ARQUITECTURAS E INFRAESTRUCTURAS PARA INTELIGENCIA ARTIFICIAL
PRÁCTICA 5: MODELO DE COMPUTACIÓN CLOUD SERVERLESS

Objetivos

- Conocer el modelo de computación Serverless

Trabajo Guiado:

En esta práctica se va a crear una aplicación serverless. Este tipo de aplicaciones se implementan sin tener en cuenta la infraestructura sobre la que funcionan. Se definen mediante funciones de código que se ejecutan cuando se activa el disparador definido.

1) Crear una aplicación de funciones

Buscamos el servicio "Aplicación de funciones"
Click en "Crear"

Detalles del proyecto:

- Subscripción: Azure para estudiantes
- Grupo de recursos: Seleccionamos uno ("grupoest")
- Detalles de la instancia
    - Nombre de la aplicación de funciones: Introduce un nombre
    - ¿Desea implementar un código o imagen de contenedor?

      Las funciones se pueden crear implementando el código de las funciones directamente o utilizando un contenedor de Docker preparado para gestionar las peticiones. En este caso seleccionamos código.

    - Pila del entorno en tiempo de ejecución: Elegimos el lenguaje en el que queremos ejecutar la función
    - Versión: Versión del lenguaje utilizado. Generalmente se utilizará la versión más actual.
    - Región: Se intentará elegir la más cercana a los clientes. En nuestro caso "West Europe"
- Sistema operativo: Solo está disponible Linux
- Hospedaje
    - Opciones y planes de hospedaje: Elegimos consumo (sin servidor). "Functions Premium"
- Click en "Siguiente: Storage"
    - Dejamos la opción por defecto para crear una nueva cuenta de almacenamiento que guarde el código de la aplicación
- Click en "Siguiente: Redes"
    - Habilitar el acceso público: Si se activa, la función es accesible desde Internet. Si está desactivado sólo es accesible desde la red interna de Azure, es decir, que sólo podrán acceder otros recursos creados en Azure.
- Click en Siguiente: Supervisión
    - Application Insights
        - Habilitar Application Insights: No

- Click en "Siguiente: Implementación"
  - Configuración de Acciones de GitHub
    - Implementación continua: No.

      Si se tiene un repositorio de GitHub con el código, es posible hacer que cuando se realiza un push, se actualice automáticamente el código de la función en Azure.

- Click en "Siguiente: Etiquetas"
  - Asigna la etiqueta y el valor correspondiente
- Click en "Siguiente: Revisar y Crear"
- Una vez validado, hacemos click en "Crear"

2) Activar una aplicación de funciones

Dentro de una aplicación de funciones, podemos crear las funciones que necesitemos. Estas funciones pueden tener diferentes criterios de activación, y son independientes entre ellas. Vamos a crear una función que se active cuando recibe una petición de tipo get.

Buscamos el servicio "Aplicación de funciones"

- Hacemos click en la aplicación que acabamos de crear

  En esta pantalla podemos observar todas las opciones de la aplicación de funciones.

- En el apartado "Funciones" hacemos click en el menú "Funciones"
- Hacemos click en "Crear".
- Development environment:

  Hay tres opciones: La primera opción "VS Code" permite utilizar este IDE para crear, ver y modificar el código. La segunda opción te permite modificar el código con cualquier editor y actualizar la función utilizando las herramientas por línea de comandos. La tercera opción consiste en utilizar un editor directamente en el navegador. Elegimos la tercera opción "Develop in portal".

- Select a template:

  Permite elegir que disparador utilizaremos para ejecutar la función. Depende del disparador que elijamos la función se ejecutará cuando ocurra cierto evento. En este caso elegimos "HTTP trigger", que ejecutará la función cuando se reciba una petición HTTP. Otros tipos de disparadores son, por ejemplo, "Azure Blob Storage Trigger", que permite ejecutar una función cuando se agregan ficheros a un contenedor de almacenamiento.

- Template details
  - New Function: Elegimos el nombre de la función
  - Nivel de autorización: Elegimos FUNCTION.
- Click en "Crear"

  Puedes revisar el código por defecto de la plantilla en el menú "Código y prueba". El código por defecto es capaz de recibir una petición y responder.

Para probar la función introducimos en el navegador la siguiente url:
https://<your-function-app>.azurewebsites.net/api/<your-function>?code=<your-function-key>

Aunque se ha desarrollado esta función utilizando el editor web disponible en Azure, se recomienda el desarrollo de las funciones utilizando Visual Studio Code.

En la siguiente URL puedes encontrar un tutorial sobre cómo utilizar VS Code: https://learn.microsoft.com/en-us/azure/azure-functions/functions-develop-vs-code?tabs=csharp

Un aspecto a destacar es que las funciones Serverless tienen un límite de memoria RAM de 1'5GB en su plan de consumo por defecto, lo que limita el uso de estas funciones a aplicaciones de cómputo ligeras.

Los diferentes planes disponibles con sus características se pueden encontrar en la siguiente url: https://learn.microsoft.com/en-us/azure/azure-functions/functions-scale#service-limits

3) Crear una página web estática en la cuenta de almacenamiento

Las cuentas de almacenamiento es el servicio de almacenamiento de objetos (blobs) de Azure. Su funcionalidad principal es ofrecer almacenamiento seguro y altamente disponible de los objetos almacenados, con un pago por uso basado en el tamaño de los datos almacenados, y los accesos y escrituras sobre estos. Los archivos subidos se organizan en contenedores. Se puede configurar el acceso de cada objeto para que permita acceso a usuarios anónimos o no. Si el acceso permite usuarios anónimos, el acceso es público y se asigna una URL a cada objeto, que será accesible desde internet. Utilizando este servicio, es posible montar una página web estática con un coste mínimo.

