CelerData Cloud BYOC security whitepaper

Introduction

CelerData Cloud Private, a powerful cloud offering for StarRocks, provides enhanced security capabilities. This whitepaper aims to provide you with a comprehensive understanding of the security measures implemented within the CelerData Cloud Private platform and equip you with the necessary knowledge to ensure the confidentiality, integrity, and availability of your data.

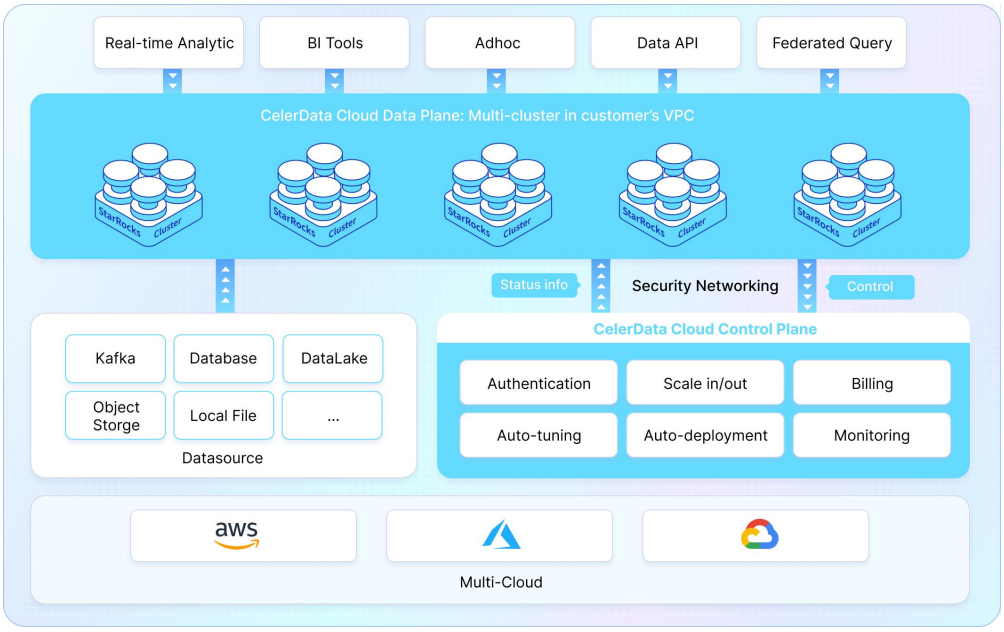
CelerData Cloud Private is a Bring Your Own Cloud (BYOC) service that enables organizations to leverage the power and flexibility of StarRocks, a high-performance analytics database, in their preferred cloud environment. With CelerData, you can harness the capabilities of StarRocks while maintaining control over your cloud infrastructure.

These principles were key in building the CelerData Cloud service, and continue to be adhered to as new features are added and in how the service is managed:

1. Data Sovereignty: The data plane resides within the customer's Virtual Private Cloud (VPC). CelerData's control plane handles cluster management, while user data remains under the customer's control.
2. Principle of Least Privilege(PoLP): This principle is consistently applied:
    a. Minimal IAM permissions are set during StarRocks deployment
    b. Access to ec2 host is controlled by the customer during troubleshooting
    c. RBAC is used for all operations to minimize unnecessary exposure.
3. Data Encryption and Compliance: CelerData Cloud Private prioritizes data security with encryption for transmission and storage. Holding SOC 2 and GDPR certifications showcases our commitment to rigorous security and compliance standards, ensuring data protection and regulatory adherence.

In the following sections, we will dive into the architecture of CelerData Cloud Private, highlighting the underlying components and their roles in ensuring the security of your data. We will then explore account management, authentication methods such as LDAP and Okta, privilege systems such as the RBAC model, and external privilege system integration.

Architecture



CelerData Cloud BYOC consists of two main components: the Control plane and the Data plane.

- Control plane: The Control plane serves as the central management hub for StarRocks clusters. It encompasses cluster deployment, monitoring, scaling, billing, optimization, WebGUI, and more. The Control plane is deployed within CelerData's VPC.
- Data plane: The Data plane includes the BE (Backend Engine) and FE (Frontend Engine) of the StarRocks cluster. The BE handles data storage and SQL execution, while the FE manages metadata storage and SQL plan generation and optimization. The Data plane is deployed within the customer's VPC. CelerData clusters support two modes: Classic cluster and Elastic cluster, corresponding to StarRocks community versions' shared-nothing and shared-data modes.
- Communication between the two planes occurs via a secure network connection (e.g., AWS Private Link). CelerData uses gRPC over TLS connections for both public access and private link service.

An example is worth one thousand words

Reading through this short example will provide you with a better understanding of where your data resides, who has access to it, and how that access is controlled.

