

Project Title: Hosting a Pet Shop Website Using AWS S3 Services

- **Hosting a Pet Shop Website Using AWS S3 Services:**

This document provides a step-by-step guide to host a Pet Shop static website using Amazon S3, which is an object storage service offered by AWS. The guide demonstrates how to configure your S3 bucket for public static website hosting and how to manage permissions and logging.

- **Key Features & Implementations:**

1. **Static Frontend Hosting:**

HTML/CSS/JavaScript-based UI mimicking the Pet shopping experience.

2. **Fast and Scalable Hosting via AWS S3:**

Hosted on an Amazon S3 bucket configured for static website hosting. Auto-scales globally with S3's performance and reliability.

3. **Public Accessibility:**

Static website is publicly accessible using a clean S3 website URL.
Bucket policies and ACLs allow public read access to site files.

4. **Versioning and Backup:**

S3 Versioning enabled to keep track of all versions of the files.
Allows easy rollback and backups in case of issues.

5. **Access Control:**

Controlled via AWS IAM policies, ensuring only authorized users can update or manage the bucket. Fine-grained permissions allow for dev/admin role separation.

6. **Logging and Monitoring:**

Server access logging enabled to track visits and activity on the site.
Logs stored in a separate S3 bucket (e.g., Pet Shop-log) for analytics and auditing.

7. **(Optional) CI/CD Deployment:**

Integration with GitHub Actions or AWS Code Pipeline for automated file uploads on commit.

- **Technology Used:** -

The Pet Shop static website hosted on AWS S3 primarily uses the following technologies:

1. Frontend Technologies:

- **HTML** – For structuring the content
- **CSS** – For styling (can include frameworks like Bootstrap or Tailwind)
- **JavaScript** – For client-side interactivity (can include jQuery or vanilla JS)
- **Images & Assets** – Product images, logos, etc.

2. Hosting & Cloud Infrastructure:

These AWS technologies are used for hosting:

- **Amazon S3 (Simple Storage Service)** – Main hosting service for static content
- **S3 Static Website Hosting** – Special configuration to serve files as a website
- **Access Control Lists (ACLs)** – For managing public access to files
- **Bucket Policies** – (Optional) For fine-grained access control
- **Versioning** – To keep backup versions of your files
- **Server Access Logging** – For tracking usage and access.

- **Outcome:** -

1. Live Static Website:

The Pet Shop Static Website is publicly accessible through a custom S3 website URL, simulating a professional e-commerce front end.

2. Secure & Scalable Hosting:

AWS S3 provides high availability and durability for static content, making your site reliable without server management.

3. Cost-Effective Deployment:

Since S3 is a pay-as-you-go service, hosting static sites is extremely cheap—ideal for prototypes, portfolios, and frontend projects.

- **Impact:** -

1. Real-World Cloud Experience:

Demonstrates practical knowledge of AWS infrastructure, particularly for web developers and cloud beginners.

2. Portfolio Enhancement:

Hosting a live clone of a known platform (like Pet Shop) showcases your frontend and cloud skills to potential employers or clients.

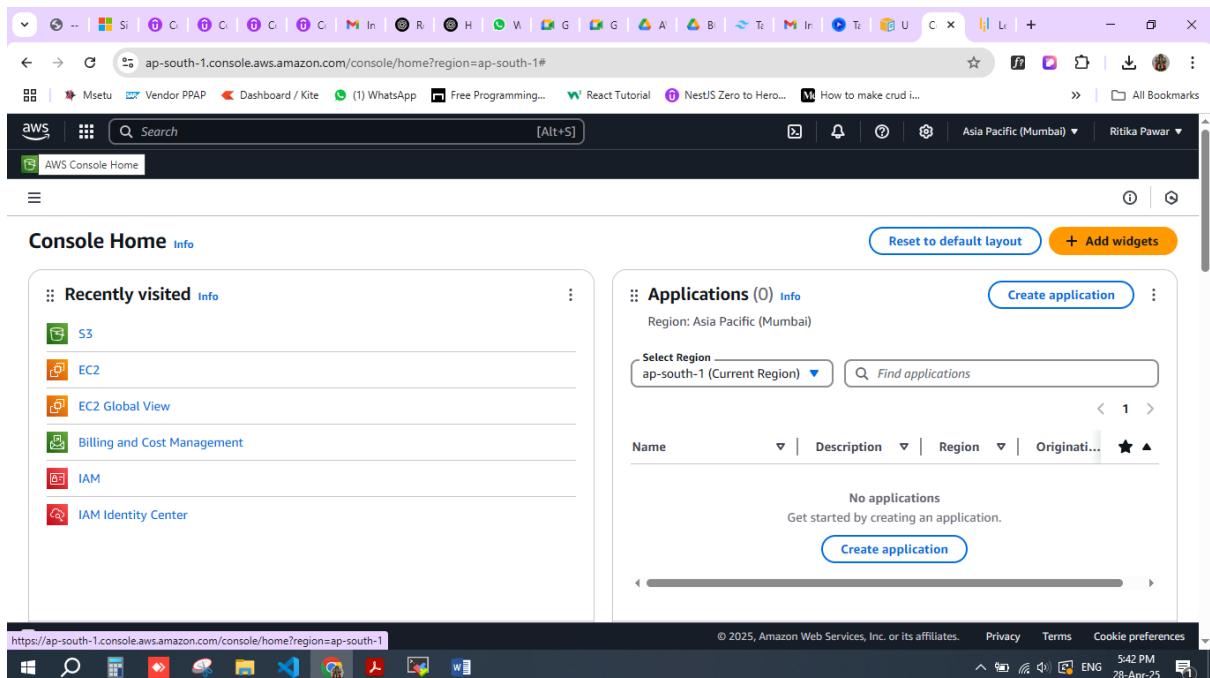
3. Foundation for Future Projects:

Sets the stage for enhancements like integrating a backend (with Lambda, API Gateway, or Firebase), adding a domain name, enabling HTTPS with CloudFront, or CI/CD pipelines.

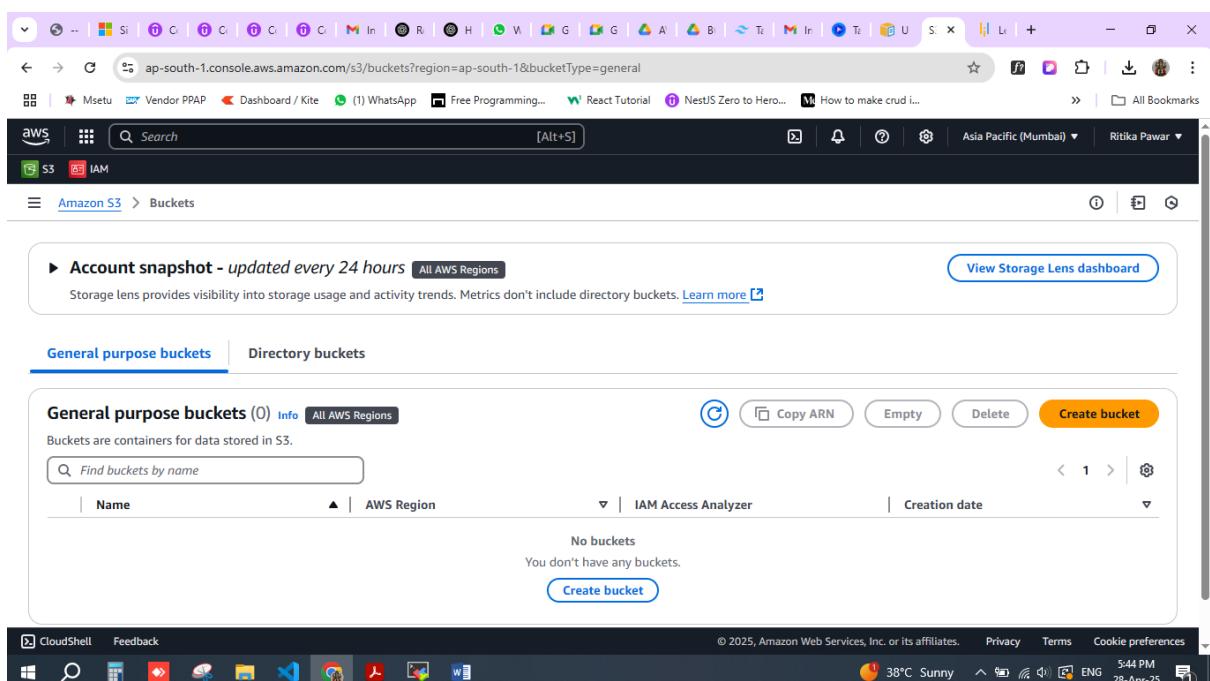
- **Step-by-Step Guide: -**

Step 1: Create S3 Bucket:

1. Open the AWS Management Console



2. Navigate to S3 under the Services section



3. Click on “Create bucket”

The screenshot shows the AWS S3 Buckets page. At the top, there's a navigation bar with various icons and links. Below it, a search bar and a breadcrumb trail ('Amazon S3 > Buckets'). A prominent 'Create bucket' button is located in the top right corner of the main content area.

