



Prácticas recomendadas con Amazon Q Developer para la generación de código en línea y asistente

AWS Guía prescriptiva



AWS Guía prescriptiva: Prácticas recomendadas con Amazon Q Developer para la generación de código en línea y asistente

Copyright © 2025 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
Objetivos	1
Flujos de trabajo de	3
Diseño y planificación	4
Codificación	4
Revisión de código	5
Integración e implementación	5
Capacidades avanzadas	6
Transformación del código de Amazon Q Developer	6
Personalizaciones de Amazon Q Developer	6
Mejores prácticas de codificación	8
Mejores prácticas para la incorporación	8
Requisitos previos para Amazon Q Developer	8
Prácticas recomendadas al utilizar Amazon Q Developer	8
Privacidad de datos y uso del contenido en Amazon Q Developer	9
Prácticas recomendadas para la generación de código	9
Prácticas recomendadas para las recomendaciones de código	10
Ejemplos de código	12
Python ejemplos	12
Genera clases y funciones	12
Código del documento	14
Generar algoritmos	15
Genere pruebas unitarias	18
Java ejemplos	18
Genera clases y funciones	19
Código del documento	22
Genere algoritmos	24
Genere pruebas unitarias	26
Ejemplos de chat	27
Pregunte acerca de Servicios de AWS	27
Generar código	29
Genera pruebas unitarias	30
Explica el código	32
Solución de problemas	36

Generación de código vacío	36
Comentarios continuos	37
Generación de código en línea incorrecta	38
Resultados inadecuados de los chats	43
FAQs	48
¿Qué es Amazon Q Developer?	48
¿Cómo puedo acceder a Amazon Q Developer?	48
¿Qué lenguajes de programación admite Amazon Q Developer?	48
¿Cómo puedo proporcionar contexto a Amazon Q Developer para una mejor generación de código?	48
¿Qué debo hacer si la generación de código en línea con Amazon Q Developer no es precisa?	49
¿Cómo puedo utilizar la función de chat para desarrolladores de Amazon Q para la generación de código y la solución de problemas?	49
¿Cuáles son algunas de las prácticas recomendadas para utilizar Amazon Q Developer?	49
¿Puedo personalizar Amazon Q Developer para que genere recomendaciones basadas en mi propio código?	50
Pasos a seguir a continuación	51
Recursos	52
AWS blogs	52
AWS documentación	52
AWS talleres	52
Colaboradores	53
Historial de documentos	54
Glosario	55
#	55
A	56
B	59
C	61
D	64
E	69
F	71
G	73
H	74
I	76
L	78

M	79
O	84
P	87
Q	90
R	90
S	93
T	97
U	99
V	99
W	100
Z	101
.....	cii

Prácticas recomendadas con Amazon Q Developer para la generación de código en línea y asistente

Amazon Web Services ([Colaboradores](#))

Agosto de 2024 ([Historial del documento](#))

Tradicionalmente, los desarrolladores han confiado en su propia experiencia, documentación y fragmentos de código de diversas fuentes para escribir y mantener el código. Si bien estos métodos han sido útiles para el sector, pueden llevar mucho tiempo y ser propensos a errores humanos, lo que genera ineficiencias y posibles errores.

Aquí es donde Amazon Q Developer interviene para mejorar la experiencia del desarrollador. Amazon Q Developer es un potente asistente AWS generativo basado en inteligencia artificial diseñado para acelerar las tareas de desarrollo de código al proporcionar recomendaciones y generación de código inteligentes.

Sin embargo, como ocurre con cualquier tecnología nueva, pueden surgir desafíos. Las expectativas poco realistas, las dificultades de incorporación, la resolución de problemas de generación de código inexacta y el uso adecuado de las capacidades de Amazon Q son obstáculos comunes a los que se pueden enfrentar los desarrolladores. Esta guía completa aborda estos desafíos y proporciona escenarios reales, mejores prácticas detalladas, solución de problemas y ejemplos prácticos de código reales específicos para Python y Java, dos de los lenguajes de programación más adoptados.

Esta guía se centra en el uso de Amazon Q Developer para realizar tareas de desarrollo de código, tales como:

- Finalización del código: genere sugerencias en línea a medida que los desarrolladores programan en tiempo real.
- Mejoras y consejos sobre el código: analice el desarrollo de software, genere código nuevo con lenguaje natural y mejore el código existente.

Objetivos

El objetivo de esta guía es ayudar a los desarrolladores que son usuarios nuevos o continuos de Amazon Q Developer, ayudándoles a utilizar el servicio con éxito en sus tareas de programación

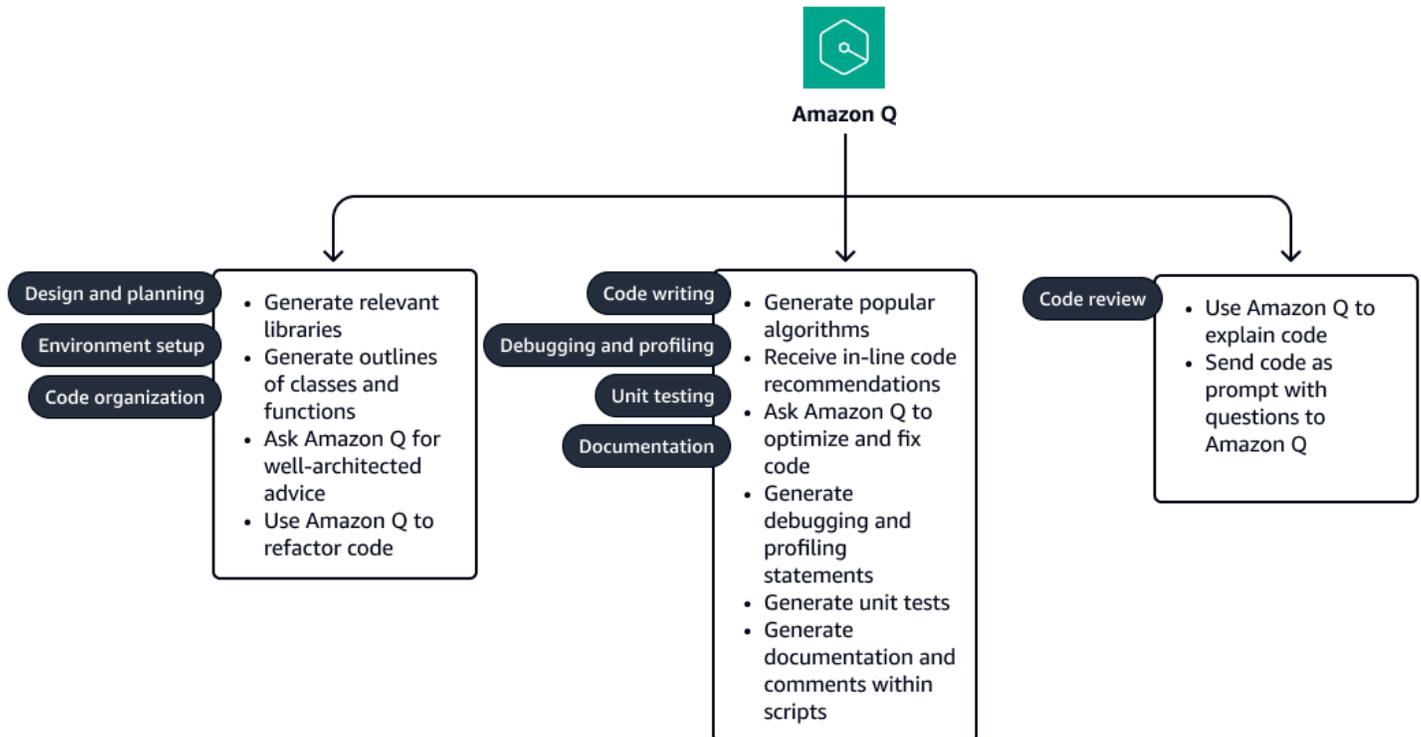
diarias. Los directores de los equipos de desarrollo también pueden beneficiarse de la lectura de esta guía.

Esta guía le ofrece la siguiente información sobre el uso de Amazon Q Developer:

- Comprenda el uso efectivo de Amazon Q Developer para el desarrollo de código
 - Describa las prácticas recomendadas para integrar Amazon Q Developer en el [flujo de trabajo de un desarrollador](#).
 - Ofrezca step-by-step orientación con ejemplos para una [generación de código exitosa y recomendaciones](#).
- Mitigue los desafíos comunes y promueva la claridad de los desarrolladores a la hora de utilizar Amazon Q Developer
 - Ofrezca [estrategias](#) e ideas para cumplir con las expectativas de los desarrolladores y superar los obstáculos relacionados con la precisión y el rendimiento de la generación de código.
- Comportamiento de errores y gestión de errores
 - Proporcione a los desarrolladores la [guía de solución de problemas de generación de código para desarrolladores de Amazon Q](#) para abordar los resultados imprecisos o los comportamientos inesperados.
 - Proporcione [ejemplos reales y escenarios](#) específicos para Python y Java.
- Optimice los flujos de trabajo y la productividad
 - Optimice los flujos de trabajo de desarrollo de código con Amazon Q Developer.
 - Analice las estrategias para mejorar la [productividad de los desarrolladores](#).

Uso de Amazon Q Developer en los flujos de trabajo de desarrolladores

Los desarrolladores siguen un flujo de trabajo estándar que abarca las etapas de recopilación de requisitos, [diseño y planificación](#), [codificación](#), pruebas, [revisión del código](#) e [implementación](#). Esta sección se centra en cómo puede utilizar las capacidades de Amazon Q Developer para optimizar los pasos clave del desarrollo.



El diagrama anterior muestra cómo Amazon Q Developer puede acelerar y agilizar las siguientes tareas comunes en las etapas del desarrollo del código:

- Diseño y planificación | Configuración del entorno | Organización del código
 - Generar bibliotecas relevantes
 - Genere esquemas de clases y funciones
 - Pide a Amazon Q consejos sobre una buena arquitectura
 - Uso de Amazon Q para refactorizar código
- Escritura de código | Depuración y creación de perfiles | Pruebas unitarias | Documentación
 - Generar algoritmos populares

- Reciba recomendaciones de código en línea
- Pide a Amazon Q que optimice y corrija el código
- Genere declaraciones de depuración y creación de perfiles
- Generar pruebas unitarias
- Genere documentación y comentarios dentro de los scripts
- Revisión de código
 - Pide a Amazon Q que te explique el código
 - Envíe el código cuando se le solicite con sus preguntas a Amazon Q

Diseño y planificación

Tras recopilar los requisitos comerciales y técnicos, los desarrolladores diseñan nuevas bases de código o amplían las existentes. Durante esta fase, Amazon Q Developer puede ayudar a los desarrolladores a realizar las siguientes tareas:

- Genere bibliotecas y esquemas de clases y funciones relevantes para obtener consejos bien estructurados.
- Proporcione orientación para consultas de ingeniería, compatibilidad y diseño arquitectónico.

