



メインフレーム移行用のASCII環境でのEBCDICアプリケーションのリホスト

AWS 規範ガイド



AWS 規範ガイド: メインフレーム移行用のASCII環境でのEBCDICアプリケーションのリホスト

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は、Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

序章	1
データ移行の課題	2
文字セット変換	2
ゾーン 10 進数形式	2
BINARY (COMP または COMP-4)	5
パックされた 10 進形式 (COMP-3)	5
複雑なレコードレイアウト	6
部分的なレコード記述を使用するプログラム	9
リレーショナルデータベース	11
データボリューム	11
アプリケーション移行の課題	13
ソースコードの依存関係	13
照合順序	14
物理環境の課題	16
メリットとデメリット	17
残り EBCDIC	17
への移行 ASCII	17
ハイブリッドソリューションの問題	18
ドキュメント履歴	19
用語集	20
#	20
A	21
B	23
C	25
D	29
E	32
F	34
G	36
H	36
I	38
L	40
M	41
O	45
P	48

Q	50
R	51
S	53
T	57
U	58
V	59
W	59
Z	60
.....	lxii

メインフレーム移行用のASCII環境でのEBCDICアプリケーションのリホスト

Peter West (Amazon Web Services (AWS))

2022年10月 ([ドキュメント履歴](#))

メインフレーム移行の一環として、EBCDIC (拡張バイナリコード小数交換コード) アプリケーションを AWS クラウド内の (米国情報交換標準コード) または ASCII (ANSI) 米国国立標準研究所環境に移行するときに直面する可能性のある課題を理解することが重要です。課題は、メインフレームアプリケーションがEBCDIC文字セットを使用し、Windows、Unix、Linux 環境がASCII文字セットを使用するという事実から発生します。このガイドでは、これらの課題を分析し、メインフレームアプリケーションを に維持すること、EBCDICまたはそれらを に移行することの長所と短所を比較検討するのに役立ちますASCII。このガイドでは、課題を次のカテゴリに分類します。

- データ移行の課題
- アプリケーション移行の課題
- 物理環境の課題

データ移行の課題

リレーショナルデータベースに加えて、ほとんどのメインフレーム環境には、シーケンシャルファイルや Virtual Storage Access Method (VSAM) ファイルなどの物理データセットの組み合わせがあります。データベースの移行では、多くの場合、データを抽出、変換、およびターゲットデータベースに挿入するツールが必要です。このセクションでは、物理データセットの課題について概説します。

文字セット変換

データファイルを から EBCDIC に変換するには、変換テーブルを使用する必要があります。ASCII。[変換テーブル](#)は、byte-to-byteファイル内のすべてのバイトに適用される変換を記述します。変換テーブルを使用して、英数字データを含むCOBOLファイルを明示的な符号文字で正しく変換できます。

次の表は、変換テーブルを使用する際に EBCDIC-to-ASCII変換の問題によって影響を受けるデータ型を示しています。

文字セット変換の問題がないデータ型	文字セット変換の問題があるデータ型
PIC X(n)	PIC S9 (n)
PIC (n)	PIC S9 または 9 BINARY/COMP
PIC 9 (n) (署名なし)	PIC S9 または 9 COMP-3
PIC S9 (n) SIGN SEPARATE	

ゾーン 10 進数形式

[ゾーン 10 進数形式](#)のデータは、通常の文字数値データのように見えます。違いは記号の保存方法にあります。ゾーン付き小数値フィールドは、署名付き数値表示フィールドの場合は PICinS9(n)、署名なし数値表示フィールドの場合は PIC 9(n) のように、符号インジケータを除く通常の数値表示フィールドのようになります。ゾーン 10 進数フィールドでは、符号は最後のバイトの上位ニブルに格納されます。

次の表は、データと 値の両方で「+/-」 1~9 を表すデータに保存されているEBCDICASCII値を示しています。

数字	EBCDIC 16 進数 バイナリ	EBCDIC 表示	ASCII 16 進数バ イナリ	ASCII 表示
+0	X"C0"-1100 0000	{	X"30"-0011 0000	0
+1	X"C1"-1100 0001	A	X"31"-0011 0001	1
2+	X"C2"-1100 0010	B	X"32"-0011 0010	2
+3	X"C3"-1100 0011	C	X"33"-0011 0011	3
+4	X"C4"-1100 0100	D	X"34"-0011 0100	4
5+	X"C5"-1100 0101	E	X"35"-0011 0101	5
+6	X"C6"-1100 0110	F	X"36"-0011 0110	6
+7	X"C7"-1100 0111	G	X"37"-0011 0111	7
+8	X"C8"-1100 1000	H	X"38"-0011 1000	8
+9	X"C9"-1100 1001	I	X"39"-0011 1001	9
-0	X"D0"-1101 0000	}	X"70"-0111 0000	p
-1	X"D1"-1101 0001	J	X"71"-0111 0001	q

-2	X"D2"-1101 0010	K	X"72"-0111 0010	r
-3	X"D3"-1101 0011	L	X"73"-0111 0011	s
-4	X"D4"-1101 0100	M	X"74"-0111 0100	t
-5	X"D5"-1101 0101	N	X"75"-0111 0101	u
-6	X"D6"-1101 0110	O	X"76"-0111 0110	v
-7	X"D7"-1101 0111	P	X"77"-0111 0111	w
-8	X"D8"-1101 0100	Q	X"78"-0111 1000	x
-9	X"D9"-1101 1001	R	X"79"-0111 1001	y

例えば、PICS9(4)に「-1234」値が含まれている場合、その値はX「F1F2F3D4EBCDIC」としてに保存されます。各バイトは変換テーブルでEBCDIC-to-ASCII検索され、対応するASCII値に変換されます。この例では、最初の3 EBCDIC文字は数値からASCII数値X「313233」に正しく変換されます。ただし、パンチ超過記号のニブルX「D4」「M」を含む最後の文字は、「MASCII」に相当する「X」4Dに変換されます。

このゾーン付き小数文字の正しいASCII値はX「74」「t」です。EBCDIC-to-ASCII変換テーブルを使用してこのフィールドを変換する場合、結果はX3132334Dv「123M」であり、正しいX31323374「123t」ではありません。このデータを正しく処理しないと、アプリケーションでデータ破損や無効な数値データエラーが発生する可能性があります。

BINARY (COMP または COMP-4)

[BINARY \(COMP または COMP-4\)](#) データはバイナリ形式で保存されます。COMP フィールドは通常、2 バイトの倍数で検出され、同等の数値表示項目の数値範囲を超える数値データ値を保存するために使用されます。例えば、PIC9(4) は、「0000」～「9999」の値を含めることができる 2 バイトの数値表示項目です。値「1234」は、「F1F2F3F4」としてに保存EBCDICされ、「31323334」はに変換されますASCII。

PIC 9(4) COMPは 2 バイトのバイナリフィールド (C の短いint と同じ) で、「0」～「65535」の値を含めることができます。最大値を表示数値フィールドに保存するには PIC 9 (6) が必要です。これにはさらに 2 バイトがかかります。次の例は、2 進数で表される値 "1234" を示しています。

バイト 1	バイト 2
0 0 0 0 0 1 0 0	1 1 0 1 0 0 1 0

16 進数データとして表される前述の例には、値 X"04D2"が含まれています。変換テーブルを使用して EBCDIC-to-ASCIIこのデータを変換すると、データは X「1A4B」に変換されます。数値として解釈すると、値は "6731" で、予想される "1234" ではありません。

COMP データは EBCDICと の両方に同じように保存されるため、変換は必要ありません ASCII。COMP データを正しく処理しないと、データ破損が発生する可能性があります。COMP データ項目内のすべての値は有効な数値であるため、アプリケーションによって報告される数値処理エラーはありません。値が単純に正しくないのです。前述の例が示すように、「1234」の値は "6731" になります。

パックされた 10 進形式 (COMP-3)

[COMP-3 フィールド](#)は、各バイトの低いニブルのみを保存することで、1 バイトあたり 2 桁を保存します。例えば、値「1」は、の X「F1EBCDIC」との X「31」で表されますASCII。値「2」は、の X「F2EBCDIC」との X「32」で表されますASCII。COMP-3 フィールドでは、各バイトのニブルは 1 バイトに保存されます。したがって、の「12」X「F1F2」または「3132」値は、「012C」として COMP-3 にASCII保存されます。

COMP-3 フィールドには、最後のバイトの低いニブルに符号なし値と正の署名値の「C」文字、負の署名値の「D」文字が含まれます。COMP-3 データは EBCDICと に同じように保存されるため、変換は必要ありませんASCII。EBCDIC-to-ASCII 変換テーブルを使用して COMP-3 データを変換すると、データが破損し、アプリケーションで無効な数値データエラーが発生します。

複雑なレコードレイアウト

ゾーン化された 10 進数、または COMP-3 フィールドを含むファイルはCOMP、データファイルから EBCDIC に変換するときには考慮すべき唯一の問題ではありませんASCII。ファイル自体には、さまざまなレコードレイアウトや、REDEFINES句を使用して特別な処理が必要なデータ領域の組み合わせを導入する領域を含めることができます。このような場合は、Micro Focus [データファイルエディタ](#)を使用して、ファイルの構造を定義するレコードレイアウトを作成し、から EBCDIC にASCII正しく変換できるようにします。レコードレイアウトの作成を開始する前に、まず作業するファイルの種類と構造を特定する必要があります。考えられるタイプには次のようなものがあります。

- 非テキストデータを含む単一レコード構造
- の領域内の非テキストデータを含む単一レコード構造 REDEFINES
- 非テキストデータを含む複数の 01 レコード構造

非テキストデータを含む単一レコード構造

非テキストデータを含む単一のレコード構造は、を含まない単一の 01 レコード領域を含む任意のファイルでREDEFINES、データ領域には ZD、COMP、または COMP-3 データ型が含まれます。非テキストデータを含む単一レコード構造には、Micro Focus データファイルエディタを使用して構築された単純な「デフォルトレコードレイアウト」が必要です。次の例は、非テキストデータを含む単一のレコード構造を示しています。

```
01 S-PARTS-RECORD.  
 05 S-PART-ID          PIC 9(9) COMP.  
 05 S-PART-TYPE        PIC X(2).  
 05 S-PART-NAME        PIC X(40).  
 05 S-SUB-PART-DATA.  
 10 S-SUB-DESC         PIC X(40).  
 10 S-SUB-COST         PIC S9(4)V99 COMP.  
 10 S-SUB-WEIGHT       PIC 9(4)V99 COMP-3.  
 10 FILLER             PIC X(34).
```

の領域内の非テキストデータを含む単一レコード構造 REDEFINES

の領域内に非テキストデータを含む単一のレコード構造REDEFINESは、を含む単一の 01 レコード領域を含む任意のファイルでREDEFINES、データ領域には ZD、COMP、または COMP-3 データ型が含まれます。次の例は、テキストデータと非テキストデータを組み合わせた共通の PIC X(80) エリアREDEFINESを定義する 2 つの例を示しています。

```
01 PARTS-RECORD.  
05 PART-ID PIC 9(9) COMP.  
05 PART-TYPE PIC X(2).  
05 PART-NAME PIC X(40).  
05 PART-DATA PIC X(80).  
05 MAIN-PART REDEFINES PART-DATA.  
10 MAIN-DESC PIC X(40).  
10 MAIN-SUB-COUNT PIC 9(2) COMP.  
10 MAIN-ASSEMBLIES OCCURS 10.  
15 MAIN-SUB-ID PIC 9(9) COMP.  
05 SUB-PART REDEFINES PART-DATA.  
10 SUB-DESC PIC X(40).  
10 SUB-COST PIC S9(4)V99 COMP.  
10 SUB-WEIGHT PIC 9(4)V99 COMP-3.  
10 FILLER PIC X(34).
```

