Who Owns Security in the Cloud?

A breakdown of the shared responsibility model when using MongoDB Atlas

MongoDB.

Table of Contents

Introduction

Businesses once doubted the long-term benefits of moving their traditional, on-premises workloads to the cloud. But those doubts didn't last long. The ROI of cloud migration was realized quickly and decisively. Businesses eagerly shifted their workloads to the cloud and realized the financial benefits of doing so. However, in the rush to the cloud, clarity on security – and who is responsible for it – is often lost.

Before the cloud (and in current on-premises data centers), an organization owned and managed the entire environment and was responsible for implementing best practices to ensure the security and integrity of all the assets under its control. Cloud security follows a different approach. Some security responsibilities fall clearly on the organization, some on public cloud providers, and some on the vendors of the cloud services being used. This is known as the shared responsibility model.

Security in the cloud is only possible when everyone is clear on their roles and responsibilities. Shared responsibility recognizes that cloud vendors such as MongoDB must ensure the security and availability of their services and infrastructure, and that our customers must take appropriate steps to protect the data they keep in the cloud. The cloud providers are responsible for the security and availability of their infrastructure. These are the prerequisites for cloud security and resilience.

Cloud Service Models

Instead of businesses purchasing IT assets, they consume services through the cloud in pay-as-you-go or subscription models. This relieves them from the operational overhead associated with purchasing, maintaining, and upgrading on-premises infrastructure and software. Cloud resources are elastic – they scale out and in, often automatically, depending on demand. From a customer's perspective, these resources appear to be unlimited and can be provisioned in any quantity that's needed at any given moment.

There are three cloud service models in use today: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS):

- **IaaS:** Infrastructure is hosted within a cloud service provider's facility instead of the customer's data center. Compute, storage, and networking are delivered to customers on demand while being fully managed by the service provider.
- **PaaS:** A cloud service provider hosts the hardware and software on their own infrastructure allowing developers to focus on developing and running applications.
- **SaaS:** A ready-to-use, cloud-hosted application.

MongoDB Atlas is a fully managed cloud database that is offered through the software-as-a-service model. It is delivered through a distributed architecture, which allows you to deploy databases in more than 80 regions provided by our trusted cloud-service providers: AWS, Google Cloud, and Azure. Atlas is a SaaS offering, but it can be more specifically described as a database as a service (DBaaS).

Shared Responsibility in the Cloud

The security responsibilities for software, platform, and infrastructure services differ depending on the cloud service model being offered.

When a customer migrates certain aspects of their enterprise architecture to a cloud service provider, they share or, in some cases, relinquish certain responsibilities for securing those environments. A shared responsibility model is a technical abstraction of the security responsibilities held by customers and cloud providers.

Shared Responsibility With MongoDB

When you deploy a new database on Atlas, it's created on the infrastructure offered by cloud service providers. MongoDB is responsible for the security and availability of the services we offer, and for everything within the scope of our responsibilities as a SaaS vendor. Customers are responsible for the security of everything above the application layer (accounts, identities, devices, and data). Cloud providers are responsible for the operational health of their infrastructure, the operating system and virtualization layer, and the physical security of facilities where services operate. The concept of shared responsibility comes into play when customers interact with the tools we make available to them through the services we provide. (See figure 1.)

