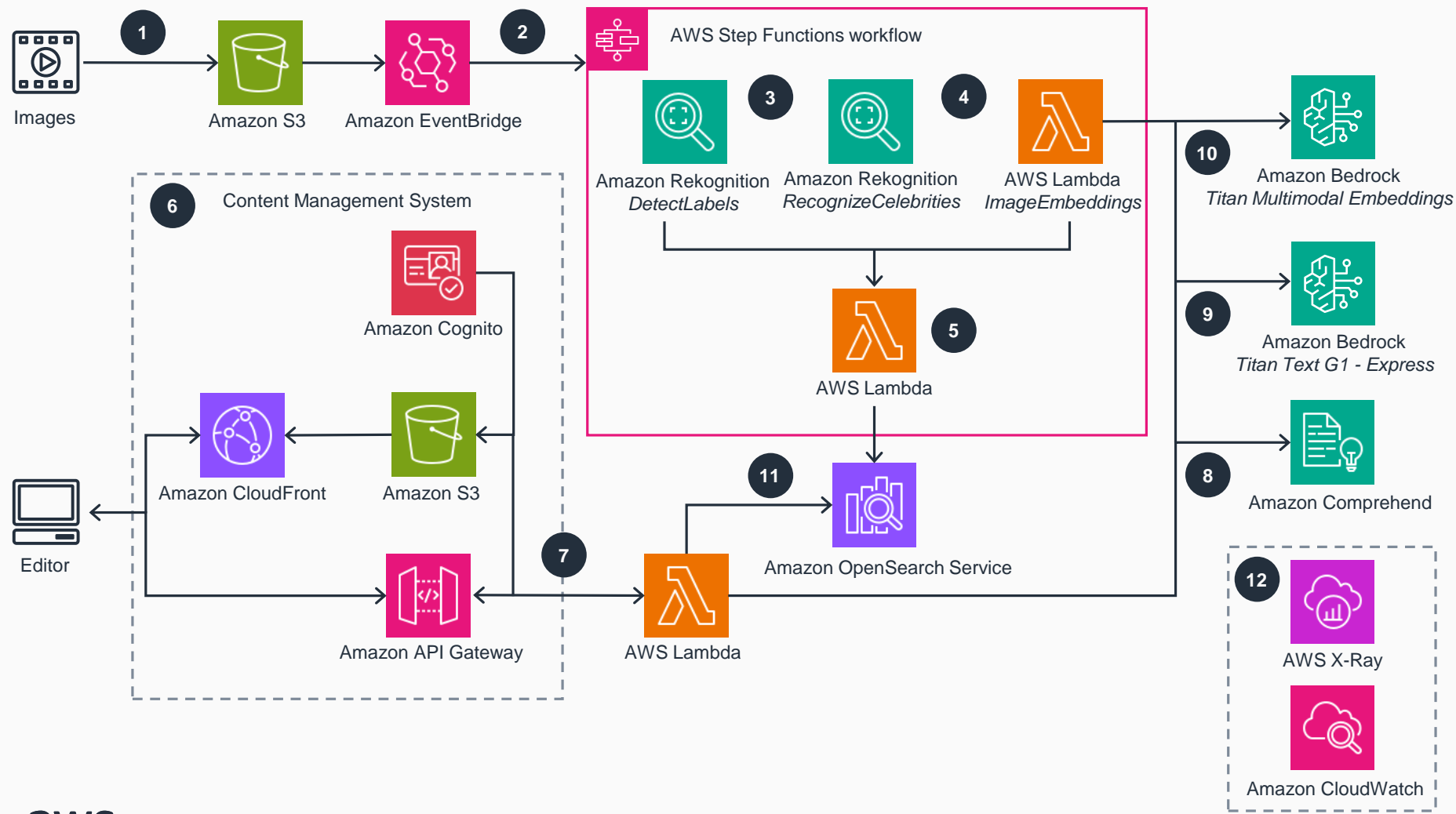


Guidance for Automatic Image Matching for Articles Using Machine Learning on AWS

This architecture diagram shows how to generate embeddings of images and use an editor to find semantically similar images. This slide details Steps 1-5, showing the process of image ingestion to populate the image index.

