Reset trace parameters:

CLI	`maprcli trace reset`

trace resize

Resizes the trace buffer.

Syntax

CLI	`maprcli trace resize`
	`    [ -host <host> ]`
	`    [ -port <port> ]`
	`    -size <size>`

REST	None.

Parameters

Parameter	Description
host	The IP address of the node on which to resize the trace buffer. Default: `localhost`
port	The port to use. Default: `5660`
size	The size of the trace buffer, in kilobytes. Default: `2097152` Minimum: `1`

Examples

Resize the trace buffer to 1000

CLI	`maprcli trace resize -size 1000`

trace setlevel

Sets the trace level on one or more modules.

Syntax

CLI	`maprcli trace setlevel`
	`    [ -host <host> ]`
	`    -level <trace level>`
	`    -module <module name>`
	`    [ -port <port> ]`

REST	None.

Parameters

Parameter	**Description**
host	The node on which to set the trace level. Default: `localhost`
module	The module on which to set the trace level. If set to `all`, sets the trace level on all modules.
level	The new trace level. If set to `default`, sets the trace level to its default.
port	The port to use. Default: `5660`

Examples

Set the trace level of the log module to INFO:

CLI	`maprcli trace setlevel -module log -level info`

Set the trace levels of all modules to their defaults:

CLI	`maprcli trace setlevel -module all -level default`

trace setmode

Sets the trace mode. There are two modes:

- Default
- Continuous

In default mode, all trace messages are saved in a memory buffer. If there is an error, the buffer is dumped to stdout. In continuous mode, every allowed trace message is dumped to stdout in real time.

Syntax

CLI	`maprcli trace setmode`
	`    [ -host <host> ]`
	`    -mode default|continuous`
	`    [ -port <port> ]`

REST	None.

Parameters

Parameter	Description
host	The IP address of the host on which to set the trace mode
mode	The trace mode.
port	The port to use.

Examples

Set the trace mode to continuous:

CLI	`maprcli trace setmode -mode continuous`

urls

The urls command displays the status page URL for the specified service.

Syntax

CLI

```
maprcli urls
    [ -cluster <cluster> ]
    -name <service name>
    [ -zkconnect <zookeeper connect string> ]
```

REST

```
http[s]://<host>:<port>/rest/urls/<name>
```

Parameters

Parameter	Description
cluster	The name of the cluster on which to save the configuration.
name	The name of the service for which to get the status page: • cldb • jobtracker • tasktracker
zkconnect	ZooKeeper Connect String

Examples

Display the URL of the status page for the CLDB service:

CLI	`maprcli urls -name cldb`
REST	`https://r1n1.sj.us:8443/rest/maprcli/urls/cldb`

virtualip

The virtualip commands let you work with virtual IP addresses for NFS nodes:

- virtualip add adds a ranges of virtual IP addresses
- virtualip list lists virtual IP addresses
- virtualip remove removes a range of virtual IP addresses

Virtual IP Fields

Field	Description
macaddress	The MAC address of the virtual IP.
netmask	The netmask of the virtual IP.
virtualipend	The virtual IP range end.

virtualip add

Adds a virtual IP address. Permissions required: `fc` or a

Syntax

CLI	`maprcli virtualip add`
	`    [ -cluster <cluster> ]`
	`    [ -macaddress <MAC address>`
	`    -netmask <netmask>`
	`    -virtualip <virtualip>`
	`    [ -virtualipend <virtual IP range end> ]`

REST	`http[s]://<host>:<port>/rest/virtualip/add?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
macaddress	The MAC address of the virtual IP.
netmask	The netmask of the virtual IP.
virtualip	The virtual IP.
virtualipend	The virtual IP range end.

virtualip edit

Edits a virtual IP (VIP) range. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli virtualip edit`
	`    [ -cluster <cluster> ]`
	`    [ -macs <mac address(es)> ]`
	`    -netmask <netmask>`
	`    -virtualip <virtualip>`
	`    [ -virtualipend <range end> ]`

REST	`http[s]://<host>:<port>/rest/virtualip/edit?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
macs	The MAC address or addresses to associate with the VIP or VIP range.
netmask	The netmask for the VIP or VIP range.
virtualip	The start of the VIP range, or the VIP if only one VIP is used.
virtualipend	The end of the VIP range if more than one VIP is used.

virtualip list

Lists the virtual IP addresses in the cluster.

Syntax

CLI	`maprcli virtualip list`
	`    [ -cluster <cluster> ]`
	`    [ -columns <columns> ]`
	`    [ -filter <filter> ]`
	`    [ -limit <limit> ]`
	`    [ -nfsmacs <NFS macs> ]`
	`    [ -output <output> ]`
	`    [ -range <range> ]`
	`    [ -start <start> ]`

REST	`http[s]://<host>:<port>/rest/virtualip/list?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.

columns	The columns to display.
filter	A filter specifying VIPs to list. See Filters for more information.
limit	The number of records to return.
nfsmacs	The MAC addresses of servers running NFS.
output	Whether the output should be `terse` or `verbose`.
range	The VIP range.
start	The index of the first record to return.

virtualip remove

Removes a virtual IP (VIP) or a VIP range. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli virtualip remove`
	`    [ -cluster <cluster> ]`
	`    -virtualip <virtual IP>`
	`    [ -virtualipend <Virtual IP Range End> ]`

