SingleStore Helios®

Cloud Security White Paper

Table of Contents

1. Security at SingleStore

At SingleStore, data security is a top priority and key focus for our SingleStore Helios data platform. Backed by cybersecurity professionals with decades of experience and industry-leading certifications in cloud security, security is embedded into everything we do — from software development lifecycle processes to operations of SingleStore Helios on AWS, Azure, and GCP.

Our secure by design approach lays a foundation of security providing a defence-in-depth (DiD) posture for the SingleStore Helios data platform. We complement these practices with cloud-native security CNAPP platforms to continuously assess our cloud accounts and workloads for vulnerabilities, misconfigurations, and policy violations. Through continuous evaluation of the security properties of confidentiality, integrity, and availability, paired with validation of authentication, authorization and auditing controls, we maintain high levels of security assurance for SingleStore products and services.

2. Scope

The scope of this document is limited to the SingleStore Helios cloud offering. SingleStore Self-managed and other self-hosted deployments are excluded and are not covered in this paper.

3. Trust Center and Compliance Posture

SingleStore maintains a formal, independently-audited security and privacy program, with all controls and policies detailed in the SingleStore Security & Trust Center (https://singlestore.trust.site). Its cloud offerings have achieved key third-party certifications including **ISO/IEC 27001, SOC 2 Type 2**, and **HIPAA Type 1**, and support compliance with global privacy regulations such as **GDPR** and **CCPA/CPRA**. SingleStore Helios supports regulated workloads, including HIPAA-aligned use cases. For **workloads involving PCI DSS requirements, a Shared Responsibility Model** is in place: SingleStore secures the platform and infrastructure, while customers are responsible for application design, cardholder data handling, and configuration. Ongoing security assurance is provided through annual audits, regular risk assessments, formal Secure SDLC practices, independent penetration testing, and periodic disaster recovery exercises.

4. Shared Security Responsibility

SingleStore Helios has built in security controls that make it a secure environment to run customer workloads. The responsibility of keeping data secure is shared between the user and SingleStore. SingleStore Helios is designed with strong security by default so that there is minimal overhead on the user. The default configuration includes encryption at rest, encryption in transit, removal of public access, and deployment within strong network boundaries. Users are responsible for configuring the necessary levels of control which is based on the security posture of their organization. Please refer to the online documentation for more details.

5. Platform Architecture Overview

The SingleStore Helios data platform is designed to allow organizations power data & AI applications at scale by combining the world's first Distributed SQL cloud database with a suite of cloud services to autonomously handle deployment, operation, and observability at scale. SingleStore Helios is designed and architected to handle and recover from failures automatically. To maintain each cluster's availability, SingleStore Helios runs on a highly resilient software cloud infrastructure with built-in high availability (HA).

The SingleStore Helios service is composed of two distinct components; the **Control Plane** and the **Data Plane**. The Control Plane is an online portal that provides a user with a unified view of all database deployments in one place; it provides insights into performance metrics, resource utilization, cluster health, security, and access control. The Control Plane is also responsible for the orchestration and deployment of the customer's cluster resources. The Data Plane refers to the compute and the storage resources within the service that includes the database clusters. It can be deployed in all three major cloud providers and in any combination of supported regions simultaneously.

5.1. Control Plane

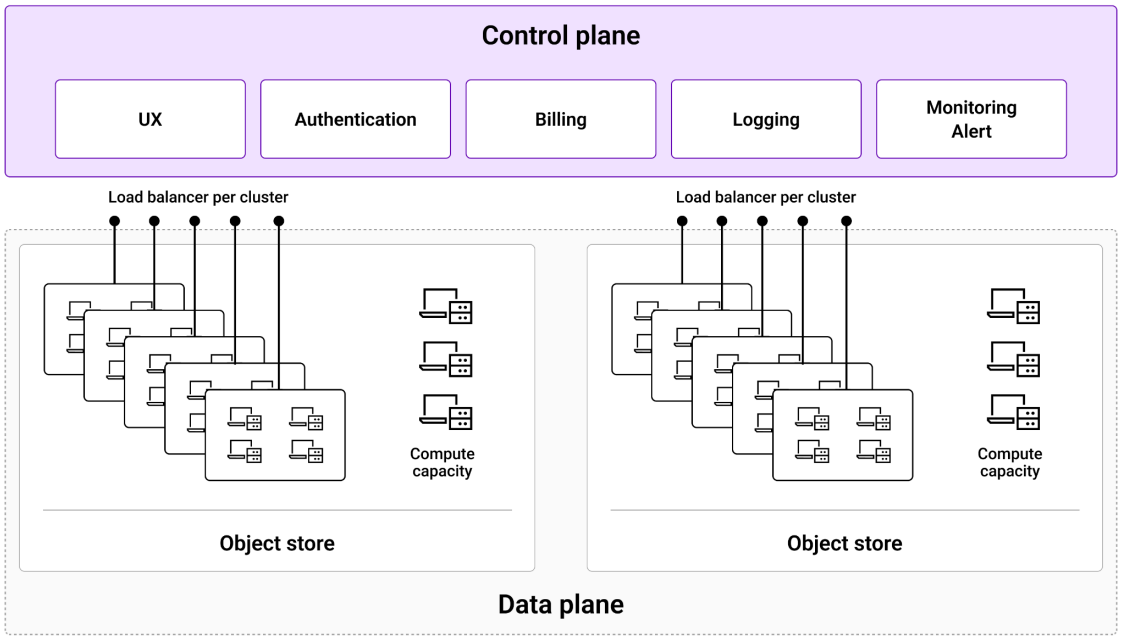
The Control Plane for SingleStore Helios consists of various services which talk to each other via HTTPS APIs. All internal connectivity uses TLS 1.2 and above. These services interact with the Data Plane via the Kubernetes REST API.

The primary service exposes an HTTPS API to the administrative portal (frontend). The customer-facing portal runs in a web browser (portal.singlestore.com) and the static assets are stored in object storage and served worldwide via Cloudfront.

The HTTPS API used between the Control Plane and the Data Plane is secured using signed JSON Web Tokens (JWT tokens) with HTTP Authentication. These JWT tokens are generated by our identity provider and identify a single user of SingleStore Helios. The mapping between users and their respective organizations is managed by the Control Plane.

The Control Plane has a data store which is used to store organizations, metadata about all clusters, billing data, quotas, etc.

