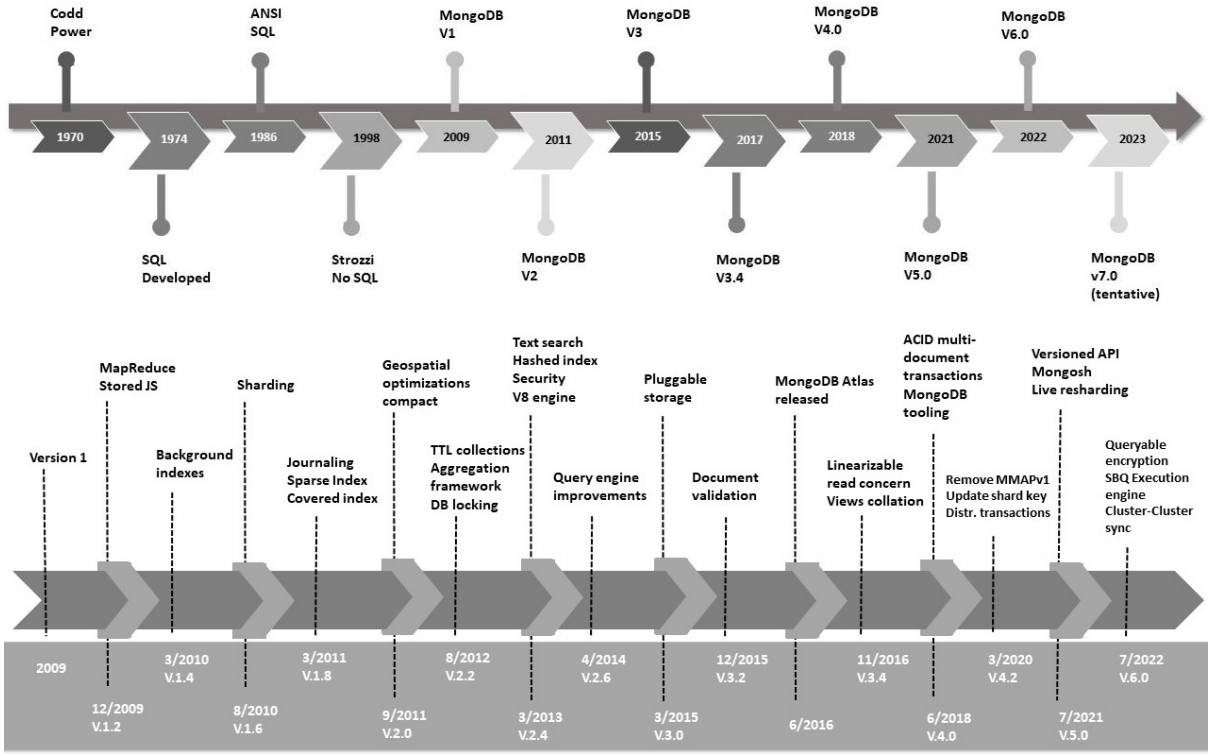
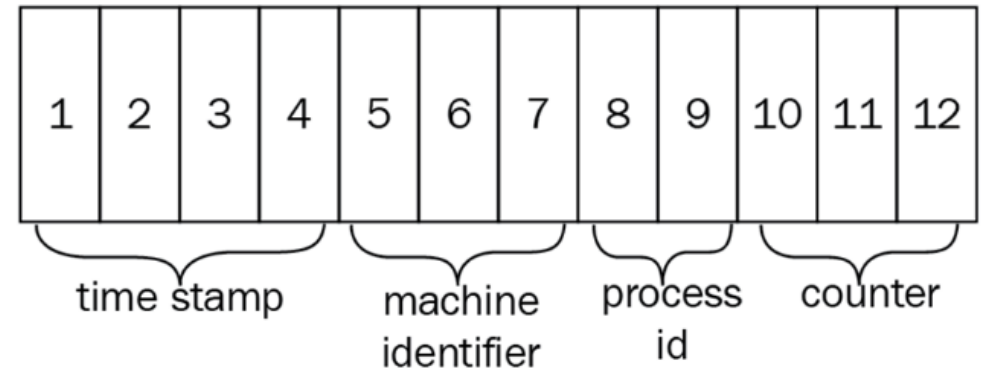
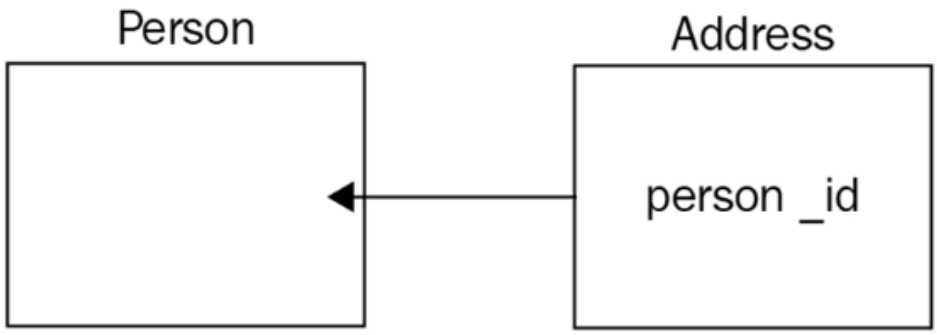
Chapter 1: MongoDB – A Database for the Modern Web

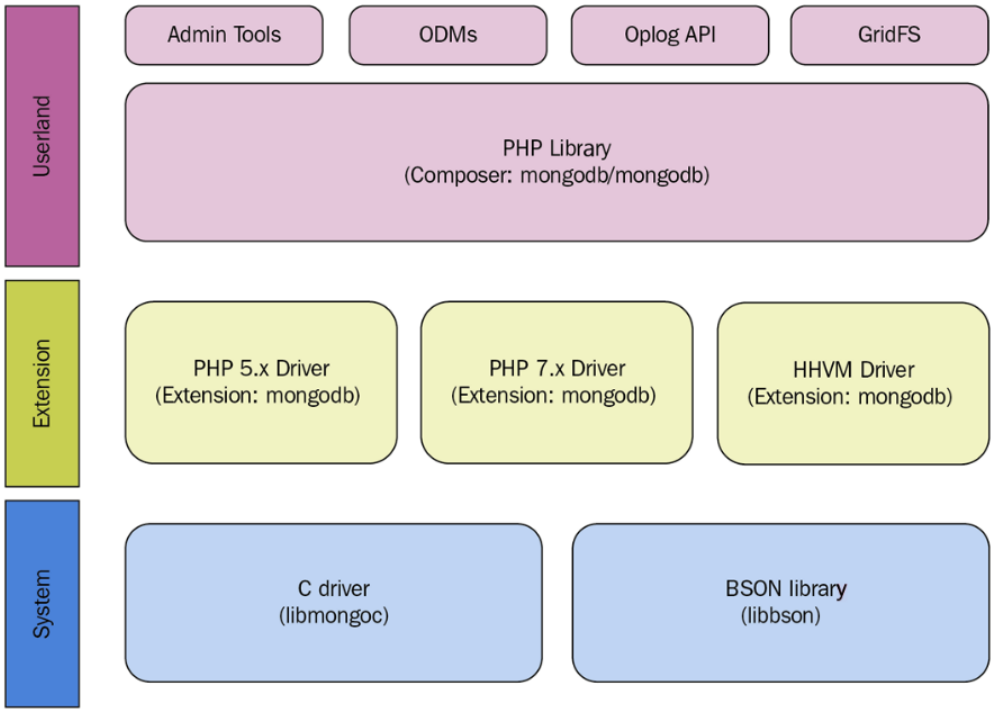


SQL	MongoDB
Database	Database
Table	Collection
Index	Index
Row	Document
Column	Field
Joins	Embed in a document or link via `DBRef`
CREATE TABLE employee (name VARCHAR(100))	db.createCollection("employee")
INSERT INTO employees VALUES (Alex, 36)	db.employees.insert({name: "Alex", age: 36})
SELECT * FROM employees	db.employees.find()
SELECT * FROM employees LIMIT 1	db.employees.findOne()
SELECT DISTINCT name FROM employees	db.employees.distinct("name")
UPDATE employees SET age = 37 WHERE name = 'Alex'	db.employees.update({name: "Alex"}, {$set: {age: 37}}, {multi: true})
DELETE FROM employees WHERE name = 'Alex'	db.employees.remove({name: "Alex"})
CREATE INDEX ON employees (name ASC)	db.employees.ensureIndex({name: 1})