REST	`http[s]://<host>:<port>/rest/virtualip/remove`
	`?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
virtualip	The virtual IP or the start of the VIP range to remove.
virtualipend	The end of the VIP range to remove.

volume

The volume commands let you work with volumes, snapshots and mirrors:

- volume create creates a volume
- volume dump create creates a volume dump
- volume dump restore restores a volume from a volume dump
- volume info displays information about a volume
- volume link create creates a symbolic link
- volume link remove removes a symbolic link
- volume list lists volumes in the cluster
- volume mirror push pushes a volume's changes to its local mirrors
- volume mirror start starts mirroring a volume
- volume mirror stop stops mirroring a volume
- volume modify modifies a volume
- volume mount mounts a volume
- volume move moves a volume
- volume remove removes a volume
- volume rename renames a volume
- volume snapshot create creates a volume snapshot
- volume snapshot list lists volume snapshots
- volume snapshot preserve prevents a volume snapshot from expiring

- volume snapshot remove removes a volume snapshot
- volume unmount unmounts a volume

volume create

Creates a volume. Permissions required: `cv`, `fc`, or `a`

Syntax

CLI

```
maprcli volume create
[ -advisoryquota <advisory quota> ]
[ -ae <accounting entity> ]
[ -aetype <accounting entity type> ]
[ -canBackup <list of users and groups> ]
[ -canDeleteAcl <list of users and groups> ]
[ -canDeleteVolume <list of users and groups> ]
[ -canEditconfig <list of users and groups> ]
[ -canMirror <list of users and groups> ]
[ -canMount <list of users and groups> ]
[ -canSnapshot <list of users and groups> ]
[ -canViewConfig <list of users and groups> ]
[ -cluster <cluster> ]
[ -groupperm <group:allowMask list> ]
[ -localvolumehost <localvolumehost> ]
[ -localvolumeport <localvolumeport> ]
[ -minreplication <minimum replication factor> ]
[ -mount 0|1 ]
-name <volume name>
[ -path <mount path> ]
[ -quota <quota> ]
[ -readonly <read-only status> ]
[ -replication <replication factor> ]
[ -rereplicationtimeoutsec <seconds> ]
[ -rootdirperms <root directory permissions> ]
[ -schedule <ID> ]
[ -source <source> ]
[ -topology <topology> ]
[ -type 0|1 ]
[ -userperm <user:allowMask list> ]
```

REST

```
http[s]://<host>:<port>/rest/volume/create?<parameters>
```

Parameters

Parameter	Description
advisoryquota	The advisory quota for the volume as `integer` plus `unit. Example: quota=500G; Units:` B, K, M, G, T, P
ae	The accounting entity that owns the volume.
aetype	The type of accounting entity: - `0=user` - `1=group`
canBackup	The list of users and groups who can back up the volume.
canDeleteAcl	The list of users and groups who can delete the volume access control list (ACL)

canDeleteVolume	The list of users and groups who can delete the volume.
canEditconfig	The list of users and groups who can edit the volume properties.
canMirror	The list of users and groups who can mirror the volume.
canMount	The list of users and groups who can mount the volume.
canSnapshot	The list of users and groups who can create a snapshot of the volume.
canViewConfig	The list of users and groups who can view the volume properties.
cluster	The cluster on which to create the volume.
groupperm	List of permissions in the format `group:allowMask`
localvolumehost	The local volume host.
localvolumeport	The local volume port. Default: 5660
minreplication	The minimum replication level. Default: 0
mount	Specifies whether the volume is mounted at creation time.
name	The name of the volume to create.
path	The path at which to mount the volume.
quota	The quota for the volume as `integer` plus `unit. Example: quota=500G; Units:` `B`, `K`, `M`, `G`, `T`, `P`
readonly	Specifies whether the volume is read-only: • `0` = read/write • `1` = read-only
replication	The desired replication level. Default: 0
rereplicationtimeoutsec	The re-replication timeout, in seconds.
rootdirperms	Permissions on the volume root directory.
schedule	The ID of a schedule. If a schedule ID is provided, then the volume will automatically create snapshots (normal volume) or sync with its source volume (mirror volume) on the specified schedule.
source	For mirror volumes, the source volume to mirror, in the format `<source volume>@<cluster>` (Required when creating a mirror volume).
topology	The rack path to the volume.
type	The type of volume to create: • `0` - standard volume • `1` - mirror volume
userperm	List of permissions in the format `user:allowMask`.

Examples

Create the volume "test-volume" mounted at "/test/test-volume":

CLI	`maprcli volume create -name test-volume -path /test/test-volume`
REST	`https://r1n1.sj.us:8443/rest/volume/create?name=test-volume&path=/test/test-volume`

Create Volume with a Quota and an Advisory Quota

This example creates a volume with the following parameters:

- advisoryquota: 100M
- name: volumename
- path: /volumepath
- quota: 500M
- replication: 3
- schedule: 2
- topology: /East Coast
- type: 0

CLI	`maprcli volume create -name volumename -path /volumepath -advisoryquota 100M`
	`-quota 500M -replication 3 -schedule 2`
	`-topology "/East Coast" -type 0`

REST	`https://r1n1.sj.us:8443/rest/volume/create?advisoryquota=100M&name=volumename&path=/volumepath&quota=500M&replication=3&schedule=2&topology=/East%20Coast&type=0`

Create the mirror volume "test-volume.mirror" from source volume "test-volume" and mount at "/test/test-volume-mirror":

CLI	`maprcli volume create -name test-volume.mirror`
	`-source test-volume@src-cluster-name`
	`-path /test/test-volume-mirror`

REST	`https://r1n1.sj.us:8443/rest/volume/create?name=test-volume.mirror&sourcetest-volume@src-cluster-name&path=/test/test-volume-mirror`

volume dump create

The volume dump create command creates a volume *dump file* containing data from a volume for distribution or restoration.

