

Technical and Organizational Security Measures

[Register Now](#)

Please register to be notified of any changes to this page. If a change occurs, you will receive an email to the address that you provide.

Last updated: August 31, 2022. To see what has changed, [click here](#).

These Technical and Organizational Security Measures (“**Security Measures**”) are incorporated into and form part of your applicable agreement with MongoDB with respect to your use of MongoDB Atlas (the “**Agreement**”). These Security Measures also apply to MongoDB Atlas for Government, as modified by the MongoDB Atlas for Government Addendum to the Agreement.

The Security Measures set out the security features, processes, and controls applicable to MongoDB Atlas, including configurable options available to Customer, which employ industry standard information security best practices.

1. Definitions

The following terms have the following meanings when used in the Security Measures. Any capitalized terms that are not defined in the Security Measures have the meaning provided in your Agreement.

1.1. "**Cloud Provider**" means Amazon Web Services (AWS), Microsoft Azure (Azure), or Google Cloud Platform (GCP), as selected by Customer.

1.2. "**Customer Data**" means any data you or your end users upload into MongoDB Atlas.

1.3. "**Data Breach**" means a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to Customer Data.

1.4. "**Information Security Program**" means MongoDB’s written security

program, policies, and procedures that set forth the administrative, technical, and physical safeguards designed to protect Customer Data.

1.5. **"MongoDB Atlas Cluster"** means each replica set or sharded cluster of data-bearing nodes running the MongoDB database software that is managed by MongoDB Atlas, subject to your selected configurations.

1.6. **"MongoDB Atlas Project"** means one or more associated MongoDB Atlas Clusters with a shared set of authorization and network configurations.

1.7. **"MongoDB Systems"** means MongoDB's internal infrastructure, including development, testing, and production environments, for MongoDB Atlas.

1.8. **"Privileged User"** means a select MongoDB employee or third-party contractor who has been granted unique authority to access Customer Data or MongoDB Systems as required to perform their job function.

1.9. **"Security Incident Response Plan"** means MongoDB's documented protocols for evaluating suspected security threats and responding to confirmed Data Breaches and other security incidents.

2. Information Security Program Overview.

2.1. **General.** MongoDB maintains a comprehensive written Information Security Program to establish effective administrative, technical, and physical safeguards for Customer Data, and to identify, detect, protect against, respond to, and recover from security incidents. MongoDB's Information Security Program complies with applicable Data Protection Law and is aligned with the NIST Cyber Security Framework (NIST). Additionally, MongoDB Atlas is certified against ISO 27001:2013, ISO 27017:2015, ISO 27018:2019, SOC 2 Type II, Payment Card Industry Data Security Standard v.4, and Cloud Security Alliance (CSA) Security, Trust, Assurance, and Risk (STAR) Level 2. MongoDB Atlas has also undergone a HIPAA examination validated by a qualified third-party assessor and can be configured to build HIPAA compliant applications.

2.2. **Maintenance and Compliance.** MongoDB's Information Security Program is maintained by a dedicated security team, led by our Chief Information Security Officer. MongoDB monitors compliance with its Information Security Program, and conducts ongoing education and training of personnel to ensure compliance. The Information Security Program is reviewed and updated at least annually to reflect changes to our organization, business practices, technology, services, and applicable

laws and regulations. We will not alter or modify the Information Security Program in a way that materially weakens or compromises the effectiveness of its security controls.

2.3. MongoDB Personnel Controls.

2.3.1. Background Checks. MongoDB performs industry standard background checks on all MongoDB employees as well as any third-party contractor with access to Customer Data or MongoDB Systems.

2.3.2. Personnel Obligations. Any Privileged User authorized to access Customer Data is required to commit in writing to information security and confidentiality obligations that survive termination and change of employment. MongoDB maintains a formal disciplinary procedure for violations by MongoDB personnel of its security policies and procedures.

2.3.3. Training. Upon hire and subsequently at least once per year, Privileged Users authorized to access Customer Data undergo required training on specific security topics, including phishing, secure coding, insider threats, and the secure handling of Customer Data and personally identifiable information. Further, MongoDB implements mandatory, role-specific training for Privileged Users who are authorized to access Customer Data. MongoDB maintains records of training occurrence and content. In addition to these mandatory trainings, MongoDB offers employees additional training resources, such as internal security awareness and education groups and hackathons.

