

Docs Home →
MongoDB Manual

Configuration File Options

On this page

Configuration File

File Format

Use the Configuration File

Core Options

systemLog Options

processManagement Options

cloud Options

net Options

security Options

setParameter Option



Docs Menu

storage Options

operationProfiling Options

replication Options

sharding Options

MongoDB Documentation

On this page

Configuration File

Give Feedback
File Format

MongoDB Manual

6.0 (current) ▾

- ▶ Introduction
- ▶ Installation
- MongoDB Shell (mongosh)
- ▶ MongoDB CRUD Operations
- ▶ Aggregation Operations
- ▶ Data Models
- ▶ Indexes
- ▶ Security
- ▶ Replication
- ▶ Sharding
- ▶ Change Streams
- ▶ Time Series
- ▶ Transactions
- ▶ Administration
- ▶ Storage
- ▶ Frequently Asked Questions
- ▼ Reference

auditLog Options

snmp Options

mongos -only Options

Windows Service Options

Removed MMAPv1 Options

The following page describes the configuration options available in MongoDB 6.0. For configuration file options for other versions of MongoDB, see the appropriate version of the MongoDB Manual.

Use the Configuration File

Core Options

systemLog Options

processManagement Options

cloud Options

net Options

security Options

setParameter Option

storage Options

operationProfiling Options

replication Options

sharding Options

auditLog Options

snmp Options

mongos -only Options

Windows Service Options

Removed MMAPv1 Options

Configuration File

You can configure `mongod` and `mongos` instances at startup using a configuration file. The configuration file contains settings that are equivalent to the `mongod` and `mongos` command-line options. See

[Configuration File Settings and Command-Line Options Mapping](#)

[Give Feedback](#)

Options mapping.

Using a configuration file makes managing `mongod` and `mongos` options easier, especially for large-scale deployments.

You can also add comments to the configuration file to explain the server's settings.

- If you installed MongoDB with a package manager such as `yum` or `apt` on Linux or `brew` on macOS, or with the MSI installer on Windows, a default configuration file has been provided as part of your installation:

Platform	Method
----------	--------

Give Feedback

Platform	Method
----------	--------

Linux	apt, yum, or zypper Package Manager
-------	---

macOS	brew Package Manager
-------	----------------------------

Windows	MSI Installer
---------	------------------

- If you

installed MongoDB via a downloaded TGZ or ZIP file, you will need to create your own configuration file.

The basic example configuration is a good place to start.

[Give Feedback](#)

File Format

MongoDB configuration files use the YAML[↗] format [1].

The following sample configuration file contains several `mongod` settings that you may adapt to your local configuration:

ⓘ NOTE

YAML does not support tab characters for indentation: use spaces instead.

```
systemLog:  
  destination:  
    path: "/var/log/mongodb/mongod.log"  
  logAppend: true  
storage:  
  journal:  
    enabled: true
```



[Give Feedback](#)

```
processManagement:  
  fork: true  
net:  
  bindIp: 127.0.0.1  
  port: 27017  
setParameter:  
  enableLocalTimezone: true  
  ...
```

The Linux package init scripts included in the official MongoDB packages depend on specific values for `systemLog.path`, `storage.dbPath`, and `processManagement`. If you modify these settings in the default configuration file, `mongod` may not start.

[1] YAML is a superset of JSON.

Externally Sourced Values

New in version 4.2:
MongoDB supports using expansion directives in configuration files to

[Give Feedback](#)

load externally sourced values.
Expansion directives can load values for specific configuration file options or load the entire configuration file.

The following expansion directives are available:

Expansion Directive	Description
<code>--rest</code>	Allows users to specify a REST endpoint as the external source for configuration options or the full configuration file. If the configuration file includes the <code>rest</code>

Give Feedback

Expansion Directive

expansion,
Description
on

Linux/macO

S, the read access to the configuration file must be limited to the user running the

`mongod`

`mongos`

process

only.

--exec

Allows users to specify a shell or terminal command as the external source for configuration file options or the full configuration file

[Give Feedback](#)

Expansion Directive

The `__exec` directive includes the configuration file.

includes the

`__exec` expansion, on Linux/macOS, the write access to the configuration file must be limited to the user running the `mongod` or `mongos` process only.

For complete documentation, see [Externally Sourced Configuration File Values](#).

Use the Configuration File

[Give Feedback](#)

To configure mongod or mongos using a config file, specify the

config file with the --config option or the -f option, as in the following examples:

For example, the following uses

```
mongod --config <c>
```

```
mongos --config <c>
```

```
mongod --conf
```

```
mongos --conf
```

You can also use the -f alias to specify the configuration file, as in the following:

```
mongod -f /etc
```

```
mongos -f /etc
```

If you installed from a package and have started MongoDB using your system's init script, you are

[Give Feedback](#)

already using a configuration file.

Expansion Directives and `--configExpand`

If you are using expansion directives in the configuration file, you must include the `--configExpand` option when starting the `mongod` or `mongos`. For example:

```
mongod --conf  
mongos --conf
```

If the configuration file includes an expansion directive and you start the `mongod` or `mongos` without specifying that directive in the `--configExpand` option, the `mongod` or `mongos` fails to start.

For complete documentation, see [Externally Sourced Configuration Files](#)

[Give Feedback](#)

Configuration file
Values.

Core Options

systemLog Options

```
systemLog:   
    verbosity:  
    quiet: <bo  
    traceAllEx  
    syslogFaci  
    path: <str  
    logAppend:  
    logRotate:  
    destination  
    timeStampF  
    component:  
        accessC  
        verbos  
        command  
        verbos  
  
    # COMMEN
```

systemLog.verbosity

Type: integer

Default: 0

The default log message verbosity level for components. The

[Give Feedback](#)

verbosity level
determines the
amount of
Informational and
Debug
messages
MongoDB outputs.

[2]

The verbosity level
can range from 0
to 5:

- 0 is the
MongoDB's
default log
verbosity
level, to
include
Informational
messages.
- 1 to 5
increases the
verbosity
level to
include
Debug
messages.

To use a different
verbosity level for
a named
component, use
the component's
verbosity setting.

Give Feedback

For example, use
the
`systemLog.compo`

to set the verbosity
level specifically
for `ACCESS`
components.

See the
`systemLog.compo`
settings for
specific
component
verbosity settings.

For various ways to
set the log
verbosity level, see
[Configure Log
Verbosity Levels](#).

[Give Feedback](#)

[2] Starting in version 4.2, MongoDB includes the Debug verbosity level (1-5) in the log messages. For example, if the verbosity level is 2, MongoDB logs D2. In previous versions, MongoDB log messages only specified D for Debug level.

systemLog.quiet

Type: boolean

Default: false

Run mongos or mongod in a quiet mode that attempts to limit the amount of output.

[Give Feedback](#)

systemLog.quiet

is **not**
recommended for

production
systems as it may
make tracking
problems during
particular
connections much
more difficult.

systemLog.traceAll

Type: boolean

Default: false

Print verbose
information for
debugging. Use for
additional logging
for support-related
troubleshooting.

systemLog.syslogFacility

Type: string

Default: user

The facility level
used when logging
messages to
syslog. The value
you specify must

[Give Feedback](#)

you specify must
be supported by
your operating
system's

implementation of
syslog. To use this
option, you must
set

`systemLog.destination`
to `syslog`.

systemLog.path

Type: string

The path of the log
file to which
`mongod` or `mongos`
should send all
diagnostic logging
information, rather
than the standard
output or the
host's syslog.

MongoDB creates
the log file at the
specified path.

The Linux package
init scripts do not
expect
`systemLog.path`
to change from the
defaults. If you use
the Linux

[Give Feedback](#)

packages and
change
`systemLog.path`,

you will have to
use your own init
scripts and disable
the built-in scripts.

`systemLog.logAppend`

Type: boolean

Default: false

When `true`,
`mongos` or `mongod`
appends new
entries to the end
of the existing log
file when the
`mongos` or `mongod`
instance restarts.
Without this
option, `mongod` will
back up the
existing log and
create a new file.

`systemLog.logRotate`

Type: string

Default: rename

Determines the

[Give Feedback](#)

Determines the behavior for the `logRotate` command when rotating the server log and/or the audit log. Specify either `rename` or `reopen`:

- `rename` renames the log file.
- `reopen` closes and reopens the log file following the typical Linux/Unix log rotate behavior. Use `reopen` when using the Linux/Unix logrotate utility to avoid log loss.

If you specify `reopen`, you must also set

Give Feedback

systemLog.level
to true.

systemLog.destination

Type: string

The destination to which MongoDB sends all log output. Specify either file or syslog. If you specify file, you must also specify systemLog.path

If you do not specify systemLog.destination, MongoDB sends all log output to standard output.

⚠ WARNING

The syslog daemon generates timestamps when it logs a message,

[Give Feedback](#)

not
when
MongoDB
issues
the
message.
This can
lead to
misleading
timestamps
for log
entries,
especially
when
the
system
is under
heavy
load.
We
recommend
using
the
`file`
option
for
production
systems
to
ensure
accurate
timestamps.

Give Feedback

systemLog.timeStamp

Type: string

Default: iso8601-local

The time format
for timestamps in
log messages.
Specify one of the
following values:

Value	Description
iso8601-utc	Displays timestamps in UTC.
iso8601-local	Displays timestamps in local time.



NOTE

Starting
in
MongoDB
4.4,

[Give Feedback](#)

systemLog

no

longer

supports

ctime.

An

example

of

ctime

formatted

date is:

Wed Dec 3

systemLog.component Options

systemLog:



component:

accessC

verb

command

verb

COMM

replica

verb

elect

ve

hear

ve

init

ve

roll

Give Feedback

```
storage:  
verb:  
journal:  
verb:  
recovery:  
verb:  
write:  
verb:
```

NOTE

Starting in version 4.2, MongoDB includes the Debug verbosity level (1-5) in the log messages. For example, if the verbosity level is 2, MongoDB logs D2. In previous versions, MongoDB log

[Give Feedback](#)

messages
only

specified D
for Debug
level.

systemLog.componen

Type: integer

Default: 0

The log message
verbosity level for
components
related to access
control. See
[ACCESS](#)
components.

The verbosity level
can range from 0
to 5:

- 0 is the
MongoDB's
default log
verbosity
level, to
include
Informational

[Give Feedback](#)

messages.

- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.components

Type: integer

Default: 0

The log message verbosity level for components related to commands. See COMMAND components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to

[Give Feedback](#)

include
Informational
messages.

- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.components

Type: integer

Default: 0

The log message verbosity level for components related to control operations. See CONTROL components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to

[Give Feedback](#)

include
Informational
messages.

- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.components

Type: integer

Default: 0

The log message verbosity level for components related to diagnostic data collection operations. See

FTDC components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity

[Give Feedback](#)

level, to

include
Informational
messages.

- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.componen

Type: integer

Default: 0

The log message verbosity level for components related to geospatial parsing operations. See GEO components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's

[Give Feedback](#)

MongoDB

default log

verbosity

level, to

include

Informational

messages.

- 1 to 5

increases the
verbosity

level to

include

Debug

messages.

systemLog.components

Type: integer

Default: 0

The log message
verbosity level for
components
related to indexing
operations. See
[INDEX](#)
components.

The verbosity level
can range from 0
to 5:

- 0 is the
MongoDB's

[Give Feedback](#)

mongodbs

default log

verbosity

level, to

include

Informational

messages.

- 1 to 5

increases the

verbosity

level to

include

Debug

messages.

systemLog.components

Type: integer

Default: 0

The log message
verbosity level for
components
related to
networking
operations. See
NETWORK
components.

The verbosity level
can range from 0
to 5:

- 0 is the

[Give Feedback](#)

- ⌂ ⌂ ⌂ ⌂

MongoDB's
default log
verbosity

level, to
include
Informational
messages.

- 1 to 5
increases the
verbosity
level to
include
Debug
messages.

systemLog.component

Type: integer

Default: 0

The log message
verbosity level for
components
related to query
operations. See
[QUERY](#)
components.

The verbosity level
can range from 0
to 5:

- 0 is the

[Give Feedback](#)



MongoDB's
default log
verbosity

level, to
include
Informational
messages.

- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.components

Type: integer

Default: 0

The log message verbosity level for components related to replication. See REPL components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's

[Give Feedback](#)

mongodbs

default log

verbosity

level, to

include

Informational

messages.

- 1 to 5

increases the

verbosity

level to

include

Debug

messages.

systemLog.componen

Type: integer

Default: 0

New in version 4.2.

The log message
verbosity level for
components
related to election.

See ELECTION
components.

If

systemLog.compo
is unset,

systemLog.compo
level also applies

Give Feedback

to election
components.

The verbosity level
can range from 0
to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: 0

The log message verbosity level for components related to

[Give Feedback](#)

heartbeats. See

`REPL_HB`

components.

If

`systemLog.compo`

is unset,

`systemLog.compo`

level also applies

to heartbeats

components.

The verbosity level

can range from `0`

to `5`:

- `0` is the MongoDB's default log verbosity level, to include Informational messages.
- `1` to `5` increases the verbosity level to include Debug messages.

`systemLog.componen`

Give Feedback

Type: integer

Default: 0

New in version 4.2.

The log message verbosity level for components related to initialSync. See INITSYNC components.

If

systemLog.compo is unset, systemLog.compo level also applies to initialSync components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.

Give Feedback

- 1 to 5 increases the verbosity

level to include Debug messages.

`systemLog.component`

Type: integer

Default: 0

The log message verbosity level for components related to rollback.

See `ROLLBACK` components.

If

`systemLog.component` is unset, `systemLog.component` level also applies to rollback components.

The verbosity level can range from 0 to 5:

- 0 is the

Give Feedback

MongoDB's default log verbosity level, to include informational messages.

- 1 to 5 increases the verbosity level to include debug messages.

systemLog.components

Type: integer

Default: 0

The log message verbosity level for components related to sharding. See SHARDING components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's

[Give Feedback](#)

mongodbs

default log

verbosity

level, to

include

Informational

messages.

- 1 to 5

increases the

verbosity

level to

include

Debug

messages.

systemLog.componen

Type: integer

Default: 0

The log message
verbosity level for
components
related to storage.

See STORAGE
components.

If

systemLog.compo
is unset,

systemLog.compo
level also applies
to journaling

Give Feedback

components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: 0

The log message verbosity level for components related to

[Give Feedback](#)

journaling. See
`JOURNAL`
components.

If
`systemLog.compo`
is unset, the
journaling
components have
the same verbosity
level as the parent
storage
components: i.e.
either the
`systemLog.compo`
level if set or the
default verbosity
level.

The verbosity level
can range from `0`
to `5`:

- `0` is the
MongoDB's
default log
verbosity
level, to
include
Informational
messages.
- `1` to `5`
increases the

Give Feedback

verbosity

level to

include

Debug

messages.

systemLog.componentVerbosity

Type: integer

Default: 0

New in version 4.0.

The log message verbosity level for components related to recovery. See [RECOVERY](#) components.

If

[systemLog.components](#)

is unset,

[systemLog.componentVerbosity](#)

level also applies

to recovery

components.

The verbosity level

can range from 0

to 5:

[Give Feedback](#)

- `0` is the MongoDB's default log verbosity level, to include Informational messages.

- `1` to `5` increases the verbosity level to include Debug messages.

`systemLog.componen`

Type: integer

Default: -1

New in version 5.3.

The log message verbosity level for components related to the WiredTiger storage engine. See `WT` components.

The verbosity level can range from `0`

[Give Feedback](#)

to 5:

- 0 is the MongoDB's default log verbosity level, to include Informational messages.
- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.component

Type: integer

Default: -1

New in version 5.3.

The log message verbosity level for components related to backup operations performed by the

[Give Feedback](#)

WiredTiger storage engine. See `WTBACKUP` components.

The verbosity level can range from `0` to `5`:

- `0` is the MongoDB's default log verbosity level, to include informational messages.
- `1` to `5` increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components

[Give Feedback](#)

related to
checkpoint
operations

performed by the
WiredTiger storage
engine. See
`WTCHKPT`
components.

The verbosity level
can range from `0`
to `5`:

- `0` is the
MongoDB's
default log
verbosity
level, to
include
Informational
messages.
- `1` to `5`
increases the
verbosity
level to
include
Debug
messages.

`systemLog.components`

Type: integer

[Give Feedback](#)

Default: -1

New in version 5.3.

The log message verbosity for components related to compaction operations performed by the WiredTiger storage engine. See [WTCMPCT](#) components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include

[Give Feedback](#)

Debug
messages.

systemLog.components

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components related to eviction operations performed by the WiredTiger storage engine. See [WTEVICT](#) components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include Informational messages.

[Give Feedback](#)

- 1 to 5 increases the verbosity

level to include Debug messages.

systemLog.component

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components related to history store operations performed by the WiredTiger storage engine. See [WTHS](#) components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level to

[Give Feedback](#)

level, to include Informational messages.

- 1 to 5 increases the verbosity level to include Debug messages.

systemLog.componentVerbosity

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components related to recovery operations performed by the WiredTiger storage engine. See [WTRECOV](#) components.

