



Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra
Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab

Assignment 01

Aim:

Part 1 : To develop a website and host it on your local machine on a VM

Part 2 : To host the website developed as part 1 of Assignment 1 using AWS.

Roll No.	42
Name	Naikwadi Yash Shivdas
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.
Grade:	

AIM : To develop a website and host it on your local machine on a VM Reference and hosting a static website on Amazon S3 (AWS).

THEORY :

Introduction

In DevOps, this experiment involves developing a website using a tech stack including HTML, CSS, JavaScript, and frameworks like React or Angular, and backend technologies such as Node.js or Python. The website is first hosted on a local development environment, then transitioned to Amazon S3 for scalable cloud-based hosting. This approach highlights the advantages of both local and cloud environments.

Hosting on a Local Machine Using Xampp

Setting Up a Local Development Environment with XAMPP :

XAMPP is an open-source cross-platform web server solution stack package developed by Apache Friends. It includes:

- Apache: A widely-used web server software.
- MySQL/MariaDB: Database management systems.
- PHP: A server-side scripting language.
- Perl: A high-level programming language.

Pros :

- Complete control over the development environment.
- Useful for development and testing phases.

Cons :

- Limited scalability.
- Requires manual management of infrastructure and updates.

Hosting a Static Website on Amazon S3 (AWS)

A static website consists of fixed content with HTML files and does not require server-side processing. This type of website is typically faster and easier to host.

Introduction to AWS S3

Amazon S3 (Simple Storage Service) is a scalable object storage service that provides a simple web services interface to store and retrieve any amount of data at any time from anywhere on the web.

Pros :

- Highly scalable and cost-effective.
- Minimal management required.
- High availability and durability of data.

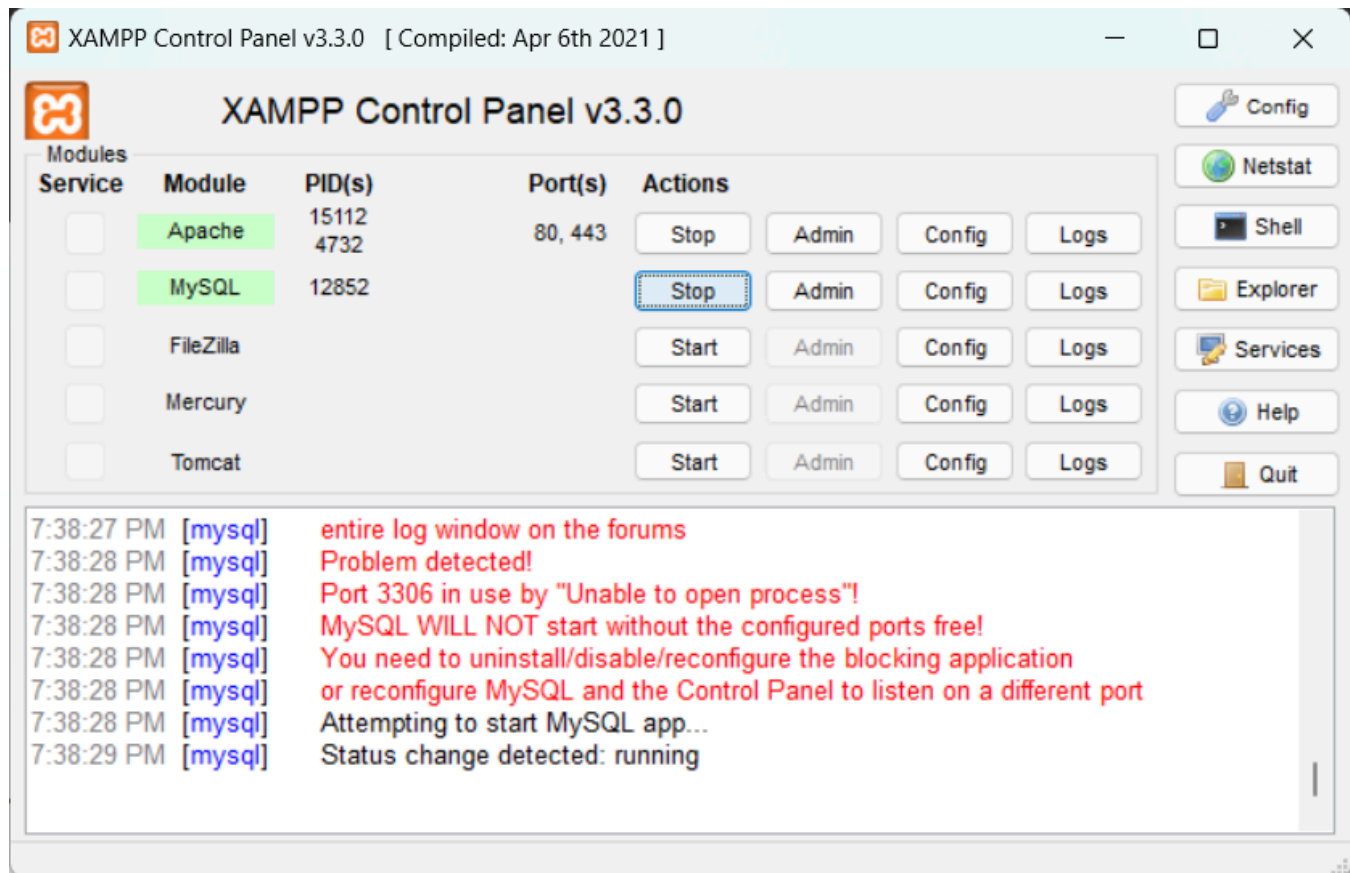
Cons :

