



Aplicación del AWS Well-Architected marco de Amazon Timestream para InfluxDB

AWS Orientación prescriptiva



AWS Orientación prescriptiva: Aplicación del AWS Well-Architected marco de Amazon Timestream para InfluxDB

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Introducción	1
Destinatarios previstos	2
Objetivos	2
Excelencia operativa	3
Automatice la implementación mediante un enfoque de IaC	3
Cambios frecuentes, pequeños y reversibles	4
Anticipación de los errores	4
Lecciones de los errores operativos	5
Uso de características de registro para supervisar la actividad no autorizada o anómala	5
Seguridad	7
Implementación la seguridad de los datos	7
Protección de las redes	8
Implementación de la autenticación y autorización	8
Fiabilidad	10
Arquitectura de la carga de trabajo, lo que abarca las cuotas de servicio y los patrones de implementación	10
Patrones de implementación	10
Administre y escale Timestream para InfluxDB	11
Eficiencia del rendimiento	13
Modelado de datos de afluencia y optimización de consultas	13
Optimiza las escrituras	15
Optimización de costos	17
Comprenda los requisitos y los costos de su caso de uso	17
Seleccionar los recursos prestando atención al costo	17
Escalado para satisfacer las necesidades empresariales sin gastar de más	18
Ajustar el tamaño del almacenamiento y la transferencia de datos	19
Sostenibilidad	20
Región de AWS selección	20
Base el consumo de recursos en los patrones de comportamiento de los usuarios	21
Optimización de los patrones de arquitectura y desarrollo de software	21
Recursos	23
Referencias	23
Publicaciones de blog	23
Historial de documentos	24

..... XXV

Aplicación del marco AWS Well-Architected para Amazon Timestream para InfluxDB

Balwanth Bobilli, Amazon Web Services

Enero de 2025 ([historial del documento](#))

Puede crear time-series-based soluciones en Amazon Web Services (AWS) mediante [Amazon Timestream](#). Amazon Timestream ofrece motores de bases de datos de series temporales totalmente gestionados y diseñados específicamente para cargas de trabajo, desde consultas de baja latencia hasta ingestión de datos a gran escala. Con [Amazon Timestream para InfluxDB](#), puede ejecutar bases de datos de InfluxDB de código abierto para aplicaciones de series temporales, como alertas AWS en tiempo real y monitoreo de la confiabilidad de la infraestructura, con tiempos de respuesta de milisegundos. Timestream para InfluxDB ofrece una disponibilidad de hasta el 99,9 por ciento.

Esta guía proporciona una guía prescriptiva para aplicar los principios del [AWS Well-Architected Framework](#) al planificar su implementación de Timestream para InfluxDB. El AWS Well-Architected Framework le ayuda a crear infraestructuras seguras, de alto rendimiento, resilientes y eficientes para una variedad de aplicaciones y cargas de trabajo. También proporciona una estrategia coherente para evaluar las arquitecturas e implementar diseños escalables.

El AWS Well-Architected Framework se basa en los seis pilares siguientes:

- Excelencia operativa
- Seguridad
- Fiabilidad
- Eficiencia del rendimiento
- Optimización de costos
- Sostenibilidad

Esta guía proporciona información sobre los pilares de diseño de AWS Well-Architected Framework. Considere la posibilidad de utilizar estas prácticas recomendadas cuando implemente Amazon Timestream for InfluxDB en AWS.

 Note

Algunos Servicios de AWS no están disponibles en todos. Regiones de AWS Para conocer la disponibilidad regional, consulta la página [de límites y cuotas del servicio](#) en la AWS documentación y elige el enlace correspondiente al servicio.

Destinatarios previstos

Esta guía está destinada a ingenieros de datos, arquitectos de soluciones y analistas de datos que diseñan e implementan soluciones para datos de series temporales. AWS

Objetivos

Esta guía puede serle útil a usted y a su organización para hacer lo siguiente:

- Elija entre las opciones de implementación compatibles, realice escrituras optimizadas e implemente un acceso detallado.
- Siga los patrones de diseño de AWS Well-Architected que ayudan a mejorar la resiliencia y la seguridad.
- Diseñe sus consultas para obtener un rendimiento óptimo.
- Aprenda a ser eficiente desde el punto de vista operativo al administrar su instancia de Timestream for InfluxDB en producción.

Pilar de excelencia operativa

El pilar de [excelencia operativa](#) del AWS Well-Architected Framework se centra en el funcionamiento y la supervisión de los sistemas, y en la mejora continua de los procesos y procedimientos para ofrecer valor empresarial. El pilar de la excelencia operativa incluye la capacidad de respaldar el desarrollo y ejecutar las cargas de trabajo de manera eficaz, así como de obtener información sobre su funcionamiento.

