

# HOSTING A WEB-APPLICATION ON AWS CLOUD PLATFORM

Dr T. Sathish Kumar<sup>1</sup>, S.V. Manasa<sup>2</sup>, R. Sahithya<sup>3</sup>, V. Mahalakshmi Sai<sup>4</sup>, M. Teja Rani<sup>5</sup>

<sup>1</sup>Associate Professor, Computer Science Engineering, Hyderabad Institute of Technology and Management.

<sup>2</sup>Student, Computer Science Engineering, Hyderabad Institute of Technology and Management.

<sup>3</sup>Student, Computer Science Engineering, Hyderabad Institute of Technology and Management.

<sup>4</sup>Student, Computer Science Engineering, Hyderabad Institute of Technology and Management.

<sup>5</sup>Student, Computer Science Engineering, Hyderabad Institute of Technology and Management.

**Abstract**— By offering access to apps locally or online without the need for additional hardware or software, cloud computing has completely changed modern computing. With this paradigm change, web-based solutions replace traditional computer programs on the internet, making it a platform for a variety of applications. Reputable businesses like Google and Microsoft provide a wide range of services, all based on reliable cloud infrastructure, from social networking to photo sharing. This essay examines the benefits of the mutually beneficial interaction between cloud computing and web applications. This abstract seeks to increase accessibility to a wider audience by demystifying complicated ideas.

**Keywords**—Elastic Beanstalk, S3 Bucket, Cloud Computing, Web Application

## INTRODUCTION

The way we manage IT infrastructure has entirely altered as a result of cloud computing. Organizations used to have to manage every aspect of server maintenance and security on their own. However, things are now a lot simpler thanks to cloud computing systems like Amazon Web Services (AWS). By handling performance and security concerns, AWS enables businesses to only pay for the resources they actually utilize. Virtualization, a technique that produces virtual representations of resources like servers and storage, makes all of this feasible. We can use virtualization to increase efficiency and save costs by making better use of our hardware.

AWS offers a multitude of services to meet different needs. For example, if you utilize Software as a Service (SaaS), you can use apps hosted by AWS without having to install them yourself. Thanks to Platform as a Service (PaaS), AWS provides tools for developing and running apps without requiring concern for the underlying infrastructure. Furthermore, AWS offers Infrastructure as a Service (IaaS), [1] which allows you to build and execute your applications using all the hardware and infrastructure components you need. Among the biggest and most comprehensive cloud service providers is AWS



Figure 1: AWS CLOUD PLATFORM

It offers a wide range of services, including as networking, processing, storage, and security. Additionally, AWS provides managed services like Amazon RDS, which is used for databases, and AWS Lambda, which enables code to run without server administration.

What sets AWS apart is its global design that spans many availability zones and regions. This ensures that your applications are always dependable and available, no matter where your users are located. With AWS's vast ecosystem of tools and services, you can develop quickly and easily while expanding your business. Whether you're hosting a website, managing data analytics, or developing artificial intelligence models, AWS has everything you need to succeed in the cloud.

Virtualization, which generates virtualized representations of resources like servers and storage, makes this feasible. We may use virtualization to increase efficiency and save expenses by making greater use of our hardware.

#### Benefits of Deploying with AWS Services

##### AWS:

- Agile deployments
- Flexible options
- Scalable capacity
- Automation capabilities

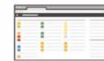


Figure 2: Benefits of Deploying

## LITERATURE SURVEY

In terms of virtual machine migration, Nagamani H. Shahapure and P. Jayarekha[2] present a novel paradigm based on the length of time required for the service to be moved. They make use of data on network traffic as well as the locations of each datacenter. In order to improve performance, they took the clients' requests into consideration and forwarded them to the closest datacenters. They suggested an algorithm for time-sensitive applications.

