



Free Questions for AZ-400 by [braindumpscollection](#)

Shared by [Fox](#) on 09-08-2024

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

SIMULATION

Task 8

You need to ensure that the <https://contoso.com/statushook> webhook is called every time a repository named az40038443478acr1 receives a new version of an image named dotnetapp

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To ensure that the webhook at <https://contoso.com/statushook> is called every time the repository named az40038443478acr1 receives a new version of an image named dotnetapp, you can follow these steps to configure a webhook in Azure Container Registry:

Navigate to the Azure Container Registry:

Go to the Azure Portal.

Find and select your Azure Container Registry instance az40038443478acr1.

Create a New Webhook:

Under Services, select Webhooks.

Click on + Add to create a new webhook.

Fill in the form with the following information:

Webhook name: Enter a unique name for your webhook.

Service URI: Enter `https://contoso.com/statushook`.

Custom headers: (Optional) Add any headers you want to pass along with the POST request.

Trigger actions: Select Push to trigger the webhook on image push events.

Scope: Specify the scope as `az40038443478acr1:dotnetapp` to target the specific image.

Status: Set to Enabled.

Save the Webhook Configuration:

Review the information and click Create to save the webhook.

Once configured, the webhook will send a POST request to <https://contoso.com/statushook> whenever a new version of the dotnetapp image is pushed to the az40038443478acr1 repository in your Azure Container Registry1.

This setup will automate the notification process, ensuring that the specified webhook is called with each new image version, thus fulfilling the task requirements.

Question 2

Question Type: MultipleChoice

SIMULATION

Task 7

You need to prepare a network security group (NSG) named az400 38443478 nsg1 to host an Azure DevOps pipeline agent. The solution must allow only the required outbound port for Azure DevOps and deny all other inbound and outbound access to the Internet

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To prepare a Network Security Group (NSG) named az400-38443478-nsg1 for hosting an Azure DevOps pipeline agent, while allowing only the required outbound port for Azure DevOps and denying all other inbound and outbound access to the Internet, follow these steps:

Create the NSG:

Navigate to the Azure Portal.

Go to Network Security Groups and click on + Create.

Fill in the details, including the name az400-38443478-nsg1, and create the NSG.

Configure Outbound Security Rules:

Once the NSG is created, go to its settings.

Navigate to Outbound security rules.

Click on + Add to create a new rule.

[Set the Destination port ranges to 443, which is the required port for Azure DevOps12.](#)

Set the Protocol to TCP.

Set the Action to Allow.

Assign a Priority number (e.g., 100) that does not conflict with existing rules.

Provide a meaningful Name for the rule (e.g., AllowAzureDevOps).

Configure Default Rules to Deny All Other Traffic:

In the same Outbound security rules section, edit the default rule to deny all traffic.

Change the Action to Deny for the rule with the lowest priority (highest number).

Ensure that this rule applies to all protocols, source and destination IP ranges, and port ranges.

Associate the NSG with the Appropriate Resource:

Associate the NSG with the subnet or network interface of the virtual machine or resource where the Azure DevOps pipeline agent will be hosted.

By following these steps, you will ensure that the Azure DevOps pipeline agent can communicate with Azure DevOps services over the required port while blocking all other inbound and outbound Internet access, adhering to the principle of least privilege and security best practices.

Question 3

Question Type: MultipleChoice

SIMULATION

Task 6

You have an Azure function hosted in an App Service plan named az400-38443478-funct

You need to configure az400 38443478-funct to update the functions automatically whenever new code is committed to the main branch of <https://github.com/Azure-Samples/functions-quickstart>.

NOTE: Access to GitHub is blocked in the exam environment. Access to GitHub is NOT required to complete the task successfully.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To configure your Azure Function az400-38443478-funct to automatically update whenever new code is committed to the main branch of the specified GitHub repository, you can use GitHub Actions for continuous deployment. Here's how to set it up:

Create a GitHub Actions Workflow:

In your GitHub repository, navigate to the `.github/workflows/` directory.

Create a new file for your workflow (e.g., `azure-function-cd.yml`).

Define the Workflow:

In the workflow file, define the steps for the build and deployment process.

Use the `Azure/functions-action` to deploy to your Azure Function App.

Set up triggers for the main branch to initiate the workflow on every commit.

Generate Deployment Credentials:

In the Azure Portal, navigate to your Function App `az400-38443478-funct`.

Download the publish profile from the Overview section by clicking on `Get publish profile`.

Store the Publish Profile as a GitHub Secret:

In your GitHub repository, go to `Settings > Secrets and variables > Actions`.

Create a new secret (e.g., `AZURE_FUNCTIONAPP_PUBLISH_PROFILE`) and paste the content of the publish profile.

Configure the Workflow to Use the Secret:

In the workflow file, reference the secret to authenticate the deployment to Azure.

