# AI-102 Advanced Topics for Trainers

This repository contains comprehensive technical documents on advanced topics critical for trainers delivering the Microsoft Azure AI Engineer Associate (AI-102) course. These documents are designed to provide in-depth explanations, real-world use cases, key considerations, best practices, and sophisticated code examples in both C# and Python, based on innovative customer scenarios.

## Table of Contents

* [Overview](#overview)
* [Advanced Topics Covered](#advanced-topics-covered)
* [Installation and Setup](#installation-and-setup)
* [Usage](#usage)
* [Contributing](#contributing)
* [Code of Conduct](#code-of-conduct)
* [License](#license)

## Overview

The AI-102 course focuses on developing AI solutions with Azure AI services. To effectively deliver this course, trainers need to possess a deep understanding of not just the foundational concepts but also advanced implementation patterns and best practices. This documentation aims to bridge that gap by providing detailed insights into complex topics that are highly relevant in real-world AI development.

Each technical document delves into a specific advanced topic, offering:

* A clear explanation of the core concept.
* Relevant use cases and implementation scenarios, often inspired by innovative customer stories.
* Key considerations and best practices for successful deployment.
* Elaborate and sophisticated sample implementations in both C# and Python, demonstrating practical application.

This resource is intended to empower trainers to deliver the AI-102 material with authority, technical depth, and practical relevance, ensuring students gain a comprehensive understanding of building cutting-edge AI solutions on Azure.

## Advanced Topics Covered

This repository covers the following advanced topics:

1. [**Generative AI and AI Agents with Azure AI and Semantic Kernel:**](documents/Advanced_Topic_1_Generative_AI_and_AI_Agents_with_Azure_AI_and_Semantic_Kernel.md) Exploring the synergy between generative models and autonomous agents, orchestrated by Semantic Kernel.
2. [**Retrieval-Augmented Generation (RAG):**](documents/Advanced_Topic_2_Retrieval_Augmented_Generation.md) Enhancing LLM responses by grounding them in external, authoritative data sources.
3. [**Fine-Tuning and Using Custom Models:**](documents/Advanced_Topic_3_Fine_Tuning_and_Using_Custom_Models.md) Adapting pre-trained models to specific domains and tasks for improved performance.
4. [**Integrate MCP Tools with Azure AI Agents:**](documents/Advanced_Topic_4_Integrate_MCP_Tools_with_Azure_AI_Agents.md) Enabling AI agents to dynamically discover and utilize external functionalities through the Model Context Protocol.

Each topic has a dedicated Markdown document in the documents/ directory, which includes detailed explanations, use cases, best practices, and code examples. Testing instructions for each code example are also integrated directly into their respective documents.

## Installation and Setup

To set up your environment and run the code examples, please follow these general steps. Specific prerequisites and setup instructions for each example are detailed within their respective technical documents.

### Prerequisites

* **Azure Subscription:** An active Azure subscription is required to provision Azure AI services.
* **Azure OpenAI Service:** Access to Azure OpenAI Service with deployed models (e.g., gpt-3.5-turbo, gpt-4, or fine-tuned models) is necessary for most examples. Ensure you have the endpoint, API key, and deployment names.
* **Azure AI Search:** For RAG examples, an Azure AI Search service with a populated index is required. You will need the service endpoint, API key, and index name.
* **.NET SDK:** For C# examples, .NET SDK (version 6.0 or higher) is required.
* **Python:** For Python examples, Python 3.8 or higher is required.

### Environment Variables

Before running any code example, you must set the necessary environment variables with your Azure credentials. These variables are typically:

* AZURE\_OPENAI\_ENDPOINT
* AZURE\_OPENAI\_KEY
* AZURE\_OPENAI\_DEPLOYMENT\_NAME (for chat models)
* AZURE\_OPENAI\_FINE\_TUNED\_DEPLOYMENT\_NAME (for fine-tuned models)
* AZURE\_SEARCH\_SERVICE\_ENDPOINT
* AZURE\_SEARCH\_API\_KEY
* AZURE\_SEARCH\_INDEX\_NAME

**Example for Bash/Zsh (Linux/macOS):**

export AZURE\_OPENAI\_ENDPOINT="https://your-openai-resource.openai.azure.com/"  
export AZURE\_OPENAI\_KEY="your-openai-api-key"  
export AZURE\_OPENAI\_DEPLOYMENT\_NAME="your-chat-model-deployment-name"  
export AZURE\_SEARCH\_SERVICE\_ENDPOINT="https://your-search-service.search.windows.net"  
export AZURE\_SEARCH\_API\_KEY="your-search-api-key"  
export AZURE\_SEARCH\_INDEX\_NAME="your-search-index-name"

**Example for PowerShell (Windows):**

$env:AZURE\_OPENAI\_ENDPOINT="https://your-openai-resource.openai.azure.com/"  
$env:AZURE\_OPENAI\_KEY="your-openai-api-key"  
$env:AZURE\_OPENAI\_DEPLOYMENT\_NAME="your-chat-model-deployment-name"  
$env:AZURE\_SEARCH\_SERVICE\_ENDPOINT="https://your-search-service.search.windows.net"  
$env:AZURE\_SEARCH\_API\_KEY="your-search-api-key"  
$env:AZURE\_SEARCH\_INDEX\_NAME="your-search-index-name"

### Project Structure

The technical documents are located in the documents/ directory. Each document (.md file) contains the explanation, use cases, best practices, and the C# and Python code examples with their respective testing instructions.

.github/  
documents/  
├── Advanced\_Topic\_1\_Generative\_AI\_and\_AI\_Agents\_with\_Azure\_AI\_and\_Semantic\_Kernel.md  
├── Advanced\_Topic\_2\_Retrieval\_Augmented\_Generation.md  
├── Advanced\_Topic\_3\_Fine\_Tuning\_and\_Using\_Custom\_Models.md  
└── Advanced\_Topic\_4\_Integrate\_MCP\_Tools\_with\_Azure\_AI\_Agents.md  
README.md

## Usage

To utilize this resource, navigate to the documents/ directory. Each Markdown file corresponds to an advanced topic. Open the relevant .md file to access the comprehensive technical document.

Within each document, you will find:

* **Concept Explanation:** A detailed breakdown of the advanced AI concept.
* **Use Cases and Scenarios:** Real-world applications and how the concept can be implemented.
* **Key Considerations and Best Practices:** Important factors and recommendations for effective implementation.
* **Sample Implementations (C# and Python):** Fully functional code examples demonstrating the topic in practice.
* **How to Test:** Step-by-step instructions for setting up, configuring, running, and verifying the C# and Python code examples, along with expected results.

To run the code examples:

1. **Review the Prerequisites:** Ensure all necessary software (e.g., .NET SDK, Python) and Azure resources are provisioned.
2. **Set Environment Variables:** Configure your environment variables with your Azure credentials as described in the [Installation and Setup](#installation-and-setup) section.
3. **Follow Document-Specific Instructions:** Each code example within its respective technical document includes precise instructions on how to set up the project, install dependencies, and execute the code. Pay close attention to these details.

We encourage you to experiment with the code, modify it, and adapt it to your own learning or teaching scenarios. The examples are designed to be a starting point for deeper exploration.