



Designing and Implementing Microsoft DevOps Solutions



EXAMKILLER

Help Pass Your Exam At First Try

Microsoft

Exam AZ-400

Microsoft Azure DevOps Solutions

Version: 37.1

[Total Questions: 424]

Topic break down

Topic	No. of Questions
Topic 1: Litware inc. Case Study: 1	14
Topic 2: Contoso Case Study: 2	10
Topic 3: Woodgrove bank	10
Topic 4: Mix Questions Set	390

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 system based on ASP.NET web forms that use logic written in VB.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 and involve 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 existing 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 in 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 for 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 are configured 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 No : 1 HOTSPOT - (Topic 1)

HOTSPOT

You have an Azure virtual machine named VM1 that runs Linux.

You plan to deploy the Desired State Configuration (DSC) extension to VM1.

You need to grant the Log Analytics agent the appropriate directory permissions.

How should you complete the command? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

setfacl -m u:omsagent:

r	/lib
x	/etc
rx	/tmp
rwx	/usr

Answer:

setfacl -m u:omsagent:

r	/lib
x	/etc
rx	/tmp
rwx	/usr

setfacl -m u:omsagent:

r	/lib
x	/etc
rx	/tmp
rwx	/usr

Explanation:

Box 1: rwx

The Log Analytics agent for Linux runs as the omsagent user. To grant >write permission to the omsagent user, run the command setfacl -m u:omsagent:rwx /tmp.

Box 2: /tmp

Deploying DSC to a Linux node uses the /tmp folder.

Question No : 2 HOTSPOT - (Topic 1)

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

Required secrets:

▼
Certificate
Personal access token
Shared Access Authorization token
Username and password

Storage location:

▼
Azure Data Lake
Azure Key Vault
Azure Storage with HTTP access
Azure Storage with HTTPS access

Answer:

Required secrets:

▼
Certificate
Personal access token
Shared Access Authorization token
Username and password

Storage location:

▼
Azure Data Lake
Azure Key Vault
Azure Storage with HTTP access
Azure Storage with HTTPS access

Required secrets:

Certificate
Personal access token
Shared Access Authorization token
Username and password

Storage location:

Azure Data Lake
Azure Key Vault
Azure Storage with HTTP access
Azure Storage with HTTPS access

Explanation:

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.

References: <https://docs.microsoft.com/en-us/rest/api/storageservices/authorize-with-shared-key>

Question No : 3 - (Topic 1)

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

- A. a trigger
- B. a pre deployment approval
- C. a post-deployment approval

D. a deployment gate

Answer: B

Question No : 4 HOTSPOT - (Topic 1)

HOTSPOT

You are using PowerShell to administer Azure Log Analytics workspaces.

You need to list the available workspaces and their properties.

How should you complete the command? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer:

Explanation:

Box 1: Get-AzResource

Use the following command to examine the access control mode for all workspaces in the subscription:

PowerShell

Get-Az Resource –Resource Type Microsoft. Operational Insights/workspaces –Expand Properties | for each {s.Name + " : " +

\$_ .Properties.features.enableLogAccessUsingOnlyResourcePermissions

Box 2: -Resource Type

Question No : 5 - (Topic 1)

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?

- A. path filters
- B. batch changes
- C. scheduled builds
- D. branch filters

Answer: B

Explanation:

Batching CI runs

If you have many team members uploading changes 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 the repository. 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>

Question No : 6 HOTSPOT - (Topic 1)

How should you complete the code to initialize App Center in the mobile application? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection a worth one point.

```
MSAppCenter.start
( "{Your App Secret}",
  withServices:
)
```

Answer:

```
MSAppCenter.start
( "{Your App Secret}",
  withServices:
)
```

```
MSAppCenter.start
( "{Your App Secret}",
  withServices:
)
```

Explanation:

Scenario: Visual Studio App Center must be used to centralize the reporting of mobile application crashes and device types in use.

In order to use App Center, you need to opt in to the service(s) that you want to use, meaning by default no services are started and you will have to explicitly call each of them when starting the SDK.

Insert the following line to start the SDK in your app's App Delegate class in the didFinishLaunchingWithOptions method.

```
MSAppCenter.start("{Your App Secret}", withServices: [MSAnalytics.self, MSCrashes.self])
```

References: <https://docs.microsoft.com/en-us/appcenter/sdk/getting-started/ios>

Question No : 7 HOTSPOT - (Topic 1)

How should you configure 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.