Here's a sample GitHub Actions workflow snippet:

name: Deploy Azure Function

on:

push:

branches:

- main

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up Python version

uses: actions/setup-python@v2

with:

python-version: '3.x'

- name: Install dependencies

run: |

```
pip install -r requirements.txt
```

- name: Deploy to Azure Functions

uses: Azure/functions-action@v1

with:

app-name: az400-38443478-funct

publish-profile: \${{ secrets.AZURE_FUNCTIONAPP_PUBLISH_PROFILE }}

package: .

Replace the app-name with the name of your Azure Function App and ensure the Python version and dependencies match your application's requirements.

By following these steps, your Azure Function will automatically update whenever new code is pushed to the main branch of the GitHub repository. This setup minimizes manual effort and ensures that your function app is always running the latest code.

Question 4

Question Type: MultipleChoice

SIMULATION

Task 5

You plan to store signed images in an Azure Container Registry instance named az40038443478act1.

You need to modify the SKU for az40038443478aa1 to support the planned images. The solution must minimize costs.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To store signed images in an Azure Container Registry (ACR) instance and support your planned images while minimizing costs, you need to modify the SKU of your ACR instance to one that supports content trust and image signing. Here's how you can do it:

Determine the Appropriate SKU:

[Content trust and image signing are features of the Premium service tier of Azure Container Registry1.](#)

If cost minimization is a priority, ensure that the Premium tier is necessary for your use case. If you require content trust, the Premium tier is the appropriate choice.

Modify the SKU of the ACR Instance:

Navigate to the Azure Portal.

Go to your ACR instance az40038443478act1.

Select Update from the overview pane.

[Choose the Premium SKU from the SKU drop-down menu2.](#)

Review the changes and pricing, then save the configuration.

By upgrading to the Premium SKU, you'll be able to store signed images in your ACR instance. Remember to monitor your usage and costs to ensure they align with your budget and requirements.

Question 5

Question Type: MultipleChoice

SIMULATION

Task 4

You need to configure a virtual machine template in a DevTest Labs environment named az400-38443478-dtl1. The operating system must be based on Windows Server 2016 Datacenter. Virtual machines created from the DevTest Lab must include the Selenium tool and the Google Chrome browser.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To configure a virtual machine template in your DevTest Labs environment named az400-38443478-dtl1 with Windows Server 2016 Datacenter that includes the Selenium tool and the Google Chrome browser, follow these steps:

Create a Custom Image with Windows Server 2016 Datacenter:

In the Azure Portal, go to your DevTest Lab az400-38443478-dtl1.

Navigate to Configuration and policies > Custom images.

Use an existing VM or create a new one with Windows Server 2016 Datacenter.

[After setting up the VM, capture it to create a custom image1.](#)

Install Selenium and Google Chrome on the VM:

Connect to the VM via RDP.

[Download and install the Selenium WebDriver for your preferred programming language from the official Selenium website2.](#)

[For Google Chrome, download the offline installer from the official website and install it on the VM3.](#)

Generalize the VM:

Run the sysprep command to generalize the VM, which prepares it to be used as a template.

Shut down the VM after sysprep completes.

Capture the Generalized VM to Create a Template:

In the Azure Portal, navigate to the VM and select Capture.

Provide the required details and create the image.

Add Selenium and Google Chrome Artifacts to the Template:

Go back to the DevTest Lab az400-38443478-dtl1.

Select Artifacts and add Selenium and Google Chrome artifacts to the template.

Ensure these artifacts are configured to install during the VM creation process.

Create VMs from the Template:

Now, when you create a new VM in the DevTest Lab, select the custom image you created.

The VM will be provisioned with Windows Server 2016 Datacenter, and the Selenium tool and Google Chrome browser will be installed automatically.

By following these steps, you can ensure that all virtual machines created from this template in your DevTest Lab will have the required operating system, tools, and browser installed. Remember to replace placeholder names with the actual names of your resources where necessary.

Question 6

Question Type: MultipleChoice

SIMULATION

Task 3

You need to ensure that an Azure Web App named az400-38443478-main supports rolling upgrades. The solution must ensure that only 10 percent of users who connect to az400-38443478 main use updated versions of the app. The solution must minimize administrative effort.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To ensure that your Azure Web App named az400-38443478-main supports rolling upgrades and only 10 percent of users connect to the updated version of the app, you can use deployment slots with the following steps:

Create a Deployment Slot:

Navigate to the Azure Portal.

Go to your Web App az400-38443478-main.

Select Deployment slots in the menu.

Click on Add Slot.

Name the slot (e.g., staging) and if needed, clone settings from the production slot.

Configure the Traffic Percentage:

In the Deployment Slots menu, you will see a column for Traffic %.

[Set the traffic percentage to 10% for the staging slot1.](#)

This will route only 10% of the traffic to the updated version of the app in the staging slot.

Deploy the Updated App to the Staging Slot:

Deploy your updated application to the staging slot.

Test the application in the staging slot to ensure it's working as expected.

Complete the Rolling Upgrade:

Once you're satisfied with the performance and stability of the app in the staging slot, you can gradually increase the percentage of traffic until you're ready to swap with the production slot.

To swap slots, go to the Deployment slots menu and click on Swap with the production slot.

