- Transactions
- Administration
- Storage
- Frequently Asked Questions
- Reference
  - Collation
  - Configuration File Options
  - Connection Strings
  - Database Commands
  - Default MongoDB Port
  - Default MongoDB Read Concerns/Write Concerns
  - Exit Codes and Statuses
  - Explain Results
  - Glossary
  - Log Messages
  - MongoDB Cluster Parameters
  - MongoDB Limits and Thresholds
  - MongoDB Package Components
    - mongod
    - mongos
    - mongod.exe
    - mongos.exe
    - mongoldap
    - mongokerberos
    - install_compass
    - MongoDB Database Tools
  - MongoDB Server Parameters
  - MongoDB Wire Protocol
  - mongosh Methods
  - Operators
  - Server Sessions
  - Slot-Based Query Execution Engine
  - Stable API
  - System Collections
  - Legacy mongo Shell
- Release Notes
- Technical Support

mongos

On this page

- Synopsis
- Considerations
- Options
  - Core Options
  - Sharded Cluster Options
  - TLS Options
  - SSL Options (Deprecated)
  - Audit Options
  - Profiler Options
  - LDAP Authentication and Authorization Options
  - Additional Options

Synopsis

For a sharded cluster, the `mongos` instances provide the interface between the client applications and the sharded cluster. The `mongos` instances route queries and write operations to the shards. From the perspective of the application, a `mongos` instance behaves identically to any other MongoDB instance.

Considerations

- Never change the name of the `mongos` binary.
- Starting in version 4.4, `mongos` supports hedged reads to minimize latencies.
- MongoDB disables support for TLS 1.0 encryption on systems where TLS 1.1+ is available. For more details, see Disable TLS 1.0.
- The `mongos` binary cannot connect to `mongod` instances whose feature compatibility version (fCV) is greater than that of the `mongos`. For example, you cannot connect a MongoDB 6.0 version `mongos` to a 7.0 sharded cluster with fCV set to 7.0. You can, however, connect a MongoDB 6.0 version `mongos` to a 7.0 sharded cluster with fCV set to 6.0.
- `mongod` includes a Full Time Diagnostic Data Capture mechanism to assist MongoDB engineers with troubleshooting deployments. If this thread fails, it terminates the originating process. To avoid the most common failures, confirm that the user running the process has permissions to create the FTDC `diagnostic.data` directory. For `mongod` the directory is within `storage.dbPath`. For `mongos` it is parallel to `systemLog.path`.

Options

TIP

See also:
Configuration File Settings and Command-Line Options Mapping

NOTE

- MongoDB deprecates the SSL options and instead adds new corresponding TLS options.
- MongoDB adds `--tlsClusterCAFile`/`net.tls.clusterCAFile`.

NOTE

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

Core Options

`--help, -h`

Returns information on the options and use of `mongos`.

`--version`

Returns the `mongos` release number.

`--config <filename>, -f <filename>`

Specifies a configuration file for runtime configuration options. The configuration file is the preferred method for runtime configuration of `mongos`. The options are equivalent to the command-line configuration options. See Configuration File Options for more information.

Ensure the configuration file uses ASCII encoding. The `mongos` instance does not support configuration files with non-ASCII encoding, including UTF-8.

`--configExpand <none|rest|exec>`

Default: none

New in version 4.2.

Enables using Expansion Directives in configuration files. Expansion directives allow you to set externally sourced values for configuration file options.

`--configExpand` supports the following expansion directives:

Value	Description
`none`	Default. `mongos` does not expand expansion directives. `mongos` fails to start if any configuration file settings use expansion directives.

Value	Description
`rest`	`mongos` expands `__rest` expansion directives when parsing the configuration file.
`exec`	`mongos` expands `__exec` expansion directives when parsing the configuration file.

You can specify multiple expansion directives as a comma-separated list, for example: `rest, exec`. If the configuration file contains expansion directives not specified to `--configExpand`, the `mongos` returns an error and terminates.

See Externally Sourced Configuration File Values for configuration files for more information on expansion directives.

`--verbose, -v`

Increases the amount of internal reporting returned on standard output or in log files. Increase the verbosity with the `-v` form by including the option multiple times, for example: `-vvvvv`.

`--quiet`

Runs `mongos` in a quiet mode that attempts to limit the amount of output.

This option suppresses:

- output from database commands
- replication activity
- connection accepted events
- connection closed events

`--port <port>`

Default: 27017

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

Changed in version 7.0.3: The `--port` option accepts a range of values between `0` and `65535`. Setting the port to `0` configures `mongos` to use an arbitrary port assigned by the operating system.

`--bind_ip <hostnames|ipaddresses|Unix domain socket paths>`

Default: localhost

The hostnames and/or IP addresses and/or full Unix domain socket paths on which `mongos` should listen for client connections. You may attach `mongos` to any interface. To bind to multiple addresses, enter a list of comma-separated values.

EXAMPLE

localhost,/tmp/mongod.sock

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

EXAMPLE

localhost, 2001:0DB8:e132:ba26:0d5c:2774:e7f9:d513

NOTE

If specifying an IPv6 address *or* a hostname that resolves to an IPv6 address to `--bind_ip`, you must start `mongos` with `--ipv6` to enable IPv6 support. Specifying an IPv6 address to `--bind_ip` does not enable IPv6 support.

Share Feedback

If specifying a link-local IPv6 address (`fe80::/10`), you must append the zone index to that address (i.e. `fe80::<address>%<adapter-name>`).

EXAMPLE

localhost,fe80::a00:27ff:fee0:1fcf%enp0s3

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 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 validation and will not start.