You can use volume dump create to create two types of files:

- *full* dump files containing all data in a volume
- *incremental* dump files that contain changes to a volume between two points in time

A full dump file is useful for restoring a volume from scratch. An incremental dump file contains the changes necessary to take an existing (or restored) volume from one point in time to another. Along with the dump file, a full or incremental dump operation can produce a *state* file (specified by the -e parameter) that contains a table of the version number of every container in the volume at the time the dump file was created. This represents the *end point* of the dump file, which is used as the *start point* of the next incremental dump. The main difference between creating a full dump and creating an incremental dump is whether the -s parameter is specified; if -s is not specified, the volume create command includes all volume data and creates a full dump file. If you create a full dump followed by a series of incremental dumps, the result is a sequence of dump files and their accompanying state files:

> *dumpfile1 statefile1*

> *dumpfile2 statefile2*
>
> *dumpfile3 statefile3*
>
> ...

You can restore the volume from scratch, using the volume dump restore command with each dump file.

Syntax

CLI	`maprcli volume dump create`
	`    [ -cluster <cluster> ]`
	`    -dumpfile <dump file>`
	`    [-e <end state file> ]`
	`    -name volumename`
	`    [-o ]`
	`    [-s <start state file>`
	`]{anchor:cli-syntax-end}`

REST	None.

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
dumpfile	The name of the dump file (ignored if -o is used).
e	The name of the state file to create for the end point of the dump.
name	A volume name.
o	This option dumps the volume to stdout instead of to a file.
s	The start point for an incremental dump.

Examples

Create a full dump:

CLI	`maprcli volume create -e statefile1 -dumpfile fulldump1 -name volume`

Create an incremental dump:

CLI	`maprcli volume dump -s statefile1 -e statefile2 -name volume -dumpfile incrdump1`

volume dump restore

The `volume dump restore` command restores or updates a volume from a dump file. Permissions required: `dump`, `fc`, or a

There are two ways to use `volume dump restore`:

- With a full dump file, `volume dump restore` recreates a volume from scratch from volume data stored in the dump

file.
- With an incremental dump file, `volume dump restore` updates a volume using incremental changes stored in the dump file.

The volume that results from a `volume dump restore` operation is a mirror volume whose source is the volume from which the dump was created. After the operation, this volume can perform mirroring from the source volume.

When you are updating a volume from an incremental dump file, you must specify an existing volume and an incremental dump file. To restore from a sequence of previous dump files would involve first restoring from the volume's full dump file, then applying each subsequent incremental dump file.

A restored volume may contain mount points that represent volumes that were mounted under the original source volume from which the dump was created. In the restored volume, these mount points have no meaning and do not provide access to any volumes that were mounted under the source volume. If the source volume still exists, then the mount points in the restored volume will work if the restored volume is associated with the source volume as a mirror.

Syntax

CLI	`maprcli volume dump restore`
	`    [ -cluster <cluster> ]`
	`    -dumpfile dumpfilename`
	`    [ -i ]`
	`    [ -n ]`
	`    -name <volume name>`

REST	None.

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
dumpfile	The name of the dumpfile (ignored if `-i` is used).
i	This option reads the dump file from `stdin`.
n	This option creates a new volume if it doesn't exist.
name	A volume name, in the form `volumename`

Examples

Restore a volume from a full dump file:

CLI	`maprcli volume dump restore -name volume`
	`-dumpfile fulldump1`

Apply an incremental dump file to a volume:

CLI	`maprcli volume dump restore -name volume`
	`-dumpfile incrdump1`

volume fixmountpath

Corrects the mount path of a volume. Permissions required: `fc` or `a`

The CLDB maintains information about the mount path of every volume. If a directory in a volume's path is renamed (by a `hadoop fs` command, for example) the information in the CLDB will be out of date. The `volume fixmountpath` command does a reverse path walk from the volume and corrects the mount path information in the CLDB.

Syntax

CLI	`maprcli volume fixmountpath`
	`-name <name>`

REST	`http[s]://<host>:<port>/rest/volume/fixmountpath?<parameters>`

Parameters

Parameter	Description
name	The volume name.

Examples

Fix the mount path of volume v1:

CLI	`maprcli volume fixmountpath -name v1`
REST	`https://r1n1.sj.us:8443/rest/volume/fixmountpath?name=v1`

volume info

Displays information about the specified volume.

Syntax

CLI	`maprcli volume info`
	`    [ -cluster <cluster> ]`
	`    [ -name <volume name> ]`
	`    [ -output terse|verbose ]`
	`    [ -path <path> ]`

REST	`http[s]://<host>:<port>/rest/volume/info?<parameters>`

Parameters

You must specify either name or path.

