Microsoft

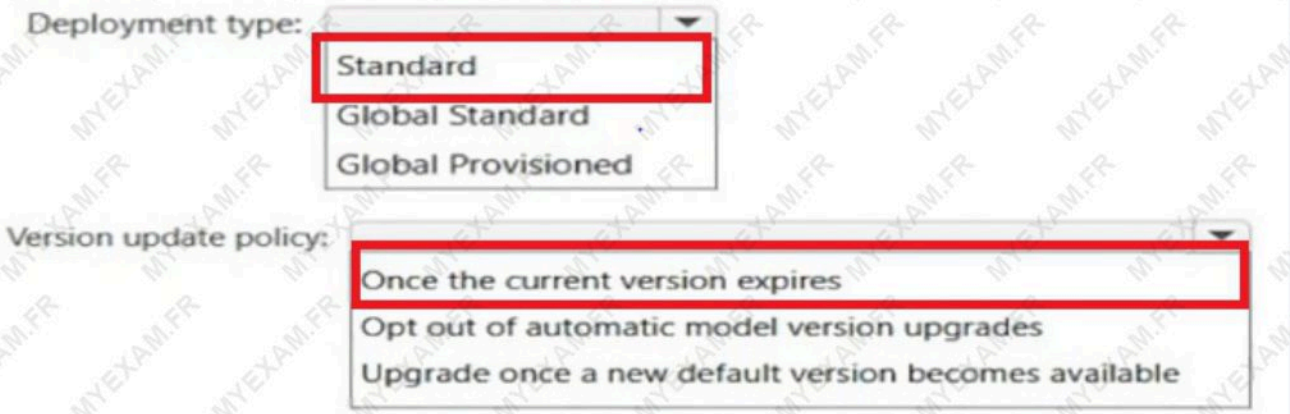
(AI-103)

Developing AI Apps and Agents on Azure

Total: **117 Questions**

Link:

Answer Area



Explanation:

Deployment Type: Selected as **Standard** because the workload requires dynamic scaling for variable traffic without reserved throughput capacity (eliminating Provisioned options), while adhering to strict regional data residency guidelines (e.g., data remaining within a specific zone or region like the EU).

Version Update Policy: Selected as **Once the current version expires** because the model version must remain consistent to guarantee stable, predictable responses over time, rather than automatically upgrading as soon as a new default version becomes available.

Why the other answer are incorrect:

Global Standard: Incorrect because "Global" routing sends traffic dynamically to any region worldwide with available capacity. This violates strict **data residency/compliance** rules if your data must remain within a specific geographic boundary (like the EU or US).

Global Provisioned: Incorrect because "Provisioned" requires buying reserved Throughput Units (PTUs). This incurs a high, fixed cost regardless of usage, making it wrong for workloads with **variable, low-volume traffic** where minimizing costs is a priority.

Opt out of automatic model version upgrades: Incorrect because this policy is deprecated or not recommended for long-term consistency. When a model version reaches its official retirement date, it will force-upgrade anyway, meaning you cannot permanently opt out of upgrades.

Upgrade once a new default version becomes available: Incorrect because this causes the model to update automatically as soon as Microsoft releases a new default. This can **break application code** or change prompt behaviors unexpectedly, violating requirements for strict consistency.

Question: 2

Case Study -
This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.
To answer the case study questions, you will bed to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.
A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to

Why Other Options are Less Suitable:

Option A: Enable role-based access control (RBAC) for the Azure AI Search resource
While RBAC is crucial for controlling access, it does not directly address the central management of credentials for accessing the Azure AI Search resource. RBAC defines what actions can be performed but does not consolidate credential management across multiple agents.
Suitability for Central Credential Management: Low
Option B: Disable key-based access control on the Azure AI Search resource
Disabling key-based access would likely increase security risks by potentially forcing the use of less secure methods or complicating access management. This does not contribute to centralizing credential management.
Suitability for Central Credential Management: Very Low (Counterproductive)
Option D: Create a managed private endpoint that connects to the Azure AI Search resource
While a managed private endpoint enhances security by providing a private, secure connection to the Azure AI Search resource, it focuses on network security rather than the central management of credentials for access by multiple agents.
Suitability for Central Credential Management: Low

Recommendation Summary: Given the need for central management of Azure AI Search credentials across multiple agents in Project1, **Option C** is the most direct and effective solution, facilitating secure, centralized credential management.

