NebulaGraph Cloud

20250522

NebulaGraph

Table of contents

1. What is NebulaGraph Cloud

NebulaGraph Cloud (Cloud for short) is a fully-managed cloud service that integrates the NebulaGraph database with database-as-a-service (DBaaS). Cloud supports one-click deployment, allowing users to create graph databases within minutes and quickly scale computing, storage, and other resources.

1.1 Features

- NebulaGraph Cloud provides a pay-as-you-go and pay-on-demand subscription. You can create NebulaGraph instances at any time and choose from various resource combinations tailored to your business needs.
- You can quickly become a graph expert, even without previous experience in graph database operations. Easily manage graphs, import datasets, explore data insights, and run GQL statements—all from the intuitive visual interface of NebulaGraph Cloud.
- You can create private endpoints to securely connect to the cloud database using SDKs in different programming languages.
- NebulaGraph Cloud provides extensive monitoring metrics to help you infer the internal state of the database service and its infrastructure in real time.

1.2 Product advantages

- Always-on and zero-admin: NebulaGraph Cloud offers a fully managed NebulaGraph database service, allowing you to focus on application development without worrying about operations and maintenance.
- Flexible and efficient: You can quickly build a NebulaGraph database without purchasing physical servers, storage devices, computing, or network components. It only takes 5-10 minutes to prepare a complete environment for use.
- Easy to use: NebulaGraph Cloud provides online management of NebulaGraph clusters on the Web, corresponding visualized tools of NebulaGraph, and a demo dataset to help users get started quickly.
- Official support: You can quickly create tickets for technical support, consulting problems you have encountered during use.

1.3 Scenarios

NebulaGraph Cloud is suitable for the following scenarios:

- You want to quickly subscribe to the NebulaGraph service and deploy a fully-managed graph database on AWS without coding.
- You want to leverage NebulaGraph Cloud as the infrastructure to rapidly and economically enjoy the Nebula AI Application Platform service.

1.4 Version compatibility

The table below shows the compatible versions of NebulaGraph Cloud and the corresponding NebulaGraph for accurate deployment.

NebulaGraph Cloud version	NebulaGraph version
20250522	5.0.2
20250427	5.0.2
20250415	5.0.2
20250402	5.0.2
20250313	5.0.2
20250228	5.0.0
20250214	5.0.0
20250128	5.0.0

2. Quick start

2.1 NebulaGraph Cloud quickstart guide

NebulaGraph Cloud is a cloud-native graph database service offered by NebulaGraph. It provides a web-based user interface (UI) for you to create and manage databases without the trouble of managing the underlying infrastructure.

To quickly master NebulaGraph Cloud, see the following topics:

- Step 1 Create a database instance
- Step 2 Get started with Cloud

2.2 Create a database instance

Before getting started, you need to create a NebulaGraph Cloud account, create a database, and then connect to it. This topic covers how to create a database instance.

2.2.1 Prerequisites

An AWS account is prepared.

2.2.2 Step 1. Sign up and log in to NebulaGraph Cloud

1. Open this URL in your browser.
2. Click **Sign Up Now** to create a NebulaGraph Cloud account using your email, Github or Google account.

   If using your email to sign up, you will receive an email containing a **Verify** button after registration. Click the button to verify your email address and set your password.

2.2.3 Step 2. Create a database

1. On the **Database** page that appears by default, click **Create Database**.
2. On the **Create database** page, specify the following information:

- Database Tier: Select a type of plan based on your requirements. For more information about the differences between the paid plans, see Upgrade to a paid plan.

- Basic
- **Database Name**: Customize a name for the database.
- **Cloud Provider**: **AWS** is selected by default.
- **Region**: Choose the region based on your business requirements. For more information about the available regions for your business, see AWS Regions.
- `us-east-2` : US East (Ohio)
- `us-west-2` : US West (Oregon)

- Capacity: Choose node and storage specifications based on your requirements.

  New users can enjoy a 14-day free trial per NebulaGraph Cloud account, which comes with one database instance of 2 vCPUs, 4 GB of RAM, and 10 GB of storage, without requiring any payment information. For more information, see Try NebulaGraph Cloud for free.

- Payment Method: Click **Add a payment method** to bind your AWS account as the payment method. You are prompted to log in to your AWS account and purchase NebulaGraph Cloud on the AWS Marketplace.

> **Note**
>
> After you complete the binding, it may take a few minutes for the payment method to be displayed on the NebulaGraph Cloud Web UI. You can refresh the page to check the status.

3. Review the summary on the right, then click **Create**.

2.2.4 (Optional) Step 3. Connect to your database using SDKs

If you want to use SDKs to connect to your database, complete **this step** to configure network access. Otherwise, skip **this step**.

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane, click **Network access** under **Security**. NebulaGraph Cloud allows you to access the database through public network or private network. Follow either of the following steps based on your requirements:

- Access the database through public network:

a. On the **IP Allow List** tab, click the toggle switch to enable accessing the database through public network.

b. Click **+ Add IP Addresses** to add custom IP addresses to the allow list.

- Access the database through private network:

a. Click the **Private Endpoint** tab.

b. On the **Create Private Link Service** pane, click **Create** to enable the private link service.

c. On the **Create Private Link Endpoint** pane, specify your VPC ID and subnet ID, and run the automatically generated command in your AWS CLI to create the private link endpoint.

3. On the left-side navigation pane, click **Overview** under **Basic**.
4. In the upper-right corner of the **Overview** page, click **Connect**.
5. In the **Connect to your database** dialog box, specify the following items and click **Confirm**:

- Choose a connection type: Select **Public (Connect via public network)** or **Private (Connect via private endpoint)**.
- Install the client: Select **NebulaGraph Java**, **NebulaGraph Go**, or **NebulaGraph Python**, and click **Download installation package** for installation.
- Connect: Click **Generate** and take note of the following connection information: **Host**, **Port**, **Username**, and **Password**. If you need to refresh the password, click **Regenerate**.

6. Set up the SDKs with the connection information. For more information, see Java SDK guide, Go SDK guide, and Python SDK guide.

2.2.5 What's next

- Get started with Cloud
- Manage database instances

2.3 Get started with NebulaGraph Cloud

This guide covers how to quickly get started with NebulaGraph Cloud.

> **Note**
>
> If you need to log in to the GQL documentation center during any operations, enter the name of your NebulaGraph Cloud account and the initial password `Nebula.123`. You will be prompted to change the initial password when you log in for the first time.

2.3.1 Prerequisites

A database instance is created. For more information, see Create a database instance.

2.3.2 Step 1. Create graphs

Before you import data, you must create a graph to store the data to be imported. In NebulaGraph Cloud, graphs are created based on graph types, which define the data structure of the graphs. To ensure that your data is imported correctly, the graph type must match the data structure of the source data.

We have provided a demo dataset for you to quickly try out NebulaGraph Cloud. The demo dataset contains a graph type and a graph, which are created based on the demo data. If you want to use it, skip this step and jump to step 2. If you want to create your own graph type and graph, follow the steps below.

1. Click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. In the upper-left corner of the **Graph** page, follow the instructions introduced in Create a graph type and Create a graph to create a graph type and a graph.

2.3.3 Step 2. Import data

1. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
2. You can import demo data or your custom data into the database. Follow either of the following methods based on your requirements.

- Use the demo dataset

  On the **Graph** page, click **Use Demo Dataset** to load the demo dataset in NebulaGraph Cloud. For more information, see Demo dataset.

- Import custom data

a. Go to the database details page, and click **Import** on the left-side navigation pane.

b. On the **Import** page, click **Import Data** and follow the instructions introduced in Create import task to import your data.

2.3.4 Step 3. Run queries and explore data

On the left-side navigation pane of the database details page:

- Click **Explore** to explore data on canvas. For detailed instructions, see Explore graph.

- Click **Query** to perform GQL queries in the console.

  For how to write GQL queries, see GQL guide. (Initial password: Nebula.123)

  For how to navigate around the console, see Data query.

2.3.5 Step 4. Monitor database information and metrics

On the left-side navigation pane of the database details page:

- To view the basic information and core metrics of the database, click **Overview** under **Basic**.
- On the **Basic** pane, you can view the basic information about the database, such as the cloud provider, status, and capacity.
- On the **Core Metrics** pane, you can select the time range, start time, and end time to view the core database metrics.

- To view all the database metrics, click **Metrics** under **Monitoring**. Then, select the time range, start time, end time, and fresh rate to view the metrics, including:

- The metrics of NebulaGraph service on the **Service** pane:
- **Query Per Second (QPS)**: The number of queries executed per second.
- **Total Sessions**: The total number of sessions within a specified period.
- **Query Error Rate**: The percentage of failed queries.
- **Query Latency (P95)**: The 95th percentile latency of all queries sent by the Graph service to the Storage service in the last 5 seconds.
- **DML Latency (P95)**: The 95th percentile latency of DML queries sent by the Graph service to the Storage service in the last 5 seconds.
- **DQL Latency (P95)**: The 95th percentile latency of DQL queries sent by the Graph service to the Storage service in the last 5 seconds.

- The metrics of storage nodes and query nodes on the **Infrastructure** pane:
- **CPU Utilization**: The percentage of CPU used by the database.
- **Memory Utilization**: The percentage of memory used by the database.
- **Storage Used**: The amount of storage used by the database.
- **Write IOPS**: The write input or output operations per second (IOPS) for each Amazon Elastic Block Store (EBS).
- **Read IOPS**: The read input or output operations per second (IOPS) for each Amazon Elastic Block Store (EBS).

2.3.6 Related topics

- Graph management
- Data import
- Data exploration
- Data query

2.4 Demo dataset

This topic introduces the demo dataset that you can use in NebulaGraph Cloud and demonstrates how to load it in NebulaGraph Cloud.

2.4.1 Dataset introduction

The demo dataset is a knowledge graph dataset for movie recommendation. The following figure shows the entities and relationships in the knowledge graph:

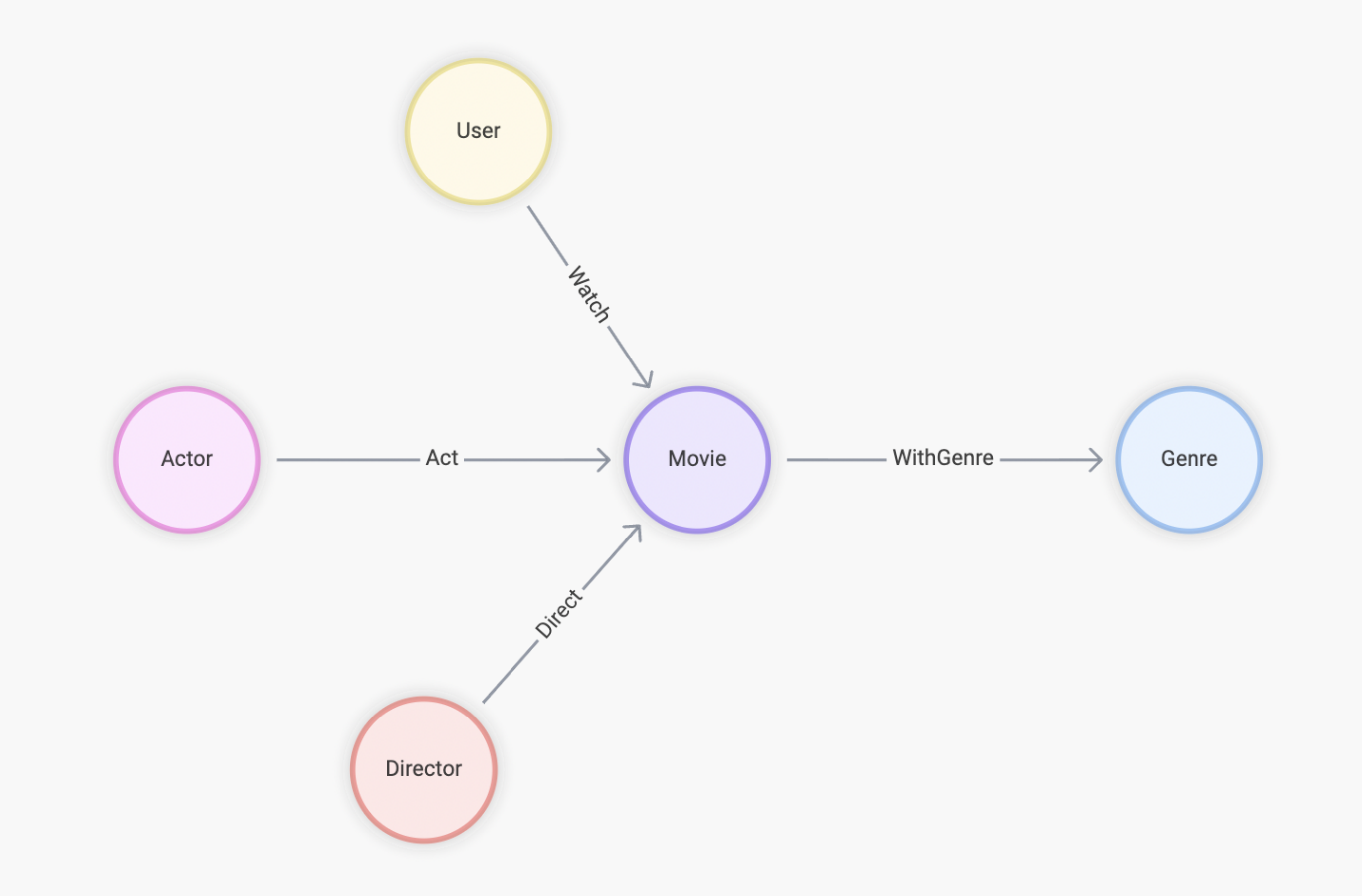


Figure 1: Entities and relationships in the knowledge graph

To implement the dataset in GQL, the following node types and edge types are introduced:

Node type name	Label	Property	Primary key
User	`Person`	`id`	`id`
Actor	`Person`	`id` , `name` , and `birthDate`	`id`
Director	`Person`	`id` , `name` , and `birthDate`	`id`
Movie	`Movie`	`id` and `name`	`id`
Genre	`Genre`	`id` and `name`	`id`

Table 1: Node types in the dataset

Edge type name	Label	Property	Source node type name	Destination node type name
Watch	`Watch`	`rate`	`User`	`Movie`
Act	`Act`	None	`Actor`	`Movie`
Direct	`Direct`	None	`Director`	`Movie`
WithGenre	`WithGenre`	None	`Movie`	`Genre`

Table 2: Edge types in the dataset

The following figure shows the graph model of the dataset in a tabular format:

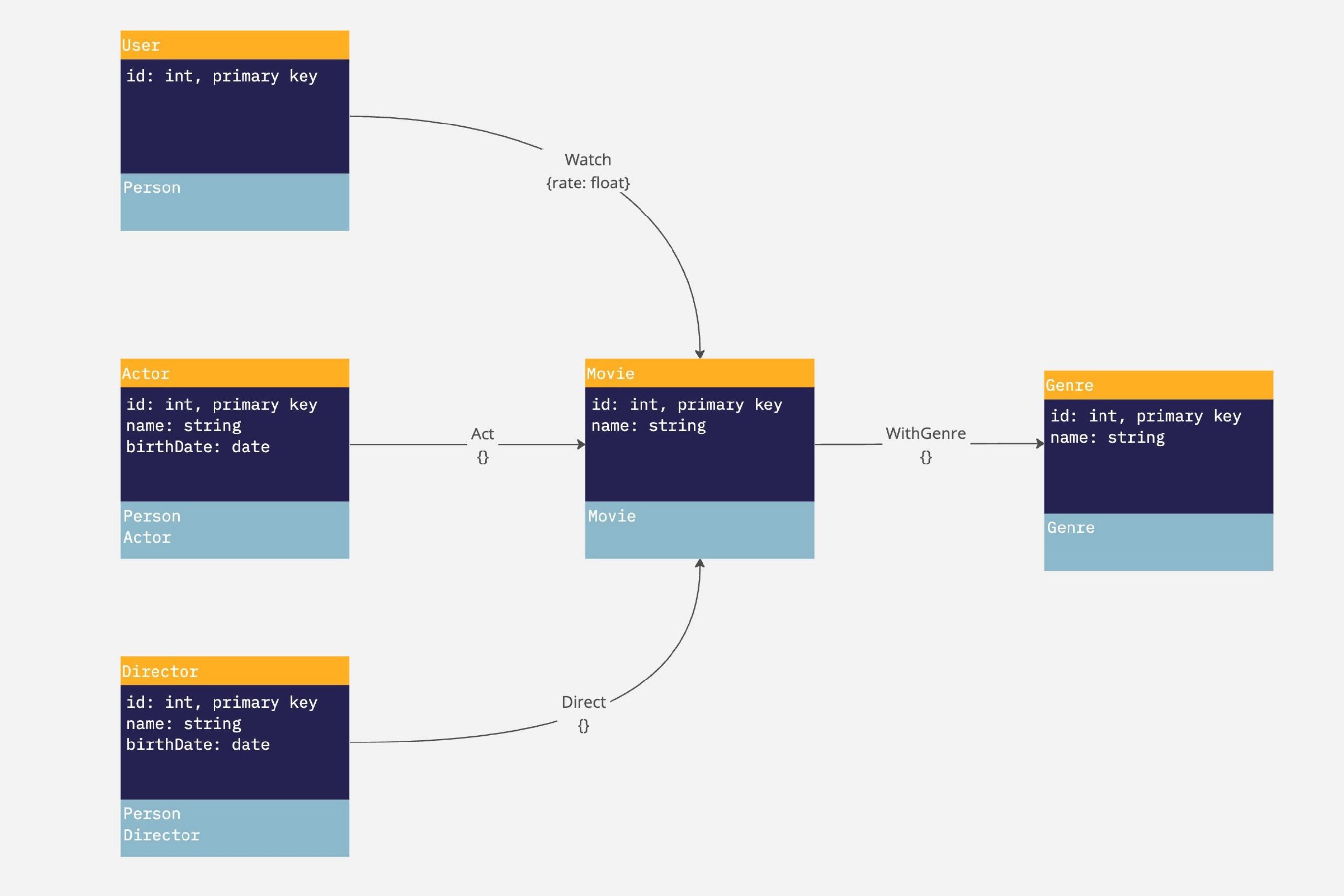


Figure 2: Graph model of dataset

2.4.2 Load the demo dataset into NebulaGraph Cloud

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. On the **Graph** page, click **Use Demo Dataset** to load the demo dataset.

2.4.3 Download the dataset files

To obtain the original dataset files, click here to download the complete dataset files in CSV format.

3. Pricing

This topic introduces the pricing model for NebulaGraph Cloud.

3.1 Overview

NebulaGraph Cloud offers pay-as-you-go pricing for all of its services. You pay only for the resources you use, with no upfront costs or long-term contracts. However, you do need to subscribe to NebulaGraph Cloud on the AWS Marketplace to use the service.

After creating a database instance on NebulaGraph Cloud, you are charged based on three factors: database node specification, disk storage specification and data transfer usage.

3.1.1 Node specification

The charges for node specification differ depending on the plan you choose.

For standard plan

For standard plan users, the node specifications are determined by the number of vCPUs and the amount of RAM. The following table lists the available node specifications and their corresponding hourly prices:

Node specification	Price/hour
2 vCPUs, 4 GB RAM	$0.049
2 vCPUs, 8 GB RAM	$0.91
4 vCPUs, 16 GB RAM	$1.83
8 vCPUs, 32 GB RAM	$3.66
8 vCPUs, 64 GB RAM	$3.76

For professional plan

For professional plan users, node specifications are divided into storage node specifications and query node specifications. The following table lists the available node specifications and their corresponding hourly prices:

- Storage node specification

Storage node specification	Price/hour
2 vCPUs, 16 GB RAM	$0.79
4 vCPUs, 32 GB RAM	$1.59
8 vCPUs, 64 GB RAM	$3.17

- Query node specification

Query node specification	Price/hour
2 vCPUs, 8 GB RAM	$0.49
4 vCPUs, 16 GB RAM	$1.01
4 vCPUs, 32 GB RAM	$1.04
8 vCPUs, 32 GB RAM	$2.01
8 vCPUs, 64 GB RAM	$2.08

3.1.2 Disk storage specification

The following table lists disk storage specifications and their corresponding hourly prices for both standard and professional plan users:

Storage size per node	Price/GB/hour
From 30 GB to 160 GB	$0.0001
From 161 GB to 320 GB	$0.0002

The charges for the disk storage follow AWS pricing. For more information about AWS EBS pricing, see the AWS documentation.

3.1.3 Data transfer usage

For both standard and professional plan users, the charges for data transfer follow AWS pricing and consist of the following:

- Fixed cost: the hourly charge of an AWS Network Load Balancer (NLB), which is $0.0225.
- Capacity unit cost: the number of Network Load Balancer Capacity Units (NLCU) used by the NLB per hour, which is $0.006 per NLCU-hour.

For more information about AWS NLB pricing, see the AWS documentation.

3.2 Try NebulaGraph Cloud for free

NebulaGraph Cloud offers a 14-day free trial that allows you to explore the service at no cost, which includes the following features:

- 14-day trial: You can use a database instance with 2 vCPUs, 4 GB of RAM, and 10 GB of storage for free. Your free trial expires 14 days after instance creation or when the instance is deleted, whichever comes first.
- Upgrade anytime: You can subscribe to NebulaGraph Cloud at any time during the free trial to upgrade to a paid plan. Once you upgrade, the free trial ends.
- No payment information required: Your payment account is required only when you decide to upgrade to a paid plan.

3.3 Upgrade to a paid plan

To unlock more features and continue using NebulaGraph Cloud, click **Trial - x day left** in the top-right corner of NebulaGraph Cloud. Follow the instructions to upgrade to a paid plan, starting at $0.05 per hour, which includes 2 vCPUs, 4 GB of RAM, and 10 GB of storage.

The paid plans offer a wide range of enterprise-grade features. The following table compares the paid plans, detailing the specific features available in each plan:

Feature	Standard plan	Professional plan
Scenario	For users who want to quickly set up a database for development, learning, experimentation, or prototyping.	For users, such as startups and scale-ups, who require a production-ready database with a storage-compute separation architecture that ensures high availability, scalability, and better performance.
Pricing factors	• Node specification • Disk storage specification • Data transfer usage	• Storage node specification • Query node specification • Disk storage specification • Data transfer usage
Scale your database on demand	Supported	Supported
Capacity	Supported a single node with up to 8 vCPUs, 64 GB RAM, and 320 GB of storage size.	Supported up to 10 query nodes and 10 storage nodes, offering up to 8 vCPUs, 64 GB RAM, and 320 GB of storage size for each storage node.
Private endpoint and IP allow list	Supported	Supported
High availability (HA)	N/A	Supported
Technical support	Supported	Supported with faster response time
Advanced service	N/A	• Larger storage sizes • Higher specification resource options

3.4 Billing

To view your billing details, go to AWS Billing Console.

3.5 Cancelation

To cancel your subscription, go to AWS Marketplace and unsubscribe from the NebulaGraph Cloud service. Note that once the subscription is canceled, your database instance will be immediately deleted and all data will be lost.

4. Instance management

This topic introduces how to manage database instances in NebulaGraph Cloud, including suspending, resuming, scaling, and deleting instances.

4.1 Prerequisites

A database instance is created. For more information, see Create a database instance.

4.2 Suspend & resume an instance

To suspend an instance, click **...** of an instance on the **Database** page or its details page. In the drop-down list, select **Suspend** and confirm this action in a pop-up window.

When the instance is suspended, note that:

- Only the following costs will be charged:
- Disk storage costs: $0.0002/GB/hour
- Fixed costs for data transfer usage: $0.0225/hour (charged if access to the instance through a public or private network is enabled)
- The instance is unaccessible, and read and write operations are impossible.
- The instance metrics are not monitored.
- The instance cannot be scaled.

To resume a suspended instance, click the **Resume** button for a database instance on the **Database** page. After confirming this action in a pop-up window, the instance's status will change from **Suspended** to **Resuming**, and then to **Running**. At this point, you will be charged fully based on your service plan and the affected items above will be restored.

4.3 Scale an instance

Note

- To scale an instance, you must upgrade to a paid plan and the instance is in the **Running** status.
- During scaling, the instance will be temporarily unaccessible.

When you first upgrade to a paid plan, the instance remains at 2 vCPUs, 4 GB of RAM, and 10 GB of storage. You can purchase a capacity bundle to reallocate the baseline resources.

After purchasing a capacity bundle, you can further scale your instance according to your workload requirements by using the following methods:

- Resize the node specification
- Increase the storage size

4.3.1 Purchase a capacity bundle

To purchase a capacity bundle, follow these steps:

1. Click **...** of an instance on the **Database** page or its details page.
2. In the drop-down list, select **Upgrade capacity**.
3. In the pop-up window, select a capacity bundle and click **Upgrade capacity**.