Puede reducir la complejidad operativa mediante cargas de trabajo con capacidad de recuperación automática, que detectan y solucionan la mayoría de los problemas sin intervención humana. Para lograr este objetivo, siga las prácticas recomendadas que se describen en esta sección. Utilice CloudWatch las métricas de [Amazon](#) para Amazon Timestream para InfluxDB, el punto final de métricas nativo de InfluxDB, y los mecanismos para responder cuando su carga de APIs trabajo se desvíe del comportamiento esperado.

Este análisis del pilar de excelencia operativa se centra en las siguientes áreas clave:

- Infraestructura como código (IaC)
- Administración de cambios
- Estrategias de resiliencia
- Administración de incidentes
- Registro y supervisión con fines de auditoría

Automatice la implementación mediante un enfoque de IaC

Las mejores prácticas para automatizar la implementación en Timestream para InfluxDB mediante IaC incluyen las siguientes:

- Aplique iAC para implementar Timestream para InfluxDB siempre que sea posible. Para una configuración del entorno coherente, utilice una [AWS CloudFormation](#) plantilla o [HashiCorp Terraform para crear todos los recursos](#) necesarios para su instancia. [Kit de desarrollo de la nube de AWS \(AWS CDK\)](#)
- Automatice la transmisión temporal de los procedimientos operativos de InfluxDB, como el cambio de tamaño de las instancias.

- Utilice etiquetas para añadir metadatos a su flujo temporal de recursos de InfluxDB y realice un seguimiento del uso en función de las etiquetas. Para obtener más información, consulte [Etiquetado de Amazon Timestream](#) para InfluxDB.

Cambios frecuentes, pequeños y reversibles

Las siguientes recomendaciones se centran en cambios pequeños y reversibles para minimizar la complejidad y reducir la probabilidad de que se interrumpa la carga de trabajo:

- Guarde las plantillas y scripts de IaC en un servicio de control de código fuente, como o. GitHub GitLab No almacene las AWS credenciales en el control de código fuente.
- Exija que las implementaciones de IaC utilicen un servicio de integración y entrega continuas (CI/CD), como [AWS CodeDeploy](#) o [AWS CodeBuild](#). Estos servicios compilan, prueban e implementan código en un entorno no de producción que contiene una instancia efímera de InfluxDB antes de afectar a la instancia de InfluxDB de producción.
- Pruebe las consultas de infraestructura y aplicaciones en un entorno inferior antes de implementarlas en producción. Esto minimiza la probabilidad de que se produzca una interrupción y ayuda a garantizar que funcionen bien con su carga de trabajo y su escalabilidad.

Anticipación de los errores

Una infraestructura que se recupere automáticamente ejemplifica la excelencia operativa, pues anticipa los errores y trata de resolver cualquier problema sin intervención. Las siguientes recomendaciones le ayudarán a alcanzar esa madurez con Timestream para InfluxDB:

- Utilice métricas para supervisar la memoria, la CPU y el uso de almacenamiento. Puede configurar CloudWatch para notificar cuándo cambian los patrones de uso o cuándo se está llegando al límite de capacidad de la implementación. De esta forma, puede mantener el rendimiento y la disponibilidad del sistema.
- Amplíe su instancia de base de datos cuando se acerque al límite de recursos. Debe tener búfer de almacenamiento y de memoria para asumir incrementos imprevistos de la demanda de las aplicaciones.
- Si la carga de trabajo de su base de datos requiere I/O más de lo que ha aprovisionado, la recuperación tras una conmutación por error o un fallo en la base de datos será lenta. Para

umentar la capacidad. I/O capacity of a DB instance, migrate to a different DB instance that has higher I/O

- Si la aplicación cliente almacena en caché los datos de DNS de las instancias de base de datos, establezca un valor time-to-live (TTL) inferior a 30 segundos. La dirección IP subyacente de una instancia de base de datos puede cambiar después de producirse una conmutación por error. El almacenamiento en caché de los datos del DNS durante un período prolongado puede provocar fallos de conexión. Es posible que tu aplicación intente conectarse a una dirección IP que ya no esté en servicio.
- Si su aplicación necesita sobrevivir a una Región de AWS interrupción total, considere configurar la replicación o escribirla a otra región como parte de sus planes de recuperación ante desastres (DR). Comprenda las limitaciones a la hora de configurar la replicación. Para obtener más información sobre la replicación, consulte la documentación de [InfluxDB](#).

Lecciones de los errores operativos

Una infraestructura autorreparable es un esfuerzo a largo plazo que se desarrolla de forma iterativa cuando se producen problemas poco frecuentes o las respuestas no son tan eficaces como se desearía. Para centrarse en lograr una infraestructura que se recupere automáticamente, adopte las siguientes prácticas:

- Aprenda de los errores para impulsar la mejora.
- Comparta las conclusiones con los equipos y la organización. Si varios equipos de una organización utilizan Timestream para InfluxDB, cree una sala de chat o un grupo de usuarios común para compartir las lecciones aprendidas y las mejores prácticas.

