



Best practice con Amazon Q Developer per la generazione di codice in linea e assistita

AWS Guida prescrittiva



AWS Guida prescrittiva: Best practice con Amazon Q Developer per la generazione di codice in linea e assistita

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Introduzione	1
Obiettivi	1
Flussi di lavoro per gli sviluppatori	3
Progettazione e pianificazione	4
Codifica	4
Revisione del codice	5
Integrazione e implementazione	5
Funzionalità avanzate	6
Trasformazione del codice Amazon Q Developer	6
Personalizzazioni di Amazon Q Developer	6
Le migliori pratiche di codifica	8
Le migliori pratiche per l'onboarding	8
Prerequisiti per Amazon Q Developer	8
Procedure consigliate per l'utilizzo di Amazon Q Developer	8
Privacy dei dati e utilizzo dei contenuti in Amazon Q Developer	9
Le migliori pratiche per la generazione di codice	9
Le migliori pratiche per i consigli sul codice	10
Esempi di codice	12
Python esempi	12
Genera classi e funzioni	12
Codice del documento	14
Genera algoritmi	15
Genera test unitari	18
Java esempi	18
Genera classi e funzioni	19
Codice del documento	21
Genera algoritmi	24
Genera test unitari	26
Esempi di chat	27
Chiedi informazioni Servizi AWS	27
Genera codice	29
Genera test unitari	30
Spiega il codice	32
Risoluzione dei problemi	36

Generazione di codice vuoto	36
Commenti continui	37
Generazione errata del codice in linea	38
Risultati inadeguati delle chat	43
FAQs	48
Cos'è Amazon Q Developer?	48
Come posso accedere ad Amazon Q Developer?	48
Quali linguaggi di programmazione supporta Amazon Q Developer?	48
Come posso fornire un contesto ad Amazon Q Developer per una migliore generazione di codice?	48
Cosa devo fare se la generazione di codice in linea con Amazon Q Developer non è accurata?	49
Come posso utilizzare la funzionalità di chat di Amazon Q Developer per la generazione di codice e la risoluzione dei problemi?	49
Quali sono alcune best practice per l'utilizzo di Amazon Q Developer?	49
Posso personalizzare Amazon Q Developer per generare consigli basati sul mio codice?	49
Passaggi successivi	51
Risorse	52
AWS blog	52
AWS documentazione	52
AWS workshop	52
Collaboratori	53
Cronologia dei documenti	54
Glossario	55
#	55
A	56
B	59
C	61
D	64
E	68
F	70
G	72
H	73
I	75
L	77
M	78

O	83
P	85
Q	88
R	89
S	92
T	96
U	97
V	98
W	98
Z	100
.....	ci

Best practice con Amazon Q Developer per la generazione di codice in linea e assistita

Amazon Web Services ([Collaboratori](#))

agosto 2024 () [Cronologia dei documenti](#)

Tradizionalmente, gli sviluppatori si affidavano alla propria esperienza, alla propria documentazione e a frammenti di codice provenienti da varie fonti per scrivere e gestire il codice. Sebbene questi metodi abbiano dato buoni risultati al settore, possono richiedere molto tempo e sono soggetti a errori umani, con conseguenti inefficienze e potenziali bug.

È qui che interviene Amazon Q Developer per migliorare il percorso dello sviluppatore. Amazon Q Developer è un potente assistente AWS generativo basato sull'intelligenza artificiale progettato per accelerare le attività di sviluppo del codice fornendo generazione di codice e consigli intelligenti.

Tuttavia, come per qualsiasi nuova tecnologia, possono esserci delle sfide. Aspettative irrealistiche, difficoltà di onboarding, risoluzione di problemi di generazioni di codice imprecise e utilizzo corretto delle funzionalità di Amazon Q sono ostacoli comuni che gli sviluppatori potrebbero incontrare. Questa guida completa affronta queste sfide, fornendo scenari reali, best practice dettagliate, risoluzione dei problemi ed esempi di codice pratici specifici per Python e Java, due dei linguaggi di programmazione più diffusi.

Questa guida si concentra sull'utilizzo di Amazon Q Developer per eseguire attività di sviluppo di codice come:

- Completamento del codice: genera suggerimenti in linea mentre gli sviluppatori programmano in tempo reale.
- Miglioramenti e consigli sul codice: discuti dello sviluppo del software, genera nuovo codice con linguaggio naturale e migliora il codice esistente.

Obiettivi

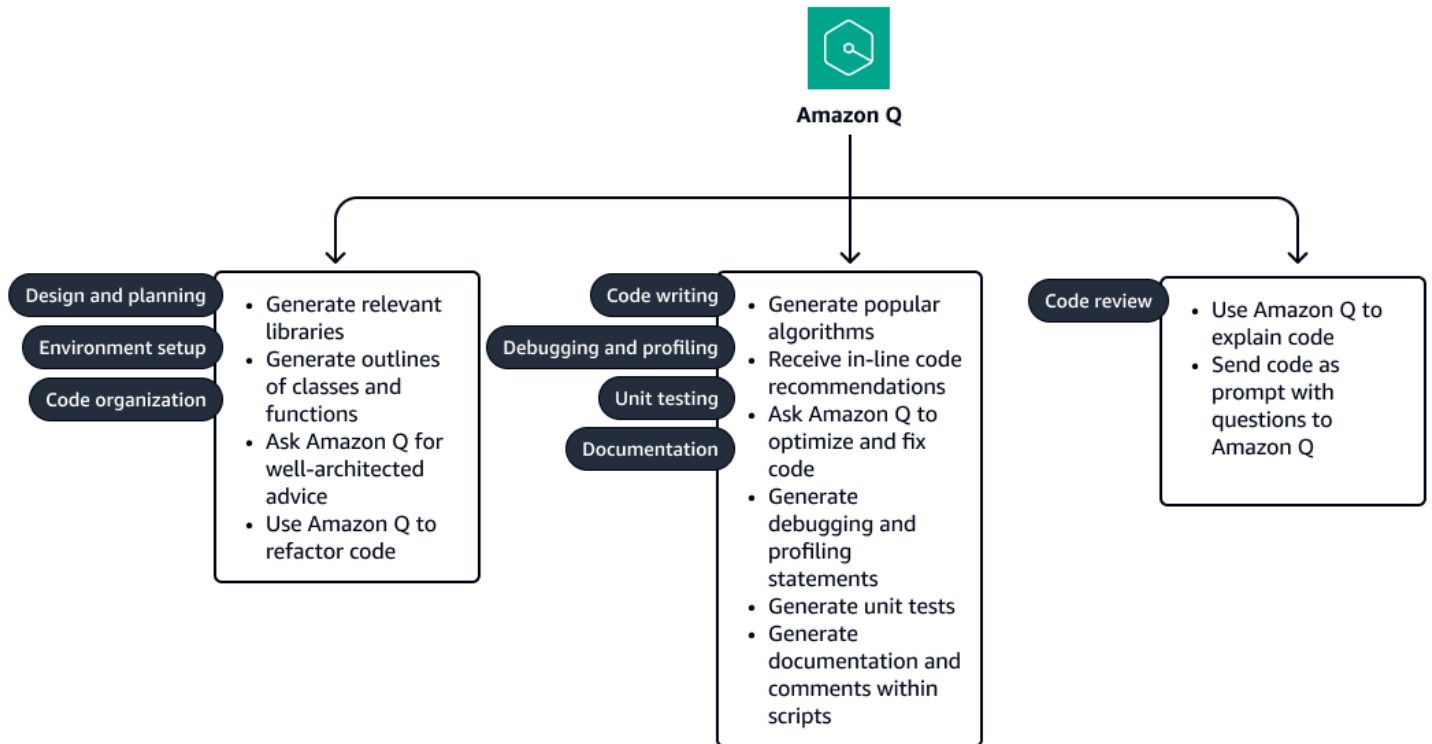
L'obiettivo di questa guida è supportare gli sviluppatori che sono utenti nuovi o continui di Amazon Q Developer, aiutandoli a utilizzare il servizio con successo nelle loro attività quotidiane di programmazione. Anche i responsabili dei team di sviluppo possono trarre vantaggio dalla lettura di questa guida.

Questa guida fornisce le seguenti informazioni sull'uso di Amazon Q Developer:

- Comprendi l'uso efficace di Amazon Q Developer per lo sviluppo di codice
 - Fornisci le migliori pratiche per integrare Amazon Q Developer nel [flusso di lavoro di uno sviluppatore](#).
 - Offri step-by-step una guida con esempi per [generare codice](#) con successo e [consigli](#).
- Mitiga le sfide comuni e promuovi la chiarezza degli sviluppatori per l'utilizzo di Amazon Q Developer
 - Offri [strategie](#) e approfondimenti per soddisfare le aspettative degli sviluppatori e superare gli ostacoli legati alla precisione e alle prestazioni della generazione del codice.
- Fornisci la risoluzione dei problemi e la gestione degli errori
 - Fornisci agli sviluppatori [linee guida per la risoluzione dei problemi](#) di generazione di codice Amazon Q Developer per risolvere problemi di risultati imprecisi o comportamenti imprevisti.
 - Fornisci [esempi e scenari reali specifici per e](#). Python Java
- Ottimizza i flussi di lavoro e la produttività
 - Ottimizza i flussi di lavoro di sviluppo del codice con Amazon Q Developer.
 - Discuti le strategie per migliorare la [produttività degli sviluppatori](#).

Utilizzo di Amazon Q Developer nei flussi di lavoro degli sviluppatori

Gli sviluppatori seguono un flusso di lavoro standard che comprende le fasi di raccolta dei requisiti, progettazione e pianificazione, codifica, test, revisione del codice e implementazione. Questa sezione si concentra su come utilizzare le funzionalità di Amazon Q Developer per ottimizzare le fasi di sviluppo chiave.



Il diagramma precedente mostra come Amazon Q Developer può accelerare e semplificare le seguenti attività comuni nelle fasi di sviluppo del codice:

- Progettazione e pianificazione | Configurazione dell'ambiente | Organizzazione del codice
 - Generazione di librerie pertinenti
 - Genera schemi di classi e funzioni
 - Chiedi ad Amazon Q una consulenza ben progettata
 - Utilizza Amazon Q per rifattorizzare il codice
- Scrittura di codice | Debug e profilazione | Unit testing | Documentazione
 - Genera algoritmi popolari

- Ricevere informazioni di codice in linea
- Chiedi ad Amazon Q di ottimizzare e correggere il codice
- Genera istruzioni di debug e profilazione
- Genera test unitari
- Genera documentazione e commenti all'interno degli script
- Revisione del codice
 - Chiedi ad Amazon Q di spiegare il codice
 - Invia il codice non appena richiesto con domande ad Amazon Q

Progettazione e pianificazione

Dopo aver raccolto i requisiti aziendali e tecnici, gli sviluppatori progettano nuove basi di codice o estendono quelle esistenti. Durante questa fase, Amazon Q Developer può aiutare gli sviluppatori a svolgere le seguenti attività:

- Genera librerie e schemi di classi e funzioni pertinenti per consigli ben strutturati.
- Fornisci indicazioni per domande di ingegneria, compatibilità e progettazione architettonica.

Codifica

Il processo di codifica utilizza Amazon Q Developer per accelerare lo sviluppo nei seguenti modi:

- Configurazione dell'ambiente: installalo Kit di strumenti AWS nel tuo ambiente di sviluppo integrato (IDE) (ad esempio, VS Code o IntelliJ). Quindi, usa Amazon Q per generare librerie o ricevere suggerimenti di configurazione in base agli obiettivi del tuo progetto. Per ulteriori dettagli, consulta [Best practice per l'onboarding di Amazon Q Developer](#).
- Organizzazione del codice: rifattorizza il codice o ottieni consigli sull'organizzazione da Amazon Q in linea con gli obiettivi del tuo progetto.
- Scrittura di codice: utilizza i suggerimenti in linea per generare codice durante lo sviluppo o chiedi ad Amazon Q di generare codice utilizzando il pannello di chat di Amazon Q nel tuo IDE. Per ulteriori dettagli, consulta [Best practice per la generazione di codice con Amazon Q Developer](#).
- Debug e profilazione: genera comandi di profilazione o utilizza opzioni di Amazon Q come Fix and Explain per eseguire il debug dei problemi.

- **Test unitari:** fornisci il codice come richiesta ad Amazon Q durante una sessione di chat e richiedi la generazione di unit test applicabile. Per ulteriori informazioni, consulta [Esempi di codice con Amazon Q Developer](#).
- **Documentazione:** usa i suggerimenti in linea per creare commenti e docstring oppure usa l'opzione Explain per generare riepiloghi dettagliati per le selezioni di codice. Per ulteriori informazioni, consulta [Esempi di codice con Amazon Q Developer](#).

Revisione del codice

I revisori devono comprendere il codice di sviluppo prima di promuoverlo alla produzione. Per accelerare questo processo, utilizza le opzioni Amazon Q Explain and Optimize o invia selezioni di codice con istruzioni personalizzate ad Amazon Q in una sessione di chat. Per ulteriori informazioni, consulta Esempi di [chat](#).

Integrazione e implementazione

Chiedi ad Amazon Q indicazioni sull'integrazione continua, sulle pipeline di distribuzione e sulle migliori pratiche di implementazione specifiche per l'architettura del tuo progetto.

Utilizzando questi consigli, puoi imparare a sfruttare efficacemente le funzionalità di Amazon Q Developer, ottimizzando i flussi di lavoro e aumentando la produttività durante l'intero ciclo di vita dello sviluppo.

Funzionalità avanzate di Amazon Q Developer

Sebbene questa guida si concentri sull'uso di Amazon Q Developer in attività pratiche di programmazione, è importante conoscere le sue seguenti funzionalità avanzate:

- Trasformazione del codice Amazon Q Developer
- Personalizzazioni di Amazon Q Developer

Trasformazione del codice Amazon Q Developer

Amazon Q Developer Agent per la trasformazione del codice può aggiornare la versione in linguaggio di codice dei tuoi file senza dover riscrivere il codice manualmente. Funziona analizzando i file di codice esistenti e riscrivendoli automaticamente per utilizzare una versione più recente del linguaggio. Ad esempio, Amazon Q trasforma un singolo modulo se lavori in un IDE come Eclipse. Se utilizzi Visual Studio Code, Amazon Q può trasformare un intero progetto o spazio di lavoro.

Usa Amazon Q quando desideri eseguire attività comuni di aggiornamento del codice come le seguenti:

- Aggiorna il codice per adattarlo alla nuova sintassi della versione linguistica.
- Esegui test unitari per convalidare la corretta compilazione ed esecuzione.
- Verifica e risolvi i problemi di distribuzione.

Amazon Q può far risparmiare agli sviluppatori giorni o mesi di lavoro noioso e ripetitivo per aggiornare le basi di codice.

A partire da giugno 2024, Amazon Q Developer supporta l'aggiornamento Java codice e può trasformare Java Codice 8 per versioni più recenti come Java 11 o 17.

Personalizzazioni di Amazon Q Developer

Grazie alla sua capacità di personalizzazione, Amazon Q Developer può fornire suggerimenti in linea basati sulla base di codice di un'azienda. L'azienda fornisce il proprio repository di codice ad Amazon Simple Storage Service (Amazon S3) o tramite AWS CodeConnections, precedentemente noto come Connections. AWS CodeStar Quindi, Amazon Q utilizza l'archivio di

codice personalizzato con sicurezza abilitata per consigliare modelli di codifica pertinenti per gli sviluppatori di quell'organizzazione.

Quando utilizzi le personalizzazioni di Amazon Q Developer, tieni presente quanto segue:

- A partire da giugno 2024, la funzionalità Amazon Q Developer Customizations è in modalità anteprema. Di conseguenza, la funzionalità potrebbe essere limitata in termini di disponibilità e supporto.
- I suggerimenti di codice in linea personalizzati saranno accurati solo in base alla qualità degli archivi di codice forniti. Ti consigliamo di esaminare un [punteggio di valutazione](#) per ogni personalizzazione che crei.
- Per ottimizzare le prestazioni, ti consigliamo di includere almeno 20 file di dati contenenti la lingua specificata in cui tutti i file di origine superano i 10 MB. Assicurati che il tuo repository sia composto da codice sorgente referibile e non da file di metadati (ad esempio file di configurazione, file di proprietà e file readme).

Utilizzando le personalizzazioni di Amazon Q Developer, puoi risparmiare tempo nei seguenti modi:

- Utilizza consigli basati sul codice proprietario della tua azienda.
- Aumentate la riutilizzabilità delle basi di codice esistenti.
- Crea modelli ripetibili generalizzati in tutta l'azienda.

Le migliori pratiche di codifica con Amazon Q Developer

Questa sezione illustra le best practice per la codifica con Amazon Q Developer. Le best practice includono le seguenti categorie:

- [Onboarding](#): metodi e considerazioni durante l'onboarding
- Generazione di [codice - Guida per utilizzare con successo la generazione](#) di codice
- [Raccomandazioni sul codice](#): tecniche per migliorare il codice

Le migliori pratiche per l'onboarding di Amazon Q Developer

Amazon Q Developer è un potente assistente generativo per la codifica AI disponibile tramite programmi popolari IDEs come Visual Studio Code e JetBrains. Questa sezione si concentra sulle best practice per l'accesso e l'onboarding di Amazon Q Developer nel tuo ambiente di sviluppo di codice.