このタイプのレコード構造は、まず分解して、レコード構造の複数のビューを作成するREDEFINESステートメントを削除する必要があります。レコード構造を分解しない場合、作成された構造ファイルは、下の領域を無視REDEFINESし、領域全体をテキストデータとして扱います。前の例では、REDEFINES句の下に2つの異なる非テキスト構造について説明します。作成した構造ファイルには、これらの領域をデータコンバーターが対象とする一意の構造の一部として記述する必要があります。次の例は、REDEFINESが削除された後の2つの一意のレイアウトを示しています。

```
01 M-PARTS-RECORD.  
05 M-PART-ID PIC 9(9) COMP.  
05 M-PART-TYPE PIC X(2).  
05 M-PART-NAME PIC X(40).  
05 M-PART-DATA.  
10 M-MAIN-DESC PIC X(40).  
10 M-MAIN-SUB-COUNT PIC 9(2) COMP.  
10 M-MAIN-ASSEMBLIES OCCURS 10.  
15 M-MAIN-SUB-ID PIC 9(9) COMP.  
01 S-PARTS-RECORD.  
05 S-PART-ID PIC 9(9) COMP.  
05 S-PART-TYPE PIC X(2).  
05 S-PART-NAME PIC X(40).  
05 S-SUB-PART-DATA.  
10 S-SUB-DESC PIC X(40).  
10 S-SUB-COST PIC S9(4)V99 COMP.  
10 S-SUB-WEIGHT PIC 9(4)V99 COMP-3.  
10 FILLER PIC X(34).
```

次のステップは、あるレコードレイアウトを他方のレコードレイアウトから分離するために使用できる条件ステートメントを決定することです。条件ステートメントを決定するには、対象分野の専門家に相談するか、ソースコードを調べることを推奨します。次の例はソースコードを示しています。

```
MOVE "M" TO PART-TYPE
MOVE "MAIN ASSEMBLY" TO PART-NAME
MOVE "S" TO PART-TYPE
MOVE "SUB ASSEMBLY 1" TO PART-NAME
```

ソースコードでは、PART「-TYPE」フィールドを使用してレコードタイプを特定できます。値「M」は「M-PART-RECORD」に使用され、値「S」は「S-PART-RECORD」に使用されます。これで、「M-ID」フィールドと「SPART-ID」フィールドで識別された条件を使用して、2 PARTつの条件レコードのいずれかを含む構造ファイルを作成できます。または、1つのデフォルトレイアウトと1つの条件付きレイアウトを作成することもできます。

非テキストデータを含む複数の 01 レコード構造

非テキストデータを含む複数の 01 レコード構造は、次の例に示すように、ZD、、COMPまたはCOMP-3 データ型を含む複数の 01 レコード領域を含む任意のファイルです。

```
01 M-PARTS-RECORD.
  05 M-PART-ID          PIC 9(9) COMP.
  05 M-PART-TYPE       PIC X(2).
  05 M-PART-NAME       PIC X(40).
  05 M-PART-DATA.
    10 M-MAIN-DESC     PIC X(40).
    10 M-MAIN-SUB-COUNT PIC 9(2) COMP.
    10 M-MAIN-ASSEMBLIES OCCURS 10.
    15 M-MAIN-SUB-ID   PIC 9(9) COMP.

01 S-PARTS-RECORD.
  05 S-PART-ID          PIC 9(9) COMP.
  05 S-PART-TYPE       PIC X(2).
  05 S-PART-NAME       PIC X(40).
  05 S-SUB-PART-DATA.
    10 S-SUB-DESC     PIC X(40).
    10 S-SUB-COST     PIC S9(4)V99 COMP.
    10 S-SUB-WEIGHT   PIC 9(4)V99 COMP-3.
    10 FILLER         PIC X(34).
```

最初のステップは、1つのレコードレイアウトを別のレイアウトから分離するために使用できる条件ステートメントを決定することです。条件ステートメントを決定するには、対象分野の専門家に相談するか、ソースコードを調べることを推奨します。次の例はソースコードを示しています。

```
MOVE "M" TO PART-TYPE
MOVE "MAIN ASSEMBLY" TO PART-NAME
MOVE "S" TO PART-TYPE
MOVE "SUB ASSEMBLY 1" TO PART-NAME
```

ソースコードでは、PART「-TYPE」フィールドを使用してレコードタイプを特定できます。値「M」は「M-PART-RECORD」に使用され、値「S」は「S-PART-RECORD」に使用されます。これで、「M--ID」フィールドと「SPART--ID」フィールドで識別された条件を使用して、2 PARTつの条件レコードのいずれかを含む構造ファイルを作成できます。または、1つのデフォルトレイアウトと1つの条件付きレイアウトを作成することもできます。

部分的なレコード記述を使用するプログラム

コピーファイルを使用してファイルの形式全体を記述するのが良いプログラミングスタイルですが、特定のレコードやデータ構造を対象とする場合、プログラムに必要な構造のみをプログラマーが手作業でコーディングすることがあります。これには、大きなFILLER項目によってマスクされるファイルのセクションや、ファイルの一部を記述する複数のプログラムが含まれる場合があります。このような場合は、次のPROGRAM1およびPROGRAM2例に示すように、ファイルの完全な説明を含む単一のプログラムを作成する必要があります。

PROGRAM1 例 :

```
01 M-PARTS-RECORD.
   05 M-PART-ID          PIC 9(9) COMP.
   05 M-PART-TYPE       PIC X(2).
   05 M-PART-NAME       PIC X(40).
   05 M-PART-DATA.
      10 M-MAIN-DESC     PIC X(40).
      10 M-MAIN-SUB-COUNT PIC 9(2) COMP.
      10 M-MAIN-ASSEMBLIES OCCURS 10.
      15 M-MAIN-SUB-ID   PIC 9(9) COMP.
```

PROGRAM2 例 :

```
01 S-PARTS-RECORD.
```

```

05 S-PART-ID          PIC 9(9) COMP.
05 S-PART-TYPE        PIC X(2).
05 S-PART-NAME        PIC X(40).
05 S-SUB-PART-DATA.
   10 S-SUB-DESC      PIC X(40).
   10 S-SUB-COST      PIC S9(4)V99 COMP.
   10 S-SUB-WEIGHT    PIC 9(4)V99 COMP-3.
   10 FILLER          PIC X(34).

```

前述の例では、各プログラムのデータディクショナリを1つずつ読み込んだ後に、さまざまな構造をレコードレイアウトに追加できます。ただし、次の PROGRAM1 や PROGRAM2 の例に示すように、より複雑な状況が発生する可能性があります。

PROGRAM1 例：

```

01 M-PARTS-RECORD.
   05 M-PART-ID          PIC 9(9) COMP.
   05 M-PART-TYPE        PIC X(2).
   05 M-PART-NAME        PIC X(40).
   05 FILLER            PIC X(82).

```

PROGRAM2 例：

```

01 S-PARTS-RECORD.
   05 S-PART-ID          PIC 9(9) COMP.
   05 S-PART-TYPE        PIC X(2).
   05 S-PART-NAME        PIC X(40).
   05 S-SUB-PART-DATA.
     10 S-SUB-DESC      PIC X(40).
     10 S-SUB-COST      PIC S9(4)V99 COMP.
     10 S-SUB-WEIGHT    PIC 9(4)V99 COMP-3.
     10 FILLER          PIC X(34).

```

この場合、のレコードレイアウトの一部PROGRAM1は、FILLERステートメントを使用してマスクされます。この情報を使用してレコードレイアウトを構築すると、「M-PARTS-RECORD」のFILLERブロック内のデータはテキストとして扱われ、誤って変換されます。開発者は、レコードレイアウトを構築する前に、デューデリジェンスを実施してファイルの絶対構造を確認する必要があります。

リレーショナルデータベース

メインフレームのリレーショナルデータベースを分散型リレーショナルデータベースに変換する際には、いくつかの考慮事項があります。例えば、ASCII または ANSI文字セットに移動すると、シーケンスの問題がデータと照合される可能性があります。これらの問題の詳細については、このガイドの「アプリケーション移行の課題」にある「照合順序」のセクションを参照してください。

例えば、キー列は、メインフレームで返される順序とは異なる順序でデータを返す場合があります。CURSORS および WHERE句は、データベースの照合順序を尊重します。データにEBCDIC照合シーケンスへの依存関係がある場合は、可能であればデータベースにEBCDIC照合シーケンスを使用します。または、必要に応じてアプリケーションロジックを変更してEBCDIC照合順序を使用します。SQL ステートメントで代替照合シーケンスを指定すると、ステートメントの実行時にレイテンシーが増加することを考慮してください。

データボリューム

すべての移行と同様に、メインフレームから分散環境にデータを移行する必要があります。どの移行ツールを使用するかを検討する前に、まず移行するデータの量を検討することが重要です。データの移行は次の4つに分けることができます。

- 統合テストデータ
- システムテストデータ
- カットオーバーデータまたは稼働開始データ
- 履歴データとアーカイブデータ

統合データとシステムテストデータの移行は、使用前に長期間にわたってデータを移動できるため、通常は大きな問題にはなりません。代わりに、カットオーバーデータや稼働開始データ、履歴データやアーカイブデータに時間と労力を費やすほうがよいでしょう。

物理ファイルとデータベーステーブルの両方を含め、移行の早い段階で稼働開始イベント用のデータ量を決定するのがベストプラクティスです。ファイルのエクスポート、このデータの から EBCDIC への変換ASCII、ターゲットシステムへのデータのインポートから収集されたデータを使用することで、メインフレームからのデータのカットオーバーにかかる時間を見積もることができます。ビジネスがダウンする可能性のある時間や、稼働開始のデプロイ後に取り戻す時間を要するような状況によっては、データ移行ウィンドウが許容できない場合があります。次に、にとどまるかどうかを決定する必要があります。これによりEBCDIC、データカットを最大 50% 削減するか、事業停止や復旧の計画など、他の調整を行う可能性があります。

履歴データは並行プロジェクトとして実施されることが多く、通常は稼働開始のデプロイ前に完成しません。履歴データの変換は、特にレコードやテーブルのレイアウトが時間の経過とともに変化していると、難しい場合があります。組織がコンプライアンスや規制の要件を満たすことができるように、必要なデータを移行する計画を立てることを推奨します。

アプリケーション移行の課題

アプリケーションを から EBCDIC に移行するときはASCII、アプリケーション内のEBCDIC依存関係を分析して軽減することをお勧めします。これには、ソースコードの依存関係の分析とシーケンスの照合が含まれます。

ソースコードの依存関係

ソースコードが から に変換されると、これらの文字が変更されるため、数値を表すために使用される文字を確認することが重要ですEBCDICASCII。一部のメインフレームプログラマーは、次の例に示すように、コピーブックまたは作業ストレージのCOMPフィールド内の数値データを表すために、次の手法を使用します。

```
01 CHAR-74          PIC 99 COMP VALUE "¢".
```

