Bonnes pratiques avec Amazon Q Developer pour la génération de code en ligne et par assistant

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Bonnes pratiques avec Amazon Q Developer pour la génération de code en ligne et par assistant

Table of Contents

Bonnes pratiques avec Amazon Q Developer pour la génération de code en ligne et par assistant

Amazon Web Services (Collaborateurs)

août 2024 (1Historique du document)

Traditionnellement, les développeurs s'appuyaient sur leur propre expertise, leur propre documentation et des extraits de code provenant de diverses sources pour écrire et gérer le code. Bien que ces méthodes aient bien servi le secteur, elles peuvent être chronophages et sujettes à des erreurs humaines, ce qui entraîne des inefficiences et des bogues potentiels.

C'est là qu'Amazon Q Developer intervient pour améliorer le parcours du développeur. Amazon Q Developer est un puissant assistant AWS génératif alimenté par l'IA conçu pour accélérer les tâches de développement de code en fournissant une génération de code intelligente et des recommandations.

Cependant, comme pour toute nouvelle technologie, elle peut présenter des défis. Les attentes irréalistes, les difficultés d'intégration, le dépannage des générations de code inexactes et l'utilisation appropriée des fonctionnalités d'Amazon Q sont des obstacles courants auxquels les développeurs peuvent être confrontés. Ce guide complet répond à ces défis en proposant des scénarios réels, des meilleures pratiques détaillées, des solutions de dépannage et des exemples de code pratiques et concrets spécifiquement pour Python and Java, deux des langages de programmation les plus répandus.

Ce guide se concentre sur l'utilisation d'Amazon Q Developer pour effectuer des tâches de développement de code telles que :

- Achèvement du code : générez des suggestions en ligne pendant que les développeurs codent en temps réel.
- Améliorations du code et conseils — Discutez du développement de logiciels, générez du nouveau code en langage naturel et améliorez le code existant.

Objectifs

L'objectif de ce guide est de soutenir les développeurs qui sont des utilisateurs nouveaux ou réguliers d'Amazon Q Developer, en les aidant à utiliser le service avec succès dans leurs tâches de codage

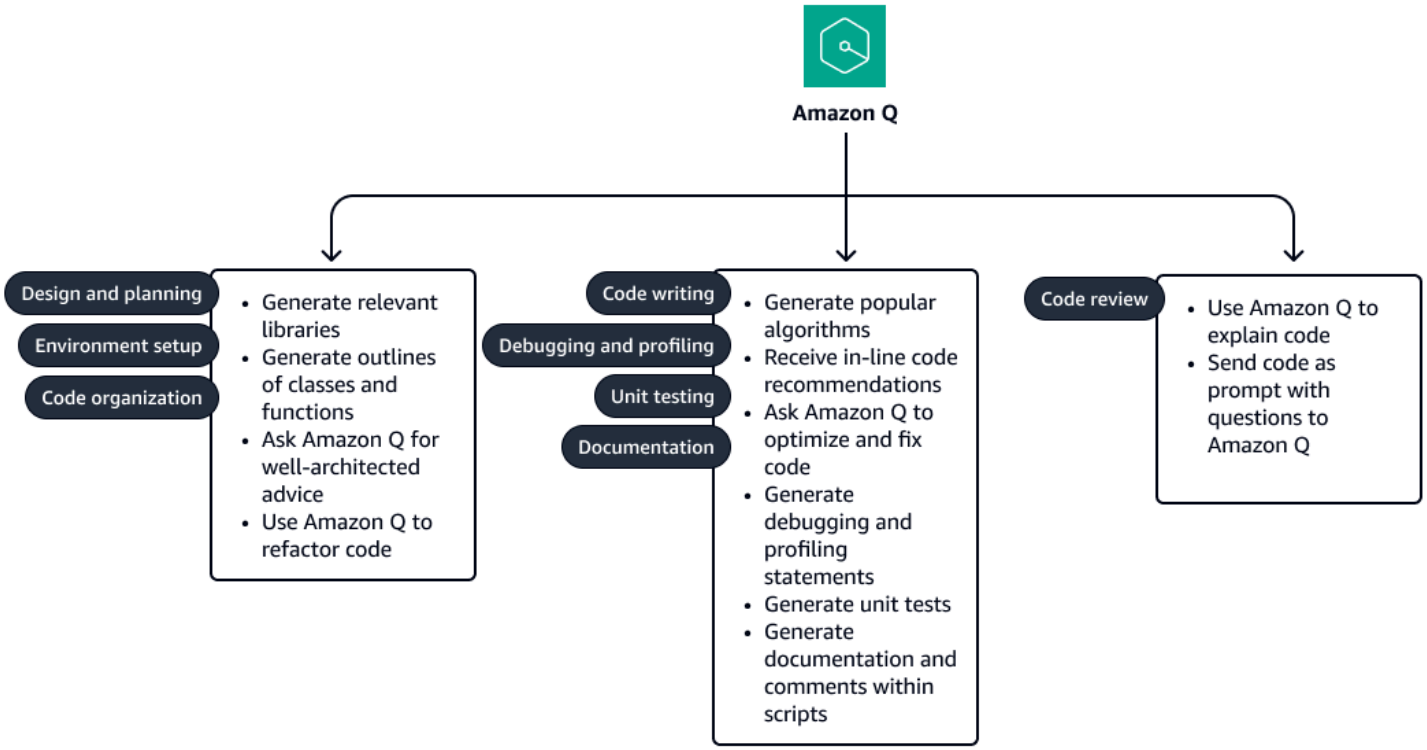
quotidiennes. Les chefs d'équipe de développement peuvent également tirer parti de la lecture de ce guide.

Ce guide fournit les informations suivantes sur l'utilisation d'Amazon Q Developer :

- Découvrez l'utilisation efficace d'Amazon Q Developer pour le développement de code
  - Donnez les meilleures pratiques pour intégrer Amazon Q Developer dans le flux de travail d'un développeur.
  - Offrez step-by-step des conseils avec des exemples pour une génération de code réussie et des recommandations.
- Réduisez les défis courants et encouragez les développeurs à clarifier l'utilisation d'Amazon Q Developer
  - Proposez des stratégies et des informations pour répondre aux attentes des développeurs et surmonter les obstacles liés à la précision et aux performances de la génération de code.
- Fournir des services de dépannage et de gestion des erreurs
  - Fournissez aux développeurs des conseils de résolution des problèmes liés à la génération de code Amazon Q Developer afin de remédier aux résultats inexacts ou aux comportements inattendus.
  - Fournissez des exemples concrets et des scénarios spécifiques à Python and Java.
- Optimisez les flux de travail et la productivité
  - Optimisez les flux de travail de développement de code avec Amazon Q Developer.
  - Discutez des stratégies visant à améliorer la productivité des développeurs.

Utilisation d'Amazon Q Developer dans les flux de travail des développeurs

Les développeurs suivent un flux de travail standard comprenant les étapes de collecte des exigences, de conception et de planification, de codage, de test, de révision du code et de déploiement. Cette section explique comment utiliser les fonctionnalités d'Amazon Q Developer pour optimiser les principales étapes de développement.



Le schéma précédent montre comment Amazon Q Developer peut accélérer et rationaliser les tâches courantes suivantes au cours des étapes de développement du code :

- Conception et planification | Configuration de l'environnement | Organisation du code
  - Génération de bibliothèques pertinentes
  - Génération de plans de classes et de fonctions
  - Demandez à Amazon Q des conseils bien conçus
  - Utilisation d'Amazon Q pour refactoriser du code
- Écriture de code | Débogage et profilage | Tests unitaires | Documentation
  - Générez des algorithmes populaires

  - Recevez des recommandations de code en ligne
  - Demandez à Amazon Q d'optimiser et de corriger le code
  - Génération d'instructions de débogage et de profilage
  - Générer des tests unitaires
  - Générez de la documentation et des commentaires dans les scripts
- Vérification de code
  - Demandez à Amazon Q de vous expliquer le code
  - Envoyez le code avec des questions à Amazon Q

Conception et planification

Après avoir recensé les exigences commerciales et techniques, les développeurs conçoivent de nouvelles bases de code ou étendent les bases de code existantes. Au cours de cette phase, Amazon Q Developer peut aider les développeurs à effectuer les tâches suivantes :

- Générez des bibliothèques pertinentes et des schémas de classes et de fonctions pour des conseils bien conçus.
- Fournir des conseils pour les questions d'ingénierie, de compatibilité et de conception architecturale.

Codage

Le processus de codage utilise Amazon Q Developer pour accélérer le développement de la manière suivante :

- Configuration de l'environnement : installez le AWS Toolkit dans votre environnement de développement intégré (IDE) (par exemple, VS Code ou IntelliJ). Utilisez ensuite Amazon Q pour générer des bibliothèques ou recevoir des suggestions de configuration en fonction des objectifs de votre projet. Pour plus de détails, consultez les meilleures pratiques pour l'intégration d'Amazon Q Developer.
- Organisation du code : refactorisez le code ou obtenez des recommandations d'organisation auprès d'Amazon Q qui correspondent aux objectifs de votre projet.
- Rédaction de code - Utilisez des suggestions en ligne pour générer du code lors du développement ou demandez à Amazon Q de générer du code en utilisant le panneau de discussion Amazon Q

dans votreIDE. Pour plus de détails, consultez les meilleures pratiques en matière de génération de code avec Amazon Q Developer.

- Débogage et profilage : générez des commandes de profilage ou utilisez les options Amazon Q telles que Fix et Explain pour corriger les problèmes.
- Tests unitaires : fournissez le code sous forme d'invite à Amazon Q lors d'une session de chat et demandez la génération de tests unitaires applicables. Pour plus d'informations, consultez Exemples de code avec Amazon Q Developer.
- Documentation - Utilisez des suggestions en ligne pour créer des commentaires et des docstrings, ou utilisez l'option Explain pour générer des résumés détaillés des sélections de code. Pour plus d'informations, consultez Exemples de code avec Amazon Q Developer.

Vérification de code

Les réviseurs doivent comprendre le code de développement avant de le mettre en production. Pour accélérer ce processus, utilisez les options Amazon Q Explain et Optimize, ou envoyez des sélections de code avec des instructions personnalisées à Amazon Q lors d'une session de chat. Pour plus d'informations, consultez la section Exemples de chat.

Intégration et déploiement

Demandez à Amazon Q des conseils sur l'intégration continue, les pipelines de livraison et les meilleures pratiques de déploiement spécifiques à l'architecture de votre projet.

À l'aide de ces recommandations, vous pouvez apprendre à exploiter efficacement les fonctionnalités d'Amazon Q Developer, à optimiser vos flux de travail et à augmenter la productivité tout au long du cycle de développement.

Fonctionnalités avancées d'Amazon Q Developer

Bien que ce guide se concentre sur l'utilisation d'Amazon Q Developer dans le cadre de tâches de programmation pratiques, il est important de connaître les fonctionnalités avancées suivantes :

- Transformation du code Amazon Q Developer
- Personnalisations d'Amazon Q Developer

Transformation du code Amazon Q Developer

L'agent de développement Amazon Q pour la transformation du code peut mettre à niveau la version en langage de code de vos fichiers sans que vous ayez à réécrire le code manuellement. Il fonctionne en analysant vos fichiers de code existants et en les réécrivant automatiquement pour utiliser une version plus récente du langage. Par exemple, Amazon Q transforme un seul module si vous travaillez dans un IDE tel que Eclipse. Si vous utilisez Visual Studio Code, Amazon Q peut transformer l'intégralité d'un projet ou d'un espace de travail.

Utilisez Amazon Q lorsque vous souhaitez effectuer des tâches courantes de mise à niveau du code, telles que les suivantes :

- Mettez à jour le code pour qu'il fonctionne avec la nouvelle syntaxe de la version linguistique.
- Exécutez des tests unitaires pour valider la réussite de la compilation et de l'exécution.
- Vérifiez et résolvez les problèmes de déploiement.

Amazon Q permet aux développeurs d'économiser des jours, voire des mois, de travail fastidieux et répétitif pour mettre à niveau les bases de code.

Depuis juin 2024, Amazon Q Developer prend en charge la mise à niveau Java code et peut transformer Java Code 8 vers des versions plus récentes telles que Java 11 ou 17.

Personnalisations d'Amazon Q Developer

Grâce à ses fonctionnalités de personnalisation, Amazon Q Developer peut fournir des suggestions en ligne basées sur la propre base de code de l'entreprise. La société fournit son référentiel de code soit à Amazon Simple Storage Service (Amazon S3), soit par le AWS CodeConnections biais

de Connections, anciennement connu sous le nom de AWS CodeStar Connections. Amazon Q utilise ensuite le référentiel de code personnalisé doté d'une sécurité activée pour recommander des modèles de codage pertinents pour les développeurs de cette organisation.

Lorsque vous utilisez les personnalisations d'Amazon Q Developer, tenez compte des points suivants :

- Depuis juin 2024, la fonctionnalité Amazon Q Developer Customizations est en mode aperçu. Par conséquent, la disponibilité et le support de cette fonctionnalité peuvent être limités.
- Les suggestions de code personnalisées en ligne ne seront exactes que compte tenu de la qualité des référentiels de code fournis. Nous vous recommandons de consulter un score d'évaluation pour chaque personnalisation que vous créez.
- Pour optimiser les performances, nous vous recommandons d'inclure au moins 20 fichiers de données contenant la langue donnée, tous les fichiers source ayant une taille supérieure à 10 Mo. Assurez-vous que votre référentiel est composé de code source référençable et non de fichiers de métadonnées (par exemple, des fichiers de configuration, des fichiers de propriétés et des fichiers readme).

En utilisant les personnalisations d'Amazon Q Developer, vous pouvez gagner du temps de différentes manières :

- Utilisez des recommandations basées sur le code propriétaire de votre entreprise.
- Améliorez la réutilisabilité des bases de code existantes.
- Créez des modèles reproductibles qui sont généralisés dans l'ensemble de votre entreprise.

Meilleures pratiques de codage avec Amazon Q Developer

Cette section décrit les meilleures pratiques en matière de codage avec Amazon Q Developer. Les meilleures pratiques incluent les catégories suivantes :

- Intégration - Méthodes et considérations relatives à l'intégration
- Génération de code - Conseils pour utiliser correctement la génération de code
- Recommandations relatives au code - Techniques pour améliorer le code

Bonnes pratiques pour l'intégration d'Amazon Q Developer

Amazon Q Developer est un puissant assistant de codage par IA génératif disponible via des logiciels populaires IDEs tels que Visual Studio Code et JetBrains. Cette section se concentre sur les meilleures pratiques pour accéder à Amazon Q Developer et l'intégrer à votre environnement de développement de codage.

Conditions requises pour Amazon Q Developer

Amazon Q Developer est disponible dans le cadre de AWS Toolkit for Visual Studio Code et AWS Toolkit for JetBrains (par exemple, IntelliJ et PyCharm). Pour Visual Studio Code et JetBrains IDEs Amazon Q Developer prend en charge Python, Java JavaScript, C# TypeScript, Go, Rust, PHP, Ruby, Kotlin, C, C++, les scripts Shell, SQL et Scala.

Pour obtenir des instructions détaillées sur l'installation du code AWS Toolkit pour Visual Studio et d'un JetBrains IDE, consultez la section Installation de l'extension ou du plug-in Amazon Q Developer dans votre IDE dans le manuel Amazon Q Developer User Guide.

Bonnes pratiques lors de l'utilisation d'Amazon Q Developer

Les meilleures pratiques générales relatives à l'utilisation d'Amazon Q Developer sont les suivantes :

- Fournissez un contexte pertinent pour obtenir des réponses plus précises, comme les langages de programmation, les frameworks et les outils utilisés. Décomposez les problèmes complexes en composants plus petits.
- Faites des essais et répétez en fonction de vos demandes et questions. La programmation implique souvent d'essayer différentes approches.

- Passez toujours en revue les suggestions de code avant de les accepter, et modifiez-les au besoin pour vous assurer qu'elles correspondent exactement à vos attentes.
- Utilisez la fonctionnalité de personnalisation pour informer Amazon Q Developer de vos bibliothèques internes APIs, de vos meilleures pratiques et de vos modèles architecturaux afin d'obtenir des recommandations plus pertinentes.

Confidentialité des données et utilisation du contenu dans Amazon Q Developer

Lorsque vous décidez d'utiliser Amazon Q Developer, vous devez comprendre comment vos données et votre contenu sont utilisés. Les points essentiels sont les suivants :

- Pour les utilisateurs d'Amazon Q Developer Pro, le contenu de votre code n'est pas utilisé pour améliorer le service ou former des modèles.
- Les utilisateurs d'Amazon Q Developer Free Tier peuvent refuser que leur contenu soit utilisé pour améliorer le service via les paramètres ou les AWS Organizations politiques de l'IDE.
- Le contenu transmis est crypté et tout contenu stocké est sécurisé grâce au chiffrement au repos et à des contrôles d'accès. Pour plus d'informations, consultez la section Chiffrement des données dans Amazon Q Developer dans le manuel Amazon Q Developer User Guide.

