DUMPSBOSS.

Operationalizing Machine Learning and Generative AI Solutions

Microsoft AI-300

Version Demo

Total Demo Questions: 5

Total Premium Questions: 57

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Manage an Azure Machine Learning workspace	6
Topic 2, Manage data and compute for machine learning	4
Topic 3, Train and evaluate machine learning models	6
Topic 4, Deploy and manage machine learning models	7
Topic 5, Implement generative AI solutions	26
Topic 6, Monitor and maintain AI solutions	8
Total	57

QUESTION NO: 1 - (HOTSPOT)

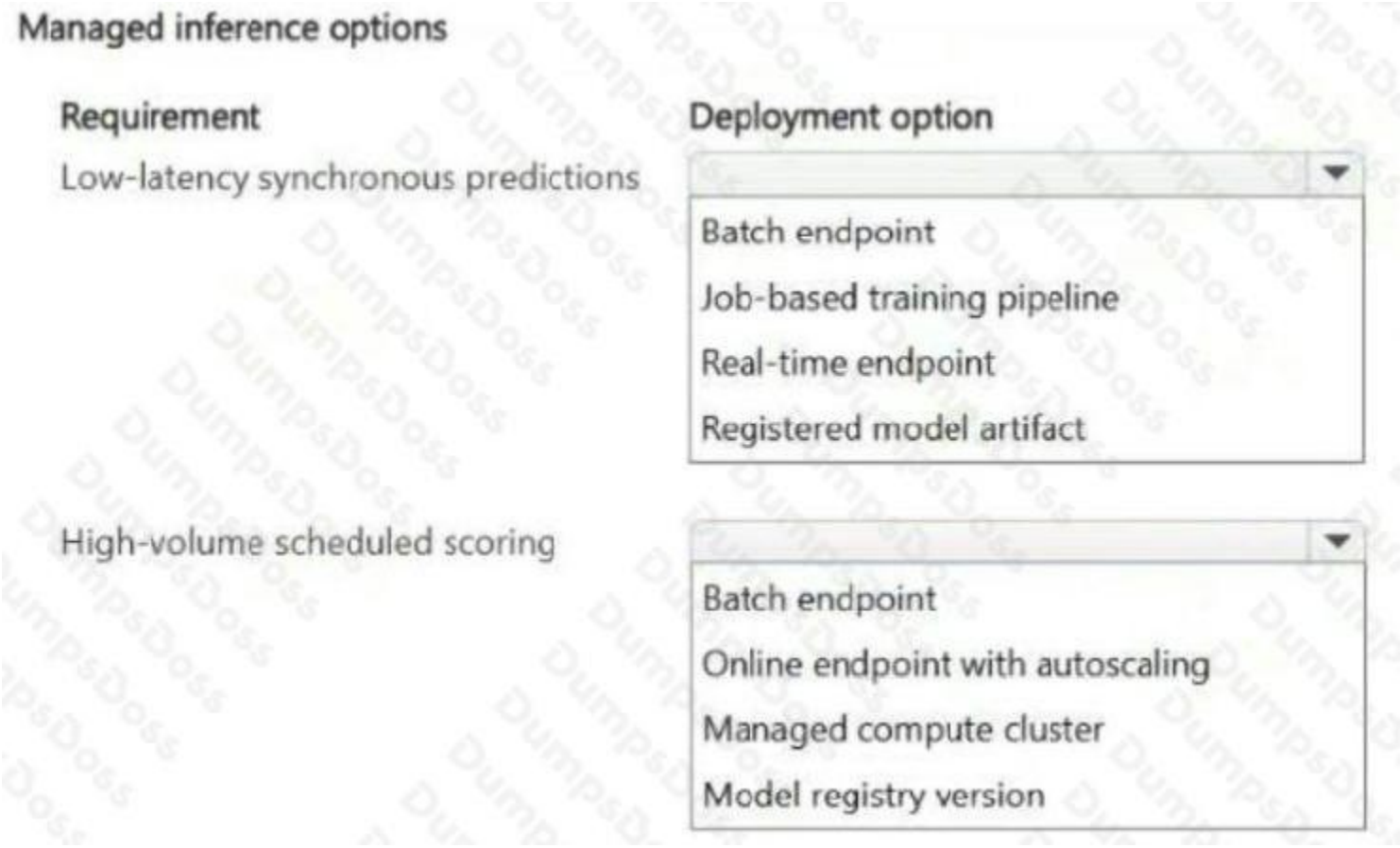
HOTSPOT -

A team trains an MLflow model that scores customer churn risk. The model will be consumed by different downstream systems. One system requests predictions synchronously during customer interactions.

