MongoDB 3.6: What's New

November 2017

mongoDB

Table of Contents

Introduction

Every industry is being transformed by data and digital technologies. As you build or remake your company for a digital world, **speed matters** – measured by how fast you build and evolve apps, how fast you scale them, and how fast you can gain insights from the data they generate. These are the keys to applications that provide better customer experiences, enable deeper, data-driven insights or make new products or business models possible. If you can't do this, you'll get supplanted by competitors that can.

MongoDB 3.6 helps you **move at the speed of your data** - turning developers, operations teams, and analysts into a growth engine for the business. It enables new digital initiatives and modernized applications to be delivered to market faster, running reliably and securely at scale, and unlocking insights and intelligence ahead of your competitors.

Speed to Develop

MongoDB 3.6 accelerates how quickly you can bring new apps and services to market by making developers even more productive. The new 3.6 release delivers key innovations that once again raise the bar for what developers should demand from their database, enabling them to maximize the time focused on customer experiences. It's now faster and easier to build always-on applications that react in real time to data changes streamed across distributed systems. With Compass Community, all developers can move beyond the command line to innovate faster within an intuitive GUI.

Speed to Scale

Once an application has been developed, it needs to be deployed and run securely, at scale. MongoDB 3.6 accelerates speed to production with enhanced operational tooling providing prescriptive management and flexible backups of distributed database clusters, along with schema governance, extended security controls, and end-to-end compression for network and storage efficiency.

Speed to Insight

MongoDB 3.6 increases your ability to drive real-time action by generating complex analytics and visualizations from live operational data, eliminating the complexity and latency that comes from having to move data into dedicated analytics systems. The aggregation pipeline is expanded to enable more expressive data transformations and JOINs directly within the database. The MongoDB Connector for BI takes advantage of these enhancements to deliver faster time to insight, while the R Driver for MongoDB enables developers, statisticians, and data scientists with idiomatic access to multi-structured data for new data analysis workloads.

Run Anywhere

Organizations want the flexibility to run applications anywhere. MongoDB provides complete platform independence: on-premises, hybrid deployments, or as a fully managed service in the cloud, with the freedom to move between each platform as business requirements change. MongoDB Atlas adds multi-region replication and automatic storage scaling to deliver continuously available, globally distributed applications.

Get Started, Fast

MongoDB 3.6 can be downloaded today as a Release Candidate (RC), ready for your evaluation. General Availability (GA) will follow shortly, at which time MongoDB 3.6 will be available for production deployment on either your own infrastructure, or in the cloud using the MongoDB Atlas managed database service. The Major Version Upgrade service from MongoDB global consulting is designed to accelerate your transition to MongoDB 3.6. You will receive guidance from a consulting engineer on the necessary steps to upgrade, get a walk through of the upgrade process, and get help on evaluating the upgrade in a testing environment. No cost training is also available from the MongoDB University.

Each of the enhancements delivered by MongoDB 3.6 is covered in more detail through the rest of this whitepaper. Review the MongoDB 3.6 release notes for documentation on the new features.

Speed to Develop

MongoDB has always been a developer-first technology. Its document data model maps naturally to objects in application code, making it simple for developers to learn and use. A document's schema can be dynamically created and modified without downtime, making it fast to build and evolve applications. Native, idiomatic drivers are provided for 10+ languages – and the community has built dozens more – enabling ad-hoc queries, real-time aggregation and rich indexing to provide powerful programmatic ways to access and analyze data of any structure. MongoDB 3.6 builds upon these core capabilities to allow developers to create rich apps and customer experiences, all with less code.

Change Streams

Change streams enable developers to build reactive, real-time, web, mobile, and IoT apps that can view, filter, and act on data changes as they occur in the database. Change streams enable seamless data movement across distributed database and application estates, making it simple to stream data changes and trigger actions wherever they are needed, using a fully reactive programming style.

Implemented as an API on top of MongoDB's operation log (oplog), consumers can open change streams against collections and filter on relevant events using the $match, $project, and $redact aggregation pipeline stages. The application can register for notifications whenever a document or collection is modified, enabling downstream applications and consumers to act on new data in real time, without constantly querying the entire collection to identify changes. Applications can consume change streams directly, via a message queue, or through a backend service such as MongoDB Stitch (coming soon).

