A PROJECT REPORT ON

IOT Wind Turbine Insights and Intelligence using GGSA

- Keep it running

A dissertation submitted in partial fulfillment of the academic requirement for the award of the degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Rohit Lingala **1602-18-735-031**

Under the guidance of

Mr. Chandra Perni		**Mr. V. Krishna Mohan**
Manager		**Assistant Professor**
Oracle Energy and Water	&	Department
Integrations Development		Of
Oracle India Private Limited		Electronics and Communication

Department of Electronics and Communication Engineering

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

IBRAHIMBAGH, HYDERABAD-500031

2018-2022

Department of Electronics and Communication Engineering

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

IBRAHIMBAGH, HYDERABAD-500031

Date:

CERTIFICATE

This is to certify that the project work titled "**IOT Wind Turbine Insights and Intelligence using GGSA**" submitted by

Rohit Lingala **1602-18-735-031**

students of Electronics and Communication Engineering Department, Vasavi College of Engineering in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Electronics and Communication Engineering is a record of the bonafide work carried out by them during the academic year 2021-2022. The result embodied in this project report has not been submitted to any other university or institute for the award of any degree

External Guide	Internal Guide	Head of the Department
Chandra Perni	**V. Krishna Mohan**	**Dr. E.SREENIVASA RAO**
Integrations Manager	Assistant Professor	Professor & HoD
Oracle India Private Limited	E.C.E Department	E.C.E Department

External Examiner

PROJECT COMPLETION CERTIFICATE

This is to certify that the project work titled "**IOT Wind Turbine Insights and Intelligence using GGSA**" submitted by

Rohit Lingala **1602-18-735-031**

student of Electronics and Communication Engineering Department, Vasavi College of Engineering in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Electronics and Communication Engineering is a record of the bona fide work carried out by them during the academic year 2021-2022 in the form of Internship work under the Guidance of Manager, Chandra Perni making his work eligible to be credited as legitimate and is permitted to consider for credit scoring for academic purposes.

The result embodied in this project report has not been submitted to any other university or institute for the award of any degree.

Project Internship Guide

Chandra Perni

Energy and Water, Integrations Manager

Oracle India Private Limited

DECLARATION

This is to state that the work presented in this thesis titled **"IOT Wind Turbine Insights and Intelligence using GGSA"** is a record of work done by us in the Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad. No part of the thesis is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the project work done entirely by us and not copied from any other source. I hereby declare that the matter embedded in this thesis has not been submitted by me in full or partial thereof for the award of any degree/diploma of any other institution or university previously.

6/1/2022

Signature of the student

Rohit Lingala 1602-18-735-031

ACKNOWLEDGEMENTS

This satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our project, which is the result of a studied blend of both research and knowledge.

A special thanks to Oracle India for providing me the opportunity to work and develop myself. I would like to express my sincere gratitude for all the time and effort they have invested in me. I want to express my heartfelt gratitude to my external guides who are, my manager, **Mr. Chandra Perni**, and my mentor **Mr. Ramakrishna Tadi**, who have not only encouraged me to excel in my project work but also to learn and achieve more throughout my time at Oracle

It is our privilege to express our earnest gratitude and venerable regards to our internal guide **Mr. V. Krishna Mohan**, Assistant Professor, E.C.E. Department, Vasavi College of Engineering, Ibrahimbagh, for abounding and able guidance during the preparation and execution of the project work. We are grateful for his cooperation and his valuable suggestions.

We record with pleasure our deep sense of gratitude to **Dr. E. SREENIVASA RAO**, Head of the Department, E.C.E. for his simulating guidance and profuse assistance we have received, which helped throughout the project.

We also thank our Project coordinators, **Ms**. **G.R.Padmini**, Associate Professor, Department of ECE, and **Mrs. A.SriLakshmi**, Associate Professor, Department of ECE, for their continuous guidance and valuable suggestions throughout our project.

Our sincere thanks to the **Principal** and **Management, Vasavi College of Engineering, Hyderabad** for providing all the facilities in carrying out the project successfully.

Also, we acknowledge with thanks for the support extended by all the staff members and technical staff in shaping up our project. We are thankful to one and all who co-operated with us during the course of our project work.

Rohit Lingala **1602-18-735-031**

TABLE OF CONTENTS

ABSTRACT 6

LIST OF TABLES 7

LIST OF FIGURES 8

LIST OF ABBREVIATIONS 11

1. INTRODUCTION 0

1.1 About Organization 0

1.1.1 Oracle: 0

1.1.2 Global Business Units (GBU) 1

1.1.3 Oracle Energy and Water 1

1.1.4 Integrations 2

1.1.5 Innovation Center 2

1.2 Aim and Objective: 3

2. LITERATURE SURVEY 4

2.1 Traditional Maintenance Programs: 4

2.1.1 Reactive maintenance 4

2.1.2 Preventive maintenance 4

2.2 Modern Maintenance Program 5

2.2.1 Condition Monitoring (Methodology) 5

2.2.2 Predictive Maintenance 6

2.3 Market and the potential opportunities in Condition Monitoring 7

2.4 Comparing the maintenance programs 8

2.5 Industrial Maintenance Statistics 9

3. Propose Solution 12

3.1 High-level view 12

3.2 Block Diagram 12

3.3 Tools & Technologies used 12

3.3.1 Oracle IoT Digital Twin Simulator 13

3.3.1.1 What is a Digital Twin? 13

3.3.2 Golden Gate Stream Analytics (GGSA) 14

3.3.2.1 What is Streaming analytics? 14

3.3.2.2 What is Streaming Data? 14

3.3.2.3 About Apache Spark 15

3.3.3 Why and What is Kafka? 17

3.3.4 Need for Web Services 20

3.3.4.1. What is SOAP? 21

3.3.4.2 What is REST? 22

3.3.4.3 What is Postman? 23

3.3.4.4 Need of Soap UI? 23

3.3.5 Oracle Cloud Infrastructure (OCI) 24

3.3.6 Oracle Autonomous Database 25

3.3.7 Oracle Utilities Work and Asset Cloud Service (WACS) 26

3.3.8 Oracle Integration Cloud (OIC) 28

3.3.8.1 What is Integration? 28

3.4 System Design 29

3.5Results 31

4. CONCLUSION AND FUTURE WORK 41

4.1 Conclusion 41

4.2 Future Scope 42

REFERENCES 43

APPENDIX 45

ABSTRACT

Develop a solution that can effectively monitor assets related to utilities by minimizing human intervention, labor, and repair costs.

As it is known that despite exhaustive routine maintenance programs faults develop in machinery, therefore it is crucial to have a proper maintenance program to limit breakdowns actively and optimize maintenance resources wherever required. In addition to identifying developing faults within machinery, factors responsible for causing these faults to build, have to be identified.

In our case, our asset is the Wind Turbine and our goal is to monitor Wind Turbine continually and identify a developing fault in advance before sufficient damage has taken place, and raise tickets automatically to inspect the device and respond accordingly.

This project is part of Oracle's Innovation Lab initiative. For IoT Projects an environment has been set up, which contains the digital twins of different devices such as Wind turbines, Transformers, Sensors, etc. The entire lab is also a simulated household, complete with electrical appliances, and gas and water meter connections. This lab's data is retrieved and ingested into a data analytics platform. The goal of this project is not only to raise tickets by monitoring the device's health but also to generate real-time insights on the data that flows into this platform and then create visualizations based on those insights to assist customers in making informed decisions.

LIST OF TABLES

Table 2.2. 1 Pros and Cons of Condition Monitoring 5
Table 2.2. 2 Pros and Cons of Predictive Maintenance 6
Table 2.4 1 Condition Based vs Predictive Maintenance ……………………………………..8
Table 2.4 2 Comparing all the Maintenance Programs 8

LIST OF FIGURES

Figure 2.1. 1 Reactive vs Predictive vs Preventive 4
Figure 2.2. 1 Condition based Monitoring 5
Figure 2.2. 2 Resistance to failure vs Time 6
Figure 2.3 1 Potential Opportunities in Predictive Maintenance Market 7
Figure 2.3 2 Market of Condition Monitoring 7
Figure 2.3 3 Growth of Predictive Maintenance from 2017-2022 7
Figure 2.5 1 what affects plant Productivity? 9
Figure 2.5 2 Plant Engineering Industrial Maintenance Study about unplanned downtime .. 9
Figure 2.5 3 Plans to decrease Unscheduled downtime 10
Figure 2.5 4 Impact of IIOT on Maintenance Operations 10
Figure 2.5 5 Benefits of Sensors in Maintenance 11
Figure 2.5 6 Technology Implementation Plans 11
Figure 2.5 7 Report on Predictive Maintenance 11
Figure 3.1. 1 High Level view of the Solution 12
Figure 3.2 1 Block Diagram 12
Figure 3.3. 1 Iot Digital Twin Simulator dashboard 13
Figure 3.3. 2 Stream Analytics Architecture 14
Figure 3.3. 3 Connectors in Apache Spark 15
Figure 3.3. 4 GGSA Connectors 16
Figure 3.3. 5 Kafka Architecture 17
Figure 3.3. 6 Famous Firms using Kafka 17
Figure 3.3. 7 Coupling in absence of messaging system 18

Figure 3.3. 8 Kafka decouples source and target 18

Figure 3.3. 9 Kafka Internal Working 19

Figure 3.3. 10 Soap request response 21

Figure 3.3. 11 SOAP vs REST 21

Figure 3.3. 12 REST API Architecture 22

Figure 3.3. 13 Postman UI 23

Figure 3.3. 14 Services Provided by OCI 25

Figure 3.3. 15 WAM Dashboard 27

Figure 3.3. 16 ETL - Data Integration Process 28

Figure 3.4. 1 System Design 29

Figure 3.4. 2 Connections and Targets in GGSA 29

Figure 3.4. 3 OIC Adapters Used 30

Figure 3.5. 1 Iot Dashboard 31

Figure 3.5. 2 Testing Kafka in OIC GGSA instance 31

Figure 3.5. 3 API to fetch wind turbine attributes 32

Figure 3.5. 4 Testing Kafka with GGSA 32

Figure 3.5. 5 Data Flow in GGSA 32

Figure 3.5. 6 Fetching sensor data to cloud and saving into a json file using python script. . 33

Figure 3.5. 7 Pushing the json file from cloud to Kafka and pulling it from GGSA 33

Figure 3.5. 8 Making the pushing and pulling process recurring using bash script 33

Figure 3.5. 9 Sensor data being populated in GGSA 34

Figure 3.5. 10 Data Pipeline designed in GGSA 34

Figure 3.5. 11 Max wind speed and Power generated stored in the database for every Hr... 35

Figure 3.5. 12 Outage event simulated in IoT cloud service 35

Figure 3.5. 13 Outage event detected in GGSA which triggers OIC Integrations.................. 36

Figure 3.5. 14 OIC dashboard where integration status can be verified................................ 36

Figure 3.5. 15 API to check the status of the activity... 37

Figure 3.5. 16 Wind turbine work order in WAM.. 37

Figure 3.5. 17 Activity created automatically in WAM by OIC Integration......................... 38

Figure 3.5. 18 API to turn off the Wind turbine 38

Figure 3.5. 19 Activating Activity in WAM 40

Figure 3.5. 20 Finishing Activity in WAM 39

Figure 3.5. 21 Asset score is modified... 39

Figure 3.5. 22 After the activity completion state of the wind turbine is restored using API 39

Figure 3.5. 23 Event details are stored in the database.. 40

Figure 3.5. 24 Soap UI... 40

LIST OF ABBREVIATIONS

OCI	Oracle Cloud Infrastructure
OIC	Oracle Integration Cloud
WACS	Oracle Utilities Work and Asset Cloud Service
OFS	Oracle Field Service
REST	Representational State Transfer
ONNX	Open Neural Network Exchange
PMML	Predictive Model Markup Language
GGSA	GoldenGate Stream Analytics
API	Application Program Interface
SOAP	Simple object access protocol

1. INTRODUCTION

1.1 About Organization

1.1.1 Oracle:

Oracle is a business-based multinational computer technology corporation, the world's leading provider of business software. It develops hardware, cloud engineering services, and corporate software such as database management systems and is best known for software products and services like Oracle, and Java. Its headquarters are located in Redwood City, California, USA. Its presence is over in 175 countries. Larry Ellison is the company's founder and was founded in 1977. Its size has grown organically and via acquisitions of other firms. Oracle became the largest Database management company in the world, in 1987. Oracle stood as the world's 3rd largest software company in the world by market capitalization and revenue in 2020.

Today, we're tackling real-world problems with exciting developing technology.

Our work, which ranges from improving energy efficiency to reinventing online commerce, is not only reshaping the world of business but also it's advancing governments, empowering organizations, and equipping billions of people with the skills they need to keep up with change.

The company is perhaps best recognized for its flagship product, the Oracle Database. The firm also creates database development tools and systems, as well as Enterprise Resource Planning (ERP), Enterprise Performance Management (EPM), Customer Relationship Management (CRM), Human Capital Management (HCM), and Supply Chain Management (SCM) software.

1.1.2 Global Business Units (GBU)

In Oracle, a business unit is an entity that autonomously executes one or more business tasks and may be rolled up in a management hierarchy. Global Business Units or GBUs are semi-autonomous business groups that are industry-specific.

There are seven such units in Oracle:

1. Communications Global Business Unit
2. Financial Services Global Business Unit
3. Health Sciences Global Business Unit
4. Construction and Engineering Global Business Unit
5. Hospitality Global Business Unit
6. Retail Global Business Unit
7. Oracle Energy and Water

My team Integrations is a part of Utilities Global Business Unit or UGBU, now renamed Oracle Energy and Water.

1.1.3 Oracle Energy and Water

Energy and Water is a global unit in oracle whose main goal is to thrive in sustainable, affordable energy and water future. This business unit is concerned with software innovations connected to the use of water, gas, and electricity. There are numerous teams under this business unit that produce various software solutions for various needs. This project was created as a proof of concept. This group deals with data collected by smart meters installed in homes. They aim to generate relevant insights that will assist consumers in better understanding how to use the different utilities that they use.

Oracle's Energy and Water provides different solutions such as

1. Customer Service and Billing
2. Advanced Metering
3. Energy Efficiency and Demand Management
4. Customer Experience
5. Network Management
6. Work and Asset Management
7. Field Service
8. Capital Asset Lifecycle Management
9. Enterprise Resource Planning
10. Cloud Infrastructure
11. Human Capital Management

Our project is part of the Innovation Lab initiative by Oracle.

1.1.4 Integrations

We at Integrations develop integration flows that bundle the business flows with Energy and Water applications. It is the process of enabling independently designed applications to work together to solve a business need. It ensures that different applications work together to maximize data consistency and productivity.

1.1.5 Innovation Center

The Innovation Lab is a multi-story building in Chicago, Illinois. Oracle showcases all of its cutting-edge technology created in-house. UGBU has been given an entire floor to demonstrate its technology. A residential set up with water, gas, and electric meter connections, as well as electrical equipment, is present. A client may come into this setup, turn on an appliance, and see real-time feedback from apps created by UGBU teams. Our project collects data from the digital meters in this lab, generates tickets, develops insights, and visualizes it in real. This assignment is a part of Oracle's Energy and Water in connection with Innovation Lab.

1.2 Aim and Objective:

The project aims to develop an efficient solution for maintaining an asset (Wind Turbine).

The key objectives are

1. To generate tickets automatically for the pre-failure events in the wind turbine by monitoring its state continuously.
2. To provide some visualizations to the customers so that they can draw some insights out of it.
3. To optimize repair cost and maintenance cost.
4. To provide a real-time platform to customers that enable them to make educated decisions about the asset.
5. To make the customer better understand their asset.

1.3 Problem severity:

It is a well-known fact that across industries operating costs, and maintenance costs take a significant amount of share. It is concluded that maintenance costs would be more than half of the overall production costs excluding the costs for planned and unplanned downtime, however, production costs would vary from industry to industry. So that’s the reason why industries go the extra mile to tie up with the companies for an optimal solution for monitoring their asset, which reduces the chances of breakdown and improves efficiency.

1.4 Motivation:

Repair costs would be expensive if there is unplanned downtime, i.e when a machine comes offline. It is very vital to keep the machinery running. Therefore to improve production efficiency and allow companies to run their operations without any obstacles an effective maintenance program is needed, which can not only save downtime but also cost. However, all programs are not equally effective, they have their pros and cons. The maintenance program has to be selected in such a way that the cost of the plan and maintenance cost has to be balanced.

It's tough to know how often a machine should be repaired and how to balance the hazards of lost production time vs the risks of a potential breakdown.

2. LITERATURE SURVEY

2.1 Traditional Maintenance Programs:

The traditional Maintenance Programs are broadly classified into two categories based on the time of reaction, which are:

1. **Reactive Maintenance**
2. **Preventive Maintenance**

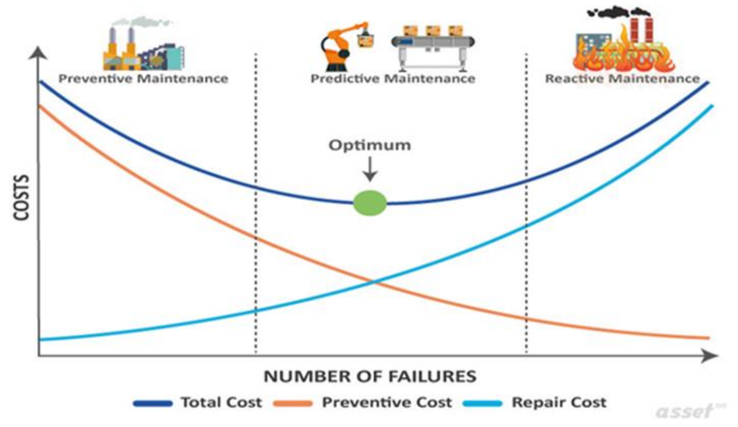


Figure 2.1. 1 Reactive vs Predictive vs Preventive

2.1.1 Reactive maintenance is a strategy that is based on the principle of **run-to-failure**, which means letting the machine run until it breaks and immediate action is needed thereafter. No resources are involved in maintaining the equipment. This was the most widely used maintenance mode till the late 2000s. In this case, utmost priority has to be given, which is one of the drawbacks. The other major drawback with this strategy is the repair cost since the asset would have been damaged completely by the time we react, hence repair and replacement of the device becomes an issue.

2.1.2 Preventive maintenance or planned maintenance is a strategy that includes a regularly scheduled program for each piece of equipment, therefore there is more probability of finding and correcting the issues before it becomes a major issue. The lifespan of the asset would increase, and it also minimizes the unplanned downtime but the drawback with this strategy is that it requires more labor, the maintenance cost is also increased and planned downtime is also high. Though the maintenance cost is high in this strategy the overall cost would be less than that of the reactive maintenance because the life of the asset is prolonged in this type.

