Melhores práticas com o Amazon Q Developer para geração de código em linha e assistente

# AWS Orientação prescritiva

# AWS Orientação prescritiva: Melhores práticas com o Amazon Q Developer para geração de código em linha e assistente

# Table of Contents

# Melhores práticas com o Amazon Q Developer para geração de código em linha e assistente

Amazon Web Services (Colaboradores)

agosto de 2024 () Histórico do documento

Tradicionalmente, os desenvolvedores confiam em sua própria experiência, documentação e trechos de código de várias fontes para escrever e manter o código. Embora esses métodos tenham servido bem ao setor, eles podem ser demorados e propensos a erros humanos, levando a ineficiências e possíveis bugs.

É aqui que o Amazon Q Developer intervém para melhorar a jornada do desenvolvedor. O Amazon Q Developer é um poderoso assistente AWS generativo baseado em IA projetado para acelerar as tarefas de desenvolvimento de código, fornecendo geração e recomendações inteligentes de código.

No entanto, como acontece com qualquer nova tecnologia, pode haver desafios. Expectativas irreais, dificuldades de integração, solução de problemas de geração de código impreciso e uso adequado dos recursos do Amazon Q são obstáculos comuns que os desenvolvedores podem enfrentar. Este guia abrangente aborda esses desafios, fornecendo cenários da vida real, melhores práticas detalhadas, solução de problemas e exemplos práticos de código da vida real especificamente para Python e Java, duas das linguagens de programação mais amplamente adotadas.

Este guia se concentra no uso do Amazon Q Developer para realizar tarefas de desenvolvimento de código, como:

- Preenchimento de código — Gere sugestões em linha à medida que os desenvolvedores programam em tempo real.
- Aconselhamento e melhorias no código — Discuta o desenvolvimento de software, gere novos códigos com linguagem natural e melhore o código existente.

## Objetivos

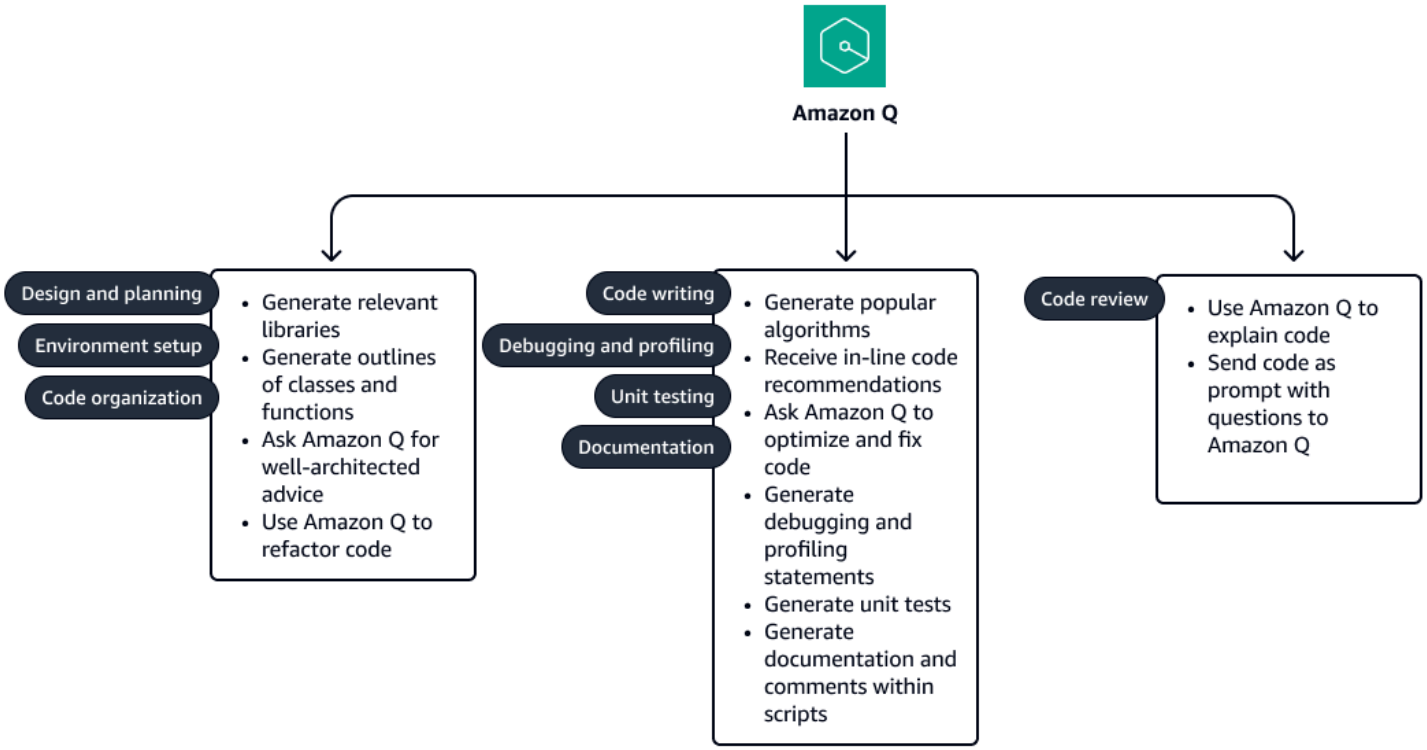
O objetivo deste guia é apoiar desenvolvedores que são usuários novos ou contínuos do Amazon Q Developer, ajudando-os a usar o serviço com sucesso em suas tarefas diárias de codificação. Os gerentes da equipe de desenvolvimento também podem se beneficiar da leitura deste guia.

Este guia fornece os seguintes insights sobre o uso do Amazon Q Developer:

- Entenda o uso efetivo do Amazon Q Developer para desenvolvimento de código
  - Forneça as melhores práticas para integrar o Amazon Q Developer ao fluxo de trabalho de um desenvolvedor.
  - Ofereça step-by-step orientação com exemplos para geração de código e recomendações bem-sucedidas.
- Reduza os desafios comuns e promova a clareza do uso do Amazon Q Developer pelos desenvolvedores
  - Ofereça estratégias e insights para atender às expectativas dos desenvolvedores e superar os obstáculos relacionados à precisão e ao desempenho da geração de código.
- Fornecer solução de problemas e tratamento de erros
  - Equipe os desenvolvedores com a orientação de solução de problemas de geração de código do Amazon Q Developer para lidar com resultados imprecisos ou comportamentos inesperados.
  - Forneça exemplos da vida real e cenários específicos para Python e Java.
- Otimize os fluxos de trabalho e a produtividade
  - Otimize os fluxos de trabalho de desenvolvimento de código com o Amazon Q Developer.
  - Discuta estratégias para aumentar a produtividade do desenvolvedor.

# Usar o Amazon Q Developer em fluxos de trabalho de desenvolvedores

Os desenvolvedores seguem um fluxo de trabalho padrão que abrange os estágios de coleta de requisitos, projeto e planejamento, codificação, teste, revisão de código e implantação. Esta seção se concentra em como você pode usar os recursos do Amazon Q Developer para otimizar as principais etapas de desenvolvimento.




O diagrama anterior mostra como o Amazon Q Developer pode acelerar e simplificar as seguintes tarefas comuns em estágios de desenvolvimento de código:

- Projeto e planejamento | Configuração do ambiente | Organização do código
  - Gere bibliotecas relevantes
  - Gere esboços de classes e funções
  - Peça conselhos bem arquitetados à Amazon Q
  - Usar o Amazon Q para refatorar código
- Redação de código | Depuração e criação de perfil | Teste unitário | Documentação
  - Gere algoritmos populares

  - Receba recomendações sobre código em linha
  - Peça à Amazon Q para otimizar e corrigir o código
  - Gere declarações de depuração e criação de perfil
  - Gerar testes unitários
  - Gere documentação e comentários dentro de scripts
- Revisão de código
  - Peça à Amazon Q para explicar o código
  - Envie o código conforme solicitado com perguntas para a Amazon Q

## Design e planejamento

Depois de reunir os requisitos comerciais e técnicos, os desenvolvedores projetam novas bases de código ou ampliam as existentes. Durante essa fase, o Amazon Q Developer pode ajudar os desenvolvedores a realizar as seguintes tarefas:

- Gere bibliotecas relevantes e esboços de classes e funções para obter conselhos bem arquitetados.
- Forneça orientação para consultas de engenharia, compatibilidade e design arquitetônico.

## Codificação

O processo de codificação usa o Amazon Q Developer para acelerar o desenvolvimento das seguintes formas:

- Configuração do ambiente - Instale o AWS Toolkit em seu ambiente de desenvolvimento integrado (IDE) (por exemplo, VS Code ou IntelliJ). Em seguida, use o Amazon Q para gerar bibliotecas ou receber sugestões de configuração com base nas metas do seu projeto. Para obter mais detalhes, consulte Melhores práticas para integrar o Amazon Q Developer.
- Organização do código - Refatore o código ou obtenha recomendações organizacionais do Amazon Q que estejam alinhadas aos objetivos do seu projeto.
- Redação de código - Use sugestões em linha para gerar código durante o desenvolvimento ou peça ao Amazon Q que gere código usando o painel de bate-papo do Amazon Q em seuIDE. Para obter mais detalhes, consulte Melhores práticas para geração de código com o Amazon Q Developer.

- Depuração e criação de perfil - Gere comandos de criação de perfil ou use as opções do Amazon Q, como Fix e Explain, para depurar problemas.
- Teste unitário — forneça o código como um aviso para a Amazon Q durante uma sessão de bate-papo e solicite a geração de teste unitário aplicável. Para obter mais informações, consulte Exemplos de código com o Amazon Q Developer.
- Documentação - Use sugestões em linha para criar comentários e docstrings, ou use a opção Explain para gerar resumos detalhados para seleções de código. Para obter mais informações, consulte Exemplos de código com o Amazon Q Developer.

## Revisão de código

Os revisores precisam compreender o código de desenvolvimento antes de promovê-lo para produção. Para acelerar esse processo, use as opções Amazon Q Explain and Optimize ou envie seleções de código com instruções personalizadas para o Amazon Q em uma sessão de bate-papo. Para obter mais informações, consulte Exemplos de bate-papo.

## Integração e implantação

Peça orientação ao Amazon Q sobre integração contínua, canais de entrega e melhores práticas de implantação que são específicas para a arquitetura do seu projeto.

Usando essas recomendações, você pode aprender a aproveitar com eficácia os recursos do Amazon Q Developer, otimizando seus fluxos de trabalho e aumentando a produtividade em todo o ciclo de vida do desenvolvimento.

# Capacidades avançadas do Amazon Q Developer

Embora este guia se concentre no uso do Amazon Q Developer em tarefas práticas de programação, é importante estar ciente dos seguintes recursos avançados:

- Transformação de código do Amazon Q Developer
- Personalizações do Amazon Q Developer

## Transformação de código do Amazon Q Developer

O Amazon Q Developer Agent para transformação de código pode atualizar a versão em linguagem de código dos seus arquivos sem a necessidade de reescrever o código manualmente. Ele funciona analisando seus arquivos de código existentes e reescrevendo-os automaticamente para usar uma versão mais recente da linguagem. Por exemplo, o Amazon Q transforma um único módulo se você estiver trabalhando em um IDE como Eclipse. Se você estiver usando o Visual Studio Code, o Amazon Q pode transformar um projeto ou espaço de trabalho inteiro.

Use o Amazon Q quando quiser realizar tarefas comuns de atualização de código, como as seguintes:

- Atualize o código para funcionar com a nova sintaxe da versão do idioma.
- Execute testes de unidade para validar a compilação e a execução bem-sucedidas.
- Verifique e resolva problemas de implantação.

O Amazon Q pode salvar os desenvolvedores de dias a meses de trabalho tedioso e repetitivo para atualizar as bases de código.

A partir de junho de 2024, o Amazon Q Developer oferece suporte à atualização Java código e pode transformar Java 8 códigos para versões mais recentes, como Java 11 ou 17.

## Personalizações do Amazon Q Developer

Com sua capacidade de personalização, o Amazon Q Developer pode fornecer sugestões em linha com base na base de código da própria empresa. A empresa fornece seu repositório de código para o Amazon Simple Storage Service (Amazon S3) ou por meio do AWS CodeConnections,

anteriormente conhecido como Connections. AWS CodeStar Em seguida, o Amazon Q usa o repositório de código personalizado com segurança habilitada para recomendar padrões de codificação relevantes para os desenvolvedores dessa organização.

Ao usar as personalizações do Amazon Q Developer, esteja ciente do seguinte:

- Em junho de 2024, o recurso Amazon Q Developer Customizations está em modo de pré-visualização. Como resultado, o recurso pode ter disponibilidade e suporte limitados.
- As sugestões personalizadas de código em linha só serão precisas devido à qualidade dos repositórios de código fornecidos. Recomendamos que você revise uma pontuação de avaliação para cada personalização criada.
- Para otimizar o desempenho, recomendamos que você inclua pelo menos 20 arquivos de dados contendo o idioma especificado, em que todos os arquivos de origem tenham mais de 10 MB. Certifique-se de que seu repositório consiste em código-fonte referenciável e não em arquivos de metadados (por exemplo, arquivos de configuração, arquivos de propriedades e arquivos readme).

Ao usar as personalizações do Amazon Q Developer, você pode economizar tempo das seguintes formas:

- Use recomendações baseadas no código proprietário da sua própria empresa.
- Aumente a reutilização das bases de código existentes.
- Crie padrões repetíveis que sejam generalizados em toda a sua empresa.

# Melhores práticas de codificação com o Amazon Q Developer

Esta seção discute as melhores práticas para codificação com o Amazon Q Developer. As melhores práticas incluem as seguintes categorias:

- Integração - Métodos e considerações ao fazer a integração
- Geração de código - Orientação para usar a geração de código com sucesso
- Recomendações de código - Técnicas para melhorar o código

## Melhores práticas para integrar o Amazon Q Developer

O Amazon Q Developer é um poderoso assistente generativo de codificação de IA que está disponível por meio de programas populares IDEs como o Visual Studio Code e. JetBrains Esta seção se concentra nas melhores práticas para acessar e integrar o Amazon Q Developer ao seu ambiente de desenvolvimento de codificação.

### Pré-requisitos para o Amazon Q Developer

O Amazon Q Developer está disponível como parte do AWS Toolkit for Visual Studio Code e do AWS Toolkit for JetBrains (por exemplo, PyCharm IntelliJ e). Para Visual Studio Code e JetBrains IDEs, o Amazon Q Developer oferece suporte a Python, Java,, JavaScript TypeScript, C#, Go, Rust, PHP, Ruby, Kotlin, C, C++, scripts Shell, SQL e Scala.

Para obter instruções detalhadas sobre como instalar o AWS Toolkit Visual Studio Code e um JetBrains IDE, consulte Instalando a extensão ou o plug-in do Amazon Q Developer em seu IDE no Guia do usuário do desenvolvedor do Amazon Q.

### Melhores práticas ao usar o Amazon Q Developer

As melhores práticas gerais ao usar o Amazon Q Developer incluem o seguinte:

- Forneça contexto relevante para obter respostas mais precisas, como linguagens de programação, estruturas e ferramentas que estão em uso. Divida problemas complexos em componentes menores.

- Experimente e repita suas solicitações e perguntas. A programação geralmente envolve tentar abordagens diferentes.
- Sempre analise as sugestões de código antes de aceitá-las e edite-as conforme necessário para garantir que elas façam exatamente o que você pretendia.
- Use o recurso de personalização para informar o Amazon Q Developer sobre suas bibliotecas internas APIs, melhores práticas e padrões arquitetônicos para obter recomendações mais relevantes.

## Privacidade de dados e uso de conteúdo no Amazon Q Developer

Ao decidir usar o Amazon Q Developer, você deve entender como seus dados e conteúdo são usados. A seguir estão os pontos-chave:

- Para usuários do Amazon Q Developer Pro, seu conteúdo de código não é usado para aprimoramento de serviços ou treinamento de modelos.
- Para usuários do Amazon Q Developer Free Tier, você pode optar por não ter seu conteúdo usado para melhorar o serviço por meio de configurações ou AWS Organizations políticas do IDE.
- O conteúdo transmitido é criptografado e qualquer conteúdo armazenado é protegido com criptografia em repouso e controles de acesso. Para obter mais informações, consulte Criptografia de dados no Amazon Q Developer no Amazon Q Developer User Guide.