References:

For further understanding of managing connections and security in Azure and Microsoft Foundry, refer to:

1. **Microsoft Documentation - Manage connections in Microsoft Foundry**: https://learn.microsoft.com/en-us/microsoft-365/apps/foundry/manage-connections?view=o365-worldwide
2. **Azure Documentation - Secure your Azure AI Search with Azure RBAC**: https://learn.microsoft.com/en-us/azure/search/security-azure-rbac

Question: 4

HOTSPOT -
Your company is piloting a customer support agent in a Microsoft Foundry project name Project1. Project1 is connected to an existing Application Insights resource, and the company's support team reviews runs in the Traces tab.
The Foundry Agent Service is configured to perform the following actions:
Retrieve the Application Insights connection string by calling project_client.telemetry.get_application_insights_connection_string().
Call configure_azure_monitor(connection_string=...) to enable telemetry.
A separate LangChain service is configured to use OpenTelemetry and has the following configurations:
Uses AzureAIOpenTelemetryTracer(connection_string=..., enable_content_recording=False)
Passes the tracer by using config= "callbacks":[azure_tracer]
Company policy has the following requirements:
Telemetry from LangChain and OpenTelemetry must be distinguishable within the same Application Insights resource.
Secrets and credentials must NOT be stored in prompts, tool arguments, or span attributes.
For each of the following statements, select Yes if the statement is true. Otherwise, select No.
NOTE: Each correct selection is worth one point.

Authenticates by using a Microsoft Entra managed identity
Sends prompts to a deployed model by using the Azure OpenAI Responses API
How should you complete the Python code? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
from azure.identity import DefaultAzureCredential
from azure.ai.projects import AIProjectClient
credential = [ ▼ ]()

