Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'mdadm.conf.5'

$ man mdadm.conf.5

MDADM.CONF(5) File Formats Manual MDADM.CONF(5)

NAME

mdadm.conf - configuration for management of Software RAID with mdadm

SYNOPSIS

/etc/mdadm.conf

DESCRIPTION

mdadm is a tool for creating, managing, and monitoring RAID devices us?
ing the md driver in Linux.

Some common tasks, such as assembling all arrays, can be simplified by describing the devices and arrays in this configuration file.

SYNTAX

The file should be seen as a collection of words separated by white space (space, tab, or newline). Any word that beings with a hash sign (#) starts a comment and that word together with the remainder of the line is ignored.

Spaces can be included in a word using quotation characters. Either single quotes (') or double quotes (") may be used. All the characters from one quotation character to next identical character are protected and will not be used to separate words to start new quoted strings. To include a single quote it must be between double quotes. To include a double quote it must be between single quotes.

Any line that starts with white space (space or tab) is treated as though it were a continuation of the previous line.

Empty lines are ignored, but otherwise each (non continuation) line must start with a keyword as listed below. The keywords are case in?sensitive and can be abbreviated to 3 characters.

The keywords are:

DEVICE A device line lists the devices (whole devices or partitions) that might contain a component of an MD array. When looking for the components of an array, mdadm will scan these devices (or any devices listed on the command line).

The device line may contain a number of different devices (sepa?rated by spaces) and each device name can contain wild cards as defined by glob(7).

Also, there may be several device lines present in the file.

Alternatively, a device line can contain either or both of the words containers and partitions. The word containers will cause mdadm to look for assembled CONTAINER arrays and included them as a source for assembling further arrays.

The word partitions will cause mdadm to read /proc/partitions and include all devices and partitions found therein. mdadm does not use the names from /proc/partitions but only the major and minor device numbers. It scans /dev to find the name that matches the numbers.

If no DEVICE line is present in any config file, then "DEVICE partitions containers" is assumed.

For example:

```
DEVICE /dev/hda* /dev/hdc*
DEV    /dev/sd*
DEVICE /dev/disk/by-path/pci*
DEVICE partitions
```

ARRAY The ARRAY lines identify actual arrays. The second word on the line may be the name of the device where the array is normally assembled, such as /dev/md1 or /dev/md/backup. If the name does not start with a slash ('/'), it is treated as being in /dev/md/. Alternately the word <ignore> (complete with angle brackets) can be given in which case any array which matches the rest of the line will never be automatically assembled. If no device name is given, mdadm will use various heuristics to determine an appropriate name.

Subsequent words identify the array, or identify the array as a member of a group. If multiple identities are given, then a component device must match ALL identities to be considered a match. Each identity word has a tag, and equals sign, and some value. The tags are:

uuid= The value should be a 128 bit uuid in hexadecimal, with punctuation interspersed if desired. This must match the uuid stored in the superblock.

name= The value should be a simple textual name as was given to mdadm when the array was created. This must match the name stored in the superblock on a device for that device to be included in the array. Not all superblock formats support names.

super-minor=

The value is an integer which indicates the minor number that was stored in the superblock when the array was created. When an array is created as /dev/mdX, then the minor number X is stored.

devices=

The value is a comma separated list of device names or device name patterns. Only devices with names which match one entry in the list will be used to assemble the array. Note that the devices listed there must also be listed on a DEVICE line.

level= The value is a RAID level. This is not normally used to identify an array, but is supported so that the output of

mdadm --examine --scan

can be use directly in the configuration file.

num-devices=

The value is the number of devices in a complete active ar?ray. As with level= this is mainly for compatibility with the output of

mdadm --examine --scan.

spares=

The value is a number of spare devices to expect the array to have. The sole use of this keyword and value is as fol?lows: mdadm --monitor will report an array if it is found to have fewer than this number of spares when --monitor starts or when --oneshot is used.

spare-group=

The value is a textual name for a group of arrays. All ar?rays with the same spare-group name are considered to be part of the same group. The significance of a group of ar?rays is that mdadm will, when monitoring the arrays, move a spare drive from one array in a group to another array in that group if the first array had a failed or missing drive but no spare.

auto= This option is rarely needed with mdadm-3.0, particularly if use with the Linux kernel v2.6.28 or later. It tells mdadm whether to use partitionable array or non-partitionable ar?rays and, in the absence of udev, how many partition devices to create. From 2.6.28 all md array devices are partition?able, hence this option is not needed.