4.3.2 Resize node specification

To increase or decrease the vCPU and RAM resources for an instance, follow these steps:

1. Click **...** of an instance on the **Database** page or its details page.
2. In the drop-down list, select **Resize**.
3. In the pop-up window, select a node specification and click **Resize**.

4.3.3 Increase storage size

To increase the storage capacity for an instance, follow these steps:

1. Click **...** of an instance on the **Database** page or its details page.
2. In the drop-down list, select **Increase storage size**.
3. In the pop-up window, set the new storage capacity and click **Increase storage size**.

Note that after the storage size is increased, you are not allowed to change it within the following 6 hours due to the limitation of Amazon EBS.

4.4 Delete an instance

To delete an instance, click **...** of an instance on the **Database** page or its details page. In the drop-down list, select **Delete** and confirm this action in a pop-up window.

For free trial users, your instance will be automatically deleted once the trial ends unless you upgrade to a paid plan beforehand.

5. Graph management

5.1 Graph management

A schema in NebulaGraph is a system object that serves as a container for graph types and graphs. NebulaGraph Cloud provides a visual interface for you to manage graph types and graphs in a default schema `/default_schema`. For details, see the following topics:

- Manage graph types
- Manage graphs

5.2 Manage graph types

This topic introduces how to manage graph types in NebulaGraph Cloud, including creating, deleting, modifying, and viewing graph types.

5.2.1 Create a graph type

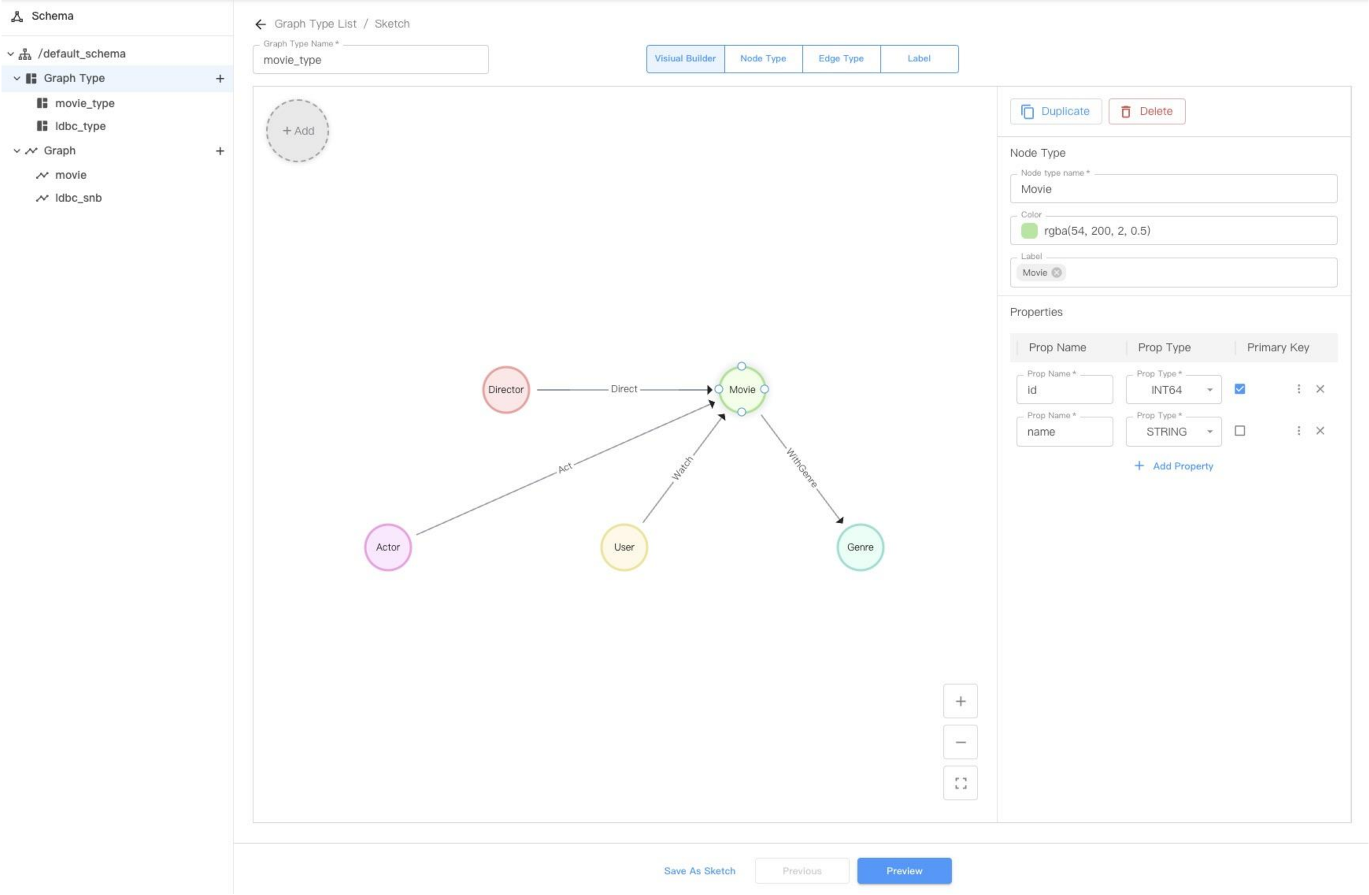
1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. In the upper-left corner of the **Graph** page, click the plus icon next to **Graph Type**.
4. Enter a name for the graph type in the **Graph Type Name** field.
5. Drag the **+Add** circle to the canvas to create a node type. For each node type, specify the following information:

- **Node Type Name**: The name of the node type, which must be unique among all node types and edge types within the graph type.
- **Color**: The color of the node type.
- **Label**: The label of the node type, which must be unique within the node type.
- **Properties**: The properties of the node type. Each node type must have at least one property. For each property, specify the following information:
- **Prop Name**: The name of the property, which must be unique within the node type.
- **Prop Type**: The data type of the property.
- **Primary Key**: Whether the property serves as the primary key of the node type. Each node type must have exactly one primary key, which may consist of one or more properties.
- **Is List**: Whether the property value is a list.
- **Nullable**: Whether the value of the property can be `NULL`.
- **Default Value**: The default value of the property.

6. Repeat step 5 to create more node types based on your requirements.
7. Select two node types and draw a line to connect them to create an edge type. For each edge type, specify the following information:

- **Edge Type Name**: The name of the edge type, which must be unique among all node types and edge types within the graph type.
- **Direction**: The direction of the edge type.
- **Color**: The color of the edge type.
- **Label**: The label of the edge type, which must be unique within the edge type.
- **Multi Edge Key Mode**: The multi-edge key mode of the edge type. Valid values are as follows:
- **None**: The edge type has no multi-edge key. In this case, multiple edges between two nodes are not allowed.
- **Auto**: The multi-edge key is automatically generated by the system.
- **Customize**: You need to specify the properties that form the multi-edge key.
- **Properties**: The properties of the edge type. For each property, specify the following information:
- **Prop Name**: The name of the property, which must be unique within the edge type.
- **Prop Type**: The data type of the property.
- **Multi Edge Key**: Whether the property serves as the multi-edge key of the edge type. This field is displayed only when the field **Multi Edge Key Mode** is set to **Customize**. Each edge type may have one multi-edge key, which may consist of zero or more properties.
- **Is List**: Whether the property value is a list.
- **Nullable**: Whether the value of the property can be `NULL`.
- **Default Value**: The default value of the property.

8. Repeat step 7 to create more edge types based on your requirements.
9. Click **Preview** to preview the GQL statement that is generated based on the graph type you designed.
10. Click **Create Graph Type** to create the graph type.



Graph Type Name *: movie_type

Node Type

Node type name *: Movie

Color: 🟩 rgba(54, 200, 2, 0.5)

Label: Movie

Properties

Prop Name	Prop Type	Primary Key
id	INT64	☑
name	STRING	☐

For more information about the GQL syntax for creating graph types, see CREATE GRAPH TYPE. (Initial password: Nebula.123)

5.2.2 Delete a graph type

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. In the upper-left corner of the **Graph** page, click **Graph Type**.
4. On the **Graph Type List** page that appears on the right, select the graph type you want to delete.
5. Click ••• at the right of the graph type and select **Delete**.
6. In the confirmation dialog box that appears, click **Delete**.

For more information about the GQL syntax for deleting graph types, see DROP GRAPH TYPE. (Initial password: Nebula.123)

5.2.3 Modify a graph type

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. In the upper-left corner of the **Graph** page, click the graph type you want to modify under the **Graph Type** section.
4. Click the **Edit Graph Type** button in the upper-right corner.
5. On the canvas that appears, click the node type or edge type you want to modify and make the necessary changes.

6. Click **Preview** to preview the GQL statement that is generated based on the changes you made.
7. Click **Commit Edit** to apply the changes.

For more information about the GQL syntax for altering graph types, see ALTER GRAPH TYPE. (Initial password: Nebula.123)

5.2.4 View a graph type

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. In the upper-left corner of the **Graph** page, click **Graph Type**.
4. On the **Graph Type List** page that appears on the right, select the graph type you want to view.

- Click ⋯ at the right of the graph type and click **View DDL** to view the GQL statement that defines the graph type.
- Alternatively, click the name of the graph type to enter the detail page:
- On the **Visual Builder** tab, you can view the graph type you selected as a property graph. You can click a node type or edge type to view more detailed information.
- On the **Node Type** tab, you can view all node types in the graph type. Besides, you can click the single-arrow icon to display all properties of a specific node type.
- On the **Edge Type** tab, you can view all edge types in the graph type. Besides, you can click click the single-arrow icon to display all properties of a specific edge type.
- On the **Label** tab, you can view all labels in the graph type by node type or edge type.

5.3 Manage graphs

This topic introduces how to manage graphs in NebulaGraph Cloud.

5.3.1 Prerequisites

A graph type is created. For more information, see Create a graph type.

5.3.2 Create a graph

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. In the upper-right corner, click the **Create Graph** button.
4. In the dialog box that appears, select a graph type from the drop-down list and enter a name for the graph.
5. Click **OK**. The newly created graph is displayed on the **Graph List** page.

5.3.3 Delete a graph

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. On the **Graph List** page that appears on the right, select the graph you want to delete.
4. Click ••• at the right of the graph and select **Delete**.
5. In the confirmation dialog box that appears, click **Delete**.