ではEBCDIC、前の例の値は X「41」(数値として「74」)です。ソースデータが に変換されるとASCII、この文字は X「5B」(数値として「91」)に変更されます。この変更によってコンパイラの問題は発生しませんが、デバッグが困難なプログラミング上の問題が発生する可能性があります。次の例は、ソースを実際の数値を含むように変更する効果的なソリューションを示しています。

```
01 CHAR-74          PIC 99 COMP VALUE 74.
```

また、次の例のように、埋め込まれた 16 進文字が使用されているかどうかを確認することを推奨します。

```
IF WS-DATA(1:1) = X"5B" ...
```

ではEBCDIC、前の例では、フィールドの最初の文字をテストして、それが「\$」であるかどうかを調べています。「\$」のASCII表現は X「24」です。次の例は、目的の文字を使用する効果的なソリューションを示しています。

```
IF WS-DATA(1:1) = "$" ...
```

すべての X"nn" 値が問題を引き起こすわけではありません。例えば、定数 LOW-VALUE は X「00」、HIGH-VALUE は X「FF」です。LOW-VALUES (X「00」) は通常、署名なしバイナリフィールドの初期化に使用されます。例えば、次のステートメントは変更の必要がありません。

```
MOVE ALL X"00" TO WS-DATA
```

HIGH-VALUES (X「FF」) は、レコードが SORT または の末尾に配置され、ファイルの末尾にマークを付けるために、データマーカとして使用されることがあります。例えば、次のステートメントは変更の必要がありません。

```
MOVE ALL X"FF" TO WS-DATA
```

上記のステートメントはどちらも、EBCDIC または ASCII 環境で同じように動作します。

照合順序

ではEBCDIC、アルファベット文字が数値の前に続きます。ではASCII、数値はアルファベットの前にあります。したがって、 の英数字キーフィールド、VSAM またはアルファベットと数値の混合を含むデータベースは、移行後に異なる順序でデータを返します。照合順序の影響をアプリケーションレベルで分析して、正しい修復戦略を決定することを推奨します。

次の表は、英数字データの簡単な例と、このデータを EBCDIC および でソートする方法を示していますASCII。

EBCDIC	ASCII
A1	1A
A2	2B
1A	A1
2B	A2

照合順序は、以下の結果に影響します。

- VSAM キー付きアクセス (注: VSAMファイルは、ファイルが作成される照合順序の順序でデータを返します。EBCDIC 照合シーケンスを使用するようにCOBOLプログラムがコンパイルされていても、ASCII照合シーケンスで構築されたVSAMファイルはASCII、照合シーケンスを使用してデータを返します。ASCII 照合シーケンスは、アプリケーションの問題を引き起こす可能性があります。)
- 内部関数または外部SORT関数
- 条件ステートメント

- SQL カーソル (注: SQLカーソルは、照合シーケンスがDECLAREステートメントで定義されていない限り、データベースが作成される照合シーケンスの順序でデータを返します。EBCDIC 照合シーケンスを使用するようにCOBOLプログラムがコンパイルされていても、ASCII照合シーケンスを使用して構築されたデータベースはASCII、照合シーケンスを使用してデータを返します。ASCII 照合シーケンスは、アプリケーションの問題を引き起こす可能性があります。)

照合順序の依存関係は、アプリケーションだけでなく物理世界にも及ぶ可能性があります。例えば、倉庫内のストレージビンのバーコードは、EBCDIC値を使用して順序付けることができます。アプリケーションをに変更するにはASCII、ストレージビンのバーコードを物理的に変更する必要があります。

物理環境の課題

物理環境は、メインフレームの移行において最も見過ごされがちな要素の1つです。物理環境を分析すると、組織がにとどまるかEBCDIC、に移行するかを判断するのに役立ちますASCII。物理環境に関する次の点について考慮することを推奨します。

- オンプレミスの問題 – 例えば、組織がウェアハウスを管理し、ウェアハウスビンにEBCDIC英数字キーを使用してラベル付けEBCDICする場合、物理的な依存関係がある可能性があります。英数字キーのソートは EBCDIC と で異なるためASCII、作業現場の順序がASCII照合順序と一致するように、物理的なビンのラベルを変更する必要があります。
- 外部の問題 – 他の顧客との外部インターフェイスでは、 でファイルの送受信が必要になる場合がありますEBCDIC。サポートのために移行されたアプリケーションに加えられた変更はASCII、外部クライアントによるこのデータの処理に誤って影響する可能性があります。少なくとも、インバウンドデータを から EBCDIC に変換し、ASCIIアウトバウンドデータを から ASCII に戻すには、抽出、変換、ロード (ETL) プロセスを導入する必要がありますEBCDIC。

メリットとデメリット

移行の設計段階では、このガイドを使用して、にとどまるEBCDICか、に移行するかの決定を通知しますASCII。次のセクションでは、への移行EBCDICと比較した、に残っていることの長所と短所の概要を示しますASCII。

残り EBCDIC

メリット

- から EBCDIC へのデータの変換を回避することでリスクを低減 ASCII
- EBCDIC 依存関係が欠落するアプリケーションの問題を回避することでリスクを軽減
- 外部クライアントがEBCDICデータを送受信する際のリスク軽減 (注: ETL ファイルタイプの変換には多少あるが、データ変換は行われたい可能性が高い)
- 物理的なEBCDIC依存関係を回避することでリスクを軽減する
- データ移行、特に稼働開始デプロイでのデータ移行に必要な時間を短縮することによるリスク軽減
- データ移行にかかる時間を短縮することによるコスト削減
- EBCDIC 依存関係のコード分析に必要な時間を短縮することでコストを削減
- コードの移行とテストにかかる時間を短縮することによるコスト削減

デメリット

- 非COBOLダウンストリーム処理、特にASCII照合シーケンスを使用するように記述されたアプリケーションの複雑さが増す
- ASCII データを必要とするアップストリームインターフェイスとダウンストリームインターフェイスの複雑さが増す

への移行 ASCII

メリット

- ネイティブの文字セットと照合順序を使用可能
- 非COBOL環境との統合を簡素化

デメリット

- メインフレーム環境とのデータ交換が複雑化

ハイブリッドソリューションの問題

プログラミングの柔軟性により、ハイブリッドソリューションが可能になります。例えば、データを変換ASCIIしてでアプリケーションを実行しながらASCII、EBCDIC照合シーケンスを使用することもできます。この方法は多くの問題を解決できますが、次のような課題も生じます。

- すべてのプログラミング言語が、作業文字セットと照合順序を個別に選択的に変更できるわけではありません。これにより、非COBOLアプリケーションがEBCDIC照合シーケンスを使用できないため、データを正しく処理できない状況が発生する可能性があります。例えば、SQLテーブルからのデータはEBCDIC照合シーケンスを使用して返されますが、プログラム内の条件ステートメントはASCII照合シーケンスに依存します。
- すべてのサードパーティ製ツールやアプリケーションがハイブリッドに機能するわけではありません。
- プログラマーが、新しいアプリケーションのハイブリッド要件を知らない可能性があります。

理想的には、最初のオプションは に留まるか、 に完全にEBCDIC移行することですASCII。ハイブリッドソリューションを検討するのは、 にとどまるか、 EBCDIC に移行するという課題を克服ASCIIできない場合のみです。

ドキュメント履歴

このガイドは、このドキュメントの大きな変更点をまとめたものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#)をサブスクライブできます。

変更	説明	日付
初版発行	—	2022 年 10 月 4 日

AWS 規範的ガイドの用語集

以下は、AWS 規範的ガイドが提供する戦略、ガイド、パターンで一般的に使用される用語です。エントリを提案するには、用語集の最後のフィードバックの提供リンクを使用します。

数字

7 Rs

アプリケーションをクラウドに移行するための7つの一般的な移行戦略。これらの戦略は、ガートナーが2011年に特定した5Rsに基づいて構築され、以下で構成されています。

- リファクタリング/アーキテクチャの再設計 — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更します。これには、通常、オペレーティングシステムとデータベースの移植が含まれます。例: オンプレミスの Oracle データベースを Amazon Aurora PostgreSQL 互換エディションに移行します。
- リプラットフォーム (リフトアンドリシェイプ) – アプリケーションをクラウドに移行し、クラウド機能を活用するためある程度の最適化を導入します。例: オンプレミスの Oracle データベースをの Oracle 用 Amazon Relational Database Service (Amazon RDS) に移行します AWS クラウド。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: 顧客関係管理 (CRM) システムを Salesforce.com に移行します。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: オンプレミスの Oracle データベースをの EC2 インスタンス上の Oracle に移行します AWS クラウド。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) – 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。サーバーをオンプレミスプラットフォームから同じプラットフォームのクラウドサービスに移行します。例: Microsoft Hyper-V アプリケーションをに移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらに移行するためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。
- 使用停止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

[「属性ベースのアクセスコントロール」](#)を参照してください。

抽象化されたサービス

[「マネージドサービス」](#)を参照してください。

ACID

[「アトミック性、一貫性、分離性、耐久性」](#)を参照してください。

アクティブ - アクティブ移行

(双方向レプリケーションツールまたは二重書き込み操作を使用して) ソースデータベースとターゲットデータベースを同期させ、移行中に両方のデータベースが接続アプリケーションからのトランザクションを処理するデータベース移行方法。この方法では、1 回限りのカットオーバーの必要がなく、管理された小規模なバッチで移行できます。アクティブ [/パッシブ移行](#) よりも柔軟ですが、より多くの作業が必要です。

アクティブ - パッシブ移行

ソースデータベースとターゲットデータベースを同期させながら、データがターゲットデータベースにレプリケートされている間、接続しているアプリケーションからのトランザクションをソースデータベースのみで処理するデータベース移行の方法。移行中、ターゲットデータベースはトランザクションを受け付けません。

集計関数

行のグループを操作し、グループの単一の戻り値を計算するSQL関数。集計関数の例としては、SUMや MAX などが含まれます。

AI

[「人工知能」](#)を参照してください。

AIOps

[「人工知能オペレーション」](#)を参照してください。

匿名化

データセット内の個人情報を完全に削除するプロセス。匿名化は個人のプライバシー保護に役立ちます。匿名化されたデータは、もはや個人データとは見なされません。

アンチパターン

繰り返し起こる問題に対して頻繁に用いられる解決策で、その解決策が逆効果であったり、効果がなかったり、代替案よりも効果が低かったりするもの。

アプリケーションコントロール

マルウェアからシステムを保護するために、承認されたアプリケーションのみを使用できるようにするセキュリティアプローチ。

アプリケーションポートフォリオ

アプリケーションの構築と維持にかかるコスト、およびそのビジネス価値を含む、組織が使用する各アプリケーションに関する詳細情報の集まり。この情報は、[ポートフォリオの検出と分析プロセス](#)の需要要素であり、移行、モダナイズ、最適化するアプリケーションを特定し、優先順位を付けるのに役立ちます。

人工知能 (AI)

コンピューティングテクノロジーを使用し、学習、問題の解決、パターンの認識など、通常は人間に関連づけられる認知機能の実行に特化したコンピュータサイエンスの分野。詳細については、「[人工知能 \(AI\) とは何ですか?](#)」を参照してください。

人工知能オペレーション (AIOps)

機械学習技術を使用して運用上の問題を解決し、運用上のインシデントと人の介入を減らし、サービス品質を向上させるプロセス。移行戦略での AIOps の使用方法の詳細については、AWS「[オペレーション統合ガイド](#)」を参照してください。

