Remote workforce check-in system

Dhaniesh M[1], Abilash G[2], Abinaya R[3]

[1] *Student, CSE, Bannari amman institute of technology, Tamilnadu, India*
[2] *Student, ECE, Bannari amman institute of technology, Tamilnadu, India*
[3] *Student, CSE, Bannari amman institute of technology, Tamilnadu, India*

ABSTRACT

Monitoring the workforce in an organization is becoming challenging as the number of resources increases. In addition to that, monitoring resources working remotely is quite hard to manage.

Services using the internet are improving drastically in the modern days with the help of evolving technologies. This problem statement is sorted out by using the mode of internet as a service. Cloud services are used as a medium of data processing. All the data is maintained in AWS (Amazon Web Services) to ensure data is easy to process and provide scalability.

The acronym for Amazon Web Services is AWS. It is a platform for cloud computing that offers a variety of services including processing power, database storage, content distribution, and other functions to assist companies and individuals in scaling and expanding their operations.

Keyword: - AWS Cloud, Terraform, Flask, API gateway

---

1. AWS

AWS, provided by Amazon, is a Cloud Service Provider which provides "Infrastructure as a service". AWS enables other businesses to set up their computing environment virtually anywhere by taking care of primary infrastructure like servers and configuration.

Configurations, in other ways, infrastructure, can be setup and used according to the client's will. A whole product can be deployed in AWS and can be accessed worldwide. It provides a whole variety of services for a whole different purposes. Those services include Lambda, API gateway, S3 Bucket, RDS (Relational Database Service), etc.

1.1 Terraform

AWS, provided by Amazon, is a Cloud Service Provider which provides "Infrastructure as a service". AWS enables other businesses to set up their computing environment virtually anywhere by taking care of primary infrastructure like servers and configuration.

Configurations, in other ways, infrastructure, can be setup and used according to the client's will. A whole product can be deployed in AWS and can be accessed worldwide. It provides a whole variety of services for a whole different purposes. Those services include Lambda, API gateway, S3 Bucket, RDS (Relational Database Service), etc.

1.2 Terraform

Terraform, provided by HashiCorp, an AWS Partner Network, is an "infrastructure as code" tool. It allows the developer to create, version, update and maintain their own AWS infrastructure without taking too much time. The main advantage of terraform is, infrastructure is maintained in code, so that, each and every infrastructure of AWS, like API gateway, Lambda, etc., can be created and updated in a single run.

Additionally, Terraform code can be written, not only for AWS, but also for other cloud service providers like GCP (Google Cloud Platform), Microsoft Azure, IBM cloud, etc. Terraform eliminates the need for creating each and every piece of infrastructure manually, thus it saves time and provides better maintainability.

2. PROPOSED METHOD

Microservice structure

In this section the microservice structure is defined and represented, generally a microservice is a service which works independently in a whole system which is developed individually and then integrated with the other microservices.

The existing system for the project is the legacy microservice i.e SuccessFactors and the current system is Darwin Box in which the data flow happens from the earlier to the later through a publish subscribe system. These systems will interact with each other with the help of this pub-sub model so there is no traffic/distraction between the microservices
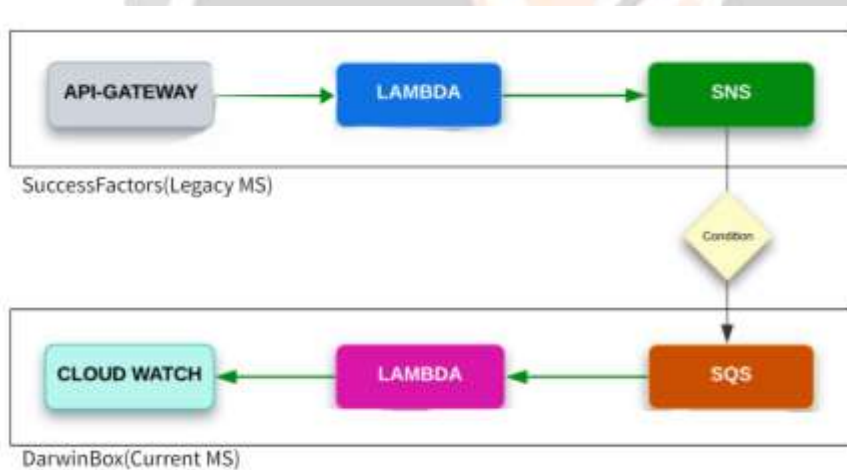

Figure 3.1 Interactions between microservices

2.1 Software Architecture

This software architecture is designed in such a way that the microservices are working and functioning to work together with other microservices present in the model. First the AWS region used here is Mumbai region as its the nearest edge location for India, so the latency of the services is low which can speed up the process internally. Then the data is flowed from Sucessfactors(SF) through a SNS(Simple notification service) and the Darwin Box SQS(Simple notification service) will be subscribing the data, then the data will be pushed to an ingestor lambda. This lambda will insert the data received from the SF into the Database.