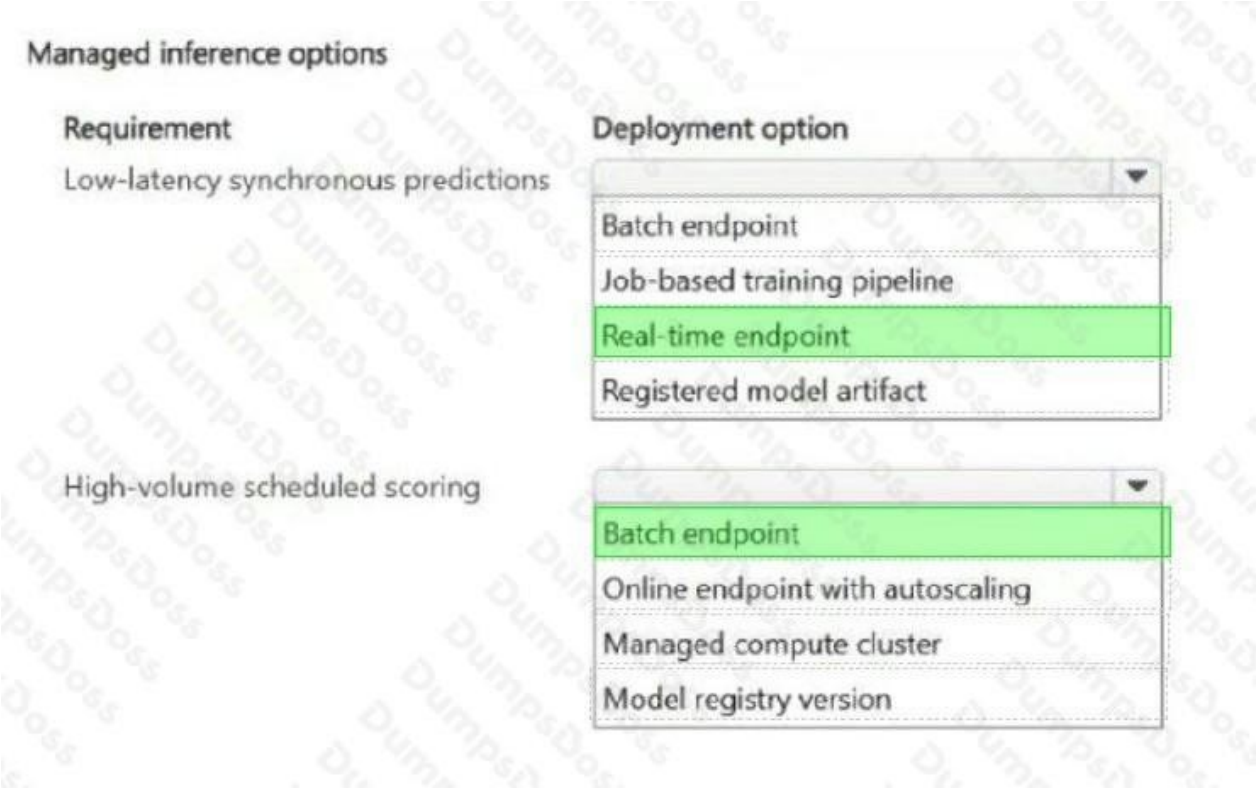
Another system submits files containing millions of records for scheduled scoring.

You need to deploy the model by using managed inference options that match each usage pattern.

Which option should you use for each usage pattern? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.



ANSWER:



Explanation:

For low-latency synchronous predictions during live customer interactions, you should deploy the MLflow model to a managed online (real-time) endpoint. Online endpoints are built for request/response scoring over HTTPS, keeping compute warm so the model can return predictions quickly and consistently. This is the standard managed inference choice when an application needs immediate results as part of an interactive workflow.

