Free Questions for AZ-400

Shared by Patrick on 04-10-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

SIMULATION

Task 12

You need to create a personal access token (PAT) named Token! that has only the following capabilities

* Read write, and manage code

* Read and execute builds

* Read releases

Token1 must expire in 60 days.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Navigate to Personal Access Tokens

Sign in to Azure DevOps:

Go toAzure DevOpsand sign in with your credentials.

Access User Settings:

Click on your profile picture in the top right corner.

SelectUser settings.

Open Personal Access Tokens:

In the user settings menu, selectPersonal access tokens.

Step 2: Create a New Personal Access Token

Create a New Token:

Click on+ New Token.

Configure the Token:

Name: EnterToken1.

Organization: Select the organization where you want to use the token.

Expiration: Set the expiration to60 days.

Set Scopes:

Code: SelectRead, Write, & Manage.

Build: SelectRead & Execute.

Release: SelectRead.

Create the Token:

Click onCreate.

Step 3: Save the Token

Copy the Token:

Once the token is created, copy it immediately as it will not be displayed again.

Store the token in a secure location.

By following these steps, you will have successfully created a personal access token namedToken1with the specified capabilities and a 60-day expiration

Question 2

Question Type: MultipleChoice

SIMULATION

Task 11

You need to write a KQL query that will count the number of inbound requests for each source IP address, for any connection made during the last three months of 2021.

Open Azure Data Explorer by using the following link:

https//dataexploter azure.com/clusters/help/databases/Securiitylogs

The requests are contained in a table named InboundBrowsing in the Securitylogs connection.

The query must return two columns named NumberOfRequests and SourcelP.

Export the query result to C:\Samples

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Write the KQL Query

Open Azure Data Explorer:

Navigate to Azure Data Explorer and sign in with your credentials.

Access the Securitylogs Database:

Open theSecuritylogsdatabase.

Write the Query:

Use the following KQL query to count the number of inbound requests for each source IP address:

```
InboundBrowsing
| where Timestamp between (datetime(2021-10-01) .. datetime(2021-12-31))
| summarize NumberOfRequests = count() by SourceIP
| project SourceIP, NumberOfRequests
```

Step 2: Export the Query Results

Run the Query:

Execute the query in Azure Data Explorer.

Export the Results:

Once the query results are displayed, click on theExportbutton.

Choose the export format (e.g., CSV) and specify the export path asC:\Samples.

By following these steps, you will have successfully written a KQL query to count the number of inbound requests for each source IP address during the last three months of 2021 and exported the results toC:\Samples

Question 3

Question Type: **MultipleChoice**

SIMULATION

Task 10

For Project1, you need to ensure that artifacts, symbols, and attachments are retained for 60 days.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Navigate to Project Settings

Navigate to Azure DevOps:

Go to Azure DevOps and sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Access Project Settings:

In the left-hand menu, scroll down and selectProject settings.

Step 2: Configure Retention Policies

Navigate to Pipelines Settings:

UnderPipelines, selectSettings.

Set Retention Policies:

In theRetentionsection, set the number of days to keep artifacts, symbols, and attachments to60 days.

Ensure that the retention policy is applied to all relevant pipelines and branches.

Save Changes:

Click onSaveto apply the retention policy1.

By following these steps, you will have successfully configured the retention policy to retain artifacts, symbols, and attachments for 60 days inProject1

Question 4

Question Type: **MultipleChoice**

SIMULATION

Task 9

In Project1, you need to create a variable group named varGroup1 that will contain the following variables:

serverNane: server1 dbName: db1

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Navigate to the Library

Navigate to Azure DevOps:

Go to Azure DevOps and sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Access the Library:

In the left-hand menu, selectPipelines>Library.

Step 2: Create a Variable Group

Create a New Variable Group:

On the Library page, click on+ Variable group.

Configure the Variable Group:

Name: EntervarGroup1.

Description: Optionally, add a description for the variable group.

Add Variables:

Click on+ Addto add a new variable.

Variable Name: EnterserverName.

Value: Enterserver1.

ClickOK.

Click on+ Addagain to add another variable.

Variable Name: EnterdbName.

Value: Enterdb1.

ClickOK.

Save the Variable Group:

Click onSaveto save the variable group.

