How it works

Agents for Amazon Bedrock consists of the following two main sets of API operations to help you set up and run an agent:

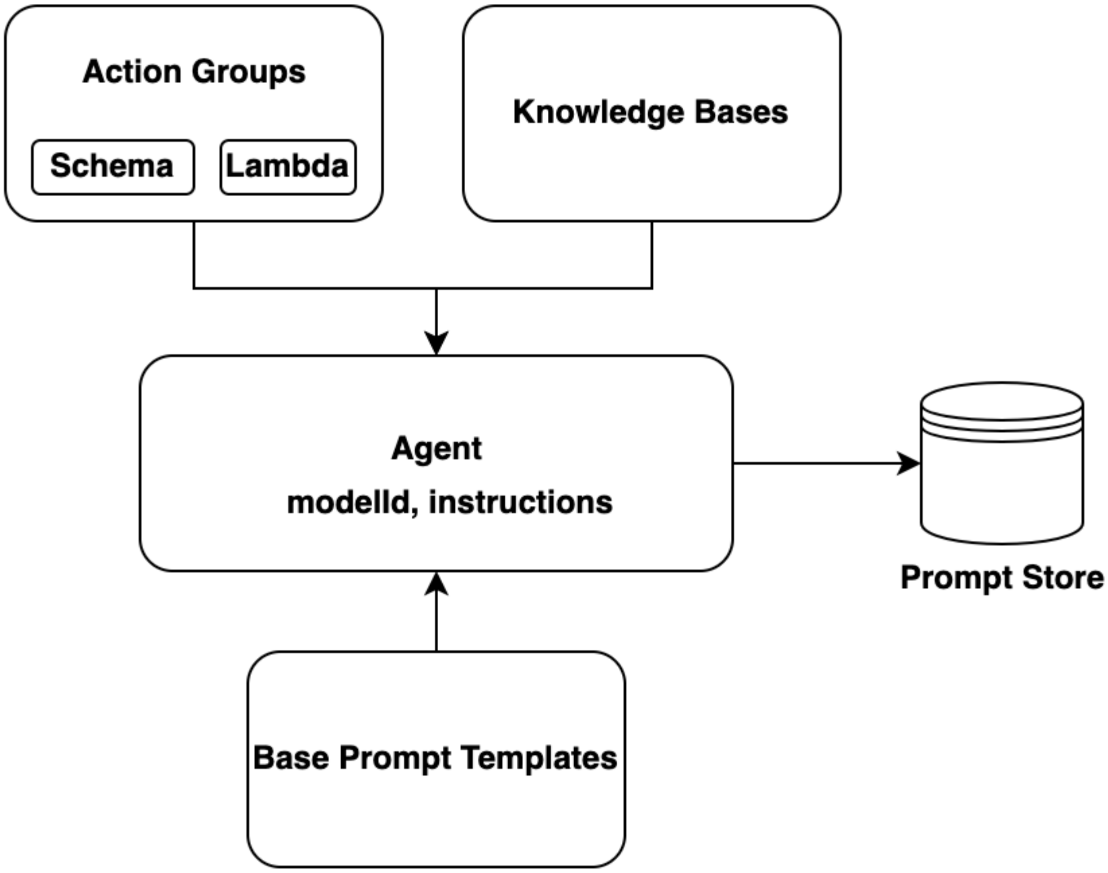
- **Build-time API** operations to create, configure, and manage your agents and their related resources
- **Runtime API** operations to invoke your agent with user input and to initiate orchestration to carry out a task.

Build-time configuration

An agent consists of the following components:

- **Foundation model** – You choose a foundation model (FM) that the agent invokes to interpret user input and subsequent prompts in its orchestration process. The agent also invokes the FM to generate responses and follow-up steps in its process.
- **Instructions** – You write 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 each step's output.
- **Action groups** (Optional) – You define the actions that the agent should perform through providing the following resources:
    - An OpenAPI schema to define the API operations that the agent can invoke to perform its tasks.
    - A Lambda function with the following input and output:
    - **Input** – The API operation and parameters identified during orchestration.
    - **Output** – The result of the API invocation.
- **Knowledge bases** (Optional) – Associate knowledge bases with an agent. The agent queries the knowledge base for extra context to augment response generation and input into steps of the orchestration process.
- **Prompt templates** – Prompt templates are the basis for creating prompts to be provided to the FM. Agents for Amazon Bedrock exposes the default four base prompt templates that are used during the pre-processing, orchestration, knowledge base response generation, and post-processing. You can optionally edit these base prompt templates to customize your agent's behavior at each step of its sequence. You can also turn off steps for troubleshooting purposes or if you decide that a step is unnecessary. For more information, see Advanced prompts in Amazon Bedrock.

At build-time, all these components are gathered to construct base prompts for the agent to perform 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 meet your needs. You then prepare your agent, which packages all the components of the agents, including security configurations. Preparing the agent brings it into a state where it can be tested in runtime. The following image shows how build-time API operations construct your agent.



Runtime process

Runtime is managed by the InvokeAgent API operation. This operation starts the agent sequence, which consists of the following three main steps.

