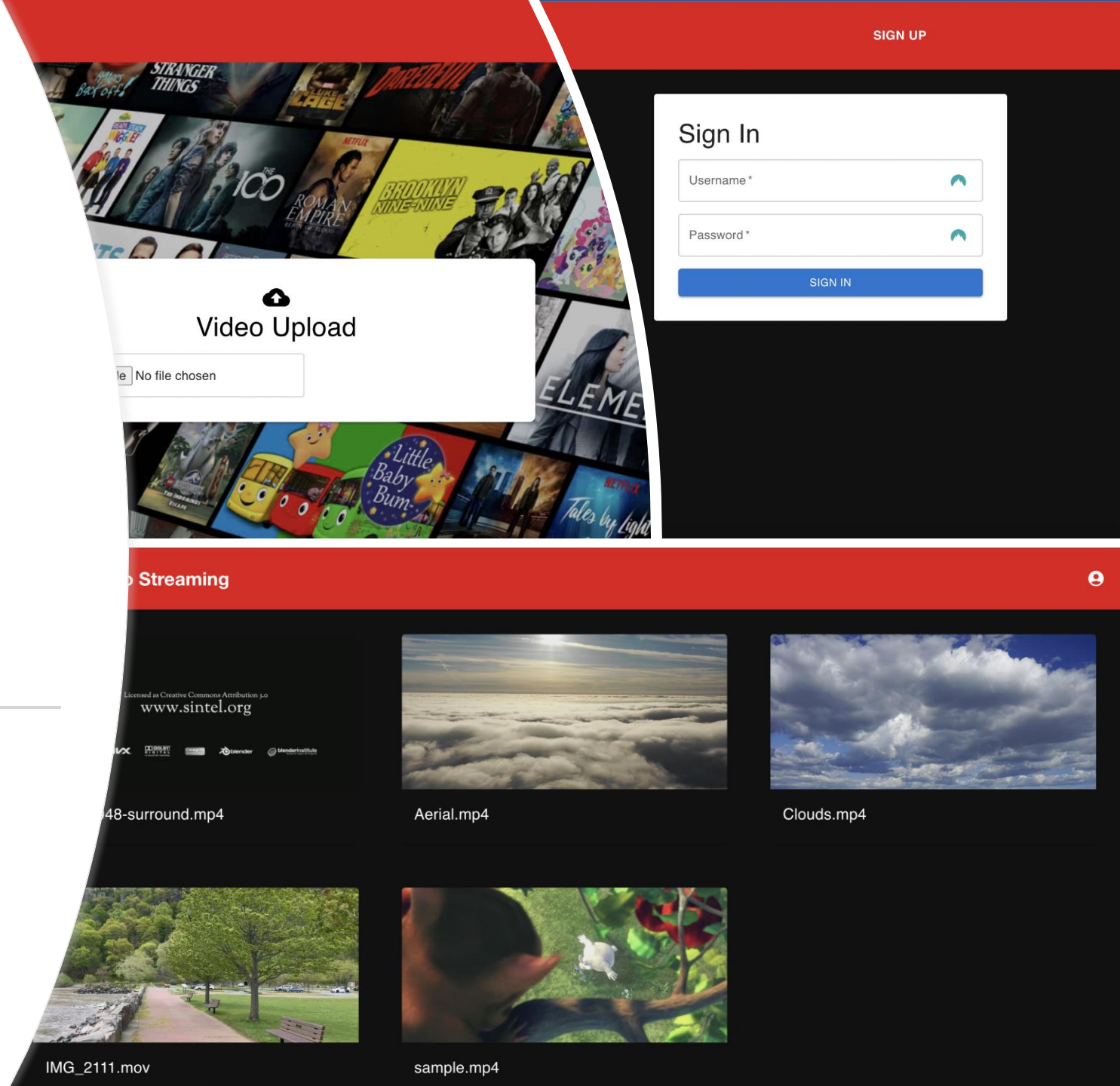


CLOUD VIDEO STREAMING DESIGN



Group Members

- ❑ Shubham Singh
- ❑ Amey Rane
- ❑ Shubham Patil
- ❑ Abhishek Sharma





Preface

- In today's digital landscape, video has emerged as the primary means of information sharing and learning. Businesses, educational institutions, and government agencies are increasingly leveraging the power of video to communicate, educate, and engage with their audiences. As a result, the demand for secure and innovative solutions to deliver and protect video content has never been greater.
- For organizations like medical institutions, universities, schools, corporations, OTT and media businesses, intellectual privacy and piracy organizations, lawyers, and government offices, security is of paramount importance. Both live streaming and