Ajuste de los SQL parámetros de Postgre en Amazon RDS y Amazon Aurora

AWS Guía prescriptiva

AWS Guía prescriptiva: Ajuste de los SQL parámetros de Postgre en Amazon RDS y Amazon Aurora

Table of Contents

Ajuste de los parámetros de PostgreSQL en Amazon RDS y Amazon Aurora

Sumana Yanamandra, Ramu Jagini y Rohit Kapoor, Amazon Web Services ()AWS

Febrero de 2024 (historial del documento)

Amazon Aurora PostgreSQL Compatible Edition y Amazon Relational Database Service (Amazon RDS) para PostgreSQL son sofisticados servicios de bases de datos relacionales de código abierto que ofrecen una amplia gama de características. Puede utilizar estos servicios para configurar su base de datos PostgreSQL en una variedad de plataformas y aplicaciones.

Aurora y Amazon RDS ofrecen una forma simplificada de administrar y operar sus bases de datos PostgreSQL. Están diseñados para administrar la infraestructura de la base de datos y proporcionar alta disponibilidad, durabilidad y escalabilidad, a la vez que usted se centra en el desarrollo de aplicaciones. Sin embargo, es posible que las configuraciones predeterminadas de estos servicios no sean óptimas para todas las cargas de trabajo. De forma predeterminada, estos servicios están configurados para ejecutarse en todas partes con la menor cantidad de recursos posible y sin introducir vulnerabilidades. Ajustar los parámetros puede ayudarle a lograr un mejor rendimiento, reducir el tiempo de inactividad y mejorar la eficiencia general de la base de datos. Al optimizar los parámetros para su carga de trabajo específica, puede aprovechar al máximo las capacidades que ofrecen Amazon RDS y Aurora y maximizar sus beneficios.

Por ejemplo, puede mejorar el rendimiento optimizando Aurora y Amazon RDS para PostgreSQL y configurando sus parámetros. También debe tener en cuenta el rendimiento al crear consultas a la base de datos. Incluso al optimizar la configuración de la base de datos, el sistema puede funcionar mal si las consultas escanean tablas completas, utilizan un índice o ejecutan costosas operaciones de combinación o agregación.

Esta guía está destinada a desarrolladores de bases de datos, ingenieros de bases de datos y administradores que desean ajustar los parámetros de memoria, aspiración automática, registro y replicación lógica para sus bases de datos PostgreSQL. La guía también incluye los parámetros que son específicos de Amazon RDS for PostgreSQL y Aurora, compatible con PostgreSQL. Ajustar estos parámetros puede ayudarle a optimizar el rendimiento de la base de datos y a reducir el uso de recursos para su carga de trabajo específica, lo que se traduce en un mejor rendimiento y en un ahorro de costes.

Uso de grupos de parámetros de base de datos y clúster de bases de datos

Amazon RDS y Aurora pueden determinar automáticamente los valores de los parámetros más adecuados para determinadas configuraciones en función del tamaño de la instancia de base de datos. También admiten la personalización de parámetros para ajustar el rendimiento mediante grupos de parámetros para instancias y clústeres de bases de datos.

Puede usar grupos de parámetros de bases de datos y clústeres de bases de datos para modificar los parámetros que controlan varios aspectos del comportamiento del motor de base de datos, como el uso de la memoria, las E/S del disco, las redes y el bloqueo. Al ajustar estos parámetros, puede optimizar el motor de base de datos para su carga de trabajo específica y mejorar el rendimiento.

Puede crear y configurar grupos de parámetros de bases de datos y clústeres de bases de datos mediante la AWS Management Console API, AWS Command Line Interface (AWS CLI) o Amazon RDS. En esta guía se asume que está utilizando la AWS CLI. Para obtener instrucciones sobre la consola y la API, consulte Trabajar con grupos de parámetros de base de datos y Trabajar con grupos de parámetros de clústeres de bases de datos en la documentación de Amazon RDS.

> **Important**
>
> Para utilizar los AWS CLI comandos que se proporcionan en esta guía, primero debe instalar y configurar el AWS CLI.

Para crear y configurar un grupo de parámetros de base de datos:

```
# Create a new DB parameter group
aws rds create-db-parameter-group \
  --db-parameter-group-name mydbparamgroup \
  --db-parameter-group-family postgres13 \
  --description "My DB Parameter Group"

# Modify a parameter on the DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <param group name> \
  --parameters "ParameterName=max_connections<parameter-
name>,ParameterValue=<value>,ApplyMethod=immediate"
```

```
# Verify DB parameters
aws rds describe-db-parameters \
  --db-parameter-group-name aurora-instance-1
```

Para crear y configurar un grupo de parámetros de un clúster de base de datos:

```
# Create a new DB cluster parameter group
aws rds create-db-cluster-parameter-group \
  --db-cluster-parameter-group-name myparametergroup \
  --db-parameter-group-family postgres12 \
  --description "My new parameter group"

# Modify a parameter on the DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name aws-guide-cluster \
  --parameters "ParameterName=<parameter-name>,ParameterValue=,ApplyMethod=immediate"

# Allocate the new DB cluster parameter to your cluster
aws rds modify-db-cluster  \
  --db-cluster-identifier  \
  --db-cluster-parameter-group-name=-cluster

# Verify cluster parameters
aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name=-cluster
```

> **Note**
>
> Aurora y Amazon RDS proporcionan un grupo de parámetros predeterminado con valores preconfigurados que no se pueden cambiar.

Los grupos de parámetros se pueden configurar como estáticos o dinámicos. Los parámetros dinámicos se aplican inmediatamente, independientemente de si la `ApplyMethod=immediate` opción está habilitada. Los parámetros estáticos requieren un reinicio manual para que surtan efecto.

Ajustando los parámetros de la memoria

Ajustar los parámetros de la memoria es una tarea esencial para optimizar el rendimiento de las bases de datos compatibles con Amazon RDS y Aurora PostgreSQL. La asignación adecuada de la memoria para diversas operaciones de la base de datos, como la ejecución de consultas, la clasificación, la indexación y el almacenamiento en caché, puede mejorar considerablemente el rendimiento de la base de datos. En esta sección se describen algunos de los parámetros de memoria más importantes de Amazon RDS y Aurora compatibles con PostgreSQL, incluidos sus valores predeterminados, las fórmulas para calcular los valores adecuados y cómo cambiarlos. Para obtener una lista completa de parámetros, consulte Trabajar con parámetros en su instancia de base de datos de RDS para PostgreSQL en la documentación de Amazon RDS y Parámetros de Amazon Aurora PostgreSQL en la documentación de Aurora.

La optimización de estos parámetros requiere un conocimiento profundo de la carga de trabajo de la base de datos, así como de los recursos disponibles en la instancia de base de datos Aurora o Amazon RDS. El rendimiento del sistema se ve influido por dos amplias categorías de parámetros: los parámetros vitales y los parámetros contingentes.

Los parámetros vitales son parámetros indispensables que ejercen una influencia significativa y directa en el rendimiento del sistema y son esenciales para lograr resultados óptimos.

- shared_buffers
- temp_buffers
- tamaño_caché efectivo
- work_mem
- maintenance_work_mem

Los parámetros contingentes son parámetros específicos del escenario y del negocio. Están supeditados a otros factores y desempeñan un papel indirecto pero fundamental a la hora de respaldar parámetros vitales y maximizar el rendimiento general del sistema.

- costado_de_página aleatorio
- seq_page_cost
- track_activity_query_size
- in_transaction_session_timeout

- statement_timeout
- search_path
- conexiones máximas

Estos parámetros se analizan con más detalle en las siguientes secciones.

shared_buffers

El `shared_buffers` parámetro controla la cantidad de memoria que utiliza PostgreSQL para almacenar en caché los datos en la memoria. Establecer este parámetro en un valor adecuado puede ayudar a mejorar el rendimiento de las consultas.

En Amazon RDS, el valor predeterminado de `shared_buffers` se establece en `{DBInstanceClassMemory/32768}` bytes, en función de la memoria disponible para la instancia de base de datos. Para Aurora, el valor predeterminado se establece en`{DBInstanceClassMemory/12038,-50003}`, en función de la memoria disponible para la instancia de base de datos. El valor óptimo para este parámetro depende de varios factores, como el tamaño de la base de datos, el número de conexiones simultáneas y la memoria disponible de la instancia.

AWS CLI sintaxis

El siguiente comando cambia `shared_buffers` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify shared_buffers on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=shared_buffers,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify shared_buffers on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters "ParameterName=shared_buffers,ParameterValue=<new-
value>,ApplyMethod=immediate"
```

Tipo: estático (la aplicación de los cambios requiere un reinicio)

Valor predeterminado: `{DBInstanceClassMemory/32768}` bytes en Amazon RDS para PostgreSQL, en Aurora compatible con `{DBInstanceClassMemory/12038,-50003}` PostgreSQL. En la mayoría de los casos, esta ecuación equivale aproximadamente al 25 por ciento de la memoria del sistema. Siguiendo esta guía, la `shared_buffers` configuración del grupo de parámetros se configura utilizando las unidades predeterminadas de PostgreSQL de búferes de 8K en lugar de bytes o kilobytes.

La configuración de los `shared_buffers` parámetros puede tener un impacto significativo en el rendimiento, por lo que le recomendamos que pruebe los cambios minuciosamente para asegurarse de que el valor es el adecuado para su carga de trabajo.

Ejemplo

Supongamos que tiene una aplicación de servicios financieros que ejecuta una base de datos PostgreSQL en Amazon RDS o Aurora. Esta base de datos se utiliza para almacenar los datos de las transacciones de los clientes. Tiene una gran cantidad de tablas y múltiples aplicaciones acceden a ella en una gran cantidad de servidores. El rendimiento de las consultas de la aplicación es lento y el uso de la CPU es elevado. Usted determina que ajustar el `shared_buffers` parámetro puede ayudar a mejorar el rendimiento.

En Amazon RDS for PostgreSQL, el `shared_buffers` valor predeterminado de se `{DBInstanceClassMemory/32768}` establece en bytes de `db.r5.xlarge` memoria disponible (por ejemplo, 3 GB). Para determinar el valor adecuado`shared_buffers`, ejecute una serie de pruebas con valores variables de`shared_buffers`, empezando por el valor predeterminado de la memoria disponible y aumentando el valor gradualmente. Para cada prueba, se mide el rendimiento de la consulta y el uso de la CPU de la base de datos.

En función de los resultados de las pruebas, usted determina que si se establece el valor en 8 GB, se obtiene el mejor rendimiento general de `shared_buffers` las consultas y el mejor uso de la CPU para su carga de trabajo. El valor se determina mediante una combinación de pruebas y análisis de las características de la carga de trabajo, incluido el tamaño de la base de datos, el número y la complejidad de las consultas, el número de usuarios simultáneos y los recursos del sistema disponibles. Tras realizar el cambio, los sistemas de supervisión comprueban el rendimiento de la base de datos para garantizar que el nuevo valor sea adecuado para la carga de trabajo. A continuación, puede ajustar los parámetros adicionales según sea necesario para mejorar aún más el rendimiento.

temp_buffers

`temp_buffers`es un parámetro de configuración clave en Aurora, compatible con PostgreSQL y Amazon RDS for PostgreSQL, que puede afectar significativamente al rendimiento de las cargas de trabajo que implican operaciones de clasificación, hashes y agregación en tablas temporales. Este parámetro determina la cantidad de memoria asignada a los búferes temporales, lo que, a su vez, afecta a la eficiencia y la velocidad de dichas operaciones. Si no hay suficiente memoria asignada`temp_buffers`, es posible que el sistema tenga que utilizar métodos más lentos y menos eficientes para las operaciones de clasificación, hashes y agregación en las tablas temporales, lo que provocará un rendimiento inferior al óptimo.

AWS CLI sintaxis

El siguiente comando cambia `temp_buffers` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify temp_buffers on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=temp_buffers,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify temp_buffers on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=temp_buffers,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: 8 MB

Para obtener más información sobre este parámetro, consulte Consumo de recursos en la documentación de PostgreSQL.

Ejemplo

Si su carga de trabajo implica muchas operaciones de clasificación, codificación y agregación en tablas temporales, es `temp_buffers` posible que no asigne suficiente memoria. En este caso,

es posible que el sistema tenga que realizar operaciones de clasificación en tablas temporales, lo que lleva a utilizar métodos basados en discos más lentos, en lugar de realizar operaciones de clasificación, codificación y agregación en memoria. Esto puede provocar una ralentización significativa del rendimiento de las consultas, especialmente en el caso de las consultas que implican conjuntos de datos de gran tamaño. Aumentar el valor de `temp_buffers` puede garantizar que haya suficiente memoria disponible para realizar dichas operaciones en la memoria, lo que se traduce en una mejora significativa del rendimiento.

Para encontrar el valor óptimo`temp_buffers`, supervise el rendimiento del sistema e identifique las áreas en las que el rendimiento no sea óptimo. Si observa tiempos de respuesta a las consultas lentos o un uso elevado de la CPU, considere la posibilidad de realizar ajustes. `temp_buffers` Por ejemplo, si su carga de trabajo incluye muchas tablas temporales, aumentar el valor de `temp_buffers` puede ayudar a garantizar que estas tablas se almacenen en la memoria. Esto puede ser mucho más rápido que usar E/S de lectura/escritura desde el almacenamiento.

Experimente con valores diferentes o `temp_buffers` en pequeños incrementos y supervise cuidadosamente el rendimiento del sistema después de cada cambio. Analice el impacto de los distintos valores en el rendimiento y ajuste la configuración en función de las características específicas de su carga de trabajo.

efective_cache_size

El `effective_cache_size` parámetro especifica la cantidad de memoria que PostgreSQL debe suponer que está disponible para almacenar datos en caché. Si se establece este parámetro correctamente, se puede mejorar el rendimiento al permitir que PostgreSQL haga un mejor uso de la memoria disponible.

AWS CLI sintaxis

El siguiente comando cambia `effective_cache_size` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify effective_cache_size on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=effective_cache_size,ParameterValue=<new_value>,ApplyMethod=immediate"
```

```
# Modify effective_cache_size on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=effective_cache_size,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: SUM(DBInstanceClassMemory/12038,-50003) KB

Ejemplo

Una plataforma de aprendizaje en línea tiene una gran base de datos de materiales del curso, datos de los estudiantes y otro contenido a la que los usuarios acceden con frecuencia. La aplicación se ejecuta en una instancia de Amazon RDS db.r5.xlarge for PostgreSQL con 32 GB de memoria. La aplicación experimenta un rendimiento lento cuando los usuarios intentan leer el contenido al que acceden con frecuencia. Tras analizar el uso de recursos del servidor de bases de datos, se determina que PostgreSQL no está haciendo un uso óptimo de la memoria disponible.

El effective_cache_size parámetro de Amazon RDS for PostgreSQL controla la cantidad de memoria que utiliza el servidor para el almacenamiento en caché del disco. El valor predeterminado está establecido en SUM({DBInstanceClassMemory/12038},-50003) KB para la clase de instanciadb.r5.xlarge, pero es posible que este valor predeterminado no sea adecuado para todas las cargas de trabajo. En este ejemplo, es posible que el servidor de la base de datos almacene grandes cantidades de materiales del curso y datos de los alumnos a los que se accede con frecuencia. Si se aumenta el valor del effective_cache_size parámetro, se pueden almacenar más datos en la memoria caché, lo que reduce el número de lecturas de disco necesarias y mejora el rendimiento de las consultas.

Al ejecutar una consulta, Amazon RDS for PostgreSQL comprueba primero si los datos requeridos por la consulta ya están en la memoria caché. Si es así, los datos se pueden leer desde la memoria en lugar de leerse desde el disco. Si los datos no están en la memoria caché, deben leerse del disco, lo que puede ser una operación lenta.

En el caso de la plataforma de aprendizaje en línea, puede decidir effective_cache_size configurarla en 16 GB (la mitad de la memoria disponible) tras realizar pruebas y análisis. Este valor permite a PostgreSQL aprovechar mejor la memoria disponible, lo que reduce la cantidad de lecturas de disco necesarias y mejora el rendimiento de las consultas.

work_mem

El `work_mem` parámetro controla la cantidad de memoria que utilizan las consultas para las operaciones de clasificación y hash. Su valor predeterminado es de 4 MB. Si una consulta incluye varias operaciones, puede utilizar hasta 4 MB para cada operación. Si se aumenta el valor de, `work_mem` se puede mejorar el rendimiento de las consultas que requieren clasificación o codificación hash, ya que estas operaciones requieren más memoria. Sin embargo, si se establece este parámetro en un valor demasiado alto, se puede producir un uso excesivo de memoria y, por lo tanto, una degradación del rendimiento.

Para calcular el valor óptimo`work_mem`, puede utilizar la siguiente fórmula:

```
work_mem = (available_memory / (max_connections * work_mem_fraction))
```

donde `available_memory` es la cantidad total de memoria disponible en el servidor, max_connections es el número máximo de conexiones permitidas y work_mem_fraction es una fracción que determina la cantidad de memoria disponible que debe asignarse a cada conexión.

AWS CLI sintaxis

El siguiente comando cambia `work_mem` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify work_mem on a DB parameter group
aws rds modify-db-parameter-group \
   --db-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=work_mem,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify work_mem on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
   --db-cluster-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=work_mem,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: 4 MB

Ejemplo

Una herramienta de análisis de redes sociales procesa una gran cantidad de datos, y las consultas que implican operaciones complejas de clasificación y unión provocan una gran cantidad de E/S en el disco y se transfieren al disco. Si aumenta el valor `work_mem` de 4 MB a 16 MB, PostgreSQL puede utilizar más memoria para estas operaciones. Esto reduce la cantidad de E/S y mejora el rendimiento de las consultas.

maintenance_work_mem

El `maintenance_work_mem` parámetro controla la cantidad de memoria que utilizan las operaciones de mantenimiento, como la creación de índices, y la creación de índices. `VACUUM ANALYZE` El valor predeterminado de este parámetro en Amazon RDS y Aurora es de 64 MB.

Para calcular el valor adecuado para este parámetro, puede usar esta fórmula:

```
maintenance_work_mem = (total_memory - shared_buffers) / (max_connections * 5)
```

Aurora PostgreSQL Compatible Edition y Amazon RDS for PostgreSQL aplican la siguiente fórmula para establecer el valor óptimo:

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

AWS CLI sintaxis

El siguiente comando cambia `maintenance_work_mem` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify maintenance_work_mem on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=maintenance_work_mem,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify maintenance_work_mem on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=maintenance_work_mem,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: 64 MB

Ejemplo

Su aplicación a gran escala utiliza una base de datos PostgreSQL alojada en Aurora o Amazon RDS. Observa que la base de datos es lenta y no responde durante las actividades de mantenimiento, como la limpieza y la indexación. Puede supervisar métricas como el uso de la memoria, los tiempos de las operaciones de mantenimiento y el uso de la CPU para determinar si el valor actual de `maintenance_work_mem` está causando problemas.

Para determinar el valor óptimo`maintenance_work_mem`, puede ajustar el parámetro y supervisar su impacto. Si el uso de la memoria es constantemente elevado o los tiempos de operación son más largos de lo esperado durante las operaciones de mantenimiento, aumentarla `maintenance_work_mem` podría ser útil. Por el contrario, si el uso de la CPU es constantemente elevado durante las operaciones de mantenimiento, disminuirlo `maintenance_work_mem` podría ser útil. Al realizar ajustes y pruebas, podrá encontrar el valor óptimo `maintenance_work_mem` que ofrezca el mejor equilibrio entre el uso de la memoria, los tiempos de las operaciones de mantenimiento y el uso de la CPU.

Durante la investigación, supongamos que el valor predeterminado de 64 MB `maintenance_work_mem` es demasiado bajo para el tamaño de la base de datos. Como resultado, las operaciones de mantenimiento tardan más en completarse, provocan un tiempo de inactividad excesivo y ralentizan el rendimiento de la aplicación. Para solucionar este problema, puede decidir ajustar el `maintenance_work_mem` parámetro incrementándolo de 64 MB a 512 MB (lo que identifica como el valor óptimo). La aplicación del cambio puede mejorar los tiempos de las operaciones de mantenimiento en dos tercios. Por ejemplo, una operación de aspiración que antes tardaba 30 minutos en completarse ahora puede tardar solo 10 minutos. Como resultado de esta optimización, la base de datos ahora puede gestionar las actividades de mantenimiento de forma más eficiente.

random_page_cost

El `random_page_cost` parámetro ayuda a determinar el costo de realizar un acceso aleatorio a las páginas. El planificador de consultas de Amazon RDS y Aurora utiliza este parámetro, junto con otras estadísticas de la tabla, para determinar el plan más eficaz para ejecutar una consulta.

Los `random_page_cost` parámetros `seq_page_cost` y están estrechamente relacionados y, por lo general, el planificador los usa juntos para comparar los costos de los distintos métodos de acceso y decidir cuál es el más eficiente. Por lo tanto, si cambia uno de estos parámetros, también debe considerar si es necesario ajustar el otro parámetro.

En general, el planificador de consultas intenta minimizar el coste de ejecutar una consulta. El costo se determina mediante una combinación del número de lecturas de páginas del disco y el valor de`random_page_cost`. Un valor más alto de `random_page_cost` tiende a favorecer los escaneos secuenciales, mientras que un valor más bajo tiende a favorecer los escaneos indexados. Un valor más bajo también tiende a favorecer las uniones en bucle anidadas en lugar de las uniones hash.

El `random_page_cost` parámetro utiliza el valor predeterminado del motor PostgreSQL (4), a menos que se establezca un valor en el grupo de parámetros o en la sesión local. Puede ajustar este valor en función de las características específicas del servidor y de la carga de trabajo. Si la mayoría de los índices utilizados en la carga de trabajo caben en la memoria o en la caché por niveles de Aurora, `seq_page_cost` es apropiado cambiar el valor de `random_page_cost` por un valor cercano a.

AWS CLI sintaxis

El siguiente comando cambia `random_page_cost` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify random_page_cost on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=random_page_cost,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify random_page_cost on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=random_page_cost,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: 4

Ejemplo

Supongamos que tiene una base de datos que almacena una gran cantidad de datos en una tabla que se consulta con frecuencia con filtros en columnas no indexadas. Las consultas tardan mucho en completarse y el planificador de consultas no selecciona el plan más eficaz para acceder a los datos.

Una forma de mejorar el rendimiento sería reducir el `random_page_cost` parámetro. Si lo estableces en 1, el coste del acceso aleatorio a la página sería cuatro veces inferior al valor predeterminado. Si se mantiene `random_page_cost` el valor predeterminado de 4, el acceso aleatorio a las páginas sería cuatro veces más caro que el acceso secuencial a las páginas (según lo determine el `seq_page_cost` parámetro, que es 1,0 de forma predeterminada). Sin embargo, en este caso específico, el acceso aleatorio a las páginas podría resultar mucho más caro según el tipo de almacenamiento.

Si se reduce el valor del `random_page_cost` parámetro, es más probable que el planificador de consultas seleccione un plan basado en índices o utilice un método de acceso diferente que se adapte mejor a las características específicas de la tabla.

Te recomendamos que supervises el rendimiento de las consultas después de cambiar el parámetro y que realices los ajustes necesarios. También deberías consultar el planificador de consultas con una `EXPLAIN` declaración para comprobar si está seleccionando un plan eficiente.

Este es solo un ejemplo. La configuración óptima depende de las características específicas de su carga de trabajo. Además, este es solo un aspecto del ajuste del rendimiento; también debe tener en cuenta otros parámetros y opciones de configuración que pueden afectar al rendimiento de las consultas.

seq_page_cost

El `seq_page_cost` parámetro ayuda a determinar el costo de realizar el acceso secuencial a las páginas. El planificador de consultas de Amazon RDS y Aurora utiliza este parámetro, junto con otras estadísticas de la tabla, para determinar el plan más eficaz para ejecutar una consulta.

Los `random_page_cost` parámetros `seq_page_cost` y están estrechamente relacionados y, por lo general, el planificador los usa juntos para comparar los costos de los distintos métodos de acceso y decidir cuál es el más eficiente. Por lo tanto, si cambia uno de estos parámetros, también debe considerar si es necesario ajustar el otro parámetro.

Cuando se accede a una tabla de forma secuencial, PostgreSQL puede utilizar la memoria caché del sistema operativo para proporcionar un acceso más rápido. De forma predeterminada, `seq_page_cost` se establece en 1.0, lo que supone que el acceso secuencial a las páginas es

tan económico como la lectura de un solo bloque de disco. Si tiene una tabla a la que se accede principalmente de forma secuencial, pero el acceso al disco es lento porque está limitado por un menor número de IOPS, puede que desee aumentar el valor de `seq_page_cost` para reflejar el costo adicional del acceso al disco.

El cambio del valor de este parámetro afecta a todas las consultas que se ejecutan en el sistema, por lo que le recomendamos que pruebe las consultas con valores diferentes para determinar el valor óptimo para su caso de uso específico.

AWS CLI sintaxis

El siguiente comando cambia `seq_page_cost` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify seq_page_cost on a DB parameter group
aws rds modify-db-parameter-group \
   --db-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=seq_page_cost,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify seq_page_cost on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
   --db-cluster-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=seq_page_cost,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: 1.0

Ejemplo

Supongamos que tiene una base de datos que almacena una gran cantidad de datos en una tabla a la que se accede principalmente de forma secuencial. La tabla se utiliza principalmente para elaborar informes, las consultas se realizan muy lentamente y el planificador de consultas no puede seleccionar el plan más eficaz para acceder a los datos.

Una forma de mejorar el rendimiento sería reducir el `seq_page_cost` parámetro. El valor predeterminado es 1,0, lo que supone que el acceso secuencial a una página es tan económico como la lectura de un solo bloque de disco. Sin embargo, en este caso específico, el acceso

secuencial a las páginas podría resultar más caro debido al tipo de almacenamiento (en función de las E/S). Si se establece `seq_page_cost` en 0,5, el coste del acceso secuencial a las páginas es la mitad del valor predeterminado. Este cambio puede aumentar las probabilidades de que el planificador de consultas seleccione un plan que utilice un método de acceso secuencial que se adapte mejor a las características específicas de la tabla.

Se recomienda supervisar el rendimiento de las consultas después de cambiar el parámetro y realizar los ajustes necesarios. También deberías comprobar el plan de consultas con una `EXPLAIN` declaración para comprobar si está seleccionando un plan eficiente.

Este es solo un ejemplo. La configuración óptima depende de las características específicas de su carga de trabajo. Además, este es solo un aspecto del ajuste del rendimiento; también debe tener en cuenta otros parámetros y opciones de configuración que afectan al rendimiento de las consultas.

track_activity_query_size

El `track_activity_query_size` parámetro controla el tamaño de la cadena de consulta que se registra para cada sesión activa en la vista. `pg_stat_activity` De forma predeterminada, solo los primeros 1024 bytes de la cadena de consulta se registran en Amazon RDS para PostgreSQL y los 4096 bytes se registran en Aurora, compatible con PostgreSQL. Si desea registrar consultas más largas, puede establecer este parámetro en un valor más alto.

AWS CLI sintaxis

El siguiente comando cambia `track_activity_query_size` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify track_activity_query_size on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=track_activity_query_size,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify track_activity_query_size on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=track_activity_query_size,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: estático (la aplicación de los cambios requiere un reinicio)

Valor predeterminado: 1024 bytes (Amazon RDS para PostgreSQL), 4096 bytes (compatible con Aurora PostgreSQL)

Ejemplo

El rendimiento de las consultas de su base de datos Amazon RDS for PostgreSQL es lento y sospecha que el problema podría estar relacionado con consultas de larga duración. Puede investigar más a fondo registrando las consultas más largas en la vista. `pg_stat_activity`

El aumento del valor del `track_activity_query_size` parámetro puede provocar un aumento del registro, lo que puede afectar al rendimiento de la base de datos. Se recomienda volver a establecer el parámetro en su valor predeterminado de 1.024 una vez resuelto el problema.

idle_in_transaction_session_timeout

El parámetro controla el tiempo que espera una transacción inactiva antes de detenerse`idle_in_transaction_session_timeout`.

El valor predeterminado de este parámetro en Amazon RDS para PostgreSQL y Aurora PostgreSQL compatible es de 86 400 000 milisegundos.

AWS CLI sintaxis

El siguiente comando cambia `idle_in_transaction_session_timeout` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify idle_in_transaction_session_timeout on a DB parameter group
aws rds modify-db-parameter-group \
   --db-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=idle_in_transaction_session_timeout,ParameterValue=<new_value>,ApplyMethod=imme

# Modify idle_in_transaction_session_timeout on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
   --db-cluster-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=idle_in_transaction_session_timeout,ParameterValue=<new_value>,ApplyMethod=imme
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: 86.400.000 milisegundos (compatible con Aurora PostgreSQL)

Ejemplo

Tiene una aplicación de comercio electrónico que procesa los pedidos en línea. La aplicación utiliza una base de datos PostgreSQL alojada en Amazon RDS o Aurora. Cada vez que un cliente realiza un pedido, la aplicación inicia una nueva transacción para actualizar el inventario y los registros de pedidos.

Si una transacción permanece inactiva durante mucho tiempo, puede impedir que otras transacciones accedan a los mismos registros, lo que puede provocar problemas de rendimiento e incluso el tiempo de inactividad de la aplicación. Además, las transacciones inactivas que no se detienen adecuadamente pueden consumir valiosos recursos del sistema, como la memoria y la CPU.

Para evitar estos problemas, puede establecer el `idle_in_transaction_session_timeout` parámetro en un valor que se adapte a su aplicación. Por ejemplo, puede establecerlo en 5 minutos (300 segundos) para que cualquier transacción que quede inactiva durante más de 5 minutos se detenga automáticamente. Esto puede ayudar a garantizar que los recursos del sistema se utilicen de manera eficiente y que la aplicación pueda gestionar una gran cantidad de pedidos sin ralentizarse. Al establecer el valor adecuado para`idle_in_transaction_session_timeout`, puede garantizar que su aplicación funcione de forma óptima en Amazon RDS o Aurora.

statement_timeout

El `statement_timeout` parámetro establece la cantidad máxima de tiempo que se puede ejecutar una consulta antes de que se detenga.

AWS CLI sintaxis

El siguiente comando cambia `statement_timeout` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify statement_timeout on a DB parameter group
aws rds modify-db-parameter-group \
```

```
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=statement_timeout,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify statement_timeout on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=statement_timeout,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: 0 milisegundos (sin tiempo de espera)

Ejemplo

Su aplicación web permite a los usuarios buscar en una gran base de datos de productos. A veces, las consultas de búsqueda pueden tardar mucho en completarse, lo que provoca tiempos de respuesta lentos para los usuarios. Para solucionar este problema, puede establecer el `statement_timeout` parámetro en un valor bajo, por ejemplo, 10 segundos, lo que obligaría a detener cualquier consulta que demore más de 10 segundos.

Puede parecer una medida drástica, pero en realidad puede ser muy eficaz para mejorar el rendimiento. En muchos casos, las consultas de larga duración se deben a consultas SQL mal optimizadas o a índices ineficientes. Al establecer un `statement_timeout` valor bajo, puede identificar estas consultas problemáticas y tomar medidas para optimizarlas.

Por ejemplo, supongamos que descubres que el tiempo de espera de una consulta de búsqueda concreta se agota constantemente. Puedes usar herramientas como `EXPLAIN` y `EXPLAIN ANALYZE` para analizar la consulta e identificar cualquier obstáculo en el rendimiento. Cuando haya identificado el problema, puede tomar medidas para optimizar la consulta añadiendo nuevos índices, reescribiendo la consulta o utilizando un algoritmo de búsqueda diferente. Al analizar y optimizar continuamente las consultas SQL de esta manera, puede mejorar considerablemente el rendimiento de la aplicación.

search_path

El `search_path` parámetro determina el orden en el que se buscan los objetos en los esquemas en las sentencias SQL. El valor predeterminado es`$user, public`, lo que significa que

PostgreSQL busca los objetos primero en el esquema que coincide con el nombre del usuario y, después, en el esquema público.

Si tiene una gran cantidad de esquemas o necesita acceder a los objetos de un esquema específico, cambiar el `search_path` parámetro puede ayudar a mejorar el rendimiento. Cuando se establece `search_path` un esquema específico, PostgreSQL puede encontrar los objetos más rápidamente sin tener que buscar en varios esquemas.

Para cambiar el `search_path` parámetro en Amazon RDS y Aurora, puede usar el siguiente comando para el `ROLE` nivel:

```
ALTER ROLE <username> SET search_path = <schema>;
```

AWS CLI sintaxis

El siguiente comando cambia `search_path` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify search_path on a DB parameter group
aws rds modify-db-parameter-group \
   --db-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=search_path,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify search_path on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
   --db-cluster-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=search_path,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: `$user, public`

Ejemplo

Tiene una aplicación multiusuario con esquemas independientes para cada inquilino que utiliza una base de datos compatible con Amazon RDS for PostgreSQL o Aurora PostgreSQL, y necesita ejecutar una consulta que implique unir datos de varios esquemas.

De forma predeterminada, Amazon RDS y Aurora utilizan la ruta de búsqueda para determinar qué esquema usar para una tabla determinada. La ruta de búsqueda es una lista de nombres de esquemas que PostgreSQL busca en orden cuando se hace referencia a una tabla sin especificar el nombre del esquema. De forma predeterminada, Amazon RDS y Aurora buscan primero la tabla en el esquema que tiene el mismo nombre que el usuario actual y, a continuación, buscan en el esquema público.

Supongamos que desea ejecutar una consulta que implique la unión de tablas de varios esquemas, denominadas `tenant1tenant2`, y. `tenant3` Para usar los esquemas arrendatarios, puede incluir los nombres de los esquemas en la consulta:

```
SELECT *
FROM tenant1.table1
JOIN tenant2.table2 ON tenant1.table1.id = tenant2.table2.id
JOIN tenant3.table3 ON tenant2.table2.id = tenant3.table3.id;
```

Sin embargo, un método más eficaz consiste en cambiar el `search_path` parámetro para incluir los esquemas arrendatarios mediante los comandos de la sección de AWS CLI sintaxis. También puede usar el `SET` comando en una sesión de PostgreSQL:

```
SET search_path = tenant1, tenant2, tenant3, public;
```

A continuación, puede escribir la consulta sin especificar los nombres de los esquemas:

```
SELECT *
FROM table1
JOIN table2 ON table1.id = table2.id
JOIN table3 ON table2.id = table3.id;
```

Esto puede hacer que la consulta sea más concisa y fácil de leer, y también puede simplificar el código de la aplicación si tiene muchas consultas que implican unir tablas de varios esquemas.

max_connections

El `max_connections` parámetro establece el número máximo de conexiones simultáneas para la base de datos PostgreSQL.

AWS CLI sintaxis

El siguiente comando cambia `max_connections` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify max_connections on a DB parameter group
aws rds modify-db-parameter-group \
--db-parameter-group-name <parameter_group_name> \
--parameters
 "ParameterName=max_connections,ParameterValue=<new_value>,ApplyMethod=pending-reboot"

# Modify max_connections on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
--db-cluster-parameter-group-name <parameter_group_name> \
--parameters
 "ParameterName=max_connections,ParameterValue=<new_value>,ApplyMethod=pending-reboot"
```

Tipo: estático (la aplicación de los cambios requiere un reinicio)

Valor predeterminado: `LEAST(DBInstanceClassMemory/9531392,5000)` conexiones

Para optimizar el uso de `max_connections` Amazon RDS o Aurora y minimizar su impacto en el rendimiento, tenga en cuenta las siguientes prácticas recomendadas:

- Establezca el valor del parámetro en función de los recursos del sistema disponibles.
- Supervise el uso de la conexión para evitar que se alcance el límite rápidamente.
- Utilice la agrupación de conexiones para reducir la cantidad de conexiones necesarias.
- Utilice Amazon RDS Proxy para agrupar conexiones.

Cuando ajuste Amazon RDS for PostgreSQL o Amazon Aurora PostgreSQL compatible, tenga `max_connections` en cuenta los tipos de instancias disponibles y sus recursos asignados, y céntrese en la capacidad de la memoria y la CPU. Los detalles de almacenamiento y E/S se gestionan mediante AWS, por lo que puede supervisar las características generales de la carga de trabajo y las métricas del sistema, por ejemplo, `FreeableMemory` a través de Amazon CloudWatch o la consola Amazon RDS para confirmar que hay suficiente memoria para las conexiones. Supervise `CPUUtilization` los valores altos, ya que podrían indicar que es necesario realizar ajustes. Evite un ajuste `max_connections` demasiado alto, ya que puede afectar al uso de la memoria y, potencialmente, influir en la E/S de forma indirecta. Tenga en cuenta que cada conexión consume memoria. Para encontrar el equilibrio adecuado, aumente lentamente `max_connections`

y compruebe cómo afecta a su sistema. Esté atento a las señales de un rendimiento más lento o un mayor uso de la CPU. Comprueba si la aplicación sigue funcionando bien. Utilice funciones como las réplicas de lectura en Aurora para distribuir el tráfico de lectura y reducir la carga en la instancia principal. Revíselas y ajústelas `max_connections` periódicamente en función de los patrones de uso observados para garantizar un rendimiento óptimo de la base de datos dentro de las limitaciones de recursos dadas.

Ajuste de los parámetros de autovacuum

Las bases de datos de Amazon RDS for PostgreSQL y las compatibles con Aurora PostgreSQL requieren un mantenimiento periódico conocido como aspiración. Autovacuum es una utilidad PostgreSQL integrada que elimina los datos obsoletos o innecesarios para liberar espacio en la base de datos. El proceso de autovacuum ejecuta el `VACUUM` comando en segundo plano a intervalos regulares.

Ajustar la configuración de autovacuum es un paso crucial para mantener el rendimiento, la estabilidad y la disponibilidad de su sistema de base de datos compatible con Amazon RDS for PostgreSQL o Aurora PostgreSQL. Al ajustar los parámetros de aspiración automática para adaptarlos a la carga de trabajo y al tamaño de la base de datos, puede optimizar el rendimiento del proceso de aspiración automática y reducir su impacto en los recursos del sistema, mejorando así el estado general de la base de datos.

Además de ajustar la configuración del autovacuum, es importante supervisar el rendimiento de la base de datos y sus componentes mediante las herramientas y métricas disponibles en Amazon RDS y Aurora. Al monitorear las métricas de rendimiento, como la sobrecarga, el espacio libre y los tiempos de ejecución de las consultas, puede identificar posibles problemas antes de que se conviertan en problemas graves y tomar las medidas adecuadas para resolverlos.

En esta sección se analizan los siguientes temas y parámetros de la aspiradora automática:

- comandos VACUUM y ANALYZE
- Comprobando si hay hinchazón
- aspiradora automática
- autovacuum_work_mem
- autovacuum_naptime
- autovacuum_max_workers
- factor de escala de vacío automático
- umbral de vacío automático
- autovacuum_analyze_scale_factor
- umbral de análisis de autovacío
- autovacuum_vacuum_cost_limit

Para obtener información adicional sobre la aspiradora automática, consulte los siguientes enlaces:

- Descripción del autovacuum en entornos Amazon RDS para PostgreSQL (entrada del blog)
- Uso de la aspiradora automática de PostgreSQL en Amazon RDS para PostgreSQL (documentación de Amazon RDS)
- Aspiración paralela en Amazon RDS para PostgreSQL y Amazon Aurora PostgreSQL (entrada del blog)

comandos VACUUM y ANALYZE

`VACUUM`recolecta basura y, opcionalmente, analiza una base de datos. Para la mayoría de las aplicaciones, basta con dejar que el demonio de la aspiradora automática realice la aspiración. Sin embargo, es posible que algunos administradores deseen modificar los parámetros de la base de datos para el autovacuum o complementar o reemplazar las actividades del daemon mediante `VACUUM` comandos gestionados manualmente que se puedan ejecutar según un programador.

`VACUUM`recupera el espacio de almacenamiento ocupado por tuplas muertas. En las operaciones estándar de PostgreSQL, cuando las tuplas se eliminan o se vuelven obsoletas debido a una actualización, no se eliminan físicamente de las tablas hasta `VACUUM` que se realiza una operación. Por lo tanto, le recomendamos que las ejecute `VACUUM` periódicamente, especialmente en las tablas que se actualizan con frecuencia.

El ajuste de `VACUUM` los parámetros es especialmente importante en Amazon RDS para PostgreSQL y Aurora compatible con PostgreSQL, ya que estos servicios de bases de datos gestionadas tienen características diferentes en comparación con las bases de datos PostgreSQL autogestionadas. Estas diferencias pueden afectar al rendimiento de las operaciones de vacío. El ajuste de `VACUUM` los parámetros es esencial para optimizar el uso de los recursos y garantizar que las operaciones de vacío no afecten negativamente al rendimiento y la disponibilidad del sistema de bases de datos.

Estos son algunos de los parámetros que puede usar con el `VACUUM` comando en Aurora PostgreSQL compatible y Amazon RDS for PostgreSQL:

- `FULL`
- `FREEZE`
- `VERBOSE`
- `ANALYZE`

- `DISABLE_PAGE_SKIPPING`
- `table_name`
- `column_name`

`VACUUM ANALYZE`realiza una `VACUUM` operación seguida de una operación para cada tabla seleccionada. `ANALYZE` Proporciona una forma eficiente de realizar el mantenimiento de rutina.

Si se utiliza el `VACUUM` comando sin la `FULL` opción, se recupera espacio para su reutilización. No requiere un bloqueo exclusivo sobre la mesa, por lo que puede ejecutar este comando durante las operaciones de lectura y escritura estándar. Sin embargo, en la mayoría de los casos, el comando no devuelve espacio adicional al sistema operativo, sino que lo mantiene disponible para su reutilización en la misma tabla. `VACUUM FULL`reescribe todo el contenido de la tabla en un nuevo archivo de disco sin espacio adicional y permite devolver al sistema operativo el espacio no utilizado. Este formulario es mucho más lento y requiere `ACCESS EXCLUSIVE` bloquear cada tabla.

Para obtener información completa sobre estos parámetros, consulte la documentación de PostgreSQL.

En Aurora y Amazon RDS, autovacuum es un proceso daemon (utilidad en segundo plano) que ejecuta los `ANALYZE` comandos `VACUUM` and con regularidad para limpiar los datos redundantes de la base de datos y el servidor. Incluso si utiliza la aspiración automática, le recomendamos que revise y ajuste la configuración de la aspiradora automática que se describe en las siguientes secciones para garantizar un rendimiento óptimo.

Comprobar si hay hinchazón

La siguiente consulta SQL examina cada tabla del esquema XML e identifica las filas inactivas (tuplas) que desperdician espacio en disco:

```
SELECT schemaname || '.' || relname as tuplename,
    n_dead_tup,
    (n_dead_tup::float / n_live_tup::float) * 100 as pfrag
FROM pg_stat_user_tables
WHERE schemaname = 'xml' and n_dead_tup > 0 and n_live_tup > 0 order by pfrag desc;
```

Si esta consulta devuelve un porcentaje alto (pfrag) de tuplas inactivas, puede utilizar el `VACUUM` comando para recuperar espacio.

Para supervisar el tamaño de los datos antes y después de las transacciones, ejecuta la siguiente consulta en el shell después de conectarte a una base de datos específica:

```
SELECT pg_size_pretty(pg_relation_size('table_name'));
```

autovacuum

Puede configurar el autovacuum de forma global mediante el parámetro de `autovacuum` configuración, o puede cambiarlo por tabla configurando la `autovacuum_enabled` columna de la tabla en o para `true` una `pg_class` tabla específica. `false`

Al activar el vacío automático en una tabla, el servidor de la base de datos escanea periódicamente la tabla en busca de filas y tuplas inactivas y las elimina en segundo plano, sin la intervención del administrador de la base de datos. Esto ayuda a mantener la tabla pequeña, a mejorar el rendimiento de las consultas y a reducir el tamaño de las copias de seguridad.