5.3.4 Manage indexes on graphs

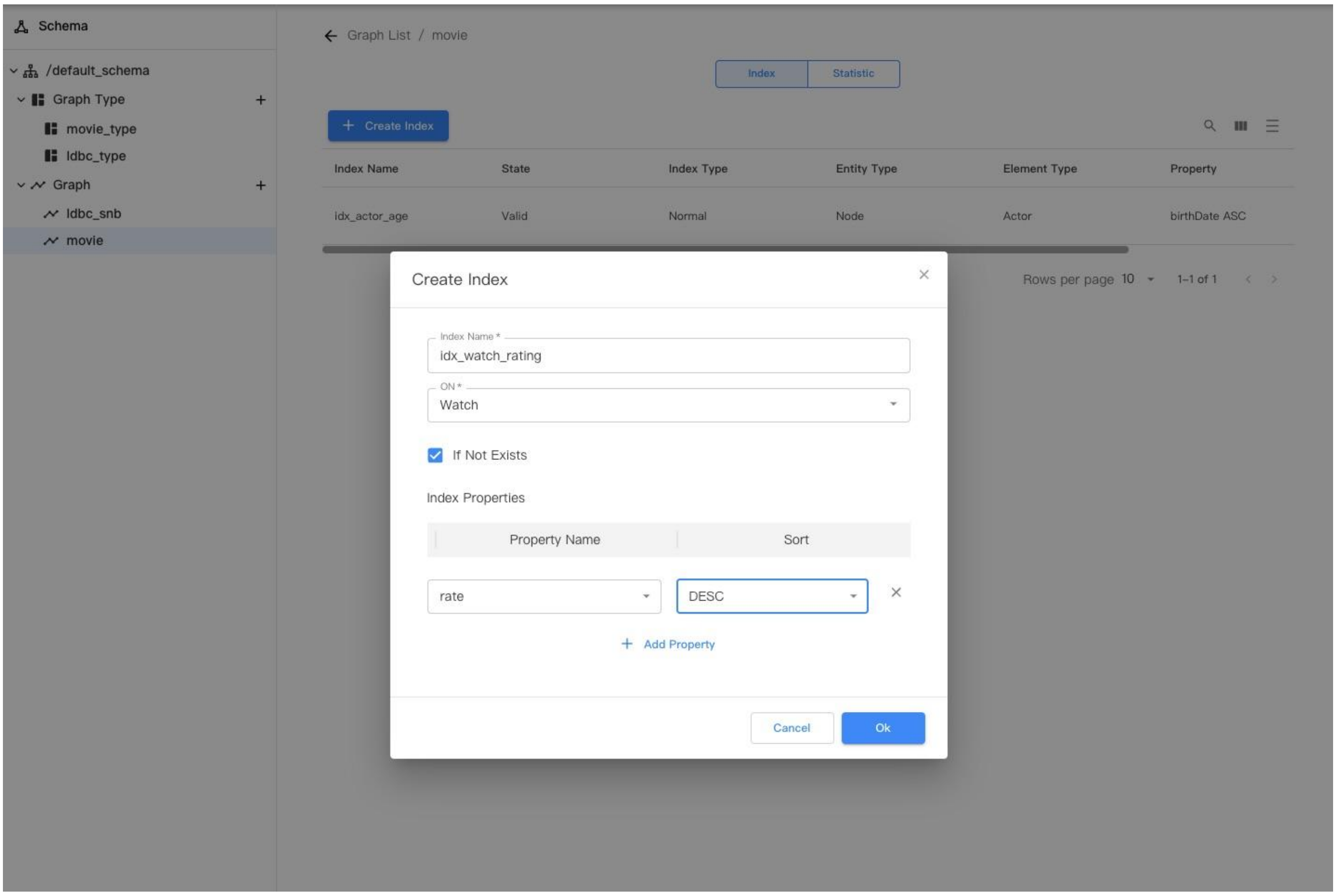
NebulaGraph Cloud allows you to manage indexes on graphs, including creating, deleting, and rebuilding indexes.

Create an index

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. On the **Graph List** page that appears on the right, select and click the target graph.
4. On the **Index** tab, click the **Create Index** button. For each index, specify the following information:

- **Index Name**: The name of the index.
- **On**: The node type or edge type on which the index is created.
- **Index Properties**: The list of properties based on which the index is created. Specify at least one property. For each property, specify the following information:
- **Property Name**: The name of the property.
- **Sort**: The sorting order of property values in the index. Valid values: **ASC** and **DESC**.

5. Click **OK** to create the index.



Index Name	State	Index Type	Entity Type	Element Type	Property
idx_actor_age	Valid	Normal	Node	Actor	birthDate ASC

Create Index

Index Name *: idx_watch_rating

ON *: Watch

☑ If Not Exists

Index Properties

Property Name	Sort
rate	DESC

For more information about the GQL syntax for creating indexes, see CREATE INDEX. (Initial password: Nebula.123)

Delete an index

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. On the **Graph List** page that appears on the right, select and click the target graph.
4. On the **Index** tab, select the index you want to delete.
5. Click the delete icon at the right of the index to delete it.

Rebuild an index

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. On the **Graph List** page that appears on the right, select and click the target graph.
4. On the **Index** tab, select the index you want to rebuild.
5. Click the **Repair** button to rebuild the index data.

5.3.5 View statistics about a graph

1. On the **Database** page, click the name of the database that you created to view the database details.
2. On the left-side navigation pane of the database details page, click **Graph** under **Data**.
3. On the **Graph List** page that appears on the right, select and click the target graph.
4. Click the **Statistic** tab.

5. To view the latest statistics, click the **Execute Stats** button to update the statistics about the graph. The statistics provide information about the number of nodes and edges in the graph.

6. Data import

This topic introduces how to import data into NebulaGraph Cloud.

6.1 Prerequisites

- A database instance is created. For more information, see Create a database instance.
- A graph type and a graph are created. For more information, see Graph management.

6.2 Create an import task

On the left-side navigation pane of the database details page, click **Import** > **Import Data** to create an import task, which includes uploading files, configuring files, and mapping data.

6.2.1 Step 1. Select a graph and upload files

1. In the upper-left corner of the **Import** page, select a graph from the **Load to Graph** drop-down list for data import.
2. On the **Upload files** tab, choose one of the following data sources to upload data, then click **Next**.

- **Local file**: Upload CSV files from your local machine. The size of each file must be less than 400 MB.
- **Amazon S3**: Click **+Add**, enter your Amazon S3 URL and click **Connect**. You can add up to 10 Amazon S3 URLs. Only CSV files are supported.

6.2.2 (Optional) Step 2. Configure files

If your CSV files do not contain headers, click **Next** to skip this step.

1. On the **Configure files** tab, check the uploaded data files.
2. In the upper-right corner of each file, click and select the **With header** checkbox to exclude headers from mapping.
3. Click **Next**.

6.2.3 Step 3. Map data

1. On the **Mapping** tab, specify the following items:

- **Type**: Select a node type or edge type you created before.
- **Select a file**: Select an uploaded file.
- **Map from file** or **Direction**: Map data from specified columns to graph type properties using the following methods.
- To map data individually, select each column name from the file.
- To map data in batches, click **Quick Mapping** on the right side. If the column name matches the property name, the system maps them automatically. If they do not match, the system maps them sequentially.

2. In the bottom-left corner, click **View DML** to verify mappings.
3. In the upper-right corner, click **Import** and check the mapping information on the **Check & Run** page.
4. Click **Confirm** to start the import task and view the task status on the **Import** page.

For a detailed example about mapping, see Best practice for mapping.

6.3 Manage import tasks

1. On the left-side navigation pane of the database details page, click **Import**.
2. On the **Import** page, you can view and manage the import tasks. The task list includes:

- ID: The import task ID
- Graph: The graph where the data is imported
- Status: The status of the import task, including **Running**, **Completed**, and **Failed**. If errors appear, click **Logs** in this column to troubleshoot.
- Size: The size of the imported data
- Loaded lines: The number of lines loaded from files
- Created at: The time when the import task was created
- Duration: The time taken to complete the import task
- Actions: To delete an import task, click **Delete** in this column.

6.4 Best practice for mapping

To show you a best practice for mapping in NebulaGraph Cloud, we take the following local files and graph type of the demo dataset as an example.

Demo file to be imported	Node/edge type	Property
`sub_user.csv` : Contains the information about users.	User	`id`
`sub_movie.csv` : Contains the information about movies.	Movie	`id` and `name`
`sub_user_watched_movies.csv` : Contains the information about the relationship between users and movies.	Watch	`rate`

Before you start, make sure:

- Demo files are downloaded. You can click the above file names to download files directly.
- The node types `User` and `Movie`, the edge type `Watch`, and the graph `movie` are created according to the demo dataset. For more information, see Create a graph type and Create a graph.
- Step 1 and Step 2 are completed.

Follow the steps below to map the data in `sub_user.csv`, `sub_movie.csv`, and `sub_user_watched_movies.csv` to the node types `User` and `Movie`, and the edge type `Watch`, respectively.

1. In the upper-left corner of the **Import** page, select the graph `movie` from the **Load to Graph** drop-down list.
2. On the **Mapping** tab, specify the items according to the table:

Type	Select a file	Direction	Map from file
User	sub_use.csv	NA	Property: id ⬅ Column name: user_id(187)
Movie	sub_movie.csv	NA	Property: id ⬅ Column name: movie_id(4) Property: name ⬅ Column name: name(Estado De Sitio)
Watch	sub_user_watched_movies.csv	Property: id ⬅ Column name: user_id(187) Property: name ⬅ Column name: movie_id(1)	Property: rate ⬅ Column name: rate(3.5)

Note

The first row of imported data is displayed in the parentheses `()` next to the column name.

3. Click **Import** once you have mapped all the data.

7. Data exploration

7.1 Data exploration

NebulaGraph Explorer is a visualization tool embedded in NebulaGraph Cloud, designed for exploring relationships and discovering insights in data hosted with NebulaGraph. It provides an easy-to-use interface and various features for data visualization and exploration.

7.1.1 Features

NebulaGraph Explorer provides the following features to help you explore your data:

- Interactive exploration: You can select multiple nodes to find their neighbors at one click, or explore the paths among these nodes based on different edge types.
- Data querying and filtering: You can query nodes, paths, and subgraphs based on specific conditions and apply various filters to customize the visualization.
- Rich visualization: You can customize the visualization style with different layouts, node and edge styles, and color schemes.
- Data storage and export: You can save the exploration results as images or export them as JSON or CSV files for further analysis.

7.1.2 Scenarios

NebulaGraph Explorer is useful in the following scenarios:

- You have no coding experience and want to explore the relationships in your data visually.
- You want to quickly understand the structure of your data, find the underlying patterns, and discover insights.

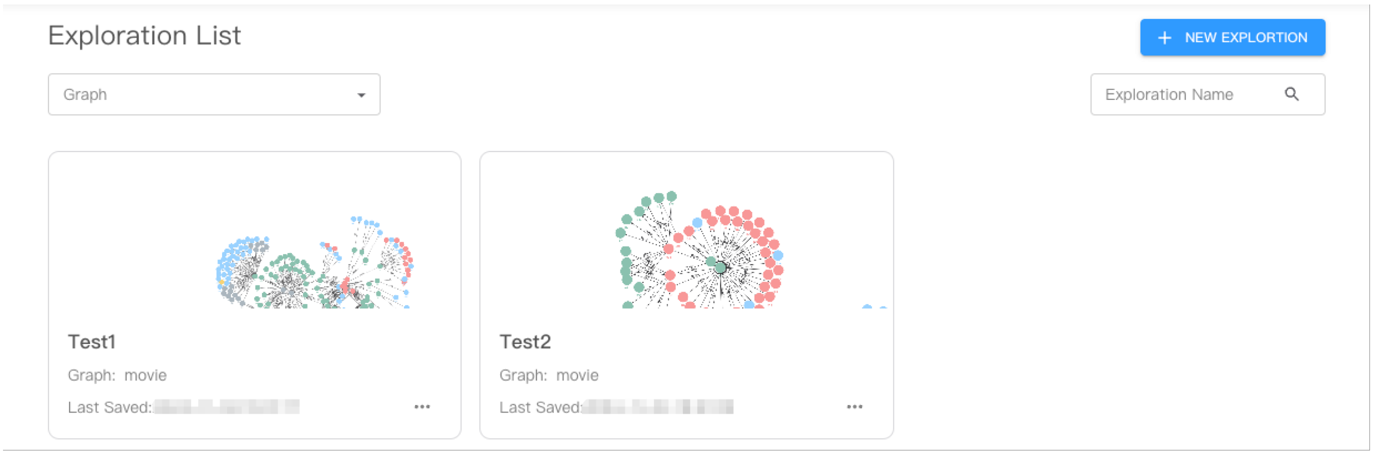
7.1.3 Related topics

For more information about data exploration, see the topics below.

- Manage explorations: Create, delete, and rename explorations.
- Explore subgraphs: Query nodes, explore relations, and query paths.
- Manage canvas data: Manage the data displayed on the canvas, including filtering, viewing, saving, exporting, and zooming.

7.2 Manage explorations

An exploration in NebulaGraph Explorer is a canvas where you can query and visualize data from a NebulaGraph graph. You can create, delete, and rename explorations in NebulaGraph Explorer. This topic introduces how to manage explorations in NebulaGraph Explorer.



Exploration List

Test1

Graph: movie

Last Saved:[redacted]

Test2

Graph: movie

Last Saved:[redacted]

7.2.1 Prerequisites

A graph is created. For more information, see Manage graphs or CREATE GRAPH. (Initial password: Nebula.123)

7.2.2 Create an exploration

1. In the top navigation bar, click the **EXPLORATION** tab.
2. In the upper right corner, click **+ NEW EXPLORATION**.
3. In the dialog box that appears, enter a name for the exploration and select the graph that you want to explore.
4. Click **Create**.

7.2.3 Delete an exploration

1. In the top navigation bar, click the **EXPLORATION** tab.
2. On the **Exploration List** page, select the exploration that you want to delete.
3. Click ••• in the lower right corner and then click **Delete Exploration**.
4. In the confirmation dialog box that appears, click **Confirm**.

7.2.4 Rename an exploration

1. In the top navigation bar, click the **EXPLORATION** tab.
2. On the **Exploration List** page, select the exploration that you want to rename.
3. Click ••• in the lower right corner and then click **Edit**.
4. In the dialog box that appears, enter a new name for the exploration and then click **Confirm**.