video-on-demand (VOD) platforms offer powerful tools for businesses and enterprises to boost revenue, communicate effectively, and build an audience.
- Our video streaming platform aims to provide a secure, on-demand solution that rivals industry leaders like Hulu and Netflix. By harnessing the power of AWS, we have created a fully automated and secure video-on-demand solution using Amazon S3 and Amazon CloudFront, eliminating the need for traditional server-based transcoding.
- In this project, we'll build you through the process of creating our own video streaming platform, taking advantage of AWS services to deliver a streamlined, cost-effective, and secure experience for your users.

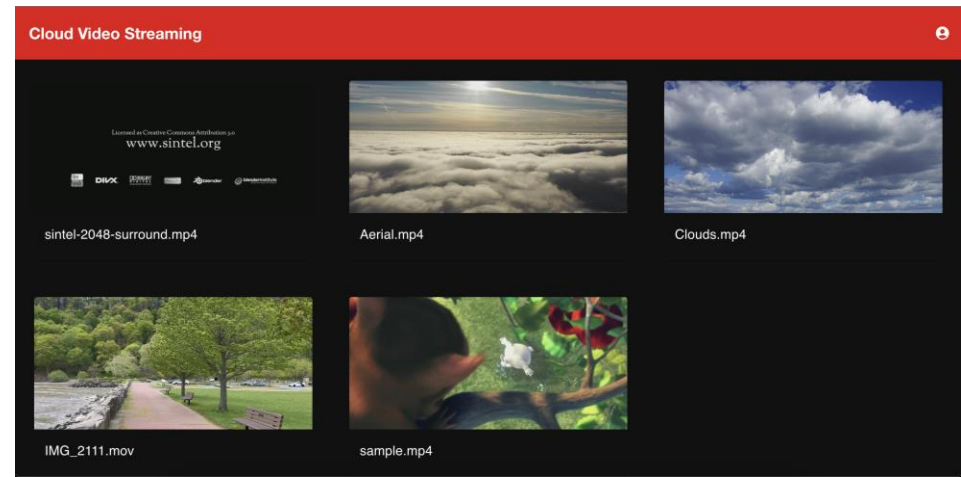
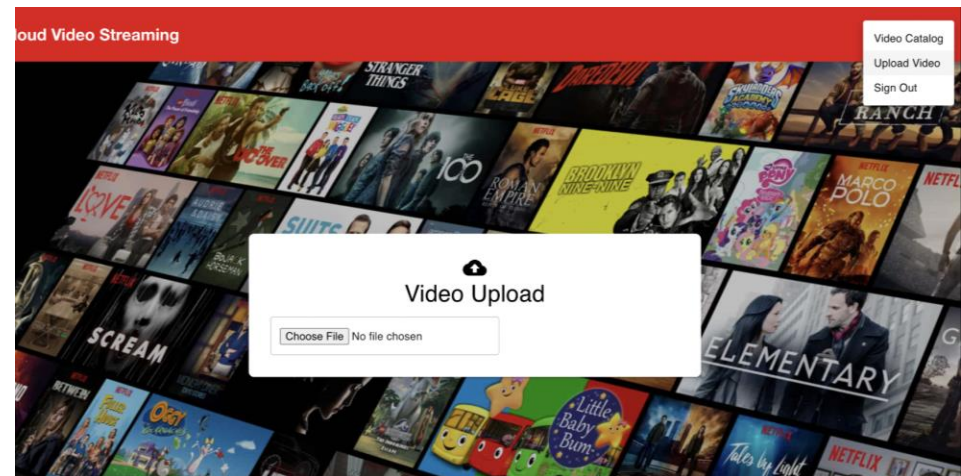
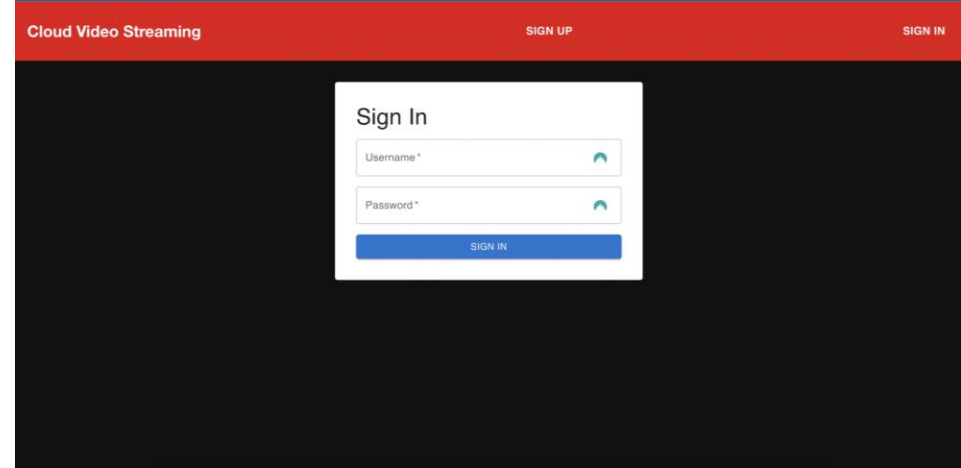