project_client = AIProjectClient(
    endpoint="https://contosoai.services.ai.azure.com/api/projects/project1",
    credential=credential,
)
with project_client.get_openai_client() as openai_client:
    response = openai_client.responses.[ ▼ ](
        model="trail-guide-chat",
        input="Create a 3-day hiking itinerary near Seattle.",
    )
    print(response.output_text)
```

AzureKeyCredential
ClientSecretCredential
DefaultAzureCredential

compact
create
retrieve

Answer:

Storage access:

When Azure AI Content Safety retrieves screenshots or file attachments via direct blob URLs, it requires specific data-plane permissions on the underlying storage repository.

System-assigned managed identity eliminates password management overhead.

Storage Blob Data Reader satisfies the principle of **least privilege**, providing full read access to download and review image binaries without granting unnecessary write or delete capabilities (which would be included in the Contributor role).

Question: 9

You have a Microsoft Foundry project that contains three agents as shown in the following table.

Name	Description
TriageAgent	Classifies incoming customer requests
PolicyAgent	Answers policy questions by searching internal content
ActionAgent	Creates or updates tickets by calling an HTTP API

You need to orchestrate the agents to ensure that the customer requests meet the following requirements:
Support a deterministic, step-based process that uses conditional branching and shared state across the agents.
Optionally trigger a ticket action based on the triage result.
The solution must minimize development effort.
What should you include in the solution?

A. a workflow
B. threads and runs without a workflow
C. a multi-agent group chat session
D. separate agent runs coordinated in the application code

Answer: A

Explanation:

A. a workflow.

Why a workflow is correct: In Microsoft Azure AI Foundry, agentic workflows are purposely built to orchestrate multiple agents using declarative, predefined sequences. They natively support deterministic step-by-step logic, if/else conditional branching, and automatic variable/state sharing across agents without requiring developers to write complex application orchestration or synchronization code, minimizing overall development effort.

Why other options are incorrect:

Group chat sessions (C) are designed for autonomous, fluid multi-agent conversations where agents dynamically choose when to chime in or pass control. They do not natively follow strict, deterministic, step-based workflows.

Separate application code loops or standalone threads (B & D) require significant manual engineering effort to establish shared states, handle condition mappings, and write error-handling glue logic.

Answer Area

Explanation:

ask_question: In declarative agent workflows, ask_question is used to explicitly pause execution and collect external feedback or manual intervention (such as human authorization/approval) before proceeding, unlike basic_chat or data_transformation which run automatically.

approval == "approved" To ensure that the execute_refund step runs conditionally only if approval has been explicitly granted, the expression evaluates the string output from the preceding id: approval step against the expected value "approved".

Question: 12

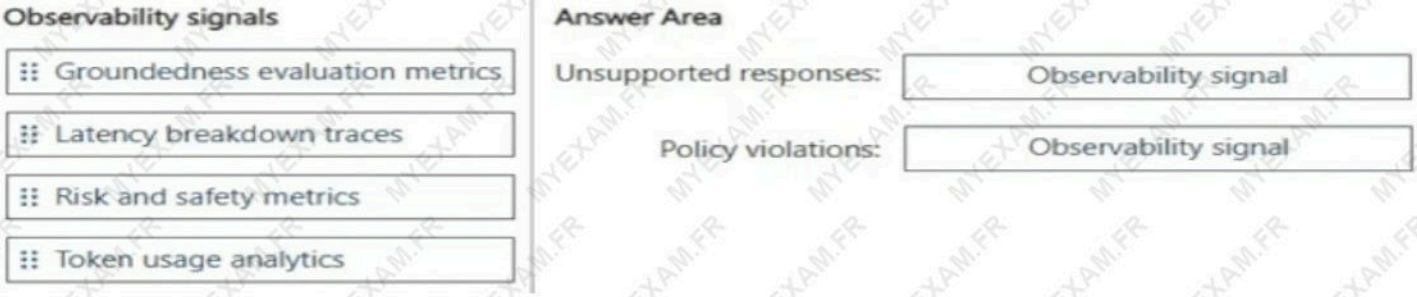
You have an Azure Speech in Foundry Tools resource that hosts a custom speech to text model deployed to a custom endpoint. An agent uses the endpoint to perform real-time speech recognition.
You are approaching the expiration date of the custom speech to text model.
What is the expected behavior when the model expires?

