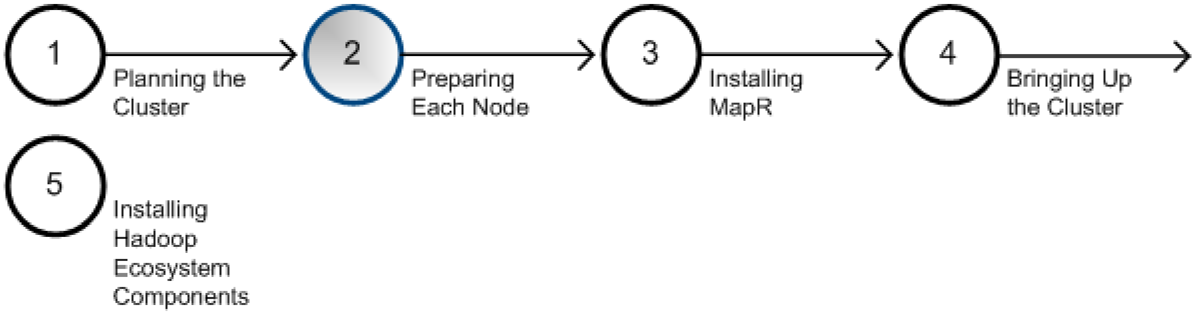
Preparing Each Node



Each node contributes to the cluster designed in the previous step, so each must be able to run MapR and Hadoop software.

	Requirements
CPU	64-bit
OS	Red Hat, CentOS, SUSE, or Ubuntu
Memory	8 GB minimum, more in production
Disk	Raw, unformatted drives and partitions
DNS	Hostname, reaches all other nodes
Users	Common users across all nodes; passwordless ssh (optional)
Java	Must run Java
Other	NTP, Syslog, PAM

Use the following sections as a checklist to make each candidate node suitable for its assigned roles. Once each node has been prepared or disqualified, proceed to Step 3, Installing MapR Software.

2.1 CPU and Operating System

☑ a. Processor is 64-bit

To determine the processor type, run

```
$ uname -m
x86_64
```

If the output includes "x86_64," the processor is 64-bit. If it includes "i386," "i486," "i586," or "i686," it is a 32-bit processor, which is not supported by MapR software.

If the results are "unknown," or none of the above, try one of these alternative commands.

```
$ uname -a
Linux mach-name 2.6.35-22-server #33-Ubuntu SMP Sun Sep 19 20:48:58 UTC 2012 x86_64
GNU/Linux
```

In the `cpuinfo` file, the flag 'lm' (for "long-mode") indicates a 64-bit processor.

```
$ grep flags /proc/cpuinfo
flags               : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss syscall nx rdtscp lm constant_tsc up
arch_perfmon pebs bts rep_good xtopology tsc_reliable nonstop_tsc aperfmperf pni
pclmulqdq ssse3 cx16 sse4_1 sse4_2 popcnt aes hypervisor lahf_lm ida arat
```

☑ b. Operating System is supported

Run the following command to determine the name and version of the installed operating system. (If the lsb_release command reports "No LSB modules are available," this is not a problem.)

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 10.10
Release:        10.10
Codename:       maverick
```

The operating system must be one of the following:

Operating System	Minimum version
RedHat Enterprise Linux (RHEL) or Community Enterprise Linux (CentOS)	6.1 or later
SUSE Enterprise Linux Server	11 SP2 or later
Ubuntu Linux	12.04 or later

If the `lsb_release` command is not found, try one of the following alternatives.

```
$ cat /proc/version
Linux version 2.6.35-22-server (build@allspice) (gcc version 4.4.5 (Ubuntu/Linaro
4.4.4-14ubuntu4) ) #33-Ubuntu SMP Sun Sep 19 20:48:58 UTC 2012
```

```
$ cat /etc/*-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=10.10
DISTRIB_CODENAME=maverick
DISTRIB_DESCRIPTION="Ubuntu 10.10"
```

If you determine that the node is running an older version of a supported OS, upgrade to at least a supported version and test the upgrade before proceeding. If you find a different Linux distribution, such as Fedora or Gentoo, the node must be reformatted and a supported distro installed.

2.2 Memory and Disk Space

☑ a. Minimum Memory

A minimum of 8 GB total memory is required on a node. MapR recommends at least 16 GB for a production environment, and typical MapR production nodes have 32 GB or more.

Run `free -g` to display total and available memory in gigabytes.

