ITExams

Exam code or name...   Q

Microsoft Azure DevOps Solutions v1.0 (AZ-400)

[10 questions per page ▾]

Question 36 (Testlet 3)

Case Study -
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.
To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study -
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

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

Existing Environment -
Contoso plans to improve its IT development and operations processes by 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.

Requirements -

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 Repos 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 Project1.
Whenever possible, use Azure resources.
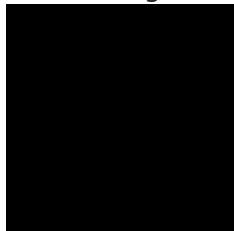Avoid using deprecated technologies.
Implement a code flow strategy for Project2 that will:
-Enable Team2 to submit pull requests for Project2.
-Enable Team2 to work independently on changes to a copy of Project2.
-Ensure that any intermediary changes performed by Team2 on a copy of Project2 will be 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.

DRAG DROP -
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 them in the correct order.
Select and Place:

Actions   **Answer Area**

Create a fork

Create a branch

Add a build policy for the fork   ⊗
                                         ⊙
Add a build policy for the master   ⊗
branch

Create a repository

Add an application access policy.

Expose Correct Answer

Answer :

Actions   **Answer Area**

                                     Create a repository

Create a branch   Create a fork

                                     Add a build policy for the fork
                                  ⊗   ⊙
Add a build policy for the master ⊙   ⊙
branch

Add an application access policy.

Explanation:

Step 1: Create a repository -
A Git repository, or repo, is a folder that you've told Git to help you track file changes in. You can
have any number of repos on your computer, each stored in their own folder.
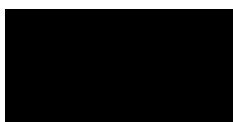
Step 2: Create a fork -
Step 3: Add a build policy for the fork
Build policies help teams protect their important branches of development. Policies enforce your
team's code quality and change management standards.
Scenario:
Implement a code flow strategy for Project2 that will:
Enable Team2 to submit pull requests for Project2.

-> Enable Team2 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.

> 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.

Reference:

https://docs.microsoft.com/en-us/azure/devops/repos/git/manage-your-branches

Next Question

Question 37 (Testlet 3)

Case Study -

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study -

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

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

Existing Environment -
Contoso plans to improve its IT development and operations processes by 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.

Requirements -

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 Repos 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 Project1.
Whenever possible, use Azure resources.
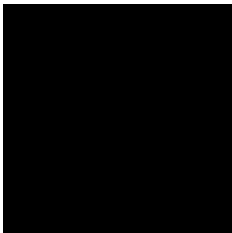Avoid using deprecated technologies.
Implement a code flow strategy for Project2 that will:
-Enable Team2 to submit pull requests for Project2.
-Enable Team2 to work independently on changes to a copy of Project2.
-Ensure that any intermediary changes performed by Team2 on a copy of Project2 will be 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.

DRAG DROP -

You need to configure Azure Automation for the computers in Pool7.

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.

Select and Place:

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

Expose Correct Answer

Answer :

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

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.

Reference:

https://docs.microsoft.com/en-us/powershell/module/azurerm.automation/import-azurermautomationdscconfiguration https://docs.microsoft.com/en-us/powershell/module/azurerm.automation/start-azurermautomationdsccompilationjob

Implement DevOps Development Processes

Next Question

Question 38 (Question Set 2)

You are automating the testing process for your company.
You need to automate UI testing of a web application.
Which framework should you use?

A. JaCoco
B. Selenium
C. Xamarin.UITest
D. Microsoft.CodeAnalysis

Expose Correct Answer

Answer : **B**

Explanation:
Performing user interface (UI) testing as part of the release pipeline is a great way of detecting unexpected changes, and need not be difficult. Selenium can be used to test your website during a continuous deployment release and test automation.
Reference:
https://docs.microsoft.com/en-us/azure/devops/pipelines/test/continuous-test-selenium?view=azure-devops

Next Question

Question 39 (Question Set 2)

You have an Azure DevOps organization named Contoso, an Azure DevOps project named Project1, an Azure subscription named Sub1, and an Azure key vault named vault1.
You need to ensure that you can reference the values of the secrets stored in vault1 in all the pipelines of Project1. The solution must prevent the values from being stored in the pipelines.
What should you do?

A. Create a variable group in Project1.
B. Add a secure file to Project1.
C. Modify the security settings of the pipelines.
D. Configure the security policy of Contoso.

Expose Correct Answer

Answer : **A**

Explanation:
Use a variable group to store values that you want to control and make available across multiple pipelines.
Reference:
https://docs.microsoft.com/en-us/azure/devops/pipelines/library/variable-groups

Next Question

Question 40 (Question Set 2)

DRAG DROP -
You are configuring Azure Pipelines for three projects in Azure DevOps as shown in the following table.

Project name	Project Details

Project1	The project team provides preconfigured YAML files that it wants to use to manage future pipeline configuration changes.
Project2	The sensitivity of the project requires that the source code be hosted on the managed Windows server on your company's network.
Project3	The project team requires a centralized version control system to ensure that developers work with the most recent version.

Which version control system should you recommend for each project? To answer, drag the appropriate version control systems to the correct projects. Each version control system 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.

Select and Place:

Version Control Systems

Assembla Subversion

Bitbucket Cloud

Git in Azure Repos

GitHub Enterprise

Answer Area

Project1: [   ]

Project2: [   ]

Project3: [   ]

Expose Correct Answer

Answer :

Version Control Systems

Assembla Subversion

Bitbucket Cloud

Git in Azure Repos

GitHub Enterprise

Answer Area

Project1: Git in Azure Repos

Project2: GitHub Enterprise

Project3: Bitbucket Cloud

Explanation:

Project1:Git in Azure Repos -

Project2: Github Enterprise -
GitHub Enterprise is the on-premises version of GitHub.com. GitHub Enterprise includes the same great set of features as GitHub.com but packaged for running on your organization's local network. All repository data is stored on machines that you control, and access is integrated with your organization's authentication system (LDAP, SAML, or CAS).

Project3: Bitbucket cloud -
One downside, however, is that Bitubucket does not include support for SVN but this can be easily

amended migrating the SVN repos to Git with tools such as
SVN Mirror for Bitbucket .

Note: SVN is a centralized version control system.
Incorrect Answers:
Bitbucket:
Bitbucket comes as a distributed version control system based on Git.
Note: 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.
Our systems support two types of source control: Git (distributed) and Team Foundation Version Control (TFVC). TFVC is a centralized, client-server system. In both Git and TFVC, you can check in files and organize files in folders, branches, and repositories.
Reference:
https://www.azuredevopslabs.com/labs/azuredevops/yaml/
https://enterprise.github.com/faq

Next Question

Page: 8 / 49
Total 243 questions

← Previous Page   Next Page →

10 questions per page ▾

DMCA & LEGAL