4. Name the bucket `demo-pet-shop-website` (Bucket names must be globally unique).

The screenshot shows the 'Create bucket' configuration page under 'General configuration'. It includes fields for 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'), 'Bucket type' (set to 'General purpose'), and 'Bucket name' (set to 'demo-pet-shop-website'). There are also sections for 'Copy settings from existing bucket - optional' and 'Choose bucket'. The bottom of the page shows a toolbar and system status information.

Step 2: Upload Website Files:

1. After creating the bucket, go inside it.

The screenshot shows the AWS S3 console interface. The URL in the browser is `ap-south-1.console.aws.amazon.com/s3/buckets/demo-pet-shop-website?region=ap-south-1&bucketType=general&tab=objects`. The page title is "demo-pet-shop-website". The main content area is titled "Objects (0)" and contains a message: "No objects. You don't have any objects in this bucket." Below this message is a blue "Upload" button. The top navigation bar has tabs for "Objects", "Properties", "Permissions", "Metrics", "Management", and "Access Points". The "Objects" tab is currently selected. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time as "28-Apr-25".

2. Click on the Upload button.

This screenshot is identical to the one above, showing the AWS S3 console with an empty bucket. The difference is that the "Upload" button at the bottom of the "Objects (0)" section is now highlighted with a red box, indicating it is the target for the user's click.

3. Upload all your website's static files — like `index.html`, `styles.css`, `script.js`, images, etc.

The screenshot shows the AWS S3 console interface. At the top, there is a green success message: "Upload succeeded" with a note: "For more information, see the Files and folders table." Below this, a section titled "Upload: status" displays the following information:

- Destination:** s3://demo-pet-shop-website
- Succeeded:** 141 files, 1.7 MB (100.00%)
- Failed:** 0 files, 0 B (0%)

Below the status summary, there are two tabs: "Files and folders" (which is selected) and "Configuration". Under the "Files and folders" tab, it says "Files and folders (141 total, 1.7 MB)".

The screenshot shows the AWS S3 console interface, specifically the "Objects" view for the "demo-pet-shop-website" bucket. The top navigation bar shows the path: "Amazon S3 > Buckets > demo-pet-shop-website".

The main area displays a table of objects:

Name	Type	Last modified	Size	Storage class
about.html	html	April 28, 2025, 23:03:42 (UTC+05:30)	20.3 KB	Standard
blog.html	html	April 28, 2025, 23:03:42 (UTC+05:30)	24.8 KB	Standard
contact.html	html	April 28, 2025, 23:03:42 (UTC+05:30)	14.1 KB	Standard
css/	Folder	-	-	-
detail.html	html	April 28, 2025, 23:03:43 (UTC+05:30)	22.1 KB	Standard

At the bottom of the page, there is a footer with standard AWS links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences. The date and time shown are 11:04 PM 28-Apr-25.

Step 3: Enable Bucket Versioning:

1. Go to the Properties tab of the bucket.

The screenshot shows the AWS S3 Bucket Properties page for the bucket 'demo-pet-shop-website'. The 'Properties' tab is selected. In the 'Bucket overview' section, it shows the AWS Region as 'Asia Pacific (Mumbai) ap-south-1', the Amazon Resource Name (ARN) as 'arn:aws:s3:::demo-pet-shop-website', and the Creation date as 'April 28, 2025, 22:58:32 (UTC+05:30)'. The 'Bucket Versioning' section is visible below, showing that versioning is disabled. The browser's address bar shows the URL 'ap-south-1.console.aws.amazon.com/s3/buckets/demo-pet-shop-website?region=ap-south-1&bucketType=general&tab=properties'.

2. Scroll to the Bucket Versioning section.

The screenshot shows the same AWS S3 Bucket Properties page as before, but the view is focused on the 'Bucket Versioning' section. This section contains a detailed description of what bucket versioning is and how it can be used. It also includes a note about multi-factor authentication (MFA) delete. The 'Edit' button is located at the top right of this section. The browser's address bar shows the URL 'ap-south-1.console.aws.amazon.com/s3/buckets/demo-pet-shop-website?region=ap-south-1&bucketType=general&tab=properties'.

3. Click Edit and enable versioning.

The screenshot shows the 'Edit Bucket Versioning' page for the 'demo-pet-shop-website' bucket. The 'Bucket Versioning' section is open, showing two options: 'Suspend' (radio button is empty) and 'Enable' (radio button is checked). A note below states: 'After enabling Bucket Versioning, you might need to update your lifecycle rules to manage previous versions of objects.' Under 'Multi-factor authentication (MFA) delete', it says 'Disabled'. At the bottom right are 'Cancel' and 'Save changes' buttons.

