A PRELIMINARY REPORT ON

INTERNSHIP IN

CLOUD COMPUTING

THIRD YEAR ENGINEERING (COMPUTER ENGINEERING)
SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,
PUNE

SUBMITTED BY

CHAUDHARI TANISHA PRAKASH

PRN: 72251699B

UNDER GUIDANCE OF
MR. R. S. JAGALE

DEPARTMENT OF COMPUTER ENGINEERING
GES'S R. H. SAPAT COE MANAGEMENT STUDIES AND
RESEARCH PRIN T A KULKARNI VIDYANAGAR, NASHIK - 422005
SAVITRIBAI PHULE PUNE
UNIVERSITY

ACADEMIC YEAR- 2024 -2025

Conducts of the Internship Training

Place of Internship:

Name: Arrow Technologies and Solutions

Address: 1ST Floor D J House ,Bhosala Circle, College Road, Nashik-422005

Supervisor: Mr. Akash Patil

Department Supervisor:
Mrs. Prajakta Kshirsagar (Co- Founder Arrow Technologies and Solutions)

CERTIFICATE

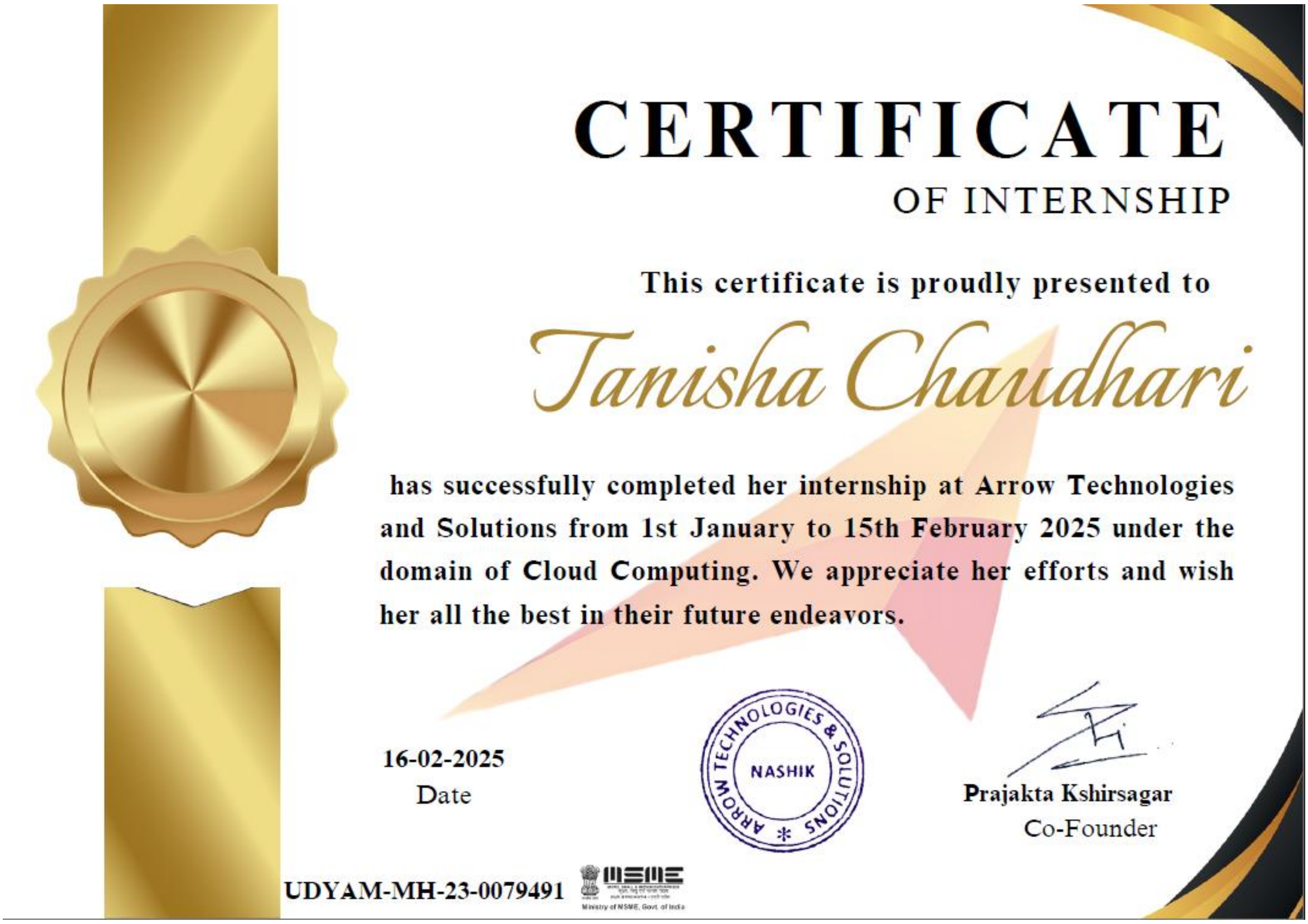
This is to certify that the Internship report entitles **Cloud Computing** Submitted by **Chaudhari Tanisha Prakash PRN: 72251699B** is a bonafide student of this institute and the work has been carried out by her under the supervision of **Mr. R. S. Jagale** and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University.

(Mr. R. S. Jagale)	**(Dr. D. V. Patil)**	**(Dr. S. S. Sane)**
Guide	Head,	Principal,
Department Of	Department of	Gokhale Edu. Society's
Computer Engineering	Computer Engineering	R. H. Sapat COE MSR, Nashik–5

Place: Nashik

Date:

Certificate of Internship :



CERTIFICATE

OF INTERNSHIP

This certificate is proudly presented to

Tanisha Chaudhari

has successfully completed her internship at Arrow Technologies and Solutions from 1st January to 15th February 2025 under the domain of Cloud Computing. We appreciate her efforts and wish her all the best in their future endeavors.

