



# MySQL

## Enterprise applications

NetApp  
January 02, 2026

# Tabla de contenidos

- MySQL ..... 1
  - Descripción general ..... 1
  - Configuración de la base de datos. .... 1
    - Estructura de archivos ..... 1
    - Parámetros de configuración ..... 4
      - innodb\_log\_file\_size ..... 5
      - innodb\_flush\_log\_at\_trx\_commit ..... 5
      - innodb\_doublewrite ..... 6
      - innodb\_buffer\_pool\_size ..... 6
      - innodb\_flush\_method ..... 6
      - innodb\_io\_capacity ..... 7
      - innodb\_lru\_scan\_profund ..... 8
      - open\_file\_limits ..... 8
  - Configuración de hosts ..... 8
    - Colocación de contenedores ..... 9
    - Mesas de NFSv3 ranuras ..... 9
    - Programadores de I/O..... 9
    - Descriptores de archivo ..... 10
  - Configuración del almacenamiento ..... 10
    - NFS ..... 11
    - SAN ..... 12

# MySQL

## Descripción general

MySQL y sus variantes, incluyendo MariaDB y Percona MySQL, es la base de datos más popular del mundo.



Esta documentación sobre ONTAP y la base de datos MySQL sustituye a *TR-4722: Base de datos MySQL sobre las mejores prácticas de ONTAP que se había publicado anteriormente*.

ONTAP es una plataforma ideal para la base de datos MySQL porque ONTAP está literalmente diseñada para las bases de datos. Numerosas funciones, como las optimizaciones de latencia de I/O aleatorias, pasando por una calidad de servicio avanzada o una funcionalidad FlexClone básica, se crearon específicamente para cubrir las necesidades de cargas de trabajo de bases de datos.

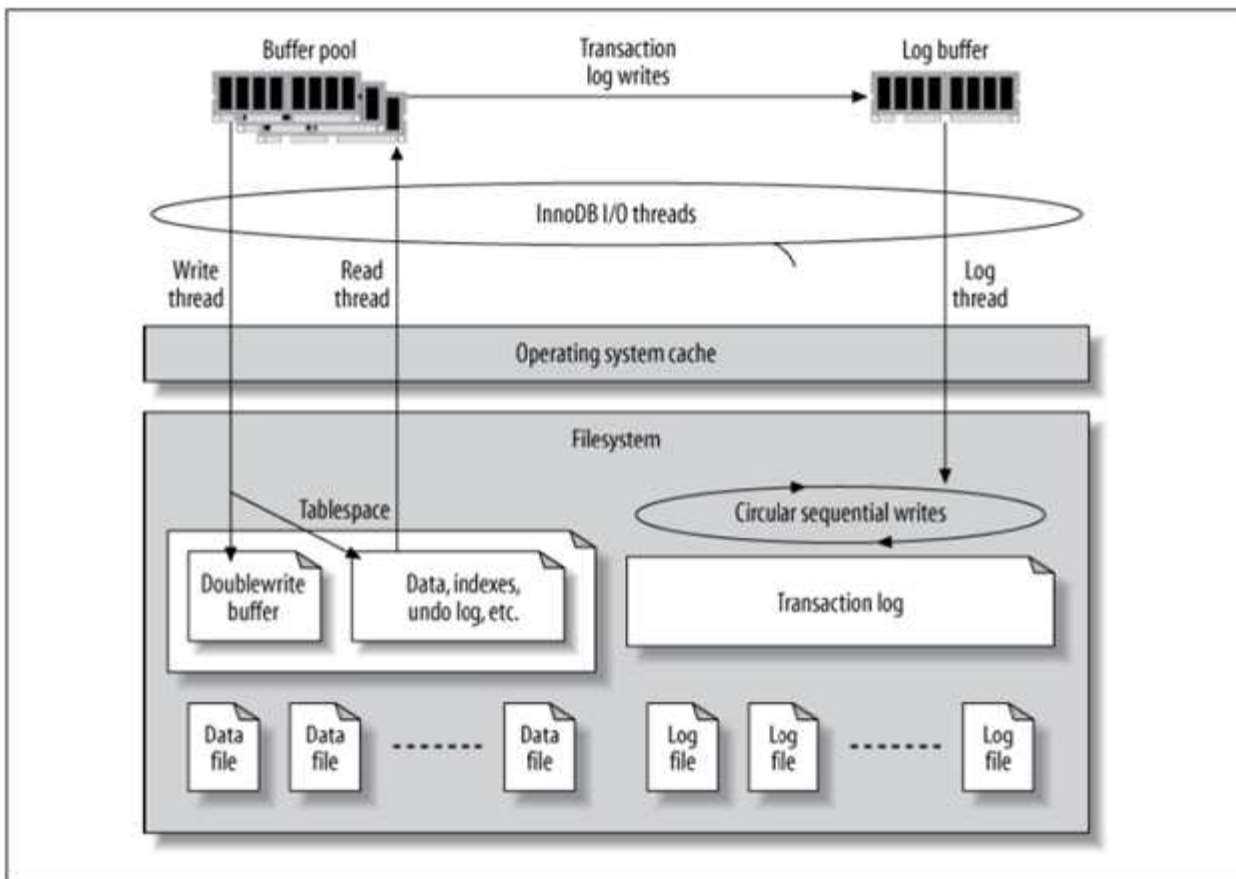
Otras funciones como las actualizaciones no disruptivas (entre ellas la sustitución de almacenamiento) garantizan que sus bases de datos cruciales seguirán estando disponibles. También se puede disponer de recuperación ante desastres instantánea para entornos grandes mediante MetroCluster o seleccionar bases de datos usando SnapMirror active sync.

