ELEVE8



ENERGY UTILITY PROJECT

TOOLS USED: -

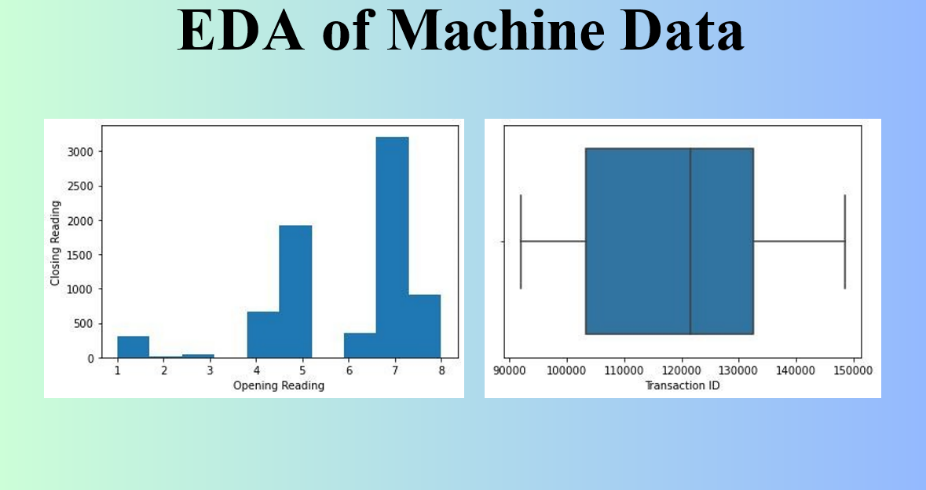
* Python
* Node Red
* AWS
* PostgreSQL / Influx DB
* Grafana
* ESP8266 Module

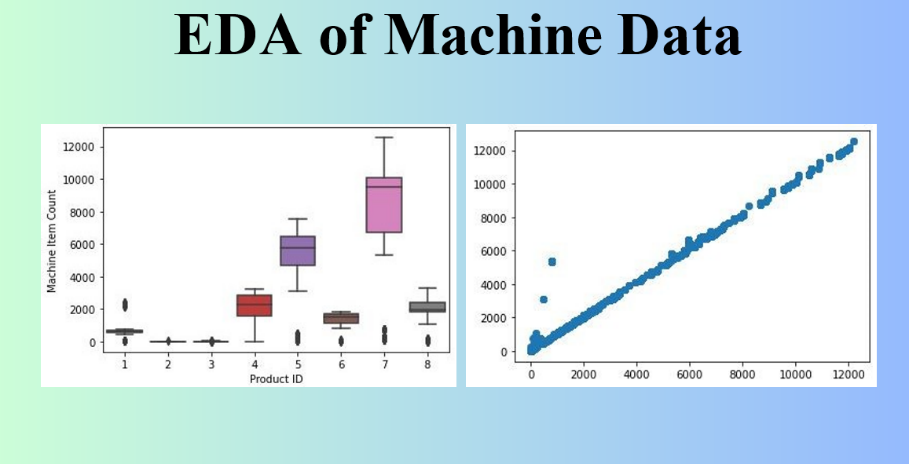
WORK FLOW OF PROJECT: -

* Making Mind Map
* Parsing of Data
* EDA of Machine Data
* Node Red
* Node Red to AWS
* XLSX to Node Red Connectivity
* PostgreSQL / Influx DB
* Database to Grafana Connectivity
* Grafana data visualization (DASHBOARD)
* Machine Learning Application with mathematical intuition
* Hardware + Software Configuration Flow Diagram
* Client – Server Architecture using ESP8266
* Interactive Application using Flask (PYTHON)

