Guide for creating AI Lead Qualification Agent in Azure AI Studio

Josh Arrington
Partner, CMTO
Kapturall

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, 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

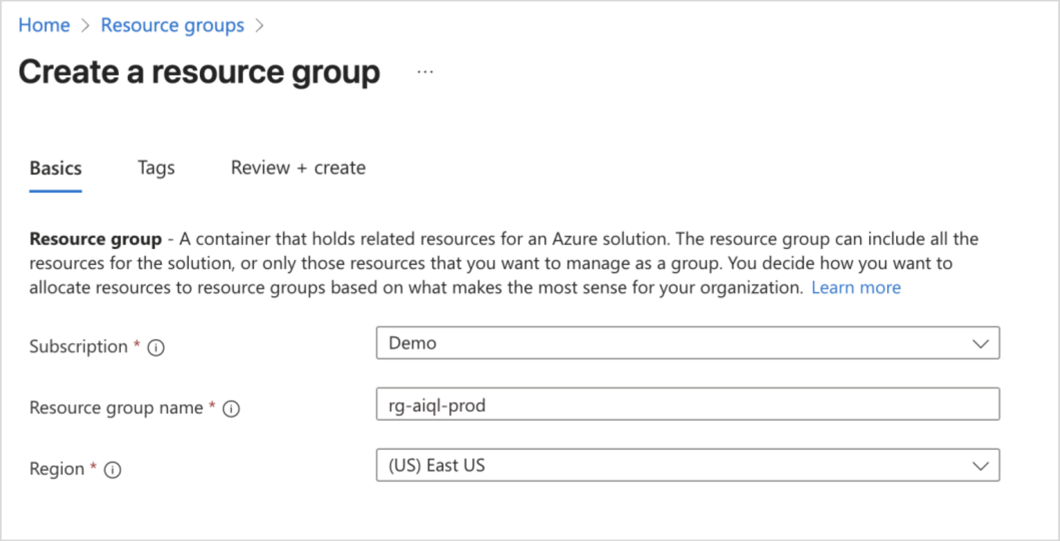
- **Azure Account –** If you don't have an Azure account you can create on for free here https://azure.microsoft.com
- **Marketo API Credentials:** Munchkin ID, REST URL, Client ID / Client Secret,

---

Create a Resource Group

1. Azure Portal → Resource groups → Create.

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



Home > Resource groups >

Create a resource group ...

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Subscription * : Demo

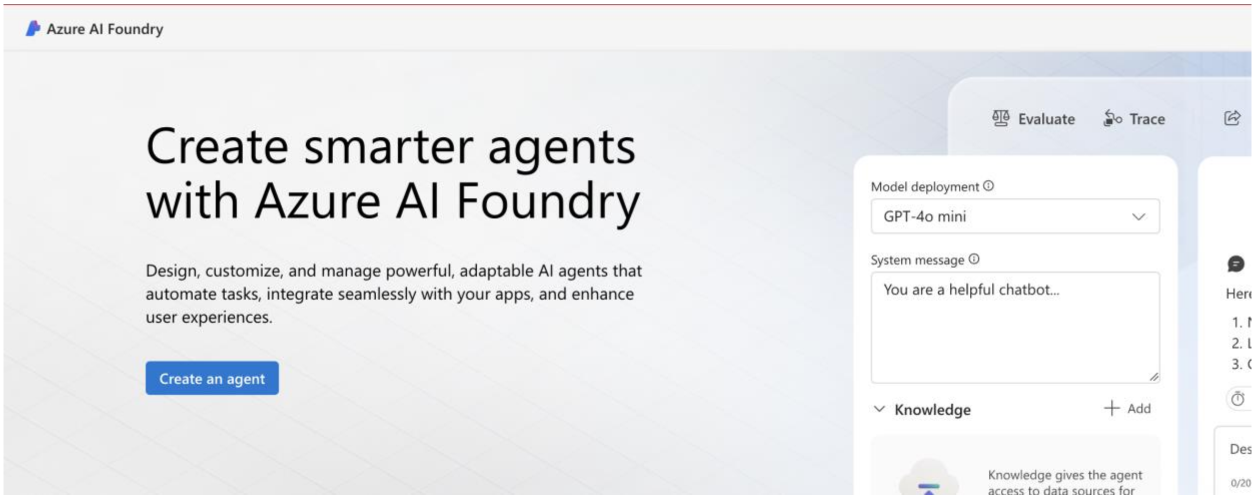
Resource group name * : rg-aiql-prod

Region * : (US) East US

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").



Create smarter agents with Azure AI Foundry

Design, customize, and manage powerful, adaptable AI agents that automate tasks, integrate seamlessly with your apps, and enhance user experiences.

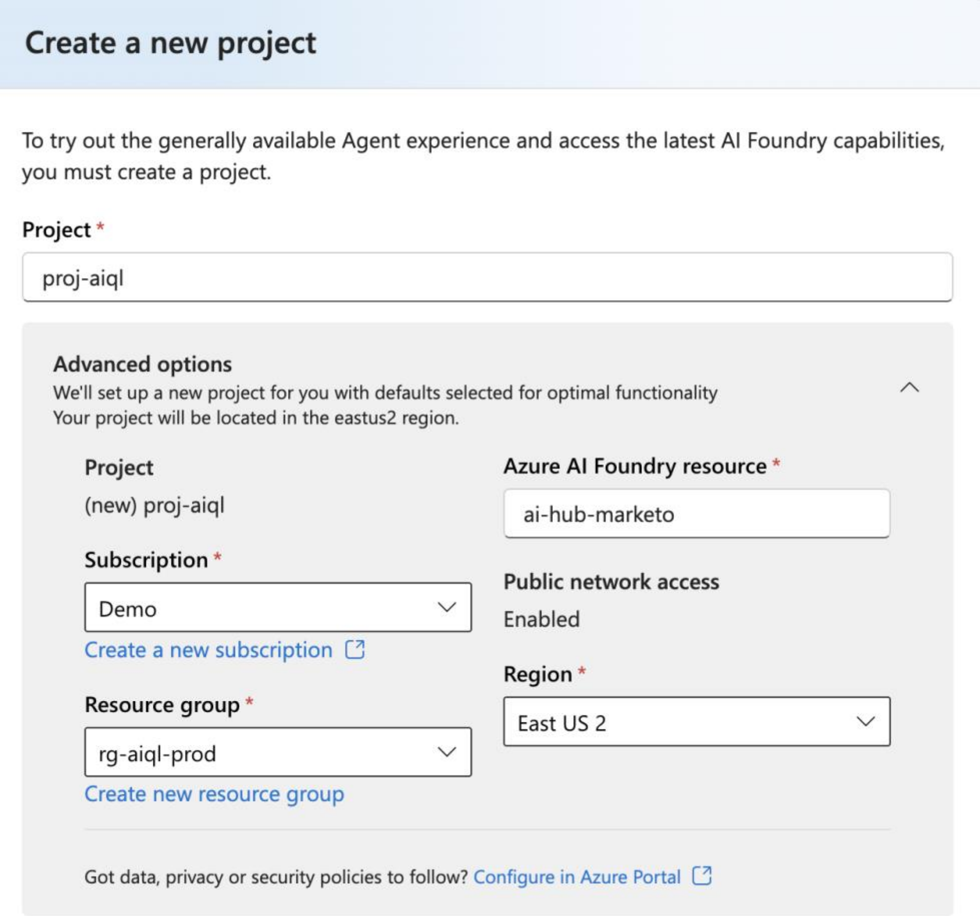
Model deployment: GPT-4o mini

System message: You are a helpful chatbot...

Knowledge gives the agent access to data sources for

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 a new project

To try out the generally available Agent experience and access the latest AI Foundry capabilities, you must create a project.

Project *: proj-aiql

Advanced options

We'll set up a new project for you with defaults selected for optimal functionality
Your project will be located in the eastus2 region.

Project: (new) proj-aiql

Subscription *: Demo

Resource group *: rg-aiql-prod

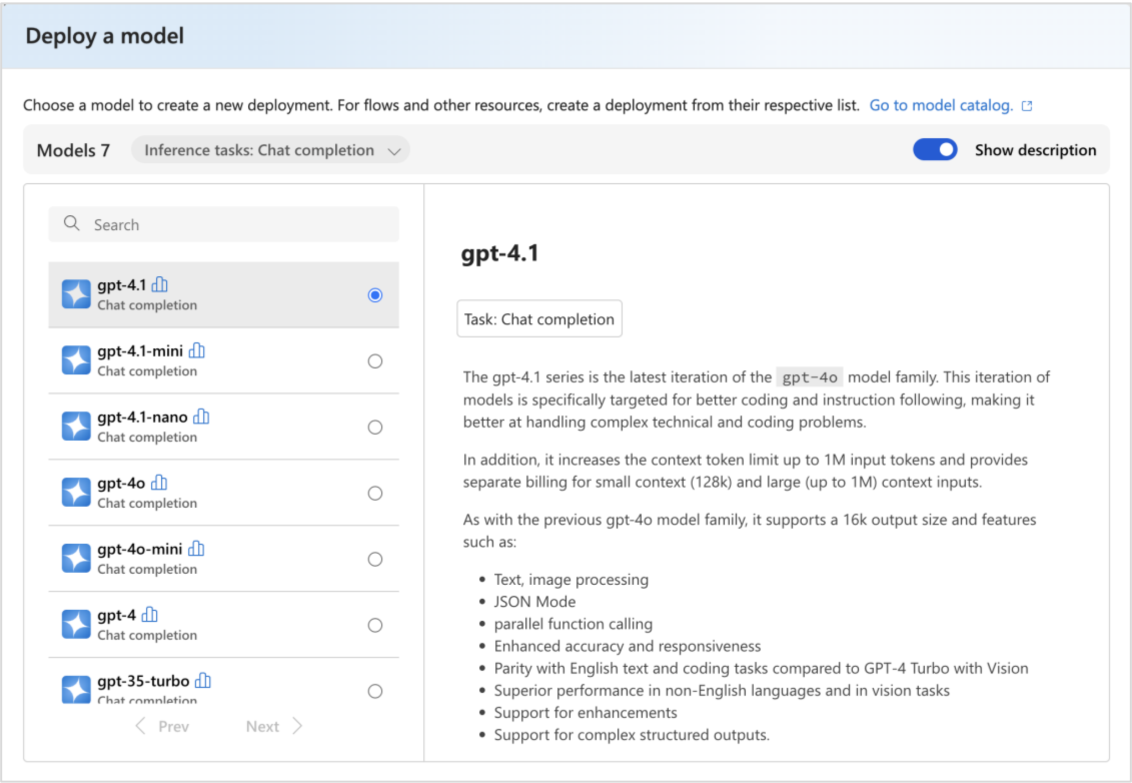
Azure AI Foundry resource *: ai-hub-marketo

Public network access: Enabled

Region *: East US 2

Got data, privacy or security policies to follow? Configure in Azure Portal

3. Select a model



Deploy a model

Choose a model to create a new deployment. For flows and other resources, create a deployment from their respective list. Go to model catalog.

gpt-4.1

Task: Chat completion

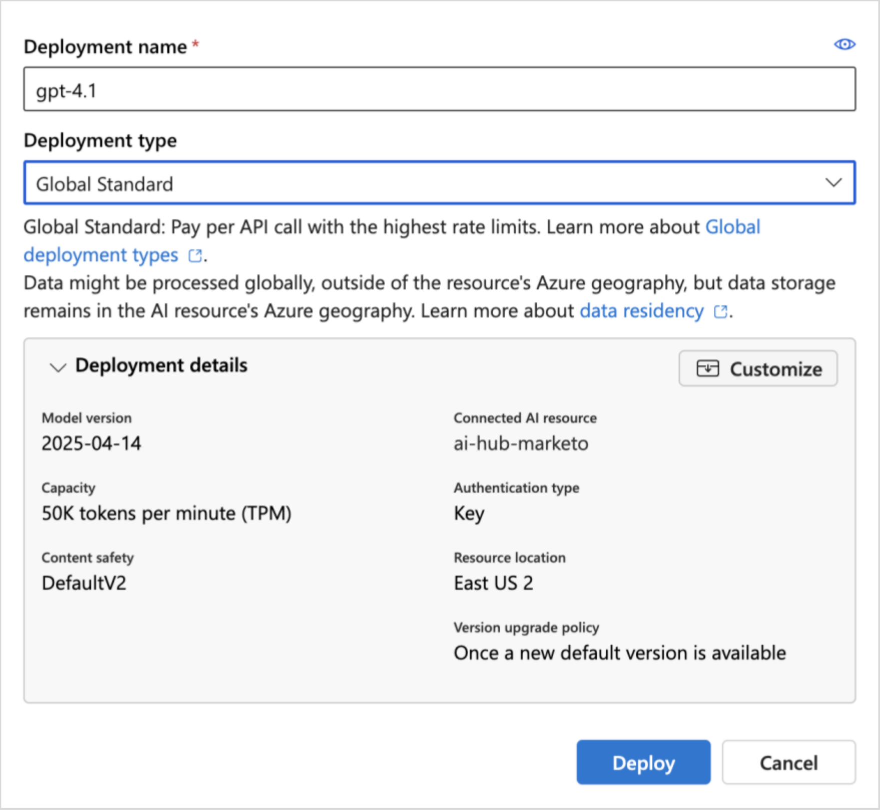
The gpt-4.1 series is the latest iteration of the `gpt-4o` model family. This iteration of models is specifically targeted for better coding and instruction following, making it better at handling complex technical and coding problems.

In addition, it increases the context token limit up to 1M input tokens and provides separate billing for small context (128k) and large (up to 1M) context inputs.