Uso de características de registro para supervisar la actividad no autorizada o anómala

Para observar patrones anómalos de rendimiento y actividad, tenga en cuenta las siguientes prácticas:

- Habilite la [entrega de registros](#) para almacenar los registros de InfluxDB en [Amazon Simple Storage Service \(Amazon S3\)](#). Los registros de InfluxDB registran información que puede ayudar a comprobar lo siguiente:
 - [Eventos de la API del plano de datos](#)

- Tiempos de respuesta
- Detalles de compactación
- Cualquier error o advertencia crítica detectada por el sistema

Revise los registros para ver si hay anomalías o accesos no autorizados. En general, el registro proporciona información de diagnóstico para la solución de problemas.

- Timestream for InfluxDB admite el registro de las acciones del plano de control mediante el uso de AWS CloudTrail Para obtener más información, consulte [Registrar la transmisión temporal de las llamadas a la API de InfluxDB con. AWS CloudTrail](#)
- Puede supervisar **CPUUtilization** y medir las **DiskUtilization** métricas desde **MemoryUtilization** Timestream/InfluxDB > < Namespace > en. CloudWatch

[Para obtener más información, consulte la documentación de Timestream for InfluxDB.](#)

Pilar de seguridad

El [pilar de seguridad](#) le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida al utilizar Timestream para InfluxDB.

El pilar de seguridad incluye las siguientes áreas clave:

- Seguridad de los datos
- Seguridad de la red
- Autenticación y autorización

En las siguientes secciones, se muestra cómo configurar Timestream para InfluxDB a fin de cumplir sus objetivos de seguridad y conformidad. También aprenderá a usar otros Servicios de AWS que le ayuden a monitorear y proteger sus recursos de Timestream for InfluxDB.

Implementación la seguridad de los datos

Las brechas de seguridad y las filtraciones de datos ponen en riesgo a sus clientes y pueden tener un impacto negativo sustancial en su empresa. El [modelo de responsabilidad AWS compartida se aplica](#) a la protección de datos en Timestream for InfluxDB. Las siguientes prácticas ayudan a proteger los datos de sus clientes de una exposición inadvertida o malintencionada:

- No incluyas información confidencial en los nombres de las instancias, las etiquetas, los grupos de parámetros, las funciones AWS Identity and Access Management (IAM) y otros metadatos. Es posible que esos datos aparezcan en los registros de facturación o diagnóstico.
- Utilice el cifrado para aumentar la protección de los datos de las aplicaciones que se despliegan en la nube. Las instancias cifradas proporcionan una capa adicional de protección de datos, ya que ayudan a proteger sus datos contra el acceso no autorizado al almacenamiento subyacente. El flujo temporal para el cifrado de InfluxDB está activado de forma predeterminada.
- Utilice la parametrización siempre que sea posible. APIs

Para obtener más información sobre la protección y el cifrado de datos, consulte la documentación de [Timestream](#) for InfluxDB.

Protección de las redes

Puede crear una instancia de Timestream for InfluxDB solo en una nube privada virtual (VPC) activada. AWS Para proteger aún más la instancia, haga lo siguiente:

- Restrinja el acceso público al punto final de la instancia mediante una de las siguientes acciones:
 - Seleccione No accesible públicamente en la página Crear base de datos de InfluxDB (seleccionada de forma predeterminada).
 - Establezca la [PubliclyAccessible](#) propiedad false en AWS CloudFormation (`AWS::Timestream::InfluxDBInstance`).

Si eliges No accesible públicamente o configuras esta opción false, solo se puede acceder PubliclyAccessible a los puntos finales de la instancia dentro de la VPC. Por lo general, se accede a los puntos de enlace desde una instancia de [Amazon Elastic Compute Cloud \(Amazon EC2\) que se](#) ejecuta en la misma VPC que la instancia de Timestream for InfluxDB.

- Utilice grupos de seguridad para proteger aún más el acceso de su red a Timestream for InfluxDB dentro de la VPC.
- Úselo para comunicarse SSL/TLS con los recursos. AWS Timestream para InfluxDB requiere TLS 1.2 y recomendamos TLS 1.3.
- Conéctese de forma segura a su instancia siguiendo las instrucciones de [Cómo crear una instancia de Timestream for InfluxDB y conectarse a ella](#).
- [Establezca una conexión privada entre su VPC y los puntos de enlace de la API del plano de control de Amazon Timestream for InfluxDB mediante la creación de un punto de enlace de VPC de interfaz.](#)

[Para obtener más información sobre la seguridad, consulte la documentación de Timestream for InfluxDB.](#)

Implementación de la autenticación y autorización

Utilice las credenciales de IAM para controlar las acciones de administración en Timestream para las instancias de InfluxDB. Cuando se conecta a Timestream para InfluxDB mediante credenciales de IAM, su función de IAM debe tener políticas de IAM que concedan los permisos necesarios para realizar las operaciones de administración de Timestream for InfluxDB. Asegúrese de seguir el principio de privilegios mínimos y conceder solo los permisos necesarios para completar una tarea.