Figure 1. SingleStore Helios Control Plane / Data Plane Architecture



5.2. Data Plane

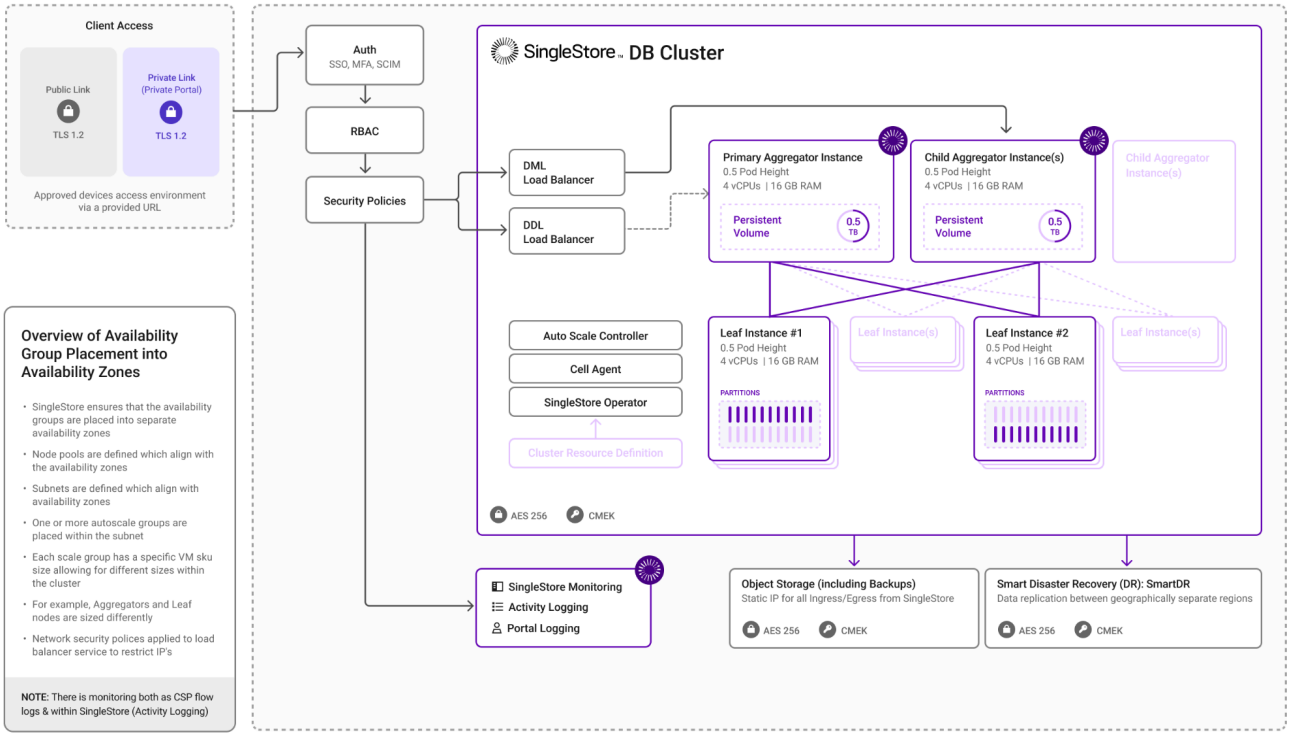
The Data Plane is implemented as one or more containerized deployments per region. Each containerized deployment manages the compute, storage, and load balancers used to host customer SingleStore Helios clusters. Compute (vCPU, Memory) is isolated for each customer using a container running on the Host VM along with individually provisioned block storage for caching. All clusters use dedicated object store buckets. Therefore, resources are isolated by customer deployment. Each data plane resides in a discrete account in the cloud provider region. Access to these accounts is secured with the provider's native identity and access management controls together with SingleStore's internal IDP groups.

SingleStore Helios leverages the power of containerization to automate functionality and provide a seamless user experience. Container architecture is used for the Data Plane shown in Figure 1. This is coupled with the Control Plane for managing resources, and the Data Plane to provide capacity such as vCPU, memory, network, and storage so that the containers can run and connect to a network.

The Control Plane communicates with the Data Plane using a finely-scoped, well-defined API. This connection is secured by TLS1.2, secure certificate and access key authentication, over a private non-routable network. A SingleStore resource is created in the data plane, which is managed by the Control Plane.

If a customer wishes to terminate their cluster, a request is issued through the administrative portal (control plane). The Control Plane will issue an API request to the Data Plane to deallocate the cluster. At this point, the cluster is shut down and usage stops accruing usage. When a workspace is terminated, its compute resources (including leaf nodes) are deallocated and any cached data on those nodes is automatically and securely purged. The underlying encrypted database data and backups remain in object storage for the configured retention period (7 days by default), which allows point-in-time recovery in case of accidental deletion. After this retention window expires, the data in object storage is automatically and securely deleted.

Figure 2. SingleStore Helios Data Plane Architecture



5.2.1. Data Plane Architecture for Region Types

SingleStore provides flexible deployment options across multiple regions to meet your infrastructure, security, and compliance needs. "Regions" refer to geographical locations and infrastructure zones provided by AWS, Google Cloud (GCP), and Microsoft Azure where SingleStore can be deployed. SingleStore Helios can be deployed in the following region types: Managed, Dedicated.

Managed regions support all security features, are fully managed, and are a feature-rich way to deploy SingleStore. These regions deliver the best overall TCO (Total Cost of Ownership) and provide full access to all the SingleStore capabilities. Managed regions are available across AWS, Azure, and GCP and support Shared, Standard, and Enterprise editions of SingleStore.

Dedicated regions support the same feature set and functionality as Managed regions, but they are specifically designed for organizations that require a single-tenant data plane. Dedicated regions offer the same level of security as Managed regions and maintain the same Helios availability SLAs and compliance certifications as Managed Regions, but consume compute credits at a faster rate.

5.2.2. SingleStore Product Editions

Shared, Standard, Enterprise

SingleStore provides several editions of SingleStore for Clusters and Databases to meet the unique needs of a variety of customer workloads. These include Shared, Standard, and Enterprise, each providing specific features to meet customer workloads and needs. Each edition shares the Control Plane infrastructure, but may deploy and manage Data Plane distinctly to meet operational, compliance or data governance goals.

