

Media Streaming with IBM Cloud Video Streaming

Phase 5: Project Documentation & Submission

Team:

Prisha C (211521104118)

Devadarshini S (211521104035)

Heshiya Gayathri Saravanan (211521104053)

Deepthi J Mercy (211521104034)

Esha G (211521104042)

Design Thinking:

Select a Cloud Provider:

To choose a cloud provider that offers media streaming capabilities. Popular choices include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), or specialized media streaming providers like Vimeo, Brightcove, or IBM Watson Media. Consider factors like cost, geographic reach, and service offerings when making your selection.

Content Preparation:

To prepare media content by encoding and transcoding it into suitable formats and bitrates for adaptive streaming. Most cloud streaming providers offer encoding/transcoding services or integrations with third-party tools.

Cloud Storage:

By Uploading media content to the cloud storage provided by your chosen cloud provider. Ensure proper organization and access controls to manage your content effectively.

Content Security:

Implementing security measures to protect your media content. This may include content encryption, token-based access control, and DRM solutions to prevent unauthorized access and piracy.

CDN Integration:

To set up integration with a Content Delivery Network (CDN) to distribute your media content efficiently to end-users. Configure caching, edge server locations, and CDN settings for optimal performance and low latency.

Player Integration:

Embedding a media player into your web or mobile applications using the provided SDKs or APIs from your cloud streaming service. Customize the player to match your brand and user experience requirements.

Scalability and Elasticity:

To Configure auto-scaling rules to handle varying levels of traffic. Ensure that your cloud infrastructure can automatically scale up or down based on demand to maintain a smooth streaming experience during peak periods.

Monitoring and Analytics:

To implement monitoring and analytics tools to track the performance of your media streaming service. Monitor viewer statistics, user engagement, and error logs to identify and address issues promptly.

Cost Management:

To monitor and manage costs by setting up budget alerts and optimizing resource usage and ensure that we understand the pricing structure of your cloud provider to avoid unexpected expenses.

Testing and Quality Assurance:

To conduct thorough testing of our media streaming service before going live. Test on various devices and network conditions to ensure a consistent and high-quality user experience.

Documentation and Training:

To document our setup and configurations, including security policies and procedures. Provide training to our team members responsible for managing and maintaining the media streaming service.

Launch and Monitor:

To launch our media streaming service to the public or our intended audience. Continuously monitor its performance and user feedback to make improvements and adjustments as needed.

Scaling and Optimization:

As audience grows or requirements change, we must be prepared to scale our cloud streaming infrastructure accordingly. Optimize our setup based on performance data and user behavior.

Stay Informed:

Keeping abreast of industry trends and updates in cloud streaming technology. Cloud services evolve rapidly, and staying informed can help you take advantage of new features and improvements.

Customer Support and Feedback:

Being responsive to user feedback and provide excellent customer support. Address issues promptly to maintain a positive user experience.

SYSTEM DESIGN:

✓ CONTENT UPLOADING:

1. Video ingestion:

As the first step let's consider uploading data to the platform by creators. Let's say they upload a video at 4K resolution and .mp4 format to some storage along with title,

description, and tag. The storage should be secure, resilient, scalable, and cost effective. For this an ideal storage cloud service is **Amazon s3** service.

2. Database:

We need a database to store all tags, descriptions, and title which we'll need it when we search for a particular video. Using relational database will be a problem as they are not horizontally scalable. So for that we'll use a NOSQL database called **ElasticSearch** which is used for storing text fields and optimize search for that fields.

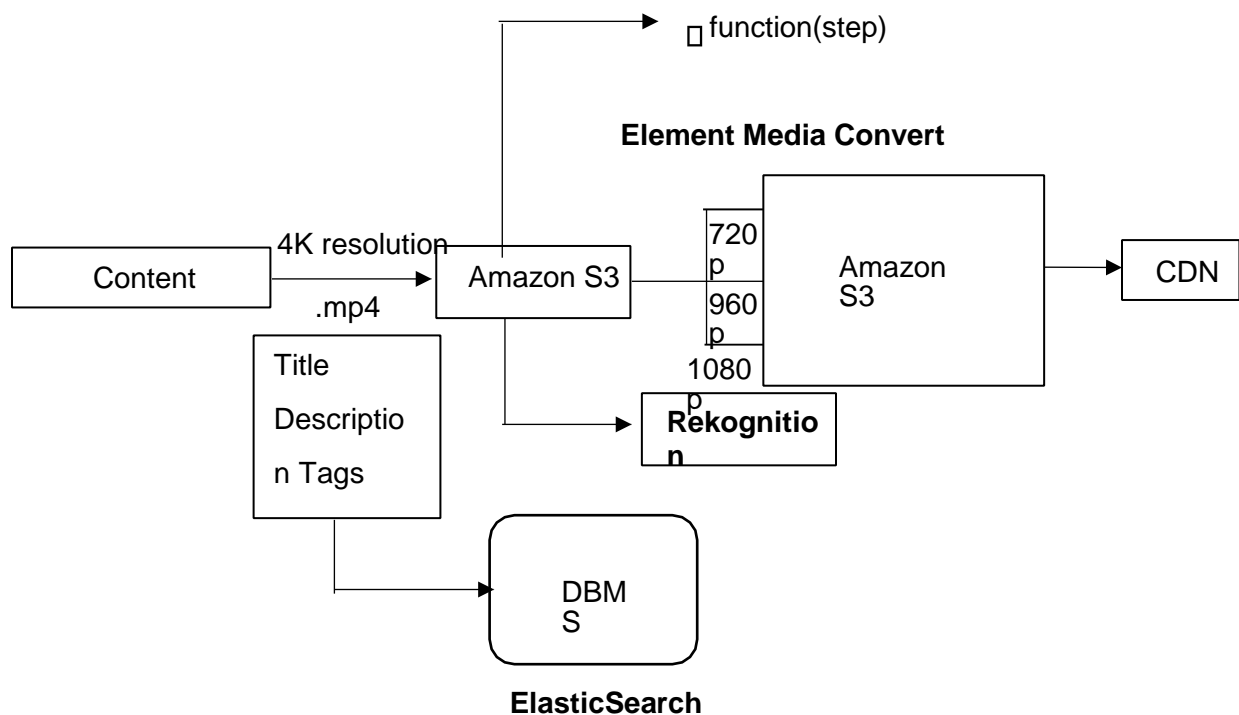


FIG.2 System architecture for video uploading

3. Video encoding:

Even if we upload in 4K resolution not everyone can watch in that resolution. It depends on the specs of the device which they are using to view the content. So this 4K resolution needs to be converted into different forms to be consumed

by different devices. For this we use AWS service called **Element Media Convert**.

4. Adult content detection:

For this we will be using an AIML service from amazon called **amazon recognition**. It gets the video analyses every frame against a presorted database and detects the adult content.

5. Content Delivery Network:

Content Delivery Network(CDN) refers to a geographically distributed group of servers located at different places which work together to provide fast delivery of content. This is not only scalable but provides a very low latency to the end user. If a person from India tries to view some video instead of coming to S3, the video will be accessed from CDN, which makes our system cost efficient.