16-02-2025
Date

ARROW TECHNOLOGIES & SOLUTIONS
NASHIK

Prajakta Kshirsagar
Co-Founder

UDYAM-MH-23-0079491

MSME
Ministry of MSME, Govt. of India

ACKNOWLEDGEMENT

The successful completion of this internship report would not have been possible without the support and assistance of many individuals and organizations. I feel immensely blessed to have gotten this during the course of my internship program. I would like to take this opportunity to offer my sincere gratitude to everyone for providing their invaluable guidance, comments and
Suggestions throughout the process of my internship.

First I would like to thank Arrow Technologies and Solutions, for giving me the opportunity to do an internship within the organization.

I would like to specially thank Mr. R. S. Jagale for timely checking my progress, constantly motivating me to work harder and being a very supportive mentor.

I am highly indebted to the head of the computer engineering department Dr. D.V. Patil and Principal Dr. S. S. Sane, for the facilities provided to accomplish this internship.

I am extremely grateful to my department staff members who helped me in successful completion of this internship.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

CHAUDHARI TANISHA PRAKASH

Abstract

Cloud web hosting provides an efficient, scalable, and flexible way to deploy and manage websites and applications. Amazon Web Services (AWS) offers a wide range of cloud services, with Elastic Compute Cloud (EC2) and Simple Storage Service (S3) being two of the most commonly used for web hosting. EC2 provides resizable compute capacity in the cloud, allowing users to launch virtual servers (instances) that can be easily scaled based on traffic demands. This elasticity ensures optimal performance and high availability for web applications. EC2 offers a variety of instance types tailored to different workloads, enabling users to choose the appropriate resources for their specific needs. In addition, EC2's auto-scaling features allow for the automatic adjustment of resources, providing seamless scaling in response to changes in user traffic.

S3, on the other hand, is a highly scalable object storage service that enables the storage and retrieval of data, such as images, videos, documents, and backups. It is designed for static content and is often used in conjunction with EC2 to handle the storage and delivery of assets for web applications. S3 provides high durability and availability by replicating data across multiple facilities, ensuring data integrity and security. With its cost-effective pricing models and user-friendly interface, S3 serves as a key solution for managing static assets in cloud-based web hosting environments.

Together, EC2 and S3 form the backbone of cloud web hosting solutions, where EC2 handles dynamic content and application logic, while S3 efficiently manages static content storage and delivery. This integration enables businesses to build and host scalable, reliable, and cost-effective web application

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1st WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	01/01/25	Monday	Introduction to Cloud
	02/01/25	Tuesday	Services Of Cloud
	03/01/25	Wednesday	Introduction to Networking
	04/01/25	Thursday	Windows Vs Linux, Design of Network
	05/01/25	Friday	OSI Model, IP Addressing
	06/01/25	Saturday	Selection of Connections, Use of Command prompt on cisco Packet tracer Tool

2nd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	08/01/25	Monday	Session on Network Establishment
	09/01/25	Tuesday	Task To establish LAN network
	10/01/25	Wednesday	Session on WAN Interface Card, Creating switch to switch, router to router network
	11/01/25	Thursday	Hands on practice on router commands using tools
	12/01/25	Friday	Static Routing and Dynamic routing, RIP Version 1 and Version 2
	13/01/25	Saturday	Network Establishment using different protocols
	14/01/25	Sunday	Installation Of VMWARE Workstation

3rd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	15/01/25	Monday	Creating Virtual Machines, Use of Basic Commands of VMWARE workstation

	16/01/25	Tuesday	Introduction to AWS And AWS Account Creation
	17/01/25	Wednesday	Session on Aws Console, Services, Pricing
	18/01/25	Thursday	IAAS, PAAS, SAAS
	19/01/25	Friday	Introduction to EC2 Service
	20/01/25	Saturday	Using EC2 creating Instances, connection of Instances

4th WEEK	**DATE**	**DAY**	**NAME OF THE TOPIC/MODULE COMPLETED**
	22/01/25	Monday	Key Value Pair generation for connecting Instances, Terminating instances.
	23/01/25	Tuesday	Introduction to S3 Service
		Wednesday	Creating Bucket in S3
	25/01/25	Thursday	Hosting Using S3
	26/01/25	Friday	Generating Policy to host Public Platform

5th WEEK	**DATE**	**DAY**	**NAME OF THE TOPIC/MODULE COMPLETED**
	4/02/25	Monday	Cloud web hosting with S3
	15/02/25	Tuesday	Presentation of Internship

Contents

Chapter 1: Introduction

1.1 Introduction

Cloud web hosting refers to the hosting of websites or applications on cloud computing platforms rather than traditional physical servers. In this model, hosting resources such as storage, computing power, and networking are provided over the internet from virtualized data centers. Cloud hosting offers greater flexibility, scalability, and cost-effectiveness compared to traditional hosting methods, which often rely on dedicated physical servers. As businesses and individuals increasingly demand highly available, reliable, and scalable web applications, cloud web hosting has emerged as a dominant solution. One of the key players in the cloud hosting space is Amazon Web Services (AWS), a comprehensive and widely adopted cloud platform offering a range of services to meet the needs of modern web hosting.

1.2 Objective

The objective of this project is to explore and implement cloud web hosting using AWS services, specifically EC2 and S3. The goal is to leverage these services to build a scalable, cost-efficient, and high-performance hosting environment for a website or web application. By configuring EC2 for dynamic content processing and S3 for static content storage, the project aims to optimize resource management, enhance website performance, and ensure high availability. Additionally, the project seeks to highlight the practical use of cloud computing technologies in modern web hosting and demonstrate the advantages of cloud over traditional hosting solutions.