Shared edition clusters are deployed with shared vCPU, Memory, and Storage distributed across customers. Resources are allocated using resource governance, and strong user-access, management, and encryption protocols are used to ensure data is isolated and protected from unauthorised access.

Standard edition clusters and databases are designed for general-purpose production workloads that need high performance, scalability, and access to the full set of core SingleStore features across multiple clouds and regions. They provide the complete relational and multi-model storage engine (including MySQL compatibility, JSON, time-series, full-text search, vector, and geospatial), combined with high availability and load-balanced failover, multi-AZ resilience, and continuous backups.

Enterprise edition clusters & databases provide additional controls including advanced security features such as Audit logging, Smart DR, and Customer Managed Encryption Keys (CMEK). The keys used to encrypt the data are backed by KMS and the Encryption Keys are encrypted and stored within the deployment. This allows customers to integrate SingleStore Helios with their cloud KMS giving full control over data access. If CMEK is revoked from SingleStore Helios, all data stored using CMEK is made unavailable. Significant care should be used with this feature, as SingleStore has no way to recover this data in the event that keys are destroyed.

5.3. Data Centers

SingleStore Helios runs on the cloud infrastructure of the three largest public cloud providers: Amazon Web Services (AWS), Google Cloud (GCP), and Microsoft Azure (Azure). Customer data is stored in SingleStore Helios clusters & databases. Within an organization, projects can be created using **Shared, Standard, or Enterprise** editions, and a single customer can run multiple editions simultaneously across different projects.

Public cloud providers' data centers follow strict compliance guidelines that SingleStore inherits as part of the service orchestration along with the key security capabilities that SingleStore adds on top. Further information can be found on the compliance page of your selected cloud provider:

- AWS compliance
- Azure compliance
- Google Cloud compliance

5.4. SingleStore Resiliency & Disaster Recovery

Multi-AZ high availability

SingleStore Helios deploys both the **Control Plane** and **Data Plane** across multiple **Availability Zones** in each region so that node or AZ failures do not interrupt service. Clusters continue to run on healthy nodes in other AZs. Customers can choose to deploy clusters in a single AZ or across multiple AZs depending on workload uptime requirements.

Multi-region data protection with Smart DR

For protection against **regional** outages, SingleStore Helios provides **Smart Disaster Recovery (Smart DR):**

- Smart DR maintains **continuous asynchronous replication** of your databases between **two geographically separate regions.**
- It preserves your **cluster topology, users, permissions, and other metadata** across regions.

- Failover and failback are **fully automated**, and Smart DR can be configured in two modes, in “cold” mode without a failover cluster for the lowest TCO, or with a “hot” failover cluster for minimal failover time.
- On failover in “cold” mode, Helios provisions compute in the failover region, attaches the replicated databases, and exposes a **new connection string** for your applications with **RPO as low as ~10 minutes** and no need to run “hot” compute in the DR region.
- In “hot” mode users can optionally **pre-provision** compute in the DR region to shorten failover time further.

As part of the disaster recovery process, Helios performs **automatic continuous backups** to durable object storage and supports **Restore** (Standard) or **Point-in-Time Recovery** (Enterprise) to restore databases to any time within the retention window (typically 7 days).

Control Plane disaster recovery

The Helios **Control Plane** (portal and management APIs) does **not store customer data**, but is important for operations (for example, autoscaling, cluster management). To ensure its availability:

- The Control Plane backing database is **replicated asynchronously across regions**, and a failover typically completes in about **3 minutes**, which is the primary contributor to portal downtime during a DR event.
- Control Plane compute runs in **two clusters in different regions**, with a warm secondary site that can be promoted quickly during failover.
- DR is executed through a **standard, audited runbook**, and our target **RTO for the Control Plane is 30 minutes**, with **RPO of 60 minutes** for configuration metadata; neither affects the durability of customer data protected by Smart DR and continuous backups.

This combination of **Multi-AZ HA, multi-region Smart DR**, and **Control Plane DR** provides layered protection against node, AZ, and regional failures, while keeping ongoing DR costs low.

6. Connectivity and Network Security

At SingleStore we empower our customers to help secure their data from unauthorised users. We use a layered approach to security, starting with IP allowlisting to ensure only devices you trust, and have given access to, can access your cluster or your data. We then ensure that the data passing between your trusted devices and SingleStore Helios is encrypted with TLS 1.2 and above to protect it from being intercepted during transit.

Access to the service can be achieved either directly via the internet, through private connectivity. When using VPC endpoints such as AWS PrivateLink, Azure Private Link or GCP Private Service Connect, connectivity is bound between the endpoint in the service and your environment, both of which must reside within the same region. SingleStore Helios private connectivity ensures data does not leave a customer's environment and is not exposed to the wider internet.

Support and development teams are provided with read-only access to telemetry and observability data. Generally, such data does not contain any personal identifiable information (PII) data or query text from the customers. Administrative access to the Control Plane is limited to the SingleStore Site Reliability Engineering (SRE) team only. A list of administrators is maintained and regularly reviewed. Access requires connectivity to a certificate-based VPN as well as additional connection to a bastion host. Administrative actions require additional short-lived authentication and authorization to connect to API endpoints. Additionally, SingleStore also supports multi-person approval workflow for just in-time requests for access to customers' environments. These requests

require both internal and customer approvals before access to the infrastructure is granted, and access to these services is logged for audit purposes.

The SingleStore SRE team can update security groups, and such actions are logged in the portal and in logging systems. Our SRE team can also access leaf nodes which are logged in the audit trail. This is granted based on least privilege. The SREs are trained on their responsibilities annually and access is reviewed quarterly and removed when no longer appropriate/required or when an employee leaves SingleStore.

6.1. Cloud Native Connectivity

SingleStore Helios supports cloud native connectivity by exposing a cluster endpoint to which clients and applications can connect. A SingleStore cluster endpoint looks like this:

```
[mysql -u admin -h svc-ee432f9e-37a2-4236-a8fa-05936b6e292d-dml.aws-ireland-1.
svc.singlestore.com -P 3306 --default-auth=mysql_native_password -p]
```

Endpoints are accessible only from IP addresses which have been allowlisted in SingleStore's firewall rules, and connections are authenticated and secured using TLS/SSL. Complete details on connecting to your cluster from a command line or from client tools can be found here.