Chapter 2: Schema Design and Data Modeling







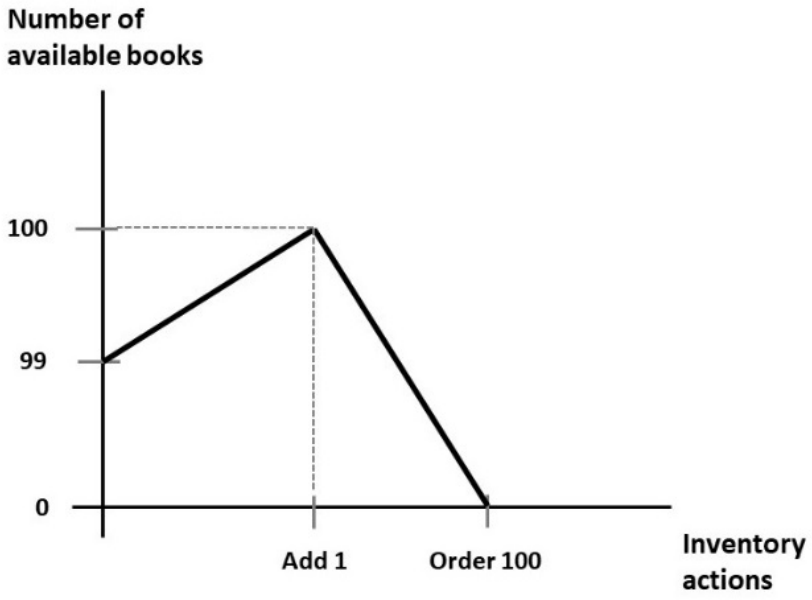
Type	Number	Alias	Notes
Double	1	`double`	
String	2	`string`	
Object	3	`object`	
Array	4	`array`	
Binary data	5	`binData`	
ObjectID	7	`objectId`	
Boolean	8	`bool`	
Date	9	`date`	
Null	10	`null`	
Regular expression	11	`regex`	
JavaScript	13	`javascript`	
32-bit integer	16	`int`	
Timestamp	17	`timestamp`	
64-bit integer	18	`long`	
Decimal128	19	`decimal`	New in version 3.4
Min. key	-1	`minKey`	
Max. key	127	`maxKey`	
Undefined	6	`undefined`	Deprecated
DBPointer	12	`dbPointer`	Deprecated
Symbol	14	`symbol`	Deprecated
JavaScript (with scope)	15	`javascriptWithScope`	Deprecated

Option value	Description
`Database`	The database name.
`Hosts`	Our database hosts.
`Write/w`	The write concern (default is 1).
`Auth_mech`	Authentication mechanism. Valid options are: `:scram`, `:mongodb_cr`, `:mongodb_x509`, and `:plain`. The default option on 3.0 is `:scram`, whereas the default on 2.4 and 2.6 is `:plain`.

Option value	Description
`Auth_source`	The authentication source for our authentication mechanism.
`Min_pool_size/max_pool_size`	Minimum and maximum pool size for connections.
`SSL`, `ssl_cert`, `ssl_key`, `ssl_key_pass_phrase`, `ssl_verify`	A set of options regarding SSL connections to the database.
`Include_root_in_json`	Includes the root model name in JSON serialization.
`Include_type_for_serialization`	Includes the `_type` field when serializing MongoDB objects.
`Use_activesupport_time_zone`	Uses active support's time zone when converting timestamps between server and client.

Name	Description
`minPoolSize/maxPoolSize`	Minimum and maximum pool size for connections.
`w`	Write concern option.
`wtimeoutMS`	Timeout for write concern operations.
`Journal`	Journal options.
`readPreference`	Read preference to be used for replica sets. Available options are: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`.
`maxStalenessSeconds`	Specifies in seconds how stale (data lagging behind the primary) a secondary can be before the client stops using it for read operations.
`SSL`	Using SSL to connect to the database.
`authSource`	Used in conjunction with username, this specifies the database associated with the user's credentials. When we use external authentication mechanisms, this should be `$external` for LDAP or Kerberos.
`authMechanism`	An authentication mechanism can be used for connections. Available options for MongoDB are **SCRAM-SHA-1**, **MONGODB-CR**, **MONGODB-X.509**. MongoDB Enterprise (the paid version) offers two more options: **GSSAPI** (Kerberos), **PLAIN** (**LDAP SASL**)

Chapter 3: MongoDB CRUD Operations

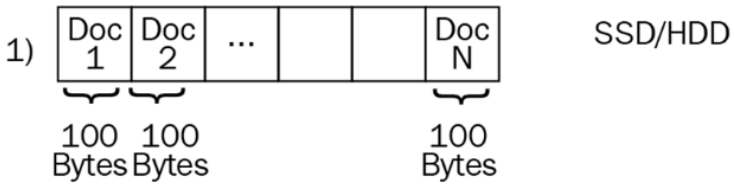


1) 1004 inserts → 1) 1000 inserts
2) 4 inserts

2) 998 updates → 3) 998 updates

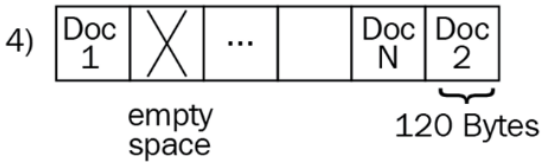
3) 1004 deletes → 4) 1000 deletes

4) 5 inserts → 5) 4 deletes
6) 5 inserts



2) DOC 2 size=90 Bytes of data,10B padding

3) Update Doc 2. New size 110 Bytes

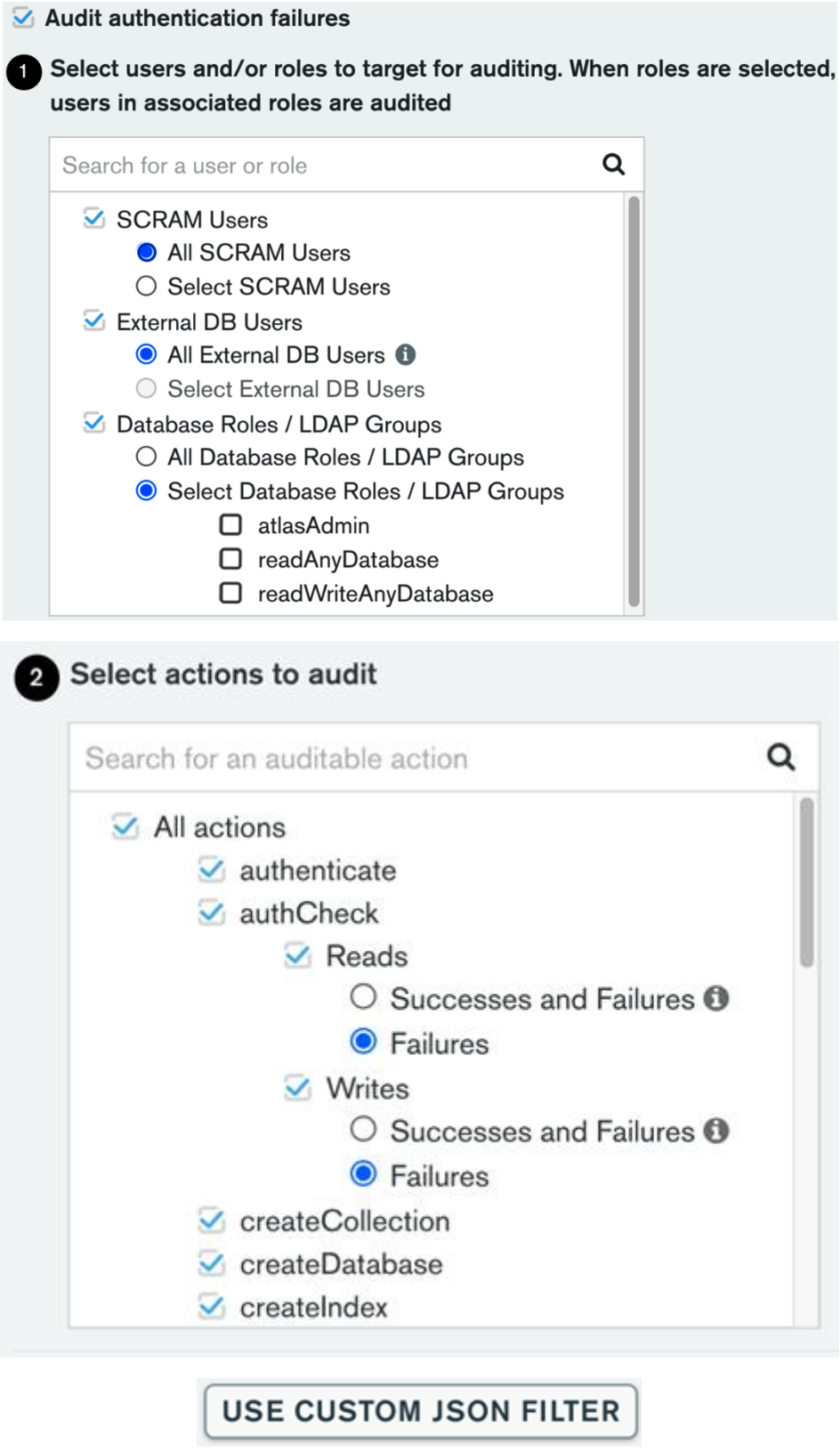


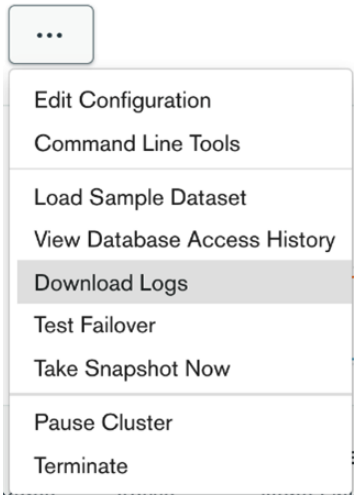
Shell helpers	JavaScript equivalents
`show dbs, show databases`	`db.adminCommand('listDatabases')`
`use <database_name>`	`db = db.getSiblingDB('<database_name>')`
`show collections`	`db.getCollectionNames()`
`show users`	`db.getUsers()`
`show roles`	`db.getRoles({showBuiltinRoles: true})`
`show log <logname>`	`db.adminCommand({ 'getLog' : '<logname>' })`
`show logs`	`db.adminCommand({ 'getLog' : '*' })`