WARNING

> **WARNING**
>
> Before you bind your instance to a publicly-accessible IP address, you must secure your cluster from unauthorized access. For a complete list of security recommendations, see Security Checklist. At 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 to avoid filename pattern expansion). Alternatively, use the `net.bindIpAll` setting.

> **NOTE**
>
> - `--bind_ip` and `--bind_ip_all` are mutually exclusive. Specifying both options causes `mongos` to throw an error and terminate.
> - The command-line option `--bind` overrides the configuration file setting `net.bindIp`.

`--bind_ip_all`

If specified, the `mongos` instance binds to all IPv4 addresses (i.e. `0.0.0.0`). If `mongos` starts with `--ipv6`, `--bind_ip_all` also binds to all IPv6 addresses (i.e. `::`).

`mongos` only supports IPv6 if started with `--ipv6`. Specifying `--bind_ip_all` alone does not enable IPv6 support.

> **WARNING**
>
> Before you bind your instance to a publicly-accessible IP address, you must secure your cluster from unauthorized access. For a complete list of security recommendations, see Security Checklist. At minimum, consider enabling authentication and hardening network infrastructure.

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

Alternatively, you can set the `--bind_ip` option to `::,0.0.0.0` or, starting in MongoDB 4.2, to an asterisk `"*"` (enclose the asterisk in quotes to avoid filename pattern expansion).

> **NOTE**
>
> `--bind_ip` and `--bind_ip_all` are mutually exclusive. That is, you can specify one or the other, but not both.

`--listenBacklog <number>`

Default: Target system `SOMAXCONN` constant

The maximum number of connections that can exist in the listen queue.

> **WARNING**
>
> Consult your local system's documentation to understand the limitations and configuration requirements before using this parameter.

Share Feedback

> **IMPORTANT**
>
> To prevent undefined behavior, specify a value for this parameter between `1` and the local system `SOMAXCONN` constant.

The default value for the `listenBacklog` parameter is set at compile time to the target system `SOMAXCONN` constant. `SOMAXCONN` is the maximum valid value that is documented for the *backlog* parameter to the *listen* system call.

Some systems may interpret `SOMAXCONN` symbolically, and others numerically. The actual *listen backlog* applied in practice may differ from any numeric interpretation of the `SOMAXCONN` constant or argument to `--listenBacklog`, and may also be constrained by system settings like `net.core.somaxconn` on Linux.

Passing a value for the `listenBacklog` parameter that exceeds the `SOMAXCONN` constant for the local system is, by the letter of the standards, undefined behavior. Higher values may be silently integer truncated, may be ignored, may cause unexpected resource consumption, or have other adverse consequences.

On systems with workloads that exhibit connection spikes, for which it is empirically known that the local system can honor higher values for the *backlog* parameter than the `SOMAXCONN` constant, setting the `listenBacklog` parameter to a higher value may reduce operation latency as observed by the client by reducing the number of connections which are forced into a backoff state.

`--maxConns <number>`

The maximum number of simultaneous connections that `mongos` accepts. 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.

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 `maxIncomingConnections` 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 of the sharded cluster.

`--logpath <path>`

Sends all diagnostic logging information to a log file instead of to standard output or to the host's syslog system. MongoDB creates the log file at the path you specify.

By default, MongoDB will move any existing log file rather than overwrite it. To instead append to the log file, set the `--logappend` option.

`--syslog`

Sends all logging output to the host's syslog system rather than to standard output or to a log file (`--logpath`).

The `--syslog` option is not supported on Windows.

> **WARNING**
>
> The `syslog` daemon generates timestamps when it logs a message, 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 `--logpath` option for production systems to ensure accurate timestamps.

Starting in version 4.2, MongoDB includes the component in its log messages to `syslog`.

```
...  ACCESS   [repl writer worker 5] Unsupported modification to roles collec
```

`--syslogFacility <string>`

Default: user

Specifies the facility level used when logging messages to syslog. The value you specify must be supported by your operating system's implementation of syslog. To use this option, you must enable the `--syslog` option.

`--logappend`

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

`--logRotate <string>`

Default: rename

Share 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 use `--logappend`.

`--redactClientLogData`

Available in MongoDB Enterprise only.

A `mongos` running with `--redactClientLogData` redacts any message accompanying a given log event before logging. This prevents the `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 `--redactClientLogData` in conjunction with Encryption at Rest and TLS/SSL (Transport Encryption) to assist compliance with regulatory requirements.

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

Diagnostics on a `mongos` running with `--redactClientLogData` 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 `--redactClientLogData` on log output.

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

`--timeStampFormat <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 Coordinated Universal Time (UTC) in the ISO-8601 format. For example, for New York at the start of the Epoch: `1970-01-01T00:00:00.000Z`
`iso8601-local`	Displays timestamps in local time in the ISO-8601 format. For example, for New York at the start of the Epoch: `1969-12-31T19:00:00.000-05:00`

NOTE

Starting in MongoDB 4.4, `--timeStampFormat` no longer supports `ctime`. An example of `ctime` formatted date is: `Wed Dec 31 18:17:54.811`.

`--pidfilepath <path>`

Specifies a file location to store the process ID (PID) of the `mongos` process. The user running the `mongod` or `mongos` process must be able to write to this path. If the `--pidfilepath` option is not specified, the process does not create a PID file. This option is generally only useful in combination with the `--fork` option.

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.d` directory, or a systemd unit file registered with `systemctl`. Only use the `--pidfilepath` 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, PID file management is generally handled by `brew`. Only use the `--pidfilepath` option if you are not using `brew` on your macOS system. For more information, please see the respective Installation Guide for your operating system.