In addition to securing and encrypting all connections to the SingleStore Helios cluster, SingleStore Helios also supports private network routing across all major platforms, including AWS PrivateLink, Azure Private Link and Google Private Service Connect. Please contact SingleStore for more information on connecting to your deployment from AWS, GCP or Azure.

6.2. Private link to the management API

While cloud connectivity is secured using the private link and firewall rules, SingleStore Helios also supports the same network security for the management API used for building API automation. This includes IP level filtering as well as using secure private link connectivity to the API endpoint. This ensures the end to end network security from customers VPC to SingleStore Helios.

6.3. IP Allowlisting

There are multiple ways that users can ensure only a select number of IPs can communicate with their SingleStore Helios cluster. First and foremost, users have the option to add a new IP or their current IP during the cluster creation process. Users may also open access to any IP, but this is not the recommended approach. If a user chooses not to make IP selections during cluster creation or wants to modify the settings, they can do so after the fact. Please contact SingleStore support if you desire further information on configuring IP allowlisting.

6.4. Web Application Firewall (WAF)

To offer a secure, robust, and scalable data management solution, SingleStore Helios uses Web Application Firewalls (WAFs) in front of its internet-facing APIs as part of a layered defense-in-depth strategy. A WAF helps protect web applications by filtering and monitoring HTTP traffic between clients and the service, adding protection on top of existing network, authentication, and access-control mechanisms.

SingleStore Helios uses an NGINX-based WAF with industry-standard rules to detect and block common web attacks such as SQL injection (SQLi) and cross-site scripting (XSS), and can be extended with additional rules as

needed over time. This helps safeguard Helios APIs and customer workloads from a broad class of application-layer threats while preserving performance and reliability.

6.5. Configuring Client Connections

SingleStore Helios is MySQL wire-compatible and exposes secure endpoints that can be used by standard MySQL-compatible tools and drivers, as well as SingleStore-native clients.

Customers can connect using a broad range of SQL clients and IDEs, including tools such as DBeaver, DbVisualizer, JetBrains DataGrip, MySQL clients, SQL Developer, SQL Workbench, Sequel Pro, the SingleStore SQL Editor, and Visual Studio Code.

SingleStore also provides native JDBC and ODBC drivers, language-specific libraries (for example C/C++, .NET, Go, Java, Node.js, PHP, Python, Ruby, Rust), and data access APIs such as the SingleStore Data API and integrations with platforms like Hasura.

Beyond direct client connectivity, SingleStore Helios integrates with a wide ecosystem of data integration, analytics, and application platforms. This includes ETL and data integration tools (such as AWS Glue, Apache Beam, Apache Flink, Apache NiFi, Azure Data Factory, Azure Event Hubs, Data Virtuality Pipes, dbt, Estuary Flow, Fivetran, HVR, Informatica PowerCenter, Liquibase, Qlik Replicate, the SingleStore Kafka Sink Connector, Spark, StreamSets, and Tableau Prep), analytics and BI platforms (including Power BI, Tableau, Looker, ThoughtSpot, and others), AI/ML platforms (such as Dataloop), and application platforms such as Twilio Segment and Vercel.

The most up-to-date list of supported clients, drivers, and integrations, along with step-by-step configuration guides, is available in the Connect to SingleStore and Integrate with SingleStore Helios documentation.

7. Identity and Access Management

7.1. Authentication

7.1.1. Portal Access Authentication

The SingleStore Helios Cloud Portal authenticates users to their SingleStore account using secure, short-lived, token-based sessions (established either via local login or SSO). Once issued, these short-lived tokens are used across the portal for subsequent requests.

Customer administrators control which users can access which organizations and workspace groups and can centrally manage users and groups either directly in Helios or via SCIM 2.0 user provisioning from supported IdPs (for example Okta and Microsoft Entra ID).

Helios supports:

- Local username/password accounts for portal sign-in, with baseline password complexity and security standards enforced by SingleStore.
- Multi-Factor Authentication (MFA) and federated SSO via SAML 2.0 and OIDC, allowing customers to enforce their own identity policies (MFA, conditional access, device posture, etc.) at their IdP.

Portal authentication credentials and management secrets are encrypted and managed by SingleStore within a secure credential store. Portal sessions are backed by short-lived tokens and cookies; authorization within the

portal is governed by role-based access control (RBAC) at the organization and cluster level. Customers can further harden access by combining IdP controls with Helios features such as IP allowlisting on database/cluster endpoints (see below).

7.1.2. Multifactor Authentication (MFA)

Multi-Factor Authentication (MFA) provides an additional layer of security by requiring a second form of verification beyond the primary authentication method. SingleStore Helios enforces MFA to protect against unauthorized access, even in the event of credential compromise.

Supported MFA Methods:

Email (Default): A one-time verification code is sent to the user's registered email address during login.

Authenticator Apps: Users can configure time-based one-time password (TOTP) authentication using compatible apps such as Google Authenticator or FreeOTP.

MFA Enforcement Policy:

Non-SSO users: Helios MFA is available and, by default, email-based MFA is enforced for all such users except for a predefined set of exemptions documented in the portal (for example, certain legacy Keycloak-MFA users).

SSO users: By default, users who log in via SAML or OIDC SSO are exempt from Helios-native MFA and rely on their IdP's MFA and conditional access policies.

Administrators can optionally enable an "**Enforce MFA**" toggle on the IdP connection so that SSO users must also satisfy Helios MFA in addition to their IdP MFA.

Users are prompted for MFA verification on each login by default. To reduce authentication friction, users may opt to trust a device for a configurable duration, up to a maximum of 30 days, during which MFA prompts are bypassed on that device. More details related to MFA can be found here.

7.1.3. SAML 2.0 Authentication (SSO)

SingleStore Helios supports SAML 2.0-based SSO with cloud-native identity providers such as Okta, Microsoft Entra ID, Ping, and others.

The portal acts as a SAML service provider:

- Users are redirected to the customer's IdP, authenticate there, and receive a signed SAML assertion.
- The assertion is exchanged for short-lived Helios session tokens using standard web-SSO flows that rely on one-time authorization codes, which are invalidated after use and mitigate token replay.
- Access to SSO endpoints can be further constrained by customer-side policies and, where required, by Helios IP allowlisting on administrative endpoints.