The verbosity level can range from 0 to 5;

- 0 is the

[Give Feedback](#)

- `0` is true

MongoDB's
default log
verbosity

level, to
include
Informational
messages.

- `1` to `5`

increases the
verbosity
level to
include
Debug
messages.

systemLog.components

Type: integer

Default: -1

New in version 5.3.

The log message
verbosity for
components
related to rollback
to stable (RTS)
operations
performed by the
WiredTiger storage
engine. See `WTRTS`
components.

[Give Feedback](#)

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components related to salvage operations performed by the

[Give Feedback](#)

WiredTiger storage engine. See [WTSLVG](#) components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components

[Give Feedback](#)

related to tiered storage operations performed by the

WiredTiger storage engine. See `WT_TIER` components.

The verbosity level can range from `0` to `5`:

- `0` is the MongoDB's default log verbosity level, to include Informational messages.
- `1` to `5` increases the verbosity level to include Debug messages.

`systemLog.components`

Type: integer

Default: -1

Give Feedback

New in version 5.3.

The log message verbosity for components related to timestamps used by the WiredTiger storage engine.

See [WTTS](#)

components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

[Give Feedback](#)

systemLog.componen

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components related to transaction operations performed by the WiredTiger storage engine. See WTTXN components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity

Give Feedback

level to

include
Debug
messages.

systemLog.components

Type: integer

Default: -1

New in version 5.3.

The log message
verbosity for
components
related to
verification
operations
performed by the
WiredTiger storage
engine. See

WTVRFY

components.

The verbosity level
can range from 0
to 5:

- 0 is the
MongoDB's
default log
verbosity

[Give Feedback](#)

verbosity

level, to

include

Informational

messages.

- 1 to 5 increases the verbosity level to include Debug messages.

`systemLog.componen`

Type: integer

Default: -1

New in version 5.3.

The log message verbosity for components related to log write operations performed by the WiredTiger storage engine. See `WTWRTLOG` components.

The verbosity level

Give Feedback

can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: 0

New in version 4.0.2.

The log message verbosity level for components related to transaction. See [TXN components](#)

[Give Feedback](#)

Log Components.

The verbosity level can range from 0 to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

`systemLog.components`

Type: integer

Default: 0

The log message verbosity level for components related to write

[Give Feedback](#)

operations. See
`WRITE`
components.

The verbosity level
can range from 0
to 5:

- 0 is the MongoDB's default log verbosity level, to include informational messages.
- 1 to 5 increases the verbosity level to include debug messages.

processManagement Options

```
processManagement
  fork: <bool>
  pidFilePath:
  timeZoneInfo:
```

`processManagement.`

Give Feedback

Type: boolean

Default: false

Enable a daemon mode that runs the mongos or mongod process in the background. By default mongos or mongod does not run as a daemon: typically you will run mongos or mongod as a daemon, either by using processManagement or by using a controlling process that handles the daemonization process (e.g. as with upstart and systemd).

The processManagement option is not supported on Windows.

The Linux package

Give Feedback

init scripts do not expect `processManagement` to change from the defaults. If you use the Linux packages and change `processManagement`, you will have to use your own init scripts and disable the built-in scripts.

processManagement.

Type: string

Specifies a file location to store the process ID (PID) of the `mongos` or `mongod` process. The user running the `mongod` or `mongos` process must be able to write to this path. If the `processManagement` option is not specified, the process does not create a PID file

[Give Feedback](#)

`createOrOpen`

This option is generally only useful in combination with the `processManagement` setting.

ⓘ NOTE

Linux

On Linux, PID file management is generally the responsibility of your distro's init system: usually a service file in the `/etc/init` directory, or a `systemd`

[Give Feedback](#)

systemd unit file registered with `systemctl`. Only use the `processManager` option if you are not using one of these init systems. For more information, please see the respective Installation Guide for your operating system.

 **NOTE**

macOS
On macOS

[Give Feedback](#)

...
PID file
management
is
generally
handled
by
`brew`.
Only
use the
`processManagement`
option if
you are
not
using
`brew` on
your
macOS
system.
For
more
information,
please
see the
respective
Installation
Guide
for your
operating
system.

processManagement.

Give Feedback

Type: string

The full path from which to load the time zone database. If this option is not provided, then MongoDB will use its built-in time zone database.

The configuration file included with Linux and macOS packages sets the time zone database path to `/usr/share/zone` by default.

The built-in time zone database is a copy of the Olson/IANA time zone database. It is updated along with MongoDB releases, but the time zone database release cycle differs from

Give Feedback

the MongoDB
release cycle. The
most recent

release of the time
zone database is
available on our
download site[↗].

 **WARNING**

MongoDB
uses the
third
party
timelib[↗]
library
to
provide
accurate
conversions
between
timezones.

Due to a
recent
update,
`timelib`
could
create
inaccurate
time
zone
conversions

[Give Feedback](#)

CONNECTIONS

in older

versions
of
MongoDB.

To
explicitly
link to
the time
zone
database
in
versions
of
MongoDB
prior to
5.0,
4.4.7,
4.2.14,
and
4.0.25,
download
the
time
zone
database
. and
use the
timeZoneI

cloud Options

New in version 4.0.

[Give Feedback](#)



`cloud.monitoring.f`

Type: string

New in version 4.0:

Available for
MongoDB
Community
Edition.

Enables or disables
free MongoDB
Cloud monitoring
•
`cloud.monitorin`
accepts the
following values:

Give Feedback

`runtime` Default. You can disable frequent monitoring at runtime.

To enable frequent monitoring, run `db.enableMonitoring()` and `db.disableMonitoring()`.

To enable frequent monitoring while the mongod process is running without restarting it, users must have the `monitoring` privilege. Run `db.enableMonitoring()` and `db.disableMonitoring()` for details.

`on` Enables frequent monitoring at startup; i.e. monitoring starts at the beginning of the startup, you can enable monitoring later.

`off` Disables frequent monitoring at startup, regardless of whether you have previously enabled it. For free monitoring to work, you must disable a frequent monitoring configuration at runtime.

[Give Feedback](#)



Once enabled, the free monitoring state remains enabled until explicitly disabled. That is, you do not need to re-enable each time you start the server.

For the corresponding command-line option, see
`--enableFreeMon`

`cloud.monitoring.f`

Type: string

New in version 4.0:
Available for
MongoDB
Community
Edition.

Optional tag to describe environment context. The tag can be sent as

[Give Feedback](#)

part of the
free MongoDB
Cloud monitoring
registration at
start up.

For the
corresponding
command-line
option, see

`--freeMonitoring`

net Options

Changed in version

4.2: MongoDB 4.2
deprecates `ssl`
options in favor of `tls`
options with identical
functionality.

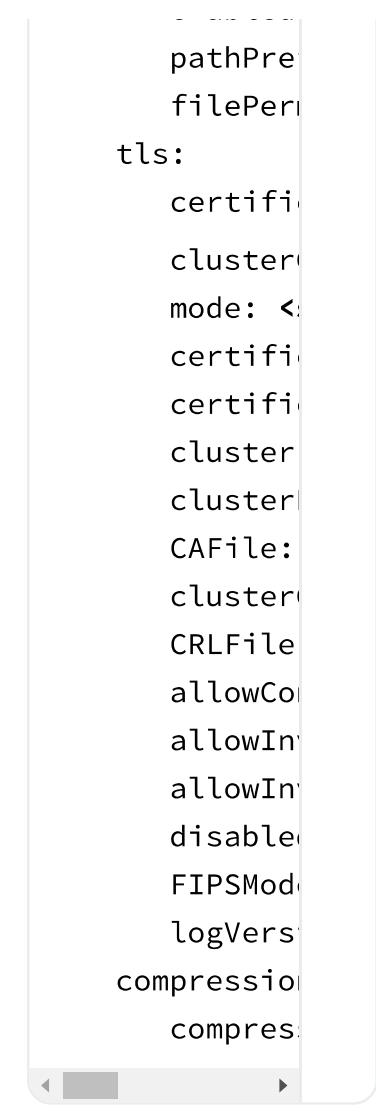
Changed in version

5.0: MongoDB removes
the
`net.serviceExecutor`
configuration option
and the corresponding
`--serviceExecutor`
command-line option.

net:
port: <int>
bindIp: <str>
bindIpAll:
maxIncomingClients:
wireObjectLimit:
ipv6: <bool>
unixDomainSocketPath:
enabled



Give Feedback



```
pathPrepend
filePermit
tls:
  certificate
  cluster
  mode: <string>
  certificate
  certificate
  cluster
  cluster
  CAFile:
  cluster
  CRLFile
  allowConnect
  allowInsecure
  allowInsecure
  disable
  FIPSMode
  logVersion
  compression
  compression
```

net.port

Type: integer

Default:

- 27017 for
`mongod` (if
not a shard
member or a
config server
member) or
`mongos`
instance

[Give Feedback](#)

- 27018 if
mongod is a shard member

- 27019 if
mongod is a config server

The TCP port on which the MongoDB instance listens for client connections.

net.bindIp

Type: string

Default: localhost

The hostnames and/or IP addresses and/or full Unix domain socket paths on which mongos or mongod should listen for client connections. You may attach mongos or mongod to any interface.

To bind to multiple addresses, enter a list of comma-separated values.

[Give Feedback](#)

separated values.

 EXAMPLE

localhost

You can specify both IPv4 and IPv6 addresses, or hostnames that resolve to an IPv4 or IPv6 address.

 EXAMPLE

localhost

 NOTE

If specifying an IPv6 address or a hostname that resolves to an IPv6 address to

[Give Feedback](#)

`net.bindIp`

, you

must

start

`mongos`

or

`mongod`

with

`net.ipv6`

to

enable

IPv6

support.

Specifying

an IPv6

address

to

`net.bindIp`

does

not

enable

IPv6

support.

If specifying a

link-local IPv6 ↗
address

(`fe80::/10`), you

must append the

zone index ↗ to

that address (i.e.

`fe80::<address>`

Give Feedback

EXAMPLE`localhost`**IMPORTANT**

To avoid configuration updates due to IP address changes, use DNS hostnames instead of IP addresses. It is particularly important to use a DNS hostname instead of an IP address when configuring replica

[Give Feedback](#)

set
members
or

sharded
cluster
members.

Use
hostnames
instead
of IP
addresses
to
configure
clusters
across a
split
network
horizon.

Starting
in
MongoDB
5.0,
nodes
that are
only
configured
with an
IP
address
will fail
startup

[Give Feedback](#)

validation

and will
not
start.

 **WARNING**

Before
binding
to a
non-
localhost
(e.g.
publicly
accessible)
IP
address,
ensure
you
have
secured
your
cluster
from
unauthorized
access.
For a
complete

[Give Feedback](#)

list of security recommendations. See the Security Checklist. At a minimum, consider enabling authentication and hardening network infrastructure.

For more information about IP Binding, refer to the IP Binding documentation.

To bind to all IPv4 addresses, enter `0.0.0.0`.

To bind to all IPv4 and IPv6 addresses, enter `::, 0.0.0.0` or starting in MongoDB 4.2, an asterisk `"*"` (enclose the asterisk in quotes).

[Give Feedback](#)

Use quotes

to distinguish from

YAML alias nodes

). Alternatively, use

the

`net.bindIpAll`

setting.

NOTE

- `net.bindIp` and `net.bindIpAll` are mutual exclusive. That is, you can specify one or the other, but not both.

- The command line

[Give Feedback](#)

option
--bind
overrid
the
config
file
setting
`net.bindIp`

To configure cluster nodes for split horizon DNS², use host names instead of IP addresses.

Starting in MongoDB v5.0, `replSetInitiate` and `replSetReconfig` reject configurations that use IP addresses instead of hostnames.

Use `disableSplitHorizon` to modify nodes that cannot be updated to use

[Give Feedback](#)

host names. The parameter only

applies to the configuration commands.

`mongod` and `mongos` do not rely on `disableSplitHorizon` for validation at startup. Legacy `mongod` and `mongos` instances that use IP addresses instead of host names will start after an upgrade.

Instances that are configured with IP addresses log a warning to use host names instead of IP addresses.

`net.bindIpAll`

Type: boolean

Default: false

[Give Feedback](#)

If true, the `mongos` or `mongod` instance binds to

all IPv4 addresses (i.e. `0.0.0.0`). If `mongos` or `mongod` starts with `net.ipv6 : true`, `net.bindIpAll` also binds to all IPv6 addresses (i.e. `::`).

`mongos` or `mongod` only supports IPv6 if started with `net.ipv6 : true`. Specifying `net.bindIpAll` alone does not enable IPv6 support.

 **WARNING**

Before binding to a non-localhost (e.g. publicly

[Give Feedback](#)

accessible)

IP

address,

ensure

you

have

secured

your

cluster

from

unauthorized

access.

For a

complete

list of

security

recommend

see

Security
Checklist

. At

minimum,

consider

enabling
authentication

and

hardening
network
infrastructure

For more
information about
IP Binding, refer to
the IP Binding

Give Feedback

documentation.

Alternatively, set `net.bindIp` to `::, 0.0.0.0` or, starting in MongoDB 4.2, to an asterisk `"*"` (enclose the asterisk in quotes to distinguish from YAML alias `nodes`) to bind to all IP addresses.

 NOTE

`net.bindIp` and `net.bindIp` are mutually exclusive. Specifying both options causes `mongos` or `mongod`

[Give Feedback](#)

to throw
an error
and
terminate.

`net.maxIncomingCon`

Type: integer

Default: 65536

The maximum number of simultaneous connections that mongos or mongod will accept. This setting has no effect if it is higher than your operating system's configured maximum connection tracking threshold.

Do not assign too low of a value to this option, or you will encounter errors during normal application operation.

[Give Feedback](#)

This is particularly useful for a `mongos` if you

have a client that creates multiple connections and allows them to timeout rather than closing them.

In this case, set `maxIncomingConn` to a value slightly higher than the maximum number of connections that the client creates, or the maximum size of the connection pool.

This setting prevents the `mongos` from causing connection spikes on the individual shards. Spikes like these may disrupt the operation and memory allocation

[Give Feedback](#)

of the
sharded cluster.

`net.wireObjectChec`

Type: boolean

Default: true

When true, the mongod or mongos instance validates all requests from clients upon receipt to prevent clients from inserting malformed or invalid BSON into a MongoDB database.

For objects with a high degree of sub-document nesting,

`net.wireObjectC` can have a small impact on performance.

`net.ipv6`

Type: boolean

Default: false

[Give Feedback](#)

Set `net.ipv6` to
`true` to enable
IPv6 support.

`mongos mongod`
disables IPv6
support by default.

Setting `net.ipv6`
does *not* direct the
`mongos mongod` to
listen on any local
IPv6 addresses or
interfaces. To
configure the
`mongos mongod` to
listen on an IPv6
interface, you
must either:

- Configure
`net.bindIp`
with one or
more IPv6
addresses or
hostnames
that resolve
to IPv6
addresses, **or**
- Set
`net.bindIpA`
to `true`.

[Give Feedback](#)

net.unixDomainSocket

Options

net:	↳
unixDomainSocket:	↳
enabled:	↳
pathPrefix:	↳
filePermissions:	↳

net.unixDomainSocket.enabled

Type: boolean

Default: true

Enable or disable listening on the UNIX domain socket.

net.unixDomainSocket.enabled applies only to Unix-based systems.

When

net.unixDomainSocket.enabled is true, mongos or mongod listens on the UNIX socket.

The mongos or mongod process always listens on the UNIX socket

Give Feedback

unless one of the
following is true:

- `net.unixDomainSocket` is `false`
- `--nounixsocket` is set. The command line option takes precedence over the configuration file setting.
- `net.bindIp` is not set
- `net.bindIp` does not specify `localhost` or its associated IP address

`mongos` or `mongod` installed from official .deb and .rpm packages have the `bind_ip` configuration set

[Give Feedback](#)

to `127.0.0.1` by default.

`net.unixDomainSocketPath`

Type: string

Default: `/tmp`

The path for the UNIX socket.

`net.unixDomains` applies only to Unix-based systems.

If this option has no value, the

`mongos` or `mongod` process creates a socket with `/tmp` as a prefix.

MongoDB creates and listens on a UNIX socket unless one of the following is true:

-

`net.unixDomainSocketPath` is `false`

- `--nounixsocketPath` is set

[Give Feedback](#)

- `net.bindIp`
is not set

- `net.bindIp`
does not
specify
`localhost`
or its
associated IP
address

`net.unixDomainSocketPermissions`

Type: int

Default: 0700

Sets the
permission for the
UNIX domain
socket file.

`net.unixDomains`
applies only to
Unix-based
systems.