# Melhores práticas para geração de código com o Amazon Q Developer

O Amazon Q Developer fornece geração automática de código, preenchimento automático e sugestões de código em linguagem natural. A seguir estão as melhores práticas para usar a assistência de codificação em linha do Amazon Q Developer:

- Forneça contexto para ajudar a melhorar a precisão das respostas

  Comece com o código existente, importe bibliotecas, crie classes e funções ou estabeleça esqueletos de código. Esse contexto ajudará a melhorar significativamente a qualidade da geração de código.
- Codifique naturalmente

Use a geração de código do Amazon Q Developer como um mecanismo robusto de preenchimento automático. Codifique como você faz normalmente e deixe que a Amazon Q forneça sugestões enquanto você digita ou pausa. Se a geração de código não estiver disponível ou você estiver com problemas de código, inicie o Amazon Q digitando Alt+C em um PC ou Option +C no macOS. Para obter mais informações sobre ações comuns que você pode realizar ao usar sugestões em linha, consulte Usando teclas de atalho no Amazon Q Developer User Guide.

- Inclua bibliotecas de importação que sejam relevantes para os objetivos do seu script

  Inclua bibliotecas de importação relevantes para ajudar o Amazon Q a entender o contexto e gerar código de acordo. Você também pode pedir à Amazon Q que sugira declarações de importação relevantes.

- Mantenha um contexto claro e focado

  Mantenha seu script focado em objetivos específicos e modularize funcionalidades distintas em scripts separados com contexto relevante. Evite contextos ruidosos ou confusos.

- Experimente com instruções

  Explore diferentes instruções para estimular o Amazon Q a produzir resultados úteis na geração de código. Por exemplo, experimente as seguintes abordagens:

  - Use blocos de comentários padrão para solicitações em linguagem natural.
  - Crie esqueletos com comentários para preencher classes e funções.
  - Seja específico em suas solicitações, fornecendo detalhes em vez de generalizar.

- Converse com o Amazon Q Developer e peça ajuda

  Se o Amazon Q Developer não estiver fornecendo sugestões precisas, converse com o Amazon Q Developer em seu IDE. Ele pode fornecer trechos de código ou classes e funções completas para iniciar seu contexto. Para obter mais informações, consulte Conversando com o Amazon Q Developer sobre código no Amazon Q Developer User Guide.

# Melhores práticas para recomendações de código com o Amazon Q Developer

O Amazon Q Developer pode tirar dúvidas do desenvolvedor e avaliar o código para oferecer recomendações que vão desde a geração de código e correções de bugs até orientações usando linguagem natural. A seguir estão as melhores práticas para usar o chat no Amazon Q:

- Gere código do zero

  Para novos projetos ou quando você precisar de uma função geral (por exemplo, copiar arquivos do Amazon S3), peça ao Amazon Q Developer que gere exemplos de código usando prompts em linguagem natural. O Amazon Q pode fornecer links relacionados a recursos públicos para posterior validação e investigação.
- Busque conhecimento de codificação e explicações de erros

  Ao se deparar com problemas de codificação ou mensagens de erro, forneça o bloco de código (com mensagem de erro, se aplicável) e sua pergunta como um aviso para o Amazon Q Developer. Esse contexto ajudará a Amazon Q a fornecer respostas precisas e relevantes.
- Melhore o código existente

  Para corrigir erros conhecidos ou otimizar o código (por exemplo, para reduzir a complexidade), selecione o bloco de código relevante e envie-o para o Amazon Q Developer com sua solicitação. Seja específico com suas instruções para obter melhores resultados.
- Explique a funcionalidade do código

  Ao explorar novos repositórios de código, selecione um bloco de código ou um script inteiro e envie-o para o Amazon Q Developer para obter explicações. Reduza o tamanho da seleção para obter explicações mais específicas.
- Gere testes unitários

  Depois de enviar um bloco de código como solicitação, peça ao Amazon Q Developer que gere testes unitários. Essa abordagem pode economizar tempo e custos de desenvolvimento relacionados à cobertura de código DevOps e.
- Encontre AWS respostas

  O Amazon Q Developer é um recurso valioso para desenvolvedores que trabalham com ele Serviços da AWS porque contém uma grande quantidade de conhecimento relacionado AWS

a. Se você está enfrentando desafios com um problema específico AWS service (Serviço da AWS), encontrando mensagens de erro específicas ou tentando aprender algo novo AWS service (Serviço da AWS), o Amazon Q geralmente fornece informações relevantes e úteis. AWS

Sempre revise as recomendações que o Amazon Q Developer fornece a você. Em seguida, faça as edições necessárias e execute os testes para garantir que o código atenda à funcionalidade pretendida.

# Exemplos de código com o Amazon Q Developer

Esta seção fornece exemplos realistas de como melhorar sua experiência e a geração de código com foco no Python and Java idiomas. Além de exemplos em linha, alguns cenários mostram como você pode usar o recurso de bate-papo do Amazon Q Developer para solicitar recomendações de código durante a programação.

Para obter mais informações, consulte os tópicos a seguir.

- Python exemplos
- Java exemplos
- Exemplos de bate-papo

## Python exemplos

Os seguintes exemplos de Python exemplos demonstram como você pode começar a gerar sugestões de código em linha com o Amazon Q Developer. Esses exemplos ilustrativos mostram como acionar o assistente de codificação para seus próprios projetos. A maioria dos exemplos a seguir se aplica a outros idiomas:

- Gere classes e funções
- Código do documento
- Gere algoritmos
- Gere testes unitários

### Gere classes e funções

Para gerar a implementação total ou parcial da classe, use comentários de código. O comentário descreve a intenção da turma. Comece de forma simples e adicione mais detalhes, se necessário. Depois de gerar uma classe, você pode continuar gerando funções dentro de seu esqueleto.

Ao escrever comentários, é preferível usar a estrutura de comentários do idioma. Em Python, para comentários de linha única, use # .Para comentários de várias linhas, use `'''` `'''` ou`"""` `"""`.

Entrada:

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

Saída:

```
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
```

Agora que há uma classe inicial, uma função adicional pode ser gerada com comentários adicionais.

Entrada:

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

Saída:

```
def read_document(self, path):
        """Reads document from path and prints first line"""
        import os
        if os.path.isfile(path):
            with open(path, 'r') as f:
                print(f.readline())
        else:
            print(f"File not found at {path}")
```

# Código do documento

Reduza o tempo na documentação do código confiando na documentação em linha. Adicionar Python pausas para comentários`''' '''`.

Entrada:

```
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}")
```

Saída:

```
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}")
```

## Gere algoritmos

O Amazon Q Developer pode gerar algoritmos populares, especialmente relacionados AWS SDKs a. Em seu script, adicione um prompt que especifique a função que você deseja.

Este exemplo mostra como gerar um dicionário de chaves de identificação de usuário para nomear valores e, em seguida, gerar uma função de classificação usando o algoritmo de classificação por bolhas.

Entrada:

```
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
    '''
```

Saída:

```
    @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)
```

Além disso, você pode criar solicitações para procedimentos padrão. Neste exemplo, o usuário cria um prompt para armazenar o dicionário em uma tabela do Amazon DynamoDB. No início do script, o usuário adiciona uma solicitação para adicionar bibliotecas do DynamoDB para ajudar a fornecer contexto adicional ao Amazon Q Developer.

Entrada:

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

Saída:

```
'''
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')
```

Agora, o usuário pode facilmente pedir ajuda ao Amazon Q para armazenar o dicionário no DynamoDB.

Entrada:

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

Saída:

```
@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'])
```

## Gere testes unitários

Com comentários em linha, o Amazon Q Developer pode gerar testes unitários para realizar afirmações e outros testes. Neste exemplo, o usuário solicita à Amazon Q que gere bibliotecas

aplicáveis para testes unitários para definir o contexto do script de teste. Em seguida, o usuário solicita à Amazon Q que escreva uma função de teste para a `sort_user_id_name_map` função.

Entrada:

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

Saída:

```
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()
```

Além disso, em um bate-papo entre o usuário e o Amazon Q Developer, ele sugere e gera testes unitários com base nos prompts de entrada de código do usuário. Para obter mais informações, consulte Exemplos de bate-papo.

## Java exemplos

Os seguintes exemplos de Java exemplos demonstram como os usuários podem começar a gerar sugestões de código em linha. Esses exemplos ilustrativos mostram como acionar o assistente de codificação para seus próprios projetos. A maioria dos exemplos a seguir se aplica a outros idiomas:

- Gere classes e funções
- Código do documento
- Gere algoritmos

- Gere testes unitários

## Gere classes e funções

Para gerar a implementação total ou parcial da classe, use comentários de código. O comentário descreve a intenção da turma. Comece de forma simples e adicione mais detalhes, se necessário. Depois de gerar uma classe, você pode continuar gerando funções dentro de seu esqueleto.

Ao escrever comentários, é preferível usar a estrutura de comentários do idioma. Em Java, isso é `/ ** */`.

Entrada:

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

Saída:

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

Agora que há uma classe inicial, uma função adicional pode ser gerada com comentários adicionais.

Entrada:

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

Saída:

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

## Código do documento

Reduza o tempo na documentação do código confiando na documentação em linha. Adicione um Java pausa para comentários`/* */`.

Entrada:

```
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 a geração em linha não ocorrer, execute uma sugestão em linha do Amazon Q Developer depois de colocar o cursor onde você deseja a documentação.

Saída:

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

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

Para explicações mais simples do código (classes, funções ou elementos de script), use caracteres de encaminhamento (`//`) que estão estrategicamente colocados antes das seções de código nas quais você deseja adicionar documentação.

Entrada:

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

Saída:

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

## Gere algoritmos

O Amazon Q Developer pode gerar algoritmos populares, especialmente relacionados AWS SDKs a. Em seu script, adicione um prompt que especifique a função que você deseja.

Este exemplo gera um mapa de hash das chaves de identificação do usuário para nomear valores e, em seguida, gera uma função de classificação usando o algoritmo de classificação por bolhas.

Entrada:

```
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
    */


}
```

Saída:

```
/*
* 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;
}
```

Além disso, os usuários podem criar solicitações para procedimentos padrão. Neste exemplo, o usuário cria um prompt para armazenar o mapa de hash em uma tabela do Amazon DynamoDB. No início do script, o usuário adiciona um prompt para adicionar o DynamoDB Java bibliotecas para ajudar a fornecer contexto adicional ao Amazon Q Developer.

Entrada:

```
// 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
     */
```

Saída:

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

```
    }
```

## Gere testes unitários

Com comentários em linha, o Amazon Q Developer pode gerar testes unitários para realizar afirmações e outros JUnit testes. Neste exemplo, o usuário primeiro solicita à Amazon Q que gere bibliotecas aplicáveis JUnit para definir o contexto do script de teste. Em seguida, o usuário cria uma classe pública seguida por um declarador e sequências de comentários.

Entrada:

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

public class UnitTests {

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

Saída:

```
@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);
}
```

Além disso, em um bate-papo entre o usuário e o Amazon Q Developer, ele sugere e gera testes unitários com base nos prompts de entrada de código do usuário. Para obter mais informações, consulte Exemplos de bate-papo.

# Exemplos de bate-papo

O Amazon Q Developer Chat auxilia os desenvolvedores em uma variedade de tarefas, desde codificação, teste e atualização de aplicativos até diagnóstico de erros, execução de verificações e correções de segurança e otimização de recursos. AWS Esta seção aborda como o bate-papo com o Amazon Q pode ajudar um desenvolvedor em tarefas típicas de programação, como as seguintes:

- Pergunte sobre Serviços da AWS
- Gerar código
- Gere testes unitários
- Explique o código

## Pergunte sobre Serviços da AWS

Os usuários podem fazer perguntas ao Amazon Q Developer sobre Serviços da AWS, como encontrar o serviço certo ou entender as melhores práticas.

Converse sobre Serviços da AWS

Usuário:

Qual é o melhor serviço de contêiner para executar minha carga de trabalho se eu precisar manter meus custos baixos?

Amazon Q:

Com base nas informações fornecidas, parece que você está procurando um serviço de contêiner econômico para executar suas cargas de trabalho. Nesse caso, eu recomendaria considerar o Amazon Elastic Container Service (Amazon ECS) com o Fargate. [1]

O Amazon ECS com Fargate é um mecanismo de computação sem servidor para contêineres que permite que você execute seus contêineres sem precisar gerenciar nenhuma infraestrutura subjacente. Com o Fargate, você não precisa provisionar ou gerenciar servidores — o Fargate cuida disso para você. Isso pode ajudar a manter seus custos baixos, pois você paga apenas pelos recursos computacionais que seus contêineres realmente usam, sem precisar pagar pela capacidade ociosa do servidor. [2]

Alguns dos principais benefícios de usar o Amazon ECS com o Fargate:

• Não há necessidade de gerenciar servidores ou clusters — o Fargate cuida do provisionamento e gerenciamento da infraestrutura para você.

• Pay-as-you-go preços — você paga apenas pelos recursos computacionais que seus contêineres usam, sem precisar pagar pela capacidade ociosa.

• Escalabilidade automática — O Fargate pode escalar automaticamente seus contêineres para cima e para baixo com base na demanda.

• Forte integração com outros, Serviços da AWS como balanceamento de carga, registro e monitoramento.

Os recursos fornecidos também mencionam que o serviço ECS oferece flexibilidade em termos de agendamento e posicionamento de contêineres, o que pode ajudar a otimizar ainda mais seus custos.

Eu recomendaria revisar os detalhes de preços do Amazon ECS com Fargate no AWS site para ter uma ideia melhor dos custos com base em seus requisitos específicos de carga de trabalho. E, claro, sinta-se à vontade para me avisar se tiver outras dúvidas!

Origens

[1] Qual serviço de AWS contêiner é melhor para você?

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

[2] Java opções de contêiner para cargas de trabalho não HTTP

stackoverflow.com > perguntas > 43417016

Encerre o bate-papo do Amazon Q sobre Serviços da AWS

# Gerar código

Os usuários podem pedir ajuda ao Amazon Q Developer para escrever trechos de código.

Converse sobre como escrever código

Usuário:

Você pode me dar um exemplo de como listar todos os buckets do S3 usando o? AWS SDK for Python (Boto)

Amazon Q:

Certamente, aqui está um exemplo de como listar todos os buckets do S3 usando o AWS SDK for Python (Boto)[1]:

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

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

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

Veja como o código funciona:

1. Primeiro importamos a biblioteca boto3, que é o SDK para Python.
2. Criamos um cliente S3 usando`boto3.client('s3')`.
3. Chamamos o `list_buckets()` método no cliente S3, que retorna uma resposta contendo uma lista de todos os buckets na conta.
4. Em seguida, repassamos a `Buckets` lista na resposta e imprimimos o nome de cada bucket.

Esse é um exemplo básico, mas AWS SDK for Python (Boto) fornece muitos recursos mais avançados para trabalhar com o Amazon S3, como criar, excluir e gerenciar objetos em buckets, definir políticas de bucket e muito mais. Você pode encontrar mais exemplos e documentação na documentação AWS SDK for Python (Boto)). [2]

Lembre-se de que, ao trabalhar com Serviços da AWS, é importante seguir as melhores práticas para lidar com credenciais e informações confidenciais, como armazenar chaves de AWS acesso em variáveis de ambiente em vez de codificá-las no código.

Fontes: [1] Exemplos do Amazon S3 usando SDK para Python ( AWS Boto3) - Exemplos de códigos de SDK docs.aws.amazon.com > code-library > latest-1x >... > python-s3.html

[2] Comece a usar buckets e objetos do Amazon S3 usando um AWS SDK - AWS SDK Code Examples docs.aws.amazon.com > code-library > latest-1x >... > s3_example_s3_buckets_ _sect... GettingStarted

Encerre o bate-papo do Amazon Q sobre como escrever código

## Gere testes unitários

Os usuários podem enviar o código selecionado como um aviso para o Amazon Q Developer, onde instruções adicionais são aplicadas, como a geração de um teste unitário. Para step-by-step obter instruções, consulte Enviar código para o Amazon Q no Guia do usuário do desenvolvedor do Amazon Q.

Converse sobre testes unitários

Usuário:

Escreva um teste unitário com dados simulados para a seguinte função:

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

Amazon Q:

Para escrever um teste de unidade com dados simulados para a função pollMessages, podemos usar uma estrutura de simulação como o Mockito. Veja um exemplo abaixo:

```
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);
    }
```

```
}
```

Neste exemplo, estamos usando o Mockito para simular o cliente AmazonSQS e configurar o comportamento esperado para o método receiveMessage para que ele retorne duas mensagens simuladas com os corpos “message1" e “message2". Em seguida, chamamos o método pollMessages e afirmamos que a lista de mensagens retornada corresponde aos valores esperados.

Os pontos principais são:

1. Usamos a anotação @ ExtendWith (MockitoExtension.class) para habilitar a integração do Mockito com o. JUnit
2. Usamos a anotação @Mock para criar uma instância simulada do cliente AmazonSQS.
3. Usamos a InjectMocks anotação @ para injetar o cliente simulado do AmazonSQS na classe. SQSUtils
4. No método de teste, configuramos o comportamento esperado do método receiveMessage usando mockito.when () .thenReturn ().
5. Chamamos o método pollMessages e afirmamos que a lista de mensagens retornada corresponde aos valores esperados.

Essa abordagem nos permite testar o método pollMessages sem realmente interagir com o serviço Amazon SQS, o que pode tornar o teste mais rápido e confiável.

Encerre o bate-papo do Amazon Q sobre testes unitários

## Explique o código

Para processos de revisão de código, os desenvolvedores podem enviar uma seleção de código para o Amazon Q Developer para entender melhor o código. Além disso, a explicação fornecida pelo Amazon Q também pode servir para fins de documentação. No exemplo a seguir, o Amazon Q explica a função do manipulador com uma AWS Lambda função e, em seguida, propõe uma docstring.

Converse sobre como explicar o código

Usuário:

Explique a seguinte parte do meu código:

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

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

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

Amazon Q:

O bloco de código fornecido parece ser uma AWS Lambda função escrita em Python. Deixe-me explicar os principais aspectos desse código:

1. A função é chamada lambda_handler e é o ponto de entrada para a função Lambda.
2. Ele usa a biblioteca boto3 para interagir com os serviços AWS DynamoDB e S3.
3. A função recupera uma chave de objeto do S3 e um nome de bucket dos dados de eventos recebidos.

4. Em seguida, ele chama uma função detect_text (que não é mostrada no código fornecido) para extrair o conteúdo de texto completo do objeto S3.
5. A função também recupera metadados sobre o objeto S3 usando o método head_object.
6. Em seguida, ele extrai a hora atual e chama uma função detect_restaurant_closure (também não mostrada) para determinar o motivo e o resultado do fechamento de um restaurante com base no texto extraído.
7. Por fim, a função armazena as seguintes informações em uma tabela do DynamoDB:
   - A chave do objeto S3 como id
   - A hora da criação
   - O motivo e o resultado do fechamento do restaurante
   - O texto do objeto S3 extraído

No geral, essa função do Lambda parece fazer parte de um sistema maior que processa objetos do S3, extrai conteúdo de texto, analisa informações sobre o fechamento de restaurantes e armazena os resultados em uma tabela do DynamoDB. Os detalhes específicos das funções detect_text e detect_restaurant_closure seriam necessários para entender completamente a funcionalidade desse código.

Fim da resposta do Amazon Q

Usuário: resuma esta explicação em uma docstring para esta função lambda_handler

Amazon Q: Aqui está uma docstring sugerida para a função 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...
```

Essa docstring fornece um resumo de alto nível da finalidade da função, dos parâmetros de entrada esperados e do valor de retorno esperado (ou a falta dele, neste caso). Ele fornece ao usuário uma compreensão clara do que a função lambda_handler faz sem se aprofundar nos detalhes da implementação.

Fontes: [1] Defina o manipulador da função Lambda em C# - AWS Lambda https://docs.aws.amazon.com/lambda/latest/dg/csharp-handler.html -: ~:text=O manipulador da função Lambda é,, sai ou expira.

Fim do bate-papo do Amazon Q sobre como explicar o código

# Solução de problemas de cenários de geração de código no Amazon Q Developer

Você pode encontrar os seguintes cenários comuns com geração de código e resoluções imprecisas ao usar o Amazon Q Developer:

- Geração de código vazio
- Comentários contínuos
- Geração incorreta de código em linha
- Resultados inadequados de bate-papos

## Geração de código vazio

Ao desenvolver o código, você pode observar os seguintes problemas:

- O Amazon Q não fornece uma sugestão.
- A mensagem “Nenhuma sugestão do Amazon Q” aparece em seu IDE.

Seu primeiro pensamento pode ser que o Amazon Q não está funcionando corretamente. No entanto, a causa raiz desses problemas geralmente está associada ao contexto no script ou ao projeto aberto no IDE.

Se o Amazon Q Developer não fornecer uma sugestão automaticamente, você poderá usar os seguintes atalhos para executar manualmente as sugestões em linha do Amazon Q:

- PC - Alt+C
- macOS - Opção+C

Para obter mais informações, consulte Usando teclas de atalho no Amazon Q Developer User Guide.

Na maioria dos cenários, o Amazon Q gera uma sugestão. Quando o Amazon Q retornar uma mensagem “Nenhuma sugestão da Amazon Q”, revise as seguintes resoluções comuns para esse problema:

- Contexto inadequado — Certifique-se de que as bibliotecas aplicáveis estejam presentes. Certifique-se de que as classes e funções definidas no script estejam relacionadas ao novo código.
- Solicitação ambígua — Se a solicitação for confusa ou obscura, o Amazon Q terá um desempenho inferior ao fazer sugestões de código em linha. Nesses cenários, seja mais específico em seu código e comentários para obter respostas adequadas.

Se você estiver preso, continue programando da melhor maneira possível. Normalmente, o Amazon Q reconhece sua intenção, o que faz com que ele ofereça sugestões. Outra opção é enviar um trecho de código com instruções imediatas em um chat do Amazon Q. Para obter mais informações, consulte Enviar código para o Amazon Q no Guia do usuário do desenvolvedor do Amazon Q.

## Comentários contínuos

Você pode escrever uma solicitação usando comentários e receber apenas linhas de comentários genéricas ou repetidas da Amazon Q.

Entrada:

```
// Kafka test
```

Resultado:

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

Normalmente, esse problema ocorre devido à falta de contexto no script ou às instruções de aviso inadequadas. Recomendamos que você use o formato padrão para comentários de acordo com o idioma. Neste exemplo, o usuário alterou o prompt `/* */` para usar em um bloco de comentários, e essa alteração resultou em um melhor desempenho do Amazon Q.

Entrada:

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

Resultado:

Nesse resultado, o usuário aceitou várias sugestões sequencialmente.

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

# Geração incorreta de código em linha

Ocasionalmente, as sugestões em linha do Amazon Q Developer não são úteis ou não atendem ao objetivo do seu prompt. Às vezes, sugestões de código em linha podem fornecer uma função genérica semelhante a um método simples do Hello World.

Quando as sugestões de código em linha não forem úteis, experimente as seguintes práticas recomendadas:

- Analise o contexto do script. Normalmente, quanto mais contexto você fornece, melhores resultados você obtém do Amazon Q.
- Se o contexto variar e/ou for ruidoso, modularize seu código. Agrupe funções semelhantes no mesmo script. Separe diferentes classes e funções de acordo com seus objetivos.
- Escreva seus nomes de classes e funções de forma tática para ajudar a Amazon Q a fornecer recomendações melhores.

Neste exemplo, o desenvolvedor está escrevendo um script do zero que carrega arquivos para o Hadoop.

Entrada:

```
class  MyFunction{

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

Resultado:

```
class  MyFunction{

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

}
```

Embora o Amazon Q tenha fornecido uma recomendação, ela não atendeu à expectativa do desenvolvedor. Para melhorar o desempenho, o desenvolvedor importa bibliotecas e altera o nome da classe para representar os utilitários do Hadoop.

Correção:

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

```
import java.io.FileReader;

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

class  HadoopUtilities{

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

Resultado:

Nesse resultado, o desenvolvedor aceitou várias sugestões sequencialmente.

```
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!");
}
```

No mesmo HadoopUtilities script, o desenvolvedor adiciona um prompt de comentário para ler um tópico do Amazon Simple Queue Service (Amazon SQS).

Entrada:

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

Resultado:

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

Em vez de usar métodos Java para ler uma mensagem SQS do Amazon SQS, o Amazon Q sugere imprimir um método de string. No entanto, essa sugestão não satisfaz o objetivo original do desenvolvedor.

Nesse caso, é preferível remodularizar o código. O foco do script está nos métodos do Hadoop. Portanto, pode ser uma abordagem melhor criar um script utilitário específico para operações do SQS.

A criação de uma classe separada para utilitários SQS melhora a sugestão em linha, que pode ser usada no HadoopUtilities roteiro.

Entrada:

```
// 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
}
```

Resultado:

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

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

Se a sugestão em linha ainda não estiver fornecendo os resultados esperados, peça ao Amazon Q Developer em uma sessão de bate-papo para gerar o código.

## Converse sobre a geração de código

Usuário:

Gere uma função java para ler uma mensagem de uma fila do Amazon SQS

Amazon Q:

Aqui está um exemplo de função Java para ler uma mensagem de uma fila do 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;
        }
    }
```

Esse código cria uma classe SQSMessage Reader que usa um cliente AmazonSQS e uma URL de fila como parâmetros do construtor. O método readMessage () usa o ReceiveMessageRequest para buscar até 1 mensagem da fila, com um tempo máximo de espera de 10 segundos.

Se uma mensagem for recebida, o corpo da mensagem será impresso e a mensagem será excluída da fila usando o. DeleteMessageRequest O corpo da mensagem é então retornado. Se nenhuma mensagem for recebida, null será retornado.

Encerre o bate-papo do Amazon Q sobre a geração de código

# Resultados inadequados de bate-papos

Ao desenvolver o código, os desenvolvedores podem pedir à Amazon Q que gere funções comuns, forneça recomendações ou explique o código. Às vezes, após receber uma solicitação (como uma pergunta ou um trecho de código), o Amazon Q pode fornecer feedback geral ou resultados que não atendem às suas expectativas. Nesses cenários, tente o seguinte:

- Experimente diferentes instruções para melhorar os resultados de geração de código que você obtém do Amazon Q.
- Siga o URLs que o Amazon Q frequentemente fornece em suas respostas às suas solicitações ou perguntas.
- Forneça feedback para ajudar a melhorar o desempenho futuro do Amazon Q.

Por exemplo, um desenvolvedor está criando uma solução de Retrieval Augmented Generation (RAG) e precisa fragmentar seu documento de acordo com as relações semânticas entre as frases. O desenvolvedor conversa com o Amazon Q da seguinte forma.

### Converse sobre uma solução RAG

Usuário:

Para uma solução RAG, gere código para fragmentação semântica

Amazon Q:

Certamente, aqui está um exemplo de como você pode implementar a fragmentação semântica para uma solução de Geração Aumentada de Recuperação (RAG) em Python usando a LangChain biblioteca:

```
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
```

Neste exemplo, estamos usando o TokenTextSplitter de LangChain para dividir o texto de entrada em partes de 500 caracteres com uma sobreposição de 50 caracteres. A ferramenta HuggingFaceEmbeddings o modelo é usado para gerar incorporações para cada pedaço, e o Chroma O armazenamento vetorial é usado para armazenar os pedaços e suas incorporações.

A função semantic_chunk pega o texto de entrada, o divide em partes usando o text_splitter, gera incorporações para cada bloco usando o modelo de incorporações e, em seguida, adiciona as partes e suas incorporações ao vectorstore.

Essa abordagem permite dividir o texto de entrada em partes semanticamente significativas, que podem ser úteis para uma variedade de tarefas, como responder a perguntas ou resumir, em uma solução RAG.

Observe que este é um exemplo básico e talvez seja necessário ajustar os parâmetros (por exemplo, chunk_size, chunk_overlap) ou a escolha do divisor de texto e do modelo de incorporação com base em seu caso de uso e dados específicos.

Fim do bate-papo do Amazon Q sobre uma solução RAG

No entanto, essa abordagem fornece apenas uma maneira básica de fragmentar documentos usando a técnica de janela deslizante. O desenvolvedor quer uma forma avançada que dependa da medição de incorporações. Nesses cenários, o desenvolvedor deve adicionar mais detalhes ao prompt de acordo com seu objetivo. Quando o desenvolvedor se refere à conhecida função de divisor de AI21 texto com LangChain, ele recebe uma resposta aprimorada, conforme mostrado no bate-papo a seguir.

### Converse sobre uma solução RAG (com aviso aprimorado)

Usuário:

A solicitação acima se refere apenas à divisão em relação ao tamanho do token por bloco. Eu quero usar o divisor de texto AI21 semântico. Gere uma função usando a divisão semântica de texto A121 com langchain.

Amazon Q:

Ok, vamos usar o Divisor de Texto AI21 Semântico com LangChain para implementar a fragmentação semântica. Aqui está um exemplo: [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
```

Neste exemplo, estamos usando o AI21 SemanticTextSplitter from LangChain para dividir o texto de entrada em partes semanticamente significativas. A ferramenta HuggingFaceEmbeddings o modelo é usado para criar incorporações para cada pedaço, e o Chroma O armazenamento vetorial é usado para armazenar os pedaços e suas incorporações.

A função semantic_chunk pega o texto de entrada, o divide em partes usando o text_splitter, cria incorporações para cada bloco usando o modelo de incorporações e, em seguida, adiciona as partes e suas incorporações ao vectorstore.

Essa abordagem permite dividir o texto de entrada em partes semanticamente significativas, que podem ser úteis para uma variedade de tarefas, como responder a perguntas ou resumir, em uma solução RAG.

Observe que talvez seja necessário ajustar os parâmetros ou a escolha do modelo de incorporação com base em seu caso de uso e dados específicos.

Fontes [1] Aprenda a usar divisores de texto em LangChain

soosweb3.hashnode.dev > -divisores de texto the-ultimate-langchain-series

Fim do bate-papo do Amazon Q sobre uma solução RAG (com aviso aprimorado)

Com base nos resultados do bate-papo, o desenvolvedor pode usar a função e navegar até a fonte das informações.

Em geral, tópicos e perguntas avançadas exigem um contexto mais específico no prompt quando você conversa com o Amazon Q Developer. Se você acredita que os resultados do seu bate-papo não são precisos, use o ícone de polegar para baixo para fornecer feedback sobre a resposta do Amazon Q. O Amazon Q Developer usa continuamente o feedback para melhorar futuros lançamentos. Para interações que produziram resultados positivos, é útil fornecer seu feedback usando o ícone de polegar para cima.

# FAQssobre o Amazon Q Developer

Esta seção fornece respostas para perguntas frequentes sobre o uso do Amazon Q Developer para desenvolvimento de código.

## O que é o Amazon Q Developer?

O Amazon Q Developer é um poderoso serviço generativo baseado em IA projetado para acelerar as tarefas de desenvolvimento de código, fornecendo geração e recomendações inteligentes de código. Em 30 de abril de 2024, a Amazon CodeWhisperer tornou-se parte do Amazon Q Developer.

## Como faço para acessar o Amazon Q Developer?

O Amazon Q Developer está disponível como parte dos AWS kits de ferramentas para Visual Studio Code e JetBrains IDEs, como IntelliJ e. PyCharm Para começar, instale a AWS Toolkit versão mais recente.

## Quais linguagens de programação são compatíveis com o Amazon Q Developer?

Para Visual Studio Code e JetBrains IDEs, o Amazon Q Developer oferece suporte Python, Java, JavaScript, TypeScript, C#, Go, Rust, PHP, Ruby, Kotlin, C, C++, scripts de Shell e Scala. SQL Embora este guia se concentre em Python e Java por exemplo, os conceitos são aplicáveis a qualquer linguagem de programação suportada.

