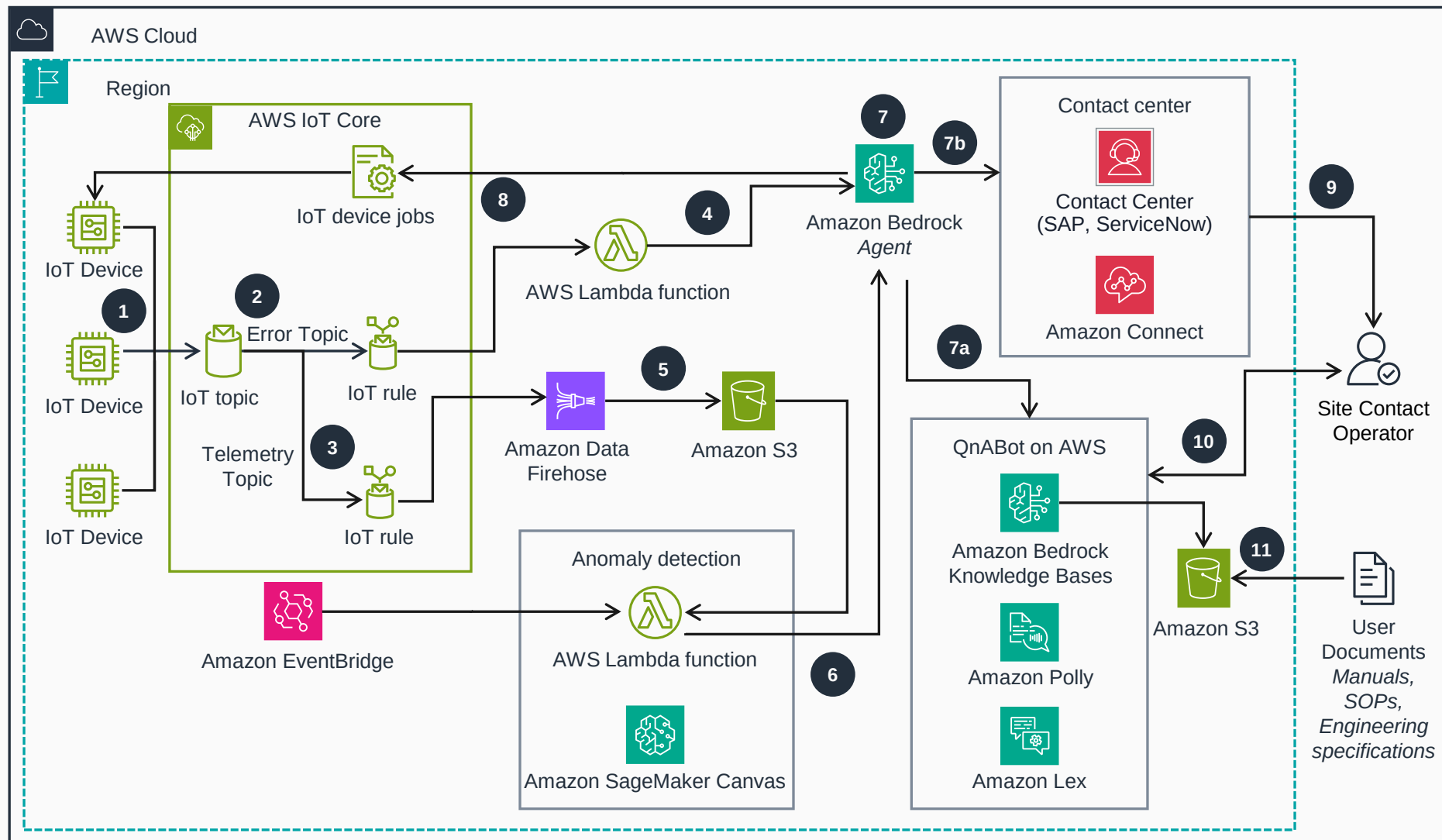


Guidance for Connecting Automated Inputs to Contact Centers on AWS

AI-Powered Automated Incident Detection and Response System Using Bedrock Agents

This architecture diagram shows how to enable real-time error, anomaly detection, automated incident resolution, and seamless contact center integration for IoT-connected systems. This servers to reduce downtime and optimize issue resolution through AI-driven workflows.



