

FORT SYSTEM | GOOGLE CLOUD PLATFORM MIGRATION

Thanushree S

4th year student, Dept of Computer Science and Engg
R V College of Engineering, Bangalore

thanushrees.cs18@rvce.edu.in

ABSTRACT

Now in a day's having all the services and workloads on various platforms is very difficult, time consuming and cost effective due to which many business organization fails upon fulfilling the client needs. Also it needs professionals well versed in all the aspects of software development. If at something goes wrong at some places it difficult to find the exact place where the problem was because it needs assistance from scratch and all the developers and professional needs to be involved. Which takes a lot of effort, so here in this paper we focus on demonstrating how Testing has to be done remotely if any software is installed on some other platforms. Here we focus on migrating the services and workloads from various platform to Google cloud platform.

1. INTRODUCTION

The platform features sophisticated drop-ship capabilities, billing, rating, communication, reporting, and analytic capabilities to enhance efficiencies and optimise supply chain demands, in addition to simple picking and shipping capability. The business has a lot of experience in the field of commerce.

Since JavaScript expertise is a prerequisite for advanced UI interactions in most Web development projects, the idea of using one programming language for all becomes very appealing. The architecture of Node.js makes it simple to use a highly expressive, functional language for server programming without compromising performance and withdrawing from the system.

Application console can also be used separately and can be incorporated into other devices. The technology that has been developed is quicker than current solutions, has more functionality and has been reviewed by automated testing and by actual users. In the research an administrative method was developed for analyzing the MongoDB database scheme. The e-mail directs the user to visit a Web site where they are asked to renew personal information, such as passwords and credit card, social security, and bank account numbers, that the legitimate organization already has. The Web site, on the other hand, is bogus and set up only to steal the user's information.

1.2 Motivation

The motivation of the project has been with a broad customer base and the number of customers rising increasingly day by day which lead to difficulties in assigning cases to SREs manually. As GCP has all the necessary services and workloads on a single platform. Then writing a code, building the project, testing, and deployment become so ease.

The time complexity is very less compared to other cloud providers. GCP is cost efficient. With all these advantages Fort Systems want their services and workloads which are located on different platforms needs to be migrated on to Google Cloud Platform.

1.3 Problem Statement

Modernization and migration of various services and workloads of Fort Systems being deployed on different platform altogether on to Google Cloud Platform(GCP).Each System Reliability Engineer was nominated and allocated cases manually before the Sapphire tool came into existence. The task manager performed this process. This resulted in the division of cases among System Reliability Engineers being unequal. For instance a given SRE may have had to take six cases on a specific day, whilst another SRE may only have been allocated two cases. Such unequal and discriminatory cases were an important reversal. Another problem was that as the amount of cases rose on a certain day, the duty manager had to make much more effort to manually propose and allocate cases on a given day. At this level, manual efforts required mistakes to flow and a more streamlined and efficient method to suggest, plan, and monitor the work of the system reliability engineers had to be followed.

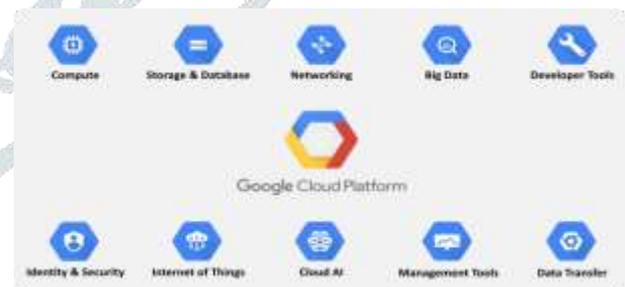


Fig 1- Overview of GCP Services

1.4 Objective

The aims of the project are as follows:

1. Efficient access to all the services, workloads and databases all at one single platform called GCP. Reduced cost, time complexity and latency in switching between various databases.
2. Assess whether cases might surpass the specified period of their SLA. Again, the duty management must
3. Identify, along with the case numbers, the number of cases approved or denied by the System Reliability Engineer. Once this data is collected, publish it on a dashboard that automatically refreshes and enables you to examine and filter the data for an.