A. Speech recognition requests will return a 4xx error until a new custom model is deployed.

B. Speech recognition requests will continue to use the expired custom model until the model is removed manually.

C. Speech recognition requests will fall back to the most recent base model for the same locale.

Which observability signals should you use for each issue? To answer, drag the appropriate observability signals to the correct issues. Each observability signal may be used once, more than once, or not at all. You may need to drag the spit bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.



Answer:

Explanation:

Unsupported responses

Groundedness evaluation metrics

Policy violations:

Risk and safety metrics

Unsupported responses: When an AI model generates responses containing facts or assertions not present in the reference context, it results in an ungrounded or unsupported response (hallucination). Tracking **Groundedness evaluation metrics** assesses the exact ratio of sentences validated by the source text.

Policy violations: These occur when inputs or outputs conflict with regulatory or internal content guidelines (e.g., hate speech, violence, or sexual content). **Risk and safety metrics** directly monitor hits against predefined Azure AI Content Safety filter thresholds.

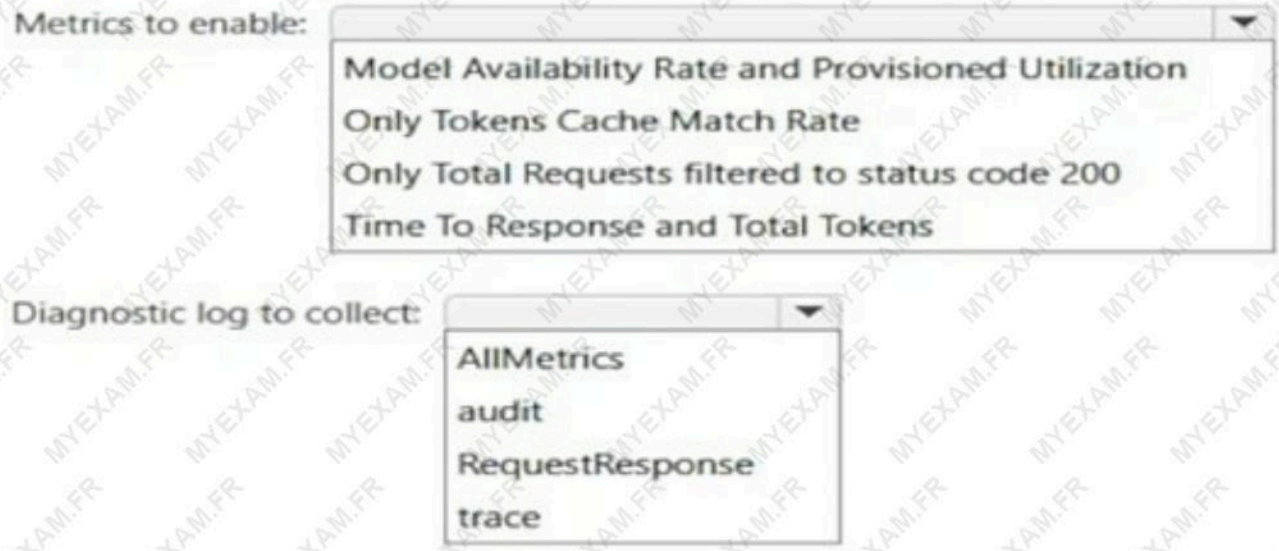
Question: 16

You have a Microsoft Foundry project named Project1 that contains an agent. The agent uses an OpenAPI 3.0 specification to call an external weather service.
The weather service requires a key to be passed in an HTTP header. The key value is stored as a connection in Project1.
You need to ensure that the key value from the connection is included automatically whenever the OpenAPI tool is invoked.
What should you configure in the OpenAPI specification?

A. a header parameter defined for each operation
B. an Azure Key Vault connection
C. an API key security scheme
D. a Bearer token security scheme

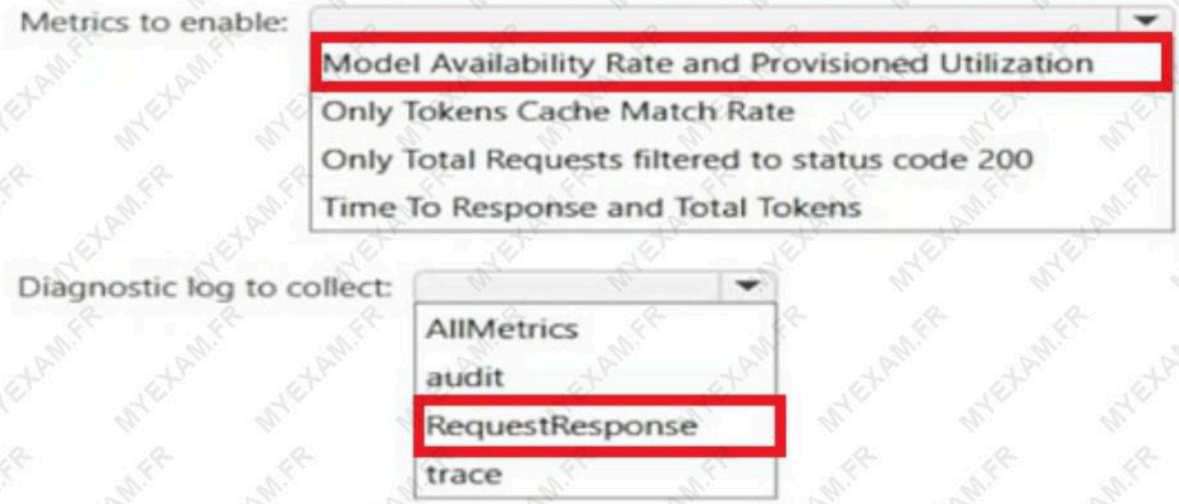
You have a Microsoft Foundry project that contains an internal Q&A agent.
Users report the following issues when they ask the agent questions:
An increase in the following response: “No relevant information found”
Periodic HTTP 429 rate limit exceeded errors during peak hours
You need to identify whether each issue is caused by model unavailability, resource limits, or inference failures.
What should you do? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area



Answer:

Answer Area



Explanation:

Metrics to enable :

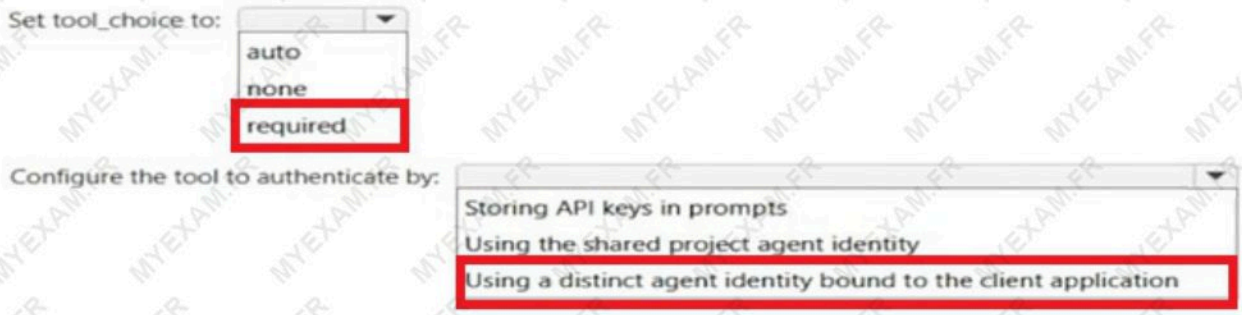
Model Availability Rate and Provisioned Utilization

Model Availability Rate isolates whether service errors or operation drop-offs are caused by downstream, server-side model unavailability.

Provisioned Utilization provides direct observability into capacity and resource limits. When it approaches or exceeds 100%, requests will automatically experience rate-limiting triggers and throw standard HTTP 429 Too Many Requests errors.

Diagnostic log to collect

Answer Area



Explanation:

Set tool_choice to

required

Configure the tool to authenticate by:

Using a distinct agent identity bound to the client application

Set tool_choice to (required): Setting this to **required** ensures that the AI model is deterministically forced to call one or more tools before returning a final answer, rather than autonomously deciding whether to skip the tool (as it would with auto).