Use cases enabled by MongoDB change streams include:

- Powering trading applications that need to be updated in real time as stock prices rise and fall.
- Synchronizing updates across serverless and microservices architectures by triggering an API call when a document is inserted or modified. For example,

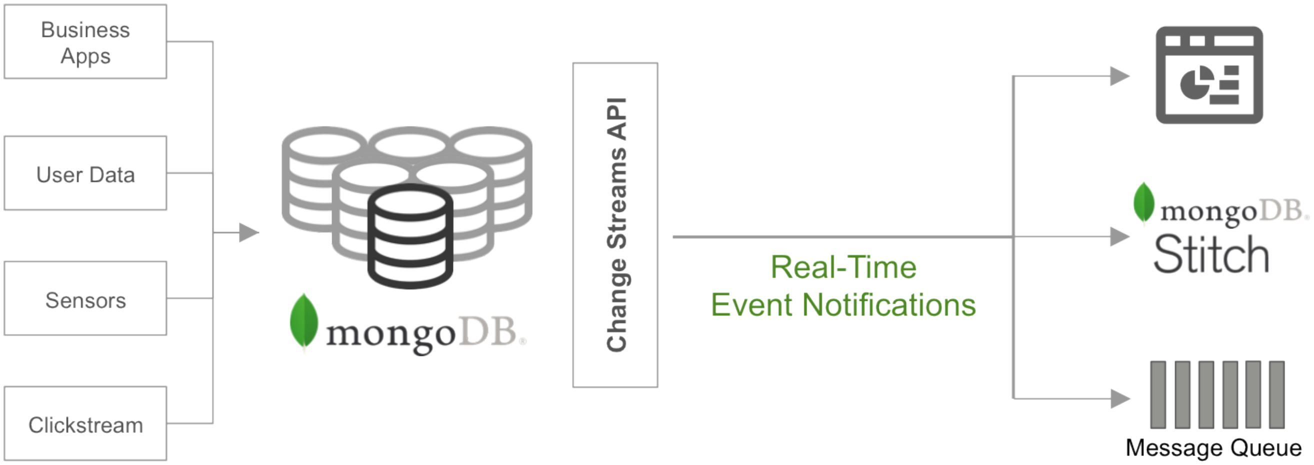


Figure 1: MongoDB change streams enable consumers to react to data changes in real time

new customer orders written to the database may automatically trigger functions to generate invoices and delivery schedules.

- Updating dashboards, analytics systems, and search engines as operational data changes.
- Creating powerful IoT data pipelines that can react whenever the state of physical objects change. For example, generating alarms whenever a connected vehicle moves outside of a geo-fenced area.
- Pushing new credit card transactions into machine learning training models to re-score fraud classifications.
- Refreshing scoreboards in multiplayer games.

Some MongoDB users requiring real-time notifications have built their own change data capture processes that “tail” the oplog. By migrating to change streams, these users can reduce development and operational overhead, improve usability, and increase data reliability. When compared to both oplog tailing and change notifications implemented by alternative databases, MongoDB change streams offer a number of advantages:

- Change streams are **flexible** – users can register to receive just the individual deltas from changes to a document, or receive a copy of the full document.
- Change streams are **consistent** – by utilizing a global logical clock, change streams ensure a total ordering of event notifications across shards. As a result, MongoDB guarantees the order of changes will be preserved, and can be safely processed by the consuming application in the order received from the stream.
- Change streams are **secure** – users are able to create change streams only on collections to which they have been granted read access.
- Change streams are **reliable** – notifications are only sent on majority committed write operations, and are durable when nodes or the network fails.
- Change streams are **resumable** – when nodes recover after a failure, change streams can be automatically resumed, assuming that the last event received by the application has not rolled off the oplog.
- Change streams are **familiar** – the API syntax takes advantage of the established MongoDB drivers and query language, and are independent of the underlying oplog format.
- Change streams are highly **concurrent** – up to 1,000 change streams can be opened against each MongoDB instance with minimal performance degradation.

Review the MongoDB change streams documentation to learn more.

Retryable Writes

The addition of retryable writes to MongoDB moves the complexity of handling temporary system failures from the application to the database. Now, rather than the developer having to implement custom, client-side code, the MongoDB driver can automatically retry writes in the event of transient network failures or a primary replica election, while the MongoDB server enforces exactly-once processing semantics.

By assigning a unique transaction identifier to each write operation, the driver re-sends that ID to enable the server to evaluate success of the previous write attempt, or retry the write operation as needed. This implementation of retryable writes offers a number of benefits over approaches taken by other databases:

- Retryable writes are not limited to idempotent operations only. They can also be applied to operations such as incrementing or decrementing a counter, or processing orders against stock inventory.
- Retryable writes are safe for operations that failed to acknowledge success back to the application due to timeout exceptions, for example due to a transient network failure.
- Retryable writes do not require developers to add any extra code to their applications, such as retry logic or savepoints.

Applications that cannot afford any loss of write availability, such as e-commerce applications, trading exchanges, and IoT sensor data ingestion, immediately benefit from retryable writes. When coupled with self-healing node recovery – typically within 2-seconds or less – MongoDB's retryable writes enable developers to deliver always-on, global availability of write operations, without the risks of data loss and stale reads imposed by eventually consistent, multi-master systems.

