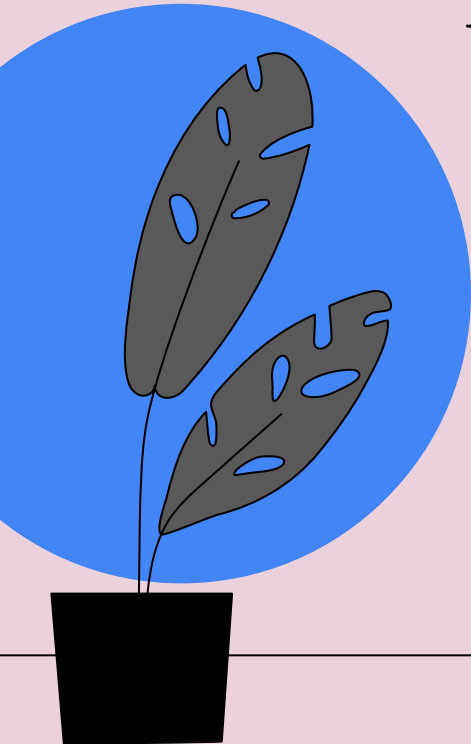


Presentación

Nebula Graph (Distributed Graph database)



INTEGRANTES:

- ☐ *Clavijo Alisson*
- ☐ *Iza Adrian*

Índice



- Introducción
- Objetivos
- Características de Nebula Graph
- .Arquitectura distribuida
- Modelo NoSQL
- Implementación
- Conclusiones
- Recomendaciones
- Bibliografía

y



Introducción

NebulaGrap es una base de datos de grafos distribuida de alta escalabilidad, diseñada para gestionar grandes volúmenes de datos no estructurados. En este trabajo se presenta una visión general de NebulaGrap y sus características clave.



Objetivos

- Investigar y documentar todo sobre Nebula Graph.
- Realizar 1 ejemplo práctico.

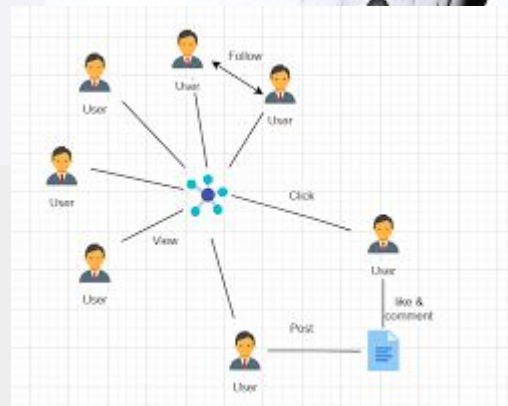


Características de NebulaGraph

1 Escalabilidad horizontal 



2 Modelo NoSQL 



3 Soporte para lenguajes de consulta standard

nebula-contrib/**nebula-node**

Nebula Graph Client for Node.js



NebulaGraph

6

Contributors

7

Used by

21

Stars

9

Forks



4 Arquitectura distribuida

**Nebula Graph
Architecture**

distributed architecture

A Bird's Eye View

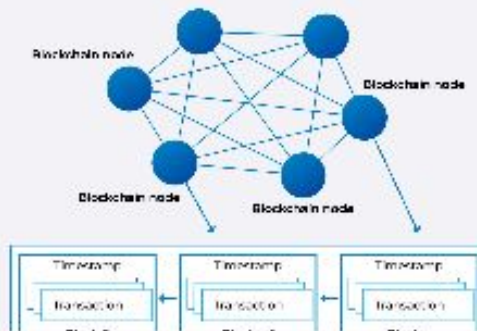


Arquitectura distribuida



Cluster de servidores

NebulaGraph se puede desplegar en un cluster de servidores distribuidos para manejar grandes volúmenes de datos.



Redundancia de datos

La arquitectura distribuida asegura que los datos estén redundados y disponibles en caso de fallo de los servidores.



Escalabilidad

La arquitectura distribuida permite escalar la base de datos agregando más servidores al cluster a medida que se necesite.

Modelo NoSQL



Manejo eficiente de datos no estructurados

NebulaGrap utiliza un modelo NoSQL que permite manejar rápidamente grandes volúmenes de datos no estructurados, como grafos y redes.

Alto rendimiento en búsquedas complejas

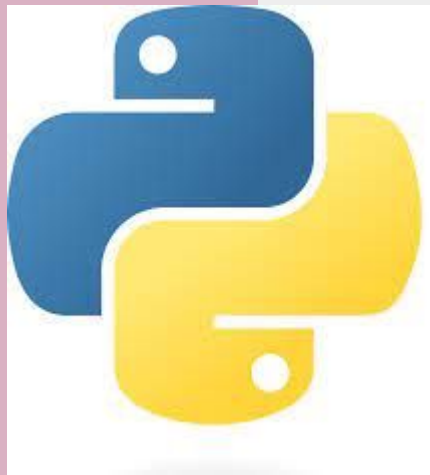
El modelo NoSQL de NebulaGrap permite realizar búsquedas complejas y analíticas de datos, lo que brinda un alto rendimiento y una mayor eficiencia.