Configure the tool to authenticate by (Using a distinct agent identity bound to the client application): To meet compliance standards stating that tool access must utilize an identity **isolated from other project resources** and support full audit tracing, assigning a distinct agent identity bound explicitly to the client application ensures complete isolation and individual request tracking.

Question: 21

You have a Microsoft Foundry project named Project1 that contains the following:
An OpenAPI tool that calls an external API
A project connection named Connection1 that stores the API key of the external API
When an agent calls the OpenAPI tool, the API returns a 401 unauthorized error, and traces show that the API key header is NOT being sent.
You need to ensure that the OpenAPI tool automatically includes the API key from Connection1 on all requests.
What should you do?

A. Enable identity passthrough so that the tool uses the Microsoft Entra token of the caller.
B. Add the API key header manually to the OpenAPI specification.
C. Configure the tool to use the default connection of Project1.
D. Connect the tool to Connection1.

Answer: D

Explanation:

Technical Justification for Correct Answer D

Question: 24

Note: This section contains one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem. You must determine whether the solution meets the stated goals. More than one solution in the set might solve the problem. It is also possible that none of the solutions in the set solve the problem.
After you answer a question in this section, you will NOT be able to return. As a result, these questions do not appear on the Review Screen.
You have a multimodal AI generative model that accepts image uploads and uses extracted image text to generate responses.
You discover that users can upload unsafe images and embed hidden instructions into images to manipulate the model.
You need to implement controls to mitigate the risk.
Solution: You configure image moderation to block unsafe content before processing the images.
Does this meet the goal?

A. Yes
B. No

Answer: B

Explanation:

Technical Justification for Solution Evaluation

Scenario Recap: Mitigate risks in a multimodal AI generative model by preventing unsafe image uploads that embed hidden instructions to manipulate the model.

Proposed Solution: Configure image moderation to block unsafe content before processing the images.

Evaluation of the Solution:

Why the Correct Answer is B (No):
Insufficiency in Addressing Hidden Instructions: Image moderation primarily focuses on identifying and blocking explicit, inappropriate, or unsafe visible content (e.g., nudity, violence). However, it may not effectively detect hidden instructions embedded in images (e.g., steganography, subtle text overlays not immediately recognizable as harmful).
Scope of "Unsafe Content": The solution assumes image moderation's definition of "unsafe" aligns with the threat of embedded instructions, which might not be the case. Embedded instructions could be crafted to avoid moderation flags.
Why Other Options are Not Provided but an Explanation for A (Yes) if it were Considered:
Hypothetical Defense for A (Yes): If the image moderation service used had advanced capabilities explicitly including detection of subtle, embedded textual instructions within images (a less common feature), then A could be justified. However, this is not a standard assumption for most image moderation tools.

Conclusion:Given the standard capabilities of image moderation services, the proposed solution does **not** fully meet the goal of mitigating the risk of embedded hidden instructions in images. Additional or alternative measures (e.g., more sophisticated image analysis for steganography, pre-processing to normalize images before text extraction) would be necessary to address the specified threat comprehensively.