2.4. Third Parties. MongoDB maintains and adheres to a documented process for the evaluation and approval of third-party service providers prior to onboarding, which includes appropriate due diligence regarding each third party's security processes and controls. We require third parties to contractually commit to confidentiality, security responsibilities, security controls, and data reporting obligations, and we perform ongoing targeted due diligence on a quarterly basis.

2.5. Security Contact. If you have security concerns or questions, you may contact us via your normal Support channels, via support.mongodb.com, or by emailing security@mongodb.com.

3. MongoDB Atlas Security Controls.

3.1. Data Centers and Physical Storage. MongoDB Atlas runs on AWS, Azure, and GCP, and you control which Cloud Provider to use for deploying

your MongoDB Atlas Clusters. Each Cloud Provider is responsible for the security of its data centers, which are compliant with a number of physical security and information security standards detailed at the Cloud Provider's respective websites:

- <https://aws.amazon.com/security/>
- <https://www.microsoft.com/en-us/trustcenter/security/azure-security>
- <https://cloud.google.com/security/>

At least twice per year, each of our Cloud Providers is subject to due diligence performed by MongoDB or third-party auditors, which includes obtaining and reviewing security compliance certifications.

In addition to selecting which Cloud Provider to use, you also control the region where your MongoDB Atlas Clusters are deployed. This gives you the flexibility to decide where your Customer Data is physically stored, and you may choose to deploy your Customer Data in a specific geographic region (for example, only within the European Union or only within the United States).

3.2. Encryption.

3.2.1. Encryption in Transit. All MongoDB Atlas network traffic is protected by Transport Layer Security (TLS), which is enabled by default and cannot be disabled. Customer Data that you transmit to MongoDB Atlas, as well as Customer Data transmitted between nodes of your MongoDB Atlas Cluster, is encrypted in transit using TLS. You can select which TLS version to use for your MongoDB Atlas Clusters, with TLS 1.2 being the recommended default and a minimum key length of 128 bits.

3.2.1.1. Key Management Procedures for Encryption in Transit. All encryption in transit is supported by the use of OpenSSL FIPS Object Module. We maintain documented cryptography and key management guidelines for the secure transmission of Customer Data, and we configure our TLS encryption key protocols and parameters accordingly. MongoDB's key management procedures include: (i) generation of keys with approved key length; (ii) secure distribution, activation and storage, recovery and replacement, and update of keys; (iii) recovery of keys that are lost, corrupted, or expired; (iv) backup/archive of keys; (v) maintenance of key history; (vi) allocation of defined key activation and deactivation dates; (vii) restriction of key access to authorized individuals; and (viii) compliance with legal and

regulatory requirements. When a key is compromised, it is revoked, retired, and replaced to prevent further use (except for limited use of that compromised key to remove or verify protections). Keys are protected in storage by encryption and are stored separately from encrypted data. TLS certificates are obtained from a major, widely trusted third-party public certificate authority. In the course of standard TLS key negotiation for active sessions, ephemeral session keys are generated which are never persisted to disk, as per the design of the TLS protocol.

3.2.2. Encryption at Rest. Upon creation of a MongoDB Atlas Cluster, by default, Customer Data is encrypted at rest using AES-256 to secure all volume (disk) data. That process is automated by the transparent disk encryption of your selected Cloud Provider, and the Cloud Provider fully manages the encryption keys. You may also choose to enable database-level encryption via the WiredTiger Encrypted Storage Engine (using AES-256), as well as to bring your own encryption key with AWS Key Management Service (KMS), GCP KMS, or Azure Key Vault (KV).

3.2.3. Encryption in Use. MongoDB Atlas also supports automatic encryption of individual data fields of Customer Data before they are sent to MongoDB Atlas. If you enable this client-side field level encryption feature for a selected data field, an application-side component built into the MongoDB drivers encrypts that field of Customer Data before leaving the driver to be sent to MongoDB Atlas, and only decrypts it upon return to the application once inside the driver. With respect to the Customer Data for which you enable client-side field level encryption, MongoDB Atlas never sees your unencrypted Customer Data and you control the encryption keys, which you can secure using any KMIP-compliant key management service.