According to Naga Malleswari TYJ and Vadivu G et al.[3], virtual machines running on overloaded servers can be moved to underloaded servers, and some virtual machines running on unwanted underloaded servers can be moved to those servers that are not overloaded. All of these virtual machine moves will occur while the server is online, so they suggested reducing the amount of time needed for the live virtual machine migration, which will improve application performance and decrease downtime.

The research of all live virtual machine migration scenarios is presented by Anita Choudhary, Mahesh Chandra Govil, et al.[4], who also conduct a thorough investigation of the previously suggested procedures. They also discussed numerous security risks that arise during live virtual machine migration and demonstrated various duplication strategies for VM migration. They give professionals all the background and details about the virtual machine migration, mostly so that they may determine which of the suggested approaches is the best and refine the approaches even further for even greater gains.

In addition to discussing the models and architecture of cloud computing and the services it offers, such as networking, storage, and compute, Onuora, A. C., Emereonye et al.[5] also explain the features of cloud computing, such as availability, scalability, resilience, and migration. For users who are accustomed to using the cloud and for those who are considering making the transfer to cloud computing technology, they took into account both the security and resilience of the cloud. They go over how dangers, both known and new, can be prevented or detected by the cloud and dealt with in public or enterprise clouds.

## METHODOLOGY

A simple and affordable approach to publish your static website online is to host it as a web application on AWS S3. Amazon Web Services offers a robust and highly scalable storage solution called AWS S3 (Simple Storage Service).[6] You may serve HTML, CSS, JavaScript, and other static files straight from an S3 bucket by taking advantage of the static website hosting features offered by S3. For portfolio pages, personal websites, and other static web apps, this configuration is perfect.

How to Use AWS S3 to Host a Web Application:

1. Make an account on Aws platform.

Register: Go to the AWS website and create an account. Both your payment information and personal details must be submitted. For tiny applications, AWS provides a free tier with restricted S3 storage.

2. Open the AWS Management Console and log in.

Get to the Console: Proceed to the AWS Management Console and sign in with the credentials you just created. The web-based interface for managing AWS services is called the console.

3. Install the AWS CLI (Required).

Command-Line Tool:

You can use terminal commands to communicate with AWS services through the AWS Command Line Interface (CLI). While not required for a basic setup, this is helpful for automation.

Installation: If you decide to use the AWS CLI installation page, follow the installation instructions found there.

4. Get the files for your website ready.

Create Your Website: To build the HTML, CSS, and JavaScript files for your website, use a code editor such as Visual Studio Code or Sublime Text.

Version Control: Use a version control system, such as Git, to organize your code. If you're working in a group, this promotes cooperation and keeps your code organized.

5. Make an S3 container.

Go to S3 by searching for "S3" in the AWS Management Console and choosing the S3 service.

Make a bucket: Once you have selected the location most convenient for your users, click "Create bucket" and give your bucket a unique name (my-web-app-bucket, for example). Click "Create bucket" after leaving the default parameters.

6. Create a Bucket to Host a Website:

To enable website hosting, open your bucket, select "Properties" from the menu, then "Edit" next to "Static website hosting." Choose "Host a static website" after selecting "Enable".

Name the Documents: As the index document, enter your primary HTML file (index.html), and optionally, an error document (error.html). Save the modifications.

**7. Upload the files for your website:**

**File Upload:** Open your bucket's "Objects" tab. Drag and drop your JavaScript, HTML, and CSS files into the upload field, then click "Upload" to begin the upload process.

**8. Publicize Files:**

**Set Permissions:** Click "Actions" after selecting the uploaded files, then pick "Make public" to make your files available to the public.

