Cloud Based Productivity Web Application

Aditi Patil
Department of Engineering and Computer Science
California State University, Fullerton
Fullerton, California, USA
aditipatil138@csu.fullerton.edu

Anjali Iyer
Department of Engineering and Computer Science
California State University, Fullerton
Fullerton, California, USA
anjali.iyer@csu.fullerton.edu

Ankita Kulkarni
Department of Engineering and Computer Science
California State University, Fullerton
Fullerton, California, USA
ankitak@csu.fullerton.edu

Shraddha Yadav
Department of Engineering and Computer Science
California State University, Fullerton
Fullerton, California, USA
shraddhayadav@csu.fullerton.edu

***Abstract*—The goal of this project is to design, implement, and evaluate a cloud based productivity web application on a cloud environment. The traditional cloud based storage is centralized without considering the location of the data. The user will be able to add, sort, delete and mark complete the tasks for the day. If the user wishes to receive an email with the status of their to-dos, they can do so by clicking on a button within the application. Only authorized users will be able to use this application service (User Authentication).**

***Keywords*— Cloud, React.js, AWS, Amplify, Cognito, DynamoDB, GraphQL.**

I. Introduction

2020 will be remembered as the year in which the COVID-19 pandemic brought our "normal" lives to a grinding halt. We have been forced to stay at home and practice social distancing in an effort to "flatten the curve", so to speak, and reduce the spread of the coronavirus. This has led to a "work-from-home" system, where the working class no longer needs to commute to and from work. There are many benefits to the work-from-home system, such as virtually no commute time and increased family interaction. However, there are certain drawbacks as well, such as loss of productivity and increased procrastination. In this paper, we propose to build a cloud based productivity web application that helps its users to make daily to-do lists, which will ultimately help them improve productivity and overall mental satisfaction. The app will have a feature that emails the user their to do list status at the end of the day, in order to remind them to stay on track.

II. Cloud Computing Architectural Framework

Cloud computing integrates different computing systems to provide end-users with services. It is necessary for us to briefly introduce the principles that apply to cloud computing to understand the security issues related to cloud computing. A commonly agreed definition of cloud computing by the National Institute of Standards and Technology (NIST). NIST concept sees cloud computing as a three-fold service delivery model comprising: (a) core functionality, (b) model service and (c) model implementation. In the light of the NIST description, the cloud computing definitions are shown below.

2.1 ESSENTIAL CHARACTERISTICS

2.1.1 On demand service

Without human contact with the CSP, customers can request and access services from the cloud. The services and related resources are given as and when necessary. Typically this is done via web services and management interfaces.

2.1.2. Broad network access

Using standard frameworks and protocols, the customer's services as well as applications and data present on the cloud must be available. The function also requires that resources be available to serve a heterogeneous thin or thick setting (for example, mobile phones, laptops, workstations, tablets). Broad network access in literature is often called omnipresent network access.

2.1.3. Resource pooling

The resources of the Cloud are pooled into a multi-rental environment between many customers. The position of the services is clear for the customers. A map is made accessible to customers between physical and virtual tools.

2.1.4. Rapid elasticity

The resources can be extended quickly and elastically as needed by the customer. The customer has an overview of limitless resources which can be bought as required.

2.1.5. Measured service

The upward and downward scaling of capital takes place dynamically and the customers and CSP are aware of the use of these services. The calculation also automatically helps to maximize the usage of services as users are redeemed as they are paying for.

2.1.6. Multi-tenancy

The NIST describes the above five cloud computing functions. The Cloud Security Alliance (CSA) therefore adds multi-tenant technology as an essential cloud computing feature (although not an essential characteristic). Multi-tenancy is the property which allows multiple clients who can belong to the same organization to use a single

resource. Multimedia results in maximum resource usage and logically distinguish various customers.