Codificación

El proceso de codificación utiliza Amazon Q Developer para acelerar el desarrollo de las siguientes maneras:

- Configuración del entorno: instálelo AWS Toolkit en su entorno de desarrollo integrado (IDE) (por ejemplo, VS Code o IntelliJ). A continuación, utilice Amazon Q para generar bibliotecas o recibir sugerencias de configuración en función de los objetivos de su proyecto. Para obtener más información, consulta [las prácticas recomendadas para la incorporación de Amazon Q Developer](#).
- Organización del código: Refactoriza el código u obtén recomendaciones de organización de Amazon Q que se ajusten a los objetivos de tu proyecto.
- Redacción de código: utilice sugerencias en línea para generar código durante el desarrollo o pida a Amazon Q que genere código mediante el panel de chat de Amazon Q de su IDE. Para obtener más información, consulte [Prácticas recomendadas para la generación de código con Amazon Q Developer](#).

- Depuración y creación de perfiles: genere comandos de creación de perfiles o utilice opciones de Amazon Q, como Fix y Explain, para depurar problemas.
- Pruebas unitarias: proporciona el código como mensaje a Amazon Q durante una sesión de chat y solicita la generación de pruebas unitarias correspondiente. Para obtener más información, consulte [Ejemplos de código con Amazon Q Developer](#).
- Documentación: utilice las sugerencias en línea para crear comentarios y cadenas de documentos, o utilice la opción Explicar para generar resúmenes detallados de las selecciones de código. Para obtener más información, consulte [Ejemplos de código con Amazon Q Developer](#).

Revisión de código

Los revisores deben comprender el código de desarrollo antes de promocionarlo para su fase de producción. Para acelerar este proceso, usa las opciones de Amazon Q Explain y Optimize, o envía selecciones de códigos con instrucciones rápidas personalizadas a Amazon Q en una sesión de chat. Para obtener más información, consulta los [ejemplos de chat](#).

Integración e implementación

Pida a Amazon Q orientación sobre la integración continua, los procesos de entrega y las mejores prácticas de implementación específicas de la arquitectura de su proyecto.

Con estas recomendaciones, podrá aprender a aprovechar de forma eficaz las funciones de Amazon Q Developer, a optimizar sus flujos de trabajo y a aumentar la productividad durante todo el ciclo de vida del desarrollo.

Capacidades avanzadas de Amazon Q Developer

Si bien esta guía se centra en el uso de Amazon Q Developer en tareas prácticas de programación, es importante conocer sus siguientes capacidades avanzadas:

- Transformación del código de Amazon Q Developer
- Personalizaciones de Amazon Q Developer

Transformación del código de Amazon Q Developer

El Amazon Q Developer Agent para la transformación de código puede actualizar la versión en lenguaje de código de sus archivos sin necesidad de volver a escribir el código manualmente. Funciona analizando los archivos de código existentes y reescribiéndolos automáticamente para utilizar una versión más reciente del lenguaje. Por ejemplo, Amazon Q transforma un solo módulo si trabajas en un IDE como Eclipse. Si utiliza Visual Studio Code, Amazon Q puede transformar todo un proyecto o un espacio de trabajo.

Utilice Amazon Q cuando desee realizar tareas comunes de actualización de código, como las siguientes:

- Actualice el código para que funcione con la nueva sintaxis de la versión del idioma.
- Ejecute pruebas unitarias para validar que la compilación y la ejecución se hayan compilado y ejecutado correctamente.
- Compruebe y resuelva los problemas de implementación.

Amazon Q puede ahorrar a los desarrolladores días o meses de trabajo tedioso y repetitivo para actualizar las bases de código.

A partir de junio de 2024, Amazon Q Developer admite la actualización Java codifica y puede transformar Java Código 8 para versiones más recientes, como Java 11 o 17.

Personalizaciones de Amazon Q Developer

Gracias a su capacidad de personalización, Amazon Q Developer puede ofrecer sugerencias en línea basadas en el código base de la propia empresa. La empresa proporciona su repositorio de

código a Amazon Simple Storage Service (Amazon S3) o AWS CodeConnections a través de lo que antes se conocía como AWS CodeStar Connections. Luego, Amazon Q usa el repositorio de código personalizado con seguridad habilitada para recomendar patrones de codificación que sean relevantes para los desarrolladores de esa organización.

Cuando utilice las personalizaciones de Amazon Q Developer, tenga en cuenta lo siguiente:

- A partir de junio de 2024, la función de personalización para desarrolladores de Amazon Q estará en modo de vista previa. Como resultado, es posible que la función tenga una disponibilidad y un soporte limitados.
- Las sugerencias de código en línea personalizadas solo serán precisas si se tiene en cuenta la calidad de los repositorios de código que se proporcionan. Te recomendamos que revise una [puntuación de evaluación](#) para cada personalización que crees.
- Para optimizar el rendimiento, le recomendamos que incluya al menos 20 archivos de datos que contengan el idioma indicado y que todos los archivos fuente ocupen más de 10 MB. Asegúrese de que su repositorio esté formado por código fuente de referencia y no por archivos de metadatos (por ejemplo, archivos de configuración, archivos de propiedades y archivos readme).

Al utilizar las personalizaciones para desarrolladores de Amazon Q, puede ahorrar tiempo de las siguientes maneras:

- Utilice recomendaciones basadas en el código propietario de su propia empresa.
- Aumente la reutilización de las bases de código existentes.
- Cree patrones repetibles que se generalicen en toda su empresa.

Mejores prácticas de codificación con Amazon Q Developer

En esta sección se describen las prácticas recomendadas para programar con Amazon Q Developer. Las prácticas recomendadas incluyen las siguientes categorías:

- [Incorporación](#): métodos y consideraciones al momento de la incorporación
- [Generación de código](#): orientación para utilizar la generación de código con éxito
- [Recomendaciones de código](#): técnicas para mejorar el código

Mejores prácticas para la incorporación de Amazon Q Developer

Amazon Q Developer es un potente asistente de codificación de IA generativo que está disponible a través de populares IDEs como Visual Studio Code y JetBrains. Esta sección se centra en las prácticas recomendadas para acceder e incorporar Amazon Q Developer a su entorno de desarrollo de programación.

Requisitos previos para Amazon Q Developer

Amazon Q Developer está disponible como parte de AWS Toolkit for Visual Studio Code y AWS Toolkit for JetBrains (por ejemplo, PyCharm IntelliJ y). Para Visual Studio Code y JetBrains IDEs Amazon Q Developer es compatible con Python, Java, JavaScript, C# TypeScript, Go, Rust, PHP, Ruby, Kotlin, C, C++, scripts de Shell, SQL y Scala.

Para obtener instrucciones detalladas sobre cómo instalar el AWS Toolkit código de Visual Studio y un JetBrains IDE, consulte [Instalación de la extensión o el complemento Amazon Q Developer en su IDE](#) en la Guía del usuario para desarrolladores de Amazon Q.

Prácticas recomendadas al utilizar Amazon Q Developer

Entre las prácticas recomendadas generales a la hora de utilizar Amazon Q Developer se incluyen las siguientes:

- Proporcione el contexto pertinente para obtener respuestas más precisas, como los lenguajes de programación, los marcos y las herramientas que se utilizan. Divida los problemas complejos en componentes más pequeños.
- Experimenta y repite tus indicaciones y preguntas. La programación a menudo implica probar diferentes enfoques.

- Revisa siempre las sugerencias de código antes de aceptarlas y edítalas según sea necesario para asegurarte de que cumplen exactamente lo que pretendías.
- Utilice la [capacidad de personalización](#) para que Amazon Q Developer conozca sus bibliotecas internas APIs, mejores prácticas y patrones arquitectónicos para obtener recomendaciones más relevantes.

Privacidad de datos y uso del contenido en Amazon Q Developer

Cuando decida utilizar Amazon Q Developer, debe comprender cómo se utilizan sus datos y su contenido. Los puntos clave son los siguientes:

- Para los usuarios de Amazon Q Developer Pro, el contenido de su código no se utiliza para mejorar el servicio ni para formar modelos.
- Los usuarios de Amazon Q Developer Free Tier pueden optar por que su contenido no se utilice para mejorar el servicio mediante la configuración o AWS Organizations las políticas del IDE.
- El contenido transmitido está cifrado y cualquier contenido almacenado está protegido con cifrado en reposo y controles de acceso. Para obtener más información, consulte [Cifrado de datos en Amazon Q Developer](#) en la Guía del usuario para desarrolladores de Amazon Q.

Prácticas recomendadas para la generación de código con Amazon Q Developer

Amazon Q Developer ofrece generación automática de código, finalización automática y sugerencias de código en lenguaje natural. Las siguientes son las prácticas recomendadas para utilizar la asistencia de codificación en línea para desarrolladores de Amazon Q:

- Proporcione un contexto para ayudar a mejorar la precisión de las respuestas

Comience con el código existente, importe bibliotecas, cree clases y funciones o establezca esqueletos de código. Este contexto ayudará a mejorar significativamente la calidad de la generación de código.

- Codifica naturalmente

Utilice la generación de código para desarrolladores de Amazon Q como un potente motor de autocompletado. Codifica como lo haces normalmente y deja que Amazon Q te dé sugerencias

mientras escribes o pausas. Si la generación de código no está disponible o tienes un problema con el código, inicia Amazon Q escribiendo Alt+C en un PC u Opción+C en macOS. Para obtener más información sobre las acciones habituales que puede realizar al utilizar las sugerencias integradas, consulte [Uso de atajos de teclado](#) en la Guía del usuario para desarrolladores de Amazon Q.

- Incluye bibliotecas de importación que sean relevantes para los objetivos de tu script

Incluya las bibliotecas de importación pertinentes para ayudar a Amazon Q a entender el contexto y generar código en consecuencia. También puedes pedir a Amazon Q que te sugiera las declaraciones de importación pertinentes.

