Federated Cloud for Water Resources

Liyakhanya Candy Tabata
Department of Computer Sciences
University of the Western Cape
Bellville, South Africa
3678013@myuwc.ac.za

Mogammad Uzair Jones
Department of Computer Sciences
University of the Western Cape
Bellville, South Africa
3687046@myuwc.ac.za

Abdullah McDonald
Department of Computer Sciences
University of the Western Cape
Bellville, South Africa
3708949@myuwc.ac.za

***Abstract*—The objective of this project is to implement Federated Cloud model for Water Resource Management assuming we are collaborating with the Water Resource Department. Water resource management tool is one of the tools that are essential to the communities as water is one of our vital substances for survival. In this paper we implemented a water resource management tool using IBM cloud services assuming the Water Resource Department chose the path of using the cloud services to digitize all the operations which include three departments, namely, storage department, billing department, and the administrative department where each department is treated like a ”cloudlet”. Application of cloud computing in this operation could be efficient when it comes to costs and offering of agile services. This project can be useful in similar studies and research fields.**

***Index Terms*—Cloud Computing, Federated Cloud, Water Resource Management.**

I. Introduction

Federated Cloud for Water Resources is a deployment and management of several external and internal cloud computing services that allows a water resources department to store, maintain or analyze health water consumption information. Federation in clouds can be seen as a future of Internet of Things (IoT), Big Data applications and Cloud computing [1]. The application of the federation resources increases reliability, it is cost efficient and improves the quality of services. The demand and growth in use of the cloud services increased risks in management of resources from different cloudlets which caused risks for the end users, such as lack of trust between departments, security and lack of knowledge across the departments, resulting in poor delivery of resources [1]. Strategies to manage the resources are related to the three cloud models namely Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). Different parts of the world including Africa, United Kingdom, India etc [2] [3] have faced or are facing issues with water resource management which could have been caused by inadequate rainfall, political breakdown, economic development factors, climate change and other factors [3].

The objective of this project is to implement a federated cloud model for the water resource department in managing water resources based on three departments; a user/client information database, a billing unit, and a administration/visualization unit. Our solution to a Federated Cloud for Water Resources consists of 4 components, namely, water meter sensors that captures households water consumption, database for storage of user/client information, billing unit that calculates the billable amount for the amount of water used, and an administration/visualization unit that allows for visualization & analysis of the current water consumption on a daily and monthly basis as well as an interface where users/clients can view their current water bill.

II. Literature Review

In this section we review the state-of-art of water resource management and water consumption management based on cloud computing. The aim of applying the water resource management is to reduce the effects of the challenges to the communities and businesses and challenges facing the water departments [4]. Authors [5] and [6] suggested that some parts of the world's water consumption is not visualized nor measured in real-time leading to water leakage not being detected and unnecessary water being wasted.

Author [5] proposed a smart water measurement consumption system that provides real-time analytics and measurements of water consumption. The proposed system consists of water consumption data being collected from house-holds through sensors in JSON format and sent to to the a smart meter and stored in the edge-gateway that uses a NoSQL database CouchDB. The obtained data is sent to the cloud server where it is stored in a NoSQL database Cloudant with the house-hold members information obtained via GPS and analyzed for any leakage using a leakage algorithm they implemented. The data analysis were performed through IBM Stream Flow Analysis platform that enables visualization of different data sizes in real-time. A web-portal was used to display the analysis of smart meters at a specific location in real-time was deployed in Azure Cloud. However, author [6] proposed a similar system, an efficient water monitoring and management system based on IoT, a real-time monitoring and cost-effective system that detects water leakage and notifies the end-user. The proposed system uses IoT sensors to measure water consumption in a house-hold and the obtained data is sent to the cloud. Based on the data obtained, bills are generated at the end of each period and sent to the end-user. Author [4] also 3 proposed a smart water management system which monitors the water levels in real-time, identifies distribution leaks in the system when monitoring the water quality using IoT sensors and analytics techniques. The system populates the database based from the data obtained from the sensors into the cloud, and alert in form of an email of text is sent to the end-user if there are changes with the water levels or daily analysis of the water levels. Author [7] proposed a cloud based IoT smart water consumption monitoring system implemented on Android Studio Application and Thingspeak cloud services, which enable use of IoT data to be aggregated, analyzed and visualized in real-time. The proposed system uses water meter sensors that record water flow in litres, fitted house-hold pipelines. The obtained data in every second is sent to an IoT device. The system uses Thingspeak communication Application Programming Interfaces (APIs) to host live data onto the cloud. Android studio developed application receives real-time data from the cloud leading to more information such as reports and billing costs to being generated for the end-user. The data obtained from the cloud was statistically analysed and visualised using MATLAB.