7.1.4. OpenID Connect (OIDC) (SSO)

SingleStore Helios also supports OIDC-based SSO, providing a modern, self-service integration path that is often simpler to configure and more flexible than SAML:

- Customers can integrate OIDC IdPs such as Okta, Microsoft Entra ID, JumpCloud using standard OIDC authorization-code flows.

- The IdP issues ID tokens and authorization codes, which the portal exchanges for Helios session tokens; security and step-up controls (MFA, conditional access) are inherited from the customer's IdP configuration.

These SAML and OIDC integrations allow customers to standardize on their own enterprise identity platform while delegating only the minimum required identity information to SingleStore.

7.1.5. Database Access Authentication

Database access to SingleStore Helios clusters support multiple authentication methods, allowing customers to combine traditional credentials with modern, passwordless flows.

7.1.6. Password-based database authentication

For users with database access, SingleStore supports local username/password authentication with a configurable password policy.

Database/cluster endpoints can be further protected by firewall / IP allowlisting, so only explicitly approved client networks can connect.

7.1.7. Mutual TLS (mTLS) for database clients

SingleStore Helios supports mutual TLS (mTLS) as a native authentication option for users with database access:

- Each cluster can be configured with a trusted Certificate Authority (CA) bundle via the Security tab in the portal; this bundle defines which client certificates are trusted.
- Clients present X.509 certificates during the TLS handshake; SingleStore validates that the certificate is cryptographically valid, chains to the trusted CA, and is not expired or revoked.
- Users can be configured with:
    - REQUIRE X509 – require a valid, CA-trusted client certificate for that user.
    - REQUIRE SUBJECT – optionally bind access to a specific certificate subject DN.

Connections that do not present acceptable certificates are rejected during the TLS handshake, before any password or JWT is evaluated, aligning with zero-trust and regulated-industry requirements.

mTLS is also supported in key integrations, such as Teleport-based RBAC access and Kafka pipelines / Kafka Connector connecting securely to SingleStore.

7.1.8. JWT-based database and API authentication

SingleStore Helios supports JSON Web Token (JWT)–based authentication for databases and the Management API:

- JWTs follow the RFC 7519 standard and provide a compact, signed representation of user or service identity that SingleStore verifies before granting access.
- JWTs may be issued by the Helios portal (for browser-based SSO into databases) or by customer-run identity providers and then presented directly to database clusters as a password alternative.

To simplify JWT-based and **passwordless** access:

- **For human users**, the open-source singlestore-auth-helper utility helps users complete a browser-based SSO flow and obtain ready-to-use, short-lived connection parameters for clients and tools, without storing raw passwords.
- **For servers and automation**, the singlestore-auth-iam library integrates with cloud IAM (AWS IAM, Azure AD, GCP IAM) so services can:
    - Obtain a short-lived identity token from the cloud provider.
    - Exchange it for a SingleStore-signed JWT.
    - Use that JWT to authenticate to databases or the Management API(upcoming) — without any static database passwords, and with automatic credential rotation and revocation.

Authorization remains governed by SingleStore roles and privileges; JWTs and IAM roles determine who you are, while SingleStore RBAC controls what you can do.

7.1.9. Connection links and stored credentials

Connection links provide a mechanism to store connection details (credentials and configuration) for external systems such as **S3, Azure Blob, GCS, HDFS, and Kafka**, and to reference them from commands like BACKUP, RESTORE, CREATE PIPELINE, and SELECT … INTO LINK without re-embedding secrets in every statement.

- Using a link is **more secure than putting credentials directly into SQL commands**, and access is controlled by granular permissions (CREATE LINK, SHOW LINK, DROP LINK). Only users with CREATE LINK need to know the underlying connection details; others can use links without seeing the credentials.

Together, these portal and database authentication mechanisms let customers:

- Use existing IdPs and IAM systems (SAML, OIDC, SCIM, cloud IAM).
- Enforce strong, centralized policies (MFA, password policies, certificate management).
- Gradually move both people and services toward passwordless, certificate-based, short-lived, auditable authentication for SingleStore Helios.

7.1.10. Password Policies

SingleStore Helios enables customers to enforce password expiration, reuse, and complexity requirements for users with **database** access. The following variables are available:

- **password_expiration_seconds** – Time in seconds before a password expires. Default 0 (no expiration).
- **expire_root_password** – Whether the root password is subject to expiration. Default OFF (root never expires unless this is set to ON).
- **password_expiration_mode** – Behavior when a password has expired:
    - **NO_ACCESS** (default): user cannot log in after expiration.
    - **LIMITED_ACCESS**: user can log in only to change their password (for example via ALTER USER or SET PASSWORD; other commands are blocked except SET SESSION / SET LOCAL).

- **password_history_count** – Number of previous passwords remembered and disallowed for reuse (includes the current password). Default 0 (any previous password allowed), maximum 10.
- **password_min_length** – Minimum password length (characters).
- **password_min_uppercase_chars** – Minimum number of uppercase letters required.
- **password_min_lowercase_chars** – Minimum number of lowercase letters required.
- **password_min_numeric_chars** – Minimum number of numeric digits required.
- **password_min_special_chars** – Minimum number of special (non-alphanumeric) characters required.
- **password_max_consec_sequential_chars** – Maximum allowed length of sequential character runs (for example, with value 3, 1234 or abcd is rejected).
- **password_max_consec_repeat_chars** – Maximum allowed length of repeated character runs (for example, with value 3, aaaa or 1111 is rejected).

For all password complexity variables (password_min_length, password_min_uppercase_chars, password_min_lowercase_chars, password_min_numeric_chars, password_min_special_chars, password_max_consec_sequential_chars, password_max_consec_repeat_chars), the **default is 0 (disabled) and accepted values are integers from 0 to 100.**

Additional behavior:

- **A warning is raised on every query starting 14 days** before a password expires.
- When password-complexity settings are changed, **existing passwords are not re-evaluated**; the policy applies to new or changed passwords only.

7.2. Access Control

7.2.1. Portal RBAC – SingleStore Helios

SingleStore Helios uses a role-based access control (RBAC) framework in the Cloud Portal to govern access to administrative features such as organizations, projects, clusters, databases, and billing.

