MongoDB Database 4.4: Guide to What’s New

June 2020
# Table of Contents

Introduction 1

MongoDB Query Language and Drivers 1
  Union 1
  Custom Aggregation Expressions 2
  Additional MQL and Driver Enhancements 2

Scale-Out Flexibility and Performance 3
  Refinable Shard Keys 4
  Compound Hashed Shard Keys 4
  Hedged Reads for Consistent Low Latencies 4
  Additional Enhancements 5

Resilience and Security 5
  Mirrored Reads 5
  Resumable Initial Sync 5
  Additional Resilience Enhancements 6
  AWS IAM Authentication 6
  Additional Security Enhancements 6

MongoDB Cloud Platform 7
  MongoDB Atlas Data Lake 7

Getting Started with MongoDB 4.4 8

We Can Help 8

Resources 9
Introduction

MongoDB 4.4 delivers the features and enhancements most demanded by you. We think of 4.4 as "user-driven engineering", building on the MongoDB 4.x release family as the ideal foundation for the future of data.

The result is a database that enables you to build transactional, operational, and analytical applications faster and more efficiently than any other. You can scale them out globally, with the flexibility to define and refine data distribution at any time as your requirements evolve. All while giving you the most sophisticated latency, resilience, and security controls anywhere.

MongoDB 4.4 is currently a beta release, ready for evaluation today. Key highlights of 4.4 are discussed through this What’s New guide.

MongoDB Query Language and Drivers

The MongoDB Query Language (MQL) is rich and expressive, enabling you to build almost any class of transactional and analytical application with a query language that is idiomatic to modern programming languages.

A key part of MQL is the Aggregation Pipeline. With dozens of stages and over 150 operators and expressions, you can process, transform and analyze data of any structure at scale, using any programming language. MongoDB 4.4 continues to enhance the power of the Aggregation Pipeline.

Union

Through the new $unionWith aggregation pipeline stage, you can now blend data from multiple collections into a single result set in the database, enabling deeper exploration and analysis of your data.

To illustrate the power of Union consider a time-series application that stores data in separate collections, with each bucketed by month. Creating a union that blends those monthly collections into a single result set enables you to analyze trends over longer time frames without having to return data to the application and creating the
union there, or moving data into a data warehouse. You have less code to write, no data movement incurring infrastructure overhead, no data staleness comprising query results – providing your users with fresher and faster insights.

$unionWith gives you the same functionality as the SQL UNION ALL operator, but takes advantage of MongoDB’s flexible document data model and distributed systems design. This means your source collections aren’t constrained by needing matching schemas, and Union aggregations can be scaled-out across collections spanning multiple shards.

Custom Aggregation Expressions

You can extend the functionality of MongoDB for your specific use cases by defining your own custom expressions in JavaScript and now have them execute on the database server as part of an aggregation pipeline.

There are many use-cases for Custom Aggregation Expressions. Some of the most common include filtering and projecting complex computed fields for data analysis, i.e. creating custom arithmetic and accumulator expressions, or for complex ETL pipelines.

By co-locating the execution of your custom expressions with your data server-side, you reduce the amount of data you need to pull back to the application, delivering higher performance to your users. Before 4.4, you could execute custom expressions via MapReduce in MongoDB. By now making those expressions available as part of an aggregation pipeline, you get improved ease-of-use through simplified and more efficient syntax and execution.

Additional MQL and Driver Enhancements

New Aggregation Operators and Expressions

Simplifying your code, it’s now easier to search and manipulate strings and evaluate the first and last elements of an array. To simplify schema analysis and performance optimization, you can use new expressions to return the size of binary or BSON objects in your documents. Learn more about the new aggregation pipeline operators in MongoDB 4.4.