## Como posso fornecer contexto ao Amazon Q Developer para uma melhor geração de código?

Comece com o código existente, importe bibliotecas relevantes, crie classes e funções ou estabeleça esqueletos de código. Use blocos de comentários padrão para solicitações em linguagem natural. Mantenha seu script focado em objetivos específicos e modularize funcionalidades distintas em scripts separados com contexto relevante. Para obter mais informações, consulte Melhores práticas de codificação com o Amazon Q Developer.

## O que devo fazer se a geração de código em linha com o Amazon Q Developer não for precisa?

Analise o contexto do script, verifique se as bibliotecas estão presentes e certifique-se de que as classes e funções estejam relacionadas ao novo código. Modularize seu código e separe diferentes classes e funções de acordo com seus objetivos. Escreva instruções ou comentários claros e específicos. Se você ainda não tiver certeza sobre a precisão do código e não conseguir continuar com ele, inicie um bate-papo com o Amazon Q e envie o trecho do código com as instruções. Para obter mais informações, consulte Solução de problemas de cenários de geração de código no Amazon Q Developer.

## Como posso usar o recurso de bate-papo do Amazon Q Developer para geração de código e solução de problemas?

Converse com o Amazon Q para gerar funções comuns, pedir recomendações ou explicar o código. Se a resposta inicial não for satisfatória, experimente com instruções diferentes e siga as instruções fornecidas. URLs Além disso, forneça feedback à Amazon Q para ajudar a melhorar o desempenho do chat no futuro. Use os ícones de polegar para cima e polegar para baixo para fornecer seu feedback. Para obter mais informações, consulte Exemplos de bate-papo.

## Quais são algumas das práticas recomendadas para usar o Amazon Q Developer?

Forneça contexto relevante, experimente e repita as solicitações, analise as sugestões de código antes de aceitá-las, use recursos de personalização e entenda as políticas de privacidade de dados e uso de conteúdo. Para obter mais informações, consulte Melhores práticas para geração de código com o Amazon Q Developer e Melhores práticas para recomendações de código com o Amazon Q Developer.

## Posso personalizar o Amazon Q Developer para gerar recomendações com base no meu próprio código?

Sim, use personalizações, que são um recurso avançado do Amazon Q Developer. Com as personalizações, as empresas podem fornecer seus próprios repositórios de código para permitir

que o Amazon Q Developer recomende sugestões de código em linha. Para obter mais informações, consulte Recursos avançados do Amazon Q Developer and Resources.

# Próximas etapas no uso do Amazon Q Developer

Com o conhecimento obtido neste guia abrangente, você pode usar o Amazon Q Developer em seu fluxo de trabalho de codificação de forma eficaz. Instale o AWS Toolkit em seu IDE preferido (Visual Studio Code ou JetBrains) e comece a explorar a geração generativa de código com inteligência artificial e as recomendações do Amazon Q Developer.

A maneira mais eficaz de liberar todo o potencial do Amazon Q Developer é por meio da experiência prática com seu próprio código. Ao integrar o Amazon Q ao seu ciclo de vida de desenvolvimento, consulte este guia para obter melhores práticas, solução de problemas e exemplos reais.

Além disso, mantenha-se informado revisando os AWS blogs e os guias do desenvolvedor que são referenciados em Recursos. Esses recursos fornecem as últimas atualizações, melhores práticas e insights para ajudar você a otimizar seu uso do Amazon Q Developer.

Seu feedback é inestimável para melhorar este guia e ajudá-lo a continuar sendo um recurso valioso para desenvolvedores. Compartilhe suas experiências, desafios e sugestões para futuras versões. Sua opinião ajudará a aprimorar o guia com exemplos adicionais, cenários de solução de problemas e insights personalizados de acordo com suas necessidades.

# Recursos

## AWS blogs

- Acelerar seu ciclo de vida de desenvolvimento de software com o Amazon Q
- Reimaginando o desenvolvimento de software com o Amazon Q Developer Agent
- Cinco exemplos de solução de problemas com o Amazon Q
- Personalize o Amazon Q Developer em você IDE com sua base de código privada
- Três maneiras pelas quais o agente Amazon Q Developer para transformação de código acelera as atualizações do Java
- Aproveitando o Amazon Q Developer para depuração e manutenção eficientes de código
- Testando seus aplicativos com o Amazon Q Developer

## AWS documentação

- Guia do usuário do desenvolvedor Amazon Q
- Personalização do código do Amazon Q Developer
- Transformação de código do Amazon Q Developer

## AWS oficinas

- Dia de imersão para desenvolvedores do Amazon Q
- Workshop para desenvolvedores do Amazon Q — Criando o aplicativo Q-Words
- Workshop para desenvolvedores do Amazon Q — Criação de prompts eficazes

# Colaboradores

As seguintes pessoas contribuíram para este guia:

- Joe King, Cientista de dados sênior, AWS
- Prateek Gupta, líder da equipe — Sr. CAA, AWS
- Manohar Reddy Arranagu, arquiteto, DevOps AWS
- Soumik Roy, arquiteto de aplicativos em nuvem, AWS
- Sanket Shinde, consultor, AWS

# Histórico do documento

A tabela a seguir descreve alterações significativas feitas neste guia. Se desejar receber notificações sobre futuras atualizações, inscreva-se em um RSSfeed.

| Alteração | Descrição | Data |
| --- | --- | --- |
| Publicação inicial | — | 16 de agosto de 2024 |

# AWS Glossário de orientação prescritiva

A seguir estão os termos comumente usados em estratégias, guias e padrões fornecidos pela Orientação AWS Prescritiva. Para sugerir entradas, use o link Fornecer feedback no final do glossário.

## Números

7 Rs

Sete estratégias comuns de migração para mover aplicações para a nuvem. Essas estratégias baseiam-se nos 5 Rs identificados pela Gartner em 2011 e consistem em:

- Refatorar/rearquitetar: mova uma aplicação e modifique sua arquitetura aproveitando ao máximo os recursos nativos de nuvem para melhorar a agilidade, a performance e a escalabilidade. Isso normalmente envolve a portabilidade do sistema operacional e do banco de dados. Exemplo: migre seu banco de dados Oracle local para a edição compatível com o Amazon Aurora PostgreSQL.
- Redefinir a plataforma (mover e redefinir [mover e redefinir (lift-and-reshape]): mova uma aplicação para a nuvem e introduza algum nível de otimização a fim de aproveitar os recursos da nuvem. Exemplo: Migre seu banco de dados Oracle local para o Amazon Relational Database Service (Amazon RDS) for Oracle no. Nuvem AWS
- Recomprar (drop and shop): mude para um produto diferente, normalmente migrando de uma licença tradicional para um modelo SaaS. Exemplo: migre seu sistema de gerenciamento de relacionamento com o cliente (CRM) para a Salesforce.com.
- Redefinir a hospedagem (mover sem alterações [lift-and-shift])mover uma aplicação para a nuvem sem fazer nenhuma alteração a fim de aproveitar os recursos da nuvem. Exemplo: Migre seu banco de dados Oracle local para o Oracle em uma EC2 instância no. Nuvem AWS
- Realocar (mover o hipervisor sem alterações [hypervisor-level lift-and-shift]): mover a infraestrutura para a nuvem sem comprar novo hardware, reescrever aplicações ou modificar suas operações existentes. Você migra servidores de uma plataforma local para um serviço em nuvem para a mesma plataforma. Exemplo: Migrar um Microsoft Hyper-V aplicativo para o. AWS
- Reter (revisitar): mantenha as aplicações em seu ambiente de origem. Isso pode incluir aplicações que exigem grande refatoração, e você deseja adiar esse trabalho para um

momento posterior, e aplicações antigas que você deseja manter porque não há justificativa comercial para migrá-las.

- Retirar: desative ou remova aplicações que não são mais necessárias em seu ambiente de origem.

# A

ABAC

Consulte controle de acesso baseado em atributos.

serviços abstratos

Veja os serviços gerenciados.

ACID

Veja atomicidade, consistência, isolamento, durabilidade.

migração ativa-ativa

Um método de migração de banco de dados no qual os bancos de dados de origem e de destino são mantidos em sincronia (por meio de uma ferramenta de replicação bidirecional ou operações de gravação dupla), e ambos os bancos de dados lidam com transações de aplicações conectadas durante a migração. Esse método oferece suporte à migração em lotes pequenos e controlados, em vez de exigir uma substituição única. É mais flexível, mas exige mais trabalho do que a migração ativa-passiva.

migração ativa-passiva

Um método de migração de banco de dados no qual os bancos de dados de origem e de destino são mantidos em sincronia, mas somente o banco de dados de origem manipula as transações dos aplicativos de conexão enquanto os dados são replicados no banco de dados de destino. O banco de dados de destino não aceita nenhuma transação durante a migração.

função agregada

Uma função SQL que opera em um grupo de linhas e calcula um único valor de retorno para o grupo. Exemplos de funções agregadas incluem `SUM` e. `MAX`

AI

Veja a inteligência artificial.

AIOps

Veja as operações de inteligência artificial.

anonimização

O processo de excluir permanentemente informações pessoais em um conjunto de dados. A anonimização pode ajudar a proteger a privacidade pessoal. Dados anônimos não são mais considerados dados pessoais.

antipadrões

Uma solução frequentemente usada para um problema recorrente em que a solução é contraproducente, ineficaz ou menos eficaz do que uma alternativa.

controle de aplicativos

Uma abordagem de segurança que permite o uso somente de aplicativos aprovados para ajudar a proteger um sistema contra malware.

portfólio de aplicações

Uma coleção de informações detalhadas sobre cada aplicação usada por uma organização, incluindo o custo para criar e manter a aplicação e seu valor comercial. Essas informações são fundamentais para o processo de descoberta e análise de portfólio e ajudam a identificar e priorizar as aplicações a serem migradas, modernizadas e otimizadas.

inteligência artificial (IA)

O campo da ciência da computação que se dedica ao uso de tecnologias de computação para desempenhar funções cognitivas normalmente associadas aos humanos, como aprender, resolver problemas e reconhecer padrões. Para obter mais informações, consulte O que é inteligência artificial?

operações de inteligência artificial (AIOps)

O processo de usar técnicas de machine learning para resolver problemas operacionais, reduzir incidentes operacionais e intervenção humana e aumentar a qualidade do serviço. Para obter mais informações sobre como AIOps é usado na estratégia de AWS migração, consulte o guia de integração de operações.

criptografia assimétrica

Um algoritmo de criptografia que usa um par de chaves, uma chave pública para criptografia e uma chave privada para descriptografia. É possível compartilhar a chave pública porque ela não é usada na descriptografia, mas o acesso à chave privada deve ser altamente restrito.

atomicidade, consistência, isolamento, durabilidade (ACID)

Um conjunto de propriedades de software que garantem a validade dos dados e a confiabilidade operacional de um banco de dados, mesmo no caso de erros, falhas de energia ou outros problemas.

controle de acesso por atributo (ABAC)

A prática de criar permissões minuciosas com base nos atributos do usuário, como departamento, cargo e nome da equipe. Para obter mais informações, consulte ABAC AWS na documentação AWS Identity and Access Management (IAM).

fonte de dados autorizada

Um local onde você armazena a versão principal dos dados, que é considerada a fonte de informações mais confiável. Você pode copiar dados da fonte de dados autorizada para outros locais com o objetivo de processar ou modificar os dados, como anonimizá-los, redigi-los ou pseudonimizá-los.

Zona de disponibilidade

Um local distinto dentro de um Região da AWS que está isolado de falhas em outras zonas de disponibilidade e fornece conectividade de rede barata e de baixa latência a outras zonas de disponibilidade na mesma região.

AWS Estrutura de adoção da nuvem (AWS CAF)

Uma estrutura de diretrizes e melhores práticas AWS para ajudar as organizações a desenvolver um plano eficiente e eficaz para migrar com sucesso para a nuvem. AWS O CAF organiza a orientação em seis áreas de foco chamadas perspectivas: negócios, pessoas, governança, plataforma, segurança e operações. As perspectivas de negócios, pessoas e governança têm como foco habilidades e processos de negócios; as perspectivas de plataforma, segurança e operações concentram-se em habilidades e processos técnicos. Por exemplo, a perspectiva das pessoas tem como alvo as partes interessadas que lidam com recursos humanos (RH), funções de pessoal e gerenciamento de pessoal. Nessa perspectiva, o AWS CAF fornece orientação para desenvolvimento, treinamento e comunicação de pessoas para ajudar a preparar a organização

para a adoção bem-sucedida da nuvem. Para obter mais informações, consulte o site da AWS CAF e o whitepaper da AWS CAF.

AWS Estrutura de qualificação da carga de trabalho (AWS WQF)

Uma ferramenta que avalia as cargas de trabalho de migração do banco de dados, recomenda estratégias de migração e fornece estimativas de trabalho. AWS O WQF está incluído com AWS Schema Conversion Tool ()AWS SCT. Ela analisa esquemas de banco de dados e objetos de código, código de aplicações, dependências e características de performance, além de fornecer relatórios de avaliação.

# B

bot ruim

Um bot destinado a perturbar ou causar danos a indivíduos ou organizações.

BCP

Veja o planejamento de continuidade de negócios.

gráfico de comportamento

Uma visualização unificada e interativa do comportamento e das interações de recursos ao longo do tempo. É possível usar um gráfico de comportamento com o Amazon Detective para examinar tentativas de login malsucedidas, chamadas de API suspeitas e ações similares. Para obter mais informações, consulte Dados em um gráfico de comportamento na documentação do Detective.

sistema big-endian

Um sistema que armazena o byte mais significativo antes. Veja também endianness.

classificação binária

Um processo que prevê um resultado binário (uma de duas classes possíveis). Por exemplo, seu modelo de ML pode precisar prever problemas como “Este e-mail é ou não é spam?” ou “Este produto é um livro ou um carro?”

filtro de bloom

Uma estrutura de dados probabilística e eficiente em termos de memória que é usada para testar se um elemento é membro de um conjunto.

blue/green deployment (implantação azul/verde)

Uma estratégia de implantação em que você cria dois ambientes separados, mas idênticos. Você executa a versão atual do aplicativo em um ambiente (azul) e a nova versão do aplicativo no outro ambiente (verde). Essa estratégia ajuda você a reverter rapidamente com o mínimo de impacto.

bot

Um aplicativo de software que executa tarefas automatizadas pela Internet e simula a atividade ou interação humana. Alguns bots são úteis ou benéficos, como rastreadores da Web que indexam informações na Internet. Alguns outros bots, conhecidos como bots ruins, têm como objetivo perturbar ou causar danos a indivíduos ou organizações.

botnet

Redes de bots infectadas por malware e sob o controle de uma única parte, conhecidas como pastor de bots ou operador de bots. As redes de bots são o mecanismo mais conhecido para escalar bots e seu impacto.

ramo

Uma área contida de um repositório de código. A primeira ramificação criada em um repositório é a ramificação principal. Você pode criar uma nova ramificação a partir de uma ramificação existente e, em seguida, desenvolver recursos ou corrigir bugs na nova ramificação. Uma ramificação que você cria para gerar um recurso é comumente chamada de ramificação de recurso. Quando o recurso estiver pronto para lançamento, você mesclará a ramificação do recurso de volta com a ramificação principal. Para obter mais informações, consulte Sobre filiais (GitHub documentação).

acesso em vidro quebrado

Em circunstâncias excepcionais e por meio de um processo aprovado, um meio rápido para um usuário obter acesso a um Conta da AWS que ele normalmente não tem permissão para acessar. Para obter mais informações, consulte o indicador Implementar procedimentos de quebra de vidro na orientação do Well-Architected AWS .

estratégia brownfield

A infraestrutura existente em seu ambiente. Ao adotar uma estratégia brownfield para uma arquitetura de sistema, você desenvolve a arquitetura de acordo com as restrições dos sistemas e da infraestrutura atuais. Se estiver expandindo a infraestrutura existente, poderá combinar as estratégias brownfield e greenfield.

cache do buffer

A área da memória em que os dados acessados com mais frequência são armazenados.

capacidade de negócios

O que uma empresa faz para gerar valor (por exemplo, vendas, atendimento ao cliente ou marketing). As arquiteturas de microsserviços e as decisões de desenvolvimento podem ser orientadas por recursos de negócios. Para obter mais informações, consulte a seção Organizados de acordo com as capacidades de negócios do whitepaper Executar microsserviços conteinerizados na AWS.

planejamento de continuidade de negócios (BCP)

Um plano que aborda o impacto potencial de um evento disruptivo, como uma migração em grande escala, nas operações e permite que uma empresa retome as operações rapidamente.