- Limited to static content.
- Less control over the hosting environment compared to a VM.

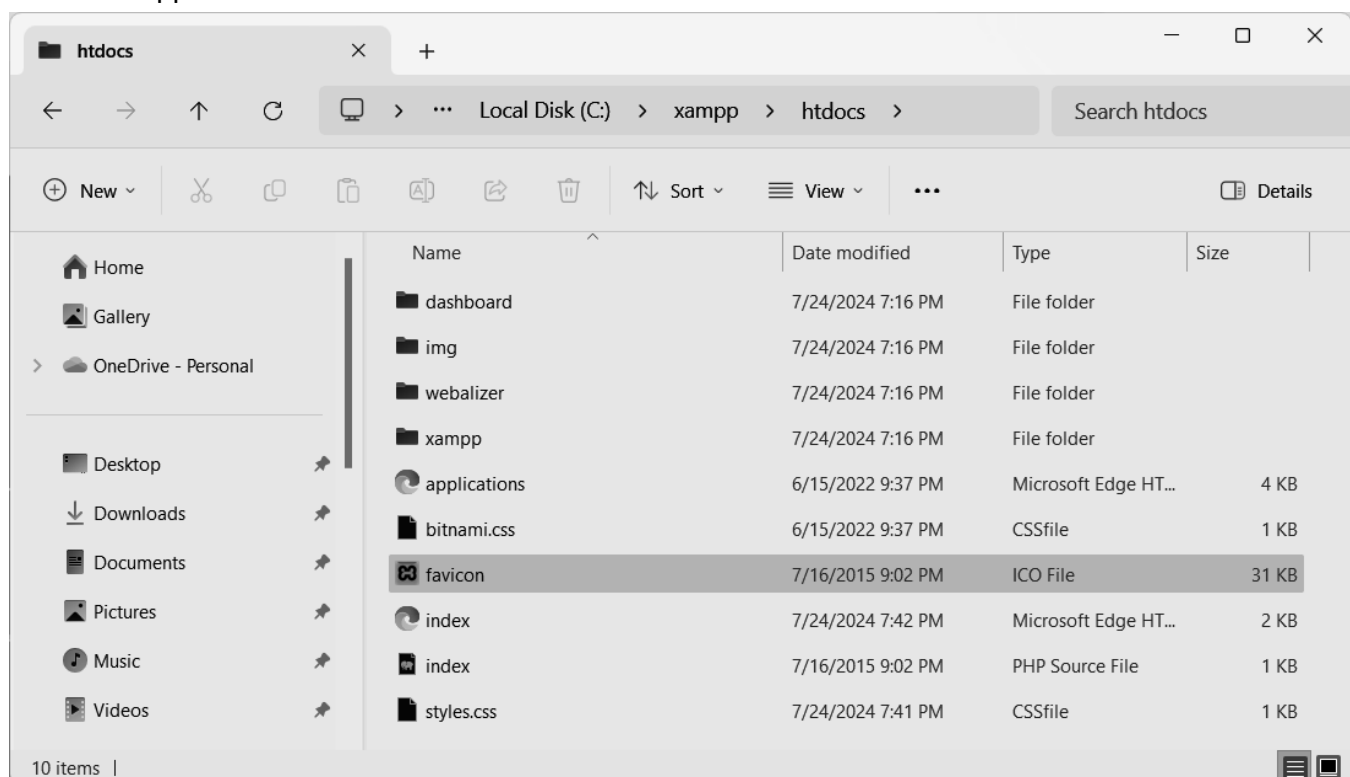
Aspect	Local Hosting (VM with XAMPP)	Cloud Hosting (AWS S3)
Performance	Works well for development but may struggle with high traffic in production.	Usually performs better with global access and fast content delivery.
Security	You're responsible for setting up and managing security.	AWS provides strong security features, reducing your management burden.
Accessibility	Only accessible on your local network unless configured otherwise.	Accessible from anywhere, great for public websites.
Cost	No extra hosting fees, but you handle all maintenance and updates.	Pay for what you use, which can be cost-effective but needs monitoring.
Deployment	Requires manual file transfers and updates.	Supports automated deployment with tools like AWS CodePipeline.
Continuous Integration	Less automated; may need custom scripts.	Easily integrates with CI/CD tools for automated updates.
Scalability	Limited; adding more servers is manual and complex.	Automatically adjusts to traffic changes and scales easily.
Maintenance	You handle all updates and server management.	AWS manages the infrastructure, so less manual work for you.

