Guide for creating AI Lead Qualification Agent in Azure AI Studio

Josh Arrington
Chief Marketing Technology Officer, Kapturall
Adobe Marketo Engage Champion

Adobe

Table of Contents

Introduction

In today's competitive landscape, speed and accuracy in lead qualification are critical. Manually sifting through inbound leads is time-consuming and prone to inconsistency. This guide provides a comprehensive, step-by-step walkthrough for building an intelligent AI agent that automates this process, ensuring high-quality leads are identified and routed to sales faster than ever.

By leveraging the power of Microsoft Azure AI Studio and integrating it with Marketo Engage, you will create a sophisticated agent capable of analyzing lead data, enriching it with external information, evaluating it against your Ideal Customer Profile (ICP), and taking direct action within your marketing automation platform. This powerful combination allows you to build a scalable, consistent, and highly efficient lead qualification engine tailored to your business needs.

Prerequisites and Create a Resource Group

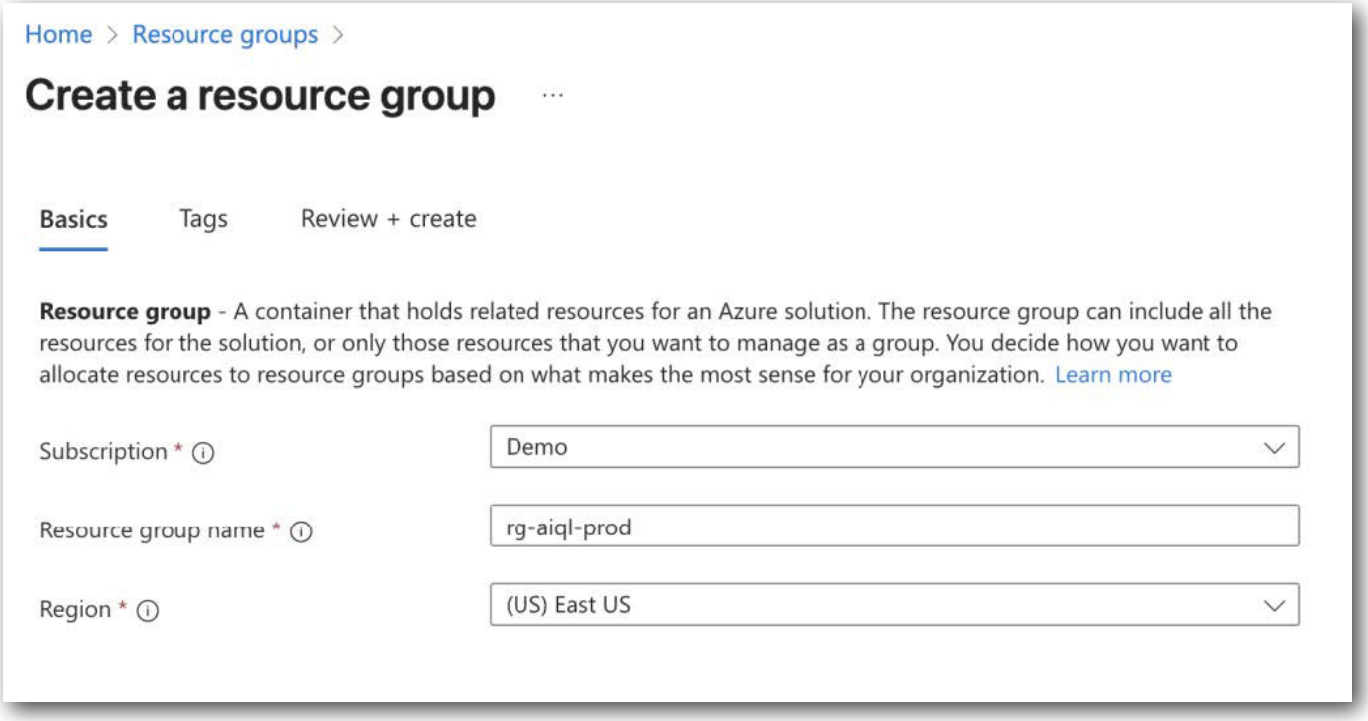
Prerequisites:

- Azure Account – If you don't have an Azure account, you can create on for free here

- Marketo API Credentials: Munchkin ID, REST URL, Client ID / Client Secret

Create a Resource Group:

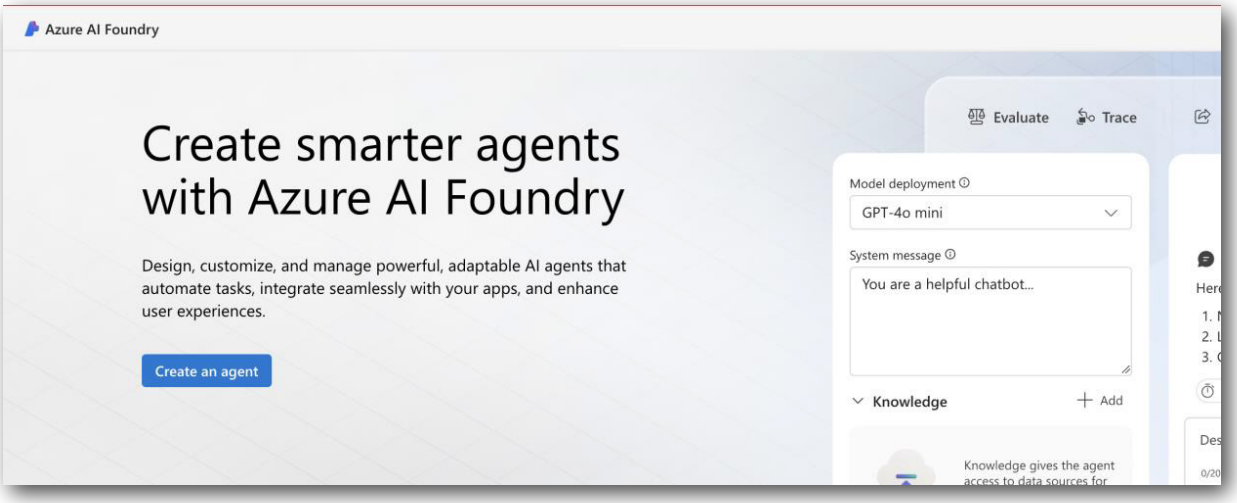
1. Azure Portal → Resource groups → Create.
2. Name: rg-aiql-prod (or your naming standard), choose subscription + region → Review + Create.



Create the Azure AI Studio workspace

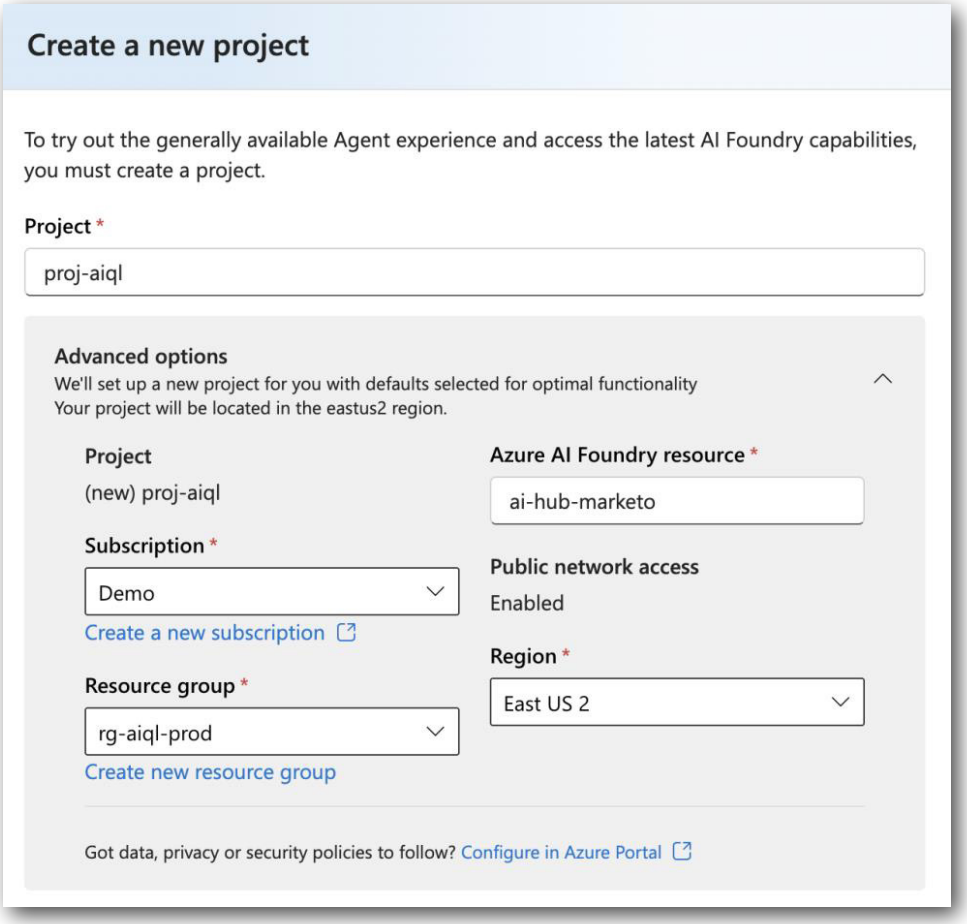
Azure AI Studio uses a **Hub** → **Project** structure and attaches dependent services.

1. Go to Azure AI Studio (formerly “Foundry”).



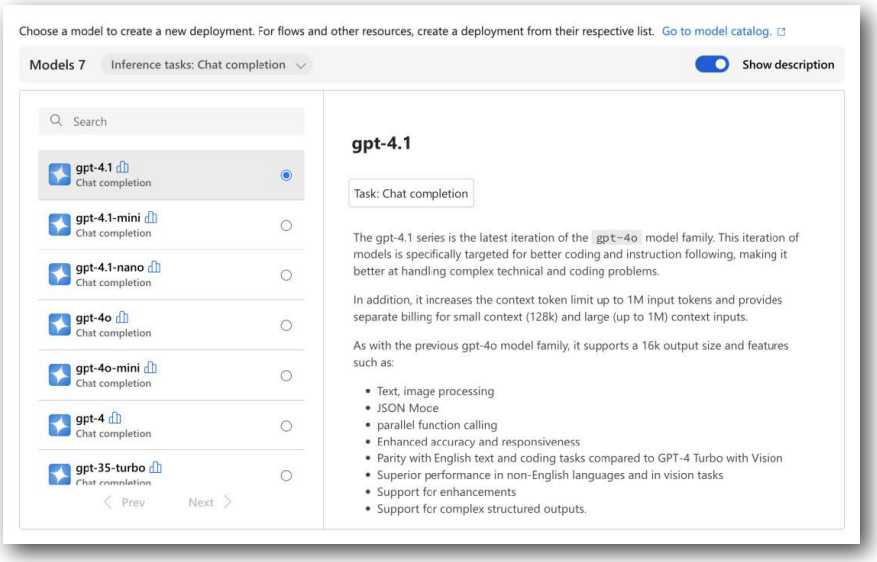
2.. Click the **Create an agent button.**

This will create an AI Foundry Hub, Project and Agent. Give both the project and AI Foundry Resource a name. Use the resource group you created before and select the same region.

