

 [Workshops](#)

  duc.vu ▼

☰ Event ends in 2 days 15 hours 13 minutes. ✕ ⓘ

[Event dashboard](#) > Agents

Agents

Overview

In this lab, you will learn about Agents for Amazon Bedrock, a new Amazon Bedrock capability that lets you harness the Foundation Model's (FM's) reasoning skills to execute multi-steps business tasks using natural language. You can simply state your problem, like "help me update my product catalogue" and the agent breaks down the problem using the FM's reasoning capabilities and executes the steps to fulfill your request. You set up an agent with access to your organization's enterprise systems, processes, knowledge bases, and some building block functions. Then the agent comes up with the logic, figures out what APIs to call and when to call them, and completes the transactions in the right sequence. When an agent needs a piece of information from the user, it automatically asks the user for those additional details using natural language. And the best part about agents — it's leveraging the most up to date information you have and gives you relevant answers securely and privately.

An agent consists of the following components:

Foundation model – You choose a foundation model that the agent invokes to interpret user input and subsequent prompts in its orchestration process, and to generate responses and follow-up steps in its process.

Instructions – You write up instructions that describe what the agent is designed to do. With advanced prompts, you can further customize instructions for the agent at every step of orchestration and include Lambda functions to parse the output of each step.

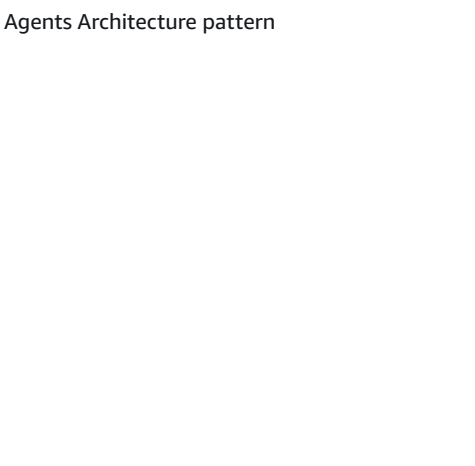
(Optional) Action groups – You define the actions that the agent should carry out through providing two resources.

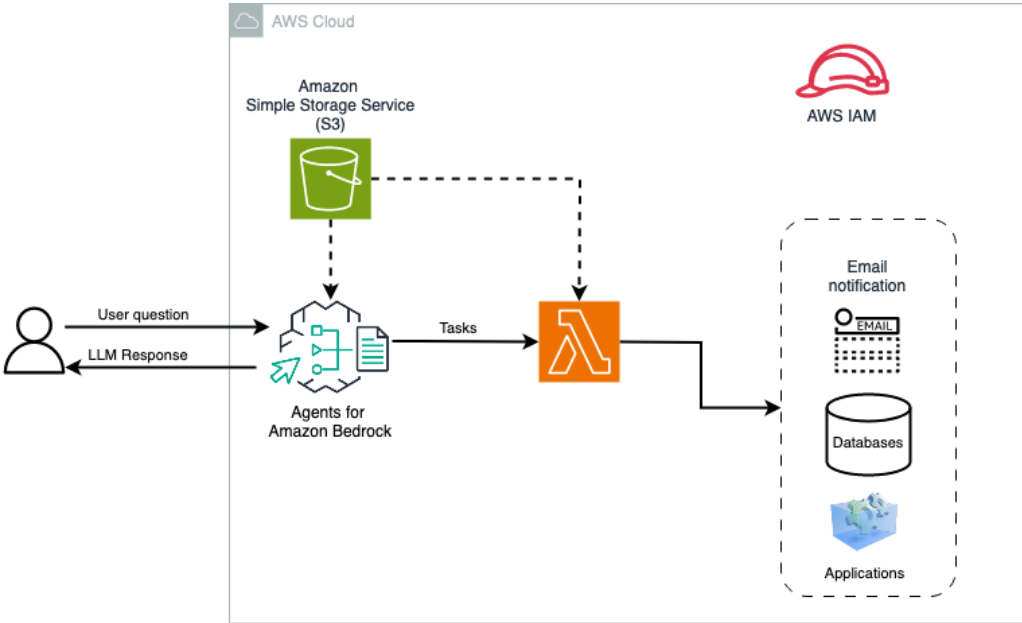
- An OpenAPI schema to define the APIs that the agent can invoke to carry out its tasks.
- A Lambda function with the following input and output.
 - Input – The API and parameters identified during orchestration.
 - Output – The result of the API invocation.

