**A Mini Project Report**

on

“DESIGN & DEPLOYMENT OF IAAS FOR SERVERLESS WEB APPS”

Submitted to

CLOUD COMPUTING LAB

(20BT61231)

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

*Submitted by*

|  |  |
| --- | --- |
| **A KOUSHIK** | **21121A3501** |
| **A JAMES** | **21121A3502** |
| **C POOJITHA** | **21121A3503** |
| **C NAVEEN** | **21121A3504** |
| **CH AKHILESH** | **21121A3505** |



**Department of Information Technology**

**SREE VIDYANIKETHAN ENGINEERING COLLEGE**

(AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE, Accredited by NBA & NAAC) Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2023-2024

**ABSTRACT**

Web application deployment using serverless services has emerged as a transformative approach in the realm of cloud computing. This paradigm shift in deployment architecture offers numerous advantages, including enhanced scalability, reduced operational overhead, and optimized cost management. In this paper, we delve into the core concepts of serverless computing, elucidating how it fundamentally alters the traditional deployment model by abstracting away infrastructure management. Through a comprehensive exploration of leading serverless platforms such as AWS Lambda, Azure Functions, and Google Cloud Functions, we elucidate the mechanics of deploying web applications in a serverless environment. Furthermore, this addresses the challenges inherent in serverless deployment, such as cold start latency, resource limitations, and vendor lock-in concerns, and proposes strategies for mitigating these issues. By leveraging event-driven architecture and microservices principles, developers can architect resilient and scalable web applications that adapt dynamically to fluctuating workloads. Real-world case studies and best practices are presented to provide practical insights into effective serverless deployment strategies, illustrating its applicability across diverse use cases, ranging from e-commerce applications to IoT solutions. Ultimately, this addresses the transformative potential of serverless computing in revolutionizing web application deployment, fostering innovation, agility, and cost efficiency in the modern software development landscape.

**KEYWORDS :** Lambda, API Gateway, CloudFormation, DynamoDB, AWS S3, SAM, Cognito.

**INTRODUCTION**

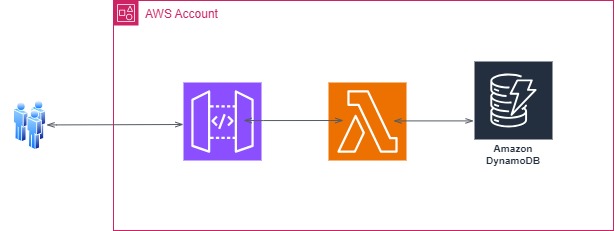
Web application deployment using serverless services represents a pivotal shift in modern software development, offering developers a streamlined approach to deploying and managing applications in the cloud. This document outlines a comprehensive procedure encompassing various steps to facilitate the deployment of a web application utilizing serverless architecture on cloud platforms such as AWS. By adhering to the prescribed steps, developers can effectively harness the advantages of serverless computing, including enhanced scalability, cost-effectiveness, and simplified operational management.

Testing the application is the subsequent imperative, necessitating the use of the invoke URL to access the application in a new tab. Developers are encouraged to interact with the application, filling out the provided form and submitting it to verify the successful storage of user details in the DynamoDB table.

Deploying web applications using serverless architecture on cloud platforms like AWS offers numerous advantages to developers. Firstly, it enables enhanced scalability, allowing applications to automatically scale up or down based on demand without the need for manual intervention. This dynamic scalability ensures optimal performance even during peak usage periods while minimizing costs during periods of low activity. Additionally, serverless computing eliminates the need for infrastructure provisioning and management, reducing administrative overhead and allowing developers to focus on core application logic.

Furthermore, the pay-per-execution pricing model of serverless services results in cost-effectiveness, as developers only pay for the resources consumed by their applications, rather than for idle capacity. Moreover, serverless architecture simplifies operational management by abstracting away infrastructure concerns, such as server maintenance and security patching, thereby enabling teams to iterate and deploy new features more rapidly. Overall, leveraging serverless services for web application deployment offers developers a streamlined and efficient approach, enhancing scalability, reducing costs, and simplifying operational management.