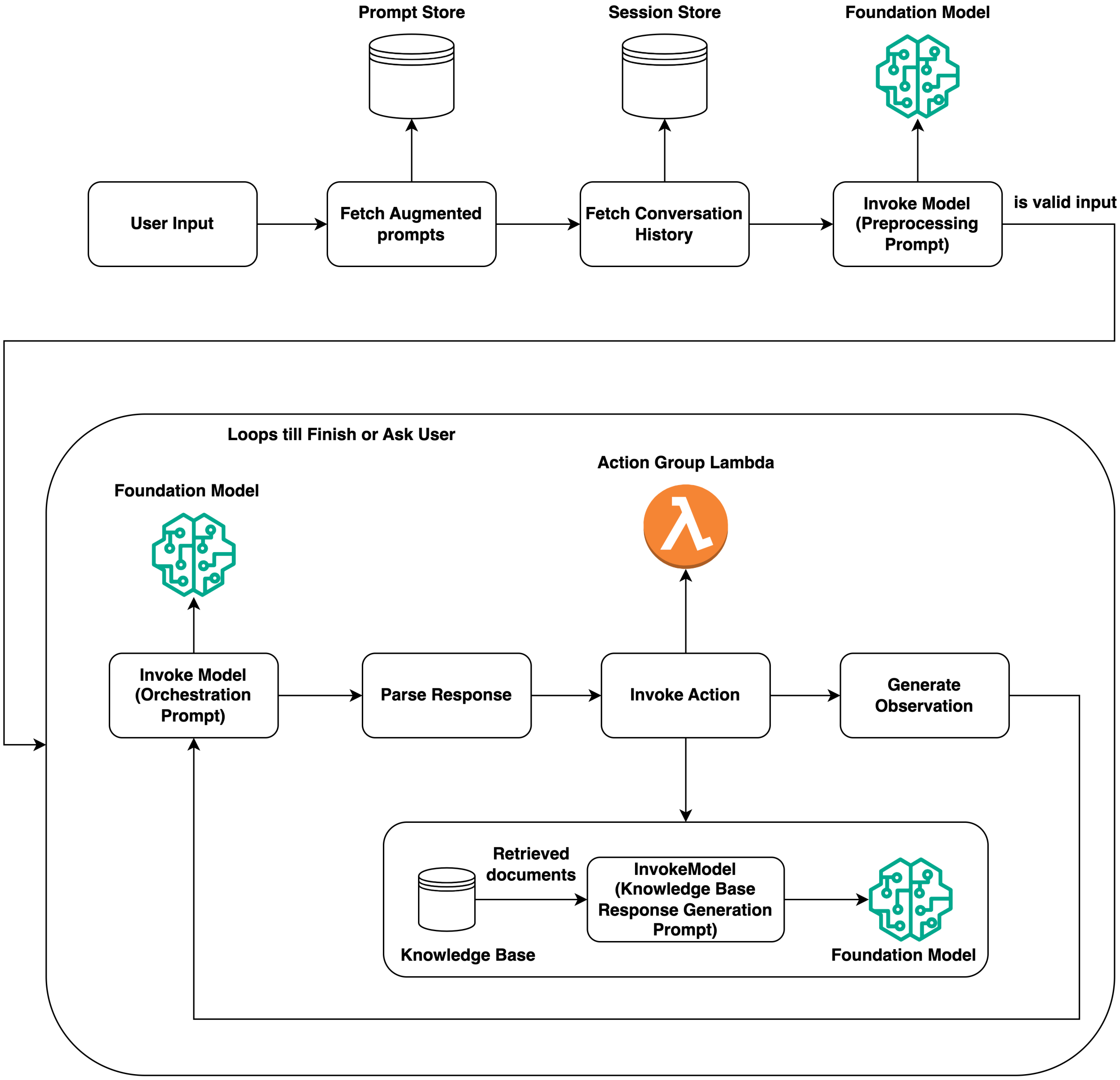
- **Pre-processing** – Manages how the agent contextualizes and categorizes user input and can be used to validate input.
- **Orchestration** – Interprets the user input, invokes action groups and queries knowledge bases, and returns output to the user or as input to continued orchestration. Orchestration consists of the following steps:
  - The agent interprets the input with a foundation model and generates a rationale that lays out the logic for the next step it should take.
  - The agent invokes action groups and queries knowledge bases (**Knowledge base response generation**) to retrieve additional context and summarize the data to augment its generation.
  - The agent generates an output, known as an observation, from invoking action groups and summarizing results from knowledge bases. The agent uses the observation to augment the base prompt, which is then interpreted with a foundation model. The agent then determines if it needs to reiterate the orchestration process.
  - This loop continues until the agent returns a response to the user or until it needs to prompt the user for extra information.

  During orchestration, the base prompt template is augmented with the agent instructions, action groups, and knowledge bases that you added to the agent. Then, the augmented base prompt is used to invoke the FM. The FM predicts the best possible steps and trajectory to fulfill the user input. At each iteration of orchestration, the FM predicts the API operation to invoke or the knowledge base to query.

- **Post-processing** – The agent formats the final response to return to the user. This step is turned off by default.

When you invoke your agent, you can turn on a trace at runtime. With the trace, you can track the agent's rationale, actions, queries, and observations at each step of the agent sequence. The trace includes the full prompt sent to the foundation model at each step and the outputs from the foundation model, API responses, and knowledge base queries. You can use the trace to understand the agent's reasoning at each step. For more information, see Trace events in Amazon Bedrock

As the user session with the agent continues through more InvokeAgent requests, the conversation history is preserved. The conversation history continually augments the orchestration base prompt template with context, helping improve the agent's accuracy and performance. The following diagram shows the agent's process during runtime:



Previous

Next