**Inlämningsuppgift2**

Datum: **2021-10-27**

Författare: **Ali Mesbahi molaei**

Utbildningsgrupp: **CUA20G**

Resource Group (Azure portal): **Assignment2-Ali**

**Teoridel**

**\_ Ni skall förklara och motivera varför Azure App Fabric skall användas och när det kan vara aktuellt med en implementering**

Azure Service Fabric är en plattform byggd skalbarhet, kan hantera, balansera och distribuera containers med hjälp av mikrotjänster service, även det har förmågan att integrera traditionella Windows applikationer med varandra samt dom servicetjänster.

Azure service Fabrice låter utvecklare att fokusera på sina delar det betyder att utvecklare blir av med dom extra uppgifter och kan fokusera på byggnadsfunktioner. Med Service App Fabrice plattform olika utvecklare med olika kodspråk som C#, java och mm kan skapa sina microtjänter bekymmerslös samt det förstår den tillgängliga infrastrukturen och reseurs behovet för applikationer, det själv kan automatisera skalbarheter, uppdatering samt fel lösningar(debug).

I slutet, Service fabric kan användas för att hantera mikrotjänster, automatisk scale up and scale out, när det behövs att integrera alla mikrotjänster, kunna samverka mellan applikation på applikation i Azure molntjänsten och mikrotjänster.

**\_ Förklara skillnaden mellan Stateless respektive Stateful services i Azure App Fabric**

Med Stateless Service kan hanteras inget data på egen hand Men på Stateful service kan hanteras sitt eget data genom Service Fabric’s State Managment. Stateless Service utnyttjar istället Azure SQL, Azure CosmosDB, Azure Storage. Med Stateless Service alla instanser betjäna förfrågningar medan med Stateful Service är bara den primära repliken aktiv som standard.

**\_ Beskriv vad Azure Serverless Computing är**

Serverless computing gör det möjligt för utvecklare att bygga mikrotjänster snabbare genom att hoppa behovet av att hantera infrastruktur. Med hjälp av serverlösa applikationer, skalar och hanterar molntjänstleverantören automatiskt den infrastruktur som krävs för att köra koden. Det är viktig att notera att servrar fortfarande kör koden på serverlös. Med serverlös gör det möjligt för utvecklare att ha mer fokus på affärslogiken och levererar mer värde till kärnan i verksamheten och låter organisationer ge en hög kvalitet resurser och fokuserad på nyelse/upgrating.

**\_ Beskriv vad Azure Serverless API Gateway är för något och när det är aktuellt att sätta upp en sådan**

En Api gateway sitter mellan klienter och tjänster. Det funkar som en omvänd proxy och reagerar begäranden från klienter till tjänster samt det kan funkar liksom authentication, SSL termination och begränsar hastigheten när det behövs. Om det finns ingen gateway, klienter skickar begäran direkt till klienttjänster, det kan orsaka reaktion problem från klient tjänster. Det betyder att klienten måste hålla reda på flera slutpunkter och klient ska hantera fel på ett rätt sätt. En endå åtgärd kan kräva anrop till flera tjänster, det kan leda till flera nätverksfördröjningar mellan klient och tjänster, vilket ger längre svar tiden när det finns ingen koppling som gateway mellan klient och servern. Samt tjänster måste exponera ett klientvänligt protokoll som http men utan gateway det blir ett begränsat kommunikationsprotokoll.

Gateways kan utföra ett antal olika funktioner somGateway-routing: Det använd som en omvänd proxy. Gateway-aggregering: Det använd för att aggregera flera enskilda begäranden. Gateway-avlastning: Det använd för att avlasta funktioner från enskilda tjänster.

**Praktik del**

Resource Group (Azure portal): **Assignment2-Ali**

Under ’assignment2-Ali’ har skapat 2 funktioner som studentfunktionapp och adminfunktionapp.

**studentfunktionapp**: AlisHttpTrygger1 funktion funkar utmärk när man gör Run/test och skickar ett meddelande till ’outqueue’ som output

**adminfunktionapp**: AlisTimerTrigger1 funkar en gång på vardagar kl 9.30 (0.30.9.\*.\*.1–5).

**AlisHttpTrygger1** (koden kan kopiera och klistra in)

#r "Newtonsoft.Json"

using System.Net;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Primitives;

using Newtonsoft.Json;

public static async Task<IActionResult> Run(HttpRequest req, ICollector<string> outputQueueItem, ILogger log)

{

log.LogInformation("C# HTTP trigger function processed a request.");

string name = req.Query["name"];

string requestBody = await new StreamReader(req.Body).ReadToEndAsync();

dynamic data = JsonConvert.DeserializeObject(requestBody);

name = name ?? data?.name;

string responseMessage = string.IsNullOrEmpty(name)

? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response."

: $"Hej!, Jag heter {name}. Det är Alis HTTP triggered funktion som funkar utmärkt.";

outputQueueItem.Add("Name passed to the function: " + name);

return new OkObjectResult(responseMessage);

}

**AlisTimerTrigger1** (koden kan kopiera och klistra in)

using System;

[FunctionName("AlisTimerTrigger1")]

public static void Run([TimerTrigger("0 30 9 \* \* 1-5")]TimerInfo myTimer, ILogger log)

{

if (myTimer.IsPastDue)

{

log.LogInformation("Timer is running late!");

}

log.LogInformation($"C# Timer trigger function executed at: {DateTime.Now}");

}

Logic App**: studentappflow**

**Beskrivning**:

När en meddeleande kommer genom specifika e-postadress exempelvis: Michaelsemail till min outlook (det är markerat som high importance). Därefter Logic app kontrollerar att Subjekt är inkluderat ”JensenYH project” eller inte, Om nej, det ska skickas till skärp, om Ja, det ska kontrollera ’Attachments’ för att köra ’http request’ som har skickad genom Michaels epost och sedan AlisHttpTrigger1 som en funktion ska köras i nästa steg.

