



Universitat d'Alacant
Universidad de Alicante

Aplicaciones Distribuídas en Internet

Sergio Baeza Carrasco
Eugenio Benito López
Jorge Bueno Calderón
Nicolás Medina Díaz

Ingeniería Informática Universidad de
Alicante

Índice

1. Introducción a los microservicios

- a) Definición
- b) Arquitectura orientada a microservicios
- c) Ventajas y desventajas de usar microservicios

2. Introducción a la nube

- a) Definición
- b) Beneficios de desplegar microservicios en la nube
- c) Proveedores de cloud computing

3. Azure functions

- a) Introducción
- b) Como como azure functions se relaciona con los microservicios

4. Desarrollo de Microservicios con Azure Functions (video demo)

5. Referencias

1. Introducción a los microservicios

a) Definición

En cuanto a la definición de microservicios diremos que es una forma de desarrollar y desplegar una aplicación como un conjunto de servicios pequeños y autónomos que se comunican entre sí a través de una interfaz estándar.

Cada servicio se encarga de una tarea específica y se despliega de manera independiente, lo que permite una mayor escalabilidad, flexibilidad y escalabilidad.

Estos servicios son desarrollados, desplegados y mantenidos de manera independiente, y pueden ser escritos en diferentes lenguajes de programación y utilizar diferentes sistemas de bases de datos.

b) Arquitectura orientada a microservicios

La arquitectura orientada a microservicios es un enfoque para el diseño y desarrollo de aplicaciones basado en la idea de dividir una aplicación en pequeños servicios independientes y autónomos que se comunican entre sí a través de una interfaz estándar.

En esta arquitectura cada servicio es responsable por una tarea específica y se ejecuta en un contenedor o una instancia de una máquina virtual, permitiendo su escalabilidad y flexibilidad.

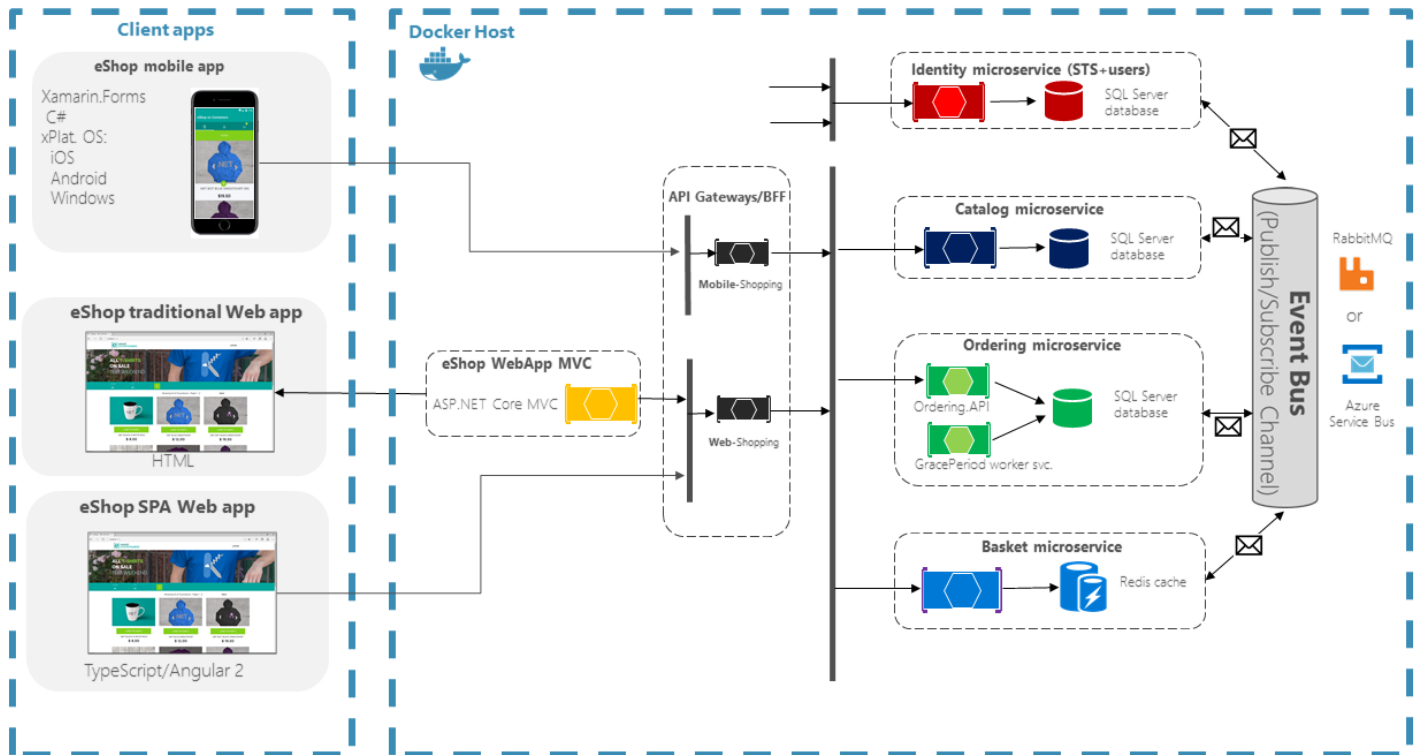
Los servicios se despliegan y ejecutan de manera independiente, y pueden ser desarrollados en diferentes lenguajes de programación y utilizar diferentes sistemas de bases de datos.