As with the previous gpt-4o model family, it supports a 16k output size and features such as:

- Text, image processing
- JSON Mode
- parallel function calling
- Enhanced accuracy and responsiveness
- Parity with English text and coding tasks compared to GPT-4 Turbo with Vision
- Superior performance in non-English languages and in vision tasks
- Support for enhancements
- Support for complex structured outputs.

4. Deploy your agent



Deployment name *

gpt-4.1

Deployment type

Global Standard

Global Standard: Pay per API call with the highest rate limits. Learn more about Global deployment types.

Data might be processed globally, outside of the resource's Azure geography, but data storage remains in the AI resource's Azure geography. Learn more about data residency.

Deployment details

Model version: 2025-04-14

Connected AI resource: ai-hub-marketo

Capacity: 50K tokens per minute (TPM)

Authentication type: Key

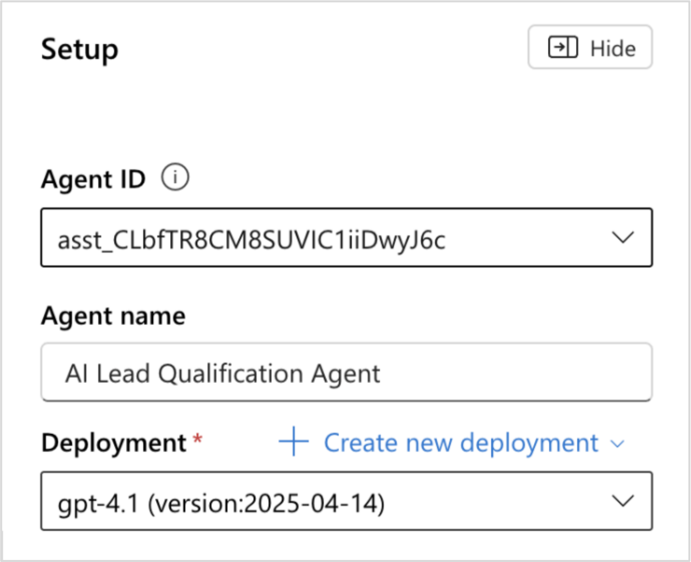
Content safety: DefaultV2

Resource location: East US 2

Version upgrade policy: Once a new default version is available

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

5. Give the agent a name in the right panel



Setup

Agent ID: asst_CLbfTR8CM8SUVIC1iiDwyJ6c

Agent name: AI Lead Qualification Agent

Deployment *: gpt-4.1 (version:2025-04-14)

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).

Decision process

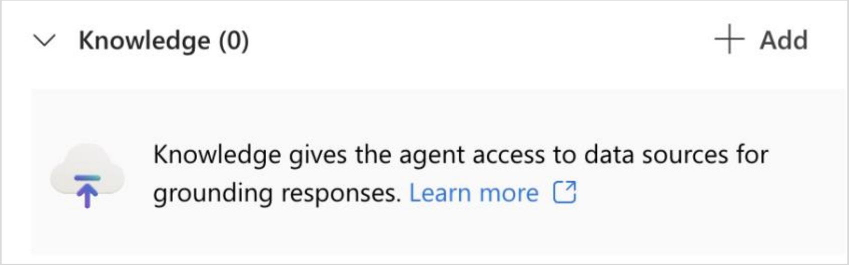
1. Always check the ICP fit in the vector store first.
2. Weigh recent lead activity and engagement.
3. Use company firmographics to validate alignment with corporate real estate opportunities.
4. Choose the most appropriate aiAction.
5. Fill in all fields with clear, defensible reasoning.

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

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

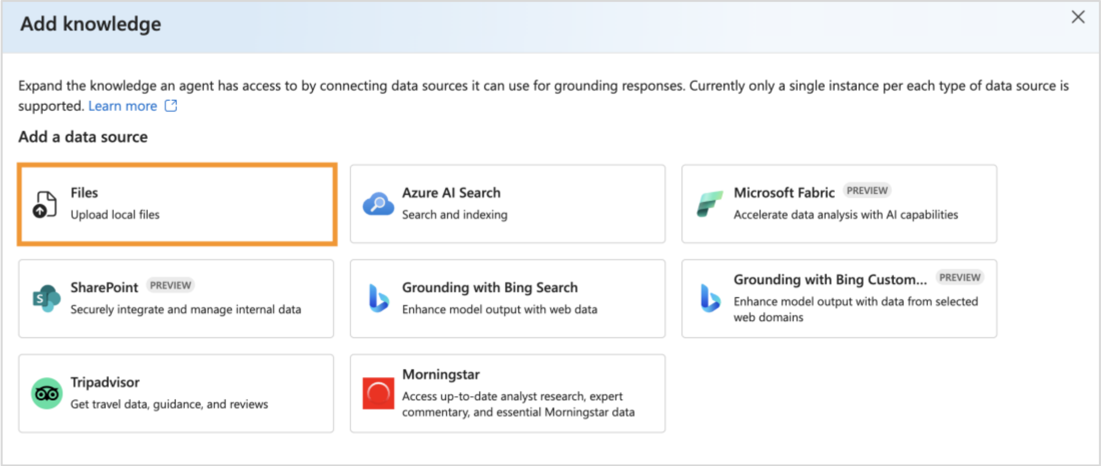
Load Knowledge into a Vector Store

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



Knowledge gives the agent access to data sources for grounding responses. Learn more

2. Select **Files**



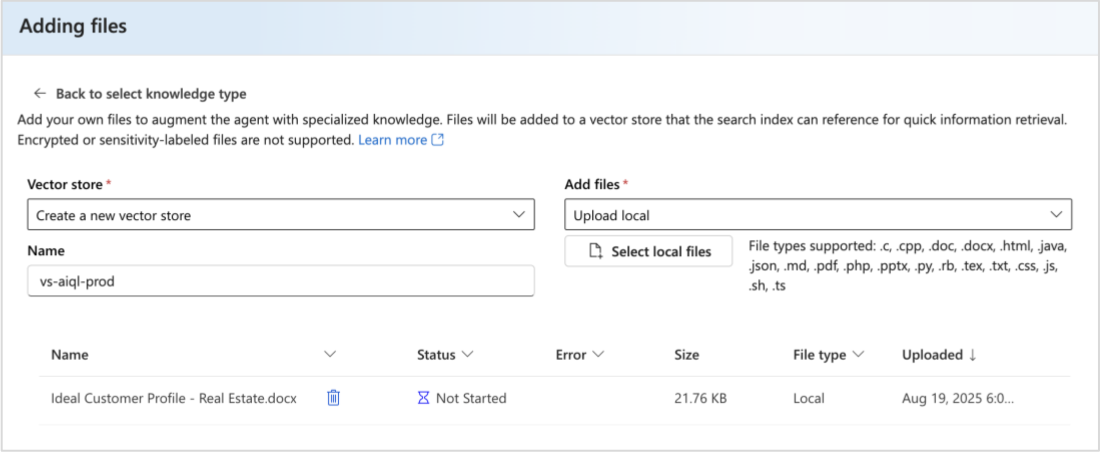
Add knowledge

Expand the knowledge an agent has access to by connecting data sources it can use for grounding responses. Currently only a single instance per each type of data source is supported. Learn more

Add a data source

- Files – Upload local files
- Azure AI Search – Search and indexing
- Microsoft Fabric PREVIEW – Accelerate data analysis with AI capabilities
- SharePoint PREVIEW – Securely integrate and manage internal data
- Grounding with Bing Search – Enhance model output with web data
- Grounding with Bing Custom... PREVIEW – Enhance model output with data from selected web domains
- Tripadvisor – Get travel data, guidance, and reviews
- Morningstar – Access up-to-date analyst research, expert commentary, and essential Morningstar data

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



Adding files

← Back to select knowledge type

Add your own files to augment the agent with specialized knowledge. Files will be added to a vector store that the search index can reference for quick information retrieval. Encrypted or sensitivity-labeled files are not supported. Learn more

Vector store *: Create a new vector store

Name: vs-aiql-prod

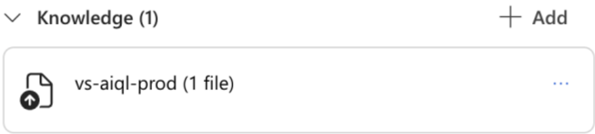
Add files *: Upload local

Select local files

File types supported: .c, .cpp, .doc, .docx, .html, .java, .json, .md, .pdf, .php, .pptx, .py, .rb, .tex, .txt, .css, .js, .sh, .ts

Name	Status	Error	Size	File type	Uploaded ↓
Ideal Customer Profile - Real Estate.docx	Not Started		21.76 KB	Local	Aug 19, 2025 6:0...

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



Knowledge (1) + Add

vs-aiql-prod (1 file) ···

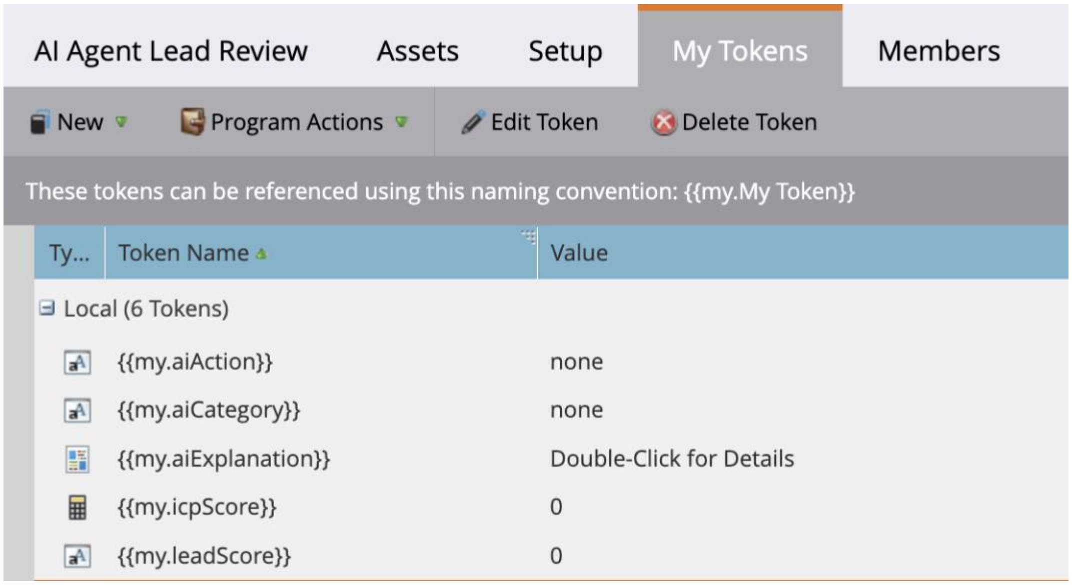
Why vector store? It lets the agent find the **most relevant passages semantically**, not just exact keywords.

Setting up Marketo

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 when triggering our smart campaign. We need to create a smart campaign for each of these actions.

Create Marketo Program & Tokens

Create program tokens for each value we've asked the agent 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.

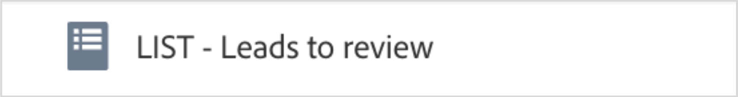


These tokens can be referenced using this naming convention: {{my.My Token}}

Ty...	Token Name	Value
Local (6 Tokens)		
	{{my.aiAction}}	none
	{{my.aiCategory}}	none
	{{my.aiExplanation}}	Double-Click for Details
	{{my.icpScore}}	0
	{{my.leadScore}}	0

Create Watch List

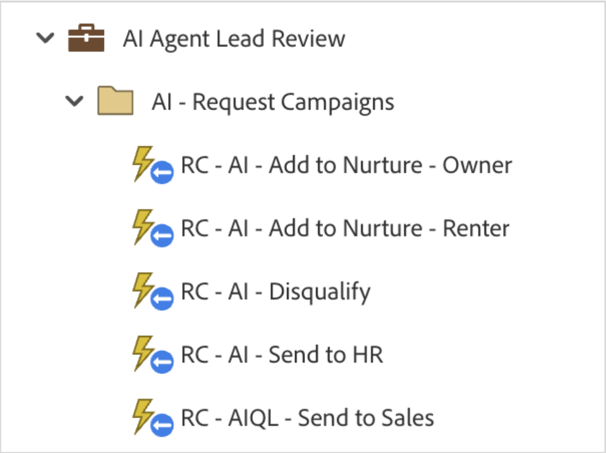
Create a static list. We will pull leads from this list for our agent to process



LIST - Leads to review

You can find the list ID in the Marketo URL. Navigate to your static list https://experience.adobe.com/#/@accountName/so:123-ABC-456/marketo-engage/classic/ST1020A1LA1

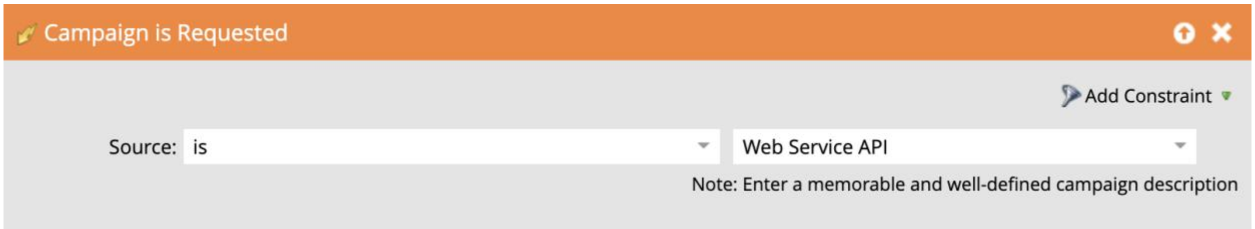
Create Smart Campaigns



- AI Agent Lead Review
  - AI - Request Campaigns
    - RC - AI - Add to Nurture - Owner
    - RC - AI - Add to Nurture - Renter
    - RC - AI - Disqualify
    - RC - AI - Send to HR
    - RC - AIQL - Send to Sales