Para obtener más información, consulte [Identity and Access Management for Amazon Timestream for InfluxDB](#).

Puede usar [Amazon Timestream para las acciones de InfluxDB en la política de IAM para](#) controlar quién puede administrar su instancia de Timestream for InfluxDB.

[Para proporcionar un control de acceso detallado a los datos almacenados en su instancia de Amazon Timestream for InfluxDB, proporcione a los usuarios los tokens de API de InfluxDB.](#)

Para interactuar con otros Servicios de AWS, Amazon Timestream para InfluxDB utiliza funciones vinculadas al servicio de IAM. Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Timestream para InfluxDB. Timestream para InfluxDB predefine los roles vinculados al servicio e incluyen todos los permisos que el servicio necesita para llamar a otros en su nombre. Servicios de AWS Para obtener más información, consulte [Uso de funciones vinculadas a servicios para Amazon Timestream for InfluxDB](#).

Pilar de fiabilidad

El [pilar de la confiabilidad](#) abarca la capacidad de una carga de trabajo para realizar la función prevista de manera correcta y coherente cuando se espera que lo haga. Esto incluye la capacidad de operar y probar la carga de trabajo durante todo su ciclo de vida.

La configuración de una carga de trabajo confiable comienza con decisiones de diseño iniciales tanto para el software como para la infraestructura. Sus elecciones respecto a la arquitectura afectarán al comportamiento de su carga de trabajo en todos los pilares de Well-Architected. Para lograr la confiabilidad, debe seguir patrones específicos.

El pilar de fiabilidad se centra en las siguientes áreas clave:

- Arquitectura de la carga de trabajo, lo que abarca las cuotas de servicio y los patrones de implementación
- Administrar y escalar instancias de InfluxDB

Arquitectura de la carga de trabajo, lo que abarca las cuotas de servicio y los patrones de implementación

Cada una Cuenta de AWS tiene cuotas para los recursos que se ofrecen en cada una. Región de AWS Por ejemplo, cada región tiene una [cuota de Timestream para las instancias de InfluxDB](#), independientemente del tamaño de la instancia. Cuando se alcanza el número máximo de instancias en una región, las llamadas adicionales para crear instancias fallan, salvo una excepción. El volumen de almacenamiento de una instancia de Timestream para InfluxDB puede crecer hasta un tamaño máximo de 16 terabytes (Tb) en total. TiBs Regiones de AWS

Patrones de implementación

Para obtener [una alta disponibilidad](#) y compatibilidad con la conmutación por error de las instancias de Timestream for InfluxDB, puede utilizar las implementaciones Multi-AZ con una única instancia de base de datos en espera. Este tipo de implementación se denomina Implementación de instancia de base de datos Multi-AZ. Amazon Timestream para InfluxDB utiliza la tecnología de conmutación por error de Amazon. En una implementación de instancia de base de datos Multi-AZ, Amazon Timestream aprovisiona y mantiene automáticamente una réplica síncrona en espera dentro de una

zona de disponibilidad diferente. Para proporcionar redundancia de datos, la instancia de base de datos principal se replica de forma sincrónica en todas las zonas de disponibilidad hasta la réplica en espera.

La ejecución de una instancia de base de datos con alta disponibilidad puede proporcionar disponibilidad durante un fallo de la instancia de base de datos o una interrupción en la zona de disponibilidad. Si una interrupción imprevista de la instancia de base de datos se debe a un defecto en la infraestructura, Amazon Timestream for InfluxDB cambia automáticamente a la réplica en espera. El tiempo requerido para completar la conmutación por error dependerá de la actividad de la base de datos y de otras condiciones existentes en el momento en que la instancia de base de datos principal deja de estar disponible.

Los tiempos de conmutación por error suelen estar entre 60–120 segundos. Sin embargo, las transacciones grandes con datos de alta cardinalidad o un proceso de recuperación prolongado con requisitos previos al calentamiento pueden aumentar el tiempo de conmutación por error. Una vez finalizada la conmutación por error, es posible que se necesite más tiempo antes de que la consola de Timestream refleje la nueva zona de disponibilidad.

Si su aplicación debe permanecer disponible durante una Región de AWS interrupción total, considere configurar la replicación o enviarla a otra región como parte de sus planes de recuperación ante desastres (DR). Sin embargo, antes de configurar la replicación, asegúrese de entender las limitaciones. Para obtener más información, consulte la documentación de [InfluxDB](#).

