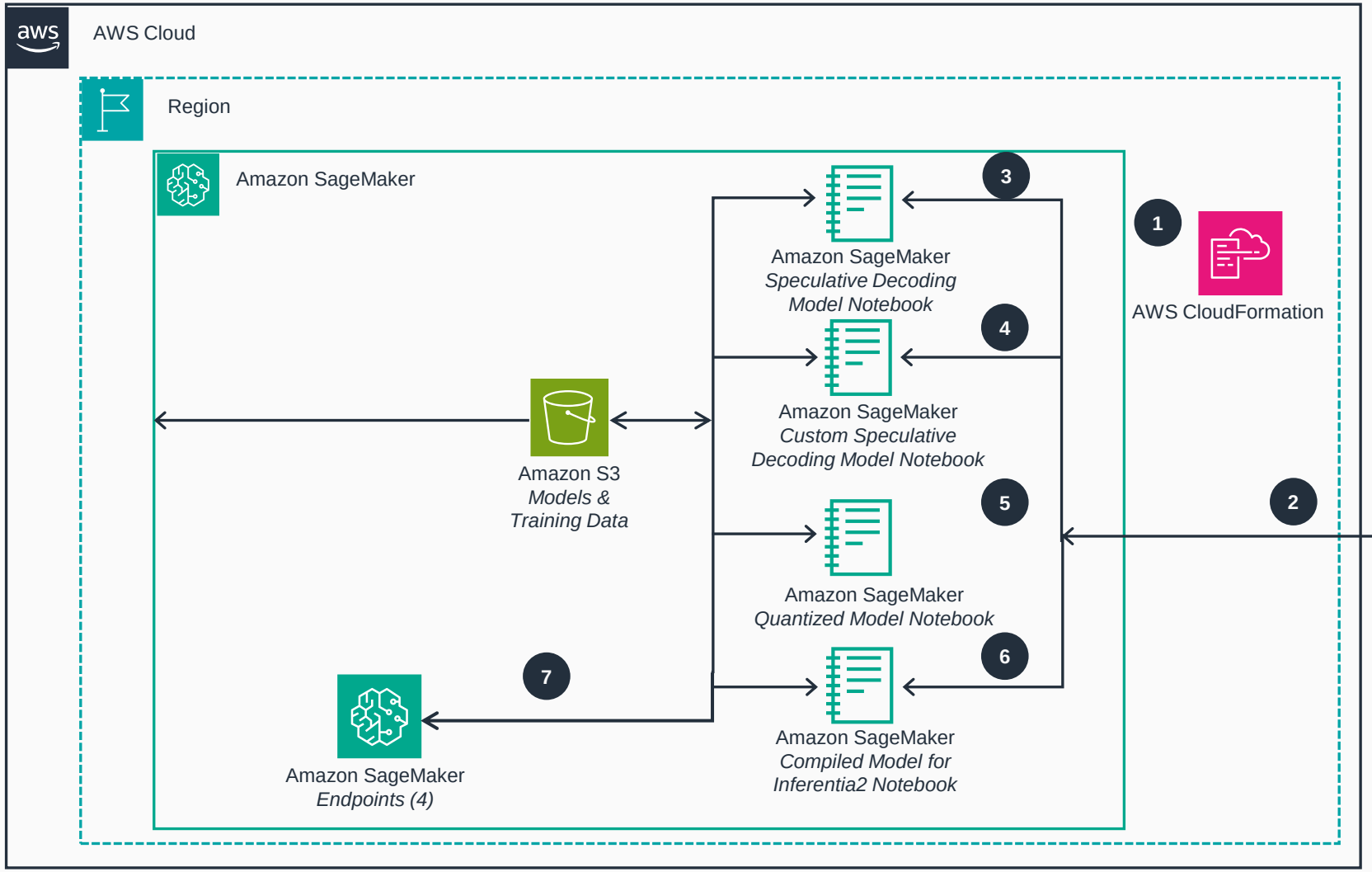
Guidance for Generative AI Model Optimization Using Amazon SageMaker

Optimization for data scientists

This architecture diagram shows how data scientists can optimize Large Language Models (LLMs) within Amazon SageMaker to deliver responses that are not only faster, but also more accurate and cost-effective. The subsequent slide outlines the deployment of the optimized LLMs in Amazon SageMaker.



1. Deploy the optimization toolkit with the **AWS CloudFormation** template to your AWS account.
2. Run each of the four Python-based Jupyter notebooks within **Amazon SageMaker Studio** to configure, optimize, and test the selected foundation model. Host the models and the training data in the **Amazon Simple Storage Service (Amazon S3)**, which serves as the default hosting location.
3. Fine-tune pre-optimized models in **Amazon SageMaker** using the Speculative Decoding Model notebook. This model accelerates inference on ml.p4d.24xlarge instances using a faster draft model to predict candidate outputs in parallel.
4. Improve inference speed by deploying a custom speculative decoding model using the Custom Speculative Decoding Model notebook on ml.p4d.24xlarge instances. Access draft models from the Hugging Face model hub or your own **Amazon S3** bucket.
5. Reduce model memory requirements through quantization, using the Quantized Model notebook on the ml.g5.12xlarge instance. This uses Activation-Aware Weight Quantization (AWQ) to shrink the memory footprint while maintaining quality, enhancing throughput, and reducing latency.
6. Compile and deploy your tuned models on performance-optimized **AWS Inferentia** hardware (ml.inf2.48xlarge instances) with the Compiled Model for Inferentia 2 notebook.
7. Deploy **SageMaker** endpoints to access your models from applications.