```
$ free -g
              total        used        free      shared     buffers      cached
Mem:              3           2           1           0           0           1
-/+ buffers/cache:            0           2
Swap:             2           0           2
```

If the `free` command is not found, there are many alternatives: `grep MemTotal: /proc/meminfo`, `vmstat -s -SM`, `top`, or various GUI system information tools.

MapR does not recommend using the numad service, since it has not been tested and validated with MapR. Using numad can cause artificial memory constraints to be set which can lead to performance degradation under load. To disable numad:

1. Stop the service by issuing the command `service numad stop`.
2. Set the numad service *not* to start on reboot: `chkconfig numad off`

MapR does not recommend using *overcommit* because it may lead to the kernel memory manager killing processes to free memory, resulting in killed MapR processes and system instability. Set `vm.overcommit_memory` to 0:

1. Edit the file `/etc/sysctl.conf` and add the following line:

   ```
   vm.overcommit_memory=0
   ```

2. Save the file and run:

   ```
   sysctl -p
   ```

> You can try MapR out on non-production equipment, but under the demands of a production environment, memory needs to be balanced against disks, network and CPU.

b. Storage

MapR manages raw, unformatted devices directly to optimize performance and offer high availability. For data nodes, allocate at least 3 unmounted physical drives or partitions for MapR storage. MapR uses disk spindles in parallel for faster read/write bandwidth and therefore groups disks into sets of three.

> MapR requires a minimum of one disk or partition for MapR data. However, file contention for a shared disk decreases performance. In a typical production environment, multiple physical disks on each node are dedicated to the distributed file system, which results in much better performance.

Drive Configuration

Do not use RAID or Logical Volume Management with disks that will be added to MapR. While MapR supports these technologies, using them incurs additional setup overhead and can affect your cluster's performance. Due to the possible formatting requirements that are associated with changes to the drive settings, configure the drive settings prior to installing MapR.

If you have a RAID controller, configure it to run in HBA mode. For LSI MegaRAID controllers that do not support HBA, configure the following drive group settings for optimal performance:

Property **(The actual name depends on the version)**	**Recommended Setting**
RAID Level	RAID0
Stripe Size	>=256K

Cache Policy or I/O Policy	Cached IO or Cached
Read Policy	Always Read Ahead or Read Ahead
Write Policy	Write-Through
Disk Cache Policy or Drive Cache	Disabled

> Enabling the Disk Cache policy can improve performance. However, MapR does not recommend enabling the Disk Cache policy because it increases the risk of data loss if the node loses power before the disk cache is committed to disk.

Minimum Disk Space

OS Partition. Provide at least 10 GB of free disk space on the operating system partition.

MapR-FS. Provide at least 8 GB of free disk space for MapR-FS.

Disk. Provide 10 GB of free disk space in the `/tmp` directory and 128 GB of free disk space in the `/opt` directory. Services, such as JobTracker and TaskTracker, use the `/tmp` directory. Files, such as logs and cores, use the `/opt` directory.

Swap space. Provide sufficient swap space for stability, 10% more than the node's physical memory, but not less than 24 GB and not more than 128 GB.

ZooKeeper. On ZooKeeper nodes, dedicate a partition, if practicable, for the `/opt/mapr/zkdata` directory to avoid other processes filling that partition with writes and to reduce the possibility of errors due to a full `/opt/mapr/zkdata` directory. This directory is used to store snapshots that are up to 64 MB. Since the four most recent snapshots are retained, reserve at least 500 MB for this partition. Do not share the physical disk where `/opt/mapr/zkdata` resides with any MapR File System data partitions to avoid I/O conflicts that might lead to ZooKeeper service failures.

2.3 Connectivity

☑ a. Hostname

Each node in the cluster must have a unique hostname, resolvable forward and backward with every other node with both normal and reverse DNS name lookup.

Run `hostname -f` to check the node's hostname. For example:

```
$ hostname -f
node125
```

If `hostname -f` returns a name, run `` getent hosts `hostname` `` to return the node's IP address and fully-qualified domain name (FQDN).