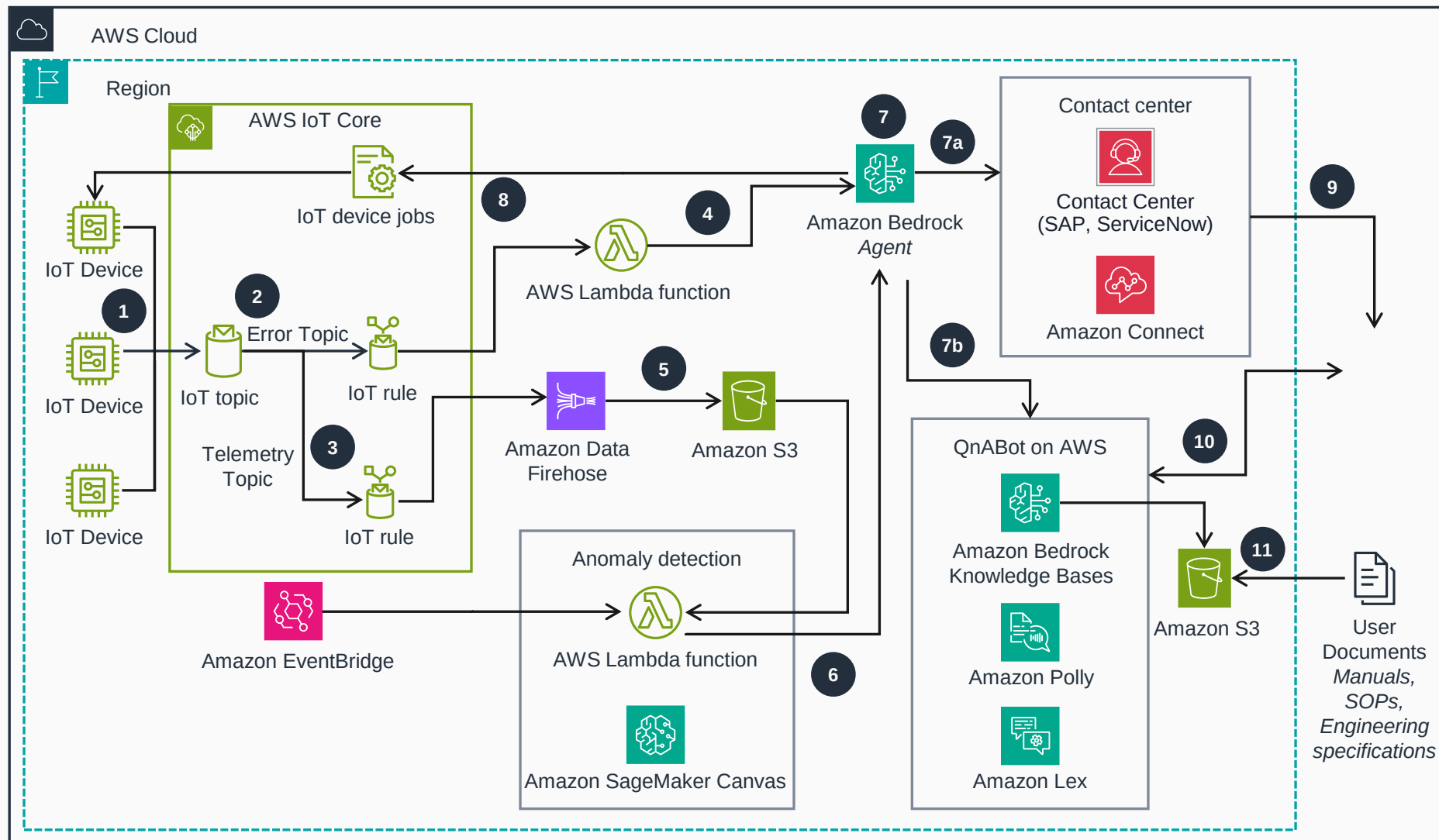
- 1 IoT devices publish to **AWS IoT Core**.
- 2 Errors or alerts generated by devices are published to designated IoT topics, such as an "error topic" or "basic ingest topic". These inputs are then routed to an **AWS Lambda** function based on the defined IoT rule.
- 3 Devices publish standard telemetry data to an IoT topic. This real-time data can be acted on, while a copy is also routed to a data store for the purposes of analytics and anomaly detection.
- 4 The **Lambda** function transmits the error code, device ID, and any additional relevant information to **Amazon Bedrock Agents**.
- 5 The data is streamed through **Amazon Data Firehose** and ingested into an **Amazon Simple Storage Service** (Amazon S3) data store.
- 6 An **Amazon EventBridge** rule invokes a **Lambda** function at a schedule frequency specified by the user using a cron expression. If anomaly is detected in the stored data using **Amazon SageMaker Canvas**, it sends relevant information including device ID(s) to the **Lambda** function.
- 7 **Amazon Bedrock Agents** reviews the received input and takes appropriate action by triggering the right action group.
- 7a The agent queries the QnABot on AWS knowledge base to retrieve any relevant supplementary information or policies, including remediation procedures. This is appended to the ticket if required in step 7b.
- 7b The agent creates a ticket within the contact center and enriches the message with any data from step 7a.