For high-volume scheduled scoring where another system submits files containing millions of records, you should use a managed batch endpoint. Batch endpoints are designed for asynchronous, large-scale inference: you point the job at data in storage, Azure Machine Learning spins up the required compute, processes the dataset in parallel, and writes outputs back to storage. This pattern fits scheduled scoring and large file-based workloads because it optimizes throughput and cost rather than per-request latency.

These two endpoint types are the core managed inference options in Azure Machine Learning for interactive versus offline scoring. Online endpoints (real-time) target low latency and can be configured for scaling, while batch endpoints target high throughput for large datasets and scheduled runs. References: Azure ML online endpoints and Azure ML batch endpoints.

QUESTION NO: 2 - (DRAG DROP)

DRAG DROP -

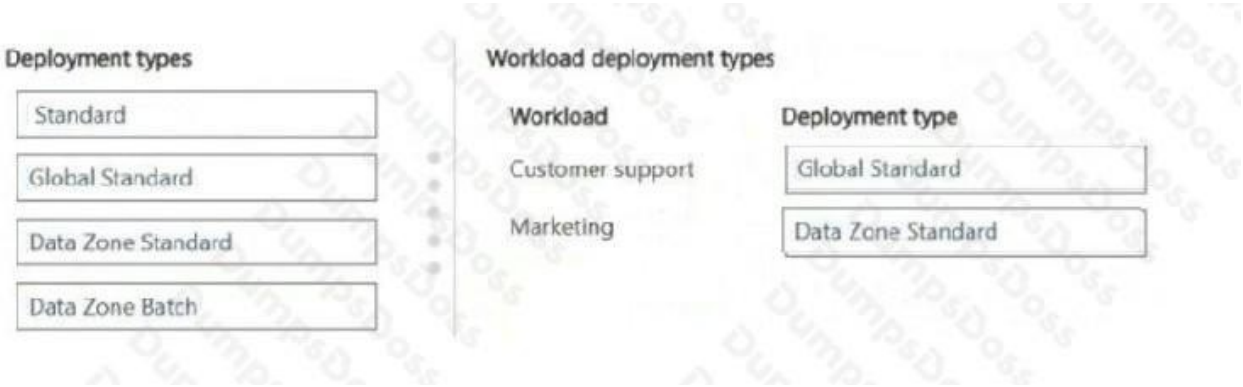
A company plans to deploy a foundation model in Microsoft Foundry. The mode must support the following workloads:

A customer support workload used across multiple regions

A marketing workload that must remain within a specific region due to data residency requirements You need to select the deployment type.

Which deployment type should you use for each workload? To answer, move the appropriate deployment types to the correct requirements. You may use each deployment type once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content. NOTE: Each correct selection is worth one point.

ANSWER:



Explanation:

For the customer support workload, the key requirement is that it's used across multiple regions. In Microsoft Foundry, the deployment type that best matches a multi-region, globally distributed usage pattern is **Global Standard**. This deployment type is intended to provide broad geographic availability so users in different regions can access the model with appropriate routing and performance characteristics, which is exactly what you want for a customer support solution serving multiple geographies.

For the marketing workload, the requirement is explicitly about **data residency**: the workload must remain within a specific region. In Foundry, the deployment type that aligns to keeping inference and data handling within a defined geographic boundary is **Data Zone Standard**. "Data Zone" deployments are designed to constrain processing to a particular data zone/region to help meet residency and sovereignty requirements, while "Standard" indicates it's an online, real-time style deployment suitable for interactive marketing use cases.

Putting those together, you use **Global Standard** when the workload needs multi-region reach, and **Data Zone Standard** when the workload must stay within a specific region for residency. This matches the intent of Foundry's deployment options: global distribution for broad access versus data-zone constrained deployment for compliance-driven locality.