Bonnes pratiques pour la génération de code avec Amazon Q Developer

Amazon Q Developer fournit des fonctionnalités de génération automatique de code, de saisie automatique et de suggestions de code en langage naturel. Voici les meilleures pratiques pour utiliser l'assistance de codage en ligne d'Amazon Q Developer :

- Fournir un contexte pour améliorer la précision des réponses

  Commencez par le code existant, importez des bibliothèques, créez des classes et des fonctions ou établissez des squelettes de code. Ce contexte permettra d'améliorer de manière significative la qualité de génération du code.
- Codez naturellement

Utilisez la génération de code Amazon Q Developer comme un puissant moteur d'auto-complétion. Codez comme vous le faites habituellement et laissez Amazon Q vous proposer des suggestions au fur et à mesure que vous tapez ou que vous faites une pause. Si la génération de code n'est pas disponible ou si vous êtes bloqué par un problème de code, lancez Amazon Q en tapant Alt+C sur un PC ou Option+C sur macOS. Pour plus d'informations sur les actions courantes que vous pouvez effectuer lors de l'utilisation des suggestions en ligne, consultez la section Utilisation des touches de raccourci dans le manuel Amazon Q Developer User Guide.

- Incluez des bibliothèques d'importation adaptées aux objectifs de votre script

  Incluez les bibliothèques d'importation pertinentes pour aider Amazon Q à comprendre le contexte et à générer du code en conséquence. Vous pouvez également demander à Amazon Q de suggérer des déclarations d'importation pertinentes.

- Maintenir un contexte clair et ciblé

  Concentrez votre script sur des objectifs spécifiques et modularisez les fonctionnalités distinctes dans des scripts distincts avec un contexte pertinent. Évitez tout contexte bruyant ou confus.

- Expérimentez avec des instructions

  Découvrez différentes instructions pour inciter Amazon Q à produire des résultats utiles lors de la génération de code. Essayez par exemple les approches suivantes :

  - Utilisez des blocs de commentaires standard pour les instructions en langage naturel.
  - Créez des squelettes avec des commentaires pour renseigner les classes et les fonctions.
  - Soyez précis dans vos instructions, en fournissant des détails plutôt que des généralisations.

- Discutez avec le développeur Amazon Q et demandez de l'aide

  Si Amazon Q Developer ne fournit pas de suggestions précises, discutez avec Amazon Q Developer dans votre IDE. Il peut fournir des extraits de code ou des classes et fonctions complètes pour démarrer votre contexte. Pour plus d'informations, consultez la section Discuter avec un développeur Amazon Q à propos du code dans le guide de l'utilisateur Amazon Q Developer.

Bonnes pratiques pour les recommandations de code avec Amazon Q Developer

Amazon Q Developer peut répondre aux questions des développeurs et évaluer le code pour proposer des recommandations allant de la génération de code à la correction de bogues en passant par des conseils en langage naturel. Voici les bonnes pratiques relatives à l'utilisation du chat dans Amazon Q :

- Générer du code à partir de zéro

  Pour les nouveaux projets ou lorsque vous avez besoin d'une fonction générale (par exemple, copier des fichiers depuis Amazon S3), demandez à Amazon Q Developer de générer des exemples de code à l'aide d'instructions en langage naturel. Amazon Q peut fournir des liens connexes vers des ressources publiques à des fins de validation et d'investigation supplémentaires.
- Recherchez des connaissances en matière de codage et des explications sur les erreurs

  En cas de problèmes de codage ou de messages d'erreur, fournissez le bloc de code (avec le message d'erreur, le cas échéant) et votre question sous forme d'invite à Amazon Q Developer. Ce contexte aidera Amazon Q à fournir des réponses précises et pertinentes.
- Améliorez le code existant

  Pour corriger les erreurs connues ou optimiser le code (par exemple, pour réduire la complexité), sélectionnez le bloc de code approprié et envoyez-le à Amazon Q Developer avec votre demande. Soyez précis dans vos instructions pour de meilleurs résultats.
- Expliquer les fonctionnalités du code

  Lorsque vous explorez de nouveaux référentiels de code, sélectionnez un bloc de code ou un script complet et envoyez-le à Amazon Q Developer pour obtenir des explications. Réduisez la taille de la sélection pour des explications plus spécifiques.
- Générer des tests unitaires

  Après avoir envoyé un bloc de code en guise d'invite, demandez à Amazon Q Developer de générer des tests unitaires. Cette approche permet d'économiser du temps et des coûts de développement liés à la couverture du code et DevOps.
- Trouvez des AWS réponses

Amazon Q Developer est une ressource précieuse pour les développeurs qui travaillent avec elle, Services AWS car elle contient une grande quantité de connaissances liées à AWS. Que vous rencontriez des difficultés liées à un sujet spécifique Service AWS, que vous rencontriez des messages d' AWS erreur spécifiques ou que vous essayiez d'en apprendre davantage Service AWS, Amazon Q fournit souvent des informations pertinentes et utiles.

Consultez toujours les recommandations qu'Amazon Q Developer vous fournit. Ensuite, apportez les modifications nécessaires et exécutez des tests pour vous assurer que le code répond aux fonctionnalités que vous souhaitez.

Exemples de code avec Amazon Q Developer

Cette section fournit des exemples réalistes de la manière d'améliorer votre expérience et la génération de code en mettant l'accent sur Python and Java langues. Outre des exemples en ligne, certains scénarios montrent comment vous pouvez utiliser la fonctionnalité de chat d'Amazon Q Developer pour demander des recommandations de code pendant que vous programmez.

Pour plus d'informations, consultez les rubriques suivantes :

- Python exemples
- Java exemples
- Exemples de chat

Python exemples

Procédez comme suit : Python des exemples montrent comment vous pouvez commencer à générer des suggestions de code en ligne avec Amazon Q Developer. Ces exemples illustratifs vous montrent comment déclencher l'assistant de codage pour vos propres projets. La plupart des exemples suivants sont applicables à d'autres langues :

- Génération de classes et de fonctions
- Code du document
- Générer des algorithmes
- Générer des tests unitaires

Génération de classes et de fonctions

Pour générer une implémentation de classe complète ou partielle, utilisez des commentaires de code. Le commentaire décrit l'intention de la classe. Commencez simplement et ajoutez plus de détails si nécessaire. Après avoir généré une classe, vous pouvez continuer à générer des fonctions dans son squelette.

Lorsque vous rédigez des commentaires, il est préférable d'utiliser la structure de commentaires de la langue. Entrée Python, pour les commentaires d'une seule ligne, utilisez # .Pour les commentaires multilignes, utilisez ' ' ' ' ' ' ou" " " " " ".

Entrée :

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

Sortie :

```
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
```

Maintenant qu'il existe une classe initiale, une fonction supplémentaire peut être générée avec des commentaires supplémentaires.

Entrée :

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

Sortie :

```
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}")
```

Code du document

Réduisez le temps consacré à la documentation du code en vous appuyant sur la documentation en ligne. Addition Python pauses dans les commentaires`''' '''`.

Entrée :

```
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}")
```

Sortie :

```
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}")
```

Générer des algorithmes

Amazon Q Developer peut générer des algorithmes populaires, en particulier ceux liés à AWS SDKs. Dans votre script, ajoutez une invite qui indique la fonction que vous souhaitez utiliser.

Cet exemple montre comment générer un dictionnaire de clés d'identification utilisateur pour nommer des valeurs, puis comment générer une fonction de tri à l'aide de l'algorithme de tri à bulles.

Entrée :

```
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
    '''
```

Sortie :

```
    @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)
```

En outre, vous pouvez créer des instructions pour les procédures standard. Dans cet exemple, l'utilisateur crée une invite pour stocker le dictionnaire dans une table Amazon DynamoDB. Au début du script, l'utilisateur ajoute une invite à ajouter des bibliothèques DynamoDB afin de fournir un contexte supplémentaire à Amazon Q Developer.

Entrée :

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

Sortie :

```
'''
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')
```

Désormais, l'utilisateur peut facilement demander à Amazon Q de l'aider à stocker le dictionnaire dans DynamoDB.

Entrée :

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

Sortie :

```
@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'])
```

Générer des tests unitaires

Grâce aux commentaires intégrés, Amazon Q Developer peut générer des tests unitaires pour effectuer des assertions et d'autres tests. Dans cet exemple, l'utilisateur demande à Amazon Q de générer les bibliothèques applicables aux tests unitaires afin de définir le contexte du script de test. L'utilisateur demande ensuite à Amazon Q d'écrire une fonction de test pour cette `sort_user_id_name_map` fonction.

Entrée :

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

Sortie :

```
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()
```

En outre, lors d'une discussion entre l'utilisateur et Amazon Q Developer, celui-ci suggère et génère des tests unitaires basés sur les instructions de saisie du code de l'utilisateur. Pour plus d'informations, consultez la section Exemples de chat.

Java exemples

Procédez comme suit : Java des exemples montrent comment les utilisateurs peuvent commencer à générer des suggestions de code en ligne. Ces exemples illustratifs vous montrent comment

déclencher l'assistant de codage pour vos propres projets. La plupart des exemples suivants sont applicables à d'autres langues :

- Génération de classes et de fonctions
- Code du document
- Générer des algorithmes
- Générer des tests unitaires

Génération de classes et de fonctions

Pour générer une implémentation de classe complète ou partielle, utilisez des commentaires de code. Le commentaire décrit l'intention de la classe. Commencez simplement et ajoutez plus de détails si nécessaire. Après avoir généré une classe, vous pouvez continuer à générer des fonctions au sein de son squelette.

Lorsque vous rédigez des commentaires, il est préférable d'utiliser la structure de commentaires de la langue. Entrée Java, c'est `/** */`.

Entrée :

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

Sortie :

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

Maintenant qu'il existe une classe initiale, une fonction supplémentaire peut être générée avec des commentaires supplémentaires.

Entrée :

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

Sortie :

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

Code du document

Réduisez le temps consacré à la documentation du code en vous appuyant sur la documentation en ligne. Ajoutez un Java pause dans les commentaires`/* */`.

Entrée :

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

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

Si la génération en ligne n'a pas lieu, exécutez une suggestion en ligne d'Amazon Q Developer après avoir placé votre curseur à l'endroit où vous souhaitez obtenir de la documentation.

Sortie :

```
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;
        };
```

```
    }
}
```

Pour des explications plus simples du code (classes, fonctions ou éléments de script), utilisez des caractères directs (`//`) placés stratégiquement avant les sections de code dans lesquelles vous souhaitez ajouter de la documentation.

Entrée :

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

Sortie :

```
// 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;
        };
```

```
    }
}
```

Générer des algorithmes

Amazon Q Developer peut générer des algorithmes populaires, notamment liés à AWS SDKs. Dans votre script, ajoutez une invite qui indique la fonction que vous souhaitez utiliser.

Cet exemple génère une carte de hachage des clés d'identification utilisateur vers les valeurs des noms, puis génère une fonction de tri à l'aide de l'algorithme de tri par bulles.

Entrée :

```
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
    */


}
```

Sortie :

```
/*
* 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;
}
```

En outre, les utilisateurs peuvent créer des instructions pour les procédures standard. Dans cet exemple, l'utilisateur crée une invite pour stocker la carte de hachage dans une table Amazon DynamoDB. Au début du script, l'utilisateur ajoute une invite pour ajouter DynamoDB Java bibliothèques pour aider à fournir un contexte supplémentaire à Amazon Q Developer.

Entrée :

```
// 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
     */
```

Sortie :

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

Générer des tests unitaires

Grâce aux commentaires intégrés, Amazon Q Developer peut générer des tests unitaires pour effectuer des assertions et d'autres JUnit tests. Dans cet exemple, l'utilisateur demande d'abord à Amazon Q de générer les bibliothèques applicables JUnit pour définir le contexte du script de test. Ensuite, l'utilisateur crée une classe publique suivie d'un déclarateur et de chaînes de commentaires.

Entrée :

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

public class UnitTests {

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

Sortie :

```
@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);
}
```

En outre, lors d'une discussion entre l'utilisateur et Amazon Q Developer, celui-ci suggère et génère des tests unitaires basés sur les instructions de saisie du code de l'utilisateur. Pour plus d'informations, consultez la section Exemples de chat.

Exemples de chat

Amazon Q Developer Chat aide les développeurs à effectuer diverses tâches, telles que le codage, le test et la mise à niveau des applications, le diagnostic des erreurs, la réalisation d'analyses de sécurité et de correctifs, ainsi que l'optimisation AWS des ressources. Cette section explique comment le chat avec Amazon Q peut aider un développeur à effectuer des tâches de programmation classiques, telles que les suivantes :

- Renseignez-vous sur Services AWS
- Générer du code
- Générer des tests unitaires
- Expliquer le code

Renseignez-vous sur Services AWS

Les utilisateurs peuvent poser des questions aux développeurs Amazon Q Services AWS, par exemple pour trouver le bon service ou comprendre les meilleures pratiques.

Discutez de Services AWS

Utilisateur :

Quel est le meilleur service de conteneur à utiliser pour gérer ma charge de travail si je dois réduire mes coûts ?

Amazon Q :

Sur la base des informations fournies, il semble que vous recherchiez un service de conteneur rentable pour gérer vos charges de travail. Dans ce cas, je recommande d'envisager Amazon Elastic Container Service (Amazon ECS) avec Fargate. [1]

Amazon ECS with Fargate est un moteur de calcul sans serveur pour les conteneurs qui vous permet d'exécuter vos conteneurs sans avoir à gérer d'infrastructure sous-jacente. Avec Fargate, vous n'avez pas besoin de provisionner ou de gérer des serveurs : Fargate s'en charge pour vous. Cela peut vous aider à réduire vos coûts, car vous ne payez que pour les ressources de calcul réellement utilisées par vos conteneurs, sans avoir à payer pour la capacité de serveur inactive. [2]

Voici quelques avantages clés liés à l'utilisation d'Amazon ECS avec Fargate :

• Pas besoin de gérer des serveurs ou des clusters : Fargate s'occupe du provisionnement et de la gestion de l'infrastructure pour vous.

• Pay-as-you-go tarification : vous ne payez que pour les ressources informatiques utilisées par vos conteneurs, sans avoir à payer pour les capacités inutilisées.

• Dimensionnement automatique — Fargate peut automatiquement redimensionner vos conteneurs à la hausse ou à la baisse en fonction de la demande.

• Intégration étroite avec d'autres outils Services AWS tels que l'équilibrage de charge, la journalisation et la surveillance.

Les ressources fournies mentionnent également que le service ECS offre une flexibilité en termes de planification et de placement des conteneurs, ce qui peut vous aider à optimiser davantage vos coûts.

Je vous recommande de consulter le détail des tarifs d'Amazon ECS with Fargate sur AWS le site Web afin d'avoir une meilleure idée des coûts en fonction de vos exigences spécifiques en matière de charge de travail. Et bien sûr, n'hésitez pas à me faire savoir si vous avez d'autres questions !

Sources

[1] Quel service de AWS conteneur vous convient le mieux ?

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

[2] Java options de conteneur pour les charges de travail non HTTP

stackoverflow.com > Questions > 43417016

Mettre fin à la discussion sur Amazon Q Services AWS

Générer du code

Les utilisateurs peuvent demander de l'aide à Amazon Q Developer pour écrire des extraits de code.

Discuter de l'écriture de code

Utilisateur :

Pouvez-vous me donner un exemple de la façon de répertorier tous les compartiments S3 à l'aide du ? AWS SDK for Python (Boto)

Amazon Q :

Voici certainement un exemple de la façon de répertorier tous les compartiments S3 à l'aide du 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'])
```

Voici comment fonctionne le code :

1. Nous importons d'abord la bibliothèque boto3, qui est le SDK pour Python.
2. Nous créons un client S3 en utilisant`boto3.client('s3')`.
3. Nous appelons la `list_buckets()` méthode sur le client S3, qui renvoie une réponse contenant une liste de tous les compartiments du compte.
4. Nous parcourons ensuite la `Buckets` liste dans la réponse et imprimons le nom de chaque compartiment.

Il s'agit d'un exemple simple, mais il AWS SDK for Python (Boto) fournit de nombreuses autres fonctionnalités avancées pour travailler avec Amazon S3, telles que la création, la suppression et la gestion d'objets dans des compartiments, la définition de politiques de compartiment, etc. Vous trouverez d'autres exemples et de la documentation dans la documentation AWS SDK for Python (Boto)(). [2]

N'oubliez pas que lorsque vous travaillez avec Services AWS, il est important de suivre les meilleures pratiques en matière de gestion des informations d'identification et des informations sensibles, telles que le stockage des clés AWS d'accès dans des variables d'environnement au lieu de les coder en dur dans le code.

Sources : [1] Exemples Amazon S3 utilisant le SDK pour Python (AWS Boto3) - Exemples de code SDK docs.aws.amazon.com > code-library > latest-1x >... > python-s3.html

[2] Commencez à utiliser les buckets et les objets Amazon S3 à l'aide d'un AWS SDK - Exemples de code AWS SDK docs.aws.amazon.com > code-library > latest-1x >... > s3_example_s3_buckets_ _sect... GettingStarted

Fin de la discussion Amazon Q sur l'écriture de code

Générer des tests unitaires

Les utilisateurs peuvent envoyer le code sélectionné sous forme d'invite à Amazon Q Developer, où des instructions supplémentaires sont appliquées, telles que la génération d'un test unitaire. Pour step-by-step obtenir des instructions, consultez la section Envoyer du code à Amazon Q dans le manuel Amazon Q Developer User Guide.

Discuter des tests unitaires

Utilisateur :

Rédigez un test unitaire avec des données fictives pour la fonction suivante :

```
// 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 :

Pour écrire un test unitaire avec des données fictives pour la fonction PollMessages, nous pouvons utiliser un framework de simulation tel que Mockito. Voici un exemple :

```
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);
    }
}
```

Dans cet exemple, nous utilisons Mockito pour simuler le client AmazonSQS et configurer le comportement attendu de la méthode ReceiveMessage afin qu'elle renvoie deux messages fictifs dont les corps sont « message1 » et « message2 ». Ensuite, nous appelons la méthode PollMessages et affirmons que la liste de messages renvoyée correspond aux valeurs attendues.