**Bucket Policy:** As an alternative, you can grant read access to the public using a bucket policy. Click "Bucket Policy" under the "Permissions" tab, then add the following JSON policy:

**NOTE:**

Enter the name of your actual bucket in place of my-web-app-bucket. To use the policy, save it.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-web-app-bucket/*"
    }
  ]
}
```

**9. Get on Your Page:**

**Website URL:** Locate your website's URL (for example, <http://my-web-app-bucket.s3-website-us-west-1.amazonaws.com>) under the "Properties" tab's "Static website hosting" section.

**Visit Your Website:** To view your live website, open this URL in a web browser.

**10. (Details optional) Boost Efficiency with CDN:**

**Amazon CloudFront:** Use Amazon CloudFront, a Content Delivery Network (CDN), for improved performance.

**Configure CloudFront:** Open the AWS Management Console, navigate to CloudFront, create a new distribution, and select your S3 bucket as the origin. This will provide quicker load times by caching your content closer to users.

**IMPLEMENTATION**

Logging into the AWS Management Console and accessing the S3 service is the first step towards hosting a web application on AWS S3. After choosing the location nearest to your users, click "Create bucket" and give it a distinctive name. Open your bucket and select the "Properties" tab after it has been created. Turn on "Static website hosting" and designate the index document as your primary HTML file, which is typically index.html. Another option is to set an error document, such as error.html.

Next, upload the HTML, CSS, and JavaScript files for your website. Click "Upload" under the "Objects" tab in your bucket to accomplish this. Click "Upload" after dragging and dropping your

files into the upload field.

Verify that the files are available to the public after uploading. Click "Actions" and pick "Make public" after selecting all the files. As an alternative, you can provide public read access by setting a bucket policy. Select "Bucket Policy" under the "Permissions" tab, then add a JSON policy to your files that allows the public to read them.

The website endpoint URL for your bucket may be found under "Static website hosting" when you return to the "Properties" tab. This is the URL at which your website is now active and available to all users. Using AWS's strong infrastructure for scalability and dependability, this easy setup guarantees that your web application is accessible online.

## RESULTS & DISCUSSIONS

Installing a web application on Amazon S3 is a great option for anyone looking for an easy, affordable, and safe hosting solution. S3 adjusts automatically to different traffic volumes, so