The screenshot shows the 'Properties' tab of the 'demo-pet-shop-website' bucket's properties page. A green success message at the top states: 'Successfully edited Bucket Versioning' and 'To transition, archive, or delete older object versions, configure lifecycle rules for this bucket.' Below this, the 'Bucket overview' section shows the AWS Region as 'Asia Pacific (Mumbai) ap-south-1', the ARN as 'arn:aws:s3:::demo-pet-shop-website', and the Creation date as 'April 28, 2025, 22:58:32 (UTC+05:30)'. The 'Bucket Versioning' section shows 'Enabled'. At the bottom right are 'Edit' and 'Cancel' buttons.

Step 4: Enable Static Web Hosting:

1. In the Properties tab, scroll to Static website hosting.

The screenshot shows the AWS S3 Bucket Properties page for the bucket 'demo-pet-shop-website'. The 'Requester pays' section is visible, showing that it is currently disabled. The 'Static website hosting' section is also visible, showing that it is currently disabled. A blue callout box highlights the 'Create Amplify app' button.

2. Click Edit, then:

- ✓ Select Enable.
- ✓ Set the Index document to 'index.html'.
- ✓ (Optional) Set an error document like 'error.html'.

The screenshot shows the 'Edit static website hosting' configuration page for the 'demo-pet-shop-website' bucket. The 'Static website hosting' section is set to 'Enable'. The 'Hosting type' section is set to 'Host a static website'. The 'Index document' field is set to 'index.html'. A blue callout box highlights a note about making content publicly readable via S3 Block Public Access settings.

Step 5: Unblock Public Access:

1. Go to the Permissions tab

The screenshot shows the AWS S3 console with the 'demo-pet-shop-website' bucket selected. The 'Permissions' tab is active in the navigation bar. Under the 'Permissions overview' section, there is a note about access findings and a link to view the analyzer for the specific region. Below this, the 'Block public access (bucket settings)' section is visible, showing that 'Block all public access' is turned 'On'. The browser's address bar shows the URL for the bucket's properties page.

2. Click on Block public access (bucket settings).

The screenshot shows the 'Edit Block public access (bucket settings)' page for the 'demo-pet-shop-website' bucket. The 'Block public access (bucket settings)' section is expanded, showing five options under 'Block all public access': 'Block public access to buckets and objects granted through new access control lists (ACLS)', 'Block public access to buckets and objects granted through any access control lists (ACLS)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies'. Each option has a brief description below it. The browser's address bar shows the URL for the edit page.

3. Click Edit, uncheck all options to disable blocking public access.

The screenshot shows the AWS S3 console with the path: Amazon S3 > Buckets > demo-pet-shop-website > Edit Block public access (bucket settings). The 'Block all public access' checkbox is checked. Below it, there are five other checkboxes for specific access control lists (ACLs) and policies, all of which are also checked. A note at the top states: "Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases." At the bottom right are 'Cancel' and 'Save changes' buttons.

4. Confirm the warning and save.

The screenshot shows the same AWS S3 console page as above, but now a modal dialog box is open over the page. The dialog title is 'Edit Block public access (bucket settings)'. It contains a warning message: '⚠️ Updating the Block Public Access settings for this bucket will affect this bucket and all objects within. This may result in some objects becoming public.' Below this is a text input field with the placeholder 'To confirm the settings, enter confirm in the field.' and a 'confirm' button. At the bottom right of the dialog are 'Cancel' and 'Confirm' buttons. The background of the page shows the same configuration options as the previous screenshot, with the 'Block all public access' checkbox unchecked. The status bar at the bottom indicates it's 11:14 PM on April 28, 2025, in Mumbai, India.

Step 6: Change Object Ownership & Enable Ownership:

1. Still under Permissions, go to Object Ownership.

The screenshot shows the AWS S3 Bucket Permissions page for the bucket 'demo-pet-shop-website'. Under 'Object Ownership', it is set to 'Bucket owner enforced'. A note states: 'Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.' Under 'Access control list (ACL)', it says 'Grant basic read/write permissions to other AWS accounts.' A note indicates: 'This bucket has the bucket owner enforced setting applied for Object Ownership. When bucket owner enforced is applied, use bucket policies to control access.' The 'Grantee' section shows 'Bucket owner (your AWS account)' with 'List, Write' and 'Read, Write' permissions. The browser status bar at the bottom shows 'CloudShell Feedback' and system information like '29°C Clear' and '11:15 PM 28-Apr-25'.

The screenshot shows the 'Edit Object Ownership' page. It displays two options for 'Object Ownership': 'ACLs disabled (recommended)' and 'ACLs enabled'. The 'ACLs disabled' option is selected, with a note: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' The 'ACLs enabled' option is also shown with a note: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.' At the bottom right are 'Cancel' and 'Save changes' buttons. The browser status bar at the bottom shows 'CloudShell Feedback' and system information like '29°C Clear' and '11:16 PM 28-Apr-25'.

