	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of Artificial Intelligence and Data Science**		
Student Name : Aniket Patil			
Class: TY-B TECH	Division: B	Roll No: 372004	
Semester: 5	Academic Year: 2023-24		
Subject Name & Code: Cloud Computing & Analytics ADUA31203			
Title of Assignment: Deploying a Web Application on AWS Cloud (PHP/Python/Node.js or any Application)			
Date of Performance: 15/9/23	Date of Submission: 30/9/23		

Aim:

Deploying a Web Application on AWS Cloud (PHP/Python/Node.js or any Application)

Write-up :

Introduction

Cloud computing is a technology that allows individuals and organizations to access and use computing resources (such as servers, storage, databases, networking, software, and analytics) over the internet. Deploying a web application on a cloud platform, such as Amazon Web Services (AWS), offers scalability, reliability, and flexibility. In this guide, we will walk you through the process of deploying a web application on AWS Cloud, using any popular web development stack like PHP, Python, or Node.js.

Cloud Computing Definition

Cloud computing is the delivery of various services over the internet, offering on-demand access to a pool of computing resources, including servers, storage, databases, networking, software, and more. These resources are hosted and managed by cloud service providers like AWS, making them accessible to users on a pay-as-you-go basis.

Cloud Service Models and Deployment Models

Cloud Service Models:

Infrastructure as a Service (IaaS): In IaaS, cloud providers offer virtualized computing resources over the internet. Users can rent virtual machines (VMs) and manage the operating systems, applications, and data on these VMs. Examples include AWS EC2 and Google Compute Engine.

Platform as a Service (PaaS): PaaS provides a platform that allows developers to build, deploy, and manage applications without worrying about the underlying infrastructure. Examples include AWS Elastic Beanstalk and Google App Engine.

Software as a Service (SaaS): SaaS delivers software applications over the internet, eliminating the need for users to install and maintain the software locally. Examples include Gmail and Microsoft Office 365.

Deployment Models:

Public Cloud: Public clouds are owned and operated by third-party cloud service providers. These providers offer services to the public and can be accessed by anyone over the internet. AWS, Microsoft Azure, and Google Cloud Platform are examples of public cloud providers.

Private Cloud: Private clouds are used exclusively by a single organization. They can be hosted on-premises or by a third-party provider. Private clouds offer more control and security but require higher upfront costs.

Hybrid Cloud: Hybrid clouds combine both public and private cloud models, allowing data and applications to be shared between them. This approach provides flexibility and can help organizations optimize their computing resources.

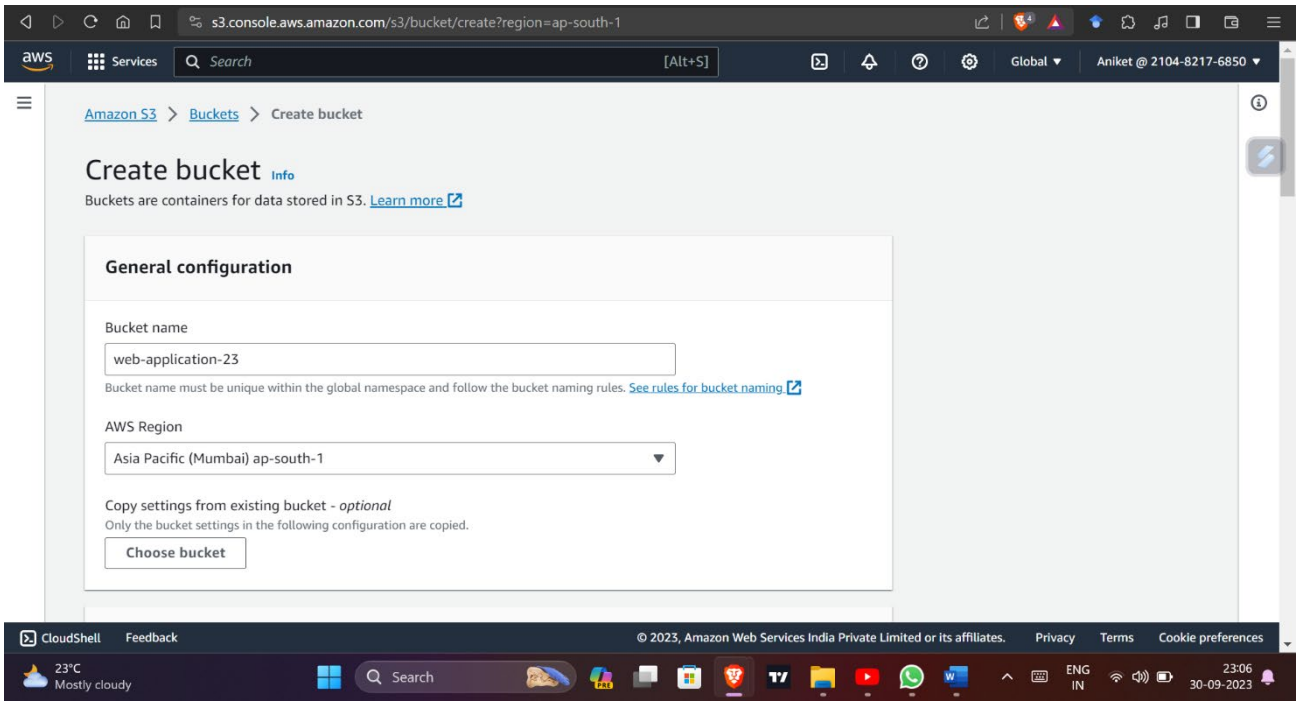
Results of Experimentation:

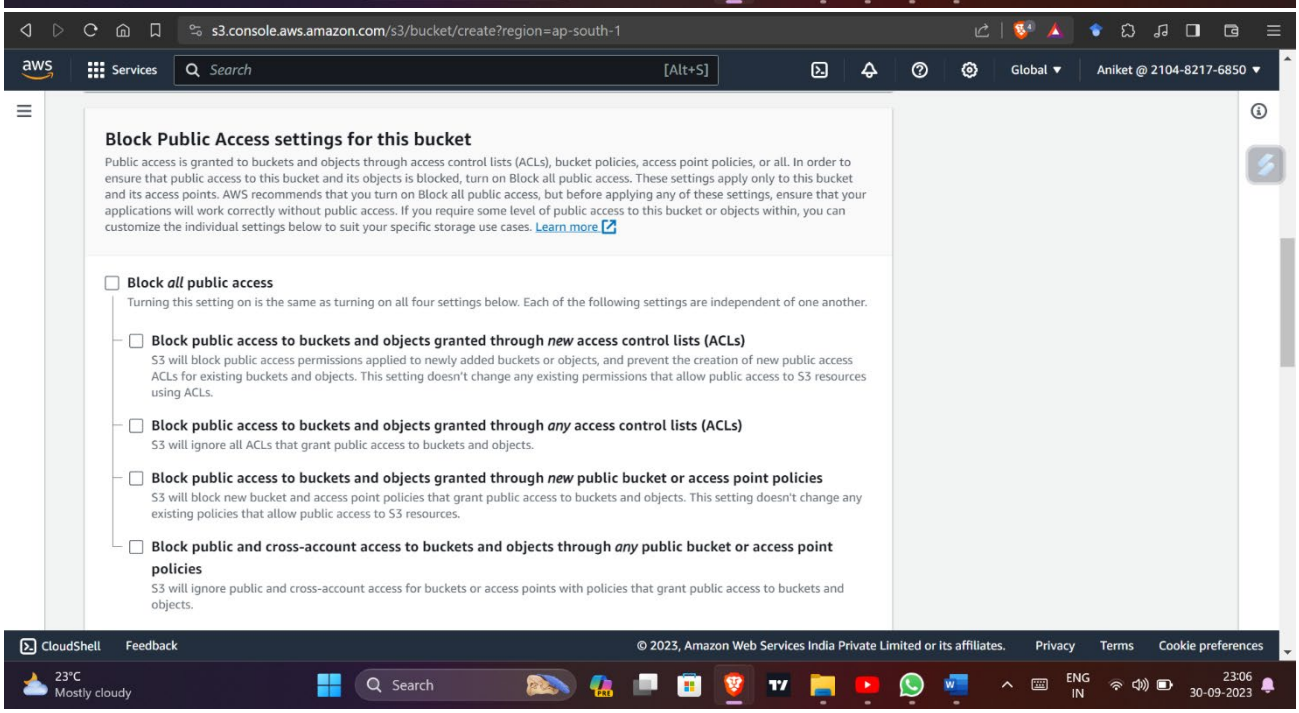
Step-by-Step Guide: Deploying a Web Application on AWS:

Step 1: Sign in to AWS Console

- Visit the AWS Management Console.
- Sign in using your AWS account credentials.