2.2. SERVICE MODELS

The NIST divides the services provided by the cloud computing into three categories, namely: (a) software as a service (SaaS), (b) platform as a service (PaaS), and (c) infrastructure as a service (IaaS). The cloud service model is referred to as SPI (software, platform, and infrastructure), (d) Function as a Service (FaaS), (e) Integration Platform as a service (iPaaS), (f) Identity as a service (IDaaS)

2.2.1. SaaS

This form of public cloud computing offers internet applications through the browser. Google's G Suite and Microsoft's Office 365 are the most common SaaS applications for companies; Salesforce is leading the way among corporate applications. But almost all firms, including Oracle and SAP ERP suites, have taken the SaaS model. SaaS applications usually provide comprehensive configuration options and development environments that allow customers to program their own additions and modifications.

2.2.2. PaaS

PaaS offers collections of resources and workflows that directly target developers, who can use shared tools, processes, and APIs to accelerate the creation, testing and deployment of applications. Salesforce's Heroku and Force.com are common public cloud PaaS offerings; Pivotal's Cloud Foundry and Red Hat's OpenShift can be installed on premises or accessed via the major public clouds. For companies, PaaS will ensure developers have ready access to resources, monitor certain processes and only use a certain set of services while operators manage the infrastructure underlying them.

2.2.3. IaaS

At the fundamental level, public cloud IaaS providers provide pay-per-use storage and computing services. However all of the major public cloud companies provide a broad variety of services – extremely scalable databases, virtual private networks, big data processing, machine learning, app monitoring, etc. The leading IaaS provider was Amazon Web Services, followed by Microsoft Azure, Google Cloud, and IBM Cloud.

2.2.4. FaaS

FaaS, the cloud version of serverless computers, introduces another abstraction layer to PaaS, so developers can be totally insulated from everything under their code in the stack. They upload narrowly usable blocks of codes instead of using virtual servers, containers and applications, and set them to be triggered by a certain event (such as a form submission or uploaded file). FaaS is a top of IaaS in all big clouds – AWS Lambda, Azure Functions, Google Cloud and IBM OpenWhisk. FaaS programs have a special advantage in that they will not use IaaS services before an incident happens, which eliminates pay-per-use costs.

2.2.5. iPaaS

Data integration is a key issue for any large business, but especially for SaaS companies. iPaaS providers usually provide prebuilt data sharing connectors between common SaaS applications and company on-site applications, although providers may focus more or less on the integration of B-to-B and e-commerce applications, cloud integrations or traditional SOA types. As a part of the integration-building phase, the iPaaS cloud offers from suppliers such as Dell Boomi, Informatica, MuleSoft and SNAPLogic can also be adopted by users.

2.2.6 IDaaS

Cloud storage management and its related privileges and permissions are the most challenging problem in terms of protection on private data centers and public cloud sites. IDaaS providers maintain cloud-based user accounts that authenticate users and permit the access to security policies, user groups and individual privileges for resources or applications. It is critical that we can integrate and provide different directory services (Active Directory, LDAP, etc.). CA, Centrify, IBM, Microsoft, Oracle and Ping have cloud applications on site. Okta is the pioneer in cloud-based IDaaS.

2.3. DEPLOYMENT MODELS

There are four models that can be used to deploy a cloud computing infrastructure, namely: (a) private cloud, (b) public
cloud, (c) community cloud, and (d) hybrid cloud.

2.3.1. Private cloud

The infrastructure used to manage public IaaS clouds is reduced to a privately-owned cloud, which can be installed and managed in the data center of a client. Like a public cloud, internal customers can create, test and run applications using their own virtual resources and reload metering from resource usage departments. The private cloud is the pinnacle in data center automation for administrators and minimizes manual supply and management. The stack of VMware Software Based Data Center is the most common private commercial cloud software and the OpenStack leader.

However, note that the private cloud does not completely comply with the cloud computing concept. It was a service for cloud computing. A private cloud allows a company to develop and manage its underlying cloud infrastructure; it is perceived as a cloud computer service only by internal cloud users.

2.3.2. Public cloud

The physical architecture of the cloud is CSP-owned and is accessible to the general public. All customers are provided with services. Customers compensate the owner of the cloud according to their services and money. The physical infrastructure is customer-friendly and is operated by the CSP.