Parameter	Description

cluster	The cluster on which to run the command.
name	The volume for which to retrieve information.
output	Whether the output should be terse or verbose.
path	The mount path of the volume for which to retrieve information.

volume link create

Creates a link to a volume. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli volume link create`
	`    -path <path>`
	`    -type <type>`
	`    -volume <volume>`

REST	`http[s]://<host>:<port>/rest/volume/link/remove?<parameters>`

Parameters

Parameter	Description
path	The path parameter specifies the link path and other information, using the following syntax: `/link/[maprfs::][volume::]<volume type>::<volume name>` • `link` - the link path • `maprfs` - a keyword to indicate a special MapR-FS link • `volume` - a keyword to indicate a link to a volume • `volume type` - writeable or mirror • `volume name` - the name of the volume Example: `/abc/maprfs::mirror::abc`
type	The volume type: `writeable` or `mirror`.
volume	The volume name.

Examples

Create a link to v1 at the path v1. mirror:

CLI	`maprcli volume link create -volume v1 -type mirror -path /v1.mirror`

REST	`https://r1n1.sj.us:8443/rest/volume/link/create?path=/v1.mirror&type=mirror&volume=v1`

volume link remove

Removes the specified symbolic link. Permissions required: `fc` or `a`

Syntax

CLI	`maprcli volume link remove`
	`    -path <path>`

REST	`http[s]://<host>:<port>/rest/volume/link/remove?<parameters>`

Parameters

Parameter	Description

path	The symbolic link to remove. The path parameter specifies the link path and other information about the symbolic link, using the following syntax:
	`/link/[maprfs::][volume::]<volume type>::<volume name>`
	• `link` - the symbolic link path
	*`maprfs` - a keyword to indicate a special MapR-FS link
	• `volume` - a keyword to indicate a link to a volume
	• `volume type` - `writeable` or `mirror`
	• `volume name` - the name of the volume
	Example:
	`/abc/maprfs::mirror::abc`

Examples

Remove the link /abc:

CLI	`maprcli volume link remove -path /abc/maprfs::mirror::abc`
REST	`https://r1n1.sj.us:8443/rest/volume/link/remove?path=/abc/maprfs::mirror::abc`

volume list

Lists information about volumes specified by name, path, or filter.

Syntax

CLI	`maprcli volume list`
	`    [ -alarmedvolumes 1 ]`
	`    [ -cluster <cluster> ]`
	`    [ -columns <columns> ]`
	`    [ -filter <filter> ]`
	`    [ -limit <limit> ]`
	`    [ -nodes <nodes> ]`
	`    [ -output terse | verbose ]`
	`    [ -start <offset> ]`
	`{anchor:cli-syntax-end}`

REST	`http[s]://<host>:<port>/rest/volume/list?<parameters>`

Parameters

Parameter	Description
alarmedvolumes	Specifies whether to list alarmed volumes only.
cluster	The cluster on which to run the command.
columns	A comma-separated list of fields to return in the query. See the Fields table below.
filter	A filter specifying volumes to list. See Filters for more information.
limit	The number of rows to return, beginning at start. Default: 0
nodes	A list of nodes. If specified, `volume list` only lists volumes on the specified nodes.
output	Specifies whether the output should be `terse` or `verbose`.
start	The offset from the starting row according to sort. Default: 0

Field	Description
volumeid	Unique volume ID.
volumetype	Volume type: • `0` = normal volume • `1` = mirror volume
volumename	Unique volume name.
mountdir	Unique volume path (may be null if the volume is unmounted).
mounted	Volume mount status: • `0` = unmounted • `1` = mounted
rackpath	Rack path.

creator	Username of the volume creator.
aename	Accountable entity name.
aetype	Accountable entity type: - `0=user` - `1=group`
uacl	Users ACL (comma-separated list of user names.
gacl	Group ACL (comma-separated list of group names).
quota	Quota, in MB; `0` = no quota.
advisoryquota	Advisory quota, in MB; `0` = no advisory quota.
used	Disk space used, in MB, not including snapshots.
snapshotused	Disk space used for all snapshots, in MB.
totalused	Total space used for volume and snapshots, in MB.
readonly	Read-only status: - `0` = read/write - `1` = read-only
numreplicas	Desired replication factor (number of replications).
minreplicas	Minimum replication factor (number of replications)
actualreplication	The actual current replication factor by percentage of the volume, as a zero-based array of integers from 0 to 100. For each position in the array, the value is the percentage of the volume that is replicated index number of times. Example: `arf=[5,10,85]` means that 5% is not replicated, 10% is replicated once, 85% is replicated twice.
schedulename	The name of the schedule associated with the volume.
scheduleid	The ID of the schedule associated with the volume.
mirrorSrcVolumeId	Source volume ID (mirror volumes only).
mirrorSrcVolume	Source volume name (mirror volumes only).
mirrorSrcCluster	The cluster where the source volume resides (mirror volumes only).
lastSuccessfulMirrorTime	Last successful Mirror Time, milliseconds since 1970 (mirror volumes only).
mirrorstatus	Mirror Status (mirror volumes only: - `0` = success - `1` = running - `2` = error
mirror-percent-complete	Percent completion of last/current mirror (mirror volumes only).
snapshotcount	Snapshot count .
SnapshotFailureAlarm	Status of SNAPSHOT_FAILURE alarm: - `0` = Clear - `1` = Raised

AdvisoryQuotaExceededAlarm	Status of VOLUME_ALARM_ADVISORY_QUOTA_EXCEEDED alarm: • 0 = Clear • 1 = Raised
QuotaExceededAlarm	Status of VOLUME_QUOTA_EXCEEDED alarm: • 0 = Clear • 1 = Raised
MirrorFailureAlarm	Status of MIRROR_FAILURE alarm: • 0 = Clear • 1 = Raised
DataUnderReplicatedAlarm	Status of DATA_UNDER_REPLICATED alarm: • 0 = Clear • 1 = Raised
DataUnavailableAlarm	Status of DATA_UNAVAILABLE alarm: • 0 = Clear • 1 = Raised

Output

Information about the specified volumes.