Amazon Timestream para InfluxDB realiza copias de seguridad internas de forma periódica y las retiene durante 24 horas para garantizar la disponibilidad y la durabilidad. Las instantáneas se toman durante las eliminaciones y se retienen durante 30 días para poder realizar restauraciones. Para acceder a ellas o utilizarlas, cree un caso en. [AWS Support](#)

Administre y escale Timestream para InfluxDB

Timestream for InfluxDB admite clases de instancias que son ideales para ejecutar cargas de trabajo que consumen mucha memoria en bases de datos de InfluxDB de código abierto. Las distintas [clases de instancias de db.flux](#) tienen límites en cuanto a v, memoria, almacenamiento y ancho de banda de red. CPUs Para elegir la clase de instancia que se ajuste a los requisitos de latencia de escritura y consulta de tu aplicación CloudWatch CPUUtilizationMemoryUtilization, observa Amazon y DiskUtilization las métricas durante las pruebas. Puedes [escalar tus instancias hacia arriba o hacia abajo](#) en función de tus requisitos de carga de trabajo. Timestream for InfluxDB proporciona varios niveles de almacenamiento preconfigurados con las IOPS óptimas y el

rendimiento necesarios para los diferentes tipos de cargas de trabajo. Elija lo que mejor se adapte a su carga de trabajo en función de sus requisitos.

Si sus necesidades de escalado cambian en momentos predecibles, puede usar una [AWS Lambda función](#) o un programador personalizado y ejecutar una API o un SDK para escalar hacia arriba y hacia abajo con un tiempo de búfer.

La configuración de InfluxDB se administra en Timestream para InfluxDB mediante los parámetros de un grupo de parámetros. Los grupos de parámetros actúan como contenedores de las opciones de configuración de InfluxDB que se aplican a una o más instancias de base de datos. Al modificar los parámetros de los grupos de parámetros, comprenda la diferencia entre los parámetros estáticos y dinámicos y cómo y cuándo se aplican. Para ver la configuración aplicada actualmente, utilice la acción de la [GetDbParameterGroup](#) API.

Pilar de eficiencia de rendimiento

El [pilar de eficiencia del rendimiento](#) del AWS Well-Architected Framework se centra en cómo optimizar el rendimiento al ingerir o consultar datos. La optimización del rendimiento es un proceso gradual y continuo que consiste en lo siguiente:

- Confirmación de los requisitos empresariales
- Medir el rendimiento de la carga de trabajo
- Identificar los componentes de bajo rendimiento
- Ajustar los componentes para que se adapten a las necesidades de su empresa

El pilar de la eficiencia del rendimiento proporciona pautas que pueden ayudarlo a elegir un modelo de datos de alto rendimiento. El pilar de la eficiencia del rendimiento incluye las mejores prácticas de optimización de consulta y escritura.

El pilar de la eficiencia del rendimiento se centra en las siguientes áreas clave:

- Modelado de datos de afluencia y optimización de consultas
- Optimización de escritura

Modelado de datos de afluencia y optimización de consultas

Diseñar un esquema efectivo es crucial para optimizar el rendimiento y las capacidades de consulta de los datos de series temporales en InfluxDB. Comience por elegir las etiquetas y los campos correctos. InfluxDB indexa las etiquetas, por lo que el motor de consultas no necesita escanear todos los registros de una medición para localizar el valor de una etiqueta. Esto significa que consultar etiquetas es más eficiente que consultar campos. Para compactar y almacenar los datos, el motor de almacenamiento agrupa los valores de los campos por clave de serie y, a continuación, los ordena por tiempo. Una clave de serie se define mediante la medición, la clave y el valor de la etiqueta y la clave de campo. Para obtener más información sobre el diseño de datos, consulte la documentación de [InfluxDB](#).

El motor de almacenamiento utiliza un formato de datos de árbol de fusión estructurado en el tiempo (TSM). [Para obtener más información sobre el formato de datos TSM, consulte la documentación de InfluxDB.](#)

Imagine que está recopilando datos (`timestamp`, `host_id`, `region`, `cpu`, `memory`, `network_in_bytes`, `network_out_bytes`, `disk_io`) como parte de un DevOps caso de uso. Las etiquetas, incluida la marca de tiempo del registro, proporcionan un contexto que ayuda a identificar el quién, el qué, el cuándo y el dónde de un registro. Las etiquetas se utilizan para organizar y clasificar los datos y para filtrar los datos como parte de una consulta.

Las `region` etiquetas `host_id` y son etiquetas ideales para organizar y categorizar el caso de DevOps uso. Estas columnas ayudan a filtrar los datos de un host en particular o a ejecutar un análisis basado en la columna de la región.

Las medidas proporcionan la base para realizar cálculos matemáticos (como calcular los totales, los promedios y las diferencias en la tasa de cambio) y el análisis cuantitativo de los datos. Por lo tanto `cpu`, `memory`, `network_in_bytes`, `network_out_bytes`, y `disk_io` capturan métricas importantes relacionadas con las DevOps que cambian con el tiempo. Puede utilizar estas métricas para realizar varios análisis, como calcular la CPU y la memoria en distintos hosts. Puede usar estos valores métricos para tomar decisiones basadas en datos que le ayuden a evitar interrupciones en la producción y a planificar la infraestructura.

