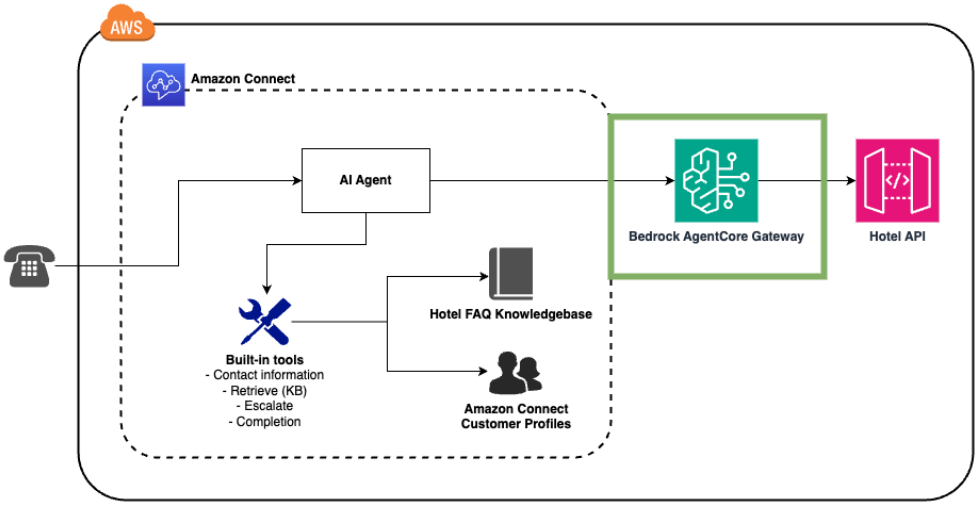
Task 2: Creating Your MCP Server

The first task you'll need to do is create an MCP server that acts as the interface between the AI Agent we'll create later and AnyCompany Hotel's API. The API and backend database have been deployed and you create a Bedrock AgentCore Gateway for it.



The API

Your MCP server will provide these core functions to the AI agent based on the AnyCompany's API:

Hotel Search Functions

- `searchHotels` - Search for hotels in a specific city

Customer Service Functions

- `getCustomerReservations` - Retrieve customer's existing reservations

Booking Management Functions

- `createBooking` - Create new hotel reservations
- `modifyBooking` - Update existing reservations
- `cancelBooking` - Cancel reservations

Step 1: Understand Pre-Deployed Hotel API

Your workshop environment includes a hotel reservation API that's ready to use. The CloudFormation templates have been deployed to your account with the backend infrastructure.

Pre-Deployed Resources:

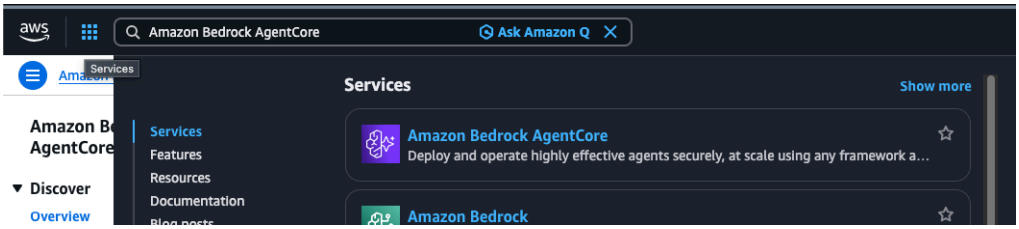
Your workshop account includes:

- Hotel API with Lambda functions (with mock data)
- IAM roles with appropriate permissions (ready to use)
- DynamoDB tables with mock customer and reservation data

- CloudWatch logging for monitoring (active and collecting data)

Step 2: Navigate to AgentCore Gateway

Open the AWS Console and search for "Amazon Bedrock AgentCore" and click.



