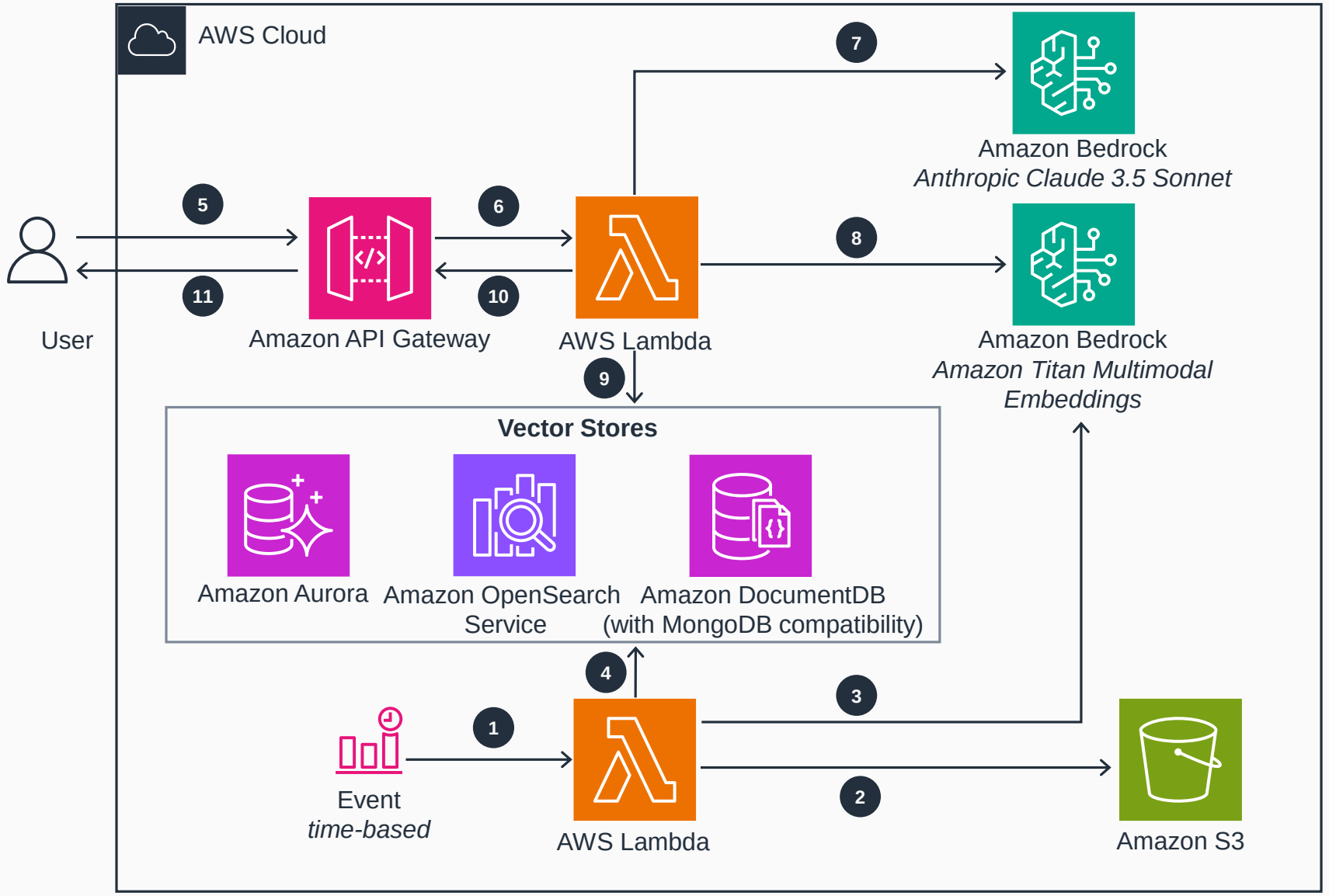
Guidance for Visual Search on AWS

This architecture diagram helps build a simple visual search capability, allowing your users to upload a product image and discover similar looking products from the ecommerce website.



1. A time-based **Amazon EventBridge** scheduler invokes an **AWS Lambda** function to populate search indexes with multimodal embeddings and product meta-data.
2. The **Lambda** function first retrieves product feed stored as a JSON file in **Amazon Simple Storage Service (Amazon S3).**
3. The **Lambda** function then invokes the Amazon Titan Multimodal Embedding G1 model hosted on **Amazon Bedrock** to create vector embeddings for each product in the catalog. These embeddings are created based on the primary image and product description for each item in the product catalog.
4. The **Lambda** function finally persists these vector embeddings as k-nearest neighbor (k-NN) vectors, along with product meta-data in the vector store, such as **Amazon OpenSearch Service**, **Amazon DocumentDB**, or **Amazon Aurora**. This index is used as the source for semantic image searches.
5. The user initiates a visual search request through a frontend application by uploading a product image.
6. The application uses the **Amazon API Gateway** REST API to invoke a pre-configured proxy **Lambda** function to process the visual search request.
7. *Optional step for better search results:* The **Lambda** function first generates the caption for the input image using the **Anthropic Claude 3.5 Sonnet** model hosted on **Amazon Bedrock**.
8. The **Lambda** function then invokes the Titan Multimodal Embeddings model. This model generates a multimodal embedding based on the input image uploaded by the user and the image caption, if one was generated in step 7.
9. The **Lambda** function then performs a k-NN search on the vector store index to find semantically similar results for the embedding generated in step 8.
10. The resultant semantic search results retrieved from the vector store are then filtered to eliminate any duplicate entries. The filtered results are then enriched with product metadata from the search index before being passed back to the **API Gateway**.
11. Finally, the response from **API Gateway** is returned to the client application, which then displays the search results to the end user.

Reviewed for technical accuracy July 5, 2024

AWS Reference Architecture