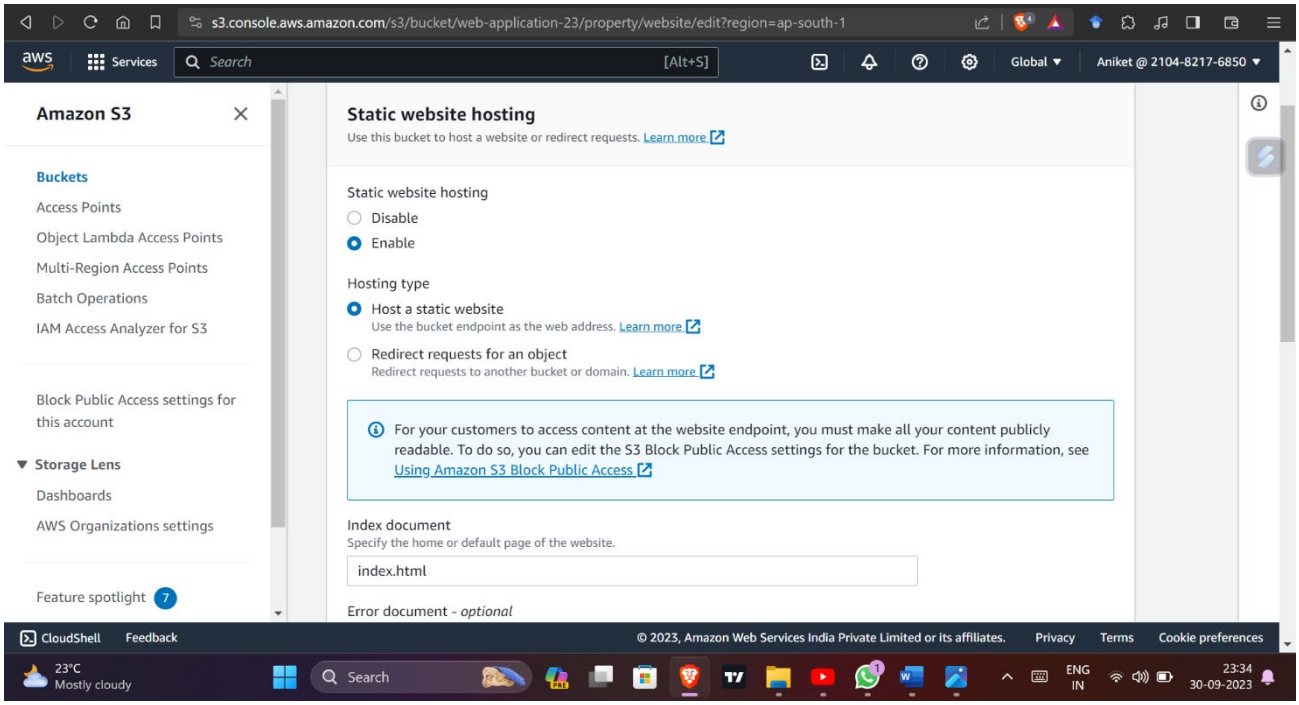
Step 2: Create an S3 Bucket





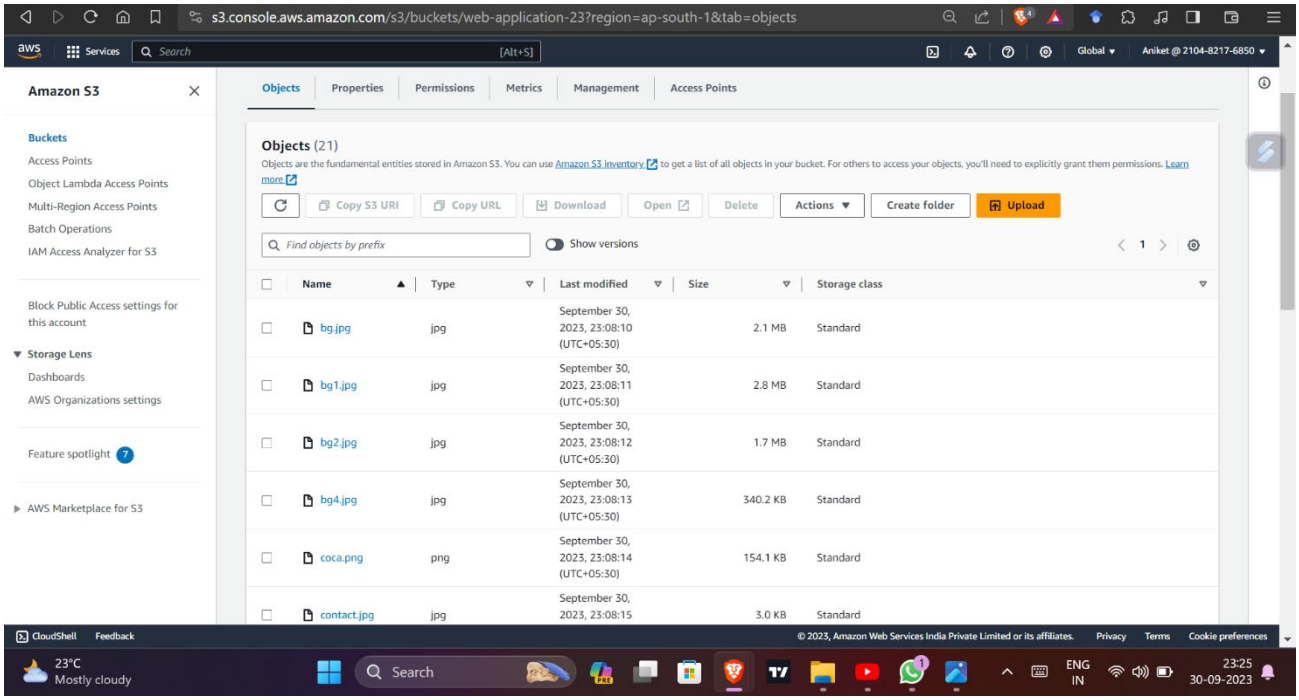
- Navigate to the Amazon S3 service.
- Click on the "Create bucket" button.
- Choose a unique and meaningful name for your bucket. This name will be part of your website's URL.
- Select the AWS region for your bucket. Choose a region close to your target audience for better performance.
- Configure additional settings as needed (e.g., versioning, logging), and click "Create."