Guidance for Connecting Automated Inputs to Contact Centers on AWS

AI-Powered Automated Incident Detection and Response System Using Bedrock Agents

This architecture diagram shows how to enable real-time error, anomaly detection, automated incident resolution, and seamless contact center integration for IoT-connected systems. This servers to reduce downtime and optimize issue resolution through AI-driven workflows.

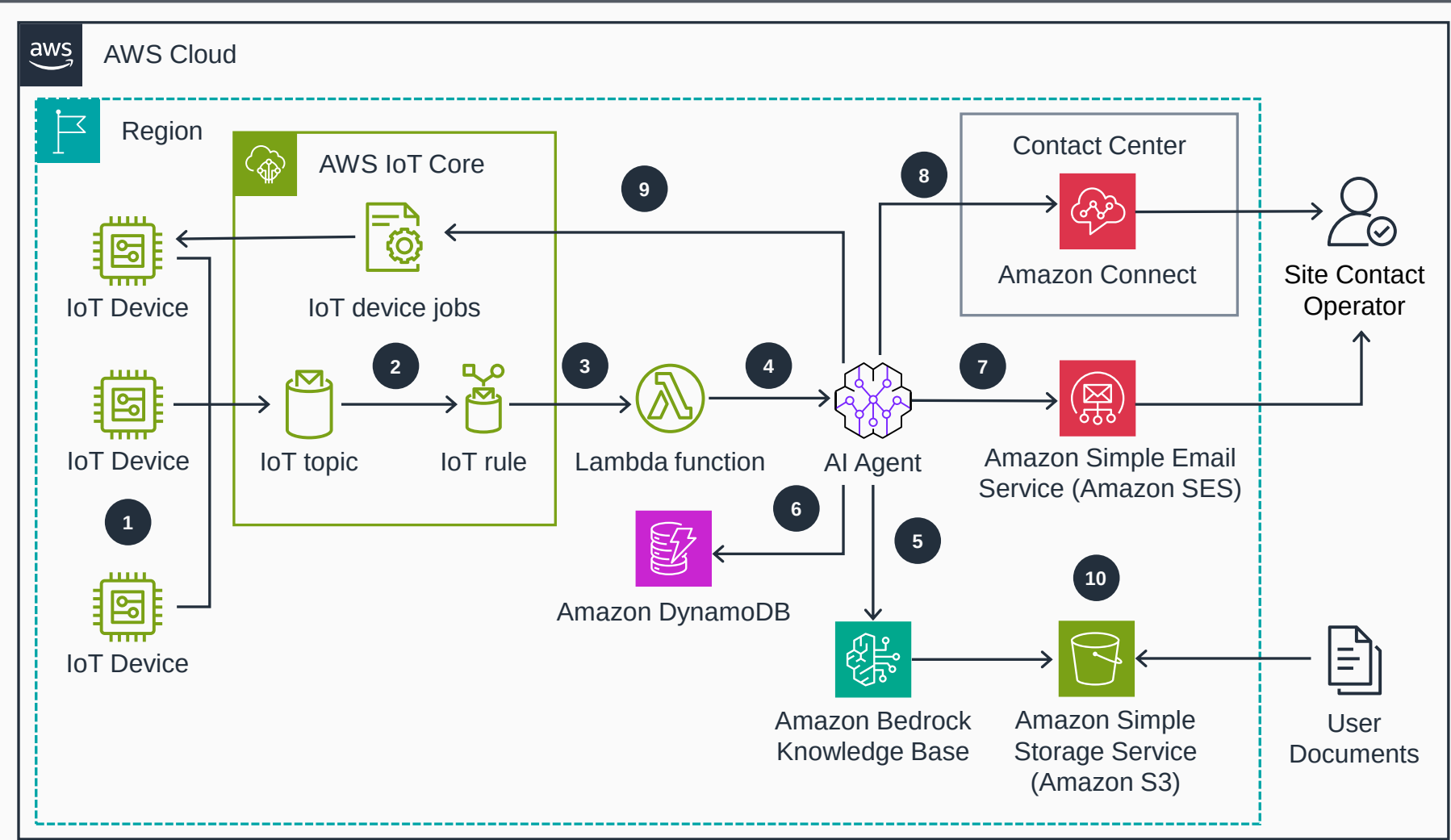


- 8 If the automated remediation policy is enabled, the system will automatically remediate the IoT device by executing an AWS IoT Job.
- 9 The site operator, owner, or customer is notified of the incident and any actions taken.
- 10 The site operator engages with the QnABot on AWS solution along with the ability to request additional information or access historical data as needed.
- 11 The company's manuals, standard operating procedures (SOPs), and engineering specifications are ingested into an **Amazon S3** bucket, which the knowledge base then references.

Guidance for Connecting Automated Inputs to Contact Centers on AWS

AI-Powered Automated Incident Detection and Response System Using Amazon Bedrock AgentCore

This architecture diagram illustrates how to effectively support connecting automated inputs to contact centers on AWS. It enables near real-time error monitoring and resolution, and seamless contact center integration for IoT-connected systems, reducing downtime and optimizing issue resolution through AI-driven workflows.

