Dashboard using Serverless Computing

Suraj R[1], Gurubasava[2]

PG Scholar, School of CS & IT[1]
Assistant Professor, School of CS & IT[2]
Department of MCA[1,2]
Jain Deemed to be University, Bangalore – 560069[1,2]

Abstract : With various cloud platforms it is difficult to integrate all of their services data and cost explorer into one single channel, we present a Dashboard of all cloud providers integrated in one single channel. Monitoring resources on regular basis plays a vital role, instead we use a serverless model to automatically monitor and trigger any events. Experiments show that this helps developers and operation teams in various ways.

1. INTRODUCTION

Serverless architecture makes api calls and integrate with all the services. API gateway which handles all the POST, GET method request and also verifies a authentication token for a valid request, without token every request will get abort by API Gateway, this makes system more secure**.**

From Cloud Billing and Resources Decommissioning Dashboard, we can list and delete the unused AWS service’s resources. so that our company does not need to pay for any unused AWS resources. By deleting unused AWS resources, we can control cost and from dashboard we can analysis which AWS services is costs more so that we can make decisions accordingly

.

2. Literature Review

Resources control from Cloud Billing Dashboard, we can list and delete the unused AWS service’s resources. so that our company does not need to pay for any unused AWS resources. Cost control by deleting unused AWS resources, we can control cost and from dashboard we can analysis which AWS services is costs more so that we can make decisions accordingly. Multi-Account resource’s control. We have four accounts like production, training, overseas and developer. For each account we can list and delete resources from dashboard, so we can clean up the AWS resources after every training, corporate training

.

3. Context Flow Diagram

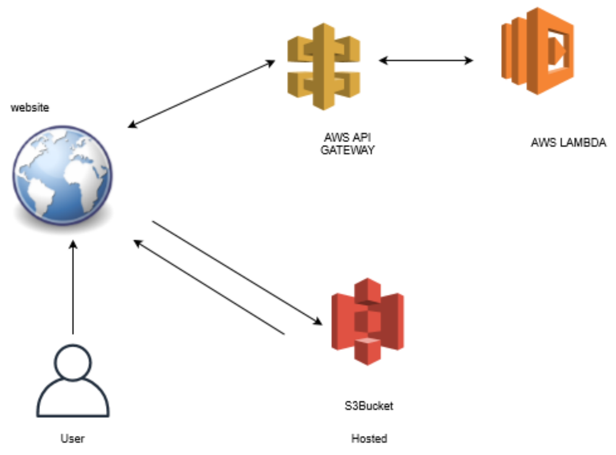

Figure 1 CONTEXT FLOW DIAGRAM

4. PROPOSED SYSTEM

We are using AWS serverless architecture for our project to achieve this we are using AWS services like AWS S3 Bucket, Cognito, API Gateway, Lambda function. serverless architecture let user build and run application and service without thinking about server. In this architecture, the static web pages are uploaded to S3bucket for web hosting. To authenticate that users are from CloudThat only we are using AWS Cognito, To make a request to main logic also known as AWS Lambda, we are using mediator call as API gateway which handles all the POST, GET method request and also verifies a authentication token for a valid ajax request, without token every ajax request will get abort by API Gateway, this makes system more secure, authentication token can be retrieve when user sign-in into dashboard. Finally, AWS Lambda Executes function as per request from API Gateway and sends response back to Front-End.

It is useful for a daily monitoring of AWS Service's resources in each account and for the analyze cost of each service. Also Cloud Billing Dashboard helps to delete unused AWS service's resources.