2.3.3. Community cloud

A variety of organizations and/or customers that create a community share the community cloud. The community generally shares common interests, such as mission, safety

requirements and policy considerations and compliance issues. Any company in the community or a third party may control the community cloud. It can be placed in or outside the premise in the same manner.

2.3.4. Hybrid cloud

The incorporation of a private cloud into a public cloud is a hybrid cloud. The most innovative of its kind is the hybrid cloud, where apps can seamlessly switch between public and private clouds. In other situations, the databases can remain in the customer data center and integrate with public cloud applications—or, during high demand periods, virtualized data center workloads may be diverted to the cloud. The types of cloud-based integration differ widely, but they need to be comprehensive to be designated in hybrid clouds.

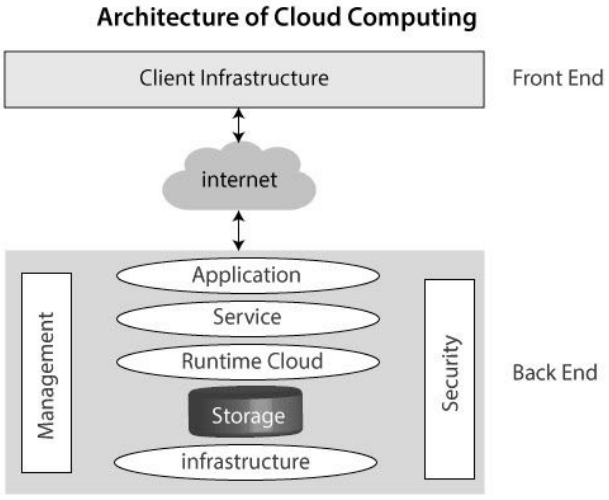

Figure: Cloud Computing Architecture

III. Cloud Services

A. Amplify

AWS Amplify is a complete solution for the building and deployment of stable, scalable, full-post, AWS driven applications by mobile and front end web developers. You can set up apps within minutes with Amplify, connect them to your app in a few lines of code and use static web apps in three steps.

B. Cognito

Amazon Cognito helps you to quickly and easily add user registration, authentication and access control to your site and mobile apps. Amazon Cognito is able to reach millions of users and supports sign-in through SAML 2.0 with providers of social identity, such as Facebook, Google and Amazon.

C. GraphQL

GraphQL is an API language that lets you easily query and handle data through an intuitive and versatile syntax. GraphQL offers a syntax that defines data needs and interactions so that you can ask exactly what you need and get consistent answers. In this way, the numbers of network calls and bandwidth requirements can also be minimized in a single application, saving the battery lifetime and CPU cycles that the applications consume. Updates to your data are made easy by transformations, so you can explain how you want to modify your data. You also can install real-time solutions via subscriptions fast with GraphQL. Combined with powerful developer tools, all these features make GraphQL vital for your database management.

D. DynamoDB

Amazon DynamoDB is a storage of key value and documents offering an output of 1 digit millisecond in any size. It is a fully-managed, multi-regional, multimedia, long-lasting, stable, back-up and in-memory caching database for Internet applications. With over 10 billion requests a day, DynamoDB can accommodate peaks of more than 20 million requests per second. Many of the world's most rapidly expanding businesses, such as Lyft, Airbnb, and Redfin, as well as companies such as Samsung, Toyota and Capital One, rely on DynamoDB 's degree and efficiency for supporting their tasks. AWS customers of hundreds of thousands select DynamoDB as their smartphone, web, ad tech, IoT and other low-latency data access database. The data are available in all sizes. Build a new table and let DynamoDB deal with the rest of your submission.

IV. Front End Technologies

A. ReactJS

ReactJS is a JavaScript library used to create UI components that can be reused. The following is the description, according to React's official documentation. React is a library for designing user interfaces that are composable. It promotes the development of UI components that are reusable, which present information that changes over time. Offering a simpler programming model and better results, React abstracts the DOM from you. React can also render using Node on the server, and using React Native, it can control native apps. React provides one-way reactive data flow, reducing the boilerplate and making it simpler to reason than standard data binding.

