CASSANDRA REST API

TEAM 15
PROJECT 7

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

PROBLEM STATEMENT

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

- Operation on keyspace.

  creating a keyspace, retrieving information about the keyspaces, dropping keyspaces and list keyspaces.
- Operations on column family

  creating a column family, get description of a column family, dropping column families and list column families.
- Rows Operations

  Implementing a tool for inserting rows, deleting rows, retrieving them and list them.
- Node Operations

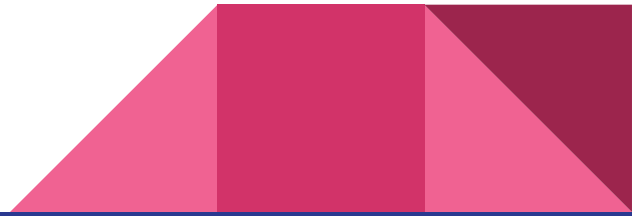
  Implementing a tool for adding machines, deleting and listing them.
- Pagination

  Showing the query result from cassandra with pagination for all the above queries.

BASIC DESIGN

The project will be primarily consist of three parts.

- Back-end would deal with connecting with Cassandra and fetching the required data.
  - Backend - This will involve implementing the backend server for the data tables along with the above mentioned functionalities in python.
- Front-end would deal with presenting an UI from which user can access and modify the database.
  - 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.
- Another part of the project is to follow the design of Pragmatic Restful API.
  - GET - List
  - POST - Create
  - DELETE - Delete
  - GET with data - Info about that particular thing.



PLANNING

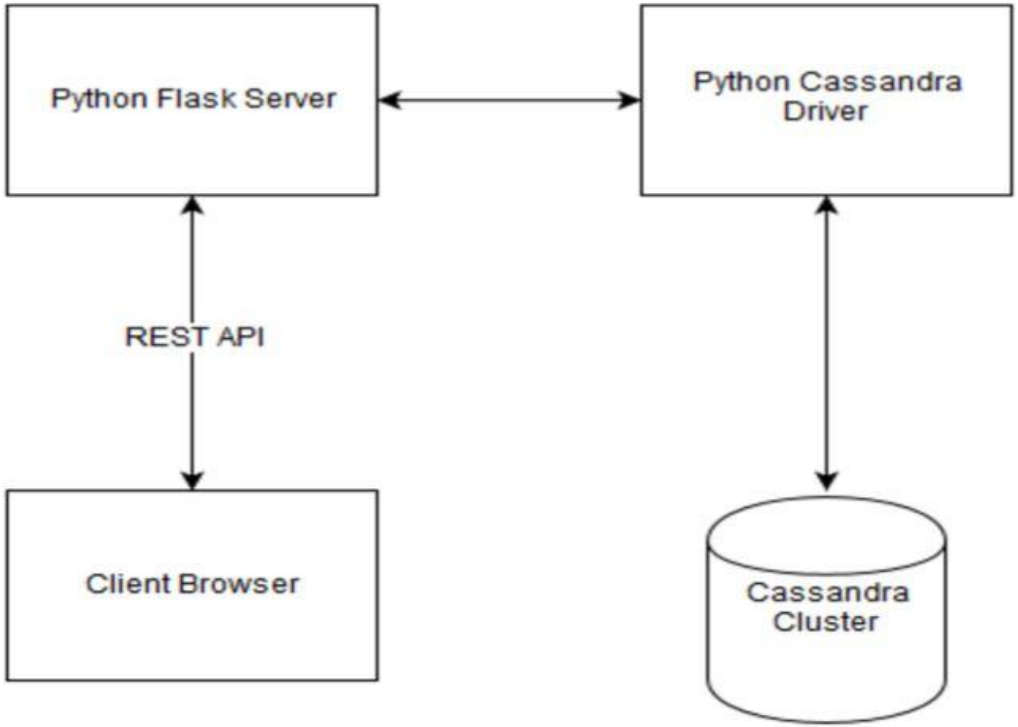
Backend planning

- Creating a cassandra cluster and configuring python using Python Cassandra Driver to communicate with cassandra access the data tables and nodes.
- 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(HDL)



API's

Example:- **Keyspace**

- ➔ List all the Keyspaces - [GET] http://127.0.0.1:5000/keyspaces/
- ➔ Get Info about a keyspace - [GET] http://127.0.0.1:5000/keyspaces/keyspaceid
- ➔ Creates a new keyspace - [POST] http://127.0.0.1:5000/keyspaces/ - With Body containing data
- ➔ Updates the keyspace - [PUT] http://127.0.0.1:5000/keyspaces/keyspaceid - With Body containing data
- ➔ Deletes the keyspace - [DELETE] http://127.0.0.1:5000/keyspaces/keyspaceid - With Body containing data

- Relation between areas are handled as mentioned by the rules

Example:- **Keyspace and Column Family**

- ➔ Get info of a column family - [GET] http://127.0.0.1:5000/keyspace/keyspaceid/columnfamilys/columnfamilyid

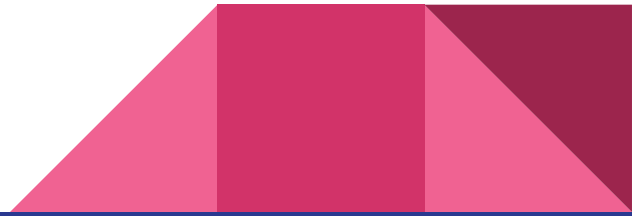
It will give us the column family of the keyspace with id - keyspace id

TOOLS AND 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.

Future Enhancements

- **Security** is very important for any software that a developer builds.
- We will implement an authorization component to secure the user’s info.
- **Abstraction** is an important concept to remove the implementation detail. So, we will add an export feature to this software which will provide different formats to the user.
- **Export** and **Import** Option



Thank You

- Team - 15
- Project no 7