CLOUD APPLICATION DEVELOPMENT (CAD)

Project : Media Streaming with IBM cloud Video Streaming

Phase 3 : Development part-1

INTRODUCTION:

Media streaming with IBM Cloud Video Streaming is a powerful and versatile solution for businesses and content creators looking to deliver high-quality video and audio content to a global audience. This cloud-based service, offered by IBM, enables the distribution, management, and monetization of live and on-demand video content over the internet. It is designed to meet the demands of various industries, including entertainment, education, e-commerce, and enterprise communications.

KEY COMPONENTS:

- Live streaming
- Video On-Demand
- Security
- Analytics
- API Integration

DEVELOPMENT IDEAS:

To create a Python program for media streaming with IBM Cloud Video Streaming, you would typically use the IBM Cloud Video Streaming API. Here's a basic outline of how you might approach it, but keep in mind that the specific API endpoints and methods may have changed since my last knowledge update in September 2021. Please refer to the IBM Cloud Video Streaming API documentation for the most up-to-date information.

1. **Set Up Your IBM Cloud Account:**

   You need to have an IBM Cloud account and create a resource for Video Streaming.

2. **Install Required Libraries:**

   You may need to install some Python libraries to work with HTTP requests, such as requests.

   “pip install requests “

3. **Get API Key and Access Token:**

Generate an API key and access token from the IBM Cloud Console. You'll need these for authentication.

4. **Write Python Code:**

```
import requests
# Define API endpoint and headers
api_endpoint = "https://api.video.ibm.com"
api_key = "YOUR_API_KEY"
access_token = "YOUR_ACCESS_TOKEN"
headers = {
   "Content-Type": "application/json",
   "Authorization": f"Bearer {access_token}"
}
# Create a live event
def create_live_event(event_name):
   data = {
      "name": event_name,
      # Add other parameters as needed
   }
   response = requests.post(f"{api_endpoint}/liveevents", json=data, headers=headers)

   if response.status_code == 200:
      event_id = response.json()["id"]
      return event_id
   else:
      print("Failed to create a live event.")
      return None
# Start a live event
def start_live_event(event_id):
```

```
    response = requests.post(f"{api_endpoint}/liveevents/{event_id}/broadcast",
headers=headers)


    if response.status_code == 200:
        print("Live event started.")
    else:
        print("Failed to start the live event.")
# Main program
if __name__ == "__main__":
    event_name = "MyLiveEvent"
    event_id = create_live_event(event_name)


    if event_id:
        start_live_event(event_id)
```

This is a basic example of how to create and start a live event. You'll need to consult the IBM Cloud Video Streaming API documentation for more advanced features and options.

5. **Run Your Python Program:**

    Save the Python code to a file and run it using python **your_program.py.**

    Remember that the code provided here is a basic example. Depending on your specific use case, you may need to include error handling, manage video content, and implement additional functionality. The IBM Cloud Video Streaming API documentation should be your primary resource for implementing more advanced features.

CONCLUSION:

Media streaming with IBM Cloud Video Streaming offers a powerful platform for organizations and developers to deliver high-quality live and on-demand video content to their audiences.

In conclusion, IBM Cloud Video Streaming is a robust platform for organizations and individuals looking to leverage the power of media streaming. It combines ease of use with advanced features, making it a versatile solution for delivering video content to a global

audience. However, it's essential to review the latest features and pricing options, as these can change over time, and consider your specific requirements and budget when choosing a streaming platform.

THANK YOU...