Azure Service Fabric

# Estudio

## Introducción:

Service Fabric es una plataforma de empaquetado, administración e implementación de microservicios y contenedores; la creación de clústeres está disponible tanto en Windows server como en Linux.

Service Fabric se basa en clústeres, que pueden contener hasta miles de maquinas, cada maquina es un nodo, que contiene tres aplicaciones: Fabric.exe, Fabric.exe y FrabicGateway.exe. Estos tres programas se encargan de gestionar toda la información y funcionamiento del nodo.

Una característica importante de la plataforma es la importancia que se le da a los servicios con estado, esto es por dos motivos principales:

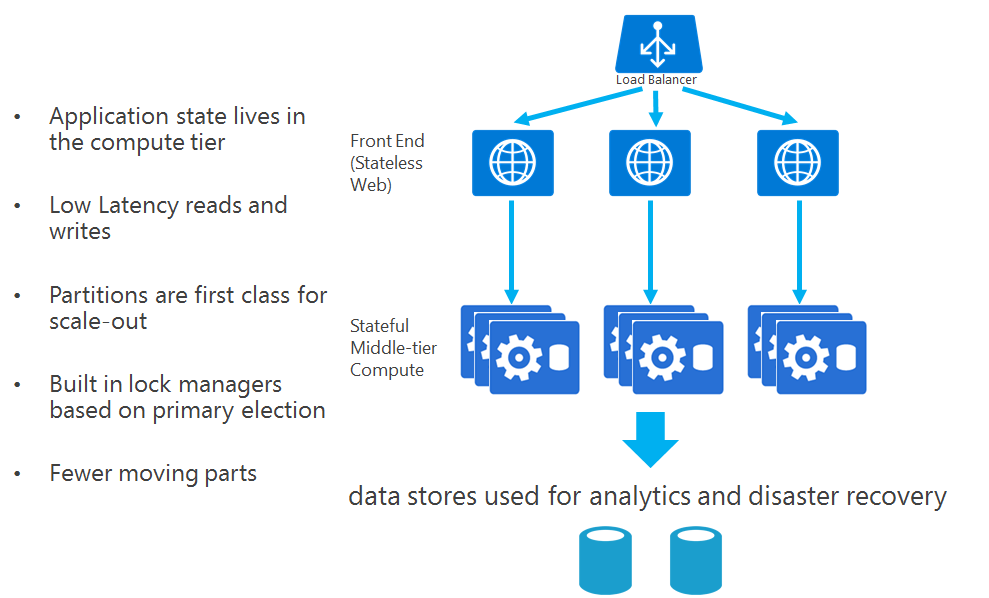
* Se pueden crear servicios de procesamiento de transacciones en línea que sean tolerantes a errores, con poca latencia entre cliente y servidor y un alto rendimiento a la hora de mantener código y datos en una sola maquina.
* Y para poder simplificar el diseño de la aplicación ya que los microservicios con estado eliminan la necesidad de colas y caches adicionales, y ya que estas características son las que más memoria, latencia y disponibilidad ocupan tradicionalmente en una aplicación sin estado.

Los servicios basados en esta plataforma pueden estar diseñados y escritos en varios modelos de programación distintos como por ejemplo:

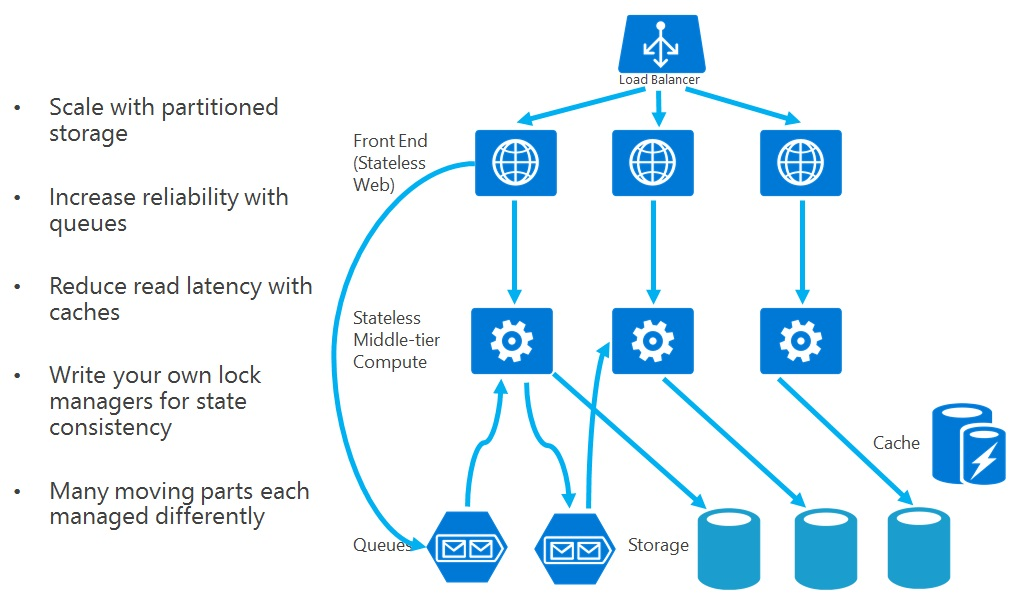
* **Ejecutables de invitado**, que pueden estar escritos en cualquier lenguaje de programación, pero no podrán acceder a las utilidades del API de Service Fabric.
* **Contenedores**, que son la forma predeterminada de ejecutar los servicios, puede implementarse cualquier repositorio de contenedor al en la maquina siempre que sea compatible con Linux o Windows Server 2016 o posterior.
* **Reliable services**, es un framework integrado con Service Fabri que utiliza minimamente la API, lo cual lo hace increíblemente ligero, prefiriendo dejar a la plataforma regular el ciclo de vida del servicio, además de dar una amplia capacidad de diseño del servicio.
* **ASP.NET Core**, marco de código abierto multiplataforma, diseñado con aplicaciones basadas en la nube como back-ends móviles y aplicaciones IoT.
* **Reliable Actors**, marco basado en Realiable Services, es un marco de aplicaciones que implementa el modelo de actor computacional. Usando unidades independientes del proceso y el estado llamadas actores. Este marco proporciona a los actores comunicación integrada y configuración de escalado horizontal. Está totalmente integrado con la plataforma Service Fabric al estar basado en Reliable Services.