Mejora la escalabilidad y la disponibilidad

El modelo NoSQL de NebulaGrap permite la escalabilidad y la redundancia de datos, lo que mejora la disponibilidad de la base de datos.



Implementación



Docker Desktop

Upgrade plan

Search for images, containers, volumes, extensions and more...

Ctrl+K

baio117

Containers

Images

Volumes

Dev Environments BETA

Docker Scout EARLY ACCESS

Learning Center

Extensions

NebulaGraph

+ Add Extensions

NebulaGraph

Studio in Browser

Legacy Studio

Console

Stars 9.3k

Home

Resources

Get Started

Docs

Console

NebulaGraph AI

1 Login

2 Activate StorageD

3 Load Dataset

4 Query NebulaGraph

5 Finish

Step 1 - Login to the Studio.

First, open the Studio in your browser:

Studio in Browser

Second, fill in access info & credentials, then click **Connect**:

Field	Value
Host Address	graphd:9669
User Name	root
Password	nebula

NebulaGraph Studio

Schema Import Console Schema diff

succeed

Always show the welcome page

X

Welcome to **NebulaGraph Studio**

Functions introduction

RAM 3.56 GB

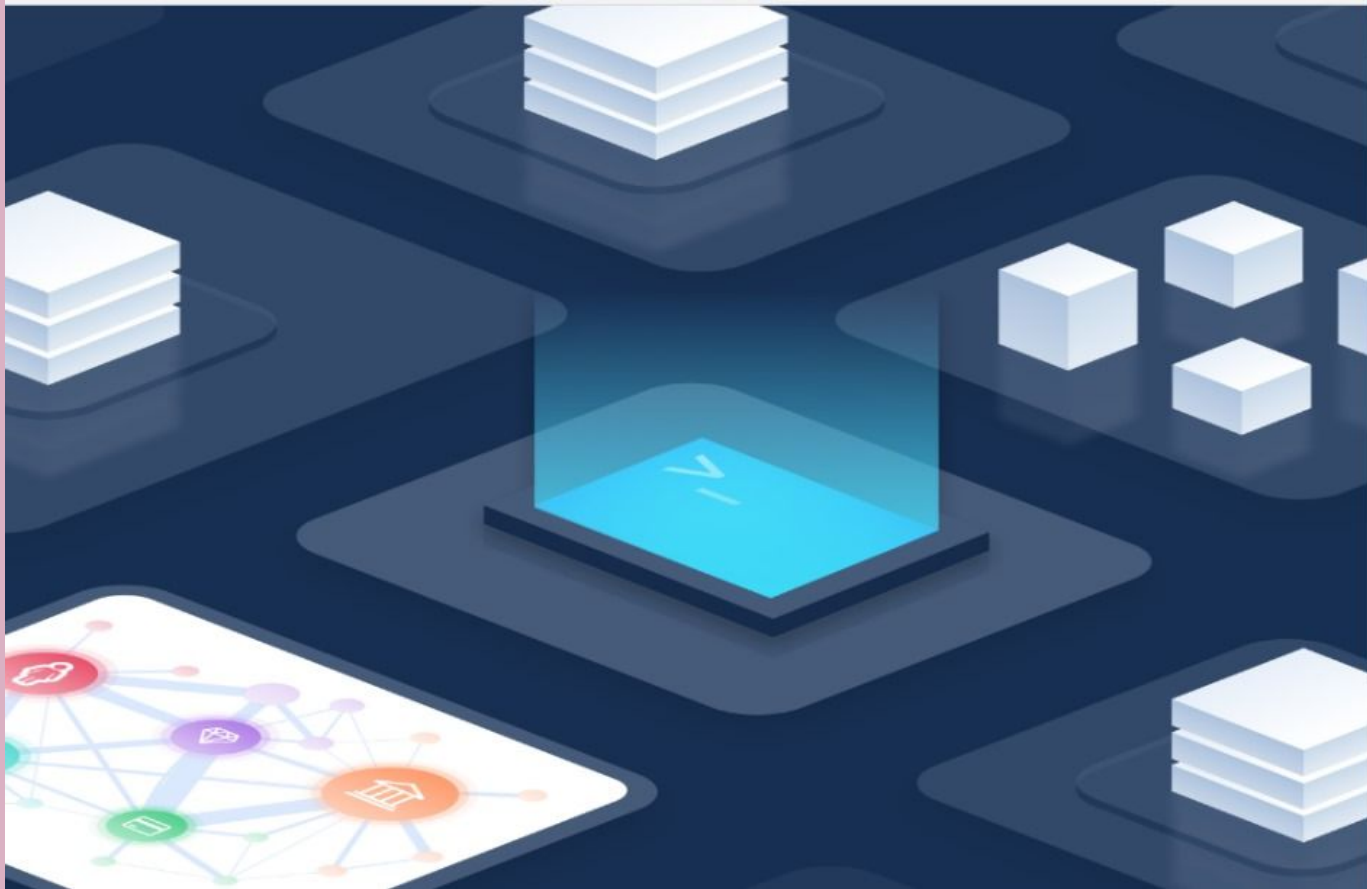
CPU 1.54%

Connected to Hub

v4.21.1

Activar Windows
Ve a Configuración para activar Windows.