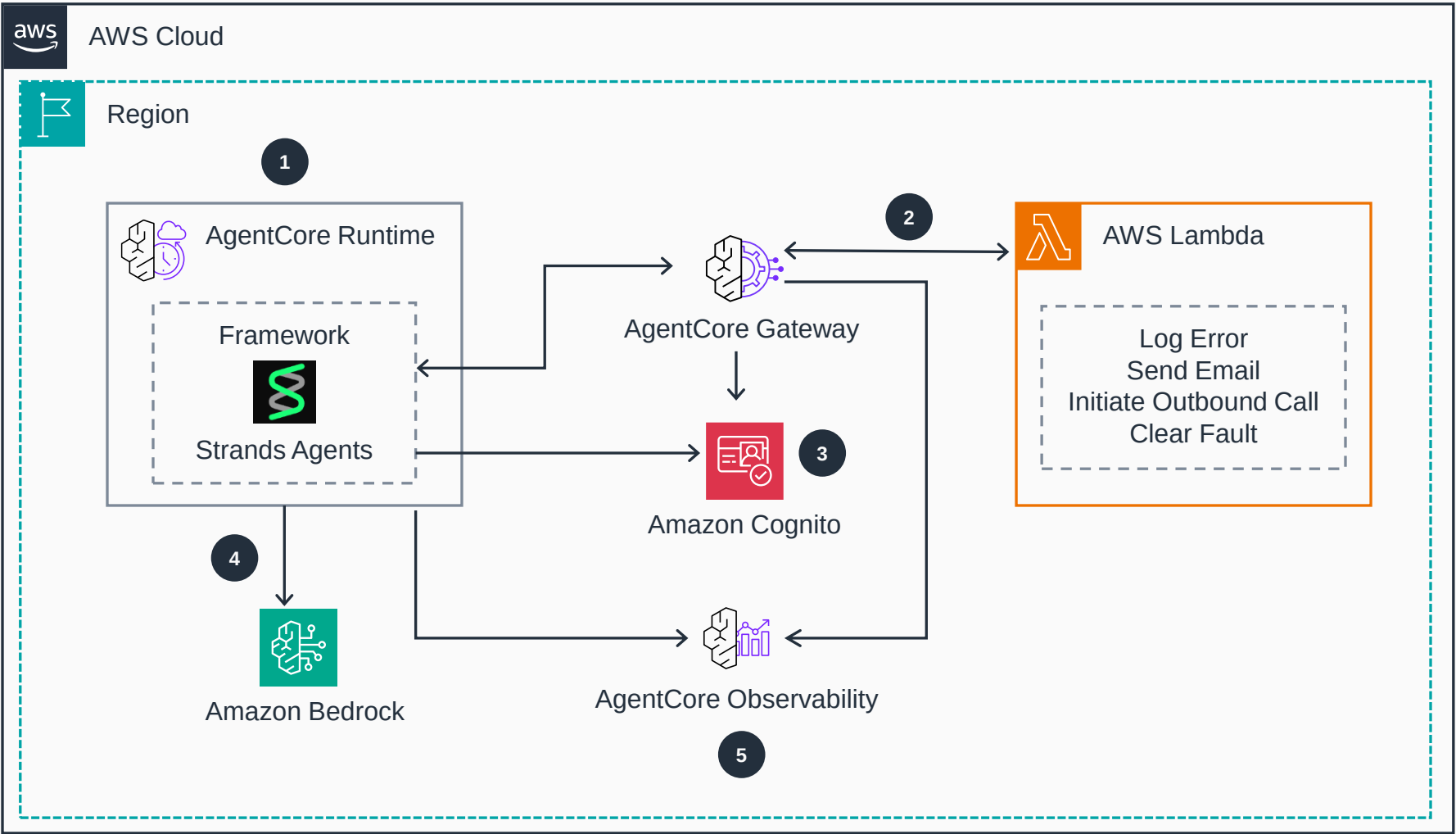


- 1 IoT devices publish telemetry and error messages to **AWS IoT Core**.
- 2 Errors or alerts generated by devices are published to designated IoT topics, such as an "error topic" or a "basic ingest topic."
- 3 These inputs are then routed to an **AWS Lambda** function based on the defined IoT rule.
- 4 The **Lambda** function transmits the error code, device ID, and any additional relevant information to AI Agent.
- 5 AI Agent queries **Amazon Bedrock Knowledge Bases** to retrieve any relevant supplementary information or policies, including remediation procedures. AI Agent reviews the received input and takes appropriate action by triggering the right tools. See detailed AI Agent architecture in slide 2.
- 6 AI Agent creates a ticket in **Amazon DynamoDB**.
- 7 AI Agent sends an email to the site contact operator using **Amazon Simple Email Service (Amazon SES)**.
- 8 If the remediation step includes calling the site contact operator, AI Agent initiates an outbound call using **Amazon Connect**.
- 9 If the automated remediation policy is enabled, the system will automatically remediate the IoT device by executing an **AWS IoT** job.
- 10 The user documents, such as company manuals, standard operating procedures (SOPs), and engineering specifications are ingested into an **Amazon Simple Storage Service (Amazon S3)** bucket. They are synced with **Amazon Bedrock Knowledge Base**.

Guidance for Connecting Automated Inputs to Contact Centers on AWS

AI-Powered Automated Incident Detection and Response System Using Amazon Bedrock AgentCore (AI Agent Flow)