`--keyFile <file>`

Specifies 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 `client authorization`. See Internal/Membership Authentication 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 either:

- A single key string (same as in earlier versions)
- A sequence of key strings

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

`--setParameter <options>`

Specifies one of the MongoDB parameters described in MongoDB Server Parameters. You can specify multiple `setParameter` fields.

`--noscripting`

Disables the scripting engine. 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`.

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

New in version 4.4.

`--nounixsocket`

Disables listening on the UNIX domain socket. `--nounixsocket` applies only to Unix-based systems.

The `mongos` process always listens on the UNIX socket unless one of the following is true:

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

`mongos` installed from official .deb and .rpm packages have the `bind_ip` configuration set to `127.0.0.1` by default.

`--unixSocketPrefix <path>`

Default: /tmp

The path for the UNIX socket. `--unixSocketPrefix` applies only to Unix-based systems.

If this option has no value, the `mongos` 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.unixDomainSocket.enabled` is `false`
- `--nounixsocket` is set
- `net.bindIp` is not set
- `net.bindIp` does not specify `localhost` or its associated IP address

`--filePermissions <path>`

Default: `0700`

Sets the permission for the UNIX domain socket file.

`--filePermissions` applies only to Unix-based systems.

`--fork`

Enables a daemon mode that runs the `mongos` process in the background. The `--fork` option is not supported on Windows.

By default `mongos` does not run as a daemon. You run `mongos` as a daemon by using either `--fork` or a controlling process that handles daemonization, such as `upstart` or `systemd`.

Using the `--fork` option requires that you configure log output for the `mongos` with one of the following:

- `--logpath`
- `--syslog`

`--transitionToAuth`

Allows the `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 `--keyFile`.

For example, if using keyfiles for internal authentication, the `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 `mongos` utilizes a non-authenticated connection instead.

A `mongos` running with `--transitionToAuth` does not enforce user access controls. Users may connect to your deployment without any access control checks and perform read, write, and administrative operations.

> **NOTE**
>
> A `mongos` running with internal authentication and *without* `--transitionToAuth` requires clients to connect using user access controls. Update clients to connect to the `mongos` using the appropriate user prior to restarting `mongos` without `--transitionToAuth`.

`--networkMessageCompressors <string>`

Default: snappy,zstd,zlib

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

- other members of the sharded cluster
- `mongosh`
- drivers that support the `OP_COMPRESSED` message format.

MongoDB supports the following compressors:

- snappy
- zlib
- zstd

Both `mongod` and `mongos` instances default to `snappy,zstd,zlib` 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 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` specifies `snappy`, messages between `mongosh` and `mongod` are not compressed.

`--timeZoneInfo <path>`

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/zoneinfo` 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 the MongoDB release cycle. The most recent release of the time zone database is available on our download site.

```
wget https://downloads.mongodb.org/olson_tz_db/timezonedb-latest.zip
unzip timezonedb-latest.zip
mongos --timeZoneInfo timezonedb-2017b/
```

> **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 in older versions of MongoDB.
>
> To explicitly link to the time zone database in versions of MongoDB prior to 5.0, 4.4.7, and 4.2.14, download the time zone database. and use the `timeZoneInfo` parameter.

`--outputConfig`

New in version 4.2.

Outputs the `mongos` instance's configuration options, formatted in YAML, to `stdout` and exits the `mongos` instance. For configuration options that uses Externally Sourced Configuration File Values, `--outputConfig` returns the resolved value for those options.

> **WARNING**
>
> This may include any configured passwords or secrets previously obfuscated through the external source.

For usage examples, see:

- Output the Configuration File with Resolved Expansion Directive Values
- Convert Command-Line Options to YAML

Share Feedback

Sharded Cluster Options

`--configdb <replicasetName>/<config1>,<config2>...`

Specifies the configuration servers for the sharded cluster.

Config servers for sharded clusters are deployed as a replica set. The replica set config servers must run the WiredTiger storage engine.

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.

```
sharding:
  configDB: <configReplSetName>/cfg1.example.net:27019, cfg2.example.net:2701
```

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.

`--localThreshold`

Default: 15

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

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

- Finds the member of the set with the lowest ping time.
- Constructs 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 `--localThreshold` option, `mongos` constructs the list of replica members that are within the latency allowed by this value.
- Selects a member to read from at random from this list.

The ping time used for a member compared by the `--localThreshold` 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 Read Preference for Replica Sets section of the read preference documentation for more information.

TLS Options

TIP

See:
Configure `mongod` and `mongos` for TLS/SSL for full documentation of MongoDB's support.

`--tlsMode <mode>`

New in version 4.2.

Enables TLS used for all network connections. The argument to the `--tlsMode` option can be one of the following:

Value	Description
`disabled`	The server does not use TLS.
`allowTLS`	Connections between servers do not use TLS. For incoming connections, the server accepts both TLS and non-TLS.
`preferTLS`	Connections between servers use TLS. For incoming connections, the server accepts both TLS and non-TLS.
`requireTLS`	The server uses and accepts only TLS encrypted connections.

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 `--tlsCertificateSelector`.

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

`--tlsCertificateKeyFile <filename>`

New in version 4.2.

NOTE

On macOS or Windows, you can use a certificate from the operating system's secure store instead of specifying a PEM file. See `--tlsCertificateSelector`.

Share Feedback

Specifies the `.pem` file that contains both the TLS certificate and key.

- On Linux/BSD, you must specify `--tlsCertificateKeyFile` when TLS is enabled.
- On Windows or macOS, you must specify either `--tlsCertificateKeyFile` or `--tlsCertificateSelector` when TLS is enabled.

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

`--tlsCertificateKeyFilePassword <value>`

New in version 4.2.

