



Level: Advanced

Microsoft Azure Exam AZ-104 Certification

[← Back to the Course](#)

Azure Subscriptions & Governance – Practice Mode

Completed on Sun, 12 Oct 2025

1st
Attempt2/5
Marks Obtained40.00%
Your ScoreFAIL
ResultShare this Report in Social Media [Share](#)[Download Report](#)

Domain wise Quiz Performance Report

No.	Domain	Total Question	Correct	Incorrect	Unattempted	Marked for Review
1	Manage Azure identities and governance	5	2	3	0	0
Total	All Domains	5	2	3	0	0

Review the Answers

[Filter By](#)

Question 1

Incorrect

Domain: Manage Azure identities and governance

[View Case Study](#)

Fabrikam must ensure that resource creation in all subscriptions is restricted to East US and West Europe regions. The solution must also remediate existing resources that violate this policy. What should you do?

- A. Assign the Allowed locations policy with a Deny effect at the FabrikamRoot management group level
- B. Create a custom policy definition to restrict regions, set the Modify effect, and assign it at the subscription level. wrong
- C. Apply the built-in Allowed locations policy at the management group level with the deployIfNotExists effect. right

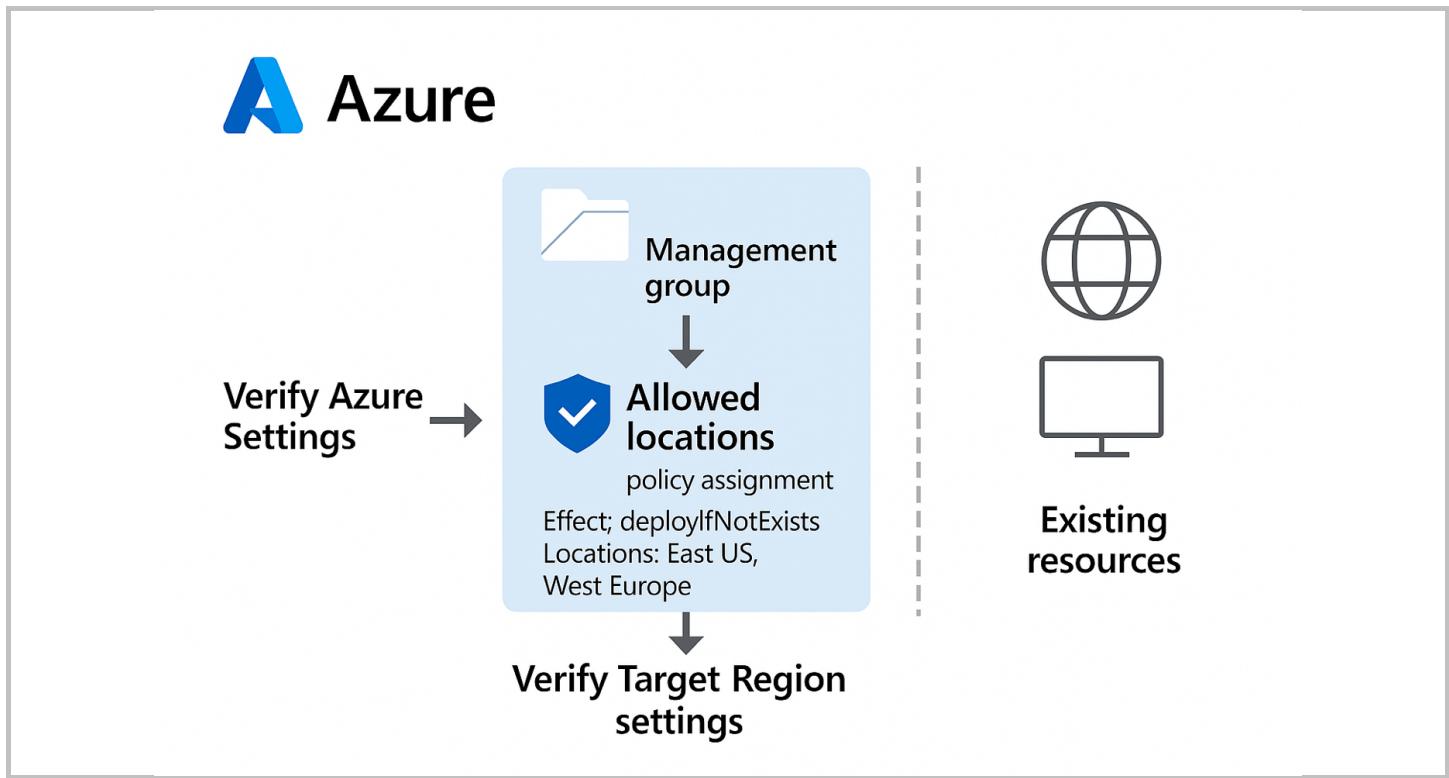
D. Add the Allowed locations policy to a new initiative and assign it at the subscription level with the Audit effect