Another drawback is the wastage of resources since the schedule is followed though there is life left in the lubricants or parts.

2.2 Modern Maintenance Program

2.2.1 Condition Monitoring (Methodology)

Condition-based maintenance entails keeping track of the status of equipment by monitoring a specific condition in machinery (such as vibration, temperature, and so on) for changes that could indicate a developing defect and performing maintenance as needed.

It's a crucial component of predictive maintenance as condition monitoring enables maintenance to be scheduled and preventative steps to be done to avoid future failure and unscheduled downtime. Traditional condition monitoring relied mostly on vibration analysis, while more modern, innovative systems employ sensors to continuously evaluate several parameters and provide alerts when a change is detected.

You don't have to be concerned about executing condition-based maintenance too early as compared to preventive maintenance. After anything goes wrong, but before it stops working, sensors alert you that maintenance is required. Because you must check your equipment regularly, condition-based maintenance is also called condition-based monitoring. The disadvantage is that you can't plan maintenance because you don't realize you need it until the situation changes.

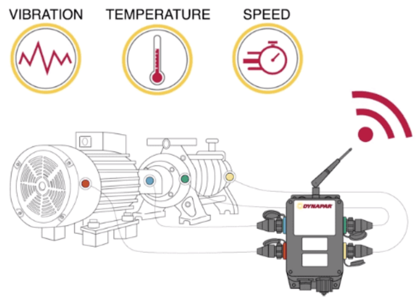


Figure 2.2. 1 Condition-based Monitoring

Advantages	Disadvantages
The maintenance cost is lower	Short-term investment is required.
Increase the life span of machinery	The equipment for condition monitoring is expensive.
Maximizes the production output	The sensors of condition monitoring may not survive depending on the environment.
Perfect ordering of maintenance tasks.	It exhibits unpredictable maintenance periods

Table 2.2. 1 Pros and Cons of Condition Monitoring

2.2.2 Predictive Maintenance

Predictive maintenance combines the benefits of preventative and condition-based maintenance. With this sort of maintenance, a computerized system uses previous data and sensor readings such as temperature, vibration, and noise to forecast when maintenance is required, allowing problems to be repaired before they cause downtime. Predictive maintenance allows you to plan work ahead of time while extending the life of your equipment. Predictive technology, on the other hand, can be expensive upfront.

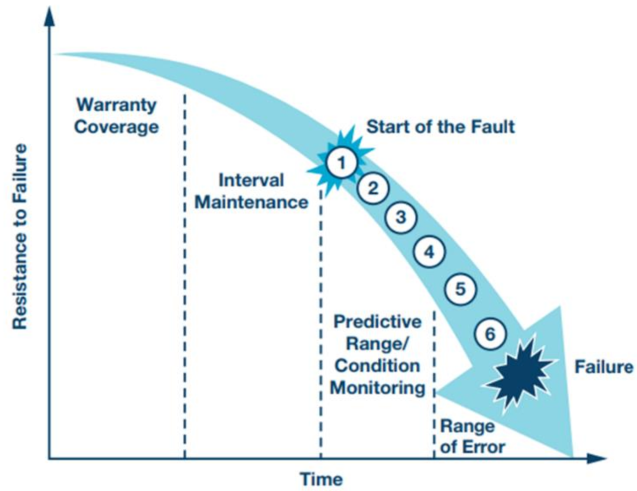


Figure 2.2. 2 Resistance to failure vs Time

Predictive maintenance aims to tackle the problems of reactive and preventive maintenance without causing failure or replacing a working element. Predictive maintenance reduces scheduled and unplanned downtime, excessive maintenance costs, the risk of further asset damage, and wasteful maintenance on operational assets.

To put it another way, predictive maintenance uses data to make better maintenance decisions, provide more insight into asset condition, and improve system connections.

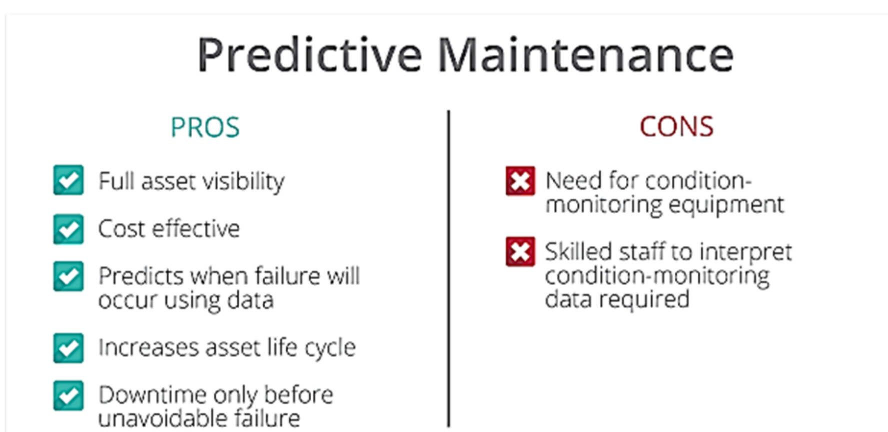


Table 2.2. 2 Pros and Cons of Predictive Maintenance

2.3Market and the potential opportunities in Condition Monitoring

With the evolution of IIOT (Industrial IoT), the Condition Monitoring market is expected to reach 3.2B by 2023 and 3.6 billion by 2026 with a CAGR (compound annual growth rate) of 7.1% from 2021 to 2022, and the predictive maintenance market is expected to reach USD 15.9 billion by 2026 with a CAGR of 30.6%.

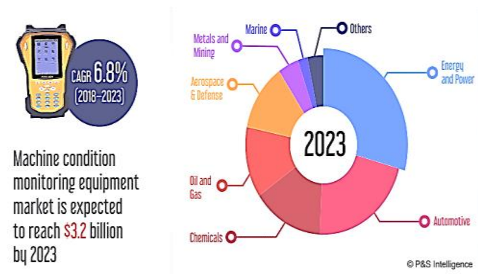


Figure 2.3 2 Market of Condition Monitoring

e: estimated; p: projected

Source: Secondary Research, Expert Interviews, and MarketsandMarkets Analysis

Figure 2.3 1 Potential Opportunities in Predictive Maintenance Market

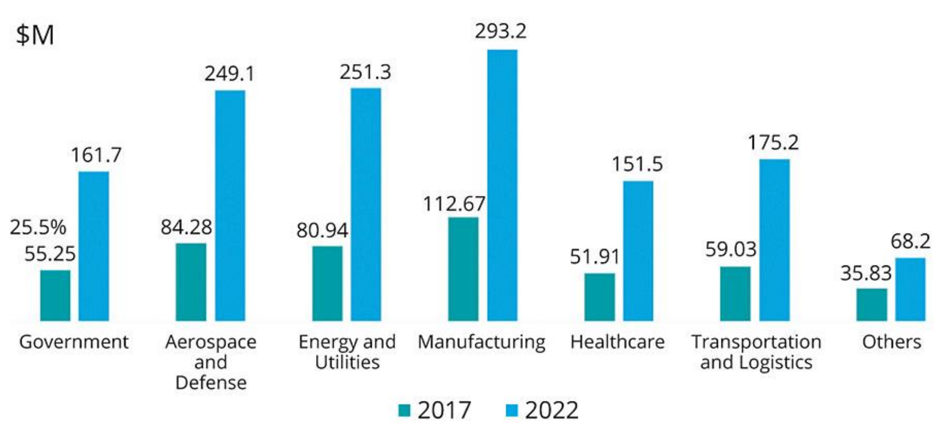


Figure 2.3 3 Growth of Predictive Maintenance from 2017-2022

2.4 Comparing the maintenance programs

Condition-Based Maintenance	Predictive Maintenance
Some types relies on data	Relies on data
Human defines decision-rule	Data defines decision-rule
Static decision rule	Dynamic decision rule
Tells if something is wrong here and now	Predicts failures in the future
Can lead to excessive maintenance	Can be used for just-in-time maintenance
Sensitive to noise	Less sensitive to noise
Preventive approach	Predictive approach

Table 2.4 1 Condition Based vs Predictive Maintenance

Positive Feature	Reactive	Preventive	Predictive
Low Capital Cost	✓	✓	
fewer maintenance People	✓		✓
Increased Component Life Cycle		✓	✓
Energy savings		✓	✓
Increased Plant up-time		✓	✓
Optimized maintenance activities			✓
Better product quality			✓
Improved personnel safety			✓

Table 2.4 2 Comparing all the Maintenance Programs

The conclusions which we can draw from the above tables are that each maintenance program has its pros and cons, and the type of program should be chosen based on the budget, type of asset, etc. Generally, a hybrid one is chosen. It is observed that the most effective maintenance programs are mostly < 10% reactive, 25-35% preventive and 45-55% predictive. The final goal of a maintenance program is to ensure that services are constantly up and running smoothly.

2.5 Industrial Maintenance Statistics

Let's gain some insights by hearing from the industries about their maintenance strategies, future goals, and learnings from their mistakes in the past. According to Industrial Maintenance Status, Trends and Forecasts Report, which is a Plant Engineering survey published in 2021 in partnership with Advanced Technology Services (ATS) states that according to 80% of the respondents the combination of machine breaking down, machine downtime, and aging of the equipment affects the productivity negatively. ATS is a global solution provider with a proven experience of 30 years in asset management and technology-driven industrial maintenance.

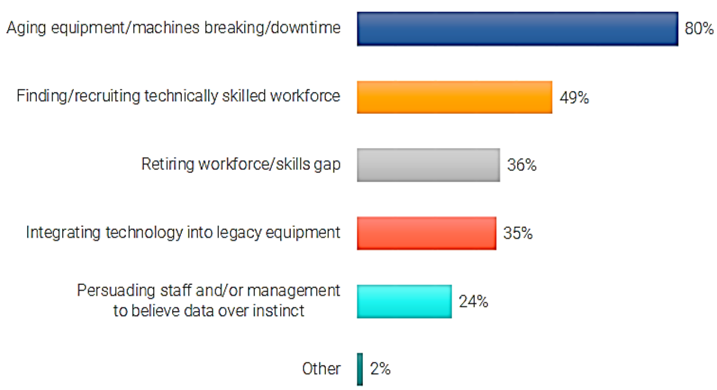


Figure 2.5 1 what affects plant Productivity?

It also states that 34% of unplanned downtimes are due to the aging of equipment, which can be minimized by preventive type maintenance. 6% are due to lack of maintenance strategy failure. 9% are due to lack of time to perform maintenance, it's the case with the reactive type of maintenance. This tells how efficient our maintenance program should be.

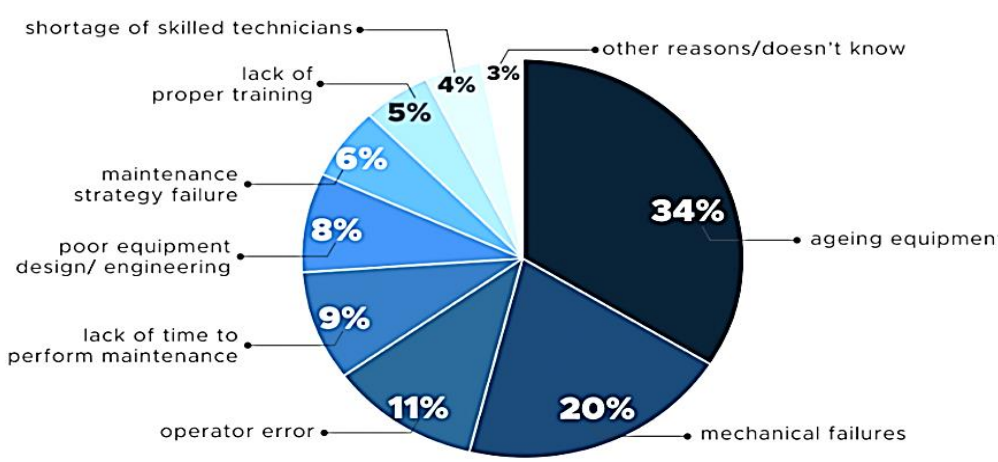


Figure 2.5 2 Plant Engineering Industrial Maintenance Study about unplanned downtime

Some of the findings, which I found useful in the Report, are:

a. 93% of companies consider their maintenance processes are not very efficient
b. 44% of plants spend more than 40 hours a week performing maintenance
c. 29% of plants spend 5 to 10% of their annual budget on maintenance; 24% spend 11 to 15%, and 17% spend more than 15%.
d. In 2021, 42.5% of companies in the US spent 21-40% of their operating budget on cleaning and maintenance equipment/supplies.

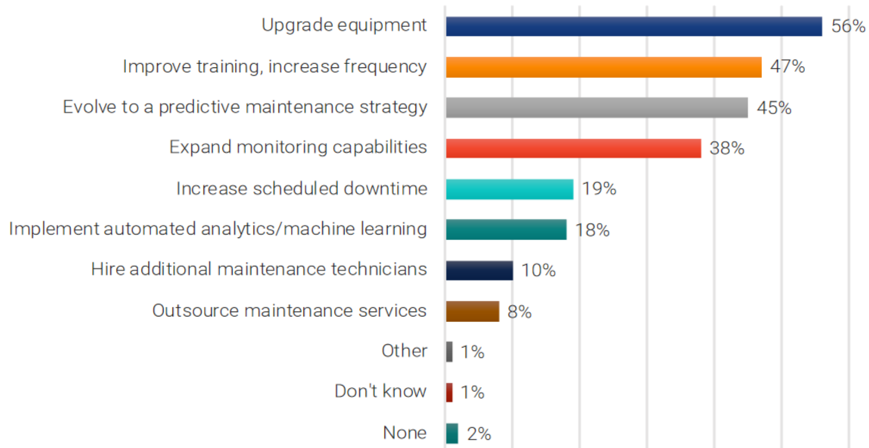


Figure 2.5 3 Plans to decrease Unscheduled downtime

These were the methods the industries are planning to adopt in the future to decrease the unplanned downtime of the asset. Most of them would like to implement automated analytics/Machine Learning monitoring and would like to switch to a predictive type of maintenance.

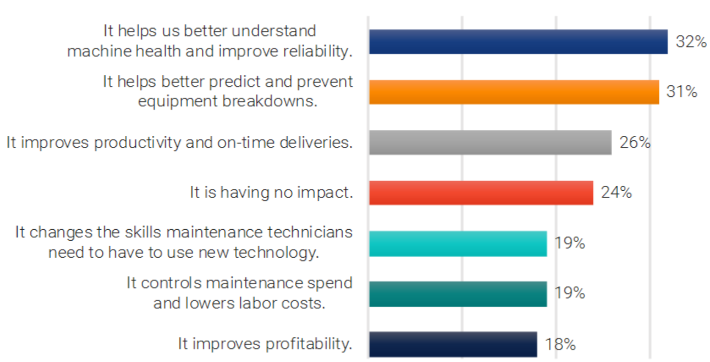
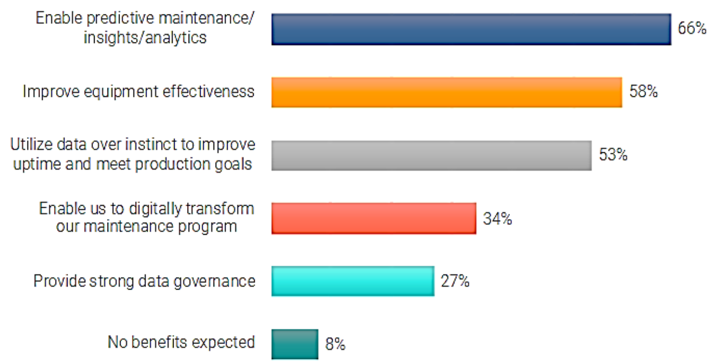


Figure 2.5 4 Impact of IIOT on Maintenance Operations

Most companies agree that the evolution of Industrial IoT improved productivity, maximized profits, and helped in a better understanding of the health of the asset.



Most industries are planning to take a technological move, 38% are expected to implement data analytics for making educated decisions, 36% are targeting remote/sensors monitoring, and 25% wish to move to CMMS (Computerized Maintenance Management System)

Figure 2.5 5 Benefits of Sensors in Maintenance

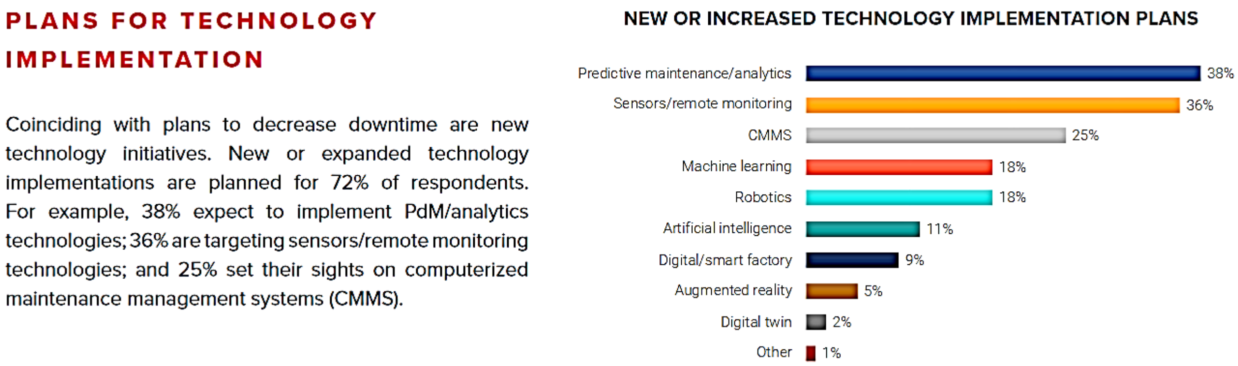


Figure 2.5 6 Technology Implementation Plans

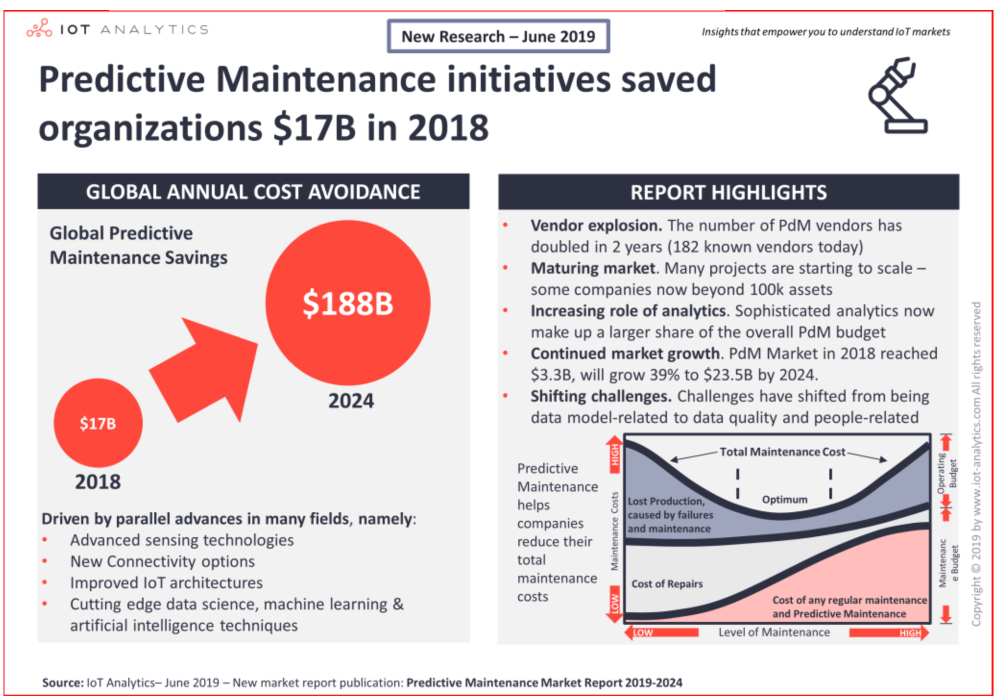


Figure 2.5 7 Report on Predictive Maintenance

3. Propose Solution

3.1 High-level view

Simply put, we collect Wind Turbine sensor data in the cloud and send it to an analytics platform, where it is constantly monitored for changes. When parameters such as Machine temperature, Hub Rotational Speed, and Tower Vibration deviate from their normal behavior, an alert is automatically generated and the user is notified of the impending failure. We also provide valuable insights through visualization.