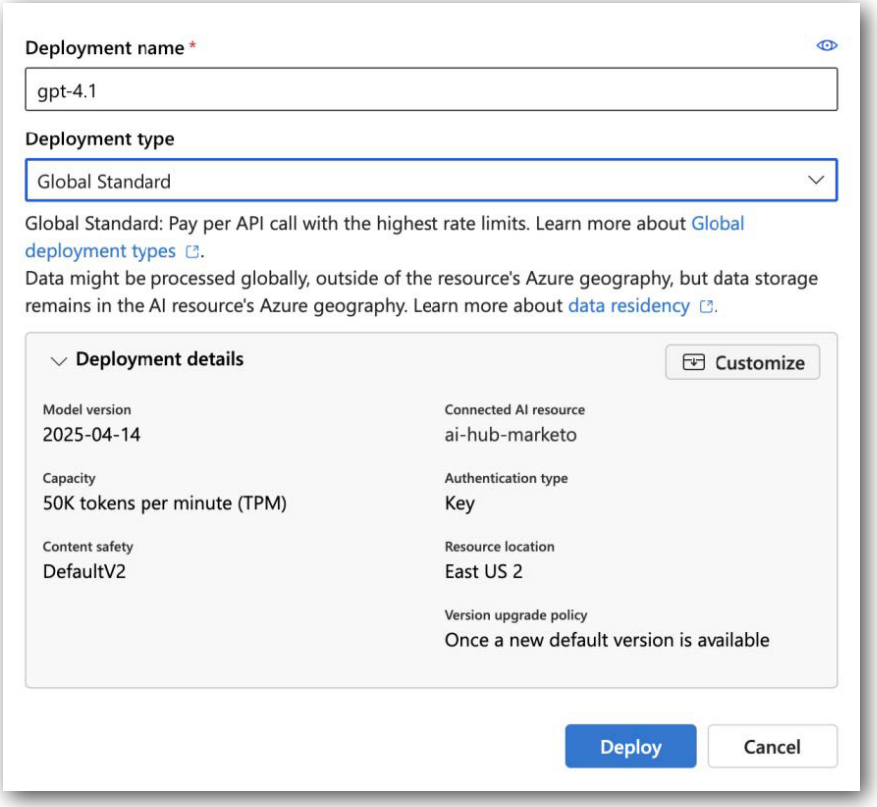


Create the Azure AI Studio workspace

3. Select a model



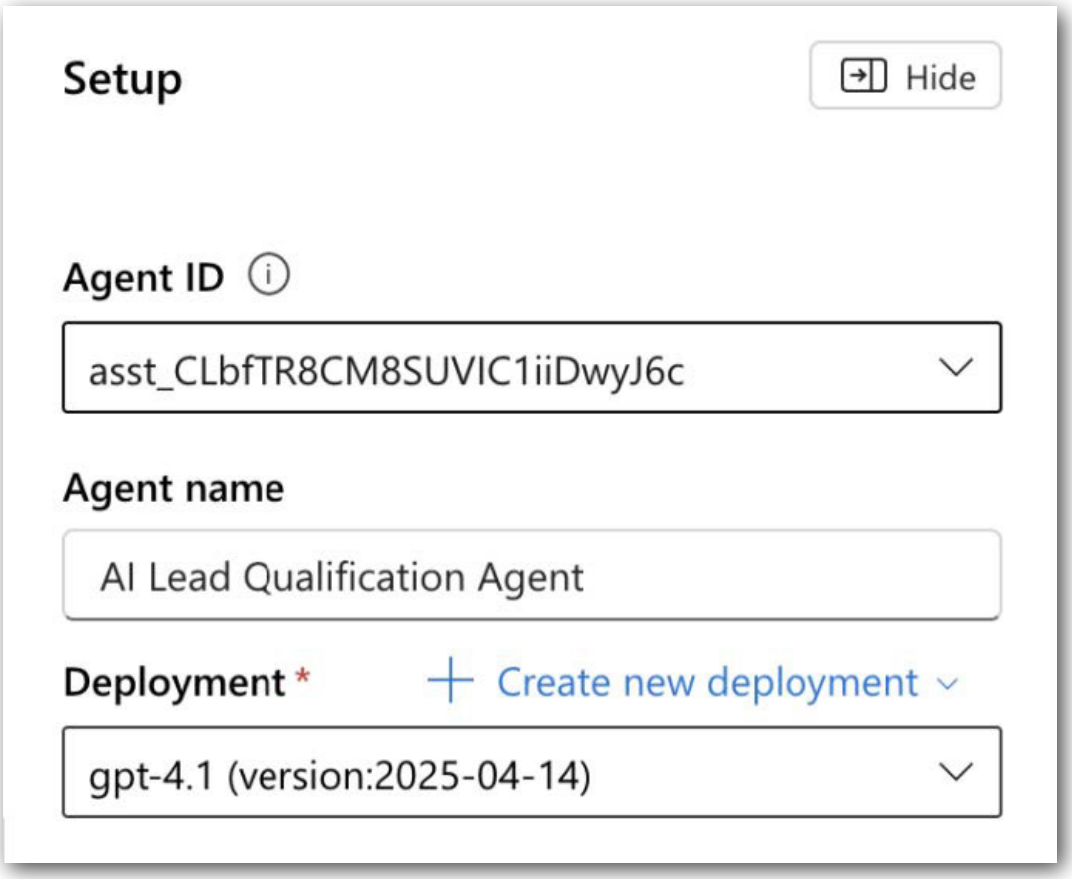
4. Deploy your agent



Tip: Keeping everything in the **same region & subscription** avoids cross-boundary headaches.

Create the Azure AI Studio workspace

5. Give the agent a name in the right panel



Take note of your Agent ID here. We will use it later to trigger the agent.

Give your Agent Instructions

Give your agent clear directions on what to do and how to do it. Include specific tasks, their order, and any special instructions like tone or engagement style.

Your job is to analyze inbound leads for our corporate real estate company and decide the most appropriate next step based on available data.

What To Review

- Lead data (behavioral, engagement, activity logs).
- Company data (firmographics such as size, industry, geography).
- Ideal Customer Profile (ICP) — check the vector store to evaluate how closely the lead and their company match our ICP.

After reviewing, you will call the Logic App by sending a JSON payload with the following fields:

- **leadID** – The Marketo ID of the lead.
- **aiAction** – The action you recommend for the lead. This must exactly match one of these values:
  - **qualify** - The lead is a good fit and ready for sales follow-up.
  - **disqualify** - The lead is not a fit and should be removed from active pursuit.
  - **nurture** - The lead is promising but not ready for direct sales contact; should enter a nurture program.
  - **hr** - The lead is an internal HR inquiry and should be routed to the HR team.
- **aiExplanation** – A concise summary of why you chose this action. Reference key factors like ICP match, company profile, lead behavior (e.g., form fills, email engagement), and intent signals.
- **aiCategory** – A short label for the lead's classification (e.g., "Strong ICP Fit," "Low Engagement," "Competitor," "HR Inquiry").
- **leadScore** – A numeric score (0–100) reflecting how likely the lead is to convert based on behavior and engagement.
- **icpScore** – A numeric score (0–100) reflecting how closely the lead matches our corporate real estate ICP (from the vector store).

Give your Agent Instructions

Decision process

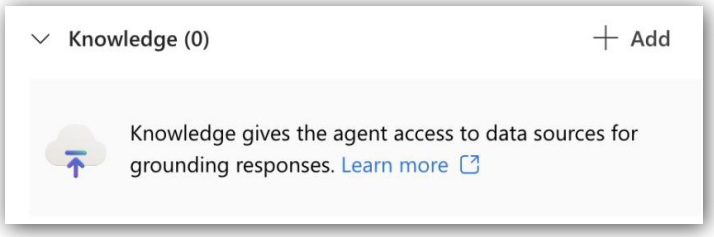
1. Always check the ICP fit in the vector store first to ensure alignment with the Ideal Customer Profile.
2. Weigh recent lead activity and engagement.
3. Use company firmographics to validate alignment with corporate real estate opportunities.
4. Choose the most appropriate aiAction.
5. Fill in all fields with clear, defensible reasoning.

If you are uncertain, you may call the **getApproval** tool before finalizing.

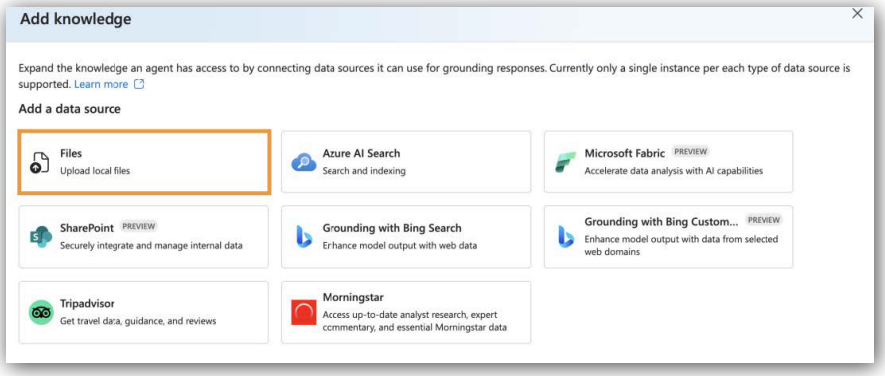
Once a decision has been made, call **requestMarketoCampaign** to trigger the appropriate action.

Load Knowledge into a Vector Store

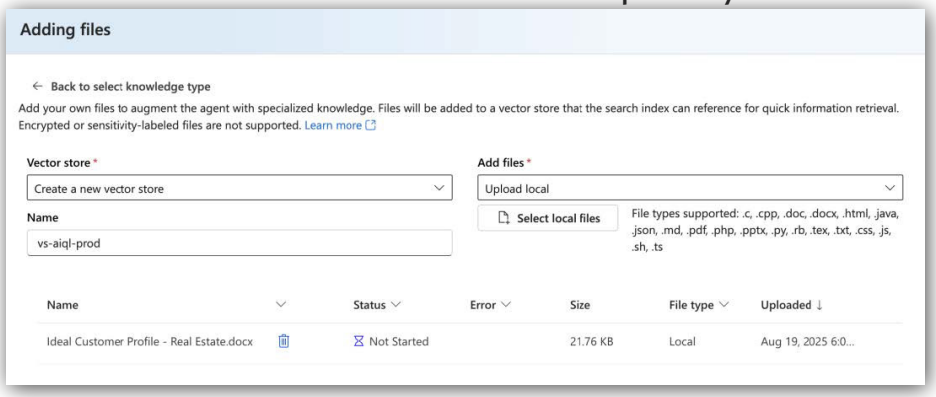
1. On the right panel find the **Knowledge** section and click **+ Add**



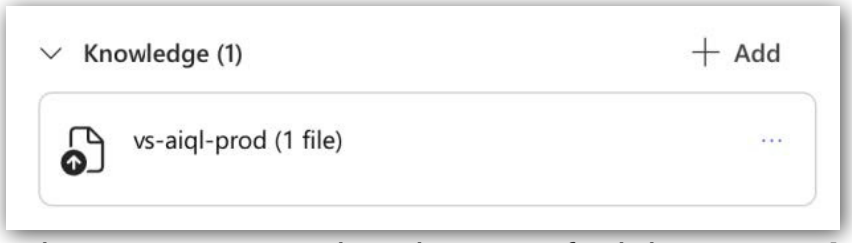
2. Select **Files**



3. Select or create a vector store and upload your ICP Document



4. Click the **Upload and save** button. You should now see the vector store under the knowledge section



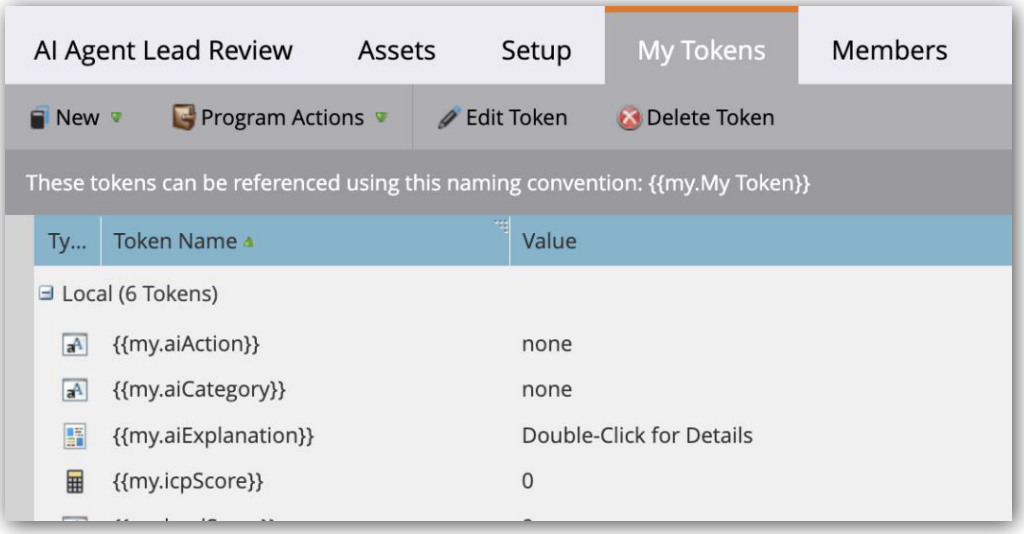
Why vector store? It lets the agent find the **most relevant passages semantically,** not just exact keywords.