AWS CLI sintaxis

El siguiente comando habilita un grupo `autovacuum` de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters "ParameterName=autovacuum,ParameterValue=true,ApplyMethod=immediate"

# Modify autovacuum on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters "ParameterName=autovacuum,ParameterValue=true,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: activado

También puedes activar o desactivar el autovacuum en una tabla específica mediante psql:

```
ALTER TABLE <table_name> SET (autovacuum_enabled = true);
```

Aspirar demasiado puede afectar al rendimiento, por lo que es importante supervisar el rendimiento del proceso de aspiración automática, así como el rendimiento de la base de datos, y ajustar la configuración según sea necesario.

Ejemplo

La base de datos PostgreSQL tiene una tabla que recibe un gran volumen de operaciones de escritura y eliminación. Sin el autovacuum, esta tabla acabaría llenándose de filas muertas (es decir, filas que se han marcado para su eliminación pero que aún no se han eliminado físicamente de la tabla). Estas filas inactivas ocuparían espacio en el disco, ralentizarían las consultas y aumentarían el tamaño de las copias de seguridad. Puede activar la aspiradora automática de la mesa para buscar automáticamente filas inactivas y eliminarlas para mitigar estos problemas.

autovacuum_work_mem

`autovacuum_work_mem`es un parámetro de configuración de PostgreSQL que controla la cantidad de memoria que utiliza el proceso de autovacuum cuando realiza tareas de mantenimiento de tablas, como aspirar o analizar.

En Aurora y Amazon RDS, puede ajustar el valor de `autovacuum_work_mem` para optimizar el rendimiento.

AWS CLI sintaxis

El siguiente comando habilita un grupo `autovacuum_work_mem` de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_work_mem on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_work_mem,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify autovacuum_work_mem on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_work_mem,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: GREATEST({DBInstanceClassMemory/32768},131072) KB en Aurora compatible con PostgreSQL, 64 MB en Amazon RDS para PostgreSQL. Sin embargo, el valor predeterminado puede variar en función de la versión específica de Amazon RDS o Aurora que utilice.

Ejemplo

La base de datos de Amazon RDS for PostgreSQL tiene una tabla grande que se actualiza con frecuencia. Con el tiempo, observa que la base de datos se vuelve más lenta y sospecha que el autovacuum tarda demasiado en completarse.

Como parte de la investigación, comprueba los registros del sistema, utiliza la pg_stat_activity vista para ver qué consultas y procesos se están ejecutando actualmente, comprueba la pg_stat_user_tables vista para ver las estadísticas de cada tabla, utiliza la pg_settings vista para comparar el valor de autovacuum_work_mem con la memoria disponible en el sistema y monitorea el uso de la memoria para detectar picos. Tras recopilar esta información, puede establecer autovacuum_work_mem el valor óptimo que necesite su carga de trabajo. Para encontrar el equilibrio adecuado entre el uso de la memoria y el rendimiento, puede decidir configurarlo en una cuarta parte de la memoria disponible en el sistema. Después de cambiar el valor, supervisa el rendimiento de la base de datos y es posible que observe que el autovacuum se completa mucho más rápido que antes y que la base de datos funciona más rápido en general.

autovacuum_naptime

El autovacuum_naptime parámetro controla el intervalo de tiempo entre ejecuciones sucesivas del proceso de aspiración automática. El valor predeterminado es 15 segundos para Amazon RDS para PostgreSQL y 5 segundos para Aurora compatible con PostgreSQL.

Por ejemplo, supongamos que su base de datos de Amazon RDS for PostgreSQL tiene una tabla que recibe un gran volumen de operaciones de escritura y eliminación. Si mantiene la configuración predeterminada, los frecuentes escaneos con vacío automático afectarán a esta tabla altamente transactiva. Si establece este parámetro en un valor alto, habrá un intervalo más largo entre escaneos sucesivos y las filas muertas se eliminarán con menos frecuencia.

Puede utilizarlo autovacuum_naptime para gestionar la carga provocada por el proceso de vacío, especialmente si tiene un servidor ocupado que ya tiene una gran carga de CPU o E/S. Cuanto más

tiempo establezca la hora de la siesta, con menos frecuencia se ejecutará la aspiradora automática, lo que reduce la carga del servidor. Sin embargo, si se establece `autovacuum_naptime` un valor muy alto, las tablas de PostgreSQL pueden crecer y las filas muertas se acumularán, lo que provocará una disminución del rendimiento. Le recomendamos que supervise el rendimiento del proceso de aspiración automática y que ajuste la `autovacuum_naptime` configuración según sea necesario.

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_naptime` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_naptime on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_naptime,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify autovacuum_naptime on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_naptime,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: 15 segundos (Amazon RDS para PostgreSQL), 5 segundos (compatible con Aurora PostgreSQL)

autovacuum_max_workers

El `autovacuum_max_workers` parámetro controla el número máximo de procesos de trabajo que puede crear el proceso de autovacuum. Cada proceso de trabajo es responsable de aspirar o analizar una sola mesa.

Por ejemplo, supongamos que tiene una base de datos grande con muchas tablas que se actualizan y eliminan con frecuencia. Si establece un valor bajo, como 1, solo se podrá aspirar

una tabla a la vez y se tardará más en limpiar todas las tablas. `autovacuum_max_workers` Si se establece `autovacuum_max_workers` un valor alto, como 8, se pueden aspirar hasta ocho mesas simultáneamente. Esto puede acelerar el proceso de limpieza de las bases de datos que contienen muchas tablas.

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_max_workers` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_max_workers on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_max_workers,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify autovacuum_max_workers on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_max_workers,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: estático (la aplicación de los cambios requiere un reinicio)

Valor predeterminado: `GREATEST(DBInstanceClassMemory/64371566592,3)` trabajadores

Si se aumenta la `autovacuum_max_workers` configuración, se puede aumentar la carga del servidor, lo que puede afectar al rendimiento si no se dispone de recursos suficientes. La configuración óptima depende de los requisitos específicos de la base de datos, de su tamaño y del número de tablas que contenga. Le recomendamos que experimente con diferentes valores y supervise el rendimiento para encontrar la configuración óptima para su caso de uso.

autovacuum_vacuum_scale_factor

El parámetro de `autovacuum_vacuum_scale_factor` configuración controla la intensidad del proceso de aspiración automática al aspirar una mesa.

El factor de escala de vacío es una fracción del número total de tuplas de una tabla que deben modificarse antes de que la aspiradora automática limpie la mesa. El valor predeterminado es

0,1 (es decir, se debe modificar el 10 por ciento de las tuplas). Por ejemplo, si una tabla tiene 1 000 000 de tuplas y 100 000 de esas tuplas están marcadas como muertas o eliminadas, la aspiradora automática aspira la tabla en función del valor de como factor de control. `autovacuum_vacuum_threshold`

El `autovacuum_vacuum_scale_factor` parámetro le ayuda a controlar la frecuencia con la que se ejecuta el proceso de vacío. Si una tabla recibe muchas operaciones de escritura, puede que desee reducir el factor de escala de vacío para que la aspiradora automática funcione con más frecuencia y mantener la tabla más pequeña. Por el contrario, si una tabla recibe pocas operaciones de escritura, puede que desee aumentar el factor de escala de vacío para que el vacío automático se ejecute con menos frecuencia y ahorrar recursos.

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_vacuum_scale_factor` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_vacuum_scale_factor on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_vacuum_scale_factor,ParameterValue=<new_value>,ApplyMethod=immediate

# Modify autovacuum_vacuum_scale_factor on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_vacuum_scale_factor,ParameterValue=<new_value>,ApplyMethod=immediate
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: 0.1

El `autovacuum_vacuum_scale_factor` parámetro funciona junto con los `autovacuum_vacuum_cost_limit, and autovacuum_naptime` parámetros`autovacuum_vacuum_threshold`,. Para obtener información adicional sobre este parámetro, consulte la entrada del AWS blog Cómo entender el autovacuum en entornos Amazon RDS for PostgreSQL.

autovacuum_vacuum_threshold

El `autovacuum_vacuum_threshold` parámetro controla el número mínimo de operaciones de actualización o eliminación de tuplas que deben realizarse en una tabla antes de que la aspiradora automática la vacíe. Este ajuste puede resultar útil para evitar tener que aspirar innecesariamente en mesas que no tengan una alta tasa de estas operaciones. El valor predeterminado es 50, que es el valor predeterminado del motor PostgreSQL, tanto para Amazon RDS for PostgreSQL como para Aurora compatible con PostgreSQL.

Por ejemplo, supongamos que tiene una tabla con 100 000 filas y está establecida en 50. `autovacuum_vacuum_threshold` Si la tabla recibe solo 49 actualizaciones o eliminaciones, la aspiradora automática no la aspirará. Si la tabla recibe 50 o más actualizaciones o eliminaciones, la aspiradora automática la aspirará en función del factor de control `autovacuum_vacuum_scale_factor` multiplicado por el número de filas de la tabla.

Si se establece este parámetro en un valor demasiado alto, es posible que la tabla crezca y se acumulen filas muertas, lo que puede afectar al rendimiento.

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_vacuum_threshold` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_vacuum_threshold on a DB parameter group
aws rds modify-db-parameter-group \
   --db-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=autovacuum_vacuum_threshold,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify autovacuum_vacuum_threshold on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
   --db-cluster-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=autovacuum_vacuum_threshold,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: 50 operaciones

El `autovacuum_vacuum_threshold` parámetro funciona junto con los `autovacuum_vacuum_cost_limit, and autovacuum_naptime` parámetros`autovacuum_vacuum_scale_factor`,. La configuración óptima depende de los requisitos específicos de la base de datos y del tamaño de la tabla.

Para obtener información adicional sobre este parámetro, consulte la entrada del AWS blog Cómo entender el autovacuum en entornos Amazon RDS for PostgreSQL.

autovacuum_analyze_scale_factor

El `autovacuum_analyze_scale_factor` parámetro controla qué tan agresivo debe ser el proceso de autovacío a la hora de analizar (recopilar) estadísticas sobre la distribución de los datos en una tabla.

El proceso de aspiración automática utiliza este parámetro para calcular un umbral en función del número de tuplas de una tabla. Si el número de tuplas insertadas, actualizadas o eliminadas supera este umbral, autovacuum analiza la tabla. El valor predeterminado es 0,05 (es decir, se debe modificar el 5 por ciento de las tuplas) tanto para Amazon RDS for PostgreSQL como para Aurora PostgreSQL compatible.

Por ejemplo, supongamos que su tabla tiene 1 000 000 de tuplas y usted mantiene el valor predeterminado en 0,05. `autovacuum_analyze_scale_factor` Si la tabla recibe 50 000 o más actualizaciones o eliminaciones, la aspiradora automática la aspira en función del `autovacuum_analyze_threshold` valor y sumando el número de filas de la tabla como factor de control.

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_analyze_scale_factor` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_analyze_scale_factor on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_analyze_scale_factor,ParameterValue=<new_value>,ApplyMethod=immediat

# Modify autovacuum_analyze_scale_factor on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
```

```
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
"ParameterName=autovacuum_analyze_scale_factor,ParameterValue=<new_value>,ApplyMethod=immediat
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: 0,05 (5 por ciento)

Es esencial que el planificador de consultas recopile estadísticas para poder tomar decisiones informadas, por ejemplo, sobre cómo acceder a los datos y cómo organizarlos, por lo que le recomendamos que supervise el rendimiento del proceso de aspiración automática y ajuste la configuración según sea necesario para garantizar que las estadísticas estén actualizadas.

El `autovacuum_analyze_scale_factor` parámetro funciona junto con los `autovacuum_analyze_threshold` `autovacuum_analyze_cost_limit, and` `autovacuum_naptime` parámetros. La configuración óptima depende de los requisitos específicos de la base de datos y del tamaño de la tabla, así como de la frecuencia de las actualizaciones. Para obtener información adicional sobre este parámetro, consulte la entrada del AWS blog Cómo entender el autovacuum en entornos Amazon RDS for PostgreSQL.

autovacuum_analyze_threshold

`autovacuum_vacuum_threshold`El parámetro es similar a. `autovacuum_analyze_threshold` Controla el número mínimo de inserciones, actualizaciones o eliminaciones de tuplas que deben producirse en una tabla antes de que Autovacuum la analice. Esta configuración puede resultar útil para evitar tener que pasar la aspiradora innecesaria en las tablas que no tienen una alta tasa de estas operaciones. El valor predeterminado es 50, que es el valor predeterminado del motor PostgreSQL, tanto para Amazon RDS for PostgreSQL como para Aurora compatible con PostgreSQL.

Por ejemplo, supongamos que tiene una tabla con 100 000 filas y mantiene el valor predeterminado en 50. `autovacuum_analyze_threshold` Si la tabla recibe solo 49 inserciones, actualizaciones o eliminaciones, Autovacuum no la analizará. Si la tabla recibe 50 o más inserciones, actualizaciones o eliminaciones, autovacuum la analizará y mantendrá el valor `autovacuum_analyze_scale_factor` multiplicado por el número de filas de la tabla como factor de control.

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_analyze_threshold` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_analyze_threshold on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_analyze_threshold,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify autovacuum_analyze_threshold on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_analyze_threshold,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: 50 operaciones

Este parámetro funciona junto con el `autovacuum_analyze_scale_factor` parámetro, así que tenga en cuenta ambos ajustes al configurar el autovacuum.

Es esencial que el planificador de consultas recopile estadísticas para poder tomar decisiones informadas, como la forma de acceder a los datos y cómo organizarlos. Si se establece un valor `autovacuum_analyze_threshold` demasiado alto, las estadísticas pueden quedar obsoletas y, por lo tanto, un rendimiento deficiente. Le recomendamos que supervise el rendimiento del proceso de aspiración automática y que ajuste los ajustes según sea necesario.

Para obtener información adicional sobre este parámetro, consulte la entrada del AWS blog Cómo entender el autovacuum en entornos Amazon RDS for PostgreSQL.

autovacuum_vacuum_cost_limit

El `autovacuum_vacuum_cost_limit` parámetro controla la cantidad de recursos de CPU y E/S que puede consumir una aspiradora automática.

Limitar el uso de recursos de los procesos de aspiración automática puede ayudar a evitar que consuman demasiadas E/S de CPU o disco, lo que podría afectar al rendimiento de otras consultas

que se ejecutan en el mismo sistema. El parámetro especifica un límite de coste, que es una unidad de trabajo que el trabajador puede realizar antes de tener que hacer una pausa y comprobar si sigue por debajo del límite. Por ejemplo, si el parámetro está establecido en 2000, el trabajador puede procesar 2000 unidades de trabajo antes de hacer una pausa.