1.3 Overview of Cloud Web Hosting

Cloud web hosting is the practice of hosting websites or applications on cloud platforms that distribute data and resources across multiple virtual servers. This hosting method is increasingly popular due to its scalability, cost-effectiveness, and flexibility. Unlike traditional hosting, which uses physical servers in fixed data centers, cloud hosting allows users to scale resources based on demand, only paying for the resources they use. Cloud hosting has revolutionized the web hosting industry by offering reliability, faster deployment, and simplified management for businesses and developers.

1.4 AWS (Amazon Web Services) Overview

Amazon Web Services (AWS) is a leading cloud service provider that offers a broad range of infrastructure services including compute, storage, and networking. Among AWS's most notable services for web hosting are EC2 (Elastic Compute Cloud) and S3 (Simple Storage Service). EC2 allows users to run virtual machines (instances) to host web applications, while S3 provides scalable storage for static assets like images, videos, and web pages. AWS’s global infrastructure ensures that web hosting can be scaled efficiently across regions, providing high availability and performance for users worldwide.

1.5 Services Used in the Project

1)EC2 (Elastic Compute Cloud): EC2 is a scalable compute service that provides virtual servers (instances) for running applications and services. It offers flexibility in choosing the computing power, memory, and storage size depending on the needs of the application.

2)S3 (Simple Storage Service): S3 is an object storage service designed to store and retrieve large amounts of data. It is widely used for hosting static content such as HTML files, CSS, images, and JavaScript, due to its scalability, durability, and ease of access.

Chapter 2 System Analysis and Requirement

The architecture consists of an EC2 instance for hosting dynamic web pages and a linked S3 bucket for static files. The EC2 instance processes requests for dynamic content, while static assets like images, CSS, and JavaScript are served from the S3 bucket. This integration ensures a balanced load and optimized performance.

2.1 System Overview

The system is designed to host a website or web application that incorporates both dynamic and static content. AWS services such as Amazon EC2 and Amazon S3 are utilized to ensure efficient resource management, scalability, and high availability.

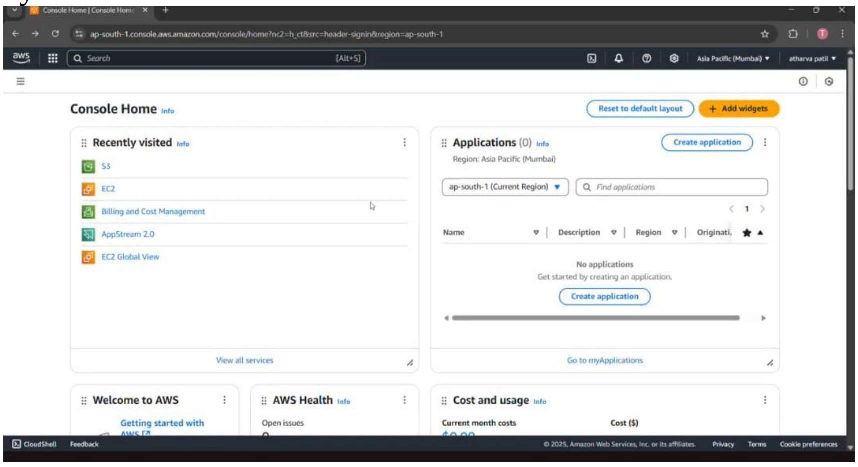
Amazon EC2 (Elastic Compute Cloud) provides the necessary computing resources for hosting dynamic content. EC2 instances act as virtual servers where the web application runs, handling requests, processing server-side logic, and generating responses.

Amazon S3 (Simple Storage Service) is leveraged for static content hosting, such as images, CSS files, JavaScript files, and other assets that don't require processing on the server-side. S3 offers scalable, highly durable, and low-cost storage, making it ideal for hosting static files.
This setup benefits from the scalability of AWS to handle varying levels of traffic. The EC2 instances can be scaled up or down based on demand, and static content stored in S3 is distributed globally with low latency

2.2 System Architecture

1)EC2 Instances: These virtual machines host the dynamic aspects of the website or application. When users visit the site, the EC2 instance serves dynamic content, such as HTML generated from server-side code (PHP, Node.js, Python, etc.), database queries, or API responses.

2)Amazon S3: S3 is used for storing static files such as images, stylesheets, JavaScript files, and other resources that don't require processing. S3 serves these assets via HTTP/S when requested by the web browser.



No applications

Get started by creating an application.

Figure 1 System Architecture

2.3 Requirements

To set up a web hosting system using EC2 and S3, the following requirements must be met:

2.3.1. Hardware Requirements:

- **EC2 Instance:** The compute resources needed depend on the expected traffic and workload. EC2 instances come in a variety of types and sizes, from t2.micro (for light workloads) to c5.18xlarge (for compute-heavy applications). The choice of instance type (e.g., T3, C5, M5) depends on:
  - CPU requirements
  - Memory (RAM) needs
  - Storage capacity (EC2 can use both local storage and EBS volumes)
- **S3 Storage:** S3 storage requirements depend on the volume of static content. Since S3 is scalable, there is no strict limit, but cost will be tied to the amount of data stored and the number of requests made.

2.3.2. Software Requirements:

- **Web Servers:** EC2 instances will typically run web servers such as Apache or Nginx to handle HTTP requests for dynamic content. These servers can be configured with additional modules or handlers depending on the technology stack (e.g., PHP for Apache).
- **Application Software:** The application software running on EC2 depends on the website or application’s technology stack. For instance:
  - PHP, Node.js, Python (Django/Flask) for backend applications.
  - MySQL, PostgreSQL, or MongoDB for relational or NoSQL databases.
  - If required, AWS Lambda functions can be used to handle specific serverless workloads.
- **Operating System:** The EC2 instance can run on various operating systems, with common choices being Ubuntu (for open-source technologies) or Windows Server (for Microsoft technologies).