Setting up in Marketo Engage

Once our agent reviews and evaluates the leads we've asked it to pass back one of four actions (qualify, disqualify, nurture, hr) and some additional details that we will pass to Marketo Engage when triggering our smart campaign. We need to create a Smart Campaign for each of these actions

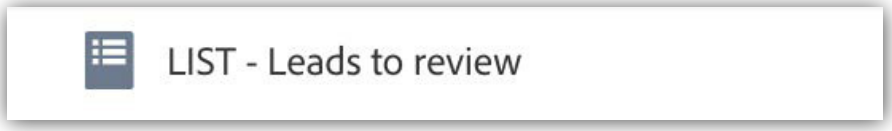
Create Programs and Tokens

Create Program Tokens for each value the agent is expected to send. The value we set at this moment isn't important, because when our agent calls the Smart Campaign it will pass in dynamic values for each lead.



Create Watch List

Create a static list. We will pull leads from this list for our agent to process



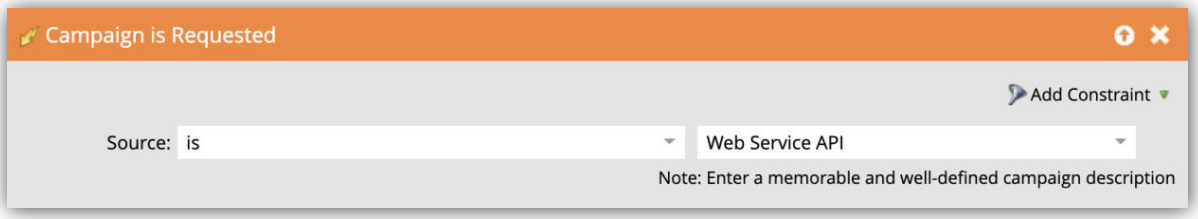
You can find the list ID in the URL. Navigate to your static list
https://experience.adobe.com/#/@accountName/so:123-ABC-456/marketo-engage/classic/ST**1020**A1LA1

Setting up in Marketo Engage

Create Smart Campaign

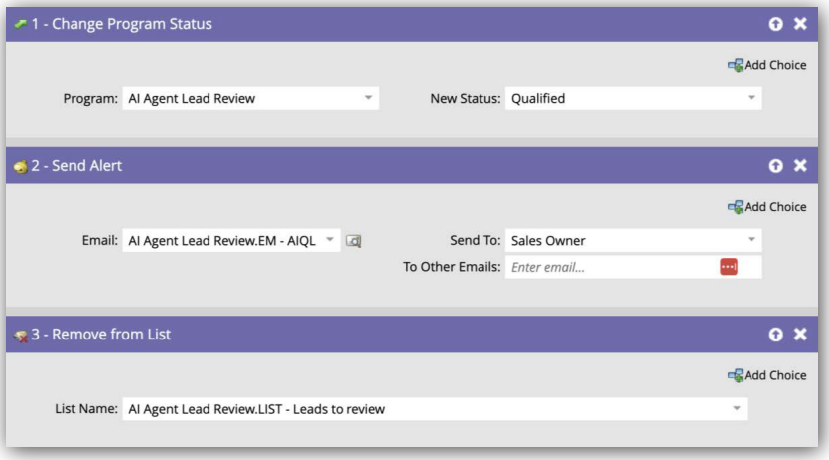
Campaign Smart List

The flow for our smart campaign will all use a single trigger which should be **Campaign is Requested** with source of **Web Service API** which will make our smart campaigns triggerable via the API.



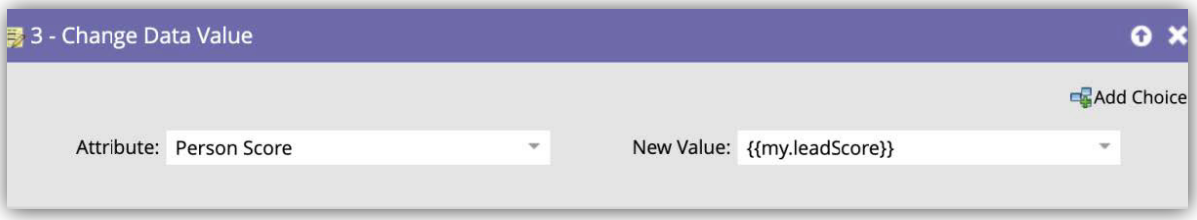
Campaign Flow

The flow for each campaign should match the action passed, but it can be whatever we want it to be. However, for our flow it is important to remove the user from the watchlist so it isn't processed again.



Setting up in Marketo Engage

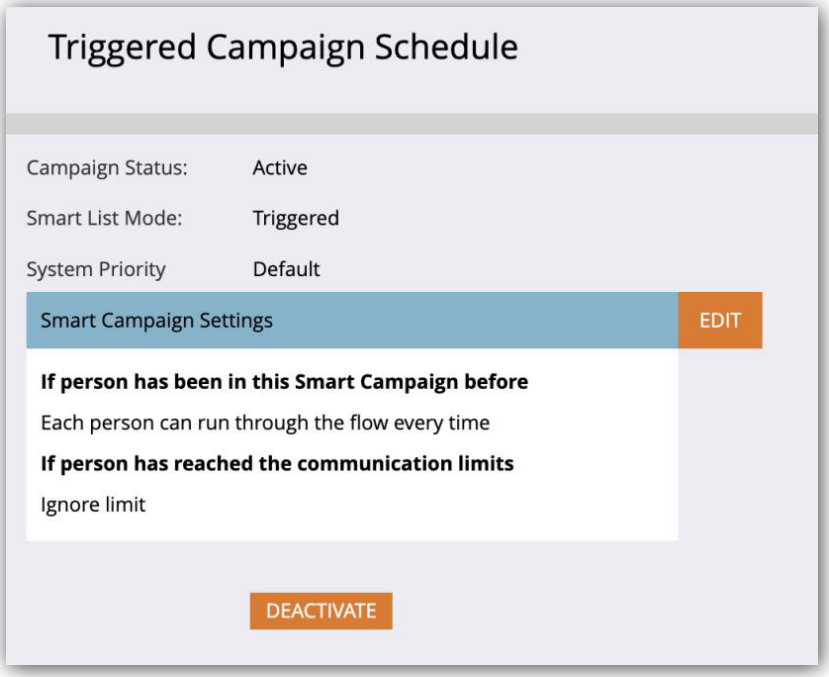
We can use all of the program tokens to **change data value**



We can also use them in the body of our emails. For a **sales alert.**

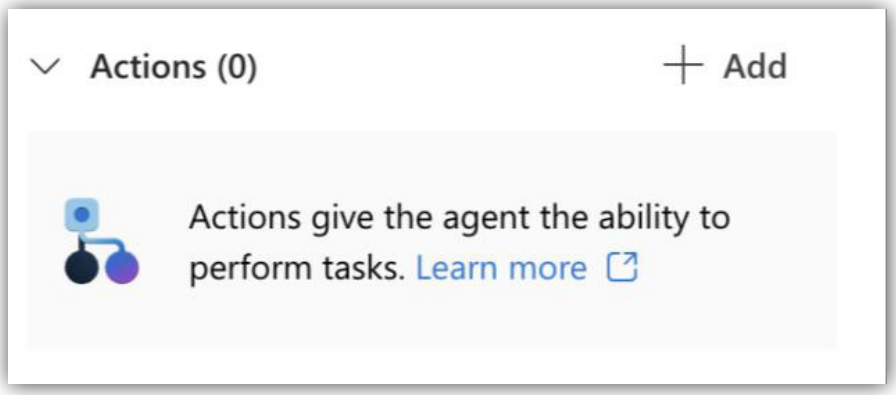
Campaign Schedule

Activate the campaigns so they are now available



Giving our AI Agent tools

In the right panel you will see an area for “Actions” where we can add tools for our agent to use, but first we need to create this functionality. In our case we will use Logic Apps to build our agents tools. These are no-code options for building data flows.



We will create three key actions:

- enrichLeadData
- getLeadActivity
- requestMarketoCampaign

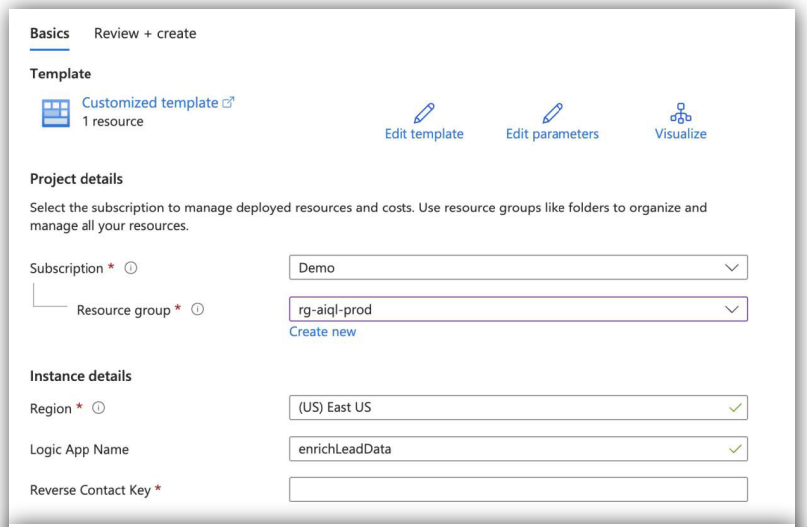
Giving our AI Agent tools

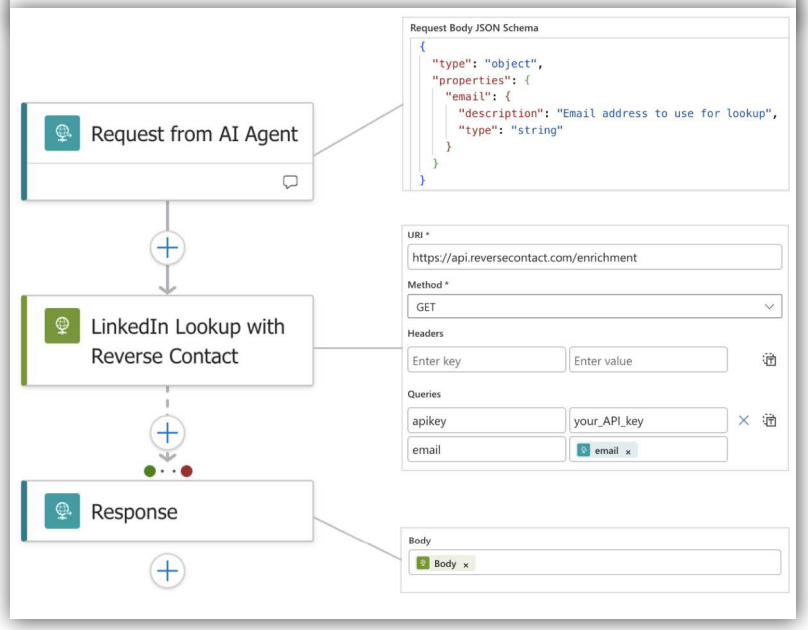
enrichLeadData

Deploy to Azure

For lead enrichment, I'm using Reverse Contact via a simple HTTP API call to get a more complete picture of the lead and their company. You can use any third-party or internal resource for this purpose, but be sure to follow applicable regulations like GDPR and your company's policies. Some other great options are ZoomInfo, Lusha, and FullContact

Place the logic app in the same resource group and region that we created before. Here, you'll need to enter your reverse contact key. If you want to use a different service, you can still use this template. Just enter any value here and then remove this action from the Logic App flow. Later in this guide there is a guide to adding additional Logic Apps actions to our agent.