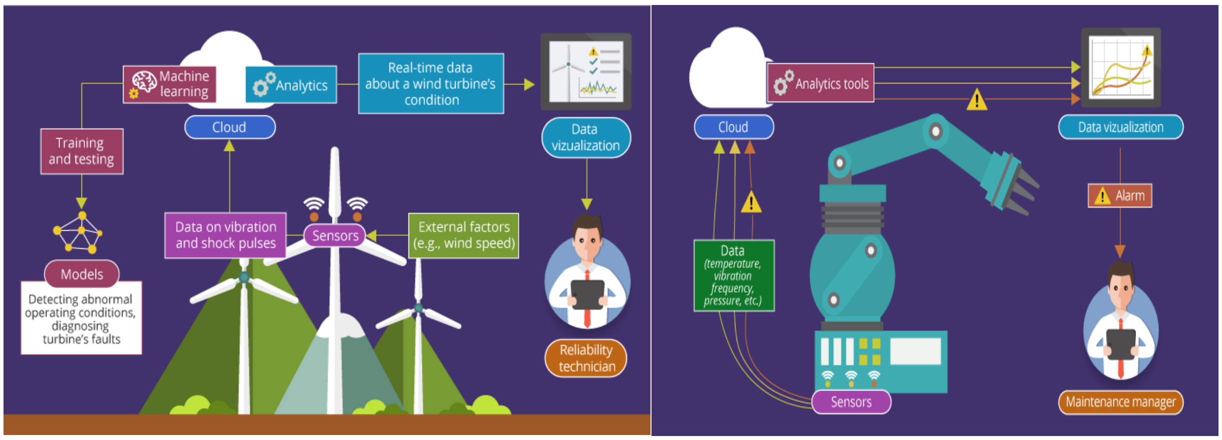


Figure 3.1. 1 High-Level view of the Solution

3.2 Block Diagram

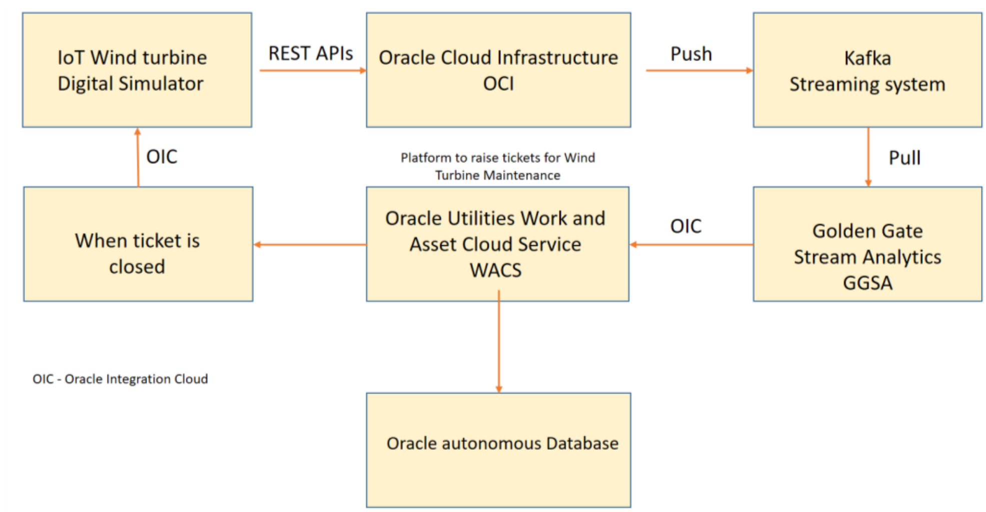


Figure 3.2 1 Block Diagram

3.3 Tools & Technologies used

We used many tools to build this solution; first, I'd like to explain the tools and technologies that we used.

3.3.1 Oracle IoT Digital Twin Simulator

First of all, we need a source, the data to work on right? Since it's hard to directly develop and test using a real-time device (wind Turbine), we have gone for its digital twin.

3.3.1.1 What is a Digital Twin?

A digital twin is a digital representation of a physical product, service, or activity. It's a virtual representation of a real-world object. It's a computer program that analyses real-world data to construct simulations to forecast how a gadget would operate. Oracle has applied the IoT digital twin idea very comprehensively.

[1] Without having to connect or set up hardware, the IoT digital twin simulator allows you to create simulated assets for your environment. These emulated devices can generate customizable live data, alarms, and events. You can construct simulation models and then use them to create simulated device instances using the IoT digital twin simulator. The IoT digital twin simulator also includes in-built simulation models that you may employ to develop simulated devices.

Simulated devices function like real devices: you may control and monitor them in real-time, turn them off and on, and configure them to generate alerts and events. Using these simulated devices you can test your applications and perform some analytics. In your simulation models, you can configure the device properties, alerts, and actions as well.

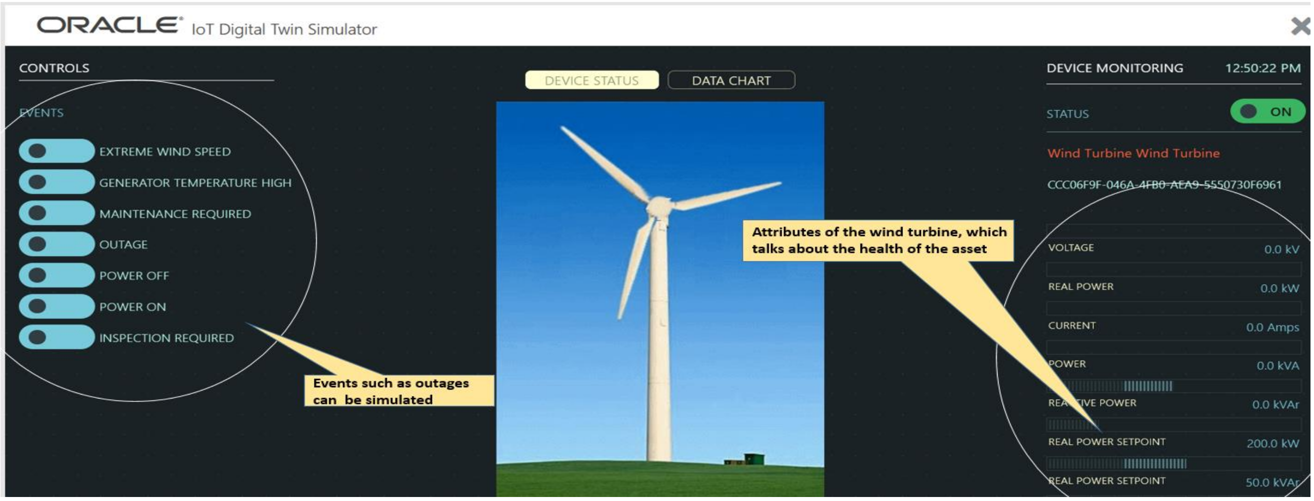


Figure 3.3. 1 Iot Digital Twin Simulator dashboard

The data generated from the IoT devices is continuous and changes from time to time, once a particular period has crossed data may be useless, so it has to be processed as early as possible for better results, Now we need a low latency stream analytics platform to process the wind Turbine data.

3.3.2 Golden Gate Stream Analytics (GGSA)

3.3.2.1 What is Streaming analytics?

It simply means processing and analyzing the streaming data rather than in batches. It is also known as event-stream processing because they are triggered by a specific event such as IoT devices, machine failure, or any such measurable activity. Using streaming analytics platforms, organizations can extract business value from streaming data in the same way that traditional analytics solutions do for data at rest, but first of all what's streaming data?

3.3.2.2 What is Streaming Data?

Streaming data is large volumes of current and **in-motion** data that are generated continuously by hundreds of data sources, which typically send data records at the same time. The information might come from the Internet of Things (IoT), cloud applications, Sensors, etc. A streaming platform must deal with this constant inflow of data and analyze it sequentially and progressively.

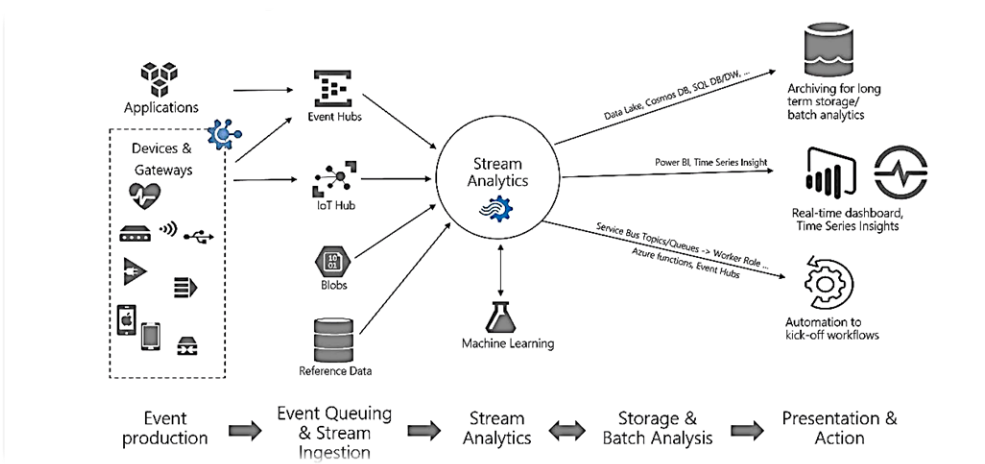


Figure 3.3. 2 Stream Analytics Architecture

The Oracle GoldenGate Stream Analytics (**GGSA**) runtime component is a comprehensive platform for developing real-time event filtering, correlation, and processing applications. It proves to be a versatile, high-performance event-processing engine with various deployment options of stand-alone Spark or Hadoop-YARN. Fast Data and the Internet of Things (IoT) are enabled by GGSA, which delivers actionable insight and maximizes value on enormous volumes of high-velocity data from a variety of data sources in real-time. By bringing business logic to the network edge, it provides dispersed intelligence and low latency responsiveness.

The deployment option which we have chosen is Spark as it is 100 times faster than Hadoop for large scale data processing

3.3.2.3 About Apache Spark

Apache Spark is a big data and machine learning analytics engine that runs at incredible speeds. It was created in 2009 at the Amp Lab at the University of California, Berkeley. Later, the Apache Software Foundation acquired the Spark codebase and now it's the Data processing's biggest open source project.

Since its release, Apache Spark, the unified analytics engine, has seen rapid adoption by enterprises across a wide range of industries. Internet powerhouses such as Netflix, Yahoo, and eBay have deployed Spark at a massive scale, collectively processing multiple petabytes of data on clusters of over 8,000 nodes. It has quickly become the largest open source community in big data, with over 1000 contributors from 250+ organizations.

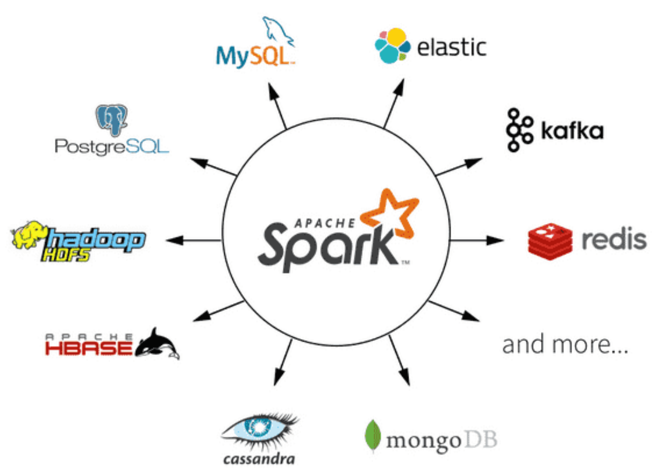


Figure 3.3. 3 Connectors in Apache Spark

In an Apache Spark-based system, GoldenGate Stream Analytics (GGSA) enables the building of custom operational dashboards that provide real-time monitoring and analysis of event streams. Customers can use it to detect events of interest in an Apache Spark-based system, run real-time queries against those event streams, and use the results to generate operational dashboards or raise warnings.

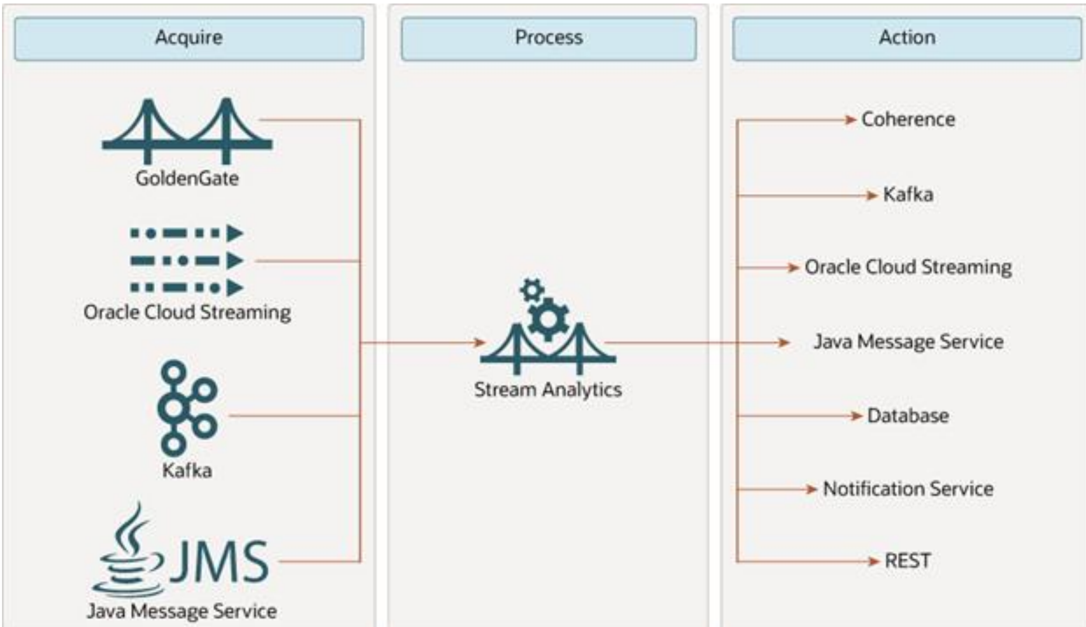


Figure 3.3. 4 GGSA Connectors

The different artifacts in GGSA are:

- Connections – These are the basis for the creation of sources.
- Stream – It's the data in motion, which is flowing.
- Pipeline – Defines the logic and consists of a sequence of stages, it starts with a stream and ends with a target.
- Target – It's the connection to an external system, where the results are stored after processing the stream.
- Dashboard – Shows inter-related visualizations, it's a place from where clients can gain valuable insights.
-

GGSA allows to stream data to it only in 4 ways; GoldenGate, JMS (Jave Messaging Service), File, and Kafka. We have chosen Kafka to stream data into GGSA.

3.3.3 Why and What is Kafka?

Kafka is trusted and used by more than 80% of Fortune 100 enterprises. Thousands of enterprises rely on Kafka, an open-source distributed event streaming platform, for streaming analytics, data pipelines, data integration, etc. As a replacement for a more traditional message broker, Kafka works well. Message brokers are employed for several purposes (to decouple processing from data producers, to buffer/log unprocessed messages, etc). Kafka offers higher throughput, built-in partitioning, replication, and fault-tolerance than most messaging systems, making it a viable choice for large-scale message processing applications. It is based on the publisher-subscriber model.

Kafka can be readily integrated with a variety of large data processing frameworks such as Apache Spark and Apache Storm.

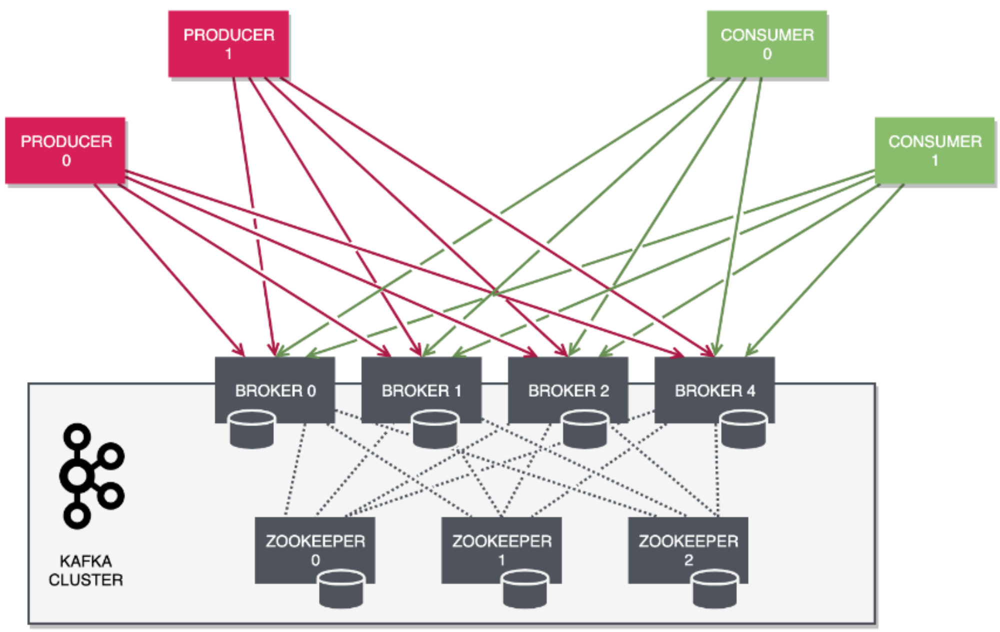


Figure 3.3. 5 Kafka Architecture

Figure 3.3. 6 Famous Firms using Kafka

Many businesses have embraced Apache Kafka in recent years. It is one of the most widely used tools on the market. Kafka is supposedly used by over 20,500 enterprises, including LinkedIn, Facebook, Uber, Shopify, and Spotify. Kafka is frequently used in the big data domain as a reliable mechanism to ingest and transport enormous volumes of data streams extremely quickly due to its fault tolerance and scalability.