2.3.3. AWS Services Configuration:

- **EC2 Instance Configuration**: EC2 instances must be configured with the correct instance type (e.g., t2.small, c5.xlarge), security groups (firewall rules), and key pairs (SSH keys for secure access). Auto-scaling policies can be implemented to handle varying loads.

- **Amazon S3 Configuration:**

  - Set up an S3 bucket for static website hosting by enabling the "Static website hosting" option in the S3 management console.

  - Define bucket policies to allow public access for static content or restrict access through IAM roles if needed.

  - Configure lifecycle policies to manage data storage costs effectively (e.g., transitioning old files to S3 Glacier for archival storage).

Chapter 3 Task and Activities Performed

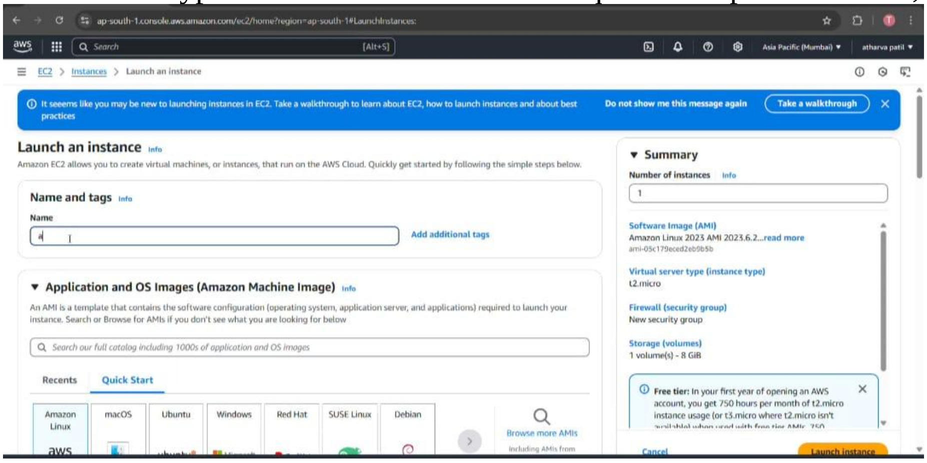
To successfully set up cloud web hosting for a website or application, several key tasks were carried out using AWS services like EC2 (Elastic Compute Cloud) for hosting dynamic content and S3 (Simple Storage Service) for hosting static files. Below are the specific tasks and activities that were performed:

3.1 EC2 Instance Creation and Setup

Task: Created EC2 instances for hosting the dynamic content of the website.

Activity:

1)Launch EC2 Instance: Using the AWS Management Console, an EC2 instance was launched. The instance type was selected based on the required compute resources, such as the instance size..



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

Search [Alt+S]

Asia Pacific (Mumbai) ▾ | atharva patil ▾

EC2 > Instances > Launch an instance

It seeems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Do not show me this message again | Take a walkthrough

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: a

Add additional tags

▾ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian | Browse more AMIs including AMIs from

▾ Summary

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-05c179eced2eb9b5b

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750

Cancel | Launch instance

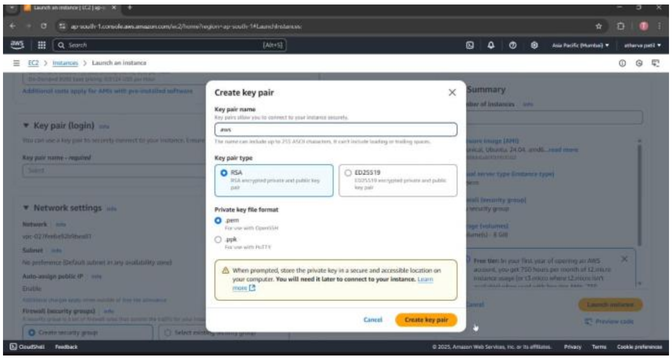
Figure 2: Instance Creation

2)Operating System Selection: An operating system was chosen (e.g., Ubuntu for a Linux-based server or Windows Server depending on the application requirements).

3)Security Group Configuration: A security group was configured to define which inbound and outbound traffic was allowed to the instance. Typically, port 22 (SSH) was opened for SSH access, and port 80 (HTTP) and port 443 (HTTPS) were opened to allow web traffic.

Key Pair Generation for Secure EC2 Access

4)Key Pair Creation: A key pair (comprising a public and private key) was generated from the AWS Management Console. This key pair ensures secure SSH access to the EC2 instance.



Launch an instance | EC2 | ap-s

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

Search [Alt+S]

Asia Pacific (Mumbai) ▾ | atharva patil ▾

EC2 > Instances > Launch an instance

Additional costs apply for AMIs with pre-installed software

▾ Key pair (login) Info

Key pair name - required

▾ Network settings Info

Network: vpc-027fee6e52b9bea87

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Firewall (security groups)

● Create security group ○ Select existing security group

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

aws

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Cancel | Create key pair

Summary

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more

Storage (volumes): 1 volume(s) - 8 GiB

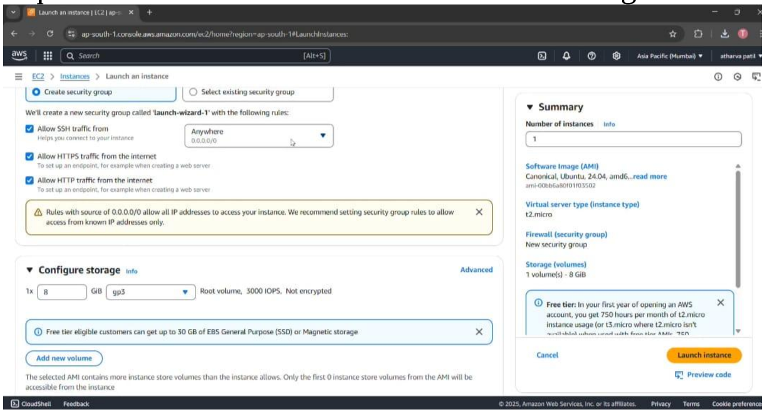
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't

Cancel | Launch instance | Preview code

CloudShell | Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences

Figure 3: Key Pair Generation

5)Download Key Pair: The private key file (.pem) was downloaded and stored securely, as it is required for SSH authentication when connecting to the EC2 instance.