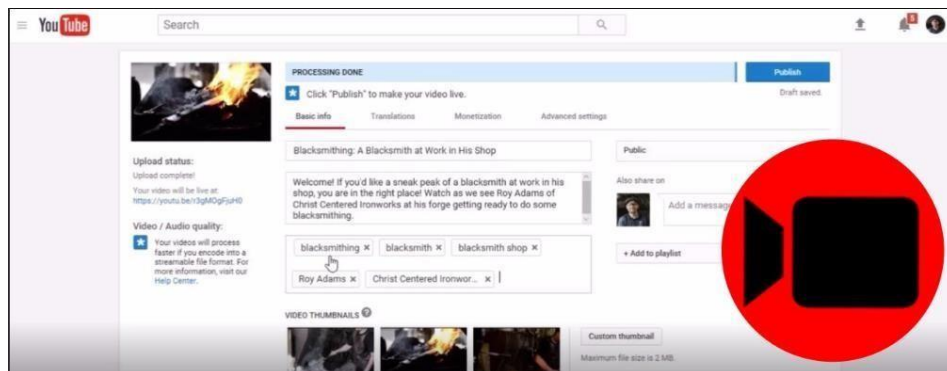


FIG. 3 Video uploading in YouTube

✓ **VIDEO SEARCHING AND VIEWING:**

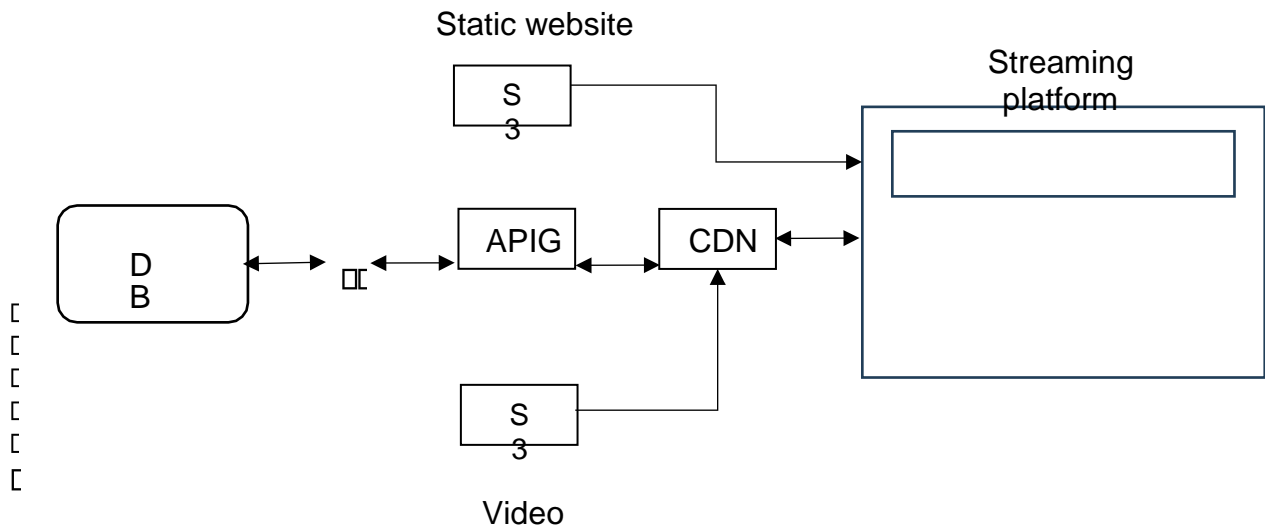


FIG. 4 System architecture for searching and viewing videos

- When you search for a video with it's name and click search it is not a static content anymore.
- It goes to CDN and searches for the video and comes back populates the screen with relevant videos.
- If it is not in CDN it calls the hosted API gateway and it calls lambda which searches in ElasticSearch Database.
- Then it loads the tags and ids of relevant video into CDN so that if another user searches for the same content it does not need to go back to the database.
- After user clicks on a particular video, with that id it checks in CDN whether the video is cached or not.
- If cached it plays from that CDN else CDN goes to the origin which is the S3 bucket and displays it.
- It also saves the video in particular resolution in CDN so that next user who wants to view the same content can directly view it from the CDN.
- The problem with MP4 format is that it can't be viewed in chunks.

- So we store them in **HLS format** (http live streaming).
- The HLS format breaks the whole video into chunks and delivers the video chunk by chunk and does not deliver the whole video.
- So if user stops to view the video after some time it only delivers the chunks worth of that time.
- It saves the network bandwidth, faster and cost efficient.

DEVELOPMENT PART:

Sign-up:

To sign up for a media streaming service in cloud computing, such as IBM Cloud Videos follow these general steps:

1. Visit the website of the media streaming service provider, in this case, IBM Cloud Video.
2. Look for a "Sign Up" or "Get Started" button on the homepage or navigation menu.
3. Click on the button to initiate the sign-up process.
4. User may be prompted to create an account by providing their email address, username, and password. Follow the instructions provided by the service provider.
5. Once the user has entered their account information, review and accept the terms of service and any other relevant agreements.
6. Some services may require additional information, such as billing details or company information, depending on the type of account the user is creating.
7. Complete any additional steps or forms as required by the service provider.
8. After successfully signing up, the user may receive a confirmation email with further instructions or a link to verify their account.
9. Follow the instructions in the confirmation email, if applicable, to verify their account and gain access to the media streaming service.

It's important to note that the specific steps and requirements may vary depending on the media streaming service provider you choose. It's recommended to visit the official website of the service provider and refer to their documentation or support resources for detailed instructions on signing up.

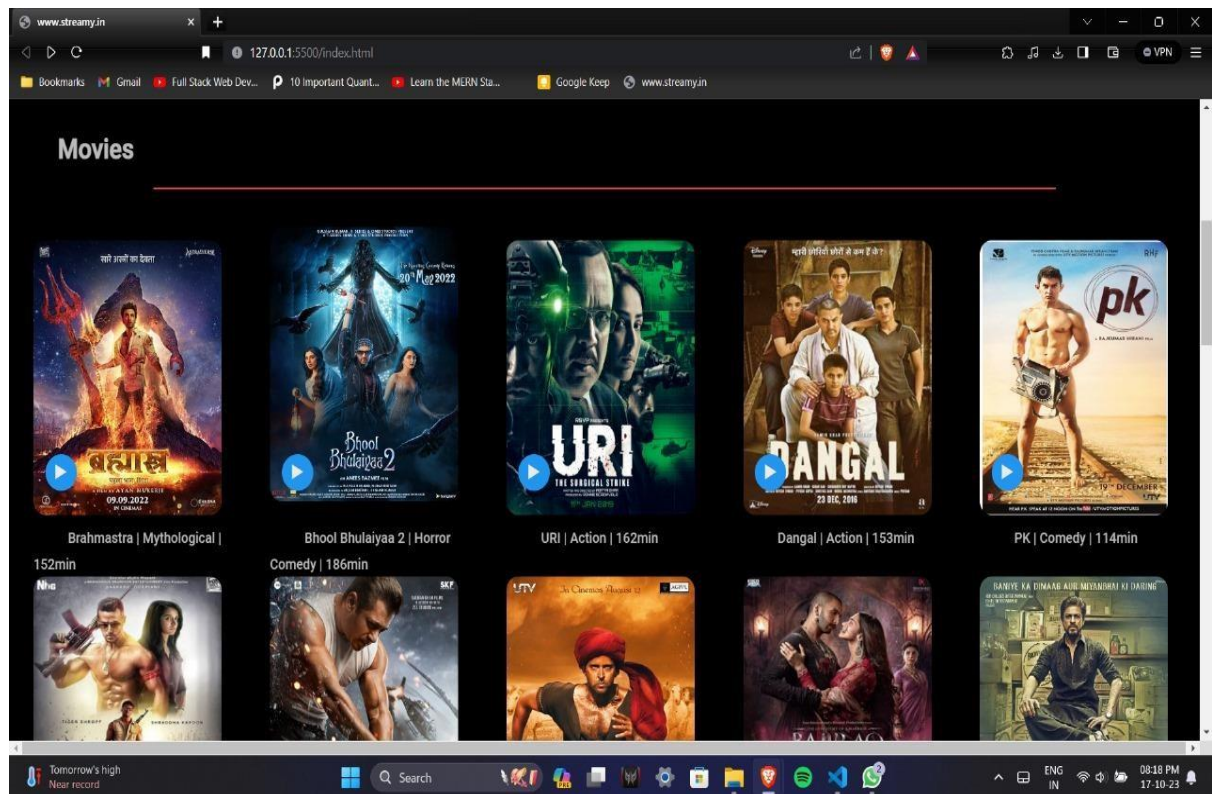
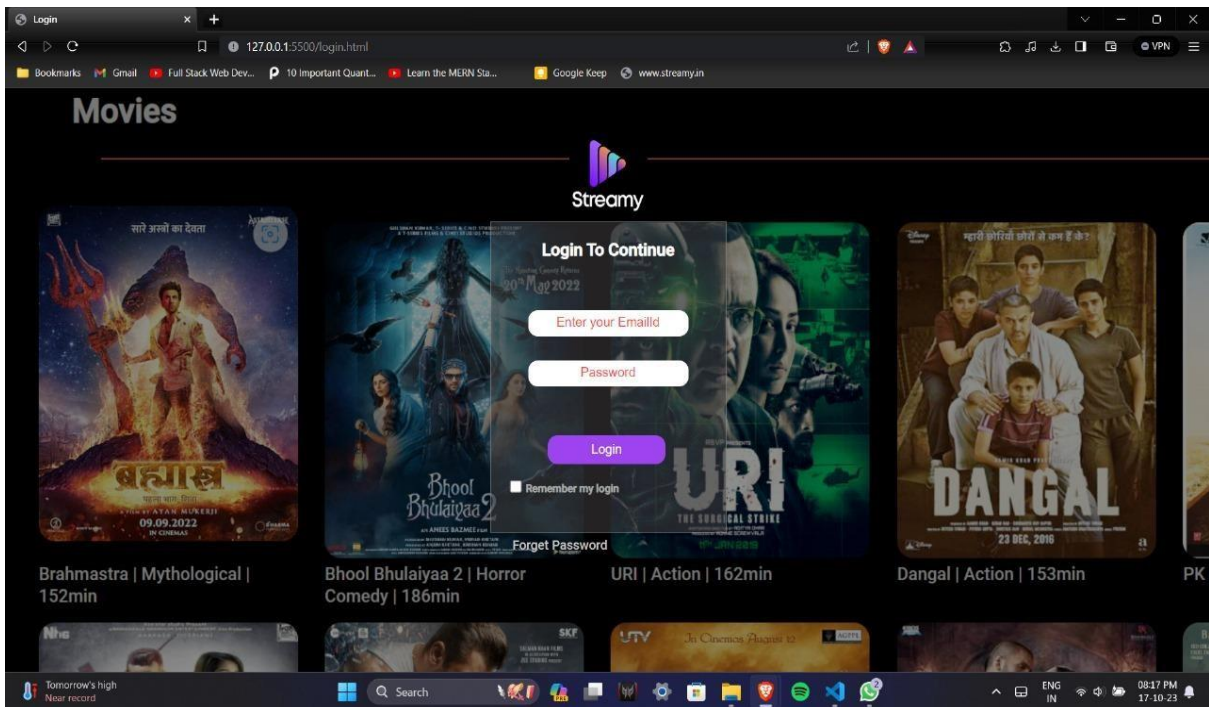
Login:

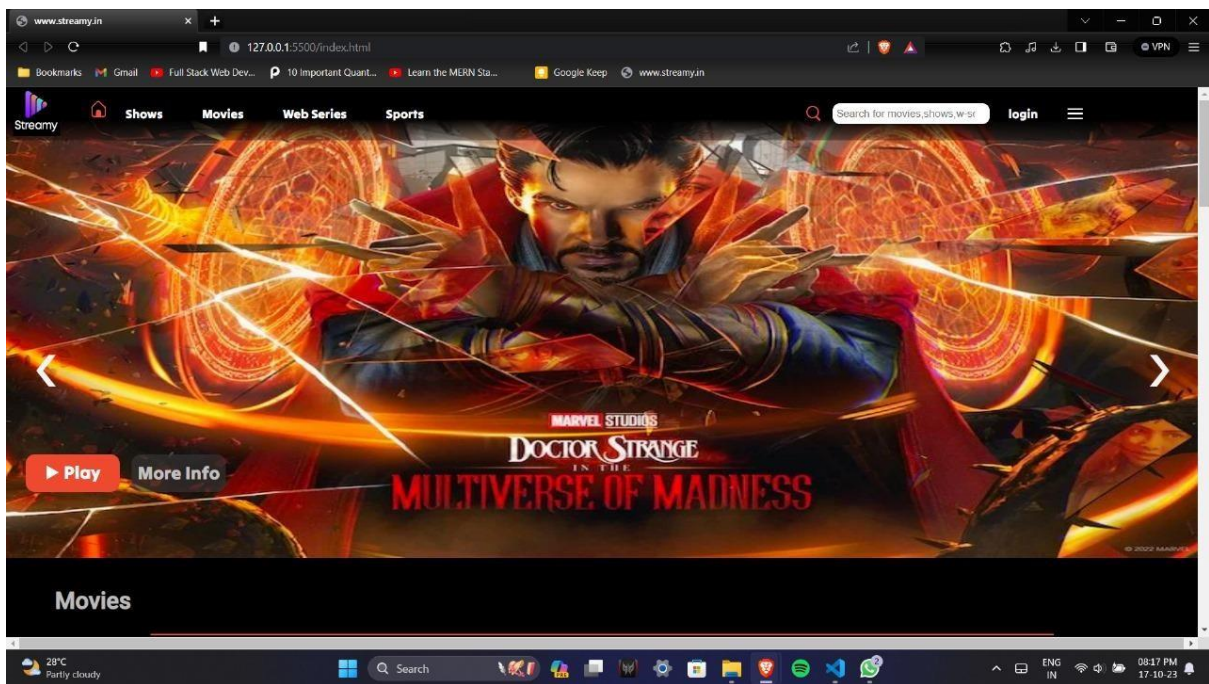
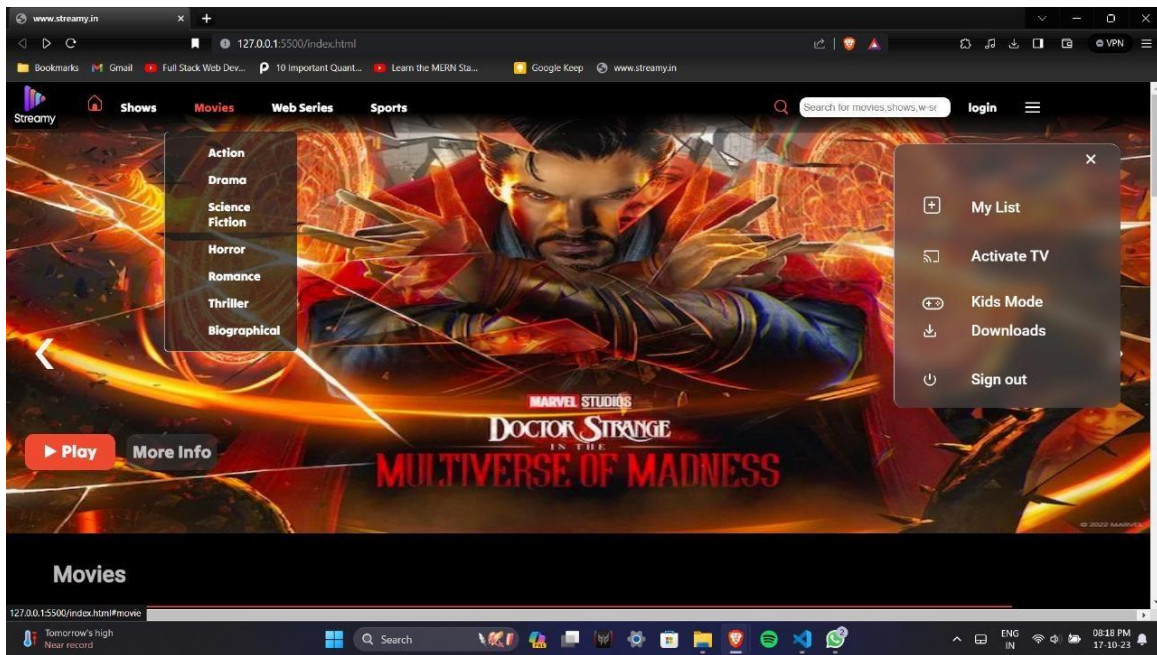
The login procedure for media streaming platforms can vary depending on the specific service the user is using.