- Mantenga un contexto claro y centrado

Mantén tu guion centrado en objetivos específicos y modulariza las distintas funcionalidades en guiones separados con el contexto pertinente. Evite el contexto ruidoso o confuso.

- Experimenta con las indicaciones

Explore diferentes indicaciones para impulsar a Amazon Q a producir resultados útiles en la generación de código. Por ejemplo, pruebe los siguientes enfoques:

- Usa bloques de comentarios estándar para las indicaciones en lenguaje natural.
 - Crea esqueletos con comentarios para completar las clases y funciones.
 - Sea específico en sus indicaciones y proporcione detalles en lugar de generalizar.
- Chatea con un desarrollador de Amazon Q y pide ayuda

Si Amazon Q Developer no proporciona sugerencias precisas, chatea con Amazon Q Developer en tu IDE. Puede proporcionar fragmentos de código o clases y funciones completas para poner en marcha su contexto. Para obtener más información, consulta [Cómo hablar sobre el código con un desarrollador de Amazon Q](#) en la Guía del usuario para desarrolladores de Amazon Q.

Prácticas recomendadas para recomendaciones de código con Amazon Q Developer

El desarrollador de Amazon Q puede responder a las preguntas de los desarrolladores y evaluar el código para ofrecer recomendaciones que van desde la generación de código y la corrección de errores hasta orientación sobre el uso del lenguaje natural. Estas son las prácticas recomendadas para usar el chat en Amazon Q:

- Genera código desde cero

Para proyectos nuevos o cuando necesite una función general (por ejemplo, copiar archivos de Amazon S3), pida a Amazon Q Developer que genere ejemplos de código utilizando instrucciones en lenguaje natural. Amazon Q puede proporcionar enlaces relacionados a recursos públicos para su posterior validación e investigación.

- Busque conocimientos de codificación y explicaciones sobre los errores

Cuando tengas problemas de codificación o mensajes de error, proporciona el bloque de código (con el mensaje de error, si corresponde) y tu pregunta como mensaje de error al desarrollador de Amazon Q. Este contexto ayudará a Amazon Q a ofrecer respuestas precisas y relevantes.

- Mejore el código existente

Para corregir errores conocidos u optimizar el código (por ejemplo, para reducir la complejidad), selecciona el bloque de código correspondiente y envíalo a Amazon Q Developer junto con tu solicitud. Sea específico con sus indicaciones para obtener mejores resultados.

- Explica la funcionalidad del código

Cuando explore nuevos repositorios de código, seleccione un bloque de código o un script completo y envíelo al desarrollador de Amazon Q para que le dé una explicación. Reduzca el tamaño de la selección para obtener explicaciones más específicas.

- Genera pruebas unitarias

Tras enviar un bloque de código como mensaje, pide a Amazon Q Developer que genere pruebas unitarias. Este enfoque puede ahorrar tiempo y costos de desarrollo relacionados con la cobertura del código y DevOps.

- Encuentre AWS respuestas

Amazon Q Developer es un recurso valioso para los desarrolladores que trabajan con él Servicios de AWS porque contiene una gran cantidad de conocimientos relacionados con AWS. Ya sea que se enfrente a problemas relacionados con un tema específico Servicio de AWS, reciba mensajes de error específicos o intente aprender algo nuevo Servicio de AWS, Amazon Q suele proporcionar información relevante y útil. AWS

Revisa siempre las recomendaciones que te proporciona Amazon Q Developer. A continuación, realice las modificaciones necesarias y ejecute las pruebas para asegurarse de que el código cumple con la funcionalidad prevista.

Ejemplos de código con Amazon Q Developer

En esta sección se proporcionan ejemplos realistas de cómo mejorar la experiencia y la generación de código, centrándose en la Python y Java idiomas. Junto con los ejemplos en línea, algunos escenarios muestran cómo puede utilizar la capacidad de chat de Amazon Q Developer para solicitar solicitudes de recomendaciones de código mientras está programando.

Para obtener más información, consulte los temas siguientes:

- [Python ejemplos](#)
- [Java ejemplos](#)
- [Ejemplos de chat](#)

Python ejemplos

Los siguientes ejemplos de Python Los ejemplos muestran cómo puede empezar a generar sugerencias de código en línea con Amazon Q Developer. Estos ejemplos ilustrativos le muestran cómo activar el asistente de codificación para sus propios proyectos. La mayoría de los siguientes ejemplos son aplicables a otros lenguajes:

- [Genera clases y funciones](#)
- [Código del documento](#)
- [Genera algoritmos](#)
- [Genera pruebas unitarias](#)

Genera clases y funciones

Para generar una implementación de clase total o parcial, utilice los comentarios de código. El comentario describe la intención de la clase. Comience de forma sencilla y añada más detalles si es necesario. Después de generar una clase, puede continuar generando funciones dentro de su estructura básica.

Al escribir comentarios, es preferible utilizar la estructura de comentarios del idioma. In Python, para comentarios de una sola línea, utilice # .Para comentarios de líneas múltiples, utilice ''' ''' o """ """.

Input:

```
'''
DocumentReader class. has a name, children and optional parent.
Includes I/O functions to read and
write document
'''
```

Salida:

```
class DocumentReader:
    def __init__(self, name, children, parent=None):
        self.name = name
        self.parent = parent
        self.children = children

    def read(self):
        """Reads current document and returns text"""
        pass

    def write(self, text):
        """Writes text to current document"""
        pass
```

Ahora que hay una clase inicial, se puede generar una función adicional con comentarios adicionales.

Input:

```
'''
function to read document from local directory path and print the first line of the
document
'''
```

Salida:

```
def read_document(self, path):
    """Reads document from path and prints first line"""
    import os
    if os.path.isfile(path):
        with open(path, 'r') as f:
```

```
        print(f.readline())
    else:
        print(f"File not found at {path}")
```

Código del documento

Reduzca el tiempo dedicado a la documentación del código al confiar en la documentación en línea. Add (Suma) Python pausas `''' '''` de comentarios.

Input:

```
class S3Utils:
    """
    <place cursor here>
    """
    def __init__(self):
        self.base_url = "https://s3"

    def generate_s3_url(self, bucket_name, region):
        return f"{self.base_url}.{region}.amazonaws.com/{bucket_name}"

# Example usage:
if __name__ == "__main__":
    s3_util = S3Utils()
    bucket_name = "my-example-bucket"
    region = "us-west-1"
    s3_url = s3_util.generate_s3_url(bucket_name, region)
    print(f"S3 URL for bucket '{bucket_name}' in region '{region}': {s3_url}")
```

Salida:

```
class S3Utils:
    """
    Class to generate S3 URL from bucket name and region
    """
    def __init__(self):
        self.base_url = "https://s3"

    def generate_s3_url(self, bucket_name, region):
        return f"{self.base_url}.{region}.amazonaws.com/{bucket_name}"

# Example usage:
```

```
if __name__ == "__main__":
    s3_util = S3Utils()
    bucket_name = "my-example-bucket"
    region = "us-west-1"
    s3_url = s3_util.generate_s3_url(bucket_name, region)
    print(f"S3 URL for bucket '{bucket_name}' in region '{region}': {s3_url}")
```

Generar algoritmos

Amazon Q Developer puede generar algoritmos populares, especialmente los relacionados con AWS SDKs. En el script, añade un mensaje que especifique la función que desea.

En este ejemplo se muestra cómo generar un diccionario de claves de identificación de usuario para asignar nombres a valores y, a continuación, generar una función de ordenación mediante el algoritmo de ordenación de burbujas.

Input:

```
class SortList:
    @staticmethod
    def get_user_id_name_map():
        user_id_name_map = {
            1: "Amit",
            2: "Rahul",
            3: "Vijay",
            4: "Raj",
            5: "Kumar"
        }
        return user_id_name_map

'''
Sort the list of user id's alphabetically using bubble sort
'''
```

Salida:

```
@staticmethod
def sort_user_id_name_map():
    user_id_name_map = SortList.get_user_id_name_map()
    list_of_user_ids = list(user_id_name_map.keys())
```

```

list_of_user_names = list(user_id_name_map.values())
print("Before sort: ", list_of_user_ids)
for i in range(len(list_of_user_ids)):
    for j in range(len(list_of_user_ids) - i - 1):
        if list_of_user_names[j] > list_of_user_names[j + 1]:
            list_of_user_ids[j], list_of_user_ids[j + 1] = list_of_user_ids[j +
1], list_of_user_ids[j]
            list_of_user_names[j], list_of_user_names[j + 1] =
list_of_user_names[j + 1], list_of_user_names[j]
    print("After sort: ", list_of_user_ids)

```

Además, puede crear solicitudes para los procedimientos estándar. En este ejemplo, el usuario crea un mensaje para almacenar el diccionario en una tabla de Amazon DynamoDB. Al principio del script, el usuario añade un mensaje para añadir bibliotecas de DynamoDB a fin de proporcionar contexto adicional a Amazon Q Developer.

Input:

```

...
Add libraries for Amazon Dynamo DB
...

```

Salida:

```

...
Add libraries for Amazon Dynamo DB and JSON.

import necessary libraries for working with Amazon DynamoDB and JSON data.
...

import boto3
import json
from botocore.exceptions import ClientError

def load_data_to_dynamodb(table_name, data_file):
    # Create a DynamoDB resource
    dynamodb = boto3.resource('dynamodb')

    # Get a reference to the table
    table = dynamodb.Table(table_name)

    # Load data from the JSON file

```

```

with open(data_file, 'r') as file:
    data = json.load(file)

# Iterate over the data and load it to the table
for item in data:
    try:
        table.put_item(Item=item)
        print(f"Successfully loaded data: {item}")
    except ClientError as e:
        print(f"Error loading data: {e.response['Error']['Message']}")

# Example usage
load_data_to_dynamodb('myTable', 'data.json')

```

Ahora, el usuario puede pedir fácilmente ayuda a Amazon Q para almacenar el diccionario en DynamoDB.

Input:

```

'''
function to store the above sorted dict to dynamodb table named users
'''

```

Salida:

```

@staticmethod
def store_user_id_name_map_in_dynamodb():
    user_id_name_map = SortList.get_user_id_name_map()
    list_of_user_ids = list(user_id_name_map.keys())
    list_of_user_names = list(user_id_name_map.values())
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('users')
    for i in range(len(list_of_user_ids)):
        try:
            response = table.put_item(
                Item={
                    'user_id': list_of_user_ids[i],
                    'user_name': list_of_user_names[i]
                }
            )
        except ClientError as e:
            print(e.response['Error']['Message'])

```

Genere pruebas unitarias

Con los comentarios en línea, Amazon Q Developer puede generar pruebas unitarias para realizar afirmaciones y otras pruebas. En este ejemplo, el usuario pide a Amazon Q que genere las bibliotecas aplicables para las pruebas unitarias a fin de establecer el contexto del script de prueba. A continuación, el usuario pide a Amazon Q que escriba una función de prueba para la `sort_user_id_name_map` función.