非対称暗号化

暗号化用のパブリックキーと復号用のプライベートキーから成る 1 組のキーを使用した、暗号化のアルゴリズム。パブリックキーは復号には使用されないため共有しても問題ありませんが、プライベートキーの利用は厳しく制限する必要があります。

アトミック性、一貫性、分離性、耐久性 (ACID)

エラー、停電、その他の問題が発生した場合でも、データベースのデータ有効性と運用上の信頼性を保証する一連のソフトウェアプロパティ。

属性ベースのアクセスコントロール (ABAC)

部署、役職、チーム名など、ユーザーの属性に基づいてアクセス許可をきめ細かく設定する方法。詳細については、AWS Identity and Access Management (IAM) ドキュメントの「[for ABAC AWS](#)」を参照してください。

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリーバージョンを保存する場所。匿名化、編集、仮名化など、データを処理または変更する目的で、信頼できるデータソースから他の場所にデータをコピーすることができます。

アベイラビリティゾーン

他のアベイラビリティゾーンの障害から AWS リージョン 隔離され、同じリージョン内の他のアベイラビリティゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドに正常に移行 AWS するための効率的で効果的な計画を立てるのに役立つ、からのガイドラインとベストプラクティスのフレームワークです。AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを編成します。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、AWS CAF は、組織がクラウド導入を成功させるための準備に役立つ、人材開発、トレーニング、コミュニケーションに関するガイダンスを提供します。詳細については、[AWS CAFウェブサイト](#)と[AWS CAFホワイトペーパー](#)を参照してください。

AWS ワークロード認定フレームワーク (AWS WQF)

データベース移行ワークロードを評価し、移行戦略を推奨し、作業見積もりを提供するツール。AWS WQF は AWS Schema Conversion Tool (AWS SCT) に含まれています。データベーススキーマとコードオブジェクト、アプリケーションコード、依存関係、およびパフォーマンス特性を分析し、評価レポートを提供します。

B

不正なボット

個人または組織に混乱や損害を与えることを目的とした[ボット](#)。

BCP

[事業継続計画を参照してください](#)。

動作グラフ

リソースの動作とインタラクションを経時的に示した、一元的なインタラクティブビュー。Amazon Detective で動作グラフを使用して、失敗したログオン試行、疑わしいAPI呼び出し、および同様のアクションを調べることができます。詳細については、Detective ドキュメントの[Data in a behavior graph](#)を参照してください。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。[エンディアンネス](#) も参照してください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの一つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

要素がセットのメンバーであるかどうかをテストするために使用される、確率的でメモリ効率の高いデータ構造。

ブルー/グリーンデプロイ

2 つの異なる同一の環境を作成するデプロイ戦略。現在のアプリケーションバージョンは 1 つの環境 (ブルー) で実行し、新しいアプリケーションバージョンは他の環境 (グリーン) で実行します。この戦略は、影響を最小限に抑えながら迅速にロールバックするのに役立ちます。

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティやインタラクションをシミュレートするソフトウェアアプリケーション。インターネット上の情報のインデックスを作成するウェブクローラーなど、一部のボットは有用または有益です。悪質なボットと呼ばれる他のボットの中には、個人や組織に混乱を与えたり、損害を与えたりすることを意図しているものがあります。

ボットネット

[マルウェア](#)に感染し、[ボット](#)のヘルダーまたはボットオペレーターと呼ばれる、単一関係者の管理下にあるボットのネットワーク。ボットは、ボットとその影響をスケールするための最もよく知られているメカニズムです。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチと
いいます。既存のブランチから新しいブランチを作成し、その新しいブランチで機能を開発した
り、バグを修正したりできます。機能を構築するために作成するブランチは、通常、機能ブラン
チと呼ばれます。機能をリリースする準備ができたなら、機能ブランチをメインブランチに統合し
ます。詳細については、[「ブランチについて」](#) (GitHub ドキュメント) を参照してください。

ブレイクグラスアクセス

例外的な状況や承認されたプロセスを通じて、ユーザーが通常アクセス許可を持たない AWS ア
カウント にすばやくアクセスできるようになります。詳細については、Well-Architected [ガイド
ンスの「ブレイクグラス手順の実装」](#) インジケータ AWS を参照してください。

ブラウフィールド戦略

環境の既存インフラストラクチャ。システムアーキテクチャにブラウフィールド戦略を導入す
る場合、現在のシステムとインフラストラクチャの制約に基づいてアーキテクチャを設計しま
す。既存のインフラストラクチャを拡張している場合は、ブラウフィールド戦略と[グリーン
フィールド](#)戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

価値を生み出すためにビジネスが行うこと (営業、カスタマーサービス、マーケティングなど)。
マイクロサービスのアーキテクチャと開発の決定は、ビジネス能力によって推進できます。詳細
については、ホワイトペーパー [AWSでのコンテナ化されたマイクロサービスの実行](#) の [ビジネス
機能を中心に組織化](#) セクションを参照してください。

事業継続計画 (BCP)

大規模移行など、中断を伴うイベントが運用に与える潜在的な影響に対処し、ビジネスを迅速に
再開できるようにする計画。

C

CAF

[AWS 「クラウド導入フレームワーク」](#) を参照してください。

Canary デプロイ

エンドユーザーへのバージョンの低速かつ増分的なリリース。確信できたら、新しいバージョンをデプロイし、現在のバージョン全体を置き換えます。

CCoE

[「Cloud Center of Excellence」](#)を参照してください。

CDC

[「データキャプチャの変更」](#)を参照してください。

データキャプチャの変更 (CDC)

データソース (データベーステーブルなど) の変更を追跡し、その変更に関するメタデータを記録するプロセス。は、同期を維持するために、ターゲットシステムの変更を監査またはレプリケートするなど、CDCさまざまな目的で使用できます。

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストします。[AWS Fault Injection Service \(AWS FIS \)](#) を使用して、AWS ワークロードに負荷をかけてレスポンスを評価する実験を実行できます。

CI/CD

[「継続的インテグレーションと継続的デリバリー」](#)を参照してください。

分類

予測を生成するのに役立つ分類プロセス。分類問題の機械学習モデルは、離散値を予測します。離散値は、常に互いに区別されます。例えば、モデルがイメージ内に車があるかどうかを評価する必要がある場合があります。

クライアント側の暗号化

ターゲットがデータ AWS のサービスを受信する前に、ローカルでデータを暗号化します。

Cloud Center of Excellence (CCoE)

クラウドのベストプラクティスの作成、リソースの移動、移行のタイムラインの確立、大規模変革を通じて組織をリードするなど、組織全体のクラウド導入の取り組みを推進する学際的なチーム。詳細については、AWS クラウド エンタープライズ戦略ブログ [CCoEの投稿](#) を参照してください。

クラウドコンピューティング

リモートデータストレージと IoT デバイス管理に通常使用されるクラウドテクノロジー。クラウドコンピューティングは、一般的に[エッジコンピューティング](#)テクノロジーに接続されています。

クラウド運用モデル

IT 組織において、1 つ以上のクラウド環境を構築、成熟、最適化するために使用される運用モデル。詳細については、[「クラウド運用モデルの構築」](#)を参照してください。

導入のクラウドステージ

組織が に移行するときに通常実行する 4 つのフェーズ AWS クラウド :

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基盤 — クラウド導入を拡大するための基本的な投資 (ランディングゾーンの作成、 の定義 CCoE、運用モデルの確立など)
- 移行 — 個々のアプリケーションの移行
- 再発明 — 製品とサービスの最適化、クラウドでのイノベーション

これらのステージは、AWS クラウド エンタープライズ戦略ブログのブログ記事[「クラウドファーストへのジャーニー」](#)と[「導入のステージ」](#)で Stephen Orban によって定義されました。移行戦略とどのように関連しているかについては、AWS [「移行準備ガイド」](#)を参照してください。

CMDB

[「設定管理データベース」](#)を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、GitHub または が含まれます AWS CodeCommit。コードの各バージョンはブランチと呼ばれます。マイクロサービスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

空である、または、かなり空きがある、もしくは、古いデータや無関係なデータが含まれているバッファキャッシュ。データベースインスタンスはメインメモリまたはディスクから読み取る必

要があり、バッファキャッシュから読み取るよりも時間がかかるため、パフォーマンスに影響します。

コールドデータ

めったにアクセスされず、通常は過去のデータです。この種類のデータをクエリする場合、通常は低速なクエリでも問題ありません。このデータを低パフォーマンスで安価なストレージ階層またはクラスに移動すると、コストを削減することができます。

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオなどのビジュアル形式から情報を分析および抽出する [AI](#) の分野。例えば、AWS Panorama はオンプレミスのカメラネットワークに CV を追加するデバイスを提供し、Amazon SageMaker は CV の画像処理アルゴリズムを提供します。

設定ドリフト

ワークロードの場合、設定は想定した状態から変化します。これにより、ワークロードが非準拠になる可能性があり、通常は段階的かつ意図的ではありません。

設定管理データベース (CMDB)

データベースとその IT 環境 (ハードウェアとソフトウェアの両方のコンポーネントとその設定を含む) に関する情報を保存、管理するリポジトリ。通常、移行のポートフォリオ検出および分析段階で CMDB のデータを使用します。

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。テンプレートを使用して、コンフォーマンスパックを AWS アカウント およびリージョンの単一のエンティティとしてデプロイすることも、組織全体にデプロイすることもできます。YAML。詳細については、AWS Config ドキュメントの「[コンフォーマンスパック](#)」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「[継続的デリバリーの利点](#)」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「[継続的デリバリーと継続的なデプロイ](#)」を参照してください。

CV

「[コンピュータビジョン](#)」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

ネットワーク内のデータを重要度と機密性に基づいて識別、分類するプロセス。データに適した保護および保持のコントロールを判断する際に役立つため、あらゆるサイバーセキュリティのリスク管理戦略において重要な要素です。データ分類は、AWS Well-Architected フレームワークのセキュリティの柱のコンポーネントです。詳細については、[データ分類](#)を参照してください。

データドリフト

実稼働データと ML モデルのトレーニングに使用されたデータとの間に有意な差異が生じたり、入力データが時間の経過と共に有意に変化したりすることです。データドリフトは、ML モデル予測の全体的な品質、精度、公平性を低下させる可能性があります。

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

一元化された管理とガバナンスにより、分散型の分散型データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

厳密に必要なデータのみを収集し、処理するという原則。でデータ最小化を実践 AWS クラウドすることで、プライバシーリスク、コスト、分析のカーボンフットプリントを削減できます。

データ境界

AWS 環境内の一連の予防ガードレール。信頼できる ID のみが、期待されるネットワークから信頼できるリソースにアクセスしていることを確実にします。詳細については、「[でのデータ境界の構築 AWS](#)」を参照してください。

データの事前処理

raw データをお客様の機械学習モデルで簡単に解析できる形式に変換すること。データの事前処理とは、特定の列または行を削除して、欠落している、矛盾している、または重複する値に対処することを意味します。

データ出所

データの生成、送信、保存の方法など、データのライフサイクル全体を通じてデータの出所と履歴を追跡するプロセス。

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには通常、大量の履歴データが含まれており、クエリや分析によく使用されます。

データベース定義言語 (DDL)

データベース内のテーブルやオブジェクトの構造を作成または変更するためのステートメントまたはコマンド。

データベース操作言語 (DML)