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 is Requested

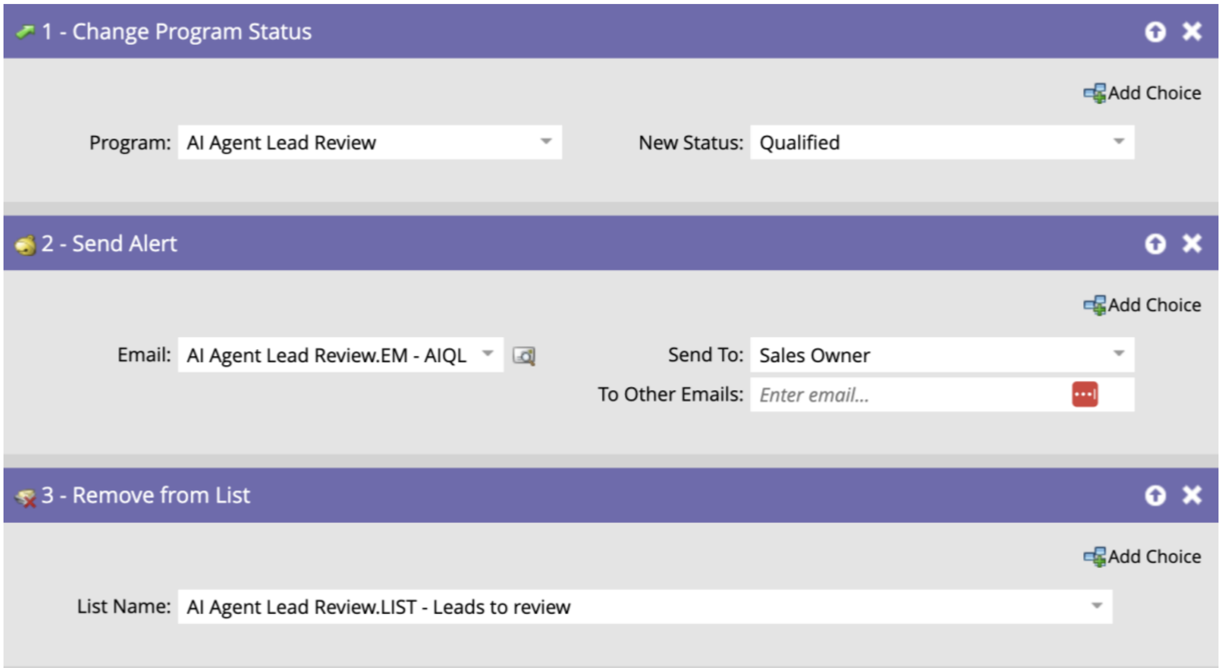
Add Constraint

Source: is | Web Service API

Note: Enter a memorable and well-defined campaign description

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.



1 - Change Program Status

Program: AI Agent Lead Review

New Status: Qualified

2 - Send Alert

Email: AI Agent Lead Review.EM - AIQL

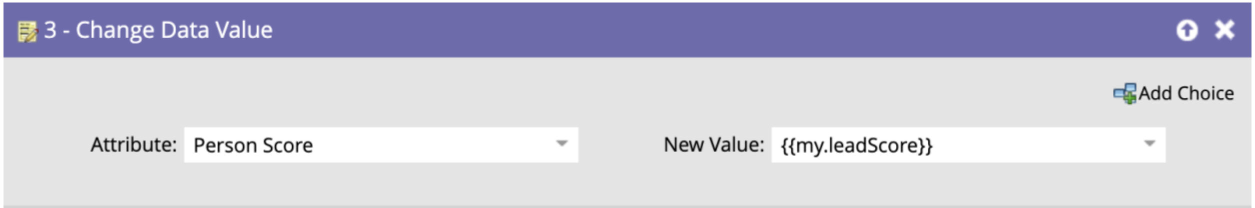
Send To: Sales Owner

To Other Emails: *Enter email...*

3 - Remove from List

List Name: AI Agent Lead Review.LIST - Leads to review

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

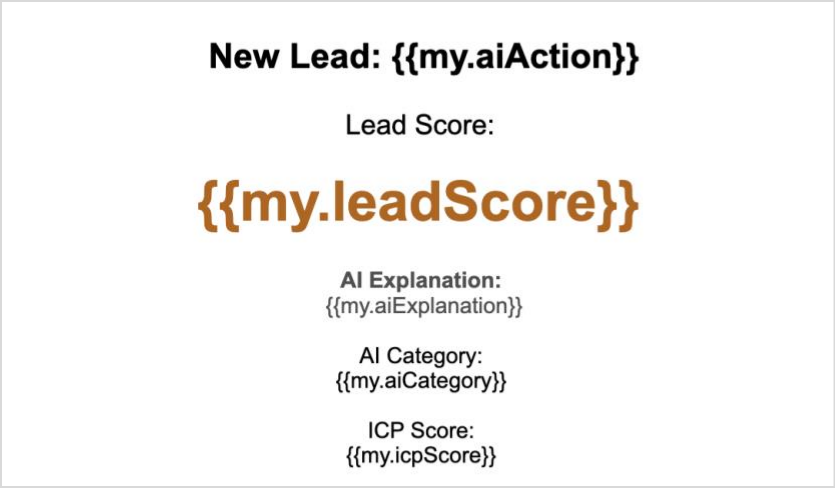


3 - Change Data Value

Attribute: Person Score

New Value: {{my.leadScore}}

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



New Lead: {{my.aiAction}}

Lead Score:

{{my.leadScore}}

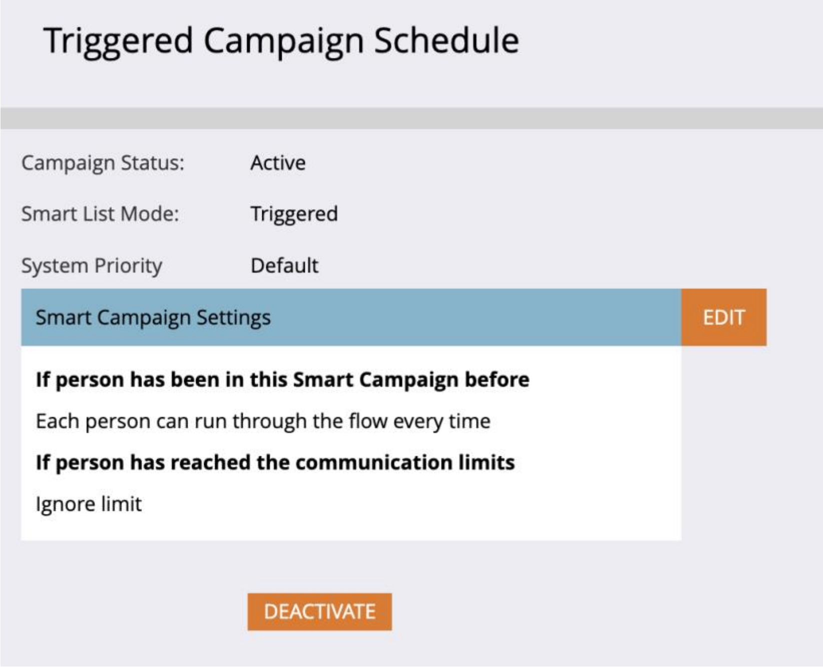
AI Explanation:
{{my.aiExplanation}}

AI Category:
{{my.aiCategory}}

ICP Score:
{{my.icpScore}}

Campaign Schedule

Activate the campaigns so they are now available



Triggered Campaign Schedule

Campaign Status: Active

Smart List Mode: Triggered

System Priority Default

Smart Campaign Settings

If person has been in this Smart Campaign before

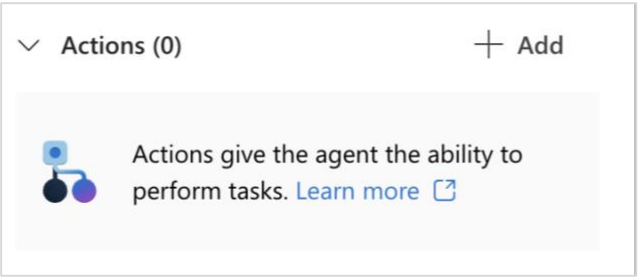
Each person can run through the flow every time

If person has reached the communication limits

Ignore limit

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.



Actions give the agent the ability to perform tasks. Learn more

We will create three key actions:

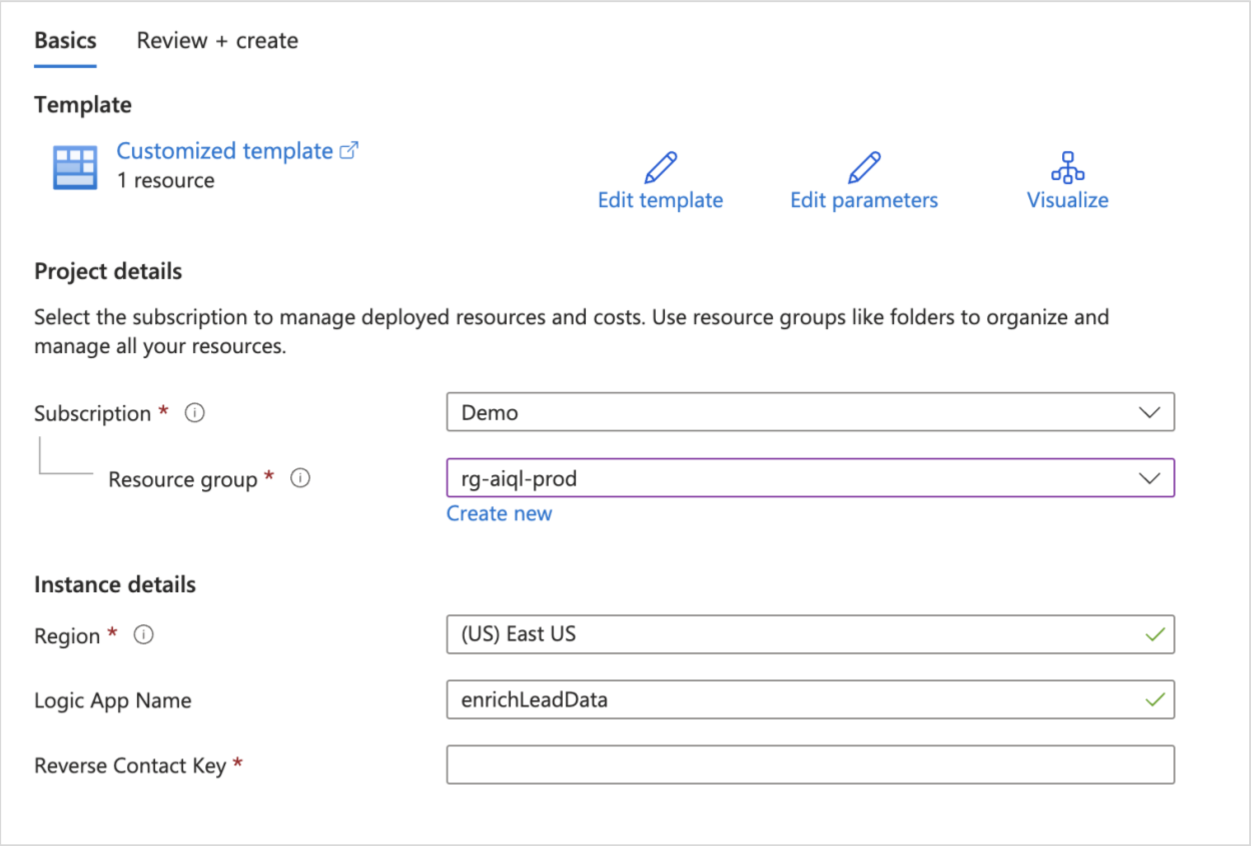
- enrichLeadData
- getLeadActivity
- requestMarketoCampaign

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.