3.3. Network Connectivity Options.

3.3.1. Network Isolation. You may choose to deploy your MongoDB Atlas Clusters in a dedicated virtual environment or a shared multi-tenant system. Dedicated MongoDB Atlas Clusters are deployed in a VPC (for AWS and GCP) or VNet (for Azure) that fully isolates your Customer Data and is configured to prevent inbound network access from the internet. Each such MongoDB Atlas VPC or VNet utilizes security groups that act as a virtual firewall for your dedicated MongoDB Atlas Clusters.

3.3.2. Atlas IP Access List. In order to allow inbound network access to your MongoDB Atlas VPC or VNet, you must configure an Atlas IP Access List to enable specific networks to connect to the MongoDB Atlas Clusters within your MongoDB Atlas Project. Unless the Atlas IP Access List for a MongoDB Atlas Project includes a specific network's IP addresses, network traffic is prevented from accessing your MongoDB Atlas Clusters in that MongoDB Atlas Project.

3.3.3. Virtual Private Cloud Peering. You may enable peering between your MongoDB Atlas VPC or VNet to your own dedicated application tier virtual private network with the Cloud Provider of your choice (VPC or VNet). Peering permits you to route encrypted traffic between your MongoDB Atlas VPC or VNet and your own application tier VPC or VNet privately, rather than traversing the public internet. Subject to the capabilities of your selected Cloud Provider, you may also choose to peer your MongoDB Atlas VPC or VNet to your application tier VPC or VNet across regions.

3.3.4. Private Endpoints. MongoDB Atlas also supports private endpoints on AWS using the AWS PrivateLink feature and on Azure using the Azure Private Link feature. If you enable this feature for any MongoDB Atlas Cluster, that MongoDB Atlas Cluster will only allow a one-way connection from your AWS VPC or Azure VNet to the MongoDB Atlas Cluster and that MongoDB Atlas Cluster cannot initiate connections back to your AWS VPC or Azure VNet. Private endpoints also enable you to reach your MongoDB Atlas Cluster transitively over the network from other application tier AWS VPCs and Azure VNets that you have peered with the private endpoint, or through your own self-managed virtual private network including via AWS DirectConnect and Azure ExpressRoute.

3.4. Configuration Management. The MongoDB Atlas environment, including our production environment and your MongoDB Atlas Clusters, leverages configuration management systems to fully automate configuration based on one-time decisions that are securely applied to new and existing environments to ensure consistency every time. Our production environment and your MongoDB Atlas Clusters use in-house built machine images with secure configuration management applied via industry standard automation software, which includes hardening steps.

4. Access Controls.

4.1. Customer Access. MongoDB Atlas supports multiple authentication and authorization options and methods to give you the flexibility to meet your individualized requirements and needs. You are responsible for

understanding the security configuration options available to you and the impact of your selected configurations on your MongoDB Atlas environment, which consists of a web application administrative interface (“**MongoDB Atlas UI**”) and any MongoDB Atlas Cluster you deploy. MongoDB Atlas provides you with configurable authentication and authorization options for both the MongoDB Atlas UI and your MongoDB Atlas Clusters.

4.1.1. MongoDB Atlas UI Authentication and Authorization. User credentials for the MongoDB Atlas UI are stored using industry standard and audited one-way hashes. The MongoDB Atlas UI supports multi-factor authentication (MFA), including a security key/ biometrics option that enables you to use hardware security keys or built-in authenticators. The MongoDB Atlas UI also supports federated authentication functionality for Single Sign-On (SSO) utilizing Security Assertion Markup Language (SAML).

4.1.2. MongoDB Atlas Cluster Authentication and Authorization. Authentication control for a MongoDB Atlas Cluster is enabled by default with the Salted Challenge Response Authentication Mechanism (SCRAM). You may choose to manage user authentication with self-managed X.509 certificates or through AWS IAM Users or Roles. MongoDB Atlas allows you to define permissions for individual users or applications in order to restrict the Customer Data that is accessible in a query. Further, you may choose to assign each user a MongoDB Atlas Project-specific role, which authorizes that user to perform specific actions on the MongoDB Atlas Clusters within that MongoDB Atlas Project. The MongoDB Atlas UI allows you to tailor your access controls by combining multiple roles and privileges for particular users. You can review, limit, and revoke user access to your MongoDB Atlas Clusters at any time. MongoDB Atlas also provides you with the ability to manage user authentication and authorization using your own Lightweight Directory Access Protocol (LDAP) server over TLS. A single LDAP over TLS (LDAPS) configuration applies to all MongoDB Atlas Clusters in a MongoDB Atlas Project.