By following these steps, you will have successfully created a variable group namedvarGroup1containing the specified variables

Question 5

Question Type: **MultipleChoice**

SIMULATION

Task 8

Initialize the default main branch, if it does not exist already.

In Project 1, you need to create a new Azure Pipelines YAML pipeline by using the ASP.NET template.

The pipeline must use Azure Repos as the hosting platform and must be created in a new branch named azure-pipelines.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Initialize the Default Main Branch

Navigate to Azure DevOps:

Go to Azure DevOps and sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Initialize the Main Branch:

Go toRepos>Files.

If the main branch does not exist, you will see an option to initialize it.Click onInitializeand follow the prompts to create the main branch1.

Step 2: Create a New Branch for the Pipeline

Navigate to Branches:

Go toRepos>Branches.

Click onNew branch.

Create the Branch:

Enterazure-pipelinesas the branch name.

Selectmainas the base branch.

ClickCreate2.

Step 3: Create a New Azure Pipelines YAML Pipeline

Navigate to Pipelines:

Go toPipelines>New pipeline.

Select the Repository:

ChooseAzure Repos Gitand select the relevant repository.

Configure the Pipeline:

SelectStarter pipeline.

Replace the default YAML with the ASP.NET template. You can find the ASP.NET template in the Azure Pipelines documentation or use the following example:

```
trigger:

- main

pool:

vmImage: 'windows-latest'

variables:

buildConfiguration: 'Release'

steps:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

installationPath: $(Agent.ToolsDirectory)/dotnet

- script: |

dotnet build --configuration $(buildConfiguration)

displayName: 'Build project'

- script: |
```

dotnet test --no-build --configuration $(buildConfiguration)

displayName: 'Run tests'

Save the Pipeline:

Click onSaveand enterazure-pipelinesas the branch name.

Click onSave and runto save the pipeline to the new branch namedazure-pipelines3.

By following these steps, you will have successfully initialized the main branch, created a new branch namedazure-pipelines, and set up a new Azure Pipelines YAML pipeline using the ASP.NET template

Question 6

Question Type: **MultipleChoice**

SIMULATION

Task 7

Initialize the default main branch, if it does not exist already

In the User 1-42147509 organization, you need to install the Microsoft Security DevOps extension.

Next, create a new starter pipeline named starter1 that will use the following starter code.

```
trigger:
    - starter
pool:
    # ubuntu-latest also supported.
    vmImage: 'windows-latest'
steps:
```

Ensure that starter! includes a task that executes the extension and uses the following input*:

* Command: run

* Policy aruredevops

* Publish: true

Save the pipeline to a new branch named starter

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Initialize the Default Main Branch

Navigate to Azure DevOps:

Go toAzure DevOpsand sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Initialize the Main Branch:

Go toRepos>Files.

If the main branch does not exist, you will see an option to initialize it.Click onInitializeand follow the prompts to create the main branch1.

Step 2: Install the Microsoft Security DevOps Extension

Navigate to Extensions:

In Azure DevOps, click on theShopping Bagicon in the top right corner and selectBrowse Marketplace.

Search for the Extension:

Search forMicrosoft Security DevOps.

Install the Extension:

Click onGet it free.

Select your organization (User1-42147509) and clickInstall.

Follow the prompts to complete the installation2.

Step 3: Create a New Starter Pipeline

Navigate to Pipelines:

Go toPipelines>New pipeline.

Select the Repository:

ChooseAzure Repos Gitand select the relevant repository.

Configure the Pipeline:

SelectStarter pipelineand replace the default YAML with the following starter code:

```
trigger:

- starter

pool:

vmImage: 'windows-latest'

steps:

- task: MicrosoftSecurityDevOps@1
```

inputs:

command: 'run'

policy: 'azuredevops'

publish: true

Save the Pipeline:

Click onSaveand enterstarteras the branch name.

Click onSave and runto save the pipeline to the new branch namedstarter3.

By following these steps, you will have successfully initialized the main branch, installed the Microsoft Security DevOps extension, and created a new starter pipeline namedstarter1that includes the specified task

Question 7

Question Type: **MultipleChoice**

SIMULATION

Task 6

Initialize the default main branch, if it does not exist already.