データベース内の情報を変更 (挿入、更新、削除) するためのステートメントまたはコマンド。

DDL

[「データベース定義言語」](#)を参照してください。

ディープアンサンブル

予測のために複数の深層学習モデルを組み合わせる。ディープアンサンブルを使用して、より正確な予測を取得したり、予測の不確実性を推定したりできます。

ディープラーニング

人工ニューラルネットワークの複数層を使用して、入力データと対象のターゲット変数の間のマッピングを識別する機械学習サブフィールド。

defense-in-depth

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリティの手法。この戦略をに採用するときは AWS、AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加して、リソースの安全性を確保します。例えば、defense-in-depth アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

では AWS Organizations、互換性のあるサービスが AWS メンバーアカウントを登録して組織のアカウントを管理し、そのサービスのアクセス許可を管理できます。このアカウントを、そのサービスの委任管理者と呼びます。詳細、および互換性のあるサービスの一覧は、AWS Organizations ドキュメントの[AWS Organizationsで利用できるサービス](#)を参照してください。

デプロイメント

アプリケーション、新機能、コードの修正をターゲットの環境で利用できるようにするプロセス。デプロイでは、コードベースに変更を施した後、アプリケーションの環境でそのコードベースを構築して実行します。

開発環境

[「環境」](#)を参照してください。

検出管理

イベントが発生したときに、検出、ログ記録、警告を行うように設計されたセキュリティコントロール。これらのコントロールは副次的な防衛手段であり、実行中の予防的コントロールをすり抜けたセキュリティイベントをユーザーに警告します。詳細については、Implementing security controls on AWSの[Detective controls](#)を参照してください。

開発値ストリームマッピング (DVSM)

ソフトウェア開発ライフサイクルのスピードと品質に悪影響を及ぼす制約を特定し、優先順位を付けるために使用されるプロセス。DVSM は、もともとリーンマニユファクチャリングプラクティス用に設計されたバリューストリームマッピングプロセスを拡張します。ソフトウェア開発プロセスを通じて価値を創造し、動かすために必要なステップとチームに焦点を当てています。

デジタルツイン

建物、工場、産業機器、生産ラインなど、現実世界のシステムを仮想的に表現したものです。デジタルツインは、予知保全、リモートモニタリング、生産最適化をサポートします。

ディメンションテーブル

[スタースキーマ](#)では、ファクトテーブル内の量的データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は通常、テキストフィールドまたはテキストのように動作する離散数値です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けに一般的に使用されます。

ディザスタ

ワークロードまたはシステムが、導入されている主要な場所でのビジネス目標の達成を妨げるイベント。これらのイベントは、自然災害、技術的障害、または意図しない設定ミスやマルウェア攻撃などの人間の行動の結果である場合があります。

ディザスタリカバリ (DR)

[災害](#)によるダウンタイムとデータ損失を最小限に抑えるために使用する戦略とプロセス。詳細については、AWS Well-Architected [フレームワークの「でのワークロードのディザスタリカバリ AWS: クラウドでのリカバリ」](#)を参照してください。

DML

[「データベース操作言語」](#)を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計:ソフトウェアの中心における複雑さへの取り組み)で紹介されています (ボストン: Addison-Wesley Professional、2003)。strangler fig パターンでドメイン駆動型設計を使用する方法については、[「従来の Microsoft のモダナイズ」を参照してくださいASP.NET \(ASMX\) コンテナと Amazon API Gateway を使用してウェブサービスを段階的に行う。](#)

DR

[「ディザスタリカバリ」](#)を参照してください。

ドリフト検出

ベースライン設定からの偏差の追跡。例えば、AWS CloudFormation を使用して[システムリソースのドリフトを検出したり](#)、を使用して AWS Control Tower ガバナンス要件への準拠に影響を与える可能性のある[ランディングゾーンの変更を検出したり](#)できます。

DVSM

[「開発値ストリームマッピング」](#)を参照してください。

E

EDA

[「探索的データ分析」](#)を参照してください。

エッジコンピューティング

IoT ネットワークのエッジにあるスマートデバイスの計算能力を高めるテクノロジー。[クラウドコンピューティング](#)と比較すると、エッジコンピューティングは通信レイテンシーを短縮し、応答時間を短縮できます。

暗号化

人間が読み取り可能なプレーンテキストデータを暗号文に変換するコンピューティングプロセス。

暗号化キー

暗号化アルゴリズムが生成した、ランダム化されたビットからなる暗号文字列。キーの長さは決まっておらず、各キーは予測できないように、一意になるように設計されています。

エンディアン

コンピュータメモリにバイトが格納される順序。ビッグエンディアンシステムでは、最上位バイトが最初に格納されます。リトルエンディアンシステムでは、最下位バイトが最初に格納されます。

エンドポイント

[「サービスエンドポイント」](#)を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) でホストして他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これらのアカウントまたはプリンシパルは、インターフェイスエンドポイントを作成することで、VPCエンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon) [ドキュメントの「エンドポイントサービスの作成」](#)を参照してください。VPC

エンタープライズリソース計画 (ERP)

エンタープライズの主要なビジネスプロセス (アカウンティング、プロジェクト管理など) を自動化[MES](#)および管理するシステム。

エンベロープ暗号化

暗号化キーを、別の暗号化キーを使用して暗号化するプロセス。詳細については、AWS Key Management Service (AWS KMS) [ドキュメントの「エンベロープ暗号化」](#)を参照してください。

環境

実行中のアプリケーションのインスタンス。クラウドコンピューティングにおける一般的な環境の種類は以下のとおりです。

- 開発環境 — アプリケーションのメンテナンスを担当するコアチームのみが利用できる、実行中のアプリケーションのインスタンス。開発環境は、上位の環境に昇格させる変更をテストするときに使用します。このタイプの環境は、テスト環境と呼ばれることもあります。
- 下位環境 — 初期ビルドやテストに使用される環境など、アプリケーションのすべての開発環境。
- 本番環境 — エンドユーザーがアクセスできる、実行中のアプリケーションのインスタンス。CI/CD パイプラインでは、本番環境が最後のデプロイ環境になります。
- 上位環境 — コア開発チーム以外のユーザーがアクセスできるすべての環境。これには、本番環境、本番前環境、ユーザー承認テスト環境などが含まれます。

エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、AWS CAFセキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。AWS 移行戦略のエピックの詳細については、[プログラム実装ガイド](#) を参照してください。

ERP

[「エンタープライズリソース計画」](#) を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、サマリー統計を計算し、データの視覚化を作成することによって実行されます。

F

ファクトテーブル

[スタースキーマ](#) の中央テーブル。事業運営に関する定量的データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 種類の列が含まれます。

フェイルファスト

頻繁で段階的なテストを使用して開発ライフサイクルを短縮する哲学。これはアジャイルアプローチの重要な部分です。

障害分離境界

では AWS クラウド、アベイラビリティゾーン AWS リージョン、コントロールプレーン、データプレーンなどの境界で、障害の影響を制限し、ワークロードの耐障害性を向上させるのに役立ちます。詳細については、[AWS「障害分離境界」](#)を参照してください。

機能ブランチ

[「ブランチ」](#)を参照してください。

特徴量

お客様が予測に使用する入力データ。例えば、製造コンテキストでは、特徴量は製造ラインから定期的にキャプチャされるイメージの可能性もあります。

特徴量重要度

モデルの予測に対する特徴量の重要性。これは通常、Shapley Additive Explanations (SHAP) や積分勾配など、さまざまな手法で計算できる数値スコアとして表されます。詳細については、[「を使用した機械学習モデルの解釈可能性 : AWS」](#)を参照してください。

機能変換

追加のソースによるデータのエンリッチ化、値のスケーリング、単一のデータフィールドからの複数の情報セットの抽出など、機械学習プロセスのデータを最適化すること。これにより、機械学習モデルはデータの恩恵を受けることができます。例えば、「2021-05-27 00:15:37」の日付を「2021年」、「5月」、「木」、「15」に分解すると、学習アルゴリズムがさまざまなデータコンポーネントに関連する微妙に異なるパターンを学習するのに役立ちます。

FGAC

[「きめ細かなアクセスコントロール」](#)を参照してください。

きめ細かなアクセスコントロール (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

段階的なアプローチを使用するのではなく、[変更データキャプチャ](#)による継続的なデータレプリケーションを使用して、可能な限り短い時間でデータを移行するデータベース移行方法。目的はダウンタイムを最小限に抑えることです。

G

ジオブロッキング

[「地理的制限」](#)を参照してください。

地理的制限 (ジオブロッキング)

Amazon では CloudFront、特定の国のユーザーがコンテンツディストリビューションにアクセスできないようにするオプションです。アクセスを許可する国と禁止する国は、許可リストまたは禁止リストを使って指定します。詳細については、CloudFront ドキュメントの[「コンテンツの地理的ディストリビューションの制限」](#)を参照してください。

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローはレガシーと見なされ、[トランクベースのワークフロー](#)はモダンで推奨されるアプローチです。

グリーンフィールド戦略

新しい環境に既存のインフラストラクチャが存在しないこと。システムアーキテクチャにグリーンフィールド戦略を導入する場合、既存のインフラストラクチャ (別名[ブラウンフィールド](#)) との互換性の制約を受けることなく、あらゆる新しいテクノロジーを選択できます。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

ガードレール

組織単位 () 全体のリソース、ポリシー、コンプライアンスの管理に役立つ大まかなルールOU。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。これらは、サービスコントロールポリシーとIAMアクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは、AWS Config、Amazon AWS Security Hub、GuardDuty、Amazon Inspector AWS Trusted Advisor、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

[「高可用性」](#)を参照してください。

異種混在データベースの移行

別のデータベースエンジンを使用するターゲットデータベースへお客様の出典データベースの移行 (例えば、Oracle から Amazon Aurora)。異種間移行は通常、アーキテクチャの再設計作業の一部であり、スキーマの変換は複雑なタスクになる可能性があります。[AWS は、スキーマの変換に役立つ AWS SCTを提供します。](#)

ハイアベイラビリティ (HA)

課題や災害が発生した場合に、介入なしにワークロードを継続的に運用できること。HA システムは、自動的にフェイルオーバーし、一貫して高品質のパフォーマンスを提供し、パフォーマンスへの影響を最小限に抑えながらさまざまな負荷や障害を処理するように設計されています。

ヒストリアンのモダナイゼーション

製造業のニーズによりよく応えるために、オペレーションテクノロジー (OT) システムをモダナイズし、アップグレードするためのアプローチ。ヒストリアンは、工場内のさまざまなソースからデータを収集して保存するために使用されるデータベースの一種です。

同種データベースの移行

ソースデータベースを、同じデータベースエンジンを共有するターゲットデータベースに移行する (Microsoft SQL Server から Amazon RDS for SQL Server など)。同種間移行は、通常、リホストまたはリプラットフォーム化の作業の一部です。ネイティブデータベースユーティリティを使用して、スキーマを移行できます。

ホットデータ

リアルタイムデータや最近の翻訳データなど、頻繁にアクセスされるデータ。通常、このデータには高速なクエリ応答を提供する高性能なストレージ階層またはクラスが必要です。

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性のため、通常、修正は一般的な DevOps リリースワークフローの外で行われます。