# C

CAF

Consulte Estrutura de adoção da AWS nuvem.

implantação canária

O lançamento lento e incremental de uma versão para usuários finais. Quando estiver confiante, você implanta a nova versão e substituirá a versão atual em sua totalidade.

CCoE

Veja o Centro de Excelência em Nuvem.

CDC

Veja a captura de dados de alterações.

captura de dados de alterações (CDC)

O processo de rastrear alterações em uma fonte de dados, como uma tabela de banco de dados, e registrar metadados sobre a alteração. É possível usar o CDC para várias finalidades, como auditar ou replicar alterações em um sistema de destino para manter a sincronização.

engenharia do caos

Introduzir intencionalmente falhas ou eventos disruptivos para testar a resiliência de um sistema. Você pode usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estressam suas AWS cargas de trabalho e avaliar sua resposta.

CI/CD

Veja a integração e a entrega contínuas.

classificação

Um processo de categorização que ajuda a gerar previsões. Os modelos de ML para problemas de classificação predizem um valor discreto. Os valores discretos são sempre diferentes uns dos outros. Por exemplo, um modelo pode precisar avaliar se há ou não um carro em uma imagem.

criptografia no lado do cliente

Criptografia de dados localmente, antes que o alvo os AWS service (Serviço da AWS) receba.

Centro de excelência em nuvem (CCoE)

Uma equipe multidisciplinar que impulsiona os esforços de adoção da nuvem em toda a organização, incluindo o desenvolvimento de práticas recomendas de nuvem, a mobilização de recursos, o estabelecimento de cronogramas de migração e a liderança da organização em transformações em grande escala. Para obter mais informações, consulte as publicações CCo E no Blog de Estratégia Nuvem AWS Empresarial.

computação em nuvem

A tecnologia de nuvem normalmente usada para armazenamento de dados remoto e gerenciamento de dispositivos de IoT. A computação em nuvem geralmente está conectada à tecnologia de computação de ponta.

modelo operacional em nuvem

Em uma organização de TI, o modelo operacional usado para criar, amadurecer e otimizar um ou mais ambientes de nuvem. Para obter mais informações, consulte Criar seu modelo operacional de nuvem.

estágios de adoção da nuvem

As quatro fases pelas quais as organizações normalmente passam quando migram para o Nuvem AWS:

- Projeto: executar alguns projetos relacionados à nuvem para fins de prova de conceito e aprendizado
- Fundação — Fazer investimentos fundamentais para escalar sua adoção da nuvem (por exemplo, criar uma landing zone, definir um CCo E, estabelecer um modelo de operações)
- Migração: migrar aplicações individuais
- Reinvenção: otimizar produtos e serviços e inovar na nuvem

Esses estágios foram definidos por Stephen Orban na postagem do blog The Journey Toward Cloud-First & the Stages of Adoption no blog de estratégia Nuvem AWS empresarial. Para obter informações sobre como eles se relacionam com a estratégia de AWS migração, consulte o guia de preparação para migração.

CMDB

Consulte o banco de dados de gerenciamento de configuração.

repositório de código

Um local onde o código-fonte e outros ativos, como documentação, amostras e scripts, são armazenados e atualizados por meio de processos de controle de versão. Os repositórios de nuvem comuns incluem GitHub ouBitbucket Cloud. Cada versão do código é chamada de ramificação. Em uma estrutura de microsserviços, cada repositório é dedicado a uma única peça de funcionalidade. Um único pipeline de CI/CD pode usar vários repositórios.

cache frio

Um cache de buffer que está vazio, não está bem preenchido ou contém dados obsoletos ou irrelevantes. Isso afeta a performance porque a instância do banco de dados deve ler da memória principal ou do disco, um processo que é mais lento do que a leitura do cache do buffer.

dados frios

Dados que raramente são acessados e geralmente são históricos. Ao consultar esse tipo de dados, consultas lentas geralmente são aceitáveis. Mover esses dados para níveis ou classes de armazenamento de baixo desempenho e menos caros pode reduzir os custos.

visão computacional (CV)

Um campo da IA que usa aprendizado de máquina para analisar e extrair informações de formatos visuais, como imagens e vídeos digitais. Por exemplo, a Amazon SageMaker AI fornece algoritmos de processamento de imagem para CV.

desvio de configuração

Para uma carga de trabalho, uma alteração de configuração em relação ao estado esperado. Isso pode fazer com que a carga de trabalho se torne incompatível e, normalmente, é gradual e não intencional.

banco de dados de gerenciamento de configuração (CMDB)

Um repositório que armazena e gerencia informações sobre um banco de dados e seu ambiente de TI, incluindo componentes de hardware e software e suas configurações. Normalmente, os dados de um CMDB são usados no estágio de descoberta e análise do portfólio da migração.

pacote de conformidade

Um conjunto de AWS Config regras e ações de remediação que você pode montar para personalizar suas verificações de conformidade e segurança. Você pode implantar um pacote de conformidade como uma entidade única em uma Conta da AWS região ou em uma organização usando um modelo YAML. Para obter mais informações, consulte Pacotes de conformidade na documentação. AWS Config

integração contínua e entrega contínua (CI/CD)

O processo de automatizar os estágios de origem, criação, teste, preparação e produção do processo de lançamento do software. CI/CD é comumente descrito como um pipeline. CI/CD pode ajudá-lo a automatizar processos, melhorar a produtividade, melhorar a qualidade do código e entregar com mais rapidez. Para obter mais informações, consulte Benefícios da entrega contínua. CD também pode significar implantação contínua. Para obter mais informações, consulte Entrega contínua versus implantação contínua.

CV

Veja visão computacional.

# D

dados em repouso

Dados estacionários em sua rede, por exemplo, dados que estão em um armazenamento.

classificação de dados

Um processo para identificar e categorizar os dados em sua rede com base em criticalidade e confidencialidade. É um componente crítico de qualquer estratégia de gerenciamento de riscos de

segurança cibernética, pois ajuda a determinar os controles adequados de proteção e retenção para os dados. A classificação de dados é um componente do pilar de segurança no AWS Well-Architected Framework. Para obter mais informações, consulte Classificação de dados.

desvio de dados

Uma variação significativa entre os dados de produção e os dados usados para treinar um modelo de ML ou uma alteração significativa nos dados de entrada ao longo do tempo. O desvio de dados pode reduzir a qualidade geral, a precisão e a imparcialidade das previsões do modelo de ML.

dados em trânsito

Dados que estão se movendo ativamente pela sua rede, como entre os recursos da rede.

malha de dados

Uma estrutura arquitetônica que fornece propriedade de dados distribuída e descentralizada com gerenciamento e governança centralizados.

minimização de dados

O princípio de coletar e processar apenas os dados estritamente necessários. Praticar a minimização de dados no Nuvem AWS pode reduzir os riscos de privacidade, os custos e a pegada de carbono de sua análise.

perímetro de dados

Um conjunto de proteções preventivas em seu AWS ambiente que ajudam a garantir que somente identidades confiáveis acessem recursos confiáveis das redes esperadas. Para obter mais informações, consulte Construindo um perímetro de dados em. AWS

pré-processamento de dados

A transformação de dados brutos em um formato que seja facilmente analisado por seu modelo de ML. O pré-processamento de dados pode significar a remoção de determinadas colunas ou linhas e o tratamento de valores ausentes, inconsistentes ou duplicados.

proveniência dos dados

O processo de rastrear a origem e o histórico dos dados ao longo de seu ciclo de vida, por exemplo, como os dados foram gerados, transmitidos e armazenados.

titular dos dados

Um indivíduo cujos dados estão sendo coletados e processados.

data warehouse

Um sistema de gerenciamento de dados que oferece suporte à inteligência comercial, como análises. Os data warehouses geralmente contêm grandes quantidades de dados históricos e geralmente são usados para consultas e análises.

linguagem de definição de dados (DDL)

Instruções ou comandos para criar ou modificar a estrutura de tabelas e objetos em um banco de dados.

linguagem de manipulação de dados (DML)

Instruções ou comandos para modificar (inserir, atualizar e excluir) informações em um banco de dados.

DDL

Consulte a linguagem de definição de banco de dados.

deep ensemble

A combinação de vários modelos de aprendizado profundo para gerar previsões. Os deep ensembles podem ser usados para produzir uma previsão mais precisa ou para estimar a incerteza nas previsões.

Aprendizado profundo

Um subcampo do ML que usa várias camadas de redes neurais artificiais para identificar o mapeamento entre os dados de entrada e as variáveis-alvo de interesse.

defense-in-depth

Uma abordagem de segurança da informação na qual uma série de mecanismos e controles de segurança são cuidadosamente distribuídos por toda a rede de computadores para proteger a confidencialidade, a integridade e a disponibilidade da rede e dos dados nela contidos. Ao adotar essa estratégia AWS, você adiciona vários controles em diferentes camadas da AWS Organizations estrutura para ajudar a proteger os recursos. Por exemplo, uma defense-in-depth abordagem pode combinar autenticação multifatorial, segmentação de rede e criptografia.

administrador delegado

Em AWS Organizations, um serviço compatível pode registrar uma conta de AWS membro para administrar as contas da organização e gerenciar as permissões desse serviço. Essa conta

é chamada de administrador delegado para esse serviço Para obter mais informações e uma lista de serviços compatíveis, consulte Serviços que funcionam com o AWS Organizations na documentação do AWS Organizations .

implantação

O processo de criar uma aplicação, novos recursos ou correções de código disponíveis no ambiente de destino. A implantação envolve a implementação de mudanças em uma base de código e, em seguida, a criação e execução dessa base de código nos ambientes da aplicação

ambiente de desenvolvimento

Veja o ambiente.

controle detectivo

Um controle de segurança projetado para detectar, registrar e alertar após a ocorrência de um evento. Esses controles são uma segunda linha de defesa, alertando você sobre eventos de segurança que contornaram os controles preventivos em vigor. Para obter mais informações, consulte Controles detectivos em Como implementar controles de segurança na AWS.

mapeamento do fluxo de valor de desenvolvimento (DVSM)

Um processo usado para identificar e priorizar restrições que afetam negativamente a velocidade e a qualidade em um ciclo de vida de desenvolvimento de software. O DVSM estende o processo de mapeamento do fluxo de valor originalmente projetado para práticas de manufatura enxuta. Ele se concentra nas etapas e equipes necessárias para criar e movimentar valor por meio do processo de desenvolvimento de software.

gêmeo digital

Uma representação virtual de um sistema real, como um prédio, fábrica, equipamento industrial ou linha de produção. Os gêmeos digitais oferecem suporte à manutenção preditiva, ao monitoramento remoto e à otimização da produção.

tabela de dimensões

Em um esquema em estrela, uma tabela menor que contém atributos de dados sobre dados quantitativos em uma tabela de fatos. Os atributos da tabela de dimensões geralmente são campos de texto ou números discretos que se comportam como texto. Esses atributos são comumente usados para restringir consultas, filtrar e rotular conjuntos de resultados.

desastre

Um evento que impede que uma workload ou sistema cumpra seus objetivos de negócios em seu local principal de implantação. Esses eventos podem ser desastres naturais, falhas técnicas ou o resultado de ações humanas, como configuração incorreta não intencional ou ataque de malware.

Recuperação de desastres (RD)

A estratégia e o processo que você usa para minimizar o tempo de inatividade e a perda de dados causados por um desastre. Para obter mais informações, consulte Recuperação de desastres de cargas de trabalho em AWS: Recuperação na nuvem no AWS Well-Architected Framework.

DML

Veja a linguagem de manipulação de banco de dados.

design orientado por domínio

Uma abordagem ao desenvolvimento de um sistema de software complexo conectando seus componentes aos domínios em evolução, ou principais metas de negócios, atendidos por cada componente. Esse conceito foi introduzido por Eric Evans em seu livro, Design orientado por domínio: lidando com a complexidade no coração do software (Boston: Addison-Wesley Professional, 2003). Para obter informações sobre como usar o design orientado por domínio com o padrão strangler fig, consulte Modernizar incrementalmente os serviços web herdados do Microsoft ASP.NET (ASMX) usando contêineres e o Amazon API Gateway.

DR

Veja a recuperação de desastres.

detecção de deriva

Rastreando desvios de uma configuração básica. Por exemplo, você pode usar AWS CloudFormation para detectar desvios nos recursos do sistema ou AWS Control Tower para detectar mudanças em seu landing zone que possam afetar a conformidade com os requisitos de governança.

DVSM

Veja o mapeamento do fluxo de valor do desenvolvimento.

# E

EDA

Veja a análise exploratória de dados.

EDI

Veja intercâmbio eletrônico de dados.

computação de borda

A tecnologia que aumenta o poder computacional de dispositivos inteligentes nas bordas de uma rede de IoT. Quando comparada à computação em nuvem, a computação de ponta pode reduzir a latência da comunicação e melhorar o tempo de resposta.

intercâmbio eletrônico de dados (EDI)

A troca automatizada de documentos comerciais entre organizações. Para obter mais informações, consulte O que é intercâmbio eletrônico de dados.

Criptografia

Um processo de computação que transforma dados de texto simples, legíveis por humanos, em texto cifrado.

chave de criptografia

Uma sequência criptográfica de bits aleatórios que é gerada por um algoritmo de criptografia. As chaves podem variar em tamanho, e cada chave foi projetada para ser imprevisível e exclusiva.

endianismo

A ordem na qual os bytes são armazenados na memória do computador. Os sistemas big-endian armazenam o byte mais significativo antes. Os sistemas little-endian armazenam o byte menos significativo antes.

endpoint

Veja o endpoint do serviço.

serviço de endpoint

Um serviço que pode ser hospedado em uma nuvem privada virtual (VPC) para ser compartilhado com outros usuários. Você pode criar um serviço de endpoint com AWS PrivateLink e conceder permissões a outros diretores Contas da AWS ou a AWS Identity and Access Management (IAM).

Essas contas ou entidades principais podem se conectar ao serviço de endpoint de maneira privada criando endpoints da VPC de interface. Para obter mais informações, consulte Criar um serviço de endpoint na documentação do Amazon Virtual Private Cloud (Amazon VPC).

planejamento de recursos corporativos (ERP)

Um sistema que automatiza e gerencia os principais processos de negócios (como contabilidade, MES e gerenciamento de projetos) para uma empresa.

criptografia envelopada

O processo de criptografar uma chave de criptografia com outra chave de criptografia. Para obter mais informações, consulte Criptografia de envelope na documentação AWS Key Management Service (AWS KMS).

ambiente

Uma instância de uma aplicação em execução. Estes são tipos comuns de ambientes na computação em nuvem:

- ambiente de desenvolvimento: uma instância de uma aplicação em execução que está disponível somente para a equipe principal responsável pela manutenção da aplicação. Ambientes de desenvolvimento são usados para testar mudanças antes de promovê-las para ambientes superiores. Esse tipo de ambiente às vezes é chamado de ambiente de teste.
- ambientes inferiores: todos os ambientes de desenvolvimento para uma aplicação, como aqueles usados para compilações e testes iniciais.
- ambiente de produção: uma instância de uma aplicação em execução que os usuários finais podem acessar. Em um CI/CD pipeline, o ambiente de produção é o último ambiente de implantação.
- ambientes superiores: todos os ambientes que podem ser acessados por usuários que não sejam a equipe principal de desenvolvimento. Isso pode incluir um ambiente de produção, ambientes de pré-produção e ambientes para testes de aceitação do usuário.

epic

Em metodologias ágeis, categorias funcionais que ajudam a organizar e priorizar seu trabalho. Os epics fornecem uma descrição de alto nível dos requisitos e das tarefas de implementação. Por exemplo, os épicos de segurança AWS da CAF incluem gerenciamento de identidade e acesso, controles de detetive, segurança de infraestrutura, proteção de dados e resposta a incidentes. Para obter mais informações sobre epics na estratégia de migração da AWS , consulte o guia de implementação do programa.