Tunable Consistency

With tunable consistency, MongoDB affords developers precise control over routing queries across a distributed cluster, balancing data consistency guarantees with performance requirements. MongoDB 3.4 added linearizable reads, which were central to MongoDB passing Jepsen – some of the most stringent data safety and correctness tests in the database industry. Now the MongoDB 3.6 release introduces additional consistency controls:

- Causal consistency – guaranteeing that every read operation within a client session will always see the previous write operation, regardless of which replica is serving the request. By enforcing strict, causal ordering of operations within a session, causal consistency ensures every read is always logically consistent, enabling monotonic reads from a distributed system – guarantees that cannot be met by most multi-node databases. Causal consistency allows developers to maintain the benefits of strict data consistency enforced by legacy single node relational databases, while modernizing their infrastructure to take advantage of the scalability and availability benefits of modern distributed data platforms.
- Consistent reads against secondary replica set members in sharded clusters, even while data is being balanced across nodes. In MongoDB 3.6, secondary replicas are now "chunk aware", enabling them to filter any orphaned documents from result sets. This enhancement improves data consistency and read scalability for MongoDB databases deployed across elastic, globally distributed, sharded clusters.

MongoDB Compass

As the GUI for MongoDB, Compass has become an indispensable tool for developers and DBAs, enabling graphical schema discovery and query optimization. Compass now offers several new features:

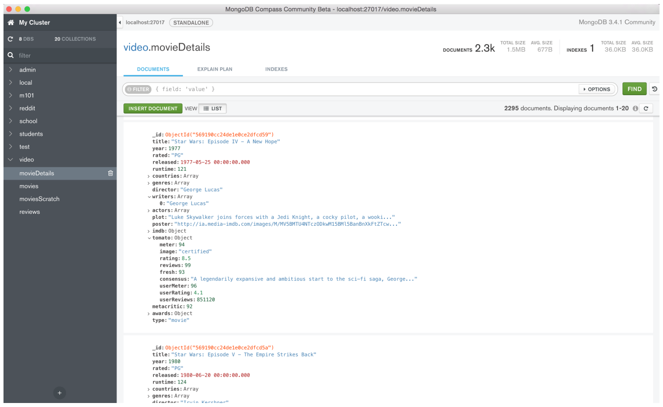
- **Auto-complete**: Enables developers to simplify query development with Compass providing suggestions for field names and MongoDB operators, in addition to matching braces and quotes as they code.
- **Query History**: Allows developers to re-run their most recently executed queries, and save common queries to run on-demand.
- **Table View**: Now developers can view documents as conventional tables, as well as JSON documents.

MongoDB Compass is not just a single tool – it's a framework built to allow for the addition of modular components. Compass now exposes this as the **Compass Plugin Framework**, making Compass extensible by any user with the same methods used by MongoDB's software engineers. Using the plugin API, users can build plugins to add new features to Compass. Examples include a GridFS viewer, a sample data generator, a hardware stats viewer, a log collector/analyzer, and more.

MongoDB Compass Community

With the MongoDB 3.6 release, the Compass family has expanded to now include the new, no-cost Compass Community edition.

Compass Community provides developers an intuitive visual interface to use alongside the MongoDB shell. It includes the core features of Compass, enabling users to review the hierarchy and size of databases and collections, inspect documents, and insert / update / delete documents. Developers can use the GUI to build queries, examine how they're executed, and add or drop indexes to improve performance. Compass Community also supports the latest Compass functionality available with MongoDB 3.6, making developers even more productive.



```
  _id: ObjectId("569190cc24de1e0ce2dfcd59")
  title: "Star Wars: Episode IV - A New Hope"
  year: 1977
  rated: "PG"
  released: 1977-05-25 00:00:00.000
  runtime: 121
> countries: Array
> genres: Array
  director: "George Lucas"
v writers: Array
    0: "George Lucas"
> actors: Array
  plot: "Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a wooki..."
  poster: "http://ia.media-imdb.com/images/M/MV5BMTU4NTczODkwM15BMl5BanBnXkFtZTcw..."
> imdb: Object
v tomato: Object
    meter: 94
    image: "certified"
    rating: 8.5
    reviews: 99
    fresh: 93
    consensus: "A legendarily expansive and ambitious start to the sci-fi saga, George..."
    userMeter: 96
    userRating: 4.1
    userReviews: 851120
  metacritic: 92
> awards: Object
  type: "movie"

  _id: ObjectId("569190cc24de1e0ce2dfcd5a")
  title: "Star Wars: Episode V - The Empire Strikes Back"
  year: 1980
  rated: "PG"
  released: 1980-06-20 00:00:00.000
  runtime: 124
> countries: Array
> genres: Array
  director: "Irvin Kershner"
```

Figure 2: MongoDB Compass Community, new no-cost GUI for MongoDB developers

MongoDB Compass Community is available from the MongoDB download center.

Fully Expressive Array Updates

Arrays are a powerful construct in MongoDB's document data model, allowing developers to represent complex objects in a single document that can be efficiently retrieved in one call to the database. Before MongoDB 3.6, however, it was only possible to atomically update the first matching array element in a single update command.

With fully expressive array updates, developers can now perform complex array manipulations against matching elements of an array – including elements embedded in nested arrays – all in a single atomic update operation. MongoDB 3.6 adds a new `arrayFilters` option, allowing the update to specify which elements to modify in the array field. This enhancement allows even more flexibility in data modeling. It also delivers higher performance than alternative databases supporting JSON data as entire documents do not need to be rewritten when only selective array elements are updated.