Cindy is a data engineer working on a new dataset. This is her workflow:

Create a CelerData cluster

This is a new project with a new dataset, so Cindy decides to create a new CelerData cluster to work in. Cluster creation is initiated in the CelerData Cloud Private Control plane.

- Cindy logs in to CelerData Cloud with her company-provided Okta SSO credentials.
- Cindy creates a new cluster. After selecting the instance types, there is a choice to deploy manually or use a quick deployment, and whether to deploy in an existing VPC or a new one. Whichever choice is made, the destination for the cluster is in an AWS account that she has administrative rights to. The cluster is in the Data plane. Some important distinctions to call out here that are directly related to security and privacy are:
  - The Control plane and Data plane are separate
  - The Control plane is in a CelerData VPC and is an administrative UI
  - The Data plane is in your VPC and is where your data is stored; only your users and people authorized by you have access to your Data plane
  - You can create a VPC that meets your company security standards and deploy the Data plane there
  - Authentication to the Control plane and Data plane are separate. Separate accounts, and separate authentication methods are used.
- Cindy hands the Cloudformation details to her colleague who has the AWS privileges to create the cluster and the cluster is created. The initial password for the database is set during this process, and ongoing authentication will be set up afterward to comply with the rules set by Cindy's company.
- Once the Cloudformation job finishes Cindy clicks the **Finish** button in the Control plane and the cluster status is available in the Control plane.

Work with the CelerData cluster

The cluster connection details, other than the password as that is set by you during the Cloudformation step, are available in the Control plane on the specific cluster's overview page.

After collecting the hostname and port from the overview page in the Control plane, Cindy leaves the Control plane and from now on is working in the Data plane, which is hosted in a VPC in her company AWS account. Her first priority is to integrate the CelerData cluster with her company LDAP server so that she can collaborate with her team.

Once logged in as the database admin Cindy configues the authentication policy for the cluster to use her company LDAP service.

Work with your data

Now that the cluster is created and LDAP is configured Cindy and her colleagues can get to work. The data is stored in a VPC owned by Cindy's company, and configured to the company data access standards.

Collaborating with CelerData support

Sometimes you need a hand. CelerData support engineers do not have access to your Data plane, so they cannot see your queries, data, logs, etc. At some point, Cindy and her team may need help. They can request that CelerData support connect to their database by contacting support and inviting the support engineer to join the organization. After the session, the support engineer is removed from the organization.