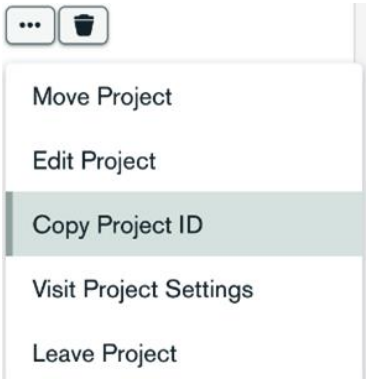
`it`	`cur = db.collection_name_here.find()`
	`if ( cur.hasNext() ){`
	`cur.next();`
	`}`

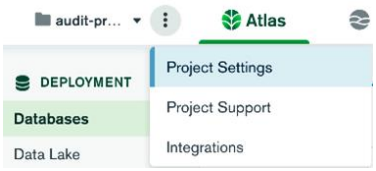
SQL	Aggregation framework
WHERE/HAVING	$match
GROUP BY	$group
SELECT	$project
ORDER BY	$sort
LIMIT	$limit
sum()/count()	$sum
Join	$lookup

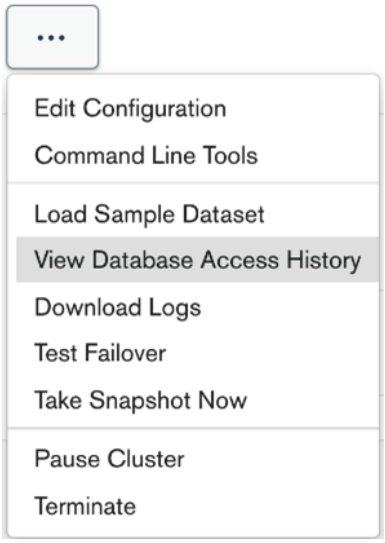
Chapter 4: Auditing



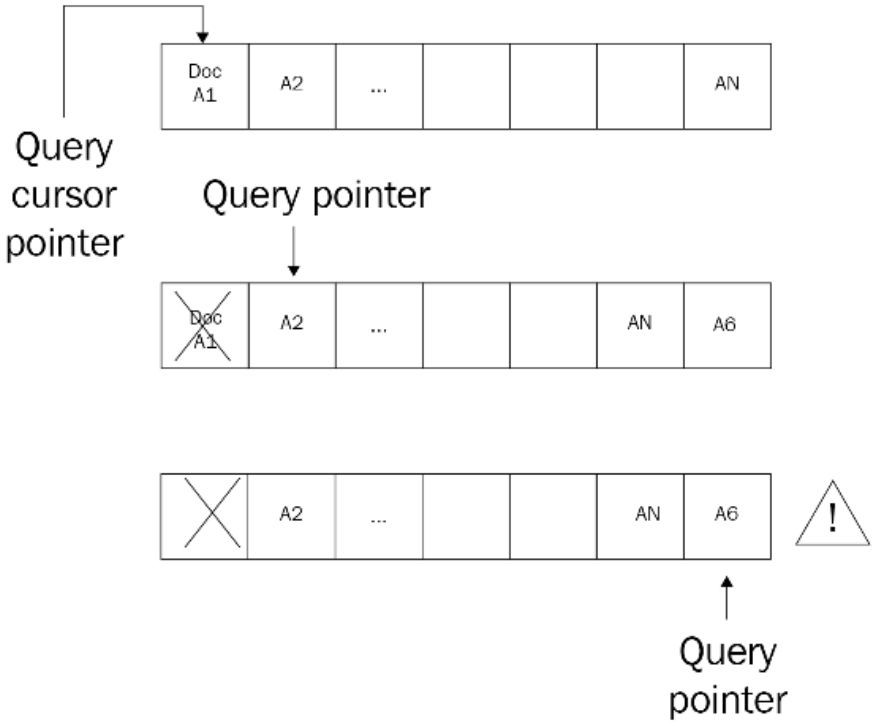

Edit Configuration
Command Line Tools
Load Sample Dataset
View Database Access History
Download Logs
Test Failover
Take Snapshot Now
Pause Cluster
Terminate


Move Project
Edit Project
Copy Project ID
Visit Project Settings
Leave Project


audit-pr...
Atlas
DEPLOYMENT
Databases
Data Lake
Project Settings
Project Support
Integrations


Edit Configuration
Command Line Tools
Load Sample Dataset
View Database Access History
Download Logs
Test Failover
Take Snapshot Now
Pause Cluster
Terminate

Chapter 5: Advanced Querying



Option	Description
`allow_partial_results`	This is for use with sharded clusters. If a shard is down, it allows the query to return results from the shards that are up, potentially getting only a portion of the results.
`batch_size(Integer)`	This can change the batch size that the cursor will fetch from MongoDB. This is done on each `GETMORE` operation (for example, by typing it on the mongo shell).
`comment(String)`	With this command, we can add a comment in our query for documentation reasons.
`hint(Hash)`	We can force usage of an index using `hint()`.
`limit(Integer)`	We can limit the result set to the number of documents specified by `Integer`.
`max_scan(Integer)`	We can limit the number of documents that will be scanned. This will return incomplete results and is useful if we are performing operations where we want to guarantee that they won't take a long time, such as when we connect to our production database.
`no_cursor_timeout`	If we don't specify this parameter, MongoDB will close any inactive cursor after 600 seconds. With this parameter, our cursor will never be closed.
`projection(Hash)`	We can use this parameter to fetch or exclude specific attributes from our results. This will reduce data transfer over the wire. An example of this is `client[:books].find.projection(:price => 1)`.
`read(Hash)`	We can specify a read preference to be applied only for this query: `client[:books].find.read(:mode => :secondary_preferred)`.
`show_disk_loc(Boolean)`	We should use this option if we want to find the actual location of our results on a disk.
`skip(Integer)`	This can be used to skip the specified number of documents. It's useful for paginating results.
`snapshot`	This can be used to execute our query in snapshot mode. This is useful when we want a more stringent consistency.
`sort(Hash)`	We can use this to sort our results; for example, `client[:books].find.sort(:name => -1)`.

Name	Description
`$eq`	Will match with all test values that are equal to the target value
`$gt`	Will match with all test values that are greater than the target value
`$gte`	Will match with all test values that are greater than or equal to the target value
`$lt`	Will match with all test values that are less than the target value
`$lte`	Will match with all test values that are less than or equal to the target value
`$ne`	Will match with all test values that are NOT equal to the target value
`$in`	Will match the target value with ANY of the test values in the array
`$nin`	Will NOT match the target value with ANY of the test values in the array

Name	Description
`$inc`	Will increment the value of the target field by the specified value.
`$mul`	Will multiply the value of the target field by the specified value.
`$rename`	Will rename a field.
`$setOnInsert`	Will set the value of the target field if an update results in an insert of a document. Will not modify an existing document or apply to any update operations.
`$set`	Will set the value of the target field to our specified value.
`$unset`	Will remove the target field from the document.
`$min`	Will update the target field to our specified value if the target value is less than the existing one.
`$max`	Will update the target field to our specified value if the target value is greater than the existing one.
`$currentDate`	Will set the value of the target field to the present date. We can define the value as a timestamp (Unix epoch time) or a date.

Option	Description
`i`	This option queries case insensitivity.
`m`	This option only applies for multiline strings with anchors (`^` for the start and `$` for the end). In this case, defining the `m` option will match the pattern at the beginning or end of each line. Without the `m` option, the anchors will match at the beginning or end of the string.

Option	Description
`x`	This option will ignore all whitespace characters in the `$regex` pattern. It will also ignore any content between and including an unescaped hash or pound character and the next newline. This can be used to include comments. It will not have any effect if the characters are included in a character class or escaped otherwise. It will not have any effect on handling the VT character.
`s`	This option allows the dot character (that is, `.`) to match all characters, including newline characters.

