Microsoft Microsoft AZ-400 PDF

Microsoft Microsoft AZ-400 PDF Questions Available Here at:

https://www.certification-questions.com//microsoft-dumps/az-400.html

Enrolling now you will get access to 562 questions in a unique set of Microsoft AZ-400

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.

Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the permissions of package feeds
• Visual Studio App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• By default, all App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• Code quality and release quality are critical. During release, deployments must not proceed between stages if any active bugs are logged against the release.

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmanme
  -ConfigurationMode 'ApplyOnly'
```

over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

Question 1

To resolve the current technical issue, what should you do to theRegister-AzureRmAutomationDscNodecommand?

Options:

A. Change the value of theConfigurationModeparameter.

B. Replace theRegister-AzureRmAutomationDscNodecmdlet withRegister-AzureRmAutomationScheduledRunbook

C. Add theAllowModuleOverwriteparameter.

D. Add theDefaultProfileparameter.

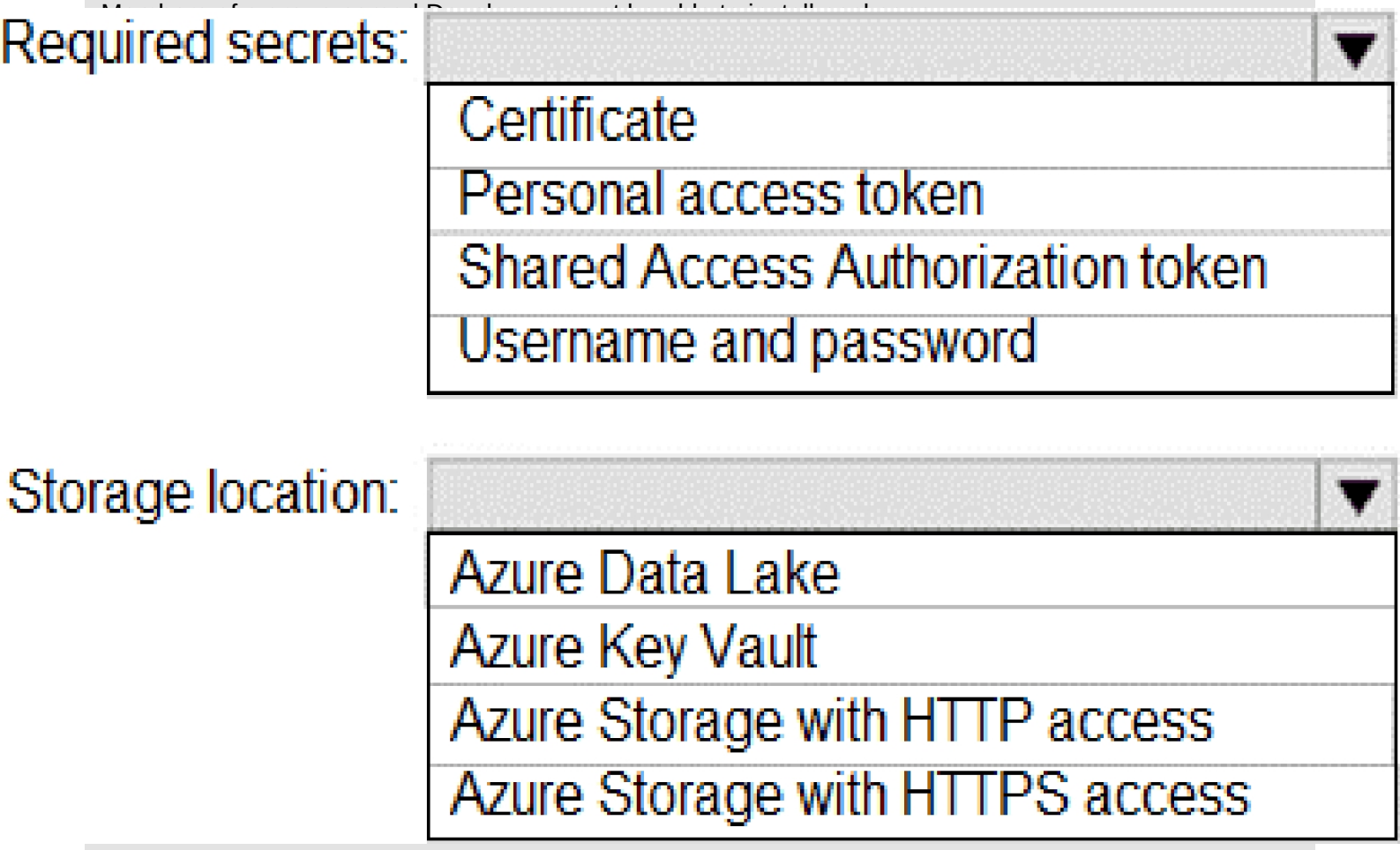
Answer: A

Explanation:
Change the ConfigurationMode parameter from ApplyOnly to ApplyAndAutocorrect.
The Register-AzureRmAutomationDscNode cmdlet registers an Azure virtual machine as an APS Desired State Configuration (DSC) node in an Azure Automation account.
Scenario: Current Technical Issue
