

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

Diagram

Description automatically generated

**Cloud Application Deployment.**

**Submitted by : MOHIT BISHESH**

**Batch- B4 (CCVT NH)**

**SAP ID- 500087456**

**Enrollment Number- R2142201858**

**Submitted to: Prof. Saurabh Shanu**

**A serverless taxi booking web application.**

**Concept Note**

In this project we are going to build a taxi booking system which will be a serverless web-app and will enable the customer to book a ride from a preferred location. The user can choose the location of pickup from the map, and for which we will be using [ArcGIS.com](https://www.arcgis.com/sharing/oauth2/authorize?client_id=arcgisonline&response_type=code&state=%7B%22portalUrl%22%3A%22https%3A%2F%2Fwww.arcgis.com%22%2C%22uid%22%3A%227UZdnOFSpqMiJpfZnPnWugR17bhlt0B5fbxGeeqlimw%22%2C%22useLandingPage%22%3Atrue%2C%22clientId%22%3A%22arcgisonline%22%7D&expiration=20160&locale=en-us&redirect_uri=https%3A%2F%2Fwww.arcgis.com%2Fhome%2Faccountswitcher-callback.html&force_login=true&redirectToUserOrgUrl=true&code_challenge=749Ieee-5F7xsN60kDHzblWUiBOTUZkTSKpvtlf8-e0&code_challenge_method=S256&display=default&hideCancel=true&showSignupOption=true&canHandleCrossOrgSignIn=true&signuptype=esri&allow_verification=true) to show the map.

We will deploy our project in AWS and will use the AWS services such as, AWS Amplify, AWS Cognito, API Gateway, Lambda function, and DynamoDB.

We will Upload our HTML, CSS, and JS and other website content file on AWS Amplify which will build a backend server for us.

Now whenever the user will interact with our website, he will get the option to authenticate first. So he can do so either by signing through the google or Gmail or via his actual existing e-mail id so that only the trusted and genuine users can access our website and book the ride.

As the user will be authenticated, he can choose the pickup location, after which APIs will be called through the API Gateway which will trigger the lambda function to finally book a ride and store the data of the user in the DynamoDB.

Here as the data is being stored to DynamoDB by trigging lambda function, so its making it a bit dynamic.

Once the user will successfully book the ride, a display will appear which will show the booking confirmation.

**Why we are deploying our web-app on cloud.**

1. AWS Amplify: it is a service offered by AWS which helps the users to build and deploy a full-Stack application with highly scalability and security.

So, if we were not using the cloud based solution, then we need to setup our own infrastructure, and need to manage the servers and the backend application.

Also, amplify provides us a public link through which the customer can access our website, so we do not need to purchase for a public DNS for our website in the initial phases and hence is cost effective. Also we have used our on premise server, then we need to be worry about to setup the proper environment to run our code and manage the physical servers.

1. AWS Cognito: It is a service offered by AWS which helps us for Authentication and authorisation of the users accessing our website or application.

Cognito maintains user pool and identity pool. Under user pool, it maintains tha data of the user and is responsible for authentication whereas identity pool will provide the permissions (authorization) to the users to after the successful authentication is done.

Whereas, if we have used on premise strategy, then the authorisation and authentication processes are really complex and challenging to perform (or merely impossible to be done for small Sartups).

Hence, it increases the security of our website and easily filters the unknown (unauthorized users).

1. DynamoDB: It is the fully managed NoSQL database provided by the AWS to store the data in the form of rows and columns with high scalability. It is highly scalable so we only need to pay for the amount of storage we will use as per our traffic.

So, in our previous days, with low Expenditure cost, we do not need think for the overhead of storage capacity.

1. Lambda function: it helps us to run our code on high availability compute resources. It is event triggered and has the capacity to automatically scale.

with the help of lambda function on cloud, we are making our application as serverless. The lambda function also provides us the 1 millions of instruction for free of cost and even responsible of scaling of resources in case of sudden increase in high request. Whereas, if was very difficult to process, manage and respond to a million of instructions in case of on premise setup and would have required high capex and opex with high skilled IT professionals.

**Flowchart:**

API calls

AWS cognito

Authenticate

HTML, CSS, JS

DynamoDB

AWS lambda

AWS API Gateway

AWS Amplify

Web Server

**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.