Además, se suele contar con una capa de comunicación a través de protocolos y un sistema de balanceo de carga, para lograr la alta disponibilidad y escalabilidad de la aplicación.

La idea es que los cambios en un servicio no afectan necesariamente a otros servicios, lo que hace que el desarrollo, la implementación y el mantenimiento de las aplicaciones sea más fácil.

Un ejemplo de arquitectura orientada a microservicios podría ser la siguiente:

eShopOnContainers reference application (Development environment architecture)



Dónde se puede observar que cada funcionalidad tiene su propio microservicio lo que proporciona aplicaciones estables, escalables y resilientes.

c) *Ventajas y desventajas de usar microservicios*

En este apartado comentaremos las ventajas y desventajas de emplear microservicios. Por lo que respecta a las ventajas, destacaremos la escalabilidad, flexibilidad, capacidad de adaptación y mejorar la resiliencia.

En cuanto a la escalabilidad los servicios pueden ser escalados de manera independiente, lo que permite ajustar el rendimiento según las necesidades de cada servicio.

Por otra parte, en cuanto a la flexibilidad, cada servicio puede ser desarrollado, desplegado y mantenido de manera independiente, lo que permite a los equipos de desarrollo trabajar de manera más eficiente.

La capacitación de adaptación se produce en un servicio y no afectan necesariamente a otros servicios, lo que permite a las aplicaciones adaptarse de manera más rápida a los cambios en los requisitos del negocio.

Y por último, se mejora la resiliencia separando responsabilidades, se asegura que en un microservicio caído no afecte al funcionamiento del sistema completo.

En cuanto a las desventajas, destacaremos una mayor complejidad, un mayor coste económico y una mayor latencia.

Apreciamos una mayor complejidad en las aplicaciones basadas en microservicios que las aplicaciones tradicionales, ya que requieren la gestión de una gran cantidad de servicios y la comunicación entre ellos.

Además tiene un coste elevado implementar y mantener una arquitectura basada en microservicios que una arquitectura tradicional debido a la necesidad de más infraestructura y herramientas de gestión.

Y por último se produce una mayor latencia debido a que los servicios se comunican entre sí a través de una interfaz estándar.

En general, usar microservicios es una gran ventaja cuando se trata de aplicaciones grandes y complejas que requieren escalabilidad, flexibilidad y adaptabilidad.

Sin embargo, debe ser considerado cuidadosamente en comparación a una arquitectura monolítica, ya que la complejidad y los costos adicionales pueden no ser aceptables para ciertos casos de uso.

2. Introducción a la nube

a) *Definición*

La nube, también conocida como cloud computing, se refiere al uso de recursos de tecnología de la información a través de Internet. Es un

modelo de servicio que proporciona acceso a una variedad de recursos informáticos, como almacenamiento, procesamiento, aplicaciones y servicios, mediante una conexión a Internet. El modelo de nube se caracteriza por la posibilidad de acceder a los recursos de forma dinámica, según las necesidades del usuario, lo que permite una mayor escalabilidad y flexibilidad en comparación con los modelos tradicionales de tecnología de la información. Además, en general, los costos de la nube son más bajos ya que se comparten entre varios usuarios y no se requiere una inversión inicial en hardware y software.

Hay varios tipos de nubes como la nube pública, privada e híbrida. La nube pública es administrada por compañías de terceros y se ofrecen a cualquier persona que desee utilizarlas, mientras que las nubes privadas son administradas por una organización específica y solo están disponibles para su uso interno. Las nubes híbridas combinan elementos de ambos tipos, brindando a las organizaciones la flexibilidad de utilizar recursos internos y externos según sus necesidades.

b) *Beneficios de desplegar microservicios en la nube*

Los microservicios y el cloud computing tienen una relación estrecha, ya que ambos buscan mejorar la escalabilidad, flexibilidad y eficiencia en el desarrollo y despliegue de aplicaciones.