The value of this option can be "yes" or "md" to indicate that a traditional, non-partitionable md array should be created, or "mdp", "part" or "partition" to indicate that a partitionable md array (only available in linux 2.6 and later) should be used. This later set can also have a number appended to indicate how many partitions to create device files for, e.g. auto=mdp5. The default is 4.

bitmap=

The option specifies a file in which a write-intent bitmap should be found. When assembling the array, mdadm will provide this file to the md driver as the bitmap file. This has the same function as the --bitmap-file option to --assemble.

metadata=

Specify the metadata format that the array has. This is mainly recognised for comparability with the output of mdadm -Es.

container=

Specify that this array is a member array of some container. The value given can be either a path name in /dev, or a UUID of the container array.

member=

Specify that this array is a member array of some container. Each type of container has some way to enumerate member arrays, often a simple sequence number. The value identifies which member of a container the array is. It will usually accompany a "container=" word.

MAILADDR

The mailaddr line gives an E-mail address that alerts should be sent to when mdadm is running in --monitor mode (and was given the --scan option). There should only be one MAILADDR line and it should have only one address. Any subsequent addresses are silently ignored.

MAILFROM

The mailfrom line (which can only be abbreviated to at least 5 characters) gives an address to appear in the "From" address for alert mails. This can be useful if you want to explicitly set a domain, as the default from address is "root" with no domain. All words on this line are catenated with spaces to form the ad?dress.

Note that this value cannot be set via the mdadm commandline. It is only settable via the config file. There should only be one MAILADDR line and it should have only one address. Any sub?sequent addresses are silently ignored.

PROGRAM

The program line gives the name of a program to be run when mdadm --monitor detects potentially interesting events on any of the arrays that it is monitoring. This program gets run with two or three arguments, they being the Event, the md device, and possibly the related component device.

There should only be one program line and it should be given only one program. Any subsequent programs are silently ignored.

CREATE The create line gives default values to be used when creating arrays, new members of arrays, and device entries for arrays.

There should only be one create line. Any subsequent lines will override the previous settings.

Keywords used in the CREATE line and supported values are:

owner=

group= These can give user/group ids or names to use instead of system defaults (root/wheel or root/disk).

mode= An octal file mode such as 0660 can be given to override the default of 0600.

auto= This corresponds to the --auto flag to mdadm. Give yes, md, mdp, part ? possibly followed by a number of partitions ? to indicate how missing device entries should be created.

metadata=

The name of the metadata format to use if none is explicitly given. This can be useful to impose a system-wide default of version-1 superblocks.

names=yes

Since Linux 2.6.29 it has been possible to create md devices
with a name like md_home rather than just a number, like
md3. mdadm will use the numeric alternative by default as
other tools that interact with md arrays may expect only
numbers. If names=yes is given in mdadm.conf then mdadm
will use a name when appropriate. If names=no is given,
then non-numeric md device names will not be used even if
the default changes in a future release of mdadm.

bbl=no By default, mdadm will reserve space for a bad block list
(bbl) on all devices included in or added to any array that
supports them. Setting bbl=no will prevent this, so newly
added devices will not have a bad block log.

HOMEHOST

The homehost line gives a default value for the --homehost= op?
tion to mdadm. There should normally be only one other word on
the line. It should either be a host name, or one of the spe?
cial words <system>, <none> and <ignore>. If <system> is given,
then the gethostname(2) systemcall is used to get the host name.
This is the default.

