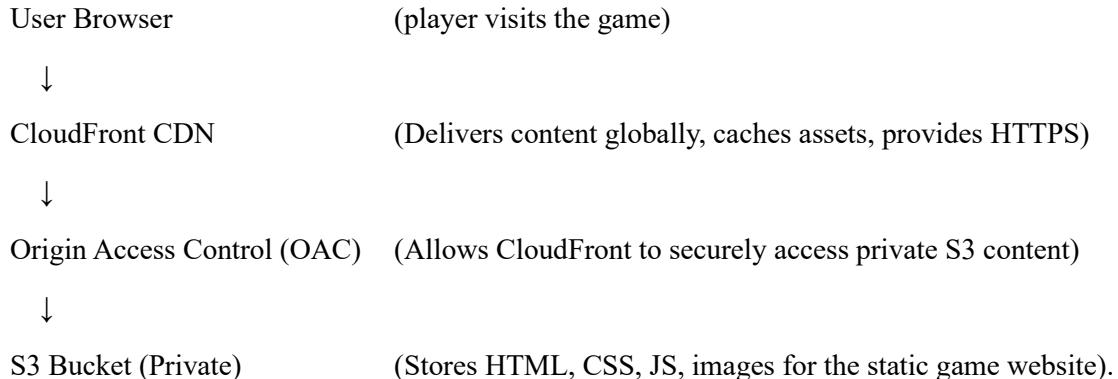


Deploy a Static Website on Amazon S3 and CloudFront

ARCHITECTURAL DIAGRAM:



DESCRIPTION:

- A fully deployed static website hosted on Amazon S3 and served securely through CloudFront with HTTPS.
- Includes optional custom domain setup using Route 53 and SSL certificates via AWS Certificate Manager.

The image shows two screenshots of the AWS Management Console. The top screenshot is for the Amazon S3 service, displaying the storage interface with options for creating a new bucket or viewing pricing details. The bottom screenshot is for the Amazon CloudFront service, showing the networking and content delivery interface with options for creating a distribution or viewing free tier details. Both interfaces include navigation menus on the left and descriptive text and buttons on the right.

Here's a clear, complete walkthrough of **all steps involved** in creating and deploying a fully static website on **Amazon S3**, served securely over **CloudFront**, with optional **Route 53 + ACM** for a custom domain.

STEP 1 — CREATE AN S3 BUCKET:

1. Open AWS Console → S3.

2. Click “Create bucket”.

3. Enter a bucket name

Bucket name: **aws-s3-cloudfront-card-game**

4. Uncheck “**Block all public access.**”

5. Click **Create bucket**.

The screenshot shows the 'Create bucket' wizard on the 'General configuration' step. It includes fields for AWS Region (US East (N. Virginia) us-east-1), Bucket type (set to 'General purpose'), Bucket name (aws-s3-cloudfront-card-game), and a 'Copy settings from existing bucket - optional' section.

The screenshot shows the 'Block Public Access settings for this bucket' step. It contains a warning about turning off block all public access, a list of checkboxes for different access control options, and a note about turning off block all public access resulting in a public bucket. A checkbox for acknowledging the risk is checked.

The screenshot shows the 'Buckets' list page. It displays the newly created bucket 'aws-s3-cloudfront-card-game' under the 'General purpose buckets' section. Other sections like 'Account snapshot' and 'External access summary' are also visible.

STEP 2 — UPLOAD WEBSITE FILES:

1. Open the bucket. (bucket - **aws-s3-cloudfront-card-game**)
2. Click **Upload** → **Add files**.
3. Upload:
 - o index.html
 - o CSS/Javascript/images folders
4. Click **Upload**.

The screenshot shows the 'Upload' step in the AWS S3 console. At the top, there's a breadcrumb navigation: 'Amazon S3 > Buckets > aws-s3-cloudfront-card-game > Upload'. Below it is a 'Upload' section with a 'Info' link. A note says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more.' A large blue dashed box is labeled 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a 'Files and folders (0)' section with a table header: 'Name' (with a checkbox), 'Folder', 'Type', and 'Size'. A message at the bottom says 'No files or folders' and 'You have not chosen any files or folders to upload.'