Y lo que es más importante, ONTAP ofrece un rendimiento sin igual con la capacidad de dimensionar la solución en función de sus necesidades únicas. Nuestros sistemas de gama alta pueden ofrecer más de 1M 000 IOPS con latencias de microsegundos, pero si solo necesita 100K 000 IOPS, puede ajustar el tamaño de su solución de almacenamiento con una controladora más pequeña aún ejecutando exactamente el mismo sistema operativo de almacenamiento.

## Configuración de la base de datos

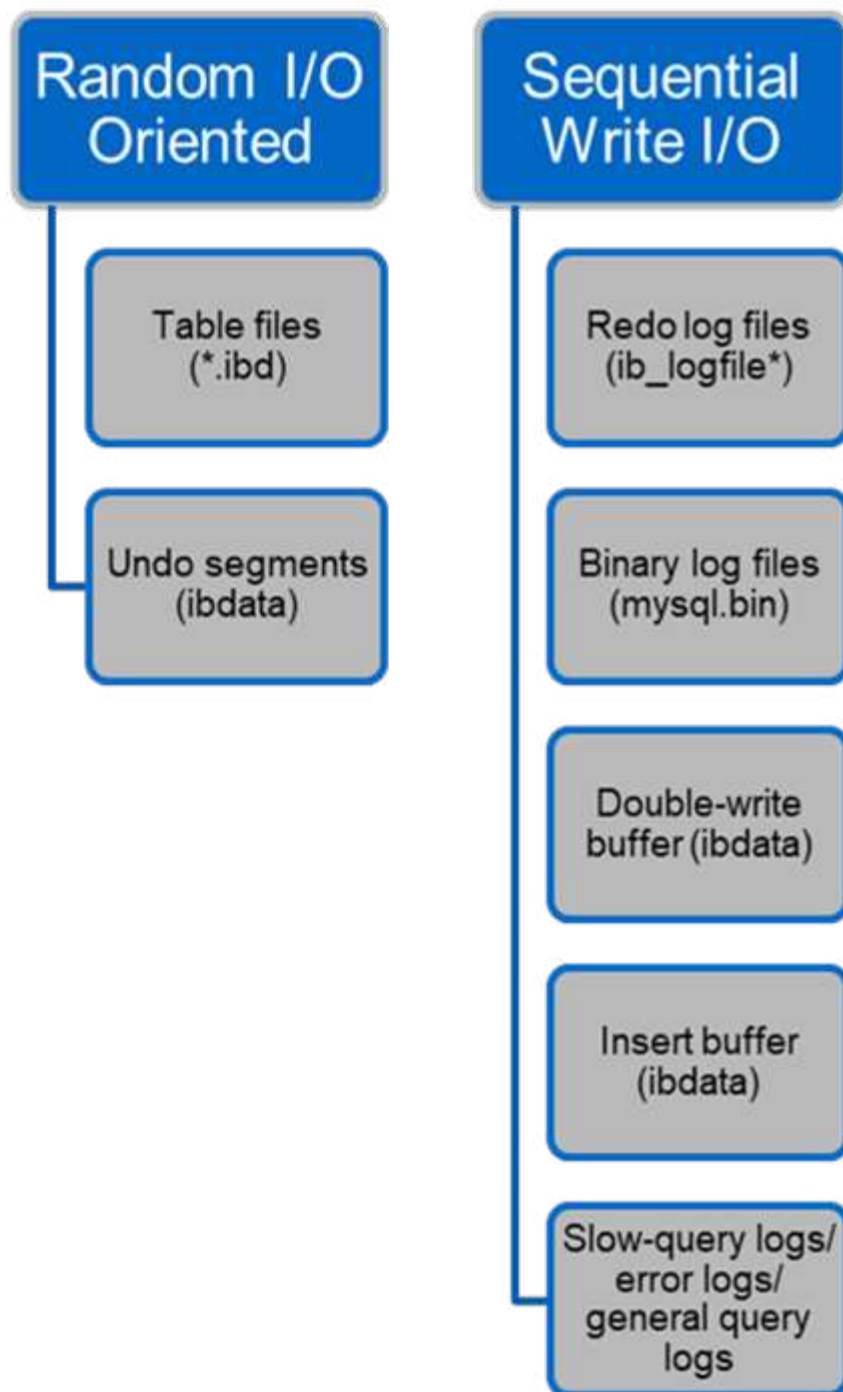
### Estructura de archivos

InnoDB actúa como la capa media entre el almacenamiento y el servidor MySQL, almacena los datos en las unidades.