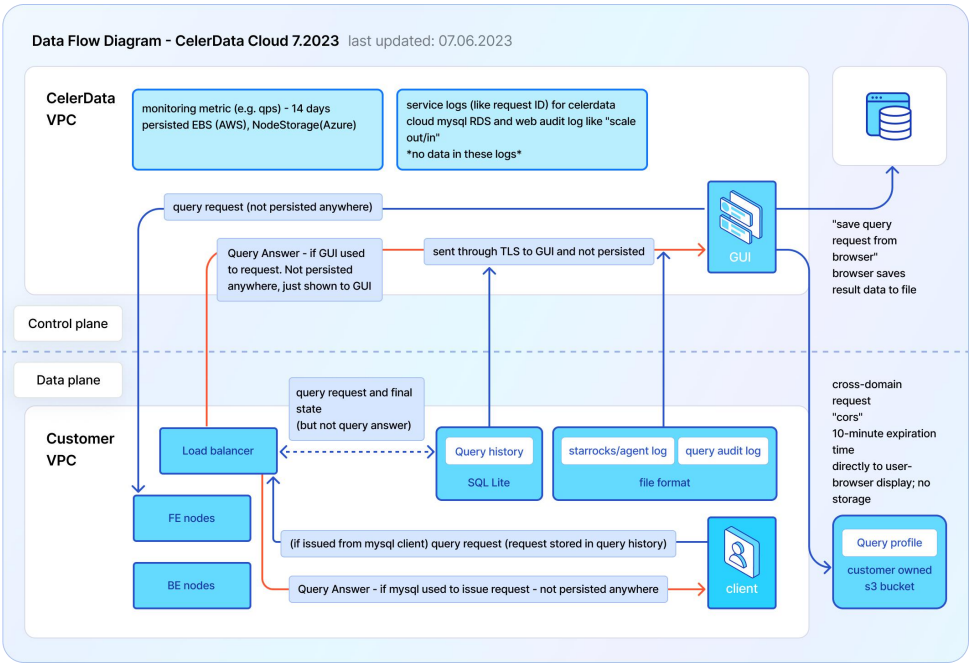
Data access

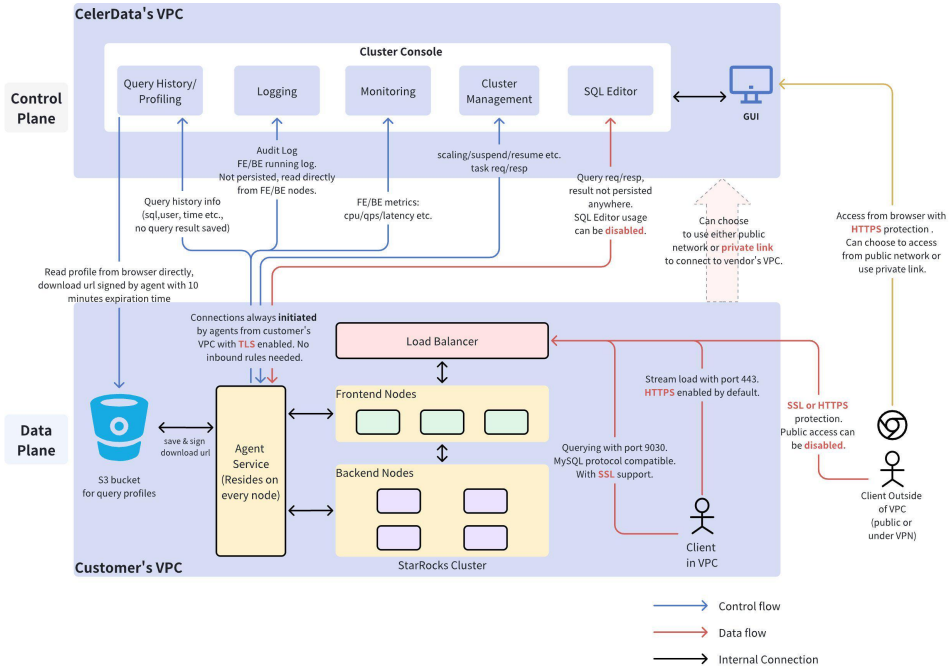
Data can be categorized into three levels:

- Level 0: User's data and metadata. All this information is stored within the customer's Data plane, and CelerData cannot access it unless granted explicit permission by users. Stored data and metadata are encrypted.
- Level 1: User query history, profiles, and audit logs. This data might contain sensitive information. CelerData can access this data in the Data plane with user authorization to assist in diagnosing slow queries and optimizing performance.
- Level 2: StarRocks logs and machine monitoring data unrelated to user-specific information. CelerData has default access to this data for monitoring and ensuring customer SLAs.

CelerData's Control plane does not store any of the above data.

For specific data storage and access methods, refer to the diagram below:





Accounts

CelerData Cloud Private includes both a data plane and a control plane, resulting in two separate account systems.

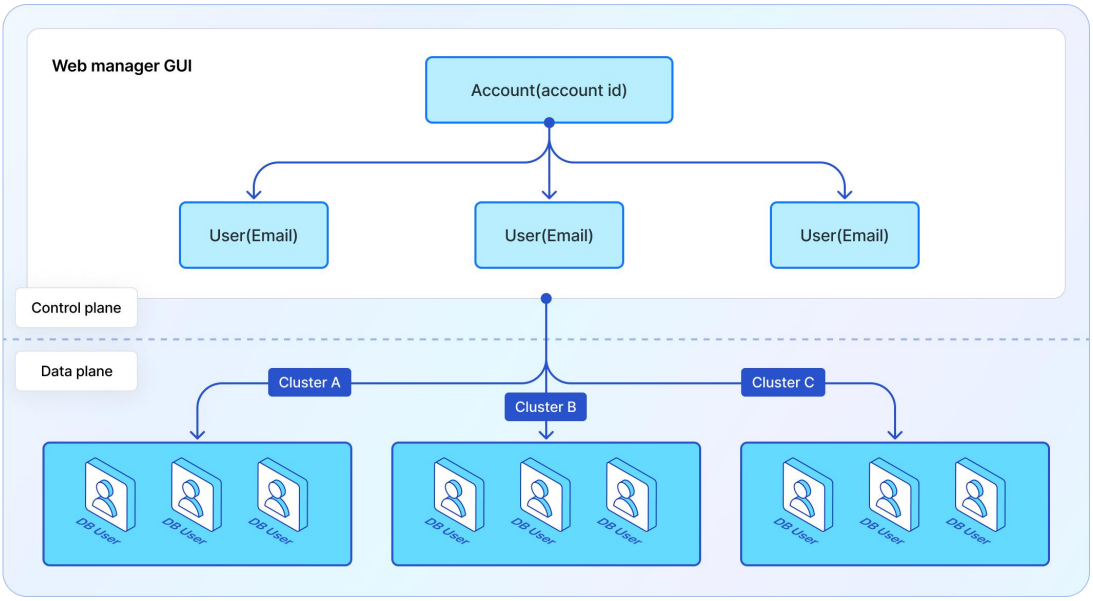
Control plane

The control plane is accessed with a web browser. Within an account, the following actions can be performed:

- Cluster Management
  - View Cluster details (including node inspection/cluster status monitoring)
  - Creation/deletion of clusters
  - Configuration changes
  - View alerts
- Manage AWS Authorization Information
- API Management
- User Management (addition, deletion, modification, and retrieval of the StarRocks cloud user under the account)
- View Usage Consumption Statistics
- Billing-related
  - Historical bill viewing
  - Current bill viewing
  - Settlement

Data plane

The data plane, on the other hand, is within the user's VPC and access is through database user accounts using the MySQL protocol supporting LDAP as a third-party authentication solution.