Learn more from the documentation.

More Expressive Query Language

MongoDB 3.6 exposes the ability to use aggregation expressions within the query language to enable richer queries with less client-side code. This enhancement allows the referencing of other fields in the same document when executing comparison queries, as well as powerful expressions such as multiple JOIN conditions and uncorrelated subqueries. The addition of the new expression operator allows the equivalent of `SELECT * FROM T1 WHERE a>b` in SQL syntax. Learn more from the $expr documentation.

Speed to Scale

Unlike the traditional scale-up systems of the past, distributed systems enable applications to scale further and faster while maintaining continuous availability in the face of outages and maintenance. However, they can impose more complexity on the ops team, potentially slowing down the pace of delivering, scaling, and securing apps in production. MongoDB 3.6 takes another important step in making it easier for operations teams to deploy and run massively scalable, always-on global applications that benefit from the power of a distributed systems architecture.

Ops Manager

MongoDB Ops Manager is the best way to run MongoDB on your own infrastructure, making operations staff

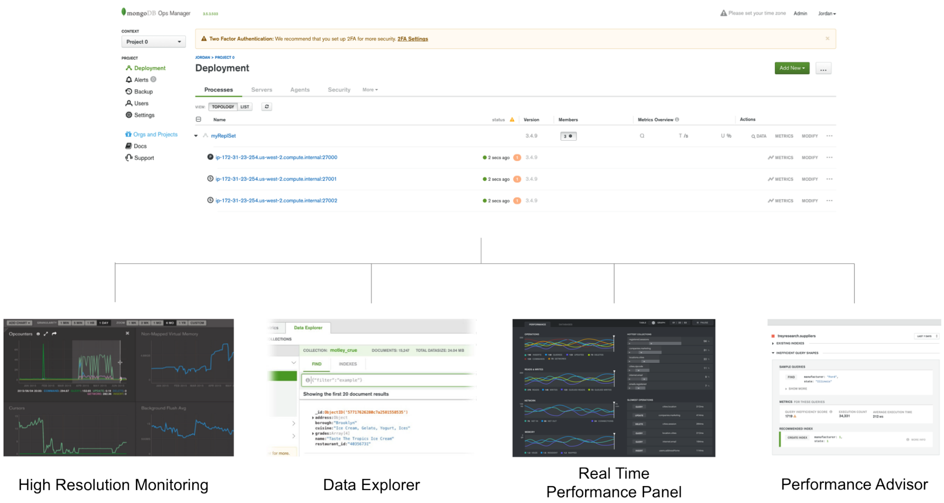


Figure 3: Ops Manager performance telemetry and prescriptive recommendations speeds time to scale

10x-20x more productive. Advanced management and administration delivered with Ops Manager 3.6 allow operations teams to manage, optimize, and backup distributed MongoDB clusters faster and at higher scale than ever before. Deeper operational visibility allows proactive database management, while streamlined backups reduce the costs and time of data protection.

Simplified Monitoring and Management

It is now easier than ever for administrators to synthesize schema design against real-time database telemetry and receive prescriptive recommendations to optimize database performance and utilization – all from a single pane of glass.

- The **Data Explorer** allows operations teams to examine the database's schema by running queries to review document structure, viewing collection metadata, and inspecting index usage statistics, directly within the Ops Manager UI.
- The **Real Time Performance Panel** provides insight from live server telemetry, enabling issues to be immediately identified and diagnosed. The panel displays all operations in flight, network I/O, memory consumption, the hottest collections, and slowest queries. Administrators also have the power to kill long running operations from the UI.
- The new Performance Advisor, available for both Ops Manager and MongoDB Atlas, continuously highlights slow-running queries and provides intelligent index recommendations to improve performance. Using Ops Manager automation, the administrator can then roll out the recommended indexes automatically, without incurring any application downtime.

Ops Manager Organizations

To simplify management of global MongoDB estates, Ops Manager now provides a new Organizations and Projects hierarchy. Previously Projects, formerly called "groups", were managed as individual entities. Now multiple Projects can be placed under a single organization, allowing operations teams to centrally view and administer all Projects under the organization hierarchy. Projects can be assigned tags, such as a "production" tag, against which global alerting policies can be configured.

Faster, Cheaper and Queryable Backups

Ops Manager continuously maintains backups of your data, so if an application issue, infrastructure failure, or user error compromises your data, the most recent backup is only moments behind, minimizing exposure to data loss. Ops Manager offers point-in-time backups of replica sets, and cluster-wide snapshots of sharded clusters, guaranteeing consistency and no data loss. You can restore to precisely the moment you need, quickly and safely. Ops Manager backups are enhanced with a range of new features:

- Queryable Backups, first introduced in MongoDB Atlas, allow partial restores of selected data, and the ability to query a backup file in-place, without having to restore it. Now users can query the historical state of the database to track data and schema modifications – a common demand of regulatory reporting. Directly querying backups also enables administrators to identify the best point in time to restore a system by comparing data from multiple snapshots, thereby improving both RTO and RPO. No other non-relational database offers the ability to query backups in place.