**RESOURCE DIAGRAM**

****

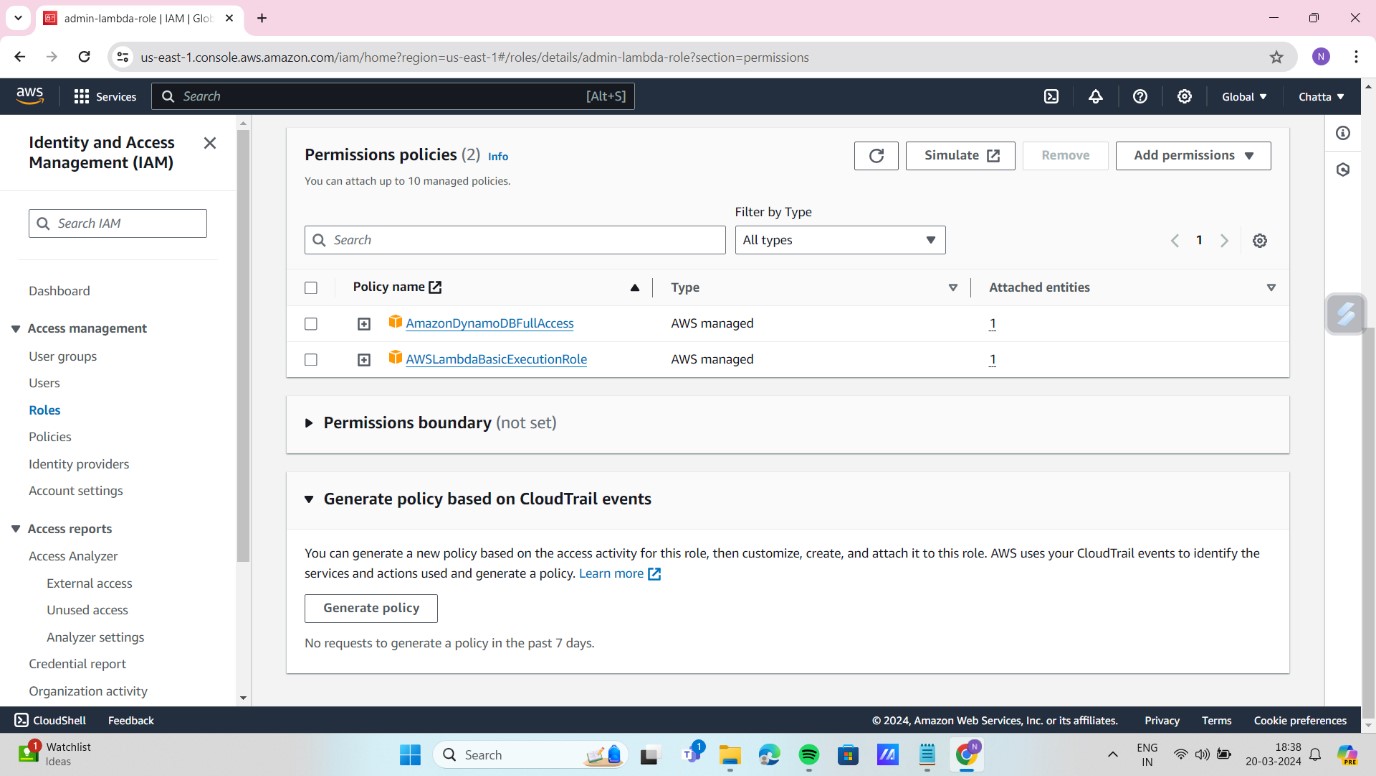
The image you sent shows a resource diagram of an AWS account connected to Amazon DynamoDB. Amazon DynamoDB is a NoSQL database service offered by Amazon Web Services (AWS). It is a fast and scalable database that can handle a high volume of data requests. The resource diagram illustrates how these two resources can work together.