Specifies the password to decrypt the certificate-key file (i.e. `--tlsCertificateKeyFile`). Use the `--tlsCertificateKeyFilePassword` option only if the certificate-key file is encrypted. In all cases, the `mongos` redacts the password from all logging and reporting output.

- On Linux/BSD, if the private key in the PEM file is encrypted and you do not specify the `--tlsCertificateKeyFilePassword` option, MongoDB prompts for a passphrase. See TLS/SSL Certificate Passphrase.

- On macOS or Windows, if the private key in the PEM file is encrypted, you must explicitly specify the `--tlsCertificateKeyFilePassword` option. Alternatively, you can use a certificate from the secure system store (see `--tlsCertificateSelector`) instead of a PEM file or use an unencrypted PEM file.

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

`--clusterAuthMode <option>`

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. Accept only keyfiles.
`sendKeyFile`	For rolling upgrade purposes. Send a keyfile for authentication but can accept both keyfiles and x.509 certificates.
`sendX509`	For rolling upgrade purposes. Send the x.509 certificate for authentication but can accept both keyfiles and x.509 certificates.
`x509`	Recommended. Send the x.509 certificate for authentication and accept only x.509 certificates.

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 `--tlsCertificateSelector`.

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

`--tlsClusterFile <filename>`

New in version 4.2.

> **NOTE**
>
> On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM file. See `--tlsClusterCertificateSelector`.

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

If `--tlsClusterFile` does not specify the `.pem` file for internal cluster authentication or the alternative `--tlsClusterCertificateSelector`, the cluster uses the `.pem` file specified in the `--tlsCertificateKeyFile` option or the certificate returned by the `--tlsCertificateSelector`.

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

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 TLS and MongoDB, see Configure `mongod` and `mongos` for TLS/SSL and TLS/SSL Configuration for Clients .

`--tlsClusterPassword <value>`

Share Feedback

New in version 4.2.

Specifies the password to decrypt the x.509 certificate-key file specified with `--tlsClusterFile`. Use the `--tlsClusterPassword` option only if the certificate-key file is encrypted. In all cases, the `mongos` redacts the password from all logging and reporting output.

- On Linux/BSD, if the private key in the x.509 file is encrypted and you do not specify the `--tlsClusterPassword` option, MongoDB prompts for a passphrase. See TLS/SSL Certificate Passphrase.
- On macOS or Windows, if the private key in the x.509 file is encrypted, you must explicitly specify the `--tlsClusterPassword` option. Alternatively, you can either use a certificate from the secure system store (see `--tlsClusterCertificateSelector`) instead of a cluster PEM file or use an unencrypted PEM file.

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

`--tlsCAFile <filename>`

New in version 4.2.

Specifies 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.

On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM key file. See `--tlsCertificateSelector`. When using the secure store, you do not need to, but can, also specify the `--tlsCAFile`.

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

`--tlsClusterCAFile <filename>`

New in version 4.2.

Specifies 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.

If `--tlsClusterCAFile` 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 `--tlsCAFile` option.

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

On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM key file. See `--tlsClusterCertificateSelector`. When using the secure store, you do not need to, but can, also specify the `--tlsClusterCAFile`.

Requires that `--tlsCAFile` is set.

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

`--tlsCertificateSelector <parameter>=<value>`

New in version 4.2: Available on Windows and macOS as an alternative to `--tlsCertificateKeyFile`.

The `--tlsCertificateKeyFile` and `--tlsCertificateSelector` 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.

`--tlsCertificateSelector` accepts an argument of the format `<property>=<value>` where the property can be one of the following:

Property	Value type	Description
`subject`	ASCII string	Subject name or common name on certificate
`thumbprint`	hex string	A sequence of bytes, expressed as hexadecimal, used to identify a public key by its SHA-1 digest. The `thumbprint` is sometimes referred to as a `fingerprint`.

When using the system SSL certificate store, OCSP (Online Certificate Status Protocol) is used to validate the revocation status of certificates.

> **NOTE**
>
> You cannot use the `rotateCertificates` command or the `db.rotateCertificates()` shell method when using `net.tls.certificateSelector` or `--tlsCertificateSelector` set to `thumbprint`

`--tlsClusterCertificateSelector <parameter>=<value>`

New in version 4.2: Available on Windows and macOS as an alternative to `--tlsClusterFile`.

`--tlsClusterFile` and `--tlsClusterCertificateSelector` 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 use for internal authentication.

`--tlsClusterCertificateSelector` accepts an argument of the format `<property>=<value>` where the property can be one of the following:

Property	Value type	Description
`subject`	ASCII string	Subject name or common name on certificate
`thumbprint`	hex string	A sequence of bytes, expressed as hexadecimal, used to identify a public key by its SHA-1 digest. The `thumbprint` is sometimes referred to as a `fingerprint`.

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.

`--tlsCRLFile <filename>`

New in version 4.2.

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

> **NOTE**
>
> - You cannot specify a CRL file 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. See `--tlsCertificateSelector` in MongoDB 4.2+ to use the system SSL certificate store.
> - Starting in version 4.4, to check for certificate revocation, MongoDB `enables` the use of OCSP (Online Certificate Status Protocol) by default as an alternative to specifying a CRL file or using the system SSL certificate store.

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

`--tlsAllowConnectionsWithoutCertificates`

New in version 4.2.

For clients that don't provide certificates, `mongod` or `mongos` encrypts the TLS/SSL connection, assuming the connection is successfully made.

For clients that present a certificate, however, `mongos` performs certificate validation using the root certificate chain specified by `--tlsCAFile` and reject clients with invalid certificates.

Use the `--tlsAllowConnectionsWithoutCertificates` option if you have a mixed deployment that includes clients that do not or cannot present certificates to the `mongos`.