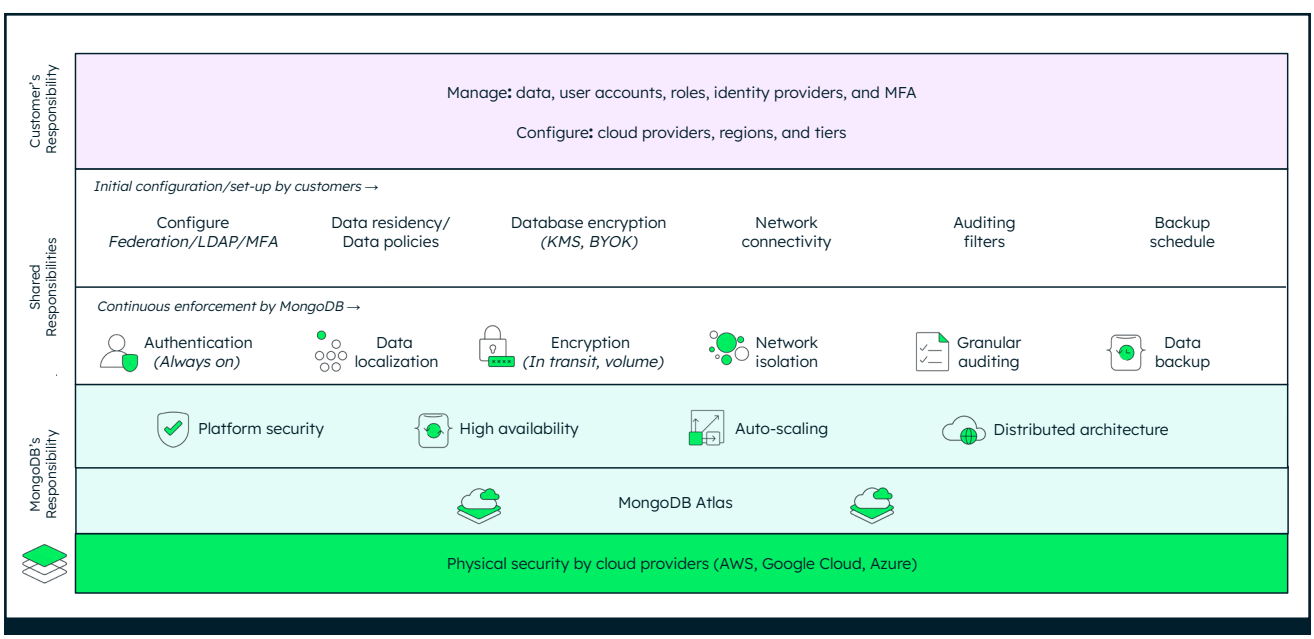

Figure 1: The MongoDB shared responsibility model

MongoDB's Responsibilities

MongoDB implements numerous safeguards to secure our services so customers can focus on fulfilling their business requirements. We enforce security policies for things like authentication and network isolation so developers can be comfortable that risks are appropriately governed while using our services. Atlas provides tools for ensuring secure best practices such as encryption, database access, auto-scaling, and granular auditing.

Platform Security

MongoDB Atlas is designed with strong security defaults so that the burden of securely using the service is minimized for the customer. These defaults include always-on authentication, authorization, encryption in transit, encryption at rest, and restricted access from the Internet by default. Additionally, MongoDB provides automatic patching and single-click upgrades.

MongoDB policies require the principle of least privilege and separation of duties. Developers are provided access to developer environments only and production environments are limited to personnel who have an operational need and appropriate authorizations.

High Availability

MongoDB ensures high availability for the services we provide. If a failure is detected on an Atlas primary node, a secondary node is automatically promoted with little to no interruption in service.

Maintenance releases are automatically applied through a rolling deployment, ensuring zero downtime and uninterrupted access to the cluster. Specific maintenance windows can be customized. Urgent maintenance activities, such as security patches, will be applied outside of these windows if necessary. MongoDB also allows zero-downtime upgrades for major versions and provides a runbook for testing and validating in a staging environment before performing the upgrade in production.

Auto-scaling

Auto-scaling reduces costs, strengthens resilience, and ensures business continuity. With auto-scaling, customers pay for total usage rather than capacity. Auto-scaling can also help protect against potential application and network failure. Auto-scaling policies help customers detect anomalous spikes in usage and replace any unhealthy instances that may lead to disruptions in service.

Distributed Architecture

MongoDB enables always-on availability through replica sets and native sharding. Built-in replication both within and across regions ensures high availability, even in the event of regional outages. Distributed data also allows for low-latency user access while enforcing data sovereignty controls for data privacy regulations such as GDPR.

