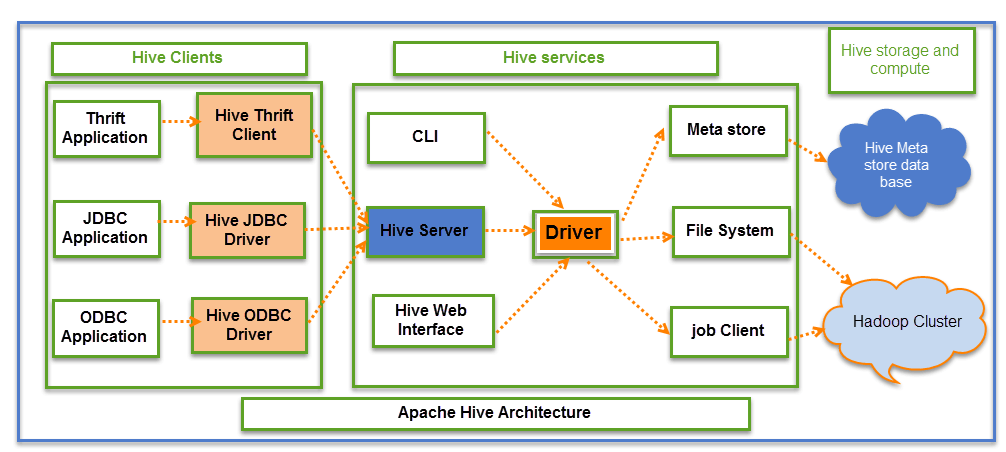
1. **Hive Architecture**



The above screenshot explains the[Apache](https://www.guru99.com/apache.html)Hive architecture in detail

Hive Consists of Mainly 3 core parts

1. **Hive Clients**
2. **Hive Services**
3. **Hive Storage and Computing**

**Hive Clients:**

Hive provides different drivers for communication with a different type of applications. For Thrift based applications, it will provide Thrift client for communication.

For[Java](https://www.guru99.com/java-tutorial.html)related applications, it provides JDBC Drivers. Other than any type of applications provided ODBC drivers. These Clients and drivers in turn again communicate with Hive server in the Hive services.

**Hive Services:**

Client interactions with Hive can be performed through Hive Services. If the client wants to perform any query related operations in Hive, it has to communicate through Hive Services.

CLI is the command line interface acts as Hive service for DDL (Data definition Language) operations. All drivers communicate with Hive server and to the main driver in Hive services as shown in above architecture diagram.

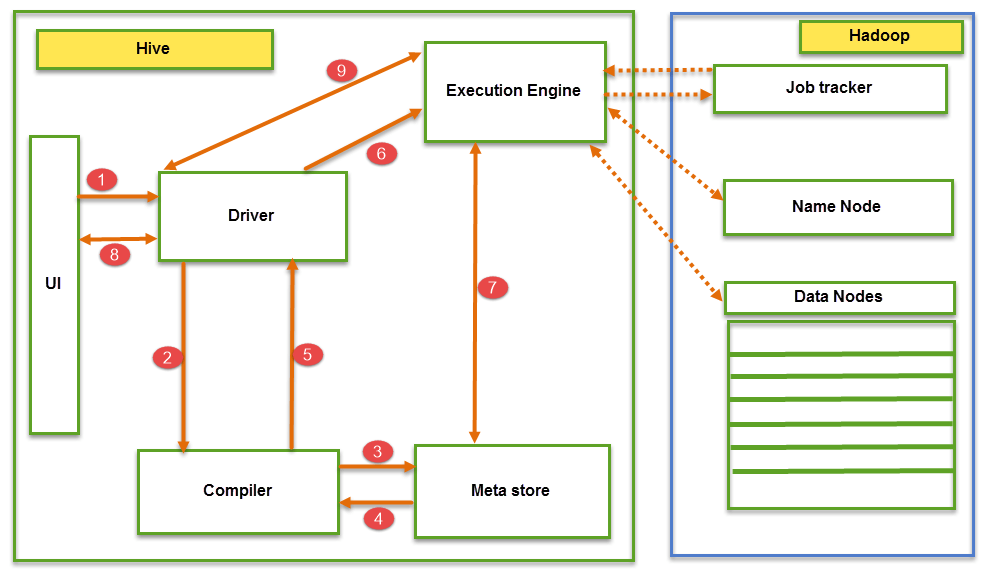
Driver present in the Hive services represents the main driver, and it communicates all type of JDBC, ODBC, and other client specific applications. Driver will process those requests from different applications to meta store and field systems for further processing.

**Hive Storage and Computing:**

Hive services such as Meta store, File system, and Job Client in turn communicates with Hive storage and performs the following actions

* Metadata information of tables created in Hive is stored in Hive "Meta storage database".
* Query results and data loaded in the tables are going to be stored in Hadoop cluster on HDFS.

**Job execution flow:**



From the above screenshot we can understand the Job execution flow in Hive with Hadoop

**Components OF HIVE:**

The data flow in Hive behaves in the following pattern;

1. Executing Query from the UI( User Interface)
2. The driver is interacting with Compiler for getting the plan. (Here plan refers to query execution) process and its related metadata information gathering
3. The compiler creates the plan for a job to be executed. Compiler communicating with Meta store for getting metadata request
4. Meta store sends metadata information back to compiler
5. Compiler communicating with Driver with the proposed plan to execute the query
6. Driver Sending execution plans to Execution engine
7. Execution Engine (EE) acts as a bridge between Hive and Hadoop to process the query. For DFS operations.

* EE should first contacts Name Node and then to Data nodes to get the values stored in tables.
* EE is going to fetch desired records from Data Nodes. The actual data of tables resides in data node only. While from Name Node it only fetches the metadata information for the query.
* It collects actual data from data nodes related to mentioned query
* Execution Engine (EE) communicates bi-directionally with Meta store present in Hive to perform DDL (Data Definition Language) operations. Here DDL operations like CREATE, DROP and ALTERING tables and databases are done. Meta store will store information about database name, table names and column names only. It will fetch data related to query mentioned.
* Execution Engine (EE) in turn communicates with Hadoop daemons such as Name node, Data nodes, and job tracker to execute the query on top of Hadoop file system

1. Fetching results from driver
2. Sending results to Execution engine. Once the results fetched from data nodes to the EE, it will send results back to driver and to UI ( front end)

Hive Continuously in contact with Hadoop file system and its daemons via Execution engine. The dotted arrow in the Job flow diagram shows the Execution engine communication with Hadoop daemons.