Global release:

- Set the default retention policy to 30 days.
- Set the maximum retention policy to 30 days.
- Set the stage retention policy to 30 days.
- Set the stage retention policy to 60 days.

Production stage:

- Set the default retention policy to 30 days.
- Set the maximum retention policy to 60 days.
- Set the stage retention policy to 30 days.
- Set the stage retention policy to 60 days.

Answer:

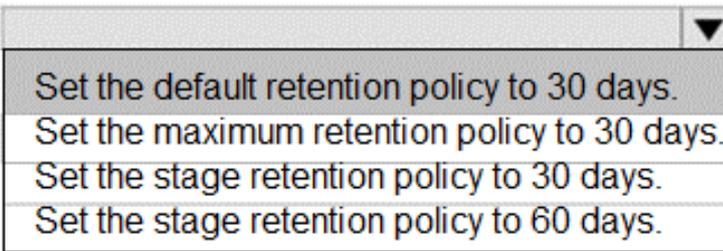
Global release:

- Set the default retention policy to 30 days.
- Set the maximum retention policy to 30 days.
- Set the stage retention policy to 30 days.
- Set the stage retention policy to 60 days.

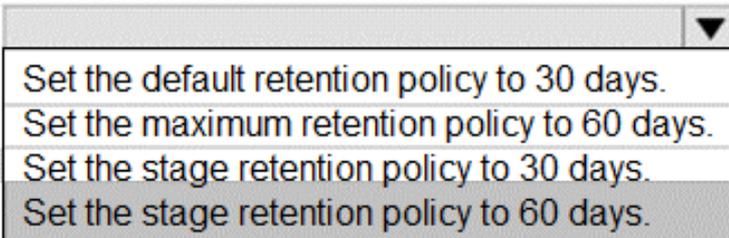
Production stage:

- Set the default retention policy to 30 days.
- Set the maximum retention policy to 60 days.
- Set the stage retention policy to 30 days.
- Set the stage retention policy to 60 days.

Explanation:

Global release: 

- Set the default retention policy to 30 days.
- Set the maximum retention policy to 30 days.
- Set the stage retention policy to 30 days.
- Set the stage retention policy to 60 days.

Production stage: 

- Set the default retention policy to 30 days.
- Set the maximum retention policy to 60 days.
- Set the stage retention policy to 30 days.
- Set the stage retention policy to 60 days.

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 default retention 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.

References: <https://docs.microsoft.com/en-us/azure/devops/pipelines/policies/retention>

Question No : 8 - (Topic 1)

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

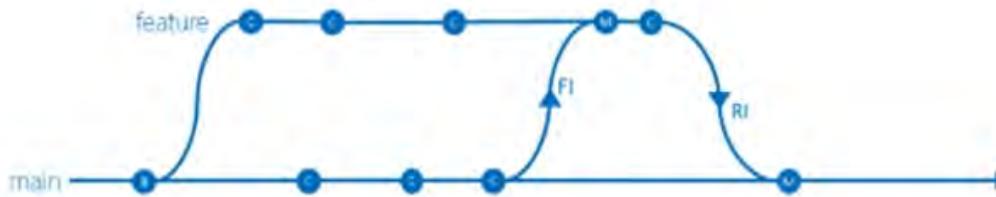
- A. release isolation
- B. main only
- C. development isolation
- D. feature isolation

Answer: D

Explanation:

Scenario: A branching strategy that supports developing new functionality 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 particular feature, it might be a good idea to create a feature branch.

Question No : 9 - (Topic 1)

To resolve the current technical issue, what should you do to the Register-AzureRmAutomationDscNode command?

- A. Change the value of the ConfigurationMode parameter.
- B. Replace the Register-AzureRmAutomationDscNode cmdlet with Register-AzureRmAutomationScheduledRunbook
- C. Add the AllowModuleOverwrite parameter.
- D. Add the DefaultProfile parameter.