Step 3: Configure Static Website Hosting



- Select the newly created bucket in the S3 dashboard.
- Click on the "Properties" tab.
- Under the "Static website hosting" section, choose "Use this bucket to host a website."
- Specify the index document (e.g., index.html) and an optional error document.
- Click "Save.

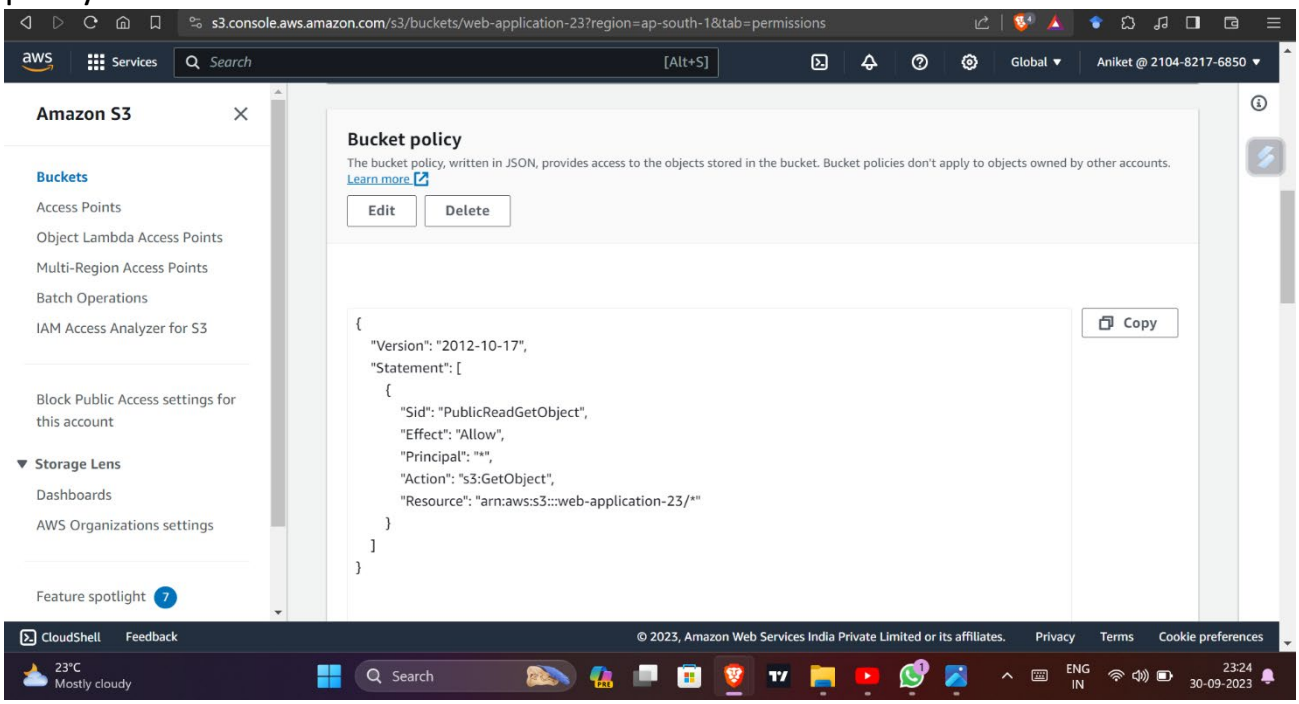
Step 4: Upload Website Content



- Go to the "Overview" tab of your bucket.
- Click the "Upload" button to upload your website's static files (HTML, CSS, JavaScript, images, etc.) to the bucket. You can also create folders to organize your content.