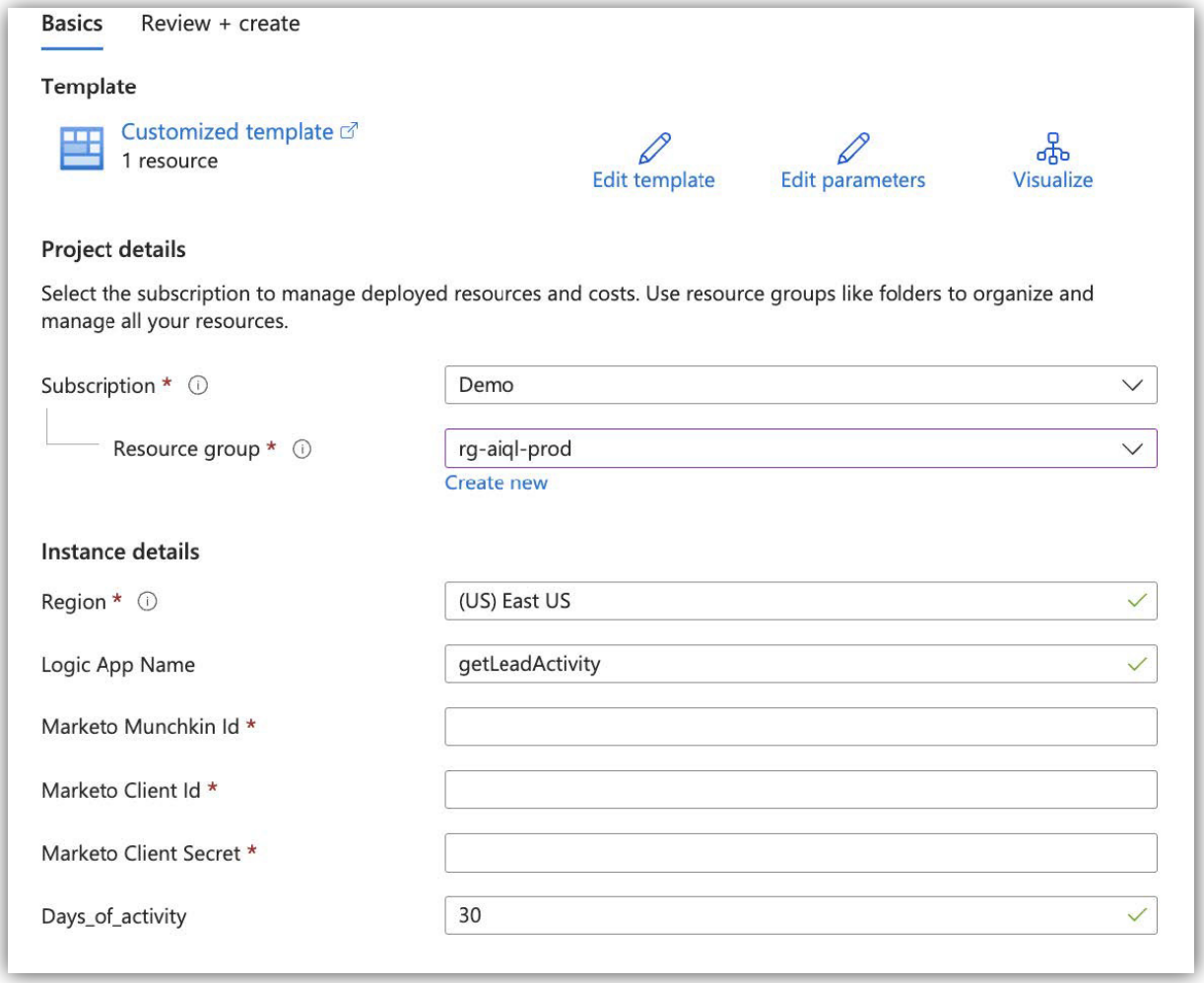
Giving our AI Agent tools

getLeadActivity

This tool allows the agent to pull the lead's activity history to analyze as part of the holistic review. Unfortunately, there is currently no built in Marketo Engage connector for activities in Azure Logic Apps, and this one gets a little complicated. To help with this I've created a template you can deploy from this link:

Deploy to Azure

Place the logic app in the same resource group and region that we created before. Here you'll need to enter your Marketo REST API credentials and select the number of number of days of activity you want the agent to pull. Click **Review + create** and deploy.

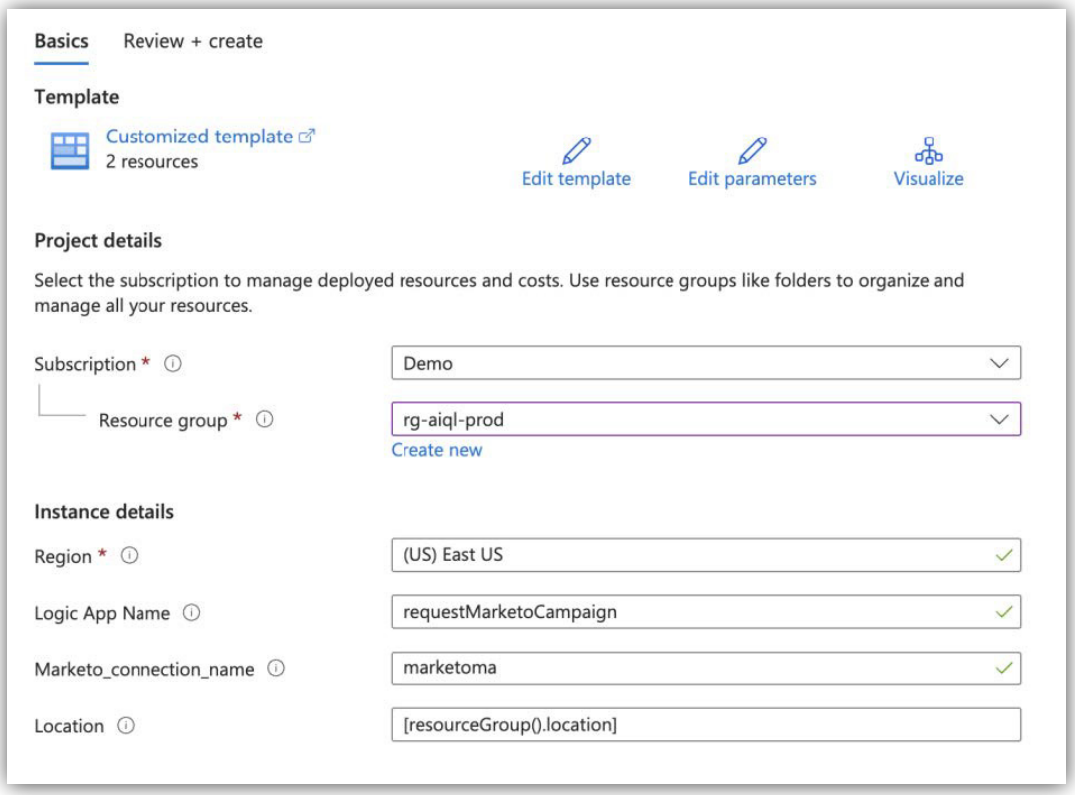


Once created you will see the logic app in your Azure resource group.

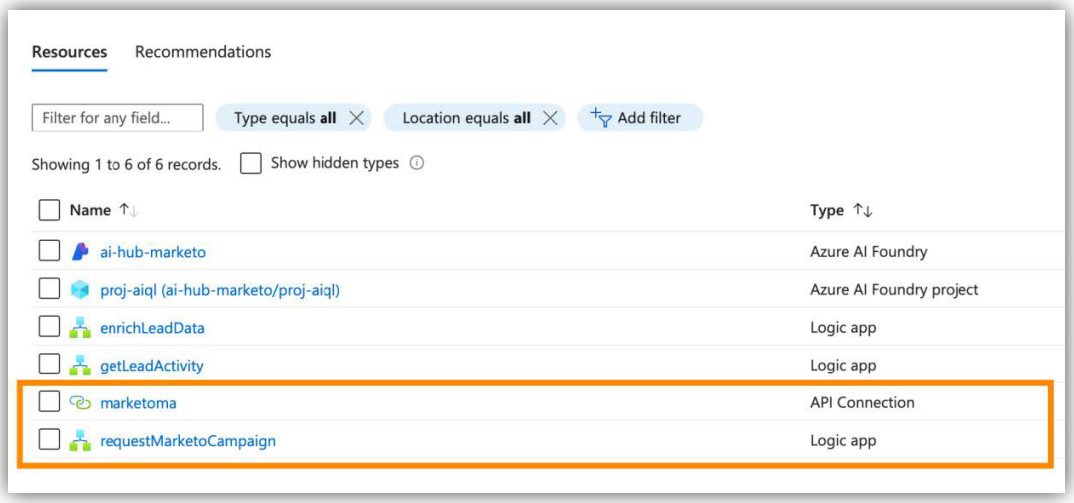
Giving our AI Agent tools

requestMarketoCampaign

Deploy to Azure

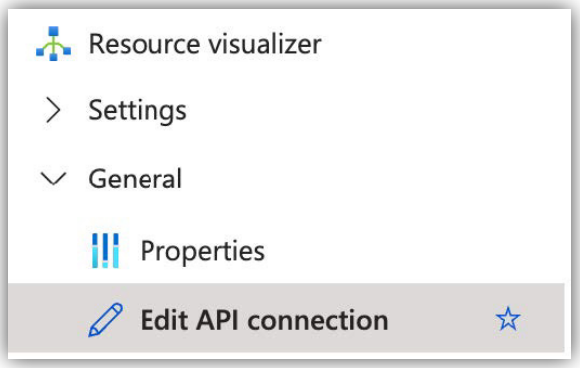


This template will create both the logic app and an API Connection: **marketoma**. First we need to set up the Marketo API connection. Click on it.