```
$ getent hosts `hostname`
10.250.1.53     node125.corp.example.com
```

To troubleshoot hostname problems, edit the `/etc/hosts` file as root. A simple `/etc/hosts` might contain:

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

A common problem is an incorrect loopback entry (127.0.x.x) that prevents the IP address from being assigned to the hostname. For example, on Ubuntu, the default `/etc/hosts` file might contain:

```
127.0.0.1     localhost
127.0.1.1     node125.corp.example.com
```

A loopback (`127.0.x.x`) entry with the node's hostname will confuse the installer and other programs. Edit the `/etc/hosts` file and delete any entries that associate the hostname with a loopback IP. Only associate the hostname with the actual IP address.

For more information about Ubuntu's default `/etc/hosts` file, see https://bugs.launchpad.net/ubuntu/+source/cloud-init/+bug/871966.

Use the `ping` command to verify that each node can reach the others using each node's hostname. For more information, see the hosts(5) man page.

b. Common Users

A user that accesses the cluster must have the same credentials and user ID (uid) on each node in the cluster. Every person or department that runs MapR jobs must have an account and must also belong to a common group ID (gid). The uid for each user, and the gid for each group, must be consistent across all nodes.

A 'mapr' user must exist. The 'mapr' user has full privileges to administer the cluster. If you create the 'mapr' user before you install MapR, you can test for connectivity issues. If you do not create the 'mapr' user, installing MapR automatically creates the user for you. The 'mapr' user ID is automatically created on each node if you do not use a directory service, such as LDAP.

To create a group, add a user to the group, or create the 'mapr' user, run the following command as root substituting a uid for *m* and a gid for *n*. (The error "cannot lock /etc/passwd" suggests that the command was not run as root.)

```
$ useradd mapr --gid n --uid m
```

Example:

```
$ groupadd -g 5000 mapr
$ useradd -g 5000 -u 5000 mapr
```

To verify that the users or groups were created, `su mapr`. Verify that a home directory was created (usually `/home/mapr`) and that the users or groups have read-write access to it. The users or groups must have write access to the /tmp directory, or the warden will fail to start services.

c. Optional: Passwordless ssh

If you plan to use the scripted rolling upgrade procedure to upgrade the cluster in the future, it is very helpful for the common user to be able to ssh from each webserver node to any other node without providing a password. Otherwise, passwordless ssh between nodes is optional because MapR will run without it.

Setting up passwordless ssh is straightforward. On each webserver node, generate a key pair and append the key to an authorization file. Then copy this authorization file to each node, so that every node is available from the webserver node.

`su mapr` (if you are not already logged in as mapr)
`ssh-keygen -t rsa -P '' -f ~/`*`filename`*

The `ssh-keygen` command creates *`filename`*, containing the private key, and *`filename`*`.pub`, containing the public key. For convenience, you may want to name the file for the hostname of the node. For example, on the node with hostname "node10.10.1.1,"