**RESOURCES LIST:**

* IAM Role
* IAM Policies
* Lambda Function
* DynamoDB Table
* API Gateway
* REST API

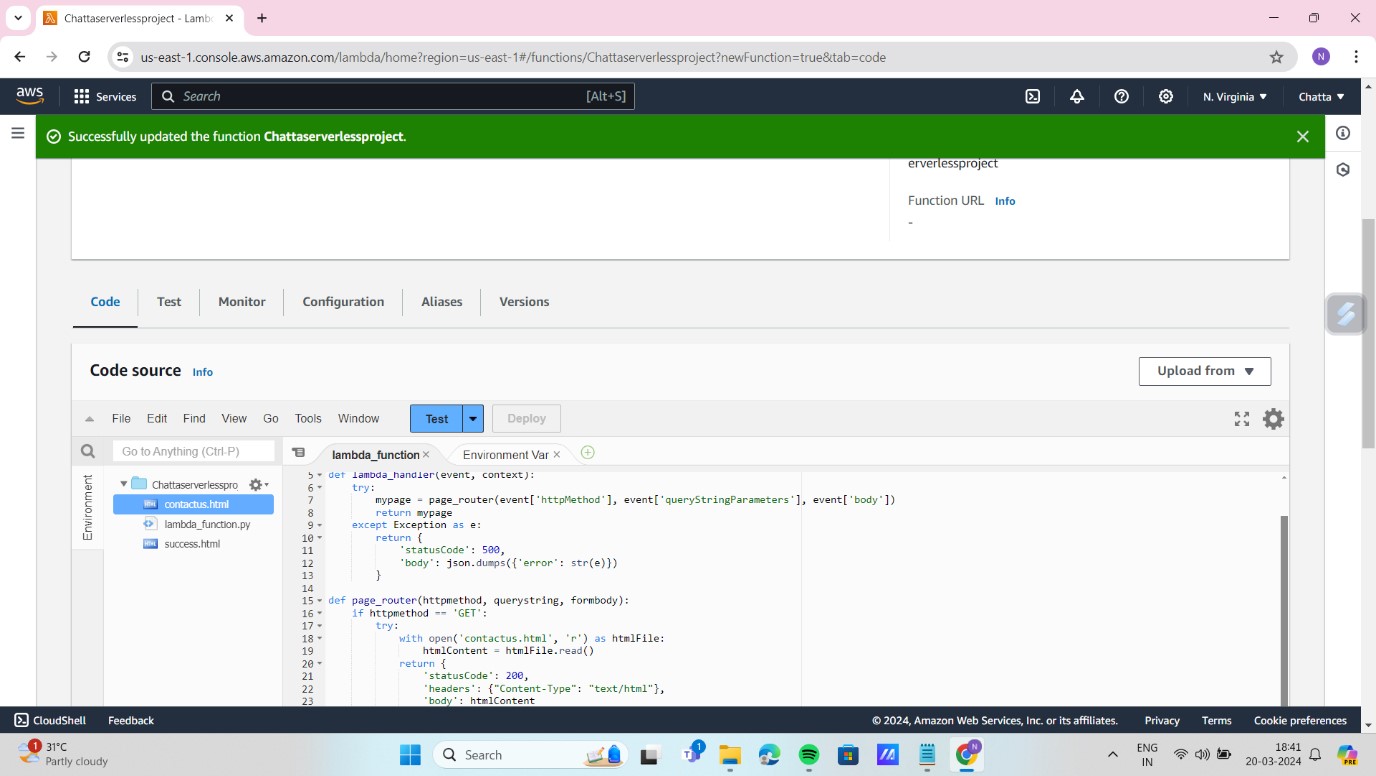