1. Visit the media streaming platform: Open a web browser or launch the mobile app of the media streaming service the user wish to access.
2. Locate the login option: Look for a "Sign In" or "Login" button on the platform's homepage or navigation menu. Click on it to proceed.
3. Enter the credentials: On the login page, the user will usually be prompted to enter their username or email address and password associated with their account. Some platforms may also offer alternative login methods, such as using social media accounts or single sign-on options.
4. Verify the account: Depending on the platform's security measures, the user may be required to complete an additional step to verify their account. This could involve entering a verification code sent to their registered email or phone number.
5. Access the account: Once they have successfully entered their credentials and completed any necessary verification steps, the user will be granted access to their media streaming account. From there, the user can browse and enjoy the available content, create playlists, customize preferences, and perform other actions based on the platform's features.

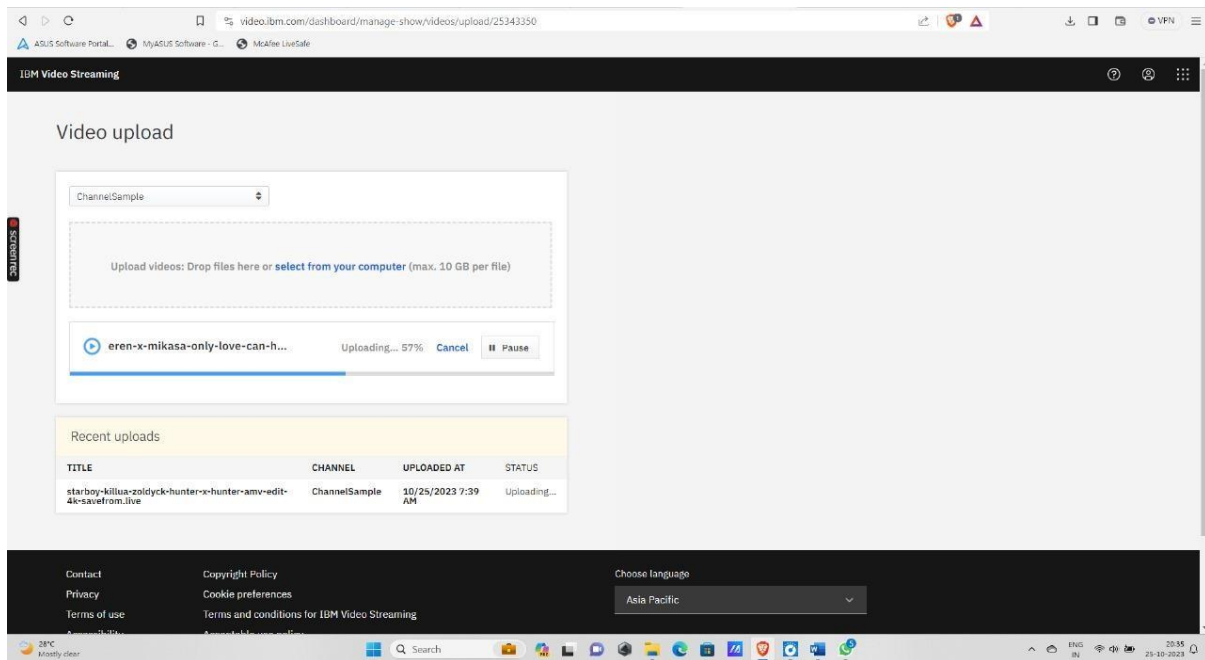
It's important to note that the specific steps and user interface may differ across different media streaming platforms. Therefore, it's recommended to refer to the platform's documentation or support resources for detailed instructions on their specific login procedure.

Media Streaming platform Frontend Output Screenshots :





Video upload:



1. Uploading videos

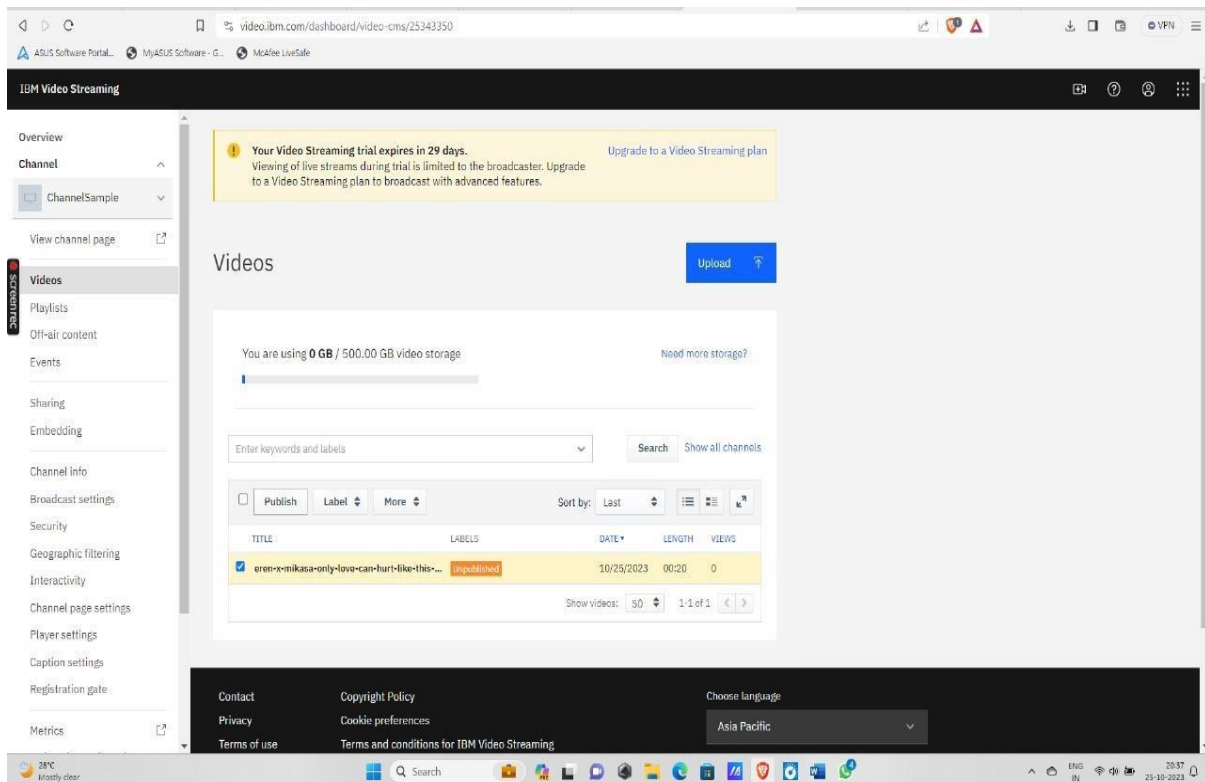
You can add videos to your IBM Video Streaming account by recording your live broadcasts, or by uploading previously recorded content using IBM Video Streaming upload tools.