If <ignore> is given, then a flag is set so that when arrays are
being auto-assembled the checking of the recorded homehost is
disabled. If <ignore> is given it is also possible to give an
explicit name which will be used when creating arrays. This is
the only case when there can be more that one other word on the
HOMEHOST line. If there are other words, or other HOMEHOST
lines, they are silently ignored.

If <none> is given, then the default of using gethostname(2) is
over-ridden and no homehost name is assumed.

When arrays are created, this host name will be stored in the
metadata. When arrays are assembled using auto-assembly, arrays
which do not record the correct homehost name in their metadata
will be assembled using a "foreign" name. A "foreign" name al?
way ends with a digit string preceded by an underscore to dif?

ferentiate it from any possible local name. e.g. /dev/md/1_1 or /dev/md/home_0.

HOMECLUSTER

The homcluster line gives a default value for the --homecluster= option to mdadm. It specifies the cluster name for the md de? vice. The md device can be assembled only on the cluster which matches the name specified. If homcluster is not provided, mdadm tries to detect the cluster name automatically.

There should only be one homecluster line. Any subsequent lines will be silently ignored.

AUTO A list of names of metadata format can be given, each preceded by a plus or minus sign. Also the word homehost is allowed as is all preceded by plus or minus sign. all is usually last.

When mdadm is auto-assembling an array, either via --assemble or --incremental and it finds metadata of a given type, it checks that metadata type against those listed in this line. The first match wins, where all matches anything. If a match is found that was preceded by a plus sign, the auto assembly is allowed. If the match was preceded by a minus sign, the auto assembly is disallowed. If no match is found, the auto assembly is allowed.

If the metadata indicates that the array was created for this host, and the word homehost appears before any other match, then the array is treated as a valid candidate for auto-assembly.

This can be used to disable all auto-assembly (so that only ar? rays explicitly listed in mdadm.conf or on the command line are assembled), or to disable assembly of certain metadata types which might be handled by other software. It can also be used to disable assembly of all foreign arrays - normally such arrays are assembled but given a non-deterministic name in /dev/md/.

The known metadata types are 0.90, 1.x, ddf, imsm.

AUTO should be given at most once. Subsequent lines are silently ignored. Thus a later config file in a config direc? tory will not overwrite the setting in an earlier config file.

POLICY This is used to specify what automatic behavior is allowed on devices newly appearing in the system and provides a way of marking spares that can be moved to other arrays as well as the migration domains. Domain can be defined through policy line by specifying a domain name for a number of paths from /dev/disk/by-path/. A device may belong to several domains. The domain of an array is a union of domains of all devices in that array. A spare can be automatically moved from one array to an? other if the set of the destination array's domains contains all the domains of the new disk or if both arrays have the same spare-group.

To update hot plug configuration it is necessary to execute mdadm --udev-rules command after changing the config file

Keywords used in the POLICY line and supported values are:

domain=

any arbitrary string

metadata=

0.9 1.x ddf or imsm

path= file glob matching anything from /dev/disk/by-path

type= either disk or part.

action=

include, re-add, spare, spare-same-slot, or force-spare

auto= yes, no, or homehost.

The action item determines the automatic behavior allowed for de? vices matching the path and type in the same line. If a device matches several lines with different actions then the most permis? sive will apply. The ordering of policy lines is irrelevant to the end result.

include

allows adding a disk to an array if metadata on that disk matches that array

re-add will include the device in the array if it appears to be a

current member or a member that was recently removed and the

array has a write-intent-bitmap to allow the re-add func?
tionality.

spare as above and additionally: if the device is bare it can be?
come a spare if there is any array that it is a candidate
for based on domains and metadata.

spare-same-slot

as above and additionally if given slot was used by an array
that went degraded recently and the device plugged in has no
metadata then it will be automatically added to that array
(or it's container)

force-spare

as above and the disk will become a spare in remaining cases

PART-POLICY

This is similar to POLICY and accepts the same keyword assign?
ments. It allows a consistent set of policies to applied to
each of the partitions of a device.