ERP

Veja o planejamento de recursos corporativos.

análise exploratória de dados (EDA)

O processo de analisar um conjunto de dados para entender suas principais características. Você coleta ou agrega dados e, em seguida, realiza investigações iniciais para encontrar padrões, detectar anomalias e verificar suposições. O EDA é realizado por meio do cálculo de estatísticas resumidas e da criação de visualizações de dados.

# F

tabela de fatos

A tabela central em um esquema em estrela. Ele armazena dados quantitativos sobre as operações comerciais. Normalmente, uma tabela de fatos contém dois tipos de colunas: aquelas que contêm medidas e aquelas que contêm uma chave externa para uma tabela de dimensões.

falham rapidamente

Uma filosofia que usa testes frequentes e incrementais para reduzir o ciclo de vida do desenvolvimento. É uma parte essencial de uma abordagem ágil.

limite de isolamento de falhas

No Nuvem AWS, um limite, como uma zona de disponibilidade, Região da AWS um plano de controle ou um plano de dados, que limita o efeito de uma falha e ajuda a melhorar a resiliência das cargas de trabalho. Para obter mais informações, consulte Limites de isolamento de AWS falhas.

ramificação de recursos

Veja a filial.

recursos

Os dados de entrada usados para fazer uma previsão. Por exemplo, em um contexto de manufatura, os recursos podem ser imagens capturadas periodicamente na linha de fabricação.

importância do recurso

O quanto um recurso é importante para as previsões de um modelo. Isso geralmente é expresso como uma pontuação numérica que pode ser calculada por meio de várias técnicas, como

Shapley Additive Explanations (SHAP) e gradientes integrados. Para obter mais informações, consulte Interpretabilidade do modelo de aprendizado de máquina com AWS.

transformação de recursos

O processo de otimizar dados para o processo de ML, incluindo enriquecer dados com fontes adicionais, escalar valores ou extrair vários conjuntos de informações de um único campo de dados. Isso permite que o modelo de ML se beneficie dos dados. Por exemplo, se a data “2021-05-27 00:15:37” for dividida em “2021”, “maio”, “quinta” e “15”, isso poderá ajudar o algoritmo de aprendizado a aprender padrões diferenciados associados a diferentes componentes de dados.

solicitação rápida

Fornecer a um LLM um pequeno número de exemplos que demonstram a tarefa e o resultado desejado antes de solicitar que ele execute uma tarefa semelhante. Essa técnica é uma aplicação do aprendizado contextual, em que os modelos aprendem com exemplos (fotos) incorporados aos prompts. Solicitações rápidas podem ser eficazes para tarefas que exigem formatação, raciocínio ou conhecimento de domínio específicos. Veja também a solicitação zero-shot.

FGAC

Veja o controle de acesso refinado.

Controle de acesso refinado (FGAC)

O uso de várias condições para permitir ou negar uma solicitação de acesso.

migração flash-cut

Um método de migração de banco de dados que usa replicação contínua de dados por meio da captura de dados alterados para migrar dados no menor tempo possível, em vez de usar uma abordagem em fases. O objetivo é reduzir ao mínimo o tempo de inatividade.

FM

Veja o modelo da fundação.

modelo de fundação (FM)

Uma grande rede neural de aprendizado profundo que vem treinando em grandes conjuntos de dados generalizados e não rotulados. FMs são capazes de realizar uma ampla variedade de tarefas gerais, como entender a linguagem, gerar texto e imagens e conversar em linguagem natural. Para obter mais informações, consulte O que são modelos básicos.

# G

IA generativa

Um subconjunto de modelos de IA que foram treinados em grandes quantidades de dados e que podem usar uma simples solicitação de texto para criar novos conteúdos e artefatos, como imagens, vídeos, texto e áudio. Para obter mais informações, consulte O que é IA generativa.

bloqueio geográfico

Veja as restrições geográficas.

restrições geográficas (bloqueio geográfico)

Na Amazon CloudFront, uma opção para impedir que usuários em países específicos acessem distribuições de conteúdo. É possível usar uma lista de permissões ou uma lista de bloqueios para especificar países aprovados e banidos. Para obter mais informações, consulte Restringir a distribuição geográfica do seu conteúdo na CloudFront documentação.

Fluxo de trabalho do GitFlow

Uma abordagem na qual ambientes inferiores e superiores usam ramificações diferentes em um repositório de código-fonte. O fluxo de trabalho do Gitflow é considerado legado, e o fluxo de trabalho baseado em troncos é a abordagem moderna e preferida.

imagem dourada

Um instantâneo de um sistema ou software usado como modelo para implantar novas instâncias desse sistema ou software. Por exemplo, na manufatura, uma imagem dourada pode ser usada para provisionar software em vários dispositivos e ajudar a melhorar a velocidade, a escalabilidade e a produtividade nas operações de fabricação de dispositivos.

estratégia greenfield

A ausência de infraestrutura existente em um novo ambiente. Ao adotar uma estratégia greenfield para uma arquitetura de sistema, é possível selecionar todas as novas tecnologias sem a restrição da compatibilidade com a infraestrutura existente, também conhecida como brownfield. Se estiver expandindo a infraestrutura existente, poderá combinar as estratégias brownfield e greenfield.

barreira de proteção

Uma regra de alto nível que ajuda a governar recursos, políticas e conformidade em todas as unidades organizacionais ()OUs. Barreiras de proteção preventivas impõem políticas para

garantir o alinhamento a padrões de conformidade. Elas são implementadas usando políticas de controle de serviço e limites de permissões do IAM. Barreiras de proteção detectivas detectam violações de políticas e problemas de conformidade e geram alertas para remediação. Eles são implementados usando AWS Config, AWS Security Hub, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector e verificações personalizadas AWS Lambda .

# H

HA

Veja a alta disponibilidade.

migração heterogênea de bancos de dados

Migrar seu banco de dados de origem para um banco de dados de destino que usa um mecanismo de banco de dados diferente (por exemplo, Oracle para Amazon Aurora). A migração heterogênea geralmente faz parte de um esforço de redefinição da arquitetura, e converter o esquema pode ser uma tarefa complexa. O AWS fornece o AWS SCT para ajudar nas conversões de esquemas.

alta disponibilidade (HA)

A capacidade de uma workload operar continuamente, sem intervenção, em caso de desafios ou desastres. Os sistemas AH são projetados para realizar o failover automático, oferecer consistentemente desempenho de alta qualidade e lidar com diferentes cargas e falhas com impacto mínimo no desempenho.

modernização de historiador

Uma abordagem usada para modernizar e atualizar os sistemas de tecnologia operacional (OT) para melhor atender às necessidades do setor de manufatura. Um historiador é um tipo de banco de dados usado para coletar e armazenar dados de várias fontes em uma fábrica.

dados de retenção

Uma parte dos dados históricos rotulados que são retidos de um conjunto de dados usado para treinar um modelo de aprendizado de máquina. Você pode usar dados de retenção para avaliar o desempenho do modelo comparando as previsões do modelo com os dados de retenção.

migração homogênea de bancos de dados

Migrar seu banco de dados de origem para um banco de dados de destino que compartilha o mesmo mecanismo de banco de dados (por exemplo, Microsoft SQL Server para Amazon RDS para SQL Server). A migração homogênea geralmente faz parte de um esforço de redefinição da hospedagem ou da plataforma. É possível usar utilitários de banco de dados nativos para migrar o esquema.

dados quentes

Dados acessados com frequência, como dados em tempo real ou dados translacionais recentes. Esses dados normalmente exigem uma camada ou classe de armazenamento de alto desempenho para fornecer respostas rápidas às consultas.

hotfix

Uma correção urgente para um problema crítico em um ambiente de produção. Devido à sua urgência, um hotfix geralmente é feito fora do fluxo de trabalho normal de DevOps lançamento.

período de hipercuidados

Imediatamente após a substituição, o período em que uma equipe de migração gerencia e monitora as aplicações migradas na nuvem para resolver quaisquer problemas. Normalmente, a duração desse período é de 1 a 4 dias. No final do período de hipercuidados, a equipe de migração normalmente transfere a responsabilidade pelas aplicações para a equipe de operações de nuvem.

## eu

IaC

Veja a infraestrutura como código.

Política baseada em identidade

Uma política anexada a um ou mais diretores do IAM que define suas permissões no Nuvem AWS ambiente.

aplicação ociosa

Uma aplicação que tem um uso médio de CPU e memória entre 5 e 20% em um período de 90 dias. Em um projeto de migração, é comum retirar essas aplicações ou retê-las on-premises.

IIoT

Veja a Internet das Coisas industrial.

infraestrutura imutável

Um modelo que implanta uma nova infraestrutura para cargas de trabalho de produção em vez de atualizar, corrigir ou modificar a infraestrutura existente. Infraestruturas imutáveis são inerentemente mais consistentes, confiáveis e previsíveis do que infraestruturas mutáveis. Para obter mais informações, consulte as melhores práticas de implantação usando infraestrutura imutável no Well-Architected AWS Framework.

VPC de entrada (admissão)

Em uma arquitetura de AWS várias contas, uma VPC que aceita, inspeciona e roteia conexões de rede de fora de um aplicativo. A Arquitetura de Referência de AWS Segurança recomenda configurar sua conta de rede com entrada, saída e inspeção VPCs para proteger a interface bidirecional entre seu aplicativo e a Internet em geral.

migração incremental

Uma estratégia de substituição na qual você migra a aplicação em pequenas partes, em vez de realizar uma única substituição completa. Por exemplo, é possível mover inicialmente apenas alguns microsserviços ou usuários para o novo sistema. Depois de verificar se tudo está funcionando corretamente, mova os microsserviços ou usuários adicionais de forma incremental até poder descomissionar seu sistema herdado. Essa estratégia reduz os riscos associados a migrações de grande porte.

Indústria 4.0

Um termo que foi introduzido por Klaus Schwab em 2016 para se referir à modernização dos processos de fabricação por meio de avanços em conectividade, dados em tempo real, automação, análise e IA/ML.

infraestrutura

Todos os recursos e ativos contidos no ambiente de uma aplicação.

Infraestrutura como código (IaC)

O processo de provisionamento e gerenciamento da infraestrutura de uma aplicação por meio de um conjunto de arquivos de configuração. A IaC foi projetada para ajudar você a centralizar o gerenciamento da infraestrutura, padronizar recursos e escalar rapidamente para que novos ambientes sejam reproduzíveis, confiáveis e consistentes.

Internet industrial das coisas (IIoT)

O uso de sensores e dispositivos conectados à Internet nos setores industriais, como manufatura, energia, automotivo, saúde, ciências biológicas e agricultura. Para obter mais informações, consulte Criando uma estratégia de transformação digital industrial da Internet das Coisas (IIoT).

VPC de inspeção

Em uma arquitetura de AWS várias contas, uma VPC centralizada que gerencia as inspeções do tráfego de rede entre VPCs (na mesma ou em diferentes Regiões da AWS) a Internet e as redes locais. A Arquitetura de Referência de AWS Segurança recomenda configurar sua conta de rede com entrada, saída e inspeção VPCs para proteger a interface bidirecional entre seu aplicativo e a Internet em geral.

Internet das Coisas (IoT)

A rede de objetos físicos conectados com sensores ou processadores incorporados que se comunicam com outros dispositivos e sistemas pela Internet ou por uma rede de comunicação local. Para obter mais informações, consulte O que é IoT?

interpretabilidade

Uma característica de um modelo de machine learning que descreve o grau em que um ser humano pode entender como as previsões do modelo dependem de suas entradas. Para obter mais informações, consulte Interpretabilidade do modelo de aprendizado de máquina com AWS.

IoT

Consulte Internet das Coisas.

Biblioteca de informações de TI (ITIL)

Um conjunto de práticas recomendadas para fornecer serviços de TI e alinhar esses serviços a requisitos de negócios. A ITIL fornece a base para o ITSM.

Gerenciamento de serviços de TI (ITSM)

Atividades associadas a design, implementação, gerenciamento e suporte de serviços de TI para uma organização. Para obter informações sobre a integração de operações em nuvem com ferramentas de ITSM, consulte o guia de integração de operações.

ITIL

Consulte a biblioteca de informações de TI.

ITSM

Veja o gerenciamento de serviços de TI.

# L

controle de acesso baseado em etiqueta (LBAC)

Uma implementação do controle de acesso obrigatório (MAC) em que os usuários e os dados em si recebem explicitamente um valor de etiqueta de segurança. A interseção entre a etiqueta de segurança do usuário e a etiqueta de segurança dos dados determina quais linhas e colunas podem ser vistas pelo usuário.

zona de pouso

Uma landing zone é um AWS ambiente bem arquitetado, com várias contas, escalável e seguro. Um ponto a partir do qual suas organizações podem iniciar e implantar rapidamente workloads e aplicações com confiança em seu ambiente de segurança e infraestrutura. Para obter mais informações sobre zonas de pouso, consulte Configurar um ambiente da AWS com várias contas seguro e escalável.

modelo de linguagem grande (LLM)

Um modelo de IA de aprendizado profundo pré-treinado em uma grande quantidade de dados. Um LLM pode realizar várias tarefas, como responder perguntas, resumir documentos, traduzir texto para outros idiomas e completar frases. Para obter mais informações, consulte O que são LLMs.

migração de grande porte

Uma migração de 300 servidores ou mais.

LBAC

Veja controle de acesso baseado em etiquetas.

privilégio mínimo

A prática recomendada de segurança de conceder as permissões mínimas necessárias para executar uma tarefa. Para obter mais informações, consulte Aplicar permissões de privilégios mínimos na documentação do IAM.

mover sem alterações (lift-and-shift)

Veja 7 Rs.

sistema little-endian

Um sistema que armazena o byte menos significativo antes. Veja também endianness.

LLM

Veja um modelo de linguagem grande.

ambientes inferiores

Veja o ambiente.

# M

machine learning (ML)

Um tipo de inteligência artificial que usa algoritmos e técnicas para reconhecimento e aprendizado de padrões. O ML analisa e aprende com dados gravados, por exemplo, dados da Internet das Coisas (IoT), para gerar um modelo estatístico baseado em padrões. Para obter mais informações, consulte Machine learning.

ramificação principal

Veja a filial.

malware

Software projetado para comprometer a segurança ou a privacidade do computador. O malware pode interromper os sistemas do computador, vazar informações confidenciais ou obter acesso não autorizado. Exemplos de malware incluem vírus, worms, ransomware, cavalos de Tróia, spyware e keyloggers.

serviços gerenciados

Serviços da AWS para o qual AWS opera a camada de infraestrutura, o sistema operacional e as plataformas, e você acessa os endpoints para armazenar e recuperar dados. O Amazon Simple Storage Service (Amazon S3) e o Amazon DynamoDB são exemplos de serviços gerenciados. Eles também são conhecidos como serviços abstratos.

sistema de execução de manufatura (MES)

Um sistema de software para rastrear, monitorar, documentar e controlar processos de produção que convertem matérias-primas em produtos acabados no chão de fábrica.

MAP

Consulte Migration Acceleration Program.

mecanismo

Um processo completo no qual você cria uma ferramenta, impulsiona a adoção da ferramenta e, em seguida, inspeciona os resultados para fazer ajustes. Um mecanismo é um ciclo que se reforça e se aprimora à medida que opera. Para obter mais informações, consulte Construindo mecanismos no AWS Well-Architected Framework.

conta de membro

Todos, Contas da AWS exceto a conta de gerenciamento, que fazem parte de uma organização em AWS Organizations. Uma conta só pode ser membro de uma organização de cada vez.

MES

Veja o sistema de execução de manufatura.