The absence of Kafka Creates a Strong coupling between sender and receiver, fault tolerance makes it more cumbersome, replacing nodes difficult, and doesn't scale well

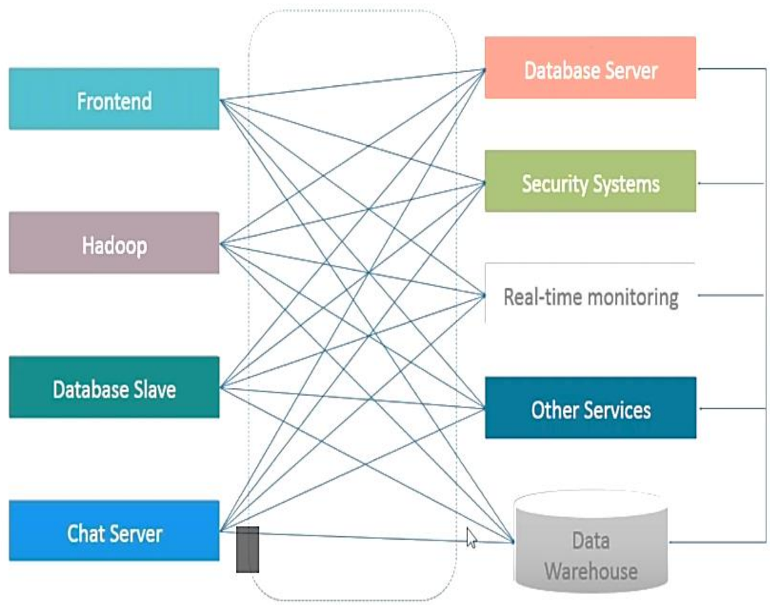


Figure 3.3. 7 Coupling in absence of messaging system

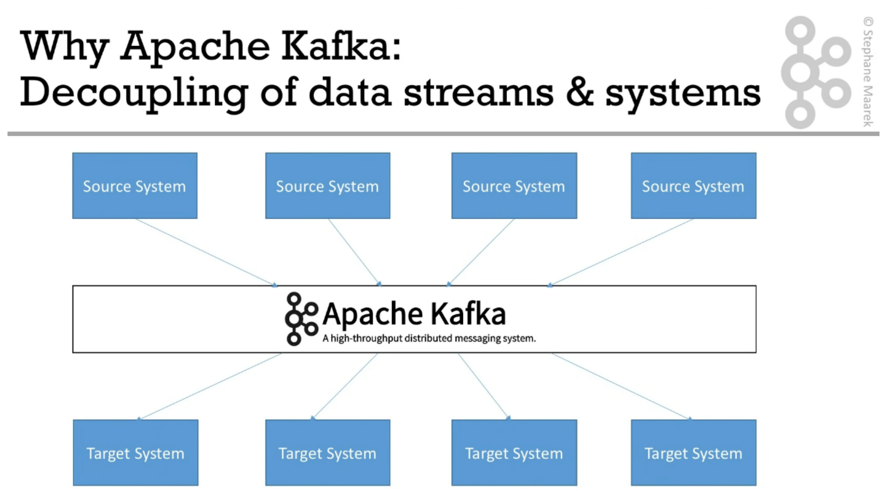


Figure 3.3. 8 Kafka decouples source and target

Advantages of Kafka:

- Distributed
- Resilient architecture
- Fault-tolerant
- High performance
- Latency < 10ms
- Used by 2000+ firms of the fortune 500
- Highly scalable
    - Can scale up to 100s of brokers
    - Can scale to millions of messages per sec

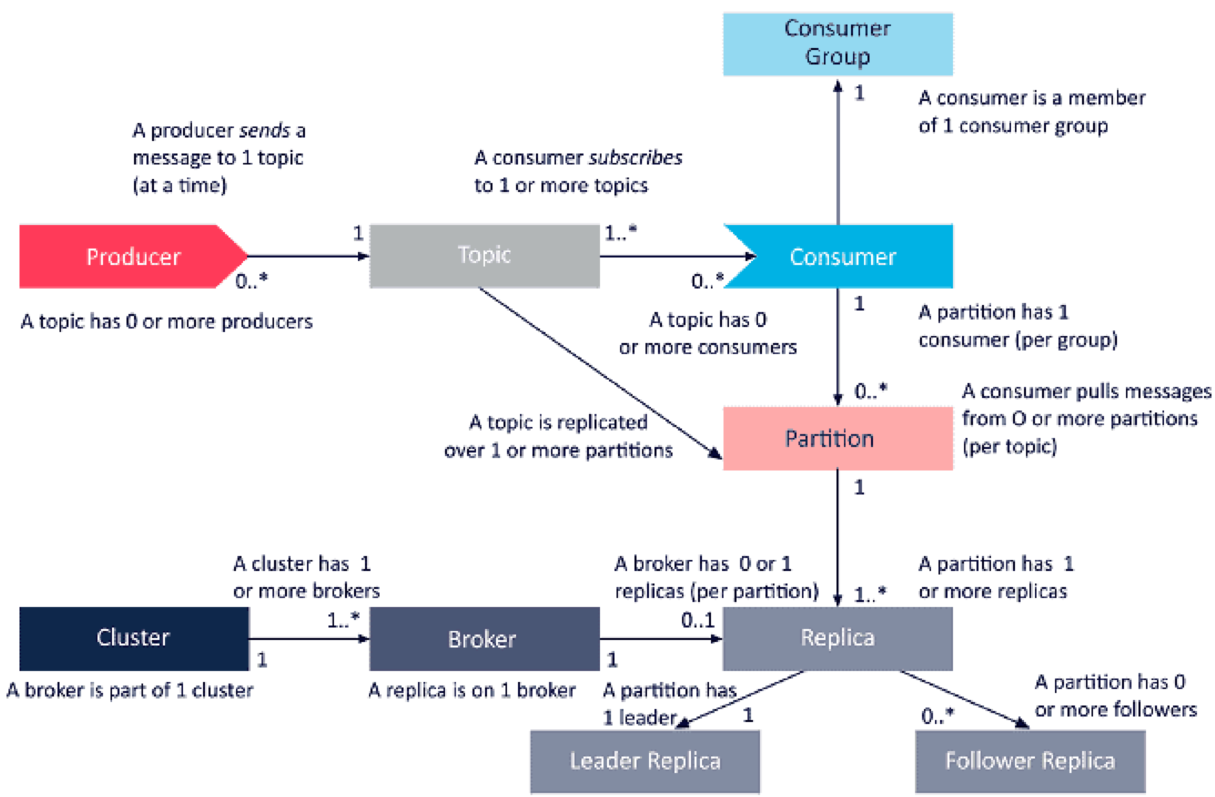


Figure 3.3. 9 Kafka Internal Working

Oracle IoT digital twin simulator doesn't support Kafka, so we need a mechanism to fetch sensor data from it to the cloud and then publish it to Kafka. One of the simple solutions is to use the web services as the platform provides REST APIs which allow you to run Oracle IoT Cloud Service functions and services from a remote location.

3.3.4 Need for Web Services

Web apps must be able to connect through the Internet with each other. Because all Internet browsers and servers support HTTP, it is the best mechanism to communicate between apps. SOAP and REST were established for this purpose.

Any piece of software that makes itself available via the internet and employs a standardized XML messaging protocol is referred to as a web service. All communications to a web service are encoded in XML. A client might, for example, invoke a web service by sending an XML message and then waiting for an XML answer. Web services are not limited to a single operating system or programming language because all communication is done over XML. Java can communicate with Perl, and Windows applications can communicate with Unix applications. This feature is called interoperability.

Using open standards like HTML, XML, WSDL, and SOAP, a web service allows communication between different applications. For a web service

- To tag the data, you'll need to use XML.
- To send a message, use SOAP (Simple Object Access Protocol).
- The availability of a service is described using the WSDL.

On Solaris, you can create a Java-based web service that can be accessed from a Windows-based Python program.

3.3.4.1. What is SOAP?

SOAP allows applications running on multiple operating systems, using different technologies and programming languages, to communicate with one another.

The World Wide Web Consortium and its member editors developed the Simple Object Access Protocol (SOAP), a messaging standard. SOAP declares its request and response messages in an XML data format, relying on XML Schema and other technologies to regulate the structure of its payloads.

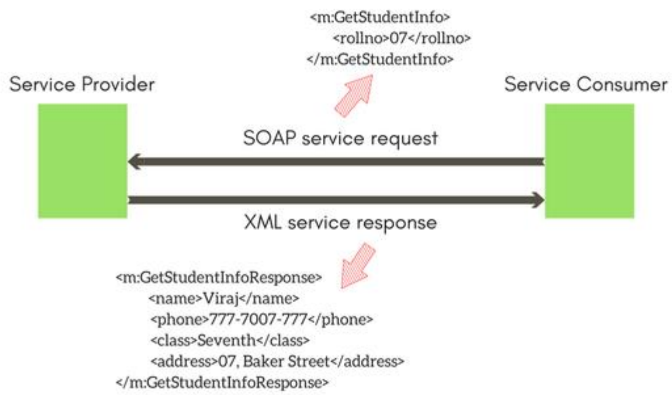


Figure 3.3. 10 Soap request-response

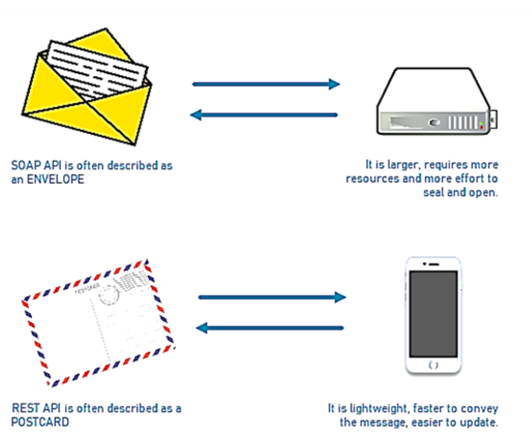


Figure 3.3. 11 SOAP vs REST

3.3.4.2 What is REST?

Representational State Transfer (REST) is an architectural style that specifies a set of restrictions for developing web services. REST API is a simple and flexible method of accessing web services that do not require any processing.

Because REST requires less bandwidth and is simple and adaptable, it is more suitable for internet use than the more robust Simple Object Access Protocol (SOAP). It is used to get or send data from a web service. Only HTTP requests are used in all REST API communication.

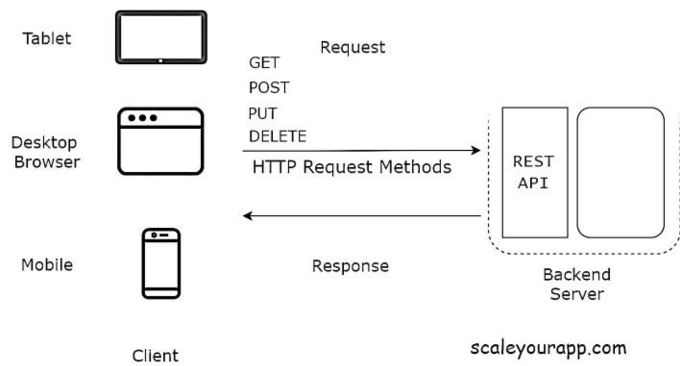


Figure 3.3. 12 REST API Architecture

Client-server communication is a stateless operation. That is, each communication between the client and the server is treated as if it were the first time. There is no information or recollection from earlier communications passed over. As a result, every time a client interacts with the back-end, the client must also communicate the authentication information to it. The back-end can then determine whether or not the client is authorized to view the data.

When a REST API is implemented, the client talks with the back-end endpoints. This completely separates the back-end and client code. The URL of the service that the client could access is referred to as an API/REST/Back-end endpoint. For example, https://myservice.com/users/{username} is a back-end endpoint for retrieving the user details of a specific user from the service.

We also need an environment to develop and test APIs, for that, I’ve used postman and Soap UI.

3.3.4.3 What is Postman?

Postman is the most popular API platform in the world. Postman's capabilities help you create better APIs faster by simplifying each step of the API development process and streamlining communication. Today, Postman is used by over 20 million developers and 500,000 companies all around the world.

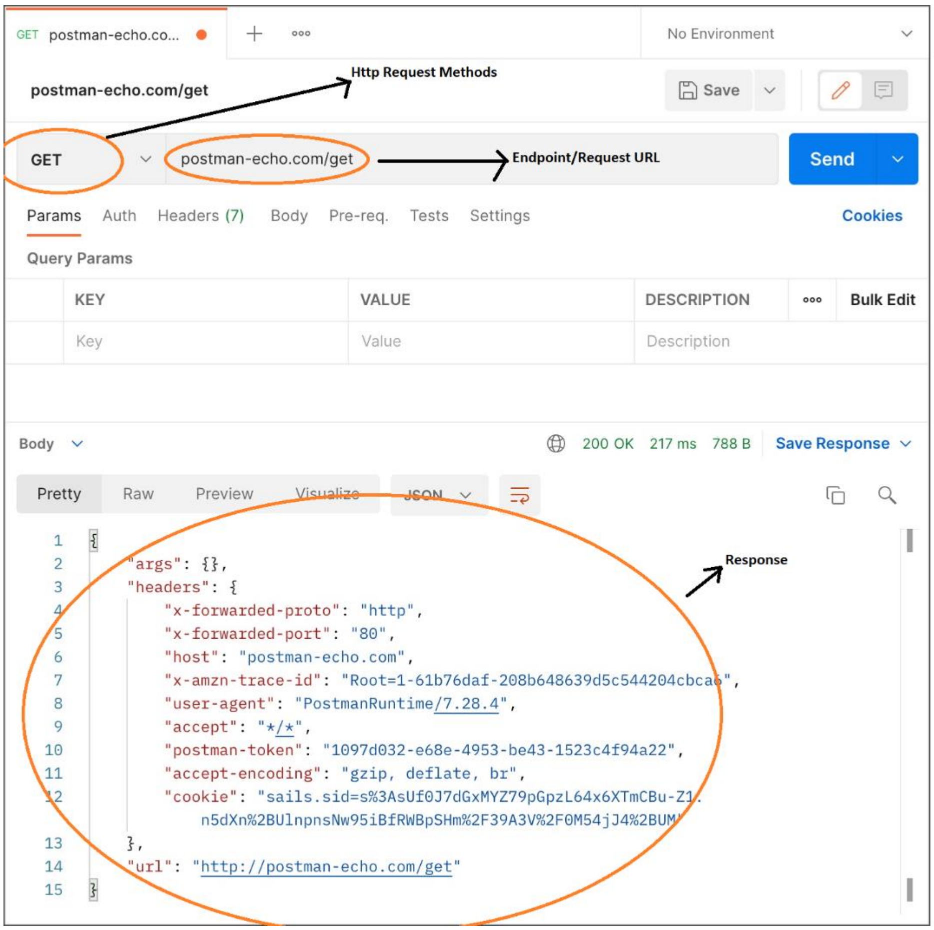


Figure 3.3. 13 Postman UI

3.3.4.4 Need of Soap UI?

For developers and testers who want to speed up the delivery of SOAP, REST, and GraphQL APIs. The quickest and most straightforward method to get started with API testing is with SoapUI.

Now, we need a cloud to

1. deploy a database, to store the final results and we have chosen the oracle autonomous database.
2. Deploy GGSA application
3. To fetch the sensor data from Oracle Internet of Things Cloud Service and push it to Kafka.
4. Even the Oracle Internet of Things Cloud Service, Oracle Integration Cloud (OIC) is deployed in OCI.

So we have chosen OCI for that purpose and it full fills all our needs.

3.3.5 Oracle Cloud Infrastructure (OCI)

The next-generation cloud is designed to run any application, faster and more securely, for less. Learn how to define a robust cloud strategy, plan for successful workload migration, and ensure complete manageability of cloud environments with OCI's Cloud Adoption Framework. Oracle Cloud Infrastructure has all the services you need to migrate, build, and run all your IT, from existing enterprise workloads to new cloud-native applications and data platforms.

Oracle Cloud is the first public cloud built from the ground up to be a better cloud for every application. By rethinking core engineering and systems design for cloud computing, we created innovations that accelerate migrations, deliver better reliability and performance for all applications, and offer the complete services customers need to build innovative cloud applications.

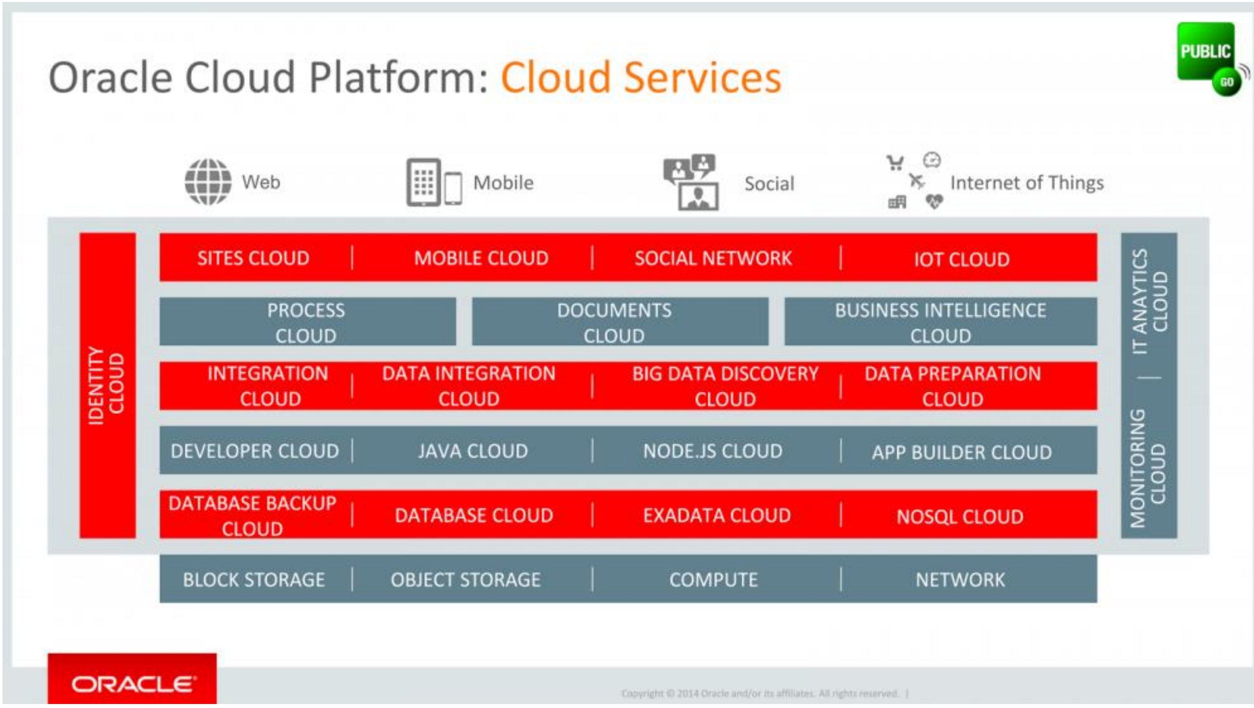


Figure 3.3. 14 Services Provided by OCI

3.3.6 Oracle Autonomous Database

Databases are necessary for the successful operation of modern businesses since they hold crucial business data. Database administrators (DBAs) are frequently overloaded by the time-consuming manual operations of database management and maintenance. Current workload demands might result in DBA errors, which can have a devastating effect on availability, performance, and security.