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

`--tlsAllowInvalidCertificates`

New in version 4.2.

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

> **NOTE**
>
> If you specify `--tlsAllowInvalidCertificates` or `tls.allowInvalidCertificates: true` when using x.509 authentication, an invalid certificate is only sufficient to establish a TLS connection but is *insufficient* for authentication.

When using the `--tlsAllowInvalidCertificates` setting, MongoDB logs a warning regarding the use of the invalid certificate.

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

`--tlsAllowInvalidHostnames`

New in version 4.2.

Disables the validation of the hostnames in TLS certificates, when connecting to other members of the replica set or sharded cluster for inter-process authentication. This allows `mongos` to connect to other members if the hostnames in their certificates do not match their configured hostname.

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

`--tlsDisabledProtocols <protocol(s)>`

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.

`--tlsDisabledProtocols` recognizes the following protocols: `TLS1_0`, `TLS1_1`, `TLS1_2`, and `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_1`.
- To list multiple protocols, specify as a comma separated list of protocols. For example `TLS1_0,TLS1_1`.

- Specifying an unrecognized protocol prevents the server from starting.
- The specified disabled protocols overrides any default disabled protocols.

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 `--tlsDisabledProtocols`. 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

`--tlsFIPSMode`

New in version 4.2.

Directs the `mongos` to use the FIPS mode of the TLS library. Your system must have a FIPS compliant library to use the `--tlsFIPSMode` option.

> **NOTE**
>
> FIPS-compatible TLS/SSL is available only in MongoDB Enterprise. See Configure MongoDB for FIPS for more information.

SSL Options (Deprecated)

> **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 MongoDB supports TLS 1.0 and later.

> **TIP**
>
> **See:**
> Configure `mongod` and `mongos` for TLS/SSL for full documentation of MongoDB's support.

`--sslOnNormalPorts`

Deprecated since version 2.6: Use `--tlsMode requireTLS` instead.

Enables TLS/SSL for `mongos`.

With `--sslOnNormalPorts`, a `mongos` requires TLS/SSL encryption for all connections on the default MongoDB port, or the port specified by `--port`. By default, `--sslOnNormalPorts` is disabled.

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

`--sslMode <mode>`

Deprecated since version 4.2: Use `--tlsMode` instead.

Enables TLS/SSL or mixed TLS/SSL used for all network connections. The argument to the `--sslMode` option can be one of the following:

Value	Description
`disabled`	The server does not use TLS/SSL.
`allowSSL`	Connections between servers do not use TLS/SSL. For incoming connections, the server accepts both TLS/SSL and non-TLS/non-SSL.
`preferSSL`	Connections between servers use TLS/SSL. For incoming connections, the server accepts both TLS/SSL and non-TLS/non-SSL.
`requireSSL`	The server uses and accepts only TLS/SSL encrypted connections.

If `--tlsCAFile`/`net.tls.CAFile` (or their aliases `--sslCAFile`/`net.ssl.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/SSL-enabled server.

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

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

`--sslPEMKeyFile <filename>`

Deprecated since version 4.2: Use `--tlsPEMKeyFile` instead.

> **NOTE**
>
> On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM file. See `--sslCertificateSelector`.

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

- On Linux/BSD, you must specify `--sslPEMKeyFile` when TLS/SSL is enabled.
- On Windows or macOS, you must specify either `--sslPEMKeyFile` or `--sslCertificateSelector` when TLS/SSL is enabled.

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

`--sslPEMKeyPassword <value>`

Deprecated since version 4.2: Use `--tlsPEMKeyPassword` instead.

Specifies the password to decrypt the certificate-key file (i.e. `--sslPEMKeyFile`). Use the `--sslPEMKeyPassword` option only if the certificate-key file is encrypted. In all cases, the `mongos` redacts the password from all logging and reporting output.

- On Linux/BSD, if the private key in the PEM file is encrypted and you do not specify the `--sslPEMKeyPassword` option, MongoDB prompts for a passphrase. See TLS/SSL Certificate Passphrase.
- On macOS or Windows, if the private key in the PEM file is encrypted, you must explicitly specify the `--sslPEMKeyPassword` option. Alternatively, you can use a certificate from the secure system store (see `--sslCertificateSelector`) instead of a PEM key file or use an unencrypted PEM file.

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

`--sslClusterFile <filename>`

Deprecated since version 4.2: Use `--tlsClusterFile` instead.

> **NOTE**
>
> On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM key file. See `--sslClusterCertificateSelector`.

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

If `--sslClusterFile` does not specify the `.pem` file for internal cluster authentication or the alternative `--sslClusterCertificateSelector`, the cluster uses the `.pem` file specified in the `--sslPEMKeyFile` option or the certificate returned by the `--sslCertificateSelector`.

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

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

`--sslClusterPassword <value>`

Deprecated since version 4.2: Use `--tlsClusterPassword` instead.

Specifies the password to decrypt the x.509 certificate-key file specified with `--sslClusterFile`. Use the `--sslClusterPassword` option only if the certificate-key file is encrypted. In all cases, the `mongos` redacts the password from all logging and reporting output.

- On Linux/BSD, if the private key in the x.509 file is encrypted and you do not specify the `--sslClusterPassword` option, MongoDB prompts for a passphrase. See TLS/SSL Certificate Passphrase.
- On macOS or Windows, if the private key in the x.509 file is encrypted, you must explicitly specify the `--sslClusterPassword` option. Alternatively, you can either use a certificate from the secure system store (see `--sslClusterCertificateSelector`) instead of a cluster PEM file or use an unencrypted PEM file.

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

`--sslCAFile <filename>`

Deprecated since version 4.2: Use `--tlsCAFile` instead.

Specifies 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.