La cardinalidad es la combinación de valores de etiqueta únicos. Intente mantener la cardinalidad lo más baja posible. Si su aplicación requiere un identificador único para cada punto de datos, utilice valores de campo en lugar de valores de etiqueta. Esto se traducirá en una latencia de consulta significativamente mejor. Un buen diseño del esquema puede evitar una alta cardinalidad de las series, lo que se traduce en un mejor rendimiento de las consultas. Si nota que las lecturas y escrituras de datos se ralentizan o si quiere saber cómo afecta la cardinalidad al rendimiento, consulte la documentación de [Timestream](#) for InfluxDB.

Si tu aplicación emite objetos JSON, conviértelos en columnas individuales (etiquetas o campos) y carga las columnas en InfluxDB. InfluxDB está diseñado para datos de series temporales, por lo que organizar los datos con columnas individuales es una buena práctica para aprovechar al máximo las capacidades del servicio.

Una sola instancia de OSS de InfluxDB v2.7 admite aproximadamente 20 depósitos de InfluxDB que se escriben o consultan activamente en todas las organizaciones. Más de 20 cubos pueden afectar negativamente al rendimiento. Hay límites en algunas opciones de configuración de InfluxDB y hay algunas opciones que puede configurar en función de su caso de uso. Valide la configuración en función de la carga de trabajo de la aplicación durante la fase de prueba. Las retenciones de datos se configuran a nivel de bucket, por lo que los datos con diferentes requisitos de retención de

datos deben almacenarse en diferentes cubos. Para obtener más información sobre las opciones de configuración, consulte la documentación de [Timestream](#) for InfluxDB.

Guarde los datos en valores de etiqueta o valores de campo, no en claves de etiqueta, claves de campo o medidas. Si diseñas tu esquema para almacenar datos en valores de etiquetas y campos, tus consultas serán más fáciles de escribir y más eficientes. Para obtener más información sobre las mejores prácticas sobre el modelado de datos, consulte [Diseñar para mejorar el rendimiento](#).

Utilice [las tareas de InfluxDB](#) para agregar previamente los datos, cargarlos en diferentes medidas o grupos y generar datos para los paneles y las visualizaciones a partir de ellos.

InfluxDB OSS expone un `/metrics` [punto final](#) que devuelve métricas de rendimiento, recursos y uso formateadas en el formato de exposición de texto plano de Prometheus. Utilice las plantillas de InfluxDB para configurar la [supervisión y las alertas](#) a fin de detectar problemas de forma proactiva, como la alta latencia de las consultas, la degradación del rendimiento de escritura o los picos de uso de los recursos.

Timestream para InfluxDB proporciona almacenamiento incluido en Influx IO. La selección del tamaño de IOPS adecuado puede acelerar significativamente la ejecución de las consultas. Esto resulta especialmente útil para las consultas que necesitan escanear grandes cantidades de datos o gestionar una gran variedad de solicitudes. En algunas situaciones, puede ser necesaria una combinación de ampliación de la instancia y mejora de las IOPS para lograr las mejoras de rendimiento deseadas.

Recomendamos combinar los entornos de desarrollo y producción (clase de instancia, tipo de almacenamiento, configuraciones). Pruebe los cambios en el entorno inferior para cada versión antes de pasar a la fase de producción. En los volúmenes de almacenamiento incluidos en Influx IO, Timestream para InfluxDB ofrece tres niveles de almacenamiento preconfigurados con las IOPS óptimas (3000, 12 000 y 16 000) y el rendimiento necesario para diferentes tipos de cargas de trabajo. La mayoría de los casos de uso requieren menos de 3000 IOPS. Elija 12 000 o 16 000 solo si las pruebas de rendimiento indican la necesidad de altas IOPS. Para obtener más información, consulte la [sección de configuración](#) de la documentación de Timestream for InfluxDB.

Optimiza las escrituras

Para optimizar las escrituras en InfluxDB, recomendamos escribir los datos en lotes de 5000 líneas de protocolo de línea por solicitud para minimizar la sobrecarga de la red. Para obtener un mejor rendimiento, ordene las etiquetas por clave en orden lexicográfico antes de escribir los puntos de

datos. El uso de la precisión temporal más aproximada posible para las marcas de tiempo, en lugar de nanosegundos, también puede mejorar el rendimiento. Habilitar la compresión gzip es otra forma de acelerar las escrituras y reducir el ancho de banda de la red. En la configuración del plugin de `influxdb_v2` salida de tu `telegraf.conf` archivo, establece la `content_encoding` opción en `gzip`. La implementación de estas optimizaciones puede mejorar significativamente el rendimiento y la eficiencia de la escritura de datos en InfluxDB. [Para obtener más información sobre las mejores prácticas de escritura de InfluxDB, consulte Optimizar las escrituras en InfluxDB.](#)

El rendimiento de escritura de InfluxDB suele estar estrechamente relacionado con las IOPS disponibles. Al escribir datos, InfluxDB necesita realizar una cantidad significativa de operaciones de E/S para almacenar los datos. Al aumentar las IOPS, InfluxDB puede procesar más escrituras por segundo.