Authentication

To ensure the security isolation of data, the user management system of CelerData Cloud Private is divided into two sets—one for the control plane and one for the data plane. Therefore, authentication also consists of two independent systems. In the Control plane, users can log in using their CelerData accounts or integrate with third-party Single Sign-On (SSO) solutions like Okta. In the Data plane, login occurs through the MySQL protocol, supporting LDAP as a third-party authentication solution.

Web manager authentication

OKTA

SSO (Single Sign-On) enhances user experience and security by allowing access to multiple applications with a single set of credentials. Okta offers robust SSO solutions for seamless and secure authentication. StarRocks SSO support enables unified access control through integration with Okta, streamlining user management and bolstering data security.

Cluster authentication

MySQL authentication

StarRocks utilizes the MySQL protocol for compatibility, allowing direct connections through MySQL clients. CelerData ensures secure connectivity via HTTPS. Alternatively, you can establish a direct connection using the internal FE 9030 port, ensuring data bypasses CelerData's network entirely.

LDAP

LDAP (Lightweight Directory Access Protocol) is a protocol used for managing and accessing distributed directory services. It finds extensive use in scenarios requiring centralized user authentication and information storage, such as corporate internal email systems and other databases. Its mechanism involves hierarchical data organization, allowing efficient data retrieval and modification across various networked devices and platforms.
StarRocks integrates LDAP authentication to enhance security and user management. LDAP (Lightweight

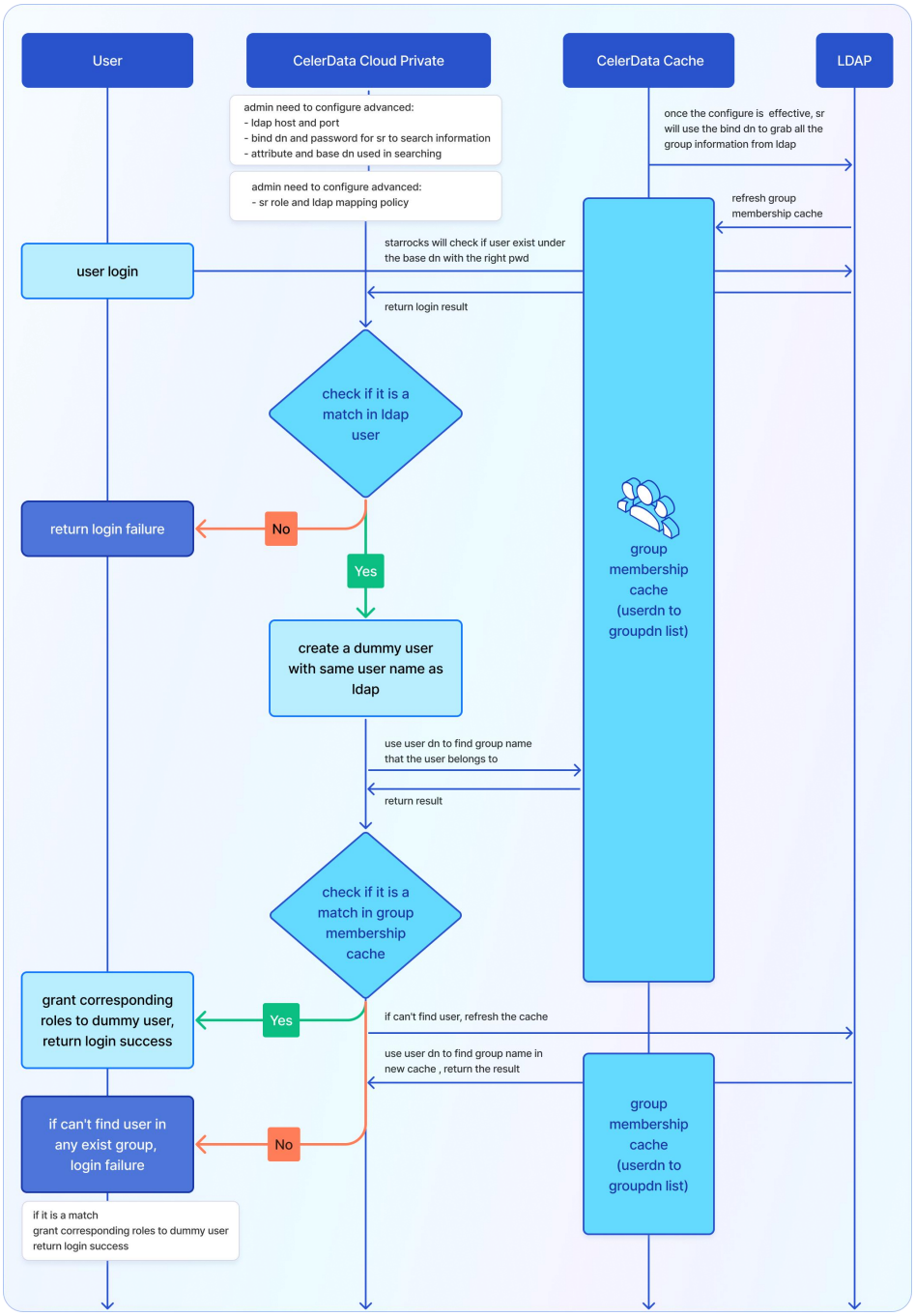
Directory Access Protocol) is utilized as an external authentication source. Users' credentials and roles are stored in the LDAP directory. When a user tries to access StarRocks, the system communicates with the LDAP server to verify the provided credentials. Upon successful authentication, the user's roles and permissions are fetched from LDAP. This integration streamlines user management, promotes consistency across systems, and enables centralized control over access rights in StarRocks, bolstering overall data security.