Video upload capabilities

Upload in your browser

Upload through the API

Upload using professional services



2. Video file formats

Supported formats and file types

- Container: mkv, mp4, mov, flv, avi, wmv, mpeg2
- Video codec: h264, h263, mpeg4 (and variants), vp6, vp8, theora, wmv
- Audio codec: mp3, aac-lc, nellymoser, pcm (16 bit max), speex, vorbis, wma

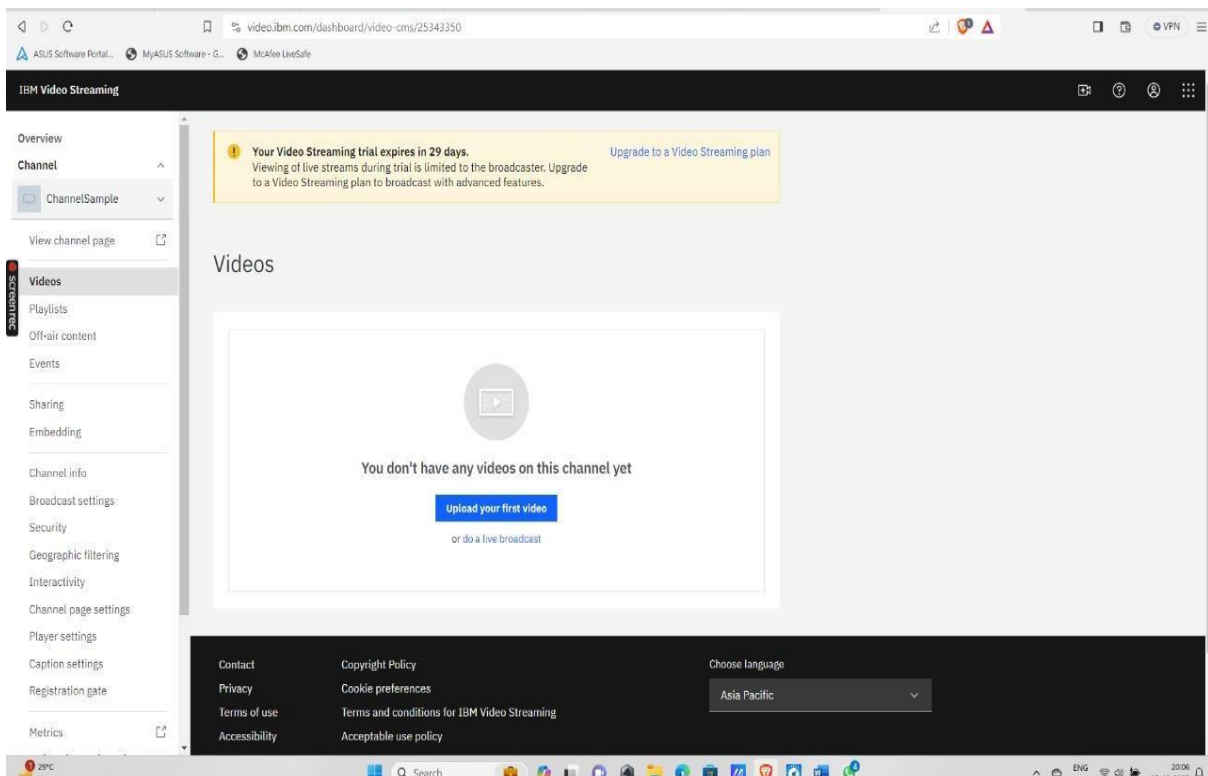
Specifications

Requirements

- Max file size: 10 GB
- Minimum length: 4 sec
- Minimum bitrate: 64 kbps

Recommendations

- Video Codec: h.264
- Audio Codec: AAC-LC



Streaming integration:

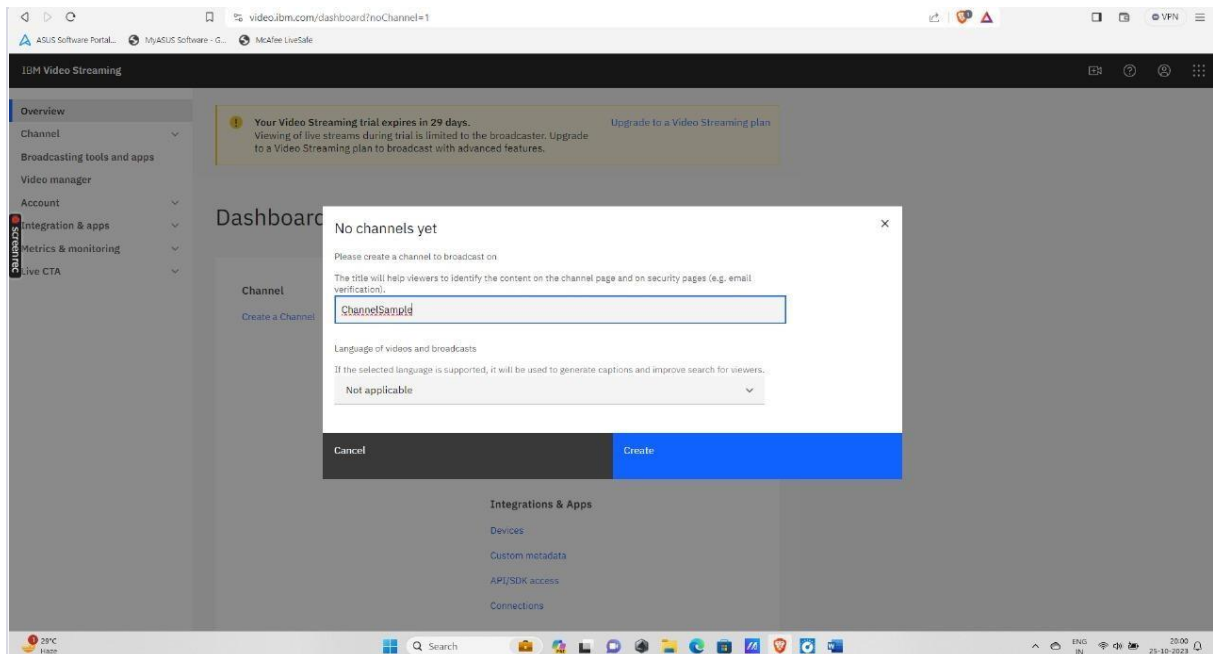
Step 1: Choose the Right IBM Streaming Service

IBM Cloud provides a variety of streaming services to cater to different use cases. It's crucial to select the one that aligns with your specific requirements. Some popular IBM streaming services include IBM Cloud Streaming Analytics, IBM Cloud Video

Streaming, and IBM Event Streams for Apache Kafka. Carefully evaluate your project's needs and goals before making a choice.

Step 2: Provision Your Streaming Service

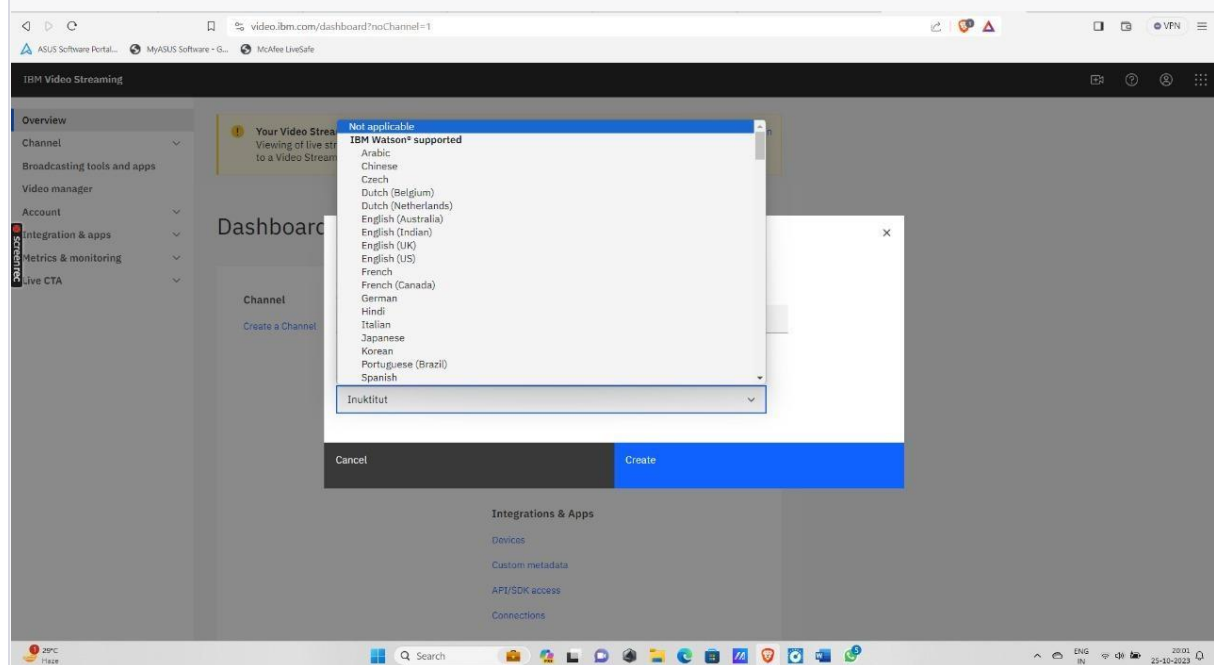
Once you've chosen the appropriate streaming service, the next step is to provision it through the IBM Cloud console. During this process, you may need to configure various settings and allocate resources. For instance, in IBM Event Streams, you may create an instance, define topics, and set access controls.