Prerequisiti per Amazon Q Developer

Amazon Q Developer è disponibile come parte di AWS Toolkit for Visual Studio Code e AWS Toolkit for JetBrains (ad esempio, IntelliJ e PyCharm). Per Visual Studio Code e JetBrains IDEs, Amazon Q Developer supporta Python, Java, C#, JavaScript TypeScript, Go, Rust, PHP, Ruby, Kotlin, C, C++, Shell scripting, SQL e Scala.

Per istruzioni dettagliate sull'installazione sia Kit di strumenti AWS per Visual Studio Code che per un JetBrains IDE, consulta [Installazione dell'estensione o del plug-in Amazon Q Developer nel tuo IDE](#) nella Amazon Q Developer User Guide.

Procedure consigliate per l'utilizzo di Amazon Q Developer

Le best practice generali per l'utilizzo di Amazon Q Developer includono quanto segue:

- Fornisci il contesto pertinente per ottenere risposte più accurate, ad esempio linguaggi di programmazione, framework e strumenti in uso. Suddividi problemi complessi in componenti più piccoli.
- Sperimenta e ripeti i tuoi suggerimenti e le tue domande. La programmazione spesso implica provare approcci diversi.

- Esamina sempre i suggerimenti sul codice prima di accettarli e modificali se necessario per assicurarti che facciano esattamente ciò che intendevi.
- Usa la [funzionalità di personalizzazione](#) per far conoscere ad Amazon Q Developer le tue librerie interne APIs, le best practice e i modelli architettonici per consigli più pertinenti.

Privacy dei dati e utilizzo dei contenuti in Amazon Q Developer

Quando decidi di utilizzare Amazon Q Developer, devi capire come vengono utilizzati i tuoi dati e contenuti. Di seguito sono riportati i punti chiave:

- Per gli utenti di Amazon Q Developer Pro, il contenuto del codice non viene utilizzato per il miglioramento del servizio o la formazione dei modelli.
- Per gli utenti del piano gratuito di Amazon Q Developer, puoi scegliere di non utilizzare i tuoi contenuti per migliorare il servizio tramite impostazioni o AWS Organizations politiche IDE.
- I contenuti trasmessi sono crittografati e tutti i contenuti archiviati sono protetti con crittografia a riposo e controlli di accesso. Per ulteriori informazioni, consulta la sezione [Crittografia dei dati in Amazon Q Developer](#) User Guide nella Amazon Q Developer User Guide.

Best practice per la generazione di codice con Amazon Q Developer

Amazon Q Developer fornisce generazione automatica di codice, completamento automatico e suggerimenti di codice in linguaggio naturale. Di seguito sono riportate le best practice per l'utilizzo dell'assistenza di codifica in linea di Amazon Q Developer:

- Fornisci un contesto per contribuire a migliorare la precisione delle risposte

Inizia con il codice esistente, importa librerie, crea classi e funzioni o stabilisci scheletri di codice. Questo contesto contribuirà a migliorare in modo significativo la qualità della generazione del codice.

- Codice in modo naturale

Usa la generazione di codice Amazon Q Developer come un potente motore di completamento automatico. Codifica come fai normalmente e lascia che Amazon Q fornisca suggerimenti durante la digitazione o la pausa. Se la generazione di codice non è disponibile o hai un problema relativo al codice, avvia Amazon Q digitando Alt+C su un PC o Option+C su macOS. Per ulteriori

informazioni sulle azioni comuni che puoi intraprendere utilizzando i suggerimenti in linea, consulta Using [shortcut keys](#) nella Amazon Q Developer User Guide.

- Includi librerie di importazione pertinenti agli obiettivi del tuo script

Includi le librerie di importazione pertinenti per aiutare Amazon Q a comprendere il contesto e generare il codice di conseguenza. Puoi anche chiedere ad Amazon Q di suggerire dichiarazioni di importazione pertinenti.

- Mantieni un contesto chiaro e mirato

Mantieni lo script incentrato su obiettivi specifici e modularizza funzionalità distinte in script separati con un contesto pertinente. Evita un contesto rumoroso o confuso.

- Sperimenta con i prompt

Esplora diverse istruzioni per spingere Amazon Q a produrre risultati utili nella generazione di codice. Ad esempio, prova i seguenti approcci:

- Utilizza blocchi di commenti standard per i prompt in linguaggio naturale.
 - Crea scheletri con commenti per compilare classi e funzioni.
 - Sii specifico nelle tue istruzioni, fornendo dettagli anziché generalizzazioni.
- Chatta con Amazon Q Developer e chiedi assistenza

Se Amazon Q Developer non fornisce suggerimenti accurati, chatta con Amazon Q Developer nel tuo IDE. Può fornire frammenti di codice o classi e funzioni complete per dare il via al tuo contesto. Per ulteriori informazioni, consulta [Chatting with Amazon Q Developer sul codice](#) nella Amazon Q Developer User Guide.

Le migliori pratiche per i consigli sul codice con Amazon Q Developer

Amazon Q Developer può rispondere alle domande degli sviluppatori e valutare il codice per offrire consigli che vanno dalla generazione di codice e correzioni di bug alle indicazioni sull'uso del linguaggio naturale. Di seguito sono riportate le best practice per l'utilizzo della chat in Amazon Q:

- Genera codice da zero

Per nuovi progetti o quando hai bisogno di una funzione generale (ad esempio, copiare file da Amazon S3), chiedi ad Amazon Q Developer di generare esempi di codice utilizzando istruzioni in

linguaggio naturale. Amazon Q può fornire link correlati a risorse pubbliche per ulteriori convalide e indagini.

- Cerca conoscenze di programmazione e spiegazioni sugli errori

In caso di problemi di codifica o messaggi di errore, fornisci il blocco di codice (con un messaggio di errore, se applicabile) e la tua domanda come prompt ad Amazon Q Developer. Questo contesto aiuterà Amazon Q a fornire risposte accurate e pertinenti.

- Migliora il codice esistente

Per correggere errori noti o ottimizzare il codice (ad esempio, per ridurre la complessità), seleziona il blocco di codice pertinente e invialo ad Amazon Q Developer con la tua richiesta. Sii specifico con le tue istruzioni per ottenere risultati migliori.

- Spiega la funzionalità del codice

Quando esplori nuovi repository di codice, seleziona un blocco di codice o un intero script e invialo ad Amazon Q Developer per una spiegazione. Riduci la dimensione della selezione per spiegazioni più specifiche.

- Genera test unitari

Dopo aver inviato un blocco di codice come prompt, chiedi ad Amazon Q Developer di generare test unitari. Questo approccio può far risparmiare tempo e costi di sviluppo relativi alla copertura del codice e DevOps.

- Trova le AWS risposte

Amazon Q Developer è una risorsa preziosa per gli sviluppatori con cui collaborano Servizi AWS perché contiene una grande quantità di conoscenze relative a AWS. Che tu stia affrontando problemi con un messaggio di errore specifico Servizio AWS, riscontrando messaggi di errore specifici o cercando di impararne uno nuovo Servizio AWS, Amazon Q fornisce spesso informazioni pertinenti e utili. AWS

Consulta sempre i consigli che Amazon Q Developer ti fornisce. Quindi, apporta le modifiche necessarie ed esegui i test per assicurarti che il codice soddisfi le funzionalità previste.

Esempi di codice con Amazon Q Developer

Questa sezione fornisce esempi realistici su come migliorare l'esperienza e la generazione di codice, con particolare attenzione a Python e Java lingue. Oltre agli esempi in linea, alcuni scenari mostrano come utilizzare la funzionalità di chat di Amazon Q Developer per richiedere suggerimenti sul codice durante la programmazione.

Per ulteriori informazioni, consulta i seguenti argomenti:

- [Python esempi](#)
- [Java esempi](#)
- [Esempi di chat](#)

Python esempi

I seguenti Python alcuni esempi dimostrano come iniziare a generare suggerimenti di codice in linea con Amazon Q Developer. Questi esempi illustrativi mostrano come attivare l'assistente di programmazione per i tuoi progetti. La maggior parte dei seguenti esempi è applicabile ad altre lingue:

- [Genera classi e funzioni](#)
- [Codice del documento](#)
- [Genera algoritmi](#)
- [Genera test unitari](#)

Genera classi e funzioni

Per generare un'implementazione completa o parziale della classe, utilizzate i commenti al codice. Il commento descrive l'intenzione della classe. Inizia in modo semplice e aggiungi altri dettagli se necessario. Dopo aver generato una classe, potete continuare a generare funzioni all'interno della relativa struttura.

Quando scrivi commenti, è preferibile utilizzare la struttura dei commenti del linguaggio. In Python, per commenti a riga singola, usa `#`. Per commenti su più righe, usa `''' '''` o `""" """`.

Input:

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

Output:

```
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
```

Ora che esiste una classe iniziale, è possibile generare una funzione aggiuntiva con commenti aggiuntivi.

Input:

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

Output:

```
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}")
```

Codice del documento

Riduci i tempi di redazione della documentazione in codice affidandoti alla documentazione in linea. Add (Aggiungi) Python interruzioni di commento. ''' '''

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}")
```

Output:

```
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}")
```

Genera algoritmi

Amazon Q Developer può generare algoritmi popolari, in particolare quelli relativi a AWS SDKs. Nello script, aggiungi un prompt che specifica la funzione che desideri.

Questo esempio mostra come generare un dizionario di chiavi ID utente per denominare i valori e quindi generare una funzione di ordinamento utilizzando l'algoritmo di ordinamento a bolle.

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
...
```

Output:

```
@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)
```

Inoltre, è possibile creare istruzioni per le procedure standard. In questo esempio, l'utente crea un prompt per memorizzare il dizionario in una tabella Amazon DynamoDB. All'inizio dello script, l'utente aggiunge un prompt per aggiungere librerie DynamoDB per fornire ad Amazon Q Developer un contesto aggiuntivo.

Input:

```
'''
Add libraries for Amazon Dynamo DB
'''
```

Output:

```
'''
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')
```

Ora, l'utente può facilmente chiedere ad Amazon Q di aiutarlo a memorizzare il dizionario in DynamoDB.

Input:

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

Output:

```
@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'])
```

Genera test unitari

Con i commenti in linea, Amazon Q Developer può generare test unitari per eseguire asserzioni e altri test. In questo esempio, l'utente chiede ad Amazon Q di generare librerie applicabili per il test unitario per impostare il contesto dello script di test. Quindi, l'utente chiede ad Amazon Q di scrivere una funzione di test per la `sort_user_id_name_map` funzione.

Input:

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

Output:

```
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()
```

Inoltre, in una chat tra l'utente e Amazon Q Developer, suggerisce e genera test unitari in base alle richieste di immissione del codice dell'utente. Per ulteriori informazioni, consulta Esempi di [chat](#).

Java esempi

I seguenti Java gli esempi dimostrano come gli utenti possono iniziare a generare suggerimenti di codice in linea. Questi esempi illustrativi mostrano come attivare l'assistente di programmazione per i propri progetti. La maggior parte dei seguenti esempi è applicabile ad altre lingue:

- [Genera classi e funzioni](#)

- [Codice del documento](#)
- [Genera algoritmi](#)
- [Genera test unitari](#)

Genera classi e funzioni

Per generare un'implementazione completa o parziale della classe, utilizzate i commenti al codice. Il commento descrive l'intenzione della classe. Inizia in modo semplice e aggiungi altri dettagli se necessario. Dopo aver generato una classe, potete continuare a generare funzioni all'interno della relativa struttura.

Quando si scrivono commenti, è preferibile utilizzare la struttura dei commenti del linguaggio. In Java, questo è `/** */`.

Input:

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

Output:

```
/**
 * 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);
    }
}
```

Ora che esiste una classe iniziale, è possibile generare una funzione aggiuntiva con commenti aggiuntivi.

Input:

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

Output:

```
/**
 * 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());
    }
}
```

Codice del documento

Riduci i tempi di redazione della documentazione in codice affidandoti alla documentazione in linea. Aggiungi un Java interruzione di commento `/** */`.

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;
        };
    }
}
```

Se la generazione in linea non avviene, esegui un suggerimento in linea per Amazon Q Developer dopo aver posizionato il cursore dove desideri la documentazione.

Output:

```
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;
        };
    }
}
```

Per spiegazioni più semplici del codice (classi, funzioni o elementi di script), usa i caratteri forward (//) posizionati strategicamente prima delle sezioni di codice in cui desideri aggiungere la documentazione.

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;
        };
    }
}
```

Output:

```
// 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;
        };
    }
}
```

Genera algoritmi

Amazon Q Developer può generare algoritmi popolari, in particolare relativi a AWS SDKs. Nello script, aggiungi un prompt che specifica la funzione che desideri.

Questo esempio genera una mappa hash delle chiavi ID utente per denominare i valori e quindi genera una funzione di ordinamento utilizzando l'algoritmo di ordinamento a bolle.

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
     */
}
```

Output:

```
/*
 * 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;
}

```

Inoltre, gli utenti possono creare istruzioni per procedure standard. In questo esempio, l'utente crea un prompt per memorizzare la mappa hash in una tabella Amazon DynamoDB. All'inizio dello script, l'utente aggiunge un prompt per aggiungere DynamoDB Java librerie per contribuire a fornire ad Amazon Q Developer un contesto aggiuntivo.

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
 */

```

Output:

```

// 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);
        }});
    }
}
```

Genera test unitari

Con i commenti in linea, Amazon Q Developer può generare test unitari per eseguire asserzioni e altri JUnit test. In questo esempio, l'utente chiede innanzitutto ad Amazon Q di generare librerie applicabili JUnit per impostare il contesto dello script di test. Successivamente, l'utente crea una classe pubblica seguita da un dichiaratore e da stringhe di commenti.

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
     */
}
```

Output:

```
@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);
}
```

Inoltre, in una chat tra l'utente e Amazon Q Developer, suggerisce e genera test unitari in base alle richieste di immissione del codice dell'utente. Per ulteriori informazioni, consulta Esempi di [chat](#).

Esempi di chat

La chat per sviluppatori di Amazon Q assiste gli sviluppatori in una serie di attività, dalla codifica, al test e all'aggiornamento delle applicazioni, alla diagnosi degli errori, all'esecuzione di scansioni e correzioni di sicurezza e all'ottimizzazione delle risorse. AWS Questa sezione spiega in che modo la chat con Amazon Q può supportare uno sviluppatore in attività di programmazione tipiche come le seguenti:

- [Chiedi informazioni Servizi AWS](#)
- [Genera codice](#)
- [Genera test unitari](#)
- [Spiega il codice](#)

Chiedi informazioni Servizi AWS

Gli utenti possono porre domande agli sviluppatori di Amazon Q Servizi AWS, ad esempio su come trovare il servizio giusto o comprendere le best practice.

Parla di Servizi AWS

Utente:

Qual è il miglior servizio di container da utilizzare per eseguire il mio carico di lavoro se devo mantenere bassi i costi?

Amazon Q:

In base alle informazioni fornite, sembra che tu stia cercando un servizio container conveniente per eseguire i tuoi carichi di lavoro. In tal caso, consiglieri di prendere in considerazione Amazon Elastic Container Service (Amazon ECS) con Fargate. [1]

Amazon ECS con Fargate è un motore di elaborazione serverless per container che consente di eseguire i container senza dover gestire alcuna infrastruttura sottostante. Con Fargate, non è necessario effettuare il provisioning o gestire i server: Fargate se ne occupa per voi. Questo può aiutarti a mantenere bassi i costi, in quanto paghi solo per le risorse di elaborazione effettivamente utilizzate dai container, senza dover pagare per la capacità inattiva del server. [2]

Alcuni vantaggi chiave dell'utilizzo di Amazon ECS con Fargate:

- Non è necessario gestire server o cluster: Fargate gestisce il provisioning e la gestione dell'infrastruttura per voi.
- Pay-as-you-go prezzi: paghi solo per le risorse di elaborazione utilizzate dai container, senza dover pagare per la capacità inattiva.
- Scalabilità automatica: Fargate può aumentare e ridurre automaticamente i container in base alla domanda.
- Stretta integrazione con altri strumenti Servizi AWS come il bilanciamento del carico, la registrazione e il monitoraggio.

Le risorse fornite indicano inoltre che il servizio ECS offre flessibilità in termini di pianificazione e posizionamento dei container, il che può aiutare a ottimizzare ulteriormente i costi.

Ti consiglio di consultare i dettagli dei prezzi di Amazon ECS con Fargate AWS sul sito Web per avere un'idea migliore dei costi in base ai requisiti specifici del carico di lavoro. E, naturalmente, non esitate a farmi sapere se avete altre domande!

Origini

[1] Qual è AWS il servizio di container migliore per te?

aptgetops.tech > - -tu aws-container-services-which one-is-right-for

[2] Java opzioni di contenitore per carichi di lavoro non HTTP

stackoverflow.com > domande > 43417016

Termina la chat su Amazon Q Servizi AWS

Genera codice

Gli utenti possono chiedere assistenza ad Amazon Q Developer per scrivere frammenti di codice.

Parla sulla scrittura di codice

Utente:

Puoi darmi un esempio di come elencare tutti i bucket S3 usando? AWS SDK per Python (Boto)

Amazon Q:

Certamente, ecco un esempio di come elencare tutti i bucket S3 usando AWS SDK per 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'])
```

Ecco come funziona il codice:

1. Per prima cosa importiamo la libreria boto3, che è l'SDK per Python.
2. Creiamo un client S3 utilizzando `boto3.client('s3')`
3. Chiamiamo il `list_buckets()` metodo sul client S3, che restituisce una risposta contenente un elenco di tutti i bucket dell'account.
4. Quindi eseguiamo un'iterazione sull'elenco contenuto nella risposta e stampiamo il nome di ogni bucket.