A PART-POLICY line should set type=disk and identify the path to
one or more disk devices. Each partition on these disks will be
treated according to the action= setting from this line. If a
domain is set in the line, then the domain associated with each
patition will be based on the domain, but with "-partN" ap?
pended, when N is the partition number for the partition that
was found.

SYSFS The SYSFS line lists custom values of MD device's sysfs at?
tributes which will be stored in sysfs after the array is assem?
bled. Multiple lines are allowed and each line has to contain
the uuid or the name of the device to which it relates. Lines
are applied in reverse order.

uuid= hexadecimal identifier of MD device. This has to match the
uuid stored in the superblock.

name= name of the MD device as was given to mdadm when the array
was created. It will be ignored if uuid is not empty.

MONITORDELAY

The monitordelay line gives a delay in seconds mdadm shall wait before pooling md arrays when mdadm is running in --monitor mode. -d/--delay command line argument takes precedence over the config file.

If multiple MINITORDELAY lines are provided, only first non-zero value is considered.

FILES

/etc/mdadm.conf

The default config file location, used when mdadm is running without --config option.

/etc/mdadm.conf.d

The default directory with config files. Used when mdadm is running without --config option, after successful reading of the /etc/mdadm.conf default config file. Files in that directory are read in lexical order.

/etc/mdadm/mdadm.conf

Alternative config file that is read, when mdadm is running without --config option and the /etc/mdadm.conf default config file was not opened successfully.

/etc/mdadm/mdadm.conf.d

The alternative directory with config files. Used when mdadm is run? ninng without --config option, after reading the /etc/mdadm/mdadm.conf alternative config file whether it was successful or not. Files in that directory are read in lexical order.

EXAMPLE

```
DEVICE /dev/sd[bcdjkl]1

DEVICE /dev/hda1 /dev/hdb1

# /dev/md0 is known by its UUID.

ARRAY /dev/md0 UUID=3aaa0122:29827cfa:5331ad66:ca767371

# /dev/md1 contains all devices with a minor number of

#   1 in the superblock.

ARRAY /dev/md1 superminor=1

# /dev/md2 is made from precisely these two devices
```

```
ARRAY /dev/md2 devices=/dev/hda1,/dev/hdb1
# /dev/md4 and /dev/md5 are a spare-group and spares
#  can be moved between them
ARRAY /dev/md4 uuid=b23f3c6d:aec43a9f:fd65db85:369432df
           spare-group=group1
ARRAY /dev/md5 uuid=19464854:03f71b1b:e0df2edd:246cc977
           spare-group=group1
# /dev/md/home is created if need to be a partitionable md array
# any spare device number is allocated.
ARRAY /dev/md/home UUID=9187a482:5dde19d9:eea3cc4a:d646ab8b
           auto=part
# The name of this array contains a space.
ARRAY /dev/md9 name='Data Storage'
POLICY domain=domain1 metadata=imsm path=pci-0000:00:1f.2-scsi-*
           action=spare
POLICY domain=domain1 metadata=imsm path=pci-0000:04:00.0-scsi-[01]*
           action=include
# One domain comprising of devices attached to specified paths  is  de?
fined.
#  Bare  device  matching  first path will be made an imsm spare on hot
plug.
# If more than one array is created on devices belonging to domain1 and
# one of them becomes degraded, then any imsm spare matching  any  path
for
# given domain name can be migrated.
MAILADDR root@mydomain.tld
PROGRAM /usr/sbin/handle-mdadm-events
CREATE group=system mode=0640 auto=part-8
HOMEHOST <system>
AUTO +1.x homehost -all
SYSFS name=/dev/md/raid5 group_thread_cnt=4 sync_speed_max=1000000
SYSFS    uuid=bead5eb6:31c17a27:da120ba2:7dfda40d    group_thread_cnt=4
sync_speed_max=1000000
```

```
MONITORDELAY 60
```

SEE ALSO

mdadm(8), md(4).