MySQL I/O se clasifica en dos tipos:

- I/O de archivo aleatoria
- E/S de archivo secuencial



Los archivos de datos se leen y se sobrescriben aleatoriamente, lo que da como resultado un elevado número de IOPS. Por lo tanto, se recomienda el almacenamiento SSD.

Los archivos de registro de recuperación y los archivos de registro binarios son registros transaccionales. Se escriben secuencialmente, por lo que puede obtener un buen rendimiento en HDD con la caché de escritura. Se produce una lectura secuencial en la recuperación, pero rara vez causa un problema de rendimiento, porque el tamaño de los archivos de registro suele ser menor que el de los archivos de datos y las lecturas secuenciales son más rápidas que las lecturas aleatorias (se producen en archivos de datos).

El buffer de doble escritura es una característica especial de InnoDB. En primer lugar, InnoDB escribe las páginas vaciadas en el buffer de doble escritura y, a continuación, escribe las páginas en sus posiciones

correctas en los archivos de datos. Este proceso evita la corrupción de páginas. Sin el búfer de doble escritura, la página puede dañarse si se produce un fallo de alimentación durante el proceso de escritura en unidades. Como la escritura en el búfer de doble escritura es secuencial, está muy optimizada para HDD. Las lecturas secuenciales se producen en la recuperación.

Como la NVRAM de ONTAP ya proporciona protección de escritura, no es necesario el almacenamiento en búfer de doble escritura. MySQL tiene un parámetro, `skip_innodb_doublewrite`, para desactivar el buffer de doble escritura. Esta característica puede mejorar sustancialmente el rendimiento.

El buffer INSERT también es una característica especial de InnoDB. Si los bloques de índice secundarios no únicos no están en la memoria, InnoDB inserta entradas en el buffer de inserción para evitar operaciones de E/S aleatorias. Periódicamente, el buffer de inserción se fusiona en los árboles de índice secundarios de la base de datos. El buffer de inserción reduce el número de operaciones de E/S fusionando las solicitudes de E/S en el mismo bloque; las operaciones de E/S aleatorias pueden ser secuenciales. El búfer de inserción también está altamente optimizado para HDD. Tanto las escrituras secuenciales como las lecturas se producen durante las operaciones normales.

Los segmentos de deshacer están orientados a E/S aleatorias. Para garantizar la concurrencia multiversión (MVCC), InnoDB debe registrar imágenes antiguas en los segmentos de deshacer. La lectura de imágenes anteriores de los segmentos de deshacer requiere lecturas aleatorias. Si ejecuta una transacción larga con lecturas repetibles (como `mysqldump`, transacción única) o ejecuta una consulta larga, pueden producirse lecturas aleatorias. Por lo tanto, almacenar segmentos de deshacer en SSD es mejor en esta instancia. Si ejecuta sólo transacciones o consultas cortas, las lecturas aleatorias no suponen un problema.

**NetApp recomienda** el siguiente diseño de almacenamiento debido a las características de E/S de InnoDB.



- Un volumen para almacenar archivos de MySQL orientados a I/O aleatorios y secuenciales
- Otro volumen para almacenar archivos de MySQL orientados a I/O puramente secuenciales

Este diseño también le ayuda a diseñar políticas y estrategias de protección de datos.

## Parámetros de configuración

NetApp recomienda algunos parámetros de configuración MySQL importantes para obtener un rendimiento óptimo.

Parámetros	Valores
<code>innodb_log_file_size</code>	256M
<code>innodb_flush_log_at_trx_commit</code>	2
<code>innodb_doublewrite</code>	0
<code>innodb_flush_method</code>	<code>fsync</code>
<code>innodb_buffer_pool_size</code>	11G
<code>innodb_io_capacity</code>	8192
<code>innodb_buffer_pool_instances</code>	8
<code>innodb_lru_scan_profund</code>	8192
<code>open_file_limit</code>	65535

Para establecer los parámetros descritos en esta sección, debe cambiarlos en el archivo de configuración de MySQL (my.cnf). Las mejores prácticas de NetApp se deben a las pruebas que se realizan internamente.

## **innodb\_log\_file\_size**

Seleccionar el tamaño correcto para el tamaño del archivo de registro InnoDB es importante para las operaciones de escritura y para tener un tiempo de recuperación decente después de un fallo del servidor.

