

Project Title :- Hosting a Carvilla Website Using AWS S3 Services

- **Hosting a Carvilla Website Using AWS S3 Services:**

This document provides a step-by-step guide to host a Carvilla 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 Car 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., carvilla-log) for analytics and auditing.

7. (Optional) CI/CD Deployment:

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

• Technology Used:-

The Carvilla 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 Carvilla 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 Carvilla) showcases your frontend and cloud skills to potential employers or clients.

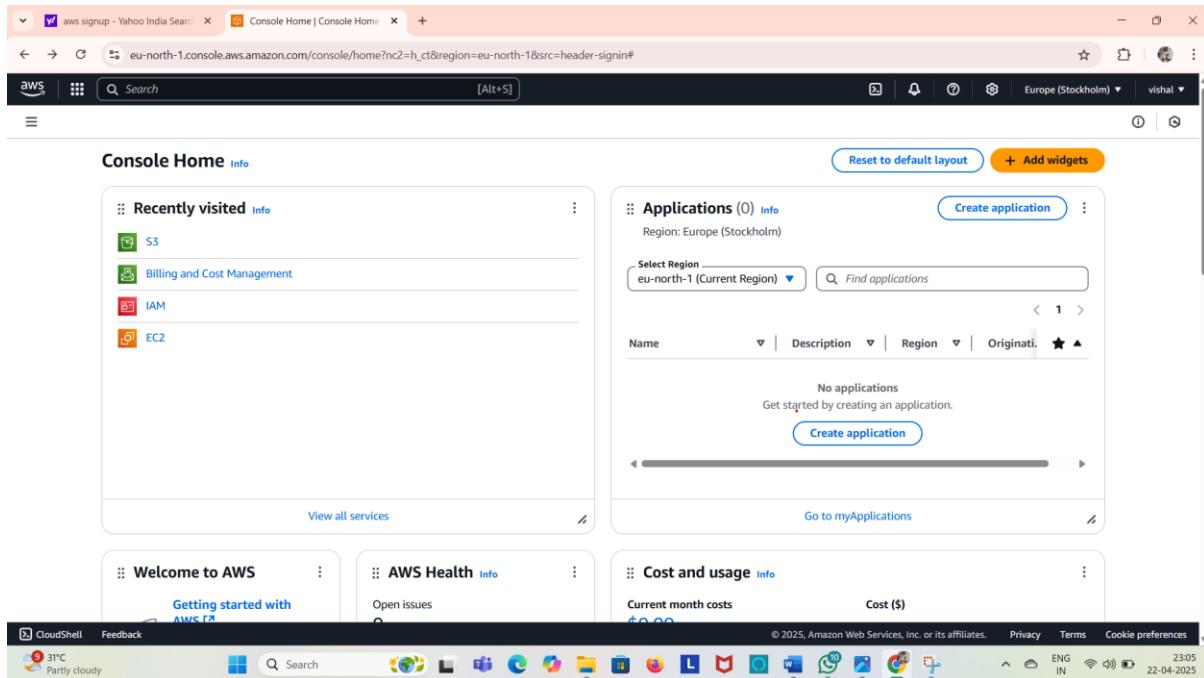
- 3. Foundation for Future Projects:**

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.