For Project 1, you need to implement branch merging restrictions to enable squash merge for all changes merged into the main branch.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Initialize the Default Main Branch

Navigate to Azure DevOps:

Go to Azure DevOps and sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Initialize the Main Branch:

Go toRepos>Files.

If the main branch does not exist, you will see an option to initialize it.Click onInitializeand follow the prompts to create the main branch1.

Step 2: Enable Squash Merge for the Main Branch

Navigate to Branch Policies:

Go toRepos>Branches.

Find themainbranch and click on the...(ellipsis) next to it.

SelectBranch policies.

Enable Squash Merge:

UnderPolicies, scroll down to theMerge strategysection.

SelectSquash mergeas the required merge strategy2.

Save Changes:

Click onSave changesto apply the policies.

Step 3: Verify the Squash Merge Policy

Create a Pull Request:

Make a change in a branch and create a pull request to merge it into the main branch.

Complete the Pull Request:

Ensure that the pull request uses the squash merge strategy by selectingSquash commitunder theMerge typein the Complete pull request dialog

Question 8

Question Type: MultipleChoice

SIMULATION

Task 5

For Project1, you need to create a project wiki named Wiki1 that uses the Mermaid syntax to render a diagram A sample of the desired output is stored in C:\Resources\TCPHandshake.png.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Create a Project Wiki

Navigate to Azure DevOps:

Go to Azure DevOps and sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Create a Wiki:

In the left-hand menu, selectWiki.

Click onCreate project wiki.

Enter the nameWiki1and clickCreate.

Step 2: Add Mermaid Syntax to Render a Diagram

Open the Wiki Page:

Navigate to the newly createdWiki1.

Edit the Wiki Page:

Click onEditto start editing the wiki page.

Insert Mermaid Diagram:

Use the following Mermaid syntax to render a diagram. For example, to render a simple flowchart, you can use:

```mermaid

graph TD;

A-->B;

A-->C;

B-->D;

C-->D;

Save the Page:

Click onSaveto save your changes.

Step 3: Render the TCP Handshake Diagram

Convert TCPHandshake.png to Mermaid Syntax:

Since you have a sample diagram inC:\Resources\TCPHandshake.png, you need to convert this diagram into Mermaid syntax. Here's an example of how a TCP handshake might look in Mermaid syntax:
```

```mermaid

sequenceDiagram

participant Client

participant Server

Client->>Server: SYN

Server-->>Client: SYN-ACK

Client->>Server: ACK
```

Add the Diagram to the Wiki:

Replace the sample Mermaid syntax with the TCP handshake diagram syntax in the wiki page.

Save the Page:

Click onSaveto save your changes.

By following these steps, you will have created a project wiki namedWiki1and used Mermaid syntax to render a diagram

Question 9

Question Type: **MultipleChoice**

SIMULATION

Task 4

Initialize the default main branch, if it does not exist already.

In the Userl-42147509 Azure DevOps project, you need to implement an approval process for the default main branch.

The process must use the four-eyes principle and have at least one approval on every iteration.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

Step 1: Initialize the Default Main Branch

Navigate to Azure DevOps:

Go to Azure DevOps and sign in with your credentials.

Select Your Project:

ChooseProject1from your list of projects.

Initialize the Main Branch:

Go toRepos>Files.

If the main branch does not exist, you will see an option to initialize it. Click onInitializeand follow the prompts to create the main branch.

Step 2: Implement an Approval Process for the Main Branch

Navigate to Branch Policies:

Go toRepos>Branches.

Find themainbranch and click on the...(ellipsis) next to it.

SelectBranch policies.

Enable Required Reviewers:

UnderPolicies, enableMinimum number of reviewers.

Set the minimum number of reviewers to2to enforce the four-eyes principle.

Add Required Reviewers:

Add the users who should review the changes. Ensure that at least one approval is required on every iteration.

Enable Reset Code Reviewer Votes:

Enable theReset code reviewer votes when there are new changesoption to ensure that any new changes require re-approval.

Save Changes:

Click onSave changesto apply the policies.

Step 3: Verify the Approval Process

Create a Pull Request:

Make a change in a branch and create a pull request to merge it into the main branch.

Review and Approve:

Ensure that the pull request requires at least two reviewers to approve it before it can be merged.

By following these steps, you will have successfully initialized the main branch and implemented an approval process that adheres to the four-eyes principle, ensuring that every iteration has at least one approval