Les points essentiels sont les suivants :

1. Nous utilisons l'annotation @ ExtendWith (MockitoExtension.class) pour activer l'intégration de Mockito avec. JUnit
2. Nous utilisons l'annotation @Mock pour créer une instance fictive du client AmazonSQS.
3. Nous utilisons l'InjectMocks annotation @ pour injecter le faux client AmazonSQS dans la SQSUtils classe.
4. Dans la méthode de test, nous avons défini le comportement attendu de la méthode ReceiveMessage à l'aide de Mockito.when () .thenReturn ().
5. Nous appelons la méthode PollMessages et affirmons que la liste des messages renvoyés correspond aux valeurs attendues.

Cette approche nous permet de tester la méthode PollMessages sans réellement interagir avec le service Amazon SQS, ce qui peut rendre le test plus rapide et plus fiable.

Fin de la discussion Amazon Q sur les tests unitaires

Expliquer le code

Pour les processus de révision du code, les développeurs peuvent envoyer une sélection de code à Amazon Q Developer afin de mieux comprendre le code. En outre, l'explication fournie par Amazon Q peut également servir à des fins de documentation. Dans l'exemple suivant, Amazon Q explique la fonction de gestion à l'aide d'une AWS Lambda fonction, puis propose une docstring.

Discuter sur l'explication du code

Utilisateur :

Expliquez la partie suivante de mon code :

```
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 :

Le bloc de code fourni semble être une AWS Lambda fonction écrite en Python. Permettez-moi de vous expliquer les principaux aspects de ce code :

1. La fonction s'appelle lambda_handler et constitue le point d'entrée de la fonction Lambda.
2. Il utilise la bibliothèque boto3 pour interagir avec AWS DynamoDB et les services S3.
3. La fonction extrait une clé d'objet S3 et un nom de compartiment à partir des données d'événements entrantes.
4. Il appelle ensuite une fonction detect_text (qui n'apparaît pas dans le code fourni) pour extraire le contenu en texte intégral de l'objet S3.
5. La fonction récupère également les métadonnées relatives à l'objet S3 à l'aide de la méthode head_object.
6. Il extrait ensuite l'heure actuelle et appelle une fonction detect_restaurant_closure (également non illustrée) pour déterminer la raison et le résultat de la fermeture d'un restaurant sur la base du texte extrait.
7. Enfin, la fonction stocke les informations suivantes dans une table DynamoDB :
   - La clé de l'objet S3 comme identifiant
   - L'heure de création
   - La raison et le résultat de la fermeture du restaurant
   - Le texte extrait de l'objet S3

Dans l'ensemble, cette fonction Lambda semble faire partie d'un système plus vaste qui traite les objets S3, extrait le contenu du texte, l'analyse pour y détecter les informations relatives à la fermeture des restaurants et stocke les résultats dans une table DynamoDB. Les détails spécifiques des fonctions detect_text et detect_restaurant_closure seraient nécessaires pour bien comprendre les fonctionnalités de ce code.

Fin de la réponse d'Amazon Q

Utilisateur : résumez cette explication dans une docstring pour cette fonction lambda_handler

Amazon Q : Voici une chaîne de documentation suggérée pour la fonction 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...
```

Cette docstring fournit un résumé de haut niveau de l'objectif de la fonction, des paramètres d'entrée attendus et de la valeur de retour attendue (ou de son absence, dans ce cas). Cela permet à l'utilisateur de comprendre clairement ce que fait la fonction lambda_handler sans entrer dans les détails de l'implémentation.

Sources : [1] Définissez le gestionnaire de fonctions Lambda en C# - AWS Lambda https://docs.aws.amazon.com/lambda/latest/dg/csharp-handler.html - : ~:text=Le gestionnaire de fonctions Lambda est, sort ou expire.

Fin de la discussion Amazon Q sur l'explication du code

Résolution des problèmes liés aux scénarios de génération de code dans Amazon Q Developer

Lorsque vous utilisez Amazon Q Developer, vous pouvez être confronté aux scénarios courants suivants, avec une génération de code et des résolutions inexactes :

- Génération de code vide
- Commentaires continus
- Génération de code en ligne incorrecte
- Résultats inadéquats des chats

Génération de code vide

Lors du développement du code, vous pouvez rencontrer les problèmes suivants :

- Amazon Q ne fournit aucune suggestion.
- Le message « Aucune suggestion d'Amazon Q » apparaît dans votre IDE.

Vous pensez peut-être d'abord qu'Amazon Q ne fonctionne pas correctement. Cependant, la cause première de ces problèmes est généralement associée au contexte du script ou du projet ouvert dans l'IDE.

Si Amazon Q Developer ne fournit pas de suggestion automatiquement, vous pouvez utiliser les raccourcis suivants pour exécuter manuellement les suggestions en ligne d'Amazon Q :

- PC - Alt+C
- macOS - Option+C

Pour plus d'informations, consultez la section Utilisation des touches de raccourci dans le manuel Amazon Q Developer User Guide.

Dans la plupart des scénarios, Amazon Q génère une suggestion. Lorsqu'Amazon Q renvoie un message « Aucune suggestion de la part d'Amazon Q », passez en revue les solutions courantes suivantes pour ce problème :

- Contexte inadéquat — Assurez-vous que les bibliothèques applicables sont présentes. Assurez-vous que les classes et les fonctions définies dans le script sont liées au nouveau code.
- Demande ambiguë : si l'invite prête à confusion ou n'est pas claire, Amazon Q sera moins performant lorsqu'il proposera du code en ligne. Dans ces scénarios, soyez plus précis dans votre code et dans vos commentaires afin d'obtenir des réponses appropriées.

Si vous êtes bloqué, continuez à coder au mieux de vos capacités. En général, Amazon Q reconnaît votre intention, ce qui l'incite à proposer des suggestions. Une autre option consiste à envoyer un extrait de code avec des instructions rapides dans un chat Amazon Q. Pour plus d'informations, consultez la section Envoyer du code à Amazon Q dans le manuel Amazon Q Developer User Guide.

Commentaires continus

Vous pouvez rédiger une invite à l'aide de commentaires et ne recevoir que des lignes de commentaires génériques ou répétées de la part d'Amazon Q.

Entrée :

```
// Kafka test
```

Résultat:

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

Ce problème est généralement dû à un manque de contexte dans le script ou à de mauvaises instructions rapides. Nous vous recommandons d'utiliser le format standard pour les commentaires en fonction de la langue. Dans cet exemple, l'utilisateur a modifié l'invite à utiliser `/* */` pour un bloc de commentaires, et cette modification a amélioré les performances d'Amazon Q.

Entrée :

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

Résultat:

Dans ce résultat, l'utilisateur a accepté plusieurs suggestions de manière séquentielle.

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

Génération de code en ligne incorrecte

Parfois, les suggestions en ligne d'Amazon Q Developer ne sont pas utiles ou ne répondent pas à l'objectif de votre invite. Parfois, les suggestions de code en ligne peuvent fournir une fonction générique similaire à une simple méthode Hello World.

Lorsque les suggestions de code intégrées ne sont pas utiles, essayez les meilleures pratiques suivantes :

- Passez en revue le contexte du script. En général, plus vous fournissez de contexte, meilleurs sont les résultats que vous obtenez avec Amazon Q.
- Si le contexte varie et/ou est bruyant, modularisez votre code. Regroupez des fonctions similaires dans le même script. Séparez les différentes classes et fonctions en fonction de leur objectif.
- Rédigez les noms de vos classes et de vos fonctions de manière tactique pour aider Amazon Q à fournir de meilleures recommandations.

Dans cet exemple, le développeur écrit un script à partir de zéro qui télécharge des fichiers sur Hadoop.

Entrée :

```
class  MyFunction{

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

Résultat:

```
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));

}
```

Bien qu'Amazon Q ait fourni une recommandation, celle-ci n'a pas répondu aux attentes du développeur. Pour améliorer les performances, le développeur importe des bibliothèques et modifie le nom de classe pour représenter les utilitaires Hadoop.

Correctif :

```
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
}
```

Résultat:

Dans ce résultat, le développeur a accepté plusieurs suggestions de manière séquentielle.

```
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!");
}
```

Dans le même HadoopUtilities script, le développeur ajoute une invite de commentaire pour lire une rubrique relative à Amazon Simple Queue Service (Amazon SQS).

Entrée :

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

Résultat:

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

Plutôt que d'utiliser des méthodes Java pour lire un message SQS depuis Amazon SQS, Amazon Q suggère d'imprimer une méthode sous forme de chaîne. Cependant, cette suggestion ne répond pas à l'objectif initial du développeur.

Dans ce cas, il est préférable de re-modulariser le code. Le script se concentre sur les méthodes Hadoop. Par conséquent, il peut être préférable de créer un script utilitaire spécialement conçu pour les opérations SQS.

La création d'une classe distincte pour les utilitaires SQS améliore la suggestion en ligne, qui peut être utilisée dans HadoopUtilities script.

Entrée :

```
// 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
}
```

Résultat:

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

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

Si la suggestion en ligne ne donne toujours pas les résultats escomptés, demandez à Amazon Q Developer de générer du code lors d'une session de chat.

Discuter de la génération de code

Utilisateur :

Génère une fonction Java pour lire un message depuis une file d'attente Amazon SQS

Amazon Q :

Voici un exemple de fonction Java permettant de lire un message depuis une file d'attente 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;
        }
    }
```

Ce code crée une classe SQSMessage Reader qui prend un client AmazonSQS et une URL de file d'attente comme paramètres de constructeur. La méthode readMessage () utilise le ReceiveMessageRequest pour extraire jusqu'à 1 message de la file d'attente, avec un temps d'attente maximal de 10 secondes.

Si un message est reçu, le corps du message est imprimé et le message est supprimé de la file d'attente à l'aide du DeleteMessageRequest. Le corps du message est ensuite renvoyé. Si aucun message n'est reçu, nul est renvoyé.

Fin de la discussion Amazon Q sur la génération de code

Résultats inadéquats des chats

Lors du développement du code, les développeurs peuvent demander à Amazon Q de générer des fonctions communes, de fournir des recommandations ou d'expliquer le code. Parfois, à la suite d'une demande (telle qu'une question ou un extrait de code), Amazon Q peut fournir des commentaires généraux ou des résultats qui ne répondent pas à vos attentes. Dans ces scénarios, essayez ce qui suit :

- Testez différentes instructions pour améliorer les résultats de génération de code fournis par Amazon Q.

- Suivez URLs les instructions fréquemment fournies par Amazon Q dans ses réponses à vos questions ou à vos questions.
- Faites-nous part de vos commentaires afin d'améliorer les performances futures d'Amazon Q.

Par exemple, un développeur est en train de créer une solution RAG (Retrieval Augmented Generation) et il doit fragmenter son document en fonction des relations sémantiques entre les phrases. Le développeur discute avec Amazon Q comme suit.

Discuter d'une solution RAG

Utilisateur :

Pour une solution RAG, générez du code pour le découpage sémantique

Amazon Q :

Voici certainement un exemple de la façon dont vous pourriez implémenter le découpage sémantique pour une solution de génération augmentée de récupération (RAG) dans Python en utilisant la LangChain bibliothèque :

```
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
```

Dans cet exemple, nous utilisons le TokenTextSplitter de LangChain pour diviser le texte d'entrée en blocs de 500 caractères avec un chevauchement de 50 caractères. Le HuggingFaceEmbeddings le modèle est utilisé pour générer des intégrations pour chaque morceau, et le Chroma Un magasin vectoriel est utilisé pour stocker les morceaux et leurs intégrations.

La fonction semantic_chunk prend le texte saisi, le divise en morceaux à l'aide du text_splitter, génère des intégrations pour chaque segment à l'aide du modèle d'intégration, puis ajoute les morceaux et leurs intégrations au vectorstore.

Cette approche vous permet de diviser le texte d'entrée en segments sémantiquement significatifs, ce qui peut être utile pour diverses tâches, telles que la réponse à des questions ou la synthèse, dans une solution RAG.

Notez qu'il s'agit d'un exemple simple et que vous devrez peut-être ajuster les paramètres (par exemple, chunk_size, chunk_overlap) ou le choix du séparateur de texte et du modèle d'intégration en fonction de votre cas d'utilisation et de vos données spécifiques.

Fin de la discussion Amazon Q à propos d'une solution RAG

Toutefois, cette approche ne fournit qu'un moyen simple de fragmenter des documents en utilisant la technique de la fenêtre coulissante. Le développeur souhaite une méthode avancée qui repose sur la mesure des intégrations. Dans ces scénarios, le développeur doit ajouter plus de détails à l'invite en fonction de son objectif. Lorsque le développeur fait référence à la fonction connue de séparation de AI21 texte avec LangChain, il reçoit une réponse améliorée, comme indiqué dans le chat suivant.

Discuter d'une solution RAG (avec une invite améliorée)

Utilisateur :

L'invite ci-dessus concerne uniquement le fractionnement en fonction de la taille du jeton par morceau. Je souhaite utiliser un séparateur de texte AI21 sémantique. Générez une fonction en utilisant le découpage sémantique de texte A121 avec langchain.