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 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'
```

References: <https://docs.microsoft.com/en-us/powershell/module/azurermsautomation/register-azurermsautomationdscnode?view=azurermps-6.13.0>

Question No : 10 DRAG DROP - (Topic 1)

Which package feed access levels should be assigned to the Developers and Team Leaders groups for the investment planning applications suite? To answer, drag the appropriate access levels to the correct groups. Each access level may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Access Levels

- Collaborator
- Contributor
- Owner
- Reader

Answer Area

Developers:

Team Leaders:

Answer:

Access Levels

- Collaborator
- Contributor
- Owner
- Reader

Answer Area

Developers:

Team Leaders:

Explanation:

Developers:

Team Leaders:

Box 1: Reader

Members of a group named Developers must be able to install packages.

Feeds have four levels of access: Owners, Contributors, Collaborators, and Readers. Owners can add 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.

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				✓

Question No : 11 - (Topic 1)

You have a project in Azure DevOps.

You plan to deploy a self-hosted agent by using an unattended configuration script.

Which two values should you define in the configuration script? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. authorization credentials
- B. the project name
- C. the deployment group name
- D. the organization URL
- E. the agent pool name

Answer: A,D

Explanation:

Unattended config:

The agent can be set up from a script with no human intervention. You must pass --unattended and the answers to all questions.

To configure an agent, it must know the URL to your organization or collection and credentials of someone authorized to set up agents. All other responses are optional.
Reference:

<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-windows>

Question No : 12 - (Topic 1)

You are using GitHub as a 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?

- 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>

Question No : 13 HOTSPOT - (Topic 1)

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.

Build agent: ▼

- A hosted service
- A source control system
- The developers' computers

Release agent: ▼

- A hosted service
- A source control system
- The developers' computers

Answer:

Build agent: ▼

- A hosted service
- A source control system
- The developers' computers

Release agent: ▼

- A hosted service
- A source control system
- The developers' computers

Explanation:

Build agent: ▼

- A hosted service
- A source control system
- The developers' computers

Release agent: ▼

- A hosted service
- A source control system
- The developers' computers

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 agent meets 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.

References:

<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-osx?view=azure-devops>

Question No : 14 HOTSPOT - (Topic 1)

How should you configure 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.

Required secrets:

▼
Certificate
Personal access token
Shared Access Authorization token
Username and password

Storage location:

▼
Azure Data Lake
Azure Key Vault
Azure Storage with HTTP access
Azure Storage with HTTPS access

Answer:

Required secrets:

▼
Certificate
Personal access token
Shared Access Authorization token
Username and password

Storage location:

▼
Azure Data Lake
Azure Key Vault
Azure Storage with HTTP access
Azure Storage with HTTPS access

Explanation:

Required secrets: 

Certificate
Personal access token
Shared Access Authorization token
Username and password

Storage location: 

Azure Data Lake
Azure Key Vault
Azure Storage with HTTP access
Azure Storage with HTTPS access

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.

References: <https://docs.microsoft.com/en-us/rest/api/storageservices/authorize-with-shared-key>

Topic 2, Contoso Case Study: 2

Overview

Existing Environment

Contoso, Ltd. is a manufacturing company that has a main office in Chicago.

Requirements

Contoso plans to improve its IT development and operations processes implementing Azure