Input:

```
'''
Import all the required unit test libraries for python and then generate a unit test
for sort_user_id_name_map function.
'''
```

Salida:

```
import unittest
from unittest.mock import patch
from sort_user_id_name_map import sort_user_id_name_map

class TestSortUserIdNameMap(unittest.TestCase):
    @patch('sort_user_id_name_map.sort_user_id_name_map')
    def test_sort_user_id_name_map(self, mock_sort):
        mock_sort.return_value = {'id1': 'name1', 'id2': 'name2'}
        result = sort_user_id_name_map({'id2': 'name2', 'id1': 'name1'})
        self.assertEqual(result, {'id1': 'name1', 'id2': 'name2'})

if __name__ == '__main__':
    unittest.main()
```

Además, en un chat entre el usuario y el desarrollador de Amazon Q, sugiere y genera pruebas unitarias basadas en las instrucciones de entrada de código del usuario. Para obtener más información, consulte los [ejemplos de chat](#).

Java ejemplos

Los siguientes ejemplos de Java los ejemplos demuestran cómo los usuarios pueden empezar a generar sugerencias de código en línea. Estos ejemplos ilustrativos le muestran cómo activar el

asistente de codificación para sus propios proyectos. La mayoría de los siguientes ejemplos son aplicables a otros lenguajes:

- [Genera clases y funciones](#)
- [Código del documento](#)
- [Genera algoritmos](#)
- [Genera pruebas unitarias](#)

Genera clases y funciones

Para generar una implementación de clase total o parcial, utilice los comentarios de código. El comentario describe la intención de la clase. Comience de forma sencilla y añada más detalles si es necesario. Después de generar una clase, puede continuar generando funciones dentro de su estructura básica.

Al escribir comentarios, es preferible utilizar la estructura de comentarios del idioma. In Java, esto es `/** */`.

Input:

```
/**
 * DocumentReader class. has a name, children and optional parent.
 * Includes I/O functions to read and
 * write document
 */
```

Salida:

```
/**
 * DocumentReader class. has a name, children and optional parent.
 * Includes I/O functions to read and
 * write document
 */
import java.util.ArrayList;
public class DocumentReader {
    private String name;
    private DocumentReader parent;
    private ArrayList<DocumentReader> children;

    /**
```

```
* Constructor
* @param name name of the document
* @param parent parent of the document
*/
public DocumentReader(String name, DocumentReader parent) {
    this.name = name;
    this.parent = parent;
    this.children = new ArrayList<>();
}

/**
 * Constructor
 * @param name name of the document
 */
public DocumentReader(String name) {
    this(name, null);
}

/**
 * Getter for name
 * @return name of the document
 */
public String getName() {
    return name;
}

/**
 * Getter for parent
 * @return parent of the document
 */
public DocumentReader getParent() {
    return parent;
}

/**
 * Getter for children
 * @return children of the document
 */
public ArrayList<DocumentReader> getChildren() {
    return children;
}

/**
 * Adds a child to the document
```

```
    * @param child child to be added
    */
    public void addChild(DocumentReader child) {
        children.add(child);
    }

    /**
     * Removes a child from the document
     * @param child child to be removed
     */
    public void removeChild(DocumentReader child) {
        children.remove(child);
    }
}
```

Ahora que hay una clase inicial, se puede generar una función adicional con comentarios adicionales.

Input:

```
/**
 * function to read document from local directory path
 * and print first line of document
 */
```

Salida:

```
/**
 * function to read document from local directory path
 * and print first line of document
 */
public void readDocument() {
    try {
        BufferedReader reader = new BufferedReader(new FileReader(name));
        String line = reader.readLine();
        System.out.println(line);
        reader.close();
    } catch (IOException e) {
        System.out.println("Error reading document: " + e.getMessage());
    }
}
```

Código del documento

Reduzca el tiempo dedicado a la documentación del código al confiar en la documentación en línea. Añada un Java pausa de comentarios/* */.

Input:

```
import java.util.function.Function;

public class S3Utils {
    /*
     * <place cursor here>
     */
    public static Function<String, String> generateS3Function(String bucketName, String
region) {
        return (String key) -> {
            String url = generateS3Url(bucketName, key, region);
            return url;
        };
    }
}
```

Si no se produce la generación en línea, ejecute una sugerencia en línea para desarrolladores de Amazon Q después de colocar el cursor donde desee obtener la documentación.

Salida:

```
import java.util.function.Function;

public class S3Utils {
    /*
     * Generate a function that returns the S3 URL for a given key
     * @param bucketName the name of the bucket
     * @param region the region of the bucket
     * @return the function that generates the S3 URL
     */
    public static Function<String, String> generateS3Function(String bucketName, String
region) {
        return (String key) -> {
            String url = generateS3Url(bucketName, key, region);
            return url;
        };
    }
}
```

```
}  
}
```

Para obtener explicaciones más sencillas sobre el código (clases, funciones o elementos del script), utilice caracteres (//) situados estratégicamente antes de las secciones del código en las que desee añadir la documentación.

Input:

```
//  
public class S3Utils {  
    /*  
     * Generate a function that returns the S3 URL for a given key  
     * @param bucketName the name of the bucket  
     * @param region the region of the bucket  
     * @return the function that generates the S3 URL  
     */  
    public static Function<String, String> generateS3Function(String bucketName, String  
region) {  
        return (String key) -> {  
            String url = generateS3Url(bucketName, key, region);  
            return url;  
        };  
    }  
}
```

Salida:

```
// This class is used to generate a function that returns the S3 URL for a given key  
public class S3Utils {  
    /*  
     * Generate a function that returns the S3 URL for a given key  
     * @param bucketName the name of the bucket  
     * @param region the region of the bucket  
     * @return the function that generates the S3 URL  
     */  
    public static Function<String, String> generateS3Function(String bucketName, String  
region) {  
        return (String key) -> {  
            String url = generateS3Url(bucketName, key, region);  
            return url;  
        };  
    }  
}
```

```
}  
}
```

Genere algoritmos

Amazon Q Developer puede generar algoritmos populares, especialmente relacionados con AWS SDKs. En el script, añada un mensaje que especifique la función que desea.

Este ejemplo genera un mapa hash de claves de identificación de usuario para nombrar valores y, a continuación, genera una función de clasificación mediante el algoritmo de clasificación por burbujas.

Input:

```
import java.util.HashMap;  
public class SortList {  
  
    // generate key value pair hash map for  
    // user id and name  
    public static HashMap<Integer, String> getUserIdNameMap() {  
        HashMap<Integer, String> userIdNameMap = new HashMap<Integer, String>();  
        userIdNameMap.put(1, "Amit");  
        userIdNameMap.put(2, "Rahul");  
        userIdNameMap.put(3, "Vijay");  
        userIdNameMap.put(4, "Raj");  
        userIdNameMap.put(5, "Kumar");  
        return userIdNameMap;  
    }  
  
    /*  
    * Create function to sort UserIdName map by name alphabetically using bubble sort  
    */  
  
}
```

Salida:

```
/*  
* Create function to sort UserIdName map by name alphabetically using bubble sort  
*/  
public static HashMap<Integer, String> sortUserIdNameMap(HashMap<Integer, String>  
    userIdNameMap) {
```

```

int size = userIdNameMap.size();
for (int i = 0; i < size; i++) {
    for (int j = i + 1; j < size; j++) {
        if (userIdNameMap.get(i).compareTo(userIdNameMap.get(j)) > 0) {
            String temp = userIdNameMap.get(i);
            userIdNameMap.put(i, userIdNameMap.get(j));
            userIdNameMap.put(j, temp);
        }
    }
}
return userIdNameMap;
}

```

Además, los usuarios pueden crear solicitudes para los procedimientos estándar. En este ejemplo, el usuario crea un mensaje para almacenar el mapa hash en una tabla de Amazon DynamoDB. Al principio del script, el usuario añade un mensaje para añadir DynamoDB Java bibliotecas para ayudar a proporcionar contexto adicional a Amazon Q Developer.

Input:

```

// add libraries for Amazon DynamoDB
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;

/*
 * Store sorted UserIdName map into Amazon DynamoDB table named users
 */

```

Salida:

```

// add libraries for Amazon DynamoDB
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;

/*
 * Store sorted UserIdName map into Amazon DynamoDB table named users
 */
public static void storeSortedUserIdNameMap(HashMap<Integer, String> userIdNameMap)
{
    // create Amazon DynamoDB client
    AmazonDynamoDB client = AmazonDynamoDBClientBuilder.defaultClient();
}

```

```
// store sorted UserIdName map into Amazon DynamoDB table named users
for (int userId : userIdNameMap.keySet()) {
    String name = userIdNameMap.get(userId);
    client.putItem("users", new HashMap<String, Object>() {{
        put("userId", userId);
        put("name", name);
    }});
}
}
```

Genere pruebas unitarias

Con los comentarios en línea, Amazon Q Developer puede generar pruebas unitarias para realizar afirmaciones y otras JUnit pruebas. En este ejemplo, el usuario primero pide a Amazon Q que genere las bibliotecas aplicables JUnit para establecer el contexto del script de prueba. A continuación, el usuario crea una clase pública seguida de un declarador y cadenas de comentarios.

Input:

```
// import libraries for java unit tests
import org.junit.Test;
import static org.junit.Assert.*;

public class UnitTests {

    @Test
    /*
     * Create unit test for getUserIdNameMap
     */
}
```

Salida:

```
@Test
/*
 * Create unit test for getUserIdNameMap
 */
public void testGetUserIdNameMap() {
    // create a new instance of the UserIdNameMap class
    UserIdNameMap userIdNameMap = new UserIdNameMap();
```

```
// create a new instance of the User class
User user = new User("John", "Doe");

// add the user to the map
userIdNameMap.addUser(user);

// get the user from the map
User retrievedUser = userIdNameMap.getUser("John");

// assert that the retrieved user is the same as the user added to the map
assertEquals(user, retrievedUser);
}
```

Además, en un chat entre el usuario y el desarrollador de Amazon Q, sugiere y genera pruebas unitarias basadas en las instrucciones de entrada de código del usuario. Para obtener más información, consulte los [ejemplos de chat](#).