Dado que hay tantas transacciones registradas en el archivo, el tamaño del archivo de registro es importante para las operaciones de escritura. Cuando se modifican los registros, el cambio no se vuelve a escribir inmediatamente en el tablespace. En su lugar, el cambio se registra al final del archivo de registro y la página se marca como sucia. InnoDB utiliza su registro para convertir las operaciones de I/O aleatorias en operaciones de I/O secuenciales

Cuando el log está lleno, la página sucia se escribe en el tablespace en secuencia para liberar espacio en el archivo log. Por ejemplo, supongamos que un servidor se bloquea en medio de una transacción y que las operaciones de escritura solo se registran en el archivo de registro. Antes de que el servidor pueda volver a activarse, debe pasar por una fase de recuperación en la que se reproduzcan los cambios registrados en el archivo de registro. Cuantas más entradas haya en el archivo de registro, más tiempo tardará el servidor en recuperarse.

En este ejemplo, el tamaño del archivo log afecta tanto al tiempo de recuperación como al rendimiento de escritura. Al elegir el número correcto para el tamaño del archivo log, equilibre el tiempo de recuperación con respecto al rendimiento de escritura. Normalmente, cualquier cosa entre 128M y 512M es una buena relación calidad-precio.

## **innodb\_flush\_log\_at\_trx\_commit**

Cuando se produce un cambio en los datos, este no se escribe inmediatamente en el almacenamiento.

En su lugar, los datos se registran en un buffer de log, que es una parte de la memoria que InnoDB asigna a los cambios de buffer que se registran en el archivo log. InnoDB vacía el buffer en el archivo log cuando se confirma una transacción, cuando el buffer se llena, o una vez por segundo, lo que ocurra primero. La variable de configuración que controla este proceso es `innodb_flush_log_at_trx_commit`. Las opciones de valor incluyen:

- Cuando lo ajuste `innodb_flush_log_trx_at_commit=0`, InnoDB escribe los datos modificados (en el grupo de buffers de InnoDB) en el archivo log (`ib_logfile`) y vacía el archivo log (escribir en almacenamiento) cada segundo. Sin embargo, no hace nada cuando se confirma la transacción. Si hay un fallo de alimentación o un bloqueo del sistema, ninguno de los datos sin vaciar es recuperable, ya que no se escribe en el archivo de registro o en las unidades.
- Cuando lo ajuste `innodb_flush_log_trx_commit=1`, InnoDB escribe el buffer de log en el log de transacciones y se vacía en un almacenamiento duradero para cada transacción. Por ejemplo, para todas las confirmaciones de transacciones, InnoDB escribe en el log y, a continuación, escribe en el almacenamiento. Un almacenamiento más lento afecta negativamente al rendimiento; por ejemplo, se reduce el número de transacciones InnoDB por segundo.
- Cuando lo ajuste `innodb_flush_log_trx_commit=2`, InnoDB escribe el buffer de log en el archivo log en cada confirmación; sin embargo, no escribe datos en el almacenamiento. InnoDB vacía los datos una vez cada segundo. Incluso si hay un fallo de alimentación o un fallo del sistema, los datos de la opción 2 están disponibles en el archivo de registro y son recuperables.

Si el rendimiento es el objetivo principal, establezca el valor en 2. Como InnoDB escribe en las unidades una vez por segundo, no por cada confirmación de transacción, el rendimiento mejora drásticamente. Si se produce un fallo en el suministro eléctrico o un fallo, los datos se pueden recuperar del registro de transacciones.

Si la seguridad de los datos es el objetivo principal, establezca el valor en 1 para que, para cada confirmación de transacción, InnoDB se vacíe en las unidades. Sin embargo, el rendimiento puede verse afectado.



**NetApp recomienda** Establecer el valor `innodb_flush_log_trx_commit` en 2 para un mejor rendimiento.

## **innodb\_doublewrite**

Cuando `innodb_doublewrite` Está activado (por defecto), InnoDB almacena todos los datos dos veces: Primero en el buffer de doble escritura y luego en los archivos de datos reales.

Es posible desactivar este parámetro con `--skip-innodb_doublewrite` para pruebas de rendimiento o cuando le preocupa más el rendimiento superior que la integridad de los datos o posibles fallos. InnoDB utiliza una técnica de vaciado de archivos denominada doble escritura. Antes de escribir páginas en los archivos de datos, InnoDB las escribe en un área contigua denominada buffer de doble escritura. Una vez que se hayan completado las operaciones de escritura y vaciado en el buffer de doble escritura, InnoDB escribe las páginas en sus posiciones adecuadas en el archivo de datos. Si el sistema operativo o un proceso `mysqld` se bloquea durante la escritura de una página, InnoDB puede encontrar más tarde una buena copia de la página desde el buffer de doble escritura durante la recuperación de fallos.



**NetApp recomienda** deshabilitar el buffer de doble escritura. NVRAM de ONTAP sirve la misma función. El almacenamiento en búfer doble dañará innecesariamente el rendimiento.

## **innodb\_buffer\_pool\_size**

El pool de buffers de InnoDB es la parte más importante de cualquier actividad de ajuste.

