Diving Deep into Bedrock AgentCore

Diving Deep into Bedrock AgentCore
Prerequisites
Amazon Bedrock AgentCore Fundamentals
AgentCore Runtime
Hosting Runtime Agent
Hosting MCP server
Advanced Concepts for Agentcore Runtime
AgentCore Gateway
AgentCore Identity
AgentCore Memory: Short-Term Memory
AgentCore Memory: Long-Term Memory Strategies
AgentCore Tools
AgentCore 1P Tool - AgentCore Code Interpreter
AgentCore 1P Tool - AgentCore Browser
AgentCore Browser - Nova Act SDK
AgentCore Browser - Browser-Use
AgentCore Observability
AgentCore Observability for Runtime hosted Agent
Observability for Non-Runtime hosted Agents

AgentCore
AgentCore Documentation

AWS account access

Workshop catalog in AWS Builder Center

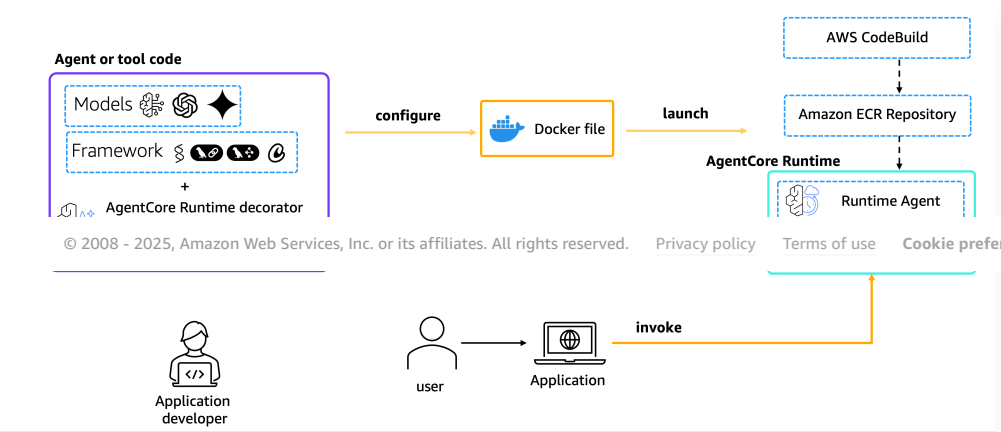
Content preferences

Exit event

Event ends in 18 hours 1 minute.

Amazon Bedrock AgentCore Runtime provides a secure, serverless and purpose-built hosting environment for deploying and running AI agents or tools, shortening the time to value from experiments to production-grade agents.

How it works



1. **Container ingestion** - The developer pushes an ARM64 container image to Amazon ECR. Runtime stores the image digest alongside metadata describing required ports (8080 for HTTP, 8000 for MCP).
2. **Runtime creation** - `CreateAgentRuntime` registers the image, assigns a workload identity, and stamps Version 1. A `DEFAULT` endpoint is generated that targets V1.
3. **Session bootstrap** - On the first `InvokeAgentRuntime` with a new `runtimeSessionId`, the control plane provisions an isolated execution environment. The bootstrap time is minimized through cached image digests and lazy layer extraction.
4. **Invocation handling** - The request payload (up to 100 MB) is streamed into the container. The agent produces either a JSON document or a Server-Sent Events stream. Runtime relays those bytes directly to the caller while simultaneously capturing observability data.
5. **Health negotiation** - AgentCore Runtime polls the `/ping` route. The agent replies with `Healthy` when idle or `HealthyBusy` while processing background tasks—allowing long-running workflows to persist across await points.
6. **Credential exchange** - If the agent calls an external tool requiring OAuth, it exchanges its workload token for a scoped access token via AgentCore Identity. Tokens are short-lived and bound to the invoking user where applicable.
7. **Termination** - After 15 minutes of inactivity or eight hours total runtime, the environment is terminated. Memory pages are wiped, temporary storage is discarded, and the session ID no longer maps to that environment.
8. **Version update** - Publishing a new image triggers Version `n+1`. Updating an endpoint redirects new sessions to the new version while existing sessions on the previous version continue until they naturally terminate.

Learn more about how AgentCore Runtime works

Labs:

In the following labs, you'll learn how to:

- **Lab: Hosting Runtime Agent** - Deploy and invoke your agent in AgentCore Runtime without managing infrastructure
- **Lab: Hosting MCP Server** - implement an MCP Server and deploy it in AgentCore Runtime
- **Lab: Advanced Concepts** - Implement streaming responses, manage agent session and context, and process large multi-modal payloads

Previous Next