Questo è un esempio di base, ma AWS SDK per Python (Boto) offre molte funzionalità più avanzate per lavorare con Amazon S3, come la creazione, l'eliminazione e la gestione di oggetti nei bucket,

l'impostazione delle politiche dei bucket e altro ancora. Puoi trovare altri esempi e documentazione nella () documentazione. AWS SDK per Python (Boto)[2]

Ricorda che quando lavori con Servizi AWS, è importante seguire le migliori pratiche per la gestione delle credenziali e delle informazioni sensibili, ad esempio la memorizzazione delle chiavi di AWS accesso nelle variabili di ambiente anziché inserirle nel codice.

Fonti: [1] Esempi di [Amazon S3 che utilizzano SDK for Python \(AWS Boto3\) - Esempi di codice SDK docs.aws.amazon.com > code-library > latest-1x >... > python-s3.html](https://docs.aws.amazon.com/code-library/latest-1x/python-s3.html)

[2] [Inizia a usare i bucket e gli oggetti Amazon S3 usando un AWS SDK - Esempi di codice SDK docs.aws.amazon.com > code-library > latest-1x AWS >... > s3_example_s3_buckets__sect...](https://docs.aws.amazon.com/code-library/latest-1x/AWS%20%20%20s3_example_s3_buckets__sect...)
GettingStarted

Termina la chat di Amazon Q sulla scrittura del codice

Genera test unitari

Gli utenti possono inviare il codice selezionato come prompt ad Amazon Q Developer, dove vengono applicate istruzioni aggiuntive, come la generazione di un test unitario. Per step-by-step istruzioni, consulta [Send code to Amazon Q](#) nella Amazon Q Developer User Guide.

Parla dei test unitari

Utente:

Scrivi un test unitario con dati fittizi per la seguente funzione:

```
// 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:

Per scrivere un test unitario con dati fittizi per la funzione `pollMessages`, possiamo usare un framework di simulazione come Mockito. Ecco un esempio:

```
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);
  }
}
```

In questo esempio, utilizziamo Mockito per simulare il client AmazonSQS e impostare il comportamento previsto per il metodo `ReceiveMessage` in modo che restituisca due messaggi fittizi con corpi «message1» e «message2». Quindi, chiamiamo il metodo `pollMessages` e affermiamo che l'elenco di messaggi restituito corrisponde ai valori previsti.

I punti chiave sono:

1. Utilizziamo l'annotazione `@ ExtendWith (MockitoExtension.class)` per abilitare l'integrazione di Mockito con. JUnit
2. Utilizziamo l'annotazione `@Mock` per creare un'istanza fittizia del client AmazonSQS.
3. Utilizziamo l'annotazione `@ InjectMocks` per iniettare il client Amazon SQS fittizio nella classe. `SQSUtils`
4. Nel metodo di test, impostiamo il comportamento previsto del metodo `ReceiveMessage` utilizzando `mockito.when () .thenReturn ()`.
5. Chiamiamo il metodo `pollMessages` e affermiamo che l'elenco di messaggi restituito corrisponde ai valori previsti.

Questo approccio ci consente di testare il metodo `pollMessages` senza interagire effettivamente con il servizio Amazon SQS, il che può rendere il test più rapido e affidabile.

Termina la chat di Amazon Q sui test unitari

Spiega il codice

Per i processi di revisione del codice, gli sviluppatori possono inviare una selezione di codice ad Amazon Q Developer per comprendere meglio il codice. Inoltre, la spiegazione fornita da Amazon Q può servire anche a scopo di documentazione. Nell'esempio seguente, Amazon Q spiega la funzione di gestione con una AWS Lambda funzione, quindi propone una docstring.

Parla della spiegazione del codice

Utente:

Spiega la seguente parte del mio codice:

```
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:

Il blocco di codice fornito sembra essere una AWS Lambda funzione scritta in Python. Lasciatemi spiegare gli aspetti chiave di questo codice:

1. La funzione si chiama `lambda_handler` ed è il punto di ingresso per la funzione Lambda.
2. Utilizza la libreria `boto3` per interagire con i servizi AWS DynamoDB e S3.
3. La funzione recupera una chiave oggetto S3 e il nome del bucket dai dati dell'evento in entrata.

4. Quindi chiama una funzione `detect_text` (che non è mostrata nel codice fornito) per estrarre il contenuto del testo completo dell'oggetto S3.
5. La funzione recupera anche i metadati sull'oggetto S3 utilizzando il metodo `head_object`.
6. Quindi estrae l'ora corrente e chiama una funzione `detect_restaurant_closure` (anch'essa non mostrata) per determinare il motivo e l'esito della chiusura di un ristorante in base al testo estratto.
7. Infine, la funzione memorizza le seguenti informazioni in una tabella DynamoDB:
 - La chiave dell'oggetto S3 come id
 - L'ora di creazione
 - Il motivo e l'esito della chiusura del ristorante
 - Il testo estratto dall'oggetto S3

Nel complesso, questa funzione Lambda sembra far parte di un sistema più ampio che elabora oggetti S3, estrae il contenuto di testo, lo analizza per le informazioni sulla chiusura dei ristoranti e archivia i risultati in una tabella DynamoDB. I dettagli specifici delle funzioni `detect_text` e `detect_restaurant_closure` sarebbero necessari per comprendere appieno la funzionalità di questo codice.

Fine della risposta di Amazon Q

Utente: riassumi questa spiegazione in una docstring per questa funzione `lambda_handler`

Amazon Q: Ecco una docstring suggerita per la funzione `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:
        event (dict): The event data that triggered the Lambda function, containing
        information about the S3 object.
        context (object): The context object provided by AWS Lambda, containing
        information about the runtime environment.

    Returns:
        None
    """

    # function implementation details...
```


Risoluzione dei problemi relativi agli scenari di generazione di codice in Amazon Q Developer

Quando usi Amazon Q Developer, potresti riscontrare i seguenti scenari comuni con generazione e risoluzioni di codice imprecise:

- [Generazione di codice vuoto](#)
- [Commenti continui](#)
- [Generazione errata del codice in linea](#)
- [Risultati inadeguati delle chat](#)

Generazione di codice vuoto

Durante lo sviluppo del codice, potresti notare i seguenti problemi:

- Amazon Q non fornisce alcun suggerimento.
- Il messaggio «Nessun suggerimento da Amazon Q» viene visualizzato nel tuo IDE.

Il tuo primo pensiero potrebbe essere che Amazon Q non funzioni correttamente. Tuttavia, la causa principale di questi problemi è in genere associata al contesto dello script o del progetto aperto all'interno dell'IDE.

Se Amazon Q Developer non fornisce automaticamente un suggerimento, puoi utilizzare le seguenti scorciatoie per eseguire manualmente i suggerimenti in linea di Amazon Q:

- PC - Alt+C
- macOS - Opzione+C

Per ulteriori informazioni, consulta [Using shortcut keys](#) nella Amazon Q Developer User Guide.

Nella maggior parte degli scenari, Amazon Q genera un suggerimento. Quando Amazon Q restituisce un messaggio «Nessun suggerimento da Amazon Q», esamina le seguenti soluzioni comuni per questo problema:

- **Contesto inadeguato:** assicurati che siano presenti le librerie applicabili. Assicuratevi che le classi e le funzioni definite nello script si riferiscano al nuovo codice.
- **Richiesta ambigua:** se il prompt è confuso o poco chiaro, Amazon Q avrà prestazioni inferiori quando fornisce suggerimenti di codice in linea. In questi scenari, sii più specifico nel codice e nei commenti per ottenere risposte corrette.

Se sei bloccato, continua a programmare al meglio delle tue capacità. In genere, Amazon Q riconoscerà la tua intenzione, il che lo spinge a offrire suggerimenti. Un'altra opzione è inviare un frammento di codice con istruzioni rapide in una chat di Amazon Q. Per ulteriori informazioni, consulta [Send code to Amazon Q](#) nella Amazon Q Developer User Guide.

Commenti continui

Potresti scrivere un prompt utilizzando i commenti e ricevere solo righe di commento generiche o ripetute da Amazon Q.

Input:

```
// Kafka test
```

Risultato:

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

In genere, questo problema è dovuto alla mancanza di contesto nello script o a istruzioni tempestive insufficienti. In base alla lingua, si consiglia di utilizzare il formato standard per i commenti. In questo esempio, l'utente ha modificato il prompt da utilizzare `/* */` per un blocco di commenti e tale modifica ha comportato un miglioramento delle prestazioni di Amazon Q.

Input:

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

Risultato:

In questo risultato, l'utente ha accettato diversi suggerimenti in sequenza.

```
/*
 * 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());
    }
}
```

Generazione errata del codice in linea

A volte, i suggerimenti in linea per gli sviluppatori di Amazon Q non sono utili o non soddisfano l'obiettivo del prompt. A volte, i suggerimenti di codice in linea possono fornire una funzione generica simile a un semplice metodo Hello World.

Se i suggerimenti sul codice in linea non sono utili, prova le seguenti best practice:

- Esamina il contesto dello script. In genere, maggiore è il contesto fornito, migliori sono i risultati ottenuti da Amazon Q.
- Se il contesto varia and/or è rumoroso, modularizza il codice. Raggruppa funzioni simili all'interno dello stesso script. Separa classi e funzioni diverse in base al loro obiettivo.
- Scrivi i nomi delle classi e delle funzioni in modo tattico per aiutare Amazon Q a fornire consigli migliori.

In questo esempio, lo sviluppatore sta scrivendo uno script da zero che carica i file su Hadoop.

Input:

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

Risultato:

```
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));  
  
    }  
}
```

Sebbene Amazon Q abbia fornito una raccomandazione, questa non ha soddisfatto le aspettative dello sviluppatore. Per migliorare le prestazioni, lo sviluppatore importa le librerie e modifica il nome della classe per rappresentare le utilità Hadoop.

Correzione:

```
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  
}
```

Risultato:

In questo risultato, lo sviluppatore ha accettato diversi suggerimenti in sequenza.

```
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!");  
    }  
}
```

Nello stesso HadoopUtilities script, lo sviluppatore aggiunge un prompt di commento per leggere un argomento di Amazon Simple Queue Service (Amazon SQS).

Input:

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

Risultato:

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

Invece di utilizzare metodi Java per leggere un messaggio SQS da Amazon SQS, Amazon Q suggerisce di stampare un metodo stringa. Tuttavia, questo suggerimento non soddisfa l'obiettivo originale dello sviluppatore.

In questo caso, è preferibile rimodularizzare il codice. Lo script si concentra sui metodi Hadoop. Pertanto, potrebbe essere un approccio migliore creare uno script di utilità specifico per le operazioni SQS.

La creazione di una classe separata per le utilità SQS migliora il suggerimento in linea, che può essere utilizzato nello script. HadoopUtilities

Input:

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

class SQSUtils {

    // instantiate SQS client
    private AmazonSQS sqsClient;

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

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

Risultato:

```
// 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;
```

Se il suggerimento in linea continua a non fornire i risultati previsti, chiedi ad Amazon Q Developer in una sessione di chat di generare il codice.

Parla della generazione di codice

Utente:

Genera una funzione java per leggere un messaggio da una coda Amazon SQS

Amazon Q:

Ecco un esempio di funzione Java per leggere un messaggio da una coda 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;
}
}
```

Questo codice crea una classe SQSMessage Reader che accetta un client AmazonSQS e un URL di coda come parametri di costruzione. Il metodo readMessage () utilizza il ReceiveMessageRequest per recuperare fino a 1 messaggio dalla coda, con un tempo di attesa massimo di 10 secondi.

Se viene ricevuto un messaggio, il corpo del messaggio viene stampato e il messaggio viene eliminato dalla coda utilizzando il DeleteMessageRequest Il corpo del messaggio viene quindi restituito. Se non viene ricevuto alcun messaggio, viene restituito null.

Termina la chat di Amazon Q sulla generazione di codice

Risultati inadeguati delle chat

Durante lo sviluppo del codice, gli sviluppatori possono chiedere ad Amazon Q di generare funzioni comuni, fornire consigli o spiegare il codice. A volte, quando richiesto (ad esempio una domanda o un frammento di codice), Amazon Q potrebbe fornire un feedback generale o risultati che non soddisfano le tue aspettative. In questi scenari, prova quanto segue:

- Sperimenta diverse istruzioni per migliorare i risultati di generazione del codice che ottieni da Amazon Q.
- Segui le URLs informazioni che Amazon Q fornisce frequentemente nelle sue risposte alle tue richieste o domande.
- Fornisci feedback per contribuire a migliorare le prestazioni future di Amazon Q.

Ad esempio, uno sviluppatore sta creando una soluzione Retrieval Augmented Generation (RAG) e deve suddividere il documento in parti in base alle relazioni semantiche tra le frasi. Lo sviluppatore parla con Amazon Q nel modo seguente.

Parla di una soluzione RAG

Utente:

Per una soluzione RAG, genera codice per la suddivisione in blocchi semantici

Amazon Q:

Certamente, ecco un esempio di come è possibile implementare il chunking semantico per una soluzione Retrieval Augmented Generation (RAG) utilizzando la libreria: Python 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
```

In questo esempio, utilizziamo il modulo LangChain per dividere il testo TokenTextSplitter di input in blocchi di 500 caratteri con una sovrapposizione di 50 caratteri. Il HuggingFaceEmbeddings modello viene utilizzato per generare incorporamenti per ogni blocco e il Chroma vector store viene utilizzato per memorizzare i blocchi e i relativi incorporamenti.

La funzione `semantic_chunk` raccoglie il testo di input, lo divide in blocchi utilizzando `text_splitter`, genera incorporamenti per ogni blocco utilizzando il modello di incorporamento e quindi aggiunge i blocchi e i relativi incorporamenti a `vectorstore`

Questo approccio consente di suddividere il testo di input in blocchi semanticamente significativi, che possono essere utili per una varietà di attività, come la risposta a domande o il riepilogo, in una soluzione RAG.

Tieni presente che questo è un esempio di base e potrebbe essere necessario modificare i parametri (ad esempio, `chunk_size`, `chunk_overlap`) o la scelta del modello di divisione del testo e degli incorporamenti in base al caso d'uso e ai dati specifici.

Fine della chat di Amazon Q su una soluzione RAG

Tuttavia, questo approccio fornisce solo un metodo di base per suddividere i documenti utilizzando la tecnica della finestra scorrevole. Lo sviluppatore desidera un metodo avanzato che si basi sulla misurazione degli incorporamenti. In questi scenari, lo sviluppatore deve aggiungere ulteriori dettagli al prompt in base al proprio obiettivo. Quando lo sviluppatore fa riferimento alla nota funzione di divisione del AI21 testo con `LangChain`, riceve una risposta migliore, come mostrato nella seguente chat.

Parla di una soluzione RAG (con prompt migliorato)

Utente:

Il prompt precedente si riferisce solo alla suddivisione in base alla dimensione del token per blocco. Voglio usare lo splitter di testo semantico AI21 . Genera una funzione usando la suddivisione semantica del testo AI21 con `langchain`.

Amazon Q:

Ok, usiamo il AI21 Semantic Text Splitter con `LangChain` per implementare il chunking semantico. Ecco un esempio: [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
```

In questo esempio, utilizziamo il form LangChain per dividere il AI21 SemanticTextSplitter testo di input in blocchi semanticamente significativi. Il HuggingFaceEmbeddings modello viene utilizzato per creare incorporamenti per ogni blocco e il Chroma vector store viene utilizzato per memorizzare i blocchi e i relativi incorporamenti.

La funzione semantic_chunk raccoglie il testo di input, lo divide in blocchi utilizzando text_splitter, crea incorporamenti per ogni blocco utilizzando il modello di incorporamento e quindi aggiunge i blocchi e i relativi incorporamenti a. vectorstore

Questo approccio consente di suddividere il testo di input in blocchi semanticamente significativi, che possono essere utili per una varietà di attività, come la risposta a domande o il riepilogo, in una soluzione RAG.

Tieni presente che potrebbe essere necessario modificare i parametri o la scelta del modello di incorporamento in base al caso d'uso e ai dati specifici.

Fonti [1] Scopri come utilizzare i divisori di testo in LangChain

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

Fine della chat di Amazon Q su una soluzione RAG (con prompt migliorato)

In base ai risultati della chat, lo sviluppatore può utilizzare la funzione e accedere alla fonte delle informazioni.

In generale, argomenti e domande avanzati richiedono un contesto più specifico all'interno del prompt quando chatti con Amazon Q Developer. Se ritieni che i risultati della chat non siano accurati, utilizza l'icona con il pollice in giù per fornire feedback sulla risposta di Amazon Q. Amazon Q Developer utilizza continuamente il feedback per migliorare le versioni future. Per le interazioni che hanno prodotto risultati positivi, è utile fornire il proprio feedback utilizzando l'icona con il pollice rivolto verso l'alto.