Application of cloud federation can be an efficient method of over coming limitations and delivering quality to the end user [8] when it comes to management of water resources. Water leakage is a factor in the lack of water management thus the benefit of IoT techniques, which can lead to increases in bills and other negative factors. Use of the cloud if convenient as water consumption notifications and analysis can be viewed everywhere for different scales which benefits the water department.

The organization of the rest of this paper is as follows; Section III discusses the methodology followed for this project, Section IV mentions all the IBM Cloud tools and programming languages used, Section V shows the working of federated cloud, Section VI explains the UML for this project, Section VII mentions the individual experience of each team member, Section VIII explains the limitations during implementation, IX discusses how we can improve on this project, and Section X concludes the paper.

III. Methodology

This section describes the methodology we followed in implementing the project. This section is divided into 4 subsections based on the three departments namely, the system architecture, database, processing and administration.

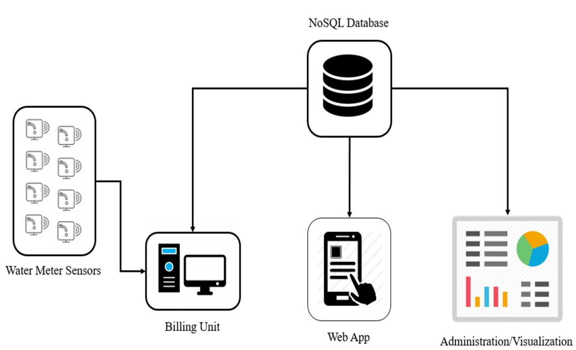
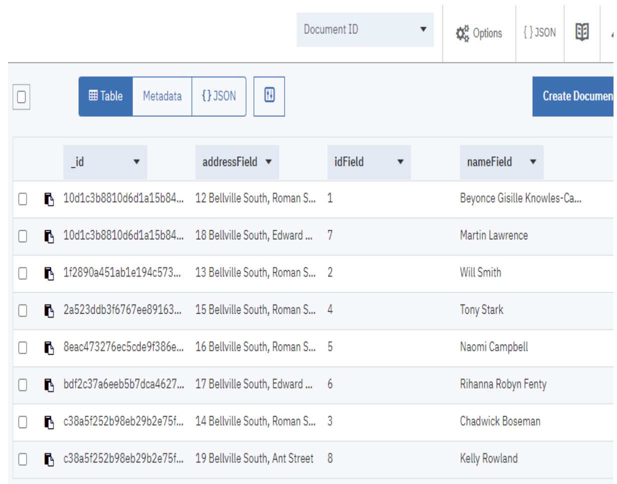


Fig. 1: Architecture for Federated Cloud for Water Resources

A. Architecture

The architecture for our Federated Cloud for Water Resources consists of 4 parts, namely, water meter sensors, a NoSQL database, a billing unit, and an administration/visualization unit. Figure 1 shows the architecture for our project. As the implementation of this project is a proof of concept, we decided to use 8 water meter sensors that represents 8 different household in two specific areas. Area 1 & 2 contains 4 household (and water meters sensors) each. These water meters sensors record the water consumption for the household it is attached to. The rate at which these water meter sensors records data and sends it back to the billing unit is on a per-minute basis.

The billing unit collects data from each of the water meter sensors and then calculates the billing amount for that specific amount of water used. Once this is done, it immediately stores this amount along with other data in the NoSQL data. The NoSQL database stores documents for two databases, a dataset that contains information amount the households and streaming data that contains the water consumption and billing amounts for specific households. The Administration/Visualization unit retrieves the stored streamed data from the NoSQL database and does analysis on water consumption per household or area. The streamed data stored in the NoSQL database is also used by a web app to provide households with an interface in which they can view their current billing amount for a month.



_id	addressField	idField	nameField
10d1c3b8810d6d1a15b84...	12 Bellville South, Roman S...	1	Beyonce Gisille Knowles-Ca...
10d1c3b8810d6d1a15b84...	18 Bellville South, Edward ...	7	Martin Lawrence
1f2890a451ab1e194c573...	13 Bellville South, Roman S...	2	Will Smith
2a523ddb3f6767ee89163...	15 Bellville South, Roman S...	4	Tony Stark
8eac473276ec5cde9f386e...	16 Bellville South, Roman S...	5	Naomi Campbell
bdf2c37a6eeb5b7dca4627...	17 Bellville South, Edward ...	6	Rihanna Robyn Fenty
c38a5f252b98eb29b2e75f...	14 Bellville South, Roman S...	3	Chadwick Boseman
c38a5f252b98eb29b2e75f...	19 Bellville South, Ant Street	8	Kelly Rowland

Fig. 2: Client/user information database

B. Database

We could not find a publicly available dataset that contained user/client information thus we opted to generate fictitious user/client information. We followed an IBM publicly available tutorial to push the random dataset into the IBM Cloudant database, a NoSQL database. Figure 2 shows the database in a table format. The database consists of the following fields:

- id_, a field used by the database to identify each entry.

