

Impulsar la continuidad de los negocios con SQL Server



Impulsar la continuidad de los negocios con SQL Server

Contenido

01

Actualmente, todas las empresas necesitan minimizar el tiempo de inactividad

02

Soporte de SQL Server para HADR

03

Instancias de clúster de conmutación por error

04

Grupos de disponibilidad

05

Conmutación por error y recuperación ante desastres

06

Conclusión

© 2018 Microsoft Corporation. Todos los derechos reservados. Este documento se proporciona "tal cual". La información y las opiniones que aquí se expresan, incluidas las direcciones URL y otras referencias a sitios web de Internet, están sujetas a cambios sin previo aviso. Usted asume el riesgo de usarlo.

Este documento no le otorga derecho legal alguno a ningún aspecto de propiedad intelectual de ninguno de los productos de Microsoft. Puede copiar y usar este documento para uso interno como material de consulta.

Acerca de este libro

En este libro se utilizan con frecuencia determinados términos comunes y específicos del sector en el contexto de plataformas de bases de datos. Para el propósito de nuestro análisis, se usan las siguientes definiciones:

Un servidor es una máquina física o virtual que hospeda software de SQL Server junto con un sistema operativo.

Un clúster es un conjunto de servidores agrupados para proporcionar redundancia. El administrador de clústeres supervisa el estado y la capacidad de respuesta de los miembros del clúster (nodos). En Windows Server, el administrador se denomina WSFC ([administrador del] clúster de conmutación por error de Windows Server). Los distribuidores de Linux usan Pacemaker como administrador de clústeres. Un clúster se administra en el nivel del sistema operativo.

Un nodo es un miembro de un clúster de conmutación por error. Cada nodo del clúster tiene la misma capacidad para dar respuesta a solicitudes y tiene acceso a los mismos datos que el resto de nodos del clúster. Cada nodo debe estar disponible para responder a solicitudes de manera oportuna para que el administrador de clústeres lo considere "correcto".

Una base de datos es una estructura de datos que almacena datos de aplicaciones y metadatos en archivos físicos en disco.

Una instancia es una colección de bases de datos de SQL Server, trabajos, entre otros, ejecutada por un solo servicio de SQL Server que se ejecuta en la memoria de un equipo específico en un momento dado. Es posible tener acceso a una instancia desde una dirección IP única y todas las solicitudes se envían a esa IP. La fuente de la solicitud no sabe dónde está la ubicación física de la respuesta que recibe.

Una réplica (o réplica de bases de datos) es un conjunto de dos o más instancias de SQL Server agrupadas en una configuración de conmutación por error. En otras palabras, una réplica es, en el nivel de base de datos, lo que un clúster es en el nivel de sistema operativo.

¿Quiénes deben leer este eBook?

Este eBook está destinado a arquitectos de bases de datos, administradores y desarrolladores que buscan mantener el tiempo de actividad de las aplicaciones críticas. La lectura de este eBook le enseñará cómo SQL Server puede ayudarlo a lograr los objetivos de alta disponibilidad y recuperación ante desastres mediante la tecnología de creación de reflejos y de agrupación en clústeres integrada con características en las implementaciones basadas en Linux y Windows. En este eBook se analizan herramientas y características como las instancias del clúster de conmutación por error Always On, los grupos de disponibilidad Always On o el trasvase de registros. Además, se proporciona información técnica sobre cómo poner en práctica estas capacidades.

Vivimos en un mundo impulsado por los datos y en el que las empresas y los usuarios finales consumen y generan información de muchas maneras.

Esperan poder tener acceso rápido y permanente a los datos dondequiera que estén. Debido a esta mayor demanda de datos y la existencia de una economía más global, una preocupación importante de las organizaciones es garantizar un acceso rápido e ininterrumpido a los datos. La inaccesibilidad afecta de manera considerable la capacidad de ofrecer servicios a los clientes o de llevar a cabo los procesos cotidianos.

Mantener el tiempo de actividad de las aplicaciones críticas resulta indispensable en el mundo actual, un mundo que siempre está en línea. La manera de lograrlo es minimizar el tiempo de inactividad imprevisto. El tiempo de inactividad puede deberse a varias causas: actualizaciones de mantenimiento de hardware y software, cortes de red o del suministro eléctrico, error de hardware, infracción de seguridad o ciberataque (debido a ataques informáticos, virus, etc.). Para minimizar el tiempo de inactividad y el impacto en la empresa, necesita una solución de copia de seguridad y recuperación ante desastres; de esta forma, mantendrá la empresa en marcha sin interrupciones o con un mínimo de interrupciones.

En el sector de las tecnologías de la información, todos conocen los conceptos de alta disponibilidad (HA) y recuperación ante desastres (DR). En pocas palabras, la alta disponibilidad se refiere a la implementación de una solución que mantenga funcionando el negocio en caso de que se produzca un error catastrófico. Enmascara los efectos de un error de hardware o software y mantiene la disponibilidad de las aplicaciones, por lo que se minimiza el tiempo de inactividad que perciben los usuarios. La recuperación ante desastres se refiere a la implementación de soluciones que permitan superar la pérdida de datos después de producirse un desastre. Un sistema de alta disponibilidad ayuda a garantizar el acceso a los datos. Sin embargo, también se puede implementar una solución de recuperación ante desastres para mantener la precisión de los datos.

Soluciones de alta disponibilidad y recuperación ante desastres (HADR) para la continuidad empresarial

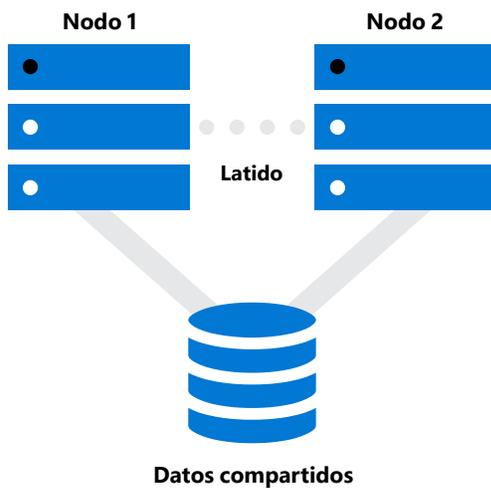
Las dos técnicas más comunes para lograr alta disponibilidad y continuidad empresarial son la creación de reflejos y la agrupación en clústeres. La creación de reflejos es una solución de continuidad en el nivel de base de datos, mientras que la agrupación en clústeres proporciona una solución en el nivel de servidor.

La creación de reflejos de bases de datos es compatible con la conmutación por error casi instantánea mediante el mantenimiento de una base de datos en espera: una copia completa (reflejo) de la base de datos activa en hardware independiente. Puede funcionar en un modo sincrónico (alta seguridad), donde una transacción entrante se compromete con todos los servidores al mismo tiempo, o en un modo asincrónico (alto rendimiento), donde una transacción entrante se compromete con la base de datos activa y, a continuación (en algún momento predeterminado), se copia al reflejo. En otras palabras, cada reflejo mantiene su propia fuente de datos independiente. La creación de reflejos es una solución de nivel de base de datos y solo funciona con bases de datos que usan el modelo de recuperación completa.