Then the API methods (GET, POST, PUT) will be making a request to the Microservice and the response will be given to the requestor through an appropriate lambda. The results are then published to an outbox table which has the history of data flow and data temporarily to be used by other microservices. Here an SES(Simple email service) is used to notify the employee if the user hasn't logged in for the day through an email. Then the changes done in the staff and outbox table is then published to a SNS which then transfers the data to another microservice for further proceedings and storage
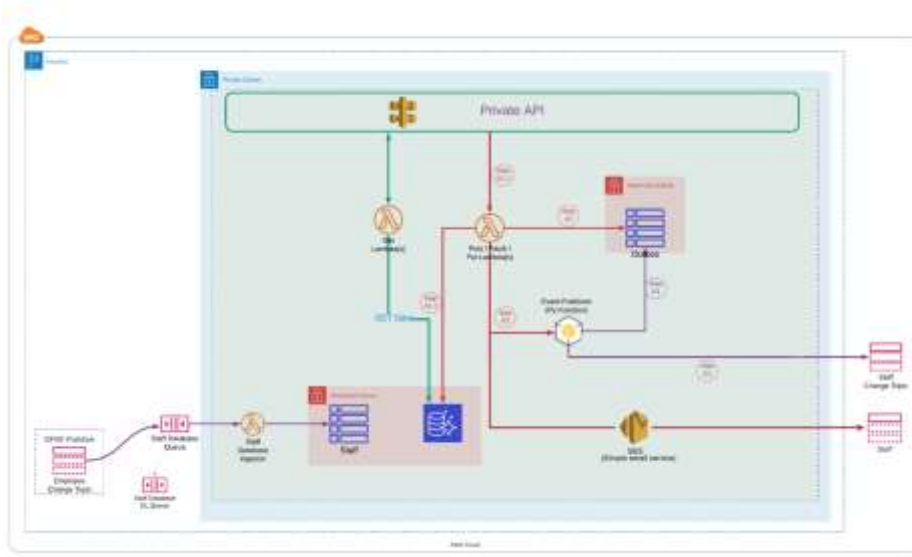

Figure 3.2 Software architecture of microservice

3. Services inside architecture

 Lambda's

Lambda is a serverless event-driven computing service that runs code virtually from almost any application or backend without managing the background computing. It allows to use custom storage called S3 Buckets and DynamoDB through the cloud

API methods

        Currently there are 3 main methods used in the microservice, namely

GET methods:
get_clockin_details
get_employee_details
get_present_details
get_absent_details

get_leave_details

POST methods:

post_employee_details

post_leave_details

PUT methods:

put_employee_details

put_leave_details

: Database connections:

Main table

      The main data table contains the employee details along with their presence details in the database which are connected to the respective tables
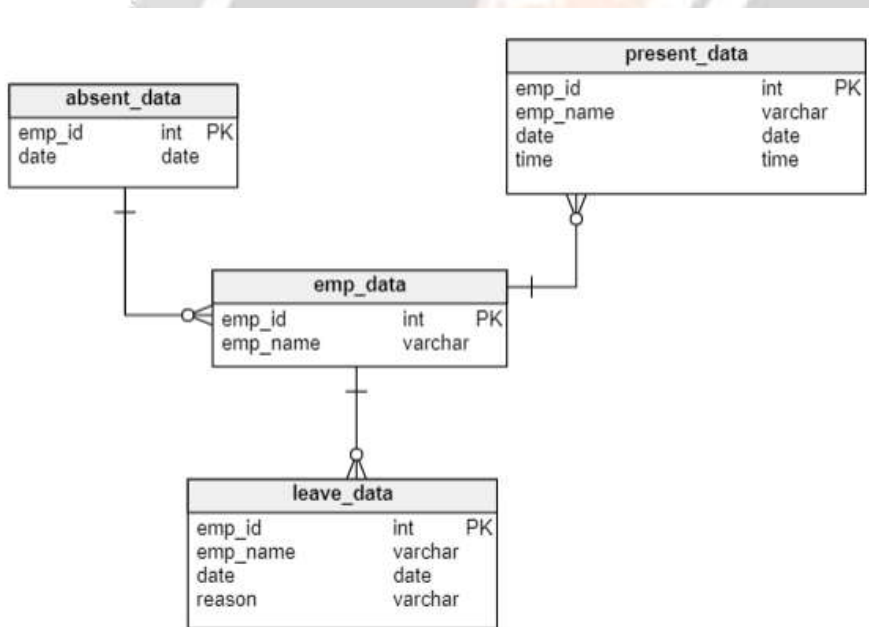

Figure 3.1 Main database table for current project

Outbox table

      The outbox table contains data that are processed by the microservice itself internally, this is primarily to track and use the data processed inside the microservice

Figure 3.2 Outbox database table for current project

4. CONCLUSIONS

Herewith, the project Remote workforce clock in system uses multiple resources like AWS services like API gateway, Lambda, SNS, SQS, SES, micro services, Databases and terraform resources to establish console infrastructure in a code and frontend resources like Flask to implement the frontend modules and rendering templates to have a complete website and to be used by the user

5. REFERENCES

1)   Awa, H.O., Ojiabo, O.U. and Orokor, L.E., 2017. Integrated technology-organization-environment (TOE) taxonomies for technology adoption. Journal of Enterprise Information Management.

2)   Bayrak, T., 2013. A decision framework for SME Information Technology (IT) managers: Factors for evaluating whether to outsource internal applications to Application Service Providers. Technology in Society, 35(1), pp.14-21.

3)   Dutta, P. and Dutta, P., 2019. Comparative Study of Cloud Services Offered by Amazon, Microsoft & Google. International Journal of Trend in Scientific Research and Development (ijtsrd), 3, pp.981-985.

4)   Gandhi, A. and Chan, J., 2015. Analyzing the network for aws distributed cloud computing. ACM SIGMETRICS Performance Evaluation Review, 43(3), pp.12-15.

5)   Kirchgaessner, S., 2013. Cloud storage carries potent security risk. Financial Times. Viitattu, 31, p.2014.

6)   Layo, I. 2013. Cloud Computing Advantages for SMEs. Available online: http://cloudtimes.org/2013/09/18/cloud-computing-advantages-for-smes/ (accessed on 6 December 2020).

7)   Mosca, P., Zhang, Y., Xiao, Z. and Wang, Y., 2014. Cloud security: Services, risks, and a case study on amazon cloud services. Int'l J. of Communications, Network and System Sciences, 7(12), p.529.