7.3 Explore subgraphs

This topic introduces how to explore subgraphs in NebulaGraph Explorer.

7.3.1 Prerequisites

An exploration is created. For more information, see Manage explorations.

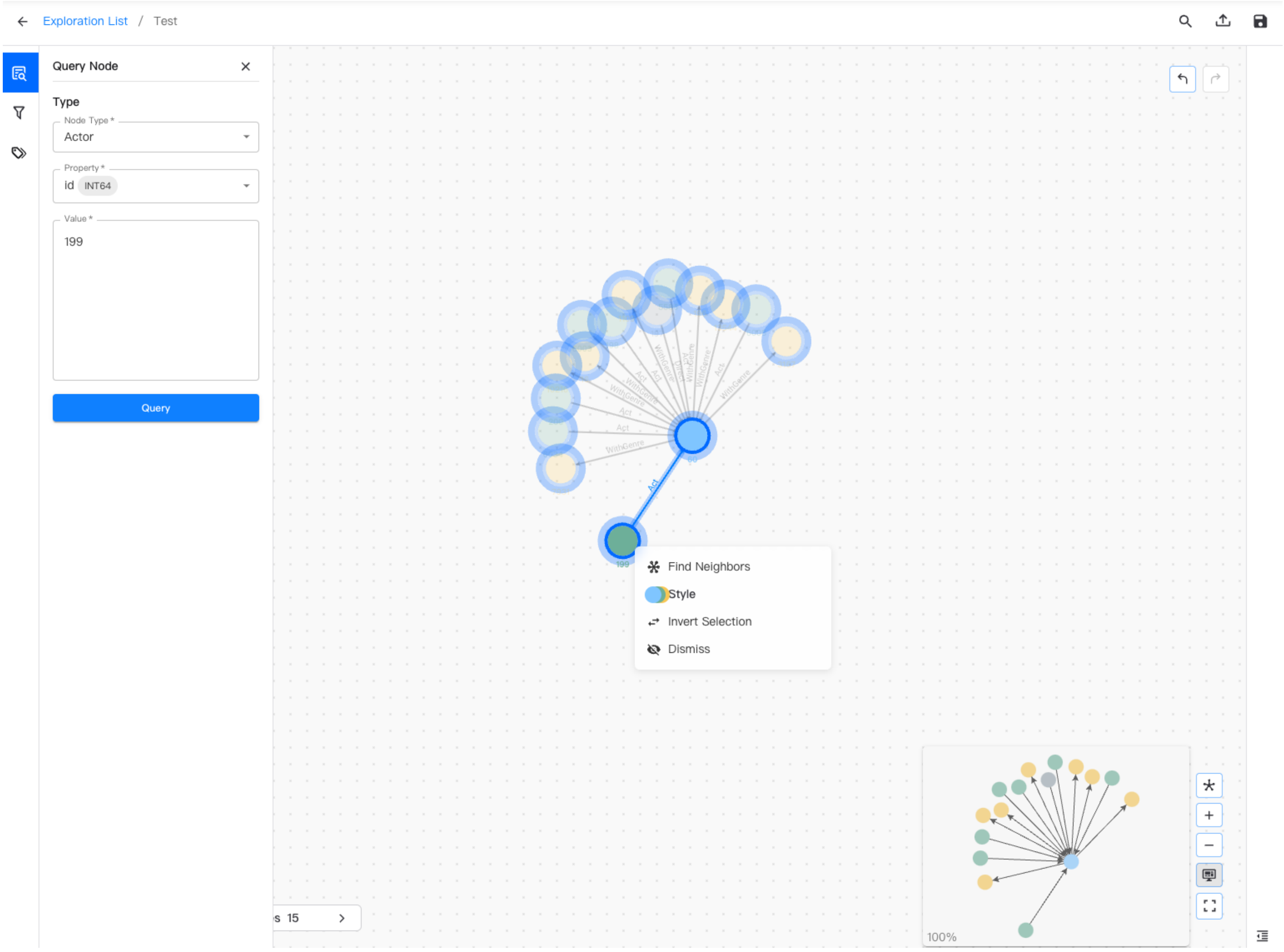
7.3.2 Query nodes

1. In the top navigation bar, click the **EXPLORATION** tab.
2. On the **Exploration List** page, click the exploration that you want to use.
3. On the left-side pane, click the **Graph Query** icon.
4. On the expanded pane that appears, click **Query Nodes** and specify the following information:

- **Node Type**: Select the type of nodes that you want to query.
- **Property**: Select the property of the node type.
- **Value**: Enter the value of the property. To query multiple nodes, specify multiple values separated by newlines.

5. Click **Query**. The nodes that meet the query conditions are selected in highlight on the canvas.
6. To explore further, select one or more nodes on the canvas and right-click to perform the following operations:

- **Find Neighbors**: Find the neighbors of the selected nodes.
- **Style**: Change the style of the selected nodes.
- **Invert Selection**: Invert the selection of the nodes.
- **Dismiss**: Dismiss the selected nodes so that they are grayed out on the canvas.



7.3.3 Explore relations

1. In the top navigation bar, click the **EXPLORATION** tab.

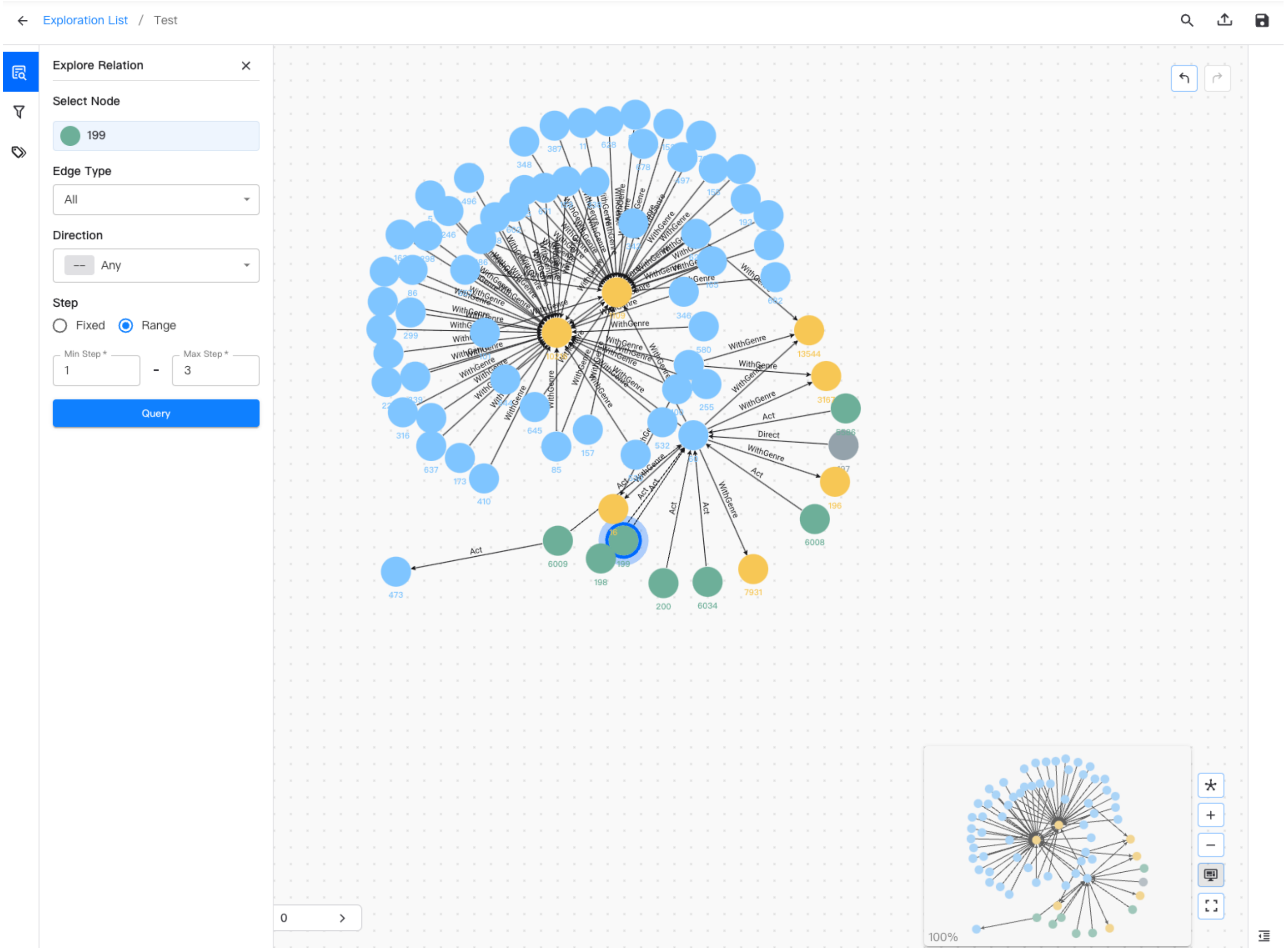
2. On the **Exploration List** page, click the exploration that you want to use.

3. On the left-side pane, click the **Graph Query** icon.

4. On the expanded pane that appears, click **Explore Relation** and specify the following information:

- **Select Node**: Select one or more nodes that you want to explore on the canvas.
- **Edge Type**: Select the types of edges that connect the nodes from the drop-down list.
- **Direction**: Select the direction of the edges from the drop-down list.
- **Step**: Specify the number of steps to explore. The following options are available:
- **Fixed**: Specify a fixed number of steps.
- **Range**: Specify a range of steps.

5. Click **Query**. The nodes and edges that meet the query conditions are selected in highlight on the canvas.



Exploration List / Test

Explore Relation

Select Node

199

Edge Type

All

Direction

Any

Step

○ Fixed ● Range

Min Step * 1 - Max Step * 3

Query

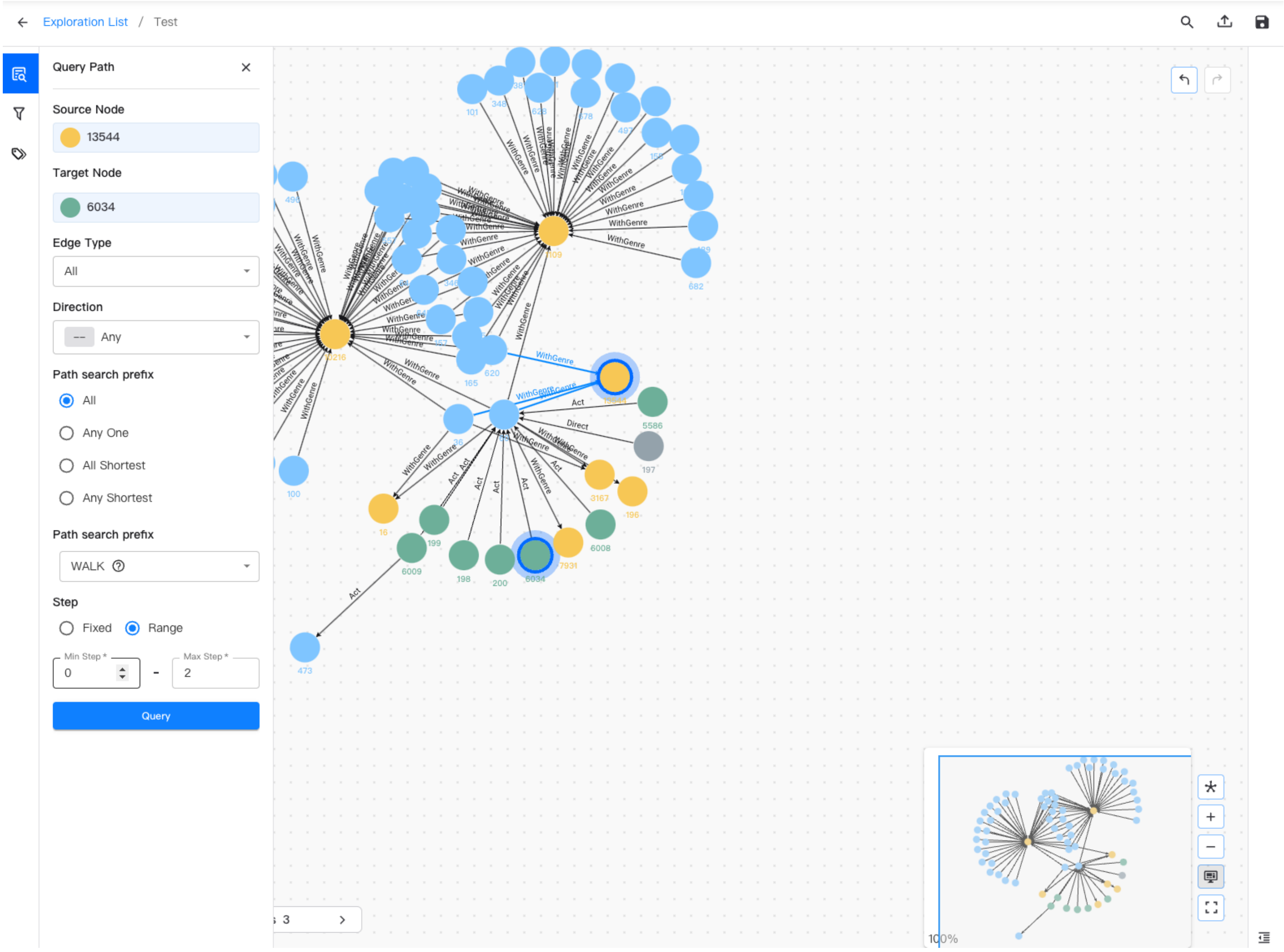
7.3.4 Query paths

1. In the top navigation bar, click the **EXPLORATION** tab.
2. On the **Exploration List** page, click the exploration that you want to use.
3. On the left-side pane, click the **Graph Query** icon.