The test servers are configured correctly when first deployed, but they experience configuration drift over time. Azure AutomationState Configuration fails to correct the configurations.
Azure Automation State Configuration nodes are registered by using the following command.

Reference:https://docs.microsoft.com/en-us/powershell/module/azurerm.automation/register-azurermautomationdscnode?view=azurermps-6.13.0

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS

mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.



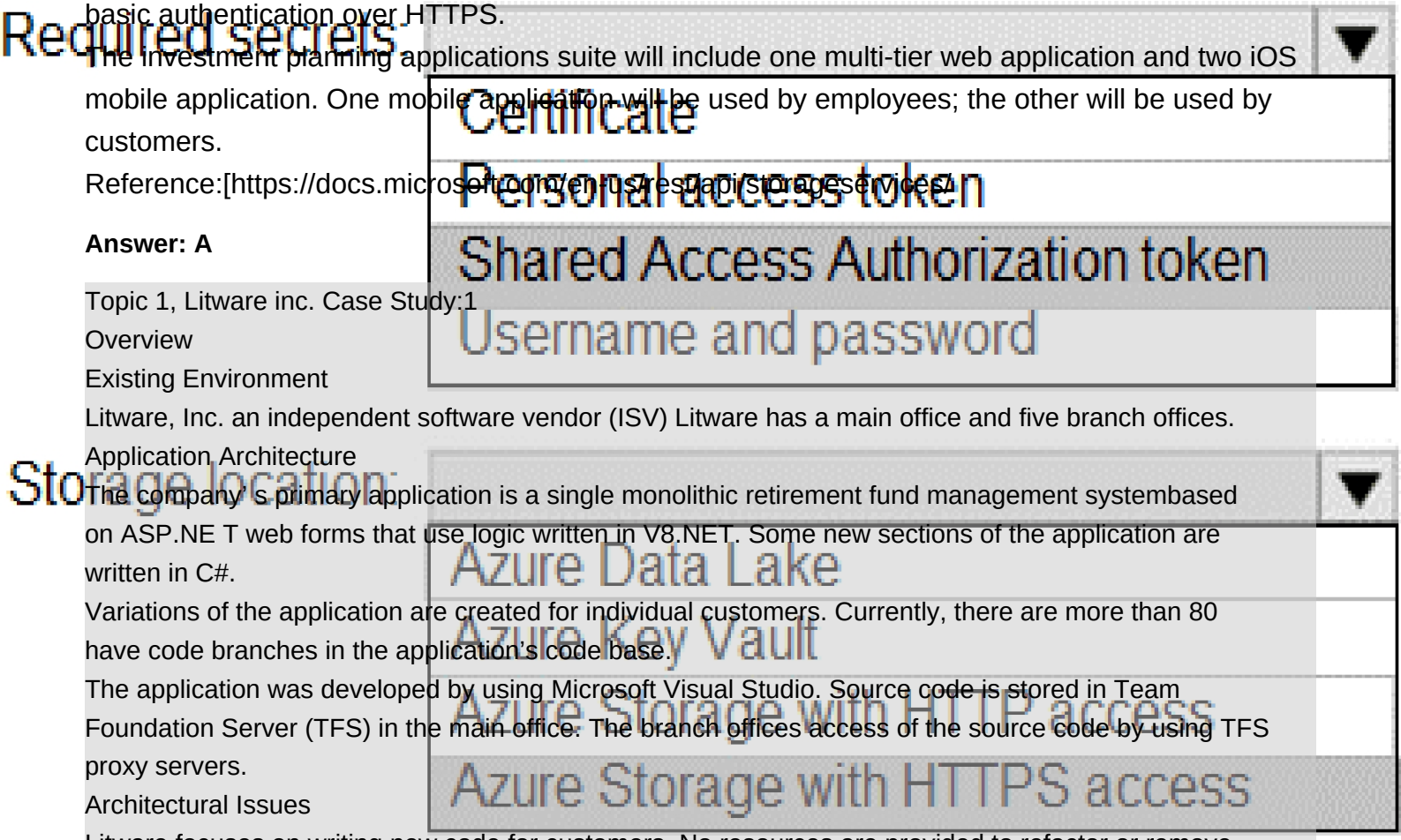
Azure Automation State Configuration nodes are registered by using the following command.

Question 2

How should you confrere the release retention policy for the investment planning depletions suite?
To answer, select the appropriate options in the answer area.
NOTE: Eachcorrect selection is worth one point.

Options:

A.



Every request made against a storage service must be authorized, unless the request is for a blob or
container resource that has been made available for public or signed access. One option for authorizing a request is by using Shared Key.
Scenario: The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HTTPS.
The investment planning applications suite will include one multi-tier web application and two iOS mobile application. One mobile application will be used by employees; the other will be used by customers.
Reference:https://docs.microsoft.com/en-us/rest/api/storageservices/

Answer: A

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment

planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmanme
  -ConfigurationMode 'ApplyOnly'
```

fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.
• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

Question 3

You need to configure a cloud service to store the secrets required by the mobile applications to call

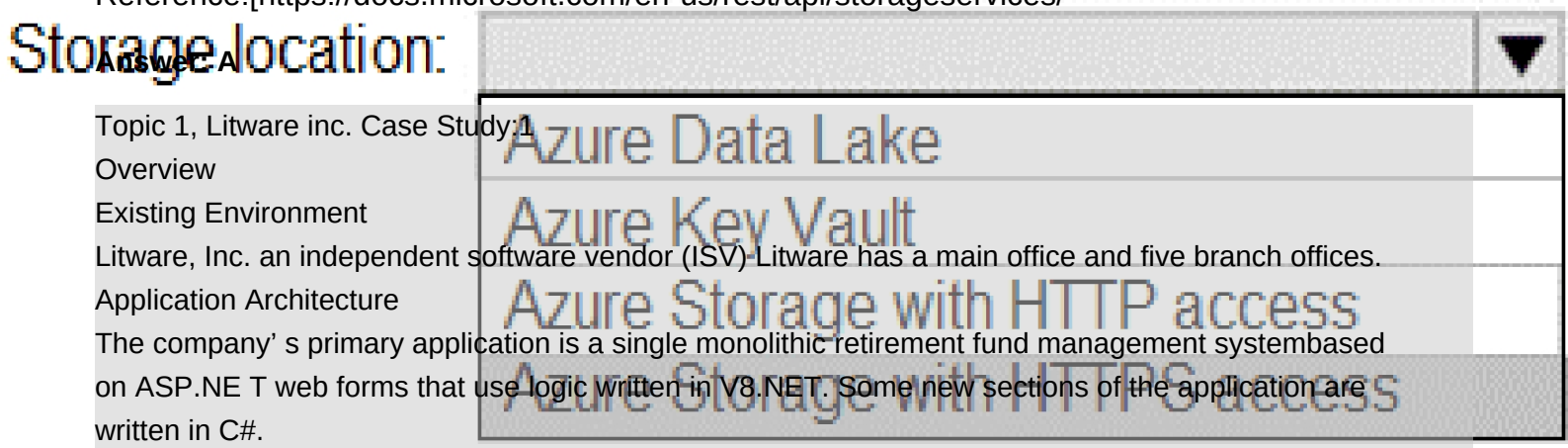
the share.
What should you include in thesolution? To answer, select the appropriate options m the answer

area, NOTE: Each correct selection is worth one point.

Options:

A.