CelerData Cloud Private enhances the LDAP functionality within StarRocks through the following feature updates:

1. Direct LDAP Authentication: Users can now directly log in using their LDAP credentials, eliminating the need for manual user creation in StarRocks.

2. Automated Role Assignment: Users' roles are automatically determined based on their LDAP groups, alleviating the need for manual role assignments.

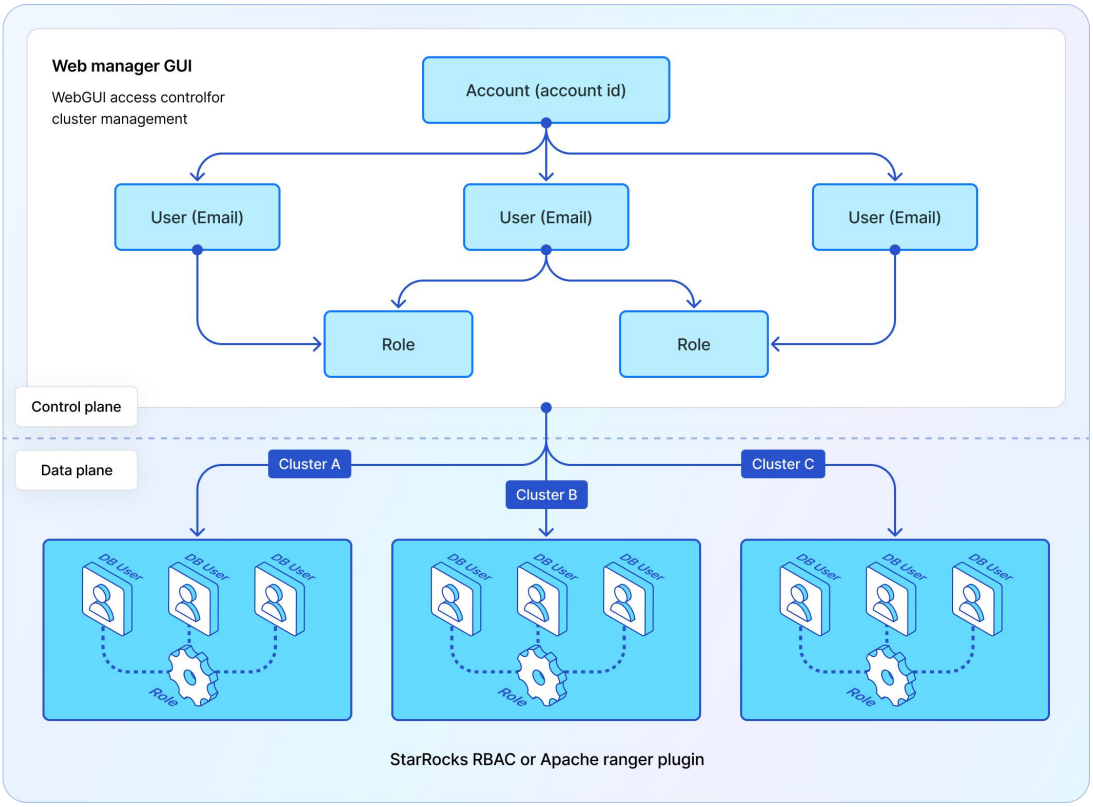
The following diagram shows the flow of interaction between StarRocks and LDAP:



By integrating this feature, organizations that wish to centrally manage user information via LDAP and match roles with LDAP groups can seamlessly manage user authentication and authorization within the CelerData Cloud platform. The implementation involves configuring connectivity, specifying authentication order, creating role mappings, and refreshing the cache to ensure synchronization between LDAP and StarRocks.