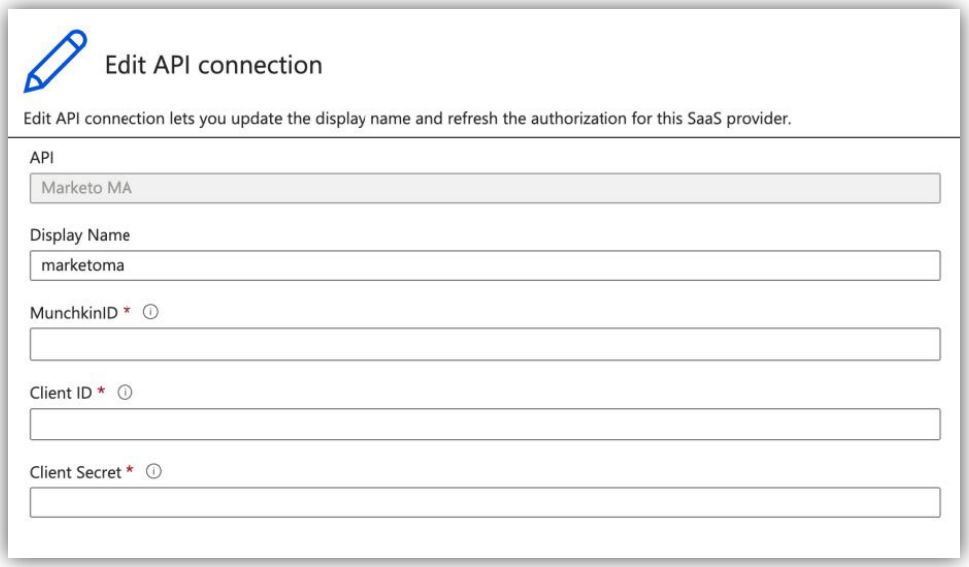


Giving our AI Agent tools

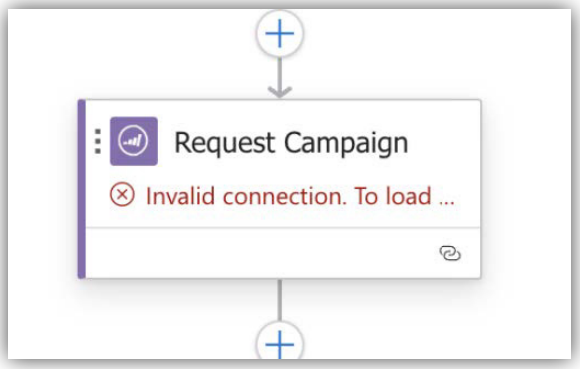
Next, in the left side menu select **Resource visualizer > General > Edit API connection**



Here fill in your Marketo REST API credentials and click **Save** To help generating the credentials, check out the authentication documentation here.



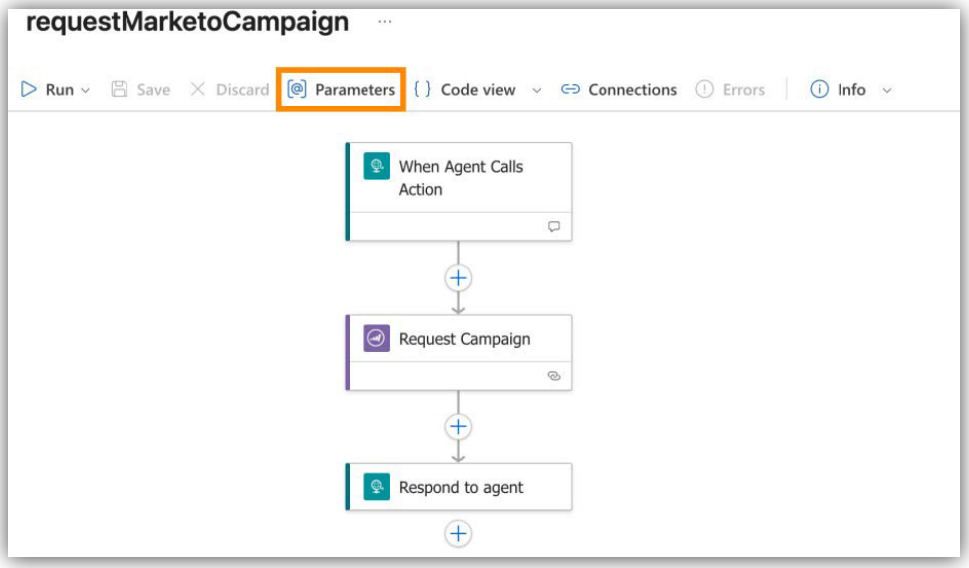
If you don't set this up, you will see an error in the Request Campaign step in the Logic App:



Giving our AI Agent tools

Next go back to the logic app and Edit

Edit the parameters

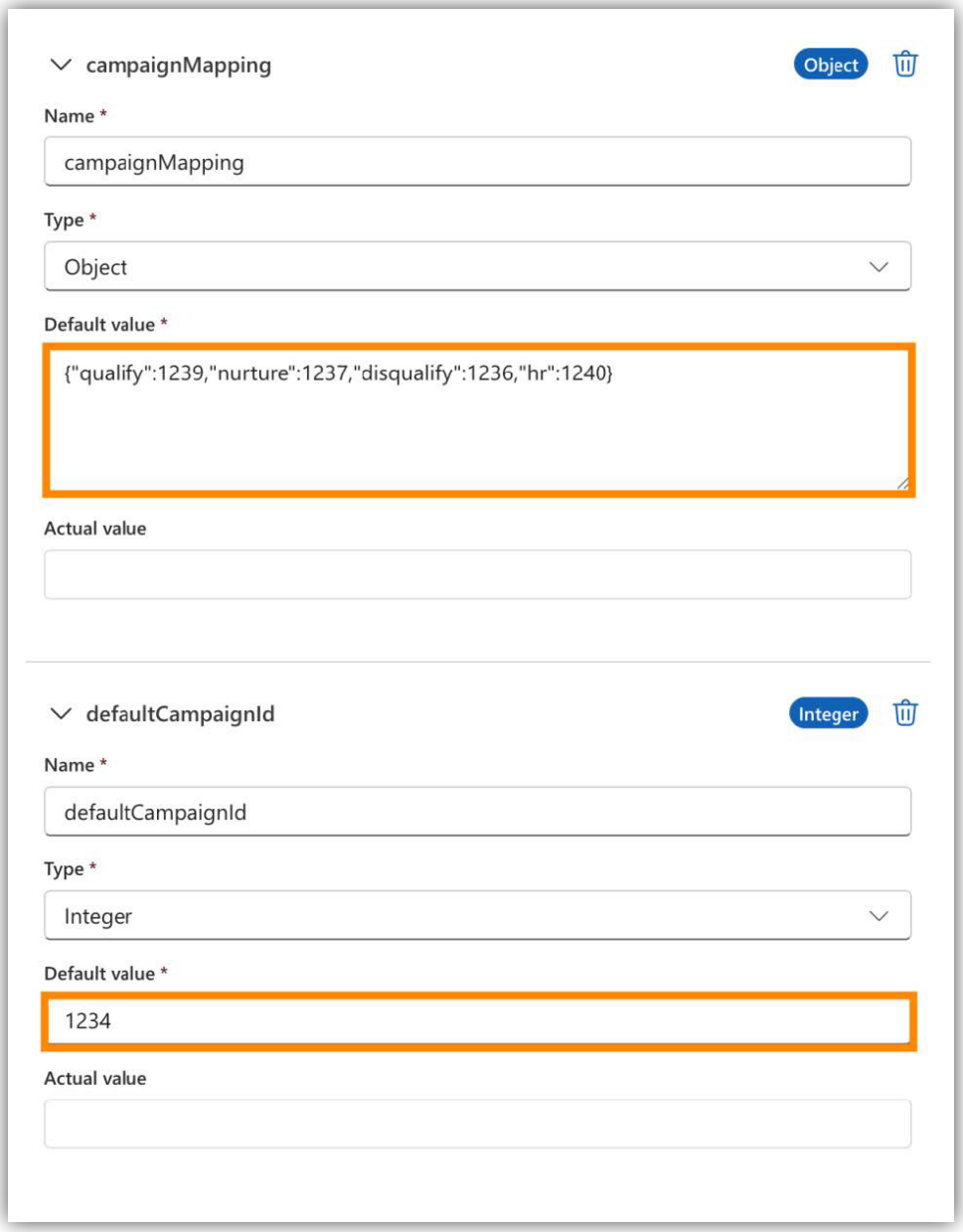


Giving our AI Agent tools

Here we can map the aiActions we've asked the agent to generate, to specific smart campaigns in Marketo. Update these to match your smart campaign IDs and add or change the AI actions you want. Also add a default smart campaign that will be called as a fallback when the AI Action isn't mapped to a smart campaign

You can find the Smart Campaign ID in the Marketo URL. Navigate to your Smart Campaign

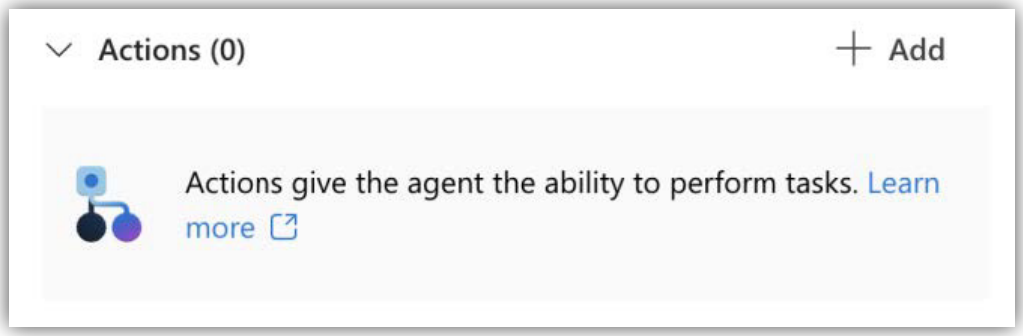
https://experience.adobe.com/#/@accountName/so:123-ABC-456/marketo-engage/classic/SC**1236**A1ZN38



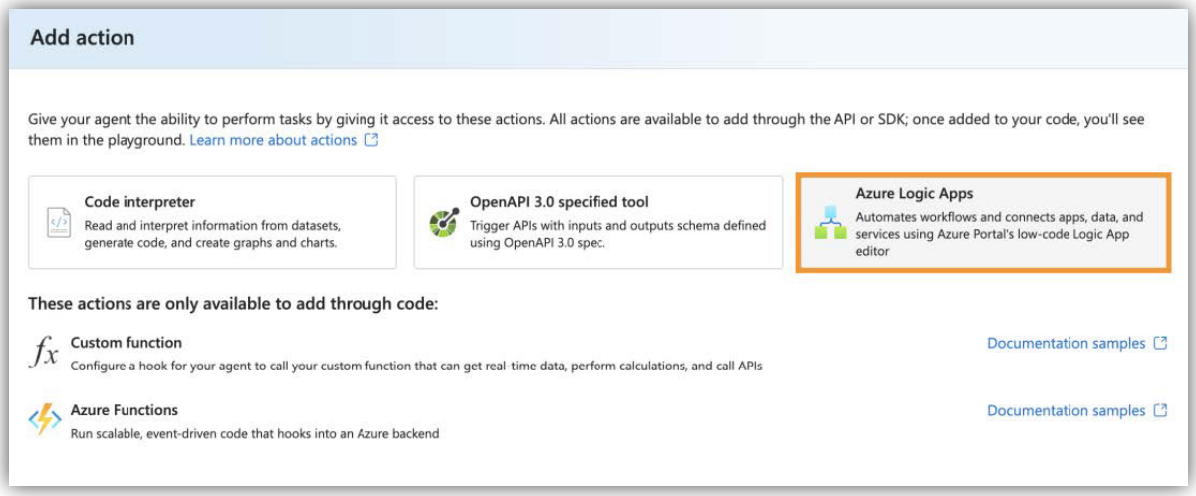
Register the Logic App as a Tool in the Agent

Now that we've created our actions we need to tell our AI Agent about them.