References

Microsoft Azure Content Moderator Documentation: Highlights the capabilities and limitations of standard image moderation services.
Steganography Detection Techniques: Provides insight into the complexities of detecting hidden data in images, underscoring why standard moderation might fail.

More Suitable Approaches (Not Listed but for Context)

Image Preprocessing and Sanitization: Techniques to scrub or detect hidden data within images before processing.
Model Input Validation and Parser Security: Enhancing the security of the text extraction process to identify and reject manipulated inputs.
Adversarial Training or Robustness Enhancements: Modifying the model to be more resilient against manipulated inputs.

References

For further reading on securing AI models and image processing security:

Microsoft Azure - Secure Your AI Models: https://azure.microsoft.com/en-us/services/cognitive-services/security/
OWASP - Secure Image Uploading: https://owasp.org/www-project-security-cheatsheet/cheatsheet/Validation_Cheat_Sheet#Secure_Image_Uploading

Question: 27

Case Study -
This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.
To answer the case study questions, you will bed to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.
A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.
To start the case study -
To display the first question in this case study, select the “Next” button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the “Question” button to return to the question.
Overview -
Company Information -
Contoso, Ltd is a multinational retail company that builds, deploys, and manages generative AI and agent-based solutions by using Microsoft Foundry.
Existing Environment -
Identity Environment -
Contoso uses Microsoft Entra ID for identity management, authentication, and authorization capabilities that enable agents to access organizational resources and services.
Contoso recently formed a new AI engineering team named Agent1Dev Team to optimize and maintain existing AI solutions.
The team collaborates with solution architects, DevOps engineers, and security engineers to design, implement. monitor, and secure AI applications.
Contoso also has a team named Agent1Test Team that is responsible for validating AI solutions before the solution deployments.
Generative Environment -
Contoso has a Microsoft Foundry deployment that contains two projects named Project1 and Project2.
Project1 -
Project1 contains a customer support agent named Agent1 that assists customers with product inquiries and troubleshooting requests.
Agent1 has the following configurations:
Agent1 uses a base model deployment.
A safety evaluation pipeline is NOT enabled.
Tool invocation approval workflows are NOT enabled.

Access to the Azure resources must follow the principle of least privilege.
The developers at Contoso must authenticate to Microsoft Foundry resources by using Microsoft Entra authentication.
Access to Project1 must be assigned to the members of Agent1Dev Team by using a security group named SC_Agent1_Dev.
Access to Project1 must be assigned to the members of Agent1Test Team by using a security group named SC_Agent1_Test.
Agent1 must never reveal customer information, even if a document that contains customer data is added erroneously to the product sheet repository in storage1.
The product sheets might contain images that include embedded text. Agent1 must be protected from malicious instructions potentially hidden within the images.
Business Requirements -
Contoso identifies the following business requirements:
Users that interact with Agent1 must have a personalized experience in future interactions, including the ability for Agent1 to retain conversation context and recall relevant information from previous interactions.
Agent1 must answer questions only about the products sold by Contoso.
You need to configure Agent1 to answer customer questions about only the Contoso products. The solution must meet the business requirements.
What should you do?

A. Modify the system message instructions.
B. Add few-shot examples.
C. Apply top-p sampling.
D. Increase the value of the temperature parameter.

Answer: A

Explanation:

Technical Justification for Correct Answer: A. Modify the system message instructions

To address the business requirement that "Agent1 must answer questions only about the products sold by Contoso," modifying the system message instructions is the most appropriate action. Here's why:

Why A is the best choice:

System Message Instructions are used to provide initial or contextual instructions to the model, defining its scope, behavior, or the domain of knowledge it should operate within. By modifying these instructions, you can explicitly limit Agent1's response domain to Contoso's products, aligning with the business requirement for personalized and relevant interactions.
This approach directly influences the model's understanding of its operational boundaries without altering the underlying model's capabilities unnecessarily.

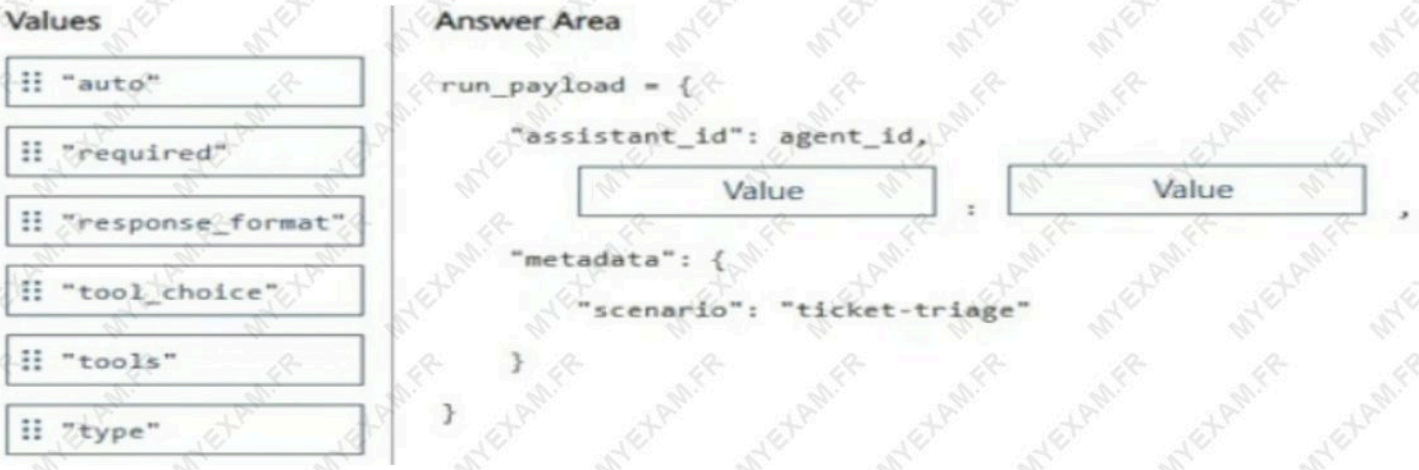
Why other options are less suitable:

B. Add few-shot examples:

While few-shot examples can guide the model towards understanding the desired output for specific inputs, they do not guarantee a strict limitation to Contoso products across all possible queries. This method is more about fine-tuning for specific examples rather than setting a broad operational boundary.
C. Apply top-p sampling:

Top-p sampling is a technique to control the diversity of generated text by only considering the top probabilities. It does not restrict the model's domain knowledge or ensure responses are limited to Contoso products; it merely influences the variability of responses.
D. Increase the value of the temperature parameter:

The temperature parameter in generation models controls the randomness of the output. Increasing it would



Answer:

Explanation:

"tool_choice": This parameter key overrides the default tool execution behavior for a specific thread run.

required": Setting this value forces the agent to deterministically invoke at least one configured tool (such as the MCP tool mentioned in prior scenario requirements) before providing a final response, preventing the agent from bypassing tool execution.

Question: 31

You are building a web app named App1 that generates responses by using a model deployed to a Microsoft Foundry project named Project1.
Before sending the prompts to the model, App1 must retrieve documents by using Azure AI Search.
You need to integrate Project1 and App1. The solution must meet the following requirements:
Multiple client applications must use the same search configuration.
A security policy must prevent key-based authentication.
Administrative effort must be minimized.
What should you do?

A. Create a custom HTTP connection in Foundry and manually configure Azure AI Search endpoints per application.

B. Configure an Azure AI Search connection in Project1 and reference the connection in each application.

C. Call Azure AI Search directly from each application by using Microsoft Entra authentication.

D. Enable a managed identity for each application and call Azure AI Search directly.

What should you do?