Template

Customized template
1 resource

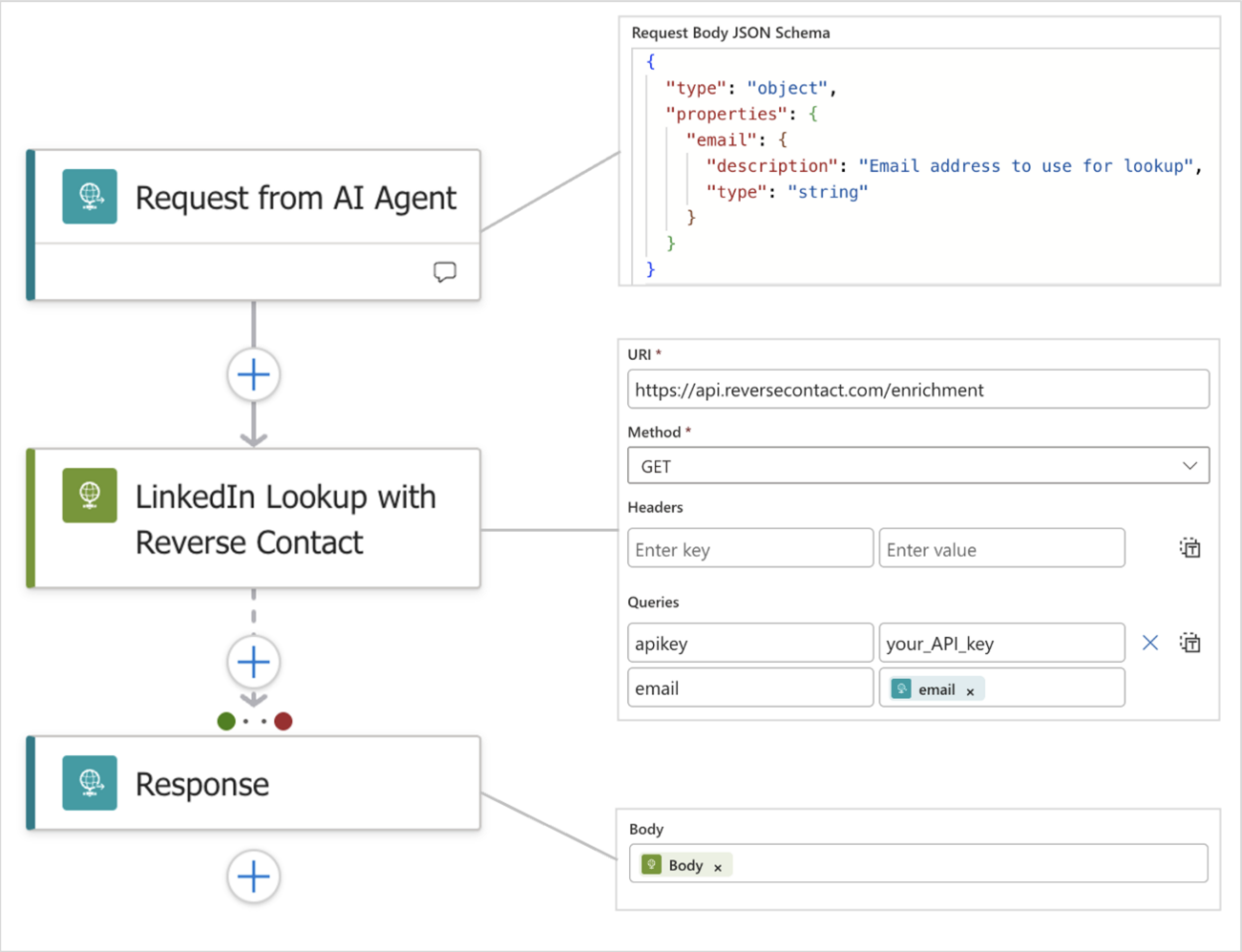
Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Demo
Resource group *	rg-aiql-prod

Instance details

Region *	(US) East US ✓
Logic App Name	enrichLeadData ✓
Reverse Contact Key *	



Request Body JSON Schema

```
{
  "type": "object",
  "properties": {
    "email": {
      "description": "Email address to use for lookup",
      "type": "string"
    }
  }
}
```

URI *
https://api.reversecontact.com/enrichment

Method *
GET

Headers

Enter key	Enter value

Queries

apikey	your_API_key
email	email

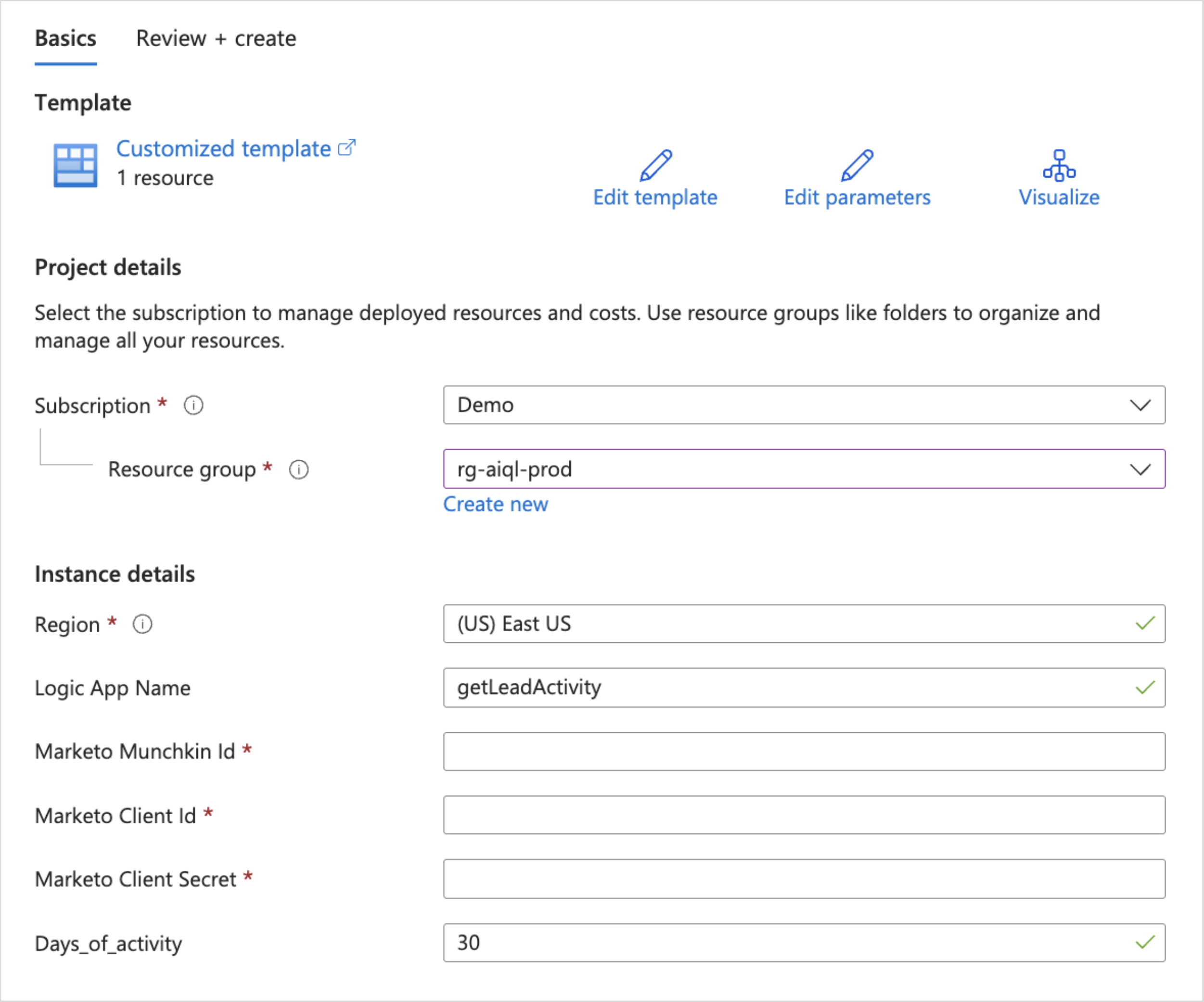
Body
Body

getLeadActivity

Deploy to Azure

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 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:

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.



Template

Customized template
1 resource

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Demo
Resource group *	rg-aiql-prod

Instance details

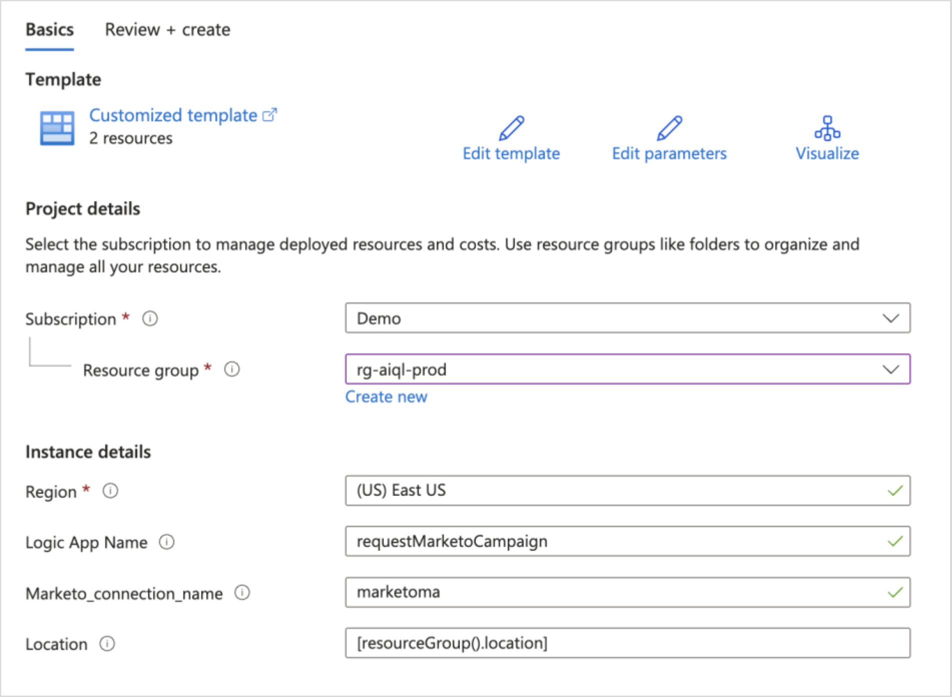
Region *	(US) East US
Logic App Name	getLeadActivity
Marketo Munchkin Id *	
Marketo Client Id *	
Marketo Client Secret *	
Days_of_activity	30

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

requestMarketoCampaign

Click this button to deploy to Azure

Deploy to Azure



Template

Customized template
2 resources

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

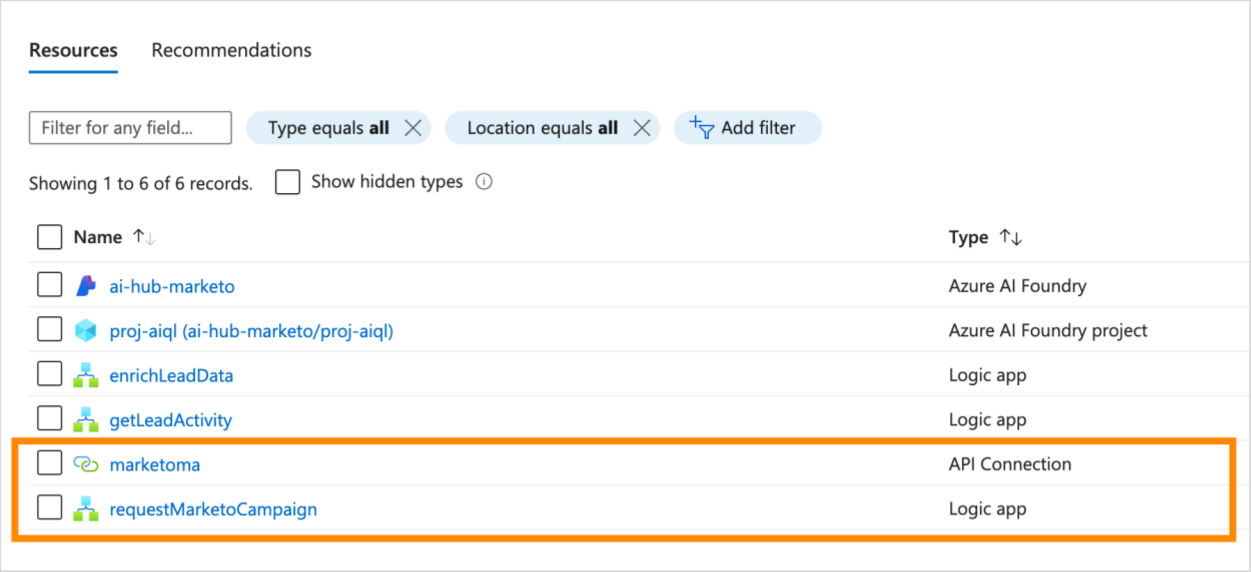
Subscription *	Demo
Resource group *	rg-aiql-prod

Create new

Instance details

Region *	(US) East US
Logic App Name	requestMarketoCampaign
Marketo_connection_name	marketoma
Location	[resourceGroup().location]

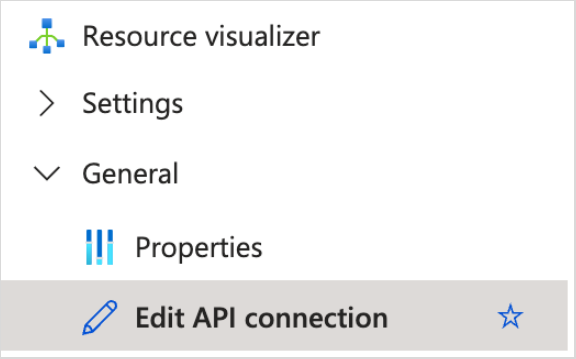
This template will create both the logic app and an API Connection: **marketoma**. First we need to set up the Marketo API connection. Click on it.