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

EC2 > Instances > Launch an instance

● Create security group ○ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

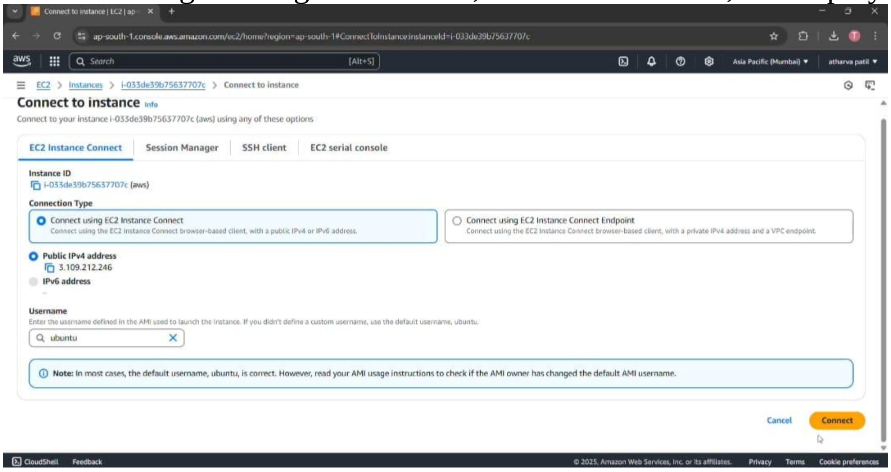
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750

6)Connecting to EC2: Using the private key, an SSH connection was established to the EC2 instance to begin configuration tasks, install web servers, and deploy the application.



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-033de39b75637707c

EC2 > Instances > i-033de39b75637707c > Connect to instance

Connect to instance

Connect to your instance i-033de39b75637707c (aws) using any of these options

Instance ID: i-033de39b75637707c (aws)

Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

● Public IPv4 address: 3.109.212.246

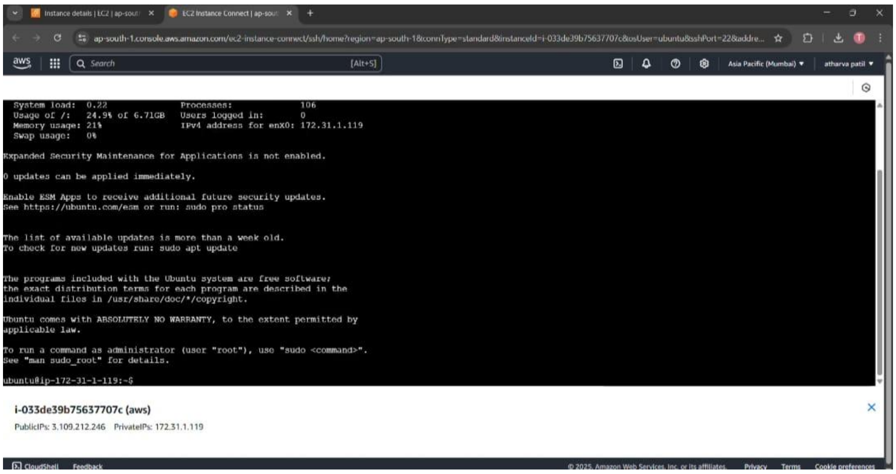
○ IPv6 address: -

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Figure 4: Connection With Instance



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-033de39b75637707c&osUser=ubuntu&sshPort=22&addre...

```
  System load:  0.22               Processes:             106
  Usage of /:   24.9% of 6.71GB   Users logged in:       0
  Memory usage: 21%                IPv4 address for enX0: 172.31.1.119
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-1-119:~$
```

i-033de39b75637707c (aws)

PublicIPs: 3.109.212.246 PrivateIPs: 172.31.1.119

Instance details | EC2 | ap-sout
EC2 Instance Connect | ap-sout
Search
[Alt+S]
Asia Pacific (Mumbai)
atharva patil
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Figure 5: Creating virtual server

3.2 S3 Bucket Creation for Hosting Static Content

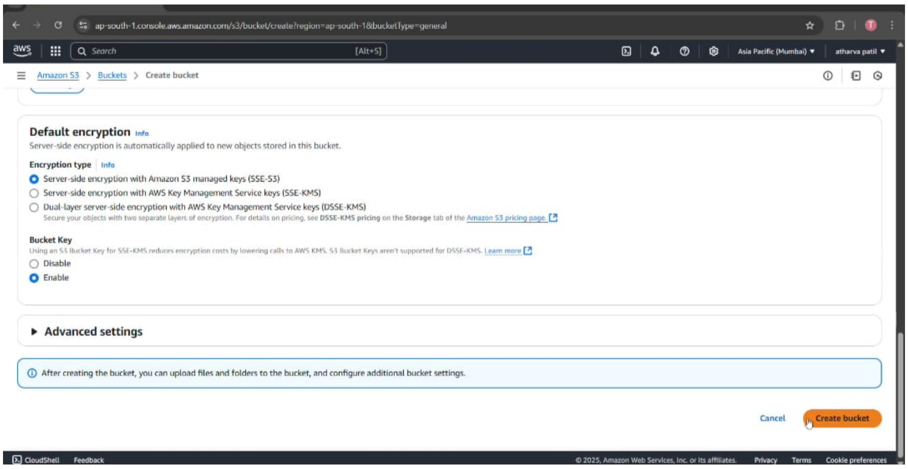
Task: Created an S3 bucket for storing and hosting static assets such as images, CSS, and JavaScript files.

