TESTPASSPORT

練 習 問 題



高 品 質 な 学 習 リ ソ ー ス

https://www.testpassport.jp

1年間無料アップデート

Exam : **AI-103**

Title : Developing AI Apps and Agents on Azure

Version : DEMO

1. Plan and manage an Azure AI solution

Testlet 1

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.
⇨ Conversation memory constraints are NOT configured.

Agent1 interacts with customers by using digital support channels and answers general questions about Contoso products.
Project1 is deployed to an Azure region located in the European Union (EU). Agent1Dev Team will use Project1 to optimize and maintain Agent1.

Project2
Project2 contains a deployed video generation model. The marketing department at Contoso has access to Project2 and plans to use the model to develop a video creation solution.
Development of the solution is incomplete.

Data Environment
Contoso stores product-related information in Azure resources that support AI applications.
The Azure environment contains an Azure Blob Storage account named storage1 that stores product detail sheets for all the Contoso products.
The product sheets include specifications, feature descriptions, and product support information that Agent1 can use to answer customer questions. The product sheets are stored in the PDF format.

Problem Statements
Contoso identifies the following issues:
⇨ Agent1 has only general knowledge of the Contoso products.
⇨ A recent chat interaction with Agent1 was analyzed for sentiment. The results of the analysis have NOT been processed yet.
⇨ Agent1 does NOT use the detailed product information in the product sheets stored in storage1 when responding to customer questions.
⇨ The finance department at Contoso reports that vendor invoices must be reviewed manually to ensure that the invoices match the terms defined in the vendor contracts. The invoices contain tables, logos, and varied layouts that make the documents difficult to process consistently.

Requirements
Planned Changes
Contoso plans to implement the following changes:
⇨ Implement a solution for Project1 that analyzes the vendor invoices by evaluating both the visual layout and the textual content of the invoices, so that the invoice details can be verified against the vendor contract terms.

☞ Update the base model deployment used by Agent1 and standardize the model version to ensure continuity and consistent responses.
☞ Enable Agent1 to retrieve and use the detailed product information from the product sheets stored in storage1.
☞ Implement an indexing solution for the product sheets that Agent1 can use to answer customer questions.
Complete the development of the video creation solution.

Technical Requirements
Contoso identifies the following technical requirements:
☞ The model deployment used by Agent1 must support scalable, high-throughput generative AI workloads and dynamically scale to handle variable customer support traffic, without requiring reserved throughput capacity.
☞ The product sheets must be processed by using an indexing pipeline that enables semantic and vector search, so that Agent1 can retrieve the relevant product information.
☞ Responses generated by using the product sheet information must be relevant, complete, and accurate.
☞ Agent1 must be able to use the product sheets to answer natural language questions about product details.
☞ The model version used by Agent1 must remain consistent to ensure stable responses.
☞ The data processed by the model must remain within the EU.

Security and Compliance Requirements
Contoso identifies the following security and compliance requirements:
☞ API keys must NOT be used to access Foundry-deployed models.
☞ 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.

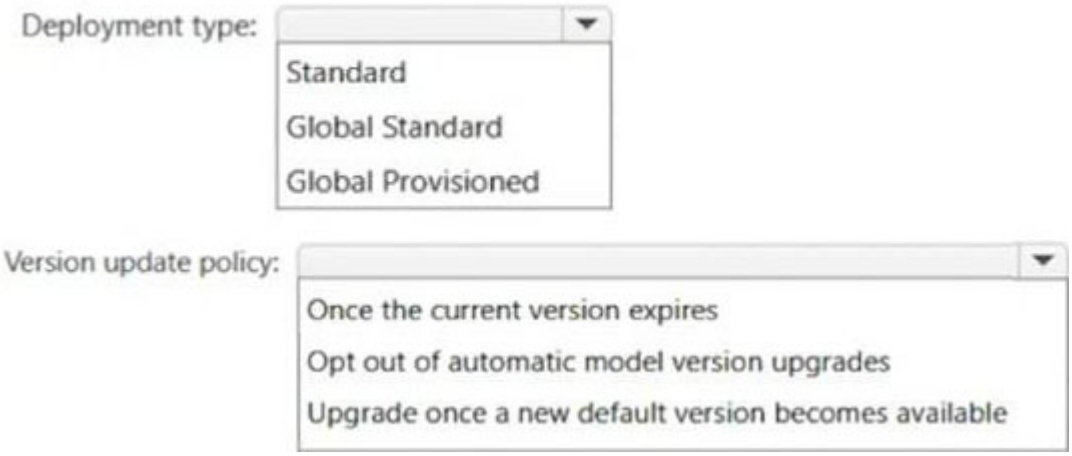
Business Requirements
Contoso identifies the following business requirements:
☞ Users that interact with Agent1 must have a personalized experience in future interactions, including the ability for Agent1 to retain conversation context and recall relevant information from previous interactions.
☞ Agent1 must answer questions only about the products sold by Contoso.