DevOps principles. Contoso has an Azure subscription and creates an Azure DevOps organization.

The Azure DevOps organization includes:

- ✍ The Docker extension
- ✍ A deployment pool named Pool7 that contains 10 Azure virtual machines that run Windows Server 2016.

The Azure subscription contains an Azure Automation account.

Planned Changes

Contoso plans to create projects in Azure DevOps as shown in the following table.

Project name	Project details
Project 1	Project1 will provide support for incremental builds and third-party SDK components
Project 2	Project2 will use an automatic build policy. A small team of developers named Team2 will work independently on changes to the project. The Team2 members will not have permissions to Project2.
Project 3	Project3 will be integrated with SonarQube
Project 4	Project4 will provide support for a build pipeline that creates a Docker image and pushes the image to the Azure Container Registry. Project4 will use an existing Dockerfile.
Project 5	Project5 will contain a Git repository in Azure Reports and a continuous integration trigger that will initiate a build in response to any change except for changes within /folder1 of the repository.
Project 6	Project6 will provide support for build and deployment pipelines. Deployment will be allowed only if the number of current work items representing active software bugs is 0.
Project 7	Project7 will contain a target deployment group named Group7 that maps to Pool7. Project7 will use Azure Automation State Configuration to maintain the desired state of the computers in Group7.

Technical Requirements

Contoso identifies the following technical requirements:

- Implement build agents for Project 1.
- Whenever possible, use Azure resources
- Avoid using deprecated technologies
- Implement a code flow strategy for Project2 that will:
- Enable Team 2 to submit pull requests for Project2.
- Enable Team 2 to work independently on changes to a copy of Project2?
- Ensure that any intermediary changes performed by Team2 on a copy of Project2 will be subject to the same restrictions as the ones defined in the build policy of Project2.
- Whenever possible, implement automation and minimize administrative effort.
- Implement Project3, Project5, Project6, and Project7 based on the planned changes.
- Implement Project4 and configure the project to push Docker images to Azure Container Registry.

Question No : 15 DRAG DROP - (Topic 2)

You need to configure Azure Automation for the computer in Group7.

Which three 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.

Actions	Answer Area
Run the <code>Import-AzureRmAutomationDscConfiguration</code> Azure PowerShell cmdlet.	
Create a Desired State Configuration (DSC) configuration file that has an extension of <code>.ps1</code> .	
Run the <code>New-AzureRmResourceGroupDeployment</code> Azure PowerShell cmdlet.	⏪ ⏩
Run the <code>Start-AzureRmAutomationDscCompilationJob</code> Azure PowerShell cmdlet.	
Create an Azure Resource Manager template file that has an extension of <code>.json</code> .	

Answer:

Actions	Answer Area
Run the <code>Import-AzureRmAutomationDscConfiguration</code> Azure PowerShell cmdlet.	
Create a Desired State Configuration (DSC) configuration file that has an extension of <code>.ps1</code> .	⏪ ⏩
Run the <code>New-AzureRmResourceGroupDeployment</code> Azure PowerShell cmdlet.	⏪ ⏩
Run the <code>Start-AzureRmAutomationDscCompilationJob</code> Azure PowerShell cmdlet.	⏪ ⏩
Create an Azure Resource Manager template file that has an extension of <code>.json</code> .	

Create a Desired State Configuration (DSC) configuration file that has an extension of `.ps1`.

Run the `Import-AzureRmAutomationDscConfiguration` Azure PowerShell cmdlet.

Run the `Start-AzureRmAutomationDscCompilationJob` Azure PowerShell cmdlet.

Explanation:

Step 1: Create a Desired State Configuration (DSC) configuration file that has an extension of `.ps1`.

Step 2: Run the `Import-AzureRmAutomationDscConfiguration` Azure Powershell cmdlet
 The `Import-AzureRmAutomationDscConfiguration` cmdlet imports an APS Desired State Configuration (DSC) configuration into Azure Automation. Specify the path of an APS script that contains a single DSC configuration.

Example:

PS C:\>Import-AzureRmAutomationDscConfiguration -AutomationAccountName