Activity:

1)Create S3 Bucket: A new S3 bucket was created from the AWS Management Console. The bucket was given a unique name and configured with the desired settings (e.g., region selection).

2)Upload Static Content: Files such as images, stylesheets (CSS), and JavaScript (JS) were uploaded to the S3 bucket. This was done either via the AWS Management Console or using the AWS CLI (Command Line Interface) for bulk uploads.

3)Set Bucket Permissions: Permissions were configured for the S3 bucket to ensure that the files were publicly accessible. By default, S3 objects are private, so it was necessary to configure access controls.


ap-south-1.console.aws.amazon.com/s3/bucket/create?region=ap-south-1&bucketType=general
Search [Alt+S]
Asia Pacific (Mumbai)
atharva patil
Amazon S3 > Buckets > Create bucket
Advanced settings
Cancel
Create bucket
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

- ● Server-side encryption with Amazon S3 managed keys (SSE-S3)
- ○ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ○ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
  Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.

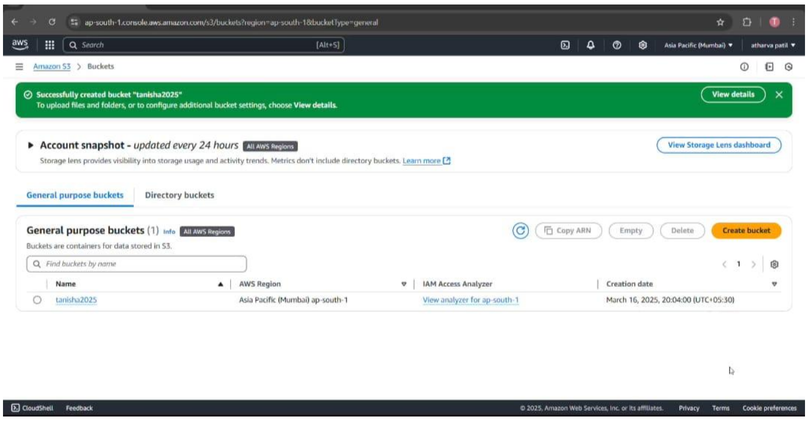
Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

- ○ Disable
- ● Enable

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Figure 6: Creation of Bucket



Successfully created bucket "tanisha2025"
To upload files and folders, or to configure additional bucket settings, choose View details.

Account snapshot - *updated every 24 hours* All AWS Regions

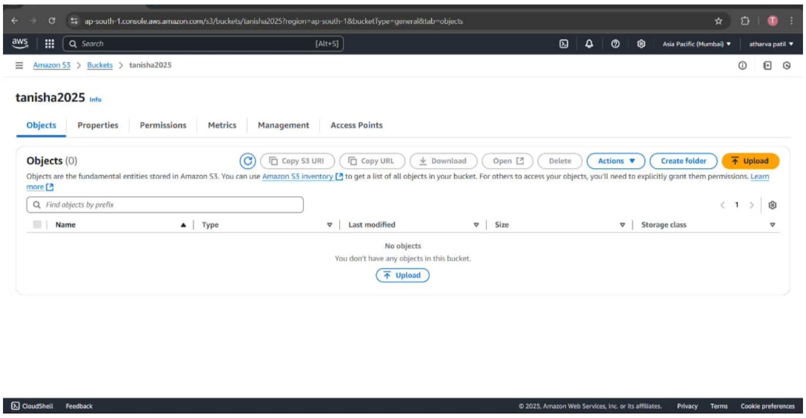
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
tanisha2025	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	March 16, 2025, 20:04:00 (UTC+05:30)

Figure 7: Uploading files in bucket



tanisha2025 Info

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class

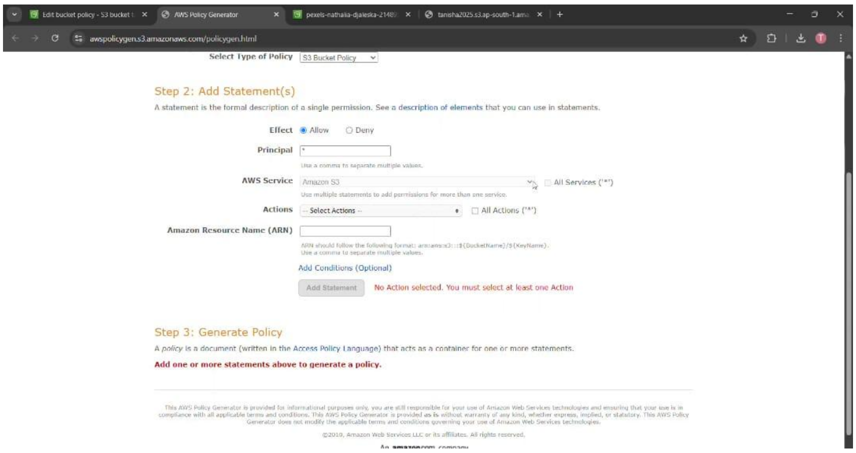
No objects

You don't have any objects in this bucket.

3.2.1. Generating Bucket Policy to Make Files Public

Task:

1)Create Bucket Policy: To enable public access to the uploaded files (images, CSS, JavaScript), a bucket policy was created in the S3 console. This policy allows public read access to all objects within the bucket.