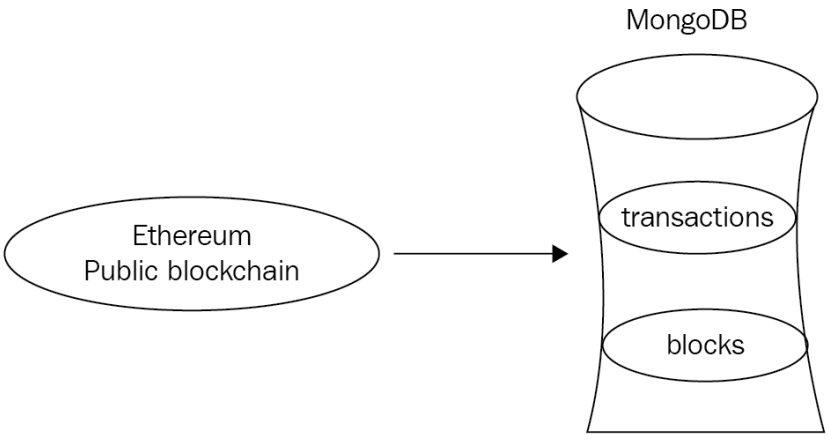
`fullDocument`	This is the new state of the document, which can include the following: If it's a delete operation, this field is omitted as the document no longer exists. If it's an insert or replace operation, this will be the new value of the document. If it's an update operation and we have enabled `'updateLookup'`, then it will have the most recently major-committed version of the document modified by the update operation.
`operationType`	This is the type of operation; it can be either `insert`, `delete`, `replace`, `update`, or `invalidate`.
`documentKey`	This is the ID (`ObjectID`) of the document that was modified by the operation.
`updateDescription.updatedFields / removedFields`	This is a document or an array of keys showing the data that was updated or removed by the update or remove operation, respectively.
`txnNumber`	The transaction ID number when the operation is part of a multi-document atomicity, consistency, isolation, durability (ACID) transaction.
`lsid`	This is the session ID when the operation is part of a multi-document ACID transaction.

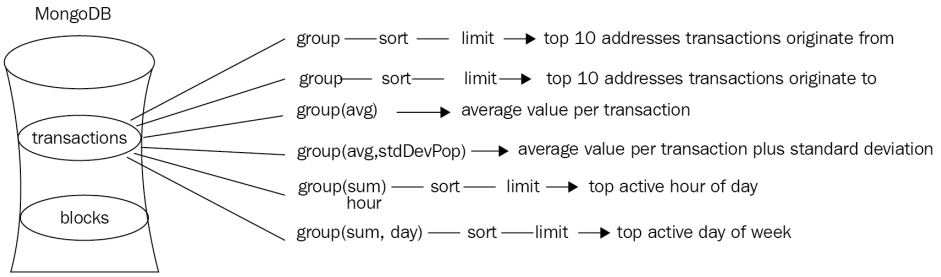
Chapter 6: Multi-Document ACID Transactions

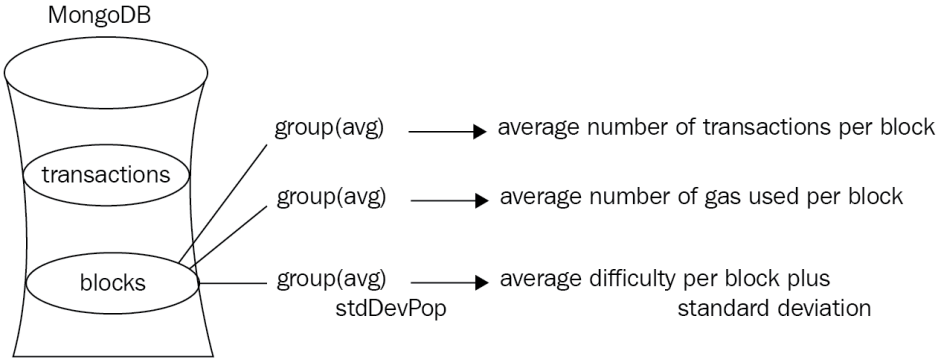
Isolation Level	Lost Updates	Dirty Reads	Non-Repeatable Reads	Phantoms
Read uncommitted	Don't occur	May occur	May occur	May occur
Read committed	Don't occur	Don't occur	May occur	May occur
Repeatable read	Don't occur	Don't occur	Don't occur	May occur
Serializable	Don't occur	Don't occur	Don't occur	Don't occur

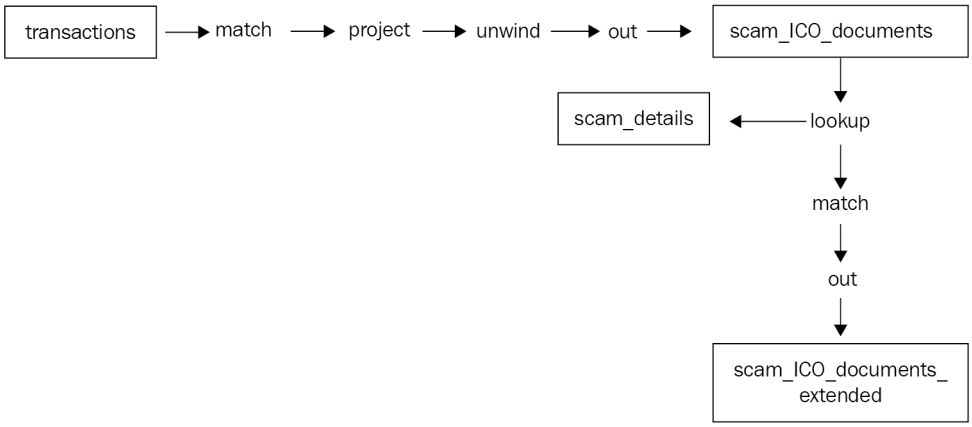
Keyword	Validates on Type	Description
`enum`	All	The enum of allowed values in a field
`type`	All	The enum of allowed types in a field
`minimum/maximum`	Numeric	The minimum and maximum values for a numeric field
`minLength/maxLength`	String	The minimum and maximum lengths allowed for a string field
`pattern`	String	The regex pattern that the string field must match
`required`	Objects	The document must contain all the strings defined in the required property array
`minItems/maxItems`	Arrays	The minimum and maximum lengths of items in the array
`uniqueItems`	Arrays	If set to true, all items in the array must have unique values
`title`	N/A	A descriptive title for the developer's use
`description`	N/A	A description of the developer's use