Showing 1 to 6 of 6 records. □ Show hidden types

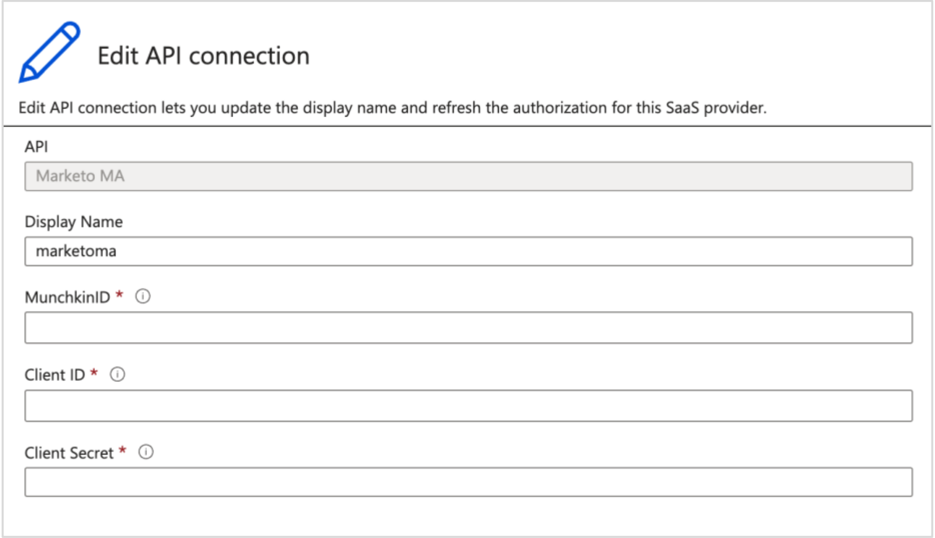
Name	Type
□ ai-hub-marketo	Azure AI Foundry
□ proj-aiql (ai-hub-marketo/proj-aiql)	Azure AI Foundry project
□ enrichLeadData	Logic app
□ getLeadActivity	Logic app
□ marketoma	API Connection
□ requestMarketoCampaign	Logic app

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.



Edit API connection

Edit API connection lets you update the display name and refresh the authorization for this SaaS provider.

API: Marketo MA

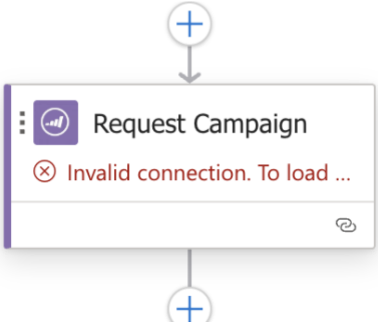
Display Name: marketoma

MunchkinID *

Client ID *

Client Secret *

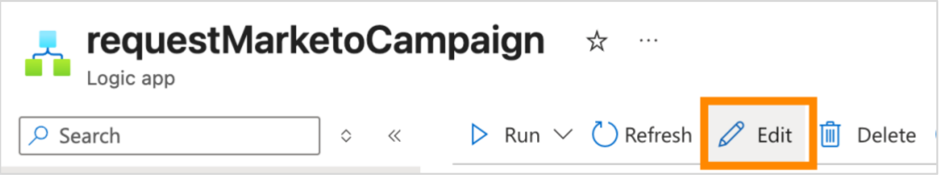
If you don't set this up, you will see an error in the Request Campaign step in the Logic App:



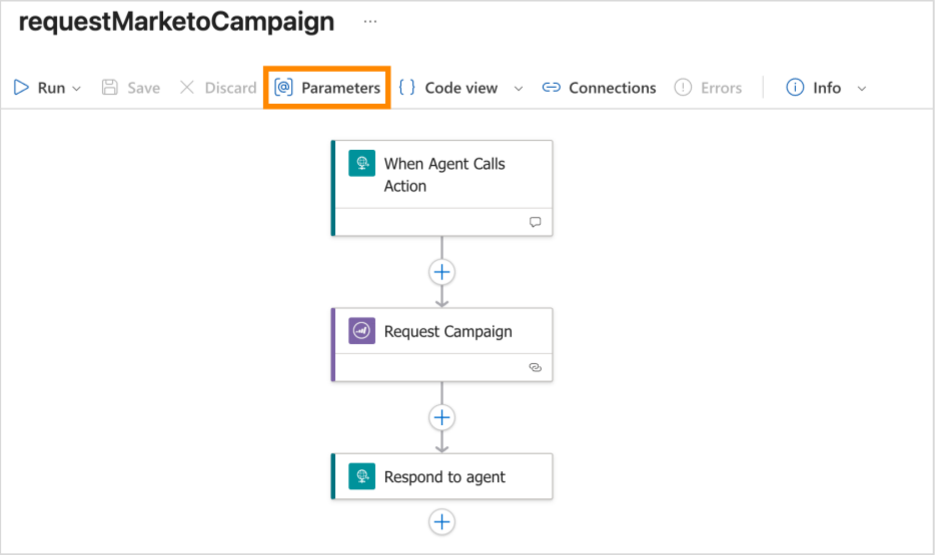
Request Campaign

Invalid connection. To load ...