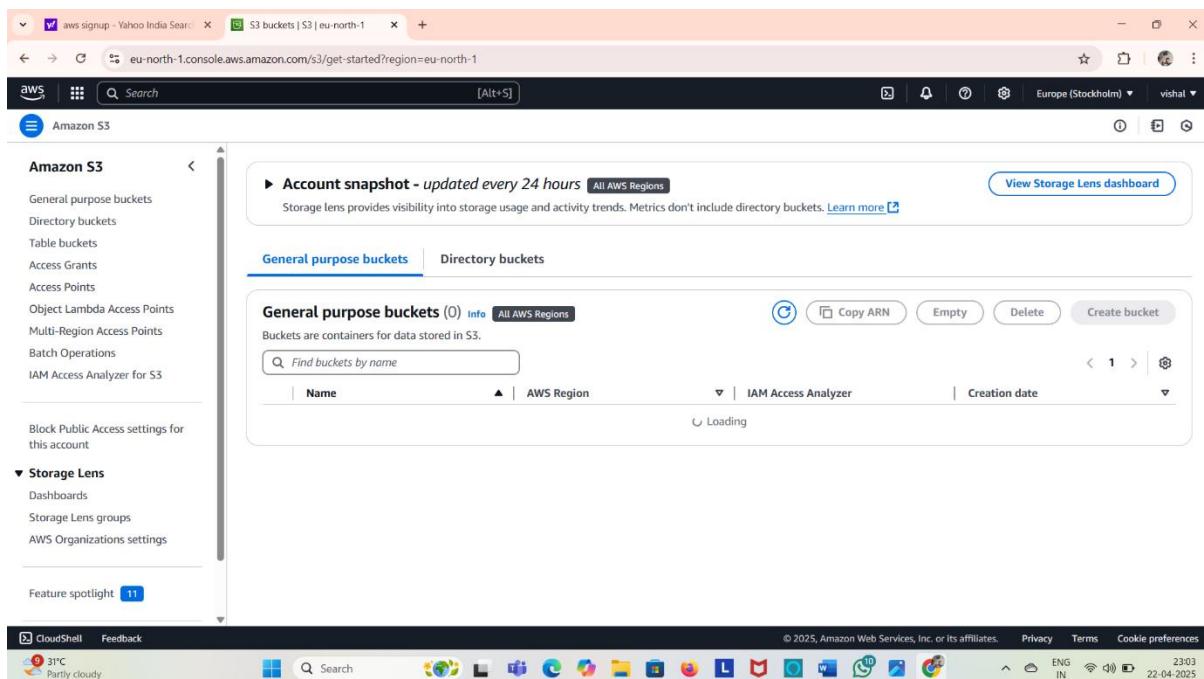
- **Step-by-Step Guide:-**

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 header bar with tabs for "General purpose buckets" and "Directory buckets". Below the header, a table lists six buckets. Each row contains the bucket name, AWS Region, IAM Access Analyzer link, and Creation date. A "Create bucket" button is located at the top right of the table area.

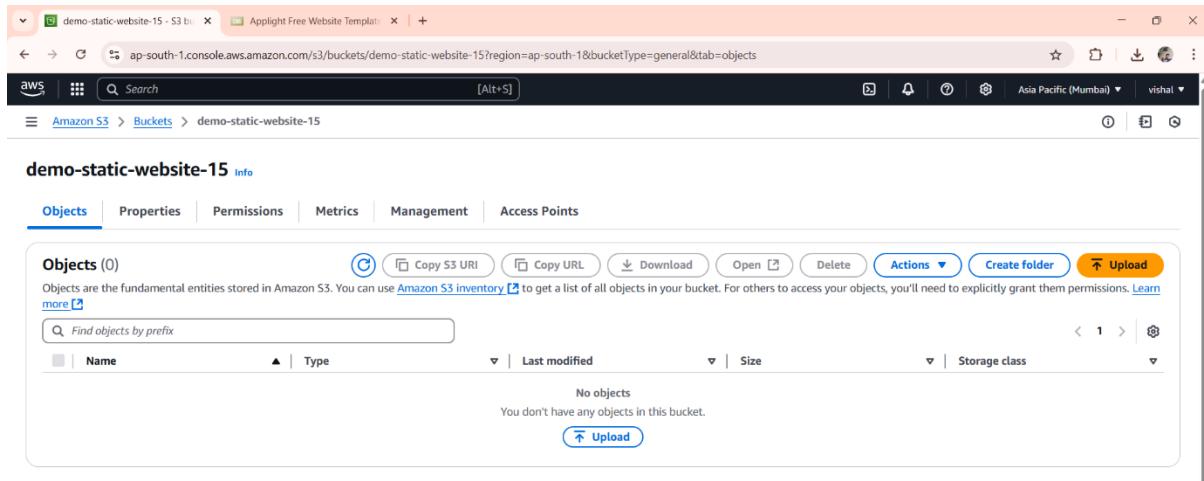
Name	AWS Region	IAM Access Analyzer	Creation date
demo-152002	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 19, 2025, 21:31:30 (UTC+05:30)
demo-8080	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 19, 2025, 22:23:52 (UTC+05:30)
demo-static-website-123	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 22, 2025, 16:53:42 (UTC+05:30)
sarjine-123	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 20, 2025, 17:35:18 (UTC+05:30)
vishu-152002	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 22, 2025, 08:32:55 (UTC+05:30)
vishu-5078	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 19, 2025, 22:11:51 (UTC+05:30)