A cloud database that employs machine learning algorithms to automate database tasks such as database tuning, checking security policies, regular data backup, regular updates, and other routine administrative chores usually performed by DBAs is known as an autonomous database. An autonomous database, unlike a traditional database, can handle all of these activities and more without the need for human interaction.

We have used WACS for generating maintenance tickets for the Wind Turbine when an event is detected. WACS is the cloud version of the WAM, functionality-wise both are one and the same.

3.3.7 Oracle Utilities Work and Asset Cloud Service (WACS)

Oracle Utilities Work and Asset Management (WAM) is a leading asset, work, and supply chain management software system that addresses many mission-critical functions for utilities. The product is an ideal workflow solution because it supports a wide range of business functions and users within an organization.

Work processes are supported through the convenient routing of information that enables staff to accomplish increasing work demands despite decreasing resources. Offering out-of-the-box tools such as asset condition scoring and our generic map viewer WAM enables utilities to minimize their operational and maintenance costs and optimize their capital expenditures. To assist with capital work WAM offers construction work management tools such as work estimating and compatible units that are intricately tied to our regulatory accounting module. Also, WAM's extensive cost-tracking and operational accounting capabilities empower executive and operations management with the financial insight they require to make strategic business decisions.

The application consists of more than 100 integrated modules that provide features and functionality to support mission-critical functions such as:

1. **Assets Management** - manage all types of assets, measurements, and components.
2. **Resources Management** -manage timesheets, crews, equipment, etc.
3. **Work Management** - is where all the work is requested, entered, researched, planned, tracked, and completed.
4. **Purchasing Management** - manages your vendor data and enables utilities to plan, initiate, and process purchases and purchasing contracts.
5. **Inventory Management** - manage the receiving process, the transferring of stock items between storerooms, and requests for materials

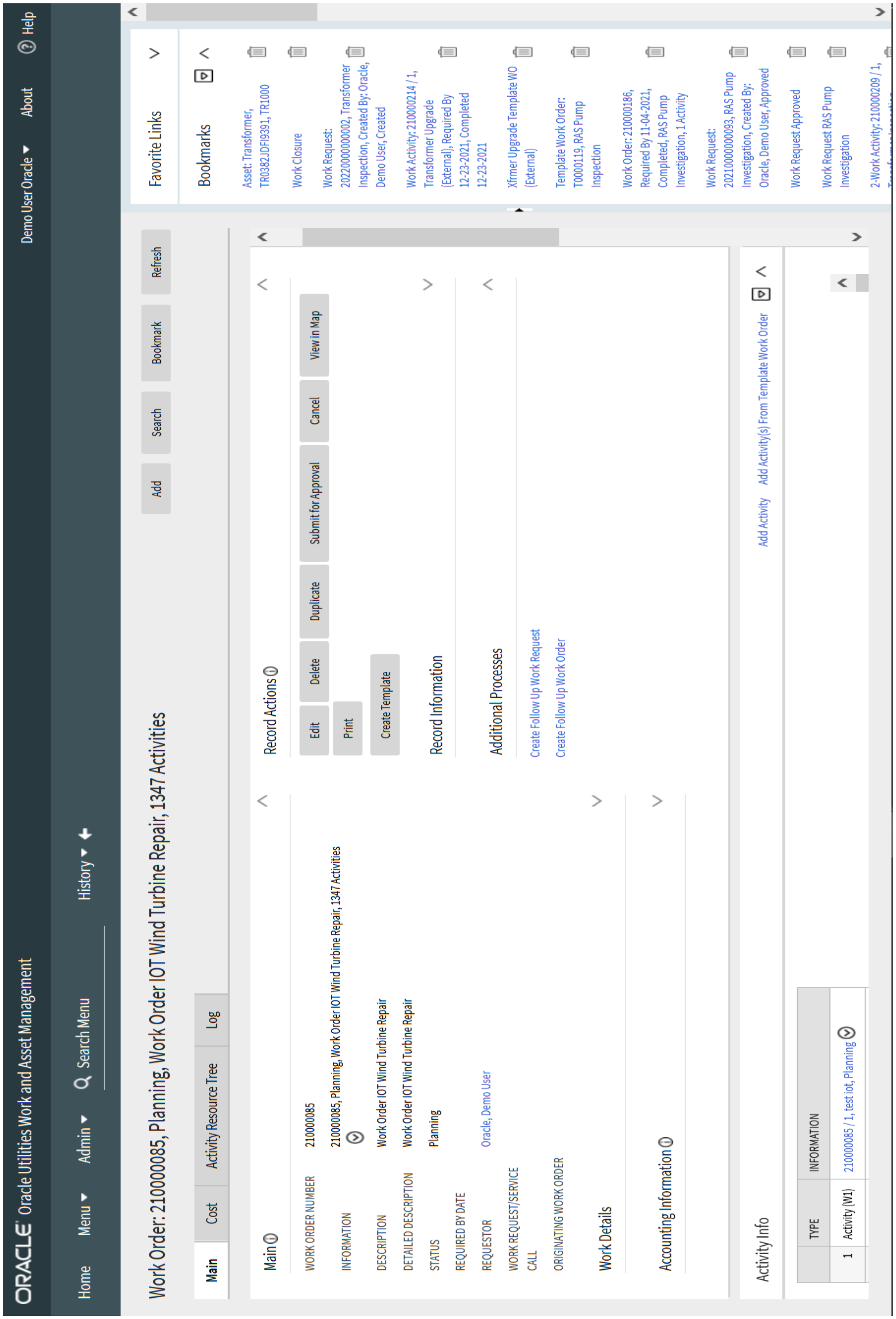


Figure 3.3. 15 WAM Dashboard

Oracle Internet of Things Cloud Service, GGSA, and WACS are different SaaS applications. To Connect different Saas application integrations would be helpful.

3.3.8 Oracle Integration Cloud (OIC)

3.3.8.1 What is Integration?

Integrations are links between applications and systems that allow them to share information and data as a whole. Integrations are based on APIs (application programming interfaces) and allow data to flow between apps, integrating your software so that it can all work together.

In OIC, Integrations allow us to create a link between each source and target system using the predefined/customized application adapters, and then define orchestration processes that allow data to be sent between various systems.

Oracle Integration is a platform for connecting and automating enterprise applications, business processes, APIs, and data. With a visual development experience, prebuilt connections, and embedded best practices, developers and cloud architects can link SaaS and on-premises applications six times faster.

OIC stands for Oracle Integration Cloud. Oracle Integration Cloud Service (ICS) is a cloud-based integration solution that is meant to integrate apps in the cloud but also includes features to integrate with in-house systems. OIC is a PaaS service is available from Oracle. Based on the excellent Oracle Service Bus application, OIC is intended to be more user-friendly and to open a better approach to integration creation. All settings are browser-specific and the XSL data mapping/processing process is utilized.

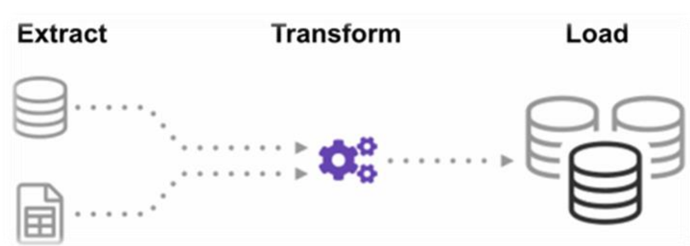


Figure 3.3. 16 ETL - Data Integration Process

3.4 System Design

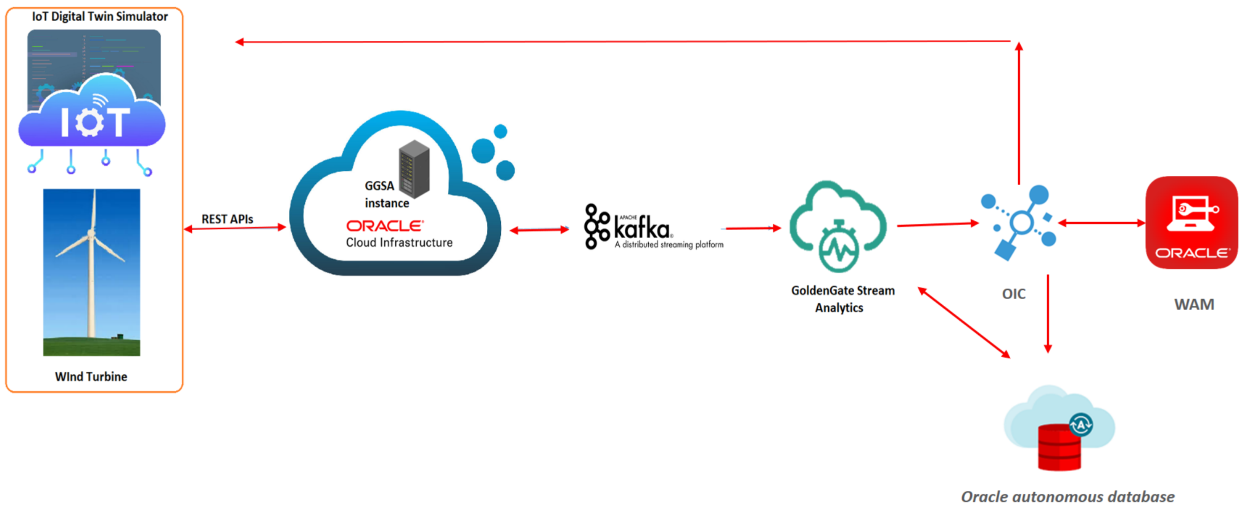


Figure 3.4. 1 System Design

The steps, which I've followed in building the solution are:

Initial Setup:

1. Create and Test the Kafka environment in the cloud. [5]. Refer Appendix.
2. [9] Creating an autonomous database in OIC and accessing it from SQL developer using wallet [12].
3. Creating connections and targets in GGSA [7].
    a. Connections required:
        - Kafka (localhost)
    b. Targets Required:
        - REST
        - Database Table

Figure 3.4. 2 Connections and Targets in GGSA

Procedure:

1. First of all, I've fetched the wind Turbine's attributes to the cloud using REST APIs.
2. REST APIs were developed and tested in the postman tool.
3. Now, in the cloud, I've written a python script to fetch the attributes from the Oracle Internet of Things Cloud Service of the wind turbine and push them to Kafka.
4. To make this process recurring and run in the background as a child process I've written a bash script.
5. Then I've designed a data pipeline in GGSA to pull data from Kafka as a stream, check for the pre-defined conditions, and trigger OIC integration if there are any pre-failure events detected.
6. The pipeline also plots some visualizations such as average power output per minute, per hour, and day. It also calculates maximum power output and windspeed per x minutes.
7. OIC Integrations was designed in such a way that it creates an activity in WAM as soon as it is called and brings down the IoT device. As soon as the ticket is closed it brings the device up using Rest services. Activities are only created if the status of the previous similar activity is completed, using this we have overcome the creation of duplicate activities.
8. It also alerts the client about the event using mail services and also stores the occurrence of the event along with their time of occurrence, and WAM activity Id in the database.
9. OIC integrations were also tested using Postman and SoapUI.
10. In OIC we have different adapters for connecting to different SaaS applications. For connecting to WAM we have used a utilities adapter and database adapter for connecting to the Oracle autonomous database.