This screenshot shows the same 'Upload' step after files have been added. The 'Files and folders (0)' section now lists four items: 'style.css' (text/css, 1.9 KB), 'icon.png' (image/png, 23.0 KB), 'index.html' (text/html, 758.0 B), and 'script.js' (text/javascript, 2.0 KB). The rest of the interface remains the same, including the 'Destination' section which points to 's3://aws-s3-cloudfront-card-game'.

This screenshot shows the 'Objects' section of the AWS S3 console for the 'aws-s3-cloudfront-card-game' bucket. At the top, there's a breadcrumb navigation: 'Amazon S3 > Buckets > aws-s3-cloudfront-card-game'. Below it is a 'aws-s3-cloudfront-card-game' section with an 'Info' link. A table header includes 'Actions', 'Create folder', and an 'Upload' button. The main area shows a table of objects with columns: 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The objects listed are 'icon.png' (png, November 25, 2025, 08:33:52, 23.0 KB, Standard), 'index.html' (html, November 25, 2025, 08:33:53, 758.0 B, Standard), 'script.js' (js, November 25, 2025, 08:33:53, 2.0 KB, Standard), and 'style.css' (css, November 25, 2025, 08:33:51, 1.9 KB, Standard).

STEP 3 — ENABLE STATIC WEBSITE HOSTING:

1. Open your bucket.
2. Go to **Properties**.
3. Scroll to **Static website hosting**.
4. Click **Edit → Enable**.
5. Enter:
 - **Index document:** index.html
6. Click **Save changes**.
7. Copy the **Website endpoint URL** (for testing).

The screenshot shows the AWS S3 Bucket Properties page for 'aws-s3-cloudfront-card-game'. In the 'Static website hosting' section, the 'Enabled' radio button is selected. A callout box provides a link to AWS Amplify Hosting. The 'Index document' field is set to 'index.html'. The 'Requester pays' section is also visible.

The screenshot shows the 'Edit static website hosting' configuration page. Under 'Static website hosting', the 'Enable' radio button is selected. Under 'Hosting type', 'Host a static website' is chosen. A note about public access is present. The 'Index document' field contains 'index.html'.

The screenshot shows the final S3 Bucket Properties page. The 'Static website hosting' section now shows 'Enabled' under 'S3 static website hosting'. The 'Bucket website endpoint' field displays the generated URL: <http://aws-s3-cloudfront-card-game.s3-website-us-east-1.amazonaws.com>.

STEP 4—ADD BUCKET POLICY (SO FILES CAN BE VIEWED):

1. Go to **Permissions** tab.
2. Scroll to **Bucket Policy** → **Edit**.
3. Paste the policy and Save the policy.

BUCKET POLICY:

...

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::aws-s3-cloudfront-card-game/*"  
    }  
  ]  
}
```

...

The screenshot shows the 'Edit bucket policy' page in the AWS S3 console. The URL is [Amazon S3 > Buckets > aws-s3-cloudfront-card-game > Edit bucket policy](#). The policy is defined in JSON:

```
1 | {  
2 |   "Version": "2012-10-17",  
3 |   "Statement": [  
4 |     {  
5 |       "Effect": "Allow",  
6 |       "Principal": "*",  
7 |       "Action": "s3:GetObject",  
8 |       "Resource": "arn:aws:s3:::aws-s3-cloudfront-card-game/*"  
9 |     }  
10 |   ]  
11 | }  
12 |
```

On the right, there's a sidebar with 'Edit statement' and 'Select a statement' sections. A link to 'Policy examples' is also visible.

The screenshot shows the 'Edit bucket policy' page in the AWS S3 console. The URL is [Amazon S3 > Buckets > aws-s3-cloudfront-card-game](#). The policy is defined in JSON:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::aws-s3-cloudfront-card-game/*"  
    }  
  ]  
}
```

Below the policy editor, there's a section for 'Block all public access' which is set to 'Off'. There's also a note about individual block public access settings for the bucket.