4. On the expanded pane that appears, click **Query Path** and specify the following information:

- **Source Node**: Select the source node of the path on the canvas.
- **Destination Node**: Select the destination node of the path on the canvas.
- **Edge Type**: Select the types of edges that are allowed in the path from the drop-down list.
- **Direction**: Select the direction of the edges that are allowed in the path from the drop-down list.
- **Path Search Prefix**: Select the path search prefix. The following options are available:
- **All**: Query all paths that match the pattern.
- **Any One**: Query any path that matches the pattern.
- **All Shortest**: Query all the shortest paths that match the pattern.
- **Any Shortest**: Query any shortest path that matches the pattern.
- **Path Mode**: Select the path mode from the drop-down list. The following options are available:
- **WALK**: Applies no filtering to the paths. This is the default path mode.
- **TRAIL**: Filters out paths with repeated edges.
- **ACYCLIC**: Filters out paths with repeated nodes.
- **SIMPLE**: Filters out paths with repeated nodes unless these nodes are either the first or last nodes in the path.
- **Step**: Specify the number of steps to explore. The following options are available:
- **Fixed**: Specify a fixed number of steps.
- **Range**: Specify a range of steps.

5. Click **Query**. The paths that meet the query conditions are selected in highlight on the canvas.



Exploration List / Test

Query Path

Source Node

13544

Target Node

6034

Edge Type

All

Direction

Any

Path search prefix

● All

○ Any One

○ All Shortest

○ Any Shortest

Path search prefix

WALK

Step

○ Fixed ● Range

Min Step * 0 - Max Step * 2

Query

7.4 Manage canvas data

After querying and displaying graph data (nodes and edges) on the canvas, you can manage the data. This topic explains how to manage canvas data, including filtering, viewing, saving, exporting, and zooming.

7.4.1 Prerequisites

Before managing canvas data, you need to query data on the **Graph Query** pane and display it on the canvas. For more information, see Explore subgraphs.

7.4.2 Filter canvas data

You can filter the data on the canvas to better view specific data.

Filter conditionally

You can filter specific data to be displayed on the canvas based on filter conditions. The steps are as follows:

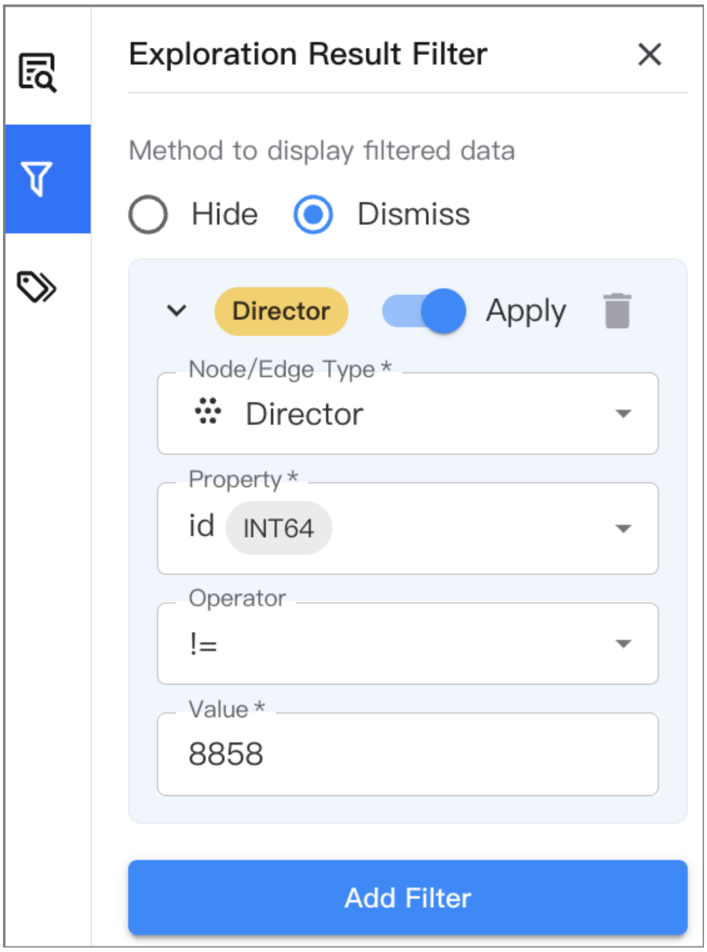
1. In the left navigation bar of the canvas, click the ▽ icon.
2. Set the display method for data that does not meet the conditions: **Hide** or **Dismiss**.

- **Hide**: Data that does not meet the conditions is hidden and not displayed on the canvas.
- **Dismiss**: Data that does not meet the conditions is grayed out.

3. In the **Exploration Result Filter** pane, set the filter conditions. Data that meets the conditions is displayed on the canvas, while data that does not meet the conditions is hidden or grayed out.

- **Node/Edge Type**: The node type or edge type to be displayed. Data of this type that does not meet the conditions is filtered out.
- **Property**: The property of the node or edge to be displayed.
- **Operator**: The operator for the filter condition.
- **Value**: The value of the property.

4. Click **Apply**.



Exploration Result Filter ×

Method to display filtered data

○ Hide ● Dismiss

Director — Apply

Node/Edge Type *: Director

Property *: id INT64

Operator: !=

Value *: 8858

Add Filter

To filter multiple node types or edge types, click **Add Filter** and repeat the above steps.

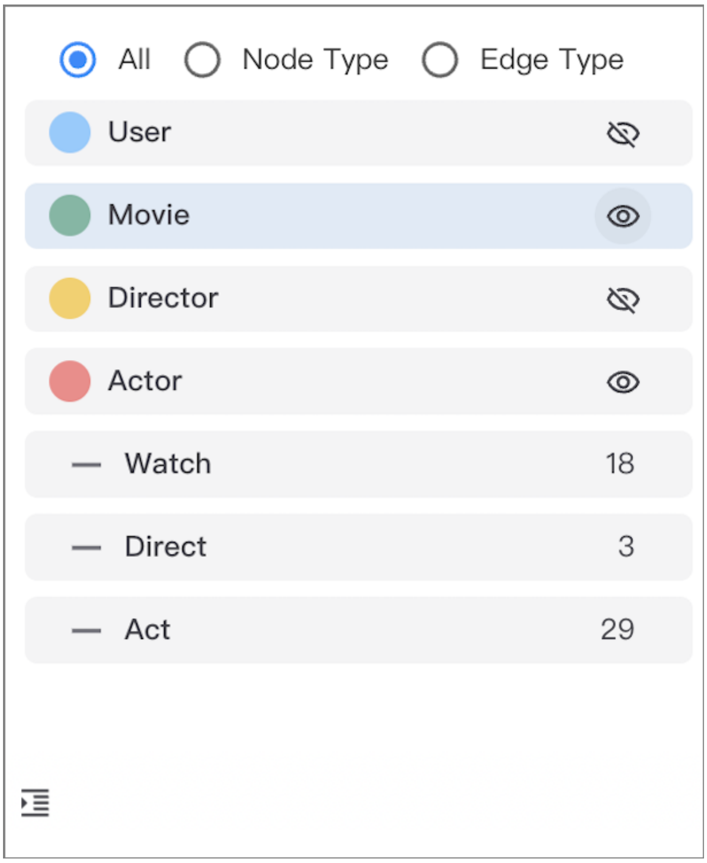
Filter by data type

In addition to filtering specific data based on conditions, you can also filter data of the same type based on the selected node type or edge type. The steps are as follows:

1. In the lower right navigation bar of the canvas, click ☰.
2. On the **All** tab, all node types and edge types on the canvas are displayed by default.

- You can click the **Node Type** tab to filter nodes by node type.
- You can click the **Edge Type** tab to filter edges by edge type.

3. Select the node type or edge type you want to hide, and click 👁. To show the data again, click ⊘.



Filter using the search box

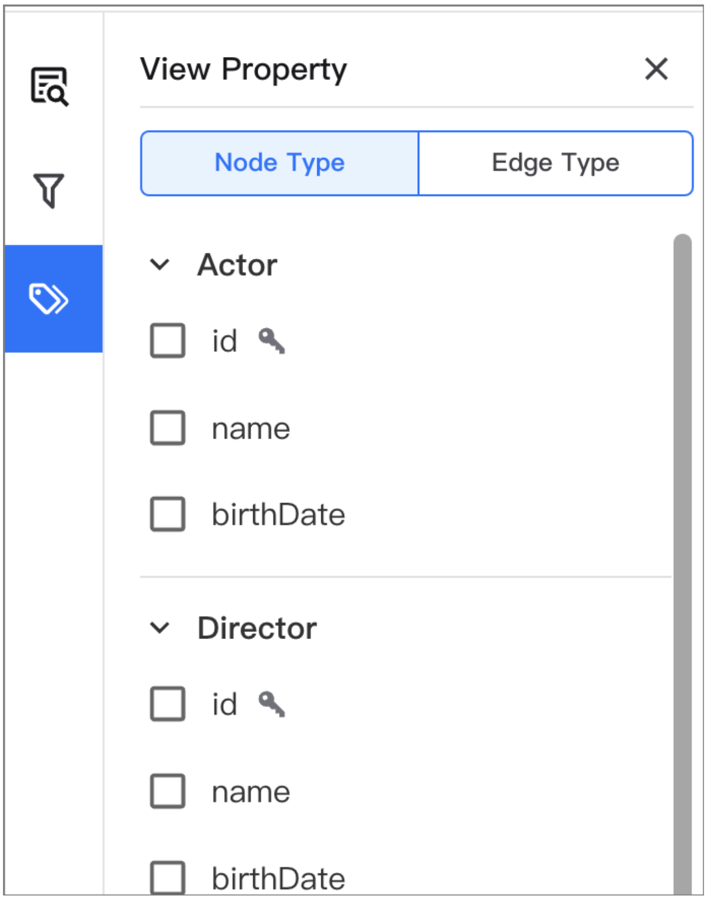
In the top navigation bar of the canvas, click 🔍, and enter the property value of the node or edge in the search box to filter the data to be displayed based on that value.

7.4.3 Display canvas data properties

You can configure whether to display the properties of nodes and edges on the canvas.

1. In the left navigation bar of the canvas, click the 🏷 icon.
2. Configure whether to display the properties of nodes and edges on the canvas.

- To display properties, check the property names.
- To hide properties, uncheck the property names.



View Property ✕

Node Type | Edge Type

Actor

□ id

□ name

□ birthDate

Director

□ id

□ name

□ birthDate

7.4.4 View canvas data information

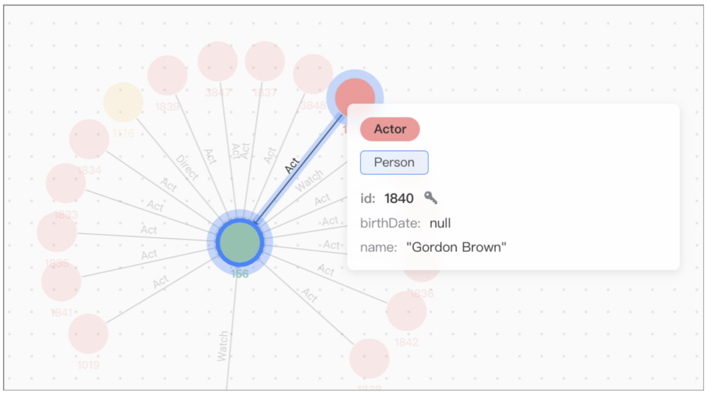
You can view information about the data on the canvas, including data volume and properties.