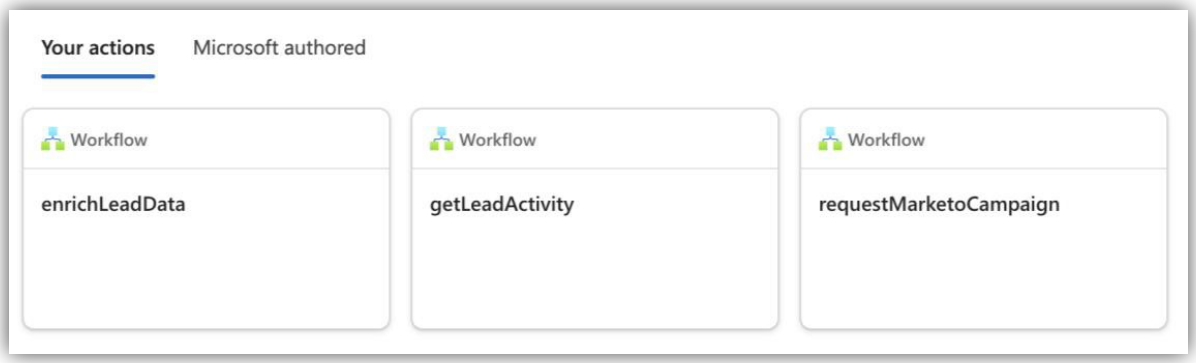
Back in AI Foundry, find the Actions section and click +Add



Select **Azure Logic Apps**



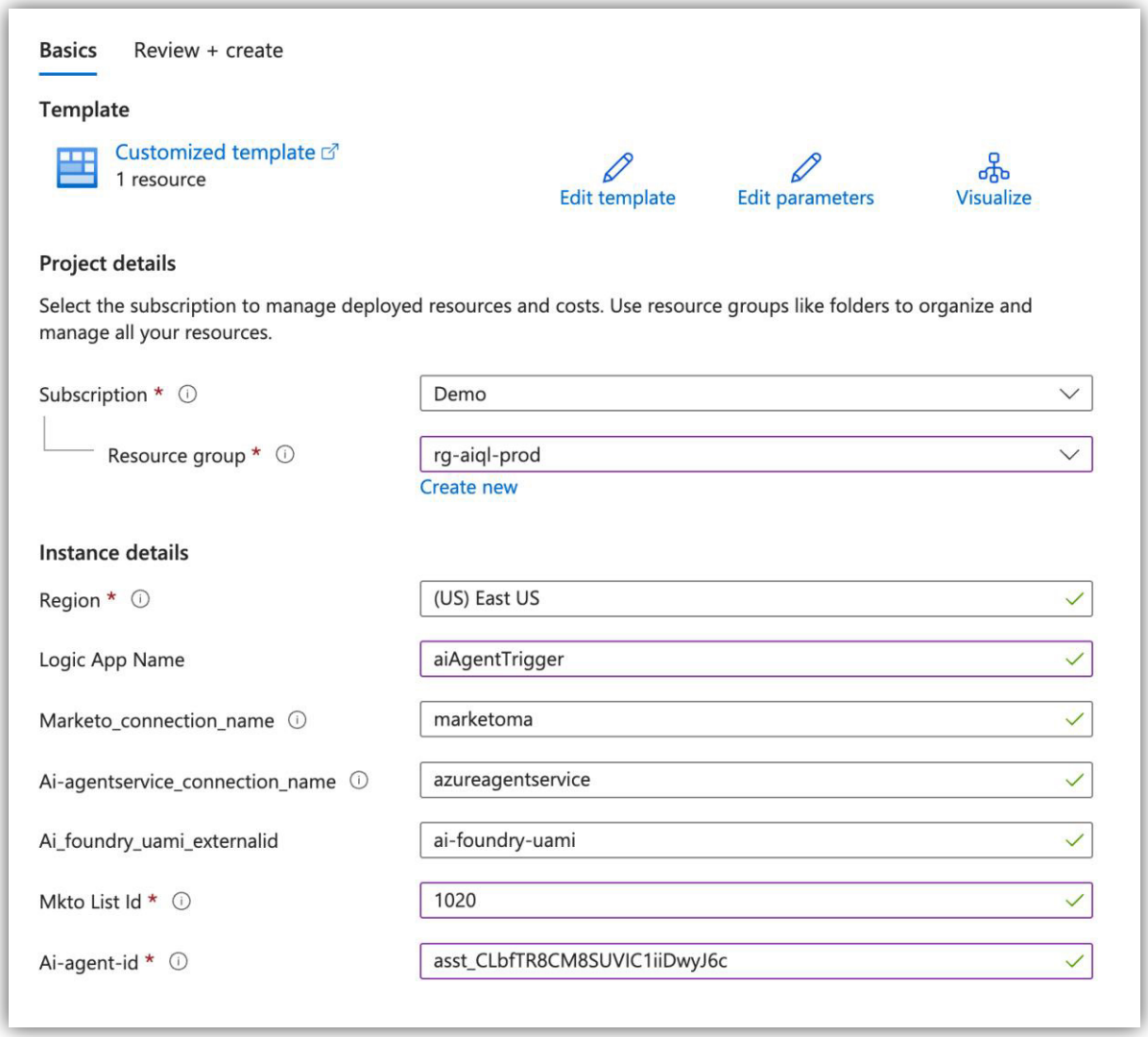
You should now see the Logic Apps created before. Select one and click **Create** then repeat this process for each action.



Setting up a trigger for our AI Agent

Now that our agent is setup with knowledge and tools, we need a way to trigger it. Specifically, we need to give it leads to review. Here again we will use a Logic App. I will set a schedule to check out Marketo Engage list every 10 minutes and process the leads inside. Here you have a template you can deploy to Azure.

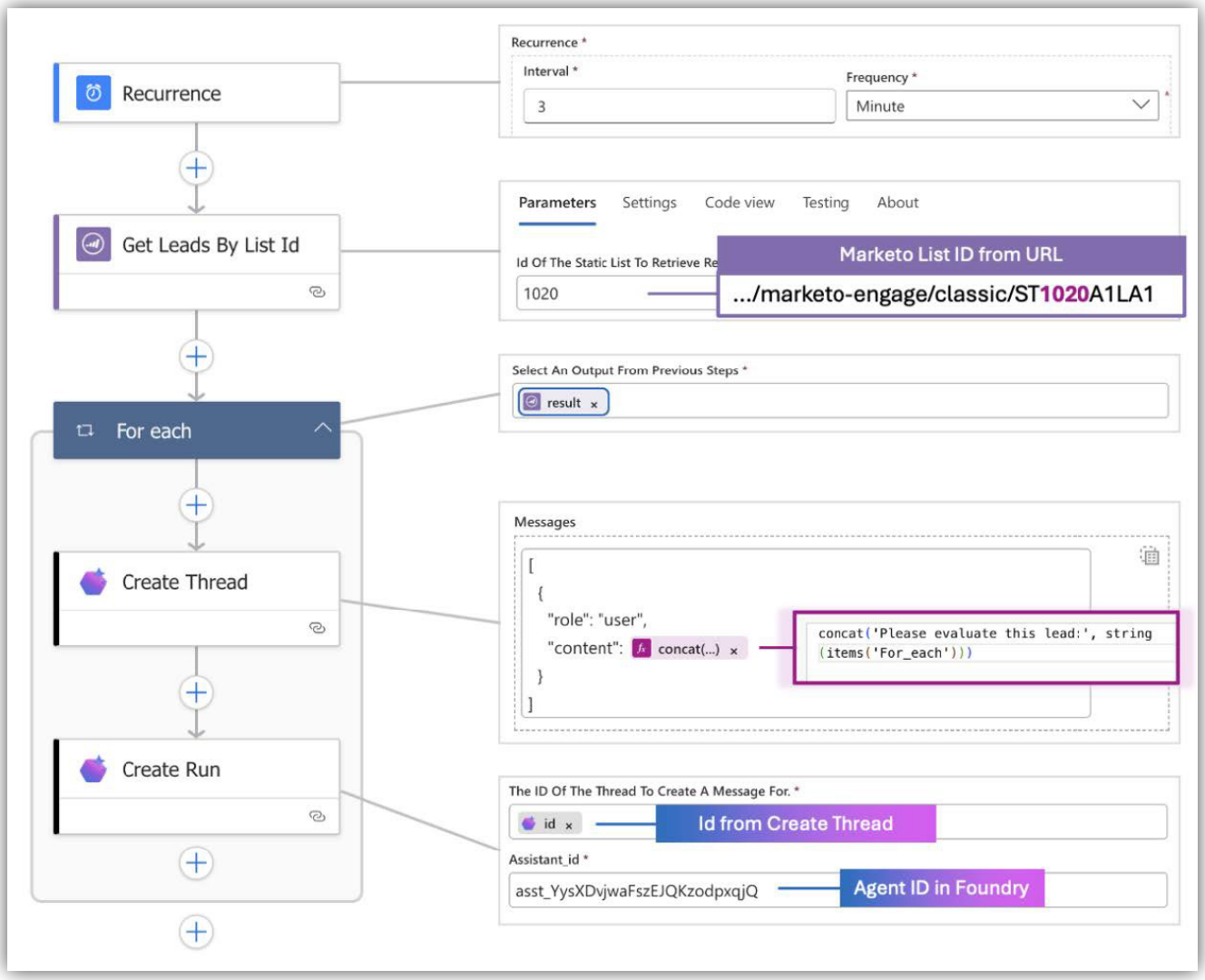
Deploy to Azure



Setting up a trigger for our AI Agent

We will use the Marketo static list we created.

Timer → Agent: A Logic App **Recurrence** trigger (e.g., every 10 minutes) pulls leads from our static list and calls the Agent for each record.



Once configured the app will begin checking our static list for leads on the schedule we've set and send them along to our agent to process.

Test (manual and evaluation runs)

1. In **Agents → Test**, paste a sample lead JSON (company/contact + recent activity).
2. Watch tool calling:
3. If data is missing, it should call **Enrich**.
4. If strong fit/intent, it should call **Request Campaign** and include token values.
5. Validate the **Response** from the Logic App and confirm the **Smart Campaign** was triggered with the right **program tokens.**

For repeatable testing, use **Evaluations/Prompt Flow** to run batches of sample leads and compare outcomes.