Output Fields

See the Fields table above.

volume mirror push

Pushes the changes in a volume to all of its mirror volumes in the same cluster, and waits for each mirroring operation to complete. Use this command when you need to push recent changes.

Syntax

<table>
<tr><td><b>CLI</b></td><td><pre>maprcli volume mirror push
    [-cluster <cluster>]
    -name <volume name>
    [-verbose true|false]</pre></td></tr>
<tr><td><b>REST</b></td><td>None.</td></tr>
</table>

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
name	The volume to push.

verbose	Specifies whether the command output should be verbose. Default: true

Output

Sample Output

```
Starting mirroring of volume mirror1
Mirroring complete for volume mirror1
Successfully completed mirror push to all local mirrors of volume volume1
```

Examples

Push changes from the volume "volume1" to its local mirror volumes:

CLI	`maprcli volume mirror push -name volume1 -cluster mycluster`

volume mirror start

Starts mirroring on the specified volume from its source volume. License required: M5 Permissions required: `fc` or a

When a mirror is started, the mirror volume is synchronized from a hidden internal snapshot so that the mirroring process is not affected by any concurrent changes to the source volume. The `volume mirror start` command does not wait for mirror completion, but returns immediately. The changes to the mirror volume occur atomically at the end of the mirroring process; deltas transmitted from the source volume do not appear until mirroring is complete.

To provide rollback capability for the mirror volume, the mirroring process creates a snapshot of the mirror volume before starting the mirror, with the following naming format: `<volume>.mirrorsnap.<date>.<time>`.

Normally, the mirroring operation transfers only deltas from the last successful mirror. Under certain conditions (mirroring a volume repaired by `fsck`, for example), the source and mirror volumes can become out of sync. In such cases, it is impossible to transfer deltas, because the state is not the same for both volumes. Use the `-full` option to force the mirroring operation to transfer all data to bring the volumes back in sync.

Syntax

CLI	`maprcli volume mirror start`
	`    [ -cluster <cluster> ]`
	`    [ -full true|false ]`
	`    -name <volume name>`

REST	`http[s]://<host>:<port>/rest/volume/mirror/start?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
full	Specifies whether to perform a full copy of all data. If false, only the deltas are copied.
name	The volume for which to stop the mirror.

Output

Sample Output

```
messages
Started mirror operation for volumes 'test-mirror'
```

Examples

Start mirroring the mirror volume "test-mirror":

CLI	`maprcli volume mirror start -name test-mirror`

volume mirror stop

Stops mirroring on the specified volume. License required: M5 Permissions required: `fc` or `a`

The `volume mirror stop` command lets you stop mirroring (for example, during a network outage). You can use the `volume mirror start` command to resume mirroring.

Syntax

CLI	`maprcli volume mirror stop`
	`    [ -cluster <cluster> ]`
	`    -name <volume name>`

REST	`http[s]://<host>:<port>/rest/volume/mirror/stop?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
name	The volume for which to stop the mirror.

Output

Sample Output

```
messages
Stopped mirror operation for volumes 'test-mirror'
```

Examples

Stop mirroring the mirror volume "test-mirror":

CLI	`maprcli volume mirror stop -name test-mirror`

volume modify

Modifies an existing volume. Permissions required: `m`, `fc`, or `a`

An error occurs if the name or path refers to a non-existent volume, or cannot be resolved.

Syntax

<table>
<tr><td><b>CLI</b></td><td><pre>
maprcli volume modify
    [-advisoryquota <advisory quota>]
    [-ae <accounting entity>]
    [-aetype <aetype>]
    [-canBackup <list of users and groups>]
    [-canDeleteAcl <list of users and groups>
]
    [-canDeleteVolume <list of users and
groups>]
    [-canEditconfig <list of users and
groups>]
    [-canMirror <list of users and groups>]
    [-canMount <list of users and groups>]
    [-canSnapshot <list of users and groups>
]
    [-canViewConfig <list of users and
groups>]
    [-cluster <cluster>]
    [-groupperm <list of group:allowMask>]
    [-minreplication <minimum replication>]
    -name <volume name>
    [-quota <quota>]
    [-readonly <readonly>]
    [-replication <replication>]
    [-schedule <schedule ID>]
    [-source <source>]
    [-topology <topology>]
    [-userperm <list of user:allowMask>]
</pre></td></tr>
<tr><td><b>REST</b></td><td>http[s]://<host>:<port>/rest/volume/modify?<parameters&gt;</td></tr>
</table>