Pilar de optimización de costos

El [pilar de optimización de costos](#) del AWS Well-Architected Framework se centra en evitar costos innecesarios y en construir arquitecturas de una manera optimizada en cuanto a costos. Las siguientes recomendaciones pueden ayudarle a cumplir los principios de diseño de optimización de costes y las prácticas recomendadas de arquitectura de Amazon Timestream for InfluxDB.

El pilar de optimización de costos se centra en las siguientes áreas clave:

- Comprenda los requisitos y los costos de su caso de uso
- Seleccionar los recursos prestando atención al costo
- Escalado para satisfacer las necesidades empresariales sin gastar de más
- Dimensionamiento correcto del almacenamiento y la transferencia de datos

Comprenda los requisitos y los costos de su caso de uso

Recomendamos no utilizar Timestream para InfluxDB en los siguientes casos de uso:

- Si su modelo de datos tiene datos relacionales, Timestream para InfluxDB no es la solución adecuada.
- Si no puede utilizar filtros de tiempo en sus consultas, Influx escaneará todas las series, lo que resulta ineficiente.

Seleccionar los recursos prestando atención al costo

Los costos de las [instancias de InfluxDB](#) se basan en una tarifa por hora por las horas de ejecución de la instancia. Las instancias representan, en promedio, el 85 por ciento del costo total de ejecutar una base de datos AWS, por lo que el tamaño correcto puede tener importantes implicaciones de costo. La mejor manera de ajustar el tamaño de las instancias es probar el rendimiento de las aplicaciones:

- ¿Están MemoryUtilization constantemente altos o bajos? CPUUtilization
- ¿Cuál es el equilibrio entre precio y rendimiento?

Los costos de las instancias se escalan linealmente. El costo por hora de la `db.influx.2xlarge` instancia es el doble que el de la `db.influx.xlarge` instancia, aunque también tiene el doble de la asignación de recursos. La `db.influx.16xlarge` instancia es 16 veces el costo por hora de la `db.influx.xlarge` instancia.

Calcule el número de escrituras y lecturas de su carga de trabajo para un período de tiempo específico (segundos, minutos, horas o días). Las instancias de Timestream para InfluxDB admiten entre 50 000 y más de 500 000 escrituras por segundo y entre 10 y 100 consultas por segundo (QPS) según el tipo de instancia. Por ejemplo, `db.influx.2xlarge` normalmente admite hasta 150 000 escrituras por segundo y aproximadamente 25 QPS. Con un modelo de datos eficiente y consultas eficientes, puede superar ese rendimiento. Si sus requisitos varían según la hora del día, la semana o el mes, puede programar la ampliación y la reducción de la escala de la siguiente manera:

- [Cree y programe una AWS Lambda función.](#)
- Usa un programador personalizado y ejecuta una API o un SDK para [ampliarla o reducirla](#) con un tiempo de búfer.

Escalado para satisfacer las necesidades empresariales sin gastar de más

Para experimentar a nivel básico con Timestream para InfluxDB, puede usar `y. db.influx.medium` `db.influx.large`. Estas instancias son lo suficientemente grandes como para adquirir experiencia con Timestream para InfluxDB antes de invertir en instancias más grandes.

Las `db.influx.large` instancias `db.influx.medium` y son adecuadas para entornos de desarrollo de bajo coste. Sin embargo, tienen una RAM más pequeña (8 GiB y 16 GiB), menos v (CPUs 1 vCPU y 2 vCPUs) y un rendimiento de red de solo 10 GB. No todas las cargas de trabajo son adecuadas para estas clases de instancias. `MemoryUtilizationSupervise` `CPUUtilization` y amplíe o reduzca según sea necesario. A menudo, hay una relación coherente entre memoria y vCPU. La clase de instancia `db.flux` tiene una `memory-to-vCPU` proporción similar a la clase de instancia `r7g` de Amazon EC2. Recomendamos encarecidamente realizar pruebas de end-to-end rendimiento o carga antes de pasar a la fase de producción.

El modelado de datos eficiente, la escritura por lotes y las consultas optimizadas requieren menos uso de memoria y cómputo. Cuando se requieren menos recursos, puede utilizar instancias más pequeñas.

Ajustar el tamaño del almacenamiento y la transferencia de datos

Para almacenar datos, utilice las siguientes prácticas recomendadas:

- Almacene solo datos de series temporales en Timestream para InfluxDB.
- Establezca la retención adecuada en el depósito de InfluxDB para que los datos anteriores a la retención se eliminen y los fragmentos se compacten periódicamente de forma automática. [Para obtener más información, consulte la documentación de InfluxDB.](#)
- Optimice el uso del disco para futuras escrituras.
- Elimine todos los depósitos de InfluxDB que no sean necesarios para sus cargas de trabajo. InfluxDB admite eliminaciones. Puede realizar limpiezas programadas si se ajusta a su caso de uso.

