

# Project Author

**Name:** Waleed Ahmed

**GitHub:** <https://github.com/Waleedcoderarch/Netflix-clone-using-AWS.git>

**Institute:** SDHUB Qutub Shahi

**Course:** Cloud DevOps

---

## TITLE:

### Netflix Clone — Cloud-Based Video Streaming Platform (AWS)

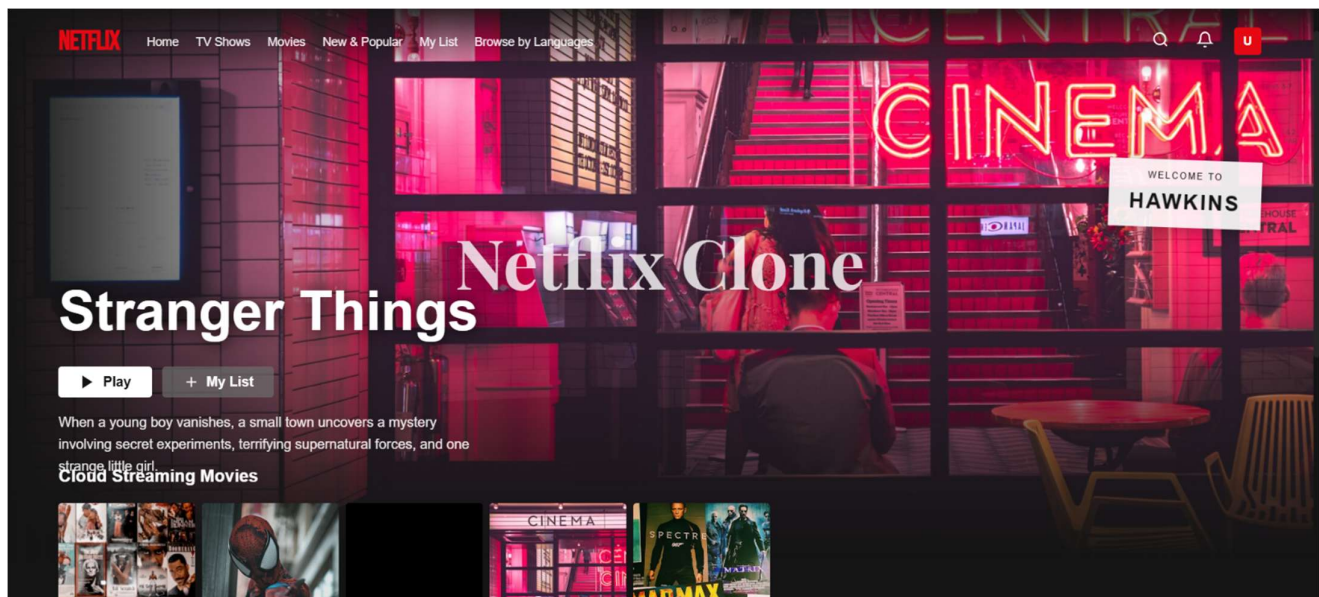
A Netflix-inspired video streaming web application developed using **HTML, CSS, and JavaScript** and deployed on **AWS Cloud Infrastructure** utilizing **Amazon S3 Static Website Hosting** and **Amazon CloudFront Content Delivery Network (CDN)**.

This project demonstrates a **real-world production-grade cloud streaming architecture**, focusing on **performance optimization, scalability, security, and global content delivery**.

