

7. Cassandra Data Tables as REST API

High-Level Design Document

Team No : 15

Vinayak Bharti (201405522)
Akash Kant (201230194)
Mihir Wadwekar(201202026)
SVK Rohit (201356204)

Problem Statement

To provide a web-based UI based on REST API for an user to manage Cassandra data tables using python with the following functionalities :

- Operation on keyspace.
creating a keyspace, retrieving information about the keyspaces, retrieving properties about the keyspaces and dropping keyspaces.
- Operations on column family
creating a column family, get description of a column family, dropping column families.
- Rows and Columns Operations
Implementing a tool for inserting rows and columns, deleting rows and columns.
- Pagination
Showing the query result from cassandra with pagination

Approach

Broadly speaking the project will be primarily consist of two parts. Back-end would deal with connecting with Cassandra and fetching the required data. Front-end would deal with presenting an UI from which user can access and modify the database.

The approach for the project implementation would be split as follows.

- Backend - This will involve implementing the backend server for the data tables along with the above mentioned functionalities in python.

- Frontend - This will involve implementing the user interface in the form of a web-based server from which a user can securely and easily access and retrieve information from the database using Jinja2 templates.

Project Plan

The project will be divided into two modules - Backend and the front-end.

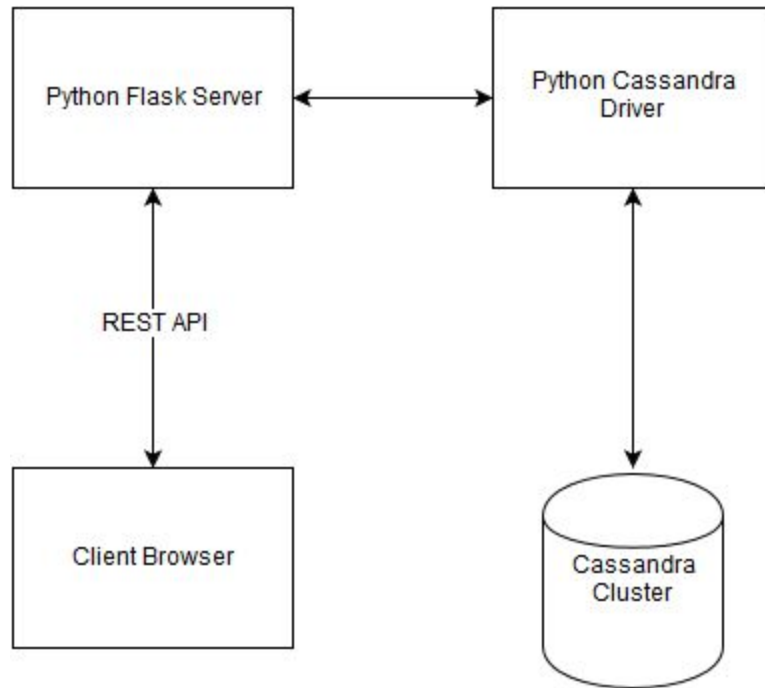
Backend planning

- Creating a cassandra cluster and configuring python to communicate with cassandra access the data tables.
- A python application to execute queries on Cassandra passed on from the Web server.
- A python flask server that will accept requests from the user, process it and forward it to cassandra. Then retrieve the results from the data tables and respond back to the user in HTML format using tools such as jinja2 templates.

Front-end planning

- Designing a layout listing out all the functionalities of our web application.
- Appropriately presenting the information retrieved from the server to the users in an secured and easy to manage way.
- User should be able to carry out all the mentioned functionalities without the knowledge of the underlying Cassandra database system.

High Level Design



Tools / Libraries

- **Cassandra** : Apache Cassandra is an open source distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple datacenters, with asynchronous masterless replication allowing low latency operations for all clients. The Apache Cassandra database provides high scalability and high availability without compromising performance.
- **Python** : The programming language used to build the backend server.
- **Python Flask** : Flask is a micro web application framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. Flask provides ability to rapidly deploy web applications using RESTful request dispatching through python.

- **Jinja2** : Jinja2 is one of the most used template engines for Python. We will be using to develop our User frontend interface.

Final Deliverables

At the end of this project, the outcomes expected are as follows:

- An User Friendly web application where the user can access the Cassandra Server, providing the user with the above-mentioned functionalities. The user would be able to operate without the need for understanding the underlying Cassandra Query Language.
- Display data tables retrieved from the database with pagination.