HOTSPOT

You need to configure the model deployment for Agent1 to meet the technical requirements.

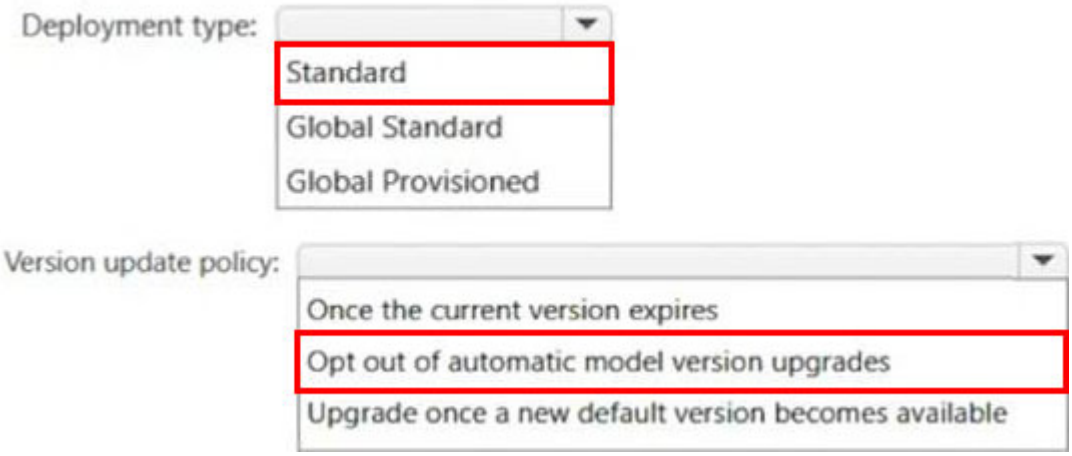
What should you configure? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Answer Area



Answer:

Answer Area



Explanation:

Scenario: Technical Requirements

The model deployment used by Agent1 must support scalable, high-throughput generative AI workloads and dynamically scale to handle variable customer support traffic, without requiring reserved throughput capacity.

The data processed by the model must remain within the EU.

Environment

Project1 contains a customer support agent named Agent1 that assists customers with product inquiries and troubleshooting requests.

Project1 is deployed to an Azure region located in the European Union (EU).

Box 1: Standard

Deployment type

The best deployment type among the choices provided is Standard.

Data Residency: A Standard deployment ensures that both data at rest and inferencing/processing remain strictly within the chosen local Azure region. Because your agent is located in an EU Azure region, selecting a Standard deployment guarantees that data processing does not leave the EU boundary.
No Reserved Capacity: Standard deployments operate on a flexible, pay-as-you-go, on-demand token model. They dynamically scale to accommodate variable customer support traffic without requiring pre-purchased or reserved capacity.
Incorrect:
Here is why Standard is the correct choice, along with explanations of why the other options do not meet the requirements:
Global Standard & Global Provisioned: While global deployments offer the highest initial throughput and leverage Microsoft Azure's global infrastructure to dynamically route traffic, they do not guarantee that data processing stays within the EU. Microsoft specifies that prompts and completions sent to "Global" deployment types can be processed in any global Azure region where capacity is available. This violates your strict requirement that data processed by the model must remain within the EU.
Incorrect:
Standard: This deployment type restricts model inference strictly to a single, local Azure region. It is highly susceptible to regional capacity limits and does not support the massive, highly scalable throughput needed for variable customer traffic compared to global routing.
Global Provisioned: This deployment type requires reserved throughput capacity. It forces you to pre-purchase a fixed number of Provisioned Throughput Units (PTUs), making it poorly suited for scenarios where avoiding reserved capacity is a strict requirement
Box 2: Opt out of automatic model version upgrades
Version update policy
Scenario:
Planned changes: Update the base model deployment used by Agent1 and standardize the model version to ensure continuity and consistent responses.
Technical requirements:
*-> The model version used by Agent1 must remain consistent to ensure stable responses.
The best version update policy for this scenario is Opt out of automatic model version upgrades (also known as specifying a Specific model version or setting to manual upgrade).
Why This Policy Is Best
Ensures Consistent and Stable Responses: By locking the deployment to a exact, specific model version, you prevent sudden shifts in the agent's behavior, latency, reasoning, or tone that naturally occur during automated model switches.
Provides Deployment Control: This policy guarantees that the model will never automatically upgrade behind the scenes. The model version remains pinned until you manually decide to test, validate, and migrate to a newer version.
Protects Customer-Facing Production: Because your agent interacts directly with public users via digital support channels, spontaneous upgrades under an automated policy could accidentally break prompt formatting, alter content safety guardrails, or introduce unexpected hallucinations.
Reference:
https://learn.microsoft.com/en-us/azure/foundry/foundry-models/concepts/deployment-types
https://learn.microsoft.com/en-us/azure/foundry/openai/how-to/working-with-models