2. Click Edit and select ACLs enabled (bucket owner preferred or object writer preferred).

S | C | C | C | M | R | H | V | C | N | A | B | T | M | L | E | L | F | +

ap-south-1.console.aws.amazon.com/s3/bucket/demo-pet-shop-website/property/oo/edit?region=ap-south-1&bucketType=general

Msetu | Vendor PPAP | Dashboard / Kite | (1) WhatsApp | Free Programming... | React Tutorial | NestJS Zero to Hero... | How to make crud i...

All Bookmarks

aws IAM

Search [Alt+S]

Amazon S3 > Buckets > demo-pet-shop-website > Edit Object Ownership

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

⚠ Enabling ACLs turns off the bucket owner enforced setting for Object Ownership
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.
 I acknowledge that ACLs will be restored.

Object Ownership

Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer
The object writer remains the object owner.

ⓘ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

[Cancel](#) [Save changes](#)

Successfully edited Object Ownership.

demo-pet-shop-website Info

Objects | Properties | **Permissions** | Metrics | Management | Access Points

Permissions overview

Access finding
Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#).

[View analyzer for ap-south-1](#)

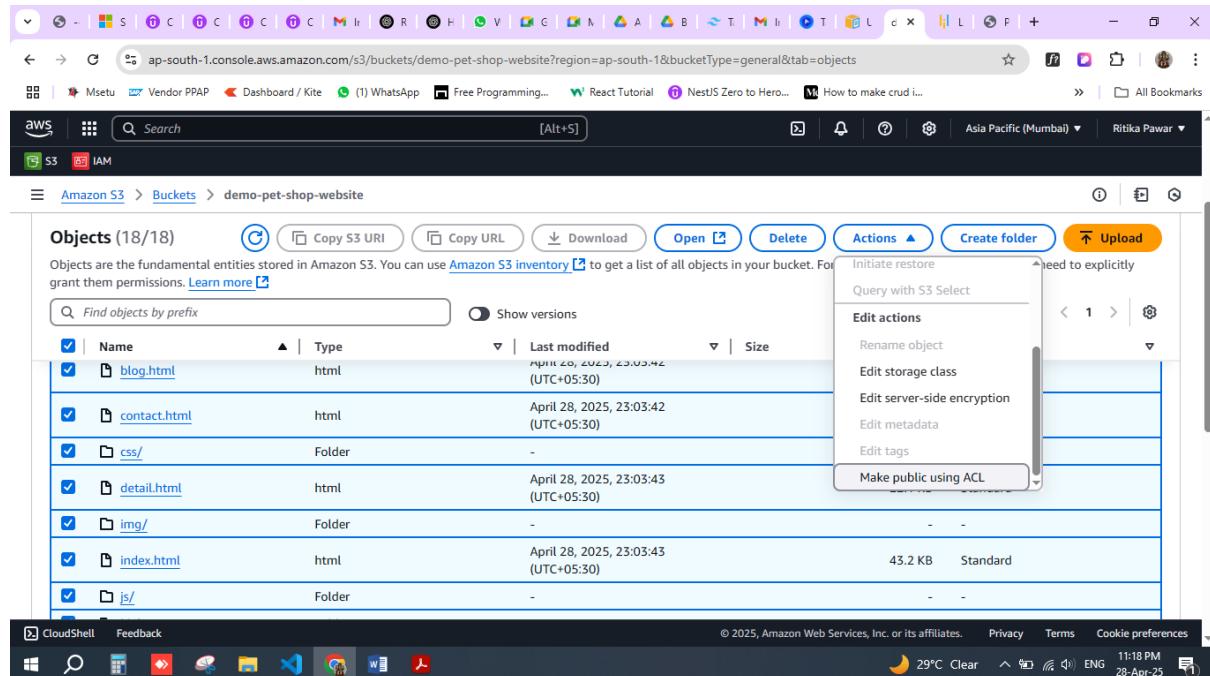
Block public access (bucket settings)