FAQs informazioni su Amazon Q Developer

Questa sezione fornisce risposte alle domande frequenti sull'uso di Amazon Q Developer per lo sviluppo di codice.

Cos'è Amazon Q Developer?

Amazon Q Developer è un potente servizio generativo basato sull'intelligenza artificiale progettato per accelerare le attività di sviluppo del codice fornendo generazione di codice e consigli intelligenti. Il 30 aprile 2024, Amazon CodeWhisperer è entrata a far parte di Amazon Q Developer.

Come posso accedere ad Amazon Q Developer?

Amazon Q Developer è disponibile come parte dei AWS Toolkit for Visual Studio Code e JetBrains IDEs, ad esempio, IntelliJ e. PyCharm Per iniziare, [installa](#) la versione più recente. Kit di strumenti AWS

Quali linguaggi di programmazione supporta Amazon Q Developer?

Per Visual Studio Code e JetBrains IDEs, Amazon Q Developer supporta PythonJava, JavaScript, TypeScript, C#, PHP GoRust,, KotlinRuby, C, C++, Shell scripting, SQL e Scala. Sebbene questa guida si concentri su Python Java scopi esemplificativi, i concetti sono applicabili a qualsiasi linguaggio di programmazione supportato.

Come posso fornire un contesto ad Amazon Q Developer per una migliore generazione di codice?

Inizia con il codice esistente, importa le librerie pertinenti, crea classi e funzioni o stabilisci scheletri di codice. Usa blocchi di commenti standard per i prompt in linguaggio naturale. Mantieni lo script incentrato su obiettivi specifici e modularizza funzionalità distinte in script separati con un contesto pertinente. Per ulteriori informazioni, consulta [Best coding practices with Amazon Q Developer](#).

Cosa devo fare se la generazione di codice in linea con Amazon Q Developer non è accurata?

Esamina il contesto dello script, assicurati che le librerie siano presenti e assicurati che le classi e le funzioni siano correlate al nuovo codice. Modularizza il codice e separa le diverse classi e funzioni in base al loro obiettivo. Scrivi istruzioni o commenti chiari e specifici. Se hai ancora dubbi sulla precisione del codice e non riesci a procedere, avvia una chat con Amazon Q e inviagli lo snippet di codice con le istruzioni. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi agli scenari di generazione di codice in Amazon Q Developer](#).

Come posso utilizzare la funzionalità di chat di Amazon Q Developer per la generazione di codice e la risoluzione dei problemi?

Chatta con Amazon Q per generare funzioni comuni, chiedere consigli o spiegare il codice. Se la risposta iniziale non è soddisfacente, prova diverse istruzioni e segui quelle fornite. URLs Inoltre, fornisci feedback ad Amazon Q per aiutarlo a migliorare le sue future prestazioni di chat. Usa le icone con il pollice su e il pollice giù per fornire il tuo feedback. [Per ulteriori informazioni, consulta Esempi di chat](#).

Quali sono alcune best practice per l'utilizzo di Amazon Q Developer?

Fornisci il contesto pertinente, sperimenta e ripeti le istruzioni, esamina i suggerimenti sul codice prima di accettarli, utilizza le funzionalità di personalizzazione e comprendi le politiche sulla privacy dei dati e sull'utilizzo dei contenuti. Per ulteriori informazioni, consulta [Best practice per la generazione di codice con Amazon Q Developer](#) e [Best practice per consigli sul codice con Amazon Q Developer](#).

Posso personalizzare Amazon Q Developer per generare consigli basati sul mio codice?

Sì, usa le personalizzazioni, che sono una funzionalità avanzata di Amazon Q Developer. Con le personalizzazioni, le aziende possono fornire i propri repository di codice per consentire ad Amazon

Q Developer di consigliare suggerimenti di codice in linea. Per ulteriori informazioni, consulta [Funzionalità avanzate di Amazon Q Developer](#) and [Resources](#).

Fasi successive dell'utilizzo di Amazon Q Developer

Grazie alle conoscenze acquisite con questa guida completa, puoi utilizzare Amazon Q Developer nel tuo flusso di lavoro di codifica in modo efficace. Installalo Kit di strumenti AWS nel tuo IDE preferito ([Visual Studio Code o JetBrains](#)) e [inizia a esplorare la generazione di codice](#) generativo basata sull'intelligenza artificiale e i consigli di Amazon Q Developer.

Il modo più efficace per sfruttare appieno il potenziale di Amazon Q Developer è l'esperienza pratica con il proprio codice. Mentre integri Amazon Q nel tuo ciclo di vita di sviluppo, consulta questa guida per le migliori pratiche, la risoluzione dei problemi ed esempi reali.

[Inoltre, tieniti informato consultando AWS i blog e le guide per sviluppatori a cui si fa riferimento in Risorse.](#) Queste risorse forniscono gli aggiornamenti, le best practice e gli approfondimenti più recenti per aiutarti a ottimizzare l'uso di Amazon Q Developer.

Il tuo feedback è prezioso per migliorare questa guida e aiutarla a rimanere una risorsa preziosa per gli sviluppatori. Condividi esperienze, sfide e suggerimenti per le versioni future. Il tuo contributo contribuirà a migliorare la guida con esempi aggiuntivi, scenari di risoluzione dei problemi e approfondimenti personalizzati in base alle tue esigenze.

Risorse

AWS blog

- [Accelerate your Software Development Lifecycle with Amazon Q](#)
- [Reinventare lo sviluppo del software con Amazon Q Developer Agent](#)
- [Cinque esempi di risoluzione dei problemi con Amazon Q](#)
- [Personalizza Amazon Q Developer nel tuo account IDE con la tua base di codice privata](#)
- [Tre modi in cui l'agente Amazon Q Developer per la trasformazione del codice accelera gli aggiornamenti Java](#)
- [Sfruttare Amazon Q Developer per un debug e una manutenzione efficienti del codice](#)
- [Test delle applicazioni con Amazon Q Developer](#)

AWS documentazione

- [Guida per l'utente di Amazon Q per sviluppatori](#)
- [Personalizzazione del codice Amazon Q Developer](#)
- [Trasformazione del codice Amazon Q Developer](#)

AWS workshop

- [Giornata di immersione per sviluppatori Amazon Q](#)
- [Workshop per sviluppatori Amazon Q - Creazione dell'app Q-Words](#)
- [Workshop per sviluppatori Amazon Q - Creazione di suggerimenti efficaci](#)

Collaboratori

Le seguenti persone hanno contribuito a questa guida:

- Joe King, Scienziato senior dei dati, AWS
- Prateek Gupta, responsabile del team — Sr. CAA, AWS
- Manohar Reddy Arranagu, architetto, DevOps AWS
- Soumik Roy, architetto di applicazioni cloud, AWS
- Sanket Shinde, consulente, AWS

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un [RSSfeed](#).

Modifica	Descrizione	Data
Pubblicazione iniziale	—	16 agosto 2024

AWS Glossario delle linee guida prescrittive

I seguenti sono termini di uso comune nelle strategie, nelle guide e nei modelli forniti da AWS Prescriptive Guidance. Per suggerire voci, utilizza il link [Fornisci feedback](#) alla fine del glossario.

Numeri

7 R

Sette strategie di migrazione comuni per trasferire le applicazioni sul cloud. Queste strategie si basano sulle 5 R identificate da Gartner nel 2011 e sono le seguenti:

- **Rifattorizzare/riprogettare:** trasferisci un'applicazione e modifica la sua architettura sfruttando appieno le funzionalità native del cloud per migliorare l'agilità, le prestazioni e la scalabilità. Ciò comporta in genere la portabilità del sistema operativo e del database. Esempio: migra il tuo database Oracle locale all'edizione compatibile con Amazon Aurora PostgreSQL.
- **Ridefinire la piattaforma (lift and reshape):** trasferisci un'applicazione nel cloud e introduci un certo livello di ottimizzazione per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale ad Amazon Relational Database Service (Amazon RDS) per Oracle in Cloud AWS
- **Riacquistare (drop and shop):** passa a un prodotto diverso, in genere effettuando la transizione da una licenza tradizionale a un modello SaaS. Esempio: migra il tuo sistema di gestione delle relazioni con i clienti (CRM) su Salesforce.com.
- **Eseguire il rehosting (lift and shift):** trasferisci un'applicazione sul cloud senza apportare modifiche per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale a Oracle su un'istanza EC2 in Cloud AWS
- **Trasferire (eseguire il rehosting a livello hypervisor):** trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Esegui la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione su Microsoft Hyper-V. AWS
- **Riesaminare (mantenere):** mantieni le applicazioni nell'ambiente di origine. Queste potrebbero includere applicazioni che richiedono una rifattorizzazione significativa che desideri rimandare a un momento successivo e applicazioni legacy che desideri mantenere, perché non vi è alcuna giustificazione aziendale per effettuarne la migrazione.
- **Ritirare:** disattiva o rimuovi le applicazioni che non sono più necessarie nell'ambiente di origine.

A

ABAC

Vedi controllo degli accessi [basato sugli attributi](#).

servizi astratti

Vedi [servizi gestiti](#).

ACIDO

Vedi [atomicità, consistenza, isolamento, durata](#).

migrazione attiva-attiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati (utilizzando uno strumento di replica bidirezionale o operazioni di doppia scrittura) ed entrambi i database gestiscono le transazioni provenienti dalle applicazioni di connessione durante la migrazione. Questo metodo supporta la migrazione in piccoli batch controllati anziché richiedere una conversione una tantum. È più flessibile ma richiede più lavoro rispetto alla migrazione [attiva-passiva](#).

migrazione attiva-passiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati, ma solo il database di origine gestisce le transazioni provenienti dalle applicazioni di connessione mentre i dati vengono replicati nel database di destinazione. Il database di destinazione non accetta alcuna transazione durante la migrazione.

funzione di aggregazione

Una funzione SQL che opera su un gruppo di righe e calcola un singolo valore restituito per il gruppo. Esempi di funzioni aggregate includono SUM e MAX.

Intelligenza artificiale

Vedi [intelligenza artificiale](#).

AIOps

Guarda le [operazioni di intelligenza artificiale](#).

anonimizzazione

Il processo di eliminazione permanente delle informazioni personali in un set di dati.

L'anonimizzazione può aiutare a proteggere la privacy personale. I dati anonimi non sono più considerati dati personali.

anti-modello

Una soluzione utilizzata frequentemente per un problema ricorrente in cui la soluzione è controproducente, inefficace o meno efficace di un'alternativa.

controllo delle applicazioni

Un approccio alla sicurezza che consente l'uso solo di applicazioni approvate per proteggere un sistema dal malware.

portfolio di applicazioni

Una raccolta di informazioni dettagliate su ogni applicazione utilizzata da un'organizzazione, compresi i costi di creazione e manutenzione dell'applicazione e il relativo valore aziendale. Queste informazioni sono fondamentali per [il processo di scoperta e analisi del portfolio](#) e aiutano a identificare e ad assegnare la priorità alle applicazioni da migrare, modernizzare e ottimizzare.

intelligenza artificiale (IA)

Il campo dell'informatica dedicato all'uso delle tecnologie informatiche per svolgere funzioni cognitive tipicamente associate agli esseri umani, come l'apprendimento, la risoluzione di problemi e il riconoscimento di schemi. Per ulteriori informazioni, consulta la sezione [Che cos'è l'intelligenza artificiale?](#)

operazioni di intelligenza artificiale (AIOps)

Il processo di utilizzo delle tecniche di machine learning per risolvere problemi operativi, ridurre gli incidenti operativi e l'intervento umano e aumentare la qualità del servizio. Per ulteriori informazioni su come AIOps viene utilizzata nella strategia di AWS migrazione, consulta la [guida all'integrazione delle operazioni](#).

crittografia asimmetrica

Un algoritmo di crittografia che utilizza una coppia di chiavi, una chiave pubblica per la crittografia e una chiave privata per la decrittografia. Puoi condividere la chiave pubblica perché non viene utilizzata per la decrittografia, ma l'accesso alla chiave privata deve essere altamente limitato.

atomicità, consistenza, isolamento, durabilità (ACID)

Un insieme di proprietà del software che garantiscono la validità dei dati e l'affidabilità operativa di un database, anche in caso di errori, interruzioni di corrente o altri problemi.

Controllo degli accessi basato su attributi (ABAC)

La pratica di creare autorizzazioni dettagliate basate su attributi utente, come reparto, ruolo professionale e nome del team. Per ulteriori informazioni, consulta [ABAC AWS](#) nella documentazione AWS Identity and Access Management (IAM).

fonte di dati autorevole

Una posizione in cui è archiviata la versione principale dei dati, considerata la fonte di informazioni più affidabile. È possibile copiare i dati dalla fonte di dati autorevole in altre posizioni allo scopo di elaborarli o modificarli, ad esempio anonimizzandoli, oscurandoli o pseudonimizzandoli.

Zona di disponibilità

Una posizione distinta all'interno di un edificio Regione AWS che è isolata dai guasti in altre zone di disponibilità e offre una connettività di rete economica e a bassa latenza verso altre zone di disponibilità nella stessa regione.

AWS Cloud Adoption Framework (CAF)AWS

Un framework di linee guida e best practice AWS per aiutare le organizzazioni a sviluppare un piano efficiente ed efficace per passare con successo al cloud. AWS CAF organizza le linee guida in sei aree di interesse chiamate prospettive: business, persone, governance, piattaforma, sicurezza e operazioni. Le prospettive relative ad azienda, persone e governance si concentrano sulle competenze e sui processi aziendali; le prospettive relative alla piattaforma, alla sicurezza e alle operazioni si concentrano sulle competenze e sui processi tecnici. Ad esempio, la prospettiva relativa alle persone si rivolge alle parti interessate che gestiscono le risorse umane (HR), le funzioni del personale e la gestione del personale. In questa prospettiva, AWS CAF fornisce linee guida per lo sviluppo delle persone, la formazione e le comunicazioni per aiutare a preparare l'organizzazione all'adozione del cloud di successo. Per ulteriori informazioni, consulta il [sito web di AWS CAF](#) e il [white paper AWS CAF](#).

AWS Workload Qualification Framework (WQF)AWS

Uno strumento che valuta i carichi di lavoro di migrazione dei database, consiglia strategie di migrazione e fornisce stime del lavoro. AWS WQF è incluso in (). AWS Schema Conversion Tool AWS SCT Analizza gli schemi di database e gli oggetti di codice, il codice dell'applicazione, le dipendenze e le caratteristiche delle prestazioni e fornisce report di valutazione.

B

bot difettoso

Un [bot](#) che ha lo scopo di interrompere o causare danni a individui o organizzazioni.

BCP

Vedi la [pianificazione della continuità operativa](#).

grafico comportamentale

Una vista unificata, interattiva dei comportamenti delle risorse e delle interazioni nel tempo. Puoi utilizzare un grafico comportamentale con Amazon Detective per esaminare tentativi di accesso non riusciti, chiamate API sospette e azioni simili. Per ulteriori informazioni, consulta [Dati in un grafico comportamentale](#) nella documentazione di Detective.

sistema big-endian

Un sistema che memorizza per primo il byte più importante. Vedi anche [endianness](#).

Classificazione binaria

Un processo che prevede un risultato binario (una delle due classi possibili). Ad esempio, il modello di machine learning potrebbe dover prevedere problemi come "Questa e-mail è spam o non è spam?" o "Questo prodotto è un libro o un'auto?"

filtro Bloom

Una struttura di dati probabilistica ed efficiente in termini di memoria che viene utilizzata per verificare se un elemento fa parte di un set.

implementazione blu/verde

Una strategia di implementazione in cui si creano due ambienti separati ma identici. La versione corrente dell'applicazione viene eseguita in un ambiente (blu) e la nuova versione dell'applicazione nell'altro ambiente (verde). Questa strategia consente di ripristinare rapidamente il sistema con un impatto minimo.

bot

Un'applicazione software che esegue attività automatizzate su Internet e simula l'attività o l'interazione umana. Alcuni bot sono utili o utili, come i web crawler che indicizzano le informazioni su Internet. Alcuni altri bot, noti come bot dannosi, hanno lo scopo di disturbare o causare danni a individui o organizzazioni.

botnet

Reti di [bot](#) infettate da [malware](#) e controllate da un'unica parte, nota come bot herder o bot operator. Le botnet sono il meccanismo più noto per scalare i bot e il loro impatto.

ramo

Un'area contenuta di un repository di codice. Il primo ramo creato in un repository è il ramo principale. È possibile creare un nuovo ramo a partire da un ramo esistente e quindi sviluppare funzionalità o correggere bug al suo interno. Un ramo creato per sviluppare una funzionalità viene comunemente detto ramo di funzionalità. Quando la funzionalità è pronta per il rilascio, il ramo di funzionalità viene ricongiunto al ramo principale. Per ulteriori informazioni, consulta [Informazioni sulle filiali](#) (documentazione). GitHub

accesso break-glass

In circostanze eccezionali e tramite una procedura approvata, un mezzo rapido per consentire a un utente di accedere a un sito a Account AWS cui in genere non dispone delle autorizzazioni necessarie. Per ulteriori informazioni, vedere l'indicatore [Implementate break-glass procedures](#) nella guida Well-Architected AWS .

strategia brownfield

L'infrastruttura esistente nell'ambiente. Quando si adotta una strategia brownfield per un'architettura di sistema, si progetta l'architettura in base ai vincoli dei sistemi e dell'infrastruttura attuali. Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e [greenfield](#).

cache del buffer

L'area di memoria in cui sono archiviati i dati a cui si accede con maggiore frequenza.

capacità di business

Azioni intraprese da un'azienda per generare valore (ad esempio vendite, assistenza clienti o marketing). Le architetture dei microservizi e le decisioni di sviluppo possono essere guidate dalle capacità aziendali. Per ulteriori informazioni, consulta la sezione [Organizzazione in base alle funzionalità aziendali](#) del whitepaper [Esecuzione di microservizi containerizzati su AWS](#).

pianificazione della continuità operativa (BCP)

Un piano che affronta il potenziale impatto di un evento che comporta l'interruzione dell'attività, come una migrazione su larga scala, sulle operazioni e consente a un'azienda di riprendere rapidamente le operazioni.

