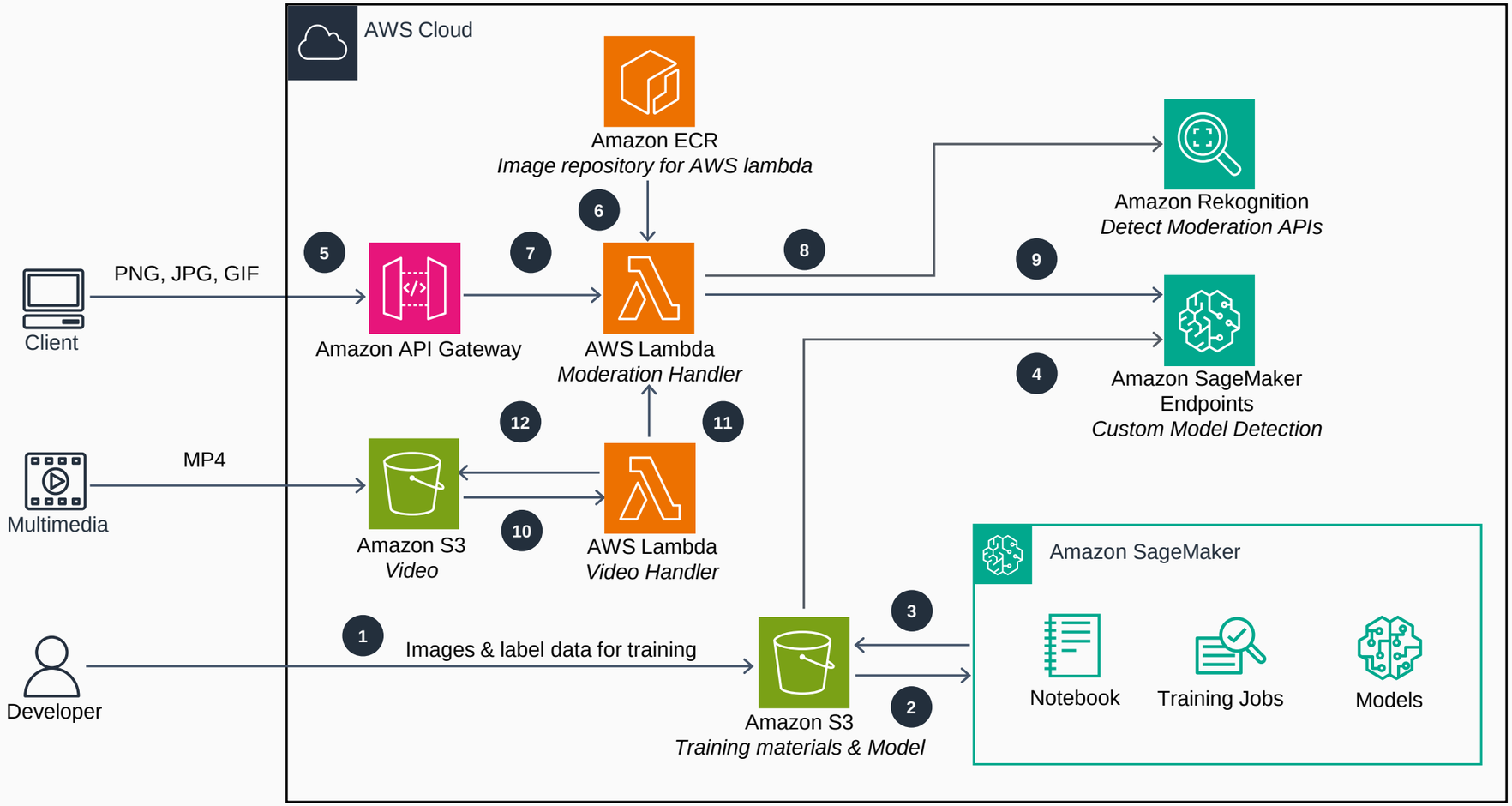
Guidance for Responsible Content Moderation with AI Services on AWS

This architecture diagram shows how game developers can moderate user-generated content (UGC) such as images and texts with AWS managed services and custom machine learning (ML) models. Through well-designed APIs for quick integration with existing upstreaming applications, developers can monitor illegal or unhealthy content and reduce the risks inherent with social networking operations.



1. The developer prepares the images and label data in **Amazon Simple Storage Service (Amazon S3)**.
2. **Amazon SageMaker** notebooks and training jobs use training materials to train the custom model.
3. On completion, training jobs output the custom model in the **Amazon S3** bucket.
4. The developer deploys the model to a **SageMaker** endpoint for a custom moderation handler.
5. **Amazon API Gateway** receives images for image moderation requests.
6. **Amazon Elastic Container Registry (Amazon ECR)** hosts a container image repository for **AWS Lambda**.
7. The moderation handler in **Lambda** processes the request based on the parameters in the URL and merges the results from the handlers.
8. The moderation handler distributes the request to **Amazon Rekognition** based on the image moderation labels.
9. The moderation handler distributes the request to the **SageMaker** endpoint based on the image moderation labels.
10. (Optional) A video file is uploaded to **Amazon S3** to invoke the video handler in **Lambda.**
11. (Optional) **Amazon S3** stores the key frame images from videos and starts the invocation of the moderation handler to process the detection of the images.
12. (Optional) The video handler processes the moderation result and returns it to the original **Amazon S3** bucket.

Reviewed for technical accuracy February 22, 2024

AWS Reference Architecture