Video Ingestion & Inference (Take-Home)

Overview

You will build an inference pipeline to process video frames and run inference in AWS. Specifically:

1. **RTSP Video Source** → Kafka
2. **Kafka Consumer** (batch of 25 frames) → Kubernetes-based inference
3. **Inference Output** (bounding boxes) → Post-processed and uploaded to S3

Instructions

1. **Create a New AWS Account**

   - Create a fresh AWS account.
2. **AWS Stack**

   - Stand up a minimal **Kafka** environment (MSK).
   - Deploy a **Kubernetes** cluster (EKS).
   - Use **Infrastructure-as-Code** (Terraform, CloudFormation, Pulumi, etc.) **as much as possible**.
   - We understand that certain things will be easier done through the Console. Hence, you may do those steps via the Console; we prefer seeing your K8s manifests and any IaC scripts that can replicate the setup at scale.

3. **Video Ingestion**

   - Use a local RTSP server deployed on t3.micro instance to stream a demo video (e.g., a looping MP4).
   - For reference, you can run:

None

```
docker run --rm -p 8554:8554 \
  -v /path/to/video.mp4:/media/video.mp4 \
  aler9/rtsp-simple-server
```

- Publish frames to Kafka (one topic per video stream).
- You must batch frames in groups of **25** before sending them to the inference service.

-Inference Pipeline

- Containerize a minimal object detection or classification model (CPU-based).
- Deploy it to EKS (Deployment + Service).
- Your consumer service (outside or inside K8s) should call this inference service with each batch of 25 frames.
- For **post-processing**, draw bounding boxes (or relevant annotation) on at least one frame per batch, then upload the annotated image to an S3 bucket/folder for verification. This should **not** be done in the container hosting the model.
- ***BONUS*****:** Autoscale the inference Deployment on **Kafka lag**.

- .

-Deliverables

- **Code Repository**: Provide with all code, Dockerfiles, IaC scripts, and K8s manifests. ***BONUS:*** CI/CD pipelines for deployment. This should NOT be just a .sh script - a true CI/CD pipeline.