[Edit](#)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

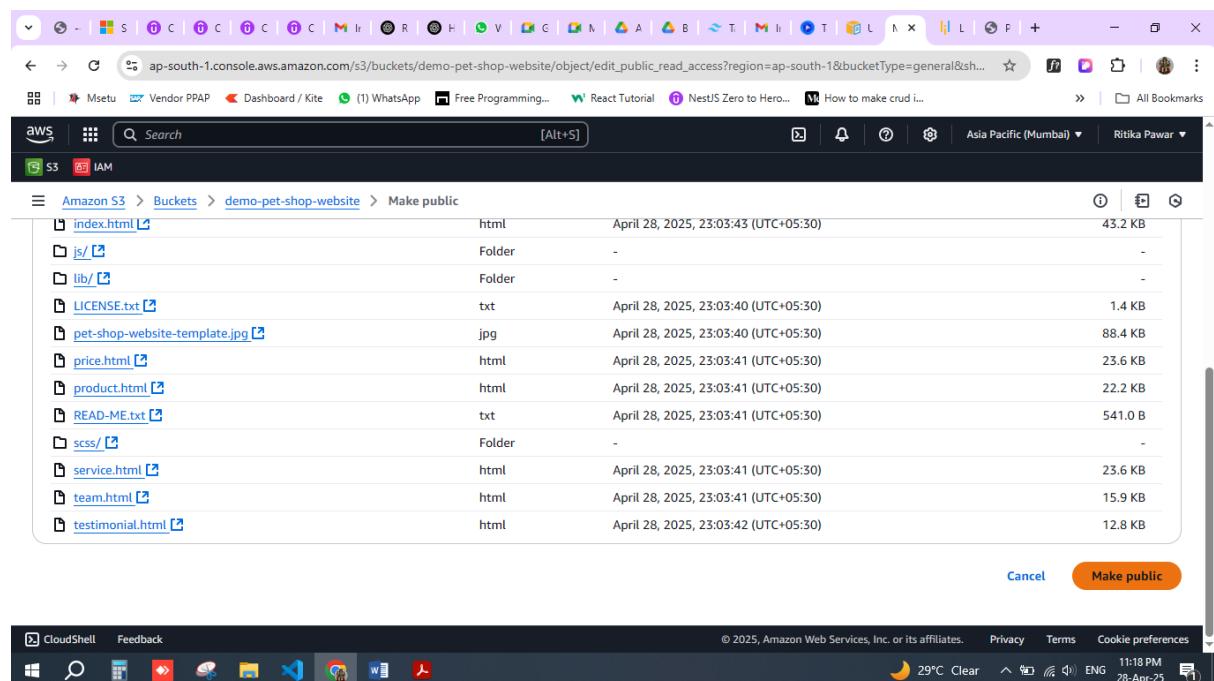
Step 7: Make Uploaded File Public Using ACL:

1. Go to the Objects, select all files and click Actions → Make public.



The screenshot shows the AWS S3 console interface. In the center, there is a table titled 'Objects (18/18)' listing various files and folders. Several files are selected, indicated by a checkmark icon next to them. To the right of the table, a context menu is open under the 'Actions' heading. The 'Edit actions' section is visible, and at the bottom of this list, the option 'Make public using ACL' is highlighted with a yellow background and a black border.

2. Confirm the action. Each object will now have a public URL accessible via browser.



The screenshot shows a confirmation dialog box titled 'Amazon S3 > Buckets > demo-pet-shop-website > Make public'. Inside the dialog, a table lists the selected objects: index.html, js/, lib/, LICENSE.txt, pet-shop-website-template.jpg, price.html, product.html, READ-ME.txt, scss/, service.html, team.html, and testimonial.html. Each row includes the file name, type, last modified date, and size. At the bottom of the dialog, there are two buttons: 'Cancel' on the left and 'Make public' on the right, which is highlighted with a yellow background and a black border.

The screenshot shows the AWS S3 console with a green success message: "Successfully edited public access". It displays two sections: "Successfully edited public access" (141 objects, 1.7 MB) and "Failed to edit public access" (0 objects). Below these are tabs for "Failed to edit public access" and "Configuration". The status bar at the bottom indicates "11:19 PM 28-Apr-25".