References: Azure AI Foundry deployments overview, Azure AI Foundry data zones

QUESTION NO: 3 - (DRAG DROP)

DRAG DROP -

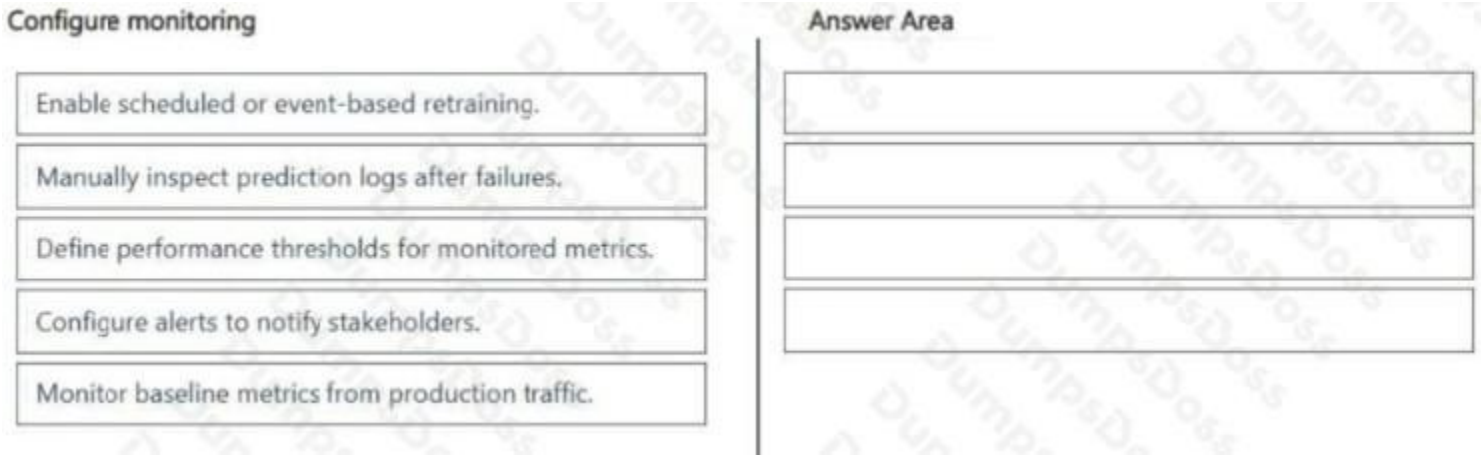
A team deploys a classification model to production and monitors performance and data changes.

The team wants to ensure that significant drops in prediction accuracy automatically trigger the following: Stakeholders must be notified of the drops.

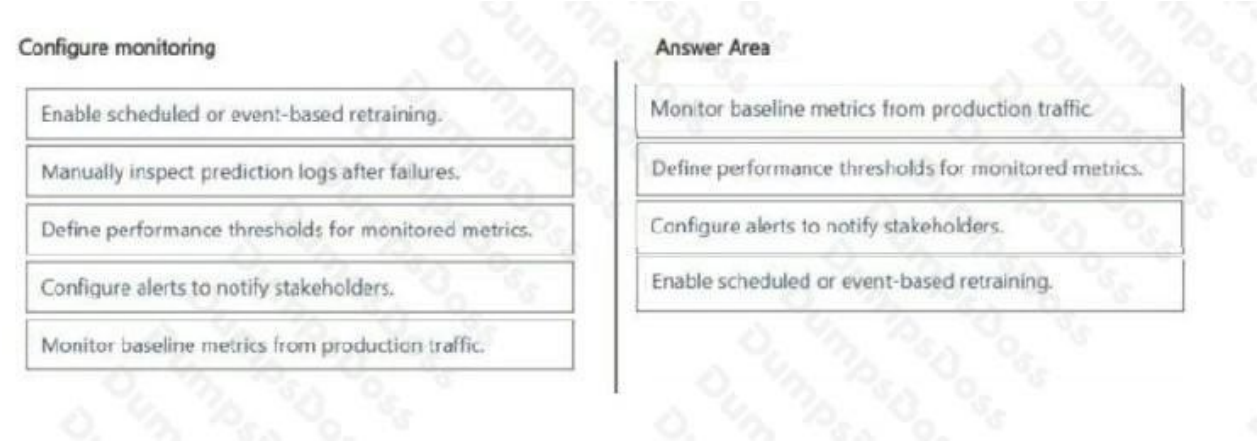
Retraining must be initiated when thresholds are exceeded

You need to configure monitoring to meet the requirements.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



ANSWER:



Explanation:

To meet the requirement that accuracy drops automatically notify stakeholders and also kick off retraining, you configure monitoring in the same way you'd build any operational control loop: measure, decide, act, and automate. Start by monitoring baseline metrics from production traffic so the system is actually collecting the performance signals you care about (for example, accuracy-related metrics computed from labeled feedback, or proxy metrics and drift indicators when labels arrive later). Once metrics are being tracked, define performance thresholds for the monitored metrics so there is an explicit trigger condition that represents a "significant drop." With thresholds in place, configure alerts to notify stakeholders so that when the monitored metric crosses the threshold, notifications are sent automatically through the configured action group/channel. Finally, enable scheduled or event-based retraining so that the same threshold breach (or the monitoring event that represents it) can initiate a retraining pipeline/job without manual intervention. This sequence creates an automated monitoring-and-response workflow that satisfies both notification and retraining initiation requirements. For more details on Azure Monitor alerts and action groups, see Azure Monitor alerts overview and Create and manage action groups. For retraining automation patterns in Azure Machine Learning, see Schedule pipeline jobs.

QUESTION NO: 4 - (HOTSPOT)

HOTSPOT -

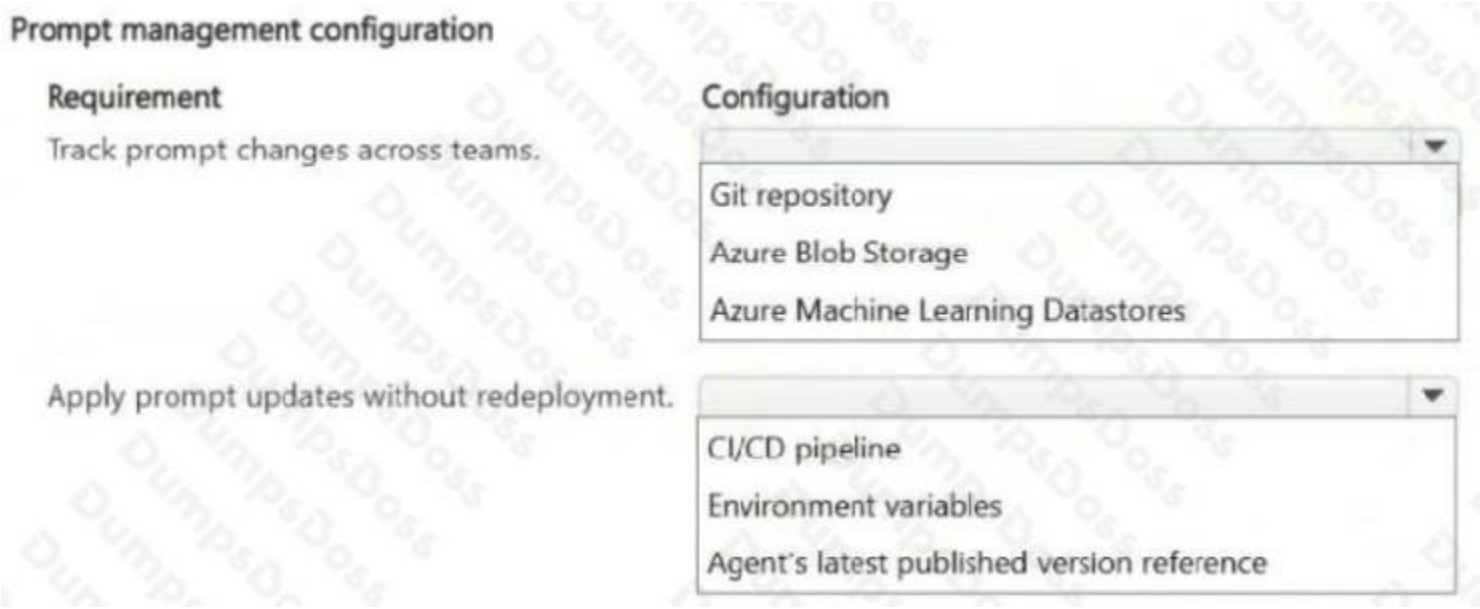
A team is building a generative AI agent by using Retrieval-Augmented Generation (RAG) in Microsoft Foundry.

The team frequently updates prompt content. The team must be able to track changes across contributors while avoiding full application redeployments.