Transporte de telemetria de enfileiramento de mensagens (MQTT)

Um protocolo de comunicação leve machine-to-machine (M2M), baseado no padrão de publicação/assinatura, para dispositivos de IoT com recursos limitados.

microsserviço

Um serviço pequeno e independente que se comunica de forma bem definida APIs e normalmente é de propriedade de equipes pequenas e independentes. Por exemplo, um sistema de seguradora pode incluir microsserviços que mapeiam as capacidades comerciais, como vendas ou marketing, ou subdomínios, como compras, reclamações ou análises. Os benefícios dos microsserviços incluem agilidade, escalabilidade flexível, fácil implantação, código reutilizável e resiliência. Para obter mais informações, consulte Integração de microsserviços usando serviços sem AWS servidor.

arquitetura de microsserviços

Uma abordagem à criação de aplicações com componentes independentes que executam cada processo de aplicação como um microsserviço. Esses microsserviços se comunicam por meio

de uma interface bem definida usando leveza. APIs Cada microsserviço nessa arquitetura pode ser atualizado, implantado e escalado para atender à demanda por funções específicas de uma aplicação. Para obter mais informações, consulte Implementação de microsserviços em. AWS

Programa de Aceleração da Migração (MAP)

Um AWS programa que fornece suporte de consultoria, treinamento e serviços para ajudar as organizações a criar uma base operacional sólida para migrar para a nuvem e ajudar a compensar o custo inicial das migrações. O MAP inclui uma metodologia de migração para executar migrações legadas de forma metódica e um conjunto de ferramentas para automatizar e acelerar cenários comuns de migração.

migração em escala

O processo de mover a maior parte do portfólio de aplicações para a nuvem em ondas, com mais aplicações sendo movidas em um ritmo mais rápido a cada onda. Essa fase usa as práticas recomendadas e lições aprendidas nas fases anteriores para implementar uma fábrica de migração de equipes, ferramentas e processos para agilizar a migração de workloads por meio de automação e entrega ágeis. Esta é a terceira fase da estratégia de migração para a AWS.

fábrica de migração

Equipes multifuncionais que simplificam a migração de workloads por meio de abordagens automatizadas e ágeis. As equipes da fábrica de migração geralmente incluem operações, analistas e proprietários de negócios, engenheiros de migração, desenvolvedores e DevOps profissionais que trabalham em sprints. Entre 20 e 50% de um portfólio de aplicações corporativas consiste em padrões repetidos que podem ser otimizados por meio de uma abordagem de fábrica. Para obter mais informações, consulte discussão sobre fábricas de migração e o guia do Cloud Migration Factory neste conjunto de conteúdo.

metadados de migração

As informações sobre a aplicação e o servidor necessárias para concluir a migração. Cada padrão de migração exige um conjunto de metadados de migração diferente. Exemplos de metadados de migração incluem a sub-rede, o grupo de segurança e AWS a conta de destino.

padrão de migração

Uma tarefa de migração repetível que detalha a estratégia de migração, o destino da migração e a aplicação ou o serviço de migração usado. Exemplo: rehospede a migração para a Amazon EC2 com o AWS Application Migration Service.

Avaliação de Portfólio para Migração (MPA)

Uma ferramenta on-line que fornece informações para validar o caso de negócios para migrar para o. Nuvem AWS O MPA fornece avaliação detalhada do portfólio (dimensionamento correto do servidor, preços, comparações de TCO, análise de custos de migração), bem como planejamento de migração (análise e coleta de dados de aplicações, agrupamento de aplicações, priorização de migração e planejamento de ondas). A ferramenta MPA (requer login) está disponível gratuitamente para todos os AWS consultores e consultores parceiros da APN.

Avaliação de Preparação para Migração (MRA)

O processo de obter insights sobre o status de prontidão de uma organização para a nuvem, identificar pontos fortes e fracos e criar um plano de ação para fechar as lacunas identificadas, usando o CAF. AWS Para mais informações, consulte o guia de preparação para migração. A MRA é a primeira fase da estratégia de migração para a AWS.

estratégia de migração

A abordagem usada para migrar uma carga de trabalho para o. Nuvem AWS Para obter mais informações, consulte a entrada de 7 Rs neste glossário e consulte Mobilize sua organização para acelerar migrações em grande escala.

ML

Veja o aprendizado de máquina.

modernização

Transformar uma aplicação desatualizada (herdada ou monolítica) e sua infraestrutura em um sistema ágil, elástico e altamente disponível na nuvem para reduzir custos, ganhar eficiência e aproveitar as inovações. Para obter mais informações, consulte Estratégia para modernizar aplicativos no Nuvem AWS.

avaliação de preparação para modernização

Uma avaliação que ajuda a determinar a preparação para modernização das aplicações de uma organização. Ela identifica benefícios, riscos e dependências e determina o quão bem a organização pode acomodar o estado futuro dessas aplicações. O resultado da avaliação é um esquema da arquitetura de destino, um roteiro que detalha as fases de desenvolvimento e os marcos do processo de modernização e um plano de ação para abordar as lacunas identificadas. Para obter mais informações, consulte Avaliação da prontidão para modernização de aplicativos no. Nuvem AWS

aplicações monolíticas (monólitos)

Aplicações que são executadas como um único serviço com processos fortemente acoplados. As aplicações monolíticas apresentam várias desvantagens. Se um recurso da aplicação apresentar um aumento na demanda, toda a arquitetura deverá ser escalada. Adicionar ou melhorar os recursos de uma aplicação monolítica também se torna mais complexo quando a base de código cresce. Para resolver esses problemas, é possível criar uma arquitetura de microsserviços. Para obter mais informações, consulte Decompor monólitos em microsserviços.

MAPA

Consulte Avaliação do portfólio de migração.

MQTT

Consulte Transporte de telemetria de enfileiramento de mensagens.

classificação multiclasse

Um processo que ajuda a gerar previsões para várias classes (prevendo um ou mais de dois resultados). Por exemplo, um modelo de ML pode perguntar “Este produto é um livro, um carro ou um telefone?” ou “Qual categoria de produtos é mais interessante para este cliente?”

infraestrutura mutável

Um modelo que atualiza e modifica a infraestrutura existente para cargas de trabalho de produção. Para melhorar a consistência, confiabilidade e previsibilidade, o AWS Well-Architected Framework recomenda o uso de infraestrutura imutável como uma prática recomendada.

# O

OAC

Veja o controle de acesso de origem.

CARVALHO

Veja a identidade de acesso de origem.

OCM

Veja o gerenciamento de mudanças organizacionais.

migração offline

Um método de migração no qual a workload de origem é desativada durante o processo de migração. Esse método envolve tempo de inatividade prolongado e geralmente é usado para workloads pequenas e não críticas.

OI

Veja a integração de operações.

OLA

Veja o contrato em nível operacional.

migração online

Um método de migração no qual a workload de origem é copiada para o sistema de destino sem ser colocada offline. As aplicações conectadas à workload podem continuar funcionando durante a migração. Esse método envolve um tempo de inatividade nulo ou mínimo e normalmente é usado para workloads essenciais para a produção.

OPC-UA

Consulte Comunicação de processo aberto — Arquitetura unificada.

Comunicação de processo aberto — Arquitetura unificada (OPC-UA)

Um protocolo de comunicação machine-to-machine (M2M) para automação industrial. O OPC-UA fornece um padrão de interoperabilidade com esquemas de criptografia, autenticação e autorização de dados.

acordo de nível operacional (OLA)

Um acordo que esclarece o que os grupos funcionais de TI prometem oferecer uns aos outros para apoiar um acordo de serviço (SLA).

análise de prontidão operacional (ORR)

Uma lista de verificação de perguntas e melhores práticas associadas que ajudam você a entender, avaliar, prevenir ou reduzir o escopo de incidentes e possíveis falhas. Para obter mais informações, consulte Operational Readiness Reviews (ORR) no Well-Architected AWS Framework.

tecnologia operacional (OT)

Sistemas de hardware e software que funcionam com o ambiente físico para controlar operações, equipamentos e infraestrutura industriais. Na manufatura, a integração dos sistemas OT e de tecnologia da informação (TI) é o foco principal das transformações da Indústria 4.0.

integração de operações (OI)

O processo de modernização das operações na nuvem, que envolve planejamento de preparação, automação e integração. Para obter mais informações, consulte o guia de integração de operações.

trilha organizacional

Uma trilha criada por ela AWS CloudTrail registra todos os eventos de todos Contas da AWS em uma organização em AWS Organizations. Essa trilha é criada em cada Conta da AWS que faz parte da organização e monitora a atividade em cada conta. Para obter mais informações, consulte Criação de uma trilha para uma organização na CloudTrail documentação.

gerenciamento de alterações organizacionais (OCM)

Uma estrutura para gerenciar grandes transformações de negócios disruptivas de uma perspectiva de pessoas, cultura e liderança. O OCM ajuda as organizações a se prepararem e fazerem a transição para novos sistemas e estratégias, acelerando a adoção de alterações, abordando questões de transição e promovendo mudanças culturais e organizacionais. Na estratégia de AWS migração, essa estrutura é chamada de aceleração de pessoas, devido à velocidade de mudança exigida nos projetos de adoção da nuvem. Para obter mais informações, consulte o guia do OCM.

controle de acesso de origem (OAC)

Em CloudFront, uma opção aprimorada para restringir o acesso para proteger seu conteúdo do Amazon Simple Storage Service (Amazon S3). O OAC oferece suporte a todos os buckets S3 Regiões da AWS, criptografia do lado do servidor com AWS KMS (SSE-KMS) e solicitações dinâmicas ao bucket S3. `PUT` `DELETE`

Identidade do acesso de origem (OAI)

Em CloudFront, uma opção para restringir o acesso para proteger seu conteúdo do Amazon S3. Quando você usa o OAI, CloudFront cria um principal com o qual o Amazon S3 pode se autenticar. Os diretores autenticados podem acessar o conteúdo em um bucket do S3 somente por meio de uma distribuição específica. CloudFront Veja também OAC, que fornece um controle de acesso mais granular e aprimorado.

ORR

Veja a análise de prontidão operacional.

OT

Veja a tecnologia operacional.

VPC de saída (egresso)

Em uma arquitetura de AWS várias contas, uma VPC que gerencia conexões de rede que são iniciadas de dentro de um aplicativo. A Arquitetura de Referência de AWS Segurança recomenda configurar sua conta de rede com entrada, saída e inspeção VPCs para proteger a interface bidirecional entre seu aplicativo e a Internet em geral.

# P

limite de permissões

Uma política de gerenciamento do IAM anexada a entidades principais do IAM para definir as permissões máximas que o usuário ou perfil podem ter. Para obter mais informações, consulte Limites de permissões na documentação do IAM.

Informações de identificação pessoal (PII)

Informações que, quando visualizadas diretamente ou combinadas com outros dados relacionados, podem ser usadas para inferir razoavelmente a identidade de um indivíduo. Exemplos de PII incluem nomes, endereços e informações de contato.

PII

Veja as informações de identificação pessoal.

manual

Um conjunto de etapas predefinidas que capturam o trabalho associado às migrações, como a entrega das principais funções operacionais na nuvem. Um manual pode assumir a forma de scripts, runbooks automatizados ou um resumo dos processos ou etapas necessários para operar seu ambiente modernizado.

PLC

Consulte controlador lógico programável.

AMEIXA

Veja o gerenciamento do ciclo de vida do produto.

política

Um objeto que pode definir permissões (consulte a política baseada em identidade), especificar as condições de acesso (consulte a política baseada em recursos) ou definir as permissões máximas para todas as contas em uma organização em AWS Organizations (consulte a política de controle de serviços).

persistência poliglota

Escolher de forma independente a tecnologia de armazenamento de dados de um microsserviço com base em padrões de acesso a dados e outros requisitos. Se seus microsserviços tiverem a mesma tecnologia de armazenamento de dados, eles poderão enfrentar desafios de implementação ou apresentar baixa performance. Os microsserviços serão implementados com mais facilidade e alcançarão performance e escalabilidade melhores se usarem o armazenamento de dados mais bem adaptado às suas necessidades. Para obter mais informações, consulte Habilitar a persistência de dados em microsserviços.

avaliação do portfólio

Um processo de descobrir, analisar e priorizar o portfólio de aplicações para planejar a migração. Para obter mais informações, consulte Avaliar a preparação para a migração.

predicado

Uma condição de consulta que retorna `true` ou`false`, normalmente localizada em uma `WHERE` cláusula.

pressão de predicados

Uma técnica de otimização de consulta de banco de dados que filtra os dados na consulta antes da transferência. Isso reduz a quantidade de dados que devem ser recuperados e processados do banco de dados relacional e melhora o desempenho das consultas.

controle preventivo

Um controle de segurança projetado para evitar que um evento ocorra. Esses controles são a primeira linha de defesa para ajudar a evitar acesso não autorizado ou alterações indesejadas em sua rede. Para obter mais informações, consulte Controles preventivos em Como implementar controles de segurança na AWS.

principal (entidade principal)

Uma entidade AWS que pode realizar ações e acessar recursos. Essa entidade geralmente é um usuário raiz para um Conta da AWS, uma função do IAM ou um usuário. Para obter mais informações, consulte Entidade principal em Termos e conceitos de perfis na documentação do IAM.

privacidade por design

Uma abordagem de engenharia de sistema que leva em consideração a privacidade em todo o processo de desenvolvimento.

zonas hospedadas privadas

Um contêiner que contém informações sobre como você deseja que o Amazon Route 53 responda às consultas de DNS para um domínio e seus subdomínios em um ou mais. VPCs Para obter mais informações, consulte Como trabalhar com zonas hospedadas privadas na documentação do Route 53.

controle proativo

Um controle de segurança projetado para impedir a implantação de recursos não compatíveis. Esses controles examinam os recursos antes de serem provisionados. Se o recurso não estiver em conformidade com o controle, ele não será provisionado. Para obter mais informações, consulte o guia de referência de controles na AWS Control Tower documentação e consulte Controles proativos em Implementação de controles de segurança em AWS.

gerenciamento do ciclo de vida do produto (PLM)

O gerenciamento de dados e processos de um produto em todo o seu ciclo de vida, desde o design, desenvolvimento e lançamento, passando pelo crescimento e maturidade, até o declínio e a remoção.

ambiente de produção

Veja o ambiente.

controlador lógico programável (PLC)

Na fabricação, um computador altamente confiável e adaptável que monitora as máquinas e automatiza os processos de fabricação.

encadeamento imediato

Usando a saída de um prompt do LLM como entrada para o próximo prompt para gerar respostas melhores. Essa técnica é usada para dividir uma tarefa complexa em subtarefas ou para refinar ou expandir iterativamente uma resposta preliminar. Isso ajuda a melhorar a precisão e a relevância das respostas de um modelo e permite resultados mais granulares e personalizados.

pseudonimização

O processo de substituir identificadores pessoais em um conjunto de dados por valores de espaço reservado. A pseudonimização pode ajudar a proteger a privacidade pessoal. Os dados pseudonimizados ainda são considerados dados pessoais.

publish/subscribe (pub/sub)

Um padrão que permite comunicações assíncronas entre microsserviços para melhorar a escalabilidade e a capacidade de resposta. Por exemplo, em um MES baseado em microsserviços, um microsserviço pode publicar mensagens de eventos em um canal no qual outros microsserviços possam se inscrever. O sistema pode adicionar novos microsserviços sem alterar o serviço de publicação.

# Q

plano de consulta

Uma série de etapas, como instruções, usadas para acessar os dados em um sistema de banco de dados relacional SQL.

regressão de planos de consultas

Quando um otimizador de serviço de banco de dados escolhe um plano menos adequado do que escolhia antes de uma determinada alteração no ambiente de banco de dados ocorrer. Isso pode ser causado por alterações em estatísticas, restrições, configurações do ambiente, associações de parâmetros de consulta e atualizações do mecanismo de banco de dados.

# R

Matriz RACI

Veja responsável, responsável, consultado, informado (RACI).

RAG

Consulte Geração Aumentada de Recuperação.

ransomware

Um software mal-intencionado desenvolvido para bloquear o acesso a um sistema ou dados de computador até que um pagamento seja feito.

Matriz RASCI

Veja responsável, responsável, consultado, informado (RACI).

RCAC

Veja o controle de acesso por linha e coluna.

réplica de leitura

Uma cópia de um banco de dados usada somente para leitura. É possível encaminhar consultas para a réplica de leitura e reduzir a carga no banco de dados principal.

rearquiteta

Veja 7 Rs.

objetivo de ponto de recuperação (RPO).