Hosting on a Local Machine Using Xampp

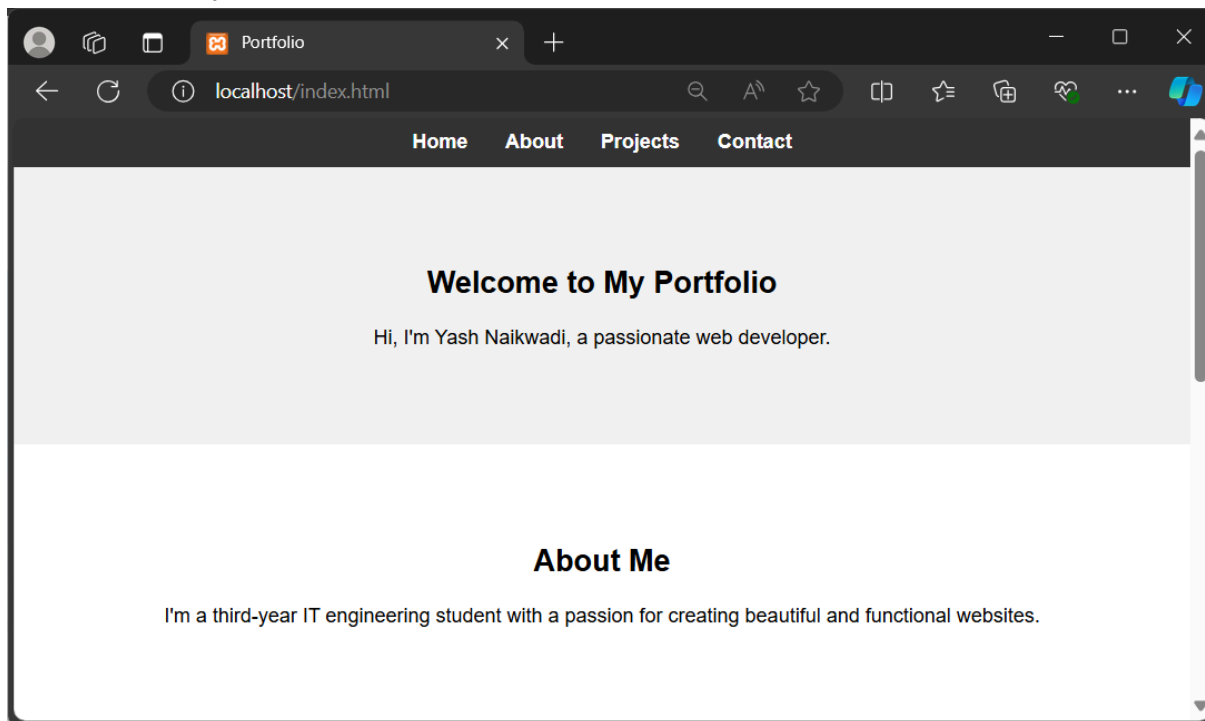
1. Download, Install and Launch Xampp. Start the actions of Apache and MySQL.



2. Create an index.html and its corresponding css file. Save both files in an appropriate folder as xampp => htdocs => index.html.