Ejemplos de chat

El chat para desarrolladores de Amazon Q ayuda a los desarrolladores a realizar diversas tareas, desde codificar, probar y actualizar aplicaciones hasta diagnosticar errores, realizar análisis y correcciones de seguridad y optimizar AWS los recursos. En esta sección se explica cómo chatear con Amazon Q puede ayudar a un desarrollador en tareas de programación típicas, como las siguientes:

- [Pregunte acerca de Servicios de AWS](#)
- [Generar código](#)
- [Genera pruebas unitarias](#)
- [Explica el código](#)

Pregunte acerca de Servicios de AWS

Los usuarios pueden hacer preguntas a los desarrolladores de Amazon Q Servicios de AWS, por ejemplo, sobre cómo encontrar el servicio adecuado o comprender las mejores prácticas.

Charle sobre Servicios de AWS

Usuario:

¿Cuál es el mejor servicio de contenedores que puedo utilizar para gestionar mi carga de trabajo si necesito mantener mis costes bajos?

Amazon Q:

Según la información proporcionada, parece que está buscando un servicio de contenedores rentable para ejecutar sus cargas de trabajo. En ese caso, te recomendaría considerar Amazon Elastic Container Service (Amazon ECS) con Fargate. [1]

Amazon ECS with Fargate es un motor de cómputo sin servidor para contenedores que le permite ejecutar sus contenedores sin tener que administrar ninguna infraestructura subyacente. Con Fargate, no necesita aprovisionar ni administrar servidores; Fargate se encarga de ello por usted. Esto puede ayudarle a mantener sus costes bajos, ya que solo paga por los recursos informáticos que realmente utilizan sus contenedores, sin tener que pagar por la capacidad de los servidores inactivos. [2]

Algunas de las principales ventajas de usar Amazon ECS con Fargate:

- No es necesario administrar servidores ni clústeres: Fargate se encarga del aprovisionamiento y la administración de la infraestructura por usted.
- Pay-as-you-go precios: solo paga por los recursos informáticos que utilizan sus contenedores, sin tener que pagar por la capacidad inactiva.
- Escalado automático: Fargate puede escalar automáticamente sus contenedores hacia arriba y hacia abajo en función de la demanda.
- Estrecha integración con otros, Servicios de AWS como el equilibrio de carga, el registro y la supervisión.

Los recursos proporcionados también mencionan que el servicio ECS ofrece flexibilidad en términos de programación y ubicación de los contenedores, lo que puede ayudar a optimizar aún más sus costos.

Te recomiendo revisar los detalles de precios de Amazon ECS con Fargate en el sitio AWS web para tener una mejor idea de los costos en función de tus requisitos de carga de trabajo específicos. Y, por supuesto, ¡no dude en hacérmelo saber si tiene alguna otra pregunta!

Orígenes

[1] ¿Qué servicio de AWS contenedores es mejor para usted?

[aptgetops.tech > - -tú aws-container-services-which one-is-right-for](#)

[2] Java opciones de contenedor para cargas de trabajo que no son HTTP

[stackoverflow.com > preguntas > 43417016](#)

Finaliza la charla de Amazon Q sobre Servicios de AWS

Generar código

Los usuarios pueden solicitar ayuda a Amazon Q Developer para escribir fragmentos de código.

Charla sobre cómo escribir código

Usuario:

¿Puede darme un ejemplo de cómo enumerar todos los cubos de S3 utilizando el AWS SDK for Python (Boto)?

Amazon Q:

Sin duda, he aquí un ejemplo de cómo enumerar todos los buckets de S3 utilizando el AWS SDK for Python (Boto)[1]:

```
import boto3
# Create an S3 client
s3 = boto3.client('s3')

# List all the buckets
response = s3.list_buckets()

# Print the name of each bucket
for bucket in response['Buckets']:
    print(bucket['Name'])
```

Así es como funciona el código:

1. Primero importamos la biblioteca boto3, que es el SDK para Python.
2. Creamos un cliente S3 usando `boto3.client('s3')`
3. Llamamos al `list_buckets()` método en el cliente S3, que devuelve una respuesta que contiene una lista de todos los cubos de la cuenta.

4. A continuación, repasamos la Buckets lista de la respuesta e imprimimos el nombre de cada depósito.

Este es un ejemplo básico, pero AWS SDK for Python (Boto) proporciona muchas más funciones avanzadas para trabajar con Amazon S3, como crear, eliminar y administrar objetos en depósitos, establecer políticas de depósitos y más. Puede encontrar más ejemplos y documentación en la documentación AWS SDK for Python (Boto)(). [2]

Recuerde que, al trabajar con ellas Servicios de AWS, es importante seguir las mejores prácticas para gestionar las credenciales y la información confidencial, como almacenar las claves de AWS acceso en las variables de entorno en lugar de codificarlas en el código.

Fuentes: [1] [Ejemplos de Amazon S3 con el SDK para Python \(Boto3\) - Ejemplos de código de SDK docs.aws.amazon.com > code-library AWS > latest-1x >... > python-s3.html](https://docs.aws.amazon.com/code-library/AWS/latest-1x/python-s3.html)

[2] [Comience a utilizar los buckets y objetos de Amazon S3 mediante un AWS SDK: AWS ejemplos de código de SDK docs.aws.amazon.com > code-library > latest-1x >... > s3_example_s3_buckets_gettingstarted.html](https://docs.aws.amazon.com/code-library/AWS/latest-1x/s3_example_s3_buckets_gettingstarted.html)

Finaliza la charla de Amazon Q sobre cómo escribir código

Genera pruebas unitarias

Los usuarios pueden enviar el código seleccionado como mensaje al desarrollador de Amazon Q, donde se les aplican instrucciones adicionales, como la generación de una prueba unitaria. Para step-by-step obtener instrucciones, consulta [Enviar código a Amazon Q](#) en la Guía del usuario para desarrolladores de Amazon Q.

Hable sobre las pruebas unitarias

Usuario:

Escribe una prueba unitaria con datos simulados para la siguiente función:

```
// poll messages from Amazon SQS queue name
public List<String> pollMessages(String queueName) {
    List<String> messages = new ArrayList<>();
    // poll messages from queue
    ReceiveMessageResult = sqsClient.receiveMessage(new ReceiveMessageRequest().
    for (Message : receiveMessageResult.getMessages()) {
```

```
        messages.add(message.getBody());
    }
    return messages;
}
```

Amazon Q:

Para escribir una prueba unitaria con datos simulados para la función `PollMessages`, podemos usar un marco de simulación como Mockito. A continuación se muestra un ejemplo:

```
import com.amazonaws.services.sqs.AmazonSQS;
import com.amazonaws.services.sqs.model.Message;
import com.amazonaws.services.sqs.model.ReceiveMessageRequest;
import com.amazonaws.services.sqs.model.ReceiveMessageResult;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.Mockito;
import org.mockito.junit.jupiter.MockitoExtension;

import java.util.Arrays;
import java.util.List;

import static org.junit.jupiter.api.Assertions.assertEquals;

@ExtendWith(MockitoExtension.class)
public class SQSUtilsTest {

    @Mock
    private AmazonSQS sqsClient;

    @InjectMocks
    private SQSUtils sqsUtils;

    @Test
    public void testPollMessages() {
        // Arrange
        String queueName = "test-queue";
        Message message1 = new Message().withBody("message1");
        Message message2 = new Message().withBody("message2");
        ReceiveMessageResult receiveMessageResult = new
        ReceiveMessageResult().withMessages(message1, message2);
```

```
Mockito.when(sqsClient.receiveMessage(new
ReceiveMessageRequest().withQueueUrl(queueName)))
    .thenReturn(receiveMessageResult);

// Act
List<String> messages = sqsUtils.pollMessages(queueName);

// Assert
assertEquals(Arrays.asList("message1", "message2"), messages);
}
}
```

En este ejemplo, utilizamos Mockito para simular el cliente AmazonSQS y configurar el comportamiento esperado del método `ReceiveMessage`, de modo que devuelva dos mensajes simulados con los cuerpos «message1" y «message2". Luego, llamamos al método `pollMessages` y afirmamos que la lista de mensajes devuelta coincide con los valores esperados.

Los puntos clave son:

1. Usamos la anotación `@ ExtendWith (MockitoExtension.class)` para permitir la integración de Mockito con. JUnit
2. Usamos la anotación `@Mock` para crear una instancia simulada del cliente AmazonSQS.
3. Usamos la `InjectMocks` anotación `@` para introducir el cliente simulado de AmazonSQS en la clase. `SQSUtils`
4. En el método de prueba, configuramos el comportamiento esperado del método `ReceiveMessage` mediante `Mockito.when () .thenReturn ()`.
5. Llamamos al método `pollMessages` y afirmamos que la lista de mensajes devuelta coincide con los valores esperados.

Este enfoque nos permite probar el método `PollMessages` sin interactuar realmente con el servicio Amazon SQS, lo que puede hacer que la prueba sea más rápida y fiable.

Finaliza la charla de Amazon Q sobre las pruebas unitarias

Explica el código

Para los procesos de revisión del código, los desarrolladores pueden enviar una selección de código a Amazon Q Developer para comprender mejor el código. Además, la explicación que proporciona

Amazon Q también puede servir para fines de documentación. En el siguiente ejemplo, Amazon Q explica la función de controlador con una AWS Lambda función y, a continuación, propone una cadena de documentos.