InnoDB depende en gran medida del pool de buffers para almacenar en caché los índices y remar los datos, el índice hash adaptativo, el buffer INSERT y muchas otras estructuras de datos utilizadas internamente. El pool de búfer también almacena en búfer los cambios en los datos para que las operaciones de escritura no sean necesarias inmediatamente en el almacenamiento, lo que mejora el rendimiento. El pool de buffers es una parte integral de InnoDB y su tamaño debe ajustarse en consecuencia. Tenga en cuenta los siguientes factores al definir el tamaño del pool de buffers:

- Para una máquina exclusiva de InnoDB, establezca el tamaño del pool de buffers en 80% o más de RAM disponible.
- Si no es un servidor dedicado de MySQL, establezca el tamaño en 50% de RAM.

## **innodb\_flush\_method**

El parámetro `innodb_flush_method` especifica cómo InnoDB abre y vacía los archivos log y de datos.



## Optimizaciones

En la optimización de InnoDB, la configuración de este parámetro modifica el rendimiento de la base de datos cuando es aplicable.

Las siguientes opciones son para vaciar los archivos a través de InnoDB:

- `fsync`. InnoDB utiliza el `fsync()` llamada del sistema para vaciar los archivos de datos y de registro. Esta opción es el valor predeterminado.
- `O_DSYNC`. InnoDB utiliza el `O_DSYNC` opción para abrir y vaciar los archivos de registro y `fsync()` para vaciar los archivos de datos. InnoDB no utiliza `O_DSYNC` Directamente, porque ha habido problemas con él en muchas variedades de UNIX.
- `O_DIRECT`. InnoDB utiliza el `O_DIRECT` opción (`o_directio()` En Solaris) para abrir los archivos de datos y usos `fsync()` para vaciar los archivos de datos y de registro. Esta opción está disponible en algunas versiones de GNU/Linux, FreeBSD y Solaris.
- `O_DIRECT_NO_FSYNC`. InnoDB utiliza el `O_DIRECT` Durante el vaciado de I/O; sin embargo, omite el `fsync()` llamada del sistema posterior. Esta opción no es adecuada para algunos tipos de sistemas de archivos (por ejemplo, XFS). Si no está seguro de si su sistema de archivos requiere un `fsync()` llamada al sistema (por ejemplo, para conservar todos los metadatos del archivo), utilice el `O_DIRECT` en su lugar.

### Observación

En las pruebas de laboratorio de NetApp, el `fsync` La opción predeterminada se utilizó en NFS y SAN, y fue un gran improvisador de rendimiento `O_DIRECT`. Mientras se utiliza el método de vaciado como `O_DIRECT` Con ONTAP, hemos observado que el cliente escribe muchas escrituras de un solo byte en el borde del bloque de 4096 KB de forma en serie. Estas escrituras aumentan la latencia en la red y el rendimiento disminuye.

## innodb\_io\_capacity

En el plug-in InnoDB, se agregó un nuevo parámetro llamado `innodb_io_capacity` desde MySQL 5.7.

Controla el número máximo de IOPS que realiza InnoDB (lo que incluye la tasa de vaciado de páginas desfasadas y el tamaño de lote [ibuf] del buffer de inserción). El parámetro `innodb_io_capacity` define un límite superior de IOPS por tareas en segundo plano de InnoDB, como vaciar páginas del pool de buffers y fusionar datos del buffer de cambios.

Defina el parámetro `innodb_io_capacity` en el número aproximado de operaciones de E/S que el sistema puede realizar por segundo. Lo ideal es mantener la configuración lo más baja posible, pero no tan baja que las actividades en segundo plano se ralenticen. Si el valor es demasiado alto, los datos se eliminan del pool de buffers e insertan el buffer demasiado rápido para que el almacenamiento en caché proporcione una ventaja significativa.



**NetApp recomienda** que si utiliza esta configuración en NFS, analice el resultado de la prueba de IOPS (SysBench/fio) y establezca el parámetro en consecuencia. Utilice el valor más pequeño posible para vaciar y depurar para mantener el ritmo a menos que vea más páginas modificadas o sucias de las que desee en el pool de buffers de InnoDB.



No utilice valores extremos como 20.000 o más a menos que haya demostrado que los valores más bajos no son suficientes para su carga de trabajo.

El parámetro `InnoDB_IO_CAPACITY` regula las tasas de vaciado y la E/S relacionada



Puede dañar seriamente el rendimiento configurando este parámetro o el parámetro `innodb_io_capacity_max` demasiado alto y `wastin`

## **`innodb_lru_scan_profund`**

La `innodb_lru_scan_depth` El parámetro influye en los algoritmos y la heurística de la operación de vaciado para el pool de buffers de InnoDB.

Este parámetro es principalmente interesante para los expertos en rendimiento que ajustan las cargas de trabajo con un gran volumen de I/O. Para cada instancia de pool de buffers, este parámetro especifica hasta qué punto en la lista de páginas de uso menos reciente (LRU) el thread del limpiador de páginas debe continuar escaneando, buscando páginas sucias para vaciar. Esta operación en segundo plano se realiza una vez por segundo.