Chapter 7: Aggregation

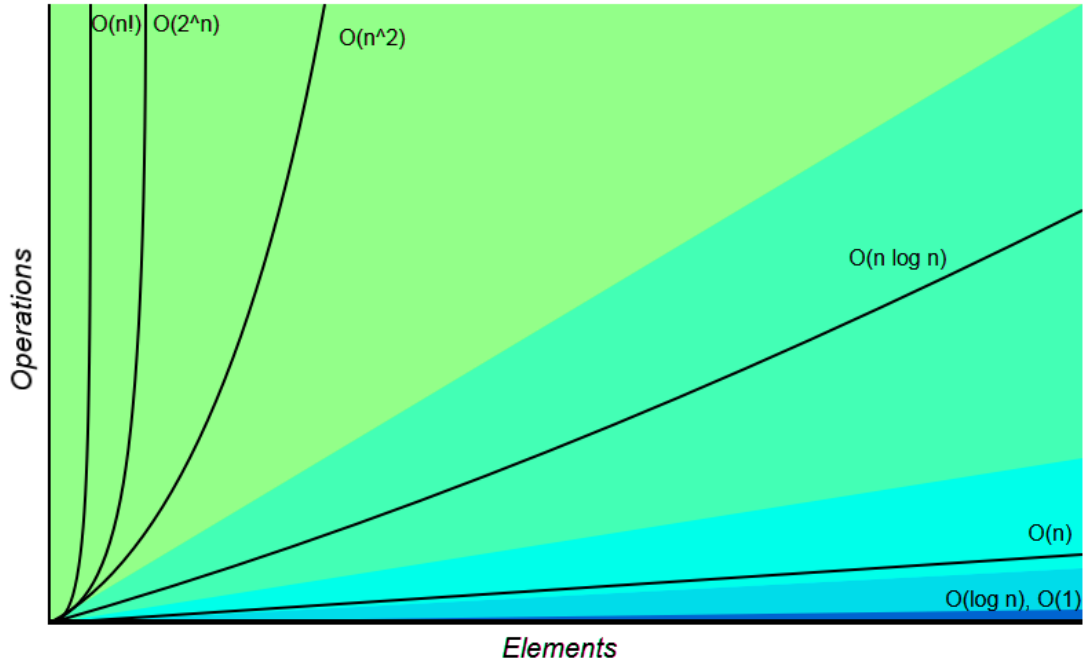


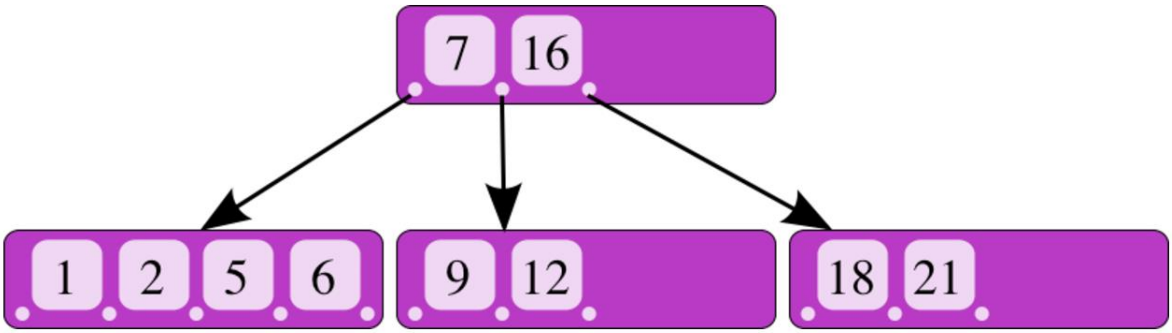


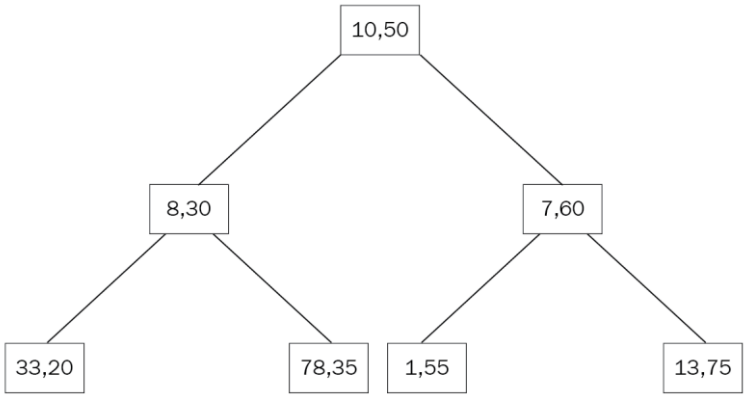




Chapter 8: Indexing

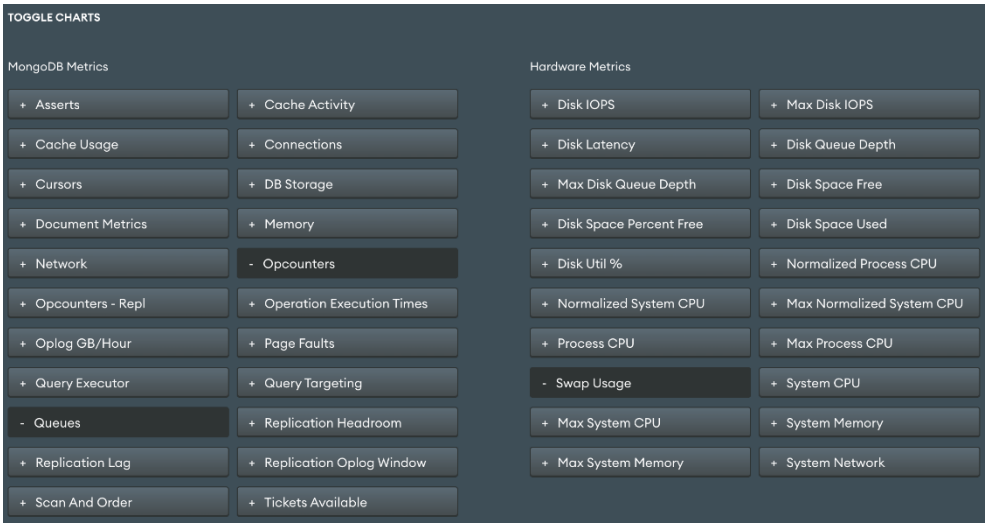




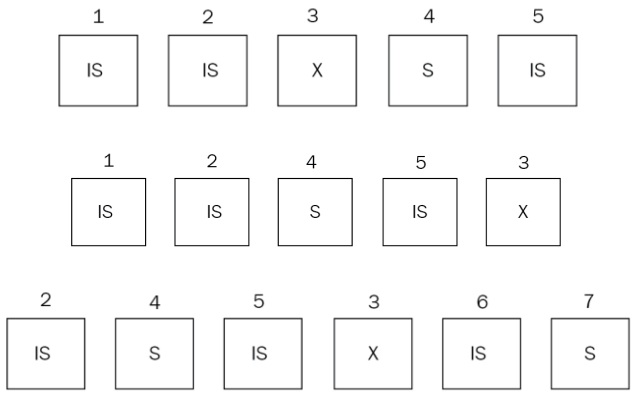


Strength Value	Description
`1`	The primary level of comparison. Comparison based on string values, ignoring any other differences, such as case and diacritics.
`2`	The secondary level of comparison is the comparison based on the primary level, and if this is equal, then it compares diacritics (that is, accents).
`3` (default)	The tertiary level of comparison. As with level `2`, adding case and variants.
`4`	The quaternary level. Limited for specific use cases to consider the punctuation when levels `1-3` ignore punctuation, or for processing Japanese text.
`5`	Identical level. Limited for specific use cases: a tie-breaker.

Chapter 9: Monitoring, Backup, and Security



Chapter 10: Managing Storage Engines

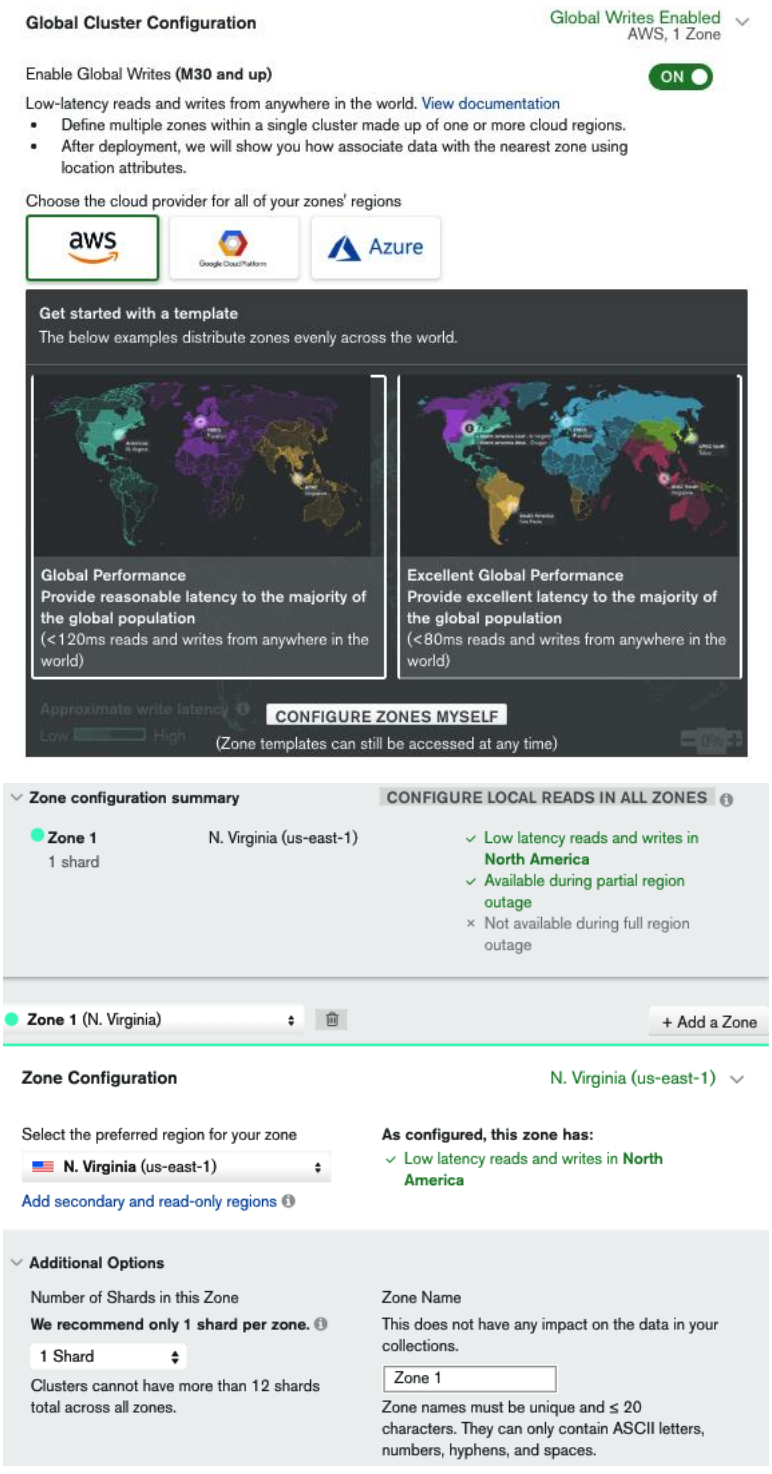


Mode	NL	IS	IX	S	SIX	X
NL	Yes	Yes	Yes	Yes	Yes	Yes
IS	Yes	Yes	Yes	Yes	Yes	No
IX	Yes	Yes	Yes	No	No	No
S	Yes	Yes	No	Yes	No	No
SIX	Yes	Yes	No	No	No	No
X	Yes	No	No	No	No	No

Key	Value
`block_allocation`	Best or first
`allocation_size`	512 bytes through to 128 MB; the default is 4 KB
`block_compressor`	None, `.lz4`, `.snappy`, `.zlib`, `.zstd`, or a custom compressor identifier string, depending on the configuration
`memory_page_max`	512 bytes through to 10 TB; the default 5 MB
`os_cache_max`	An integer greater than or equal to zero; the default is zero

Command	Lock
`find()`	S
`it() (query cursor)`	S
`insert()`	X
`remove()`	X
`update()`	X
`mapreduce()`	Both S and X, depending on the use case. Some MapReduce chunks can run in parallel.
`index()`	Foreground indexing: Database lock. Background indexing: No lock, except for administrative commands that will return an error. Also, background indexing will take considerably more time.
`aggregate()`	S

Chapter 11: MongoDB Tooling



Cluster Tier

M30 (8 GB RAM, 40 GB Storage)
120 IOPS, Encrypted, Auto-expand Storage

Base hourly rate is for a MongoDB replica set with **3 data bearing servers**.

