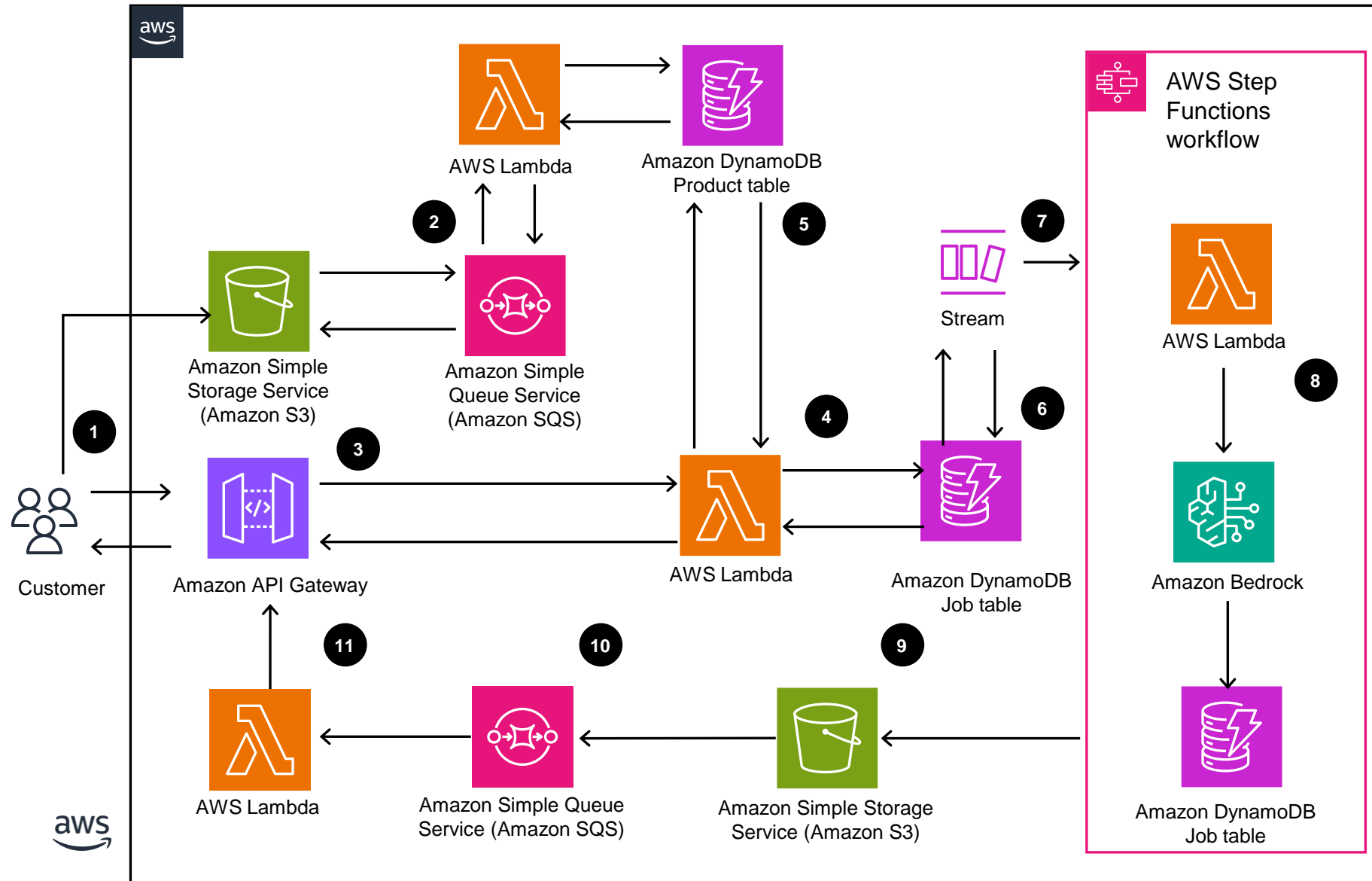


# Guidance for Virtual Try-On on AWS

This architecture diagram shows how to use the Amazon Nova Canvas model for Virtual Try-On (VTO) use cases, combined with AWS serverless architecture to provide accurate, fast product try-on experiences while maintaining scalability and reliability for production environments.



- 1 Amazon S3 stores the uploaded customer model photos and product images.
- 2 Each upload generates a message sent to an Amazon SQS queue. Lambda creates the corresponding metadata, S3 path, and stores it in the Amazon DynamoDB product table for later retrieval.
- 3 Amazon API Gateway manages the WebSocket connections for real-time status updates between the client and the VTO.
- 4 AWS Lambda processes initial requests by retrieving product information from the Amazon DynamoDB product table and creating job entries in DynamoDB.
- 5 Amazon DynamoDB: The products table (vto-products) stores catalog items available for the VTO, notably the Amazon S3 picture location.
- 6 The Virtual Try-On Jobs DynamoDB Table (vto-jobs) tracks the state of each try-on request.
- 7 Amazon DynamoDB Streams asynchronously triggers AWS Step Functions workflows on job creation for processing try-on requests.
- 8 AWS Step Functions orchestrates the VTO generation. It triggers a Lambda function that calls the Amazon Nova Canvas model through Amazon Bedrock. The DynamoDB job table is updated with the VTO status.
- 9 Amazon S3 stores the generated try-on images with job ID metadata.
- 10 Amazon SQS handles Amazon S3 event notifications for completed try-on images.
- 11 AWS Lambda Function sends the Amazon S3 URL of the result back to the user via WebSocket.