4. Identify the seriousness and the greater seniority of the matter based on the feedback of the consumer.
5. Determine if a case should be allocated to SREs with appropriate competence owing to a problem in the core infrastructure and label it accordingly.

1.6 Methodology

The Auto Case Assignment Bot or Sapphire is implemented with NodeJS. The bot depends on a number of additional features and functions that make it work properly. The Google Sheets are continually updated with the availability and expertise of every SRE. Shift information. In MongoDB, these details are found and stored via the Google Sheets API. 8X8 is software which firms utilise at that particular time for assessing staff status and call status. 8X8 is utilised to record information of the SRE call status during a day because if it is currently on the call, the high priority case will not be allocated to the SRE. 8X8 APIs are therefore utilised to obtain the details of how many SREs are present throughout the queue, when Sapphire is going to propose the cases. All of this data, including with the information on who the day's Duty Manager (DM), the slack Identification ID of each and every SRE, the amount of SRE cases chosen every day, etc., are saved in several MongoDB collections. This knowledge is considered by Sapphire's working logic through the REST API before suggesting cases which lie at an SRE in the queue. A case recommendation is produced in SLACK, which covers all SREs through a group conversation. The slack APIs are included in the code. The bot is held and staged on a server through Heroku. All logs produced during operation may be viewed through Timber.IO.

1.5 Procedure

Cloud computing services have the following main characteristics, according to service users:

- The supplier hosts and maintains the site. In their own facility, the cloud hosting provider acquires, hosts, and maintains the essential hardware and software. Users avoid the construction expenditures and maintenance problems that would be incurred if the service was established on-premise.
- Self-service via a web-based interface. Through an internet interface, service users can initiate certain service functions and increase or reduce their service consumption level, with little or no interaction with the service vendor.
- You must pay to utilise the facility. Users of the service only pay for the services they use. When compared to the traditional way of establishing on-site IT capabilities, this can result in significant cost savings.
- designed for highest usage scenarios, only to have that capacity underutilised for the majority of the time.
- Scalability that is nearly endless. Cloud computing service providers usually have the infrastructure in place to deliver their services on a large scale. For customers of cloud services, this means that the cloud can readily handle business expansion or periodic spikes in commission demand.



Fig 2: Classic Cloud Computing Proces

1.7 Types of Cloud Computing Services:

The many services provided by cloud computing firms are usually divided into three categories:

- Infrastructure as a Service (IaaS) is a service that allows you (IaaS). Within the context of a secure data centre, IaaS gives consumers access to raw computing resources such as processor power, data storage capacity, and networking.
- Platform-as-a-Service (PaaS) (PaaS). PaaS products cater to software development teams by providing computing and storage infrastructure as well as a development platform layer that includes web servers, management tools, and software development kits (SDKs) for a variety of programming languages.
- Software As a Service (SaaS) (SaaS). Customer relationship management (CRM), marketing automation, and business analytics are just a few of the application-level services offered by SaaS companies.
 - PaaS clouds are designed by experts, frequently inside IaaS clouds (which we'll discuss further below), to provide scalability and distribution for any application and to help a company's expenses be more predictable.
- The most significant benefit of this service is that for little or no money, you may get started on your application with less stress and development. Because it uses cloud computing, its design provides for a lot of scalability. The sole disadvantage of a PaaS cloud is that it may come with some limitations that will not operate with your products under any situations.
- Infrastructure as a Service (IaaS) is a type of cloud (IaaS)
 - This solution allows businesses to access crucial online infrastructure such as servers, connectivity, and storage space without having to shop for or manage it themselves. The managed service provider and thus the business that uses the infrastructure benefit from economies of specialisation and scale. For example, IaaS lets an online firm to develop and grow quickly. Because the company that is providing the software as a service is also maintaining the infrastructure to execute the programme in the first place, both PaaS and SaaS clouds are based on IaaS. By going with an IaaS cloud, your organisation will have to deal with more complexity, but it will also give you more freedom.
 - This cloud computing service is more advanced, and its terms are similar to those used in cloud computing. Cloud apps make it possible to use the cloud as a software infrastructure. Because the appliance is run on computers owned by the seller, the burden of support, maintenance, and operations is reduced.