## Escenarios de aplicación:

Aunque la compilación de aplicaciones con roles de trabajo nativa a Azure es un servicio sin estado se puede trabajar con microservicios con estado más allá de la solicitud y la respuesta. Estos servicios con estado permiten a todo tipo de aplicaciones tener alta disponibilidad. También se permite una mezcla, teniendo servicios son estado (ASP.NET o Node.js por ejemplo) que llamen a servicios de nivel intermedio con estado.

La arquitectura que Service Fabric implementa permite realizar un análisis de los datos de cualquier aplicación (tenga estado o no) casi en tiempo real, calculo en memoria, transacciones paralelas y procesamiento de eventos en sus aplicaciones. Tambien permite escalar o reducir verticalmente las aplicaciones ya que dispone de requisitos de recursos variables.

Las aplicaciones que mejor aprovechan las características de Service Fabric:

* Compilación de datos, procesamiento e IoT: por su baja latencia y los servicios de estado a gran escala.
* Aplicaciones interactivas con sesión: si una aplicación requiere lecturas y escrituras de baja latencia como juegos en línea o mensajería pueden aprovechar la posibilidad de crear aplicaciones con estado sin requerimientos de almacén o memoria caché independientes.
* Análisis de flujo de datos: si una aplicación requiere procesar de forma confiable eventos o flujos de datos es muy importante que haya poco tiempo de espera en operaciones de lectura y escritura que en Service Fabric están optimizadas. Tambien se benefician de las canalizaciones de procesamiento de aplicaciones en las que los resultados han de pasar sin perdidas, como en sistemas financieros.
* Calculo de datos: al no ser necesario un sistema cache elimina la latencia de red asociada con recuperar los datos de un almacenamiento externo.
* Servicios de alta disponibilidad: mediante la creación de replicas de servicio secundarias podemos disponer de nodos, procesos y servicios redundantes ante fallos, pudiendo elevar una replica de secundaria a principal cuando un error ocurra sin perder apenas servicio.
* Servicios escalables, permitiendo el escalado horizontal de los servicios mediante particiones de cada servicio individual.

# Implementación

Primero hemos de estar subscritos a Azure y conectarnos a ello mediante Azure Powershell. Posteriormente creamos un grupo de recursos de cluster administrado por Service Fabric:

Graphical user interface, text

Description automatically generated

Para crear un cluster administrad ode Sevice Fabric usamos el comando New-AzServiceFabricManagedCluster de Azure Powershell que requiere los siguiente parámetros:

* Nombre del cluster, que ha de ser un nombre único.
* Contraseña de admin.
* Huella digital del certificado de cliente, puede trabajarse con un certificado autofirmado.
* SKU de cluster, el tipo de cluster que se quiere implementar.

$clusterName = "<unique cluster name>"

$password = "Password4321!@#" | ConvertTo-SecureString -AsPlainText -Force

$clientThumbprint = "<certificate thumbprint>"

$clusterSku = "Standard"

New-AzServiceFabricManagedCluster -ResourceGroupName $resourceGroup -Location $location -ClusterName $clusterName -ClientCertThumbprint $clientThumbprint -ClientCertIsAdmin -AdminPassword $password -Sku $clusterSKU -Verbose

Una vez creado el cluster podemos incorporar nuevos nodos principales mediante el comando New-AzServiceFabricManagedNodeType que tiene como parámetros el nombre del tipo de nodo, el recuerso y cluster al que se va a añadir el nodo y el número de instancias que se crearán.

Como insertar los nodos de uno en uno podemos usar el comando Set-AzServiceFabricManagedNodeType para escalar más o menos instancias en un clúster. Este nuevo clúster empezará a ejecutarse de inmediato en unos minutos podemos ver los nuevos nodos creados.

Por ultimo debemos implementar nuestra aplicación. Para ello primero hemos de conectarnos al clúster mediante la huella digital del certificado del clúster. Podemos conectarnos al clúster con el comando Conect-ServiceFabricCluster y el endpoint al que queremos conectarnos.

Text

Description automatically generated

Una vez establecida la conexión podemos continuar, teniendo ya una aplicación debidamente empaquetada, guardando la ruta de acceso donde ha quedado empaquetada ya que se utilizará para la ruta de acceso en el siguiente comando:

Graphical user interface, text

Description automatically generated

Para eliminar la aplicación tan solo hemos de usar el comando Remove-ServiceFabricApplication.

# Comparativa AWS/Google Cloud

Aunque la implementación es ligeramente distinta, Amazon Web Services permite la integración completa con Service Fabric, sin necesidad de cambiar las llamadas de API ni la lógica del sistema.