Outbound Auth

Introduction

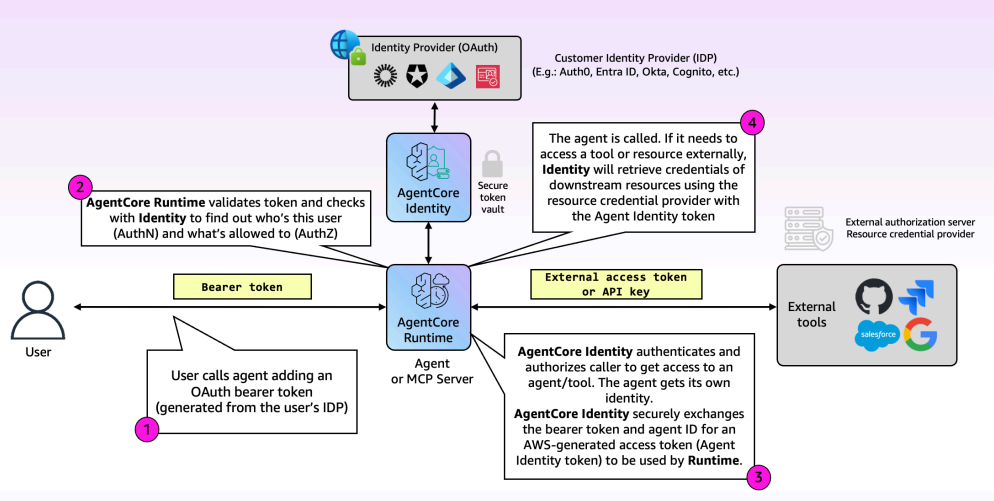
Outbound Auth allows agents and the AgentCore Gateway to securely access AWS resources and third-party services on behalf of users who have been authenticated and authorized during Inbound Auth. To integrate authorization with an AWS resource or third-party service, it's necessary to configure both

With just-enough access and secure permission delegation supported by AgentCore Identity, agents can seamlessly and securely access AWS resources and third-party tools such as GitHub, Google, Salesforce, and Slack. Agents can perform actions on these services either on behalf of users or independently, provided there is pre-authorized user consent. Additionally, you can reduce consent fatigue using a secure token vault and create streamlined AI agent experiences.

Outbound Authentication Configuration

First, you register your client application with third-party providers and then create an Outbound Auth. You specify how you want to validate access to the AWS resource or third-party service or AgentCore Gateway targets. You can use OAuth 2LO/3LO or API keys. With OAuth, you can select from providers that AgentCore Identity provides. In which case you enter the configuration details for the providers from AgentCore Identity. Alternatively, you can supply details for a custom provider.

When a user wants access to an AWS resource or third-party service or AgentCore Gateway target, the Outbound Auth confirms that the access tokens provided by Incoming Auth are valid and if so, allows access to the resource.



Why is this Important?

Modern AI agents need to interact with multiple external services to provide comprehensive functionality. Without proper Outbound Auth:

- **Credential sprawl** leads to security vulnerabilities and management overhead
- **Hard-coded secrets** in agent code create security risks
- **Manual token management** is error-prone and doesn't scale

- **Lack of centralized control** makes credential rotation and auditing difficult
- **User consent fatigue** degrades user experience with repeated authorization prompts

Outbound Auth solves these challenges by providing centralized credential management with secure token vaults and streamlined user consent flows.

Try it out

At an AWS Event

In the JupyterLab UI, navigate to `03-AgentCore-identity/04-Outbound Auth example`

Example	Framework	Model	Description
Outbound Auth example API Keys	Strands Agents	Amazon Bedrock	Outbound auth example with API keys
Outbound Auth Google 3lo	Strands Agents	Amazon Bedrock	Outbound auth example with 3LO using Google
Outbound Auth GitHub 3lo	Strands Agents	Amazon Bedrock	Outbound auth example with 3LO using GitHub

Self-paced

Here are notebooks that let you try out the above and extend the patterns to other frameworks and models:

Example	Framework	Model	Description
Outbound Auth example API Keys	Strands Agents	Amazon Bedrock	Outbound auth example with API keys
Outbound Auth Google 3lo	Strands Agents	Amazon Bedrock	Outbound auth example with 3LO using Google
Outbound Auth GitHub 3lo	Strands Agents	Amazon Bedrock	Outbound auth example with 3LO using GitHub

Previous Next