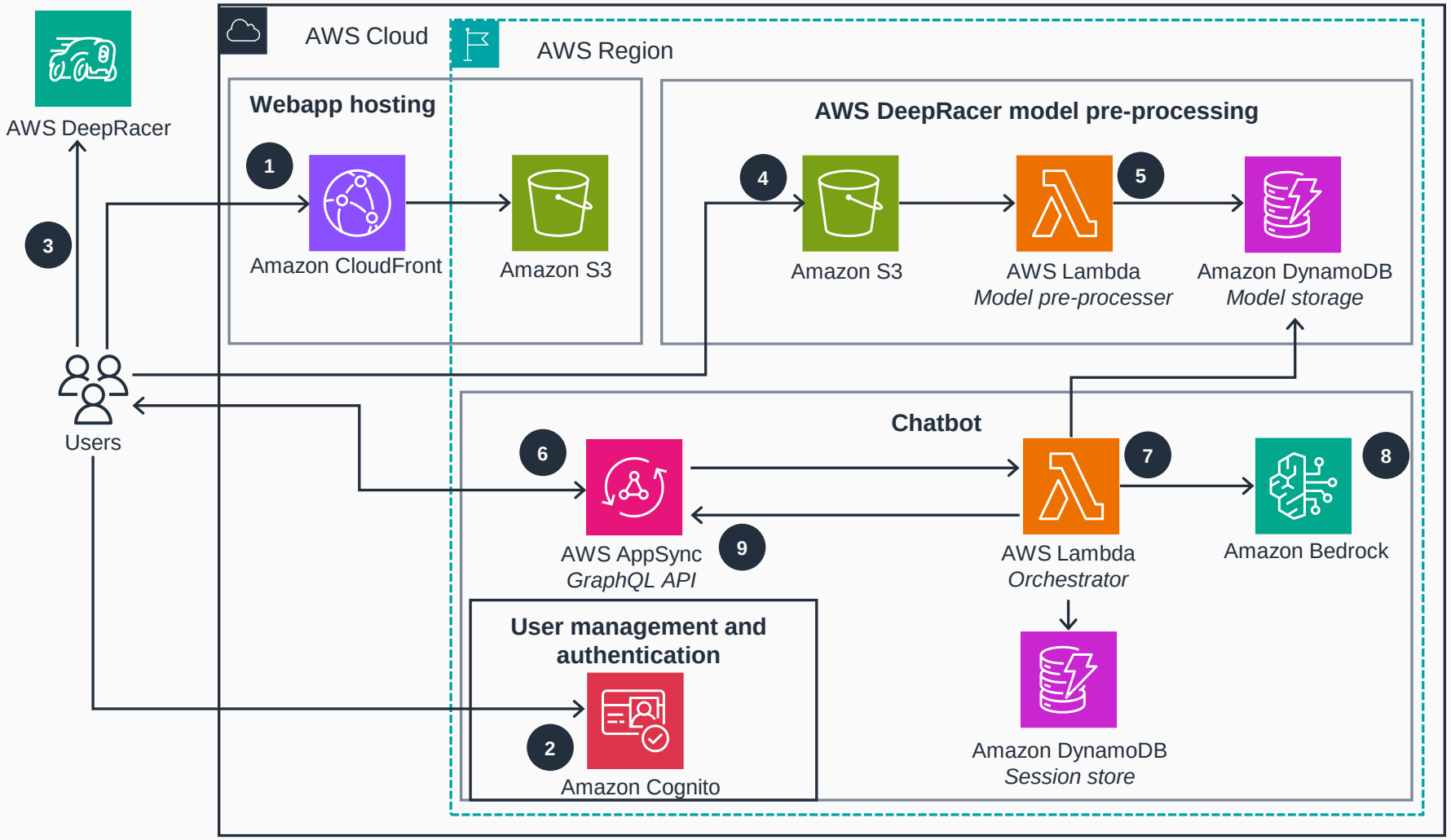
Guidance for Deploying an AWS DeepRacer Chatbot

This architecture diagram illustrates how to deploy a generative AI-powered AWS DeepRacer Chatbot. It shows the key components and their interactions, providing an overview of the architecture's structure and functionality



1. The users navigate to the **Amazon CloudFront** URL to fetch the **AWS DeepRacer** Chatbot webapp.
2. The users authenticate with **Amazon Cognito**.
3. The users export and download their trained **AWS DeepRacer** models to their laptop and store as a zip file. Refer to Import and export models in the AWS DeepRacer console for more details on how to export models.
4. The users upload the compressed model package to **Amazon Simple Storage Service (Amazon S3).**
5. When a new object is created in **Amazon S3**, an **AWS Lambda** function is invoked to download and parse logs inside the compressed model. The result is stored in an **Amazon DynamoDB** table and the uploaded model is deleted from **Amazon S3.**
6. When a user asks a question about the model, the message is sent to **AWS AppSync** with a **Lambda** resolver.
7. The **Lambda** function will fetch any previous conversation from the session store. It updates the prompt before forwarding the message to the **Amazon Bedrock** Converse API. It also uses function calling to fetch the parsed model from **DynamoDB.**
8. The large language model (LLM) in **Amazon Bedrock** will utilize the system prompt, model data, and the user's question to analyze the **AWS DeepRacer** model.
9. The model response is streamed through **Lambda** and **AWS Appsync** to the subscribed users.

Reviewed for technical accuracy November 14, 2024

AWS Reference Architecture