C

CAF

Vedi [Cloud Adoption AWS Framework](#).

implementazione canaria

Il rilascio lento e incrementale di una versione agli utenti finali. Quando sei sicuro, distribuisce la nuova versione e sostituisci la versione corrente nella sua interezza.

CCoE

Vedi [Cloud Center of Excellence](#).

CDC

Vedi [Change Data Capture](#).

Change Data Capture (CDC)

Il processo di tracciamento delle modifiche a un'origine dati, ad esempio una tabella di database, e di registrazione dei metadati relativi alla modifica. È possibile utilizzare CDC per vari scopi, ad esempio il controllo o la replica delle modifiche in un sistema di destinazione per mantenere la sincronizzazione.

ingegneria del caos

Introduzione intenzionale di guasti o eventi dirompenti per testare la resilienza di un sistema. Puoi usare [AWS Fault Injection Service \(AWS FIS\)](#) per eseguire esperimenti che stressano i tuoi AWS carichi di lavoro e valutarne la risposta.

CI/CD

Vedi [integrazione continua e distribuzione continua](#).

classificazione

Un processo di categorizzazione che aiuta a generare previsioni. I modelli di ML per problemi di classificazione prevedono un valore discreto. I valori discreti sono sempre distinti l'uno dall'altro. Ad esempio, un modello potrebbe dover valutare se in un'immagine è presente o meno un'auto.

crittografia lato client

Crittografia dei dati a livello locale, prima che il destinatario li Servizio AWS riceva.

Centro di eccellenza cloud (CCoE)

Un team multidisciplinare che guida le iniziative di adozione del cloud in tutta l'organizzazione, tra cui lo sviluppo di best practice per il cloud, la mobilitazione delle risorse, la definizione delle tempistiche di migrazione e la guida dell'organizzazione attraverso trasformazioni su larga scala. Per ulteriori informazioni, consulta gli [CCoE post](#) sull' Cloud AWS Enterprise Strategy Blog.

cloud computing

La tecnologia cloud generalmente utilizzata per l'archiviazione remota di dati e la gestione dei dispositivi IoT. Il cloud computing è generalmente collegato alla tecnologia di [edge computing](#).

modello operativo cloud

In un'organizzazione IT, il modello operativo utilizzato per creare, maturare e ottimizzare uno o più ambienti cloud. Per ulteriori informazioni, consulta [Building your Cloud Operating Model](#).

fasi di adozione del cloud

Le quattro fasi che le organizzazioni in genere attraversano quando migrano verso Cloud AWS:

- Progetto: esecuzione di alcuni progetti relativi al cloud per scopi di dimostrazione e apprendimento
- Fondamento: effettuare investimenti fondamentali per scalare l'adozione del cloud (ad esempio, creazione di una landing zone, definizione di una CCo E, definizione di un modello operativo)
- Migrazione: migrazione di singole applicazioni
- Reinvenzione: ottimizzazione di prodotti e servizi e innovazione nel cloud

Queste fasi sono state definite da Stephen Orban nel post sul blog The [Journey Toward Cloud-First & the Stages of Adoption on the Enterprise Strategy](#). Cloud AWS [Per informazioni su come si relazionano alla strategia di AWS migrazione, consulta la guida alla preparazione alla migrazione.](#)

CMDB

Vedi [database di gestione della configurazione](#).

repository di codice

Una posizione in cui il codice di origine e altri asset, come documentazione, esempi e script, vengono archiviati e aggiornati attraverso processi di controllo delle versioni. Gli archivi cloud più comuni includono GitHub oBitbucket Cloud. Ogni versione del codice è denominata ramo. In una struttura a microservizi, ogni repository è dedicato a una singola funzionalità. Una singola pipeline CI/CD può utilizzare più repository.

cache fredda

Una cache del buffer vuota, non ben popolata o contenente dati obsoleti o irrilevanti. Ciò influisce sulle prestazioni perché l'istanza di database deve leggere dalla memoria o dal disco principale, il che richiede più tempo rispetto alla lettura dalla cache del buffer.

dati freddi

Dati a cui si accede raramente e che in genere sono storici. Quando si eseguono interrogazioni di questo tipo di dati, le interrogazioni lente sono in genere accettabili. Lo spostamento di questi dati su livelli o classi di storage meno costosi e con prestazioni inferiori può ridurre i costi.

visione artificiale (CV)

Un campo dell'[intelligenza artificiale](#) che utilizza l'apprendimento automatico per analizzare ed estrarre informazioni da formati visivi come immagini e video digitali. Ad esempio, Amazon SageMaker AI fornisce algoritmi di elaborazione delle immagini per CV.

deriva della configurazione

Per un carico di lavoro, una modifica della configurazione rispetto allo stato previsto. Potrebbe causare la non conformità del carico di lavoro e in genere è graduale e involontaria.

database di gestione della configurazione (CMDB)

Un repository che archivia e gestisce le informazioni su un database e il relativo ambiente IT, inclusi i componenti hardware e software e le relative configurazioni. In genere si utilizzano i dati di un CMDB nella fase di individuazione e analisi del portafoglio della migrazione.

Pacchetto di conformità

Una raccolta di AWS Config regole e azioni correttive che puoi assemblare per personalizzare i controlli di conformità e sicurezza. È possibile distribuire un pacchetto di conformità come singola entità in una regione Account AWS and o all'interno di un'organizzazione utilizzando un modello YAML. Per ulteriori informazioni, consulta i [Conformance](#) Pack nella documentazione. AWS Config

integrazione e distribuzione continua (continuous integration and continuous delivery, CI/CD)

Il processo di automazione delle fasi di origine, compilazione, test, gestione temporanea e produzione del processo di rilascio del software. CI/CD viene comunemente descritto come una pipeline. CI/CD può aiutarvi ad automatizzare i processi, migliorare la produttività, migliorare la qualità del codice e velocizzare le consegne. Per ulteriori informazioni, consulta [Vantaggi](#)

[della distribuzione continua](#). CD può anche significare continuous deployment (implementazione continua). Per ulteriori informazioni, consulta [Distribuzione continua e implementazione continua a confronto](#).

CV

Vedi [visione artificiale](#).

D

dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

classificazione dei dati

Un processo per identificare e classificare i dati nella rete in base alla loro criticità e sensibilità. È un componente fondamentale di qualsiasi strategia di gestione dei rischi di sicurezza informatica perché consente di determinare i controlli di protezione e conservazione appropriati per i dati. La classificazione dei dati è un componente del pilastro della sicurezza nel AWS Well-Architected Framework. Per ulteriori informazioni, consulta [Classificazione dei dati](#).

deriva dei dati

Una variazione significativa tra i dati di produzione e i dati utilizzati per addestrare un modello di machine learning o una modifica significativa dei dati di input nel tempo. La deriva dei dati può ridurre la qualità, l'accuratezza e l'equità complessive nelle previsioni dei modelli ML.

dati in transito

Dati che si spostano attivamente attraverso la rete, ad esempio tra le risorse di rete.

rete di dati

Un framework architettonico che fornisce la proprietà distribuita e decentralizzata dei dati con gestione e governance centralizzate.

riduzione al minimo dei dati

Il principio della raccolta e del trattamento dei soli dati strettamente necessari. Praticare la riduzione al minimo dei dati in the Cloud AWS può ridurre i rischi per la privacy, i costi e l'impronta di carbonio delle analisi.

perimetro dei dati

Una serie di barriere preventive nell' AWS ambiente che aiutano a garantire che solo le identità attendibili accedano alle risorse attendibili delle reti previste. Per ulteriori informazioni, consulta [Building a data perimeter](#) on. AWS

pre-elaborazione dei dati

Trasformare i dati grezzi in un formato che possa essere facilmente analizzato dal modello di ML. La pre-elaborazione dei dati può comportare la rimozione di determinate colonne o righe e l'eliminazione di valori mancanti, incoerenti o duplicati.

provenienza dei dati

Il processo di tracciamento dell'origine e della cronologia dei dati durante il loro ciclo di vita, ad esempio il modo in cui i dati sono stati generati, trasmessi e archiviati.

soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

data warehouse

Un sistema di gestione dei dati che supporta la business intelligence, come l'analisi. I data warehouse contengono in genere grandi quantità di dati storici e vengono generalmente utilizzati per interrogazioni e analisi.

linguaggio di definizione del database (DDL)

Istruzioni o comandi per creare o modificare la struttura di tabelle e oggetti in un database.

linguaggio di manipolazione del database (DML)

Istruzioni o comandi per modificare (inserire, aggiornare ed eliminare) informazioni in un database.

DDL

Vedi linguaggio di [definizione del database](#).

deep ensemble

Combinare più modelli di deep learning per la previsione. È possibile utilizzare i deep ensemble per ottenere una previsione più accurata o per stimare l'incertezza nelle previsioni.

deep learning

Un sottocampo del ML che utilizza più livelli di reti neurali artificiali per identificare la mappatura tra i dati di input e le variabili target di interesse.

defense-in-depth

Un approccio alla sicurezza delle informazioni in cui una serie di meccanismi e controlli di sicurezza sono accuratamente stratificati su una rete di computer per proteggere la riservatezza, l'integrità e la disponibilità della rete e dei dati al suo interno. Quando si adotta questa strategia AWS, si aggiungono più controlli a diversi livelli della AWS Organizations struttura per proteggere le risorse. Ad esempio, un defense-in-depth approccio potrebbe combinare l'autenticazione a più fattori, la segmentazione della rete e la crittografia.

amministratore delegato

In AWS Organizations, un servizio compatibile può registrare un account AWS membro per amministrare gli account dell'organizzazione e gestire le autorizzazioni per quel servizio. Questo account è denominato amministratore delegato per quel servizio specifico. Per ulteriori informazioni e un elenco di servizi compatibili, consulta [Servizi che funzionano con AWS Organizations](#) nella documentazione di AWS Organizations .

implementazione

Il processo di creazione di un'applicazione, di nuove funzionalità o di correzioni di codice disponibili nell'ambiente di destinazione. L'implementazione prevede l'applicazione di modifiche in una base di codice, seguita dalla creazione e dall'esecuzione di tale base di codice negli ambienti applicativi.

Ambiente di sviluppo

[Vedi ambiente.](#)

controllo di rilevamento

Un controllo di sicurezza progettato per rilevare, registrare e avvisare dopo che si è verificato un evento. Questi controlli rappresentano una seconda linea di difesa e avvisano l'utente in caso di eventi di sicurezza che aggirano i controlli preventivi in vigore. Per ulteriori informazioni, consulta [Controlli di rilevamento](#) in Implementazione dei controlli di sicurezza in AWS.

mappatura del flusso di valore dello sviluppo (DVSM)

Un processo utilizzato per identificare e dare priorità ai vincoli che influiscono negativamente sulla velocità e sulla qualità nel ciclo di vita dello sviluppo del software. DVSM estende il processo di

mappatura del flusso di valore originariamente progettato per pratiche di produzione snella. Si concentra sulle fasi e sui team necessari per creare e trasferire valore attraverso il processo di sviluppo del software.

gemello digitale

Una rappresentazione virtuale di un sistema reale, ad esempio un edificio, una fabbrica, un'attrezzatura industriale o una linea di produzione. I gemelli digitali supportano la manutenzione predittiva, il monitoraggio remoto e l'ottimizzazione della produzione.

tabella delle dimensioni

In uno [schema a stella](#), una tabella più piccola che contiene gli attributi dei dati quantitativi in una tabella dei fatti. Gli attributi della tabella delle dimensioni sono in genere campi di testo o numeri discreti che si comportano come testo. Questi attributi vengono comunemente utilizzati per il vincolo delle query, il filtraggio e l'etichettatura dei set di risultati.

disastro

Un evento che impedisce a un carico di lavoro o a un sistema di raggiungere gli obiettivi aziendali nella sua sede principale di implementazione. Questi eventi possono essere disastri naturali, guasti tecnici o il risultato di azioni umane, come errori di configurazione involontari o attacchi di malware.

disaster recovery (DR)

La strategia e il processo utilizzati per ridurre al minimo i tempi di inattività e la perdita di dati causati da un [disastro](#). Per ulteriori informazioni, consulta [Disaster Recovery of Workloads su AWS: Recovery in the Cloud in the AWS Well-Architected Framework](#).

DML

Vedi linguaggio di manipolazione [del database](#).

progettazione basata sul dominio

Un approccio allo sviluppo di un sistema software complesso collegandone i componenti a domini in evoluzione, o obiettivi aziendali principali, perseguiti da ciascun componente. Questo concetto è stato introdotto da Eric Evans nel suo libro, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). Per informazioni su come utilizzare la progettazione basata sul dominio con il modello del fico strangolatore (Strangler Fig), consulta la sezione [Modernizzazione incrementale dei servizi Web Microsoft ASP.NET \(ASMX\) legacy utilizzando container e il Gateway Amazon API](#).

DOTT.

Vedi [disaster recovery](#).

rilevamento della deriva

Tracciamento delle deviazioni da una configurazione di base. Ad esempio, è possibile AWS CloudFormation utilizzarlo per [rilevare deviazioni nelle risorse di sistema](#) oppure AWS Control Tower per [rilevare cambiamenti nella landing zone](#) che potrebbero influire sulla conformità ai requisiti di governance.

DVSM

Vedi la [mappatura del flusso di valore dello sviluppo](#).

E

EDA

Vedi [analisi esplorativa dei dati](#).

MODIFICA

Vedi [scambio elettronico di dati](#).

edge computing

La tecnologia che aumenta la potenza di calcolo per i dispositivi intelligenti all'edge di una rete IoT. Rispetto al [cloud computing](#), [l'edge computing](#) può ridurre la latenza di comunicazione e migliorare i tempi di risposta.

scambio elettronico di dati (EDI)

Lo scambio automatizzato di documenti aziendali tra organizzazioni. Per ulteriori informazioni, vedere [Cos'è lo scambio elettronico di dati](#).

crittografia

Un processo di elaborazione che trasforma i dati in chiaro, leggibili dall'uomo, in testo cifrato.

chiave crittografica

Una stringa crittografica di bit randomizzati generata da un algoritmo di crittografia. Le chiavi possono variare di lunghezza e ogni chiave è progettata per essere imprevedibile e univoca.

endianità

L'ordine in cui i byte vengono archiviati nella memoria del computer. I sistemi big-endian memorizzano per primo il byte più importante. I sistemi little-endian memorizzano per primo il byte meno importante.

endpoint

[Vedi](#) service endpoint.

servizio endpoint

Un servizio che puoi ospitare in un cloud privato virtuale (VPC) da condividere con altri utenti. Puoi creare un servizio endpoint con AWS PrivateLink e concedere autorizzazioni ad altri Account AWS o a AWS Identity and Access Management (IAM) principali. Questi account o principali possono connettersi al servizio endpoint in privato creando endpoint VPC di interfaccia. Per ulteriori informazioni, consulta [Creazione di un servizio endpoint](#) nella documentazione di Amazon Virtual Private Cloud (Amazon VPC).

pianificazione delle risorse aziendali (ERP)

Un sistema che automatizza e gestisce i processi aziendali chiave (come contabilità, [MES](#) e gestione dei progetti) per un'azienda.

crittografia envelope

Il processo di crittografia di una chiave di crittografia con un'altra chiave di crittografia. Per ulteriori informazioni, vedete [Envelope encryption](#) nella documentazione AWS Key Management Service (AWS KMS).

ambiente

Un'istanza di un'applicazione in esecuzione. Di seguito sono riportati i tipi di ambiente più comuni nel cloud computing:

- ambiente di sviluppo: un'istanza di un'applicazione in esecuzione disponibile solo per il team principale responsabile della manutenzione dell'applicazione. Gli ambienti di sviluppo vengono utilizzati per testare le modifiche prima di promuoverle negli ambienti superiori. Questo tipo di ambiente viene talvolta definito ambiente di test.
- ambienti inferiori: tutti gli ambienti di sviluppo di un'applicazione, ad esempio quelli utilizzati per le build e i test iniziali.