Buscamos el servicio de Azure llamado "Cuentas de almacenamiento"
Click en "Crear"
Detalles del proyecto:

- Subscripción: Azure para estudiantes
- Grupo de recursos: grupoest

Detalles de la instancia:

- Nombre de la cuenta de almacenamiento: almacenamientoweb
- Región: West Europe
- Rendimiento: Estándar
- Redundancia: Almacenamiento con redundancia local (LRS)

  La redundancia define la seguridad de los datos almacenados en el contenedor. El tipo de redundancia indica la manera en la que se mantienen las réplicas de los datos almacenados. Las opciones van desde copias almacenadas en diferentes unidades de un mismo centro de datos (redundancia local), hasta copias almacenadas en diferentes zonas geográficas (redundancia de zona geográfica).

  Para esta aplicación no necesitamos gran seguridad, por eso elegimos el almacenamiento con redundancia local, que es más barato

- Click en "Siguiente: Opciones avanzadas"
  - En "Seguridad", habilitamos el acceso anónimo en contenedores individuales, y dejamos el resto de opciones por defecto
- Click en "Siguiente: Red"
  - Conectividad de red
    - Acceso de red: Habilitar el acceso público desde todas las redes
  - Enrutamiento de red

        - Preferencia de enrutamiento: Enrutamiento de internet
- Click en "Revisar"
- Click en "Crear"

Dentro de la cuenta de almacenamiento, podemos crear contenedores donde almacenar archivos.

Crear un contenedor dentro de la cuenta de almacenamiento recién creada:

- Abrir la cuenta de almacenamiento recién creada
- En el menú lateral izquierdo, selecciona "Contenedores"
- Click en "+ Contenedor"
- Nuevo contenedor:
    - Nombre: contenedorweb
    - Nivel de acceso anónimo: Blob (acceso de lectura anónimo solo para blobs)
- Click en "Crear"

Entramos en el contenedor que hemos creado y subimos los ficheros de la siguiente web:
Guarda el siguiente código html en un fichero llamado "index.html".

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Button Example</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
        }
        button {
            padding: 10px 20px;
            font-size: 16px;
        }
    </style>
</head>
<body>
    <h1>Do NOT click the button:</h1>
    <button id="myButton">the button</button>
    <p id="output"></p>

    <script>
        // JavaScript to handle the button click event
        document.getElementById("myButton").addEventListener("click", function() {
            document.getElementById("output").textContent = "Are you happy now?";
        });
    </script>
</body>
</html>
```

Ahora sube el fichero dentro del contenedor:

- Click en "Cargar"
- Arrastra el fichero "index.html" al cuadro indicado
- Click en "Cargar"

Ya hemos subido correctamente el fichero al contenedor de almacenamiento. Cada archivo "blob" que se sube tiene una url pública única, si lo hemos configurado para que sea accesible públicamente. Vamos a buscar la URL del archivo:

- Haz click en el archivo "index.html"
- En "Propiedades", copia la URL que aparece

Pegamos la URL en el navegador y observamos el contenido de la página. La URL tiene incluida el nombre de la cuenta de almacenamiento, el nombre del contenedor, y el nombre del archivo. No obstante, esta forma de acceder al fichero de la página sólo funciona si la página tiene un único fichero. Es posible configurar en la cuenta de almacenamiento una página web estática, que tendrá su propio contenedor.

Para crear una página web estática abrimos la cuenta de almacenamiento:

- En el menú lateral izquierdo seleccionamos "Sitio web estático"
- En "Sitio web estático" seleccionamos "Habilitado"
- El "Punto de conexión principal" es la URL generada para el sitio web
- En "Nombre del documento de índice" escribimos el documento al que se accede cuando se accede a la URL sin especificar un fichero concreto
  - Escribimos "index.html"
- En "Ruta de acceso del documento de error" se indica el documento que se muestra si se accede a un recurso que no existe (error 404).
  - Lo dejamos en blanco
- Click en Guardar

Se ha creado un contenedor especial llamado **$web**. En este contenedor se colocarán los ficheros de la web.

- Accede al contenedor y sube el fichero "index.html"

Una vez hayas subido el fichero "index.html" al contenedor $web, accede a la URL indicada en el "Punto de conexión principal" y comprueba que la web se muestra correctamente.

Trabajo:

Redacta una memoria de prácticas en la que describas los pasos seguidos, los problemas encontrados y las decisiones de diseño para la resolución del trabajo a realizar. Añade un apartado de conclusiones del trabajo realizado. Incluye las capturas de pantallas que veas necesarias para demostrar que has seguido los pasos indicados en el enunciado.

Normas de entrega:

- La realización del trabajo es por parejas.
- El documento debe seguir el formato definido para las publicaciones de *Lecture Notes in Computer Science* de *Springer*
  https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines
- La entrega se realizará a través de un adjunto a una tutoría de campus virtual.
- Los formatos válidos del documento son *MS Word* (.doc, .docx), *OpenDocument* (.odt) o *Portable Document Format* (.pdf).