A. Create and reuse a conversation by storing the conversation's ID and supplying the ID on subsequent requests.

B. Persist only the final model response stored in the client application and prepend the response to future prompts.

C. Enable memory summarization on the agent definition to persist the context automatically.

Answer: A

Explanation:

Technical Justification for Correct Option (A)

To ensure the Microsoft Foundry Agent Service automatically reloads the complete history on each new turn for resuming support cases with full historical context, **Option A** is the most suitable choice. Here's why:

Why A is Correct:

Multi-turn and Cross-session Continuity: Creating and reusing a conversation by storing the conversation's ID allows for both multi-turn continuity within a session and cross-session continuity for the same case. This approach maintains the conversation's state across different interactions.
Access to Full Interaction History: By supplying the stored conversation ID on subsequent requests, the agent can retrieve the entire interaction history, including user messages, agent messages, tool calls, and tool outputs, ensuring the agent has the full context.
Automatic Reload of History: This method inherently enables the automatic reloading of the conversation's history upon providing the ID, meeting the requirement without needing additional logic for history retrieval.

Why Other Options are Less Suitable:

B. Persist only the final model response:

Inadequate Context: Only storing the final model response would lose the detailed interaction history, making it impossible to fully resume the case with all context.
Lacks Multi-turn Continuity: Prepending the final response to future prompts does not recreate the conversational flow or provide access to all previous messages and tool interactions.
C. Enable memory summarization on the agent definition:

Summarization Limitations: Memory summarization might not capture the entirety of the interaction history with the same fidelity as storing the conversation ID. It's designed for highlighting key points rather than preserving a complete record.
Unclear Persistence Across Sessions: There's no clear indication that memory summarization persists across sessions in the same detailed manner as reusing a conversation ID. It's more suited for in-session context management.

References

Microsoft Documentation: Conversations in Microsoft Bot Framework
Microsoft Foundry Documentation: Managing Conversation State **Note:** Due to the dynamic nature of documentation and the specificity of "Microsoft Foundry Agent Service" (which might be a hypothetical or less-documented service in the provided context), the second link is assumed based on common Microsoft service patterns. For actual implementation, always refer to the most current and specific documentation for your service.

C. Chain of Thought:

Primary Focus: More focused on generating a step-by-step reasoning process for a model's answer rather than efficient, parallel retrieval from an external index.
Not Designed for External Index Interaction: Less optimized for interacting with an external vectorized index in the manner required.

D. Classic Retrieval Augmented Generation (RAG):

Single-Pass Retrieval: Classic RAG typically involves a single retrieval pass, which is less effective for complex, multi-chunk queries compared to the iterative, adaptive approach of Agentic RAG.
Limited Contextual Adaptation: Lacks the agentic capability to adapt retrieval plans based on multi-turn conversation context.

References

Microsoft Azure - Azure Cognitive Search for Complex Queries
Research on Agentic Retrieval Augmented Generation for Conversational AI