This architecture diagram illustrates how to effectively support connecting automated inputs to contact centers on AWS. It enables real-time error monitoring and resolution, and seamless contact center integration for IoT-connected systems, reducing downtime and optimizing issue resolution through AI-driven workflows. This slide shows Steps 1-5 for the AI Agent.



- 1 AI Agent is built using **Strands Agents** pattern, which enables complex reasoning and decision-making workflows for automated customer service applications. The AI Agent is deployed on **Amazon Bedrock AgentCore Runtime**, which provides purpose-built infrastructure with the scalability, reliability, and security, critical to real-world deployment.
- 2 **AWS Lambda** functions, such as log errors, send emails, initiate outbound calls, and clear faults, are exposed as MCP Servers in **Amazon Bedrock AgentCore Gateway**. This allowed AWS Lambda functions to be accessible to AI Agent as MCP-compatible tools.
- 3 **Amazon Cognito** handles authentication and authorization for secure connection between Amazon Bedrock AgentCore Runtime and Amazon Bedrock AgentCore Gateway.
- 4 AI Agent processes inputs using large language models (LLMs) hosted on **Amazon Bedrock**. The agent enhances its responses by retrieving relevant information from **Amazon Bedrock Knowledge Bases** through retrieval augmented generation, allowing it to incorporate domain-specific knowledge into its decision-making process.
- 5 **Amazon Bedrock AgentCore Observability** tracks and monitors all AI Agent's activities.