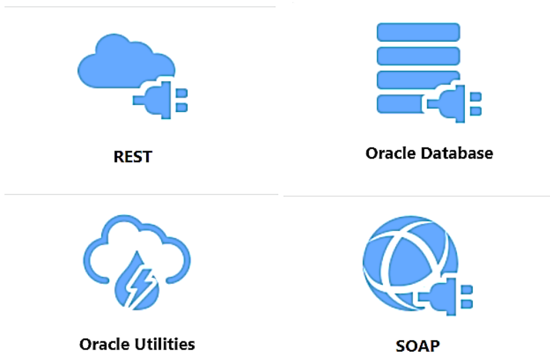


Figure 3.4. 3 OIC Adapters Used

3.5 Results

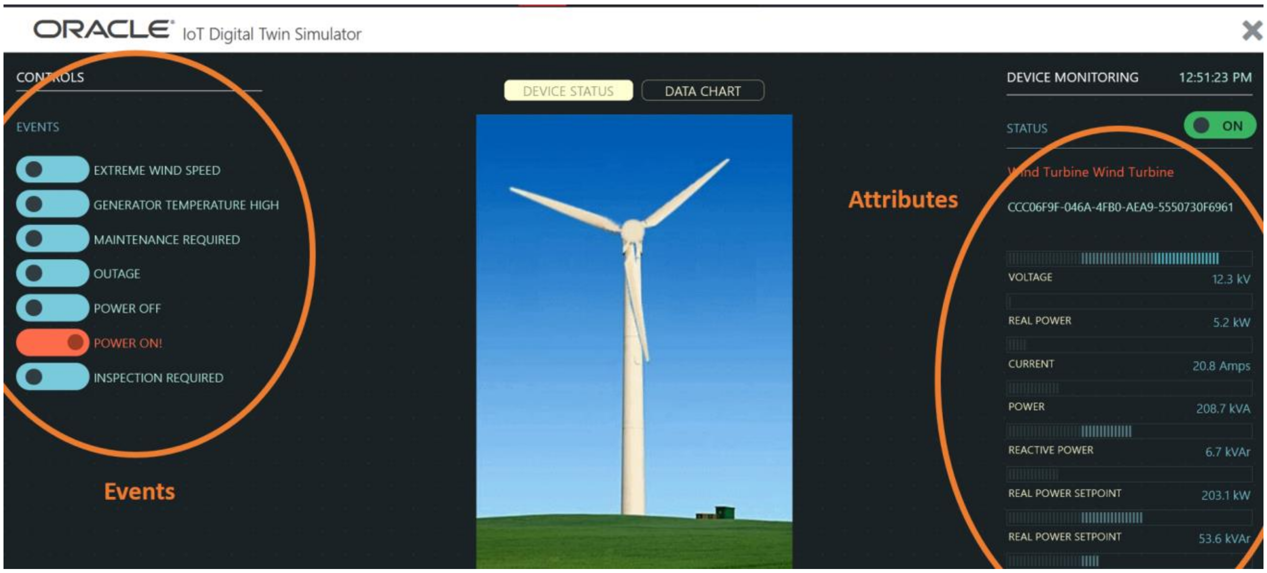


Figure 3.5. 1 IoT Dashboard

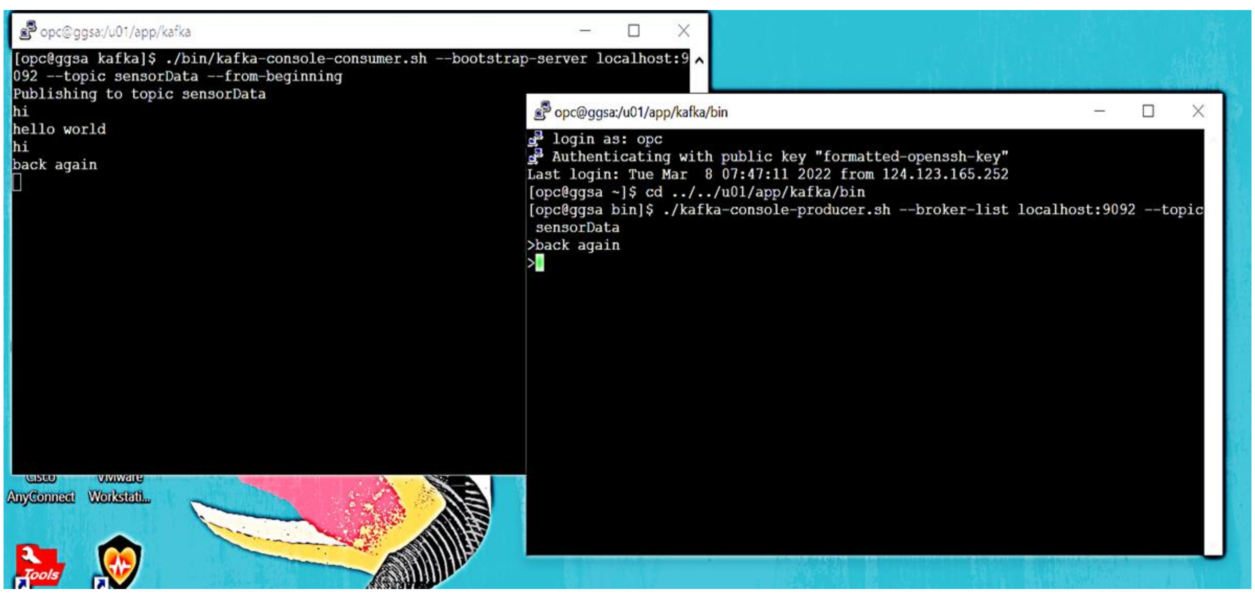


Figure 3.5. 2 Testing Kafka in OIC GGSA instance

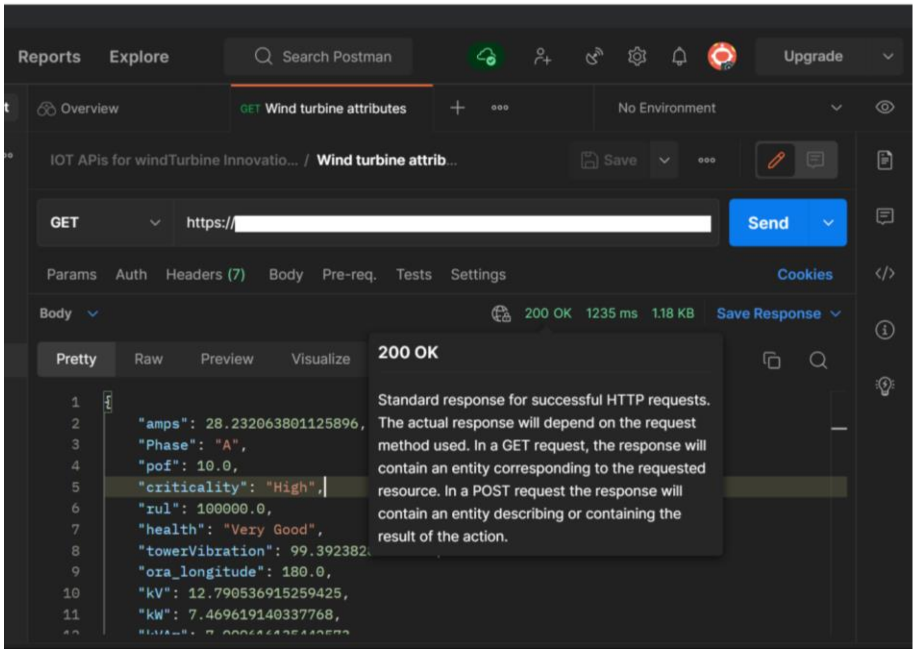


Figure 3.5. 3 API to fetch wind turbine attributes

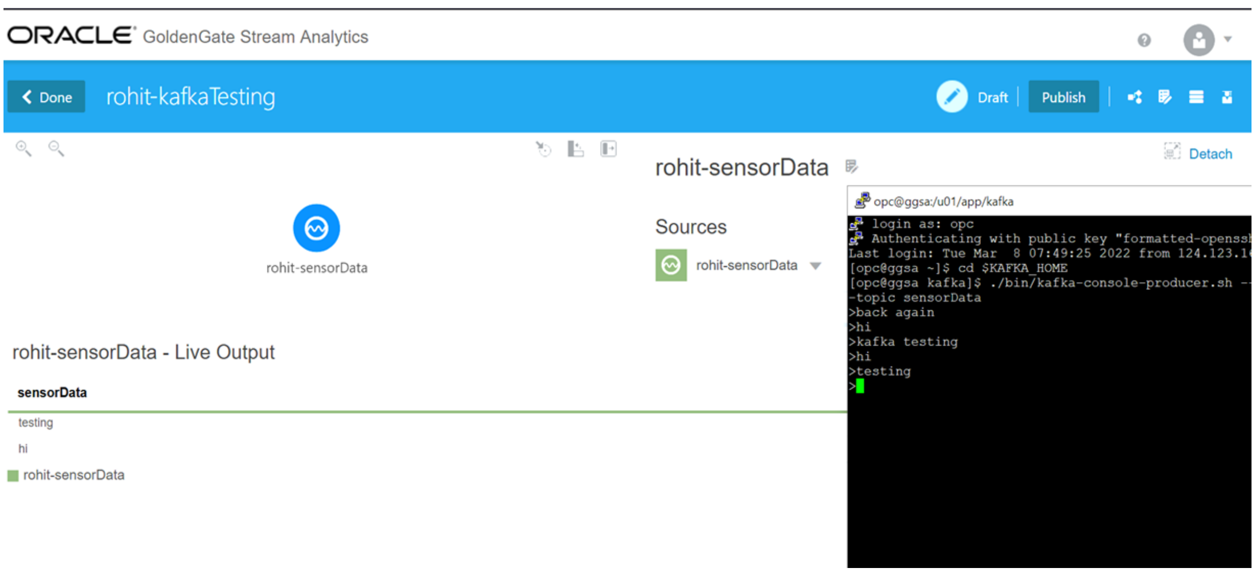


Figure 3.5. 4 Testing Kafka with GGSA

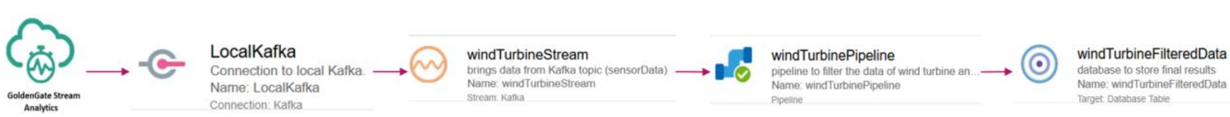


Figure 3.5. 5 Data Flow in GGSA

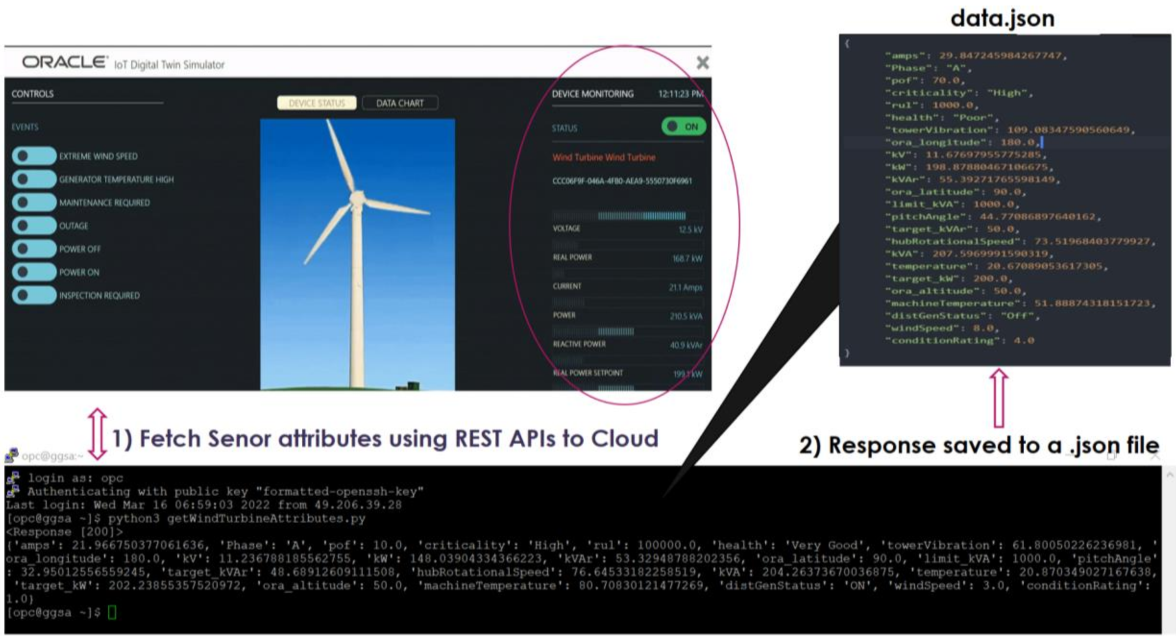


Figure 3.5. 6 Fetching sensor data to the cloud and saving it into a json file using python script.

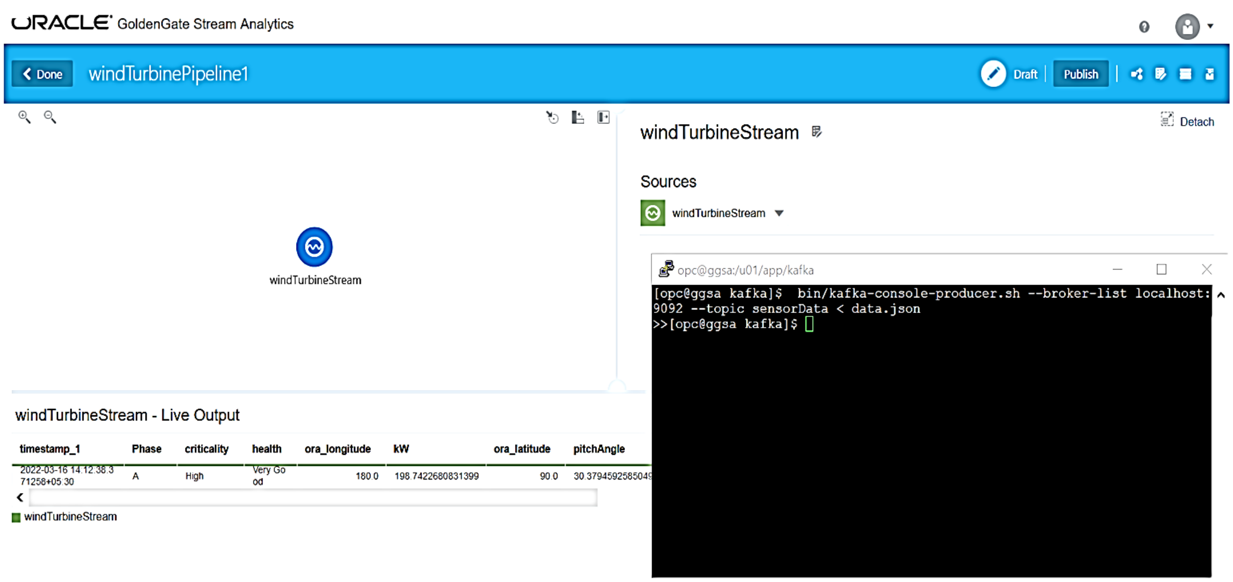


Figure 3.5. 7 Pushing the json file from the cloud to Kafka and pulling it from GGSA

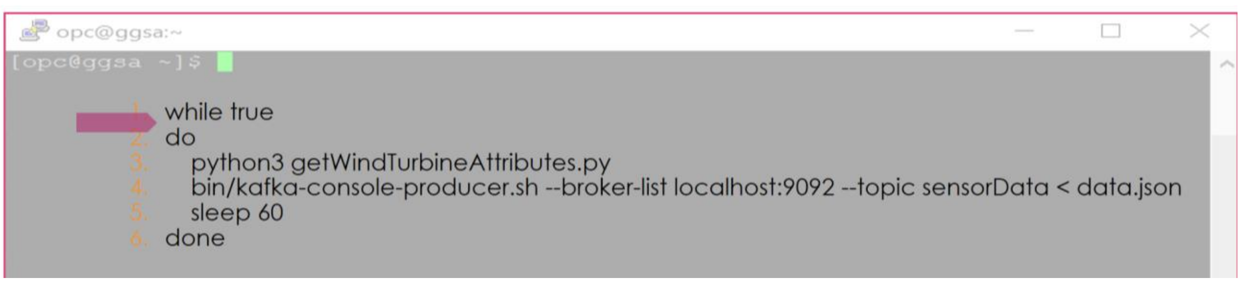


Figure 3.5. 8 Making the pushing and pulling process recurring using bash script

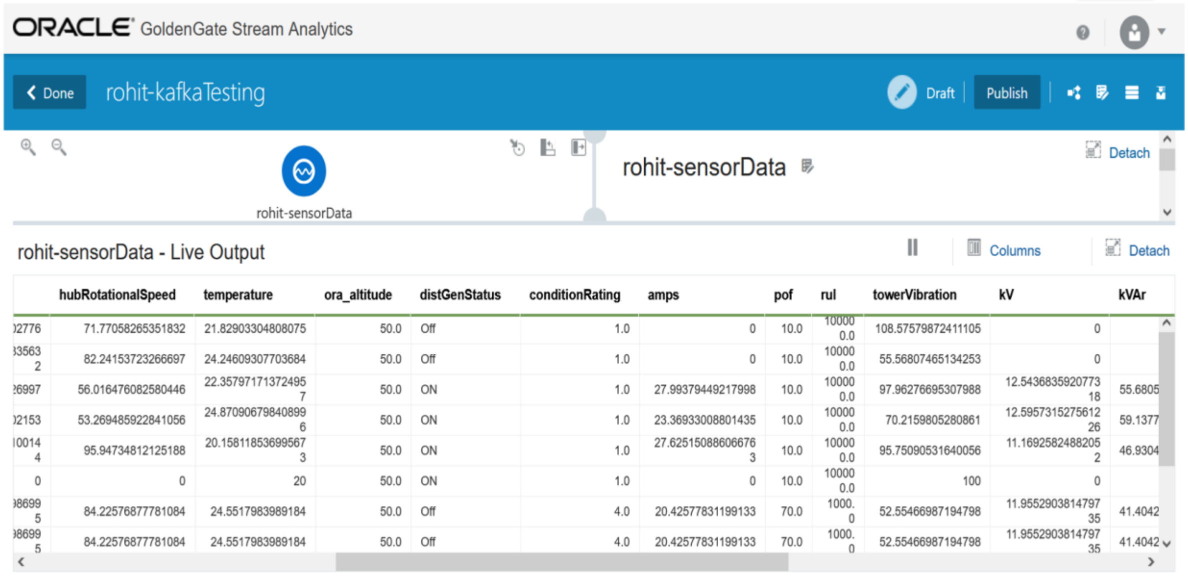


Figure 3.5. 9 Sensor data being populated in GGSA

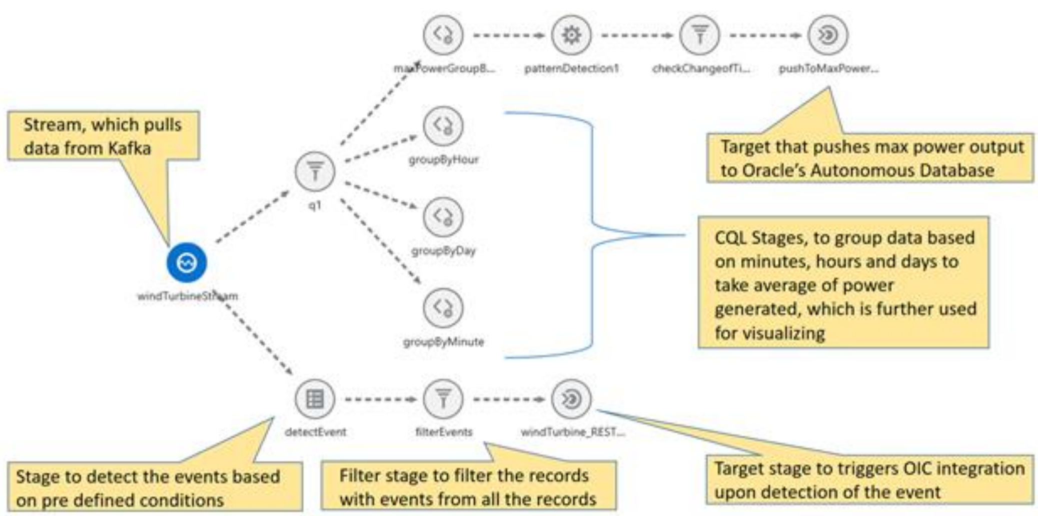


Figure 3.5. 10 Data Pipeline designed in GGSA

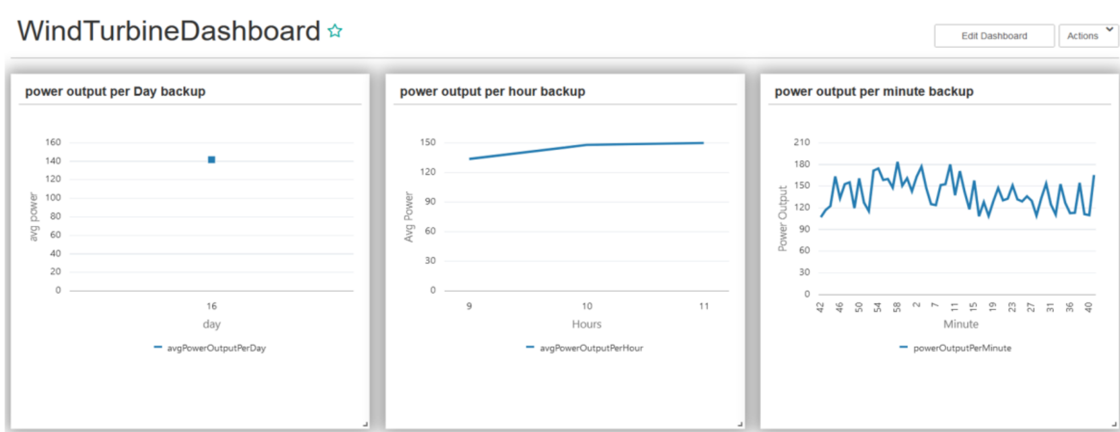


Figure 4.1. 1 GGSA Dashboard

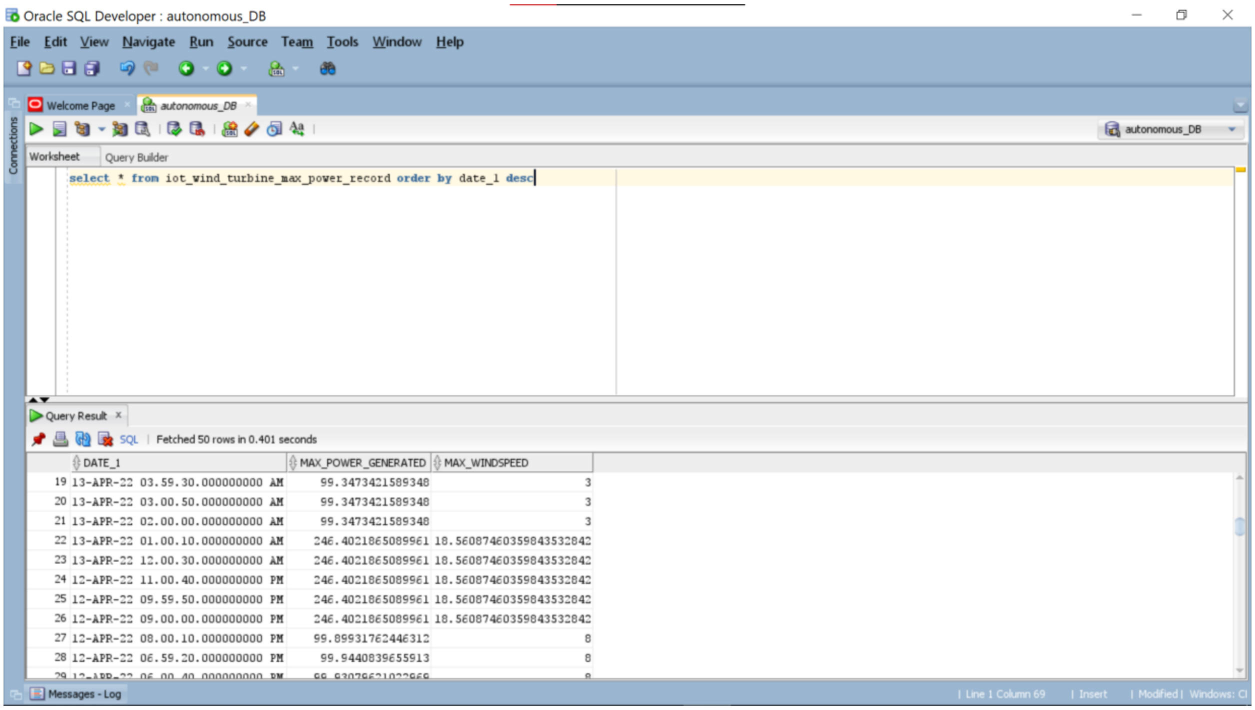


Figure 3.5. 11 Max wind speed and Power generated stored in the database for every Hr

