configure.sh

Run configure.sh to set up a MapR cluster node, or to set up a MapR client node for communication with one or more clusters. You can also run configure.sh to update the configuration of a node. For example, you can use configure.sh to change the services running on a node, specify a my SQL database for storing MapR Metrics data or specify the user that runs MapR services.

> On a Windows client, the configure.sh script is named configure.bat. It requires the -c parameter and does not accept the -Z parameter, but otherwise works in a similar way.

Steps Performed by configure.sh

Each time configure.sh is run, it performs the following steps:

- **Updates /opt/mapr/conf/mapr-clusters.conf with the cluster name.** It creates or modifies a line in /opt/mapr/conf/mapr-clusters.conf containing a cluster name followed by a list of CLDB nodes. New entries are added to mapr-clusters.conf when the cluster name passed to the -N parameter is different from the existing cluster name in that file.
- **Checks that the node has at least 4GB of RAM and that the /tmp and /opt partitions each have at least 1 GB of free space.** If these conditions are not met, the script asks for confirmation before continuing.
- **Disables standard NFS daemons.** If the node has the mapr-nfs role, the script disables the standard Linux NFS daemon because both nfs processes cannot run on the same node.
- **Updates additional *.conf and *.xml files related to the cluster and the services running on the node.** For example, yarn-site.xml, warden.conf, and cldb.conf may be updated based on input to configure.sh.
- **On the cluster nodes, it creates a group named shadow, adds the MapR user to this group, and then enables members of the shadow group to view the etc/shadow file.** The read-access to the etc/shadow file enables MapR users to authenticate with the MapR cluster.
- **Starts newly installed services.** As long as warden is running at the time you run configure.sh, new services are started.
- **All changes of configuration options or system files are logged to /opt/mapr/logs/configure.log.** You can use the -L parameter to specify a different log file name.

When you include disk setup options (-D or -F) on nodes with the mapr-fileserver role, the script also performs the following steps:

- **Runs disksetup to create the disktab file.** configure.sh takes the values you specify in the -disk-opts option and passes the value to disksetup. For example, if you include -disk-opts FW5 when you run configure.sh, configure.sh runs disksteup -F -W5
- **Starts Zookeeper and Warden.** When the configure.sh script starts services, the message starting <servicename> is echoed to the standard output, to enable the user to see which services are starting. When Warden starts, Warden and ZooKeeper services are added to the inittab file as the first available inittab IDs, enabling these services to restart automatically upon failure.

  > You can specify the -no-autostart option to prevent the script from starting Zookeeper or Warden when you run configure.sh with the -F or -D options.

Syntax

You can use the following syntax in the `/opt/mapr/server/configure.sh` file:

```
-C cldb_list (hostname[:port_no] [,hostname[:port_no]...])
-M cldb_mh_list (hostname[:port_no][,[hostname[:port_no]...])
-Z zookeeper_list (hostname[:port_no][,hostname[:port_no]...])
-D /dev/disks
-F /path/file.txt
[ -N cluster_name ]
[ -v ]
[-no-autostart]
[ -disk-opts <options> ]
[ -on-prompt-cont [yn] ]
[ -c ]
[ --isvm ]
[-HS <IP address>]
[ -J <CLDB JMX port> ]
[ -L <log file> ]
[ -M7 ]
[ -noDB ]
[ -N <cluster name> ]
[ -R ]
[-RM <IP address>]
```

```
    [ -d <host>:<port> ]
    [ -du <database username> ]
    [ -dp <database password> ]
    [ -ds <schema> ]
    [ --create-user | -a ]
    [ -U <user ID> ]
    [ -u <username> ]
    [ -G <group ID> ]
    [ -g <group name> ]
    [ -H <port_no> ]
    [ -f ]
    [ -genkeys ]
    [ -certdomain <domain> ]
    [ -nocerts ]
    [ -S | -secure ]
    [ -maprpam ]
    [ -K | -kerberosEnable ]
    [ -P "<cldbPrincipal>" ]
    [ -no-auto-permission-update ]
```

