ature is in a pre-release state and might change or have limited support. For more information, see the product launc
s (/products/#product-launch-stages).

The Video Intelligence Streaming API supports standard live streaming protocols like RTSP, RTMP, and HLS. The AIStreamer (https://github.com/google/aistreamer/tree/master/ingestion) ingestion pipeline behaves as a streaming proxy, converting from live streaming protocols to bidirectional streaming gRPC connection.

To support live streaming protocols, the Video Intelligence uses the gStreamer (https://gstreamer.freedesktop.org/) open media framework.

A named pipe is created to communicate between gStreamer and the AIStreamer ingestion proxy. The two processes are running inside the same Docker container.

C++ examples (https://github.com/google/aistreamer/tree/master/ingestion/client/cpp) are available for you to use. The examples include a single binary (https://github.com/google/aistreamer/blob/master/ingestion/client/cpp/BUILD) that supports all features. To build the examples, see the build instructions (#build_instructions).

The following example shows how to use the binary from the command line.

Here, `$GOOGLE_APPLICATION_CREDENTIALS` specifies the file path of the JSON file that contains your service account key.

You can find an example configuration file—`$CONFIG` in the previous example here at github (https://github.com/google/aistreamer/tree/master/ingestion/client/cpp/config).

Make sure to set the correct timeout flag in the command line. If you need to stream 1 hour of video, timeout value should be at least 3600 seconds.

gStreamer supports multiple live streaming protocols including but not limited to:

- HTTP Live Streaming (HLS)
- Real-time Streaming Protocol (RTSP)
- Real-time Protocol (RTP)
- Real-time Messaging Protocol (RTMP)
- WebRTC
- Streaming from Webcam

The Video Intelligence uses the gStreamer pipeline to convert from these live streaming protocols to a decodable video stream, and writes the stream into the named pipe created in Step 1.

The following examples demonstrate how to use live streaming library using HLS, RTSP and RTMP protocols.

You can choose any muxer that supports streaming (for example, flvmux or mpegtsmux)

The binary example (https://github.com/google/aistreamer/blob/master/ingestion/client/cpp/BUILD) is built using Bazel (https://bazel.build). A Docker example (https://github.com/google/aistreamer/blob/master/ingestion/env/Dockerfile) that has all build dependencies configured is also provided. You can find the compiled `streaming_client_main` binary in the `$BIN_DIR` directory of the Docker image.

For more information on using Docker, see Using Docker & Kubernetes (/video-intelligence/docs/streaming/docker-kubernetes).

The Video Intelligence Streaming API server has inherent flow control.

`StreamingAnnotateVideoRequest` (/video-intelligence/docs/reference/rpc/google.cloud.videointelligence.v1#google.cloud.videointelligence.v1.StreamingAnnotateVideoRequest)

requests are rejected, and gRPC streaming connections are stopped immediately in the following two cases:

- The AIStreamer ingestion client is sending requests to Google servers too frequently.
- The AIStreamer ingestion client is sending too much data to Google servers (beyond 20Mbytes per second).