`net.http` Options

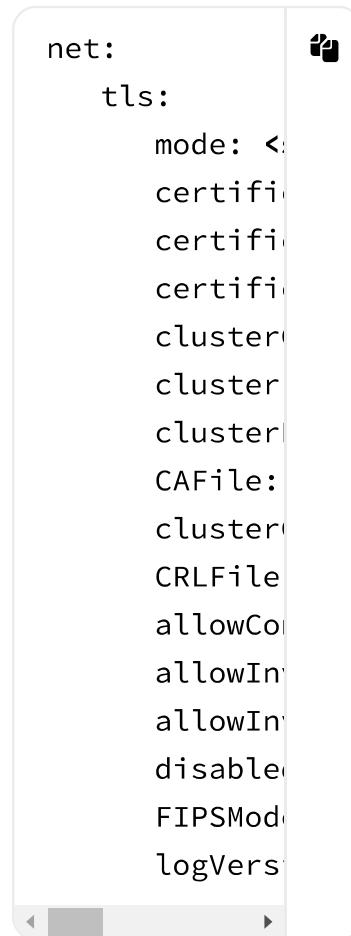
Changed in version
3.6: MongoDB 3.6
removes the
deprecated `net.http`
options. The options
have been deprecated

[Give Feedback](#)

since version 3.2.

net.tls Options

New in version 4.2: The `tls` options provide identical functionality as the previous `ssl` options.



net.tls.mode

Type: string

New in version 4.2.

Enables TLS used
for all network

[Give Feedback](#)

connections. The argument to the

net.tls.mode

setting can be one of the following:

Value	Description
disabled	The server does not use TLS.
allowTLS	Connects between servers not using TLS. For incoming connections, the server accepts both TLS and non-TLS.
preferTLS	Connects between servers using TLS. For incoming connections, the server accepts both TLS and non-TLS.
requireTLS	The server uses a

[Give Feedback](#)

Value	Description
accept	Accept untrusted certificates.
desc	Display certificate details.
encrypt	Encrypt connections.
connect	Connect to a TLS-enabled server.
caFile	Path to a certificate authority file.
certFile	Path to a certificate file.
keyFile	Path to a key file.
tlsCAFile	Path to a certificate authority file.
tlsCertFile	Path to a certificate file.
tlsKeyFile	Path to a key file.
tls	Configure TLS options.

If `--tlsCAFile` or `tls.CAFile` is not specified and you are not using x.509 authentication, the system-wide CA certificate store will be used when connecting to an TLS-enabled server.

If using x.509 authentication, `--tlsCAFile` or `tls.CAFile` must be specified unless using `--tlsCertificatePEM`.

For more information about TLS and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#).

[Give Feedback](#)

net.tls.certificat

Type: string

New in version 4.2:

The .pem file that contains both the TLS certificate and key.

Starting with MongoDB 4.0 on macOS or Windows, you can use the

net.tls.certifi setting to specify a certificate from the operating system's secure certificate store instead of a PEM key file.

certificateKeyF

and

net.tls.certifi are mutually exclusive. You can only specify one.

- On

Linux/BSD,
you must

Give Feedback

specify

`net.tls.cer`

when TLS is
enabled.

- On Windows
or macOS,
you must
specify either

`net.tls.cer`

or

`net.tls.cer`

when TLS is
enabled.



IMPORTANT

For
Windows
only,
MongoDB
4.0
and
later
do
not
support
encrypted
PEM
files.

[Give Feedback](#)

The `mongod` service fails to start if it encounters an encrypted PEM file. To securely store and access a certificate for use with TLS on Windows, use `net.start`.

For more information about TLS and MongoDB, see Configuration Options.

[Give Feedback](#)

configuration options

and

TLS/SSL
Configuration for
Clients

.

`net.tls.certificateKeyFile`

Type: string

New in version 4.2:

The password to
de-crypt the
certificate-key file
(i.e.

`certificateKeyFile`
). Use the
`net.tls.certifi`
option only if the
certificate-key file
is encrypted. In all
cases, the `mongos`
or `mongod` will
redact the
password from all
logging and
reporting output.

Starting in
MongoDB 4.0:

- On
Linux/BSD, if
the private
key in the

[Give Feedback](#)

PEM file is
encrypted
and you do

not specify
the
`net.tls.cer`
option,
MongoDB will
prompt for a
passphrase.

See

TLS/SSL
Certificate
Passphrase.

- On macOS, if the private key in the PEM file is encrypted, you must explicitly specify the `net.tls.cer` option.

Alternatively,
you can use a
certificate
from the
secure
system store
(see

`net.tls.cer`
instead of a

[Give Feedback](#)

/ instead of a

PEM key file

or use an
unencrypted
PEM file.

- On Windows,
MongoDB
does not
support
encrypted
certificates.
The `mongod`
fails if it
encounters
an encrypted
PEM file. Use
`net.tls.cer`
instead.

For more
information about
TLS and MongoDB,
see
[Configure mongoda](#)
and
[TLS/SSL Configuration for Clients](#)

.

`net.tls.certificat`

Type: string

Give Feedback

New in version 4.2:

Available on
Windows and

macOS as an
alternative to
`net.tls.certifi`
. In MongoDB 4.0,
see
`net.ssl.certifi`

Specifies a
certificate
property in order
to select a
matching
certificate from
the operating
system's
certificate store to
use for TLS/SSL.

`net.tls.certifi`
and
`net.tls.certifi`
options are
mutually exclusive.
You can only
specify one.

`net.tls.certifi`
accepts an
argument of the
format
`<property>=<val`

[Give Feedback](#)

where the property can be one of the following:

Property	Value type
----------	------------

subject	ASCII string
---------	--------------

thumbprint	hex string
------------	------------



When using the system SSL certificate store, OCSP (Online

[Give Feedback](#)

Certificate Status
Protocol) is used to

validate the
revocation status
of certificates.

The `mongod`
searches the
operating system's
secure certificate
store for the CA
certificates
required to
validate the full
certificate chain of
the specified TLS
certificate.

Specifically, the
secure certificate
store must contain
the root CA and
any intermediate
CA certificates
required to build
the full certificate
chain to the TLS
certificate. Do **not**
use

`net.tls.CAFile`

or

`net.tls.cluster`

[Give Feedback](#)

to specify the root
and intermediate
CA certificate

For example, if the
TLS certificate was
signed with a
single root CA
certificate, the
secure certificate
store must contain
that root CA
certificate. If the
TLS certificate was
signed with an
intermediate CA
certificate, the
secure certificate
store must contain
the intermedia CA
certificate *and* the
root CA
certificate.

 **NOTE**

You
cannot
use the
`rotateCer`
command
or the
`db_rotateC`

[Give Feedback](#)

shell
method
when
using
`net.tls.caCertFile`
or
`--tlsCertificateKeyFile`
set to
`thumbprint`

net.tls.clusterCertificate

Type: string

New in version 4.2.
Available on
Windows and
macOS as an
alternative to
`net.tls.cluster`

Specifies a
certificate
property to select
a matching
certificate from
the operating
system's secure
certificate store to
use for
internal x.509
membership
authentication.

[Give Feedback](#)

`net.tls.cluster`

and

`net.tls.cluster`

options are

mutually exclusive.

You can only

specify one.

`net.tls.cluster`

accepts an

argument of the

format

`<property>=<val`

where the property

can be one of the

following:

Property	Value type
<code>subject</code>	ASCII string

Give Feedback

Property	Value type
----------	------------

thumbprint	hex string
------------	---------------

The mongod

searches the

operating system's

secure certificate

store for the CA

certificates

required to

validate the full

Give Feedback

certificate chain of the specified cluster certificate.

Specifically, the secure certificate store must contain the root CA and any intermediate CA certificates required to build the full certificate chain to the cluster certificate.

Do **not** use

`net.tls.CAFile`

or

`net.tls.cluster`

to specify the root and intermediate CA certificate.

For example, if the cluster certificate was signed with a single root CA certificate, the secure certificate store must contain that root CA certificate. If the cluster certificate was signed with an

[Give Feedback](#)

intermediate CA
certificate, the
secure certificate

store must contain
the intermediate
CA certificate *and*
the root CA
certificate.

*Changed in
version 4.4:*

`mongod / mongos`
logs a warning on
connection if the
presented x.509
certificate expires
within 30 days of
the
`mongod/mongos`
host system time.

See

[x.509 Certificates
Nearing Expiry
Trigger Warnings](#)
for more
information.

`net.tls.clusterFile`

Type: string

New in version 4.2:
The .pem file that
contains the x.509
certificate-key file
for
membership
authentication
for the cluster or

[Give Feedback](#)

replica set.

Starting with
MongoDB 4.0 on
macOS or
Windows, you can
use the

`net.tls.cluster`
option to specify a
certificate from
the operating
system's secure
certificate store
instead of a PEM
key file.

`net.tls.cluster`
and
`net.tls.cluster`
options are
mutually exclusive.
You can only
specify one.

If
`net.tls.cluster`
does not specify
the `.pem` file for
internal cluster
authentication or
the alternative
`net.tls.cluster`

[Give Feedback](#)

, the cluster uses
the `.pem` file
specified in the

`certificateKeyF`
setting or the
certificate
returned by the
`net.tls.certifi`

If using x.509
authentication,
`--tlsCAFile` or
`tls.CAFile` must
be specified unless
using
`--tlsCertificat`

*Changed in
version 4.4:*

`mongod` / `mongos`
logs a warning on
connection if the
presented x.509
certificate expires
within `30` days of
the
`mongod/mongos`
host system time.

See

[x.509 Certificates
Nearing Expiry
Trigger Warnings](#)
for more
information.

For more
information about

[Give Feedback](#)

TLS and MongoDB,
see
[Configure mongoda](#)

and
TLS/SSL
Configuration for
Clients

•

! IMPORTANT

For
Windows
only,
MongoDB
4.0 and
later do
not
support
encrypted
PEM
files.

The
[mongod](#)
fails to
start if it
encounters
an
encrypted
PEM file.

To
securely
store

[Give Feedback](#)

and
access
a
certificate
for use
with
membership
authentication
on
Windows,
use
`net.tls.c`

net.tls.clusterPas

Type: string

New in version 4.2:

The password to
de-crypt the x.509
certificate-key file
specified with

`--sslClusterFil`

Use the
`net.tls.cluster`
option only if the
certificate-key file
is encrypted. In all
cases, the `mongos`
or `mongod` will
redact the
password from all
logging and
reporting output.

[Give Feedback](#)

Starting in MongoDB 4.0:

- On Linux/BSD, if the private key in the x.509 file is encrypted and you do not specify the `net.tls.clu` option,

MongoDB will prompt for a passphrase.

See

TLS/SSL Certificate Passphrase.

- On macOS, if the private key in the x.509 file is encrypted, you must explicitly specify the `net.tls.clu` option.

Alternatively,
you can

[Give Feedback](#)

you can either use a certificate from the secure system store (see `net.tls.clu`) instead of a cluster PEM file or use an unencrypted PEM file.

- On Windows, MongoDB does not support encrypted certificates. The `mongod` fails if it encounters an encrypted PEM file. Use `net.tls.clu`

For more information about TLS and MongoDB, see `Configure mongoda` and `TLS/SSL ..`.

[Give Feedback](#)

Configuration for Clients

•

`net.tls.CAFile`

Type: string

New in version 4.2:

The .pem file that contains the root certificate chain from the Certificate Authority. Specify the file name of the .pem file using relative or absolute paths.

Windows/macOS Only

If using

`net.tls.cert`

and/or

`net.tls.clus`

, do **not** use

`net.tls.CAFile`

to specify the

root and

intermediate

CA

certificates.

Store all CA

certificates

required to

validate the

full trust chain

of the

`net.tls.cert`

and/or

[Give Feedback](#)

`net.tls.clus`
certificates in
the secure

certificate
store.

For more
information about
TLS and MongoDB,
see
[Configure mongoda](#)
and
[TLS/SSL Configuration for Clients](#)
.

`net.tls.clusterCAF`

Type: string

New in version 4.2:
The .pem file that
contains the root
certificate chain
from the
Certificate
Authority used to
validate the
certificate
presented by a
client establishing
a connection.

Specify the file
name of the .pem
file using relative
or absolute paths.

`net.tls.cluster`

[Give Feedback](#)

requires that
`net.tls.CAFile`
is set.

If
`net.tls.cluster`
does not specify
the .pem file for
validating the
certificate from a
client establishing
a connection, the
cluster uses the
.pem file specified
in the
`net.tls.CAFile`
option.

`net.tls.cluster`
lets you use
separate
Certificate
Authorities to
verify the client to
server and server
to client portions
of the TLS
handshake.

Starting in 4.0, on
macOS or
Windows, you can
use a certificate

[Give Feedback](#)

from the operating system's secure store instead of a PEM key file. See

`net.tls.cluster`

. When using the secure store, you do not need to, but can, also specify the

`net.tls.cluster`

Windows/macOS Only

If using

`net.tls.cert`

and/or

`net.tls.clus`

, do **not** use

`net.tls.clus`

to specify the root and

intermediate

CA

certificates.

Store all CA

certificates

required to

validate the

full trust chain

of the

`net.tls.cert`

and/or

`net.tls.clus`

certificates in

the secure

certificate

store

Give Feedback

store.

For more information about TLS and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#).

net.tls.CRLFile

Type: string

New in version 4.2:
In MongoDB 4.0 and earlier, see [net.ssl.CRLFile](#)

The .pem file that contains the Certificate Revocation List. Specify the file name of the .pem file using relative or absolute paths.

NOTE

- Starting

[Give Feedback](#)

in
MongoDB
4.0,

you
cannot
specify
net.tls.caFile
on
macOS.
Instead,
you
can
use
the
system
SSL
certificate
store,
which
uses
OCSP
(Online
Certificate
Status
Protocol)
to
validate
the
revocation
status
of
certificates.

[Give Feedback](#)

certific
See
`net.s`
in
Mongo
4.0
and
`net.t`
in
Mongo
4.2+
to
use
the
system
SSL
certific
store.

- Startin
in
version
4.4,
to
check
for
certific
revoca
Mongo
enable
the
use
of

Give Feedback

OCSP
(Online Certificate Status Protocol) by default as an alternative to specify a CRL file or using the system SSL certificate store.

For more information about TLS and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#).

•

[Give Feedback](#)

net.tls.allowConne

Type: boolean

New in version 4.2.

For clients that do not present certificates, mongos or mongod bypasses TLS/SSL certificate validation when establishing the connection.

For clients that present a certificate, however, mongos or mongod performs certificate validation using the root certificate chain specified by CAFile and reject clients with invalid certificates.

Use the net.tls.allowCo option if you have a mixed

[Give Feedback](#)

deployment that includes clients that do not or

cannot present certificates to the `mongos` or `mongod`

For more information about TLS and MongoDB, see

[Configure mongoda](#) and [TLS/SSL Configuration for Clients](#)

.

`net.tls.allowInval`

Type: boolean

New in version 4.2.

Enable or disable the validation checks for TLS certificates on other servers in the cluster and allows the use of invalid certificates to connect.

 NOTE

[Give Feedback](#)

If you specify
--tlsAllow or
tls.allow when using x.509 authentication with an invalid certificate is only sufficient to establish a TLS connection but is *insufficient* for authentication.

When using the net.tls.allowIn setting, MongoDB logs a warning regarding the use of the invalid certificate.

[Give Feedback](#)

For more information about TLS and MongoDB, see Configure mongoda and TLS/SSL Configuration for Clients.

net.tls.allowInval

Type: boolean

Default: false

When

net.tls.allowInval is true, MongoDB disables the validation of the hostnames in TLS certificates, allowing mongod to connect to MongoDB instances if the hostname their certificates do not match the specified hostname.

For more

[Give Feedback](#)

information about
TLS and MongoDB,
see

Configure mongoda
and
TLS/SSL
Configuration for
Clients

.

net.tls.disabledProtocols

Type: string

New in version 4.2.

Prevents a
MongoDB server
running with TLS
from accepting
incoming
connections that
use a specific
protocol or
protocols. To
specify multiple
protocols, use a
comma separated
list of protocols.

net.tls.disable
recognizes the
following
protocols: TLS1_0,
TLS1_1, TLS1_2,

[Give Feedback](#)

and starting in
version 4.0.4 (and
3.6.9), `TLS1_3`.

- On macOS,
you cannot
disable
`TLS1_1` and
leave both
`TLS1_0` and
`TLS1_2`
enabled. You
must disable
at least one
of the other
two, for
example,

`TLS1_0,TLS1`

- To list
multiple
protocols,
specify as a
comma
separated list
of protocols.

For example

`TLS1_0,TLS1`

- Specifying an
unrecognized
protocol will
prevent the
server from

[Give Feedback](#)

starting.

- The specified disabled protocols overrides any default disabled protocols.

Starting in version 4.0, MongoDB disables the use of TLS 1.0 if TLS 1.1+ is available on the system. To enable the disabled TLS 1.0, specify `none` to `net.tls.disable`. See [Disable TLS 1.0](#).

Members of replica sets and sharded clusters must speak at least one protocol in common.



TIP

[Give Feedback](#)

See

also:

Disallow
Protocols

`net.tls.FIPSMode`

Type: boolean

New in version 4.2.

Enable or disable the use of the FIPS mode of the TLS library for the mongos or mongod . Your system must have a FIPS compliant library to use the net.tls.FIPSMOD option.

NOTE

FIPS-compatible TLS/SSL is available only in MongoDB Enterprise . See

[Give Feedback](#)

Configure
MongoDB
for FIPS
for more
information.