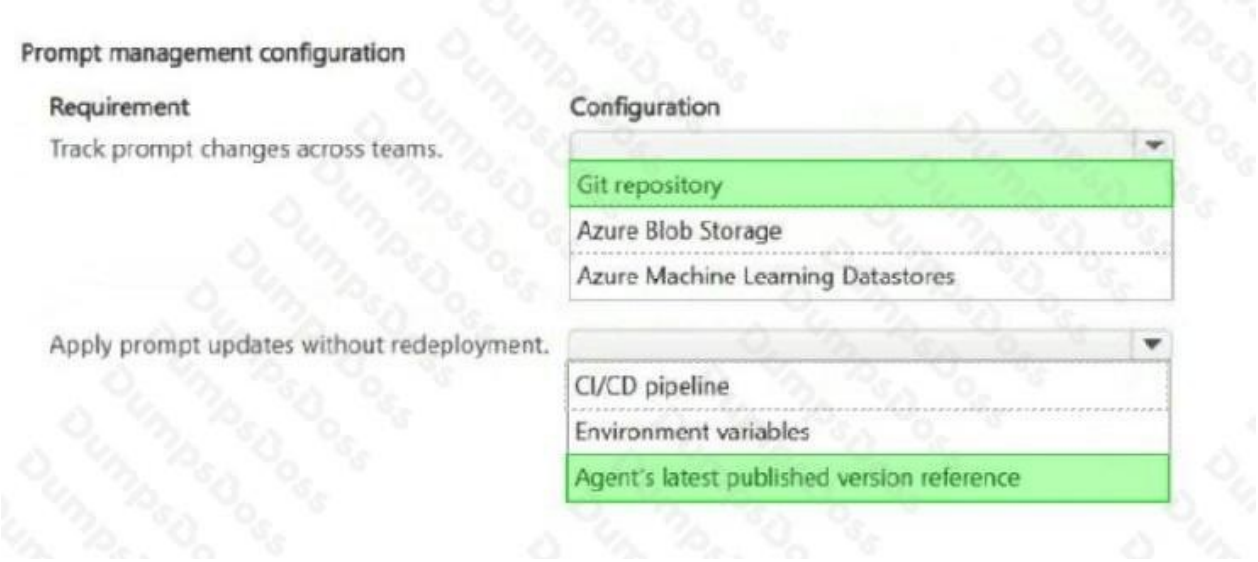
You need to enable rapid prompt iteration with traceability. Applications consuming the agent must be able to use updated prompts without

requiring redeployment.

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



ANSWER:



Explanation:

To enable rapid prompt iteration with traceability, you want prompt content to live in a system that natively supports collaboration features like commit history, author attribution, pull requests, and easy rollback. A **Git repository** is purpose-built for this: every change is versioned, you can review and approve updates, and you can audit exactly who changed what and when. This directly satisfies the need to track prompt changes across contributors and teams while iterating quickly.

To ensure applications consuming the agent can use updated prompts **without requiring redeployment**, the consuming app should avoid hard-coding a specific prompt version or bundling prompt files into the application package. Instead, it should call the agent using an **agent reference that always resolves to the most recent published version**. Configuring the app to use the **Agent's latest published version reference** means that when the team publishes a new agent version (with updated prompts), the application continues calling the same logical endpoint/reference and automatically gets the newest published prompt behavior—no application redeploy needed. This aligns with common "version alias" patterns where the runtime resolves "latest" to the current published artifact.

References: Git-based collaboration and versioning concepts are described at https://learn.microsoft.com/en-us/devops/develop/git/what-is-git. For Azure AI Foundry agent concepts and publishing/versioning patterns, see https://learn.microsoft.com/en-us/azure/ai-foundry/.

QUESTION NO: 5

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 need 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.

Background -

Fabrikam Inc. is a mid-sized healthcare analytics company that provides population health dashboards and predictive insights to regional hospital systems across the United States. Fabrikam Inc. customers rely on near real time analytics to monitor patient flow, staffing needs, and readmission risks. They use multiple traditional forecasting machine learning models for predictions.