4. Name the bucket `demo-static-website-15` (Bucket names must be globally unique).

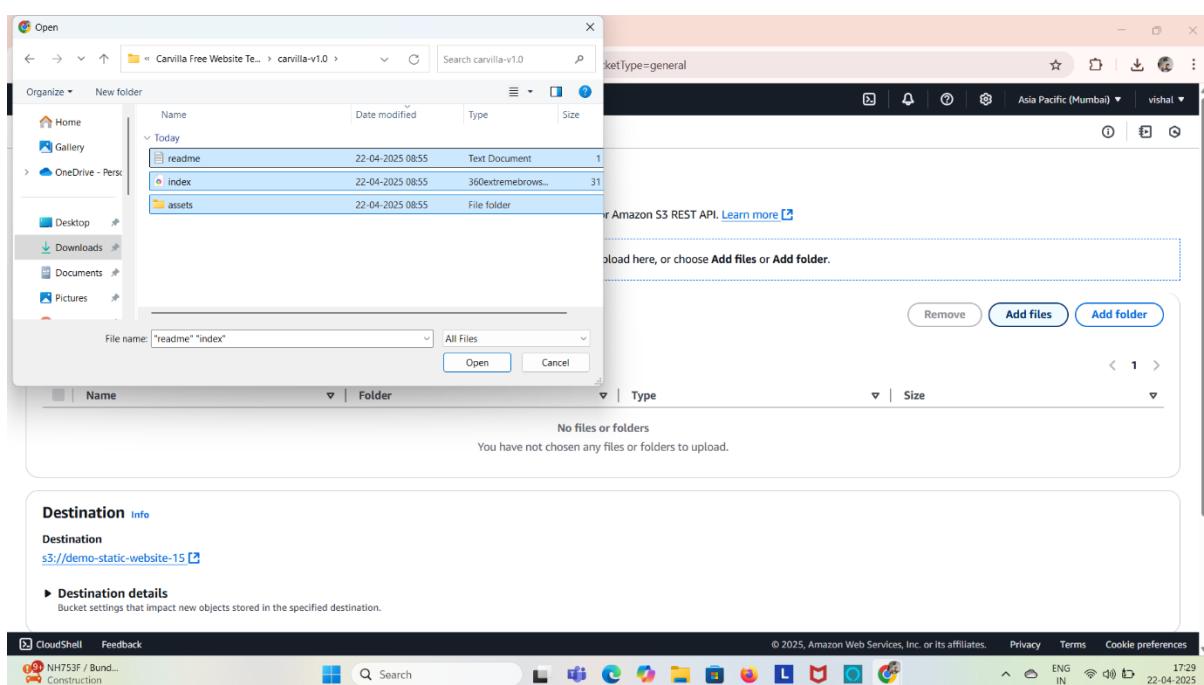
The screenshot shows the "Create bucket" page. It has sections for "General configuration", "Object Ownership", and "Copy settings from existing bucket - optional". In the "General configuration" section, the "Bucket type" is set to "General purpose". The "Bucket name" field contains "demo-static-website-15". The "Object Ownership" section notes that object ownership determines who can specify access to objects. The "Copy settings from existing bucket - optional" section indicates that only the bucket settings in the following configuration are copied, with a "Choose bucket" button and a prefix input field.

Step 2: Upload Website Files:

1. After creating the bucket, go inside it.



2. Click on the Upload button.



- 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's a navigation bar with tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected. Below the navigation bar, a message says '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.' There's also a link to 'Learn more'. A search bar labeled 'Find objects by prefix' is present. A table lists three objects:

Name	Type	Last modified	Size	Storage class
assets/	Folder		-	-
index.html	html	April 22, 2025, 17:30:08 (UTC+05:30)	30.2 KB	Standard
readme.txt	txt	April 22, 2025, 17:30:07 (UTC+05:30)	889.0 B	Standard

At the bottom of the page, there's a footer with links for 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'. It also shows the date and time as '22-04-2025'.

Step 3: Enable Bucket Versioning:

- Go to the Properties tab of the bucket.

The screenshot shows the AWS S3 console interface with the 'Properties' tab selected. The 'Objects' table is identical to the one in the previous screenshot. In the top right corner of the main content area, there's a 'Versioning' section with the status 'Status: Enabled'. Below this, there's a 'Version ID' field containing '1'. The rest of the properties section is mostly empty or contains placeholder text like 'No version history found'.