`net.tls.logVersion`

Type: string

Instructs `mongos`
or `mongod` to log a
message when a
client connects
using a specified
TLS version.

Specify either a
single TLS version
or a comma-
separated list of
multiple TLS
versions.

⚠ EXAMPLE

To
instruct
`mongos`
or
`mongod`
to log a
message
when a

[Give Feedback](#)

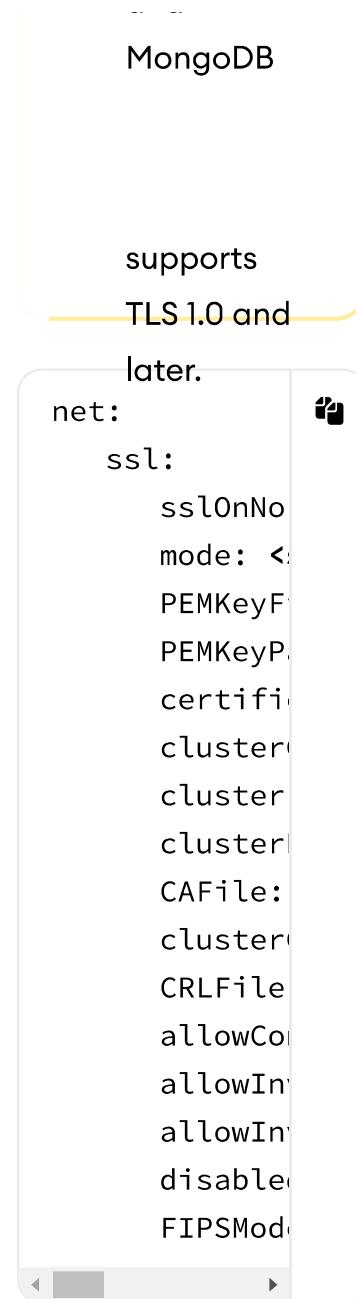
client
connects
using
either
TLS 1.2
or TLS
1.3, set
`net.tls.l`
to
`"TLS1_2,TI`

net.ssl Options

! IMPORTANT

All SSL options are deprecated since 4.2. Use the TLS counterparts instead, as they have identical functionality to the SSL options. The SSL protocol is deprecated and

[Give Feedback](#)



net.ssl.sslOnNormalPort

Type: boolean

Deprecated since version 2.6: Use [net.tls.mode: r](#) instead.

Enable or disable

[Give Feedback](#)

TLS/SSL for mongos or mongod

With

`net.ssl.sslOnNo`

, a `mongos` or

`mongod` requires

TLS/SSL

encryption for all

connections on the

default MongoDB

port, or the port

specified by

`net.port`. By

default,

`net.ssl.sslOnNo`

is disabled.

For more
information about
TLS/SSL and
MongoDB, see
Configure `mongoda`
and
TLS/SSL
Configuration for
Clients

.

`net.ssl.mode`

Type: string

*Deprecated since
version 4.2: Use
`net.tls.mode`*

[Give Feedback](#)

instead.

Enables TLS/SSL or mixed TLS/SSL used for all network connections. The argument to the `net.ssl.mode` setting can be one of the following:

Value	Description
<code>disabled</code>	The server does not use TLS/SSL.
<code>allowSSL</code>	Connections between servers not using TLS/SSL incoming connections, the server accepts both TLS/SSL and no TLS/SSL.

[Give Feedback](#)

Value	Description
-------	-------------

`preferSSL` Connection between servers over TLS/SSL. If an incoming connection, the server accepts both TLS/SSL and no TLS/SSL. It rejects SSL.

`requireSSL` The server uses an accepted connection only over TLS/SSL encryption.

Starting in version

3.4, if

`--tlsCAFile`/`net`

(or their aliases

`--sslCAFile`/`net`

is not specified

and you are not

using x.509

authentication, the

[Give Feedback](#)

system-wide CA
certificate store
will be used when

connecting to an
TLS/SSL-enabled
server.

To use x.509
authentication,
`--tlsCAFile` or
`net.tls.CAFile`
must be specified
unless you are
using
`--tlsCertificateKeyFile`
or
`--net.tls.certifi`

For more
information about
TLS/SSL and
MongoDB, see
Configure mongoda
and
TLS/SSL
Configuration for
Clients

.

`net.ssl.PEMKeyFile`

Type: string

*Deprecated since
version 4.2: Use*

[Give Feedback](#)

`net.tls.certifi`
instead.

The .pem file that contains both the TLS/SSL certificate and key.

Starting with MongoDB 4.0 on macOS or Windows, you can use the `net.ssl.certifi` setting to specify a certificate from the operating system's secure certificate store instead of a PEM key file.

`PEMKeyFile` and `net.ssl.certifi` are mutually exclusive. You can only specify one.

- On Linux/BSD, you must specify `net.ssl.PEM`

[Give Feedback](#)

when
TLS/SSL is
enabled.

- On Windows or macOS, you must specify either `net.ssl.PEM` or `net.ssl.cer` when TLS/SSL is enabled.



IMPORTANT

For Windows only, MongoDB 4.0 and later do not support encrypted PEM files. The `mongod` fails

[Give Feedback](#)

to start if it encourages an encryption PEM file. To secure store and access a certificate for use with TLS/SSL on Windows use `net.start`.

For more information about TLS/SSL and MongoDB, see [Configure mongod](#) and [TLS/SSL](#).

[Give Feedback](#)

Configuration for Clients

•

`net.ssl.PEMKeyPass`

Type: string

Deprecated since version 4.2: Use `net.tls.certifi` instead.

The password to de-crypt the certificate-key file (i.e. `PEMKeyFile`).

Use the

`net.ssl.PEMKeyP` option only if the certificate-key file is encrypted. In all cases, the `mongos` or `mongod` will redact the password from all logging and reporting output.

Starting in MongoDB 4.0:

- On Linux/BSD, if the private

[Give Feedback](#)

key in the PEM file is encrypted and you do not specify the `net.ssl.PEM` option, MongoDB will prompt for a passphrase.

See

TLS/SSL Certificate Passphrase.

- On macOS, if the private key in the PEM file is encrypted, you must explicitly specify the `net.ssl.PEM` option.

Alternatively, you can use a certificate from the secure system store (see

`net.ssl.cer` instead of a

[Give Feedback](#)

/ instead of a

PEM key file

or use an
unencrypted
PEM file.

- On Windows,

MongoDB
does not
support
encrypted
certificates.

The `mongod`
fails if it
encounters
an encrypted
PEM file. Use
`net.ssl.cer`
instead.

For more
information about
TLS/SSL and
MongoDB, see
Configure `mongoda`
and
TLS/SSL
Configuration for
Clients

.

`net.ssl.certificat`

Type: string

Give Feedback

Deprecated since version 4.2: Use
`net.tls.certifi`

instead.

New in version 4.0:

Available on Windows and macOS as an alternative to

`net.ssl.PEMKeyF`

Specifies a certificate property in order to select a matching certificate from the operating system's certificate store to use for TLS/SSL.

`net.ssl.PEMKeyF`

and

`net.ssl.certifi`

options are

mutually exclusive.

You can only specify one.

`net.ssl.certifi`

accepts an

argument of the

format

Give Feedback

format`<property>=<val`

where the property
can be one of the
following:

Property	Value type
<code>subject</code>	ASCII string
<code>thumbprint</code>	hex string

[Give Feedback](#)

When using the system SSL certificate store,

OCSP (Online Certificate Status Protocol) is used to validate the revocation status of certificates.

The `mongod` searches the operating system's secure certificate store for the CA certificates required to validate the full certificate chain of the specified TLS/SSL certificate.

Specifically, the secure certificate store must contain the root CA and any intermediate CA certificates required to build the full certificate chain to the TLS/SSL

[Give Feedback](#)

certificate. Do **not**

use

`net.ssl.CAFile`

or

`net.ssl.cluster`

to specify the root

and intermediate

CA certificate

For example, if the

TLS/SSL certificate

was signed with a

single root CA

certificate, the

secure certificate

store must contain

that root CA

certificate. If the

TLS/SSL certificate

was signed with an

intermediate CA

certificate, the

secure certificate

store must contain

the intermediate CA

certificate *and* the

root CA

certificate.

`net.ssl.clusterCert`

Type: string

Give Feedback

Deprecated since version 4.2: Use `net.tls.cluster` instead.

*New in version 4.0:
Available on Windows and macOS as an alternative to `net.ssl.cluster`*

Specifies a certificate property to select a matching certificate from the operating system's secure certificate store to use for internal x.509 membership authentication.

`net.ssl.cluster` and `net.ssl.ca` options are mutually exclusive. You can only specify one.

`net.ssl.cluster` accepts an argument of the format

[Give Feedback](#)

<property>=<val

where the property
can be one of the
following:

Property	Value type
----------	------------

subject ASCII
string

thumbprint hex
string



The mongod
searches the
operating system's
secure certificate

Give Feedback

store for the CA certificates required to validate the full certificate chain of the specified cluster certificate. Specifically, the secure certificate store must contain the root CA and any intermediate CA certificates required to build the full certificate chain to the cluster certificate.

Do **not** use

`net.ssl.CAFile`

or

`net.ssl.cluster`

to specify the root and intermediate CA certificate.

For example, if the cluster certificate was signed with a single root CA certificate, the secure certificate store must contain that root CA

[Give Feedback](#)

certificate. If the cluster certificate was signed with an

intermediate CA certificate, the secure certificate store must contain the intermedia CA certificate *and* the root CA certificate.

net.ssl.clusterFile

Type: string

Deprecated since version 4.2: Use **net.tls.cluster** instead.

The .pem file that contains the x.509 certificate-key file for membership authentication for the cluster or replica set.

Starting with MongoDB 4.0 on macOS or Windows, you can use the

[Give Feedback](#)

`net.ssl.cluster`

option to specify a
certificate from

the operating
system's secure
certificate store
instead of a PEM
key file.

`net.ssl.cluster`

and

`net.ssl.cluster`

options are
mutually exclusive.

You can only
specify one.

If

`net.ssl.cluster`

does not specify
the `.pem` file for
internal cluster
authentication or
the alternative

`net.ssl.cluster`

, the cluster uses
the `.pem` file
specified in the

`PEMKeyFile`

setting or the
certificate
returned by the
`net.ssl.certifi`

Give Feedback

To use x.509 authentication,
--tlsCAFile or
net.tls.CAFile must be specified unless you are using
--tlsCertificate or
--net.tls.certificate

For more information about TLS/SSL and MongoDB, see Configure mongod and TLS/SSL Configuration for Clients.

! **IMPORTANT**

For Windows only, MongoDB 4.0 and later do not support encrypted

[Give Feedback](#)

PEM files. The `mongod` fails to start if it encounters an encrypted PEM file.

To securely store and access a certificate for use with membership authentication on Windows, use `net.ssl.caCertFile`.

`net.ssl.clusterPas`

Type: string

Deprecated since version 4.2: Use

[Give Feedback](#)

`net.tls.cluster`
instead.

The password to
de-crypt the x.509
certificate-key file
specified with

`--sslClusterFil`

Use the

`net.ssl.cluster`

option only if the
certificate-key file
is encrypted. In all
cases, the `mongos`
or `mongod` will
redact the
password from all
logging and
reporting output.

Starting in
MongoDB 4.0:

- On
Linux/BSD, if
the private
key in the
x.509 file is
encrypted
and you do
not specify
the

[Give Feedback](#)

`net.ssl.clu`

option,

MongoDB will

prompt for a

passphrase.

See

TLS/SSL
Certificate
Passphrase.

- On macOS, if the private key in the x.509 file is encrypted, you must explicitly specify the `net.ssl.clu` option.

Alternatively,

you can

either use a

certificate

from the

secure

system store

(see

`net.ssl.clu`

) instead of a

cluster PEM

file or use an

unencrypted

PEM file.

Give Feedback

- On Windows,
MongoDB
does not
support
encrypted
certificates.
The `mongod`
fails if it
encounters
an encrypted
PEM file. Use
`net.ssl.clu`

For more
information about
TLS/SSL and
MongoDB, see
Configure mongoda
and
TLS/SSL
Configuration for
Clients

.

`net.ssl.CAFile`

Type: string

*Deprecated since
version 4.2: Use
`net.tls.CAFile`
instead.*

The .pem file that
contains the root
certificate chain
from the
Certificate

[Give Feedback](#)

Certificate

Authority. Specify
the file name of

the .pem file using
relative or
absolute paths.

**Windows/macOS
Only**

If using

`net.ssl.cert`

and/or

`net.ssl.clus`

, do **not** use

`net.ssl.CAFI`

to specify the
root and
intermediate
CA

certificates.

Store all CA
certificates
required to
validate the
full trust chain
of the
`net.ssl.cert`
and/or
`net.ssl.clus`
certificates in
the secure
certificate
store.

For more
information about
TLS/SSL and

[Give Feedback](#)

MongoDB, see
Configure mongoda

and

TLS/SSL
Configuration for
Clients

.

net.ssl.clusterCAF

Type: string

*Deprecated since
version 4.2: Use
net.tls.cluster
instead.*

The .pem file that
contains the root
certificate chain
from the
Certificate
Authority used to
validate the
certificate
presented by a
client establishing
a connection.

Specify the file
name of the .pem
file using relative
or absolute paths.

net.ssl.cluster
requires that

[Give Feedback](#)

`net.ssl.CAFile`

is set.

If

`net.ssl.cluster`

does not specify
the .pem file for
validating the
certificate from a
client establishing
a connection, the
cluster uses the
.pem file specified
in the

`net.ssl.CAFile`

option.

`net.ssl.cluster`

lets you use
separate
Certificate
Authorities to
verify the client to
server and server
to client portions
of the TLS
handshake.

Starting in 4.0, on
macOS or
Windows, you can
use a certificate

Give Feedback

from the operating system's secure store instead of a PEM key file. See

`net.ssl.cluster`

. When using the secure store, you do not need to, but can, also specify the

`net.ssl.cluster`

Windows/macOS Only

If using

`net.ssl.cert`

and/or

`net.ssl.clus`

, do **not** use

`net.ssl.clus`

to specify the root and

intermediate

CA

certificates.

Store all CA

certificates

required to

validate the

full trust chain

of the

`net.ssl.cert`

and/or

`net.ssl.clus`

certificates in

the secure

certificate

store

Give Feedback

store.

For more information about TLS/SSL and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#).

.

net.ssl.CRLFile

Type: string

Deprecated since version 4.2: Use [net.tls.CRLFile](#) instead.

The .pem file that contains the Certificate Revocation List. Specify the file name of the .pem file using relative or absolute paths.

NOTE

- Starting

[Give Feedback](#)

in
Mongo
4.0,

you
cannot
specify

`net.s...`

on
macOS

Instead

you
can
use
the
system

SSL
certific

store,

which
uses

OCSP

(Online

Certific

Status

Protocol

to
validat

the

revoca

status

of

certific

Give Feedback

certific
See
`net.s`
in
Mongo
4.0
and
`net.t`
in
Mongo
4.2
to
use
the
system
SSL
certific
store.

- Startin
in
version
4.4,
Mongo
enable
,,
by
default
the
use
of
OCSP
(Online

Give Feedback

CertificateStatusProtocol to check for certificate revocation as an alternative to specify a CRL file or using the system SSL certificate store.

For more information about TLS/SSL and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#)

[Give Feedback](#)

net.ssl.allowConne

Type: boolean

Deprecated since version 4.2: Use net.tls.allowCo instead.

For clients that do not present certificates, mongos or mongod bypasses TLS/SSL certificate validation when establishing the connection.

For clients that present a certificate, however, mongos or mongod performs certificate validation using the root certificate chain specified by CAFile and reject clients with invalid certificates.

[Give Feedback](#)

Use the `net.ssl.allowCo` option if you have a mixed deployment that includes clients that do not or cannot present certificates to the `mongos` or `mongod`

For more information about TLS/SSL and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#).

`net.ssl.allowInval`

Type: boolean

Deprecated since version 4.2: Use `net.tls.allowIn` instead.

Enable or disable the validation checks for TLS/SSL certificates on other servers in the

[Give Feedback](#)

other servers in the

cluster and allows

the use of invalid
certificates to
connect.

NOTE

Starting
in
MongoDB
4.2, if
you
specify

`--tlsAllow`

or

`net.tls.a`

when
using
x.509
authentication

an
invalid
certificate

is only

sufficient

to

establish

a TLS

connection

but it is

[Give Feedback](#)

*but it is
insufficient
for
authentication.*

When using the `net.ssl.allowInval` setting, MongoDB logs a warning regarding the use of the invalid certificate.

For more information about TLS/SSL and MongoDB, see [Configure mongoda](#) and [TLS/SSL Configuration for Clients](#).

`net.ssl.allowInval`

Type: boolean

Default: false

Deprecated since version 4.2.

Use `net.tls.allowInval` instead.

[Give Feedback](#)

When

`net.ssl.allowIn`
is `true`, MongoDB
disables the
validation of the
hostnames in
TLS/SSL
certificates,
allowing `mongod`
to connect to
MongoDB
instances if the
hostname their
certificates do not
match the
specified
hostname.