On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM key file. See `--sslCertificateSelector`. When using the secure store, you do not need to, but can, also specify the `--sslCAFile`.

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

`--sslClusterCAFile <filename>`

Deprecated since version 4.2: Use `--tlsClusterCAFile` instead.

Specifies 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.

If `--sslClusterCAFile` 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 `--sslCAFile` option.

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

On macOS or Windows, you can use a certificate from the operating system's secure store instead of a PEM key file. See `--sslClusterCertificateSelector`. When using the secure store, you do not need to, but can, also specify the `--sslClusterCAFile`.

Requires that `--sslCAFile` is set.

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

`--sslCertificateSelector <parameter>=<value>`

Deprecated since version 4.2: Use `--tlsCertificateSelector` instead.

Available on Windows and macOS as an alternative to `--tlsCertificateKeyFile`.

`--tlsCertificateKeyFile` and `--sslCertificateSelector` 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.

`--sslCertificateSelector` accepts an argument of the format `<property>=<value>` where the property can be one of the following:

Property	Value type	Description
`subject`	ASCII string	Subject name or common name on certificate
`thumbprint`	hex string	A sequence of bytes, expressed as hexadecimal, used to identify a public key by its SHA-1 digest. The `thumbprint` is sometimes referred to as a `fingerprint`.

When using the system SSL certificate store, OCSP (Online Certificate Status Protocol) is used to validate the revocation status of certificates.

`--sslClusterCertificateSelector <parameter>=<value>`

Deprecated since version 4.2: Use `--tlsClusterCertificateSelector` instead.

Available on Windows and macOS as an alternative to `--sslClusterFile`.

`--sslClusterFile` and `--sslClusterCertificateSelector` 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 use for internal authentication.

`--sslClusterCertificateSelector` accepts an argument of the format `<property>=<value>` where the property can be one of the following:

Property	Value type	Description
`subject`	ASCII string	Subject name or common name on certificate
`thumbprint`	hex string	A sequence of bytes, expressed as hexadecimal, used to identify a public key by its SHA-1 digest. The `thumbprint` is sometimes referred to as a `fingerprint`.

`--sslCRLFile <filename>`

Deprecated since version 4.2: Use `--tlsCRLFile` instead.

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

NOTE

- You cannot specify a CRL file 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. See `--tlsCertificateSelector` in MongoDB 4.2+ to use the system SSL certificate store.

- Starting in version 4.4, to check for certificate revocation, MongoDB `enables` the use of OCSP (Online Certificate Status Protocol) by default as an alternative to specifying a CRL file or using the system SSL certificate store.

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

`--sslAllowConnectionsWithoutCertificates`

Deprecated since version 4.2: Use `--tlsAllowConnectionsWithoutCertificates` instead.

For clients that don't provide certificates, `mongod` or `mongos` encrypts the TLS/SSL connection, assuming the connection is successfully made.

For clients that present a certificate, however, `mongos` performs certificate validation using the root certificate chain specified by `--sslCAFile` and reject clients with invalid certificates.

Use the `--sslAllowConnectionsWithoutCertificates` option if you have a mixed deployment that includes clients that do not or cannot present certificates to the `mongos`.

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

`--sslAllowInvalidCertificates`

Deprecated since version 4.2: Use `--tlsAllowInvalidCertificates` instead.

Bypasses the validation checks for TLS/SSL certificates on other servers in the cluster and allows the use of invalid certificates to connect.

NOTE

Starting in MongoDB 4.0, if you specify any of the following x.509 authentication options, an invalid certificate is sufficient only to establish a TLS connection but it is *insufficient* for authentication:

- `--sslAllowInvalidCertificates` or `net.ssl.allowInvalidCertificates: true` for MongoDB 4.0 and later
- `--tlsAllowInvalidCertificates` or `net.tls.allowInvalidCertificates: true` for MongoDB 4.2 and later

When using the `--sslAllowInvalidCertificates` setting, MongoDB logs a warning regarding the use of the invalid certificate.

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

`--sslAllowInvalidHostnames`

Deprecated since version 4.2: Use `--tlsAllowInvalidHostnames` instead.

Disables the validation of the hostnames in TLS/SSL certificates, when connecting to other members of the replica set or sharded cluster for inter-process authentication. This allows `mongos` to connect to other members if the hostnames in their certificates do not match their configured hostname.

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

`--sslDisabledProtocols <protocol(s)>`

Deprecated since version 4.2: Use `--tlsDisabledProtocols` 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.

`--sslDisabledProtocols` recognizes the following protocols: `TLS1_0`, `TLS1_1`, `TLS1_2`, and `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_1`.
- To list multiple protocols, specify as a comma separated list of protocols. For example `TLS1_0,TLS1_1`.
- Specifying an unrecognized protocol prevents the server from starting.
- The specified disabled protocols overrides any default disabled protocols.

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 `--sslDisabledProtocols`. 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

`--sslFIPSMode`

Deprecated since version 4.2: Use `--tlsFIPSMode` instead.

Directs the `mongos` to use the FIPS mode of the TLS/SSL library. Your system must have a FIPS compliant library to use the `--sslFIPSMode` option.

NOTE

FIPS-compatible TLS/SSL is available only in MongoDB Enterprise. See Configure MongoDB for FIPS for more information.

Audit Options

`--auditCompressionMode`

New in version 5.3.

Specifies the compression mode for audit log encryption. You must also enable audit log encryption using either `--auditEncryptionKeyUID` or `--auditLocalKeyFile`.

`--auditCompressionMode` can be set to one of these values:

Value	Description
`zstd`	Use the zstd algorithm to compress the audit log.
`none` *(default)*	Do not compress the audit log.