ハイパーケア期間

カットオーバー直後、移行したアプリケーションを移行チームがクラウドで管理、監視して問題に対処する期間。通常、この期間は 1~4 日です。ハイパーケア期間が終了すると、アプリケーションに対する責任は一般的に移行チームからクラウドオペレーションチームに移ります。

|

IaC

[「Infrastructure as Code」](#) を参照してください。

ID ベースのポリシー

AWS クラウド 環境内のアクセス許可を定義する 1 つ以上のIAMプリンシパルにアタッチされたポリシー。

アイドル状態のアプリケーション

90 日間の平均使用量CPUとメモリ使用量が 5~20% のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するか、オンプレミスに保持するのが一般的です。

IIoT

[「産業モノのインターネット」](#) を参照してください。

イミュータブルインフラストラクチャ

既存のインフラストラクチャを更新、パッチ適用、または変更するのではなく、本番ワークロード用の新しいインフラストラクチャをデプロイするモデル。イミュータブルなインフラストラクチャは、[本質的にミュータブルなインフラストラクチャ](#) よりも一貫性、信頼性、予測性が高くなります。詳細については、AWS Well-Architected フレームワークの[「変更不可能なインフラストラクチャを使用したデプロイ」](#) のベストプラクティスを参照してください。

インバウンド (インGRESS) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション外からのネットワーク接続を受け入れ、検査し、ルーティングVPCする。[AWS セキュリティリファレンスアーキテクチャ](#) では、アプリケーションとより広範なインターネット間の双方向インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、検査でネットワークアカウントを設定することをお勧めします。

増分移行

アプリケーションを 1 回ですべてカットオーバーするのではなく、小さい要素に分けて移行するカットオーバー戦略。例えば、最初は少数のマイクロサービスまたはユーザーのみを新しいシステムに移行する場合があります。すべてが正常に機能することを確認できたら、残りのマイクロサービスやユーザーを段階的に移行し、レガシーシステムを廃止できるようにします。この戦略により、大規模な移行に伴うリスクが軽減されます。

|

インダストリー 4.0

2016 年に [Klaus Schwab](#) によって導入された用語で、接続、リアルタイムデータ、自動化、分析、AI/ML の進歩によるビジネスプロセスのモダナイズを指します。

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

アプリケーションのインフラストラクチャを一連の設定ファイルを使用してプロビジョニングし、管理するプロセス。IaC は、新しい環境を再現可能で信頼性が高く、一貫性のあるものにするため、インフラストラクチャを一元的に管理し、リソースを標準化し、スケールを迅速に行えるように設計されています。

産業モノのインターネット (IIoT)

製造、エネルギー、自動車、ヘルスケア、ライフサイエンス、農業などの産業部門におけるインターネットに接続されたセンサーやデバイスの使用。詳細については、[「産業モノのインターネット \(IIoT\) デジタルトランスフォーメーション戦略の構築」](#)を参照してください。

検査 VPC

AWS マルチアカウントアーキテクチャでは、VPCs (同一または異なる 内 AWS リージョン)、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査VPCを管理する一元化されたです。[AWS セキュリティリファレンスアーキテクチャ](#)では、アプリケーションとより広範なインターネット間の双方向インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、検査でネットワークアカウントを設定することをお勧めします。

IoT

インターネットまたはローカル通信ネットワークを介して他のデバイスやシステムと通信する、センサーまたはプロセッサが組み込まれた接続済み物理オブジェクトのネットワーク。詳細については、[「IoT とは」](#)を参照してください。

解釈可能性

機械学習モデルの特性で、モデルの予測がその入力にどのように依存するかを人間が理解できる度合いを表します。詳細については、[「による機械学習モデルの解釈可能性AWS」](#)を参照してください。

IoT

[「モノのインターネット」](#)を参照してください。

IT 情報ライブラリ (ITIL)

IT サービスを提供し、これらのサービスをビジネス要件に合わせるための一連のベストプラクティス。ITIL は、 の基盤を提供しますITSM。

IT サービス管理 (ITSM)

組織の IT サービスの設計、実装、管理、およびサポートに関連する活動。クラウドオペレーションとITSMツールの統合については、 [「オペレーション統合ガイド」](#) を参照してください。

ITIL

[「IT 情報ライブラリ」](#) を参照してください。

ITSM

[「IT サービス管理」](#) を参照してください。

L

ラベルベースのアクセスコントロール (LBAC)

ユーザーとデータ自体にそれぞれセキュリティラベル値が明示的に割り当てられている必須アクセスコントロール (MAC) の実装。ユーザーセキュリティラベルとデータセキュリティラベルが交差する部分によって、ユーザーに表示される行と列が決まります。

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロードとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、 [安全でスケーラブルなマルチアカウント AWS 環境のセットアップ](#) を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

[「ラベルベースのアクセスコントロール」](#) を参照してください。

最小特権

タスクの実行には必要最低限の権限を付与するという、セキュリティのベストプラクティス。詳細については、IAMドキュメントの「[最小特権のアクセス許可を適用する](#)」を参照してください。

リフトアンドシフト

[「7R」](#)を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。[エンディアンネス](#)も参照してください。

下位環境

[「環境」](#)を参照してください。

M

機械学習 (ML)

パターン認識と学習にアルゴリズムと手法を使用する人工知能の一種。MLは、モノのインターネット (IoT) データなどの記録されたデータを分析して学習し、パターンに基づく統計モデルを生成します。詳細については、「[機械学習](#)」を参照してください。

メインブランチ

[「ブランチ」](#)を参照してください。

マルウェア

コンピュータのセキュリティまたはプライバシーを侵害するように設計されているソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスにつながる可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWSのサービスがインフラストラクチャレイヤー、オペレーティングシステム、プラットフォームAWSを運用し、ユーザーがエンドポイントにアクセスしてデータを保存および取得します。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するためのソフトウェアシステム。これにより、加工品を現場の完成製品に変換します。

MAP

[「移行促進プログラム」](#)を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、調整のために結果を検査する完全なプロセス。メカニズムは、動作中にそれ自体を強化して改善するサイクルです。詳細については、AWS Well-Architected フレームワークの[「メカニズムの構築」](#)を参照してください。

メンバーアカウント

の組織の一部である管理アカウント AWS アカウント を除くすべての AWS Organizations。アカウントが組織のメンバーになることができるのは、一度に1つのみです。

MES

[「製造実行システム」](#)を参照してください。

メッセージキューイングテレメトリトランスポート (MQTT)

リソースに制約のある IoT デバイス用の、[パブリッシュ/サブスクライブ](#)パターンに基づく軽量の machine-to-machine (M2M) 通信プロトコル。

マイクロサービス

明確に定義された上で通信APIsし、通常は小規模で自己完結型のチームが所有する小規模で独立したサービス。例えば、保険システムには、販売やマーケティングなどのビジネス機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、[AWS 「サーバーレスサービスを使用したマイクロサービスの統合」](#)を参照してください。

マイクロサービスアーキテクチャ

各アプリケーションプロセスをマイクロサービスとして実行する独立したコンポーネントを使用してアプリケーションを構築するアプローチ。これらのマイクロサービスは、軽量なを使用して明確に定義されたインターフェイスを介して通信しますAPIs。このアーキテクチャの各マイクロサービスは、アプリケーションの特定の機能に対する需要を満たすように更新、デプロイ、およびスケーリングできます。詳細については、「[でのマイクロサービスの実装 AWS](#)」を参照してください。

Migration Acceleration Program (MAP)

コンサルティングサポート、トレーニング、サービスを提供し、組織がクラウドへの移行のための強固な運用基盤を構築し、移行の初期コストを相殺するのに役立つ AWS プログラム。MAP には、従来の移行を系統的に実行するための移行方法論と、一般的な移行シナリオを自動化して高速化するための一連のツールが含まれています。

大規模な移行

アプリケーションポートフォリオの大部分を次々にクラウドに移行し、各ウェーブでより多くのアプリケーションを高速に移動させるプロセス。この段階では、以前の段階から学んだベストプラクティスと教訓を使用して、移行ファクトリー チーム、ツール、プロセスのうち、オートメーションとアジャイルデリバリーによってワークロードの移行を合理化します。これは、[AWS 移行戦略](#) の第 3 段階です。

移行ファクトリー

自動化された俊敏性のあるアプローチにより、ワークロードの移行を合理化する部門横断的なチーム。移行ファクトリーチームには、通常、オペレーション、ビジネスアナリストと所有者、移行エンジニア、デベロッパー、スプリントに取り組む DevOps プロフェッショナルが含まれます。エンタープライズアプリケーションポートフォリオの 20~50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、このコンテンツセットの[移行ファクトリーに関する解説](#)と[Cloud Migration Factory ガイド](#)を参照してください。

移行メタデータ

移行を完了するために必要なアプリケーションおよびサーバーに関する情報。移行パターンごとに、異なる一連の移行メタデータが必要です。移行メタデータの例には、ターゲットサブネット、セキュリティグループ、AWS アカウントなどがあります。

移行パターン

移行戦略、移行先、および使用する移行アプリケーションまたはサービスを詳述する、反復可能な移行タスク。例: Application Migration Service EC2を使用して Amazon AWS への移行をリホストします。

移行ポートフォリオ評価 (MPA)

に移行するためのビジネスケースを検証するための情報を提供するオンラインツール AWS クラウド。MPA は、詳細なポートフォリオ評価 (サーバーの適切なサイズ設定、料金設定、TCO 比較、移行コスト分析) と移行計画 (アプリケーションデータ分析とデータ収集、アプリケーショングループ化、移行の優先順位付け、ウェーブプランニング) を提供します。[MPA ツール](#) (ロギ

ンが必要)は、すべてのAWS コンサルタントとAPNパートナーコンサルタントが無料で利用できます。

移行準備状況評価 (MRA)

を使用して、組織のクラウドの準備状況に関するインサイトを取得し、長所と短所を特定し、特定されたギャップを埋めるためのアクションプランを構築するプロセスAWS CAF。詳細については、[移行準備状況ガイド](#)を参照してください。MRAは[AWS 移行戦略の最初のフェーズ](#)です。

移行戦略

ワークロードを に移行するために使用されるアプローチAWS クラウド。詳細については、この用語集の「[7 Rs エントリ](#)」と「[組織を動員して大規模な移行を加速する](#)」を参照してください。

ML

[「機械学習」](#)を参照してください。

モダナイゼーション

古い(レガシーまたはモノリシック)アプリケーションとそのインフラストラクチャをクラウド内の俊敏で弾力性のある高可用性システムに変換して、コストを削減し、効率を高め、イノベーションを活用します。詳細については、「」の「[アプリケーションをモダナイズするための戦略AWS クラウド](#)」を参照してください。

モダナイゼーション準備状況評価

組織のアプリケーションのモダナイゼーションの準備状況を判断し、利点、リスク、依存関係を特定し、組織がこれらのアプリケーションの将来の状態をどの程度適切にサポートできるかを決定するのに役立つ評価。評価の結果として、ターゲットアーキテクチャのブループリント、モダナイゼーションプロセスの開発段階とマイルストーンを詳述したロードマップ、特定されたギャップに対処するためのアクションプランが得られます。詳細については、「」の「[アプリケーションのモダナイゼーション準備状況の評価AWS クラウド](#)」を参照してください。

モノリシックアプリケーション (モノリス)

緊密に結合されたプロセスを持つ単一のサービスとして実行されるアプリケーション。モノリシックアプリケーションにはいくつかの欠点があります。1つのアプリケーション機能エクスペリエンスの需要が急増する場合は、アーキテクチャ全体をスケーリングする必要があります。モノリシックアプリケーションの特徴を追加または改善することは、コードベースが大きくなると複雑になります。これらの問題に対処するには、マイクロサービスアーキテクチャを使用できます。詳細については、[モノリスをマイクロサービスに分解する](#)を参照してください。