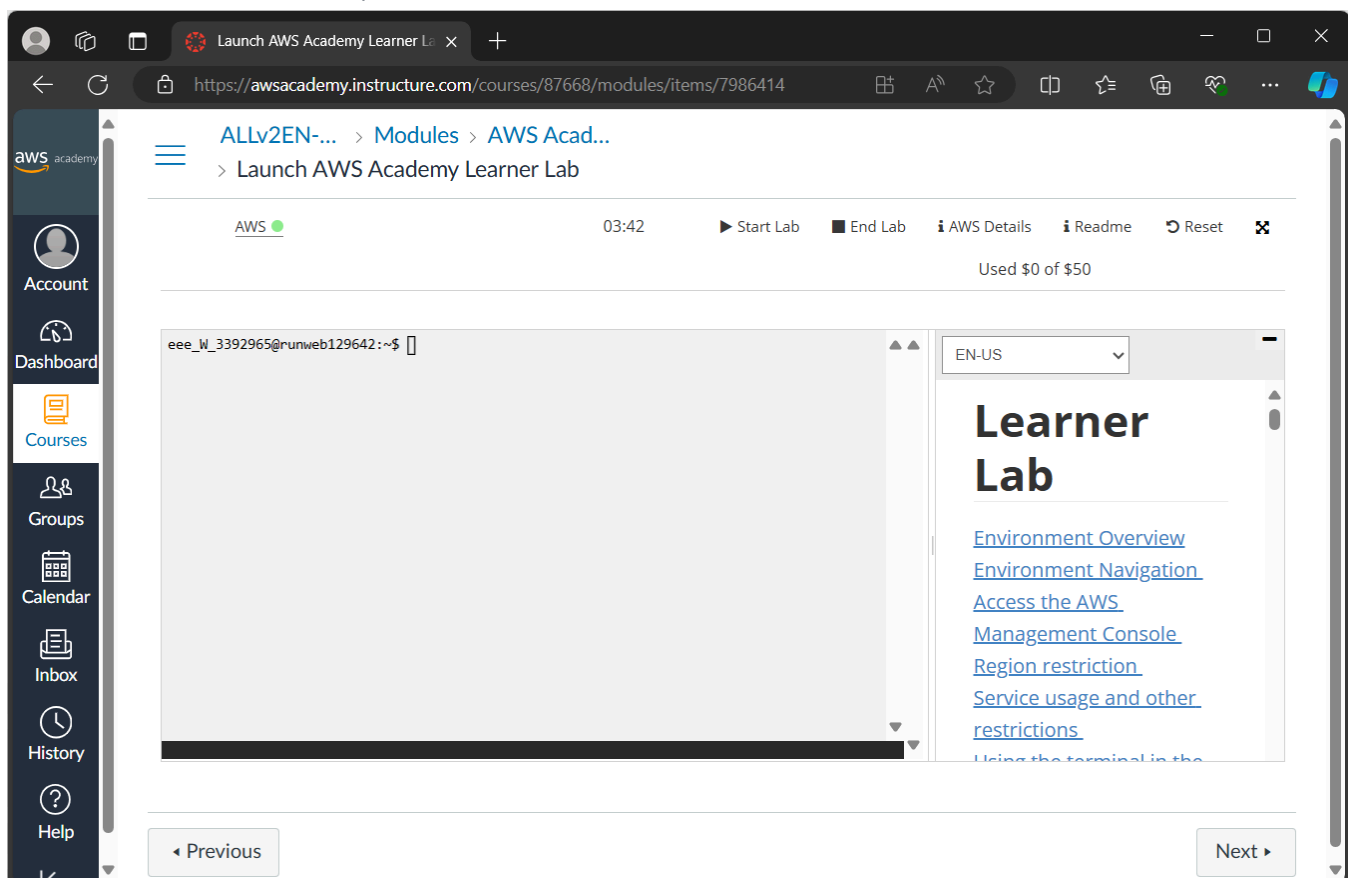


3. Go to any browser and search for <http://localhost/index.html>

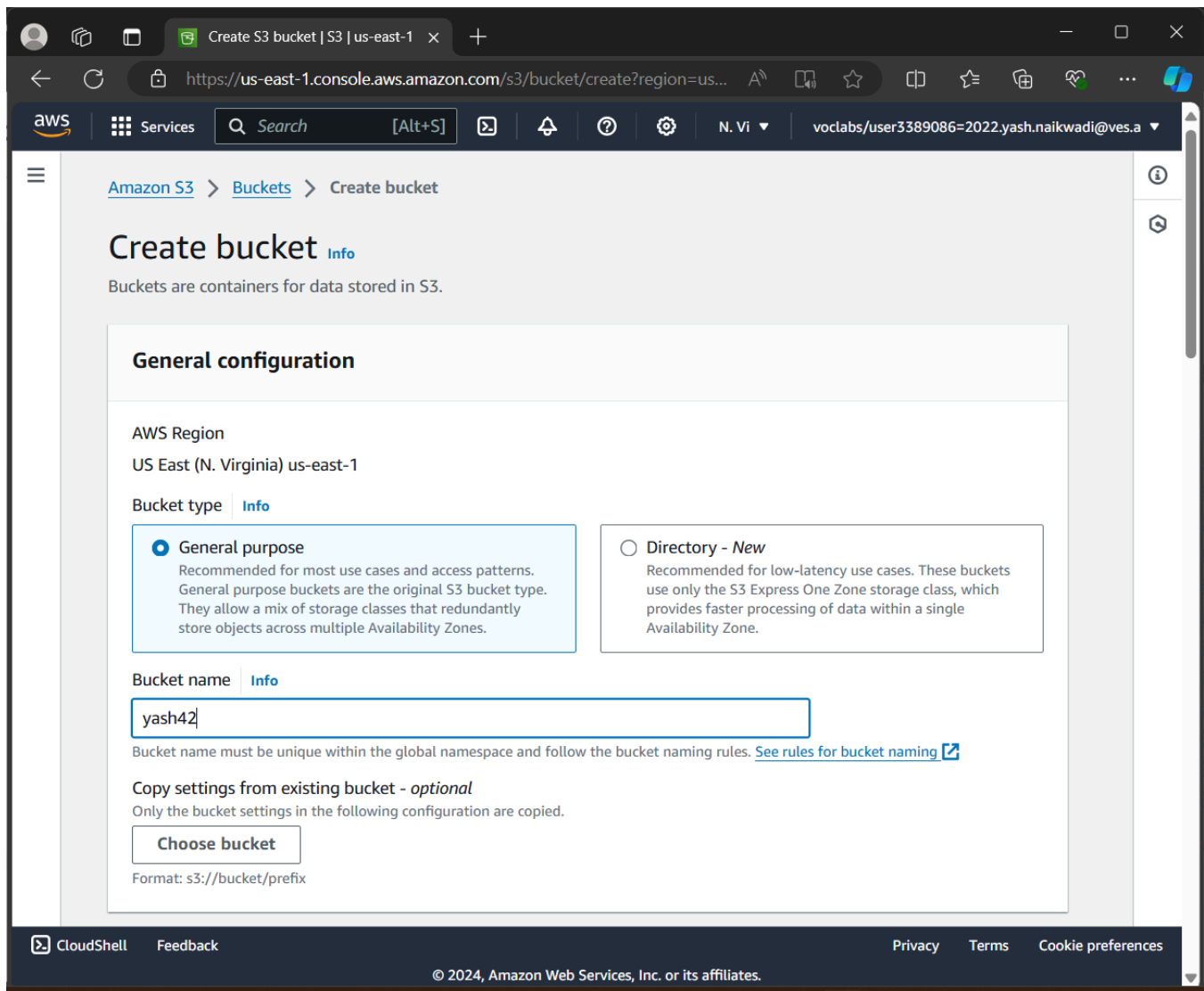


Hosting a Static Website on Amazon S3 (AWS)

1. Go to AWS academy website. Solve the Module Knowledge Check. Launch AWS Academy Learner Lab and click on **AWS** beside the green logo. (logo will become green once the start lab is clicked.)



2. Search for S3 and create a bucket.



The screenshot displays the AWS Management Console interface for creating a new S3 bucket. The browser address bar shows the URL: `https://us-east-1.console.aws.amazon.com/s3/bucket/create?region=us...`. The console header includes the AWS logo, a search bar, and the user's profile information.

The main content area is titled "Create bucket" and includes a sub-header "Buckets are containers for data stored in S3." Below this, the "General configuration" section is visible. It contains the following details:

- AWS Region:** US East (N. Virginia) us-east-1
- Bucket type:** General purpose (selected) and Directory - New (unselected). The "General purpose" option is described as recommended for most use cases and access patterns, allowing a mix of storage classes that redundantly store objects across multiple Availability Zones.
- Bucket name:** yash42. A note below the field states: "Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)".
- Copy settings from existing bucket - optional:** Only the bucket settings in the following configuration are copied. A "Choose bucket" button is present.
- Format:** s3://bucket/prefix

The footer of the console shows links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with the copyright notice: © 2024, Amazon Web Services, Inc. or its affiliates.

