



Intelligently Manage and Store Data Where You Want With MongoDB

Increasingly, data is stored in a public cloud as companies realize the agility and cost benefits of running on cloud infrastructure instead of maintaining their own data centers. At any given time, however, organizations must know where their data is located, replicated, and stored — as well as how it is collected and processed to constantly ensure personal data privacy.

Giving you the freedom to run anywhere

Creating a proper structure for storing your data just where you want it can be complex, especially with the shift towards geographically dispersed data and the need to comply with local and regional privacy and data security requirements. Organizations without a strong handle on where their data is stored potentially risk millions of dollars in regulatory fines for mishandling data, loss of brand credibility, and distrust from customers.

Geographically dispersed data and various compliance regulations also impact how organizations design their applications, and many see these challenges as an opportunity to transform how they engage with data.

For example:

- Organizations get the benefits of a multi-cloud strategy and avoid vendor lock-in, knowing that they can still run on-premises or on a different cloud provider
- MongoDB runs where you want your data to be – on-prem, in the cloud, or as an on-demand, fully managed global cloud database
- A flexible data model is needed to keep data within the confines of the country or region where the data originates

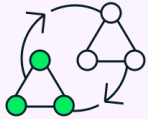
Major considerations for managing data

When managing data, organizations must answer questions in several key areas, including:

- **Process:** How is your company going to scale security practices and automate compliance for the most prevalent data security and privacy regulatory frameworks?
- **Penalties:** Are your business leaders fully aware of the costs associated with not adhering to regulations when storing and managing data?
- **Scalability:** Do you have an application that you anticipate will grow in the future and can scale automatically as demand requires?
- **Infrastructure:** Is legacy infrastructure keeping you from being able to easily comply with data regulations?
- **Flexibility:** Is your data architecture agile enough to meet regulations quickly as they grow in breadth and complexity?
- **Cost:** Are you wasting time and money with manual processes when adhering to regulations and risking hefty fines related to noncompliance?

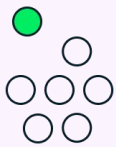
How companies use MongoDB to store data where they want and need it

When storing and managing data in different regions and countries, organizations must understand the rules and regulations that apply. MongoDB is uniquely positioned to support organizations to meet their data goals with intuitive security features and privacy controls, as well as the ability to geographically deploy their data clusters and backups in one or several regions.



Zones in Sharded Clusters

MongoDB uses [sharding](#) to support deployments with very large data sets and high-throughput operations. In sharded clusters, you can create zones of sharded data based on the shard key, which helps improve the locality of data.



Network Isolation and Access

Each MongoDB Atlas project is provisioned into its own virtual private cloud ([VPC](#)), thereby isolating your data and underlying systems from other MongoDB Atlas users. This approach allows businesses to meet data requirements while staying highly available within each region. Each shard of data will have multiple nodes that automatically and transparently failover for zero downtime, all within the same region.



Multi-Cloud Clusters

MongoDB Atlas is the only globally distributed, [multi-cloud](#) database. It lets you deploy a single cluster across AWS, Microsoft Azure, and Google Cloud without the operational complexity of managing data replication and migration across clouds. With the ability to define a geographic location for each document, your teams can also keep relevant data close to end users for regulatory compliance.



IP whitelists

[IP whitelists](#) allow you to specify a specific range of IP addresses against which access will be granted, delivering granular control over data.



Queryable Encryption

[Queryable encryption](#) enables encryption of sensitive data from the client side, stored as fully randomized, encrypted data on the database server side. This feature delivers the utmost in security without sacrificing performance and is available on both MongoDB Atlas and Enterprise Advanced.



MongoDB Atlas Global Clusters

Atlas global clusters allow organizations with distributed applications to geographically partition a fully managed deployment in a few clicks and control the distribution and placement of their data with sophisticated policies that can be easily generated and changed. Not only can your organization achieve compliance with local data protection regulations more easily, but you can also reduce overhead at the same time.



Client-side field-level encryption

[MongoDB's Client-Side Field Level Encryption \(FLE\)](#) dramatically reduces the risk of unauthorized access or disclosure of sensitive data. Fields are encrypted before they leave your application, protecting them everywhere: in motion over the network, in database memory, at rest in storage and backups, and in system logs.