MPA

[「移行ポートフォリオ評価」](#)を参照してください。

MQTT

[「Message Queuing Telemetry Transport」](#)を参照してください。

多クラス分類

複数のクラスの予測を生成するプロセス(2つ以上の結果の1つを予測します)。例えば、機械学習モデルが、「この製品は書籍、自動車、電話のいずれですか?」または、「このお客様にとって最も関心のある商品のカテゴリはどれですか?」と聞くかもしれません。

変更可能なインフラストラクチャ

本番ワークロードの既存のインフラストラクチャを更新および変更するモデル。Well-Architected AWS Framework では、一貫性、信頼性、予測可能性を向上させるために、[イミュータブルインフラストラクチャ](#)の使用をベストプラクティスとして推奨しています。

O

OAC

[「オリジンアクセスコントロール」](#)を参照してください。

OAI

[「オリジンアクセスアイデンティティ」](#)を参照してください。

OCM

[「組織変更管理」](#)を参照してください。

オフライン移行

移行プロセス中にソースワークロードを停止させる移行方法。この方法はダウンタイムが長くなるため、通常は重要ではない小規模なワークロードに使用されます。

OI

[「オペレーション統合」](#)を参照してください。

OLA

[「運用レベルの契約」](#)を参照してください。

オンライン移行

ソースワークロードをオフラインにせずにターゲットシステムにコピーする移行方法。ワークロードに接続されているアプリケーションは、移行中も動作し続けることができます。この方法はダウンタイムがゼロから最小限で済むため、通常は重要な本番稼働環境のワークロードに使用されます。

OPC-UA

[「Open Process Communications - Unified Architecture」](#)を参照してください。

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

産業用オートメーション用の machine-to-machine (M2M) 通信プロトコル。OPC-UA は、データの暗号化、認証、認可スキームを備えた相互運用性標準を提供します。

運用レベルの契約 (OLA)

サービスレベルアグリーメント (SLA) をサポートするために、どの機能 IT グループが相互に提供することを約束するかを明確にする契約 SLA。

運用準備状況レビュー (ORR)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小に役立つ質問とそれに関連するベストプラクティスのチェックリスト。詳細については、AWS Well-Architected フレームワークの [「運用準備状況レビュー \(ORR\)」](#) を参照してください。

運用テクノロジー (OT)

産業運用、機器、インフラストラクチャを制御するために物理環境と連携するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、[Industry 4.0](#) トランスフォーメーションの主要な焦点です。

オペレーション統合 (OI)

クラウドでオペレーションをモダナイズするプロセスには、準備計画、オートメーション、統合が含まれます。詳細については、[オペレーション統合ガイド](#) を参照してください。

組織の証跡

組織 AWS アカウント 内のすべてのイベントをログ AWS CloudTrail に記録するによって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウントに作成され、各アカウントのアクティビティを追跡します。詳細については、ドキュメントの [「組織の証跡の作成」](#) を参照してください。CloudTrail

組織の変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCMは、変化の導入を加速し、移行に伴う問題に対処し、文化的および組織的な変化を推進することで、組織が新しいシステムや戦略の準備と移行を支援します。AWS移行戦略では、クラウド導入プロジェクトに必要な変化のスピードから、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、[OCM「」ガイド](#)を参照してください。

オリジンアクセスコントロール (OAC)

では CloudFront、Amazon Simple Storage Service (Amazon S3) コンテンツを保護するためのアクセスを制限するための拡張オプションです。OACは、すべてののすべての S3 バケット AWS リージョン、AWS KMS (-SSEKMS) によるサーバー側の暗号化、S3 バケットへの動的 PUT および DELETE リクエストをサポートします。

オリジンアクセスアイデンティティ (OAI)

では CloudFront、Amazon S3 コンテンツを保護するためのアクセスを制限するオプションです。を使用するとOAI、は Amazon S3 が認証できるプリンシパル CloudFront を作成します。認証されたプリンシパルは、特定の CloudFront ディストリビューションを介してのみ S3 バケット内のコンテンツにアクセスできます。「」も参照してください。これにより[OAC](#)、より詳細で拡張されたアクセスコントロールが提供されます。

ORR

[「運用準備状況レビュー」](#)を参照してください。

OT

[「運用技術」](#)を参照してください。

アウトバウンド (出力) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション内から開始されるネットワーク接続VPCを処理する。[AWS セキュリティリファレンスアーキテクチャ](#)では、アプリケーションとより広範なインターネット間の双方向インターフェイスVPCsを保護するために、インバウンド、アウトバウンド、検査でネットワークアカウントを設定することをお勧めします。

P

アクセス許可の境界

ユーザーまたはロールが持つことができるアクセス許可の上限を設定するためにプリンIAMシパルにアタッチされるIAM管理ポリシー。詳細については、IAMドキュメントの[「アクセス許可の境界」](#)を参照してください。

個人を特定できる情報 (PII)

直接閲覧した場合、または他の関連データと組み合わせた場合に、個人の身元を合理的に推測するために使用できる情報。例としてPIIは、名前、住所、連絡先情報などがあります。

PII

[個人を特定できる情報を参照してください。](#)

プレイブック

クラウドでのコアオペレーション機能の提供など、移行に関連する作業を取り込む、事前定義された一連のステップ。プレイブックは、スクリプト、自動ランブック、またはお客様のモダナイズされた環境を運用するために必要なプロセスや手順の要約などの形式をとることができます。

PLC

[「プログラム可能なロジックコントローラー」](#)を参照してください。

PLM

[「製品ライフサイクル管理」](#)を参照してください。

ポリシー

アクセス許可の定義 ([アイデンティティベースのポリシー](#) を参照)、アクセス条件の指定 ([リソーススペースのポリシー](#) を参照)、または の組織内のすべてのアカウントに対する最大アクセス許可の定義 AWS Organizations ([サービスコントロールポリシー](#) を参照) が可能なオブジェクト。

多言語の永続性

データアクセスパターンやその他の要件に基づいて、マイクロサービスのデータストレージテクノロジーを個別に選択します。マイクロサービスが同じデータストレージテクノロジーを使用している場合、実装上の問題が発生したり、パフォーマンスが低下する可能性があります。マイクロサービスは、要件に最も適合したデータストアを使用すると、より簡単に実装でき、パフォーマンスとスケーラビリティが向上します。詳細については、[マイクロサービスでのデータ永続性の有効化](#) を参照してください。

ポートフォリオ評価

移行を計画するために、アプリケーションポートフォリオの検出、分析、優先順位付けを行うプロセス。詳細については、「[移行準備状況ガイド](#)」を参照してください。

述語

true または を返すクエリ条件。false 通常は WHERE 句にあります。

述語プッシュダウン

転送前にクエリ内のデータをフィルタリングするデータベースクエリ最適化手法。これにより、リレーショナルデータベースから取得して処理する必要があるデータの量が減少し、クエリのパフォーマンスが向上します。

予防的コントロール

イベントの発生を防ぐように設計されたセキュリティコントロール。このコントロールは、ネットワークへの不正アクセスや好ましくない変更を防ぐ最前線の防御です。詳細については、Implementing security controls on AWSの[Preventative controls](#)を参照してください。

プリンシパル

アクションを実行し AWS、リソースにアクセスできる のエンティティ。このエンティティは通常、IAMロール AWS アカウント、または ユーザーのルートユーザーです。詳細については、IAMドキュメントの「[ロールの用語と概念](#)」の「プリンシパル」を参照してください。

プライバシーバイデザイン

エンジニアリングプロセス全体を通してプライバシーを考慮に入れたシステムエンジニアリングのアプローチ。

プライベートホストゾーン

Amazon Route 53 が 1 つ以上の 内のドメインとそのサブドメインのDNSクエリにどのように応答するかに関する情報を保持するコンテナVPCs。詳細については、Route 53 ドキュメントの「[プライベートホストゾーンの使用](#)」を参照してください。

プロアクティブコントロール

非準拠のリソースのデプロイを防止するように設計された[セキュリティコントロール](#)。これらのコントロールは、プロビジョニング前にリソースをスキャンします。リソースがコントロールに準拠していない場合、プロビジョニングされません。詳細については、AWS Control Tower ドキュメントの「[コントロールリファレンスガイド](#)」および「[でのセキュリティコントロールの実装](#)」の「[プロアクティブコントロール](#)」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

設計、開発、発売から成長と成熟まで、製品のデータとプロセスのライフサイクル全体にわたる管理、および辞退と削除。

本番環境

[「環境」](#)を参照してください。

プログラム可能なロジックコントローラー (PLC)

製造では、マシンをモニタリングし、承認プロセスを自動化する、信頼性が高く、適応性の高いコンピュータです。

仮名化

データセット内の個人識別子をプレースホルダー値に置き換えるプロセス。仮名化は個人のプライバシー保護に役立ちます。仮名化されたデータは、依然として個人データとみなされます。

パブリッシュ/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。スケーラビリティと応答性を向上させます。例えば、マイクロサービスベースの [MES](#)、マイクロサービスは他のマイクロサービスがサブスクライブできるチャンネルにイベントメッセージを発行できます。システムは、公開サービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

SQL リレーショナルデータベースシステム内のデータにアクセスするために使用される手順などの一連のステップ。

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

[責任、説明責任、相談、通知 \(RACI\)](#) を参照してください。

ランサムウェア

決済が完了するまでコンピュータシステムまたはデータへのアクセスをブロックするように設計された、悪意のあるソフトウェア。

RASCI マトリックス

[責任、説明責任、相談、通知 \(RACI\)](#) を参照してください。

RCAC

[「行と列のアクセスコントロール」](#) を参照してください。

リードレプリカ

読み取り専用で使用されるデータベースのコピー。クエリをリードレプリカにルーティングして、プライマリデータベースへの負荷を軽減できます。

再構築

[「7 Rs」](#) を参照してください。

目標復旧時点 (RPO)

最後のデータリカバリポイントからの最大許容時間です。これにより、最後の回復時点からサービスが中断されるまでの間に許容できるデータ損失の程度が決まります。

目標復旧時間 (RTO)

サービスの中断から復旧までの最大許容遅延時間。

リファクタリング

[「7 R」](#) を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョンは、耐障害性、安定性、耐障害性を提供するために、他のから分離され、独立しています。詳細については、[AWS リージョン「を使用できるアカウントを指定する」](#) を参照してください。

回帰

数値を予測する機械学習手法。例えば、「この家はどれくらいの値段で売れるでしょうか?」という問題を解決するために、機械学習モデルは、線形回帰モデルを使用して、この家に関する既知の事実 (平方フィートなど) に基づいて家の販売価格を予測できます。

リホスト

[「7 R」を参照してください。](#)

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

[「7 R」を参照してください。](#)

プラットフォーム変更

[「7 R」を参照してください。](#)

再購入

[「7 R」を参照してください。](#)

回復性

中断に耐えたり、中断から回復したりするアプリケーションの機能。で障害耐性を計画する場合、[高可用性](#)と[ディザスタリカバリ](#)が一般的な考慮事項です AWS クラウド。詳細については、[AWS クラウド「レジリエンス」](#)を参照してください。