Por otro lado, el cloud computing ofrece una plataforma para implementar y ejecutar estos servicios de forma eficiente. Los proveedores de nube ofrecen una amplia variedad de servicios, como almacenamiento, bases de datos y procesamiento, que pueden ser utilizados por los microservicios. Además, el cloud computing permite un despliegue automatizado de los servicios, lo que facilita el proceso de implementación y escalabilidad.

Podemos resumir los beneficios de desplegar microservicios en la nube en los siguientes puntos

1. **Escalabilidad:** Los microservicios se pueden escalar de forma independiente, lo que permite una mayor flexibilidad para adaptarse a los cambios en el tráfico y la demanda.
2. **Flexibilidad:** Al desplegar microservicios en la nube, se pueden utilizar diferentes tecnologías y plataformas, lo que permite una mayor flexibilidad para adaptarse a los cambios en el negocio.

3. **Agilidad:** Los microservicios permiten un desarrollo y despliegue más rápido, lo que facilita la capacidad de la organización para responder rápidamente a los cambios del mercado.
4. **Fiabilidad:** Los microservicios en la nube se pueden configurar para automáticamente escalar y recuperar servicios en caso de fallos, lo que aumenta la disponibilidad y la resiliencia del sistema.
5. **Costos:** Al utilizar la nube, las organizaciones no tienen que invertir en infraestructura costosa y pueden pagar solo por los recursos que utilizan, lo que reduce los costos a largo plazo.
6. **Mejora en la seguridad:** Al desplegar microservicios en la nube, se pueden aprovechar las características de seguridad y cumplimiento de los proveedores de nube.
7. **Mayor capacidad de innovación:** Al utilizar microservicios en la nube, las organizaciones pueden experimentar con nuevas tecnologías y actualizar rápidamente sus sistemas, lo que les permite innovar con más facilidad.

c) *Proveedores de cloud computing*

Los principales proveedores de cloud computing son:

Amazon Web Services (AWS): Es el proveedor de cloud computing más grande y popular del mundo, ofreciendo una amplia variedad de servicios, como almacenamiento, bases de datos, procesamiento y servicios de inteligencia artificial.



Microsoft Azure: Es el segundo proveedor de cloud computing más grande del mundo, ofreciendo servicios similares a AWS, con un énfasis especial en la integración con otras soluciones de Microsoft.

Google Cloud Platform (GCP): Es el tercer proveedor de cloud computing más grande del mundo, ofreciendo servicios de almacenamiento, bases de datos, procesamiento y servicios de inteligencia artificial, con un enfoque en la escalabilidad y rendimiento.



Alibaba Cloud: Es el principal proveedor de cloud computing de China, ofreciendo servicios similares a los de AWS, Azure y GCP.



IBM Cloud: Es un proveedor de cloud computing con un enfoque en la seguridad y el cumplimiento, ofreciendo servicios de almacenamiento, bases de datos, procesamiento y servicios de inteligencia artificial.



Al momento de hablar de conceptos como microservicios, arquitecturas distribuidas y cloud computing, merece la pena pararse a observar la historia de la relación entre AWS y Netflix.

Netflix y Amazon Web Services tienen una historia estrechamente relacionada, ya que Netflix ha sido uno de los principales clientes de AWS desde el inicio de su adopción de la nube.

En el año 2007, Netflix comenzó a experimentar con la nube de AWS para alojar algunos de sus servicios. A medida que Netflix se expandió y su tráfico en línea creció, se dio cuenta de que la nube de AWS podría escalar fácilmente para satisfacer sus necesidades. En 2010, Netflix decidió migrar

toda su infraestructura a AWS y utilizarlo como su plataforma principal para el alojamiento de sus servicios en línea.

La relación entre Netflix y AWS ha sido mutuamente beneficiosa. Netflix ha sido capaz de escalar su infraestructura para manejar su creciente tráfico y ha logrado reducir significativamente sus costos de infraestructura. Por otro lado, AWS ha obtenido un gran cliente en Netflix y ha sido capaz de mostrar cómo su plataforma puede manejar la escala y el tráfico de una gran empresa de entretenimiento en línea.