O máximo período de tempo aceitável desde o último ponto de recuperação de dados. Isso determina o que é considerado uma perda aceitável de dados entre o último ponto de recuperação e a interrupção do serviço.

objetivo de tempo de recuperação (RTO)

O máximo atraso aceitável entre a interrupção e a restauração do serviço.

refatorar

Veja 7 Rs.

Região

Uma coleção de AWS recursos em uma área geográfica. Cada um Região da AWS é isolado e independente dos outros para fornecer tolerância a falhas, estabilidade e resiliência. Para obter mais informações, consulte Especificar o que Regiões da AWS sua conta pode usar.

regressão

Uma técnica de ML que prevê um valor numérico. Por exemplo, para resolver o problema de “Por qual preço esta casa será vendida?” um modelo de ML pode usar um modelo de regressão linear para prever o preço de venda de uma casa com base em fatos conhecidos sobre a casa (por exemplo, a metragem quadrada).

redefinir a hospedagem

Veja 7 Rs.

versão

Em um processo de implantação, o ato de promover mudanças em um ambiente de produção.

realocar

Veja 7 Rs.

redefinir a plataforma

Veja 7 Rs.

recomprar

Veja 7 Rs.

resiliência

A capacidade de um aplicativo de resistir ou se recuperar de interrupções. Alta disponibilidade e recuperação de desastres são considerações comuns ao planejar a resiliência no. Nuvem AWS Para obter mais informações, consulte Nuvem AWS Resiliência.

política baseada em recurso

Uma política associada a um recurso, como um bucket do Amazon S3, um endpoint ou uma chave de criptografia. Esse tipo de política especifica quais entidades principais têm acesso permitido, ações válidas e quaisquer outras condições que devem ser atendidas.

matriz responsável, accountable, consultada, informada (RACI)

Uma matriz que define as funções e responsabilidades de todas as partes envolvidas nas atividades de migração e nas operações de nuvem. O nome da matriz é derivado dos tipos de responsabilidade definidos na matriz: responsável (R), responsabilizável (A), consultado (C) e informado (I). O tipo de suporte (S) é opcional. Se você incluir suporte, a matriz será chamada de matriz RASCI e, se excluir, será chamada de matriz RACI.

controle responsivo

Um controle de segurança desenvolvido para conduzir a remediação de eventos adversos ou desvios em relação à linha de base de segurança. Para obter mais informações, consulte Controles responsivos em Como implementar controles de segurança na AWS.

reter

Veja 7 Rs.

aposentar-se

Veja 7 Rs.

Geração Aumentada de Recuperação (RAG)

Uma tecnologia de IA generativa na qual um LLM faz referência a uma fonte de dados autorizada que está fora de suas fontes de dados de treinamento antes de gerar uma resposta. Por exemplo, um modelo RAG pode realizar uma pesquisa semântica na base de conhecimento ou nos dados personalizados de uma organização. Para obter mais informações, consulte O que é RAG.

alternância

O processo de atualizar periodicamente um segredo para dificultar o acesso das credenciais por um invasor.

controle de acesso por linha e coluna (RCAC)

O uso de expressões SQL básicas e flexíveis que tenham regras de acesso definidas. O RCAC consiste em permissões de linha e máscaras de coluna.

RPO

Veja o objetivo do ponto de recuperação.

RTO

Veja o objetivo do tempo de recuperação.

runbook

Um conjunto de procedimentos manuais ou automatizados necessários para realizar uma tarefa específica. Eles são normalmente criados para agilizar operações ou procedimentos repetitivos com altas taxas de erro.

# S

SAML 2.0

Um padrão aberto que muitos provedores de identidade (IdPs) usam. Esse recurso permite o login único federado (SSO), para que os usuários possam fazer login AWS Management Console ou chamar as operações da AWS API sem que você precise criar um usuário no IAM para todos em sua organização. Para obter mais informações sobre a federação baseada em SAML 2.0, consulte Sobre a federação baseada em SAML 2.0 na documentação do IAM.

SCADA

Veja controle de supervisão e aquisição de dados.

SCP

Veja a política de controle de serviços.

secret

Em AWS Secrets Manager, informações confidenciais ou restritas, como uma senha ou credenciais de usuário, que você armazena de forma criptografada. Ele consiste no valor secreto e em seus metadados. O valor secreto pode ser binário, uma única string ou várias strings. Para obter mais informações, consulte O que há em um segredo do Secrets Manager? na documentação do Secrets Manager.

segurança por design

Uma abordagem de engenharia de sistema que leva em consideração a segurança em todo o processo de desenvolvimento.

controle de segurança

Uma barreira de proteção técnica ou administrativa que impede, detecta ou reduz a capacidade de uma ameaça explorar uma vulnerabilidade de segurança. Existem quatro tipos principais de controles de segurança: preventivos, detectivos, responsivos e proativos.

fortalecimento da segurança

O processo de reduzir a superfície de ataque para torná-la mais resistente a ataques. Isso pode incluir ações como remover recursos que não são mais necessários, implementar a prática recomendada de segurança de conceder privilégios mínimos ou desativar recursos desnecessários em arquivos de configuração.

sistema de gerenciamento de eventos e informações de segurança (SIEM)

Ferramentas e serviços que combinam sistemas de gerenciamento de informações de segurança (SIM) e gerenciamento de eventos de segurança (SEM). Um sistema SIEM coleta, monitora e analisa dados de servidores, redes, dispositivos e outras fontes para detectar ameaças e violações de segurança e gerar alertas.

automação de resposta de segurança

Uma ação predefinida e programada projetada para responder ou remediar automaticamente um evento de segurança. Essas automações servem como controles de segurança responsivos ou detectivos que ajudam você a implementar as melhores práticas AWS de segurança. Exemplos de ações de resposta automatizada incluem a modificação de um grupo de segurança da VPC, a correção de uma instância EC2 da Amazon ou a rotação de credenciais.

Criptografia do lado do servidor

Criptografia dos dados em seu destino, por AWS service (Serviço da AWS) quem os recebe.

política de controle de serviços (SCP)

Uma política que fornece controle centralizado sobre as permissões de todas as contas em uma organização em AWS Organizations. SCPs defina barreiras ou estabeleça limites nas ações que um administrador pode delegar a usuários ou funções. Você pode usar SCPs como listas de permissão ou listas de negação para especificar quais serviços ou ações são permitidos ou proibidos. Para obter mais informações, consulte Políticas de controle de serviço na AWS Organizations documentação.

service endpoint (endpoint de serviço)

O URL do ponto de entrada para um AWS service (Serviço da AWS). Você pode usar o endpoint para se conectar programaticamente ao serviço de destino. Para obter mais informações, consulte Endpoints do AWS service (Serviço da AWS) na Referência geral da AWS.

acordo de serviço (SLA)

Um acordo que esclarece o que uma equipe de TI promete fornecer aos clientes, como tempo de atividade e performance do serviço.

indicador de nível de serviço (SLI)

Uma medida de um aspecto de desempenho de um serviço, como taxa de erro, disponibilidade ou taxa de transferência.

objetivo de nível de serviço (SLO)

Uma métrica alvo que representa a integridade de um serviço, conforme medida por um indicador de nível de serviço.

modelo de responsabilidade compartilhada

Um modelo que descreve a responsabilidade com a qual você compartilha AWS pela segurança e conformidade na nuvem. AWS é responsável pela segurança da nuvem, enquanto você é responsável pela segurança na nuvem. Para obter mais informações, consulte o Modelo de responsabilidade compartilhada.

SIEM

Veja informações de segurança e sistema de gerenciamento de eventos.

ponto único de falha (SPOF)

Uma falha em um único componente crítico de um aplicativo que pode interromper o sistema.

SLA

Veja o contrato de nível de serviço.

ESGUIO

Veja o indicador de nível de serviço.

SLO

Veja o objetivo do nível de serviço.

split-and-seed modelo

Um padrão para escalar e acelerar projetos de modernização. À medida que novos recursos e lançamentos de produtos são definidos, a equipe principal se divide para criar novas equipes de produtos. Isso ajuda a escalar os recursos e os serviços da sua organização, melhora a produtividade do desenvolvedor e possibilita inovações rápidas. Para obter mais informações, consulte Abordagem em fases para modernizar aplicativos no. Nuvem AWS

CUSPE

Veja um único ponto de falha.

esquema de estrelas

Uma estrutura organizacional de banco de dados que usa uma grande tabela de fatos para armazenar dados transacionais ou medidos e usa uma ou mais tabelas dimensionais menores

para armazenar atributos de dados. Essa estrutura foi projetada para uso em um data warehouse ou para fins de inteligência comercial.

padrão strangler fig

Uma abordagem à modernização de sistemas monolíticos que consiste em reescrever e substituir incrementalmente a funcionalidade do sistema até que o sistema herdado possa ser desativado. Esse padrão usa a analogia de uma videira que cresce e se torna uma árvore estabelecida e, eventualmente, supera e substitui sua hospedeira. O padrão foi apresentado por Martin Fowler como forma de gerenciar riscos ao reescrever sistemas monolíticos. Para ver um exemplo de como aplicar esse padrão, consulte Modernizar incrementalmente os serviços Web herdados do Microsoft ASP.NET (ASMX) usando contêineres e o Amazon API Gateway.

sub-rede

Um intervalo de endereços IP na VPC. Cada sub-rede fica alocada em uma única zona de disponibilidade.

controle de supervisão e aquisição de dados (SCADA)

Na manufatura, um sistema que usa hardware e software para monitorar ativos físicos e operações de produção.

symmetric encryption (criptografia simétrica)

Um algoritmo de criptografia que usa a mesma chave para criptografar e descriptografar dados.

testes sintéticos

Testar um sistema de forma que simule as interações do usuário para detectar possíveis problemas ou monitorar o desempenho. Você pode usar o Amazon CloudWatch Synthetics para criar esses testes.

prompt do sistema

Uma técnica para fornecer contexto, instruções ou diretrizes a um LLM para direcionar seu comportamento. Os prompts do sistema ajudam a definir o contexto e estabelecer regras para interações com os usuários.

# T

tags

Pares de valores-chave que atuam como metadados para organizar seus recursos. AWS As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos. Para obter mais informações, consulte Marcar seus recursos do AWS.

variável-alvo

O valor que você está tentando prever no ML supervisionado. Ela também é conhecida como variável de resultado. Por exemplo, em uma configuração de fabricação, a variável-alvo pode ser um defeito do produto.

lista de tarefas

Uma ferramenta usada para monitorar o progresso por meio de um runbook. Uma lista de tarefas contém uma visão geral do runbook e uma lista de tarefas gerais a serem concluídas. Para cada tarefa geral, ela inclui o tempo estimado necessário, o proprietário e o progresso.

ambiente de teste

Veja o ambiente.

treinamento

O processo de fornecer dados para que seu modelo de ML aprenda. Os dados de treinamento devem conter a resposta correta. O algoritmo de aprendizado descobre padrões nos dados de treinamento que mapeiam os atributos dos dados de entrada no destino (a resposta que você deseja prever). Ele gera um modelo de ML que captura esses padrões. Você pode usar o modelo de ML para obter previsões de novos dados cujo destino você não conhece.

gateway de trânsito

Um hub de trânsito de rede que você pode usar para interconectar sua rede com VPCs a rede local. Para obter mais informações, consulte O que é um gateway de trânsito na AWS Transit Gateway documentação.

fluxo de trabalho baseado em troncos

Uma abordagem na qual os desenvolvedores criam e testam recursos localmente em uma ramificação de recursos e, em seguida, mesclam essas alterações na ramificação principal. A

ramificação principal é então criada para os ambientes de desenvolvimento, pré-produção e produção, sequencialmente.

Acesso confiável

Conceder permissões a um serviço que você especifica para realizar tarefas em sua organização AWS Organizations e em suas contas em seu nome. O serviço confiável cria um perfil vinculado ao serviço em cada conta, quando esse perfil é necessário, para realizar tarefas de gerenciamento para você. Para obter mais informações, consulte Usando AWS Organizations com outros AWS serviços na AWS Organizations documentação.

tuning (ajustar)

Alterar aspectos do processo de treinamento para melhorar a precisão do modelo de ML. Por exemplo, você pode treinar o modelo de ML gerando um conjunto de rótulos, adicionando rótulos e repetindo essas etapas várias vezes em configurações diferentes para otimizar o modelo.

equipe de duas pizzas

Uma pequena DevOps equipe que você pode alimentar com duas pizzas. Uma equipe de duas pizzas garante a melhor oportunidade possível de colaboração no desenvolvimento de software.

# U

incerteza

Um conceito que se refere a informações imprecisas, incompletas ou desconhecidas que podem minar a confiabilidade dos modelos preditivos de ML. Há dois tipos de incertezas: a incerteza epistêmica é causada por dados limitados e incompletos, enquanto a incerteza aleatória é causada pelo ruído e pela aleatoriedade inerentes aos dados. Para obter mais informações, consulte o guia Como quantificar a incerteza em sistemas de aprendizado profundo.

tarefas indiferenciadas

Também conhecido como trabalho pesado, trabalho necessário para criar e operar um aplicativo, mas que não fornece valor direto ao usuário final nem oferece vantagem competitiva. Exemplos de tarefas indiferenciadas incluem aquisição, manutenção e planejamento de capacidade.

ambientes superiores

Veja o ambiente.

# V

aspiração

Uma operação de manutenção de banco de dados que envolve limpeza após atualizações incrementais para recuperar armazenamento e melhorar a performance.

controle de versões

Processos e ferramentas que rastreiam mudanças, como alterações no código-fonte em um repositório.

emparelhamento da VPC

Uma conexão entre duas VPCs que permite rotear o tráfego usando endereços IP privados. Para ter mais informações, consulte O que é emparelhamento de VPC? na documentação da Amazon VPC.

Vulnerabilidade

Uma falha de software ou hardware que compromete a segurança do sistema.

# W

cache quente

Um cache de buffer que contém dados atuais e relevantes que são acessados com frequência. A instância do banco de dados pode ler do cache do buffer, o que é mais rápido do que ler da memória principal ou do disco.

dados mornos

Dados acessados raramente. Ao consultar esse tipo de dados, consultas moderadamente lentas geralmente são aceitáveis.

função de janela

Uma função SQL que executa um cálculo em um grupo de linhas que se relacionam de alguma forma com o registro atual. As funções de janela são úteis para processar tarefas, como calcular uma média móvel ou acessar o valor das linhas com base na posição relativa da linha atual.

workload

Uma coleção de códigos e recursos que geram valor empresarial, como uma aplicação voltada para o cliente ou um processo de back-end.

workstreams

Grupos funcionais em um projeto de migração que são responsáveis por um conjunto específico de tarefas. Cada workstream é independente, mas oferece suporte aos outros workstreams do projeto. Por exemplo, o workstream de portfólio é responsável por priorizar aplicações, planejar ondas e coletar metadados de migração. O workstream de portfólio entrega esses ativos ao workstream de migração, que então migra os servidores e as aplicações.

MINHOCA

Veja escrever uma vez, ler muitas.

WQF

Consulte Estrutura de qualificação AWS da carga de trabalho.

escreva uma vez, leia muitas (WORM)

Um modelo de armazenamento que grava dados uma única vez e evita que os dados sejam excluídos ou modificados. Os usuários autorizados podem ler os dados quantas vezes forem necessárias, mas não podem alterá-los. Essa infraestrutura de armazenamento de dados é considerada imutável.

# Z

exploração de dia zero

Um ataque, geralmente malware, que tira proveito de uma vulnerabilidade de dia zero.

vulnerabilidade de dia zero

Uma falha ou vulnerabilidade não mitigada em um sistema de produção. Os agentes de ameaças podem usar esse tipo de vulnerabilidade para atacar o sistema. Os desenvolvedores frequentemente ficam cientes da vulnerabilidade como resultado do ataque.

aviso zero-shot

Fornecer a um LLM instruções para realizar uma tarefa, mas sem exemplos (fotos) que possam ajudar a orientá-la. O LLM deve usar seu conhecimento pré-treinado para lidar com a tarefa. A

eficácia da solicitação zero depende da complexidade da tarefa e da qualidade da solicitação. Veja também a solicitação de algumas fotos.

aplicação zumbi

Uma aplicação que tem um uso médio de CPU e memória inferior a 5%. Em um projeto de migração, é comum retirar essas aplicações.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.