- addressField, which contains the address of each client or user in the system.
- idField, which contains each household identification number.
- nameField, which contains each user or clients full name.

The user/client information will be used to display the billing information of households in an area.

Fig. 3: Billing Unit/Processing

C. Billing Unit (Processing)

The billing unit calculates the billing for the amount of water used by a specific household. This unit functions using of 3 components as shown in Figure 3, namely, a streaming source, processing part, and a database. The streaming source used for this project is the IBM Watson Studio IoT Platform and it has been configured to use 8 water meter sensors. These 8 water meter sensors are situated in two areas, Area 1 and Area 2, with each area having 4 water meter sensors. The water meter sensors are configured to send a value between 0 and 1 to the billing unit which represents the amount of water used by a household every minute in litres. Once a reading from a water meter sensor is received by the billing unit, it multiplies the reading by 0.0002 and takes it as the billing amount for that reading. This reading along with the water meter sensor ID, area ID, date, time and calculated billing amount is then stored in the IBM Cloudant database.

D. Administration

The primary aim of this unit is to provide analyses and feedback on the amount of water used by an area and a household. It will contain two sections, a visualization dashboard and a web app. The visualization will provide analysis on the amount of water used by each household and by each area, which households are currently consuming the most water and also which area is consuming the most water. With this analysis, a water resources department can save water by simply alerting households or areas when they use too much water. The web app's main purpose is to provide households with interface to view their current water consumption and billing amount. Other capabilities that this web app will provide is visualization graphs for the water consumption per day and per month.

E. Continuous Delivery

As soon as the Web Application was created, a continuous delivery pipeline was created which automatically builds and deploys the application during every commit from the Gitlab repository that the code was saved on.

IV. Experimental Setup

A. Stream Setup

Using IBM Watson's IoT platform, sensors were created that allowed us to produce streaming data for water meters. A notebook on IBM Cloud Pak for Data allowed us to generate water meter readings using the sensors created. We then used IBM Watson Studio service to created a stream flows that connected the IoT platform to a code operator and NoSQL Cloudant database. These services where connected through APIs with Identity Access Management as a means of accessing the services.

B. Website Setup

We made use of the Django Python service on IBM Cloud. Through this service, a Web application was created which automatically generated a delivery pipeline as well as a Kubernetes cluster. The delivery pipeline allowed the source code to be saved Gitlab instantaneously. The Cloudant database was then connected to the Web application to display bills and graphs.

V. Results and System Testing

In this section, we will explore features of our Federated Cloud for Water Resources and show the results of the following:

- Streaming Source & Billing Unit
- Visualization dashboard
- Web App

A. Streaming Source & Billing Unit

As mention earlier, we used the IBM Watson IoT Platform as our Streaming Source and it simulates water meter sensors. These water meters produce values between 0 and 1 that represent the amount for water used (in litres) per minute for each household. As soon as these values are generated for each household's water meter, it is sent to the billing unit for calculation of the current bill. Figure 4 shows the Streaming Source (Watson IoT) and the Billing Unit working in tandem in an IBM Streams Flow. IBM Streams Flow allow for connecting various components together for streaming purposes. The Billing Unit isn't an IBM Service but rather a code operator that allows for processing of streamed data. In Figure 5 we can see the code carried out in this Unit. The code operator within the Billing Unit also allows us to send the processed data to a NoSQL Cloudant database. Once this data is stored, it can then be pull for visualization and analysis purposes.