Explanation:

Correct Answer: C

Option C: Apply the built-in Allowed locations policy at the management group level with the deployIfNotExists effect is correct

because the Allowed locations policy can limit resource creation to the specified regions (East US and West Europe). The deployIfNotExists effect ensures that any existing resources outside the allowed locations can be remediated. For example, it can flag and correct non-compliant resources by moving them to compliant regions or applying remediation logic as defined by the policy. Applying the policy at the management group level ensures consistent governance across all subscriptions, eliminating the need for repeated assignments and ensuring scalability for future subscriptions under the hierarchy. Simplified governance by centralizing policy management at the management group level. Built-in policies reduce operational overhead compared to creating and managing custom policies. Ensures compliance with corporate standards across all existing and future subscriptions.



Option A: Assign the Allowed locations policy with a Deny effect at the FabrikamRoot management group level is incorrect because while the Deny effect ensures that new resources cannot be created outside the specified regions, it does not address the requirement to remediate existing non-compliant resources. A Deny effect only prevents new violations but does not apply corrective actions to existing resources. Fabrikam explicitly requires remediation of already non-compliant resources, which this approach does not fulfill.

Option B: Create a custom policy definition to restrict regions, set the Modify effect, and assign it at the subscription level is incorrect because the Modify effect is primarily used to enforce or update specific resource configurations (e.g., adding tags or setting parameters). It is not suitable for restricting resource creation to specific regions. Additionally, the effort to create and manage a custom policy is unnecessary when a built-in policy, such as Allowed locations, already exists and can meet the requirements. Assigning it only at the subscription level also lacks scalability for enforcing organization-wide governance.

Option D: Add the Allowed locations policy to a new initiative and assign it at the subscription level with the Audit effect is incorrect

because the Audit effect only monitors and logs non-compliant resources but does not prevent new violations or remediate existing issues. While assigning the policy as part of an initiative could improve the organization, using the Audit effect fails to enforce compliance or take corrective action, making it insufficient to meet Fabrikam's requirements.

References:

<https://learn.microsoft.com/en-us/azure/governance/policy/concepts/effect-basics>

<https://learn.microsoft.com/en-us/azure/governance/policy/samples/built-in-policies#general>

[Ask our Experts](#)

Did you like this Question?

**Question 2**

Correct

Domain: Manage Azure identities and governance

[View Case Study](#)