- ambiente di produzione: un'istanza di un'applicazione in esecuzione a cui gli utenti finali possono accedere. In una CI/CD pipeline, l'ambiente di produzione è l'ultimo ambiente di distribuzione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

Nelle metodologie agili, categorie funzionali che aiutano a organizzare e dare priorità al lavoro. Le epiche forniscono una descrizione di alto livello dei requisiti e delle attività di implementazione. Ad esempio, le epiche della sicurezza AWS CAF includono la gestione delle identità e degli accessi, i controlli investigativi, la sicurezza dell'infrastruttura, la protezione dei dati e la risposta agli incidenti. Per ulteriori informazioni sulle epiche, consulta la strategia di migrazione AWS , consulta la [guida all'implementazione del programma](#).

ERP

Vedi [pianificazione delle risorse aziendali](#).

analisi esplorativa dei dati (EDA)

Il processo di analisi di un set di dati per comprenderne le caratteristiche principali. Si raccolgono o si aggregano dati e quindi si eseguono indagini iniziali per trovare modelli, rilevare anomalie e verificare ipotesi. L'EDA viene eseguita calcolando statistiche di riepilogo e creando visualizzazioni di dati.

F

tabella dei fatti

Il tavolo centrale in uno [schema a stella](#). Memorizza dati quantitativi sulle operazioni aziendali. In genere, una tabella dei fatti contiene due tipi di colonne: quelle che contengono misure e quelle che contengono una chiave esterna per una tabella di dimensioni.

fallire velocemente

Una filosofia che utilizza test frequenti e incrementali per ridurre il ciclo di vita dello sviluppo. È una parte fondamentale di un approccio agile.

limite di isolamento dei guasti

Nel Cloud AWS, un limite come una zona di disponibilità Regione AWS, un piano di controllo o un piano dati che limita l'effetto di un errore e aiuta a migliorare la resilienza dei carichi di lavoro. Per ulteriori informazioni, consulta [AWS Fault Isolation Boundaries](#).

ramo di funzionalità

Vedi [filiale](#).

caratteristiche

I dati di input che usi per fare una previsione. Ad esempio, in un contesto di produzione, le caratteristiche potrebbero essere immagini acquisite periodicamente dalla linea di produzione.

importanza delle caratteristiche

Quanto è importante una caratteristica per le previsioni di un modello. Di solito viene espresso come punteggio numerico che può essere calcolato con varie tecniche, come Shapley Additive Explanations (SHAP) e gradienti integrati. Per ulteriori informazioni, consulta [Interpretabilità del modello di machine learning con AWS](#).

trasformazione delle funzionalità

Per ottimizzare i dati per il processo di machine learning, incluso l'arricchimento dei dati con fonti aggiuntive, il dimensionamento dei valori o l'estrazione di più set di informazioni da un singolo campo di dati. Ciò consente al modello di ML di trarre vantaggio dai dati. Ad esempio, se suddividi la data "2021-05-27 00:15:37" in "2021", "maggio", "giovedì" e "15", puoi aiutare l'algoritmo di apprendimento ad apprendere modelli sfumati associati a diversi componenti dei dati.

prompt con pochi scatti

Fornire a un [LLM](#) un numero limitato di esempi che dimostrino l'attività e il risultato desiderato prima di chiedergli di eseguire un'attività simile. Questa tecnica è un'applicazione dell'apprendimento contestuale, in cui i modelli imparano da esempi (immagini) incorporati nei prompt. I prompt con pochi passaggi possono essere efficaci per attività che richiedono una formattazione, un ragionamento o una conoscenza del dominio specifici. [Vedi anche zero-shot prompting](#).

FGAC

Vedi il controllo [granulare degli accessi](#).

controllo granulare degli accessi (FGAC)

L'uso di più condizioni per consentire o rifiutare una richiesta di accesso.

migrazione flash-cut

Un metodo di migrazione del database che utilizza la replica continua dei dati tramite l'[acquisizione dei dati delle modifiche](#) per migrare i dati nel più breve tempo possibile, anziché utilizzare un approccio graduale. L'obiettivo è ridurre al minimo i tempi di inattività.

FM

[Vedi modello di base.](#)

modello di fondazione (FM)

Una grande rete neurale di deep learning che si è addestrata su enormi set di dati generalizzati e non etichettati. FMs sono in grado di svolgere un'ampia varietà di attività generali, come comprendere il linguaggio, generare testo e immagini e conversare in linguaggio naturale. Per ulteriori informazioni, consulta [Cosa sono i modelli Foundation](#).

G

IA generativa

Un sottoinsieme di modelli di [intelligenza artificiale](#) che sono stati addestrati su grandi quantità di dati e che possono utilizzare un semplice messaggio di testo per creare nuovi contenuti e artefatti, come immagini, video, testo e audio. Per ulteriori informazioni, consulta [Cos'è l'IA generativa](#).

blocco geografico

Vedi [restrizioni geografiche](#).

limitazioni geografiche (blocco geografico)

In Amazon CloudFront, un'opzione per impedire agli utenti di determinati paesi di accedere alle distribuzioni di contenuti. Puoi utilizzare un elenco consentito o un elenco di blocco per specificare i paesi approvati e vietati. Per ulteriori informazioni, consulta [Limitare la distribuzione geografica dei contenuti](#) nella CloudFront documentazione.

Flusso di lavoro di GitFlow

Un approccio in cui gli ambienti inferiori e superiori utilizzano rami diversi in un repository di codice di origine. Il flusso di lavoro Gitflow è considerato obsoleto e il flusso di lavoro [basato su trunk è l'approccio moderno e preferito](#).

immagine dorata

Un'istantanea di un sistema o di un software utilizzata come modello per distribuire nuove istanze di quel sistema o software. Ad esempio, nella produzione, un'immagine dorata può essere utilizzata per fornire software su più dispositivi e contribuire a migliorare la velocità, la scalabilità e la produttività nelle operazioni di produzione dei dispositivi.

strategia greenfield

L'assenza di infrastrutture esistenti in un nuovo ambiente. Quando si adotta una strategia greenfield per un'architettura di sistema, è possibile selezionare tutte le nuove tecnologie senza il vincolo della compatibilità con l'infrastruttura esistente, nota anche come [brownfield](#). Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e greenfield.

guardrail

Una regola di alto livello che aiuta a governare le risorse, le politiche e la conformità tra le unità organizzative (). OUs I guardrail preventivi applicano le policy per garantire l'allineamento agli standard di conformità. Vengono implementati utilizzando le policy di controllo dei servizi e i limiti delle autorizzazioni IAM. I guardrail di rilevamento rilevano le violazioni delle policy e i problemi di conformità e generano avvisi per porvi rimedio. Sono implementati utilizzando Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector e controlli personalizzati AWS Lambda .

H

AH

Vedi [disponibilità elevata](#).

migrazione di database eterogenea

Migrazione del database di origine in un database di destinazione che utilizza un motore di database diverso (ad esempio, da Oracle ad Amazon Aurora). La migrazione eterogenea fa in

genere parte di uno sforzo di riprogettazione e la conversione dello schema può essere un'attività complessa. [AWS offre AWS SCT](#) che aiuta con le conversioni dello schema.

alta disponibilità (HA)

La capacità di un carico di lavoro di funzionare in modo continuo, senza intervento, in caso di sfide o disastri. I sistemi HA sono progettati per il failover automatico, fornire costantemente prestazioni di alta qualità e gestire carichi e guasti diversi con un impatto minimo sulle prestazioni.

modernizzazione storica

Un approccio utilizzato per modernizzare e aggiornare i sistemi di tecnologia operativa (OT) per soddisfare meglio le esigenze dell'industria manifatturiera. Uno storico è un tipo di database utilizzato per raccogliere e archiviare dati da varie fonti in una fabbrica.

dati di blocco

[Una parte di dati storici etichettati che viene trattenuta da un set di dati utilizzata per addestrare un modello di apprendimento automatico.](#) È possibile utilizzare i dati di holdout per valutare le prestazioni del modello confrontando le previsioni del modello con i dati di holdout.

migrazione di database omogenea

Migrazione del database di origine in un database di destinazione che condivide lo stesso motore di database (ad esempio, da Microsoft SQL Server ad Amazon RDS per SQL Server). La migrazione omogenea fa in genere parte di un'operazione di rehosting o ridefinizione della piattaforma. Per migrare lo schema è possibile utilizzare le utilità native del database.

dati caldi

Dati a cui si accede frequentemente, come dati in tempo reale o dati di traduzione recenti. Questi dati richiedono in genere un livello o una classe di storage ad alte prestazioni per fornire risposte rapide alle query.

hotfix

Una soluzione urgente per un problema critico in un ambiente di produzione. A causa della sua urgenza, un hotfix viene in genere creato al di fuori del tipico DevOps flusso di lavoro di rilascio.

periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura

da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

IaC

Vedi l'[infrastruttura come codice](#).

Policy basata su identità

Una policy associata a uno o più principi IAM che definisce le relative autorizzazioni all'interno dell'Cloud AWS ambiente.

applicazione inattiva

Un'applicazione che prevede un uso di CPU e memoria medio compreso tra il 5% e il 20% in un periodo di 90 giorni. In un progetto di migrazione, è normale ritirare queste applicazioni o mantenerle on-premise.

IloT

Vedi [Industrial Internet of Things](#).

infrastruttura immutabile

Un modello che implementa una nuova infrastruttura per i carichi di lavoro di produzione anziché aggiornare, applicare patch o modificare l'infrastruttura esistente. [Le infrastrutture immutabili sono intrinsecamente più coerenti, affidabili e prevedibili delle infrastrutture mutabili](#). Per ulteriori informazioni, consulta la best practice [Deploy using immutable infrastructure in Well-Architected AWS Framework](#).

VPC in ingresso (ingresso)

In un'architettura AWS multi-account, un VPC che accetta, ispeziona e indirizza le connessioni di rete dall'esterno di un'applicazione. La [AWS Security Reference Architecture](#) consiglia di configurare l'account di rete con funzionalità in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

migrazione incrementale

Una strategia di conversione in cui si esegue la migrazione dell'applicazione in piccole parti anziché eseguire una conversione singola e completa. Ad esempio, inizialmente potresti spostare

I

solo alcuni microservizi o utenti nel nuovo sistema. Dopo aver verificato che tutto funzioni correttamente, puoi spostare in modo incrementale microservizi o utenti aggiuntivi fino alla disattivazione del sistema legacy. Questa strategia riduce i rischi associati alle migrazioni di grandi dimensioni.

Industria 4.0

Un termine introdotto da [Klaus Schwab](#) nel 2016 per riferirsi alla modernizzazione dei processi di produzione attraverso progressi in termini di connettività, dati in tempo reale, automazione, analisi e AI/ML.

infrastruttura

Tutte le risorse e gli asset contenuti nell'ambiente di un'applicazione.

infrastruttura come codice (IaC)

Il processo di provisioning e gestione dell'infrastruttura di un'applicazione tramite un insieme di file di configurazione. Il processo IaC è progettato per aiutarti a centralizzare la gestione dell'infrastruttura, a standardizzare le risorse e a dimensionare rapidamente, in modo che i nuovi ambienti siano ripetibili, affidabili e coerenti.

IIoInternet delle cose industriale (T)

L'uso di sensori e dispositivi connessi a Internet nei settori industriali, come quello manifatturiero, energetico, automobilistico, sanitario, delle scienze della vita e dell'agricoltura. Per ulteriori informazioni, vedere [Creazione di una strategia di trasformazione digitale per l'Internet of Things \(IIoT\) industriale](#).

VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPCs (nello stesso o in modo diverso Regioni AWS), Internet e le reti locali. La [AWS Security Reference Architecture](#) consiglia di configurare l'account di rete con informazioni in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

Internet of Things (IoT)

La rete di oggetti fisici connessi con sensori o processori incorporati che comunicano con altri dispositivi e sistemi tramite Internet o una rete di comunicazione locale. Per ulteriori informazioni, consulta [Cos'è l'IoT?](#)

interpretabilità

Una caratteristica di un modello di machine learning che descrive il grado in cui un essere umano è in grado di comprendere in che modo le previsioni del modello dipendono dai suoi input. Per ulteriori informazioni, vedere Interpretabilità del modello di [machine learning](#) con AWS

IoT

Vedi [Internet of Things](#).

libreria di informazioni IT (ITIL)

Una serie di best practice per offrire servizi IT e allinearli ai requisiti aziendali. ITIL fornisce le basi per ITSM.

gestione dei servizi IT (ITSM)

Attività associate alla progettazione, implementazione, gestione e supporto dei servizi IT per un'organizzazione. Per informazioni sull'integrazione delle operazioni cloud con gli strumenti ITSM, consulta la [guida all'integrazione delle operazioni](#).

ITIL

Vedi la [libreria di informazioni IT](#).

ITSM

Vedi [Gestione dei servizi IT](#).

L

controllo degli accessi basato su etichette (LBAC)

Un'implementazione del controllo di accesso obbligatorio (MAC) in cui agli utenti e ai dati stessi viene assegnato esplicitamente un valore di etichetta di sicurezza. L'intersezione tra l'etichetta di sicurezza utente e l'etichetta di sicurezza dei dati determina quali righe e colonne possono essere visualizzate dall'utente.

zona di destinazione

Una landing zone è un AWS ambiente multi-account ben progettato, scalabile e sicuro. Questo è un punto di partenza dal quale le organizzazioni possono avviare e distribuire rapidamente carichi di lavoro e applicazioni con fiducia nel loro ambiente di sicurezza e infrastruttura. Per ulteriori

informazioni sulle zone di destinazione, consulta la sezione [Configurazione di un ambiente AWS multi-account sicuro e scalabile](#).

modello linguistico di grandi dimensioni (LLM)

Un modello di [intelligenza artificiale](#) di deep learning preaddestrato su una grande quantità di dati. Un LLM può svolgere più attività, come rispondere a domande, riepilogare documenti, tradurre testo in altre lingue e completare frasi. [Per ulteriori informazioni, consulta Cosa sono. LLMs](#)

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

Vedi controllo degli accessi [basato su etichette](#).

Privilegio minimo

La best practice di sicurezza per la concessione delle autorizzazioni minime richieste per eseguire un'attività. Per ulteriori informazioni, consulta [Applicazione delle autorizzazioni del privilegio minimo](#) nella documentazione di IAM.

eseguire il rehosting (lift and shift)

Vedi [7 R](#).

sistema little-endian

Un sistema che memorizza per primo il byte meno importante. Vedi anche [endianità](#).

LLM

Vedi modello [linguistico di grandi dimensioni](#).

ambienti inferiori

Vedi [ambiente](#).

M

machine learning (ML)

Un tipo di intelligenza artificiale che utilizza algoritmi e tecniche per il riconoscimento e l'apprendimento di schemi. Il machine learning analizza e apprende dai dati registrati, come i dati

dell'Internet delle cose (IoT), per generare un modello statistico basato su modelli. Per ulteriori informazioni, consulta la sezione [Machine learning](#).

ramo principale

Vedi [filiale](#).

malware

Software progettato per compromettere la sicurezza o la privacy del computer. Il malware potrebbe interrompere i sistemi informatici, divulgare informazioni sensibili o ottenere accessi non autorizzati. Esempi di malware includono virus, worm, ransomware, trojan horse, spyware e keylogger.

servizi gestiti

Servizi AWS per cui AWS gestisce il livello di infrastruttura, il sistema operativo e le piattaforme e si accede agli endpoint per archiviare e recuperare i dati. Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) e Amazon DynamoDB sono esempi di servizi gestiti. Questi sono noti anche come servizi astratti.

sistema di esecuzione della produzione (MES)

Un sistema software per tracciare, monitorare, documentare e controllare i processi di produzione che convertono le materie prime in prodotti finiti in officina.

MAP

Vedi [Migration Acceleration Program](#).

meccanismo

Un processo completo in cui si crea uno strumento, si promuove l'adozione dello strumento e quindi si esaminano i risultati per apportare le modifiche. Un meccanismo è un ciclo che si rafforza e si migliora man mano che funziona. Per ulteriori informazioni, consulta [Creazione di meccanismi nel AWS Well-Architected Framework](#).

account membro

Tutti gli account Account AWS diversi dall'account di gestione che fanno parte di un'organizzazione in. AWS Organizations Un account può essere membro di una sola organizzazione alla volta.

MEH

Vedi [sistema di esecuzione della produzione](#).