Objectives and Goals

- **Highly available & scalable streaming service application:** A functional video streaming platform that can efficiently serve content to users with minimal delays and high quality.
- **AWS architecture diagram:** A detailed and well-organized diagram illustrating the AWS architecture components, their relationships, and data flows within the system.
- **Working frontend applications:** Web application that allows users to access the streaming service, browse content, and manage their profiles.
- **Backend services and APIs:** A set of serverless functions and APIs that handle user requests, content aggregation, and other essential operations.
- **Content transcoding and adaptive streaming:** A system for converting video files into multiple formats and resolutions.
- **Documentation:** Clear and concise documentation that covers the system architecture, components, and deployment steps, enabling other persons to understand the design and implementation.

Live Demo of the Application

- Hosted on:
- <http://ott.mahwish.me>

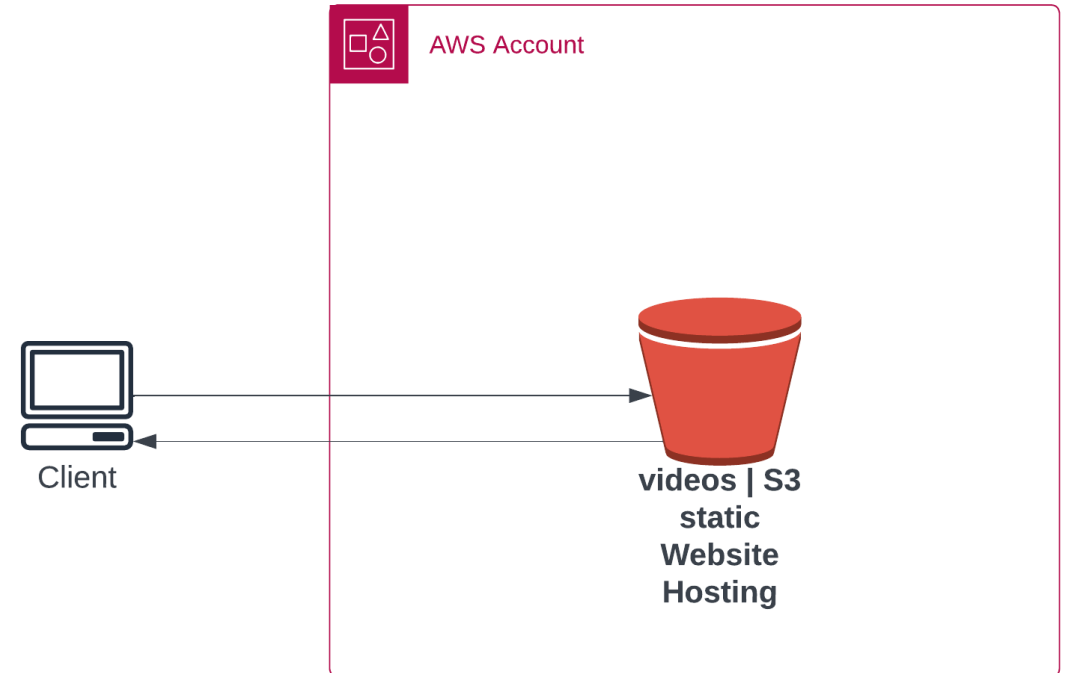


Architectural Evolution

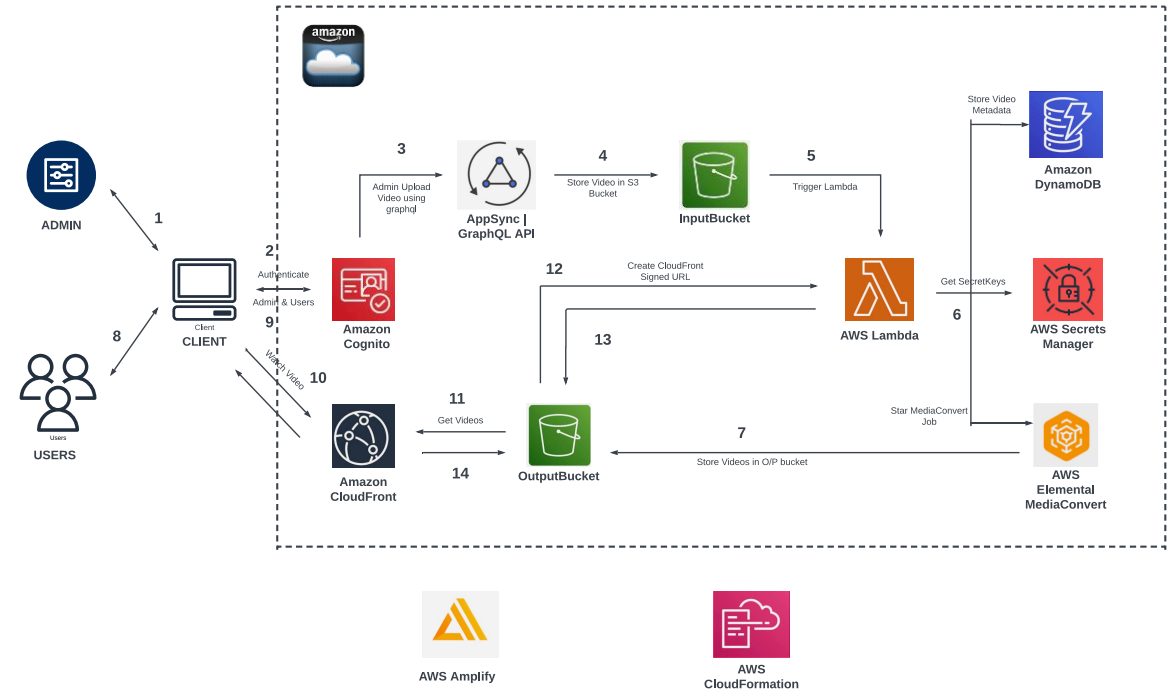
- The project has undergone a significant evolution in terms of architecture, starting from a simple UI to a fully-fledged Video On Demand solution using AWS services.

Initial UI and S3 Video Playback

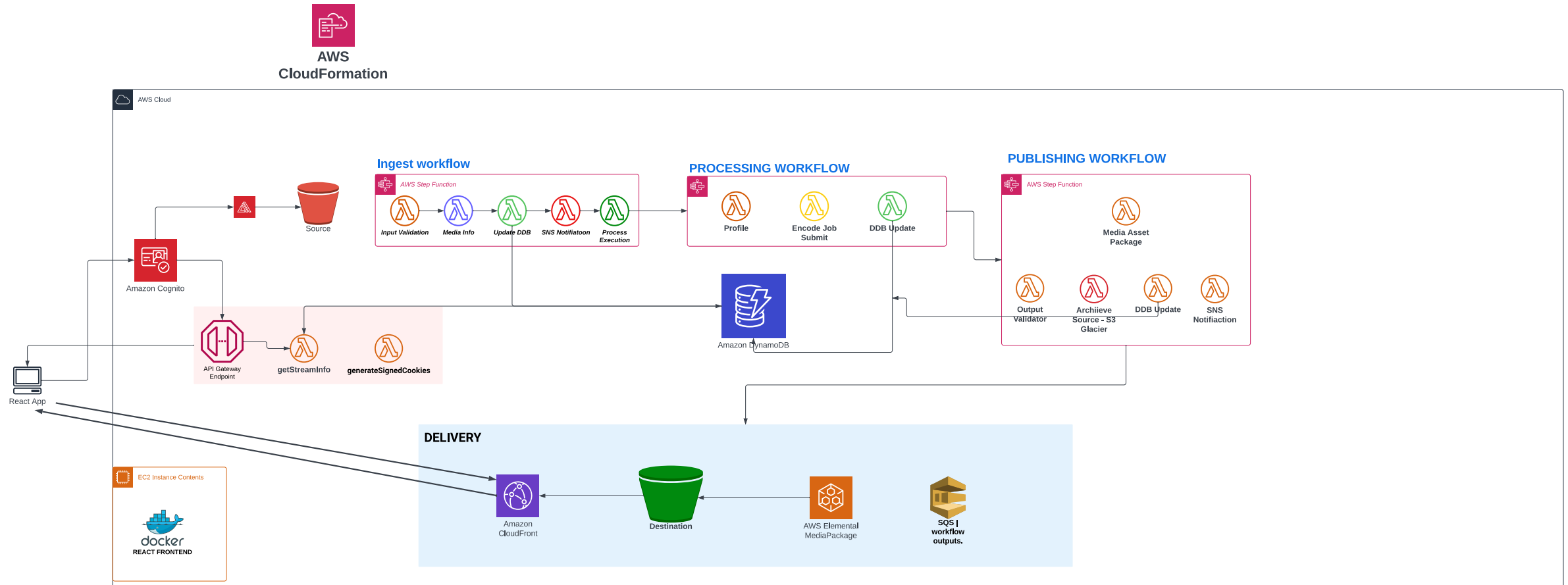
- The project began with a simple user interface that played videos uploaded to Amazon S3.
- CORS was used to enable cross-origin resource sharing for video playback.
-