(Optional) Knowledge bases – Associate knowledge bases with an agent to allow it to query the knowledge base for extra context to augment response generation and input into steps of the orchestration process.

The following image schematizes the components of your agent.

Agents Architecture pattern

© 2008 - 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy policy](#)
[Terms of use](#)
[Cookie preferences](#)



Agents components



Instructions: "you are an agent that ..."



Actions

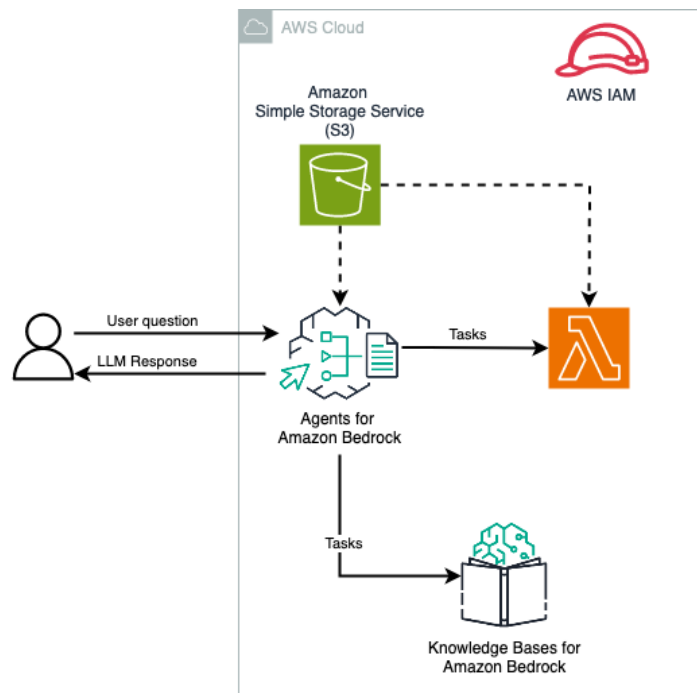


Knowledge Bases



Foundation Model

Agent's Patterns with KnowledgeBase



In build-time, all these components are gathered to construct base prompts for the agent in order to carry out orchestration until the user request is completed. With advanced prompts, you can modify these base prompts with additional logic and few-shot examples to improve accuracy for each step of agent invocation. The base prompt templates contain instructions, action descriptions, knowledge base descriptions, and conversation history, all of which you can customize to modify the agent to the best of your needs. You then prepare your agent, which packages all the components of the agents, including security configurations, and brings the agent into a state where it is ready for testing in runtime.

Relevance

Generative AI applications often need to execute multistep tasks across company systems and data sources. Agents for Bedrock allows you to automatically orchestrate and analyze the task and break it down into the correct logical sequence using the FM's reasoning abilities. Agents automatically call the necessary APIs to transact with the company systems and processes to fulfill the request, determining along the way if they can proceed or if they need to gather more information. Using encryption in transit and at rest and IAM roles, agents provide secure access to enterprise data and APIs. With the Action Groups integration with AWS Lambda functions, agents lets you choose the implementation language for your API's connection. Thanks to the fully managed infrastructure provided, you don't have to worry about provisioning or managing infrastructure for your agent.

Target Audience

- Software Developers, Data Scientists, Solutions Architects and anyone else building Generative AI applications with access to internal APIs or Knowledge Bases

Challenges

Because agents architecture may include external components, debugging errors could be complex. This workshop will guide you through the debugging process of the most common challenges.

Sub-patterns

During this workshop, you will cover three modules:

1. [Building Agents for Insurance using API](#): Building Agents for Bedrock using Boto3 SDK: In this module, you will create agents for Bedrock programmatically using the insurance claim agent example. The files for this module are available in the `insurance_claims_agent/without_kb` folder.
2. [Integrating a Bedrock Knowledge Base to a Bedrock Agent](#): Integrating Knowledge Bases to your Agents: In this module, you will create and integrate a Knowledge Base to your insurance claims agent both via AWS console and via Boto3 SDK. The files for this module are available in the `insurance_claims_agent/with_kb` folder.



3. [Creating an agent using OpenSource via LangChain](#): Creating an Agent using LangChain: In this module, you will learn how to retrieve data automatically from APIs using an OpenSource solution based on LangChain.

[Previous](#)[Next](#)