Netflix también ha contribuido al ecosistema de AWS al desarrollar y compartir herramientas y soluciones de código abierto con la comunidad de AWS, como el sistema de auto-regulación de la capacidad de sus servidores, denominado Chaos Monkey.

La propia web de AWS tiene varios posts muy interesantes en su web hablando de la [arquitectura distribuida](#) de Netflix en AWS. Asimismo, se recomienda echar un vistazo al perfil en Medium de [Netflix Technology Blog](#) (uno de los más leídos en la plataforma), donde mantienen al día a sus lectores sobre la tecnología que crean.

3. Azure functions

a) *Introducción*

Azure Functions es una plataforma de Azure para crear aplicaciones serverless, que se ejecutan como un servicio en la nube y se escalan automáticamente según la demanda. Es una solución para ejecutar pequeñas piezas de código (llamadas funciones) sin tener que preocuparse por la configuración de servidores ni de infraestructura subyacente. Las funciones pueden ser desarrolladas en varios lenguajes de programación, como C#, F#, JavaScript, Java, y Python. Estas funciones pueden ser desencadenadas por una variedad de eventos, tales como, mensajes de una cola, actualizaciones en un documento en una base de datos, o una petición HTTP. El precio es solo por uso, sin costos por tiempo de inactividad.

b) *Como como azure functions se relaciona con los microservicios*

En cuanto a cómo se relaciona Azure Functions con los microservicios, en primer lugar, ambos enfoques se centran en la idea de dividir una aplicación en pequeños componentes que se pueden desarrollar, probar y escalar de manera independiente. Esto es especialmente importante en aplicaciones de gran escala o complejas, ya que permite trabajar en partes

específicas de la aplicación de manera aislada, lo que facilita el desarrollo, el testing y el mantenimiento.

En segundo lugar, Azure Functions es una plataforma adecuada para el desarrollo de microservicios, ya que permite crear y ejecutar pequeñas piezas de código que se pueden desencadenar por diferentes eventos, lo que es esencial para la arquitectura de microservicios. Estos eventos pueden ser, por ejemplo, un nuevo mensaje en una cola de mensajería, una actualización en una base de datos o una petición HTTP. Al tener estos desencadenadores, las funciones se pueden ejecutar de manera automática y escalar según la demanda, lo que es clave en un enfoque de microservicios.

Ejemplos de uso de Azure Functions para implementar Microservicios:

Hay varios ejemplos de cómo se pueden usar Azure Functions para implementar microservicios. Algunos de ellos son:

- **Procesamiento de datos en tiempo real:** se pueden crear funciones que se desencadenan por eventos, como la llegada de un nuevo mensaje a una cola, para procesar los datos y almacenarlos en una base de datos.
- **Integración con redes sociales:** se pueden crear funciones que se desencadenan por eventos, como la publicación de un nuevo mensaje en una red social, para realizar tareas como enviar notificaciones, analizar datos o generar estadísticas.
- **Procesamiento de imágenes:** se pueden crear funciones que se desencadenan por eventos, como la subida de una nueva imagen a un almacenamiento, para procesar las imágenes, como recortarlas, escalarlas o aplicar filtros.
- **Servicios de chatbot:** Se pueden crear funciones que se desencadenan por eventos, como la entrada de un usuario a un chatbot, para procesar las preguntas y responder automáticamente.
- **Análisis de datos de IoT:** Se pueden crear funciones que se desencadenan por eventos, como la llegada de nuevos datos desde dispositivos IoT, para analizar los datos y tomar decisiones automatizadas en base a ellos.

4. Desarrollo de Microservicios con Azure Functions (video demo)

Hemos realizado un vídeo donde se explica cómo desarrollar un proyecto local de Azure Functions y sus principales características en C#.

https://youtu.be/C4Rajr_oohA

5. Referencias

<https://www.chakray.com/es/que-son-los-microservicios-definicion-caracteristicas-y-ventajas-y-desventajas/>

<https://www.redhat.com/es/topics/microservices>

<https://azure.microsoft.com/en-us/products/functions/>

<https://learn.microsoft.com/es-es/azure/azure-functions/>

<https://learn.microsoft.com/es-es/dotnet/architecture/microservices/multi-container-microservice-net-applications/microservice-application-design>

<https://www.divante.com/blog/10-companies-that-implemented-the-microservice-architecture-and-paved-the-way-for-others>

<https://aws.amazon.com/es/solutions/case-studies/netflix/>

<https://www.youtube.com/watch?v=WDDkLOT8SCk&list=PLhr1KZpdzukdeX8mQ2qO73bg6UKQHYsHb&index=12>