Parameters

Parameter	Description
advisoryquota	The advisory quota for the volume as `integer` plus `unit. Example: quota=500G; Units:` B, K, M, G, T, P
ae	The accounting entity that owns the volume.
aetype	The type of accounting entity: • `0=user` • `1=group`
canBackup	The list of users and groups who can back up the volume.
canDeleteAcl	The list of users and groups who can delete the volume access control list (ACL).
canDeleteVolume	The list of users and groups who can delete the volume.
canEditconfig	The list of users and groups who can edit the volume properties.

canMirror	The list of users and groups who can mirror the volume.
canMount	The list of users and groups who can mount the volume.
canSnapshot	The list of users and groups who can create a snapshot of the volume.
canViewConfig	The list of users and groups who can view the volume properties.
cluster	The cluster on which to run the command.
groupperm	A list of permissions in the format `group:allowMask`
minreplication	The minimum replication level. Default: 0
name	The name of the volume to modify.
quota	The quota for the volume as `integer` plus `unit. Example: quota=500G; Units:` B, K, M, G, T, P
readonly	Specifies whether the volume is read-only. • `0` = read/write • `1` = read-only
replication	The desired replication level. Default: 0
schedule	A schedule ID. If a schedule ID is provided, then the volume will automatically create snapshots (normal volume) or sync with its source volume (mirror volume) on the specified schedule.
source	(Mirror volumes only) The source volume from which a mirror volume receives updates, specified in the format `<volume>@<cluster>`.
topology	The rack path to the volume.
userperm	List of permissions in the format `user:allowMask`.

Examples

Change the source volume of the mirror "test-mirror":

CLI	`maprcli volume modify -name test-mirror -source volume-2@my-cluster`
REST	`https://r1n1.sj.us:8443/rest/volume/modify?name=test-mirror&source=volume-2@my-cluster`

volume mount

Mounts one or more specified volumes. Permissions required: `mnt`, `fc`, or `a`

Syntax

CLI	`maprcli volume mount`
	`    [ -cluster <cluster> ]`
	`    -name <volume list>`
	`    [ -path <path list> ]`

REST	`http[s]://<host>:<port>/rest/volume/mount?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
name	The name of the volume to mount.
path	The path at which to mount the volume.

Examples

Mount the volume "test-volume" at the path "/test":

CLI	`maprcli volume mount -name test-volume -path /test`
REST	`https://r1n1.sj.us:8443/rest/volume/mount?name=test-volume&path=/test`

volume move

Moves the specified volume or mirror to a different topology. Permissions required: `m`, `fc`, or `a`

Syntax

CLI	`maprcli volume move`
	`    [ -cluster <cluster> ]`
	`    -name <volume name>`
	`    -topology <path>`

REST	`http[s]://<host>:<port>/rest/volume/move?<parameters>`

Parameters

Parameter	Description

cluster	The cluster on which to run the command.
name	The volume name.
topology	The new rack path to the volume.

volume remove

Removes the specified volume or mirror. Permissions required: `d`, `fc`, or `a`

Syntax

CLI	`maprcli volume remove`
	`    [ -cluster <cluster> ]`
	`    [ -force ]`
	`    -name <volume name>`

REST	`http[s]://<host>:<port>/rest/volume/remove?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
force	Forces the removal of the volume, even if it would otherwise be prevented.
name	The volume name.

volume rename

Renames the specified volume or mirror. Permissions required: `m`, `fc`, or `a`

Syntax

CLI	`maprcli volume rename`
	`    [ -cluster <cluster> ]`
	`    -name <volume name>`
	`    -newname <new volume name>`

REST	`http[s]://<host>:<port>/rest/volume/rename?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
name	The volume name.
newname	The new volume name.

volume snapshot create

Creates a snapshot of the specified volume, using the specified snapshot name. License required: M5 Permissions required: `snap`, `fc`, or `a`

Syntax

CLI	`maprcli volume snapshot create`
	`    [ -cluster <cluster> ]`
	`    -snapshotname <snapshot>`
	`    -volume <volume>`

REST	`http[s]://<host>:<port>/rest/volume/snapshot/create?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
snapshotname	The name of the snapshot to create.
volume	The volume for which to create a snapshot.

Examples

Create a snapshot called "test-snapshot" for volume "test-volume":

CLI	`maprcli volume snapshot create -snapshotname test-snapshot -volume test-volume`
REST	`https://r1n1.sj.us:8443/rest/volume/snapshot/create?volume=test-volume&snapshotname=test-snapshot`

volume snapshot list

Displays info about a set of snapshots. You can specify the snapshots by volumes or paths, or by specifying a filter to select volumes with certain characteristics.

Syntax

<table>
<tr><td><b>CLI</b></td><td><pre>
maprcli volume snapshot list
    [-cluster <cluster>]
    [-columns <fields>]
    (-filter <filter> | -path <volume path
list> | -volume <volume list>)
    [-limit <rows>]
    [-output (terse\|verbose)]
    [-start <offset>]
</pre></td></tr>
<tr><td><b>REST</b></td><td><pre>
http[s]://<host>:<port>/rest/volume/snapshot/
list?<parameters>
</pre></td></tr>
</table>

Parameters

Specify exactly one of the following parameters: `volume`, `path`, or `filter`.