Every request made against a storage service must be authorized, unless the request is for a blob or
container resource that has been made available for public or signed access. One option for authorizing a request is by using Shared Key.
Scenario: The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HTTPS.
The investment planning applications suite will include one multi-tier web application and two iOS mobile application. One mobile application will be used by employees, the other will be used by customers.
Reference:[https://docs.microsoft.com/en-us/rest/api/storageservices/



Answer: A

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging

frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the permissions of package feeds
• Visual Studio App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• By default, all App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• Code quality and release quality are critical. During release, deployments must not proceed between stages if any active bugs are logged against the release.
• The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.
• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

Question 4

What should you use to implementthe code quality restriction on the release pipeline for the investment planning applications suite?

Options:

A. a trigger

B. a pre deployment approval

C. a post-deployment approval

D. a deployment gate

Answer: D

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NET web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS

mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the permissions of package feeds
• Visual Studio App Center must be used to centralize the reporting of mobile application crashes and device types in use.

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmanme
  -ConfigurationMode 'ApplyOnly'
```

Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

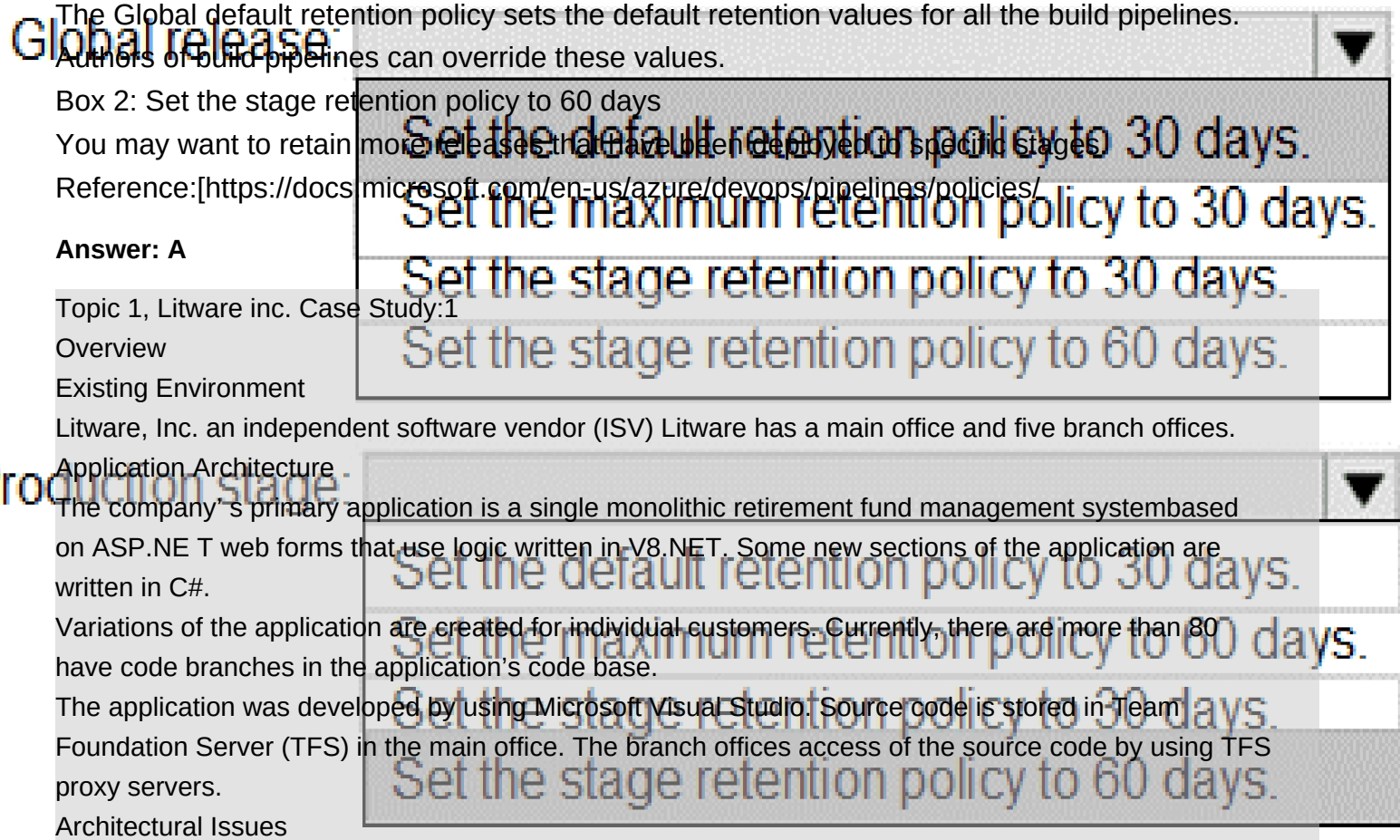
Question 5

How should youconfigure the release retention policy for the investment planning applications suite?
To answer, select the appropriate options in the answer area.

NOTE:Each correct selection is worth one point.

Options:

A.



Scenario: By default, all releases must remain available for 30 days, except for production releases,
which must be kept for 60 days.
Box 1: Set the defaultretention policy to 30 days
The Global default retention policy sets the default retention values for all the build pipelines. Authors of build pipelines can override these values.
Box 2: Set the stage retention policy to 60 days
You may want to retain more releases that have been deployed to specific stages.
Reference:[https://docs.microsoft.com/en-us/azure/devops/pipelines/policies/

Answer: A

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company's primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment

planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmanme
  -ConfigurationMode 'ApplyOnly'
```

fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.
• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

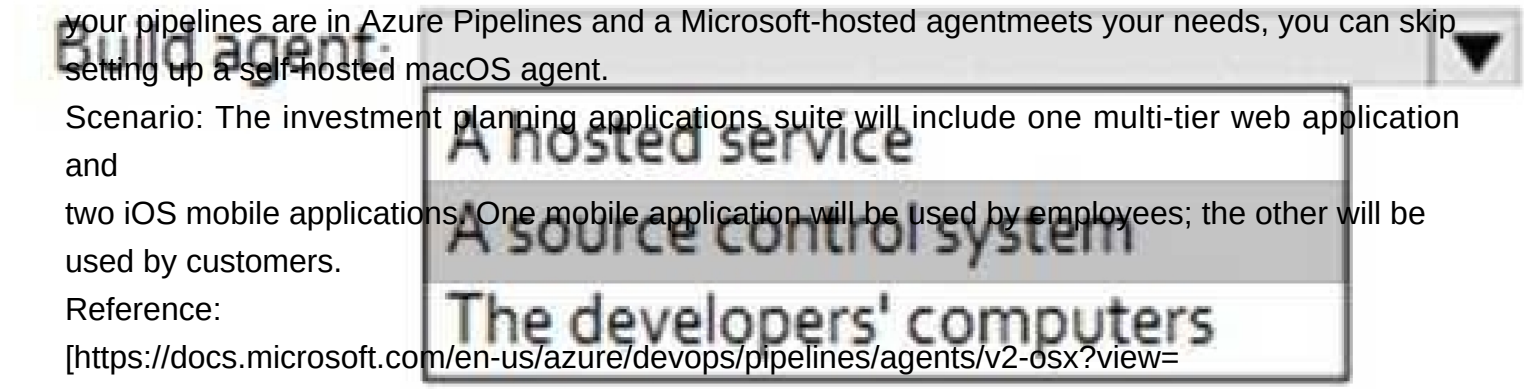
Question 6

Where should the build and release agents for the investment planning application suite run? To

answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Options:

A.



Box 1: A source control system
A source control system, also called a version control system, allows developers to collaborate on code and track changes.Source control is an essential tool for multi-developer projects.
Box 2: A hosted service
To build and deploy Xcode apps or Xamarin.iOS projects, you'll need at least one macOS agent. If your pipelines are in Azure Pipelines and a Microsoft-hosted agentmeets your needs, you can skip setting up a self-hosted macOS agent.
Scenario: The investment planning applications suite will include one multi-tier web application and
two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Reference:
[https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-osx?view=

Answer: A

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.

Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmanme
  -ConfigurationMode 'ApplyOnly'
```

• By default, all App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• Code quality and release quality are critical. During release, deployments must not proceed between stages if any active bugs are logged against the release.
• The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.
• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

Question 7

Which branching strategy should you recommend for the investment planning applications suite?

Options:

A. release isolation

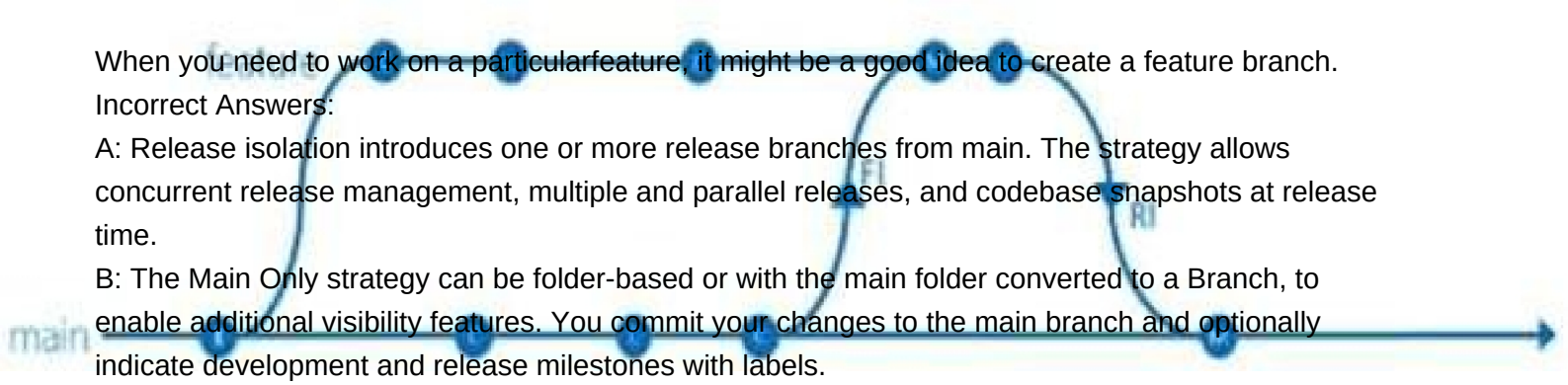
B. main only

C. development isolation

D. feature isolation

Answer: D

Explanation:
Scenario: A branching strategy that supports developing newfunctionality in isolation must be used.
Feature isolation is a special derivation of the development isolation, allowing you to branch one or more feature branches from main, as shown, or from your dev branches.



When you need to work on a particularfeature, it might be a good idea to create a feature branch.
Incorrect Answers:
A: Release isolation introduces one or more release branches from main. The strategy allows concurrent release management, multiple and parallel releases, and codebase snapshots at release time.
B: The Main Only strategy can be folder-based or with the main folder converted to a Branch, to enable additional visibility features. You commit your changes to the main branch and optionally indicate development and release milestones with labels.
C: Development isolation: When you need to maintain and protect a stable main branch, you can branch one or more dev branches from main. It enables isolation and concurrent development. Work can be isolated in development branches by feature, organization, or temporary collaboration.
Reference:
https://docs.microsoft.com/en-us/azure/devops/repos/tfvc/branching-strategies-with-tfvc?view=azure-devops

Topic 1, Litware inc. Case Study:1
Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80

have code branches in the application’s code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the permissions of package feeds
• Visual Studio App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• By default, all App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• Code quality and release quality are critical. During release, deployments must not proceed between stages if any active bugs are logged against the release.
• The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.

• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

Permission	Reader	Collaborator	Contributor	Owner
List and restore/install packages	✓	✓	✓	✓
Save packages from upstream sources		✓	✓	✓
Push packages			✓	✓
Unlist/deprecate packages			✓	✓
Delete/unpublish package				✓
Edit feed permissions				✓
Rename and delete feed				✓

any type of identity-individuals, teams, and groups-to any access level.
Box 2: Owner
Members of a group named Team Leaders must be able to create new packages and edit the

permissions of package feeds.

Answer: A

Topic 1, Litware inc. Case Study:1

Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the permissions of package feeds
• Visual Studio App Center must be used to centralize the reporting of mobile application crashes

and device types in use.
• By default, all App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• Code quality and release quality are critical. During release, deployments must not proceed between stages if any active bugs are logged against the release.
• The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.
• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmname
  -ConfigurationMode 'ApplyOnly'
```

Question 9

You are using GitHub asa source code repository.
You create a client-side Git hook on the commit-msg event. The hook requires that each commit message contain a custom work item tag.
You need to make a commit that does not have a work item tag.
Which git commit parameter should you use?