Next go back to the logic app and Edit



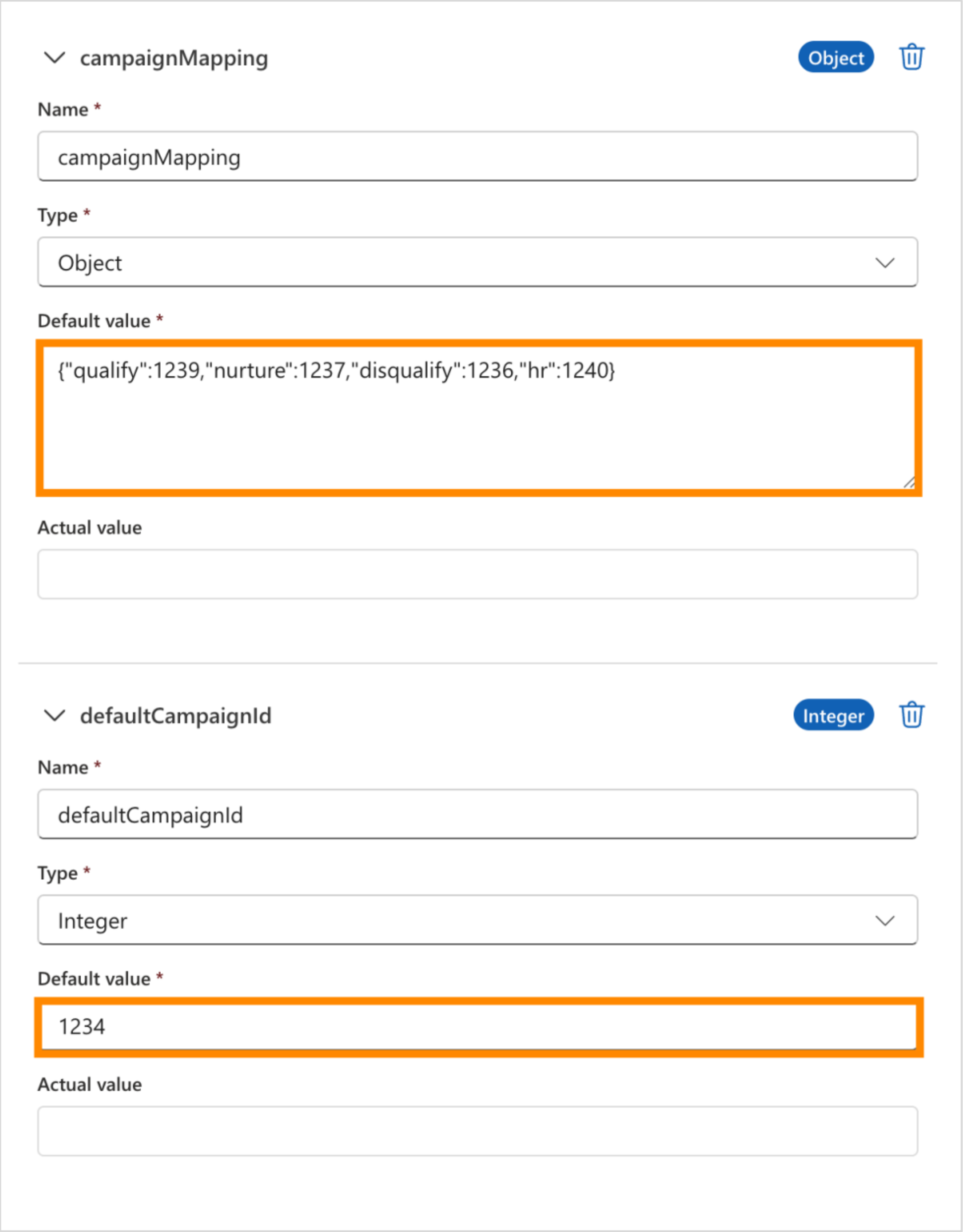
Edit the Parameters



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/SC1236A1ZN38
```



campaignMapping

Name *

campaignMapping

Type *

Object

Default value *

{"qualify":1239,"nurture":1237,"disqualify":1236,"hr":1240}

Actual value

defaultCampaignId

Name *

defaultCampaignId

Type *

Integer

Default value *

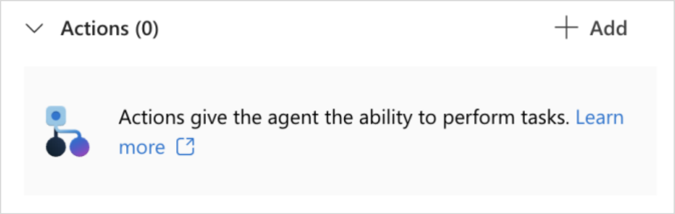
1234

Actual value

Register the Logic App as a Tool in the Agent

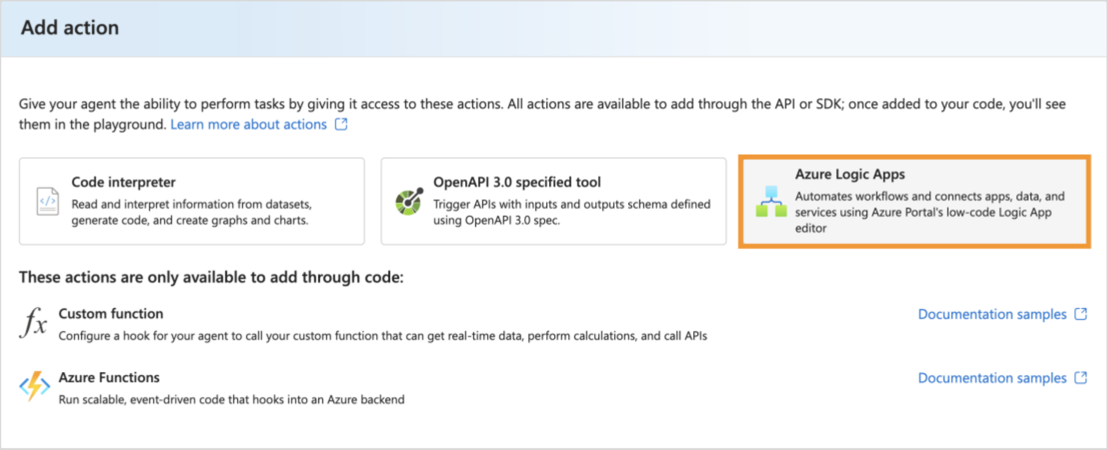
Now that we’ve created our actions we need to tell our AI Agent about them.

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



Actions give the agent the ability to perform tasks. Learn more

Select **Azure Logic Apps**



Give your agent the ability to perform tasks by giving it access to these actions. All actions are available to add through the API or SDK; once added to your code, you'll see them in the playground. Learn more about actions

Code interpreter
Read and interpret information from datasets, generate code, and create graphs and charts.

OpenAPI 3.0 specified tool
Trigger APIs with inputs and outputs schema defined using OpenAPI 3.0 spec.

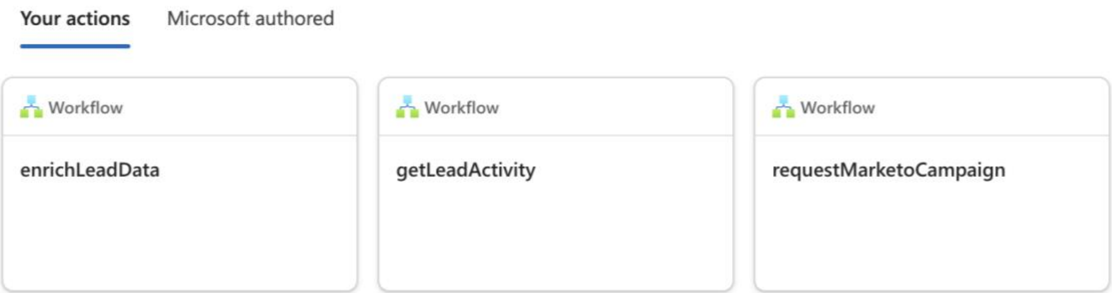
Azure Logic Apps
Automates workflows and connects apps, data, and services using Azure Portal's low-code Logic App editor

These actions are only available to add through code:

Custom function
Configure a hook for your agent to call your custom function that can get real-time data, perform calculations, and call APIs
Documentation samples

Azure Functions
Run scalable, event-driven code that hooks into an Azure backend
Documentation samples

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



enrichLeadData

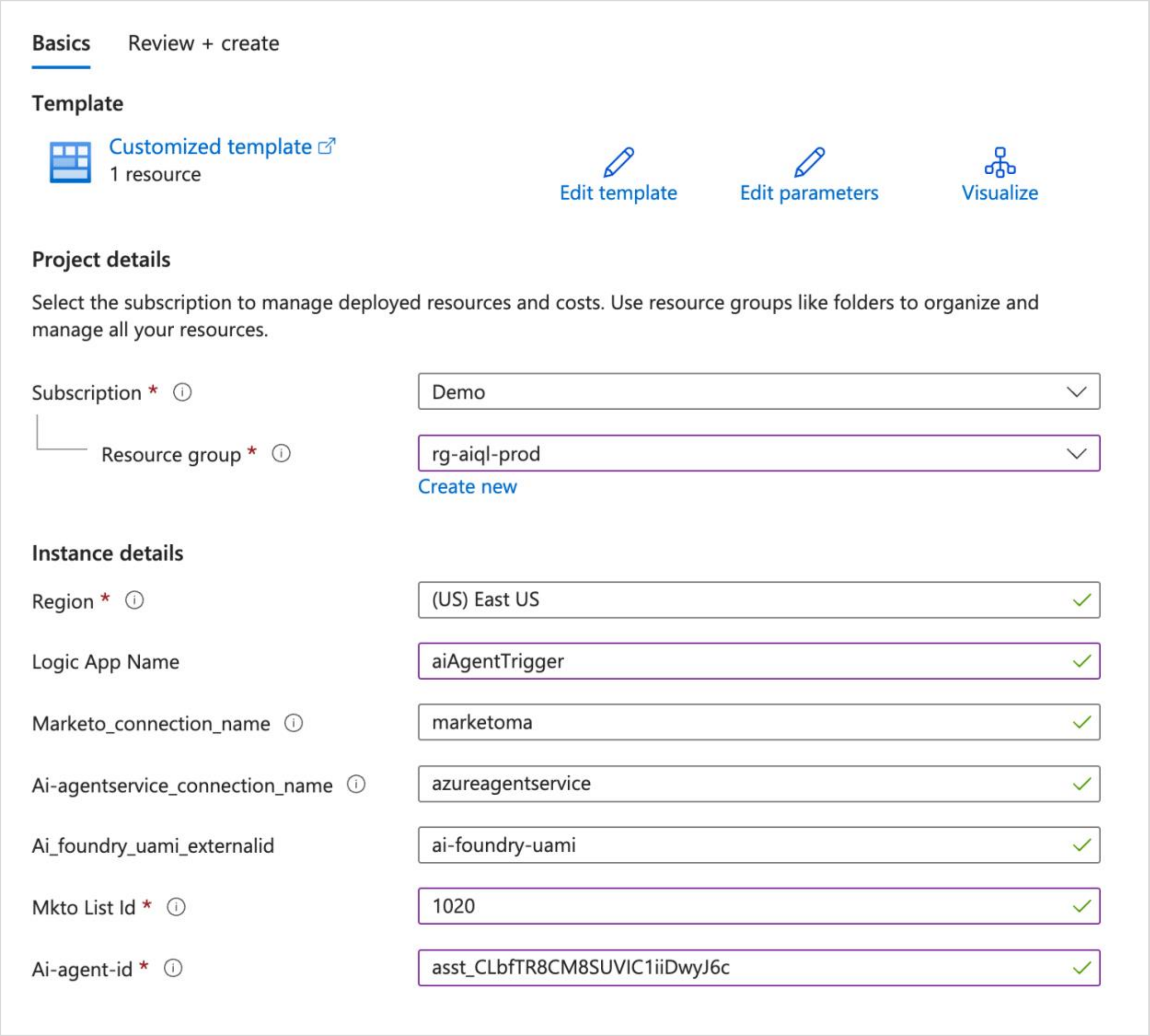
getLeadActivity

requestMarketoCampaign

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 list every 10 minutes and process the leads inside. Here you have a template you can deploy to Azure.

Deploy to Azure



Template

Customized template
1 resource

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

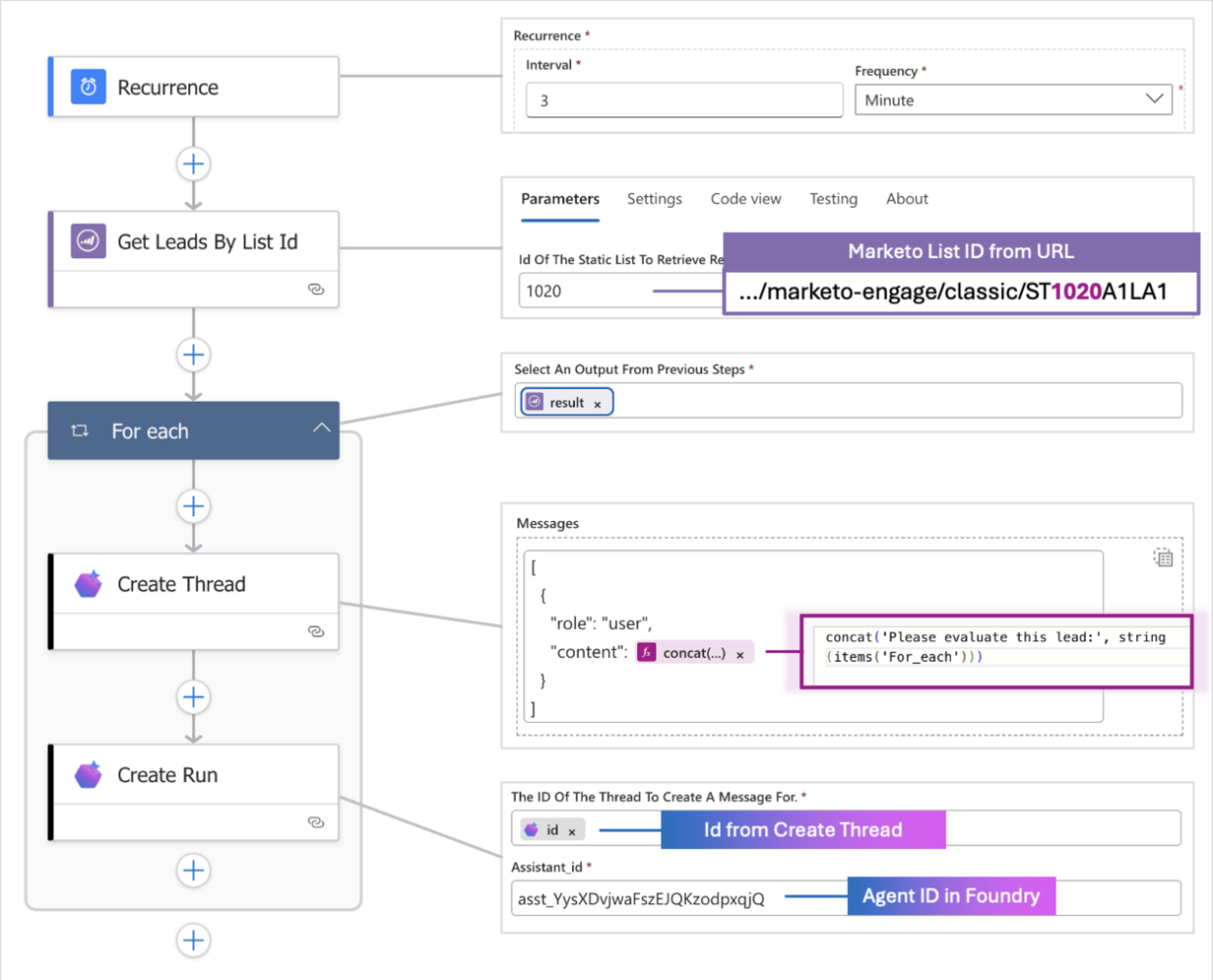
Subscription *	Demo
Resource group *	rg-aiql-prod Create new

Instance details

Region *	(US) East US ✓
Logic App Name	aiAgentTrigger ✓
Marketo_connection_name	marketoma ✓
Ai-agentservice_connection_name	azureagentservice ✓
Ai_foundry_uami_externalid	ai-foundry-uami ✓
Mkto List Id *	1020 ✓
Ai-agent-id *	asst_CLbfTR8CM8SUVIC1iiDwyJ6c ✓

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.



Recurrence *

Interval *: 3

Frequency *: Minute

Id Of The Static List To Retrieve Re: 1020

.../marketo-engage/classic/ST1020A1LA1

Select An Output From Previous Steps *: result

Messages