For more
information about
TLS/SSL and
MongoDB, see
[Configure mongoda](#)
and
[TLS/SSL Configuration for Clients](#)

.

`net.ssl.disabledPr`

Type: string

*Deprecated since
version 4.2. Use*

[Give Feedback](#)

Version 4.2.0
`net.tls.disable`
instead.

Prevents a MongoDB server running with TLS/SSL from accepting incoming connections that use a specific protocol or protocols. To specify multiple protocols, use a comma separated list of protocols.

`net.ssl.disable` recognizes the following protocols: `TLS1_0`, `TLS1_1`, `TLS1_2`, and starting in version 4.0.4 (and 3.6.9), `TLS1_3`.

- On macOS, you cannot disable `TLS1_1` and leave both

[Give Feedback](#)

`TLS1_0` and

`TLS1_2`

enabled. You must disable at least one of the other two, for example,

`TLS1_0,TLS1`

- To list multiple protocols, specify as a comma separated list of protocols.

For example

`TLS1_0,TLS1`

- Specifying an unrecognized protocol will prevent the server from starting.

- The specified disabled protocols overrides any default disabled protocols.

[Give Feedback](#)

Starting in version 4.0, MongoDB disables the use of TLS 1.0 if TLS 1.1+ is available on the system. To enable the disabled TLS 1.0, specify `none` to `net.ssl.disable`. See [Disable TLS 1.0](#).

Members of replica sets and sharded clusters must speak at least one protocol in common.

 **TIP**

See

also:

[Disallow Protocols](#)

`net.ssl.FIPSMode`

Type: boolean

Deprecated since version 4.2: Use `net.tls.FIPSMod`

[Give Feedback](#)

instead.

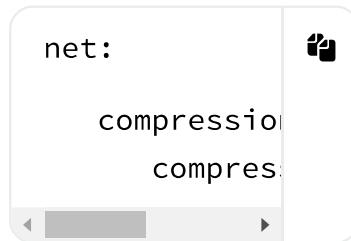
Enable or disable the use of the FIPS mode of the TLS/SSL library for the `mongos` or `mongod`. Your system must have a FIPS compliant library to use the `net.ssl.FIPSMOD` option.

 NOTE

FIPS-compatible TLS/SSL is available only in MongoDB Enterprise. See [Configure MongoDB for FIPS](#) for more information.

`net.compression`

[Give Feedback](#)

net.compression**Option****net.compression.co**

Default:

snappy,zstd,zlib

Specifies the default compressor(s) to use for communication between this mongod or mongos instance and:

- other members of the deployment if the instance is part of a replica set or a sharded cluster
- mongosh
- drivers that support the

[Give Feedback](#)

support the
OP_COMPRESS
message
format.

MongoDB supports
the following
compressors:

- snappy
- zlib
(Available
starting in
MongoDB 3.6)
- zstd
(Available
starting in
MongoDB
4.2)

In versions 3.6 and
4.0, mongod and
mongos enable
network
compression by
default with
snappy as the
compressor.

Starting in version
4.2, mongod and
mongos instances
default to both
snappy, zstd, zlib

Give Feedback

compressors, in
that order.

To disable network
compression, set
the value to
`disabled`.

! IMPORTANT

Messages
are
compressed
when
both
parties
enable
network
compression.
Otherwise,
messages
between
the
parties
are
uncompressed.

If you specify
multiple
compressors, then
the order in which

[Give Feedback](#)

the order in which you list the compressors matter as well as the communication initiator. For example, if mongosh specifies the following network compressors zlib, snappy and the mongod specifies snappy, zlib, messages between mongosh and mongod uses zlib.

If the parties do not share at least one common compressor, messages between the parties are uncompressed. For example, if mongosh specifies the network compressor zlib and mongod

[Give Feedback](#)

specifies snappy,
messages between

mongosh and
mongod are not
compressed.

security Options

```
security:   
  keyFile: <string>  
  clusterAuthMechanism: <string>  
  authorization: <string>  
  transitionTo: <string>  
  javascriptAuthentication: <string>  
  redactClients: <bool>  
  clusterIpSieve: <bool>  
  - <string>  
  sasl:  
    hostName: <string>  
    service: <string>  
    saslauthdb: <string>  
    enableEncryption: <bool>  
    encryptionLevel: <string>  
    encryptionKey: <string>  
    kmip:  
      keyIdentifier: <string>  
      rotateMasterKey: <bool>  
      serverName: <string>  
      port: <int>  
      clientCertificate: <string>  
      clientCertificatePath: <string>  
      clientCertificateKey: <string>  
      serverCertificates: <string>
```

Give Feedback

```
connect
connect
ldap:
servers

bind:
method
sasl
query
query
use0:
transport
timeout
userToDB
authz:
query
validate
```

security.keyFile

Type: string

The path to a key file that stores the shared secret that MongoDB instances use to authenticate to each other in a sharded cluster or replica set.

`keyFile` implies

`security.authorization`

. See

[Internal/Membership Authentication](#)

[Give Feedback](#)

for more information.

Starting in MongoDB 4.2, keyfiles for internal membership authentication use YAML format to allow for multiple keys in a keyfile. The YAML format accepts content of:

- a single key string (same as in earlier versions),
- multiple key strings (each string must be enclosed in quotes), or
- sequence of key strings.

The YAML format is compatible with the existing single-key keyfiles that use the text file format.

[Give Feedback](#)

security.clusterAuthSource

Type: string

Default: keyFile

The authentication mode used for cluster authentication. If you use internal x.509 authentication, specify so here. This option can have one of the following values:

Value	Description
keyFile	Use a keyfile for authentication. A keyfile only contains a keyfile.
sendKeyFile	For rolling upgrades. Send a keyfile for authentication but accept a keyfile for x.509 certificate.

[Give Feedback](#)

Value	Description
-------	-------------

sendX509	For roles that support upgrading from x.509 certificates, send the X.509 certificate and private key to the mongod instance. This option is only applicable for roles that support upgrading from x.509 certificates.
----------	---

x509	Recognized. See the x.509 section for authentication on an accepted CA. x.509 certificates are used for roles that support upgrading from x.509 certificates.
------	---

If `--tlsCAFfile` or `tls.CAFfile` is not specified and you are not using x.509 authentication, the system-wide CA certificate store will be used when connecting to the mongod instance.

[Give Feedback](#)

connecting to an
TLS-enabled
server.

If using x.509
authentication,
`--tlsCAFile` or
`tls.CAFile` must
be specified unless
using
`--tlsCertificate`

For more
information about
TLS and MongoDB,
see
[Configure mongoda](#)
and
[TLS/SSL Configuration for Clients](#)

security.authorization

Type: string

Default: disabled

Enable or disable
Role-Based Access
Control (RBAC) to
govern each user's
access to
database
resources and

[Give Feedback](#)

operations.

Set this option to one of the following:

Value	Description
enabled	A user can access or modify the database resources and perform any actions for which they have been granted privileges
disabled	A user cannot access any database and perform any actions



See

Role-Based Access Control
for more information.

The `security.authorization` setting is available

Give Feedback

only for mongod

security.transition

Type: boolean

Default: false

Allows the mongod or mongos to accept and create authenticated and non-authenticated connections to and from other mongod and mongos instances in the deployment. Used for performing rolling transition of replica sets or sharded clusters from a no-auth configuration to internal authentication. Requires specifying a internal authentication mechanism such as security.keyFile

Give Feedback

For example, it
using keyfiles for
internal
authentication
, the mongod or
mongos creates an
authenticated
connection with
any mongod or
mongos in the
deployment using
a matching keyfile.

If the security
mechanisms do
not match, the
mongod or mongos
utilizes a non-
authenticated
connection
instead.

A mongod or
mongos running
with
security.transi
does not enforce
user access
controls
. Users may
connect to your
deployment
without any
access control
checks and
perform read.

Give Feedback

write, and administrative operations.

NOTE

A mongod or mongos running with internal authentication and *without* security. It requires clients to connect using user access controls. Update clients to connect to the mongod or mongos using the

Give Feedback

use appropriate user prior to restarting `mongod` or `mongos` without `security.javascript`.

`security.javascript`

Type: boolean

Default: true

Enables or disables

server-side JavaScript execution.
When disabled, you cannot use operations that perform server-side execution of JavaScript code, such as the `$where` query operator, `mapReduce` command, `$accumulator`, and `$function`.

[Give Feedback](#)

[Add to Collection](#)

If you do not use these operations, disable server-side scripting.

Starting in version 4.4, the

`security.redactcli` setting is available for both `mongod` and `mongos`. In earlier versions, the setting is only available for `mongod`

`security.redactcli`

Type: boolean

Available in MongoDB Enterprise only.

A `mongod` or `mongos` running with `security.redact` redacts any message

[Give Feedback](#)

accompanying a
given log event
before logging.

This prevents the

`mongod` or `mongos`
from writing
potentially
sensitive data
stored on the
database to the
diagnostic log.

Metadata such as
error or operation
codes, line
numbers, and
source file names
are still visible in
the logs.

Use

`security.redact`
in conjunction with
Encryption at Rest
and
TLS/SSL (Transport
Encryption)
to assist
compliance with
regulatory
requirements.

For example, a
MongoDB
deployment might

[Give Feedback](#)

store Personally Identifiable Information (PII) in one or more collections. The mongod or mongos logs events such as those related to CRUD operations, sharding metadata, etc. It is possible that the mongod or mongos may expose PII as a part of these logging operations. A mongod or mongos running with security.redact removes any message accompanying these events before being output to the log, effectively removing the PII.

Diagnostics on a mongod or mongos running with security.redact

[Give Feedback](#)

may be more difficult due to the lack of data

related to a log event. See the process logging manual page for an example of the effect of `security.redact` on log output.

On a running `mongod` or `mongos`, use `setParameter` with the `redactClientLog` parameter to configure this setting.

`security.clusterIp`

Type: list

New in version 5.0.

A list of IP addresses/CIDR (Classless Inter-Domain Routing) ranges against which the `mongod`

[Give Feedback](#)

validates authentication requests from other members of the replica set and, if part of a sharded cluster, the mongos instances. The mongod verifies that the originating IP is either explicitly in the list or belongs to a CIDR range in the list. If the IP address is not present, the server does not authenticate the mongod or mongos

`security.cluste` has no effect on a mongod started without authentication.

`security.cluste` requires specifying each IPv4/6 address or Classless Inter-

Give Feedback

Domain Routing (CIDR[↗]) range as a YAML list:

```
security:  
  clusterIpRanges:  
    - 192.0.0.0/16  
    - 127.0.0.1/32  
    - ::1
```

 **IMPORTANT**

Ensure `security.clusterIpRanges` includes the IP address or CIDR ranges that include the IP address of each replica set member or `mongos` in the deployment to

[Give Feedback](#)

ensure
healthy
communication

between
cluster
components

security.clusterIp

Type: list

*Deprecated in
version 5.0: Use
security.clusterIp
instead.*

A list of IP
addresses/CIDR (Classless Inter-Domain Routing) ranges against which the mongod validates authentication requests from other members of the replica set and, if part of a sharded cluster, the mongos instances. The mongod verifies

[Give Feedback](#)

that the originating IP is either explicitly in

the list or belongs to a CIDR range in the list. If the IP address is not present, the server does not authenticate the `mongod` or `mongos`

`security.clusterIPs` has no effect on a `mongod` started without authentication.

`security.clusterIPs` requires specifying each IPv4/6 address or Classless Inter-Domain Routing (CIDR²) range as a YAML list:

```
security:  
  clusterIPs:  
    - 192.0.0.1  
    - 127.0.0.1  
    - ::1
```

[Give Feedback](#)

! IMPORTANT

Ensure `security.config` includes the IP address or CIDR ranges that include the IP address of each replica set member or `mongos` in the deployment to ensure healthy communication between cluster components.

Configuration Options

```
security:  
  enableEncryption  
  encryptionKeyFile  
  encryptionKeyFileTLS  
  kmip:  
    keyIdentifier  
    rotateMasterKey  
    serverName  
  port: <port>  
  clientCertificates  
  clientCertificatePaths  
  clientCertificatePem  
  serverCertificates  
  connectTimeoutMS  
  connectWithTimeouts  
  activateTLS  
  keyStat
```

security.enableEnc

Type: boolean

Default: false

Enables encryption for the WiredTiger storage engine.

You must set to

true to pass in encryption keys and

[Give Feedback](#)

configurations.

NOTE

**Enterprise
Feature**

Available
in
MongoDB
Enterprise
only.

security.encrypted

Type: string

Default:

AES256-CBC

The cipher mode
to use for
encryption at rest:

Mode	Description
AES256-CBC	256-bit Advanced Encryption Standard Cipher Chain Mode

[Give Feedback](#)

Mode	Description
AES256-GCM	256-bit Advanced Encryption Standard, Galois/ Counter Mode Available only on Linux. <i>Changes version MongoDB Enterprise on Windows no longer supports AES256-GCM. This cipher is now available only on Linux.</i>

**NOTE**

**Enterprise
Feature**

[Give Feedback](#)

Available
in
MongoDB

Enterprise
only.

`security.encryptio`

Type: string

The path to the local keyfile when managing keys via process *other than* KMIP. Only set when managing keys via process other than KMIP. If data is already encrypted using KMIP, MongoDB will throw an error.

Requires
`security.enable`
to be `true`.

ⓘ NOTE

Enterprise

Feature

Available
in

Give Feedback

III
MongoDB
Enterprise
only.

security.kmip.keyId

Type: string

Unique KMIP identifier for an existing key within the KMIP server.

Include to use the key associated with the identifier as the system key.

You can only use the setting the first time you enable encryption for the mongod instance.

Requires security.enable to be true.

If unspecified, MongoDB will request that the KMIP server create a new key to utilize as the system key.

[Give Feedback](#)

100% READ

If the KMIP server cannot locate a key with the specified identifier or the data is already encrypted with a key, MongoDB will throw an error.

 **NOTE**

Enterprise Feature
Available in MongoDB Enterprise only.

`security.kmip.rotateMasterKey`

Type: boolean

Default: false

If true, rotate the master key and re-encrypt the internal keystore.

 **NOTE**

[Give Feedback](#)

Enterprise**Feature**

Available

in

MongoDB

Enterprise

only.

 **TIP****See****also:**

KMIP
Master
Key
Rotation

`security.kmip.server`

Type: string

Hostname or IP

address of the
KMIP server to
connect to.

Requires

`security.enable`

to be true.

Starting in

MongoDB 4.2.1

(and 4.0.14), you

[Give Feedback](#)

can specify multiple KMIP servers as a comma-separated list, e.g.

```
server1.example
```

On startup, the mongod will attempt to establish a connection to each server in the order listed, and will select the first server to which it can successfully establish a connection. KMIP server selection occurs only at startup.

When connecting to a KMIP server, the mongod verifies that the specified security.kmip.s matches the Subject Alternative Name SAN (or, if SAN is not present,

[Give Feedback](#)

the Common Name `CN`) in the certificate

presented by the KMIP server. If `SAN` is present, `mongod` does not match against the `CN`. If the hostname does not match the `SAN` (or `CN`), the `mongod` will fail to connect.

Starting in MongoDB 4.2, when performing comparison of SAN, MongoDB supports comparison of DNS names or IP addresses. In previous versions, MongoDB only supports comparisons of DNS names.

 NOTE

Enterprise

Give Feedback

Feature

Available
in

MongoDB
Enterprise
only.

security.kmip.port

Type: string

Default: 5696

Port number to use
to communicate
with the KMIP
server. Requires
`security.kmip.s`
. Requires
`security.enable`
to be true.

If specifying
multiple KMIP
servers with
`security.kmip.s`
, the `mongod` will
use the port
specified with
`security.kmip.p`
for all provided
KMIP servers.

[Give Feedback](#)

NOTE**Enterprise****Feature**

Available

in

MongoDB

Enterprise

only.

`security.kmip.client`*Type:* string

String containing
the path to the
client certificate
used for
authenticating
MongoDB to the
KMIP server.

Requires that a

`security.kmip.server`

be provided.

NOTE

Starting
in 4.0,
on
macOS
or

Give Feedback

Windows, you can use a certificate from the operating system's secure store instead of a PEM key file. See `security.kmip.client`.

 **NOTE**

Enterprise Feature Available in MongoDB Enterprise only.

`security.kmip.client`

Type: string

The password to

[Give Feedback](#)

decrypt the client certificate (i.e. `security.kmip.c`), used to authenticate MongoDB to the KMIP server. Use the option only if the certificate is encrypted.

 **NOTE**

Enterprise Feature
Available in MongoDB Enterprise only.

`security.kmip.cle`

Type: string

New in version 4.0: (and 4.2.15, 4.4.7, and 5.0)

Available on Windows and macOS as an alternative to

[Give Feedback](#)

`security.kmip.c`

`security.kmip.c`

and

`security.kmip.c`

options are

mutually exclusive.

You can only

specify one.

Specifies a certificate property in order to select a matching certificate from the operating system's certificate store to authenticate MongoDB to the KMIP server.