Puede configurar el `autovacuum_vacuum_cost_limit` parámetro mediante el `SET` comando en una sesión de PostgreSQL o mediante un comando. AWS CLI

AWS CLI sintaxis

El siguiente comando cambia `autovacuum_vacuum_cost_limit` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify autovacuum_vacuum_cost_limit on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_vacuum_cost_limit,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify autovacuum_vacuum_cost_limit on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=autovacuum_vacuum_cost_limit,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los establecesApplyMethod=immediate)

Valor predeterminado: `GREATEST({log(DBInstanceClassMemory/21474836480)*600},200)` unidades de trabajo

Si establece un valor `autovacuum_vacuum_cost_limit` demasiado alto, el proceso de aspiración automática podría consumir demasiados recursos y ralentizar otras consultas. Si lo establece en un nivel demasiado bajo, es posible que el proceso de aspiración automática no recupere suficiente espacio, lo que provocará que la mesa aumente de tamaño con el tiempo. Es fundamental encontrar el equilibrio adecuado que se adapte a su sistema.

Este parámetro solo afecta al proceso de aspiración automática, no a los `VACUUM` comandos manuales. Además, solo se aplica a los procesos de autovacío para`VACUUM`, pero no para. `ANALYZE`

Ajuste de los parámetros de registro

El ajuste de los parámetros de registro en PostgreSQL ayuda a garantizar que se recopila la información correcta sin generar registros de gran tamaño que sobrecarguen el sistema.

La optimización de los parámetros de registro es crucial para equilibrar los detalles del registro con el rendimiento del sistema y el uso del disco. Puede personalizar los siguientes parámetros de registro para capturar el nivel de detalle adecuado en los registros, diagnosticar problemas e investigar los incidentes de forma eficaz y, al mismo tiempo, minimizar el impacto en el rendimiento del sistema y el uso del disco.

- rds.force_autovacuum_logging
- rds.force_admin_logging_level
- duración_registro
- declaración log_min_duration_
- log_error_verbosidad
- declaración_registro
- log_statement_stats
- declaración log_min_error_
- log_min_messages
- log_temp_files
- conexiones de registro
- log_disconnections

Estos parámetros se analizan con más detalle en las siguientes secciones.

> **Warning**
>
> La mejor configuración para estos parámetros depende de las políticas y los requisitos de conformidad de la organización. Sin embargo, habilitar los parámetros de registro puede generar una gran cantidad de registros y mensajes, lo que puede consumir espacio de almacenamiento y afectar al rendimiento, especialmente en el caso de una base de datos muy ocupada. Le recomendamos que utilice estos parámetros con cuidado. Por ejemplo,

puede decidir habilitarlos temporalmente para reducir un problema con una sentencia SQL de rendimiento lento y desactivarlos cuando finalice el período de supervisión.

rds.force_autovacuum_logging

El `rds.force_autovacuum_logging` parámetro (disponible solo en Amazon RDS para PostgreSQL) controla si las acciones de autovacuum se registran en el registro del servidor. Sus valores son`disabled`,,`debug5`,`debug4`,`debug3`,`debug2`,`debug1`,,`info`, `noticewarning`,`error`. `log` `fatal` `panic` El valor predeterminado es `warning`.

Al activar`rds.force_autovacuum_logging`, se registran todas las acciones del proceso de aspiración automática, como cuándo comienza el proceso, cuándo termina y cuántas filas aspira. Esto resulta útil para depurar o solucionar problemas de rendimiento de la aspiradora automática.

AWS CLI sintaxis

El siguiente comando cambia `rds.force_autovacuum_logging` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify rds.force_autovacuum_logging on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=rds.force_autovacuum_logging,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify rds.force_autovacuum_logging on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=rds.force_autovacuum_logging,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: `warning`

Ejemplo

Puede utilizar el `rds.force_autovacuum_logging` parámetro para analizar el rendimiento de la aspiradora automática en una mesa que tenga una tasa de escritura muy alta. Por ejemplo, si la

tabla recibe un gran número de operaciones de escritura y borrado por segundo y el rendimiento es lento, puede activar el parámetro para registrar las horas de inicio y finalización de cada ejecución de la aspiradora automática y determinar cuántas filas se aspiraron. Esto puede proporcionar información valiosa sobre la frecuencia con la que se ejecuta la aspiradora automática, el tiempo que tarda en funcionar y el número de filas que aspira. A continuación, puede utilizar esta información para ajustar con precisión los ajustes de la aspiradora automática, por ejemplo `autovacuum_vacuum_scale_factorautovacuum_vacuum_threshold`, y para optimizar el rendimiento. `autovacuum_naptime`

rds.force_admin_logging_level

El `rds.force_admin_logging_level` parámetro (disponible solo en Amazon RDS para PostgreSQL) controla el nivel de detalle de los registros que generan las operaciones administrativas, como la extracción, el análisis y la reindexación. Acepta los valores`debug5`,,,`debug4`,`debug3`,,`debug2`,`debug1`,`log`, `info` `notice` `warning``error`, `log` y (predeterminado). `fatal` `off` La configuración óptima depende del caso de uso. Por ejemplo, si está solucionando un problema, puede que desee establecer el parámetro en un nivel de depuración. De lo contrario, puede usar la `warning` configuración `loginfo`, o.

Si lo establece `rds.force_admin_logging_level``debug1`, puede registrar información detallada para una operación de reindexación, como las horas de inicio y finalización, el número de filas procesadas y cualquier error o advertencia que se produzca durante el proceso. Esto puede proporcionar información valiosa sobre el rendimiento del proceso de reindexación y ayudarle a solucionar cualquier problema que se produzca.

AWS CLI sintaxis

El siguiente comando cambia `rds.force_admin_logging_level` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify rds.force_admin_logging_level on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=rds.force_admin_logging_level,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify rds.force_admin_logging_level on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
```

```
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
"ParameterName=rds.force_admin_logging_level,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: off

Ejemplo

Se puede utilizar rds.force_admin_logging_level para supervisar y analizar el rendimiento de las operaciones administrativas en varias tablas de una base de datos grande. Por ejemplo, supongamos que tiene una base de datos grande con muchas tablas y desea optimizar el rendimiento de estas tablas mediante la ejecución periódica de operaciones de aspiración y análisis en ellas. Si establece el rds.force_admin_logging_level parámetro en info olog, puede registrar las horas de inicio y finalización de cada operación y las tablas afectadas. Puede utilizar esta información para realizar un seguimiento del rendimiento de las operaciones administrativas en distintas tablas e identificar las tablas que podrían requerir un mantenimiento más frecuente o intensivo.

Algunos de los niveles de registro generan una gran cantidad de archivos de registro y mensajes que pueden ocupar espacio en disco rápidamente, especialmente si tiene una base de datos muy ocupada. Le recomendamos que utilice este parámetro con cuidado y que lo desactive cuando finalice el período de supervisión.

log_duration

El log_duration parámetro controla si la duración de cada consulta (es decir, el tiempo que tarda en ejecutarse) se registra con la consulta. Si se establece este parámetro enon, el tiempo que tarda en ejecutarse cada consulta se incluye en la salida del registro junto con el texto de la consulta. El tiempo se mide en milisegundos.

El principal caso de uso del log_duration parámetro es ayudar a ajustar el rendimiento y solucionar problemas. Al registrar la duración de cada consulta, puede identificar las consultas que tardan más en ejecutarse y, a continuación, centrar sus esfuerzos en optimizarlas. Esto puede ayudarle a identificar y corregir los cuellos de botella en el rendimiento y a mejorar el rendimiento general de la base de datos.

AWS CLI sintaxis

El siguiente comando cambia `log_duration` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_duration on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_duration,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_duration on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_duration,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: `off`

Ejemplo

Puede usar este parámetro si sospecha que una consulta específica o un conjunto de consultas están causando problemas de rendimiento. Al habilitar el `log_duration` parámetro y examinar la salida del registro, puede ver qué consultas tardan más en ejecutarse y, a continuación, tomar las medidas adecuadas, como optimizar los índices, agregar nuevos índices o volver a escribir la consulta.

Si `log_duration` se habilita, se puede aumentar el volumen de la salida del registro. Le recomendamos que lo utilice solo cuando sea necesario y que lo apague durante las operaciones estándar para evitar llenar el espacio de almacenamiento o dificultar la lectura de los registros.

log_min_duration_statement

El `log_min_duration_statement` parámetro controla el tiempo mínimo, en milisegundos, que se ejecuta una sentencia SQL antes de que se registre.

Este parámetro le ayuda a identificar las consultas de ejecución prolongada que pueden provocar problemas de rendimiento. Puede establecerlo en un valor umbral (un tiempo de ejecución que se considera demasiado largo para una carga de trabajo específica) para capturar las consultas que

superen ese umbral e identificar posibles obstáculos en el rendimiento. Para ver un ejemplo de caso de uso, consulte Uso de parámetros de registro para capturar variables de enlace, más adelante en esta guía.

AWS CLI sintaxis

El siguiente comando cambia `log_min_duration_statement` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_min_duration_statement on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_min_duration_statement,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_ min_duration_statement on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_min_duration_statement,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: 1 (deshabilitado, que es el valor predeterminado del motor PostgreSQL)

Ejemplo

El siguiente comando registra cualquier sentencia que tarde más de 100 milisegundos en ejecutarse:

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name <parameter_group_name> \
   --parameters
 "ParameterName=log_min_duration_statement,ParameterValue=100,ApplyMethod=immediate"
```

log_error_verbosity

El `log_error_verbosity` parámetro controla el nivel de detalle incluido en la salida del registro para los errores y los mensajes que se registran con ese nivel de error o superior. Este parámetro puede tomar uno de estos tres valores:`terse`,`default`, `overbose`.

- `terse`incluye solo el texto del mensaje, el nivel de error y el número de archivo y línea en los que se produjo el error.
- `default`incluye el texto del mensaje, el nivel de error, el número de archivo y línea y el contexto del error.
- `verbose`incluye el texto del mensaje, el nivel de error, el número de archivo y línea, el contexto del error y el mensaje de error completo.

Defina el parámetro en `verbose` para obtener la información más detallada para la solución de problemas y la depuración en un entorno que no sea de producción. En un entorno de producción, es posible que desee configurarlo para `default terse` que solo proporcione información esencial y no llene el almacenamiento de registros con demasiados detalles.

AWS CLI sintaxis

El siguiente comando cambia `log_error_verbosity` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_error_verbosity on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_error_verbosity,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_error_verbosity on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_error_verbosity,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: `default`

log_statement

El `log_statement` parámetro controla qué sentencias SQL se registran en el registro del servidor. El parámetro puede tomar uno de los siguientes valores:

- none(predeterminado) no registra ninguna sentencia
- ddlregistra solo las sentencias del lenguaje de definición de datos (DDL), como y CREATE TABLE ALTER TABLE
- modregistra solo las declaraciones que modifican datosINSERT, como, y UPDATE DELETE
- allregistra todas las sentencias SQL

Puede usar el log_statement parámetro para controlar la cantidad de información que se escribe en el registro documentando solo los tipos específicos de declaraciones que son relevantes para su caso de uso.

AWS CLI sintaxis

El siguiente comando cambia log_statement para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_statement on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_statement,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_statement on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_statement,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: none

Ejemplo

En un entorno de producción, es posible que desee configurar esta opción log_statement ddl para registrar únicamente las sentencias DDL y realizar un seguimiento de los cambios realizados en el esquema de la base de datos. En un entorno de desarrollo, es posible que desee configurar el parámetro para registrar todas las sentencias all a fin de facilitar la depuración y la solución

de problemas. Para ver otro ejemplo de caso de uso, consulte Uso de parámetros de registro para capturar variables de enlace, más adelante en esta guía.

La activación `log_statement` puede aumentar el volumen de salida de registros, así que úsala solo cuando sea necesario y desactívala para evitar que se llene el espacio de almacenamiento o dificulte la lectura de los registros.

Le recomendamos que supervise el sistema y ajuste el valor de este parámetro para lograr el equilibrio adecuado entre la cantidad de información registrada y el almacenamiento y el rendimiento del sistema.

log_statement_stats

El `log_statement_stats` parámetro controla si las estadísticas asociadas a la ejecución de una sentencia SQL se registran con la sentencia. Al activar este parámetro, en la salida del registro se incluyen estadísticas como el número de filas afectadas, el número de bloques de disco leídos y escritos y el tiempo que tarda en ejecutarse la sentencia.

Puede usar el `log_statement_stats` parámetro para recopilar información adicional sobre el rendimiento de las declaraciones individuales y la carga de trabajo general. Al registrar las estadísticas de las declaraciones, puede identificar patrones en el rendimiento de las consultas y el uso de los recursos, y utilizar esa información para optimizar la base de datos y mejorar el rendimiento general.

AWS CLI sintaxis

El siguiente comando cambia `log_statement_stats` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_statement_stats on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_statement_stats,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_statement_stas on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
```

```
  --parameters
 "ParameterName=log_statement_stats,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: off (predeterminado del motor PostgreSQL); utilice 0 o 1 (booleano) para establecer los grupos de parámetros

Ejemplo

Se puede utilizar log_statement_stats para analizar el comportamiento de una consulta específica, ver cómo utiliza recursos como la CPU, la memoria y las E/S del disco, e identificar si la consulta se puede optimizar. También puede usar este parámetro para ver si una tabla específica se lee con frecuencia (lo que podría indicar la necesidad de crear un índice en una columna específica) o si la tabla se escanea con demasiada frecuencia.

Si log_statement_stats se habilita, se puede aumentar el volumen de salida del registro, así que úselo solo cuando sea necesario y desactívelo para evitar que se llene el espacio de almacenamiento o dificulte la lectura de los registros.

log_min_error_statement

El log_min_error_statement parámetro controla qué sentencias SQL que generen un error se registrarán. Sus valores son debug5debug4,debug3, debug2debug1,info,notice,warning,error, logfatal, ypanic. Esta configuración controla la cantidad de información que se escribe en el registro para que pueda filtrar los mensajes de menor gravedad. Puede establecer este parámetro en un nivel de gravedad más alto para reducir la cantidad de salida del registro y encontrar los mensajes importantes con mayor facilidad.

AWS CLI sintaxis

El siguiente comando cambia log_min_error_statement para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_min_error_statement on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_min_error_statement,ParameterValue=<new_value>,ApplyMethod=immediate"
```

```
# Modify log_min_error_statement on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_min_error_statement,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: `error` (predeterminado del motor PostgreSQL)

Ejemplo

Puede considerar usarlo `log_min_error_statement` cuando solucione un problema específico y desee ver los mensajes de error de las sentencias SQL que causan errores.

log_min_messages

El `log_min_messages` parámetro controla el nivel de gravedad que se escribe en el registro. Puede establecer el parámetro en`debug5`,,`debug4`, `debug3debug2`,`debug1`,`info`,`notice`,`warning`,`error`,`log`,`fatal`, o`panic`. Esta configuración controla la cantidad de información que se escribe en el registro para que pueda filtrar los mensajes de menor gravedad. Puede establecer este parámetro en un nivel de gravedad más alto para reducir la cantidad de salida del registro y encontrar los mensajes importantes con mayor facilidad.

AWS CLI sintaxis

El siguiente comando cambia `log_min_messages` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_min_messages on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_min_messages,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_min_messages on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
```

```
  --parameters
 "ParameterName=log_min_messages,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configurasApplyMethod=immediate)

Valor predeterminado: notice

Ejemplo

Si estás solucionando un problema específico y quieres ver todos los mensajes de error, puedes configurar este parámetro error para que registre solo los errores y los problemas de mayor gravedad. Si está interesado en supervisar el rendimiento del sistema, puede configurar este parámetro en info para ver información más detallada, como la duración y las estadísticas de cada sentencia.

Si log_min_messages se establece un nivel de gravedad más alto, se reduce el volumen de registros. Se recomienda ajustar este parámetro en función de su caso de uso específico, del tamaño del registro que desee comprobar y de la cantidad de espacio en disco del que disponga.

log_temp_files

El log_temp_files parámetro controla el registro de los nombres y tamaños de los archivos temporales. Se aplica a los archivos temporales creados con fines tales como la ordenación, los códigos hash y los resultados de consultas temporales. Cuando este parámetro está activado, se genera una entrada de registro para cada archivo temporal al eliminarlo, incluido su tamaño de archivo, en bytes. Puede establecer este parámetro en 0 (cero) para un registro completo de toda la información de los archivos temporales o en un valor positivo para registrar los archivos que superen ese tamaño (en kilobytes, si no se especifican las unidades). Esto puede resultar útil para identificar y resolver los cuellos de botella en el rendimiento u otros problemas relacionados con el almacenamiento temporal. De forma predeterminada, el registro de archivos temporales está desactivado.

AWS CLI sintaxis

El siguiente comando cambia log_temp_files para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_temp_files on a DB parameter group
```

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_temp_files,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_temp_files on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_temp_files,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: -1 (valor predeterminado del motor PostgreSQL)

Ejemplo

Puede activar este parámetro si sospecha que el sistema está utilizando demasiado almacenamiento temporal o que los archivos temporales no se están eliminando correctamente. Al examinar el resultado del registro, puede ver las consultas u operaciones que generan los archivos temporales y el uso que se hacen de estos archivos.