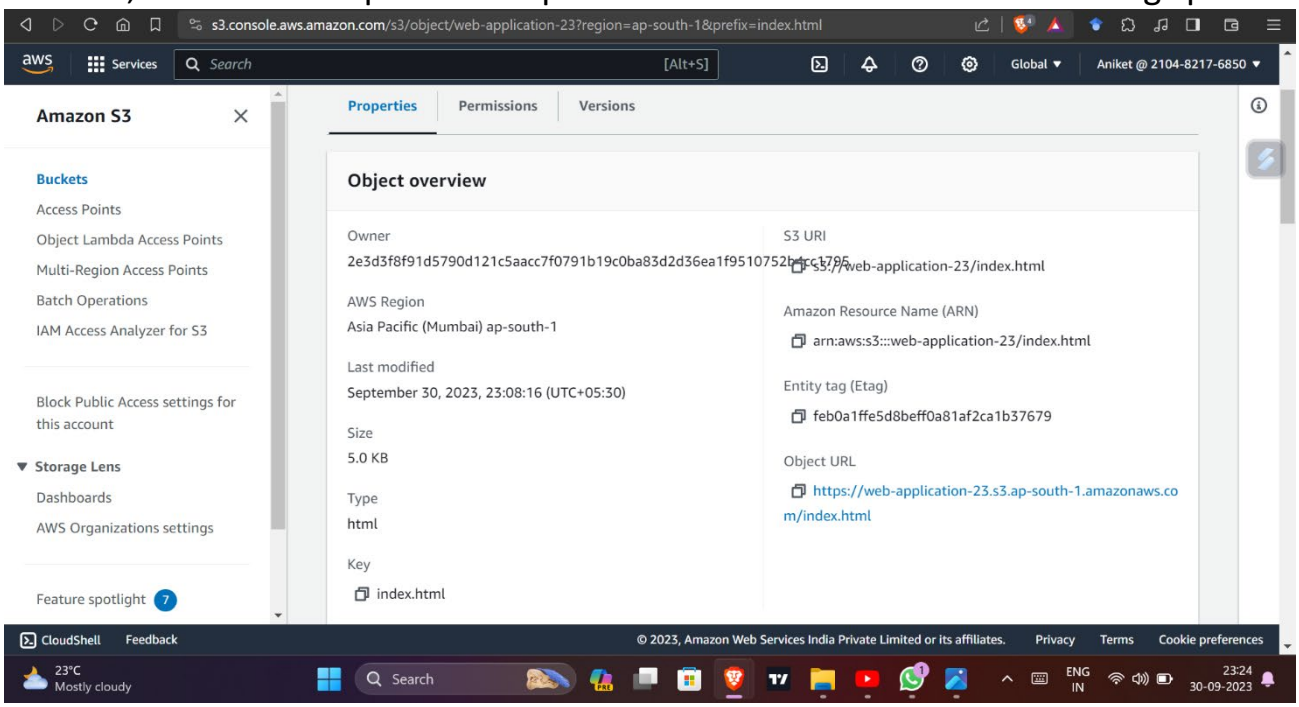
Step 5: Set Bucket Policy

If you want to restrict or grant specific permissions, you can create a bucket policy. For a basic setup that allows public access, you can use the following policy:



Step 6: Access Your Website

After configuring the bucket for static website hosting and uploading your content, note the "Endpoint" URL provided on the "Static website hosting" panel.



You can access your website by navigating to this URL. It should display the contents of your index.html or the specified index document.

Link of my static website:

http://web-application-23.s3-website.ap-south-1.amazonaws.com/

Conclusion:

In this assignment, we have learned how to host a static website using Amazon S3. Amazon S3's simplicity, scalability, and cost-effectiveness make it an excellent choice for hosting static web content. Whether you're creating a personal blog, a portfolio site, or a simple landing page, you can easily get your content online using S3's static website hosting feature.