

Effective IP Camera Video Surveillance With Motion Detection and Cloud Services

Veeramani Ganesan
Mobile and OTT Engineer
New Jersey, United States of America

Abstract:- In comparison to its analog forerunners, the digital IP video surveillance system offers a wealth of features and capabilities, but it still faces significant difficulty in building and deploying intelligent and cost-effective systems. The current study is mostly focused on determining the system's uses. This research study provides a concept and implementation method for successful cloud-based video surveillance that makes use of cloud computing features like parallel processing, large storage spaces, scalability, and motion detection. The system can be built using Amazon cloud services, RTSP protocol, and the open-source Motion service SDK to implement video input processing, real-time picture analysis, storage, live streaming, and playback processes. A mobile web application can be used to see the video stream data and image analysis results remotely. This is a low-cost, highly portable, and easily extensible system with increased data security and minimal maintenance requirements.

Keywords:- Video Surveillance; Motion Detection; IP Camera Cloud Storage; Cloud Camera Systems; IP Camera RTSP Streaming; Mobile Web Apps; Live Streaming.

I. INTRODUCTION

IP video technology enables flexible and scalable video monitoring for a wide range of enterprises and applications. With an IP-based video surveillance system, users may remotely watch, record, and analyze video using an IP network as the system's backbone. However, their use is constrained in terms of cost. For successful video recording, image processing, and storage, more memory space, better computing power, and efficient power dissipation are all criteria that may be satisfied with cloud services and motion detection. Cloud-based NVR IP camera solutions offer considerable cost, security, storage, and maintenance benefits over traditional NVR and DVR IP camera systems. This paper demonstrates how to build more effective IP camera video surveillance systems by leveraging Amazon cloud services such as EC2 for computation, S3 for storage, and DynamoDB for the database. Also details the motion detection, image processing and live streaming implementation utilizing Motion services SDK and a React native mobile web application for configuration settings, live streaming, and playback. The paper presented here is organized as follows.

- Overview of the system architecture.
- Briefs about the Amazon EC2 instance and EBS volume selection.
- Describes how to set up S3 and integrate DynamoDB for configuration management.
- Describes how to establish a daemon process for motion detection, image processing, and web streaming using the open-source Motion SDK.
- Explains daemon process implementation for recording and storing streams to S3 storage.
- Hosting a mobile web application for configuration, live streaming, and replay.
- Future Scope.
- Conclusion.

II. OVERVIEW OF SYSTEM ARCHITECTURE

The system architecture for effective IP camera video surveillance described in this work is based on motion services, cloud computing and storage. Amazon EC2, S3 Bucket, DynamoDB, RTSP Stream Manager, Configuration Manager, and React Mobile Web Application comprises the architecture. Fig 1 depicts the overall architecture of the system, with each component playing a distinct role in the process of video acquisition, analysis, storing, and viewing video data from IP cameras.

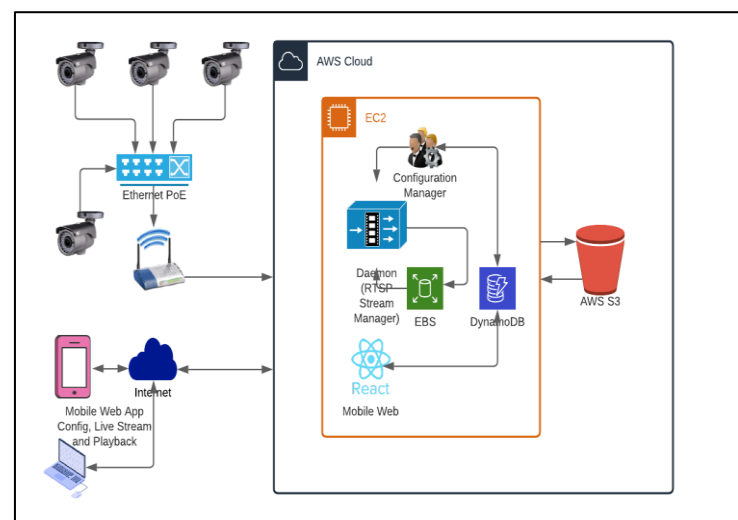


Fig. 1. System Architecture

III. SELECTION OF AMAZON EC2 AND EBS

Amazon EC2 provides a variety of instance types with varying CPU, memory, storage, and networking capabilities. For this project, we may use T2 as the general-purpose instance. This instance provides a base level of CPU performance with the flexibility to burst to a higher level as needed depending on our work demand. T2 Unlimited instances will provide adequate performance for the majority of general-purpose workloads without any additional expenses.