リソースベースのポリシー

Amazon S3 バケット、エンドポイント、暗号化キーなどのリソースにアタッチされたポリシー。このタイプのポリシーは、アクセスが許可されているプリンシパル、サポートされているアクション、その他の満たすべき条件を指定します。

責任、説明責任、相談、情報 (RACI) マトリックス

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R)、説明責任 (A)、協議 (C)、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートを含めると、行列はRASCI行列と呼ばれ、除外すると行RACI列と呼ばれます。

レスポンスコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、Implementing security controls on AWSの[Responsive controls](#)を参照してください。

保持

[「7 Rs」を参照してください。](#)

廃止

[「7 Rs」を参照してください。](#)

ローテーション

攻撃者が認証情報にアクセスすることをより困難にするために、シークレットを定期的に更新するプロセス。

行と列のアクセスコントロール (RCAC)

アクセスルールが定義されている基本的で柔軟なSQL式の使用。RCAC は、行のアクセス許可と列マスクで構成されます。

RPO

「目標[復旧時点](#)」を参照してください。

RTO

「目標[復旧時間](#)」を参照してください。

ランブック

特定のタスクを実行するために必要な手動または自動化された一連の手順。これらは通常、エラー率の高い反復操作や手順を合理化するために構築されています。

S

SAML 2.0

多くの ID プロバイダー (IdPs) が使用するオープンスタンダード。この機能により、フェデレーテッドシングルサインオン (SSO) が有効になるため、ユーザーは AWS Management Console にログインしたり、AWS API組織内のすべてのユーザーIAMに対して でユーザーを作成したりすることなく、オペレーションを呼び出すことができます。2SAML.0 ベースのフェデレーション

の詳細については、IAMドキュメント[SAMLの「2.0 ベースのフェデレーションについて」](#)を参照してください。

SCADA

[「監視コントロールとデータ収集」](#)を参照してください。

SCP

[「サービスコントロールポリシー」](#)を参照してください。

シークレット

では AWS Secrets Manager、暗号化された形式で保存されるパスワードやユーザー認証情報などの機密情報または制限付き情報。シークレット値とそのメタデータで構成されます。シークレット値は、バイナリ、単一の文字列、または複数の文字列にすることができます。詳細については、[Secrets Manager ドキュメントの「Secrets Manager シークレットの内容」](#)を参照してください。

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、[予防的](#)、[検出的](#)、[???応答的](#)、[プロアクティブ](#)の4つの主なタイプがあります。

セキュリティ強化

アタックサーフェスを狭めて攻撃への耐性を高めるプロセス。このプロセスには、不要になったリソースの削除、最小特権を付与するセキュリティのベストプラクティスの実装、設定ファイル内の不要な機能の無効化、といったアクションが含まれています。

セキュリティ情報とイベント管理 (SIEM) システム

セキュリティ情報管理 (SIM) システムとセキュリティイベント管理 (SEM) システムを組み合わせたツールとサービス。SIEM システムは、サーバー、ネットワーク、デバイス、その他のソースからデータを収集、監視、分析して、脅威やセキュリティ違反を検出し、アラートを生成します。

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修正するように設計された、事前定義されたプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ[検出的](#)または[応答的](#)な AWS セキュリティコントロールとして機能します。自動レスポンスアクションの例としては、VPCセキュリティグループの変更、Amazon EC2インスタンスへのパッチ適用、認証情報のローテーションなどがあります。

サーバー側の暗号化

送信先にあるデータの、それを受け取る AWS のサービス による暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCPはガードレールを定義するか、管理者がユーザーまたはロールに委任できるアクションの制限を設定します。を許可リストまたは拒否リストSCPとして使用して、許可または禁止されるサービスまたはアクションを指定できます。詳細については、AWS Organizations ドキュメントの「[サービスコントロールポリシー](#)」を参照してください。

サービスエンドポイント

URL のエンドポイントの AWS のサービス。ターゲットサービスにプログラムで接続するには、エンドポイントを使用します。詳細については、AWS 全般のリファレンスの「[AWS のサービス エンドポイント](#)」を参照してください。

サービスレベルアグリーメント (SLA)

サービスのアップタイムやパフォーマンスなど、IT チームがお客様に提供すると約束したものを明示した合意書。

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

サービスレベルのインジケータによって測定される、サービスの状態を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、お客様はクラウドのセキュリティを担当します。詳細については、[責任共有モデル](#)を参照してください。

SIEM

[「セキュリティ情報とイベント管理システム」](#)を参照してください。

単一障害点 (SPOF)

システムを中断させる可能性のあるアプリケーションの単一の重要なコンポーネントの障害。

SLA

[「サービスレベルアグリーメント」](#)を参照してください。

SLI

[「サービスレベルインジケータ」](#)を参照してください。

SLO

[「サービスレベルの目標」](#)を参照してください。

split-and-seed モデル

モダナイゼーションプロジェクトのスケールアップと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、[「」の「アプリケーションをモダナイズするための段階的アプローチ AWS クラウド」](#)を参照してください。

SPOF

[単一障害点](#)を参照してください。

star スキーマ

トランザクションデータまたは測定データを保存するために1つの大きなファクトテーブルを使用し、データ属性を保存するために1つ以上の小さなディメンションテーブルを使用するデータベースの組織構造。この構造は、[データウェアハウス](#)またはビジネスインテリジェンスの目的で使用するように設計されています。

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主にとって代わるイチジクのつるを例えています。そのパターンは、モノリシックシステムを書き換えるときのリスクを管理する方法として[Martin Fowler](#)により提唱されました。このパターンを適用する方法の例については、[「従来の Microsoft のモダナイズ」](#)を参照してください。ASP.NET (ASMX) ウェブサービスは、[コンテナと Amazon API Gateway](#)を使用して段階的に行います。

サブネット

内の IP アドレスの範囲VPC。サブネットは、1つのアベイラビリティゾーンに存在する必要があります。

監視コントロールとデータ収集 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと生産オペレーションをモニタリングするシステム。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

ユーザーインタラクションをシミュレートして潜在的な問題を検出したり、パフォーマンスをモニタリングしたりする方法でシステムをテストします。[Amazon CloudWatch Synthetics](#) を使用してこれらのテストを作成できます。

T

タグ

AWS リソースを整理するためのメタデータとして機能するキーと値のペア。タグは、リソースの管理、識別、整理、検索、フィルタリングに役立ちます。詳細については、「[AWS リソースのタグ付け](#)」を参照してください。

ターゲット変数

監督された機械学習でお客様が予測しようとしている値。これは、結果変数のことも指します。例えば、製造設定では、ターゲット変数が製品の欠陥である可能性があります。

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要がある一般的なタスクのリストが含まれています。各一般的なタスクには、推定所要時間、所有者、進捗状況が含まれています。

テスト環境

[「環境」](#) を参照してください。

トレーニング

お客様の機械学習モデルに学習するデータを提供すること。トレーニングデータには正しい答えが含まれている必要があります。学習アルゴリズムは入力データ属性をターゲット (お客様が予測したい答え) にマッピングするトレーニングデータのパターンを検出します。これらのパター

ンをキャプチャする機械学習モデルを出力します。そして、お客様が機械学習モデルを使用して、ターゲットがわからない新しいデータでターゲットを予測できます。

トランジットゲートウェイ

VPCs とオンプレミスのネットワークを相互接続するために使用できるネットワークトランジットハブ。詳細については、AWS Transit Gateway ドキュメントの「[トランジットゲートウェイとは](#)」を参照してください。

トランクベースのワークフロー

デベロッパーが機能ブランチで機能をローカルにビルドしてテストし、その変更をメインブランチにマージするアプローチ。メインブランチはその後、開発環境、本番前環境、本番環境に合わせて順次構築されます。

信頼されたアクセス

ユーザーに代わって AWS Organizations およびそのアカウントで組織内でタスクを実行するために指定するサービスへのアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要とときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、ドキュメントの「[AWS Organizations を他の AWS のサービスで使用する AWS Organizations](#)」を参照してください。

チューニング

機械学習モデルの精度を向上させるために、お客様のトレーニングプロセスの側面を変更する。例えば、お客様が機械学習モデルをトレーニングするには、ラベル付けセットを生成し、ラベルを追加します。これらのステップを、異なる設定で複数回繰り返して、モデルを最適化します。

ツーピザチーム

2つのピザを食べることができる小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

予測機械学習モデルの信頼性を損なう可能性がある、不正確、不完全、または未知の情報を指す概念。不確実性には、次の2つのタイプがあります。認識論的不確実性は、限られた、不完全なデータによって引き起こされ、弁論的不確実性は、データに固有のノイズとランダム性によって引き起こされます。詳細については、[深層学習システムにおける不確実性の定量化](#) ガイドを参照してください。

未分化なタスク

ヘビーリフティングとも呼ばれ、アプリケーションの作成と運用には必要だが、エンドユーザーに直接的な価値をもたらさなかったり、競争上の優位性をもたらしたりしない作業です。未分化なタスクの例としては、調達、メンテナンス、キャパシティプランニングなどがあります。

上位環境

[「環境」](#)を参照してください。

V

バキューミング

ストレージを再利用してパフォーマンスを向上させるために、増分更新後にクリーンアップを行うデータベースのメンテナンス操作。

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

プライベート IP アドレスを使用してトラフィックをルーティングVPCsできる 2 つの間の接続。詳細については、Amazon VPCドキュメントの[VPC「ピアリングとは」](#)を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

頻繁にアクセスされる最新の関連データを含むバッファキャッシュ。データベースインスタンスはバッファキャッシュから、メインメモリまたはディスクからよりも短い時間で読み取りを行うことができます。

ウォームデータ

アクセス頻度の低いデータ。この種類のデータをクエリする場合、通常は適度に遅いクエリでも問題ありません。

ウィンドウ関数

現在のレコードに関連する行のグループに対して計算を実行するSQL関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなどのタスクを処理するのに役立ちます。

ワークロード

ビジネス価値をもたらすリソースとコード (顧客向けアプリケーションやバックエンドプロセスなど) の総称。

ワークストリーム

特定のタスクセットを担当する移行プロジェクト内の機能グループ。各ワークストリームは独立していますが、プロジェクト内の他のワークストリームをサポートしています。たとえば、ポートフォリオワークストリームは、アプリケーションの優先順位付け、ウェーブ計画、および移行メタデータの収集を担当します。ポートフォリオワークストリームは、これらの設備を移行ワークストリームで実現し、サーバーとアプリケーションを移行します。

WORM

[「書き込み 1 回」](#)を参照し、[多くの](#)を読み取ります。

WQF

[AWS 「ワークロード認定フレームワーク」](#)を参照してください。

書き込み 1 回、読み取り数 (WORM)

データを 1 回書き込み、データの削除や変更を防ぐストレージモデル。承認されたユーザーは、必要な回数だけデータを読み取ることができますが、変更することはできません。このデータストレージインフラストラクチャは [イミュータブルな](#) と見なされます。

Z

ゼロデイ 익스プロイト

[ゼロデイ脆弱性](#) を利用する攻撃、通常はマルウェア。

ゼロデイ脆弱性

実稼働システムにおける未解決の欠陥または脆弱性。脅威アクターは、このような脆弱性を利用してシステムを攻撃する可能性があります。開発者は、よく攻撃の結果で脆弱性に気付きます。

ゾンビアプリケーション

平均使用量CPUとメモリ使用量が 5% 未満のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するのが一般的です。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。