3. Click on the created bucket and upload the index.html and its corresponding css.

The screenshot shows the AWS S3 console interface for a bucket named 'yash42'. The breadcrumb navigation is 'Amazon S3 > Buckets > yash42'. The bucket name 'yash42' is displayed with an 'Info' link. Below the bucket name are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is active, showing 'Objects (2)' with an 'Info' link. Above the object list are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', and 'Delete'. Below these are 'Actions' (a dropdown), 'Create folder', and 'Upload' buttons. A text box explains that objects are fundamental entities in S3 and provides links to 'Amazon S3 inventory' and 'Learn more'. A search bar labeled 'Find objects by prefix' is present. The object list table has columns for Name, Type, Last modified, Size, and Storage class. It contains two entries: 'index.html' (1.4 KB, Standard) and 'styles.css' (830.0 B, Standard), both last modified on July 24, 2024, at 21:43:51 and 21:43:52 respectively. The footer includes 'CloudShell', 'Feedback', 'Privacy', 'Terms', 'Cookie preferences', and a copyright notice for 2024 Amazon Web Services, Inc.

Amazon S3 > Buckets > yash42

yash42 [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2) [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#)

[Actions](#) [Create folder](#) [Upload](#)

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](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	index.html	html	July 24, 2024, 21:43:51 (UTC+05:30)	1.4 KB	Standard
<input type="checkbox"/>	styles.css	css	July 24, 2024, 21:43:52 (UTC+05:30)	830.0 B	Standard

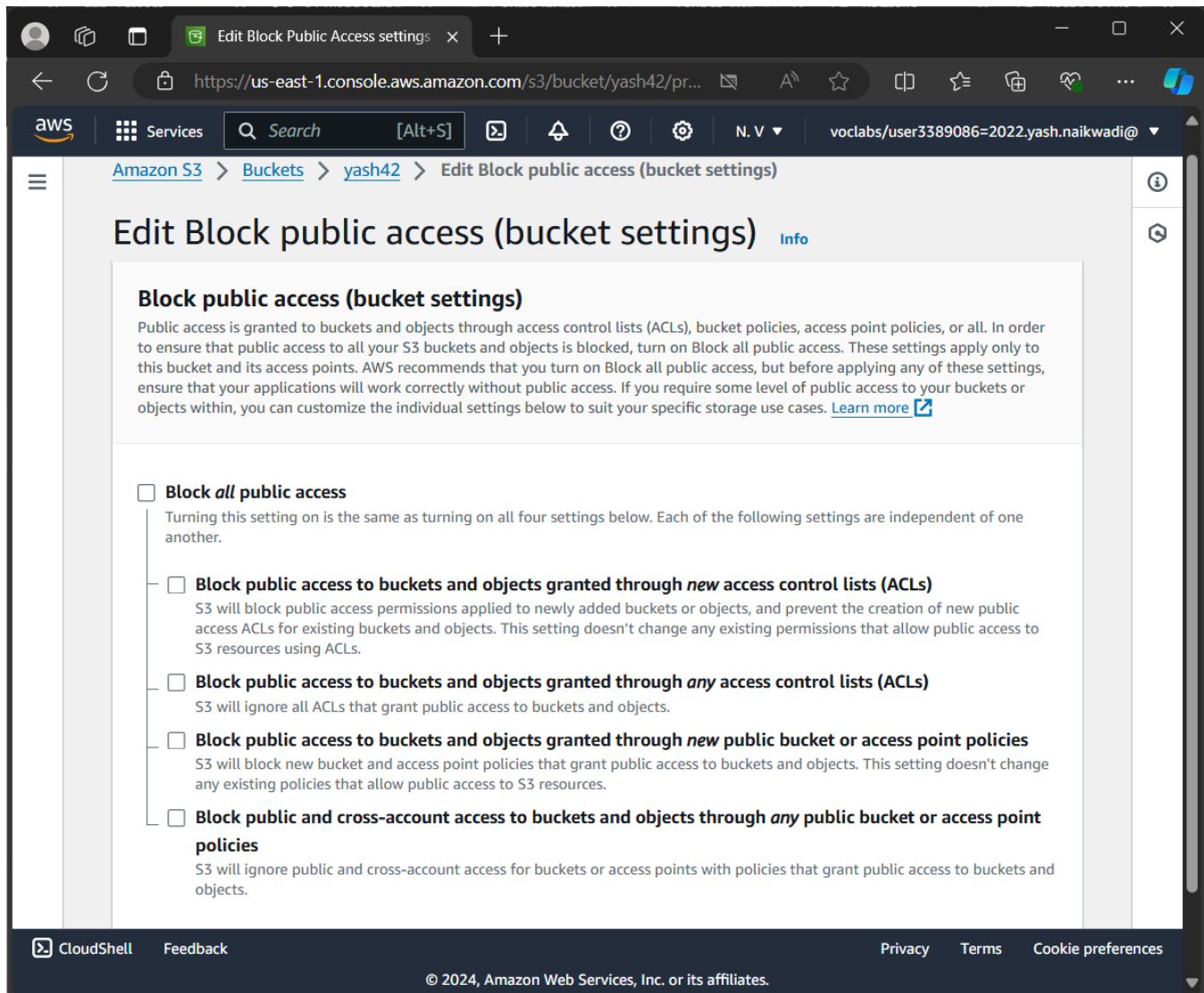
CloudShell Feedback Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