View canvas data statistics

In the lower right navigation bar of the canvas, click ☰ to view the number of nodes and edges on the canvas.

View canvas data properties

In the canvas, hover the mouse over the target node or edge to automatically display the data information, including data type, labels, properties, etc.



Actor

Person

id: 1840

birthDate: null

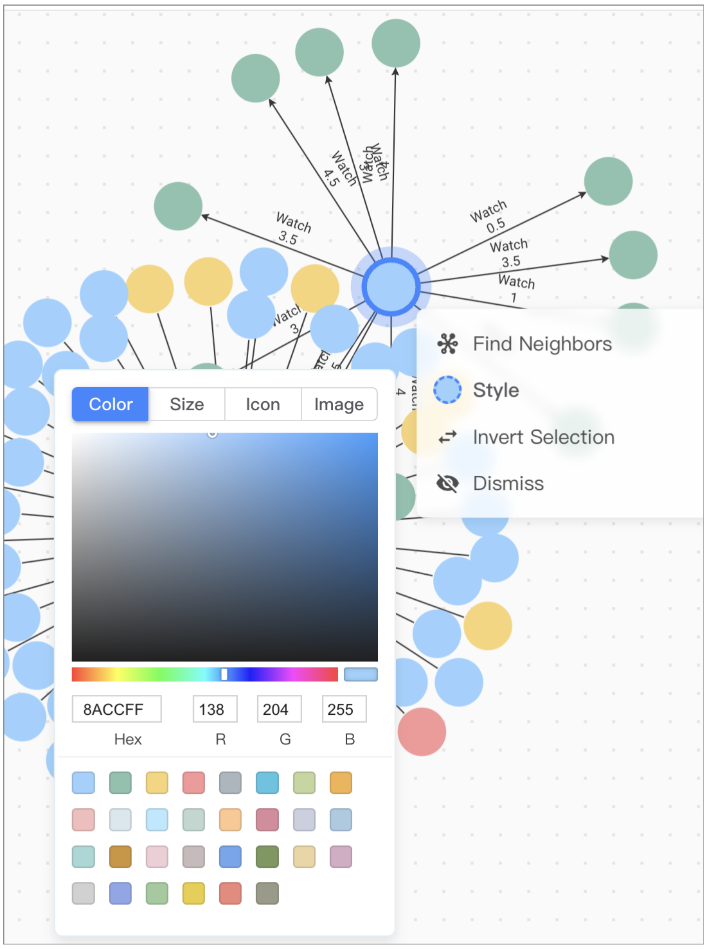
name: "Gordon Brown"

7.4.5 Set canvas data styles

Set style for a single node or edge

1. On the canvas, select the target node or edge and right-click.
2. Click **Style**.
3. In the style setting panel, set the style for the node or edge.

- For the node, you can set the color, size, icon, and image.
- For the edge, you can set the color and size.



8ACCFF Hex
138 R
204 G
255 B

Set style for multiple nodes or edges of the same node type or edge type

1. In the lower right navigation bar of the canvas, click the ☰ icon.
2. Select the target node type or edge type to configure the style for all nodes or edges of that type simultaneously.

- For nodes of the same type, you can set the color, size, icon, and image.
- For edges of the same type, you can set the color and size.

7.4.6 Set canvas data layouts

In the lower right navigation bar of the canvas, click ✱ to set the layout of the canvas data.

NebulaGraph supports 8 layout methods to display the relationships between data on the canvas.

Layout	Icon
Force	✱
Dagre	⋔
Dagre-LR	⊰
Circular	⬡
Grid	≶
Radial	✳
Concentric	⬠
Neural Network	<

7.4.7 Save canvas data

In the upper right corner of the canvas, click save to save the data to the exploration. The saved exploration can be viewed on the **Exploration List** page.

7.4.8 Export canvas data

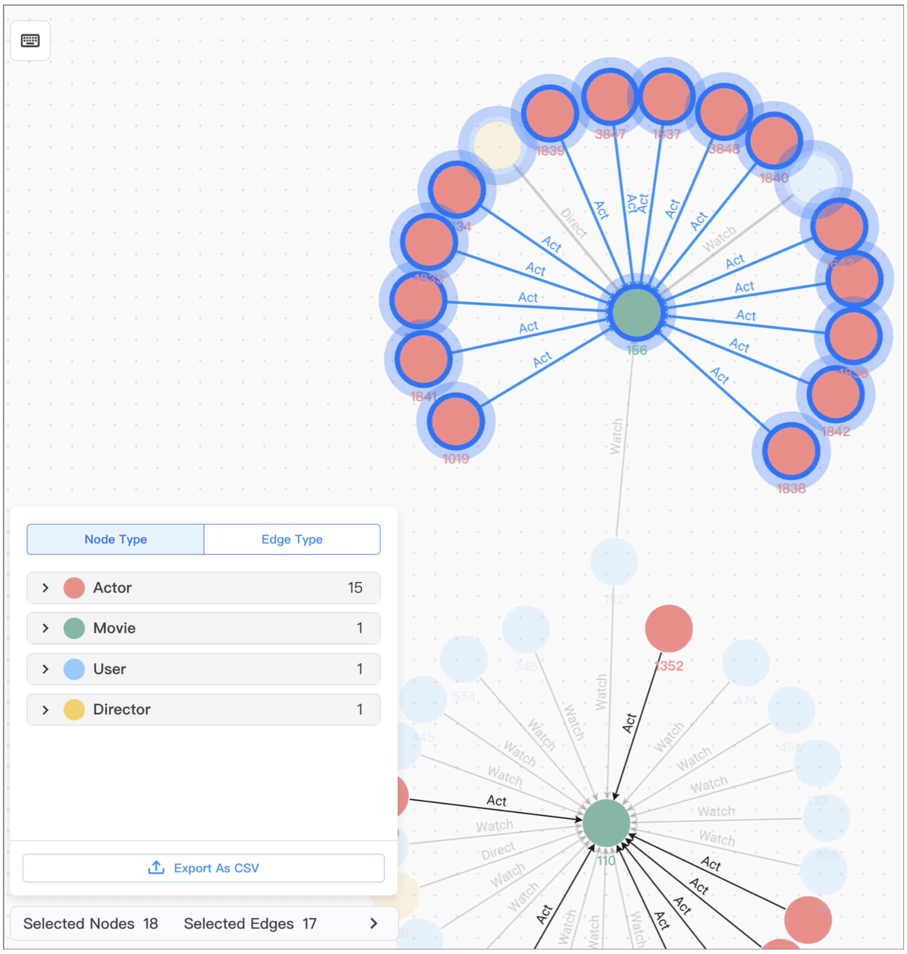
Export all canvas data

In the upper right corner of the canvas, click ⇪ to export the data on the canvas. The export formats include PNG, JSON, and CSV.

Export selected canvas data

In addition to exporting the entire canvas data, you can also export the selected nodes or edges. The export format is CSV.

1. Select the nodes or edges to be exported.
2. In the lower left corner of the canvas, click › .
3. Click **Export as CSV**.



7.4.9 Zoom canvas data

In the lower right corner of the canvas, you can zoom the canvas data in the following ways:

- Click + to zoom in on the canvas.
- Click − to zoom out on the canvas.
- Click 🖥 to move the canvas screen.
- Click ⛶ to display the canvas data in full screen and click ⤡ to exit full screen.

7.5 Canvas operations and shortcuts

This topic lists the basic operations that you can perform on a canvas and the shortcuts that you can use in NebulaGraph Explorer.

7.5.1 Basic operations

Operation	Description
Move the canvas	Hold down the left mouse button and drag the canvas.
Zoom in or out	Use the mouse wheel to zoom in or out.
Select a single node or edge	Left-click a node or an edge.
Select multiple nodes and edges	Hold Shift and left-click nodes and edges.
Batch selection	Hold down the right mouse button and drag to frame nodes and edges; or hold Shift, left-click, and then drag to frame nodes and edges.
Move selected nodes	Left-click and drag the selected nodes.

7.5.2 Shortcuts

Shortcut	Description
Shift + '-'	Zoom out
Shift + '+'	Zoom in
Ctrl/Cmd + 'Z'	Undo
Ctrl/Cmd + Shift + 'Z'	Redo
Ctrl/Cmd + 'A'	Select all nodes
Selected elements + Enter	Expand
Selected elements + 'Backspace'	Hide selected elements
Selected elements + Shift + 'Backspace'	Hide unselected elements

8. Data query

This topic introduces how to perform GQL queries and navigate around the query console in NebulaGraph Cloud.

8.1 Steps

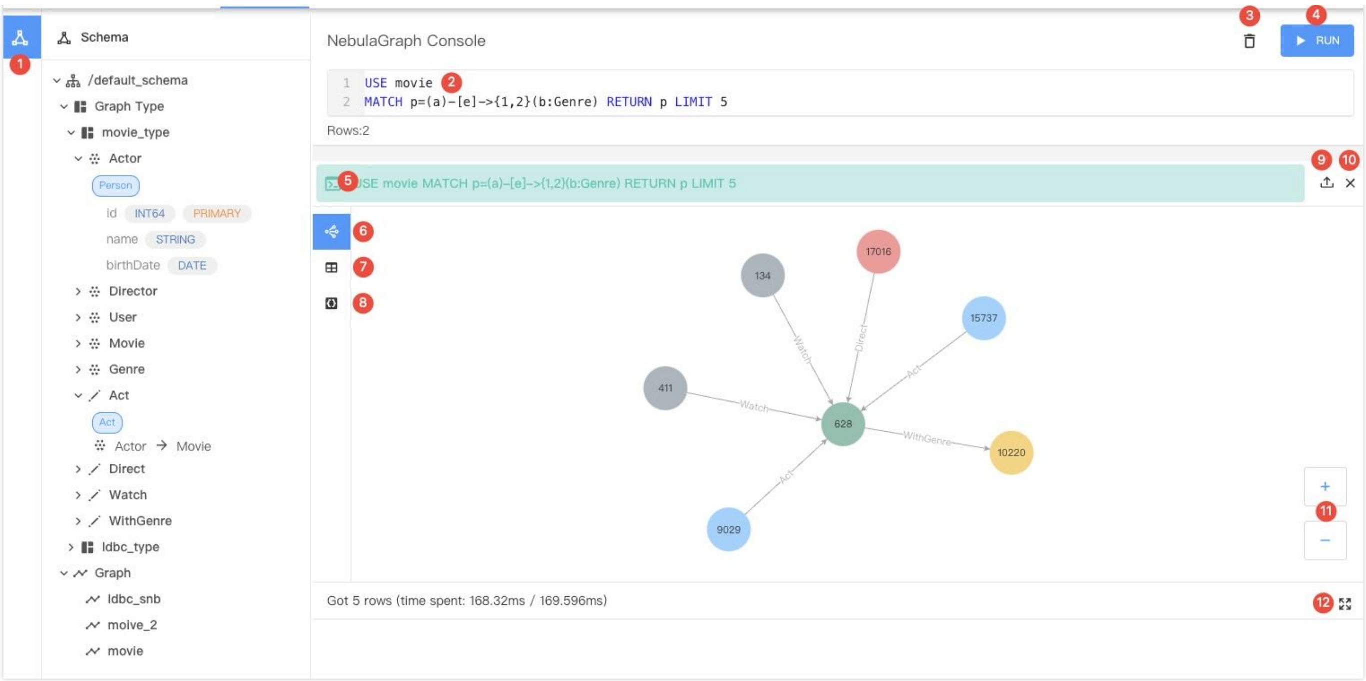
1. Sign up and log in to NebulaGraph Cloud. For more information, see Login.
2. On the **Database** page, click the name of the database that you created to view the database details.
3. On the left-side navigation pane of the database details page, click **Query**.
4. On the **Query** page, enter a GQL statement in the input area.

   For how to write GQL queries, see GQL guide.(Initial password: Nebula.123)

   For how to navigate around the console, see UI components.