# PROCEDURE

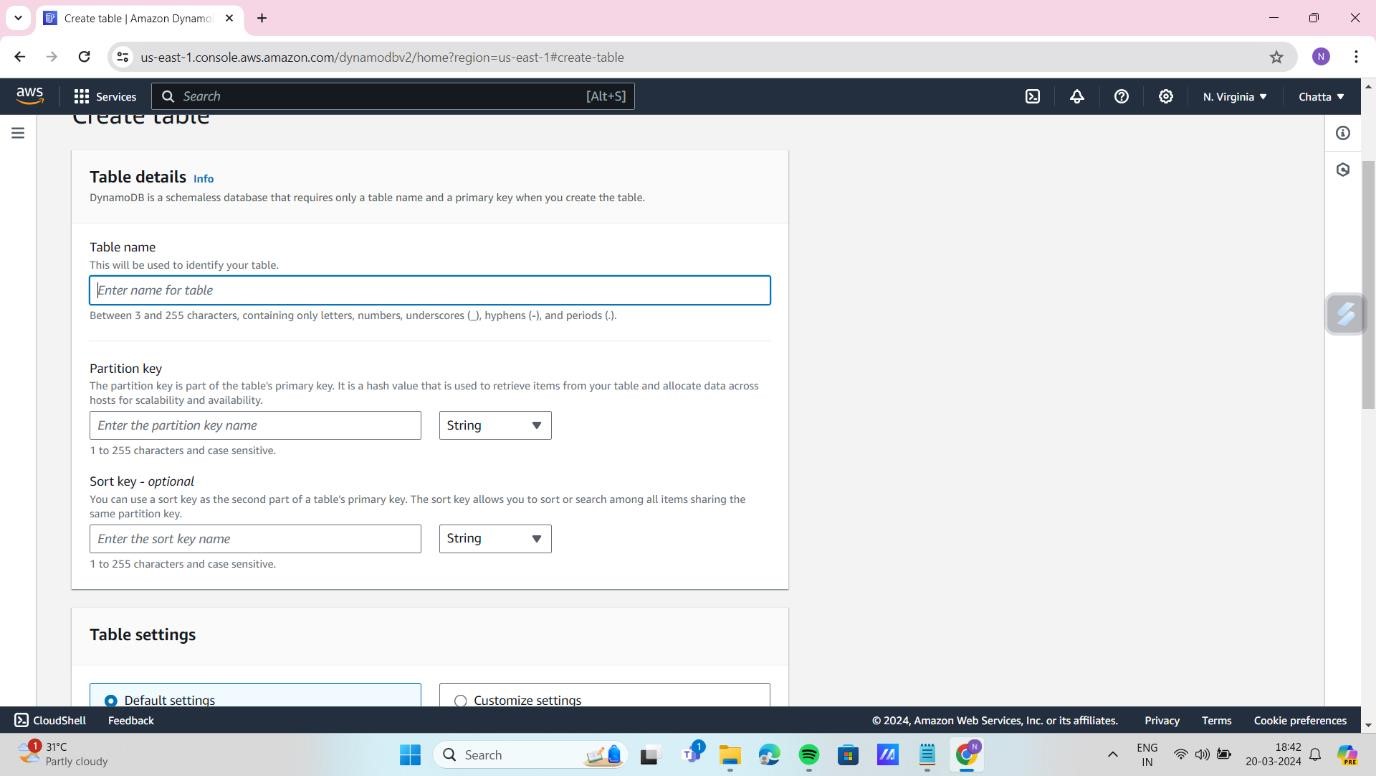
1.Create one iam role and attach two policies that is mentioned in below picture.