Charla sobre cómo explicar el código

Usuario:

Explica la siguiente parte de mi código:

```
def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    s3_client = boto3.client('s3')
    table = dynamodb.Table('DYNAMODB_TABLE')
    # Get s3 object from the event
    print(event)
    print(context)
    # get object from event
    bucket_key = event['Records'][0]['s3']['object']['key']
    object_key = unquote_plus(bucket_key)
    fullText = detect_text(object_key, bucket)

    response = s3_client.head_object(Bucket=bucket, Key=object_key)
    print(response)
    print(fullText)
    metadata = response['ResponseMetadata']
    metadata = json.dumps(metadata)

    now = datetime.now()
    current_time = now.strftime("%A %d %B, %Y %H:%M:%S")
    print("Current Time =", current_time + "\n\n")
    reason, outcome = detect_restaurant_closure(fullText, current_time)
    table.put_item(
        Item={
            'id': object_key,
            'creationTime': now.strftime("%Y-%m-%d %H:%M:%S"),
            'reason': reason,
            'text': fullText,
            'metadata': metadata
        }
    )
```

Amazon Q:

El bloque de código proporcionado parece ser una AWS Lambda función escrita en Python. Permítanme explicar los aspectos clave de este código:

1. La función se denomina `lambda_handler` y es el punto de entrada de la función Lambda.
2. Utiliza la biblioteca `boto3` para interactuar con los servicios AWS DynamoDB y S3.
3. La función recupera una clave de objeto de S3 y un nombre de bucket a partir de los datos del evento entrante.
4. A continuación, llama a una función `detect_text` (que no se muestra en el código proporcionado) para extraer el contenido de texto completo del objeto S3.
5. La función también recupera metadatos sobre el objeto S3 mediante el método `head_object`.
6. A continuación, extrae la hora actual y llama a la función `detect_restaurant_closure` (que tampoco se muestra) para determinar el motivo y el resultado del cierre de un restaurante en función del texto extraído.
7. Por último, la función almacena la siguiente información en una tabla de DynamoDB:
 - La clave del objeto S3 como identificador
 - La hora de creación
 - ¿El motivo y el resultado del cierre del restaurante
 - El texto del objeto S3 extraído

En general, esta función Lambda parece formar parte de un sistema más grande que procesa los objetos S3, extrae el contenido del texto, lo analiza en busca de información sobre el cierre de restaurantes y almacena los resultados en una tabla de DynamoDB. Los detalles específicos de las funciones `detect_text` y `detect_restaurant_closure` serían necesarios para comprender completamente la funcionalidad de este código.

Fin de la respuesta de Amazon Q

Usuario: resume esta explicación en una cadena de documentos para esta función `lambda_handler`

Amazon Q: Esta es una cadena de documentación sugerida para la función `lambda_handler` [1]:

```
def lambda_handler(event, context):  
    """  
    AWS Lambda function that processes an S3 object, extracts text content, analyzes  
    it, and stores the results.  
    Args:
```


Resolución de problemas de generación de código en Amazon Q Developer

Al utilizar Amazon Q Developer, es posible que te encuentres con los siguientes escenarios comunes con resoluciones y generación de código inexactas:

- [Generación de código vacío](#)
- [Comentarios continuos](#)
- [Generación de código en línea incorrecta](#)
- [Resultados inadecuados de los chats](#)

Generación de código vacío

Al desarrollar el código, es posible que notes los siguientes problemas:

- Amazon Q no ofrece ninguna sugerencia.
- Aparece el mensaje «No hay sugerencias de Amazon Q» en su IDE.

Lo primero que piensas es que Amazon Q no funciona correctamente. Sin embargo, la causa principal de estos problemas suele estar asociada con el contexto del script o del proyecto abierto dentro del IDE.

Si el desarrollador de Amazon Q no proporciona una sugerencia automáticamente, puedes usar los siguientes atajos para ejecutar manualmente las sugerencias en línea de Amazon Q:

- PC: Alt+C
- macOS - Opción+C

Para obtener más información, consulte [Uso de atajos de teclado](#) en la Guía del usuario para desarrolladores de Amazon Q.

En la mayoría de los casos, Amazon Q genera una sugerencia. Cuando Amazon Q devuelva el mensaje «No hay sugerencias de Amazon Q», consulta las siguientes soluciones habituales para este problema:

- Contexto inadecuado: asegúrese de que estén presentes las bibliotecas correspondientes. Asegúrese de que las clases y funciones definidas en el script estén relacionadas con el código nuevo.
- Mensaje ambiguo: si el mensaje es confuso o poco claro, Amazon Q tendrá un rendimiento inferior al hacer sugerencias de código en línea. En estos escenarios, sea más específico en el código y en los comentarios para obtener las respuestas adecuadas.

Si te quedas atascado, continúa programando lo mejor que puedas. Por lo general, Amazon Q reconocerá tu intención, lo que hace que te ofrezca sugerencias. Otra opción es enviar un fragmento de código con instrucciones rápidas en un chat de Amazon Q. Para obtener más información, consulta [Enviar código a Amazon Q](#) en la Guía del usuario para desarrolladores de Amazon Q.

Comentarios continuos

Puedes escribir un mensaje con comentarios y recibir solo líneas de comentarios genéricas o repetidas de Amazon Q.

Input:

```
// Kafka test
```

Resultado:

```
// Kafka test  
// Kafka test  
// Kafka test  
// Kafka test
```

Por lo general, este problema se debe a la falta de contexto en el guion o a que las instrucciones rápidas son deficientes. Se recomienda utilizar el formato estándar para los comentarios, dado el idioma. En este ejemplo, el usuario cambió el mensaje `/* */` para usarlo en un bloque de comentarios y ese cambio se tradujo en un mejor rendimiento de Amazon Q.

Input:

```
/*  
 * Create a unit test to push a message to a Kafka topic
```

```
*/
```

Resultado:

En este resultado, el usuario aceptó varias sugerencias de forma secuencial.

```
/*
 * Create a unit test to push a message to a Kafka topic
 */
import org.junit.Test;
import java.util.Properties;
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;
import static org.junit.Assert.assertEquals;
// create function for a Kafka test
public class KafkaTest {
    @Test
    public void testKafka() {
        // create Kafka producer
        Properties props = new Properties();
        props.put("bootstrap.servers", "localhost:9092");
        props.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
        props.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
        KafkaProducer<String, String> producer = new KafkaProducer<>(props);
        // create Kafka topic
        String topic = "test";
        // send message to Kafka topic
        producer.send(new ProducerRecord<>(topic, "key", "value"));
        // close Kafka producer
        producer.close();
        // assert that message was sent successfully
        assertEquals("value", KafkaTest.consumer.poll().value());
    }
}
```

Generación de código en línea incorrecta

En ocasiones, las sugerencias en línea para desarrolladores de Amazon Q no son útiles o no cumplen el objetivo de tu mensaje. A veces, las sugerencias de código en línea pueden proporcionar una función genérica similar a un método simple de Hello World.

Cuando las sugerencias de código en línea no sean útiles, prueba las siguientes prácticas recomendadas:

- Revise el contexto del guion. Por lo general, cuanto más contexto proporciones, mejores resultados obtendrás de Amazon Q.
- Si el contexto varía o es ruidoso, modularice el código. Agrupe funciones similares dentro del mismo script. Separe las diferentes clases y funciones según su objetivo.
- Escribe los nombres de tus clases y funciones de forma táctica para ayudar a Amazon Q a ofrecer mejores recomendaciones.

En este ejemplo, el desarrollador está escribiendo un script desde cero que carga archivos a Hadoop.

Input:

```
class MyFunction{  
  
    // create function to read data locally and upload to HDFS filesystem
```

Resultado:

```
class MyFunction{  
  
    // create function to read local data file path and upload to HDFS filesystem  
    public void readAndUploadToHDFS(String inputPath, String outputPath) throws  
    IOException {  
        // read data from local file  
        BufferedReader in = new BufferedReader(new FileReader(inputPath));  
  
    }  
}
```

Aunque Amazon Q hizo una recomendación, no cumplió con las expectativas del desarrollador. Para mejorar el rendimiento, el desarrollador importa las bibliotecas y cambia el nombre de la clase para representar las utilidades de Hadoop.

Solución:

```
import java.io.BufferedReader;  
import java.io.IOException;
```

```
import java.io.FileReader;

// import applicable hadoop libraries for IO operations
import org.apache.hadoop.fs.Path;

class HadoopUtilities{

    // create function to read local data file path and upload to HDFS filesystem
}
```

Resultado:

En este resultado, el desarrollador aceptó varias sugerencias de forma secuencial.

```
class HadoopUtilities{

    // create function to read local data file path and upload to HDFS filesystem
    public static void uploadToHDFS(String localFilePath, String hdfsFilePath) throws
IOException {
        // create HDFS file path object
        Path hdfsPath = new Path(hdfsFilePath);
        // create local file path object
        Path localPath = new Path(localFilePath);
        // create hadoop file system object
        org.apache.hadoop.fs.FileSystem hdfs = org.apache.hadoop.fs.FileSystem.get(new
org.apache.hadoop.conf.Configuration());
        // upload local file to HDFS
        hdfs.copyFromLocalFile(localPath, hdfsPath);
        // close HDFS file system
        hdfs.close();
        System.out.println("File uploaded successfully!");
    }
}
```

En el mismo HadoopUtilities script, el desarrollador añade un mensaje de comentarios para leer un tema del Amazon Simple Queue Service (Amazon SQS).

Input:

```
// read message from SQS
```

Resultado:

```
// read message from SQS
public static void readMessageFromSQS(String message) {
    System.out.println("Received message from SQS: " + message);
}
```

En lugar de utilizar métodos Java para leer un mensaje SQS de Amazon SQS, Amazon Q sugiere imprimir un método de cadena. Sin embargo, esta sugerencia no satisface el objetivo original del desarrollador.

En este caso, es preferible volver a modularizar el código. El guion se centra en los métodos de Hadoop. Por lo tanto, podría ser mejor crear un script de utilidad específico para las operaciones de SQS.

La creación de una clase independiente para las utilidades de SQS mejora la sugerencia en línea, que se puede utilizar en HadoopUtilities guión.

Input:

```
// import function for SQS
import com.amazonaws.services.sqs.AmazonSQS;
import java.util.ArrayList;
import java.util.List;

class SQSUtills {

    // instantiate SQS client
    private AmazonSQS sqsClient;

    // load SQS object
    public SQSUtills(AmazonSQS sqsClient) {
        this.sqsClient = sqsClient;
    }

    // poll message from Amazon SQS given queue name
}
```

Resultado:

```
// poll message from Amazon SQS given queue name
public List<String> pollMessages(String queueName) {
    List<String> messages = new ArrayList<>();
}
```

```
// poll messages from queue
receiveMessageResult = sqsClient.receiveMessage(new
ReceiveMessageRequest(queueName));
for (Message message : receiveMessageResult.getMessages()) {
    messages.add(message.getBody());
}
return messages;
```

Si la sugerencia en línea sigue sin dar los resultados esperados, pídele a Amazon Q Developer en una sesión de chat que genere código.