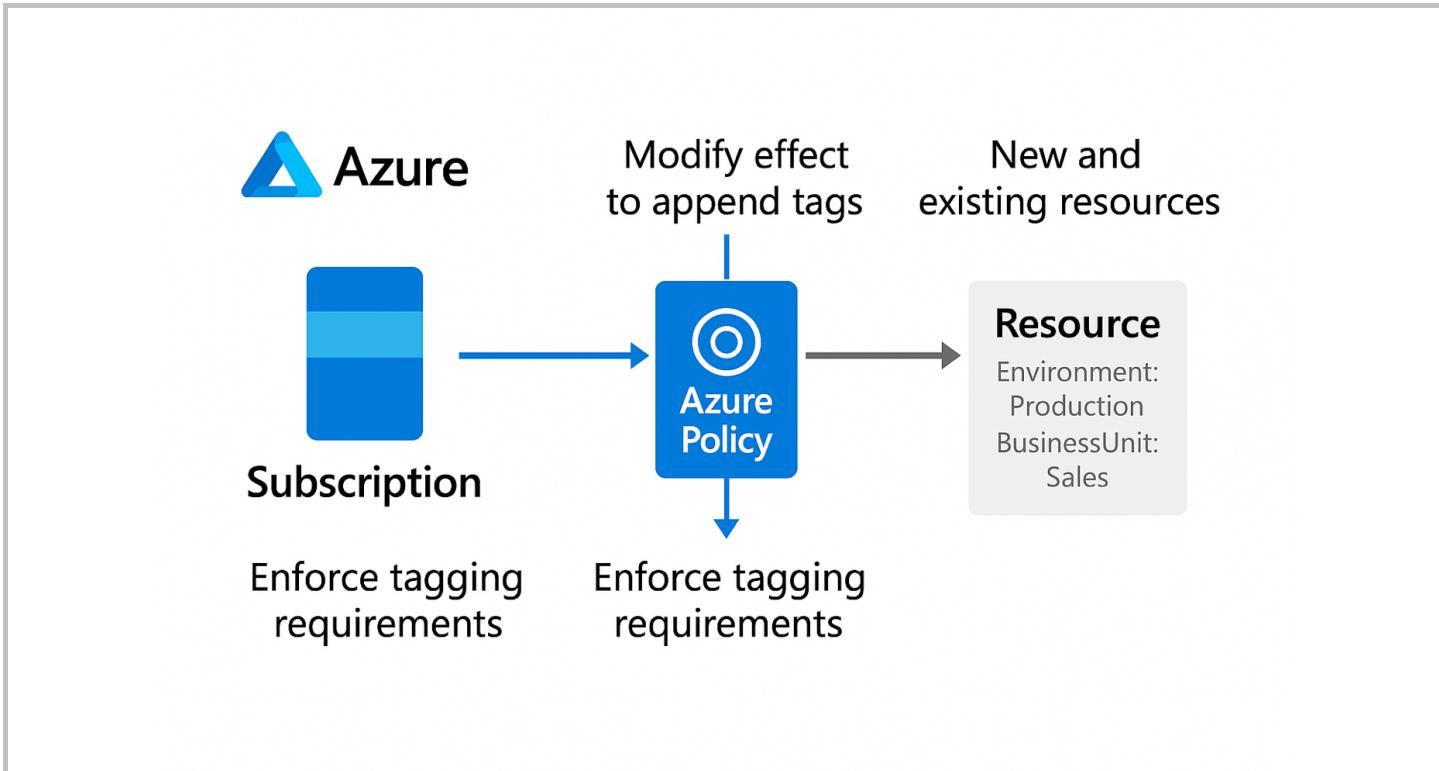
Fabrikam requires all resources to include the tags Environment and BusinessUnit, both for existing and newly created resources. How should this requirement be implemented?

- A. Assign the Append tag and its value policy at the root management group to enforce tagging for all new resources
- B. Manually apply the required tags to existing resources and create a policy to enforce tags for future resources
- C. Assign the Modify effect at the management group level without remediation for existing resources
- D. Use the Modify effect in Azure Policy to append tags and assign them to each subscription right

Explanation:

Correct Answer: D

Option D: Use the Modify effect in Azure Policy to append tags and assign them to each subscription is correct because the Modify effect in Azure Policy is capable of both enforcing tags for new resources and retroactively applying tags to existing resources through remediation tasks. The Modify effect ensures that tags are added automatically during resource creation, even if they are not explicitly defined by the resource creator. Azure Policy allows remediation tasks to bring existing resources into compliance. By using the Modify effect, Fabrikam can append the required tags (Environment and BusinessUnit) to all existing resources in the scope. Assigning the policy to each subscription ensures that all resources within those subscriptions are compliant. While assigning at the management group level might simplify governance, assigning directly at the subscription level provides flexibility to target specific subscriptions if needed. Ensures full compliance with tagging requirements for both existing and future resources. Utilizes Azure Policy's built-in capabilities, minimizing manual effort. Provides visibility and control over resources for better cost and operational management.



Option A: Assign the Append tag and its value policy at the root management group to enforce tagging for all new resources is incorrect because the Append effect ensures that a tag and its value are added during the creation of new resources. However, it does not address the requirement to tag existing resources, which is a critical part of Fabrikam's requirements. Assigning the policy at the root management group level ensures coverage across all subscriptions, but it falls short of meeting the requirement for existing resources.

Option B: Manually apply the required tags to existing resources and create a policy to enforce tags for future resources is incorrect because it relies on a manual approach to tag existing resources. Manual tagging is time-consuming, prone to errors, and difficult to scale, especially for organizations with a large number of resources. Moreover, this approach does not take advantage of Azure Policy's automated remediation capabilities, which can efficiently tag resources without manual intervention. While creating a policy for future resources is valid, the manual tagging step does not align with best practices for automation and governance.