- The Ops Manager 3.6 backup agent has been updated to use a faster and more robust initial sync process. Now, transient network errors will not cause the initial sync to restart from the beginning of the backup process, but rather resume from the point the error occurred. In addition, refactoring of the agent will speed data transfer from MongoDB to the backup repository, with the performance gain dependent on document size and complexity.
- Reducing backup storage overhead by 1x of your logical production data and further improving speed to recovery, Point-in-Time snapshots will now be created at the destination node for the restore operation, rather than at the backup server, therefore reducing network hops. The restore process now transfers backup snapshots directly to the destination node, and then applies the oplog locally, rather than applying it at the daemon server first and then pushing the complete restore image across the network. Note that this enhancement does not apply to restores via SCP.
- Extending support for the AWS S3 object store, backups can now be routed to on-premises object stores such as EMC ECS or IBM Cleversafe. MongoDB's backup integration provides administrators with greater choice in selecting the backup storage architecture that best meets specific organizational requirements for data protection. It enables them to take advantage of cheap, durable, and quickly growing object storage used within the enterprise. By limiting backups to filesystems or S3 only, most other databases fail to match the storage flexibility offered by MongoDB.
- With cross-project restores, users can now perform restores into a different Ops Manager Project than the backup snapshot source. This allows DevOps teams to easily execute tasks such as creating multiple staging or test environments that match recent production data, while configured with different user access privileges or running in different regions.

Ops Manager 3.6 will ship when MongoDB 3.6 reaches GA.

Schema Validation

MongoDB 3.6 introduces Schema Validation via syntax derived from the proposed IETF JSON Schema standard. This new schema governance feature extends the capabilities of document validation, originally introduced in MongoDB 3.2.

While MongoDB's flexible schema is a powerful feature for many users, there are situations where strict guarantees on data structure and content are required. MongoDB's existing document validation controls can be used to require that any documents inserted or updated follow a set of validation rules, expressed using MongoDB query syntax. While this allows for the definition of required content for each document, it had no mechanism to restrict users from adding documents containing fields beyond those specified in the validation rules. In addition, there is no way for administrators to specify and enforce control over the complete structure of documents, including data nested inside arrays.

Using schema validation, DevOps and DBA teams can now define a prescribed document structure for each collection, which can reject any documents that do not conform to it. With schema validation, MongoDB enforces controls over JSON data that are unmatched by any other database:

- **Complete schema governance**. Administrators can define when additional fields are allowed to be added to a document, and specify a schema on array elements including nested arrays.
- **Tunable controls**. Administrators have the flexibility to tune schema validation according to use case – for example, if a document fails to comply with the defined structure, it can be either be rejected, or still written to the collection while logging a warning message. Structure can be imposed on just a subset of fields – for example requiring a valid customer a name and address, while others fields can be freeform, such as social media handle and cellphone number. And of course, validation can be turned off entirely, allowing complete schema flexibility, which is especially useful during the development phase of the application.

- **Queryable**. The schema definition can be used by any query to inspect document structure and content. For example, DBAs can identify all documents that do not conform to a prescribed schema.

With schema validation, developers and operations teams have complete control over balancing the agility and flexibility that comes from a dynamic schema, with strict data governance controls enforced across entire collections. As a result, they spend less time defining data quality controls in their applications, and instead delegate these tasks to the database. Specific benefits of schema validation include:

1. **Simplified application logic**. Guarantees on the presence, content, and data types of fields eliminates the need to implement extensive error handling in the application. In addition, the need to enforce a schema through application code, or via a middleware layer such as an Object Document Mapper, is removed.
2. **Enforces control**. Database clients can no longer compromise the integrity of a collection by inserting or updating data with incorrect field names or data types, or adding new attributes that have not been previously approved.
3. **Supports compliance**. In some regulated industries and applications, it is required that Data Protection Officers demonstrate that data is stored in a specific format, and that no additional attributes have been added. For example, the EU's General Data Protection Regulation (GDPR) requires an impact assessment against all Personally Identifiable Information (PII), prior to any processing taking place.

Extending Security Controls

MongoDB offers among the most extensive and mature security capabilities of any modern database, providing robust access controls, end-to-end data encryption, and complete database auditing. MongoDB 3.6 continues to build out security protection with two new enhancements that specifically reduce the risk of unsecured MongoDB instances being unintentionally deployed into production.

From the MongoDB 2.6 release onwards, the binaries from the official MongoDB RPM and DEB packages bind to localhost by default. With MongoDB 3.6, this default behavior is extended to all MongoDB packages across all platforms. As a result, all networked connections to the database will be denied unless explicitly configured by an administrator. Review the documentation to learn more about the changes introduced by localhost binding. Combined with new **IP whitelisting**, administrators can configure MongoDB to only accept external connections from approved IP addresses or CIDR ranges that have been explicitly added to the whitelist.