8.2 UI components

The following figure shows the console UI.



```
USE movie
MATCH p=(a)-[e]->{1,2}(b:Genre) RETURN p LIMIT 5
```

Rows:2

USE movie MATCH p=(a)-[e]->{1,2}(b:Genre) RETURN p LIMIT 5

Got 5 rows (time spent: 168.32ms / 169.596ms)

The following table lists the UI components of the console.

No.	Component	Description
1	Schema	Display the home schema and all graph types and graphs in the schema.
2	Input area	The area where GQL statements are entered. The input is displayed in different colors based on the syntax.Auto-completion is supported.Comments must start with double slashes(`//`). Right-clicking a character string displays a context menu that provides options such as Change All Occurrences, Cut, Copy, and Command Palette.
3	Clear button	Clear the input area.
4	Run button	Execute the GQL statement in the input area. You can also press `Shift + Enter` to execute the statement.
5	Running status	Display the running status after statement execution. The statement is displayed in green if the execution is successful and in red if it fails.You can click a statement to paste it into the input area.
6	Graph window	Display the exeuction result as a property graph. You can click the nodes and edges in the graph to view their details.
7	Table window	Display the execution result in a tabular format. You can click the column header to sort the data.
8	Text window	Display the execution result in JSON format.
9	Export button	Export the execution result as a CSV file or a JSON file.
10	Close button	Close the execution result window.
11	Zoom in/out button	Zoom in or out the Graph window.
12	Full screen button	Display the execution result in full screen.

9. SDKs

9.1 Java SDK guide

This guide introduces how to set up the Java SDK to connect to NebulaGraph and perform simple queries.

9.1.1 Prerequisites

- JDK 8 is installed.
- The Java SDK installation package and NebulaGraph connection information are obtained from NebulaGraph Cloud. For more information, see Connect to your database.

9.1.2 Steps

1. Import the Java SDK into your Java project. For example, to import with Maven, add the following dependency to the `pom.xml` file:

```
<dependency>
  <groupId>com.vesoft</groupId>
  <artifactId>driver</artifactId>
  <version>5.0.0</version>
</dependency>
```

Note: After you reference the NebulaGraph SDK dependency in a project and package it as a JAR file, if you encounter the exception `java.lang.IllegalArgumentException: Address types of NameResolver 'unix' for '192.168.15.8:9669' not supported by transport` when running this JAR file, add the plugin `maven-shade-plugin` to your project's `pom.xml` file and configure the transformer as follows:

```
  <plugin>
              <groupId>org.apache.maven.plugins</groupId>
              <artifactId>maven-shade-plugin</artifactId>
              <version>3.4.1</version>
              <configuration>
                  <!-- put your configurations here -->
                  <filters>
                      <filter>
                          <artifact>*:*</artifact>
                      </filter>
                  </filters>
                  <transformers>
                      <transformer implementation="org.apache.maven.plugins.shade.resource.ServicesResourceTransformer">
                      </transformer>
                  </transformers>
              </configuration>
              <executions>
                  <execution>
                      <phase>package</phase>
                      <goals>
                          <goal>shade</goal>
                      </goals>
                  </execution>
              </executions>
          </plugin>
```

2. Select either of the following methods to connect to NebulaGraph and execute queries:

- If only one client needs to interact with NebulaGraph, you can use the client to establish a direct connection and execute queries.

```
NebulaClient client = null;
try {
    client = NebulaClient.builder("<host>:<port>", "<username>", "<password>")
            .withAuthOptions(Collections.emptyMap())
            .withConnectTimeoutMills(1000)
            .withRequestTimeoutMills(3000)
            .build();
    client.execute("USE nba MATCH (v:player) RETURN v.id, v.name, v.score, v.gender, v.rate");
} catch (Exception e) {
    throw e;
} finally {
    if (client != null) {
        client.close();
    }
}
```

- If multiple clients need to interact with NebulaGraph simultaneously, you can create a connection pool to manage a set of connections. You can use the pool to get a client and return it after queries are executed.

```
NebulaPool pool = null;
try {
    pool = NebulaPool
            .builder("<host>:<port>", "<username>", "<password>")
            .withMaxClientSize(10)
            .withMinClientSize(1)
            .withConnectTimeoutMills(1000)
            .withRequestTimeoutMills(30000)
            .withBlockWhenExhausted(true)
            .withMaxWaitMills(Long.MAX_VALUE)
            .build();
    NebulaClient client = pool.getClient();
    client.execute("USE nba MATCH (v:player) RETURN v.id, v.name, v.score, v.gender, v.rate");
    pool.returnClient(client);
} catch (Exception e) {
    throw e;
} finally {
    if (pool != null) {
        pool.close();
    }
}
```

Replace `<host>`, `<port>`, `<username>`, and `<password>` with the actual connection information obtained from NebulaGraph Cloud.

9.1.3 More information

- Java SDK API reference

9.2 Go SDK guide

This guide introduces how to set up the Go SDK to connect to NebulaGraph and perform simple queries.

9.2.1 Prerequisites

The Go SDK installation package and NebulaGraph connection information are obtained from NebulaGraph Cloud. For more information, see Connect to your database.

9.2.2 Steps

1. Modify the Go module dependencies to use the local version of the Go SDK. Example:

```
workspace=/app/myapp
tar zxvf nebula-golang-5.0.0.tar.gz -o /tmp/golang
# In your project, run go mod edit
cd ${workspace}
go mod edit -replace github.com/vesoft-inc/nebula-ng-tools/golang=/tmp/golang
```

For more information, see go.mod file reference.

2. Use the SDK to connect to NebulaGraph and perform queries as follows:

```
package main

import (
 "fmt"

 nebula "github.com/vesoft-inc/nebula-ng-tools/golang"
)

const (
 address  = "<host>:<port>"
 username = "<username>"
 password = "<password>"
)

// Initialize logger
var log = nebula.DefaultLogger

func basicClient() {
 client, err := nebula.NewNebulaClient(address, username, password)
 if err != nil {
  log.Error(err.Error())
  return
}
resp, err := client.Execute("return 1 as a")
if err != nil {
 log.Error(err.Error())
 return
}
log.Info(fmt.Sprintf("columns: %v", resp.Columns()))
for resp.HasNext() {
 row, err := resp.Next()
 if err != nil {
  log.Error(err.Error())
  return
 }
 v1, err := row.GetValueByIndex(0)
 if err != nil {
  log.Error(err.Error())
  return
 }
 log.Info(v1.String())
 v2, err := row.GetValueByName("a")
  if err != nil {
   log.Error(err.Error())
   return
  }
  log.Info(v2.String())
 }
}

func basicPool() {
 pool, err := nebula.NewNebulaPool(address, username, password)
 if err != nil {
  log.Error(err.Error())
  return
 }
 client, err := pool.GetClient()
 if err != nil {
  log.Error(err.Error())
  return
 }
 resp, err := client.Execute("return 1 as a")
 if err != nil {
  log.Error(err.Error())
  return
 }
 log.Info(fmt.Sprintf("columns: %v", resp.Columns()))
 for resp.HasNext() {
  row, err := resp.Next()
  if err != nil {
   log.Error(err.Error())
   return
  }
  v1, err := row.GetValueByIndex(0)
  if err != nil {
   log.Error(err.Error())
   return
  }
  log.Info(v1.String())
  v2, err := row.GetValueByName("a")
  if err != nil {
   log.Error(err.Error())
   return
  }
  log.Info(v2.String())
 }
}

func main() {
 basicClient()
 basicPool()
}
```

Replace `<host>`, `<port>`, `<username>`, and `<password>` with the actual connection information obtained from NebulaGraph Cloud.

9.3 Python SDK guide

This topic introduces how to set up the Python SDK to connect to NebulaGraph and perform simple queries.

9.3.1 Prerequisites

The Python SDK installation package and NebulaGraph connection information are obtained from NebulaGraph Cloud. For more information, see Connect to your database.

9.3.2 Steps

1. Install the Python SDK. Example:

```
pip install nebulagraph_python-5.0.0-py3-none-any.whl
```

2. Use the SDK to connect to NebulaGraph and perform queries as follows:

```
from nebulagraph_python.client import NebulaClient

# Create client
client = NebulaClient(
    hosts=["<host>:<port>"],
    username="<username>",
    password="<password>",
)

query = """
    RETURN 1 AS a, 2 AS b
"""
# Execute query
result = client.execute(query)
```

Replace `<host>`, `<port>`, `<username>`, and `<password>` with the actual connection information obtained from NebulaGraph Cloud.

3. To inspect the query result, refer to the following methods:

- Print the result in table style

```
result.print()
```

Return:

```
┌─────┬─────┐
│     │     │
│  a  │  b  │
│     │     │
├─────┼─────┤
│     │     │
│  1  │  2  │
│     │     │
└─────┴─────┘

Summary
├── Rows: 1
└── Latency: 1450µs
```

- Retrieve specific values from the query set

```
# Get one row
row = result.one()

# Get one value
cell = row["a"].cast_primitive()

# Print its value
print(cell, type(cell))
```

Return:

```
1 <class 'int'>
```

- Retrieve primitive values from the result set

```
print(result.as_primitive_by_column())
print(list(result.as_primitive_by_row()))
```

Return:

```
{'a': [1], 'b': [2]}
[{'a': 1, 'b': 2}]
```

- Retrieve a Pandas DataFrame from the query set

```
result = client.execute(query)
df = result.as_pandas_df()
```

Return:

```
   a  b
0  1  2
```

9.3.3 More information

- Python SDK API reference

10. NebulaGraph Cloud release notes

10.1 20250522

- Features:
- Implemented a notification system to ensure seamless account management:

  If your AWS account or subscription encounters an issue, you will receive email notifications prompting you to resolve it within a 14-day grace period. During this period, your database instances and data will be suspended, and you cannot create new instances until the issue is resolved. If the issue remains unresolved after the grace period, your database instances and data will be deleted.

- Enhancements:
- Added metrics to the **Infrastructure** and **Service** pane on the **Metrics** page. You can monitor infrastructure-level metrics for both storage and query nodes. For more information, see Monitor database information and metrics.
- Removed the schema drop-down list from the **Graph** page to streamline graph management within a single schema, enhancing usability.

10.2 20250427

- Features:
- Supported multiple plan tiers: Standard and Professional. You can choose the following plan according to your needs.
- The **Standard** plan: Ideal for users who need to quickly set up a database for development, learning, experimentation, or prototyping.
- The **Professional** plan: Tailored to users who require a production-ready database with a storage-compute separation architecture, offering high availability, scalability, and better performance.

  For more information, see Upgrade to a paid plan.

- Enhancements:
- Updated the disk storage pricing from a fixed model to a tiered model. For more information, see Disk storage specification.

10.3 20250415

- Enhancements:
- Supported 6-hour grace period after unsubscribing from the paid plan. During this period, your database instances and data are retained, and you can re-subscribe to the paid plan to continue using the service. Otherwise, the instances and data will be deleted after the grace period.

10.4 20250402

- Features:
- Added a "Get started" guide at the top of the overview page to help you quickly master NebulaGraph Cloud.
- Added a "Feedback" button in the upper-right corner of the NebulaGraph Cloud interface for you to share suggestions with us.

- Enhancements:
- Supported selecting the `us-west-2` region for database creation.
- Performance
- Improved import efficiency by leveraging optimized GQL statements during the import process.
- Introduced a "Check & Run" step for you to verify mappings before starting the import task.

- Usability
- Enhanced the user experience with seamless navigation to the "Query" or "Graph" page.

10.5 20250313

- Features:
- Supported database instance management, including suspending, resuming, resizing instances, and increasing their storage capacity.
- Integrated a visualized database management tool, NebulaGraph Explorer, to help you explore data insights in an intuitive way.

- Enhancements:
- Performance
- Upgraded the NebulaGraph database to version 5.0.2, compatible with the NebulaGraph Cloud of version 20250313, for better performance and stability.

10.6 20250228

- Features:
- Added support for a 14-day free trial for new users without requiring any payment information. For more information, see Try NebulaGraph Cloud for free.

- Enhancements:
- Optimized the design of the "Help" pop-up window on the NebulaGraph Cloud user interface.
- Added a "Learn GQL" quick-access button to the upper-right corner of the NebulaGraph Cloud user interface.
- Optimized the user experience design on the database creation page.
- Highlighted the database status on the database list page.

10.7 20250214

- Enhancements:
- Supported setting metrics refresh intervals for real-time monitoring, with options ranging from 5 seconds to 5 minutes.
- Added the terms of service and privacy policy to the login page.

https://www.nebula-graph.io/docs/cloud/master