Option C: Assign the Modify effect at the management group level without remediation for existing resources is incorrect because assigning the Modify effect at the management group level without remediation for existing resources would only apply the tagging policy to newly created resources. The requirement specified in the case study is to ensure that both existing and new resources are tagged with Environment and BusinessUnit. Without remediation for existing resources, this policy would not address the existing non-compliant resources, failing to meet the full tagging requirement. Therefore, this option does not fully satisfy the needs outlined in the case study.

Reference:

<https://learn.microsoft.com/en-us/azure/governance/policy/concepts/effect-modify>

Ask our Experts

Did you like this Question?



Question 3

Incorrect

Domain: Manage Azure identities and governance

[View Case Study](#)

The finance team at Fabrikam requires an email alert when spending on the Research subscription exceeds 75% of its \$15,000 monthly budget. What steps should the IT team take?

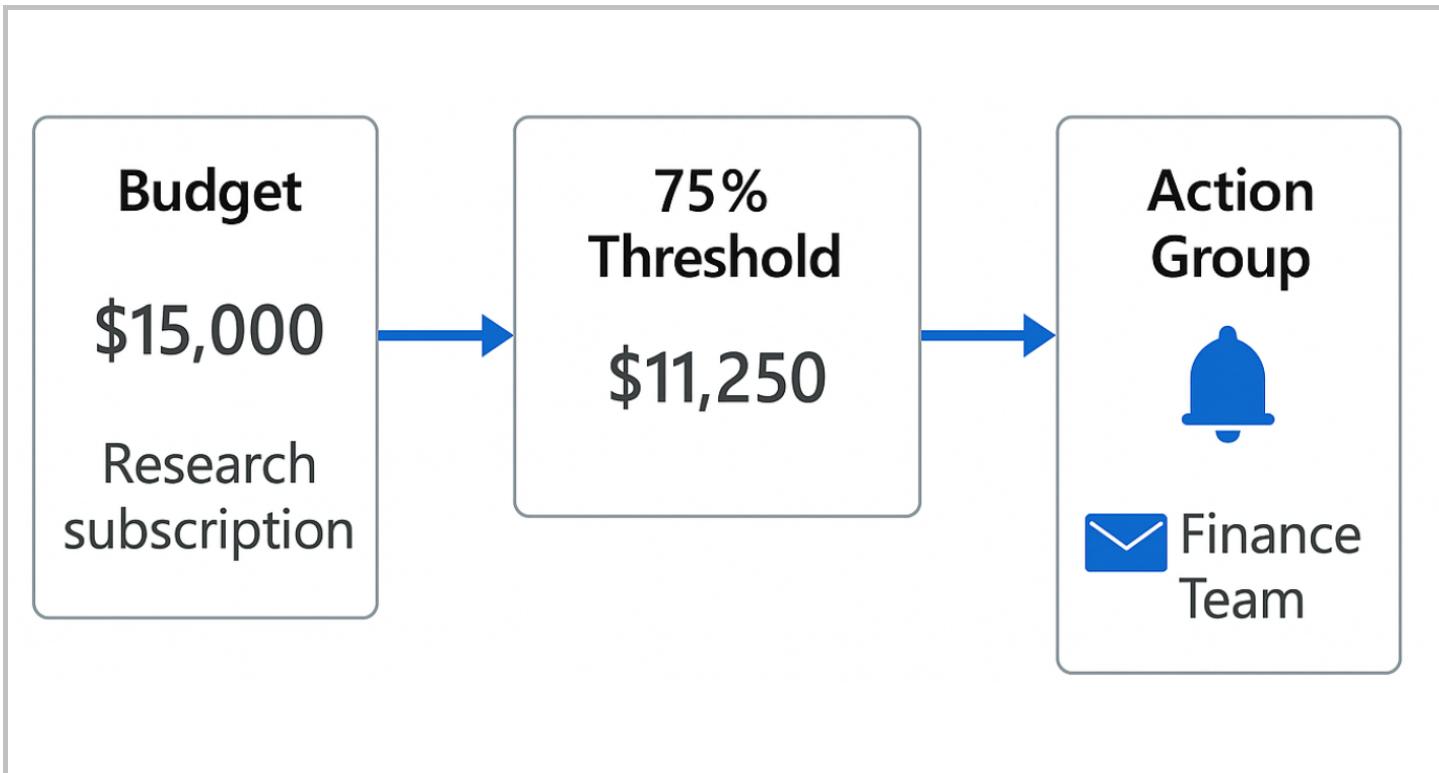
- A. Configure a cost alert in Azure Advisor and create an action group for email notifications wrong
- B. Set up a budget in Azure Cost Management, configure a 75% threshold, and assign an action group for email notifications right
- C. Use Azure Monitor to create a custom alert rule and configure email notifications for cost thresholds
- D. Implement a Logic App to monitor subscription spending and trigger an email when the threshold is exceeded