Amazon Q :

OK, utilisons le séparateur de texte AI21 sémantique LangChain pour implémenter le découpage sémantique. Voici un exemple : [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
```

Dans cet exemple, nous utilisons le formulaire AI21 SemanticTextSplitter LangChain pour diviser le texte saisi en segments sémantiquement significatifs. Le HuggingFaceEmbeddings le modèle est utilisé pour créer des intégrations pour chaque morceau, et le Chroma Un magasin vectoriel est utilisé pour stocker les morceaux et leurs intégrations.

La fonction semantic_chunk prend le texte d'entrée, le divise en morceaux à l'aide du text_splitter, crée des intégrations pour chaque segment à l'aide du modèle d'intégration, puis ajoute les morceaux et leurs intégrations au vectorstore.

Cette approche vous permet de diviser le texte d'entrée en segments sémantiquement significatifs, ce qui peut être utile pour diverses tâches, telles que la réponse à des questions ou la synthèse, dans une solution RAG.

Veuillez noter que vous devrez peut-être ajuster les paramètres ou le choix du modèle d'intégration en fonction de votre cas d'utilisation et de vos données spécifiques.

Sources [1] Apprenez à utiliser les séparateurs de texte dans LangChain

soosweb3.hashnode.dev > -séparateurs de texte the-ultimate-langchain-series

Fin de la discussion Amazon Q à propos d'une solution RAG (avec invite améliorée)

Sur la base des résultats du chat, le développeur peut utiliser la fonction et accéder à la source des informations.

En général, les sujets et questions avancés nécessitent un contexte plus spécifique lorsque vous discutez avec Amazon Q Developer. Si vous pensez que les résultats de votre chat ne sont pas exacts, utilisez l'icône représentant un pouce vers le bas pour nous faire part de vos commentaires sur la réponse d'Amazon Q. Amazon Q Developer utilise en permanence les commentaires pour améliorer les futures versions. Pour les interactions qui ont produit des résultats positifs, il est utile de fournir vos commentaires à l'aide de l'icône représentant le pouce levé.

FAQsà propos d'Amazon Q Developer

Cette section fournit des réponses aux questions fréquemment posées concernant l'utilisation de Developer Amazon Q pour le développement de code.

Qu'est-ce qu'Amazon Q Developer ?

Amazon Q Developer est un puissant service génératif basé sur l'IA conçu pour accélérer les tâches de développement de code en fournissant une génération de code intelligente et des recommandations. Le 30 avril 2024, Amazon a CodeWhisperer rejoint Amazon Q Developer.

Comment accéder à Amazon Q Developer ?

Amazon Q Developer est disponible dans le cadre des AWS boîtes à outils pour Visual Studio Code et JetBrains IDEs, tels que IntelliJ et. PyCharm Pour commencer, installez la dernière AWS Toolkit version.

Quels sont les langages de programmation pris en charge par Amazon Q Developer ?

Pour Visual Studio Code et JetBrains IDEs, Amazon Q Developer prend en charge Python, Java, JavaScript, TypeScript, C#, Go, Rust, PHP, Ruby, Kotlin, C, C++, scripts Shell et SQL Scala. Bien que ce guide se concentre sur Python and Java à titre d'exemple, les concepts sont applicables à tous les langages de programmation pris en charge.

Comment puis-je fournir du contexte à Amazon Q Developer pour une meilleure génération de code ?

Commencez par le code existant, importez les bibliothèques pertinentes, créez des classes et des fonctions ou établissez des squelettes de code. Utilisez des blocs de commentaires standard pour les instructions en langage naturel. Concentrez votre script sur des objectifs spécifiques et modularisez les fonctionnalités distinctes dans des scripts distincts avec un contexte pertinent. Pour plus d'informations, consultez la section Meilleures pratiques de codage avec Amazon Q Developer.

Que dois-je faire si la génération de code en ligne avec Amazon Q Developer n'est pas précise ?

Vérifiez le contexte du script, assurez-vous que les bibliothèques sont présentes et que les classes et les fonctions sont liées au nouveau code. Modularisez votre code et séparez les différentes classes et fonctions en fonction de leur objectif. Rédigez des instructions ou des commentaires clairs et précis. Si vous n'êtes toujours pas sûr de l'exactitude du code et que vous ne pouvez pas continuer, lancez une discussion avec Amazon Q et envoyez-lui l'extrait de code avec les instructions. Pour plus d'informations, consultez la section Résolution des problèmes liés aux scénarios de génération de code dans Amazon Q Developer.

Comment puis-je utiliser la fonctionnalité de chat Amazon Q Developer pour générer du code et résoudre les problèmes ?

Discutez avec Amazon Q pour générer des fonctions communes, demander des recommandations ou expliquer le code. Si la réponse initiale n'est pas satisfaisante, testez différentes instructions et suivez les instructions fourniesURLs. Envoyez également des commentaires à Amazon Q afin de l'aider à améliorer ses futures performances de chat. Utilisez les icônes du pouce vers le haut et du pouce vers le bas pour nous faire part de vos commentaires. Pour plus d'informations, consultez la section Exemples de chat.

Quelles sont les bonnes pratiques pour utiliser Amazon Q Developer ?

Fournissez un contexte pertinent, testez et répétez les instructions, examinez les suggestions de code avant de les accepter, utilisez les fonctionnalités de personnalisation et comprenez les politiques de confidentialité des données et d'utilisation du contenu. Pour plus d'informations, consultez les meilleures pratiques pour la génération de code avec Amazon Q Developer et les meilleures pratiques pour les recommandations de code avec Amazon Q Developer.

Puis-je personnaliser Amazon Q Developer pour générer des recommandations basées sur mon propre code ?

Oui, utilisez les personnalisations, qui constituent une fonctionnalité avancée d'Amazon Q Developer. Grâce aux personnalisations, les entreprises peuvent fournir leurs propres référentiels de code pour

permettre à Amazon Q Developer de recommander des suggestions de code en ligne. Pour plus d'informations, consultez la section Fonctionnalités avancées d'Amazon Q Developer and Resources.

Prochaines étapes de l'utilisation d'Amazon Q Developer

Grâce aux connaissances acquises grâce à ce guide complet, vous pouvez utiliser Amazon Q Developer de manière efficace dans votre flux de travail de codage. Installez-le AWS Toolkit dans votre IDE préféré (Visual Studio Code ou JetBrains) et commencez à explorer la génération de code générative basée sur l'IA et les recommandations d'Amazon Q Developer.

Le moyen le plus efficace d'exploiter tout le potentiel d'Amazon Q Developer consiste à acquérir une expérience pratique avec votre propre code. Lorsque vous intégrez Amazon Q à votre cycle de développement, consultez ce guide pour découvrir les meilleures pratiques, le dépannage et des exemples concrets.

En outre, tenez-vous au courant en consultant les AWS blogs et les guides de développement référencés dans Ressources. Ces ressources fournissent les dernières mises à jour, les meilleures pratiques et les informations nécessaires pour vous aider à optimiser votre utilisation d'Amazon Q Developer.

Vos commentaires sont précieux pour améliorer ce guide et l'aider à rester une ressource précieuse pour les développeurs. Partagez vos expériences, vos défis et vos suggestions pour les versions futures. Vos commentaires permettront d'améliorer le guide avec des exemples supplémentaires, des scénarios de résolution des problèmes et des informations adaptées à vos besoins.

Ressources

AWS blogues

- Accélération du cycle de développement de vos logiciels avec Amazon Q
- Réinventer le développement logiciel avec l'agent de développement Amazon Q
- Cinq exemples de résolution des problèmes avec Amazon Q
- Personnalisez Amazon Q Developer dans votre environnement IDE grâce à votre base de code privée
- Trois manières dont l'agent Amazon Q Developer pour la transformation du code accélère les mises à niveau de Java
- Tirer parti d'Amazon Q Developer pour un débogage et une maintenance de code efficaces
- Tester vos applications avec Amazon Q Developer

AWS documentation

- Guide de l'utilisateur Amazon Q pour les développeurs
- Personnalisation du code Amazon Q Developer
- Transformation du code Amazon Q Developer

AWS ateliers

- Journée d'immersion pour développeurs Amazon Q
- Atelier pour développeurs Amazon Q - Création de l'application Q-Words
- Atelier pour développeurs Amazon Q - Création d'instructions efficaces

Collaborateurs

Les personnes suivantes ont contribué à ce guide :

- Joe King, data scientist senior, AWS
- Prateek Gupta, chef d'équipe — Sr. CAA, AWS
- Manohar Reddy Arranagu, architecte, DevOps AWS
- Soumik Roy, architecte d'applications cloud, AWS
- Sanket Shinde, consultante, AWS

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un RSSfil RSS.

Modification	Description	Date
Publication initiale	—	16 août 2024

AWS Glossaire des directives prescriptives

Les termes suivants sont couramment utilisés dans les stratégies, les guides et les modèles fournis par les directives AWS prescriptives. Pour suggérer des entrées, veuillez utiliser le lien Faire un commentaire à la fin du glossaire.

Nombres

7 R

Sept politiques de migration courantes pour transférer des applications vers le cloud. Ces politiques s'appuient sur les 5 R identifiés par Gartner en 2011 et sont les suivantes :

- Refactorisation/réarchitecture : transférez une application et modifiez son architecture en tirant pleinement parti des fonctionnalités natives cloud pour améliorer l'agilité, les performances et la capacité de mise à l'échelle. Cela implique généralement le transfert du système d'exploitation et de la base de données. Exemple : migrez votre base de données Oracle sur site vers l'édition compatible avec Amazon Aurora PostgreSQL.
- Replateformer (déplacer et remodeler) : transférez une application vers le cloud et introduisez un certain niveau d'optimisation pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle sur site vers Amazon Relational Database Service (Amazon RDS) pour Oracle dans le. AWS Cloud
- Racheter (rachat) : optez pour un autre produit, généralement en passant d'une licence traditionnelle à un modèle SaaS. Exemple : migrez votre système de gestion de la relation client (CRM) vers Salesforce.com.
- Réhéberger (lift and shift) : transférez une application vers le cloud sans apporter de modifications pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle locale vers Oracle sur une EC2 instance du AWS Cloud.
- Relocaliser (lift and shift au niveau de l'hyperviseur) : transférez l'infrastructure vers le cloud sans acheter de nouveau matériel, réécrire des applications ou modifier vos opérations existantes. Vous migrez des serveurs d'une plateforme sur site vers un service cloud pour la même plateforme. Exemple : migrer une Microsoft Hyper-V application vers AWS.
- Retenir : conservez les applications dans votre environnement source. Il peut s'agir d'applications nécessitant une refactorisation majeure, que vous souhaitez retarder, et d'applications existantes que vous souhaitez retenir, car rien ne justifie leur migration sur le plan commercial.

- Retirer : mettez hors service ou supprimez les applications dont vous n'avez plus besoin dans votre environnement source.

A

ABAC

Voir contrôle d'accès basé sur les attributs.

services abstraits

Consultez la section Services gérés.

ACIDE

Voir atomicité, consistance, isolation, durabilité.

migration active-active

Méthode de migration de base de données dans laquelle la synchronisation des bases de données source et cible est maintenue (à l'aide d'un outil de réplication bidirectionnelle ou d'opérations d'écriture double), tandis que les deux bases de données gèrent les transactions provenant de la connexion d'applications pendant la migration. Cette méthode prend en charge la migration par petits lots contrôlés au lieu d'exiger un basculement ponctuel. Elle est plus flexible mais demande plus de travail qu'une migration active-passive.

migration active-passive

Méthode de migration de base de données dans laquelle la synchronisation des bases de données source et cible est maintenue, mais seule la base de données source gère les transactions provenant de la connexion d'applications pendant que les données sont répliquées vers la base de données cible. La base de données cible n'accepte aucune transaction pendant la migration.

fonction d'agrégation

Fonction SQL qui agit sur un groupe de lignes et calcule une valeur de retour unique pour le groupe. Des exemples de fonctions d'agrégation incluent `SUM` et`MAX`.

AI

Voir intelligence artificielle.

AIOps

Voir les opérations d'intelligence artificielle.

anonymisation

Processus de suppression définitive d'informations personnelles dans un ensemble de données. L'anonymisation peut contribuer à protéger la vie privée. Les données anonymisées ne sont plus considérées comme des données personnelles.

anti-motif

Solution fréquemment utilisée pour un problème récurrent lorsque la solution est contre-productive, inefficace ou moins efficace qu'une alternative.

contrôle des applications

Une approche de sécurité qui permet d'utiliser uniquement des applications approuvées afin de protéger un système contre les logiciels malveillants.

portefeuille d'applications

Ensemble d'informations détaillées sur chaque application utilisée par une organisation, y compris le coût de génération et de maintenance de l'application, ainsi que sa valeur métier. Ces informations sont essentielles pour le processus de découverte et d'analyse du portefeuille et permettent d'identifier et de prioriser les applications à migrer, à moderniser et à optimiser.

intelligence artificielle (IA)

Domaine de l'informatique consacré à l'utilisation des technologies de calcul pour exécuter des fonctions cognitives généralement associées aux humains, telles que l'apprentissage, la résolution de problèmes et la reconnaissance de modèles. Pour plus d'informations, veuillez consulter Qu'est-ce que l'intelligence artificielle ?

opérations d'intelligence artificielle (AIOps)

Processus consistant à utiliser des techniques de machine learning pour résoudre les problèmes opérationnels, réduire les incidents opérationnels et les interventions humaines, mais aussi améliorer la qualité du service. Pour plus d'informations sur son AIOps utilisation dans la stratégie de AWS migration, consultez le guide d'intégration des opérations.

chiffrement asymétrique

Algorithme de chiffrement qui utilise une paire de clés, une clé publique pour le chiffrement et une clé privée pour le déchiffrement. Vous pouvez partager la clé publique, car elle n'est pas utilisée pour le déchiffrement, mais l'accès à la clé privée doit être très restreint.

atomicité, cohérence, isolement, durabilité (ACID)

Ensemble de propriétés logicielles garantissant la validité des données et la fiabilité opérationnelle d'une base de données, même en cas d'erreur, de panne de courant ou d'autres problèmes.

contrôle d'accès par attributs (ABAC)

Pratique qui consiste à créer des autorisations détaillées en fonction des attributs de l'utilisateur, tels que le service, le poste et le nom de l'équipe. Pour plus d'informations, consultez ABAC pour AWS dans la documentation AWS Identity and Access Management (IAM).

source de données faisant autorité

Emplacement où vous stockez la version principale des données, considérée comme la source d'information la plus fiable. Vous pouvez copier les données de la source de données officielle vers d'autres emplacements à des fins de traitement ou de modification des données, par exemple en les anonymisant, en les expurgant ou en les pseudonymisant.

Zone de disponibilité

Un emplacement distinct au sein d'un Région AWS réseau isolé des défaillances dans d'autres zones de disponibilité et fournissant une connectivité réseau peu coûteuse et à faible latence aux autres zones de disponibilité de la même région.

AWS Cadre d'adoption du cloud (AWS CAF)

Un cadre de directives et de meilleures pratiques visant AWS à aider les entreprises à élaborer un plan efficace pour réussir leur migration vers le cloud. AWS La CAF organise ses conseils en six domaines prioritaires appelés perspectives : les affaires, les personnes, la gouvernance, les plateformes, la sécurité et les opérations. Les perspectives d'entreprise, de personnes et de gouvernance mettent l'accent sur les compétences et les processus métier, tandis que les perspectives relatives à la plateforme, à la sécurité et aux opérations se concentrent sur les compétences et les processus techniques. Par exemple, la perspective liée aux personnes cible les parties prenantes qui s'occupent des ressources humaines (RH), des fonctions de dotation en personnel et de la gestion des personnes. Dans cette perspective, la AWS CAF fournit des conseils pour le développement du personnel, la formation et les communications afin de préparer

l'organisation à une adoption réussie du cloud. Pour plus d'informations, veuillez consulter le site Web AWS CAF et le livre blanc AWS CAF.

AWS Cadre de qualification de la charge de travail (AWS WQF)

Outil qui évalue les charges de travail liées à la migration des bases de données, recommande des stratégies de migration et fournit des estimations de travail. AWS Le WQF est inclus avec AWS Schema Conversion Tool (AWS SCT). Il analyse les schémas de base de données et les objets de code, le code d'application, les dépendances et les caractéristiques de performance, et fournit des rapports d'évaluation.

B

mauvais bot

Un bot destiné à perturber ou à nuire à des individus ou à des organisations.

BCP

Consultez la section Planification de la continuité des activités.

graphique de comportement

Vue unifiée et interactive des comportements des ressources et des interactions au fil du temps. Vous pouvez utiliser un graphique de comportement avec Amazon Detective pour examiner les tentatives de connexion infructueuses, les appels d'API suspects et les actions similaires. Pour plus d'informations, veuillez consulter Data in a behavior graph dans la documentation Detective.

système de poids fort

Système qui stocke d'abord l'octet le plus significatif. Voir aussi endianité.

classification binaire

Processus qui prédit un résultat binaire (l'une des deux classes possibles). Par exemple, votre modèle de machine learning peut avoir besoin de prévoir des problèmes tels que « Cet e-mail est-il du spam ou non ? » ou « Ce produit est-il un livre ou une voiture ? ».

filtre de Bloom

Structure de données probabiliste et efficace en termes de mémoire qui est utilisée pour tester si un élément fait partie d'un ensemble.

déploiement bleu/vert

Stratégie de déploiement dans laquelle vous créez deux environnements distincts mais identiques. Vous exécutez la version actuelle de l'application dans un environnement (bleu) et la nouvelle version de l'application dans l'autre environnement (vert). Cette stratégie vous permet de revenir rapidement en arrière avec un impact minimal.

bot