Parameters

Parameter	Description
-C	Use the -C option only for CLDB servers that have a single IP address each. This option takes a list of the CLDB nodes that this machine uses to connect to the MapR cluster. The list is in the following format: • hostname[:port_no] [,hostname[:port_no]...]
-M	Use the -M option only for multihomed CLDB servers that have more than one IP address. This option takes a list of the multihomed CLDB nodes that this machine uses to connect to the MapR cluster. The list is in the follwing format: • hostname[:port_no][,[hostname[:port_no]...]]
-Z	The -Z option is required unless -c (lowercase) or -R is specified. This option takes a list of the ZooKeeper nodes in the cluster. The list is in the following format: • hostname[:port_no][,hostname[:port_no]...]
-D *disks*	Specifies a comma-delimited list of disks and partitions to use with the MapR file system. With the -D option, you cannot specify partitions. By default, the configure.sh script automatically starts cluster services after configuration finishes successfully. If you do not want cluster services to be restarted, include the -no-autostart option along with the -D option.
-F *path to file*	Specifies a path to a text file that specifies the disks and partitions to use with the MapR file system. By default, the configure.sh script automatically starts cluster services after configuration finishes successfully. If you do not want cluster services to be restarted, include the -no-autostart option along with the -F option.
-v	In addition to logging information, also prints to `stdout`.
-no-autostart	Specifies that the script should not start Zookeeper or Warden when you run configure.sh.
-disk-opts *options*	Enables you to specify a series of `disksetup` formatting options. Do not include spaces or commas between the disksetup options. For example, you can specify -disk-opts FW5 to format the disks (F) and configure 5 disks per storage pool (W5).
-on-prompt-cont *yn*	Specify y to automatically respond Yes to all prompts. Specify n to automatically respond No to all prompts.
--isvm	Specifies virtual machine setup. Required when configure.sh on a cluster node that is on a virtual machine. This option configures the script to use less memory.
-c	Specifies client setup. See Setting Up the Client.
-J	Specifies the JMX port for the CLDB. Default: 7220
-H <port_no>	Specifies the HTTPS port number for connecting to the CLDB. The default port is 7443.
-HS	Specifies the IP or hostname of the node in the cluster that has the HistoryServer role. This is parameter is required when a node in the cluster contains the HistoryServer role.
-L	Specifies a log file. If not specified, configure.sh logs errors to /opt/mapr/logs/configure.log .