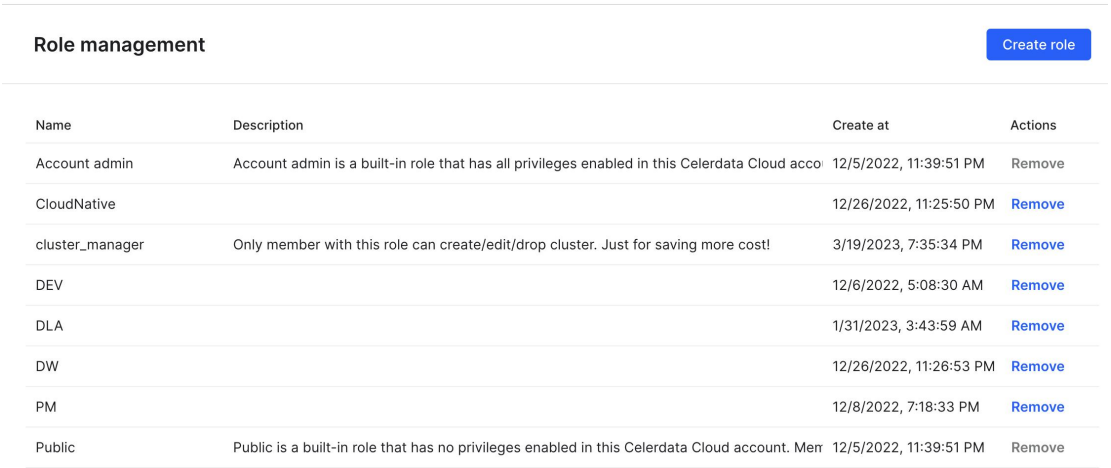
This advancement streamlines user management, optimizes security, and enhances the overall experience of LDAP integration within the CelerData Cloud data ecosystem.

Authorization



Web manager access control

In the CelerData Cloud Private web manager, you can create new roles and manage existing roles:

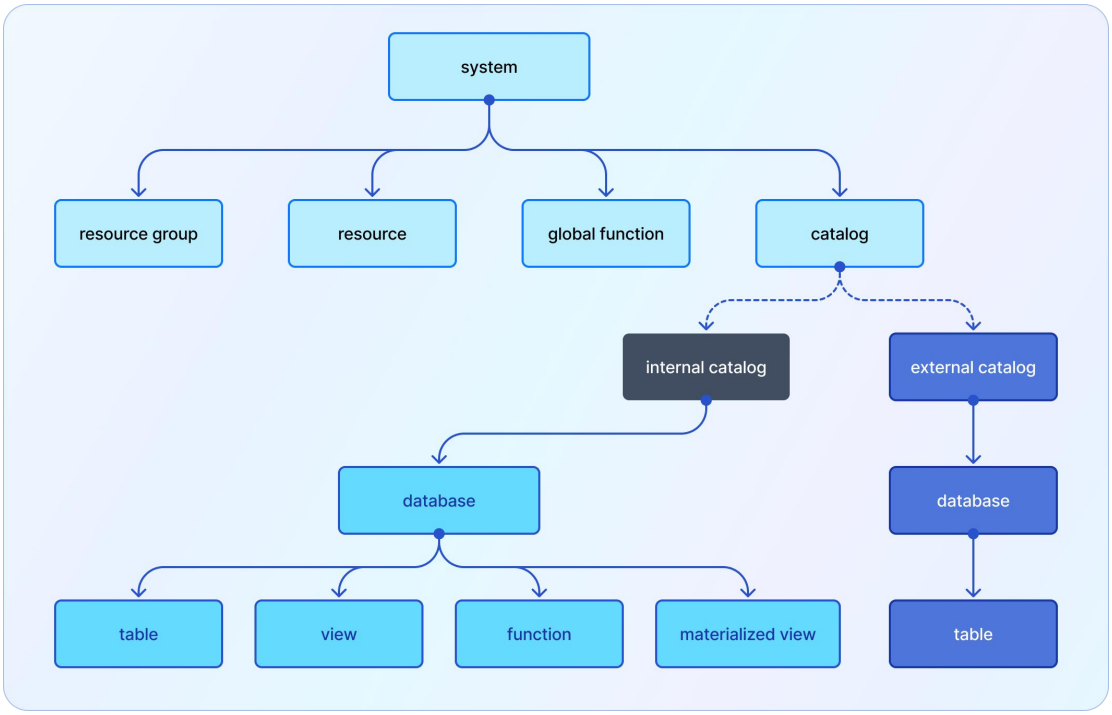


Cluster access control

StarRocks employs a role-based approach to permission management. Instead of assigning privileges directly

to users, privileges are assigned to roles. Users are then associated with one or more roles, simplifying user management and privilege assignment. However, to ensure compatibility with the old MySQL-style privilege system, we have retained the option to assign privileges directly to users.