Explanation:

Correct Answer: B

Option B: Set up a budget in Azure Cost Management, configure a 75% threshold, and assign an action group for email notifications is **correct** because Azure Cost Management allows you to create budgets for subscriptions, resource groups, or services. The steps include:

1. Creating a Budget: Define a budget of \$15,000 for the Research subscription within Azure Cost Management.
2. Setting Alerts: Configure a threshold of 75% of the budget. This means an alert will trigger when spending reaches \$11,250.
3. Notification Configuration: Assign an action group to the budget alert to send an email notification to the finance team. Action groups allow multi-channel notifications, including email, SMS, and webhook.



Azure Cost Management budgets are specifically designed for monitoring and alerting spending thresholds, making this the most appropriate and efficient solution. Additionally, this approach aligns with Azure's built-in capabilities for cost governance, reducing the need for custom solutions.

Option A: Configure a cost alert in Azure Advisor and create an action group for email notifications is incorrect because Azure Advisor provides cost optimization recommendations but does not have a feature to configure cost alerts directly. While Azure Advisor integrates with recommendations for reducing expenses and improving cost efficiency, it cannot be used to set thresholds for budget monitoring or trigger alerts when spending crosses a specific percentage. Action groups can indeed be used for notifications, but they cannot be triggered by Advisor directly for budget-related scenarios.

Option C: Use Azure Monitor to create a custom alert rule and configure email notifications for cost thresholds is incorrect because Azure Monitor is primarily used for tracking metrics, logs, and telemetry data from Azure resources and applications. While custom alert rules can be created in Azure Monitor for various conditions, it is not the recommended solution for monitoring cost thresholds. Azure Cost Management is explicitly designed for this purpose, making it a more efficient and straightforward approach. Using Azure Monitor for cost alerts would introduce unnecessary complexity and require additional setup for tracking cost data.

Option D: Implement a Logic App to monitor subscription spending and trigger an email when the threshold is exceeded is incorrect because using a Logic App to monitor spending and trigger email notifications involves significant manual effort and custom



Support



would require integrating with cost APIs and managing the workflow logic, which is unnecessary given the availability of out-of-the-box features in Azure Cost Management.

Reference:

<https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-acm-create-budgets?tabs=psbudget>

[Ask our Experts](#)

Did you like this Question?



Question 4

Correct

Domain: Manage Azure identities and governance

[View Case Study](#)