Step 3: Add API Key to AgentCore

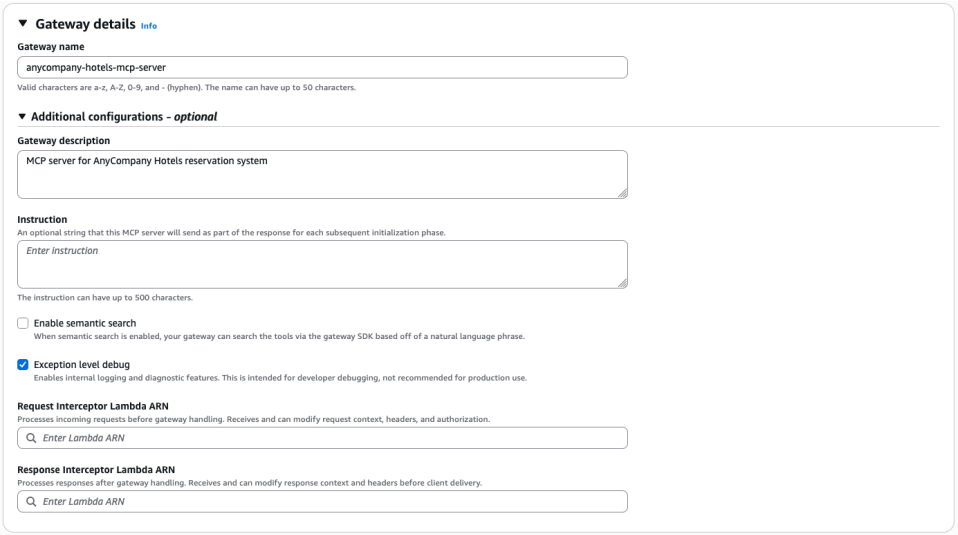
The API that has been deployed to your account has created an API key for providing AgentCore Gateway access to your API. We'll need to add that API Key to the identity section of AgentCore before creating our gateway.

1. **Click Identity** in the left pane navigation under Build.
2. **Click the Add OAuth client / API key** button and select `Add API key`
3. **Create API Key details:**
   - **Name:** `hotel-api-key`
   - **API key:** Paste the **HotelApiKey** value from your `workshop-values.txt` file.
4. **Click Add** to save.

Step 4: Create Your MCP Server

1. **Click "Gateways"** in the left hand navigation pane
2. **Click Create gateway** button
3. **Enter the Gateway details:**

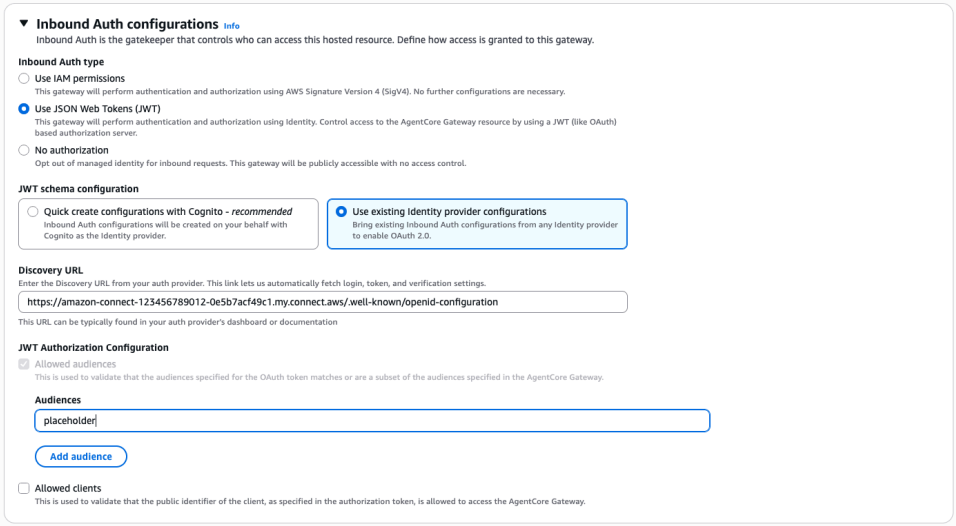
   - **Gateway Name:** `anycompany-hotels-mcp-server`
   - **Description (in Additional configurations):** `MCP server for AnyCompany Hotels reservation system`



4. **Inbound auth** - This controls how Amazon Connect authenticates when calling your gateway.