Shared Clusters

M0	Shared RAM	512 MB Storage	Shared vCPUs	Free forever
M2	**Shared** RAM	**2 GB** Storage	**Shared vCPUs**	$9 / MONTH
M5	**Shared** RAM	**5 GB** Storage	**Shared vCPUs**	$25 / MONTH

Dedicated Development Clusters

M10	**2 GB** RAM	**10 GB** Storage	**0.2 vCPUs**	from **$0.08**/hr
M20	**4 GB** RAM	**20 GB** Storage	**0.4 vCPUs**	from **$0.20**/hr

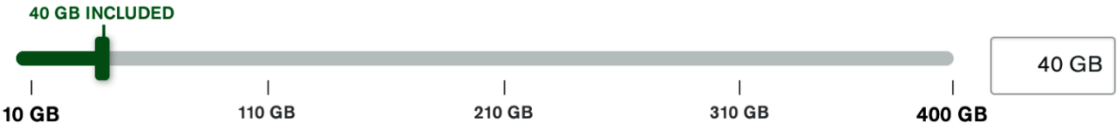
Dedicated Production Clusters

✓ M30	**8 GB** RAM	**40 GB** Storage	**2 vCPUs**	from **$0.54**/hr
› M40	**16 GB** RAM	**80 GB** Storage	**4 vCPUs**	from **$1.04**/hr
› M50	**32 GB** RAM	**160 GB** Storage	**8 vCPUs**	from **$2.00**/hr
› M60	**64 GB** RAM	**320 GB** Storage	**16 vCPUs**	from **$3.95**/hr
M80 LOW CPU	**122 GB** RAM	**750 GB** Storage	**16 vCPUs**	from **$5.61**/hr
M100	**160 GB** RAM	**1000 GB** Storage	**40 vCPUs**	from **$9.16**/hr
› M200	**256 GB** RAM	**1500 GB** Storage	**64 vCPUs**	from **$14.59**/hr
M400 LOW CPU	**488 GB** RAM	**3000 GB** Storage	**64 vCPUs**	from **$22.40**/hr

Customize Your Storage

Storage Capacity

The ranges below are different for each cluster tier. If you need more than 400 GB, we recommend selecting the next cluster tier.



Storage Speed

Standard	Fast	Fastest
120 max IOPS	660 max IOPS	1200 max IOPS

Auto-expand Storage

When 90% of disk capacity is used, automatically expand storage so that only 70% of disk capacity is used.

YES

Additional Settings

MongoDB 4.0, Backup
Continuous Backup

Select a Version

All clusters launch with the WiredTiger™ storage engine.

4.0 with WiredTiger™

Turn on Backup **(M10 and up)**

YES

You can easily enable or disable backups at any time after deploying.
Compare Backup Solutions

Continuous
Point-in-time data recovery and fast, granular data restores

- **$1.50 - $2.50/GB Month***
- Restore to any point in time within the last 24 hours
- Quickly restore with queryable snapshots
- Backups are stored in the same country** as your database
- 12 months of snapshots by default

* *Charges will vary depending on database size and backup retention schedule.*
** *For the US, UK, Ireland, Germany, and Australia. Learn more*

NEW

Cloud Provider Snapshots
Localized backup storage with fast restore times for snapshot images

- **Starting at $0.19/GB month***
- Fast time to restore
- Backups are stored in the same cloud region as your database
- Snapshots are taken at 24 hour increments
- 3 Snapshots stored by default

* *Total charges will vary depending on the region you choose, the size of each snapshot and the number of snapshots retained. Learn more*

Advanced Settings

Enable Business Intelligence Connector **(M10 and up)**

NO

The BI Connector allows you to visualize your data on relational business intelligence tools (e.g. Tableau, MicroStrategy, Qlik).

BI Connector pricing for M30	**$27.18**/day for sustained monthly usage **or $77.02**/day, up to $826.93/month maximum

Encryption at Rest using your Key Management **(M10 and up)**

NO

Encrypt with your configured AWS KMS or Azure Key Vault to ensure that database files written to the filesystem and any backup snapshots are encrypted by the WiredTiger™ Encrypted Storage Engine using keys that you control.

More Configuration Options

Set Oplog Size

Set the maximum size of this cluster's oplog.
View documentation

2048 MB *30,720 MB available*

Set Minimum TLS Protocol Version

Configure this cluster to only accept connections using selected protocols below.
View documentation

TLS 1.1 and ab..

Enforce Index Key Limit

Documents can only be updated or inserted if, for all indexed fields, the corresponding index entries do not exceed 1024 bytes.

If disabled (`failIndexKeyTooLong=false`), documents with indexed fields exceeding the index limit will be inserted or modified successfully, but not indexed. *View documentation*

YES

Require Indexes for All Queries

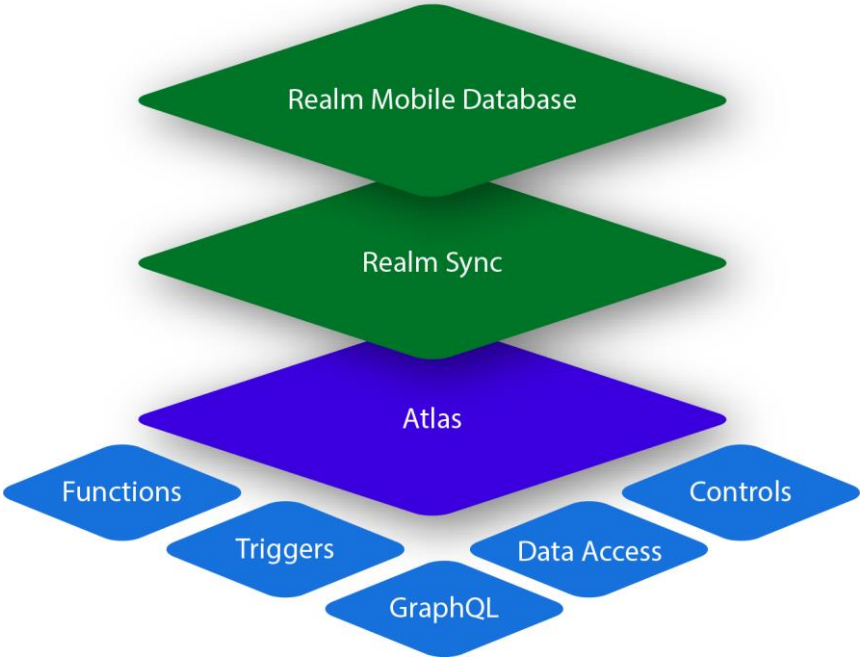
Do not run queries that require a collection scan; return an error instead. *View documentation*