STEP 5 — CREATE A CLOUDFRONT DISTRIBUTION (NO CUSTOM DOMAIN):

1. Go to CloudFront → **Create distribution**.

Distribution name: aws-s3-cloudfront-card-game-distribution

2. Under **Origin**, choose your **S3 static website endpoint**.

3. Set:

- **Viewer protocol policy:** Redirect HTTP to HTTPS

4. At bottom:

- **Default root object:** index.html

5. Click **Create distribution**.

~ Wait 5–10 minutes.

~ You get a URL like:

<https://d15ofsgzywtlqm.cloudfront.net> ← your site now works with HTTPS!

The screenshot shows the 'Get started' step of the CloudFront distribution creation wizard. On the left, a sidebar lists steps: Step 1 (Get started, selected), Step 2 (Specify origin), Step 3 (Enable security), Step 4 (Get TLS certificate), and Step 5 (Review and create). The main area is titled 'Get started' and contains fields for 'Distribution name' (aws-s3-cloudfront-card-game-distribution) and 'Description - optional'. Below these are sections for 'Distribution type': 'Single website or app' (selected) and 'Multi-tenant architecture - New'. Both options have explanatory text.

The screenshot shows the 'Select S3 location' dialog box. It lists three buckets: 'aws-s3-cloudfront-card-game' (selected), 's3-dynamic-visitor-counter', and 's3-static-bday-countdown'. The 'aws-s3-cloudfront-card-game' entry is highlighted with a blue border. At the bottom right of the dialog are 'Cancel' and 'Choose' buttons. The background shows the 'Create distribution' wizard with the 'Origin' section visible.

The screenshot shows the 'Origin' configuration section of the 'Create distribution' wizard. It includes fields for 'S3 origin' (aws-s3-cloudfront-card-game.s3.us-east-1.amazonaws.com) and 'Origin path - optional' (/path). A note at the bottom states: 'This S3 bucket has static web hosting enabled. If you plan to use this distribution as a website, we recommend using the S3 website endpoint rather than the bucket endpoint.' There is also a 'Use website endpoint' button. The background shows the 'Create distribution' wizard with the 'Origin' section visible.

~ must select “use Website endpoint”.

The screenshot shows the 'Create distribution' wizard at Step 2: Specify origin. It lists three types of origins: VOD, Private VPC, and Public Origin. The 'S3 origin' option is selected, with a text input field containing 'aws-s3-cloudfront-card-game.s3-website-us-east-1.amazonaws.com' and a 'Browse S3' button. An 'Origin path - optional' field contains '/path'. In the 'Settings' section, 'Use recommended origin settings' is selected. The overall interface is clean with blue highlights on active tabs and buttons.

AWS WAF:

- It is a web application firewall that protects apps and APIs from common attacks like SQL Injection (SQLi) and Cross-Site Scripting (XSS).
- It lets you create rules to block, allow, or monitor HTTP requests.
- Works with CloudFront, ALB, API Gateway, and offers managed security rules.

~ AWS WAF is enabled to prevent cyber-attacks.

The screenshot shows the 'Create distribution' wizard at Step 4: Review and create. It displays four main sections: General configuration, Origin, Cache settings, and Security. The General configuration section shows a distribution name of 'aws-s3-cloudfront-card-game-distribution', a description field, and a billing option of 'Pay-as-you-go (\$0/month)'. The Origin section shows an S3 origin, an empty origin path, and connection attempts set to 3. The Cache settings section indicates that CloudFront will apply default cache settings. The Security section shows security protections enabled, monitor mode disabled, and no existing WAF configuration. The overall interface is clean with blue highlights on active tabs and buttons.