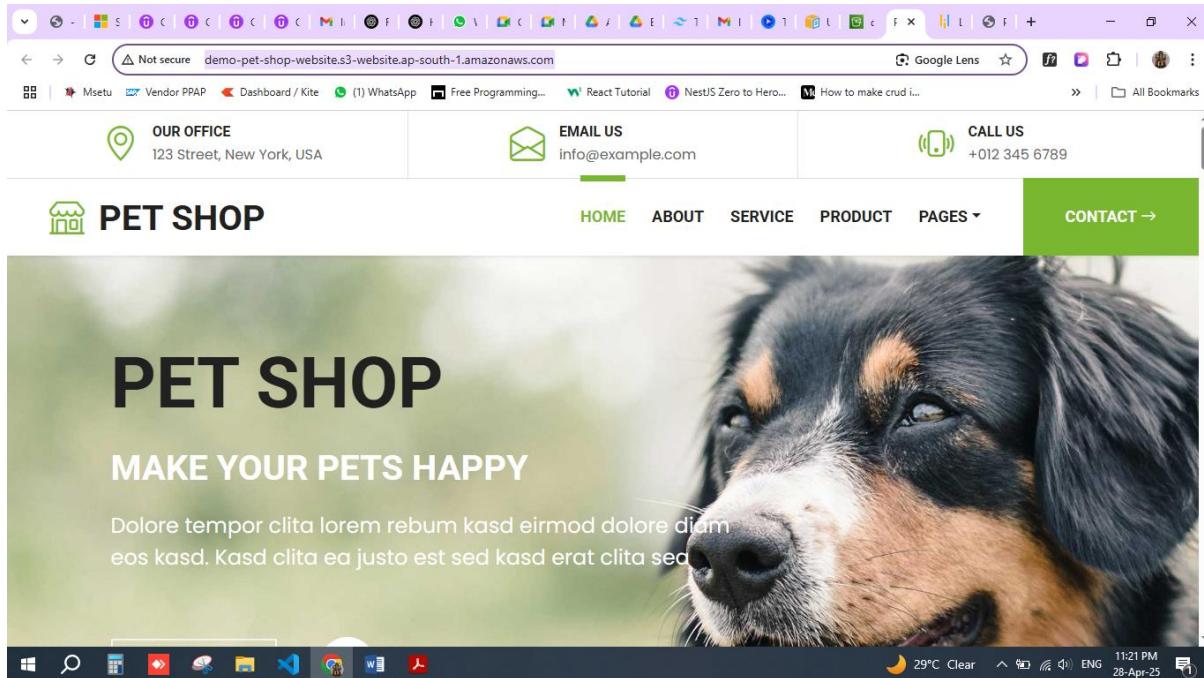
Step 8: Access the Static Website:

After enabling hosting and permissions, your site can be accessed at:
<http://demo-pet-shop-website.s3-website.ap-south-1.amazonaws.com/>

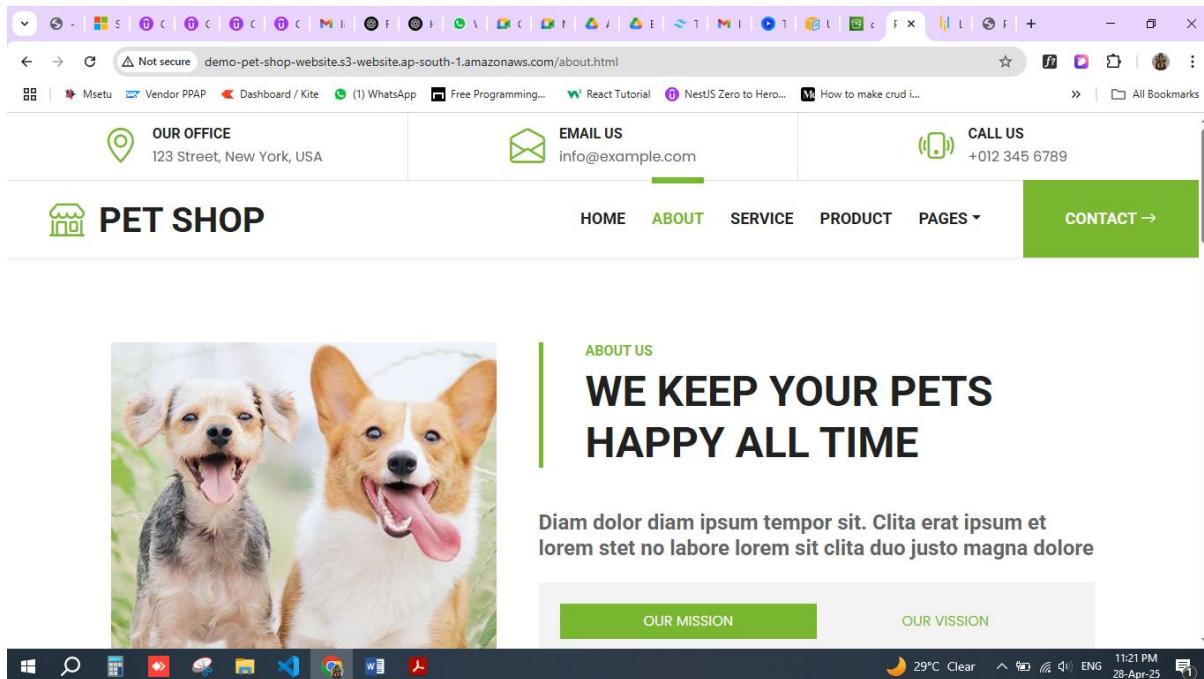
The screenshot shows the AWS S3 console under the "Properties" tab for the "demo-pet-shop-website" bucket. It highlights the "Static website hosting" section, which is enabled. It includes a note about using AWS Amplify Hosting and a "Create Amplify app" button. Other sections shown include "S3 static website hosting" (Enabled), "Hosting type" (Bucket hosting), and "Bucket website endpoint" (Endpoint: http://demo-pet-shop-website.s3-website.ap-south-1.amazonaws.com). The status bar at the bottom indicates "11:20 PM 28-Apr-25".