4. Go to the Properties section and enable the static website hosting.

The screenshot shows the AWS Management Console interface for editing static website hosting on an S3 bucket named 'yash42'. The breadcrumb navigation at the top indicates the path: Amazon S3 > Buckets > yash42 > Edit static website hosting. The main heading is 'Edit static website hosting' with an 'Info' link. Below this, the 'Static website hosting' section is expanded, showing two options: 'Disable' and 'Enable', with 'Enable' selected. The 'Hosting type' section also has two options: 'Host a static website' (selected) and 'Redirect requests for an object'. A blue information box states: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access'. At the bottom, the 'Index document' section has a text input field containing 'index.html'. The footer of the console shows 'CloudShell', 'Feedback', 'Privacy', 'Terms', 'Cookie preferences', and a copyright notice for 2024 Amazon Web Services, Inc. or its affiliates.

5. Unselect the (main) option of Block public access. (By default, it is selected while creating the bucket). It gives public access to use our website.



The screenshot shows the AWS console interface for editing block public access settings for a bucket named 'yash42'. The breadcrumb navigation is 'Amazon S3 > Buckets > yash42 > Edit Block public access (bucket settings)'. The main heading is 'Edit Block public access (bucket settings)' with an 'Info' link. Below this is a section titled 'Block public access (bucket settings)' with explanatory text. The main setting, 'Block all public access', is unchecked. Below it are four sub-settings, all of which are also unchecked:

- ☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

The footer of the console shows 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

6. Go to the permission section in the new created bucket and edit bucket policy.

The screenshot shows the AWS Management Console interface for editing the bucket policy of a bucket named 'yash42'. The breadcrumb navigation at the top indicates the path: Amazon S3 > Buckets > yash42 > Edit bucket policy. The main heading is 'Edit bucket policy' with an 'Info' link. Below this, there are buttons for 'Policy examples' and 'Policy generator'. A descriptive text states: '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](#)'. The 'Bucket ARN' is displayed as 'arn:aws:s3:::yash42'. The 'Policy' section shows a JSON snippet with a single statement: 'PublicReadGetObject', which is expanded to show its details: 'Sid': 'PublicReadGetObject', 'Effect': 'Allow', 'Principal': {'AWS': '*'}, 'Action': 's3:GetObject', and 'Resource': 'arn:aws:s3:::yash42/*'. To the right of the JSON editor is a panel titled 'Edit statement' which contains the text 'Select a statement' and 'Select an existing statement in the policy or add a new statement.', along with an 'Add new statement' button. The footer of the console includes links for 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences', and a copyright notice for 2024 Amazon Web Services, Inc. or its affiliates.