EDA: -

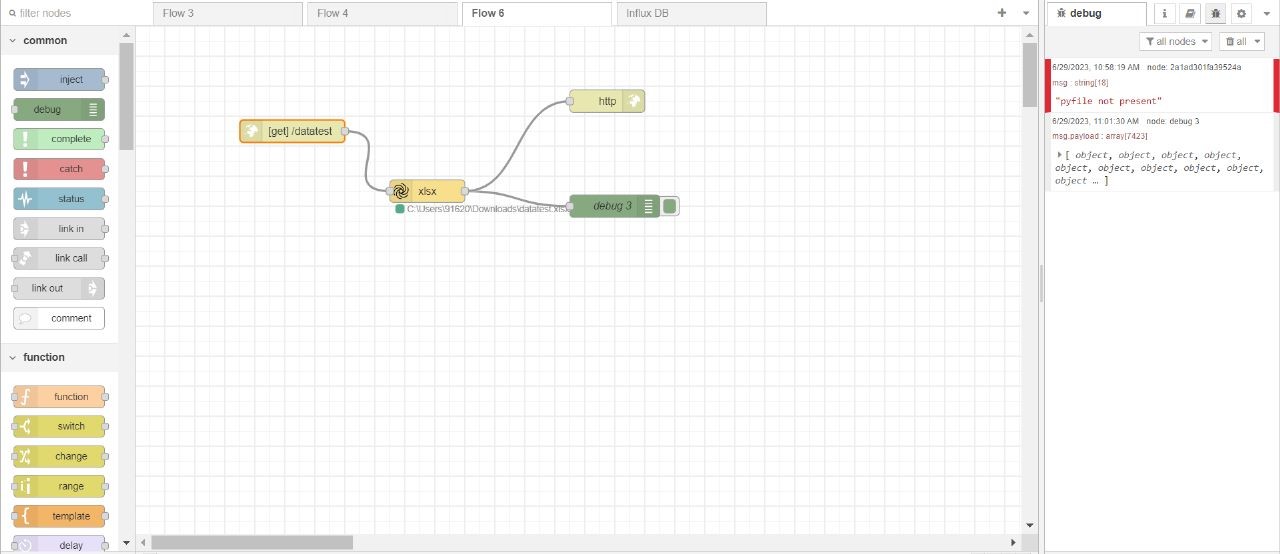
* Eda is a phenomenon under data analysis used for gaining a better understanding of data. It includes:
* Main features of data
* Variables and relationships between them Identifying which variable is important
* Descriptive statistics is a way to understand characteristics of your data. Visualization and EDA consists of:
* Histograms Box Plots Scatter Plot etc.





NODE RED: -

* Node Red is a stream-based advancement instrument for visual programming. Its main focus is on visual apparatus for wiring the Internet of Things.
* It is for wiring together hardware equipment (sensors etc.) and APIs. It was developed my IBM Emerging Technology organization.