NOTE

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

`--auditDestination`

Enables auditing and specifies where `mongos` sends all audit events.

`--auditDestination` can have one of the following values:

Value	Description
`syslog`	Output the audit events to syslog in JSON format. Not available on Windows. Audit messages have a syslog severity level of `info` and a facility level of `user`. The syslog message limit can result in the truncation of audit messages. The auditing system neither detects the truncation nor errors upon its occurrence.
`console`	Output the audit events to `stdout` in JSON format.
`file`	Output the audit events to the file specified in `--auditPath` in the format specified in `--auditFormat`.

NOTE

Available only in MongoDB Enterprise and MongoDB Atlas.

`--auditEncryptionKeyUID`

New in version 6.0.

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

You cannot use `--auditEncryptionKeyUID` and `--auditLocalKeyFile` together.

NOTE

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

`--auditFormat`

Specifies the format of the output file for auditing if `--auditDestination` is `file`. The `--auditFormat` option can have one of the following values:

Value	Description
`JSON`	Output the audit events in JSON format to the file specified in `--auditPath`.
`BSON`	Output the audit events in BSON binary format to the file specified in `--auditPath`.

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 Atlas.

`--auditLocalKeyFile`

New in version 5.3.

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

> **NOTE**
>
> Only use `--auditLocalKeyFile` for testing because the key is not secured. To secure the key, use `--auditEncryptionKeyUID` and an external Key Management Interoperability Protocol (KMIP) server.

You cannot use `--auditLocalKeyFile` and `--auditEncryptionKeyUID` together.

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

`--auditPath`

Specifies the output file for auditing if `--auditDestination` has value of `file`. The `--auditPath` option can take either a full path name or a relative path name.

> **NOTE**
>
> Available only in MongoDB Enterprise and MongoDB Atlas.

`--auditFilter`

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

```
{ <field1>: <expression1>, ... }
```

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, you must use the YAML format of the configuration file.

Share Feedback

> **NOTE**
>
> Available only in MongoDB Enterprise and MongoDB Atlas.

Profiler Options

`--slowms <integer>`

Default: 100

The *slow* operation time threshold, in milliseconds. 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 `slowOpSampleRate`.

At higher `logLevel` settings, all operations appear in the diagnostic log regardless of their latency.

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

`--slowOpSampleRate <double>`

Default: 1.0

The fraction of *slow* operations that should be logged. `--slowOpSampleRate` accepts values between 0 and 1, inclusive.

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

LDAP Authentication and Authorization Options

`--ldapServers <host1>:<port>,<host2>:<port>,...,<hostN>:<port>`

Available in MongoDB Enterprise only.

The LDAP server against which the `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 partitions the LDAP directory over multiple LDAP servers, specify *one* LDAP server or any of its replicated instances to `--ldapServers`. MongoDB supports following LDAP referrals as defined in RFC 4511 4.1.10. Do not use `--ldapServers` for listing every LDAP server in your infrastructure.

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

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

`--ldapValidateLDAPServerConfig <boolean>`

Available in MongoDB Enterprise

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

- If `true`, the `mongos` instance performs the availability check and only continues to start up if the LDAP server is available.
- If `false`, the `mongos` instance skips the availability check; i.e. the instance starts up even if the LDAP server is unavailable.

`--ldapQueryUser <string>`

Available in MongoDB Enterprise only.

The identity with which `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 `username transformation`.
- The LDAP server disallows anonymous binds

You must use `--ldapQueryUser` with `--ldapQueryPassword`.

If unset, `mongos` doesn't attempt to bind to the LDAP server.

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

> **NOTE**
>
> Windows MongoDB deployments can use `--ldapBindWithOSDefaults` instead of `--ldapQueryUser` and `--ldapQueryPassword`. You cannot specify both `--ldapQueryUser` and `--ldapBindWithOSDefaults` at the same time.

`--ldapQueryPassword <string>`

Available in MongoDB Enterprise only.

The password used to bind to an LDAP server when using `--ldapQueryUser`. You must use `--ldapQueryPassword` with `--ldapQueryUser`.

If unset, `mongos` doesn't attempt to bind to the LDAP server.

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

> **NOTE**
>
> Windows MongoDB deployments can use `--ldapBindWithOSDefaults` instead of `--ldapQueryPassword` and `--ldapQueryPassword`. You cannot specify both `--ldapQueryPassword` and `--ldapBindWithOSDefaults` at the same time.

`--ldapBindWithOSDefaults <bool>`

Default: false

Available in MongoDB Enterprise for the Windows platform only.

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

Only required if:

- Using LDAP authorization.
- Using an LDAP query for `username transformation`.
- The LDAP server disallows anonymous binds

Use `--ldapBindWithOSDefaults` to replace `--ldapQueryUser` and `--ldapQueryPassword`.

`--ldapBindMethod <string>`

Default: simple

Available in MongoDB Enterprise only.

The method `mongos` uses to authenticate to an LDAP server. Use with `--ldapQueryUser` and `--ldapQueryPassword` to connect to the LDAP server.

`--ldapBindMethod` supports the following values:

- `simple` - `mongos` uses simple authentication.
- `sasl` - `mongos` uses SASL protocol for authentication

If you specify `sasl`, you can configure the available SASL mechanisms using `--ldapBindSaslMechanisms`. `mongos` defaults to using `DIGEST-MD5` mechanism.

`--ldapBindSaslMechanisms <string>`

Default: DIGEST-MD5

Available in MongoDB Enterprise only.

A comma-separated list of SASL mechanisms `mongos` can use when authenticating to the LDAP server. The `mongos` and the LDAP server must agree on at least one mechanism. The `mongos` dynamically loads any SASL mechanism libraries installed on the host machine at runtime.