`security.kmip.c`

accepts an

argument of the

format

`<property>=<val`

where the property

can be one of the

following:

Give Feedback

Property	Value type
----------	------------

subject	ASCII string
---------	--------------

thumbprint	hex string
------------	------------



 **NOTE**

Enterprise

[Give Feedback](#)

Feature

Available
in
MongoDB
Enterprise
only.

security.kmip.serv

Type: string

Path to CA File.
Used for validating
secure client
connection to
KMIP server.

NOTE

Starting
in 4.0,
on
macOS
or
Windows,
you can
use a
certificate
from the
operating
system's
secure

[Give Feedback](#)

store instead of a PEM key file. See `security.kms_PROVIDERS`. When using the secure store, you do not need to, but can, also specify the `security.kms_PROVIDERS`.

 **NOTE**

Enterprise Feature Available in MongoDB Enterprise only.

[Give Feedback](#)

security.kmip.connectRetries

Type: int

Default: 0

New in version 4.4.

How many times to retry the initial connection to the KMIP server. Use together with `connectTimeoutMS` to control how long the `mongod` waits for a response between each retry.

NOTE

Enterprise Feature
Available in MongoDB Enterprise only.

security.kmip.connectRetries

Type: int

Give Feedback

Default: 5000

New in version 4.4.

Timeout in milliseconds to wait for a response from the KMIP server. If the `connectRetries` setting is specified, the `mongod` will wait up to the value specified with `connectTimeoutMS` for each retry.

Value must be `1000` or greater.

 **NOTE**

Enterprise Feature
Available in MongoDB Enterprise only.

`security.kmip.acti`

[Give Feedback](#)

Type: boolean

Default: true

New in version 5.3.

Activates all newly created KMIP keys upon creation and then periodically checks those keys are in an active state.

When

`security.kmip.activateKey` is `true` and you have existing keys on a KMIP server, the key must be activated first or the `mongod` node will fail to start.

If the key being used by the `mongod` transitions into a non-active state, the `mongod` node will shut down unless `kmipActivateKey` is false. To ensure

[Give Feedback](#)

you have an active key, rotate the

KMIP master key by using
`security.kmip.r`

security.kmip.keys

Type: int

Default: 900 seconds

New in version 5.3.

Frequency in seconds at which mongod polls the KMIP server for active keys.

To disable disable polling, set the value to `-1`.

security.sasl

Options

security:
sasl:
hostName:
service:
saslauthd:

Give Feedback

security.sasl.host

Type: string

A fully qualified server domain name for the purpose of configuring SASL and Kerberos authentication.

The SASL hostname overrides the hostname only for the configuration of SASL and Kerberos.

security.sasl.serv

Type: string

Registered name of the service using SASL. This option allows you to override the default Kerberos service name component of the Kerberos principal name, on a per-instance basis. If

[Give Feedback](#)

instance basis. If

unspecified, the default value is `mongodb`.

MongoDB permits setting this option only at startup.

The

`setParameter` can not change this setting.

This option is available only in MongoDB Enterprise.

 **IMPORTANT**

Ensure that your driver supports alternate service names.

For `mongosh` and other MongoDB tools to

[Give Feedback](#)

connect
to the
new
`serviceNam`, see the
`gssapiServ`
option.

security.sasl.sasl

Type: string

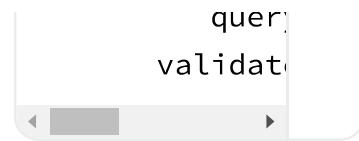
The path to the
UNIX domain
socket file for
`saslauthd`.

security.ldap

Options

`security:` 
`ldap:`
`servers`
`bind:`
`method`
`sasl`
`query`
`query`
`use0`
`transpo`
`timeout`
`userToD`
`authz:`

[Give Feedback](#)



security.ldap.serv

Type: string

*Available in
MongoDB
Enterprise only.*

The LDAP server against which the mongod or mongos authenticates users or determines what actions a user is authorized to perform on a given database. If the LDAP server specified has any replicated instances, you may specify the host and port of each replicated server in a comma-delimited list.

If your LDAP infrastructure

[Give Feedback](#)

partitions the
LDAP directory
over multiple LDAP

servers, specify
one LDAP server or
any of its
replicated
instances to
`security.ldap.s`
. MongoDB
supports following
LDAP referrals as
defined in
RFC 4511 4.1.10[↗].
Do not use
`security.ldap.s`
for listing every
LDAP server in
your infrastructure.

This setting can be
configured on a
running `mongod` or
`mongos` using
`setParameter`

If unset, `mongod` or
`mongos` cannot
use
LDAP
authentication or
authorization.

`security.ldap.bind`

Give Feedback

Type: string

*Available in
MongoDB
Enterprise only.*

The identity with
which mongod or
mongos binds as,
when connecting
to or performing
queries on an
LDAP server.

Only required if
any of the
following are true:

- Using
LDAP
authorization.
- Using an
LDAP query
for
security.ld
- The LDAP
server
disallows
anonymous
binds

You must use
queryUser with

[Give Feedback](#)

```
queryUserWith  
queryPassword
```

If unset, mongod or mongos will not attempt to bind to the LDAP server.

This setting can be configured on a running mongod or mongos using setParameter

NOTE

Windows MongoDB deployment can use useOSDefault instead of queryUser and queryPassword. You cannot specify both queryUser and

[Give Feedback](#)

useOSDefault

at the
same

security.tlsLDAP.bind

Type: string or
array

Available in
MongoDB
Enterprise only.

The password used
to bind to an LDAP
server when using
`queryUser`. You
must use
`queryPassword`
with `queryUser`

If not set, `mongod`
or `mongos` does
not attempt to
bind to the LDAP
server.

You can configure
this setting on a
running `mongod` or
`mongos` using
`setParameter`

Give Feedback

Starting in
MongoDB 4.4, the
`ldapQueryPasswo`

`setParameter`
command accepts
either a string or
an array of strings.

If

`ldapQueryPasswo`
is set to an array,
MongoDB tries
each password in
order until one
succeeds. Use a
password array to
roll over the LDAP
account password
without downtime.

 NOTE

Windows
MongoDB
deployment
can use
`useOSDefau`
instead
of
`queryUser`
and
`queryPassw`
you

[Give Feedback](#)

• You cannot specify both `queryPassword` and `useOSDefaultBind` at the same time.

security.ldap.bind

Type: boolean

Default: false

Available in MongoDB Enterprise for the Windows platform only.

Allows `mongod` or `mongos` to authenticate, or bind, using your Windows login credentials when connecting to the LDAP server.

Only required if:

[Give Feedback](#)

- Using
LDAP
authorization.

- Using an
LDAP query
for
`username tr`

- The LDAP
server
disallows
anonymous
binds

Use

`useOSDefaults`

to replace

`queryUser` and

`queryPassword`

`security.ldap.bind`

Type: string

Default: simple

Available in
MongoDB
Enterprise only.

The method
`mongod` or `mongos`
uses to
authenticate to an
LDAP server. Use

[Give Feedback](#)

with `queryUser`
and

`queryPassword`
to connect to the
LDAP server.

`method` supports
the following
values:

- `simple` -
`mongod` or
`mongos` uses
simple
authentication
- `sasl` -
`mongod` or
`mongos` uses
SASL
protocol for
authentication

If you specify
`sasl`, you can
configure the
available SASL
mechanisms using
`security.ldap.b`
. `mongod` or
`mongos` defaults to
using `DIGEST-MD5`
mechanism.

[Give Feedback](#)

security.ldap.bind

Type: string

Default: DIGEST-MD5

Available in MongoDB Enterprise only.

A comma-separated list of SASL mechanisms mongod or mongos can use when authenticating to the LDAP server.

The mongod or mongos and the LDAP server must agree on at least one mechanism.

The mongod or mongos dynamically loads any SASL mechanism libraries installed on the host machine at runtime.

Install and

[Give Feedback](#)

configure the appropriate libraries for the selected SASL mechanism(s) on both the `mongod` or `mongos` host and the remote LDAP server host. Your operating system may include certain SASL libraries by default. Defer to the documentation associated with each SASL mechanism for guidance on installation and configuration.

If using the `GSSAPI` SASL mechanism for use with Kerberos Authentication, verify the following for the `mongod` or `mongos` host machine:

[Give Feedback](#)

Linux

- The `KRB5_CLI` environment variable resolves to the name of the client `Linux Keytab Files` for the host machine. For more on Kerberos environment variables, please defer to the Kerberos document.
- The client keytab includes a `User Principal` for the

[Give Feedback](#)

`mongod`

or

`mongos`

to use
when
connectin
to the
LDAP
server
and
execute
LDAP
queries.

Windows

If connecting
to an Active
Directory
server, the
Windows
Kerberos
configuration
automatically
generates a
Ticket-
Granting- Ticket
when the user
logs onto the
system. Set
`useOSDefault`
to `true` to
allow `mongod`
or `mongos` to
use the
generated
credentials

[Give Feedback](#)

when
connecting to
the Active
Directory

server and
execute
queries.

Set `method` to
`sasl` to use this
option.

ⓘ NOTE

For a complete list of SASL mechanisms see the IANA listing. Defer to the documentation for your LDAP or Active Directory service for identifying the SASL

[Give Feedback](#)

mechanisms
compatible
with the
service.

MongoDB
is not a
source
of SASL
mechanism
libraries,
nor is
the
MongoDB
documentat
a
definitive
source
for
installing
or
configuring
any
given
SASL
mechanism.

For
documentat
and
support,
defer to
the
SASL
mechanism

[Give Feedback](#)

mechanism
library

vendor
or
owner.

For
more
information
on SASL,
defer to
the
following
resources:

- For Linux, please see the Cyrus SASL documentation.
- For Windows, please see the Windows SASL documentation.

security.ldap.trans

Give Feedback

Type: string

Default: tls

*Available in
MongoDB
Enterprise only.*

By default, mongod or mongos creates a TLS/SSL secured connection to the LDAP server.

For Linux deployments, you must configure the appropriate TLS Options in /etc/openldap/l file. Your operating system's package manager creates this file as part of the MongoDB Enterprise installation, via the libldap dependency. See the documentation for TLS Options in the

[Give Feedback](#)

ldap.cont
OpenLDAP documentation for more complete instructions.

For Windows deployment, you must add the LDAP server CA certificates to the Windows certificate management tool. The exact name and functionality of the tool may vary depending on operating system version. Please see the documentation for your version of Windows for more information on certificate management.

Set `transportSecurity` to `none` to disable TLS/SSL between `mongod` or `mongos` and the LDAP server.

[Give Feedback](#)

⚠ WARNING

Setting `transport` to `none` transmits plaintext information and possibly credentials between `mongod` or `mongos` and the LDAP server.

`security.ldap.timeout`

Type: int

Default: 10000

Available in
MongoDB
Enterprise only.

The amount of
time in

...

[Give Feedback](#)

milliseconds
mongod or mongos
should wait for an

LDAP server to
respond to a
request.

Increasing the
value of
`timeoutMS` may
prevent
connection failure
between the
MongoDB server
and the LDAP
server, if the
source of the
failure is a
connection
timeout.

Decreasing the
value of
`timeoutMS`
reduces the time
MongoDB waits for
a response from
the LDAP server.

This setting can be
configured on a
running `mongod` or
`mongos` using

[Give Feedback](#)

setParameter

security.ldap.user'

Type: string

Available in
MongoDB
Enterprise only.

Maps the
username
provided to
mongod or mongos
for authentication
to a LDAP
Distinguished
Name (DN). You
may need to use
userToDNMapping
to transform a
username into an
LDAP DN in the
following
scenarios:

- Performing
LDAP
authentication
with simple
LDAP
binding,
where users
authenticate

Give Feedback

to MongoDB
with
usernames

that are not
full LDAP
DNs.

- Using an `LDAP author` that requires a DN.
- Transforming the usernames of clients authenticating to Mongo DB using different authentication mechanisms (e.g. x.509, kerberos) to a full LDAP DN for authorization.

`userToDNMapping` expects a quote-enclosed JSON-string representing an ordered array of documents

[Give Feedback](#)

or documents.

Each document
contains a regular
expression `match`

and either a
`substitution` or
`ldapQuery`
template used for
transforming the
incoming
username.

Each document in
the array has the
following form:

```
{  
  match: "..."  
  substitution: "..."  
}  
...
```

Field	Description
<code>match</code>	A regular expression used to match the incoming username.

Give Feedback

Field	Des																						
<code>match</code>	An ECM form regular expression (regex) matching a principal user. Each pattern is enclosed in <code>\$(...)\$</code> . The regular expression is grouped by <code>(...)</code> , and the entire group is substituted by <code>\$1</code> .																						
<code>substitution</code>	An L-shaped substitution table. The first column lists the names of the fields to be substituted. The second column lists the values to be substituted. The table is defined as follows: <table border="1"><tr><td>distination</td><td>An L-shaped substitution table.</td></tr><tr><td>name</td><td>form</td></tr><tr><td>template</td><td>tem</td></tr><tr><td>connection</td><td>con</td></tr><tr><td>authSource</td><td>auth</td></tr><tr><td>name</td><td>nam</td></tr><tr><td>by token</td><td>by tl</td></tr><tr><td>regular expression</td><td>regEx</td></tr><tr><td>LDA</td><td>LDA</td></tr><tr><td>curl</td><td>curl</td></tr><tr><td>encl</td><td>encl</td></tr></table>	distination	An L-shaped substitution table.	name	form	template	tem	connection	con	authSource	auth	name	nam	by token	by tl	regular expression	regEx	LDA	LDA	curl	curl	encl	encl
distination	An L-shaped substitution table.																						
name	form																						
template	tem																						
connection	con																						
authSource	auth																						
name	nam																						
by token	by tl																						
regular expression	regEx																						
LDA	LDA																						
curl	curl																						
encl	encl																						

[Give Feedback](#)

Field	Description
num	is re
the	

corr

regEx
cap
group

extra

the

auth

user

the (

regEx

The

the :

must

RFC

escape

ldapQuery

A LDAP query form template. It consists of an insertion point, authentication name, password, and a regular expression for the LDAPI encoder response. It also includes an RFC 2256 URL, an RFC 4514 URL, and an encoding for curly braces. It also includes an envelope number and a regular expression for the response.

Give Feedback

Field	the Des corr
	rege cap grou

extra

the

auth

user

the [

expri

mon

mon

exec

quei

the l

serv

retri

LDA

the

auth

user

or m

requ

exact

retu

for t

tran

to b

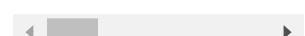
succ

mon

mon

this

tran



Give Feedback

 NOTE

An explanation of RFC4514[↗], RFC4515[↗], RFC4516[↗], or LDAP queries is out of scope for the MongoDB Documentation. Please review the RFC directly or use your preferred LDAP resource.

For each document in the array, you must

[Give Feedback](#)

use either
substitution or
ldapQuery. You

*cannot specify
both in the same
document.*

When performing
authentication or
authorization,

`mongod` or `mongos`
steps through
each document in
the array in the
given order,
checking the
authentication
username against
the `match` filter. If
a match is found,
`mongod` or `mongos`
applies the
transformation
and uses the
output for
authenticating the
user. `mongod` or
`mongos` does not
check the
remaining
documents in the
array.

[Give Feedback](#)

If the given document does not match the provided authentication name, mongod or mongos continues through the list of documents to find additional matches. If no matches are found in any document, or the transformation the document describes fails, mongod or mongos returns an error.

Starting in MongoDB 4.4, mongod or mongos also returns an error if one of the transformations cannot be evaluated due to networking or authentication failures to the LDAP server.

[Give Feedback](#)

mongod or mongos
rejects the
connection

request and does
not check the
remaining
documents in the
array.

Starting in
MongoDB 5.0,
`userToDNMapping`
accepts an empty
string "" or empty
array [] in place
of a mapping
document. If
providing an
empty string or
empty array to
`userToDNMapping`
, MongoDB will
map the
authenticated
username as the
LDAP DN.
Previously,
providing an
empty mapping
document would
cause mapping to
fail.

Give Feedback

EXAMPLE

The
following
shows
two
transformat
documents.

The first
document
matches
against
any
string
ending
in
`@ENGINEER`:
placing
anything
preceding
the
suffix
into a
regex
capture
group.

The
second
document
matches
against

[Give Feedback](#)

against
any
string
ending
in @DBA,
placing
anything
preceding
the
suffix
into a
regex
capture
group.



IMPORTANT

You
must
pass
the
array
to
userl
as
a
string

"[
{
ma
su

[Give Feedback](#)

```
}
```

```
{
```

```
ma
```

```
ld
```

```
}
```

```
]"
```

A user
with
username
`alice@ENG`

matches
the first
document.

The
regex
capture
group
`{0}`

corresponds
to the
string
`alice.`

The
resulting
output
is the
DN

`"cn=alice`

Give Feedback

A user
with
username
`bob@DBA.EU`

matches
the
second
document.

The
regex
capture
group
`{0}`

corresponds
to the
string
`bob`.