Connection Monitoring and Pooling

Enhancing developer ease-of-use and the observability of your application under increased load, we have standardized the way MongoDB drivers configure and manage connection pooling behavior.
Each driver exposes an API that allows you to subscribe to events emitted from the connection pool, providing you visibility into when the driver opens and closes connections, and clears the pool. Through the API, you can configure connection pool behavior: defining the minimum and maximum number of connections, the maximum idle time, and how long a thread must wait for a connection to become available before throwing an exception. Review the full connection monitoring and pooling spec to learn more.

Global Read and Write Concerns

You can configure read isolation and write durability guarantees that are applied cluster-wide. This allows you to set the most appropriate defaults server-side for your application, maintaining consistent behavior across all your clients.

But it's not all or nothing – you still have the flexibility to override the default on a per-client or per-session basis in the driver.

New MongoDB Drivers

- **MongoDB Rust Driver GA**: Rust is gaining incredible momentum and we are keeping up the pace. Following the alpha release in December of 2019, the MongoDB Rust Driver is officially GA. The highlight of this new GA release is async support, including asynchronous connections and API as well as non-blocking server selection, enabling developers to write safe and fast code more efficiently than ever before.

- **MongoDB Swift Driver GA**: The new driver enables developers to apply their existing front-end skills to back-end code, creating a universal programming experience. The MongoDB Swift Driver supports the async API for fast and safe I/O, an ergonomic BSON API to make working with documents more effortless, including CRUD operations, the full breadth of aggregation pipeline operators and stages, along with serialization and conversion to Swift data types.

New MongoDB Developer Tools

- **New MongoDB Shell (beta)**: Connect to MongoDB to work with your data and configure your database. With its enhanced usability features (command autocomplete and syntax highlighting, easy to understand error messages and contextual help), it's the quickest way to connect to MongoDB and Atlas and work with your data and your cluster configuration.

- **MongoDB for VS Code (beta)**: Easily work with MongoDB directly from within your VS Code environment. By using the MongoDB extension available from the Visual Studio Code Marketplace, you can interact with MongoDB and Atlas in a way that fits into your native workflow and development tools.

- **MongoDB Compass 1.22 (beta)**: As the GUI for MongoDB, the latest Compass release embeds the MongoDB shell and adds support for the new 4.4 features in the Aggregation Pipeline Builder along with the latest MongoDB Atlas and Atlas Data Lake enhancements, discussed later in this Guide.

Scale-Out Flexibility and Performance

MongoDB's native sharding is highly valued by developers. It enables you to horizontally scale-out your database across multiple nodes and geographic regions to accommodate application growth and to colocate data with your distributed applications and users. It's elastic so you scale your database cluster out and in at any time to align consumption with resource requirements, and it's transparent to your applications.

MongoDB sharding is also highly flexible, allowing data to be partitioned by range, hash, or zone – the latter implemented as Global Clusters in our fully-managed MongoDB Atlas global cloud database service. With this flexibility, you can shard your database to match both your query patterns and data residency needs. This is something you don't get from most other distributed databases that purely hash a partition key, resulting in data being randomly sprayed across a cluster.

Even with the flexibility of MongoDB sharding, the inability to modify either the shard key or shard key value has been a frustration for some developers. This is because choosing a shard key that could not accommodate changing application requirements could result in data being...
unevenly distributed across the cluster, leading to inefficient use of provisioned resources and potentially poor performance.

Through developments in both the prior MongoDB 4.2 release and the new 4.4 release, sharding becomes even more flexible and adaptable:

- **MongoDB 4.2** introduced the ability to modify shard key values using a distributed, multi-document ACID transaction to change the placement of a document in a cluster. This is useful when you want to move a document to a different geographic region or age data out to a slower storage tier.

- **MongoDB 4.4** brings the ability to refine the shard key itself, and to create compound hashed shard keys to allow for an even more fine-grained distribution of data within the cluster. No other distributed database gives you this level of flexibility.

With these enhancements, sharding your database is now even easier, giving you more flexibility in selecting your shard key and making ongoing scaling operations more seamless as your applications evolve.