Adoption of AWS Video On Demand Solution



Extending AWS Video On Demand Solution



Video On Demand Solution Architecture

- **Frontend:** React application with AWS Amplify integration for authentication, storage, and API access.
- **User Management:** AWS Cognito Service for user authentication and role-based access control.
- **Video Processing:** AWS Media Package for video transcoding, packaging, and delivery.
- **Content Distribution:** Amazon CloudFront for fast and secure content delivery.
- **Storage:** Amazon S3 for storing video files and associated metadata.
- **Serverless Functions:** AWS Lambda for executing custom logic in response to events.
- **Orchestration:** AWS Step Functions for managing and coordinating serverless functions.
- **Monitoring and Logging:** Amazon CloudWatch for monitoring performance and logging events.

Benefits of Architecture

- **Serverless Architecture leading to low cost:**
By leveraging a serverless architecture, inspired by [A Cloud Guru](#), the platform minimizes infrastructure costs and simplifies scaling.
- **Modular architecture with Step Functions:**
The use of AWS Step Functions allows for a modular architecture, making it easier to manage, update, and maintain the solution.
- **Easy User Management with AWS Cognito Service:**
AWS Cognito Service streamlines user management, providing a secure and efficient way to authenticate and manage users.
- **Seamless connection of AWS Resources using AWS Amplify library:**
The AWS Amplify library simplifies the integration of AWS resources with the client application.
 - `amplify add auth`: Sets up Cognito for user authentication.
 - `amplify add storage`: Configures Amazon S3 for storage.
 - `amplify import storage`: Imports existing Amazon S3 storage.
- **React-based modular UI:**
The user interface is built using React, which promotes modular design and a responsive, user-friendly experience.
- **Project Management with Azure DevOps:**
Azure DevOps was employed to manage the project using Agile methodologies, ensuring efficient development and timely delivery.

Challenges faced & Learnings

- **CustomVideoPlayerIntegration:**

One of the primary challenges was the integration of a video player that supports HLS and provides manual control over the bit rate. To overcome this challenge, a custom video plugin was developed for better bit rate management, offering enhanced control and performance.

- **CostManagement:**

The project aimed to minimize costs while delivering a high-quality solution. This objective was successfully achieved by leveraging AWS services and adopting a serverless architecture, which substantially reduced infrastructure costs.