2.You need to configure Agent1 to meet the security and compliance requirements.
What should you use?
A. self-harm content filtering
B. prompt shields
C. Personally identifiable information (PII) Detection
D. violence content filtering
Answer: C
Explanation:
To prevent an AI agent from revealing customer data even when sensitive information is accidentally uploaded to the underlying storage, you must configure Agent Guardrails with PII Detection and Tool Response Filtering.
Because the data repository contains unstructured product sheets and could accidentally ingest unstructured customer data, relying solely on access controls at the storage level is not enough.
Scenario:
Security and Compliance Requirements
*-> 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.
Reference: https://techcommunity.microsoft.com/blog/azure-ai-foundry-blog/securely-build-and-manage-agents-in-azure-ai-foundry/4415186

3.You need to configure Agent1 to meet the security and compliance requirements.
What should you include in the solution?
A. prompt shields
B. protected material detection
C. violence content filtering
D. self-harm content filtering
Answer: B
Explanation:
Correct:
* Personally identifiable information (PII) Detection
To prevent an AI agent from revealing customer data even when sensitive information is accidentally uploaded to the underlying storage, you must configure Agent Guardrails with PII Detection and Tool Response Filtering.
Because the data repository contains unstructured product sheets and could accidentally ingest unstructured customer data, relying solely on access controls at the storage level is not enough.
* protected material detection
Protected material detection in Azure AI is a safety feature that scans AI-generated text and code to identify matches against known copyrighted or restricted content.
Text Scanning: Checks model outputs against an index of third-party natural language content, such as song lyrics, articles, recipes, and selected web pages.
Code Scanning: Compares generated code snippets against public GitHub repositories to spot matching

source code and associated software licenses.
Action: Flags, blocks, or reviews the matching output before it reaches the user to reduce intellectual property and copyright risks.
Key Benefits
Legal Compliance: Helps organizations avoid unintentionally publishing or distributing copyrighted text.
Risk Mitigation: Protects businesses from reputation damage and licensing violations.
Originality: Ensures that AI agents produce original text and custom code rather than reproducing restricted works.
Incorrect:
* prompt shields
* self-harm content filtering
* violence content filtering
Scenario:
Security and Compliance Requirements
*-> 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.
Reference: https://techcommunity.microsoft.com/blog/azure-ai-foundry-blog/securely-build-and-manage-agents-in-
azure-ai-foundry/4415186
https://learn.microsoft.com/th-th/azure/ai-services/content-safety/concepts/protected-material

4. Plan and manage an Azure AI solution
Question Set 2

You are planning a Microsoft Foundry project named Project1 that will contain multiple agents. Each agent will access the same Azure AI Search resource.
You need to recommend a solution to centrally manage the Azure AI Search credentials within Project1. The solution must be implemented across all the agents.
What should you recommend?
A. Enable role-based access control (RBAC) for the Azure AI Search resource.
B. Disable key-based access control on the Azure AI Search resource.
C. Add a connection to the Azure AI Search resource.
D. Create a managed private endpoint that connects to the Azure AI Search resource.
Answer: C
Explanation:
To best manage security and centrally handle credentials across multiple agents, you should add a connection to the Azure AI Search resource at the Azure AI Foundry project level.
Why This Works
Central Hub: The project acts as the single security perimeter for all your agents.
Credential Masking: Agents inherit access without hardcoding secrets, API keys, or connection strings in their code.
Identity Management: It allows you to leverage Microsoft Entra ID (formerly Azure AD) for role-based