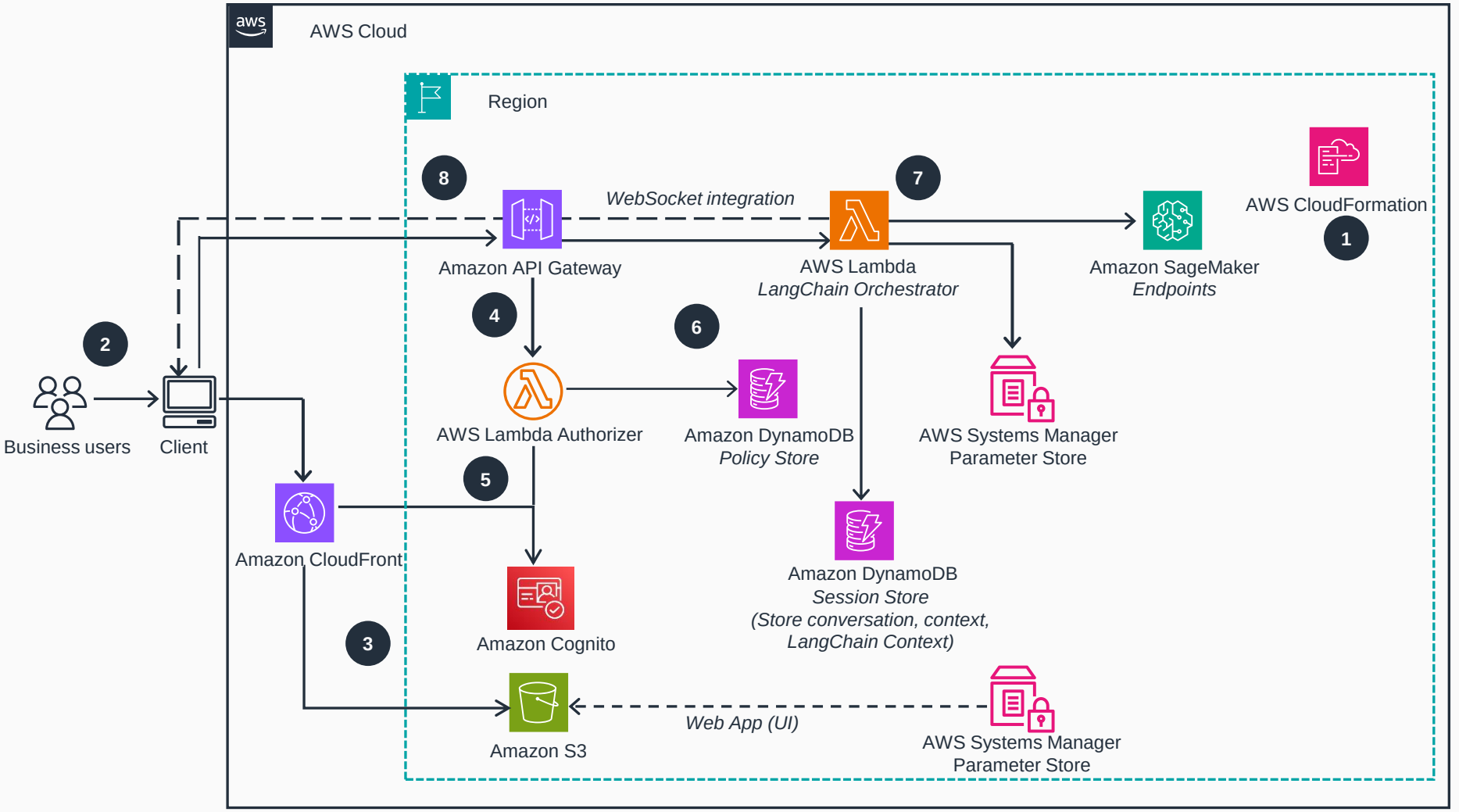
Reviewed for technical accuracy September 6, 2024

AWS Reference Architecture

Guidance for Generative AI Model Optimization Using Amazon SageMaker

LLM deployment in applications

This architecture diagram shows how to deploy the optimized LLMs in SageMaker, including using AWS CloudFormation to provision all the necessary application resources.



1. Deploy all application resources using **CloudFormation**, including **SageMaker** endpoints for interacting with optimized large language models (LLMs).
2. Log in through the application interface.
3. Host web UI content in **Amazon S3**. Deliver that content quickly to users in any location through **Amazon CloudFront.**
4. Provide a persistent WebSocket connection to the application with **Amazon API Gateway.**
5. Authenticate users through **Amazon Cognito** for every transaction conducted through **CloudFront** and **API Gateway.**
6. Implement business logic through LangChain Orchestrator functions in **AWS Lambda** to fulfill requests. The Orchestrator retrieves configured LLM options and session information from the Parameter Store, a capability of **AWS Systems Manager** and **Amazon DynamoDB**.
7. Using the chat history and query, the LangChain Orchestrator creates the final prompt and sends the request to the LLM hosted on **SageMaker** using Amazon SageMaker Jumpstarts.
8. Send the LLM response back to the user through **API Gateway**.

Reviewed for technical accuracy September 6, 2024

AWS Reference Architecture