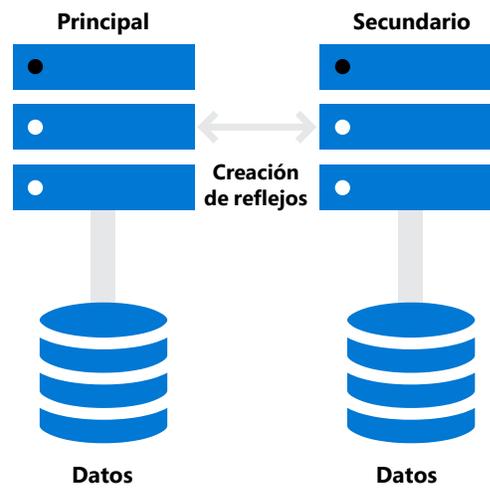
La creación de reflejos está en desuso y se eliminará en una próxima versión de SQL Server. Los grupos de disponibilidad Always On mejoran esta funcionalidad y deben usarse de ahora en adelante.

La agrupación de bases de datos en clústeres es el proceso de combinar más de un servidor (que comparten un solo almacenamiento de datos) en lo que al usuario le parece una sola instancia. Los usuarios se conectan a la instancia y nunca necesitan saber qué servidor de la instancia está actualmente activo. Si un servidor falla o debe desconectarse para realizar tareas de mantenimiento, la experiencia del usuario no cambia. El administrador de clústeres supervisa cada servidor del clúster con un latido, de modo que detecta el momento en que el servidor activo del clúster se desconecta e intenta cambiar sin problemas al siguiente servidor del clúster (aunque hay un retraso de tiempo variable mientras se produce el cambio). Puesto que los datos se comparten entre todos los nodos del clúster, se conserva la continuidad de los datos. ■

Agrupación en clústeres



Creación de reflejos



Microsoft SQL Server está diseñado para admitir HADR, de modo que se garantice que los componentes esenciales de la infraestructura empresarial estén siempre activos, disponibles y puedan sobrevivir sin ningún error.

SQL Server ofrece soluciones para varios escenarios de HADR e incluye un conjunto de características y capacidades que pueden ayudar a que las organizaciones alcancen una amplia gama de SLA de disponibilidad. SQL Server permite disfrutar de más seguridad, fiabilidad, escalabilidad y más opciones para una mejor administración de clústeres y almacenamiento. Integra la creación de reflejos y la tecnología de agrupación en clústeres con las características disponibles en implementaciones basadas en Linux y Windows: instancias del clúster de conmutación por error (FCI) Always On, grupos de disponibilidad Always On y trasvase de registros.

Instancias del clúster de conmutación por error Always On para alta disponibilidad de nivel de instancia

Las FCI son herramientas de alta disponibilidad y recuperación ante desastres que ofrecen continuidad empresarial en el nivel de instancia o servidor. Las FCI ofrecen protección frente a errores del servidor provocados por problemas de redes, hardware, sistema operativo o software. Puesto que la FCI usa una sola dirección de red para recibir solicitudes, el usuario o la aplicación que hacen la solicitud no tendrán que cambiar la información de conexión. Si, por ejemplo, el servidor primario de una FCI se queda sin electricidad, el administrador de clústeres simplemente "promueve" uno de los servidores secundarios del clúster a la posición principal y redistribuye la solicitud a este en su lugar. El miembro del clúster que no responde no recibirá solicitudes hasta que vuelva a estar en línea.

Grupos de disponibilidad Always On para alta disponibilidad de nivel de base de datos

Los grupos de disponibilidad Always On son una solución de nivel empresarial de alta disponibilidad y recuperación ante desastres que permite maximizar la disponibilidad para una o más bases de datos de usuarios. Un grupo de disponibilidad es uno o más grupos de bases de datos de usuarios que conmutan por error juntos a una copia de seguridad. Los grupos de disponibilidad Always On requieren que las instancias de SQL Server residan en nodos de clúster.



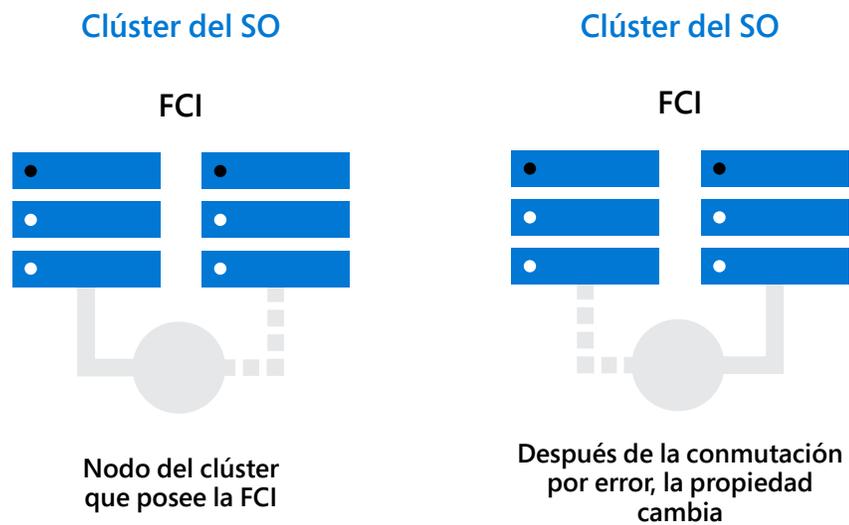
Comparación de la conmutación por error en niveles de instancia y de base de datos

Con la conmutación por error en el nivel de instancia que usa SQL Server, una FCI tiene una dependencia en el almacenamiento compartido. En cambio, la conmutación por error en el nivel de base de datos que usa grupos de disponibilidad Always On no tiene dependencia en el almacenamiento compartido. Hay más diferencias conceptuales entre la conmutación por error en el nivel de instancia y la conmutación por error en el nivel de base de datos, tal como se muestra en la siguiente tabla:

	Nivel de instancia	Nivel de base de datos
Usa un clúster	Sí	Sí
Nivel de protección	Instancia	Base de datos
Tipo de almacenamiento	Compartida	No compartida Si bien las réplicas de un grupo de disponibilidad no comparten almacenamiento, una réplica hospedada por una FCI usa una solución de almacenamiento compartido, según los requisitos de esa FCI. La solución de almacenamiento solo se comparte por nodos en la FCI, no entre réplicas de los grupos de disponibilidad.
Soluciones de almacenamiento	Conectado directamente, SAN, punto de montaje, CIFS	Según el tipo de nodo
Recursos de conmutación por error	Servidor, instancia y base de datos	Solo base de datos

Las FCI son un método eficaz de ofrecer disponibilidad para toda la instalación de SQL Server, lo que se conoce como una instancia.

Un clúster de conmutación por error está compuesto de dos o más copias idénticas de una instancia. Esto significa que en cada nodo existe una copia duplicada de todo el contenido de la instancia, incluidas bases de datos, trabajos del Agente SQL Server y servidores vinculados. Solo puede haber un nodo activo a la vez. Los otros son pasivos. La propiedad de la instancia se transferirá a uno de los servidores pasivos si se produce algún problema en el servidor activo. Todas las FCI requieren algún tipo de almacenamiento compartido, aunque se proporcione a través de redes.