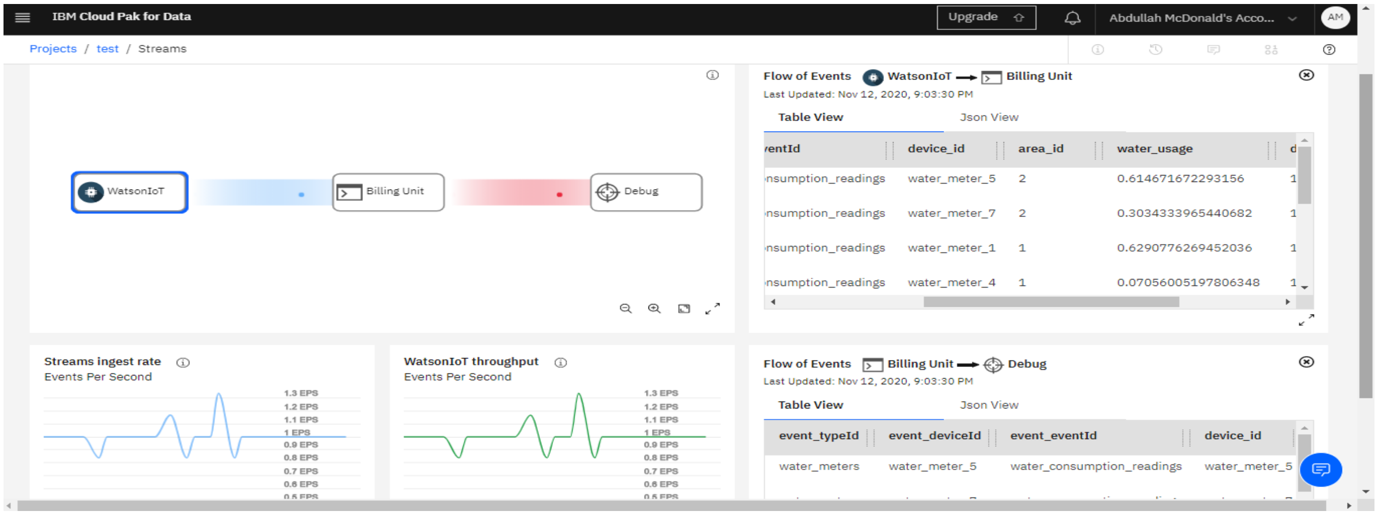


Fig. 4: IBM Streams Flow

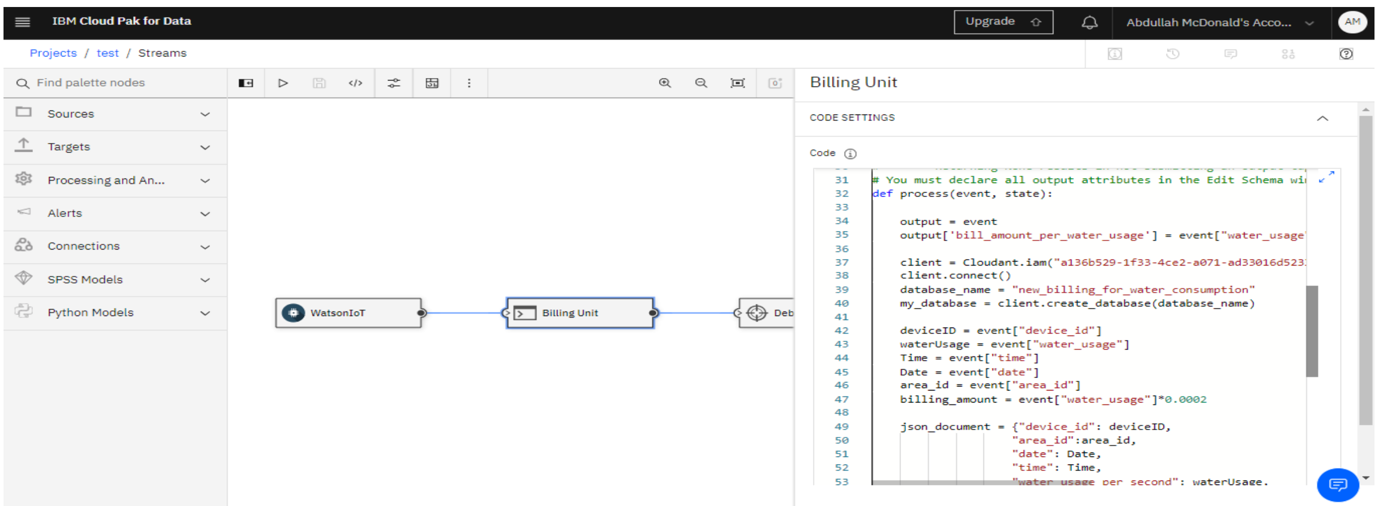


Fig. 5: Billing Unit

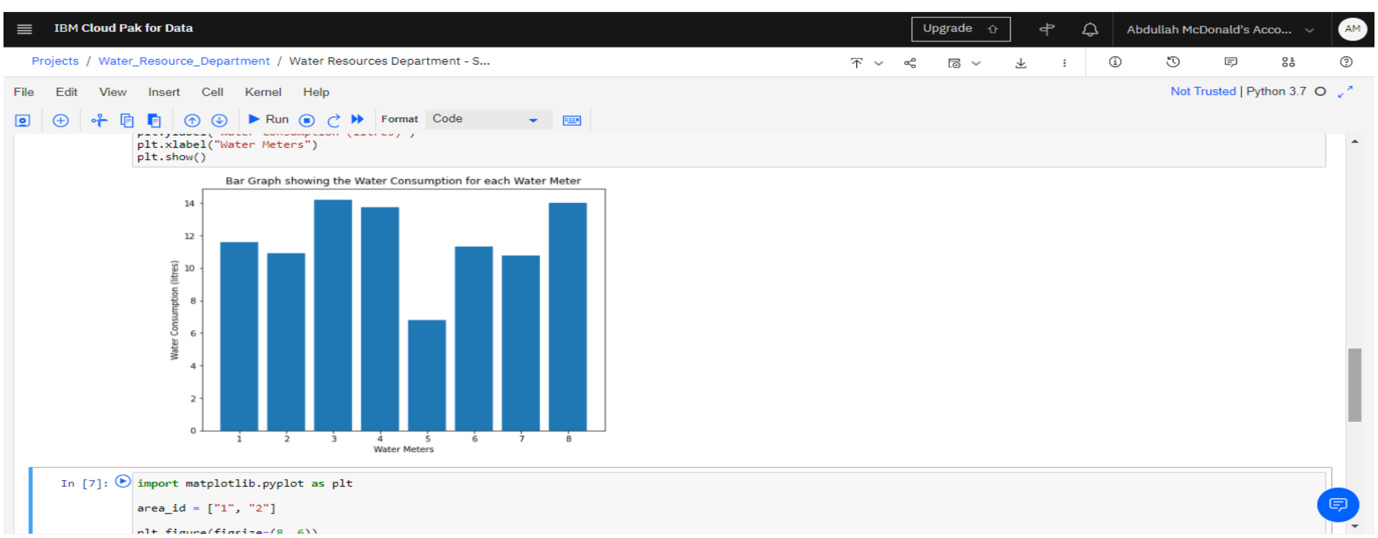


Fig. 6: Water Consumption per Household

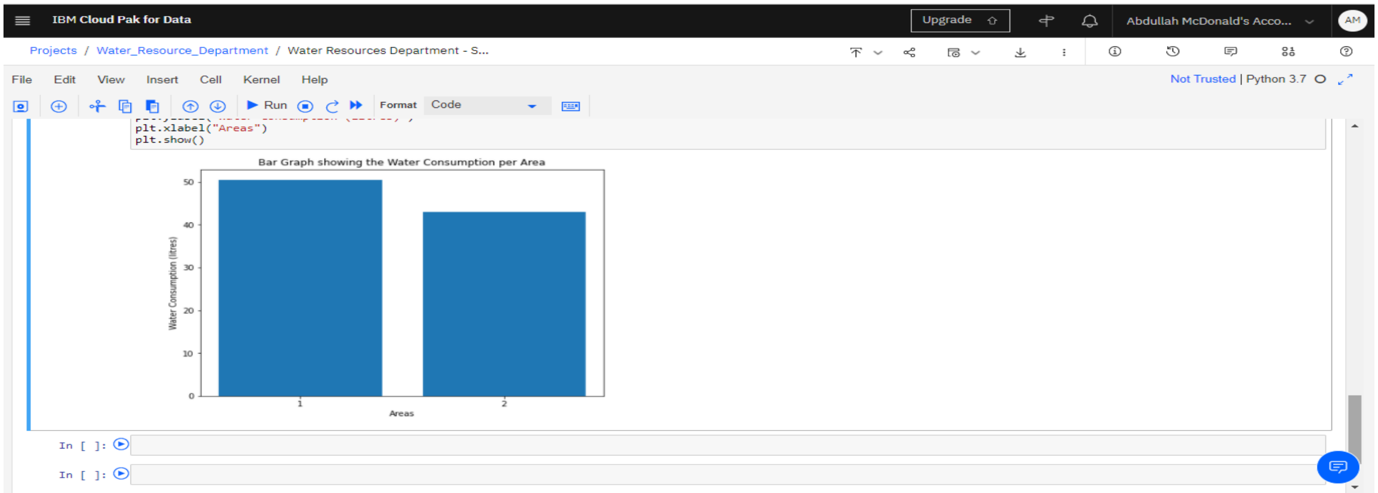