Options:

A. --squash

B. --no-verify

C. --message ''

D. --no-post-rewrite

Answer: B

Explanation:
The commit-msg hook is invoked by git-commit and git-merge, and can be bypassed with the --no-verify option.
Reference:
https://git-scm.com/docs/githooks

Topic 1, Litware inc. Case Study:1

Overview
Existing Environment
Litware, Inc. an independent software vendor (ISV) Litware has a main office and five branch offices.
Application Architecture
The company' s primary application is a single monolithic retirement fund management systembased on ASP.NE T web forms that use logic written in V8.NET. Some new sections of the application are written in C#.
Variations of the application are created for individual customers. Currently, there are more than 80 have code branches in the application's code base.
The application was developed by using Microsoft Visual Studio. Source code is stored in Team Foundation Server (TFS) in the main office. The branch offices access of the source code by using TFS proxy servers.
Architectural Issues
Litware focuses on writing new code for customers. No resources are provided to refactor or remove existing code. Changes to the code base take a long time, AS dependencies are not obvious to individual developers.
Merge operations of the code often take months andinvolve many developers. Code merging frequently introduces bugs that are difficult to locate and resolve.
Customers report that ownership costs of the retirement fund management system increase continually. The need to merge unrelated code makes even minor code changes expensive.
Requirements
Planned Changes
Litware plans to develop a new suite of applications for investment planning. The investment planning Applications will require only minor integration with the easting retirement fund management system.
The investment planning applications suite will include one multi-tier web application and two iOS mobile applications. One mobile application will be used by employees; the other will be used by customers.
Litware plans to move to a more agile development methodology. Shared code will be extracted into a series of package.
Litware has started an internal cloud transformation process and plans to use cloud based services whenever suitable.
Litware wants to become proactive m detecting failures, rather than always waning for customer bug reports.
Technical Requirements
The company's investment planning applications suite must meet the following technical requirements:
• New incoming connections through the firewall must be minimized.
• Members of a group named Developers must be able to install packages.
• The principle of least privilege must be used for all permission assignments
• A branching strategy that supports developing new functionality in isolation must be used.
• Members of a group named Team leaders must be able to create new packages and edit the permissions of package feeds
• Visual Studio App Center must be used to centralize the reporting of mobile application crashes

and device types in use.
• By default, all App Center must be used to centralize the reporting of mobile application crashes and device types in use.
• Code quality and release quality are critical. During release, deployments must not proceed between stages if any active bugs are logged against the release.
• The mobile applications must be able to call the share pricing service of the existing retirement fund management system. Until the system is upgraded, the service will only support basic authentication over HUPS.
• The required operating system configuration tor the test servers changes weekly. Azure Automation State Configuration must be used to ensure that the operating system on each test servers configured the same way when the servers are created and checked periodically.
Current Technical
The test servers areconfigured correctly when first deployed, but they experience configuration drift over time. Azure Automation State Configuration fails to correct the configurations.

Azure Automation State Configuration nodes are registered by using the following command.

```
Register-AzureRmAutomationDscNode
  -ResourceGroupName 'TestResourceGroup'
  -AutomationAccountName 'LitwareAutomationAccount'
  -AzureVMName $vmarnme
  -ConfigurationMode 'ApplyOnly'
```

Question 10

You have Azure Pipelines and GitHub integrated as a source code repository.
The build pipeline has continuous integration enabled.
You plan to trigger an automated build whenever code changes are committed to the repository.
You need to ensure that the system will wait until a build completes before queuing another build.
What should you implement?

Options:

A. path filters

B. batch changes

C. scheduled builds

D. branch filters

Answer: B

Explanation:
Batching CI runs
If you have many team members uploadingchanges often, you may want to reduce the number of runs you start. If you set batch to true, when a pipeline is running, the system waits until the run is completed, then starts another run with all changes that have not yet been built.
Example:
specific branch build with batching

trigger:
batch: true
branches:
include:
- master

To clarify this example, let us say that a push A to master caused the above pipeline to run. While that pipeline is running, additional pushes B and C occur into therepository. These updates do not start new independent runs immediately. But after the first run is completed, all pushes until that point of time are batched together and a new run is started.

Reference:
https://docs.microsoft.com/en-us/azure/devops/pipelines/repos/github