-M7	Deprecated as of version 4.0.1.
-noDB	Specifies that MapR-DB is not in use.
-genkeys	Generates needed keys and certificates for the initial CLDB node in a secure cluster.
-certdomain <domain>	Specifies a DNS domain for generated SSL wildcard certificates. This domain overrides the default DNS domain.
-nocerts	When specified, the `configure.sh` script does not generate SSL certificates even when the -genkeys option is specified.
-S | -secure	Specifies that this cluster is a secure cluster. Cluster security is off by default.
-maprpam	When specified, the configure.sh script installs MapR's version of Pluggable Authentication Modules (PAM). This option is ignored if -S is not set.
-K | -kerberosEnable	Indicates that Kerberos security has been enabled. Kerberos security is disabled by default.
-P "<cldbPrincipal>"	Specifies the Kerberos instance which is used to form a CLDB Kerberos principal in the form of mapr/<instance-name>@<realm-name>. Enclose this value in quotes ("). This value is ignored if Kerberos security is not enabled.
-N	Specifies the cluster name. If you do not specify a name, configure.sh applies a default name (my.cluster.com) to the cluster. Whenever you run configure.sh, you must be aware of the existing cluster name or names in mapr-clusters.conf and specify the -N parameter accordingly. If you specify a name that does not exist, a new line is created in mapr-clusters.conf and treated as a configuration for a separate cluster. Subsequent runs of configure.sh without the -N parameter will operate on this default cluster. If you specify a name when you first run configure.sh, you can modify the CLDB and ZooKeeper settings corresponding to the named cluster by specifying the same name and running configure.sh again. Whenever you need to re-run configure.sh on a given cluster (to add or rename nodes, for example), be sure to specify the same cluster name that you used when you ran configure.sh for the first time.
-R	After initial node configuration, specifies that configure.sh should use the previously configured ZooKeeper and CLDB nodes. The -C and -Z parameters are not required when -R is specified. When -R is specified, the CLDB credentials are read from mapr-clusters.conf and the ZooKeeper credentials are read from warden.conf. Use the -R option when you make changes to the services configured on a node without changing the CLDB and ZooKeeper nodes.
-RM	In 4.0.2, this parameter is not required unless you want to configure manual or automatic failover; zero configuration failover is enabled by default. In 4.0.1, this parameter specifies the nodes in the cluster with the ResourceManager role. List the nodes in the following format: `hostname[,hostname]...]` For more information, see ResourceManager High Availability.
-d	The host and port of the MySQL database to use for storing MapR Metrics data.
-du	The username for logging into the MySQL database used for storing MapR Metrics data.
-dp	The password for logging into the MySQL database used for storing MapR Metrics data.
-ds <schema>	Name of the database schema to use for the MySQL database used for storing MapR Metrics data. The default schema name is metrics.
--create-user or -a	Create a local user to run MapR services, using the specified user from -u or the environment variable $MAPR_USER.
-U	The user ID to use when creating $MAPR_USER with the --create-user or -a option; corresponds to the -u or --uid option of the useradd command in Linux.
-u	The user name under which MapR services will run.
-G	The group ID to use when creating $MAPR_USER with the -create-user or -a option; corresponds to the -g or -gid option of the useradd command in Linux.
-g	The group name under which MapR services will run.
-f	Specifies that the node should be configured without the system prerequisite check.
-no-auto-permission-update	Pass this option to prevent MapR from silently altering permissions in `/etc/shadow`.

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 nodeA,nodeB,nodeC -Z nodeA,nodeB,nodeC
```

Configure a client to work with cluster my.cluster.com, which has one CLDB at nodeA:

On a Linux client, run the following command:

```
/opt/mapr/server/configure.sh -N my.cluster.com -c -C nodeA
```

On a Windows 7 client, run the following command:

```
C:\opt\mapr\server\configure.bat -N my.cluster.com -c -C nodeA
```

Add a second cluster to the configuration:

On a node in the second cluster your.cluster.com, run the following command:

```
configure.sh -C nodeZ -N your.cluster.com -Z <zkNodeA,zkNodeB,zkNodeC>
```

Adding CLDB servers with multiple IP addresses to a cluster:

In this example, the cluster my.cluster.com has CLDB servers at nodeA, nodeB, nodeC, and nodeD. The CLDB servers nodeB and nodeD have two NICs each at eth0 and eth1.

On a node in the cluster my.cluster.com, run the following command:

```
configure.sh -N my.cluster.com -C nodeAeth0,nodeCeth0 -M nodeBeth0,nodeBeth1 -M nodeDeth0,nodeDeth1 -Z zknodeA
```

Starting cluster in secure mode with configure.sh

In this example, the cluster my.cluster.com has two CLDB servers at nodeA and nodeB. The ZooKeeper node for this cluster is at nodeC. To start the cluster in secure mode, run the following command on nodeA::

```
configure.sh  -N my.cluster.com –C nodeA,nodeB –Z nodeC –secure –genkeys –F <disklist file>
```

This command creates the `ssl_truststore`, `ssl_keystore`, `maprserverticket`, and `cldb.key` files. Copy those files from nodeA's `/opt/mapr/conf` directory to nodeB's `/opt/mapr/conf` directory.

On nodeB, change the permissions on these files to the mapr user with the following command:

```
chown 600 ssl_truststore ssl_keystore maprserverticket cldb.key
```

On nodeB, run the following command:

```
configure.sh –N mycluster.com –C nodeA,nodeB –Z nodeC –secure –F <disklist file>
```