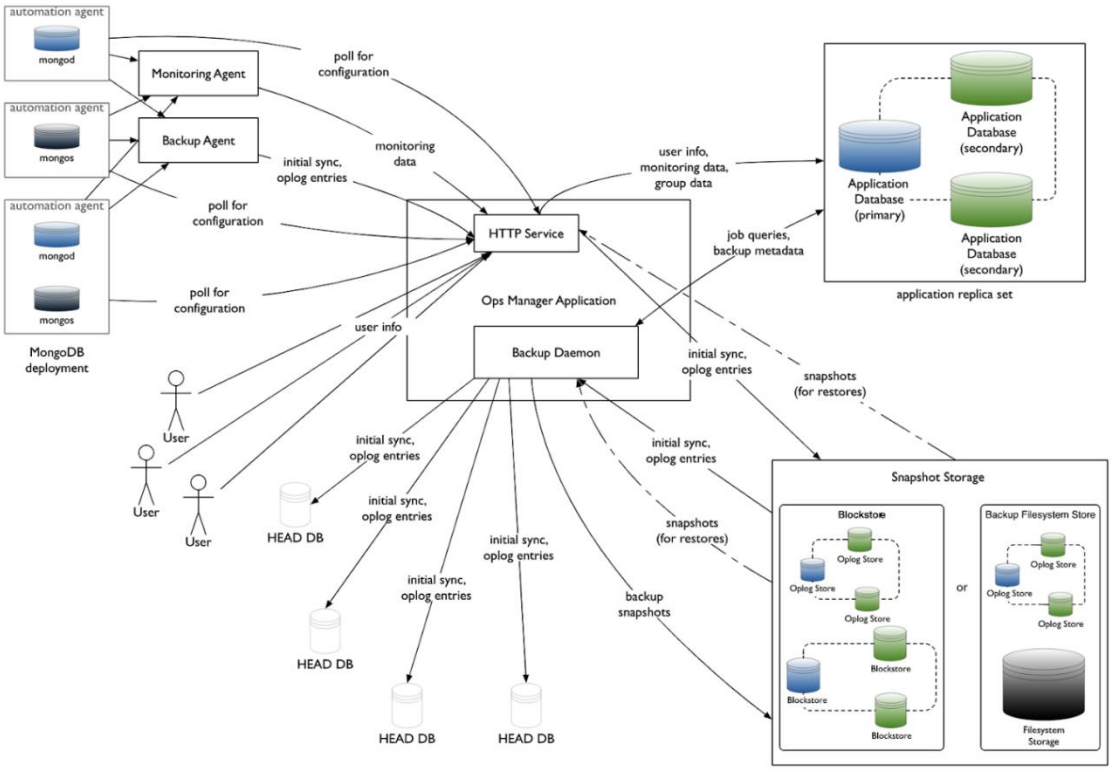
NO

Allow Server-Side JavaScript

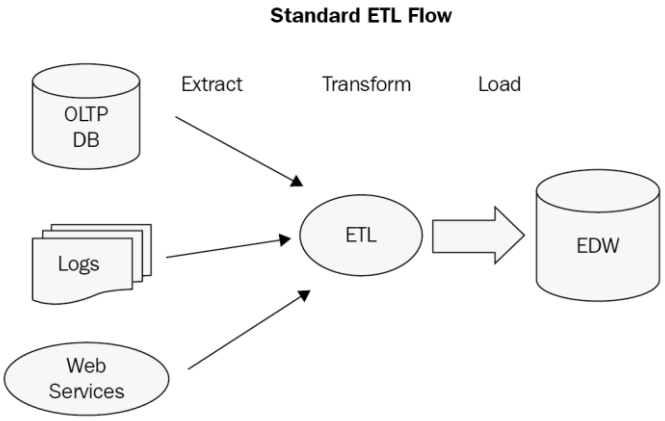
Operations that perform server-side execution of JavaScript are allowed (e.g. `$where` query operator, `mapReduce`, etc.). *View documentation*

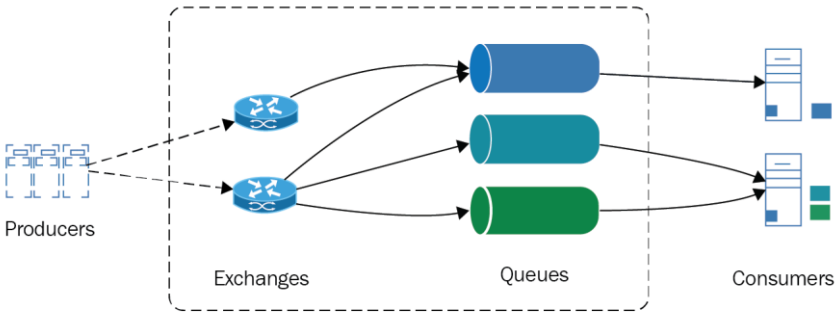
YES


Realm Mobile Database
Realm Sync
Atlas
Functions
Triggers
Data Access
Controls
GraphQL

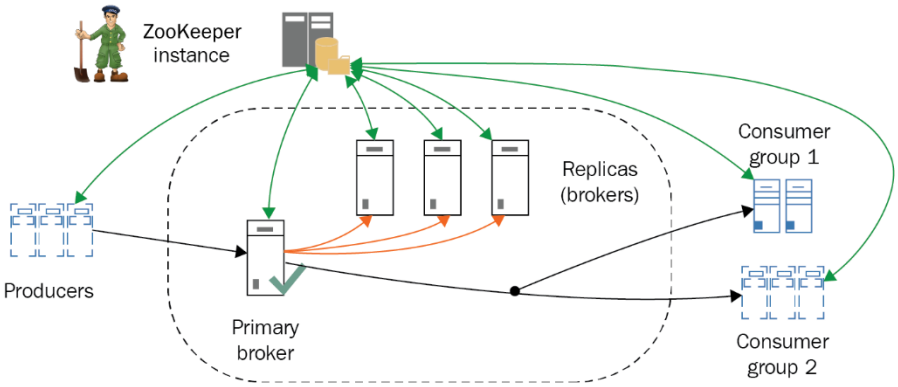

automation agent
mongod
mongos
MongoDB deployment
Monitoring Agent
Backup Agent
poll for configuration
monitoring data
initial sync, oplog entries
user info
HTTP Service
Ops Manager Application
Backup Daemon
user info, monitoring data, group data
job queries, backup metadata
Application Database (primary)
Application Database (secondary)
application replica set
User
HEAD DB
snapshots (for restores)
backup snapshots
Snapshot Storage
Blockstore
Oplog Store
Backup Filesystem Store
Filesystem Storage
or

Chapter 12: Harnessing Big Data with MongoDB

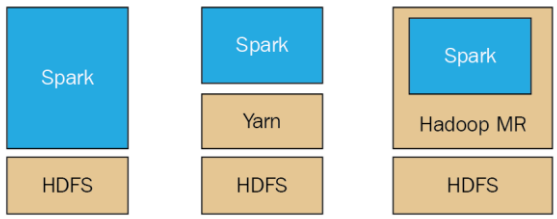


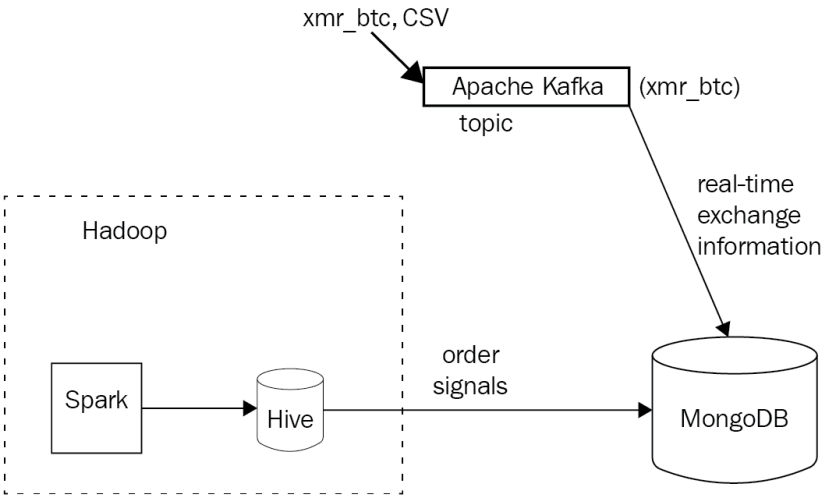


RabbitMQ broker



Apache Kafka architecture

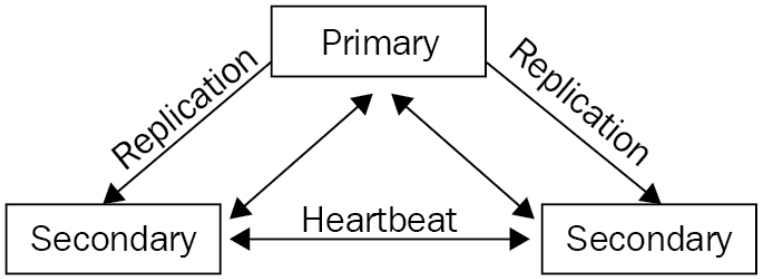


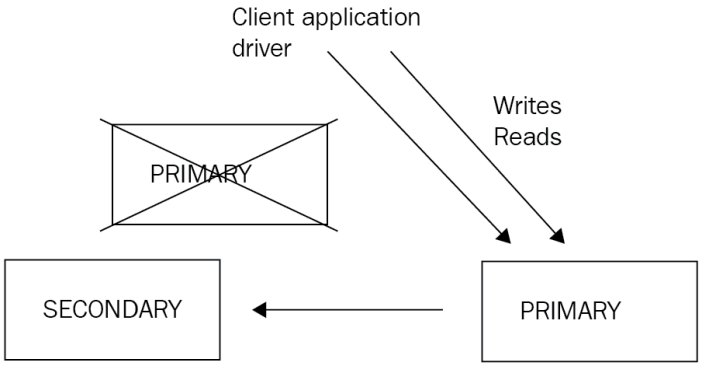


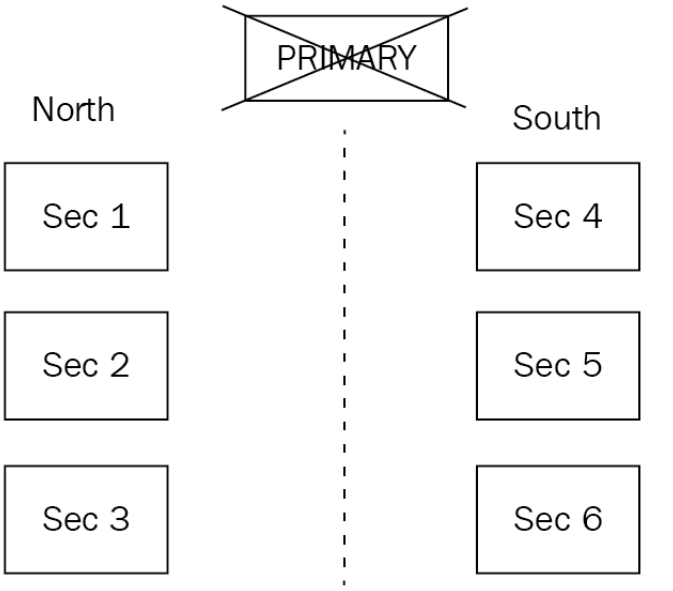
	Hadoop MapReduce	**Apache Spark**
Written in	Java	Scala
Programming model	MapReduce	RDD
Client bindings	Most high-level languages	Java, Scala, Python
Ease of use	Moderate, with high-level abstractions (Pig, Hive, and so on)	Good
Performance	High throughput in batch	High throughput in streaming and batch mode
Uses	Disk (**input/output** (**I/O**) bound)	Memory, degrading performance if a disk is needed
Typical node	Medium	Medium-large

