🧠 Simple RAG Pipeline using Azure & .NET

This workshop walks through how to build a simple Retrieval-Augmented Generation (RAG) pipeline using:

* Azure Blob Storage
* Azure AI Search (Cognitive Search)
* Azure OpenAI (GPT-4)
* .NET 6 Console App or API

# 🔧 What You'll Build

* Upload documents to Azure Blob
* Index them using Azure Cognitive Search
* Retrieve relevant chunks via .NET
* Pass them to Azure OpenAI for a GPT-4 generated response

# 📦 Prerequisites & Requirements

## 🔧 Development Environment

* .NET 9 SDK
* Visual Studio Code or Visual Studio 2022
* PowerShell 7+
* Git

## ☁️ Azure Requirements

* Azure Subscription with permissions
* Azure CLI
* Azure OpenAI Service Access
* Azure Cognitive Search service availability
* Azure Blob Storage service availability

## 🔑 Required Azure Services & Permissions

* Azure OpenAI Service with GPT-4 or GPT-4o deployment
* Azure Cognitive Search (Basic tier or higher)
* Azure Blob Storage (Standard LRS or higher)
* Contributor access to Resource Groups

# 💻 Local Setup & Steps

1. Clone the repo and navigate to project folder
2. Verify .NET installation with `dotnet --version`
3. Run provisioning script from `azure-setup/`
4. Run `dotnet run` from `src/RagApi`

# 🔐 .env File Template

AZURE\_OPENAI\_API\_KEY=your-openai-api-key  
AZURE\_OPENAI\_ENDPOINT=https://**[your-region].**api.cognitive.microsoft.com/  
AZURE\_OPENAI\_DEPLOYMENT=gpt-4o  
AZURE\_OPENAI\_API\_VERSION=2025-01-01-preview  
  
AZURE\_SEARCH\_SERVICE\_NAME=your-search-service-name  
AZURE\_SEARCH\_API\_KEY=your-search-api-key  
AZURE\_SEARCH\_INDEX\_NAME=rag-index  
AZURE\_SEARCH\_API\_VERSION=2023-07-01-Preview  
  
AZURE\_STORAGE\_ACCOUNT\_NAME=your-storage-account-name  
AZURE\_STORAGE\_ACCOUNT\_KEY=your-storage-account-key  
AZURE\_STORAGE\_CONTAINER\_NAME=documents  
  
AZURE\_REGION=eastus  
AZURE\_TENANT\_ID=your-tenant-id  
AZURE\_SUBSCRIPTION\_ID=your-subscription-id

# 🚀 QuickStart Commands

Provision Azure Resources:

cd azure-setup  
./create-resources.ps1

Run the API:

cd src/RagApi  
dotnet restore  
dotnet build  
dotnet run

# ✅ Test the API

* Use Swagger UI at <http://localhost:5000/swagger>
* Use curl or the console app to send test queries

# 🧪 Troubleshooting

* Ensure .env file exists and has valid keys
* Verify Azure services are deployed correctly
* Check port availability or change with `--urls`

# 📋 Checklist

* .NET 9 installed
* Azure CLI installed and logged in
* Resources deployed or created manually
* .env file populated
* Index created and documents uploaded