Fig. 7: Water Consumption per Area

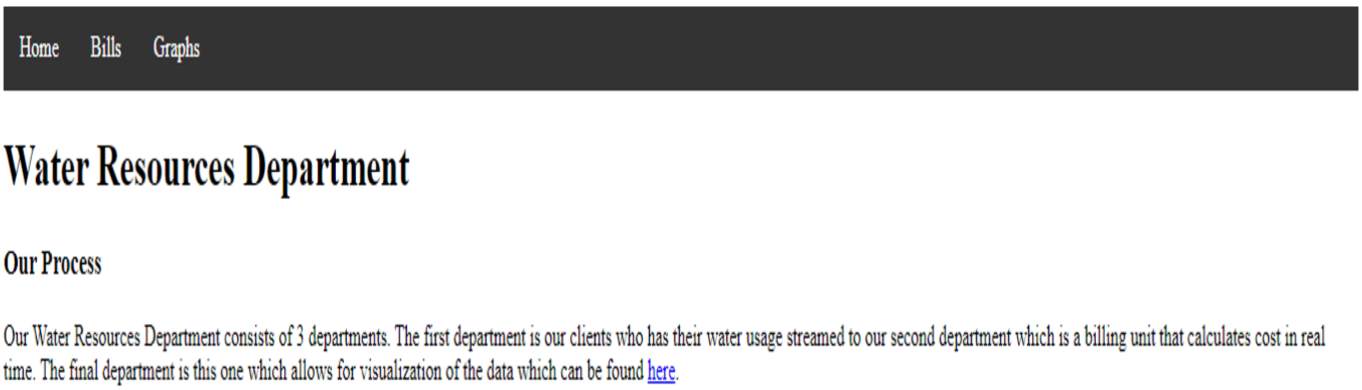
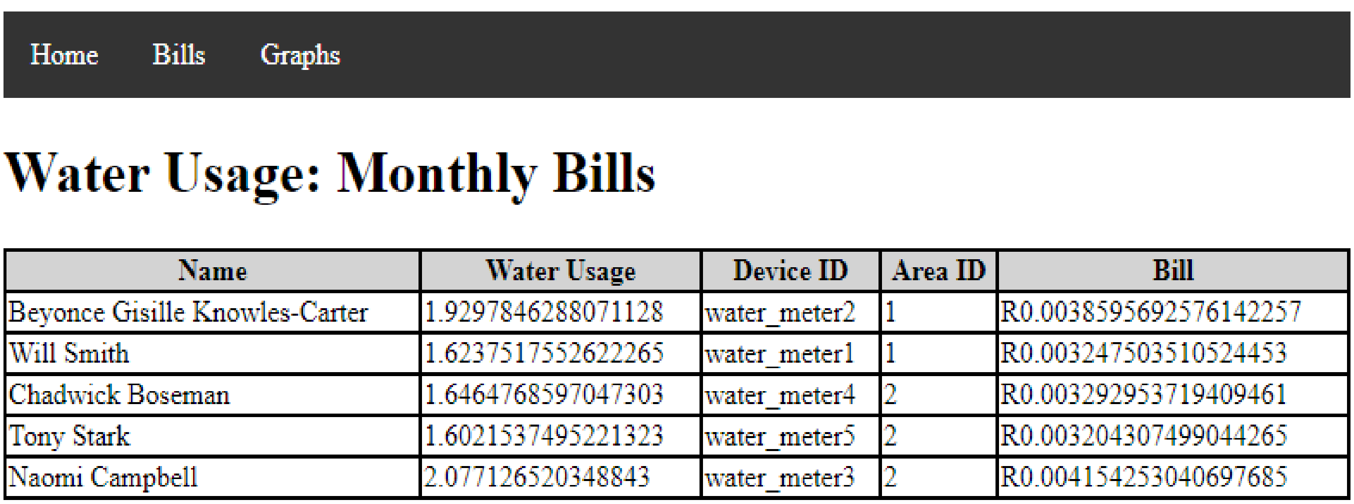


Fig. 8: Web App Home Page



Name	Water Usage	Device ID	Area ID	Bill
Beyonce Gisille Knowles-Carter	1.9297846288071128	water_meter2	1	R0.0038595692576142257
Will Smith	1.6237517552622265	water_meter1	1	R0.003247503510524453
Chadwick Boseman	1.6464768597047303	water_meter4	2	R0.003292953719409461
Tony Stark	1.6021537495221323	water_meter5	2	R0.003204307499044265
Naomi Campbell	2.077126520348843	water_meter3	2	R0.004154253040697685