The
resulting
output
is the
LDAP
query

`"ou=dba,dc=mongod,dc=org"`

or
`mongos`
executes
this
query
against
the
LDAP

[Give Feedback](#)

LDAP

server,

returning

the

result

"cn=bob,ou

If

`userToDNMapping`

is unset, `mongod` or

`mongos` applies no

transformations to

the username

when attempting

to authenticate or

authorize a user

against the LDAP

server.

This setting can be

configured on a

running `mongod` or

`mongos` using the

`setParameter`

database

command.

security.ldap.auth

Type: string

Available in

Give Feedback

MongoDB
Enterprise only.

A relative LDAP query URL formatted conforming to RFC4515² and RFC4516³ that mongod executes to obtain the LDAP groups to which the authenticated user belongs to.

The query is relative to the host or hosts specified in

`security.ldap.s`

In the URL, you can use the following substitution tokens:

Substitution Token

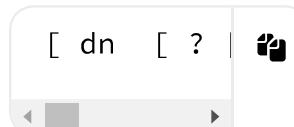
{USER}

Give Feedback

Substitution Token

{PROVIDED_USER}

When constructing
the query URL,
ensure that the
order of LDAP
parameters
respects RFC4516:



If your query
includes an
attribute, `mongod`
assumes that the
query retrieves a
list of the DNs
which this entity is
a member of.

[Give Feedback](#)

If your query does not include an attribute, mongod assumes the query retrieves all entities which the user is member of.

For each LDAP DN returned by the query, mongod assigns the authorized user a corresponding role on the admin database. If a role on the on the admin database exactly matches the DN, mongod grants the user the roles and privileges assigned to that role. See the db.createRole() method for more information on creating roles.

EXAMPLE

This

Give Feedback

LDAP
query
returns

any
groups
listed in
the
LDAP
user
object's
memberOf
attribute.

"{USER}?"

Your
LDAP
configuration
may not
include
the
memberOf
attribute
as part
of the
user
schema,
may
possess
a
~~different~~

[Give Feedback](#)

a different attribute for reporting group membership or may not track group membership through attributes.

Configure your query with respect to your own unique LDAP configuration.

If unset, mongod cannot authorize users using LDAP.

This setting can be configured on a running mongod using the

[Give Feedback](#)

`setParameter`
`database`
`command.`

 **NOTE**

An explanation of RFC4515², RFC4516³ or LDAP queries is out of scope for the MongoDB Documentation. Please review the RFC directly or use your preferred LDAP resource.

`security.ldap.valid`

Type: boolean

[Give Feedback](#)

`type: boolean`

Default: true

Available in
MongoDB
Enterprise

A flag that
determines if the
`mongod` or `mongos`
instance checks
the availability of
the
`LDAP server(s)`
as part of its
startup:

- If `true`, the
`mongod` or
`mongos`
instance
performs the
availability
check and
only
continues to
start up if the
LDAP server
is available.
- If `false`, the
`mongod` or
`mongos`

[Give Feedback](#)

instance
skips the
availability

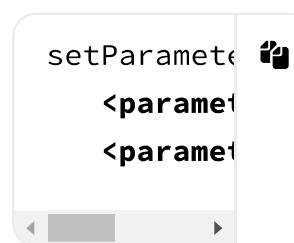
check; i.e. the
instance
starts up
even if the
LDAP server
is
unavailable.

setParameter **Option**

setParameter

Set MongoDB
parameter or
parameters
described in
MongoDB Server
Parameters

To set parameters
in the YAML
configuration file,
use the following
format:



For example, to

[Give Feedback](#)

specify the
`enableLocalhost`

in the
configuration file:



LDAP Parameters

`setParameter.ldapU`

Type: int

Default: 30

For use with
`mongod` servers
using
LDAP
Authorization.

The interval (in
seconds) `mongod`
waits between
external user
cache flushes.

After `mongod`
flushes the
external user
cache, MongoDB
reacquires

[Give Feedback](#)

authorization data from the LDAP server the next time an LDAP-authorized user issues an operation.

Increasing the value specified increases the amount of time mongod and the LDAP server can be out of sync, but reduces the load on the LDAP server. Conversely, decreasing the value specified decreases the time mongod and the LDAP server can be out of sync while increasing the load on the LDAP server.

setParameter:
ldapUserCa

storage Options

Give Feedback

Changed in version

4.4:

- MongoDB removes the `storage.indexBuild` option and the corresponding command-line option.
- MongoDB deprecates `storage.wiredT` option. The option has no effect starting in MongoDB 4.4.

```
storage:   
  dbPath: <string>  
  journal:  
    enabled  
    commitInterval  
    directoryPath  
    syncPeriod  
  engine: <string>  
  wiredTiger:  
    engineConfig  
    cacheSize  
    journal  
    directoryPath  
    maxConns
```

[Give Feedback](#)

```
collect
blockSize
indexCacheSize
prefetch
inMemory:
  engineCacheSize
  inMemorySize
oplogMinReplayTime
```

storage.dbPath

Type: string

Default:

- `/data/db` on Linux and macOS
- `\data\db` on Windows

The directory where the `mongod` instance stores its data.

The `storage.dbPath` setting is available only for `mongod`

NOTE

**Configuration
Files**

[Give Feedback](#)

The
default
`mongod.conf`

configuration
file
included
with
package
manager
installations
uses the
following
platform-
specific
default
values
for
`storage.dl`

Platform

RHEL /
CentOS
and
Amazon

SUSE

Ubuntu
and
Debian

macOS

Give Feedback

Platform

The
Linux
package
init
scripts
do not
expect
`storage.dl`
to
change
from the
defaults.
If you
use the
Linux
packages
and
change
`storage.dl`
, you will
have to
use your
own init
scripts
and
disable
the
built-in
scripts

[Give Feedback](#)

scripts.

`storage.journal.en`

Type: boolean

Default: true on

64-bit systems,

false on 32-bit

systems

Enable or disable
the durability
journal to ensure
data files remain
valid and
recoverable. This
option applies only
when you specify
the

`storage.dbPath`
setting. `mongod`
enables journaling
by default.

The
`storage.journal`
setting is available
only for `mongod`

Not available for
`mongod` instances

[Give Feedback](#)

that use the
in-memory storage
engine.

Starting in
MongoDB 4.0, you
cannot specify
`--nojournal`
option or
`storage.journal`
for replica set
members that use
the WiredTiger
storage engine.

`storage.journal.co`

Type: number

Default: 100

The maximum
amount of time in
milliseconds that
the `mongod`
process allows
between journal
operations. Values
can range from 1 to
500 milliseconds.

Lower values
increase the
durability of the
journal, at the

[Give Feedback](#)

expense of disk performance.

On WiredTiger, the default journal commit interval is 100 milliseconds. Additionally, a write that includes or implies `j : true` will cause an immediate sync of the journal. For details or additional conditions that affect the frequency of the sync, see [Journaling Process](#).

The `storage.journal` setting is available only for `mongod`

Not available for `mongod` instances that use the in-memory storage engine.



NOTE

[Give Feedback](#)

Known
Issue in

4.2.0:
The
`storage.jc`
is
missing
in 4.2.0.

storage.directoryPerDB

Type: boolean

Default: false

When `true`,
MongoDB uses a
separate directory
to store data for
each database.
The directories are
under the
`storage.dbPath`
directory, and
each subdirectory
name corresponds
to the database
name.

The
`storage.directoryPerDB`
setting is available

[Give Feedback](#)

only for `mongod`

Not available for
`mongod` instances
that use the
in-memory storage
engine.

Starting in
MongoDB 5.0,
dropping the final
collection in a
database (or
dropping the
database itself)
when

`storage.directo`
is enabled deletes
the newly empty
subdirectory for
that database.

To change the
`storage.directo`
option for existing
deployments:

- For
standalone
instances:

1. Use
`mongod`

Give Feedback

on the
existing
`mongod`
instance
to
generate
a
backup.

2. Stop the
`mongod`
instance

3. Add the
`storage`
value
and
configure
a new
data
directory

4. Restart
the
`mongod`
instance

5. Use
`mongore`
to
populate
the new
data
directory

- For replica

[Give Feedback](#)

sets:

1. Stop a
seconda
member.

2. Add the
storage
value
and
configure
a new
data
directory
to that
seconda
member.

3. Restart
that
seconda

4. Use
initial
sync
to
populate
the new
data
directory

5. Update
remainin
seconda

Give Feedback

in the
same
fashion.

6. Step
down
the
primary,
and
update
the
stepped-
down
member
in the
same
fashion.

storage.syncPeriod

Type: number

Default: 60

The amount of
time that can pass
before MongoDB
flushes data to the
data files via an
fsync operation.

**Do not set this
value on
production
systems. It almost**

[Give Feedback](#)

systems. In almost every situation, you should use the default setting.

 **WARNING**

If you set `storage.syncPer` to `0`, MongoDB will not sync the memory mapped files to disk.

The `mongod` process writes data very quickly to the journal and lazily to the data files.

`storage.syncPer` has no effect on the `journal` files or journaling, but if `storage.syncPer` is set to `0` the

[Give Feedback](#)

journal will eventually consume all available disk space. If you set `storage.syncPeriodMS` to 0 for testing purposes, you should also set `--nojournal` to true.

The `storage.syncPeriodMS` setting is available only for mongod

Not available for mongod instances that use the in-memory storage engine.

storage.engine

Default:

wiredTiger

NOTE

Starting in version 4.2, MongoDB removes

[Give Feedback](#)

the
deprecated

MMAPv1
storage
engine.

The storage engine
for the `mongod`
database.

Available values
include:

Value	Description
wiredTiger	To specify the WiredTiger Storage Engine.

Give Feedback

Value	Description
-------	-------------

`inMemory` To spe

In-Mem
Storage

Availab

Mongod

Enterpr

If you c

start a

with a

storage

that co

data fi

produ

storag

other t

one sp

storage

, mongod

refuse

◀ ▶

`storage.oplogMinRe`

Type: double

New in version 4.4:
Specifies the

Give Feedback

minimum number
of hours to
preserve an oplog
entry, where the

decimal values
represent the
fractions of an
hour. For example,
a value of 1.5
represents one
hour and thirty
minutes.

The value must be
greater than or
equal to 0. A value
of 0 indicates that
the mongod should
truncate the oplog
starting with the
oldest entries to
maintain the
configured
maximum oplog
size.

Defaults to 0.

A mongod started
with
oplogMinRetenti
only removes an
oplog entry if:

- The oplog
has reached

[Give Feedback](#)

the maximum

configured
oplog size
and

- The oplog entry is older than the configured number of hours based on the host system clock.

The mongod has the following behavior when configured with a minimum oplog retention period:

- The oplog can grow without constraint so as to retain oplog entries for the configured number of hours. This

Give Feedback

may result in reduction or exhaustion of system disk space due to a combination of high write volume and large retention period.

- If the oplog grows beyond its maximum size, the `mongod` may continue to hold that disk space even if the oplog returns to its maximum size *or* is configured for a smaller maximum size. See

Reducing
Oplog Size
Does Not
Immediately
Return Disk
Space.

[Give Feedback](#)

- The mongod compares the system wall

clock to an oplog entries creation wall clock time when enforcing oplog entry retention.

Clock drift between cluster components may result in unexpected oplog retention behavior. See

Clock Synchronization for more information on clock synchronization across cluster members.

To change the minimum oplog retention period after starting the

[Give Feedback](#)

mongod, use

replSetResizeOp

•

replSetResizeOp

enables you to

resize the oplog

dynamically

without restarting

the mongod

process. To persist

the changes made

using

replSetResizeOp

through a restart,

update the value

of

oplogMinRetenti

storage.wiredTiger

Options

storage:

wiredTiger

engineC

cach

journ

dire

maxC

collect

block

indexCo

pref



Give Feedback

storage.wiredTiger

Type: float

Defines the maximum size of the internal cache that WiredTiger will use for all data. The memory consumed by an index build (see maxIndexBuildMe) is separate from the WiredTiger cache memory.

Values can range from 0.25 GB to 10000 GB.

Starting in MongoDB 3.4, the default WiredTiger internal cache size is the larger of either:

- 50% of (RAM - 1 GB), or
- 256 MB.

For example, on a system with a total

[Give Feedback](#)

of 4GB of RAM the
WiredTiger cache
will use 1.5GB of
RAM

$(0.5 * (4 \text{ GB} - 1))$

Conversely, a
system with a total
of 1.25 GB of RAM
will allocate 256
MB to the
WiredTiger cache
because that is
more than half of
the total RAM
minus one
gigabyte

$(0.5 * (1.25 \text{ GB} - 1))$

 NOTE

In some
instances,
such as
when
running
in a
container,
the
database
can
have
memory
constraints

[Give Feedback](#)

CONSTRAINTS

that are
lower
than the

total
system
memory.

In such
instances,
this
memory
limit,
rather
than the
total
system
memory,
is used
as the
maximum
RAM
available.

To see
the
memory
limit,
see
`hostInfo.s`

Avoid increasing

[Give Feedback](#)

the WiredTiger internal cache size

above its default value.

With WiredTiger, MongoDB utilizes both the WiredTiger internal cache and the filesystem cache.

Via the filesystem cache, MongoDB automatically uses all free memory that is not used by the WiredTiger cache or by other processes.

NOTE

The `storage.w...` limits the size of the WiredTiger internal cache.

[Give Feedback](#)

The operating system will use the available free memory for filesystem cache, which allows the compressed MongoDB data files to stay in memory.

In addition, the operating system will use any free RAM to buffer file system blocks and files.

[Give Feedback](#)

and the
system
cache.

To accommodate the additional consumers of RAM, you may have to decrease WiredTiger internal cache size.

The default WiredTiger internal cache size value assumes that there is a single `mongod` instance per machine. If a single machine contains multiple MongoDB instances, then you should decrease the setting to

[Give Feedback](#)

accommodate the other `mongod` instances.

If you run `mongod` in a container (e.g. `lxc`, `cgroups`, Docker, etc.) that does *not* have access to all of the RAM available in a system, you must set

`storage.wiredTi` to a value less than the amount of RAM available in the container.

The exact amount depends on the other processes running in the container. See

`memLimitMB`

`storage.wiredTiger`

Default: snappy

Specifies the type of compression to use to compress WiredTiger journal data

[Give Feedback](#)

Available compressors are:

- none
 - snappy
 - zlib
 - zstd
- (Available starting in MongoDB 4.2)

`storage.wiredTiger`

Type: boolean

Default: false

When

`storage.wiredTi` is true, mongod stores indexes and collections in separate subdirectories under the data (i.e. `storage.dbPath`) directory.

Specifically,

mongod stores the indexes in a subdirectory

[Give Feedback](#)

named `index` and
the collection data

in a subdirectory
named
`collection`.

By using a
symbolic link, you
can specify a
different location
for the indexes.
Specifically, when
`mongod` instance is
not running, move
the `index`
subdirectory to the
destination and
create a symbolic
link named `index`
under the data
directory to the
new destination.

`storage.wiredTiger`

Type: float

 NOTE

Deprecated
in
MongoDB

[Give Feedback](#)

4.4

MongoDB
deprecates

the
`storage.wt`
option.

The
option
has no
effect
starting
in
MongoDB
4.4.

Specifies the
maximum size (in
GB) for the
"lookaside (or
cache overflow)
table" file
`WiredTigerLAS.wt`
for MongoDB 4.2.1-
4.2.x and 4.0.12-
4.0.x. The file no
longer exists
starting in version
4.4.

The setting can
accept the

[Give Feedback](#)

following values:

Value	Description
-------	-------------

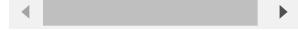
0 The default value. If set to 0, the file size is unbounded.

numb
er >= 0.1 The maximum size (in GB). If the WiredTiger LAS.wt file exceeds this size,

`mongod`

exits with a fatal assertion. You can clear the WiredTiger LAS.wt file and restart

`mongod`



To change the maximum size during runtime, use the `wiredTigerMaxCa` parameter.

[Give Feedback](#)

*Available starting
in MongoDB 4.2.1
(and 4.0.12)*

storage.wiredTiger

Type: integer

Default: 6

Specifies the level of compression applied when using the zstd compressor.

Values can range from 1 to 22.

The higher the specified value for `zstdCompression` the higher the compression which is applied.

Only applicable when `blockCompressor` is set to `zstd`.

*Available starting
in MongoDB 5.0*

storage.wiredTiger

[Give Feedback](#)

Default: snappy

Specifies the default compression for collection data. You can override this on a per-collection basis when creating collections.

Available compressors are:

- none
 - snappy
 - zlib
 - zstd
- (Available starting MongoDB 4.2)

`storage.wiredTl`
affects all
collections
created. If you
change the value
of
`storage.wiredTl`

Give Feedback

on an existing MongoDB deployment, all new collections will use the specified compressor. Existing collections will continue to use the compressor specified when they were created, or the default compressor at that time.

`storage.wiredTiger`

Default: true

Enables or disables prefix compression for index data.

Specify `true` for `storage.wiredTiger` to enable prefix compression for index data, or `false` to disable prefix compression for index data.