CloudFront > Distributions > E1ELJ1ZYH1FRWH

Successfully created new distribution.

aws-s3-cloudfront-card-game-distribution Standard

[View metrics](#)

Details	Billing	ARN	Last modified
Distribution domain name d15ofsgzywtlqm.cloudfront.net	Billing	arn:aws:cloudfront::820314685364:distribution/E1ELJ1ZYH1FRWH	Deploying

[General](#) | [Security](#) | [Origins](#) | [Behaviors](#) | [Error pages](#) | [Invalidations](#) | [Logging](#) | [Tags](#)

Settings

Name aws-s3-cloudfront-card-game-distribution	Alternate domain names	Standard logging Off
Description	Add domain	Cookie logging Off
Price class Geo all edge locations (best performance)		Default root object

[Edit](#)

CloudFront > Distributions > E1ELJ1ZYH1FRWH

aws-s3-cloudfront-card-game-distribution Standard

[View metrics](#)

CloudFront	Distributions	Behaviors
Policies		
Functions		
Static IPs		
VPC origins		
SaaS		
Multi-tenant distributions		
Distribution tenants		
Telemetry		
Monitoring		
Alarms		
Logs		
Reports & analytics		
Cache statistics		
Popular objects		

[General](#) | [Security](#) | [Origins](#) | [Behaviors](#) | [Error pages](#) | [Invalidations](#) | [Logging](#) | [Tags](#)

Behaviors (1)

Preced...	Path pattern	Origin or origin ...	Viewer protocol policy	Cache policy name	Origin request p...	Res...
0	Default (*)	aws-s3-cloudfron...	Redirect HTTP to HTTPS	Managed-CachingOptimized	-	-

LinkedIn post creation

d15ofsgzywtlqm.cloudfront.net

Ask Google about this page - d15ofsgzywtlqm.cloudfront.net

Pick-a-Card Memory Game

Match all the pairs to win!

Restart Game

Made with ❤ by Yokesh B

★ OPTIONAL (RECOMMENDED FOR SECURE SETUP)

STEP 6 — TURN OFF PUBLIC ACCESS ON S3 AND USE OAC (ORIGIN ACCESS CONTROL):

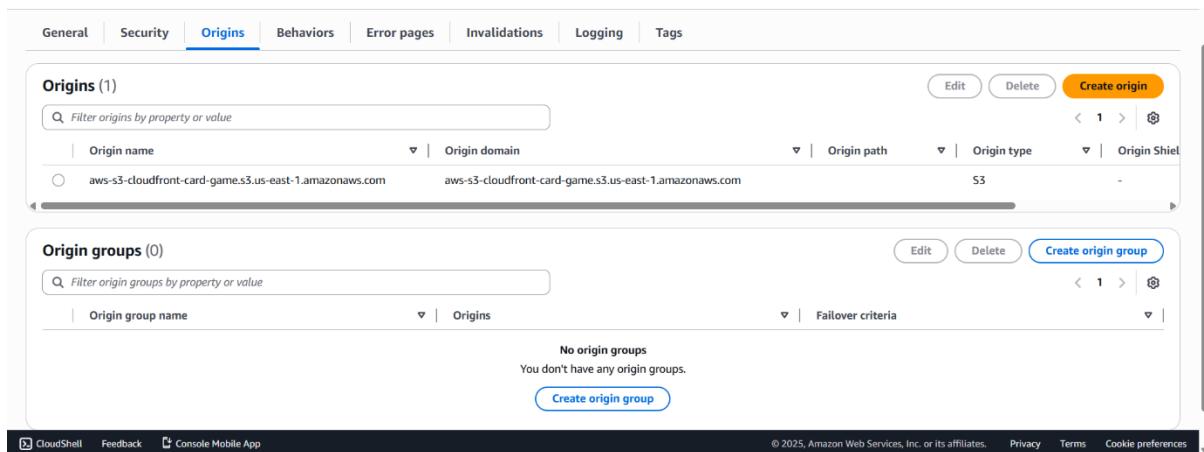
~ This ensures your S3 bucket is **PRIVATE**, and only CloudFront can access it.

