**MEDIA STREAMING WITH IBM CLOUD VIDEO STREAMING**

**What is IBM Video Streaming?**

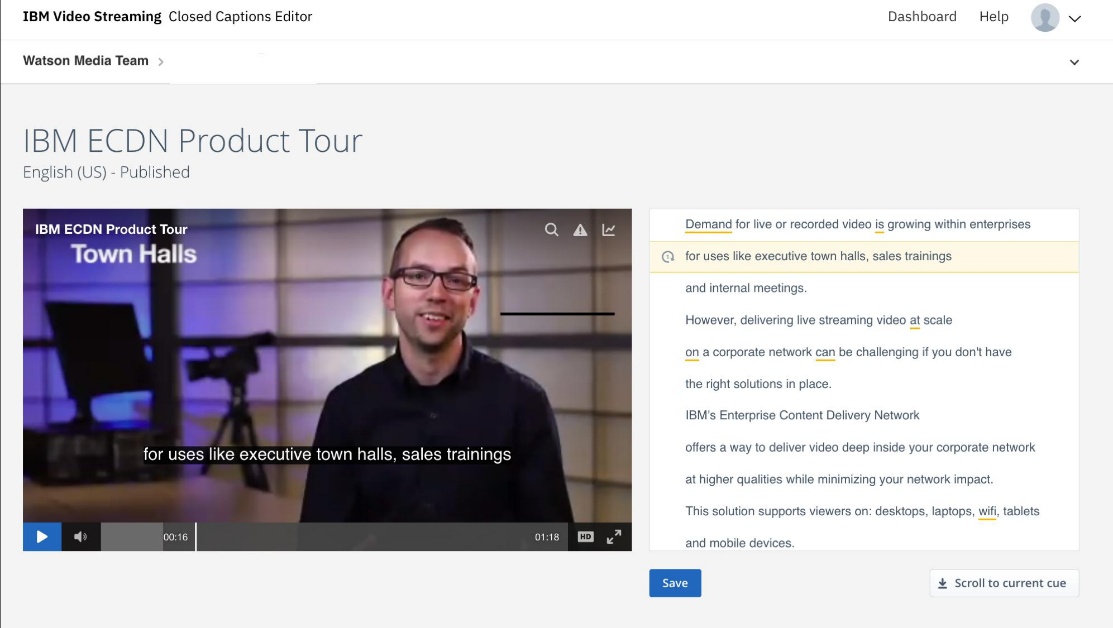
IBM Watson Media provides a video streaming and virtual event platform, powered by Watson AI. There are two editions: IBM Enterprise Video Streaming (private virtual events or internal corporate communications) IBM Video Streaming (external virtual events or unsecured communications) Both editions include storage, event support, custom branding, content management, multi-user accounts, and real-time analytics. Both are powered by IBM Watson AI for automatic speech to text conversion and automated closed captioning. IBM Enterprise Video Streaming - For streaming to authenticated individuals who are either all part of the same organization or, for external audiences, an administrator uploads authorized email addresses in advance to allow access via email authentication (or a 3rd party registration system can be used via API connector).  If a 3rd party system is used, IBM Enterprise Video Streaming along with the IBM Authorization API are used to seamlessly connect. Live streams and videos can be embedded in any other website as well and retain security settings when used this way. To request a free trial of this product: http://watson.media.ibm.com/enterprise-video-free-trial.html   IBM Video Streaming - For streaming to individuals either via Open URLs or adding secure passwords before entry. Includes an optional registration gate (the form captures contact detail for tracking, but does not restrict anyone from watching). Live streams and videos can be embedded in any other website as well and often times customers use their own web application for having users register and then just use the embeds from IBM Video