Edit bucket policy - S3 bucket
AWS Policy Generator
pexels-nathalia-djaleska-2148
tanisha2025.s3.ap-south-1.ama
awspolicygen.s3.amazonaws.com/policygen.html

Select Type of Policy: S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect: ● Allow ○ Deny

Principal: *

Use a comma to separate multiple values.

AWS Service: Amazon S3 □ All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions: -- Select Actions -- □ All Actions ('*')

Amazon Resource Name (ARN):

ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement — No Action selected. You must select at least one Action

Step 3: Generate Policy

A *policy* is a document (written in the Access Policy Language) that acts as a container for one or more statements.

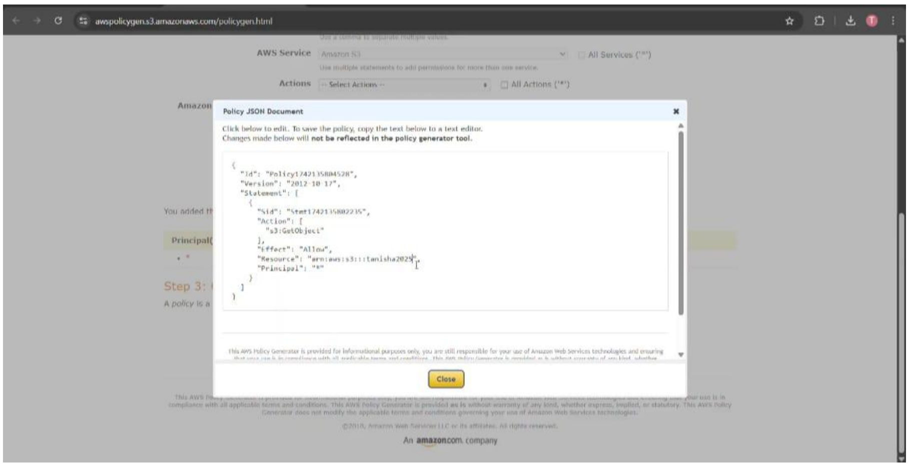
Add one or more statements above to generate a policy.

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.

Figure 8: Generating Policy

Example policy:



Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool.**

```
{
  "Id": "Policy1742135804528",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1742135802235",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::tanisha2025",
      "Principal": "*"
    }
  ]
}
```

Figure 9: Policy Code

2)Apply Policy: The policy was applied to the S3 bucket to allow any user to access the files publicly. This enabled the hosted files to be served over HTTP/S.

3.2.3. Hosting Static Content Publicly

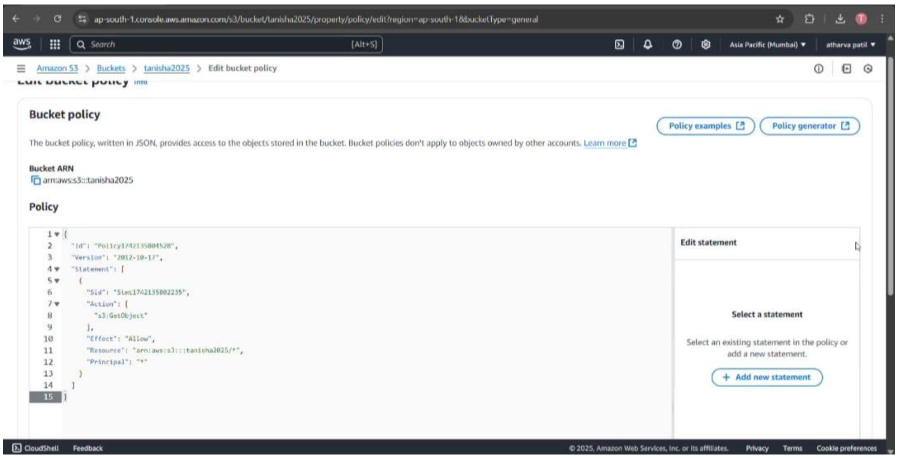
Task: Set up public access to the S3 bucket for hosting static content.

Activity:

1)Enable Static Website Hosting: The S3 bucket was configured to serve as a static website by enabling the "Static website hosting" option. The relevant settings were configured:

- Index Document: The name of the main HTML page (e.g., index.html) was specified as the index document.

- Error Document: An error document (e.g., error.html) was provided in case of any issues when loading a file.



Amazon S3 > Buckets > tanisha2025 > Edit bucket policy

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN

arn:aws:s3:::tanisha2025

Policy

```
{
  "Id": "Policy1742135804528",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1742135802235",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::tanisha2025/*",
      "Principal": "*"
    }
  ]
}
```

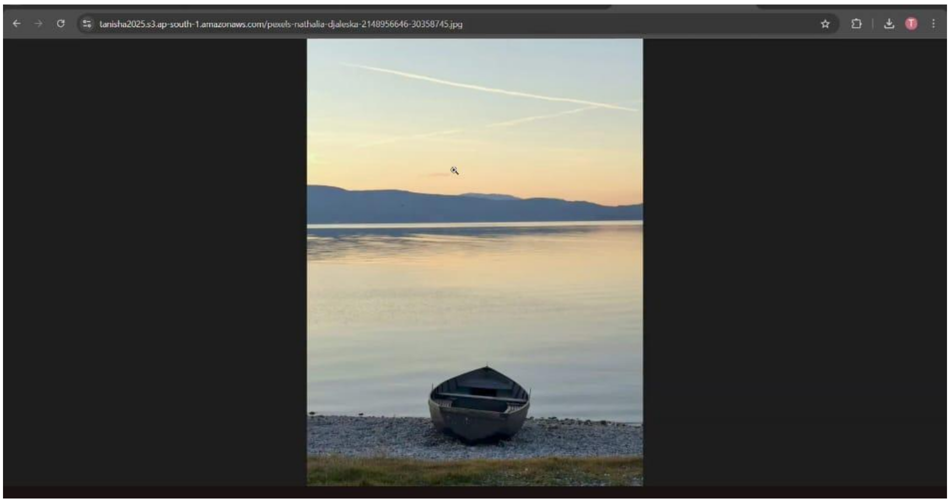
Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

ap-south-1.console.aws.amazon.com/s3/bucket/tanisha2025/property/policy/edit?region=ap-south-1&bucketType=general
Search
[Alt+S]
Asia Pacific (Mumbai)
atharva patil
Policy examples
Policy generator
+ Add new statement
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