The privilege system allows administrators to grant or revoke specific privileges at different levels of granularity, such as databases, tables, columns, and functions. This ensures that users have precisely the necessary permissions without unnecessary access. Here is a diagram to describe the StarRocks privilege item tree:



1. Predefined Roles: StarRocks comes equipped with various built-in roles for new users. These roles include root, cluster_admin, db_admin, and user_admin.

2. Role Hierarchy and Privilege Inheritance: Custom roles can be crafted to match specific business needs. Roles can also be assigned to other roles, establishing a hierarchy for efficient privilege inheritance and streamlining management. The maximum inheritance depth for a role is 16 levels.

3. Principle of Least Privilege via Role Switching: Users can switch to different roles using the "set role" command, adhering to the principle of least privilege. Active roles provide users access to role-specific privileges during their session. Default roles, activated upon user login, enhance privilege protection. Users without a default role still have the automatically activated public role.

Row access policy and column masking policy

In scenarios involving business data, there's a requirement to dynamically determine whether a role/user has permission to access specific rows, sensitive data columns, or obfuscated data. This dynamic capability ensures that data access is based on session information rather than storing multiple copies of the data. While views can address some of these needs, they have limitations when applied across multiple tables. Views lack flexibility in permission delegation and can lead to management complexities.

Column Masking and Row Access policies are employed to safeguard sensitive data from unauthorized access while allowing authorized users to query sensitive data during runtime. These policies do not modify the actual data within existing tables but rather apply data transformations based on user-defined conditions. This enables column-level data masking or row-level data filtering. For instance:

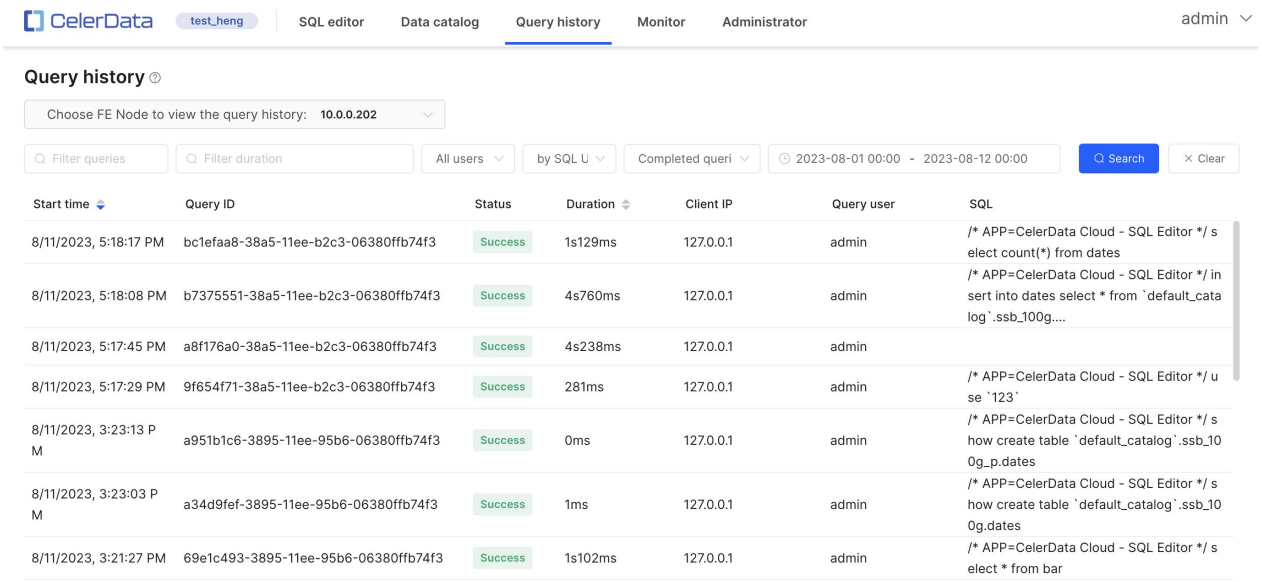
- A Column Masking Policy can dynamically return either encrypted or unencrypted data columns based on a user's role.
- A Row Access Policy filters rows based on a user's role and specific attributes, for example, `region`.

The benefits of these policies include flexibility for multiple use cases, simplified data modification, avoidance of excessive view creation, and more granular access control. They can be applied to TABLEs, VIEWs, MaterializedViews, and External Table objects. The policies are effective in queries, insertions, CTAS, updates, and deletions.