Algunas consultas u operaciones crean una gran cantidad de archivos temporales, por lo que su activación `log_temp_files` podría afectar al rendimiento general del sistema.

log_connections

El `log_connections` parámetro controla si se registran las conexiones a la base de datos. Si se establece este parámetro enon, el registro contiene información sobre cada conexión correcta a la base de datos, como la dirección IP del cliente, el nombre de usuario, el nombre de la base de datos y la fecha y hora de la conexión.

Puede utilizar el `log_connections` parámetro para supervisar y solucionar problemas de las conexiones a la base de datos. Puede ver los usuarios, las aplicaciones, los terminales y los bots que se conectan a la base de datos, desde dónde se conectan y con qué frecuencia. Esta información puede resultar útil para identificar y resolver problemas relacionados con la conexión o para rastrear los patrones de uso.

AWS CLI sintaxis

El siguiente comando cambia `log_connections` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_connections on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_connections,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_connections on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_connections,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: `off` (predeterminado del motor PostgreSQL)

Ejemplo

Puede usar este parámetro si sospecha que demasiadas conexiones a la base de datos o un usuario o una dirección IP específicos que se conectan con demasiada frecuencia están afectando al rendimiento. Al habilitar el `log_connections` parámetro y examinar la salida del registro, puede ver el número y los detalles de todas las conexiones.

Antes de activar este parámetro, compruebe las políticas de su organización y tenga en cuenta las implicaciones de seguridad del registro de direcciones IP y nombres de usuario.

log_disconnections

El `log_disconnections` parámetro controla el registro de las desconexiones de la base de datos. Si se establece este parámetro enon, registra la información sobre el final de cada sesión, como la dirección IP del cliente, el nombre de usuario, el nombre de la base de datos y la fecha y hora de la desconexión.

Puede usar el `log_disconnections` parámetro para supervisar y solucionar problemas relacionados con las terminaciones de las sesiones de la base de datos. Puede ver los usuarios,

las aplicaciones, los terminales y los bots que se desconectan de la base de datos, cuándo y por qué. Por ejemplo, puede revisar las terminaciones inesperadas, como una caída o una desconexión iniciada por el administrador. Esta información puede resultar útil para identificar y resolver problemas relacionados con las desconexiones o para realizar un seguimiento de los patrones de uso.

AWS CLI sintaxis

El siguiente comando cambia `log_disconnections` para un grupo de parámetros de base de datos específico. Este cambio se aplica a todas las instancias o clústeres que utilizan el grupo de parámetros.

```
# Modify log_disconnections on a DB parameter group
aws rds modify-db-parameter-group \
  --db-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_disconnections,ParameterValue=<new_value>,ApplyMethod=immediate"

# Modify log_disconnections on a DB cluster parameter group
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name <parameter_group_name> \
  --parameters
 "ParameterName=log_disconnections,ParameterValue=<new_value>,ApplyMethod=immediate"
```

Tipo: dinámico (los cambios se aplican inmediatamente si los configuras`ApplyMethod=immediate`)

Valor predeterminado: `off` (predeterminado del motor PostgreSQL)

Ejemplo

Puede usarlo `log_disconnections` si sospecha que hay demasiados usuarios que se están desconectando de la base de datos o que un usuario o una dirección IP específicos se desconectan con demasiada frecuencia. Al habilitar el `log_disconnections` parámetro y examinar la salida del registro, puede ver el número y los detalles de todas las desconexiones, incluidos quién, cuándo y si se produjo algún error antes de la desconexión.

Antes de activar este parámetro, compruebe las políticas de su organización y tenga en cuenta las implicaciones de seguridad del registro de direcciones IP y nombres de usuario.

Uso de parámetros de registro para capturar variables de enlace

Un caso de uso típico para capturar variables de enlace en PostgreSQL es depurar y ajustar el rendimiento de las consultas SQL. Una variable de enlace le permite pasar datos a una consulta cuando la ejecuta. Al capturar las variables de enlace, puede ver los datos de entrada que se pasaron a una consulta, lo que puede ayudarle a identificar cualquier problema con los datos o con el rendimiento de la consulta. La captura de las variables de enlace también puede ayudarle a auditar los datos de entrada y a detectar posibles riesgos de seguridad o actividades maliciosas.

Hay varias formas de capturar variables de enlace para PostgreSQL. Un método consiste en habilitar los parámetros `debug_print_parse` y`debug_print_rewritten`. Esto hace que PostgreSQL envíe las versiones analizadas y reescritas de las sentencias SQL, junto con las variables enlazadas, al registro del servidor.

- `debug_print_parse`: Al habilitar este parámetro, el árbol de análisis de las consultas entrantes se imprime en el registro del servidor. Esto puede resultar útil para comprender la estructura de una consulta y los valores de cualquier parámetro enlazado.
- `debug_print_rewritten`: Al habilitar este parámetro, las formas reescritas de las consultas entrantes se imprimen en el registro del servidor. Esto puede resultar útil para entender cómo el planificador de consultas interpreta una consulta y los valores de cualquier parámetro enlazado.

Puede usar dos parámetros adicionales en Amazon RDS y Aurora para capturar variables de enlace en sus bases de datos de PostgreSQL:

- `log_min_duration_statement`: Este parámetro establece la duración mínima de una sentencia antes de que se registre, en milisegundos. Cuando una sentencia tarda más de lo especificado, sus valores de enlace se incluyen en la salida del registro.
- `log_statement`: Este parámetro controla qué sentencias SQL se registran. Establezca este parámetro en `all` o enlace para incluir los valores enlazados en el registro. El aumento del nivel de registro afecta al rendimiento, por lo que se recomienda revertir los cambios después de solucionar el problema.

También puede usar la `pg_stat_statements` extensión, que proporciona estadísticas de rendimiento para todas las sentencias SQL ejecutadas por un servidor, incluidos el texto de la consulta y los valores enlazados. Esta extensión le permite utilizar pgAdmin o herramientas similares para supervisar y analizar el rendimiento de las consultas.

Otra opción es utilizar la `pg_bind_parameter_status()` función para obtener los valores de los parámetros enlazados a partir de una sentencia preparada o utilizar la `pg_get_parameter_status (paramname)` función para recuperar el estado o el valor de un parámetro de tiempo de ejecución específico.

Además, puede utilizar herramientas de terceros, como PgBadger, para analizar los registros de PostgreSQL y extraer las variables de enlace y otra información para su posterior análisis.

Ajuste de los parámetros de replicación

En PostgreSQL, puede replicar los cambios de datos de una base de datos de PostgreSQL a otra mediante la replicación lógica en lugar de la replicación física basada en archivos. La replicación lógica utiliza el registro de escritura anticipada (WAL) para capturar los cambios y admite la replicación de tablas seleccionadas o bases de datos completas.

Tanto Amazon RDS for PostgreSQL como los compatibles con Aurora PostgreSQL admiten la replicación lógica, por lo que puede configurar una arquitectura de base de datos escalable y de alta disponibilidad que pueda gestionar el tráfico de lectura y escritura de varios orígenes. Estos servicios utilizan pglogical, que es una extensión de código abierto para PostgreSQL, para implementar la replicación lógica.

Ajustar la replicación lógica en Aurora y Amazon RDS es importante para lograr un rendimiento, escalabilidad y disponibilidad óptimos. Puede ajustar los parámetros de la extensión pglogical para administrar el rendimiento de la replicación lógica. Por ejemplo, puede hacer lo siguiente:

- Mejore el rendimiento de la replicación aumentando la cantidad de procesos de trabajo o ajustando su asignación de memoria.
- Reduzca el riesgo de demoras en la replicación ajustando la frecuencia de sincronización entre las bases de datos de origen y réplica.
- Optimice el uso de los recursos ajustando la asignación de memoria y CPU de los procesos de trabajo.
- Asegúrese de que el proceso de replicación no cause un impacto indebido en el rendimiento de la base de datos de origen.

Puede usar los siguientes parámetros en Aurora y Amazon RDS para controlar y configurar la replicación lógica:

- `max_replication_slots`establece el número máximo de ranuras de replicación que se pueden crear en el servidor. Una ranura de replicación es una reserva persistente con nombre para que una conexión de replicación envíe datos de WAL a una réplica.
- `max_wal_senders`establece el número máximo de procesos de envío WAL conectados simultáneamente. Los procesos de envío de WAL se utilizan para transmitir el WAL desde el servidor principal a la réplica.

- `wal_sender_timeout`establece el tiempo máximo, en milisegundos, durante el que un remitente WAL espera una respuesta de la réplica antes de darse por vencido y volver a conectarse.
- `wal_receiver_timeout`establece el tiempo máximo, en milisegundos, durante el que una réplica espera los datos de WAL de la base de datos principal antes de que se agote el tiempo de espera.
- `log_replication_commands`, cuando se establece enon, ejecuta las sentencias SQL relacionadas con la replicación.

Al habilitar el `rds.logical_replication` parámetro (configurándolo en 1), el `wal_level` parámetro se establece en`logical`, lo que significa que todos los cambios realizados en la base de datos se escriben en el WAL en un formato que se puede leer y aplicar a una réplica. Esta configuración es necesaria para habilitar la replicación lógica. Esta configuración también permite la replicación de `SELECT` sentencias.

Si se establece `wal_level` en, `logical` se puede aumentar la cantidad de datos que se escriben en el WAL y, por lo tanto, en el disco, lo que puede afectar al rendimiento del sistema. Se recomienda tener en cuenta el espacio disponible en disco y el rendimiento del sistema al habilitar la replicación lógica.

Ejemplo

Desea replicar los datos de la base de datos principal a una base de datos secundaria con fines de copia de seguridad y recuperación ante desastres. Sin embargo, la base de datos secundaria tiene un gran volumen de operaciones de lectura, por lo que debe asegurarse de que el proceso de replicación sea lo más rápido y eficiente posible sin comprometer la integridad de los datos.

Los valores predeterminados para la replicación lógica en Amazon RDS y Aurora dan prioridad a la coherencia por encima del rendimiento, por lo que es posible que no sean óptimos para este caso de uso. Para optimizar la velocidad y la eficiencia de la configuración de replicación lógica, puede personalizar los parámetros de la siguiente manera:

- Aumente `max_replication_slots` de 10 (predeterminado para Amazon RDS) o 20 (predeterminado para Aurora) a 30 para adaptarse a las posibles necesidades futuras de crecimiento y replicación.
- Aumente `max_wal_senders` de 10 (predeterminado) a 20 para garantizar que haya suficientes procesos de envío de WAL para satisfacer la demanda de replicación.

- Disminuya `wal_sender_timeout` de 30 segundos (predeterminado) a 15 segundos para garantizar que los procesos inactivos de los remitentes WAL finalicen más rápidamente, lo que libera recursos para la replicación activa.
- Disminuya `wal_receiver_timeout` de 30 segundos (predeterminado) a 15 segundos para garantizar que los procesos del receptor WAL inactivos finalicen más rápidamente, lo que libera recursos para la replicación activa.
- Aumente `max_logical_replication_workers` de 4 (predeterminado) a 8 para garantizar que haya suficientes procesos de trabajo de replicación lógica para satisfacer la demanda de replicación.

Estas optimizaciones proporcionan una replicación de datos más rápida y eficiente, a la vez que mantienen la integridad y la seguridad de los datos.

Por ejemplo, si se produjera un desastre y la base de datos principal dejara de estar disponible, la base de datos secundaria ya tendría los datos más recientes disponibles gracias al proceso de replicación optimizado. Esto permitiría que sus operaciones empresariales siguieran proporcionando servicios críticos sin interrupciones.

Prácticas recomendadas

Ajustar la replicación lógica con cargas de trabajo enormes puede ser una tarea compleja que depende de diversos factores, como el tamaño del conjunto de datos, el número de tablas que se replican, el número de réplicas y los recursos disponibles. Estos son algunos consejos generales para ajustar la replicación lógica con cargas de trabajo enormes:

- Supervise el retraso de la replicación. El retraso de replicación es la diferencia de tiempo entre el servidor principal y los servidores en espera. Supervisar el retraso en la replicación puede ayudarlo a identificar posibles cuellos de botella y a tomar medidas para mejorar el rendimiento de la replicación. Puede utilizar la `pg_current_wal_lsn()` función para comprobar el retraso de replicación actual.
- Ajuste la configuración de WAL. La `pg_logical` extensión usa WAL para transmitir los cambios del servidor principal al servidor en espera. Si la configuración de WAL no se ajusta correctamente, la replicación puede volverse lenta y poco fiable. Asegúrese de establecer los `max_replication_slots` parámetros `max_wal_senders` y en los valores adecuados, en función de sus cargas de trabajo.

- Tenga una estrategia de indexación. Tener los índices adecuados en el servidor principal puede ayudar a mejorar el rendimiento de la replicación lógica, reducir la E/S del servidor principal y reducir la carga del sistema.
- Utilice la replicación paralela. El uso de la replicación paralela puede ayudar a aumentar la velocidad de replicación al permitir que varios procesos de trabajo paralelos repliquen los datos. Esta función está disponible en PostgreSQL 12 y versiones posteriores.