4.1.3. Credential Requirements. As part of the configuration options, you may establish minimum password requirements (e.g., length, complexity) through your identity provider after federating authentication to the MongoDB Atlas UI via SAML and to the MongoDB Atlas Clusters via LDAPS.

4.1.4. Customer Database Auditing. MongoDB Atlas offers granular auditing that monitors actions in your MongoDB Atlas environment and is designed to prevent and detect any unauthorized access to Customer Data, including create, read, update, and delete (CRUD)

operations, encryption key management, and role-based access controls. You are responsible for enabling database auditing and selecting the users, roles, groups, and event actions that you want to audit.

4.2. MongoDB Personnel Access to MongoDB Atlas Clusters.

4.2.1. Privileged User Access. As a general matter, MongoDB personnel do not have authorization to access your MongoDB Atlas Clusters. Only a small group of Privileged Users are authorized to access your MongoDB Atlas Clusters in rare cases where required to investigate and restore critical services. MongoDB adheres to the principle of “least privilege” with respect to those Privileged Users, and any access is limited to the minimum time and extent necessary to repair the critical issue. Privileged Users may only access your MongoDB Atlas Clusters via a gated process that uses a bastion host, requires MFA both to log in to our MongoDB Systems and to establish a Secure Shell connection (SSH) via the bastion host, and requires approval by MongoDB senior management.

4.2.2. Restricting MongoDB Personnel Access. MongoDB Atlas provides you with the option to entirely restrict access by all MongoDB personnel, including Privileged Users, to your MongoDB Atlas Clusters. If you choose to restrict such access and MongoDB determines that access is necessary to resolve a particular support issue, MongoDB must first request your permission and you may then decide whether to temporarily restore Privileged User access for up to 24 hours. You can revoke the temporary 24-hour access grant at any time. Enabling this restriction may result in increased time for the response and resolution of support issues and, as a result, may negatively impact the availability of your MongoDB Atlas Clusters. If you enable client-side field level encryption, even Privileged Users will be unable to access Customer Data within your MongoDB Atlas Clusters in the clear unless you provide MongoDB with the encryption keys.

4.2.3. Credential Requirements. Privileged User accounts may only be used for privileged activities, and Privileged Users must use a separate account to perform non-privileged activities. Privileged User accounts may not use shared credentials. The password requirements described in Section 4.3.3 also apply to Privileged User accounts.

4.2.4. Access Review and Auditing. MongoDB reviews Privileged User access authorization on a quarterly basis. Additionally, we revoke a Privileged User’s access when it is no longer needed, including within 24 hours of that Privileged User changing roles or leaving the

company. We also log any access by MongoDB personnel to your MongoDB Atlas Clusters. Audit logs are retained for at least six years, and include a timestamp, actor, action, and output. MongoDB utilizes a combination of automated and human review to scan those audit logs.

4.3. MongoDB Personnel Access to MongoDB Systems.

4.3.1. General. MongoDB's policies and procedures regarding access to MongoDB Systems adhere to the principles of role-based access control (RBAC), least privilege, and separation of duties. In accordance with these principles, with respect to MongoDB Atlas, MongoDB developers are only granted access to our development environments, and access to our production environment is limited to Privileged Users with appropriate authorizations. We review access authorizations to MongoDB Systems on a quarterly basis and we review any changes to authorizations for Privileged Users immediately. As part of the employee off-boarding process, access to MongoDB Systems is revoked within 24 hours of an employee's departure.

4.3.2. Access to MongoDB Atlas Production Environment. Our backend production environment that runs MongoDB Atlas is only accessible by a dedicated group of Privileged Users whose privileges must be approved by senior management. Privileged Users may only access our backend production environment via a bastion host and doing so requires MFA both to log in and to establish a SSH via the bastion host.

4.3.3. Credential Requirements. All MongoDB personnel passwords must conform to industry-standard complexity rules. Additionally, MFA is mandatory for all MongoDB personnel and cannot be disabled.