Auditing

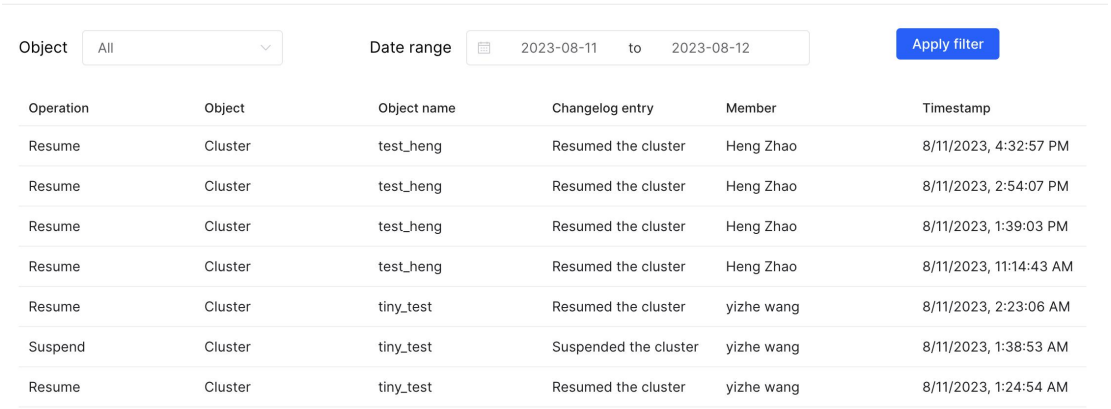
Audit logs play a crucial role in monitoring and maintaining the security and compliance of systems. They provide a detailed record of various actions and activities performed within a system, such as database operations, configuration changes, and user interactions. Audit logs enable organizations to track and analyze these events, ensuring accountability, detecting anomalies, and aiding in incident investigations.

You can find all the database-related audit logs in StarRocks' `fe.audit.log`, which encompass activities like DDL (Data Definition Language) and DML (Data Manipulation Language) operations. These audit logs are maintained at the cluster level, allowing you to access comprehensive information through the query history page on the Cluster GUI. Admins (users who have `user_admin` and `db_admin` roles) have the privilege to check all of the query history, and users without `user_admin` and `db_admin` roles can only access history query records run by themselves. No cluster audit logs are allowed for these users.



Additionally, within the CelerData web manager page, you can access a range of information related to cluster actions such as purchases, deletions, scaling, suspensions, and resumptions.

The diagram below illustrates this functionality.



Data Security

Data encryption in transit

CelerData Cloud Private uses SSL during transit.

Data encryption at rest

StarRocks persists metadata on the FE node's disk. This metadata includes, but is not limited to, all of the table schema and data distribution information. In CelerData Cloud, we provide optional disk encryption capability. This ability ensures all the data in EBS cannot be directly parsed in plain text after the data is obtained, which guarantees data security and minimizes potential safety hazards caused by data leakage.

Also, in an elastic cluster, CelerData Cloud can persist the table data in the customer's cloud storage, for example, in an S3 bucket. With the SSE encryption capability enabled by default in AWS S3, CelerData Cloud (BYOC) ensures that the data in the elastic cluster will not be accessed in plain text by the AWS S3 server. On this basis, users can choose to encrypt the S3 bucket with their own AWS KMS key to SSE encrypt the data in an elastic cluster in a customized way.

User identity information such as username/password is stored in the database with field-level encryption (Bcrypt algorithm) to ensure that the user identity information cannot be parsed in plain text on the server side.

Backup service

In the event of hardware failure, human error, malicious attacks, software bugs, and natural disasters, and to meet compliance requirements, backup and recovery serve as an important safeguard for database systems to protect data security.

StarRocks backup refers to the process of creating copies of data to prevent data loss, while restore is the process of utilizing backup files to reconstruct the database back to the state at the time the backup was taken.

Starrocks supports backing up to remote object storage and HDFS, supports incremental backups based on partitions, and supports backup and recovery at the database level. The destination for a backup is known as a repository.

Here are some scenarios of backup and restore:

1. Regular Backup

To prevent data loss, databases need to be backed up regularly, usually by setting a backup policy, e.g. backing up the database every night at midnight. Regular backups ensure that the database can be restored to a previous state in case of any problems.

2. Backup Before Upgrade

When upgrading the database version, a backup could be performed before the upgrade to prevent any accidents during the upgrade process. The backup can be used to restore the database.

3. Data Migration

When migrating a cluster to a new cluster, a backup should be done first, then the backup data can be restored in the new cluster to ensure data integrity during migration.

4. Emergency Recovery

If the database fails or data corruption occurs due to an attack during operation, the backup can be used to perform emergency recovery and restore the database back to the state before the failure.

5. Test Environment Restore

Developers often need to use real data in test environments for testing. Migrating a production backup into a development environment can be used to provide the necessary real data.

Disclaimer: Information provided in this Security Whitepaper is aligned with current laws and regulations as of April 2024. Content may change from time to time due to changes in local or international laws and regulations. A new version of this document will be published at the time revisions have been made. Any questions related to CelerData's security process and procedures can be directed to privacy@celerdata.com