| Instance | vCPU* | CPU Credits / hour | Mem (GiB) | Storage | Network Performance |
|-----------|-------|--------------------|-----------|----------|---------------------|
| t2.nano | 1 | 3 | 0.5 | EBS-Only | Low |
| t2.micro | 1 | 6 | 1 | EBS-Only | Low to Moderate |
| t2.small | 1 | 12 | 2 | EBS-Only | Low to Moderate |
| t2.medium | 2 | 24 | 4 | EBS-Only | Low to Moderate |
| t2.large | 2 | 36 | 8 | EBS-Only | Low to Moderate |

Fig. 2. Example EC2 Instances

For volume storage, Amazon EBS Solid State Drives (SSD) can be chosen. Amazon EBS offers a variety of choices for optimizing storage performance and cost for our workload. Ubuntu server is the operating system of choice here. This is open-source software that works well with almost any hardware or virtualization platform.



Fig. 3. Example Ubuntu Server Amazon Machine Image

IV. AMAZON S3 SETUP AND DYNAMODB INTEGRATION

Amazon S3 storage can be used to archive video feeds from IP cameras for later video analysis and playback. S3 offers low-cost storage and makes it simple to manage and securely share video files across the internet. The AWS S3 console is used to construct the S3 buckets. Buckets are objects that can store and retrieve any volume of data from any location. S3 security may be managed using options such as restricting public access and enabling / disabling ACLs etc. Secure transmission can also be achieved by using server-side or client-side encryption utilizing the AES encrypted Key.

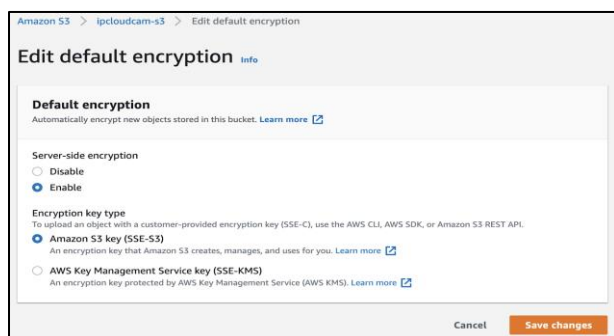


Fig. 4. S3 Security Settings

Amazon DynamoDB can be used to store IP camera configuration parameters and other user preferences for stream settings and image processing. It is a fully managed NoSQL database service for internet mobile web applications, with built-in security, backup and restore, and in-memory caching. DynamoDB console can be used to create the required tables with partition key, read / write capacity mode and encryption.

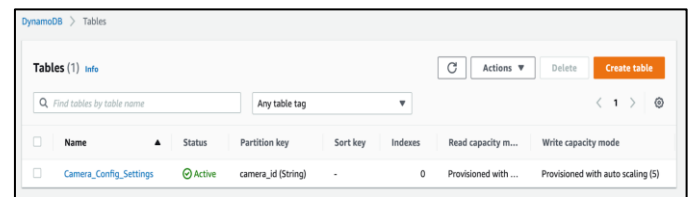


Fig. 5. Configuration Settings Table in DynamoDB

V. DAEMON PROCESS FOR MOTION DETECTION, IMAGE PROCESSING, AND STREAMING

The daemon process can be implemented to detect the motion in video streaming using image processing. The open-source c language-based Motion services SDK can be used to perform image analysis that can detect the movement from the camera RTSP streams. Also, we can set the threshold for the motion declaring. The threshold is the number of changed pixels counted after noise filtering, masking, despeckle, and labeling. If motion detects too many birds or moving trees, the threshold number can be adjusted accordingly. Practical values would be from a few hundred to thousands.