2. Scroll to the Bucket Versioning section.

The screenshot shows the AWS S3 Bucket Overview page for the bucket 'demo-sap-123'. The 'Bucket Versioning' section is visible, showing that it is currently disabled. There is a link to learn more about Multi-factor authentication (MFA) delete.

3. Click Edit and enable versioning.

The screenshot shows the 'Edit Bucket Versioning' page for the bucket 'demo-static-website-15'. The 'Bucket Versioning' section is shown with the 'Enable' option selected. A note indicates that lifecycle rules may need to be updated after enabling versioning. The 'Save changes' button is visible at the bottom right.

Step 4: Enable Static Web Hosting:

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

The screenshot shows the AWS S3 console interface for a bucket named 'demo-static-website-15'. The 'Properties' tab is selected. In the 'Static website hosting' section, there is a note: 'We recommend using AWS Amplify Hosting for static website hosting. Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more or View your existing Amplify apps.' Below this, there is a button labeled 'Create Amplify app'. The 'S3 static website hosting' status is currently 'Disabled'.

2. Click Edit, then:
 - Select Enable.
 - Set the Index document to `index.html`.
 - (Optional) Set an error document like `error.html`.

Edit static website hosting [Info](#)

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
 Disable
 Enable

Hosting type
 Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Index document
Specify the home or default page of the website.
index.html

Error document - optional
This is returned when an error occurs.
error.html

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 39°C Partly sunny ENG IN 17:32 22-04-2025

Step 5: Unblock Public Access:

1. Go to the Permissions tab

demo-static-website-15 [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)
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](#)

Block all public access
 On
► Individual Block Public Access settings for this bucket

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](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 39°C Partly sunny ENG IN 17:33 22-04-2025

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

Block public access (bucket settings)

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

Block public and cross-account access to buckets and objects through any public bucket or access point policies

Cancel Save changes

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

Block public access (bucket settings)

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

Block public and cross-account access to buckets and objects through any public bucket or access point policies

Cancel Save changes

4. Confirm the warning and save.

The screenshot shows the 'Edit Block public access (bucket settings)' page in the AWS S3 console. The 'Block all public access' checkbox is checked. Below it, several other options are listed, each with a descriptive subtitle and a brief explanation. At the bottom right of the page, there are 'Cancel' and 'Save changes' buttons.

Step 6: Change Object Ownership & Enable Ownership:

1. Still under Permissions, go to Object Ownership.

The screenshot shows the 'Edit Object Ownership' page in the AWS S3 console. It features two radio button options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. Below these options, the 'Object Ownership' section states 'Bucket owner enforced'. At the bottom right, there are 'Cancel' and 'Save changes' buttons.

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

The screenshot shows the 'Edit Object Ownership' page in the AWS S3 console. The 'Object Ownership' section has two options: 'ACLs disabled (recommended)' and 'ACLs enabled'. 'ACLs enabled' is selected, indicated by a blue border around the radio button and the explanatory text below it. A note below 'ACLs disabled' states: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' A note below 'ACLs enabled' states: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.' Below these notes are two informational boxes: one about enabling ACLs and another about disabling them.

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](#)

Step 7: Make Uploaded File Public Using ACL:

- Go to the Objects, Select all files and click Actions → Make public.

The screenshot shows the 'Objects' tab in the AWS S3 console for the 'demo-static-website-15' bucket. The 'Actions' menu is open over a selection of three objects: 'assets/' (Folder), 'index.html' (html), and 'readme.txt' (txt). The 'Actions' menu includes options like 'Upload', 'Download', 'Open', 'Delete', 'Create folder', and several detailed actions. The 'Make public using ACL' option is highlighted with a blue border.

demo-static-website-15

Objects (3/3)

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, [more](#)

Name ▲ | Type ▼ | Last modified ▼ | Size ▼

Name	Type	Last modified	Size
assets/	Folder	-	
index.html	html	April 22, 2025, 17:30:08 (UTC+05:30)	
readme.txt	txt	April 22, 2025, 17:30:07 (UTC+05:30)	

Actions

- Download as
- Share with a presigned URL
- Calculate total size
- Copy
- Move
- Initiate restore
- Query with S3 Select
- Edit actions
- Rename object
- Edit storage class
- Edit server-side encryption
- Edit metadata
- Edit tags
- Make public using ACL**

2. Confirm the action.

Each object will now have a public URL accessible via browser.