Shared Responsibilities

MongoDB is responsible for continually enforcing critical security capabilities for customers so they can focus on application development and managing their business requirements. Developers don't have to customize their application to enforce higher security standards because the security controls of the database are configured by default. Developers are responsible for initial setup and configuration of key capabilities, so it's enabled and ready for use. Key capabilities like authentication, data encryption in transit and at rest, network isolation, data backup, and granular auditing have strict default security settings.

Authentication

Authentication control for Atlas clusters is enabled by default with the Salted Challenge Response Authentication Mechanism (SCRAM) and cannot be disabled. Where human operator intervention from MongoDB is authorized and necessary to maintain the availability of critical systems for customers, we have formal policies and procedures to guide, monitor, and limit involvement so as to maintain customer security and confidentiality. Under normal circumstances, MongoDB engineers are prohibited from accessing customer data. This would only happen under "break glass" reliability situations.

By default, we use a combination of technical and logical controls to limit access to systems with sensitive data. We do this by limiting access to underlying hosts to only privileged users, and having them authenticate using multifactor authentication (MFA) and a bastion host.

Atlas infrastructure is only accessible through bastion hosts. Bastion hosts are configured to require SSH keys, not passwords. Bastion hosts also require MFA, and users must additionally be approved by senior management for backend access.

Encryption

In-transit and at-rest encryption are always on in Atlas and cannot be disabled. Atlas encrypts all cluster storage and snapshot volumes, ensuring that any stored data (data "at rest") is always secure.

Traffic from clients to Atlas (both database clusters and the web UI/control plane) is authenticated and encrypted in-transit, and traffic between the customer's internally managed MongoDB nodes is also authenticated and encrypted in-transit using TLS. TLS 1.2 is the default; customers can select TLS 1.1 or 1.0 if needed (but note that MongoDB 4.0 and later disables support for TLS 1.0 where TLS 1.1+ is available). The MongoDB Security Team continuously monitors the status of transport protocols, and requirements are continually updated in order to ensure weak ciphers are deprecated.

Atlas also supports client-side field-level encryption (FLE), which can ensure that data is never exposed in plaintext while on the server. With FLE, even a compromised administrator account reading memory dumps cannot decrypt your data. FLE is optional and must be configured by the customer.

Network Isolation

Atlas user data and underlying systems are fully isolated from other users. Database resources are associated with a user group, which is contained in its own virtual private cloud (VPC). Access must be granted by IP access lists, VPC peering, or private endpoints.

The following network ports are used by MongoDB and cannot be changed:

- 27017 for mongod (database server)
- 27016 for mongos (query router for sharded clusters)
- 27015 for the BI connector
- If LDAPS is enabled, MongoDB requires LDAPS network 636 on the customer side open to inbound traffic by Atlas

Data Backup

Atlas takes incremental backup snapshots of data in your cluster and allows you to restore from those snapshots. All data held in backups is encrypted and can only be accessed by authorized parties. If customers enable encryption key management integration with AWS KMS, Azure Key Vault, or Google Cloud KMS, then AWS Customer Master Key (CMK), Azure Key Vault Secret Key, Google Cloud Service Account Key, and IAM credentials are required to perform restores of backup snapshots. If a customer terminates an Atlas cluster, the backup associated with the managed cluster is also terminated. If a customer terminates backup, all snapshots become unavailable immediately.

Granular Auditing

Granular database auditing in Atlas allows administrators to answer detailed questions about systems activity by tracking all DDL, DML, and DCL commands against the database. All DML commands can be audited, including reads, creations, updates, and deletions. Admins can select the actions they want to audit, as well as the MongoDB users, Atlas roles, and LDAPS groups whose actions they want audited from the Atlas UI. A single auditing configuration applies to all database clusters within an Atlas project. When needed, audit logs can be downloaded in the UI or retrieved using the Atlas API.

Data Localization

Atlas allows customers to distribute workloads to different geographic regions and with different cloud providers to help mitigate regional or vendor-specific outages. Customers can choose the regions for their database clusters and storage. Atlas supports AWS, Azure, and Google Cloud regions globally. This includes U.S., EMEA, and APAC locations.