```
[
  {
    "role": "user",
    "content": concat(...)
  }
]
```

```
concat('Please evaluate this lead:', string
(items('For_each')))
```

The ID Of The Thread To Create A Message For. *: id

Assistant_id *: asst_YysXDvjwaFszEJQKzodpxqjQ

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 the Agent (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 **Marketo 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.

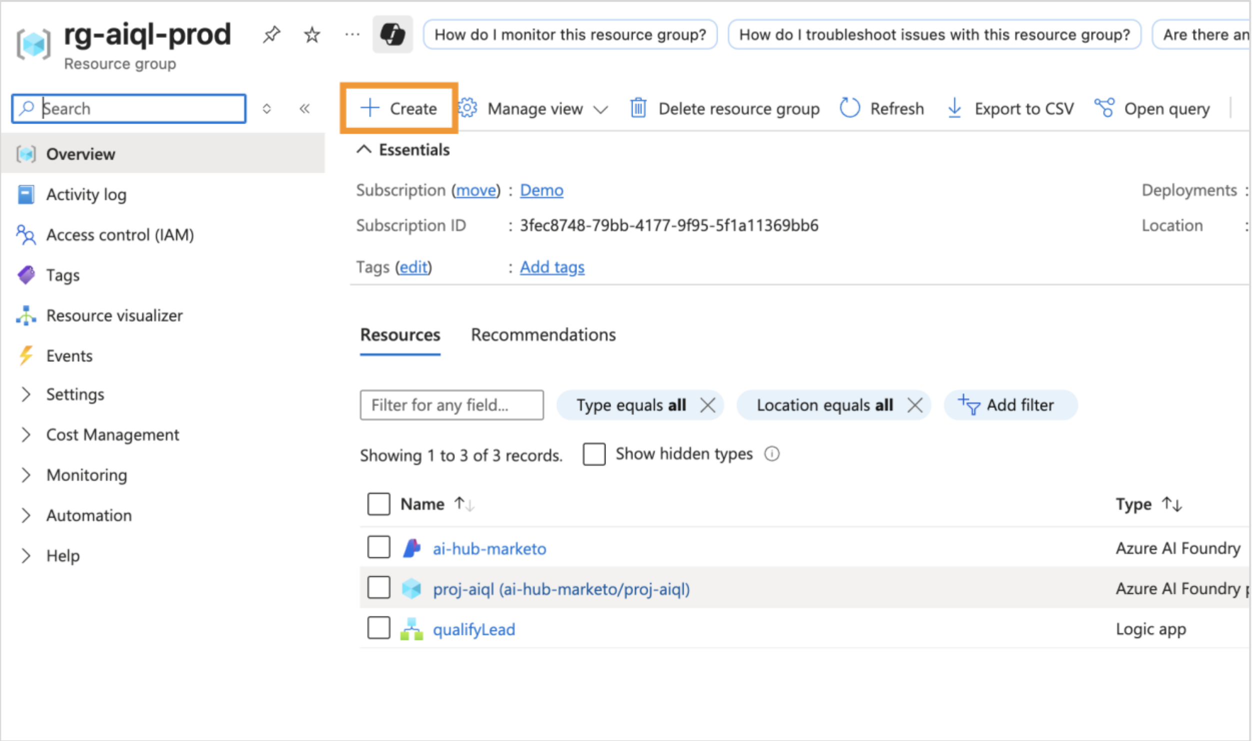
Creating Additional Logic Apps

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.



Subscription (move) : Demo

Subscription ID : 3fec8748-79bb-4177-9f95-5f1a11369bb6

Tags (edit) : Add tags

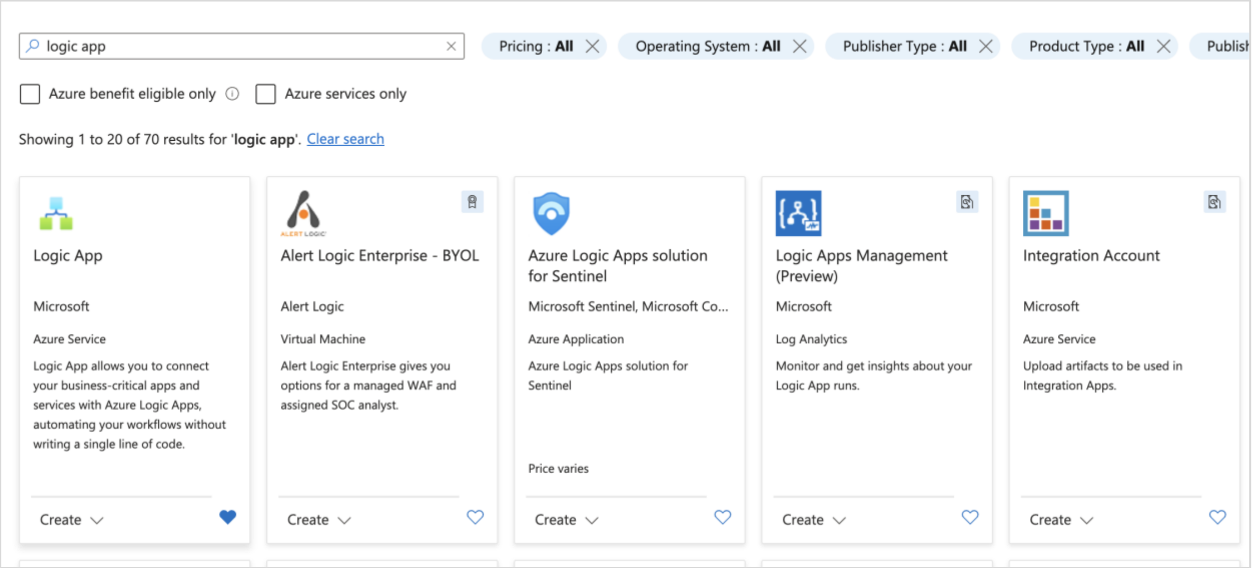
Deployments :

Location :

Showing 1 to 3 of 3 records.

□	Name	Type
□	ai-hub-marketo	Azure AI Foundry
□	proj-aiql (ai-hub-marketo/proj-aiql)	Azure AI Foundry p
□	qualifyLead	Logic app

Select Logic App > Create



Showing 1 to 20 of 70 results for 'logic app'. Clear search

Logic App
Microsoft
Azure Service
Logic App allows you to connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.

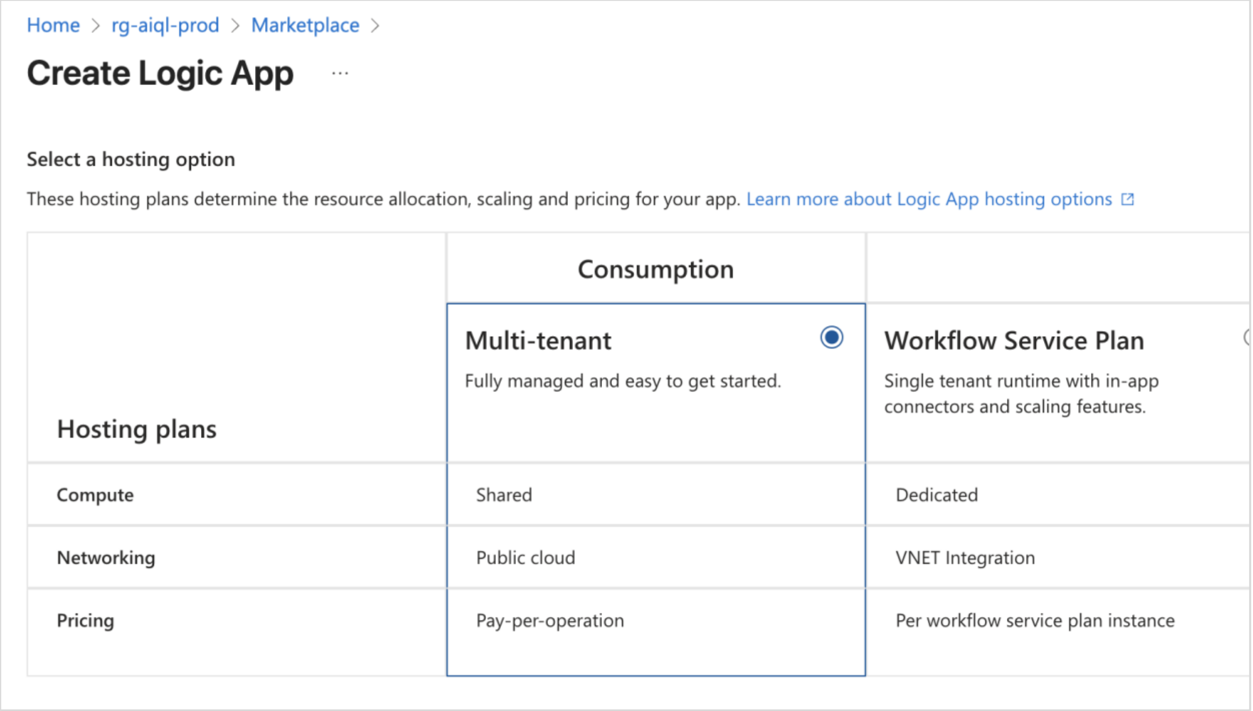
Alert Logic Enterprise - BYOL
Alert Logic
Virtual Machine
Alert Logic Enterprise gives you options for a managed WAF and assigned SOC analyst.

Azure Logic Apps solution for Sentinel
Microsoft Sentinel, Microsoft Co...
Azure Application
Azure Logic Apps solution for Sentinel
Price varies

Logic Apps Management (Preview)
Microsoft
Log Analytics
Monitor and get insights about your Logic App runs.

Integration Account
Microsoft
Azure Service
Upload artifacts to be used in Integration Apps.

Select **Consumption**



Home > rg-aiql-prod > Marketplace >

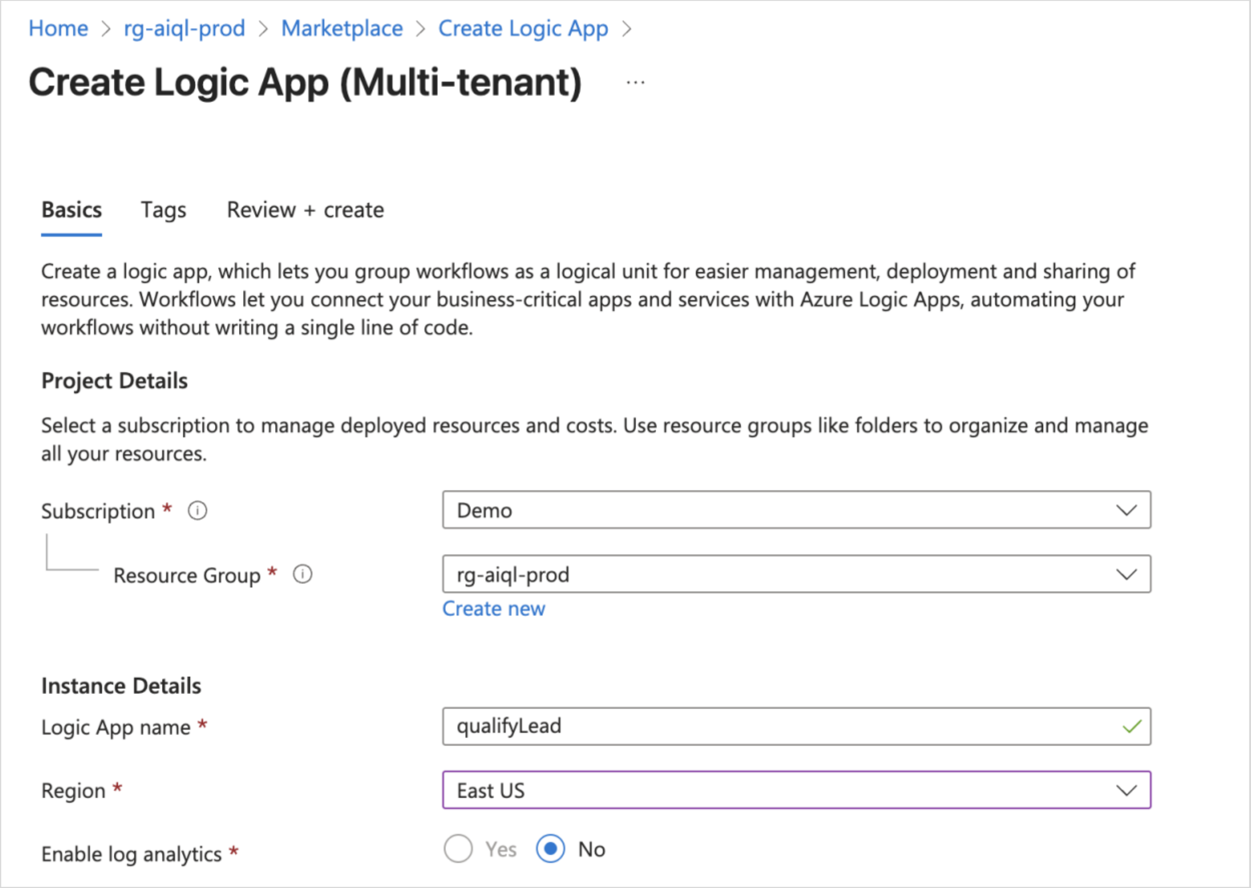
Create Logic App ...

Select a hosting option

These hosting plans determine the resource allocation, scaling and pricing for your app. Learn more about Logic App hosting options

	Consumption	
Hosting plans	**Multi-tenant** ● Fully managed and easy to get started.	**Workflow Service Plan** ○ Single tenant runtime with in-app connectors and scaling features.
Compute	Shared	Dedicated
Networking	Public cloud	VNET Integration
Pricing	Pay-per-operation	Per workflow service plan instance

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



Home > rg-aiql-prod > Marketplace > Create Logic App >

Create Logic App (Multi-tenant)

Basics | Tags | Review + create

Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Demo

Resource Group * : rg-aiql-prod

Create new

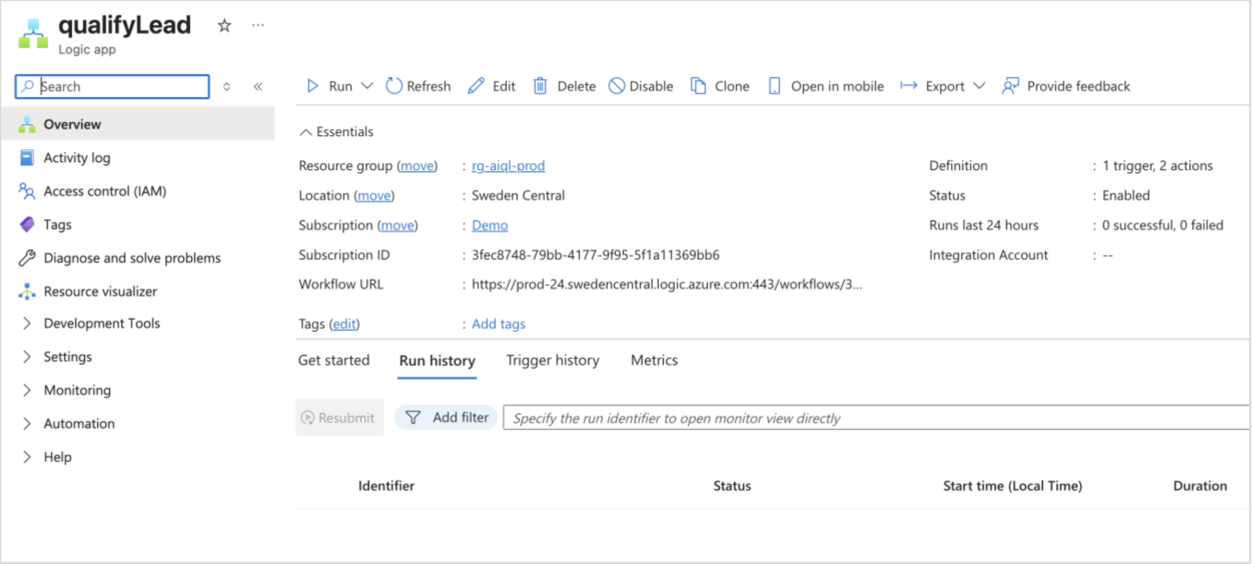
Instance Details

Logic App name * : qualifyLead ✓

Region * : East US

Enable log analytics * : ○ Yes ● No

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



qualifyLead ☆ ...

Logic app

Essentials

Resource group (move)	: rg-aiql-prod	Definition	: 1 trigger, 2 actions
Location (move)	: Sweden Central	Status	: Enabled
Subscription (move)	: Demo	Runs last 24 hours	: 0 successful, 0 failed
Subscription ID	: 3fec8748-79bb-4177-9f95-5f1a11369bb6	Integration Account	: --
Workflow URL	: https://prod-24.swedencentral.logic.azure.com:443/workflows/3...		
Tags (edit)	: Add tags		

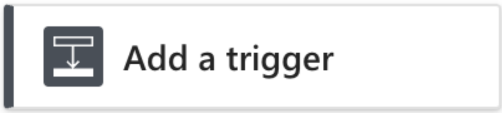
Specify the run identifier to open monitor view directly

Identifier	Status	Start time (Local Time)	Duration

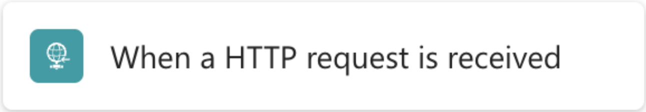
For requesting Marketo smart campaigns our logic app will need 3 parts:

Trigger:

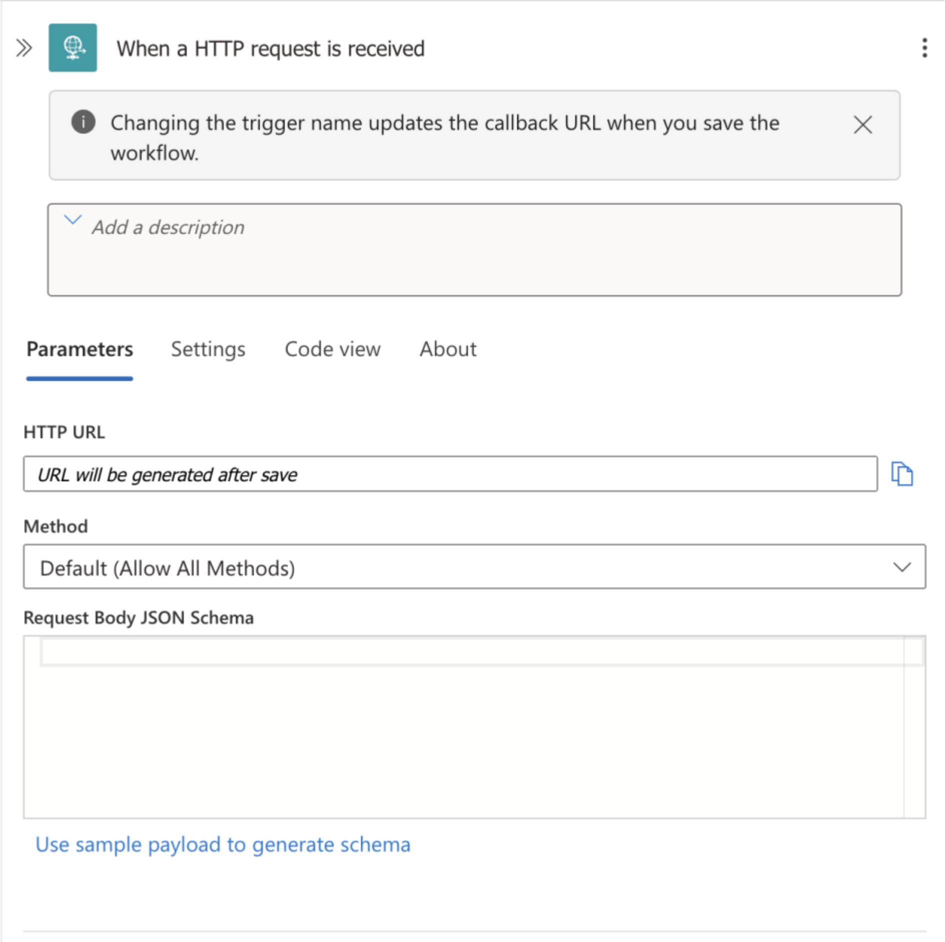
Click **Add a trigger**



Select **When a HTTP request is received**



Important – The HTTP request must have a description



When a HTTP request is received

Changing the trigger name updates the callback URL when you save the workflow.

HTTP URL

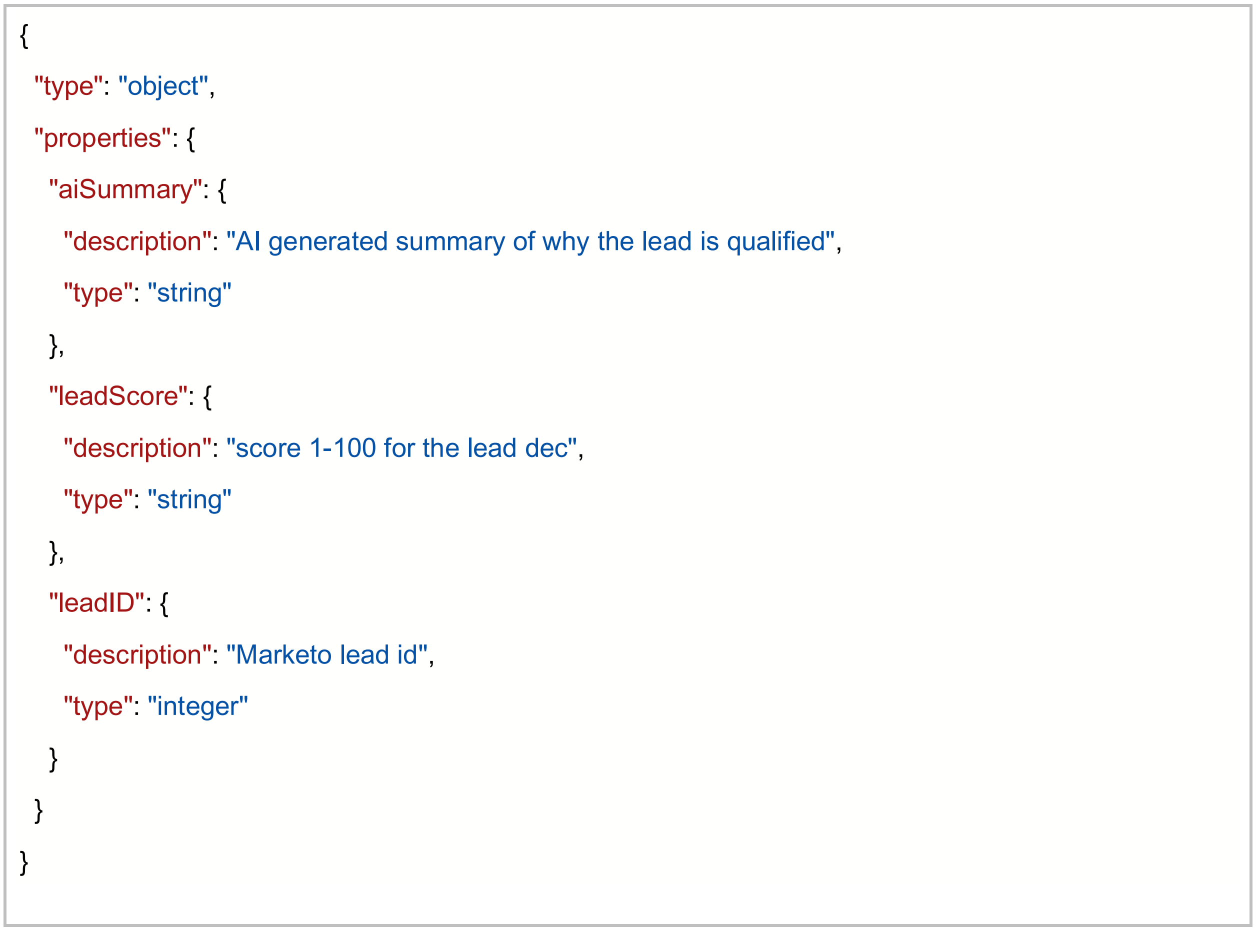
URL will be generated after save

Method

Default (Allow All Methods)

Request Body JSON Schema

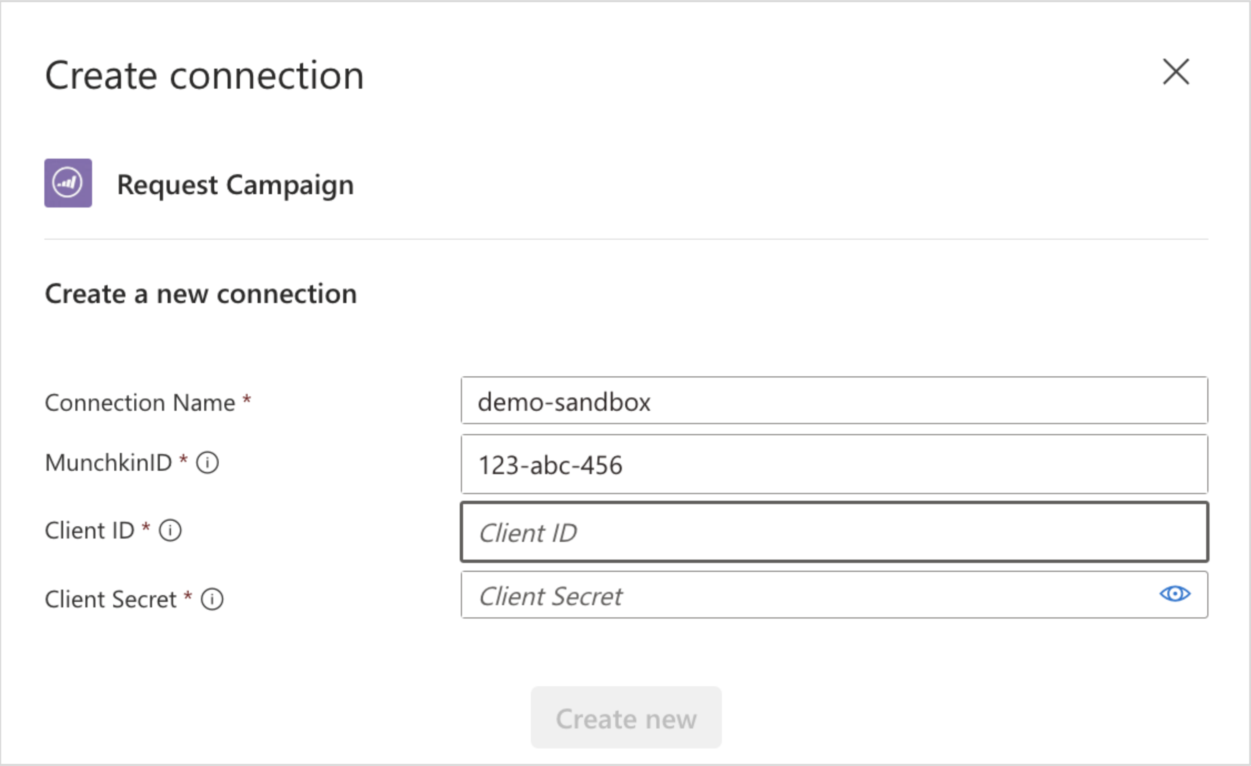
Under Request Body JSON Schema enter some version of the schema below. The only required parameter is leadID, which allows the agent to pass in the correct lead ID. The other parameters will be tokens that will be passed to Marketo.



