Platform Components and Services

The following are built-in components & services –

1. UI Dashboard

- **“Services” tab** - Interface for managing all the micro services running in the system from a single interface as well as setting its resources allocation and monitoring its usage

- **“Identity” tab** - Users can set various permission roles which restrict view and reports to allowed access

- **“Data tab”** - browse your data: files, objects, tables (with its partitions and rows), all organized in an hierarchical order.

2. Data Fabric

Iguazio has built-in data layer for storing and analyzing various types of data structures such as key value, time series, streaming,
files and objects.

- **Multi Model data layer**
  The platform introduces a unique multi-data model. The platform maintains multiple indexes to the data delivering high performance regardless of the access pattern, eliminating the need for multiple data stores, constant synchronization, complex pipelines, and painfull ETL processes.

- **NoSQL (KV wide columnar) database -**
  The database has been built to take advantage of distributed cluster of flash memory based machines (and VMs) delivering in-memory performance while keeping flash economy and density. SAme data can be accessed via: Rest API, SQL (Presto) , Spark dataframe and Pandas (& Dask) data frames.

- **Streaming engine** - users can stream data directly to the platform using Rest API with AWS Kinesis like semantics.

- **FIle system & Objects** - Iguazio provides a file system (linux File System and HDFS) for storing files as well as S3 like objects. Users can often use it to store CSV, Parquet , Avro, images and videos or any other file. Data inserted using Rest API (S3 like) can be read also as File (HDFS, Linux FS).

- **Time series database** - Iguazio provides time series database with a rich set of features for analyzing and storing time series data efficiently. The time series service is based on iguazio open source library called V3IO TSDB which exposes an API for creating, updating, querying, and deleting time series databases, and comes with a complementary command-line interface (CLI) tool (**tsdbctl**). Iguazio has built-in Prometheus service integrated the time series database (with Prometheus enhanced storage using Iguazio database) making it a robust, scalable and high performing solution.

- **WebAPI** - Users can write and read data from the platform using RestAPI. The API is similar to the AWS web APIs semantic (e.g. Kinesis for streaming, Dynamodb for NoSQL and S3 for objects). The WebAPI endpoint service can be found under the services screen.
  API reference guide - https://www.iguazio.com/docs/reference/latest-release/api-reference/

3. External Data sources, an open environment

iguazio is an open platform composed of standard open source analytics and ML tools. Storing data using its built-in data layer is optional. Its tools can access external RDBMS, traditional Hadoop “data lake” , AWS S3 and many common data sources. The Platform uses the open source versions with iguazio driver taking advantage of its indexing, data processing, pre-aggregation and underling fast database to achieve significant acceleration and PaaS ease of use.

4. Tools to Collect, Analyze, Train and Deploy your data

- **Serverless framework** (Nuclio)- Enterprise edition of Nuclio, the leading multi-cloud open source Serverless Functions project. Delivers high performance low latency framework with wide support of tools and event triggering.
  - Can be used for collecting data as an ongoing basis. It has built-in templates for collecting data from common source (e.g. kafka, databases etc..) with examples of data enrichment and data pipeline processing.
  - Can be used for running machine learning models in the serving layer supporting high throughput on demand with elastic resource allocation
  - It has a simple integration with Jupyter notebook enabling users to create their code in Jupyter (model, feature vector and application) and use single command to deploy all as a Serverless Function running in the serving layer

- **Spark** - Spark users can access files, tables or streams stored on iguazio data platform through the native spark Dataframe interface, or access external data sources like RDBMS or traditional Hadoop “data lake”. The Platform uses Spark open source version with iguazio driver for further acceleration of its data

processing. Spark supports, among others: Spark SQL, Spark MLib, SparkR and Spark Streaming.

- **SQL engine** (Presto) - Iguazio support ANSI SQL for interactive SELECT statements. e.g. - users can run SQL command from Jupyter, Nuclio or from a Presto client. Its ODBC/JDBC API allows seamless integration with wide range of tools like Tableau, QlikView and more, running at scale and high performance. In order to get extra horsepower you should launch through the Services tab another Presto service. This will use the underling Kubernetes to easily allocate resources and run.

- **Prometheus Time series database** - Iguazio provides time series database with Prometheus service. Customer can run queries using Prometheus.
  For a full overview of Iguazio time series go to:
  https://igzdocsdev.wpengine.com/techpreview-dev/tutorials/latest-release/tsdb/
  For specific time series examples go to: <Jupy Time series NB>

- **Pandas, Dask -** runs with its data frame enabling high performance Python based data processing.
- **Machine Learning Packages** - the platform has built-in integrated and accelerated ML packages such as: Scikit learn , Pyplot , numpy, Pytorch and Tensorflow.

- **Grafana** -Grafana is a metric analytics & visualization suitetool. Users can easily create RT dashboard and reports based on iguazio NoSQL tables or time series with Prometheus.

- **Notebook** -
  - Jupyter is the main tool for data exploration and train. It is integrated with all key analytics services allowing users to ingest data, run Spark jobs, SQL (via Presto) and Pandas dataframe from a single interface, tools are running on same data concurrently with no need to move the data.
  - For running python jobs at scale user can run distributed pandas jobs by leverage the built-in Dask library

  ** Please note that users can work with Zeppelin notebook as well (see “Services” UI screen).