Channel Creation using IBM streaming platform

Step 3: Develop a Producer Application

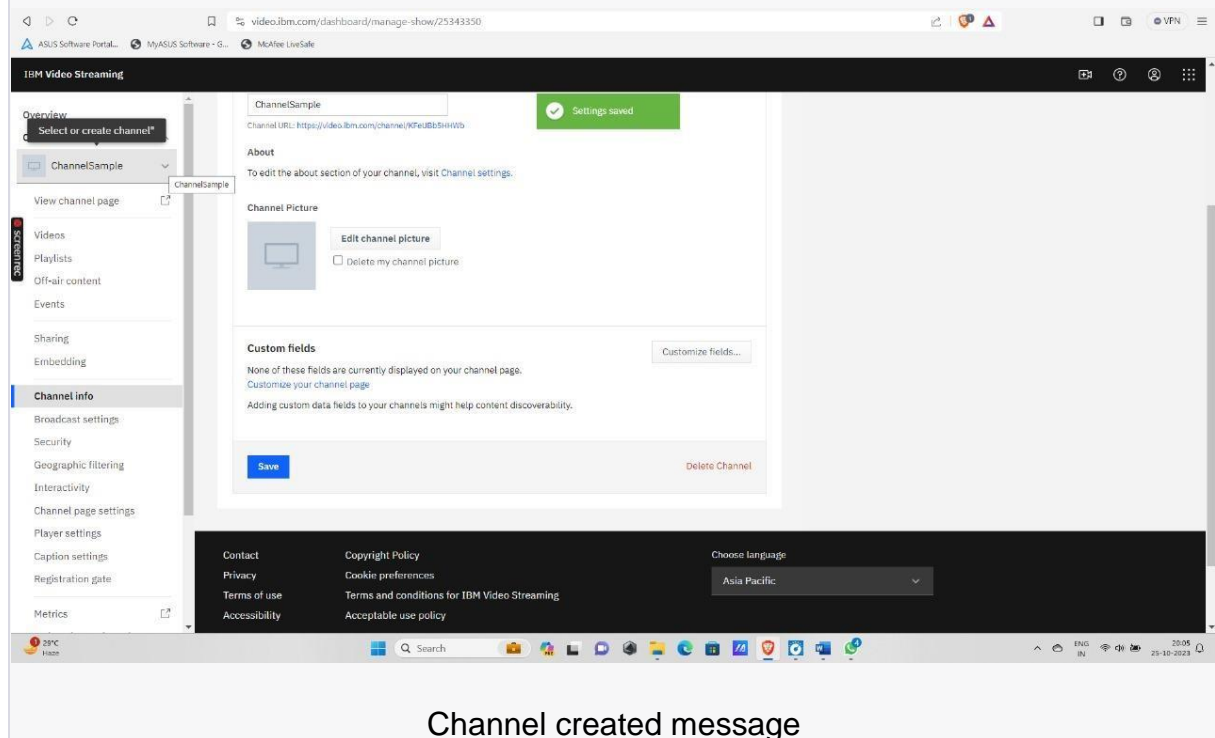
To send data or media to your streaming service, you'll need to develop a producer application. This application can be a web app, a mobile app, or a server -side script. Ensure that your producer application is correctly configured to connect to your chosen streaming service using the provided credentials and endpoints.



Language selection for channel created

Step 4: Create a Consumer Application

If you intend to consume or process the streaming data, you'll also need a consumer application. This application should connect to the streaming service and retrieve or process the incoming data. The consumer application can be designed according to your specific use case, whether it involves real-time analytics, content distribution, or any other purpose.



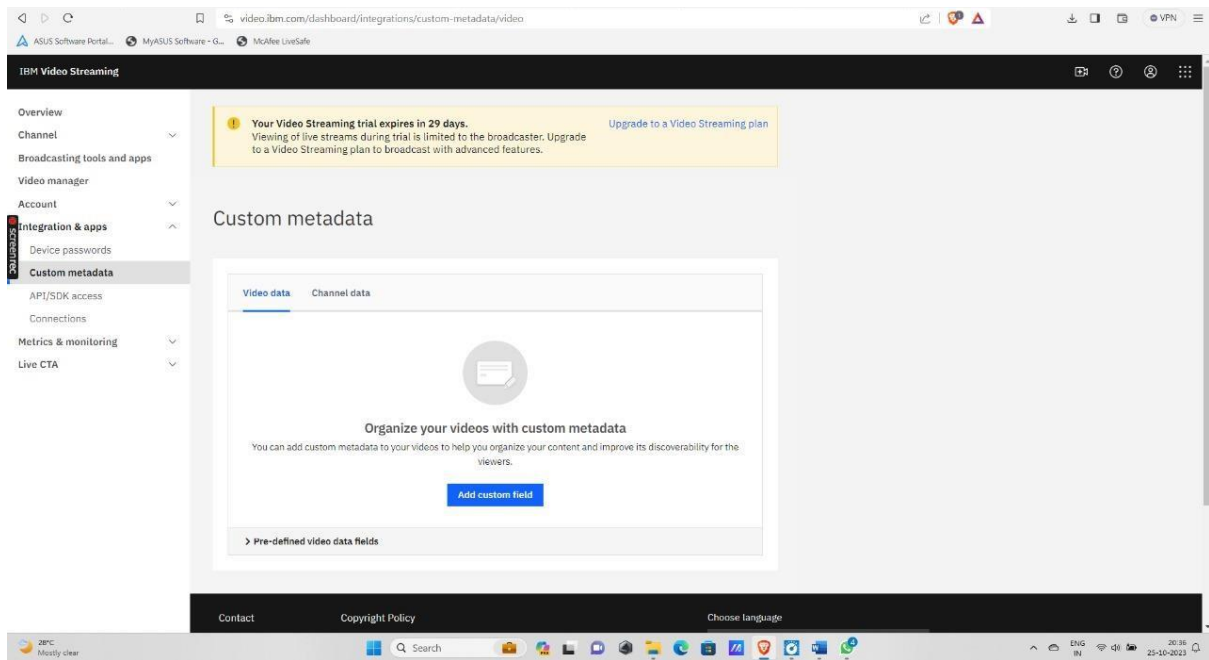
Channel created message

Step 5: Data Ingestion

With your producer application in place, you can start sending data to your streaming service. The type of data can vary widely, from sensor readings and log files to video streams and real-time events. The streaming service will take care of distributing this data to your consumer applications in real-time.