```
{
  "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"
    }
  }
}
```

Next add an action and select Marketo 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.



Create connection

Request Campaign

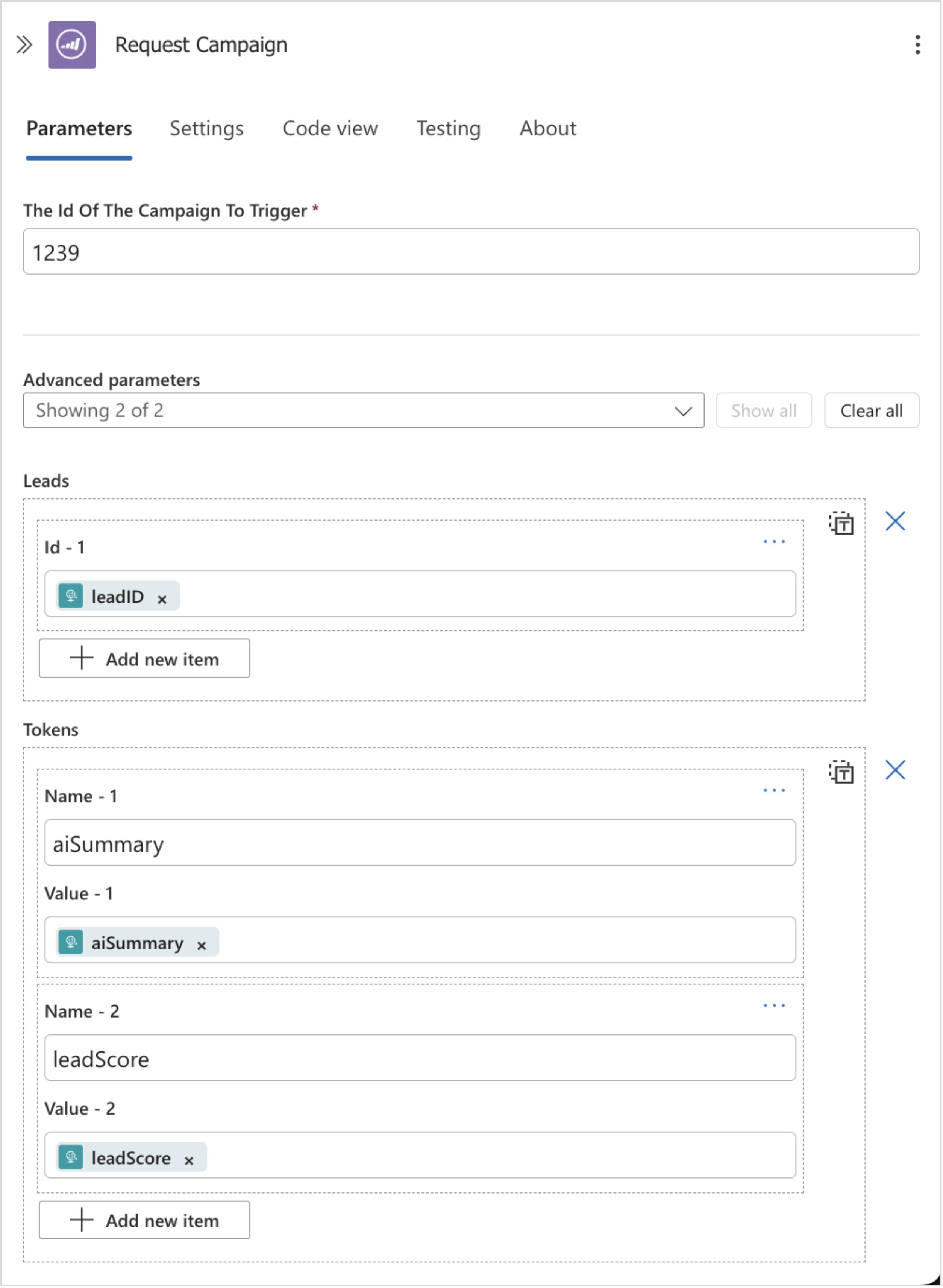
Create a new connection

Connection Name *: demo-sandbox

MunchkinID * ⓘ: 123-abc-456

Client ID * ⓘ: *Client ID*

Client Secret * ⓘ: *Client Secret*



Request Campaign

The Id Of The Campaign To Trigger *

1239

Advanced parameters

Showing 2 of 2

Leads

Id - 1

leadID

Tokens

Name - 1

aiSummary

Value - 1

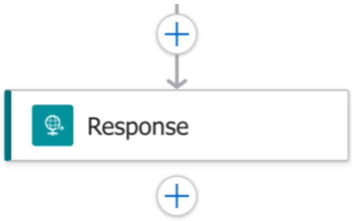
aiSummary

Name - 2

leadScore

Value - 2

leadScore



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. Graduate person-in-the-loop rules as confidence grows.

Quick Checklist

	Azure Account Created
	Marketo API Credentials Obtained
	Azure Resource Group Created
	Azure AI Studio Workspace & Agent Deployed
	Agent Instructions Defined
	ICP Document Uploaded to Vector Store
	Marketo Program & Tokens Created
	Marketo Watch List Created
	Marketo Smart Campaigns Created & Activated (Qualify, Disqualify, Nurture, HR)
	enrichLeadData Logic App Deployed & Configured
	getLeadActivity Logic App Deployed & Configured
	requestMarketoCampaign Logic App Deployed & Configured
	All Logic Apps Registered as Tools in Agent
	Trigger Logic App Deployed & Scheduled
	Manual End-to-End Test Completed Successfully
	Evaluation Run with Batch of Leads Completed