Parameter	Description
cluster	The cluster on which to run the command.
columns	A comma-separated list of fields to return in the query. See the Fields table below. Default: none
filter	A filter specifying snapshots to preserve. See Filters for more information.
limit	The number of rows to return, beginning at start. Default: 0
output	Specifies whether the output should be `terse` or `verbose`. Default: `verbose`
path	A comma-separated list of paths for which to preserve snapshots.
start	The offset from the starting row according to sort. Default: 0
volume	A comma-separated list of volumes for which to preserve snapshots.

Fields

The following table lists the fields used in the sort and columns parameters, and returned as output.

Field	Description
snapshotid	Unique snapshot ID.
snapshotname	Snapshot name.
volumeid	ID of the volume associated with the snapshot.
volumename	Name of the volume associated with the snapshot.
volumepath	Path to the volume associated with the snapshot.
ownername	Owner (user or group) associated with the volume.

ownertype	Owner type for the owner of the volume: • `0=user` • `1=group`
dsu	Disk space used for the snapshot, in MB.
creationtime	Snapshot creation time, milliseconds since 1970
expirytime	Snapshot expiration time, milliseconds since 1970; `0` = never expires.

Output

The specified columns about the specified snapshots.

Sample Output

```
creationtime   ownername  snapshotid  snapshotname
expirytime     diskspaceused  volumeid  volumename                 ownertype  volumepath
1296788400768  dummy      363         ATS-Run-2011-01-31-160018.2011-02-03.19-00-00
1296792000001  1063191        362       ATS-Run-2011-01-31-160018  1          /dummy
1296789308786  dummy      364         ATS-Run-2011-01-31-160018.2011-02-03.19-15-02
1296792902057  753010         362       ATS-Run-2011-01-31-160018  1          /dummy
1296790200677  dummy      365         ATS-Run-2011-01-31-160018.2011-02-03.19-30-00
1296793800001  0              362       ATS-Run-2011-01-31-160018  1          /dummy
dummy      1          14        test-volume-2  /dummy      102
test-volume-2.2010-11-07.10:00:00  0              1289152800001  1289239200001
```

Output Fields

See the Fields table above.

Examples

List all snapshots:

CLI	`maprcli volume snapshot list`
REST	`https://r1n1.sj.us:8443/rest/volume/snapshot/list`

volume snapshot preserve

Preserves one or more snapshots from expiration. Specify the snapshots by volumes, paths, filter, or IDs. License required: M5 Permissions required: `snap`, `fc`, or `a`

Syntax

CLI	`maprcli volume snapshot preserve`
	`    [ -cluster <cluster> ]`
	`    ( -filter <filter> | -path <volume path list> | -snapshots <snapshot list> | -volume <volume list> )`

REST	`http[s]://<host>:<port>/rest/volume/snapshot/preserve?<parameters>`

Parameters

Specify exactly one of the following parameters: volume, path, filter, or snapshots.

Parameter	Description
cluster	The cluster on which to run the command.
filter	A filter specifying snapshots to preserve. See Filters for more information.
path	A comma-separated list of paths for which to preserve snapshots.
snapshots	A comma-separated list of snapshot IDs to preserve.
volume	A comma-separated list of volumes for which to preserve snapshots.

Examples

Preserve two snapshots by ID:

First, use `volume snapshot list` to get the IDs of the snapshots you wish to preserve. Example:

```
# maprcli volume snapshot list
creationtime   ownername  snapshotid  snapshotname
expirytime     diskspaceused  volumeid  volumename                 ownertype  volumepath
1296788400768  dummy      363         ATS-Run-2011-01-31-160018.2011-02-03.19-00-00
1296792000001  1063191        362       ATS-Run-2011-01-31-160018  1          /dummy
1296789308786  dummy      364         ATS-Run-2011-01-31-160018.2011-02-03.19-15-02
1296792902057  753010         362       ATS-Run-2011-01-31-160018  1          /dummy
1296790200677  dummy      365         ATS-Run-2011-01-31-160018.2011-02-03.19-30-00
1296793800001  0              362       ATS-Run-2011-01-31-160018  1          /dummy
dummy      1          14        test-volume-2  /dummy      102
test-volume-2.2010-11-07.10:00:00  0              1289152800001  1289239200001
```

Use the IDs in the `volume snapshot preserve` command. For example, to preserve the first two snapshots in the above list, run the commands as follows:

CLI	`maprcli volume snapshot preserve -snapshots 363,364`
REST	`https://r1n1.sj.us:8443/rest/volume/snapshot/preserve?snapshots=363,364`

volume snapshot remove

Removes one or more snapshots. License required: M5 Permissions required: `snap`, `fc`, or `a`

Syntax

CLI	`maprcli volume snapshot remove`
	`    [ -cluster <cluster> ]`
	`    ( -snapshotname <snapshot name> |`
	`-snapshots <snapshots> | -volume <volume name>`
	`)`

REST	`http[s]://<host>:<port>/rest/volume/snapshot/`
	`remove?<parameters>`

Parameters

Specify exactly one of the following parameters: snapshotname, snapshots, or volume.

Parameter	Description
cluster	The cluster on which to run the command.
snapshotname	The name of the snapshot to remove.
snapshots	A comma-separated list of IDs of snapshots to remove.
volume	The name of the volume from which to remove the snapshot.