Configure LDAP/IDP/MFA

User authentication and authorization against Atlas clusters can be managed through a customer's Lightweight Directory Access Protocol (LDAPS) server over TLS. A single LDAPS configuration applies to all database clusters within an Atlas project. For customers running their LDAPS server in an AWS Virtual Private Cloud (VPC), a peering connection is recommended between that environment and the VPC containing their Atlas databases.

For the Atlas web UI, user credentials are stored using industry-standard and audited one-way hashing mechanisms. Additionally, customers can make MFA optional or require users in their Atlas organization to use MFA. MFA options include SMS, voice call, a multi-factor app, or a multi-factor device (such as a YubiKey). Sensitive customer data provided within the GUI, such as passwords, keys, and credentials that must be used as part of the service, are encrypted.

The Atlas web UI supports authentication using username and password, and MFA. Control plane user identities are managed in a MongoDB-controlled Okta instance, encrypted, and stored securely. Federated identity with SAML identity providers such as Okta or OneLogin is supported.

Backup Scheduling

Continuous backup in Atlas allows customers to restore business operations by quickly recovering their data, enabling the organization to meet regulatory and compliance obligations. Customers have a responsibility to manage their own business continuity and disaster recovery planning and define their own RTO and RPO according to their acceptable criteria (e.g., RTO/RPO of 0-4 hours), which can be achieved independently of the Atlas control plane through the use of specific product features available to customers. These features include:

- Selection of the underlying cloud provider(s) – AWS, Google Cloud, Azure – for deploying MongoDB clusters in order to mitigate the risk of a cloud provider failure
- Selection of one or more cloud provider(s) regions in order to mitigate the risk of a region failure
- Selection of a cluster tier – shared or dedicated, sharded or unsharded – to mitigate the impact of workload spikes
- Selection of network connectivity options to Atlas for high availability
- Selection of backup and restore options, and backup schedule

Database Encryption (KMS/BYOK)

Customers running Atlas may choose to "bring your own key" and enable database-level encryption for sensitive workloads through the WiredTiger Encrypted Storage Engine. Use of self-managed keys with the WiredTiger Encrypted Storage Engine can help customers achieve additional levels of confidentiality and data segmentation. The master key in the context of a customer's cloud service generates and decodes data keys. When the WiredTiger Encrypted Storage Engine is enabled for an Atlas project, customer databases can only be started or backed up when the customer's master key is active and valid. Once a master key is destroyed, all project cluster data becomes inaccessible and unrecoverable, including previously encrypted backups.

Network Connectivity

By default, Atlas clusters do not allow access from the internet. Each Atlas cluster is deployed within a VPC configured to prohibit inbound access by default. Customers connect to Atlas through either public IPs, which are protected with IP access lists, or private IPs through network peering or private endpoints. Customers can also connect to Atlas with a private endpoint using a one-way connection from their own VPC to the Atlas VPC. Atlas VPCs can't initiate connections back to customer VPCs. This ensures that the network trust boundary is not extended. Connections to private endpoints within the customer's VPC can be made transitively from another VPC peered to the private endpoint-connected VPC or an on-premises data center connected with DirectConnect to the private endpoint-connected VPC.

Audit Filters

Audit trails in Atlas log all access and actions executed against the database. The Atlas auditing framework captures administrative actions such as schema operations, authentication and authorization activities, and read and write operations to the database. Atlas allows administrators to audit all events triggered from the Atlas UI at the Project or Organization level. The log is available in the Atlas UI or the API. For dedicated clusters (M10 and above), Atlas provides an easy-to-read log of database authentication events – including both successes and failures – such as database user, source IP address, and timestamp. This can be accessed either within the Atlas UI or through the API.

Comply with Data Policies

Security policies serve as a guide for organizations to ensure protection and security for the data they collect, manage, and store. Security policies help organizations manage risks and comply with data protection standards and regulations. At MongoDB, we have our own security policies that we are obligated to follow. We have no view into the sensitivity and value of the data our customers store with us, but we are obligated to follow the guidelines established by our security policies nonetheless, just as our customers are responsible for following their organizations' policies.