### Refinable Shard Keys

With the ability to define and refine your shard keys at any time, you can now adapt data distribution across your cluster as your database grows and applications evolve, without impacting system availability.

You can start scaling-out with a simple data distribution strategy, and then refine the shard key as needed by adding a suffix to it. For example, you may select `customer_id` as the shard key for your orders collection. As customers place more orders you want to split new order data evenly across more shards to accommodate growing sales volumes. What you can do is add `order_id` as a suffix to the shard key to provide a more evenly balanced distribution of the orders collection across the cluster.

Refining the shard key is non-blocking and transparent, imposing no impact to application availability.

Even with refinable shard keys, it is still important to properly evaluate selection of your shard key. No amount of refinement can overcome the selection of a poor initial shard key. The documentation provides guidance on choosing an appropriate shard key.

### Compound Hashed Shard Keys

By adding support for hashing any single field in a compound shard key, you get higher database throughput by more evenly distributing load across shards, without losing data locality or creating hot shards as your applications scale.

You have the flexibility to hash either the suffix or the prefix of the compound shard key. This gives you more fine-grained and even partitioning of documents for specific use cases, such as:

1. When data is distributed across a subset of shards that are co-located within a geographic region for data residency.
2. When a field of your preferred shard key has a monotonic value, such as 'customer_id' where your newest customers are also the most active, resulting in most traffic hitting a single shard.

Prior to MongoDB 4.4, you could only hash a single-field shard key. To accommodate the use cases above, you had to implement hashing on the application side, and store the hash within the document; a non-hashed compound shard key would then be defined across the client-hashed field and one or more other fields.

Bringing support for hashed fields in compound shard keys creates more scalable and performant applications by balancing read and write distribution with data locality, while reducing application and schema complexity.

### Hedged Reads for Consistent Low Latencies

High latency directly correlates to lost revenue. A Google study measured bounce rates increasing 50% when page load times exceeded 3 seconds, while Akamai found that each 100ms website delay resulted in ecommerce conversion rates declining by 7%.

To minimize p95 and p99 latencies, the MongoDB query router can be configured with Hedged Reads that submit
read requests to multiple replicas in a sharded cluster, returning results to the client as soon as the quickest node responds.

When you specify read preference nearest to the driver, the mongos query router will automatically hedge reads to the two closest nodes, measured by network latency. This avoids queries waiting on one node that could be busy syncing to disk, applying an index build, or where there is a transient network or server issue.

As a result of Hedged Reads, your users get higher and more predictable performance by eliminating long-tail query latency outliers.

Additional Scalability and Performance Enhancements

Streaming Replication

Oplog messages are now continuously streamed from the primary to secondary nodes, improving the freshness of secondary reads and reducing latency for applications using the majority committed write concern.

Prior to MongoDB 4.4, secondary replicas would poll the primary, receiving new oplog messages in 16MB batches. The secondary would then have to send a getMore command and wait on a network round-trip before receiving the next batch of messages.

While "mileage may vary" based on your specific environment, internal tests of streaming replication have measured performance improvements of up to 50% over high load, high latency networks using w:majority writes.

Hidden Indexes

Making it faster and more efficient for you to tune performance as your application evolves, you can now hide an index from the query planner to evaluate the impact of removing it, without actually dropping the index.

If you later determine the index is required, then you simply unhide it, avoiding the expense of a full index rebuild. The hidden index is fully maintained alongside all other indexes on your collection, and can be hidden and unhidden on-demand.

Simultaneous Indexing

New indexes are now built concurrently on the primary and all secondaries, and are available when all eligible nodes in the replica set have successfully created the index. With this new approach, chances of replication lag are dramatically reduced, ensuring queries targeting secondaries are served with fresher data.

Resilience and Security

Maintaining application uptime and secure access to data are non-negotiable for pretty much every business and developer today. MongoDB continues to enhance these capabilities in the latest 4.4 release.