The

`storage.wiredTiger`

[Give Feedback](#)

storage.wiredT

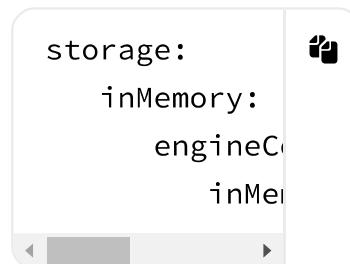
setting affects all indexes created. If you change the

value of

storage.wiredTi

on an existing MongoDB deployment, all new indexes will use prefix compression.

Existing indexes are not affected.

storage.inmemory**Options****storage.inMemory.e**

Type: float

Default: 50% of physical RAM less 1 GB

Changed in version 3.4: Values can range from

[Give Feedback](#)

256MB to 10TB and can be a float.

Maximum amount of memory to allocate for in-memory storage engine data, including indexes, oplog if the `mongod` is part of replica set, replica set or sharded cluster metadata, etc.

By default, the in-memory storage engine uses 50% of physical RAM minus 1 GB.

 **NOTE**

Enterprise Feature

Available in MongoDB Enterprise only.

[Give Feedback](#)

operationProfiling Options

```
operationProfiling
  mode: <string>
  slowOpThresholdMS: <number>
  slowOpSampleRate: <number>
  filter: <string>
```

operationProfiling

Type: string

Default: off

Specifies which operations should be profiled. The following profiler levels are available:

Level	Description
off	The profiler is off and does not collect any data. This is the default profiler level.

[Give Feedback](#)

Level	Description
slowOp	The profiler collects data for operations that take longer than the value of <code>slowms</code> .
all	The profiler collects data for all operations.

◀ ▶

! IMPORTANT

Profiling can impact performance and shares settings with the system log. Carefully consider

Give Feedback

any performance and security implications before configuring and enabling the profiler on a production deployment.

See Profiler Overhead for more information on potential performance degradation.

operationProfiling

Type: integer

Default: 100

The *slow* operation time threshold, in milliseconds.

[Give Feedback](#)

Operations that

run for longer than
this threshold are
considered *slow*.

When `logLevel` is
set to `0`, MongoDB
records *slow*
operations to the
diagnostic log at a
rate determined by
`slowOpSampleRat`

At higher
`logLevel`
settings, all
operations appear
in the diagnostic
log regardless of
their latency with
the following
exception: the
logging of
slow oplog entry
messages by the
secondaries
. The secondaries
log only the slow
oplog entries;
increasing the
`logLevel` does
not log all oplog

[Give Feedback](#)

entries.

*Changed in
version 4.0:* The `slowOpThreshold` setting is available for `mongod` and `mongos`. In earlier versions, `slowOpThreshold` is available for `mongod` only.

- For `mongod` instances, the setting affects both the diagnostic log and, if enabled, the profiler.
- For `mongos` instances, the setting affects the diagnostic log only and not the profiler since profiling is not available on `mongos`.

[Give Feedback](#)

on mongos

operationProfiling

Type: double

Default: 1.0

The fraction of
slow operations
that should be
profiled or logged.

operationProfiling
accepts values
between 0 and 1,
inclusive.

Changed in
version 4.0: The
slowOpSampleRatio
setting is available
for mongod and
mongos. In earlier
versions,
slowOpSampleRatio
is available for
mongod only.

- For mongod instances, the setting affects both the diagnostic log and, if enabled, the profiler.

Give Feedback

- For `mongos` instances, the setting affects the diagnostic log only and not the profiler since profiling is not available on `mongos`

operationProfiling

Type: string
representation of a query document

A filter expression that controls which operations are profiled and logged.

When `filter` is set, `slowOpThreshold` and `slowOpSampleRate` are not used for profiling and slow-query log lines.

When you set a ~~slowOpThreshold~~

[Give Feedback](#)

profile filter in the configuration file, the filter applies to all databases in the deployment. To set a profile filter for a specific database, use the `db.setProfiling` method.

The option takes a string representation of a query document of the form:



The `<field>` can be any field in the profiler output. The `<expression>` is a query condition expression.

To specify a profiling filter in a configuration file, you must:

[Give Feedback](#)

- Enclose the filter document in single quotes to pass the document as a string.
- Use the YAML format of the configuration file.

For example, the following `filter` configures the profiler to log `query` operations that take longer than 2 seconds:



*New in version
4.4.2.*

replication Options

replication:
oplogSizeM
replSetName

Give Feedback

enableMajorityOplog

replication.oplogSize

Type: integer

The maximum size in megabytes for the replication operation log (i.e., the oplog).

NOTE

Starting in MongoDB 4.0, the oplog can grow past its configured size limit to avoid deleting the majority of

By default, the

Give Feedback

mongod process creates an oplog based on the maximum amount of space available. For 64-bit systems, the oplog is typically 5% of available disk space.

Once the mongod has created the oplog for the first time, changing the replication.opl option will not affect the size of the oplog. To change the maximum oplog size after starting the mongod, use

replSetResizeOp

.

replSetResizeOp enables you to resize the oplog dynamically without restarting the mongod process. To persist the changes made

Give Feedback

using

`replSetResizeOp`

through a restart,
update the value
of `oplogSizeMB`

See Oplog Size for
more information.

The

`replication.opl`
setting is available
only for `mongod`

`replication.replSet`

Type: string

The name of the
replica set that the
`mongod` is part of.

All hosts in the
replica set must
have the same set
name.

If your application
connects to more
than one replica
set, each set must
have a distinct
name. Some
drivers group

[Give Feedback](#)

replica set
connections by
replica set name.

The
`replication.rep`
setting is available
only for `mongod`

Starting in
MongoDB 4.0:

- The setting
`replication`
cannot be
used in
conjunction
with
`storage.ind`
- For the
`WiredTiger`
storage
engine,
`storage.jou`
cannot be
used in
conjunction
with
`replication`

`replication.enable`

Default: true

[Give Feedback](#)

Configures support for "majority" read concern. Starting in MongoDB 5.0, `enableMajorityR` cannot be changed and is always set to `true`. Attempting to start a storage engine that does not support majority read concern with the `--enableMajorityR` option will fail and return an error message.

In earlier versions of MongoDB, `enableMajorityR` was configurable.

 **WARNING**

If you are using a three-member

[Give Feedback](#)

primary-
secondary-
arbiter

(PSA)
architecture
consider
the
following:

- The write concern "majority" can cause performance issues if a secondary is unavailable or lagging. For advice on how to mitigate these issues, see

[Give Feedback](#)

Mitigat
Perform
Issues
PSA Re
Set.

- If you are using a global default `"majority"` and the write concern is less than the size of the majority your queries may return stale (not fully replicated) data.

[Give Feedback](#)

sharding Options

sharding:	
clusterRole	
archiveMove	

sharding.clusterRole

Type: string

The role that the mongod instance has in the sharded cluster. Set this setting to one of the following:

Value	Description
-------	-------------

[Give Feedback](#)

Value	Description
<code>configsvr</code>	<p>Start this instance as a config server. The instance starts on port <code>27019</code> by default.</p> <p>When you configure a MongoDB instance as a cluster Registrar (<code>clusterRegistry</code>), you must specify the <code>replSet</code>.</p>

[Give Feedback](#)

Value	Description
-------	-------------

`shardsvr` Start this instance of shard. This instance starts on port `27018` by default.

When you configure MongoDB instances in a cluster, you must specify the `replicaSet`.



NOTE

Setting `sharding` requires the `mongod`

[Give Feedback](#)

instance
to be
running
with
replication.
To
deploy
the
instance
as a
replica
set
member,
use the
`replSetName`
setting
and
specify
the
name of
the
replica
set.

The
`sharding.clusters`
setting is available
only for `mongod`

sharding.archiveMode

Type: boolean

[Give Feedback](#)

*Changed in
version 3.2:
Starting in 3.2,*

MongoDB uses
`false` as the
default.

During chunk
migration, a shard
does not save
documents
migrated from the
shard.

auditLog Options

ⓘ NOTE

Available
only in
MongoDB
Enterprise
and
MongoDB  Atlas

auditLog:
destination:
format: <str>
path: <str>
filter: <str>



Give Feedback

auditLog.auditEncr

Type: string

New in version 6.0.

Specifies the unique identifier of the Key Management Interoperability Protocol (KMIP) key for audit log encryption.

You cannot use

auditLog.auditE

and

auditLog.localA

together.

NOTE

Available only in MongoDB Enterprise . MongoDB Enterprise and Atlas have different

[Give Feedback](#)

configuration
requirement

`auditLog.compressi`

Type: string

New in version 5.3.

Specifies the compression mode for audit log encryption. You must also enable audit log encryption using either

`auditLog.auditE`

or

`auditLog.localA`

.

`auditLog.compre`

can be set to one of these values:

Value	Description
-------	-------------

<code>zstd</code>	Use the zstd algorithm to compress the audit log.
-------------------	---

[Give Feedback](#)

none	Do not
Value (it)	Description the audit log.

◀ ▶

 NOTE

Available
only in
MongoDB
Enterprise
. .
MongoDB
Enterprise
and
Atlas
have
different
configuratio
requirement

auditLog.destination

Type: string

When set,

auditLog.destination
enables auditing
and specifies
where mongos or
mongod sends all
audit events.

Give Feedback

auditLog.destination

can have one of
the following

values:

Value	Description
syslog	Output the events to syslog in JSON format. This is available on Linux and Windows. The syslog server will receive messages at the specified facility level from the user.
console	Output the events to standard output in JSON format.

[Give Feedback](#)

Value	Description
-------	-------------

`file` Output the events to the specified `auditLog` in the format specified in `auditLog`



NOTE

Available only in MongoDB Enterprise and MongoDB Atlas

`auditLog.filter`

Type: string representation of a document

The filter to limit the

[Give Feedback](#)

types of operations the audit system records. The option takes a string representation of a query document of the form:



The `<field>` can be any field in the audit message, including fields returned in the param document.

The `<expression>` is a query condition expression.

To specify an audit filter, enclose the filter document in single quotes to pass the document as a string.

To specify the audit filter in a configuration file,

[Give Feedback](#)

you must use the

YAML format of
the configuration
file.

 **NOTE**

Available
only in
MongoDB
Enterprise
and
MongoDB
Atlas
.

`auditLog.format`

Type: string

The format of the
output file for
auditing if
`destination` is
`file`. The
`auditLog.format`
option can have
one of the
following values:

[Give Feedback](#)

following values.

Value	Description
JSON	Output the audit events in JSON format to the file specified in <code>auditLog.path</code> .
BSON	Output the audit events in BSON binary format to the file specified in <code>auditLog.path</code> .

JSON Output the audit events in JSON format to the file specified in `auditLog.path`.

BSON Output the audit events in BSON binary format to the file specified in `auditLog.path`.

◀ ▶

Printing audit events to a file in JSON format degrades server performance more than printing to a file in BSON format.

NOTE

Available only in MongoDB Enterprise and MongoDB Cloud.

[Give Feedback](#)

ATLAS**auditLog.localAuditKeyPath**

Type: string

New in version 5.3.

Specifies the path and file name for a local audit key file for audit log encryption.

NOTE

Only use `auditLog.localAuditKeyPath` for testing because the key is not secured. To secure the key, use `auditLog.auditKeyFile` and an external

[Give Feedback](#)

Key Management Interoperability Protocol (KMIP) server.

You cannot use `auditLog.localAuditLog` and `auditLog.auditEvent` together.

 **NOTE**

Available only in MongoDB Enterprise. MongoDB Enterprise and Atlas have different configuration requirements.

auditLog.path

[Give Feedback](#)

Type: string

The output file for auditing if destination has value of file. The auditLog.path option can take either a full path name or a relative path name.

auditLog.runtimeCo

Type: boolean

Specifies if a node allows runtime configuration of audit filters and the auditAuthorization variable. If true the node can take part in Online Audit Filter Management.

NOTE

Available only in

Give Feedback

MongoDB
Enterprise

and

MongoDB
Atlas

snmp Options

NOTE

MongoDB
Enterprise
on macOS
does *not*
include
support for
SNMP due
to
SERVER-
29352.

snmp:
disabled:
subagent:
master: <ba



snmp.disabled

Type: boolean

Default: false

[Give Feedback](#)

Disables SNMP
access to `mongod`.
The option is

incompatible with
`snmp.subagent`
and `snmp.master`

Set to `true` to
disable SNMP
access.

The
`snmp.disabled`
setting is available
only for `mongod`

*New in version
4.0.6.*

snmp.subagent

Type: boolean

When
`snmp.subagent` is
`true`, SNMP runs
as a subagent. The
option is
incompatible with
`snmp.disabled`
set to `true`.

The
`snmp.subagent`
setting is available
only for `mongod`

[Give Feedback](#)

only for mongod

snmp.master

Type: boolean

When

snmp.master is true, SNMP runs as a master. The option is incompatible with snmp.disabled set to true.

The snmp.master setting is available only for mongod



TIP

See also:

- Monitor MongoDB With SNMP on Linux
- Monitor MongoDB Windows with SNMP
- Troubleshoot SNMP

mongod only

Give Feedback

mongos -only Options

Changed in version

3.4: MongoDB 3.4

removes

`sharding.chunkSize`

and

`sharding.autoSplit`

settings.



`replication.localP`

Type: integer

Default: 15

The ping time, in milliseconds, that mongos uses to determine which secondary replica set members to pass read operations from clients. The default value of 15

[Give Feedback](#)

corresponds to the default value in all of the client drivers.

When `mongos` receives a request that permits reads to secondary members, the `mongos` will:

- Find the member of the set with the lowest ping time.
- Construct a list of replica set members that is within a ping time of 15 milliseconds of the nearest suitable member of the set.

If you specify a value for the `replication` option,

[Give Feedback](#)

`mongos` will construct the list of replica members that are within the latency allowed by this value.

- Select a member to read from at random from this list.

The ping time used for a member compared by the `replication.loc` setting is a moving average of recent ping times, calculated at most every 10 seconds. As a result, some queries may reach members above the threshold until the `mongos` recalculates the average.

See the

Give Feedback

Read Preference
for Replica Sets
section of the

read preference
documentation for
more information.

sharding.configDB

Type: string

*Changed in
version 3.2.*

The
configuration
servers
for the
sharded cluster.

Starting in
MongoDB 3.2,
config servers for
sharded clusters
can be deployed
as a replica set.
The replica set
config servers
must run the
WiredTiger storage
engine
. MongoDB 3.2
deprecates the use
of three mirrored
mongod instances

[Give Feedback](#)

Feedback

for config servers.

Specify the config server replica set name and the hostname and port of at least one of the members of the config server replica set.



The mongos instances for the sharded cluster must specify the same config server replica set name but can specify hostname and port of different members of the replica set.

Windows Service Options

[Give Feedback](#)



processManagement.

Type: string

Default: MongoDB

The service name
of mongos or
mongod when
running as a
Windows Service.
Use this name with
the
`net start <name>`
and
`net stop <name>`
operations.

You must use
`processManagement`
in conjunction with
either the
`--install` or
`--remove` option.

[Give Feedback](#)

processManagement.

Type: string

Default: MongoDB

The name listed for MongoDB on the Services administrative application.

processManagement.

Type: string

Default: MongoDB

Server

Run mongos or mongod service description.

You must use processManagement in conjunction with the --install option.

For descriptions that contain spaces, you must enclose the description in quotes.

Give Feedback

processManagement.

Type: string

The mongos or mongod service in the context of a certain user. This user must have "Log on as a service" privileges.

You must use processManagement in conjunction with the --install option.

processManagement.

Type: string

The password for <user> for mongos or mongod when running with the processManagement option.

You must use processManagement in conjunction with the --install option.

[Give Feedback](#)

option.

Removed MMAPv1 Options

Starting in version 4.2, MongoDB removes the deprecated MMAPv1 storage engine and the MMAPv1-specific configuration options:

Removed Configuration Files

`storage.mmapv1.journal`

`storage.mmapv1.journalSize`

`storage.mmapv1.nsSize`

`storage.mmapv1.prealloc`

`storage.mmapv1.quota.enable`

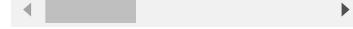
`storage.mmapv1.quota.size`

`storage.mmapv1.smallFile`

`storage.repairPath`

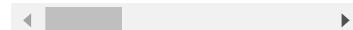
[Give Feedback](#)

Removed Configuration File replication.secondary:



For earlier versions of MongoDB, refer to the corresponding version of the manual. For example:

- <https://www.mongodb.com/docs/manual/reference/configuration-options/>
- <https://www.mongodb.com/docs/manual/replica-set/>
- <https://www.mongodb.com/docs/manual/replica-set/secondary/>



About

Careers

Investor Relations

Legal Notices

Privacy Notices

Security Information

Trust Center

Give Feedback

Support

[Contact Us](#)[Customer Portal](#)[Atlas Status](#)[Paid Support](#)

Social

[Github](#)[Stack Overflow](#)[LinkedIn](#)[Youtube](#)[Twitter](#)[Twitch](#)[Facebook](#)

© 2022 MongoDB, Inc.

[Give Feedback](#)