Step 6: Data Processing (Optional)

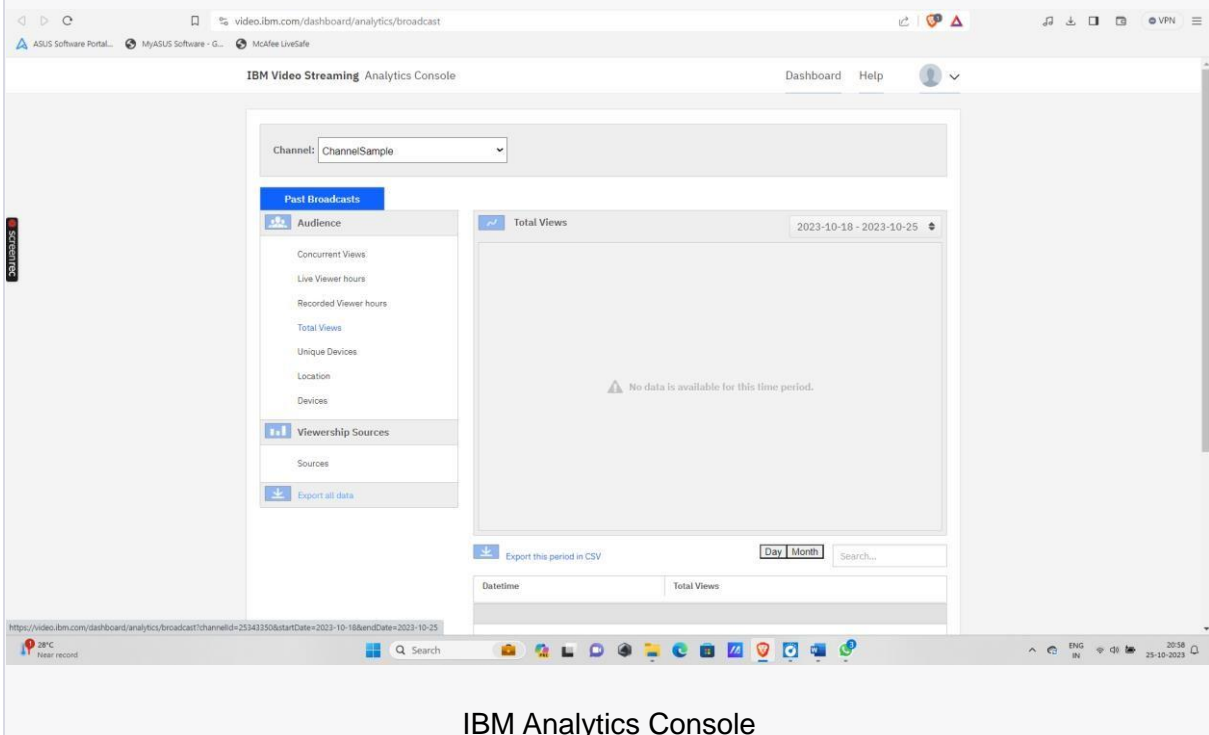
Depending on your project's requirements, you may want to apply data processing or real-time analytics to the streaming data. IBM Cloud typically provides tools for data transformation and analysis. This step allows you to derive meaningful insights from the incoming data, which can be a valuable feature for data-driven applications.



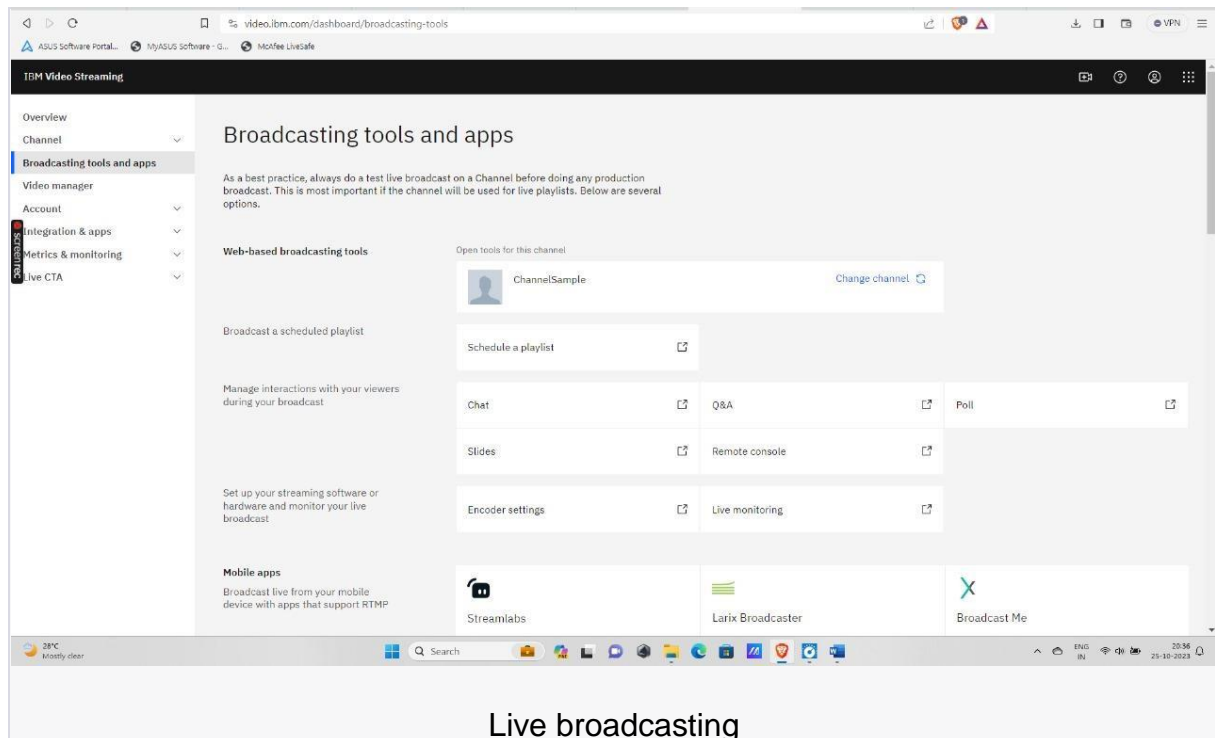
Metadata Storage in IBM Streaming

Step 7: Scaling and Monitoring

As your streaming platform grows and the volume of data or media increases, you may need to scale your streaming infrastructure to handle the added load. IBM Cloud services often offer auto-scaling capabilities, which can simplify the process of handling increased demand. Additionally, use monitoring and logging tools to keep a vigilant eye on the health and performance of your streaming applications.