Fig. 9: Web App Bills Tab

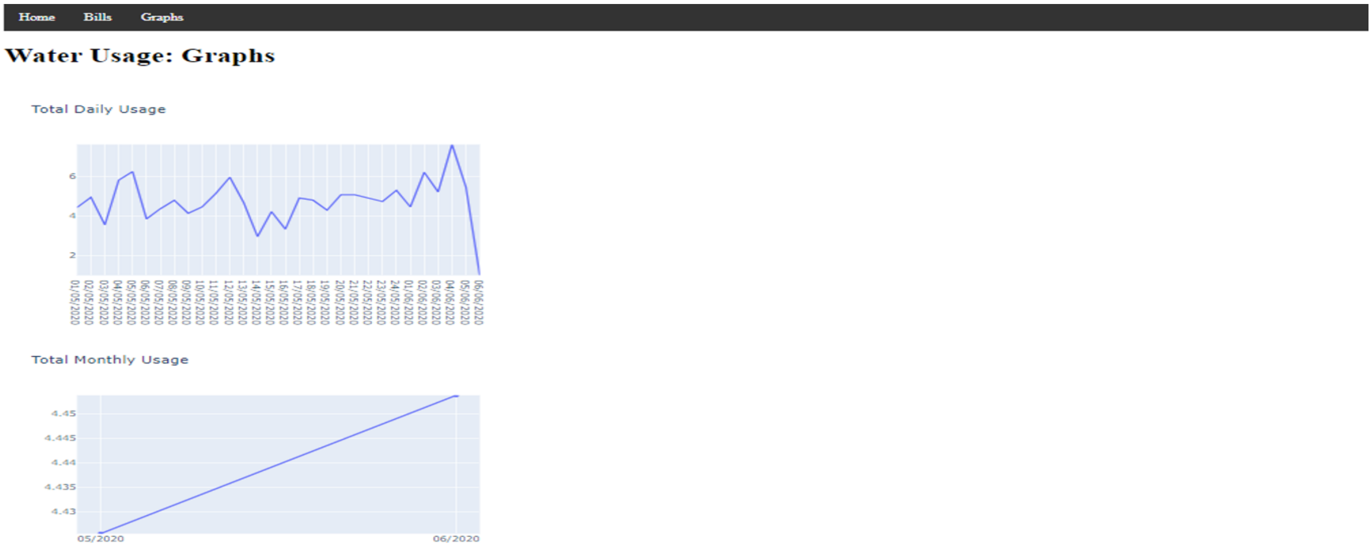


Fig. 10: Web App Graphs Tab

B. Visualization Dashboard

The visualization Dashboard is used to show graphs that depict the amount of water used by each household and each area. Since we used a NoSQL database, we were restricted to using a Jupyter Notebook on IBM Cloud to create our visualization dashboard rather than using a visualization tool like Grafana. In Figure 6 shows the amount of water used by each household or water meter. Here we can also see that household/water meter 3 consumed the most water. Figure 7 shows the amount of water used by each area. We can see that Area 1 is currently using the most water. Using these figures, a Water Resource Department can do further analysis on them and can also alert people in a certain household or area that they are consuming a lot of water.

C. Web App

The purpose of our Web App is to provide an interface by which people/households can view their current water bill for a month and therefore using this they can decide whether to use less water or not. The Django Python Web framework on IBM Cloud was used to develop this Web App. Figure 8 shows the home page of our Web App. There are two other tabs included in the Web App, a Bills tab and a Graphs tab. The Bills tab allows the people/households to view their bill while the Graphs shows the amount of water used on a daily basis in the 2 areas, current monthly water usage for the 2 areas, and the amount of water used per household/water meter. The Bills tab is shown in Figure 9 while the Graphs tab is shown in Figure 10.

VI. Use-Case Diagram

In this section we describe the projects use-case diagram. Figure 11 has two actors the client and water resource management department (WRD). The WRD performs the following actions; logs the client information into the NoSQL database, performs administration actions which include visualizing the data obtained from the systems cloud into a graph format and the billing information and lastly the billing unit, where the bill amount for the water usage per household are calculated based on the data obtained from the households streaming source and stored into cloud. The client can access the systems web application and view the data in a graph format or can view the billing tab.