Key points:

- RBAC is applied over a hierarchy of objects: organization → project → clusters/databases.
- Users are assigned roles, often via teams. Teams group users with similar functions so you can grant/revoke access by managing team membership rather than individual users.
- Actions in the portal are shown/hidden or enabled/disabled based on the roles a user has at each scope (organization vs cluster/database).
- Built-in Helios roles (for example, Reader/Writer/Operator) map to database roles and groups, so when a user is given a role assigned to a cluster, Helios automatically syncs a corresponding user/group and grants appropriate permissions for that database.

For full details and predefined Helios roles/teams, see Role-Based Access Control (RBAC) for SingleStore Helios.

7.2.2. Database RBAC

Database access is controlled by the RBAC model, using users, roles, groups, and privileges.

- You define users, roles, and groups, and use GRANT to assign privileges and connect them:
    - A role can have multiple privileges.
    - A group can have multiple roles and users.

A user can belong to multiple groups and can also have roles directly.

Users inherit all roles and privileges from the groups they are in.

- The docs provide a recommended set of standard roles as a starting point.
- Diagnostic cluster commands such as SHOW USERS, SHOW ROLES, and SHOW GROUPS help you audit which users have which roles and group memberships.

For full SQL examples and role definitions, see Role-Based Access Control (RBAC) at Database Level.

7.3. Row Level Security

For finer-grained control **within a table**, SingleStore supports **row-level security (RLS)** by combining RBAC with a special “roles list” column and a view.

Conceptually:

- RLS lets you restrict which **rows** a user can read based on their roles (for example, a salesperson only sees their own deals).
- Implementation uses:
    - A VARBINARY column (commonly called ACCESS_ROLES) that stores a **comma-delimited list of roles**, always with leading and trailing commas (for example, ,ROLE_A,ROLE_B,).
    - A **view** over the table that filters rows using SECURITY_LISTS_INTERSECT(CURRENT_SECURITY_ROLES(), ACCESS_ROLES). Only rows whose role list intersects the current user’s roles are returned.

Key configuration details:

- For new tables, create the roles column as:
  ACCESS_ROLES VARBINARY(<SIZE>) DEFAULT "," – the default **must** be a single comma for RLS to work correctly.
- For existing tables, add the column with:
  ALTER TABLE <table> ADD COLUMN ACCESS_ROLES VARBINARY(<SIZE>) DEFAULT ",";
- To **grant** row access, append role names (always ending with a comma) into ACCESS_ROLES for the relevant rows (for example, CONCAT(ACCESS_ROLES, "ROLE_X,")).
- To **revoke** row access, use REPLACE(ACCESS_ROLES, ',ROLE_X,', ',') so the role is removed cleanly from the list.

For a full deployment walkthrough, see the **Row-Level Security (RLS) Deployment Guide**.

7.4. Secure automation and cloud workload identity

Management API and Terraform: A documented **Management API** and official **Terraform integration** allow organizations to manage clusters, databases, users, and configurations as code, enabling controlled, auditable, and reviewable changes instead of ad-hoc manual operations.

Cloud workload identity and delegated entities (AWS): SingleStore Helios is designed to integrate with cloud provider's native workload identity and federated authentication mechanisms so that pipelines, applications, and other non-human workloads can access cloud resources and SingleStore without long-lived static credentials. Instead of embedding access keys or database passwords, services run under a cloud identity (for example, an IAM role (AWS), managed identity (Azure), or service account (GCP)), obtain short-lived tokens from the cloud provider, and use those tokens to authenticate to SingleStore or to assume narrowly scoped roles defined by the customer. This approach improves security by eliminating stored secrets, reducing credential-management overhead, and keeping authorization under the customer's existing identity and RBAC controls.

7.5. System of Cross-Domain Identity Management (SCIM)

SingleStore Helios implements SCIM 2.0 (RFC 7643/7644), allowing customers to synchronize users and groups – including user status – between their SAML 2.0 or OIDC identity provider (IdP) and SingleStore Helios.

Today, SingleStore provides documented and tested integrations for Okta and Microsoft Entra ID.

Because we follow the SCIM 2.0 standard, other IdPs that implement SCIM 2.0 may be supported as well. For IdPs other than Okta and Entra ID, please contact SingleStore Support to validate and enable SCIM configuration.

Benefits of using SCIM with SingleStore Helios:

- User management at a single point i.e., at your identity provider. When you add a user, they are added to SingleStore Helios, when you remove a user, they are removed from SingleStore Helios.
- Users should use SCIM in combination with SSO to completely simplify the user access management.
- Groups are automatically synchronized with SingleStore Teams once setup is complete. Hence customers can easily add or remove users from pre-defined teams with required RBAC controls.
- Setting up SCIM is easy, and users are managed at the organization level.

7.6. Privilege Access Management & JIT Access Provisioning

Privileged access by SingleStore personnel to customer Helios cloud accounts is managed using Okta Identity Governance and Access Requests. This just-in-time (JIT) model provides temporary, least-privilege elevated access for SingleStore employees via Okta Workflows when operationally required (e.g. for troubleshooting customer issues/incidents). JIT enforces an approval based workflow, and audits the duration, and business justification for privileged permission sets and accounts and guarantees access is auto-revoked after the approved duration. All requests are logged in Okta reports access logs.

8. Cryptography and Encryption

Encryption encapsulates the processes and controls used to ensure data remains inaccessible to unauthorized users and to protect data between the end user, client apps, and servers involved. In accordance with industry guidelines and best practice, SingleStore Helios applies both encryption to data in transit and data at rest.