Application logicielle qui exécute des tâches automatisées sur Internet et simule l'activité ou l'interaction humaine. Certains robots sont utiles ou bénéfiques, comme les robots d'exploration Web qui indexent des informations sur Internet. D'autres robots, appelés « bots malveillants », sont destinés à perturber ou à nuire à des individus ou à des organisations.

botnet

Réseaux de robots infectés par des logiciels malveillants et contrôlés par une seule entité, connue sous le nom d'herder ou d'opérateur de bots. Les botnets sont le mécanisme le plus connu pour faire évoluer les bots et leur impact.

branche

Zone contenue d'un référentiel de code. La première branche créée dans un référentiel est la branche principale. Vous pouvez créer une branche à partir d'une branche existante, puis développer des fonctionnalités ou corriger des bogues dans la nouvelle branche. Une branche que vous créez pour générer une fonctionnalité est communément appelée branche de fonctionnalités. Lorsque la fonctionnalité est prête à être publiée, vous fusionnez à nouveau la branche de fonctionnalités dans la branche principale. Pour plus d'informations, consultez À propos des branches (GitHub documentation).

accès par brise-vitre

Dans des circonstances exceptionnelles et par le biais d'un processus approuvé, c'est un moyen rapide pour un utilisateur d'accéder à un accès auquel Compte AWS il n'est généralement pas autorisé. Pour plus d'informations, consultez l'indicateur Implementation break-glass procedures dans le guide Well-Architected AWS .

stratégie existante (brownfield)

L'infrastructure existante de votre environnement. Lorsque vous adoptez une stratégie existante pour une architecture système, vous concevez l'architecture en fonction des contraintes des systèmes et de l'infrastructure actuels. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

cache de tampon

Zone de mémoire dans laquelle sont stockées les données les plus fréquemment consultées.

capacité métier

Ce que fait une entreprise pour générer de la valeur (par exemple, les ventes, le service client ou le marketing). Les architectures de microservices et les décisions de développement peuvent être dictées par les capacités métier. Pour plus d'informations, veuillez consulter la section Organisation en fonction des capacités métier du livre blanc Exécution de microservices conteneurisés sur AWS.

planification de la continuité des activités (BCP)

Plan qui tient compte de l'impact potentiel d'un événement perturbateur, tel qu'une migration à grande échelle, sur les opérations, et qui permet à une entreprise de reprendre ses activités rapidement.

C

CAF

Voir le cadre d'adoption du AWS cloud.

déploiement de Canary

Diffusion lente et progressive d'une version pour les utilisateurs finaux. Lorsque vous êtes sûr, vous déployez la nouvelle version et remplacez la version actuelle dans son intégralité.

CCo E

Voir le Centre d'excellence du cloud.

CDC

Voir capture des données de modification.

capture des données de modification (CDC)

Processus de suivi des modifications apportées à une source de données, telle qu'une table de base de données, et d'enregistrement des métadonnées relatives à ces modifications. Vous pouvez utiliser la CDC à diverses fins, telles que l'audit ou la réplication des modifications dans un système cible afin de maintenir la synchronisation.

ingénierie du chaos

Introduire intentionnellement des défaillances ou des événements perturbateurs pour tester la résilience d'un système. Vous pouvez utiliser AWS Fault Injection Service (AWS FIS) pour effectuer des expériences qui stressent vos AWS charges de travail et évaluer leur réponse.

CI/CD

Découvrez l'intégration continue et la livraison continue.

classification

Processus de catégorisation qui permet de générer des prédictions. Les modèles de ML pour les problèmes de classification prédisent une valeur discrète. Les valeurs discrètes se distinguent toujours les unes des autres. Par exemple, un modèle peut avoir besoin d'évaluer la présence ou non d'une voiture sur une image.

chiffrement côté client

Chiffrement des données localement, avant que la cible ne les Service AWS reçoive.

Centre d'excellence du cloud (CCoE)

Une équipe multidisciplinaire qui dirige les efforts d'adoption du cloud au sein d'une organisation, notamment en développant les bonnes pratiques en matière de cloud, en mobilisant des ressources, en établissant des délais de migration et en guidant l'organisation dans le cadre de transformations à grande échelle. Pour plus d'informations, consultez les CCoarticles électroniques du blog sur la stratégie AWS Cloud d'entreprise.

cloud computing

Technologie cloud généralement utilisée pour le stockage de données à distance et la gestion des appareils IoT. Le cloud computing est généralement associé à la technologie informatique de pointe.

modèle d'exploitation du cloud

Dans une organisation informatique, modèle d'exploitation utilisé pour créer, faire évoluer et optimiser un ou plusieurs environnements cloud. Pour plus d'informations, consultez la section Création de votre modèle d'exploitation cloud.

étapes d'adoption du cloud

Les quatre phases que les entreprises traversent généralement lorsqu'elles migrent vers AWS Cloud :

- Projet : exécution de quelques projets liés au cloud à des fins de preuve de concept et d'apprentissage
- Base : réaliser des investissements fondamentaux pour accélérer votre adoption du cloud (par exemple, créer une zone de landing zone, définir un CCo E, établir un modèle opérationnel)
- Migration : migration d'applications individuelles
- Réinvention : optimisation des produits et services et innovation dans le cloud

Ces étapes ont été définies par Stephen Orban dans le billet de blog The Journey Toward Cloud-First & the Stages of Adoption publié sur le blog AWS Cloud Enterprise Strategy. Pour plus d'informations sur leur lien avec la stratégie de AWS migration, consultez le guide de préparation à la migration.

CMDB

Voir base de données de gestion de configuration.

référentiel de code

Emplacement où le code source et d'autres ressources, comme la documentation, les exemples et les scripts, sont stockés et mis à jour par le biais de processus de contrôle de version. Les référentiels cloud courants incluent GitHub ouBitbucket Cloud. Chaque version du code est appelée branche. Dans une structure de microservice, chaque référentiel est consacré à une seule fonctionnalité. Un seul pipeline CI/CD peut utiliser plusieurs référentiels.

cache passif

Cache tampon vide, mal rempli ou contenant des données obsolètes ou non pertinentes. Cela affecte les performances, car l'instance de base de données doit lire à partir de la mémoire principale ou du disque, ce qui est plus lent que la lecture à partir du cache tampon.

données gelées

Données rarement consultées et généralement historiques. Lorsque vous interrogez ce type de données, les requêtes lentes sont généralement acceptables. Le transfert de ces données vers des niveaux ou classes de stockage moins performants et moins coûteux peut réduire les coûts.

vision par ordinateur (CV)

Domaine de l'IA qui utilise l'apprentissage automatique pour analyser et extraire des informations à partir de formats visuels tels que des images numériques et des vidéos. Par exemple, Amazon SageMaker AI fournit des algorithmes de traitement d'image pour les CV.

dérive de configuration

Pour une charge de travail, une modification de configuration par rapport à l'état attendu. Cela peut entraîner une non-conformité de la charge de travail, et cela est généralement progressif et involontaire.

base de données de gestion des configurations (CMDB)

Référentiel qui stocke et gère les informations relatives à une base de données et à son environnement informatique, y compris les composants matériels et logiciels ainsi que leurs configurations. Vous utilisez généralement les données d'une CMDB lors de la phase de découverte et d'analyse du portefeuille de la migration.

pack de conformité

Ensemble de AWS Config règles et d'actions correctives que vous pouvez assembler pour personnaliser vos contrôles de conformité et de sécurité. Vous pouvez déployer un pack de conformité en tant qu'entité unique dans une région Compte AWS et, ou au sein d'une organisation, à l'aide d'un modèle YAML. Pour plus d'informations, consultez la section Packs de conformité dans la AWS Config documentation.

intégration continue et livraison continue (CI/CD)

Processus d'automatisation des étapes de source, de construction, de test, de préparation et de production du processus de publication du logiciel. CI/CD is commonly described as a pipeline. CI/CDpeut vous aider à automatiser les processus, à améliorer la productivité, à améliorer la qualité du code et à accélérer les livraisons. Pour plus d'informations, veuillez consulter Avantages de la livraison continue. CD peut également signifier déploiement continu. Pour plus d'informations, veuillez consulter Livraison continue et déploiement continu.

CV

Voir vision par ordinateur.

D

données au repos

Données stationnaires dans votre réseau, telles que les données stockées.

classification des données

Processus permettant d'identifier et de catégoriser les données de votre réseau en fonction de leur sévérité et de leur sensibilité. Il s'agit d'un élément essentiel de toute stratégie de gestion des risques de cybersécurité, car il vous aide à déterminer les contrôles de protection et de conservation appropriés pour les données. La classification des données est une composante du pilier de sécurité du AWS Well-Architected Framework. Pour plus d'informations, veuillez consulter Classification des données.

dérive des données

Une variation significative entre les données de production et les données utilisées pour entraîner un modèle ML, ou une modification significative des données d'entrée au fil du temps. La dérive des données peut réduire la qualité, la précision et l'équité globales des prédictions des modèles ML.

données en transit

Données qui circulent activement sur votre réseau, par exemple entre les ressources du réseau.

maillage de données

Un cadre architectural qui fournit une propriété des données distribuée et décentralisée avec une gestion et une gouvernance centralisées.

minimisation des données

Le principe de collecte et de traitement des seules données strictement nécessaires. La pratique de la minimisation des données AWS Cloud peut réduire les risques liés à la confidentialité, les coûts et l'empreinte carbone de vos analyses.

périmètre de données

Ensemble de garde-fous préventifs dans votre AWS environnement qui permettent de garantir que seules les identités fiables accèdent aux ressources fiables des réseaux attendus. Pour plus d'informations, voir Création d'un périmètre de données sur AWS.

prétraitement des données

Pour transformer les données brutes en un format facile à analyser par votre modèle de ML. Le prétraitement des données peut impliquer la suppression de certaines colonnes ou lignes et le traitement des valeurs manquantes, incohérentes ou en double.

provenance des données

Le processus de suivi de l'origine et de l'historique des données tout au long de leur cycle de vie, par exemple la manière dont les données ont été générées, transmises et stockées.

sujet des données

Personne dont les données sont collectées et traitées.

entrepôt des données

Un système de gestion des données qui prend en charge les informations commerciales, telles que les analyses. Les entrepôts de données contiennent généralement de grandes quantités de données historiques et sont généralement utilisés pour les requêtes et les analyses.

langage de définition de base de données (DDL)

Instructions ou commandes permettant de créer ou de modifier la structure des tables et des objets dans une base de données.

langage de manipulation de base de données (DML)

Instructions ou commandes permettant de modifier (insérer, mettre à jour et supprimer) des informations dans une base de données.

DDL

Voir langage de définition de base de données.

ensemble profond

Sert à combiner plusieurs modèles de deep learning à des fins de prédiction. Vous pouvez utiliser des ensembles profonds pour obtenir une prévision plus précise ou pour estimer l'incertitude des prédictions.

deep learning

Un sous-champ de ML qui utilise plusieurs couches de réseaux neuronaux artificiels pour identifier le mappage entre les données d'entrée et les variables cibles d'intérêt.

defense-in-depth

Approche de la sécurité de l'information dans laquelle une série de mécanismes et de contrôles de sécurité sont judicieusement répartis sur l'ensemble d'un réseau informatique afin de protéger la confidentialité, l'intégrité et la disponibilité du réseau et des données qu'il contient. Lorsque vous adoptez cette stratégie AWS, vous ajoutez plusieurs contrôles à différentes couches de

la AWS Organizations structure afin de sécuriser les ressources. Par exemple, une defense-in-depth approche peut combiner l'authentification multifactorielle, la segmentation du réseau et le chiffrement.

administrateur délégué

Dans AWS Organizations, un service compatible peut enregistrer un compte AWS membre pour administrer les comptes de l'organisation et gérer les autorisations pour ce service. Ce compte est appelé administrateur délégué pour ce service. Pour plus d'informations et une liste des services compatibles, veuillez consulter la rubrique Services qui fonctionnent avec AWS Organizations dans la documentation AWS Organizations .

déploiement

Processus de mise à disposition d'une application, de nouvelles fonctionnalités ou de corrections de code dans l'environnement cible. Le déploiement implique la mise en œuvre de modifications dans une base de code, puis la génération et l'exécution de cette base de code dans les environnements de l'application.

environnement de développement

Voir environnement.

contrôle de détection

Contrôle de sécurité conçu pour détecter, journaliser et alerter après la survenue d'un événement. Ces contrôles constituent une deuxième ligne de défense et vous alertent en cas d'événements de sécurité qui ont contourné les contrôles préventifs en place. Pour plus d'informations, veuillez consulter la rubrique Contrôles de détection dans Implementing security controls on AWS.

cartographie de la chaîne de valeur du développement (DVSM)

Processus utilisé pour identifier et hiérarchiser les contraintes qui nuisent à la rapidité et à la qualité du cycle de vie du développement logiciel. DVSM étend le processus de cartographie de la chaîne de valeur initialement conçu pour les pratiques de production allégée. Il met l'accent sur les étapes et les équipes nécessaires pour créer et transférer de la valeur tout au long du processus de développement logiciel.

jumeau numérique

Représentation virtuelle d'un système réel, tel qu'un bâtiment, une usine, un équipement industriel ou une ligne de production. Les jumeaux numériques prennent en charge la maintenance prédictive, la surveillance à distance et l'optimisation de la production.

tableau des dimensions

Dans un schéma en étoile, table plus petite contenant les attributs de données relatifs aux données quantitatives d'une table de faits. Les attributs des tables de dimensions sont généralement des champs de texte ou des nombres discrets qui se comportent comme du texte. Ces attributs sont couramment utilisés pour la contrainte des requêtes, le filtrage et l'étiquetage des ensembles de résultats.

catastrophe

Un événement qui empêche une charge de travail ou un système d'atteindre ses objectifs commerciaux sur son site de déploiement principal. Ces événements peuvent être des catastrophes naturelles, des défaillances techniques ou le résultat d'actions humaines, telles qu'une mauvaise configuration involontaire ou une attaque de logiciel malveillant.

reprise après sinistre (DR)

La stratégie et le processus que vous utilisez pour minimiser les temps d'arrêt et les pertes de données causés par un sinistre. Pour plus d'informations, consultez Disaster Recovery of Workloads on AWS : Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Voir langage de manipulation de base de données.

conception axée sur le domaine

Approche visant à développer un système logiciel complexe en connectant ses composants à des domaines évolutifs, ou objectifs métier essentiels, que sert chaque composant. Ce concept a été introduit par Eric Evans dans son ouvrage Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston : Addison-Wesley Professional, 2003). Pour plus d'informations sur l'utilisation du design piloté par domaine avec le modèle de figuier étrangleur, veuillez consulter Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

Voir reprise après sinistre.

détection de dérive

Suivi des écarts par rapport à une configuration de référence. Par exemple, vous pouvez l'utiliser AWS CloudFormation pour détecter la dérive des ressources du système ou AWS Control Tower

pour détecter les modifications de votre zone d'atterrissage susceptibles d'affecter le respect des exigences de gouvernance.

DVSM

Voir la cartographie de la chaîne de valeur du développement.

E

EDA

Voir analyse exploratoire des données.

EDI

Voir échange de données informatisé.

informatique de périphérie

Technologie qui augmente la puissance de calcul des appareils intelligents en périphérie d'un réseau IoT. Comparé au cloud computing, l'informatique de pointe peut réduire la latence des communications et améliorer le temps de réponse.

échange de données informatisé (EDI)

L'échange automatique de documents commerciaux entre les organisations. Pour plus d'informations, voir Qu'est-ce que l'échange de données informatisé ?

chiffrement

Processus informatique qui transforme des données en texte clair, lisibles par l'homme, en texte chiffré.

clé de chiffrement

Chaîne cryptographique de bits aléatoires générée par un algorithme cryptographique. La longueur des clés peut varier, et chaque clé est conçue pour être imprévisible et unique.

endianisme

Ordre selon lequel les octets sont stockés dans la mémoire de l'ordinateur. Les systèmes de poids fort stockent d'abord l'octet le plus significatif. Les systèmes de poids faible stockent d'abord l'octet le moins significatif.

point de terminaison

Voir point de terminaison de service.

service de point de terminaison

Service que vous pouvez héberger sur un cloud privé virtuel (VPC) pour le partager avec d'autres utilisateurs. Vous pouvez créer un service de point de terminaison avec AWS PrivateLink et accorder des autorisations à d'autres Comptes AWS ou à AWS Identity and Access Management (IAM) principaux. Ces comptes ou principaux peuvent se connecter à votre service de point de terminaison de manière privée en créant des points de terminaison d'un VPC d'interface. Pour plus d'informations, veuillez consulter Création d'un service de point de terminaison dans la documentation Amazon Virtual Private Cloud (Amazon VPC).

planification des ressources d'entreprise (ERP)

Système qui automatise et gère les principaux processus métier (tels que la comptabilité, le MES et la gestion de projet) pour une entreprise.

chiffrement d'enveloppe

Processus de chiffrement d'une clé de chiffrement à l'aide d'une autre clé de chiffrement. Pour plus d'informations, consultez la section Chiffrement des enveloppes dans la documentation AWS Key Management Service (AWS KMS).