Examples

Remove the snapshot "test-snapshot":

CLI	`maprcli volume snapshot remove -snapshotname`
	`test-snapshot`

REST	`https://10.250.1.79:8443/api/volume/snapshot/`
	`remove?snapshotname=test-snapshot`

Remove two snapshots by ID:

First, use `volume snapshot list` to get the IDs of the snapshots you wish to remove. Example:

```
# maprcli volume snapshot list
creationtime   ownername  snapshotid  snapshotname
expirytime     diskspaceused  volumeid  volumename                 ownertype  volumepath
1296788400768  dummy      363         ATS-Run-2011-01-31-160018.2011-02-03.19-00-00
1296792000001  1063191        362       ATS-Run-2011-01-31-160018  1          /dummy
1296789308786  dummy      364         ATS-Run-2011-01-31-160018.2011-02-03.19-15-02
1296792902057  753010         362       ATS-Run-2011-01-31-160018  1          /dummy
1296790200677  dummy      365         ATS-Run-2011-01-31-160018.2011-02-03.19-30-00
1296793800001  0              362       ATS-Run-2011-01-31-160018  1          /dummy
dummy      1          14        test-volume-2  /dummy      102
test-volume-2.2010-11-07.10:00:00  0              1289152800001  1289239200001
```

Use the IDs in the `volume snapshot remove` command. For example, to remove the first two snapshots in the above list, run the commands as follows:

CLI	`maprcli volume snapshot remove -snapshots 363,364`
REST	`https://r1n1.sj.us:8443/rest/volume/snapshot/remove?snapshots=363,364`

volume unmount

Unmounts one or more mounted volumes. Permissions required: `mnt`, `fc`, or `a`

Syntax

CLI	`maprcli volume unmount`
	`    [ -cluster <cluster> ]`
	`    [ -force 1 ]`
	`    -name <volume name>`

REST	`http[s]://<host>:<port>/rest/volume/unmount?<parameters>`

Parameters

Parameter	Description
cluster	The cluster on which to run the command.
force	Specifies whether to force the volume to unmount.
name	The name of the volume to unmount.

Examples

Unmount the volume "test-volume":

CLI	`maprcli volume unmount -name test-volume`
REST	`https://r1n1.sj.us:8443/rest/volume/unmount?name=test-volume`

Glossary

Term	Definition
.dfs_attributes	A special file in every directory, for controlling the compression and chunk size used for the directory and its subdirectories.

.rw	A special mount point in a top-level volume (or read-only mirror) that points to the writable original copy of the volume.
access control list	A list of permissions attached to an object. An access control list (ACL) specifies users or system processes that can perform specific actions on an object.
accounting entity	A clearly defined economics unit that is accounted for separately.
ACL	See access control list .
advisory quota	An advisory disk capacity limit that can be set for a volume, user, or group. When disk usage exceeds the advisory quota, an alert is sent.
AE	See accounting entity .
bitmask	A binary number in which each bit controls a single toggle.
CLDB	See container location database .
container	The unit of sharded storage in a Greenplum HD EE cluster.
container location database	A service, running on one or more Greenplum HD EE nodes, that maintains the locations of services, containers, and other cluster information.
dump file	A file containing data from a volume for distribution or restoration. There are two types of dump files: *full* dump files containing all data in a volume, and *incremental* dump files that contain changes to a volume between two points in time.
entity	A user or group. Users and groups can represent accounting entities .
full dump file	See dump file .
Hbase	A distributed storage system, designed to scale to a very large size, for managing massive amounts of structured data.
incremental dump file	See dump file .
JobTracker	The process responsible for submitting and tracking MapReduce jobs. The JobTracker sends individual tasks to TaskTrackers on nodes in the cluster.
Mapr-FS	The NFS-mountable, distributed, high-performance Greenplum HD EE data storage system.
mirror	A read-only physical copy of a volume.
Network File System	A protocol that allows a user on a client computer to access files over a network as though they were stored locally.
NFS	See Network File System .
node	An individual server (physical or virtual machine) in a cluster.
quota	A disk capacity limit that can be set for a volume, user, or group. When disk usage exceeds the quota, no more data can be written.
recovery point objective	The maximum allowable data loss as a point in time. If the recovery point objective is 2 hours, then the maximum allowable amount of data loss that is acceptable is 2 hours of work.
recovery time objective	The maximum alllowable time to recovery after data loss. If the recovery time objective is 5 hours, then it must be possible to restore data up to the recovery point objective within 5 hours. See also recovery point objective
replication factor	The number of copies of the data, not including the original.
RPO	See recovery point objective .
RTO	See recovery time objective .

schedule	A group of rules that specify recurring points in time at which certain actions are determined to occur.
snapshot	A read-only logical image of a volume at a specific point in time.
storage pool	A unit of storage made up of one or more disks. By default, Greenplum HD EE storage pools contain two or three disks. For high-volume reads and writes, you can create larger storage pools when initially formatting storage during cluster creation.
stripe width	The number of disks in a storage pool .
super group	The group that has administrative access to the Greenplum HD EE cluster.
TaskTracker	The process that starts and tracks MapReduce tasks on a node. The TaskTracker receives task assignments from the JobTracker and reports the results of each task back to the JobTracker on completion.
volume	A tree of files, directories, and other volumes, grouped for the purpose of applying a policy or set of policies to all of them at once.
warden	A Greenplum HD EE process that coordinates the starting and stopping of configured services on a node.
ZooKeeper	A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.