8.1. Encryption at Rest

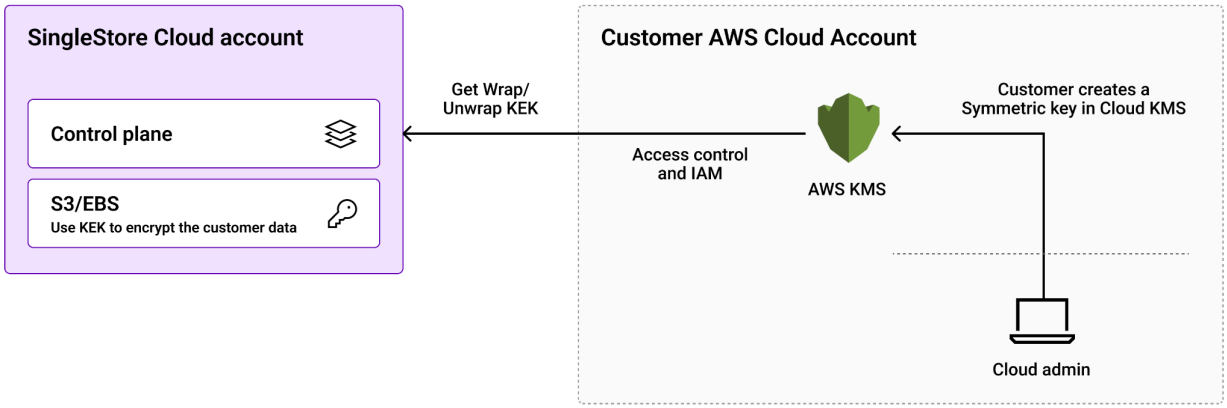
For data at rest, SingleStore Helios paired with hosting partners AWS, Azure, and Google Cloud, utilizes the best practice solution, AES-256, an encryption algorithm using a 256-bit key length. This is currently the strongest encryption algorithm available from and supported by our cloud hosting partners. In a standard configuration, the cloud-managed KMS key is used to encrypt persistent-storage (i.e., block devices & object storage) at rest.

8.1.1. Encryption at rest using Customer managed encryption key (CMEK)

Customers who require additional control of data at rest encryption may configure SingleStore Helios to use an encryption key that they manage through the cloud provider's key vault (i.e., AWS KMS, Azure Key Vault, GCP KMS). This feature is available as part of Enterprise edition. This allows customers to import their own encryption key and control access to it. If a customer chooses to revoke access to this key or delete the key, encrypted data will be unable to be decrypted by SingleStore or anyone else who does not have the key. This support helps customers meet their corporate policy requirements and align with Zero Trust principles.

Customer managed encryption keys control access to data through their configuration as Key Encrypting Keys (KEK). KEKs are used to encrypt the data keys that actually encrypt and decrypt data at rest. When data at rest is encrypted using this method, access to the KEK is needed to unwrap the data key(s) that will be used to decrypt the data at rest. If the KEK is not available the data keys can not be unwrapped and the data can not be decrypted. Details of the encryption method available here.

Refer to the layout of the customer managed encryption for data protection at rest.



Users should use the cloud provider's key management service to generate or import an encryption key that the user has control over. Users should grant permissions to SingleStore Helios to use this key to protect storage volumes and object storage buckets. If at any time the user wishes to prevent decryption of your data in the cloud, they may simply revoke access to the managed key or remove the key.

Note: SingleStore Helios will be unable to decrypt any data at rest that is protected by customer- managed key if access to the key is removed or if the key is deleted from the environment. Additionally, deleted customer

managed keys are your responsibility as the customer and cannot be recovered by SingleStore. Also, customers should follow their cloud provider's recommended best practices and controls for configuring and using customer-managed keys in its native key management service (for example, AWS KMS, Azure Key Vault, and Google Cloud KMS). The key you provide is used to protect all the data within the scope of that key.

8.2. Encryption of Data in Motion/Data in Transit

For data in transit - For all connections to the database SingleStore Helios supports TLS 1.2. Transport Layer Security (TLS) uses a combination of symmetric and asymmetric encryption focusing on the uses of key pairs, a public key and a private key.

To ensure a secure connection to SingleStore Helios, SQL clients must be properly configured to both require a secure connection and to verify the supplied server certificate. If you have REQUIRE SSL configured, then users will not be able to connect without SSL configured. Not having TLS configured can compromise security and lead to man-in-the-middle attacks, where a would-be attacker can impersonate a server when SSL is disabled or create a secure connection by impersonating a server using an illegitimate server certificate.

Additionally, for data movement within the clusters including the distributed join operations across leaf nodes, data transmission between the cluster and the object stores (AWS S3, Azure blob store or Google cloud storage) always uses TLS 1.2 for encryption of data in motion.

9. Logging and Monitoring

With data being today's currency it is important to know who can access it, who has viewed it, and why.

The internal telemetry feed is available to SingleStore site reliability engineers solely for operational and support purposes and does not include customer table data. And, only aggregated usage statistics are available to the SingleStore product team. Observability data stored in the customer tenant is limited to a maximum of 30-day retention unless otherwise specified.

Helios exposes rich cluster-level metrics and dashboards so customers can monitor health, query performance, and resource usage, and detect abnormal behavior proactively.

In Enterprise edition, SingleStore Helios provides **comprehensive logging and audit capabilities** across both the **Control Plane** (portal and management APIs) and the **Data Plane** (database activity). These audit logs are designed to support security auditing, incident investigation, and validation of access-control policies.

9.1.1. Control Plane Audit Logs

Control Plane audit logs record **user and system actions in the Cloud Portal,** including cluster lifecycle operations, firewall and RBAC changes, user and team management, billing and payment updates, authentication/SCIM events, API key generation, and more.

- Logs are accessible via the **/v1/auditLogs** endpoint of the Management API and are returned as **JSON records** containing fields such as organization ID, user identity, event type, source (Portal/Admin/SystemJob), timestamp, and descriptive reason.
- Event types are grouped into categories (Management, Cluster, Firewall, User/Teams, Security & Access, Billing & Payment, SCIM Integration, etc.), enabling fine-grained filtering and correlation.
- Control Plane audit logs are accessed via the Management API and can be integrated into existing security monitoring pipelines.

These logs allow administrators to **trace all sensitive configuration and access-control changes** in the Control Plane for compliance and forensic analysis.

9.1.2. Data Plane Audit Logs

Data Plane audit logs capture database-level activity, including login attempts, DDL and DML statements, system responses, and other engine operations.

- Audit logs are generated in the Data Plane and **forwarded to external destinations**, where they are parsed as JSON and can be consumed by customer tools.
- Helios supports forwarding to a range of third-party observability platforms, including Amazon CloudWatch, Amazon S3, Azure Blob, Azure Log Analytics, Datadog, Google Cloud Logging (via Stackdriver), New Relic, and Splunk; configuration is done in collaboration with SingleStore Support and requires standard destination parameters (for example, log group name, bucket, region, API key).
- Helios provides 11 audit levels, from LOGINS-ONLY (only successful/failed logins) up to ALL-RESULTS-INCLUDING-PARSE-FAILS (queries, full text, and results), grouped into:
    - Levels that log only valid statements/queries (LOGIN, ADMIN-ONLY, WRITES-ONLY, ALL-QUERIES, ALL-QUERIES-PLAINTEXT, ALL-RESULTS).
    - "INCLUDING-PARSE-FAILS" variants that also log invalid statements that could not be parsed.
    - By default, Helios uses ADMIN-ONLY-INCLUDING-PARSE-FAILS, which records admin-level operations and login attempts (including parse failures) but omits ordinary SELECTs unless they involve administrative actions.
    - Data Plane audit logs are **streamed to customer-controlled destinations**, where retention and access are governed by the customer's own log management platform and policies.