Hable sobre la generación de código

Usuario:

Generar una función java para leer un mensaje de una cola de Amazon SQS

Amazon Q:

A continuación, se muestra un ejemplo de una función Java para leer un mensaje de una cola de Amazon SQS:

```
import com.amazonaws.services.sqs.AmazonSQS;
import com.amazonaws.services.sqs.model.Message;
import com.amazonaws.services.sqs.model.ReceiveMessageRequest;
import com.amazonaws.services.sqs.model.ReceiveMessageResult;
public class SQSMessageReader {
    private AmazonSQS sqsClient;
    private String queueUrl;

    public SQSMessageReader(AmazonSQS sqsClient, String queueUrl) {
        this.sqsClient = sqsClient;
        this.queueUrl = queueUrl;
    }

    public String readMessage() {
        ReceiveMessageRequest receiveMessageRequest = new
ReceiveMessageRequest(queueUrl)
            .withMaxNumberOfMessages(1)
            .withWaitTimeSeconds(10);

        ReceiveMessageResult receiveMessageResult =
sqsClient.receiveMessage(receiveMessageRequest);
```

```
    if (receiveMessageResult.getMessages().size() > 0) {
        Message message = receiveMessageResult.getMessages().get(0);
        String messageBody = message.getBody();
        String receiptHandle = message.getReceiptHandle();

        // Process the message
        System.out.println("Received message: " + messageBody);

        // Delete the message from the queue
        sqsClient.deleteMessage(queueUrl, receiptHandle);

        return messageBody;
    } else {
        return null;
    }
}
```

Este código crea una clase SQSMessage Reader que toma un cliente de AmazonSQS y una URL de cola como parámetros del constructor. El método readMessage () lo utiliza ReceiveMessageRequest para recuperar hasta 1 mensaje de la cola, con un tiempo de espera máximo de 10 segundos.

Si se recibe un mensaje, se imprime el cuerpo del mensaje y se elimina de la cola mediante el DeleteMessageRequest A continuación, se devuelve el cuerpo del mensaje. Si no se recibe ningún mensaje, se devuelve un valor nulo.

Finaliza el chat de Amazon Q sobre la generación de código

Resultados inadecuados de los chats

Mientras desarrollan el código, los desarrolladores pueden pedir a Amazon Q que genere funciones comunes, proporcione recomendaciones o explique el código. A veces, ante un aviso (como una pregunta o un fragmento de código), Amazon Q puede proporcionar comentarios generales o resultados que no cumplen tus expectativas. En estos casos, intenta lo siguiente:

- Experimente con diferentes indicaciones para mejorar los resultados de generación de código que obtiene de Amazon Q.
- Siga URLs lo que Amazon Q proporciona con frecuencia en sus respuestas a sus solicitudes o preguntas.
- Envía comentarios para ayudar a mejorar el rendimiento futuro de Amazon Q.

Por ejemplo, un desarrollador está creando una solución de generación aumentada de recuperación (RAG) y necesita segmentar su documento teniendo en cuenta las relaciones semánticas entre las oraciones. El desarrollador conversa con Amazon Q de la siguiente manera.

Hable sobre una solución RAG

Usuario:

Para una solución RAG, genere código para la fragmentación semántica

Amazon Q:

Sin duda, he aquí un ejemplo de cómo se podría implementar la fragmentación semántica para una solución de generación aumentada de recuperación (RAG) en Python usando la biblioteca: LangChain

```
from langchain.text_splitter import CharacterTextSplitter, TokenTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Chroma

# Initialize the text splitter
text_splitter = TokenTextSplitter(
    chunk_size=500,
    chunk_overlap=50,
    length_function=len
)
# Initialize the embeddings model
embeddings = HuggingFaceEmbeddings()

# Create the vector store
vectorstore = Chroma(
    embedding_function=embeddings.embed_documents,
    persist_directory="./chroma-data"
)

# Chunk the text using semantic similarity
def semantic_chunk(text):
    chunks = text_splitter.split_text(text)
    embeddings = embeddings.embed_documents(chunks)
    vectorstore.add_texts(chunks, embeddings)
    return chunks
```

En este ejemplo, estamos usando el `TokenTextSplitter` de `LangChain` para dividir el texto de entrada en bloques de 500 caracteres con una superposición de 50 caracteres. La `HuggingFaceEmbeddings` el modelo se utiliza para generar incrustaciones para cada fragmento, y el `Chroma` El almacén de vectores se utiliza para almacenar los fragmentos y sus incrustaciones.

La función `semantic_chunk` recoge el texto de entrada, lo divide en fragmentos mediante el `text_splitter`, genera incrustaciones para cada fragmento mediante el modelo de incrustaciones y, a continuación, añade los fragmentos y sus incrustaciones al `vectorstore`.

Este enfoque le permite dividir el texto de entrada en fragmentos semánticamente significativos, lo que puede ser útil para una variedad de tareas, como responder preguntas o resumir, en una solución RAG.

Ten en cuenta que este es un ejemplo básico y es posible que tengas que ajustar los parámetros (p. ej., `chunk_size`, `chunk_overlap`) o la elección del divisor de texto y el modelo de incrustaciones en función de tu caso de uso y datos específicos.

Fin de la charla de Amazon Q sobre una solución RAG

Sin embargo, este enfoque solo proporciona una forma básica de fragmentar documentos mediante la técnica de ventanas deslizantes. El desarrollador quiere una forma avanzada que se base en la medición de las incrustaciones. En estos escenarios, el desarrollador debe añadir más detalles al mensaje en función de su objetivo. Cuando el desarrollador hace referencia a la conocida función divisora de A121 texto con `LangChain`, recibe una respuesta mejorada, tal y como se muestra en el siguiente chat.

Hable sobre una solución RAG (con un mensaje mejorado)

Usuario:

La pregunta anterior solo se refiere a la división en función del tamaño del token por fragmento. Quiero usar un divisor de texto A121 semántico. Genera una función mediante la división de texto semántico A121 con `langchain`.

Amazon Q:

Bien, usemos el divisor de texto A121 semántico `LangChain` para implementar la fragmentación semántica. He aquí un ejemplo: [1]

```
from langchain.text_splitter import AI21SemanticTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
```

```
from langchain.vectorstores import Chroma

# Initialize the text splitter
text_splitter = AI21SemanticTextSplitter()

# Initialize the embeddings model
embeddings = HuggingFaceEmbeddings()

# Create the vector store
vectorstore = Chroma(
    embedding_function=embeddings.embed_documents,
    persist_directory="./chroma-data"
)

# Chunk the text using semantic similarity
def semantic_chunk(text):
    chunks = text_splitter.split_text(text)
    embeddings = embeddings.embed_documents(chunks)
    vectorstore.add_texts(chunks, embeddings)
    return chunks
```

En este ejemplo, utilizamos el formulario LangChain para dividir el texto AI21 SemanticTextSplitter de entrada en fragmentos semánticamente significativos. La HuggingFaceEmbeddings el modelo se usa para crear incrustaciones para cada fragmento, y el Chroma El almacén de vectores se utiliza para almacenar los fragmentos y sus incrustaciones.

La función semantic_chunk recoge el texto de entrada, lo divide en trozos con el text_splitter, crea incrustaciones para cada fragmento mediante el modelo de incrustaciones y, a continuación, añade los fragmentos y sus incrustaciones al vectorstore.

Este enfoque le permite dividir el texto de entrada en fragmentos semánticamente significativos, lo que puede ser útil para una variedad de tareas, como responder preguntas o resumir, en una solución RAG.

Tenga en cuenta que es posible que deba ajustar los parámetros o la elección del modelo de incrustaciones en función de su caso de uso y sus datos específicos.

Fuentes [1] Aprenda a utilizar los divisores de texto en LangChain

[soosweb3.hashnode.dev > -text-splitters the-ultimate-langchain-series](https://soosweb3.hashnode.dev/text-splitters-the-ultimate-langchain-series)

Fin de la charla de Amazon Q sobre una solución RAG (con un mensaje mejorado)

Según los resultados del chat, el desarrollador puede usar la función y navegar hasta la fuente de la información.

En general, los temas y preguntas avanzados requieren un contexto más específico en el mensaje cuando conversas con un desarrollador de Amazon Q. Si crees que los resultados de tu chat no son precisos, usa el icono del pulgar hacia abajo para enviar comentarios sobre la respuesta de Amazon Q. El desarrollador de Amazon Q utiliza continuamente los comentarios para mejorar las versiones futuras. En el caso de las interacciones que produjeron resultados positivos, es útil enviar sus comentarios mediante el icono del pulgar hacia arriba.

FAQs acerca de Amazon Q Developer

En esta sección, se brindan respuestas a las preguntas frecuentes sobre el uso de Amazon Q Developer para el desarrollo de código.

¿Qué es Amazon Q Developer?

Amazon Q Developer es un potente servicio generativo basado en inteligencia artificial diseñado para acelerar las tareas de desarrollo de código al proporcionar recomendaciones y generación de código inteligentes. El 30 de abril de 2024, Amazon CodeWhisperer pasó a formar parte de Amazon Q Developer.

¿Cómo puedo acceder a Amazon Q Developer?

Amazon Q Developer está disponible como parte de los AWS kits de herramientas para Visual Studio Code y JetBrains IDEs, como IntelliJ y PyCharm. Para empezar, [instale la versión más reciente AWS Toolkit](#).

¿Qué lenguajes de programación admite Amazon Q Developer?

Para Visual Studio Code y JetBrains IDEs, Amazon Q Developer admite Python, Java, JavaScript, TypeScript, C#, Go, Rust, PHP, Ruby, Kotlin, C, C++/SQL, script de Shell y Scala. Aunque esta guía se centra en Python y Java por ejemplo, los conceptos son aplicables a cualquier lenguaje de programación compatible.

¿Cómo puedo proporcionar contexto a Amazon Q Developer para una mejor generación de código?

Comience con el código existente, importe las bibliotecas pertinentes, cree clases y funciones o establezca esqueletos de código. Usa bloques de comentarios estándar para las indicaciones en lenguaje natural. Mantén tu guion centrado en objetivos específicos y modulariza las distintas funcionalidades en guiones separados con un contexto relevante. Para obtener más información, consulte [Prácticas recomendadas de codificación con Amazon Q Developer](#).

¿Qué debo hacer si la generación de código en línea con Amazon Q Developer no es precisa?

Revise el contexto del script, asegúrese de que las bibliotecas estén presentes y asegúrese de que las clases y funciones estén relacionadas con el nuevo código. Modulariza tu código y separa las diferentes clases y funciones según su objetivo. Escribe indicaciones o comentarios claros y específicos. Si aún no estás seguro de la precisión del código y no puedes continuar con él, inicia una conversación con Amazon Q y envíale el fragmento de código con las instrucciones. Para obtener más información, consulte [Solución de problemas de generación de código en Amazon Q Developer](#).