you can keep your app accessible and responsive without having to do anything manually. S3's versioning capability is very useful since it makes it simple to create backups and restore earlier iterations, protecting against unintentional data loss. With AWS's extensive controls, which include options for encryption and access management, security is strong. Whether using the Command Line Interface (CLI) or the AWS Management Console, uploading files to S3 is simple and makes deployment possible even for those with less cloud knowledge. Furthermore, by lowering latency and speeding up load times for consumers everywhere, connecting S3 with custom domains and Amazon CloudFront, a global content delivery network (CDN), improves performance. Because it is easy to use, scalable, secure, and performant, S3 is a practical and efficient choice for hosting small to medium-sized websites.

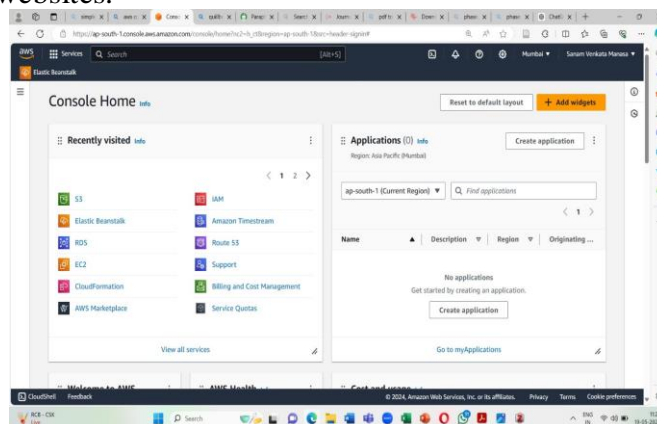


Figure 3: AWS Console

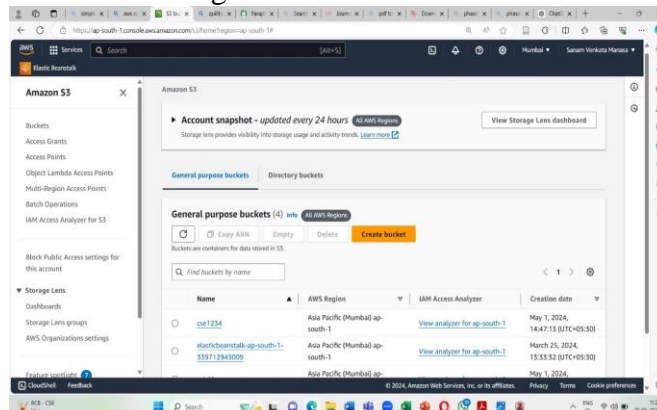


Figure 4: AWS S3 Bucket



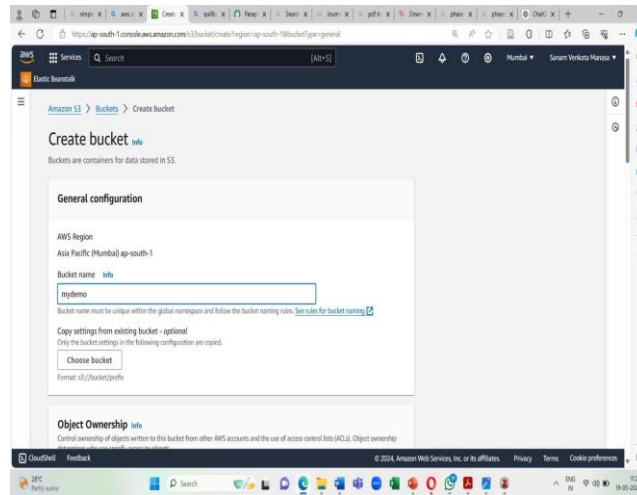


Figure 5: Create A Bucket

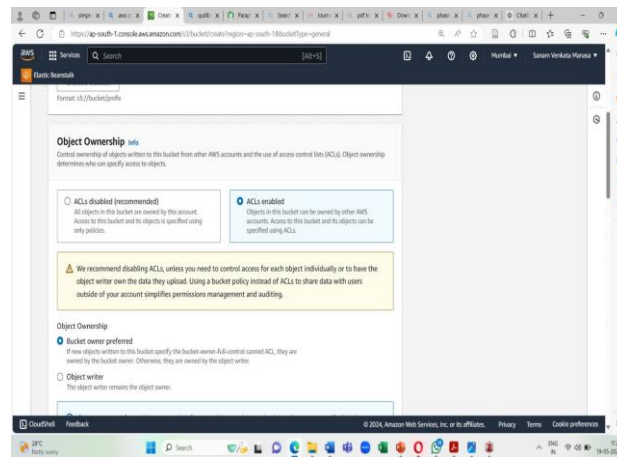


Figure 6: Enable ACLs

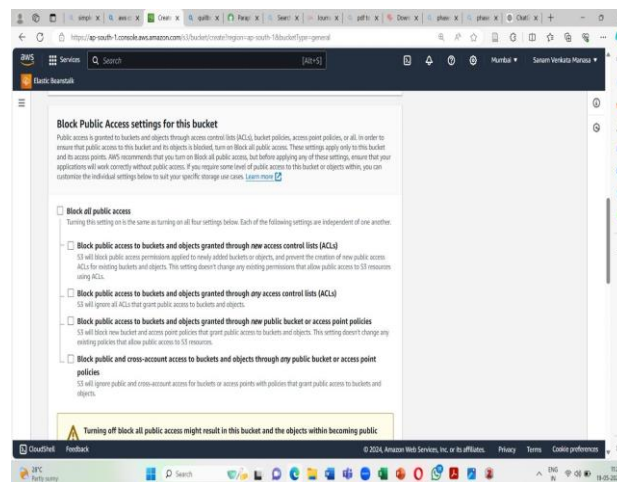


Figure 7: Disable Public Access

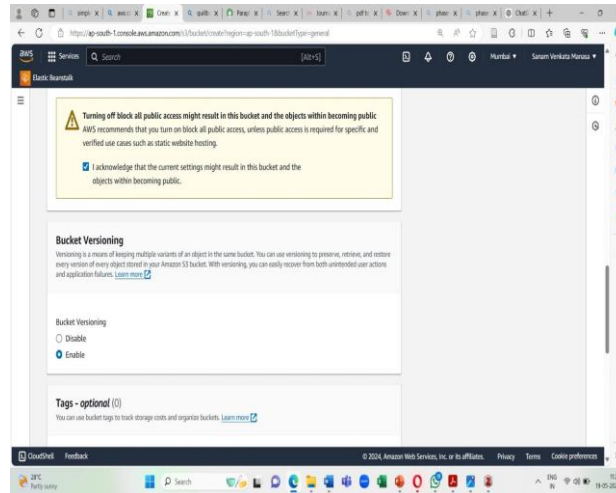


Figure 8: Enable Bucket Versioning

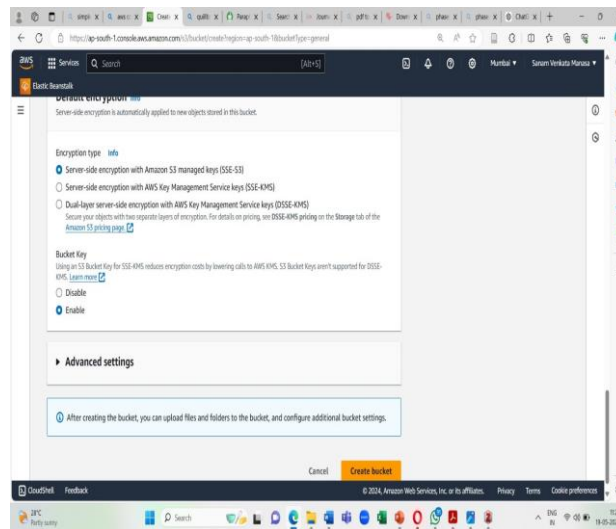