Motion is part of the Ubuntu repositories and can be installed with the apt tool as **sudo apt-get install motion**. It can be integrated with the daemon process and run using the configuration settings once installed. Configuration settings can include system control, image processing, motion detection, live stream, and camera parameters. Daemon process can be started or stopped using the **start-stop-daemon**.

Example Motion Configuration Settings as follows

1. System control configuration
daemon on
vid_control_params saturation=50, contrast=100
2. Network camera configuration
netcam_url rtsp://cameraip:port/camera/specific/url
3. Image processing configuration
width 640, height 480, frame rate 20
4. Motion detection configuration
emulate_motion off, threshold 1500, noise_level 32
5. Video output configuration
movie_output on, movie_quality 45, movie_codec mkv
6. Live stream configuration
stream_port 8081, stream_localhost off
7. Script execution configuration
on_event_start, on_event_end, on_movie_end
8. Web control configuration
webcontrol_port 8080, webcontrol_parms 0

VI. RECORDING AND STORING VIDEO STREAMS TO S3 STORAGE

The daemon process can record the RTSP streams from the camera using the motion movie output configuration such as `movie_filename`, `movie_max_time`, `movie_quality`, `movie_bps`, `movie_codec` etc. Motion supports wide variety of container formats such as `mp4`, `mkv`, `hevc`, `mpeg4`, `flv`, `mov`, `swf`, `mpeg4`. We can use `mkv` because it offers the Matroska container with the H264 codec. The H264 codec provides superior compression with high-quality video while consuming the least amount of bandwidth. The streams are momentarily recorded into the EBS volume and then migrated to the S3 on a regular basis. This frees up space in the EBS for more recording. The process uses the events such as

`on_motion_detected`, `on_movie_start`, `on_movie_end` for recording and transferring video streams into S3 storage. Streams can be moved in to the S3 using command `aws s3 mv ${basePath}/${file} s3://${bucket_name}/${cam_name}/${year}/${month}/${day}/${file}`

Fig 6 illustrates the system diagram for the RTSP stream manager, a daemon process that contains key components such as Motion Detection and Configuration Manager for image processing, Stream Receiver for receiving streams from the camera sources, Stream Recorder for recording video streams locally, Stream Transferer for sending streams to S3 storage, and Video Playback & Analytics for on-demand playback and live streaming.

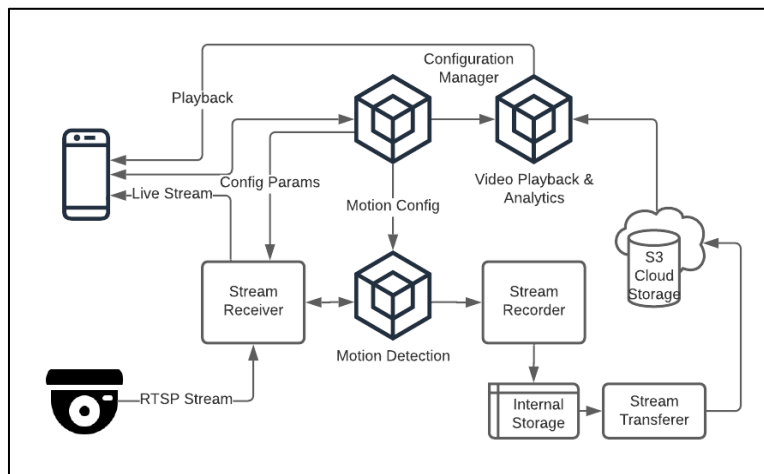


Fig. 6. Daemon Process System Diagram (RTSP Maanger)