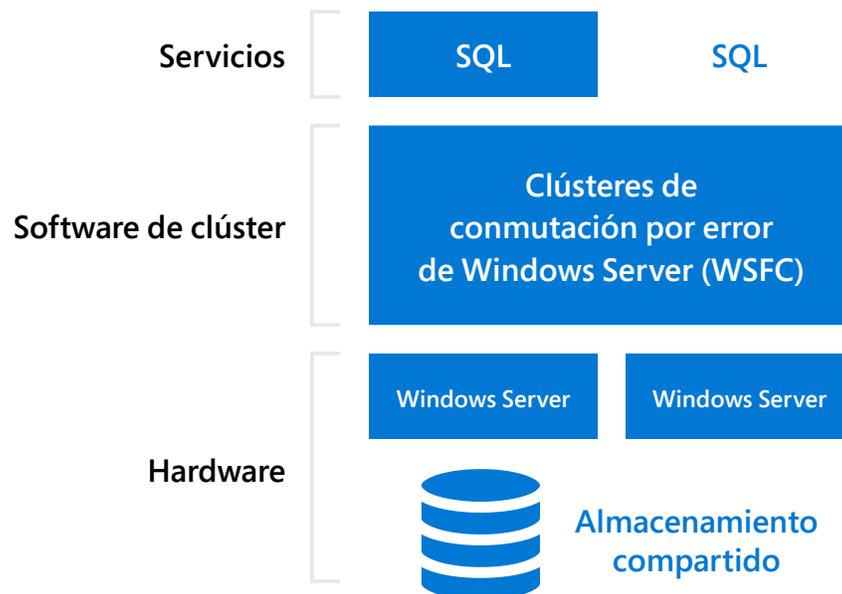


Una FCI necesita que un administrador de clústeres organice los recursos de los clústeres. Las FCI de SQL Server están disponibles en Windows y en Linux. En Windows, las FCI usan los clústeres de conmutación por error de Windows Server (WSFC) como administrador del clúster. Linux usa un administrador del clúster de open source llamado Pacemaker junto con un sistema de comunicaciones de servidor de open source llamado Corosync. Independientemente del sistema operativo que se use, estas herramientas realizan a grandes rasgos la misma función: crear una configuración de clúster activo/pasivo.

Instancias de clúster de conmutación por error: SQL Server en Windows

Las FCI en Windows aprovechan la funcionalidad WSFC para proporcionar alta disponibilidad mediante la redundancia en el nivel de instancia. El WSFC proporciona características de infraestructura que admiten los escenarios HADR de SQL Server. Supervisa y administra los roles actuales de las réplicas de disponibilidad que pertenecen a un determinado grupo de disponibilidad a la vez que determina cómo un caso de conmutación por error afecta las réplicas de disponibilidad. Se crea un grupo de recursos WSFC por cada grupo de disponibilidad que cree. El WSFC supervisa este grupo de recursos para evaluar el estado de la réplica principal.

Con la FCI, siempre hay un nodo maestro o activo en el clúster. Los demás nodos pasan a ser nodos secundarios o pasivos. Una FCI de SQL Server se ejecuta en un grupo de recursos WSFC. Cada nodo del grupo de recursos mantiene una copia sincronizada de los ajustes de configuración y las claves de registro controladas para garantizar la plena funcionalidad de la FCI después de una conmutación por error. Solo uno de los nodos del clúster posee el grupo de recursos cada vez. En caso de conmutación por error, la propiedad del grupo de recursos se mueve a otro nodo WSFC. Este proceso es transparente para el cliente o la aplicación que se conecta a SQL Server. Así se minimiza el tipo de inactividad que la aplicación o el cliente experimentan durante una conmutación por error.



Servidores unidos al dominio

En las versiones anteriores de Windows Server, la creación de la FCI en WSFC necesitaba Active Directory. La implementación de un WSFC mediante Windows Server siempre ha necesitado que los nodos que participan en un WSFC estén unidos al mismo dominio de Active Directory. Cuando se crea un WSFC, un objeto de nombre de clúster (CNO) o una cuenta y un objeto de equipo virtual (VCO) se generan en Active Directory. Sin embargo, esta dependencia entre un WSFC y Active Directory fue el principal desafío y obstáculo en la implementación y administración de FCI. Para solucionar este problema, SQL Server introdujo clústeres de conmutación por error independientes del dominio que permiten que los administradores implementen un WSFC sin un dominio de Active Directory.

Configuraciones de almacenamiento de WSFC

Una FCI debe usar almacenamiento compartido entre todos los nodos de la FCI para el almacenamiento de bases de datos y registros. Cada nodo de la FCI en WSFC necesita almacenamiento compartido, según la instalación de la instancia del clúster de conmutación por error de SQL Server estándar. El almacenamiento puede usar canal de fibra, iSCSI, FCoE o SAS para el almacenamiento de disco compartido o almacenamiento conectado localmente con espacios de almacenamiento directo (S2D). Con este almacenamiento compartido, todos los nodos de la FCI tienen la misma vista de los datos de la instancia cuando se produce una conmutación por error. El almacenamiento compartido tiene el potencial de ser el único punto de error. La FCI depende de la solución de almacenamiento subyacente para garantizar la protección de los datos.

Conmutación por error de WSFC

Cuando la FCI arranca, uno de los nodos asume la propiedad del grupo de recursos y pone en línea la instancia de SQL Server. Algunos de los recursos que posee este nodo son el nombre de red, la dirección IP, los discos compartidos, el servicio de motor de base de datos de SQL Server y el servicio del Agente SQL Server. En cualquier momento dado, solo el propietario del grupo de recursos (y ningún otro nodo de la FCI) ejecuta sus respectivos servicios de SQL Server en el grupo de recursos. En caso de conmutación por error, el servicio WSFC transfiere la propiedad de los recursos de la instancia a un nodo de conmutación por error designado. Después, la instancia de SQL Server se reinicia en el nodo de conmutación por error y las bases de datos se recuperan como de costumbre. En cualquier momento dado, solo un único nodo del clúster puede alojar la FCI y los recursos subyacentes.

Capa de cuórum de WSFC

Cada nodo de un WSFC participa en una comunicación periódica de latido para compartir el estado de mantenimiento del nodo con los demás nodos. Los nodos que están en buen estado se consideran "activos" y los nodos que no responden se consideran "con error". El cuórum es un mecanismo que garantiza que el WSFC está funcionando con suficientes recursos que responden en línea. Si el WSFC tiene suficiente miembros "que votan" (activos), está en buen estado y es capaz de brindar tolerancia a errores en el nivel de nodo.

Un modo de cuórum dicta la metodología que se usa para la votación del cuórum y cuándo hay que realizar una conmutación por error automática o desconectar el clúster. Si el WSFC se desconecta (ya sea por un desastre no planificado, un problema persistente de hardware o como resultado de un error en las comunicaciones), es necesaria una intervención manual. Un administrador tendrá que forzar un cuórum y después volver a conectar los nodos supervivientes del clúster en una configuración tolerante a errores.

Instancias de clúster de conmutación por error: SQL Server en Linux

En Linux, los clústeres de conmutación por error funcionan igual que en SQL Server pero están estructurados de manera ligeramente diferente. En lugar de haber un solo administrador de WSFC, SQL Server en Linux usa dos componentes que funcionan en conjunto para realizar funciones de WSFC: Pacemaker y Corosync.

Linux es un sistema operativo basado en paquetes donde los servicios del sistema se instalan de manera individual o en grupos para proporcionar distintas funciones. Las tres distribuciones de Linux compatibles, que son Red Hat Enterprise Linux (RHEL), Ubuntu y SUSE Linux Enterprise Server (SLES), incluyen funciones de alta disponibilidad que incorporan Pacemaker y Corosync. Pacemaker es el cerebro de la operación: supervisa los miembros del clúster, administra los recursos del clúster y toma decisiones sobre cuáles son los nodos que deben estar activos según lo estipulado en la directiva. Corosync proporciona una comunicación bidireccional dedicada entre Pacemaker y los nodos que administra.