environment

Instance d'une application en cours d'exécution. Les types d'environnement les plus courants dans le cloud computing sont les suivants :

- Environnement de développement : instance d'une application en cours d'exécution à laquelle seule l'équipe principale chargée de la maintenance de l'application peut accéder. Les environnements de développement sont utilisés pour tester les modifications avant de les promouvoir dans les environnements supérieurs. Ce type d'environnement est parfois appelé environnement de test.
- Environnements inférieurs : tous les environnements de développement d'une application, tels que ceux utilisés pour les générations et les tests initiaux.
- Environnement de production : instance d'une application en cours d'exécution à laquelle les utilisateurs finaux peuvent accéder. Dans un pipeline CI/CD, l'environnement de production est le dernier environnement de déploiement.
- Environnements supérieurs : tous les environnements accessibles aux utilisateurs autres que l'équipe de développement principale. Ils peuvent inclure un environnement de production, des

environnements de préproduction et des environnements pour les tests d'acceptation par les utilisateurs.

épopée

Dans les méthodologies agiles, catégories fonctionnelles qui aident à organiser et à prioriser votre travail. Les épopées fournissent une description détaillée des exigences et des tâches d'implémentation. Par exemple, les points forts de la AWS CAF en matière de sécurité incluent la gestion des identités et des accès, les contrôles de détection, la sécurité des infrastructures, la protection des données et la réponse aux incidents. Pour plus d'informations sur les épopées dans la stratégie de migration AWS , veuillez consulter le guide d'implémentation du programme.

ERP

Voir Planification des ressources d'entreprise.

analyse exploratoire des données (EDA)

Processus d'analyse d'un jeu de données pour comprendre ses principales caractéristiques. Vous collectez ou agrégez des données, puis vous effectuez des enquêtes initiales pour trouver des modèles, détecter des anomalies et vérifier les hypothèses. L'EDA est réalisée en calculant des statistiques récapitulatives et en créant des visualisations de données.

F

tableau des faits

La table centrale dans un schéma en étoile. Il stocke des données quantitatives sur les opérations commerciales. Généralement, une table de faits contient deux types de colonnes : celles qui contiennent des mesures et celles qui contiennent une clé étrangère pour une table de dimensions.

échouer rapidement

Une philosophie qui utilise des tests fréquents et progressifs pour réduire le cycle de vie du développement. C'est un élément essentiel d'une approche agile.

limite d'isolation des défauts

Dans le AWS Cloud, une limite telle qu'une zone de disponibilité Région AWS, un plan de contrôle ou un plan de données qui limite l'effet d'une panne et contribue à améliorer la résilience des

charges de travail. Pour plus d'informations, consultez la section Limites d'isolation des AWS pannes.

branche de fonctionnalités

Voir succursale.

fonctionnalités

Les données d'entrée que vous utilisez pour faire une prédiction. Par exemple, dans un contexte de fabrication, les fonctionnalités peuvent être des images capturées périodiquement à partir de la ligne de fabrication.

importance des fonctionnalités

Le niveau d'importance d'une fonctionnalité pour les prédictions d'un modèle. Il s'exprime généralement sous la forme d'un score numérique qui peut être calculé à l'aide de différentes techniques, telles que la méthode Shapley Additive Explanations (SHAP) et les gradients intégrés. Pour plus d'informations, voir Interprétabilité du modèle d'apprentissage automatique avec AWS.

transformation de fonctionnalité

Optimiser les données pour le processus de ML, notamment en enrichissant les données avec des sources supplémentaires, en mettant à l'échelle les valeurs ou en extrayant plusieurs ensembles d'informations à partir d'un seul champ de données. Cela permet au modèle de ML de tirer parti des données. Par exemple, si vous décomposez la date « 2021-05-27 00:15:37 » en « 2021 », « mai », « jeudi » et « 15 », vous pouvez aider l'algorithme d'apprentissage à apprendre des modèles nuancés associés à différents composants de données.

invitation en quelques coups

Fournir à un LLM un petit nombre d'exemples illustrant la tâche et le résultat souhaité avant de lui demander d'effectuer une tâche similaire. Cette technique est une application de l'apprentissage contextuel, dans le cadre de laquelle les modèles apprennent à partir d'exemples (prises de vue) intégrés dans des instructions. Les instructions en quelques étapes peuvent être efficaces pour les tâches qui nécessitent un formatage, un raisonnement ou des connaissances de domaine spécifiques. Voir également l'invite Zero-Shot.

FGAC

Découvrez le contrôle d'accès détaillé.

contrôle d'accès détaillé (FGAC)

Utilisation de plusieurs conditions pour autoriser ou refuser une demande d'accès.

migration instantanée (flash-cut)

Méthode de migration de base de données qui utilise la réplication continue des données par le biais de la capture des données de modification afin de migrer les données dans les plus brefs délais, au lieu d'utiliser une approche progressive. L'objectif est de réduire au maximum les temps d'arrêt.

FM

Voir le modèle de fondation.

modèle de fondation (FM)

Un vaste réseau neuronal d'apprentissage profond qui s'est entraîné sur d'énormes ensembles de données généralisées et non étiquetées. FMs sont capables d'effectuer une grande variété de tâches générales, telles que comprendre le langage, générer du texte et des images et converser en langage naturel. Pour plus d'informations, voir Que sont les modèles de base ?

G

IA générative

Sous-ensemble de modèles d'IA qui ont été entraînés sur de grandes quantités de données et qui peuvent utiliser une simple invite textuelle pour créer de nouveaux contenus et artefacts, tels que des images, des vidéos, du texte et du son. Pour plus d'informations, consultez Qu'est-ce que l'IA générative.

blocage géographique

Voir les restrictions géographiques.

restrictions géographiques (blocage géographique)

Sur Amazon CloudFront, option permettant d'empêcher les utilisateurs de certains pays d'accéder aux distributions de contenu. Vous pouvez utiliser une liste d'autorisation ou une liste de blocage pour spécifier les pays approuvés et interdits. Pour plus d'informations, consultez la section Restreindre la distribution géographique de votre contenu dans la CloudFront documentation.

Flux de travail Gitflow

Approche dans laquelle les environnements inférieurs et supérieurs utilisent différentes branches dans un référentiel de code source. Le flux de travail Gitflow est considéré comme existant, et le flux de travail basé sur les troncs est l'approche moderne préférée.

image dorée

Un instantané d'un système ou d'un logiciel utilisé comme modèle pour déployer de nouvelles instances de ce système ou logiciel. Par exemple, dans le secteur de la fabrication, une image dorée peut être utilisée pour fournir des logiciels sur plusieurs appareils et contribue à améliorer la vitesse, l'évolutivité et la productivité des opérations de fabrication des appareils.

stratégie inédite

L'absence d'infrastructures existantes dans un nouvel environnement. Lorsque vous adoptez une stratégie inédite pour une architecture système, vous pouvez sélectionner toutes les nouvelles technologies sans restriction de compatibilité avec l'infrastructure existante, également appelée brownfield. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

barrière de protection

Règle de haut niveau qui permet de régir les ressources, les politiques et la conformité au sein des unités organisationnelles (OUs). Les barrières de protection préventives appliquent des politiques pour garantir l'alignement sur les normes de conformité. Elles sont mises en œuvre à l'aide de politiques de contrôle des services et de limites des autorisations IAM. Les barrières de protection de détection détectent les violations des politiques et les problèmes de conformité, et génèrent des alertes pour y remédier. Ils sont implémentés à l'aide d'Amazon AWS Config AWS Security Hub GuardDuty AWS Trusted Advisor, d'Amazon Inspector et de AWS Lambda contrôles personnalisés.

H

HA

Découvrez la haute disponibilité.

migration de base de données hétérogène

Migration de votre base de données source vers une base de données cible qui utilise un moteur de base de données différent (par exemple, Oracle vers Amazon Aurora). La migration hétérogène fait généralement partie d'un effort de réarchitecture, et la conversion du schéma peut s'avérer une tâche complexe. AWS propose AWS SCT qui facilite les conversions de schémas.

haute disponibilité (HA)

Capacité d'une charge de travail à fonctionner en continu, sans intervention, en cas de difficultés ou de catastrophes. Les systèmes HA sont conçus pour basculer automatiquement, fournir constamment des performances de haute qualité et gérer différentes charges et défaillances avec un impact minimal sur les performances.

modernisation des historiens

Approche utilisée pour moderniser et mettre à niveau les systèmes de technologie opérationnelle (OT) afin de mieux répondre aux besoins de l'industrie manufacturière. Un historien est un type de base de données utilisé pour collecter et stocker des données provenant de diverses sources dans une usine.

données de rétention

Partie de données historiques étiquetées qui n'est pas divulguée dans un ensemble de données utilisé pour entraîner un modèle d'apprentissage automatique. Vous pouvez utiliser les données de blocage pour évaluer les performances du modèle en comparant les prévisions du modèle aux données de blocage.

migration de base de données homogène

Migration de votre base de données source vers une base de données cible qui partage le même moteur de base de données (par exemple, Microsoft SQL Server vers Amazon RDS for SQL Server). La migration homogène s'inscrit généralement dans le cadre d'un effort de réhébergement ou de replateforme. Vous pouvez utiliser les utilitaires de base de données natifs pour migrer le schéma.

données chaudes

Données fréquemment consultées, telles que les données en temps réel ou les données translationnelles récentes. Ces données nécessitent généralement un niveau ou une classe de stockage à hautes performances pour fournir des réponses rapides aux requêtes.

correctif

Solution d'urgence à un problème critique dans un environnement de production. En raison de son urgence, un correctif est généralement créé en dehors du flux de travail de DevOps publication habituel.

période de soins intensifs

Immédiatement après le basculement, période pendant laquelle une équipe de migration gère et surveille les applications migrées dans le cloud afin de résoudre les problèmes éventuels. En règle générale, cette période dure de 1 à 4 jours. À la fin de la période de soins intensifs, l'équipe de migration transfère généralement la responsabilité des applications à l'équipe des opérations cloud.

I

IaC

Considérez l'infrastructure comme un code.

politique basée sur l’identité

Politique attachée à un ou plusieurs principaux IAM qui définit leurs autorisations au sein de l' AWS Cloud environnement.

application inactive

Application dont l'utilisation moyenne du processeur et de la mémoire se situe entre 5 et 20 % sur une période de 90 jours. Dans un projet de migration, il est courant de retirer ces applications ou de les retenir sur site.

IIo T

Voir Internet industriel des objets.

infrastructure immuable

Modèle qui déploie une nouvelle infrastructure pour les charges de travail de production au lieu de mettre à jour, d'appliquer des correctifs ou de modifier l'infrastructure existante. Les infrastructures immuables sont intrinsèquement plus cohérentes, fiables et prévisibles que les infrastructures mutables. Pour plus d'informations, consultez les meilleures pratiques de déploiement à l'aide d'une infrastructure immuable dans le AWS Well-Architected Framework.

VPC entrant (d'entrée)

Dans une architecture AWS multi-comptes, un VPC qui accepte, inspecte et achemine les connexions réseau depuis l'extérieur d'une application. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes

et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

migration incrémentielle

Stratégie de basculement dans le cadre de laquelle vous migrez votre application par petites parties au lieu d'effectuer un basculement complet unique. Par exemple, il se peut que vous ne transfériez que quelques microservices ou utilisateurs vers le nouveau système dans un premier temps. Après avoir vérifié que tout fonctionne correctement, vous pouvez transférer progressivement des microservices ou des utilisateurs supplémentaires jusqu'à ce que vous puissiez mettre hors service votre système hérité. Cette stratégie réduit les risques associés aux migrations de grande ampleur.

Industry 4.0

Terme introduit par Klaus Schwab en 2016 pour désigner la modernisation des processus de fabrication grâce aux avancées en matière de connectivité, de données en temps réel, d'automatisation, d'analyse et d'IA/ML.

infrastructure

Ensemble des ressources et des actifs contenus dans l'environnement d'une application.

infrastructure en tant que code (IaC)

Processus de mise en service et de gestion de l'infrastructure d'une application via un ensemble de fichiers de configuration. IaC est conçue pour vous aider à centraliser la gestion de l'infrastructure, à normaliser les ressources et à mettre à l'échelle rapidement afin que les nouveaux environnements soient reproductibles, fiables et cohérents.

Internet industriel des objets (IIoT)

L'utilisation de capteurs et d'appareils connectés à Internet dans les secteurs industriels tels que la fabrication, l'énergie, l'automobile, les soins de santé, les sciences de la vie et l'agriculture. Pour plus d'informations, voir Élaboration d'une stratégie de transformation numérique de l'Internet des objets (IIoT) industriel.

VPC d'inspection

Dans une architecture AWS multi-comptes, un VPC centralisé qui gère les inspections du trafic réseau VPCs entre (identique ou Régions AWS différent), Internet et les réseaux locaux. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau

avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

Internet des objets (IoT)

Réseau d'objets physiques connectés dotés de capteurs ou de processeurs intégrés qui communiquent avec d'autres appareils et systèmes via Internet ou via un réseau de communication local. Pour plus d'informations, veuillez consulter la section Qu'est-ce que l'IoT ?.

interprétabilité

Caractéristique d'un modèle de machine learning qui décrit dans quelle mesure un être humain peut comprendre comment les prédictions du modèle dépendent de ses entrées. Pour plus d'informations, voir Interprétabilité du modèle d'apprentissage automatique avec AWS.

IoT

Voir Internet des objets.

Bibliothèque d'informations informatiques (ITIL)

Ensemble de bonnes pratiques pour proposer des services informatiques et les aligner sur les exigences métier. L'ITIL constitue la base de l'ITSM.

gestion des services informatiques (ITSM)

Activités associées à la conception, à la mise en œuvre, à la gestion et à la prise en charge de services informatiques d'une organisation. Pour plus d'informations sur l'intégration des opérations cloud aux outils ITSM, veuillez consulter le guide d'intégration des opérations.

ITIL

Consultez la bibliothèque d'informations informatiques.

ITSM

Voir Gestion des services informatiques.

L

contrôle d'accès basé sur des étiquettes (LBAC)

Une implémentation du contrôle d'accès obligatoire (MAC) dans laquelle une valeur d'étiquette de sécurité est explicitement attribuée aux utilisateurs et aux données elles-mêmes. L'intersection

entre l'étiquette de sécurité utilisateur et l'étiquette de sécurité des données détermine les lignes et les colonnes visibles par l'utilisateur.

zone de destination

Une zone d'atterrissage est un AWS environnement multi-comptes bien conçu, évolutif et sécurisé. Il s'agit d'un point de départ à partir duquel vos entreprises peuvent rapidement lancer et déployer des charges de travail et des applications en toute confiance dans leur environnement de sécurité et d'infrastructure. Pour plus d'informations sur les zones de destination, veuillez consulter Setting up a secure and scalable multi-account AWS environment.

grand modèle de langage (LLM)

Un modèle d'intelligence artificielle basé sur le deep learning qui est préentraîné sur une grande quantité de données. Un LLM peut effectuer plusieurs tâches, telles que répondre à des questions, résumer des documents, traduire du texte dans d'autres langues et compléter des phrases. Pour plus d'informations, voir Que sont LLMs.

migration de grande envergure

Migration de 300 serveurs ou plus.

LBAC

Voir contrôle d'accès basé sur des étiquettes.

principe de moindre privilège

Bonne pratique de sécurité qui consiste à accorder les autorisations minimales nécessaires à l'exécution d'une tâche. Pour plus d'informations, veuillez consulter la rubrique Accorder les autorisations de moindre privilège dans la documentation IAM.

lift and shift

Voir 7 Rs.

système de poids faible

Système qui stocke d'abord l'octet le moins significatif. Voir aussi endianité.

LLM

Voir le grand modèle de langage.

environnements inférieurs

Voir environnement.

M

machine learning (ML)

Type d'intelligence artificielle qui utilise des algorithmes et des techniques pour la reconnaissance et l'apprentissage de modèles. Le ML analyse et apprend à partir de données enregistrées, telles que les données de l'Internet des objets (IoT), pour générer un modèle statistique basé sur des modèles. Pour plus d'informations, veuillez consulter Machine Learning.

branche principale

Voir succursale.

malware

Logiciel conçu pour compromettre la sécurité ou la confidentialité de l'ordinateur. Les logiciels malveillants peuvent perturber les systèmes informatiques, divulguer des informations sensibles ou obtenir un accès non autorisé. Parmi les malwares, on peut citer les virus, les vers, les rançongiciels, les chevaux de Troie, les logiciels espions et les enregistreurs de frappe.

services gérés

Services AWS pour lequel AWS fonctionnent la couche d'infrastructure, le système d'exploitation et les plateformes, et vous accédez aux points de terminaison pour stocker et récupérer des données. Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB sont des exemples de services gérés. Ils sont également connus sous le nom de services abstraits.

système d'exécution de la fabrication (MES)

Un système logiciel pour le suivi, la surveillance, la documentation et le contrôle des processus de production qui convertissent les matières premières en produits finis dans l'atelier.

MAP

Voir Migration Acceleration Program.

mécanisme