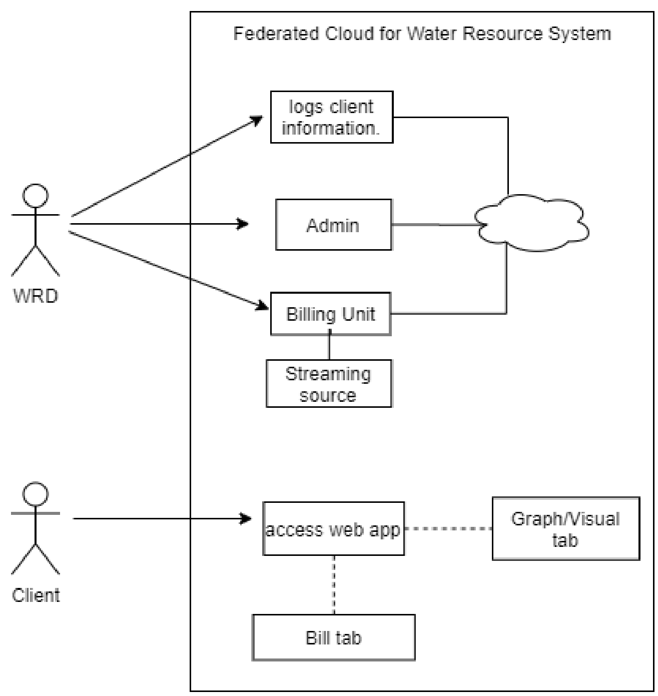


Fig. 11: Use-case diagram.

VII. Individual Experience

A. Mogammad Uzair Jones

I learnt more in this project than in the lecture sessions. I tested out many different services in the cloud catalog to decide what was best.

B. Liyakhanya Candy Tabata

We divided the tasks amongst each other, for my tasks I didn't experience any problems or difficulties based in the work nor working with the group. Everything was easily available and accessible to learn and understand via the internet.

C. Abdullah McDonald

The main problem for me in implementing this project was the restriction on our IBM Cloud accounts. The IBM Streaming Analytics service for example only provide us to stream data at a certain rate, which was quite low, if we went over this rate we would get kicked out and weren't able to use the service on our accounts anymore. Also, some services only allowed us to use it for 30 days so we are not sure whether these services will still be active when our project gets marked. In some cases, like with the Streaming Analytics service, we did find a work-around though which was just using another team member's account. Another problem I found was that of storing our streaming data in a NoSQL Database. This was a problem because we were set on using a visualization tool like Grafana for our visualization dashboard but this couldn't be done because we could only use the Cloudant NoSQL database on IBM Cloud and visualization tools do not support this database. The experiences I've picked up from this project is working as a team and figuring out peoples' strengths to allow for a more fluid working environment.

VIII. Limitations

One of the most problematic limitations were based on documentation. There are a lot of documentation that was outdated where methods could not be used anymore. The stream service on Watson is extremely slow and not all data is stored. Another limitation is costs as some tools are not free. During the assignment, deployment on the continues delivery pipeline would fail yet when we ran it again it would pass and/or fail again and error logs where not helpful in solving the problem. Another limitation was that some services would expire during the development and these services had to be deleted and tasks had to be start on from over to an extent.

IX. Future Improvements

Implementing Fuzzy Systems to determine whether the water usage is low, medium or high based on its area as well as having extra data for clients such as number of members in a household. Due to the time constraint on this project, we could not setup user logins. The could be a helpful feature to use next time as each household will be able to see their water bill independently of others.

X. Conclusion

The objective of this project was to develop a Federated Cloud for Water Resource Management model assuming we are working with the Water Resource Department. Application of the federated cloud has proven to produce adequate results, especially under the conditions and limitations. The developed model was based on IBM Cloud tools, following cloud-computing principles, collaborating three different departments working simultaneously, namely, the client/user information storage, streaming and billing unit, and lastly, the administration department. Under different conditions, the model would be reliable to be applied into real-world conditions.

References

[1] M. Liaqat, V. Chang, A. Gani, S. H. Ab Hamid, M. Toseef, U. Shoaib, and R. L. Ali, "Federated cloud resource management: Review and discussion," *Journal of Network and Computer Applications*, vol. 77, pp. 87–105, 2017.

[2] IBM, "Turning lives around with water," Available at https://www.ibm.com/thought-leadership/water/.

[3] I. Molobela, "Management of water resources in south africa: A review," *African Journal of Environmental Science and Technology*, vol. 5, 12 2011.

[4] K. M. Shahanas and P. B. Sivakumar, "Framework for a smart water management system in the context of smart city initiatives in india," *Procedia Computer Science*, vol. 92, pp. 142–147, 2016.

[5] H. Fuentes and D. Mauricio, "Smart water consumption measurement system for houses using iot and cloud computing," *Environmental Monitoring and Assessment*, vol. 192, no. 9, pp. 1–16, 2020.

[6] S. Shobha, R. Pavithra, K. Pramila, P. B. Morki, and G. Nikitha, "An efficient water management and monitoring system based on iot."

[7] G. L. Harika, H. Chowdary, and T. S. Kiranmai, "Cloud-based internet of things for smart water consumption monitoring system," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2020, pp. 967–972.

[8] S. Rebai, "Resource allocation in cloud federation," Ph.D. dissertation, 2017.