```
ssh-keygen -t rsa -P '' -f ~/node10.10.1.1
```

In this example, append the file `/home/mapr/node10.10.1.1.pub` to the `authorized_keys` file.

Append each webserver node's public key to a single file, using a command like `cat` *`filename`*`.pub >> authorized_keys`. (The key file is simple text, so you can append the file in several ways, including a text editor.) When every webserver node's empty passphrase public key has been generated, and the public key file has been appended to the master "authorized_keys" file, copy this master keys file to each node as `~/.ssh/authorized_keys`, where ~ refers to the mapr user's home directory (typically `/home/mapr`).

2.4 Software

☑ a. Java

MapR services require the Java runtime environment (JRE) and Java Development Kit (JDK).

Run `java -version`. Verify that one of these versions is installed on the node:

- Sun Java JDK 1.7 or 1.8
- OpenJDK 1.7 or 1.8

If the `java` command is not found, download and install Oracle/Sun Java or use a package manager to install OpenJDK. Obtain the Oracle/Sun Java Runtime Environment (JRE), Standard Edition (Java SE), available at Oracle's Java SE website. Find Java SE 7 in the archive of previous versions.

> The `openjdk-devel` package includes the `jps` command that lists running Java processes and can show whether the CLDB has started. This command is not supported in the Sun Java SDK.

Use a package manager, such as **yum** (RedHat or CentOS), **apt-get** (Ubuntu) or **zypper** (SuSE) to install or update OpenJDK on the node. The command will be something like one of these:

Red Hat or CentOS

```
# yum install java-1.7.0-openjdk-devel
```

Ubuntu

```
# apt-get install openjdk-7-jdk
```

SUSE

```
# zypper install openjdk-7-jdk
```

> If you have CentOS or RedHat nodes that use Java 1.7, verify that nss 3.19 or greater is installed on each node in the cluster to prevent SSL connection errors.

☑ b. MySQL

The MapR Metrics service requires access to a MySQL server running version 5.1 or later. MySQL does not have to be installed on a node in the cluster, but it must be on the same network as the cluster. If you do not plan to use MapR Metrics, MySQL is not required.

2.5 Infrastructure

☑ a. Network Time

To keep all cluster nodes time-synchronized, MapR requires software such as a Network Time Protocol (NTP) server 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 MapR services. MapR raises a Time Skew alarm on any out-of-sync nodes. See http://www.ntp.org/ for more information about obtaining and installing NTP.

Advanced: Installing an internal NTP server keeps your cluster synchronized even when an outside NTP server is inaccessible.

☑ b. Syslog

Syslog must be enabled on each node to preserve logs regarding killed processes or failed jobs. Modern versions such as syslog-ng and rsyslog are possible, making it more difficult to be sure that a syslog daemon is present. One of the following commands should suffice:

```
syslogd -v
service syslog status

rsyslogd -v
service rsyslog status
```

☑ c. Default umask

Ensure that the default umask for the root user is set to 0022 on all mapr nodes in the cluster. The umask setting is changed in the /etc/profile file, or in the .cshrc or .login file. The root user must have a 0022 umask because the MapR admin user requires access to all files and directories under the /opt/mapr directory, even those initially created by root services.

☑ d. ulimit

`ulimit` is a command that sets limits on the user's access to system-wide resources. Specifically, it provides control over the resources available to the shell and to processes started by it.

The mapr-warden script uses the `ulimit` command to set the maximum number of file descriptors (`nofile`) and processes (`nproc`) to 64000. Higher values are unlikely to result in an appreciable performance gain. Lower values, such as the default value of 1024, are likely to result in task failures.

> ⚠ MapR's recommended value is set automatically every time warden is started.

Depending on your environment, you might want to set limits manually rather than relying on Warden to set them automatically using `ulimit`. The following examples show how to do this, using the recommended value of 64000.

Setting resource limits on Centos/Redhat

1. Edit `/etc/security/limits.conf` and add the following line:

   ```
   <MAPR_USER> - nofile 64000
   ```

2. Edit `/etc/security/limits.d/90-nproc.conf` and add the following line:

   ```
   <MAPR_USER> - nproc 64000
   ```

3. Check that the `/etc/pam.d/su` file contains the following settings:

```
#%PAM-1.0

   auth            sufficient      pam_rootok.so

   # Uncomment the following line to implicitly trust users in the "wheel"
group.

   #auth           sufficient      pam_wheel.so trust use_uid

   # Uncomment the following line to require a user to be in the "wheel"
group.

   #auth           required        pam_wheel.so use_uid

   auth            include         system-auth

   account         sufficient      pam_succeed_if.so uid = 0 use_uid quiet

   account         include         system-auth

   password        include         system-auth

   session         include         system-auth

   session         required        pam_limits.so

   session         optional        pam_xauth.so
```

Setting resource limits on Ubuntu

1. Edit `/etc/security/limits.conf` and add the following lines:

```
<MAPR_USER> - nofile 64000
<MAPR_USER> - nproc 64000
```

2. Edit `/etc/pam.d/su` and uncomment the following line:

```
session    required    pam_limits.so
```

Use `ulimit` to verify settings:

1. Reboot the system.
2. Run the following command as the MapR user (not root) at a command line:

```
ulimit -n
```

The command should report `64000`.

e. PAM

Nodes that will run the **MapR Control System** (the `mapr-webserver` service) can take advantage of Pluggable Authentication Modules (PAM) if found. Configuration files in `/etc/pam.d/` directory are typically provided for each standard Linux command. MapR can use, but does not require, its own profile.

For more detail about configuring PAM, see PAM Configuration.

☑ f. Security - SELinux, AppArmor

SELinux (or the equivalent on other operating systems) must be disabled during the install procedure. If the MapR services run as a non-root user, SELinux can be enabled after installation and while the cluster is running.

☑ g. TCP Retries

On each node, set the number of TCP retries to 5 so that MapR can detect unreachable nodes with less latency.

1. Edit the file `/etc/sysctl.conf` and add the following line:

   ```
   net.ipv4.tcp_retries2=5
   ```

2. Save the file and run:

   ```
   sysctl -p
   ```

☑ h. NFS

Disable the stock Linux NFS server on nodes that will run the MapR NFS server.

☑ i. iptables

Enabling iptables on a node may close ports that are used by MapR. If you enable iptables, make sure that required ports remain open. Check your current IP table rules with the following command:

```
$ service iptables status
```

☑ j. Transparent Huge Pages (THP)

For data-intensive workloads, MapR recommends disabling the Transparent Huge Pages (THP) feature in the Linux kernel.

RHEL

```
$ echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
```

Ubuntu

```
$ echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

Automated Configuration

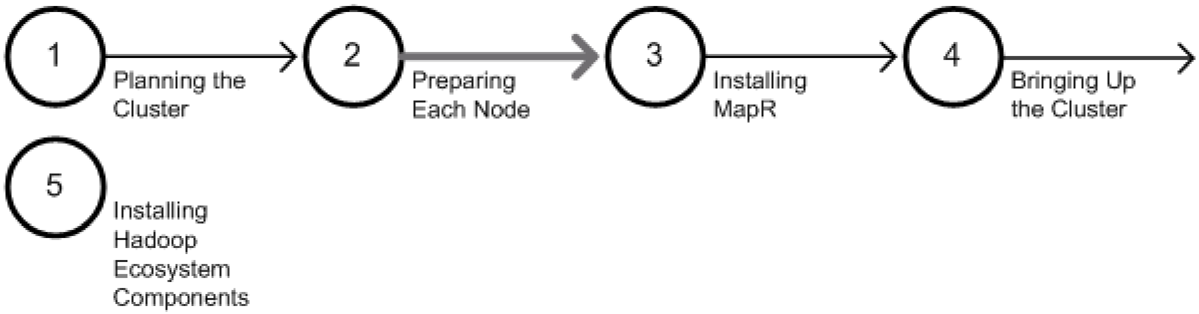
Some users find tools like Puppet or Chef useful to configure each node in a cluster. Make sure, however, that any configuration tool does not reset changes made when MapR packages are later installed. Specifically, do not let automated configuration tools overwrite changes to the following files:

- `/etc/sudoers`
- `/etc/sysctl.conf`
- `/etc/security/limits.conf`
- `/etc/udev/rules.d/99-mapr-disk.rules`

Next Step

Each prospective node in the cluster must be checked against the requirements presented here. Failure to ensure that each node is suitable for use generally leads to hard-to-resolve problems with installing Hadoop.

After each node has been shown to meet the requirements and has been prepared, you are ready to Install MapR components.


1
Planning the Cluster
2
Preparing Each Node
3
Installing MapR
4
Bringing Up the Cluster
5
Installing Hadoop Ecosystem Components