Siguientes pasos

Tras optimizar los parámetros de memoria, replicación, aspiración automática y registro de su base de datos compatible con Amazon RDS for PostgreSQL o Aurora PostgreSQL, tenga en cuenta estos pasos para mejorar aún más el rendimiento de la base de datos:

- Supervise su base de datos. Realice un seguimiento del rendimiento de su base de datos a lo largo del tiempo mediante herramientas de supervisión integradas o soluciones de terceros. Supervise las métricas clave de rendimiento, como la utilización de la CPU, la E/S del disco, el uso de la memoria y los tiempos de ejecución de las consultas, para identificar posibles obstáculos y áreas de mejora.
- Ajuste los parámetros de forma continua. A medida que su carga de trabajo evolucione, siga supervisando y ajustando los parámetros de la base de datos para garantizar un rendimiento óptimo. Revise periódicamente los registros del sistema, los mensajes de error y las métricas de rendimiento para identificar nuevas oportunidades de ajuste.
- Implemente el almacenamiento en caché. Utilice el almacenamiento en caché para reducir el número de consultas que llegan a la base de datos. Puede implementar el almacenamiento en caché a nivel de aplicación mediante herramientas como Memcached o Redis, o puede usar Amazon ElastiCache para proporcionar una caché en memoria para su base de datos.
- Optimice sus consultas. Las consultas mal diseñadas pueden afectar significativamente al rendimiento de la base de datos. Utilice `EXPLAIN` y otras herramientas de ajuste de consultas para identificar las consultas lentas, optimizarlas y eliminar las consultas innecesarias.

Si sigue estas pautas, puede optimizar el rendimiento de su base de datos Aurora o Amazon RDS for PostgreSQL y asegurarse de que satisfaga las necesidades de su aplicación con un mejor rendimiento de la base de datos, mayor confiabilidad, menor tiempo de inactividad, mayor seguridad y ahorro de costos. Al optimizar los parámetros de configuración para adaptarlos a su carga de trabajo, puede asegurarse de que su base de datos funcione de manera eficiente y utilice los recursos de manera eficaz, lo que se traduce en un mejor rendimiento y una aplicación con mayor capacidad de respuesta. Además, los parámetros correctamente configurados pueden reducir la probabilidad de errores y vulnerabilidades, lo que se traduce en una mayor fiabilidad y una mejor seguridad. Esto puede traducirse en un ahorro de costes en términos de reducción del mantenimiento y del tiempo de inactividad, así como en una mejor experiencia y satisfacción generales del usuario.

Recursos

- Parámetros de Amazon Aurora PostgreSQL, parte 1: Administración AWS de planes de memoria y consultas (entrada de blog)
- Parámetros de Amazon Aurora PostgreSQL, parte 2: Replicación, seguridad y registro AWS (entrada de blog)
- Parámetros de Amazon Aurora PostgreSQL, parte 3: parámetros del optimizador (entrada de blog)AWS
- Parámetros de Amazon Aurora PostgreSQL, parte 4: opciones de compatibilidad con ANSI (entrada de blog)AWS
- Uso de Amazon Aurora AWS PostgreSQL (documentación)
- Uso de Amazon RDS para PostgreSQL (documentación)AWS
- Supervisión de la carga de bases de datos con Performance Insights en Amazon RDS (AWS documentación)
- Uso de CloudWatch las métricas de Amazon (AWS documentación)
- pg_stats_statements (documentación de PostgreSQL)

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Información actualizada sobre los parámetros de la memoria y del autoaspirador	Se actualizó la descripción del parámetro random_page_cost; se agregaron las unidades faltantes a los valores predeterminados de los parámetros de memoria y autovacuum; se actualizó la sintaxis del AWS CLI parámetro max_connections.	27 de febrero de 2024
Información actualizada sobre `autovacuum`	Se corrigió la configuración predeterminada de la aspiradora automática (habilitada).	27 de diciembre de 2023
Información actualizada sobre `max_connections`	Se actualizó la sección max_connections con nuevas instrucciones sobre cómo ajustar este parámetro.	15 de noviembre de 2023
Publicación inicial	—	31 de octubre de 2023

AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por AWS Prescriptive Guidance. Para sugerir entradas, utilice el enlace Enviar comentarios al final del glosario.

Números

Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- Refactorizar/rediseñar: traslade una aplicación y modifique su arquitectura mediante el máximo aprovechamiento de las características nativas en la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: migre su base de datos Oracle local a la edición compatible con PostgreSQL de Amazon Aurora.
- Redefinir la plataforma (transportar y redefinir): traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: migre su base de datos Oracle local a Amazon Relational Database Service (Amazon RDS) para Oracle en el. Nube de AWS
- Recomprar (readquirir): cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: migre su sistema de gestión de relaciones con los clientes (CRM) a Salesforce.com.
- Volver a alojar (migrar mediante lift-and-shift): traslade una aplicación a la nube sin realizar cambios para aprovechar las capacidades de la nube. Ejemplo: migre su base de datos Oracle local a Oracle en una instancia EC2 del. Nube de AWS
- Reubicar: (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma local a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- Retener (revisitar): conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

A

ABAC

Consulte control de acceso basado en atributos.

servicios abstractos

Consulte servicios gestionados.

ACID

Consulte atomicidad, consistencia, aislamiento y durabilidad.

migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que la migración activa-pasiva.

migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la base de datos de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la base de datos de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

función agregada

Función SQL que opera en un grupo de filas y calcula un único valor de retorno para el grupo. Entre los ejemplos de funciones agregadas se incluyen `SUM` y`MAX`.

IA

Véase inteligencia artificial.

AIOps

Consulte las operaciones de inteligencia artificial.

anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

control de aplicaciones

Un enfoque de seguridad que permite el uso únicamente de aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para el proceso de detección y análisis de la cartera y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte ¿Qué es la inteligencia artificial?

operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo se utiliza AIOps en la estrategia de migración de AWS , consulte la Guía de integración de operaciones.

cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte ABAC AWS en la documentación AWS Identity and Access Management (IAM).

origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de ayudar a preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la Página web de AWS CAF y el Documento técnico de AWS CAF.

AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS

Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

B

Un bot malo

Un bot destinado a interrumpir o causar daño a personas u organizaciones.

BCP

Consulte la planificación de la continuidad del negocio.

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte Datos en un gráfico de comportamiento en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Véase también endianness.

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

implementación azul/verde

Una estrategia de despliegue en la que se crean dos entornos separados pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación en el otro entorno (verde). Esta estrategia le ayuda a revertirla rápidamente con un impacto mínimo.

bot

Una aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan información en Internet. Algunos otros bots, conocidos como bots malos, tienen como objetivo interrumpir o causar daños a personas u organizaciones.

botnet

Redes de bots que están infectadas por malware y que están bajo el control de una sola parte, conocida como pastor u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

rama

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte Acerca de las sucursales (GitHub documentación).

acceso con cristales rotos

En circunstancias excepcionales y mediante un proceso aprobado, un usuario puede acceder rápidamente a un sitio para el Cuenta de AWS que normalmente no tiene permisos de acceso. Para obtener más información, consulte el indicador Implemente procedimientos de rotura de cristales en la guía Well-Architected AWS .

estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección Organizado en torno a las capacidades empresariales del documento técnico Ejecutar microservicios en contenedores en AWS.

planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

C

CAF

Consulte el marco AWS de adopción de la nube.

despliegue canario

El lanzamiento lento e incremental de una versión para los usuarios finales. Cuando se tiene confianza, se despliega la nueva versión y se reemplaza la versión actual en su totalidad.

CCoE

Consulte el Centro de excelencia en la nube.

CDC

Consulte la captura de datos de cambios.

captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

ingeniería del caos

Introducir intencionalmente fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

CI/CD

Consulte la integración continua y la entrega continua.

clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las publicaciones de CCoE en el blog de estrategia Nube de AWS empresarial.

computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar conectada a la tecnología de computación perimetral.

modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte Creación de su modelo operativo de nube.

etapas de adopción de la nube

Las cuatro fases por las que suelen pasar las organizaciones cuando migran a Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realización de inversiones fundamentales para escalar la adopción de la nube (p. ej., crear una zona de aterrizaje, definir un CCoE, establecer un modelo de operaciones)
- Migración: migración de aplicaciones individuales

- Reinvención: optimización de productos y servicios e innovación en la nube

Stephen Orban definió estas etapas en la entrada del blog The Journey Toward Cloud-First & the Stages of Adoption del blog Nube de AWS Enterprise Strategy. Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de preparación para la migración.

CMDB

Consulte la base de datos de administración de la configuración.

repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Los repositorios en la nube más comunes incluyen GitHub o AWS CodeCommit. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola canalización de CI/CD puede utilizar varios repositorios.

caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

visión artificial (CV)

Campo de la IA que utiliza el aprendizaje automático para analizar y extraer información de formatos visuales, como imágenes y vídeos digitales. Por ejemplo, AWS Panorama ofrece dispositivos que añaden CV a las redes de cámaras locales, y Amazon SageMaker proporciona algoritmos de procesamiento de imágenes para CV.

desviación de configuración

En el caso de una carga de trabajo, un cambio de configuración con respecto al estado esperado. Puede provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntario.

base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

paquete de conformidad

Conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus comprobaciones de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los paquetes de conformidad en la documentación. AWS Config

integración y entrega continuas (CI/CD)

El proceso de automatización de las etapas de origen, compilación, prueba, presentación y producción del proceso de lanzamiento del software. La CI/CD se describe comúnmente como una canalización. La CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar con mayor rapidez. Para obtener más información, consulte Beneficios de la entrega continua. CD también puede significar implementación continua. Para obtener más información, consulte Entrega continua frente a implementación continua.

CV

Consulte visión artificial.

D

datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de datos es un componente del pilar de seguridad

del AWS Well-Architected Framework. Para obtener más información, consulte Clasificación de datos.

desviación de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada a lo largo del tiempo. La desviación de los datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

malla de datos

Un marco arquitectónico que proporciona una propiedad de datos distribuida y descentralizada con administración y gobierno centralizados.

minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte Crear un perímetro de datos sobre. AWS

preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

titular de los datos

Persona cuyos datos se recopilan y procesan.

almacenamiento de datos

Un sistema de administración de datos que respalde la inteligencia empresarial, como el análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para consultas y análisis.

lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

DDL

Consulte el lenguaje de definición de bases de datos.

conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

defense-in-depth

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un defense-in-depth enfoque podría combinar la autenticación multifactorial, la segmentación de la red y el cifrado.

administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta

cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte Servicios que funcionan con AWS Organizations en la documentación de AWS Organizations .

Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

entorno de desarrollo

Consulte entorno.

control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte Controles de detección en Implementación de controles de seguridad en AWS.

asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

tabla de dimensiones

En un esquema en estrella, tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos de una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se utilizan habitualmente para restringir consultas, filtrar y etiquetar conjuntos de resultados.

desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

recuperación de desastres (DR)

La estrategia y el proceso que se utilizan para minimizar el tiempo de inactividad y la pérdida de datos ocasionados por un desastre. Para obtener más información, consulte Recuperación ante desastres de cargas de trabajo en AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Consulte el lenguaje de manipulación de bases de datos.

diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Este concepto lo introdujo Eric Evans en su libro, Diseño impulsado por el dominio: abordando la complejidad en el corazón del software (Boston: Addison-Wesley Professional, 2003). Para obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

DR

Consulte recuperación ante desastres.

detección de desviaciones

Seguimiento de las desviaciones con respecto a una configuración de referencia. Por ejemplo, puedes usarlo AWS CloudFormation para detectar desviaciones en los recursos del sistema o puedes usarlo AWS Control Tower para detectar cambios en tu landing zone que puedan afectar al cumplimiento de los requisitos de gobierno.

DVSM

Consulte el mapeo del flujo de valor del desarrollo.

E

EDA

Consulte el análisis exploratorio de datos.

computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con la computación en nube, la computación perimetral puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

cifrado

Proceso informático que transforma datos de texto plano, legibles por humanos, en texto cifrado.

clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Los sistemas big-endianos almacenan primero el byte más significativo. Los sistemas Little-Endian almacenan primero el byte menos significativo.

punto de conexión

Consulte el punto final del servicio.

servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otros directores Cuentas de AWS o a AWS Identity and Access Management (IAM). Estas cuentas o entidades principales pueden conectarse a su servicio de punto de conexión de forma privada mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte Creación de un servicio de punto de conexión en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

planificación de recursos empresariales (ERP)

Un sistema que automatiza y gestiona los procesos empresariales clave (como la contabilidad, el MES y la gestión de proyectos) de una empresa.

cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el cifrado de sobres en la documentación de AWS Key Management Service (AWS KMS).

environment

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.
- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En una canalización de CI/CD, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS , consulte la Guía de implementación del programa.

PERP

Consulte planificación de recursos empresariales.

análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para

encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

F

tabla de datos

La tabla central de un esquema en forma de estrella. Almacena datos cuantitativos sobre las operaciones comerciales. Normalmente, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

fallan rápidamente

Una filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de un enfoque ágil.

límite de aislamiento de fallas

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para obtener más información, consulte Límites de AWS aislamiento de errores.

rama de característica

Consulte la sucursal.

características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con:AWS.

transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

FGAC

Consulte el control de acceso detallado.

control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.

migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la captura de datos modificados para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

G

bloqueo geográfico

Consulta las restricciones geográficas.

restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta Restringir la distribución geográfica del contenido en la CloudFront documentación.

Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, y el flujo de trabajo basado en enlaces troncales es el enfoque moderno preferido.

estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como implementación sobre infraestructura existente. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y la conformidad en todas las unidades organizativas (OU). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

H

JA

Consulte alta disponibilidad.

migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. AWS ofrece AWS SCT, lo cual ayuda con las conversiones de esquemas.

alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server). La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, las revisiones suelen realizarse fuera del flujo de trabajo habitual de las DevOps versiones.

periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

I

IaC

Vea la infraestructura como código.

políticas basadas en identidad

Política asociada a uno o más directores de IAM que define sus permisos en el Nube de AWS entorno.

aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

IIoT

Consulte Internet de las cosas industrial.

infraestructura inmutable

Un modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar, parchear o modificar la infraestructura existente. Las infraestructuras inmutables son intrínsecamente más consistentes, fiables y predecibles que las infraestructuras mutables. Para obtener más información, consulte las prácticas recomendadas para implementar con una infraestructura inmutable en Well-Architected Framework AWS .

VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La Arquitectura de referencia de seguridad de AWS recomienda configurar su cuenta de red con VPC entrantes, salientes y de inspección para proteger la interfaz bidireccional entre su aplicación e Internet en general.

migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

Industria 4.0

Un término que Klaus Schwab introdujo en 2016 para referirse a la modernización de los procesos de fabricación mediante avances en la conectividad, los datos en tiempo real, la automatización, el análisis y la inteligencia artificial/aprendizaje automático.

infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

Internet de las cosas industrial (IIoT)

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la agricultura. Para obtener más información, consulte Creación de una estrategia de transformación digital del Internet de las cosas industrial (IIoT).

VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red entre las VPC (iguales o Regiones de AWS diferentes), Internet y las redes locales. La Arquitectura de referencia de seguridad de AWS recomienda configurar su cuenta de red con VPC entrantes, salientes y de inspección para proteger la interfaz bidireccional entre su aplicación e Internet en general.

Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte ¿Qué es IoT?.

interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para más información, consulte Interpretabilidad del modelo de machine learning con AWS.

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la Guía de integración de operaciones.

ITIL

Consulte la biblioteca de información de TI.

ITSM

Consulte Administración de servicios de TI.

L

control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte Configuración de un entorno de AWS seguro y escalable con varias cuentas.

migración grande

Migración de 300 servidores o más.

LBAC

Consulte el control de acceso basado en etiquetas.

privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte Aplicar permisos de privilegio mínimo en la documentación de IAM.

migrar mediante lift-and-shift

Ver 7 Rs.

sistema little-endian

Un sistema que almacena primero el byte menos significativo. Véase también endianness.

entornos inferiores

Véase entorno.

M

machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte Machine learning.

rama principal

Ver sucursal.

malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware puede interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los registradores de pulsaciones de teclas.

servicios gestionados

Servicios de AWS para los que AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y usted accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios gestionados. También se conocen como servicios abstractos.

sistema de ejecución de fabricación (MES)

Un sistema de software para rastrear, monitorear, documentar y controlar los procesos de producción que convierten las materias primas en productos terminados en el taller.

MAP

Consulte Migration Acceleration Program.

mecanismo

Un proceso completo en el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para realizar ajustes. Un mecanismo es un ciclo que se refuerza y mejora a sí mismo a medida que funciona. Para obtener más información, consulte Creación de mecanismos en el AWS Well-Architected Framework.

cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización. AWS Organizations Una cuenta no puede pertenecer a más de una organización a la vez.

MES

Consulte el sistema de ejecución de la fabricación.

Transporte telemétrico de Message Queue Queue (MQTT)

Un protocolo de comunicación ligero machine-to-machine (M2M), basado en el patrón de publicación/suscripción, para dispositivos de IoT con recursos limitados.

microservicio

Un servicio pequeño e independiente que se comunica a través de API bien definidas y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo, un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte Integrar microservicios mediante servicios sin servidor. AWS

arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de

una interfaz bien definida mediante API ligeras. Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte Implementación de microservicios en. AWS

Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la estrategia de migración de AWS.

fábrica de migración

Equipos multifuncionales que agilizan la migración de las cargas de trabajo mediante enfoques automatizados y ágiles. Los equipos de las fábricas de migración suelen incluir a analistas y propietarios de operaciones, empresas, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la discusión sobre las fábricas de migración y la Guía de fábricas de migración a la nube en este contenido.

metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: rehospede la migración a Amazon EC2 AWS con Application Migration Service.

Migration Portfolio Assessment (MPA)

Una herramienta en línea que proporciona información para validar el modelo de negocio para migrar a. Nube de AWS La MPA ofrece una evaluación detallada de la cartera (adecuación del tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La herramienta MPA (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores asociados de APN.

Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la Guía de preparación para la migración. La MRA es la primera fase de la estrategia de migración de AWS.

estrategia de migración

El enfoque utilizado para migrar una carga de trabajo a. Nube de AWS Para obtener más información, consulte la entrada de las 7 R de este glosario y consulte Movilice a su organización para acelerar las migraciones a gran escala.

ML

Consulte el aprendizaje automático.

modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia y aprovechar las innovaciones. Para obtener más información, consulte Estrategia para modernizar las aplicaciones en el Nube de AWS.

evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para obtener más información, consulte Evaluación de la preparación para la modernización de las aplicaciones en el Nube de AWS.

aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte Descomposición de monolitos en microservicios.

MAPA

Consulte la evaluación de la cartera de migración.

MQTT

Consulte Message Queue Queue Telemetría y Transporte.

clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar "¿Este producto es un libro, un automóvil o un teléfono?" o "¿Qué categoría de productos es más interesante para este cliente?".

infraestructura mutable

Un modelo que actualiza y modifica la infraestructura existente para las cargas de trabajo de producción. Para mejorar la coherencia, la fiabilidad y la previsibilidad, el AWS Well-Architected Framework recomienda el uso de una infraestructura inmutable como práctica recomendada.

O

OAC

Consulte el control de acceso de origen.

OAI

Consulte la identidad de acceso de origen.

OCM

Consulte gestión del cambio organizacional.

migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

OI

Consulte integración de operaciones.

OLA

Véase el acuerdo a nivel operativo.

migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

OPC-UA

Consulte Open Process Communications: arquitectura unificada.

Comunicaciones de proceso abierto: arquitectura unificada (OPC-UA)

Un protocolo de comunicación machine-to-machine (M2M) para la automatización industrial. El OPC-UA proporciona un estándar de interoperabilidad con esquemas de cifrado, autenticación y autorización de datos.

acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

revisión de la preparación operativa (ORR)

Una lista de preguntas y las mejores prácticas asociadas que le ayudan a comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles fallos. Para obtener más información, consulte Operational Readiness Reviews (ORR) en AWS Well-Architected Framework.

tecnología operativa (OT)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En la industria manufacturera, la

integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de la industria 4.0.

integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la Guía de integración de las operaciones.

registro de seguimiento organizativo

Un registro creado por el AWS CloudTrail que se registran todos los eventos para todos Cuentas de AWS los miembros de una organización AWS Organizations. Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte Crear un registro para una organización en la CloudTrail documentación.

administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la Guía de OCM.

control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor AWS KMS (SSE-KMS) y las solicitudes dinámicas `PUT` y `DELETE` dirigidas al bucket de S3.

identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el OAC, que proporciona un control de acceso más detallado y mejorado.

O

Consulte la revisión de la preparación operativa.

NO

Consulte tecnología operativa.

VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La Arquitectura de referencia de seguridad de AWS recomienda configurar su cuenta de red con VPC entrantes, salientes y de inspección para proteger la interfaz bidireccional entre su aplicación e Internet en general.

P

límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte Límites de permisos en la documentación de IAM.

información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

PII

Consulte la información de identificación personal.

manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

PLC

Consulte controlador lógico programable.

PLM

Consulte la gestión del ciclo de vida del producto.

política

Un objeto que puede definir los permisos (consulte la política basada en la identidad), especifique las condiciones de acceso (consulte la política basada en los recursos) o defina los permisos máximos para todas las cuentas de una organización AWS Organizations (consulte la política de control de servicios).

persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades. Para obtener más información, consulte Habilitación de la persistencia de datos en los microservicios.

evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la Evaluación de la preparación para la migración.

predicate

Una condición de consulta que devuelve `true` o`false`, por lo general, se encuentra en una cláusula. `WHERE`

pulsar un predicado

Técnica de optimización de consultas de bases de datos que filtra los datos de la consulta antes de transferirlos. Esto reduce la cantidad de datos que se deben recuperar y procesar de la base de datos relacional y mejora el rendimiento de las consultas.

control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte Controles preventivos en Implementación de controles de seguridad en AWS.

entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en Términos y conceptos de roles en la documentación de IAM.

Privacidad desde el diseño

Un enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de ingeniería.

zonas alojadas privadas

Contenedor que aloja información acerca de cómo desea que responda Amazon Route 53 a las consultas de DNS de un dominio y sus subdominios en una o varias VPC. Para obtener más información, consulte Uso de zonas alojadas privadas en la documentación de Route 53.

control proactivo

Un control de seguridad diseñado para evitar el despliegue de recursos no conformes. Estos controles escanean los recursos antes de aprovisionarlos. Si el recurso no cumple con el control, significa que no está aprovisionado. Para obtener más información, consulte la guía de referencia de controles en la AWS Control Tower documentación y consulte Controles proactivos en Implementación de controles de seguridad en AWS.

gestión del ciclo de vida del producto (PLM)

La gestión de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta el rechazo y la retirada.

entorno de producción

Consulte el entorno.

controlador lógico programable (PLC)

En la fabricación, una computadora adaptable y altamente confiable que monitorea las máquinas y automatiza los procesos de fabricación.

seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

publicar/suscribirse (pub/sub)

Un patrón que permite las comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un MES basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se puedan suscribir otros microservicios. El sistema puede añadir nuevos microservicios sin cambiar el servicio de publicación.

Q

plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas, restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

R

Matriz RACI

Véase responsable, responsable, consultado, informado (RACI).

ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

Matriz RASCI

Véase responsable, responsable, consultado, informado (RACI).

RCAC

Consulte control de acceso por filas y columnas.

read replica

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

rediseñar

Ver 7 Rs.

objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

refactorizar

Ver 7 Rs.

Región

Una colección de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para obtener más información, consulte Regiones de AWS Especificar qué cuenta puede usar.

regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de "¿A qué precio se venderá esta casa?", un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

volver a alojar

Consulte 7 Rs.

versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción.

trasladarse

Ver 7 Rs.

redefinir la plataforma

Ver 7 Rs.

recompra

Ver 7 Rs.

resilencia

La capacidad de una aplicación para resistir las interrupciones o recuperarse de ellas. La alta disponibilidad y la recuperación ante desastres son consideraciones comunes a la hora de planificar la resiliencia en el. Nube de AWS Para obtener más información, consulte Nube de AWS Resiliencia.

política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte Controles receptivos en Implementación de controles de seguridad en AWS.

retain

Consulte 7 Rs.

jubilarse

Ver 7 Rs.

rotación

Proceso de actualizar periódicamente un secreto para dificultar el acceso de un atacante a las credenciales.

control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

RPO

Consulte el objetivo del punto de recuperación.

RTO

Consulte el objetivo de tiempo de recuperación.

manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

S

SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión AWS Management Console o llamar a las operaciones de la AWS API sin tener que crear un usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte Acerca de la federación basada en SAML 2.0 en la documentación de IAM.

SCADA

Consulte el control de supervisión y la adquisición de datos.

SCP

Consulte la política de control de servicios.

secreta

Información confidencial o restringida, como una contraseña o credenciales de usuario, que almacene de forma cifrada. AWS Secrets Manager Se compone del valor secreto y sus

metadatos. El valor secreto puede ser binario, una sola cadena o varias cadenas. Para obtener más información, consulta ¿Qué hay en un secreto de Secrets Manager? en la documentación de Secrets Manager.

control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Hay cuatro tipos principales de controles de seguridad: preventivos, de detección, de respuesta y proactivos.

refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

automatización de la respuesta de seguridad

Una acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o remediarlo. Estas automatizaciones sirven como controles de seguridad detectables o adaptables que le ayudan a implementar las mejores prácticas AWS de seguridad. Algunos ejemplos de acciones de respuesta automatizadas incluyen la modificación de un grupo de seguridad de VPC, la aplicación de parches a una instancia de Amazon EC2 o la rotación de credenciales.

cifrado del servidor

Cifrado de los datos en su destino, por parte de quien Servicio de AWS los recibe.

política de control de servicio (SCP)

Una política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. Las SCP definen barreras de protección o establecen límites a las acciones que un administrador puede delegar en los usuarios o roles. Puede utilizar las SCP como listas de permitidos o rechazados, para especificar qué servicios o acciones se

encuentra permitidos o prohibidos. Para obtener más información, consulte las políticas de control de servicios en la AWS Organizations documentación.

punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte Puntos de conexión de Servicio de AWS en Referencia general de AWS.

acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

objetivo de nivel de servicio (SLO)

Una métrica objetivo que representa el estado de un servicio, medido mediante un indicador de nivel de servicio.

modelo de responsabilidad compartida

Un modelo que describe la responsabilidad que compartes con respecto a la seguridad y AWS el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el Modelo de responsabilidad compartida.

SIEM

Consulte la información de seguridad y el sistema de gestión de eventos.

punto único de fallo (SPOF)

Una falla en un único componente crítico de una aplicación que puede interrumpir el sistema.

SLA

Consulte el acuerdo de nivel de servicio.

SLI

Consulte el indicador de nivel de servicio.

ASÍ QUE

Consulte el objetivo de nivel de servicio.

split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para obtener más información, consulte Enfoque gradual para modernizar las aplicaciones en el. Nube de AWS

SPOT

Consulte el punto único de falla.

esquema en forma de estrella

Estructura organizativa de una base de datos que utiliza una tabla de datos grande para almacenar datos transaccionales o medidos y una o más tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para usarse en un almacén de datos o con fines de inteligencia empresarial.

patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda desmantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue presentado por Martin Fowler como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo con la aplicación de este patrón, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

supervisión, control y adquisición de datos (SCADA)

En la industria manufacturera, un sistema que utiliza hardware y software para monitorear los activos físicos y las operaciones de producción.

cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

pruebas sintéticas

Probar un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o monitorear el rendimiento. Puede usar Amazon CloudWatch Synthetics para crear estas pruebas.

T

etiquetas

Pares clave-valor que actúan como metadatos para organizar los recursos. AWS Las etiquetas pueden ayudarle a administrar, identificar, organizar, buscar y filtrar recursos. Para obtener más información, consulte Etiquetado de los recursos de AWS.

variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

entorno de prueba

Consulte entorno.

entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

puerta de enlace de tránsito

Centro de tránsito de red que puede utilizar para interconectar las VPC y las redes en las instalaciones. Para obtener más información, consulte Qué es una pasarela de tránsito en la AWS Transit Gateway documentación.

flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración por usted. Para obtener más información, consulte AWS Organizations Utilización con otros AWS servicios en la AWS Organizations documentación.

ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

U

incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos. Para más información, consulte la guía Cuantificación de la incertidumbre en los sistemas de aprendizaje profundo.

tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

entornos superiores

Ver entorno.

V

succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

Emparejamiento de VPC

Conexión entre dos VPC que permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte ¿Qué es una interconexión de VPC? en la documentación de Amazon VPC.

vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

W

caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

función de ventana

Función SQL que realiza un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para procesar tareas, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

GUSANO

Mira, escribe una vez, lee muchas.

WQF

Consulte el marco de calificación de cargas de trabajo de AWS.

escribe una vez, lee muchas (WORM)

Un modelo de almacenamiento que escribe los datos una sola vez y evita que los datos se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no pueden cambiarlos. Esta infraestructura de almacenamiento de datos se considera inmutable.

Z

ataque de día cero

Un ataque, normalmente de malware, que aprovecha una vulnerabilidad de día cero.

vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.