5. Design Architecture

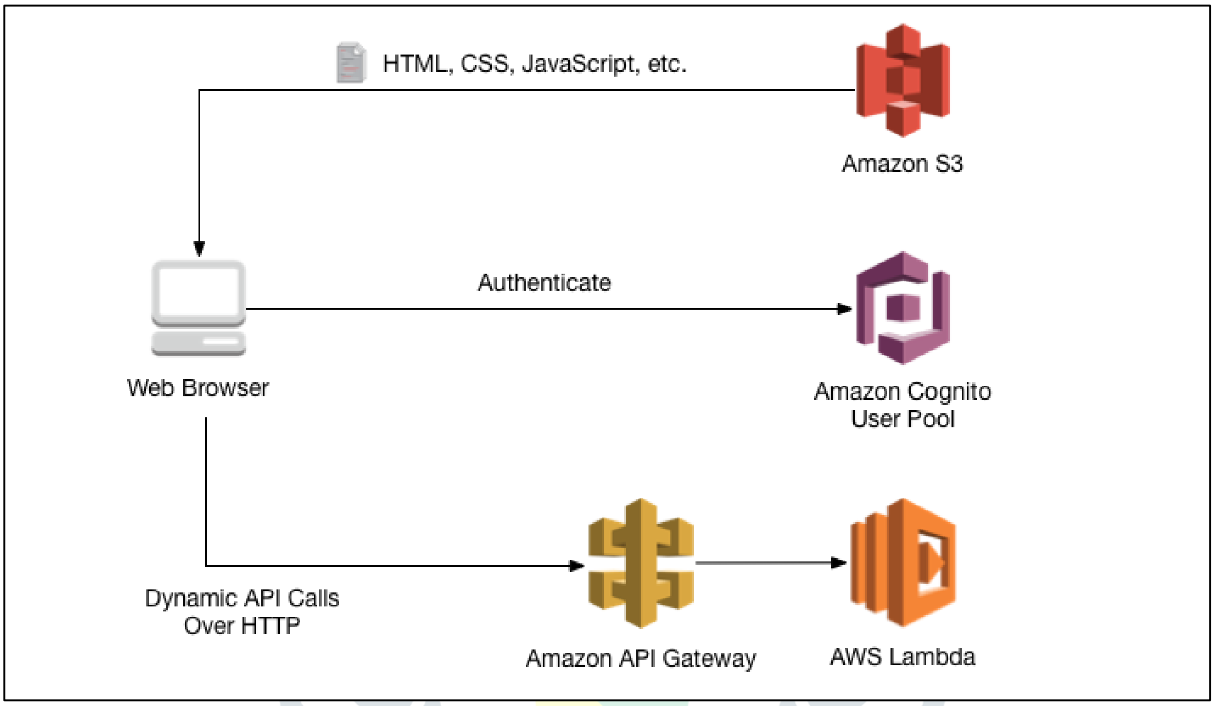

FIGURE 2 – DESIGN ARCHITECTURE

- API gateway which handles all the POST, GET method request and also verifies a authentication.
- AWS Lambda Executes function as per request from API Gateway and sends response back to Front-End.

6. IMPLEMENTATION

The main motive of coding is to create a design in the best possible way. The coding activity effects on both testing and operations. The time spent in coding is much lesser then that of the total software cost, whereas testing and maintenance consume the most. Thus, it should be clear that the motive during coding shouldn't be to reduce implementation cost, but the motive is to reduce the cost in later process, even if it means the motive during the process is not the job of the coder. Rather, the goal should be the job of the tester and maintainer or operations. It is important that the programmers are often concerned about how to complete their job quickly, without keeping the next phase. This thought should be kept in mind that the program should not be constructed easy to write, so that they are easy to read and understand, program is read more and by lot more people during later phases. There are many different ways for judging a program, including read method, size of the program, execution time and memory utilized. Having readability and understand ability as a clear objective of the coding can itself help in producing software that is more maintainable. So in serverless computing we only write the code the execution and maintenance is done by the service.

7. CONCLUSION

In this project we have implemented a dashboard using Lambda and api gateways. This makes easier for the users and for an organization to keep track of resources. Using AWS Cognito pool we have saved our organizations users list so only they can access the site. We are able to successfully use the Dashboard with full optimization and integrated with Lambda. We will be adding multiple cloud platforms in this Dashboard. The running services and cost report are being generated for all accounts. Additional Stack Creation and User creation has been working successfully.

8. FUTURE ENHANCEMENT

In future We will implement auto deletion of resources, when AWS service's resource exceeds the amount of cost which we have define to each resource. It will trigger AWS Lambda for deletion. Automatic Stack and User creation using Lambda.

9. REFERENCES

1. https://docs.aws.amazon.com/lambda/latest/dg/welcome.html
2. https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools-working-with-aws-lambda-triggers.html
3. https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-app-integration.html