Puede ajustar el valor hacia arriba o hacia abajo para minimizar el número de páginas libres. No establezca el valor mucho más alto de lo necesario, ya que los escaneos pueden tener un costo de rendimiento significativo. Además, considere ajustar este parámetro al cambiar el Núm. De instancias del pool de buffers, porque `innodb_lru_scan_depth * innodb_buffer_pool_instances` define la cantidad de trabajo que realiza el thread de limpieza de páginas cada segundo.

Una configuración inferior a la predeterminada es adecuada para la mayoría de las cargas de trabajo. Considere aumentar el valor solo si cuenta con capacidad de I/O de reserva con una carga de trabajo típica. Por el contrario, si una carga de trabajo con gran cantidad de escritura satura la capacidad de E/S, disminuya el valor, especialmente si tiene un pool de buffers grande.

## **`open_file_limits`**

La `open_file_limits` el parámetro determina el número de archivos que el sistema operativo permite que `mysqld` abra.

El valor de este parámetro en tiempo de ejecución es el valor real permitido por el sistema y puede ser diferente del valor especificado al iniciar el servidor. El valor es 0 en sistemas donde MySQL no puede cambiar el número de archivos abiertos. Eficaz `open_files_limit` el valor se basa en el valor especificado en el inicio del sistema (si lo hay) y en los valores de `max_connections` y `table_open_cache` mediante el uso de estas fórmulas:

- $10 + \text{max\_connections} + (\text{table\_open\_cache} / 2)$
- `max_connections` 5
- Límite del sistema operativo si es positivo
- Si el límite del sistema operativo es infinito: `open_files_limit` el valor se especifica en el inicio; 5.000 si no hay ninguno

El servidor intenta obtener el número de descriptores de archivo utilizando el máximo de estos cuatro valores. Si no se pueden obtener muchos descriptores, el servidor intenta obtener tantos como el sistema permita.

## **Configuración de hosts**

## Colocación de contenedores

La contenerización de bases de datos MySQL es cada vez más frecuente.

La gestión de contenedores de bajo nivel casi siempre se realiza a través de Docker. Las plataformas de gestión de contenedores, como OpenShift y Kubernetes, simplifican aún más la gestión de entornos de contenedores de gran tamaño. Los beneficios de la contenerización incluyen costos más bajos, porque no hay necesidad de licenciar un hipervisor. Además, los contenedores permiten que varias bases de datos se ejecuten aisladas entre sí mientras comparten el mismo kernel y sistema operativo subyacente. Los contenedores se pueden aprovisionar en microsegundos.

NetApp ofrece Astra Trident para proporcionar funcionalidades de gestión avanzadas del almacenamiento. Por ejemplo, Astra Trident permite que un contenedor creado en Kubernetes aprovisiona automáticamente su almacenamiento en el nivel apropiado, aplique políticas de exportación, establezca políticas de Snapshot e incluso clone un contenedor a otro. Para obtener más información, consulte la ["Documentación de Astra Trident"](#).

## Mesas de NFSv3 ranuras

El rendimiento de NFSv3 en Linux depende de un parámetro llamado `tcp_max_slot_table_entries`.

Las tablas de ranuras TCP son equivalentes a NFSv3 a la profundidad de la cola del adaptador de bus de host (HBA). En estas tablas se controla el número de operaciones de NFS que pueden ser extraordinarias a la vez. El valor predeterminado suele ser 16, que es demasiado bajo para un rendimiento óptimo. El problema opuesto ocurre en los kernels más nuevos de Linux, que pueden aumentar automáticamente el límite de la tabla de ranuras TCP a un nivel que sature el servidor NFS con solicitudes.

Para obtener un rendimiento óptimo y evitar problemas de rendimiento, ajuste los parámetros del núcleo que controlan las tablas de ranuras TCP.

Ejecute el `sysctl -a | grep tcp.*.slot_table` command, y observe los siguientes parámetros:

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

Todos los sistemas Linux deben incluir `sunrpc.tcp_slot_table_entries`, pero solo algunos incluyen `sunrpc.tcp_max_slot_table_entries`. Ambos deben establecerse en 128.



Si no se establecen estos parámetros, puede tener efectos significativos en el rendimiento. En algunos casos, el rendimiento es limitado porque el sistema operativo linux no está emitiendo suficiente I/O. En otros casos, las latencias de I/O aumentan cuando el sistema operativo linux intenta emitir más operaciones de I/O de las que se pueden mantener.

## Programadores de I/O.

El kernel de Linux permite un control de bajo nivel sobre la forma en que se programa la E/S para bloquear los dispositivos.

Los valores predeterminados en varias distribuciones de Linux varían considerablemente. MySQL recomienda que utilice `NOOP` o a `deadline` Planificador de I/O con I/O asíncrono nativo (AIO) en Linux. En general, los clientes de NetApp y las pruebas internas muestran mejores resultados con NoOps.