1. Go to **CloudFront** → **Distributions** → **Your Distribution** → **Origins**
2. Select your origin → **Edit**
3. Under “Origin access”, choose:

Origin Access Control (recommended)

→ **Create new OAC**

4. Save changes
5. CloudFront shows a yellow banner:
“Update the S3 bucket policy”
6. Click **Copy Policy**
7. Go to **S3** → **Your Bucket** → **Permissions** → **Bucket policy** → **Edit**
8. Replace any existing policy with the CloudFront OAC policy
9. Save



The screenshot shows the AWS CloudFront Origins configuration page. The 'Origins' tab is selected. There is one origin entry listed:

Origin name	Origin domain	Origin path	Origin type	Origin Shield
aws-s3-cloudfront-card-game.s3.us-east-1.amazonaws.com	aws-s3-cloudfront-card-game.s3.us-east-1.amazonaws.com		S3	-

Below the origins, the 'Origin groups' section is shown, indicating 'No origin groups' found.

~ Now your S3 bucket becomes **private**, and CloudFront becomes the only gateway.

STEP 7 — UPLOAD YOUR STATIC WEBSITE TO S3:

1. Go to **S3** → **Your bucket**
2. Click **Upload**
3. Add all site files:
 - index.html
 - CSS files
 - JS files
 - Images

~ CloudFront automatically fetches the content from S3.

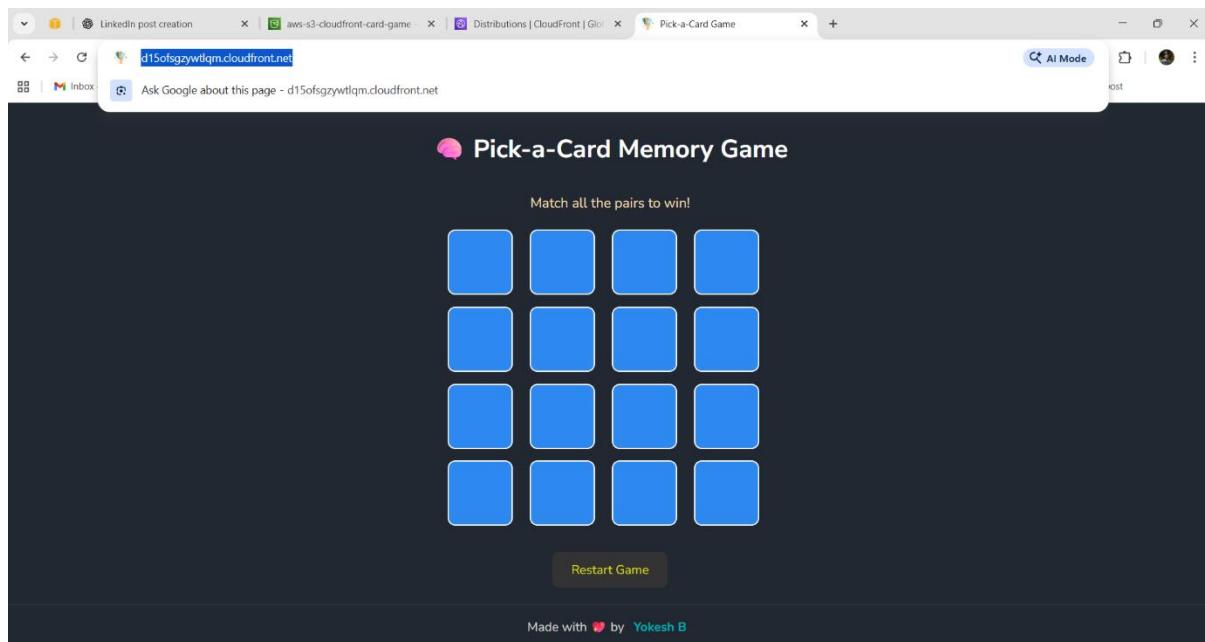
STEP 8 — TEST YOUR WEBSITE:

→ Open your CloudFront URL:

<https://d15ofsgzywtlqm.cloudfront.net>

→ You should see your website.

- ✓ Works with HTTPS
- ✓ Global CDN performance
- ✓ Free Tier friendly
- ✓ No domain needed.



Your static site is now:

- ✓ Hosted on S3
- ✓ Served securely by CloudFront
- ✓ Uses OAC for security