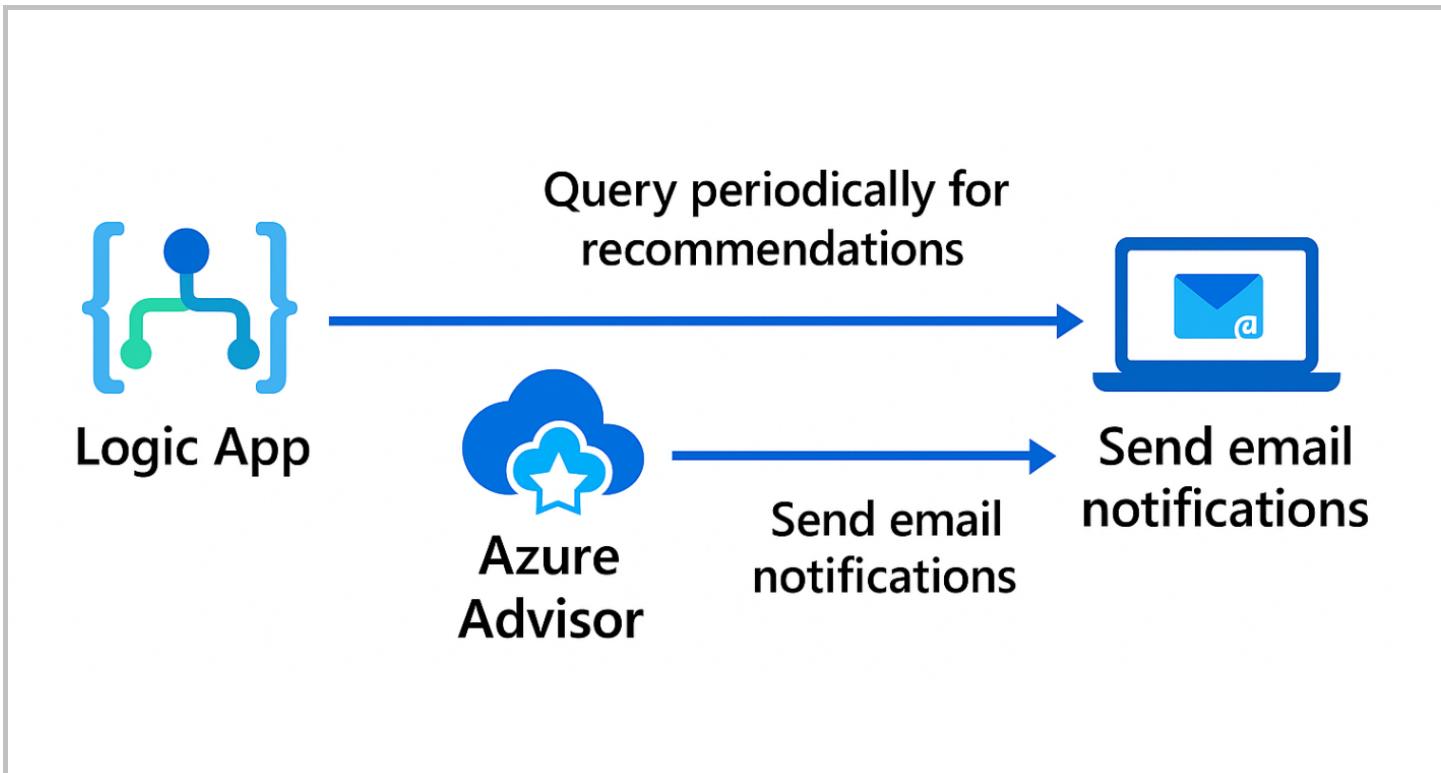
To ensure that Azure Advisor cost recommendations are not overlooked, Fabrikam wants an automated process to notify the IT team whenever new recommendations are available. What is the best solution?

- A. Use Azure Event Grid to trigger notifications for new cost recommendations, but only if they meet specific cost thresholds
- B. Set up an alert in Azure Monitor to track Advisor changes and notify the IT team
- C. Deploy a Logic App to query Azure Advisor periodically and send email notifications for new recommendations right
- D. Assign the Reader role to IT staff and instruct them to check Advisor recommendations manually

Explanation:

Correct Answer: C

Option C: Deploy a Logic App to query Azure Advisor periodically and send email notifications for new recommendations is correct because Azure Logic Apps provide a flexible platform to automate workflows by interacting with various Azure services, including Azure Advisor through its REST API. By configuring the Logic App to make periodic calls to the Azure Advisor API, Fabrikam can retrieve recommendations programmatically. The workflow can parse the API response to identify new recommendations and use connectors like email or Teams to notify the IT team. This approach ensures a fully automated process that eliminates manual effort and ensures recommendations are not overlooked. Although Azure Advisor lacks a direct Logic Apps connector, the HTTP connector in Logic Apps enables seamless integration with the REST API. This solution aligns with Fabrikam's requirement for automation, scalability, and proactive notifications, ensuring cost recommendations are addressed promptly.



Option A: Use Azure Event Grid to trigger notifications for new cost recommendations, but only if they meet specific cost thresholds is incorrect because Azure Event Grid is designed to handle event-driven architectures by delivering notifications for specific Azure services and custom events. However, Azure Event Grid does not natively support Azure Advisor as an event source. This means it cannot directly trigger notifications for new cost recommendations from Azure Advisor. Additionally, while Event Grid can filter events based on specific thresholds or criteria, the lack of direct integration with Azure Advisor makes it unsuitable for this requirement. Therefore, relying on Event Grid would not fulfill Fabrikam's need for an automated and reliable process to notify the IT team about new Advisor recommendations.

Option B: Set up an alert in Azure Monitor to track Advisor changes and notify the IT team is incorrect because Azure Monitor does not provide built-in capabilities to track changes in Azure Advisor recommendations. Azure Monitor focuses on collecting and analyzing telemetry data and metrics from Azure resources, but it lacks integration with Advisor for monitoring updates. While Azure Monitor can generate alerts for performance or log-based conditions, it cannot directly monitor Advisor's cost recommendations, making this approach infeasible for Fabrikam's requirements.