End-to-End Compression

Adding to intra-cluster network compression released in MongoDB 3.4, the new 3.6 release adds wire protocol compression to network traffic between the client and the database.

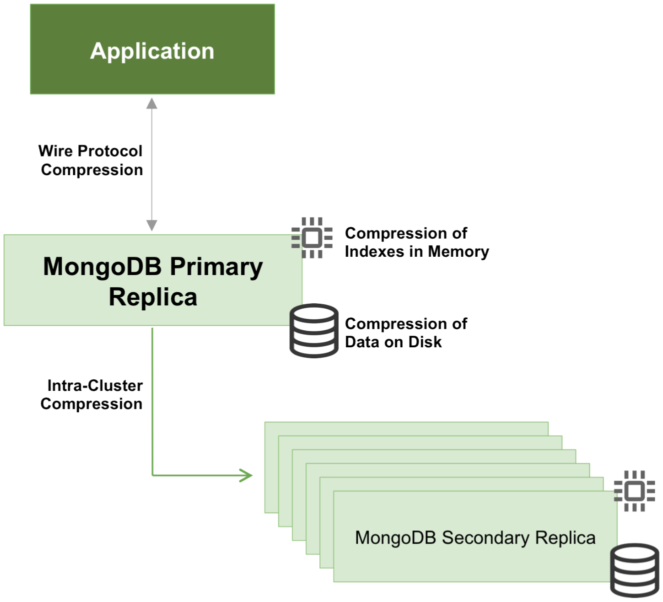

Application
Wire Protocol Compression
MongoDB Primary Replica
Compression of Indexes in Memory
Compression of Data on Disk
Intra-Cluster Compression
MongoDB Secondary Replica

Figure 4: Creating highly efficient distributed systems with end to end compression

Wire protocol compression can be configured with the snappy or zLib algorithms, allowing up to 80% savings in network bandwidth. This reduction brings major performance gains to busy network environments and reduces connectivity costs, especially in public cloud environments, or when connecting remote assets such as IoT devices and gateways.

With compression configurable across the stack – for client traffic, intra-cluster communications, indexes, and disk storage – MongoDB offers greater network, memory, and storage efficiency than almost any other database.

Enhanced Operational Management in Multi-Tenant Environments

Many MongoDB customers have built out their database clusters to serve multiple applications and tenants. MongoDB 3.6 introduces two new features that simplify management and enhance scalability:

Operational session management enables operations teams to more easily inspect, monitor, and control each user session running in the database. They can view, group, and search user sessions across every node in the cluster, and respond to performance issues in real time. For example, if a user or developer error is causing runaway queries, administrators now have the fine-grained operational oversight to view and terminate that session by removing all associated session state across a sharded cluster in a single operation. This is especially useful for multi-tenant MongoDB clusters running diverse workloads, providing a much simpler interface for identifying active operations in the database cluster, recovering from cluster overloads, and monitoring active users on a system. Review the sessions documentation to learn more.

Improved scalability with the WiredTiger storage engine to better support common MongoDB use cases that create hundreds of thousands of collections per database, for example:

- Multi-tenant SaaS-based services that create a collection for each user.
- IoT applications that write all sensor data ingested over an hour or a day into a unique collection.

As the collection count increased, MongoDB performance could, in extreme cases, degrade as the WiredTiger session cache managing a cursor's access to collections and indexes became oversubscribed. MongoDB 3.6 introduces a refactoring of the session cache from a list to hash table, with improved cache eviction policies and checkpointing algorithms, along with higher concurrency by replacing mutexes with Read/Write locks. As a result of this refactoring, a single MongoDB instance running with the WiredTiger storage engine can support over 1 million collections. Michael Cahill, director of Storage Engineering, presented a session on the development work at the MongoDB World '17 customer conference. Review the session slides to learn more.

Speed to Insight

How quickly an organization can **unlock and act on insights** from data generated by new applications has become a material source of competitive advantage. Collecting data in operational systems and then relying on batch ETL (Extract, Transform, Load) processes to update an expensive data warehouse or complex and ungoverned data lake is no longer sufficient. Speed to insight is critical, and so analytics performed against live data to drive operational intelligence is fast becoming a necessity, without having to employ armies of highly skilled and scarce data engineers and scientists. MongoDB 3.6 delivers a number of new features and capabilities that allow organizations to enable real-time analytics and action.

MongoDB Connector for BI: Faster and Simpler

MongoDB 3.6 brings a number of performance and ease-of-use enhancements to the BI Connector, enabling faster time to insight using SQL-based BI and Analytics platforms.

Faster
The connector takes advantage of enhancement to the aggregation pipeline – discussed below – to deliver higher performance, with more operations pushed natively to the database. Prior to MongoDB 3.6, only left outer equijoins could be pushed down to the database – all other JOIN types had to be executed within the BI connector layer, which firstly required all matching data to be extracted from the database. With MongoDB 3.6 support is being extended to non-equijoins and the equivalent of SQL subqueries. These enhancements will reduce the amount of data that needs to be moved and computed in the BI layer, providing faster time to insight.