Amazon Web Services (AWS) is an Amazon subsidiary (a leading company in eCommerce). Amazon Web Services (AWS) delivers on-demand cloud computing platforms such as storage, data analysis, and other services. Amazon lends its services to individuals, businesses, and governments, with a 35 percent market share. Amazon Web Services members can use a full-featured virtual cluster of computers at any moment, depending on their needs. The entire service is made possible by the internet. To track their own cases, includes a social networking plug-in that

allows users to enter into social networking websites to talk about their companies, provides analysis tools and other services including email alerts, Google research, and customer entitlement and contracts.

1.8 SOAP-based API

It's a type of API that encapsulates data, such as SOAP (Simple object access protocol), which is used to interact across distributed applications. SOAP can be used as a lower-level protocol, such as a http request. The SOAP packet has a header and a payload, and each API request's packets are filed based on the header.

SOAP has evolved into an important component of web services and SOA architecture. Because the majority of the output required is in XML format, SOAP is described at a higher level using the XML format. This structure aids in the clarification of the objectives. The advantage of using SOAP is that it is a lightweight protocol, as is the resulting XML output. It is simple to build with web servers because it uses HTTP to conduct query operation.

1.9 REST protocol

Data records, a collection of records, or a query are all examples of data records. The resources are identified by URIs and can be accessed by HTTP methods such as GET, PATCH, POST, and DELETE. Figure 2.3 illustrates this. To communicate with SFDC, we must first obtain summary information about the API versions that are available to us, as well as complete information on SFDC sets such as users, accounts, and any other custom. Then, to edit or delete records, execute a query. Rest APIs can be stateless, which means that each client-to-server request should include. The server response can either be cached or not. A generic interface is used to access the resources, which are named using a base URI. It works with OAuth2, which is an open mechanism for API authorization. It also supports JSON and XML response types. It has the option of compressing the request and answer. In this instance, the client should include a content-encoding header in the request, allowing the request to be decompressed before processing.

The goal of software testing is to find errors or flaws in individual programme components. During testing, the components are connected to form a complete system. Testing will indicate that the function meets the required functional objectives and does not behave abnormally at this point. The test cases are chosen so that system comportability may be confirmed for all possible combinations.

As a result, the projected system behaviour is defined in a variety of ways. As a result, test cases with expected inputs and outputs, non-valid inputs and acceptable messages, and inputs that are not frequently encountered and may be regarded odd conditions are chosen. Because both hardware and software are involved, the hardware will be checked as well to ensure the system's overall functionality. In an analysis of an experiment, the inputs to the process that define the result of the process are usually evaluated. They also contribute to identifying the relevant inputs and thereby targeting the intended result. To check the accuracy of the system, the system output is compared to the predicted output. For comparison, there are numerous measures. Analysis of the experimental performance checks if the assessment methods are met. This section addresses the system's performance features.