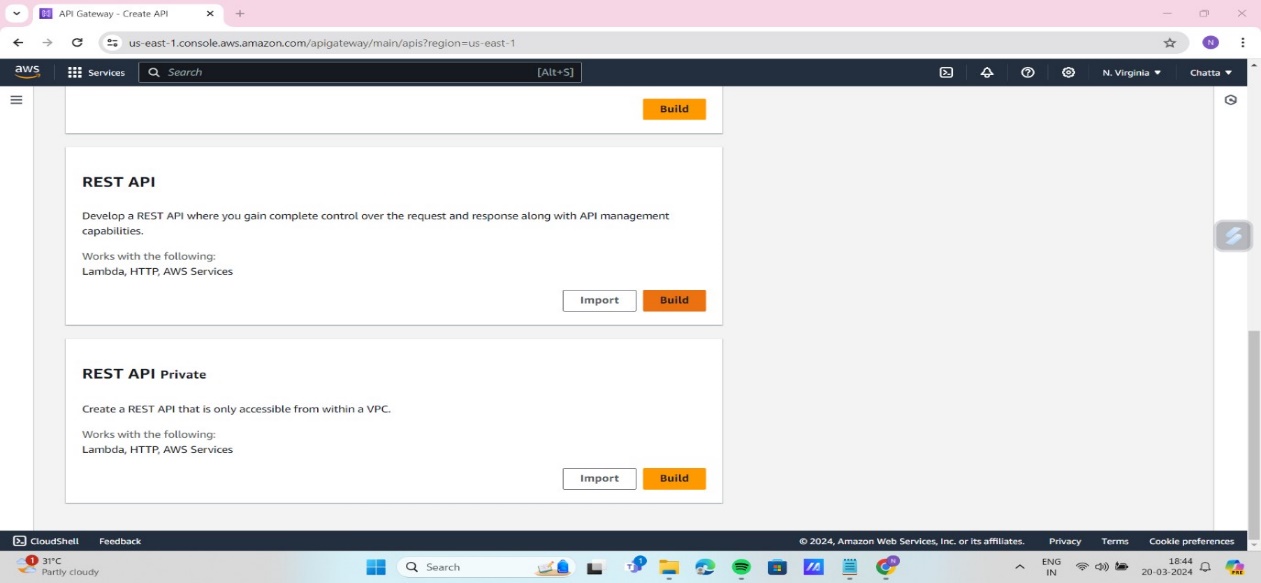
2.Create one lambda function , select runtime as python and role is existing role that is created before.