Fabrikam Inc. has an established Microsoft Azure footprint. The company uses Jupyter Notebooks that run on a local server as the primary development environment. The data science team is experiencing scalability, asset management and code management issues with the current development platform. Fabrikam Inc. plans to migrate to a cloud-based development environment to mitigate the issues.

Additionally, the company plans to implement a Retrieval-Augmented Generation (RAG)-based chat application for client support. Leadership requires the application to be developed and deployed with a low operational risk.

Current Environment -

Fabrikam Inc. operates a single Azure subscription that has the following components:

Azure Data Lake Storage Gen2 that contains de-identified clinical and operational datasets Azure AI Search indexing curated analytical documents and reference materials

A small set of Python-based training scripts maintained by data scientists Azure OpenAI Service with deployed foundational models

A Microsoft Foundry resource for building a RAG-based solution

Evaluation data has manually defined expected responses.

The current challenges faced by the data science team include the following:

Model training jobs are run manually from notebooks. Experiment tracking is inconsistent

Model versions are registered without standardized metadata.

Deployment is performed manually by data scientists, with limited rollback capability. The team has no standardized evaluation process for generative AI outputs.

The environment currently allows public network access. Authentication relies on user accounts rather than managed identities. Compute targets are manually created and shared across experiments. This has led to resource contention during peak usage.

Business Requirements -

Fabrikam Inc. has the following business requirements for the modernization initiative:

Provide a conversational interface that answers analytics questions by using internal documents and datasets. Ensure that sensitive healthcare-related data is not exposed outside the Fabrikam Inc. Azure tenant.

Enable repeatable and auditable model training and deployment processes. Support experimentation to compare prompt strategies and fine-tuned models.

Align the model with the ranked preferences and optimize behavior for the long term. Minimize disruption to existing analytics workloads during rollout.

Technical Requirements -

To support the business goals, Fabrikam Inc. identifies these technical requirements:

Use Azure Machine Learning workspaces to centrally manage data assets, models, and environments. Implement experiment tracking and model versioning for all training jobs.

Orchestrate training and evaluation by using pipelines rather than manually running notebooks. Deploy traditional machine learning models with support for staged rollout and rollback.

Improve RAG-based solution output quality.

Use the existing evaluation datasets that are based on real data with input-output pairs.

Issues and Constraints -

Fabrikam Inc. must comply with internal security policies that require the company to restrict network access and avoid long-lived secrets. The data science team has limited Azure DevOps experience, so solutions must favor managed services and automation over custom infrastructure. Cost predictability is important. Leadership prefers serverless or managed compute options where possible but is willing to approve dedicated compute for stable production workloads.

Problem Statement -

Fabrikam Inc. must design and implement an Azure-based AI operations solution that enables reliable training, evaluation, deployment, and iteration of generative AI models. The solution must support experimentation and gradual rollout while ensuring governance, security, and operational stability. The data science and platform teams must collaborate to deliver this solution by using Azure Machine Learning and Microsoft Foundry capabilities.

You need to isolate training workloads while remaining cost-aware to address Fabrikam Inc.’s issues, constraints, and technical requirements.

What should you implement?

A. Training jobs that run on a single shared compute cluster

B. Fixed-size compute cluster

C. Dedicated compute clusters per experiment

D. Managed compute targets with autoscaling

ANSWER: D

Explanation:

Managed compute targets with autoscaling is the best fit because Azure Machine Learning compute clusters are a managed service designed to run training jobs in an isolated, repeatable way while controlling cost through scale-out/scale-in behavior. With an AML compute cluster, each submitted job runs in its own isolated environment (containerized execution) and can be scheduled without requiring data scientists to manually create and share ad-hoc compute targets, which directly addresses the current resource contention problem. Autoscaling also supports cost predictability: you can set minimum nodes to 0 for dev/test so the cluster scales down when idle, and set maximum nodes to cap spend during peak usage. This approach aligns with the requirement to orchestrate training via pipelines and to standardize experiment tracking and model versioning, because pipelines can target the same managed compute and reliably reproduce runs. It also supports the

security constraint to avoid long-lived secrets by enabling managed identity-based access from jobs to data stores and other Azure resources, while keeping workloads within the tenant’s controlled network boundaries when combined with private networking patterns.

References: Azure Machine Learning compute targets, Create an Azure Machine Learning compute cluster (autoscale)