Message Queuing Telemetry Transport (MQTT)

[Un protocollo di comunicazione machine-to-machine \(M2M\) leggero, basato sul modello di pubblicazione/sottoscrizione, per dispositivi IoT con risorse limitate.](#)

microservizio

Un servizio piccolo e indipendente che comunica tramite canali ben definiti ed è in genere di proprietà di piccoli team autonomi. APIs Ad esempio, un sistema assicurativo potrebbe includere microservizi che si riferiscono a funzionalità aziendali, come vendite o marketing, o sottodomini, come acquisti, reclami o analisi. I vantaggi dei microservizi includono agilità, dimensionamento flessibile, facilità di implementazione, codice riutilizzabile e resilienza. Per ulteriori informazioni, consulta [Integrazione dei microservizi utilizzando servizi serverless](#). AWS

architettura di microservizi

Un approccio alla creazione di un'applicazione con componenti indipendenti che eseguono ogni processo applicativo come microservizio. Questi microservizi comunicano attraverso un'interfaccia ben definita utilizzando sistemi leggeri. APIs Ogni microservizio in questa architettura può essere aggiornato, distribuito e dimensionato per soddisfare la richiesta di funzioni specifiche di un'applicazione. Per ulteriori informazioni, vedere [Implementazione dei microservizi](#) su. AWS

Programma di accelerazione della migrazione (MAP)

Un AWS programma che fornisce consulenza, supporto, formazione e servizi per aiutare le organizzazioni a costruire una solida base operativa per il passaggio al cloud e per contribuire a compensare il costo iniziale delle migrazioni. MAP include una metodologia di migrazione per eseguire le migrazioni precedenti in modo metodico e un set di strumenti per automatizzare e accelerare gli scenari di migrazione comuni.

migrazione su larga scala

Il processo di trasferimento della maggior parte del portfolio di applicazioni sul cloud avviene a ondate, con più applicazioni trasferite a una velocità maggiore in ogni ondata. Questa fase utilizza le migliori pratiche e le lezioni apprese nelle fasi precedenti per implementare una fabbrica di migrazione di team, strumenti e processi per semplificare la migrazione dei carichi di lavoro attraverso l'automazione e la distribuzione agile. Questa è la terza fase della [strategia di migrazione AWS](#).

fabbrica di migrazione

Team interfunzionali che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory in genere includono addetti alle operazioni,

analisti e proprietari aziendali, ingegneri addetti alla migrazione, sviluppatori e DevOps professionisti che lavorano nell'ambito degli sprint. Tra il 20% e il 50% di un portfolio di applicazioni aziendali è costituito da schemi ripetuti che possono essere ottimizzati con un approccio di fabbrica. Per ulteriori informazioni, consulta la [discussione sulle fabbriche di migrazione](#) e la [Guida alla fabbrica di migrazione al cloud](#) in questo set di contenuti.

metadati di migrazione

Le informazioni sull'applicazione e sul server necessarie per completare la migrazione. Ogni modello di migrazione richiede un set diverso di metadati di migrazione. Esempi di metadati di migrazione includono la sottorete, il gruppo di sicurezza e l'account di destinazione. AWS

modello di migrazione

Un'attività di migrazione ripetibile che descrive in dettaglio la strategia di migrazione, la destinazione della migrazione e l'applicazione o il servizio di migrazione utilizzati. Esempio: riorganizza la migrazione su Amazon EC2 AWS con Application Migration Service.

Valutazione del portfolio di migrazione (MPA)

Uno strumento online che fornisce informazioni per la convalida del business case per la migrazione a. Cloud AWS MPA offre una valutazione dettagliata del portfolio (dimensionamento corretto dei server, prezzi, confronto del TCO, analisi dei costi di migrazione) e pianificazione della migrazione (analisi e raccolta dei dati delle applicazioni, raggruppamento delle applicazioni, prioritizzazione delle migrazioni e pianificazione delle ondate). [Lo strumento MPA](#) (richiede l'accesso) è disponibile gratuitamente per tutti i AWS consulenti e i consulenti dei partner APN.

valutazione della preparazione alla migrazione (MRA)

Il processo di acquisizione di informazioni sullo stato di preparazione al cloud di un'organizzazione, l'identificazione dei punti di forza e di debolezza e la creazione di un piano d'azione per colmare le lacune identificate, utilizzando il CAF. AWS Per ulteriori informazioni, consulta la [guida di preparazione alla migrazione](#). MRA è la prima fase della [strategia di migrazione AWS](#).

strategia di migrazione

L'approccio utilizzato per migrare un carico di lavoro verso. Cloud AWS Per ulteriori informazioni, consulta la voce [7 R](#) in questo glossario e consulta [Mobilita la tua organizzazione per](#) accelerare le migrazioni su larga scala.

ML

[Vedi machine learning.](#)

modernizzazione

Trasformazione di un'applicazione obsoleta (legacy o monolitica) e della relativa infrastruttura in un sistema agile, elastico e altamente disponibile nel cloud per ridurre i costi, aumentare l'efficienza e sfruttare le innovazioni. Per ulteriori informazioni, vedere [Strategia per la modernizzazione delle applicazioni in](#). Cloud AWS

valutazione della preparazione alla modernizzazione

Una valutazione che aiuta a determinare la preparazione alla modernizzazione delle applicazioni di un'organizzazione, identifica vantaggi, rischi e dipendenze e determina in che misura l'organizzazione può supportare lo stato futuro di tali applicazioni. Il risultato della valutazione è uno schema dell'architettura di destinazione, una tabella di marcia che descrive in dettaglio le fasi di sviluppo e le tappe fondamentali del processo di modernizzazione e un piano d'azione per colmare le lacune identificate. Per ulteriori informazioni, vedere [Valutazione della preparazione alla modernizzazione per](#) le applicazioni in. Cloud AWS

applicazioni monolitiche (monoliti)

Applicazioni eseguite come un unico servizio con processi strettamente collegati. Le applicazioni monolitiche presentano diversi inconvenienti. Se una funzionalità dell'applicazione registra un picco di domanda, l'intera architettura deve essere dimensionata. L'aggiunta o il miglioramento delle funzionalità di un'applicazione monolitica diventa inoltre più complessa man mano che la base di codice cresce. Per risolvere questi problemi, puoi utilizzare un'architettura di microservizi. Per ulteriori informazioni, consulta la sezione [Scomposizione dei monoliti in microservizi](#).

MAPPA

Vedi [Migration Portfolio Assessment](#).

MQTT

Vedi [Message Queuing Telemetry](#) Transport.

classificazione multiclasse

Un processo che aiuta a generare previsioni per più classi (prevedendo uno o più di due risultati). Ad esempio, un modello di machine learning potrebbe chiedere "Questo prodotto è un libro, un'auto o un telefono?" oppure "Quale categoria di prodotti è più interessante per questo cliente?"

infrastruttura mutabile

Un modello che aggiorna e modifica l'infrastruttura esistente per i carichi di lavoro di produzione. Per migliorare la coerenza, l'affidabilità e la prevedibilità, il AWS Well-Architected Framework consiglia l'uso di un'infrastruttura [immutabile](#) come best practice.

O

OAC

Vedi [Origin Access Control](#).

QUERCIA

Vedi [Origin Access Identity](#).

OCM

Vedi [gestione delle modifiche organizzative](#).

migrazione offline

Un metodo di migrazione in cui il carico di lavoro di origine viene eliminato durante il processo di migrazione. Questo metodo prevede tempi di inattività prolungati e viene in genere utilizzato per carichi di lavoro piccoli e non critici.

OI

Vedi [l'integrazione delle operazioni](#).

OLA

Vedi accordo a [livello operativo](#).

migrazione online

Un metodo di migrazione in cui il carico di lavoro di origine viene copiato sul sistema di destinazione senza essere messo offline. Le applicazioni connesse al carico di lavoro possono continuare a funzionare durante la migrazione. Questo metodo comporta tempi di inattività pari a zero o comunque minimi e viene in genere utilizzato per carichi di lavoro di produzione critici.

OPC-UA

Vedi [Open Process Communications - Unified Architecture](#).

Comunicazioni a processo aperto - Architettura unificata (OPC-UA)

Un protocollo di comunicazione machine-to-machine (M2M) per l'automazione industriale. OPC-UA fornisce uno standard di interoperabilità con schemi di crittografia, autenticazione e autorizzazione dei dati.

accordo a livello operativo (OLA)

Un accordo che chiarisce quali sono gli impegni reciproci tra i gruppi IT funzionali, a supporto di un accordo sul livello di servizio (SLA).

revisione della prontezza operativa (ORR)

Un elenco di domande e best practice associate che aiutano a comprendere, valutare, prevenire o ridurre la portata degli incidenti e dei possibili guasti. Per ulteriori informazioni, vedere [Operational Readiness Reviews \(ORR\)](#) nel Well-Architected AWS Framework.

tecnologia operativa (OT)

Sistemi hardware e software che interagiscono con l'ambiente fisico per controllare le operazioni, le apparecchiature e le infrastrutture industriali. Nella produzione, l'integrazione di sistemi OT e di tecnologia dell'informazione (IT) è un obiettivo chiave per le trasformazioni [dell'Industria 4.0](#).

integrazione delle operazioni (OI)

Il processo di modernizzazione delle operazioni nel cloud, che prevede la pianificazione, l'automazione e l'integrazione della disponibilità. Per ulteriori informazioni, consulta la [guida all'integrazione delle operazioni](#).

trail organizzativo

Un percorso creato da noi AWS CloudTrail che registra tutti gli eventi di un'organizzazione per tutti Account AWS . AWS Organizations Questo percorso viene creato in ogni Account AWS che fa parte dell'organizzazione e tiene traccia dell'attività in ogni account. Per ulteriori informazioni, consulta [Creazione di un percorso per un'organizzazione](#) nella CloudTrail documentazione.

gestione del cambiamento organizzativo (OCM)

Un framework per la gestione di trasformazioni aziendali importanti e che comportano l'interruzione delle attività dal punto di vista delle persone, della cultura e della leadership. OCM aiuta le organizzazioni a prepararsi e passare a nuovi sistemi e strategie accelerando l'adozione del cambiamento, affrontando i problemi di transizione e promuovendo cambiamenti culturali e organizzativi. Nella strategia di AWS migrazione, questo framework si chiama accelerazione delle

persone, a causa della velocità di cambiamento richiesta nei progetti di adozione del cloud. Per ulteriori informazioni, consultare la [Guida OCM](#).

controllo dell'accesso all'origine (OAC)

In CloudFront, un'opzione avanzata per limitare l'accesso per proteggere i contenuti di Amazon Simple Storage Service (Amazon S3). OAC supporta tutti i bucket S3 in generale Regioni AWS, la crittografia lato server con AWS KMS (SSE-KMS) e le richieste dinamiche e dirette al bucket S3.

PUT DELETE

identità di accesso origine (OAI)

Nel CloudFront, un'opzione per limitare l'accesso per proteggere i tuoi contenuti Amazon S3. Quando usi OAI, CloudFront crea un principale con cui Amazon S3 può autenticarsi. I principali autenticati possono accedere ai contenuti in un bucket S3 solo tramite una distribuzione specifica. CloudFront Vedi anche [OAC](#), che fornisce un controllo degli accessi più granulare e avanzato.

ORR

[Vedi la revisione della prontezza operativa.](#)

NON

Vedi la [tecnologia operativa](#).

VPC in uscita (egress)

In un'architettura AWS multi-account, un VPC che gestisce le connessioni di rete avviate dall'interno di un'applicazione. La [AWS Security Reference Architecture](#) consiglia di configurare l'account di rete con funzionalità in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

P

limite delle autorizzazioni

Una policy di gestione IAM collegata ai principali IAM per impostare le autorizzazioni massime che l'utente o il ruolo possono avere. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni](#) nella documentazione di IAM.

informazioni di identificazione personale (PII)

Informazioni che, se visualizzate direttamente o abbinate ad altri dati correlati, possono essere utilizzate per dedurre ragionevolmente l'identità di un individuo. Esempi di informazioni personali includono nomi, indirizzi e informazioni di contatto.

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le [informazioni di identificazione personale](#).

playbook

Una serie di passaggi predefiniti che raccolgono il lavoro associato alle migrazioni, come l'erogazione delle funzioni operative principali nel cloud. Un playbook può assumere la forma di script, runbook automatici o un riepilogo dei processi o dei passaggi necessari per gestire un ambiente modernizzato.

PLC

Vedi [controllore logico programmabile](#).

PLM

Vedi la gestione [del ciclo di vita del prodotto](#).

policy

[Un oggetto in grado di definire le autorizzazioni \(vedi politica basata sull'identità\), specificare le condizioni di accesso \(vedi politicabasata sulle risorse\) o definire le autorizzazioni massime per tutti gli account di un'organizzazione in \(vedi politica di controllo dei servizi\). AWS Organizations](#)

persistenza poliglotta

Scelta indipendente della tecnologia di archiviazione di dati di un microservizio in base ai modelli di accesso ai dati e ad altri requisiti. Se i microservizi utilizzano la stessa tecnologia di archiviazione di dati, possono incontrare problemi di implementazione o registrare prestazioni scadenti. I microservizi vengono implementati più facilmente e ottengono prestazioni e scalabilità migliori se utilizzano l'archivio dati più adatto alle loro esigenze.

valutazione del portfolio

Un processo di scoperta, analisi e definizione delle priorità del portfolio di applicazioni per pianificare la migrazione. Per ulteriori informazioni, consulta la pagina [Valutazione della preparazione alla migrazione](#).

predicate

Una condizione di interrogazione che restituisce o, in genere, si trova in una clausola `true`. `false`
`WHERE`

predicato pushdown

Una tecnica di ottimizzazione delle query del database che filtra i dati della query prima del trasferimento. Ciò riduce la quantità di dati che devono essere recuperati ed elaborati dal database relazionale e migliora le prestazioni delle query.

controllo preventivo

Un controllo di sicurezza progettato per impedire il verificarsi di un evento. Questi controlli sono la prima linea di difesa per impedire accessi non autorizzati o modifiche indesiderate alla rete. Per ulteriori informazioni, consulta [Controlli preventivi](#) in Implementazione dei controlli di sicurezza in AWS.

principale

Un'entità in AWS grado di eseguire azioni e accedere alle risorse. Questa entità è in genere un utente root per un Account AWS ruolo IAM o un utente. Per ulteriori informazioni, consulta Principali in [Termini e concetti dei ruoli](#) nella documentazione di IAM.

privacy fin dalla progettazione

Un approccio di ingegneria dei sistemi che tiene conto della privacy durante l'intero processo di sviluppo.

zone ospitate private

Un contenitore che contiene informazioni su come desideri che Amazon Route 53 risponda alle query DNS per un dominio e i relativi sottodomini all'interno di uno o più VPCs Per ulteriori informazioni, consulta [Utilizzo delle zone ospitate private](#) nella documentazione di Route 53.

controllo proattivo

Un [controllo di sicurezza](#) progettato per impedire l'implementazione di risorse non conformi. Questi controlli analizzano le risorse prima del loro provisioning. Se la risorsa non è conforme al controllo, non viene fornita. Per ulteriori informazioni, consulta la [guida di riferimento sui controlli](#) nella AWS Control Tower documentazione e consulta Controlli [proattivi in Implementazione dei controlli](#) di sicurezza su. AWS

gestione del ciclo di vita del prodotto (PLM)

La gestione dei dati e dei processi di un prodotto durante l'intero ciclo di vita, dalla progettazione, sviluppo e lancio, attraverso la crescita e la maturità, fino al declino e alla rimozione.

Ambiente di produzione

[Vedi ambiente.](#)

controllore logico programmabile (PLC)

Nella produzione, un computer altamente affidabile e adattabile che monitora le macchine e automatizza i processi di produzione.

concatenamento rapido

Utilizzo dell'output di un prompt [LLM](#) come input per il prompt successivo per generare risposte migliori. Questa tecnica viene utilizzata per suddividere un'attività complessa in sottoattività o per perfezionare o espandere iterativamente una risposta preliminare. Aiuta a migliorare l'accuratezza e la pertinenza delle risposte di un modello e consente risultati più granulari e personalizzati.

pseudonimizzazione

Il processo di sostituzione degli identificatori personali in un set di dati con valori segnaposto. La pseudonimizzazione può aiutare a proteggere la privacy personale. I dati pseudonimizzati sono ancora considerati dati personali.

publish/subscribe (pub/sub)

Un modello che consente comunicazioni asincrone tra microservizi per migliorare la scalabilità e la reattività. Ad esempio, in un [MES](#) basato su microservizi, un microservizio può pubblicare messaggi di eventi su un canale a cui altri microservizi possono abbonarsi. Il sistema può aggiungere nuovi microservizi senza modificare il servizio di pubblicazione.

Q

Piano di query

Una serie di passaggi, come le istruzioni, utilizzati per accedere ai dati in un sistema di database relazionale SQL.

regressione del piano di query

Quando un ottimizzatore del servizio di database sceglie un piano non ottimale rispetto a prima di una determinata modifica all'ambiente di database. Questo può essere causato da modifiche a statistiche, vincoli, impostazioni dell'ambiente, associazioni dei parametri di query e aggiornamenti al motore di database.

R

Matrice RACI

Vedi [responsabile, responsabile, consultato, informato \(RACI\)](#).

RAG

Vedi [Retrieval](#) Augmented Generation.

ransomware

Un software dannoso progettato per bloccare l'accesso a un sistema informatico o ai dati fino a quando non viene effettuato un pagamento.

Matrice RASCI

Vedi [responsabile, responsabile, consultato, informato \(RACI\)](#).