10



NebulaGraph Studio

Connect to NebulaGraph

Don't know the address? [Docs->](#)

Graphd IP address : 9669

Username

Password



Welcome to NebulaGraph Studio

Functions introduction



Schema

You can use the Schema page to operate graph spaces in NebulaGraph

[Start](#)[Documents](#)

Import Data

You can use the Import page to batch import vertex and edge data into NebulaGraph for graph exploration and data analysis.

[Start](#)[Documents](#)

Console

When data is imported, you can use the Console page to query graph data.

[Start](#)[Documents](#)



Graph Space List

[+ Create Space](#)

No	Name	Partition Number	Replica Factor	Charset	Collate	Vid Type	Group	Comment	Operations
1	ejemplo	10	1	utf8	utf8_bin	FIXED_STRING(20)		_EMPTY_	Schema

Tag

Edge Type

Index

Statistics

View Schema

Beta

+ Create

🔍 Please enter the tag name

Name	Property Num	Operations
<div> <div>−</div> <div>persona</div> </div>	2	<div> <div>✎</div> <div>🗑</div> </div>

persona Related Properties:

Property Name	Data Type	Defaults	Comment
nombre	string	_EMPTY_	_EMPTY_
edad	int32	_EMPTY_	_EMPTY_

NebulaGraph Console

ejemplo



Run

Cypher Parameter

```
1 MATCH (p:persona) RETURN p
```

```
[ejemplo]> MATCH (p:persona) RETURN p
```



Table

Graph

p

("Carlos":persona{edad: 28, nombre: "Carlos"})

("Ana":persona{edad: 25, nombre: "Ana"})

Activar Windows

Ve a Configuración para activar Windows.


```
[ejemplo]> MATCH (p:persona) RETURN p
```



Table

Graph

Vertex Details

persona

vid: "ricardo"

persona.edad: "23"

persona.nombre: "reinaldo"

ricardo

pedro

angelica

pepe

juanito



Execution Time 0.215147 (s)

File Edit Selection View Go Run ...

EXPLORER

- OUTLINE
- NEBULA5
 - .env
 - nebula-python
 - app.py
 - main.py

app.py

```
1 import tkinter as tk
2 from tkinter import ttk
3 from nebula3.gclient.net import ConnectionPool
4 from nebula3.Config import Config
5 from nebula3.data.ResultSet import ResultSet
6 import pandas as pd
7
8 def result_to_df(result: ResultSet) -> pd.DataFrame:
9     columns = result.keys()
10    d = {}
11    for col_num in range(result.col_size()):
12        col_name = columns[col_num]
13        col_list = result.column_values(col_name)
14        d[col_name] = [x.cast() for x in col_list]
15    return pd.DataFrame.from_dict(d, orient='columns')
16
17 def execute_query(query):
18     with connection_pool.session_context('root', 'nebula') as session:
19         session.execute('USE ejemplo')
20         result = session.execute(query)
21         return result_to_df(result)
22
23 def update_table():
24     query = 'MATCH (p:persona) RETURN p'
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
PS C:\Users\bryan\Desktop\nebula5> .venv\Scripts\activate
(.venv) PS C:\Users\bryan\Desktop\nebula5> python app.py
```

Activar Windows
Ve a Configuración para activar Windows.

Ln 21, Col 36 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier

Conclusiones y recomendaciones

Ventajas

NebulaGrap ofrece importantes ventajas sobre bases de datos relacionales y no relacionales tradicionales, especialmente en términos de escalabilidad y manejo de grandes volúmenes de datos no estructurados.

Recomendaciones

Para obtener los mejores resultados con NebulaGrap, se recomienda utilizarlo en proyectos que requieren un manejo eficiente de datos no estructurados y que necesitan escalar horizontalmente.



Bibliografía

1. Shi, X., Han, K., & Xia, A. (2013). A property graph model of the Web. Web Semantics: Science, Services and Agents on the World Wide Web, 19, 89-96.
2. Akdogan, E., & Balakrishnan, H. (2018). SEC-DB: scalable graph database with strong consistency. Proceedings of the VLDB Endowment, 11(12), 1849-1862.
3. Harish, P., & Narayanamurthy, S. (2013). Giraph++: Scaling Giraph to Billion-Node Graphs. Proceedings of the VLDB Endowment, 6(9), 773-784.