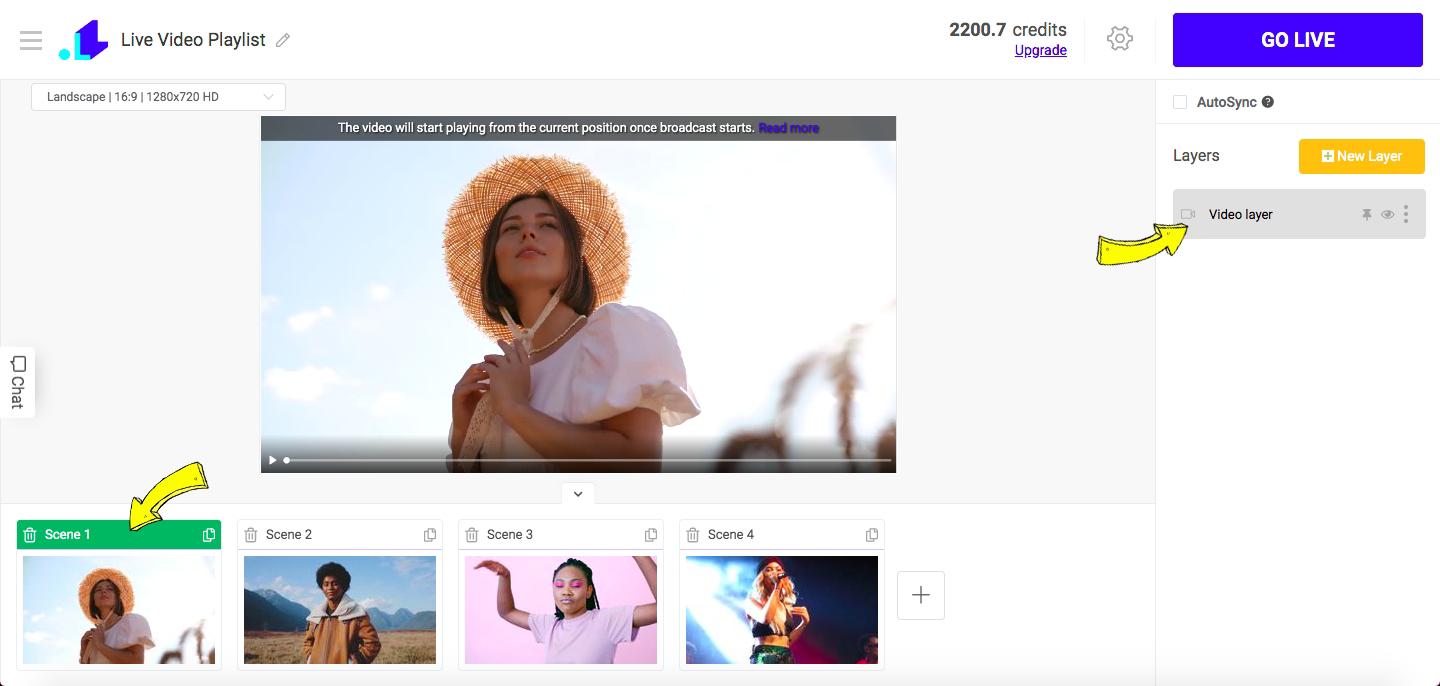
* IBM Enterprise Video Streaming can power video-based communications ranging from employee town halls, to trainings and department meetings, to digital events – boosting engagement from virtually anywhere.  
    
  Through a cloud-based solution, alleviating costly updates and continued maintenance from IT, administrators can manage a security-rich end user experience. This experience includes AI-driven deep search and the ability to track usage down to the individual user level with metrics as detailed as when content was accessed, device information, geographic location of the viewer and completion percentage.
* **Security-rich video solution**

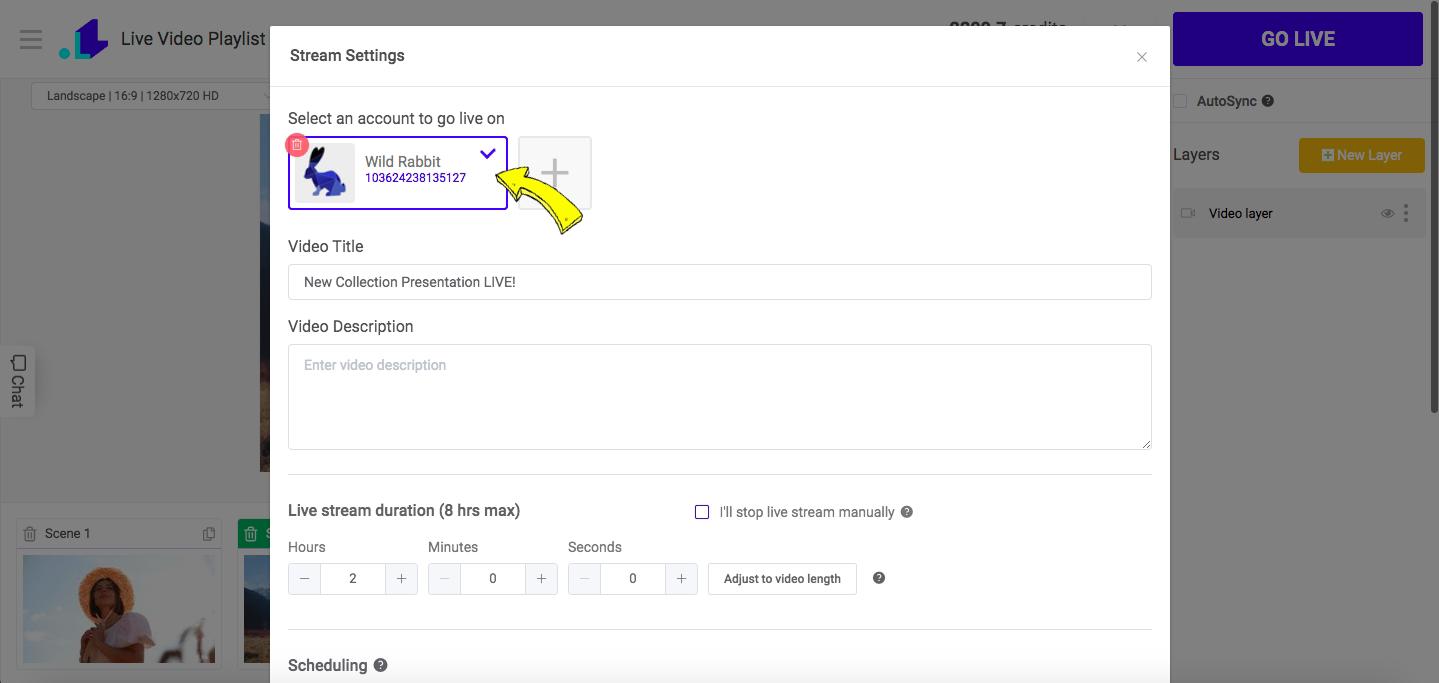
Intelligently restrict access to corporate video assets. Easily integrate with corporate directory systems for single sign-on, or use email authentication, enabling individual user tracking.