   - Below in the Inbound Auth configuration section select `Use JSON Web Tokens (JWT)`

- Select `Use existing identity provider configurations` and in the **Discovery URL** field, paste the **AgentCoreGatewayDiscoveryUrl** value from your `workshop-values.txt` file.
- **Check** `Allowed audiences` and put `placeholder` as text in the audiences box. *Don't worry—we'll come back to this shortly.*
- **Uncheck** the `Allowed clients` checkbox



5. **Permissions** Choose to create a new service role (should be selected by default)

Configure API Target:

Now you'll configure your MCP server to connect to the Hotel API. This step establishes the secure connection between AgentCore Gateway and your backend services.

1. **Basic Target Configuration:**
   - **Target Name:** `anycompany-hotels-api`
   - **Description:** `AnyCompany Hotels reservation management API`
   - **Target Type:** Select `REST API`
2. **Schema Configuration:**
   - **API Schema Type:** Select `OpenAPI schema`
   - **Schema Source:** Choose `Define with an S3 resource`
   - **S3 URI:** Paste the **HotelApiOpenApiSpecUrl** value from your `workshop-values.txt` file. (See the Hotel API OpenAPI Schema reference if you'd like to explore the API specification.)
3. **Authentication Configuration:**

Since your Hotel API requires an API key for authentication, you'll set authentication to the API key you created previously:

Outbound Auth configurations: The outbound auth defines how Bedrock AgentCore Gateway authenticates to our API.

- **Authentication Type:** Select `API Key`
- **API Key:** Select the API key you created earlier in the dropdown.

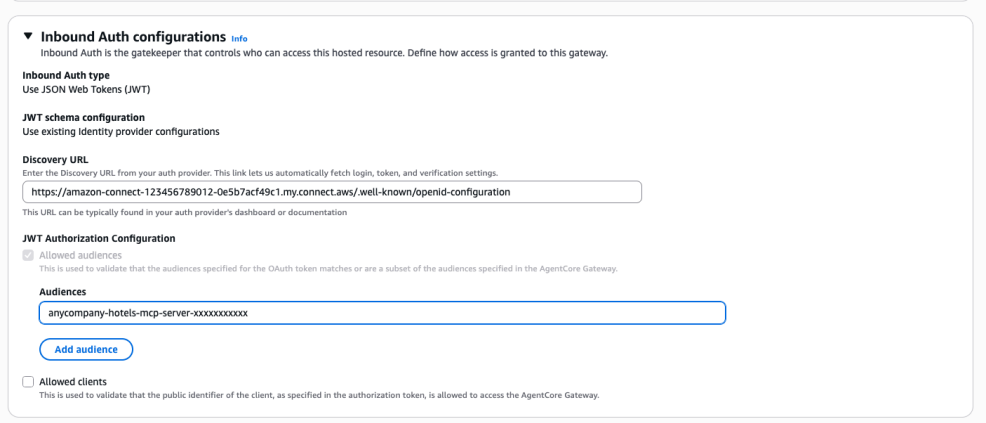
4. Click **Create gateway**!

Step 5: Update the Inbound Audience

Earlier we added `placeholder` as our inbound identity allowed audience. Now that the gateway is created we'll replace it with the actual Gateway ID.

1. **Copy the Gateway ID** from the Gateway details section.

2. **Click Edit** on the Inbound Identity below.
3. **Paste the Gateway ID** in the Audiences text box.
4. Ensure that **Allowed clients** is unchecked
5. **Save** your changes



What's Next?

Excellent! You've successfully created AgentCore Gateway MCP server. Your server is now ready to connect to Amazon Connect and provide intelligent hotel reservation capabilities to your AI agent.

In the next section, you'll associate your MCP server with Amazon Connect to enable these powerful hotel operation capabilities. The magic happens when natural language requests like "I need a suite in Seattle for next weekend" get transformed into precise function calls that your AgentCore Gateway MCP server can execute flawlessly.

Ready to make the connection? Let's associate your MCP server with Amazon Connect.

Previous Next