Processus complet au cours duquel vous créez un outil, favorisez son adoption, puis inspectez les résultats afin de procéder aux ajustements nécessaires. Un mécanisme est un cycle qui se renforce et s'améliore au fur et à mesure de son fonctionnement. Pour plus d'informations, voir Création de mécanismes dans le cadre AWS Well-Architected.

compte membre

Tous, à l' Comptes AWS exception du compte de gestion, qui font partie d'une organisation dans AWS Organizations. Un compte ne peut être membre que d'une seule organisation à la fois.

MAILLES

Voir le système d'exécution de la fabrication.

Transport télémétrique en file d'attente de messages (MQTT)

Protocole de communication léger machine-to-machine (M2M), basé sur le modèle de publication/d'abonnement, pour les appareils IoT aux ressources limitées.

microservice

Un petit service indépendant qui communique via un réseau bien défini APIs et qui est généralement détenu par de petites équipes autonomes. Par exemple, un système d'assurance peut inclure des microservices qui mappent à des capacités métier, telles que les ventes ou le marketing, ou à des sous-domaines, tels que les achats, les réclamations ou l'analytique. Les avantages des microservices incluent l'agilité, la flexibilité de la mise à l'échelle, la facilité de déploiement, la réutilisation du code et la résilience. Pour plus d'informations, consultez la section Intégration de microservices à l'aide de services AWS sans serveur.

architecture de microservices

Approche de création d'une application avec des composants indépendants qui exécutent chaque processus d'application en tant que microservice. Ces microservices communiquent via une interface bien définie en utilisant Lightweight. APIs Chaque microservice de cette architecture peut être mis à jour, déployé et mis à l'échelle pour répondre à la demande de fonctions spécifiques d'une application. Pour plus d'informations, consultez la section Implémentation de microservices sur AWS.

Programme d'accélération des migrations (MAP)

Un AWS programme qui fournit un support de conseil, des formations et des services pour aider les entreprises à établir une base opérationnelle solide pour passer au cloud, et pour aider à compenser le coût initial des migrations. MAP inclut une méthodologie de migration pour exécuter les migrations héritées de manière méthodique, ainsi qu'un ensemble d'outils pour automatiser et accélérer les scénarios de migration courants.

migration à grande échelle

Processus consistant à transférer la majeure partie du portefeuille d'applications vers le cloud par vagues, un plus grand nombre d'applications étant déplacées plus rapidement à chaque vague. Cette phase utilise les bonnes pratiques et les enseignements tirés des phases précédentes pour implémenter une usine de migration d'équipes, d'outils et de processus en vue de rationaliser la migration des charges de travail grâce à l'automatisation et à la livraison agile. Il s'agit de la troisième phase de la stratégie de migration AWS.

usine de migration

Équipes interfonctionnelles qui rationalisent la migration des charges de travail grâce à des approches automatisées et agiles. Les équipes de Migration Factory comprennent généralement des responsables des opérations, des analystes commerciaux et des propriétaires, des ingénieurs de migration, des développeurs et DevOps des professionnels travaillant dans le cadre de sprints. Entre 20 et 50 % du portefeuille d'applications d'entreprise est constitué de modèles répétés qui peuvent être optimisés par une approche d'usine. Pour plus d'informations, veuillez consulter la rubrique discussion of migration factories et le guide Cloud Migration Factory dans cet ensemble de contenus.

métadonnées de migration

Informations relatives à l'application et au serveur nécessaires pour finaliser la migration. Chaque modèle de migration nécessite un ensemble de métadonnées de migration différent. Les exemples de métadonnées de migration incluent le sous-réseau cible, le groupe de sécurité et le AWS compte.

modèle de migration

Tâche de migration reproductible qui détaille la stratégie de migration, la destination de la migration et l'application ou le service de migration utilisé. Exemple : réorganisez la migration vers Amazon EC2 avec le service de migration AWS d'applications.

Évaluation du portefeuille de migration (MPA)

Outil en ligne qui fournit des informations pour valider l'analyse de rentabilisation en faveur de la migration vers le. AWS Cloud La MPA propose une évaluation détaillée du portefeuille (dimensionnement approprié des serveurs, tarification, comparaison du coût total de possession, analyse des coûts de migration), ainsi que la planification de la migration (analyse et collecte des données d'applications, regroupement des applications, priorisation des migrations et planification des vagues). L'outil MPA (connexion requise) est disponible gratuitement pour tous les AWS consultants et consultants APN Partner.

Évaluation de la préparation à la migration (MRA)

Processus qui consiste à obtenir des informations sur l'état de préparation d'une organisation au cloud, à identifier les forces et les faiblesses et à élaborer un plan d'action pour combler les lacunes identifiées, à l'aide du AWS CAF. Pour plus d'informations, veuillez consulter le guide de préparation à la migration. La MRA est la première phase de la stratégie de migration AWS.

stratégie de migration

L'approche utilisée pour migrer une charge de travail vers le AWS Cloud. Pour plus d'informations, reportez-vous aux 7 R de ce glossaire et à Mobiliser votre organisation pour accélérer les migrations à grande échelle.

ML

Voir apprentissage automatique.

modernisation

Transformation d'une application obsolète (héritée ou monolithique) et de son infrastructure en un système agile, élastique et hautement disponible dans le cloud afin de réduire les coûts, de gagner en efficacité et de tirer parti des innovations. Pour plus d'informations, consultez la section Stratégie de modernisation des applications dans le AWS Cloud.

évaluation de la préparation à la modernisation

Évaluation qui permet de déterminer si les applications d'une organisation sont prêtes à être modernisées, d'identifier les avantages, les risques et les dépendances, et qui détermine dans quelle mesure l'organisation peut prendre en charge l'état futur de ces applications. Le résultat de l'évaluation est un plan de l'architecture cible, une feuille de route détaillant les phases de développement et les étapes du processus de modernisation, ainsi qu'un plan d'action pour combler les lacunes identifiées. Pour plus d'informations, consultez la section Évaluation de l'état de préparation à la modernisation des applications dans le AWS Cloud.

applications monolithiques (monolithes)

Applications qui s'exécutent en tant que service unique avec des processus étroitement couplés. Les applications monolithiques ont plusieurs inconvénients. Si une fonctionnalité de l'application connaît un pic de demande, l'architecture entière doit être mise à l'échelle. L'ajout ou l'amélioration des fonctionnalités d'une application monolithique devient également plus complexe lorsque la base de code s'élargit. Pour résoudre ces problèmes, vous pouvez utiliser une architecture de microservices. Pour plus d'informations, veuillez consulter Decomposing monoliths into microservices.

MPA

Voir Évaluation du portefeuille de migration.

MQTT

Voir Message Queuing Telemetry Transport.

classification multi-classes

Processus qui permet de générer des prédictions pour plusieurs classes (prédiction d'un résultat parmi plus de deux). Par exemple, un modèle de ML peut demander « Ce produit est-il un livre, une voiture ou un téléphone ? » ou « Quelle catégorie de produits intéresse le plus ce client ? ».

infrastructure mutable

Modèle qui met à jour et modifie l'infrastructure existante pour les charges de travail de production. Pour améliorer la cohérence, la fiabilité et la prévisibilité, le AWS Well-Architected Framework recommande l'utilisation d'une infrastructure immuable comme meilleure pratique.

O

OAC

Voir Contrôle d'accès à l'origine.

OAI

Voir l'identité d'accès à l'origine.

OCM

Voir gestion du changement organisationnel.

migration hors ligne

Méthode de migration dans laquelle la charge de travail source est supprimée au cours du processus de migration. Cette méthode implique un temps d'arrêt prolongé et est généralement utilisée pour de petites charges de travail non critiques.

OI

Voir Intégration des opérations.

OLA

Voir l'accord au niveau opérationnel.

migration en ligne

Méthode de migration dans laquelle la charge de travail source est copiée sur le système cible sans être mise hors ligne. Les applications connectées à la charge de travail peuvent continuer à fonctionner pendant la migration. Cette méthode implique un temps d'arrêt nul ou minimal et est généralement utilisée pour les charges de travail de production critiques.

OPC-UA

Voir Open Process Communications - Architecture unifiée.

Communications par processus ouvert - Architecture unifiée (OPC-UA)

Un protocole de communication machine-to-machine (M2M) pour l'automatisation industrielle. L'OPC-UA fournit une norme d'interopérabilité avec des schémas de cryptage, d'authentification et d'autorisation des données.

accord au niveau opérationnel (OLA)

Accord qui précise ce que les groupes informatiques fonctionnels s'engagent à fournir les uns aux autres, afin de prendre en charge un contrat de niveau de service (SLA).

examen de l'état de préparation opérationnelle (ORR)

Une liste de questions et de bonnes pratiques associées qui vous aident à comprendre, évaluer, prévenir ou réduire l'ampleur des incidents et des défaillances possibles. Pour plus d'informations, voir Operational Readiness Reviews (ORR) dans le AWS Well-Architected Framework.

technologie opérationnelle (OT)

Systèmes matériels et logiciels qui fonctionnent avec l'environnement physique pour contrôler les opérations, les équipements et les infrastructures industriels. Dans le secteur manufacturier, l'intégration des systèmes OT et des technologies de l'information (IT) est au cœur des transformations de l'industrie 4.0.

intégration des opérations (OI)

Processus de modernisation des opérations dans le cloud, qui implique la planification de la préparation, l'automatisation et l'intégration. Pour en savoir plus, veuillez consulter le guide d'intégration des opérations.

journal de suivi d'organisation

Un parcours créé par AWS CloudTrail qui enregistre tous les événements pour tous les membres Comptes AWS d'une organisation dans AWS Organizations. Ce journal de suivi est créé dans

chaque Compte AWS qui fait partie de l'organisation et suit l'activité de chaque compte. Pour plus d'informations, consultez la section Création d'un suivi pour une organisation dans la CloudTrail documentation.

gestion du changement organisationnel (OCM)

Cadre pour gérer les transformations métier majeures et perturbatrices du point de vue des personnes, de la culture et du leadership. L'OCM aide les organisations à se préparer et à effectuer la transition vers de nouveaux systèmes et de nouvelles politiques en accélérant l'adoption des changements, en abordant les problèmes de transition et en favorisant des changements culturels et organisationnels. Dans la stratégie de AWS migration, ce cadre est appelé accélération du personnel, en raison de la rapidité du changement requise dans les projets d'adoption du cloud. Pour plus d'informations, veuillez consulter le guide OCM.

contrôle d'accès d'origine (OAC)

Dans CloudFront, une option améliorée pour restreindre l'accès afin de sécuriser votre contenu Amazon Simple Storage Service (Amazon S3). L'OAC prend en charge tous les compartiments S3 dans leur ensemble Régions AWS, le chiffrement côté serveur avec AWS KMS (SSE-KMS) et les requêtes dynamiques `PUT` adressées au compartiment S3. `DELETE`

identité d'accès d'origine (OAI)

Dans CloudFront, une option permettant de restreindre l'accès afin de sécuriser votre contenu Amazon S3. Lorsque vous utilisez OAI, il CloudFront crée un principal auprès duquel Amazon S3 peut s'authentifier. Les principaux authentifiés ne peuvent accéder au contenu d'un compartiment S3 que par le biais d'une distribution spécifique CloudFront . Voir également OAC, qui fournit un contrôle d'accès plus précis et amélioré.

ORR

Voir l'examen de l'état de préparation opérationnelle.

DE

Voir technologie opérationnelle.

VPC sortant (de sortie)

Dans une architecture AWS multi-comptes, un VPC qui gère les connexions réseau initiées depuis une application. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

P

limite des autorisations

Politique de gestion IAM attachée aux principaux IAM pour définir les autorisations maximales que peut avoir l'utilisateur ou le rôle. Pour plus d'informations, veuillez consulter la rubrique Limites des autorisations dans la documentation IAM.

informations personnelles identifiables (PII)

Informations qui, lorsqu'elles sont consultées directement ou associées à d'autres données connexes, peuvent être utilisées pour déduire raisonnablement l'identité d'une personne. Les exemples d'informations personnelles incluent les noms, les adresses et les informations de contact.

PII

Voir les informations personnelles identifiables.

manuel stratégique

Ensemble d'étapes prédéfinies qui capturent le travail associé aux migrations, comme la fourniture de fonctions d'opérations de base dans le cloud. Un manuel stratégique peut revêtir la forme de scripts, de runbooks automatisés ou d'un résumé des processus ou des étapes nécessaires au fonctionnement de votre environnement modernisé.

PLC

Voir contrôleur logique programmable.

PLM

Consultez la section Gestion du cycle de vie des produits.

politique

Objet capable de définir les autorisations (voir la politique basée sur l'identité), de spécifier les conditions d'accès (voir la politique basée sur les ressources) ou de définir les autorisations maximales pour tous les comptes d'une organisation dans AWS Organizations (voir la politique de contrôle des services).

persistance polyglotte

Choix indépendant de la technologie de stockage de données d'un microservice en fonction des modèles d'accès aux données et d'autres exigences. Si vos microservices utilisent la même

technologie de stockage de données, ils peuvent rencontrer des difficultés d'implémentation ou présenter des performances médiocres. Les microservices sont plus faciles à mettre en œuvre, atteignent de meilleures performances, ainsi qu'une meilleure capacité de mise à l'échelle s'ils utilisent l'entrepôt de données le mieux adapté à leurs besoins. Pour plus d'informations, veuillez consulter Enabling data persistence in microservices.

évaluation du portefeuille

Processus de découverte, d'analyse et de priorisation du portefeuille d'applications afin de planifier la migration. Pour plus d'informations, veuillez consulter Evaluating migration readiness.

predicate

Une condition de requête qui renvoie `true` ou`false`, généralement située dans une `WHERE` clause.

prédicat pushdown

Technique d'optimisation des requêtes de base de données qui filtre les données de la requête avant le transfert. Cela réduit la quantité de données qui doivent être extraites et traitées à partir de la base de données relationnelle et améliore les performances des requêtes.

contrôle préventif

Contrôle de sécurité conçu pour empêcher qu'un événement ne se produise. Ces contrôles constituent une première ligne de défense pour empêcher tout accès non autorisé ou toute modification indésirable de votre réseau. Pour plus d'informations, veuillez consulter Preventative controls dans Implementing security controls on AWS.

principal

Entité capable d'effectuer AWS des actions et d'accéder à des ressources. Cette entité est généralement un utilisateur root pour un Compte AWS rôle IAM ou un utilisateur. Pour plus d'informations, veuillez consulter la rubrique Principal dans Termes et concepts relatifs aux rôles, dans la documentation IAM.

confidentialité dès la conception

Une approche d'ingénierie système qui prend en compte la confidentialité tout au long du processus de développement.

zones hébergées privées

Conteneur contenant des informations sur la manière dont vous souhaitez qu'Amazon Route 53 réponde aux requêtes DNS pour un domaine et ses sous-domaines au sein d'un ou de plusieurs

VPCs domaines. Pour plus d'informations, veuillez consulter Working with private hosted zones dans la documentation Route 53.

contrôle proactif

Contrôle de sécurité conçu pour empêcher le déploiement de ressources non conformes. Ces contrôles analysent les ressources avant qu'elles ne soient provisionnées. Si la ressource n'est pas conforme au contrôle, elle n'est pas provisionnée. Pour plus d'informations, consultez le guide de référence sur les contrôles dans la AWS Control Tower documentation et consultez la section Contrôles proactifs dans Implémentation des contrôles de sécurité sur AWS.

gestion du cycle de vie des produits (PLM)

Gestion des données et des processus d'un produit tout au long de son cycle de vie, depuis la conception, le développement et le lancement, en passant par la croissance et la maturité, jusqu'au déclin et au retrait.

environnement de production

Voir environnement.

contrôleur logique programmable (PLC)

Dans le secteur manufacturier, un ordinateur hautement fiable et adaptable qui surveille les machines et automatise les processus de fabrication.

chaînage rapide

Utiliser le résultat d'une invite LLM comme entrée pour l'invite suivante afin de générer de meilleures réponses. Cette technique est utilisée pour décomposer une tâche complexe en sous-tâches ou pour affiner ou développer de manière itérative une réponse préliminaire. Cela permet d'améliorer la précision et la pertinence des réponses d'un modèle et permet d'obtenir des résultats plus précis et personnalisés.

pseudonymisation

Processus de remplacement des identifiants personnels dans un ensemble de données par des valeurs fictives. La pseudonymisation peut contribuer à protéger la vie privée. Les données pseudonymisées sont toujours considérées comme des données personnelles.

publish/subscribe (pub/sub)

Modèle qui permet des communications asynchrones entre les microservices afin d'améliorer l'évolutivité et la réactivité. Par exemple, dans un MES basé sur des microservices, un microservice peut publier des messages d'événements sur un canal auquel d'autres microservices

peuvent s'abonner. Le système peut ajouter de nouveaux microservices sans modifier le service de publication.

Q

plan de requête

Série d'étapes, telles que des instructions, utilisées pour accéder aux données d'un système de base de données relationnelle SQL.

régression du plan de requêtes

Le cas où un optimiseur de service de base de données choisit un plan moins optimal qu'avant une modification donnée de l'environnement de base de données. Cela peut être dû à des changements en termes de statistiques, de contraintes, de paramètres d'environnement, de liaisons de paramètres de requêtes et de mises à jour du moteur de base de données.

R

Matrice RACI

Voir responsable, responsable, consulté, informé (RACI).

CHIFFON

Voir Retrieval Augmented Generation.