4.4. Physical Controls at MongoDB Offices. As noted in Section 3.1, Customer Data is deployed at the data centers of your selected Cloud Provider, and not at facilities owned or operated by MongoDB. At MongoDB offices, we follow industry best practices to employ physical security controls that are appropriate to the level of risk posed by the information stored and the nature of operations at our offices. In our offices, we: (i) issue access cards for all personnel through formal provisioning and approval processes; (ii) limit access to restricted areas to personnel with a need to access those areas to carry out their job functions; (iii) require visitors to sign in, execute a non-disclosure agreement, and be escorted in all non-public spaces; (iv) employ surveillance systems to monitor activity at points of entry from public spaces; and (v) revoke personnel access within 12 hours of termination.

4.5. Secure Deletion of Customer Data. If you terminate a MongoDB Atlas Cluster, it will become unavailable to you immediately and any Cloud Backup associated with that MongoDB Atlas Cluster will be terminated. MongoDB may retain a copy of the Customer Data stored in the terminated MongoDB Atlas Cluster for up to 5 days. If you terminate Cloud Backups, all snapshots will become unavailable to you immediately and it may take up to 24 hours for the Customer Data contained in the snapshots to become unrecoverable. When you terminate a MongoDB Atlas Project, the master key used to encrypt Customer Data is securely wiped, rendering all Customer Data effectively unrecoverable. If you choose to use MongoDB Atlas Online Archive, you can delete the entire archive, or pre-define automatic deletion dates for different data sections within MongoDB Atlas Online Archive to help automate any applicable retention restrictions or policies.

5. MongoDB Systems Security.

5.1. Separation of Production and Non-Production Environments. MongoDB Atlas has strict separation between production and non-production environments. Our MongoDB Atlas production environment, your MongoDB Atlas Clusters, and your Customer Data are never utilized for non-production purposes. Our non-production environments are utilized for development, testing, and staging. MongoDB also maintains firewalls to achieve strict separation of our MongoDB Atlas production environment and MongoDB's internal network.

5.2. Software Development Lifecycle. MongoDB has a dedicated security team, reporting to the Chief Information Security Officer, that leads security initiatives in the software development lifecycle (SDLC). We develop new products and features in a multistage process using industry standard methodologies that include defined security acceptance criteria and align with NIST and OWASP guidance. The SDLC includes regular code reviews, documented policies and procedures for tracking and managing all changes to our code, continuous integration of source code commits, code versioning, static and dynamic code analysis, vulnerability management, threat modeling, and bug hunts, as well as automated and manual source code analysis.

5.3. Monitoring and Alerting. MongoDB monitors the health and performance of MongoDB Atlas without needing to access your MongoDB Atlas Clusters. MongoDB maintains a centralized log management system for the collection, storage, and analysis of log data for our MongoDB Atlas production environment and your MongoDB Atlas Clusters. We use this information for health monitoring, troubleshooting, and security purposes, including intrusion detection. We maintain our log data for at least six years, and we utilize a combination of automated scanning, automated

alerting, and human review to monitor the data.

5.4. Vulnerability Management.

5.4.1. MongoDB Atlas Vulnerability Scanning. MongoDB maintains a documented vulnerability enumeration and management program that identifies internet-accessible company assets, scans for known vulnerabilities, evaluates risk, and tracks issue remediation. We conduct quarterly scans of both the underlying systems upon which MongoDB Atlas is deployed, as well as all third-party code integrated into our products. MongoDB's vulnerability management policy requires individual engineering teams to identify known vulnerabilities in system components, and develop remediation timeframes commensurate to the severity of an identified issue. We also utilize automated tooling in conjunction with monitoring security bulletins for relevant software and libraries, and implement patches if security issues are discovered.

5.4.2. Vulnerability Remediation. MongoDB uses a central company-wide ticketing system to track all security issues until remediation. We implement patches to our operating system and applications on a need-to-update basis, as determined in accordance with the Common Vulnerability Scoring System (CVSS). We are also a Mitre CVE Numbering Authority (CNA). Development tasks for all patches, bug fixes, and new features are defined as issues for specific target releases and are deployed to production only after completing requisite checkpoints, including quality assurance testing, staged deployment, and management review.

5.5. Penetration Testing and Internal Risk Assessments. MongoDB Atlas undergoes regular reviews from both internal and external security teams.

5.5.1. External Testing. Our MongoDB Atlas production environment is subject to an external penetration test by a nationally recognized security firm at least once per calendar year. Upon request, we will provide you with a summary letter of engagement that includes the number of high, medium, and low issues identified, but due to the sensitivity of the information gathered during these tests, we cannot allow customers to perform testing of our production platform. Application-level security testing uses a standard application assessment methodology (e.g., OWASP). Additionally, external engagements with security consultants may include social engineering and phishing testing.