By using deployment slots, you can achieve rolling upgrades with minimal administrative effort, as it allows you to test the new version on a subset of users before fully releasing it. Remember to adjust the traffic percentage and monitor the application's performance throughout the process.

Question 7

Question Type: MultipleChoice

SIMULATION

Task 2

You need to create an instance of Azure Application Insights named az400-38443478-main and configure the instance to receive telemetry data from an Azure web app named az400-38443478-main

You must create a Log Analytics workspace before this task.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To create an instance of Azure Application Insights named az400-38443478-main and configure it to receive telemetry data from an Azure web app with the same name, you'll need to follow these steps:

Create a Log Analytics Workspace:

Go to the Azure Portal.

Search for Log Analytics Workspaces and select Add.

Select a Subscription and either use an existing Resource Group or create a new one.

Provide a unique name for your Log Analytics workspace.

Choose the Region that is appropriate for you.

[Review the settings and then select Create1.](#)

Create an Azure Application Insights Instance:

In the Azure Portal, navigate to Application Insights.

Click on + Create.

Fill in the instance details, ensuring the name is az400-38443478-main.

Link the instance to the Log Analytics workspace you created in the previous step.

[Review and create the Application Insights instance2.](#)

Configure the Azure Web App to Send Telemetry Data:

Go to the Azure Web App az400-38443478-main.

Under Monitoring, select Application Insights.

Choose to use an existing resource and select the Application Insights instance you created.

Follow the prompts to set up the connection, which may involve adding the appropriate SDK to your web app and configuring the connection string or instrumentation key.

Verify Telemetry Data Reception:

After setting up, send some test traffic to your web app.

Then, go to the Application Insights instance and check the Overview or Performance sections to see if telemetry data is being received.

Remember to replace placeholder names with the actual names of your resources where necessary. These steps will help you set up Azure Application Insights to monitor your web app effectively.

Question 8

Question Type: MultipleChoice

SIMULATION

Task 1

You need to ensure that an Azure Web App named az400-38443478-main can retrieve secrets from an Azure key vault named az400-3844J478-kv1 by using a system managed identity. The solution must use the principle of least privilege.

Options:

A- See the solution below in explanation

Answer:

A

Explanation:

To ensure that your Azure Web App named az400-38443478-main can retrieve secrets from an Azure Key Vault named az400-3844J478-kv1 using a system managed identity with the principle of least privilege, follow these detailed steps:

Enable a System Managed Identity for the Azure Web App:

Navigate to the Azure Portal.

Go to the Azure Web App az400-38443478-main.

Select Identity under the Settings section.

In the System assigned tab, switch the Status to On.

Click Save to apply the changes.

Grant the Web App Access to the Key Vault:

Go to the Azure Key Vault az400-3844J478-kv1.

Select Access policies under the Settings section.

Click on Add Access Policy.

Choose Secret permissions and select Get and List. This grants the app the ability to read secrets, adhering to the principle of least privilege.

Click on Select principal, search for your Web App name az400-38443478-main, and select it.

Click Add to add the policy.

Don't forget to click Save to save the access policy changes.

Retrieve Secrets in the Web App Code:

In your Web App's code, use the Azure SDK to retrieve the secrets.

For example, in a .NET application, you can use the `Azure.Identity` and `Azure.Security.KeyVault.Secrets` namespaces.

Utilize the `DefaultAzureCredential` class which will automatically use the system managed identity when running on Azure.

```
using Azure.Identity;
```

```
using Azure.Security.KeyVault.Secrets;
```

```
var client = new SecretClient(new Uri('https://az400-3844J478-kv1.vault.azure.net/'), new DefaultAzureCredential());
```

```
KeyVaultSecret secret = await client.GetSecretAsync('my-secret-name');
```

```
string secretValue = secret.Value;
```

Replace 'my-secret-name' with the actual name of the secret you want to retrieve.

By following these steps, your Azure Web App will be able to securely retrieve secrets from the Azure Key Vault using a system managed identity, without needing to store credentials in the code, and adhering to the principle of least privilege. Remember to replace the placeholder names with the actual names of your Web App and Key Vault.

Question 9

Question Type: Hotspot

You have a .NET app named App1.

You need to upload App1 to GitHub Packages.

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 Area

Answer:

make	nuget push	"bin/Release/app1.1.0.0.nupkg" --api-key PAT_TOKEN --source "github"
dotnet	nuget add source	
git	nuget push	
make	pack	
npm	publish	

Question 10

Question Type: MultipleChoice

You have an Azure subscription.

You use Bicep templates to deploy websites and Azure SQL infrastructure.

You need to automate the deployments by using Azure Pipelines and a self-hosted agent that runs on two virtual machines. The solution must minimize administrative effort.

What should you do first?

Options:

A- Create an Azure Automation account.

- B-** On each virtual machine, enable a system-assigned managed identity.
- C-** Create a user-assigned managed identity.
- D-** Create a service principal.

Answer:

D

To Get Premium Files for AZ-400 Visit

<https://www.p2pexams.com/products/az-400>

For More Free Questions Visit

<https://www.p2pexams.com/microsoft/pdf/az-400>