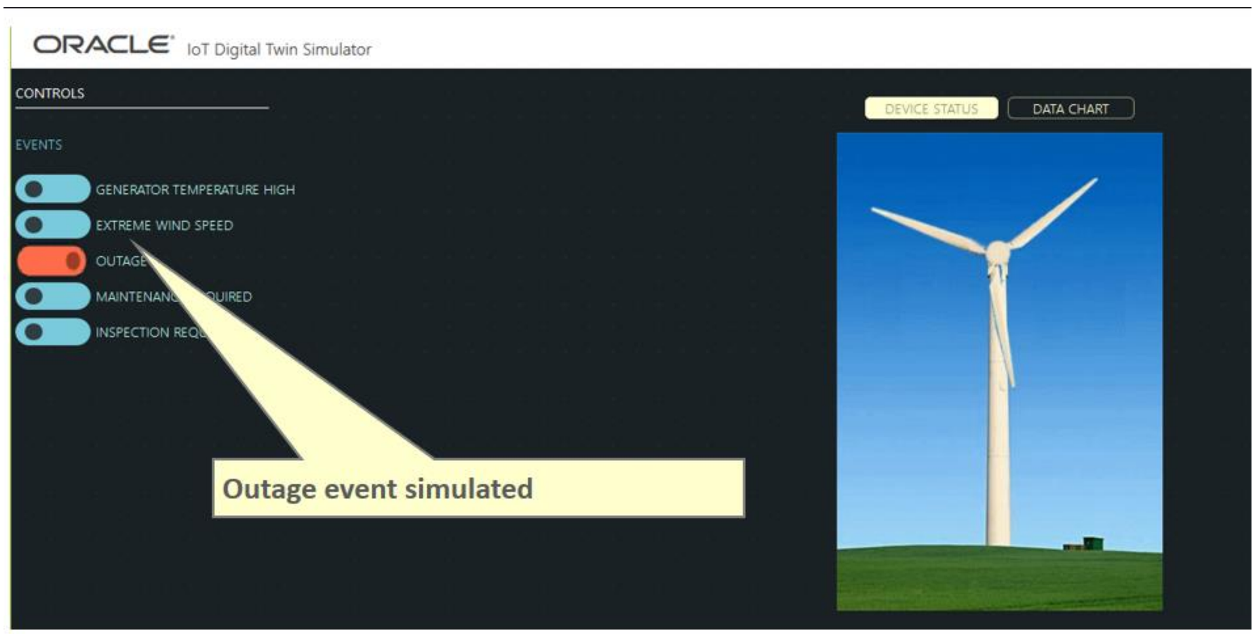


Figure 3.5. 12 Outage event simulated in IoT cloud service

DocuSign Envelope ID: E3BCB725-9E48-4AEE-A279-FF0870EF0770

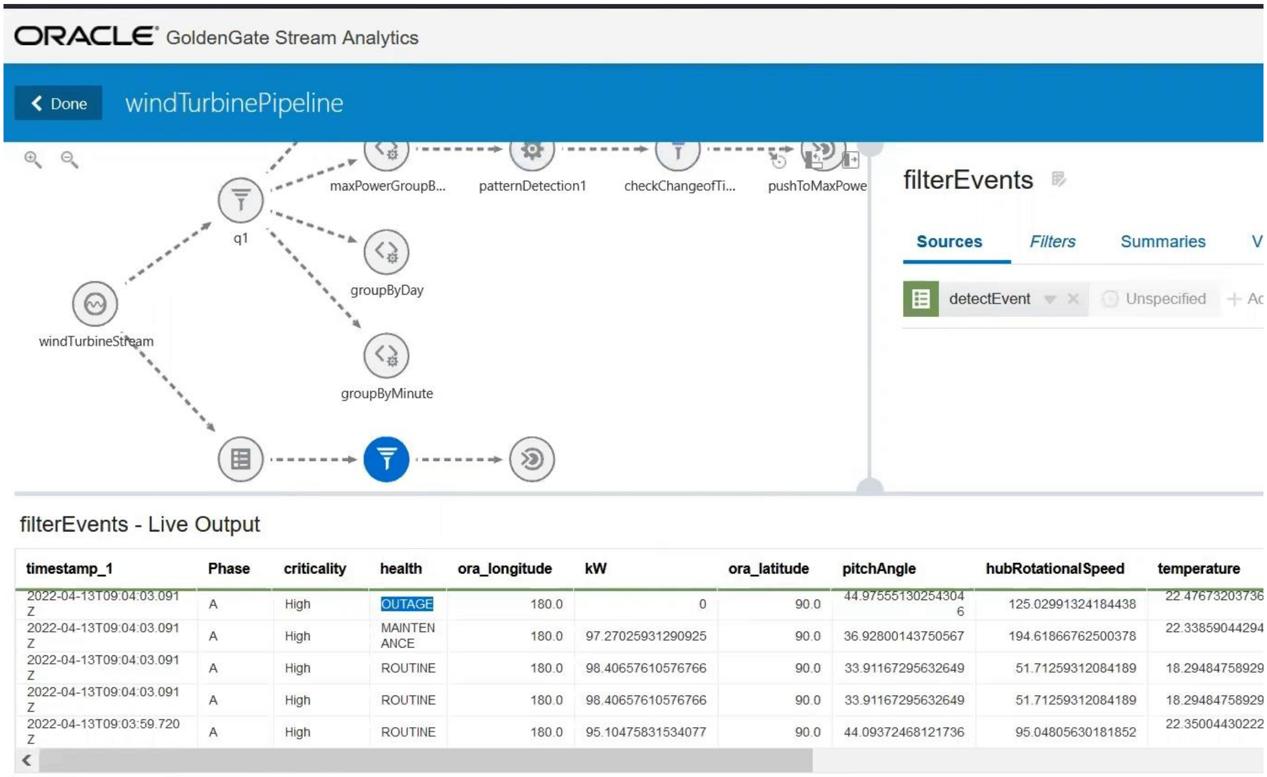


Figure 3.5. 13 Outage event detected in GGSA which triggers OIC Integrations

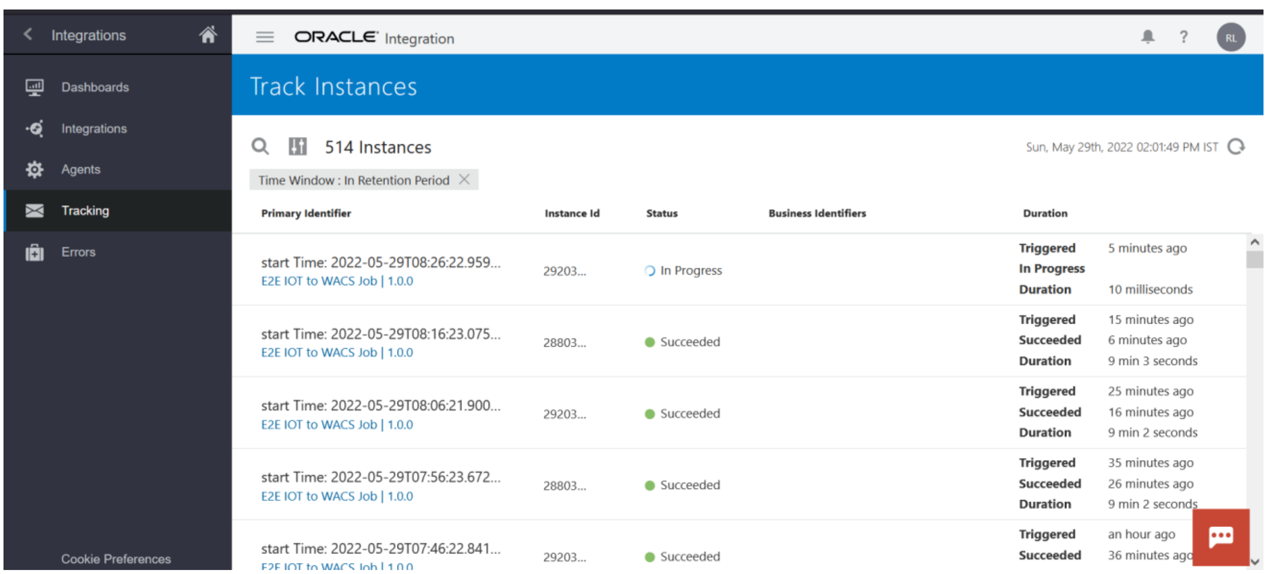


Figure 3.5. 14 OIC dashboard where integration status can be verified

Integration which was called by GGSA on event detection creates WAM activity if there are no pending activities of that type.

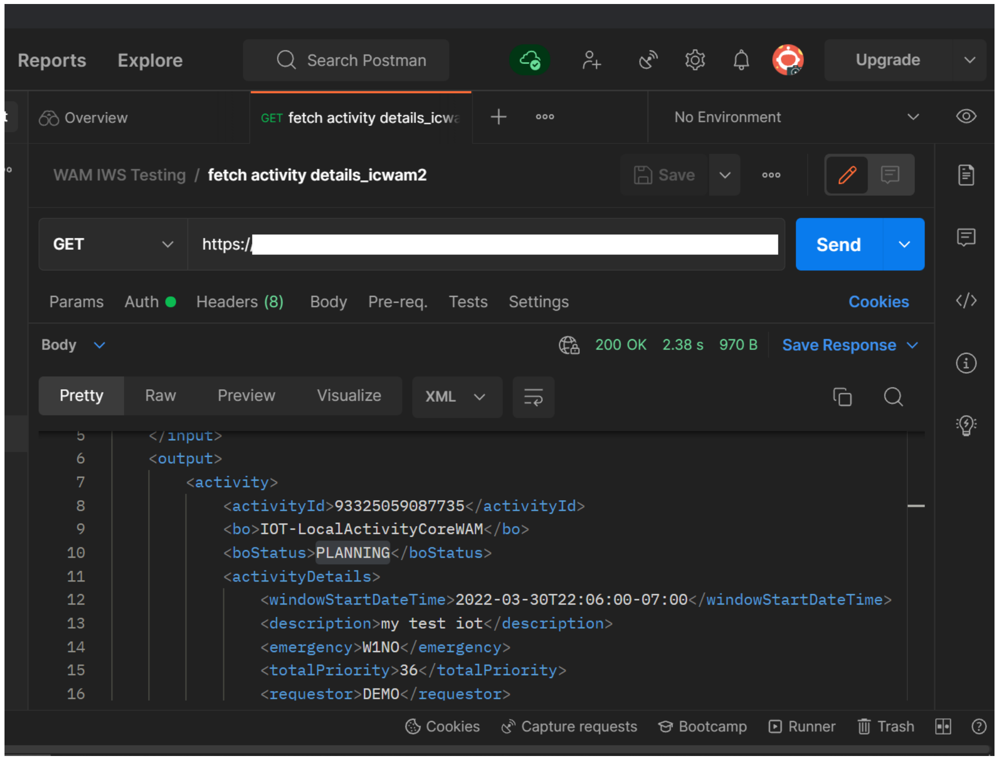


Figure 3.5. 15 API to check the status of the activity

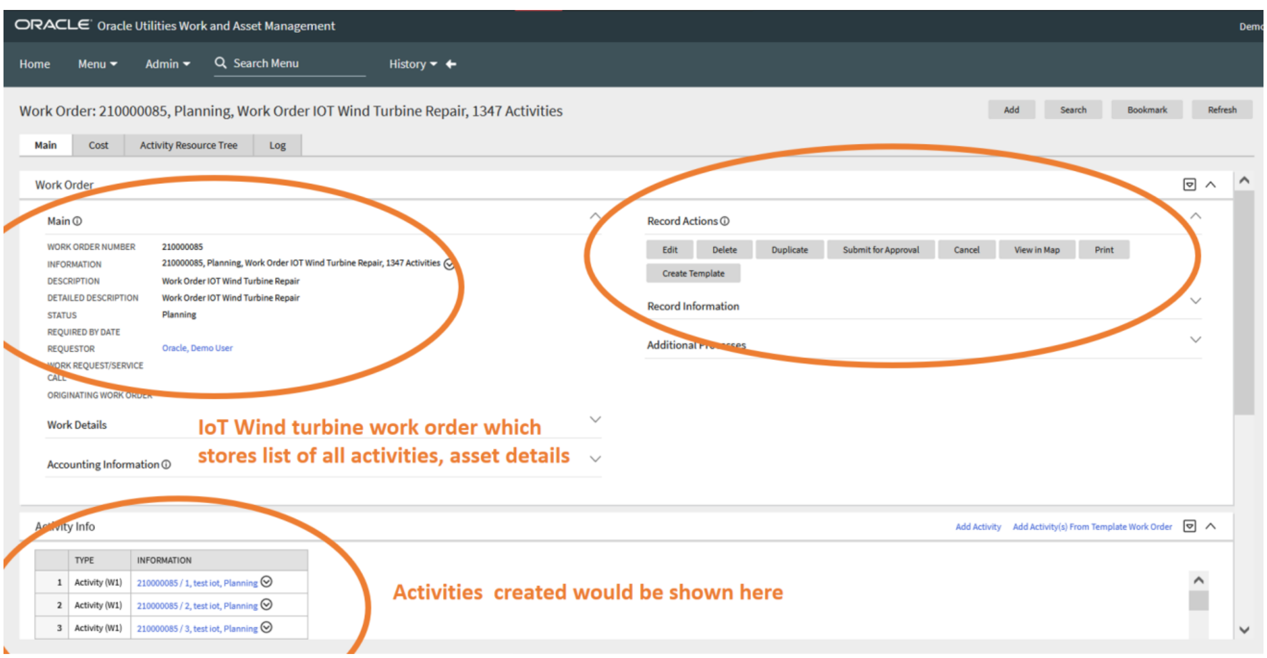


Figure 3.5. 16 Wind turbine work order in WAM

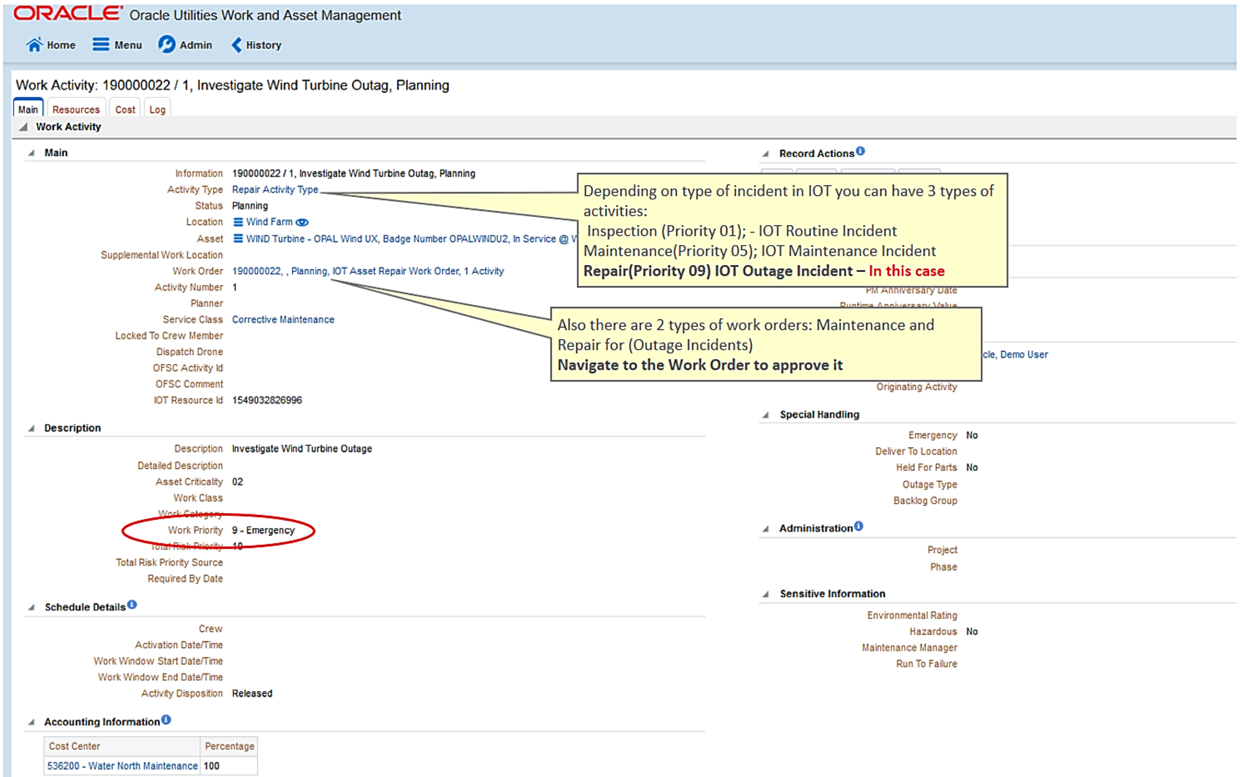


Figure 3.5. 17 Activity created automatically in WAM by OIC Integration

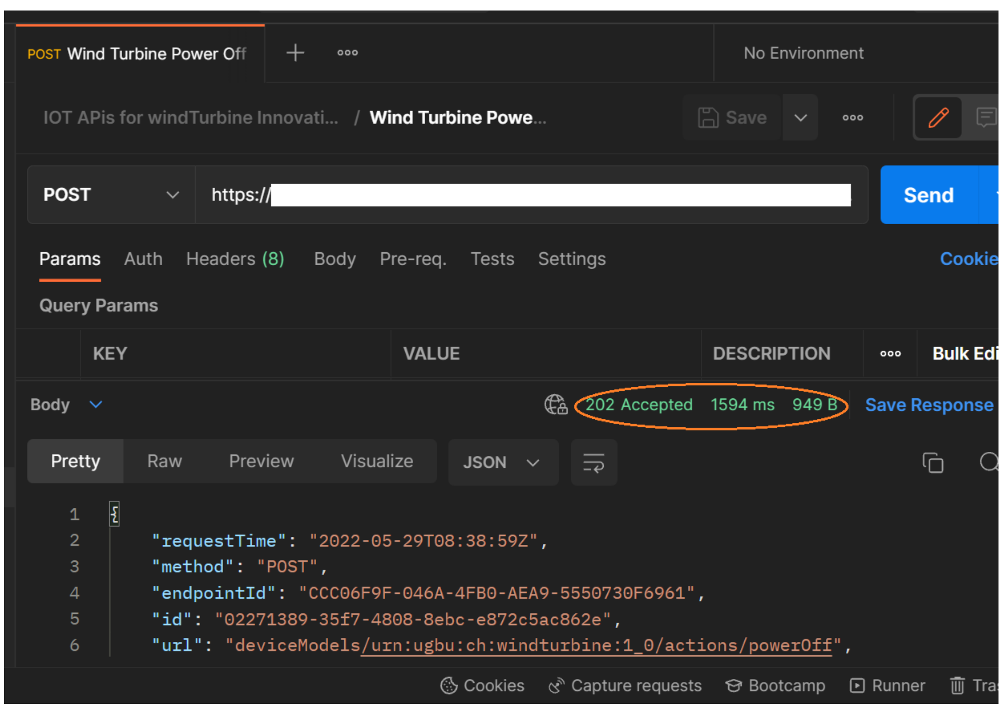


Figure 3.5. 18 API to turn off the Wind turbine

As soon as the activity is created wind turbine is turned off to prevent further damage.