V. Proposed Work

Weeks	**Milestones of the week towards your final deliverables of your project.**
Week1	Project topic selection. Language selection for the application.
Week2	Getting Familiar with React.js.
Week3	Getting Familiar with AWS services (Simple Email Service, Cognito, DynamoDb, GraphQL and Amplify).

Week4	Implementing theoretical knowledge gained by developing a mini-project on React.js and AWS
Week5	Backend development.
Week6	Backend development.
Week7	Frontend development
Week8	Integration of the Frontend with the Backend
Week9	Hosting application on AWS
Week10	Testing the application for bugs and errors

VI. Implementation

A. Key Functionalities

Whenever the user visits the application, the very first thing he sees is the sign-in page. This web application provides access only to authorized users. Therefore, if the user already has an account, he can just simply login with a valid email address and password. However, if a user is new and does not have an account, he must create his account first in order to access the application through the "create account" option provided on the login page.

To create an account user must provide a valid email address as the username and a strong password with a combination of special characters, upper- and lower-case letters. In case if the user forgets their password, the app does provide an option to reset the password. Through the "Reset Password" option provided on the login page and users can reset their new password.

After creating an account user can login into the application successfully with valid credentials. As soon as the user logs in, the very next page user sees is the Amplify Todos. To create a to-do user has to enter the name of the task they want to perform in the provided name field. Along with this, users can even enter the description for the to-do task they want to achieve in the description field. For instance, if a user wants to watch a movie, he can add that in the Name field and the details of the movie in the description box.

Further, the application even provides a feature that sends an email to the users daily at the end of every day to let them know of their progress so far, how many todos they have completed, and how many are remaining. To implement this feature, we have used (SES) simple email service from aws. The application also offers an option to set the to-do task according to their priority levels, high, medium, and low. This feature allows users to sort their most and least important task as per their requirement.

Lastly, to sign out from the application, the user can avail sign-out option provided on amplify todos page. Upon sign out, the user will not lose any of the tasks created in the app, and to store all this data we have used DynamoDB. For this the DynamoDB console shows different details such as the time task "created At", "updated At" and the name of Todo.

VII. Future Scope

[1] Developing a Reactive Native App for iOS and Android based devices.

[2] We can add Machine Learning Libraries to auto suggest users tasks that are added repeatedly, at regular intervals.

[3] We can add a reminder for a task so that the user gets an alarm or a push notification about the task.

[4] We can integrate this application with third party applications like Google Calendar or iCal for seamless access.

[5] Sorting the task according to the priorities, high, medium and low.

VIII. References

[1] Thomas Erl, Ricardo Puttini, Zaigham Mahmood, (May 10th 2013) - Copyright © 2013. "Cloud Computing: Concepts, Technology & Architecture, 1st edition".ISBN-13: 9780133387520.

[2] "Enhancing Database Security in Cloud Computing." International Journal of Recent Technology and Engineering 8.6 (2020): 393-97.

[3] Qiang, W. Performance and security in cloud computing. J Supercomput 75, 1–3 (2019). https://doi-org.lib-proxy.fullerton.edu/10.1007/s11227-018-2671-4, Published 22 January 2019, Issue Date 09 January 2019, DOI: https://doi-org.lib-proxy.fullerton.edu/10.1007/s11227-018-2671-4

[4] A. J. Choudhury, P. Kumar, M. Sain, H. Lim and H. Jae-Lee, "A Strong User Authentication Framework for Cloud Computing," 2011 IEEE Asia-Pacific Services Computing Conference, Jeju Island, 2011, pp. 110-115, doi: 10.1109/APSCC.2011.14.

[5] Sivasubramanian, S. (2012). Amazon dynamoDB. Proceedings of the 2012 International Conference on Management of Data - SIGMOD '12. doi:10.1145/2213836.2213945

[6] Michael J. Kavis, (2014)."Architecturing the Cloud: Design decisions for cloud computing service models (Saas, PaaS, IaaS)". ISBN 978-1-118-61761-8, ISBN 978-1-118-82627-0 (epdf).

[7] AWS Amplify: Console User Guide Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All

https://docs.aws.amazon.com/amplify/latest/userguide/amplify-console-ug.pdf