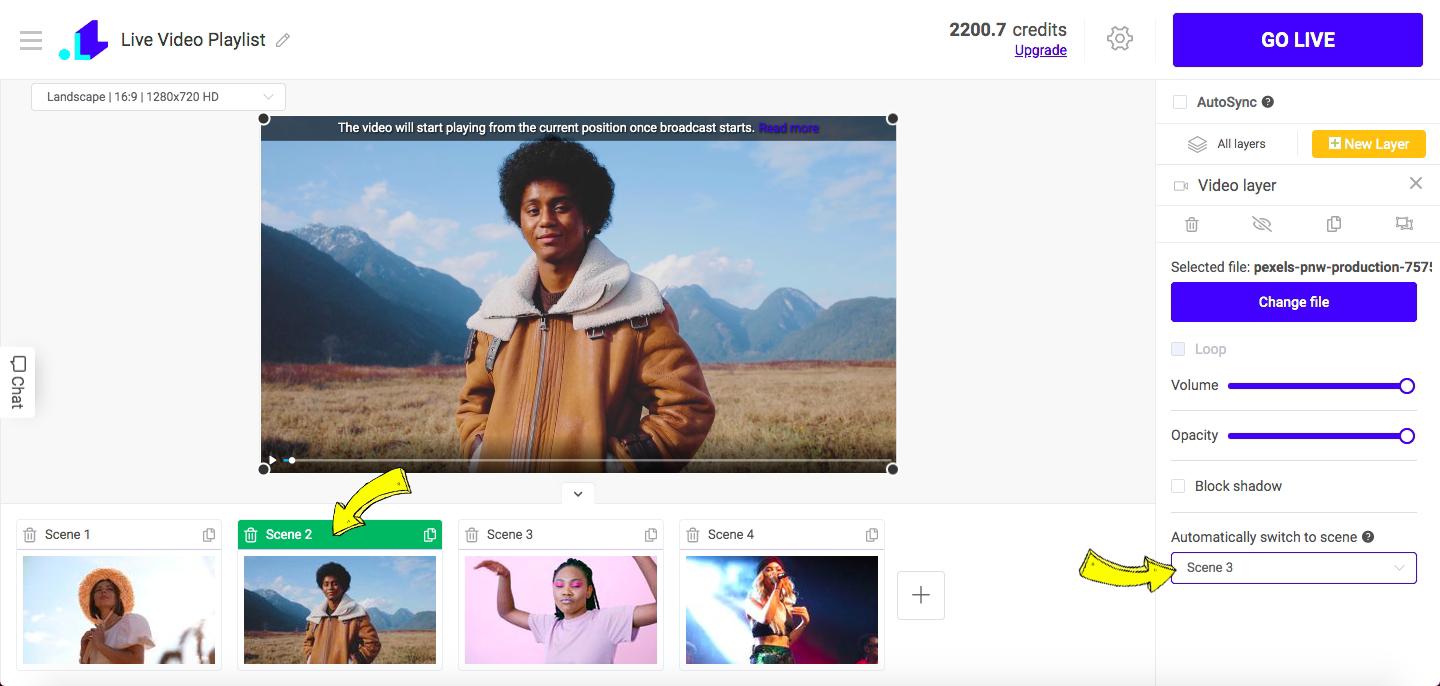
* **AI-driven deep video search**

Through automated, AI-driven transcription, video content can be searched at the library level, returning results based on keyword searches. Viewers can jump to specific moments based on AI insights.









**Video streaming 'backend ' python program**

# camera.py

import cv2

class VideoCamera(object):

def \_\_init\_\_(self):

# Using OpenCV to capture from device 0. If you have trouble capturing

# from a webcam, comment the line below out and use a video file

# instead.

self.video = cv2.VideoCapture(0)

# If you decide to use video.mp4, you must have this file in the folder

# as the main.py.

# self.video = cv2.VideoCapture('video.mp4')

def \_\_del\_\_(self):

self.video.release()

def get\_frame(self):

success, image = self.video.read()

# We are using Motion JPEG, but OpenCV defaults to capture raw images,

# so we must encode it into JPEG in order to correctly display the

# video stream.

ret, jpeg = cv2.imencode('.jpg', image)

return jpeg.tobytes()

# main.py

from flask import Flask, render\_template, Response

from camera import VideoCamera

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

return render\_template('index.html')

def gen(camera):

while True:

frame = camera.get\_frame()

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video\_feed')

def video\_feed():

return Response(gen(VideoCamera()),

mimetype='multipart/x-mixed-replace; boundary=frame')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', debug=True)