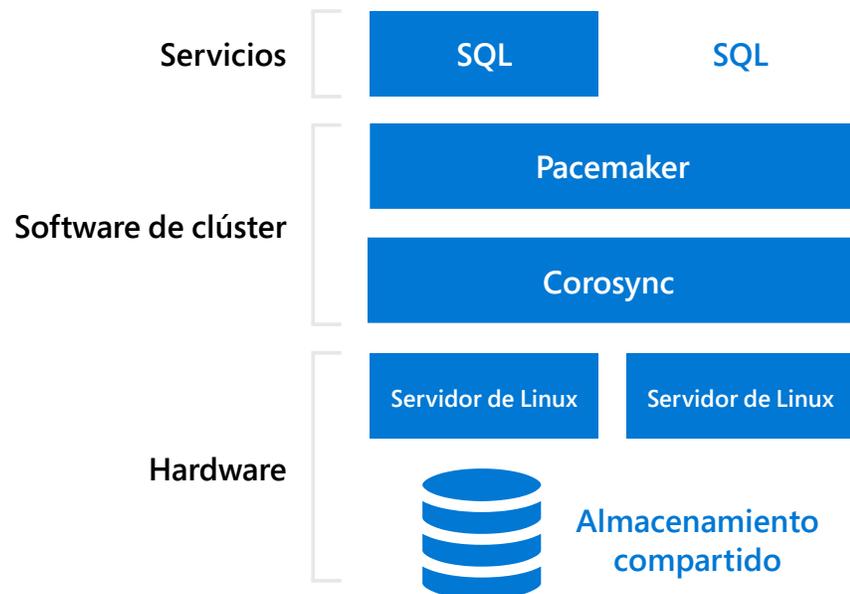
Del mismo modo que ocurre con los clústeres de conmutación por error de Windows Server, los clústeres de conmutación por error de Linux proporcionan continuidad en el nivel de instancia, conmutación por error automática y manual, conmutación por error transparente para aplicaciones y clientes y recuperación en cuestión de minutos o incluso segundos.

Opciones de almacenamiento compartido de clúster

Todas las FCI requieren algún tipo de almacenamiento compartido. Este almacenamiento se presenta a todos los servidores que pueden hospedar la FCI, pero solo un único servidor puede usar el almacenamiento para la FCI en un momento dado. Algunas de las opciones disponibles de almacenamiento compartido con Linux son iSCSI, Network File System (NFS) y [Common Internet File System \(CIFS\)](#). Como con cualquier recurso de almacenamiento en Linux, el almacenamiento de datos compartido se monta en cada uno de los nodos del clúster de la FCI.

Conmutación por error de Pacemaker

En un clúster administrado por Pacemaker, como en un clúster administrado por WSFC, uno de los nodos se considera activo mientras los otros están inactivos. Tan solo el nodo activo tiene permiso para conectarse a los recursos de almacenamiento compartido. Cuando Pacemaker detecta un error, cambia el nodo activo a un miembro del clúster correcto y diferente, y elimina el nodo con error del grupo de miembros de clúster disponibles. Este proceso se denomina barrera (se comenta más detenidamente a continuación en la sección [Barrera y STONITH](#)).



Cuórum de Pacemaker

Cuórum es un término genérico de alta disponibilidad que Pacemaker utiliza de manera similar que WSFC. Cuando más de la mitad de los nodos previstos están en línea, se dice que un clúster tiene cuórum. Esto es importante porque la configuración del cuórum determina el número de nodos fallidos que un clúster puede tener y permanecer aún en línea. En Linux, la capa de comunicación Corosync administra el sondeo de cuórum y notifica los cambios de estado a Pacemaker. Cuando se pierde el cuórum, Pacemaker tiene la capacidad de hacer caso omiso del cuórum y mantener los servicios de clúster con un solo nodo hasta que se restablezca el cuórum. Esto quiere decir que los servicios siguen disponibles para los usuarios. Sin embargo, un clúster con un solo nodo no tiene verdadera alta disponibilidad. Por lo tanto, Pacemaker seguirá intentando poner los nodos de nuevo en línea y restablecer el cuórum.

Barrera y STONITH

Los administradores de recursos de clústeres (WSFC en Windows y Pacemaker en Linux) supervisan el estado de los nodos del clúster y administran cualquier problema que surja. En algunas circunstancias, puede ser necesario o prudente poner en cuarentena un nodo con un comportamiento erróneo o que no se comunica. Esta puesta en cuarentena, en términos de alta disponibilidad, se denomina "barrera". Esto es, básicamente, una manera de controlar el acceso a los recursos de un clúster. Puede utilizarse en el nivel de recurso (para bloquear el acceso a determinados servicios como un servidor web o servidor de bases de datos) o en el nivel de nodo (para evitar que un nodo de clúster se vuelva "activo"). La barrera puede proteger al clúster de un nodo o servicio con un comportamiento erróneo o permitirle que marque un servicio o nodo sin conexión para mantenimiento, aunque se comporte con normalidad.

La retirada de nodos con un comportamiento erróneo del clúster (barrera en el nivel de nodo) se denomina STONITH, que significa "Shoot the Other Node in the Head" (Disparar al otro nodo en la cabeza) o, en otras palabras, apagarlo ahora. Aunque hay muchas maneras de hacer esto que dependen del hardware del servidor y el presupuesto, cada método apaga de inmediato un nodo, normalmente al cortando de forma repentina la alimentación. El STONITH se utiliza para garantizar que un nodo no pueda volver a conectarse sin la intervención de una persona e indica al administrador de recursos de clúster (WSFC en Windows o Pacemaker en Linux) que deje de intentar corregirlo.

Tiempo de conmutación por error predecible

Según el momento en que la instancia de SQL Server realizó la última operación de punto de control, es posible que haya una cantidad considerable de páginas desfasadas en la memoria caché del búfer. En consecuencia, las conmutaciones por error duran el tiempo que sea necesario para escribir las páginas desfasadas restantes en disco. Esto puede dar lugar a un tiempo de conmutación por error largo e impredecible.

La FCI puede usar puntos de control indirectos para limitar el número de páginas desfasadas en la memoria caché del búfer. Aunque esto no consume más recursos bajo la carga de trabajo habitual, hace que el tiempo de conmutación por error sea más predecible y configurable. Esto resulta muy útil cuando el acuerdo de nivel de servicio de la organización especifica el objetivo de tiempo de recuperación (RTO) para la solución de alta disponibilidad.

Comparación de FCI de SQL Server en Windows y Linux

Estas son las principales diferencias entre FCI en Windows y FCI en Linux:

Linux únicamente admite una sola instalación de SQL Server por servidor, por lo que todas las FCI de Linux serán una instancia de un solo nodo por defecto. Si se necesitan varias instancias en Linux, se recomienda el uso de contenedores. Windows admite más de una instalación de SQL Server por servidor. El número de instalaciones depende de la versión de Windows que esté ejecutando, por lo que debe comprobar los límites del SO.

El número de FCI que es posible controlar depende del clúster del SO.

Windows admite un máximo de 25 FCI por WSFC. Un clúster de Pacemaker solo puede tener un máximo de 16 FCI, ya que cada nodo es una sola instancia.

Una FCI implementada en Linux con la edición estándar de SQL Server admite un máximo de dos nodos por clúster, aun cuando Pacemaker tenga la capacidad de admitir hasta 16 nodos.