RCAC

Vedi controllo dell'[accesso a righe e colonne](#).

replica di lettura

Una copia di un database utilizzata per scopi di sola lettura. È possibile indirizzare le query alla replica di lettura per ridurre il carico sul database principale.

riprogettare

Vedi [7 Rs](#).

obiettivo del punto di ripristino (RPO)

Il periodo di tempo massimo accettabile dall'ultimo punto di ripristino dei dati. Questo determina ciò che si considera una perdita di dati accettabile tra l'ultimo punto di ripristino e l'interruzione del servizio.

obiettivo del tempo di ripristino (RTO)

Il ritardo massimo accettabile tra l'interruzione del servizio e il ripristino del servizio.

rifattorizzare

Vedi [7 R.](#)

Region

Una raccolta di AWS risorse in un'area geografica. Ciascuna Regione AWS è isolata e indipendente dalle altre per fornire tolleranza agli errori, stabilità e resilienza. Per ulteriori informazioni, consulta [Specificare cosa può usare Regioni AWS il tuo account.](#)

regressione

Una tecnica di ML che prevede un valore numerico. Ad esempio, per risolvere il problema "A che prezzo verrà venduta questa casa?" un modello di ML potrebbe utilizzare un modello di regressione lineare per prevedere il prezzo di vendita di una casa sulla base di dati noti sulla casa (ad esempio, la metratura).

riospitare

Vedi [7 R.](#)

rilascio

In un processo di implementazione, l'atto di promuovere modifiche a un ambiente di produzione.

trasferisco

Vedi [7 Rs.](#)

ripiattaforma

Vedi [7 Rs.](#)

riacquisto

Vedi [7 Rs.](#)

resilienza

La capacità di un'applicazione di resistere alle interruzioni o di ripristinarle. [L'elevata disponibilità e il disaster recovery](#) sono considerazioni comuni quando si pianifica la resilienza in Cloud AWS. [Per ulteriori informazioni, vedere Cloud AWS Resilience.](#)

policy basata su risorse

Una policy associata a una risorsa, ad esempio un bucket Amazon S3, un endpoint o una chiave di crittografia. Questo tipo di policy specifica a quali principali è consentito l'accesso, le azioni supportate e qualsiasi altra condizione che deve essere soddisfatta.

matrice di assegnazione di responsabilità (RACI)

Una matrice che definisce i ruoli e le responsabilità di tutte le parti coinvolte nelle attività di migrazione e nelle operazioni cloud. Il nome della matrice deriva dai tipi di responsabilità definiti nella matrice: responsabile (R), responsabile (A), consultato (C) e informato (I). Il tipo di supporto (S) è facoltativo. Se includi il supporto, la matrice viene chiamata matrice RASCI e, se la escludi, viene chiamata matrice RACI.

controllo reattivo

Un controllo di sicurezza progettato per favorire la correzione di eventi avversi o deviazioni dalla baseline di sicurezza. Per ulteriori informazioni, consulta [Controlli reattivi](#) in Implementazione dei controlli di sicurezza in AWS.

retain

Vedi [7 R](#).

andare in pensione

Vedi [7 Rs](#).

Retrieval Augmented Generation (RAG)

Una tecnologia di [intelligenza artificiale generativa](#) in cui un [LLM](#) fa riferimento a una fonte di dati autorevole esterna alle sue fonti di dati di formazione prima di generare una risposta. Ad esempio, un modello RAG potrebbe eseguire una ricerca semantica nella knowledge base o nei dati personalizzati di un'organizzazione. Per ulteriori informazioni, consulta [Cos'è il RAG](#).

rotazione

Processo di aggiornamento periodico di un [segreto](#) per rendere più difficile l'accesso alle credenziali da parte di un utente malintenzionato.

controllo dell'accesso a righe e colonne (RCAC)

L'uso di espressioni SQL di base e flessibili con regole di accesso definite. RCAC è costituito da autorizzazioni di riga e maschere di colonna.

RPO

Vedi [obiettivo del punto di ripristino](#).

VERSO

Vedi [obiettivo del tempo di ripristino](#).

runbook

Un insieme di procedure manuali o automatizzate necessarie per eseguire un'attività specifica. In genere sono progettati per semplificare operazioni o procedure ripetitive con tassi di errore elevati.

S

SAML 2.0

Uno standard aperto utilizzato da molti provider di identità (IdPs). Questa funzionalità abilita il single sign-on (SSO) federato, in modo che gli utenti possano accedere Console di gestione AWS o chiamare le operazioni AWS API senza che tu debba creare un utente in IAM per tutti i membri dell'organizzazione. Per ulteriori informazioni sulla federazione basata su SAML 2.0, consulta [Informazioni sulla federazione basata su SAML 2.0](#) nella documentazione di IAM.

SCADA

Vedi [controllo di supervisione e acquisizione dati](#).

SCP

Vedi la [politica di controllo del servizio](#).

Secret

In Gestione dei segreti AWS, informazioni riservate o riservate, come una password o le credenziali utente, archiviate in forma crittografata. È costituito dal valore segreto e dai relativi metadati. Il valore segreto può essere binario, una stringa singola o più stringhe. Per ulteriori informazioni, consulta [Cosa c'è in un segreto di Secrets Manager?](#) nella documentazione di Secrets Manager.

sicurezza fin dalla progettazione

Un approccio di ingegneria dei sistemi che tiene conto della sicurezza durante l'intero processo di sviluppo.

controllo di sicurezza

Un guardrail tecnico o amministrativo che impedisce, rileva o riduce la capacità di un autore di minacce di sfruttare una vulnerabilità di sicurezza. [Esistono quattro tipi principali di controlli di sicurezza: preventivi, investigativi, reattivi e proattivi.](#)

rafforzamento della sicurezza

Il processo di riduzione della superficie di attacco per renderla più resistente agli attacchi. Può includere azioni come la rimozione di risorse che non sono più necessarie, l'implementazione di best practice di sicurezza che prevedono la concessione del privilegio minimo o la disattivazione di funzionalità non necessarie nei file di configurazione.

sistema di gestione delle informazioni e degli eventi di sicurezza (SIEM)

Strumenti e servizi che combinano sistemi di gestione delle informazioni di sicurezza (SIM) e sistemi di gestione degli eventi di sicurezza (SEM). Un sistema SIEM raccoglie, monitora e analizza i dati da server, reti, dispositivi e altre fonti per rilevare minacce e violazioni della sicurezza e generare avvisi.

automazione della risposta alla sicurezza

Un'azione predefinita e programmata progettata per rispondere o porre rimedio automaticamente a un evento di sicurezza. Queste automazioni fungono da controlli di sicurezza [investigativi](#) o [reattivi](#) che aiutano a implementare le migliori pratiche di sicurezza. AWS Esempi di azioni di risposta automatizzate includono la modifica di un gruppo di sicurezza VPC, l'applicazione di patch a un'istanza Amazon EC2 o la rotazione delle credenziali.

Crittografia lato server

Crittografia dei dati a destinazione, da parte di chi li riceve. Servizio AWS

Policy di controllo dei servizi (SCP)

Una politica che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in. AWS Organizations SCPs definire barriere o fissare limiti alle azioni che un amministratore può delegare a utenti o ruoli. È possibile utilizzarli SCPs come elenchi consentiti o elenchi di rifiuto, per specificare quali servizi o azioni sono consentiti o proibiti. Per ulteriori informazioni, consulta [le politiche di controllo del servizio](#) nella AWS Organizations documentazione.

endpoint del servizio

L'URL del punto di ingresso per un Servizio AWS. Puoi utilizzare l'endpoint per connetterti a livello di programmazione al servizio di destinazione. Per ulteriori informazioni, consulta [Endpoint del Servizio AWS](#) nei Riferimenti generali di AWS.

accordo sul livello di servizio (SLA)

Un accordo che chiarisce ciò che un team IT promette di offrire ai propri clienti, ad esempio l'operatività e le prestazioni del servizio.

indicatore del livello di servizio (SLI)

Misurazione di un aspetto prestazionale di un servizio, ad esempio il tasso di errore, la disponibilità o la velocità effettiva.

obiettivo a livello di servizio (SLO)

[Una metrica target che rappresenta lo stato di un servizio, misurato da un indicatore del livello di servizio.](#)

Modello di responsabilità condivisa

Un modello che descrive la responsabilità condivisa AWS per la sicurezza e la conformità del cloud. AWS è responsabile della sicurezza del cloud, mentre tu sei responsabile della sicurezza nel cloud. Per ulteriori informazioni, consulta [Modello di responsabilità condivisa](#).

SIEM

Vedi il [sistema di gestione delle informazioni e degli eventi sulla sicurezza](#).

punto di errore singolo (SPOF)

Un guasto in un singolo componente critico di un'applicazione che può disturbare il sistema.

SLAM

Vedi il contratto sul [livello di servizio](#).

SLI

Vedi l'indicatore del [livello di servizio](#).

LENTA

Vedi obiettivo del [livello di servizio](#).

split-and-seed modello

Un modello per dimensionare e accelerare i progetti di modernizzazione. Man mano che vengono definite nuove funzionalità e versioni dei prodotti, il team principale si divide per creare nuovi team di prodotto. Questo aiuta a dimensionare le capacità e i servizi dell'organizzazione, migliora la produttività degli sviluppatori e supporta una rapida innovazione. Per ulteriori informazioni, vedere [Approccio graduale alla modernizzazione delle applicazioni in](#). Cloud AWS

SPOF

Vedi [punto di errore singolo](#).

schema a stella

Una struttura organizzativa di database che utilizza un'unica tabella dei fatti di grandi dimensioni per archiviare i dati transazionali o misurati e utilizza una o più tabelle dimensionali più piccole per memorizzare gli attributi dei dati. Questa struttura è progettata per l'uso in un [data warehouse](#) o per scopi di business intelligence.

modello del fico strangolatore

Un approccio alla modernizzazione dei sistemi monolitici mediante la riscrittura e la sostituzione incrementali delle funzionalità del sistema fino alla disattivazione del sistema legacy. Questo modello utilizza l'analogia di una pianta di fico che cresce fino a diventare un albero robusto e alla fine annienta e sostituisce il suo ospite. Il modello è stato [introdotto da Martin Fowler](#) come metodo per gestire il rischio durante la riscrittura di sistemi monolitici. Per un esempio di come applicare questo modello, consulta [Modernizzazione incrementale dei servizi Web legacy di Microsoft ASP.NET \(ASMX\) mediante container e Gateway Amazon API](#).

sottorete

Un intervallo di indirizzi IP nel VPC. Una sottorete deve risiedere in una singola zona di disponibilità.

controllo di supervisione e acquisizione dati (SCADA)

Nella produzione, un sistema che utilizza hardware e software per monitorare gli asset fisici e le operazioni di produzione.

crittografia simmetrica

Un algoritmo di crittografia che utilizza la stessa chiave per crittografare e decrittografare i dati.

test sintetici

Test di un sistema in modo da simulare le interazioni degli utenti per rilevare potenziali problemi o monitorare le prestazioni. Puoi usare [Amazon CloudWatch Synthetics](#) per creare questi test.

prompt di sistema

Una tecnica per fornire contesto, istruzioni o linee guida a un [LLM](#) per indirizzarne il comportamento. I prompt di sistema aiutano a impostare il contesto e stabilire regole per le interazioni con gli utenti.

T

tag

Coppie chiave-valore che fungono da metadati per l'organizzazione delle risorse. AWS Con i tag è possibile a gestire, identificare, organizzare, cercare e filtrare le risorse. Per ulteriori informazioni, consulta [Tagging delle risorse AWS](#).

variabile di destinazione

Il valore che stai cercando di prevedere nel machine learning supervisionato. Questo è indicato anche come variabile di risultato. Ad esempio, in un ambiente di produzione la variabile di destinazione potrebbe essere un difetto del prodotto.

elenco di attività

Uno strumento che viene utilizzato per tenere traccia dei progressi tramite un runbook. Un elenco di attività contiene una panoramica del runbook e un elenco di attività generali da completare. Per ogni attività generale, include la quantità stimata di tempo richiesta, il proprietario e lo stato di avanzamento.

ambiente di test

[Vedi ambiente.](#)

training

Fornire dati da cui trarre ispirazione dal modello di machine learning. I dati di training devono contenere la risposta corretta. L'algoritmo di apprendimento trova nei dati di addestramento i pattern che mappano gli attributi dei dati di input al target (la risposta che si desidera prevedere). Produce un modello di ML che acquisisce questi modelli. Puoi quindi utilizzare il modello di ML per creare previsioni su nuovi dati di cui non si conosce il target.

Transit Gateway

Un hub di transito di rete che puoi utilizzare per interconnettere le tue reti VPCs e quelle locali. Per ulteriori informazioni, consulta [Cos'è un gateway di transito](#) nella AWS Transit Gateway documentazione.

flusso di lavoro basato su trunk

Un approccio in cui gli sviluppatori creano e testano le funzionalità localmente in un ramo di funzionalità e quindi uniscono tali modifiche al ramo principale. Il ramo principale viene quindi integrato negli ambienti di sviluppo, preproduzione e produzione, in sequenza.

Accesso attendibile

Concessione delle autorizzazioni a un servizio specificato dall'utente per eseguire attività all'interno dell'organizzazione AWS Organizations e nei suoi account per conto dell'utente. Il servizio attendibile crea un ruolo collegato al servizio in ogni account, quando tale ruolo è necessario, per eseguire attività di gestione per conto dell'utente. Per ulteriori informazioni, consulta [Utilizzo AWS Organizations con altri AWS servizi](#) nella AWS Organizations documentazione.

regolazione

Modificare alcuni aspetti del processo di training per migliorare la precisione del modello di ML. Ad esempio, puoi addestrare il modello di ML generando un set di etichette, aggiungendo etichette e quindi ripetendo questi passaggi più volte con impostazioni diverse per ottimizzare il modello.

team da due pizze

Una piccola DevOps squadra che puoi sfamare con due pizze. Un team composto da due persone garantisce la migliore opportunità possibile di collaborazione nello sviluppo del software.

U

incertezza

Un concetto che si riferisce a informazioni imprecise, incomplete o sconosciute che possono minare l'affidabilità dei modelli di machine learning predittivi. Esistono due tipi di incertezza: l'incertezza epistemica, che è causata da dati limitati e incompleti, mentre l'incertezza aleatoria è causata dal rumore e dalla casualità insiti nei dati. Per ulteriori informazioni, consulta la guida [Quantificazione dell'incertezza nei sistemi di deep learning](#).

compiti indifferenziati

Conosciuto anche come sollevamento di carichi pesanti, è un lavoro necessario per creare e far funzionare un'applicazione, ma che non apporta valore diretto all'utente finale né offre vantaggi competitivi. Esempi di attività indifferenziate includono l'approvvigionamento, la manutenzione e la pianificazione della capacità.

ambienti superiori

[Vedi ambiente.](#)

V

vacuum

Un'operazione di manutenzione del database che prevede la pulizia dopo aggiornamenti incrementali per recuperare lo spazio di archiviazione e migliorare le prestazioni.

controllo delle versioni

Processi e strumenti che tengono traccia delle modifiche, ad esempio le modifiche al codice di origine in un repository.

Peering VPC

Una connessione tra due VPCs che consente di indirizzare il traffico utilizzando indirizzi IP privati. Per ulteriori informazioni, consulta [Che cos'è il peering VPC?](#) nella documentazione di Amazon VPC.

vulnerabilità

Un difetto software o hardware che compromette la sicurezza del sistema.

W

cache calda

Una cache del buffer che contiene dati correnti e pertinenti a cui si accede frequentemente. L'istanza di database può leggere dalla cache del buffer, il che richiede meno tempo rispetto alla lettura dalla memoria dal disco principale.

dati caldi

Dati a cui si accede raramente. Quando si eseguono interrogazioni di questo tipo di dati, in genere sono accettabili query moderatamente lente.

funzione finestra

Una funzione SQL che esegue un calcolo su un gruppo di righe che si riferiscono in qualche modo al record corrente. Le funzioni della finestra sono utili per l'elaborazione di attività, come il calcolo di una media mobile o l'accesso al valore delle righe in base alla posizione relativa della riga corrente.

Carico di lavoro

Una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione rivolta ai clienti o un processo back-end.

flusso di lavoro

Gruppi funzionali in un progetto di migrazione responsabili di una serie specifica di attività. Ogni flusso di lavoro è indipendente ma supporta gli altri flussi di lavoro del progetto. Ad esempio, il flusso di lavoro del portfolio è responsabile della definizione delle priorità delle applicazioni, della pianificazione delle ondate e della raccolta dei metadati di migrazione. Il flusso di lavoro del portfolio fornisce queste risorse al flusso di lavoro di migrazione, che quindi migra i server e le applicazioni.

VERME

Vedi [scrivere una volta, leggere molti](#).

WQF

Vedi [AWS Workload Qualification Framework](#).

scrivi una volta, leggi molte (WORM)

Un modello di storage che scrive i dati una sola volta e ne impedisce l'eliminazione o la modifica. Gli utenti autorizzati possono leggere i dati tutte le volte che è necessario, ma non possono modificarli. Questa infrastruttura di archiviazione dei dati è considerata [immutabile](#).

Z

exploit zero-day

[Un attacco, in genere malware, che sfrutta una vulnerabilità zero-day.](#)

vulnerabilità zero-day

Un difetto o una vulnerabilità assoluta in un sistema di produzione. Gli autori delle minacce possono utilizzare questo tipo di vulnerabilità per attaccare il sistema. Gli sviluppatori vengono spesso a conoscenza della vulnerabilità causata dall'attacco.

prompt zero-shot

Fornire a un [LLM](#) le istruzioni per eseguire un'attività ma non esempi (immagini) che possano aiutarla. Il LLM deve utilizzare le sue conoscenze pre-addestrate per gestire l'attività. L'efficacia del prompt zero-shot dipende dalla complessità dell'attività e dalla qualità del prompt. [Vedi anche few-shot prompting.](#)

applicazione zombie

Un'applicazione che prevede un utilizzo CPU e memoria inferiore al 5%. In un progetto di migrazione, è normale ritirare queste applicazioni.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.