By design, **credentials and PII in valid statements are obfuscated** in audit logs; however, when a statement fails to parse, the literal query text is logged and may contain sensitive values, which customers should account for in their SIEM and log-retention policies.

These logging and monitoring capabilities provide **end-to-end visibility** over administrative actions and data access patterns in SingleStore Helios, enabling customers to meet typical audit, compliance, and incident-response requirements.

9.2. Helios SIEM

SingleStore uses a centralized SIEM platform to continuously ingest and analyse sensitive events from Helios control plane audit logs. This enables proactive security monitoring across the following domains on behalf of our customers:

- Suspicious authentication and authorization activity (RBAC events), including failed logins, privilege escalations, and role changes
- Critical resource modifications and system configuration changes
- Administrative and privileged user actions, including actions performed by SingleStore operators
- Infrastructure lifecycle events

- Detection of anomalous behaviour and deviations from established baselines
- Indicators of potential account compromise or insider threat activity
- Other security-relevant events and potential anomalies

Importantly, SingleStore does not monitor or inspect customer data contents. The monitoring is limited to control plane and security metadata (such as audit events, timestamps, user identifiers, and system actions) required for security operations, compliance, and incident response purposes.

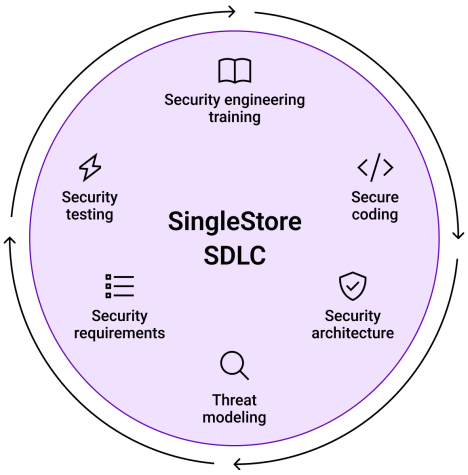
This approach provides real-time visibility into security events, security metrics and trend analysis, automated alerting, and rapid incident response capabilities.

10. Security Assurance in Software Development Lifecycle Practices

One of the goals of SingleStore engineering processes is to ensure a secure Software Development Lifecycle (SDLC) that simultaneously:

i) provides you the assurance that our products and services meet your security requirements and align with industry standards, guidelines and best practices; and

ii) empowers our engineering teams with security skills that ensure security is practiced and embedded within every phase of the software development lifecycle.



In order to achieve these goals, SingleStore strives to model its SDLC implementation after OWASP SAMM principles and implements the following practices for our product software:

- **Domain-specific security training for all of SingleStore engineers and software development professionals** (content including but not limited to aspects such as the OWASP Top 10, best practices on secure coding, threat modelling, and use of security tooling in the SDLC);
- **Threat modelling**, leveraging industry-renowned threat modelling frameworks is carried out in order to determine if there are design weaknesses or vulnerabilities in our features and for major changes; modeling is a significant catalyst for the creation of security requirements for our features and overall product, and the resulting threat model is continuously updated as the system evolves over time;
- **Security requirements are defined, discussed, documented** and embedded in the early design stages of the engineering process for any new features, and are reviewed multiple times to ensure correct implementation as part of the release process;
- **Security architecture vetting** of product features and any embedded third-party components is performed by our security experts in liaison with our experienced engineering teams;
- **Secure coding practices & code review** meaning that new code incorporated into our codebase is reviewed by our security-trained engineers and our security experts where applicable for weaknesses, vulnerabilities and adherence to best practices;
- **Automated CI/CD security testing** is performed by a myriad of automated tools in our pipelines including Static Application Security Testing (SAST), Software Composition Analysis (SCA), Dynamic Application Security Testing (DAST), as well as container image, secrets, IaC (Infrastructure-as-Code) scanning

leveraging cloud-native application protection (CNAPP) platforms such as Prisma Cloud alongside other security tooling.

- **Curated security testing** through penetration tests ran internally by our own security team and by reputable third-party service providers (note: SingleStore engages with an independent third-party service provider for pentesting our Helios offering annually at a minimum);
- **Transparent community-driven responsible disclosure** meaning that we recognize the benefits in engaging with the ethical hacker/security researcher community and leverage a vulnerability disclosure program accordingly to transparently improve upon our product in terms of security bugs;
- **Formal vulnerability & patch management practices** ensure that any findings from the previous controls are addressed. Each finding goes from their identification to resolution within documented timeframes in compliance with our security policies. SingleStore applies security patches and updates automatically.

11. Security Incident Management

SingleStore is committed to implementing, operationalizing and continuously improving a set of robust security incident management practices. We align our practices with the guidelines set forth by NIST's SP 800-61 and have defined provisions and security incident response plans accordingly for different scenarios on SingleStore Helios.

Our security incident management processes are tested yearly by an independent third-party service provider with knowledge and experience in this subject matter area and outputs of said tests are used to improve our security incident management capabilities.

In the event of a security incident where customer data has been exposed in any way or form, or where data protection notification requirements have been triggered, SingleStore will directly notify affected customers within the timeframes defined by regulations and or established contractual agreements, while simultaneously running through its incident response procedures to contain and eradicate the threat, and recover from the event.

For general security announcements issued by SingleStore to the public, please refer to the following resources:

- Security Section from SingleStore's Knowledge Base
- SingleStore Security Bulletins

Conclusion

Data security is a core fundamental of SingleStore Helios, and we align with industry guidelines and best practices to ensure data is protected throughout its lifecycle. In addition to the systems and practices we have already implemented for data security, we have continued investment in measures and certifications to ensure SingleStore stays on the cutting edge of security.

If you have further questions regarding security on SingleStore Helios, please do not hesitate to contact security@SingleStore.com.