

warning "Backend update (October 2026)" GitHub Models inference was retired on July 30, 2026. Any references below are retained only for historical context while this lab is migrated. Use Lab 013 ? Free LLM backends to configure Foundry Local or Ollama before continuing.

Lab 009: Azure OpenAI Service Quickstart

Level: L100 Path: ? Microsoft Foundry Time: ~30 min ? Cost: Azure Free ? Azure free account, minimal usage

What You'll Learn

What Azure OpenAI Service is and how it differs from OpenAI and GitHub Models How to create an Azure OpenAI resource and deploy a model How to call Azure OpenAI via the Python SDK and REST API API key authentication vs. Microsoft Entra ID (managed identity) When to use Azure OpenAI vs. GitHub Models vs. Ollama

Introduction

Azure OpenAI Service provides access to OpenAI's models (GPT-4o, GPT-4o-mini, o1, and embeddings) through Azure infrastructure. This means:

Enterprise compliance: Data stays in your Azure tenant, governed by your policies Private

networking: Deploy behind a VNet, no public internet exposure Azure integrations: Works

natively with Azure AI Search, Azure AI Foundry, Key Vault, and more SLAs: Enterprise-grade uptime guarantees

info "Azure OpenAI vs. GitHub Models vs. OpenAI" | | GitHub Models | OpenAI (direct) |

Azure OpenAI | |--|-----|-----|-----| | Cost | Free tier |

Pay-per-token | Pay-per-token | | Auth | GitHub PAT | API key | API key or managed identity

| | Data residency | GitHub/Azure | OpenAI US | Your Azure region | | Enterprise features |

? | Limited | ? Full | | Rate limits | Low (dev only) | Standard | Configurable | | Best

for | Learning, dev | Consumer apps | Enterprise production |

For all L200+ labs in this hub, you can use GitHub Models (free). Azure OpenAI is the recommended path when moving to production.

Prerequisites Setup

Create an Azure Account

? Azure Free Account ? \$200 credit, 12 months free services

warning "Free tier limitations" Azure OpenAI in the free tier has very limited quota. For these exercises, a few API calls are all you need.

Request Azure OpenAI Access

Azure OpenAI requires an application: 1. Go to aka.ms/aoai/access 2. Fill out the form with your use case 3. Wait for approval (usually a few hours to 1 day)

tip "While waiting for approval" Complete this lab's conceptual sections and use GitHub Models for the hands-on exercises ? the code is identical.

Step 1: Create an Azure OpenAI Resource

1. In the Azure Portal, search for Azure OpenAI 2. Click + Create 3. Fill in: -

Subscription: your subscription - Resource group: create new ? rg-learning-hub - Region:

East US 2 (best model availability) - Name: aoai-learning-hub-[yourname] (must be globally unique) - Pricing tier: Standard S0 4. Click Review + Create ? Create

Step 2: Deploy a Model

1. Open your Azure OpenAI resource 2. Click "Go to Azure OpenAI Studio" ? oai.azure.com 3.

Click Deployments ? + Create new deployment 4. Configure: - Model: gpt-4o-mini - Deployment name: gpt-4o-mini (same as model name, easier to remember) - Tokens per minute: 10K (enough for labs) 5. Click Deploy

Also deploy an embedding model: Model: text-embedding-3-small Deployment name:

text-embedding-3-small

Step 3: Get Your Endpoint and Keys

From your Azure OpenAI resource: 1. Click Keys and Endpoint in the left menu 2. Copy KEY 1 and the Endpoint URL

Store them as environment variables:

=== "Windows (PowerShell)" [diagram or code sample]

=== "macOS / Linux" [diagram or code sample]

danger "Never commit API keys" Use environment variables, Azure Key Vault, or .env files (add .env to .gitignore). Never hardcode API keys in source code.

Step 4: Your First Chat Completion

[diagram or code sample]

[diagram or code sample]

tip "Same API, different client" The code is identical to OpenAI ? just use AzureOpenAI instead of OpenAI, add azure_endpoint and api_version. That's it.

Step 5: Embeddings with Azure OpenAI

[diagram or code sample]

Step 6: Switch Between Backends with One Variable

This pattern lets you write code once and run it against GitHub Models (free) in development and Azure OpenAI in production:

[diagram or code sample]

Usage: [diagram or code sample]

Step 7: Managed Identity (Production Best Practice)

In production, never use API keys. Use Managed Identity instead:

[diagram or code sample]

[diagram or code sample]

Assign your app the Cognitive Services OpenAI User role in Azure RBAC.

Azure OpenAI in Azure AI Foundry

Azure AI Foundry provides a unified portal for managing Azure OpenAI and other AI services:

1. Go to ai.azure.com 2. Create a Hub (once per organization) and a Project (once per application) 3. Deploy models through the Model Catalog 4. Use the Playground to test interactively 5. Access Evaluation and Monitoring tools

? See Lab 030 ? Foundry Agent Service + MCP for a full Foundry walkthrough.

? Knowledge Check

question "1. What is the key code difference between using OpenAI directly and Azure OpenAI?" Use AzureOpenAI instead of OpenAI, and provide azure_endpoint and api_version. The model parameter takes your deployment name, not the model name. Everything else (messages, temperature, etc.) is identical.

question "2. Why should you use Managed Identity instead of API keys in production?" API keys are static secrets that can be leaked, stolen, or accidentally committed to source control. Managed Identity uses Azure's built-in identity system ? no secret is ever stored in code or config files. Access is controlled through Azure RBAC, is automatically rotated, and leaves audit trails.

question "3. When would you choose GitHub Models over Azure OpenAI?" GitHub Models is ideal for development, learning, and prototyping ? it's free with just a GitHub account, no Azure subscription needed, no deployment to configure. Choose Azure OpenAI when you need: data residency requirements, private networking, enterprise SLAs, integration with Azure AI Search or Foundry, or production-grade reliability.

Summary

Concept | Key takeaway | AzureOpenAI client | Same API as OpenAI, just different constructor parameters | Deployment name | You deploy a model with a name ? use that name, not the model name | API version | Required parameter ? check the latest at aka.ms/oai/docs | Managed identity | Use in production ? no static secrets | Foundry | The portal that wraps Azure OpenAI with eval, monitoring, and agent tools |

Next Steps

Build a full agent with Foundry: ? Lab 030 ? Foundry Agent Service + MCP Add RAG to your Azure OpenAI app: ? Lab 031 ? pgvector Semantic Search on Azure Free alternative for labs: ? Lab 013 ? GitHub Models

Phase 3 modernization

warning "Verified 2026-10-27" Use the Microsoft Foundry name and GA Agent Service surfaces based on the Responses API. Evaluations are GA; hosted agents remain preview and must be labeled accordingly. Do not build new dependencies on portal Workflows scheduled for retirement on 2026-12-01.

[diagram or code sample]

Foundry Agent Service GA announcement Foundry Agent Service Foundry evaluation Foundry Workflows retirement

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

Download generated PDF Download generated ZIP

Supporting file: CHEATSHEET.md

Supporting files: Lab 009: Azure OpenAI Service Quickstart

Start with the documented local or free path. Record assumptions and validation evidence.

Keep hosted services optional unless the lab marks them required.

Supporting file: README.md

Lab 009: Azure OpenAI Service Quickstart

Use these files with the lab guide and generated kit.