rançongiciel

Logiciel malveillant conçu pour bloquer l'accès à un système informatique ou à des données jusqu'à ce qu'un paiement soit effectué.

Matrice RASCI

Voir responsable, responsable, consulté, informé (RACI).

RCAC

Voir contrôle d'accès aux lignes et aux colonnes.

réplica en lecture

Copie d'une base de données utilisée en lecture seule. Vous pouvez acheminer les requêtes vers le réplica de lecture pour réduire la charge sur votre base de données principale.

réarchitecte

Voir 7 Rs.

objectif de point de récupération (RPO)

Durée maximale acceptable depuis le dernier point de récupération des données. Il détermine ce qui est considéré comme étant une perte de données acceptable entre le dernier point de reprise et l'interruption du service.

objectif de temps de récupération (RTO)

Le délai maximum acceptable entre l'interruption du service et le rétablissement du service.

refactoriser

Voir 7 Rs.

Région

Un ensemble de AWS ressources dans une zone géographique. Chacun Région AWS est isolé et indépendant des autres pour garantir tolérance aux pannes, stabilité et résilience. Pour plus d'informations, voir Spécifier ce que Régions AWS votre compte peut utiliser.

régression

Technique de ML qui prédit une valeur numérique. Par exemple, pour résoudre le problème « Quel sera le prix de vente de cette maison ? », un modèle de ML pourrait utiliser un modèle de régression linéaire pour prédire le prix de vente d'une maison sur la base de faits connus à son sujet (par exemple, la superficie en mètres carrés).

réhéberger

Voir 7 Rs.

version

Dans un processus de déploiement, action visant à promouvoir les modifications apportées à un environnement de production.

déplacer

Voir 7 Rs.

replateforme

Voir 7 Rs.

rachat

Voir 7 Rs.

résilience

La capacité d'une application à résister aux perturbations ou à s'en remettre. La haute disponibilité et la reprise après sinistre sont des considérations courantes lors de la planification de la résilience dans le AWS Cloud. Pour plus d'informations, consultez la section AWS Cloud Résilience.

politique basée sur les ressources

Politique attachée à une ressource, comme un compartiment Amazon S3, un point de terminaison ou une clé de chiffrement. Ce type de politique précise les principaux auxquels l'accès est autorisé, les actions prises en charge et toutes les autres conditions qui doivent être remplies.

matrice responsable, redevable, consulté et informé (RACI)

Une matrice qui définit les rôles et les responsabilités de toutes les parties impliquées dans les activités de migration et les opérations cloud. Le nom de la matrice est dérivé des types de responsabilité définis dans la matrice : responsable (R), responsable (A), consulté (C) et informé (I). Le type de support (S) est facultatif. Si vous incluez le support, la matrice est appelée matrice RASCI, et si vous l'excluez, elle est appelée matrice RACI.

contrôle réactif

Contrôle de sécurité conçu pour permettre de remédier aux événements indésirables ou aux écarts par rapport à votre référence de sécurité. Pour plus d'informations, veuillez consulter la rubrique Responsive controls dans Implementing security controls on AWS.

retain

Voir 7 Rs.

se retirer

Voir 7 Rs.

Génération augmentée de récupération (RAG)

Technologie d'IA générative dans laquelle un LLM fait référence à une source de données faisant autorité qui se trouve en dehors de ses sources de données de formation avant de générer une

réponse. Par exemple, un modèle RAG peut effectuer une recherche sémantique dans la base de connaissances ou dans les données personnalisées d'une organisation. Pour plus d'informations, voir Qu'est-ce que RAG ?

rotation

Processus de mise à jour périodique d'un secret pour empêcher un attaquant d'accéder aux informations d'identification.

contrôle d'accès aux lignes et aux colonnes (RCAC)

Utilisation d'expressions SQL simples et flexibles dotées de règles d'accès définies. Le RCAC comprend des autorisations de ligne et des masques de colonnes.

RPO

Voir l'objectif du point de récupération.

RTO

Voir l'objectif relatif au temps de rétablissement.

runbook

Ensemble de procédures manuelles ou automatisées nécessaires à l'exécution d'une tâche spécifique. Elles visent généralement à rationaliser les opérations ou les procédures répétitives présentant des taux d'erreur élevés.

S

SAML 2.0

Un standard ouvert utilisé par de nombreux fournisseurs d'identité (IdPs). Cette fonctionnalité permet l'authentification unique fédérée (SSO), afin que les utilisateurs puissent se connecter AWS Management Console ou appeler les opérations de l' AWS API sans que vous ayez à créer un utilisateur dans IAM pour tous les membres de votre organisation. Pour plus d'informations sur la fédération SAML 2.0, veuillez consulter À propos de la fédération SAML 2.0 dans la documentation IAM.

SCADA

Voir Contrôle de supervision et acquisition de données.

SCP

Voir la politique de contrôle des services.

secret

Dans AWS Secrets Manager des informations confidentielles ou restreintes, telles qu'un mot de passe ou des informations d'identification utilisateur, que vous stockez sous forme cryptée. Il comprend la valeur secrète et ses métadonnées. La valeur secrète peut être binaire, une chaîne unique ou plusieurs chaînes. Pour plus d'informations, voir Que contient le secret d'un Secrets Manager ? dans la documentation de Secrets Manager.

sécurité dès la conception

Une approche d'ingénierie système qui prend en compte la sécurité tout au long du processus de développement.

contrôle de sécurité

Barrière de protection technique ou administrative qui empêche, détecte ou réduit la capacité d'un assaillant d'exploiter une vulnérabilité de sécurité. Il existe quatre principaux types de contrôles de sécurité : préventifs, détectifs, réactifs et proactifs.

renforcement de la sécurité

Processus qui consiste à réduire la surface d'attaque pour la rendre plus résistante aux attaques. Cela peut inclure des actions telles que la suppression de ressources qui ne sont plus requises, la mise en œuvre des bonnes pratiques de sécurité consistant à accorder le moindre privilège ou la désactivation de fonctionnalités inutiles dans les fichiers de configuration.

système de gestion des informations et des événements de sécurité (SIEM)

Outils et services qui associent les systèmes de gestion des informations de sécurité (SIM) et de gestion des événements de sécurité (SEM). Un système SIEM collecte, surveille et analyse les données provenant de serveurs, de réseaux, d'appareils et d'autres sources afin de détecter les menaces et les failles de sécurité, mais aussi de générer des alertes.

automatisation des réponses de sécurité

Action prédéfinie et programmée conçue pour répondre automatiquement à un événement de sécurité ou y remédier. Ces automatisations servent de contrôles de sécurité détectifs ou réactifs qui vous aident à mettre en œuvre les meilleures pratiques AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une EC2 instance Amazon ou la rotation des informations d'identification.

chiffrement côté serveur

Chiffrement des données à destination, par celui Service AWS qui les reçoit.

Politique de contrôle des services (SCP)

Politique qui fournit un contrôle centralisé des autorisations pour tous les comptes d'une organisation dans AWS Organizations. SCPs définissez des garde-fous ou des limites aux actions qu'un administrateur peut déléguer à des utilisateurs ou à des rôles. Vous pouvez les utiliser SCPs comme listes d'autorisation ou de refus pour spécifier les services ou les actions autorisés ou interdits. Pour plus d'informations, consultez la section Politiques de contrôle des services dans la AWS Organizations documentation.

point de terminaison du service

URL du point d'entrée pour un Service AWS. Pour vous connecter par programmation au service cible, vous pouvez utiliser un point de terminaison. Pour plus d'informations, veuillez consulter la rubrique Service AWS endpoints dans Références générales AWS.

contrat de niveau de service (SLA)

Accord qui précise ce qu'une équipe informatique promet de fournir à ses clients, comme le temps de disponibilité et les performances des services.

indicateur de niveau de service (SLI)

Mesure d'un aspect des performances d'un service, tel que son taux d'erreur, sa disponibilité ou son débit.

objectif de niveau de service (SLO)

Mesure cible qui représente l'état d'un service, tel que mesuré par un indicateur de niveau de service.

modèle de responsabilité partagée

Un modèle décrivant la responsabilité que vous partagez en matière AWS de sécurité et de conformité dans le cloud. AWS est responsable de la sécurité du cloud, alors que vous êtes responsable de la sécurité dans le cloud. Pour de plus amples informations, veuillez consulter Modèle de responsabilité partagée.

SIEM

Consultez les informations de sécurité et le système de gestion des événements.

point de défaillance unique (SPOF)

Défaillance d'un seul composant critique d'une application susceptible de perturber le système.

SLA

Voir le contrat de niveau de service.

SLI

Voir l'indicateur de niveau de service.

SLO

Voir l'objectif de niveau de service.

split-and-seed modèle

Modèle permettant de mettre à l'échelle et d'accélérer les projets de modernisation. Au fur et à mesure que les nouvelles fonctionnalités et les nouvelles versions de produits sont définies, l'équipe principale se divise pour créer des équipes de produit. Cela permet de mettre à l'échelle les capacités et les services de votre organisation, d'améliorer la productivité des développeurs et de favoriser une innovation rapide. Pour plus d'informations, consultez la section Approche progressive de la modernisation des applications dans le. AWS Cloud

SPOF

Voir point de défaillance unique.

schéma en étoile

Structure organisationnelle de base de données qui utilise une grande table de faits pour stocker les données transactionnelles ou mesurées et utilise une ou plusieurs tables dimensionnelles plus petites pour stocker les attributs des données. Cette structure est conçue pour être utilisée dans un entrepôt de données ou à des fins de business intelligence.

modèle de figuier étrangleur

Approche de modernisation des systèmes monolithiques en réécrivant et en remplaçant progressivement les fonctionnalités du système jusqu'à ce que le système hérité puisse être mis hors service. Ce modèle utilise l'analogie d'un figuier de vigne qui se développe dans un arbre existant et qui finit par supplanter son hôte. Le schéma a été présenté par Martin Fowler comme un moyen de gérer les risques lors de la réécriture de systèmes monolithiques. Pour obtenir un

exemple d'application de ce modèle, veuillez consulter Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

sous-réseau

Plage d'adresses IP dans votre VPC. Un sous-réseau doit se trouver dans une seule zone de disponibilité.

contrôle de supervision et acquisition de données (SCADA)

Dans le secteur manufacturier, un système qui utilise du matériel et des logiciels pour surveiller les actifs physiques et les opérations de production.

chiffrement symétrique

Algorithme de chiffrement qui utilise la même clé pour chiffrer et déchiffrer les données.

tests synthétiques

Tester un système de manière à simuler les interactions des utilisateurs afin de détecter les problèmes potentiels ou de surveiller les performances. Vous pouvez utiliser Amazon CloudWatch Synthetics pour créer ces tests.

invite du système

Technique permettant de fournir un contexte, des instructions ou des directives à un LLM afin d'orienter son comportement. Les instructions du système aident à définir le contexte et à établir des règles pour les interactions avec les utilisateurs.

T

balises

Des paires clé-valeur qui agissent comme des métadonnées pour organiser vos AWS ressources. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources. Pour plus d'informations, veuillez consulter la rubrique Balisage de vos AWS ressources.

variable cible

La valeur que vous essayez de prédire dans le cadre du ML supervisé. Elle est également qualifiée de variable de résultat. Par exemple, dans un environnement de fabrication, la variable cible peut être un défaut du produit.

liste de tâches

Outil utilisé pour suivre les progrès dans un runbook. Liste de tâches qui contient une vue d'ensemble du runbook et une liste des tâches générales à effectuer. Pour chaque tâche générale, elle inclut le temps estimé nécessaire, le propriétaire et l'avancement.

environnement de test

Voir environnement.

entraînement

Pour fournir des données à partir desquelles votre modèle de ML peut apprendre. Les données d'entraînement doivent contenir la bonne réponse. L'algorithme d'apprentissage identifie des modèles dans les données d'entraînement, qui mettent en correspondance les attributs des données d'entrée avec la cible (la réponse que vous souhaitez prédire). Il fournit un modèle de ML qui capture ces modèles. Vous pouvez alors utiliser le modèle de ML pour obtenir des prédictions sur de nouvelles données pour lesquelles vous ne connaissez pas la cible.

passerelle de transit

Un hub de transit réseau que vous pouvez utiliser pour interconnecter vos réseaux VPCs et ceux sur site. Pour plus d'informations, voir Qu'est-ce qu'une passerelle de transit dans la AWS Transit Gateway documentation.

flux de travail basé sur jonction

Approche selon laquelle les développeurs génèrent et testent des fonctionnalités localement dans une branche de fonctionnalités, puis fusionnent ces modifications dans la branche principale. La branche principale est ensuite intégrée aux environnements de développement, de préproduction et de production, de manière séquentielle.

accès sécurisé

Accorder des autorisations à un service que vous spécifiez pour effectuer des tâches au sein de votre organisation AWS Organizations et dans ses comptes en votre nom. Le service de confiance crée un rôle lié au service dans chaque compte, lorsque ce rôle est nécessaire, pour effectuer des tâches de gestion à votre place. Pour plus d'informations, consultez la section Utilisation AWS Organizations avec d'autres AWS services dans la AWS Organizations documentation.

réglage

Pour modifier certains aspects de votre processus d'entraînement afin d'améliorer la précision du modèle de ML. Par exemple, vous pouvez entraîner le modèle de ML en générant un ensemble d'étiquetage, en ajoutant des étiquettes, puis en répétant ces étapes plusieurs fois avec différents paramètres pour optimiser le modèle.

équipe de deux pizzas

Une petite DevOps équipe que vous pouvez nourrir avec deux pizzas. Une équipe de deux pizzas garantit les meilleures opportunités de collaboration possible dans le développement de logiciels.

U

incertitude

Un concept qui fait référence à des informations imprécises, incomplètes ou inconnues susceptibles de compromettre la fiabilité des modèles de ML prédictifs. Il existe deux types d'incertitude : l'incertitude épistémique est causée par des données limitées et incomplètes, alors que l'incertitude aléatoire est causée par le bruit et le caractère aléatoire inhérents aux données. Pour plus d'informations, veuillez consulter le guide Quantifying uncertainty in deep learning systems.

tâches indifférenciées

Également connu sous le nom de « levage de charges lourdes », ce travail est nécessaire pour créer et exploiter une application, mais qui n'apporte pas de valeur directe à l'utilisateur final ni d'avantage concurrentiel. Les exemples de tâches indifférenciées incluent l'approvisionnement, la maintenance et la planification des capacités.

environnements supérieurs

Voir environnement.

V

mise à vide

Opération de maintenance de base de données qui implique un nettoyage après des mises à jour incrémentielles afin de récupérer de l'espace de stockage et d'améliorer les performances.

contrôle de version

Processus et outils permettant de suivre les modifications, telles que les modifications apportées au code source dans un référentiel.

Appairage de VPC

Une connexion entre deux VPCs qui vous permet d'acheminer le trafic en utilisant des adresses IP privées. Pour plus d'informations, veuillez consulter la rubrique Qu'est-ce que l'appairage de VPC ? dans la documentation Amazon VPC.

vulnérabilités

Défaut logiciel ou matériel qui compromet la sécurité du système.

W

cache actif

Cache tampon qui contient les données actuelles et pertinentes fréquemment consultées. L'instance de base de données peut lire à partir du cache tampon, ce qui est plus rapide que la lecture à partir de la mémoire principale ou du disque.

données chaudes

Données rarement consultées. Lorsque vous interrogez ce type de données, des requêtes modérément lentes sont généralement acceptables.

fonction de fenêtre

Fonction SQL qui effectue un calcul sur un groupe de lignes liées d'une manière ou d'une autre à l'enregistrement en cours. Les fonctions de fenêtre sont utiles pour traiter des tâches, telles que le calcul d'une moyenne mobile ou l'accès à la valeur des lignes en fonction de la position relative de la ligne en cours.

charge de travail

Ensemble de ressources et de code qui fournit une valeur métier, par exemple une application destinée au client ou un processus de backend.

flux de travail

Groupes fonctionnels d'un projet de migration chargés d'un ensemble de tâches spécifique. Chaque flux de travail est indépendant, mais prend en charge les autres flux de travail du projet.

Par exemple, le flux de travail du portefeuille est chargé de prioriser les applications, de planifier les vagues et de collecter les métadonnées de migration. Le flux de travail du portefeuille fournit ces actifs au flux de travail de migration, qui migre ensuite les serveurs et les applications.

VER

Voir écrire une fois, lire plusieurs.

WQF

Voir le cadre AWS de qualification de la charge de travail.

écrire une fois, lire plusieurs (WORM)

Modèle de stockage qui écrit les données une seule fois et empêche leur suppression ou leur modification. Les utilisateurs autorisés peuvent lire les données autant de fois que nécessaire, mais ils ne peuvent pas les modifier. Cette infrastructure de stockage de données est considérée comme immuable.

Z

exploit Zero-Day

Une attaque, généralement un logiciel malveillant, qui tire parti d'une vulnérabilité de type « jour zéro ».

vulnérabilité « jour zéro »

Une faille ou une vulnérabilité non atténuée dans un système de production. Les acteurs malveillants peuvent utiliser ce type de vulnérabilité pour attaquer le système. Les développeurs prennent souvent conscience de la vulnérabilité à la suite de l'attaque.

invite Zero-Shot

Fournir à un LLM des instructions pour effectuer une tâche, mais aucun exemple (plans) pouvant aider à la guider. Le LLM doit utiliser ses connaissances pré-entraînées pour gérer la tâche. L'efficacité de l'invite zéro dépend de la complexité de la tâche et de la qualité de l'invite. Voir également les instructions en quelques clics.

application zombie

Application dont l'utilisation moyenne du processeur et de la mémoire est inférieure à 5 %. Dans un projet de migration, il est courant de retirer ces applications.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.