The screenshot shows the AWS S3 console with a green success message: "Successfully edited public access". It lists 60 objects, 3.3 MB in size. Below, the "Failed to edit public access" section shows 0 objects. The status bar at the bottom indicates "Thunderstorm w... In effect".

Step 8: Access the Static Website:

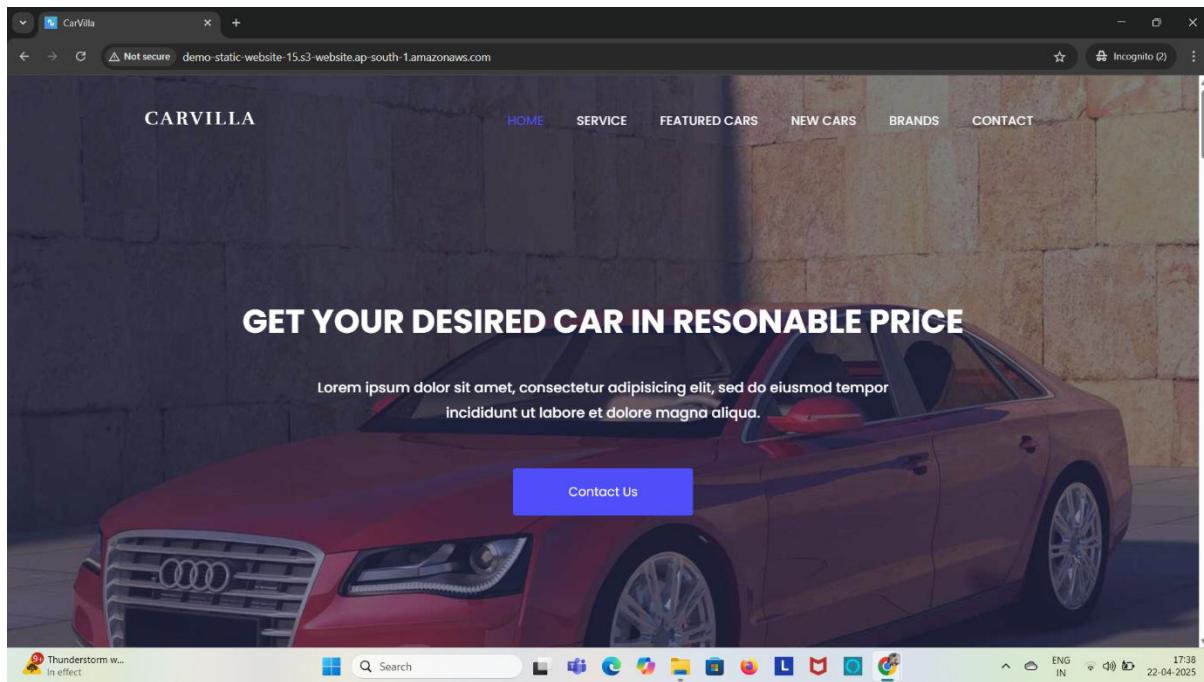
After enabling hosting and permissions, your site can be accessed at:

<http://demo-static-website-15.s3-website.ap-south-1.amazonaws.com>

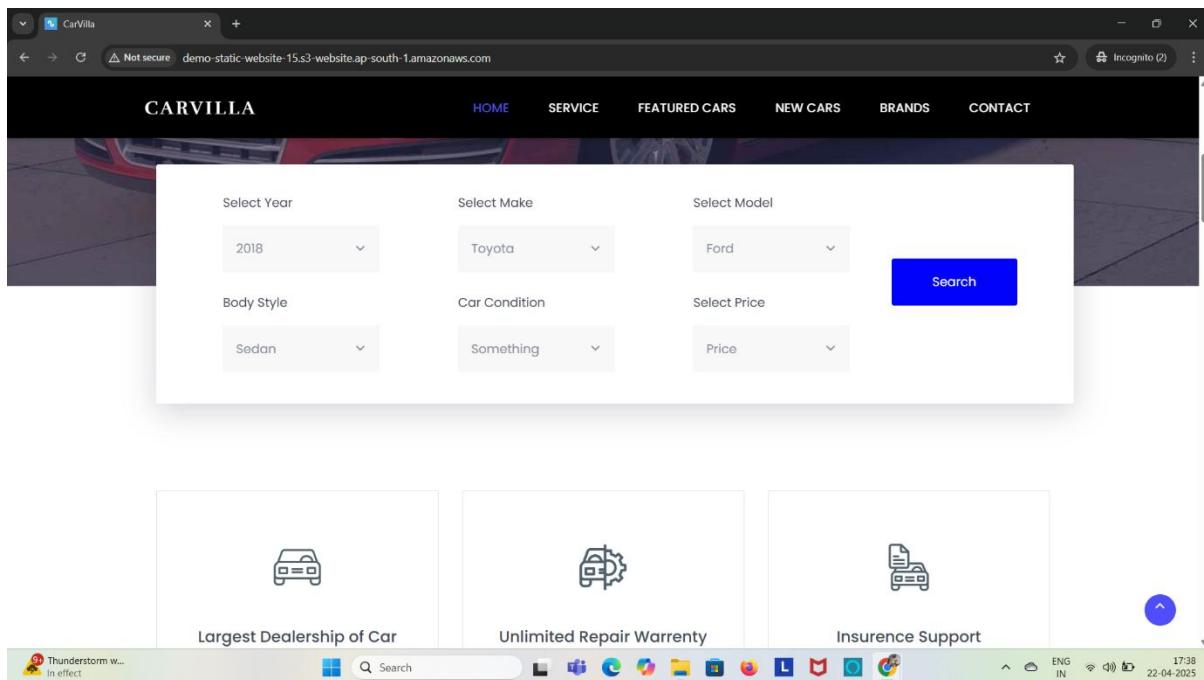
The screenshot shows the "Static website hosting" configuration for the bucket. It recommends AWS Amplify Hosting and provides a link to "Create Amplify app". The "Bucket website endpoint copied" message includes a link to the public URL: <http://demo-static-website-15.s3-website.ap-south-1.amazonaws.com>. The status bar at the bottom indicates "Thunderstorm w... In effect".

- **Web Pages:-**

1. Home Page



2. Service Page



3. Featured Cars

The screenshot shows a web browser window for the 'CarVilla' website. The header includes a 'Not secure' warning, the URL 'demo-static-website-15.s3-website.ap-south-1.amazonaws.com/#', and a 'CarVilla' logo. A navigation bar with links to HOME, SERVICE, FEATURED CARS (which is highlighted in blue), NEW CARS, BRANDS, and CONTACT. Below the navigation is a sub-header 'Checkout the Featured Cars' and a main title 'Featured Cars'. There are four car cards displayed in a row:

- BMW 6-Series Gran Coupe** - Model: 2017 | 3100 Mi | 240HP | Automatic | \$89,395 | Description: Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
- Chevrolet Camaro WMV20** - Model: 2017 | 3100 Mi | 240HP | Automatic | \$69,575 | Description: Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
- Lamborghini V520** - Model: 2017 | 3100 Mi | 240HP | Automatic | \$125,250 | Description: Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
- Audi A3 Sedan** - Model: 2017 | 3100 Mi | 240HP | Automatic | \$95,500 | Description: Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

The status bar at the bottom shows a weather icon for 'Thunderstorm w...', 'In effect', a search bar, taskbar icons, and system status like 'ENG IN', '17:39', and '22-04-2025'.

4. New Cars

The screenshot shows a web browser window for the 'CarVilla' website. The header includes a 'Not secure' warning, the URL 'demo-static-website-15.s3-website.ap-south-1.amazonaws.com/#', and a 'CarVilla' logo. A navigation bar with links to HOME, SERVICE, FEATURED CARS (highlighted in blue), NEW CARS, BRANDS, and CONTACT. Below the navigation is a sub-header 'Checkout the Latest Cars' and a main title 'Newest Cars'. A large image of a red Ferrari 488 Superfast is displayed. To its right is a detailed description and a 'View Details' button.