Option D: Assign the Reader role to IT staff and instruct them to check Advisor recommendations manually is incorrect because it introduces manual effort and is prone to errors. Assigning the Reader role to the IT team enables them to view Azure Advisor recommendations, but relying on team members to check for updates is neither efficient nor scalable. This approach does not align with Fabrikam's requirement for automation and proactive notifications, making it a suboptimal choice.

References:

<https://learn.microsoft.com/en-us/rest/api/advisor/recommendations/list?view=rest-advisor-2023-01-01&tabs=HTTP>

<https://learn.microsoft.com/en-us/azure/logic-apps/logic-apps-workflow-actions-triggers#http-request-actions>

Ask our Experts

Did you like this Question?



Question 5

Incorrect

Domain: Manage Azure identities and governance

[View Case Study](#)

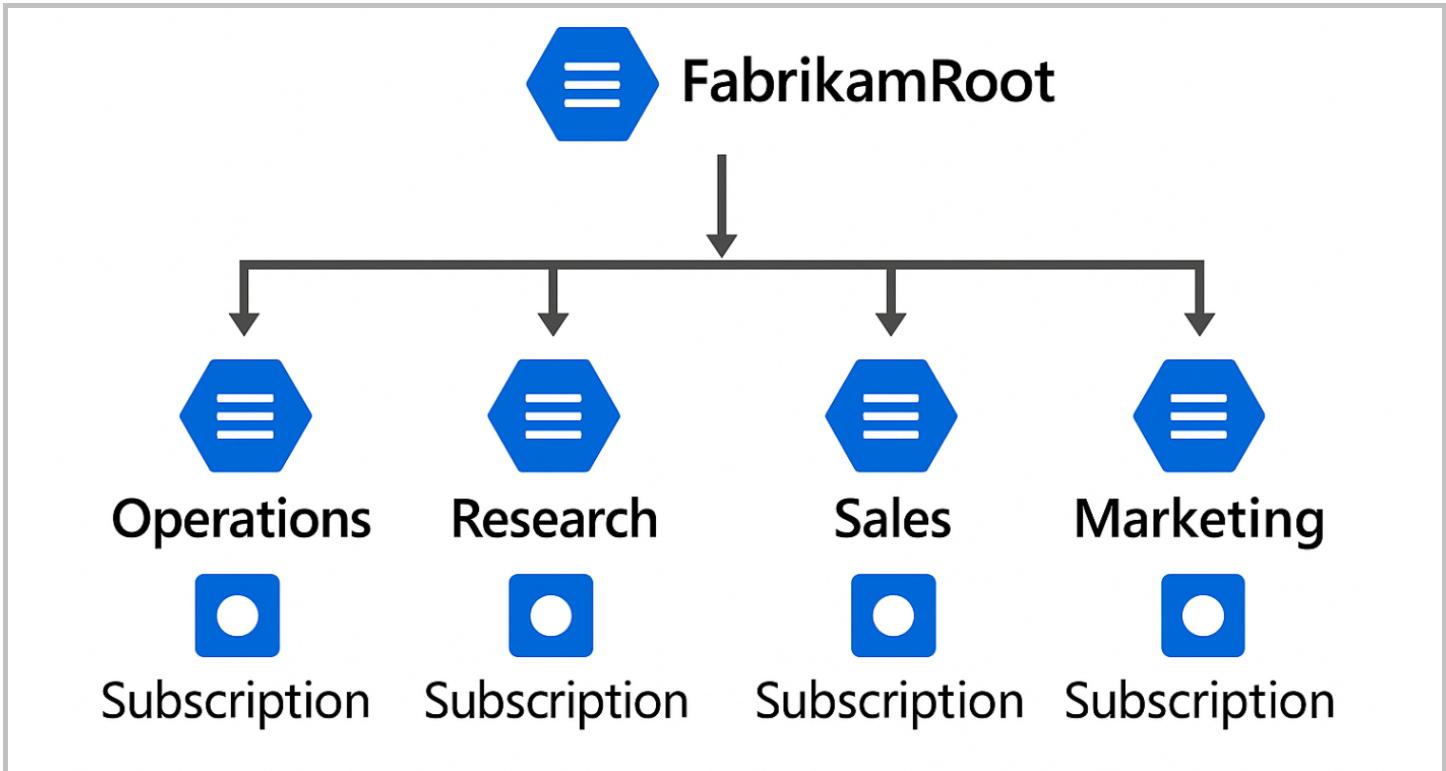
Fabrikam plans to create a multi-tier hierarchy where each subscription (Operations, Research, Sales, and Marketing) is placed under a dedicated management group. These management groups will roll up under the root FabrikamRoot group. How should the IT team proceed?