¿Cómo puedo utilizar la función de chat para desarrolladores de Amazon Q para la generación de código y la solución de problemas?

Chatea con Amazon Q para generar funciones comunes, solicitar recomendaciones o explicar el código. Si la respuesta inicial no es satisfactoria, experimenta con diferentes indicaciones y sigue las que se proporcionan URLs. Además, envía comentarios a Amazon Q para mejorar su rendimiento de chat en el futuro. Usa los íconos del pulgar hacia arriba y el pulgar hacia abajo para enviar tus comentarios. [Para obtener más información, consulta los ejemplos de chat](#).

¿Cuáles son algunas de las prácticas recomendadas para utilizar Amazon Q Developer?

Proporcione el contexto pertinente, experimente e intente seguir las indicaciones, revise las sugerencias de código antes de aceptarlas, utilice las funciones de personalización y comprenda las políticas de privacidad de datos y uso del contenido. Para obtener más información, consulte [Prácticas recomendadas para la generación de código con Amazon Q Developer](#) y [Prácticas recomendadas de código con Amazon Q Developer](#).

¿Puedo personalizar Amazon Q Developer para que genere recomendaciones basadas en mi propio código?

Sí, utilice personalizaciones, que es una función avanzada de Amazon Q Developer. Con las personalizaciones, las empresas pueden proporcionar sus propios repositorios de código para que Amazon Q Developer pueda recomendar sugerencias de código integradas. Para obtener más información, consulte [Capacidades avanzadas de Amazon Q Developer](#) and [Resources](#).

Próximos pasos para usar Amazon Q Developer

Con los conocimientos adquiridos en esta completa guía, puede utilizar Amazon Q Developer en su flujo de trabajo de codificación de forma eficaz. Instálelo AWS Toolkit en el IDE que prefiera ([Visual Studio Code](#) o [JetBrains](#)) y comience a explorar la generación de código generativo mediante IA y las recomendaciones de Amazon Q Developer.

La forma más eficaz de aprovechar todo el potencial de Amazon Q Developer es mediante la experiencia práctica con tu propio código. A medida que integre Amazon Q en su ciclo de vida de desarrollo, consulte esta guía para conocer las mejores prácticas, la solución de problemas y ejemplos del mundo real.

Además, manténgase informado consultando los AWS blogs y las guías para desarrolladores a los que se hace referencia en [Recursos](#). Estos recursos proporcionan las últimas actualizaciones, prácticas recomendadas e información para ayudarle a optimizar el uso de Amazon Q Developer.

Sus comentarios son inestimables para mejorar esta guía y ayudar a que siga siendo un recurso valioso para los desarrolladores. Comparta sus experiencias, desafíos y sugerencias para futuras versiones. Sus comentarios ayudarán a mejorar la guía con ejemplos adicionales, escenarios de solución de problemas e información adaptada a sus necesidades.

Recursos

AWS blogs

- [Accelerate your Software Development Lifecycle with Amazon Q](#)
- [Reimaginando el desarrollo de software con el Amazon Q Developer Agent](#)
- [Cinco ejemplos de solución de problemas con Amazon Q](#)
- [Personaliza Amazon Q Developer en tu IDE sitio con tu base de código privada](#)
- [Tres formas en las que el agente Amazon Q Developer para la transformación del código acelera las actualizaciones de Java](#)
- [Aprovechar Amazon Q Developer para una depuración y un mantenimiento eficientes del código](#)
- [Probar sus aplicaciones con Amazon Q Developer](#)

AWS documentación

- [Guía del usuario para desarrolladores de Amazon Q](#)
- [Personalización del código de Amazon Q Developer](#)
- [Transformación del código de Amazon Q Developer](#)

AWS talleres

- [Día de inmersión para desarrolladores de Amazon Q](#)
- [Taller para desarrolladores de Amazon Q: Creación de la aplicación Q-Words](#)
- [Taller para desarrolladores de Amazon Q: Creación de mensajes eficaces](#)

Colaboradores

Las siguientes personas contribuyeron a esta guía:

- Joe King, científico de datos sénior, AWS
- Prateek Gupta, jefe de equipo — Sr. CAA, AWS
- Manohar Reddy Arranagu, arquitecto, DevOps AWS
- Soumik Roy, arquitecto de aplicaciones en la nube, AWS
- Sanket Shinde, consultor, AWS

Historial del documento

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones de futuras [RSSactualizaciones](#).

Cambio	Descripción	Fecha
Publicación inicial	—	16 de agosto de 2024

AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. 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 EC2 instancia 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. Algunos ejemplos de funciones agregadas 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.

antipatronos

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 AIOps se utiliza en la estrategia de AWS migración, 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 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

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.

branch

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 está seguro, despliega la nueva versión y reemplaza la versión actual en su totalidad.

CCoE

Consulte [Cloud Center of Excellence](#).

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: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCoE, establecer un modelo de operaciones)

- Migración: migración de aplicaciones individuales
- Reinención: 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 en el](#) 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 Bitbucket Cloud. 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, Amazon SageMaker AI 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, puesta en escena y producción del proceso de publicación del software. CI/CD is commonly described as a pipeline. 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

Vea la [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 una administración y un 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](#) 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 deriva

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](#).

EDI

Véase [intercambio electrónico 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.

intercambio electrónico de datos (EDI)

El intercambio automatizado de documentos comerciales entre organizaciones. Para obtener más información, consulte [Qué es el intercambio electrónico de datos](#).

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).

entorno

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](#).

ERP

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.

indicaciones de unos pocos pasos

Proporcionar a un [LLM](#) un pequeño número de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que realice una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, en el que los modelos aprenden a partir de ejemplos (planos) integrados en las instrucciones. Las indicaciones con pocas tomas pueden ser eficaces para tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. [Consulte también el apartado de mensajes sin intervención](#).

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.

FM

Consulte el [modelo básico](#).

modelo de base (FM)

Una gran red neuronal de aprendizaje profundo que se ha estado entrenando con conjuntos de datos masivos de datos generalizados y sin etiquetar. FMs son capaces de realizar una amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes y conversar en lenguaje natural. Para obtener más información, consulte [Qué son los modelos básicos](#).

G

IA generativa

Un subconjunto de modelos de [IA](#) que se han entrenado con grandes cantidades de datos y que pueden utilizar un simple mensaje de texto para crear contenido y artefactos nuevos, como imágenes, vídeos, texto y audio. Para obtener más información, consulte [Qué es la IA generativa](#).

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.

imagen dorada

Instantánea de un sistema o software que se utiliza como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

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 el cumplimiento en todas las unidades organizativas (OUs). 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

HA

Consulte la [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.

datos retenidos

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de aprendizaje [automático](#). Puede utilizar los datos de reserva para evaluar el rendimiento del modelo comparando las predicciones del modelo con los datos de reserva.

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 versiones.

DevOps

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, aplicar parches 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 AWS de referencia de seguridad](#) recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación y el resto de Internet.

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.

I

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 de la Internet de las cosas \(IIoT\) industrial](#).

VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red VPCs entre Internet y las redes locales (en una misma o Regiones de AWS diferente). La [arquitectura AWS de referencia de seguridad](#) recomienda configurar su cuenta de red con entrada, salida e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

Internet de las cosas (IIoT)

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 IIoT?](#).

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 obtener más información, consulte Interpretabilidad del [modelo de aprendizaje automático](#) 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](#).

modelo de lenguaje grande (LLM)

Un modelo de [IA](#) de aprendizaje profundo que se entrena previamente con una gran cantidad de datos. Un LLM puede realizar múltiples tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. [Para obtener más información, consulte Qué son. LLMs](#)

migración grande

Migración de 300 servidores o más.

LBAC

Consulte 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](#).

LLM

Véase un modelo de lenguaje [amplio](#).

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 una red bien definida APIs 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 AWS servicios sin servidor](#).

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 un uso ligero. APIs 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: realoje la migración a Amazon EC2 con AWS 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.

ORR

Consulte la revisión de [la preparación operativa](#).

OT

Consulte la [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 AWS de referencia de seguridad](#) recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la 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](#).

policy

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 desarrollo.

zonas alojadas privadas

Un contenedor que contiene información sobre cómo desea que Amazon Route 53 responda a las consultas de DNS de un dominio y sus subdominios dentro de uno o más VPCs. 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.

encadenamiento rápido

Utilizar la salida de una solicitud de [LLM](#) como entrada para la siguiente solicitud para generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en subtareas o para refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

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.

publish/subscribe (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 \(RACI\)](#).

RAG

Consulte [Retrieval Augmented Generation](#).

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](#).

réplica de lectura

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.](#)

resiliencia

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](#).

Generación aumentada de recuperación (RAG)

Tecnología de [inteligencia artificial generativa](#) en la que un máster [hace referencia](#) a una fuente de datos autorizada que se encuentra fuera de sus fuentes de datos de formación antes de generar una respuesta. Por ejemplo, un modelo RAG podría realizar una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para obtener más información, consulte [Qué es](#) el RAG.

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.

seguridad desde el diseño

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

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. Existen cuatro tipos principales de controles de seguridad: [preventivos](#), [de detección](#), con [capacidad](#) 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 EC2 instancia de Amazon 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)

Política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. SCPs defina barreras o establezca límites a las acciones que un administrador puede delegar en usuarios o roles. Puede utilizarlas SCPs como listas de permitidos o rechazados para especificar qué servicios o acciones están 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](#).

SLO

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](#)

SPOF

Consulte el [punto único de falla](#).

esquema en forma de estrella

Estructura organizativa de una base de datos que utiliza una tabla de hechos grande para almacenar datos medidos o transaccionales 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 dismantelar 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.

indicador del sistema

Una técnica para proporcionar contexto, instrucciones o pautas a un [LLM](#) para dirigir su comportamiento. Las indicaciones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

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

Un centro de tránsito de red que puede usar para interconectar sus VPCs redes con las locales. 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

Una conexión entre dos VPCs que le 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 AWS de calificación de la carga](#) de trabajo.

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.

aviso de tiro cero

Proporcionar a un [LLM](#) instrucciones para realizar una tarea, pero sin ejemplos (imágenes) que puedan ayudar a guiarla. El LLM debe utilizar sus conocimientos previamente entrenados para realizar la tarea. La eficacia de las indicaciones cero depende de la complejidad de la tarea y de la calidad de las indicaciones. [Consulte también las indicaciones de pocos pasos](#).

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 versión original de inglés, prevalecerá la versión en inglés.