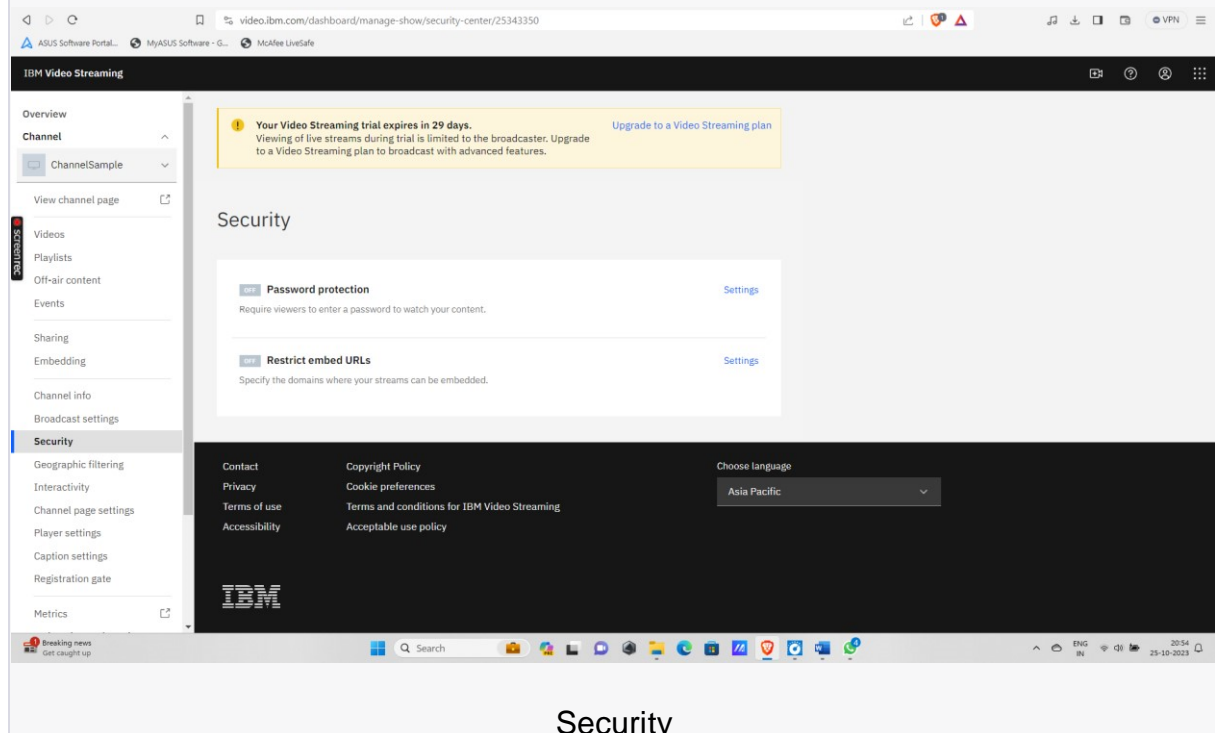


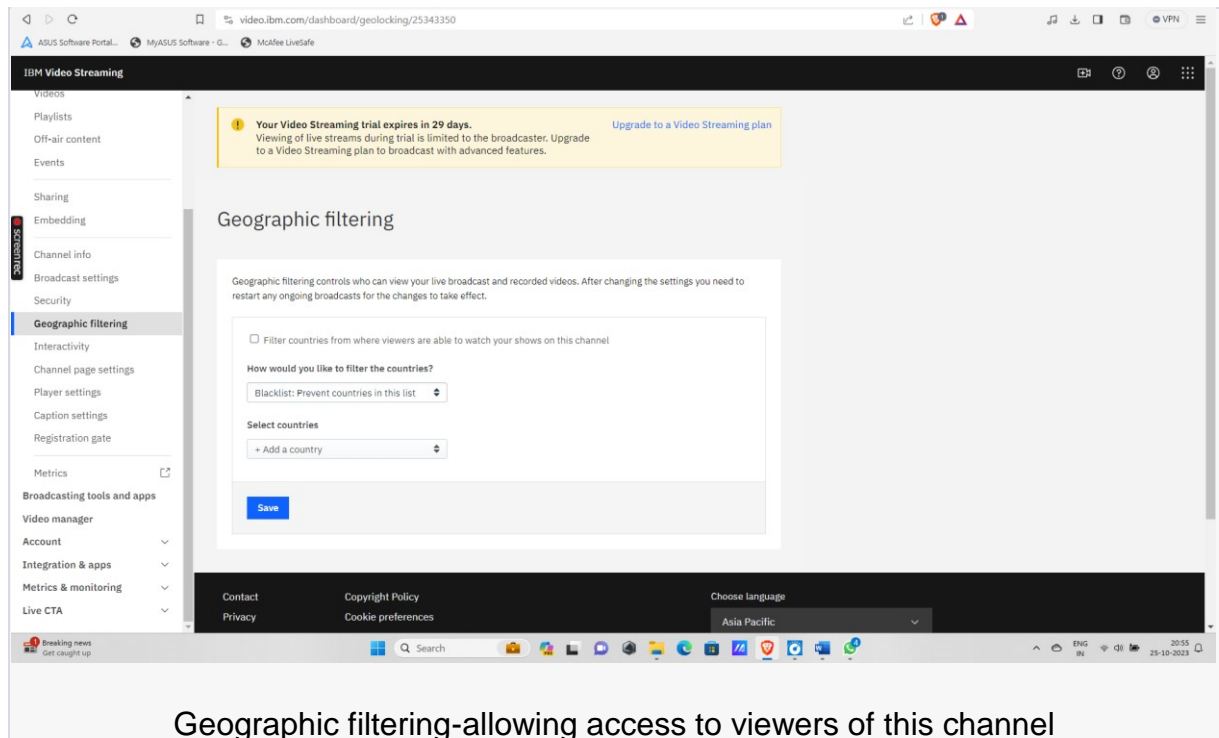
IBM Analytics Console



Step 8: Security and Access Control

Ensuring the security of your data and controlling access to your streaming service is paramount. IBM Cloud streaming services typically offer authentication and authorization mechanisms for this purpose. Properly configure these security measures to protect your data and maintain control over who can access it.





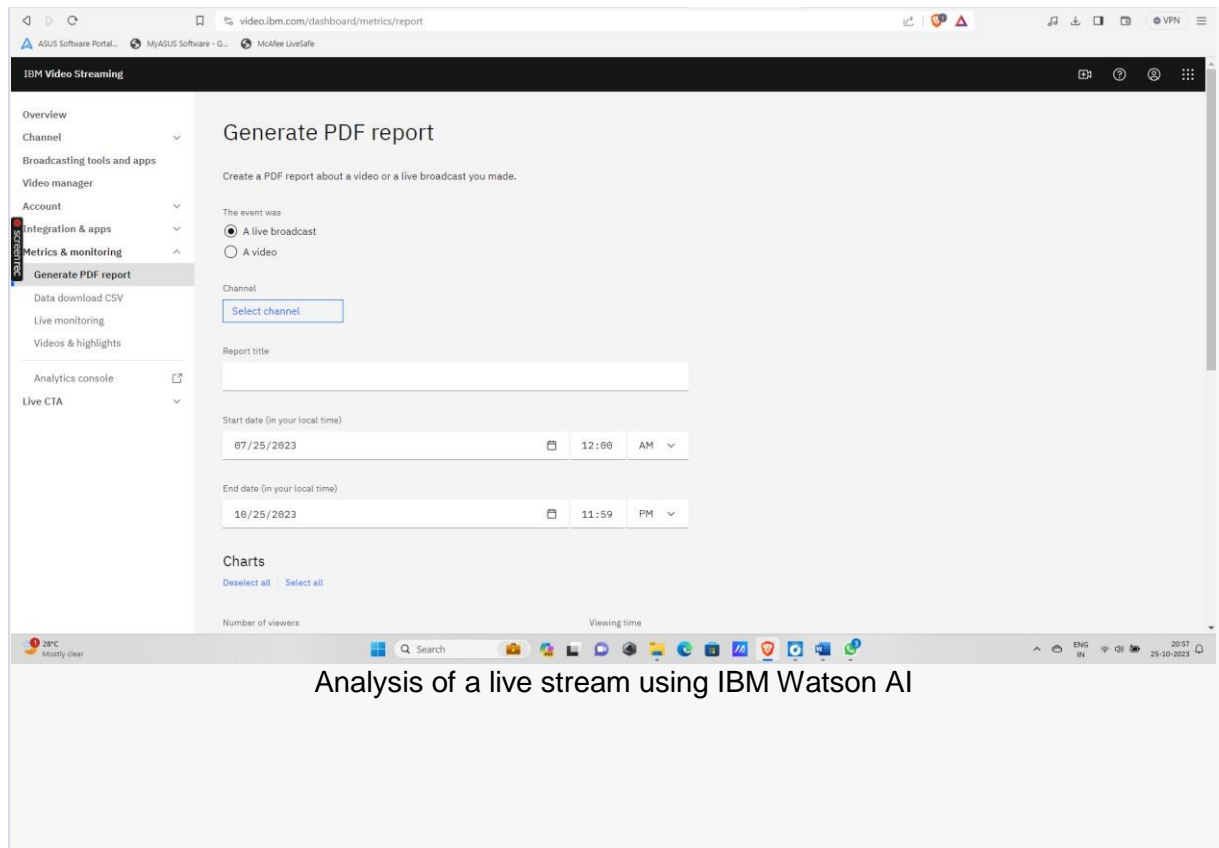
Geographic filtering-allowing access to viewers of this channel

Step 9: Testing and Optimization

Thoroughly test your streaming application to ensure it meets your performance and reliability requirements. As you collect data from the usage of your platform, continually optimize your setup to enhance the quality of your streaming service. Regular performance testing and optimizations are essential for a successful streaming platform.

Step 10: Documentation and Support

IBM Cloud offers comprehensive documentation and support resources. Familiarize yourself with these resources to troubleshoot any issues that may arise during the integration and operation of your streaming platform. These resources can be invaluable for obtaining expert guidance and making the most of your streaming solution.



Analysis of a live stream using IBM Watson AI