Para la transferencia de datos, le recomendamos implementar la aplicación al mismo Región de AWS tiempo que su instancia de base de datos Timestream for InfluxDB para evitar sobrecargas de red entre regiones. También puede haber cargos por transferencia de datos. Para obtener más información sobre la transferencia de datos, consulta la [página de precios](#).

Pilar de sostenibilidad

El [pilar de la sostenibilidad](#) se centra en minimizar los impactos ambientales de la ejecución de cargas de trabajo en la nube. El pilar de sostenibilidad incluye las siguientes áreas clave:

- Comprender su impacto
- Objetivos de sostenibilidad
- Maximizar el uso para minimizar los recursos
- Anticipación y adopción de nuevas ofertas de hardware y software más eficaces
- Uso de servicios gestionados
- Reducir el impacto descendente

Esta guía se centra en comprender su impacto. Para obtener más información sobre los demás principios de diseño de sostenibilidad, consulte [AWS Well-Architected Framework](#).

Sus elecciones y requisitos tienen un impacto en el medio ambiente. Para aumentar la sostenibilidad de su carga de trabajo, haga lo siguiente:

- Elija Regiones de AWS los que tengan una intensidad de carbono más baja.
- Dimensione sus recursos para reflejar las necesidades reales de carga de trabajo en lugar de maximizar el tiempo de actividad y la durabilidad.
- Optimice su modelo de datos y maximice el uso de los recursos informáticos.

En las siguientes secciones, se analizan las prácticas que puede adoptar para reducir el impacto ambiental en el diseño de la carga de trabajo y en las operaciones en curso.

Región de AWS selección

Algunas Regiones de AWS están cerca de los proyectos de energía renovable de Amazon o ubicados donde la intensidad de carbono publicada por la red es inferior a la de otras redes. Evalúe las regiones en función de sus [objetivos de sostenibilidad](#) y sus requisitos de carga de trabajo. A continuación, compare su lista de regiones viables con las regiones en las que [Timestream para InfluxDB](#) esté disponible.

Base el consumo de recursos en los patrones de comportamiento de los usuarios

Ajustar el consumo al tráfico y al comportamiento de los usuarios ayuda a AWS a minimizar el impacto de los servicios en el medio ambiente. Al diseñar la solución, tenga en cuenta las siguientes prácticas recomendadas:

- Supervisa CloudWatch las métricas de Amazon, por ejemplo, `MemoryUtilization` para determinar cuándo tu demanda es más alta y más baja. `CPUUtilization` Asegúrese de que los recursos de sus instancias tengan el tamaño adecuado en esos momentos.
- Considere la posibilidad de adaptar sus acuerdos de nivel de servicio a los objetivos de sostenibilidad, además de con los objetivos de continuidad empresarial. Reducir los requisitos, como la recuperación ante desastres en varias regiones, la alta disponibilidad o la retención de copias de seguridad a largo plazo, puede reducir la cantidad de recursos necesarios para cumplir esos objetivos. Los entornos que no son de producción y las cargas de trabajo que no son esenciales para la misión ofrecen oportunidades para reducir los requisitos.

Optimización de los patrones de arquitectura y desarrollo de software

Para evitar el desperdicio, optimice el modelo de datos y las consultas. Comparta los recursos informáticos para utilizar todos los recursos disponibles en la instancia de Timestream for InfluxDB. Recomendamos implementar las siguientes prácticas recomendadas:

- Anime a los equipos de desarrolladores a compartir la pila Timestream for InfluxDB para aprovechar mejor los recursos siempre que sea posible.
- Implemente patrones que maximicen el uso de los recursos y minimicen el tiempo de inactividad. Los ejemplos de patrones incluyen el uso de subprocesos paralelos para cargar datos y agrupar registros en lotes en una transacción más grande.
- Optimice sus consultas y el modelo de datos de InfluxDB para minimizar los recursos necesarios para calcular los resultados.
- Utilice [las tareas de InfluxDB](#) para agregar previamente los datos y reducir el escaneo de los mismos datos sin procesar por parte de diferentes usuarios para visualizarlos o crear paneles.

- Mantenga actualizado su flujo temporal para los entornos de InfluxDB. Las versiones más recientes de Timestream para InfluxDB admiten las instancias EC2 más recientes, como Graviton, que son más eficientes. Las versiones más recientes de la base de datos también incluyen mejoras en la optimización de consultas y correcciones de errores que reducen la cantidad de recursos necesarios para calcular las consultas.

Recursos

Referencias

- [AWS Well-Architected](#)
- [AWS Documentación de Well-Architected Framework](#)
- [Documentación de Amazon Timestream para InfluxDB](#)
- [Documentación de InfluxDB OSS v2](#)

Publicaciones de blog

- [Utilice el script de migración de AWS InfluxDB para migrar sus datos de InfluxDB OSS 2.x a Amazon Timestream para InfluxDB](#)
- [Ejecute y gestione bases de datos de código abierto de InfluxDB con Amazon Timestream](#)

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
Publicación inicial	—	28 de enero de 2025

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.