Creating Additional Logic Apps

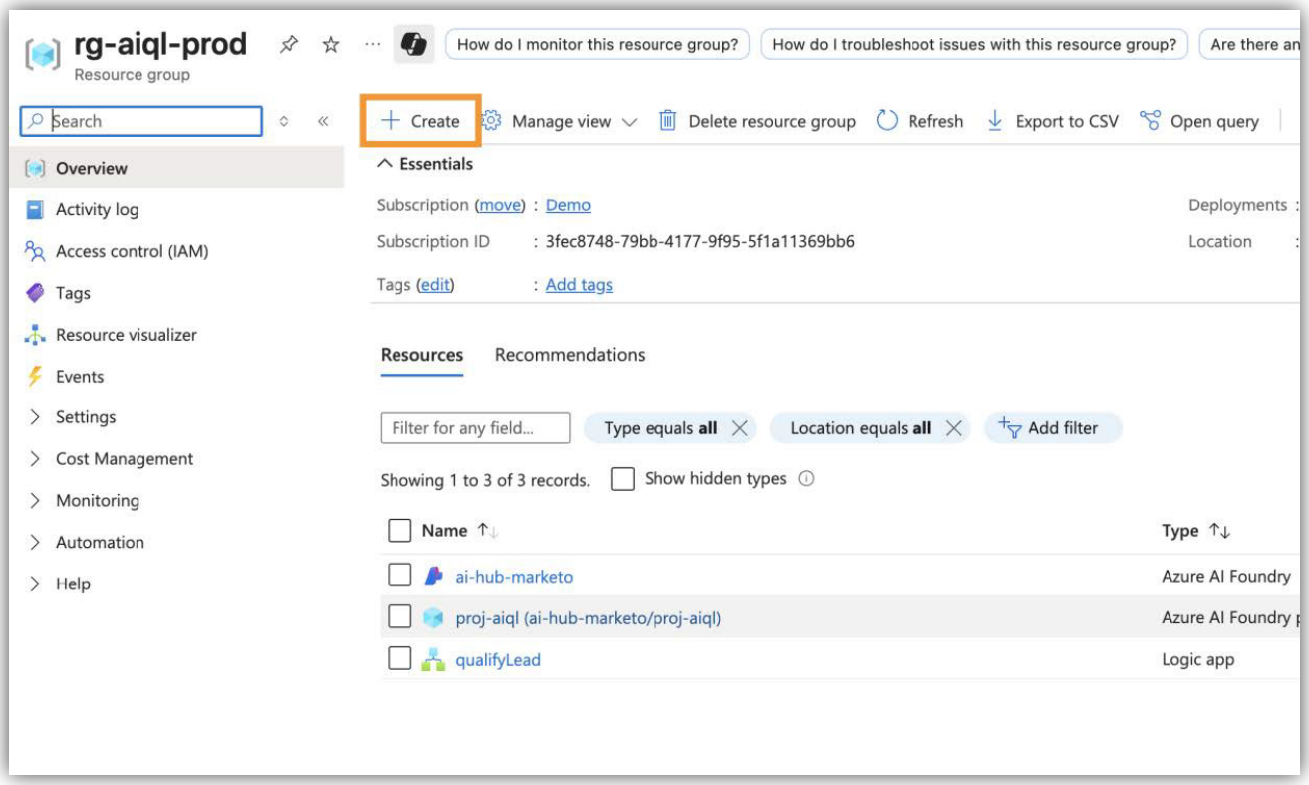
I've created templates for each of the logic apps I'm using, however you may want to create your own from scratch. This is how you can do this.

In order to use a Logic App with our AI Agent it must meet specific requirements

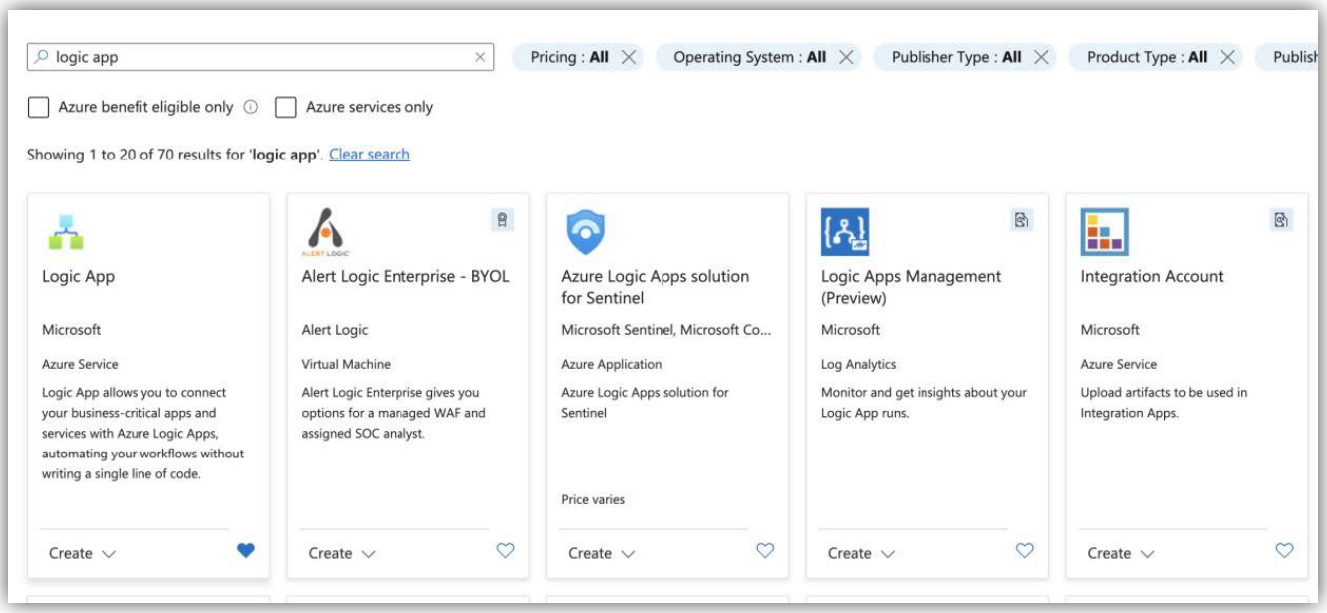
1. It must use a **Consumption** plan
2. It must be in the same subscription as your agent
3. It must have a **When a HTTP request is received** trigger with a description
4. It must have an **Response** action as well to pass back data to the agent

If you do not see your Logic App listed in your available actions, ensure it meets all of these criteria.