El motor de almacenamiento InnoDB de MySQL utiliza el subsistema de E/S asíncrono (AIO nativo) en Linux para realizar solicitudes de lectura anticipada y escritura para páginas de archivos de datos. Este comportamiento es controlado por el `innodb_use_native_aio` opción de configuración, que está activada de forma predeterminada. Con AIO nativo, el tipo de programador de E/S tiene mayor influencia en el rendimiento de E/S. Realice pruebas de rendimiento para determinar qué programador de I/O ofrece los mejores resultados para su carga de trabajo y su entorno.

Consulte la documentación relevante de Linux y MySQL para obtener instrucciones sobre la configuración del programador de I/O.

## Descriptores de archivo

Para ejecutarse, el servidor MySQL necesita descriptores de archivo y los valores predeterminados no son suficientes.

Las utiliza para abrir nuevas conexiones, almacenar tablas en la caché, crear tablas temporales para resolver consultas complicadas y acceder a las persistentes. Si `mysqld` no puede abrir nuevos archivos cuando sea necesario, puede dejar de funcionar correctamente. Un síntoma común de este problema es el error 24, “Demasiados archivos abiertos”. El número de descriptores de archivo que `mysqld` puede abrir simultáneamente se define por el `open_files_limit` opción establecida en el archivo de configuración (`/etc/my.cnf`). Pero `open_files_limit` también depende de los límites del sistema operativo. Esta dependencia hace que la configuración de la variable sea más complicada.

MySQL no puede definir su `open_files_limit` opción superior a la especificada en `ulimit 'open files'`. Por lo tanto, debe establecer explícitamente estos límites en el nivel del sistema operativo para permitir que MySQL abra archivos según sea necesario. Hay dos formas de comprobar el límite de archivos en Linux:

- La `ulimit` command le proporciona rápidamente una descripción detallada de los parámetros que se permiten o bloquean. Los cambios realizados por la ejecución de este comando no son permanentes y se borrarán tras un reinicio del sistema.
- Cambios en la `/etc/security/limit.conf` el archivo es permanente y no se ve afectado por un reinicio del sistema.

Asegúrese de cambiar los límites duros y suaves para el usuario `mysql`. Los siguientes extractos son de la configuración:

```
mysql hard nfile 65535
mysql soft nfile 65353
```

En paralelo, actualice la misma configuración en `my.cnf` para utilizar completamente los límites de archivo abierto.

## Configuración del almacenamiento

## NFS

La documentación de MySQL recomienda el uso de NFSv4 para puestas en marcha de NAS.

### Tamaños de transferencia NFS de ONTAP

De forma predeterminada, ONTAP limitará el tamaño de I/O de NFS a 64K. La E/S aleatoria con una base de datos MySQL utiliza un tamaño de bloque mucho más pequeño, que es muy inferior al máximo de 64K KB. Las E/S de bloques grandes suelen estar en paralelo, por lo que el máximo de 64K KB tampoco es una limitación.

Hay algunas cargas de trabajo en las que el máximo de 64K crea una limitación. En particular, las operaciones de un solo subproceso, como las operaciones de copia de seguridad de exploración de tabla completa, se ejecutarán de forma más rápida y eficiente si la base de datos puede realizar menos I/O pero más grandes. El tamaño óptimo de gestión de I/O para ONTAP con cargas de trabajo de base de datos es 256K. Las opciones de montaje de NFS indicadas para sistemas operativos específicos a continuación se han actualizado de la versión 64K a la 256K correspondiente.

El tamaño de transferencia máximo para una SVM de ONTAP determinada se puede cambiar de la siguiente manera:

```
Cluster01::> set advanced
```

```
Warning: These advanced commands are potentially dangerous; use them only  
when directed to do so by NetApp personnel.
```

```
Do you want to continue? {y|n}: y
```

```
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size  
262144
```



Nunca reduzca el tamaño máximo permitido de transferencia en ONTAP por debajo del valor de rsize/wsize de los sistemas de archivos NFS montados actualmente. Esto puede crear bloqueos o incluso corrupción de datos con algunos sistemas operativos. Por ejemplo, si los clientes NFS se establecen actualmente con un valor de rsize/wsize de 65536 000, el tamaño de transferencia máximo de ONTAP se podría ajustar entre 65536 000 y 1048576 000 sin que ello afecte a porque los propios clientes están limitados. Reducir el tamaño máximo de transferencia por debajo de 65536 puede dañar la disponibilidad o los datos.

### NetApp recomienda



Ajuste de la siguiente configuración de NFSv4 fstab (/etc/fstab):

```
nfs4 rw,  
hard,nointr,bg,vers=4,proto=tcp,noatime,rsize=262144,wsiz=262144
```



Un problema común con NFSv3 era los archivos de registro InnoDB bloqueados después de una interrupción del suministro eléctrico. El uso de archivos de registro de hora o cambio solucionó este problema. Sin embargo, NFSv4 tiene operaciones de bloqueo y realiza un seguimiento de archivos abiertos y delegaciones.