VII. MOBILE WEB APP FOR LIVE STREAM AND PLAYBACK

The mobile web apps can be used to access the live stream and playback. It may also be used to give camera and motion detection settings for streaming, recording and image processing. This application can also be used to provide user settings. Mobile web apps are lightweight, cost effective, and simple to access in any operating system, utilizing browsers on both mobile and desktop computers. Mobile web apps can be built using ReactJS which has excellent cross platform compatibility and can be hosted on AWS, which delivers a low cost, dependable, scalable, secure, and high-performance infrastructure for the most demanding online applications. AWS has several alternatives for web app hosting, including EC2, S3, Amplify, LightSail, and more. For this work Amazon EC2 is the preferred solution since it is already available in our eco system with Ubuntu server and that can be an affordable environment to perform this task.

manager for Node.js applications, we will use it instead of using NPM start to run our application, SSH: Secure Shell used to login into EC2 Ubuntu server from remote computer using private and public keys, both are generated by AWS.

VIII. FUTURE SCOPE

The proposed system currently utilizes the H.265 codec for encoding, which may be changed with H.266 an emerging compression standard with 50% greater compression rates than H.265. This reduces bandwidth and storage space even further, lowering the cost. Using video analytics improves surveillance system efficiency, decreases security burden, and allows you to capture the full value of security footage by making your IP camera system more intelligent in its operation. Video analytics can be built with Amazon Rekognition Video, a cloud service that delivers easy-to-integrate and cost-effective analytics tools. Mobile web apps can be upgraded with Amazon Amplify for quicker development, simpler deployment, and greater app performance while only paying for the services used. React web applications can be developed with the open source Amplify framework, which includes the Amplify CLI (Command Line Interface), Amplify Libraries, and Amplify UI Components, making app development considerably easier and faster. Amplify allows both manual and automatic continuous deployment using CI/CD pipeline.

IX. CONCLUSION

Cloud-based video surveillance is a smart solution that allows you to secure your property or business while maintaining complete control over what is going on. Data from surveillance cameras is available at any time, and it is kept for a set amount of time and may be downloaded at any time. Cloud-based video surveillance, unlike traditional systems, is incredibly cost-effective, easily extendable, and available from anywhere. By incorporating motion-based recording into the cloud video surveillance system reduces both bandwidth and recording storage. React mobile web apps are faster and cheaper to build compared to their native ones without the need to compromise on quality. Money is saved as a result, while functionality is maintained. This paper explained how to develop and deploy cloud-based video surveillance using Amazon's cloud services, motion detection SDK and mobile web application, resulting in an extremely effective IP camera video surveillance system.

ACKNOWLEDGMENT

This study is an effective addition to the field of IP camera video surveillance systems by its author. The work done here is anticipated to modify the system's present trend by switching to this strategy.

REFERENCES

- [1]. AWS Amazon Web Services 2020: Step by Step Guide to Master AWS by Ezra Sabet
- [2]. IP Video Surveillance. An Essential Guide by Alexander Lytkin
- [3]. RAMESH, D.RAJA, Recent Trends and Literature Review on Enhanced Video Compression by Using H.265/HEVC Standard (July 15, 2020). International Journal for Innovative Engineering & Management Research, 2020, Available at SSRN: <https://ssrn.com/abstract=3652049>
- [4]. Schulzrinne H., Rao A., Lanphier R. RTSP: Real Time Streaming Protocol. [(accessed on 12 April 2017)]; IETF RFC 2326 (1998) online: <https://tools.ietf.org/html/rfc2326>
- [5]. Motion: https://motion-project.github.io/motion_config.html#top
- [6]. Amazon EC2: <https://aws.amazon.com/ec2/>
- [7]. Amazon S3: <https://aws.amazon.com/s3/>
- [8]. Node JS: <https://nodejs.org/api/>
- [9]. <http://manpages.ubuntu.com/manpages/focal/man7/damon.7.html>
- [10]. <http://manpages.ubuntu.com/manpages/focal/man1/motion.1.html>
- [11]. React: A JavaScript Library for Building User Interfaces. URL: <https://facebook.github.io/react/>
- [12]. Amazon Amplify: <https://aws.amazon.com/amplify/>
- [13]. React and Amplify: <https://aws.amazon.com/getting-started/hands-on/build-react-app-amplify-graphql/>