Mirrored Reads

Reducing user impact of replica set elections following a primary failure or planned maintenance, MongoDB keeps the caches of secondaries warm by continuously mirroring a subset of reads to them.

Prior to mirrored reads, a secondary may not have had the latest data served by the primary in its memory, resulting in performance degradation immediately after it's election.

Now the electable secondary replicas are warmed with the most recently accessed data on the primary, enabling the cluster to quickly restore steady-state application performance after an election. By default 1% of all reads are mirrored. You can dial this percentage up – recognizing that this will increase load on the replica set – or you can deconfigure mirrored reads entirely.

Resumable Initial Sync

The initial sync process now automatically resumes after encountering transient network errors, avoiding the process having to start over from the beginning.

This makes it easier and faster to scale-out your database by adding new replicas to the cluster or to recover nodes that have fallen too far behind other members of the replica set.
By default, the secondary tries to resume initial sync for 24 hours – a timeframe that is user-configurable. If the secondary cannot successfully resume the initial sync process in the configured time, it selects a new healthy source node from the replica set and restarts initial sync from the beginning.

Additional Resilience Enhancements

- **Streamed topology changes**: When a new primary is elected after a node failure or maintenance event, cluster topology changes are now streamed back to the drivers in real time. As a result, clients can react immediately to cluster state changes, switching open connections as needed, reducing the user impact of primary elections.

- **Faster client authentication**: By cutting the round trips required to establish secure connections, the network latency clients will see when connecting to a new MongoDB primary is improved by up to 50%. This improvement has been achieved with multiple enhancements including enabling support for TCP Fast Open and removing redundant round trips in the SCRAM handshake. Like Mirrored Reads and Streamed Topology Changes, this also reduces the impact of primary elections.

- **Time-Based Oplog Retention**: You can now configure MongoDB’s oplog retention period by time, rather than just by its size. This additional flexibility is highly useful to DevOps teams who plan cluster maintenance around specific time windows and need to preserve oplog entries over this period. It is also useful for developers building reactive and event-driven services as it ensures they can resume a change stream from within a specific window of history.

AWS IAM Authentication

MongoDB already offers some of the complete and robust security controls of any database, along with support for regulatory compliance and data privacy standards in the cloud.

Further simplifying cloud-native security, your applications, containers, and serverless functions can now authenticate to MongoDB Atlas reusing existing regular and temporary Amazon IAM credentials.

Applications provisioned on EC2 instances, Docker containers managed by ECS, or serverless functions running on AWS Lambda can automatically obtain IAM credentials from local metadata, using them to authenticate to MongoDB Atlas, just as you can for any AWS-native service. You no longer have additional configuration files and secrets to manage in the application layer, while also eliminating the need to coordinate secrets deployment, management, and rotation across separate authentication mechanisms.

![Figure 2: Seamless integration between your AWS services and MongoDB Atlas](image)

With support for AWS IAM Authentication, there is less effort on your part to create and maintain a secure database environment. You can build apps faster on Atlas and AWS with the peace of mind and simplicity of a single, trusted security infrastructure. Review the Atlas Security Features and Setup documentation to learn more.

Additional Security Enhancements

- **TLS 1.3**: Starting with 4.4, MongoDB supports TLS 1.3 when used with a compatible OpenSSL library. TLS 1.3 and the accompanying forward secret ciphers can further help meet strict encryption-in-motion requirements. TLS 1.3 has also been backported to all supported MongoDB releases.

- **OCSP Stapling**: MongoDB can now confirm certificate validity by directly querying the CA over OCSP, eliminating the need to periodically download a Certificate Revocation List, and then restarting the server to refresh its view of certificate status.
• **Kerberos Configuration Utility:** The new `mongokerberos` command line utility automatically runs a series of checks for common issues with your MongoDB and Kerberos configuration, providing you with recommendations on the corrective actions needed to successfully deploy your cluster. This utility is available with MongoDB Enterprise Advanced.