## SAN

Las bases de datos más pequeñas pueden colocarse en parejas de LUN estándar siempre que las demandas de I/O y capacidad estén dentro de los límites de un único sistema de archivos LUN. Por ejemplo, una base de datos que requiere aproximadamente 2K 000 IOPS aleatorias se puede alojar en un único sistema de archivos con una sola LUN. Del mismo modo, una base de datos con un tamaño de solo 100GB TB podría adaptarse en una única LUN sin crear un problema de gestión.

Las bases de datos de mayor tamaño requieren varios LUN. Por ejemplo, una base de datos que requiere 100K 000 IOPS probablemente necesitará al menos ocho LUN. Una única LUN se convertiría en un cuello de botella debido al número inadecuado de canales SCSI a las unidades. Igualmente, una base de datos de 10TB TB sería difícil gestionar una sola LUN de 10TB TB. Los administradores de volúmenes lógicos están diseñados para unir las funcionalidades de rendimiento y capacidad de varias LUN y así mejorar el rendimiento y la capacidad de gestión.

En ambos casos, debería ser suficiente una pareja de volúmenes de ONTAP. Con una configuración sencilla, la LUN de archivo de datos se colocaría en un volumen dedicado, al igual que las LUN de registro. Con una configuración de gestor de volúmenes lógico, todos los LUN del grupo de volúmenes de archivos de datos estarían en un volumen dedicado, y las LUN del grupo de volúmenes de registro estarían en un segundo volumen dedicado.

**NetApp recomienda** el uso de dos sistemas de archivos para implementaciones de MySQL en SAN:

- El primer sistema de archivos almacena todos los datos MySQL, incluidos los tablespaces, los datos y el índice.
- El segundo sistema de archivos almacena todos los registros (registros binarios, registros lentos y registros de transacciones).

Hay varias razones para separar los datos de esta manera, incluyendo:



- Los patrones de E/S de los archivos de datos y los archivos de registro son diferentes. Separarlos permitirá más opciones con controles de calidad de servicio.
- Para un uso óptimo de la tecnología Snapshot es necesario poder restaurar los archivos de datos de forma independiente. La combinación de archivos de datos con archivos de registro interfiere con la restauración de archivos de datos.
- La tecnología SnapMirror de NetApp se puede usar para proporcionar una funcionalidad de recuperación ante desastres simple y con bajo objetivo de punto de recuperación para una base de datos; no obstante, se requieren diferentes programaciones de replicación para los archivos y registros de datos.



Utilice esta distribución básica de dos volúmenes para preparar la solución para el futuro, de modo que todas las funciones de ONTAP se puedan utilizar si fuera necesario.

**NetApp recomienda** formatear su unidad con el sistema de archivos ext4 debido a las siguientes características:



- Enfoque ampliado de las funciones de gestión de bloques utilizadas en el sistema de archivos de registro en diario (JFS) y las funciones de asignación retrasada del sistema de archivos extendido (XFS).
- ext4 permite sistemas de archivos de hasta 1 exbibyte ( $2^{60}$  bytes) y archivos de hasta 16 tebibytes ( $16 * 2^{40}$  bytes). Por el contrario, el sistema de archivos ext3 solo admite un tamaño máximo del sistema de archivos de 16TB GB y un tamaño máximo de archivo de 2TB GB.
- En los sistemas de archivos ext4, la asignación de bloques múltiples (mballoc) asigna varios bloques para un archivo en una sola operación, en lugar de asignarlos uno por uno, como en ext3. Esta configuración reduce la sobrecarga de llamar al asignador de bloques varias veces y optimiza la asignación de memoria.
- Aunque XFS es el valor predeterminado para muchas distribuciones de Linux, administra los metadatos de manera diferente y no es adecuado para algunas configuraciones de MySQL.



**NetApp recomienda** usar opciones de tamaño de bloque 4K con la utilidad mkfs para alinearse con el tamaño de LUN de bloque existente.

```
mkfs.ext4 -b 4096
```

Las LUN de NetApp almacenan datos en bloques físicos de 4KB KB, lo que produce ocho bloques lógicos de 512 bytes.

Si no se configura el mismo tamaño de bloque, las operaciones de I/O no se alinearán con los bloques físicos correctamente y podrían escribir en dos unidades distintas de un grupo RAID, lo que dará como resultado latencia.



Es importante alinear las operaciones de I/O para que las operaciones de lectura/escritura sean fluidas. Sin embargo, cuando las operaciones de I/O se inician en un bloque lógico que no está al inicio de un bloque físico, la I/O se desalinea. Las operaciones de I/O se alinean solo cuando comienzan con un bloque lógico, es decir, el primer bloque lógico de un bloque físico.

## Información de copyright

Copyright © 2026 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

## Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.