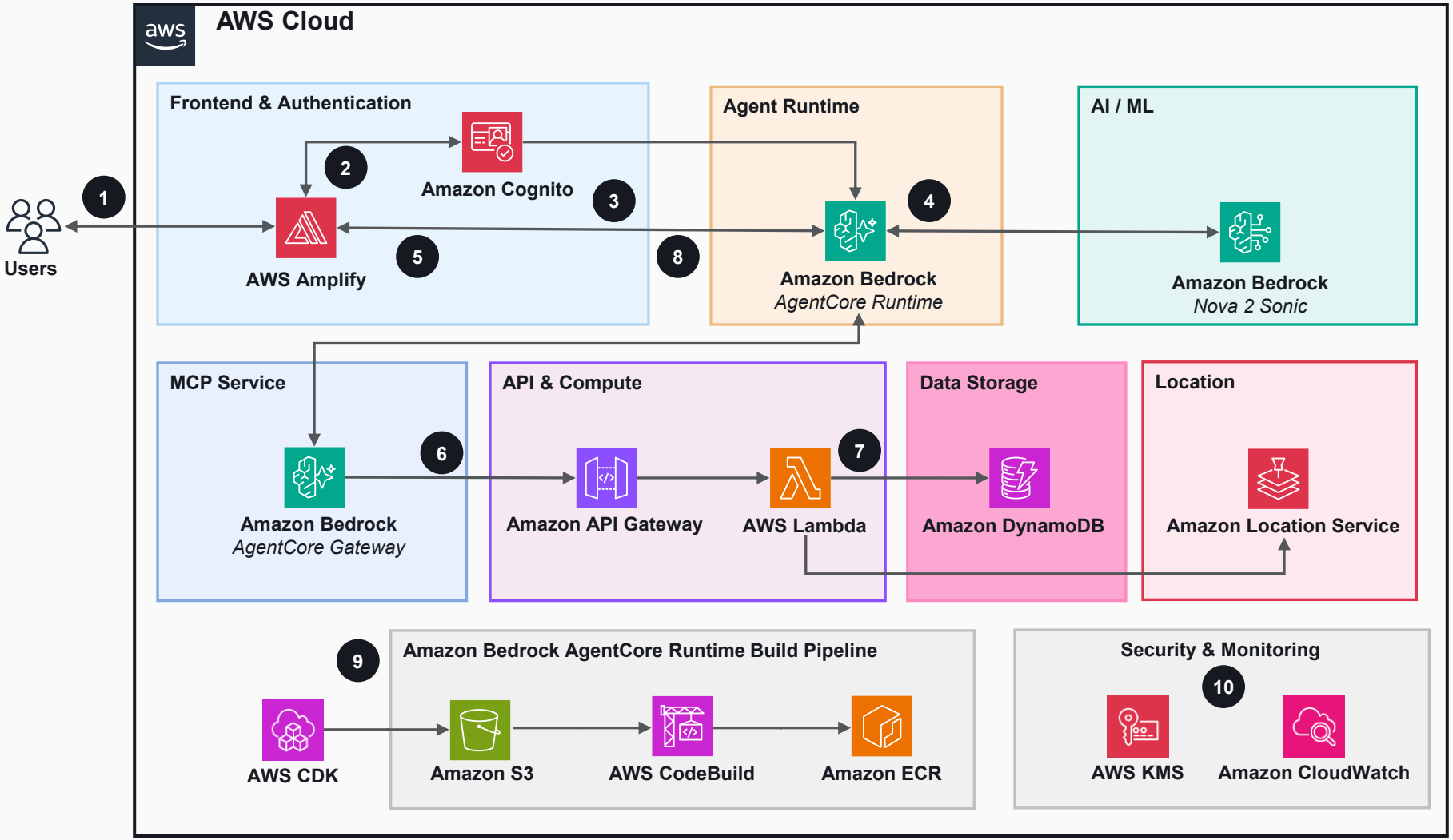
Guidance for Omnichannel Voice AI Ordering using Amazon Bedrock AgentCore

Customers expect fast, hands-free ordering, but building real-time voice AI that integrates with live backend data without conversational pauses requires complex infrastructure difficult to operate at scale. This architecture shows how to build a voice ordering system using Amazon Bedrock AgentCore and Nova 2 Sonic with MCP tool integration.



1. You access the web application hosted on **AWS Amplify** from a browser or mobile device.
2. You authenticate with **Amazon Cognito** using username and password to receive JWT tokens and temporary AWS credentials.
3. The frontend opens a SigV4-signed WebSocket connection to **Amazon Bedrock AgentCore Runtime** to begin the voice ordering session, establishing a persistent bidirectional channel for real-time voice streaming.
4. The runtime validates the token via **Amazon Cognito** and initializes **Amazon Nova 2 Sonic** through **Amazon Bedrock**, a fully managed service with built-in security, privacy, and responsible AI.
5. You speak your order into the device microphone. The agent processes voice through **Amazon Nova 2 Sonic** and invokes tool functions via **AWS Lambda** to manage your cart and retrieve menu items.
6. AgentCore Gateway converts **AWS Lambda** functions and APIs into MCP-compatible, agent-ready tools, routing tool invocations to **Amazon API Gateway** and the underlying AWS Lambda functions.
7. **AWS Lambda** functions query **Amazon DynamoDB** tables and **Amazon Location Service** for menu data, cart management, and store proximity.
8. **Amazon Nova 2 Sonic** generates a contextual voice response and streams it back to you over the WebSocket connection through AgentCore Runtime.
9. **AWS CDK** deploys the solution with a single script, uploading application code to **Amazon S3** and triggering **AWS CodeBuild** to build container images stored in **Amazon ECR** for the AgentCore runtime.
10. **Amazon CloudWatch** provides centralized monitoring, logging, and alerting across all services. All data at rest is encrypted using **AWS KMS**.

Reviewed for technical June 22, 2026

AWS Reference Architecture