**MongoDB Cloud Platform**

The best way to run MongoDB 4.4 is in the Atlas database service, part of the MongoDB Cloud Platform. The release of 4.4 is timed with a number of new features and enhancements available to you in the MongoDB Cloud:

• **Atlas Auto-Scale GA:** Enable intelligent, fully automated capacity management of your database in MongoDB Atlas and have your cluster resources **auto-scale** in response to workload changes. Atlas monitors key metrics in real-time and adjusts cluster compute and storage based on predictive modeling and proven practices from managing tens of thousands of MongoDB deployments.

• **Atlas Index and Schema Recommendations:** Proactive performance optimization with intelligent and automated recommendations delivered to you in real time. The **Performance Advisor** in MongoDB Atlas offers actionable index and schema advice based on analysis of your database metadata and logs.

• **Atlas Search GA:** Makes it easy to build fast, relevance-based **full-text search** capabilities on top of your data in the cloud. Create powerful data processing pipelines that combine search with the rich MongoDB Query Language (MQL), all with no additional infrastructure, data movement, or systems to manage.

• **Atlas Online Archive:** Tier your data across fully managed databases and cloud object storage, with the ability to query it together through a single endpoint. By automatically archiving historical data, you save on transactional database storage costs without compromising on making that data queryable. Atlas Online Archive is currently a beta release.

• **MongoDB Charts:** The new Embedding SDK provides the most powerful way to render and control charts directly within your application, making it easy to enrich your applications with live data visualizations of your MongoDB data. By using the Embedding SDK, developers can create integrated data experiences for their end users without the need to deal with additional database connections, queries, or complex charting libraries in a safe and secure way. With “Fast Charts”, data from chart queries can now be cached server-side, dramatically improving the rendering time for visualizations that are being accessed by multiple users while also reducing server load.

• **Realm Mobile Database:** Allows developers to store data locally on iOS and Android devices using a rich data model that's intuitive to them. Combined with the MongoDB Realm sync-to-Atlas, Realm makes it simple to build reactive, reliable apps that work even when users are offline.

• **MongoDB Realm:** Enables developers to validate and build key features quickly. Application development services like Realm Sync for mobile and Realm's GraphQL service, can be used with Realm Functions, Triggers, and Data Access Rules – simplifying the code required to build secure and performant apps.

• **MongoDB CLI:** The new CLI for the MongoDB Cloud Platform provides an interface that is more approachable, interactive, and easily scriptable than the existing REST APIs designed for programmatic integration. It is ideal for both ad hoc task execution and the scripting of recurrent operations. You can also quickly switch between managing your Atlas environments and your self-hosted clusters managed by Ops Manager or Cloud Manager.

**Ops Manager** is the best way to run MongoDB on-premises and in your own private cloud. Ops Manager 4.4 supports the latest release of the database, and includes new functionality to further simplify operations:

• The new Red Hat-certified **Ops Manager custom resource** allows you to run Ops Manager itself within Kubernetes, alongside your database and application resources, all controlled from a single K8s control plane.

• **Ops Manager Backup** adds support for point-in-time restore of 4.4 and 4.2 clusters, to query backups in-place, and incremental snapshots stored to your S3 bucket – fully integrated with AWS IAM.
Finally, a new MongoDB Community Kubernetes Operator makes it easy for developers and teams to deploy simple MongoDB clusters in containers.

MongoDB Atlas Data Lake

Announced in June 2019, the MongoDB Atlas Data Lake is now GA. With Atlas Data Lake you can query and analyze data across S3 and MongoDB Atlas in its native format using the MongoDB Query Language (MQL). You can combine your live and historical data without data movement and work with complex data immediately without the need for data transformation.