```
"Contoso17"-ResourceGroupName "ResourceGroup01" -SourcePath "C:\DSC\client.ps1" -Force
```

This command imports the DSC configuration in the file named client.ps1 into the Automation account named Contoso17. The command specifies the Force parameter. If there is an existing DSC configuration, this command replaces it.

Step 3: Run the Start-AzureRmAutomationDscCompilationJob Azure Powershell cmdlet
The Start-AzureRmAutomationDscCompilationJob cmdlet compiles an APS Desired State Configuration (DSC) configuration in Azure Automation.

References:

<https://docs.microsoft.com/en-us/powershell/module/azurermsautomation/import-azurermsautomationdscconfiguration>

<https://docs.microsoft.com/en-us/powershell/module/azurermsautomation/start-azurermsautomationdsccompilationjob>

Question No : 16 DRAG DROP - (Topic 2)

You need to implement Project6.

Which three 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.

Actions

- Open the release pipeline editor.
- Open the **Triggers** tab.
- Disable the continuous integration trigger.
- Enable Gates.
- Add a manual intervention task.
- Add Query Work Items.

Answer Area

- 1
- 2
- 3

Answer:

Explanation:

- Open the release pipeline editor.
- Enable Gates.
- Add Query Work Items.

Scenario: Implement Project3, Project5, Project6, and Project7 based on the planned changes

Project 6	Project6 will provide support for build and deployment pipelines. Deployment will be allowed only if the number of current work items representing active software bugs is 0.
-----------	---

Step 1: Open the release pipeline editor.

In the Releases tab of Azure Pipelines, select your release pipeline and choose Edit to open the pipeline editor.

Step 2: Enable Gates.

Choose the pre-deployment conditions icon for the Production stage to open the conditions panel. Enable gates by using the switch control in the Gates section.

Step 3: Add Query Work items.

Choose + Add and select the Query Work Items gate.

Configure the gate by selecting an existing work item query.

Deployment gates ⓘ + Add ▾

Query Work Items Enabled 

Query Work Items ⓘ

Task version ▾

Display name *

Query * ⓘ ▾

Upper threshold * ⓘ

Advanced ^

Lower threshold * ⓘ

Output Variables ^

Reference name ⓘ

Variables list

There are no output variables associated with this task [more information](#) 

Evaluation options ▾

Note: A case for release gate is:

Incident and issues management. Ensure the required status for work items, incidents, and issues. For example, ensure deployment occurs only if no priority zero bugs exist, and validation that there are no active incidents takes place after deployment.

References:

<https://docs.microsoft.com/en-us/azure/devops/pipelines/release/deploy-using-approvals?view=azure-devops#configure-gate>

Question No : 17 DRAG DROP - (Topic 2)

You need to recommend a procedure to implement the build agent for Project1.

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

Actions	Answer Area
Sign in to Azure DevOps by using an account that is assigned the Administrator service connection security role.	
Install the Azure Pipelines agent on on-premises virtual machine.	
Create a personal access token in the Azure DevOps organization of Contoso.	
Install and register the Azure Pipelines agent on an Azure virtual machine.	
Sign in to Azure DevOps by using an account that is assigned the agent pool administrator role.	

Answer:

Actions	Answer Area
Sign in to Azure DevOps by using an account that is assigned the Administrator service connection security role.	Sign in to Azure DevOps by using an account that is assigned the Administrator service connection security role.
Install the Azure Pipelines agent on on-premises virtual machine.	
Create a personal access token in the Azure DevOps organization of Contoso.	Create a personal access token in the Azure DevOps organization of Contoso.
Install and register the Azure Pipelines agent on an Azure virtual machine.	
Sign in to Azure DevOps by using an account that is assigned the agent pool administrator role.	Install and register the Azure Pipelines agent on an Azure virtual machine.

Explanation:

Sign in to Azure DevOps by using an account that is assigned the Administrator service connection security role.

Create a personal access token in the Azure DevOps organization of Contoso.

Install and register the Azure Pipelines agent on an Azure virtual machine.

Scenario:

Project 1	Project1 will provide support for incremental builds and third-party SDK components
-----------	---

Step 1: Sign in to Azure Develops by using an account that is assigned the Administrator service connection security role.

Note: Under Agent Phase, click Deploy Service Fabric Application. Click Docker Settings and then click Configure Docker settings. In Registry Credentials Source, select Azure Resource Manager Service Connection. Then select your Azure subscription.

Step 2: Create a personal access token..

A personal access token or PAT is required so that a machine can join the pool created with the Agent Pools (read, manage) scope.

Step 3: Install and register the Azure Pipelines agent on an Azure virtual machine.

By running a Azure Pipeline agent in the cluster, we make it possible to test any service, regardless of type.

References:

<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-tutorial-deploy-container-app-with-cicd-vsts>

<https://mohitgoyal.co/2019/01/10/run-azure-devops-private-agents-in-kubernetes-clusters/>

Question No : 18 DRAG DROP - (Topic 2)

You need to implement the code flow strategy for Project2 in Azure DevOps.

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

Actions	Answer Area
<input type="text" value="Create a fork"/>	
<input type="text" value="Create a branch"/>	
<input type="text" value="Add a build validation policy."/>	⬅
<input type="text" value="Add a build policy"/>	➡
<input type="text" value="Create a repository"/>	⬆
<input type="text" value="Add an application access policy."/>	⬇

Answer:

Actions	Answer Area
<input type="text" value="Create a fork"/>	
<input type="text" value="Create a branch"/>	<input type="text" value="Create a repository"/>
<input type="text" value="Add a build validation policy."/>	<input type="text" value="Create a branch"/>
<input type="text" value="Add a build policy"/>	<input type="text" value="Add a build validation policy."/>
<input type="text" value="Create a repository"/>	
<input type="text" value="Add an application access policy."/>	