- **Web Pages: -**

1. Home Page



2. About Page



3. Service Page

The screenshot shows a web browser window with the URL <https://demo-pet-shop-website.s3-website.ap-south-1.amazonaws.com/service.html>. The page title is "PET SHOP". The navigation bar includes links for HOME, ABOUT, SERVICE (which is highlighted in green), PRODUCT, PAGES, and CONTACT. A sidebar on the left is titled "SERVICES" and contains the heading "OUR EXCELLENT PET CARE SERVICES". Below this, there are two service cards: "PET BOARDING" (with a house icon) and "PET FEEDING" (with a bone icon). Both cards have placeholder text: "Kasd dolor no lorem sit tempor at justo rebum rebum stet justo elitr dolor amet sit" and "Kasd dolor no lorem sit tempor at justo rebum rebum stet justo elitr dolor amet sit". Each card has a "READ MORE →" link. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time as 28-Apr-25.

4. Products for pets

The screenshot shows a web browser window with the URL <https://demo-pet-shop-website.s3-website.ap-south-1.amazonaws.com/product.html>. The page title is "PET SHOP". The navigation bar includes links for HOME, ABOUT, SERVICE, PRODUCT (which is highlighted in green), PAGES, and CONTACT. A sidebar on the left is titled "PRODUCTS" and contains the heading "PRODUCTS FOR YOUR BEST FRIENDS". Below this, there are four product cards, each showing a bag of pet food: "QUALITY PET FOODS" (Natural Premium Bird Food), "QUALITY PET FOODS" (Nature's Lickin' Cat Food), "QUALITY PET FOODS" (Whole Earth Farm Natural Dog Food - Grain Free Recipe), and "QUALITY PET FOODS" (Whole Earth Farm Natural Dog Food - Grain Free Recipe). The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time as 28-Apr-25.

5. Pricing Plan

The screenshot shows a Windows desktop environment with a browser window open to the 'PRICING PLAN' section of the Pet Shop website. The page features three service plans: 'BASIC', 'STANDARD', and 'EXTENDED'. Each plan is described as 'The Best Choice' and includes a green price box and a list of features with checkmarks.

Plan	Price	Features
BASIC	\$49/Mo	HTML5 & CSS3
STANDARD	\$99/Mo	HTML5 & CSS3 Bootstrap v5 Responsive Layout
EXTENDED	\$149/Mo	HTML5 & CSS3

The browser's address bar shows the URL: demo-pet-shop-website.s3-website.ap-south-1.amazonaws.com/price.html. The taskbar at the bottom displays various application icons, and the system tray shows the date and time as 28-Apr-25.

6. Contact

The screenshot shows a Windows desktop environment with a browser window open to the 'CONTACT US' section of the Pet Shop website. The page has a form for users to fill out and a sidebar with contact information and a map.

CONTACT US

PLEASE FEEL FREE TO CONTACT US

Form fields:

- Your Name
- Your Email
- Subject
- Message

Contact Information:

- OUR OFFICE**: 123 Street, New York, USA
- EMAIL US**: info@example.com
- CALL US**: +012 345 6789

A small map of the New York area is displayed, showing locations like Montreal, New York, and New Jersey.

The browser's address bar shows the URL: demo-pet-shop-website.s3-website.ap-south-1.amazonaws.com/contact.html. The taskbar at the bottom displays various application icons, and the system tray shows the date and time as 28-Apr-25.

- **Summary: -**

This project demonstrates how to deploy a fully functional Pet Shop static website using Amazon S3. It covers the entire lifecycle of static website hosting—from bucket creation and file uploads to permission management and access logging. The process ensures that the frontend website is live, publicly accessible, version-controlled, and monitored.

- **Optical Enhancement: -**

To enhance the PetShop Static Website further, several advanced features can be integrated. Adding a custom domain using Amazon Route 53, along with CloudFront for HTTPS and global content delivery, gives the project a professional edge. Converting the site into a Progressive Web App (PWA) with offline support, service workers, and an installable interface improves usability across devices. Incorporating a frontend framework like React.js or Vue.js can add dynamic product interactions, while CI/CD pipelines using GitHub Actions or AWS Code Pipeline can automate deployments.