- A. Create child management groups under FabrikamRoot for each subscription and move subscriptions accordingly right
- B. Assign all subscriptions directly under FabrikamRoot and manage policies at the root level
- C. Consolidate subscriptions into a single group named BusinessUnits under the root group
- D. Use tags to distinguish subscriptions within the existing single management group structure wrong

Explanation:

Correct Answer: A

Option A: Create child management groups under FabrikamRoot for each subscription and move subscriptions accordingly is correct because it aligns with Azure's best practices for organizing resources using management groups. In this case, Fabrikam wants to create a multi-tier hierarchy, where each subscription (Operations, Research, Sales, and Marketing) is placed under its management group, and these management groups will roll up under the root group, FabrikamRoot. Azure management groups support this structure by enabling a hierarchical model where each management group can contain subscriptions, policies, access control, and compliance requirements that can be applied at the management group level. This solution ensures proper delegation, centralized management, and the ability to apply policies at different levels within the organization. It also maintains scalability, allowing the organization to expand by adding more management groups as needed.



Option B: Assign all subscriptions directly under FabrikamRoot and manage policies at the root level is incorrect because while you can assign subscriptions directly under the root management group (FabrikamRoot), it does not meet the requirement of creating a multi-tier hierarchy where each subscription has its dedicated management group. If all subscriptions are placed directly under the root group, it would make policy management and resource organization more difficult as there would be no granular control for the individual business units (Operations, Research, Sales, Marketing). Additionally, having everything under the root group limits flexibility in organizing and managing access, policies, and compliance separately for each department or subscription.

Option C: Consolidate subscriptions into a single group named BusinessUnits under the root group is incorrect because while it creates a management group under the root, it consolidates all subscriptions under a single management group (BusinessUnits). This approach does not adhere to the requirement of creating a multi-tier hierarchy with each department (Operations, Research, Sales, Marketing) having its dedicated management group. Consolidating subscriptions under a single management group would limit the ability to apply specific policies and access control tailored to each department, leading to potential conflicts in policy application or difficulty in tracking compliance at a granular level.

Option D: Use tags to distinguish subscriptions within the existing single management group structure is incorrect because tags are not a suitable tool for managing and organizing subscriptions at the level described in the question. Tags can be useful for resource-level organizations (such as VMs or storage accounts) but are not designed to manage subscriptions or implement a hierarchical structure for management groups. Tags do not provide the same level of governance, control, or policy management capabilities as management groups. While tags can help categorize resources, they cannot be used to manage subscriptions or delegate control at a higher level, making them inappropriate for this use case.

References:

<https://learn.microsoft.com/en-us/azure/governance/management-groups/overview>

<https://learn.microsoft.com/en-us/azure/governance/management-groups/manage#move-management-groups-and-subscriptions>

[Ask our Experts](#)

Did you like this Question?

[Finish Review](#)[Hands-on Labs](#)[Sandbox](#)[Subscription](#)[For Business](#)[Library](#)

Categories

Cloud Computing Certifications
Amazon Web Services (AWS)
Microsoft Azure
Google Cloud
DevOps
Cyber Security
Microsoft Power Platform
Microsoft 365 Certifications
Java Certifications

Popular Courses

AWS Certified Solutions Architect Associate
AWS Certified Cloud Practitioner
Microsoft Azure Exam AZ-204 Certification
Microsoft Azure Exam AZ-900 Certification
Google Cloud Certified Associate Cloud Engineer
Microsoft Power Platform Fundamentals (PL-900)
HashiCorp Certified Terraform Associate Certification
Snowflake SnowPro Core Certification
Docker Certified Associate

Company

About Us
Blog
Reviews
Careers
Team Account

Legal

Privacy Policy
Terms of Use
EULA
Refund Policy
Programs Guarantee

Support

Contact Us
FAQs

Need help? Please or +91 6364678444



©2025, Whizlabs Software Pvt. Ltd. All rights reserved.