Figure 9: Click Create Bucket

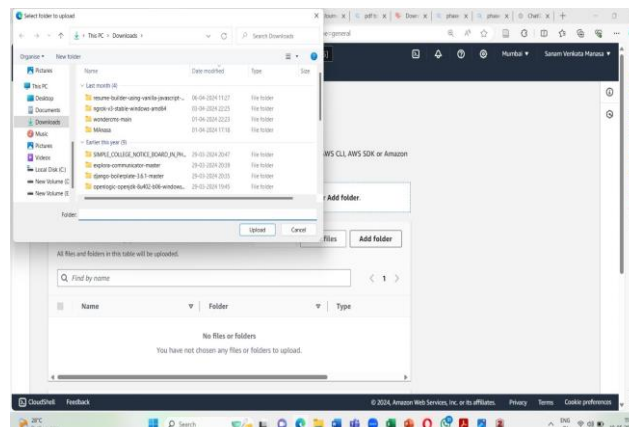


Figure 10: Add Files To Bucket

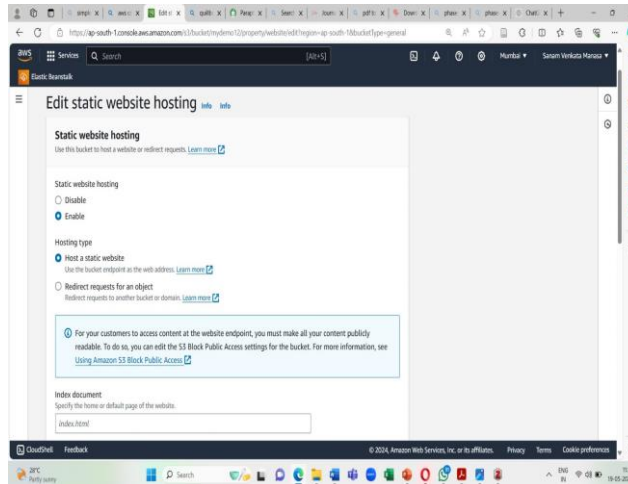


Figure 11: Edit Static Website Hosting

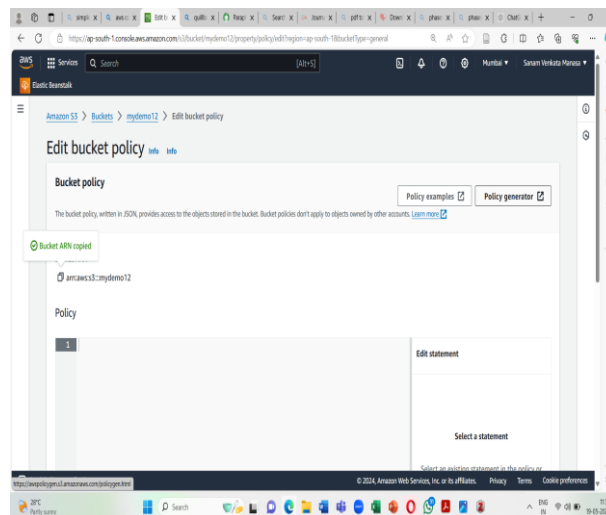


Figure 12: Edit Bucket Policy

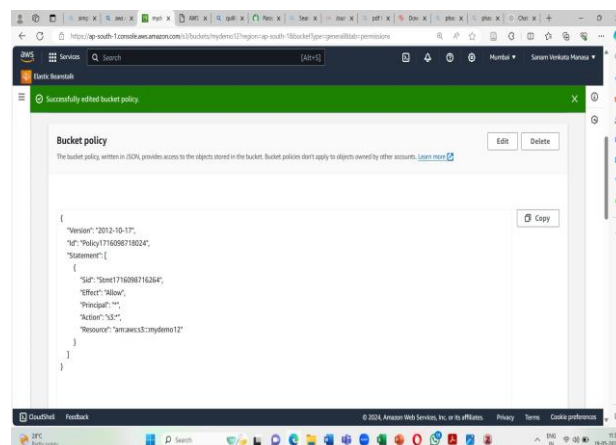


Figure 13: Generate Policy



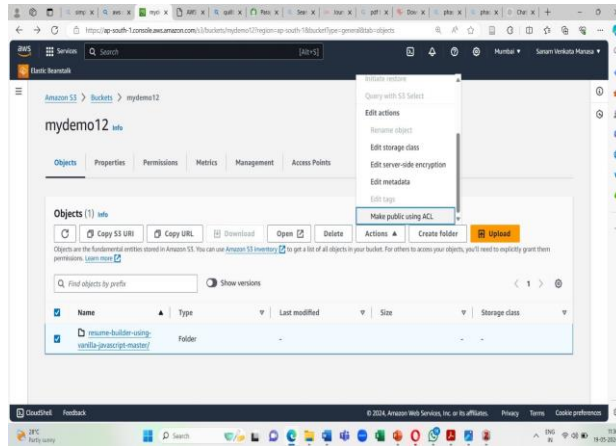


Figure 14: Make Public Using ACL

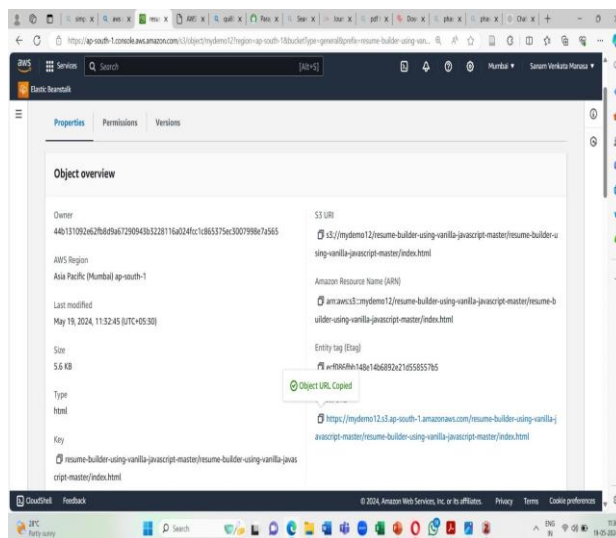


Figure 15: Object Overview

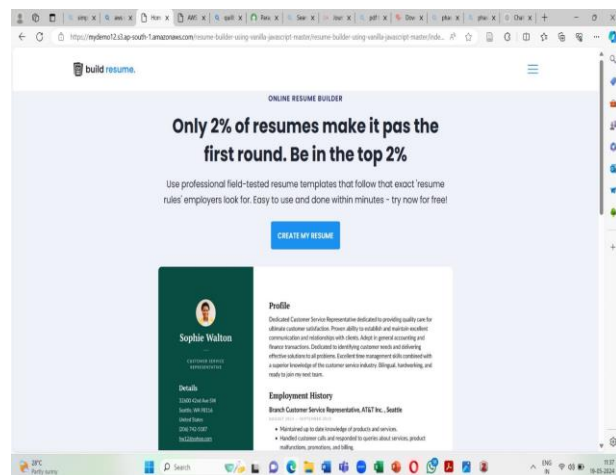


Figure 16: Deployment Successful

## CONCLUSION

In conclusion, hosting a web application on Amazon S3 is similar to renting a remarkably dependable and adaptable online space. We've shown that it's simple to use and reliable for hosting websites when configured appropriately. We can't wait to leverage AWS Lambda to add more capabilities to our website in the future, such as the ability to have it perform actions when people



interact with it. It's similar to giving our website superpowers so it can accomplish amazing things without a standard server.

### References

- [1] A. Sarkar and A. Shah, *Learning AWS: Design, build, and deploy responsive applications using AWS Cloud components*. Packt Publishing Ltd, 2018. Accessed: Feb. 18, 2024. [Online]. Available: <https://books.google.com/books?hl=en&lr=&id=-GFLDwAAQBAJ&oi=fnd&pg=PP1&dq=deployment+of+web+application+in+aws+in+cloud&ots=39Ht8Cmw9z&sig=6EjI1q8QPrdu5MpuaJDQ2wHxQiE>