Segmenting Data by Location with Sharded Clusters

As your application gets more popular, you will reach a point where your servers will reach their maximum load. Before you reach that point, you must plan for scaling your database to adjust resources to meet demand. [Scaling](#) can occur temporarily, with a sudden burst of traffic, or permanently with a constant increase in the popularity of your services.

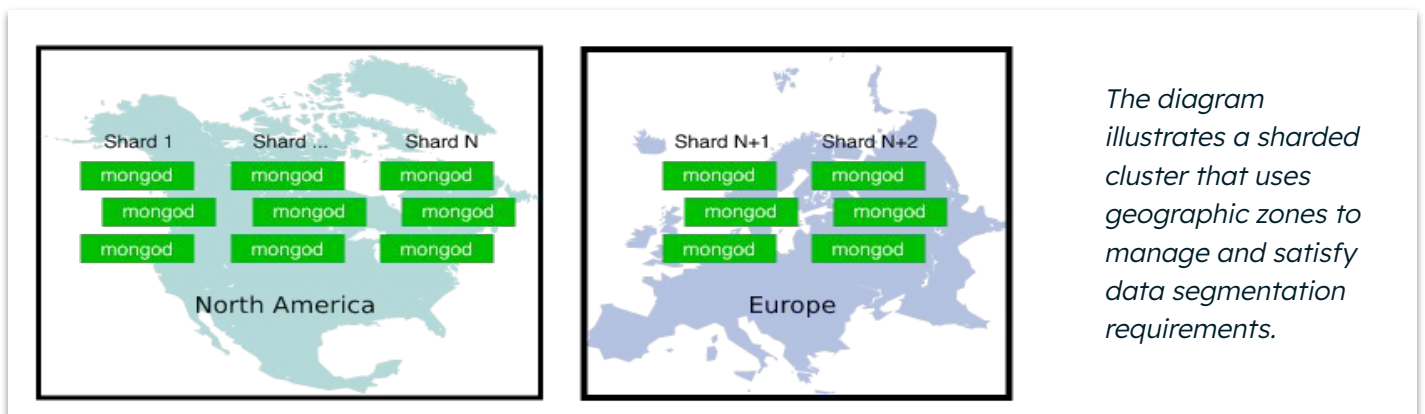
Increased usage of your application brings three main challenges to your database server:

- The CPU and/or memory becomes overloaded, and the database server either cannot respond to all the request throughput, or do so in a reasonable amount of time
- Your database server runs out of storage and thus cannot store all the data
- Your network interface is overloaded, so it cannot support all the network traffic received

When your system resource limits are reached, you will want to consider scaling your database. Horizontal scaling refers to bringing on additional nodes to share the load. This process is difficult with relational databases because of the difficulty in spreading out related data across nodes. With non-relational databases, this is made simpler because collections are self-contained and not coupled relationally. This approach allows them to be distributed across nodes more simply, as queries do not have to “join” them together across nodes.

[Horizontal scaling](#) with MongoDB Atlas is achieved through sharding. With sharded clusters, you can create [zones](#) of sharded data based on the [shard key](#). You can associate each zone with one or more shards in the cluster. A shard can be associated with any number of zones. In a balanced cluster, MongoDB migrates chunks covered by a zone only to those shards associated with the zone.

- If one of the data centers goes down, the data is still available for reads unlike a single data center distribution
- If the data center with a minority of the members goes down, the replica set can still serve write operations as well as read operations
- However, if the data center with the majority of the members goes down, the replica set becomes read-only



The diagram illustrates a sharded cluster that uses geographic zones to manage and satisfy data segmentation requirements.

Other benefits of MongoDB Atlas

MongoDB Atlas also provides you with an intuitive UI or administration [API](#) to efficiently perform tasks that would otherwise be very difficult. Upgrading your servers or setting up sharding without having to shut down your servers can be a challenge, but MongoDB Atlas [removes this layer of difficulty](#) through the features described here. With MongoDB, scaling your databases can be done with a couple of clicks.