Clústeres de conmutación por error en contenedores

Los clústeres de conmutación por error que usan contenedores en clúster son otra manera de proporcionar alta disponibilidad en ejecución en Linux en un entorno de nube. Los contenedores ofrecen una implementación ágil: es rápido y sencillo lanzar contenedores y volver a apagarlos cuando ya no son necesarios. En comparación con las máquinas virtuales, que tienen la sobrecarga adicional de un sistema operativo completo, los contenedores se ejecutan sobre un sistema operativo, de modo que puede compartimentar ciertas aplicaciones o servicios fuera de ellos. En general, los contenedores usan un script de configuración que se denomina imagen, es decir, una plantilla que define lo que incluye el contenedor. Si se produce un fallo, el administrador de clústeres puede simplemente usar la imagen para implementar un nuevo contenedor (por lo general, en cosa de segundos).

Azure Kubernetes Service (AKS), con un almacenamiento persistente para alta disponibilidad, es un ejemplo del uso de contenedores en un entorno de alta disponibilidad. Si la instancia de SQL Server falla en un contenedor de Kubernetes (denominado "pod"), Kubernetes la vuelve a crear automáticamente en un nuevo pod que se conecta al mismo almacenamiento persistente. Según la configuración específica del pod, puede tratarse de una rápida solución de recuperación.

El siguiente ejemplo describe una implementación, incluido un contenedor basado en la imagen del contenedor de SQL Server con réplica:

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: mssql-deployment
spec:
  replicas: 1
  template:
    metadata:
      labels:
        ---
    containers:
      - name: mssql
        image: microsoft/mssql-server-linux
        ports:
          - containerPort: 1433
        env:
          - name: ACCEPT_EULA
            value: "Y"
          - name: SA_PASSWORD
            ---
```

Después, crea la implementación:

```
kubectl apply -f <Path to sqldeployment.yaml file>
```

➔ [Obtenga más información](#) sobre la implementación de Kubernetes en una configuración de conmutación por error.

Los grupos de disponibilidad ofrecen protección de datos en el nivel de base datos en una configuración principal/secundaria (activa/pasiva).

SQL Server utiliza los servicios y capacidades WSFC del SO de Windows para apoyar los grupos de disponibilidad Always On y las FCI Always On. Cada miembro del clúster (nodo) se usa para determinar el cuórum, independientemente de si hospeda una base de datos en el grupo de disponibilidad.

Cada instancia que participa en un grupo de disponibilidad (denominada réplica) puede configurarse como independiente o como parte de una FCI Always On, según cómo se vaya a usar dicha réplica. SQL Server 2017 admite dos arquitecturas diferentes para los grupos de disponibilidad: **Always On** y **escalado de lectura**.

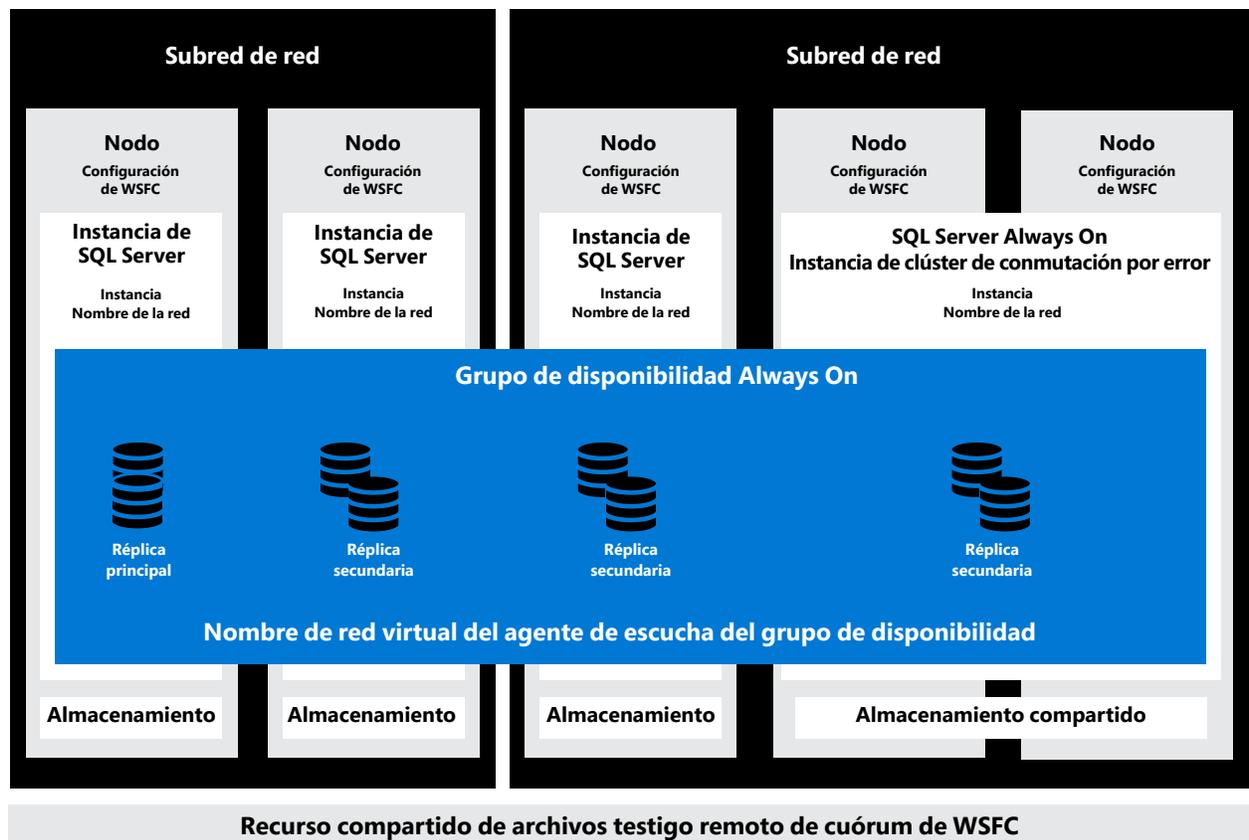
Grupos de disponibilidad Always On: estos grupos de disponibilidad proporcionan alta disponibilidad y recuperación ante desastres de bases de datos. Necesitan un administrador de clústeres de nivel de servidor (WSFC en Windows o Pacemaker en Linux); esto quiere decir que tienen detección de errores, conmutación por error automática y reconexión transparente después de la conmutación por error para proporcionar alta disponibilidad. Esto también requiere que cada réplica resida en su propio nodo de clúster. Las réplicas sincronizadas ofrecen protección de los datos para la recuperación ante desastres oportuna en el nivel de base de datos.

Grupos de disponibilidad de escalado de lectura: estos grupos de disponibilidad ofrecen redundancia de los datos sin ser parte de un clúster de alta disponibilidad. Un grupo de disponibilidad de escalado de lectura es un grupo de bases de datos que se copian a otras instancias de SQL Server para la carga de trabajo de solo lectura. Estas capacidades están disponibles para SQL Server en ejecución en Windows y Linux.

Grupos de disponibilidad con clústeres de conmutación por error

Los grupos de disponibilidad aprovechan la alta disponibilidad de los clústeres de conmutación por error para añadir otra capa de protección contra errores en los nodos de clúster en el nivel de base de datos. Las FCI garantizan que SQL Server responda a todas las solicitudes de datos. Los grupos de disponibilidad aseguran que los datos que se recuperan sean los más exactos.

Linux también admite grupos de disponibilidad con clústeres de conmutación por error. Salvo por algunas pequeñas diferencias de configuración, los grupos de disponibilidad de SQL Server en Linux son iguales que en Windows. Linux usa Pacemaker para controlar los clústeres de conmutación por error y supervisar su estado, mientras que Windows usa WSFC.



Tipo de clúster

Windows tiene tres tipos de clúster: WSFC, External y None. El WSFC es el tipo predeterminado de clúster en Windows Server y el único tipo de clúster que ofrece alta disponibilidad en el nivel de servidor. No obstante, si se desea un entorno de clúster de sistema operativo híbrido, el clúster tiene que ser de tipo None.

Linux tiene dos tipos de clúster: External y None. Con un clúster de tipo External, Pacemaker controla la FCI bajo el grupo de disponibilidad, así como el propio grupo de disponibilidad. Con un clúster de tipo None, Pacemaker no controlará el grupo de disponibilidad, aunque controle la FCI. Si un grupo de disponibilidad tiene un clúster de tipo None, la conmutación por error de una réplica principal a una réplica secundaria tendrá que efectuarse manualmente.

Modo de disponibilidad

Así como los clústeres de conmutación por error tienen una instancia principal e instancias secundarias correspondientes, los grupos de disponibilidad tienen una réplica principal y réplicas secundarias correspondientes. La réplica principal hospeda la base de datos principal (o bases de datos principales) y permite transacciones de lectura/escritura. La réplica principal registra cada transacción y envía ese registro a cada réplica secundaria para mantenerlos sincronizados. Cada réplica secundaria guarda una copia del registro antes de aplicar las transacciones registradas a sus bases de datos; este proceso se denomina "endurecimiento". Esto puede realizarse de manera sincrónica, donde la réplica principal espera a que la réplica secundaria guarde cada registro de transacciones antes de enviar otro, o de manera asincrónica, donde la réplica principal no espera a que la réplica secundaria guarde los registros antes de enviar otro.

El modo de confirmación asincrónica acelera la réplica principal, pero puede provocar que la integridad de los datos de la réplica secundaria tenga un tiempo de retardo mientras trata de ponerse al día con la réplica principal. El modo de confirmación sincrónica puede reducir la velocidad de la réplica principal mientras espera a que la réplica secundaria se ponga al día, pero sincroniza los datos de la réplica secundaria con los de la réplica principal en la mayor medida posible.

Más información:
configuración
del modo de
disponibilidad en
Windows y Linux

El comando `CREATE AVAILABILITY GROUP` de SQL Server en Windows o Linux le permite especificar el modo de disponibilidad:

```
AVAILABILITY_MODE = { {SYNCHRONOUS_COMMIT | ASYNCHRONOUS_COMMIT | CONFIGURATION_ONLY }
```

- `SYNCHRONOUS_COMMIT` especifica que la réplica principal espera la confirmación de las transacciones hasta que se hayan endurecido en esta réplica secundaria (modo de confirmación sincrónica).
 - `ASYNCHRONOUS_COMMIT` especifica que la réplica principal confirma transacciones sin esperar a que esta réplica secundaria endurezca el registro (modo de disponibilidad de confirmación sincrónica).
 - SQL Server 2017 CU 1 introduce `CONFIGURATION_ONLY`. La réplica de `CONFIGURATION_ONLY` solo se aplica a grupos de disponibilidad con `CLUSTER_TYPE = EXTERNAL` o `CLUSTER_TYPE = NONE`. `CONFIGURATION_ONLY` especifica que la réplica principal confirma de manera sincrónica los metadatos de la configuración del grupo de disponibilidad a la base de datos maestra de esta réplica. Esto no es válido cuando `CLUSTER_TYPE = WSFC`. Para obtener más información, consulte la [documentación de Microsoft](#).
- ➔ **Para obtener más información** acerca de estos modos de disponibilidad, visite [Modos de disponibilidad \(Grupos de disponibilidad Always On\)](#).

Modo de conmutación por error

Cuando una réplica principal deja de responder, el WSFC promueve una réplica secundaria al rol principal en un proceso denominado conmutación por error. Las maneras de configurar la conmutación por error dependen del tipo de modo de disponibilidad elegido (sincrónico o asincrónico).

Modo de disponibilidad	Modos de conmutación por error disponibles
Confirmación sincrónica	Conmutación por error manual planificada o conmutación por error automática
Confirmación asincrónica	Conmutación por error manual forzada

En pocas palabras, una conmutación por error manual planificada se inicia mediante un comando ejecutado por un administrador de bases de datos. Este comando ordena al clúster de disponibilidad que se asegure de que cualquier elemento de la réplica principal se sincronice con las réplicas secundarias, promueve una réplica secundaria sincronizada a la posición principal e indica al clúster que se convierta en una réplica secundaria. Una conmutación por error automática realiza casi la misma transición, pero se produce en respuesta a un fallo de la réplica principal. Con una conmutación por error automática, una réplica secundaria que responde y está sincronizada se promueve a la posición de la réplica principal. Cuando la antigua réplica principal vuelve a estar en línea, se designa como una réplica secundaria y se sincroniza. Esta réplica secundaria no está disponible para la conmutación por error hasta que está plenamente sincronizada con la nueva réplica principal.

En el modo de confirmación asincrónica, donde la réplica principal envía registros de transacciones a la réplica secundaria sin esperar a que los registros se guarden correctamente (se endurezcan), la única opción de conmutación por error es una conmutación por error manual forzada. Esta conmutación por error se inicia de inmediato, independientemente de si los registros de transacción endurecidos de dicha réplica secundaria se aplicaron a las bases de datos correspondientes. Una conmutación por error forzada pone en riesgo la integridad de los datos y puede provocar la pérdida de datos.

Más información:
[modo de conmutación por error en Windows](#)

Puede cambiar el modo de conmutación por error de una réplica de disponibilidad definida en un grupo de disponibilidad mediante el uso de SQL Server Management Studio (SSMS), Transact-SQL (T-SQL) o SQL Server PowerShell. En SSMS, es posible utilizar la lista desplegable del modo Conmutación por error en el cuadro de diálogo Propiedades de la réplica de disponibilidad para cambiar el modo de conmutación por error de la réplica.

En T-SQL, puede usar la instrucción [ALTER AVAILABILITY GROUP](#) para cambiar el modo de conmutación por error:

```
ALTER AVAILABILITY GROUP *group_name* MODIFY REPLICA ON
'*server_name*'
WITH ( {
    | FAILOVER_MODE = { AUTOMATIC | MANUAL }
} )
```

➔ [Para obtener más información](#), consulte [Cambiar el modo de disponibilidad de una réplica de disponibilidad \(SQL Server\)](#).

Más información:
[modo de conmutación por error en Linux](#)

Con Linux, cuando `CLUSTER_TYPE` es `EXTERNAL`, el administrador de clústeres externo ejecuta todas las acciones de conmutación por error manual o automática. Por ejemplo, si una solución usa Pacemaker para administrar un clúster de Linux, emplea `pcs` para realizar conmutaciones por error manuales en Red Hat Enterprise Linux (RHEL) o Ubuntu.

```
sudo pcs resource move ag_cluster-master nodeName2 --master
```

En SUSE Linux Enterprise Server (SLES), puede usar `crm`.