In addition, performance metrics are now observable via the Show Status function, enabling deeper performance insights and optimizations.

Simpler

To support easier configuration, the mongosqld process now samples and maps the MongoDB schema, caching the results internally and eliminating the need to install the separate mongodrdl component. Additionally, users can simplify lifecycle management by configuring, deploying, and monitoring the BI connector directly from Ops Manager.

To simplify the enforcement of access controls, BI Connector users can now be authenticated directly against MongoDB using new client side plugins, eliminating the need to manage TLS certificates. Review the documentation for the C and JDBC authentication plugins to learn more. Authentication via Kerberos is also now supported.

Richer Aggregation Pipeline

Developers and data scientists rely on the MongoDB aggregation pipeline for its power and flexibility in enabling sophisticated data processing and manipulation demanded by real-time analytics and data transformations. Enhancements in the aggregation pipeline unlock new use cases.

A more powerful $lookup operator extends MongoDB's JOIN capability to support the equivalent of SQL subqueries and non-equijoins. As a result, developers and analysts can write more expressive queries combining data from multiple collections, all executed natively in the database for higher performance, and with less application-side code.

In addition to `$lookup`, the aggregation pipeline offers additional enhancements:

- Support for timezone-aware aggregations. Before timezone awareness, reporting that spanned regions and date boundaries was not possible within the aggregation pipeline. Now business analysts can group data for multi-region analysis that takes account of variances in working hours and working days across different geographic regions.
- New expressions allow richer data transformations within the aggregation pipeline, including the ability to convert objects to arrays of key-value pairs, and arrays of key-value pairs to be converted to objects. The mergeObjects expression is useful for setting missing fields into default values, while the REMOVE variable allows the conditional exclusion of fields from projections based on evaluation criteria. You can learn more about the enhancements from the documentation.

R Driver for MongoDB

A recommended R driver for MongoDB is now available, enabling developers, data scientists, and statisticians to get the same first class experience with MongoDB as that offered by the other MongoDB drivers – providing idiomatic, native language access to the database. The driver supports advanced MongoDB functionality, including:

- Read and write concerns to control data consistency and durability.
- Enterprise authentication mechanisms, such as LDAP and Kerberos, to enforce security controls against the database.
- Support for advanced BSON data types such as Decimal 128 to support high precision scientific and financial analysis.

Run Anywhere

Many organizations are turning to the cloud to accelerate the speed of application development, deployment, and data discovery. Replatforming to the cloud gives them the ability to enable self-service IT, to elastically scale resources on demand, and to align costs to actual consumption. But they are also concerned about exposing the business to deeper levels of lock-in – this time from the APIs and services of the cloud providers themselves.

Increasingly, users are demanding the freedom to run anywhere: private clouds in their own data center, in the public cloud, or in a hybrid model that combines the two. This flexibility is not available when they build on a cloud-proprietary database from a single vendor. Alternatively, the platform independence provided by

MongoDB gives them the ability to respond to business or regulatory changes without incurring the complexity, risk, and time that comes from expensive database migrations whenever they need or want to transition to a new platform.

MongoDB Atlas

As a fully managed database service, MongoDB Atlas is the best way to run MongoDB in the public cloud. 2017 has already seen major evolutions in the Atlas service, with key highlights including:

- Expansion beyond Amazon Web Services (AWS) to offer Atlas on Google Cloud Platform (GCP) and Microsoft Azure.
- Achieving SOC2 Type 1 compliance.
- The launch of managed database clusters on a shared architecture, including the free M0 instances, and the M2s and M5s, which allow customers to jumpstart their projects for a low and predictable price.
- A live migration facility to move data from an existing MongoDB replica set into an Atlas cluster with minimal application impact.
- The addition of the Data Explorer and Real Time Performance Panel, now coming to Ops Manager, as discussed above.

MongoDB 3.6 will be available as a fully managed service on Atlas, along with important new features to support global applications, and with automated scalability and performance optimizations.

Turnkey Global Distribution of Clusters with Cross-Region Replication

MongoDB Atlas clusters can now span multiple regions offered by a cloud provider. This enables developers to build apps that maintain continuous availability in the event of geographic outages, and improve customer experience by locating data closer to users.

When creating a cluster or modifying its configuration, two options are now available:

- Teams can now deploy a single MongoDB database across multiple regions supported by a cloud provider for improved availability guarantees. Reads and writes will default to a “preferred region” assuming that there are no active failure or failover conditions. The nearest read preference, discussed below, can be used to route queries to local replicas in a globally distributed cluster. Replica set members in additional regions will participate in the automated election and failover process if the primary member is affected by a local outage, and can become a primary in the unlikely event that the preferred region is offline.
- Read-only replica set members can be deployed in multiple regions, allowing teams to optimize their deployments to achieve reduced read latency for a global audience. Read preference – providing a mechanism to control how MongoDB routes read operations across members of a replica set – can be configured using the drivers. For example, the nearest read preference routes queries to replicas with the lowest network latency from the client, thus providing session locality by minimizing the effects of geographic latency. As the name suggests, read-only replica set members will not participate in the automated election and failover process, and can never be become a primary.

