Cloudera Operational Database ..

Database Application Access Examples

Date published: 2020-08-14
Date modified: 2023-01-12

Legal Notice

Contents

Connect your Go applications to Cloudera Operational Database

You can connect your applications written in the Go programming language to a CDP Operational Database (COD) through Phoenix.

Before you write an application to connect to COD, do the following:

- Create a COD instance
- Get the Phoenix (Thin) client connectivity information
- Set your CDP wokload password

This application code is an example for connecting to a COD instance, creating a table, inserting a row, and then reading from the COD instance.

```
package main

import (
 "database/sql"
 "log"

 _ "github.com/apache/calcite-avatica-go"
)

func main() {
 // Connections are defined by a DSN
 // The format is http://address:port[/schema][?parameter1=value&...param
eterN=value]
 // For COD, BASIC authentication is used.
 // The workload username and password are passed as parameters avaticaUser
and avaticaPassword
 //
 // For example:
 // COD URL: 'https://gateway.env.cloudera.site/cdp-proxy-api/avatica/'
 // Workload username: jgoodson
 // Workload password: Secret1!
 // Would result in this DSN:
 dsn := "https://gateway.env.cloudera.site/cdp-proxy-api/avatica/?&authen
tication=BASIC&avaticaUser=jgoodson&avaticaPassword=Secret1!"

 log.Println("Connecting...")
 db, err := sql.Open("avatica", dsn)
 if err != nil {
  log.Fatal("Connection: ", err)
 }
 defer db.Close()

 log.Println("Create table if not exists...")
 _, err = db.Exec("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY,
 username VARCHAR)")
 if err != nil {
  log.Fatal("Create: ", err)
 }

 log.Println("Insert a row...")
 _, err = db.Exec("UPSERT INTO users VALUES (?, ?)", 1, "admin")
 if err != nil {
  log.Println("Insert: ", err)
 }
```

```
 log.Println("Reading and printing rows...")
 var (
  id       int
  username string
 )
 rows, err := db.Query("SELECT id, username from users")
 if err != nil {
  log.Fatal("Query: ", err)
 }
 defer rows.Close()
 for rows.Next() {
  err := rows.Scan(&id, &username)
  if err != nil {
   log.Fatal(err)
  }
  log.Println(id, username)
 }
}
```

Related Information

How to connect Go Applications to Cloudera Operational Database

Connect your Python applications to Cloudera Operational Database

You can connect your applications written in Python programming language to a CDP Operational Database (COD) through Phoenix.

Before you write an application to connect to COD, do the following:

- Create a COD instance
- Get the Phoenix (Thin) client connectivity information
- Set your CDP wokload password

This application code is an example of validating your connection to COD.

```
import phoenixdb
import configparser

class Database:

    def connect(self):
        REQUIRED_OPTS = ['Username', 'Password', 'Url']
        config = configparser.ConfigParser()
        config.read('config.ini')
        if not 'COD' in config:
            raise Exception("Could not find section for COD in config.ini")
        cod_config = config['COD']
        opts = {}

        # Validate the configuration
        for required_opt in REQUIRED_OPTS:
            if not required_opt in cod_config:
                raise Exception("Did not find %s in configuration" % (requ
ired_opt))

      # Provide non-required options
        if 'Truststore' in cod_config:
            opts['verify'] = cod_config['Truststore']
```

```
        if 'Authentication' in cod_config:
            opts['authentication'] = cod_config['Authentication']
        else:
            opts['authentication'] = 'BASIC'
        # Read required options
        opts['avatica_user'] = cod_config['Username']
        opts['avatica_password'] = cod_config['Password']
        return phoenixdb.connect(cod_config['Url'], autocommit=True, **opts)
```

This code is an example of creating a table and adding data to a COD instance.

```
import phoenixdb
from database import Database

def load():
    db = Database()
    conn = db.connect()
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY
, username VARCHAR)")
    cursor.execute("UPSERT INTO users VALUES (?, ?)", (1, 'admin'))
    cursor.execute("SELECT * FROM users")
    print(cursor.fetchall())
    conn.close()

if __name__ == '__main__':
    load()
```

Related Information

Building a Simple CRUD web application and image store using Cloudera Operational Database and Flask