

**University Of Petroleum and Energy Studies, Dehradun**

Diagram

Description automatically generated

**Cloud Application Deployment.**

**Submitted by : Shivam Kumar**

**Batch- B1 (CCVT H)**

**SAP ID- 500083476**

**Enrollment Number- R2142201085**

**Submitted to: Dr. Harvinder Singh**

**A serverless web application.**

We will use the AWS cloud service provider for our project because it offers an on-demand, scalable, secure, and fully managed platform.

We need a database, a serverless tool, a service to trigger the serverless tool, an authentication service, a service to automatically build and deploy our updated code (CI/CD), and a repository to store and manage our code for our project.

Amazon Cognito, the lambda function, the API gateway, and DynamoDB can all meet the above requirements.

AWS's services are simple to use and implement, scalable, safe, and offer pay-as-you-go pricing.





**Application platform and reason (contd.)**

1. **AWS Amplify:**AWS provides this service, which enables users to construct and deploy a full-Stack application that is highly scalable and secure.

Therefore, if we were not employing the cloud-based solution, we would be required to manage the servers and backend applications as well as establish our own infrastructure.

Also, because amplify gives us a public link that our customers can use to get to our website, we don't have to buy a public DNS for the first few months, which saves money. Additionally, since we have utilized our on-premise server, we must be concerned about establishing the appropriate environment to run our code and manage the physical servers.

1. **AWS Cognito:** It is a service provided by Amazon Web Services that assists us in authenticating and authorizing users who access our website or application.

A user pool and an identity pool are managed by Cognito. In the user pool, it is in charge of user data and authentication, while in the identity pool, users will receive permissions (authorization) after successful authentication.

The authorisation and authentication procedures, on the other hand, are extremely complicated and difficult to carry out if we employ the on-premise strategy (or are simply impractical for small startups).

As a result, it makes our website safer and makes it easier to filter out unknown users.

1. **DynamoDB:** It is the completely overseen NoSQL data set given by the AWS to store the information as lines and sections with high adaptability. Because it is highly scalable, we only need to pay for the storage we use in relation to our traffic.

Therefore, in the past, when expenditure costs were low, we did not need to consider the costs associated with storage capacity.

1. **Lambda function:** It enables us to execute our code on compute resources with high availability. It can scale automatically and is activated by an event.

We are making our application serverless with the assistance of the cloud lambda function. In addition, the lambda function scales resources in the event of a sudden increase in high demand and provides us with one million instructions at no cost. On the other hand, if the setup were done on-premise, it would have been extremely challenging to process, manage, and respond to a million instructions. Additionally, it would have necessitated significant capital expenditures and operational expenditures, as well as highly skilled IT professionals.

**Flowchart of the services (Designed application of public cloud-AWS):**



**Literature Review:**

[1] G. McGrath and P. R. Brenner, "Serverless Computing: Design, Implementation, and Performance," 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 2017, pp. 405-410, doi: 10.1109/ICDCSW.2017.36.

[2] R. A. P. Rajan, "Serverless Architecture - A Revolution in Cloud Computing," 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 2018, pp. 88-93, doi: 10.1109/ICoAC44903.2018.8939081.

[3] D. Kelly, F. Glavin and E. Barrett, "Serverless Computing: Behind the Scenes of Major Platforms," 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), Beijing, China, 2020, pp. 304-312, doi: 10.1109/CLOUD49709.2020.00050.

[4] S. Gandhi, A. Gore, S. Nimbarte and J. Abraham, "Implementation and Analysis of a Serverless Shared Drive with AWS Lambda," 2018 4th International Conference for Convergence in Technology (I2CT), Mangalore, India, 2018, pp. 1-6, doi: 10.1109/I2CT42659.2018.9058237.

[5] S. Neela, Y. Neyyala, V. Pendem, K. Peryala and V. V. Kumar, "Cloud Computing Based Learning Web Application Through Amazon Web Services," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 472-475, doi: 10.1109/ICACCS51430.2021.9441974.

[6] R. A. P. Rajan, "Serverless Architecture - A Revolution in Cloud Computing," 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 2018, pp. 88-93, doi: 10.1109/ICoAC44903.2018.8939081.

[7] J. Surbiryala and C. Rong, "Cloud Computing: History and Overview," 2019 IEEE Cloud Summit, Washington, DC, USA, 2019, pp. 1-7, doi: 10.1109/CloudSummit47114.2019.00007.

[8] J. Surbiryala and C. Rong, "Cloud Computing: History and Overview," 2019 IEEE Cloud Summit, Washington, DC, USA, 2019, pp. 1-7, doi: 10.1109/CloudSummit47114.2019.00007.

[9] S. Namasudra, P. Roy and B. Balusamy, "Cloud Computing: Fundamentals and Research Issues," 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), Tindivanam, India, 2017, pp. 7-12, doi: 10.1109/ICRTCCM.2017.49.

[10] S. Patidar, D. Rane and P. Jain, "A Survey Paper on Cloud Computing," 2012 Second International Conference on Advanced Computing & Communication Technologies, Rohtak, India, 2012, pp. 394-398, doi: 10.1109/ACCT.2012.15.