Creating Additional Logic Apps

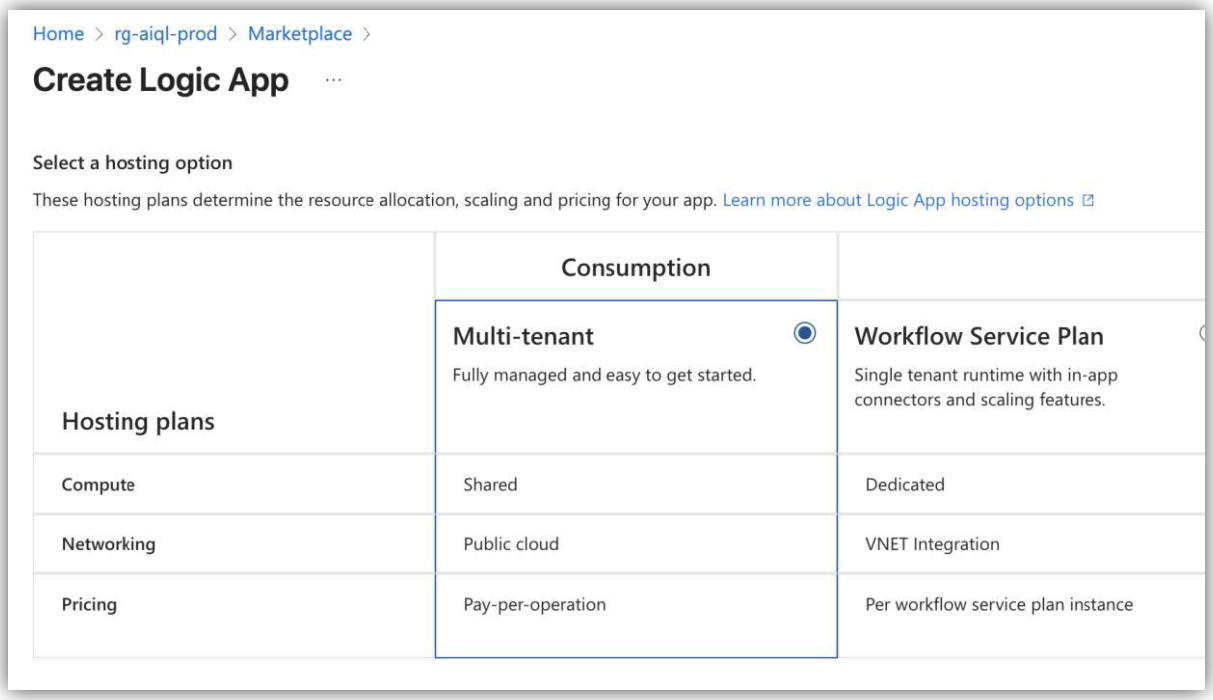


Select Logic App > Create

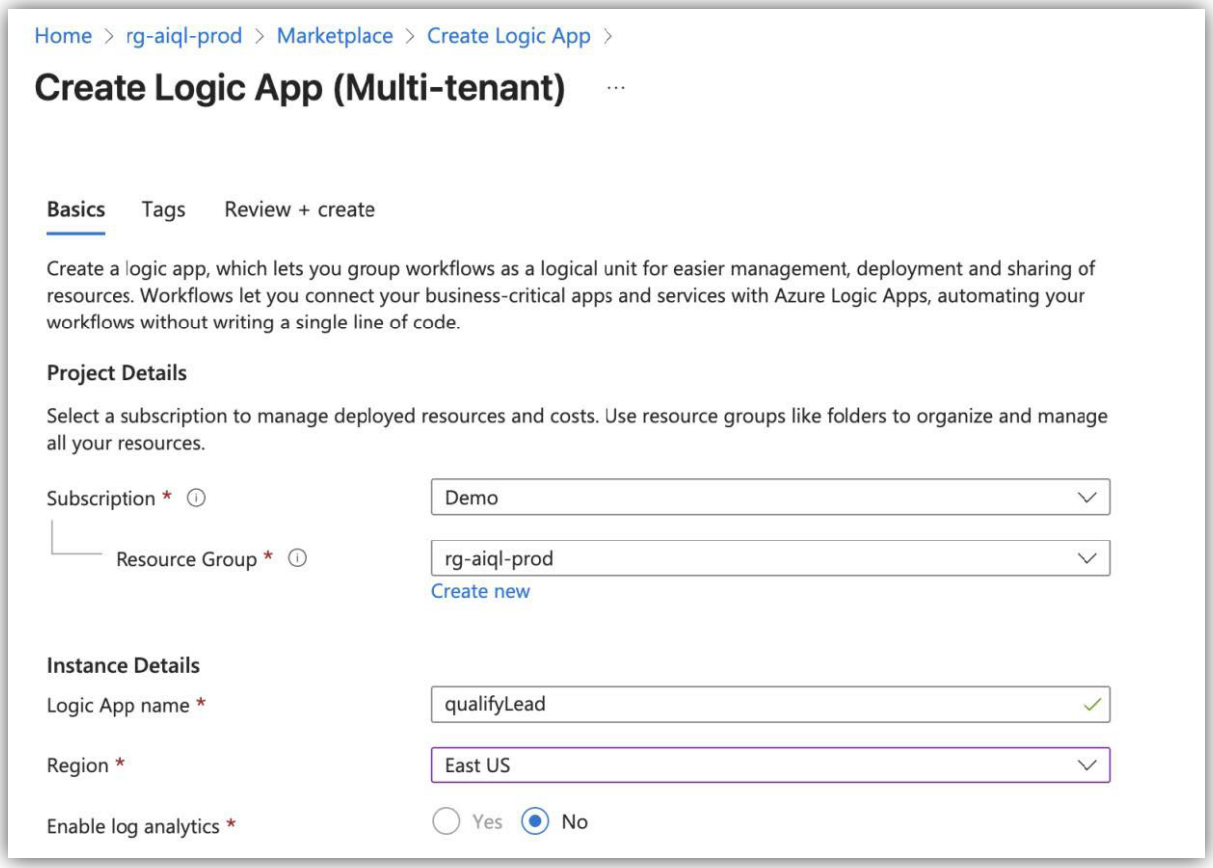


Creating Additional Logic Apps

Select **Consumption**

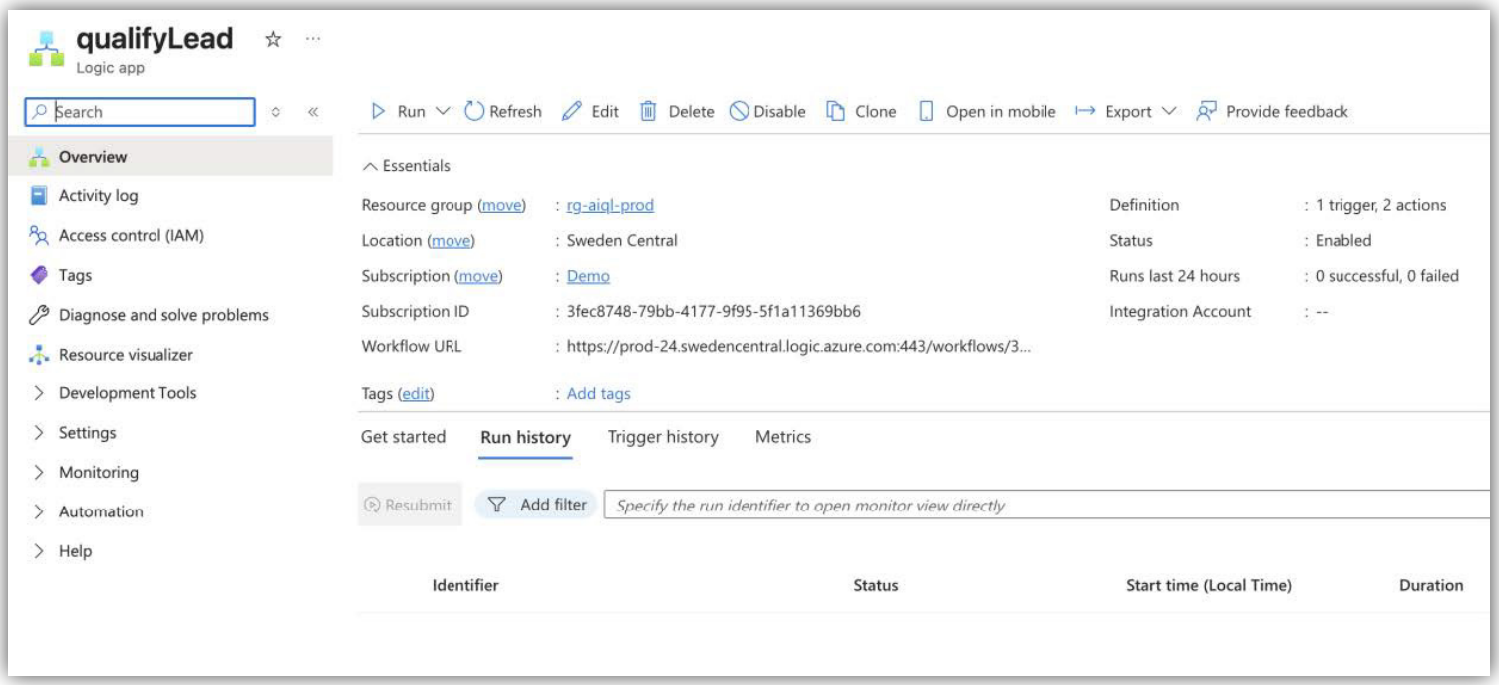


Give the logic app a clear name and click **Review + Create**



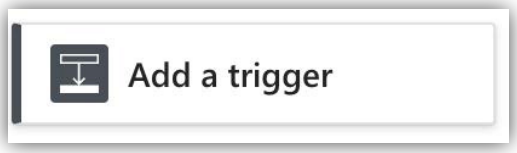
Creating Additional Logic Apps

Once deployed go to the logic app and click **Edit**

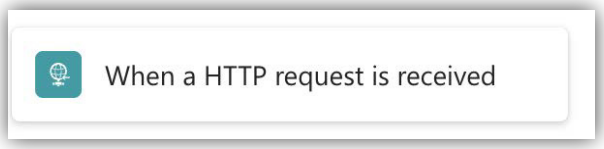


For requesting Smart Campaigns our logic app will need 3 parts:

Click **Add a trigger**

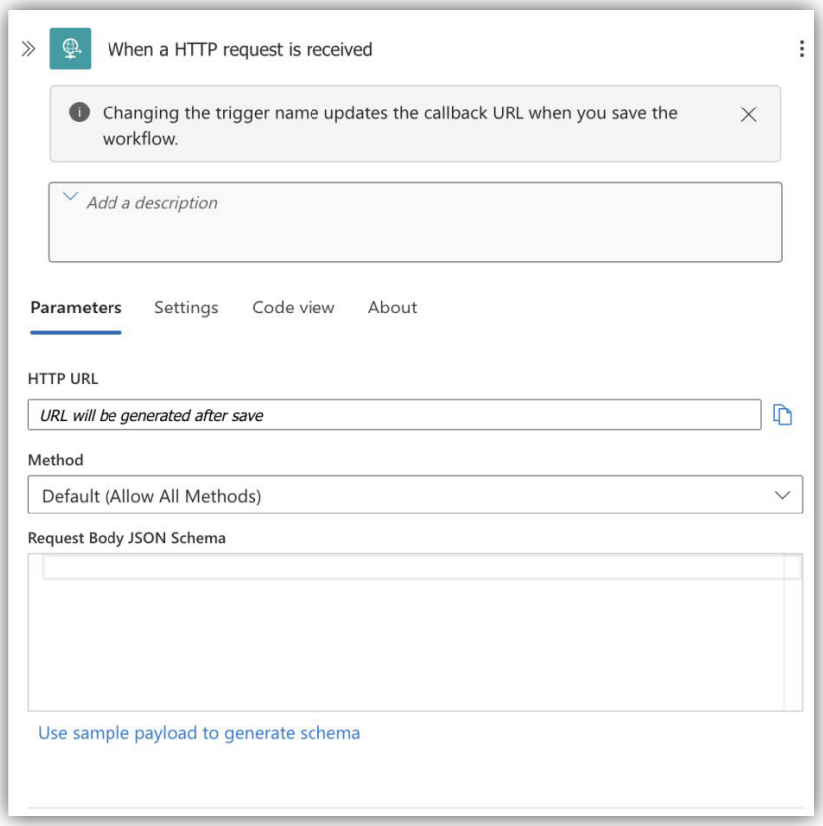


Select **When a HTTP request is received**



Creating Additional Logic Apps

Important – The HTTP request must have a description



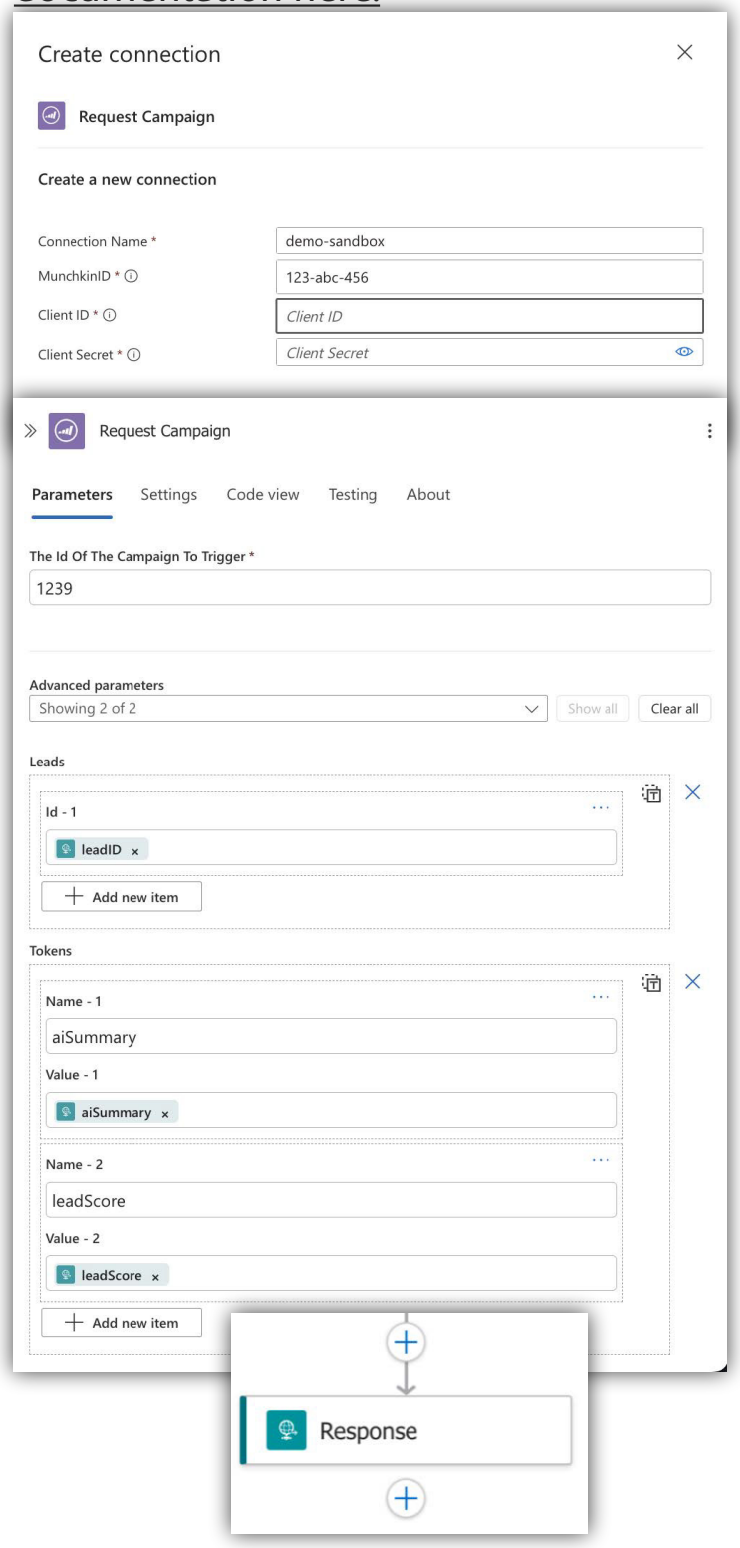
Under Request Body JSON Schema, enter some version of the schema below. The only required parameter is leadID, which allows the agent to pass in the correct lead ID. The other parameters will be tokens that will be passed to Marketo Engage.

```
{
 "type": "object",
 "properties": {
  "aiSummary": {
   "description": "AI generated summary of why the lead is qualified",
   "type": "string"
  },
  "leadScore": {
   "description": "score 1-100 for the lead dec",
   "type": "string"
  },
  "leadID": {
   "description": "Marketo lead id",
   "type": "integer"
  }
 }
}
```

Creating Additional Logic Apps

Next add an action and select Request Campaign

If you haven't created a Marketo connection yet you will need to create one with your munchkin ID, client ID and client secret. Check out the authentication documentation here.



Ongoing improvement loop

1. Review **Sales feedback** and explainability notes attached to qualified leads.
2. Tune **ICP** and **Instructions** (e.g., adjust how owner vs buyer is detected).
3. Expand **Tools** (e.g., add “Data Health Review”, “Prospecting” later).
4. Gradually reduce person-in-the-loop rules as the system's confidence and accuracy improve.

Quick Checklist

1. Azure Account Created
2. Marketo API Credentials Obtained
3. Azure Resource Group Created
4. Azure AI Studio Workspace & Agent Deployed
5. Agent Instructions Defined
6. ICP Document Uploaded to Vector Store
7. Program & Tokens Created
8. Watch List Created
9. Smart Campaigns Created & Activated (Qualify, Disqualify, Nurture, HR)
10. enrichLeadData Logic App Deployed & Configured
11. getLeadActivity Logic App Deployed & Configured
12. requestMarketoCampaign Logic App Deployed & Configured
13. All Logic Apps Registered as Tools in Agent
14. Trigger Logic App Deployed & Scheduled
15. Manual End-to-End Test Completed Successfully
16. Evaluation Run with Batch of Leads Completed


Adobe