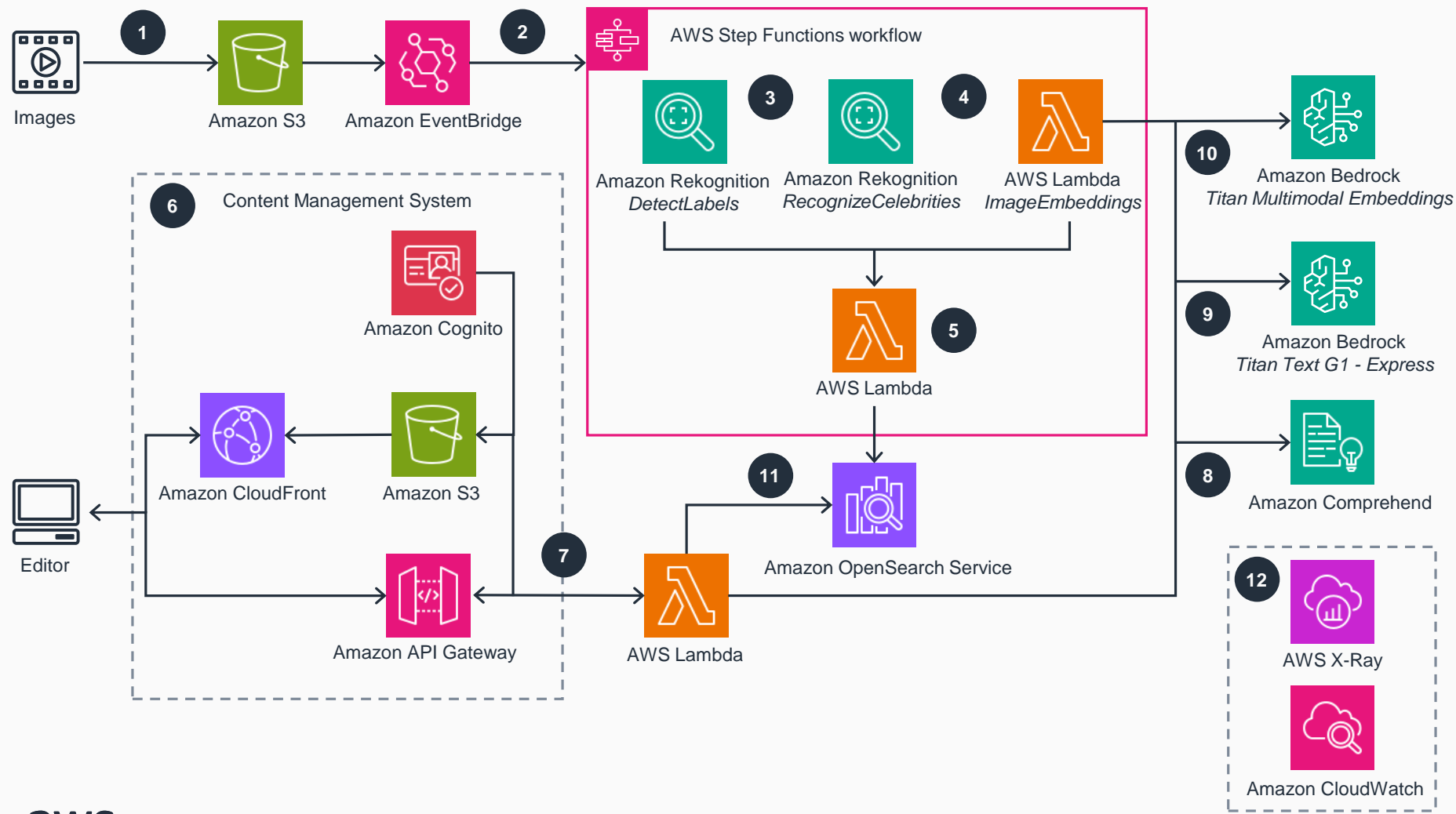


- 1 Preload **Amazon OpenSearch Service** with searchable metadata by uploading images to **Amazon Simple Storage Service (Amazon S3)**.
- 2 **Amazon EventBridge** listens for the event and initiates an **AWS Step Functions** workflow.
- 3 **Amazon Rekognition** extracts Label and Celebrity metadata from the image.
- 4 **AWS Lambda** invokes the **Amazon Bedrock Amazon Titan Multimodal Embeddings** model to generate an embedding of the image itself.
- 5 The next **Lambda** function stores the image extracted metadata and embedding in **OpenSearch Service** for easy search.



Guidance for Automatic Image Matching for Articles Using Machine Learning on AWS

This architecture diagram shows how to generate embeddings of images and use an editor to find semantically similar images. This slide details Steps 6-12, showing how to search for images with articles.



- 6 The content management system resembles the journalist environment. **Amazon S3** hosts a simple static website for the semantic image search workload, served by an **Amazon CloudFront** distribution. **Amazon Cognito** protects the web application from unauthenticated users.
- 7 After the editor has written an article, they select the search button to invoke an **Amazon API Gateway** API to the backend. The backend will extract names, summarize the article, and generate an embedding which is used as part of the search query against **OpenSearch Service**.
- 8 **Lambda** invokes **Amazon Comprehend** with the article text to detect Person entities from the article.
- 9 **Lambda** invokes the **Amazon Bedrock Amazon Titan Text G1 - Express** model to derive and order the Person names by prominence and summarize the article to meet the max prompt token count of the multimodal embeddings model.
- 10 **Lambda** invokes the **Amazon Bedrock Amazon Titan Multimodal Embeddings - G1** model to generate an embedding of the summarized article.
- 11 **Lambda** performs a query against **OpenSearch Service** using the embedding of the summarized article and the name of the most prominent Person, using exact k-nearest neighbors (k-NN) with scoring script. This returns the top semantically related images with a given celebrity (if present for the given article).
- 12 **Amazon CloudWatch** and **AWS X-Ray** provide observability into the health of the Guidance.