Ferrari 488 Superfast

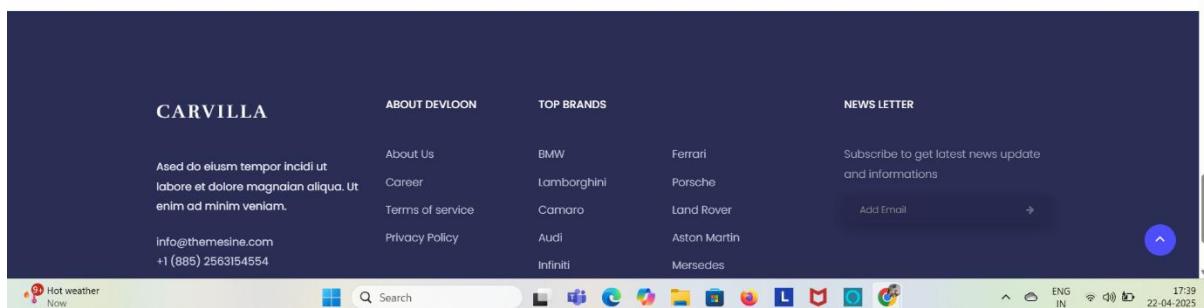
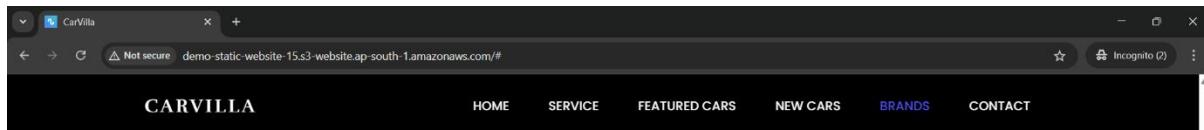
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.

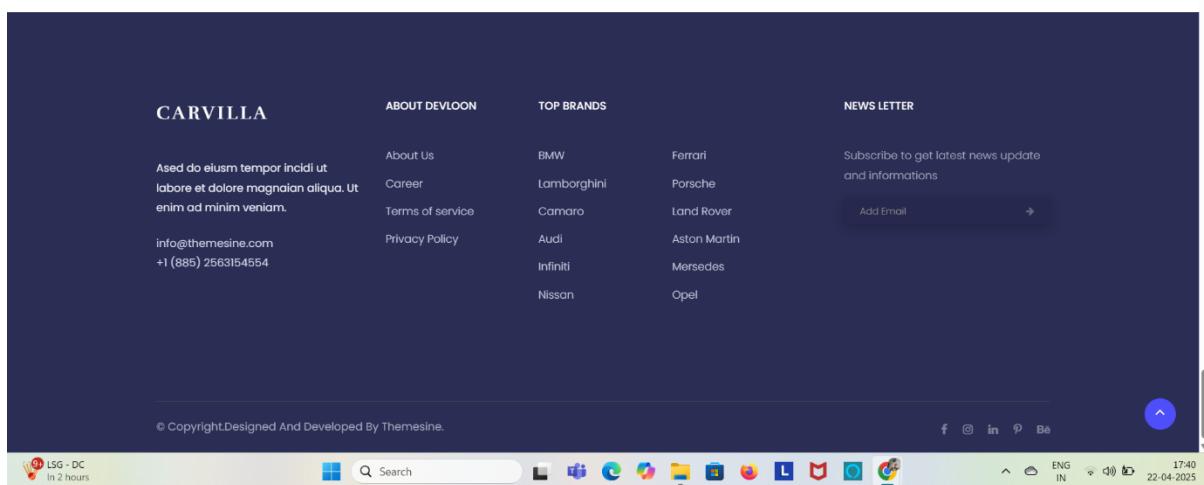
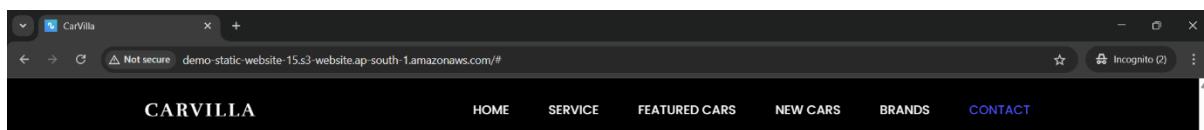
[View Details](#)

The status bar at the bottom shows a weather icon for 'Hot weather Now', a search bar, taskbar icons, and system status like 'ENG IN', '17:39', and '22-04-2025'.

5. Brands



6. Contact



- **Summary:-**

This project demonstrates how to deploy a fully functional static Carvilla 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 Carvilla 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 CodePipeline can automate deployments.

