
🌐 Amazon S3 CORS (Cross-Origin Resource Sharing)

📘 What is CORS?

CORS stands for **Cross-Origin Resource Sharing**. It is a **security feature implemented by web browsers** to **restrict web pages from making requests to a different domain** than the one that served the original web page.

🔍 In Simple Terms:

Imagine your web app is hosted on:

https://netflix.com

And you want to fetch images, videos, or JSON data from an Amazon S3 bucket at:

https://my-app-assets.s3.amazonaws.com

This is a **cross-origin request** because the domain of your frontend and the S3 bucket are **not the same**.

🛑 **By default**, **browsers block** these types of requests for security reasons.

✅ To allow such requests, you need to configure **CORS rules** on your S3 bucket.

📦 Why S3 Needs CORS?

Amazon S3 is often used to:

- Serve **static files** to websites (images, videos, CSS, JS)
- Allow **direct uploads/downloads** from the frontend
- Serve **fonts, icons**, and other assets

In all these cases, if the frontend and S3 bucket are on different origins (domains), CORS must be explicitly enabled.

🧠 Key Concepts

Term	Explanation
Origin	The combination of domain, protocol, and port (https://domain.com)
Cross-Origin	When a request is made to a different origin than the webpage's origin
CORS Policy	A set of rules (configured in JSON) that define which domains can access it
Preflight Request	A special OPTIONS request made before actual request (for methods like PUT/POST)
Browser Enforcement	CORS is enforced by browsers only — server-to-server is unaffected

---

🔧 S3 CORS Configuration – JSON Format

CORS rules in S3 are configured using a **JSON array of rules**, like this:

```
[
  {
    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "POST", "PUT"],
    "AllowedOrigins": ["https://netflix.com"],
    "ExposeHeaders": [],
    "MaxAgeSeconds": 3000
  }
]
```

🔍 Explanation of Each Field

Field	Description

AllowedOrigins	Which domains can access the bucket (e.g., https://myfrontend.com)
AllowedMethods	HTTP methods allowed from the browser (e.g., GET, PUT, POST)
AllowedHeaders	Headers that can be used in the request (e.g., Authorization, Content-Type)
ExposeHeaders	Headers the browser can see in the response (e.g., ETag, x-amz-meta-*)
MaxAgeSeconds	How long the browser can cache the preflight response

---

🔬 Real-World Analogy

Think of **CORS** like a **guest list at a secure building**.

- Your web page is trying to enter the building (S3) to get a file.
- The building checks: "Is your domain on the allowed guest list?"
- If yes: Access granted ✅
- If no: Access denied ❌

🧪 Step-by-Step Demo: Configure CORS on S3 (AWS Console)

🛠️ Step 1: Open Your S3 Bucket

- Go to the **AWS Management Console**
- Navigate to **S3**
- Select your bucket

🛠️ Step 2: Go to Permissions

- Click on the **“Permissions”** tab

🛠️ Step 3: Edit CORS Configuration

- Scroll to **“Cross-origin resource sharing (CORS)”**
- Click **“Edit”**

🛠️ Step 4: Paste CORS Rules

Example configuration:

```
[
  {
    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "POST", "PUT"],
    "AllowedOrigins": ["https://myfrontend.com"],
    "ExposeHeaders": ["ETag"],
    "MaxAgeSeconds": 3000
  }
]
```

🛠️ Step 5: Save Changes

- Click **“Save”**
- Your bucket is now configured to allow cross-origin requests

---

📊 Use Cases

Scenario	Need CORS?	Notes

Hosting website on CloudFront + S3	✅	Allow frontend domain to fetch assets from S3
Uploading files from React app to S3	✅	PUT/POST methods must be allowed
Backend Lambda function accessing S3	❌	CORS not required for server-to-server
Vue.js/Angular fetching JSON from S3	✅	JSON MIME type, must allow application/json

---

🚨 Common Issues & Troubleshooting

Issue	Solution
CORS error: No Access-Control-Allow-Origin	Make sure you added your frontend domain to AllowedOrigins
CORS error: Preflight request failed	Add OPTIONS to AllowedMethods for PUT/POST requests
GET works, POST fails	Update AllowedMethods to include POST
Using credentials with * origin	Replace * with your actual domain when using cookies/auth

---

🧪 Bonus: Test With HTML & JS

👨‍💻 Example:

```
<!DOCTYPE html>
<html>
 <body>
  <img src="https://my-bucket.s3.amazonaws.com/logo.png" crossorigin="anonymous" />
  <script>
   fetch("https://my-bucket.s3.amazonaws.com/data.json")
    .then(response => response.json())
    .then(data => console.log(data))
    .catch(error => console.error("CORS error:", error));
  </script>
```

```
  </body>
</html>
```

✅ Summary

Term	Meaning
CORS	Cross-Origin Resource Sharing
Needed For	Frontend apps accessing S3 assets directly
Config Type	JSON
Where to Set	In the **Permissions tab** of the S3 bucket
Applies To	**Browser-based requests** only