Customer Responsibilities

While Atlas provides the tools and services for maintaining a highly secure and resilient data platform, it's up to customers to configure their systems and users to restrict access to sensitive data and implement best practices for keeping critical systems online. Customers are responsible for creating users and roles to access Atlas, selecting cloud providers and regions to create their clusters, and choosing the cluster type. They can optionally enable backup, configure advanced auditing, bring their own keys for storage engine encryption, and configure client-side field-level encryption.

Configure Cloud Providers, Regions, and Tiers

While Atlas provides tools for deploying workloads to different cloud providers, geographic regions, and data repositories, it's up to customers to configure their Atlas databases according to their business needs and in accordance with applicable laws and regulations, such as GDPR and CCPA.

We've made it easy to spin up Atlas databases by design, because we believe that developers need tools that make it easier to work with data. But it's important for customers to remember that any database holding sensitive information needs to be configured with appropriate security measures, like network connectivity and isolation, and access controls to the database. While MongoDB Community Server and free tiers make it easy for developers to start building applications, it's still critical to configure all databases with appropriate security. Even free databases should be secure databases.

Manage Data, Accounts, Users, Roles, Identity Providers, and MFA

Atlas is secure by default, which means it's up to the customer to configure user authentication and assign user and role privileges. Customers must use role-based access controls (RBAC) to create policies for who has access to data and for what reason. A user may be granted one or more roles that determine their access to database resources and operations. Outside of role assignments, the user has no access to the system. We encourage our customers to enhance their internal security with MFA and by constantly training their employees on good data security hygiene.

Develop ↔ Framework

Protect

Recover ↔ Detect

Shared Responsibility Model		
	Customer	**MongoDB**
Cloud infrastructure	• Select cloud provider, region & tier • Select MongoDB version and auto-scaling options	• Provision and deploy cluster in dedicated VPC and firewalls • Ensure configuration changes are applied without service disruption
Customer data, accounts and identities	• Provide and manage customer data • Maintain accounts and identities	• Provide secure access to customer data • Provide tools to upload/store data securely
Network isolation & connectivity	• Configure network connectivity, options include: IP access list; VPC peering connections; Private endpoints	• Provision peering containers • Provision private endpoint resources • Only allow connections to the cluster from entries in a project's access list
Database access	• Configure user authentication • Assign user and role privileges • Manage certification authority • Configure AWS IAM, LDAP integration • Configure Data API access keys	• Maintain always-on authentication (SCRAM, x509 certificates) • Provide role-based access controls (RBAC) with predefined roles • Provide audit log access
Atlas access	• Create users and access • Configure MFA and federated auth • Configure API keys	• Maintain always-on authentication • Provide integrations interface with identity providers, MFA tools
Data encryption *(in transit and at rest)*	• Configure cloud provider Key Management System (KMS) • Set minimum TLS version	• Encryption always on – in-transit, and at-rest • Ensure data is stored on encrypted storage volumes with cloud provider managed keys • Encrypt data at-rest using customer provided keys
Data encryption *(in use, BYOK)*	• Configure client side field-level encryption • Configure cloud provider KMS	• Provide tools, drivers and shared libraries for field-level encryption • Drivers to communicate with KMS
Granular auditing	• Enable granular database auditing • Configure audit filter	• Maintain database access history • Security events in the activity feed
Data locality	• Set up data locality rules • Enforce data locality rules	• Support multiple regions and global clusters • Ensure the cloud snapshots for backup are located in the origin regions
Performance troubleshooting	• Enable/disable performance advisor and query profiler • Review and apply performance advisor recommendations	• Automatically run query profiler and performance advisor
Security patches and maintenance	• Set maintenance window	• Apply minor version upgrades • Apply security patches
Monitoring & alerting	• Configure alert thresholds • Enable real-time performance panel	• Collect monitoring metrics • Proactively monitor cluster health metrics
Backups	• Configure backup policy and retention • Configure point in time restore	• Operate backups according to policy • Ensure backup retention according to policy
Online archive	• Enable automatic archival • Configure archiving rules • Configure query patterns	• Provision archival storage and manage archival data format • Ensure automatic scalability of Online Archive federated query engine

Figure 2: Customer and MongoDB responsibilities in MongoDB Atlas

Compliance and Privacy

As your trusted supplier, we know that your security is our security, and vice versa. We take great care in the security of our services and infrastructure, and with your trust in us. For these reasons, MongoDB Atlas undergoes annual independent verifications of our platform security, privacy, and compliance controls, and we strive to improve our security posture daily. Our strong and growing focus on standards conformance and compliance is also meant to help you meet your regulatory and policy objectives. A brief summary of our certifications and attestations is listed below, and greater detail can be found on our MongoDB Trust Center.