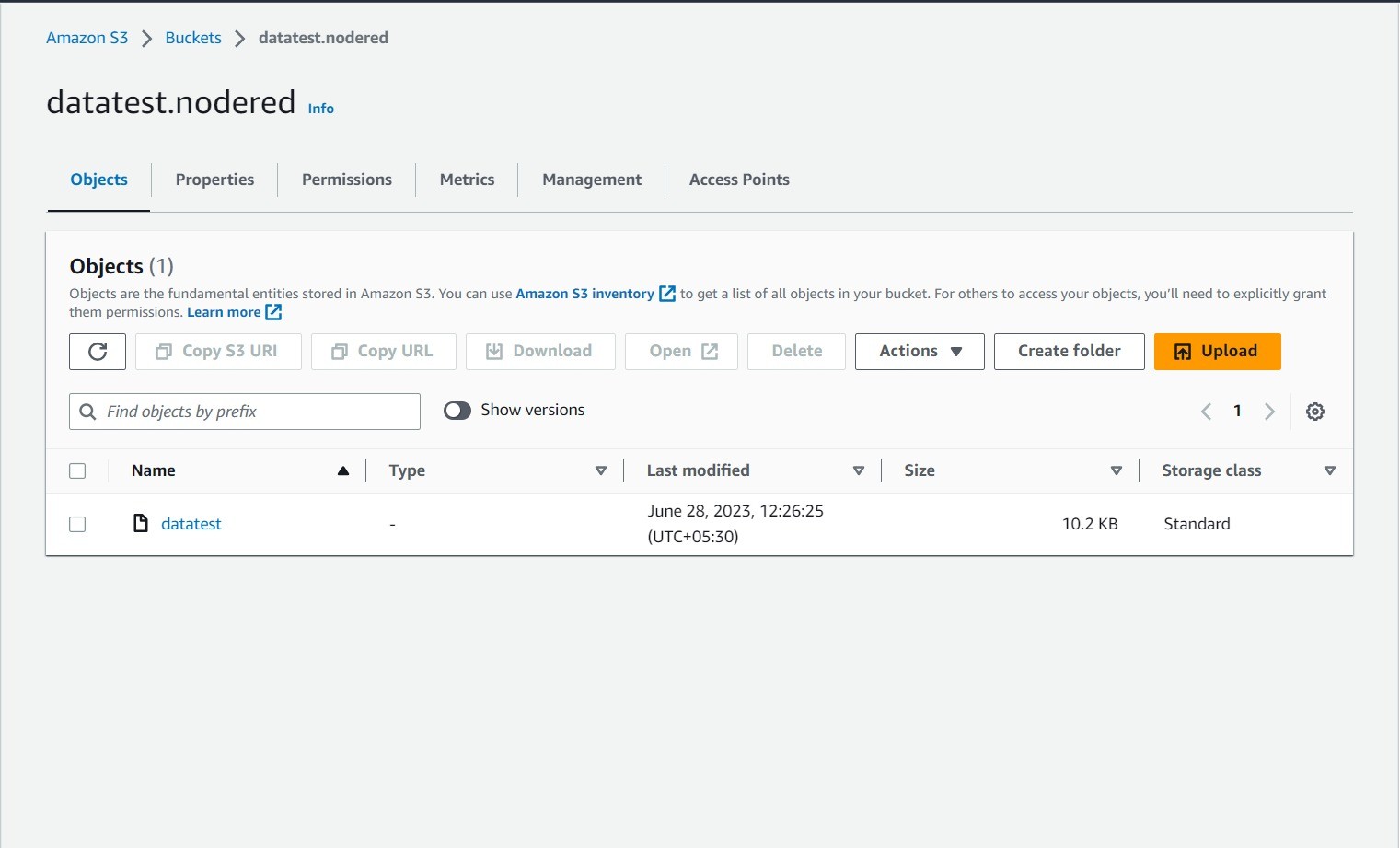
NODERED TO AWS: -

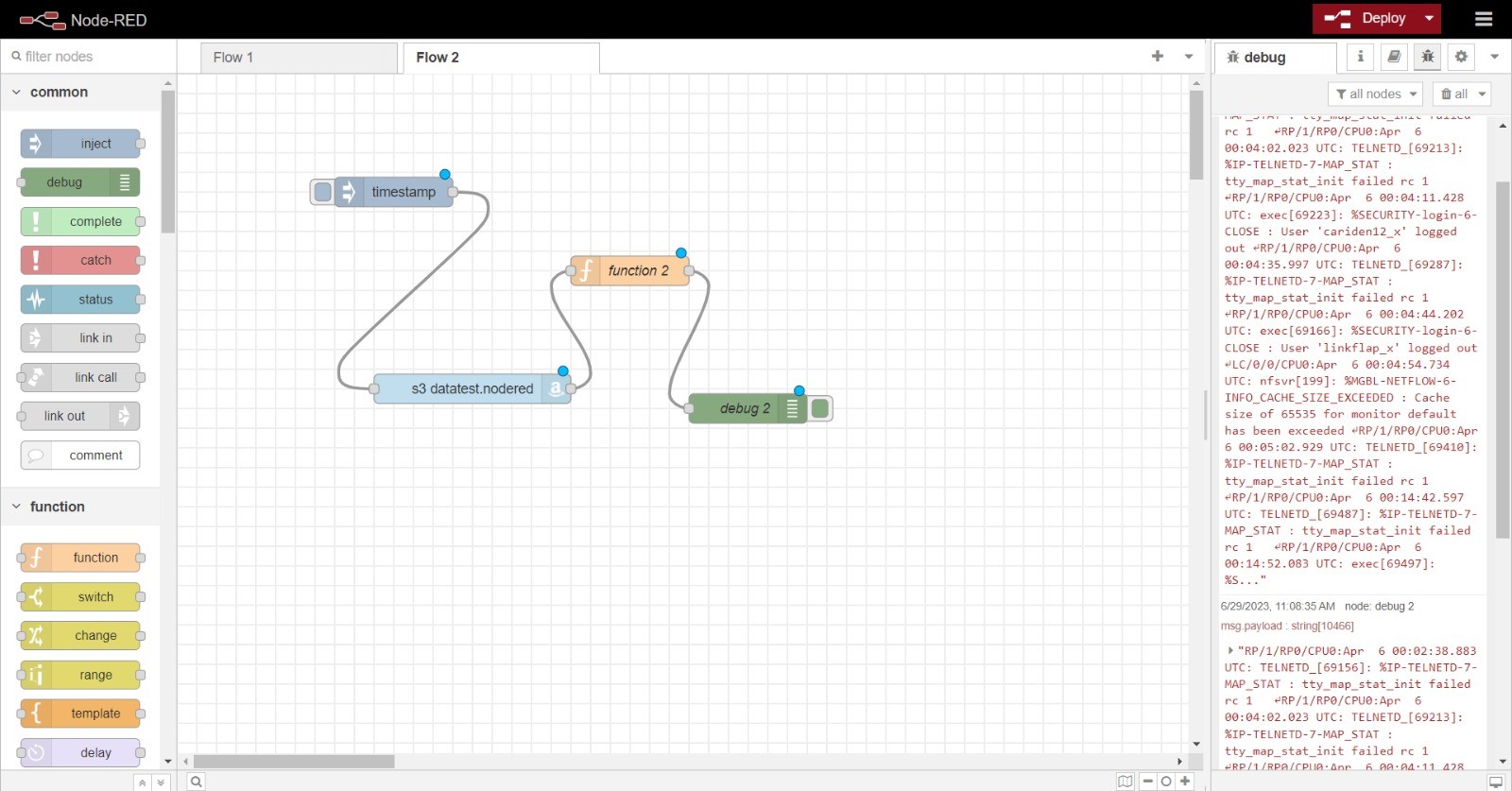
1. **Create s3 bucket in AWS console**
2. **Create IAM user in AWS console**
3. **Create Access key to access AWS from third party application**
4. **For Node red install AWS-SDK pallet**
5. **Choose S3 out node and fill the necessary details: AWS access keys**

**Bucket name**

**File location Bucket region**

1. **Choose inject node and connect it with AWS s3 out node**
2. **Deploy then click inject button on inject node**
3. **File will upload to the bucket**





XLSX to Node Red Connectivity: -

STEP 1 : Open Node Red from terminal

STEP 2 : Take a http input nod for the input

STEP 3 : Specify the address of the XLSX file

STEP 4 : HTTP response is sent out

STEP 5 : Take one debug node for the output

STEP 6 : Deploy

Grafana Visualization: -

1. **Why we choose Grafana for Visualization.**

The preference for Grafana over Power BI and Tableau, or vice versa, depends on the specific needs and requirements of the users and the use case. Each of these tools has its own strengths and weaknesses, and the choice between them should be based on factors such as data sources, visualization capabilities, ease of use, scalability, and cost.

Here are some reasons why Grafana might be preferred over Power BI and Tableau in certain situations:

Open-source nature: Grafana is an open-source platform, which means it is freely available and allows for more customization and community-driven development. This can be advantageous for organizations looking to leverage the power of open-source tools and have more control over their analytics platform.

Focus on time-series data: Grafana is particularly well-suited for monitoring and visualizing time-series data, such as metrics from servers, sensors, or other devices. It has extensive support for various time-series databases like Prometheus, Influx DB, and Graphite, making it popular for monitoring applications.

Flexibility and extensibility: Grafana offer a wide range of plugins and integrations, allowing users to connect to various data sources and extend its functionality. This flexibility makes it an attractive choice for organizations with diverse data ecosystems or unique data requirements.

User-friendly interface: Grafana is known for its intuitive and user-friendly interface, making it relatively easy for users to create and customize dashboards and visualizations without requiring extensive technical knowledge