By leveraging a serverless, scalable query service you control costs and remove the operational burden to unlock value from your data faster. Aligning with GA, Atlas Data Lake offers a number of new features:

- **Federated Query**: Allows you to run a single query to analyze your live MongoDB Atlas data and historical data on Amazon S3 together and in-place, returning a single query response.
- **$out to S3**: You can persist the results of your aggregation pipelines to S3 in a read optimized structure, and move live data from your MongoDB Atlas clusters for a cost-effective data tiering strategy.
- **$out to Atlas**: You can persist the results of complex aggregations run in Atlas Data Lake back to your live MongoDB Atlas cluster, exposing result sets to your real-time transactional and operational applications.
- **SQL Support**: Query your live MongoDB Atlas data and historical data on S3 using SQL, and connect to your favorite BI tools for data visualization and analysis.

Safe Harbor

The development, release, and timing of any features or functionality described for our products remains at our sole discretion. This information is merely intended to outline our general product direction and it should not be relied on in making a purchasing decision nor is this a commitment, promise or legal obligation to deliver any material, code, or functionality.

We Can Help

We are the company that builds and runs MongoDB. Over 17,000 organizations rely on our commercial products. We offer cloud services and software to make your life easier:

- **MongoDB Atlas** is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or Google Cloud with best-in-class automation and proven practices that guarantee availability, scalability, and compliance with security standards.
- **MongoDB Enterprise Advanced** is the best way to run MongoDB on your own infrastructure. It’s a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.
- **MongoDB Atlas Data Lake** allows you to quickly and easily query data in any format on Amazon S3 using the MongoDB Query Language and tools. You don’t have to move data anywhere, you can work with complex data immediately in its native form, and with its fully-managed, serverless architecture, you control costs and remove the operational burden.
- **MongoDB Charts** is the best way to create, share and embed visualizations of MongoDB data. Build visualizations

Getting Started with MongoDB 4.4

MongoDB 4.4 is currently a beta release and ready for evaluation today. There are multiple ways you being your evaluation:

- Spin it up in the cloud using the on-demand, fully-managed and global MongoDB Atlas database service.
- Download it to run on your own infrastructure (select 4.4.x under Version).

MongoDB 4.4 packs the features and improvements most demanded by you. An ever richer query language, the flexibility to define and refine data distribution at any time, with the most sophisticated latency, resilience, and security controls anywhere, it is the ideal foundation for modern applications.
quickly and easily to analyze complex, nested data. Embed individual charts into any web application or assemble them into live dashboards for sharing.

**Realm Mobile Database** allows developers to store data locally on iOS and Android devices using a rich data model that’s intuitive to them. Combined with the MongoDB Realm sync-to-Atlas, Realm makes it simple to build reactive, reliable apps that work even when users are offline.

**MongoDB Realm** allows developers to validate and build key features quickly. Application development services like Realm Sync for mobile and Realm’s GraphQL service, can be used with Realm Functions, Triggers, and Data Access Rules – simplifying the code required to build secure and performant apps.

**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 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](http://mongodb.com) or contact us at [sales@mongodb.com](mailto:sales@mongodb.com).

- Case Studies ([mongodb.com/customers](http://mongodb.com/customers))
- Presentations ([mongodb.com/presentations](http://mongodb.com/presentations))
- Free Online Training ([university.mongodb.com](http://university.mongodb.com))
- Webinars and Events ([mongodb.com/events](http://mongodb.com/events))
- Documentation ([docs.mongodb.com](http://docs.mongodb.com))
- MongoDB Atlas database as a service for MongoDB ([mongodb.com/cloud](http://mongodb.com/cloud))
- MongoDB Enterprise Download ([mongodb.com/download](http://mongodb.com/download))
- MongoDB Stitch Serverless Platform ([mongodb.com/cloud/stitch](http://mongodb.com/cloud/stitch))
- MongoDB Realm ([mongodb.com/realm](http://mongodb.com/realm))