- [2] N. H. Shahapure and P. Jayarekha, 'Load balancing in cloud computing: a survey', *Int. J. Adv. Eng. Technol.*, vol. 6, no. 6, pp. 2657–2664, 2014.
- [3] T. Y. J. Naga Malleswari, G. Vadivu, and D. Malathi, 'Live Virtual Machine Migration Techniques—A Technical Survey', in *Intelligent Computing, Communication and Devices*, vol. 308, L. C. Jain, S. Patnaik, and N. Ichalkaranje, Eds., in *Advances in Intelligent Systems and Computing*, vol. 308. , New Delhi: Springer India, 2015, pp. 303–319. doi: 10.1007/978-81-322-2012-1\_32.
- [4] A. Choudhary, M. C. Govil, G. Singh, L. K. Awasthi, E. S. Pilli, and D. Kapil, 'A critical survey of live virtual machine migration techniques', *J. Cloud Comput.*, vol. 6, no. 1, p. 23, Dec. 2017, doi: 10.1186/s13677-017-0092-1.
- [5] A. C. Onuora, G. I. Emereonye, R. I. Egwu-Ewah, and D. I. Nnaji, 'Cloud Security and Resilience: Principles and best Practices', *AIPFU J. Sch. Sci. AJSS*, vol. 1, no. 1, pp. 5–11, 2017.
- [6] A. Gupta, P. Goswami, N. Chaudhary, and R. Bansal, 'Deploying an application using google cloud platform', in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, IEEE, 2020, pp. 236–239. Accessed: Oct. 17, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9074911/>