5.5.2. Internal Testing. Internally, MongoDB Atlas undergoes periodic

risk assessments, including technical vulnerability discovery and analysis of business risks and concerns. The MongoDB security team is also routinely involved in source code review, architecture review, code commit peer review, and threat modeling.

6. Contingency Planning.

6.1. High Availability and Failover. Every MongoDB Atlas Cluster is deployed as a self-healing replica set that provides automatic failover in the event of a failure. Replica set members are automatically provisioned by MongoDB Atlas across multiple availability zones within a region, providing resilience to localized site failures. All replica set members are full data-bearing nodes, ensuring majority writes in the event of single node failure and higher resilience during recovery. Concurrent writes across replica sets occur in real time. MongoDB Atlas also offers multi-region and multi-cloud deployment options.

6.2. Backups. MongoDB Atlas offers Cloud Backups, which use the native snapshot functionality of your selected Cloud Provider to locally back up your Customer Data. You may enable Cloud Backups when you create or modify a MongoDB Atlas Cluster, and you have control over how often a Cloud Backup is captured and the length of time for which Cloud Backups are retained. Cloud Backup snapshots are stored with your selected Cloud Provider in the primary region of your MongoDB Atlas Cluster. All Cloud Backups are encrypted at rest and you may choose to use self-managed keys with the WiredTiger Encrypted Storage Engine. You may also optionally enable Continuous Cloud Backups with point-in-time recovery stored on our encrypted S3 buckets.

6.3. Business Continuity and Disaster Recovery. MongoDB maintains a documented business continuity and disaster recovery (“BCDR”) plan that aligns with ISO/IEC 22301:2019. Our BCDR plan includes: (i) clearly defined roles and responsibilities; (ii) availability requirements for customer services, including recovery point objectives (RPOs) and recovery time objectives (RTOs); and (iii) backup and restoration procedures. We review, update, and test our BCDR plan at least annually. In the event of an incident that triggers the BCDR plan, the RPO will depend on your impacted MongoDB Atlas Cluster and backup configurations. You can test how your application handles a replica set failover at any time using the MongoDB Atlas UI or API.

7. Incident Response and Communications.

7.1. Security Incident Response Plan. As part of the Information Security Program, MongoDB maintains an established Security Incident Response

Plan that aligns with NIST and ISO/IEC 27001:2013. In the event that MongoDB becomes aware of a Data Breach or other security incident, MongoDB will follow the Security Incident Response Plan, which includes: (i) clearly defined roles and responsibilities, including designation of a security incident task force; (ii) reporting mechanisms; (iii) procedures for assessing, classifying, containing, eradicating, and recovering from security incidents; (iv) procedures and timeframes for required notifications to relevant authorities and customers; (v) procedures for forensic investigation and preservation of event and system log data; and (vi) a process for post-incident and resolution analysis designed to prevent future similar incidents. The Security Incident Response Plan is reviewed, updated, and tested annually, including a security tabletop exercise at least once per year.

7.2. Security Incident Tracking. MongoDB maintains a comprehensive security incident tracking system that aligns with ISO/IEC 27001:2013 and documents: (i) incident type and suspected cause; (ii) whether there has been unauthorized or unlawful access, disclosure, loss, alteration, or destruction of data; (iii) if so, the categories of data affected by the incident, including categories of personal information; (iv) the time when the incident occurred or is suspected to have occurred; and (v) the remediation actions taken.

7.3. Customer Communications. MongoDB will notify you without undue delay if we become aware of any Data Breach. Taking into account the information available to us, such notice will include a description of the nature and cause of the Data Breach and the expected resolution time. To the extent possible, we will subsequently update you with information regarding evaluation of the root cause, potential impact, remediation actions taken, and actions planned to prevent a future similar event.

8. Audit Reporting.

8.1. Third-Party Certifications and Audit Reports. Upon request, and subject to the confidentiality obligations set forth in the Agreement, we will make available to you (or your independent, third-party auditor) information regarding MongoDB's compliance with the security obligations set forth in these Security Measures in the form of third-party certifications and audit reports.

8.2. Security Questionnaires. No more than once per year, we will complete a written security questionnaire provided by you regarding the controls outlined in these Security Measures.