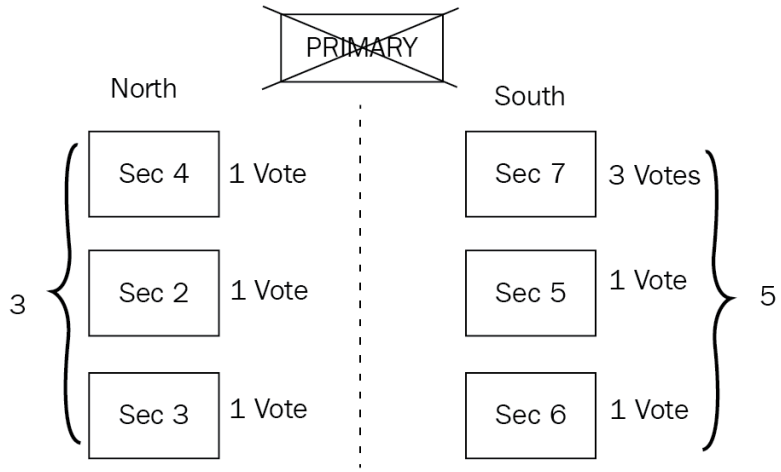
`customerid`	`Int`
`pair`	`String`
`time`	`TIMESTAMP`
`recommendation`	`Int`

Chapter 13: Mastering Replication









Read Preference Mode	Description
`primary`	This is the default mode where reads come from the `primary` server of the replica set.
`primaryPreferred`	With this mode, applications will read from the `primary` unless it is unavailable, in which case reads will come from `secondary` members.
`secondary`	Reads come exclusively from `secondary` servers.
`secondaryPreferred`	With this mode, applications will read from `secondary` members unless they are unavailable, in which case reads will come from the `primary` member.
`nearest`	Applications will read from the member of the replica set that is `nearest` in terms of network latency, not taking into account the member's type.

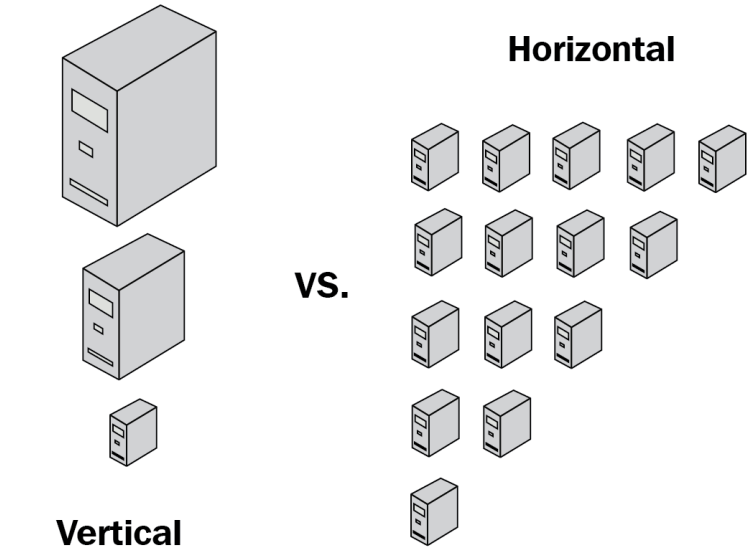
Maintenance window, in hours	Delay	Oplog size on primary, in hours
0.5	[0.5,5]	5

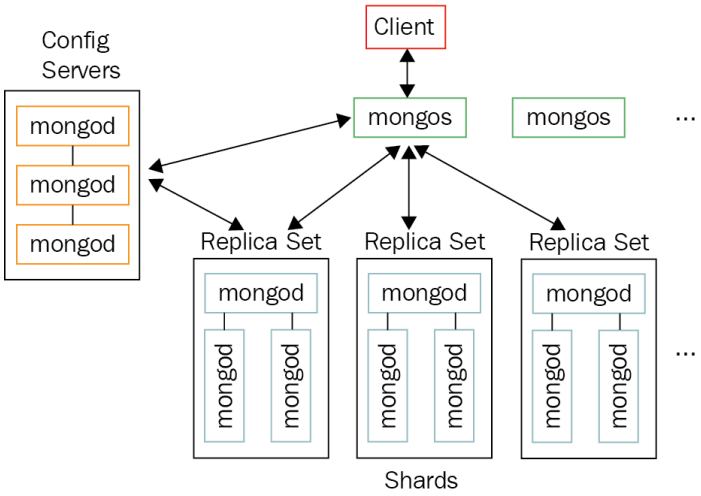
Option	Description	Type	Default
`replica_set`	As used in our example: the replica set name.	String	None
`Write`	The `write` concern options as a `hash` object; the available options are `w`, `wtimeout`, `j`, and `fsync`. That is, to specify writes to two servers, with journaling, flushing to disk (`fsync`) `true`, and a timeout of `1` second as follows: `{ write: { w: 2, j: true, wtimeout: 1000, fsync: true } }`	Hash	`{ w: 1 }`

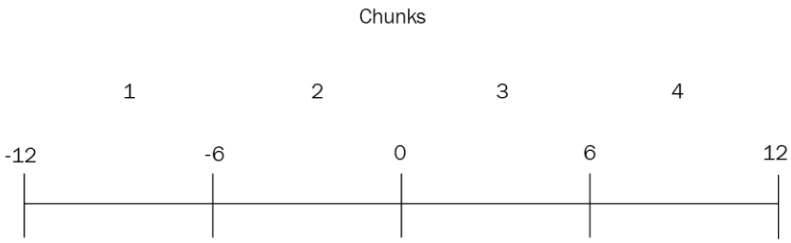
`read`	The read preference mode as a hash. Available options are `mode` and `tag_sets`.
	That is, to limit reads from secondary servers that have the `UKWrites` tag as follows:
	`{ read:`
	`{ mode: :secondary,`
	`tag_sets: [ "UKWrites" ]`
	`}`
	`}`
		Hash	`{ mode: primary }`

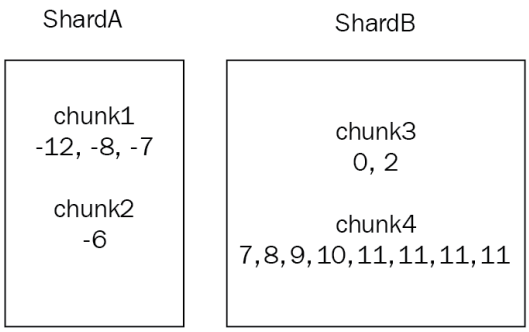
`user`	The name of the user to authenticate with.	String	None
`password`	The password of the user to authenticate with.	String	None
`connect`	Using `:direct`, we can force treat a replica set member as a standalone server, bypassing auto-discovery. Other options include `:direct`, `:replica_set`, and `:sharded`.	Symbol	None
`heartbeat_frequency`	How often replica set members will communicate to check whether they are all alive.	Float	`10`
`database`	Database connection.	String	`admin`

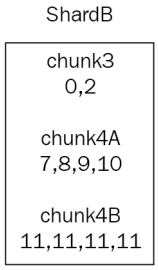
Chapter 14: Mastering Sharding

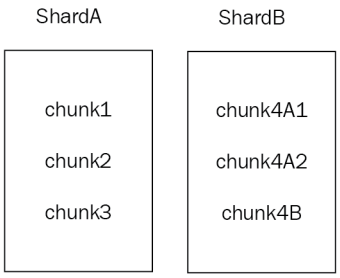












Number of chunks	Migration threshold
≤19	2
20-79	4
≥80	8

Type of operation	Query topology
Insert	Must have the shard key
Update	Can have the shard key
Query with shard key	Targeted operation
Query without shard key	Scatter-and-gather operation/fan-out query
Indexed, sorted query with shard key	Targeted operation
Indexed, sorted query without shard key	Distributed sort merge

Chapter 15: Fault Tolerance and High Availability