Edit bucket policy Info

Bucket policy

Policy examples Policy generator

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:::yash42

Policy

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "PublicReadGetObject",
6       "Effect": "Allow",
7       "Principal": {
8         "AWS": "*"
9       },
10      "Action": "s3:GetObject",
11      "Resource": "arn:aws:s3:::yash42/*"
12    }
13  ]
14 }
```

Edit statement

Select a statement

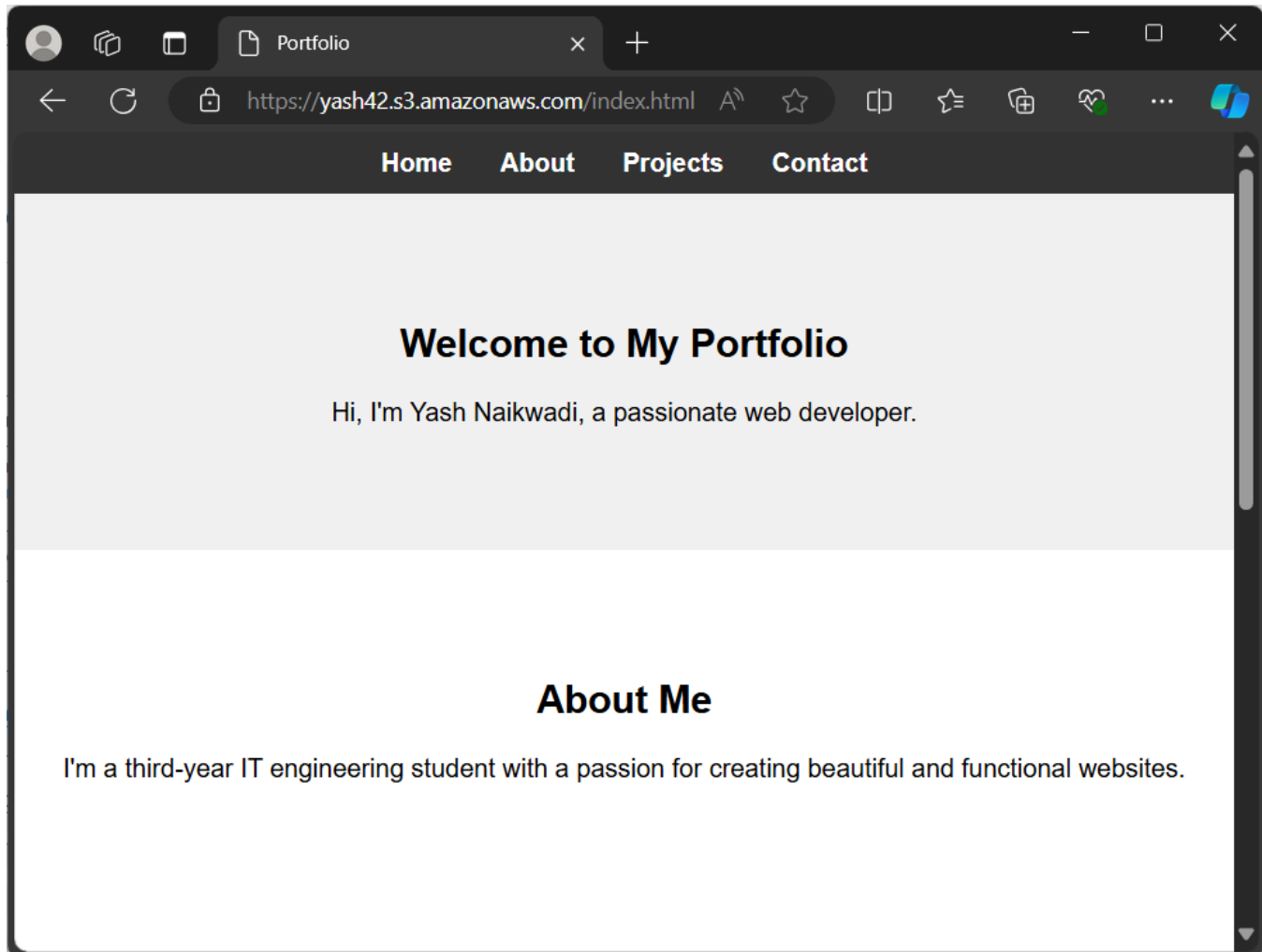
Select an existing statement in the policy or add a new statement.

+ Add new statement

CloudShell Feedback Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

7. Go to the Objects section and select the index.html file. Then the Copy URL option will get activated. Click on it and paste it on the new tab.



CONCLUSION :

This experiment demonstrates the process of developing a website and the flexibility of hosting it on different platforms. Hosting on a local VM provides insights into infrastructure management, while hosting on AWS S3 showcases the benefits of cloud-based solutions. Understanding both approaches equips one with the skills to choose the appropriate hosting solution based on the project requirements and scalability needs.