AZURE CLOUD (ENGLISH)

7. forduló

MSCI

A kategória támogatója: MSCI

Ismertető a feladatlaphoz

Az utolsó fordulókhoz érkeztünk, így megosztunk 1-2 fontos információt a továbbiakról:

- a versennyel kapcsolatos észrevételeket december 5-ig tudjátok velünk megosztani a szokásos helyen
- az utolsó fordulóhoz kapcsolódó megoldások november 30-án érhetők el
- a végeredményről tájékoztatás decemberben, részletek hamarosan

Sok sikert az utolsó fordulóhoz!

This round contains questions related to Azure Function App development, deployment, configuration and security.

1. feladat 1 pont

What is the most secure way to grant permissions for an Azure function app to be able to access Azure resources?

Válasz

- ○ Service principal using secret stored in app settings of the function app
- ○ Service principal using certificate
- ○ Service principal using secret stored in key vault referenced by app settings of function app
- ○ System assigned managed identity

2. feladat 2 pont

What actions do make it possible to restrict the access to a function app from specific IP ranges?

Válaszok

- [ ] Apply network access restrictions in the Azure portal
- [ ] Add access restriction rules using PowerShell
- [ ] Deploy the function app with access restriction using Terraform
- [ ] Restrict specific IP ranges in the configuration file of the function app

3. feladat 2 pont

You would like to apply throttling for your function app in order to increase the performance and security by limiting the maximum number of incoming request from the same IP. What are the feasible ways to implement it?

Válaszok

- [ ] Setup the throttling in the configuration file of the function app
- [ ] Implement the throttling in the code of the function app
- [ ] Use Azure Front Door and set the throttling in it
- [ ] Use Azure API Management and apply the throttling in it

4. feladat 5 pont

You need to implement an HTTP trigger in a C# (.NET 6.0, 4.x Azure function runtime) function app. The MyClient class containing the business logic implements the IMyClient interface, the running application has only one instance of this class which will be passed to the trigger by dependency injection. You decided to use the out-of-the-box dependency injection capability in Azure Functions which is built on the .NET Core Dependency Injection features. What are the required steps in order to implement the function app with the HTTP trigger and dependency injection?

Válaszok

- [ ] Using the Microsoft.Azure.Functions.Extensions.DependencyInjection namespace
- [ ] Using the Microsoft.Extensions.DependencyInjection namespace

- [ ] Implementing the FunctionsStartup abstract class with the Startup method
- [ ] Implementing the FunctionsStartup abstract class with the Configure method
- [ ] Calling IFunctionsHostBuilder.Services.AddTransient in order to add the MyClient implementation the dependency container
- [ ] The class implementing the HTTP trigger needs to have a constructor with an IMyClient input parameter
- [ ] Apply the HTTPTrigger attribute on the constructor of the MyClient class

Megoldások beküldése