3. Upload the code in the ZIP file that is also provided in my repository.

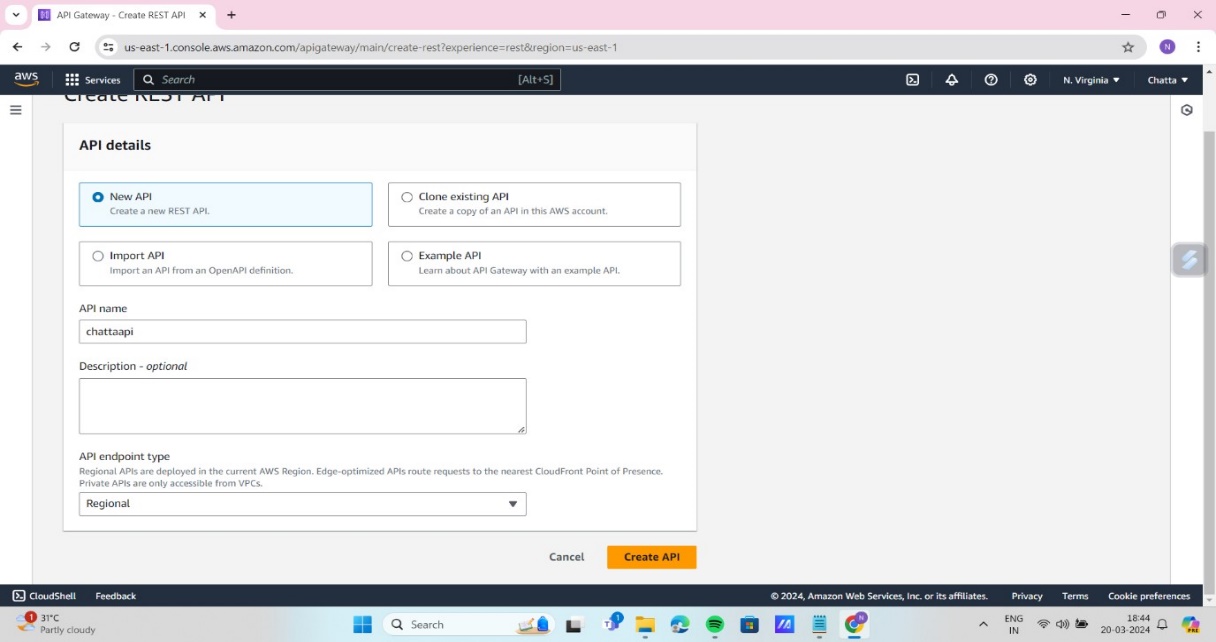


4.Search the service DynamoDB and create a table, enter the table name as that is present in the code, enter the partition key as email (String) table setting as default keep it,t hen click on create table.

Note: Give the table name as it is that is present in the lambda function code “Insert into table\_name values”.

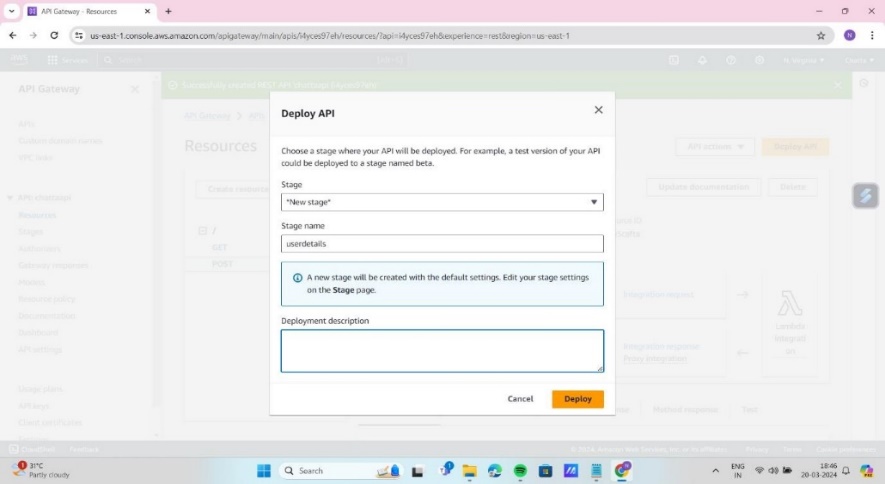
5.Search the service Api gateway click on build “rest api”.

6. In api details select new api and give your api name and api end point type is regional, then click on create.



7. After creating api click on it and click on create method for get and post method type follow this picture steps as it is. This is for get method same like create for post method also.

8. Then click on deploy api select stage as new stage give stage name your choice then deploy the api.



9. After deploying the api you will get the invoke url copy that and paste and serach in new tab you will get the application index page fill the form and click on submit you will get the success message and details you entered that will store in DynamoDB table.

**OUTPUT:**

|  |  |
| --- | --- |
|  | After entering the details to see the stored user details in dynamoDB table open table that is created for this project and click on explore items there you can see the user entered first, last name and email, description. |

# CONCLUSION

In conclusion, the deployment of web applications using serverless architecture on cloud platforms like AWS offers a transformative approach to modern software development. By abstracting away infrastructure management, serverless computing enables enhanced scalability, cost efficiency, and simplified operational management. Leveraging event-driven architecture and microservices principles, developers can architect resilient and scalable web applications that dynamically adapt to fluctuating workloads. Despite challenges such as cold start latency and vendor lock-in concerns, strategies exist to mitigate these issues. Real-world case studies and best practices underscore the applicability of serverless computing across diverse use cases, fostering innovation, agility, and cost efficiency in the modern software development landscape.

# REFERENCES

# 

* https://aws.amazon.com/websites
* <https://aws.amazon.com/getting-started/projects/host-static-website/faq>
* <https://en.wikipedia.org/wiki/Amazon_Web_Services>
* [https://aws.amazon.com/free/webapps](https://aws.amazon.com/free/webapps/)
* https://www.contino.io/insights/whos-using-aws