- Schellman & Company, LLC assesses and certifies MongoDB against the following:
  - **ISO/IEC 27001:2013** mandates numerous controls for the establishment, maintenance, and certification of MongoDB's ISMS so as to help preserve the confidentiality, integrity and availability of the end to end Customer Sensitive Information (CSI) flows.
  - **ISO/IEC 27017:2015** demonstrates MongoDB cloud service security to users to demonstrate the confidentiality, integrity and availability of end to end Customer Sensitive Information (CSI) flows.
  - **ISO/IEC 27018:2019** establishes control objectives, controls and guidelines for implementing measures to protect Personally Identifiable Information (PII) in MongoDB cloud. It also provides a set of additional controls and associated guidance intended to address public cloud PII protection requirements not addressed by the existing ISO/IEC 27002 control set.
  - **SOC 2 Type II** is an auditing procedure designed to ensure that MongoDB securely manages data to protect the interests of our organization and the privacy of our clients.
  - **HITRUST CSF** is an independent security and compliance framework that is based in part on the HIPAA regulations. HITRUST CSF is built on the concepts of the ISO 27001, SOC 2 Type II (Confidentiality, Availability and Security Principles) and HIPAA regulations. Instead of pursuing HITRUST CSF Certification, MongoDB has opted for a SOC 2 + HITRUST certification report for MongoDB's cloud services, issued by an independent auditor. This report maps the controls of MongoDB's SOC 2 Type II report to the HITRUST CSF.
- Coalfire Systems Inc. assess and certifies MongoDB against the following:
  - **PCI DSS** is an information security standard developed by the PCI Standards Security Council, and applies to all entities that store, process, and/or transmit cardholder data. MongoDB Cloud has achieved PCI DSS 3.2.1 certification as of September 8, 2020.
- The Cloud Security Alliance (CSA) CSA STAR Level 2:
  - **The CSA Security, Trust, Assurance, and Risk (STAR) Registry** is a publicly accessible registry that documents the security and privacy controls provided by popular cloud computing offerings. MongoDB has achieved both CSA STAR Level 1 and CSA STAR Level 2 by maintaining a Consensus Assessments Initiative Questionnaire (CAIQ) for MongoDB Atlas and receiving a STAR Certification after a third-party audit of MongoDB Atlas, based on ISO/IEC 27001:2013 together with the CSA Cloud Controls Matrix (CCM).
- Health Insurance Portability and Accountability Act of 1996 (HIPAA):
  - **HIPAA** is a United States legislation that provides data privacy and security provisions for safeguarding medical information, also known as personal health information (PHI). There is no certification for HIPAA. Under the HIPAA regulations, database service providers such as MongoDB are considered business associates. The Business Associate Addendum (BAA) is a MongoDB contract that is required under HIPAA regulations to ensure that MongoDB appropriately safeguards PHI. The BAA also serves to clarify and limit the permissible uses and disclosures of PHI by MongoDB. MongoDB has a standard BAA that we present to customers for signature upon contracting our services.

Conclusion

At MongoDB, our overriding mission is to make data easier to work with. This can't happen if data becomes compromised for any reason. We've designed our cloud services to be secure from day one. But we also provide the tools customers need to configure their environments for their own requirements. As more organizations move to the cloud, it's imperative for customers to know who's responsible for what when it comes to cloud security. Understanding these roles and responsibilities is crucial for ensuring cloud workloads remain secure and available.

To get started with MongoDB Atlas, sign up for a free account today.