Install and configure the appropriate libraries for the selected SASL mechanism(s) on both the `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 `mongos` host machine:

`Linux`

- The `KRB5_CLIENT_KTNAME` 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 documentation.
- The client keytab includes a User Principal for the `mongos` to use when connecting 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 `--ldapBindWithOSDefaults` to `true` to allow `mongos` to use the generated credentials when connecting to the Active Directory server and execute queries.

Set `--ldapBindMethod` 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 mechanisms compatible with the service.

MongoDB is not a source of SASL mechanism libraries, nor is the MongoDB documentation a definitive source for installing or configuring any given SASL mechanism. For documentation and support, defer to the SASL 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.

`--ldapTransportSecurity <string>`

Default: tls

Available in MongoDB Enterprise only.

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

For Linux deployments, you must configure the appropriate TLS Options in `/etc/openldap/ldap.conf` 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 ldap.conf 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 `--ldapTransportSecurity` to `none` to disable TLS/SSL between `mongos` and the LDAP server.

> **WARNING**
>
> Setting `--ldapTransportSecurity` to `none` transmits plaintext information and possibly credentials between `mongos` and the LDAP server.

`--ldapTimeoutMS <int>`

Default: 10000

Available in MongoDB Enterprise only.

The amount of time in milliseconds `mongos` should wait for an LDAP server to respond to a request.

Increasing the value of `--ldapTimeoutMS` 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 `--ldapTimeoutMS` reduces the time MongoDB waits for a response from the LDAP server.

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

`--ldapRetryCount <int>`

New in version 6.1.

Default: 0

Available in MongoDB Enterprise only.

Number of operation retries by the server LDAP manager after a network error.

`--ldapUserToDNMapping <string>`

Available in MongoDB Enterprise only.

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

- Performing LDAP authentication with simple LDAP binding, where users authenticate to MongoDB with usernames that are not full LDAP DNs.
- Using an `LDAP authorization query template` that requires a DN.
- Transforming the usernames of clients authenticating to Mongo DB using different authentication mechanisms, such as x.509 or kerberos, to a full LDAP DN for authorization.

`--ldapUserToDNMapping` expects a quote-enclosed JSON-string representing an ordered array of 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: "<regex>"
  substitution: "<LDAP DN>" | ldapQuery: "<LDAP Query>"
}
```

Field	Description	Example
`match`	An ECMAScript-formatted regular expression (regex) to match against a provided username. Each parenthesis-enclosed section represents a regex capture group used by `substitution` or `ldapQuery`.	`"(.+)ENGINEERING"` `"(.+)DBA"`
`substitution`	An LDAP distinguished name (DN) formatting template that converts the authentication name matched by the `match` regex into a LDAP DN. Each curly bracket-enclosed numeric value is replaced by the corresponding regex capture group extracted from the authentication username via the `match` regex. The result of the substitution must be an RFC4514 escaped string.	`"cn={0},ou=engineering, dc=example,dc=com"`

Field	Description	Example
`ldapQuery`	A LDAP query formatting template that inserts the authentication name matched by the `match` regex into an LDAP query URI encoded respecting RFC4515 and RFC4516. Each curly bracket-enclosed numeric value is replaced by the corresponding regex capture group extracted from the authentication username via the `match` expression. `mongos` executes the query against the LDAP server to retrieve the LDAP DN for the authenticated user. `mongos` requires exactly one returned result for the transformation to be successful, or `mongos` skips this transformation.	`"ou=engineering,dc=example, dc=com??one?(user={0})"`

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 use either `substitution` or `ldapQuery`. You *cannot* specify both in the same document.

When performing authentication or authorization, `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, `mongos` applies the transformation and uses the output for authenticating the user. `mongos` does not check the remaining documents in the array.

If the given document does not match the provided authentication name, `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, `mongos` returns an error.

Starting in MongoDB 4.4, `mongos` also returns an error if one of the transformations cannot be evaluated due to networking or authentication failures to the LDAP server. `mongos` rejects the connection request and does not check the remaining documents in the array.

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

EXAMPLE

The following shows two transformation documents. The first document matches against any string ending in `@ENGINEERING`, placing anything preceeding the suffix into a regex capture group. The second document matches against any string ending in `@DBA`, placing anything preceeding the suffix into a regex capture group.

IMPORTANT

You must pass the array to --ldapUserToDNMapping as a string.

Share Feedback

```
"[
   {
      match: "(.+)@ENGINEERING.EXAMPLE.COM",
      substitution: "cn={0},ou=engineering,dc=example,dc=com"
   },
   {
      match: "(.+)@DBA.EXAMPLE.COM",
      ldapQuery: "ou=dba,dc=example,dc=com??one?(user={0})"

   }

]"
```

A user with username `alice@ENGINEERING.EXAMPLE.COM` matches the first document. The regex capture group `{0}` corresponds to the string `alice`. The resulting output is the DN `"cn=alice,ou=engineering,dc=example,dc=com"`.

A user with username `bob@DBA.EXAMPLE.COM` matches the second document. The regex capture group `{0}` corresponds to the string `bob`. The resulting output is the LDAP query `"ou=dba,dc=example,dc=com??one?(user=bob)"`. `mongos` executes this

query against the LDAP server, returning the result `"cn=bob,ou=dba,dc=example,dc=com"`.

If `--ldapUserToDNMapping` is unset, `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 `mongos` using the `setParameter` database command.

Additional Options

`--ipv6`

Enables IPv6 support. `mongos` disables IPv6 support by default.

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

- Configure `--bind_ip` with one or more IPv6 addresses or hostnames that resolve to IPv6 addresses, **or**
- Set `--bind_ip_all` to `true`.