Fig: GCP sample file

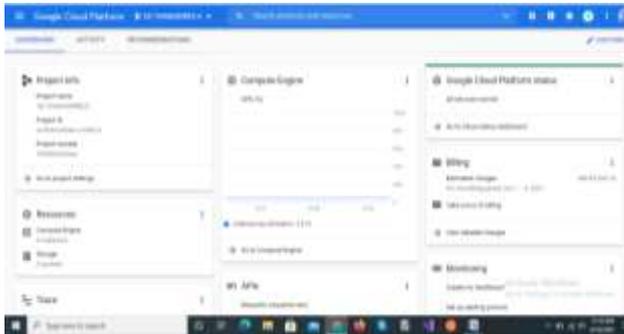


Fig: GCP Dashboard

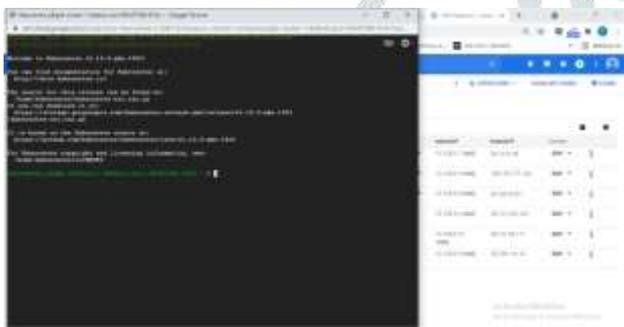


Fig: Implementation of GCP console

1.10 Conclusion and Future Enhancements

With the company's rapid growth, the number of workers and cases to be handled has expanded, necessitating the construction of a more efficient and automated management tool for planning and monitoring the staff's job progress. With an easy-to-use interface and optimised data flow, the proposed approach decreases the trouble and breadth of failures.

The present technology automates the process of assigning cases to staff, decreasing the need for human intervention. Slack's notification system works well and replies quickly. If an employee wants to reject a case, he or she can do so using the case recommendation model. The bot has other features, such as a real-time dashboard that displays information on cases that have been approved or refused. Every day, over a hundred employees benefit from the present system.

2. Limitations of the Project

Although the working system is efficient enough as per requirements, there are a couple of cases of usage which can cause loopholes within the functioning. Some of these limitations of the project are:

- The system requires accurate values on the database as

this code doesn't query to fill the collections on mongoDB.

- The system is very time dependent, so if the code misses any particular iteration then the entire functioning might go wrong.
- The code written is not very modular and few scenarios require hard coding.

2.1 Future Works Of the Project

To overcome the limitations of the project, the following will be taken up in the future:

- The employee seniority will be taken into consideration when assigning the cases and higher priority cases will be given to senior employees.
- For the graceful shutdown of the bot the kill switch with many advantages be introduced.
- Feature health monitoring has to be introduced to ensure that all the modules of the bot are working properly.

2.2 References

- [1] Takeshi Ogasawara, "Workload characterization of server-side JavaScript", 2014 IEEE International Symposium on Workload Characterization (IISWC)
- [2] Stefan Tilkov; Steve Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs", 2010 IEEE Internet Computing
- [3] Andres Ojamaa; Karl Dütina, "Assessing the security of Node.js platform" 2012 International Conference for Internet Technology and Secured Transactions
- [4] Boyu Hou, Lei Li, Yong Shi, Lixin Tao, Jigang Liu, "MongoDB NoSQL Injection Analysis and Detection" 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)
- [5] Benymol Jose Sajimon Abraham "Exploring the merits of nosql: A study based on mongodb" 2017 International Conference on Networks & Advances in Computational Technologies (NetACT)
- [6] Liberios Vokorokos, "MongoDB scheme analysis" 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES) Gerd Wagner, "Introduction to Simulation using JavaScript", 17th International Conference 2016
- [7] Sharath Gude, Munawar Hafiz, Allen Wirfs-Brock, "JavaScript: The Used Parts", IEEE 38th Annual Computer Software and Applications Conference 2014
- [8] Cornelia Györödi, Robert Györödi George Pecherle Andrada Olah, "A comparative study: MongoDB vs. MySQL", 2015 13th International Conference on Engineering of Modern Electric Systems (EMES)
- [9] Yunhua Gu; Shu Shen; Jin Wang; Jeong-Uk Kim, "Application of NoSQL database MongoDB" 2015 IEEE International Conference on Consumer Electronics Taiwan
- [10] Y. G et al., "Analysis of data storage mechanism in NoSQL database MongoDB", 2015 IEEE International Conference on Consumer Electronics-Taiwan, 2015

[11] V. A, M.R. C. Mongo Db, Oracle NoSQL, "A technical critique for design decisions", 2016 International Conference on Emerging Trends in Engineering Technology and Science (ICETETS), 2016 Avramidis and P. L'Ecuyer, "Modeling and simulation of call centers," in Proceedings of the 2005 Winter Simulation Conference, 2005.

[12] O. Garnett and A. Mandelbaum, "An introduction to skills-based routing and its operational complexities," Technion, 2001.

[13] The Force.com Multitenant Architecture Understanding the Design of Salesforce.com's Internet Application Development Platform White.