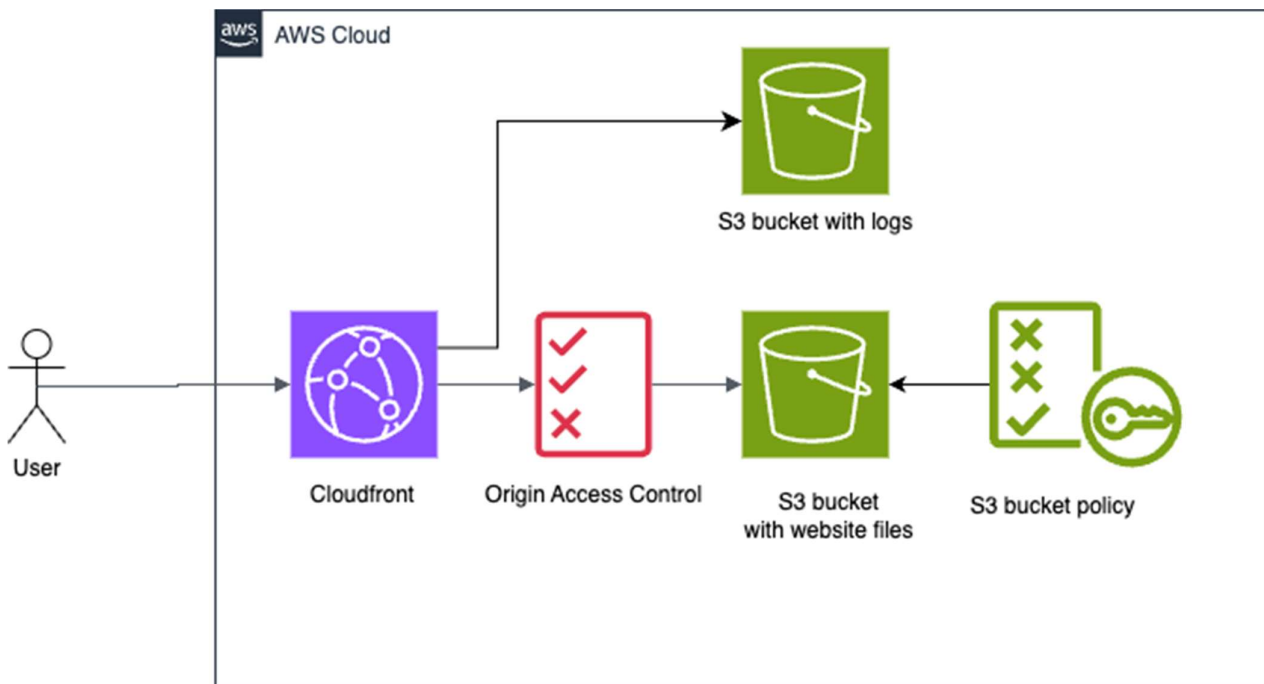
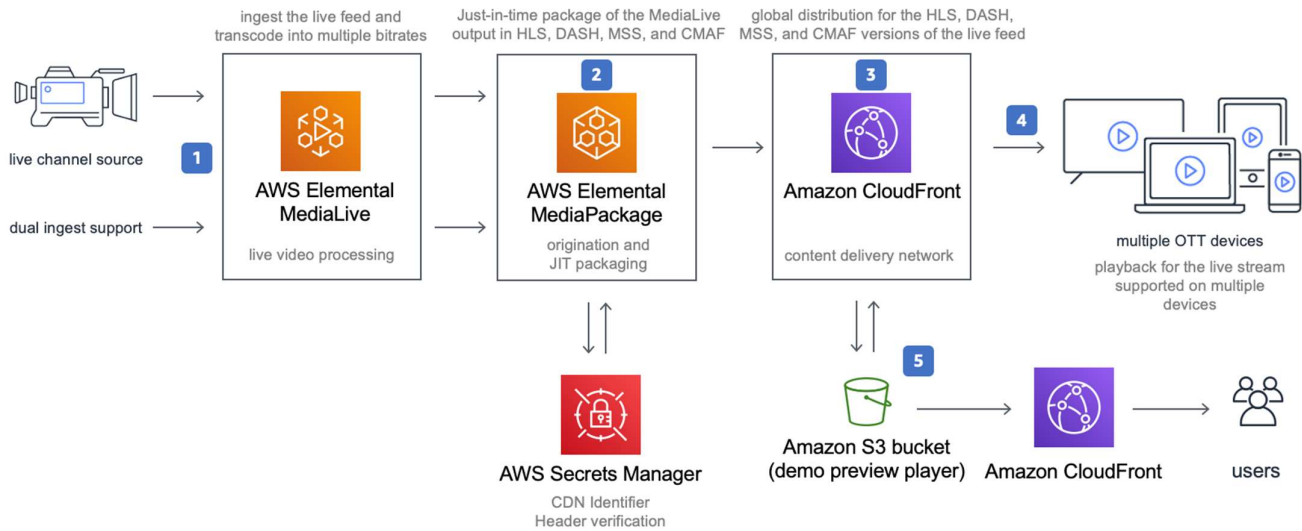
---

## Application Preview

Clean, responsive Netflix-style interface with modal video playback and smooth animations.



## Cloud Infrastructure Architecture



## Architecture Explanation

User Browser



CloudFront CDN (Global Edge Network)



AWS S3 Bucket (Website Files + Video Assets)

# Explanation

When a user opens the website, the request first goes to CloudFront CDN instead of directly reaching the S3 bucket. CloudFront helps deliver content faster to users.

CloudFront uses global edge servers. If the file is already available nearby, it is sent immediately, making the website load faster.

If the file is not in CloudFront, it is fetched from the S3 bucket and cached for future requests, improving performance and reducing load time.

---

## Technologies Used in Project

### Frontend Development

- HTML5
- CSS3
- JavaScript

### AWS (Amazon Web Service)

- Amazon S3
  - Amazon CloudFront
- 

## Implementation:

### Frontend Implementation:

The UI was developed using the Html, CSS and JavaScript what gives dynamic view of the user interface. Here, Html is used to build the base structure of the website, CSS is used for the styling and design of the code and JavaScript is used to give logic of the code.

A responsive layout was created to provide a Netflix-like experience including homepage banners, movie cards, and video player integration. Static assets such as images, stylesheets, and scripts were optimized to ensure faster loading performance.

---

### AWS S3 Implementation:

The AWS S3 bucket is used to store the file which are index.html (Code/UI) and a folder consisting of sample video inside the bucket, these sample videos will be displayed on the Netflix UI. The bucket was configured for static website hosting, allowing direct access to web assets.

After the configuration of the static web hosting, I have set the bucket policy where this explains the following;

- 1, Videos and File can be accessed through the Internet and CDN
2. Safe for the Static web hosting / 3. Allows public users to READ files

Here is the Bucket Policy:

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

---

### **(AWS CloudFront) CDN Implementation:**

The AWS CDN is used to provide the content to the user in global edge location that leads to very fast performance, latency, buffering and travels the content at high speed. I have used CloudFront service where this will ensuring faster content delivery to users worldwide.

### **The action steps performed after the creation of the S3 Bucket are as follows:**

AWS → CloudFront  
Create a Distribution  
Check Origin Tab  
Attach it to the existing S3 Bucket  
Configure HTTPS delivery

---

### **Deployment Implementation:**

After the deployment, The Netflix Clone can be successfully can be accessed through the following:

S3 Bucket: Netflix Clone can be viewed using the S3 Bucket Link but this is the static link were the main objective is to host it using the CloudFront which can be accessed globally at high speed .

CloudFront: The Netflix can be accessed by using CDN domain name that allows you to access globally which leads to serving the content of the website without buffering , users can access at high performance without using the static hosting