```
crm resource migrate ag_cluster nodeName2
```

➔ [Para obtener más información](#) acerca de la conmutación por error de grupos de disponibilidad Always On en Linux, consulte la [documentación de Microsoft sobre la conmutación por error manual y forzada](#).

Agente de escucha en el grupo de disponibilidad

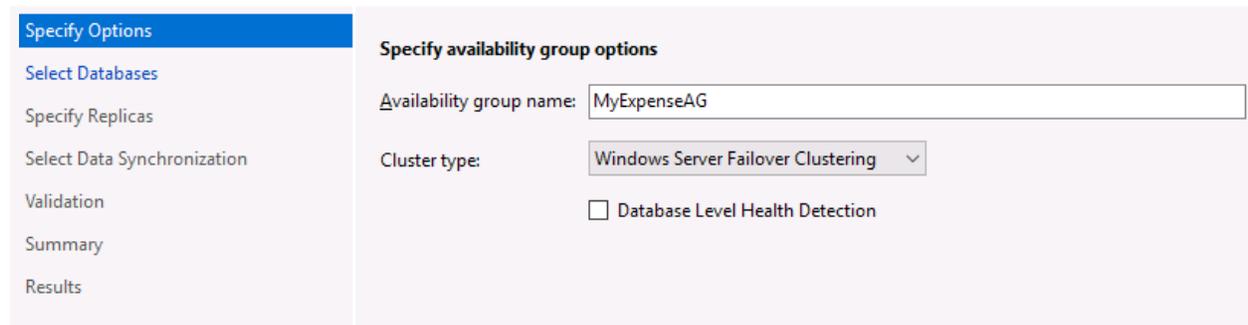
Un grupo de disponibilidad Always On usa un servicio de escucha que permite que las aplicaciones y los usuarios finales puedan conectarse sin necesidad de conocer la instancia de SQL Server que está hospedando la réplica principal. Windows consigue esto mediante el uso de IP virtuales, un nombre de red virtual y la configuración DNS. Cada grupo de disponibilidad tiene su propio agente de escucha, pero solo el nodo de la réplica principal responde a las solicitudes entrantes del cliente para conectarse al nombre de red virtual.

Al igual que en Windows, el agente de escucha es una funcionalidad opcional para un grupo de disponibilidad en Linux. Como una instancia en una FCI, un agente de escucha de un grupo de disponibilidad ofrece un solo punto de conexión de cliente sin requerir el nombre de la instancia. En Linux, hay un recurso de dirección IP que se crea en Pacemaker y puede ejecutarse en cualquier nodo. Se utiliza una entrada asociada con el recurso de IP para el agente de escucha en DNS con un nombre descriptivo. El recurso de IP para el agente de escucha solo estará activo en el servidor que hospeda la réplica principal de ese grupo de disponibilidad.

Más información: configuración de un grupo de disponibilidad en Windows

Para crear y configurar un grupo de disponibilidad en Windows, puede usar cualquiera de las siguientes herramientas en SQL Server Management Studio (SSMS), Transact-SQL (T-SQL) y SQL Server PowerShell:

1. **Asistente para nuevo grupo de disponibilidad.** Este asistente de SSMS permite completar todas las tareas necesarias para crear y configurar un grupo de disponibilidad. Por ejemplo, puede especificar el tipo de clúster como WSFC para Windows:



The screenshot shows the 'Specify Options' step of the SQL Server Availability Group wizard. The left sidebar lists the steps: Specify Options (selected), Select Databases, Specify Replicas, Select Data Synchronization, Validation, Summary, and Results. The main area is titled 'Specify availability group options' and contains the following fields:

- 'Availability group name:' with the text 'MyExpenseAG' entered in the input box.
- 'Cluster type:' with a dropdown menu showing 'Windows Server Failover Clustering' and a downward arrow.
- 'Database Level Health Detection' with an unchecked checkbox.

- ➔ **Para obtener más información,** consulte [Uso del asistente de grupo de disponibilidad \(SQL Server Management Studio\)](#).

2. **T-SQL.** Con el comando `CREATE AVAILABILITY GROUP`, especifique el tipo de clúster como WSFC destacado a continuación:

```
CREATE AVAILABILITY GROUP [ag1]
WITH (DB_FAILOVER = ON, CLUSTER_TYPE = WSFC)
FOR REPLICA ON
    N'agnode01'
WITH (
    ENDPOINT_URL = N'tcp://agnode01:1433',
    AVAILABILITY_MODE = SYNCHRONOUS_COMMIT,
    FAILOVER_MODE = EXTERNAL,
    SEEDING_MODE = AUTOMATIC
),
    N'agnode02'
WITH (
    ENDPOINT_URL = N'tcp://agnode02:1433',
    AVAILABILITY_MODE = SYNCHRONOUS_COMMIT,
    FAILOVER_MODE = EXTERNAL,
    SEEDING_MODE = AUTOMATIC
);
```

➔ **Para obtener más información** acerca de los argumentos para la sintaxis `CREATE AVAILABILITY GROUP`, consulte la [documentación de Microsoft](#).

3. **PowerShell.** Use cmdlets de PowerShell para crear y configurar un grupo de disponibilidad Always On en SQL Server 2017.

```
# Create the availability group
New-SqlAvailabilityGroup `
    -Name "MyAg" `
    -Path "SQLSERVER:\SQL\PrimaryComputer\Instance" `
    -AvailabilityReplica @($primaryReplica,$secondaryReplica) `
    -Database "MyDatabase"
```

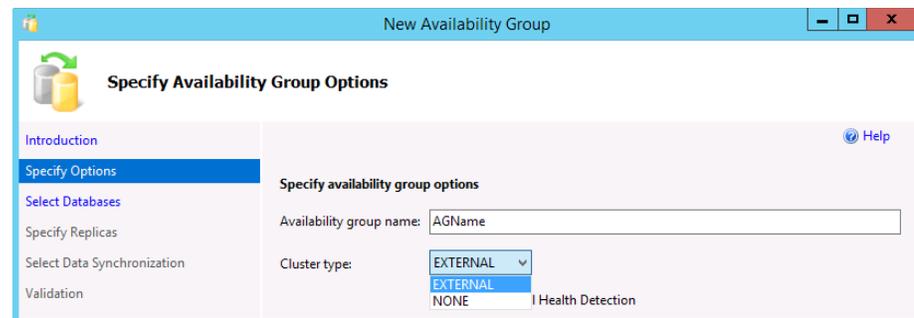
➔ **Para obtener más información**, consulte [Crear un grupo de disponibilidad \(SQL Server PowerShell\)](#).

Más información:
configuración
de un grupo
de disponibilidad
en Linux

Los pasos para crear un grupo de disponibilidad en servidores de Linux para alta disponibilidad son distintos de los pasos para un clúster de conmutación por error de Windows Server. En una FCI de Linux, debe habilitar grupos de disponibilidad en nodos de Linux con extremos y certificados. Entonces, puede utilizar Transact-SQL o el asistente para un nuevo grupo de disponibilidad en SSMS para crear un grupo de disponibilidad con un clúster de tipo External. Pacemaker es un ejemplo de entidad de clúster externo.