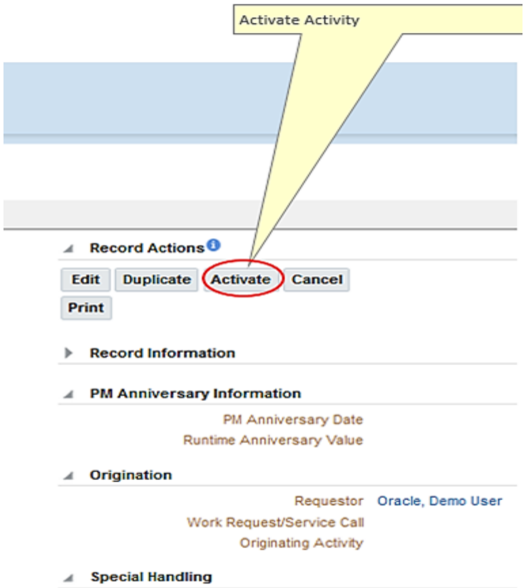


Figure 3.5. 19 Activating Activity in WAM

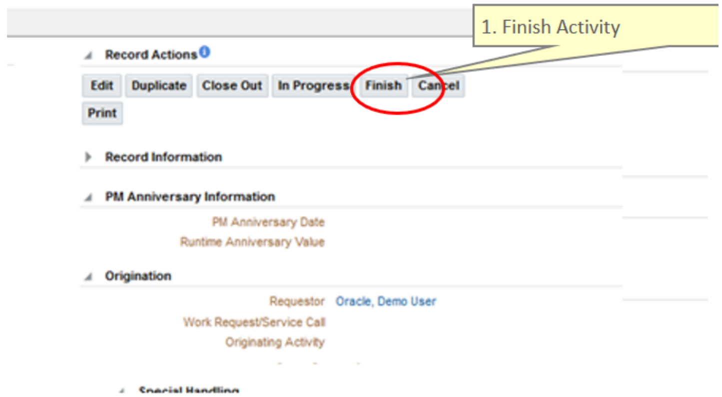


Figure 3.5. 20 Finishing Activity in WAM

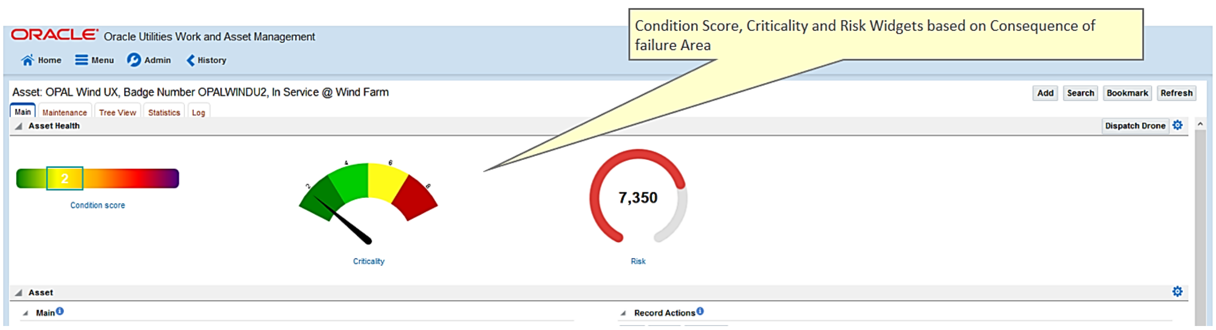


Figure 3.5. 21 Asset score is modified

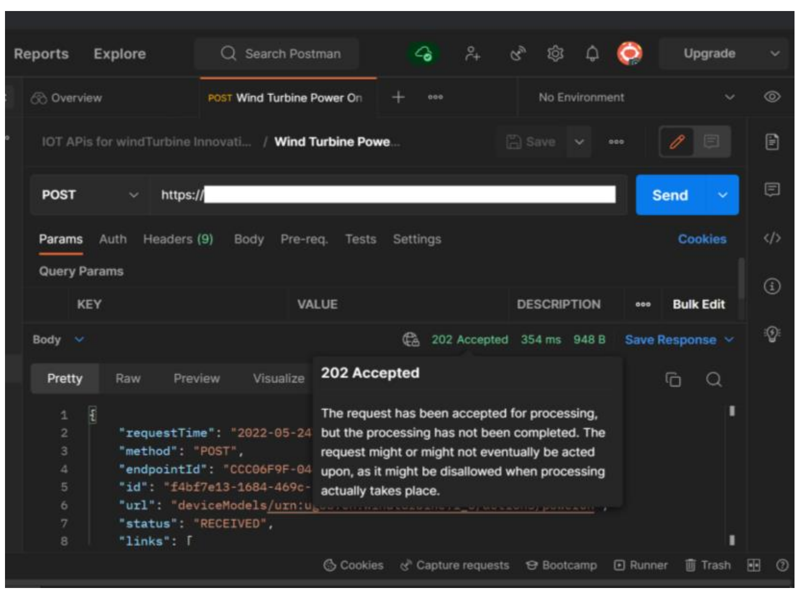


Figure 3.5. 22 After the activity completion state of the wind turbine is restored using API

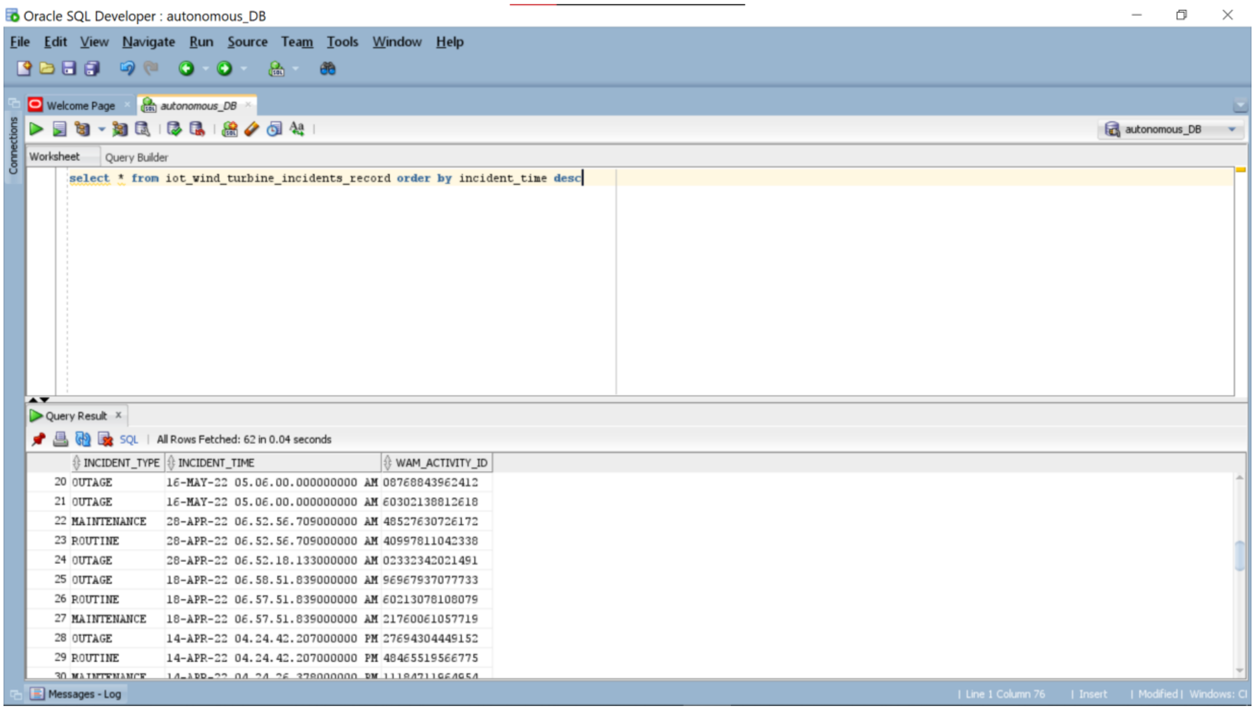


Figure 3.5. 23 Event details are stored in the database

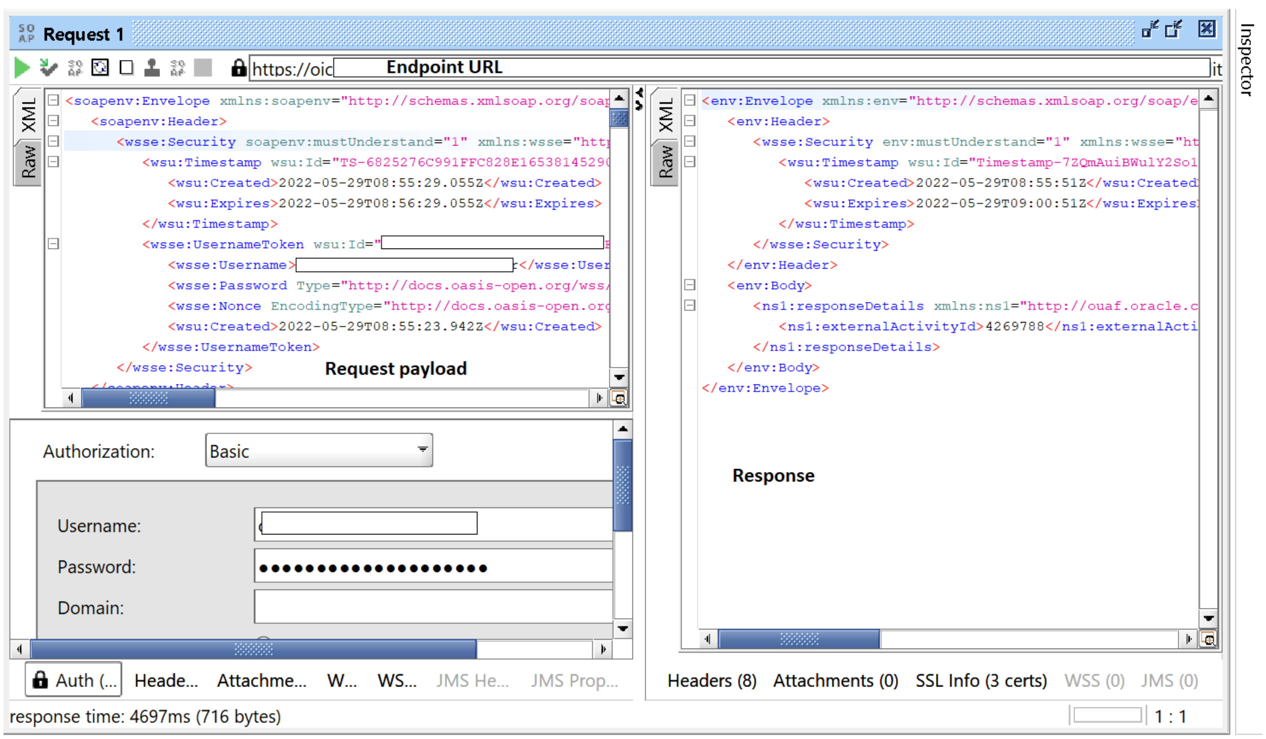


Figure 3.5. 24Soap U

4. CONCLUSION AND FUTURE WORK

4.1 Conclusion

As a consequence, a real-time data analytics platform was created to monitor data from Wind Turbine, generate activities in WACS automatically, and make educated decisions based on the insights given. The methodology which we have opted for (Condition Monitoring), reduces maintenance costs by up to 24% and prevents unplanned outages by up to 49%. This extends the lives of the machines by a few years. Our goal was to spot upcoming equipment failures so that maintenance and inspection can be proactively scheduled whenever it is required, not before, as in preventive maintenance, and not after, as in reactive maintenance. This is what separates the preventive, reactive, and predictive types of maintenances. Various cutting-edge tools were researched and learned while developing this project. The system created is scalable and can handle any quantity of data that is sent into it. The results are stable and reliable as expected. Finally, the data was displayed using charts, and the interface was made accessible via online and mobile devices.

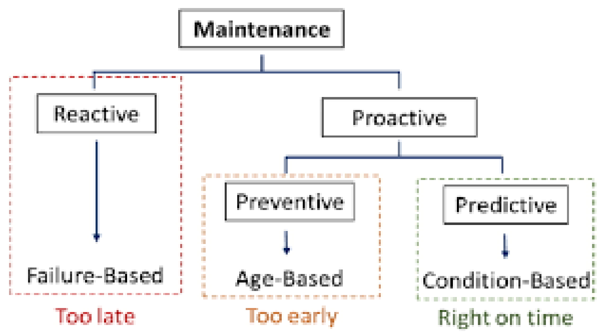


Figure 4.1. 2 Classification of Maintenance Programs

4.2 Future Scope

In the future, A machine learning / deep learning framework might be useful in predicting the events, as of now only 3 events are being detected based on pre-defined conditions but using such a predictive model also allows for the creation of more sophisticated insights than are now available. To predict an outcome, we apply an algorithm to the streaming data and that's called a predictive model. In GGSA a predictive model can be an Open Neural Network Exchange (ONNX) or Predictive Model Markup Language (PMML) file using which more events can be effectively predicted by minimizing the false positives.

Also, one may investigate the potential of producing more insights. Similarly, in terms of visualization, we might investigate new types of visualizations and more devices. Only one Wind Turbine has been used now, In the future, our solution can be scaled by including more than one Wind Turbine which can be uniquely identified based on latitude and longitude.

Our Solution becomes a complete end-to-end product by bringing Oracle Field Service Cloud (OFSC) into the picture. It is a field service management, a cloud solution that helps in scheduling activities, automatically assigning service activities to the field workers, etc

We have developed and tested the solution with the simulated data, working with real-time data would bring a lot more complexities into the solution. In the future, we are planning to access real-time data using Tesla's Power Hub APIs, which help in managing many energy assets by providing a single interface. The assets include solar, storage, and non-Tesla assets (generators, breakers, transformers), etc.

As of now, we have only focused on 3 parameters (Machine Temperatur, Hub Rotational Speed, and Tower Vibration) to predict the health of the Wind Turbine, in the future few more parameters can be included such as oil analysis, Infrared Thermography, etc.

REFERENCES

1. Oracle IoT Digital Twin Simulator documentation
   https://docs.oracle.com/en/cloud/paas/iot-cloud/iotgs/use-simulated-devices.html

2. REST APIs documentation for Oracle Internet of Things Cloud Service
   https://docs.oracle.com/en/cloud/paas/iot-cloud/iotrq/rest-endpoints.html

3. POSTMAN Official documentation
   https://learning.postman.com/docs/getting-started/introduction/

4. SOAP UI Official documentation
   https://www.soapui.org/getting-started/

5. Kafka Official documentation
   https://kafka.apache.org/documentation/

6. Oracle Cloud Infrastructure (OCI) documentation
   https://docs.oracle.com/en-us/iaas/Content/home.htm

7. Golden Gate Stream Analytics (GGSA) documentation
   https://docs.oracle.com/en/middleware/fusion-middleware/osa/19.1/index.html

8. Oracle Integration Cloud (OIC) documentation
   https://docs.oracle.com/en/cloud/paas/integration-cloud/index.html

9. Creating an Oracle Autonomous Database
https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/autonomous_database_quick_start/adb-provision/adb-provision.html#create-the-oracle-autonomous-database-instance

10. Oracle Utilities Work and Asset Cloud Service (WACS) documentation
https://docs.oracle.com/en/industries/utilities/work-asset-cloud-service/

11. Oracle Field Service Cloud (OFSC) documentation
https://docs.oracle.com/en/cloud/saas/field-service/22b/

12. Connecting to Autonomous DB presented in the cloud using SQL developer
https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/cswgs/autonomous-connect-sql-developer.html#GUID-14217939-3E8F-4782-BFF2-021199A908FD

13. Comparison between different types of maintenances
https://www.plantengineering.com/articles/the-maintenance-function-like-manufacturing-itself-is-a-rapidly-changing-environment/

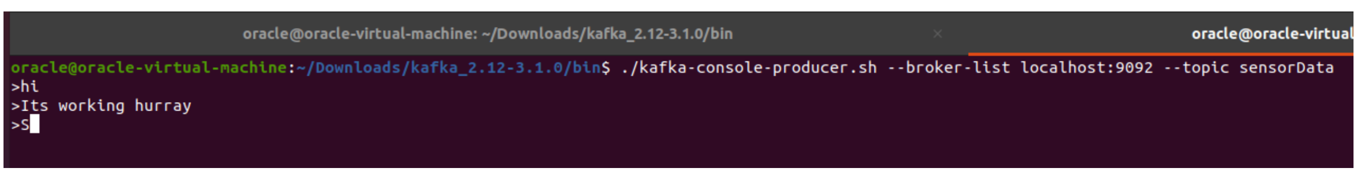
14. Detailed report and maintenance programs
https://blog.infraspeak.com/maintenance-statistics/

15. About Spark engine
https://databricks.com/spark/about

APPENDIX

Setting up Kafka

1. [Download latest version]
2. Extract: tar –xvf <filename>
3. Run Zookeeper
    a. We need to have java for that
        i. sudo apt install openjdk-17-jre-headless
        ii. java –version
    b. bin/zookeeper-server-start.sh config/zookeeper.properties
4. Run Server/broker
    a. JMX_PORT=8005 bin/kafka-server-start.sh config/server.properties
5. Create a topic
    a. ./kafka-topics.sh --create --topic sensorData --bootstrap-server localhost:9092 --replication-factor 1 --partitions 4
    b. To list topics
        i. ./kafka-topics.sh --list --bootstrap-server localhost:9092
6. Running Producer
    a. ./kafka-console-producer.sh --broker-list localhost:9092 --topic sensorData



7. Running Consumer
    a. ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic sensorData --from-beginning