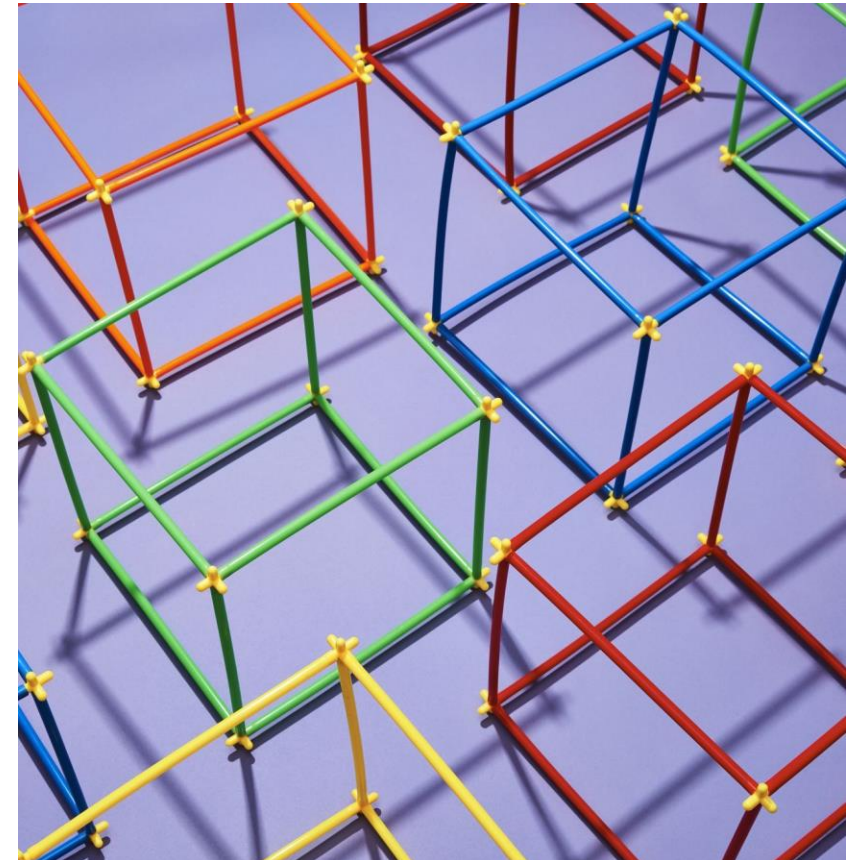
- **InitialContainerizationandTransitionto Serverless:**

Initially, the project utilized containerization with EKS and Streamlo, but due to the need for extensive customization, this approach was deemed unsuitable. Instead, a complete serverless solution was adopted using AWS Lambda, Step Functions, and other services to minimize cost and streamline the platform.



Challenges faced & Learnings

- **Signed Cookies for CloudFront:**
Implementing signed cookies for CloudFront for each video URL proved challenging. However, the team managed to generate and securely implement these cookies, ensuring secure video access and improved user experience.
- **User Segregation:**
The platform required different levels of access for users, such as admins and subscribed users. To address this, AWS Cognitive Services were employed to segregate users and provide appropriate access based on their roles.
- **Media Package vs. AWS Elastic Transcoder:**
One of the most crucial decisions was choosing between AWS Media Package and AWS Elastic Transcoder for video processing. After evaluating both solutions, the team opted for AWS Media Package due to its comprehensive feature set, which better aligned with the project's requirements and goals.



Future Scope

- **Subscription Service**

- ✓ Implement a subscription service using Stripe for secure payments and managing user subscriptions.
- ✓ Leverage the existing AWS group configuration for handling different user subscription tiers (Groups - Admin, Subscriber, Users).
- ✓ Status: In progress

- **Bitrate Control on Video Player**

- ✓ Enhance the custom video player with manual bitrate control to improve user experience and bandwidth utilization.
- ✓ Status: In progress

- **UI Improvements**

- ✓ Refine the user interface to offer a more engaging, responsive, and user-friendly experience.





Future Scope

- **Recommendation and History Microservices**
 - ✓ Introduce recommendation and history microservices to provide personalized content suggestions and keep track of user viewing history.
- **Video Thumbnails based on Frame Capture**
 - ✓ Enable frame capture from AWS Media Convertor to generate video thumbnails for a richer browsing experience.
 - ✓ Currently disabled due to cost concerns.
- **Migration to Elastic Container Service and Fargate**
 - ✓ Evaluate the potential benefits of migrating from AWS EC2 to Elastic Container Service and Fargate for better container management and scaling.
 - ✓ Not attempted yet due to cost concerns.

Conclusion

- The project provides a valuable learning opportunity for the team to gain hands-on experience in building a cloud-based web application, with a focus on video streaming architecture, scalability, performance optimization, microservices, serverless architecture, content transcoding, adaptive streaming, frontend development, and AWS services. Developing a custom OTT platform for video on demand using AWS Video On Demand Solution presented numerous challenges. However, by leveraging the power of AWS services and adopting innovative solutions, the team successfully overcame these obstacles, resulting in an efficient and cost-effective platform tailored to the project's unique needs.



Thank
You