```
CREATE AVAILABILITY GROUP [ag1]
WITH (DB_FAILOVER = ON, CLUSTER_TYPE = EXTERNAL)
FOR REPLICA ON

),
```



Tras la creación de un grupo de disponibilidad en SQL Server, los recursos correspondientes deben crearse en Pacemaker cuando se especifica un clúster de tipo External. Hay dos recursos asociados con un grupo de disponibilidad, el propio grupo y una dirección IP:

```
sudo pcs resource create ag_cluster ocf:mssql:ag ag_name=ag
--master meta notify=true
```

Para asegurarse de que la dirección IP y el grupo de disponibilidad estén en ejecución en el mismo nodo, se debe configurar una restricción de colocación. Puede crear una restricción de orden para garantizar que el grupo de disponibilidad funcione y esté en ejecución antes de la dirección IP. De este modo, es posible crear y configurar un grupo de disponibilidad para SQL Server en Linux.

➔ **Para obtener indicaciones paso a paso**, visite la [documentación de Microsoft sobre la creación y la configuración de un grupo de disponibilidad para SQL Server en Linux](#).

[Grupos de disponibilidad híbridos basados en Windows y Linux](#)

Los grupos de disponibilidad pueden abarcar varios sistemas operativos; es decir, puede haber réplicas en un grupo de disponibilidad híbrido en ejecución en distintos sistemas operativos. Sin embargo, puesto que Windows y Linux usan distintos controladores para administrar la conmutación por error de la réplica (WSFC y Pacemaker), cualquier grupo de disponibilidad que contenga una mezcla de sistemas operativos tiene que usar un clúster de tipo None y, por lo tanto, requiere una conmutación por error manual.

[Más información: grupos de disponibilidad híbridos basados en Windows y Linux](#)

Puede crear un grupo de disponibilidad híbrido con una configuración multiplataforma. Por ejemplo, una en la que Windows Server hospede la réplica principal y un servidor de Linux hospede la réplica secundaria. Sin embargo, hay que tener en cuenta que, durante la creación del grupo de disponibilidad, debe seleccionar un clúster de tipo NONE, porque no hay administrador de clústeres:

```
CREATE AVAILABILITY GROUP [ag1]
WITH (CLUSTER_TYPE = NONE)
FOR REPLICA ON
    N'<WinSQLInstance>'
WITH (
    ENDPOINT_URL = N'tcp://<WinSQLInstance>:5022',
    AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT,
    SEEDING_MODE = AUTOMATIC,
    FAILOVER_MODE = MANUAL,
    SECONDARY_ROLE (ALLOW_CONNECTIONS = ALL)
),
N'<LinuxSQLInstance>'
WITH (
    ENDPOINT_URL = N'tcp://<LinuxSQLInstance>:5022',
    AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT,
    SEEDING_MODE = AUTOMATIC,
    FAILOVER_MODE = MANUAL,
    SECONDARY_ROLE (ALLOW_CONNECTIONS = ALL)
)
GO
```

Para conectar la réplica secundaria con el grupo de disponibilidad, debe especificar el tipo de clúster como External.

```
ALTER AVAILABILITY GROUP [ag1] JOIN WITH (CLUSTER_TYPE = NONE)
ALTER AVAILABILITY GROUP [ag1] GRANT CREATE ANY DATABASE
GO
```

La réplica principal permite lecturas y escrituras. Para cambiar de réplica principal, puede realizar una conmutación por error. En un grupo de disponibilidad para alta disponibilidad, el administrador de clústeres automatiza el proceso de conmutación por error. En un grupo de disponibilidad con el tipo de clúster **None**, el proceso de conmutación por error es manual. ■

Los clústeres de conmutación por error son solo una pieza de la alta disponibilidad. Cuando la ubicación de disponibilidad principal sufre un evento catastrófico, como un terremoto o una inundación,

el negocio debe estar preparado para que los sistemas se pongan en línea en otro lugar. Algunas características como las FCI Always On, el trasvase de registros y los grupos de disponibilidad Always On pueden ayudar con la continuidad empresarial y la recuperación ante desastres que hagan falta.

FCI Always On

En relación con la recuperación ante desastres, hay otro aspecto que considerar en el caso de las FCI: el almacenamiento compartido. Los mismos discos tienen que estar disponibles en las ubicaciones principales y secundarias. Para garantizar esta disponibilidad, es necesario un método externo, como la funcionalidad que ofrece el proveedor de almacenamiento en la capa de hardware. También puede usar herramientas de sincronización de almacenamiento como Réplica de almacenamiento en Windows y rsync o Unison en Linux para garantizar que los discos utilizados por la FCI existan en otro sitio.

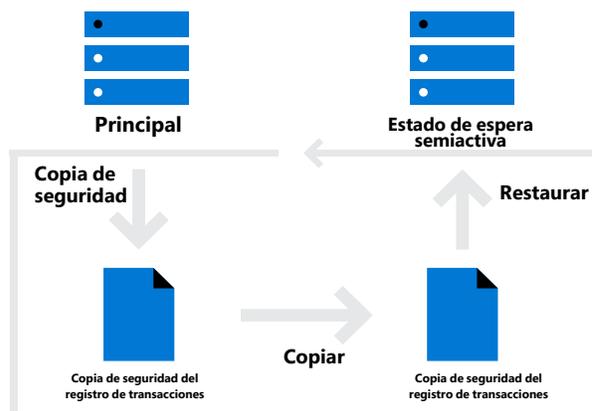
Grupos de disponibilidad Always On

Una de las ventajas de los grupos de disponibilidad Always On es que tanto la alta disponibilidad como la recuperación ante desastres pueden configurarse con una sola característica. No está obligado a garantizar que el almacenamiento compartido también tenga una alta disponibilidad. Resulta más sencillo tener réplicas locales en un centro de datos para alta disponibilidad y réplicas remotas en otros centros de datos para recuperación ante desastres, cada una con un almacenamiento independiente. La realización de copias adicionales de las bases de datos es la contrapartida de garantizar la redundancia.

Trasvase de registros

El trasvase de registros es una característica de SQL Server compatible con la alta disponibilidad. Con el trasvase de registros, se realizan copias de seguridad de los registros de transacciones de una base de datos en un servidor principal, se envían a una o más bases de datos secundarias y se restauran en esas bases de datos secundarias (tal como se ilustra en la figura). Así se mantienen las bases de datos secundarias sincronizadas con la principal. Este proceso se realiza de manera automática en intervalos preconfigurados para permitir la reversión de transacciones. Cuanto mayor sea el intervalo entre sincronizaciones, más lento será el tiempo de conmutación por error.

El trasvase de registros está disponible en Windows y en Linux. En Linux, los trabajos del Agente SQL Server no son parte de la instalación base de SQL Server. Se agregan con el paquete mssql-server-agent, que debe instalarse para usar el trasvase de registros. SQL Server en Linux usa el [Common Internet File System](#) (CIFS) y el servicio de protocolo de red Samba para almacenar las copias de seguridad de registros de transacciones en un recurso compartido de archivos de red. El Agente SQL Server en Linux ejecuta periódicamente un procedimiento almacenado que envía copias de seguridad de registros a los servidores secundarios. Al igual que en Windows, la recuperación de una base de datos protegida con trasvase de registros es una operación manual. ■



El tiempo de inactividad de aplicaciones empresariales críticas, las interrupciones de datos y la necesidad de mantener el rendimiento de las aplicaciones son las principales preocupaciones actuales de los negocios.

Para garantizar la continuidad empresarial ante la inactividad, es fundamental que las organizaciones valoren la implementación de estrategias de alta disponibilidad (HA) y recuperación ante desastres (DR). Microsoft Windows Server y Microsoft SQL Server proporcionan soluciones de HA y DR para garantizar que los negocios siempre estén en línea y disponibles con un rico conjunto de características de grupos de disponibilidad, instancias de clúster de conmutación por error y capacidades de trasvase de registros. Los grupos de disponibilidad también pueden proporcionar copias adicionales de una base de datos como parte de la misma arquitectura para escalar copias legibles. Las instancias y bases de datos de SQL Server ahora pueden convertirse en altamente disponibles en Windows y Linux mediante el uso de las mismas características. ■

[Obtenga más información sobre soluciones de alta disponibilidad para SQL Server.](#)

[Obtenga más información acerca de las instancias del clúster de conmutación por error Always On para SQL Server.](#)

[Obtenga más conocimientos sobre los grupos de disponibilidad Always On para SQL Server.](#)