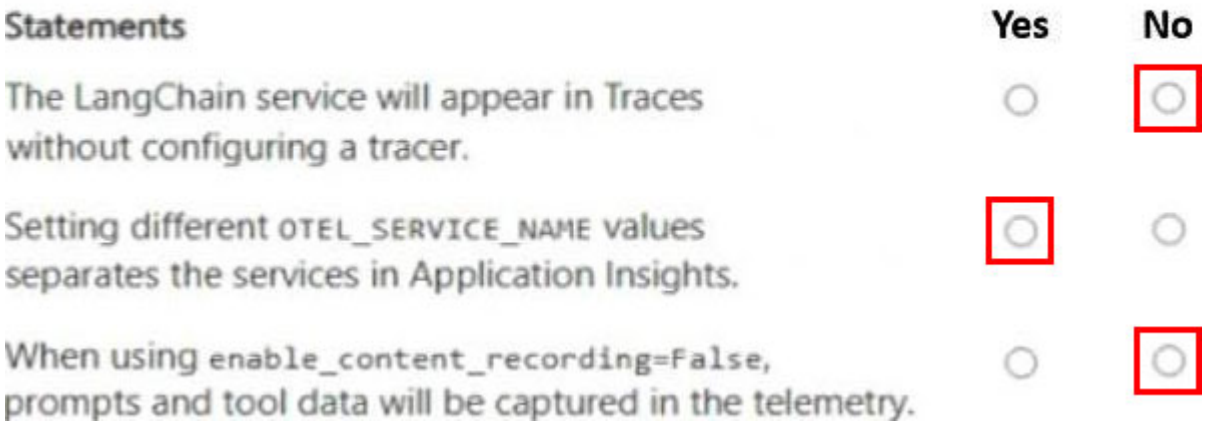
access control (RBAC).
How to Implement It
1. Navigate to your Azure AI Foundry portal.
2. Select your specific project from the dashboard.
3. Open the "Management Center" or "Project settings" tab.
4. Click on "Connected resources" or "Connections".
5. Add the Azure AI Search resource.
6. Choose Entra ID (managed identity) over API keys for maximum security.
Reference: https://partner.microsoft.com/en-us/blog/article/azure-updates-december-2025

5.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.
Answer Area

Statements	Yes	No
The LangChain service will appear in Traces without configuring a tracer.	○	○
Setting different `OTEL_SERVICE_NAME` values separates the services in Application Insights.	○	○
When using `enable_content_recording=False`, prompts and tool data will be captured in the telemetry.	○	○

Answer:

Answer Area



Explanation:

Box 1: No

No - The LangChain service will appear in Traces without configuring a tracer.

The LangChain service will not appear in Traces without configuring a tracer.

Because OpenTelemetry requires an explicit exporter or tracer provider to collect and send data, you must provide a valid tracer registration. However, since configure_azure_monitor() sets up a global OpenTelemetry tracer provider under the hood, any standard OpenTelemetry-instrumented service sharing the same process could theoretically export traces globally. Despite this, LangChain specifically relies on explicit callback handlers (like AzureAIOpenTelemetryTracer) passed in the config to map LangChain-specific execution blocks (chains, tools, and prompts) into OpenTelemetry spans. Without configuring and passing this tracer, your LangChain orchestrations will not emit telemetry to Application Insights.

Box 2: Yes

Yes - Setting different OTEL_SERVICE_NAME values separates the services in Application Insights.

Setting different OTEL_SERVICE_NAME values distinguishes the services within the same Application Insights resource.

In Azure Monitor and Application Insights, the OpenTelemetry service.name attribute automatically maps to the cloud_RoleName property on your telemetry logs and spans. By specifying distinct names via OTEL_SERVICE_NAME (or setting it in the tracer providers), you can isolate and filter telemetry originating from the core Foundry Agent Service versus the LangChain components.

Box 3: No

No - When using enable_content_recording=False, prompts and tool data will be captured in the telemetry.

Prompt text and tool data will not be captured in the telemetry.

Setting enable_content_recording=False specifically instructs the AzureAIOpenTelemetryTracer to redact and mask user prompts, LLM responses, and tool call arguments from being emitted into span attributes. This configuration successfully prevents credentials, secrets, or sensitive PII residing inside the text from bleeding into your Azure Application Insights resource.

Reference:

https://learn.microsoft.com/en-us/azure/azure-monitor/app/opentelemetry-configuration

https://learn.microsoft.com/en-us/azure/foundry/how-to/develop/langchain-traces