Teams can activate both of the options outlined above in a single database to provide continuous availability and an optimal experience for their users.

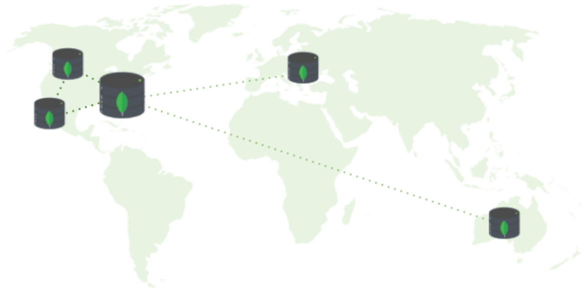


Figure 5: Globally distributed MongoDB Atlas cluster, providing resilience to regional outages and lower latency experiences for global apps

Auto-Scaling Storage and Performance Optimization

MongoDB Atlas now supports automatic scaling for the storage associated with a cluster, making it easier for you to manage capacity. Enabled by default, auto-scaling for storage detects when your disks hit 90% utilization and provisions additional storage such that your cluster reaches a disk utilization of 70% on AWS & GCP, or a maximum of 70% utilization on Azure. This automated process occurs without impact to your database or application availability.

In addition to auto-storage scaling, the new Performance Advisor discussed earlier for Ops Manager is also available in MongoDB Atlas, providing you with always-on, data-driven insights into query behavior and index recommendations.

A Cloud Database Platform for Development & Testing

New enhancements to MongoDB Atlas make it the optimal cloud database for spinning up and running test and development environments efficiently.

- You can now pause your MongoDB Atlas cluster, perfect for use cases where only intermittent access to your data is required, such as development during business hours or temporary testing. While your database instances are stopped, you are charged for provisioned storage and backup storage, but not for instance hours. You can restart your MongoDB Atlas cluster at any time on demand; your cluster configuration will be the same as when you stopped it and public DNS hostnames are retained so no modifications to your connection string are required. MongoDB Atlas clusters can be stopped for up to 7 days. If you do not start your cluster after 7 days, Atlas will automatically start your cluster. Pausing and restarting your MongoDB clusters can be triggered in the MongoDB Atlas UI or via the REST API.
- Cross-project restores, introduced with Ops Manager 3.6, are also available in MongoDB Atlas, allowing users to restore to different MongoDB Atlas projects than the backup snapshot source.

Conclusion

MongoDB 3.6 helps you **move at the speed of your data**. It enables new digital initiatives and modernized applications to be delivered to market faster, running reliably and securely at scale, and unlocking insights and intelligence ahead of your competitors.

- Change streams, retryable writes, tunable consistency, greater query and update expressivity, and Compass Community help developers move faster.
- Ops Manager, schema validation, enhanced security, end to end compression, and user session management help operations teams scale faster.
- The MongoDB aggregation pipeline, Connector for BI, and the recommended R driver help analysts and data scientists unlock insights faster.

And you have the freedom to run MongoDB anywhere – on-premises, public cloud, and as a service with MongoDB Atlas. Get started today by downloading the MongoDB 3.6 Release Candidate and reviewing the release notes. No cost training is also available from the MongoDB University.

We Can Help

We are the MongoDB experts. Over 4,300 organizations rely on our commercial products, including startups and more than half of the Fortune 100. We offer software and services to make your life easier:

MongoDB Enterprise Advanced is the best way to run MongoDB in your data center. It's a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.

MongoDB Atlas is a database as a service for MongoDB, letting you focus on apps instead of ops. With MongoDB Atlas, you only pay for what you use with a convenient hourly billing model. With the click of a button, you can scale up and down when you need to, with no downtime, full security, and high performance.

MongoDB Stitch is a backend as a service (BaaS), giving developers full access to MongoDB, declarative read/write controls, and integration with their choice of services.

MongoDB Cloud Manager is a cloud-based tool that helps you manage MongoDB on your own infrastructure. With automated provisioning, fine-grained monitoring, and continuous backups, you get a full management suite that reduces operational overhead, while maintaining full control over your databases.

MongoDB Professional helps you manage your deployment and keep it running smoothly. It includes support from MongoDB engineers, as well as access to MongoDB Cloud Manager.

Development Support helps you get up and running quickly. It gives you a complete package of software and services for the early stages of your project.

MongoDB Consulting packages get you to production faster, help you tune performance in production, help you scale, and free you up to focus on your next release.

MongoDB Training helps you become a MongoDB expert, from design to operating mission-critical systems at scale. Whether you're a developer, DBA, or architect, we can make you better at MongoDB.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.

Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Enterprise Download (mongodb.com/download)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Stitch backend as a service (mongodb.com/cloud/stitch)

mongoDB

US 866-237-8815 • INTL +1-650-440-4474 • info@mongodb.com