2)Access URL: Once the bucket was configured for static website hosting, an endpoint URL was provided (e.g., http://your-bucket-name.s3-website-region.amazonaws.com). This URL could then be used to serve the static content.



tanisha2025.s3.ap-south-1.amazonaws.com/pexels-nathalia-djaleska-2148956646-30358745.jpg

Figure 10: Hosting Image Publicly

3.2.4. Future scope

The future scope of payment options in restaurant management websites is poised for transformative advancements. As technology evolves, the integration of cutting-edge payment solutions like digital wallets, contactless payments, and cryptocurrency is anticipated. Enhanced security measures, biometric authentication, and artificial intelligence-driven fraud detection will fortify financial transactions. The integration of seamless cross-border payment gateways and loyalty programs is expected to elevate customer experiences. Additionally, the adoption of emerging technologies such as blockchain may offer decentralized and secure payment infrastructures. The future promises a dynamic landscape where restaurant management websites play a pivotal role in shaping convenient, secure, and innovative payment ecosystems.

Chapter 4 Features and Applications

Features:

I. **Scalability:**
AWS provides easy scalability through services like EC2 (Elastic Compute Cloud), allowing you to adjust server capacity based on demand, without downtime.

II. **High Availability:**
AWS has a global network of data centers, ensuring your website can remain online even if one server or region goes down. Services like Elastic Load Balancing and Amazon Route 53 ensure smooth traffic management.

III. **Security:**
AWS offers robust security features such as firewalls, DDoS protection, encryption (data at rest and in transit), identity management via IAM (Identity and Access Management), and compliance certifications (e.g., HIPAA, GDPR).

IV. **Pay-As-You-Go Pricing:**
With AWS, you only pay for the resources you use, without the need for large upfront investments. This pay-per-use model helps save costs, especially for small businesses or startups.

V. **Global Infrastructure:**
AWS provides multiple Availability Zones (AZs) in different regions around the world, giving you a global presence and lower latency for users from various geographical locations.

VI. **Managed Services:**
AWS offers fully managed services like Amazon RDS (Relational Database Service), Amazon S3 (Simple Storage Service), Amazon CloudFront (CDN), and Amazon Lambda (serverless computing), reducing the need for manual configuration and management.

VII. **Automatic Scaling:**
AWS offers Auto Scaling and Elastic Load Balancing to automatically adjust the number of instances based on traffic demands, helping to handle traffic spikes efficiently.

VIII. **Backup and Disaster Recovery:**
AWS provides automatic backup, snapshots, and disaster recovery solutions to ensure data is protected and can be restored if necessary.

IX. **Advanced Analytics:**
With services like Amazon CloudWatch and AWS X-Ray, you can monitor your website's performance, user traffic, and other critical data for insights that help in optimization and troubleshooting.

X. **Easy Integration with Other AWS Services:**
AWS allows seamless integration with other cloud services such as AI/ML (Machine Learning), big data analytics, IoT, and more, enabling advanced functionality and growth potential.

APPLICATIONS of Cloud (AWS):

I. **Web Hosting**
AWS offers scalable hosting for websites and applications using services like EC2, S3, and CloudFront.

II. **Data Storage and Backup**
Services like S3 and Glacier provide secure, scalable storage and backup solutions.

III. **Big Data Analytics**
AWS tools like EMR and Redshift enable the processing and analysis of large datasets.

IV. **Machine Learning and AI**
SageMaker and other AI services allow companies to build, train, and deploy machine learning models.

V. **Mobile Application Backend**
AWS Amplify and other tools provide backend support for mobile applications with real-time syncing.

VI. **IoT (Internet of Things)**
AWS IoT Core connects and manages devices, offering real-time data processing and insights.

VII. **Content Delivery and CDN**
CloudFront delivers content with low latency globally, improving user experience.

VIII. **Disaster Recovery**
AWS offers reliable, cost-effective solutions for disaster recovery and data continuity.

IX. **Gaming**
AWS supports online game infrastructure, providing scalable backend services like game servers and storage.

X. **Enterprise IT Infrastructure**
AWS helps businesses migrate and manage their entire IT infrastructure on the cloud, reducing costs and improving scalability.

Chapter 5 Conclusion & References

5.1 Conclusion

In conclusion, this project highlights the effectiveness of using AWS EC2 and S3 for cloud web hosting. The integration of EC2 instances for dynamic content and S3 for static content enables scalable, secure, and cost-efficient hosting solutions. AWS's flexibility allows for seamless management and easy expansion as needed. The project demonstrated the practical applications of cloud computing in web hosting, offering reliability and high availability. Key benefits include simplified infrastructure management and enhanced performance. Future improvements could involve incorporating additional AWS services for further optimization. Overall, AWS proved to be an excellent choice for hosting both dynamic and static web applications.

5.2 References

1] "AWS EC2 Documentation - https://docs.aws.amazon.com/ec2/"

2] "AWS S3 Documentation - https://docs.aws.amazon.com/s3/